Tsi721™ User Manual

Formal

July 18, 2013

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.

Table of Contents

Overview

PCIe Interface

RapidIO Interface

Bridging Modules

Secondary Modules

Clocking and Resets

Device Registers

Appendices

List of Figures

This document is confidential and is subject to an NDA.

List of Tables

About this Document

Topics discussed include the following:

- Overview
- Document Conventions
- Revision History

Overview

The *Tsi721 User Manual* discusses the features, capabilities, and configuration requirements for the Tsi721. It is intended for 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 lowercase “n”. An active-high signal has an active state of logic 1 (or the higher voltage level), and is not denoted by a special character. The following table shows the non-differential signal naming convention.

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

Differential Signal Notation

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

State	Single-line signal	Multi-line signal

Inactive	NAME_p = 0
	NAME_n = 1
		NAME_p[3] = 0
		NAME_n[3] = 1

Active	NAME_p = 1
	NAME_n = 0
		NAME_p[3] is 1
		NAME_n[3] is 0

Object Size Notation

- A *byte* is an 8-bit object.
- A PCIe *word* is a 16-bit object.
- A PCIe *doubleword* (dword) is a 32-bit object.
- An S-RIO *word* is a 32-bit object.
- An S-RIO *doubleword* (dword) is a 64-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, 0b10).
- 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 important configuration information or suggestions.

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

Revision History

July 17, 2013, Formal Status

- Updated steps 6, 7, and 9 in Slave Access Examples
- Added SOFT_RST_PORT to the RapidIO PLM Port Implementation Specific Control Register. This bit was previously defined as Reserved.
- Updated the procedure described in How to Re-enable the S-RIO Port?

October 2, 2012, Formal Status

- Added a note to Data Link Layer about ASPM operation
- Added a caution about lookup tables to PCIe Address Translation, Updating Outbound Window Registers, Outbound Window Lookup Table Zone Select Register, Outbound Window Lookup Table Data 0 Register, Outbound Window Lookup Table Data 1 Register, and Outbound Window Lookup Table Data 2 Register
- Changed the reset value of VERSION in JTAG ID Register
- Changed the reset value of RID (Revision ID) in PCI Class Register
- Changed the reset value of DEV_REV in RapidIO Device Information CAR
- Changed the reset value and description of ASPMS in PCIe Link Capabilities Register, and changed the description of ASPM in PCIe Link Control Register
- Changed the reset value of OFFSET in MSI-X Table Offset Register
- Changed the reset value of OFFSET in MSI-X Pending Bit Array Offset Register
- Changed the reset value of in NPDATA in IFB VC0 Flow Control Credit Non-Posted Configuration Register

- Changed the attribute type of the writeable fields to R/W1S in the following registers: Internal Error Reporting Test 0 Register, Uncorrectable Error Emulation Register, Correctable Error Emulation Register, and SerDes Test Mode Lane 1 Error Count Register

February 28, 2012, Formal Status

- Updated “Port failed and port degraded error handling” in Table 54
- Added a note to Bit Error Rate Testing (BERT)
- Added a new section, MRd to Maintenance Read and NREAD Processing
- Added a new paragraph to Transmit TLP Processing, and updated PCIe Device Capabilities 2 Register.CTRS and PCIe Device Control and Status 2 Register.CTV
- Updated the introduction to Latency Measurements
- Added a new section, Hot Extraction and Insertion
- Updated Table 47 and added a note to Table 55
- Updated Figure 19 and Table 21
- Added sub-bullets to Data Transfer Descriptor and to Descriptor List Processing
- Updated Throughput Measurements in Performance section
- Added a note to Methods of Disabling the S-RIO Port. Added a similar note to the following fields:
  - RapidIO Port Control CSR.PORT_LOCKOUT
  - RapidIO Port Control CSR.PORT_DIS
  - RapidIO PLM Port Event Status Register.DLT
  - RapidIO Port Error and Status CSR.OUTPUT_FAIL
- Added a note to RapidIO Port Link Maintenance Request CSR
- Added a note to RapidIO Response Timeout Register
- Added a note to Device Control Register.SR_RST_MODE. Also added the same text as a caution to S-RIO Protocol Reset from Link Partner
- Completed other minor improvements throughout the document

January 13, 2012, Preliminary Status

- Changed the endianness of registers, RapidIO Logical/Transport Layer Error Detect CSR to RapidIO Logical/Transport Layer Control Capture CSR
- Changed GB_5p0 in RapidIO Port Control 2 CSR to a reset value of 1
- Updated the description of TGT_ID_DIS and MTC_TGT_ID_DIS in the RapidIO TLM Port Control Register. Also revised destID Filtering and added ftype Filtering as per this update.
- Updated the description of ERR_RR and ERR_RATE_CNT in the RapidIO Port Error Rate CSR
- Updated IBDMA_PW and OBDMA_PW fields in the Messaging Engine Port-Write Enable Register and Messaging Engine Port-Write Register to indicate support for eight DMA channels
- Added the Inbound Messaging Engine deviceID Register
- Completed other minor improvements throughout the document

April 29, 2011, Preliminary Status

- Updated the Clocking mode descriptions
- Added Bit Error Rate Testing (BERT)
- Added Throughput Measurements and Latency Measurements

- Moved offsets 0x01008–0x0101C from the SR2PC Registers to the RapidIO Extended Error Management Registers
- Added SerDes Tx Control Register. Also, renamed the SerDes Control Register to SerDes Rx Control Register, and updated several fields in the register
- Added ackID Synchronization
- Completed numerous improvements throughout the document

October 29, 2010, Preliminary Status

This version includes numerous improvements throughout the document.

April 15, 2010, Advance Status

This is the first release of the *Tsi721 User Manual*.

Overview

1. Device Overview

Topics discussed include the following:

- Overview
- Features
- Block Diagram
- Typical Applications

1.1 Overview

IDT is the leading supplier of RapidIO® and PCI Express Interconnect solutions, providing a broad portfolio of switches, bridges, IP, and development platforms for defense aerospace, video, imaging, and wireless markets. The Tsi721 is IDT's solution for hardware-based PCIe Gen2 to RapidIO Gen2 protocol conversion in a bridging device.

The Tsi721 converts transactions from PCIe to RapidIO, and vice versa, and provides full line rate bridging at 20 Gbaud. Using the Tsi721, designers can develop heterogeneous systems that leverage the peer-to-peer networking performance of RapidIO while using multiprocessor clusters that may be only PCIe enabled. In addition, applications that require large amounts of data transferred efficiently without processor involvement can be executed using the full line rate of the Tsi721’s Block DMA Engine and Messaging Engine.

Key to the Tsi721 is the hardware bridging functionality that converts PCIe transactions to RapidIO, and vice versa. The Tsi721 supports PCIe non-transparent bridging for transaction mapping. The device has both RapidIO and PCIe endpoints embedded in the bridge, and each of its Block DMA/Messaging DMA channels can buffer up to 8 KB of data on the PCIe side.

1.2 Features

The Tsi721 supports the following features.

1.2.1 PCIe Features

- PCIe 2.1 standard compliant
- 5/2.5 Gbaud link speed
- x4/x2/x1 link width
- 128- and 256-byte maximum payload
- Advanced error reporting
- Internal error reporting
- Lane reversal
- Automatic polarity inversion
- Dynamic port width: x4 drops to x1
- ECRC support
- INTx, MSI, and MSI-X support

- Single virtual channel, VC0
- Single traffic class, TC0
  - Generates only PCIe posted/non-posted TLPs with TC0
  - Generates only PCIe Cpl/CplD TLPs with TC matching their requests
  - Accepts PCIe TLPs with any TC
- Four BARs
  - Prefetchable BAR with 32- or 64-bit addressing for PCIe-to-S-RIO bridging
  - Non-prefetchable BAR with 32- or 64-bit addressing for PCIe-to-S-RIO bridging
  - Non-prefetchable BAR with 32-bit addressing for PCIe MWr to S-RIO doorbell bridging
  - Non-prefetchable BAR with 32-bit addressing for Tsi721 internal register access
- Initial credit advertisement programmable through EEPROM
- Dynamic control of credits through registers
- Starvation prevention based on flow control credit updates
- Large buffers
  - 12 KB/2 KB/12 KB input buffers for up to 127 posted/non-posted/completion TLPs
  - 12 KB/2 KB/12 KB output buffers for up to 128 posted/non-posted/completion TLPs
- Debug features
  - Slave analog loopback through a control register
  - Slave loopback using TS1/TS2 ordered sets
  - Master loopback
  - Internal error reporting
  - ECC protection on internal memories

1.2.2 S-RIO Features

- S-RIO 2.1 standard compliant
- 5/3.125/2.5/1.25 Gbaud link speed
- x4/x2/x1 link width
- 34-, 50-, and 66-bit addressing
- 16 destID filters
- 8 S-RIO flows
- 9-KB ingress buffer (32 x 288)
- 9-KB egress buffer (32 x 288)
- Lane reversal
- Lane polarity inversion

1.2.3 Bridging Features

- Store and forward from PCIe to S-RIO
- Store and forward from S-RIO to PCIe
- Line rate support for 64 byte and larger packets
- 32 outstanding PCIe requests to root complex

- 32 outstanding S-RIO NREAD/maintenance read requests to S-RIO network
- 32 outstanding S-RIO NWRITE_R/maintenance write/doorbell requests to S-RIO network
- 12-KB completion reassembly buffer
- 8 windows from PCIe to S-RIO with 8 zones (sub windows) per window
- 8 windows from S-RIO to PCIe
- Initiates and receives the following S-RIO transactions:
  - NREAD
  - SWRITE/NWRITE/NWRITE_R
  - Maintenance read and write
  - Port-write
  - Doorbell
  - Type 8 response
  - Type 13 response
- Initiates and receives the following PCIe transactions:
  - MWr
  - MRd
  - Cpl
  - CplD
- Round-robin scheduling between Mapping Engine, Block DMA Engine, and Messaging traffic to the S-RIO link
- Round-robin scheduling between Mapping Engine, Block DMA Engine, and Messaging traffic to the PCIe link
- Forward bridge
  - Connects PCIe root complex to S-RIO network
  - PCIe Type 0 configuration header

1.2.4 Messaging Features

- 8 Tx queues with one dedicated messaging DMA engine per Tx queue
- 8 Rx queues with one dedicated messaging DMA engine per Rx queue
- Descriptor prefetch per Tx queue
- 32 outstanding PCIe requests to root complex
- 8-KB message segment reassembly buffer per Tx queue
- Round-robin scheduling among Tx queues
- One outstanding message per Tx queue
- 16 receive contexts per Rx queue

1.2.5 Block DMA Engine Features

- 8 DMA channels
- Each DMA channel can perform DMA writes from root complex to S-RIO network, or DMA reads from S-RIO network to root complex
  - DMA from PCIe port to PCIe port is not supported
  - DMA from S-RIO port to S-RIO port is not supported

- Round-robin scheduling among DMA channels
- DMA descriptors for all channels reside on PCIe side
- Scatter-and-gather with descriptor list
- Supports DMA strides
- Supports up to 64 MB data for a single descriptor
- Supports both read and write descriptors per DMA channel
- Dynamic descriptor chaining
- Flexible addressing modes
  - Linear addressing
  - Constant addressing
- Descriptor prefetch
- 32 outstanding PCIe requests to root complex
- 64 outstanding S-RIO NREAD/maintenance read requests to S-RIO network
- 64 outstanding S-RIO NWRITE_R/maintenance write requests to S-RIO network
- Supports the following S-RIO transactions:
  - NREAD
  - NWRITE
  - SWRITE
  - NWRITE_R
  - Maintenance read
  - Maintenance write

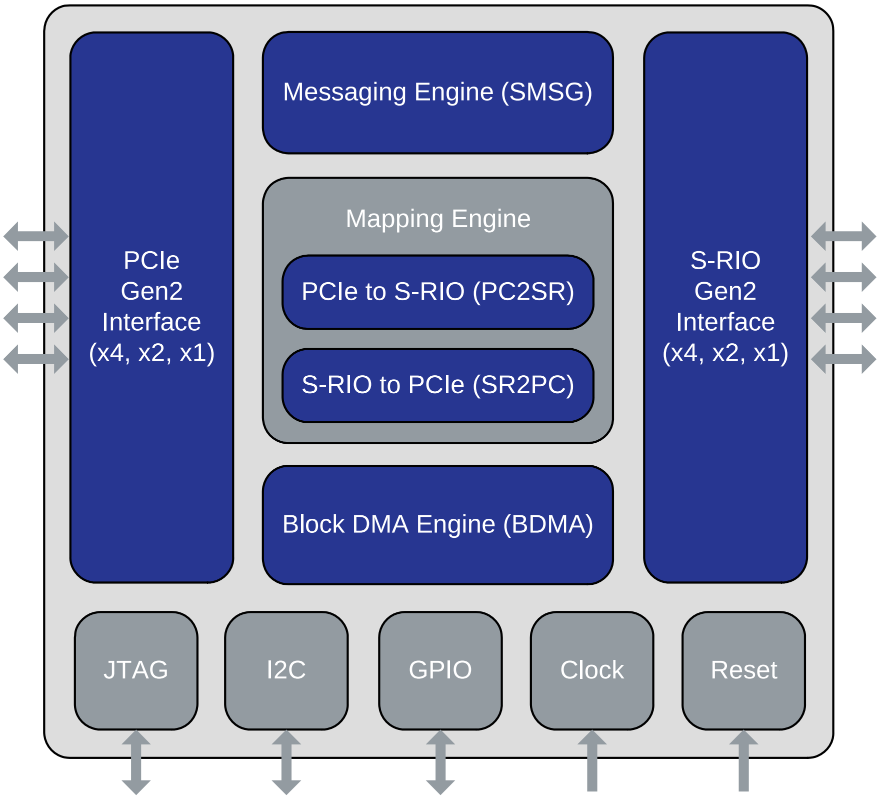
1.2.6 Miscellaneous Features

- I^{2}C interface supports the following:
  - As a slave, being read/written by an external master during normal operations
  - As a master, reading external EEPROM during boot load
  - As a master, reading/writing other external devices during normal operations
- JTAG 1149.1, 1149.6 (AC JTAG)
- 16 GPIO pins

1.3 Block Diagram

The Tsi721 block diagram is displayed in the following figure. The five main functions of the device are briefly described below.

Figure 1: Block Diagram



1.3.1 PCIe Interface

The PCIe Interface performs all the physical, data link, and transport layer protocols associated with PCIe.

1.3.2 S-RIO Interface

The S-RIO Interface performs all the physical and transport layer protocols associated with S-RIO.

1.3.3 Messaging Engine

The Messaging Engine uses S-RIO messaging logical layer functions with dedicated messaging DMA channels per Tx queue and per Rx queue.

1.3.4 Mapping Engine

The Mapping Engine maps between PCIe and S-RIO transactions, including segmentation and reassembly as required.

1.3.5 Block DMA Engine

The Block DMA Engine uses 8 DMA channels, where descriptors of each DMA channel can perform read or write.

This document is confidential and is subject to an NDA.

1.4 Typical Applications

The Tsi721 supports the following typical applications:

- Defense and aerospace
  - Radar
  - Sonar
  - Navigations systems
- Medical imaging
  - CT scanners
  - MRIs
- Video
  - Teleconferencing
  - Head end
- Wireless
  - Baseband cards with x86

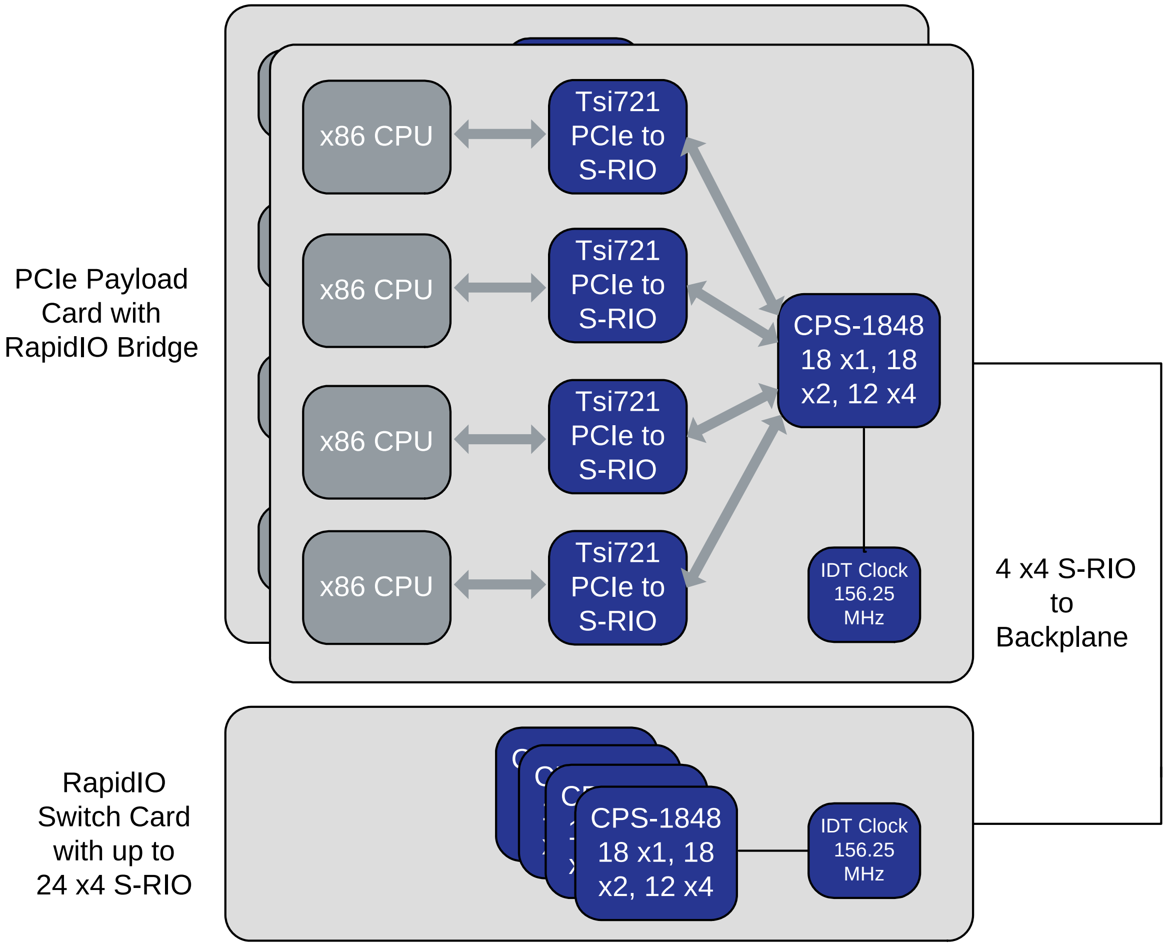
Three of Tsi721’s typical applications – defence/aerospace, video/imaging, and wireless – are discussed in the following sections.

1.4.1 Defence/Aerospace Application

In defence applications, the Tsi721 supports the use of PCIe enabled x86 processors to RapidIO backplanes. This provides system designers with the best of both worlds: the floating point and MIPs horsepower of the latest generation of x86 solutions, with the superior peer-to-peer networking performance of RapidIO architectures.

By using the Tsi721 combined with IDT’s RapidIO Gen2 switches, payload processor cards with x86 processors can be used with existing RapidIO 1.3 backplanes operating at up to 3.125 Gbaud, or the same card can be used with RapidIO Gen2 compatible backplanes operating at 5 Gbaud.

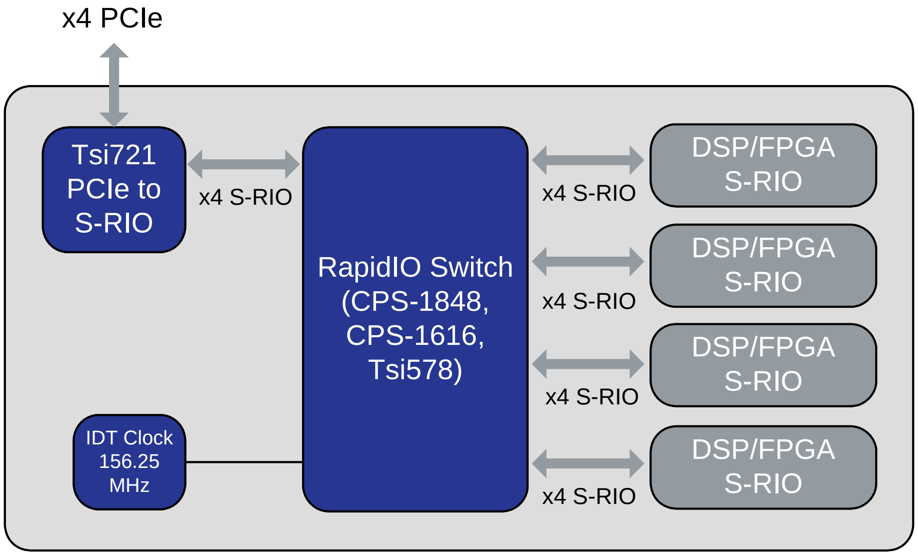
Figure 2: Defence/Aerospace Application



1.4.2 Video and Imaging Application

In video and imaging applications, system designers need to cluster large numbers of DSPs or FPGAs to perform encoding/decoding/trans coding, or do FFTs (Fast Fourier Transform) on large arrays of data. The RapidIO protocol is optimal for this DSP/FPGA cluster requirement. However, the analog front-end to the system is usually a sensor with streaming data terminated in an FPGA (for example, a camera subsystem). This is usually in an PCIe network, often with a PC back-end. In these applications the designer needs to bridge between a PCIe network and the RapidIO DSP/FPGA cluster. The Tsi721 is ideal for this application.

Figure 3: Video and Imaging Application

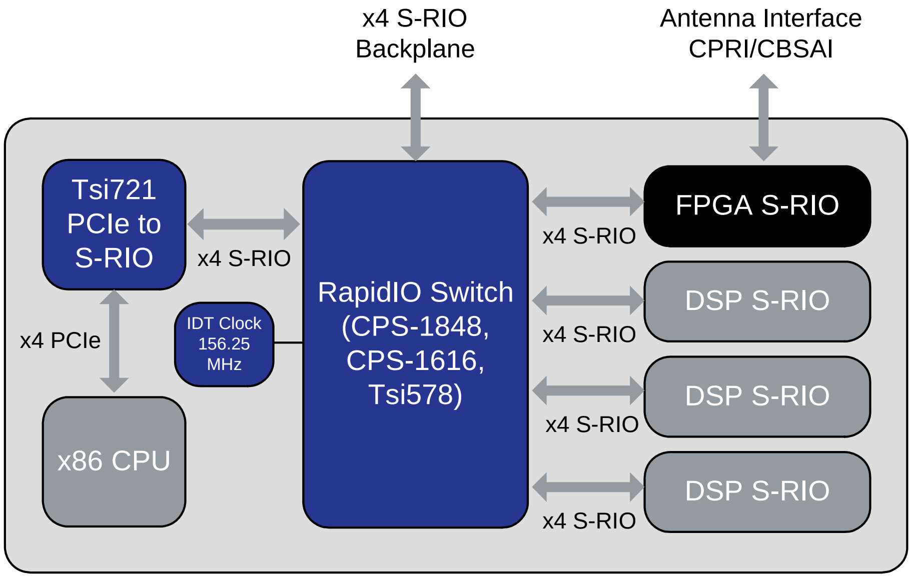


1.4.3 Wireless Application

In wireless base stations, the incumbent interconnect technology in the baseband processing cards – LTE, WiMAX, WCDMA, and TD-SCDMA – is RapidIO. RapidIO connects a cluster of DSPs, processor, and FPGA, locally on the baseband processor for MAC and PHY layer processing. However, the LTE standard pushes the performance available in existing RapidIO enabled microprocessors.

The Tsi721 provides wireless OEMs with an additional option to use an x86 processor with superior MIPs in a baseband card that is predominantly RapidIO. In these card designs, RapidIO is the interconnect between devices and functions as the backplane interconnect. x86 processors can now be used with other RapidIO devices on the baseband card and leverage the messaging performance of RapidIO for this peer-to-peer multiprocessor network.

Figure 4: Wireless Application



2. Data Path

Topics discussed include the following:

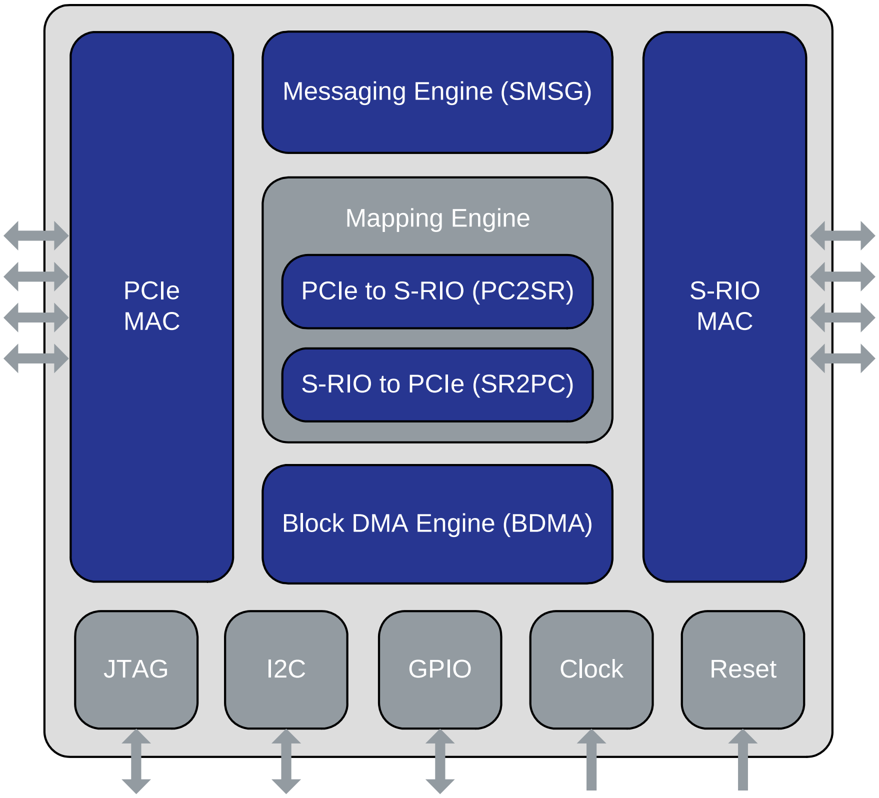
- Overview
- Ordering Rules
- Loopbacks
- Performance
- Endian Conversion
- Data Protection

2.1 Overview

As displayed in Figure 5, the Tsi721 contains the following blocks:

- PCIe MAC
- S-RIO MAC
- Block DMA Engine (BDMA)
- Mapping Engine
  - — PCIe to S-RIO mapper (PC2SR) as part of Mapping Engine
  - — S-RIO to PCIe mapper (SR2PC) as part of Mapping Engine
- S-RIO Messaging Engine (SMSG)

Figure 5: Data Flow



The Tsi721 has three independent data paths:

1. A *bridging* data path between PCIe and S-RIO that uses the following blocks:
   - PCIe MAC
   - Mapping Engine (PC2SR, SR2PC)
   - S-RIO MAC

   Received doorbells are handled differently than other transactions: they are stored in main memory doorbell queues that are co-managed by Tsi721 and software.
2. A *messaging* data path that uses the following blocks:
   - PCIe MAC
   - Messaging Engine
   - Mapping Engine (PC2SR, SR2PC)
   - S-RIO MAC

   Messages are sourced or stored from/into main memory message queues that are co-managed by Tsi721 and software.
3. A *block DMA engine* data path that uses the following blocks:
   - PCIe MAC
   - Block DMA Engine
   - Mapping Engine (PC2SR, SR2PC)
   - S-RIO MAC

   Block DMA Engine transactions are sourced or stored from/into main memory DMA queues that are co-managed by Tsi721 and software.

The Tsi721 defines inbound direction as S-RIO to PCIe, and outbound direction as PCIe to S-RIO.

2.1.1 PCIe to S-RIO Bridging Data Path

Table 1 uses the following notation: “1: Passes” means step 1; the associated block passes the transaction through the device. “2: Maps” means step 2; the associated block maps the transaction from PCIe to S-RIO or vice versa. Only listed transaction types are supported. All other transaction types are not supported, and Tsi721 generates Unsupported Request (UR) error report.

Table 1: PCIe to S-RIO Bridging Data Path

PCIe Type	Mapping	PCIe MAC	PC2SR	BDMA	SMSG	SR2PC	S-RIO MAC
Zero length MRd	MRd to NREAD	1: Passes MRd	2: Maps MRd to NREAD	N/A	N/A	N/A	3: Passes NREAD
	Response	6: Passes CplD	N/A	N/A	N/A	5: Maps NREAD response to CplD	4: Passes NREAD response
Zero length MWr	N/A	1: Passes MWr	2: Silently discards MWr	N/A	N/A	N/A	N/A
MRd+CplD	MRd-> NREAD	1: Passes MRd	2: Maps MRd to NREAD	N/A	N/A	N/A	3: Passes NREAD
	NREAD response-> CplD/Cpl	6: Passes CplD/Cpl	N/A	N/A	N/A	5: Maps NREAD response to CplD/Cpl	4: Passes NREAD response
MWr	MWr-> NWRITE MWr-> SWRITE	1: Passes MWr	2: Maps MWr to NWRITE/ SWRITE	N/A	N/A	N/A	3: Passes NWRITE/ SWRITE
MWr	MWr-> NWRITE_R	1: Passes MWr	2: Maps MWr to NWRITE_R	N/A	N/A	N/A	3: Passes NWRITE_R
	NWRITE_R response	N/A	N/A	N/A	N/A	5: Terminates NWRITE_R response with OK counting and error log	4: Passes NWRITE_R response

Table 1: PCIe to S-RIO Bridging Data Path *(Continued)*

PCIe Type	Mapping	PCIe MAC	PC2SR	BDMA	SMSG	SR2PC	S-RIO MAC
MWr	MWr->maintenance write	1: Passes MWr	2: Maps MWr to maintenance write	N/A	N/A	N/A	3: Passes maintenance write
	Maintenance write response	N/A	N/A	N/A	N/A	5: Terminates maintenance write response with OK counting and error log	4: Passes maintenance write response
MRd + CplD	MRd->maintenance read	1: Passes MRd	2: Maps MRd to maintenance read	N/A	N/A	N/A	3: Passes maintenance read
	Maintenance read response -> CplD/CPL	6: Passes CplD/Cpl	N/A	N/A	N/A	5: Maps maintenance read response to CplD/Cpl	4: Passes maintenance read response
MWr	MWr-> doorbell	1: Passes MWr	2: Maps MWr to doorbell	N/A	N/A	N/A	3: Passes doorbell
	Doorbell response	N/A	N/A	N/A	N/A	5: Terminates doorbell response with OK counting and error/retry log	4: Passes doorbell response
MWr	Tsi721 register write	1: Passes MWr	2: Terminates MWr	N/A	N/A	N/A	N/A
MRd + CplD	Tsi721 register read	1: Passes MRd	2: Terminates MRd	N/A	N/A	N/A	N/A
	CplD	4: Passes CplD/Cpl	N/A	N/A	N/A	3: Generates CplD/Cpl	N/A
CFGRD0 + CplD	N/A	1: Terminates CFGRD0 2: Generates Cpl/CplD	N/A	N/A	N/A	N/A	N/A

Table 1: PCIe to S-RIO Bridging Data Path *(Continued)*

PCIe Type	Mapping	PCIe MAC	PC2SR	BDMA	SMSG	SR2PC	S-RIO MAC
CFGWR0 + CplD	N/A	1: Terminates CFGWR0 2: Generates Cpl/CplD	N/A	N/A	N/A	N/A	N/A
INTx	N/A	2: Generates messages	N/A	N/A	N/A	1: Generates requests	N/A
MSI	N/A	2: Generates MWr	N/A	N/A	N/A	1: Generates requests	N/A
MSI-X	N/A	2: Passes MWr	N/A	N/A	N/A	1: Generates MWr	N/A
Power Management messages	N/A	1: Terminates messages 2: Generates messages	N/A	N/A	N/A	N/A	N/A
Error signaling messages	N/A	2: Generates messages	1. Generates request	N/A	N/A	N/A	N/A

The Tsi721 does not retry doorbells on receiving a doorbell RETRY response. An external device is required to retry a doorbell when Tsi721 has logged a RETRY response.

2.1.2 S-RIO to PCIe Bridging Data Path

Although doorbells are listed below, received doorbells have different data path than other transactions, and are stored into queues inside main memory that are co-managed by Tsi721 and software.

Table 1 uses the following notation: “1: Passes” means step 1; the associated block passes the transaction through the device. “2: Maps” means step 2; the associated block maps the transaction from S-RIO to PCIe or vice versa. Only listed transaction types are supported. All other transaction types are not supported.

Table 2: S-RIO to PCIe Bridging Data Path

S-RIO Type	Mapping	S-RIO MAC	SR2PC	BDMA	SMSG	PC2SR	PCIe MAC
NREAD hitting an inbound windows (see S-RIO Address Translation) + response	NREAD -> MRd	1: Passes NREAD	2: Maps NREAD to MRd	N/A	N/A	N/A	3: Passes MRd
	CplD/Cpl-> NREAD response	6: Passes NREAD response	N/A	N/A	N/A	5: Maps CplD/ Cpl to NREAD response	4: Passes CplD/ Cpl

Table 2: S-RIO to PCIe Bridging Data Path *(Continued)*

S-RIO Type	Mapping	S-RIO MAC	SR2PC	BDMA	SMSG	PC2SR	PCIe MAC
NREAD missing all inbound windows (see S-RIO Address Translation) + response	NREAD termination	1: Passes NREAD	2: Terminates NREAD	N/A	N/A	N/A	N/A
	NREAD response generation	4: Passes NREAD error response	N/A	N/A	N/A	3: Generates NREAD error response	N/A
NWRITE/SWRITE	NWRITE -> MWr SWRITE-> MWr	1: Passes NWRITE/ SWRITE	2: Maps NWRITE/SWRITE to MWr	N/A	N/A	N/A	3: Passes MWr
NWRITE_R + Response	NWRITE_R->MWr	1: Passes NWRITE_R	2: Maps NWRITE_R to MWr	N/A	N/A	N/A	3a: Passes MWr
	N/A	4b: Passes NWRITE_R response	N/A	N/A	N/A	3b: Generates NWRITE_R response	
Maintenance write + Response	Maintenance write	1: Terminates maintenance write	N/A	N/A	N/A	N/A	N/A
	Maintenance write response	2: Generates maintenance write response	N/A	N/A	N/A	N/A	N/A
Maintenance read + Response	Maintenance read	1: Terminates maintenance read	N/A	N/A	N/A	N/A	N/A
	Maintenance read response	2: Generates maintenance read response	N/A	N/A	N/A	N/A	N/A

Table 2: S-RIO to PCIe Bridging Data Path *(Continued)*

S-RIO Type	Mapping	S-RIO MAC	SR2PC	BDMA	SMSG	PC2SR	PCIe MAC
Doorbell + Response	Doorbell -> MWr	1: Passes doorbell	2: Terminates doorbell and generates MWr	N/A	N/A	N/A	3a: Passes MWr
	Doorbell response	4b: Passes doorbell response	N/A	N/A	N/A	3b: Generates doorbell response	N/A
Port-write generation	N/A	1: Generates	N/A	N/A	N/A	N/A	N/A
Port-write termination	N/A	1: Terminates	N/A	N/A	N/A	N/A	N/A

This document is confidential and is subject to an NDA.

2.1.3 Messaging Engine Data Path

The Messaging Engine supports 8 dedicated outbound DMA channels for outbound messages and 8 dedicated inbound DMA channels for inbound messages.

Table 3: Messaging Engine Data Path

Type	Subtype	SMSG	S-RIO MAC	PC2SR	BDMA	SR2PC	PCIe MAC
Outbound message + response	MRd generation	1: Generates MRd to read msg data	N/A	N/A	N/A	2: Passes data MRd	3: Passes data MRd
	CplD termination + msg generation	6: Terminates CplD and generates msg	N/A	5: Passes CplD	N/A	N/A	4: Passes CplD
	Sending msg	N/A	8: Passes msg	7: Passes msg	N/A	N/A	N/A
	Receiving msg response	N/A	9: Passes msg response	N/A	N/A	10: Passes msg response	N/A
	Msg response termination and MWr generation	11: Terminates msg response and generates status (part of outbound descriptor) update MWr	N/A	N/A	N/A	12: Passes status MWr	13: Passes status MWr
Outbound message descriptor prefetch	Descriptor prefetch MRd generation	1. Generates MRd to prefetch descriptor	N/A	N/A	N/A	2: Passes descriptor MRd	3: Passes descriptor MRd
	CplD termination	6: Terminates CplD	N/A	5: Passes CplD	N/A	N/A	4: Passes CplD

Table 3: Messaging Engine Data Path *(Continued)*

Type	Subtype	SMSG	S-RIO MAC	PC2SR	BDMA	SR2PC	PCIe MAC
Inbound message + response	Receiving msg	N/A	1: Passes msg	N/A	N/A	2: Passes msg	N/A
	Msg termination and data MWr generation	3a: Terminates msg and generates MWr to store msg data	N/A	N/A	N/A	4a: Passes data MWr	5a: Passes data MWr
	Msg termination and msg response generation	3b: Terminates msg and generates msg response	5b: Passes msg response	4b: Passes msg response	N/A	N/A	N/A
	Msg termination and inbound descriptor MWr generation	3c: Terminates msg and generates MWr to store inbound msg descriptor	N/A	N/A	N/A	4c: Passes descriptor MWr	5c: Passes descriptor MWr
Inbound free pointer prefetch	Free pointer fetch MRd generation	1. Generates MRd to fetch free pointer	N/A	N/A	N/A	2: Passes pointer MRd	3: Passes pointer MRd
	CplD termination	6: Terminates CplD	N/A	5: Passes CplD	N/A	N/A	4: Passes CplD

2.1.4 Block DMA Engine Data Path

The Block DMA Engine supports 8 DMA channels and the following transactions:

- PCIe MRd and CplD/Cpl
- PCIe MWr
- S-RIO NWRITE/SWRITE
- S-RIO NWRITE_R
- S-RIO NREAD and response
- S-RIO maintenance write
- S-RIO maintenance read and response

Table 4: Block DMA Engine Data Path[a]

Type	Subtype	BDMA	S-RIO MAC	PC2SR	SMSG	SR2PC	PCIe MAC
DMA read + response	Data MRd generation to fetch read data	1: Generates MRd to fetch data	N/A	N/A	N/A	2: Passes data MRd	3: Passes data MRd
	CplD termination and NREAD/ maintenance read generation	6: Terminates CplD and generates NREAD/ maintenance read	N/A	5: Passes CplD	N/A	N/A	4: Passes CplD
	Sending NREAD/ maintenance read	N/A	8: Passes NREAD/ maintenance read	7: Passes NREAD/ maintenance read	N/A	N/A	N/A
	Receiving NREAD/ maintenance read response	N/A	9: Passes NREAD/ maintenance read response	N/A	N/A	10: Passes NREAD/ maintenance read response	N/A
	NREAD/maintenance read response termination and data MWr generation	11a: Terminates NREAD/ maintenance read response and generates MWr from responses to store data	N/A	N/A	N/A	12a: Passes data MWr	13a: Passes data MWr
	NREAD/maintenance read response termination and status MWr generation	11b: Terminates NREAD/ maintenance read response and generates DMA descriptor status update MWr	N/A	N/A	N/A	12b: Passes status MWr	13b: Passes status MWr

Table 4: Block DMA Engine Data Path[a] *(Continued)*

Type	Subtype	BDMA	S-RIO MAC	PC2SR	SMSG	SR2PC	PCIe MAC
DMA write + response	Data MRd generation	1: Generates MRd to fetch data	N/A	N/A	N/A	2: Passes data MRd	3: Passes data MRd
	CplD termination and NWRITE_R/ maintenance write generation	6: Terminates CplD and generates NWRITE_R/ maintenance write	N/A	5: Passes CplD	N/A	N/A	4: Passes CplD
	Sending NWRITE_R/ maintenance write	N/A	8: Passes NWRITE_R/ maintenance write	7: Passes NWRITE_R/ maintenance write	N/A	N/A	N/A
	Receiving NWRITE_R/ maintenance write response	N/A	9: Passes NWRITE_R/ maintenance write response	N/A	N/A	10: Passes NWRITE_R/ maintenance write response	N/A
	NWRITE_R/ maintenance write response termination and status MWr generation	11: Terminates NWRITE_R/ maintenance write response and generates DMA descriptor status update MWr	N/A	N/A	N/A	12: Passes status MWr	13: Passes status MWr

Table 4: Block DMA Engine Data Path[a] *(Continued)*

Type	Subtype	BDMA	S-RIO MAC	PC2SR	SMSG	SR2PC	PCIe MAC
DMA write without response	Data MRd generation	1: Generates MRd to fetch data	N/A	N/A	N/A	2: Passes data MRd	3: Passes data MRd
	Receiving CplD	N/A	N/A	5: Passes CplD	N/A	N/A	4: Passes CplD
	CplD termination and NWRITE/SWRITE generation	6a: Terminates CplD and generates NWRITE/SWRITE	8a: Passes NWRITE/SWRITE	7a: Passes NWRITE/SWRITE	N/A	N/A	N/A
	CplD termination and status MWr generation	6b: Terminates CplD and generates DMA descriptor status update MWr	N/A	N/A	N/A	7b: Passes status MWr	8b: Passes status MWr
DMA data transfer descriptor prefetch	Descriptor fetch MRd generation	1. Generates MRd to prefetch descriptor	N/A	N/A	N/A	2: Passes descriptor MRd	3: Passes descriptor MRd
	CplD termination	6: Terminates CplD	N/A	5: Passes CplD	N/A	N/A	4: Passes CplD

[a] Immediate data transfer descriptors are not displayed, they are similar but without MRd generation and CplD termination.

2.2 Ordering Rules

2.2.1 Device Ordering Rules

The Tsi721 usually guarantees PCIe/S-RIO ordering rules within each of its three data paths:

- Bridging data path
- Messaging data path
- Block DMA Engine data path

The Tsi721 does not guarantee ordering between different data paths. If ordering is desired between different data paths, applications should follow procedures described in Application Enforced Inter-Data Path Synchronization.

2.2.1.1 PCIe Relaxed Ordering

For both the bridging and Block DMA Engine data paths, the Tsi721 ignores the RO bit in the PCIe header when mapping a PCIe transaction to an S-RIO packet. The Tsi721 does not reorder TLPs based on the RO bit; however, the device makes sure that a PCIe CPL/CplD has the same RO bit as its associated PCIe request. (Note that section 8.4.4 of PCIX spec 1.0b states that a PCI-X bridge is not allowed to reorder posted MWr (only host bridges and endpoints are allowed).

The Tsi721 clears the PCIe header RO bit for all PCIe transactions generated by the device.

2.2.1.2 Tsi721 S-RIO to PCIe Bridging Ordering Rules

For S-RIO transactions bridged to PCIe, the Tsi721 maps them into four queues, where S-RIO transactions are assigned to one of the queues:

1. Completion reassembly queue
   - NREAD response
   - Maintenance read response
2. Posted scratch queue
   - NWRITE request
   - SWRITE request
3. Non-posted scratch queue
   - NREAD request
4. Inbound doorbell queue

The Tsi721 enforces the following PCIe rules for these queues.

Table 5: S-RIO to PCIe Bridging Ordering Rules

Row Pass Column	Posted Scratch Queue	Non-posted Scratch Queue	Completion Reassembly Queue
Completion reassembly queue	No	Yes	No
Posted scratch queue	No	Yes	Yes
Non-posted scratch queue	No	No	No
Inbound doorbell queue	Y/N	Y/N	Y/N

- TLP in a column is older than TLP in a row
- No means that row never passes column
- Yes means that row must pass column upon blocking
- Y/N means that row is unordered relative column and can pass column when blocking or non-blocking
- Tsi721 reassembles multiple S-RIO responses associated with the same PCIe MRd into one PCIe completion, and starts to send PCIe partial Cpl/CplD only after all responses have been received
- The last response decides the age of a fully reassembled completion and this age decides the order relationship between a completion and other P/NP requests and other completions.
- Partial Cpl/CplD from same MRd is sent in increasing memory order

2.2.1.3 Tsi721 PCIe to S-RIO Bridging Ordering Rules

Table 6: PCIe Request to S-RIO Priority Mapping

PCIe Type	Mapped S-RIO Type	Mapped S-RIO Priority	Mapped S-RIO CRF
MRd	NREAD	0	CRF in zone lookup table
MWr	NWRITE SWRITE NWRITE_R	2	CRF in zone lookup table
MWr	Maintenance write	2	CRF in zone lookup table
MRd	Maintenance read	0	CRF in zone lookup table
MWr	Doorbell	2	Assigned per transaction from PCIe header

Table 7: Bridged S-RIO Response Priority Mapping

S-RIO Request Type that Bridged to PCIe TLP	Bridged S-RIO Response Priority	Bridged S-RIO Response CRF
NREAD hitting an inbound window	+1 (relative to associated S-RIO request)	Same as associated S-RIO request

Table 8: Generated S-RIO Response Priority Mapping

S-RIO Request Type	Generated S-RIO Response Priority	Generated S-RIO Response CRF
NWRITE_R	3	Same as associated request
NREAD missing all inbound windows	3	Same as associated request
Maintenance write	3	1
Maintenance read	3	1
Doorbell	3	Same as associated request
Port-write	3	1

2.2.1.4 End-to-End PCIe Ordering Rules Supported by Tsi721 Bridging Data Path

Table 9 indicates the behavior of the Tsi721 if congestion at an endpoint in the system prevents packets in the outbound queues from making forward progress. Transactions can be reordered by packet priority as per the *RapidIO Specification (Rev. 2.1)*, but when back-pressure is applied by the network certain transactions may overtake others.

The table is organized in a maRtrix of the transactions that the Tsi721 can handle. The header row across the top of the table, and the header column along the left side of the table contain the same transaction types. The matrix indicates whether a transaction in a row of the table (listed in the left column) will become reordered ahead of a transaction in a column (listed in the header row of the table).

- TLP in a column is older than TLP in a row
- No means that row never passes column
- Yes means that row must pass column upon blocking
- Y/N means that row is un-ordered relative to a column and can pass column when blocking or non-blocking
- Partial Cpl/CplD from same MRd is sent in increasing memory order
- Doorbells have its own data path as Tsi721 manages doorbell queues. As a result, doorbells are un-ordered with respect to any other types of transactions.

Table 9: End to End PCIe Ordering Rules for Tsi721 Bridging Data Path (PCIe Compliant)

Row Pass Column	MWr (->NWRITE ->SWRITE ->NWRITE_R)	MRd (->NREAD)	CPL/CplD (from MRd-> NREAD)	MWr (->doorbell)	MWr (->Maintenance Write)	MRd (->Maintenance Read)	CPL/CplD (from MRd->Maintenance Read)
MWr (->NWRITE ->SWRITE ->NWRITE_R)	No	Yes	Yes	Y/N	Y/N	Y/N	Y/N
MRd (->NREAD)	No	No	No	Y/N	Y/N	Y/N	Y/N
CPL/CplD (from MRd-> NREAD, AtomicOP)	No	Yes	No	Y/N	Y/N	Y/N	Y/N
MWr (->doorbell)	Y/N	Y/N	Y/N	Y/N	Y/N	Y/N	Y/N
MWr (->maintenance write)	Y/N	Y/N	Y/N	Y/N	Y/N	Y/N	Y/N

Table 9: End to End PCIe Ordering Rules for Tsi721 Bridging Data Path (PCIe Compliant) *(Continued)*

Row Pass Column	MWr (->NWRITE ->SWRITE ->NWRITE_R)	MRd (->NREAD)	CPL/CplD (from MRd-> NREAD)	MWr (->doorbell)	MWr (->Maintenance Write)	MRd (->Maintenance Read)	CPL/CplD (from MRd->Maintenance Read)
MRd (->maintenance read)	Y/N	Y/N	Y/N	Y/N	Y/N	Y/N	Y/N
CPL/CplD (from MRd->maintenance read)	Y/N	Y/N	Y/N	Y/N	Y/N	Y/N	Y/N

2.2.1.5 Tsi721 Messaging Ordering Rules

The Tsi721 does not guarantee order among segments of a specific message. However, for a specific S-RIO prio/CRF of a specific Tx queue of a specific mailbox, the Tsi721 guarantees end-to-end in-order transmit between messages and in-order receive between messages.

S-RIO links can reorder messages of different prio/CRF even if they originated from the same Tx queue and mailbox.

The Tsi721 supports per message prio/CRF assignment. Responses of all message segments are promoted to S-RIO prio of 3 while retaining S-RIO CRF of associated message.

2.2.1.6 Block DMA Engine Ordering Rules

Table 10: Block DMA Engine PCIe to S-RIO Priority Assignment

PCIe Type	Mapped S-RIO Type	Mapped S-RIO Priority	Mapped S-RIO CRF
MRd	NREAD Maintenance read	Per descriptor; recommended value is 0	Per descriptor
MWr	NWRITE SWRITE NWRITE_R Maintenance write	Per descriptor; recommended value is 2	Per descriptor

Table 11: Block DMA Engine End-to-End PCIe Ordering Rules

Row Pass Column	MWr	MRd	Cpl/CplD
MWr	No	Yes	Yes
MRd	No	No	No
Cpl/CplD	No	Yes	No

2.2.2 Application Enforced Inter-Data Path Synchronization

2.2.2.1 Synchronization between a Doorbell and Messages

Applications may want to ensure that an RC/host is interrupted by a doorbell after all previous messages have reached their final destination.

1. Waits for the response of a specific message to return (this ensures that the message was transferred to the PCIe link):
   - By polling its descriptor status, or
   - By setting an interrupt for its descriptor and waiting for interrupt
2. Issues an NREAD if S-RIO endpoint, or MRd if PCIe endpoint, to the associated message Tx queue and waits for an NREAD response or MRd CplD to return. This ensures that the message has reached its final destination
3. Issues the doorbell

2.2.2.2 Synchronization between a Doorbell and Block DMA Engine Writes

Applications may want to ensure that an RC/host is interrupted by a doorbell after all previous DMA writes have reached their final destination.

1. Appends a descriptor that translates into NREAD, and waits for its response to return by doing one of the following (this ensures that previous DMA writes have reached their final destination):
   - Polling the descriptor status
   - Setting an interrupt for the descriptor and waiting for interrupt
2. Issues the doorbell

2.2.2.3 Synchronization between a Doorbell and Bridging NWRITE/SWRITE

Applications may want to ensure that an RC/host is interrupted by a doorbell after all previous bridging writes have reached their final destination.

1. Issues a bridging MRd that translates into NREAD, and waits for its response to return. This ensures that the previous NWRITE/SWRITE has reached its final destination.
2. Issues the doorbell

2.3 Loopbacks

The Tsi721 supports numerous loopback modes for test and fault isolation purposes. These modes are discussed in the following sections.

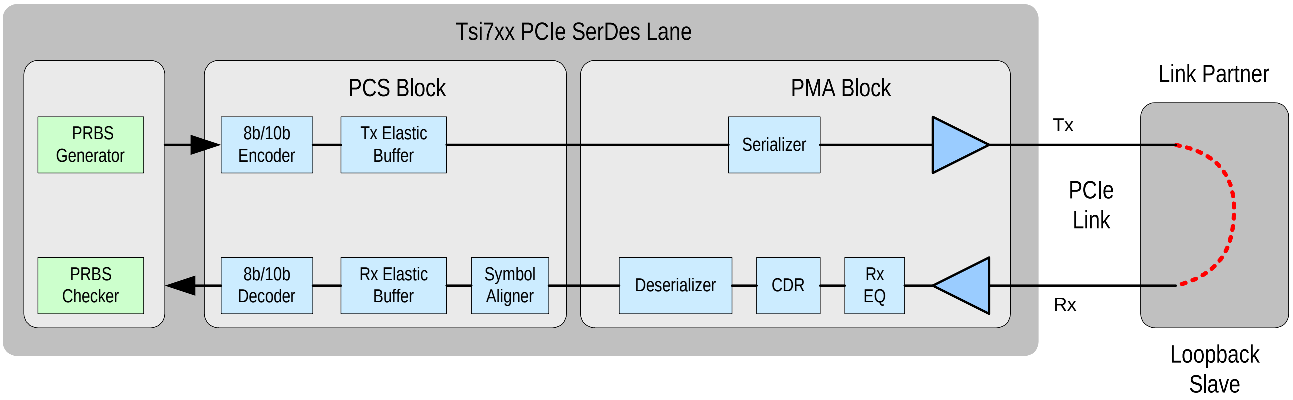
Table 12: Loopback Register Cross Reference

Register Mnemonic	Reference
STMCTL	SerDes Test Mode Control Register
STMTCTL	SerDes Test Mode Test Control Register
STMECNT0	SerDes Test Mode Lane 0 Error Count Register
STMECNT1	SerDes Test Mode Lane 1 Error Count Register
STMECNT2	SerDes Test Mode Lane 2 Error Count Register
STMECNT3	SerDes Test Mode Lane 3 Error Count Register
STMTSTS	SerDes Test Mode Test Status Register
ALLCS	Application Layer Loopback Control and Status Register
SERDESCFG	SerDes Configuration Register
RIO_PLM_SP_IMP_SPEC_CTL	RapidIO PLM Port Implementation Specific Control Register
SR_TX_CTL	PCIe SerDes Transmitter Control 2
SERDES_LANE{0..3}_LANEn_DIG_RX_OVRD_IN	SerDes Rx Control Register
SERDES_LANE{0..3}_LANEn_DIG_TX_OVRD_IN	SerDes Tx Control Register

2.3.1 PCIe Master Loopback

The following figure shows the location of where data is looped back in PCIe master loopback mode.

Figure 6: PCIe Master Loopback



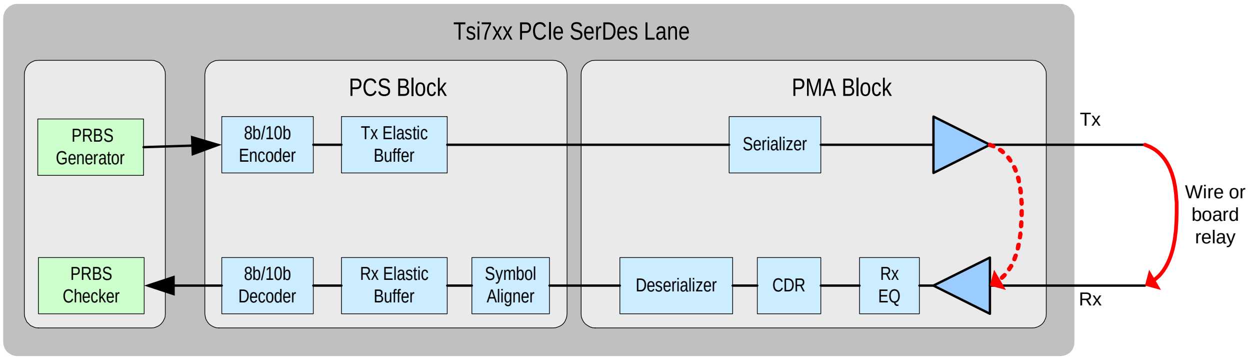
This loopback mode has the following characteristics:

- The PCIe PHY Link Training and Status State Machine (LTSSM) operates in the PCIe standard Loopback.Active state as a loopback master. This test can check the operation of the link “in-system” since the link partner will automatically operate as the loopback slave. This loopback mode requires link training with a link partner.
- It uses a PRBS pattern generator and a PRBS pattern checker located between the scrambling logic and the PCS logic.
- It exercises the transmit and receive paths in the SerDes PCS and PMA blocks (8b/10b encoder/decoder, receive elastic FIFO, serializer and de-serializer, and so on).
- There is a generator-checker pair for each SerDes lane. The generator and checker operate independently of each other.
- In order to ensure proper operation when entering this loopback, the TSEL field in the SerDes Test Mode Test Control Register must be set to PCIe Master Loopback Test before entering SerDes Test Mode through the SerDes Test Mode Control Register. Once the loopback is entered, it is not possible to change modes using the TSEL field unless SerDes Test Mode is exited by programming the CMD field in the SerDes Test Mode Control Register appropriately. A new test can then be executed by re-entering SerDes Test Mode and programming the TSEL field as desired.
- When this mode is entered, the PCIe PHY LTSSM gracefully transitions from its current state to the Loopback.Entry state. In this state, the link speed is set to the highest common value supported and advertised by both components of the link.
- Additionally, the LTSSM transmits TS1 training sequences with the loopback bit asserted until it receives a minimum of two identical training sequences on all lanes that detected a link partner. This indicates that the link partner has entered loopback slave mode and that its Clock Data Recovery (CDR) and symbol alignment logic have locked. It also indicates that the master’s receiver CDR and symbol alignment logic are locked. Once this is achieved, the master and slave proceed to the Loopback.Active state. If the master does not receive identical TS1 training sequences on all lanes that detected a link partner, the master proceeds to the Loopback.Exit state after a 48 ms timeout.
- Once in the Loopback.Active state, the PRBS pattern generator is enabled and starts transmitting a preamble consisting of 16 consecutive K28.6 (that is, COM) symbols, followed by PRBS data. While in this mode, the pattern generator transmits valid 8b/10b data and inserts/deletes SKP symbols into the received bit stream as per *PCI Express Specification (Rev. 2.1)* requirements.
- The PRBS pattern generator and checker use a PRBS32 polynomial; that is, $1+2^1+2^2+2^{22}+2^{32}$.
- The PRBS pattern checker first synchronizes to the loopback data and then starts checking for errors.
- Synchronization is achieved when the PRBS pattern checker verifies the reception of 16 consecutive bits of PRBS data (that is, the expected data matches the received data).
  - — The amount of time spent in the synchronization step can be programmed by writing to the TSYNCP field in the SerDes Test Mode Test Control Register.
  - — The SYNC field in the SerDes Test Mode Test Status Register indicates if the pattern checker was able to synchronize with the generator. Each lane has a corresponding SYNC bit.
- After synchronization is achieved, the checker looks for errors in the received pattern (that is, an error is a mismatch between the expected data value and the received data value).
- If an error is found, the COUNT field in the corresponding lane’s SerDes Test Mode Lane 0 Error Count Register is incremented. Additionally, the ERR_DET bit is set in the same register.
- The COUNT field is a 10-bit field that can count up to 1023 errors. When the number of errors detected exceeds 1023, the OVR bit is set in the corresponding lane’s SerDes Test Mode Lane 0 Error Count Register.
- It runs continuously until the TSEL field is set to “Idle” in the SerDes Test Mode Test Control Register (that is, no test is executed). The TR bit in the SerDes Test Mode Test Status Register is set to indicate that the test is executing. Once the test stops executing, the TR bit is cleared and the PHY LTSSM exits the Loopback.Active state as per *PCI Express Specification (Rev. 2.1)* requirements.
- The decision of how long a test should run is user defined, and may depend on the desired bit error rate measurement.

2.3.2 PCIe 8-bit PRBS Master Loopback

The following figure shows the location of where data is looped back in PCI 8-bit PRBS master loopback mode.

Figure 7: PCIe 8-bit PRBS Master Loopback



▪▪▪ Optional loopbacks that may be used with this test .

This loopback mode has the following characteristics:

- It uses the PRBS pattern generator and checker as in PCIe master loopback mode.
- Unlike the PCIe master loopback, in the PCIe PRBS master loopback the LTSSM operates in the SerDes_Test state. Thus, the presence of a link partner is not required, which implies that this test can be used in a test lab or production environment.
- There is a generator-checker pair for each SerDes lane.
- Data can be looped back externally or internally. In order to loop back the data internally, the ILPBSEL field in the SerDes Configuration Register must be set to 0x1 before starting the test.
- When this loopback mode is entered[1], the LTSSM gracefully transitions from its current state into the SerDes_Test state.
- Once the LTSSM enters the SerDes_Test state, the PRBS pattern generator is enabled and starts transmitting a data preamble composed of 2048 repeating K28.5 (that is, COM) characters. The receiver uses this data preamble to achieve CDR lock and symbol alignment.
  - — The amount of time spent in the preamble step can be programmed by writing to the TSYNCP field in the SerDes Test Mode Test Control Register
- Following the preamble, the pattern generator transmits valid 8b/10b PRBS data and inserts/deletes Skip (SKP) symbols into the received bit stream as per *PCI Express Specification (Rev. 2.1)* requirements.
- The PRBS pattern generator and checker use a PRBS32 polynomial (that is, $1+2^1+2^2+2^{22}+2^{32}$).
- The pattern checker first synchronizes to the loopback PRBS data and then starts checking for errors.
- Synchronization is achieved when the SerDes CDR circuit locks on to the incoming data, symbol alignment occurs, and the pattern checker verifies the reception of 16 consecutive bits of PRBS data (that is, the expected data matches the received data). Preamble data is ignored by the PRBS checker.
  - — The SYNC field in the SerDes Test Mode Test Status Register indicates if the pattern checker was able to synchronize with the generator. Each lane has a corresponding SYNC bit.

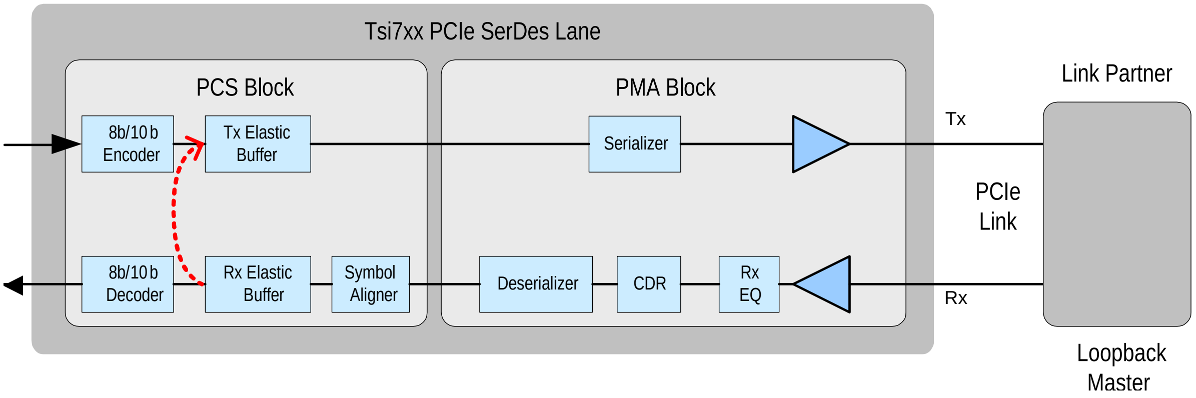
[1] In order to ensure proper operation when entering this test mode, the TSEL field in the SerDes Test Mode Test Control Register must be set to 8-bit PRBS Master Loopback Test after entering SerDes Test Mode through the SerDes Test Mode Control Register.

- After synchronization is achieved, the checker looks for errors in the received pattern (that is, an error is a mismatch between the expected data value and the received data value).
- If an error is found, the COUNT field in the corresponding lane's SerDes Test Mode Lane 0 Error Count Register is incremented. Additionally, the ERR_DET bit is set in the same register.
- The COUNT field is a 10-bit field that can count up to 1023 errors. When the number of errors detected exceeds 1023, the OVR bit is set in the corresponding lane's SerDes Test Mode Lane 0 Error Count Register.
- It runs continuously until the TSEL field is set to "Idle" in the SerDes Test Mode Test Control Register (that is, no test is executed). The TR bit in the SerDes Test Mode Test Status Register is set to indicate that the test is executing. Once the test stops executing, the TR bit is cleared in the SerDes Test Mode Test Status Register.
- The decision of how long a test should run is user defined, and may depend on the desired bit error rate measurement.

2.3.3 PCIe Slave Loopback

The following figure shows the location of where data is looped back in slave loopback mode.

Figure 8: PCIe Slave Loopback



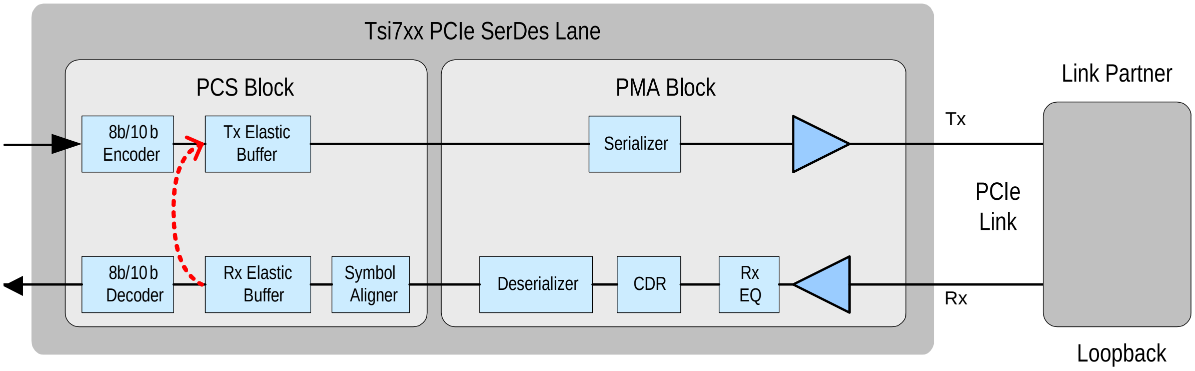
This loopback mode has the following characteristics:

- It requires link training with a link partner.
- It is entered automatically when directed by a link partner, as per the *PCI Express Specification (Rev. 2.1)* (for example, when the PCIe MAC receives training sequences with the Loopback bit asserted).
- No failure status is reported in the PCIe MAC registers. The loopback master is required to track any errors on the link.
- The PCIe PHY's LTSSM operates in the PCIe standard Loopback.Active state as a loopback slave. Thus, this test can check the operation of the link "in-system", as the link partner will automatically operate as the loopback master.
- The Tsi721 supports this mode only when operating in PCIe Common Clock Mode with its PCIe link partner.

2.3.4 PCIe 10-bit PCS Slave Loopback

The following figure shows the location of where data is looped back in 10-bit PCS slave loopback mode.

Figure 9: PCIe 10-bit PCS Slave Loopback



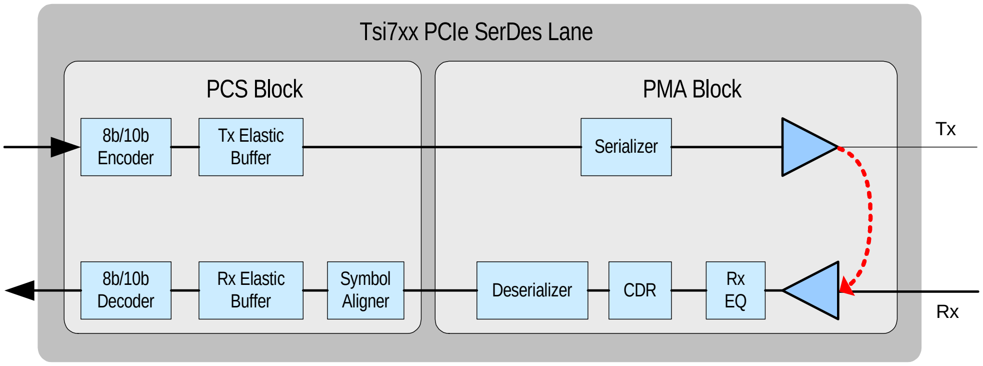
This loopback mode has the following characteristics:

- It loops back raw 10-bit codes without modification from receiver to transmitter.
- Since received SKP codes are not removed, PCIe common clock mode must be used.
- To enter this loopback:
  a. Place the PCIE MAC in SerDes Test Mode by setting SerDes Test Mode Control Register.CMD to 1.
  b. Disable the symbol aligner inside the PCIe SerDes by setting the following bits in the SerDes Rx Control Register as indicated (all other bits of the register should be set to 0):
     – EN to 1
     – ALIGN_EN to 0
     – TERM_EN, DATA_EN and PLL_EN to 1
     – INVERT as desired (set to 1 to invert the receive serial data)
  c. Set SerDes Test Mode Test Control Register.TSEL to 0b111 to indicate 10-bit PCS Slave Loopback Mode.
- After completing this loopback test, it is not possible to execute any other SerDes test unless the PCIE MAC is placed in normal operating mode and back in SerDes Test Mode (that is, by programming the CMD field in the SerDes Test Mode Control Register), or a fundamental reset is applied to the PCIE-EP.

2.3.5 PCIe PMA Loopback

The following figure shows the location of where data is looped back in Tx/Rx PMA loopback mode.

Figure 10: PCIe PMA Loopbacks



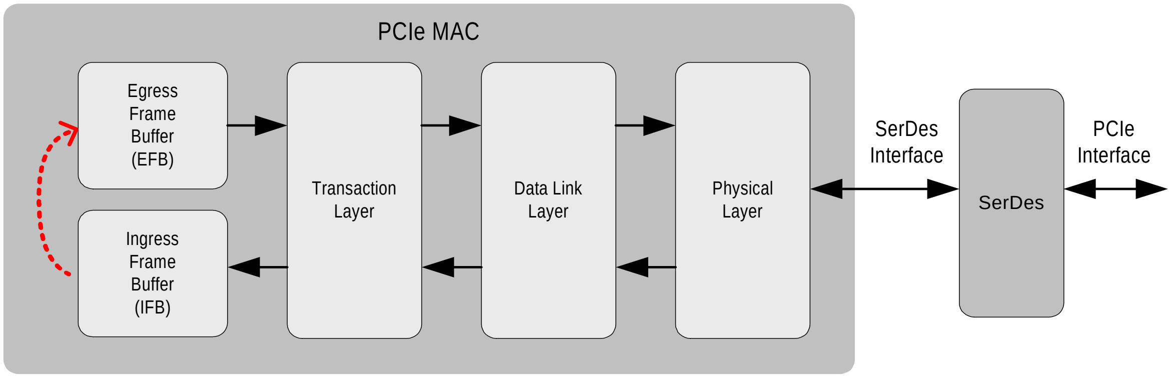
This loopback mode has the following characteristics:

- SerDes transmitter output is looped back to SerDes receiver input.
- To enter this loopback:
  a. Set ILPBSEL to 0x1 in SerDes Configuration Register
  b. Set the CLINKDIS and SCLINKEN bits to 1 in the PHY Link Configuration 0 Register.
- After enabling a loopback path during normal operation, the PCIe link must be retrained by setting FLRET to 1 in the PHY Link State 0 Register.

2.3.6 PCIe Application Layer Loopback

The Tsi721 supports an application layer loopback mode. PCIe MAC ingress frame buffer (IFB) output is looped back to PCIe MAC egress frame buffer (EFB) input. Figure 11 shows the location of where data is looped back in application layer loopback mode.

Figure 11: PCIe Application Layer Loopback



This loopback mode has the following characteristics:

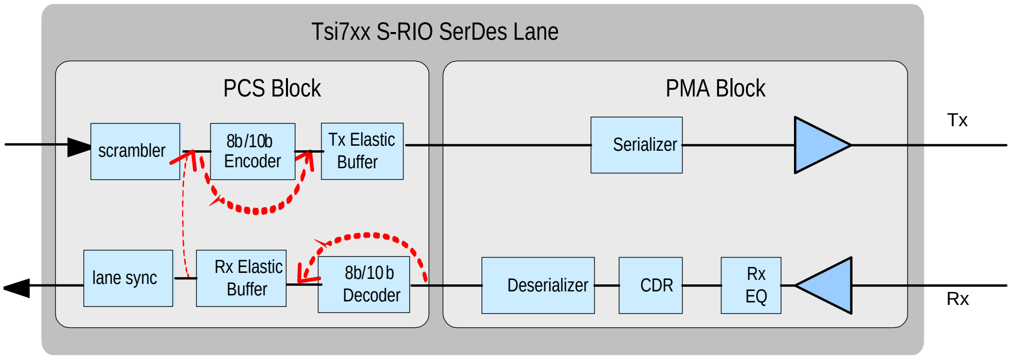
- It is enabled by setting ALLME bit to 1 in the Application Layer Loopback Control and Status Register. Since the PCIe MAC can take a few cycles to enter this mode, the ALLS bit in the same register is set by hardware to indicate that the loopback path has been established.

- The following requirements are needed to ensure correct entry and exit from Application Layer Loopback Mode. Violating these requirements produces undefined operation.
  - Software must ensure that Tsi721's internal buffers have been emptied for 1 second before the mode is enabled (for an example of how to achieve this, see Procedure for Resets Other Than Fundamental Reset).
  - The ALLS bit in the Application Layer Loopback Control and Status Register must acknowledge entry to the desired mode before TLP traffic is resumed.

2.3.7 S-RIO Line Loopback

The following figure shows the location of where data is looped back in S-RIO loopback mode.

Figure 12: S-RIO Line Loopback



This loopback mode has the following characteristics:

- The Rx elastic buffer output (raw data before 8b/10b decoding) on an S-RIO SerDes receive lane is looped back to Tx elastic buffer of the SerDes transmitter.
- The Tsi721 supports this mode only when operating in S-RIO common clock configuration with its S-RIO link partner.

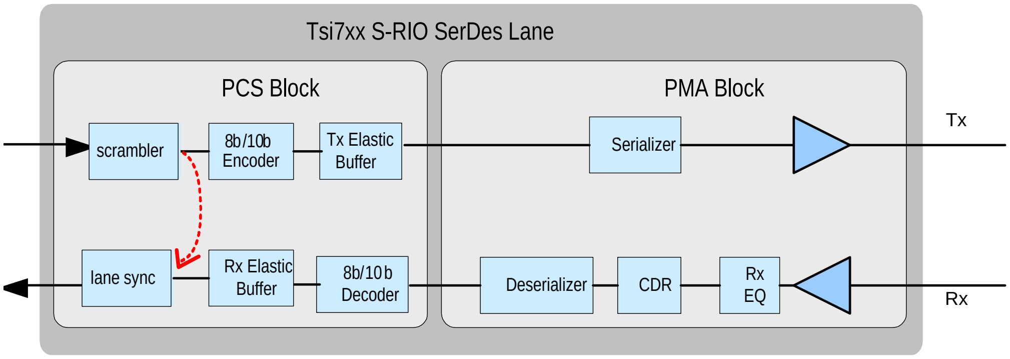
To enable loopback of each lane of a multi-lane port:

- Disable the symbol aligner inside the S-RIO SerDes by setting the following bits in the SerDes Rx Control Register as indicated (all other bits of the register should be set to 0):
  - EN to 1
  - ALIGN_EN to 0
  - TERM_EN, DATA_EN and PLL_EN to 1
  - INVERT as desired (set to 1 to invert the receive serial data)
- Enable line loopback by setting the LLB_EN bit to 1 in the RapidIO PLM Port Implementation Specific Control Register.

2.3.8 S-RIO Digital Equipment Loopbacks

The following figure shows the location where data is looped back in S-RIO digital equipment loopback mode.

Figure 13: S-RIO Digital Equipment Loopback



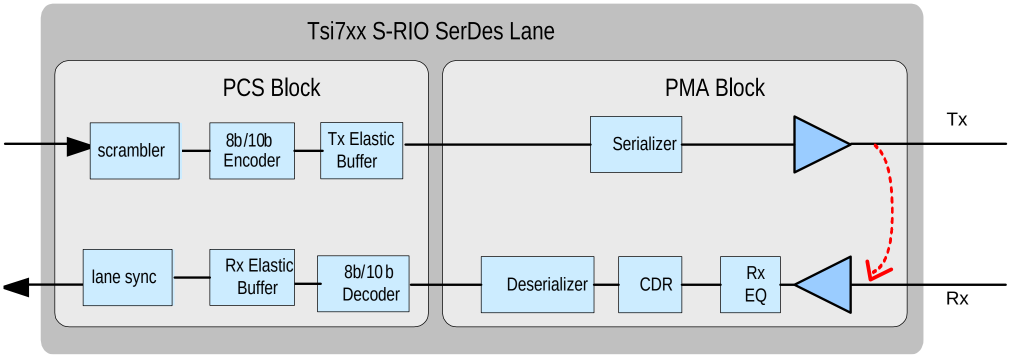
This loopback mode has the following characteristics:

- Scrambler output before 8b/10b encoding on an S-RIO SerDes transmit lane is looped back to lane synchronizer of the SerDes receiver.
- To enter this loopback, set the DLB_EN bit to 1 in the RapidIO PLM Port Implementation Specific Control Register.

2.3.9 S-RIO PMA Loopback

The following figure shows the location where data is looped back in S-RIO PMA loopback mode.

Figure 14: S-RIO PMA Loopback



This loopback mode has the following characteristics:

- S-RIO SerDes transmitter output is looped back to SerDes receiver input.
- To enter this loopback, set the LB_EN bit to 1 in the S-RIO SerDes Transmitter Control of Lanes {0..3}. If S-RIO PRBS is enabled, then set the LOOPBK_EN bit to 1 in the S-RIO SerDes Tx Control Register (for more information, see Bit Error Rate Testing (BERT)).

2.4 Performance

This section discusses the throughput and latency characteristics of the Tsi721.

2.4.1 Throughput Measurements

The Tsi721's performance is influenced by the payload size of the packets being bridged. The following payload characteristics affect performance:

- Payloads in complete increments of 64 bytes will achieve the highest throughput.
- Packets with payloads of 68 to 84 bytes will experience lower throughput due to the overhead associated with the fractions of 64 bytes that are being transferred.
- PCIe packet with a header of 3 dword (as opposed to 4 dword) also degrades the throughput.

Table 13 contains throughout measurements for the Tsi721. The measurements data is presented as a percentage of the maximum possible packet throughput for a 4x port operating at a lane speed of 5 Gbaud.

Table 13: Throughput Measurements

Transaction Type	PCIe to RapidIO Throughput	RapidIO to PCIe Throughput	Notes
Mapping Engine (Bridging)			
MWr, NWRITE/SWRITE	100%	100%	Network delays affecting the response are not included.
MRd, NREAD	100%	100%	-
Block DMA Engine (BDMA)			
MWr, NWRITE/SWRITE	100%		-
MRd, NREAD	90%		>= 1-KB payload. Network delays affecting the response are not included.
Messaging Engine (BDMA)			
Messaging, Single Channel	75%	100%	One message outstanding. Network delays affecting the response are not included.
Messaging, Multiple Channel	100%	100%	-

2.4.2 Latency Measurements

Table 14 contains latency measurements for the Tsi721. Actual latency will vary based on the characteristics and loading of the RapidIO fabric and PCIe root complex that are connected to the Tsi721.

Table 14: Latency Measurements

Transaction Type	Payload Size (Bytes)	Latency (ns)	Condition
DMA Write	256	722	From PCIe Data CplD EOP to S-RIO SOP
DMA Read	256	506	From S-RIO Response EOP to PCIe SOP
DMA Write	8	452	From PCIe Data CplD EOP to S-RIO SOP
DMA Read	8	368	From S-RIO Response EOP to PCIe SOP
Inbound SWRITE/NWRITE	256	416	From S-RIO SWRITE/NWRITE EOP to PCIe SOP
Inbound NREAD	256	518	From PCIe CplD EOP to S-RIO SOP
Inbound SWRITE/NWRITE	8	288	From S-RIO SWRITE/NWRITE EOP to PCIe SOP
Inbound NREAD	8	327	From PCIe CplD EOP to S-RIO SOP
Doorbell Inbound	-	312	From S-RIO EOP to PCIe SOP
Doorbell Outbound	-	322	From PCIe EOP to S-RIO SOP

2.5 Endian Conversion

2.5.1 Address Invariant Endian Conversion for Payload

Tsi721 endian conversion for payload bytes uses address invariant mode: PCIe payload byte at address Y is converted into S-RIO payload byte at the same address Y, and vice versa. If required, software performs further endian conversion according to the application data structure.

2.5.2 Register Endian Conversion

The S-RIO MAC’s registers are in big-endian format. All other Tsi721 registers are in little-endian format.

When reading a little-endian format register to S-RIO, bit 31 should map to S-RIO payload bit 0 of byte 0 (bits 31:0 to S-RIO payload bits 0:31).

When reading a big-endian format register to PCIe, bit 0 should map to PCIe payload bit 7 of byte 3 (bits 0:31 to PCIe payload bits 31:0).

When reading a big-endian format register to I2C/JTAG, bit 0 should map to equivalent I2C/JTAG payload bit 7 of byte 3 (bits 0:31 to equivalent I2C/JTAG payload bits 31:0).

2.6 Data Protection

- All Tsi721 internal memories are ECC protected, single-bit error correction, double-bit error detection
- Software must reset the Tsi721 when an Tsi721 uncorrectable ECC error occurs
  - — Uncorrectable ECC errors within the PCIe MAC are reported through PCIe IER. The Tsi721 automatically recovers from these errors.
  - — Uncorrectable ECC errors within the BDMA/SMSG/PC2SR/SR2PC/S-RIO MAC are flagged through INTx/MSI/MSIx, GPIO[0] signal, and SMBus alert response. Software must reset the Tsi721 to recover from these errors.

3. Signals

Topics discussed include the following:

3.1 Overview

The following conventions are used in this chapter:

- Signals with the suffix “P” are the positive half of a differential pair.
- Signals with the suffix “N” are the negative half of a differential pair.
- Signals with the suffix “n” are active low.

Signals are classified according to the types defined in the following table.

Table 15: Signal Types

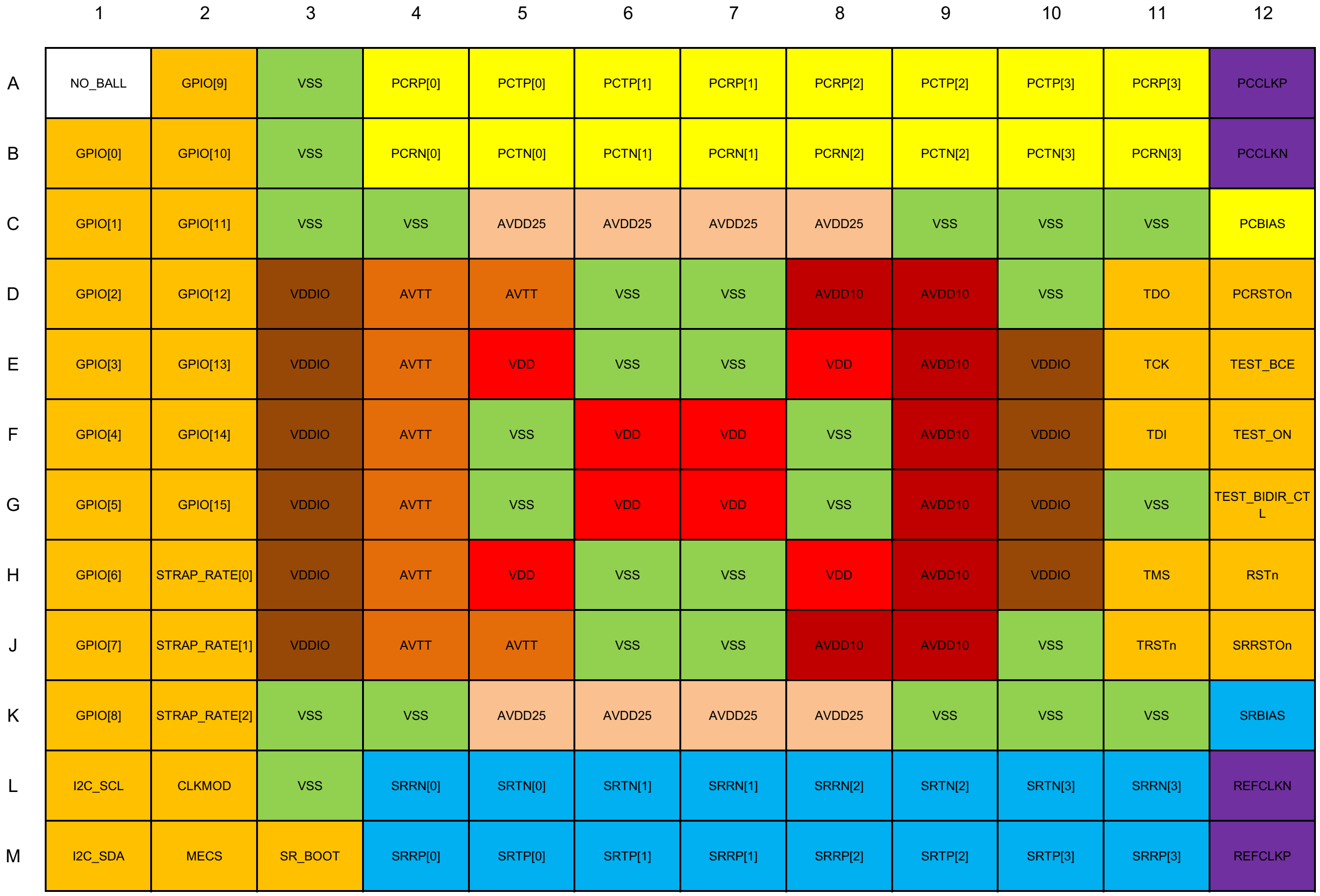
Pin Type	Definition
I	3.3/2.5V LVTTL Input
O	3.3/2.5V LVTTL Output
IO	3.3/2.5V LVTTL Bidirectional
IO-OD	3.3/2.5V LVTTL Bidirectional Open Drain
OD	3.3/2.5V LVTTL Open Drain
I-PU	3.3/2.5V LVTTL Input with Pull-up
I-PD	3.3/2.5V LVTTL Input with Pull-down

Table 15: Signal Types *(Continued)*

Pin Type	Definition
IO-PD	3.3/2.5V LVTTL Bidirectional with Pull-down
IO-PU	3.3/2.5V LVTTL Bidirectional with Pull-up
PCIE_O	Differential CML PCIe output
PCIE_I	Differential CML PCIe input
SRIO_O	Differential CML S-RIO output
SRIO_I	Differential CML S-RIO input
DIFF_I	Differential CML input
PWR	Power
GND	Ground

3.2 Ballmap

Figure 15: Ballmap



	1	2	3	4	5	6	7	8	9	10	11	12
A	NO_BALL	GPIO[9]	VSS	PCRP[0]	PCTP[0]	PCTP[1]	PCRP[1]	PCRP[2]	PCTP[2]	PCTP[3]	PCRP[3]	PCCLKP
B	GPIO[0]	GPIO[10]	VSS	PCRN[0]	PCTN[0]	PCTN[1]	PCRN[1]	PCRN[2]	PCTN[2]	PCTN[3]	PCRN[3]	PCCLKN
C	GPIO[1]	GPIO[11]	VSS	VSS	AVDD25	AVDD25	AVDD25	AVDD25	VSS	VSS	VSS	PCBIAS
D	GPIO[2]	GPIO[12]	VDDIO	AVTT	AVTT	VSS	VSS	AVDD10	AVDD10	VSS	TDO	PCRSTOn
E	GPIO[3]	GPIO[13]	VDDIO	AVTT	VDD	VSS	VSS	VDD	AVDD10	VDDIO	TCK	TEST_BCE
F	GPIO[4]	GPIO[14]	VDDIO	AVTT	VSS	VDD	VDD	VSS	AVDD10	VDDIO	TDI	TEST_ON
G	GPIO[5]	GPIO[15]	VDDIO	AVTT	VSS	VDD	VDD	VSS	AVDD10	VDDIO	VSS	TEST_BIDIR_CTL
H	GPIO[6]	STRAP_RATE[0]	VDDIO	AVTT	VDD	VSS	VSS	VDD	AVDD10	VDDIO	TMS	RSTn
J	GPIO[7]	STRAP_RATE[1]	VDDIO	AVTT	AVTT	VSS	VSS	AVDD10	AVDD10	VSS	TRSTn	SRRSTOn
K	GPIO[8]	STRAP_RATE[2]	VSS	VSS	AVDD25	AVDD25	AVDD25	AVDD25	VSS	VSS	VSS	SRBIAS
L	I2C_SCL	CLKMOD	VSS	SRRN[0]	SRTN[0]	SRTN[1]	SRRN[1]	SRRN[2]	SRTN[2]	SRTN[3]	SRRN[3]	REFCLKN
M	I2C_SDA	MECS	SR_BOOT	SRRP[0]	SRTP[0]	SRTP[1]	SRRP[1]	SRRP[2]	SRTP[2]	SRTP[3]	SRRP[3]	REFCLKP

This document is confidential and is subject to an NDA.

3.3 Pinlist

For a list-based version of Tsi721's pin to signal mapping, see the *Tsi721 Ballmap and Pinlist*.

3.4 PCIe Signals

Table 16: PCIe Signals

Name	Pin Type	Description
PCTP[3:0] PCTN[3:0]	PCIE_O	Differential transmit data for the PCIe port.
PCRP[3:0] PCRN[3:0]	PCIE_I	Differential receive data for the PCIe port.
PCCLKP PCCLKN	DIFF_I	PCIe reference clock input. When in PCIe common clock mode (CLKMOD pin is high, see the “Clocking” chapter in the Tsi721 User Manual), PCCLKP/N requires a clock frequency of 100 MHz. When in PCIe non-common clock mode (CLKMOD pin is low), PCCLKP/N requires a clock frequency as selected by CLKSEL[1:0], and must have the same clock frequency as REFCLKP/N.
PCRSTOn	IO	It is an output for normal operation and an input during scan test mode. As an asynchronous active-low reset output, this pin is low when the following occurs: • The PCIe port detects hot reset • The PCIe port is DL_DOWN

3.5 S-RIO Signals

Table 17: S-RIO Signals

Name	Pin Type	Description[a]
SRTP[3:0] SRTN[3:0]	SRIO_O	Differential transmit data for the S-RIO port.
SRRP[3:0] SRRN[3:0]	SRIO_I	Differential receive data for the S-RIO port.
SRRSTOn	IO	It is an output for normal operation and an input during scan test mode. As an asynchronous active-low reset output, this pin is low when four consecutive S-RIO reset symbols are received, and SELF_RST is set to 1 in the RapidIO PLM Port Implementation Specific Control Register
MECS	IO-PD	Asynchronous S-RIO Multicast Event Control Symbol (MECS). Its direction is controlled by the MECS_O bit in the Device Control Register. As an *input*, a rising or falling edge triggers an S-RIO MECS to be sent on the S-RIO link. Use the RIO_PLM_SP0_MECS_FWD.SUBSCRIPTION/MULT_CS and RIO_EM_MECS_TRIG_EN.CMD_EN to select the CMD field that should be set with the MECS. Multiple MECSs with different CMD fields can be generated by setting these fields appropriately. As an *output*, this signal is toggled when an S-RIO MECS is received. Only a single MECS CMD value should be selected to toggle the MECS input. Set the RIO_EM_MECS_CAP_EN.CMD_EN to select the CMD value to be propagated to the MECS pin. Note: Only 1 bit should be enabled in CMD_EN.

[a] For information on S-RIO signals that are used for power-up purposes only, see Power-up Signals.

3.6 General Signals

Table 18: General Signals

Name	Pin Type	Description
RSTn	I-PU	Fundamental reset (device reset).Assertion of this signal resets all logic inside the Tsi721.
REFCLKP REFCLKN	DIFF_I	S-RIO reference clock input. REFCLK requires a clock frequency as selected by CLKSEL[1:0].

3.7 I2C Signals

The I2C Interface is used for the following:

- As a master, downloading configuration from EEPROM
- As a master, allowing the PCIe root complex or the S-RIO host to configure other I2C expansion devices
- As a slave, exposing internal register space to an I2C master (Note: To be used for lab debug or another master-driven initialization).

Table 19: I^2C Signals

Name	Pin Type	Description[a]
I2C_SCL	IO-OD	Serial clock for the I2C Interface with a maximum frequency of 100 kHz.
I2C_SDA	IO-OD	Serial data for the I2C Interface.

[a] For information on I2C signals that are used for power-up purposes only, see Power-up Signals.

3.8 JTAG and Test Interface Signals

Table 20: JTAG Interface Signals

Name	Pin Type	Description
TCK	I-PD	IEEE 1149.1/1149.6 test access port. Clock input.
TDI	I-PU	IEEE 1149.1/1149.6 test access port. Serial data input
TDO	O	IEEE 1149.1/1149.6 test access port. Serial data output
TMS	I-PU	IEEE 1149.1/1149.6 test access port. Test mode select
TRSTn	I-PU	IEEE 1149.1/1149.6 test access port. Reset input. This input must be asserted during the assertion of RSTn. Thereafter, it can be left in either state.
TEST_ON	I-PD	Test mode pin. Tie low or NC for normal operation.
TEST_BCE	I-PU	Boundary scan compatibility enabled pin. This input aids 1149.6 testing. It must be tied to VDDIO (or NC as there is internal pull up in pad) during normal operation of the device. 0 = JTAG chain includes SerDes registers. SerDes registers are accessible to external JTAG pins. Used during ATE and lab debug of SerDes registers through an external JTAG Controller. 1 = JTAG chain does not include SerDes registers. SerDes register are accessible through the internal register bus for BAR 0 access.
TEST_BIDIR_CTL	I-PU	Test mode pin. Tie high or NC for normal operation.

3.9 GPIO Signals

Table 21: GPIO Signals

Name	Pin Type	Description
GPIO[15:0]	IO	Asynchronous general purpose I/O. • Each GPIO pin can be configured as a general purpose I/O pin. • Each pin can be configured as either an input or an output • When configured as an output, GPIO[0] is asserted high when BDMA/SMSG/PC2SR/SR2PC has an uncorrectable ECC error or S-RIO MAC has a non-data memory uncorrectable ECC error • When configured as an output, GPIO[1] is asserted high when Tsi721 PCIe port is not in the data link active state • When configured as an output, GPIO[2] is asserted high when Tsi721 has an active interrupt (for more information, see Figure 18 and Figure 19) • When configured as an output, GPIO[15:3] can be programmed through software GPIO[12:0] are used as power-up pins as displayed in Table 22. These signals must remain stable for 4000 REFCLKP/REFCLKN cycles after RSTn is de-asserted. They are ignored after reset.

Table 22: GPIO Mapping to Power-up Signals

GPIO Pin Name (Primary Function)	Power-up Pin Name[a] (Secondary Function)
GPIO[3:0]	I2C_SA[3:0]
GPIO[4]	I2C_DISABLE
GPIO[5]	I2C_SEL
GPIO[6]	I2C_MA
GPIO[7]	SP_SWAP_RX
GPIO[8]	SP_SWAP_TX
GPIO[9]	SP_HOST
GPIO[10]	SP_DEVID
GPIO[12:11]	CLKSEL[1:0]

[a] For more information about these signals, see Power-up Signals.

3.10 Power-up Signals

Table 23: Power-Up Signals

Name	Pin Type	Description
CLKMOD	I-PU	Clock mode. When high, Tsi721 uses “PCIe common clocked mode.” When low, it uses “PCIe non-common clocked mode.” It is a static signal.
CLKSEL[1:0]	IO	REFCLKP/REFCLKN clock frequency select; PCCLKP/PCCLKN clock frequency select when in PCIe non-common clock mode. • 0b11 = 125 MHz • 0b10 = 100 MHz • 0b01 = 156.25 MHz • Others = Reserved When a 100-MHz clock is used, S-RIO SerDes rates of 1.25/2.5/5 Gbaud are supported. When a 125/156.25-MHz clock is used, S-RIO SerDes rates of 1.25/2.5/3.125/5 Gbaud are supported. When either a 100/125/156.25-MHz clock is used, PCIe SerDes rates of 2.5/5 Gbaud are supported. These power-up signals are multiplexed with GPIO[12:11]. It is a static signal.
I2C_DISABLE	IO	Disable I^2C register loading after reset. When asserted, Tsi721 does not attempt to load register values from an EEPROM over the I^2C bus. 0 = Enable boot load from EEPROM 1 = Disable boot load from EEPROM This power-up signal is multiplexed with GPIO[4]. It is a static signal.
I2C_MA	IO	I^2C multi-byte address mode. If I2C_DISABLE == 0 (that is, download registers from EEPROM) then: 0 = Tsi721 uses 1-byte addressing for EEPROM 1 = Tsi721 uses 2-byte addressing for EEPROM Else I2C_DISABLE == 1 (do not download from EEPROM) • 0 = Tsi721 is boot loaded by the PCIe root complex after reset • 1 = Tsi721 is boot loaded by an external I2C master after reset This power-up signal is multiplexed with GPIO[6]. It is a static signal.
I2C_SA[3:0]	IO	I2C slave address. The values on these pins represent the values for the 7-bit address of the Tsi721 when acting as an I^2C slave. These signals, in combination with the I2C_SEL signal, determine the address of the EEPROM to boot from (see I2C_SEL pin description). The values on these pins can be overridden after a reset by writing to the I2C Slave Configuration Register. These power-up signals are multiplexed with GPIO[3:0]. It is a static signal.

Table 23: Power-Up Signals *(Continued)*

Name	Pin Type	Description

I2C_SEL	IO	I^{2}C pin select. Combined with the I2C_SA[1,0] pins, Tsi721 will determine the lower 2 bits of the 7-bit address of the EEPROM address it boots from.
		When asserted, the I2C_SA[1:0] pins represent the two LSBs of the 7-bit EEPROM slave address when Tsi721 acts as a I^{2}C master downloading from an EEPROM. The EEPROM slave address is as follows:
		A6 = 1
		A5 = 0
		A4 = 1
		A3 = 0
		A2 = 0
		A1 = I2C_SA[1]
		A0 = I2C_SA[0]
		When de-asserted, the I2C_SA[1:0] pins are ignored and the lower two bits of the EEPROM address default to 00. The values of the EEPROM address can be overridden by software after initialization.
		This power-up signal is multiplexed with GPIO[5]. It is a static signal.

SP_DEVID	IO	S-RIO base deviceID control When the SP_HOST pin is high, it configures the reset value of the RapidIO Base deviceID CSR: the LSB of the CSR's BASE_ID and LAR_BASE_ID fields are set to SP_DEVID, while other bits of these fields are set to 0. When the SP_HOST pin is low and SP_DEVID is high, it configures the reset value of the RapidIO Base deviceID CSR: the CSR's BASE_ID and LAR_BASE_ID fields are set to all ones. When the SP_HOST pin is low and SP_DEVID is low, it configures the reset value of the RapidIO Base deviceID CSR: the CSR's BASE_ID field is set to 0xFE and the CSR's LAR_BASE_ID field are set to 0x00FE. This signal is multiplexed with GPIO[10]. It is a static signal.
SP_HOST	IO	S-RIO host / slave control. This signal sets the reset value of the HOST bit of the RapidIO Port General Control CSR. 0 = Tsi721 is an S-RIO slave. 1 = Tsi721 is an S-RIO host. This signal is multiplexed with GPIO[9]. It is a static signal.
SP_SWAP_RX	IO	S-RIO receive lane swap. This signal sets the reset value of the SWAP_RX[1:0] bits of RapidIO PLM Port Implementation Specific Control Register. 0 = Disable S-RIO port receive lane swap; that is, set the SWAP_RX[1:0] register bits to 0b00. 1 = Enable S-RIO port receive 4x lane swap; that is, set the SWAP_RX[1:0] register bits to 0b10. This signal is multiplexed with GPIO[7].

Table 23: Power-Up Signals *(Continued)*

Name	Pin Type	Description
SP_SWAP_TX	IO	S-RIO transmit lane swap. This signal sets the reset value of the SWAP_TX bit of RapidIO PLM Port Implementation Specific Control Register. 0 = Disable S-RIO port transmit lane swap. 1 = Enable S-RIO port transmit lane swap. This signal is multiplexed with GPIO[8]. It is a static signal.
SR_BOOT	I-PD	Boot from S-RIO. It can be asserted high only when I2C_DISABLE is also high. 1 = The Tsi721 S-RIO link can start training immediately after a fundamental reset and Tsi721 automatically sets the SRBOOT_CMPL bit of Device Control Register. 0 = The Tsi721 S-RIO link can start training only after software sets the SRBOOT_CMPL bit. It is a static signal.
STRAP_RATE[2:0]	I-PU	S-RIO link rate. These signals control the reset value of the BAUD_SEL field of the RapidIO Port Control 2 CSR . Note that the BAUD_SEL encoding is different than that of STRAP_RATE. • 0b111 = 5 Gbaud • 0b110 = 2.5 Gbaud • 0b101 = 1.25 Gbaud • 0b010 = 3.125 Gbaud • Others: Reserved It is a static signal.

3.11 Power Supply Signals

Table 24: Power Supply Signals

Name	Pin Type	Description
VDD	PWR	1.0V core power
VDDIO	PWR	3.3/2.5V power for LVTTL IO
AVDD10	PWR	1.0V PCIe and S-RIO SerDes analog power supply
AVDD25	PWR	2.5V PCIe and S-RIO SerDes analog power supply
AVTT	PWR	1.5V PCIe and S-RIO SerDes transmitter analog voltage
VSS	GND	Shared digital and analog ground
PCBIAS	IO	Reference for the corresponding PCIe SerDes bias currents and PLL calibration circuitry. A 200 Ohm 1% 100ppm/C precision resistor should be connected from this pin to ground and isolated from any source of noise injection.
SRBIAS	IO	Reference for the corresponding S-RIO SerDes bias currents and PLL calibration circuitry. A 200 Ohm 1% 100ppm/C precision resistor should be connected from this pin to ground and isolated from any source of noise injection.

PCIe Interface

4. PCIe Interface

Topics discussed include the following:

- Features
- Physical Layer
- Data Link Layer
- Base Address Registers
- TLP Receive Processing
- Transmit TLP Processing
- Buffers
- Device Power Management

4.1 Features

- Complete PCI Express (PCIe) Gen2 x4 Endpoint
  - Physical, data link, and transaction layers
  - Configuration space registers
  - Physical Coding Sublayer (PCS)
  - SerDes
  - All required RAMs
- Standards and compatibility
  - *PCI Express Specification (Rev. 2.1)* compliant
  - Implements the following optional PCIe features
    - Advanced Error Reporting
    - End-to-End CRC (ECRC)
    - Internal Error Reporting for PCIe MAC internal errors
    - Intx, MSI, and MSIx
    - Device serial number capability
- Supports four BARs (six BAR registers, BAR 0 to BAR 5):
  - A prefetchable BAR with 32- or 64-bit addressing for PCIe to S-RIO bridging
  - A non-prefetchable BAR with 32- or 64-bit addressing for PCIe to S-RIO bridging
  - A non-prefetchable BAR with 32-bit addressing for PCIe MWr to S-RIO doorbell bridging
  - A non-prefetchable BAR with 32-bit addressing for Tsi721 internal register access
- Supports TLP generation with one virtual channel (VC0) and one traffic class (TC0)

- Port configurability
  - Automatic link width negotiation (for example, x4/x2/x1)
  - Automatic lane reversal
  - Polarity inversion
  - Dynamic link width reconfiguration (Tsi721 never initiates this by itself)
  - Link retraining
- Supports 128- and 256-byte maximum payload sizes
- SerDes configuration
  - De-emphasis
  - Drive strength
- SerDes test modes
- Reliability Availability Serviceability (RAS)
  - ECC on all internal RAMs
  - Ability to inject errors

4.1.1 Unsupported Optional PCIe Features

The Tsi721 does not support the following optional features of the *PCI Express Specification (Rev. 2.1)*:

- TLP processing hint
- TLP prefix
- AtomicOP
- Any features not related to endpoints

4.2 Physical Layer

The Tsi721 is fully compliant with the *PCI Express Specification (Rev. 2.1)*: Physical Layer Specification. In addition, the device provides the following status registers:

- Lane Status 0 Register: Per-lane error status (8b/10b errors, disparity errors, and so on)
- PHY Link Status 0 Register: Link framing errors.
- PHY Link Status 1 Register: Link training sequence errors.
- PHY Countables 0 Register, PHY Countables 1 Register, and PHY Countables Configuration Register: Link programmable counters to count error conditions.

4.3 Data Link Layer

The Tsi721 is fully compliant with the *PCI Express Specification (Rev. 2.1)*: Data Link Layer Specification.

As per ECN “ASPM Optionality,” approved August 20, 2009 against the *PCI Express Base Specification (Rev. 2.1)*, ASPM support is optional. The Tsi721 does not support ASPM. For more information, see the *Tsi721 Device Errata*.

4.4 Base Address Registers

- In the PCI architecture, Base Address Registers (BARs) implement a function’s memory and I/O address decoders.
- The PCIe MAC supports four BARs (six BAR registers, BAR 0 to BAR 5):
  - PCI BAR 0 forms a 32-bit BAR for Tsi721 internal address space
    - It must be non-prefetchable
    - The minimum BAR size is 512 KB
  - PCI BAR 1 forms a 32-bit BAR for outbound doorbells
    - It must be non-prefetchable
    - The minimum BAR size is 16 MB
  - PCI BAR 2/PCI BAR 3 are paired to form a 64-bit BAR for PCIe to S-RIO address translation except MWr to doorbell translation
    - It must be prefetchable
    - The minimum BAR size is 16 MB (MWr to Maintenance write with 24-bit S-RIO maintenance address)
  - PCI BAR 4/PCI BAR 5 are paired to form a 64-bit BAR for PCIe to S-RIO address translation except MWr to doorbell translation
    - It must be non-prefetchable
    - The minimum BAR size is 16 MB (MWr to maintenance write with 24-bit S-RIO maintenance address).
- Each BAR has a corresponding setup register. For example, PCI BAR 0 has an associated BAR 0 Setup Register. BAR setup registers allow a BAR to be enabled or disabled, control the operating mode of the BAR as well as advertised capabilities (for example, size of the BAR window).
- When an even BAR is configured for 64-bit operation, the odd BAR takes on the function of the upper 32-bits of the BADDR field of the even BAR. When an even and odd BAR are merged for 64-bit operation, the fields of the odd BAR Setup register remain read-write but have no functional effect on the operation of the device.

4.5 TLP Receive Processing

The PCIe MAC accepts the following TLPs:

- MRd
- MWr
- Cpl
- CplD
- CfgWr0
- CfgRd0
- PME_TURN_OFF message
- PM_Active_State_Nak message

- Unlocked message
- Vendor defined type 1 messages

On PCIe ingress side, the Tsi721 does not support PCIe TLP prefix and TLP Processing Hints (TPH)

- Receiving a TLP violating the above rules will trigger Tsi721 PCIe error processing (see Interrupt Moderation)

4.5.1 Memory Read Request (MRd)

If the MAE bit is set in the PCI Control and Status Register and the address of the memory read request matches that of an enabled memory BAR , then the TLP is passed to the application layer. Otherwise the request is handled as an Unsupported Request.

4.5.2 Memory Write Request (MWr)

If the MAE bit is set in the PCI Control and Status Register and the address of the memory write request matches that of an enabled memory BAR, then the TLP is passed to the application layer. Otherwise the request is handled as an unsupported request.

4.5.3 Configuration Read Type 0 (CfgRd0)

If the read request does not contain an ECRC error, then the PCIe MAC processes the configuration read request by generating a completion that contains the value of the corresponding configuration register.

- Only the value of bytes that are enabled in the request are read and returned in the completion.

If the configuration read request contains an ECRC error, then the request is silently discarded.

4.5.4 Configuration Write Type 0 (CfgWr0)

If the write request does not contain an ECRC error, and the configuration write request is poisoned, then the request is discarded and a completion with status UR is generated. The PCIe MAC processes the configuration write request by updating the contents of enabled bytes in the corresponding configuration register with the data in the request.

- The completer ID in the generated completion is set to that of the PCIe MAC.
- The requester ID, tag, and traffic class in the generated completion, are the same as that in the corresponding request.

If the configuration write request contains an ECRC error, then the request is silently discarded.

4.5.5 Configuration Read and Write Type 1 (CfgRd1 and CfgWr1)

Type 1 configuration read and write requests received by the PCIe MAC are handled as Unsupported Request errors.

4.5.6 Assert_INTx and Deassert_INTx

The PCIe MAC handles Assert_INTx and Deassert_INTx messages as malformed TLPs.

4.5.7 PM_Active_State_Nak

The PM_Active_State_Nak message is processed by the PCIe MAC for L1 ASPM and is used by the upstream component to reject an L1 ASPM entry request.

4.5.8 PM_PME

The PCIe MAC handles PM_PME messages as Unsupported Requests.

4.5.9 PME_Turn_Off

The PME_Turn_Off message is processed by the PCIe MAC and the Tsi721 generates PME_to_ACK message.

4.5.10 PME_TO_Ack

The PCIe MAC handles PM_TO_Ack messages as malformed TLPs.

4.5.11 Error Messages (ERR_COR, ERR_NONFATAL, and ERR_FATAL)

The PCIe MAC handles error messages (that is, ERR_COR, ERR_NONFATAL, ERR_FATAL) as Unsupported Requests.

4.5.12 Unlock

The PCIe MAC silently discards unlocked messages (the TLP is acknowledged, discarded, and flow control credits are returned).

4.5.13 Set_Slot_Power_Limit

The PCIe MAC processes Set_Slot_Power_Limit messages:

- The data payload is written into the PCIe Device Capabilities Register.
  - — Bits 1:0 of Byte 1 of the data payload map to the CSPLS field.
  - — Bits 7:0 of Byte 0 map to the CSPLV field.
  - — Bits 7:0 of Byte 3, 7:0 of Byte 2, and 7:2 of Byte 1 are ignored.

4.5.14 Vendor-Defined Messages

All received vendor-defined messages are processed by the PCIe MAC:

- Vendor-defined type 0 messages received by the PCIe MAC are handled as Unsupported Requests.
- Vendor-defined type 1 messages received by the PCIe MAC are silently discarded (the TLP is acknowledged, discarded, and flow control credits are returned).

4.6 Transmit TLP Processing

- All Tsi721 generated TLPs have VC0 and TC0
- Tsi721 generates the following TLPs:
  - — Assert_INTx and Deassert_INTx messages when INTx interrupts are enabled and requested
  - — PME_TO_ack in response to the PCIe-PM power removal sequence
  - — Error messages (ERR_COR, ERR_NONFATAL, ERR_FATAL) generated by Advanced Error Reporting (AER)
  - — MSI/ MSI-X MWr
  - — MWr
  - — MRd
  - — Cpl/CplD

The completion timeout period for the Tsi721 may need to be adjusted to account for latency on the PCIe bus. For the completion timeout capabilities of the Tsi721, see CTDS and CTRS in the PCIe Device Capabilities 2 Register. To disable completion timeouts, set CTD to 1 in the PCIe Device Control and Status 2 Register. The CTV field in the same register controls the completion timeout period when the CTD field is clear.

4.7 Buffers

The PCIe MAC contains an Ingress Frame Buffer (IFB) and an Egress Frame Buffer (EFB). These buffers are managed by the PCIe MAC and are not directly visible to the application layer.

4.7.1 Ingress Frame Buffer

The PCIe MAC uses an input buffer called the Ingress Frame Buffer (IFB). The IFB consists of three queues (see Table 25). The queues for the IFB are implemented using a descriptor memory and a data memory. When a TLP is received from the link, the packet is stored in the appropriate queue. The default size of each queue is displayed in the following table.

Table 25: Ingress Frame Buffer Sizes

Queue Type	Total Size and Limitations	Advertised Data Credits	Advertised Header Credits
Posted Transaction (PT)	12352 bytes and up to 127 TLPs	772	127
Non-posted Transaction (NP)	2048 bytes and up to 127 TLPs	128	127
Completion Transaction (CP)	12352 bytes and up to 127 TLPs	772	127

4.7.2 Egress Frame Buffer

The PCIe MAC uses an output buffer called an Egress Frame Buffer (EFB). The EFB provides queuing to the application layer by allowing packets to be stored in an egress port’s EFB even if the port’s link does not have sufficient credits to accept the packet.

The EFB is also used as a dynamically sized replay buffer. This allows for efficient use of the egress buffer space: when transmitted packets are not being acknowledged by the link partner the replay buffer grows to allow further transmission; when transmitted packets are successfully acknowledged by the link partner the replay buffer shrinks and this space can be used for general queuing.

Each EFB consists of three queues (see Table 26). The use of these queues allows for packet re-ordering to improve transmission efficiency on the egress link. The queues for both EFBs are implemented using a descriptor memory and a data memory. The default size of each of these queues is displayed in the following table.

Table 26: Egress Frame Buffer Sizes

Queue Type	Total Size and Limitations
Posted Transaction	12352 bytes and up to 128 TLPs
Non-posted Transaction	2048 bytes and up to 128 TLPs
Completion Transaction	12352 bytes and up to 128 TLPs

The maximum number of TLPs that can be stored in the EFB’s replay buffer is 64 TLPs.

4.8 Device Power Management

- The PCIe MAC supports the following device power management states:
  - — D0 (D0$_{uninitialized}$ and D0$_{active}$)
  - — D3$_{Hot}$
  - — D3$_{Cold}$.
- A power management state transition diagram for the states supported by the PCIe MAC is provided in Figure 16 and described in Table 27.
- Transitioning the PCIe MAC's power management state from D3$_{hot}$ to D0$_{uninitialized}$ does not result in any logic being reset or re-initialization of register values. Thus, the default value of the NOSOFTRST bit in the function's PCI Power Management Control and Status Register corresponds to the functional context being maintained in the D3$_{hot}$ state.
- When the PCIe MAC enters the D0 state (that is, D0$_{uninitialized}$ or D0$_{active}$), it transitions the link to the L0 state. When the PCIe MAC enters the D3$_{Hot}$ state, it transitions the link to the L1 state.
  - — Entry into the L1 state as a result of the function being placed in D3$_{Hot}$ can be disabled by setting the D3HOTL1D bit in the function's Power Management Proprietary Control 0 Register.

Figure 16: Device Power Management State Transitions

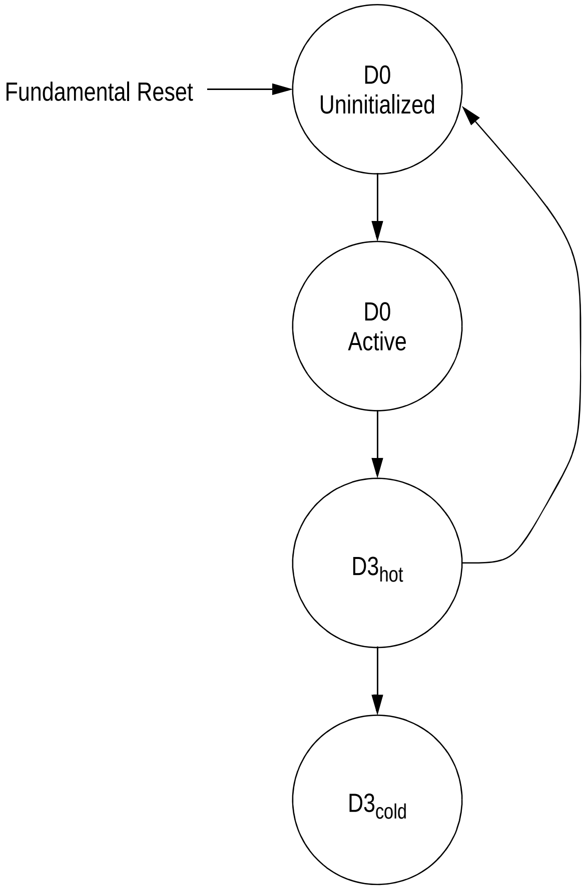


Table 27: Device Power Management State Transitions

From State	To State	Description
Any	D0 Uninitialized	Fundamental Reset
D0 Uninitialized	D0 Active	PCIe MAC configured by software
D0 Active	$D3_{hot}$	PSTATE field in the PCI Power Management Control and Status Register is written with the value that corresponds to the $D3_{hot}$ state.
$D3_{hot}$	D0 Uninitialized	PSTATE field in the PCI Power Management Control and Status Register is written with the value that corresponds to D0 state.
$OD3_{hott}$	$D3_{cold}$	Power is removed from the device.

The PCIe MAC has the following behavior when in the $D3_{hot}$ power management state:

- Accepts, processes, and completes all type 0 configuration read and write requests
- Accepts and processes all message requests that target the endpoint
- All requests received by the endpoint, except as noted above, are handled as unsupported requests (UR).
- Any error message resulting from the receipt of a TLP is reported in the same way as when the endpoint is not in $D3_{hot}$ (for example, generation of an ERR_NONFATAL message to the root).
- This requires transitioning the link to the L0 state when error reporting is enabled and the link is not in L0.
- Error messages resulting from any event other than the receipt of a TLP are discarded (that is, no error message is generated).
- All completions that target the endpoint are handled as Unexpected Completions.

4.8.1 PME Messages

The PCIe MAC does not support generation of PME messages from the D3 state.

4.8.2 PCIe Power Management Fence Protocol

The root complex takes the following steps to turn off power to a system:

1. The root places all devices in the D3 state.
2. Upon entry to D3, all devices transition their links to the L1 state.
3. The root broadcasts a PME_Turn_Off message.

   The links temporarily transition to L0 in order to transfer the message.
4. Devices acknowledge the PME_Turn_Off message by returning a PME_TO_ACK message.

   After transmitting a PME_TO_ACK, a device places its link in L2/L3-Ready state.
5. The PME_Turn_Off / PME_TO_Ack protocol can be initiated by the root complex when the PCIe MAC is in any power management state.
6. The power fence protocol is abandoned by the PCIe MAC when the port receives a TLP after having previously received a PME_Turn_Off message but before having responded with a PME_TO_Ack message. This TLP is processed normally by the PCIe MAC. Once a PME_TO_Ack message has been scheduled for transmission, received TLPs at that point are discarded.

5. Interrupt Handling

Topics discussed include the following:

- Overview
- Interrupt Sources
- MSI-X
- MSI
- INTx
- Interrupt Moderation
- PCIe Advisory Error Reporting
- SMBus Alert Response Hierarchy

5.1 Overview

The PCIe Interface can generate MSI, MSI-X, and legacy INTx messages. INTx / MSI/ MSI-X generation is moderated by a free running interrupt timer. An INTx/ MSI / MSI-X is not generated until the timer has expired.

Table 28: Interrupt Register Cross Reference

Register Mnemonic	See
PCI_CSR	PCI Control and Status Register
INTSTS	Legacy Interrupt Status Register
MSIADDR	MSI Address Register
MSIUADDR	MSI Upper Address Register
MSIMDATA	MSI Message Data Register
MSICAP	MSI Capability and Control Register
MSIMASK	MSI Mask Register
MSIPENDING	MSI Pending Register
MSIXCAP	MSI-X Capability and Control Register
MSIXTBL	MSI-X Table Offset Register
MSIXPBA	MSI-X Pending Bit Array Offset Register
MSIX_PBAL	MSI-X Pending Bit Array Lower Register
MSIX_PBAM	MSI-X Pending Bit Array Middle Register

Table 28: Interrupt Register Cross Reference *(Continued)*

Register Mnemonic	See
MSIX_PBAU	MSI-X Pending Bit Array Upper Register
MSIX_TAB_ADDRL{0..69}	MSI-X Table Entry Message Lower Address Register {0..69}
MSIX_TAB_ADDRU{0..69}	MSI-X Table Entry Message Upper Address Register {0..69}
MSIX_TAB_DATA{0..69}	MSI-X Table Entry Message Data Register {0..69}
MSIX_TAB_MSK{0..69}	MSI-X Table Entry Mask Register {0..69}
DEV_INTE	Tsi721 Interrupt Enable CSR
DEV_INT	Tsi721 Interrupt CSR
DEV_CHAN_INTE	Tsi721 Channel Interrupt Enable CSR
DEV_CHAN_INT	Tsi721 Channel Interrupt CSR
SR_CH{0..7}_INT	SR2PC Channel Interrupt Register {0..7}
SR_CH{0..7}_INTE	SR2PC Channel Interrupt Enable Register {0..7}
SR2PC_GEN_INT	SR2PC General Interrupt CSR
SR2PC_GEN_INTE	SR2PC General Interrupt Enable CSR
PC2SR_INT	PC2SR Interrupt CSR
PC2SR_INTE	PC2SR Interrupt Enable CSR
DMAC{0..7}INT	DMA Channel Interrupt Register {0..7}
DMAC{0..7}INTE	DMA Channel Interrupt Enable Register {0..7}
BDMA_INT	BDMA Interrupt CSR
BDMA_INTE	BDMA Interrupt Enable CSR
OBDMAC{0..7}INT	Outbound DMA Channel Interrupt Register {0..7}
OBDMAC{0..7}INTE	Outbound DMA Channel Interrupt Enable Register {0..7}
IBDMAC{0..7}INT	Inbound DMA Channel Interrupt Register {0..7}
IBDMAC{0..7}INTE	Inbound DMA Channel Interrupt Enable Register {0..7}
SMSG_INT	Messaging Engine Interrupt Register
SMSG_INTE	Messaging Engine Interrupt Enable Register
I2C_INT_ENABLE	I2C Interrupt Enable Register
I2C_INT_STAT	I2C Interrupt Status Register

5.2 Interrupt Sources

The Tsi721 generates interrupts in support of the following conditions:

- I2C events
- S-RIO MAC events
- PCIe MAC events
- Mapping Engine events
- Block DMA Engine events
- S-RIO Messaging Engine events

5.3 MSI-X

The Tsi721 supports 70 MSI-X vectors:

- MSI-X table size is 70
- MSI-X table offset is 0x2C000
- MSI-X table BIR is 0x0 (relative to BAR0)
- MSI-X PBA size is 70
- MSI-X PBA offset is 0x2A000
- MSI-X PBA BIR is 0x0 (relative to BAR0)

Besides Tsi721 interrupt registers and interrupt enable registers displayed in Table 29, the following registers must be configured for MSI-X operations:

- MSI-X Capability and Control Register
- MSI-X Table Offset Register
- MSI-X Pending Bit Array Offset Register
- PCI Power Management Capabilities Register
- PCI Power Management Control and Status Register
- MSI-X Pending Bit Array Lower Register
- MSI-X Table Entry Message Lower Address Register {0..69}[1]
- MSI-X Table Entry Message Upper Address Register {0..69}[1]
- MSI-X Table Entry Message Data Register {0..69}[1]
- MSI-X Table Entry Mask Register {0..69}[1]

[1] 0–69 is the MSI-X vector number.

5.3.1 MSI-X Vector Assignment

Table 29: MSI-X Vector Assignment

MSI-X Vector[a]	Source Interrupt Bits	Source/Top Interrupt Enable and Top Interrupt Bits
0–7	Vector x (x = 0–7) is associated with the following interrupt bits: • DMA Channel Interrupt Register {0..7}.IOF_DONE	Vector x (x = 0–7) is associated with the following interrupt bits: • DMA Channel Interrupt Enable Register {0..7}.IOF_DONE_EN must be fixed at 1
8–15	Vector x+8 (x = 0–7) is associated with the following interrupt bits: • All bits of DMA Channel Interrupt Register {0..7} except IOF_DONE	Vector x+ 8 (x = 0–7) is associated with the following interrupt bits: • All bits of DMA Channel Interrupt Enable Register {0..7} except IOF_DONE_EN
16	Vector 16 is associated with the following interrupt bits: • All bits of BDMA Interrupt CSR	Vector 16 is associated with the following interrupt bits: • All bits of BDMA Interrupt Enable CSR • Tsi721 Interrupt CSR.INT_BDMA_NONCH • Tsi721 Interrupt Enable CSR.INT_BDMA_NONCH_EN
17–24	Vector x+17 (x = 0–7) is associated with the following interrupt bits: • Outbound DMA Channel Interrupt Register {0..7}.IOF_DONE	Vector x+17 (x = 0–7) is associated with the following interrupt bits: • Outbound DMA Channel Interrupt Enable Register {0..7}.IOF_DONE_EN must be fixed at 1
25–32	Vector x+25 (x = 0–7) is associated with the following interrupt bits: • All bits of Outbound DMA Channel Interrupt Register {0..7} except IOF_DONE	Vector x+25 (x = 0–7) is associated with the following interrupt bits: • All bits of Outbound DMA Channel Interrupt Enable Register {0..7} except IOF_DONE_EN
33–40	Vector x+33 (x = 0–7) is associated with the following interrupt bits: • Inbound DMA Channel Interrupt Register {0..7}.DQ_RCV	Vector x+33 (x = 0–7) is associated with the following interrupt bits: • Inbound DMA Channel Interrupt Enable Register {0..7}.DQ_RCV_EN must be fixed at 1
41–48	Vector x+41 (x = 0–7) is associated with the following interrupt bits: • All bits of Inbound DMA Channel Interrupt Register {0..7} except DQ_RCV	Vector x+41 (x = 0–7) is associated with the following interrupt bits: • All bits of Inbound DMA Channel Interrupt Enable Register {0..7} except DQ_RCV_EN
49	Vector 49 is associated with the following interrupt bits: • All bits of Messaging Engine Interrupt Register	Vector 49 is associated with the following interrupt bits: • All bits of Messaging Engine Interrupt Enable Register • Tsi721 Interrupt CSR.INT_SMSG_NONCH • Tsi721 Interrupt Enable CSR.INT_SMSG_NONCH_EN

Table 29: MSI-X Vector Assignment *(Continued)*

MSI-X Vector[a]	Source Interrupt Bits	Source/Top Interrupt Enable and Top Interrupt Bits
50–57	Vector x+50 (x = 0–7) is associated with the following interrupt bits: • SR2PC Channel Interrupt Register {0..7}.IDBQ_RCV	Vector x+50 (x = 0–7) is associated with the following interrupt bits: • SR2PC Channel Interrupt Enable Register {0..7}.IDBQ_RCV_EN must be fixed at 1
58–65	Vector x+58 (x = 0–7) is associated with the following interrupt bits: • All bits of SR2PC Channel Interrupt Register {0..7} except IDBQ_RCV	Vector x+58 (x = 0–7) is associated with the following interrupt bits: • All bits of SR2PC Channel Interrupt Enable Register {0..7} except IDBQ_RCV_EN
66	Vector 66 is associated with the following interrupt bits: • All bits of SR2PC General Interrupt CSR	Vector 66 is associated with the following interrupt bits: • All bits of SR2PC General Interrupt Enable CSR • Tsi721 Interrupt CSR.INT_SR2PC_NONCH • Tsi721 Interrupt Enable CSR.INT_SR2PC_NONCH_EN
67	Vector 67 is associated with the following interrupt bits: • All bits of PC2SR Interrupt CSR	Vector 67 is associated with the following interrupt bits: • All bits of PC2SR Interrupt Enable CSR • Tsi721 Interrupt CSR.INT_PC2SR • Tsi721 Interrupt Enable CSR.INT_PC2SR_EN
68	Vector 68 is associated with the following interrupt bits: • All interrupt bit within S-RIO MAC (see Figure 19)	Vector 68 is associated with the following interrupt bits: • All interrupt register enable bit within S-RIO MAC (see Figure 19) • Tsi721 Interrupt CSR.INT_SRIO • Tsi721 Interrupt Enable CSR.INT_SRIO_EN
69	Vector 69 is associated with the following interrupt bits: • All bits of I2C Interrupt Status Register	Vector 69 is associated with the following interrupt bits: • All bits of I2C register I2C Interrupt Enable Register • Tsi721 Interrupt CSR.INT_I2C • Tsi721 Interrupt Enable CSR.INT_I2C_EN

[a] Some vectors are reserved for future purposes.

5.4 MSI

The Tsi721 supports only one MSI vector, vector 0. All Tsi721 interrupt events are mapped into vector 0. In addition to the Tsi721 interrupt registers and interrupt enable registers displayed in Table 29, the following registers must be configured for MSI operations:

- MSI Capability and Control Register
- MSI Address Register
- MSI Upper Address Register
- MSI Message Data Register
- MSI Pending Register
- MSI Mask Register

5.4.1 MSI Vector 0 Interrupt Hierarchy

Table 30: MSI Vector 0 Interrupt Hierarchy

Interrupt Tree Leaf	Interrupt Tree Level 1	Interrupt Tree Level 2	Interrupt Tree Root
BDMA channel x interrupt/enable/status/set registers: • DMA Channel Interrupt Register {0..7} • DMA Channel Interrupt Enable Register {0..7} • DMA Channel Status Register {0..7} • DMA Channel Interrupt Set Register {0..7}	SR2PC interrupt/enable bits: • Tsi721 Channel Interrupt CSR.INT_BDMA_CHAN[x] • Tsi721 Channel Interrupt Enable CSR. INT_BDMA_CHAN_EN[x]	SR2PC interrupt/enable/set bits: • Tsi721 Interrupt CSR.INT_BDMA_CH • Tsi721 Interrupt Enable CSR.INT_BDMA_CH_EN • Tsi721 Interrupt Enable CSR.INT_BDMA_CH_SET	PCIe MAC bit: • MSI Capability and Control Register.EN • MSI Mask Register.MASK[0] • MSI Pending Register.PENDING[0]
BDMA non-channelized interrupt/enable/status/set registers: • BDMA Interrupt CSR • BDMA Interrupt Enable CSR • BDMA Interrupt Set Register • BDMA ECC Log Register	SR2PC interrupt/enable/set bits: • Tsi721 Interrupt CSR.INT_BDMA_NONCH • Tsi721 Interrupt Enable CSR.INT_BDMA_NONCH_EN • Tsi721 Interrupt Enable CSR.INT_BDMA_NONCH_SET	N/A	PCIe MAC bit: • MSI Capability and Control Register.EN • MSI Mask Register.MASK[0] • MSI Pending Register.PENDING[0]

Table 30: MSI Vector 0 Interrupt Hierarchy *(Continued)*

Interrupt Tree Leaf	Interrupt Tree Level 1	Interrupt Tree Level 2	Interrupt Tree Root
SMSG channel x interrupt/enable/status/set registers: • Outbound DMA Channel Interrupt Register {0..7} • Outbound DMA Channel Interrupt Enable Register {0..7} • Outbound DMA Channel Status Register {0..7} • Outbound DMA Channel Interrupt Set Register {0..7} • Inbound DMA Channel Interrupt Register {0..7} • Inbound DMA Channel Interrupt Enable Register {0..7} • Inbound DMA Channel Status Register {0..7} • Inbound DMA Channel Interrupt Set Register {0..7}	SR2PC interrupt/enable bits: • Tsi721 Channel Interrupt CSR.INT_IBMSG_CHAN[x] • Tsi721 Channel Interrupt CSR.INT_OBMSG_CHAN[x] • Tsi721 Channel Interrupt CSR.INT_IBMSG_CHAN_EN[x] • Tsi721 Channel Interrupt CSR.INT_OBMSG_CHAN_EN[x]	SR2PC interrupt/enable/set bits: • Tsi721 Interrupt CSR.INT_SMSG_CH • Tsi721 Interrupt Enable CSR.INT_SMSG_CH_EN • Tsi721 Interrupt Enable CSR.INT_SMSG_CH_SET	PCIe MAC bit: • MSI Capability and Control Register.EN • MSI Mask Register.MASK[0] • MSI Pending Register.PENDING[0]
SMSG non-channelized interrupt/enable/status/set registers: • Messaging Engine Interrupt Register • Messaging Engine Interrupt Enable Register • Messaging Engine Interrupt Set Register • Messaging Engine non-channelized ECC LOG Register	SR2PC interrupt/enable/set bits: • Tsi721 Interrupt CSR.INT_SMSG_NONCH • Tsi721 Interrupt Enable CSR.INT_SMSG_NONCH_EN • Tsi721 Interrupt Enable CSR.INT_SMSG_NONCH_SET	N/A	PCIe MAC bit: • MSI Capability and Control Register.EN • MSI Mask Register.MASK[0] • MSI Pending Register.PENDING[0]

This document is confidential and is subject to an NDA.

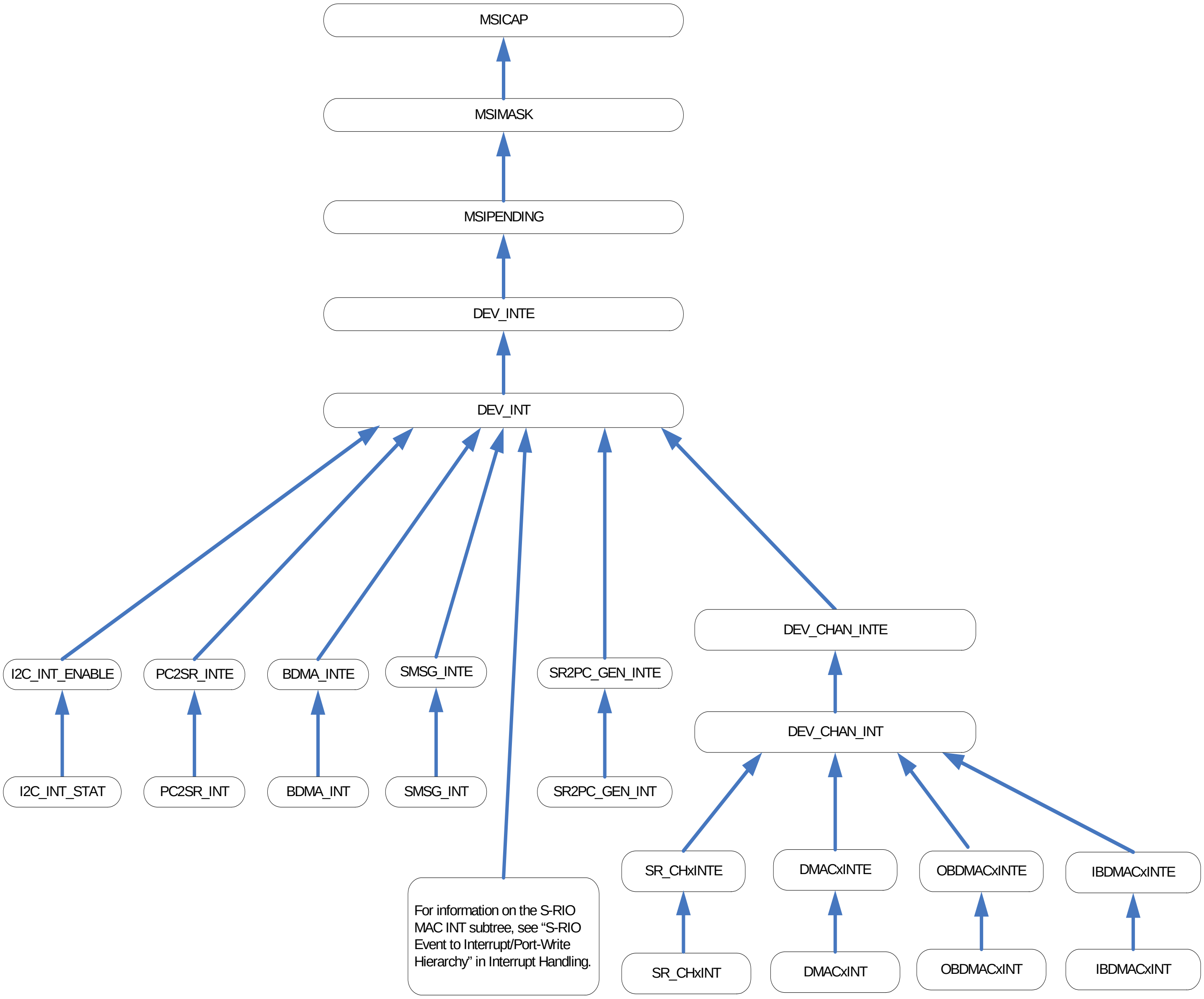
Table 30: MSI Vector 0 Interrupt Hierarchy *(Continued)*

Interrupt Tree Leaf	Interrupt Tree Level 1	Interrupt Tree Level 2	Interrupt Tree Root
SR2PC channel x interrupt/enable/status/set registers: • SR2PC Channel Interrupt Register {0..7} • SR2PC Channel Interrupt Enable Register {0..7} • SR2PC Outbound Doorbell Response Log Buffer Status Register {0..7} • SR2PC Channel Interrupt Set Register {0..7}	SR2PC interrupt/enable bits: • Tsi721 Channel Interrupt CSR.INT_SR2PC_CHAN[x] • Tsi721 Channel Interrupt CSR.INT_SR2PC_CHAN_EN[x]	SR2PC interrupt/enable/set bits: • Tsi721 Interrupt CSR.INT_SR2PC_CH • Tsi721 Interrupt Enable CSR.INT_SR2PC_CH_EN • Tsi721 Interrupt Enable CSR.INT_SR2PC_CH_SET	PCIe MAC bit: • MSI Capability and Control Register.EN • MSI Mask Register.MASK[0] • MSI Pending Register.PENDING[0]
SR2PC non-channelized interrupt/enable/status/set registers: • SR2PC General Interrupt CSR • SR2PC General Interrupt Enable CSR • SR2PC Channel Interrupt Set Register {0..7} • SR2PC Correctable ECC Log Register	SR2PC interrupt/enable/set bits: • Tsi721 Interrupt CSR.INT_SR2PC_NONCH • Tsi721 Interrupt Enable CSR.INT_SR2PC_NONCH_EN • Tsi721 Interrupt Enable CSR.INT_SR2PC_NONCH_SET	N/A	PCIe MAC bit: • MSI Capability and Control Register.EN • MSI Mask Register.MASK[0] • MSI Pending Register.PENDING[0]
PC2SR interrupt/enable/status/set registers: • PC2SR Interrupt CSR • PC2SR Interrupt Enable CSR • PC2SR Interrupt Set Register • PC2SR ECC Log Register	SR2PC interrupt/enable/set bits: • Tsi721 Interrupt CSR.INT_PC2SR • Tsi721 Interrupt Enable CSR.INT_PC2SR_EN • Tsi721 Interrupt Enable CSR.INT_PC2SR_SET	N/A	PCIe MAC bit: • MSI Capability and Control Register.EN • MSI Mask Register.MASK[0] • MSI Pending Register.PENDING[0]

Table 30: MSI Vector 0 Interrupt Hierarchy *(Continued)*

Interrupt Tree Leaf	Interrupt Tree Level 1	Interrupt Tree Level 2	Interrupt Tree Root
I2C interrupt/enable/status/set registers: • I2C Interrupt Enable Register • I2C Interrupt Set Register • I2C Interrupt Status Register	SR2PC interrupt/enable bits: • Tsi721 Interrupt CSR.INT_I2C • Tsi721 Interrupt Enable CSR.INT_I2C_EN	N/A	PCIe MAC bit: • MSI Capability and Control Register.EN • MSI Mask Register.MASK[0] • MSI Pending Register.PENDING[0]
S-RIO MAC has internal interrupt hierarchy (see Figure 19)		SR2PC interrupt/enable/set bits: • Tsi721 Interrupt CSR.INT_SRIO • Tsi721 Interrupt Enable CSR.INT_SRIO_EN • Tsi721 Interrupt Enable CSR.INT_SRIO_SE	PCIe MAC bit: • MSI Capability and Control Register.EN • MSI Mask Register.MASK[0] • MSI Pending Register.PENDING[0]

Figure 17: MSI Interrupt Hierarchy



For information on the S-RIO MAC INT subtree, see “S-RIO Event to Interrupt/Port-Write Hierarchy” in Interrupt Handling.

5.5 INTx

The Tsi721 supports legacy INTx interrupts; however, only INTA is supported. All Tsi721 interrupt events are mapped into INTA.

5.5.1 INTx Interrupt Hierarchy

Table 31: INTx Interrupt Hierarchy

Interrupt Tree Leaf	Interrupt Tree Level 1	Interrupt Tree Level 2	Interrupt Tree Root
BDMA channel x interrupt/enable/status/set registers: • DMA Channel Interrupt Register {0..7} • DMA Channel Interrupt Enable Register {0..7} • DMA Channel Status Register {0..7} • DMA Channel Interrupt Set Register {0..7}	SR2PC interrupt/enable bits: • Tsi721 Channel Interrupt CSR.INT_BDMA_CHAN[x] • Tsi721 Channel Interrupt CSR.INT_BDMA_CHAN_EN[x]	SR2PC interrupt/enable/set bits: • Tsi721 Interrupt CSR.INT_BDMA_CH • Tsi721 Interrupt Enable CSR.INT_BDMA_CH_EN • Tsi721 Interrupt Enable CSR.INT_BDMA_CH_SET	PCIe MAC bit: • PCI Control and Status Register.INTXD • PCI Miscellaneous 1 Register.INTRPIN • Legacy Interrupt Status Register.INTA
BDMA non-channelized interrupt/enable/status/set registers: • BDMA Interrupt CSR • BDMA Interrupt Enable CSR • BDMA Interrupt Set Register • BDMA ECC Log Register	SR2PC interrupt/enable/set bits: • Tsi721 Interrupt CSR.INT_BDMA_NONCH • Tsi721 Interrupt Enable CSR.INT_BDMA_NONCH_EN • Tsi721 Interrupt Enable CSR.INT_BDMA_NONCH_SET	N/A	PCIe MAC bit: • PCI Control and Status Register.INTXD • PCI Miscellaneous 1 Register.INTRPIN • Legacy Interrupt Status Register.INTA

Table 31: INTx Interrupt Hierarchy *(Continued)*

Interrupt Tree Leaf	Interrupt Tree Level 1	Interrupt Tree Level 2	Interrupt Tree Root
SMSG channel x interrupt/enable/status/set registers: • Outbound DMA Channel Interrupt Register {0..7} • Outbound DMA Channel Interrupt Enable Register {0..7} • Outbound DMA Channel Status Register {0..7} • Outbound DMA Channel Interrupt Set Register {0..7} • Inbound DMA Channel Interrupt Register {0..7} • Inbound DMA Channel Interrupt Enable Register {0..7} • Inbound DMA Channel Status Register {0..7} • Inbound DMA Channel Interrupt Set Register {0..7}	SR2PC interrupt/enable bits: • Tsi721 Channel Interrupt CSR.INT_IBMSG_CHAN[x] • Tsi721 Channel Interrupt CSR.INT_OBMSG_CHAN[x] • Tsi721 Channel Interrupt CSR.INT_IBMSG_CHAN_EN[x] • Tsi721 Channel Interrupt CSR.INT_OBMSG_CHAN_EN[x]	SR2PC interrupt/enable/set bits: • Tsi721 Interrupt CSR.INT_IBMSG_CH • Tsi721 Interrupt Enable CSR.INT_SMSG_CH_EN • Tsi721 Interrupt Enable CSR.INT_SMSG_CH_SET	PCIe MAC bit: • PCI Control and Status Register.INTXD • PCI Miscellaneous 1 Register.INTRPIN • Legacy Interrupt Status Register.INTA
SMSG non-channelized interrupt/enable/status/set registers: • Messaging Engine Interrupt Register • Messaging Engine Interrupt Enable Register • Messaging Engine Interrupt Set Register • Messaging Engine non-channelized ECC LOG Register	SR2PC interrupt/enable/set bits: • Tsi721 Interrupt CSR.INT_SMSG_NONCH • Tsi721 Interrupt Enable CSR.INT_SMSG_NONCH_EN • Tsi721 Interrupt Enable CSR.INT_SMSG_NONCH_SET	N/A	PCIe MAC bit: • PCI Control and Status Register.INTXD • PCI Miscellaneous 1 Register.INTRPIN • Legacy Interrupt Status Register.INTA

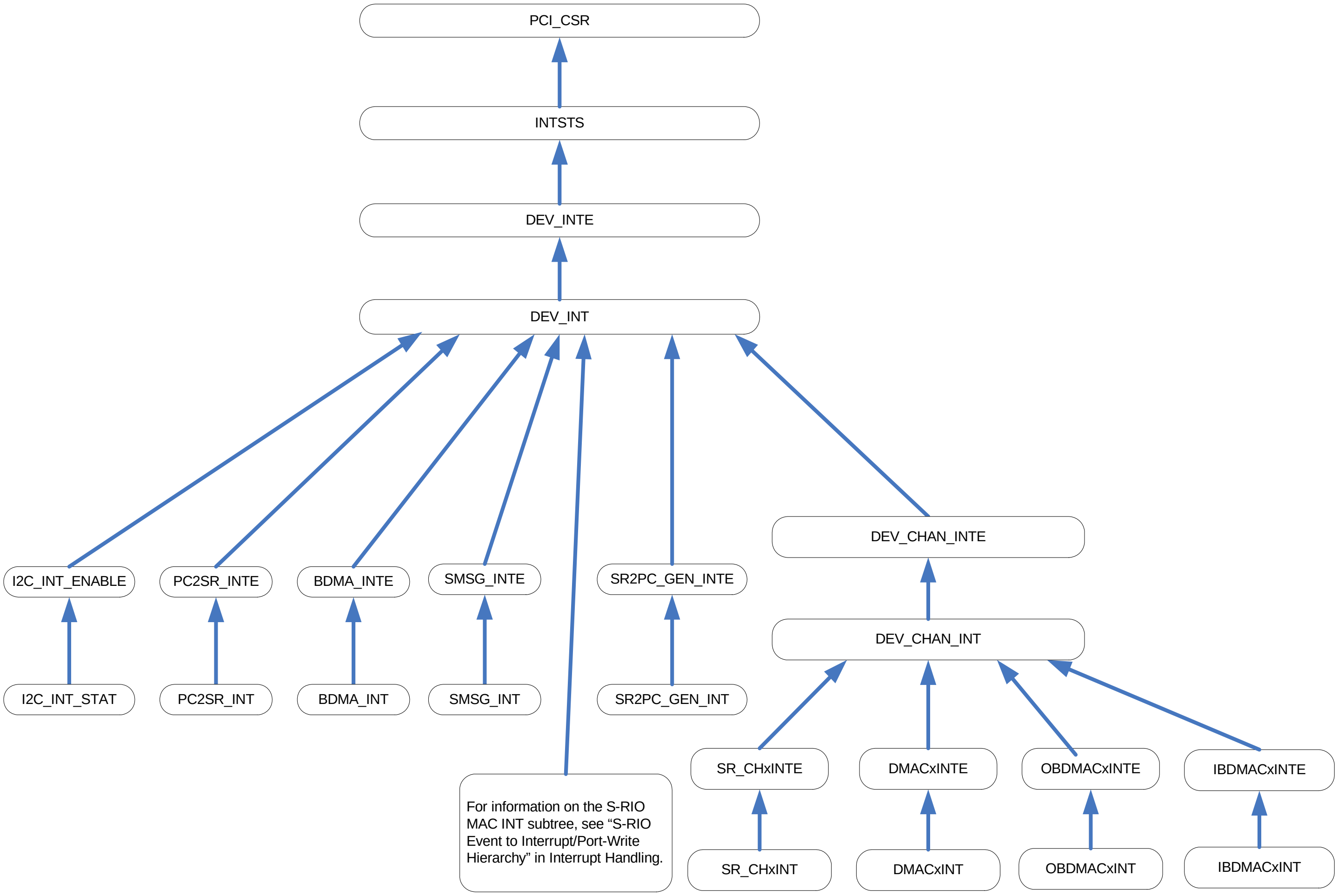
Table 31: INTx Interrupt Hierarchy *(Continued)*

Interrupt Tree Leaf	Interrupt Tree Level 1	Interrupt Tree Level 2	Interrupt Tree Root
SR2PC channel x interrupt/enable/status/set registers: • SR2PC Channel Interrupt Register {0..7} • SR2PC Channel Interrupt Enable Register {0..7} • SR2PC Outbound Doorbell Response Log Buffer Status Register {0..7} • SR2PC Channel Interrupt Set Register {0..7}	SR2PC interrupt/enable bits: • Tsi721 Channel Interrupt CSR.INT_SR2PC_CHAN[x] • Tsi721 Channel Interrupt CSR.INT_SR2PC_CHAN_EN[x]	SR2PC interrupt/enable/set bits: • Tsi721 Interrupt CSR.INT_SR2PC_CH • Tsi721 Interrupt Enable CSR.INT_SR2PC_CH_EN • Tsi721 Interrupt Enable CSR.INT_SR2PC_CH_SET	PCIe MAC bit: • PCI Control and Status Register.INTXD • PCI Miscellaneous 1 Register.INTRPIN • Legacy Interrupt Status Register.INTA
SR2PC non-channelized interrupt/enable/status/set registers: • SR2PC General Interrupt CSR • SR2PC General Interrupt Enable CSR • SR2PC Channel Interrupt Set Register {0..7} • SR2PC Correctable ECC Log Register	SR2PC interrupt/enable/set bits: • Tsi721 Interrupt CSR.INT_SR2PC_NONCH • Tsi721 Interrupt Enable CSR.INT_SR2PC_NONCH_EN • Tsi721 Interrupt Enable CSR.INT_SR2PC_NONCH_SET	N/A	PCIe MAC bit: • PCI Control and Status Register.INTXD • PCI Miscellaneous 1 Register.INTRPIN • Legacy Interrupt Status Register.INTA
PC2SR interrupt/enable/status/set registers: • PC2SR Interrupt CSR • PC2SR Interrupt Enable CSR • PC2SR Interrupt Set Register • PC2SR ECC Log Register	SR2PC interrupt/enable/set bits: • Tsi721 Interrupt CSR.INT_PC2SR • Tsi721 Interrupt Enable CSR.INT_PC2SR_EN • Tsi721 Interrupt Enable CSR.INT_PC2SR_SET	N/A	PCIe MAC bit: • PCI Control and Status Register.INTXD • PCI Miscellaneous 1 Register.INTRPIN • Legacy Interrupt Status Register.INTA
I2C interrupt/enable/status/set registers: • I2C Interrupt Enable Register • I2C Interrupt Set Register • I2C Interrupt Status Register	SR2PC interrupt/enable/set bits: • Tsi721 Interrupt CSR.INT_I2C • Tsi721 Interrupt Enable CSR.INT_I2C_EN • Tsi721 Interrupt Enable CSR.INT_I2C_SET	N/A	PCIe MAC bit: • PCI Control and Status Register.INTXD • PCI Miscellaneous 1 Register.INTRPIN • Legacy Interrupt Status Register.INTA

Table 31: INTx Interrupt Hierarchy *(Continued)*

Interrupt Tree Leaf	Interrupt Tree Level 1	Interrupt Tree Level 2	Interrupt Tree Root
S-RIO MAC has internal interrupt hierarchy (see Figure 19)		SR2PC interrupt/enable/set bits: • Tsi721 Interrupt CSR.INT_SRIO • Tsi721 Interrupt Enable CSR.INT_SRIO_EN • Tsi721 Interrupt Enable CSR.INT_SRIO_SE	PCIe MAC bit: • PCI Control and Status Register.INTXD • PCI Miscellaneous 1 Register.INTRPIN • Legacy Interrupt Status Register.INTA

Figure 18: INTx Interrupt Hierarchy



For information on the S-RIO MAC INT subtree, see “S-RIO Event to Interrupt/Port-Write Hierarchy” in Interrupt Handling.

5.6 Interrupt Moderation

When the Interrupt Moderation Register has a non-zero value, it defines an interrupt moderation timer that moderates all Tsi721 interrupt sources.

For MSI-X/MSI, an interrupt event for a specific vector is sent to the PCIe root complex through MSI/MSI-X MWr only after the timer has expired. Since there are multiple MSI-X vectors, up to one MSI-X MWr per vector per timer period can be sent.

For INTx, an interrupt event is sent to the PCIe root complex through an INTx message only after the timer has expired.

5.7 PCIe Advisory Error Reporting

5.7.1 Physical Layer Errors

Table 32: Physical Layer AER

Error Condition	Action Taken	PCIe Specification Section
Link Errors (8b/10b, loss of symbol lock, elastic buffer overflow/underflow, lane-to-lane deskew)	Correctable error processing (receiver error)	4.2.4.6
Any TLP or DLLP framing rule violation.	Correctable error processing (receiver error)	4.2.2

5.7.2 Data Link Layer Errors

Table 33: Data Link Layer AER

Error Condition	Action Taken	PCIe Specification Section
Bad TLP[a]	TLP discarded, Correctable error processing (bad TLP)	3.5.3.1
Bad DLLP[b]	DLLP discarded, Correctable error processing (bad DLLP)	3.5.2.1
Replay timeout	Correctable error processing (replay timer timeout)	3.5.2.1
REPLAY NUM rollover	Correctable error processing (REPLAY_NUM rollover)	3.5.2.1
DL Protocol Error[c]	DLLP discarded, Uncorrectable error processing (data link protocol error)	3.5.2.1
Surprise link down	Uncorrectable error processing (surprise down error)	3.5.2.1 and 3.2.1

[a] A Bad TLP is a TLP ending in EDB with LCRC that does not match inverted calculated LCRC, or a TLP with incorrect LCRC, or a TLP received with sequence number not equal to NEXT_RCV_SEQ and this is not a duplicate TLP).

[b] A bad DLLP is a DLLP with a bad LCRC.

[c] A DL protocol error occurs when an ACK or NAK DLLP is received and the sequence number specified by AckNak_Seq does not correspond to an unacknowledged TLP or to the value in ACKD_SEQ.

5.7.3 Transaction Layer Errors

Table 34: Transaction Layer AER

Error Condition	Role Based (Advisory) Error Reporting Condition	Action Taken	PCIe Specification Section
Poisoned TLP received	Advisory when the corresponding error is configured as non-fatal in the AER Uncorrectable Error Severity Register	PCI Control and Status Register.DPE bit is set. If the poisoned TLP is a completion, PCI Control and Status Register.MDPED bit is set (if PCI Control and Status Register.PERRE is set) Advisory case: correctable error processing (advisory non-fatal error) Non-advisory case: uncorrectable error processing (poisoned TLP) Poisoned TLPs are discarded	2.7.2.2
ECRC check failure	N.A.	Uncorrectable error processing (ECRC error)	2.7.1
Unsupported request	Advisory when the corresponding error is configured as non-fatal in the AER Uncorrectable Error Severity Register and the request is non-posted	Non-advisory cases: uncorrectable error processing (unsupported request). Advisory cases: correctable error processing (advisory non-fatal error). For non-posted unsupported requests, the PCIE-EP generates a completion with UR status.	See Table 36
Completion timeout	Advisory when the corresponding error is configured as non-fatal and the requester attempts recovery	Non-advisory cases: uncorrectable error processing (completion timeout) Advisory cases: correctable error processing (advisory non-fatal error)	2.8
Completer abort	Advisory when the corresponding error is configured as non-fatal in the AER Uncorrectable Error Severity Register and the request is non-posted	Non-advisory cases: uncorrectable error processing (completer abort). Advisory cases: correctable error processing (advisory non-fatal error). For non-posted unsupported requests, Tsi721 generates a completion with CA status.	See Table 37
Unexpected completion received	Advisory when the corresponding error is configured as non-fatal in the AER Uncorrectable Error Severity Register	Non-advisory cases: uncorrectable error processing (unexpected completion). Advisory cases: correctable error processing (advisory non-fatal error)	2.3.2
Receiver overflow	N/A	Uncorrectable error processing (receiver overflow) TLP header is not logged in AER. TLP is nullified.	2.6.1.2

Table 34: Transaction Layer AER *(Continued)*

Error Condition	Role Based (Advisory) Error Reporting Condition	Action Taken	PCIe Specification Section
Flow control protocol error	N/A	Not applicable (no flow control protocol error checks are performed)	2.6.1
Malformed TLP	N/A	Uncorrectable error processing (malformed) TLP is nullified.	See Malformed TLP
Internal Error (PCIe MAC ECC errors)	N/A	Correctable or uncorrectable internal error reported	6.2.9
Header Log Overflow	N/A	Correctable error processing (header log overflow).	6.2.4.2

- For Tsi721 ECC error processing, see Data Protection
- For DMA channel specific errors of block DMA, they cause DMA channel abort (see DMA Channel Abort on Error)
- For DMA channel specific errors of Messaging Engine, they cause DMA channel abort (see outbound messaging DMA Channel Abort on Error and inbound messaging DMA Channel Abort on Error)

5.7.3.1 Malformed TLP

Table 35: Malformed Error Check

TLP Type	Error Check
All	TLP must have a valid FMT/TYPE combination Data payload length <= Max_Payload_Size (that is, MPS field in PCIe Device Control and Status Register)
All TLPs with data (that is, FMT[1]=1)	LENGTH field must match actual payload data
All TLPs with ECRC (that is, TD=1)	Actual TLP length must match calculated length (HEADER + PAYLOAD + ECRC)

I/O read or write request	LENGTH = 1 (dword)
	TC = 0
	ATTR = 0
	Last dword BE[3:0] = 0b0000

Configuration read or write request	LENGTH = 1 (dword)
	TC = 0
	ATTR = 0
	Last dword BE[3:0] = 0b0000

Table 35: Malformed Error Check *(Continued)*

TLP Type	Error Check
Message Requests Power management message Unlock message Set power limit message	TC = 0
TLPs with route to root complex routing	Should not be received by endpoints
TLPs with gathered and routed to root complex routing	Should not be received by endpoints
Interrupt messages (INTx) PME Turn Off Ack messages (PME_TO_Ack)	Should not be received by endpoints
Completion TC	TC = 0
Completion byte_count	Must match expected value of associated request
Completion lower_address	Must match expected value of associated request
Completion attr[1:0]	Must match expected value of associated request
Completion with CS of CSR (configuration request retry status)	Must match expected value of associated request

5.7.3.2 Unsupported Request

Table 36: Unsupported Request Error Check

Conditions Handled as UR	Description	PCIe Specification Section
Messages with invalid message code	Reception of a message TLP with invalid message code that targets the Endpoint	2.3.1
Function in D3Hot state	Any requests	5.3.1.4.1
A request misses all BARs (BAR0/1, BAR2/3, and BAR 4/5)	BAR misses are handled as UR	2.7.2.2
Poisoned type 0 configuration write requests	Type 0 configuration write requests must not be poisoned	2.7.2.2
A TLP with header Fmt/ Type other than MRd/ MWr/ Cpl/ CplD/ CfgWr0 /CfgRd0 /PME_TURN_OFF message/ PME_TO_ACK message /PM_Active_State_Nak message/Unlocked message/Vendor defined type 1 messages	Tsi721 only supports a subset of PCIe requests as an endpoint	2.7.2.2
A MWr/MRd TLP misses all outbound windows	Tsi721 supports 8 outbound windows	2.7.2.2

Table 36: Unsupported Request Error Check *(Continued)*

Conditions Handled as UR	Description	PCIe Specification Section
A request TLP with header AT of not 00 or TH of not 0	Tsi721 supports only a subset of PCIe header format	2.7.2.2
A request TLP with header Fmt of 100b	Tsi721 supports only a subset of PCIe header format	2.7.2.2
A request TLP with associated S-RIO response errors other than S-RIO response timeout	If a TLP hitting BAR2/3 or BAR4/5 is mapped into one or more S-RIO requests and any responses to the associated request have S-RIO errors detected (other than response timeout), Tsi721 handles the TLP as UR.	2.7.2.2

- Received vendor defined type 1 messages are silently discarded (the TLP is acknowledged, discarded, and flow control credits are returned)
- Received unlocked messages are silently discarded (the TLP is acknowledged, discarded, and flow control credits are returned)

5.7.3.3 Completer Abort

Table 37: Completer Abort Error Check

TLP Type	Description
A MRd TLP hitting BAR 1	BAR 1 is for MWr-> S-RIO doorbells
A non-MSI-X register MRd/MWr TLP with length/ 1st dword BE/ last dword BE not equivalent to dword aligned 4 consecutive bytes	Tsi721 limits these register accesses to 32 bits
A MSI-X register MRd/MWr TLP with length/ 1st dword BE/ last dword BE not equivalent to dword aligned 4 consecutive bytes or QW aligned 8 consecutive bytes	Tsi721 limits these register accesses to 32 or 64 bits
A MWr/MRd TLP mapped to S-RIO maintenance write/read with length/ 1st dword BE/ last dword BE not equivalent to dword aligned 4 consecutive bytes	S-RIO maintenance requests have size and address alignment requirements
A MWr TLP hitting BAR 1 with length/1st dword BE/ last dword BE not equivalent to 2 LSB bytes (doorbell’s 16 info bits)	S-RIO doorbells have only 16 info bits

5.7.3.4 Transaction Layer Error Pollution

Table 39: Transaction Layer Error Pollution

Error	Priority
Receiver Overflow	6 (highest)
Internal Error	5
ECRC Check failure	4
Malformed TLP received	3
Unsupported Request	2
Unexpected Completion received	
Completer Abort	
Poisoned TLP received	1 (lowest)

5.7.4 AER Emulation

- The PCIE MAC can emulate error occurrence in the AER Uncorrectable Error Status Register and AER Correctable Error Status Register.
- Two error emulation registers, Uncorrectable Error Emulation Register and Correctable Error Emulation Register, located in the configuration space of the PCIe MAC, allow emulation of errors in the PCIe MAC.
- When a bit in these registers is set, it causes the hardware to emulate the detection of the corresponding error. The detection of the error is handled as displayed in Figure 6-2 of the PCIe 2.1 base specification. The corresponding error is logged in the AER status registers (that is, AER Uncorrectable Error Status Register or AER Correctable Error Status Register), and reported to the root complex.
  - To allow emulation of advisory errors, the Uncorrectable Error Emulation Register contains a bit named ADVISORYNF. When this bit is set in conjunction with another bit in the Uncorrectable Error Emulation Register, the hardware flags the error as an advisory error and handles it according to Figure 6-2 of the PCIe 2.1 base specification.
- Since the error emulation does not involve an actual TLP, the AER Header Log 1st Doubleword Register(s)[1:4]dword in the PCIe MAC have RES type, such that they can be modified by software to emulate the capturing of the TLP's header.
- The following are some usage guidelines and limitations associated with error emulation:
  - To emulate the detection of a correctable error:
    - The desired error bit must be set in the Correctable Error Emulation Register.
  - To emulate the detection of an uncorrectable fatal error:
    - The desired error bit must be set in the Uncorrectable Error Emulation Register.
    - The severity of the error must be set to fatal in the AER Uncorrectable Error Severity Register.
  - To emulate the detection of an advisory uncorrectable non-fatal error:
    - The desired error bit must be set in the Uncorrectable Error Emulation Register. The error bit selected must qualify for advisory handling as specified in the PCIe 2.1 specification; otherwise, the operation of the emulation logic is undefined.
    - The ADVISORYNF bit must be set in the Uncorrectable Error Emulation Register.
    - The severity of the error must be set to non-fatal in the AER Uncorrectable Error Severity Register.

— Due to a limitation in the hardware, it is not possible to emulate the detection of a non-advisory uncorrectable non-fatal error.

5.8 SMBus Alert Response Hierarchy

The Tsi721 supports SMBus alert response with some of its internal interrupt events.

Table 40: SMBus Alert Response Hierarchy

Interrupt Tree Leaf	Interrupt Tree Level 1	Interrupt Tree Level 2	SMBus Alert Response Root
BDMA non-channelized interrupt/enable/status/set bits: • BDMA Interrupt CSR • BDMA Interrupt Enable CSR • BDMA Interrupt Set Register • BDMA ECC Log Register	SR2PC interrupt/enable/set bits: • Tsi721 Interrupt CSR.INT_BDMA_NONCH • Tsi721 Interrupt Enable CSR.INT_BDMA_NONCH_EN • Tsi721 Interrupt Enable CSR.INT_BDMA_NONCH_SET	I2C bits: • Externally Visible I2C Status Register.BDMA_NONCH • Externally Visible I2C Enable Register.BDMA_NONCH	I2C bits: • I2C Slave Configuration Register.ALRT_EN • Externally Visible I2C Slave Access Status Register.ALERT_FLAG
SMSG non-channelized interrupt/enable/status/set registers: • Messaging Engine Interrupt Register • Messaging Engine Interrupt Enable Register • Messaging Engine Interrupt Set Register • Messaging Engine non-channelized ECC LOG Register	SR2PC interrupt/enable/set bits: • Tsi721 Interrupt CSR.INT_SMSG_NONCH • Tsi721 Interrupt Enable CSR.INT_SMSG_NONCH_EN • Tsi721 Interrupt Enable CSR.INT_SMSG_NONCH_SET	I2C bits: • Externally Visible I2C Status Register.SMSG_NONCH • Externally Visible I2C Enable Register.SMSG_NONCH	I2C bits: • I2C Slave Configuration Register.ALRT_EN • Externally Visible I2C Slave Access Status Register.ALERT_FLAG
SR2PC non-channelized interrupt/enable/status/set registers: • SR2PC General Interrupt CSR • SR2PC General Interrupt Enable CSR • SR2PC Channel Interrupt Set Register {0..7} • SR2PC Correctable ECC Log Register	SR2PC interrupt/enable/set bits: • Tsi721 Interrupt CSR.INT_SR2PC_NONCH • Tsi721 Interrupt Enable CSR.INT_SR2PC_NONCH_EN • Tsi721 Interrupt Enable CSR.INT_SR2PC_NONCH_SET	I2C bits: • Externally Visible I2C Status Register.SR2PC_NONCH • Externally Visible I2C Enable Register.SR2PC_NONCH	I2C bits: • I2C Slave Configuration Register.ALRT_EN • Externally Visible I2C Slave Access Status Register.ALERT_FLAG

Table 40: SMBus Alert Response Hierarchy *(Continued)*

Interrupt Tree Leaf	Interrupt Tree Level 1	Interrupt Tree Level 2	SMBus Alert Response Root
PC2SR interrupt/enable/status/set registers: • PC2SR Interrupt CSR • PC2SR Interrupt Enable CSR • PC2SR Interrupt Set Register • PC2SR ECC Log Register	SR2PC interrupt/enable/set bits: • Tsi721 Interrupt CSR.INT_PC2SR • Tsi721 Interrupt Enable CSR.INT_PC2SR_EN • Tsi721 Interrupt Enable CSR.INT_PC2SR_SET	I2C bits: • Externally Visible I2C Status Register.PC2SR • Externally Visible I2C Enable Register.PC2SR	I2C bits: • I2C Slave Configuration Register.ALRT_EN • Externally Visible I2C Slave Access Status Register.ALERT_FLAG
I2C interrupt/enable/status/set registers: • I2C Interrupt Enable Register • I2C Interrupt Set Register • I2C Interrupt Status Register	I2C bits: • Externally Visible I2C Status Register.I2C • Externally Visible I2C Enable Register.I2C	N/A	I2C bits: • I2C Slave Configuration Register.ALRT_EN • Externally Visible I2C Slave Access Status Register.ALERT_FLAG
I2C status bits: • Externally Visible I2C Status Register — RESET — SW_STAT2 — SW_STAT1 — SW_STAT0 — OMBW — IMBR — RESET — SW_STAT2 — SW_STAT1 — SW_STAT0 — OMBW — IMBR	N/A	N/A	I2C bits: • I2C Slave Configuration Register.ALRT_EN • Externally Visible I2C Slave Access Status Register.ALERT_FLAG
S-RIO MAC has internal interrupt hierarchy (see Figure 19)	SR2PC interrupt/enable/set bits: • Tsi721 Interrupt CSR.INT_SRIO • Tsi721 Interrupt Enable CSR.INT_SRIO_EN • Tsi721 Interrupt Enable CSR.INT_SRIO_SET	I2C bits: • Externally Visible I2C Status Register.SRIO_MAC • Externally Visible I2C Enable Register.SRIO_MAC	I2C bits: • I2C Slave Configuration Register.ALRT_EN • Externally Visible I2C Slave Access Status Register.ALERT_FLAG

Table 40: SMBus Alert Response Hierarchy *(Continued)*

Interrupt Tree Leaf	Interrupt Tree Level 1	Interrupt Tree Level 2	SMBus Alert Response Root
Interrupt/enable bits: • SR2PC General Interrupt CSR.ECC_UNCORR • SR2PC General Interrupt Enable CSR.ECC_UNCORR_EN • PC2SR Interrupt CSR.ECC_UNCORR • PC2SR Interrupt Enable CSR.ECC_UNCORR_EN • BDMA Interrupt CSR.ECC_UNCORR • BDMA Interrupt Enable CSR.ECC_UNCORR_EN • BDMA Interrupt CSR.ECC_UNCORR_CH[7:0] • BDMA Interrupt Enable CSR.ECC_UNCORR_CH_EN[7:0] • Messaging Engine Interrupt Register.ECC_UNCORR • Messaging Engine Interrupt Enable Register.ECC_UNCORR_EN • Messaging Engine Interrupt Register.ECC_UNCORR_CH[7:0] • Messaging Engine Interrupt Enable Register.ECC_UNCORR_CH_EN[7:0] • RapidIO PBM Port Status Register — IG_TAG_FATAL — IG_TFL_FATAL — IG_DOH_FATAL — EG_DOH_FATAL — EG_DNFL_FATAL	I2C bits: • Externally Visible I2C Status Register.ECC_UNCORR • Externally Visible I2C Enable Register.ECC_UNCORR	N/A	I2C bits: • I2C Slave Configuration Register.ALRT_EN • Externally Visible I2C Slave Access Status Register.ALERT_FLAG

Table 40: SMBus Alert Response Hierarchy *(Continued)*

Interrupt Tree Leaf	Interrupt Tree Level 1	Interrupt Tree Level 2	SMBus Alert Response Root
Interrupt/enable bits: • SR2PC General Interrupt CSR.DL_DOWN • SR2PC General Interrupt Enable CSR.DL_DOWN_EN	I2C bits: • Externally Visible I2C Status Register.DL_DOWN • Externally Visible I2C Enable Register.DL_DOWN_EN	N/A	I2C bits: • I2C Slave Configuration Register.ALRT_EN • Externally Visible I2C Slave Access Status Register.ALERT_FLAG

RapidIO Interface

6. S-RIO Interface

Topics discussed include the following:

- Features
- Physical Layer
- Transport Layer
- S-RIO Egress Packet Buffer (PBMe)
- S-RIO Ingress Packet Buffer (PBMi)
- Local Logical Layer Management

6.1 Features

- Complete S-RIO Gen2 x4 Endpoint
  - Physical and transport layers
  - SerDes
  - All required RAMs
- Compliant with the *RapidIO Specification (Rev. 2.1)*
- Supports 8 S-RIO flows
- Supports 31 outstanding unacknowledged packets
- Supports multicast event control symbol (MECS) generation and termination
- Supports 16 destination deviceID filters
- Supports VC0 only
- Supports automatic link width negotiation
- Programmable lane reversal
- Programmable lane polarity inversion
- Supports receiver-based flow control
- Supports redundant lanes; if a lane fails, the port continues functioning by using the remaining lanes
- Supports hot plug
- Generates maintenance port-writes with up to a 16-byte payload
- Terminates received S-RIO maintenance request transactions
  - Supports only maintenance register read/write requests with 32-bit payload
  - Generates responses for received maintenance register read/write requests
  - Captures one maintenance port-write request with up to 16-byte payload

6.1.1 Unsupported Optional S-RIO Features

The Tsi721 does not support the following optional features of the *RapidIO Specification (Rev. 2.1)*:

- RapidIO Part 6: LP-Serial Physical Layer Specification - The following optional features are not supported
    - Virtual Channels (VC1–VC8)
    - Transmitter-Controlled Flow Control
    - Baud Rate discovery/negotiation
    - Auto-detection of IDLE1/IDLE2
- RapidIO Part 5: Globally Shared Memory Logical Specification
- RapidIO Part 9: Flow Control Logical Layer Extensions Specification
- RapidIO Part 10: Data Streaming Logical Specification
- RapidIO Part 12: Virtual Output Queueing Extensions Specification

6.2 Physical Layer

The Tsi721 is compliant with the *RapidIO Specification (Rev. 2.1)*, physical layer specification. In addition, the Tsi721 provides the following status registers:

- RapidIO PLM Port Received MECS Status Register

6.2.1 Changing S-RIO Port Link Rate

To change S-RIO port link rate, software must reset the Tsi721 (see Reset With Potential Timeout).

6.2.2 Bit Error Rate Testing (BERT)

Tsi721 supports unidirectional and loopback BER testing. The test mode is selected using the UNI bit in the RapidIO PRBS Lane {0..3} Control Register. A unidirectional operation requires one of the following conditions:

- A compatible PRBS generator is enabled in the link partner and the Tsi721 tester checks the received data
- The Tsi721 PRBS generator is used to source data that is checked at a compatible tester in the link partner

In unidirectional mode (also referred to as 10-bit PRBS), the PRBS is sent directly on the link and the 8b/10b encoder is bypassed. The received serial bit stream is checked against the expected PRBS and the 8b/10b decoder is bypassed. When used for BER testing, this data may have a poor bit-transition density – that is, a long run of consecutive zeros or consecutive ones – which may result in incorrect BER measurements.

In loopback testing mode (also referred to as 8-bit PRBS), the Tsi721 PRBS generator and tester are enabled simultaneously and the serial data is looped back by either the link partner or the S-RIO PMA Loopback. The PMA loopback allows the Tsi721 BERT operation to be verified independent of the link partner.

In loopback testing mode, the PRBS is 8b/10b encoded in the same way as S-RIO data during normal operation. When first enabled, the generator transmits comma characters such that the receiver can achieve symbol alignment, and then begins sending the 8b/10b encoded PRBS characters. After every 5000 code-groups are transmitted, the PRBS is paused and the regular S-RIO clock compensation sequence is inserted (KRRR).

The use of 8-bit PRBS is required for accurate BER testing, although it is somewhat less random than the 10-bit PRBS traffic. This is because 8b/10b encoding will produce a subset of the full 10-bit space. Between 50 and 75% of the possible 10-bit combinations will never be produced by an 8b/10b encoder operating on 8-bit PRBS traffic generated in the 8-bit domain. The effect of this is to decrease the randomness of the resulting serial traffic when contrasting the 8b/10b encoded 8-bit PRBS with the alternative 10-bit PRBS. The quality of the randomization of the PRBS is not likely to be a significant problem for testing the BER in the SerDes when compared to the improvements that the 8-bit PRBS provides to the problematic 10-bit PRBS data. In fact, measuring the BER in the context of an 8-bit PRBS that is 8b/10b encoded more closely models normal operation in which all traffic is 8b/10b encoded on the link.

Single bit errors during 8-bit loopback tests may result in the detection of one or two 10-bit encoding errors. Depending on the location and number of bit errors, the PRBS checker may be unable to continue verifying the received stream.

BER testing is performed on a per-lane basis. Although the S-RIO link will not initialize, the initialization state machine must be forced to 1x mode by setting OVER_PWIDTH to 0b10 in the RapidIO Port Control CSR. This ensures that all four lanes operate independently.

The procedure for performing PRBS testing is as follows:

1. Set OVER_PWIDTH to 0b10 in the RapidIO Port Control CSR.
2. Enable the SerDes transmitters by setting the SerDes Tx Control Register as follows:
   - EN_L to 1
   - CM_EN, TX_EN, and DATA_EN to 1
   - INVERT as desired (set to 1 to invert the transmit serial data)
   - LOOPBK_EN as desired (set to 1 to enable S-RIO PMA loopback mode)
   - HALF_RATE to 1 if the S-RIO link speed is 1.25 Gbaud, and 0 for all other speeds
3. For unidirectional mode only (10-bit PRBS), disable the symbol aligner inside the S-RIO SerDes by setting the SerDes Rx Control Register as follows:
   - EN to 1
   - ALIGN_EN to 0
   - TERM_EN, DATA_EN, and PLL_EN to 1
   - INVERT as desired (set to 1 to invert the receive serial data)
4. Clear the RapidIO PRBS Lane {0..3} Error Count Register.
5. Set the PRBS seed in the RapidIO PRBS Lane {0..3} PRBS Seed Register.
6. Configure the PRBS by setting the RapidIO PRBS Lane {0..3} Control Register as follows:
   - PATTERN and UNI as desired
   - TRAIN and ENABLE to 1
   - TRANSMIT to 0
7. Start PRBS generation and training by setting TRANSMIT to 1 in the RapidIO PRBS Lane {0..3} Control Register.
8. Wait for training to complete.
9. Enable error checking by setting TRAIN to 0 in the RapidIO PRBS Lane {0..3} Control Register.
10. Wait for the desired number of bits to be transmitted/received.
11. Disable generation of PRBS traffic and freeze the error counter by setting TRANSMIT to 0 in the RapidIO PRBS Lane {0..3} Control Register. When PRBS is enabled on lanes other than lane 0, disable the other lanes before disabling lane 0.
12. Read the RapidIO PRBS Lane {0..3} Error Count Register.

6.3 Transport Layer

The Tsi721 is fully compliant with the *RapidIO Specification (Rev. 2.1)*, transport layer specification.

6.3.1 destID Filtering

For received S-RIO packets, the Tsi721 optionally performs destination ID (destID) filtering.

When destID filtering is enabled, the destID of a received S-RIO packet is first compared against RapidIO Base deviceID CSR. If a match is found then the packet is accepted by the Tsi721; otherwise, the destID is masked by the MATCH field of one of the 16 base routing registers, RapidIO TLM Base Routing Register Control Register {0..15}, and the masked result is compared with the PATTERN field of the base routing register. If the two are equal, the packet is considered a match to the base routing register. If a packet matches any one of the base routing registers, it is accepted by the Tsi721; otherwise, the packet is discarded, and the ILL_ID bit in the RapidIO Local Logical/Transport Layer Error Detect CSR is set if the register is not locked from a previous event.

To enable destID filtering:

1. Configure TGT_ID_DIS and MTC_TGT_ID_DIS in RapidIO TLM Port Control Register.
2. Set up the base routing registers in RapidIO TLM Base Routing Register Control Register {0..15}.

6.3.2 ftype Filtering

The Tsi721 can perform ftype filtering based on the ftype of inbound S-RIO packets. Packet types that are not accepted due to filtering are dropped, and the ILL_TYPE bit in the RapidIO Local Logical/Transport Layer Error Detect CSR is set if the register is not locked from a previous event.

ftype filtering is configured in the RapidIO TLM Port ftype Filter Control Register.

6.4 S-RIO Egress Packet Buffer (PBMe)

The S-RIO MAC contains an egress packet buffer (PBMe) that is implemented as a Circular Reorder Queue (CRQ) with 288 data nodes of 32 bytes each. Here, the following fixed watermarks enforce egress S-RIO ordering rules:

- When free data nodes <170, priority 0 S-RIO packets are not accepted into the CRQ
- When free data nodes <120, priority 1 S-RIO packets are not accepted into the CRQ
- When free data nodes <70, priority 2 S-RIO packets are not accepted into the CRQ

6.4.1 CRQ Reordering

Under normal conditions, the CRQ sends egress S-RIO packets based on arrival order regardless of packet priority. However, when a packet retry (PR-CS) or packet-not-accepted (PNA-CS) control symbol is received from the S-RIO link partner, the CRQ reorders packets using the policy programmed in the RapidIO PBM Port Control Register. When a reorder event occurs, the S-RIO MAC takes the following actions:

1. All CRQ entries are searched, in the current order of the CRQ, to find an entry with a higher priority than the retried packet or, optionally, the oldest entry with the highest priority.
2. If a CRQ entry was detected in step 1, that entry is promoted to the front of the CRQ for immediate transmission. If no entry is detected, the retried entry is used.
3. Steps 1 and 2 are repeated until the programmed policy returns CRQ operation to dequeueing in arrival order.

6.5 S-RIO Ingress Packet Buffer (PBMi)

The S-RIO MAC contains an ingress packet buffer (PBMi) that is implemented as a shared memory among multiple queues with 288 data nodes of 32 bytes each. In addition, the S-RIO MAC provides programmable watermarks through the RapidIO PBM Port Ingress Watermarks 0 Register to RapidIO PBM Port Ingress Watermarks 3 Register for S-RIO ingress flow control. When an ingress S-RIO packet is received with a specific priority, if the total number of free nodes for the ingress packet buffer is less than the associated watermark, then S-RIO MAC enforces its link partner to retry the packet by issuing packet retry control symbols.

Here, register field PRIOyCRF_WM (y = 0–3) within an ingress watermark register should be programmed with the same value as PRIOy_WM. Also, the minimum value for PRIOy_WM is (y+1)*9.

An example of ingress watermark programming is listed below:

- PRIO0_WM, priority 0 watermark of 225 (usually prio 0 is small NREAD with low traffic rate, and is given 63 nodes)
- PRIO1_WM, priority 1 watermark of 153 (usually prio 2 is assigned to NWWRITE/SWRITE/messages, and consists of majority of traffic; therefore, it is given a higher number of additional108 free nodes; prio 1 is assigned to NREAD response, and is given 72 additional free nodes)
- PRIO2_WM, priority 2 watermark of 45 (usually priority 3 is assigned to small response packets such as doorbell/ message responses, unless NREAD is given priority 2, which is unlikely; therefore, priority 3 is given only 36 additional nodes, for 4 MAX packets)
- PRIO3_WM, priority 3 watermark of 9

Another example of ingress watermark programming is listed below:

- PRIO0_WM, priority 0 watermark of 117 (allow all priorities to share 288–117 data nodes before triggering back-pressure; then gives each priority higher than 0 additional 36 free nodes)
- PRIO1_WM, priority 1 watermark of 81
- PRIO2_WM, priority 2 watermark of 45
- PRIO3_WM, priority 3 watermark of 9

6.5.1 ftype/ttype Filtering

The S-RIO MAC must be configured to support the following logical layer transaction types:

- NWRITE/SWRITE/NWRITE_R/NREAD
- Maintenance read and write, port-write
- Doorbells
- Messages
- Valid responses associated with the above requests

For S-RIO packets with ftype/ttype that do not fall into any of the above transaction types, the S-RIO MAC performs ftype/ttype filtering (RapidIO TLM Port ftype Filter Control Register should be programmed accordingly) and discards these packets.

6.6 Local Logical Layer Management

The LLM is responsible for local S-RIO logical error management within the S-RIO MAC. Additional S-RIO error detection is carried out in the SR2PC and SMSG modules.

The Tsi721 supports two S-RIO error reporting methods:

- To remote S-RIO host through generating S-RIO port-write packets (see S-RIO Event Management)
- To local PCIe root complex through PCIe INTx/MSI/MSI-X (see Interrupt Handling)

6.6.1 Maintenance Read/Write Request

The LLM receives and processes Maintenance Read/Write Requests with a valid payload of up to 4 bytes. This leads to packets that can be up to 24 bytes long, assuming a large system transport type (tt).

Table 41: Allowed Field Values for Maintenance Read/Write Request Packets

Field	Allowed Values
crf	Ignored on reception
pri	
tt	0b00 = For 8-bit deviceIDs 0b01 = For 16-bit deviceIDs
ttype	0b0000 = Specifies a maintenance read request 0b0001 = Specifies a maintenance write request
wdptr	Word pointer and data size are used in conjunction to identify valid payload data bytes (for allowed values, see Table 42).
rdsize/wrsize	
src TID	The packet's transaction ID. Any value is allowed.
hop count	This value must be appropriately set by the Requestor, and is ignored by a destination endpoint.
config_offset	Double-word address offset into local register bus for reads and writes (21 bits wide). Any bits above this are flagged as a local illegal transaction decode error.
data payload	Only write requests can carry a data payload, which is constrained to be 4 bytes long.

Table 42: Allowed Read and Write Size Definitions for Maintenance Read/Write Request Packets

wdptr	rdsize/wrsize	Number of Bytes[a] [b]	Byte Lanes[c]
0b0	0b0000	1	0b10000000
0b0	0b0001	1	0b01000000
0b0	0b0010	1	0b00100000
0b0	0b0011	1	0b00010000
0b1	0b0000	1	0b00001000
0b1	0b0001	1	0b00000100
0b1	0b0010	1	0b00000010
0b1	0b0011	1	0b00000001
0b0	0b0100	2	0b11000000
0b0	0b0101	3	0b11100000
0b0	0b0110	2	0b00110000

Table 42: Allowed Read and Write Size Definitions for Maintenance Read/Write Request Packets *(Continued)*

wdptr	rdsize/wrsize	Number of Bytes[a b]	Byte Lanes[c]
0b1	0b0100	2	0b00001100
0b1	0b0101	3	0b00000111
0b1	0b0110	2	0b00000011
0b0	0b1000	4	0b11110000
0b1	0b1000	4	0b00001111

a. The number of bytes is constrained to be less-than-or-equal-to 4, although the RapidIO standard allows for more.

b. All Maintenance Read Requests that are less than 4 bytes are promoted to 4-byte accesses (the equivalent of rdsize = 0b1000)

c. Note that the RapidIO standard uses big-endian notation. For more information about the byte and word alignment referenced here, see *RapidIO Specification (Rev. 2.1), Part 1: Input/Output Logical Specification.*

6.6.2 Maintenance Read/Write Response

The LLM generates a maintenance read/write response for a received maintenance read/write request. The fields of this packet type are constrained to the values in the following table.

Table 43: Allowed Field Values for Maintenance Read/Write Response Packets

Field	Allowed Values
crf	• Set to 0b111 on generation
pri	
tt	• Set to the same value of the related read/write request on generation
destID	• Set to the sourceID of the related read/write request on generation
srcID	• Set to the destID of the related read/write request on generation
ttype	• 0b0010 = Specifies a maintenance read response • 0b0011 = Specifies a maintenance write response
status	• 0b0000 = DONE – Requested transaction has completed successfully • 0b0111 = ERROR – Unrecoverable error detected
target TID	• Set to source TID of the related read/write request on generation
hop count	• Set to all ones on generation
rsvd	• Set to zero when forming a packet
data payload	• 8-byte data payload is included only for read responses. If the status is DONE, the 32 bits of read-response data is aligned according to the wdptr of the request and padded with zeros. If the status is ERROR, the 64-bit quantity, 0xDEAD_BEEF_DEAD_BEEF is used as the payload.

6.6.3 Maintenance Port-Write Request

The Tsi721 can generate maintenance port-write request packets. The fields of this packet type are specified in the following table.

Table 44: Allowed Field Values for Maintenance Port-Write Request Packets

Field	Allowed Values
crf	• Ignored by LLM on capture • Set to 0b111 on generation
prio	
tt	• Expect 0b00 = For 8-bit deviceID on capture • Expect 0b01 = For 16-bit deviceID on capture • Set to LRG_TRANS field of RapidIO Port-Write Target deviceID CSR on generation
destID	• Ignored by LLM on capture but can optionally be checked in the TLM • Set to PW_TGT_ID and (optionally) MSB_PW_ID of RapidIO Port-Write Target deviceID CSR on generation
srcID	• Ignored by LLM on capture • Set to BASE_ID and (optionally) LAR_BASE_ID of RapidIO Base deviceID CSR on generation
wdptr	• Captured in WDPTR in RapidIO Port-Write Reception Status CSR • Set to 0b1 on generation
wrsize	• Captured in WR_SIZE in RapidIO Port-Write Reception Status CSR • Set to 0b1011 on generation
hop count	• Set to all ones on generation
data payload	• Up to 256-byte payloads are supported, but only the first 16 bytes of the transaction are captured in the RapidIO Port-Write Reception Capture CSR {0..3} • Only 16-byte transactions are supported on generation

Table 45: Maintenance Port-Write Request 16-byte Data Payload

Byte Offset	Bit Select	Data Payload Content
0x0	0:31	RapidIO Component Tag CSR
0x4	0:31	RapidIO Port Error Detect CSR
0x8	0:17	RapidIO PLM Port Event Status Register
	18:20	Reserved
	21	RCS bit of RapidIO Event Management Port-Write Status Register
	22	Reserved
	23	LOCALOG bit of RapidIO Event Management Port-Write Status Register
	24:31	Port ID of the physical-layer event that triggered the port-write.
0xC	0:31	RapidIO Logical/Transport Layer Error Detect CSR

6.6.3.1 Port-Write Timeout

To guarantee reliable delivery of Tsi721 generated port-writes through the system, the LLM implements a port-write timeout timer to send a port-write repeatedly until S-RIO host software has cleared associated S-RIO events, upon which the timer is reset.

After sending a port-write request packet, the LLM starts the port-write timer with an expiry set to the PW_TIMER value in the RapidIO Port-Write Control Register. The timer is reset once all pending S-RIO events have been cleared by the S-RIO host. Specifically, the timer is reset under the following conditions:

1. For Physical Layer events, the host has cleared RapidIO Port Error and Status CSR.PORT_W_P.
2. For Logical/Transport Layer error events from SR2PC/ SMSG, the S-RIO host must first clear all active event sources listed in S-RIO Error Event Summary for SR2PC and SMSG, then clear either the RapidIO Logical/Transport Layer Error Detect CSR or RapidIO Logical/Transport Layer Error Enable CSR.
3. For link-request/reset-device events, the S-RIO host has cleared the RST_REQ bit in RapidIO Event Management Reset Request Port Status Register.
4. For local Logical/Transport Layer error requests from within S-RIO MAC, the S-RIO host must first clear associated active event sources listed in S-RIO MAC Event Detection and Notification Summary, then clear associated bits in the RapidIO Local Logical/Transport Layer Error Enable CSR.

For multiple pending events, the S-RIO host has cleared all of the events described above.

7. S-RIO Event Management

Topics discussed include the following:

- Overview
- Event Notification
- Event Handling
- Software-Assisted Error Recovery
- Hot Extraction and Insertion
- Event Simulation

7.1 Overview

The Tsi721 monitors S-RIO error events in the following modules:

- S-RIO MAC
- SR2PC
- SMSG

The device supports extensive standard and implementation-specific event detection and notification capabilities. These capabilities are integrated with the *RapidIO Specification (Rev. 2.1)* (Part 8: Error Management Extension Specification) to ease fault handling software development. The Tsi721 supports the following features:

- *RapidIO Specification (Rev. 2.1)* (Part 8: Error Management Extension Specification)
  - — Physical Layer Error “leaky bucket”
  - — Logical/Transport Layer Error detection and capture
- Maintenance port-write request generation and interrupt notification methods
- Hardware support of fault isolation to prevent error-induced congestion
- Hot swap for recovery from errors
- Event simulation to simplify software/system verification

7.2 Event Notification

Depending on how an event is enabled, notification can occur through one of the following:

- PCIe INTx/MSI/MSI-X interrupts (see Interrupt Handling)
- Generation of an S-RIO Maintenance Port-Write Request

7.2.1 Event Detection and Notification Summary

To reduce duplicated text, the key phrases in Table 46 describe common requirements for event enabling, interrupt notification enabling, and port-write notification enabling.

Table 46: Key Phrases for Event Enabling and Notification

Key Phrase	Definition
Leaky bucket enabled	Refers to the "Leaky Bucket" method for recoverable errors defined in the *RapidIO Specification (Rev. 2.1)*, Part 8 Error Management Extensions. The following registers must be configured for the leaky bucket method to detect an event: • RapidIO Port Error Rate CSR.ERR_RB sets the "leak rate" for the leaky bucket • RapidIO Port Error Rate CSR.ERR_RR sets the maximum amount the counter can exceed the following two thresholds: — RapidIO Port Error Rate Threshold CSR.ERR_RFT <> 0 sets the counter value at which an Error Rate Failed event is detected. No event is detected when this field is 0. — RapidIO Port Error Rate Threshold CSR.ERR_RDT <> 0 sets the counter value at which an Error Rate Degraded event is detected. No event is detected when this field is 0.
Leaky bucket interrupt	Refers to an interrupt notification. The following registers must be configured for leaky bucket interrupts: • RapidIO PLM Port Interrupt Enable Register.OUTPUT_FAIL = 1 asserts the port's interrupt when an Error Rate Failed event is detected. • RapidIO PLM Port Interrupt Enable Register.OUTPUT_DEGR = 1 asserts the port's interrupt when an Error Rate Degraded event is detected. • Tsi721 Interrupt Enable CSR.INT_SRIO_EN = 1 enables the assertion of the interrupt
Leaky bucket port-write	Refers to a port-write notification. The following registers must be configured for leaky bucket triggered port-writes: • RapidIO PLM Port Port-Write Enable Register.OUTPUT_FAIL = 1 generates a port-write when an Error Rate Failed event is detected. • RapidIO PLM Port Port-Write Enable Register.OUTPUT_DEGR = 1 generates a port-write an Error Rate Degraded event is detected.

Table 47 identifies the events that can be detected by the Tsi721, and the notification methods for each (see Table 48 for port-write events detected by the SR2PC and SMSG modules). The table is organized into the following sub-sections:

- RapidIO Defined Recoverable Link Errors
- RapidIO Defined Unrecoverable Physical Layer Errors
- Implementation-Specific Physical Layer Errors
- Implementation Specific Physical Layer Events
- Local Logical/Transport Layer Errors
- Device-Level Events

Table 47: S-RIO MAC Event Detection and Notification Summary

Event Name	Description	Event Enable	Interrupt Enable	Port-Write Enable
RapidIO Defined Recoverable Link Errors				
Control symbol CRC error	Received a control symbol that failed its CRC check	RapidIO Port Error Rate Enable CSR.CS_CRC_ERR_EN = 1 and Leaky Bucket Enabled	Leaky Bucket Interrupt	Leaky Bucket Port-Write
Control symbol illegal ackID error	Received a control symbol with an ackID value that is not in use	RapidIO Port Error Rate Enable CSR.CS_ILL_ID_EN = 1 and Leaky Bucket Enabled	Leaky Bucket Interrupt	Leaky Bucket Port-Write
Received PNA	Received a packet-not-accepted control symbol, indicating the link partner detected a transmission error	RapidIO Port Error Rate Enable CSR.CS_NOT_ACC_EN = 1 and Leaky Bucket Enabled	Leaky Bucket Interrupt	Leaky Bucket Port-Write
Packet with illegal ackID	Received a packet with an ackID value that is out-of-sequence	RapidIO Port Error Rate Enable CSR.PKT_ILL_ACKID_EN = 1 and Leaky Bucket Enabled	Leaky Bucket Interrupt	Leaky Bucket Port-Write
Packet with CRC error	Received a packet that failed its CRC check	RapidIO Port Control CSR.ERR_DIS = 0 and RapidIO Port Error Rate Enable CSR.PKT_CRC_ERR_EN = 1 and Leaky Bucket Enabled	Leaky Bucket Interrupt	Leaky Bucket Port-Write
Packet with illegal size	Received a packet that is longer than 276 bytes	RapidIO Port Error Rate Enable CSR.PKT_ILL_SIZE_EN = 1 and Leaky Bucket Enabled	Leaky Bucket Interrupt	Leaky Bucket Port-Write
Illegal character detected	Received a 10-bit code group that has no valid decode, or that is unused in the RapidIO protocol	RapidIO Port Error Rate Enable CSR.DELIN_ERR_EN = 1 and Leaky Bucket Enabled	Leaky Bucket Interrupt	Leaky Bucket Port-Write
IDLE1 error	A character that is not part of the IDLE1 sequence, or a column that is not all A, K, or R code-groups, has occurred after link initialization is complete	RapidIO Port Error Rate Enable CSR.DELIN_ERR_EN = 1 and Leaky Bucket Enabled	Leaky Bucket Interrupt	Leaky Bucket Port-Write
Descrambler loss of sync	Descrambler sync check has failed	RapidIO Port Error Rate Enable CSR.DSCRAM_LOS_EN = 1 and Leaky Bucket Enabled	Leaky Bucket Interrupt	Leaky Bucket Port-Write
Link-response illegal ackID	Link-response control symbol received with an ackID that is not in use	RapidIO Port Error Rate Enable CSR.LR_ACKID_ILL_EN = 1 and Leaky Bucket Enabled	Leaky Bucket Interrupt	Leaky Bucket Port-Write

Table 47: S-RIO MAC Event Detection and Notification Summary *(Continued)*

Event Name	Description	Event Enable	Interrupt Enable	Port-Write Enable
Physical Layer Protocol error	Received a control symbol that indicates the packet transfer method has failed: • Received unexpected link-response • Received restart-from-retry when not in input retry-stopped state • Received STOMP or EOP when no packet is being received • Received two link-request control symbols before a link-response is transmitted	RapidIO Port Error Rate Enable CSR.PROT_ERR_EN = 1 and Leaky Bucket Enabled	Leaky Bucket Interrupt	Leaky Bucket Port-Write
Delineation error	Received characters that indicate a transmission error occurred (for example, SerDes is not aligned). Examples include undecodable 10-bit code groups and characters that are illegal in the position they were received in. For example, a packet or control symbol is interrupted by an idle sequence.	RapidIO Port Error Rate Enable CSR.DELIN_ERR_EN = 1 and Leaky Bucket Enabled	Leaky Bucket Interrupt	Leaky Bucket Port-Write
Unexpected packet acknowledge	Received an unexpected packet-acknowledge (packet accept or packet-retry) control symbol	RapidIO Port Error Rate Enable CSR.CS_ACK_ILL_EN = 1 and Leaky Bucket Enabled	Leaky Bucket Interrupt	Leaky Bucket Port-Write
Link Timeout	Either a packet-acknowledge or a link-response was not received within the time limit	RapidIO Port Link Timeout Control CSR.TVAL <> 0 and RapidIO Port Error Rate Enable CSR.LINK_TO_EN = 1 and Leaky Bucket Enabled	Leaky Bucket Interrupt	Leaky Bucket Port-Write
Implementation specific	Packet Denial Threshold	Packet Denial Threshold is enabled (see Packet denial threshold) and RapidIO Port Error Rate Enable CSR.IMP_SPEC_EN = 1 and Leaky Bucket Enabled	Leaky Bucket Interrupt	Leaky Bucket Port-Write

This document is confidential and is subject to an NDA.

Table 47: S-RIO MAC Event Detection and Notification Summary *(Continued)*

Event Name	Description	Event Enable	Interrupt Enable	Port-Write Enable
RapidIO Defined Unrecoverable Physical Layer Errors				
Port Error	• Timed out four times in a row waiting for link-response • Four link-responses in a row had an invalid ackID Port failed to (re)initialize	Link-response failure is always detected. Dead link timer expiry is defined below.	RapidIO PLM Port Interrupt Enable Register.PORT_ERR	RapidIO PLM Port Port-Write Enable Register.PORT_ERR
Port Failed	Link reached the FAILED threshold for recoverable physical layer errors	Leaky Bucket Enabled, Port Failed Event enabled	RapidIO PLM Port Interrupt Enable Register.OUTPUT_FAIL	RapidIO PLM Port Port-Write Enable Register.OUTPUT_FAIL
Implementation-Specific Physical Layer Errors				
Packet denial threshold	Too many consecutive Packet Retry and/or Packet Not Accepted (PNA) acknowledgements received for a specific packet. Note: Received PNA is a RapidIO Standard Physical Layer Error defined above.	RapidIO PLM Port Packet Denial Control Register.CNT_RTY = 1 and/or RapidIO PLM Port Packet Denial Control Register.CNT_PNA = 1 and RapidIO PLM Port Packet Denial Control Register.DENIAL_THRESH > 0	RapidIO PLM Port Interrupt Enable Register.MAX_DENIAL and/or Leaky Bucket Interrupt	RapidIO PLM Port Port-Write Enable Register.MAX_DENIAL and/or Leaky Bucket Port-Write
Recoverable data/Packet management memory corruption	Detected a recoverable packet data/packet management memory corruption The following bits in the RapidIO PBM Port Status Register indicate recoverable data corruption: IG_DATA_COR IG_TAG_COR IG_TFL_COR IG_DOH_COR IG_DNFL_COR EG_DATA_COR EG_CRQ_COR EG_DOH_COR EG_DNFL_COR	ECC errors are always detected.	RapidIO PBM Port Interrupt Enable Register, at least one of IG_DATA_COR IG_TAG_COR IG_TFL_COR IG_DOH_COR IG_DNFL_COR EG_DATA_COR EG_CRQ_COR EG_DOH_COR EG_DNFL_COR	RapidIO PBM Port Port-Write Enable Register, at least one of IG_DATA_COR IG_TAG_COR IG_TFL_COR IG_DOH_COR IG_DNFL_COR EG_DATA_COR EG_CRQ_COR EG_DOH_COR EG_DNFL_COR

Table 47: S-RIO MAC Event Detection and Notification Summary *(Continued)*

Event Name	Description	Event Enable	Interrupt Enable	Port-Write Enable
Unrecoverable packet data corruption	The contents of a packet have been unrecoverably corrupted, as signified by RapidIO PBM Port Status Register. IG_DATA_UNCOR	ECC errors are always detected.	RapidIO PBM Port Interrupt Enable Register.IG_DATA_UNCOR = 1	RapidIO PBM Port Port-Write Enable Register.IG_DATA_UNCOR = 1
Unrecoverable memory management corruption	Detected an unrecoverable memory corruption in memory used to manage packets, as signified by the following bits in the RapidIO PBM Port Status Register: IG_TAG_FATAL IG_TFL_FATAL EG_CRQ_FATAL EG_DOH_FATAL EG_DNFL_FATAL	ECC errors are always detected.	Any one of the following bits in the RapidIO PBM Port Interrupt Enable Register is set to 1: IG_TAG_FATAL IG_TFL_FATAL EG_CRQ_FATAL EG_DOH_FATAL EG_DNFL_FATAL	Any one of the following bits in the RapidIO PBM Port Port-Write Enable Register is set to 1: IG_TAG_FATAL IG_TFL_FATAL EG_CRQ_FATAL EG_DOH_FATAL EG_DNFL_FATAL
Implementation Specific Physical Layer Events				
Packet for VC1-8	Received packet has the VC bit set in the physical layer header	Always detected.	RapidIO TLM Port Interrupt Enable Register.IG_BAD_VC = 1	RapidIO TLM Port Port-Write Enable Register.IG_BAD_VC = 1
Port degraded	Link has reached the DEGRADED threshold for recoverable physical layer errors	Leaky Bucket Enabled, Port Degraded Event enabled	RapidIO PLM Port Interrupt Enable Register.OUTPUT_DEGR	RapidIO PLM Port Port-Write Enable Register.OUTPUT_DEGR
Dead link timer	A link has RapidIO Port Error and Status CSR.PORT_UNIT status for the programmed period of time.	RapidIO PLM Port Implementation Specific Control Register.DLT_THRESH > 0	RapidIO PLM Port Interrupt Enable Register.DLT	RapidIO PLM Port Port-Write Enable Register.DLT
Link initialization occurred	Port has transitioned from RapidIO Port Error and Status CSR.PORT_UNIT to PORT_OK.	RapidIO Port Control CSR.PORT_LOCKOUT = 1	RapidIO PLM Port Interrupt Enable Register.LINK_INIT	RapidIO PLM Port Port-Write Enable Register.LINK_INIT

Table 47: S-RIO MAC Event Detection and Notification Summary *(Continued)*

Event Name	Description	Event Enable	Interrupt Enable	Port-Write Enable
MECS received (port-specific)	Port has received a MECS.	Always detected. For port-specific status, see RapidIO PLM Port Event Status Register.MECS and RapidIO PLM Port Received MECS Status Register.CMD_STAT.	MECS do not cause port-level interrupts to be raised. The interrupt is raised at the device-level - see MECS received (Device). For port-specific status, see RapidIO PLM Port Event Status Register.MECS and RapidIO PLM Port Received MECS Status Register.CMD_STAT.	
Local Logical/Transport Layer Errors				
Local illegal transaction decode	The following conditions are detected by the Tsi721 when the following is received: • Maintenance request (read or write but not port-write) with rdsize/wrsize outside those specified inTable 42 • Maintenance write request with a payload data length that is not equal to 8 bytes. • Maintenance read request with payload data • Register Bus Timeout • Maintenance request local address error	N/A This event generates a maintenance response with error status, but with no event detected and no information latched in the Tsi721.	N/A	N/A
Recoverable packet data corruption	Detected a recoverable packet data corruption. The following bit in the RapidIO Event Management Interrupt Status Register and RapidIO Event Management Port-Write Status Register indicate recoverable data corruption: IG_DATA_COR	Only detected when ECC is implemented.	RapidIO Event Management Interrupts Enable Register.IG_DATA_COR	RapidIO Event Management Port-Write Enable Register.IG_DATA_COR

Table 47: S-RIO MAC Event Detection and Notification Summary *(Continued)*

Event Name	Description	Event Enable	Interrupt Enable	Port-Write Enable
Unrecoverable packet data corruption	Detected an unrecoverable packet data corruption. The following bit in the RapidIO Event Management Interrupt Status Register and RapidIO Event Management Port-Write Status Register indicate unrecoverable data corruption: IG_DATA_UNCOR	Only detected when ECC is implemented.	RapidIO Event Management Interrupts Enable Register.IG_DATA_UNCOR	RapidIO Event Management Port-Write Enable Register.IG_DATA_UNCOR
Local illegal transaction target[a]	The following conditions are detected by the Tsi721: • Received a packet with a TT value of either 0b10 or 0b11 • Packet was discarded by TLMi (see destID Filtering)	RapidIO Local Logical/Transport Layer Error Enable CSR.ILL_ID_EN = 1	RapidIO Event Management Interrupts Enable Register.LOCALOG	RapidIO Event Management Port-Write Enable Register.LOCALOG
Local unsupported transaction	The following conditions are detected by the Tsi721: • Received a maintenance packet with ttype between 5 and 15 (inclusive).	RapidIO Local Logical/Transport Layer Error Enable CSR.ILL_TYPE_EN = 1	RapidIO Event Management Interrupts Enable Register.LOCALOG	RapidIO Event Management Port-Write Enable Register.LOCALOG
ftype filter discarded a packet	A packet was dropped based on ftype/ttype filtering. Packet was discarded by TLMi (see ftype filter discarded a packet)	RapidIO TLM Port ftype Filter Control Register must be configured.	RapidIO Local Logical/Transport Layer Error Enable CSR.ILL_TYPE	RapidIO Local Logical/Transport Layer Error Enable CSR.ILL_TYPE
Discard due to BRR destID filtering	A packet destID did not match the Base Routing Registers nor Base deviceID.	RapidIO Base deviceID CSR, RapidIO TLM Base Routing Register Control Register {0..15},and RapidIO TLM Port ftype Filter Control Register, must be configured, as well as RapidIO TLM Port Control Register.TGT_ID_DIS = 0 and RapidIO TLM Port Control Register.MTC_TGT_ID_DIS = 0	RapidIO TLM Port Interrupt Enable Register.IG_BRR_FILTER = 1	RapidIO TLM Port Port-Write Enable Register.IG_BRR_FILTER = 1

Table 47: S-RIO MAC Event Detection and Notification Summary *(Continued)*

Event Name	Description	Event Enable	Interrupt Enable	Port-Write Enable
Device-Level Events				
Port-reset request	Port has received four reset-device link-requests to be handled with an interrupt.	RapidIO PLM Port Implementation Specific Control Register.SELF_RST = 0 and PORT_SELF_RST = 0. For more information, refer to RapidIO PLM Port Implementation Specific Control Register.	RapidIO Event Management Reset Request Interrupt Enable Register.RST_INT_EN = 1 for at least one port	RapidIO Event Management Reset Request Port-Write Enable Register.RST_PW_EN = 1 for at least one port
MECS received (Device)	The device has received a Multicast Event Control Symbol (MECS).	Always detected.	RapidIO Event Management MECS Interrupt Enable Register.CMD_EN and RapidIO Event Management Interrupts Enable Register.MECS	N/A
Port-write received	A port-write has been received	Always enabled.	RapidIO Event Management Interrupts Enable Register.PW_RX	N/A

a. Undetected bit-error(s) that would be covered by the CRC-16 check can be a cause of the Illegal Transaction Target error, as the Illegal Transaction Target error may be captured before the CRC-16 has been checked (applies to packets longer than 32 bytes).

Table 48: S-RIO Error Event Summary for SR2PC and SMSG

Event Name	Description	Port-Write Enable
I/O error response	SR2PC received an NREAD/maintenance read/maintenance writ/NWRITE_R/doorbell error response	SR2PC Port-Write Enable CSR.ERR_RSP_EN
I/O illegal transaction decode	SR2PC received a packet with one of the following errors: • if NWRITE_R/NWRITE has reserved wdptr/ wrsize values	SR2PC Port-Write Enable CSR.ILL_DEC_EN
I/O illegal target	SR2PC received a request packet missing all inbound windows	SR2PC Port-Write Enable CSR.ILL_TARGET_EN
I/O response timeout	SR2PC has a response timeout	SR2PC Port-Write Enable CSR.RSP_TO_EN
I/O unsolicited response	SR2PC received an unexpected response	SR2PC Port-Write Enable CSR.UNS_RSP_EN
PCIe link down	PCIe MAC has detected that the PCIe link is not in data link active state	SR2PC Port-Write Enable CSR.DL_DOWN_EN
Excessive message retry	An outbound message DMA channel has retried message segments of a message for more than 63 times as configured by RETRY_THR field of Outbound DMA Channel Control Register {0..7}	Outbound DMA Channel Port-Write Enable Register {0..7}.ERROR_EN
Message response timeout	An outbound message DMA channel has a message response timeout	Outbound DMA Channel Port-Write Enable Register {0..7}.ERROR_EN
Message error response	An outbound message DMA channel has received a message error response	Outbound DMA Channel Port-Write Enable Register {0..7}.ERROR_EN
Message request timeout	An inbound message DMA channel has a message request timeout	Inbound DMA Channel Port-Write Enable Register {0..7}.SRTO_EN
Message unsolicited response	Outbound Messaging Engine has received an unexpected message response	SR2PC Port-Write Enable CSR.UNS_RSP_EN

7.2.1.1 SR2PC/SMSG S-RIO Error Management Capture Registers

For port-write generation due to events detected by the SR2PC or SMSG modules, the Tsi721 implements the following error capture registers:

- RapidIO Logical/Transport Layer High Address Capture CSR
- RapidIO Logical/Transport Layer Address Capture CSR
- RapidIO Logical/Transport Layer deviceID Capture CSR
- RapidIO Logical/Transport Layer Control Capture CSR

7.2.2 Hierarchy of Event Detection

Multiple errors may be detected within a packet or control symbol. Table 50 indicates the order in which errors are detected. If an error appears above another error, the top error is reported and all errors that occur later in the table are ignored. Errors are reported in order of reception completion. For example, a protocol error in an embedded control symbol would be reported and acted on before a CRC error in the packet that contains the embedded control symbol.

Table 50: Error Priority for Packet s and Control Symbols

Error Group	Justification
Loss of descrambler sync	Descrambler sync loss will corrupt both packets and control symbols, so it must be reported before any other error.
Invalid/Illegal character in packet/control symbol	Report decode errors, as these must result in CRC failure.
Packet/Control symbol CRC error	If a packet or control symbol has a CRC error, the contents are invalid. Any errors caused by interpreting invalid data would be spurious.
Physical Layer errors	Report Physical Layer errors before passing packets on to upper layers.
Transport Layer errors	Report Transport Layer errors before passing packets on to upper layers.
Logical Layer errors	Last layer for error detection.

The error events listed in Table 51 can occur simultaneously with the error conditions listed in Table 50. In this case, the event that is captured/reported is not deterministic.

Table 51: Events Occurring Asynchronous to Packet/Control Symbol Reception

Event	Notes
Timeout for link-response or packet-acknowledgement	Occurs asynchronous to packet/control symbol reception. Can occur simultaneously with loss of descrambler SYNC, invalid characters in control symbol, control symbol CRC error, or control symbol interpretation errors.
Implementation defect indications	Occurs completely asynchronous to packet/control symbol reception.
Maximum denial	Occurs nearly simultaneously with other errors.

7.2.3 S-RIO MAC Event Notification using Interrupts

All interrupts from the Tsi721 can be disabled through the RapidIO Event Management Device Interrupt Enable Register.

If four reset-device requests are received by the S-RIO port, and the RapidIO PLM Port Implementation Specific Control Register.SELF_RST field is 0 and PORT_SELF_RST = 0, the port sets the RapidIO PLM Port Event Status Register.RST_REQ bit. If the port is enabled in the RapidIO Event Management Reset Request Interrupt Enable Register.RST_INT_EN and Tsi721 Interrupt Enable CSR.INT_SRIO_EN is set, the reset request generates an interrupt. The reset-device request status for the S-RIO port is located in the RapidIO Event Management Reset Request Port Status Register. The reset status of the port is summarized in the RapidIO Event Management Interrupt Status Register.RCS bit. For more information on reset-device request handling, see RapidIO PLM Port Implementation Specific Control Register.SELF_RST and PORT_SELF_RST bits.

Reception of a port-write causes an interrupt only when the RapidIO Event Management Interrupts Enable Register.PW_RX bit is set.

7.2.3.1 MECS Interrupt Notification

MECS interrupt notification can be enabled at the device level, although the event is also detected at the port level. Note that MECS cannot cause a port-write transmission.

Interrupt notification of MECS reception at the device level is controlled by the following:

- RapidIO Event Management MECS Interrupt Enable Register.CMD_EN
- RapidIO Event Management Interrupts Enable Register.MECS

The status of all MECSs received by the device is located in the following:

- RapidIO Event Management MECS Status Register.CMD_STAT
- RapidIO Event Management Interrupt Status Register.MECS

If the port has an outstanding MECS request, it is identified in the RapidIO Event Management MECS Port Status Register; RapidIO PLM Port Received MECS Status Register also provides the MECS CMD received (both registers are for informational purposes only).

7.2.4 Event Notification using Maintenance Port-Write Request

The Tsi721 will generate a Maintenance Port-Write Request maintenance port-write request packet to notify the system host that an event has occurred (see packet format described in Table 44 and Table 45).

Port-writes are sent when the following conditions are true:

- An enabled event has been detected. It exists if one of the following occurs:
  - — An enabled event exists in the S-RIO port
  - — RapidIO Port Error and Status CSR.PORT_W_P bit is 1
  - — An enabled event exists in the RapidIO Logical/Transport Layer Error Enable CSR
  - — An enabled event exists in the RapidIO Local Logical/Transport Layer Error Detect CSR
- RapidIO Event Management Device Port-Write Enable Register.PW_EN bit is set to 1.

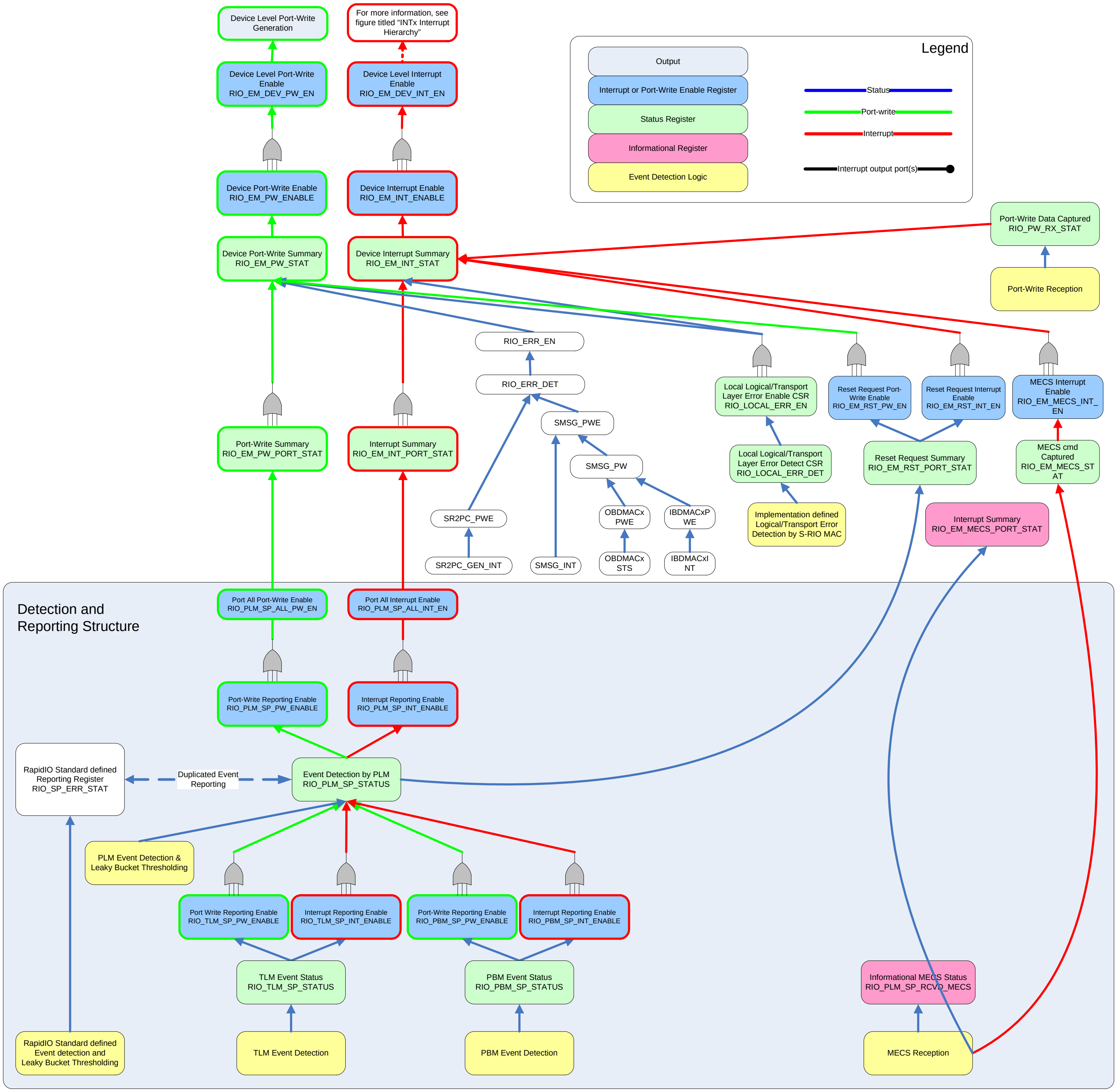
⚠ Port-writes are sent only when the RapidIO Event Management Device Port-Write Enable Register.PW_EN bit is set to 1.

The destination of a port-write is programmed in the RapidIO Port-Write Target deviceID CSR. Port-writes are routed as programmed in the RapidIO Port-Write Routing Register.

As explained in Port-Write Timeout, a port-write request is re-sent periodically in case they are not cleared by the destination. port-write transmission is controlled by the event notification hierarchy displayed in Figure 19, which is summarized in Table 47.

7.2.5 S-RIO Event to Interrupt/Port-Write Hierarchy

Figure 19: S-RIO Event to Interrupt/Port-Write Hierarchy


Device Level Port-Write Generation
For more information, see figure titled “INTx Interrupt Hierarchy”
Device Level Port-Write Enable RIO_EM_DEV_PW_EN
Device Level Interrupt Enable RIO_EM_DEV_INT_EN
Legend
Output
Interrupt or Port-Write Enable Register
Status Register
Informational Register
Event Detection Logic
Status
Port-write
Interrupt
Interrupt output port(s)
Device Port-Write Enable RIO_EM_PW_ENABLE
Device Interrupt Enable RIO_EM_INT_ENABLE
Port-Write Data Captured RIO_PW_RX_STAT
Device Port-Write Summary RIO_EM_PW_STAT
Device Interrupt Summary RIO_EM_INT_STAT
Port-Write Reception
RIO_ERR_EN
RIO_ERR_DET
SMSG_PWE
SMSG_PW
Local Logical/Transport Layer Error Enable CSR RIO_LOCAL_ERR_EN
Reset Request Port-Write Enable RIO_EM_RST_PW_EN
Reset Request Interrupt Enable RIO_EM_RST_INT_EN
MECS Interrupt Enable RIO_EM_MECS_INT_EN
Port-Write Summary RIO_EM_PW_PORT_STAT
Interrupt Summary RIO_EM_INT_PORT_STAT
Local Logical/Transport Layer Error Detect CSR RIO_LOCAL_ERR_DET
Reset Request Summary RIO_EM_RST_PORT_STAT
MECS cmd Captured RIO_EM_MECS_STAT
SR2PC_PWE
OBDMACx PWE
IBDMACxP WE
Implementation defined Logical/Transport Error Detection by S-RIO MAC
Interrupt Summary RIO_EM_MECS_PORT_STAT
SR2PC_GEN_INT
SMSG_INT
OBDMACx STS
IBDMACxI NT
Detection and Reporting Structure
Port All Port-Write Enable RIO_PLM_SP_ALL_PW_EN
Port All Interrupt Enable RIO_PLM_SP_ALL_INT_EN
Port-Write Reporting Enable RIO_PLM_SP_PW_ENABLE
Interrupt Reporting Enable RIO_PLM_SP_INT_ENABLE
RapidIO Standard defined Reporting Register RIO_SP_ERR_STAT
Duplicated Event Reporting
Event Detection by PLM RIO_PLM_SP_STATUS
PLM Event Detection & Leaky Bucket Thresholding
Port Write Reporting Enable RIO_TLM_SP_PW_ENABLE
Interrupt Reporting Enable RIO_TLM_SP_INT_ENABLE
Port-Write Reporting Enable RIO_PBM_SP_PW_ENABLE
Interrupt Reporting Enable RIO_PBM_SP_INT_ENABLE
TLM Event Status RIO_TLM_SP_STATUS
PBM Event Status RIO_PBM_SP_STATUS
Informational MECS Status RIO_PLM_SP_RCVD_MECS
RapidIO Standard defined Event detection and Leaky Bucket Thresholding
TLM Event Detection
PBM Event Detection
MECS Reception

The following table shows the registers used for S-RIO MAC events.

Table 52: S-RIO MAC Event to Interrupt/Port-Write Registers

Register Name	See
RIO_EM_DEV_PW_EN	RapidIO Event Management Device Port-Write Enable Register
RIO_EM_DEV_INT_EN	RapidIO Event Management Device Interrupt Enable Register
RIO_EM_PW_ENABLE	RapidIO Event Management Port-Write Enable Register
RIO_EM_INT_ENABLE	RapidIO Event Management Interrupts Enable Register
RIO_EM_PW_STAT	RapidIO Event Management Port-Write Status Register
RIO_EM_INT_STAT	RapidIO Event Management Interrupt Status Register
RIO_PW_RX_STAT	RapidIO Port-Write Reception Status CSR
RIO_EM_PW_PORT_STAT	RapidIO Event Management Port-Write Port Status Register
RIO_EM_INT_PORT_STAT	RapidIO Event Management Interrupt Port Status Register
RIO_LOCAL_ERR_DET	RapidIO Local Logical/Transport Layer Error Detect CSR
RIO_LOCAL_ERR_EN	RapidIO Local Logical/Transport Layer Error Enable CSR
RIO_EM_RST_PW_EN	RapidIO Event Management Reset Request Port-Write Enable Register
RIO_EM_RST_INT_EN	RapidIO Event Management Reset Request Interrupt Enable Register
RIO_EM_RST_PORT_STAT	RapidIO Event Management Reset Request Port Status Register
RIO_EM_MECS_PORT_STAT	RapidIO Event Management MECS Port Status Register
RIO_EM_MECS_INT_EN	RapidIO Event Management MECS Interrupt Enable Register
RIO_EM_MECS_STAT	RapidIO Event Management MECS Status Register
RIO_PLM_SP_ALL_PW_EN	RapidIO PLM Port All Port-Writes Enable Register
RIO_PLM_SP_ALL_INT_EN	RapidIO PLM Port All Interrupts Enable Register
RIO_PLM_SP_PW_ENABLE	RapidIO PLM Port Port-Write Enable Register
RIO_PLM_SP_INT_ENABLE	RapidIO PLM Port Interrupt Enable Register
RIO_PLM_SP_STATUS	RapidIO PLM Port Event Status Register
RIO_SP_ERR_STAT	RapidIO Port Error and Status CSR
RIO_TLM_SP_PW_ENABLE	RapidIO TLM Port Port-Write Enable Register
RIO_TLM_SP_INT_ENABLE	RapidIO TLM Port Interrupt Enable Register
RIO_TLM_SP_STATUS	RapidIO TLM Port Status Register
RIO_PBM_SP_PW_ENABLE	RapidIO PBM Port Port-Write Enable Register
RIO_PBM_SP_INT_ENABLE	RapidIO PBM Port Interrupt Enable Register

Table 52: S-RIO MAC Event to Interrupt/Port-Write Registers *(Continued)*

Register Name	See
RIO_PBM_SP_STATUS	RapidIO PBM Port Status Register
RIO_PLM_SP_RCVD_MECS	RapidIO PLM Port Received MECS Status Register

Table 53: SR2PC and SMSG Event to Port-Write Register Cross-Reference

Register Name	See
SR2PC_GEN_INT	SR2PC General Interrupt CSR
SR2PC_PWE	SR2PC Port-Write Enable CSR
RIO_ERR_DET	RapidIO Logical/Transport Layer Error Detect CSR
RIO_ERR_EN	RapidIO Logical/Transport Layer Error Enable CSR
SMSG_INT	Messaging Engine Interrupt Register
SMSG_PWE	Messaging Engine Port-Write Enable Register
SMSG_PW	Messaging Engine Port-Write Register
IBDMAC{0..7}INT	Inbound DMA Channel Interrupt Register {0..7}
IBDMAC{0..7}PWE	Inbound DMA Channel Port-Write Enable Register {0..7}
OBDMAC{0..7}STS	Outbound DMA Channel Status Register {0..7}
OBDMAC{0..7}PWE	Outbound DMA Channel Port-Write Enable Register {0..7}

7.3 Event Handling

In order to condense long descriptions for actions, some key phrases are used for event handling. These key phrases are defined in the following table.

Table 54: Key Phrases for Event Handling

Key Phrase	Definition
No user action required	This error condition is handled as part of the RapidIO standard error recovery method.
Physical Layer packet	Capture 16 bytes of the packet in the following: • RapidIO Port Packet/Control Symbol Error Capture CSR 0 • RapidIO Port Packet Error Capture CSR 1 • RapidIO Port Packet Error Capture CSR 2 • RapidIO Port Packet Error Capture CSR 3 For information on what controls which bytes are captured, see RapidIO PLM Port Implementation Specific Control Register.PAYL_CAP. RapidIO Port Attributes Capture CSR.VAL_CAPT = 1, and RapidIO Port Attributes Capture CSR.INFO_TYPE = 0.
Physical Layer control symbol	Capture a short control symbol in RapidIO Port Packet/Control Symbol Error Capture CSR 0. Capture a long control symbol in RapidIO Port Packet/Control Symbol Error Capture CSR 0 and RapidIO Port Packet Error Capture CSR 1. RapidIO Port Attributes Capture CSR.VAL_CAPT = 1, and RapidIO Port Attributes Capture CSR.INFO_TYPE = 2 (short control symbol) or 3 (long control symbol).
Clear the information latched.	Write 1 to the bit field in the “Information Latched” column in Table 55 to clear this error indication.
ackID realignment	This error may be caused by ackID misalignment between link partners. To fix this error, perform the ackID realignment procedure documented in ackID Synchronization.
Link partner reset	This error may be caused by a link partner failure. To fix this error, reset the link partner by following the procedure in Software-Assisted Error Recovery.
Port-reset	This error may be caused by a failure of the port. To fix this error, reset the port by following the procedure in S-RIO Resetor Software-Assisted Error Recovery.
Examine link and link partner for error conditions.	Check the following registers for error conditions that could prevent packet progress: • RapidIO Port Error Detect CSR • RapidIO Port Error and Status CSR • RapidIO PLM Port Event Status Register • RapidIO TLM Port Status Register • RapidIO PBM Port Status Register If possible, perform equivalent checks for the link partner.

Table 54: Key Phrases for Event Handling *(Continued)*

Key Phrase	Definition
Port failed and port degraded error handling	This refers to the "Leaky Bucket" method for recoverable errors defined in the *RapidIO Specification (Rev. 2.1)*, Part 8 Error Management Extensions, which determines when a Port Degraded or Port Failed event should be detected. This description assumes that Port Degraded and Port Failed events can be detected by the system, and that the Port Degraded threshold value is lower than the Port Failed threshold value. Port Degraded events indicate that the S-RIO port is still operational but has encountered a higher error rate than expected. Handling of a Port Degraded event requires the following steps to be completed in order to ensure that a subsequent Port Failed event can also be detected: • Write 0 to the RapidIO PLM Port Interrupt Enable Register.OUTPUT_DEGR/RapidIO PLM Port Port-Write Enable Register.OUTPUT_DEGR bits to disable notification of the Port Degraded event. • Write 0 to the RapidIO Port Error Detect CSR to clear detected errors. • Write 0 to the RapidIO Port Attributes Capture CSR to unlock all capture registers. • Write 1 to RapidIO Port Error and Status CSR.OUTPUT_ERR_ENCTR and/or INPUT_ERR_ENCTR to clear latched error recovery indications. • Write 1 to the RapidIO PLM Port Event Status Register.OUTPUT_DEGR. Port Failed indicates that the S-RIO port has encountered an error rate that is equivalent to failure. If the port that failed should continue to be used by the system, perform the following steps to clear a Port Failed event: • Write 0 to the RapidIO Port Error Rate CSR.ERR_RATE_CNT counter to stop detection of port failed and port degraded events. • Write 0 to the RapidIO Port Error Detect CSR to clear detected errors. • Write 0 to the RapidIO Port Attributes Capture CSR to unlock all capture registers. • Write 1 to RapidIO Port Error and Status CSR.OUTPUT_ERR_ENCTR and/or INPUT_ERR_ENCTR to clear latched error recovery indications. • Write 1 to the RapidIO PLM Port Event Status Register.OUTPUT_FAIL to clear the event.
Reset the port. Reconfigure the port. Realign ackIDs.	The operation of the port has been compromised by an internal error that can only be resolved by resetting the port. Reset of the port requires following the instructions for port-reset and ackID realignment, defined above.
Standard Logical/Transport Layer error information	Examine the standard Logical/Transport layer error information latched in the SR2PC module. Then clear the event(s) in the SR2PC module and clear RapidIO Event Management Port-Write Status Register.LOG by clearing the RapidIO Logical/Transport Layer Error Detect CSR.

Table 54: Key Phrases for Event Handling *(Continued)*

Key Phrase	Definition
Local Logical/Transport Layer error information	Examine the information latched in the following registers to determine the source of the implementation specific logical layer error: • RapidIO Local Logical/Transport Layer Error Detect CSR • RapidIO Local Logical/Transport Layer Error Enable CSR • RapidIO Local Logical/Transport Layer High Address Capture CSR • RapidIO Local Logical/Transport Layer Address Capture CSR • RapidIO Local Logical/Transport Layer deviceID Capture CSR • RapidIO Local Logical/Transport Layer Control Capture CSR Then clear the event(s) in the RapidIO Local Logical/Transport Layer Error Detect CSR and/or disable the corresponding event in the RapidIO Local Logical/Transport Layer Error Enable CSR to unlock the registers, as per the *RapidIO Specification (Rev. 2.1)*.
Port-write information	Use the information captured from the port-write to handle the event in the device that originated the port-write. Information captured from the port-write is located in the RapidIO Port-Write Reception Capture CSR {0..3}. To clear the event, write 1 to RapidIO Port-Write Reception Status CSR.PW_VAL and PW_DISC.

Table 55 summarizes the RapidIO hardware response to the events, the information latched for each event, as well as how to handle interrupt and port-write notifications for each event. Each event discussed is referred to by the abbreviated name defined in Table 47.

Table 55: Tsi721 S-RIO Event Handling Summary

Event Name	Hardware Reaction	Information Latched	Interrupt Handling	Port-Write Handling
RapidIO Defined Recoverable Link Errors				
Control symbol CRC error	Enter input error-stopped state and follow input error-stopped recovery process. No user action required	RapidIO Port Error Detect CSR.CS_CRC_ERR = 1 Physical Layer control symbol	Port Failed and Port Degraded Error Handling	Port Failed and Port Degraded Error Handling
Control symbol illegal ackID error	Enter output error-stopped state and follow output error-stopped recovery process. No user action required	RapidIO Port Error Detect CSR.CS_ILL_ID = 1 Physical Layer control symbol	Port Failed and Port Degraded Error Handling	Port Failed and Port Degraded Error Handling

Table 55: Tsi721 S-RIO Event Handling Summary *(Continued)*

Event Name	Hardware Reaction	Information Latched	Interrupt Handling	Port-Write Handling
Received PNA	Enter output error-stopped state and follow output error-stopped recovery process. No user action required	RapidIO Port Error Detect CSR.CS_NOT_ACC = 1 Physical Layer control symbol	Port Failed and Port Degraded Error Handling	Port Failed and Port Degraded Error Handling
Packet with illegal ackID	Enter input error-stopped state and follow input error-stopped recovery process. No user action required	RapidIO Port Error Detect CSR.PKT_ILL_ACKID Physical Layer packet	Port Failed and Port Degraded Error Handling	Port Failed and Port Degraded Error Handling
Packet with CRC error	Enter input error-stopped state and follow input error-stopped recovery process. No user action required	RapidIO Port Error Detect CSR.PKT_CRC_ERR Physical Layer packet	Port Failed and Port Degraded Error Handling	Port Failed and Port Degraded Error Handling
Packet with illegal size	Enter input error-stopped state and follow input error-stopped recovery process. No user action required	RapidIO Port Error Detect CSR.PKT_ILL_SIZE Physical Layer packet	Port Failed and Port Degraded Error Handling	Port Failed and Port Degraded Error Handling
Illegal character detected	Enter input error-stopped state and follow input error-stopped recovery process. No user action required	RapidIO Port Error Detect CSR.DELIN_ERR	Port Failed and Port Degraded Error Handling	Port Failed and Port Degraded Error Handling
IDLE1 error	Enter input error-stopped state and follow input error-stopped recovery process. No user action required	RapidIO Port Error Detect CSR.DELIN_ERR	Port Failed and Port Degraded Error Handling	Port Failed and Port Degraded Error Handling
Descrambler loss of sync	Enter input error-stopped state if not already there, send PNA with "loss of descrambler SYNC" as the cause and follow Input Error stopped recovery process. No user action required	RapidIO Port Error Detect CSR.DSCRAM_LOS	Port Failed and Port Degraded Error Handling	Port Failed and Port Degraded Error Handling

Table 55: Tsi721 S-RIO Event Handling Summary *(Continued)*

Event Name	Hardware Reaction	Information Latched	Interrupt Handling	Port-Write Handling
Link-response illegal ackID	Enter output error-stopped state and follow output error-stopped recovery process. No user action required	RapidIO Port Error Detect CSR.LR_ACKID_ILL Physical Layer control symbol	Port Failed and Port Degraded Error Handling	Port Failed and Port Degraded Error Handling
Physical Layer Protocol error	Enter output error-stopped state and follow output error-stopped recovery process for: • Received unexpected link-response Enter input error-stopped state and follow input error-stopped recovery process for: • Received restart-from-retry when not in input retry-stopped state • Received STOMP or EOP when no packet is being received • Received two link-request control symbols before a link-response is transmitted for the first • Packet or control symbol is interrupted by an idle sequence No user action required	RapidIO Port Error Detect CSR.PROT_ERR Physical Layer control symbol	Port Failed and Port Degraded Error Handling	Port Failed and Port Degraded Error Handling
Delineation error	Enter input error-stopped state and follow input error-stopped recovery process. No user action required	RapidIO Port Error Detect CSR.DELIN_ERR	Port Failed and Port Degraded Error Handling	Port Failed and Port Degraded Error Handling
Unexpected packet acknowledge	Enter output error-stopped state and follow output error-stopped recovery process. No user action required	RapidIO Port Error Detect CSR.CS_ACK_ILL Physical Layer control symbol	Port Failed and Port Degraded Error Handling	Port Failed and Port Degraded Error Handling
Link Timeout	Enter output error-stopped state and follow output error-stopped recovery process. No user action required	RapidIO Port Error Detect CSR.LINK_TO	Port Failed and Port Degraded Error Handling	Port Failed and Port Degraded Error Handling
Implementation specific	See Packet denial threshold	N/A	N/A	N/A

Table 55: Tsi721 S-RIO Event Handling Summary *(Continued)*

Event Name	Hardware Reaction	Information Latched	Interrupt Handling	Port-Write Handling
RapidIO Defined Unrecoverable Physical Layer Errors				
Port Error	After four failed link-request trials where either no response or a link-response with an ackID which is not outstanding was received, the port enters the Output Fatal Error Stopped state and no further attempt is made to recover the link.	RapidIO PLM Port Event Status Register.PORT_ERR, RapidIO Port Error and Status CSR.PORT_ERR	ackID realignment link partner reset port-reset Clear the information latched.	ackID realignment link partner reset port-reset Clear the information latched.
Port Failed	Reaction is controlled by the settings of RapidIO Port Control CSR.STOP_FAIL_EN and RapidIO Port Control CSR.DROP_EN, as described in Table 86.	RapidIO PLM Port Event Status Register.OUTPUT_FAIL, RapidIO Port Error and Status CSR.OUTPUT_FAIL and RapidIO Port Error and Status CSR.OUTPUT_DROP (if packets are dropped)	ackID Realignment Link Partner Reset port-reset Port Failed Error Handling	ackID Realignment Link Partner Reset port-reset Port Failed Error Handling
Implementation Specific Physical Layer Errors				
Packet denial threshold	No action.	RapidIO PLM Port Event Status Register.MAX_DENIAL	Examine link and link partner for error conditions. Clear the information latched.	Examine link and link partner for error conditions. Clear the information latched.

Table 55: Tsi721 S-RIO Event Handling Summary *(Continued)*

Event Name	Hardware Reaction	Information Latched	Interrupt Handling	Port-Write Handling
Recoverable data/Packet management memory corruption	No action.	RapidIO PBM Port Status Register bits: IG_DATA_COR IG_TAG_COR IG_TFL_COR IG_DOH_COR IG_DNFL_COR EG_DATA_CORR EG_CRQ_COR EG_DOH_COR EG_DNFL_COR	Clear the information latched.	Clear the information latched.
	Note: Under rare circumstances, IG_DATA_COR and EG_DATA_CORR events cannot be cleared immediately. The handling of these events should include checking that each event has been cleared and, if not, disabling IG_DATA_COR event reporting until a packet has been transferred from the S-RIO port inward, and disabling EG_DATA_CORR event reporting until a packet has been transferred from one of the internal blocks to the S-RIO port. The packet transfers clear the error conditions and allow normal error reporting to resume.			
Unrecoverable packet data corruption	On egress, the in-flight packet is marked as stomped and the packet is discarded. On ingress, the packet is marked as stomped if directed to the SR2PC and discarded if directed to the LLM.	RapidIO PBM Port Status Register bits: IG_DATA_UNCOR EG_DATA_UNCOR	Clear the information latched.	Clear the information latched.
Unrecoverable memory management corruption	Port discards packets received from the fabric. All packets received from RapidIO are either discarded or retried (see RapidIO PBM Port Control RegisterI.G_BACKPRESSURE_ON_FATAL.	RapidIO PBM Port Status Register bits: IG_TAG_UNCOR IG_TFL_UNCOR IG_DOH_UNCOR IG_DNFL_UNCOR EG_CRQ_UNCOR EG_DOH_UNCOR EG_DNFL_UNCOR	Reset the port. Reconfigure the port. Realign ackIDs.	Reset the port. Reconfigure the port. Realign ackIDs.
Implementation-Specific Physical Layer Events				
Packet for VC1-8	No action.	RapidIO TLM Port Status Register.IG_BAD_VC	Clear the information latched.	Clear the information latched.

Table 55: Tsi721 S-RIO Event Handling Summary *(Continued)*

Event Name	Hardware Reaction	Information Latched	Interrupt Handling	Port-Write Handling
Port degraded	No action	RapidIO PLM Port Event Status Register.OUTPUT_DEGR RapidIO Port Error and Status CSR.OUTPUT_DEGR	Port Failed and Port Degraded Error Handling	Port Failed and Port Degraded Error Handling
Dead link timer	Port drops all outbound packets dequeued from the PBMe. Packets received from RapidIO are forwarded.	RapidIO PLM Port Event Status Register.DLT None.	Clear the information latched.	Clear the information latched.
Link initialization occurred	No action	RapidIO PLM Port Event Status Register.LINK_INIT None.	Clear the information latched.	Clear the information latched.
MECS received (port-specific)	MECS event is sent to MECS signal	RapidIO PLM Port Event Status Register.MECS, and RapidIO PLM Port Received MECS Status Register Should be cleared only to avoid stale information.	N/A	N/A
S-RIO MAC Detected Logical/Transport Layer Errors				
Recoverable Packet Data Corruption	No action	RapidIO Event Management Interrupt Status Register.IG_DATA_COR and RapidIO Event Management Port-Write Status Register.IG_DATA_COR	Clear the information latched	Clear the information latched

Table 55: Tsi721 S-RIO Event Handling Summary *(Continued)*

Event Name	Hardware Reaction	Information Latched	Interrupt Handling	Port-Write Handling
Unrecoverable Packet Data Corruption	The packet received from the Endpoint Fabric Module (EFM) is discarded.	RapidIO Event Management Interrupt Status Register.IG_DATA_UNCOR and RapidIO Event Management Port-Write Status Register.IG_DATA_UNCOR	Clear the information latched	Clear the information latched
Local illegal transaction decode	The Maintenance request is aborted and an error response is sent back to the requestor.	N/A	N/A	N/A
Local illegal transaction target	Discard packet and capture error information.	RapidIO Local Logical/Transport Layer Error Detect CSR.ILL_ID	Local Logical/Transport Layer Error Information	Local Logical/Transport Layer Error Information
Local unsupported transaction	Discard packet and capture error information.	RapidIO Local Logical/Transport Layer Error Detect CSR.ILL_TYPE Implementation Specific Logical Layer Error Information	Clear the information latched.	Clear the information latched.
ftype filter discarded a packet	Discard packet and capture error information.	RapidIO Local Logical/Transport Layer Error Detect CSR.ILL_TYPE	Clear the information latched.	Clear the information latched.
Discard due to BRR destID filtering	Discard packet and capture error information.	RapidIO TLM Port Status Register.IG_BRR_FILTER	Clear the information latched.	Clear the information latched.

Table 55: Tsi721 S-RIO Event Handling Summary *(Continued)*

Event Name	Hardware Reaction	Information Latched	Interrupt Handling	Port-Write Handling
Device-Level Events				
Port-reset request	Reset the port, or ignore, depending on port setting.	RapidIO PLM Port Event Status Register.RST_REQ, and/or RapidIO Event Management Reset Request Port Status Register.RST_REQ	Clear RapidIO PLM Port Event Status Register.RST_REQ if required, then clear RapidIO Event Management Reset Request Port Status Register.RST_REQ	Clear RapidIO PLM Port Event Status Register.RST_REQ if required, then clear RapidIO Event Management Reset Request Port Status Register.RST_REQ
MECS received (Device)	MECS event is sent to MECS signal	RapidIO Event Management MECS Status Register and RapidIO Event Management MECS Port Status Register	Clear RapidIO Event Management MECS Status Register	N/A
Port-write received	Latch the first Port-Write received. Port-Writes received when PW_VAL is set are discarded, and cause the RapidIO Port-Write Reception Status CSR.PW_DISC bit to be set.	RapidIO Event Management Interrupts Enable Register.PW_RX = 1, RapidIO Port-Write Reception Status CSR, RapidIO Port-Write Reception Capture CSR {0..3}	Port-Write Information. Clear RapidIO Port-Write Reception Status CSR.PW_VAL and PW_DISC	N/A

Table 55: Tsi721 S-RIO Event Handling Summary *(Continued)*

Event Name	Hardware Reaction	Information Latched	Interrupt Handling	Port-Write Handling
Logical Errors Detected by SR2PC and SMSG				
I/O error response	• S-RIO header capture and port-write generation; also can contribute to PCIe INTx/MSI/MSI-X when enabled • If it is a response associated with a BDMA request, it causes the associated DMA channel to abort • If it is a bridging NREAD/maintenance read response, it is translated into PCIe cpl with UR completion status for associated MRd, and PCIe UR AER processing is performed • Otherwise, see additional processing discussed in Special PCIe-to-S-RIO Mapping Processing	-	-	-
I/O Illegal transaction decode	• S-RIO header capture and port-write generation (see Table 48); also can contribute to PCIe INTx/MSI/MSI-X when enabled • If it is a response, then the error is processed similar to I/O error response (see description of the previous row)	-	-	-
I/O illegal target	• S-RIO header capture and port-write generation; also can contribute to PCIe INTx/MSI/MSI-X when enabled	-	-	-
I/O response timeout	• Port-write generation but S-RIO header is not captured; also can contribute to PCIe INTx/MSI/MSI-X when enabled • If it is a response associated with a BDMA request, it causes the associated DMA channel to abort • If it is bridging maintenance write/NWRITE_R/doorbell response, see additional processing discussed in Special PCIe-to-S-RIO Mapping Processing	-	-	-
I/O unsolicited response	• S-RIO header capture and port-write generation; also can contribute to PCIe INTx/MSI/MSI-X when enabled	-	-	-
PCIe link down	• Port-write generation in addition to standard PCIe processing • No S-RIO header is captured	-	-	-

Table 55: Tsi721 S-RIO Event Handling Summary *(Continued)*

Event Name	Hardware Reaction	Information Latched	Interrupt Handling	Port-Write Handling
Excessive message retry	• S-RIO port-write generation; also can contribute to PCIe INTx/MSI/MSI-X when enabled • No S-RIO header is captured • causes associated outbound message DMA channel to abort	-	-	-
Message response timeout	• S-RIO port-write generation; also can contribute to PCIe INTx/MSI/MSI-X when enabled • No S-RIO header is captured • causes associated outbound message DMA channel to abort	-	-	-
Message error response	• S-RIO port-write generation; also can contribute to PCIe INTx/MSI/MSI-X when enabled • No S-RIO header is captured • causes associated outbound message DMA channel to abort	-	-	-
Message request timeout	• S-RIO port-write generation; also can contribute to PCIe INTx/MSI/MSI-X when enabled • No S-RIO header is captured	-	-	-
Message unsolicited response	• S-RIO port-write generation; also can contribute to PCIe INTx/MSI/MSI-X when enabled • No S-RIO header is captured	-	-	-

7.3.1 S-RIO MAC Interrupt Handling

S-RIO MAC interrupt handling follows the interrupt handling register tree in Figure 19. For software Interrupt Service Routine (ISR) processing, the first register read should be the RapidIO Event Management Interrupt Status Register. This register indicates what events have been detected.

To mask interrupts at the device level during interrupt handling, set the RapidIO Event Management Device Interrupt Enable Register.INT_EN bit to 0.

To enable interrupt assertion, set Tsi721 Interrupt Enable CSR.INT_SRIO_EN to 1.

If an RapidIO Event Management Interrupt Status Register.LOG event is indicated, see the Standard Logical/Transport Layer Error Information registers defined in Table 54 for information about the packet that caused the event, and the register that cleared the event.

If an RapidIO Event Management Interrupt Status Register.LOCALOG event is indicated, see the Local Logical/Transport Layer Error Information registers defined in Table 54 for information about the packet that caused the event, and the register that cleared the event.

If an RapidIO Event Management Interrupt Status Register.PW_RX event is indicated, see the port-write information registers defined in Table 54 for information about the port-write, and the register that cleared the event.

If an RapidIO Event Management Interrupt Status Register.RCS event is indicated, see the RapidIO Event Management Reset Request Port Status Register to determine if the port received a reset request. If the port received a reset request and did not perform a port self reset, write 1 to the RapidIO PLM Port Event Status Register.RST_REQ bit field to clear the port event. Next, write 1 to the RapidIO Event Management Reset Request Port Status Register bit for the port that received the reset.

RapidIO PLM Port Event Status Register.RST_REQ must be cleared before the RapidIO Event Management Reset Request Port Status Register.RST_REQ bit is cleared.

Both bits must be cleared if the port did not perform a port self reset.

If an RapidIO Event Management Interrupt Status Register.PORT event is indicated, read the RapidIO Event Management Interrupt Port Status Register to determine whether or not the port has events that require interrupt notification. If the port has an event then read the RapidIO PLM Port Event Status Register to determine which event occurred. To determine how to handle and clear each event, see Table 55.

If the RapidIO PLM Port Event Status Register.PBM_INT bit is set, read the RapidIO PBM Port Status Register to determine which events have occurred. To determine how to handle and clear each event, see Table 55.

If the RapidIO PLM Port Event Status Register.TLM_INT bit is set, read the RapidIO TLM Port Status Register to determine which events have occurred. To determine how to handle and clear each event, see Table 55.

7.3.2 MECS Interrupt Handling

If an RapidIO Event Management Interrupt Status Register.MECS event is indicated, see the RapidIO Event Management MECS Status Register to determine which MECS event has been received. To clear the MECS interrupt, clear the RapidIO Event Management MECS Status Register.

The RapidIO Event Management MECS Port Status Register indicates whether or not the S-RIO port received an MECS. To determine the MECS command value that was received, read the RapidIO PLM Port Received MECS Status Register then write 1 to clear them. These registers are provided for reference purposes only and do not need to be cleared in order to clear the MECS interrupt.

7.3.3 Handling of S-RIO MAC Captured Maintenance Port-Write Request

The *RapidIO Specification (Rev. 2.1)* allows a maintenance port-write request to be dropped if there are insufficient resources in the receiver. The Tsi721 provides enough buffer space to capture only one port-write. It also supports two methods for capturing port-writes:

- Continuous – Subsequent ingress port-write request transactions are dropped by hardware if a port-write is already captured.
- Reliable – Subsequent ingress port-write request transactions causes the S-RIO MAC to back-pressure the S-RIO ingress port if a port-write is already captured. Backpressure is released only after software clears PW_VAL in the RapidIO Port-Write Reception Status CSR.

In Reliable mode, maintenance request processing is gated by clearing the captured port-write.

A captured maintenance port-write request is removed from the port-write reception buffer by clearing RapidIO Port-Write Reception Status CSR.PW_VAL.

There are three methods for handling the port-write. The best method is to handle all events in the requesting device, as described in Handling All Events at the Requestor of a Port-Write, as this allows multiple events to be handled at the same time, and thus is the most efficient in terms of latency and bandwidth.

Another method is to check all logical layer events and all port-specific events that caused the port-write to be sent. The procedure for handling events within the port-write is described in Handling Only the Events Reported by a Maintenance Port-Write Request.

The least efficient method is to handle only the events indicated in the port-write (see Table 55).

IDT recommends that the first step of a port-write handler, regardless of the method chosen, is to mask port-write generation by clearing the RapidIO Event Management Device Port-Write Enable Register.PW_EN bit.

After the causes of port-write generation have been cleared, and the RapidIO Port Error and Status CSR.PORT_W_P bit has been cleared, then the RapidIO Event Management Device Port-Write Enable Register.PW_EN can be set to cause port-writes to be generated for pending events.

7.3.3.1 Handling Only the Events Reported by a Maintenance Port-Write Request

The host can chose to limit the scope of handling a captured maintenance port-write request to only those bits set in the Maintenance Port-Write Request 16-byte Data Payload. In this case, only the hierarchy that is directly implied by those bits need be accessed by software in the requesting device.

7.3.3.2 Handling All Events at the Requestor of a Port-Write

To handle all events that may be detected in the requestor of a maintenance port-write, the first register read should be the requestor's RapidIO Event Management Port-Write Status Register. This register indicates what events have been detected, which may include more than those flagged in the captured maintenance port-write request.

If an RapidIO Event Management Port-Write Status Register.LOG event is indicated, see the Standard Logical/Transport Layer Error Information registers defined in Table 54 for information about the packet that caused the event, and the register that cleared the event.

If a RapidIO Event Management Port-Write Status Register.LOCALOG event is indicated, see the Local Logical/Transport Layer Error Information registers defined in Table 54 for information about the packet that caused the event, and the register that cleared the event.

If a RapidIO Event Management Port-Write Status Register.RCS event is indicated, see the RapidIO Event Management Reset Request Port Status Register to determine if the S-RIO port received a reset request. If the port received a reset request and did not perform a port self reset, write 1 to the RapidIO PLM Port Event Status Register.RST_REQ bit field to clear the port event, and also write 1 to the RST_REQ bit in the RapidIO Event Management Reset Request Port Status Register.

RapidIO PLM Port Event Status Register.RST_REQ must be cleared before the RapidIO Event Management Reset Request Port Status Register bit for the port is cleared. Both bits must be cleared if the port is configured to not perform a port self reset.

If a RapidIO Event Management Port-Write Status Register.PORT event is indicated, read the RapidIO Event Management Port-Write Port Status Register to determine if the port has an event that requires port-write notification. To determine which event has occurred, read the RapidIO PLM Port Event Status Register. To determine how to handle and clear each event, see Table 55.

If the RapidIO PLM Port Event Status Register.PBM_PW bit is set, read the RapidIO PBM Port Status Register to determine which events have occurred. To determine how to handle and clear each event, see Table 55.

If the RapidIO PLM Port Event Status Register.TLM_PW bit is set, read the RapidIO TLM Port Status Register to determine which events have occurred.To determine how to handle and clear each event, see Table 55.

After all events have been cleared, clear the RapidIO Port Error and Status CSR.PORT_W_P bit.

7.4 Software-Assisted Error Recovery

The Tsi721 detects a fatal error when the ackID returned in a link-response control symbol does not match any outstanding ackID on four consecutive link-request/input-status control symbols. The device will also detect a fatal error if it does not receive a link-response to any of four consecutive link-request/input-status control symbols.

Recovery from this situation may be possible if there are no outstanding packets in either the Tsi721 or the link partner (for more information, see the algorithm for re-synchronizing ackIDs after a reset in ackID Synchronization). Otherwise, the link partner and the port must be reset to recover communication. For information on enabling and disabling the S-RIO port, see Disabling the S-RIO Port.

7.5 Hot Extraction and Insertion

When the Tsi721's RapidIO link partner is removed or reset, software must be informed of the event. The Dead Link Timer event (see DLT in RapidIO PLM Port Event Status Register) is used to inform software that the link partner has been removed or reset. When a Dead Link Timer event is active, RapidIO packets are discarded in the transmit direction. RapidIO packets can be received when the event is active after the link has retrained.

If the Dead Link Timer event is active, the Tsi721’s link partner should be configured to discard packets until software recovers the link. If the Tsi721's link partner does not discard packets, responses to requests issued by the Tsi721 may be received unexpectedly resulting in undefined operation.

The Link Initialized event informs software when the Tsi721's RapidIO link has successfully retrained. It may be necessary to resynchronize ackIDs after the link has retrained (for more information, see ackID Synchronization).

7.6 Event Simulation

To facilitate software verification, all RapidIO standard errors as well as the implementation specific events can be generated, as described in the following sections.

7.6.1 RapidIO-Defined Recoverable Link Errors

Individual recoverable link errors can be created using the following steps:

1. Write appropriate error capture information to the RapidIO Port Packet/Control Symbol Error Capture CSR 0, RapidIO Port Packet Error Capture CSR 1, RapidIO Port Packet Error Capture CSR 2, and RapidIO Port Packet Error Capture CSR 3.
2. Set the RapidIO Port Attributes Capture CSR.VAL_CAPT bit and IMPL_DEP bits as required.

To create a Port Degraded condition, follow the steps defined in above, and then write to the RapidIO Port Error Detect CSR as many times as required to cause a Port Degraded event.

7.6.2 RapidIO-Defined Unrecoverable Physical Layer Errors

A Port Error condition can be created by writing 1 to the RapidIO PLM Port Event Generate Register.PORT_ERR bit.

To create a Port Failed condition, follow the steps defined in RapidIO-Defined Recoverable Link Errors, and then write to the RapidIO Port Error Detect CSR as many times as required to cause a Port Failed event.

7.6.3 Implementation-Specific Physical Layer Errors

To create a Packet Denial Threshold event, write 1 to the RapidIO PLM Port Event Generate Register.MAX_DENIAL bit.

7.6.4 Implementation-Specific Physical Layer Events

Implementation specific physical layer events can all be created by writing to the associated bits in the RapidIO PLM Port Event Generate Register and RapidIO TLM Port Event Generate Register.

7.6.5 Device Events

LOG and LOCALOG events are created as described in Device Events. Port-write reception events are created by the following steps:

1. Set the S-RIO port into loopback mode (transmitter connected to receiver).
2. Configure port-writes to be routed out the port in loopback mode using the RapidIO Port-Write Routing Register.
3. Create an event that uses port-write notification. This will cause a port-write to be sent, which will also be received due to the loopback mode.

Bridging Modules

8. Mapping Engine

Topics discussed include the following:

8.1 Overview

The Mapping Engine connects with the PCIe Interface, S-RIO Interface, Block DMA Engine, and Messaging Engine. It performs PCIe transaction to S-RIO packet mapping, and vice versa. The mapping is used for direct PCIe to S-RIO bridging. In addition, it provides an S-RIO and PCIe path for the Messaging Engine and Block DMA Engine.

The Mapping Engine consists of two blocks (see Figure 1):

- PC2SR – Used for PCIe to S-RIO transactions
- SR2PC – Used for S-RIO to PCIe transactions

8.1.1 Features

8.1.1.1 PC2SR Features

The PC2SR block supports the following:

- 32 outstanding S-RIO NREAD/maintenance read requests
- 32 outstanding outbound doorbell requests, maintenance write requests, and NWRITE_R requests
- Four PCIe BARs (six BAR registers, BAR 0 to BAR 5)
  - BAR 0: 32-bit non-prefetchable memory used for Tsi721 internal memory map with a BAR size of at least 512 KB
  - BAR 1: 32-bit non-prefetchable memory with a BAR size of at least 16 MB for doorbells
    - Supports 8 doorbell channels
    - MWr to doorbells with S-RIO prio of 2 and configurable S-RIO crf
  - BAR 2/3: 32-bit BAR 2 or 64-bit BAR 2/BAR 3, prefetchable with a BAR size of at least 16 MB for PCIe to S-RIO bridging
  - BAR 4/5: 32-bit BAR 4 or 64-bit BAR 4/BAR 5, non-prefetchable with a BAR size of at least 16 MB for PCIe to S-RIO bridging
  - PCIe to S-RIO bridging BARs (BAR 4/5 and BAR 2/3) further
    - Includes 8 outbound address windows, where a window can be associated with either BAR 4/5 or BAR 2/3 according to their base addresses
    - 8 zones (sub windows) per window

    - MWr to NWRITE/ SWRITE with S-RIO prio of 2 and configurable S-RIO crf (selection between NWRITE and SWRITE is dynamic, and SWRITE is used as long as possible to generate an S-RIO packet with 8-byte aligned PCIe address and payload)
    - MWr to NWRITE_R with S-RIO prio of 2 and configurable S-RIO crf
    - MWr to maintenance write with S-RIO prio of 2 and configurable S-RIO crf, where MWr payload must consist of the 1st dword as only 4 bytes S-RIO maintenance write access is supported
    - MRd to NREAD with S-RIO prio of 0 and configurable S-RIO crf
    - MRd to maintenance read with S-RIO prio of 0 and configurable S-RIO crf, where MRd must request payload consist of the 1st dword as only 4 bytes S-RIO maintenance read access is supported
- Request segmentation
  - Supports MWr byte enables that cannot be mapped to S-RIO for a single NWRITE
  - Supports MRd byte enables that cannot be mapped to S-RIO for a single NREAD
- 8 KB of RESPONSE reassembly
  - Supports reassembly of multiple cplD from an original NREAD
    - Can be due to multiple MRd segmented from one NREAD by SR2PC
    - Can be due to partial cplD
  - Organized as 32 x 256-byte buffers

8.1.1.2 SR2PC Features

The SR2PC block supports the following:

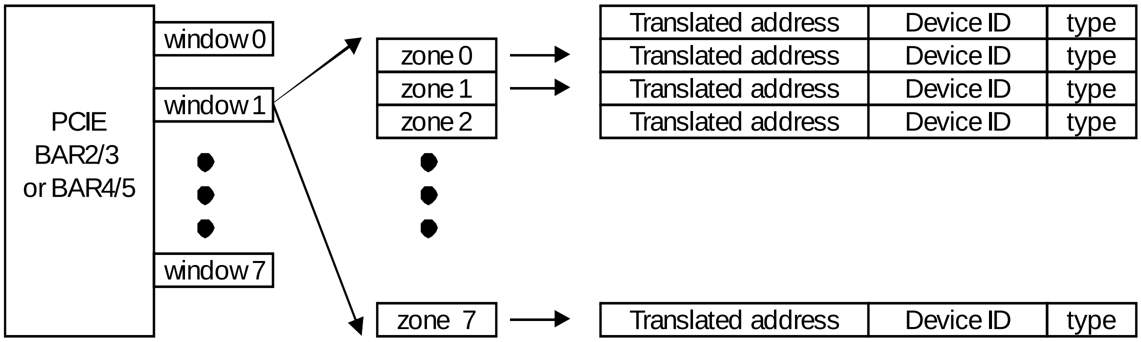
- 32 outstanding PCIe tags (PCIe MRd requests)
- 8 inbound windows for address translation
- Request segmentation
  - Required if PCIe maximum packet size is set to 128 bytes for NWRITE/SWRITE/NWRITE_R
  - Required if NWRITE/SWRITE/NWRITE_R payload represented by address+payload length crosses 4-KB boundary
  - Required if maximum read request size is set to 128 bytes for NREAD
  - Required if NREAD payload represented by address+rdsize+wdptr crosses 4-KB boundary
- 12 KB of completion reassembly buffer
  - Organized as 32 x 256 B + 4-KB buffers
  - 32, 256-byte buffers with one buffer per MRd with LENGTH less than or equal to 256 bytes
    - Multiple NREAD segmented from one MRd share one buffer
  - One 4-KB buffer shared by NREADs segmented from a MRd with LENGTH larger than 256 bytes
    - Subsequent MRd with LENGTH less than 257 bytes is not blocked until a second MRd with LENGTH larger than 256 bytes
    - A new MRd with LENGTH larger than 256 bytes is blocked until the 4-KB buffer has been released (blocked MRd will block follow on MRd)
  - Reassembly required to support multiple S-RIO responses from one MRd; that is, segmented multiple NREADs from a single MRd requesting up to 4-KB payload

8.2 PCIe Address Translation

The 32/64-bit address of a PCIe transaction is translated using four BARs implemented in the PCIe MAC. The PCIe MAC indicates which BAR a receive MRd/MWr TLP has hit as follows:

1. Addresses hitting BAR 0 are translated into a Tsi721 register access address.
2. Addresses hitting BAR 2/3 or BAR 4/5 are divided into 8 windows with 8 zones per window, where the next three PCIe address bits after a window's base address are used to select a zone (see Figure 20).

Figure 20: PC2SR Address Lookup



- Each window is defined by the following:
  - — Outbound Window Lower Base Register {0..7}
  - — Outbound Window Upper Base Register {0..7}
  - — Outbound Window Size Register {0..7}
- Windows must be non-overlapping; otherwise, the Tsi721 will experience undefined behavior
- 8 zones per window support multiple S-RIO target devices per window, with up to one target device per zone
- Supported window size is 2^N bytes, with N from 15 (32 KB) to 34 (16 GB)
- When a TLP hits a window, the $(N-1)^{th}$ to $(N-3)^{th}$ bits of its PCIe address – that is, its three MSB bits within window size – are used to select one of the 8 zones of a window
- The Tsi721 provides one software configurable window/zone lookup table with 64 entries, with 8 entries (zones) per window
- The zone lookup table, or LUT, is configured using the following registers:
  - — Outbound Window Lookup Table Zone Select Register
  - — Outbound Window Lookup Table Data 0 Register
  - — Outbound Window Lookup Table Data 1 Register
  - — Outbound Window Lookup Table Data 2 Register

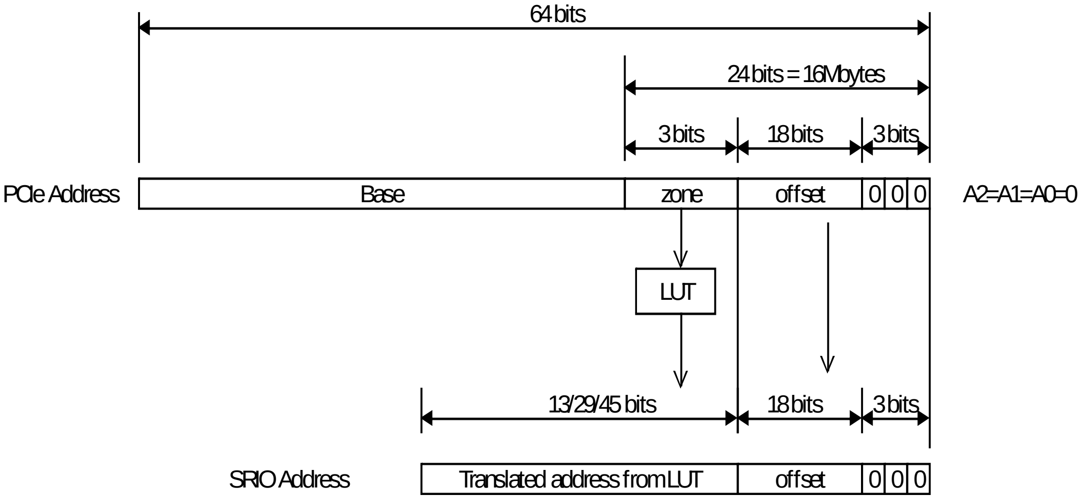
Upon power-up, all lookup table locations must be written by software prior to reading. Reading from uninitialized lookup table locations may result in uncorrectable ECC errors.

- The zone lookup table translates an incoming PCIe header into an outgoing S-RIO header. For an example, a MWr can be translated into an NWRITE/SWRITE, or NWRITE_R packet. This S-RIO ftype/transaction selection is through windows/zones.
- PCIe address to S-RIO destID translation is also completed through windows/zones
- If a MWr does not hit any window, it is discarded with AER processing
- If a MRd does not hit any window, a UR Cpl is sent with AER processing

- A MWr with an address hitting BAR 1 is translated into a doorbell
  - — The MWr's 16-bit payload is translated into the doorbell's 16 information bits, with MWr's payload byte 0/1 mapped into the most/least significant S-RIO info byte

Figure 21 shows an example of an MWr to SWRITE address translation through BAR 2/3. In this example, bit 0 is the least significant bit (LSB). The PCIe header address of the received MWr TLP is used to find a matching outbound window. In this example the matching window is programed with a size of 16 MB (24 bits). Zone selection requires 3 bits, and thus the offset is 18 bits. These offset bits are directly used as part of S-RIO address bits. Depending on whether 34/50/66-bit S-RIO addressing is used, the remaining 13/29/45 S-RIO address bits are from the address translation lookup table (MSB bits starting from bit 21). In this example, PCIe MWr A2/A1/A0 addressing bits are all zeroes as SWRITE is 8-byte aligned.

Figure 21: PCIe to S-RIO Address Translation for MWr to SWRITE Example



The Tsi721 implements a similar procedure for MWr to NWRITE/NWRITE_R and MRd to NREAD translation. In addition, since these MWr/MRd TLPs can have arbitrary address alignment, it performs segmentation and calculates S-RIO header fields WDPTR/WRSIZE according to the received PCIe TLP header.

Figure 22 shows an example of a PCIe MWr to S-RIO maintenance write translation through BAR 2/3. In this example a window is programed with a size of 1 MB (20 bits). Zone selection requires 3 bits, and thus the offset is 15 bits. These offset bits are directly used as part of S-RIO maintenance write address bits. Since maintenance write address always has 24 bits, the remaining 7 bits are translated from address translation lookup table (MSB bits starting from bit 17). In this example, PCIe MWr A1/A0 addressing bits are all zeroes. The Tsi721 calculates S-RIO header fields WDPTR/WRSIZE according to received PCIe TLP header.

Figure 22: PCIe to S-RIO Address Translation for MWr to Maintenance Write Example

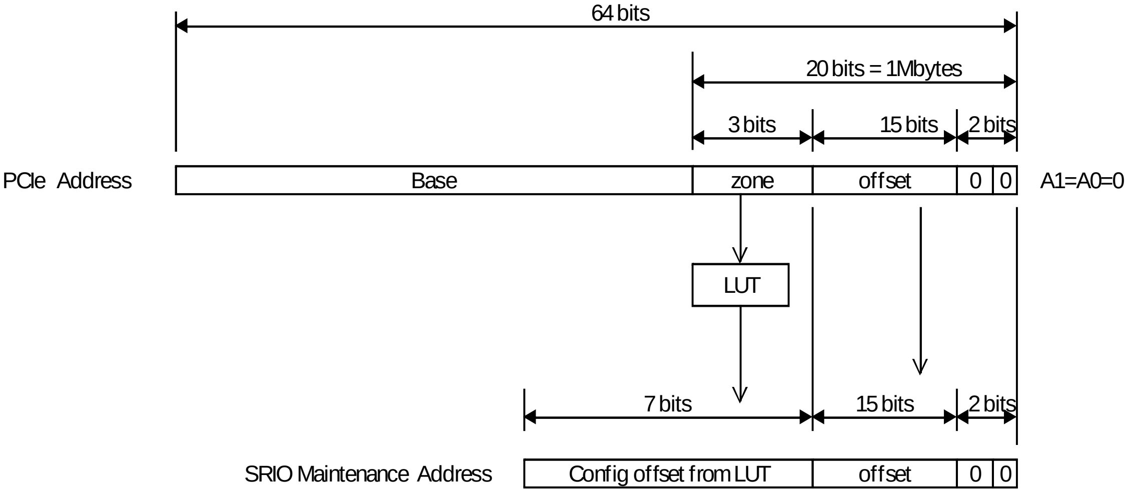


Figure 23 shows an example of a PCIe MWr to S-RIO doorbell translation through BAR 1. Bits 17:2 of MWr's address are used as the doorbell's deviceID, while bits 21:18 define a doorbell's channel number. Only channels 0–7 are supported. In this example, the 16 LSB bit MWr payload is translated into 16-bit doorbell information bits, with MWr payload byte 0 corresponding to S-RIO doorbell information byte 0 (doorbell information byte 0/bit 15 is MSB).

One application for a doorbell channel is to associate a doorbell with a block DMA/outbound messaging channel.

Figure 23: MWr to Doorbell Translation Example

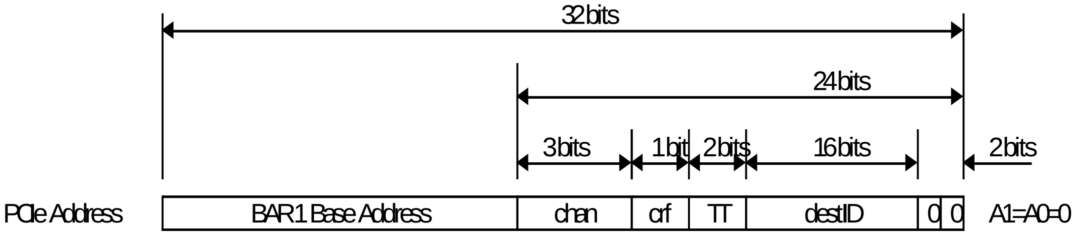


Figure 24: MWr to Doorbell Endian

PCIe Byte 0 mapped to S-RIO Doorbell Info Byte (MSB)

Byte 3	Byte 2	Byte 1	Byte 0
Unused	Unused	Info[7:0] (LSB)	Info[15:8] (MSB)

8.2.1 Updating Outbound Window Registers

To update outbound window registers (one of Outbound Window Lower Base Register {0..7}, Outbound Window Upper Base Register {0..7}, Outbound Window Size Register {0..7}, Outbound Window Lookup Table Zone Select Register, Outbound Window Lookup Table Data 0 Register, Outbound Window Lookup Table Data 1 Register, or Outbound Window Lookup Table Data 2 Register), software should ensure there are no PCIe TLPs buffered inside the Tsi721; otherwise, device behavior is undefined. The following procedure is recommended for updating outbound window registers:

1. Stop sending PCIe TLPs from any device to the Tsi721 other than as required for subsequent steps in this procedure.
2. Instruct each device (or wait for longest delay among all devices for TLPs to arrive at the Tsi721, then select one device) on the Tsi721 PCIe side to issue a PCIe MRd (Tsi721 translates it to S-RIO NREAD) to another device that sits on the Tsi721 S-RIO side, and waits for the completion to return.
3. Update the Tsi721's outbound window registers.

⚠ Upon power-up, all lookup table locations must be initialized to some value by software prior to reading. Reading from uninitialized lookup table locations may result in uncorrectable ECC errors.

8.3 S-RIO Address Translation

The 34/50/66-bit address of a S-RIO request packet (NREAD, NWRITE/SWRITE/NWRITE_R) is translated using one of the eight 66-bit inbound windows implemented in the SR2PC block. Note that maintenance request transactions have already been terminated inside S-RIO MAC ingress.

- Each window is defined by Inbound Window Translated Lower Address Register {0..7} and Inbound Window Translated Upper Address {0..7}, and Inbound Window Size Register{0..7}
- Windows must be non overlapping; otherwise, the device will have undefined behavior
- Supported window size is 2^N bytes, with N from 12 (4 KB) to 34 (16 GB)
- The Tsi721 provides Inbound Window Translated Lower Address Register {0..7} and Inbound Window Translated Upper Address {0..7} registers per window to translate an incoming S-RIO header into an outgoing PCIe header.
- If an S-RIO request packet does not hit a window, it is discarded, an interrupt is flagged, and an S-RIO error response is generated if the request is an NREAD.
- If an S-RIO request packet hits a window but its end address is beyond the upper limit of the window, the device will have undefined behavior.

If the translated address is below 4 GB, then a PCIe 3DW header is used; otherwise, a PCIe 4 dword header is used.

Figure 25: SR2PC Address Lookup

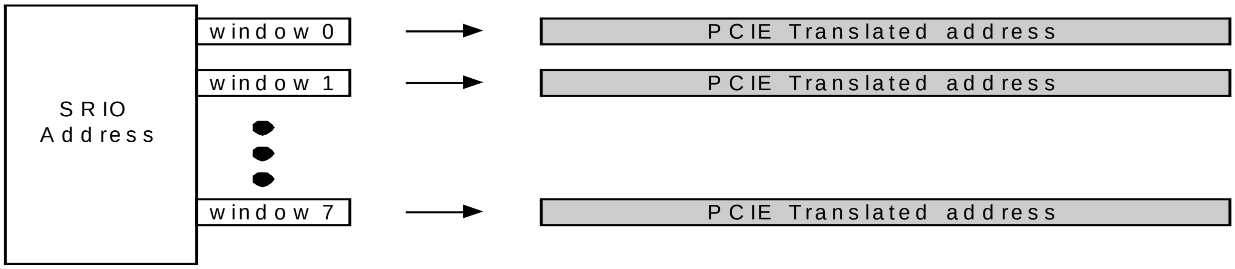
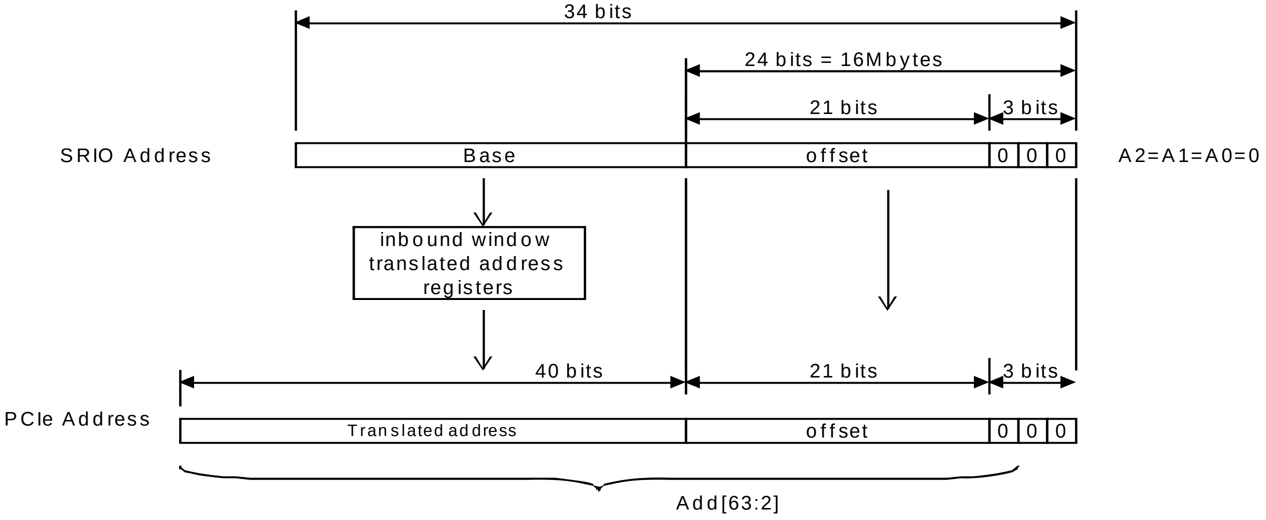


Figure 26 shows an example of an SWRITE to MWr address translation with a window size of 16 MB and 34-bit S-RIO addressing mode. In this example, the 21 S-RIO address offset bits are directly used as part of PCIe address. The S-RIO base address is used to find a matching inbound window, where associated Inbound Window Translated Lower Address Register {0..7} and Inbound Window Translated Upper Address {0..7} (MSB bits starting from bit 24) are used to generate 40 MSB PCIe address bits.

Figure 26: Sample S-RIO SWRITE to PCIe MWr Address Translation



8.3.1 Updating Inbound Window Registers

To update inbound window registers (one of Inbound Window Lower Base Register{0..7}, Inbound Window Upper Base Register{0..7}, Inbound Window Size Register{0..7}, Inbound Window Translated Lower Address Register {0..7}, and Inbound Window Translated Upper Address {0..7}), software should make sure that there are not any S-RIO packets buffered inside the Tsi721; otherwise, device behavior is undefined. A recommended procedure for updating inbound window registers is as follows:

1. Stop sending S-RIO packets from any device to the Tsi721 other than as required for subsequent steps in this procedure.
2. Instruct each device (or wait for longest delay among all devices for packets to arrive at the Tsi721, then select one device) on the Tsi721 S-RIO side to issue an S-RIO NREAD with prio/crf of 0/0 (Tsi721 translates it to PCIe MRd) to another device that sits on the Tsi721 PCIe side, and wait for the response to return.
3. Update inbound window registers.

8.4 Special PCIe-to-S-RIO Mapping Processing

Table 56 summarizes how the Tsi721 maps PCIe transactions into S-RIO packets, and indicates whether special processing is required for the transactions.

Table 56: PCIe-to-S-RIO Mapping Processing

Mapping Type		Special Processing Required?
PCIe Transaction	S-RIO Packet	
MWr	NWRITE	No
MWr	SWRITE	No
MWr	NWRITE_R (and associated NWRITE_R responses)	Yes

Table 56: PCIe-to-S-RIO Mapping Processing

Mapping Type		Special Processing Required?
PCIe Transaction	S-RIO Packet	
MWr	Doorbell (and associated doorbell responses)	Yes
MWr	Maintenance Write (and associated maintenance write responses)	Yes
MRd	Maintenance Read[a] (and associated S-RIO response to PCIe completion)	No
MRd	NREAD (and associated S-RIO response to PCIe completion)	No
Zero length MRd	1-byte NREAD	Yes

[a] Inbound maintenance reads are terminated by the S-RIO MAC and it generates maintenance read responses (see Local Logical Layer Management.

The PCIe-to-S-RIO mappings that require special processing are explained in the following sections.

8.4.1 Outbound Doorbell Processing

The Tsi721 translates a PCIe MWr hitting BAR1 into outbound doorbells as displayed in Figure 24. Note that the MWr must have a PCIe header length of 1 and byte enables as BE[3:0] = 0b0011. Since there are not corresponding PCIe constructs, the Tsi721 implements special outbound S-RIO doorbell processes as follows:

- It terminates received doorbell responses
- For each outbound doorbell channel, it provides a Outbound Doorbell Count {0..7} Register with two counters:
  - — A 16-bit ODB_OK_CNT to count total number of received OK doorbell responses
  - — A 16-bit ODB_TOT_CNT to count total number of doorbells sent
- When the first ERROR or RETRY doorbell response is received or the first doorbell response timeout for a specific doorbell channel is received, the Tsi721 logs the PCIe header of the associated PCIe MWr into an S-RIO doorbell response log buffer that can be accessed through Outbound Doorbell Response Log Buffer Data 0 Register {0..7}, Outbound Doorbell Response Log Buffer Data 1 Register {0..7}, Outbound Doorbell Response Log Buffer Data 2 Register {0..7}, Outbound Doorbell Response Log Buffer Data 3 Register {0..7}, where SR2PC Outbound Doorbell Response Log Buffer Status Register {0..7} identifies the S-RIO retry/error/timeout for the log buffer.
- All subsequent doorbell responses for this doorbell channel are still counted toward ODB_OK_CNT and ODB_TOT_CNT.
- Software must read the SR2PC Channel Interrupt Register {0..7} for the following reasons:
  - — To determine if a retry/error/timeout has occurred to a doorbell channel
  - — To retry a doorbell in response to a RETRY doorbell
  - — To recover from ERROR doorbell responses, or a doorbell response timeout

For systems that have doorbell retries, it is recommended for software to send a maximum of one doorbell of a specific outbound doorbell channel, then poll Outbound Doorbell Count {0..7} Register, SR2PC Outbound Doorbell Response Log Buffer Status Register {0..7} and SR2PC Channel Interrupt Register {0..7} to confirm if the doorbell has been successfully delivered or an ERROR/RETRY/timeout has occurred. Software must take corresponding actions accordingly before sending the next doorbell for the channel.

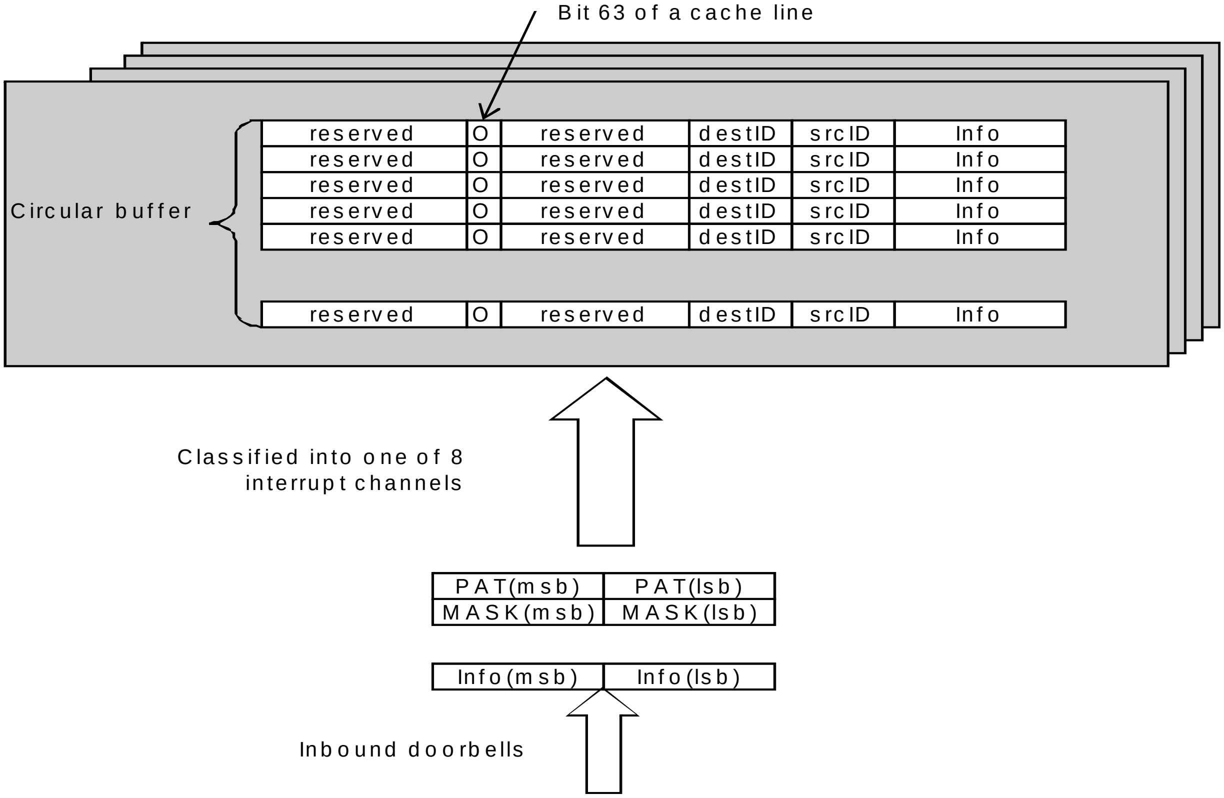
For systems that do not have doorbell retries, it is recommended for software to send doorbells of a specific outbound doorbell channel in a burst but keep sent doorbells for potential error recovery. After sending a burst of doorbells, software should wait to poll the Outbound Doorbell Count {0..7} Register to determine if all doorbells have been sent successfully. If ODB_OK_CNT equals the total number of doorbells sent, software can resume to send another burst of doorbells for the channel. If through INTx/MSI/MSI-X interrupt or polling SR2PC Outbound Doorbell Response Log Buffer Status Register {0..7} and SR2PC Channel Interrupt Register {0..7}, software finds that error responses have been received or a response timeout has occurred for the channel, software must launch the error recovery procedure. In this example, software can read the doorbell response log buffer to determine if a PCIe MWr in its doorbell queue has encountered an error/timeout. Since the S-RIO response order is undefined, software cannot identify which doorbell has an OK response. As a result, before sending a new burst of doorbells, software must poll the Outbound Doorbell Count {0..7} Register to see that ODB_TOT_CNT equals to total number of doorbells sent.

8.4.2 Inbound Doorbell Processing

The Tsi721 translates an inbound doorbell into a PCIe MWr to store the doorbell into an inbound doorbell queue. The device supports 8 queues for inbound doorbells, where each inbound doorbell queue resides in PCIe side main memory. Inbound doorbell queues are defined by Inbound Doorbell Queue Lower Base Register {0..7}, Inbound Doorbell Queue Upper Base Register {0..7}, and Inbound Doorbell Queue Size Register {0..7}.

Each inbound doorbell queue is implemented as a circular buffer. The Tsi721 performs inbound doorbell queue writes and software performs inbound doorbell queue reads. The device writes to an inbound doorbell queue until (Inbound Doorbell Queue Write Pointer Register {0..7} + 1) MOD Inbound Doorbell Queue Size Register {0..7} becomes equals to the software programmed Inbound Doorbell Queue Read Pointer Register {0..7}.

Figure 27: Inbound Doorbell Queue



S-RIO fields inside inbound doorbell MWr payload are in big-endian format as displayed in Figure 28. In this example, TT[1]/srcID[15]/destID[15]/info[15] is the MSB bit of S-RIO tt/srcID/destID/info. Note that S-RIO doorbell info byte 0 is MSB and is mapped to PCIe byte 0.

Figure 28: Inbound Doorbell Endian

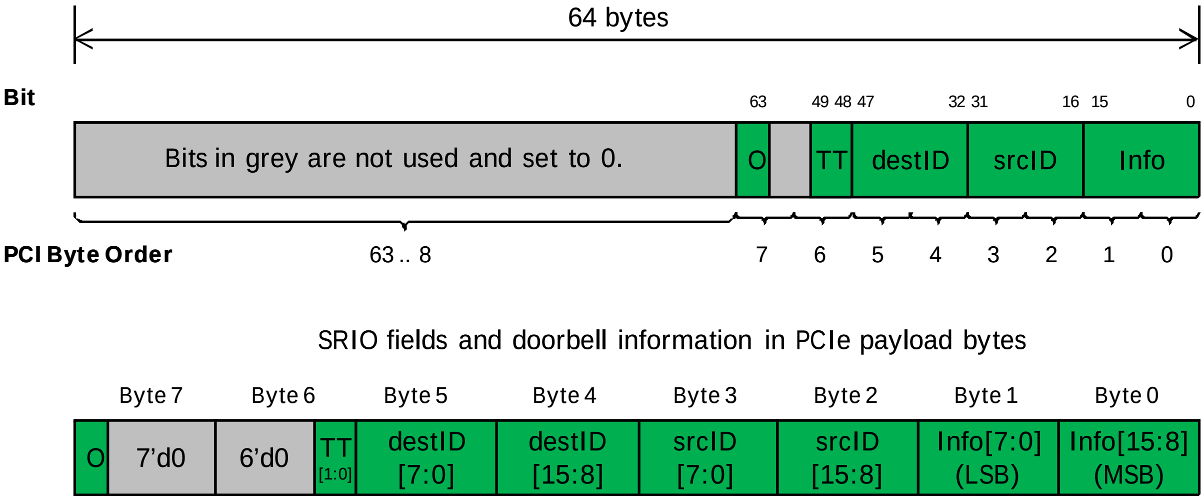
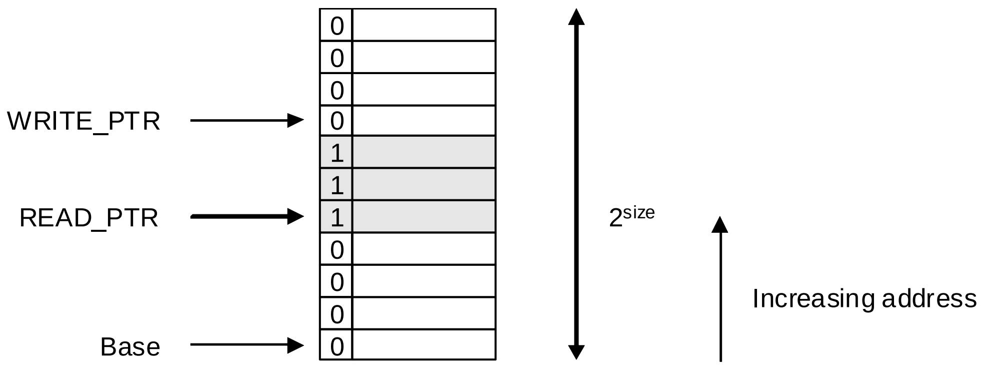


Figure 29: Circular Buffer of an Inbound Doorbell Queue



Each entry of an inbound doorbell queue has 64 bytes (a cache line) and an Tsi721 generated PCIe MWr for an inbound doorbell has 64 bytes of payload (to match cache line size for Intel performance):

- Bit 63: hardware own bit (O). High indicates that the entry is valid (Tsi721 has written the entry), low indicates that the entry is invalid (software has processed the entry).
- Bits 15–0 is doorbell info bits in big-endian format
- Bits 31–16 is doorbell srcID in big-endian format
- Bits 47–32 is doorbell destID in big-endian format
- Other bits are unused

8.4.2.1 Inbound Doorbell Classification

Doorbells received on the S-RIO MAC ingress interface are classified into interrupt channels 0–7. Classification is based on the doorbell's information bits and the Inbound Doorbell Classification Register {0..7}. In this example, if bit wise AND between a doorbell's information bits and the MASK field of a classification register equals to the PATTERN field of the same classification register, then the doorbell is considered a match to the classification register and is assigned to the associated inbound doorbell queue (Doorbell Classification Register X is assigned to inbound doorbell queue X). In this example, priority encoding is used when multiple matches are found, with Doorbell Classification Register 0 is given the highest priority.

A doorbell that does not match any of the classification registers is discarded and an interrupt (DB_MISS bit of SR2PC General Interrupt CSR) is flagged (ERROR doorbell response is still generated).

8.4.2.2 Inbound Doorbell Response Generation

The Tsi721 generates ERROR doorbell response to an inbound doorbell under one of the following conditions:

- An inbound doorbell does not match any Inbound Doorbell Classification Register {0..7}
- An inbound doorbell arrives at an un-initialized inbound doorbell queue
- An inbound doorbell arrives at a suspended (torn down) inbound doorbell queue

The Tsi721 generates RETRY doorbell response when an inbound doorbell arrives at a associated inbound doorbell queue that is full; otherwise, it generates an OK doorbell response.

8.4.2.3 Inbound Doorbell Queue Initialization and Suspend

8.4.2.3.1 Inbound Doorbell Queue Initialization

Software initializes an inbound doorbell queue by programming Inbound Doorbell Queue Lower Base Register {0..7}, Inbound Doorbell Queue Upper Base Register {0..7}, Inbound Doorbell Queue Size Register {0..7}, and Inbound Doorbell Classification Register {0..7} (these registers must be static after a channel has been initialized until the channel is re-initialized again), then setting INIT bit in the Inbound Doorbell Queue Control Register {0..7}. The Tsi721 resets registers Inbound Doorbell Queue Read Pointer Register {0..7} and Inbound Doorbell Queue Write Pointer Register {0..7} to zero when INIT is high.

Inbound doorbell processing is initiated when software writes to the Inbound Doorbell Queue Read Pointer Register {0..7}, which the Tsi721 then resets INIT.

8.4.2.3.2 Inbound Doorbell Queue Suspend and Resume

Inbound doorbell queue suspend and resume capability is provided for an application to tear down an inbound doorbell queue. Received inbound doorbells to a suspended inbound doorbell queue are discarded and ERROR responses are generated.

The operation of an inbound doorbell queue can be suspended by writing a 1 to the SUSPEND bit in the SR2PC Registers. When an inbound doorbell queue is suspended, the following actions occur:

- The inbound doorbell queue processes normally all inbound doorbells that are active (called outstanding doorbells)
- All remaining and subsequently arrived doorbells for this inbound doorbell queue are discarded and ERROR responses are generated
- Once all outstanding doorbells have been processed, the SR2PC block loads the Inbound Doorbell Queue Write Pointer Register {0..7} to point to next address after the last processed doorbell, and the SUSPENDED bit is set in the SR2PC Channel Interrupt Register {0..7}.
- If the inbound doorbell queue has become full when suspended, then the SUSPENDED bit in the SR2PC Channel Interrupt Register {0..7} is immediately set.
  - — Software should wait for the SUSPENDED bit in the SR2PC Channel Interrupt Register {0..7} to be set before resuming the inbound doorbell queue as described next. Violating this rule produces undefined behavior.
  - — When the inbound doorbell queue is full, the Inbound Doorbell Queue Write Pointer Register {0..7} points to the next address after the last processed doorbell.
- The RUN bit in Inbound Doorbell Queue Status Register {0..7} is cleared while an inbound doorbell queue is suspended.
- A suspended inbound doorbell queue can be resumed by writing the Inbound Doorbell Queue Read Pointer Register {0..7}. Resume starts regardless of the value in Inbound Doorbell Queue Read Pointer Register {0..7}, upon which the SUSPEND bit is cleared.

8.4.3 Outbound MWr to NWRITE_R Processing

The Tsi721 translates a PCIe MWr hitting an outbound window with a zone's WR_TYPE of NWRITE_R (see Outbound Window Lookup Table Data 0 Register) into NWRITE_R. Since there is not corresponding PCIe constructs, the Tsi721 implements special NWRITE_R processes as displayed below:

- The Tsi721 terminates received NWRITE_R responses
- The Tsi721 provides a NWRITE_R Count Register with two counters
  - — A 16-bit NW_OK_CNT to count total number of MWr receiving OK NWRITE_R responses only
  - — A 16-bit NW_TOT_CNT to count total number of MWr sent that are mapped into NWRITE_R

- When the first ERROR NWRITE_R response or the first NWRITE_R response timeout is received, the Tsi721 logs the PCIe header of the associated PCIe MWr into an S-RIO NWRITE_R response log buffer that can be accessed through NWRITE_R Response Log Buffer Data 0 Register, NWRITE_R Response Log Buffer Data 1 Register, NWRITE_R Response Log Buffer Data 2 Register, NWRITE_R Response Log Buffer Data 3 Register, where SR2PC Log Buffer Status Register identifies the S-RIO error/timeout for the log buffer.
- All subsequent NWRITE_R responses are still counted toward NWR_OK_CNT and NWR_TOT_CNT
- Software is responsible reading SR2PC General Interrupt CSR, and recovering from ERROR NWRITE_R responses or NWRITE_R response timeouts.

One recommendation is for software to send MWr (maps to NWRITE_R) in a burst but keep sent MWr for potential error recovery. Software should then poll the NWRITE_R Count Register to see if it is equal to the total MWr sent. If yes, software can resume to send another MWr burst mapped to NWRITE_R. Otherwise, through INTx/MSI/MSI-X interrupt or polling SR2PC Log Buffer Status Register and SR2PC General Interrupt CSR, software can find out that NWRITE_R error/timeout has occurred and software can read NWRITE_R response log buffer to find out a MWr encountered an error/timeout, and software can launch error recovery. As S-RIO response order is undefined, software cannot identify which MWr has OK response. In this example, before sending a new burst of MWr mapped to NWRITE_R, software must polls the NWRITE_R Count Register to see that NW_TOT_CNT is equal to the number of MWr (mapped to NWRITE_R) sent.

8.4.4 Inbound NWRITE_R to MWr Processing

The Tsi721 generates ERROR NWRITE_R response to a received NWRITE_R under one of the following conditions:

- A received NWRITE_R missing all inbound windows
- A received NWRITE_R with an S-RIO illegal decode error

Otherwise, the device generates an OK NWRITE_R response after sending translated PCIe MWr.

8.4.5 Outbound MWr to Maintenance Write Processing

The Tsi721 translates a PCIe MWr hitting an outbound window with a zone's WR_TYPE of maintenance write (see Outbound Window Lookup Table Data 0 Register) into a maintenance write. Since there are not any corresponding PCIe TLP types, the Tsi721 implements special maintenance write processes as displayed below:

- The Tsi721 terminates received maintenance write responses
- The Tsi721 provides a 32-bit Maintenance Write Count Register with two counters
  - — A 16-bit MW_OK_CNT to count total number of received OK maintenance write responses
  - — A 16-bit MW_TOT_CNT to count total number of maintenance writes sent
- When the first ERROR maintenance write response or the first maintenance write response timeout is received, the Tsi721 logs the PCIe header of the associated PCIe MWr into an S-RIO maintenance write response log buffer that can be accessed through Maintenance Write Response Log Buffer Data 0 Register, Maintenance Write Response Log Buffer Data 1 Register, Maintenance Write Response Log Buffer Data 2 Register, Maintenance Write Response Log Buffer Data 3 Register, where SR2PC Log Buffer Status Register identifies the S-RIO error/timeout for the log buffer.
- All subsequent maintenance write responses are still counted toward MW_OK_CNT and MW_TOT_CNT
- Software is responsible for reading SR2PC General Interrupt CSR, and recovering from ERROR maintenance write responses, or maintenance write response timeout.

One recommendation is for software to send a maintenance write in a burst but keep sent maintenance write for potential error recovery. Software should then poll the Maintenance Write Count Register to see if MW_OK_CNT is equal to the number of MWr (mapped to maintenance write) sent. If yes, software can resume to send another burst. Otherwise, through INTx/MSI/MSI-X interrupt or polling SR2PC Log Buffer Status Register and SR2PC General Interrupt CSR, software can finds out that maintenance write error/ timeout has occurred, and software can reads maintenance write response log buffer to find out a MWr encountered an error/timeout, and launches error recovery. As S-RIO response order is undefined, software cannot identify which MWr has OK response. In this example, before sending a new burst of maintenance writes, software must polls the Maintenance Write Count Register to see that MW_TOT_CNT is equal to the number of MWr (mapped to maintenance write) sent.

8.4.6 Inbound Maintenance Write to MWr Processing

The S-RIO MAC terminates maintenance writes and generates maintenance write responses (see Local Logical Layer Management).

8.4.7 MRd to Maintenance Read and NREAD Processing

If no RapidIO Maintenance Response or Response packet is received, the Tsi721 completes the MRd transaction on the PCIe bus using a Target Abort, which results in a processor Machine Check exception.

Machine Check exceptions may trigger fatal error management by the operating system. For this reason, it is recommended that DMA engines are used to generate all RapidIO requests except doorbells.

8.4.8 Zero Length MRd Processing

Since *RapidIO Specification (Rev. 2.1)* does not support zero length reads, the Tsi721 offers the following solution to zero length MRd mapping:

- For a zero length MRd hitting BAR0 (Tsi721 internal register space), the Tsi721 generates CplD according to the *PCI Express Base Specification (Rev. 2.1)*
- For a zero length MRd hitting BAR 2/3/4/5 (Tsi721 bridging BARs), the Tsi721 translates the MRd into an S-RIO NREAD requesting 1-byte payload, and translates the corresponding S-RIO response into a PCIe CplD. Since the *RapidIO Interconnect Specification (Revision 2.1)* does not support zero length read, applications must ensure that zero length MRd targets prefetchable memory space.

8.5 Mapping Engine Scheduling

8.5.1 PCIe Egress Scheduling

For PCIe egress scheduling, the Tsi721 uses the following algorithm:

1. PCIe ordering rules for bridging TLPs (those mapped from S-RIO packets except MWr mapped from inbound doorbells)
2. PCIe ordering rules for block DMA TLPs, where the Block DMA Engine has its internal scheduling, as discussed in Block DMA Engine Scheduling
3. PCIe ordering rules for Messaging Engine TLPs, where Messaging Engine has its internal scheduling, as discussed in Messaging Engine Scheduling
4. Round-robin scheduling between winners of steps 1 to 3
5. Strict priority scheduling between the following TLPs (displayed in decreasing priority order)
   — INTx/MSI/MSI-X
   — MWr from inbound doorbell queue manager

- Register read completion and Tsi721 generated UR/CA completion
- Winners of step 4

8.5.2 S-RIO Egress Scheduling

For S-RIO egress scheduling, the Tsi721 uses the following algorithm:

1. S-RIO ordering rules for bridging packets (those mapped from PCIe TLPs)
2. S-RIO ordering rules for Block DMA Engine packets
3. S-RIO ordering rules for Messaging Engine packets
4. Round-robin scheduling between winners of steps 1 to 3
5. Strict priority scheduling between the following packets (displayed in decreasing priority order)
   - NWRITE_R/doorbell responses and error responses to NREAD missing all inbound windows
   - Winners of step 4

9. Messaging Engine

Topics discussed include the following:

- Overview
- Outbound Messaging
- Inbound Messaging
- Messaging Engine Scheduling

9.1 Overview

9.1.1 Functions

The Messaging Engine provides messaging internetworking between the Tsi721's S-RIO fabric and a PCIe-based root complex.

9.1.2 Features

9.1.2.1 Outbound Messaging Features

- Message size up to 4 KB
  - Maximum of 256-byte segment
  - Maximum of 16 segments
  - If ssize is less than 256 bytes, the maximum message size is ssize*16
- Implements 8 outbound message DMA channels
  - One DMA channel per Tx queue
  - Descriptor based
  - Gather (header and data read from different memory areas)
- 8 Tx queues
  - Round-robin for S-RIO side scheduling
  - 1 transmit context per Tx queue to enforce one outstanding message per Tx queue
  - Prefetch 32 descriptors per Tx queue
  - 8-KB reassembly queue per Tx queue
- Store and forward for message reassembly queue

9.1.2.2 Inbound Messaging Features

- Message size up to 4 KB
  - Maximum of 256-byte segment
  - Maximum 16 segments

- Implements 8 inbound message DMA channels
  - — One DMA channel per Rx queue
  - — Descriptor based
  - — Scatter (header and data written to different memory areas)
- 2 Rx queues per mailbox
  - — 16 receive contexts per Rx queue supporting 4–16 simultaneous messaging sources per Rx queue
- Store and forward for inbound message segment FIFO

9.1.3 S-RIO Messaging

S-RIO messaging provides an efficient method for message passing between processors. Messages are sent with a destID/srcID, and mailbox/letter tag. There is no memory address in the packet; it is up to the receiving processor to classify the message, and transfer the payload through DMA into memory. This allows decoupling of the memory space between communicating processors.

The *RapidIO Interconnect Specification (Revision 2.1)* standard allows up to 4-KB sized messages, and with a 256 maximum payload size of a S-RIO packet, up to 16 segments. Note that the last segment of a message has the same ssize as previous segments; its actual payload size is arbitrary (less than or equal to ssize), and a receiver can determine its payload size only by checking the end of the S-RIO packet. The Tsi721 requires software to set up a 4-KB buffer inside main memory for each received or transmit message, even for single segment messages that have at most only 256 bytes of payload. The messages usually require segmentation and reassembly. If a receiver does not have a free receive context for an income message, the message is retried at the logical layer. This can be wasteful of fabric bandwidth, so the Tsi721 provides 16 receive contexts per Rx queue.

Messages with the same mailbox/letter/srcID/dstID/prio/crf must be delivered in order. Segments for a specific message may not be delivered in order due to retry, with the most common cause of retry being lack of receive contexts. S-RIO links also reorder messages of different prio/crf. When there is not a message segment retry, the device sends segments of a message in order.

It is not possible to burst back-to-back messages to the same mailbox/letter/destID. The first message is still in flight, and thus the retry/done status of all the segments is not known when it is time to send the second message. If the second message were sent optimistically, and the first segment retried then the message could be delivered out of order to the processor at the receiver. The Tsi721 provides one context per Tx queue to support at most one outstanding message per Tx queue. The device makes sure that there will never be two outstanding messages with the same mailbox/letter/destID. That is, if two or more Tx queues have messages with the same mailbox/letter/destID, it makes sure that only one of the Tx queues have an outstanding message. The device also allows software to set letter of a message on per descriptor basis.

9.1.3.1 S-RIO Message Packet Format

S-RIO messaging uses type 11 packets, as displayed in Figure 30. The field encoding for this packet type is displayed in Table 57.

Figure 30: S-RIO Message Packet – Type 11

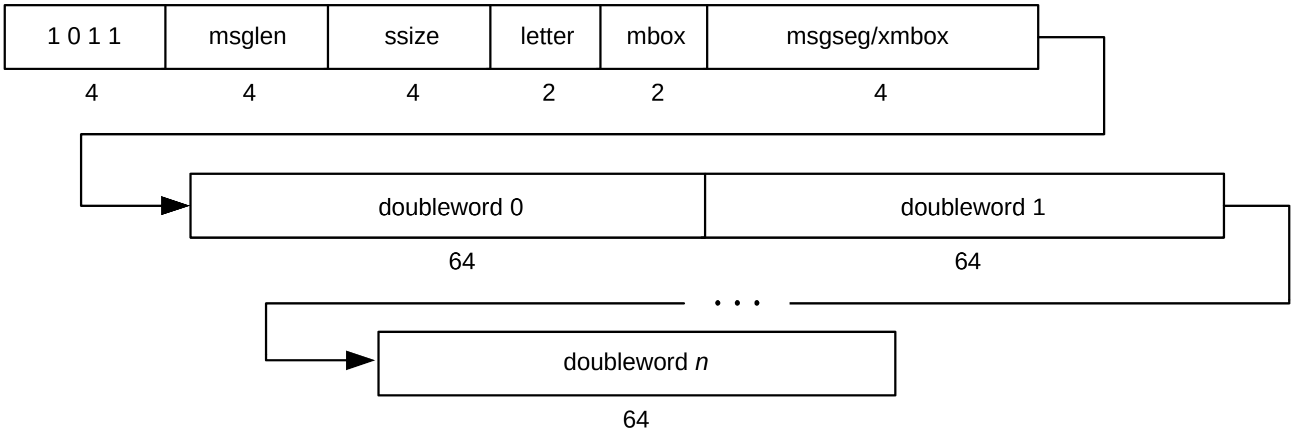


Table 57: S-RIO Message Packet

Field	Encoding	Definition
msglen	-	Total number of packets comprising this message operation. A value of 0 indicates a single-packet message. A value of 15 (0xF) indicates a 16-packet message, and so on.
msgseg	-	For multiple packet data message operations, specifies the part of the message supplied by this packet. A value of 0 indicates that this is the first packet in the message. A value of 15 (0xF) indicates that this is the sixteenth packet in the message, and so on.
xmbox	-	For single-packet data message operations, specifies the upper 4 bits of the mailbox targeted by the packet. xmbox || mbox are specified as follows: 0000 00 = Mailbox 0 0000 01 = Mailbox 1 0000 10 = Mailbox 2 0000 11 = Mailbox 3 0001 00 = Mailbox 4 ... 1111 11 = Mailbox 63

Table 57: S-RIO Message Packet *(Continued)*

Field	Encoding	Definition
ssize	-	Standard message packet data size. This field informs the receiver of a message the size of the data payload to expect for all of the packets for single message operations, except for the last packet in the message. This prevents the sender from having to pad the data field excessively for the last packet and allows the receiver to properly put the message in local memory.
	0b000–1000	Reserved
	0b1001	8 bytes
	0b1010	16 bytes
	0b1011	32 bytes
	0b1100	64 bytes
	0b1101	128 bytes
	0b1110	256 bytes
	0b1111	Reserved
mbox	-	Specifies the recipient mailbox in the target processing element
letter	-	Identifies a slot within a mailbox. This field allows a sending processing element to send up to four concurrent messages to the same mailbox on the same processing element.

9.1.3.2 S-RIO Message Response Packet Format

The S-RIO messaging response packet format is displayed in Figure 31. The field encoding for this packet type is displayed in Table 58.

Figure 31: S-RIO Message Response Packet – Type 13

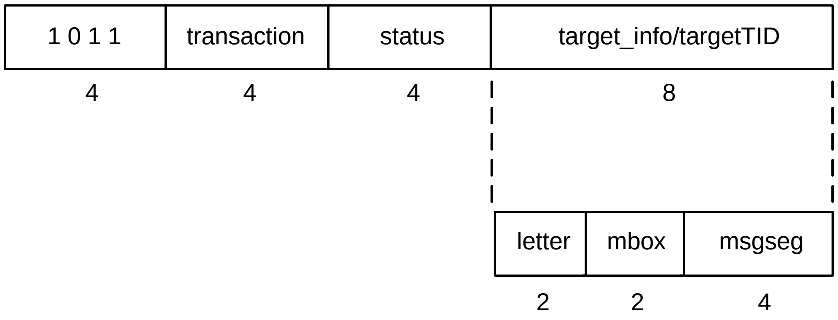


Table 58: S-RIO Message Response Packet

Field	Encoding	SUB-Field	Definition
transaction	0b0001		MESSAGE RESPONSE transaction
status	0b0000	DONE	Requested transaction has been successfully completed
	0b0011	RETRY	Requested transaction is not accepted; re-transmission of the request is needed to complete the transaction
	0b0111	ERROR	Unrecoverable error detected
target_info	msgseg		Specifies the part of the message supplied by the corresponding message packet. A value of 0 indicates that this is the response for the first packet in the message. A value of 15 (0xF) indicates that this is the response for the sixteenth (and last) packet in the message, and so on.
	mbox		Specifies the recipient mailbox from the corresponding message packet.
	letter		Identifies the slot within the target mailbox. This field allows a sending processing element to send up to four concurrent messages to the same mailbox on the same processing element.

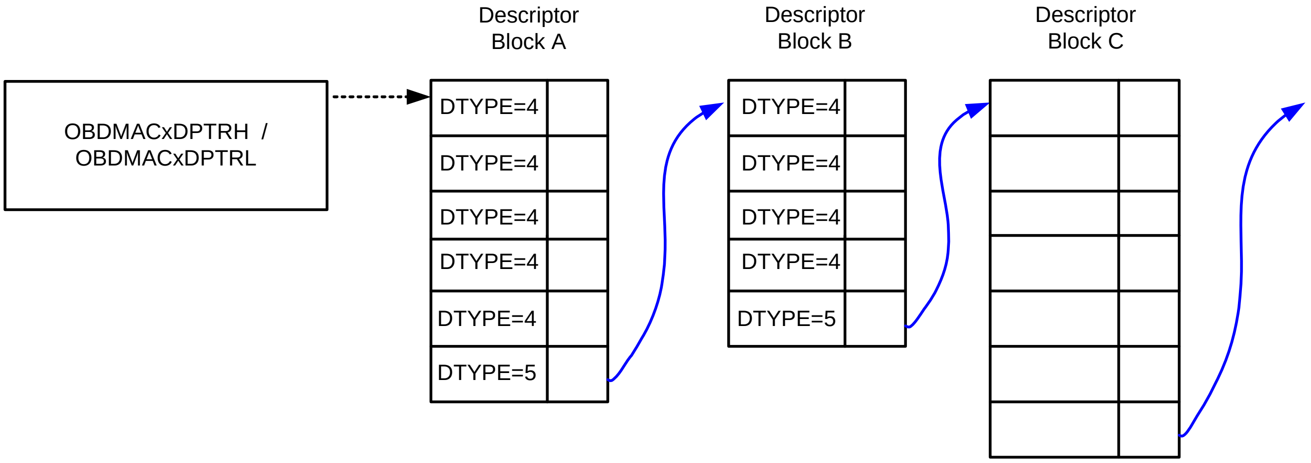
9.2 Outbound Messaging

The Outbound Messaging Engine implements 8 DMA channels, one per Tx queue. Each DMA channel is designed as a list of descriptor blocks, where DMA descriptors and associated DMA data buffers reside in PCIe side main memory. The Outbound Messaging Engine performs descriptor prefetch by issuing descriptor MRd – with up to 32 prefetched descriptors per DMA channel – and performs DMA data fetch by issuing data MRd. A total of 32 outstanding PCIe MRd is supported and they are shared by descriptor MRd and data MRd of all DMA channels (from both Inbound and Outbound Messaging Engine). Each DMA channel has its own message segment reassembly queue.

9.2.1 DMA Descriptors

- The Tsi721 implements one DMA channel per Tx queue
- A DMA channel operates by reading descriptors from memory and performing the specified processing.
- The descriptor address must not be all zeroes, as descriptor address of all zeroes is reserved to mark an invalid entry in a descriptor status FIFO residing in PCIe side main memory.
- Complex data movement operations, such as scatter gather, can be used by linking multiple descriptor blocks together into a descriptor list, where descriptors within each descriptor block are in contiguous memory space. Figure 32 shows a descriptor list (with DTYPE denotes descriptor type).

Figure 32: DMA Descriptor List



The Block DMA Engine uses two types DMA descriptors (see Table 59).

Table 59: DMA Descriptor Types

DTYPE	Name	Description
DTYPE 4	Data transfer descriptor	Used for general data movement
DTYPE 5	Block pointer descriptor	Used to link descriptor blocks

- Each DMA channel has only one DMA descriptor list.
- One DMA descriptor list can mix all types of descriptors.
- A descriptor block consists of multiple descriptors in contiguous memory space:
  - The last descriptor of a block can have any DTYPE
  - A descriptor of DTYPE 5 ends a descriptor block
  - Each 4-KB page that contains a descriptor must be prefetchable (see Descriptor Prefetching). Violation of this rule may cause undefined results due to descriptor prefetch.
- All descriptors have a fixed size of 16 bytes. They must be 16-byte aligned; otherwise, Tsi721 behavior is undefined.

9.2.1.1 Ending of Descriptor List

The last descriptor in a descriptor list is not directly encoded in the descriptor. Instead, software and the Tsi721 communicate the last descriptor through two rolling counters per DMA channel:

- Descriptor write count – A 32-bit non-saturating running count of generated descriptors by software
- Descriptor read count – A 32-bit non-saturating running count of prefetched descriptors by the Tsi721

These two counts are reset to 0 during a DMA channel initialization. The Tsi721 implements its version of the descriptor read and write counts using the Outbound DMA Descriptor Read Count Register {0..7} and Outbound DMA Descriptor Write Count Register {0..7}. Software informs the Tsi721 about the descriptor write count by updating the Outbound DMA Descriptor Write Count Register {0..7} through PCIe MWr. Under normal operations (no error occurred or DMA channel abort/ initialization/ suspending), the Tsi721 continues prefetching and processing descriptors until the Outbound DMA Descriptor Read Count Register {0..7} becomes equal to the Outbound DMA Descriptor Write Count Register {0..7}. The device then assumes that the last processed descriptor is the last one in the descriptor list, and stops descriptor processing until software updates the descriptor write count. When software updates the Outbound DMA Descriptor Write Count Register {0..7}, the Tsi721 automatically starts descriptor prefetching and processing again.

Software and the Tsi721 reset the read and write counts during a DMA channel initialization. Software can track the descriptor read count by reading the Outbound DMA Descriptor Read Count Register {0..7}. To achieve high throughput, software should monitor descriptor status – through either polling or interrupt – then timely generate enough descriptors ahead of Tsi721 DMA engine, and update the Tsi721 about descriptor write count.

9.2.1.2 Descriptor Ordering Rules

The Messaging Engine generates PCIe MRd to fetch descriptors within the PCIe side main memory. Even though cplD for descriptor MRd and data MRd arrive at the Tsi721 in an undefined order, the Tsi721 enforces the S-RIO ordering rules for descriptors of a specific DMA channel as follows (descriptors among different DMA channels are not ordered):

- Within a descriptor, message segments are sent in increasing msgseg order except retried message segments
- Among descriptors of the same DMA channel, the Messaging Engine sends messages in descriptor order
  - Within a DMA channel, generated message segments from a newer descriptor are not allowed to pass those from an older descriptor of the same DMA channel, even during a message segment retry
  - If descriptors of a specific DMA channel have different prio/crf fields, the S-RIO MAC may reorder these message segments under blocking conditions since they are assigned S-RIO prio/crf according to prio/crf fields of associated descriptors
  - Each DMA channel has at most one outstanding message
- Among descriptors of different DMA channels, messages are unordered
  - When the latest messages from any two DMA channels have the same mbox/xmbox/destID/letter, only one DMA channel is selected to have an outstanding message and the other DMA channel will be stalled.

9.2.1.3 Data Transfer Descriptor

⚠ The operation of the Tsi721 is undefined after processing a descriptor with reserved values in any descriptor field.

The data transfer descriptor has a DTYPE of 4. The format of a data transfer descriptor is displayed in Figure 33, and the fields are described in Table 60.

- The format is displayed in little-endian.
- Fields marked as R are reserved.

Figure 33: Data Transfer DMA Descriptor Format – DTYPE 4

	bte 3								byte 2								byte 1								byte 0							
DW	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	DTYPE			R	IOF	R	R	R						PRIO		CRF	DEVID															
1	R				TT[1:0]		R		MBOX		XMBOX				LETTER		SSIZE				BCOUNT[11:3]									R		
2	BUFFER_PTR[31:0]																															
3	BUFFER_PTR[63:32]																															

Table 60: Data Transfer DMA Descriptor Format – DTYPE 4

Field	dword	Bit Position	Description
DEVID15:0]	0	15:0	S-RIO destination deviceID
CRF	0	16	S-RIO critical request flow
PRIO[1:0]	0	18:17	S-RIO priority. Valid values are 0, 1, and 2 (for request flows), as PRIO of 3 is for S-RIO response per part 6 of the *RapidIO Specification (Rev. 2.1)*.
IOF	0	27	Interrupt on finished. When this bit is set, the Tsi721 will generate an interrupt when it finishes processing this descriptor, and the IOF_DONE bit is set in the corresponding Outbound DMA Channel Interrupt Register {0..7}.
DTYPE[2:0]	0	31:29	Descriptor type. This field encodes the type of descriptor, and must be set to 0x4 in DMA descriptors. • 0x4 = Data transfer DMA descriptor • 0x5 = Block pointer descriptor • Others = Reserved
BCOUNT[11:3]	1	11:3	Byte count. Total number of bytes to transfer in units of 8 bytes. • 0b0_0000_0000 = 512 (4 KB) • 0b0_0000_0001 = 1 (8 bytes) • 0b0_0000_0010 = 2 (16 bytes) • 0b1_1111_1111 = 511 (4 KB-8 bytes)
SSIZE[3:0]	1	15:12	S-RIO message standard segment size
LETTER[1:0]	1	17:16	S-RIO message letter
XMBOX[3:0]	1	21:18	S-RIO message XMBOX for single segment messages; otherwise, it should be set to 0
MBOX[1:0]	1	23:22	S-RIO message MBOX
TT[1:0]	1	27:26	S-RIO transport type. • 0b00 = 8-bit deviceID • 0b01 = 16-bit deviceID • Others = Reserved
BUFFER_ PTR[31:0]	2	31:0	PCIe address lower. Lower 32-bits of the 64-bit PCIe address.
BUFFER_ PTR[63:32]	3	31:0	PCIe address upper. Upper 32-bits of the 64-bit PCIe address.

- One DMA data transfer is for one S-RIO message, and may consist of multiple S-RIO message segments

- A data transfer DMA descriptor instructs the DMA channel to perform a data transfer operation. Processing of the descriptor completes when either the data transfer operation completes or when an error is detected.
  - — The 64-bit PCIe address, BUFFER_PTR[63:0], may have any byte alignment
  - — If the PCIe address is below 4 GB, then a MRd TLP with a 32-bit address is generated. If the address is above 4 GB, then a MRd TLP with a 64-bit address is generated.
  - — Multiple MRd are generated due to BCOUNT exceeds MRRS in the PCIe Device Control and Status Register
  - — Processing for a descriptor has completed successfully when *both* of the following conditions are met:
    - – All message requests associated with that descriptor have been sent
    - – All S-RIO responses associated with the message packets have been received successfully
- Descriptor requested payload crosses 4-KB boundary
  - — The DMA channel attempts to issue the largest memory read request possible that is less than or equal to MRRS in the PCIe Device Control and Status Register, and does not cause the read request to cross a 4-KB boundary.
  - — In response to a memory read request, one or more completions are returned. These completions are transformed by the DMA channel into one or more S-RIO message segments
    - – The DMA channel reassembles multiple completions into a message segment because a message segment with ssize of 256 bytes may need two MRd if MRRS in the PCIe Device Control and Status Register is 128B, and there can be multiple cplD for a single MRd.
    - – However, when completion payload size is larger than a message's SSIZE, the Messaging Engine splits completions into multiple message segments.

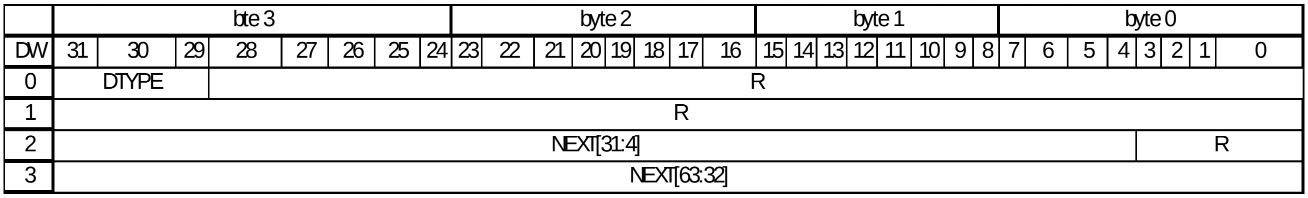
As per *PCI Express Specification (Rev. 2.1)* rules, completions associated with different requests have no ordering relationship. Thus, such completions cannot be received by a requester in the same order that the requests were issued. The Messaging Engine reassembles these completions message segments so that resulting messages from a specific DMA channel arrive at the destination location in correct order (see Descriptor Ordering Rules).

9.2.1.4 Block Pointer Descriptor

The operation of the Tsi721 is undefined after processing a descriptor with reserved values in any descriptor field.

The block pointer descriptor has a DTYPE of 5. It is essentially a pointer that links together descriptor blocks. The block pointer descriptor format is displayed in Figure 34, and explained in Table 61.

Figure 34: Block Pointer DMA Descriptor Format – DTYPE 5



	bte 3								byte 2								byte 1								byte 0							
DW	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	DTYPE			R																												
1	R																															
2	NEXT[31:4]																												R			
3	NEXT[63:32]																															

Table 61: Block Pointer DMA Descriptor Format – DTYPE 5

Field	dword	Bit Position	Description
DTYPE[2:0]	0	31:29	Descriptor type. This field encodes the type of descriptor. • 0x4 = Data transfer DMA descriptor • 0x5 = Block pointer descriptor • Others = Reserved
NEXT[31:4]	2	31:4	Next descriptor block pointer lower. {NEXT[31:4], 0b0000} (Tsi721 automatically generates all zeroes for LSB bits 0–3) forms the lower 32 bits of next descriptor block pointer. It points to the start address of the next descriptor block. It must be 16-byte aligned; otherwise, Tsi721 behavior is undefined.
NEXT[63:32]	3	31:0	Next descriptor block pointer upper. This is the upper 32 bits of the next descriptor block pointer.

9.2.2 DMA Descriptor Processing

DMA descriptor processing consists of reading a descriptor from memory, executing the operation outlined by the descriptor, and then proceeding to the next descriptor.

9.2.2.1 DMA Descriptor Status

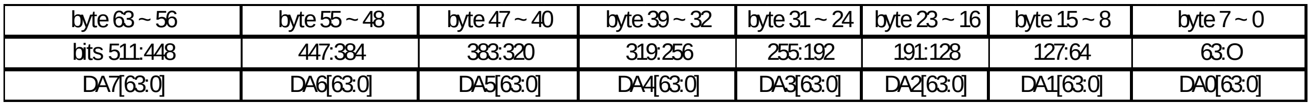
A DMA channel aborts any further DMA processing on either a PCIe or S-RIO error. As a result, all processed descriptors before error have only one status: OK status.

For each DMA channel, the Tsi721 maintains a descriptor status FIFO in the PCIe side main memory, where the FIFO stores a pointer to successfully processed DTYPE 4/5 descriptors. The status FIFO is defined by the following registers:

- Outbound DMA Descriptor Status FIFO Lower Base Register {0..7}
- Outbound DMA Descriptor Status FIFO Upper Base Register {0..7}
- Outbound DMA Descriptor Status FIFO Size Register {0..7}

The FIFO is organized as 64-byte wide, matching Intel processor's cache line size. Each FIFO entry of 64 bytes is divided into 8 elements, with 8 contiguous bytes per element (this is displayed as DA0[63:0] to DA7[63:0] in Figure 35). Each element can store one 8-byte pointer to a successfully processed descriptor by the Tsi721, where an all zero element – for example, DAj (j = 0–7) – is reserved to mark this and higher address elements (DAk, k>=j) as invalid. That is, the Tsi721 updates the descriptor status on a per cache line basis, and can store from one to eight pointers to successfully processed descriptors in one status MWr TLP into consecutive elements (starting from LSB) of one FIFO entry.

Figure 35: Descriptor Status FIFO Entry Format



byte 63 ~ 56	byte 55 ~ 48	byte 47 ~ 40	byte 39 ~ 32	byte 31 ~ 24	byte 23 ~ 16	byte 15 ~ 8	byte 7 ~ 0
bits 511:448	447:384	383:320	319:256	255:192	191:128	127:64	63:O
DA7[63:0]	DA6[63:0]	DA5[63:0]	DA4[63:0]	DA3[63:0]	DA2[63:0]	DA1[63:0]	DA0[63:0]

The FIFO start address must be 64-byte aligned. Software and the Tsi721 co-manage the descriptor status FIFO, with software performing FIFO reads and the Tsi721 performing FIFO writes. Software informs the Tsi721 about its read pointer through the RD_PTR field in the Outbound DMA Descriptor Status FIFO Read Pointer Register {0..7}, and Tsi721 updates its write pointer through WR_PTR in the Outbound DMA Descriptor Status FIFO Write Pointer Register {0..7}. When a FIFO entry is read, software clears the entry with all zeroes. Software informs the Tsi721 about its read pointer through a register write. The Tsi721 stalls FIFO write when (WR_PTR + 1) MOD Outbound DMA Descriptor Status FIFO Size Register {0..7} become equals to RD_PTR. Software should decide if an FIFO entry is valid by continuously reading the FIFO until an all zero entry (Tsi721 write pointer should not be used).

9.2.2.2 Descriptor List Processing

- Software initializes a DMA channel by programming the Outbound DMA Channel Descriptor Pointer Low Register {0..7} and Outbound DMA Channel Descriptor Pointer High Register {0..7}; together these registers form a 64-bit descriptor address. Software also clears the ERROR bit in the Outbound DMA Channel Interrupt Register {0..7}, then sets the INIT bit in the Outbound DMA Channel Control Register {0..7}.
  - — Tsi721 resets Outbound DMA Descriptor Write Count Register {0..7}, Outbound DMA Descriptor Read Count Register {0..7}, Outbound DMA Descriptor Status FIFO Write Pointer Register {0..7}, Outbound DMA Descriptor Status FIFO Read Pointer Register {0..7}, and Outbound DMA Channel Status Register {0..7} to 0 when INIT is 1 in the Outbound DMA Channel Control Register {0..7}. All other registers retain their value during INIT except under a fundamental reset.
- DMA descriptor processing is initiated when software writes to the Outbound DMA Descriptor Write Count Register {0..7}, and the Tsi721 resets INIT in the Outbound DMA Channel Control Register {0..7} upon this software write.
- When DMA descriptor processing is initiated, the Messaging Engine reads and begins processing the descriptor at the address pointed to by the 64-bit descriptor address.
  - — When DMA descriptor processing is initiated, the RUN bit in the Outbound DMA Channel Status Register {0..7} is set to 1.
- When the DMA channel finishes processing of a descriptor without error:
  - — The pointer to the descriptor is used as part of the descriptor status to update MWr
  - — If the descriptor's IOF (interrupt on finished) is 1, the Tsi721 generates an interrupt
- If the descriptor is not the end of the descriptor list (monitored through Outbound DMA Descriptor Read Count Register {0..7} and Outbound DMA Descriptor Write Count Register {0..7}, then the DMA channel proceeds to process the next descriptor
  - — If the processed descriptor is not a block pointer descriptor, the next descriptor locates at the address that is 16 bytes above the processed descriptor's address.
  - — If the processed descriptor is a block pointer descriptor, the next descriptor locates at the address pointed by the NEXT field of the block pointer descriptor.
  - — The address of the next descriptor is stored in the Outbound DMA Channel Descriptor Pointer Low Register {0..7} and Outbound DMA Channel Descriptor Pointer High Register {0..7}
  - — In actual implementation, the Tsi721 performs descriptor prefetch, and the next descriptor is fetched from a DMA channel's descriptor prefetch FIFO
  - — However, similar address calculation determines the address of the descriptor prefetch
- If the descriptor is the end of the descriptor list, the DMA channel stops the descriptor processing, sets the DONE bit in the Outbound DMA Channel Interrupt Register {0..7}, and clears the RUN bit in the Outbound DMA Channel Status Register {0..7}.
  - — The Outbound DMA Channel Descriptor Pointer Low Register {0..7} and Outbound DMA Channel Descriptor Pointer High Register {0..7} points to the next descriptor to be fetched:
    - – The address of the last successfully processed descriptor is called LADDR.

  - If LADDR is for a type 4 descriptor, then the Outbound DMA Channel Descriptor Pointer Low Register {0..7} and Outbound DMA Channel Descriptor Pointer High Register {0..7} hold the value of LADDR + 16.
  - If LADDR is for a type 5 descriptor, then Outbound DMA Channel Descriptor Pointer Low Register {0..7} and Outbound DMA Channel Descriptor Pointer High Register {0..7} hold the value of NEXT field of the type 5 descriptor.
- If the Outbound DMA Descriptor Write Count Register {0..7} is updated again by software, the DMA channel restarts descriptor processing by fetching the descriptor pointed to by the Outbound DMA Channel Descriptor Pointer Low Register {0..7} and Outbound DMA Channel Descriptor Pointer High Register {0..7}, and then continues to process the next descriptor.

9.2.2.3 DMA Channel Abort on Error

9.2.2.3.1 Errors Causing DMA Channel Abort

The processing of a DMA descriptor by a DMA channel can be aborted when an error is detected during descriptor processing. In this case, all DMA channel specific PCIe errors (ECRC error is not channel specific) and S-RIO errors cause the associated DMA channel to abort, and these errors are listed in the CS field of the Outbound DMA Channel Status Register {0..7}.

9.2.2.3.2 DMA Channel Abort

A DMA channel abort involves the following actions:

1. The DMA channel stops prefetching new descriptors
2. The DMA channel updates the Outbound DMA Channel Interrupt Register {0..7}, and sets the ERROR bit in the same register
3. The DMA channel stops issuing new requests except descriptor status MWr requests from finished descriptors, and waits for all these descriptor status MWr to be sent
4. The Outbound DMA Channel Descriptor Pointer Low Register {0..7} and Outbound DMA Channel Descriptor Pointer High Register {0..7} are undefined
5. The Outbound DMA Descriptor Read Count Register {0..7} stops incrementing
6. The Tsi721 lets all outstanding PCIe/ S-RIO requests/ completions/ responses of the DMA channel to drain through all shared resources, and then discards them.
7. If any outstanding PCIe/ S-RIO completions/ responses of the DMA channel do not arrive due to error, the Tsi721 waits for their completion/ response timeout to occur, then sets the ABORT bit in the Outbound DMA Channel Status Register {0..7}.
8. To recover from a DMA channel abort, software must re-initialize the DMA channel by setting the INIT bit in the Outbound DMA Channel Control Register {0..7}). When this occurs, the Tsi721 clears the ABORT bit in the Outbound DMA Channel Status Register {0..7}, and software clears the ERROR bit in Outbound DMA Channel Interrupt Register {0..7}. Lastly, software writes to the Outbound DMA Descriptor Write Count Register {0..7}.

9.2.2.4 Suspending and Resuming a DMA Channel

- In some applications it is desirable to append descriptors to an active descriptor list; that is, one that is being processed by a DMA channel. The DMA channel suspend and resume capability provides a race-free method to perform this operation.
- A DMA channel abort has higher priority than a DMA channel suspend. If an error is detected while completing a DMA channel suspend, then the DMA channel abort procedure takes over and the DMA channel suspend procedure is forfeited. The operation of a DMA channel can be suspended by writing a 1 to the SUSPEND bit in the Outbound DMA Channel Control Register {0..7}. When a DMA operation is suspended, the following actions occur:
  - The DMA channel stops prefetching new descriptors

    - — The DMA channel normally processes all prefetched descriptors Once all prefetched descriptors are processed, the DMA channel loads the Outbound DMA Channel Descriptor Pointer Low Register {0..7} and Outbound DMA Channel Descriptor Pointer High Register {0..7} to point to the next descriptor to be fetched, and the SUSPENDED bit is set in the Outbound DMA Channel Interrupt Register {0..7}.
    - — If the DMA channel is in the DONE state when suspended – that is, the DMA has completed processing descriptors in a list – then the SUSPENDED bit in the Outbound DMA Channel Interrupt Register {0..7} is immediately set.
        - – Software should wait (through either polling or interrupt) for the SUSPENDED bit in the Outbound DMA Channel Interrupt Register {0..7} to be set before resuming the DMA channel. Violating this rule produces undefined behavior.
        - – When a DMA channel is in the DONE state, the Outbound DMA Channel Descriptor Pointer Low Register {0..7} and Outbound DMA Channel Descriptor Pointer High Register {0..7} point to the next descriptor to be fetched (16+ descriptor address at the end of the descriptor list).
- The Tsi721 generates a status update MWr TLP
- The RUN bit in Outbound DMA Channel Status Register {0..7} is cleared while a DMA is suspended.
- Software can optionally append descriptors to an existing descriptor list while the SUSPENDED bit in the Outbound DMA Channel Interrupt Register {0..7} is set.
    - — Optionally, software can also insert new descriptor blocks by first reading the Outbound DMA Channel Descriptor Pointer Low Register {0..7} and Outbound DMA Channel Descriptor Pointer High Register {0..7}, and saving them as a link in the address – that is, use them to create a Type 5 descriptor – then overwriting the registers with a Type 5 descriptor pointing to a new descriptor block.
- A suspended DMA channel can be resumed by writing to the Outbound DMA Descriptor Write Count Register {0..7} (a suspended DMA channel should not have any errors except ECRC errors; note that unsolicited message responses are per device).
    - — The Messaging Engine clears the SUSPEND bit in the Outbound DMA Channel Control Register {0..7}.
- The Messaging Engine fetches the descriptor pointed by the Outbound DMA Channel Descriptor Pointer Low Register {0..7} and Outbound DMA Channel Descriptor Pointer High Register {0..7} and starts processing the descriptor.

9.2.2.5 Dynamic Chaining of Descriptor Lists

Dynamic chaining of descriptor lists describes a race-free method to append or modify descriptors in an active descriptor list, by suspending and resuming descriptor processing in a DMA channel. For scenarios where a suspend/resume operation is not desired (for example, to improve DMA performance), this section presents an alternative method to performing dynamic appending of descriptors to a descriptor list.

- Descriptors can be appended to an active descriptor list – that is, a descriptor list which a DMA channel is currently processing – by appending new descriptors to the last descriptor.
- After descriptors are appended to an active descriptor list, software must write the Outbound DMA Descriptor Write Count Register {0..7} with a new value.
    - — If the DMA channel has not finished processing the last descriptor in the original list when the Outbound DMA Descriptor Write Count Register {0..7} is updated, the DMA channel continues processing descriptors normally, including the newly appended descriptors.
    - — If the DMA channel completed descriptor processing of the last descriptor in the original list when the Outbound DMA Descriptor Write Count Register {0..7} is updated – that is, the DMA channel is in DONE state after processing the last descriptor in the list – the DMA channel restarts descriptor processing:
        - – The Outbound DMA Channel Descriptor Pointer Low Register {0..7} and Outbound DMA Channel Descriptor Pointer High Register {0..7} point to the next descriptor to be fetched.
        - – The Messaging Engine fetches the descriptor pointed by the Outbound DMA Channel Descriptor Pointer Low Register {0..7} and Outbound DMA Channel Descriptor Pointer High Register {0..7}, and starts processing the next descriptor.

9.2.3 TLP Attribute and Traffic

The Messaging Engine sends TLPs with Relaxed Ordering (RO) header bit cleared and Traffic Class (TC) header bits cleared. For No Snoop header bit, it is from the associated PCIe MAC bit.

9.2.4 DMA Outstanding PCIe Requests

- A total of 32 DMA outstanding requests, including both DMA data MRd and DMA descriptor MRd, are supported:
  - They are shared among all inbound and outbound DMA channels
  - 32 outstanding tags are per device, shared by the SR2PC, BDMA, and SMSG modules
- The performance of the Block DMA Engine improves when the DMA issues few read requests when processing a descriptor, and each read request transfers a large amount of data. Note that the DMA addressing specified in the descriptor determines the boundaries at which the DMA channel breaks requests. The Block DMA Engine breaks requests at every 4 KB of data transfer.

9.2.5 Descriptor Prefetching

When the amount of data moved by data transfer descriptors is small, the overhead in fetching DMA descriptors from memory between data transfer operations may limit performance. To overcome this overhead limitation, the DMA channel supports descriptor prefetching. The Messaging Engine can prefetch up to 32 descriptors per DMA channel.

- When the fill level of a DMA channel’s descriptor prefetch FIFO is less than 25, the Messaging Engine issues descriptor MRd to fetch up to 8 descriptors before previous descriptors have been completed.
- The DMA channel queues prefetched descriptors until they are processed.
  - Prefetched descriptors are discarded when the DMA channel is aborted
- Descriptor prefetching stops when the end of a descriptor list is reached.
  - Tsi721 descriptor prefetch never passes the last descriptor of a descriptor list
  - Tsi721 descriptor prefetch never crosses a 4-KB boundary

9.3 Inbound Messaging

9.3.1 Overview

The Inbound Message Engine implements 8 Rx queues. An inbound message packet is routed to a Rx queue based on its mbox fields, while messages are mapped to Rx queues 0/1/2/3 based on a mbox header field value of 0/1/2/3.

Each Rx queue has 16 contexts, so that multiple messages can be reassembled correctly without head of line blocking. The Inbound Messaging Engine implements 8 dedicated DMA channels, with one DMA channel per Rx queue. DMA descriptors and associated DMA data buffers reside in PCIe side main memory. The DMA data buffer is allocated in 4-KB or 8-KB blocks.

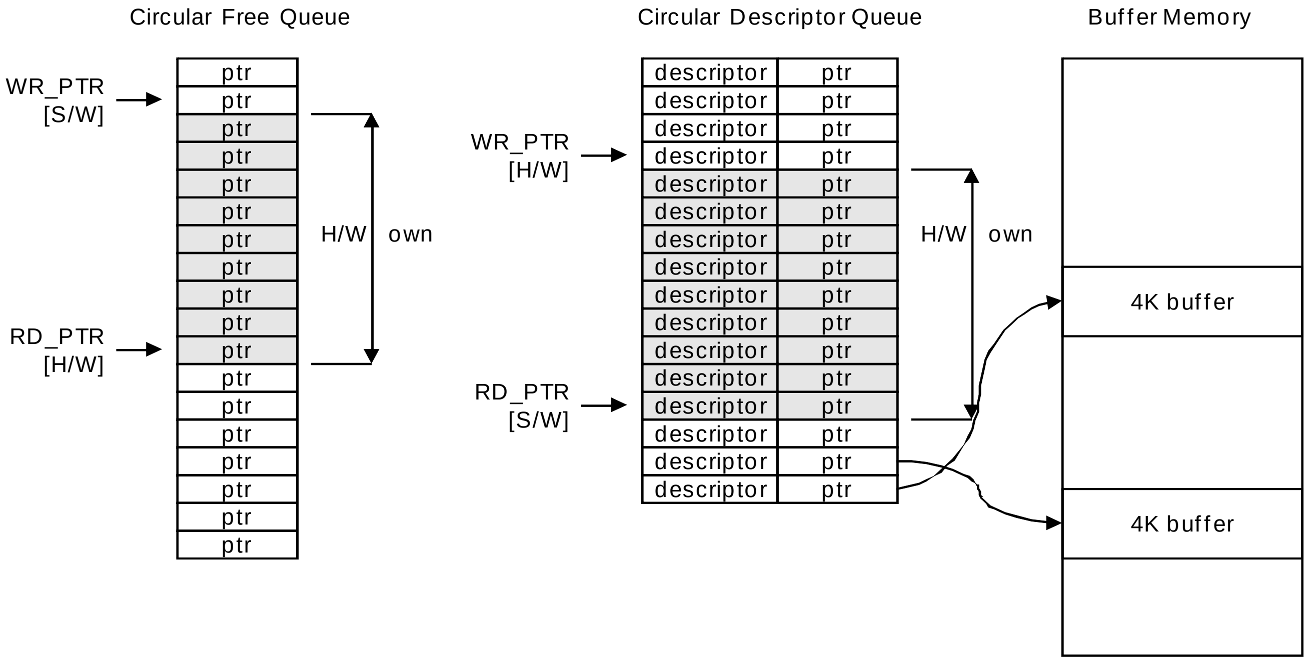
9.3.2 Data Structures in Main Memory

Software is required to set up several data structures in memory for the purpose of managing inbound S-RIO messages (see Figure 36). In this case, for each inbound DMA channel, software provides free pointers through a circular free queue, Tsi721 stores inbound message data into data buffers pointed by free pointers, and Tsi721 stores inbound message descriptors into a circular descriptor queue.

A circular free list is set up using Inbound Circular Free Queue Lower Base Register {0..7}, Inbound Circular Free Queue Upper Base Register {0..7}, and Inbound Circular Free Queue Size Register {0..7}, which defines the size and location of the memory window for the circular free queue. The ptr field inside the circular free queue points to data buffers used by the Tsi721 to store inbound message data. The base address must be 64-byte aligned. The WR_PTR in Inbound Circular Free Queue Write Pointer Register {0..7} and RD_PTR in Inbound Circular Free Queue Read Pointer Register {0..7} define what locations have new free pointers. The Tsi721 identifies the free queue as empty when WR_PTR is equal to RD_PTR. New free pointers are in the locations indicated as hardware ownership. Each 64-bit pointer points to start of the DMA data buffer, where arbitrary byte alignment is allowed. In this case, as a data buffer will be filled with a receive message in the future with payload size unknown at the time of buffer allocation, a fixed buffer size (for example, 8 KB) should be used. Inbound DMA data buffers can cross 4-KB boundaries. If a buffer pointer is not 4-KB aligned and 4-KB received messages are allowed, then an 8-KB buffer size is recommended to avoid corrupting messages due to a message straddling two data buffers. However, if applications can guarantee maximum received message size + buffer pointer offset relative to a 4-KB boundary never exceed a 4-KB boundary, then 4-KB data buffer size is sufficient. The application is required to provide proper DMA data buffers through proper free buffer pointers.

The Tsi721 increments the RD_PTR as it pulls free pointers off the circular free list. Software increments the WR_PTR as it places new free addresses on the circular free list. When the descriptor queue is written after receiving a new message, an interrupt is generated using Inbound DMA Channel Interrupt Register {0..7}.

Figure 36: Inbound Data Structures (ptr in Descriptor Queue is Part of Descriptor)



In addition, a circular descriptor queue is set up by the Inbound Descriptor Queue Lower Base Register {0..7}, Inbound Descriptor Queue Upper Base Register {0..7}, and Inbound Circular Descriptor Queue Size Register {0..7}, which defines the size and location of the memory window for the circular descriptor queue. The base address must be 64-byte aligned. For the Tsi721, the WR_PTR in Inbound Circular Descriptor Queue Write Pointer Register {0..7} and RD_PTR in Inbound Circular Descriptor Queue Read Pointer Register {0..7} define which locations have empty locations; that is, Tsi721 sees descriptor queue as full when (WR_PTR + 1) MOD Inbound Circular Descriptor Queue Size Register {0..7} becomes equal to RD_PTR. Each descriptor has 16 bytes and includes a 64-bit buffer pointer. The descriptor is generated by the Tsi721 on a per message basis; Tsi721 writes the descriptor as messages that are fully assembled in a DMA data buffer. The device increments the WR_PTR as it adds new descriptors. Software increments the RD_PTR as it processes the queue. When software finishes with the DMA data buffer, the pointer can be returned to the circular free queue.

9.3.3 DMA Descriptors

The Tsi721 implements one DMA channel per Rx queue. A DMA channel operates by fetching free pointers from the inbound circular free queue, reassembling messages using DMA data buffers pointed to by free pointers, and writing a DMA descriptor into the inbound circular descriptor queue when it finishes a message.

9.3.3.1 Descriptor Ordering Rules

For each DMA channel, the Inbound Messaging Engine generates a descriptor when it finishes reassembling a message, and descriptors are written into the inbound circular descriptor queue in FIFO order (message reassembly order). This results in the following order rules:

- Within a descriptor, message segments are sent in arrival order
- Among descriptors of the same DMA channel, descriptors are generated in message completion order
  - — A message finishing reassembly first will have its descriptor written earlier into the inbound circular descriptor queue
- Among descriptors of different DMA channels, descriptors are unordered

9.3.3.2 Data Transfer Descriptor

The data transfer descriptor has a DTYPE of 6. It has a size of 64 bytes (cache line size) The format of a data transfer descriptor is displayed in Figure 37, and its fields are described in Table 62.

- The format is displayed in little-endian.
- A field marked as R is reserved.

Figure 37: Data Transfer DMA Descriptor Format – DTYPE 6

	bte 3								byte 2								byte 1								byte 0							
DW	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	DTYPE			R	R	R						TT		PRIO		CRF	DEVID															
1	HO	CS[3:0]				R			MBOX		XMBOX				LETTER		SSIZE				BCOUNT[11:3]									R		
2	BUFFER_PTR[31:0]																															
3	BUFFER_PTR[63:32]																															
4	R																															
5	R																															
6	R																															
7	R																															
8	R																															
9	R																															
10	R																															
11	R																															
12	R																															
13	R																															
14	R																															
15	R																															

Table 62: Data Transfer DMA Descriptor Format – DTYPE 6

Field	dword	Bit Position	Description
DEVID15:0]	0	15:0	S-RIO source deviceID
CRF	0	16	S-RIO critical request flow
PRIO[1:0]	0	18:17	S-RIO priority. Valid values are 0–2 (for request flows), as PRIO of 3 is for S-RIO response per part 6 of the *RapidIO Specification (Rev. 2.1)*.

Table 62: Data Transfer DMA Descriptor Format – DTYPE 6 *(Continued)*

Field	dword	Bit Position	Description
TT	0	20:19	S-RIO transport type. • 0b00 = 8-bit deviceID • 0b01 = 16-bit deviceID • Others = Reserved
DTYPE[2:0]	0	31:29	Descriptor type. This field encodes the type of descriptor and must be set to 0x6 in DMA descriptors. • 0x6 = Data transfer DMA descriptor • Others = Reserved
BCOUNT[11:3]	1	11:3	Byte count. This is the total number of bytes to transfer in units of 8 bytes. • 0b0_0000_0000 = 512 (4 KB) • 0b0_0000_0001 = 1 (8 bytes) • 0b0_0000_0010 = 2 (16 bytes) ... • 0b1_1111_1111 = 511 (4 KB-8 bytes)
SSIZE[3:0]	1	15:12	S-RIO message standard segment size
LETTER[1:0]	1	17:16	S-RIO message letter
XMBOX[3:0]	1	21:18	S-RIO message XMBOX
MBOX[1:0]	1	23:22	S-RIO message MBOX
CS[3:0]	1	30:27	Message completion status. This value is written by the Tsi721 when it completes receiving a message, or when it detects an error for a message. • 0b0001 = Successful receipt of a message • 0b1000 = S-RIO request timeout occurred • Others = Reserved
HO	1	31	Hardware own. • 0 = Software owns this descriptor location • 1 = Tsi721 owns this descriptor location Software must clear this bit after finishing processing a descriptor. Software should use this bit instead of WR_PTR to determine which descriptor entry is valid inside the descriptor queue.
BUFFER_PTR[31:0]	2	31:0	PCIe address lower. Lower 32-bits of 64-bit PCIe address.
BUFFER_PTR[63:32]	3	31:0	PCIe address upper. Upper 32-bits of 64-bit PCIe address.

- One DMA data transfer is for one S-RIO message, and may consist of multiple PCIe data MWr and multiple S-RIO message segments
- A descriptor is generated when a message is finished reassembling, where a message is completed when either the data transfer operation completes or when an error is detected.
  - The 64-bit PCIe address, BUFFER_PTR[63:0], may have any byte alignment
  - If the PCIe address is below 4 GB, then a MWr TLP with a 32-bit address is generated. If the address is above 4 GB, then a MWr TLP with a 64-bit address is generated
  - Multiple MWr are generated due to the following:
    - A message's SSIZE exceeds MAX_PAYLOAD_SIZE
    - A message has multiple message segments
    - A message's payload crosses a 4-KB boundary

9.3.4 DMA Descriptor Processing

9.3.4.1 Descriptor Processing

Software initializes a DMA channel by programming the Inbound Circular Free Queue Lower Base Register {0..7}, Inbound Circular Free Queue Upper Base Register {0..7}, Inbound Circular Free Queue Size Register {0..7}, Inbound Descriptor Queue Lower Base Register {0..7}, and Inbound Descriptor Queue Upper Base Register {0..7} (these registers must be static after a channel has been initialized until the channel is re-initialized again), then setting the INIT bit in the Inbound DMA Channel Control Register {0..7}.

- Tsi721 resets the Inbound Circular Free Queue Read Pointer Register {0..7}, Inbound Circular Free Queue Write Pointer Register {0..7}, Inbound Circular Descriptor Queue Read Pointer Register {0..7}, and Inbound Circular Descriptor Queue Write Pointer Register {0..7} to 0 when the INIT bit is 1 in the Inbound DMA Channel Control Register {0..7}.
- DMA descriptor processing is initiated when software writes to the Inbound Circular Free Queue Write Pointer Register {0..7}, upon which the Tsi721 then resets INIT. Thereafter,
  - The Messaging Engine prefetches free pointers from the inbound free queue
  - When one or more message segments is received, the Messaging Engine starts to reassemble a message using DMA data buffer pointed by a prefetched free pointer
  - When a message reassembling is finished, the Messaging Engine generates a descriptor and sends it to an inbound descriptor queue through a descriptor MWr TLP
  - If receiving a message segment for an uninitialized inbound DMA channel, the message segment is discarded and an ERROR response is generated (see events in Inbound DMA Channel Interrupt Register {0..7} and Inbound DMA Channel Status Register {0..7})
  - If receiving a message segment for a suspended (torn down) inbound DMA channel, the message segment is discarded and an ERROR response is generated
  - If receiving a message segment without a free receiving context, the message segment is discarded and a RETRY response is generated
  - If receiving a message segment without a free pointer, the message segment is discarded and a RETRY response is generated
  - If receiving a message segment when the inbound descriptor queue is full, the message segment is discarded and a RETRY response is generated

9.3.4.2 DMA Channel Abort on Error

9.3.4.2.1 Errors Causing a DMA Channel Abort

The processing of a DMA descriptor by a DMA channel can be aborted when an error is detected during descriptor processing. In this case, all DMA channel specific PCIe errors (ECRC error is not channel specific) cause the associated DMA channel to abort, and these errors are listed in the CS field of the Inbound DMA Channel Status Register {0..7}. However, none of S-RIO errors will cause an associated DMA channel to abort because inbound messages are received from potentially multiple far-end endpoints.

9.3.4.2.2 DMA Channel Abort

A DMA channel abort involves the following actions:

1. The DMA channel stops prefetching new free pointers
2. The DMA channel updates the Inbound DMA Channel Interrupt Register {0..7}, sets the PC_ERROR bit in the register, and generates an interrupt
3. After the interrupt is generated, software launches the DMA channel suspend procedure by writing a 1 to the SUSPEND bit in the Inbound DMA Channel Control Register {0..7}
   - The DMA channel normally processes all descriptors that are already active (called outstanding descriptors)
   - Once all outstanding descriptors have been completed, the DMA channel loads the Inbound Circular Descriptor Queue Write Pointer Register {0..7} to point to the next descriptor after the last processed descriptor
   - All remaining and subsequently arrived inbound messages for this DMA channel are discarded and ERROR responses are generated.
4. If any outstanding PCIe completions (due to outstanding free pointer prefetch MRd) of the DMA channel do not arrive due to error, the Tsi721 waits for their completion timeout to occur
5. After the timeout error, the Tsi721 sets the ABORT bit in the Inbound DMA Channel Status Register {0..7} and SUSPENDED bit in the Inbound DMA Channel Interrupt Register {0..7}, and generates a SUSPENDED interrupt
6. To recover from a DMA channel abort, software must re-initialize the DMA channel (set INIT bit of the Inbound DMA Channel Control Register {0..7}, upon which, the ABORT bit in Inbound DMA Channel Status Register {0..7} is cleared; then clear the PC_ERROR bit in the Inbound DMA Channel Interrupt Register {0..7}; lastly, write to the Inbound Circular Free Queue Write Pointer Register {0..7}).

9.3.4.3 Suspending and Resuming a DMA Channel

- The DMA channel suspend and resume capability is provided for an application to suspend (tear down) a DMA channel.
  - Received inbound message segments for a suspended inbound DMA channel are discarded and ERROR responses are generated
- A DMA channel abort has a higher priority than a DMA channel suspend. If an error is detected while suspending a DMA channel, then the DMA channel abort procedure takes over and the DMA channel suspend procedure is canceled
- The operation of a DMA channel can be suspended by writing a 1 to the SUSPEND bit in the Inbound DMA Channel Control Register {0..7}. When a DMA operation is suspended, the following actions occur:
  - The DMA channel stops prefetching new free pointers
  - The DMA channel normally processes all descriptors that are already active (called outstanding descriptors)
  - All remaining unused free pointers in the free pointer prefetch FIFO are discarded (message may never arrive)

  - Once all outstanding descriptors have been completed, the DMA channel loads the Inbound Circular Descriptor Queue Write Pointer Register {0..7} to point to the next descriptor after the last processed descriptor, and the SUSPENDED bit is set in the Inbound DMA Channel Interrupt Register {0..7} after all prefetch free pointers have been received.
  - All remaining and subsequently arrived inbound messages for the DMA channel are discarded and ERROR responses are generated.
- The RUN bit in the Inbound DMA Channel Status Register {0..7} is cleared while a DMA is suspended.
- A suspended DMA channel can be resumed by writing to the Inbound Circular Free Queue Write Pointer Register {0..7}.
  - Software must wait until all remote endpoints with outstanding messages for this channel have either received all responses or timed out before resuming a suspended channel.
  - Resume starts regardless of the value in the Inbound Circular Free Queue Write Pointer Register {0..7}
  - The SUSPENDED bit is then cleared in the Inbound DMA Channel Interrupt Register {0..7}, and the Tsi721 starts a free pointer prefetch from the Inbound Circular Free Queue Read Pointer Register {0..7}

9.3.5 TLP Attribute and Traffic

The Messaging Engine sends TLPs with RO header bit cleared, TC header bits cleared, and No Snoop bit from the associated PCIe MAC bit.

9.3.6 DMA Outstanding PCIe Requests

The Tsi721 supports one outstanding free pointer MRd per DMA channel.

9.4 Messaging Engine Scheduling

9.4.1 S-RIO Egress Scheduling

The Tsi721 uses the following S-RIO egress scheduling algorithm:

1. Within each outbound DMA channel, S-RIO egress scheduling is S-RIO ordering rule based (see Descriptor Ordering Rules).
2. Among winners of each outbound DMA channel from step 1, round-robin scheduling is applied.

9.4.2 PCIe Egress Scheduling

The Tsi721 uses the following PCIe egress scheduling algorithm:

1. Simple round-robin scheduling for data MRd TLPs among outbound DMA channels
2. Simple round-robin scheduling between winners of step 1 and inbound message data MWr TLPs, as inbound message data MWr TLPs are scheduled from a FIFO that is shared among all inbound DMA channels
3. Other types of Messaging Engine originated TLPs, such as outbound descriptor prefetch MRd, outbound descriptor status MWr, inbound free pointer prefetch MRd, and inbound descriptor MWr, and so on, are scheduling using a combination of simple round-robin among inbound/outbound DMA channels and strict priority scheduling, where their priorities are higher than the winner of step 2.
4. The winner of step 3 participates in the Mapping Engine scheduling as discussed in Mapping Engine Scheduling.

10. Block DMA Engine

Topics discussed include the following:

- Overview
- Functional Overview
- Data Transfer and Addressing
- DMA Descriptors
- DMA Descriptor Processing
- TLP Attribute and Traffic
- DMA Outstanding PCIe Requests
- Descriptor Prefetching
- DMA Limitations
- Block DMA Engine Scheduling

10.1 Overview

- Supports one Block DMA Engine with 8 DMA channels.
- DMA descriptors reside in PCIe side main memory
  - — Descriptors are S-RIO native
- Each DMA channel can perform both read and write
  - — Read or write selection is per descriptor
- Enforces S-RIO ordering rules for DMA descriptors
  - — Compatible with PCIe ordering rules for DMA descriptors
- Implements 8-KB DMA write reassembly queue per DMA channel
- Store and forward for write reassembly queue

10.2 Functional Overview

Each DMA channel provides a high performance method of moving data. The 8 DMA channels operate independently. A DMA channel operates by reading descriptors from memory, then performing the operation outlined by the descriptor. Complex data movement operations can be implemented by linking DMA descriptors together to form a list of descriptor blocks, where descriptors within each descriptor block are in contiguous memory address space.

DMA descriptors reside in main memory connected through the PCIe Interface. For a DMA descriptor indicating a S-RIO NREAD or S-RIO maintenance read, DMA data source is a memory or main memory of a far end device. The far end device is either an S-RIO endpoint or a root complex connected to a remote Tsi721. The S-RIO deviceID of the S-RIO endpoint or remote Tsi721, S-RIO address/config_offset for DMA data, and other S-RIO header attributes, are part of the DMA descriptor. DMA data destination is main memory, and destination address is the PCIe address in descriptor.

- S-RIO address is source address, denoted as SADDR
- PCIe address is destination address, denoted as DADDR
- A DMA channel prefetches DMA descriptors, issues S-RIO NREADs or S-RIO maintenance reads to fetch data from far end, then transforms S-RIO responses into PCIe MWr to store data into main memory

For a DMA descriptor indicating an S-RIO NWRITE/SWRITE/NWRITE_R, or a S-RIO maintenance write, DMA data source is main memory, source address is PCIe address in DMA descriptor. DMA data destination is a memory or main memory of a far end device. The far end device is either an S-RIO endpoint or a root complex connected to a remote Tsi721. The S-RIO device ID of the S-RIO endpoint or remote Tsi721, S-RIO address/config_offset for DMA data, and other S-RIO header attributes, are part of DMA descriptor.

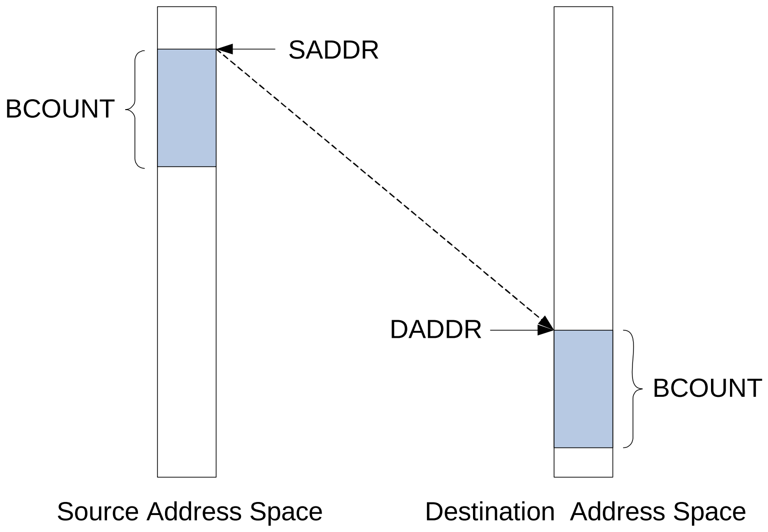
- PCIe address is source address, denoted as SADDR
- Address is destination address, denoted as DADDR
- A DMA channel prefetches DMA descriptors, issues PCIe MRd to fetch data from main memory, and then transforms PCIe data completions into NWRITE/SWRITE/NWRITE_R, or S-RIO maintenance writes to send data to far end

10.3 Data Transfer and Addressing

The Block DMA Engine supports multiple types of addressing modes. Two of the more common modes supported by the block include linear addressing and constant addressing.

The simplest form of DMA addressing is linear addressing, whereby a sequential block of data consisting of BCOUNT bytes is transferred from a starting address, SADDR, to a destination address, DADDR (see Figure 38). Constant addressing, in comparison, is used to repeatedly read/write the same address(es), and is often used when moving data to/from a memory-mapped FIFO (see Figure 40).

Figure 38: Linear Addressing



The addressing operations implemented by a DMA channel are displayed in Figure 39, and the associated parameters are described in Table 63. Although both the source and destination addressing algorithms are the same, different parameter initialization may cause different behavior:

- An addressing operation completes execution when the byte count is exhausted.

Figure 39: DMA Channel Addressing

```
addr = SADDR;
for (j=0; j<BCOUNT; ++j) {
        data = memRead(addr);
        if (SSSIZE != 0 && (j+1) % SSSIZE == 0) {
            addr = addr + SSDIST + 1;
        } else {
            addr = addr + 1;
        }
}
```

(a) Source Addressing

```
addr = DADDR;
for (j=0; j<BCOUNT; ++j) {
        memWrite(addr,data);
        if (DSSIZE != 0 && (j+1) % DSSIZE == 0) {
            addr = addr + DSDIST + 1;
        } else {
            addr = addr + 1;
        }
}
```

(b) Destination Addressing

Table 63: DMA Channel Addressing Parameters

Global Parameters		
BCOUNT	Byte Count	Total number of bytes to transfer
Source Addressing Parameters		
SADDR	Source Address	Starting source memory address. For DMA reads, it is an S-RIO address; for DMA writes, it is a PCIe address.
SSSIZE	Source Stride Size	Amount of data bytes to transfer in each source stride (a value of zero indicates an infinite stride size)
SSDIST	Source Stride Distance	Distance between two adjacent source strides (end to beginning) in bytes (two's complement representation of a positive or negative number)
Destination Addressing Parameters		
DADDR	Destination Address	Starting destination memory address. For DMA reads, it is a PCIe address; for DMA writes, it is an S-RIO address.
DSSIZE	Destination Stride Size	Amount of data bytes to transfer in each destination stride (a value of zero indicates an infinite stride size)
DSDIST	Destination Stride Distance	Distance between two adjacent destination strides (end to beginning) in bytes (two's complement representation of a positive or negative number)

10.3.1 Linear Addressing Example

Linear addressing occurs from a source address to a destination address, as displayed in Figure 38. Table 64 shows the parameters required to transfer 1 KB of data from a source address of 0x0100_0000 to a destination address of 0x0200_0001.

- BCOUNT is set to 1024 indicating the total number of bytes to transfer
- SADDR is set to the source address 0x0100_0000, and DADDR is set to the destination address 0x0200_0001
- Since strides are not used in this example, SSSIZE and DSSIZE are set to zero (all stride distances, SSDIST and DSDIST, also are do not care, and are displayed as zero). A value of zero indicates an infinite stride (only one stride). This results in the inner loop executing in Figure 39 until the byte count is exhausted.

Table 64: Linear Addressing DMA Example

BCOUNT	1024		
SADDR	0x0100_0000	DADDR	0x0200_0001
SSSIZE	0	DSSIZE	0
SSDIST	0	DSDIST	0

10.3.2 Constant Addressing Example

By setting the stride distance to a negative value, the DMA Block Engine can perform constant addressing. Constant addressing is used to repeatedly read/write the same address(es), and is often used when moving data to/from a memory-mapped FIFO.

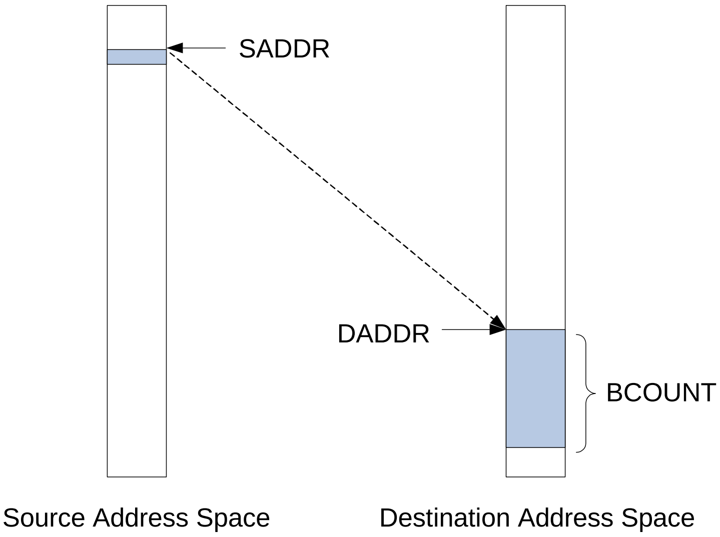
The parameters in Table 65 show an example where constant addressing is used to repeatedly read dword data from a memory-mapped FIFO and transfer (DMA) the data to a destination buffer. In this example, 1024 bytes are read a dword at a time from the address 0x0100_0000 (see also Figure 40).

Table 65: Constant Addressing DMA Example

BCOUNT	1024		
SADDR	0x0100_0000	DADDR	0x0200_0001
SSSIZE	4	DSSIZE	0
SSDIST	-4	DSDIST	0

Note that the SSSIZE value determines the size of the region from which data is read. In the example, SSSIZE is set to 4 to perform 256 read operations from the same dword, for a total of 1024 bytes transferred; that is, BCOUNT is set to 1024. Similarly, the DSSIZE value controls the size of the region to which data is written. In the above example, DSSIZE is set to 0 to indicate an infinite destination stride as the destination addressing is linear.

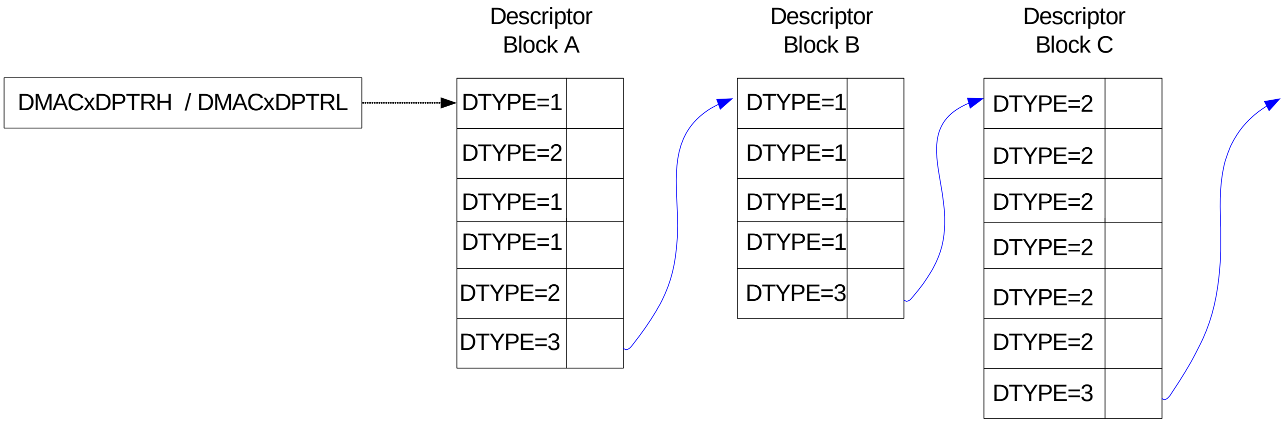
Figure 40: Constant Addressing DMA Example



10.4 DMA Descriptors

- A DMA channel operates by reading descriptors from memory and performing the specified processing.
- The descriptor address must not be all zeroes because this value is reserved to mark an invalid entry in a descriptor status FIFO residing in PCIe side main memory.
- Complex data movement operations, such as scatter gather, can be used by linking multiple descriptor blocks together into a descriptor list, where descriptors within each descriptor block are in contiguous memory space. Figure 41 shows a descriptor list, where DTYPE denotes the descriptor type.

Figure 41: DMA Descriptor List



This document is confidential and is subject to an NDA.

The Block DMA Engine uses three types DMA descriptors (see Table 66).

Table 66: DMA Descriptor Types

DTYPE	Name	Descriptor
DTYPE 1	Data transfer descriptor	This is used for general data movement with stride control parameters.
DTYPE 2	Immediate data transfer descriptor	This is used for small chunks of data movement (up to 16 bytes) with data embedded inside the descriptor itself.
DTYPE3	Block pointer descriptor	This is used to link descriptor blocks.

- Each DMA channel has only one DMA descriptor list.
- One DMA descriptor list can mix all three types of descriptors.
- All descriptors have fixed size of 32 bytes. they must be 32-byte aligned; otherwise, Tsi721 is behavior is undefined.
- A descriptor block consists of multiple descriptors in contiguous memory space:
  - The last descriptor of a block can have any DTYPE
  - A descriptor of DTYPE 3 ends a descriptor block
  - All other descriptors of a block have either DTYPE = 1 or 2
  - Each 4-KB page that contains a descriptor must be prefetchable (see Descriptor Prefetching). Violation of this rule may cause undefined side effects due to descriptor prefetch

10.4.1 Ending of Descriptor List

The last descriptor of a descriptor list is not directly encoded in the descriptor. Instead, software and the Tsi721 communicate the last descriptor through two rolling counters per DMA channel:

- Descriptor write count – A 32-bit non-saturating running count of generated descriptors by software
- Descriptor read count – A 32-bit non-saturating running count of prefetched descriptors (these descriptors may not have been processed) by the Tsi721

These two counts are reset to 0 during a DMA channel initialization. Tsi721 implements its version of descriptor read and write counts using the DMA Descriptor Read Count Register {0..7} and DMA Descriptor Write Count Register {0..7}. Software informs the Tsi721 about the descriptor write count by updating DMA Descriptor Write Count Register {0..7} through PCIe MWr. Under normal operations (no error occurred nor DMA channel abort/ initialization/ suspending), the Tsi721 continues prefetching and processing descriptors until the DMA Descriptor Read Count Register {0..7} is equal to the DMA Descriptor Write Count Register {0..7} (DMA Descriptor Read Count Register {0..7} indicates the descriptor prefetch progress for all valid types of descriptors). The Tsi721 then assumes that the last processed descriptor is the last one of the descriptor list, and stops descriptor processing until software updates the descriptor write count. When software updates the DMA Descriptor Write Count Register {0..7}, the Tsi721 automatically starts descriptor prefetching and processing again.

Software and the Tsi721 reset these two counts during a DMA channel initialization and.

Software can track the descriptor read count by reading the DMA Descriptor Read Count Register {0..7} or by reading the descriptor status FIFO (see DMA Descriptor Status). To achieve high throughput, software should monitor the descriptor status (either through polling or interrupt), timely generate enough descriptors ahead of the Block DMA Engine, and update the Tsi721 about the descriptor write count.

10.4.2 Descriptor Ordering Rules

The Block DMA Engine generates PCIe MRd to fetch descriptors within the PCIe side main memory. Even though CplD for descriptor MRd and data MRd arrives at the Tsi721 in an undefined order, the device enforces the S-RIO ordering rules for descriptors of a specific DMA channel. Descriptors among different DMA channels are not ordered and round-robin scheduling among DMA channels is used.

The ordering rules consist of the following:

- Within a descriptor involving NWRITE/ SWRITE/ NWRITE_R, if more than one NWRITE/ SWRITE/ NWRITE_R S-RIO packet is generated, the packets are sent in increasing S-RIO address order.
- Within a descriptor involving NREAD, if more than one NREAD S-RIO packet is generated, the packets are sent in increasing S-RIO address order.
- Among descriptors with RTYPE NWRITE/ SWRITE/ NWRITE_R/maintenance write of the same DMA channel, the Block DMA Engine generates S-RIO request packets in order.
    — Within the Block DMA Engine, generated NWRITE/ SWRITE/ NWRITE_R/maintenance write S-RIO packets from a newer descriptor are not allowed to pass those from an older descriptor of the same DMA channel. However, if descriptors of a specific DMA channel have different prio/crf fields, the S-RIO MAC can reorder these packets under blocking conditions since they are assigned S-RIO prio/crf according to prio/crf fields of associated descriptors.
- Among descriptors with RTYPE NREAD/maintenance read of the same DMA channel, the Block DMA Engine generates S-RIO request packets in order.
    — Within the Block DMA Engine, generated NREAD/ maintenance read S-RIO packets from a newer descriptor are not allowed to pass those from an older descriptor of the same DMA channel. However, if descriptors of a specific DMA channel have different prio/crf fields, the S-RIO MAC can reorder these packets under blocking since they are assigned S-RIO prio/crf according to prio/crf fields of associated descriptors.
- Under blocking conditions, S-RIO request packets from newer descriptors with RTYPE NWRITE/ SWRITE/ NWRITE_R/maintenance write can pass those from older descriptors with RTYPE NREAD/maintenance read of the same DMA channel.
    — This can occur only if descriptors with RTYPE NWRITE/ SWRITE/ NWRITE_R/maintenance write are assigned a higher S-RIO prio than descriptors with RTYPE NREAD/maintenance read.
- Received S-RIO responses for generated NREAD/ maintenance read of a descriptor of the same DMA channel are translated into PCIe MWr and sent in arrival order. S-RIO responses, however, have an undefined order.
- Received S-RIO responses for generated NWRITE_R/maintenance write of a descriptor are terminated inside the Tsi721 (for more information, see DMA Descriptor Status).

To support PCIe ordering rules, descriptors with RTYPE of NREAD/ maintenance read must be assigned prio/crf of 0/0, while descriptors with RTYPE of NWRITE/SWRITE/NWRITE_R/ maintenance write must be assigned prio/crf of 2/0.

10.4.3 Data Transfer Descriptor

⚠ The operation of the Tsi721 is undefined after processing a descriptor with reserved values in any descriptor field.

The data transfer descriptor has a DTYPE of 1. The format of a data transfer descriptor is displayed in Figure 42, and its fields are described in Table 67. The format is displayed in little-endian. Fields marked as R are reserved.

Figure 42: Data Transfer DMA Descriptor Format

<table>
<tr><td>``</td><td colspan="8">bte 3</td><td colspan="8">byte 2</td><td colspan="8">byte 1</td><td colspan="8">byte 0</td></tr>
<tr><td>DW</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td colspan="3">DTYPE</td><td>R</td><td>IOF</td><td>R</td><td>R</td><td colspan="2">R</td><td colspan="4">RTYPE[3:0]</td><td colspan="2">PRIO</td><td>CRF</td><td colspan="16">DEVID[15:0]</td></tr>
<tr><td>1</td><td colspan="2">RADDR[1:0]</td><td colspan="2">R</td><td colspan="2">TT[1:0]</td><td colspan="26">BCOUNT[25:0]</td></tr>
<tr><td>2</td><td colspan="8">RADDR[33:26]/HOP_CNT[7:0]</td><td colspan="24">RADDR[25:2]/CONFIG_OFFSET[23:0]</td></tr>
<tr><td>3</td><td colspan="32">RADDR[65:34]</td></tr>
<tr><td>4</td><td colspan="32">BUFFER_PTR[31:0]</td></tr>
<tr><td>5</td><td colspan="32">BUFFER_PTR[63:32]</td></tr>
<tr><td>6</td><td colspan="16">DSDST[15:0]</td><td colspan="16">SSDST[15:0]</td></tr>
<tr><td>7</td><td colspan="8">R</td><td colspan="12">DSSIZE[11:0]</td><td colspan="12">SSSIZE[11:0]</td></tr>
</table>

Table 67: Data Transfer DMA Descriptor Format

Field	dword	Bit Position	Description
DEVID15:0]	0	15:0	S-RIO destination ID.
CRF	0	16	S-RIO critical request flow.
PRIO[1:0]	0	18:17	S-RIO priority. Valid values are 0–2 (for request flows), as PRIO of 3 is for S-RIO response per part 6 of the *RapidIO Specification (Rev. 2.1)*.
RTYPE[3:0]	0	22:19	S-RIO transaction type. • 0b0000 = NREAD • 0b0001 = Last S-RIO packet uses NWRITE_R; all other S-RIO packets use NWRITE or SWRITE • 0b0010 = All S-RIO packets use NWRITE or SWRITE • 0b0011 = All S-RIO packets use NWRITE_R • 0b0100 = Maintenance read • 0b0101 = Maintenance write • Others = Reserved
IOF	0	27	Interrupt on finished. 1 = Tsi721 will generate an interrupt when it finishes processing this descriptor, and the IOF bit is set in the corresponding DMA Channel Interrupt Register {0..7}.
DTYPE[2:0]	0	31:29	Descriptor Type. This field encodes the type of descriptor. • 0x1 = Data transfer DMA descriptor

Table 67: Data Transfer DMA Descriptor Format *(Continued)*

Field	dword	Bit Position	Description
BCOUNT[25:0]	1	25:0	Byte Count. The total number of bytes to transfer. • 0x0000000 = 64 MB • 0x0000001 = 1 byte • 0x0000002 = 2 bytes • ... • 0x3FFFFFF = 64M -1 bytes If RTYPE is maintenance read or write, BCOUNT must be 4 bytes
TT[1:0]	1	27:26	S-RIO transport type. • 0b00 = 8-bit deviceID • 0b01 = 16-bit deviceID • Others = Reserved
RADDR[1:0]	1	31:30	S-RIO address bits 1:0 in little-endian format. Valid if RTYPE is not maintenance read/ write.
RADDR[33:2]	2	31:0	S-RIO address bits 33:2 in little-endian format. Valid if RTYPE is not maintenance read/ write.
CONFIG_ OFFSET[23:0]	2	23:0	S-RIO configure offset address in little-endian format when RTYPE is maintenance read or write. CONFIG_OFFSET[1:0] must be all zeroes because maintenance reads or writes must have a 32-bit aligned address.
HOP_CNT[7:0]	2	31:24	S-RIO hop count in little-endian mode when RTYPE is maintenance read or write.
RADDR[65:34]	3	31:0	S-RIO address bits 65:34 in little-endian mode. Valid if RTYPE is not maintenance read/write. Whether 34/50/66-bit S-RIO addressing mode is used is configurable per device through the RapidIO Processing Element Logical Layer Control CSR.
BUFFER_ PTR[31:0]	4	31:0	PCIe address lower. Lower 32 bits of 64-bit PCIe address.
BUFFER_ PTR[63:32]	5	31:0	PCIe address upper. Upper 32 bits of 64-bit PCIe address.
SSDIST[15:0]	6	15:0	Source Stride Distance. This field contains the source stride distance in bytes. The value in this field is a signed number in two’s complement notation. If RTYPE field is a read, this is S-RIO stride distance. If RTYPE is a write, this is PCIe stride distance.
DSDIST[15:0]	6	31:16	Destination Stride Distance. This field contains the destination stride distance in bytes. The value in this field is a signed number in two’s complement notation. If RTYPE field is a read, this is PCIe stride distance. If RTYPE is a write, this is S-RIO stride distance.

Table 67: Data Transfer DMA Descriptor Format *(Continued)*

Field	dword	Bit Position	Description
SSSIZE[11:0]	7	11:0	Source Stride Size. This field specifies the source stride in bytes. A value of zero indicates an infinite stride size. If RTYPE field is a read, this is S-RIO stride size. If RTYPE is a write, this is PCIe stride size.
DSSIZE[11:0]	7	23:12	Destination Stride Size. This field specifies the destination stride in bytes. A value of zero indicates an infinite stride size. If RTYPE field is a read, this is PCIe stride size. If RTYPE is a write, this is S-RIO stride size.

- One DMA data transfer may consist of multiple S-RIO packets
- A data transfer DMA descriptor instructs the DMA channel to perform a data transfer operation. Processing of the descriptor completes when either the data transfer operation completes or when an error is detected.
  - — The S-RIO address, RADDR[65:0], can have any byte alignment
    - – When RTYPE is NREAD, RADDR is source address, denoted as SADDR; otherwise RADDR is destination address, denoted as DADDR
  - — The 64-bit PCIe address, BUFFER_PTR[63:0], can have any byte alignment.
    - – When RTYPE is NREAD, BUFFER_PTR is destination address, denoted as DADDR; otherwise BUFFER_PTR is source address, denoted as SADDR
  - — The BCOUNT field specifies the number of bytes to transfer.
  - — The data transfer operation performed in processing the descriptor is controlled by DMA parameters as described in Data Transfer and Addressing.
    - – The SADDR, DADDR, and BCOUNT parameters are contained in the data transfer DMA descriptor.
    - – The SSSIZE, SSDIST, DSSIZE, and DSDIST parameters are also contained in the data transfer descriptor
- For a descriptor with RTYPE of NREAD or maintenance read, the Block DMA Engine generates one or more NREAD packets, and transforms associated S-RIO responses into PCIe MWr
  - — If the PCIe address is below 4 GB, then a MWr TLP with a 32-bit address is generated. If the address is above 4 GB, then a MWr TLP with a 64-bit address is generated.
  - — Multiple NREAD are generated when one of the following occurs:
    - – BCOUNT exceeds 256 bytes
    - – RADDR is not 8-byte aligned
    - – S-RIO addressing uses strides, and descriptor requested payload straddles multiple S-RIO strides of non-sequential addresses. When this occurs, the Block DMA Engine generates multiple S-RIO NREADs as required.
  - — An S-RIO response can be transformed by the DMA channel into one or more MWr TLPs.
    - – An S-RIO response can cross a PCIe 4-KB boundary
    - – An S-RIO response can exceed PCIe MAX_PAYLOAD_SIZE
    - – PCIe addressing can use strides and require that S-RIO response data be written to two or more non-sequential addresses. When this occurs, the Block DMA Engine splits an S-RIO response into multiple MWr TLPs as required.

- — Note that per S-RIO rules, responses associated with different requests have no ordering relationship. Thus, such responses may not be received by a requester (for example, the Block DMA Engine) in the same order that the requests were issued. The Block DMA Engine converts these responses into MWr TLPs on the fly. As a result, the MWr TLPs issued by the Block DMA Engine may not arrive at the destination location in the order in which the NREAD were issued.
- For a descriptor with an RTYPE other than NREAD or maintenance read, the Block DMA Engine generates one or more PCIe MRd to fetch data, and transforms the associated PCIe cplD into one or more NWRITE/SWRITE/NWRITE_R/maintenance write packets.
  - — If the PCIe address is below 4 GB, then a MRd TLP with a 32-bit address is generated. If the address is above 4 GB, then a MRd TLP with a 64-bit address is generated.
  - — Multiple MRd are generated when one of the following occurs:
    - – BCOUNT exceeds MRRS in the PCIe Device Control and Status Register
    - – Descriptor requested payload crosses a 4-KB boundary
    - – PCIe addressing uses strides, and descriptor requested payload straddles multiple PCIe strides of non-sequential addresses. When this occurs, the Block DMA Engine generates multiple MRd TLPs as required.
  - — The DMA channel attempts to issue the largest memory read request possible that is less than or equal to MRRS in the PCIe Device Control and Status Register, and does not cause the read request to cross a 4-KB boundary.
  - — In response to a memory read request, one or more completions are returned. These completions are transformed by the DMA channel into one or more NWRITE/SWRITE/NWRITE_R/maintenance write packets.
    - – The DMA channel reassembles completions into NWRITE/SWRITE/NWRITE_R/maintenance write packets to conform to S-RIO ordering rules. However, the Block DMA Engine may sometimes split completions into multiple NWRITE/SWRITE/NWRITE_R packets.
    - – Since an S-RIO address must be 8-byte aligned, and both PCIe and S-RIO addresses may have any byte alignment, the Block DMA Engine may need to segment a completion into multiple NWRITE/SWRITE/NWRITE_R packets.
    - – The S-RIO addressing may use strides and require that completion data be written to two or more non-sequential addresses. When this occurs, the Block DMA Engine splits the completion into multiple NWRITE/SWRITE/NWRITE_R packets as required.
    - – Note that as per PCIe rules, completions associated with different requests have no ordering relationship. Thus, such completions may not be received by a requester (for example, the Block DMA Engine) in the same order that the requests were issued. The Block DMA Engine reassembles these completions so that resulting NWRITE/SWRITE/NWRITE_R packets arrive at the destination location in the correct order (see Descriptor Ordering Rules).
- Stride control fields, SSSIZE/ DSSIZE/ SSDIST/ DSDIST, are used to modify DMA channel stride parameters.

10.4.4 Immediate Data Transfer Descriptor

⚠ The operation of the Tsi721 is undefined after processing a descriptor with reserved values in any descriptor field.

An immediate data transfer descriptor instructs the DMA channel to either transfer data that is embedded in the descriptor to a destination device (that is, there is no source address associated with the DMA transfer), or transfer data from a source device into data fields embedded in the descriptor (that is, there is no destination address associated with the DMA transfer).

The format of an immediate data transfer descriptor is displayed in Figure 43, and its fields are described in Table 68. The format is displayed in little-endian.

Figure 43: Immediate Data Transfer DMA Descriptor Format

<table>
<tr><th></th><th colspan="8">bte 3</th><th colspan="8">byte 2</th><th colspan="8">byte 1</th><th colspan="8">byte 0</th></tr>
<tr><td>DW</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td colspan="3">DTYPE</td><td>R</td><td>IOF</td><td>R</td><td>R</td><td colspan="2">R</td><td colspan="4">RTYPE[3:0]</td><td colspan="2">PRIO</td><td>CRF</td><td colspan="16">DEVID</td></tr>
<tr><td>1</td><td colspan="2">RADDR[1:0]</td><td colspan="2">R</td><td colspan="2">TT[1:0]</td><td colspan="22">R</td><td colspan="4">BCOUNT[3:0]</td></tr>
<tr><td>2</td><td colspan="8">RADDR[33:26]/HOP_CNT[7:0]</td><td colspan="24">RADDR[25:2]/CONFIG_OFFSET[23:0]</td></tr>
<tr><td>3</td><td colspan="32">RADDR[65:34]</td></tr>
<tr><td>4</td><td colspan="32">DATA[31:0]</td></tr>
<tr><td>5</td><td colspan="32">DATA[63:32]</td></tr>
<tr><td>6</td><td colspan="32">DATA[95:64]</td></tr>
<tr><td>7</td><td colspan="32">DATA[127:96]</td></tr>
</table>

Table 68: Immediate Data Transfer DMA Descriptor Format

Field	dword	Bit Position	Description
DEVID15:0]	0	15:0	S-RIO destination ID
CRF	0	16	S-RIO critical request flow.
PRIO[1:0]	0	18:17	S-RIO priority. Valid values are 0–2 (for request flows), as PRIO of 3 is for S-RIO response as per part 6 of the *RapidIO Specification (Rev. 2.1)*.
RTYPE[3:0]	0	22:19	S-RIO Transaction Type • 0b0000 = NREAD • 0b0001 = Reserved • 0b0010 = All S-RIO packets use NWRITE or SWRITE • 0b0011 = All S-RIO packets use NWRITE_R • 0b0100 = Maintenance read • 0b0101 = Maintenance write • Others = Reserved
IOF	0	27	Interrupt on Finished. When this bit is set, the Tsi721 will generate an interrupt when it finishes processing this descriptor, and the IOF_DONE bit is set in the corresponding DMA Channel Interrupt Register {0..7}.
DTYPE[2:0]	0	31:29	Descriptor Type. This field encodes the type of descriptor. • 0x2 = Immediate data transfer DMA descriptor
BCOUNT[3:0]	1	3:0	Byte Count. The total number of bytes to transfer. • 0x0 = 16 bytes • 0x1 = 1 byte • 0x2 = 2 bytes • ... • 0xF = 15 bytes If RTYPE is maintenance read or write, BCOUNT must be 4 bytes.

This document is confidential and is subject to an NDA.

Table 68: Immediate Data Transfer DMA Descriptor Format *(Continued)*

Field	dword	Bit Position	Description
TT[1:0]	1	27:26	S-RIO Transport Type. • 0b00 = 8-bit deviceID • 0b01 = 16-bit deviceID • Others = Reserved
RADDR[1:0]	1	31:30	S-RIO Address bits 1:0 in little-endian format. Valid if RTYPE is not a maintenance read or write.
RADDR[33:2]	2	31:0	S-RIO Address bits 33:2 in little-endian format. Valid if RTYPE is not maintenance read or write.
CONFIG_OFFSET[23:0]	2	23:0	S-RIO configure offset address in little-endian format when RTYPE is a maintenance read or write. CONFIG_OFFSET[1:0] must be all zeroes because maintenance read or write must have 32-bit aligned address.
HOP_CNT[7:0]	2	31:24	S-RIO hop count in little-endian format when RTYPE is a maintenance read or write.
RADDR[65:34]	3	31:0	S-RIO Address bits 65:34 in little-endian format. Valid if RTYPE is not a maintenance read or write. Whether 34/50/66-bit S-RIO addressing mode is used is configurable per device through the RapidIO Processing Element Logical Layer Control CSR.
DATA[31:0]	4	31:0	Data bits 31:0.
DATA[63:32]	5	31:0	Data bits 63:32.
DATA[95:64]	6	31:0	Data bits 95:64.
DATA[127:96]	7	31:0	Data bits127:96.

- An immediate data transfer descriptor is processed similarly to a data transfer descriptor, except that data is embedded inside the descriptor and can be up to 16 bytes.
  - When RTYPE is NWRITE/SWRITE/NWRITE_R/maintenance write, no PCIe MRd is issued for an immediate data transfer because data is embedded inside descriptor itself
  - When RTYPE is NREAD/maintenance read, PCIe MWr is generated using the descriptor’s address to store data into DATA fields of the descriptor
  - Immediate data transfers do not support constant addressing or strides
- For a descriptor with RTYPE of NREAD or maintenance read, the Block DMA Engine generates one or more NREAD packets, and transforms associated S-RIO responses into PCIe MWr
  - If the descriptor’s PCIe address is below 4 GB, then a MWr TLP with a 32-bit address is generated. If the address is above 4 GB, then a MWr TLP with a 64-bit address is generated.
  - Multiple NREAD are generated when RADDR is not 8-byte aligned
  - An S-RIO response can be transformed by the DMA channel into one or more MWr TLPs
    - An S-RIO response can cross PCIe 4-KB boundary

- Note that as per S-RIO rules, responses associated with different requests have no ordering relationship. Therefore, such responses may not be received by a requester (for example, the Block DMA Engine) in the same order that the requests were issued. The Block DMA Engine converts these responses into MWr TLPs on the fly. As a result, the MWr TLPs issued by the Block DMA Engine may not arrive at the destination location in the order in which the NREAD were issued.

- For a descriptor with another RTYPE, the Block DMA Engine generates one or more NWRITE/SWRITE/NWRITE_R/maintenance write packets
  - Multiple NWRITE/SWRITE/NWRITE_R are generated when RADDR is not 8-byte aligned

10.4.5 Block Pointer Descriptor

The operation of the Tsi721 is undefined after processing a descriptor with reserved values in any descriptor field.

The block pointer descriptor has a DTYPE of 3. It is essentially a pointer that links together descriptor blocks. The block pointer descriptor format is displayed in Figure 44, and its fields are described in Table 69.

Figure 44: Block Pointer DMA Descriptor Format

	bte 3								byte 2								byte 1								byte 0							
DW	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	DTYPE			R																												
1	R																															
2	NEXT[31:5]																											R				
3	NEXT[63:32]																															
4	R																															
5	R																															
6	R																															
7	R																															

Table 69: Block Pointer DMA Descriptor Format

Field	dword	Bit Position	Description
DTYPE[2:0]	0	31:29	Descriptor Type. This field encodes the type of descriptor. 0x3 = Block pointer descriptor
NEXT[31:5]	2	31:5	Next Descriptor Block Pointer Lower. {NEXT[31:5], 0b00000} (Tsi721 automatically generates all zeroes for LSB bits 0–4) forms the lower 32 bits of next descriptor block pointer. It points to the start address of the next descriptor block.
NEXT[63:32]	3	31:0	Next Descriptor Block Pointer Upper. This is upper 32 bits of the next descriptor block pointer.

10.5 DMA Descriptor Processing

DMA descriptor processing consists of reading a descriptor from memory, executing the operation outlined by the descriptor, and then proceeding to the next descriptor.

10.5.1 DMA Descriptor Status

A DMA channel aborts DMA processsing when either a PCIe error or an S-RIO error occurs. However, all processed descriptors before an error occurs have only one status: OK status.

For each DMA channel, the Tsi721 maintains a descriptor status FIFO in the PCIe side main memory, where the FIFO stores the pointer to successfully processed type 1/2/3 descriptors. The status FIFO is defined by DMA Descriptor Status FIFO Lower Base Register {0..7}, DMA Descriptor Status FIFO Upper Base Register {0..7}, and DMA Descriptor Status FIFO Size Register {0..7}.

The FIFO is organized as 64 bytes wide, matching Intel processor's cache line size. Each FIFO entry of 64 bytes is divided into 8 elements, with 8 contiguous bytes per element, displayed as DA0[63:0] to DA7[63:0] in Figure 45. Each element can store one 8-byte pointer to a successfully processed descriptor by the Tsi721, where an all zero element – for example, DAj (j = 0–7) – is reserved to mark this and higher address elements (DAk, k>=j) as invalid. That is, the Tsi721 updates the descriptor status on a per cache line basis (status MWr TLP has 64-byte payload), and can store from one to eight pointers to successfully processed descriptors in one status MWr TLP into consecutive elements (starting from LSB) of one FIFO entry.

Figure 45: Descriptor Status FIFO Entry Format

byte 63 ~ 56	byte 55 ~ 48	byte 47 ~ 40	byte 39 ~ 32	byte 31 ~ 24	byte 23 ~ 16	byte 15 ~ 8	byte 7 ~ 0
bits 511:448	447:384	383:320	319:256	255:192	191:128	127:64	63:0
DA7[63:0]	DA6[63:0]	DA5[63:0]	DA4[63:0]	DA3[63:0]	DA2[63:0]	DA1[63:0]	DA0[63:0]

The FIFO start address must be 64-byte aligned. Software and the Tsi721 co-manage the descriptor status FIFO, with software performing FIFO reads and the Tsi721 performing FIFO writes. Software informs the Tsi721 about its read pointer through RD_PTR in DMA Descriptor Status FIFO Read Pointer Register {0..7}, and the Tsi721 updates its write pointer through WR_PTR in DMA Descriptor Status FIFO Write Pointer Register {0..7}. When a FIFO entry is read, software clears the entry with all zeroes. Software informs the Tsi721 about its read pointer through a register write. The Tsi721 stalls FIFO writes when (WR_PTR + 1) modulus DMA Descriptor Status FIFO Size Register {0..7} is equal to RD_PTR. Software should decide if an FIFO entry is valid by continuously reading the FIFO until it encounters an all zero entry (Tsi721 write pointer should not be used).

When the status FIFO becomes full, the Tsi721 flags an interrupt and stalls descriptor processing for the associated DMA channel. Software should empty the status FIFO quickly to achieve high throughput.

10.5.2 Descriptor List Processing

- Software initializes a DMA channel by programming the DMA Channel Descriptor Pointer Low Register {0..7} and DMA Channel Descriptor Pointer High Register {0..7} that together form a 64-bit descriptor address, clearing ERROR bit in the DMA Channel Interrupt Register {0..7}, then setting the INIT bit in the DMA Channel Control Register {0..7}.
  - — Tsi721 resets DMA Descriptor Write Count Register {0..7}, DMA Descriptor Read Count Register {0..7}, DMA Descriptor Status FIFO Write Pointer Register {0..7}, DMA Descriptor Status FIFO Read Pointer Register {0..7}, and DMA Channel Status Register {0..7} to zero when INIT is set to 1.
- DMA descriptor processing is initiated when software writes to the DMA Descriptor Write Count Register {0..7}, and the Tsi721 resets INIT in DMA Channel Control Register {0..7} when software writes to DMA Descriptor Write Count Register {0..7}.

- When DMA descriptor processing is initiated, the Block DMA Engine reads and begins processing the descriptor at the address pointed to by the 64-bit descriptor address.
  - When DMA descriptor processing is initiated, the RUN bit in the DMA Channel Status Register {0..7} is set.
- When the DMA channel finishes processing of a descriptor without error,
  - The pointer to the descriptor is used to as part of descriptor status update MWr
  - If descriptor's IOF is high, Tsi721 generates an interrupt
  - Processing for a descriptor has successfully completed when both of the following conditions are met:
    - All S-RIO requests associated with that descriptor have been generated by the appropriate engine and buffered for transmission.
    - All S-RIO responses associated with that descriptor – if there are any – have been received successfully, specifically:
      * NREAD descriptors have finished processing when all responses have been returned.
      * NWRITE_R descriptors have finished processing when all responses have been returned.
      * NWRITE/SWRITE descriptors have finished processing when all requests have been generated and buffered for transmission
      * Maintenance read/write descriptors have finished processing when all responses have been returned
- If this descriptor is not the end of descriptor list (monitored through the DMA Descriptor Read Count Register {0..7} and DMA Descriptor Write Count Register {0..7}), then the DMA channel proceeds to process the next descriptor
  - If the processed descriptor is not a block pointer descriptor, the next descriptor locates at the address that is 32 bytes above the processed descriptor's address.
  - If the processed descriptor is a block pointer descriptor, the next descriptor locates at the address pointed by the NEXT field of the block pointer descriptor.
  - The address of the next descriptor is held in the DMA Channel Descriptor Pointer Low Register {0..7} and DMA Channel Descriptor Pointer High Register {0..7}
  - In actual implementation, Tsi721 performs descriptor prefetch, and the next descriptor is fetched from a DMA channel's descriptor prefetch FIFO
  - However, similar address calculation is used to determine address of descriptor prefetch
- If this descriptor is the end of descriptor list, the DMA channel stops descriptor processing, sets the DONE bit in the DMA Channel Interrupt Register {0..7}, and clears the RUN bit in the DMA Channel Status Register {0..7}.
  - The DMA Channel Descriptor Pointer Low Register {0..7} and DMA Channel Descriptor Pointer High Register {0..7} points to the next descriptor to be fetched
    - Assuming that the address of the last successfully processed descriptor is LADDR
    - If LADDR is for a type 1 or 2 descriptor, then DMA Channel Descriptor Pointer Low Register {0..7} and DMA Channel Descriptor Pointer High Register {0..7} together hold the value of LADDR + 32
    - If LADDR is for a type 3 descriptor, then DMA Channel Descriptor Pointer Low Register {0..7} and DMA Channel Descriptor Pointer High Register {0..7} together hold the value of NEXT field of the type 3 descriptor
  - If the DMA Descriptor Write Count Register {0..7} is updated again by software, the DMA channel restarts descriptor processing by fetching the descriptor pointed to by the DMA Channel Descriptor Pointer Low Register {0..7} and DMA Channel Descriptor Pointer High Register {0..7} and continue to process the next descriptor.

10.5.3 DMA Channel Abort on Error

10.5.3.1 Errors Causing DMA Channel Abort

The processing of a DMA descriptor by a DMA channel can be aborted when an error is detected during descriptor processing. Here, all DMA channel specific PCIe errors (ECRC error is not channel specific) and S-RIO errors cause the associated DMA channel to abort. These errors are listed in the CS[4:0] field of the DMA Channel Status Register {0..7}.

10.5.3.2 DMA Channel Abort

A DMA channel abort involves the following actions:

1. The DMA channel stops prefetching new descriptors.
2. The DMA channel updates the DMA Channel Interrupt Register {0..7}, sets the ERROR bit in the DMA Channel Interrupt Register {0..7} and flags an interrupt.
3. The DMA channel stops issuing new requests except descriptor status MWr requests from finished descriptors and waits for all these descriptor status MWr to be sent
4. The DMA Channel Descriptor Pointer Low Register {0..7} and DMA Channel Descriptor Pointer High Register {0..7} are undefined.
5. Descriptor prefetching is stopped and the DMA Descriptor Read Count Register {0..7} stops incrementing.
6. The Tsi721 lets all outstanding PCIe/ S-RIO requests/ completions/ responses of the DMA channel to drain through all shared resources, and then discards them.
7. If any outstanding PCIe/ S-RIO completions/ responses of the DMA channel do not arrive due to error, the Tsi721 waits for their completion/ response timeout to occur, then sets the ABORT bit in the DMA Channel Status Register {0..7}.
8. To recover from a DMA channel abort, software must re-initialize the DMA channel (set INIT bit of Inbound DMA Channel Control Register {0..7}), upon which the ABORT bit in DMA Channel Status Register {0..7} is cleared; then clear the ERROR bit in the DMA Channel Interrupt Register {0..7}; lastly, write to the DMA Descriptor Write Count Register {0..7}).

10.5.4 Suspending and Resuming a DMA Channel

- In some applications it is desirable to append descriptors to an active descriptor list; that is, one that is being processed by a DMA channel. The DMA channel suspend and resume capability provides a race-free method to perform this operation.
- A DMA channel abort has higher priority than a DMA channel suspend. If an error is detected while carrying out a DMA channel suspend, then the DMA channel abort procedure takes over and the DMA channel suspend procedure is forfeited.
- The operation of a DMA channel can be suspended by writing a 1 to the SUSPEND bit in the DMA Channel Control Register {0..7}. When a DMA operation is suspended, the following occurs:
  - The DMA channel stops prefetching new descriptors
  - The DMA channel normally processes all prefetched descriptors Once all prefetched descriptors have been processed, the DMA channel loads the DMA Channel Descriptor Pointer Low Register {0..7} and DMA Channel Descriptor Pointer High Register {0..7} to point to the next descriptor to the last processed descriptor, and the sets the SUSPENDED bit in the DMA Channel Interrupt Register {0..7}.
  - If the DMA channel is halted when suspended – that is, the DMA channel has completed processing descriptors in a list – then the SUSPENDED bit in the DMA Channel Interrupt Register {0..7} is immediately set.
    - Software should wait (through either polling or interrupt) for the SUSPENDED bit in the DMA Channel Interrupt Register {0..7} to be set before resuming the DMA channel. Violating this rule produces undefined behavior.
    - When a DMA channel is halted, the DMA Channel Descriptor Pointer Low Register {0..7} and DMA Channel Descriptor Pointer High Register {0..7} point to the next descriptor to be fetched (32+ descriptor address at the end of the descriptor list)
- The Tsi721 generates status update MWr TLP
- The RUN bit in the DMA Channel Status Register {0..7} is cleared while a DMA is suspended.

- Software can optionally append descriptors to an existing descriptor list while the SUSPENDED bit is set in the DMA Channel Interrupt Register {0..7}.
  - — In addition, software can also insert new descriptor blocks by first reading the DMA Channel Descriptor Pointer Low Register {0..7} and DMA Channel Descriptor Pointer High Register {0..7} and saving them as a link in address – that is, use them to create a type 3 descriptor – then overwriting the DMA Channel Descriptor Pointer Low Register {0..7} and DMA Channel Descriptor Pointer High Register {0..7} with a type 3 descriptor pointing to a new descriptor block.
- A suspended DMA channel can be resumed by writing to the DMA Descriptor Write Count Register {0..7} (a suspended DMA channel should not have any errors except ECRC errors).
  - — The Block DMA Engine clears the SUSPEND bit in the DMA Channel Control Register {0..7}.
  - — The Block DMA Engine fetches the descriptor pointed by DMA Channel Descriptor Pointer Low Register {0..7} and DMA Channel Descriptor Pointer High Register {0..7} and starts processing the descriptor.

10.5.5 Dynamic Chaining of Descriptor Lists

Suspending and Resuming a DMA Channel describes a race-free method to append or modify descriptors in an active descriptor list by suspending and resuming descriptor processing in a DMA channel. For scenarios where a suspend/resume operation is not desired (for example, to improve DMA performance), this section presents an alternative method to performing dynamic appending of descriptors to a descriptor list.

Descriptors can be appended to an active descriptor list – that is, a descriptor list which a DMA channel is currently processing – by appending new descriptors to the last descriptor. After descriptors are appended to an active descriptor list, software must write the DMA Descriptor Write Count Register {0..7} with a new value (except DMA channel INIT/suspend/error, DMA Descriptor Write Count Register {0..7} must be increasing; otherwise, behavior is undefined).

- If the DMA channel has not finished processing the last descriptor in the original list when the DMA Descriptor Write Count Register {0..7} is updated, the DMA channel will continue processing descriptors normally, including the newly appended descriptors.
- If the DMA channel has completed descriptor processing of the last descriptor in the original list when the DMA Descriptor Write Count Register {0..7} is updated – that is, the DMA channel halted after processing the last descriptor in the list – the DMA channel restarts descriptor processing.
  - — The DMA Channel Descriptor Pointer Low Register {0..7} and DMA Channel Descriptor Pointer High Register {0..7} point to the next descriptor to be fetched
  - — The Block DMA Engine fetches the descriptor pointed by the DMA Channel Descriptor Pointer Low Register {0..7} and DMA Channel Descriptor Pointer High Register {0..7}, and starts processing the next descriptor

10.6 TLP Attribute and Traffic

The Block DMA Engine sends TLPs with the Relaxed Ordering (RO) header bit cleared and Traffic Class (TC) header bits cleared. For the No Snoop header bit, it is cleared for descriptor and status update TLPs and it is set to the NSP bit field of the associated descriptor for data MRd and MWr TLPs.

10.7 DMA Outstanding PCIe Requests

The Block DMA Engine supports a total of 32 DMA outstanding requests, including both DMA data MRd and DMA descriptor MRd:

- They are shared among DMA channels.
- 32 outstanding tags are per device, shared are by the SR2PC, BDMA, and SMSG modules

The performance of the Block DMA Engine improves when the DMA generates few read requests when processing a descriptor, and each read request transfers a large amount of data. Note that the DMA addressing specified in the descriptor determines the boundaries at which the DMA channel breaks requests. Linear addressing causes the DMA channel to break requests at every 4 KB of data transfer on average.

10.8 Descriptor Prefetching

When the amount of data moved by data transfer descriptors is small (for example, when moving data associated with 64-bit packets), the overhead in fetching DMA descriptors from memory between data transfer operations may limit performance. To overcome this overhead, the Block DMA Engine supports descriptor prefetching, whereby 32 descriptors can be prefetched per DMA channel.

- When the fill level of a DMA channel's descriptor prefetch FIFO is less than 25, the Block DMA Engine issues descriptor MRd to fetch up to MAX_PAYLOAD_SIZE/32 number of descriptors before previous descriptors has been completed.
- The DMA channel queues prefetched descriptors until they are processed.
- Prefetched descriptors are discarded when a DMA channel is aborted
- Descriptor prefetching stops when the end of a descriptor list is reached. The descriptor prefetch never passes the last descriptor of a descriptor list.
- The descriptor prefetch never crosses a 4-KB boundary.

10.9 DMA Limitations

To achieve a high throughput for DMA operations, the Block DMA Engine processes multiple descriptors in a pipeline. If a descriptor with RTYPE of NREAD has buffer space and overlaps with buffer space of the next descriptor with RTYPE of NWRITE or SWRITE, data returned from the NREAD may overwrite data for the NWRITE, which results in data corruption. As a result, active descriptors should not be allocated to overlapping buffer space.

10.10 Block DMA Engine Scheduling

10.10.1 S-RIO Egress Scheduling

The Tsi721 uses the following algorithm for S-RIO egress scheduling:

1. Within each DMA channel, S-RIO egress scheduling is S-RIO ordering rule based (see Descriptor Ordering Rules).
2. Among winners of each DMA channel from step 1, round-robin scheduling is applied.

10.10.2 PCIe Egress Scheduling

The Tsi721 uses the following algorithm for PCIe egress scheduling:

1. Simple round-robin scheduling for data MRd TLPs among DMA channels
2. Simple round-robin scheduling between winners of step 1 and data MWr TLPs, as data MWr TLPs are scheduled from an on-chip FIFO shared among all DMA channels
3. Other types of Block DMA Engine originated TLPs, such as descriptor prefetch MRd and descriptor status MWr, *and so on*, are scheduled using a combination of simple round robin among DMA channels and strict priority scheduling, where their priorities are higher than the winner of step 2.
4. The winner of step 3 participates in the Mapping Engine Scheduling.

Secondary Modules

11. I2C Interface

This section discusses the following topics:

11.1 Overview

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

- Initializing device registers from an EEPROM after reset
- Reading and writing external devices on the I2C bus
- Reading and writing Tsi721's internal registers for management purposes by an external I2C master

The I2C Interface consists of the following features:

- Operates as a master or slave on the I2C bus
  - Multi-master support
    - Arbitrates among multiple masters for ownership of the I2C bus
    - Automatically retries accesses if arbitration is lost
    - Provides timeout indication if the Tsi721 is unable to arbitrate for the I2C bus
  - I2C Controller: Master interface
    - Supports 7-bit device addressing
    - Supports 0, 1, or 2-byte peripheral addressing
    - Supports 0, 1, 2, 3, or 4-byte data transfers
    - Reverts to slave mode if arbitration is lost

    - Supports clock stretching by an external slave to limit bus speed to less than 100 kHz
    - Handles timeouts and reports them through interrupts
  - I2C Controller: Slave interface
    - Slave address can be loaded from three sources: power-up signals, boot load from EEPROM, or by software configuration
    - Provides 32 bits read and write accesses to all Tsi721 registers
    - Ignores General-Call accesses
    - Ignores Start-Byte protocol
    - Provides a status register for determination of Tsi721's health
    - Slave operation enabled/disabled through power-up signals, boot load from EEPROM, or by software configuration
    - Provides mailbox registers for communicating between maintenance software operating on RapidIO-based processors and external I2C masters
- Supports I2C operations up to 100 kHz
- Provides boot-time register initialization
  - Supports 1- and 2-byte addressing of the EEPROM selected by power-up signals
  - Verifies the number of registers to be loaded is legal before loading registers
  - Supports up to 2-KB address space and up to 255 address/data pairs for register configuration in 1-byte addressing mode, or up to 65-KB address space and up to 8K-1 address/data pairs in 2-byte addressing mode.
  - Supports chaining to a different EEPROM and/or EEPROM address during initialization.

The I2C Interface does not support the following features:

- START Byte protocol
  - Tsi721 does not provide a START Byte in transactions it masters
  - Tsi721 does not respond to START Bytes in transactions initiated by other devices. The Tsi721 will respond to the repeated start following the start byte provided the 7-bit address provided matches the Tsi721 device address.
- CBUS compatibility
  - Tsi721 does not provide the DLEN signal
  - Tsi721 does not respond as a CBUS device when addressed with the CBUS address. The Tsi721 will interpret the CBUS address like any other 7-bit address and compare it to its device address without consideration for any other meaning.
- Fast Mode or High-Speed Mode (HS-MODE)
- Reserved 7-bit addresses should not be used as the Tsi721's 7-bit address. If a reserved address is programmed, the Tsi721 will respond to that address as though it were any other 7-bit address with no consideration of any other meaning.
- 10-bit addressing
  - Tsi721 must not have its device address programmed to the 10-bit address selection (11110XXb) in systems that use 10-bit addressing. The Tsi721 will interpret this address like any other 7-bit address and compare it to its device address without consideration for any other meaning.
- General Call. The general call address will be NACK'd and the remainder of the transaction ignored up to a subsequent Restart or Stop.

11.2 Protocol Overview

The I2C protocol is a two-wire serial interface that consists of a bidirectional, open-drain clock bus (I2C_SCL), and a bidirectional open-drain data bus (I2C_SDA). Multiple master and/or slave devices can be connected to an I2C bus. I2C data is transmitted from one device to another across the I2C_SDA bus with timing referenced to the I2C_SCL bus. With some exceptions, each bus can be driven low (to a logic 0) by any device, but is pulled high (to a logic 1) by an external resistor tied to VDD. This creates what is called a "wired-and" configuration, where any single device can drive a bus to a logic 0, but a bus will rise to a logic 1 only if no devices are driving to a logic 0, allowing the pull-up resistor to bring the bus to a logic 1 voltage.

I2C requires one device to assume the role of master during a transfer. The master generates the clock on the I2C_SCL bus and for controlling the overall transfer protocol, as defined by the *I^2C Specification*. One or more devices assume the role of slaves during the transfer and respond to the master by either accepting data from the I2C_SDA bus, or providing data to the I2C_SDA bus. A master transmitting a unique slave address as part of the I2C protocol selects a specific device to act as a slave. Only one device is usually configured with the specific slave address and will be the only device to respond to the master. Other parts of the I2C protocol provide for arbitration between multiple master devices, allowing more than one master device to share the bus on a one-at-a-time basis.

11.3 Architecture

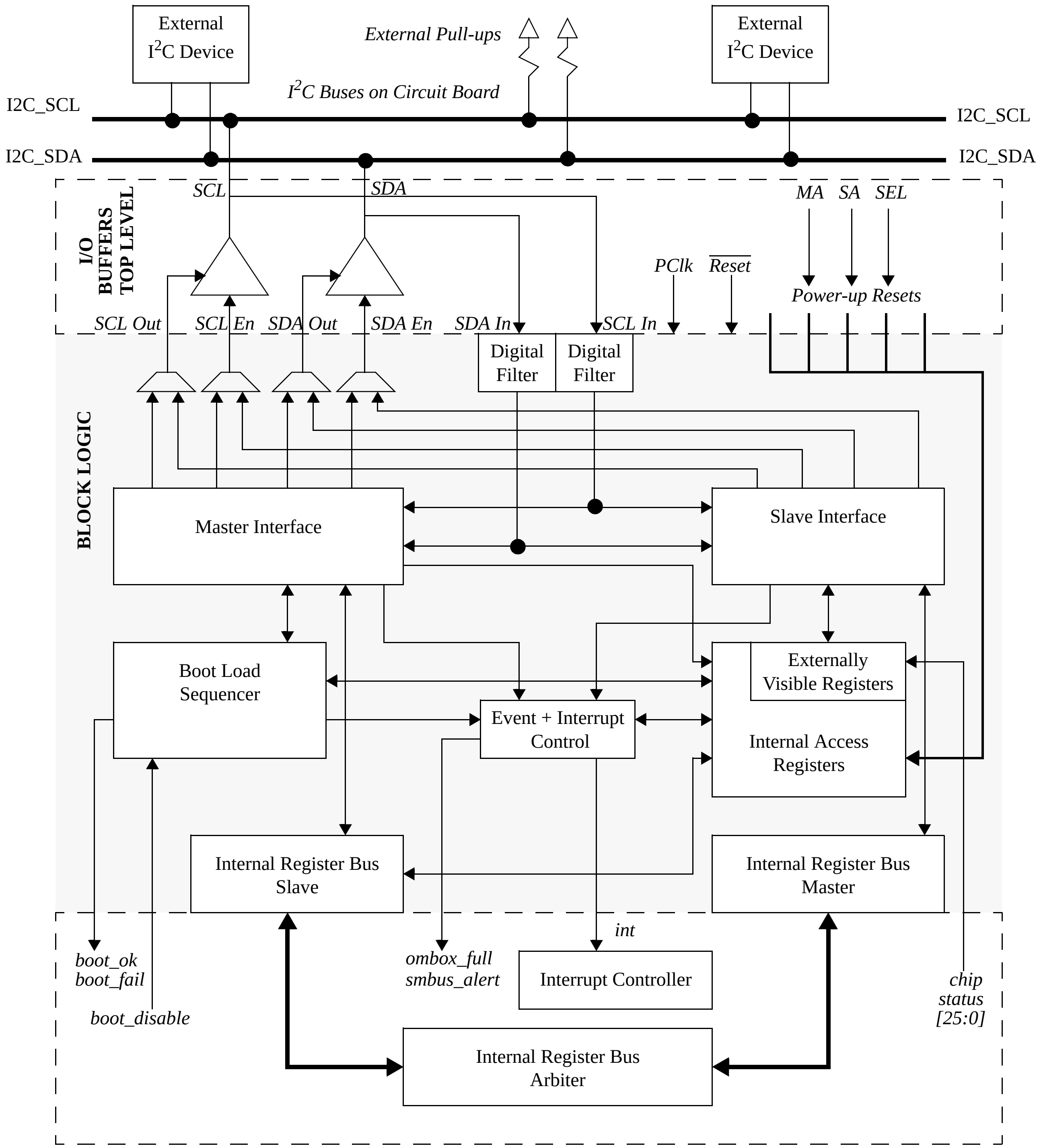
Figure 46 shows an overview of the I2C Interface. The shaded area is the module logic. The master and slave interfaces mux between control of the I2C_SDA and I2C_SCL I/O buffers, and connect through the package to the buses on the board. The reference clock (PClk) and active-low hard reset are inputs to the module. The power-up reset values are either static signals from outside the module, or connect to device signals for board-level configuration. The I2C_MA pin is a power-up configuration pin that is latched during reset.

On the core side, the I2C module connects to the internal device register bus as a slave and master:

- As a slave, it enables access to the I2C module registers by a host or processor.
- As a master, it enables access to other device registers (for example, during the I2C load at power-up).

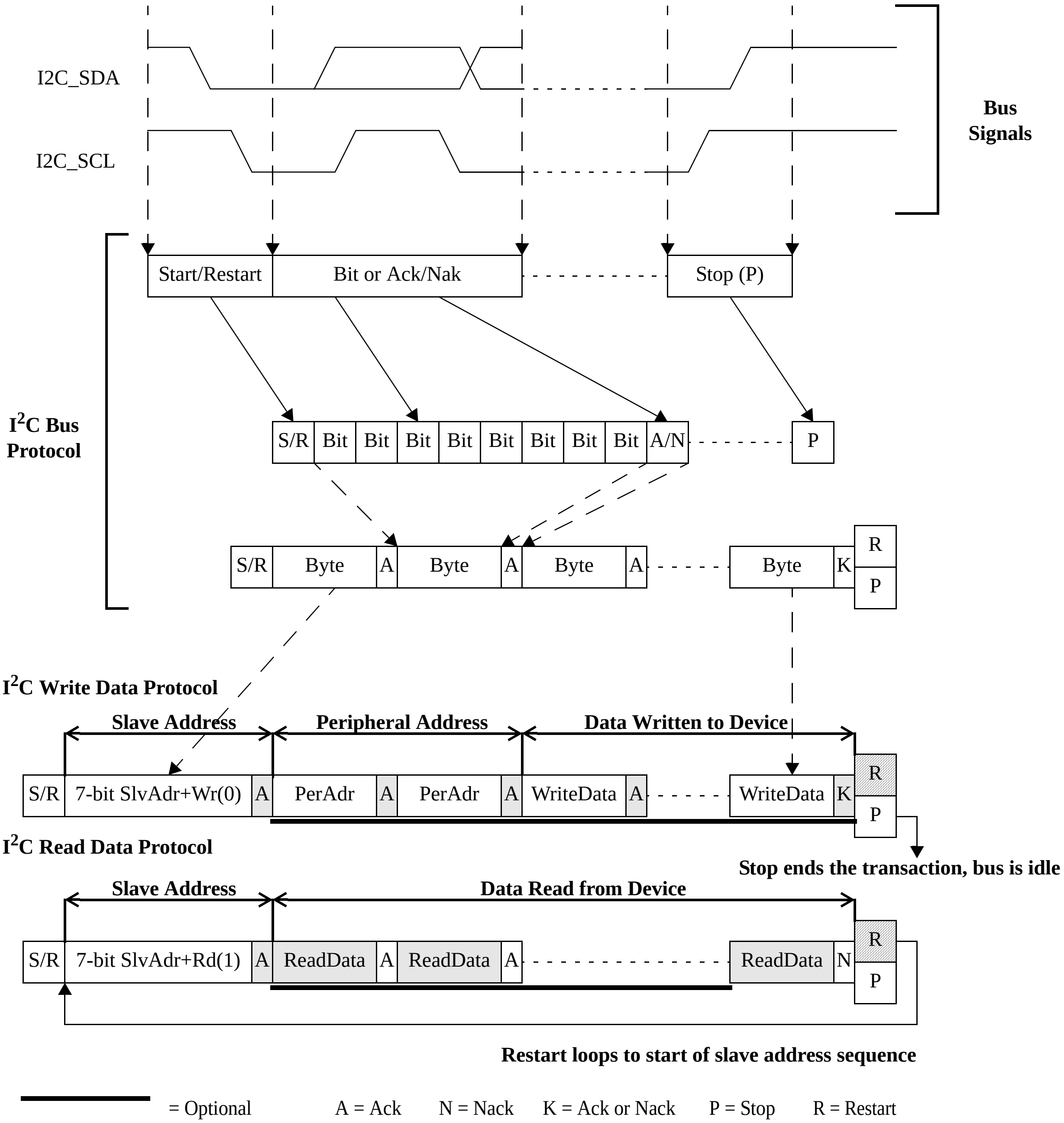
In addition, various signals relate to the boot load sequencer, an interrupt signal connects to the Tsi721 Interrupt Controller, and device-level status connects to the Externally Visible I2C Status Register in the externally visible registers.

Figure 46: I2C Block Diagram



The reference diagram in Figure 47 shows the I2C bus and data protocol. Three basic signal relationships are defined by the bus timing: Start/Restart, Bit, and Stop.

Figure 47: I2C Reference Diagram



Note: The I^{2}C read data protocol section of this figure implies that the peripheral addressing phase has already taken place. The I^{2}C Interface will remember where it left off such that future reads to the same device will not require the peripheral addressing phase; however, an initial read

The start/restart and stop conditions delineate a transaction – a master issues a start to claim ownership of the bus and a stop to release ownership. A restart is a repeated start condition between the first start and the terminating stop, and is used by a master to start a new transaction without giving up bus ownership. Between the start and stop, data is transferred one bit at a time, with the basic protocol calling for full bytes of data, this being 8 consecutive bits from master to slave or slave to master, followed by 1 more bit driven by the device receiving the data to acknowledge the receipt of the byte. The first stream of bytes following a start/restart is the device connection sequence, where the master places a slave address on the bus to make a connection to the device that it wants to communicate with. It is also during this period that primary arbitration for bus ownership is completed for the bus between multiple masters. If more than one master attempts to address a slave, then all but one master backs off and waits for a stop condition, when the bus ownership is again released.

Once a connection is made between a master and a slave, data can be transferred to or from the slave in the data read/write sequence phase of the transaction. This sequence is device-dependent, but a common protocol used by memory-oriented devices such as EEPROMs involves the master sending one or more bytes of memory address to the slave to position the slave’s memory address (or peripheral address), then the master will write/read data to/from the slave. Eventually the master ends the transaction with a stop condition, at which point the bus is free for other masters to attempt to start transactions.

These I2C master and slave operations are explained in the following sections.

11.4 Tsi721 as I2C Master

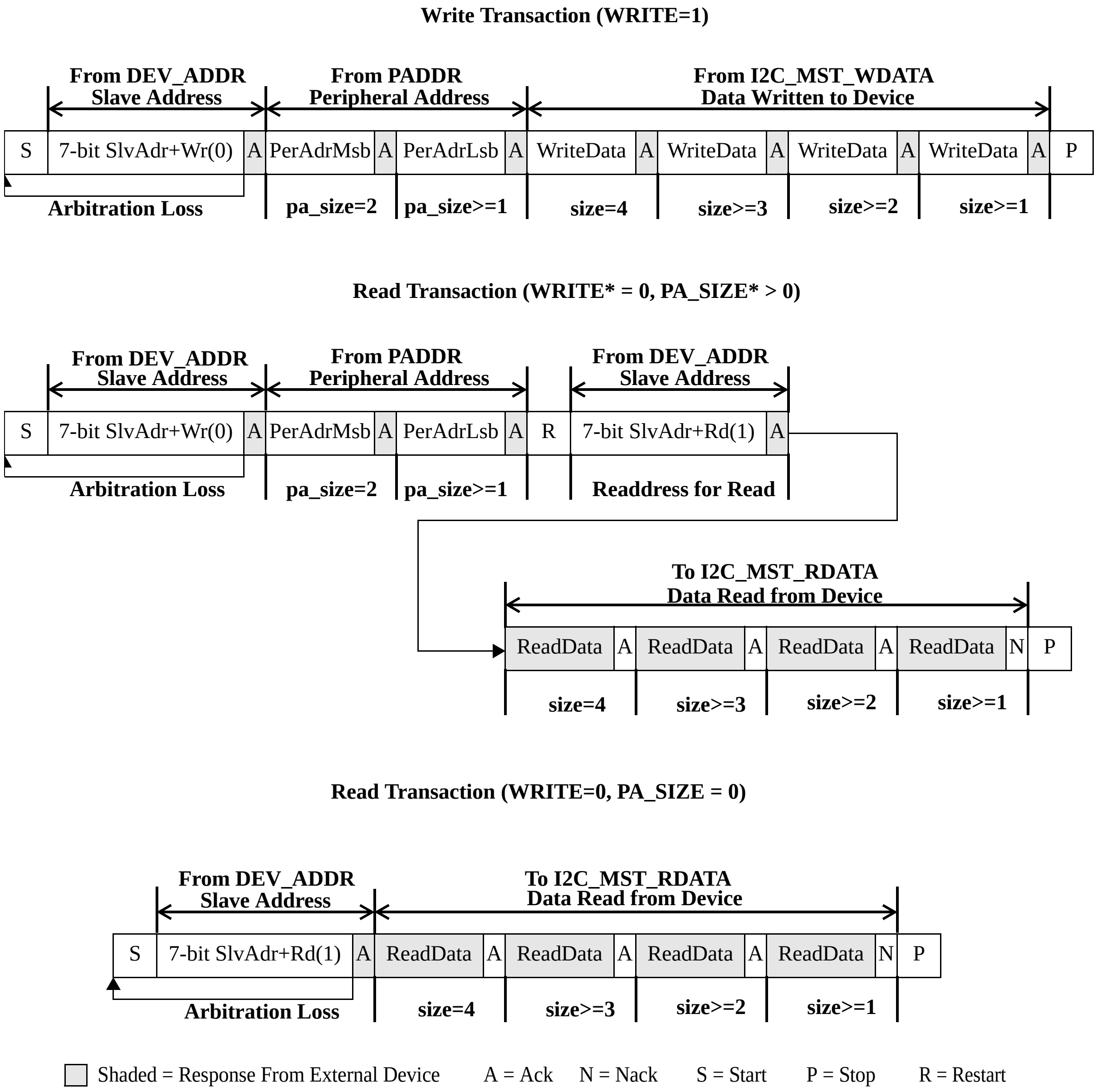
When the Tsi721 is an I2C master, it addresses an external slave device, generates the I2C_SCL clock, and controls the overall transfer protocol. There are two instances where the Tsi721 is master: boot loading (as described in Boot Load Sequence), and transactions initiated by setting the START bit in the I2C Master Control Register, as described in the following section.

Software can instruct the Tsi721 to read or write to an external slave device using the following registers:

- I2C Master Configuration Register to configure external device parameters
- I2C Master Control Register to select and start the transaction
- I2C Master Receive Data Register for data to be read (received)
- I2C Master Transmit Data Register for data to be written (transmitted)
- I2C Access Status Register for status

Figure 48 depicts the sequences on the I2C bus when the Tsi721 is mastering a read or write transaction.

Figure 48: Software-initiated Master Transactions



> Note: WRITE resides in theI2C Master Control Register; PA_SIZE resides in the I2C Master Configuration Register“.

The overall procedure is to configure the device through the I2C Master Configuration Register, load the data to be written in the I2C Master Transmit Data Register (only needed for write operations), then load the I2C Master Control Register with the specific transaction information and have the START bit in that register set to 1. This initiates the master transaction. Software usually will then wait for a master-related interrupt to know when the transaction completes, or the status can be polled using the I2C Access Status Register. If the operation was a read, the data can then be retrieved by reading the I2C Master Receive Data Register.

Once a master operation is started, it cannot be aborted by software except through a soft reset of the I2C Interface. A soft reset is not recommended, however, since it can leave the I2C bus in a state that makes it difficult to ensure that all devices are back to an idle state.

11.4.1 Master Clock Generation

The I2C clock (I2C_SCL) for master operation is generated by dividing down the register bus clock (P_CLK). The reset value for the I2C_SCL means a nominal 100-kHz operating speed. The bus speed is affected by external devices stretching the clock, or by any spread-spectrum modulation of the P_CLK.

For master operations, the clock frequency can be changed by modifying the timing parameter registers (see Bus Timing). Operation above 100 kHz is possible but the Tsi721 does not implement all the standard requirements for fast mode.

11.4.2 Master Bus Arbitration

Because the Tsi721 can operate in a multi-master I2C system, it arbitrates for the I2C bus as required by the *I^2C Specification*. During the Start and Slave Address phase, any unexpected state on the bus will cause the Tsi721 to back off, release the bus, and wait for a Stop before retrying the transaction. If the arbitration timer configured in the I2C_SCL Low and Arbitration Timeout Register expires before the device can get through the slave address phase without collision, then the transaction is aborted with the MA_ATMO status in the I2C Interrupt Status Register. An optional interrupt can also be sent to the Interrupt Controller if enabled in theI2C Interrupt Enable Register.

If the Tsi721 loses the arbitration for the I^2C bus, and the winning master selects the Tsi721 as the target of its access, the Tsi721 will respond as the slave.

11.4.3 Master External Device Addressing

A master transaction starts with a Start condition followed by the 7-bit slave address from the DEV_ADDR field of the I2C Master Configuration Register, followed by the R/W bit. Assuming the 8 bits did not collide with another master, an ACK/NACK response bit is expected. If an ACK is received, the slave device exists and the transaction proceeds. If a NACK is received, the slave device is presumed to not exist and the transaction is aborted with a Stop and an MA_NACK status in the I2C Interrupt Status Register, and an optional MA_NACK interrupt to the Interrupt Controller if enabled in MA_NACK of I2C Interrupt Enable Register. In this case, the transaction is not automatically retried and it is up to software to retry if needed.

If the master transaction addresses the slave address of the Tsi721, the slave logic will respond correctly as the target device.

11.4.4 Master Peripheral Addressing

Some devices, such as EEPROMs, require a peripheral address to be specified to set a starting position in their memory or address space for the read or write. The Tsi721 supports transactions with 0, 1, or 2 bytes of peripheral address. Because this is device dependent, the correct setting for the target device must be set in PA_SIZE of the I2C Master Configuration Register; otherwise, the transaction protocol will not be correct.

The peripheral address is also set in the I2C Master Configuration Register when the transaction is started. If the transaction is a read, the Tsi721 must switch the I2C bus protocol to read mode following the peripheral address (sending the peripheral address requires write mode). To do this, a Restart condition is generated, followed by a repeat of the slave address to readdress the device in read mode. Because the bus was not released by the Restart, this phase is not subject to the arbitration timer, therefore any mismatch on the bus aborts the transaction with a MA_COL interrupt. This restart is not required if there is no peripheral address status in the I2C Interrupt Status Register. An optional interrupt can also be sent to the Interrupt Controller if enabled in MA_COL of the I2C Interrupt Enable Register.

11.4.5 Master Data Transactions

After the peripheral address phase, if any, 0 to 4 bytes of data are read or written, followed by the Stop condition. The number of bytes to be transferred is set in the SIZE field of the I2C Master Control Register when the operation is started, the type of transaction (read or write) is set in the WRITE field of the same register, and the order of byte transfer is set in the DORDER field of the I2C Master Configuration Register.

For a write transaction, bytes are taken from the I2C Master Transmit Data Register based on DORDER and sent to the target device. Each byte must be ACKed by the device. If a NACK is received, the transaction is aborted with a Stop, an MA_NACK interrupt status in the I2C Interrupt Status Register. An optional interrupt can also be sent to the Interrupt Controller if enabled in MA_NACK of the I2C Interrupt Enable Register.

For a read transaction, bytes are received from the target device and placed in the I2C Master Receive Data Register based on DORDER. Each byte is ACKed by the Tsi721, except for the final byte which is NACKed to tell the target device to stop sending data, followed by a Stop condition to idle the bus.

When a transaction is successfully completed, the MA_OK interrupt status is updated in the I2C Interrupt Status Register. An optional interrupt can also be sent to the Interrupt Controller if enabled in MA_OK of the I2C Interrupt Enable Register.

⚠ If the Tsi721 encounters a fundamental reset while it is writing to an EEPROM, the write will not be completed and the data at the target EEPROM address may be corrupted.

11.5 Tsi721 as I2C Slave

The Tsi721 can operate as a slave device on the I2C bus. An external master device places a transaction on the bus with a device address that matches that programmed in the SLV_ADDR field of the I2C Slave Configuration Register, or matches the fixed SMBus Alert Response address. The external master can then read or write to the Tsi721 through a small block of 256 addresses called the Tsi721 peripheral address space, and do the following:

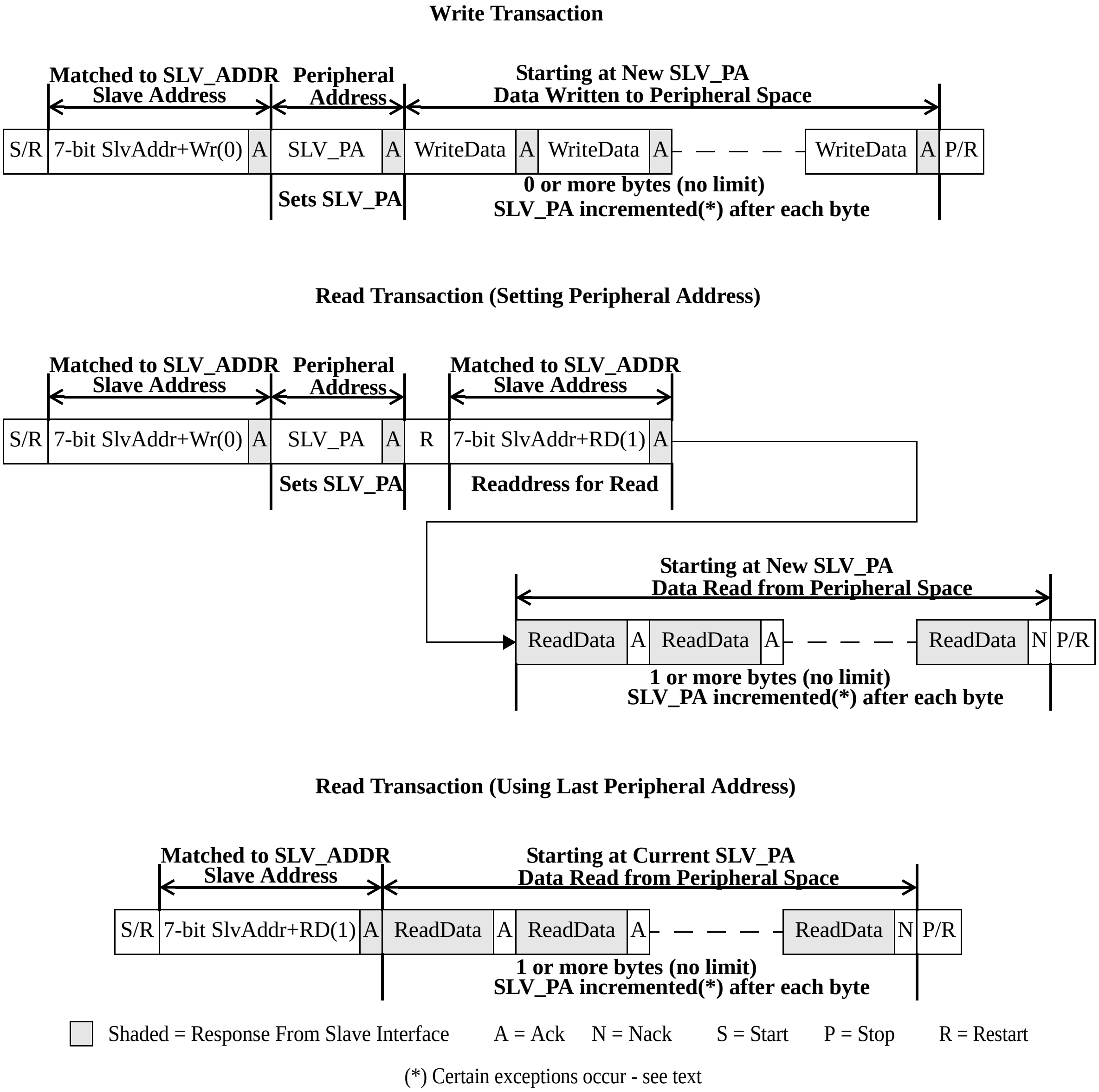
- Directly access limited status, read and write mailboxes
- Configure some options related to the slave access
- Indirectly read or write any other register in the Tsi721 that is accessible through the register bus

Figure 49 shows the bus protocols for accessing the Tsi721 as a slave device. The general procedure requires the external master to address the Tsi721, set the peripheral address to a position within the Tsi721 peripheral address space, then write or read some number of bytes. A write is terminated with a Stop or Restart, and a read is ended when the master responds to a byte with a NACK. There is no limit to the number of bytes that can be read or written in one transaction. The Tsi721 increments the peripheral address pointer after each byte, and wraps within the 256 space (see Slave Peripheral Addressing). Read and write transactions can be mixed by the external master issuing a Restart instead of a Stop, then sending a new transaction that addresses the Tsi721 (all writes must include the peripheral address byte).

At the completion of any slave transaction, either the SA_OK or SA_FAIL interrupt status is updated in the I2C Interrupt Status Register. An optional interrupt can be sent to the Interrupt Controller, if enabled in SA_OK or SA_FAIL of the I2C Interrupt Enable Register, to alert the processor/host that an external device accesses the peripheral address space.

In addition, either the SA_WRITE or SA_READ interrupt status is updated in the I2C Interrupt Status Register if a read or write to the internal register space was triggered by the access. An optional interrupt can also be sent to the Interrupt Controller if enabled in SA_WRITE or SA_READ of the I2C Interrupt Enable Register The SA_FAIL interrupt indicates the slave transaction encountered an error. An SA_FAIL includes the slave logic detecting a collision when it was responding with data or an ACK/NACK, or one of the timeouts triggering and aborting the transaction (see Timeouts). If no error condition occurred, then the SA_OK interrupt is triggered.

Figure 49: Transaction Protocols for Tsi721 as Slave


Write Transaction
Matched to SLV_ADDR Slave Address
Peripheral Address
Starting at New SLV_PA Data Written to Peripheral Space
S/R
7-bit SlvAddr+Wr(0)
A
SLV_PA
A
WriteData
A
WriteData
A
WriteData
A
P/R
Sets SLV_PA
0 or more bytes (no limit)
SLV_PA incremented(*) after each byte
Read Transaction (Setting Peripheral Address)
Matched to SLV_ADDR Slave Address
Peripheral Address
Matched to SLV_ADDR Slave Address
S/R
7-bit SlvAddr+Wr(0)
A
SLV_PA
A
R
7-bit SlvAddr+RD(1)
A
Sets SLV_PA
Readdress for Read
Starting at New SLV_PA Data Read from Peripheral Space
ReadData
A
ReadData
A
ReadData
N
P/R
1 or more bytes (no limit)
SLV_PA incremented(*) after each byte
Read Transaction (Using Last Peripheral Address)
Matched to SLV_ADDR Slave Address
Starting at Current SLV_PA Data Read from Peripheral Space
S/R
7-bit SlvAddr+RD(1)
A
ReadData
A
ReadData
A
ReadData
N
P/R
1 or more bytes (no limit)
SLV_PA incremented(*) after each byte
Shaded = Response From Slave Interface
A = Ack
N = Nack
S = Start
P = Stop
R = Restart
(*) Certain exceptions occur - see text

11.5.1 Slave Clock Stretching

When the Tsi721 is a slave, the external master generates the I2C clock (I2C_SCL). If the Tsi721 must access the internal register bus, I2C_SCL is held low until data is available on a register read, or until a register write completes.

11.5.2 Slave Device Addressing

The Tsi721 supports 7-bit device addressing. The device address of the Tsi721 is set in the SLV_ADDR field of the I2C Slave Configuration Register. For the Tsi721 to respond to an external master, the slave address on the bus must match either the address in the SLV_ADDR field, or the SMBus alert response address (see SMBus Alert Response Protocol Support. However, an address of all zeros (0000000) will never trigger a response because this address is used for the START byte and General Call protocol. Neither the START byte or General Call protocol are supported by the Tsi721.

The SLV_ADDR field can be changed at any time, either using the boot load sequence or by the processor/host. At power-up, the 7-bit device address (defined by SLV_ADDR) is loaded from the I2C_SA signals. Changing the SLV_ADDR field will not change the value of the lower 2 bits unless those bits are unlocked by first setting the SLV_UNLK field in the I2C Slave Configuration Register.

⚠ If the Tsi721 masters an I^{2}C transaction and the device address matches the 7-bit address programmed by SLV_ADDR, the Tsi721 will respond to its own transaction. This is the only method that will allow software to write to any of the externally visible registers that are read-only from the internal register bus.

11.5.3 Slave Peripheral Addressing

The Tsi721 peripheral space comprises an addressable range of 256 bytes (from 0x00 to 0xFF) that can be directly read and written by an external I2C master device. When an external master sets the peripheral address, this sets a pointer (viewable in the SLV_PA field of the I2C Access Status Register) maintained in the Tsi721 that determines where bytes read and written by the external master are within the peripheral address space.

This 256-byte space is mapped to the Externally Visible Registers within the I2C Interface; these registers all start with EXI2C (see Externally Visible I2C Internal Write Address Register). The EXI2C registers can be accessed either directly by an external master using the addresses in the peripheral address space, or by the processor/host using the internal register bus addresses. Depending on how the registers are accessed also defines their properties: for example, some registers are read-only through the peripheral address space but read/write through the internal register bus, and vice-versa.

The next section discusses the mapping between the peripheral address space and the externally visible registers.

11.5.4 External I2C Register Map

Table 70 lists the register map that is visible to external I2C devices. The lowest peripheral address maps to the LSB of the register, while the highest peripheral address maps to the MSB of the register.

The external master can set the peripheral address to any location in the 256-byte range. If that byte maps to a register, and the byte is read or written, then the specific byte within the register is read or written. These reads and writes are not through the internal address bus but are instead local to the I2C Interface. If the peripheral address does not map to a register, then a read will return 0 and a write will have no effect except to increment the peripheral address.

When a byte is read or written, the peripheral address is automatically incremented to the next value, with three exceptions listed in the table (addresses 0x07, 0x17, and 0xFF).

Table 70: Externally Visible I^{2}C Register Map

Tsi721 Peripheral Address Range	Mapped Register	Description
0x00–0x03 R/W	Externally Visible I2C Internal Write Address Register	Specifies the 4-byte aligned internal register address for the register bus write access performed when EXI2C_REG_WDATA is written.

Table 70: Externally Visible I^{2}C Register Map *(Continued)*

Tsi721 Peripheral Address Range	Mapped Register	Description
0x04–0x07 R/W	Externally Visible I2C Internal Write Data Register	Specifies the data to write to the internal register address held in EXI2C_REG_WADDR. Side effects: When address 0x07 (the MSB) is written, the data in this register is written to the internal register address held in EXI2C_REG_WADDR, and the peripheral address is returned to 0x04 (the LSB). This allows consecutive internal registers to be written in one transaction without resetting the peripheral address. Note: If 0x07 is read, the peripheral address increments to 0x08; if written, the peripheral address does not increment.
0x08–0x0F Read-Only	Reserved	This range does not map to any registers.
0x10–0x13 R/W	Externally Visible I2C Internal Read Address Register	Specifies the 4-byte aligned internal register address for the register bus read access performed when EXI2C_REG_RDATA is read.
0x14–0x17 Read-Only	Externally Visible I2C Internal Read Data Register	Returns the data read from the internal register address held in EXI2C_REG_RADDR. Side effects: When address 0x14 (the LSB) is read, the data in this register is loaded from the internal register address held in EXI2C_REG_RADDR, before the data is returned on the I2C bus. When peripheral address 0x17 (the MSB) is read, the peripheral address is returned to 0x14 (the LSB). This allows consecutive internal registers to be read in one transaction without resetting the peripheral address. Note: If 0x17 is written, the peripheral address increments to 0x18; if written, the peripheral address does not increment.
0x18–0x1F Read-Only	Reserved	This range does not map to any registers.
0x20–0x23 Read-Only	Externally Visible I2C Slave Access Status Register	Returns status information on accesses performed by external devices, on the incoming/outgoing mailboxes and on the state of the alert response flag.
0x24–0x27 R/W	Externally Visible I2C Internal Access Control Register	Provides control information on how the Tsi721 handles internal register accesses through the EXI2C_REG_RDATA and EXI2C_REG_WDATA registers.
0x28–0x7F Read-Only	Reserved	This range does not map to any registers.
0x80–0x83 Read-Only	Externally Visible I2C Status Register	Returns a summary of the internal status of the Tsi721.
0x84–0x87 R/W	Externally Visible I2C Enable Register	Enables the bits in the status summary, Externally Visible I2C Status Register, to set the alert flag in the EXI2C_ACC_STAT register.

Table 70: Externally Visible I^{2}C Register Map *(Continued)*

Tsi721 Peripheral Address Range	Mapped Register	Description
0x88–0x8B Read-Only	Reserved	This range does not map to any registers.
0x90–0x93 Read-Only	Externally Visible I2C Outgoing Mailbox Register	This register allows a RapidIO enabled processor to transfer a 32-bit message to an external I2C master.
0x94–0x97 R/W	Externally Visible I2C Incoming Mailbox Register	This register allows an external I2C master to transfer a 32-bit message to a RapidIO enabled processor.
0x98–0xFF Read-Only	Reserved	This range does not map to any registers. Side effects: When peripheral address 0xFF is read or written, the peripheral address wraps back to 0x00.

11.5.5 Slave Write Data Transactions

An external master must set the peripheral address as part of a write transaction before transferring any data. The effect of the written data depends on the register in which the peripheral address maps (see Table 70). Usually, the peripheral address is incremented after each byte such that consecutive bytes are written into increasing addresses within the peripheral address space. Certain exceptions exist, however, as indicated in Table 70. In addition, a write to the most significant byte of the Externally Visible I2C Internal Write Address Register (peripheral address 0x07) triggers a write to a register on the Tsi721 internal register bus.

11.5.6 Slave Read Data Transactions

An external master is not required to set the peripheral address as part of a read transaction, but can do so by first writing the peripheral address and then issuing a Restart before writing any data. If not set, the read data starts wherever the peripheral address pointer was left by the previous transaction. Because another could have master changed the pointer, IDT recommends that the peripheral address be set at the start of a transaction. Data is returned to the external master from consecutive bytes within the peripheral address space, with certain exceptions indicated in Table 70.

There can be side effects to reading some bytes (for more information, see Table 70 and the EXI2C register descriptions). For example, a read that hits the LSB of the Externally Visible I2C Internal Read Address Register (peripheral address 0x14), also triggers a read from a register on the Tsi721 internal register bus.

11.5.7 Slave Internal Register Accesses

The Tsi721 allows external masters to access all of its internal registers through the externally visible I2C registers.

The address in the register definitions refers to an offset only. The offset must be prefixed with the block address. For information about the Tsi721's complete register addressing, see the Registers Overview section of this document.

The Externally Visible I2C Internal Read Data Register is a special register in that a read operation to the first (least significant) byte of this register through the peripheral address space (0x14) triggers the Tsi721 to perform an internal register read operation using the address in the Externally Visible I2C Internal Read Address Register. When the internal register read operation is completed, the data is first loaded into the Externally Visible I2C Internal Read Data Register, then returned to the external I2C master byte by byte. The Externally Visible I2C Internal Access Control Register controls the internal register read operations and configures when and how the register read is performed.

Likewise, the Externally Visible I2C Internal Write Data Register triggers the Tsi721 to perform an internal register write operation using the address in the Externally Visible I2C Internal Write Address Register. The external device writes data to the Externally Visible I2C Internal Write Data Register through the peripheral space, and when the last (most significant) byte is written (0x07), the register contents is written through the internal register bus to the address in the Externally Visible I2C Internal Write Address Register. The Externally Visible I2C Internal Access Control Registercontrols the internal register write operation and configures when and how the register write is performed.

Internal register accesses can be prohibited by the processor/host through the RD_EN and WR_EN fields in the I2C Slave Configuration Register.

At the completion of a slave transaction that includes a successful read or write to an internal register, a SA_WRITE or SA_READ interrupt status is updated in the I2C Interrupt Status Register. An optional interrupt can also be sent to the Interrupt Controller if enabled in SA_WRITE and SA_READ of I2C Interrupt Enable Register. This allows the processor/host to be aware that an external device is accessing the internal registers of the Tsi721.

11.5.8 Slave Access Examples

This section shows a slave internal register access by an external master. The following abbreviations are used:

- <S> Start condition
- <R> Restart condition
- <SLVA> The 7-bit Tsi721 slave address (that matches SLV_ADDR)
- <PA=#> 8-bit peripheral address (current setting viewable in SLV_PA)
- <A> Acknowledge (ACK)
- <N> Not acknowledge (NACK)
- <P> Stop condition
- <W> Write
- <W> Read (not write)
- <WD=#> 8-bit write data (from master to Tsi721)
- <RD=#> 8-bit read data (from Tsi721 to master)

Also, registers and register fields are referenced by name.

All examples assume that the transactions occur in specific order: only one master is accessing (such that nothing is changed by another master between transactions), and nothing is changed by the processor/host during the transactions.

The following conditions pre-exist: ALERT_FLAG is set in the Externally Visible I2C Slave Access Status Register.

1. External device reads Externally Visible I2C Slave Access Status Register (LSB only). The returned value of 0x01 is the ALERT_FLAG. External device must NACK after the first read byte to stop the transfer.

   I2C Sequence: <S><SLVA><W><PA=0x20><A><R><SLVA><W><RD=0x01><N><P>

   Following the transaction, SLV_PA is 0x21 and interrupt status SA_OK asserts. An optional interrupt can also be sent to the Interrupt Controller if enabled in SA_OK of I2C Interrupt Enable Register.

2. External device reads Externally Visible I2C Status Register (all 4 bytes). Note that the data from this register is returned LSB to MSB. External device must NACK the fourth byte to stop the transfer.

   I2C Sequence: <S><SLVA><W><PA=0x80><A><R><SLVA><W>

   <RD=0x56><A><RD=0x34><A><RD=0x12><A><RD=0x80><N><P>

   Following the transaction, SLV_PA is 0x84 and interrupt status SA_OK asserts. An optional interrupt can also be sent to the Interrupt Controller if enabled in SA_OK of I2C Interrupt Enable Register.

3. External device does an Alert Response request. Because ALERT_FLAG is asserted, the alert response address is ACK'd and the Tsi721 slave address is returned.

   I2C Sequence: <S><0001100><W><A><RD=SLVA+0><N><P>

   Following the transaction, SLV_PA is 0x84 (unchanged from previous transaction) and interrupt status SA_OK asserts. An optional interrupt can also be sent to the Interrupt Controller if enabled in SA_OK of I2C Interrupt Enable Register. The ALERT_FLAG in Externally Visible I2C Slave Access Status Register is cleared because of the successful response.

4. External device writes all 4 bytes of the Externally Visible I2C Enable Register to 0, reads Externally Visible I2C Slave Access Status Register, then does another Alert Response request. The ALERT_FLAG is zero (all enables were cleared), so the alert response address is NACKed.

   I2C Sequence: <S><SLVA><W><PA=0x84><A>

   <WD=0x00><A><WD=0x00><A><WD=0x00><A><WD=0x00><A>

   <R><SLVA><W><PA=0x20><A><R><SLVA><W><RD=0x00><N>

   <R><0001100><W><N><P>

   Following the transaction, SLV_PA is 0x21, Externally Visible I2C Enable Register is 0x00000000 and interrupt status SA_OK asserts. An optional interrupt can also be sent to the Interrupt Controller if enabled in SA_OK of I2C Interrupt Enable Register.

5. External device writes Externally Visible I2C Internal Access Control Register to enable internal register auto-incrementing.

   I2C Sequence: <S><SLVA><W><PA=0x24><A><WD=0xAC><A><P>

   Following the transaction, SLV_PA is 0x25, Externally Visible I2C Internal Access Control Register is 0x000000AC and interrupt status SA_OK asserts. An optional interrupt can also be sent to the Interrupt Controller if enabled in SA_OK of I2C Interrupt Enable Register.

6. External device sets up I2C_SCL Low and Arbitration Timeout Register address (0x49354) in EXI2C_REG_WADDR, then writes three registers back-to-back with 0x11223344, 0x55667788, and 0x99AABBCC. Because of the register auto-increment, and because the PA auto-wraps from 0x07 to 0x04, the writes can be completed in a stream. Note that data is written from LSB to MSB.

   I2C Sequence: <S><SLVA><W><PA=0x00><A>
   <WD=0x54><A><WD=0x93><A><WD=0x04><A><WD=0x00><A><R>
   <0x44><A><0x33><A><0x22><A><0x11><A>

   <0x88><A><0x77><A><0x66><A><0x55><A>

   <0xCC><A><0xBB><A><0xAA><A><0x99><A><P>

   Following the transaction, SLV_PA is 0x04, Externally Visible I2C Internal Write Address Register is 0x00049360, I2C_SCL Low and Arbitration Timeout Register is 0x11223344, I2C Byte/Transaction Timeout Register is 0x55667788, and I2C Boot and Diagnostic Timer I2C_BOOT_DIAG_TIMER is 0x8000BBCC (reserved fields stay zero) and interrupt status SA_OK and SA_WRITE assert. An optional interrupt can also be sent to the Interrupt Controller if enabled in the I2C Interrupt Enable Register.

7. External device sets up I2C_SCL Low and Arbitration Timeout Register address (0x49354) in Externally Visible I2C Internal Read Address Register, then reads three registers back-to-back. Because of the register auto-increment, and because the PA auto-wraps from 0x17 to 0x14, the reads can be completed in a stream. Note that data is read from LSB to MSB.

I2C Sequence: <S><SLVA><W><PA=0x10><A>
<WD=0x54><A><WD=0x93><A><WD=0x04><A><WD=0x00><A><R>
<S><SLAV><W><A>
<RD=0x44><A><RD=0x33><A><RD=0x22><A><RD=0x11><A>
<RD=0x88><A><RD=0x77><A><RD=0x66><A><RD=0x55><A>
<RD=0xCC><A><RD=0xBB><A><RD=0x00><A><RD=0x80><N><P>

Following the transaction, SLV_PA is 0x14, Externally Visible I2C Internal Read Address Register is 0x00049360, and interrupt status SA_OK and SA_READ assert. An optional interrupt can also be sent to the Interrupt Controller if enabled in SA_OK and SA_READ of the I2C Interrupt Enable Register.

8. External device writes Externally Visible I2C Internal Access Control Register to disable internal register auto-incrementing.

   I2C Sequence: <S><SLVA><W><PA=0x24><A><WD=0xA0><A><P>

   Following the transaction, SLV_PA is 0x25, Externally Visible I2C Internal Access Control Register is 0x00000A0, and interrupt status SA_OK asserts. An optional interrupt can also be sent to the Interrupt Controller if enabled in SA_OK of I2C Interrupt Enable Register.

9. External device sets up I2C_SCL Low and Arbitration Timeout Register address (0x49354) in Externally Visible I2C Internal Read Address Register, then reads same register three times. The register address no longer auto-increments, but the PA still auto-wraps from 0x17 to 0x14, so the reads can be completed in a stream. Note that data is read from LSB to MSB.

   I2C Sequence: <S><SLVA><W><PA=0x10><A>
   <WD=0x54><A><WD=0x93><A><WD=0x04><A><WD=0x00><A><R>
   <S><SLAV><W><A>

   <RD=0x44><A><RD=0x33><A><RD=0x22><A><RD=0x11><A>

   <RD=0x44><A><RD=0x33><A><RD=0x22><A><RD=0x11><A>

   <RD=0x44><A><RD=0x33><A><RD=0x22><A><RD=0x11><N><P>

   Following the transaction, SLV_PA is 0x14, Externally Visible I2C Internal Read Address Register is 0x00049354, and interrupt status SA_OK and SA_READ assert. An optional interrupt can also be sent to the Interrupt Controller if enabled in SA_OK and SA_READ of I2C Interrupt Enable Register.

11.5.9 Resetting the I2C Slave Interface

The I2C slave interface is reset by two conditions: a fundamental reset or the detection of a START condition.

When a fundamental reset is applied, the I2C slave interface immediately returns to the idle state. Any active transfer to or from the Tsi721 when the reset is asserted is interrupted. All registers are initialized by a fundamental reset.

As required by the *I^{2}C Specification*, the Tsi721 resets its bus interface logic on receipt of a START or repeated START condition such that it anticipates receiving a device address phase, even if the START condition is not positioned according to the proper format. The I2C registers, however, are not reset.

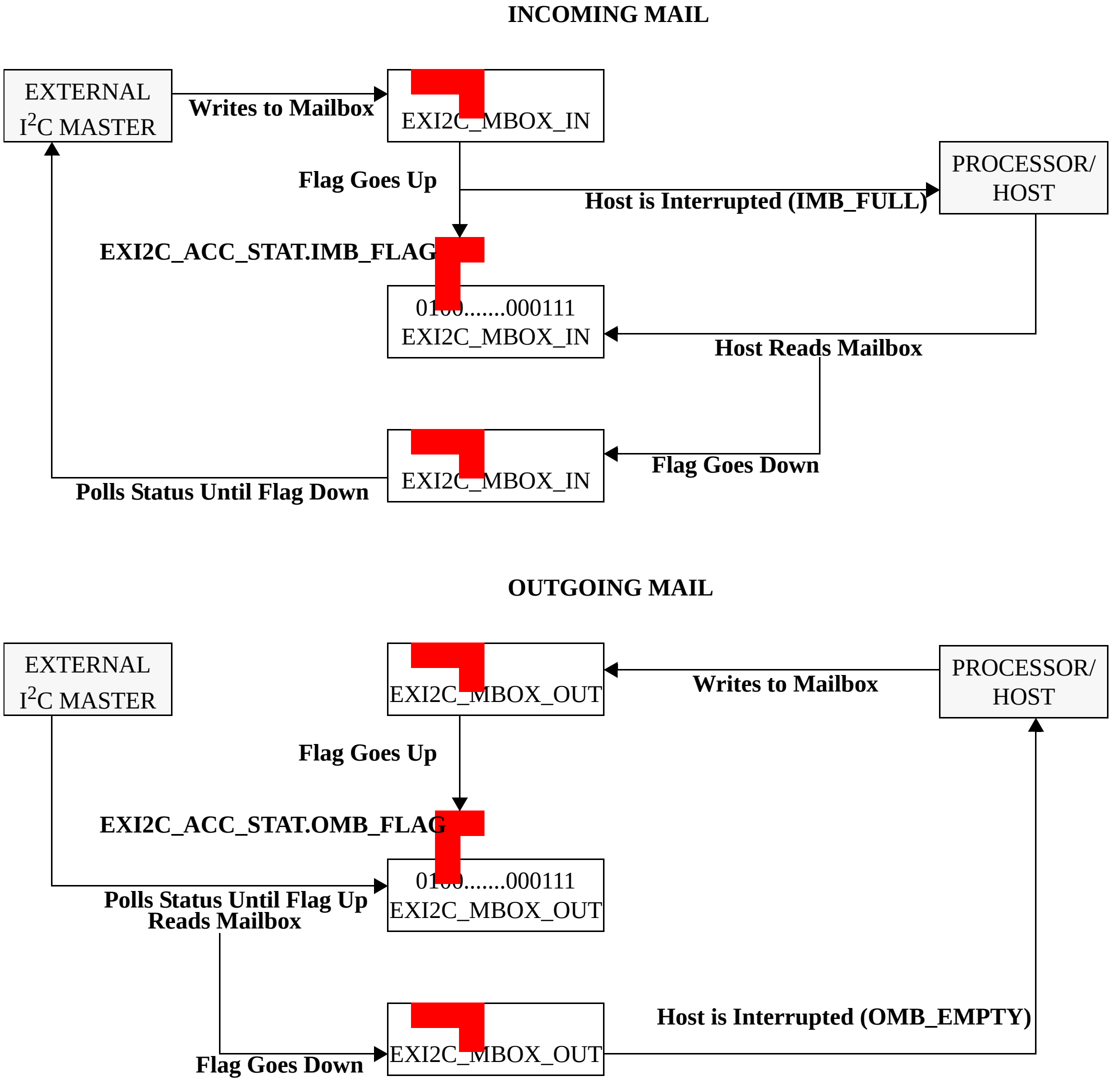
11.6 Mailboxes

As part of the peripheral address space on the Tsi721, the following registers act as I2C mailboxes for communicating information between an external I2C master and a processor or host:

- Externally Visible I2C Incoming Mailbox Register for data incoming from an external I2C master to the processor or host.
- Externally Visible I2C Outgoing Mailbox Register for data outgoing from the processor or host to an external I2C master.

In addition, flags in the Externally Visible I2C Slave Access Status Register are accessible by an external host to examine the mailbox status. Figure 50 shows the use of I2C mailboxes.

Figure 50: I2C Mailbox Operation



The incoming and outgoing I2C mailbox registers are discussed further in the following sections.

11.6.1 Incoming Mailbox

To send data to the processor/host, an external I2C master writes data to the incoming mailbox register, EXI2C_MBOX_IN, through the Tsi721 slave interface. When the Stop condition is seen (indicating the external master is finished writing to the mailbox), the slave interface sets the IMB_FLAG in the Externally Visible I2C Slave Access Status Register, and the processor/host is interrupted to let it know the mailbox is full by setting IMB_FULL status in the I2C Interrupt Status Register. An optional interrupt can be sent to the Interrupt Controller if enabled in IMB_FULL of the I2C Interrupt Enable Register.

In response to the interrupt, the processor/host reads the incoming mailbox to retrieve the data. This process of reading the register clears the IMB_FLAG in the Externally Visible I2C Slave Access Status Register. The external I2C master can poll the status register through the slave interface, and when it sees the flag go to 0, it knows the processor/host read the data and it is safe to write more. If the external I2C master writes more data before earlier data is read, the old data is overwritten. In this case, depending on timing, the processor/host could read a mixture of old and new data.

11.6.2 Outgoing Mailbox

To send data to an external I2C master, the processor/host writes data to the outgoing mailbox register, Externally Visible I2C Outgoing Mailbox Register. This sets the OMB_FLAG in the Externally Visible I2C Slave Access Status Register which the external I2C master can poll through the slave interface. When the flag goes up (1), the external I2C master reads the outgoing mailbox register through the slave interface.

Once the Stop condition is seen (indicating the external master is finished reading the mailbox), the slave interface clears the OMB_FLAG in the Externally Visible I2C Slave Access Status Register, and interrupts the processor/host with an OMB_EMPTY in the I2C Interrupt Status Register to let it know the mailbox was read and it is safe to write more data. An optional interrupt can be sent to the Interrupt Controller if enabled in OMB_EMPTY of I2C Interrupt Enable Register. If the processor/host writes more data to the mailbox before the data was read, the old data is overwritten. In this case, depending on timing, the external I2C master could read a mixture of old and new data.

11.7 SMBus Support

The I2C interface provides limited functionality for SMBus applications. The Tsi721 can act as an SMBus Host and communicate to other SMBus slave devices through a subset of the SMBus protocols (see SMBus Protocol Support).

As a host, the Tsi721 cannot effectively receive a SMBus Host Notify message sent by another non-host SMBus device acting as a master. In addition, the Tsi721 cannot effectively act as a non-host SMBus device and receive commands from an external SMHost. Although the Tsi721 will respond as a slave to the SMBus protocols, they are processed relative to the slave interface functionality. The SMBus command code is assumed to be a peripheral address, and data written to or read through the slave interface will depend on the peripheral address selected.

11.7.1 Unsupported SMBus Features

The Tsi721 does not support the following SMBus features:

- Non-host response to external SMBus host protocols, except for Alert Response Protocol
- Address Resolution Protocol (ARP) or any related commands
- SMBSUS# (suspend mode signal pin)
- SMBALERT# (alert response signal pin). External devices can signal the alert through the Tsi721's INTn or GPIO input signals. Software can then use the Alert Response Address protocol to determine the source of the alert.
- Packet Error Code (PEC). The Tsi721 does not generate, check nor expect PECs
- Process Call command
- Block write command with more than 4 data bytes
- Block read command
- Block write-block read process call command
- SMBus host notify protocol as a SMBus host device in slave mode

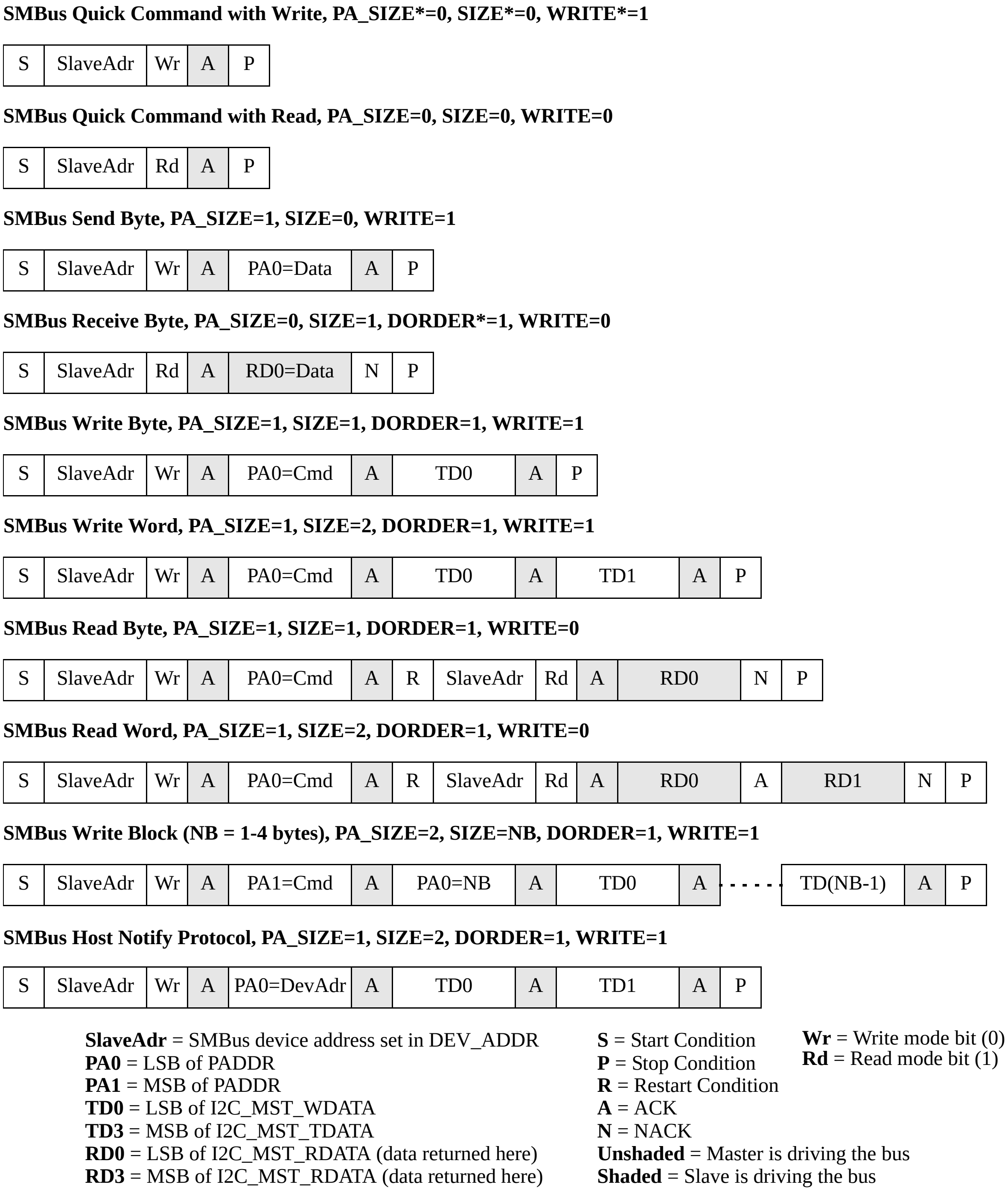
11.7.2 SMBus Protocol Support

The Tsi721 master interface functionality supports a subset of the SMBus Protocols (see Figure 51). In all cases, the Tsi721 masters a transaction to another SMBus device. All register and register field references are to the following I2C master interface registers:

- I2C Master Configuration Register
- I2C Master Control Register
- I2C Master Receive Data Register
- I2C Master Transmit Data Register

Use of the Quick Command with Read requires the target device to support the command. Under normal I^{2}C protocol, the slave device returns data on the bus following the ACK, so the master could not generate the required Stop condition. The target device must release the bus following the ACK if it is responding to this command.

Figure 51: SMBus Protocol Support



SMBus Quick Command with Write, PA_SIZE*=0, SIZE*=0, WRITE*=1

S	SlaveAdr	Wr	A	P

SMBus Quick Command with Read, PA_SIZE=0, SIZE=0, WRITE=0

S	SlaveAdr	Rd	A	P

SMBus Send Byte, PA_SIZE=1, SIZE=0, WRITE=1

S	SlaveAdr	Wr	A	PA0=Data	A	P

SMBus Receive Byte, PA_SIZE=0, SIZE=1, DORDER*=1, WRITE=0

S	SlaveAdr	Rd	A	RD0=Data	N	P

SMBus Write Byte, PA_SIZE=1, SIZE=1, DORDER=1, WRITE=1

S	SlaveAdr	Wr	A	PA0=Cmd	A	TD0	A	P

SMBus Write Word, PA_SIZE=1, SIZE=2, DORDER=1, WRITE=1

S	SlaveAdr	Wr	A	PA0=Cmd	A	TD0	A	TD1	A	P

SMBus Read Byte, PA_SIZE=1, SIZE=1, DORDER=1, WRITE=0

S	SlaveAdr	Wr	A	PA0=Cmd	A	R	SlaveAdr	Rd	A	RD0	N	P

SMBus Read Word, PA_SIZE=1, SIZE=2, DORDER=1, WRITE=0

S	SlaveAdr	Wr	A	PA0=Cmd	A	R	SlaveAdr	Rd	A	RD0	A	RD1	N	P

SMBus Write Block (NB = 1-4 bytes), PA_SIZE=2, SIZE=NB, DORDER=1, WRITE=1

S	SlaveAdr	Wr	A	PA1=Cmd	A	PA0=NB	A	TD0	A	- - - - -	TD(NB-1)	A	P

SMBus Host Notify Protocol, PA_SIZE=1, SIZE=2, DORDER=1, WRITE=1

S	SlaveAdr	Wr	A	PA0=DevAdr	A	TD0	A	TD1	A	P

SlaveAdr = SMBus device address set in DEV_ADDR
PA0 = LSB of PADDR
PA1 = MSB of PADDR
TD0 = LSB of I2C_MST_WDATA
TD3 = MSB of I2C_MST_TDATA
RD0 = LSB of I2C_MST_RDATA (data returned here)
RD3 = MSB of I2C_MST_RDATA (data returned here)

S = Start Condition
P = Stop Condition
R = Restart Condition
A = ACK
N = NACK
Unshaded = Master is driving the bus
Shaded = Slave is driving the bus

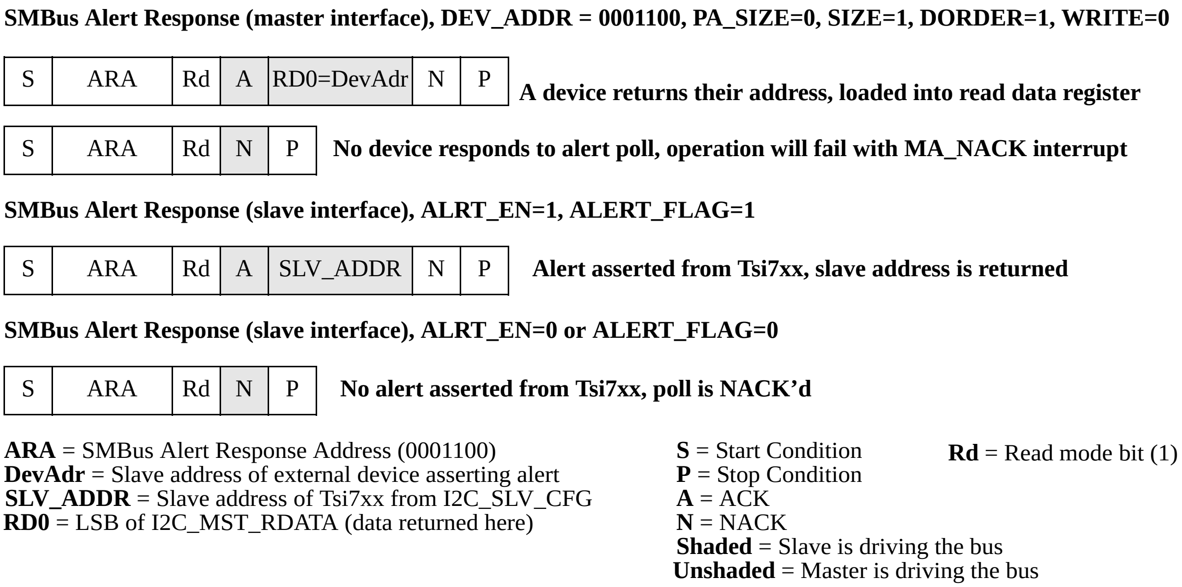
Wr = Write mode bit (0)
Rd = Read mode bit (1)

* For information about SIZE and WRITE, see I2C Master Control Register. For information about PA_SIZE and DORDER, see I2C Master Configuration Register.

11.7.3 SMBus Alert Response Protocol Support

The Tsi721 supports the SMBus Alert Response Protocol as either master or slave. As a master, an external device can be polled using a master read operation. As a slave, the Tsi721 slave interface responds to the Alert Response Address with the Tsi721's slave device address based on the value of ALERT_FLAG in the Externally Visible I2C Slave Access Status Register, if enabled in ALRT_EN of I2C Slave Configuration Register. Once the alert response is given, and the Tsi721's slave device address is returned, the ALERT_FLAG is de-asserted. For the register fields indicated in Figure 52, reference the master interface registers I2C Master Configuration Register, I2C Master Control Register, and I2C Master Receive Data Register, as well as the I2C Slave Configuration Register.

Figure 52: SMBus Alert Response Protocol



SMBus Alert Response (master interface), DEV_ADDR = 0001100, PA_SIZE=0, SIZE=1, DORDER=1, WRITE=0

S	ARA	Rd	A	RD0=DevAdr	N	P	**A device returns their address, loaded into read data register**

S	ARA	Rd	N	P	**No device responds to alert poll, operation will fail with MA_NACK interrupt**

SMBus Alert Response (slave interface), ALRT_EN=1, ALERT_FLAG=1

S	ARA	Rd	A	SLV_ADDR	N	P	**Alert asserted from Tsi7xx, slave address is returned**

SMBus Alert Response (slave interface), ALRT_EN=0 or ALERT_FLAG=0

S	ARA	Rd	N	P	**No alert asserted from Tsi7xx, poll is NACK'd**

ARA = SMBus Alert Response Address (0001100)
DevAdr = Slave address of external device asserting alert
SLV_ADDR = Slave address of Tsi7xx from I2C_SLV_CFG
RD0 = LSB of I2C_MST_RDATA (data returned here)

S = Start Condition
P = Stop Condition
A = ACK
N = NACK
Shaded = Slave is driving the bus
Unshaded = Master is driving the bus

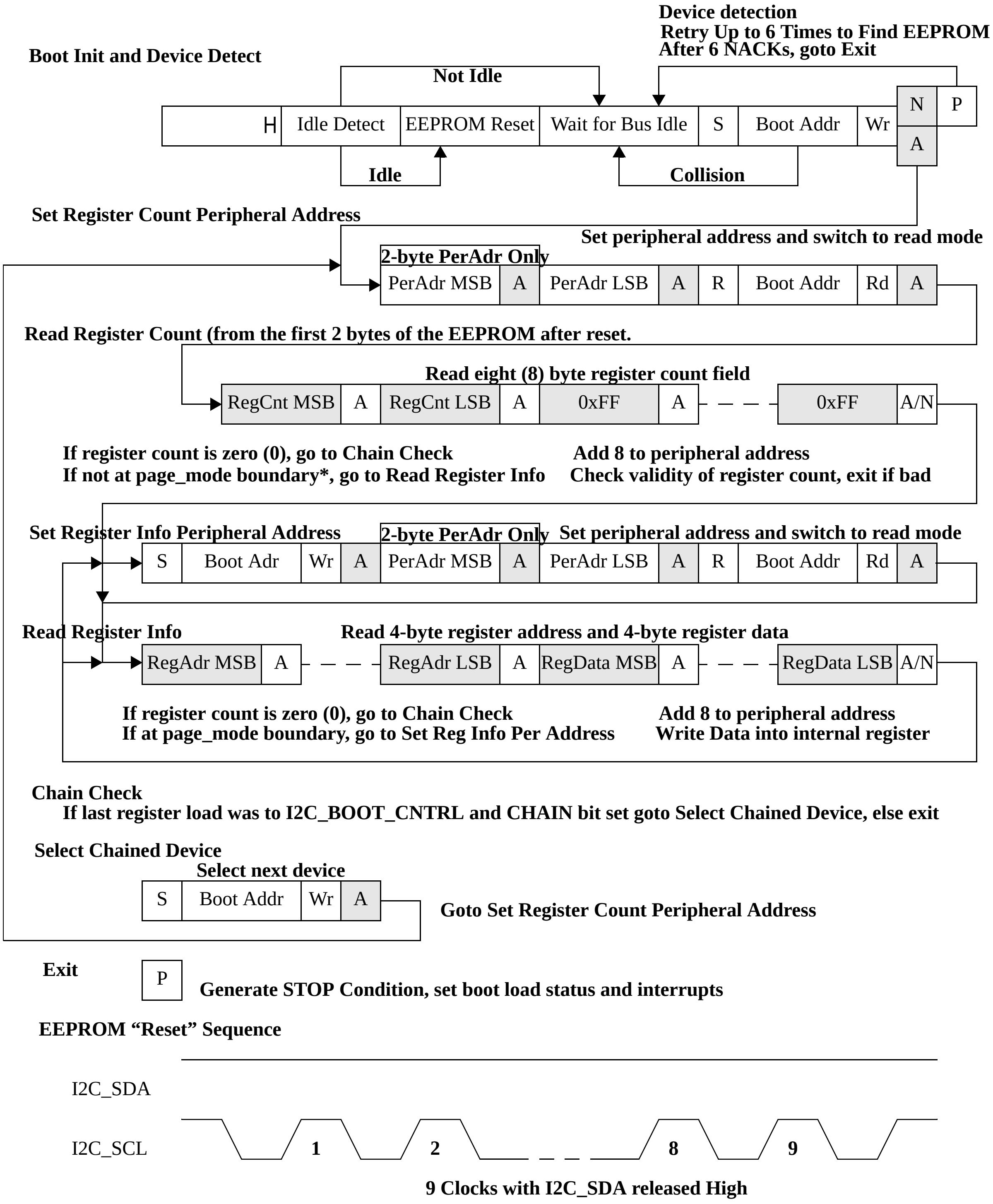
Rd = Read mode bit (1)

* For information about DEV_ADDR, DORDER, and PA_SIZE, see I2C Master Configuration Register. For more information about SIZE and WRITE, see I2C Master Control Register. For more information about ALRT_EN, see I2C Slave Configuration Register.

11.8 Boot Load Sequence

The boot load sequence is controlled by the contents of the I2C Boot Control Register.

Figure 53: Boot Load Sequence


Boot Init and Device Detect
Device detection
Retry Up to 6 Times to Find EEPROM
After 6 NACKs, goto Exit
Not Idle
H
Idle Detect
EEPROM Reset
Wait for Bus Idle
S
Boot Addr
Wr
N
P
A
Idle
Collision
Set Register Count Peripheral Address
Set peripheral address and switch to read mode
2-byte PerAdr Only
PerAdr MSB
A
PerAdr LSB
A
R
Boot Addr
Rd
A
Read Register Count (from the first 2 bytes of the EEPROM after reset.
Read eight (8) byte register count field
RegCnt MSB
A
RegCnt LSB
A
0xFF
A
0xFF
A/N
If register count is zero (0), go to Chain Check
If not at page_mode boundary*, go to Read Register Info
Add 8 to peripheral address
Check validity of register count, exit if bad
Set Register Info Peripheral Address
2-byte PerAdr Only
Set peripheral address and switch to read mode
S
Boot Adr
Wr
A
PerAdr MSB
A
PerAdr LSB
A
R
Boot Addr
Rd
A
Read Register Info
Read 4-byte register address and 4-byte register data
RegAdr MSB
A
RegAdr LSB
A
RegData MSB
A
RegData LSB
A/N
If register count is zero (0), go to Chain Check
If at page_mode boundary, go to Set Reg Info Per Address
Add 8 to peripheral address
Write Data into internal register
Chain Check
If last register load was to I2C_BOOT_CNTRL and CHAIN bit set goto Select Chained Device, else exit
Select Chained Device
Select next device
S
Boot Addr
Wr
A
Goto Set Register Count Peripheral Address
Exit
P
Generate STOP Condition, set boot load status and interrupts
EEPROM "Reset" Sequence
I2C_SDA
I2C_SCL
1
2
8
9
9 Clocks with I2C_SDA released High

* For information about page mode boundary, see PAGE_MODE inI2C Boot Control Register. For information about PA_SIZE, see I2C Master Configuration Register.

11.8.1 Idle Detect

When reset is exited, it is unknown if another master is active. The Idle Detect period determines if the I2C_SCL signal remains high long enough (roughly 50 microseconds) that it is unlikely another master is active. If I2C_SCL is seen low during this period, it is assumed another master is active, the EEPROM Reset phase is skipped, and boot sequence proceeds to the Wait for Bus Idle phase. This detection is performed whether or not the boot sequence is disabled using the I2C_DISABLE power-up signal. If the boot sequence is disabled, the BL_OK interrupt status is asserted immediately in the I2C Interrupt Status Register, and an optional interrupt can be sent to the Interrupt Controller if enabled using BL_OK in the I2C Interrupt Enable Register. If a master transaction is initiated before the idle detect completes, the transaction is started once the idle detect completes.

11.8.2 EEPROM Reset Sequence

The EEPROM reset sequence covers the condition where a hardware reset occurs while a transaction was active on the I2C bus. In this case, because the Tsi721 I2C master may have been reset and stopped generating the I2C_SCL clock, one or more slave devices may be in a hung state where they are expecting a read or write to complete, and may be holding the I2C_SDA signal low, preventing the generation of a STOP or START condition.

To try to force these devices out of their hung state, the Tsi721 will let the I2C_SDA signal stay high and generate 9 clock pulses on the I2C_SCL signal. If no device was hung, this should not cause any problems because all devices are looking for a START condition. If a device was in the middle of a receiving a byte, the remainder of the byte will appear to have all 1s, and the device will be able to generate an ACK or NACK. It could look to the device as if part of another byte is being sent, but because this is the master transmitting part of the protocol, the device will have released its control on the I2C_SDA signal, so the master can force a START or STOP condition, even in the middle of the byte. If a device was in the middle of sending a byte, the clocks pulses will allow it to finish the transmission. The I2C_SDA signal left high by the master (the Tsi721) will appear as a NACK to the device and it will not try to transmit another byte, but will leave the I2C_SDA signal free so that another master can force a START or STOP condition.

This sequence is sent only once after a hard reset, and only if the Idle Detect phase was successful, and the Tsi721 believes it is not interfering with another master.

11.8.3 Wait for Bus Idle

Before attempting to access an EEPROM device, the boot loader will wait for the bus to be idle. This will either be caused by a successful Idle Detect phase, or, if the Idle Detect phase failed, once a STOP condition is seen on the bus, indicating another master released control. In addition, if the I2C_SCL and I2C_SDA signals are both high for longer than the idle detect period while waiting for a STOP condition, the bus is assumed idle and the boot load process will proceed.

11.8.4 EEPROM Device Detection

Once the bus is available, the Tsi721 tries to connect to the EEPROM. A START condition is generated followed by BOOT_ADDR from the I2C Boot Control Register. The upper 5 bits of that field reset to 5’b10100 and the lower 2 bits are sampled from the I2C_SA signals on exit from hard reset.This allows up to four unique Tsi721 devices to boot from different EEPROMs on the same I2C bus. If an ACK is received, which indicates the device is present, the sequence proceeds to the next phase. If there is a bus collision, the loader returns to the Wait for Bus Idle phase because another master owns the bus. If a NACK is received, the process is retried from the Wait for Bus Idle phase up to 6 times in case the device was busy. If six NACKs are received, the boot load is aborted and the BL_FAIL interrupt status is asserted. An interrupt can also be sent to the Interrupt Controller if enabled using BL_FAIL in the I2C Interrupt Enable Register.

11.8.5 Loading Register Data from EEPROM

Once the EEPROM is successfully addressed, the Tsi721 will not release the bus until the boot load is complete. First, the peripheral address is set. The address resets to 0, so the first EEPROM accessed must be loaded from address 0. The peripheral address is either 1 or 2 bytes depending on the state of the I2C_MA pin, which much be set appropriately depending on the type of EEPROM connected.

The boot loader then switches to read mode and reads the first 8 bytes, expecting to find a count of the number of registers to be initialized in the first 2 bytes, followed by 6 bytes of 0xFF. A validity check is completed on this field — if the number of registers exceeds the maximum as described in EEPROM Data Format, or if any of the last 6 bytes are not 0xFF, it is assumed the EEPROM does not contain boot load data, the boot load is aborted and the BL_FAIL interrupt status is updated in the I2C Interrupt Status Register. On these boot load status bits, the optional interrupt can be forwarded to the Interrupt Controller if enabled in the I2C Interrupt Enable Register. If the register count was 0, the boot load is ended successfully and the BL_OK interrupt status is updated. An optional interrupt can also be forwarded to the Interrupt Controller if enabled in the I2C Interrupt Enable Register. For information on the expected EEPROM data format used for boot loading, see EEPROM Data Format.

The boot loader continues by reading eight bytes of data for each register to be loaded, and increments the peripheral address by 8. Depending on the PAGE_MODE field in the I2C Boot Control Register, the peripheral address is periodically reset by issuing a Restart, re-selecting the boot device, and sending the updated peripheral address. On reset, the PAGE_MODE resets to a boundary of eight so that the peripheral address is updated to the device after every register is loaded (for more information, see Accelerating Boot Load). In addition, for 1-byte peripheral addresses, if the BINC bit is 1, then when the peripheral address crosses a 256-byte boundary (that is, when the 1-byte address rolls over to 0x00), the LSB 3 bits of the BOOT_ADDR are incremented and the device is re-addressed. This supports those EEPROMs that use the lower 3 bits of their address as a 256-byte page indicator.

For each block of 8 bytes loaded, the first 4 bytes are the register address on the internal Tsi721 register bus, and the next 4 bytes are the 32-bit data value to be written to the register. No checking is completed for register address or data validity. As soon as all 8 bytes are read, the data is written to the internal address, the peripheral address count is updated, and the register count is decremented. Once the register count reaches 0 the boot load from the current EEPROM is complete, and, unless chaining is invoked, the boot load sequence is complete, a STOP condition is issued to release the bus, and the BL_OK interrupt status is updated. An optional interrupt can also be forwarded to the Interrupt Controller if enabled in the I2C Interrupt Enable Register.

11.8.6 Chaining

The boot loader provides for booting from multiple EEPROMs, or from multiple sections within a single EEPROM (or any combination of both). This process is called chaining. Chaining is invoked during the boot load sequence when three conditions occur together:

- All the registers indicated by the register count are loaded
- The final register loaded was the I2C Boot Control Register
- The value loaded into the I2C Boot Control Register had the CHAIN bit set

If these conditions are met, then the boot load sequence continues using the updated information in the I2C Boot Control Register. This allows all aspects of the boot load to be changed – the device address, the peripheral address, and so forth. When a chain occurs, the boot load sequence addresses the new device and reads a new register count from the peripheral address. This address could be non-zero, therefore on a chain it is possible to start loading from another address in an EEPROM.

On a chain, it is important to set the PSIZE, BINC, and PAGE_MODE fields so they are valid for the new EEPROM; otherwise, the boot load process will most likely be corrupted (for information about these bits, see I2C Boot Control Register).

It may also be required to use the BOOT_UNLK field to change the lower 2 bits of the EEPROM address. By default, the BOOT_UNLK field is not set, so if the BOOT_ADDR field is changed, the lower 2 bits remain at their previous value. This way the power-up reset value is not unintentionally lost. If as part of the chaining process it is required to change those bits (such as if the boot load is being switched to a common EEPROM), then a two-step process is needed. The I2C Boot Control Register should be written once with the BOOT_UNLK field set to 1, then written a second time with the correct information. The lower 2 bits of the BOOT_ADDR field are only allowed to change if the BOOT_UNLK field was a 1 before the register load.

11.8.7 EEPROM Data Format

Table 71 shows the EEPROM data format for boot loading. The first 8 bytes of the EEPROM contain the number of registers to be loaded during the boot procedure. This count is the 16-bit value in EEPROM location 0 (MSB) and location 1 (LSB). The I2C Interface is limited to 255 register loads in 1-byte address mode, and limited to 8K-1 register loads in 2-byte address mode. The remaining 6 bytes (memory locations 2 through 7) must be set to 0xFF or the register count validity check will fail and the boot load will be aborted.

⚠ When 1-byte address mode is selected, any number of registers greater than 255 (0x00FF) aborts the boot load from the EEPROM.

When 2-byte address mode is selected, any number of registers greater than 8191 (8K-1 = 0x1FFF) aborts the boot load from the EEPROM.

The register load data consists of 8-byte fields aligned to 8-byte peripheral address boundaries. The first 4 bytes are the internal register address and the second 4 bytes are the register data. Note that the address and data are ordered from MSB to LSB within increasing peripheral byte addresses.

Table 71: Format for Boot Loadable EEPROM

PerAdr	PerAdr+0	PerAdr+1	PerAdr+2	PerAdr+3
0x0	RegCnt (MSB)	RegCnt (LSB)	0xFF	0xFF
0x4	0xFF	0xFF	0xFF	0xFF
0x8	RegAdr (MSB)	RegAdr	RegAdr	RegAdr (LSB)
0xC	RegData (MSB)	RegData	RegData	RegData (LSB)
0x10	RegAdr (MSB)	RegAdr	RegAdr	RegAdr (LSB)
0x14	RegData (MSB)	RegData	RegData	RegData (LSB)
...	...	...	...	...

As an example, the following shows an EEPROM configured to load two registers and then complete – first the I2C Master Configuration Register at internal address 0x49108, loaded with data value 0x0102_0304; then the I2C Master Transmit Data Register at internal address 0x49114, loaded with data value 0x0506_0708.

Table 72: Sample EEPROM Loading Two Registers

PerAdr	PerAdr+0	PerAdr+1	PerAdr+2	PerAdr+3	Description
0x0	0x00	0x02	0xFF	0xFF	RegCnt = 2, must have 0xFFFF at end
0x4	0xFF	0xFF	0xFF	0xFF	Must be 0xFFFF_FFFF

Table 72: Sample EEPROM Loading Two Registers *(Continued)*

PerAdr	PerAdr+0	PerAdr+1	PerAdr+2	PerAdr+3	Description
0x8	0x00	0x04	0x91	0x08	RegAdr = 0x49108 I2C Master Configuration Register
0xC	0x01	0x02	0x03	0x04	RegData = 0x0102_0304
0x10	0x00	0x04	0x91	0x14	RegAdr = 0x49114 I2C Master Transmit Data Register
0x14	0x05	0x06	0x07	0x08	RegData = 0x0506_0708
>= 0x18	xx	xx	xx	xx	Unused by Boot

As a second example, the following shows an EEPROM configured to first load the I2C Master Configuration Register then chain to address 0x80 in the same EEPROM and load the I2C Master Transmit Data Register. Note that the chain requires loading the I2C Boot Control Register. The new peripheral address is 0x80 >> 3 = 0x10, because the 3 LSBs must be zero and are not part of the PADDR field.

Table 73: Sample EEPROM With Chaining

PerAdr	PerAdr+0	PerAdr+1	PerAdr+2	PerAdr+3	Description
0x0	0x00	0x02	0xFF	0xFF	RegCnt = 2, must have 0xFFFF at end
0x4	0xFF	0xFF	0xFF	0xFF	Must be 0xFFFF_FFFF
0x8	0x00	0x04	0x91	0x08	RegAdr = 0x49108 I2C Master Configuration Register
0xC	0x01	0x02	0x03	0x04	RegData = 0x0102_0304
0x10	0x00	0x04	0x91	0x40	RegAdr = 0x49140 I2C Boot Control Register
0x14	0x80	0x50	0x00	0x10	RegData = 0x8050_0010 CHAIN = 1 BOOT_ADDR = 1010000 PADDR = 0x10
0x18–0x7F	xx	xx	xx	xx	Unused by Boot
0x80	0x00	0x01	0xFF	0xFF	RegCnt = 1, must have 0xFFFF at end
0x84	0xFF	0xFF	0xFF	0xFF	Must be 0xFFFF_FFFF

Table 73: Sample EEPROM With Chaining *(Continued)*

PerAdr	PerAdr+0	PerAdr+1	PerAdr+2	PerAdr+3	Description
0x88	0x00	0x04	0x91	0x14	RegAdr = 0x49114 I2C Master Transmit Data Register
0x8C	0x05	0x06	0x07	0x08	RegData = 0x0506_0708
>= 0x90	xx	xx	xx	xx	Unused by Boot

11.8.8 I2C Boot Time

The time required to perform an I2C boot depends on the following:

- The number of registers that require configuration
- The number of devices contending for EEPROM or I2C bus access
- The number of chaining operations
- The clocking speeds of the master devices

Because many of these parameters are outside the control of the Tsi721, the boot time cannot be predicted with complete accuracy.

If there are no other devices contending for bus access, a 1-byte peripheral address is used, no boot acceleration techniques are used, and no retries are required for device detect, then boot time can be estimated as follows:

Boot_Time =

50 us idle detect time +

(9 * ClkPer) EEPROM reset time +

(102 * (RegisterCount + 1) * ClkPer) register load time +

(1 * ClkPer) STOP time

Where:

ClkPer = clock period (resets to 10 us for a 100 kHz clock)

RegisterCount is the sum of number of registers from the Register Count fields in the EEPROM (only one count field unless chaining is involved).

If a 2-byte peripheral address is used, then the “102” constant increases to “111”. The 102 constant comes from the sum of Start + (9-bit boot address) + (9-bit peripheral address) + Restart + (9-bit boot address) + (9-bit data byte * 8 bytes per register = 72 bits) = 101 clocks, but the Start and Restart take an extra 1/2 clock each, so an extra clock cycle is consumed.

For example, if 255 registers are read the boot time is:

Boot_Time = 50us + (90us EEPROM reset) + (10us * 102 * 256 register load) + 10us Stop

Boot_Time = 261,270 us = slightly over 1/4 second

11.8.9 Accelerating Boot Load

If boot load time is a design concern, the following techniques may accelerate the boot load sequence:

1. If the EEPROM supports reading of a large block of data sequentially, change PAGE_MODE in I2C Boot Control Register as the first register load. Depending on the page size, this will reduce the number of times the boot load re-addresses the device and resets the peripheral address. At the limit, if the “infinite” setting were chosen and the device did not wrap on any page boundaries, the 102 constant in the boot time formula in I2C Boot Time would be reduced to 72 cycles per register, with only one address phase first or per chain operation.
2. If the EEPROM supports reading at higher than 100-kHz clock speeds, the timing parameters can be changed during boot load. The success of this depends on the bus properties because the Tsi721 does not contain the Schmitt Triggers or slope controlled outputs needed to guarantee conformance to the 400-kHz high-speed mode. However, many configurations can be interoperable at higher speeds (for information on changing timing parameters, see Bus Timing). Timing parameters are reloaded when a chain operation occurs, so the technique is to program the timing parameters for the higher speed, set up the digital filters if required, and then invoke a chain operation using the same EEPROM but the next peripheral address. Everything from the chain onwards will be mastered at the higher speed.

11.9 Error Handling

The Tsi721 handles a number of I2C errors and reports them with status bits, as summarized in Table 74.

Table 74: I2C Error Handling

Error Cause	Access Type	Tsi721 Response	Interrupt Status Bit (Events)[a]
Master Access Errors			
Master arbitration timeout expired. Tsi721 could not successfully arbitrate for the I2C bus Arbitration is lost during device addressing phase.	Master read or write initiated through I2C Master Control Register	The I2C transaction is aborted.	MA_ATMO
Tsi721 determined that it lost arbitration for the I2C bus after the device addressing phase	Read or Write	The I2C transaction is aborted.	MA_COL
No device ACK’d the slave address, or target device NACK’d a peripheral address or write data byte.	Any read or write access during slave address phase or peripheral address phase, or any write access during the data phase.	Access aborted, STOP generated. The Externally Visible I2C Slave Access Status Register indicates where transaction was on error.	MA_NACK

Table 74: I2C Error Handling *(Continued)*

Error Cause	Access Type	Tsi721 Response	Interrupt Status Bit (Events)[a]
Timeout expired (I2C_SCL Low, Byte or Transaction). Target device was too slow, or some device was interfering with the I2C_SCL signal.	Any transfer to or from the Tsi721	Access aborted. The Externally Visible I2C Slave Access Status Register indicates where transaction was on error. For Byte or Transaction, master issues STOP at first legal opportunity. For I2C_SCL Low, bus is hung, software must recover.	MA_TMO (MSCLTO, MBTTO or MTRTO)
Slave Access Errors			
Peripheral Address selects reserved external address space	Read operation	Peripheral Address byte is acknowledged, 0x00 is returned as data.	SA_OK
	Write operation	Peripheral Address byte is acknowledged. Write data is ignored.	
Peripheral Address selects a defined register, but data burst continues into reserved address	Read operation	0x00 is returned as data.	SA_OK
	Write operation	Write data is ignored	
Programmed register address accesses a non-existent internal register block	Read operation	Read returns 0x00	SA_OK
	Write operation	Write data discarded	SA_OK
Internal register access when disabled	External master read to the EXI2C_REG_RDATA register, or write to the EXI2C_REG_WDATA register	Operation completes, returns existing RDATA or updates WDATA, but no internal register access generated.	SA_OK No SDW/SDR
Timeout expired (I2C_SCL Low, Byte or Transaction). Target device was too slow, or some device was interfering with the I2C_SCL signal.	Any transfer to or from the Tsi721	Slave releases I2C_SDA and I2C_SCL, goes into wait state.	SA_FAIL (SSCLTO, SBTTO or STRTO)
Protocol violation (collision detected)	Read data or Ack/Nack, when slave puts a 1 on the I2C_SDA signal and another device holds the signal to 0.	Slave releases I2C_SDA and I2C_SCL, goes into wait state.	SA_FAIL (SCOL)

This document is confidential and is subject to an NDA.

Table 74: I2C Error Handling *(Continued)*

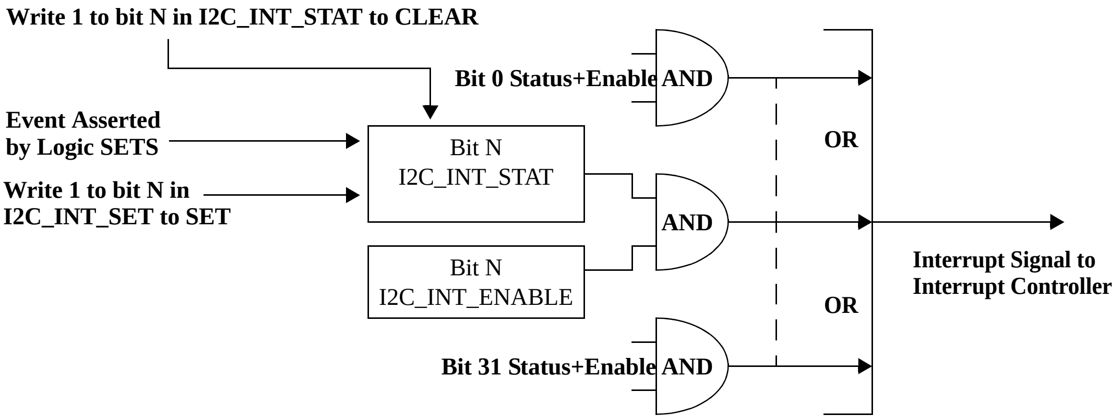
Error Cause	Access Type	Tsi721 Response	Interrupt Status Bit (Events)[a]
Register Initialization Loader Errors			
Failed to find EEPROM	Initialization read	Read operation retried up to 6 times before aborting. If not Ack'ed by the 6th try, status bits set	BL_FAIL (BLNOD)
Size field specifies more than 255 registers to load in 1-byte addressing mode, or 8K-1 registers in 2-byte addressing mode.	Initialization read	Initialization load aborted	BL_FAIL (BLSZ)
Register address selects non-existent register.	Register initialization write	Data discarded	None
Failed to arbitrate for I2C bus during boot load, boot load timer expired.	Initialization read	Initialization load aborted	BL_FAIL (BLTO)
Protocol error during boot load, including bytes 2–7 of a register count not containing 0xFF.	Initialization read	Initialization load aborted	BL_FAIL (BLERR)

a. To determine the setting of the interrupt status bits, see I2C Interrupt Status Register.

11.10 Interrupt Handling

I2C Interrupts are generated as displayed in Figure 54. An I2C event detected by the I2C Interface sets a bit in the I2C Interrupt Status Register to a 1 to assert the interrupt. This bit is then anded with the corresponding bit in theI2C Interrupt Enable Register to determine if that interrupt is enabled. Any enabled interrupt status bit asserts the interrupt output signal to the Interrupt Controller. This signal stays asserted until all enabled bits in the interrupt status register are cleared.

Figure 54: I2C Interrupt Generation


Write 1 to bit N in I2C_INT_STAT to CLEAR
Bit 0 Status+Enable
AND
Event Asserted by Logic SETS
Write 1 to bit N in I2C_INT_SET to SET
Bit N I2C_INT_STAT
OR
AND
Bit N I2C_INT_ENABLE
Interrupt Signal to Interrupt Controller
OR
Bit 31 Status+Enable
AND

The interrupt status bits are cleared by a write-one-to-clear operation to the I2C Interrupt Status Register, provided the interrupt status register was read first. For test purposes, bits in the I2C Interrupt Status Register can also be set by a write-one-to-set operation to the I2C Interrupt Set Register.

A bit that is set in the I2C Interrupt Status Register is cleared by a write-1-to-clear operation only after the register is read first, and then providing another event that would cause the interrupt condition does not occur since the register was last read. For more information, see Events versus Interrupts.

11.11 Events versus Interrupts

Interrupts are generated by I2C events. Figure 55 shows the design of the event and interrupt logic. A single interrupt status bit can be derived from one or more events. The event registers provide control over the individual events that produce the interrupt status. In the diagram, the shaded boxes represent virtual registers. These registers function correctly when read or written but can be constructed from combinational logic as opposed to flip-flops. Whether a register is virtual or not is inconsequential to their behavior from a software perspective. The distinction is displayed only for exactness.

A new event is set in the I2C New Event Register when an event is asserted in the logic, or when a 1 is written to the register (or to the related interrupt bit in the I2C Interrupt Set Register). New events are ORed with the I2C_SNAP_EVENT register (see I2C Event and Event Snapshot Registers) to create the virtual I2C_EVENT register. A snapshot operation occurs when the I2C Interrupt Status Register is read. As a result of the snapshot, the new event register is “copied” to the snapshot register by oring the new events into the current snapshot state, then clearing the new event register. Each event is anded with the corresponding enable bit in the I2C Enable Event Register, and then ored with any other enabled events that are related to a single interrupt status bit. The combined event state becomes the interrupt status bit in the I2C Interrupt Status Register, and is then anded with the corresponding enable in the I2C Interrupt Enable Register. All the enabled interrupt status bits are then ORed together to become the single interrupt signal to the Interrupt Controller.

The new event and snapshot registers separate events that occurred before a read of the interrupt status register from those that occur during or after the read. When a 1 is written to the interrupt status register to clear an interrupt, all related events that are enabled are cleared in the snapshot register. Since events are copied to the snapshot register only when the interrupt status register is read, the read must be completed first for the write 1 to clear to have effect. If no new events have occurred, this write-1-to-clear de-asserts the interrupt status. If a new event occurs, the event remains set in the new event register, so the interrupt status remains set.

For control, software can read and clear the snapshot event bits directly, allowing individual events to be cleared while leaving any new events intact. Software can also select to read or clear events using the new event register. Reading the event register shows the “or” of the new and snapshot, and thus shows whether an event is asserting. Writing a 1 to an event bit clears both the snapshot and new events bits. This clears the event entirely, unless that event is asserting again on the same cycle the clear is completed, which would set it again.

As long as all event enables are set (the reset state), then the behavior is logically as described in the section on “Interrupt Handling”.

Figure 55: I2C Event and Interrupt Logic

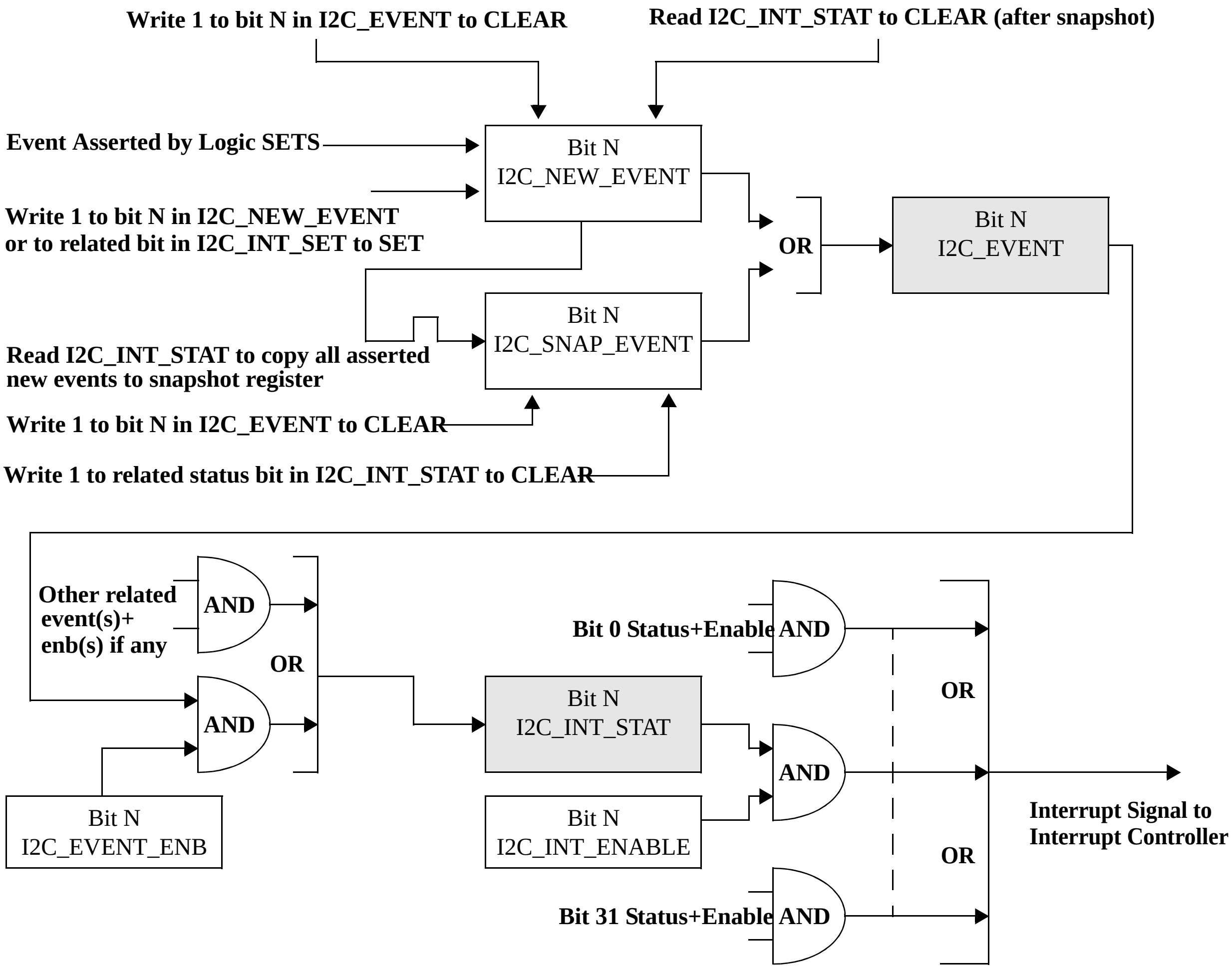

Write 1 to bit N in I2C_EVENT to CLEAR
Read I2C_INT_STAT to CLEAR (after snapshot)
Event Asserted by Logic SETS
Bit N
I2C_NEW_EVENT
Write 1 to bit N in I2C_NEW_EVENT
or to related bit in I2C_INT_SET to SET
OR
Bit N
I2C_EVENT
Bit N
I2C_SNAP_EVENT
Read I2C_INT_STAT to copy all asserted
new events to snapshot register
Write 1 to bit N in I2C_EVENT to CLEAR
Write 1 to related status bit in I2C_INT_STAT to CLEAR
Other related
event(s)+
enb(s) if any
AND
AND
OR
Bit N
I2C_EVENT_ENB
Bit 0 Status+Enable
AND
Bit N
I2C_INT_STAT
AND
Bit N
I2C_INT_ENABLE
Bit 31 Status+Enable
AND
OR
OR
Interrupt Signal to
Interrupt Controller

Table 75 shows the mapping of interrupts in the I2C Interrupt Status Register to the events in the I2C Event and Event Snapshot Registers. Any asserted and enabled event will set the corresponding interrupt status, and clearing an asserted interrupt status bit will clear all the related and enabled events.

Table 75: I2C Interrupt to Events Mapping

Interrupt Status Bit	Events Related to Interrupt
OMB_EMPTY (Outgoing Mailbox Empty)	OMBR (Outgoing Mailbox Read Event)
IMB_FULL (Incoming Mailbox Full)	IMBW (Incoming Mailbox Write Event)
BL_FAIL (Boot Load Fail)	BLTO (Boot Load Timeout Error) BLERR (Boot Load Error Event) BLSZ (Boot Load Size Error Event) BLNOD (Boot Load No Device Event)
BL_OK (Boot Load OK)	BLOK (Boot Load OK Event)

Table 75: I2C Interrupt to Events Mapping *(Continued)*

Interrupt Status Bit	Events Related to Interrupt
SA_FAIL (Slave Access Failed)	SCOL (Slave Collision Detect Event) STRTO (Slave Transaction Timeout Event) SBTTO (Slave Byte Timeout Event) SSCLTO (Slave I2C_SCL Low Timeout Event)
SA_WRITE (Slave Access Write)	SDW (Slave Internal Register Write Done Event)
SA_READ (Slave Access Read)	SDR (Slave Internal Register Read Done Event)
SA_OK (Slave Access OK)	SD (Slave Transaction Done Event)
MA_DIAG (Master Diagnostic Event)	DTIMER (Diagnostic Timer Expired Event) DHIST (Diagnostic History Filling Event) DCMDD (Diagnostic Command Done Event)
MA_COL (Master Collusion)	MCOL (Master Collision Detect Event)
MA_TMO (Master Timeout)	MTRTO (Master Transaction Timeout Event) MBTTO (Master Byte Timeout Event) MSCLTO (Master I2C_SCL Low Timeout Event)
MA_NACK (Master NACK)	MNACK (Master NACK Received Event)
MA_ATMO (Master Arbitration Timeout)	MARBTO (Master Arbitration Timeout Event)
MA_OK (Master Transaction OK)	MTD (Master Transaction Done Event)

11.12 Timeouts

The I2C Interface supports a number of timeout periods to detect a set of error conditions related to I2C operation. These timeouts, and the registers that configure them, include the following:

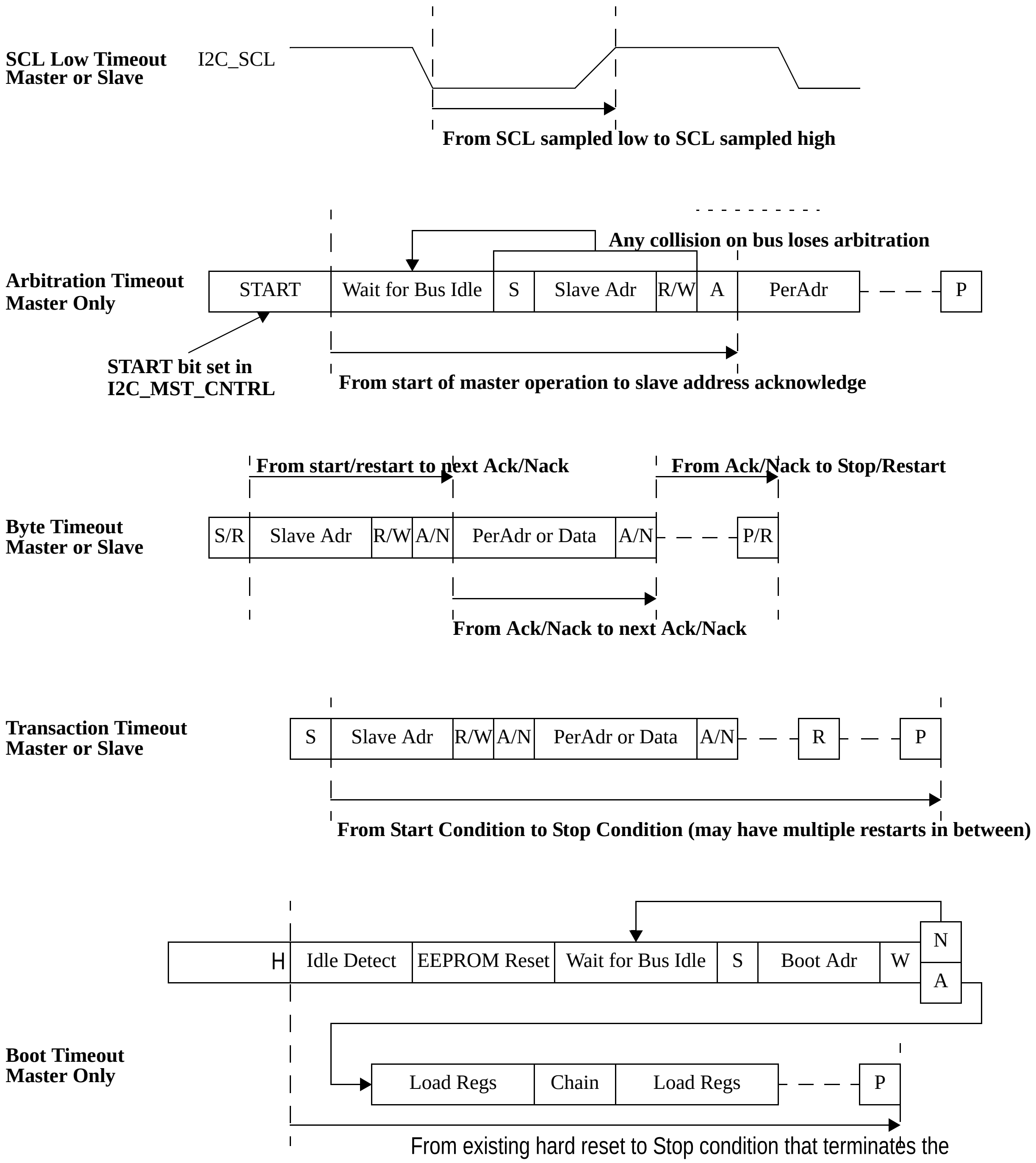
- I2C_SCL low timeout (see I2C_SCL Low and Arbitration Timeout Register) – This timeout detects a situation where a device on the bus is stuck holding the clock low. Because the clock is stuck low, no progress can be made. If enabled, this timeout expiring will set either the SSCLTO or MSCLTO events and result in a SA_FAIL or MA_TMO interrupt status being updated in the I2C Interrupt Status Register (depending on whether a master or slave operation was active). An optional interrupt can be sent to the Interrupt Controller if SA_FAIL or MA_TMO is enabled in the I2C Interrupt Enable Register. This is an extreme failure. With I2C_SCL held low, no Stop condition can be generated. Any operation is aborted, both I2C_SCL and I2C_SDA are released, and both master and slave revert their monitor-for-bus-idle phase. It is up to software to decide how to handle this error. Because any operation was aborted without correct termination (no Stop), an external device could be left in an invalid state.

- Arbitration timeout (see I2C_SCL Low and Arbitration Timeout Register) – This timeout applies only to master transactions initiated by setting the START bit in the I2C Master Control Register. Its purpose is to limit the length of time the master controller tries to gain ownership of the bus. The arbitration timer is disabled once the <Start><Slave Address><Read/Write> have been successfully transmitted without detecting another master attempting a different transaction. If the Tsi721 I2C master subsequently loses ownership of the bus after this phase of the transaction, the transaction is aborted. If the Tsi721 I2C master detects another master corrupting the <Start><Slave Address><Read/Write> bits it transmitted, the Tsi721 I2C master reverts to waiting for bus idle then tries again. The arbitration timeout continues to run in this case. If the arbitration timer expires before ownership is gained and the master is waiting for bus idle, then it aborts the operation and sets the MARBTO event which causes a MA_ATMO interrupt status to be updated in the I2C Interrupt Status Register. An optional interrupt can also be sent to the Interrupt Controller if the MA_ATMO is enabled in the I2C Interrupt Enable Register.

  If the Tsi721 I2C master was in the midst of transmitting the <Slave Address> when the timeout expires, it allows the <Slave Address> to complete. If an ACK or NACK is successfully received, the master continues as if the timeout had not expired. If another I2C master collides with <Slave Address>, the timeout immediately takes effect following the <Slave Address> bit where the collision took place.
- Byte timeout (see I2C Byte/Transaction Timeout Register) – This timeout is disabled on reset. It detects a situation where one or more devices are stretching the clock enough to slow the transfer speed on the bus beyond some limit. This timeout is available primarily to detect a violation of the SMBus TLOW:MEXT time. The response to this timeout expiring depends on the phase of the transfer and whether it is detected by the master or slave interface. For a master transaction, the master continues to generate clocks until the next bit time where it would have control of the bus; that is, writing data or generating an Ack/Nack in response to a read byte. At that time, the master generates a Stop condition, aborts the operation and sets the MBTTO event which causes an MA_TMO interrupt status to get updated in the I2C Interrupt Status Register. An optional interrupt can also be sent to the Interrupt Controller if the MA_TMO bit is enabled in the I2C Interrupt Enable Register. For a slave transaction, the slave waits for the start of the next bit time, releases the I2C_SDA and I2C_SCL signals and sets the SBTTO event, which causes an SA_FAIL interrupt status to get updated in the I2C Interrupt Status Register. An optional interrupt can be sent to the Interrupt Controller if the SA_FAIL bit is enabled in the I2C Interrupt Enable Register. The slave then reverts to looking for the next Start/Restart/Stop.
- Transaction timeout (see I2C Byte/Transaction Timeout Register) – This timeout is disabled on reset. It detects a situation where a master is keeping the bus for an extended period of time, as measured from the Start to Stop condition. This timeout is available primarily to detect a violation of the SMBus TLOW:SEXT time. The response to this timeout expiring is identical to the byte timeout, with the exception that the events are MTRTO or STRTO for the master or slave respectively.
- Boot timeout (see I2C Boot and Diagnostic Timer) – This timeout detects a situation where the boot load sequence does not complete in a reasonable time. This could occur if the EEPROM was improperly programmed with an infinite chaining loop, the bus ownership is held by some other device, or some other anomalous situation resulting in any of the timeouts. If the boot timeout expires before the normal end of the boot load sequence, the master interface will read until the next data byte and drive a Stop condition on the bus. It then sets the BLTO event, which causes a BL_FAIL interrupt status to get updated in the I2C Interrupt Status Register . An optional interrupt can also be sent to the Interrupt Controller if the BL_FAIL bit is enabled in the I2C Interrupt Enable Register. If the boot timeout is not desired, then the EEPROM programming should immediately write the I2C Boot and Diagnostic Timer.COUNT to 0 to disable the timeout.

Figure 56 shows the relationship of the I2C timeouts to I2C operations.

Figure 56: I2C Timeout Periods



This document is confidential and is subject to an NDA.

11.13 Bus Timing

Figure 57 shows the relationship of the bus timing parameters to the generation of the I2C_SCL and I2C_SDA signals on the I2C bus. These parameters are configured in the following registers:

- I2C Start Condition Setup/Hold Timing Register
- I2C_SDA Setup and Hold Timing Register
- I2C Stop/Idle Timing Register
- I2C_SCL High and Low Timing Register
- I2C_SCL Minimum High and Low Timing Register

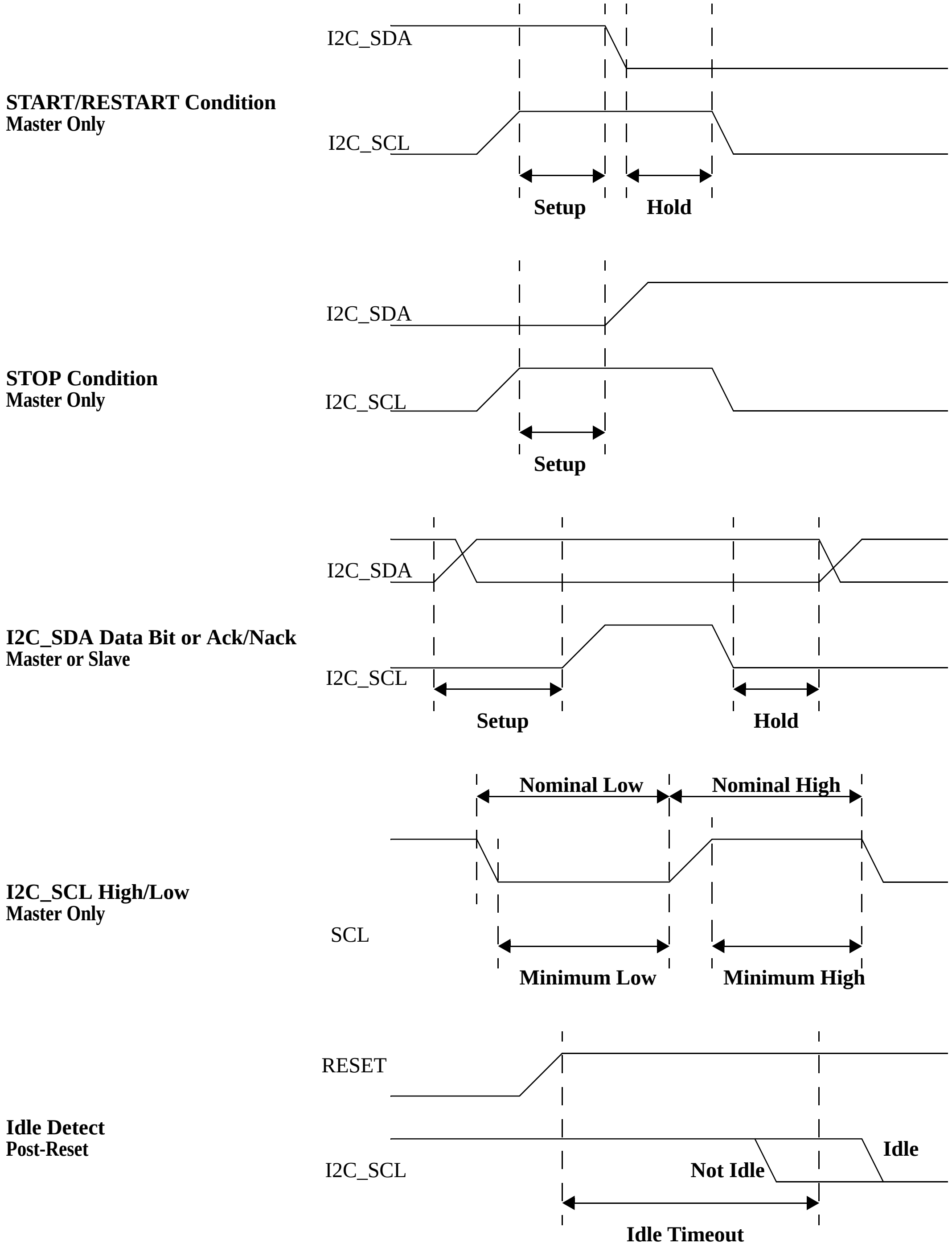
The bus timing resets to 100-kHz operation. By reprogramming the listed registers, other bus speeds can be configured. Speeds above 100 kHz are not guaranteed to conform to the *I^{2}C Specification* because of the absence of Schmitt triggers on the input of the I2C_SDA and I2C_SCL signals, and the absence of slope controlled outputs for the I2C_SDA and I2C_SCL signals. It is up to the board or system designer to decide on the applicability of operation at speeds above 100 kHz.

Bus timing does not normally change during a transaction, even if the listed registers are changed. The timing registers are sampled at certain times to prevent this from occurring. The following are the times when timing adjustments will take effect:

- On hard reset (times are reset to 100 kHz)
- On soft reset
- At the start of a master transaction through the I2C Master Control Register, when the START condition is generated
- When a chain operation occurs during boot load

Timing parameters are discussed further in the following sections.

Figure 57: I2C Bus Timing Diagrams



11.13.1 Start/Restart Condition Setup and Hold

The Start/Restart Condition is generated by a master. As displayed in Figure 57, the Start Setup time defines the minimum period both the I2C_SDA and I2C_SCL signals must be seen high (1) before the I2C_SDA signal is pulled low (0) to trigger the Start. The I2C_SDA signal must also have fulfilled the I2C_SDA Setup time before the rising edge of I2C_SCL. Once the I2C_SDA signal is seen low (0), the Start Hold time is the minimum period the I2C_SCL signal must continue to remain high (1) before it is pulled low (0). These parameters are used by the Tsi721 as a master when generating the Start condition. The times may be violated by an external master or slave pulling the I2C_SDA or I2C_SCL signals low before the setup/hold periods are expired, which could cause an arbitration loss or collision.

11.13.2 Stop Condition Setup

The Stop Condition is generated by a master. As displayed in Figure 57, the Stop Setup time defines the minimum period the I2C_SDA must be seen low (0) and the I2C_SCL signal must be seen high (1) before the I2C_SDA signal is released high (1) to trigger the Start. The I2C_SDA signal must also have fulfilled the I2C_SDA Setup time before the rising edge of I2C_SCL. There is no separate Stop Hold parameter, as the only valid condition following a Stop would be a Start; therefore, the Start Setup fulfills the same use as a Stop Hold or Stop-to-Start buffer time. This parameter is used by the Tsi721 as a master when generating the Stop condition. If the I2C_SCL signal was prematurely pulled low (0) by an external master or slave, this would be seen as a collision event.

11.13.3 I2C_SDA Setup and Hold

Either a master or a slave can be in control of the I2C_SDA signal, depending on the phase of the data transfer protocol. As displayed in Figure 57, the I2C_SDA Setup time defines the minimum period the I2C_SDA signal must set to the desired state while I2C_SCL is low (0) before the I2C_SCL signal is release high (1) to generate the high period of the clock. The I2C_SDA Hold time defines the minimum period the I2C_SDA signal will be left unchanged after the falling edge of I2C_SCL (I2C_SCL seen low). The I2C_SDA hold time can be violated by another device pulling I2C_SDA low, but this is not an error, as it normally indicates another device with a different design.

The I2C_SDA setup time is not as defined in the *I^{2}C Specification*. The setup time parameter encompasses both the maximum rise/fall time of the I2C_SDA signal plus the output hold time and must be set accordingly. There is no feedback check that the I2C_SDA signal goes to the desired state, as this could cause I2C_SCL to be held low erroneously. If another device is also controlling the I2C_SDA signal, it will likely cause an arbitration loss or collision.

11.13.4 I2C_SCL Nominal and Minimum Periods

These parameters are used by the Tsi721 as a master to generate the I2C_SCL clock. The master must obey the minimum times to conform to the *I^{2}C Specification*, and must also attempt to regulate the overall I2C_SCL frequency to a defined period. From Figure 57, it can be seen that the logic measures the minimum periods high/low from the detected rising/falling edges of the I2C_SCL signal to the point where I2C_SCL is driven low or released high to generate the opposing edge. In conjunction, a separate nominal period timer measures from driven low to released high, and released high to driven low. Both timers must expire if unaffected by external devices. If another device pulls the I2C_SCL signal low prematurely in the high period, the high period timers are expired and the lower period timers restart for the low period, so the low period can be stretched by the nominal timer. If another device holds the I2C_SCL signal low longer in the low period than the nominal low period, the high period nominal timer will likely expire early and the minimum high period timer will control the high period when the clock is finally released.

11.13.5 Idle Detect Period

This is a master-only parameter that is used in two cases. First, upon exit from reset it is unknown if another master is active. The Idle Detect timeout sees if the I2C_SCL signal remains high long enough (roughly 50 microseconds) that it is unlikely another master is active. If I2C_SCL is seen low during this period, it is assumed another master is active, and the master enters the Wait for Bus Idle phase. If the idle detect period expires without I2C_SCL seen low, then it is assumed the bus is idle and the master is free to generate a Start Condition if needed.

Second, during the Wait for Bus Idle phase, an external master that claimed the bus could stop activity without issuing a STOP condition. When a master operation is started but the bus is currently seen busy, the idle detect timer monitors the I2C_SCL and I2C_SDA signals. If the I2C_SCL and I2C_SDA signals both remain high longer than the idle detect period, the bus is then assumed idle even though a STOP had not been seen, and the master logic will attempt the requested transaction.

12. GPIO Interface

Topics discussed include the following:

- Overview
- GPIO as Inputs/Outputs

12.1 Overview

The Tsi721 contains a 16-bit, parallel GPIO Interface that supports a variety of functions, such as LED state generation or monitoring of input status. It can be configured through device registers as a standard I/O port.

The GPIO Interface includes the following features:

- 16 general purpose I/Os
- Each I/O individually programmable as input or open-drain output

12.2 GPIO as Inputs/Outputs

When the GPIO signals are used as input/output signals, they are controlled using the following registers:

- GPIO 0 Data Register controls the data value driven by GPIO signals configured as outputs. It also gives the current value of the GPIO signal
- GPIO 0 Control Register controls whether the GPIO pin is an input or an output pin.

13. JTAG Interface

Topics discussed include the following:

- Overview
- JTAG Device Identification Number
- JTAG Register Access

13.1 Overview

The JTAG Interface is compliant with IEEE 1149.6 B*oundary Scan Testing of Advanced Digital Networks,* as well as IEEE 1149.1 *Standard Test Access Port and Boundary Scan Architecture* standards. The interface has five standard signals – TMS, TCK, TDI, TDO, and TRSTn – that allow full control of the internal TAP (Test Access Port) Controller. The JTAG Interface has the following features:

- Contains a 5-pin Test Access Port (TAP) Controller, with support for the following registers:
  - Instruction register (IR)
  - Boundary scan register
  - Bypass register
  - Device ID register
  - User test data register (DR)
- Supports debug access of Tsi721's configuration registers
- Supports the following instruction opcodes:
  - Sample/Preload
  - Extest
  - EXTEST_PULSE *(1149.6)*
  - EXTEST_TRAIN *(1149.6)*
  - Bypass
  - IDCODE
  - Clamp
  - User data select

13.2 JTAG Device Identification Number

For information, see JTAG ID Register.

13.3 JTAG Register Access

Users can read and write Tsi721's registers through the JTAG Interface in order to debug issues that can affect register accesses. Register access through the interface can also be used in normal mode to do extensive read and write accesses on performance registers without affecting normal traffic in the device or during initialization.

Before using the IEEE Register Access Command feature, the Tsi721 must be reset by driving TRSTn low.

A user-defined command enables the read and write capabilities of the interface. The command is in the IEEE 1149.1 Instruction Register (IR) in the Tsi721.

- IEEE Register Access Command (IRAC)

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

13.3.1 Format

The format used to access Tsi721's registers is displayed in Figures 58 and 59. The address displayed in the figure is the JTAG register address.

Figure 58: Register Access From JTAG – Serial Data In

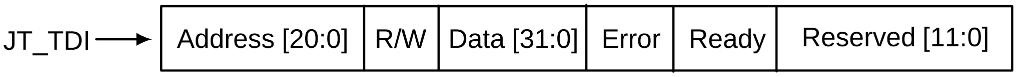
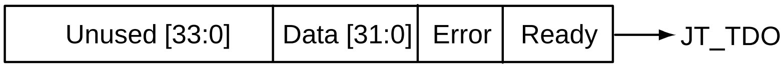


Figure 59: Register Access From JTAG – Serial Data Out



13.3.2 Write Access to Registers from the JTAG Interface

The following steps are required to write to a register through the JTAG Interface:

1. Move to the Tap Controller "Shift-IR" state and program the instruction register with IRAC instruction. This is completed by shifting in IR length instruction register, which is 62 bits of all ones except for the second last bit (for example, ...1111_1101).
2. Move to the "Shift-DR" state and shift the data[31:0], R/W = 1 and the address[20:0] serially in the TDI pin. To prevent corruption of unused bits, the full DR bits must be written with the following values:
   - DR[67:47] = ADDR[20:0]
   - DR[46] = R/W
   - DR[45:14] = DATA[31:0]
   - DR[13:12] = 0b0
   - DR[11:0] = 0b0
3. Move to the "Run-test idle" state and loop in this state for a minimum of 20 TCK cycles.
4. Move to the "Shift-DR" state again and shift-in 68 zero bits to DR[67:0], while at the same time verify the Ready and Error bits that are being shifted-out as the first two bits.
5. To to perform another write, go back to step 2.

13.3.3 Read Access to Registers from the JTAG Interface

The following steps are required to read a register through the JTAG Interface:

1. Move to the Tap Controller “Shift-IR” state and program the instruction register with IRAC instruction.

   This step is optional if the instruction register is already programmed during the write cycle.
2. Move to the “Shift-DR” state and shift the R/W = 0 and the address[20:0] serially in the TDI pin. To prevent corruption of unused bits, the full DR bits must be written with the following values:
   - DR[67:47] = ADDR[20:0]
   - DR[46] = R/W
   - DR[45:14] = DATA[31:0]
   - DR[13:12] = 0b0
   - DR[11:0] = 0b0
3. Move to the “Run-test idle” state and loop in this state for a minimum of 20 TCK cycles.
4. Move to the “Shift-DR” state and shift in 68 bits of 0. The first two bits of data shifted out are the Error and Ready bits. The next 32 bits are data. The remainder of the shifted out data, the Unused bits in Figure 59, can be discarded.
5. Shift the Error and Ready bits out at the same time.
6. Verify that the Error bit is at logic low and the Ready bit is at logic high.
7. To perform another read, go back to step 2.

Clocking and Resets

14. Clocking

Topics discussed include the following:

- Overview

14.1 Overview

The Tsi721 supports two types of PCIe clocking modes:

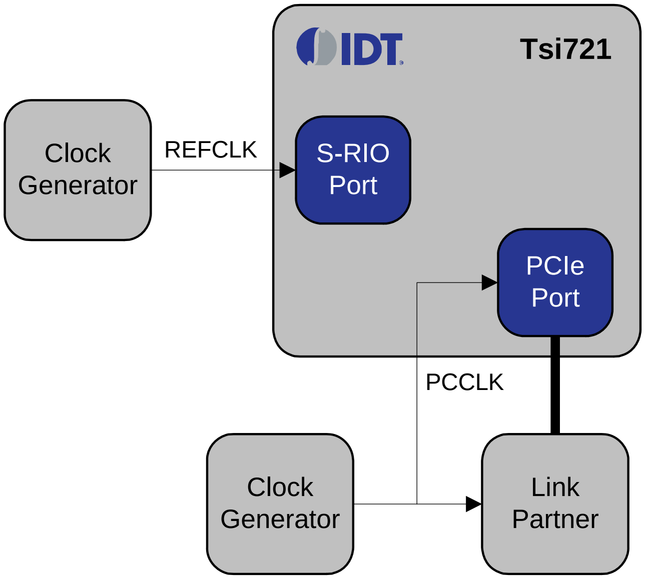
- PCIe common clock mode
- PCIe non-common clock mode

14.2 PCIe Common Clock Mode

In PCIe common clock mode, Tsi721's REFCLK and PCCLK clock signals are connected to different clock generators (see Figure 60). The PCCLK signal must have a PCIe compliant 100-MHz clock, while the REFCLK's clock source frequency is defined by CLKSEL[1:0].

The PCCLK signal can share the same clock generator as Tsi721's PCIe link partner, or use a different 100-MHz clock source. When a spread-spectrum clock is used by Tsi721's PCIe link partner, PCIe common clock mode must be used and PCCLK must have the same source as the link partner.

Figure 60: PCIe Common Clock Mode

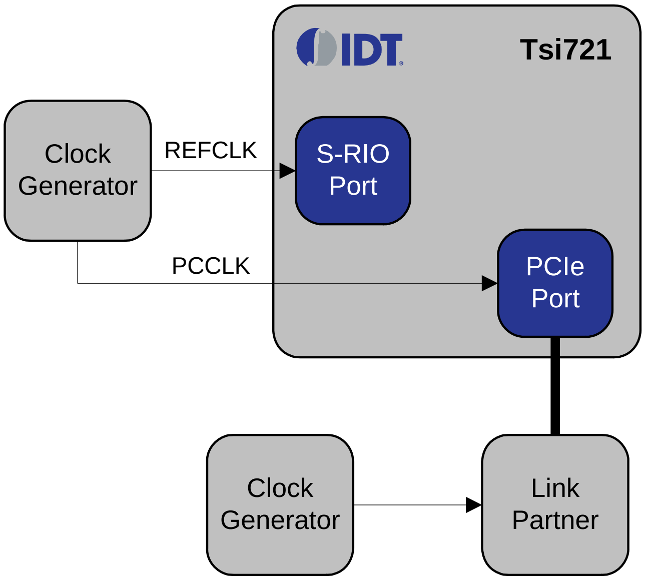


14.3 PCIe Non-Common Clock Mode

In PCIe non-common clock mode, Tsi721's REFCLK and PCCLK clock signals must have the same frequency as indicated by the CLKSEL[1:0] package pins. REFCLK and PCCLK can use either the same clock generator or different clock generators (within their own tolerances). Tsi721's PCIe link partner, however, has its own clock generator (see Figure 61).

While in non-common clock mode, a spread-spectrum clock cannot be used for the PCIe link.

Figure 61: PCIe Non-Common Clock Mode



For Tsi721 clock pin information, see “Signals” in the *Tsi721 Datasheet*. For Tsi721 clock electrical specifications, see “Reference Clock” in the *Tsi721 Datasheet.*

15. Reset, Power-up, and Initialization

Topics discussed include the following:

- Overview
- Fundamental Reset
- PCIe Hot Reset
- S-RIO Reset
- Procedure for Resets Other Than Fundamental Reset

15.1 Overview

The Tsi721 supports four types of resets:

- Fundamental reset
- PCIe hot reset
- S-RIO reset
- JTAG reset

A JTAG reset is controlled by the JTAG_TRSTn signal. The other three resets do not cause a JTAG reset.

Note that power up package pins are sampled only on rising edge of a fundamental reset.

When multiple resets are initiated concurrently, the Tsi721 uses the reset priority in Table 76. When a high priority and low priority reset are initiated concurrently, and the condition that caused the high priority reset ends before causing the low priority reset, the device transitions immediately to the reset associated with the low priority reset condition.

Table 76: Tsi721 Reset Priority

Priority	Reset Type	Reset Cause
1 (Highest)	Fundamental reset	See Fundamental Reset
2	PCIe hot reset	Reception of TS1 ordered sets on upstream port indicating a hot reset, or Data link layer of the upstream port indicates a DL_Down status (see SR2PC General Interrupt CSR.DL_DOWN)
3 (Lowest)	S-RIO reset	Reception of four consecutive S-RIO reset control symbols

15.2 Fundamental Reset

A fundamental reset (also called a device reset) can be cold or warm. A cold fundamental reset occurs after the Tsi721 is powered-on and the RSTn signal is asserted. A warm fundamental reset occurs when a reset is initiated while power remains applied to the device. The Tsi721 functions in the same way regardless of whether the fundamental reset is cold or warm.

For information about the Tsi721's power sequencing requirements, see the *Tsi721 Datasheet*.

A fundamental reset can be initiated by any of the following conditions:

- A cold fundamental reset initiated by application of power (that is, a power-on) followed by assertion of the RSTn signal.
- A warm fundamental reset initiated by assertion of RSTn while power remains applied.
- A warm fundamental reset initiated by writing a one to Device Control Register.FRST.
- A warm fundamental reset initiated when receiving four S-RIO link-request/reset-device control symbols in a sequence (without any intervening packets or control symbols, except status control symbols) from its link partner and the receiving port's RapidIO PLM Port Implementation Specific Control Register.SELF_RST is 1.

15.2.1 Fundamental Reset With EEPROM Bootload

An EEPROM bootload is controlled by the I2C_DISABLE (= 0) power-up signal and SR_BOOT (=0) signal.

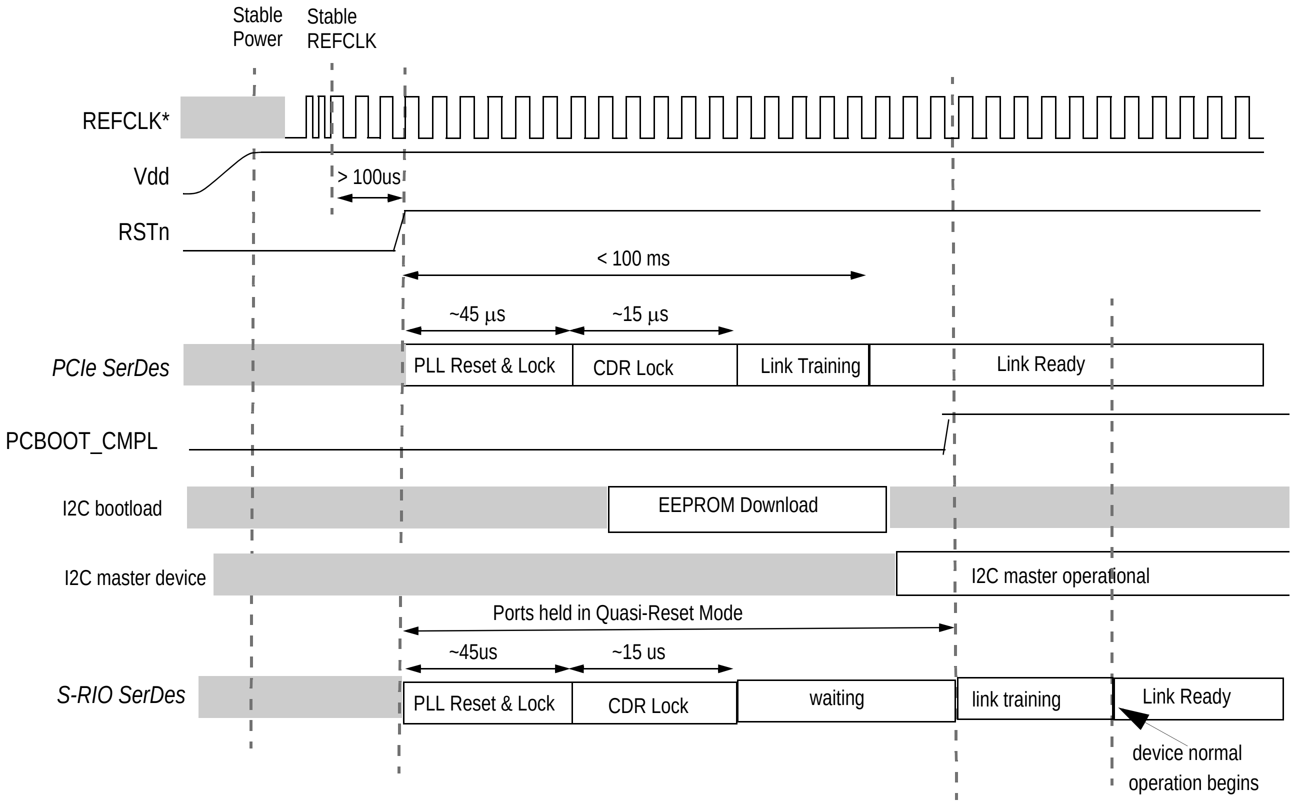
When a fundamental reset is initiated, the following sequence is executed:

1. Tsi721 waits for the fundamental reset condition to clear (for example, negation of RSTn or Device Control Register.FRST).
   — All registers are initialized to their default value.
   — PCBOOT_CMPL and SRBOOT_CMPL bits in the Device Control Register are 0.
   — REGUNLOCK bit is set to 0 in the Endpoint Control Register.
2. The on-chip PLL and SerDes are initialized (for example, PLL lock).
3. The I2C Interface is taken out of reset and initialized.
4. Within 20 ms after the fundamental reset condition clears, PCIe link training begins. While link training occurs, execution of the reset sequence continues.
5. Within 100 ms after clearing the fundamental reset condition, the PCIe link partners complete link training.
6. The contents of the EEPROM are read and appropriate registers are updated.
   — REGUNLOCK bit must be set to 1 in the Endpoint Control Register before register fields with type RES can be modified.
   — At a minimum, RE/RES bits inside PCIe MAC and S-RIO MAC should be set to correct values
   — While the contents of the EEPROM are read, the Tsi721 enters the quasi-reset state. In the quasi-reset state, the Tsi721 responds to all Type 0 configuration request TLPs with a configuration-request-retry-status completion. All other TLPs are ignored (that is, flow control credits are returned but the TLP is discarded).
   — If an error is detected during loading of the EEPROM, then the loading process is aborted. Error information is recorded in the I2C Event and Event Snapshot Registers (for example, BLERR).
   — When EEPROM initialization completes, the BLOK bit in the I2C Event and Event Snapshot Registers is set to 1.
7. As part of the EEPROM bootload, the REGUNLOCK bit in the Endpoint Control Register should be cleared (after all RES registers of PCIe MAC are loaded), and SRBOOT_CMPL bit in the Device Control Register should be set to 1.
8. Once EEPROM initialization is completed, either successfully or failed with error, the Tsi721 automatically sets the PCBOOT_CMPL bit in the Device Control Register, then the Tsi721 exits the quasi-reset state.
   — Tsi721 remains in the quasi-reset state until the PCBOOT_CMPL bit is set.

- If an error occurs during the EEPROM bootload, the PCIe root complex can still configure the Tsi721 using the sequence described in Fundamental Reset With Root Complex Bootload. The method for informing EEPROM errors to the root complex can be either timeout or by I2C master polling.
- Steps 9–14 assume that there are no EEPROM errors.

9. The S-RIO link starts training
10. The PCIe root complex can enumerate the Tsi721, and read and write Tsi721 registers but not send or receive data through the Tsi721.
11. When S-RIO link training completes successfully, the PORT_OK bit in the RapidIO Port Error and Status CSR is set to 1.
12. The S-RIO host can start device discovery process.
13. Additional Tsi721 configuration (registers except those with RE/RES type) can be performed by either S-RIO host or PCIe root complex.
14. Normal device operation starts (data can flow through the Tsi721) after the PCIe root complex sets the BME bit to 1 in the PCI Control and Status Register, and the S-RIO host sets the MAST_EN bit to 1 in the RapidIO Port General Control CSR.
    - The *PCI Express Specification (Rev. 2.1)* indicates that a device must respond to configuration request transactions within 100 ms from the end of a reset (cold, warm, or hot). Additionally, the specification indicates that a device must respond to configuration requests with a successful completion within 1.0 seconds after a reset of a device. The reset sequence above guarantees that the Tsi721 can respond successfully to configuration requests within the 1-second period as long as the EEPROM initialization completes within 200 ms. (Note: Under normal circumstances, 200 ms is more than adequate to initialize registers in the device with a master I2C operating frequency of 100 kHz.)
    - EEPROM initialization can cause writes to register fields that initiate side effects such as link retraining. These side effects are initiated at the point at which the write occurs. Therefore, EEPROM initialization should be structured to ensure proper configuration before initiation of these side effects.
    - The operation of a fundamental reset with EEPROM initialization is displayed in Figure 62.

Figure 62: Tsi721 Fundamental Reset Sequence with EEPROM Bootload



15.2.2 Fundamental Reset With Root Complex Bootload

A PCIe root complex bootload is controlled by the I2C_DISABLE (= 1) and I2C_MA (=0) power-up signals and the SR_BOOT (= 0) signal.

When a fundamental reset is initiated, the following sequence is executed:

1. Tsi721 waits for the fundamental reset condition to clear (for example, negation of RSTn).
   — All registers are initialized to their default value.
   — PCBOOT_CMPL and SRBOOT_CMPL bits in the Device Control Register are 0.
   — REGUNLOCK bit is set to 0 in the Endpoint Control Register.
2. The on-chip PLL and SerDes are initialized (for example, PLL lock).
3. The I2C interface is taken out of reset and initialized.
4. Within 20 ms after the fundamental reset condition clears, PCIe link training begins. While link training occurs, execution of the reset sequence continues.
5. Within 100 ms after clearing the fundamental reset condition, the PCIe link partners complete link training.
6. The Tsi721 automatically sets PCBOOT_CMPL to 1, and takes itself out of the quasi-reset state.

7. The root complex enumerates Tsi721, and can read and write Tsi721 registers but not send or receive data through the device.
   - The root complex sets REGUNLOCK bit in the Endpoint Control Register to allow PCIe MAC register fields with type RE/RES to be modified.
   - The root complex starts to configure all RE/RES registers inside PCIe MAC and S-RIO MAC.
8. The root complex clears the REGUNLOCK bit in the Endpoint Control Register and sets the SRBOOT_CMPL bit to 1 in the Device Control Register.
9. The S-RIO link starts training.
10. When S-RIO link training completes successfully, the PORT_OK bit in the RapidIO Port Error and Status CSR is set to 1.
11. The S-RIO host starts device discovery.
12. Additional Tsi721 configuration (registers except those with RE/RES type) can be performed by either the S-RIO host or the PCIe root complex.
13. Normal device operation starts (data can flow through the Tsi721) after the PCIe root complex sets the BME bit to 1 in the PCI Control and Status Register, and the S-RIO host sets the MAST_EN bit to 1 in the RapidIO Port General Control CSR.

15.2.3 Fundamental Reset With I2C Master Bootload

An I2C master bootload is controlled by the I2C_DISABLE (= 1) and I2C_MA (= 1) power-up signals, and the SR_BOOT (= 0) signal.

Here, an external device (for example, a board micro-controller) must configure all PCIe MAC RE registers before root complex can perform PCIe enumeration (for more information, see Overview).

When a fundamental reset is initiated, the following sequence is executed:

1. Tsi721 waits for the fundamental reset condition to clear (for example, negation of RSTn).
   - All registers are initialized to their default value.
   - PCBOOT_CMPL bit is set to 0 in the Device Control Register.
   - REGUNLOCK bit is set to 0 in the Endpoint Control Register.
2. The on-chip PLL and SerDes are initialized (for example, PLL lock).
3. The I2C interface is taken out of reset and initialized.
4. Within 20 ms after the fundamental reset condition clears, PCIe link training begins. While link training occurs, execution of the reset sequence continues.
5. Within 100 ms following clearing of the fundamental reset condition, the PCIe link partners complete link training.
6. The external I2C master polls the PCRDY bit in the Device Status Register until it is set to 1.
   - The external I2C master sets REGUNLOCK bit in the Endpoint Control Register to allow PCIe MAC register fields with type RE/RES to be modified.
   - The external I2C master starts to configure all RE/RES registers inside the PCIe MAC and S-RIO MAC.
7. The external I2C master clears the REGUNLOCK bit in the Endpoint Control Register of the PCIe MAC
8. The external I2C master sets the PCBOOT_CMPL and SRBOOT_CMPL bits in the Device Control Register, then the Tsi721 exits the quasi-reset state. The Tsi721 remains in the quasi-reset state until the PCBOOT_CMPL bit is set to 1.
9. The S-RIO link starts training.
10. The root complex enumerates the Tsi721, and can read and write Tsi721 registers but not send or receive data through the device.
11. When S-RIO link training completes successfully, the PORT_OK bit in the RapidIO Port Error and Status CSR is set to 1.

12. The S-RIO host starts device discovery.
13. Additional Tsi721 configuration (registers except those with RE/RES type) can be performed by either the S-RIO host or the PCIe root complex.
14. Normal device operation starts (data can flow through the Tsi721) after the PCIe root complex sets the BME bit to 1 in the PCI Control and Status Register, and the S-RIO host sets the MAST_EN bit to 1 in the RapidIO Port General Control CSR.

15.2.4 Fundamental Reset With S-RIO Bootload

An S-RIO bootload is controlled by the I2C_DISABLE (= 1) and I2C_MA (= 0) power-up signals, and the SR_BOOT (= 1) signal.

A remote S-RIO host can configure the Tsi721 before the PCIe root complex has enumerated Tsi721.

When a fundamental reset is initiated, the following sequence is executed:

1. Tsi721 waits for the fundamental reset condition to clear (for example, negation of RSTn).
   - All registers are initialized to their default value.
   - PCBOOT_CMPL bit is set to 0 in the Device Control Register.
   - REGUNLOCK bit is set to 0 in the Endpoint Control Register.
2. The on-chip PLL and SerDes are initialized (for example, PLL lock).
3. The I2C interface is taken out of reset and initialized.
4. The Tsi721 automatically sets the SRBOOT_CMPL bit in the Device Control Register, and all RE/RES registers within the S-RIO MAC become read only.
5. Within 20 ms after the fundamental reset condition clears, PCIe link training begins and S-RIO link training also begins. While link training occurs, execution of the reset sequence continues.
6. Within 100 ms following clearing of the fundamental reset condition, PCIe and S-RIO link partners complete link training.
7. The S-RIO host starts device discovery.
8. The S-RIO host polls the PCRDY bit in the Device Status Register until it is set to 1.
9. The S-RIO host sets REGUNLOCK bit in the Endpoint Control Register to allow PCIe MAC register fields with type RE/RES to be modified.
10. The S-RIO host starts to configure all RE/RES registers inside the PCIe MAC.
11. The S-RIO host clears the REGUNLOCK bit in the Endpoint Control Register of the PCIe MAC.
12. The S-RIO host sets the PCBOOT_CMPL bit in the Device Control Register, then the Tsi721 exits the quasi reset state. The Tsi721 remains in quasi-reset state until the PCBOOT_CMPL bit is set.
13. The root complex enumerates the Tsi721, and can read and write Tsi721 registers but not send or receive data through the device.
14. Additional Tsi721 configuration (registers except those with RE/RES type) can be performed by either the S-RIO host or the PCIe root complex.
15. Normal device operation starts (data can flow through the Tsi721) after the PCIe root complex sets the BME bit to 1 in the PCI Control and Status Register, and the S-RIO host sets the MAST_EN bit to 1 in the RapidIO Port General Control CSR.

15.3 PCIe Hot Reset

A PCIe hot reset is initiated by one of the following events:

- Reception of TS1 ordered-sets on the upstream port indicating a hot reset
- Data link layer of the upstream port indicates a DL_Down status

When a PCIe hot reset is initiated the following sequence of actions occurs:

1. The upstream port transitions its PHY LTSSM state to the appropriate state (that is, the Hot Reset state on reception of TS1 ordered-sets indicating hot reset or else the Detect state).
2. All Tsi721 logic except JTAG/I2C is logically reset to its initial state.
3. All register fields and registers except those designated Sticky, are reset to their initial value. The value of Sticky registers and fields is preserved across a hot reset.
4. As long as the condition that initiated the hot reset persists, Tsi721 remains at this step.
5. The Tsi721 starts PCIe and S-RIO link training and normal operation begins.

15.4 S-RIO Reset

15.4.1 S-RIO Protocol Reset from Link Partner

When four link-request/reset-device control symbols are received in a sequence (without any intervening packets or control symbols, except status control symbols) from its link partner, the Tsi721 initiates the S-RIO reset procedure as displayed below.

If the RapidIO PLM Port Implementation Specific Control Register.SELF_RST is 1, this triggers a fundamental reset. If SELF_RST is 0 and RapidIO PLM Port Implementation Specific Control Register.PORT_SELF_RST is 1:

1. The RST_REQ bit in the RapidIO Event Management Reset Request Port Status Register is set to 1.
2. All Tsi721 logic except JTAG/I2C is logically reset to its initial state
3. All register fields and registers except those designated Sticky, are reset to their initial value. The value of Sticky registers and fields is preserved across an S-RIO reset.

⚠ When Device Control Register.SR_RST_MODE is 0b0001, a reset request can be issued to the Tsi721 only when there are no packets being exchanged on the RapidIO link.

15.4.2 Registers Causing Tsi721 Reset Upon Update

When SR_RST_MODE of Device Control Register is set to all zeroes, changing any one of the following Tsi721 registers causes a Tsi721 reset with behavior identical to that of a PCIe hot reset:

-
- Modifying GB_1p25_EN, GB_2p5_EN, GB_3p125_EN, GB_5p0_EN, or GB_6p25_EN in the RapidIO Port Control 2 CSR

15.5 Procedure for Resets Other Than Fundamental Reset

⚠ The restrictions and procedures in this section are recommended to ensure that the Tsi721 is properly reset.

The following two procedures are recommended for an S-RIO port or PCIe port reset by a method other than a fundamental reset.

15.5.1 Reset Avoiding Timeout

This reset procedure avoids far-end S-RIO request / PCIe request timeouts throughout the system:

1. Stop inbound packet traffic to the Tsi721.
   a. Stop sending S-RIO packets from all devices to Tsi721 other than as required for subsequent steps in this procedure.

b. Instruct each device (or waits for longest delay among all devices for packets to arrive at the Tsi721, then selects one device) on the Tsi721 S-RIO side to issue an S-RIO NREAD (Tsi721 translates it to PCIe MRd) with prio/crf of 0/0 to another device (endpoint or root complex) that sits on the Tsi721 PCIe side, and waits for the response to return

c. Instruct each device (or wait for longest delay among all devices for packets to arrive at Tsi721, then select one device) on the S-RIO side of Tsi721 to issue one S-RIO maintenance read with prio/crf of 0/0 to read an SR2PC register (see SR2PC Registers), and wait for their responses to return

d. For each active SMSG inbound messaging DMA channel, instruct each device on Tsi721 S-RIO side to issue an S-RIO message (Tsi721 translates it to PCIe MWr) with prio/crf of 0/0 to another device (endpoint or root complex) that sits on the PCIe side of Tsi721, and wait for the response to return; this ensures that inbound buffers within Tsi721 are cleared

The above steps ensure that inbound buffers within Tsi721 are cleared.

2. Stop outbound packet traffic to the Tsi721.

a. Stop sending PCIe TLPs from any device to Tsi721 other than as required for subsequent steps in this procedure.

b. Instruct each device (or wait for longest delay among all devices for TLPs to arrive at Tsi721, then select one device) on the PCIe side of Tsi721 to issue a PCIe MRd (Tsi721 translates it to S-RIO NREAD) to another device that sits on the S-RIO side of Tsi721, and wait for the completion to return

c. Instruct each device (or wait for longest delay among all devices for TLPs to arrive at Tsi721, then select one device) on the PCIe side of Tsi721 to issue a PCIe MRd to read an Tsi721 register, and wait for the completion to return

d. For each BDMA DMA channel and each SMSG outbound DMA channel, invoke DMA channel suspending procedure

The above steps ensure that outbound buffers within Tsi721 are cleared.

3. Wait for the following Tsi721 indications.

a. Poll RapidIO PBM Port Status Register.IG_EMPTY and RapidIO PBM Port Status Register.EG_EMPTY bits until they are both set.

b. Poll TP bit of PCIe Device Control and Status Register until it is set.

Once the above procedure is completed, the Tsi721 can be safely reset.

15.5.2 Reset With Potential Timeout

If software directly issues a reset to the Tsi721, the device is reset immediately and PCIe/S-RIO requests buffered within the Tsi721 are lost. This scenario can cause devices with outstanding read requests to timeout.

15.5.2.1 Root Complex Reset Procedure While PCIe Link Is Alive

1. The root complex resets the Tsi721's S-RIO link partner by writing 0b011 to the CMD field of the RapidIO Port Link Maintenance Request CSR.
2. The root complex issues a PCIe hot reset to the Tsi721.

15.5.2.2 Root Complex Reset Procedure With Unexpected PCIe Link Down

When the root complex cannot access the Tsi721 due to an unexpected PCIe link down or an uncontrollable hot reset, the Tsi721 will be reset with its S-RIO port entering the Fatal Error Stop State. The following procedure should then be used:

1. The root complex resets the Tsi721's S-RIO link partner by writing 0b011 to the CMD field of RapidIO Port Link Maintenance Request CSR.
2. The root complex polls the PORT_OK bit in the RapidIO Port Error and Status CSR until it becomes high.
3. The root complex writes 0x8000_0000 to the RapidIO Port Local ackID Status CSR

15.6 Changing S-RIO Port Width or Link Rate

To change the S-RIO port width or link rate, the user must modify OVER_PWIDTH in RapidIO Port Control CSR, or GB_1p25_EN, GB_2p5_EN, GB_3p125_EN, GB_5p0_EN, or GB_6p25_EN in the RapidIO Port Control 2 CSR. In addition, the following requirements must be met:

- These fields usually should be modified after bootload procedure but before sending data traffic through the Tsi721:
  - — Or a procedure similar to Reset Avoiding Timeout can be used before modifying these fields
  - — The true requirement is that there must be no other pending transactions in any Tsi721 internal interfaces
- Set SR_RST_MODE of Device Control Register to 0b0001. This causes the S-RIO MAC logic core to reset and the S-RIO link to retrain.
- Wait for 1 ms and starts to continue configuring Tsi721 and sends traffic through Tsi721

Device Registers

16. Registers Overview

Topics discussed include the following:

- Overview
- Register Field Type

16.1 Overview

The internal registers of the Tsi721 can be accessed using the following methods:

- BAR0 access from PCIe for all non-configuration space registers
- Type 0 CFG access from PCIe all configuration space registers
- S-RIO maintenance read/write access from S-RIO for all registers
- I2C access for all registers
- JTAG access for all register

Tsi721's registers are assigned register offsets in the block-based register chapters. Actual register addresses must be calculated according to the following table.

Table 77: Register Address Calculation from Register Offset in Register Definition

Registers	PCIe Address	S-RIO Address	I2C Address	JTAG Address
Registers within PCIe MAC	config0 address = register offset	S-RIO maintenance read/ write address = 0x70000 + register offset	I2C address = 0x70000 + register offset	JTAG address = 0x70000 + register offset
All other registers	PCIe MRd/MWr address = BAR0 base address + register offset	S-RIO maintenance read/ write address = register offset	I2C address = register offset	JTAG address = register offset

16.2 Register Field Type

Table 78: Register Field Types

Register Type	Description	PCIe Hot Reset and S-RIO Reset	Fundamental Reset
R	Read only	Reset to default values	Reset to default values
RS	Read, sticky	No effect. Keep current values.	Reset to default values
RCW	Read and clear after read, and write	Reset to default values	Reset to default values
RCWS	Read and clear after read, write, sticky	No effect. Keep current values	Reset to default values
RE	Read only while Tsi721 is operational, but writeable	Reset to default values	Reset to default values
RES	Registers that must be configured during Tsi721 bootload procedure, and become read only after bootload has completed. Note: For RES registers in PCIe MAC, their values can be modified when REGUNLOCK in Endpoint Control Register is set to 1.	No effect. Keep current values	Reset to default values
R/W	Read and write	Reset to default values	Reset to default values
R/WS	Read and write, sticky	No effect. Keep current values.	Reset to default values
R/W0C	Read, write 0 to clear	Reset to default values	Reset to default values
R/W0CS	Read, write 0 to clear, sticky	No effect. Keep current values.	Reset to default values
R/W1C	Read, write 1 to clear	Reset to default values	Reset to default values
R/W1CS	Read, write 1 to clear, sticky	No effect. Keep current values.	Reset to default values
R/W1S	Read, write 1 to set (when software writes it to 1; hardware automatically clear it to 0 after setting associated interrupt bit or when software writes to a specific register)	Reset to default values	Reset to default values
RC	Read and clear after read	Reset to default values	Reset to default values
RCS	Read and clear, sticky	No effect. Keep current values	Reset to default values
Undefined	Register reset value depends on power-up signals	-	Reset to default values

17. PCIe Registers

Topics discussed include the following:

17.1 Overview

All PCIe software visible registers are contained in a 4-KB address space that corresponds to the PCIe configuration space. PCIe registers can be read and written by the PCIe root complex using PCIe configuration transactions or by the application layer through the configuration space register interface.

- Registers should be accessed with byte-enables that correspond to one dword.
- When a register is accessed by a PCIe configuration transaction, details of the operation are available on status information interface signals. The application layer can use this interface to snoop read and write operations.
- The extended configuration space access mechanism allows PCIe extended configuration space registers (that is, those starting at address 0x100 in the 4-KB configuration space) to be accessed by systems that only support access to PCI configuration space. This access mechanism is supported only through PCIe configuration requests and is not supported through the PCI configuration space register interface.
    - Accessing the Extended Configuration Space Access Data Register through the PCI configuration space register interface produces undefined results.

Register field types:

- Fields denoted as RES are defined as read-only in the *PCI Express Specification (Rev. 2.1)* but can be initialized in the endpoint.
- RES fields are normally read only. They can be modified when the REGUNLOCK bit is set in the Endpoint Control Register (that is, the register is unlocked).
- Initialization can be performed by the root using PCIe configuration transactions or by the application layer through the configuration space register interface.
- Register fields denoted as “sticky” by the *PCI Express Specification (Rev. 2.1)*, which are indicated by “S” in the Type definition, are not modified or initialized as a result of a hot reset.

17.2 Configuration Register Side Effects

There are software visible configuration registers that have a side effect when written because they can affect the PCIe Interface’s ability to respond with a completion. A configuration write to such a register returns a completion to the link partner before the side-effect action is performed.

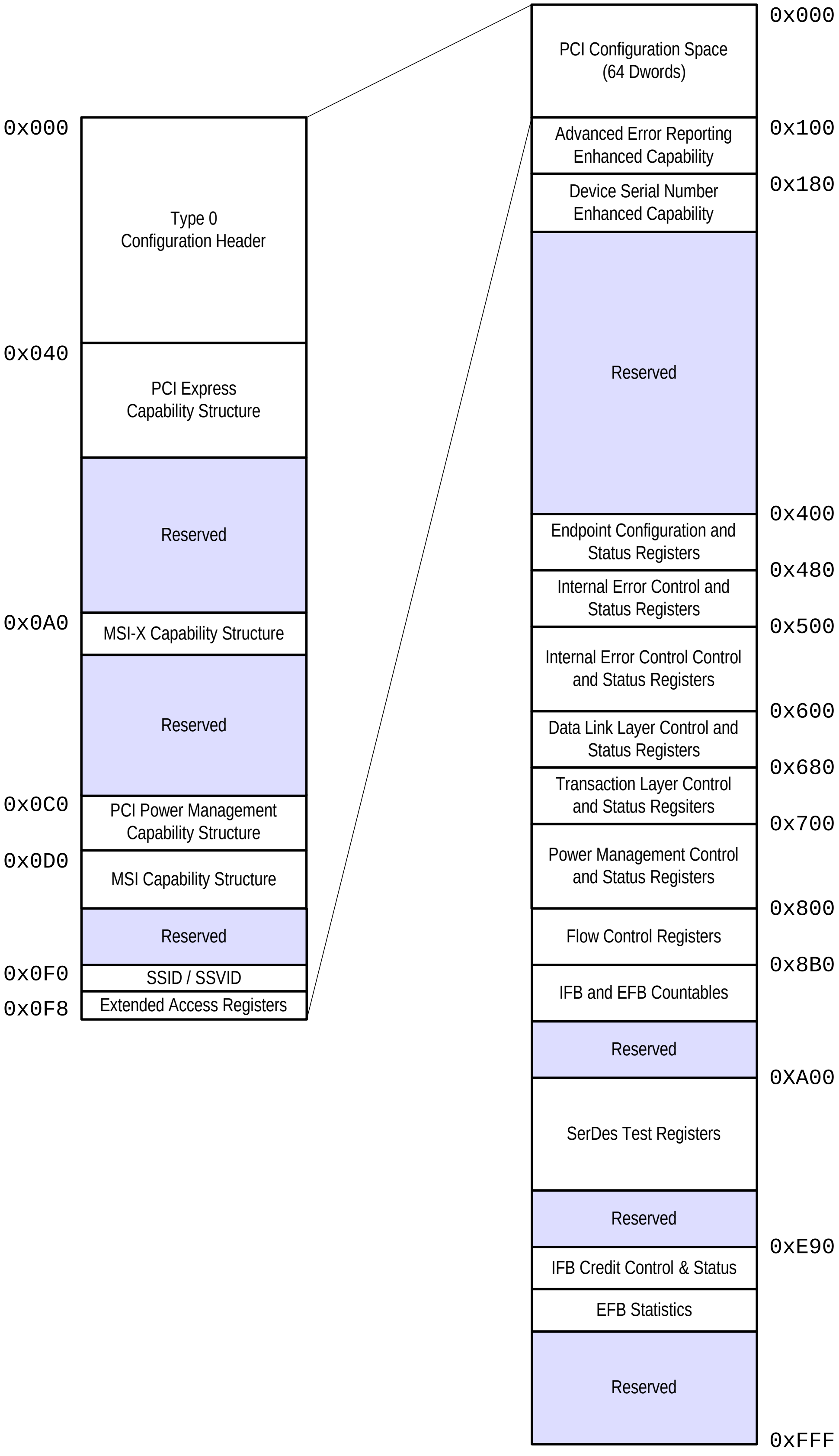
- This is implemented by delaying the side-effect action by the delay specified in the SEDELAY field in the Side Effect Delay Register following generation of the completion. If the completion is not accepted by the link partner in this time interval, then the completion is lost.
- The default value of the SEDELAY field is 1 ms.

The following register, when written[1], has a side-effect action delay.

- PHY Link State 0 Register.FLRET

[1] The SEDELAY is applied by the hardware when the registers listed are written using PCIe configuration requests, as well as through the S-RIO or I2C interfaces.

Figure 63: PCIe Register Organization



17.3 Register Map

Table 79: PCIe Register Map

Offset	Register Mnemonic	See
PCI Type 0 Configuration Header Registers		
0x000	PCI_ID	PCI Identification Register
0x004	PCI_CSR	PCI Control and Status Register
0x008	PCI_CLASS	PCI Class Register
0x00C	PCI_MISC0	PCI Miscellaneous 0 Register
0x010	PCI_BAR0	PCI BAR 0
0x014	PCI_BAR1	PCI BAR 1
0x018	PCI_BAR2	PCI BAR 2
0x01C	PCI_BAR3	PCI BAR 3
0x020	PCI_BAR4	PCI BAR 4
0x024	PCI_BAR5	PCI BAR 5
0x028	PCI_CCISPTR	PCI CardBus CIS Pointer Register
0x02C	PCI_SUBVID	PCI Subsystem ID Register
0x030	PCI_EROMBASE	PCI Expansion ROM Base Register
0x034	PCI_CAPPTR	PCI Capability Pointer Register
0x03C	PCI_MISC1	PCI Miscellaneous 1 Register
PCIe Capability Structure Registers		
0x040	PCIECAP	PCIe Capability Register
0x044	PCIEDCAP	PCIe Device Capabilities Register
0x048	PCIEDCTL	PCIe Device Control and Status Register
0x04C	PCIELCAP	PCIe Link Capabilities Register
0x050	PCIELCTL	PCIe Link Control Register
0x064	PCIEDCAP2	PCIe Device Capabilities 2 Register
0x068	PCIEDCTL2	PCIe Device Control and Status 2 Register
0x06C	PCIELCAP2	PCIe Link Capabilities 2 Register
0x070	PCIELCTL2	PCIe Link Control 2 Register

Table 79: PCIe Register Map *(Continued)*

Offset	Register Mnemonic	See
MSI-X Capability Structure Registers		
0x0A0	MSIXCAP	MSI-X Capability and Control Register
0x0A4	MSIXTBL	MSI-X Table Offset Register
0x0A8	MSIXPBA	MSI-X Pending Bit Array Offset Register
PCI Power Management Capability Structure Registers		
0x0C0	PMCAP	PCI Power Management Capabilities Register
0x0C4	PMCSR	PCI Power Management Control and Status Register
Message Signaled Interrupt Capability Structure Registers		
0x0D0	MSICAP	MSI Capability and Control Register
0x0D4	MSIADDR	MSI Address Register
0x0D8	MSIUADDR	MSI Upper Address Register
0x0DC	MSIMDATA	MSI Message Data Register
0x0E0	MSIMASK	MSI Mask Register
0x0E4	MSIPENDING	MSI Pending Register
Subsystem ID and Subsystem Vendor ID Registers		
0x0F0	SSIDSSVIDCAP	Subsystem ID and Subsystem Vendor ID Capability Register
0x0F4	SSIDSSVID	Subsystem ID and Subsystem Vendor ID Register
Extended Configuration Space Access Registers		
0x0F8	ECFGADDR	Extended Configuration Space Access Address Register
0x0FC	ECFGDATA	Extended Configuration Space Access Data Register
Advanced Error Reporting (AER) Extended Capability Registers		
0x100	AERCAP	AER Capabilities Register
0x104	AERUES	AER Uncorrectable Error Status Register
0x108	AERUEM	AER Uncorrectable Error Mask Register
0x10C	AERUESV	AER Uncorrectable Error Severity Register
0x110	AERCES	AER Correctable Error Status Register
0x114	AERCEM	AER Correctable Error Mask Register
0x118	AERCTL	AER Control Register

Table 79: PCIe Register Map *(Continued)*

Offset	Register Mnemonic	See
0x11C	AERHL1DW	AER Header Log 1st Doubleword Register
0x120	AERHL2DW	AER Header Log 2nd Doubleword Register
0x124	AERHL3DW	AER Header Log 3rd Doubleword Register
0x128	AERHL4DW	AER Header Log 4th Doubleword Register
Device Serial Number Extended Capability Registers		
0x180	SNUMCAP	Serial Number Capabilities Register
0x184	SNUMLDW	Serial Number Lower Doubleword Register
0x188	SNUMUDW	Serial Number Upper Doubleword Register
Endpoint Configuration and Status Registers		
0x400	EPCTL	Endpoint Control Register
0x404	EPSTS	Endpoint Status Register
0x40C	SEDELAY	Side Effect Delay Register
0x440	BARSETUP0	BAR 0 Setup Register
0x444	BARSETUP1	BAR 1 Setup Register
0x448	BARSETUP2	BAR 2 Setup Register
0x44C	BARSETUP3	BAR 3 Setup Register
0x450	BARSETUP4	BAR 4 Setup Register
0x454	BARSETUP5	BAR 5 Setup Register
Internal Error Control and Status Registers		
0x480	IERRORCTL	Internal Error Reporting Control Register
0x484	IERRORSTS0	Internal Error Reporting Status 0 Register
0x488	IERRORMSK0	Internal Error Reporting Mask 0 Register
0x48C	IERRORSEV0	Internal Error Reporting Severity 0 Register
0x494	IERRORTST0	Internal Error Reporting Test 0 Register
0x4B0	TOCTL	Timeout Control Register
0x4B4	IFBTOCNT	IFB Timeout Count Register
0x4B8	EFBTOCNT	EFB Timeout Count Register
0x4BC	TOTSCTL	Timeout Timestamp Control Register
0x4C0	MECTL	Memory Error Control Register

Table 79: PCIe Register Map *(Continued)*

Offset	Register Mnemonic	See
Physical Layer Control and Status Registers		
0x510	SERDESCFG	SerDes Configuration Register
0x514	SERDESSTS0	SerDes Status 0 Register
0x51C	LANESTS0	Lane Status 0 Register
0x520	LANESTS1	Lane Status 1 Register
0x524	LANESTS2	Lane Status 2 Register
0x528	PHYFSMT0	PHY State Machine Timing Configuration 0 Register
0x52C	PHYFSMT1	PHY State Machine Timing Configuration 1 Register
0x530	PHYLCFG0	PHY Link Configuration 0 Register
0x534	PHYLCFG1	PHY Link Configuration 1 Register
0x538	PHYLSTS0	PHY Link Status 0 Register
0x53C	PHYLSTS1	PHY Link Status 1 Register
0x540	PHYLSTATE0	PHY Link State 0 Register
0x544	PHYLTSSMSTS0	PHY Link LTSSM Status 0 Register
0x548	PHYLTSSMSTS1	PHY Link LTSSM Status 1 Register
0x54C	PHYCNT0	PHY Countables 0 Register
0x550	PHYCNT1	PHY Countables 1 Register
0x554	PHYCNTCFG	PHY Countables Configuration Register
0x558	PHYRECEL	PHY Recovery Entry Log Register
0x55C	PHYPRBS	PHY PRBS Seed Register
Data Link Layer Control and Status Registers		
0x600	DLCTL1	Data Link Control 1 Register
0x604	DLCTL2	Data Link Control 2 Register
0x608	DLCTL3	Data Link Control 3 Register
0x60C	DLSTS	Data Link Status Register
0x610	DLRXSTS	Data Link Receive Status Register
0x614	DLTXSTS	Data Link Transmit Status Register
0x618	DLCNT0	Data Link Countables 0 Register
0x61C	DLCNT1	Data Link Countables 1 Register

Table 79: PCIe Register Map *(Continued)*

Offset	Register Mnemonic	See
0x620	DLCNTCFG	Data Link Countables Configuration Register
Transaction Control and Status Registers		
0x680	TLSTSE	Transaction Layer Status Register
0x684	TLCTL	Transaction Layer Control Register
0x688	TLCNT0	Transaction Layer Countables 0 Register
0x68C	TLCNT1	Transaction Layer Countables 1 Register
0x690	TLCNTCFG	Transaction Layer Countables Configuration Register
0x6A0	INTSTS	Legacy Interrupt Status Register
Power Management Control and Status Registers		
0x700	PMPC0	Power Management Proprietary Control 0 Register
0x704	PMPC1	Power Management Proprietary Control 1 Register
Flow Control Registers		
0x800	FCVC0PTCC	Flow Control VC0 Posted Credits Consumed Register
0x804	FCVC0NPCC	Flow Control VC0 Non-Posted Credits Consumed Register
0x808	FCVC0CPCC	Flow Control VC0 Completion Credits Consumed Register
0x80C	FCVC0PTCL	Flow Control VC0 Posted Credit Limit Register
0x810	FCVC0NPCL	Flow Control VC0 Non-Posted Credit Limit Register
0x814	FCVC0CPCL	Flow Control VC0 Completion Credit Limit Register
0x818	FCVC0PTCA	Flow Control VC0 Posted Credits Allocated Register
0x81C	FCVC0NPCA	Flow Control VC0 Non-Posted Credits Allocated Register
0x820	FCVC0CPCA	Flow Control VC0 Completion Credits Allocated Register
0x824	FCVC0PTCR	Flow Control VC0 Posted Credits Received Register
0x828	FCVC0NPCR	Flow Control VC0 Non-Posted Credits Received Register
0x82C	FCVC0CPCR	Flow Control VC0 Completion Credits Received Register
0x860	EFBTC	Egress Frame Buffer Transmission Control Register
IFB and EFB Countables Registers		
0x8B0	IFBCNT0	IFB Countables 0 Register
0x8B4	IFBCNT1	IFB Countables 1 Register
0x8B8	IFBCNTCFG	IFB Countables Configuration Register

Table 79: PCIe Register Map *(Continued)*

Offset	Register Mnemonic	See
0x8C0	EFBCNT0	EFB Countables 0 Register
0x8C4	EFBCNT1	EFB Countables 1 Register
0x8C8	EFBCNTCFG	EFB Countables Configuration Register
AER Error Emulation Registers		
0xD90	UEEM	Uncorrectable Error Emulation Register
0xD94	CEEM	Correctable Error Emulation Register
SerDes Test Mode Registers		
0xE54	STMCTL	SerDes Test Mode Control Register
0xE58	STMSTS	SerDes Test Mode Status Register
0xE5C	STMTCTL	SerDes Test Mode Test Control Register
0xE60	STMTSTS	SerDes Test Mode Test Status Register
0xE64	STMECNT0	SerDes Test Mode Lane 0 Error Count Register
0xE68	STMECNT1	SerDes Test Mode Lane 1 Error Count Register
0xE6C	STMECNT2	SerDes Test Mode Lane 2 Error Count Register
0xE70	STMECNT3	SerDes Test Mode Lane 3 Error Count Register
Application Layer Loopback Control and Status Registers		
0xE84	ALLCS	Application Layer Loopback Control and Status Register
IFB Credit Control and Status Registers		
0xE90	IFBVC0PTCFG	IFB VC0 Flow Control Credit Posted Configuration Register
0xE94	IFBVC0NPCFG	IFB VC0 Flow Control Credit Non-Posted Configuration Register
0xE98	IFBVC0CPCFG	IFB VC0 Flow Control Credit Completion Configuration Register
0xEA8	IFCSTS	Ingress Flow Control Status Register
EFB Statistics Registers		
0xEC0	EFBVC0PTSTS	EFB VC0 Posted Credit Availability Register
0xEC4	EFBVC0NPSTS	EFB VC0 Non-Posted Credit Availability Register
0xEC8	EFBVC0CPSTS	EFB VC0 Completion Credit Availability Register
0xECC	EFBRBSTS	EFB Replay Buffer Statistics Register

17.4 PCI Type 0 Configuration Header Registers

17.4.1 PCI Identification Register

Register name: PCI_ID Reset value: 0x80AB_111D	Register offset: 0x000

Bits	7	6	5	4	3	2	1	0
31:24	DID							
23:16	DID							
15:08	VID							
07:00	VID							

Bits	Name	Description	Type	Reset Value
31:16	DID	Device Identification 0x80AB = Tsi721	RES	0x80AB
15:0	VID	Vendor Identification This field contains the 16-bit vendor ID value assigned to IDT.	RES	0x111D

17.4.2 PCI Control and Status Register

Register name: PCI_CSR Reset value: 0x0010_0000	Register offset: 0x004

Bits	7	6	5	4	3	2	1	0
31:24	DPE	SSE	RMAS	RTAS	STAS	DEVT		MDPED
23:16	FB2B	Reserved	C66MHZ	CAPL	INTS	Reserved		
15:08	Reserved					INTXD	FB2BE	SERRE
07:00	ADSTEP	PERRE	VGAS	MWI	SCE	BME	MAE	IOAE

Bits	Name	Description	Type	Reset Value
31	DPE	Detected Parity Error 1 = Receives a poisoned TLP regardless of the state of the PERRE bit in this register.	R/W1C	0x0
30	SSE	Signaled System Error 0 = No error. 1 = A fatal (ERR_FATAL) or non-fatal (ERR_NONFATAL) error is signaled, and the SERRE bit is set in this register.	R/W1C	0x0
29	RMAS	Received Master Abort 0x0 = No error 0x1 = Received a completion with an Unsupported Request completion status	R/W1C	0x0
28	RTAS	Received Target Abort This bit is set when the Tsi721 receives a completion with Completer Abort completion status. 0x0 = No error. 0x1 = Received a completion with Completer Abort completion status	R/W1C	0x0
27	STAS	Signaled Target Abort 1 = Completed a posted or non-posted request with a completer abort error.	R/W1C	0x0
26:25	DEVT	DEVSEL# TIming. Not applicable.	R	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
24	MDPED	Master Data Parity Error Detected This bit is set by the Tsi721 when the PERRE bit in the PCI Control and Status Register is set and one of the following occurs: • Tsi721 received a completion marked as “poisoned” • Tsi721 transmitted a poisoned request	R/W1C	0x0
23	FB2B	Fast Back-to-Back (FB2B). Not applicable.	R	0x0
22	Reserved	Reserved	R	0x0
21	C66MHZ	66-MHz Capable. Not applicable.	R	0x0
20	CAPL	Capabilities List This bit is hardwired to 1 to indicate that the Tsi721 implements an extended capability list item.	R	0x1
19	INTS	INTx Status. 0x1 = An INTx interrupt is pending from the Tsi721.	R	0x0
18:11	Reserved	Reserved	R	0x0
10	INTXD	INTx Disable This bit controls Tsi721’s ability to generate an INTx interrupt message. 1 = All INTx interrupts generated by the device are negated. This can change the resolved interrupt state of the Tsi721.	R/W	0x0
9	FB2BE	Fast Back-to-Back Enable. Not applicable	R	0x0
8	SERRE	SERR Enable Non-fatal and fatal errors detected by the Tsi721 are reported to the root complex when this bit is set or the bits in the PCIe Device Control and Status Register are set. 0x0 = Disable non-fatal and fatal error reporting if also disabled in the PCIe Device Control and Status Register. 0x1 = Enable non-fatal and fatal error reporting.	R/W	0x0
7	ADSTEP	Address Data Stepping. Not applicable.	R	0x0
6	PERRE	Parity Error Enable This bit controls the logging of poisoned TLPs in the MDPED field in the PCI Control and Status Register. When this bit is cleared, poisoned TLPs are not reported as master data parity errors in the PCI Control and Status Register.	R/W	0x0
5	VGAS	VGA Palette Snoop. Not applicable.	R	0x0
4	MWI	Memory Write Invalidate. Not applicable.	R	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
3	SCE	Special Cycle Enable. Not applicable.	R	0x0
2	BME	Bus Master Enable This bit controls Tsi721's ability to issue memory and I/O read and write requests, as well as MSI and MSI-X interrupt messages. It has no affect on other requests. 0x0 = Bus master disabled. 0x1 = Bus master enabled.	R/W	0x0
1	MAE	Memory Access Enable When this bit is cleared, the Tsi721 does not respond to memory and prefetchable memory space accesses received on its primary bus (that is, a TLP that targets the Tsi721's memory or prefetchable memory space is handled as an Unsupported Request). 0x0 = Disable memory space 0x1 = Enable memory space	R/W	0x0
0	IOAE	I/O Access Enable When this bit is cleared, the Tsi721 does not respond to I/O accesses received on its primary bus (that is, a TLP that targets the Tsi721's I/O space is handled as an Unsupported Request). 0x0 = Disable I/O space 0x1 = Enable I/O space (not supported) Note: The Tsi721 does not support I/O access; this bit should not be set to 1.	R/W	0x0

17.4.3 PCI Class Register

Register name: PCI_CLASS Reset value: 0x0680_0001	Register offset: 0x008

Bits	7	6	5	4	3	2	1	0
31:24	BASE							
23:16	SUB							
15:08	INTF							
07:00	RID							

Bits	Name	Description	Type	Reset Value
31:24	BASE	Base Class Code This value indicates that the Tsi721 is a bridge device.	RES	0x06
23:16	SUB	Sub Class Code This value indicates that the Tsi721 is classified as “other.”	RES	0x80
15:8	INTF	Interface No standard interface defined.	RES	0x0
7:0	RID	Revision ID This field contains the revision identification number for the Tsi721. This field must be initialized through the configuration space register interface. Reset value A1 Revision = 0x1 A0 Revision = 0x0	RES	0x1

17.4.4 PCI Miscellaneous 0 Register

Register name: PCI_MISC0 Reset value: 0x0000_0000	Register offset: 0x00C

Bits	7	6	5	4	3	2	1	0
31:24	CAPABLE	START	Reserved		CCODE			
23:16	HDR							
15:08	LTIMER							
07:00	CLS							

Bits	Name	Description	Type	Reset Value
31	CAPABLE	BIST Capability Supported BIST is not supported and this field is hardwired to zero.	R	0x0
30	START	BIST Start BIST is not supported and this field is hardwired to zero.	R	0x0
29:28	Reserved	Reserved	R	0x0
27:24	CCODE	BIST Completion Code BIST is not supported and this field is hardwired to zero.	R	0x0
23:16	HDR	Header Type This field indicates the configuration space header for a single-function Tsi721 (type 0 header).	R	0x00
15:8	LTIMER	Latency Timer Not applicable.	R	0x00
7:0	CLS	Cache Line Size This field has no effect on the Tsi721's operation but can be read and written by software. It is implemented for compatibility with legacy software.	R/W	0x00

17.4.5 PCI BAR 0

Register name: PCI_BAR0 Reset value: 0x0000_0000	Register offset: 0x010

Bits	7	6	5	4	3	2	1	0
31:24	BADDR							
23:16	BADDR							
15:08	BADDR							
07:00	BADDR				PREF	TYPE		MEMSI

Bits	Name	Description	Type	Reset Value
31:4	BADDR	Base Address This field specifies the address bits to be used by the Tsi721 in decoding and accepting transactions. The value of the SIZE field in the BAR 0 Setup Register controls which bits in this field can be modified. Bits that cannot be modified are zero. When the MEMSI indicates memory and the TYPE field indicates 64-bit addressing, the upper bits of the address of the BADDR field are contained in the next consecutive odd numbered BAR which in this case is BAR1. For more information, see the PCI and PCIe specifications.	R/W	0x0
3	PREF	Prefetchable If the MEMSI field selects memory, this field indicates if the memory is prefetchable. If the MEMSI field indicates I/O space, this field is read-only with a value of zero. The value of this field is determined by the PREF field in the BAR 0 Setup Register. 0x0 = Non-prefetchable 0x1 = Prefetchable	R	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
2:1	TYPE	Address Type If the MEMSI field indicates memory space, this field specifies if a 32-bit or 64-bit address format is used. If the MEMSI field indicates I/O space, this field is read-only with a value of zero. The value of this field is determined by the TYPE field in the BAR 0 Setup Register. 0x0 = 32-bit addressing. Located in lower 4-GB address space 0x1 = Reserved 0x2 = 64-bit addressing 0x3 = Reserved	R	0x0
0	MEMSI	Memory Space Indicator This bit determines if the base address register maps into memory space or I/O space. The value of this field is determined by the MEMSI field in the BAR 0 Setup Register. 0x0 = Memory space 0x1 = I/O space	R	0x0

17.4.6 PCI BAR 1

When the MEMSI field in the BAR 0 Setup Register is set to memory space (that is, zero) and the TYPE field is set to 64-bit addressing, BAR1 takes on the function of the upper 32 bits of the BADDR field in BAR0 (that is, all bits in this register become read-write). Otherwise, the BAR format below is used.

Register name: PCI_BAR1 Reset value: 0x0000_0000	Register offset: 0x014

Bits	7	6	5	4	3	2	1	0
31:24	BADDR							
23:16	BADDR							
15:08	BADDR							
07:00	BADDR				PREF	TYPE		MEMSI

Bits	Name	Description	Type	Reset Value
31:4	BADDR	Base Address This field specifies the address bits to be used by the Tsi721 in decoding and accepting transactions. When the MEMSI field in the BAR 0 Setup Register is set to memory space (that is, zero) and the TYPE field is set to 64-bit addressing, the SIZE field in the BAR 0 Setup Register controls which bits in this field can be modified.Otherwise, the SIZE field in the BAR 1 Setup Register controls which bits in this field can be modified. For more information, see the SIZE field in the BAR 0 Setup Register and BAR 1 Setup Register. For more information, see the PCI and PCIe specifications.	R/W	0x0
3	PREF	Prefetchable If the MEMSI field in this register selects memory, this field indicates whether or not the memory is prefetchable. If the MEMSI field indicates I/O space, this field is read-only with a value of zero. The value of this field is determined by the PREF field in the BAR 1 Setup Register. 0x0 = Non-prefetchable 0x1 = Prefetchable	R	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
2:1	TYPE	Address Type If the MEMSI field in this register indicates memory space, this field specifies if a 32-bit or 64-bit address format is used. Since this is an odd-numbered BAR, it should only be configured with a 32-bit address format. If the MEMSI field indicates I/O space, this field is read-only with a value of zero. The value of this field is determined by the TYPE field in the BAR 1 Setup Register. 0x0 = 32-bit addressing. Located in lower 4-GB address space 0x1–0x3 = Reserved	R	0x0
0	MEMSI	Memory Space Indicator This bit determines if the BAR maps into memory space or I/O space. The value of this field is determined by the MEMSI field in the BAR 1 Setup Register. 0x0 = Memory space 0x1 = I/O space	R	0x0

17.4.7 PCI BAR 2

Register name: PCI_BAR2 Reset value: 0x0000_000C	Register offset: 0x018

Bits	7	6	5	4	3	2	1	0
31:24	BADDR							
23:16	BADDR							
15:08	BADDR							
07:00	BADDR				PREF	TYPE		MEMSI

Bits	Name	Description	Type	Reset Value
31:4	BADDR	Base Address This field specifies the address bits to be used by the Tsi721 in decoding and accepting transactions. The value of the SIZE field in the BAR 2 Setup Register controls which bits in this field can be modified. Bits that cannot be modified are zero. When the MEMSI indicates memory and the TYPE field indicates 64-bit addressing, the upper bits of the address of the BADDR field are contained in the next consecutive odd numbered BAR, which in this case is BAR3. For more information, see the PCI and PCIe specifications.	R/W	0x0
3	PREF	Prefetchable If the MEMSI field in this register selects memory, this field indicates whether or not the memory is prefetchable. If the MEMSI field indicates I/O space, this field is read-only with a value of zero. The value of this field is determined by the PREF field in the BAR 2 Setup Register. 0x0 = Non-prefetchable 0x1 = Prefetchable	R	0x1

(Continued)

Bits	Name	Description	Type	Reset Value
2:1	TYPE	Address Type If the MEMSI field indicates memory space, this field specifies if a 32-bit or 64-bit address format is used. If the MEMSI field indicates I/O space, this field is read-only with a value of zero. The value of this field is determined by the TYPE field in the BAR 2 Setup Register. 0x0 = 32-bit addressing. Located in lower 4-GB address space 0x1 = Reserved 0x2 = 64-bit addressing 0x3 = Reserved	R	0x2
0	MEMSI	Memory Space Indicator This bit determines if the base address register maps into memory space or I/O space. The value of this field is determined by the MEMSI field in the BAR 2 Setup Register. 0x0 = Memory space 0x1 = I/O space	R	0x0

17.4.8 PCI BAR 3

When the MEMSI field in the BAR 2 Setup Register is set to memory space (that is, zero) and the TYPE field is set to 64-bit addressing, BAR3 takes on the function of the upper 32 bits of the BADDR field in BAR2 (that is, all bits in this register become read-write). Otherwise, the BAR format below is used.

Register name: PCI_BAR3 Reset value: 0x0000_0000	Register offset: 0x01C

Bits	7	6	5	4	3	2	1	0
31:24	BADDR							
23:16	BADDR							
15:08	BADDR							
07:00	BADDR				PREF	TYPE		MEMSI

Bits	Name	Description	Type	Reset Value
31:4	BADDR	Base Address This field specifies the address bits to be used by the Tsi721 in decoding and accepting transactions. When the MEMSI field in the BAR 2 Setup Register is set to memory space (that is, zero) and the TYPE field is set to 64-bit addressing, the SIZE field in the BAR 2 Setup Register controls which bits in this field can be modified.Otherwise, the SIZE field in the BAR 3 Setup Register controls which bits in this field can be modified. For more information, see the SIZE field in the BAR 2 Setup Register and BAR 3 Setup Register. For more information, see the PCI and PCIe specifications.	R/W	0x0
3	PREF	Prefetchable If the MEMSI field in this register selects memory, this field indicates whether or not the memory is prefetchable. When the MEMSI field indicates I/O space, this field is read-only with a value of zero. The value of this field is determined by the PREF field in the BAR 3 Setup Register. 0x0 = Non-prefetchable 0x1 = Prefetchable	R	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
2:1	TYPE	Address Type If the MEMSI field indicates memory space, this field specifies if a 32-bit or 64-bit address format is used. Since this is an odd-numbered BAR, it should only be configured with a 32-bit address format. If the MEMSI field indicates I/O space, this field is read-only with a value of zero. The value of this field is determined by the TYPE field in the BAR 3 Setup Register. 0x0 = 32-bit addressing. Located in lower 4-GB address space 0x1–0x3 = Reserved	R	0x0
0	MEMSI	Memory Space Indicator This bit determines if the base address register maps into memory space or I/O space. The value of this field is determined by the MEMSI field in the BAR 3 Setup Register. 0x0 = Memory space 0x1 = I/O space	R	0x0

17.4.9 PCI BAR 4

Register name: PCI_BAR4 Reset value: 0x0000_0004	Register offset: 0x020

Bits	7	6	5	4	3	2	1	0
31:24	BADDR							
23:16	BADDR							
15:08	BADDR							
07:00	BADDR				PREF	TYPE		MEMSI

Bits	Name	Description	Type	Reset Value
31:4	BADDR	Base Address This field specifies the address bits to be used by the Tsi721 in decoding and accepting transactions. The value of the SIZE field in the BAR 4 Setup Register controls which bits in this field can be modified. Bits that cannot be modified are zero. When the MEMSI indicates memory and the TYPE field indicates 64-bit addressing, the upper bits of the address of the BADDR field are contained in the next consecutive odd numbered BAR which in this case is BAR5. For more information, see the PCI and PCIe specifications.	R/W	0x0
3	PREF	Prefetchable If the MEMSI field in this register selects memory, this field indicates whether or not the memory is prefetchable. When the MEMSI field indicates I/O space, this field is read-only with a value of zero. The value of this field is determined by the PREF field in the BAR 4 Setup Register. 0x0 = Non-prefetchable 0x1 = Prefetchable	R	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
2:1	TYPE	Address Type If the MEMSI field indicates memory space, this field specifies if a 32-bit or 64-bit address format is used. If the MEMSI field indicates I/O space, this field is read-only with a value of zero. The value of this field is determined by the TYPE field in the BAR 4 Setup Register. 0x0 = 32-bit addressing. Located in lower 4-GB address space 0x1 = Reserved 0x2 = 64-bit addressing 0x3 = Reserved	R	0x2
0	MEMSI	Memory Space Indicator This bit determines if the base address register maps into memory space or I/O space. The value of this field is determined by the MEMSI field in the BAR 4 Setup Register. 0x0 = Memory space 0x1 = I/O space	R	0x0

17.4.10 PCI BAR 5

When the MEMSI field in the BAR 4 Setup Register is set to memory space (that is, zero) and the TYPE field is set to 64-bit addressing, BAR5 takes on the function of the upper 32-bits of the BADDR field in BAR4 (that is, all bits in this register become read-write). Otherwise, the BAR format below is used.

Register name: PCI_BAR5 Reset value: 0x0000_0000	Register offset: 0x024

Bits	7	6	5	4	3	2	1	0
31:24	BADDR							
23:16	BADDR							
15:08	BADDR							
07:00	BADDR				PREF	TYPE		MEMSI

Bits	Name	Description	Type	Reset Value
31:4	BADDR	Base Address This field specifies the address bits to be used by the Tsi721 in decoding and accepting transactions. For more information, see the PCI and PCIe specifications. When the MEMSI field in the BAR 4 Setup Register is set to memory space (that is, zero) and the TYPE field is set to 64-bit addressing, the SIZE field in the BAR 4 Setup Register controls which bits in this field can be modified.Otherwise, the SIZE field in the BAR 5 Setup Register controls which bits in this field can be modified. For more information, see the SIZE field in the BAR 4 Setup Register and BAR 5 Setup Register.	R/W	0x0
2:1	TYPE	Address Type If the MEMSI field indicates memory space, this field specifies if a 32-bit or 64-bit address format is used. Since this is an odd-numbered BAR, it should only be configured with a 32-bit address format. If the MEMSI field indicates I/O space, this field is read-only with a value of zero. The value of this field is determined by the TYPE field in the BAR 5 Setup Register. 0x0 = 32-bit addressing. Located in lower 4 GB address space. 0x1–0x3 = Reserved	R	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
3	PREF	Prefetchable If the MEMSI field in this register selects memory, this field indicates whether or not the memory is prefetchable. If the MEMSI field indicates I/O space, this field is read-only with a value of zero. The value of this field is determined by the PREF field in the BAR 5 Setup Register. 0x0 = Non-prefetchable 0x1 = Prefetchable	R	0x0
0	MEMSI	Memory Space Indicator This bit determines if the base address register maps into memory space or I/O space. The value of this field is determined by the MEMSI field in the BAR 5 Setup Register. 0x0 = Memory space 0x1 = I/O space	R	0x0

17.4.11 PCI CardBus CIS Pointer Register

Register name: PCI_CCISPTR Reset value: 0x0000_0000	Register offset: 0x028

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

Bits	Name	Description	Type	Reset Value
31:0	CCISPTR	CardBus CIS Pointer. This register is not used by the Tsi721.	R	0x0

17.4.12 PCI Subsystem ID Register

Register name: PCI_SID Reset value: 0x0000_0000	Register offset: 0x02C

Bits	7	6	5	4	3	2	1	0
31:24	SUBID							
23:16	SUBID							
15:08	SUBVID							
07:00	SUBVID							

Bits	Name	Description	Type	Reset Value
31:16	SUBID	Subsystem ID This field identifies the subsystem. It must be initialized through the configuration space register interface.	RES	0x0
15:0	SUBVID	Subsystem Vendor ID This field identifies the vendor of the subsystem. It must be initialized through the configuration space register interface.	RES	0x0

17.4.13 PCI Expansion ROM Base Register

The Tsi721 does not support Expansion ROM.

Register name: PCI_EROMBASE Reset value: 0x0000_0000	Register offset: 0x030

Bits	7	6	5	4	3	2	1	0
31:24	BADDR							
23:16	BADDR							
15:08	BADDR					RESERVED		
07:00	RESERVED							EN

Bits	Name	Description	Type	Reset Value
31:11	BADDR	Base Address This field specifies the address bits to be used by the expansion ROM in decoding and accepting transactions.	R/W	0x0
10:1	Reserved	Reserved	R	0x0
0	EN	Expansion ROM Enable 0x0 = Disable Expansion ROM address region 0x1 = Enable Expansion ROM address region Setting this bit has no effect when the MAE bit in the PCI Control and Status Register is cleared.	R/W	0x0

17.4.14 PCI Capability Pointer Register

Register name: PCI_CAPPTR Reset value: 0x0000_0040	Register offset: 0x034

Bits	7	6	5	4	3	2	1	0
31:24	RESERVED							
23:16	RESERVED							
15:08	RESERVED							
07:00	CAPPTR							

Bits	Name	Description	Type	Reset Value
31:8	Reserved	Reserved	R	0x0
7:0	CAPPTR	Capabilities Pointer This field points to the head of the capabilities structure, PCIe Capability Register.	RES	0x40

17.4.15 PCI Miscellaneous 1 Register

Register name: PCI_MISC1 Reset value: 0x0000_0100	Register offset: 0x03C

Bits	7	6	5	4	3	2	1	0
31:24	MAXLAT							
23:16	MINGNT							
15:08	INTRPIN							
07:00	INTRLINE							

Bits	Name	Description	Type	Reset Value
31:24	MAXLAT	Maximum Latency. Not applicable.	R	0x0
23:16	MINGNT	Minimum Grant. Not applicable.	R	0x0
15:8	INTRPIN	Interrupt Pin The value in this field indicates the INTx message (for example, INTA, INTB) used by this Tsi721. This field has RES type to allow the application layer to program this field such that it matches the interrupt message type generated. 0x0 = Tsi721 does not generate INTx interrupts. 0x1 = Tsi721 generates INTA interrupts. 0x2 = Tsi721 generates INTB interrupts. 0x3 = Tsi721 generates INTC interrupts. 0x4 = Tsi721 generates INTD interrupts.	RES	0x1
7:0	INTRLINE	Interrupt Line This field communicates interrupt line routing information. Values in this field are programmed by system software and are system architecture specific. The Tsi721 does not use the value in this field.	R/W	0x0

17.5 PCIe Capability Structure Registers

17.5.1 PCIe Capability Register

Register name: PCIECAP Reset value: 0x0002_C010	Register offset: 0x040

Bits	7	6	5	4	3	2	1	0
31:24	RESERVED		IMN					SLOT
23:16	TYPE				VER			
15:08	NXTPTR							
07:00	CAPID							

Bits	Name	Description	Type	Reset Value
31:30	Reserved	Reserved	R	0x0
29:25	IMN	Interrupt Message Number. Not applicable.	RE	0x0
24	SLOT	Slot Implemented. Not applicable.	R	0x0
23:20	TYPE	Port Type This field indicates that the Tsi721 is a PCIe endpoint.	R	0x0
19:16	VER	PCIe Capability Version This field indicates the PCI-SIG defined PCIe capability structure version number.	RES	0x2
15:8	NXTPTR	Next Pointer This field points to the next capability option, PCI Power Management Capabilities Register.	RES	0xC0
7:0	CAPID	Capability ID A reset value of 0x10 identifies this capability as a PCIe capability structure.	R	0x10

17.5.2 PCIe Device Capabilities Register

Register name: PCIEDCAP Reset value: 0x0000_8FC1	Register offset: 0x044

Bits	7	6	5	4	3	2	1	0
31:24	RESERVED			FLR	CSPLS		CSPLV	
23:16	CSPLV						RESERVED	
15:08	RBERR	PIP	AIP	ABP	E1AL			E0AL
07:00	E0AL		ETAG	PFS		MPAYLOAD		

Bits	Name	Description	Type	Reset Value
31:29	Reserved	Reserved	R	0x0
28	FLR	Function Level Reset Capability. Not supported.	R	0x0
27:26	CSPLS	Captured Slot Power Limit Scale This field specifies the scale used for the Slot Power Limit Value. The value of this field is set by a Set_Slot_Power_Limit Message received by the Tsi721. 0x0 = (v1) 1.0x 0x1 = (v1p1) 0.1x 0x2 = (v0p01) 0.01x 0x3 = (v0p001x) 0.001x	R	0x0
25:18	CSPLV	Captured Slot Power Limit Value This field in combination with the Slot Power Limit Scale value, specifies the upper limit on power supplied by the slot. Power limit (in Watts) calculated by multiplying the value in this field by the value in the Slot Power Limit Scale field. The value of this field is set by a Set_Slot_Power_Limit Message received by the PCIe Interface.	R	0x0
17:16	Reserved	Reserved	R	0x0
15	RBERR	Role Based Error Reporting This bit is set to indicate that the Tsi721 supports role-based error reporting.	R	0x1
14	PIP	Power Indicator Present In PCIe 1.0a when set, this bit indicates that a Power Indicator is implemented on the card/module.	R	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
13	AIP	Attention Indicator Present In PCIe 1.0a when set, this bit indicates that an Attention Indicator is implemented on the card/module.	R	0x0
12	ABP	Attention Button Present In PCIe 1.0a when set, this bit indicates that an Attention Button is implemented on the card/module.	R	0x0
11:9	E1AL	Endpoint L1 Acceptable Latency This field indicates the acceptable total latency that an endpoint can withstand due to transition from the L1 state to the L0 state. The value defaults to 0x7 indicating that the Tsi721 places no limit on the L1 to L0 latency.	RES	0x7
8:6	E0AL	Endpoint L0s Acceptable Latency This field indicates the acceptable total latency that the Tsi721 can withstand due to transition from the L0s state to the L0 state. The value defaults to 0x7 indicating that the Tsi721 places no limit on the L0s to L0 latency.	RES	0x7
5	ETAG	Extended Tag Field Support This field indicates the maximum supported size of the Tag field as a requester. 0x0 = 5-bit Tag field supported 0x1 = 8-bit Tag field supported It must be set to 0x0 for the Tsi721 because the device does not support extended tag.	RES	0x0
4:3	PFS	Phantom Functions Supported This field indicates the support for unclaimed function number to extend the number of outstanding transactions allowed by logically combining unclaimed function numbers with the TLP’s tag identifier.	R	0x0
2:0	MPAYLOAD	Maximum Payload Size Supported This field indicates the maximum payload size that the endpoint can support for TLPs. Setting this field to a value greater than 0x1 produces undefined results. 0x0 = 128-byte maximum payload size 0x1 = 256-byte maximum payload size Others = Not supported	RES	0x1

17.5.3 PCIe Device Control and Status Register

Register name: PCIEDCTL Reset value: 0x0000_2800	Register offset: 0x048

Bits	7	6	5	4	3	2	1	0
31:24	RESERVED							
23:16	RESERVED		TP	AUXPD	URD	FED	NFED	CED
15:08	IFLR	MRRS			NOSNOOP	AUXPMEN	PFEN	ETFEN
07:00	MPS			ERO	URREN	FEREN	NFEREN	CEREN

Bits	Name	Description	Type	Reset Value
31:22	Reserved	Reserved	R	0x0
21	TP	Transactions Pending 0 = No pending completion of non-posted requests 1 = Pending completion of non-posted requests This bit indicates that the Tsi721 issued non-posted requests that have not been completed.	R	0x0
20	AUXPD	Aux Power Detected 0 = No aux power detected 1 = Aux power detected For devices that require AUX power, set this bit when AUX power is detected.	RES	0x0
19	URD	Unsupported Request Detected 0 = No error detected 1 = Error detected Errors are logged in this bit regardless of whether error reporting is enabled.	R/W1C	0x0
18	FED	Fatal Error Detected 0 = No error detected 1 = Error detected Errors are logged in this bit regardless of whether error reporting is enabled.	R/W1C	0x0
17	NFED	Non-Fatal Correctable Error Detected 0 = No error detected 1 = Error detected Errors are logged in this bit regardless of whether error reporting is enabled.	R/W1C	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
16	CED	Correctable Error Detected 0 = No error detected 1 = Error detected Errors are logged in this bit regardless of whether error reporting is enabled.	R/W1C	0x0
15	IFLR	Initiate Function Level Reset. Not supported.	R	0x0
14:12	MRRS	Maximum Read Request Size This field sets the maximum read request size for the Tsi721 as a requester. 0x0 = 128 bytes 0x1 = 256 bytes 0x2 = 512 bytes 0x3 = 1024 bytes 0x4 = 2048 bytes 0x5 = 4096 bytes Others = Reserved	R/W	0x2
11	NOSNOOP	Enable No Snoop 1 = Enable the Tsi721 to set the no snoop attribute in transactions.	R/W	0x1
10	AUXPMEN	Auxiliary Power PM Enable 1 = Enable the Tsi721 to draw AUX power independent of PME AUX power.	R/WS	0x0
9	PFEN	Phantom Function Enable. Not supported.	R	0x0
8	ETFEN	Extended Tag Field Enable 1 = Request TLPs generated by the Tsi721 use an 8-bit tag (that is, allows up to 256 outstanding requests) 0 = Request TLPs generated by the Tsi721 use a 5-bit tag (that is, allows up to 32 outstanding requests) For Tsi721, this bit must be set to 0.	R	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
7:5	MPS	Maximum Payload Size This field sets the maximum TLP payload size for the Tsi721. As a receiver, the Tsi721 must handle TLPs as large as the set value. As a transmitter, the Tsi721 must not generate TLPs exceeding the set value. This field should be set to a value less than or equal to that advertised by the MPAYLOAD field in the PCIe Device Capabilities Register. Setting this field to a value larger than that advertised in the MPAYLOAD field produces undefined results. 0x0 = 128 bytes 0x1 = 256 bytes Others = Not supported	R/W	0x0
4	ERO	Enable Relaxed Ordering 1 = Tsi721 can set the relaxed ordering bit in the attribute field of transactions it initiates that do not require strong ordering.	R/W	0x0
3	URREN	Unsupported Request Reporting Enable 0 = No error detected 1 = Error detected	R/W	0x0
2	FEREN	Fatal Error Reporting Enable 0 = No error detected 1 = Error detected	R/W	0x0
1	NFEREN	Non-Fatal Error Reporting Enable 0 = No error detected 1 = Error detected	R/W	0x0
0	CEREN	Correctable Error Reporting Enable 0 = No error detected 1 = Error detected	R/W	0x0

17.5.4 PCIe Link Capabilities Register

Register name: PCIELCAP Reset value: Undefined	Register offset: 0x04C

Bits	7	6	5	4	3	2	1	0
31:24	PORTNUM							
23:16	RESERVED		LBN	DLLLA	SDERR	CPM	L1EL	
15:08	L1EL	L0SEL			ASPMS		MAXLNKWDTH	
07:00	MAXLNKWDTH				MAXLNKSPD			

Bits	Name	Description	Type	Reset Value
31:24	PORTNUM	Port Number This field indicates the PCIe port number for the corresponding link.	R	0x0
23:22	Reserved	Reserved	R	0x0
21	LBN	Link Bandwidth Notification Capability. Not applicable to endpoints.	R	0x0
20	DLLLA	Data Link Layer Link Active Reporting. Not applicable to endpoints.	R	0x0
19	SDERR	Surprise Down Error Reporting. Not applicable to endpoints.	R	0x0
18	CPM	Clock Power Management. Not supported.	R	0x0
17:15	L1EL	L1 Exit Latency This field indicates the L1 exit latency for the specific PCIe link. Transitioning from L1 to L0 requires about 2.3 us. Therefore, a value 2 us to less than 4 us is reported with a default value of 0x2.	RES	0x2
14:12	L0SEL	L0s Exit Latency This field indicates the L0s exit latency for the specific PCIe link. Transitioning from L0s to L0 requires about 2.04 us. Thus, the default value indicates an L0s exit latency between 2 us and 4 us.	RES	0x6
11:10	ASPMS	Active State Power Management (ASPM) Support Reset value A1 Revision: 0x0 = L1 and L0s are not supported. A0 Revision: 0x1 = L0s are supported.	RES	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
9:4	MAXLNKWDTH	Maximum Link Width This field indicates the maximum link width of the specific PCIe link. This field can be overridden to allow the link width to be forced to a smaller value. Setting this field to an invalid or reserved value is allowed, and results in the PCIe Interface operating at its default value. 0x0 = Reserved 0x1 = x1 link width 0x2 = x2 link width 0x4 = x4 link width Others = Reserved	RES	0x4
3:0	MAXLNKSPD	Maximum Link Speed This field indicates the supported link speeds of the PCIe Interface. 0x1 = (Gen1) 2.5 Gbps 0x2 = (Gen2) 5 Gbps Others = Reserved This field can be overridden to allow the maximum link speed to be forced to a smaller value.	RES	Undefined

17.5.5 PCIe Link Control Register

Register name: PCIELCTL Reset value: Undefined	Register offset: 0x050

Bits	7	6	5	4	3	2	1	0
31:24	LABWSTS	LBWSTS	DLLLA	SCLK	LTRAIN	Reserved	NLW	
23:16	NLW				CLS			
15:08	RESERVED				LABWINTEN	LBWINTEN	HAWD	CLKPWRMGT
07:00	ESYNC	CCLK	LRET	LDIS	RCB	RESERVED	ASPM	

Bits	Name	Description	Type	Reset Value
31	LABWSTS	Link Autonomous Bandwidth Status. Not applicable.	R	0x0
30	LBWSTS	Link Bandwidth Management Status. Not applicable.	R	0x0
29	DLLLA	Data Link Layer Link Active. Not applicable.	R	0x0
28	SCLK	Slot Clock Configuration 1 = Tsi721 uses the same physical reference clock used by its link partner (that is, common-clock configuration). The initial value of this field depends on the clocking architecture of the system (that is, common reference clock or separate reference clock) and must be initialized by the application logic.	RES	0x0
27	LTRAIN	Link Training. Not applicable.	R	0x0
26	Reserved	Reserved	R	0x0
25:20	NLW	Negotiated Link Width This field indicates the negotiated width of the link. 0b000001 = x1 0b000010 = x2 0b000100 = x4 Others = Reserved When the MAXLNKWDTH field in the PCIe Link Capabilities Register selects a width not supported by the Tsi721, the value of this field corresponds to the setting of the MAXLNKWDTH field, regardless of the negotiated link width. When the MAXLNKWDTH field selects a width supported by the Tsi721, but the link is unable to train, the value in this field is set to 0x0.	R	Undefined

(Continued)

Bits	Name	Description	Type	Reset Value
19:16	CLS	Current Link Speed This field indicates the current link speed of the PCIe Interface. 0x1 = (Gen1) 2.5 Gbps 0x2 = (Gen2) 5 Gbps Others = Reserved	R	0x1
15:12	Reserved	Reserved	R	0x0
11	LABWINTEN	Link Autonomous Bandwidth Interrupt Enable. Not applicable.	R	0x0
10	LBWINTEN	Link Bandwidth Management Interrupt Enable. Not applicable.	R	0x0
9	HAWD	Hardware Autonomous Width Disable. Not applicable	R	0x0
8	CLKPWRMGT	Enable Clock Power Management. Not applicable	R	0x0
7	ESYNC	Extended Sync 1 = Force transmission of additional ordered sets when exiting the L0s state and when in the recovery state.	R/W	0x0
6	CCLK	Common Clock Configuration 1 = This port and the port at the opposite end of the link are operating with a distributed common reference clock. After modifying this bit in both components of the link, software must trigger a link retrain by setting the link retrain bit in the upstream component's PCIe Link Control Register.	R/W	0x0
5	LRET	Link Retrain. Not applicable.	R	0x0
4	LDIS	Link Disable. Not applicable.	R	0x0
3	RCB	Read Completion Boundary 0 = Root complex has 64-byte RCB 1 = Root complex has 128-byte RCB Configuration software can set this bit if the root reports a read completion boundary of 128 bytes.	R/W	0x0
2	Reserved	Reserved	R	0x0
1:0	ASPM	Active State Power Management (ASPM) Control. This field controls the level of ASPM supported by the link. 0x0 = Disabled All other settings are not supported.	R/W	0x0

17.5.6 PCIe Device Capabilities 2 Register

Register name: PCIEDCAP2 Reset value: 0x0000_001F	Register offset: 0x064

Bits	7	6	5	4	3	2	1	0
31:24	RESERVED							
23:16	RESERVED		E2ETPS	EFMTFS	RESERVED			
15:08	RESERVED		TPHCS		LTRMS	NROEP	CASC128S	ATOPC64S
07:00	ATOPC32S	ATOPRS	ARIFS	CTDS	CTRS			

Bits	Name	Description	Type	Reset Value
31:22	Reserved	Reserved	R	0x0
21	E2ETPS	End-to-End TLP Prefix Supported. Not supported.	R	0x0
20	EFMTFS	Extended Fmt Field Supported. Not supported.	R	0x0
19:14	Reserved	Reserved	R	0x0
13:12	TPHCS	TPH Completer Supported. Not supported.	R	0x0
11	LTRMS	LTR Mechanism Supported. Not supported.	R	0x0
10	NROEP	No R-enabled PR-PR Passing. Not applicable.	R	0x0
9	CASC128S	128-bit CAS Completer Supported. Not supported.	R	0x0
8	ATOPC64S	64-bit AtomicOp Completer Supported. Not supported.	R	0x0
7	ATOPC32S	32-bit AtomicOp Completer Supported. Not supported.	R	0x0
6	ATOPRS	AtomicOp Routing Supported. Not applicable.	R	0x0
5	ARIFS	ARI Forwarding Supported. Not applicable.	R	0x0
4	CTDS	Completion Timeout Disable Supported 0x1 = Indicates support for the completion timeout disable mechanism	RES	0x1
3:0	CTRS	Completion Timeout Ranges Supported 0xF = Indicates support for the optional completion timeout programmability mechanism for Ranges A, B, C, and D. Ranges are defined as follows: • Range A = 50 us to 10 ms • Range B = 10 ms to 250 ms • Range C = 250 ms to 4 s • Range D = 4 s to 64 s	RES	0xF

17.5.7 PCIe Device Control and Status 2 Register

Register name: PCIEDCTL2 Reset value: 0x0000_0000	Register offset: 0x068

Bits	7	6	5	4	3	2	1	0
31:24	RESERVED							
23:16	RESERVED							
15:08	E2ETLPPB	RESERVED				LTRME	IDOCE	IDORE
07:00	ATOPEB	ATOPRE	ARIFEN	CTD	CTV			

Bits	Name	Description	Type	Reset Value
31:16	Reserved	Reserved	R	0x0
15	E2ETLPPB	End-to-End TLP Prefix Blocking. Not supported.	R	0x0
14:11	Reserved	Reserved	R	0x0
10	LTRME	LTR Mechanism Enable. Not supported.	R	0x0
9	IDOCE	IDO Completion Enable. Not supported.	R	0x0
8	IDORE	IDO Request Enable. Not supported.	R	0x0
7	ATOPEB	AtomicOp Egress Blocking. Not applicable.	R	0x0
6	ATOPRE	AtomicOp Requester Enable. Not supported.	R	0x0
5	ARIFEN	ARI Forwarding Enable. Not applicable.	R	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
4	CTD	Completion Timeout Disable 1 = The completion timeout mechanism is disabled.	R/W	0x0
3:0	CTV	Completion Timeout Ranges This field contains the system programmed completion timeout value. 0x0 = 50 us to 50 ms 0x1 = 50 us to 100 us 0x2 = 1 ms to 10 ms 0x5 = 16 ms to 55 ms 0x6 = 65 ms to 210 ms 0x9 = 260 ms to 900 ms 0xA = 1 s to 3.5 s 0xD = 4 s to 13 s 0xE = 17 s to 64 s Others = Reserved Note: It is strongly recommended that the Completion Timeout mechanism not expire in less than 10 ms.	R/W	0x0

17.5.8 PCIe Link Capabilities 2 Register

Register name: PCIELCAP2 Reset value: 0x0000_0000	Register offset: 0x06C

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

Bits	Name	Description	Type	Reset Value
31:0	Reserved	Reserved	R	0x0

17.5.9 PCIe Link Control 2 Register

Register name: PCIELCTL2 Reset value: 0x0000_0002	Register offset: 0x070

Bits	7	6	5	4	3	2	1	0
31:24	RESERVED							
23:16	RESERVED							CDE
15:08	RESERVED			COMP_DE	CSOS	EMC	TM	
07:00	TM	SDE	HASD	ECOMP	TLS			

Bits	Name	Description	Type	Reset Value
31:17	Reserved	Reserved	R	0x0
16	CDE	Current De-emphasis This bit indicates the current de-emphasis level when the link operates at 5.0 Gbps. 0x0 = De-emphasis level of -6.0 dB 0x1 = De-emphasis level of -3.5 dB The value of this bit is undefined when the link operates at 2.5 Gbps.	RS	0x0
15:13	Reserved	Reserved	R	0x0
12	COMP_DE	Compliance De-emphasis This bit selects the de-emphasis value in the Polling.Compliance state when this state was entered as a result of setting the ECOMP bit in this register. 0x0 = De-emphasis level of -6.0 dB 0x1 = De-emphasis level of -3.5 dB This bit is for debug, compliance testing purposes. System firmware and software can modify this bit only during debug or compliance testing.	R/WS	0x0
11	CSOS	Compliance SOS 1 = The LTSSM is required to send SOS periodically in between the compliance and modified compliance patterns.	R/WS	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
10	EMC	Enter Modified Compliance 1 = Tsi721 transmits the modified compliance pattern if the LTSSM enters the Polling.Compliance state. This bit is for debug, compliance testing purposes only. System firmware and software can modify this bit only during debug or compliance testing. In all other cases, the system must ensure that this bit is set to the default value.	R/WS	0x0
9:7	TM	Transmit Margin This field controls the value of the non de-emphasized voltage level at the transmitter signals. This field is reset to 0x0 on entry to the LTSSM Polling.Configuration substate. 0x0 = Full level as specified by TX_AMP_FULL field of PCIe SerDes Transmitter Control 0x1 = 3/4 of full level 0x2 = 1/2 of full level 0x3 = 1/4 of full level Others = Reserved This field is for debug and compliance testing purposes only. System firmware and software can modify this field only during debug or compliance testing. In all other cases, the system must ensure that this field is set to the default value. When this field is set to “Normal Operating Range”, the SerDes transmitter drive level is selected using the TX_AMP_FULL field of PCIe SerDes Transmitter Control When this field is modified, the newly selected value is not applied until the PHY LTSSM transitions through the states in which it can modify the transmit margin setting on the line (that is, Recovery.RcvrLock).	R/WS	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
6	SDE	Selectable De-emphasis As per the *PCI Express Specification (Rev. 2.1)*, this bit is not applicable to endpoints. The Tsi721, however, allows programming of this field to select the de-emphasis preference advertised by the Tsi721 using training sets during link training. The de-emphasis on the link is selected by the link partner, which depending on its implementation can accept or ignore the Tsi721's de-emphasis preference. 0x0 = De-emphasis level of -6.0 dB 0x1 = De-emphasis level of -3.5 dB This bit has no effect when the link operates at 2.5 Gbps, or when the link operates in low-swing mode. After modifying this field, IDT recommends that the link be fully retrained by setting the FLRET bit in the PHY Link State 0 Register.	RES	0x0
5	HASD	Hardware Autonomous Speed Disable. Not applicable.	R	0x0
4	ECOMP	Enter Compliance Software is permitted to force a link into compliance mode at the speed indicated by the TLS field by setting this bit in both components on a link and then initiating a hot reset on the link.	R/WS	0x0
3:0	TLS	Target Link Speed This field sets the target compliance mode speed when software is using the ECOMP bit in this register to force a link into compliance mode. 0x1 = (Gen1) 2.5 Gbps 0x2 = (Gen2) 5.0 Gbps Others = Reserved Note: Setting this field to an unsupported value produces undefined results.	R/WS	0x2

17.6 MSI-X Capability Structure Registers

17.6.1 MSI-X Capability and Control Register

Register name: MSIXCAP Reset value: 0x0045_0011	Register offset: 0x0A0

Bits	7	6	5	4	3	2	1	0
31:24	EN	MASK	Reserved			TBLSIZE		
23:16	TBLSIZE							
15:08	NXTPTR							
07:00	CAPID							

Bits	Name	Description	Type	Reset Value
31	EN	Enable 0x0 = Disable MSI-X 0x1 = Enable MSI-X	R/W	0x0
30	MASK	Mask 0x0 = Tsi721 unmasked. A vector’s mask bit determines if the vector is masked. 0x1 = Tsi721 masked. All vectors associated with the Tsi721 are masked regardless of the state of the per vector mask.	R/W	0x0
29:27	Reserved	Reserved	R	0x0
26:16	TBLSIZE	Table Size This field determines the size of the MSI-X table. The table’s size is equal to one plus the value in this field.	RES	0x45
15:8	NXTPTR	Next Pointer A reset value of 0x0 indicates this is the last capability option.	RES	0x0
7:0	CAPID	Capability ID A reset value of 0x11 identifies this capability as a MSI capability structure.	R	0x11

17.6.2 MSI-X Table Offset Register

Register name: MSIXTBL Reset value: 0x0002_C000	Register offset: 0x0A4

Bits	7	6	5	4	3	2	1	0
31:24	OFFSET							
23:16	OFFSET							
15:08	OFFSET							
07:00	OFFSET					BIR		

Bits	Name	Description	Type	Reset Value
31:3	OFFSET	BAR Offset This field indicates the starting offset of the MSI-X table from the base address of the BAR selected by the BIR field in this register. Reset value A1 Revision = 0x00005800 A0 Revision = 0x0002C000	RES	0x00005800
2:0	BIR	BAR Index This field indicates the BAR that contains the MSI-X Table. 0x0 = BAR 0 0x1 = BAR 1 0x2 = BAR 2 0x3 = BAR 3 0x4 = BAR 4 0x5 = BAR 5 0x6–0x7 = Reserved	RES	0x0

17.6.3 MSI-X Pending Bit Array Offset Register

Register name: MSIXPBA Reset value: 0x0002_A000	Register offset: 0x0A8

Bits	7	6	5	4	3	2	1	0
31:24	OFFSET							
23:16	OFFSET							
15:08	OFFSET							
07:00	OFFSET					BIR		

Bits	Name	Description	Type	Reset Value
31:3	OFFSET	BAR Offset This field indicates the starting offset of the PBA structure from the base address of the BAR selected by the BIR field in this register. <u>Reset value</u> A1 Revision = 0x00005400 A0 Revision = 0x0002A000	RES	0x00005400
2:0	BIR	BAR Index This field indicates the BAR that contains the PBA structure. 0x0 = BAR 0 0x1 = BAR 1 0x2 = BAR 2 0x3 = BAR 3 0x4 = BAR 4 0x5 = BAR 5 0x6–0x7 = Reserved	RES	0x0

17.7 PCI Power Management Capability Structure Registers

17.7.1 PCI Power Management Capabilities Register

Register name: PMCAP Reset value: 0x0003_D001	Register offset: 0x0C0

Bits	7	6	5	4	3	2	1	0
31:24	PME					D2	D1	AUXI
23:16	AUXI		DEVSP	Reserved	PMECLK	VER		
15:08	NXTPTR							
07:00	CAPID							

Bits	Name	Description	Type	Reset Value
31:27	PME	PME Support. Not supported.	R	0x0
26	D2	D2 Support. This field indicates that the Tsi721 does not support D2.	RES	0x0
25	D1	D1 Support. This field indicates that the Tsi721 does not support D1.	RES	0x0
24:22	AUXI	AUX Current This field reports auxiliary current requirements by the Tsi721 to retain PME Context when the main power rail is removed. Since this is not implemented by Tsi721, it should be set to zero. 0x0 = Self powered 0x1 = Maximum current required is 55 mA 0x2 = Maximum current required is 100 mA 0x3 = Maximum current required is 160 mA 0x4 = Maximum current required is 220 mA 0x5 = Maximum current required is 270 mA 0x6 = Maximum current required is 320 mA 0x7 = Maximum current required is 375 mA	RES	0x0
21	DEVSP	Device Specific Initialization The value of zero indicates that no device-specific initialization is required.	RES	0x0
20	Reserved	Reserved	R	0x0
19	PMECLK	PME Clock. Does not apply to PCIe.	R	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
18:16	VER	Power Management Capability Version Complies with the *PCI Bus Power Management Interface Specification, Revision 1.2*.	R	0x3
15:8	NXTPTR	Next Pointer This field points to the next capability option, MSI Capability and Control Register.	RES	0x0D0
7:0	CAPID	Capability ID A reset value of 0x1 identifies this capability as a PCI power management capability structure.	R	0x1

17.7.2 PCI Power Management Control and Status Register

Register name: PMCSR Reset value: 0x0000_0008	Register offset: 0x0C4

Bits	7	6	5	4	3	2	1	0
31:24	DATA							
23:16	BPCCE	B2B3	Reserved					
15:08	PMES	DSCALE		DSEL				PMEE
07:00	Reserved				NOSOFTRST	Reserved	PSTATE	

Bits	Name	Description	Type	Reset Value
31:24	DATA	Data. This optional field is not implemented.	R	0x0
23	BPCCE	Bus Power/Clock Control Enable. Does not apply to PCIe.	R	0x0
22	B2B3	B2/B3 Support. Does not apply to PCIe.	R	0x0
21:16	Reserved	Reserved	R	0x0
15	PMES	PME Status. Not supported.	R	0x0
14:13	DSCALE	Data Scale. The optional data register is not implemented.	R	0x0
12:9	DSEL	Data Select. The optional data register is not implemented.	R	0x0
8	PMEE	PME Enable. Not applicable since the Tsi721 does not support generation of PME events.	R	0x0
7:4	Reserved	Reserved	R	0x0
3	NOSOFTRST	No Soft Reset This bit indicates if the configuration context is preserved by the Tsi721 when the device transitions from a D3hot to D0 power management state. 0x0 = State reset 0x1 = State preserved	RES	0x1
2	Reserved	Reserved	R	0x0
1:0	PSTATE	Power State This field determines the current power state of the Tsi721 and to set a new power state. 0x0 = D0 state 0x1 = D1 state (Not supported; reserved) 0x2 = D2 state (Not supported; reserved) 0x3 = $D3_{hot}$ state	R/W	0x0

17.8 Message Signaled Interrupt Capability Structure Registers

17.8.1 MSI Capability and Control Register

Register name: MSICAP Reset value: 0x0180_F005	Register offset: 0x0D0

Bits	7	6	5	4	3	2	1	0
31:24	Reserved							MASKCAP
23:16	A64	MME			MMC			EN
15:08	NXTPTR							
07:00	CAPID							

Bits	Name	Description	Type	Reset Value
31:25	Reserved	Reserved	R	0x0
24	MASKCAP	Per Vector Masking Capable 1 = Tsi721 supports per vector masking	RES	0x1
23	A64	64-bit Address Capable 1 = Tsi721 can generate messages using a 64-bit address	RES	0x1
22:20	MME	Multiple Message Enable This field should be set to 0.	R/W	0x0
19:17	MMC	Multiple Message Capable This field contains the number of requested vectors. The PCIe Interface supports only 1 vector.	RES	0x0
16	EN	Enable 0x0 = Disable MSI 0x1 = Enable MSI	R/W	0x0
15:8	NXTPTR	Next Pointer This field points to the next capability option, Subsystem ID and Subsystem Vendor ID Capability Register.	RES	0x0F0
7:0	CAPID	Capability ID A reset value of 0x5 identifies this capability as a MSI capability structure.	R	0x5

17.8.2 MSI Address Register

Register name: MSIADDR Reset value: 0x0000_0000	Register offset: 0x0D4

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

Bits	Name	Description	Type	Reset Value
31:2	ADDR	Message Address This field specifies the lower portion of the dword address of the MSI memory write transaction.	R/W	0x0
1:0	Reserved	Reserved	R	0x0

17.8.3 MSI Upper Address Register

Register name: MSIUADDR Reset value: 0x0000_0000	Register offset: 0x0D8

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

Bits	Name	Description	Type	Reset Value
31:0	UADDR	Upper Message Address This field specifies the upper portion of the dword address of the MSI memory write transaction. If the contents of this field are non-zero, then 64-bit address is used in the MSI memory write transaction. If the contents of this field are zero, then the 32-bit address specified in the MSI Address Register is used.	R/W	0x0

17.8.4 MSI Message Data Register

Register name: MSIMDATA Reset value: 0x0000_0000	Register offset: 0x0DC

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

Bits	Name	Description	Type	Reset Value
31:16	Reserved	Reserved	R	0x0
15:0	MDATA	Message Data This field contains the lower 16 bits of data that are written when an MSI is signaled.	R/W	0x0

17.8.5 MSI Mask Register

Register name: MSIMASK Reset value: 0x0000_0000	Register offset: 0x0E0

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

Bits	Name	Description	Type	Reset Value
31:0	MASK	Mask Bits Each bit in this field represents an enabled MSI vector. When a bit is set, the corresponding MSI vector is masked from generating an interrupt.	R/W	0x0

17.8.6 MSI Pending Register

Register name: MSIPENDING Reset value: 0x0000_0000	Register offset: 0x0E4

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

Bits	Name	Description	Type	Reset Value
31:0	PENDING	Pending Bits Each bit in this field represents an enabled MSI vector. When a bit is set, the corresponding MSI vector has a pending associated message.	R	0x0

17.9 Subsystem ID and Subsystem Vendor ID Registers

17.9.1 Subsystem ID and Subsystem Vendor ID Capability Register

Register name: SSIDSSVIDCAP Reset value: 0x0000_a00D	Register offset: 0x0F0

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

Bits	Name	Description	Type	Reset Value
31:16	Reserved	Reserved	R	0x0
15:8	NXTPTR	Next Pointer This field points to the next capability option, MSI-X Capability and Control Register.	RES	0x0A0
7:0	CAPID	Capability ID A reset value of 0xD identifies this capability as a SSID/SSVID capability structure.	R	0xD

17.9.2 Subsystem ID and Subsystem Vendor ID Register

Register name: SSIDSSVID Reset value: 0x0000_0000	Register offset: 0x0F4

Bits	7	6	5	4	3	2	1	0
31:24	SSID							
23:16	SSID							
15:08	SSVID							
07:00	SSVID							

Bits	Name	Description	Type	Reset Value
31:16	SSID	Subsystem ID This field identifies the add-in card or subsystem. SSID values are assigned by the vendor.	RES	0x0
15:0	SSVID	Subsystem Vendor ID This field identifies the manufacturer of the add-in card or subsystem.	RES	0x0

17.10 Extended Configuration Space Access Registers

17.10.1 Extended Configuration Space Access Address Register

Register name: ECFGADDR Reset value: 0x0000_0000	Register offset: 0x0F8

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

Bits	Name	Description	Type	Reset Value
31:12	Reserved	Reserved	R	0x0
11:8	EREG	Extended Register Number This field selects the extended configuration register number as defined by Section 7.2.2 of the *PCI Express Specification (Rev. 2.1)*. Note: Do not program this field to point to the address offset of this register (that is, 0xF8) or the Extended Configuration Space Access Data Register (that is, 0xFC). Violation of this rule produces undefined results.	R/W	0x0
7:2	REG	Register Number This field selects the configuration register number as defined by Section 7.2.2 of the *PCI Express Specification (Rev. 2.1)*. The following restrictions apply when programming this register: Note: Do not program this field to point to the address offset of this register (that is, 0xF8) or the Extended Configuration Space Access Data Register (that is, 0xFC). Violation of this rule produces undefined results.	R/W	0x0
1:0	Reserved	Reserved	R	0x0

17.10.2 Extended Configuration Space Access Data Register

Register name: ECFGDATA Reset value: 0x0000_0000	Register offset: 0x0FC

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

Bits	Name	Description	Type	Reset Value
31:0	DATA	Configuration Data A read of this field returns the configuration space reset value pointed to by the Extended Configuration Space Access Address Register. A write to this field updates the contents of the configuration space register pointed to by the Extended Configuration Space Access Address Register with the value written. For both reads and writes, the byte enables correspond to those used to access this field. This register can be accessed only using PCIe configuration requests. Accessing this register through other interfaces (such as S-RIO or I2C) produces undefined results.	R/W	0x0

17.11 Advanced Error Reporting (AER) Extended Capability Registers

17.11.1 AER Capabilities Register

Register name: AERCAP Reset value: 0x1802_0001	Register offset: 0x100

Bits	7	6	5	4	3	2	1	0
31:24	NXTPTR							
23:16	NXTPTR				CAPVER			
15:08	CAPID							
07:00	CAPID							

Bits	Name	Description	Type	Reset Value
31:20	NXTPTR	Next Pointer. This field points to the next capability option, Serial Number Capabilities Register.	RES	0x180
19:16	CAPVER	Capability Version A reset value of 0x2 indicates compatibility with the *PCI Express Specification (Rev. 2.1)*.	R	0x2
15:0	CAPID	Capability ID A reset value of 0x1 indicates an Advanced Error Reporting capability structure.	R	0x1

17.11.2 AER Uncorrectable Error Status Register

Register name: AERUES Reset value: 0x0000_0000	Register offset: 0x104

Bits	7	6	5	4	3	2	1	0
31:24	Reserved						TLPPBE	ATOPEB
23:16	MCBLKTLP	UIE	ACSV	UR	ECRC	MALFORMED	RCVOVR	UECOMP
15:08	CABORT	COMPTO	FCPERR	POISONED	Reserved			
07:00	Reserved		SDOENERR	DLPERR	Reserved			UDEF

Bits	Name	Description	Type	Reset Value
31:26	Reserved	Reserved	R	0x0
25	TLPPBE	TLP Prefix Blocked Error Status. Not applicable.	R	0x0
24	ATOPEB	AtomicOp Egress Blocked Status. Not applicable.	R	0x0
23	MCBLKTLP	MC Blocked TLP Status. Not applicable	R	0x0
22	UIE	Uncorrectable Internal Error Status. 1 = An uncorrectable internal error associated with the Tsi721 is detected. When the IERROREN bit is cleared in the Internal Error Reporting Control Register, this field becomes read-only with a value of zero.	R/W1CS	0x0
21	ACSV	ACS Violation Status. Not applicable.	RS	0x0
20	UR	UR Status 1 = An unsupported request is detected.	R/W1CS	0x0
19	ECRC	ECRC Status 1 = An ECRC error is detected.	R/W1CS	0x0
18	MALFORMED	Malformed TLP Status 1 = A malformed TLP is detected.	R/W1CS	0x0
17	RCVOVR	Receiver Overflow Status 1 = A receiver overflow is detected.	R/W1CS	0x0
16	UECOMP	Unexpected Completion Status 1 = An unexpected completion is detected.	R/W1CS	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
15	CABORT	Completer Abort Status 1 = A completer abort error is detected.	R/W1CS	0x0
14	COMPTO	Completion Timeout Status 1 = A completion timeout error is detected.	R/W1CS	0x0
13	FCPERR	Flow Control Protocol Error Status. Not applicable. The endpoint does not support flow control protocol error checking.	R	0x0
12	POISONED	Poisoned TLP Status 1 = A poisoned TLP is detected.	R/W1CS	0x0
11:6	Reserved	Reserved	R	0x0
5	SDOENERR	Surprise Down Error Status. Not applicable.	R	0x0
4	DLPERR	Data Link Protocol Error Status 1 = A data link layer protocol error is detected.	R/W1CS	0x0
3:1	Reserved	Reserved	R	0x0
0	UDEF	Undefined. This bit is not used in the *PCI Express Specification (Rev. 2.1)*.	R/W1CS	0x0

17.11.3 AER Uncorrectable Error Mask Register

Register name: AERUEM Reset value: 0x0000_0000	Register offset: 0x108

Bits	7	6	5	4	3	2	1	0
31:24	Reserved						TLPPBE	ATOPEB
23:16	MCBLKTLP	UIE	ACSV	UR	ECRC	MALFORMED	RCVOVR	UECOMP
15:08	CABORT	COMPTO	FCPERR	POISONED	Reserved			
07:00	Reserved		SDOENERR	DLPERR	Reserved			UDEF

Bits	Name	Description	Type	Reset Value
31:26	Reserved	Reserved	R	0x0
25	TLPPBE	TLP Prefix Blocked Error Mask. Not applicable.	R	0x0
24	ATOPEB	AtomicOp Egress Blocked Mask. Not applicable.	R	0x0
23	MCBLKTLP	MC Blocked TLP Mask. Not applicable.	R	0x0
22	UIE	Uncorrectable Internal Error Mask 1 = The corresponding bit in the AER Uncorrectable Error Status Register is masked. When a bit is masked in the AER Uncorrectable Error Status Register, the corresponding event is not logged in the advanced capability structure, the FEPTR in the AER Control Register is not updated, and an error is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AER Uncorrectable Error Status Register. When the IERROREN bit is cleared in the Internal Error Reporting Control Register, this field becomes read-only with a value of zero.	R/WS	0x0
21	ACSV	ACS Violation Mask. Not applicable.	R	0x0
20	UR	UR Mask 1 = The corresponding bit in the AER Uncorrectable Error Status Register is masked. When a bit is masked in the AER Uncorrectable Error Status Register, the corresponding event is not logged in the AER Header Log registers, the FEPTR in the AER Control Register is not updated, and an error is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AER Uncorrectable Error Status Register.	R/WS	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
19	ECRC	ECRC Mask 1 = The corresponding bit in the AER Uncorrectable Error Status Register is masked. When a bit is masked in the AER Uncorrectable Error Status Register, the corresponding event is not logged in the AER Header Log registers, the FEPTR in the AER Control Register is not updated, and an error is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AER Uncorrectable Error Status Register.	R/WS	0x0
18	MALFORMED	Malformed TLP Mask 1 = The corresponding bit in the AER Uncorrectable Error Status Register is masked. When a bit is masked in the AER Uncorrectable Error Status Register, the corresponding event is not logged in the AER Header Log registers, the FEPTR in the AER Control Register is not updated, and an error is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AER Uncorrectable Error Status Register.	R/WS	0x0
17	RCVOVR	Receiver Overflow Mask 1 = The corresponding bit in the AER Uncorrectable Error Status Register is masked. When a bit is masked in the AER Uncorrectable Error Status Register, the corresponding event is not logged in the AER Header Log registers, the FEPTR in the AER Control Register is not updated, and an error is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AER Uncorrectable Error Status Register.	R/WS	0x0
16	UECOMP	Unexpected Completion Mask 1 = The corresponding bit in the AER Uncorrectable Error Status Register is masked. When a bit is masked in the AER Uncorrectable Error Status Register, the corresponding event is not logged in the AER Header Log registers, the FEPTR in the AER Control Register is not updated, and an error is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AER Uncorrectable Error Status Register.	R/WS	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
15	CABORT	Completer Abort Mask 1 = The corresponding bit in the AER Uncorrectable Error Status Register is masked. When a bit is masked in the AER Uncorrectable Error Status Register, the corresponding event is not logged in the AER Header Log registers, the FEPTR in the AER Control Register is not updated, and an error is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AER Uncorrectable Error Status Register.	R/WS	0x0
14	COMPTO	Completion Timeout Mask 1 = The corresponding bit in the AER Uncorrectable Error Status Register is masked. When a bit is masked in the AER Uncorrectable Error Status Register, the corresponding event is not logged in the AER Header Log registers, the FEPTR in the AER Control Register is not updated, and an error is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AER Uncorrectable Error Status Register.	R/WS	0x0
13	FCPERR	Flow Control Protocol Error Mask. Not applicable.	R	0x0
12	POISONED	Poisoned TLP Mask 1 = The corresponding bit in the AER Uncorrectable Error Status Register is masked. When a bit is masked in the AER Uncorrectable Error Status Register, the corresponding event is not logged in the AER Header Log registers, the FEPTR in the AER Control Register is not updated, and an error is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AER Uncorrectable Error Status Register.	R/WS	0x0
11:6	Reserved	Reserved	R	0x0
5	SDOENERR	Surprise Down Error Mask. Not applicable.	R	0x0
4	DLPERR	Data Link Protocol Error Mask 1 = The corresponding bit in the AER Uncorrectable Error Status Register is masked. When a bit is masked in the AER Uncorrectable Error Status Register, the corresponding event is not logged in the AER Header Log registers, the FEPTR in the AER Control Register is not updated, and an error is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AER Uncorrectable Error Status Register.	R/WS	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
3:1	Reserved	Reserved	R	0x0
0	UDEF	Undefined. This bit is not used in the *PCI Express Specification (Rev. 2.1)*.	R/WS	0x0

17.11.4 AER Uncorrectable Error Severity Register

Register name: AERUESV Reset value: 0x0046_2030	Register offset: 0x10C

Bits	7	6	5	4	3	2	1	0
31:24	Reserved						TLPPBE	ATOPEB
23:16	MCBLKTLP	UIE	ACSV	UR	ECRC	MALFORMED	RCVOVR	UECOMP
15:08	CABORT	COMPTO	FCPERR	POISONED	Reserved			
07:00	Reserved		SDOENERR	DLPERR	Reserved			UDEF

Bits	Name	Description	Type	Reset Value
31:26	Reserved	Reserved	R	0x0
25	TLPPBE	TLP Prefix Blocked Error Severity. Not applicable.	R	0x0
24	ATOPEB	AtomicOp Egress Blocked Severity. Not applicable.	R	0x0
23	MCBLKTLP	MC Blocked TLP Severity. Not Applicable.	R	0x0
22	UIE	Uncorrectable Internal Error Severity This bit controls the severity of the reported error. 0 = The event is reported as an uncorrectable error. 1 = The event is reported as a fatal error. When the IERROREN bit is cleared in the Internal Error Reporting Control Register, this bit is read-only with a value of one.	R/WS	0x1
21	ACSV	ACS Violation Severity. Not applicable.	R	0x0
20	UR	UR Severity This bit controls the severity of the reported error. 0 = The event is reported as an uncorrectable error. 1 = The event is reported as a fatal error.	R/WS	0x0
19	ECRC	ECRC Severity This bit controls the severity of the reported error. 0 = The event is reported as an uncorrectable error. 1 = The event is reported as a fatal error.	R/WS	0x0
18	MALFORMED	Malformed TLP Severity This bit controls the severity of the reported error. 0 = The event is reported as an uncorrectable error. 1 = The event is reported as a fatal error.	R/WS	0x1

(Continued)

Bits	Name	Description	Type	Reset Value
17	RCVOVR	Receiver Overflow Severity This bit controls the severity of the reported error. 0 = The event is reported as an uncorrectable error. 1 = The event is reported as a fatal error.	R/WS	0x1
16	UECOMP	Unexpected Completion Severity This bit controls the severity of the reported error. 0 = The event is reported as an uncorrectable error. 1 = The event is reported as a fatal error.	R/WS	0x0
15	CABORT	Completer Abort Severity This bit controls the severity of the reported error. 0 = The event is reported as an uncorrectable error. 1 = The event is reported as a fatal error.	R/WS	0x0
14	COMPTO	Completion Timeout Severity This bit controls the severity of the reported error. 0 = The event is reported as an uncorrectable error. 1 = The event is reported as a fatal error.	R/WS	0x0
13	FCPERR	Flow Control Protocol Error Severity. Not applicable.	R	0x1
12	POISONED	Poisoned TLP Status Severity. This bit controls the severity of the reported error. 0 = The event is reported as an uncorrectable error. 1 = The event is reported as a fatal error.	R/WS	0x0
11:6	Reserved	Reserved	R	0x0
5	SDOENERR	Surprise Down Error Severity. Not applicable.	R	0x1
4	DLPERR	Data Link Protocol Error Severity. This bit controls the severity of the reported error. 0 = The event is reported as an uncorrectable error. 1 = The event is reported as a fatal error.	R/WS	0x1
3:1	Reserved	Reserved	R	0x0
0	UDEF	Undefined. This bit is not used.	R/WS	0x0

17.11.5 AER Correctable Error Status Register

Register name: AERCES Reset value: 0x0000_0000	Register offset: 0x110

Bits	7	6	5	4	3	2	1	0
31:24	Reserved							
23:16	Reserved							
15:08	HLO	CIE	ADVISORY NF	RPLYTO	Reserved			RPLYROVR
07:00	BADDLLP	BADTLP	Reserved					RCVERR

Bits	Name	Description	Type	Reset Value
31:16	Reserved	Reserved	R	0x0
15	HLO	Header Log Overflow Status This bit is set when an error that requires packet-header logging occurs but the packet header cannot be logged by the Tsi721's AER Header Log registers (AERHL[1:4]DW). A packet's header cannot be logged in the AER Header Log registers when an error occurs while the FEPTR field in the AER Control Register is valid. The First Error Pointer is valid when it points to a set bit in the AER Uncorrectable Error Status Register; that is, indicating the occurrence of a prior uncorrectable error that has not been cleared by software. When the IERROREN bit is cleared in the Internal Error Reporting Control Register, this field becomes read-only with a value of zero.	R/W1CS	0x0
14	CIE	Correctable Internal Error Status This bit is set when an correctable internal error associated with the PCIe Interface is detected. When the IERROREN bit is cleared in the Internal Error Reporting Control Register, this field becomes read-only with a value of zero.	R/W1CS	0x0
13	ADVISORYNF	Advisory Non-Fatal Error Status This bit is set when an advisory non-fatal error is detected.	R/W1CS	0x0
12	RPLYTO	Replay Timer timeout Status This bit is set when the replay timer in the data link layer times out.	R/W1CS	0x0
11:9	Reserved	Reserved	R	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
8	RPLYROVR	Replay Number Rollover Status This bit is set when a replay number rollover has occurred indicating that the data link layer has abandoned replays and has requested that the link be retrained.	R/W1CS	0x0
7	BADDLLP	Bad DLLP Status This bit is set when a bad DLLP is detected.	R/W1CS	0x0
6	BADTLP	Bad TLP Status This bit is set when a bad TLP is detected.	R/W1CS	0x0
5:1	Reserved	Reserved	R	0x0
0	RCVERR	Receiver Error Status This bit is set when the physical layer detects a receiver error.	R/W1CS	0x0

17.11.6 AER Correctable Error Mask Register

Register name: AERCEM Reset value: 0x0000_E000	Register offset: 0x114

Bits	7	6	5	4	3	2	1	0
31:24	Reserved							
23:16	Reserved							
15:08	HLO	CIE	ADVISORY NF	RPLYTO	Reserved			RPLYROVR
07:00	BADDLLP	BADTLP	Reserved					RCVERR

Bits	Name	Description	Type	Reset Value
31:16	Reserved	Reserved	R	0x0
15	HLO	Header Log Overflow Mask 1 = The corresponding bit in the AER Correctable Error Status Register is masked. When a bit is masked in the AER Correctable Error Status Register, the corresponding event is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AER Correctable Error Status Register. When the IERROREN bit is cleared in the AER Correctable Error Status Register, this field becomes read-only with a value of zero.	R/WS	0x1
14	CIE	Correctable Internal Error Mask 1 = The corresponding bit in the AER Correctable Error Status Register is masked. When a bit is masked in the AER Correctable Error Status Register, the corresponding event is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AER Correctable Error Status Register. When the IERROREN bit is cleared in the Internal Error Reporting Control Register, this field becomes read-only with a value of zero.	R/WS	0x1
13	ADVISORYNF	Advisory Non-Fatal Error Mask 1 = The corresponding bit in the AER Correctable Error Status Register is masked. When a bit is masked in the AER Correctable Error Status Register, the corresponding event is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AER Correctable Error Status Register.	R/WS	0x1

(Continued)

Bits	Name	Description	Type	Reset Value
12	RPLYTO	Replay Timer timeout Mask 1 = The corresponding bit in the AER Correctable Error Status Register is masked. When a bit is masked in the AER Correctable Error Status Register, the corresponding event is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AER Correctable Error Status Register.	R/WS	0x0
11:9	Reserved	Reserved	R	0x0
8	RPLYROVR	Replay Number Rollover Mask 1 = The corresponding bit in the AER Correctable Error Status Register is masked. When a bit is masked in the AER Correctable Error Status Register, the corresponding event is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AER Correctable Error Status Register.	R/WS	0x0
7	BADDLLP	Bad DLLP Mask 1 = The corresponding bit in the AER Correctable Error Status Register is masked. When a bit is masked in the AER Correctable Error Status Register, the corresponding event is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AER Correctable Error Status Register.	R/WS	0x0
6	BADTLP	Bad TLP Mask 1 = The corresponding bit in the AER Correctable Error Status Register is masked. When a bit is masked in the AER Correctable Error Status Register, the corresponding event is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AER Correctable Error Status Register.	R/WS	0x0
5:1	Reserved	Reserved	R	0x0
0	RCVERR	Receiver Error Mask 1 = The corresponding bit in the AER Correctable Error Status Register is masked. When a bit is masked in the AER Correctable Error Status Register, the corresponding event is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AER Correctable Error Status Register.	R/WS	0x0

17.11.7 AER Control Register

Register name: AERCTL Reset value: 0x0000_00A0	Register offset: 0x118

Bits	7	6	5	4	3	2	1	0
31:24	Reserved							
23:16	Reserved							
15:08	Reserved					MHRE	MHRC	ECRCCE
07:00	ECRCCC	ECRCGE	ECRCGC	FEPTR				

Bits	Name	Description	Type	Reset Value
31:11	Reserved	Reserved	R	0x0
10	MHRE	Multiple Header Recording Enable The Tsi721 does not support recording of multiple packet headers.	R	0x0
9	MHRC	Multiple Header Recording Capable The Tsi721 does not support recording of multiple packet headers.	R	0x0
8	ECRCCE	ECRC Check Enable 1 = Enable ECRC checking	R/WS	0x0
7	ECRCCC	ECRC Check Capable 1 = Tsi721 can check ECRC	RES	0x1
6	ECRCGE	ECRC Generation Enable 1 = Enable ECRC generation	R/WS	0x0
5	ECRCGC	ECRC Generation Capable 1 = Tsi721 can generate ECRC	RES	0x1
4:0	FEPTR	First Error Pointer This field contains a pointer to the bit in the AER Uncorrectable Error Status Register that resulted in the first reported error. This field is valid only when it points to a set bit in the AER Uncorrectable Error Status Register.	RS	0x0

17.11.8 AER Header Log 1st Doubleword Register

Register name: AERHL1DW Reset value: 0x0000_0000	Register offset: 0x11C

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

Bits	Name	Description	Type	Reset Value
31:0	HL	Header Log This field contains the first dword of the TLP header that resulted in the first reported uncorrectable error.	RES	0x0

17.11.9 AER Header Log 2nd Doubleword Register

Register name: AERHL2DW Reset value: 0x0000_0000	Register offset: 0x120

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

Bits	Name	Description	Type	Reset Value
31:0	HL	Header Log This field contains the second dword of the TLP header that resulted in the first reported uncorrectable error.	RES	0x0

17.11.10 AER Header Log 3rd Doubleword Register

Register name: AERHL3DW Reset value: 0x0000_0000	Register offset: 0x124

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

Bits	Name	Description	Type	Reset Value
31:0	HL	Header Log This field contains the third dword of the TLP header that resulted in the first reported uncorrectable error.	RES	0x0

17.11.11 AER Header Log 4th Doubleword Register

Register name: AERHL4DW Reset value: 0x0000_0000	Register offset: 0x128

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

Bits	Name	Description	Type	Reset Value
31:0	HL	Header Log This field contains the fourth dword of the TLP header that resulted in the first reported uncorrectable error.	RES	0x0

17.12 Device Serial Number Extended Capability Registers

17.12.1 Serial Number Capabilities Register

Register name: SNUMCAP Reset value: 0x0001_0003	Register offset: 0x180

Bits	7	6	5	4	3	2	1	0
31:24	NXTPTR							
23:16	NXTPTR				CAPVER			
15:08	CAPID							
07:00	CAPID							

Bits	Name	Description	Type	Reset Value
31:20	NXTPTR	Next Pointer A reset value of 0x0 indicates this is the last extended capability option.	RES	0x0
19:16	CAPVER	Capability Version A reset value of 0x1 indicates compatibility with the *PCI Express Specification (Rev. 2.1)*.	R	0x1
15:0	CAPID	Capability ID A reset value of 0x3 indicates a device serial number capability structure.	R	0x3

17.12.2 Serial Number Lower Doubleword Register

Register name: SNUMLDW Reset value: 0x0000_0000	Register offset: 0x184

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

Bits	Name	Description	Type	Reset Value
31:0	SNUM	Lower 32-bits of Device Serial Number This field contains the lower 32 bits of the IEEE defined 64-bit extended unique identifier (EUI-64) assigned to the Tsi721.	RES	0x0

17.12.3 Serial Number Upper Doubleword Register

Register name: SNUMUDW Reset value: 0x0000_0000	Register offset: 0x188

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

Bits	Name	Description	Type	Reset Value
31:0	SNUM	Upper 32-bits of Device Serial Number This field contains the upper 32 bits of the IEEE defined 64-bit extended unique identifier (EUI-64) assigned to the Tsi721.	RES	0x0

17.13 Endpoint Configuration and Status Registers

17.13.1 Endpoint Control Register

Register name: EPCTL Reset value: 0x0000_0000	Register offset: 0x400

Bits	7	6	5	4	3	2	1	0
31:24	Reserved							
23:16	Reserved							
15:08	Reserved							
07:00	Reserved					EFBCTDIS	IFBCTDIS	REGUNLOCK

Bits	Name	Description	Type	Reset Value
31:3	Reserved	Reserved	R	0x0
2	EFBCTDIS	EFB Cut-Through Routing Disable. 1 = Disable cut-through routing of TLPs through the EFB	R/WS	0x0
1	IFBCTDIS	IFB Cut-Through Routing Disable 1 = Disable cut-through routing of TLPs through the IFB	R/WS	0x0
0	REGUNLOCK	Register Unlock 0 = All registers and fields denoted as RES are read-only. 1 = All registers and fields of RES type are modified when written.	R/WS	0x0

17.13.2 Endpoint Status Register

Register name: EPSTS Reset value: Undefined	Register offset: 0x404

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

Bits	Name	Description	Type	Reset Value
31:1	Reserved	Reserved	R	0x0
0	QUASIRSTSTS	Quasi Reset State When read, this bit returns a value of one when the PCIe Interface is in a quasi-reset state.	R	Undefined

17.13.3 Side Effect Delay Register

Register name: SEDELAY Reset value: 0x0000_03E8	Register offset: 0x40C

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

Bits	Name	Description	Type	Reset Value
31:16	Reserved	Reserved	R	0x0
15:0	SEDELAY	Side Effect Delay This field specifies the delay in microseconds from the generation of a completion for a configuration request with an associated side effect, to the side effect action taking place. The intent of this delay is provide sufficient time for a completion to be returned to the link partner before the side effect (for more information, see Configuration Register Side Effects). The default value corresponds to 1 millisecond.	R/WS	0x03E8

17.13.4 BAR 0 Setup Register

Register name: BARSETUP0 Reset value: 0x8000_0130	Register offset: 0x440

Bits	7	6	5	4	3	2	1	0
31:24	EN	Reserved						
23:16	Reserved							
15:08	Reserved						SIZE	
07:00	SIZE				PREF	TYPE		MEMSI

Bits	Name	Description	Type	Reset Value
31	EN	BAR Enable 0x0 = When the SIZE field is set to all zeros this disables the BAR and returns 0 when read; that is, configuration values in this register are ignored and all fields of the BAR are zero. 0x1 = Enable the BAR.	R/WS	0x1
30:10	Reserved	Reserved	R	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
9:4	SIZE	Address Space Size This field selects the size, in address bits, of the address space for the corresponding BAR or BAR pair when 64-bit addressing is selected. Assuming the SIZE field is set to a valid value, the size of the address space requested by the BADDR field in the corresponding BAR is equal to 2^{SIZE}. Bits in the BAR BADDR field correspond to PCIe address bits. For example, bit 0 or the BAR BADDR field corresponds to PCIe Address bit 4. Setting the SIZE field to a non-zero value allows bits in the BAR BADDR field that correspond to PCIe address bits greater than or equal to the SIZE field to be modified. Corresponding bits less than the SIZE field and greater than or equal to four returns a value of zero when read and cannot be modified. Setting the SIZE field to a value less than four results in all bits in the corresponding BAR BADDR field to take on a read-only zero value that effectively disables the BAR. The smallest memory size that can be requested by PCIe is 128 (that is, SIZE equal to 7) and the largest is 2^{31} bytes for 32-bit address space and 2^{63} bytes for 64-bit address space. Setting the SIZE field to a value greater than 31 when the MEMSI and TYPE fields in this register select 32-bit memory space, results in bits greater than 32 being ignored (that is, only the TYPE field can enable 64-bit addressing).	R/WS	0x13
3	PREF	Prefetchable Select This field determines the value reported in the PREF field of the corresponding BAR. 0x0 = Non-prefetchable 0x1 = Prefetchable	R/WS	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
2:1	TYPE	Address Select This field determines the value reported in the TYPE field of the corresponding BAR, and selects the address space decoding used when memory space is selected in the MEMSI field in this register. 0x0 = 32-bit addressing. Located in lower 4-GB address space 0x1 = Reserved 0x2 = 64-bit addressing 0x3 = Reserved	R/WS	0x0
0	MEMSI	MEMSI Select This field determines the MEMSI type returned in the MEMSI field of the corresponding BAR. 0x0 = Memory space 0x1 = I/O space	R/WS	0x0

17.13.5 BAR 1 Setup Register

Register name: BARSETUP1 Reset value: 0x8000_0180	Register offset: 0x444

Bits	7	6	5	4	3	2	1	0
31:24	EN	Reserved						
23:16	Reserved							
15:08	Reserved						SIZE	
07:00	SIZE				PREF	TYPE		MEMSI

Bits	Name	Description	Type	Reset Value
31	EN	BAR Enable 0x0 = When the SIZE field is set to all zeros this disables the BAR and returns 0 when read; that is, configuration values in this register are ignored and all fields of the BAR are zero. 0x1 = Enable the BAR When the MEMSI field in the BAR 0 Setup Register is set to memory space – that is, zero – and the TYPE field is set to 64-bit addressing, BAR1 takes on the function of the upper 32 bits of the BADDR field in BAR0. In this mode, this field remains R/WS but has no functional effect on the operation of the Tsi721.	R/WS	0x1
30:10	Reserved	Reserved	R	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
9:4	SIZE	Address Space Size This field selects the size, in address bits, of the address space for the corresponding BAR. When the MEMSI field in the BAR 0 Setup Register is set to memory space (that is, zero) and the TYPE field is set to 64-bit addressing, BAR1 takes on the function of the upper 32-bits of the BADDR field in BAR0. In this mode, this field remains R/WS but has no functional effect on the operation of the Tsi721. Assuming the SIZE field is set to a valid value, the size of the address space requested by the BADDR field in the corresponding BAR is equal to 2^{SIZE}. Bits in the BAR BADDR field correspond to PCIe address bits. For example, bit 0 or the BAR BADDR field corresponds to PCIe Address bit 4. Setting this SIZE field to a non-zero value allows bits in the BAR BADDR field that correspond to PCIe address bits greater than or equal to the SIZE field to be modified. Corresponding bits less than the SIZE field and greater than or equal to four returns a value of zero when read and cannot be modified. Setting the SIZE field to a value less than four results in all bits in the corresponding BAR BADDR field to take on a read-only zero value that effectively disables the BAR. The smallest memory size that can be requested by PCIe is 128 (that is, SIZE equal to 7) and the largest is 2^{31} bytes for 32-bit address space. Setting the SIZE field to a value greater than 31 results in bits greater than 32 being ignored (that is, odd BARs only support 32-bit addressing).	R/WS	0x18
3	PREF	Prefetchable Select This field determines the value reported in the PREF field of the corresponding BAR. When the MEMSI field in the BAR 0 Setup Register is set to memory space (that is, zero) and the TYPE field is set to 64-bit addressing, BAR1 takes on the function of the upper 32-bits of the BADDR field in BAR0. In this mode, this field remains R/WS but has no functional effect on the operation of the Tsi721. 0x0 = Non-prefetchable 0x1 = Prefetchable	R/WS	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
2:1	TYPE	Address Select This field determines the value reported in the TYPE field of the corresponding BAR and selects the address space decoding used when memory space is selected in the MEMSI field in this register. When the MEMSI field in the BAR 0 Setup Register is set to memory space (that is, zero) and the TYPE field is set to 64-bit addressing, BAR1 takes on the function of the upper 32-bits of the BADDR field in BAR0. In this mode, this field remains R/WS but has no functional effect on the operation of the Tsi721. 0x0 = 32-bit addressing. Located in lower 4-GB address space 0x1 = Reserved 0x2 = 64-bit addressing 0x3 = Reserved	R/WS	0x0
0	MEMSI	MEMSI Select This field determines the MEMSI type returned in the MEMSI field of the corresponding BAR. When the MEMSI field in the BAR 0 Setup Register is set to memory space (that is, zero) and the TYPE field is set to 64-bit addressing, BAR1 takes on the function of the upper 32-bits of the BADDR field in BAR0. In this mode, this field remains R/WS but has no functional effect on the operation of the Tsi721. 0x0 = Memory space 0x1 = I/O space	R/WS	0x0

17.13.6 BAR 2 Setup Register

Register name: BARSETUP2 Reset value: 0x8000_018C	Register offset: 0x448

Bits	7	6	5	4	3	2	1	0
31:24	EN	Reserved						
23:16	Reserved							
15:08	Reserved						SIZE	
07:00	SIZE				PREF	TYPE		MEMSI

Bits	Name	Description	Type	Reset Value
31	EN	BAR Enable 0x0 = When the SIZE field is set to all zeros this disables the BAR and returns 0 when read; that is, configuration values in this register are ignored and all fields of the BAR are zero. 0x1 = Enable the BAR	R/WS	0x1
30:10	Reserved	Reserved	R	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
9:4	SIZE	Address Space Size This field selects the size, in address bits, of the address space for the corresponding BAR or BAR pair when 64-bit addressing is selected. Assuming the SIZE field is set to a valid value, the size of the address space requested by the BADDR field in the corresponding BAR is equal to 2^{SIZE}. Bits in the BAR BADDR field correspond to PCIe address bits. For example, bit 0 or the BAR BADDR field corresponds to PCIe Address bit 4. Setting this SIZE field to a non-zero value allows bits in the BAR BADDR field that correspond to PCIe address bits greater than or equal to the SIZE field to be modified. Corresponding bits less than the SIZE field and greater than or equal to four returns a value of zero when read and cannot be modified. Setting the SIZE field to a value less than four results in all bits in the corresponding BAR BADDR field to take on a read-only zero value that effectively disables the BAR. The smallest memory size that can be requested by PCIe is 128 (that is, SIZE equal to 7) and the largest is 2^{31} bytes for 32-bit address space and 2^{63} bytes for 64-bit address space. When the BAR is configured to operate as an address window with lookup table address translation, valid values for the SIZE field are 14 through 37 (values greater than 31 require a 64-bit BAR). Setting the SIZE field outside this range produces undefined results. Setting the SIZE field to a value greater than 31 when the MEMSI and TYPE fields in this register select 32-bit memory space, results in bits greater than 31 being ignored (that is, only the TYPE field can enable 64-bit addressing).	R/WS	0x18
3	PREF	Prefetchable Select This field determines the value reported in the PREF field of the corresponding BAR. 0x0 = Non-prefetchable 0x1 = Prefetchable	R/WS	0x1

(Continued)

Bits	Name	Description	Type	Reset Value
2:1	TYPE	Address Select This field determines the value reported in the TYPE field of the corresponding BAR and selects the address space decoding used when memory space is selected in the MEMSI field in this register. 0x0 = 32-bit addressing. Located in lower 4-GB address space 0x1 = Reserved 0x2 = 64-bit addressing 0x3 = Reserved	R/WS	0x2
0	MEMSI	MEMSI Select This field determines the MEMSI type returned in the MEMSI field of the corresponding BAR. 0x0 = Memory space 0x1 = I/O space	R/WS	0x0

17.13.7 BAR 3 Setup Register

Register name: BARSETUP3 Reset value: 0x0000_0000	Register offset: 0x44C

Bits	7	6	5	4	3	2	1	0
31:24	EN	Reserved						
23:16	Reserved							
15:08	Reserved						SIZE	
07:00	SIZE				PREF	TYPE		MEMSI

Bits	Name	Description	Type	Reset Value
31	EN	BAR Enable 0x0 = When the SIZE field is set to all zeros this disables the BAR and return 0 when read – that is, configuration values in this register are ignored and all fields of the BAR are zero. 0x1 = Enable the BAR When the MEMSI field in BAR 2 Setup Register is set to memory space (that is, zero) and the TYPE field is set to 64-bit addressing, BAR3 takes on the function of the upper 32-bits of the BADDR field in BAR2. In this mode, this field remains R/WS but has no functional effect on the operation of the Tsi721.	R/WS	0x0
30:10	Reserved	Reserved	R	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
9:4	SIZE	Address Space Size. This field selects the size, in address bits, of the address space for the corresponding BAR. When the MEMSI field in the BAR 2 Setup Register is set to memory space (that is, zero) and the TYPE field is set to 64-bit addressing, BAR3 takes on the function of the upper 32-bits of the BADDR field in BAR2. In this mode, this field remains R/WS but has no functional effect on the operation of the Tsi721. Assuming the SIZE field is set to a valid value, the size of the address space requested by the BADDR field in the corresponding BAR is equal to 2^{SIZE}. Bits in the BAR BADDR field correspond to PCIe address bits. For example, bit 0 or the BAR BADDR field corresponds to PCIe Address bit 4. Setting this SIZE field to a non-zero value allows bits in the BAR BADDR field that correspond to PCIe address bits greater than or equal to the SIZE field to be modified. Corresponding bits less than the SIZE field and greater than or equal to four returns a value of zero when read and cannot be modified. Setting the SIZE field to a value less than four results in all bits in the corresponding BAR BADDR field to take on a read-only zero value that effectively disables the BAR. The smallest memory size that can be requested by PCIe is 128 (that is, SIZE equal to 7) and the largest is 2^{31} bytes for 32-bit address space. Setting the SIZE field to a value greater than 31 results in bits greater than 31 being ignored (that is, odd BARs only support 32-bit addressing).	R/WS	0x0
3	PREF	Prefetchable Select This field determines the value reported in the PREF field of the corresponding BAR. When the MEMSI field in the BAR 2 Setup Register is set to memory space (that is, zero) and the TYPE field is set to 64-bit addressing, BAR3 takes on the function of the upper 32 bits of the BADDR field in BAR2. In this mode, this field remains R/WS but has no functional effect on the operation of the Tsi721. 0x0 = Non-prefetchable 0x1 = Prefetchable	R/WS	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
2:1	TYPE	Address Select This field determines the value reported in the TYPE field of the corresponding BAR and selects the address space decoding used when memory space is selected in the MEMSI field in this register. When the MEMSI field in the BAR 2 Setup Register is set to memory space (that is, zero) and the TYPE field is set to 64-bit addressing, BAR3 takes on the function of the upper 32-bits of the BADDR field in BAR2. In this mode, this field remains R/WS but has no functional effect on the operation of the Tsi721. 0x0 = 32-bit addressing. Located in lower 4-GB address space 0x1 = Reserved 0x2 = 64-bit addressing 0x3 = Reserved	R/WS	0x0
0	MEMSI	MEMSI Select This field determines the MEMSI type returned in the MEMSI field of the corresponding BAR. When the MEMSI field in the BAR 2 Setup Register is set to memory space (that is, zero) and the TYPE field is set to 64-bit addressing, BAR3 takes on the function of the upper 32-bits of the BADDR field in BAR2. In this mode, this field remains R/WS but has no functional effect on the operation of the Tsi721. 0x0 = Memory space 0x1 = I/O space	R/WS	0x0

17.13.8 BAR 4 Setup Register

Register name: BARSETUP4 Reset value: 0x8000_0184	Register offset: 0x450

Bits	7	6	5	4	3	2	1	0
31:24	EN	Reserved						
23:16	Reserved							
15:08	Reserved						SIZE	
07:00	SIZE				PREF	TYPE		MEMSI

Bits	Name	Description	Type	Reset Value
31	EN	BAR Enable 0x0 = When the SIZE field is set to all zeros this disables the BAR and return 0 when read; that is, configuration values in this register are ignored and all fields of the BAR are zero. 0x1 = Enable the BAR	R/WS	0x1
30:10	Reserved	Reserved	R	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
9:4	SIZE	Address Space Size This field selects the size, in address bits, of the address space for the corresponding BAR or BAR pair when 64-bit addressing is selected. Assuming the SIZE field is set to a valid value, the size of the address space requested by the BADDR field in the corresponding BAR is equal to 2^{SIZE}. Bits in the BAR BADDR field correspond to PCIe address bits. For example, bit 0 or the BAR BADDR field corresponds to PCIe Address bit 4. Setting this SIZE field to a non-zero value allows bits in the BAR BADDR field that correspond to PCIe address bits greater than or equal to the SIZE field to be modified. Corresponding bits less than the SIZE field and greater than or equal to four returns a value of zero when read and cannot be modified. Setting the SIZE field to a value less than four results in all bits in the corresponding BAR BADDR field to take on a read-only zero value that effectively disables the BAR. The smallest memory size that can be requested by PCIe is 128 (that is, SIZE equal to 7) and the largest is 2^{31} bytes for 32-bit address space and 2^{63} bytes for 64-bit address space. When the BAR is configured to operate as an address window with lookup table address translation, valid values for the SIZE field are 14 through 37 (values greater than 31 require a 64-bit BAR). Setting the SIZE field outside this range produces undefined results. Setting the SIZE field to a value greater than 31 when the MEMSI and TYPE fields in this register select 32-bit memory space, results in bits greater than 31 being ignored (that is, only the TYPE field can enable 64-bit addressing).	R/WS	0x18
3	PREF	Prefetchable Select This field determines the value reported in the PREF field of the corresponding BAR. 0x0 = Non-prefetchable 0x1 = Prefetchable	R/WS	0x0

This document is confidential and is subject to an NDA.

(Continued)

Bits	Name	Description	Type	Reset Value
2:1	TYPE	Address Select This field determines the value reported in the TYPE field of the corresponding BAR and selects the address space decoding used when memory space is selected in the MEMSI field in this register. 0x0 = 32-bit addressing. Located in lower 4-GB address space 0x1 = Reserved 0x2 = 64-bit addressing 0x3 = Reserved	R/WS	0x2
0	MEMSI	MEMSI Select This field determines the MEMSI type returned in the MEMSI field of the corresponding BAR. 0x0 = Memory space 0x1 = I/O space	R/WS	0x0

17.13.9 BAR 5 Setup Register

Register name: BARSETUP5 Reset value: 0x0000_0000	Register offset: 0x454

Bits	7	6	5	4	3	2	1	0
31:24	EN	Reserved						
23:16	Reserved							
15:08	Reserved						SIZE	
07:00	SIZE				PREF	TYPE		MEMSI

Bits	Name	Description	Type	Reset Value
31	EN	BAR Enable 0x0 = When the SIZE field is set to all zeros this disables the BAR and return 0 when read; that is, configuration values in this register are ignored and all fields of the BAR are zero. 0x1 = Enable the BAR When the MEMSI field in the BAR 4 Setup Register is set to memory space (that is, zero) and the TYPE field is set to 64-bit addressing, BAR5 takes on the function of the upper 32-bits of the BADDR field in BAR4. In this mode, this field remains R/WS but has no functional effect on the operation of the Tsi721.	R/WS	0x0
30:10	Reserved	Reserved	R	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
9:4	SIZE	Address Space Size This field selects the size, in address bits, of the address space for the corresponding BAR. When the MEMSI field in the BAR 4 Setup Register is set to memory space (that is, zero) and the TYPE field is set to 64-bit addressing, BAR5 takes on the function of the upper 32-bits of the BADDR field in BAR4. In this mode, this field remains R/WS but has no functional effect on the operation of the Tsi721. Assuming the SIZE field is set to a valid value, the size of the address space requested by the BADDR field in the corresponding BAR is equal to 2^{SIZE}. Bits in the BAR BADDR field correspond to PCIe address bits. For example, bit 0 or the BAR BADDR field corresponds to PCIe Address bit 4. Setting this SIZE field to a non-zero value allows bits in the BAR BADDR field that correspond to PCIe address bits greater than or equal to the SIZE field to be modified. Corresponding bits less than the SIZE field and greater than or equal to four returns a value of zero when read and cannot be modified. Setting the SIZE field to a value less than four results in all bits in the corresponding BAR BADDR field to take on a read-only zero value that effectively disables the BAR. The smallest memory size that can be requested by PCIe is 128 (that is, SIZE equal to 7) and the largest is 2^{31} bytes for 32-bit address space. Setting the SIZE field to a value greater than 31 results in bits greater than 31 being ignored (that is, odd BARs only support 32-bit addressing).	R/WS	0x0
3	PREF	Prefetchable Select This field determines the value reported in the PREF field of the corresponding BAR. When the MEMSI field in the BAR 4 Setup Register is set to memory space (that is, zero) and the TYPE field is set to 64-bit addressing, BAR5 takes on the function of the upper 32 bits of the BADDR field in BAR4. In this mode, this field remains R/WS but has no functional effect on the operation of the Tsi721. 0x0 = Non-prefetchable. 0x1 = Prefetchable.	R/WS	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
2:1	TYPE	Address Select This field determines the value reported in the TYPE field of the corresponding BAR and selects the address space decoding used when memory space is selected in the MEMSI field in this register. When the MEMSI field in the BAR 4 Setup Register is set to memory space (that is, zero) and the TYPE field is set to 64-bit addressing, BAR5 takes on the function of the upper 32-bits of the BADDR field in BAR4. In this mode, this field remains R/WS but has no functional effect on the operation of the Tsi721. 0x0 = 32-bit addressing. Located in lower 4-GB address space 0x1 = Reserved 0x2 = 64-bit addressing 0x3 = Reserved	R/WS	0x0
0	MEMSI	MEMSI Select This field determines the MEMSI type returned in the MEMSI field of the corresponding BAR. When the MEMSI field in the BAR 4 Setup Register is set to memory space (that is, zero) and the TYPE field is set to 64-bit addressing, BAR5 takes on the function of the upper 32 bits of the BADDR field in BAR4. In this mode, this field remains R/WS but has no functional effect on the operation of the Tsi721. 0x0 = Memory space 0x1 = I/O space	R/WS	0x0

17.14 Internal Error Control and Status Registers

These registers control the enabling, logging, and signaling of internal errors associated with the endpoint.

17.14.1 Internal Error Reporting Control Register

Register name: IERRORCTL Reset value: 0x0000_0001	Register offset: 0x480

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

Bits	Name	Description	Type	Reset Value
31:1	Reserved	Reserved	R	0x0
0	IERROREN	Internal Error Reporting Enable 0x1 = Internal error reporting is enabled and reported through AER	R/WS	0x1

17.14.2 Internal Error Reporting Status 0 Register

Register name: IERRORSTS0 Reset value: 0x0000_0000	Register offset: 0x484

Bits	7	6	5	4	3	2	1	0
31:24	Reserved							
23:16	Reserved						RBCTLDBE	RBCTLSBE
15:08	E2EPE	EFBCTLDBE	EFBCTLSBE	EFBDATDBE	EFBDATSBE	IFBCTLDBE	IFBCTLSBE	IFBDATDBE
07:00	IFBDATSBE	EFBCPTLPTO	EFBNPTLPTO	EFBPTLPTO	Reserved	IFBCPTLPTO	IFBNPTLPTO	IFBPTLPTO

Bits	Name	Description	Type	Reset Value
31:18	Reserved	Reserved	R	0x0
17	RBCTLDBE	Replay Buffer Control Double Bit Error 1 = Detected a double bit ECC error in the Replay Buffer’s control RAM	R/W1C	0x0
16	RBCTLSBE	Replay Buffer Control Single Bit Error 1 = A single bit ECC error is detected and corrected in the Replay Buffer’s control RAM.	R/W1C	0x0
15	E2EPE	End-to-End Data Path Parity Error 1 = Detected an end-to-end data path parity error	R/W1C	0x0
14	EFBCTLDBE	EFB Control Double Bit Error 1 = Detected a double bit ECC error in the EFB control RAM	R/W1C	0x0
13	EFBCTLSBE	EFB Control Single Bit Error 1 = A single bit ECC error is detected and corrected in the EFB control RAM.	R/W1C	0x0
12	EFBDATDBE	EFB Data Double Bit Error 1 = Detected a double bit ECC error in the EFB data RAM	R/W1C	0x0
11	EFBDATSBE	EFB Data Single Bit Error 1 = A single bit ECC error is detected and corrected in the EFB data RAM.	R/W1C	0x0
10	IFBCTLDBE	IFB Control Double Bit Error 1 = Detected a double bit ECC error in the IFB control RAM	R/W1C	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
9	IFBCTLSBE	IFB Control Single Bit Error 1 = A single bit ECC error is detected and corrected in the IFB control RAM.	R/W1C	0x0
8	IFBDATDBE	IFB Data Double Bit Error 1 = Detected a double bit ECC error in the IFB data RAM	R/W1C	0x0
7	IFBDATSBE	IFB Data Single Bit Error 1 = A single bit ECC error is detected and corrected in the IFB data RAM.	R/W1C	0x0
6	EFBCPTLPTO	EFB Completion TLP Timeout 1 = Detected a completion timeout in the EFB	R/W1C	0x0
5	EFBNPTLPTO	EFB Non-Posted TLP Timeout 1 = Detected a non-posted TLP timeout in the EFB	R/W1C	0x0
4	EFBPTLPTO	EFB Posted TLP Timeout 1 = Detected a posted TLP timeout in the EFB	R/W1C	0x0
3	Reserved	Reserved	R	0x0
2	IFBCPTLPTO	IFB Completion TLP Timeout 1 = Detected a completion timeout in the IFB	R/W1C	0x0
1	IFBNPTLPTO	IFB Non-Posted TLP Timeout 1 = Detected a non-posted TLP timeout in the IFB	R/W1C	0x0
0	IFBPTLPTO	IFB Posted TLP Timeout 1 = Detected a posted TLP timeout in the IFB	R/W1C	0x0

17.14.3 Internal Error Reporting Mask 0 Register

Register name: IERRORMSK0 Reset value: 0x0002_D511	Register offset: 0x488

Bits	7	6	5	4	3	2	1	0
31:24	Reserved							
23:16	Reserved						RBCTLDBE	RBCTLSBE
15:08	E2EPE	EFBCTLDBE	EFBCTLSBE	EFBDATDBE	EFBDATSBE	IFBCTLDBE	IFBCTLSBE	IFBDATDBE
07:00	IFBDATSBE	EFBCPTLPTO	EFBNPTLPTO	EFBPTLPTO	Reserved	IFBCPTLPTO	IFBNPTLPTO	IFBPTLPTO

Bits	Name	Description	Type	Reset Value
31:18	Reserved	Reserved	R	0x0
17	RBCTLDBE	Replay Buffer Control Double Bit Error 1 = The corresponding error bit in the Internal Error Reporting Status 0 Register is masked from reporting an internal error to the AER Capability Structure of the Tsi721. This bit does not affect the state of the corresponding bit in the Internal Error Reporting Status 0 Register.	R/W	0x1
16	RBCTLSBE	Replay Buffer Control Single Bit Error 1 = The corresponding error bit in the Internal Error Reporting Status 0 Register is masked from reporting an internal error to the AER Capability Structure of the Tsi721. This bit does not affect the state of the corresponding bit in the Internal Error Reporting Status 0 Register.	R/W	0x0
15	E2EPE	End-to-End Data Path Parity Error 1 = The corresponding error bit in the Internal Error Reporting Status 0 Register is masked from reporting an internal error to the AER Capability Structure of the Tsi721. This bit does not affect the state of the corresponding bit in the Internal Error Reporting Status 0 Register.	R/W	0x1
14	EFBCTLDBE	EFB Control Double Bit Error 1 = The corresponding error bit in the Internal Error Reporting Status 0 Register is masked from reporting an internal error to the AER Capability Structure of the Tsi721. This bit does not affect the state of the corresponding bit in the Internal Error Reporting Status 0 Register.	R/W	0x1

(Continued)

Bits	Name	Description	Type	Reset Value
13	EFBCTLSBE	EFB Control Single Bit Error 1 = The corresponding error bit in the Internal Error Reporting Status 0 Register is masked from reporting an internal error to the AER Capability Structure of the Tsi721. This bit does not affect the state of the corresponding bit in the Internal Error Reporting Status 0 Register.	R/W	0x0
12	EFBDATDBE	EFB Data Double Bit Error 1 = The corresponding error bit in the Internal Error Reporting Status 0 Register is masked from reporting an internal error to the AER Capability Structure of the Tsi721. This bit does not affect the state of the corresponding bit in the Internal Error Reporting Status 0 Register.	R/W	0x1
11	EFBDATSBE	EFB Data Single Bit Error 1 = The corresponding error bit in the Internal Error Reporting Status 0 Register is masked from reporting an internal error to the AER Capability Structure of the Tsi721. This bit does not affect the state of the corresponding bit in the Internal Error Reporting Status 0 Register.	R/W	0x0
10	IFBCTLDBE	IFB Control Double Bit Error 1 = The corresponding error bit in the Internal Error Reporting Status 0 Register is masked from reporting an internal error to the AER Capability Structure of the Tsi721. This bit does not affect the state of the corresponding bit in the Internal Error Reporting Status 0 Register.	R/W	0x1
9	IFBCTLSBE	IFB Control Single Bit Error 1 = The corresponding error bit in the Internal Error Reporting Status 0 Register is masked from reporting an internal error to the AER Capability Structure of the Tsi721. This bit does not affect the state of the corresponding bit in the Internal Error Reporting Status 0 Register.	R/W	0x0
8	IFBDATDBE	IFB Data Double Bit Error 1 = The corresponding error bit in the Internal Error Reporting Status 0 Register is masked from reporting an internal error to the AER Capability Structure of the Tsi721. This bit does not affect the state of the corresponding bit in the Internal Error Reporting Status 0 Register.	R/W	0x1
7	IFBDATSBE	IFB Data Single Bit Error 1 = The corresponding error bit in the Internal Error Reporting Status 0 Register is masked from reporting an internal error to the AER Capability Structure of the Tsi721. This bit does not affect the state of the corresponding bit in the Internal Error Reporting Status 0 Register.	R/W	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
6	EFBCPTLPTO	EFB Completion TLP Timeout 1 = The corresponding error bit in the Internal Error Reporting Status 0 Register is masked from reporting an internal error to the AER Capability Structure of the Tsi721. This bit does not affect the state of the corresponding bit in the Internal Error Reporting Status 0 Register.	R/W	0x0
5	EFBNPTLPTO	EFB Non-Posted TLP Timeout 1 = The corresponding error bit in the Internal Error Reporting Status 0 Register is masked from reporting an internal error to the AER Capability Structure of the Tsi721. This bit does not affect the state of the corresponding bit in the Internal Error Reporting Status 0 Register.	R/W	0x0
4	EFBPTLPTO	EFB Posted TLP Timeout 1 = The corresponding error bit in the Internal Error Reporting Status 0 Register is masked from reporting an internal error to the AER Capability Structure of the Tsi721. This bit does not affect the state of the corresponding bit in the Internal Error Reporting Status 0 Register.	R/W	0x1
3	Reserved	Reserved	R	0x0
2	IFBCPTLPTO	IFB Completion TLP Timeout 1 = The corresponding error bit in the Internal Error Reporting Status 0 Register is masked from reporting an internal error to the AER Capability Structure of the Tsi721. This bit does not affect the state of the corresponding bit in the Internal Error Reporting Status 0 Register.	R/W	0x0
1	IFBNPTLPTO	IFB Non-Posted TLP Timeout 1 = The corresponding error bit in the Internal Error Reporting Status 0 Register is masked from reporting an internal error to the AER Capability Structure of the Tsi721. This bit does not affect the state of the corresponding bit in the Internal Error Reporting Status 0 Register.	R/W	0x0
0	IFBPTLPTO	IFB Posted TLP Timeout 1 = The corresponding error bit in the Internal Error Reporting Status 0 Register is masked from reporting an internal error to the AER Capability Structure of the Tsi721. This bit does not affect the state of the corresponding bit in the Internal Error Reporting Status 0 Register.	R/W	0x1

17.14.4 Internal Error Reporting Severity 0 Register

Register name: IERRORSEV0 Reset value: 0x0002_D511	Register offset: 0x48C

Bits	7	6	5	4	3	2	1	0
31:24	Reserved							
23:16	Reserved						RBCTLDBE	RBCTLSBE
15:08	E2EPE	EFBCTLDBE	EFBCTLSBE	EFBDATDBE	EFBDATSBE	IFBCTLDBE	IFBCTLSBE	IFBDATDBE
07:00	IFBDATSBE	EFBCPTLPTO	EFBNPTLPTO	EFBPTLPTO	Reserved	IFBCPTLPTO	IFBNPTLPTO	IFBPTLPTO

Bits	Name	Description	Type	Reset Value
31:18	Reserved	Reserved	R	0x0
17	RBCTLDBE	Replay Buffer Control Double Bit Error This bit controls how an error of the corresponding type is reported. 0 = The error is reported as an correctable internal error. 1 = The error is reported as an uncorrectable internal error.	R/W	0x1
16	RBCTLSBE	Replay Buffer Control Single Bit Error This bit controls how an error of the corresponding type is reported. 0 = The error is reported as an correctable internal error. 1 = The error is reported as an uncorrectable internal error.	R/W	0x0
15	E2EPE	End-to-End Data Path Parity Error This bit controls how an error of the corresponding type is reported. 0 = The error is reported as an correctable internal error. 1 = The error is reported as an uncorrectable internal error.	R/W	0x1
14	EFBCTLDBE	EFB Control Double Bit Error This bit controls how an error of the corresponding type is reported. 0 = The error is reported as an correctable internal error. 1 = The error is reported as an uncorrectable internal error.	R/W	0x1

(Continued)

Bits	Name	Description	Type	Reset Value
13	EFBCTLSBE	EFB Control Single Bit Error This bit controls how an error of the corresponding type is reported. 0 = The error is reported as an correctable internal error. 1 = The error is reported as an uncorrectable internal error.	R/W	0x0
12	EFBDATDBE	EFB Data Double Bit Error This bit controls how an error of the corresponding type is reported. 0 = The error is reported as an correctable internal error. 1 = The error is reported as an uncorrectable internal error.	R/W	0x1
11	EFBDATSBE	EFB Data Single Bit Error This bit controls how an error of the corresponding type is reported. 0 = The error is reported as an correctable internal error. 1 = The error is reported as an uncorrectable internal error.	R/W	0x0
10	IFBCTLDBE	IFB Control Double Bit Error This bit controls how an error of the corresponding type is reported. 0 = The error is reported as an correctable internal error. 1 = The error is reported as an uncorrectable internal error.	R/W	0x1
9	IFBCTLSBE	IFB Control Single Bit Error This bit controls how an error of the corresponding type is reported. 0 = The error is reported as an correctable internal error. 1 = The error is reported as an uncorrectable internal error.	R/W	0x0
8	IFBDATDBE	IFB Data Double Bit Error This bit controls how an error of the corresponding type is reported. 0 = The error is reported as an correctable internal error. 1 = The error is reported as an uncorrectable internal error.	R/W	0x1
7	IFBDATSBE	IFB Data Single Bit Error This bit controls how an error of the corresponding type is reported. 0 = The error is reported as an correctable internal error. 1 = The error is reported as an uncorrectable internal error.	R/W	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
6	EFBCPTLPTO	EFB Completion TLP Timeout This bit controls how an error of the corresponding type is reported. 0 = The error is reported as an correctable internal error. 1 = The error is reported as an uncorrectable internal error.	R/W	0x0
5	EFBNPTLPTO	EFB Non-Posted TLP Timeout This bit controls how an error of the corresponding type is reported. 0 = The error is reported as an correctable internal error. 1 = The error is reported as an uncorrectable internal error.	R/W	0x0
4	EFBPTLPTO	EFB Posted TLP Timeout This bit controls how an error of the corresponding type is reported. 0 = The error is reported as an correctable internal error. 1 = The error is reported as an uncorrectable internal error.	R/W	0x1
3	Reserved	Reserved	R	0x0
2	IFBCPTLPTO	IFB Completion TLP Timeout This bit controls how an error of the corresponding type is reported. 0 = The error is reported as an correctable internal error. 1 = The error is reported as an uncorrectable internal error.	R/W	0x0
1	IFBNPTLPTO	IFB Non-Posted TLP Timeout This bit controls how an error of the corresponding type is reported. 0 = The error is reported as an correctable internal error. 1 = The error is reported as an uncorrectable internal error.	R/W	0x0
0	IFBPTLPTO	IFB Posted TLP Timeout This bit controls how an error of the corresponding type is reported. 0 = The error is reported as an correctable internal error. 1 = The error is reported as an uncorrectable internal error.	R/W	0x1

17.14.5 Internal Error Reporting Test 0 Register

Register name: IERRORTST0 Reset value: 0x0000_0000	Register offset: 0x494

Bits	7	6	5	4	3	2	1	0
31:24	Reserved							
23:16	Reserved						RBCTLDBE	RBCTLSBE
15:08	E2EPE	EFBCTLDBE	EFBCTLSBE	EFBDATDBE	EFBDATSBE	IFBCTLDBE	IFBCTLSBE	IFBDATDBE
07:00	IFBDATSBE	EFBCPTLPTO	EFBNPTLPTO	EFBPTLPTO	Reserved	IFBCPTLPTO	IFBNPTLPTO	IFBPTLPTO

Bits	Name	Description	Type	Reset Value
31:18	Reserved	Reserved	R	0x0
17	RBCTLDBE	Replay Buffer Control Double Bit Error[a] 1 = Set the corresponding bit in the Internal Error Reporting Status 0 Register. If not masked, the error is reported using AER.	R/W1S	0x0
16	RBCTLSBE	Replay Buffer Control Single Bit Error[a] 1 = Set the corresponding bit in the Internal Error Reporting Status 0 Register. If not masked, the error is reported using AER.	R/W1S	0x0
15	E2EPE	End-to-End Data Path Parity Error[a] 1 = Set the corresponding bit in the Internal Error Reporting Status 0 Register. If not masked, the error is reported using AER.	R/W1S	0x0
14	EFBCTLDBE	EFB Control Double Bit Error[a] 1 = Set the corresponding bit in the Internal Error Reporting Status 0 Register. If not masked, the error is reported using AER.	R/W1S	0x0
13	EFBCTLSBE	EFB Control Single Bit Error[a] 1 = Set the corresponding bit in the Internal Error Reporting Status 0 Register. If not masked, the error is reported using AER.	R/W1S	0x0
12	EFBDATDBE	EFB Data Double Bit Error[a] 1 = Set the corresponding bit in the Internal Error Reporting Status 0 Register. If not masked, the error is reported using AER.	R/W1S	0x0
11	EFBDATSBE	EFB Data Single Bit Error[a] 1 = Set the corresponding bit in the Internal Error Reporting Status 0 Register. If not masked, the error is reported using AER.	R/W1S	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
10	IFBCTLDBE	IFB Control Double Bit Error[a] 1 = Set the corresponding bit in the Internal Error Reporting Status 0 Register. If not masked, the error is reported using AER.	R/W1S	0x0
9	IFBCTLSBE	IFB Control Single Bit Error[a] 1 = Set the corresponding bit in the Internal Error Reporting Status 0 Register. If not masked, the error is reported using AER.	R/W1S	0x0
8	IFBDATDBE	IFB Data Double Bit Error[a] 1 = Set the corresponding bit in the Internal Error Reporting Status 0 Register. If not masked, the error is reported using AER.	R/W1S	0x0
7	IFBDATSBE	IFB Data Single Bit Error[a] 1 = Set the corresponding bit in the Internal Error Reporting Status 0 Register. If not masked, the error is reported using AER.	R/W1S	0x0
6	EFBCPTLPTO	EFB Completion TLP Timeout[a] 1 = Set the corresponding bit in the Internal Error Reporting Status 0 Register. If not masked, the error is reported using AER.	R/W1S	0x0
5	EFBNPTLPTO	EFB Non-Posted TLP Timeout[a] 1 = Set the corresponding bit in the Internal Error Reporting Status 0 Register. If not masked, the error is reported using AER.	R/W1S	0x0
4	EFBPTLPTO	EFB Posted TLP Timeout[a] 1 = Set the corresponding bit in the Internal Error Reporting Status 0 Register. If not masked, the error is reported using AER.	R/W1S	0x0
3	Reserved	Reserved	R	0x0
2	IFBCPTLPTO	IFB Completion TLP Timeout[a] 1 = Set the corresponding bit in the Internal Error Reporting Status 0 Register. If not masked, the error is reported using AER.	R/W1S	0x0
1	IFBNPTLPTO	IFB Non-Posted TLP Timeout[a] 1 = Set the corresponding bit in the Internal Error Reporting Status 0 Register. If not masked, the error is reported using AER.	R/W1S	0x0
0	IFBPTLPTO	IFB Posted TLP Timeout[a] 1 = Set the corresponding bit in the Internal Error Reporting Status 0 Register. If not masked, the error is reported using AER.	R/W1S	0x0

[a] This bit returns a value of zero when read.

17.14.6 Timeout Control Register

Register name: TOCTL Reset value: 0x0000_0001	Register offset: 0x4B0

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

Bits	Name	Description	Type	Reset Value
31:1	Reserved	Reserved	R	0x0
0	ETO	Enable Timeouts 1 = Enable timeouts for the PCIe Interface. In this mode, a TLP is discarded if it has been in the PCIe Interface’s input or output queues for more than the specified timeout limit.	R/WS	0x1

17.14.7 IFB Timeout Count Register

Register name: IFBTOCNT Reset value: 0x0000_0000	Register offset: 0x4B4

Bits	7	6	5	4	3	2	1	0
31:24	Reserved							
23:16	IFBCPTOC							
15:08	IFBNPTOC							
07:00	IFBPTTOC							

Bits	Name	Description	Type	Reset Value
31:24	Reserved	Reserved	R	0x0
23:16	IFBCPTOC	IFB Completion TLP Timeout Count This field is incremented each time a TLP is discarded from the IFB completion queue because of a timeout. This counter saturates at its maximum value. Reading this field causes it to be cleared.	RCW	0x0
15:8	IFBNPTOC	IFB Non-Posted TLP Timeout Count This field is incremented each time a TLP is discarded from the IFB non-posted queue because of a timeout. This counter saturates at its maximum value. Reading this field causes it to be cleared.	RCW	0x0
7:0	IFBPTTOC	IFB Posted TLP Timeout Count This field is incremented each time a TLP is discarded from the IFB posted queue because of a timeout. This counter saturates at its maximum value. Reading this field causes it to be cleared.	RCW	0x0

17.14.8 EFB Timeout Count Register

Register name: EFBTOCNT Reset value: 0x0000_0000	Register offset: 0x4B8

Bits	7	6	5	4	3	2	1	0
31:24	Reserved							
23:16	EFBCPTOC							
15:08	EFBNPTOC							
07:00	EFBPTOC							

Bits	Name	Description	Type	Reset Value
31:24	Reserved	Reserved	R	0x0
23:16	EFBCPTOC	EFB Completion TLP Timeout Count This field is incremented each time a TLP is discarded from the EFB completion queue because of a timeout. This counter saturates at its maximum value. Reading this field causes it to be cleared.	RCW	0x0
15:8	EFBNPTOC	EFB Non-Posted TLP Timeout Count This field is incremented each time a TLP is discarded from the EFB non-posted queue because of a timeout. This counter saturates at its maximum value. Reading this field causes it to be cleared.	RCW	0x0
7:0	EFBPTOC	EFB Posted TLP Timeout Count This field is incremented each time a TLP is discarded from the EFB posted queue because of a timeout. This counter saturates at its maximum value. Reading this field causes it to be cleared.	RCW	0x0

17.14.9 Timeout Timestamp Control Register

Register name: TOTSCTL Reset value: 0x7740_0001	Register offset: 0x4BC

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

Bits	Name	Description	Type	Reset Value
31	Reserved	Reserved	R	0x0
30:22	TCOUNT	Terminal Count This field contains the value associated with bits 24 to 30 which signify a terminal count. The default value of 0x1DD corresponds to an epoch interval of about 16 sec (that is, timeout = 0xEE80_0000). The timeout value is equal to 2x the epoch interval. Thus, the default value corresponds to a timeout value of 32 sec.	R/WS	0x1DD
21:0	Reserved	Reserved	R	0x1

17.14.10 Memory Error Control Register

Register name: MECTL Reset value: 0x0000_0000	Register offset: 0x4C0

Bits	7	6	5	4	3	2	1	0
31:24	Reserved							
23:16	Reserved						RBCTLDBE	RBCTLSBE
15:08	EFBCTLDBE	EFBDATDBE	EFBCTLSBE	EFBDATSBE	IFBCTLDBE	IFBDATDBE	IFBCTLSBE	IFBDATSBE
07:00	Reserved				EIEN	Reserved		

Bits	Name	Description	Type	Reset Value
31:18	Reserved	Reserved	R	0x0
17	RBCTLDBE	Inject Replay Buffer Control Double Bit Error Writing a one to this field when the EIEN bit is set in this register results in a double bit error being injected into the next replay-buffer control quantity written. When the EIEN bit is cleared, this field returns a value of zero when read. Writing a one to this field when the EIEN bit is cleared has no effect on the Tsi721. The behavior of simultaneously injecting a single and double bit error into the same memory is undefined. Following a one being written to this field to inject an error, this field returns a one until the error is injected. After error injection, the field returns a zero when read.	R/W	0x0
16	RBCTLSBE	Inject Replay Buffer Control Single Bit Error Writing a one to this field when the EIEN bit is set in this register results in a single bit error being injected into the next replay-buffer control quantity written. When the EIEN bit is cleared, this field returns a value of zero when read. Writing a one to this field when the EIEN bit is cleared has no effect on the Tsi721. The behavior of simultaneously injecting a single and double bit error into the same memory is undefined. Following a one being written to this field to inject an error, this field returns a one until the error is injected. After error injection, the field returns a zero when read.	R/W	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
15	EFBCTLDBE	Inject EFB Control Double Bit Error Writing a one to this field when the EIEN bit is set in this register results in a double bit error being injected into the next EFB control quantity written. When the EIEN bit is cleared, this field returns a value of zero when read. Writing a one to this field when the EIEN bit is cleared has no effect on the Tsi721. The behavior of simultaneously injecting a single and double bit error into the same memory is undefined. Following a one being written to this field to inject an error, this field returns a one until the error is injected. After error injection, the field returns a zero when read.	R/W	0x0
14	EFBDATDBE	Inject EFB Data Double Bit Error Writing a one to this field when the EIEN bit is set in this register results in a double bit error being injected into the next EFB data quantity written. When the EIEN bit is cleared, this field returns a value of zero when read. Writing a one to this field when the EIEN bit is cleared has no effect on the Tsi721. The behavior of simultaneously injecting a single and double bit error into the same memory is undefined. Following a one being written to this field to inject an error, this field returns a one until the error is injected. After error injection, the field returns a zero when read.	R/W	0x0
13	EFBCTLSBE	Inject EFB Control Single Bit Error Writing a one to this field when the EIEN bit is set in this register results in a single bit error being injected into the next EFB control quantity written. When the EIEN bit is cleared, this field returns a value of zero when read. Writing a one to this field when the EIEN bit is cleared has no effect on the Tsi721. The behavior of simultaneously injecting a single and double bit error into the same memory is undefined. Following a one being written to this field to inject an error, this field returns a one until the error is injected. After error injection, the field returns a zero when read.	R/W	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
12	EFBDATSBE	Inject EFB Data Single Bit Error Writing a one to this field when the EIEN bit is set in this register results in a single bit error being injected into the next EFB data quantity written. When the EIEN bit is cleared, this field returns a value of zero when read. Writing a one to this field when the EIEN bit is cleared has no effect on the Tsi721. The behavior of simultaneously injecting a single and double bit error into the same memory is undefined. Following a one being written to this field to inject an error, this field returns a one until the error is injected. After error injection, the field returns a zero when read.	R/W	0x0
11	IFBCTLDBE	Inject IFB Control Double Bit Error Writing a one to this field when the EIEN bit is set in this register results in a double bit error being injected into the next IFB control quantity written. When the EIEN bit is cleared, this field returns a value of zero when read. Writing a one to this field when the EIEN bit is cleared has no effect on the Tsi721. The behavior of simultaneously injecting a single and double bit error into the same memory is undefined. Following a one being written to this field to inject an error, this field returns a one until the error is injected. After error injection, the field returns a zero when read.	R/W	0x0
10	IFBDATDBE	Inject IFB Data Double Bit Error Writing a one to this field when the EIEN bit is set in this register results in a double bit error being injected into the next IFB data quantity written. When the EIEN bit is cleared, this field returns a value of zero when read. Writing a one to this field when the EIEN bit is cleared has no effect on the Tsi721. The behavior of simultaneously injecting a single and double bit error into the same memory is undefined. Following a one being written to this field to inject an error, this field returns a one until the error is injected. After error injection, the field returns a zero when read.	R/W	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
9	IFBCTLSBE	Inject IFB Control Single Bit Error Writing a one to this field when the EIEN bit is set in this register results in a single bit error being injected into the next IFB control quantity written. When the EIEN bit is cleared, this field returns a value of zero when read. Writing a one to this field when the EIEN bit is cleared has no effect on the Tsi721. The behavior of simultaneously injecting a single and double bit error into the same memory is undefined. Following a one being written to this field to inject an error, this field returns a one until the error is injected. After error injection, the field returns a zero when read.	R/W	0x0
8	IFBDATSBE	Inject IFB Data Single Bit Error Writing a one to this field when the EIEN bit is set in this register results in a single bit error being injected into the next IFB data quantity written. When the EIEN bit is cleared, this field returns a value of zero when read. Writing a one to this field when the EIEN bit is cleared has no effect on the Tsi721. The behavior of simultaneously injecting a single and double bit error into the same memory is undefined. Following a one being written to this field to inject an error, this field returns a one until the error is injected. After error injection, the field returns a zero when read.	R/W	0x0
7:4	Reserved	Reserved	R	0x0
3	EIEN	Error Injection Enable 1 = Enable memory error injection. Memory error injection only applies to the IFB and EFB memories.	R/WS	0x0
2:0	Reserved	Reserved	R	0x0

17.15 Physical Layer Control and Status Registers

17.15.1 SerDes Configuration Register

Register name: SERDESCFG Reset value: 0x0000_0000	Register offset: 0x510

Bits	7	6	5	4	3	2	1	0
31:24	Reserved							
23:16	Reserved							LSE
15:08	Reserved	ILPBSEL			P2D	P1D	EIDD	FEID
07:00	Reserved				RCVD_OVRD			

Bits	Name	Description	Type	Reset Value
31:17	Reserved	Reserved	R	0x0
16	LSE	Low-Swing Mode Enable 0x0 = Enable Full-swing mode 0x1 = Enable Low-swing mode operation at the SerDes Transmit logic	R/WS	0x0
15	Reserved	Reserved	R	0x0
14:12	ILPBSEL	Internal Loopback Selection 0x0 = Disable loopback 0x1 = Enable TX/RX PMA loopback Others = Reserved	R/WS	0x0
11	P2D	P2 Power State Disable 1 = The SerDes lanes are never placed in the P2 state.	R/WS	0x0
10	P1D	P1 Power State Disable 1 = The SerDes lanes are never placed in the P1 state.	R/WS	0x0
9	EIDD	Electrical Idle Detect Disable 1 = Electrical idle detection is disabled on all lanes; that is, the electrical idle detect from the SerDes is masked. The PHY LTSSM also assume that electrical idle is never detected on all lanes when the LTSSM evaluates for this condition. Note that electrical idle can still be inferred per the rules in the *PCI Express Specification (Rev. 2.1)*. This field is not valid when the PCIe Interface operates in SerDes test mode[a].	R/WS	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
8	FEID	Force Electrical Idle Detection 1 = Electrical idle detection is forced in all lanes. The PHY LTSSM assumes that electrical idle has been detected on all lanes when the LTSSM evaluates for this condition. This field is not valid when the PCIe Interface operates in SerDes Test Mode[a].	R/WS	0x0
7:4	Reserved	Reserved	R	0x0
3:0	RCVD_OVRD	Receiver Detect Override Bit 3 = Lane 3 Bit 2 = Lane 2 Bit 1 = Lane 1 Bit 0 = Lane 0 Setting a RCVD_OVRD bit causes the associated lane to indicate that a receiver has been detected on the line. This field is not valid when the PCIe Interface operates in SerDes Test Mode[a].	R/WS	0x0

[a] For more information on SerDes test mode, see Loopbacks.

17.15.2 SerDes Status 0 Register

Register name: SERDESSTS0 Reset value: 0x0000_0000	Register offset: 0x514

Bits	7	6	5	4	3	2	1	0
31:24	Reserved			EIDLE_DETS				EIDLE_INF
23:16	EIOS_DET				EIDLE_DET			
15:08	EIDLE_DETST				CDR_LOCK			
07:00	Reserved	P0_READY	PLL_LOCK	CURR_SPEED	RCVD			

Bits	Name	Description	Type	Reset Value
31:29	Reserved	Reserved	R	0x0
28:25	EIDLE_DETS	Electrical Idle Detected-Sticky Each bit in this field corresponds to a PCIe Interface lane. This field corresponds to lanes 3 through 0. This field contains the same information as the EIDLE_DET field in this register, but this field's contents are sticky (that is, they preserve value across a hot reset).	R/W1CS	0x0
24	EIDLE_INF	Electrical Idle Inferred This bit indicates that the Electrical Idle condition has been inferred by the LTSSM, rather than detected. Electrical Idle inference rules are described in section 4.2.4.2 of the *PCI Express Base Specification (Rev. 2.1)*. This field is valid during normal operation, and is undefined when the PCIe Interface operates in SerDes Test Mode[a].	R/W1CS	0x0
23:20	EIOS_DET	Electrical Idle Ordered Set Detected Each bit in this field corresponds to a PCIe Interface lane. This field corresponds to lanes 3 through 0. A bit is set when the receiver on the corresponding lane detects an electrical idle condition as a direct result of having received an Electrical Idle Ordered Set. This field is valid during normal operation, and is undefined when the PCIe Interface operates in SerDes Test Mode[a].	R/W1CS	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
19:16	EIDLE_DET	Electrical Idle Detected Bit 19 = Lane 3 Bit 18 = Lane 2 Bit 17 = Lane 1 Bit 16 = Lane 0 A bit is set when the receiver on the corresponding lane detects an electrical idle, which can indicate that the link partner's transmitter has entered an idle state or that a loss of signal has been detected. This field is valid during normal operation, or when the PCIe Interface operates in SerDes Test Mode[a]. This field is not affected by the setting of the FEID and EIDD fields in the SerDes Configuration Register.	R/W1C	0x0
15:12	EIDLE_DETST	Electrical Idle Detection Status Bit 15 = Lane 3 Bit 14 = Lane 2 Bit 13 = Lane 1 Bit 12 = Lane 0 A bit is set when the receiver on the corresponding lane detects an electrical idle, which can indicate that the link partner's transmitter has entered an idle state or that a loss of signal has been detected. A bit is cleared when the receiver on the corresponding lane detects an exit from electrical idle. This field is valid during normal operation, or when the PCIe Interface operates in SerDes Test Mode[a]. This field is not affected by the setting of the FEID and EIDD fields in the SerDes Configuration Register.	R	0x0
11:8	CDR_LOCK	CDR Lock Bit 3 = Lane 3 Bit 2 = Lane 2 Bit 1 = Lane 1 Bit 0 = Lane 0 This field indicates that the SerDes Clock Data Recovery circuit has locked. This field is valid during normal operation, or when the PCIe Interface operates in SerDes Test Mode[a].	R	0x0
7	Reserved	Reserved	R	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
6	P0_READY	SerDes Ready in P0 State 1 = The SerDes is operating in the P0 power state whereby it can receive and transmit data. This state is reached when the SerDes internal PLL has locked, the SerDes internal blocks have been reset appropriately, and the SerDes is ready to receive and transmit data. Note that the SerDes must reach operational state before being used in normal mode or test mode. This field is valid during normal operation, or when the PCIe Interface operates in SerDes Test Mode[a].	R	0x0
5	PLL_LOCK	SerDes PLL Lock 1 = The SerDes PLL has achieved lock with respect to the reference clock. This field is valid during normal operation, or when the PCIe Interface operates in SerDes Test Mode[a].	R	0x0
4	CURR_SPEED	Current Speed This field indicates the current operational speed of the SerDes. It is valid during normal operation, or when the PCIe Interface operates in SerDes Test Mode[a]. 0x0 = (Gen1) 2.5 Gbps 0x1 = (Gen2) 5.0 Gbps	R	0x0
3:0	RCVD	Receiver Detected Bit 3 = Lane 3 Bit 2 = Lane 2 Bit 1 = Lane 1 Bit 0 = Lane 0 A bit is set when a receiver is detected by the transmitter on the corresponding lane. This field corresponds to lanes 3 through 0. This field is not valid when the PCIe Interface operates in SerDes Test Mode[a].	R/W1C	0x0

[a] For more information on SerDes test mode, see Loopbacks.

17.15.3 Lane Status 0 Register

Register name: LANESTS0 Reset value: 0x0000_0000	Register offset: 0x51C

Bits	7	6	5	4	3	2	1	0
31:24	Reserved							
23:16	Reserved				E8B10B			
15:08	Reserved							
07:00	Reserved				PDE			

Bits	Name	Description	Type	Reset Value
31:20	Reserved	Reserved	R	0x0
19:16	E8B10B	8b/10b Error Bit 19 = Lane 3 Bit 18 = Lane 2 Bit 17 = Lane 1 Bit 16 = Lane 0 A bit is set when an 8b/10b decode error is detected in the received data stream. A bit can only be set when the LTSSM is in the L0, Configuration, Disabled, or Hot Reset states.	R/W1C	0x0
15:4	Reserved	Reserved	R	0x0
3:0	PDE	PHY Disparity Error Bit 3 = Lane 3 Bit 2 = Lane 2 Bit 1 = Lane 1 Bit 0 = Lane 0 A bit is set when an 8b/10b coding violation causes a running disparity error in the received data stream. A bit can only be set when the LTSSM is in the L0 state.	R/W1C	0x0

17.15.4 Lane Status 1 Register

Register name: LANESTS1 Reset value: 0x0000_0000	Register offset: 0x520

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

Bits	Name	Description	Type	Reset Value
31:20	Reserved	Reserved	R	0x0
19:16	OVR	Receiver Overflow Detected Bit 19 = Lane 3 Bit 18 = Lane 2 Bit 17 = Lane 1 Bit 16 = Lane 0 A bit is set when the corresponding link receiver is unable to compensate for clock variance between link partners and has dropped one or more bytes. A bit can only be set when the LTSSM is in the L0 state.	R/W1C	0x0
15:4	Reserved	Reserved	R	0x0
3:0	UND	Receiver Underflow Detected Bit 3 = Lane 3 Bit 2 = Lane 2 Bit 1 = Lane 1 Bit 0 = Lane 0 A bit is set when the corresponding link receiver is unable to compensate for clock variance between link partners and has inserted one or more zero bytes into the stream. A bit can only be set when the LTSSM is in the L0 state.	R/W1CS	0x0

17.15.5 Lane Status 2 Register

Register name: LANESTS2 Reset value: 0x0000_0000	Register offset: 0x524

Bits	7	6	5	4	3	2	1	0
31:24	Reserved							
23:16	Reserved							
15:08	L3DAP				L2DAP			
07:00	L1DAP				L0DAP			

Bits	Name	Description	Type	Reset Value
31:16	RESERVED	Reserved	R	0x0
15:12	L3DAP	Lane 3 Deskew Aligner Pointer This field contains the current deskew aligner pointer for lane 3.	R	0x0
11:8	L2DAP	Lane 2 Deskew Aligner Pointer This field contains the current deskew aligner pointer for lane 2.	R	0x0
7:4	L1DAP	Lane 1 Deskew Aligner Pointer This field contains the current deskew aligner pointer for lane 1.	R	0x0
3:0	L0DAP	Lane 0 Deskew Aligner Pointer This field contains the current deskew aligner pointer for lane 0.	R	0x0

17.15.6 PHY State Machine Timing Configuration 0 Register

Register name: PHYFSMT0 Reset value: 0x6405_9010	Register offset: 0x528

Bits	7	6	5	4	3	2	1	0
31:24	EIRXPDG2						EITXPDG2	
23:16	EITXPDG2		RXDETDELAY				EIRXPDG1	
15:08	EIRXPDG1				EITXPDG1			
07:00	Reserved		REL		USCD		SSCD	

Bits	Name	Description	Type	Reset Value
31:26	EIRXPDG2	Electrical Idle RX Pipeline Delay in Gen2 When the PHY operates in Gen2 mode, this field contains the delay (in 250-MHz clock cycles) from detection of a Hi-Z condition on a lane until the LTSSM transitions into the Recovery state. This time accounts for the delay in the receive pipeline and ensures that Electrical Idle Ordered-Sets sent by the link partner are not preempted by the Hi-Z condition.	R/WS	0x19
25:22	EITXPDG2	Electrical Idle TX Pipeline Delay in Gen2 When the PHY operates in Gen2 mode, this field contains the delay (in 250-MHz clock cycles) from the transmission of the Electrical Idle ordered set to the assertion of the electrical idle control signal to the SerDes. A value of zero corresponds to no delay; that is, the control signal is asserted following the last IDL symbol of the last EIOS send to the PCS, a value of one corresponds to a one clock cycle delay, and so on.	R/WS	0x0
21:18	RXDETDELAY	Receiver-Detect Entry Delay This field controls the delay the LTSSM inserts when entering the Detect.Quiet state from any other LTSSM state. 0x0 = No delay 0x1 = Insert a delay of 1 ms when entering Detect.Quiet from another state, and remain in this state for at least 1 ms. All other values are Reserved.	R/WS	0x1

(Continued)

Bits	Name	Description	Type	Reset Value
17:12	EIRXPDG1	Electrical Idle RX Pipeline Delay in Gen1 When the PHY operates in Gen1 mode, this field contains the delay (in 250-MHz clock cycles) from detection of a Hi-Z condition on a lane until the LTSSM transitions into the Recovery state. This time accounts for the delay in the receive pipeline and ensures that Electrical Idle Ordered-Sets sent by the link partner are not preempted by the Hi-Z condition.	R/WS	0x19
11:8	EITXPDG1	Electrical Idle TX Pipeline Delay in Gen1 When the PHY operates in Gen1 mode, this field contains the delay (in 250-MHz clock cycles) from the transmission of the Electrical Idle ordered set to the assertion of the electrical idle control signal to the SerDes. A value of zero corresponds to no delay; that is, the control signal is asserted following the last IDL symbol of the last EIOS send to the PCS, a value of one corresponds to a one clock cycle delay, and so on.	R/WS	0x0
7:6	Reserved	Reserved	R	0x0
5:4	REL	Recovery Entry Latency This field selects the delay that the LTSSM waits from the time it detects a condition to enter Recovery to the time it enters the Recovery.RcvrLock state. 0x0 = No delay 0x1 = 512 ns 0x2 = 1 us 0x3 = 10 us This delay ensures that the data path is gracefully handled to avoid problems such as an underflow when the transmit side on cut-through TLPs whose egress link bandwidth is increased using LTSSM entry to Recovery. Note: Setting this field to 0x0 can result in the occurrence of the above problems.	R/WS	0x1

(Continued)

Bits	Name	Description	Type	Reset Value
3:2	USCD	Unsuccessful Speed Change Delay This field controls the amount of time that the LTSSM stays in the Recovery.Speed state during a data rate change (that is, from Gen1 to Gen2). An unsuccessful data rate upgrade can occur when the LTSSM fails to achieve symbol lock in the Recovery.RcvrLock state at the higher speed, and reverts back to the lower speed using the Recovery.Speed state. 0x0 = 6 us 0x1 = 10 us 0x2 = 100 us 0x3 = 1 ms	R/WS	0x0
1:0	SSCD	Successful Speed Change Delay This field controls the amount of time that the LTSSM stays in the Recovery.Speed state during a data rate change (that is, from Gen2 to Gen1). 0x0 = 800 ns 0x1 = 10 us 0x2 = 100 us 0x3 = 1 ms	R/WS	0x0

17.15.7 PHY State Machine Timing Configuration 1 Register

Register name: PHYFSMT1 Reset value: 0x04404_054F	Register offset: 0x52C

Bits	7	6	5	4	3	2	1	0
31:24	EIES_FTS				Reserved	NFTSCC		
23:16	NFTSCC					NFTSNCC		
15:08	NFTSNCC					SOSIP		
07:00	SOSIP							

Bits	Name	Description	Type	Reset Value
31:28	EIES_FTS	Electrical Idle Exit Symbols before FTS On exit from electrical idle while operating in Gen2 mode, this field controls the number of K28.7 (EIE) symbols transmitted before transmission of the first fast training set (FTS). For more information, see Section 4.2.4.5 of the *PCI Express Specification (Rev. 2.1)*. This field must be programmed to a value of 4, 6, or 8. Programming this field to another value produces undefined consequences. This field has no effect when the link operates in Gen1 speed.	R/WS	0x4
27	Reserved	Reserved	R	0x0
26:19	NFTSCC	N_FTS Value with a Common Clock This field contains the number of Fast Training Sequence Ordered-Sets (NFTS) required for the receiver to lock to an incoming data stream before exiting the L0s state when the PCIe Interface and its link partner operate with a common reference clock (that is, per the setting of the CCLK field in the PCIe Link Control Register). The default value specified is the minimum number required by the SerDes receiver to achieve symbol lock. Thus, programming this field to a lower value can lead to unstable operation of the SerDes receiver. The value in this field corresponds to the N_FTS when operating at 2.5 Gbps. When operating at 5.0 Gbps, the N_FTS is the value in this field multiplied by two, with a saturating value of 0xFF. When modified, this value takes effect the next time link training occurs.	R/WS	0x80

(Continued)

Bits	Name	Description	Type	Reset Value
18:11	NFTSNCC	N_FTS Value with a Non-Common Clock This field contains the number of Fast Training Sequence Ordered-Sets (NFTS) required for the receiver to lock to an incoming data stream before exiting the L0s state, when the PCIe Interface and its link partner operate in a non-common clock configuration (that is, per the setting of the CCLK field in the PCIe Link Control Register). The default value specified is the minimum number required by the SerDes receiver to achieve symbol lock. Thus, programming this field to a lower value can lead to unstable operation of the SerDes receiver. The value in this field corresponds to the N_FTS when operating at 2.5 Gbps. When operating at 5.0 Gbps, the N_FTS is the value in this field multiplied by two, with a saturating value of 0xFF. When modified, this value takes effect the next time link training occurs.	R/WS	0x80
10:0	SOSIP	Skip Ordered Set Insertion Period This field selects the minimum number of symbol times between transmissions of skip ordered sets. The period between skip ordered sets is equal to the value in this field plus one. Note that symbol time is a function of the data rate on the link: 2.5 Gbps = 400 ps symbol time 5.0 Gbps = 200 ps symbol time	R/WS	0x54F

17.15.8 PHY Link Configuration 0 Register

Register name: PHYLCFG0 Reset value: Undefined	Register offset: 0x530

Bits	7	6	5	4	3	2	1	0
31:24	Reserved		RDETECT		Reserved			
23:16	FLANEREV	SLANEREV	TLW			ECFGAREC	Reserved	
15:08	Reserved	ILSCC	SCLINKEN	PCEC	CLINKDIS	SRMBLDIS	Reserved	G1CME
07:00	LNKNUM							

Bits	Name	Description	Type	Reset Value
31:30	Reserved	Reserved	R	0x0
29:28	RDETECT	Receiver Detect Control This field controls the way a receiver is detected in the Detect.Active LTSSM state. 0x0 = Receiver detection is implemented as described in section 4.3.5.7 in the *PCI Express Base Specification (Rev. 2.1)*. 0x1 = Reserved 0x2 = This mode informs the LTSSM that a receiver has been detected on all lanes up to the maximum link width (that is, MAXLNKWDTH field in the PCIe Link Capabilities Register). 0x3 = This mode informs the LTSSM that a receiver has not been detected on any lanes.	R/WS	0x0
27:24	Reserved	Reserved	R	0x0
23	FLANEREV	Force Lane Reversal When the SLANEREV bit is set, this field controls lane reversal on the link. 0x0 = No lane reversal 0x1 = Lanes are reversed	R/WS	0x0
22	SLANEREV	Static Lane Reversal 1 = Lane reversal is statically enabled when selected by the FLANEREV bit and is not negotiated or modified during link training.	R/WS	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
21:19	TLW	Target Link Width This field indicates the target link width of the link. 0x3 = Target Link Width = Maximum Link Width (MAXLNKWDTH in the PCIe Link Capabilities Register) - 1 Others = Reserved	R	0x3
18	ECFGAREC	Enter Configuration State After Recovery 1 = The PHY LTSSM enters the Configuration state after the Recovery state. This forces link width and lane negotiation between the LTSSM and the link partner. This bit is for debug purposes only.	R/WS	0x0
17:15	Reserved	Reserved	R	0x0
14	ILSCC	Initial Link Speed Change Control This field determines whether the PCIe Interface automatically initiates a speed change to Gen2 speed, if Gen2 speed is permissible, after initial entry to L0 from Detect. 0x0 = Automatically initiate speed change to Gen2 speed, if permissible, after the first entry to L0 from Detect. 0x1 = Do not automatically initiate a speed change to Gen2 speed, stay in Gen1 speed.	R/WS	0x1
13	SCLINKEN	Self Crosslink Enable 1 = Crosslink training of the PCIe Interface to itself is enabled (that is, the serial transmit lines of the PCIe Interface may be connected to the serial receive lines). This bit has no effect when the CLINKDIS bit is set to 0x1.	R/WS	0x0
12	PCEC	Polling Compliance Entry Condition This field controls the entry condition into the Polling.Compliance state from the Polling.Active state when “less than a predetermined number of lanes that detected a receiver have detected a break in Electrical Idle”. 0x0 = LTSSM enters polling.compliance when all lanes that detected a receiver have not exited Electrical Idle (that is, predetermined number of lanes is equal to one). 0x1 = LTSSM enters polling.compliance when at least one lane that detected a receiver has not exited Electrical Idle (that is, as in PCIe 1.1, the predetermined number of lanes is equal to the total number of lanes that detected a Receiver). For more information, see Section 4.2.6.2 of the *PCI Express Specification (Rev. 2.1)*.	R/WS	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
11	CLINKDIS	Disable Crosslink 1 = Crosslink link training is disabled and the device link trains as though crosslink were not implemented.	R/WS	0x0
10	SRMBLDIS	Disable Scrambler 1 = The scrambling function on all lanes of the link is disabled. Note that scrambling is enabled during link training.	R/WS	0x0
9	Reserved	Reserved	R	0x0
8	G1CME	Gen1 Compatibility Mode Enable 1 = Set the PHY to operate in Gen1 Compatibility mode. In this mode, the PHY does not set training set bits not defined in the *PCIe 1.1 Specification*.	R/WS	Undefined
7:0	LNKNUM	Assigned Link Number This field contains the link number assigned to the PCIe Interface. The default value of this field corresponds to the port number with which the link is associated (that is, the value in the PORTNUM field in the PCIe Link Capabilities Register).	R/WS	Undefined

17.15.9 PHY Link Configuration 1 Register

Register name: PHYLCFG1 Reset value: 0x02020_0000	Register offset: 0x534

Bits	7	6	5	4	3	2	1	0
31:24	Reserved	TX_FULL_SKP	L0S_RXSKP		L0S_RXEIDL		TXEIDL	
23:16	NFTS_TOC							
15:08	Reserved							LNPOLOREN
07:00	Reserved				LNPOLOR			

Bits	Name	Description	Type	Reset Value
31	Reserved	Reserved	R	0x0
30	TX_FULL_SKP	Transmit Full Skip Ordered Set 1 = Full skip ordered set are sent to the transmit side elastic buffer (that is, the use of compressed skips is disabled). 0 = Compressed skip are sent to the transmit-side elastic buffer. Each compressed skip is then expanded in the elastic buffer, to eliminate clock differences in non-common clock mode. This field is for debug purposes only.	R/WS	0x0
29:28	L0S_RXSKP	L0s Receiver SKP Detection Control This field controls the amount of time that the PHY receiver (LTSSM and data-path) waits before starting detection of SKP ordered-sets in the Rx_L0s.FTS state. This value is specified in units of FTS (that is, 1 FTS = 4 symbol times). A value of zero FTS implies that the PHY starts detection of SKP ordered-sets immediately after entering the Rx_L0s.FTS state. 0x0 = Zero FTS 0x1 = (N_FTS / 4) 0x2 = (N_FTS / 2) 0x3 = (N_FTS)	R/WS	0x2

(Continued)

Bits	Name	Description	Type	Reset Value
27:26	L0S_RXEIDL	L0s Rx-Idle-Min Control This field controls the amount of time that the PHY receiver remains in the Rx_L0s.Entry state (that is, the amount of time before the PHY receiver looks for the occurrence of electrical-idle break on the line). 0x0 = 20 ns 0x1 = 40 ns 0x2 = 80 ns 0x3 = 160 ns Note that modifying this value can result in the loss of an initial number of received FTS during L0s exit (that is, when the link partner starts transmitting FTS but the PHY receiver remains in the Rx_L0s.Entry state).	R/WS	0x0
25:24	TXEIDL	L0s/L1 Tx-Idle-Min Control This field controls the amount of time that the PHY transmitter remains in electrical idle during the L0s and L1 states (that is, the amount of time spent in Tx_L0s.Entry and L1.Entry). 0x0 = 20 ns 0x1 = 100 ns 0x2 = 500 ns 0x3 = 1 us	R/WS	0x0
23:16	NFTS_TOC	N_FTS Timeout Control This field selects the value of the N_FTS timeout value when the PHY receiver exits L0s. For more information on N_FTS timeout, see Section 4.2.6.6.1.3 of the *PCI Express Specification (Rev. 2.1)*. The N_FTS timeout is derived as: Timeout = 40*[NFTS + NFTS_TOC]*UI NFTS refers to the N_FTS value advertised by the PCIe Interface (that is, per the setting of the NFTSCC and NFTSNCC fields in the PHY State Machine Timing Configuration 1 Register). NFTS_TOC refers to the value in this field. UI refers to the unit interval on the link (that is, 400 ps in Gen1, 200 ps in Gen2). This field should not be programmed to a value less than 0x3.	R/WS	0x20
15:9	Reserved	Reserved	R	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
8	LNPOLOREN	Lane Polarity Override Enable 0x0 = Lane polarity is controlled automatically by LTSSM 0x1 = Lane polarity of each SerDes lane is controlled by the LNPOLOR field Updates to this field take effect the next time the link is fully retrained (that is, LTSSM transition to Detect state).	R/WS	0x0
7:4	Reserved	Reserved	R	0x0
3:0	LNPOLOR	Lane Polarity Override Bit 3 = Lane 3 Bit 2 = Lane 2 Bit 1 = Lane 1 Bit 0 = Lane 0 When a bit in this field is set, the polarity of the corresponding lane is inverted. Updates to this field take effect the next time the link is fully retrained (that is, LTSSM transition to Detect state).	R/WS	0x0

17.15.10 PHY Link Status 0 Register

Register name: PHYLSTS0 Reset value: 0x0000_0000	Register offset: 0x538

Bits	7	6	5	4	3	2	1	0
31:24	LNPOLORSTS[7:0]							
23:16	Reserved							LPWUC
15:08	ILW		Reserved		L0SREC	RECCON	RECDET	Reserved
07:00	Reserved	SOPLERR	SOPEOPERR	DSOPERR	EOPPERR	SEOPERR	PADERR	STPSDP

Bits	Name	Description	Type	Reset Value
31:24	LNPOLORSTS[7:0]	Lane Polarity Status Each bit in this field corresponds to a lane associated with the PCIe Interface as follows: • Bit 7–4 = Reserved • Bit 3 = Lane 3 • Bit 2 = Lane 2 • Bit 1 = Lane 1 • Bit 0 = Lane 0 A bit in this field is set when inverted polarity has been detected on the corresponding lane. 0b0 = Normal polarity 0b1 = Inverted polarity The default value corresponds to the value before link training, and can change as a result of link training.	R	0x0
23:17	Reserved	Reserved	R	0x0
16	LPWUC	Link Partner Width Upconfiguration Capability This bit indicates a link partner’s ability to upconfigure link widths. This bit indicates the state of the upconfigure_capable variable defined by the *PCI Express Specification (Rev. 2.1)*.	R	0x0
15:14	ILW	Initial Link Width This field indicates the initial negotiated link width. This is defined as the initial link width on the first entry to L0 after Detect. 0x0 = LInk width = x1 0x1 = Link width = x2 0x2 = Link width = x4 Others = Reserved	R	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
13:12	Reserved	Reserved	R	0x0
11	L0SREC	L0s to Recovery 1 = The LTSSM transitioned from a L0s to the Recovery state. This occurs when the N_FTS requirement is not met by the link partner.	R/W1C	0x0
10	RECCON	Recovery to Configuration 1 = The LTSSM transitioned from a Recovery to the Configuration state.	R/W1C	0x0
9	RECDET	Recovery to Detect 1 = The LTSSM transitioned from a Recovery to the Detect state.	R/W1C	0x0
8:7	Reserved	Reserved	R	0x0
6	SOPLERR	SOP Lane Error 1 = A packet is received with an incorrectly placed SOP. This error never occurs in x1 link configurations. It can only be set when the LTSSM is in the L0 state.	R/W1C	0x0
5	SOPEOPERR	SOP and no EOP Error 1 = Two SOPs are received with no intervening EOP. It can only be set when the LTSSM is in the L0 state.	R/W1C	0x0
4	DSOPERR	Double SOP Error 1 = Two SDP or two STP symbols are received in the same symbol time. It can only be set when the LTSSM is in the L0 state.	R/W1C	0x0
3	EOPPERR	EOP Placement Error 1 = An EOP is received in a lane other than the (4N-1) location. It can only be set when the LTSSM is in the L0 state.	R/W1C	0x0
2	SEOPERR	Stray EOP Error 1 = An EOP is received outside of a packet boundary. It can only be set when the LTSSM is in the L0 state.	R/W1C	0x0
1	PADERR	PAD Error 1 = An EOP is received in lane K, where K is not N-1, and all the remaining lanes from K+1 to N-1 did not contain PAD. It can only be set when the LTSSM is in the L0 state.	R/W1C	0x0
0	STPSDP	STP and SDP Within a Symbol Time Error 1 = An STP and SDP symbol are seen in the same symbol time. It can only be set when the LTSSM is in the L0 state.	R/W1C	0x0

17.15.11 PHY Link Status 1 Register

Register name: PHYLSTS1 Reset value: 0x0000_0000	Register offset: 0x53C

Bits	7	6	5	4	3	2	1	0
31:24	Reserved							
23:16	Reserved			RX8B10BERR	RXOFERR	RXUFERR	RXDISPERR	RXFRERR
15:08	Reserved							
07:00	Reserved	RSKPOS	RTS1OS	RTS2OS	RIDLOS	RFTSOS	TRAINDE	Reserved

Bits	Name	Description	Type	Reset Value
31:21	Reserved	Reserved	R	0x0
20	RX8B10BERR	Receiver 8b/10b Error 1 = A PHY receiver error is caused by an 8b/10b error in the PCIe Interface.	R/W1CS	0x0
19	RXOFERR	Receiver Overflow Error 1 = A PHY receiver error is caused by an elastic buffer overflow error in the PCIe Interface.	R/W1CS	0x0
18	RXUFERR	Receiver Underflow Error 1 = A PHY receiver error is caused by an elastic buffer underflow error in the PCIe Interface.	R/W1CS	0x0
17	RXDISPERR	Receiver Disparity Error 1 = A PHY receiver error is caused by a disparity error in the PCIe Interface.	R/W1CS	0x0
16	RXFRERR	Receiver Framing Error 1 = A PHY receiver error is caused by a framing error in the PCIe Interface.	R/W1CS	0x0
15:7	Reserved	Reserved	R	0x0
6	RSKPOS	PHY SKP Ordered-Set 1 = The lane specified by the LANESEL field in the PHY Countables Configuration Register received a SKP ordered set.	R/W1C	0x0
5	RTS1OS	PHY Received TS1 Ordered-Set 1 = The lane specified by the LANESEL field in the PHY Countables Configuration Register received a TS1 ordered set.	R/W1C	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
4	RTS2OS	PHY Received TS2 Ordered-Set 1 = The lane specified by the LANESEL field in the PHY Countables Configuration Register received a TS2 ordered set.	R/W1C	0x0
3	RIDLOS	PHY Received Idle Ordered-Set 1 = The lane specified by the LANESEL field in the PHY Countables Configuration Register received a IDL ordered set.	R/W1C	0x0
2	RFTSOS	PHY Received FTS Ordered-Set 1 = The lane specified by the LANESEL field in the PHY Countables Configuration Register received a FTS ordered set.	R/W1C	0x0
1	TRAINDE	PHY Training Sequence Decode Error 1 = The lane specified by the LANESEL field in the PHY Countables Configuration Register received a malformed training sequence ordered set (TS1 or TS2). Checking is enabled when 2 consecutive good TS1 or TS2 are received, and disabled when electrical idle is detected or EIOS is received.	R/W1C	0x0
0	Reserved	Reserved	R	0x0

17.15.12 PHY Link State 0 Register

Register name: PHYLSTATE0 Reset value: 0x0000_0000	Register offset: 0x540

Bits	7	6	5	4	3	2	1	0
31:24	FLRET	Reserved	LANEREV	LNKNUM				
23:16	LNKNUM				LPNFTS			
15:08	LPNFTS				SRMBLSTAT	RXLSTATE		
07:00	TXLSTATE			LTSSMSTATE				

Bits	Name	Description	Type	Reset Value
31	FLRET	Full Link Retrain 1 = Initiate full link retraining by directing the PHY LTSSM into the DETECT state. This bit returns zero when read. This also causes the Tsi721 to return a completion to the requester before the action specified by this bit takes effect.	R/W	0x0
30	Reserved	Reserved	R	0x0
29	LANEREV	Lane Reversed 1 = The PCIe Interface has reversed its lanes to support lane reversal.	R	0x0
28:20	LNKNUM	Configured Link Number This field indicates the negotiated link number and is only valid after successful link training. 0x1FF = No link training has been attempted since the LTSSM entered the Detect state. 0x1F7 = Invalid link training 0x000–0x0FF = Valid link numbers	R	0x0
19:12	LPNFTS	Link Partner N_FTS Value This field contains the N_FTS value received from the link partner. The value represents the required number of Fast Training Sequence Ordered-Sets that the PCIe Interface needs to transmit in order for the link partner to obtain receiver lock before exiting L0s.	R	0x0
11	SRMBLSTAT	Scrambler Status 0x0 = Scrambler is enabled 0x1 = Scrambler is disabled	R	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
10:8	RXLSTATE	PHY Receive L-State This field indicates the current state of the PHY receive L-State state machine. 0x0 = RX_L0 0x1 = RX_L0S_ENTRY 0x2 = RX_L0S_IDLE 0x3 = RX_L0S_FTS 0x4 = RX_L1_ENTRY 0x5 = RX_L1_IDLE 0x6–0x7 = Reserved	R	0x0
7:5	TXLSTATE	PHY Transmit L-State This field indicates the current state of the PHY transmit L-State state machine. 0x0 = TX_L0 0x1 = TX_L0S_ENTRY 0x2 = TX_L0S_IDLE 0x3 = TX_L0S_FTS 0x4 = TX_L1_ENTRY 0x5 = TX_L1_IDLE 0x6–0x7 = Reserved	R	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
4:0	LTSSMSTATE	PHY LTSSM State Machine State This field contains the current state of the PHY Link Training and Status State Machine (LTSSM). 0x0 = XMIT_EIOS 0x1 = TMOUT_1MS 0x2 = DET_QUIET 0x3 = DET_ACTIVE 0x4 = POL_ACTIVE 0x5 = POL_COMPLIANCE 0x6 = POL_CONFIG 0x7 = RESERVE_1 0x8 = CFG_LWIDTH_START 0x9 = CFG_LWIDTH_ACCEPT 0xA = CFG_LNUM_WAIT 0xB = CFG_LNUM_ACCEPT 0xC = CFG_COMPLETE 0xD = CFG_IDLE 0xE = RESERVE_2 0xF = OVR_TMOUT 0x10 = REC_RCVR_LOCK 0x11 = REC_RCVR_CFG 0x12 = REC_IDLE 0x13 = REC_SPEED 0x14 = L0 0x15 = L0s 0x16 = L1_ENTRY 0x17 = L1_IDLE 0x18 = L2_IDLE 0x19 = L2_XMIT_WAKE 0x1A = DISABLE 0x1B = HOT_RST 0x1C = LPBK_ENTRY 0x1D = LPBK_ACTIVE 0x1E = LPBK_EXIT 0x1F = IDT_TM*	R	0x0

17.15.13 PHY Link LTSSM Status 0 Register

Register name: PHYLTSSMSTS0 Reset value: 0x0000_0000	Register offset: 0x544

Bits	7	6	5	4	3	2	1	0
31:24	IDT_TM	LBEXIT	LBACTIVE	LBENTRY	HOTRESET	DISABLED	L2XMITWAKE	L2IDLE
23:16	L1IDLE	L1ENTRY	L0S	L0	RECSPEED	RECIDLE	RECRCVCFG	RECRCVLOCK
15:08	TOUPCFG	Reserved	CIDLE	CCOMPLETE	CLNACCEPT	CLNWAIT	CLWACCEPT	CLWSTART
07:00	Reserved	PCONFIG	PCOMP	PACTIVE	DACTIVE	DQUIET	TMOUT_1MS	XMIT_EIOS

Bits	Name	Description	Type	Reset Value
31	IDT_TM	IDT Test Mode 1 = The LTSSM entered or is in the IDT Test Mode state.	R/W1CS	0x0
30	LBEXIT	Loopback Exit 1 = The LTSSM entered or is in the Loopback Exit state.	R/W1CS	0x0
29	LBACTIVE	Loopback Active 1 = The LTSSM entered or is in the Loopback Active state.	R/W1CS	0x0
28	LBENTRY	Loopback Entry 1 = The LTSSM entered or is in the Loopback Entry state.	R/W1CS	0x0
27	HOTRESET	Hot Reset 1 = The LTSSM entered or is in the Hot Reset state.	R/W1CS	0x0
26	DISABLED	Disabled 1 = the LTSSM entered or is in the Disabled state.	R/W1CS	0x0
25	L2XMITWAKE	L2 Transmit Wake 1 = The LTSSM entered or is in the L2.TransmitWake state.	R/W1CS	0x0
24	L2IDLE	L2 Idle 1 = The LTSSM entered or is in the L2.Idle state.	R/W1CS	0x0
23	L1IDLE	L1 Idle 1 = The LTSSM entered or is in the L1.Idle state.	R/W1CS	0x0
22	L1ENTRY	L1 Entry 1 = The LTSSM entered or is in the L1.Entry state.	R/W1CS	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
21	L0S	L0s 1 = The LTSSM entered or is in the L0s state.	R/W1CS	0x0
20	L0	L0 1 = The LTSSM entered or is in the L0 state.	R/W1CS	0x0
19	RECSPEED	Recovery Speed 1 = The LTSSM entered or is in the Recovery.Speed state.	R/W1CS	0x0
18	RECIDLE	Recovery Idle 1 = The LTSSM entered or is in the Recovery.Idle state.	R/W1CS	0x0
17	RECRCVCFG	Recovery Receiver Configuration 1 = The LTSSM entered or is in the Recovery.RcvCfg state.	R/W1CS	0x0
16	RECRCVLOCK	Recovery Receiver Lock 1 = The LTSSM entered or is in the Recovery.RcvrLock state.	R/W1CS	0x0
15	TOUPCFG	Timeout Up Configuration 1 = The LTSSM entered or is in the Timeout Up configuration state.	R/W1CS	0x0
14	Reserved	Reserved	R	0x0
13	CIDLE	Configuration Idle 1 = The LTSSM entered or is in the Configuraiton.Idle state.	R/W1CS	0x0
12	CCOMPLETE	Configuration Complete 1 = The LTSSM entered or is in the Configuraiton.Complete state.	R/W1CS	0x0
11	CLNACCEPT	Configuration Lanenum Accept 1 = The LTSSM entered or is in the Configuraiton.Lanenum.Accept state.	R/W1CS	0x0
10	CLNWAIT	Configuration Lane num Wait 1 = The LTSSM entered or is in the Configuraiton.Lanenum.Wait state.	R/W1CS	0x0
9	CLWACCEPT	Configuration Link width Accept 1 = The LTSSM entered or is in the Configuration.Linkwidth.Accept state.	R/W1CS	0x0
8	CLWSTART	Configuration Link width Start 1 = The LTSSM entered or is in the Configuration.Linkwidth.Start state.	R/W1CS	0x0
7	Reserved	Reserved	R	0x0
6	PCONFIG	Polling Configuration 1 = The LTSSM entered or is in the Polling.Configuration state.	R/W1CS	0x0

This document is confidential and is subject to an NDA.

(Continued)

Bits	Name	Description	Type	Reset Value
5	PCOMP	Polling Compliance 1 = The LTSSM entered or is in the Polling.Compliance state.	R/W1CS	0x0
4	PACTIVE	Polling Active 1 = The LTSSM entered or is in the Polling.Active state.	R/W1CS	0x0
3	DACTIVE	Detect Active 1 = The LTSSM entered or is in the Detected.Active state.	R/W1CS	0x0
2	DQUIET	Detect Quiet 1 = The LTSSM entered or is in the Detected.Quiet state.	R/W1CS	0x0
1	TMOUT_1MS	Timeout 1 ms 1 = The LTSSM entered or is in the TMOUT_1MS state.	R/W1CS	0x0
0	XMIT_EIOS	Transmit Electrical Idle Ordered Set 1 = The LTSSM entered or is in the XMIT_EIOS state.	R/W1CS	0x0

17.15.14 PHY Link LTSSM Status 1 Register

Register name: PHYLTSSMSTS1 Reset value: 0x0000_0000	Register offset: 0x548

Bits	7	6	5	4	3	2	1	0
31:24	Reserved							
23:16	Reserved							
15:08	Reserved							
07:00	RXL0SFTS	RXL0SIDLE	RXL0SENTRY	RXACTIVE	TXL0SFTS	TXL0SIDLE	TXL0SENTRY	TXACTIVE

Bits	Name	Description	Type	Reset Value
31:8	Reserved	Reserved	R	0x0
7	RXL0SFTS	RX FSM L0s FTS State 1 = The receive state machine has entered the L0s FTS state.	R/W1CS	0x0
6	RXL0SIDLE	RX FSM L0s Idle State 1 = The receive state machine has entered the L0s Idle state.	R/W1CS	0x0
5	RXL0SENTRY	RX FSM L0s Entry State 1 = The receive state machine has entered the L0s Entry state.	R/W1CS	0x0
4	RXACTIVE	RX FSM Active State 1 = The receive state machine has entered the Active state.	R/W1CS	0x0
3	TXL0SFTS	TX FSM L0s FTS State 1 = The transmit state machine has entered the L0s FTS state.	R/W1CS	0x0
2	TXL0SIDLE	TX FSM L0s Idle State 1 = The transmit state machine has entered the L0s Idle state.	R/W1CS	0x0
1	TXL0SENTRY	TX FSM L0s Entry State 1 = The transmit state machine has entered the L0s Entry state.	R/W1CS	0x0
0	TXACTIVE	TX FSM Active State 1 = The transmit state machine has entered the Active state.	R/W1CS	0x0

17.15.15 PHY Countables 0 Register

Register name: PHYCNT0 Reset value: 0x0000_0000	Register offset: 0x54C

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

Bits	Name	Description	Type	Reset Value
31:0	COUNT	Count This field counts the number of occurrences of the event specified in the PHYCNT0SEL field in the PHY Countables Configuration Register (for COUNT values, see Table 80). This counter saturates at its maximum value. Note that the occurrence of an error depends on the LTSSM state. Specifically, disparity, overflow/underflow, and framing errors (that is, 0x47->0x4c) are only counted in the L0 state. Training sequence errors (0x40 and 0x41) are counted in the Configuration, L0, Disabled, and Hot Reset states.This field is cleared when read.	RCWS	0x0

Table 80: COUNT Value in PHYCNT0 Register

COUNT Value	Event
0x10	PHY Disparity Error Lane 0
0x11	PHY Disparity Error Lane 1
0x12	PHY Disparity Error Lane 2
0x13	PHY Disparity Error Lane 3
0x20	PHY Clock Comp Overflow Lane 0
0x21	PHY Clock Comp Overflow Lane 1
0x22	PHY Clock Comp Overflow Lane 2
0x23	PHY Clock Comp Overflow Lane 3
0x30	PHY Clock Comp Underflow Lane 0
0x31	PHY Clock Comp Underflow Lane 1
0x32	PHY Clock Comp Underflow Lane 2
0x33	PHY Clock Comp Underflow Lane 3

Table 80: COUNT Value in PHYCNT0 Register *(Continued)*

COUNT Value	Event
0x40	Received Training Sequence Decode Error Lane X
0x41	Received Training Sequence Data Error Lane X
0x42	Received SKP-os Lane X
0x43	Received TS1-os Lane X
0x44	Received TS2-os Lane X
0x45	Received IDL-os Lane X
0x46	Received FTSos Lane X
0x47	STP and SDP Within a Symbol Time Error
0x48	PAD Error
0x49	EOP Stray Error
0x4a	SOP Continuation Error
0x4b	SOP Double Error
0x4c	SOP Lane Error
0x4d	Link Tx Error
0x4e	Alignment Error
0x4f	Recovery Error[a]
0x50	Recovery to Configuration
0x51	L0s to Recovery
0x52	TX State L1.Idle
0x53	TX State L0s.Idle
0x54	RX State L1.Idle
0x55	RX State L0s.Idle
0x56	LTSSM State Hot Reset
0x57	LTSSM State Disabled
0x58	LTSSM State Recovery.RcvrLock
0x59	LTSSM State Configuration.Linkwidth.Start
0x5a	LTSSM State Polling.Active
0x5b	LTSSM State Detect.Quiet
0x5c	LTSSM State Recovery.Speed
Others	Reserved

[a] A Recovery Error is any transition from a Recovery sub-state to Detect.

17.15.16 PHY Countables 1 Register

Register name: PHYCNT1 Reset value: 0x0000_0000	Register offset: 0x550

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

Bits	Name	Description	Type	Reset Value
31:0	COUNT	Count This field counts the number of occurrences of the event specified in the PHYCNT1SEL field in the PHY Countables Configuration Register (for COUNT values, see Table 81). This counter saturates at its maximum value and is cleared when read.Note that the occurrence of an error depends on the LTSSM state. Specifically, disparity, overflow/underflow, and framing errors (that is, 0x47->0x4c) are only counted in the L0 state. Training sequence errors (0x40 and 0x41) are counted in the Configuration, L0, Disabled, and Hot Reset states.	RCWS	0x0

Table 81: COUNT Value in PHYCNT1 Register

COUNT Value	Event
0x10	PHY Disparity Error Lane 0
0x11	PHY Disparity Error Lane 1
0x12	PHY Disparity Error Lane 2
0x13	PHY Disparity Error Lane 3
0x20	PHY Clock Comp Overflow Lane 0
0x21	PHY Clock Comp Overflow Lane 1
0x22	PHY Clock Comp Overflow Lane 2
0x23	PHY Clock Comp Overflow Lane 3
0x30	PHY Clock Comp Underflow Lane 0
0x31	PHY Clock Comp Underflow Lane 1
0x32	PHY Clock Comp Underflow Lane 2
0x33	PHY Clock Comp Underflow Lane 3

Table 81: COUNT Value in PHYCNT1 Register *(Continued)*

COUNT Value	Event
0x40	Received Training Sequence Decode Error Lane X
0x41	Received Training Sequence Data Error Lane X
0x42	Received SKP-os Lane X
0x43	Received TS1-os Lane X
0x44	Received TS2-os Lane X
0x45	Received IDL-os Lane X
0x46	Received FTS-os Lane X
0x47	STP and SDP Within a Symbol Time Error
0x48	PAD Error
0x49	EOP Stray Error
0x4a	SOP Continuation Error
0x4b	SOP Double Error
0x4c	SOP Lane Error
0x4d	Link Tx Error
0x4e	Alignment Error
0x4f	Recovery Error[a]
0x50	Recovery to Configuration
0x51	L0s to Recovery
0x52	TX State L1.Idle
0x53	TX State L0s.Idle
0x54	RX State L1.Idle
0x55	RX State L0s.Idle
0x56	LTSSM State Hot Reset
0x57	LTSSM State Disabled
0x58	LTSSM State Recovery.RcvrLock
0x59	LTSSM State Configuration.Linkwidth.Start
0x5a	LTSSM State Polling.Active
0x5b	LTSSM State Detect.Quiet
0x5c	LTSSM State Recovery.Speed
Others	Reserved

[a] A Recovery Error is any transition from a Recovery sub-state to Detect.

17.15.17 PHY Countables Configuration Register

Register name: PHYCNTCFG Reset value: 0x0000_0000	Register offset: 0x554

Bits	7	6	5	4	3	2	1	0
31:24	Reserved							
23:16	Reserved			LANESEL				
15:08	Reserved	PHYCNT1SEL						
07:00	Reserved	PHYCNT0SEL						

Bits	Name	Description	Type	Reset Value
31:21	Reserved	Reserved	R	0x0
20:16	LANESEL	Lane Select This field selects the lane whose status is reported in the PHY Link Status 1 Register. 0x0 = Lane 0 0x1 = Lane 1 0x2 = Lane 2 0x3 = Lane 3 0x4–0xF = Reserved 0x10 = Logical OR of all lanes 0x11 = Logical AND of all lanes Others = Reserved	R/WS	0x0
15	Reserved	Reserved	R	0x0
14:8	PHYCNT1SEL	PHY Countables 1 Select This field selects the events that are counted by the COUNT field in the PHY Countables 1 Register.	R/WS	0x0
7	Reserved	Reserved	R	0x0
6:0	PHYCNT0SEL	PHY Countables 0 Select This field selects the events that are counted by the COUNT field in the PHY Countables 1 Register.	R/WS	0x0

17.15.18 PHY Recovery Entry Log Register

Register name: PHYRECEL Reset value: 0x8000_0000	Register offset: 0x558

Bits	7	6	5	4	3	2	1	0
31:24	ENLOG	Reserved			FC			
23:16	Reserved		TM	L1EXIT	CITO	HOTRST	NFTSTO	EIDLL0
15:08	TSL0	OLSC	Reserved	ILSC	DSKERR	DDL	LDIS	LRET
07:00	RCOUNT							

Bits	Name	Description	Type	Reset Value
31	ENLOG	Enable Log This bit enables logging of the Recovery entry conditions defined by other fields in this register. Setting this register to 0x0 causes all other fields in this register to be reset to 0x0.	R/WS	0x1
30:28	Reserved	Reserved	R	0x0
27:24	FC	First Entry Cause[a] This field indicates the reason behind the first entry to the Recovery state from the time the ENLOG bit is set. 0x0 = No entry detected. 0x1 = Directed to retrain link 0x2 = Directed to disable link 0x3 = Directed by data link layer 0x4 = Deskew error 0x5 = Initial link speed change 0x6 = Autonomous link reliability speed change 0x7 = Other link speed change 0x8 = Training set reception in L0 0x9 = Electrical idle detected / inferred in L0 0xA = N_FTS timeout 0xB = Hot reset 0xC = Configuration idle timeout 0xD = L1 exit 0xE = Test mode entered Others = Undefined	RCS	0x0
23:22	Reserved	Reserved	R	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
21	TM	Test mode Entered[a] 1 = The PHY entered Recovery in order to transition to the IDT proprietary “Test Mode” state.	RCS	0x0
20	L1EXIT	L1 Exit[a] 1 = The PHY entered Recovery to exit the L1 state. This is the expected state transition to exit L1.	RCS	0x0
19	CITO	Configuration Idle Timeout[a] 1 = The PHY entered Recovery due to a timeout in the Configuration.Idle sub-state.	RCS	0x0
18	HOTRST	Hot Reset[a] 1 = The PHY entered Recovery as a result of being directed to enter the hot reset state.	RCS	0x0
17	NFTSTO	N_FTS timeout[a] 1 = The PHY entered Recovery due to an N_FTS timeout condition in the Rx_L0s.FTS state.	RCS	0x0
16	EIDLL0	Electrical Idle Detected/Inferred in L0[a] 1 = The PHY entered Recovery due to the detection or inference of electrical idle on all receive lanes without having received an EIOS on any lane.	RCS	0x0
15	TSL0	Training Set Reception in L0[a] 1 = The PHY entered Recovery due to the reception of training sets (that is, TS1 or TS2) on any configured lane in L0.	RCS	0x0
14	OLSC	Other Link Speed Change[a] 1 = The PHY entered Recovery in order to perform a link speed change, which is not the initial link speed change and is not due to the Autonomous Link Reliability mechanism.	RCS	0x0
13	Reserved	Reserved	R	0x0
12	ILSC	Initial Link Speed Change[a] 1 = The PHY entered Recovery in order to perform the initial link speed change (that is, from Gen1 to Gen2) after link-training to L0 from the Detect state.	RCS	0x0
11	DSKERR	Deskew Error[a] 1 = The PHY entered Recovery due to a lane deskew error.	RCS	0x0
10	DDL	Directed by Data-Link Layer[a] 1 = The PHY entered Recovery due to being directed to this state by the data-link layer (for example, replay number rollover, lack of reception of update flow-control).	RCS	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
9	LDIS	Directed to Disable Link[a] 1 = The PHY entered Recovery due to the LDIS bit being set in the PCIe Link Control Register.	RCS	0x0
8	LRET	Directed to Retrain Link[a] 1 = The PHY entered Recovery due to the LRET bit being set in the PCIe Link Control Register.	RCS	0x0
7:0	RCOUNT	Recovery Entry Count This field contains a count of the number of entries to Recovery from the time the ENLOG bit in this register is set. The count saturates at its maximum value (that is, 0xFF).	RCS	0x0

[a] The value of this field is cleared when the ENLOG bit is cleared.

17.15.19 PHY PRBS Seed Register

Register name: PHYPRBS Reset value: 0x0000_FFFF	Register offset: 0x55C

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

Bits	Name	Description	Type	Reset Value
31:16	Reserved	Reserved	R	0x0
15:0	SEED	PHY PRBS Seed Value This field contains the PHY PRBS seed value used for crosslink operation. When the value in this register is modified, the PRBS counter associated with this seed is reset to the seed value and restarts counting.	R/WS	0xFFFF

17.16 Data Link Layer Control and Status Registers

17.16.1 Data Link Control 1 Register

Register name: DLCTL1 Reset value: 0x0000_0000	Register offset: 0x600

Bits	7	6	5	4	3	2	1	0
31:24	ANTIMEOUTO	ANTIMEOUT						
23:16	ANTIMEOUT							
15:08	RPTIMEOUTO	RPTIMEOUT						
07:00	RPTIMEOUT							

Bits	Name	Description	Type	Reset Value
31	ANTIMEOUTO	Ack and Nack Latency Timeout Override 0 = When the bit is cleared, the data link layer auto-selects the timeout value based on link width. 1 = The data link layer uses the value in the ANTIMEOUT field as the Ack Nack latency timeout value.	R/WS	0x0
30:16	ANTIMEOUT	Ack and Nack Latency Timeout This field contains the value used for the data link layer Ack Nack latency timeout value when the ANTIMEOUTO bit is set. The timeout value is expressed in terms of 250-MHz clock periods.	R/WS	0x0
15	RPTIMEOUTO	Replay Timeout Override 0 = The data link layer auto-selects the timeout value based on link width. 1 = The data link layer uses the value in the RPTIMEOUT field as the replay timeout value.	R/WS	0x0
14:0	RPTIMEOUT	Replay Timeout This field contains the value used for the data link layer replay timeout when the RPTIMEOUTO bit is set. The timeout value is expressed in terms of 250-MHz clock periods.	R/WS	0x0

17.16.2 Data Link Control 2 Register

Register name: DLCTL2 Reset value: 0x0000_4000	Register offset: 0x604

Bits	7	6	5	4	3	2	1	0
31:24	DISCRCCHK	Reserved						
23:16	Reserved							
15:08	Reserved	INITFCTOVC	INITFCVALVC					
07:00	INITFCVALVC							

Bits	Name	Description	Type	Reset Value
31	DISCRCCHK	Disable CRC Check 1 = The data link layer CRC check is disabled.	R/WS	0x0
30:15	Reserved	Reserved	R	0x0
14	INITFCTOVC	VC Flow Control Initialization Timeout Value Override 0 = The data link layer uses a default timeout value of 1 ms. 1 = The data link layer uses the value in the INITFCVALVC field as the VC flow control initialization timeout value.	R/WS	0x1
13:0	INITFCVALVC	VC Flow Control Initialization Timeout Value This field contains the value used for the data link layer VC flow control initialization timeout value when the INITFCTOVC bit is set. The timeout value is expressed in terms of microseconds. A value of zero corresponds to no time limit. Expiration of the VC flow control initialization timer indicates that the VC flow control initialization process could not complete in the time limit contained in this field. When this occurs, the data link layer forces the physical layer to fully retrain the link; that is, the LTSSM transitions directly to the Detect state.	R/WS	0x0

17.16.3 Data Link Control 3 Register

Register name: DLCTL3 Reset value: 0x0011_24F8	Register offset: 0x608

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

Bits	Name	Description	Type	Reset Value
31:21	Reserved	Reserved	R	0x0
20	DLLPRXTE	DLLP Reception Timer Enable 1 = Enable DLLP reception timer	R/WS	0x1
19:17	Reserved	Reserved	R	0x0
16:0	DLLPRXTO	DLLP Reception Timeout This field contains the value used by the data link layer DLLP reception timer when the DLLPRXTE bit is set. The timeout value is expressed in terms of 250-MHz clock periods. The DLLP reception timer corresponds to the optional timer specified in section 2.6.1.2 of the *PCI Express Specification (Rev. 2.1)*. This timer expires when the DL does not receive a DLLP in a time interval specified by the value of this field. The default value of the timer corresponds to 300 us. The timer is active when the following conditions are satisfied: • The link state is L0 or L0s • The DLLPRXTE bit is set • At least one flow control credit type has advertised finite flow control credits. In all other states, the timer is reset and inactive. The DLLP reception timer is reset on receipt of any DLLP. When the timer times out, the data link layer forces the physical layer to retrain the link (through the Recovery state).	R/WS	0x124F8

17.16.4 Data Link Status Register

Register name: DLSTS Reset value: 0x0000_0000	Register offset: 0x60C

Bits	7	6	5	4	3	2	1	0
31:24	Reserved							
23:16	Reserved							
15:08	Reserved							DLBUFOVRFL
07:00	RXPROTERR	Reserved		RXFERR	Reserved		DLFSM	

Bits	Name	Description	Type	Reset Value
31:9	Reserved	Reserved	R	0x0
8	DLBUFOVRFL	Data Link Buffer Transmit Buffer Overflow 1 = The data link layer transmit buffer overflows (that is, the application layer writes to the buffer when the buffer is full). This bit is for debug purposes only.	R/W1C	0x0
7	RXPROTERR	Receive Protocol Error 1 = The data link layer detects a receive protocol error in which an Ack/Nack DLLP is received out of range.	R/W1C	0x0
6:5	Reserved	Reserved	R	0x0
4	RXFERR	Receive Framing Error 1 = The data link layer receives a DLLP with illegal size (that is, the DLLP's length is not 6 bytes) or a DLLP with and end-bad delimiter (without LCRC inverted).	R/W1C	0x0
3:2	Reserved	Reserved	R	0x0
1:0	DLFSM	Data Link Control and Management State Machine State This field contains the current state of the data link layer control and management state machine. 0x0 = Inactive state 0x1 = DL_Init_FC1 state 0x2 = DL_Init_FC2 state 0x3 = Active state	R	0x0

17.16.5 Data Link Receive Status Register

Register name: DLRXSTS Reset value: 0x0000_0000	Register offset: 0x610

Bits	7	6	5	4	3	2	1	0
31:24	Reserved							
23:16	Reserved		RXTLPLERR	RXUDLLP	RXVDDLLP	DLLPRXTO	RTLPNULL	Reserved
15:08	RDUPTLP	RBEDB	ROSEQ	PMDLLP	Reserved	UFCDLLP	Reserved	IFCDLLP2
07:00	Reserved	IFCDLLP1	ACKDLLP	NACKDLLP	DLLPCRCERR	DLLP	TLPCRCERR	TLP

Bits	Name	Description	Type	Reset Value
31:22	Reserved	Reserved	R	0x0
21	RXTLPLERR	Received TLP with Length Error 1 = The data link layer received an TLP that is smaller than the minimum length TLP (that is, 5 dwords).	R/W1C	0x0
20	RXUDLLP	Received Undefined DLLP 1 = The data link layer received an undefined DLLP.	R/W1C	0x0
19	RXVDDLLP	Received Vendor Defined DLLP 1 = The data link layer received a vendor defined DLLP.	R/W1C	0x0
18	DLLPRXTO	DLLP Reception Timer Timeout 1 = The DLLP reception timer expired and the physical layer is directed to retrain the link (see Data Link Control 3 Register).	R/W1C	0x0
17	RTLPNULL	Received TLP Nullified 1 = A received TLP is nullified by the data link layer. The DL layer nullifies received TLPs when the physical layer has detected an error in the TLP, when the DL detects an LCRC error in the TLP, when the DL detects a sequence number error in the TLP, when the received TLP ends with an EDB symbol (that is, end-bad), or when the DL receives one or more TLP(s) whose sequence number does not match the expected sequence number after the DL had issued a NAK.	R/W1C	0x0
16	Reserved	Reserved	R	0x0
15	RDUPTLP	Received Duplicate TLP 1 = A duplicate TLP is received.	R/W1C	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
14	RBEDB	Received Bad EDB 1 = A TLP ended with an EDB symbol and the value in the CRC field does not match the correct inverted CRC.	R/W1C	0x0
13	ROSEQ	Received TLP Out of Sequence 1 = A TLP is received out of sequence.	R/W1C	0x0
12	PMDLLP	Power Management DLLP 1 = A power management related DLLP is received.	R/W1C	0x0
11	Reserved	Reserved	R	0x0
10	UFCDLLP	Update Flow Control DLLP 1 = An update flow control DLLP has been received for VC0	R/W1C	0x0
9	Reserved	Reserved	R	0x0
8	IFCDLLP2	Init Flow Control 2 DLLP 1 = An INIT flow control 2 DLLP has been received for VC0.	R/W1C	0x0
7	Reserved	Reserved	R	0x0
6	IFCDLLP1	Init Flow Control 1 DLLP 1 = An INIT flow control 1 DLLP has been received for VC0.	R/W1C	0x0
5	ACKDLLP	ACK DLLP Received 1 = An ACK DLLP is received.	R/W1C	0x0
4	NACKDLLP	NACK DLLP Received 1 = An NACK DLLP is received.	R/W1C	0x0
3	DLLPCRCERR	DLLP with CRC Error Received 1 = A DLLP is received with a CRC error.	R/W1C	0x0
2	DLLP	DLLP Received 1 = A DLLP is received.	R/W1C	0x0
1	TLPCRCERR	TLP with CRC Error Received 1 = A TLP is received with a CRC error.	R/W1C	0x0
0	TLP	TLP Received 1= A TLP is received.	R/W1C	0x0

17.16.6 Data Link Transmit Status Register

Register name: DLTXSTS Reset value: 0x0000_0000	Register offset: 0x614

Bits	7	6	5	4	3	2	1	0
31:24	Reserved							VC0INITFCTO
23:16	Reserved		RPNUMRO	Reserved		REPLAYEVNT	REPLAYTO	ANTIMOUT
15:08	PMDLLP	Reserved	UFCDLLP	Reserved	IFCDLLP2	Reserved	IFCDLLP1	ACKDLLP
07:00	NACKDLLP	DLLP	Reserved		TXTLPNULL	TXRPTLP	TXNRPTLP	TLP

Bits	Name	Description	Type	Reset Value
31:25	Reserved	Reserved	R	0x0
24	VC0INITFCTO	VC0 Init FC Timeout 1 = An AVC0 initialization flow control timer timeout occurred. For more information, see Data Link Control 2 Register.	R/W1C	0x0
23:22	Reserved	Reserved	R	0x0
21	RPNUMRO	Replay Number Rollover 1 = The data link layer directed the physical layer to retrain due to a replay number counter rollover.	R/W1C	0x0
20:19	Reserved	Reserved	R	0x0
18	REPLAYEVNT	Replay Event 1 = A transmit replay event occurred.	R/W1C	0x0
17	REPLAYTO	Replay Timer Timeout 1 = The replay timer expired.	R/W1C	0x0
16	ANTIMOUT	ACK/NACK Latency Timeout 1 = An ACK/NACK latency timeout event occurred.	R/W1C	0x0
15	PMDLLP	Power Management DLLP 1 = A power management related DLLP is transmitted.	R/W1C	0x0
14	Reserved	Reserved	R	0x0
13	UFCDLLP	Update Flow Control DLLP 1 = An update flow control DLLP has been transmitted for VC0.	R/W1C	0x0
12	Reserved	Reserved	R	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
11	IFCDLLP2	Init Flow Control 1 DLLP Transmitted 1 = An INIT flow control 2 DLLP has been transmitted for VC0.	R/W1C	0x0
10	Reserved	Reserved	R	0x0
9	IFCDLLP1	Init Flow Control 1 DLLP Transmitted 1 = An an INIT flow control 1 DLLP has been transmitted for VC0.	R/W1C	0x0
8	ACKDLLP	ACK DLLP Transmitted 1 = An ACK DLLP is transmitted.	R/W1C	0x0
7	NACKDLLP	NACK DLLP Transmitted 1 = An NACK DLLP is transmitted.	R/W1C	0x0
6	DLLP	DLLP Transmitted 1 = A DLLP is transmitted.	R/W1C	0x0
5:4	Reserved	Reserved	R	0x0
3	TXTLPNULL	Transmit Nullified TLP 1 = The data link layer transmitted a nullified TLP.	R/W1C	0x0
2	TXRPTLP	Replay TLP Transmitted 1 = A replayed TLP is transmitted by the data link layer.	R/W1C	0x0
1	TXNRPTLP	Non-Replay TLP Transmitted 1 = A TLP not associated with a replay is transmitted.	R/W1C	0x0
0	TLP	TLP Transmitted 1 = Any TLP is transmitted.	R/W1C	0x0

17.16.7 Data Link Countables 0 Register

Register name: DLCNT0 Reset value: 0x0000_0000	Register offset: 0x618

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

Bits	Name	Description	Type	Reset Value
31:0	COUNT	This field counts the number of occurrences of the event specified in the DLCNT0SEL field in the Data Link Countables Configuration Register. This counter saturates at its maximum value. This field is cleared when read. 0x0 = Transmit a TLP 0x1 = Transmit an egress TLP 0x2 = Transmit a replay TLP 0x3 = Transmit a nullified TLP 0x4 = Transmit a egress nullified TLP 0x5 = Transmit a replay nullified TLP 0x6 = Transmit a DLLP 0x7 = Transmit a NAK DLLP 0x8 = Transmit a ACK DLLP 0x9 = Transmit a VC0 Init FC1 DLLP 0xA = Reserved 0xB = Transmit a VC0 Init FC2 DLLP 0xC = Reserved 0xD = Transmit a VC0 FC Update DLLP 0xE = Reserved 0xF = Transmit a power management DLLP 0x10 = ACK/NAK latency timer timeout 0x11 = Replay timer timeout 0x12 = Replay event has occurred 0x13 = Reserved 0x14 = replay number rollover 0x18 = VC0 Init FC timeout Others = Reserved	RCWS	0x0

17.16.8 Data Link Countables 1 Register

Register name: DLCNT1 Reset value: 0x0000_0000	Register offset: 0x61C

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

Bits	Name	Description	Type	Reset Value
31:0	COUNT	This field counts the number of 250-MHz cycles in which the event specified in the DLCNT1SEL field in the Data Link Countables Configuration Register occurs (that is, this field does not count the number of events, but rather the number of 250-MHz cycles in which the event was detected). This counter saturates at its maximum value. This field is cleared when read. 0x0 = Received a TLP 0x1 = Received a TLP that failed LCRC-32 check 0x2 = Received a DLLP 0x3 = Received a DLLP that failed LCRC-16 check 0x4 = Received a NAK DLLP 0x5 = Received a ACK DLLP 0x6 = Received a VC0 Initialization FC1 DLLP 0x7 = Reserved 0x8 = Received a VC0 Initialization FC2 DLLP 0x9 = Reserved 0xA = Received a VC0 FC Update DLLP 0xB = Reserved 0xC = Received a power management DLLP 0xD = Received a out of sequence TLP 0xE = Received a TLP with bad EDB 0xF = Received a duplicated TLP 0x10 = Reserved 0x11 = Received a nullified TLP (EDB) 0x12 = DLLP receive timer timeout, physical layer retrain 0x13 = Received a vendor defined DLLP 0x14 = Received a undefined type DLLP 0x15 = Received a TLP which length less than 5 dwords 0x16 = DLLP from link partner was discarded due to a protocol error Others = Reserved	RCWS	0x0

17.16.9 Data Link Countables Configuration Register

Register name: DLCNTCFG Reset value: 0x0000_0000	Register offset: 0x620

Bits	7	6	5	4	3	2	1	0
31:24	Reserved							
23:16	Reserved							
15:08	Reserved			DLCNT1SEL				
07:00	Reserved			DLCNT0SEL				

Bits	Name	Description	Type	Reset Value
31:13	Reserved	Reserved	R	0x0
12:8	DLCNT1SEL	Data Link Countables 1 Select This field selects the events that are counted by the COUNT field in the Data Link Countables 1 Register (see COUNT field for select value definitions).	R/WS	0x0
7:5	Reserved	Reserved	R	0x0
4:0	DLCNT0SEL	Data Link Countables 0 Select This field selects the events that are counted by the COUNT field in the Data Link Countables 0 Register (see COUNT field for select value definitions).	R/WS	0x0

17.17 Transaction Control and Status Registers

17.17.1 Transaction Layer Status Register

Register name: TLSTSE Reset value: 0x0000_0000	Register offset: 0x680

Bits	7	6	5	4	3	2	1	0
31:24	Reserved							
23:16	Reserved	RTLPME	IETLPME	RTLPIOPE	RTLPMPE	RTLPCPE	Reserved	RCVTLP
15:08	Reserved							
07:00	Reserved			RO	NULLIFIED	MALFORMED	Reserved	RUR

Bits	Name	Description	Type	Reset Value
31:23	Reserved	Reserved	R	0x0
22	RTLPME	Received TLP Message Event 1 = The corresponding event occurred.	R/W1C	0x0
21	IETLPME	Ingress Extracted TLP Message Event 1 = The corresponding event occurred.	R/W1C	0x0
20	RTLPIOPE	Received TLP I/O Packet Event 1 = The corresponding event occurred.	R/W1C	0x0
19	RTLPMPE	Received TLP Memory Packet Event 1 = The corresponding event occurred.	R/W1C	0x0
18	RTLPCPE	Received TLP Configuration Packet Event 1 = The corresponding event occurred.	R/W1C	0x0
17	Reserved	Reserved	R	0x0
16	RCVTLP	TLP Received 1 = A TLP is received.	R/W1C	0x0
15:5	Reserved	Reserved	R	0x0
4	RO	Receiver Overflow 1 = The number of credits for received TLPs exceeded the number allocated.	R/W1C	0x0
3	NULLIFIED	Nullified TLP Received 1 = A nullified TLP is received or the ingress stack nullifies a TLP.	R/W1C	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
2	MALFORMED	Malformed TLP Received 1 = A malformed TLP is received.	R/W1C	0x0
1	Reserved	Reserved	R	0x0
0	RUR	Receive Unsupported Request 1 = An unsupported request response is generated by the receive side transaction layer.	R/W1C	0x0

17.17.2 Transaction Layer Control Register

Register name: TLCTL Reset value: 0x0000_0000	Register offset: 0x684

Bits	7	6	5	4	3	2	1	0
31:24	Reserved							
23:16	Reserved							
15:08	Reserved				SEQTAG	FCUTIMERO	FCUTIMER	
07:00	FCUTIMER							

Bits	Name	Description	Type	Reset Value
31:12	Reserved	Reserved	R	0x0
11	SEQTAG	Sequence Tag 0 = The tag field in a posted transactions generated by the transaction layer have a zero value. 1 = The tag field in posted transactions generated by the transaction layer (for example, MSI) are incremented. When the extended tag field enable bit is set and this bit is set, the tag field increments from zero to 255. When the extended tag field enable bit is cleared and this bit is set, the tag field increments from zero to 31. This field is provided to aid in testing and debugging and has no functional effect during normal operation since the tag field in posted transactions is undefined. 0x0 = Field in generated posted transactions always zero. 0x1 = Tag field incremented in generated posted transactions.	R/WS	0x0
10	FCUTIMERO	Flow Control Update Timer Override 0 = The default transaction layer value is used. 1 = The transaction layer uses the value in the FCUTIMER field as the flow control update interval value (that is, the interval at which the PCIe Interface issues flow control update DLLPs to the link partner).	R/WS	0x0
9:0	FCUTIMER	Flow Control Update Timer This field contains the update flow control interval (in units of 250-MHz clock cycles) used when the FCUTIMERO bit is set.	R/WS	0x0

17.17.3 Transaction Layer Countables 0 Register

Register name: TLCNT0 Reset value: 0x0000_0000	Register offset: 0x688

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

Bits	Name	Description	Type	Reset Value
31:0	COUNT	Count This field counts the number of occurrences of the event specified in the TLCNT0SEL field in the Transaction Layer Countables Configuration Register. This counter saturates at its maximum value. This field is cleared when read. 0x0 = 0x6 Reserved 0x7 = TLP message received 0x8 = TLP message extracted 0x9 = TLP I/O request received 0xA = TLP memory request received 0xB = TLP configuration request received 0xD = TLP received 0xE = TLP nullified packet received 0xF = TLP malformed packet received 0x10 = Unsupported request received from the link (same as RUR bit in the Transaction Layer Status Register) 0x11 = Reserved 0x12 = TLP with ECRC received (with or without an error) 0x13 = TLP with ECRC error received 0x14 = TLP received from link with an unmapped TC 0x15 = Reserved 0x16 = Parity error detected in a TLP extracted on the ingress path (that is, configuration transaction or messages) Others = Reserved	RCWS	0x0

17.17.4 Transaction Layer Countables 1 Register

Register name: TLCNT1 Reset value: 0x0000_0000	Register offset: 0x68C

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

Bits	Name	Description	Type	Reset Value
31:0	COUNT	Count This field counts the number of occurrences of the event specified in the TLCNT1SEL field in the Transaction Layer Countables Configuration Register. This counter saturates at its maximum value. This field is cleared when read. 0x0 = 0x6-Reserved 0x7 = TLP message received 0x8 = TLP message extracted 0x9 = TLP I/O request received 0xA = TLP memory request received 0xB = TLP configuration request received 0xD = TLP received 0xE = TLP nullified packet received 0xF = TLP malformed packet received 0x10 = Unsupported request received from the link (same as RUR bit in the Transaction Layer Status Register) 0x11 = Reserved 0x12 = TLP with ECRC received (with or without an error) 0x13 = TLP with ECRC error received 0x14 = TLP received from link with an unmapped TC 0x15 = Reserved 0x16 = Parity error detected in a TLP extracted on the ingress path (that is, configuration transaction or messages) Others = Reserved	RCWS	0x0

17.17.5 Transaction Layer Countables Configuration Register

Register name: TLCNTCFG Reset value: 0x0000_0000	Register offset: 0x690

<table>
<tr><th>Bits</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>31:24</td><td colspan="8">BUS</td></tr>
<tr><td>23:16</td><td colspan="5">DEV</td><td colspan="3">FUNC</td></tr>
<tr><td>15:08</td><td colspan="6">Reserved</td><td colspan="2">TLCNT1SEL</td></tr>
<tr><td>07:00</td><td colspan="3">TLCNT1SEL</td><td colspan="5">TLCNT0SEL</td></tr>
</table>

Bits	Name	Description	Type	Reset Value
31:24	BUS	Bus Number As per the *PCI Express Specification (Rev. 2.1)*, functions capture the bus number for all type 0 configuration write requests completed by the Tsi721. This field contains the last bus number value captured by the Tsi721.	R	0x0
23:19	DEV	Device Number As per the *PCI Express Specification (Rev. 2.1)*, functions capture the device number for all type 0 configuration write requests completed by the Tsi721. This field contains the last device number value captured by the Tsi721.	R	0x0
18:16	FUNC	Function Number This field indicates the Tsi721's PCIe function number.	R	0x0
15:10	Reserved	Reserved	R	0x0
9:5	TLCNT1SEL	Transaction Layer Countables 1 Select This field selects the events that are counted by the COUNT field in the Transaction Layer Countables 1 Register. For select value definitions, see COUNT field.	R/WS	0x0
4:0	TLCNT0SEL	Transaction Layer Countables 0 Select This field selects the events that are counted by the COUNT field in the Transaction Layer Countables 0 Register. For select value definitions, see COUNT field.	R/WS	0x0

17.17.6 Legacy Interrupt Status Register

Register name: INTSTS Reset value: 0x0000_0000	Register offset: 0x6A0

Bits	7	6	5	4	3	2	1	0
31:24	Reserved							
23:16	Reserved							
15:08	Reserved							
07:00	Reserved				INTD	INTC	INTB	INTA

Bits	Name	Description	Type	Reset Value
31:4	Reserved	Reserved	R	0x0
3	INTD	INTD State 0x0 = INTD negated 0x1 = INTD asserted	R	0x0
2	INTC	INTC State 0x0 = INTC negated 0x1 = INTC asserted	R	0x0
1	INTB	INTB State 0x0 = INTB negated 0x1 = INTB asserted	R	0x0
0	INTA	INTA State 0x0 = INTA negated 0x1 = INTA asserted	R	0x0

17.18 Power Management Control and Status Registers

17.18.1 Power Management Proprietary Control 0 Register

Register name: PMPC0 Reset value: 0x06D6_0000	Register offset: 0x700

Bits	7	6	5	4	3	2	1	0
31:24	Reserved	D3HOTL1D	L1ASPMD	L0SASPMD	L0ET			
23:16	L0ET							
15:08	Reserved							
07:00	Reserved							

Bits	Name	Description	Type	Reset Value
31	Reserved	Reserved	R	0x0
30	D3HOTL1D	D3hot L1 Disable 1 = Disable entry into L1 when the Tsi721 is placed in the D3hot state. The intent of this bit is to provide an interoperability mode with link partners that do not properly support L1 by allowing the operating system to enable the D3hot feature but ensuring that the Tsi721 never initiates L1 entry.	R/WS	0x0
29	L1ASPMD	L1ASPM Disable 1 = Disable L1 ASPM on the Tsi721 regardless of the setting of the ASPM field in the PCIe Link Control Register. The intent of this bit is to provide an interoperability mode with link partners that do not properly support L1 by allowing the operating system to enable this feature but ensuring that the Tsi721 never initiates L1 entry.	R/WS	0x0
28	L0SASPMD	L0s ASPM Disable 1 = Disable L0s ASPM regardless of the setting of the ASPM field in the PCIe Link Control Register. The intent of this bit is to provide an interoperability mode with link partners that do not properly support L0s by allowing the operating system to enable this feature but ensuring that the Tsi721 never initiates L0s entry.	R/WS	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
27:16	L0ET	L0s Entry Timer This field specifies the L0s entry time value for the transmitter. If all L0s entry conditions are met for the specified amount of time, then the transmitter enters L0s. The timer value is specified in the number 4 ns clock cycles. The default value corresponds to the PCIe value of 7 us.	R/WS	0x6D6
15:0	Reserved	Reserved	R	0x0

17.18.2 Power Management Proprietary Control 1 Register

Register name: PMPC1 Reset value: 0x0000_003F	Register offset: 0x704

Bits	7	6	5	4	3	2	1	0
31:24	Reserved							
23:16	Reserved							
15:08	ENTRL0S	EXITL0S	ENTRL1P	ENTRL1C	ENTRL1T	ENTRLTA	ENTRLTR	EXITL1
07:00	Reserved		PMCS					

Bits	Name	Description	Type	Reset Value
31:16	Reserved	Reserved	R	0x0
15	ENTRL0S	Enter L0s 1 = The corresponding PM State Machine fencing condition is met. This bit is for factory testing and is a raw pre-conditioner for the corresponding state transition.Thus, the bit can toggle during PM states that are not directly applicable to the status condition of this bit.	R	0x0
14	EXITL0S	Exit L0s 1 = The corresponding PM State Machine fencing condition is met. This bit is for factory testing and is a raw pre-conditioner for the corresponding state transition.Thus, the bit can toggle during PM states that are not directly applicable to the status condition of this bit.	R	0x0
13	ENTRL1P	Enter L1 PCI PM 1 = The corresponding PM State Machine fencing condition is met. This bit is for factory testing and is a raw pre-conditioner for the corresponding state transition.Thus, the bit can toggle during PM states that are not directly applicable to the status condition of this bit.	R	0x0

This document is confidential and is subject to an NDA.

(Continued)

Bits	Name	Description	Type	Reset Value
12	ENTRL1C	Enter L1 Comply 1 = The corresponding PM State Machine fencing condition is met. This bit is for factory testing and is a raw pre-conditioner for the corresponding state transition.Thus, the bit can toggle during PM states that are not directly applicable to the status condition of this bit.	R	0x0
11	ENTRL1T	Enter L1 Timed 1 = The corresponding PM State Machine fencing condition is met. This bit is for factory testing and is a raw pre-conditioner for the corresponding state transition.Thus, the bit can toggle during PM states that are not directly applicable to the status condition of this bit.	R	0x0
10	ENTRLTA	Enter L1 Accept 1 = The corresponding PM State Machine fencing condition is met. This bit is for factory testing and is a raw pre-conditioner for the corresponding state transition.Thus, the bit can toggle during PM states that are not directly applicable to the status condition of this bit.	R	0x0
9	ENTRLTR	Enter L1 Reject 1 = The corresponding PM State Machine fencing condition is met. This bit is for factory testing and is a raw pre-conditioner for the corresponding state transition.Thus, the bit can toggle during PM states that are not directly applicable to the status condition of this bit.	R	0x0
8	EXITL1	Exit L1 1 = The corresponding PM State Machine fencing condition is met. This bit is for factory testing and is a raw pre-conditioner for the corresponding state transition.Thus, the bit can toggle during PM states that are not directly applicable to the status condition of this bit.	R	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
7:6	Reserved	Reserved	R	0x0
5:0	PMCS	Values for PM current state This field denotes the current value of the internal PM State Machine. Note that the I2C usually reads this field rather than Configuration Reads. parameter L0 0 L0s_Tx_entry_1 1 L0s_Tx_entry_2 2 L0s_Tx 3 L0s_Tx_exit_1 4 L0s_Tx_exit_2 5 L1pcipm_0 8 L1pcipm_1 9 L1pcipm_2 10 L1pcipm_3 11 L1pcipm_4 12 L1pcipm_5 13 L1pcipm_6 14 L1pcipm_7 15 L1comply_0 16 L1comply_1 17 L1comply_2 18 L1accept_0 19 L1accept_1 20 L1accept_2 21 L1reject_0 22 L1timed_0 23 L1timed_1 24 L1timed_2 25 L1timed_3 26 L1timed_nak 27 L1_entry_1 28 L1_entry_2 29 L1_entry_3 30 L1_entry_4 31 L1 32 L1_exit 33 L23ready_entry_0 40 L23ready_entry_1 41 L23ready_entry_2 42 L23ready_entry_3 43 L23ready_entry_4 44 L23ready_entry_5 45 L23ready_entry_6 46 L23ready_entry_7 47 L23ready_entry_8 48 L23ready_entry_9 49 L23ready_entry_10 50 L23ready 51 LDown 63 Others = Reserved	R	0x3F

17.19 Flow Control Registers

17.19.1 Flow Control VC0 Posted Credits Consumed Register

Register name: FCVC0PTCC Reset value: Undefined	Register offset: 0x800

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

Bits	Name	Description	Type	Reset Value
31:22	Reserved	Reserved	R	0x0
21:14	HDRFCCC	Header Flow Control Credit Count This field contains the current count for the posted header credits consumed counter.	R	Undefined
13:12	Reserved	Reserved	R	0x0
11:0	DATAFCCC	Data Flow Control Credit Count This field contains the current count for the posted data credits consumed counter.	R	Undefined

17.19.2 Flow Control VC0 Non-Posted Credits Consumed Register

Register name: FCVC0NPCC Reset value: Undefined	Register offset: 0x804

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

Bits	Name	Description	Type	Reset Value
31:22	Reserved	Reserved	R	0x0
21:14	HDRFCCC	Header Flow Control Credit Count This field contains the current count for the non-posted header credits consumed counter.	R	Undefined
13:12	Reserved	Reserved	R	0x0
11:0	DATAFCCC	Data Flow Control Credit Count This field contains the current count for the non-posted data credits consumed counter.	R	Undefined

17.19.3 Flow Control VC0 Completion Credits Consumed Register

Register name: FCVC0CPCC Reset value: Undefined	Register offset: 0x808

<table>
<tr><th>Bits</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>31:24</td><td colspan="8">Reserved</td></tr>
<tr><td>23:16</td><td colspan="2">Reserved</td><td colspan="6">HDRFCCC</td></tr>
<tr><td>15:08</td><td colspan="2">HDRFCCC</td><td colspan="2">Reserved</td><td colspan="4">DATAFCCC</td></tr>
<tr><td>07:00</td><td colspan="8">DATAFCCC</td></tr>
</table>

Bits	Name	Description	Type	Reset Value
31:22	Reserved	Reserved	R	0x0
21:14	HDRFCCC	Header Flow Control Credit Count This field contains the current count for the completion header credits consumed counter.	R	Undefined
13:12	Reserved	Reserved	R	0x0
11:0	DATAFCCC	Data Flow Control Credit Count This field contains the current count for the completion data credits consumed counter.	R	Undefined

17.19.4 Flow Control VC0 Posted Credit Limit Register

Register name: FCVC0PTCL Reset value: Undefined	Register offset: 0x80C

Bits	7	6	5	4	3	2	1	0
31:24	VALID	INFHDR	INFDAT	Reserved				
23:16	Reserved		HDRFCCC					
15:08	HDRFCCC		Reserved		DATAFCCC			
07:00	DATAFCCC							

Bits	Name	Description	Type	Reset Value
31	VALID	Credit Limit Valid 1 = The value in the DATAFCC and HDRFCC fields in this register are valid (that is, a flow control Init 1 DLLP was received from the link partner).	R	0x0
30	INFHDR	Infinite Header Credits 1 = Infinite header flow control credits were advertised by the link partner for the posted header credit limit. When this bit is set, the value of the corresponding credit counter is undefined.	R	0x0
29	INFDAT	Infinite Data Credits 1 = Infinite data flow control credits were advertised by the link partner for the posted data credit limit. When this bit is set, the value of the corresponding credit counter is undefined.	R	0x0
28:22	Reserved	Reserved	R	0x0
21:14	HDRFCCC	Header Flow Control Credit Count This field contains the current count for the posted header credit limit counter.	R	Undefined
13:12	Reserved	Reserved	R	0x0
11:0	DATAFCCC	Data Flow Control Credit Count This field contains the current count for the posted data credit limit counter.	R	Undefined

17.19.5 Flow Control VC0 Non-Posted Credit Limit Register

Register name: FCVC0NPCL Reset value: Undefined	Register offset: 0x810

Bits	7	6	5	4	3	2	1	0
31:24	VALID	INFHDR	INFDAT	Reserved				
23:16	Reserved		HDRFCCC					
15:08	HDRFCCC		Reserved		DATAFCCC			
07:00	DATAFCCC							

Bits	Name	Description	Type	Reset Value
31	VALID	Credit Limit Valid 1 = The value in the DATAFCC and HDRFCC fields are valid (that is, a flow control Init 1 DLLP was received from the link partner).	R	0x0
30	INFHDR	Infinite Header Credits 1 = Infinite header flow control credits were advertised by the link partner for the non-posted header credit limit. The value of the corresponding credit counter is undefined.	R	0x0
29	INFDAT	Infinite Data Credits 1 = Infinite data flow control credits were advertised by the link partner for the non-posted data credit limit. The value of the corresponding credit counter is undefined.	R	0x0
28:22	Reserved	Reserved	R	0x0
21:14	HDRFCCC	Header Flow Control Credit Count This field contains the current count for the non-posted header credit limit counter.	R	Undefined
13:12	Reserved	Reserved	R	0x0
11:0	DATAFCCC	Data Flow Control Credit Count This field contains the current count for the non-posted data credit limit counter.	R	Undefined

17.19.6 Flow Control VC0 Completion Credit Limit Register

Register name: FCVC0CPCL Reset value: Undefined	Register offset: 0x814

Bits	7	6	5	4	3	2	1	0
31:24	VALID	INFHDR	INFDAT	Reserved				
23:16	Reserved		HDRFCCC					
15:08	HDRFCCC		Reserved		DATAFCCC			
07:00	DATAFCCC							

Bits	Name	Description	Type	Reset Value
31	VALID	Credit Limit Valid 1 = The DATAFCC and HDRFCC fields are valid (that is, a flow control Init 1 DLLP was received from the link partner).	R	0x0
30	INFHDR	Infinite Header Credits 1 = Infinite header flow control credits were advertised by the link partner for the completion header credit limit. The value of the corresponding credit counter is also undefined.	R	0x0
29	INFDAT	Infinite Data Credits 1 = Infinite data flow control credits were advertised by the link partner for the completion data credit limit. The value of the corresponding credit counter is also undefined.	R	0x0
28:22	Reserved	Reserved	R	0x0
21:14	HDRFCCC	Header Flow Control Credit Count This field contains the current count for the completion header credit limit counter.	R	Undefined
13:12	Reserved	Reserved	R	0x0
11:0	DATAFCCC	Data Flow Control Credit Count This field contains the current count for the completion data credit limit counter.	R	Undefined

17.19.7 Flow Control VC0 Posted Credits Allocated Register

Register name: FCVC0PTCA Reset value: Undefined	Register offset: 0x818

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

Bits	Name	Description	Type	Reset Value
31:22	Reserved	Reserved	R	0x0
21:14	HDRFCCC	Header Flow Control Credit Count This field contains the current count for the posted header credits allocated counter.	R	Undefined
13:12	Reserved	Reserved	R	0x0
11:0	DATAFCCC	Data Flow Control Credit Count This field contains the current count for the posted data credits allocated counter.	R	Undefined

17.19.8 Flow Control VC0 Non-Posted Credits Allocated Register

Register name: FCVC0NPCA Reset value: Undefined	Register offset: 0x81C

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

Bits	Name	Description	Type	Reset Value
31:22	Reserved	Reserved	R	0x0
21:14	HDRFCCC	Header Flow Control Credit Count This field contains the current count for the non-posted header credits allocated counter.	R	Undefined
13:12	Reserved	Reserved	R	0x0
11:0	DATAFCCC	Data Flow Control Credit Count This field contains the current count for the non-posted data credits allocated counter.	R	Undefined

17.19.9 Flow Control VC0 Completion Credits Allocated Register

Register name: FCVC0CPCA Reset value: Undefined	Register offset: 0x820

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

Bits	Name	Description	Type	Reset Value
31:22	Reserved	Reserved	R	0x0
21:14	HDRFCCC	Header Flow Control Credit Count This field contains the current count for the completion header credits allocated counter.	R	Undefined
13:12	Reserved	Reserved	R	0x0
11:0	DATAFCCC	Data Flow Control Credit Count This field contains the current count for the completion data credits allocated counter.	R	Undefined

17.19.10 Flow Control VC0 Posted Credits Received Register

Register name: FCVC0PTCR Reset value: Undefined	Register offset: 0x824

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

Bits	Name	Description	Type	Reset Value
31:22	Reserved	Reserved	R	0x0
21:14	HDRFCCC	Header Flow Control Credit Count This field contains the current count for the posted header credits received counter.	R	Undefined
13:12	Reserved	Reserved	R	0x0
11:0	DATAFCCC	Data Flow Control Credit Count This field contains the current count for the posted data credits received counter.	R	Undefined

17.19.11 Flow Control VC0 Non-Posted Credits Received Register

Register name: FCVC0NPCR Reset value: Undefined	Register offset: 0x828

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

Bits	Name	Description	Type	Reset Value
31:22	Reserved	Reserved	R	0x0
21:14	HDRFCCC	Header Flow Control Credit Count This field contains the current count for the non-posted header credits received counter.	R	Undefined
13:12	Reserved	Reserved	R	0x0
11:0	DATAFCCC	Data Flow Control Credit Count This field contains the current count for the non-posted data credits received counter.	R	Undefined

17.19.12 Flow Control VC0 Completion Credits Received Register

Register name: FCVC0CPCR Reset value: Undefined	Register offset: 0x82C

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

Bits	Name	Description	Type	Reset Value
31:22	Reserved	Reserved	R	0x0
21:14	HDRFCCC	Header Flow Control Credit Count This field contains the current count for the completion header credits received counter.	R	Undefined
13:12	Reserved	Reserved	R	0x0
11:0	DATAFCCC	Data Flow Control Credit Count This field contains the current count for the completion data credits received counter.	R	Undefined

17.19.13 Egress Frame Buffer Transmission Control Register

Register name: EFBTC Reset value: 0x0000_0000	Register offset: 0x860

Bits	7	6	5	4	3	2	1	0
31:24	Reserved							
23:16	Reserved							
15:08	Reserved							
07:00	Reserved					VC0ICPT	VC0INPT	VC0IPTT

Bits	Name	Description	Type	Reset Value
31:3	Reserved	Reserved	R	0x0
2	VC0ICPT	VC0 Inhibit Completion TLP Transmission 1 = The Egress Buffer inhibits transmission of VC0 completion TLPs. The PCIe Interface continues normal flow control interaction with its link partner.	R/WS	0x0
1	VC0INPT	VC0 Inhibit Non-Posted TLP Transmission 1 = The Egress Buffer inhibits transmission of VC0 non-posted TLPs. The PCIe Interface continues normal flow control interaction with its link partner.	R/WS	0x0
0	VC0IPTT	VC0 Inhibit Posted TLP Transmission 1 = The Egress Buffer inhibits transmission of VC0 posted TLPs. The PCIe Interface continues normal flow control interaction with its link partner.	R/WS	0x0

17.20 IFB and EFB Countables Registers

17.20.1 IFB Countables 0 Register

Register name: IFBCNT0 Reset value: 0x0000_0000	Register offset: 0x8B0

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

Bits	Name	Description	Type	Reset Value
31:0	COUNT	This field counts the number of occurrences of the event specified in the IFBCNT0SEL field in the IFB Countables Configuration Register. This counter saturates at its maximum value. This field is cleared when read. 0x0 = Cut-through TLP 0x1 = Store and forward TLP 0x2 = TLP left IFB (broadcast is counted as 1) 0x3 = TLP discard because of nullified 0x4 = TLP discard because of time expiration 0x5 = Non-nullified NP TLP received 0x6 = Nullified NP TLP received 0x7 = Non-nullified NP TLP left the NP FIFO 0x8 = Nullified NP TLP left the NP FIFO 0x9 = Reserved 0xA = Inserted CP TLP received 0xB = Non-nullified CP TLP received 0xC = Nullified CP TLP received 0xD = Inserted CP TLP left CP FIFO 0xE = Non-nullified CP TLP left the CP FIFO 0xF = Nullified CP TLP left the CP FIFO 0x10 = Reserved 0x11 = Inserted PT TLP received 0x12 = Non-nullified PT TLP received 0x13 = Nullified PT TLP received 0x14 = Inserted PT TLP left PT FIFO 0x15 = Non-nullified PT TLP left the PT FIFO 0x16 = Nullified PT TLP left the PT FIFO 0x17 = IFB single bit error 0x18 = IFB double bit error	RCWS	0x0

17.20.2 IFB Countables 1 Register

Register name: IFBCNT1 Reset value: 0x0000_0000	Register offset: 0x8B4

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

Bits	Name	Description	Type	Reset Value
31:0	COUNT	This field counts the number of occurrences of the event specified in the IFBCNT0SEL field in the IFB Countables Configuration Register. This counter saturates at its maximum value. This field is cleared when read. This field has the same encoding as the COUNT field in the IFB Countables 0 Register.	RCWS	0x0

17.20.3 IFB Countables Configuration Register

Register name: IFBCNTCFG Reset value: 0x0000_0000	Register offset: 0x8B8

Bits	7	6	5	4	3	2	1	0
31:24	Reserved							
23:16	Reserved							
15:08	Reserved						IFBCNT1SEL	
07:00	IFBCNT1SEL			IFBCNT0SEL				

Bits	Name	Description	Type	Reset Value
31:10	Reserved	Reserved	R	0x0
9:5	IFBCNT1SEL	IFB Countables 1 Select This field selects the events that are counted by the COUNT field in the IFB Countables 1 Register (see the COUNT field for select value definition).	R/WS	0x0
4:0	IFBCNT0SEL	IFB Countables 0 Select This field selects the events that are counted by the COUNT field in the IFB Countables 0 Register (see the COUNT field for select value definition).	R/WS	0x0

17.20.4 EFB Countables 0 Register

Register name: EFBCNT0 Reset value: 0x0000_0000	Register offset: 0x8C0

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

Bits	Name	Description	Type	Reset Value
31:0	COUNT	This field counts the number of occurrences of the event specified in the EFBCNT0SEL field in the EFB Countables Configuration Register. This counter saturates at its maximum value. This field is cleared when read. 0x0 = Cut-through TLP 0x1 = Store and forward TLP 0x2 = TLP left EFB (internal or destined to link) 0x3 = Internal TLP left EFB 0x4 = TLP discard because of nullified 0x5 = TLP discard because of time expiration 0x6 = Non-nullified NP TLP received 0x7 = Nullified NP TLP received 0x8 = Non-nullified NP TLP left the NP FIFO 0x9 = Nullified NP TLP left the NP FIFO 0xA = Non-nullified CP TLP received 0xB = Nullified CP TLP received 0xC = Non-nullified CP TLP left the CP FIFO 0xD = Nullified CP TLP left the CP FIFO 0xE = Non-nullified PT TLP received 0xF = Nullified PT TLP received 0x10 = Non-nullified PT TLP left the PT FIFO 0x11 = Nullified PT TLP left the PT FIFO 0x12 = EFB single bit error 0x13 = EFB double bit error	RCWS	0x0

17.20.5 EFB Countables 1 Register

Register name: EFBCNT1 Reset value: 0x0000_0000	Register offset: 0x8C4

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

Bits	Name	Description	Type	Reset Value
31:0	COUNT	This field counts the number of occurrences of the event specified in the EFBCNT1SEL field in the EFB Countables Configuration Register. This counter saturates at its maximum value. This field is cleared when read. This field has the same encoding as the COUNT field in the EFBCNT0 field.	RCWS	0x0

17.20.6 EFB Countables Configuration Register

Register name: EFBCNTCFG Reset value: 0x0000_0040	Register offset: 0x8C8

Bits	7	6	5	4	3	2	1	0
31:24	Reserved							
23:16	Reserved							
15:08	Reserved						EFBCNT1SEL	
07:00	EFBCNT1SEL			EFBCNT0SEL				

Bits	Name	Description	Type	Reset Value
31:10	Reserved	Reserved	R	0x0
9:5	EFBCNT1SEL	EFB Countables 1 Select This field selects the events that are counted by the COUNT field in the EFB Countables 1 Register (see COUNT field for select value definitions).	R/WS	0x2
4:0	EFBCNT0SEL	EFB Countables 0 Select This field selects the events that are counted by the COUNT field in the EFB Countables 0 Register (see COUNT field for select value definitions).	R/WS	0x0

17.21 AER Error Emulation Registers

17.21.1 Uncorrectable Error Emulation Register

Register name: UEEM Reset value: 0x0000_0000	Register offset: 0xD90

Bits	7	6	5	4	3	2	1	0
31:24	ADVISORYNF	Reserved						
23:16	Reserved	UIE	Reserved	UR	ECRC	MALFORMED	RCVOVR	UECOMP
15:08	CABORT	COMPTO	Reserved	POISONED	Reserved			
07:00	Reserved			DLPERR	Reserved			

Bits	Name	Description	Type	Reset Value
31	ADVISORYNF	Advisory Non-Fatal Error Trigger If this bit is set together with another error bit in this register for which an advisory non-fatal error is possible (see the *PCI Express Specification (Rev. 2.1)*), an advisory non-fatal error is logged and reported in the PCIe Interface's AER capability structure, provided the error severity for the selected uncorrectable error is configured such that the error will be of type non-fatal. If this bit is set together with another error bit in this register for which an advisory non-fatal error is not possible, the operation is undefined. If this bit is set together with another error bit in this register for which an advisory non-fatal error is possible, but the severity of the selected uncorrectable error is fatal, then this bit is ignored and the selected error is logged and reported as a fatal error.	R/W1S	0x0
30:23	Reserved	Reserved	R	0x0
22	UIE	Uncorrectable Internal Error Trigger 1 = Causes the corresponding error to be detected and logged by the PCIe Interface. This bit returns 0x0 when read.	R/W1S	0x0
21	Reserved	Reserved	R	0x0
20	UR	UR Trigger 1 = Causes the corresponding error to be detected and logged by the PCIe Interface. This bit returns 0x0 when read.	R/W1S	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
19	ECRC	ECRC Trigger 1 = Causes the corresponding error to be detected and logged by the PCIe Interface. This bit returns 0x0 when read.	R/W1S	0x0
18	MALFORMED	Malformed TLP Trigger 1 = Causes the corresponding error to be detected and logged by the PCIe Interface. This bit returns 0x0 when read.	R/W1S	0x0
17	RCVOVR	Receiver Overflow Trigger 1 = Causes the corresponding error to be detected and logged by the PCIe Interface. This bit returns 0x0 when read.	R/W1S	0x0
16	UECOMP	Unexpected Completion Trigger 1 = Causes the corresponding error to be detected and logged by the PCIe Interface. This bit returns 0x0 when read.	R/W1S	0x0
15	CABORT	Completer Abort Status 1 = Causes the corresponding error to be detected and logged by the PCIe Interface. This bit returns 0x0 when read.	R/W1S	0x0
14	COMPTO	Completion Timeout Status 1 = Causes the corresponding error to be detected and logged by the PCIe Interface. This bit returns 0x0 when read.	R/W1S	0x0
13	Reserved	Reserved	R	0x0
12	POISONED	Poisoned TLP Trigger 1 = Causes the corresponding error to be detected and logged by the PCIe Interface. This bit returns 0x0 when read.	R/W1S	0x0
11:5	Reserved	Reserved	R	0x0
4	DLPERR	Data Link Protocol Error Trigger 1 = Causes the corresponding error to be detected and logged by the PCIe Interface. This bit returns 0x0 when read.	R/W1S	0x0
3:0	Reserved	Reserved	R	0x0

17.21.2 Correctable Error Emulation Register

Register name: CEEM Reset value: 0x0000_0000	Register offset: 0xD94

Bits	7	6	5	4	3	2	1	0
31:24	Reserved							
23:16	Reserved							
15:08	HLO	CIE	Reserved	RPLYTO	Reserved			RPLYROVR
07:00	BADDLLP	BADTLP	Reserved					RCVERR

Bits	Name	Description	Type	Reset Value
31:16	Reserved	Reserved	R	0x0
15	HLO	Header Log Overflow Trigger 1 = Causes the corresponding error to be detected and logged by the PCIe Interface. This bit returns 0x0 when read.	R/W1S	0x0
14	CIE	Correctable Internal Error Trigger 1 = Causes the corresponding error to be detected and logged by the PCIe Interface. This bit returns 0x0 when read.	R/W1S	0x0
13	Reserved	Reserved	R	0x0
12	RPLYTO	Replay Timer Timeout Trigger 1 = Causes the corresponding error to be detected and logged by the PCIe Interface. This bit returns 0x0 when read.	R/W1S	0x0
11:9	Reserved	Reserved	R	0x0
8	RPLYROVR	Replay Number Rollover Trigger 1 = Causes the corresponding error to be detected and logged by the PCIe Interface. This bit returns 0x0 when read.	R/W1S	0x0
7	BADDLLP	Bad DLLP Trigger 1 = Causes the corresponding error to be detected and logged by the PCIe Interface. This bit returns 0x0 when read.	R/W1S	0x0
6	BADTLP	Bad TLP Trigger 1 = Causes the corresponding error to be detected and logged by the PCIe Interface. This bit returns 0x0 when read.	R/W1S	0x0
5:1	Reserved	Reserved	R	0x0
0	RCVERR	Receiver Error Trigger 1 = Causes the corresponding error to be detected and logged by the PCIe Interface. This bit returns 0x0 when read.	R/W1S	0x0

17.22 SerDes Test Mode Registers

17.22.1 SerDes Test Mode Control Register

Register name: STMCTL Reset value: 0x0000_0000	Register offset: 0xE54

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

Bits	Name	Description	Type	Reset Value
31:4	Reserved	Reserved	R	0x0
3	SPEED	SerDes Speed This field controls the data rate of the SerDes when the PCIe Interface is in SerDes Test Mode. 0x0 = 2.5 Gbps 0x1 = 5.0 Gbps This field does not apply when PCIe Master Loopback test is selected in the TSEL field of the SerDes Test Mode Test Control Register, except when this test is entered by force entry (per the CMD field).	R/WS	0x0
2:0	CMD	Command This field directs the PCIe Interface to enter or exit SerDes test mode. When read this field returns the value of the last command that was written. 0x0 = No operation 0x1 = Enter SerDes test mode 0x2 = Exit SerDes test mode 0x3 = Reserved 0x4 = Reserved 0x5 = Force entry into SerDes test mode 0x6 = Reserved 0x7 = Abort command	R/WS	0x0

17.22.2 SerDes Test Mode Status Register

Register name: STMSTS Reset value: Undefined	Register offset: 0xE58

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

Bits	Name	Description	Type	Reset Value
31:3	Reserved	Reserved	R	0x0
2:1	PSTATE	PCIe Interface State This field contains the current test state of the PCIe Interface. 0x0 = Normal mode 0x1 = Entering SerDes test mode 0x2 = SerDes test mode 0x3 = Exiting SerDes test mode	RS	Undefined
0	CC	Command Completed This bit is set when a command written to the CMD field in the SerDes Test Mode Control Register is completed. This field is automatically cleared when a command is written to the CMD field in the SerDes Test Mode Control Register.	RS	0x0

17.22.3 SerDes Test Mode Test Control Register

Register name: STMTCTL Reset value: 0x0000_0000	Register offset: 0xE5C

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

Bits	Name	Description	Type	Reset Value
31:6	Reserved	Reserved	R	0x0
5:4	TSYNCP	Test Synchronization Period This field selects the amount of time spent in the synchronization step of the 10-bit and master loopback tests. The time period consists of two sub-periods: time to achieve CDR lock and time to achieve PRBS checker lock. 0x0 = 5 us (2 us for CDR lock + 3 us for PRBS checker lock) 0x1 = 10 us (4 us for CDR lock + 6 us for PRBS checker lock) 0x2 = 100 us (40 us for CDR lock + 60 us for PRBS checker lock) 0x3 = 1 ms (400 us for CDR lock + 600 us for PRBS checker lock)	R/WS	0x0
3:0	TSEL	Test Select This field selects the SerDes test mode that is enabled. 0x0 = Idle (no test executed) 0x1 = PCIe master loopback test 0x3 = 8-bit PRBS master loopback test 0x6 = Reserved 0x7 = 10-bit PCS slave loopback mode Others = Reserved	R/WS	0x0

17.22.4 SerDes Test Mode Test Status Register

Register name: STMTSTS Reset value: 0x0000_0000	Register offset: 0xE60

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

Bits	Name	Description	Type	Reset Value
31	TR	Test Running 1 = The selected master loopback test is running. This field is cleared each time a test is stopped or a new test starts execution (that is, when the TSEL field in the SerDes Test Mode Test Control Register changes, or when CMD field in the SerDes Test Mode Control Register changes). This field is not used for slave loopback tests.	RS	0x0
30:4	Reserved	Reserved	R	0x0
3:0	SYNC	Receive Synchronization Status Bit 3 = Lane 3 Bit 2 = Lane 2 Bit 1 = Lane 1 Bit 0 = Lane 0 If a test selected in the TSEL field in the SerDes Test Mode Test Control Register uses synchronization status, then a bit in this field is set when the corresponding lane has achieved synchronization. This field is cleared each time a new test starts execution (that is, when the TSEL field in the SerDes Test Mode Test Control Register changes, or when CMD field in the SerDes Test Mode Control Register changes).	RS	0x0

17.22.5 SerDes Test Mode Lane 0 Error Count Register

Register name: STMECNT0 Reset value: 0x0000_0000	Register offset: 0xE64

Bits	7	6	5	4	3	2	1	0
31:24	Reserved							
23:16	Reserved							ERR_INJ
15:08	Reserved				ERR_DET	OVR	COUNT	
07:00	COUNT							

Bits	Name	Description	Type	Reset Value
31:17	Reserved	Reserved	R	0x0
16	ERR_INJ	Error Inject For master loopback tests, setting this bit causes errors to be injected into the generated PRBS stream for the corresponding lane. When this bit is set, the PRBS generator injects at least one error into the generated stream. Once this occurs, this bit is automatically cleared by hardware. This bit can test the integrity of the PRBS checker.	R/W1S	0x0
15:12	Reserved	Reserved	R	0x0
11	ERR_DET	Error Detected For master loopback tests, this bit is set when an error is found in the corresponding lane during the test. This field can be read while the test is executing. This field is cleared each time a new test starts execution (that is, when the TSEL field in the SerDes Test Mode Test Control Register is written). This field is not cleared when the TSEL field transitions to the “Idle” state.	R/W1CS	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
10	OVR	Bit Error Rate Count Overflow 1 = The COUNT field is saturated. This field is cleared each time a new test starts execution (that is, when the TSEL field in the SerDes Test Mode Test Control Register is written). This field is not cleared when the TSEL field transitions to the “Idle” state. This field is not used for slave loopback tests.	RS	0x0
9:0	COUNT	Count For master loopback tests, this field counts the number of errors that were detected on the corresponding lane of the PCIe Interface by the SerDes PRBS data checker. The value in this field is valid only after a test completes execution. When the counter saturates at its maximum value, the OVR bit is set. This field is cleared each time a new test starts execution (that is, when the TSEL field in the SerDes Test Mode Test Control Register is written). This field is not cleared when the TSEL field transitions to the “Idle” state. This field is not used for slave loopback tests.	RS	0x0

17.22.6 SerDes Test Mode Lane 1 Error Count Register

Register name: STMECNT1 Reset value: 0x0000_0000	Register offset: 0xE68

Bits	7	6	5	4	3	2	1	0
31:24	Reserved							
23:16	Reserved							ERR_INJ
15:08	Reserved				ERR_DET	OVR	COUNT	
07:00	COUNT							

Bits	Name	Description	Type	Reset Value
31:17	Reserved	Reserved	R	0x0
16	ERR_INJ	Error Inject For master loopback tests, setting this bit causes errors to be injected into the generated PRBS stream for the corresponding lane. When this bit is set, the PRBS generator injects at least one error into the generated stream. Once this occurs, this bit is automatically cleared by hardware. This bit can test the integrity of the PRBS checker.	R/W1S	0x0
15:12	Reserved	Reserved	R	0x0
11	ERR_DET	Error Detected For master loopback tests, this bit is set when an error is found in the corresponding lane during the test. This field can be read while the test is executing. This field is cleared each time a new test starts execution (that is, when the TSEL field in the SerDes Test Mode Test Control Register is written). This field is not cleared when the TSEL field transitions to the “Idle” state.	R/W1CS	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
10	OVR	Bit Error Rate Count Overflow 1 = The COUNT field is saturated. This field is cleared each time a new test starts execution (that is, when the TSEL field in the SerDes Test Mode Test Control Register is written). This field is not cleared when the TSEL field transitions to the “Idle” state. This field is not used for slave loopback tests.	RS	0x0
9:0	COUNT	Count For master loopback tests, this field counts the number of errors that were detected on the corresponding lane of the PCIe Interface by the SerDes PRBS data checker. The value in this field is valid only after a test completes execution. When the counter saturates at its maximum value, the OVR bit is set. This field is cleared each time a new test starts execution (that is, when the TSEL field in the SerDes Test Mode Test Control Register is written). This field is not cleared when the TSEL field transitions to the “Idle” state. This field is not used for slave loopback tests.	RS	0x0

17.22.7 SerDes Test Mode Lane 2 Error Count Register

Register name: STMECNT2 Reset value: 0x0000_0000	Register offset: 0xE6C

Bits	7	6	5	4	3	2	1	0
31:24	Reserved							
23:16	Reserved							ERR_INJ
15:08	Reserved				ERR_DET	OVR	COUNT	
07:00	COUNT							

Bits	Name	Description	Type	Reset Value
31:17	Reserved	Reserved	R	0x0
16	ERR_INJ	Error Inject For master loopback tests, setting this bit causes errors to be injected into the generated PRBS stream for the corresponding lane. When this bit is set, the PRBS generator injects at least one error into the generated stream. Once this occurs, this bit is automatically cleared by hardware. This bit can test the integrity of the PRBS checker.	R/W1S	0x0
15:12	Reserved	Reserved	R	0x0
11	ERR_DET	Error Detected For master loopback tests, this bit is set when an error is found in the corresponding lane during the test. This field can be read while the test is executing. This field is cleared each time a new test starts execution (that is, when the TSEL field in the SerDes Test Mode Test Control Register is written). This field is not cleared when the TSEL field transitions to the “Idle” state.	R/W1CS	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
10	OVR	Bit Error Rate Count Overflow 1 = The COUNT field is saturated. This field is cleared each time a new test starts execution (that is, when the TSEL field in the SerDes Test Mode Test Control Register is written). This field is not cleared when the TSEL field transitions to the “Idle” state. This field is not used for slave loopback tests.	RS	0x0
9:0	COUNT	Count For master loopback tests, this field counts the number of errors that were detected on the corresponding lane of the PCIe Interface by the SerDes PRBS data checker. The value in this field is valid only after a test completes execution. When the counter saturates at its maximum value, the OVR bit is set. This field is cleared each time a new test starts execution (that is, when the TSEL field in the SerDes Test Mode Test Control Register is written). This field is not cleared when the TSEL field transitions to the “Idle” state. This field is not used for slave loopback tests.	RS	0x0

17.22.8 SerDes Test Mode Lane 3 Error Count Register

Register name: STMECNT3 Reset value: 0x0000_0000	Register offset: 0xE70

Bits	7	6	5	4	3	2	1	0
31:24	Reserved							
23:16	Reserved							ERR_INJ
15:08	Reserved				ERR_DET	OVR	COUNT	
07:00	COUNT							

Bits	Name	Description	Type	Reset Value
31:17	Reserved	Reserved	R	0x0
16	ERR_INJ	Error Inject For master loopback tests, setting this bit causes errors to be injected into the generated PRBS stream for the corresponding lane. When this bit is set, the PRBS generator injects at least one error into the generated stream. Once this occurs, this bit is automatically cleared by hardware. This bit can test the integrity of the PRBS checker.	R/W1S	0x0
15:12	Reserved	Reserved	R	0x0
11	ERR_DET	Error Detected For master loopback tests, this bit is set when an one error is found in the corresponding lane during the test. This field can be read while the test is executing. This field is cleared each time a new test starts execution (that is, when the TSEL field in the SerDes Test Mode Test Control Register is written). This field is not cleared when the TSEL field transitions to the “Idle” state.	R/W1CS	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
10	OVR	Bit Error Rate Count Overflow 1 = The COUNT field is saturated. This field is cleared each time a new test starts execution (that is, when the TSEL field in the SerDes Test Mode Test Control Register is written). This field is not cleared when the TSEL field transitions to the “Idle” state. This field is not used for slave loopback tests.	RS	0x0
9:0	COUNT	Count For master loopback tests, this field counts the number of errors that were detected on the corresponding lane of the PCIe Interface by the SerDes PRBS data checker. The value in this field is valid only after a test completes execution. When the counter saturates at its maximum value, the OVR bit is set. This field is cleared each time a new test starts execution (that is, when the TSEL field in the SerDes Test Mode Test Control Register is written). This field is not cleared when the TSEL field transitions to the “Idle” state. This field is not used for slave loopback tests.	RS	0x0

17.23 Application Layer Loopback Control and Status Registers

17.23.1 Application Layer Loopback Control and Status Register

Register name: ALLCS Reset value: 0x0000_0000	Register offset: 0xE84

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

Bits	Name	Description	Type	Reset Value
31	ALLS	Application Layer Loopback Status This bit indicates the current status of the application layer loopback mode. 0x0 = Application Layer Loopback Mode is disabled 0x1 = Application Layer Loopback Mode is enabled	RS	0x0
30:1	Reserved	Reserved	R	0x0
0	ALLME	Application Layer Loopback Mode Enable When this bit is set, the PCIe Interface enters application layer loopback mode. 0x0 = Disable Application Layer Loopback Mode 0x1 = Enable Application Layer Loopback Mode	R/WS	0x0

17.24 IFB Credit Control and Status Registers

17.24.1 IFB VC0 Flow Control Credit Posted Configuration Register

Register name: IFBVC0PTCFG Reset value: 0x0304_007F	Register offset: 0xE90

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

Bits	Name	Description	Type	Reset Value
31:26	Reserved	Reserved	R	0x0
25:16	PTDATA	Posted Data Credits This field contains the number of advertised IFB VC0 posted data flow control credits. Setting this field to a value less than one or greater than the number of supported flow control credits (that is, buffer size) produces undefined results.	R/WS	0x304
15:8	Reserved	Reserved	R	0x0
7:0	PTHDR	Posted Header Credits This field contains the number of advertised IFB VC0 posted header flow control credits. Setting this field to a value less than one or greater than the number of supported flow control credits (that is, buffer size) produces undefined results.	R/WS	0x7F

17.24.2 IFB VC0 Flow Control Credit Non-Posted Configuration Register

Register name: IFBVC0NPCFG Reset value: 0x0080_007F	Register offset: 0xE94

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

Bits	Name	Description	Type	Reset Value
31:26	Reserved	Reserved	R	0x0
25:16	NPDATA	Non-Posted Data Credits This field contains the number of advertised IFB VC0 non-posted data flow control credits. Setting this field to a value less than one or greater than the number of supported flow control credits (that is, buffer size) produces undefined results. Reset value A1 Revision = 0x080 A0 Revision = 0x100	R/WS	0x080
15:8	Reserved	Reserved	R	0x0
7:0	NPHDR	Non-Posted Header Credits This field contains the number of advertised IFB VC0 non-posted header flow control credits. Setting this field to a value less than one or greater than the number of supported flow control credits (that is, buffer size) produces undefined results.	R/WS	0x7F

17.24.3 IFB VC0 Flow Control Credit Completion Configuration Register

Register name: IFBVC0CPCFG Reset value: 0x0304_007F	Register offset: 0xE98

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

Bits	Name	Description	Type	Reset Value
31:26	Reserved	Reserved	R	0x0
25:16	CPDATA	Completion Data Credits This field contains the number of advertised IFB VC0 completion data flow control credits. Setting this field to a value less than one or greater than the number of supported flow control credits (that is, buffer size) produces undefined results.	R/WS	0x304
15:8	Reserved	Reserved	R	0x0
7:0	CPHDR	Completion Header Credits This field contains the number of advertised IFB VC0 completion header flow control credits. Setting this field to a value less than one or greater than the number of supported flow control credits (that is, buffer size) produces undefined results.	R/WS	0x7F

17.24.4 Ingress Flow Control Status Register

Register name: IFCSTS Reset value: 0x0000_0000	Register offset: 0xEA8

Bits	7	6	5	4	3	2	1	0
31:24	Reserved							
23:16	Reserved							
15:08	Reserved							
07:00	Reserved		VC0CPDO	VC0NPDO	VC0PTDO	VC0CPHO	VC0NPHO	VC0PTHO

Bits	Name	Description	Type	Reset Value
31:6	Reserved	Reserved	R	0x0
5	VC0CPDO	VC0 Completion Data Credit Overflow 1 = An ingress credit overflow has occurred on the corresponding ingress credit type.	R/W1C	0x0
4	VC0NPDO	VC0 Non-Posted Data Credit Overflow 1 = An ingress credit overflow has occurred on the corresponding ingress credit type.	R/W1C	0x0
3	VC0PTDO	VC0 Posted Data Credit Overflow 1 = An ingress credit overflow has occurred on the corresponding ingress credit type.	R/W1C	0x0
2	VC0CPHO	VC0 Completion Header Credit Overflow 1 = An ingress credit overflow has occurred on the corresponding ingress credit type.	R/W1C	0x0
1	VC0NPHO	VC0 Non-Posted Header Credit Overflow 1 = An ingress credit overflow has occurred on the corresponding ingress credit type.	R/W1C	0x0
0	VC0PTHO	VC0 Posted Header Credit Overflow 1 = An ingress credit overflow has occurred on the corresponding ingress credit type.	R/W1C	0x0

17.25 EFB Statistics Registers

The EFB statistics registers provide information on EFB credit availability, per packet type.

17.25.1 EFB VC0 Posted Credit Availability Register

Register name: EFBVC0PTSTS Reset value: Undefined	Register offset: 0xEC0

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

Bits	Name	Description	Type	Reset Value
31:27	Reserved	Reserved	R	0x0
26:16	DATA	Posted Data Credits This field contains the number of available VC0 posted data flow control credits.	R	Undefined
15:8	Reserved	Reserved	R	0x0
7:0	HDR	Posted Header Credits This field contains the number of available VC0 posted header flow control credits.	R	Undefined

17.25.2 EFB VC0 Non-Posted Credit Availability Register

Register name: EFBVC0NPSTS Reset value: Undefined	Register offset: 0xEC4

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

Bits	Name	Description	Type	Reset Value
31:27	Reserved	Reserved	R	0x0
26:16	DATA	Non-Posted Data Credits This field contains the number of available VC0 non-posted data flow control credits.	R	Undefined
15:8	Reserved	Reserved	R	0x0
7:0	HDR	Non-Posted Header Credits This field contains the number of available VC0 non-posted header flow control credits.	R	Undefined

17.25.3 EFB VC0 Completion Credit Availability Register

Register name: EFBVC0CPSTS Reset value: Undefined	Register offset: 0xEC8

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

Bits	Name	Description	Type	Reset Value
31:27	Reserved	Reserved	R	0x0
26:16	DATA	Completion Data Credits This field contains the number of available VC0 completion data flow control credits.	R	Undefined
15:8	Reserved	Reserved	R	0x0
7:0	HDR	Completion Header Credits This field contains the number of available VC0 completion header flow control credits.	R	Undefined

17.25.4 EFB Replay Buffer Statistics Register

Register name: EFBRBSTS Reset value: 0x0000_0000	Register offset: 0xECC

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

Bits	Name	Description	Type	Reset Value
31:9	Reserved	Reserved	R	0x0
8	FULL	Replay Buffer Full 1 = The replay buffer is full.	R/W1C	0x0
7:0	Reserved	Reserved	R	0x0

18. RapidIO Registers

This chapter discusses Tsi721's registers. Topics discussed include the following:

18.1 Overview

This chapter describes the Tsi721 register map, register access method, and the register definitions.

The Tsi721 supports standard RapidIO register blocks including the LP-Serial Physical Layer Registers, Error Management Extensions, and LP-serial per-lane registers.

The Tsi721 also supports implementation-specific registers. The implementation specific registers are organized into register blocks, with IDT-specific block header information to enable software compatibility and evolution.

18.2 Reserved Addresses and Fields

As required by the *RapidIO Interconnect Specification (Revision 2.1)*, Tsi721 handles reserved addresses and fields:

- Reads to reserved CAR bits return logic zeros.
- Writes to CARs do not affect the operation of Tsi721 after boot complete for those fields which are RES.
- Reads to reserved CARs do not cause an error.
- Reads to reserved CARs return logic zeros.

- Writes to reserved CARs do not cause an error.
- Writes to reserved CARs do not affect the operation of Tsi721.
- Reads to reserved CSRs and reserved Extended Features bits return logic zeros.
- Writes to reserved CSRs and reserved Extended Features bits do not affect operation of Tsi721.
- Reads to reserved CSRs and reserved Extended Features bits do not cause an error.
- Writes to reserved CSRs and reserved Extended Features registers do not affect operation of Tsi721.

RapidIO Implementation-Specific Registers reserved fields and addresses are handled:

- Reads to reserved Implementation Specific bits return undefined results.
- Writes to reserved Implementation Specific bits must be logic zero.
- Reads to reserved Implementation Specific Registers return undefined results.
- Writes to reserved Implementation Specific Registers is a programming error and may cause unpredictable results.

18.3 Register Map

Table 82: Register Map Overview

Register Group	Start Address	End Address
RapidIO Logical and Transport Registers	0x00000	0x000FC
RapidIO 1x or 4x LP-Serial Registers	0x00100	0x001BC
Reserved	0x001C0	0x00FFC
RapidIO Extended Error Management Registers	0x01000	0x0113C
Reserved	0x01140	0x02FFC
LP-Serial Lane Extended Features Block	0x03000	0x03084
Reserved	0x03088	0x0FFFC
RapidIO Implementation-Specific Registers • PLM Register Block • TLM Register Block • PBM Register Block • Event Management Registers • Port-Write Block Registers • LLM Registers • Fabric Module Registers • PRBS/BERT Control Registers	0x10000 • 0x10000 • 0x10300 • 0x10600 • 0x10900 • 0x10A00 • 0x10D00 • 0x10E00 • 0x12000	0x141FF • 0x102FF • 0x105FF • 0x108FF • 0x109FF • 0x10AFF • 0x10DFF • 0x10EFF • 0x141FF
Reserved	0x14200	0x1FFFC

Table 83: RapidIO Register Map

Offset	Register Name	See
RapidIO Logical and Transport Registers		
00000	RIO_DEV_ID	RapidIO Device Identity CAR
00004	RIO_DEV_INFO	RapidIO Device Information CAR
00008	RIO_ASBLY_ID	RapidIO Assembly Identity CAR
0000C	RIO_ASBLY_INFO	RapidIO Assembly Information CAR
00010	RIO_PE_FEAT	RapidIO Processing Element Features CAR
00018	RIO_SRC_OP	RapidIO Source Operation CAR
0001C	RIO_DEST_OP	RapidIO Destination Operations CAR
00020–00038	Reserved	
00040–00044	Reserved	
0004C	RIO_SR_XADDR	RapidIO Processing Element Logical Layer Control CSR
00050–00054	Reserved	
00060	RIO_BASE_ID	RapidIO Base deviceID CSR
00064	Reserved	
00068	RIO_HOST_BASE_ID_LOCK	RapidIO Host Base deviceID Lock CSR
0006C	RIO_COMP_TAG	RapidIO Component Tag CSR
00070–000FC	Reserved	
RapidIO 1x or 4x LP-Serial Registers		
00100	RIO_SP_MB_HEAD	RapidIO 1x or 4x Port Maintenance Block Header
00104–0011C	Reserved	
00120	RIO_SP_LT_CTL	RapidIO Port Link Timeout Control CSR
00124	RIO_SR_RSP_TO	RapidIO Response Timeout Register
00128–00138	Reserved	
0013C	RIO_SP_GEN_CTL	RapidIO Port General Control CSR
00140	RIO_SP_LM_REQ	RapidIO Port Link Maintenance Request CSR
00144	RIO_SP_LM_RESP	RapidIO Port Link Maintenance Response CSR
00148	RIO_SP_ACKID_STAT	RapidIO Port Local ackID Status CSR
0014C–00150	Reserved	
00154	RIO_SP_CTL2	RapidIO Port Control 2 CSR
00158	RIO_SP_ERR_STAT	RapidIO Port Error and Status CSR

Table 83: RapidIO Register Map *(Continued)*

Offset	Register Name	See
0015C	RIO_SP_CTL	RapidIO Port Control CSR
00160–00FFC	Reserved	
RapidIO Extended Error Management Registers		
01000	RIO_ERR_RPT_BH	RapidIO Error Reporting Block Header
01004	Reserved	
01008	RIO_ERR_DET	RapidIO Logical/Transport Layer Error Detect CSR
0100C	RIO_ERR_EN	RapidIO Logical/Transport Layer Error Enable CSR
01010	RIO_H_ADDR_CAPT	RapidIO Logical/Transport Layer High Address Capture CSR
01014	RIO_ADDR_CAPT	RapidIO Logical/Transport Layer Address Capture CSR
01018	RIO_ID_CAPT	RapidIO Logical/Transport Layer deviceID Capture CSR
0101C	RIO_CTRL_CAPT	RapidIO Logical/Transport Layer Control Capture CSR
01020–01024	Reserved	
01028	RIO_PW_TGT_ID	RapidIO Port-Write Target deviceID CSR
0102C–0103C	Reserved	
01040	RIO_SP_ERR_DET	RapidIO Port Error Detect CSR
01044	RIO_SP_RATE_EN	RapidIO Port Error Rate Enable CSR
01048	RIO_SP_ERR_ATTR_CAPT	RapidIO Port Attributes Capture CSR
0104C	RIO_SP_ERR_CAPT_0	RapidIO Port Packet/Control Symbol Error Capture CSR 0
01050	RIO_SP_ERR_CAPT_1	RapidIO Port Packet Error Capture CSR 1
01054	RIO_SP_ERR_CAPT_2	RapidIO Port Packet Error Capture CSR 2
01058	RIO_SP_ERR_CAPT_3	RapidIO Port Packet Error Capture CSR 3
0105C–01064	Reserved	
01068	RIO_SP_ERR_RATE	RapidIO Port Error Rate CSR
0106C	RIO_SP_ERR_THRESH	RapidIO Port Error Rate Threshold CSR
01070–02FFC	Reserved	
LP-Serial Lane Extended Features Block		
03000	RIO_PER_LANE_BH	RapidIO LP-Serial Per-Lane Extended Features Block Header
03004–0300C	Reserved	
03010	RIO_LANE0_STAT0	RapidIO Lane {0..3} Status 0 CSR
03014	RIO_LANE0_STAT1	RapidIO Lane {0..3} Status 1 CSR

Table 83: RapidIO Register Map *(Continued)*

Offset	Register Name	See
03018–0302C	Reserved (for RIO_LANE0_STAT2–7)	
03030–0304C	Lane 1	Same set of registers as Serial Lane 0 (03010-0302C)
03050–0306C	Lane 2	Same set of registers as Serial Lane 0 (03010-0302C)
03070–0308C	Lane 3	Same set of registers as Serial Lane 0 (03010-0302C)
03090–0FFFC	Reserved (for other capability features)	
RapidIO Implementation-Specific Registers		
PLM Register Block		
10000	RIO_PLM_BH	IDT-Specific Physical Layer Block Header Register
10004–1007C	Reserved	
10080	RIO_PLM_SP_IMP_SPEC_CTL	RapidIO PLM Port Implementation Specific Control Register
10090	RIO_PLM_SP_STATUS	RapidIO PLM Port Event Status Register
10094	RIO_PLM_SP_INT_ENABLE	RapidIO PLM Port Interrupt Enable Register
10098	RIO_PLM_SP_PW_ENABLE	RapidIO PLM Port Port-Write Enable Register
1009C	RIO_PLM_SP_EVENT_GEN	RapidIO PLM Port Event Generate Register
100A0	RIO_PLM_SP_ALL_INT_EN	RapidIO PLM Port All Interrupts Enable Register
100A4	RIO_PLM_SP_ALL_PW_EN	RapidIO PLM Port All Port-Writes Enable Register
100B4	RIO_PLM_SP_DISCOVERY_TIMER	RapidIO PLM Port Discovery Timer Register
100B8	RIO_PLM_SP_SILENCE_TIMER	RapidIO PLM Port Silence Timer Register
100BC	RIO_PLM_SP_VMIN_EXP	RapidIO PLM Port Vmin Exponent Register
100C0	RIO_PLM_SP_POL_CTL	RapidIO PLM Port Lane Polarity Control Register
100C8	RIO_PLM_SP_DENIAL_CTL	RapidIO PLM Port Packet Denial Control Register
100D0	RIO_PLM_SP_RCVD_MECS	RapidIO PLM Port Received MECS Status Register
100D4	RESERVED	
100D8	RIO_PLM_SP_MECS_FWD	RapidIO PLM Port MECS Forwarding Register
100DC	RESERVED	
100E0	RIO_PLM_SP_LONG_CS_TX1	RapidIO PLM Port Control Symbol Transmit 1 Register
100E4	RIO_PLM_SP_LONG_CS_TX2	RapidIO PLM Port Control Symbol Transmit 2 Register
100E8–102FC	RESERVED	

Table 83: RapidIO Register Map *(Continued)*

Offset	Register Name	See
TLM Register Block		
10300	RIO_TLM_BH	IDT-Specific Transport Layer Block Header Register
10304–1037C	RESERVED	
10380	RIO_TLM_SP_CONTROL	RapidIO TLM Port Control Register
10390	RIO_TLM_SP_STATUS	RapidIO TLM Port Status Register
10394	RIO_TLM_SP_INT_ENABLE	RapidIO TLM Port Interrupt Enable Register
10398	RIO_TLM_SP_PW_ENABLE	RapidIO TLM Port Port-Write Enable Register
1039C	RIO_TLM_SP_EVENT_GEN	RapidIO TLM Port Event Generate Register
103A0	RIO_TLM_SP_BRR_CSR{0..15}	RapidIO TLM Base Routing Register Control Register {0..15}
103A4	RIO_TLM_SP_BRR_PATTERN_MATCH_CSR{0,4,8,12}	RapidIO TLM Base Routing Register Pattern and Match Register 0
103B4	RIO_TLM_SP_BRR_PATTERN_MATCH_CSR{1,2,3,5,6,7,9,10,11,13,14,15}	RapidIO TLM Base Routing Register Pattern and Match Register 1
103A8-103AC	RESERVED	
103B8–103BC	RESERVED	
103C8–103CC	RESERVED	
103D8–103DC	RESERVED	
103E0	RIO_TLM_SP_FTYPE_FILTER_CSR	RapidIO TLM Port ftype Filter Control Register
103E4–105FC	RESERVED	
PBM Register Block		
10600	RIO_PBM_BH	IDT-Specific Packet Buffer Module Header Register
10680	RIO_PBM_SP_CONTROL_CSR	RapidIO PBM Port Control Register
10690	RIO_PBM_SP_STATUS	RapidIO PBM Port Status Register
10694	RIO_PBM_SP_INT_ENABLE	RapidIO PBM Port Interrupt Enable Register
10698	RIO_PBM_SP_PW_ENABLE	RapidIO PBM Port Port-Write Enable Register
1069C	RIO_PBM_SP_EVENT_GEN	RapidIO PBM Port Event Generate Register
106B0	RIO_PBM_SP_IG_WATERMARK0	RapidIO PBM Port Ingress Watermarks 0 Register
106B4	RIO_PBM_SP_IG_WATERMARK1	RapidIO PBM Port Ingress Watermarks 1 Register
106B8	RIO_PBM_SP_IG_WATERMARK2	RapidIO PBM Port Ingress Watermarks 2 Register
106BC	RIO_PBM_SP_IG_WATERMARK3	RapidIO PBM Port Ingress Watermarks 3 Register

Table 83: RapidIO Register Map *(Continued)*

Offset	Register Name	See
10880–108FC	RESERVED	
Event Management Registers		
10900	RIO_EM_BH	IDT-Specific Event Management Block Header
10910	RIO_EM_INT_STAT	RapidIO Event Management Interrupt Status Register
10914	RIO_EM_INT_ENABLE	RapidIO Event Management Interrupts Enable Register
10918	RIO_EM_INT_PORT_STAT	RapidIO Event Management Interrupt Port Status Register
1091C	RESERVED	
10920	RIO_EM_PW_STAT	RapidIO Event Management Port-Write Status Register
10924	RIO_EM_PW_ENABLE	RapidIO Event Management Port-Write Enable Register
10928	RIO_EM_PW_PORT_STAT	RapidIO Event Management Port-Write Port Status Register
1092C	RESERVED	
10930	RIO_EM_DEV_INT_EN	RapidIO Event Management Device Interrupt Enable Register
10934	RIO_EM_DEV_PW_EN	RapidIO Event Management Device Port-Write Enable Register
10938	RESERVED	
1093C	RIO_EM_MECS_STAT	RapidIO Event Management MECS Status Register
10940	RIO_EM_MECS_INT_EN	RapidIO Event Management MECS Interrupt Enable Register
10944	RIO_EM_MECS_CAP_EN	RapidIO Event Management MECS Capture Out Register
10948	RIO_EM_MECS_TRIG_EN	RapidIO Event Management MECS Trigger In Register
1094C	RIO_EM_MECS_REQ	RapidIO Event Management MECS Request Register
10950	RIO_EM_MECS_PORT_STAT	RapidIO Event Management MECS Port Status Register
1095C	RIO_EM_MECS_EVENT_GEN	RapidIO Event Management MECS and ECC Event Generate Register
10960	RIO_EM_RST_PORT_STAT	RapidIO Event Management Reset Request Port Status Register
10964	RESERVED	
10968	RIO_EM_RST_INT_EN	RapidIO Event Management Reset Request Interrupt Enable Register
1096C	RESERVED	
10970	RIO_EM_RST_PW_EN	RapidIO Event Management Reset Request Port-Write Enable Register
10974–109FC	RESERVED	

Table 83: RapidIO Register Map *(Continued)*

Offset	Register Name	See
Port-Write Block Registers		
10A00	RIO_PW_BH	IDT-Specific RapidIO Port-Write Block Header
10A04	RIO_PW_CTL	RapidIO Port-Write Control Register
10A08	RIO_PW_ROUTE	RapidIO Port-Write Routing Register
10A0C	RESERVED	
10A10	RIO_PW_RX_STAT	RapidIO Port-Write Reception Status CSR
10A14	RIO_PW_RX_EVENT_GEN	RapidIO Port-Write Reception Event Generate Register
10A18–10A1C	RESERVED	
10A20	RIO_PW_RX_CAPT0	RapidIO Port-Write Reception Capture CSR {0..3}
10A24	RIO_PW_RX_CAPT1	RapidIO Port-Write Reception Capture CSR {0..3}
10A28	RIO_PW_RX_CAPT2	RapidIO Port-Write Reception Capture CSR {0..3}
10A2C	RIO_PW_RX_CAPT3	RapidIO Port-Write Reception Capture CSR {0..3}
10A30–10CFC	RESERVED	
LLM Registers		
10D00	RIO_LLM_BH	IDT-Specific LLM Module Block Header
10D04–10D0C	RESERVED	
10D10	RIO_MTC_WR_RESTRICT	RapidIO Maintenance Write Request Restriction Control Register
10D14	RIO_MTC_PWR_RESTRICT	RapidIO Maintenance Port-Write Request Restriction Control Register
10D18	RIO_MTC_RD_RESTRICT	RapidIO Maintenance Read Request Restriction Control Register
10D1C	RESERVED	
10D24	RIO_WHITEBOARD	RapidIO Whiteboard CSR
10D28	RIO_PORT_NUMBER	RapidIO Port IP Prescalar for SRV_CLK Register
10D2C	Reserved	
10D30	RIO_PRESCALAR_SRV_CLK	RapidIO Port IP Prescalar for SRV_CLK Register
10D34	RIO_REG_RST_CTL	RapidIO Register Reset Control CSR
10D38–10D44	RESERVED	
10D48	RIO_LOCAL_ERR_DET	RapidIO Local Logical/Transport Layer Error Detect CSR
10D4C	RIO_LOCAL_ERR_EN	RapidIO Local Logical/Transport Layer Error Enable CSR
10D50	RIO_LOCAL_H_ADDR_CAPT	RapidIO Local Logical/Transport Layer High Address Capture CSR

Table 83: RapidIO Register Map *(Continued)*

Offset	Register Name	See
10D54	RIO_LOCAL_ADDR_CAPT	RapidIO Local Logical/Transport Layer Address Capture CSR
10D58	RIO_LOCAL_ID_CAPT	RapidIO Local Logical/Transport Layer deviceID Capture CSR
10D5C	RIO_LOCAL_CTRL_CAPT	RapidIO Local Logical/Transport Layer Control Capture CSR
10D60–10DFC	RESERVED	
Fabric Module Registers		
10E00	RIO_FABRIC_BH (DR)	IDT-Specific Fabric Module Block Header
10E04–10E0C	RESERVED	
10E14–10E3C	RESERVED	
10E50–13FFF	RESERVED	
PRBS/BERT Control Registers		
12000	RIO_PRBS_BH	IDT-Specific PRBS/BERT Block Header
12004	RIO_PRBS_LANE0_CTRL	RapidIO PRBS Lane {0..3} Control Register
12008	RIO_PRBS_LANE0_SEED	RapidIO PRBS Lane {0..3} PRBS Seed Register
1200C	RIO_PRBS_LANE0_ERR_COUNT	RapidIO PRBS Lane {0..3} Error Count Register
12010	Reserved	
12014–12020	Repeated for Lane 1	Same registers as 0x12004–0x12010
12024–12030	Repeated for Lane 2	Same registers as 0x12004–0x12010
12034–12040	Repeated for Lane 3	Same registers as 0x12004–0x12010
12040–17FFC	RESERVED	

18.4 RapidIO Logical and Transport Registers

A RapidIO processing element (PE) must contain a set of capability registers (CARs) that allow an external processing element to determine its capabilities through maintenance read operations. All CARs are 32 bits wide and are organized and accessed in 32-bit quantities. CARs are read-only.

A processing element must contain a set of command and status registers (CSRs) that allow an external processing element to determine and control the status of its internal hardware. All CSRs are 32 bits wide and are organized and accessed in the same way as CARs.

18.4.1 RapidIO Device Identity CAR

This register identifies the device and vendor information for the Tsi721.

Register Name: RIO_DEV_ID Reset Value: 0x80AB_0038	Register Offset: 0x00000

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 vendor-assigned part number of the device. 0x80AB = Tsi721	RES	0x80AB
16:31	DEV_VEN_ID	Device Vendor Identifier This field identifies IDT as the vendor that manufactured the device. This value is assigned by the RapidIO Trade Association.	RES	0x0038

18.4.2 RapidIO Device Information CAR

This register identifies the revision level of the device.

Register Name: RIO_DEV_INFO Reset Value: 0x0000_00001	Register Offset: 0x00004

Bits	0	1	2	3	4	5	6	7
00:7	DEV_REV							
07:15	DEV_REV							
16:23	DEV_REV							
24:31	DEV_REV							

Bits	Name	Description	Type	Reset Value
0:31	DEV_REV	Device Revision level Reset value Revision A1 = 0x1 Revision A0 = 0x0	RES	1

18.4.3 RapidIO Assembly Identity CAR

This register contains assembly identification information about the Tsi721.

Register Name: RIO_ASBLY_ID Reset Value: 0x0000_0038	Register Offset: 0x00008

Bits	0	1	2	3	4	5	6	7
00:7	ASBLY_ID							
07:15	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.	RES	0x0000
16:31	ASBLY_VEN_ID	Assembly Vendor ID Identifies the vendor that manufactured the assembly or subsystem that contains the device.	RES	0x0038

18.4.4 RapidIO Assembly Information CAR

This register contains additional information about the device's assembly.

Register Name: RIO_ASBLY_INFO Reset Value: 0x0000_0100	Register Offset: 0x0000C

Bits	0	1	2	3	4	5	6	7
00:7	ASBLY_REV							
07:15	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	RES	0x0000
16:31	EXT_FEAT_PTR	Extended Features Pointer This is the pointer to the first entry in the extended features list. This indicates that address 0x100 of the RapidIO maintenance space contains a valid RapidIO register block header.	RES	0x0100

18.4.5 RapidIO Processing Element Features CAR

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

Register Name: RIO_PE_FEAT Reset Value: 0xC000_003F	Register Offset: 0x00010

Bits	0	1	2	3	4	5	6	7
00:07	BRDG	MEM	PROC	SW	MULT_P	Reserved		
08:15	Reserved							
16:23	Reserved				FLOW_ARB	MC	ERTC	SRTC
24:31	FLOW_CTRL	Reserved	CRF	CTLS	EXT_FEA	EXT_AS		

Bits	Name	Description	Type	Reset Value
0	BRDG	Bridge 0 = Device is not a bridge 1 = Device can bridge to another interface	RES	1
1	MEM	Endpoint 0 = Device is not a RapidIO endpoint addressable for reads and writes 1 = Device has physically addressable local address space and can be accessed as an endpoint through Logical I/O (that is, NREAD and NWRITE) transactions.	RES	1
2	PROC	Processor 0 = Device is not a processor 1 = Device 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 above).	RES	0
3	SW	Switching Capabilities Device can bridge to another external RapidIO interface. For example, a device with two S-RIO ports and a local endpoint is a two-port switch, not a three-port switch, regardless of its internal architecture. 0 = Device is not a switch 1 = Device is a switch. ftype 8 packets with hop count equal to 0 are routed to the register bus.	RES	0

(Continued)

Bits	Name	Description	Type	Reset Value
4	MULT_P	Multiport 0 = Device does not implement multiple external S-RIO ports. 1 = Device implements multiple external S-RIO ports.	RES	0
5:19	Reserved	Reserved	R	0
20	FLOW_ARB	Flow Arbitration Support 0 = Device does not support the flow arbitration extension 1 = Reserved	RES	0
21	MC	Multicast 0 = Device does not support the multicast extension 1 = Device supports the multicast extension	RES	0
22	ERTC	Extended Route Table Configuration Support 0 = Device does not support the extended route table configuration method 1 = Device supports the extended route table configuration method	RES	0
23	SRTC	Standard Route Table Configuration Support 0 = Device does not support the standard route table configuration method 1 = Device supports the standard route table configuration method	RES	0
24	FLOW_CTRL	Flow Control support 0 = Device does not support the flow control extension 1 = Reserved	RES	0
25	Reserved	Reserved	R	0
26	CRF	Critical Request Flow support 0 = Device does not support Critical Request Flow 1 = Device supports Critical Request Flow The use of CRF in selecting a VOQ is set in RapidIO TLM Port Control Register.VOQ_SELECT.	RES	1
27	CTLS	Common Transport Large System support The srcID and destID of the endpoint are generated in accordance with this bit. 0 = Device supports 8-bit destIDs only 1 = Device supports 8- and 16-bit destIDs	RES	1

(Continued)

Bits	Name	Description	Type	Reset Value
28	EXT_FEA	Extended Features Pointer is valid This is the pointer to the first entry in the extended features list.	RES	1
29:31	EXT_AS	Indicates the number of address bits supported by the PE both as a source and target of an operation. All PEs must support 34-bit addresses at a minimum. 0b111 = PE supports 66, 50, and 34-bit addresses 0b101 = PE supports 66 and 34-bit addresses 0b011 = PE supports 50 and 34-bit addresses 0b001 = PE supports 34-bit addresses All other encodings are reserved.	RES	0x7

18.4.6 RapidIO Source Operation CAR

This register identifies the set of RapidIO I/O logical operations that can be issued by an application. The Tsi721 must be able to generate I/O logical maintenance read and write requests if it is required to access CARs and CSRs in other devices.

Register Name: RIO_SRC_OP Reset Value: 0x0000_FC04	Register Offset: 0x00018

Bits	0	1	2	3	4	5	6	7
00:07	G_READ	G_IREAD	G_READ_OWN	G_DC_INVALIDATE	G_CASTOUT	G_DC_FLUSH	G_IO_READ	G_IC_INVALIDATE
08:15	G_TLB_INVALIDATE	G_TLB_SYNC	RIO_RSVD_10	RIO_RSVD_11	DS_TM	DS	IMPLEMENT_DEF	
16:23	READ	WRITE	STRM_WR	WR_RES	D_MSG	DBELL	ACSWAP	ATSWAP
24:31	A_INC	A_DEC	A_SET	A_CLEAR	A_SWAP	PORT_WR	IMPLEMENT_DEF2	

Bits	Name	Description	Type	Reset Value
0	G_READ	1 = GSM read operation is supported	RES	0
1	G_IREAD	1 = GSM instruction read operation is supported	RES	0
2	G_READ_OWN	1 = GSM read-for-ownership operation is supported	RES	0
3	G_DC_INVALIDATE	1 = GSM data cache invalidate operation is supported	RES	0
4	G_CASTOUT	1 = GSM castout operation is supported	RES	0
5	G_DC_FLUSH	1 = GSM data cache flush is supported	RES	0
6	G_IO_READ	1 = GSM I/O read operation is supported	RES	0
7	G_IC_INVALIDATE	1 = GSM instruction cache invalidate operation is supported	RES	0
8	G_TLB_INVALIDATE	1 = GSM TLB invalidate-entry operation is supported	RES	0
9	G_TLB_SYNC	1 = GSM TLB invalidate entry sync operation is supported	RES	0
10	RIO_RSVD_10	This field is reserved by the *RapidIO Interconnect Specification (Revision 2.1)*. Tsi721 allows this field to be set to support some future and as yet undefined operation.	RES	0
11	RIO_RSVD_11	This field is reserved by the *RapidIO Interconnect Specification (Revision 2.1)*. Tsi721 allows this field to be set to support some future and as yet undefined operation.	RES	0
12	DS_TM	1 = Data streaming traffic management is supported	RES	0
13	DS	1 = Data streaming operation is supported	RES	0

(Continued)

Bits	Name	Description	Type	Reset Value
14:15	IMPLEMENT_DEF	Implementation defined	RES	0
16	READ	1 = Read operation is supported	RES	1
17	WRITE	1 = Write operation is supported	RES	1
18	STRM_WR	1 = Streaming write operation is supported	RES	1
19	WR_RES	1 = Write-with-response operation is supported	RES	1
20	D_MSG	1 = Data messaging is supported	RES	1
21	DBELL	1 = Doorbells are supported	RES	1
22	ACSWAP	1 = Atomic (compare-and-swap) operation is supported	RES	0
23	ATSWAP	1 = Atomic (test-and-swap) operation is supported	RES	0
24	A_INC	1 = Atomic (increment) operation is supported	RES	0
25	A_DEC	1 = Atomic (decrement) operation is supported	RES	0
26	A_SET	1 = Atomic (set) operation is supported	RES	0
27	A_CLEAR	1 = Atomic (clear) operation is supported	RES	0
28	A_SWAP	1 = Atomic (swap) operation is supported	RES	0
29	PORT_WR	1 = Port-write operation is supported	RES	1
30:31	IMPLEMENT_DEF2	Implementation defined	RES	0

18.4.7 RapidIO Destination Operations CAR

This register defines the set of RapidIO I/O logical operations that can be supported by the Tsi721. The device can respond to Maintenance read and write requests.

Register Name: RIO_DEST_OP Reset Value: 0x0000_FC04	Register Offset: 0x0001C

Bits	0	1	2	3	4	5	6	7
00:07	G_READ	G_IREAD	G_READ_OWN	G_DC_INVALIDATE	G_CASTOUT	G_DC_FLUSH	G_IO_READ	G_IC_INVALIDATE
08:15	G_TLB_INVALIDATE	G_TLB_SYNC	RIO_RSVD_10	RIO_RSVD_11	DS_TM	DS	IMPLEMENT_DEF	
16:23	READ	WRITE	STRM_WR	WR_RES	D_MSG	DBELL	ACSWAP	ATSWAP
24:31	A_INC	A_DEC	A_SET	A_CLEAR	A_SWAP	PORT_WR	IMPLEMENT_DEF2	

Bits	Name	Description	Type	Reset Value
0	G_READ	1 = GSM read operation is supported	RES	0
1	G_IREAD	1 = GSM instruction read operation is supported	RES	0
2	G_READ_OWN	1 = GSM read-for-ownership operation is supported	RES	0
3	G_DC_INVALIDATE	1 = GSM data cache invalidate operation is supported	RES	0
4	G_CASTOUT	1 = GSM castout operation is supported	RES	0
5	G_DC_FLUSH	1 = GSM data cache flush is supported	RES	0
6	G_IO_READ	1 = GSM I/O read operation is supported	RES	0
7	G_IC_INVALIDATE	1 = GSM instruction cache invalidate operation is supported	RES	0
8	G_TLB_INVALIDATE	1 = GSM TLB invalidate-entry operation is supported	RES	0
9	G_TLB_SYNC	1 = GSM TLB invalidate entry sync operation is supported	RES	0
10	RIO_RSVD_10	This field is reserved by the *RapidIO Interconnect Specification (Revision 2.1)*. Tsi721 allows this field to be set to support some future and as yet undefined operation.	RES	0
11	RIO_RSVD_11	This field is reserved by the *RapidIO Interconnect Specification (Revision 2.1)*. Tsi721 allows this field to be set to support some future and as yet undefined operation.	RES	0
12	DS_TM	1 = Data streaming traffic management is supported	RES	0
13	DS	1 = Data streaming operation is supported	RES	0

(Continued)

Bits	Name	Description	Type	Reset Value
14:15	IMPLEMENT_DEF	Implementation defined	RES	0
16	READ	1 = Read operation is supported	RES	1
17	WRITE	1 = Write operation is supported	RES	1
18	STRM_WR	1 = Streaming write operation is supported	RES	1
19	WR_RES	1 = Write-with-response operation is supported	RES	1
20	D_MSG	1 = Data messaging is supported	RES	1
21	DBELL	1 = Doorbells are supported	RES	1
22	ACSWAP	1 = Atomic (compare-and-swap) operation is supported	RES	0
23	ATSWAP	1 = Atomic (test-and-swap) operation is supported	RES	0
24	A_INC	1 = Atomic (increment) operation is supported	RES	0
25	A_DEC	1 = Atomic (decrement) operation is supported	RES	0
26	A_SET	1 = Atomic (set) operation is supported	RES	0
27	A_CLEAR	1 = Atomic (clear) operation is supported	RES	0
28	A_SWAP	1 = Atomic (swap) operation is supported	RES	0
29	PORT_WR	1 = Port-write operation is supported	RES	1
30:31	IMPLEMENT_DEF2	Implementation defined	RES	0

18.4.8 RapidIO Processing Element Logical Layer Control CSR

This register defines S-RIO extended address control for the Tsi721. It is a standard S-RIO register.

Register name: RIO_SR_XADDR Reset value: 0x0000_0001	Register offset: 0x0004C

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

Bits	Name	Description	Type	Reset value
31:3	RESERVED	RESERVED	R	0x0
2:0	EA_CTL	Controls the number of address bits for the Tsi721 as a source and as a target: 0b100 = 66-bit addresses 0b010 = 50-bit addresses 0b001 = 34-bit addresses Others = Reserved	R/WS	0x1

18.4.9 RapidIO Base deviceID CSR

This is a global device register that supports ingress packet filtering based on destID. It contains the base deviceID values for the device. A device may have multiple deviceID values but these are not defined in a standard CSR.

Register Name: RIO_BASE_ID Reset Value: Undefined	Register Offset: 0x00060

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

Bits	Name	Description	Type	Reset Value
0:7	Reserved	Reserved	R	0
8:15	BASE_ID	This is the 8-bit base ID of the device in a small common transport system.	R/WS	Undefined
16:31	LAR_BASE_ID	This is the 16-bit base ID of the device in a large common transport system (valid only for endpoint device and if bit 27 of the RapidIO Processing Element Features CAR is set).	R/WS	Undefined

18.4.10 RapidIO Host Base deviceID Lock CSR

This register contains the base deviceID 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 that provides a lock function. Once this field is written, all subsequent writes to the field are ignored except where the value written matches the value contained in the field. In this case, the register is re-initialized to 0xFFFF. After writing to HOST_BASE_ID, a processing element must then read the register 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: 0x00068

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

Bits	Name	Description	Type	Reset Value
0:15	Reserved	Reserved	R	0
16:31	HOST_BASE_ID	Base deviceID for the device that is initializing this endpoint.	R/WS	0xFFFF

18.4.11 RapidIO 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: Undefined	Register Offset: 0x0006C

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

Bits	Name	Description	Type	Reset Value
0:31	CTAG	Component Tag	R/WS	Undefined

18.5 RapidIO 1x or 4x LP-Serial Registers

This section describes the 1x/4x LP-Serial Command and Status Registers (CSR). All registers in the set are 32 bits long and are aligned to a 32-bit boundary. These registers allow an external device to determine the capabilities, configuration, and status of a device using this 1x/4x LP-Serial physical layer. The registers can be accessed using the maintenance operations defined in *Part I: Input/Output Logical Specification*. The registers are located in the 1x/4x LP-Serial physical features block, which is an Extended Features block in the Extended Features Space.

18.5.1 RapidIO 1x or 4x Port Maintenance Block Header

This register contains the block header information.

Register Name: RIO_SP_MB_HEAD Reset Value: 0x1000_0002	Register Offset: 0x00100

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 The reset value points to the next entry in the extended feature list, RapidIO Error Reporting Block Header.	RS	0x1000
16:31	EF_ID	Extended Features ID 0x0002 = Endpoint device with software-assisted error recovery option	RS	0x0002

18.5.2 RapidIO Port Link Timeout Control CSR

This register contains the timeout timer value for the port. This timeout is for link events such as sending a packet to receiving the corresponding acknowledge, and sending a link-request to receiving the corresponding link-response.

This timer provides a heart-beat signal that is scaled from the SRV_CLK, which decrements per packet timers in the PLM (see RapidIO Port IP Prescalar for SRV_CLK Register).

Register Name: RIO_SP_LT_CTL Reset Value: 0xFFFF_FF00	Register Offset: 0x00120

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

Bits	Name	Description	Type	Reset Value
0:23	TVAL	Timeout interval value, is in the range provided by: • Shortest: SRV_CLK * TVAL * 3 • Longest: SRV_CLK * TVAL * 4 0 = Disable the timer	R/WS	0xFFFFFF
24:31	Reserved	Reserved	R	0

Table 84: Port Link Timeout Programming Values

Default Timer Value (uS)				3 to 6 Seconds	
TVAL				FFFFFF	
IP_CLK Frequency (MHz)	IP_CLK Period (uS)	PRESCALAR_SRV_CLK	SRV_CLK period	Shortest	Longest
312.5	0.003	31	0.10	4.99	6.66
307.2	0.003	31	0.10	5.08	6.77
250	0.004	25	0.10	5.03	6.71
245.76	0.004	25	0.10	5.12	6.83
156.25	0.006	16	0.10	5.15	6.87
153.6	0.007	15	0.10	4.92	6.55
125	0.008	13	0.10	5.23	6.98
78.125	0.013	8	0.10	5.15	6.87
76.8	0.013	8	0.10	5.24	6.99
62.5	0.016	6	0.10	4.83	6.44

Note: see Table 92 for Tsi721 IP_CLK frequency

18.5.3 RapidIO Response Timeout Register

This register defines the S-RIO response timeout. It is a standard S-RIO register.

Register name: RIO_SR_RSP_TO Reset value: 0x00FF_FFFF	Register offset: 0x00124

Bits	0	1	2	3	4	5	6	7
00:07	RESERVED							
08:15	RSP_TO							
16:23	RSP_TO							
24:31	RSP_TO							

Bits	Name	Description	Type	Reset value
0:7	RESERVED	RESERVED	R	0x0
8:31	RSP_TO	Response timeout It defines response timeout threshold. The actual timeout range is between 188 ns*(RSP_TO + 1) and 2*188 ns*(RSP_TO + 1). A value of 0 disables the response timeout. Note: The format of this field is not compliant to the *RapidIO Interconnect Specification (Revision 2.1)*.	R/WS	0xFFFFFF

18.5.4 RapidIO Port General Control CSR

Register Name: RIO_SP_GEN_CTL Reset Value: Undefined	Register Offset: 0x0013C

Bits	0	1	2	3	4	5	6	7
00:07	HOST	MAST_EN	DISC	Reserved				
08:15	Reserved							
16:23	Reserved							
24:31	Reserved							

Bits	Name	Description	Type	Reset Value
0	HOST	Host Device A host device performs system exploration, initialization, and maintenance. Agent or slave devices are initialized by host devices. 1 = Host device 0 = Agent or slave device	R/WS	Undefined
1	MAST_EN	Master Enable This controls whether a device can issue requests. If MAST_EN is not set, a device can respond only to requests. 1 = Enabled to issue requests 0 = Not enabled to issue requests	R/WS	Undefined
2	DISC	Discovered The Tsi721 has been located by the device responsible for system configuration. 1 = Tsi721 is discovered by system host 0 = Tsi721 is not discovered	R/WS	Undefined
3:31	Reserved	Reserved	R	0

18.5.5 RapidIO Port Link Maintenance Request CSR

This register is provided for software-assisted error recovery.

Software must not generate a link request/input-status when the link is operating error free. Writing 0b100 to the CMD field when the link is operating error free will result in non-deterministic operation, and may create an unrecoverable error condition.

The Tsi721 expects a link-response for each link-request triggered by the user unless the CMD is reset-device. If a response is not received, the port times out to allow the user to send another link-request. The *RapidIO Interconnect Specification (Revision 2.1)* requires that only one link maintenance request can be outstanding at a time. Software must ensure there are no outstanding link-requests before writing to this register by waiting a period of time equal to five[1] times the Port Link Timeout period (see RapidIO Port Link Timeout Control CSR). link-requests can be outstanding because Tsi721 initiated a link-request or a previous write to this register did not complete (the RapidIO Port Link Maintenance Response CSR indicates the link-request's completion status).

When the CMD field is written with 0b011 (reset-device), the Tsi721 produces four consecutive link-request/reset-device control symbols to trigger a reset of the link partner. Tsi721 does not expect a link-response for link-request/reset-device control symbols.

Register Name: RIO_SP_LM_REQ Reset Value: 0x0000_0000	Register Offset: 0x140

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	Reserved	R	0
29:31	CMD	Command Command to be sent in the link-request control symbol. If read, this field returns the last written value. 0b011 = Reset-device 0b100 = Input-status Other = Reserved Note: Four link-request/reset-request control symbols are sent to the link partner each time 0b011 is written to this field.	R/WS	0

[1] One for the initial link-request plus 4 for the retries performed by Tsi721.

This register should not be written if Tsi721 is processing outbound packets. The user can set the PORT_LOCKOUT bit to disable packet traffic.

If Tsi721 has an outstanding link-request, writes to the CMD field are ignored unless the value is 0b011 (reset-device).

Link-responses from link-requests generated by writing to this register are captured only in the RapidIO Port Link Maintenance Response CSR. The captured ackID is not checked for validity and is not used to accept any outstanding packets or restart transmission of packets. The user must write to the RapidIO Port Local ackID Status CSR to accept or retry packets and restart transmission of packets.

18.5.6 RapidIO Port Link Maintenance Response CSR

This register can be accessed by both the local master and an external RapidIO device. When this register is read it returns the status from the last link-request response. This register is used only for endpoint devices with a software-assisted error recovery option.

Register Name: RIO_SP_LM_RESP Reset Value: 0x0000_0000	Register Offset: 0x144

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 1 = If the link-request causes a link-response, this bit indicates that the link-response has been received and the status fields are valid. If link-request does not cause a link-response, this bit indicates that the link-request 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.	RCS	0
1:20	Reserved	Reserved	R	0
21:26	ACK_ID_STAT	ackID status field from the link-response control symbol. The value of the next ackID expected by the receiver.	RS	0
27:31	LINK_STAT	Link-status field from the link-response control symbol 0b00010 = Error 0b00100 = Retry-stopped 0b00101 = Error-stopped 0b10000 = OK Other = Reserved	RS	0

18.5.7 RapidIO Port Local ackID Status CSR

This register is provided for software-assisted error recovery in the event that ackID synchronization with the link partner has been lost. This register is for software-assisted error recovery and should not be written during normal operation.

Register Name: RIO_SP_ACKID_STAT Reset Value: 0x0000_0000	Register Offset: 0x148

Bits	0	1	2	3	4	5	6	7
00:07	CLR_OUTSTD_ACKID	Reserved	INB_ACKID					
08:15	Reserved							
16:23	Reserved		OUTSTD_ACKID					
24:31	Reserved		OUTB_ACKID					

Bits	Name	Description	Type	Reset Value
0	CLR_OUTSTD_ACKID	Clear Outstanding ackIDs 1 = Discard all outstanding unacknowledged packets and set OUTSTD_ACKID equal to OUTB_ACKID. This bit is 0 when read.	R/W1C	0
1	Reserved	Reserved	R	0
2:7	INB_ACKID	Input port next expected ackID value. Bit 2 is valid only for long control symbols. INB_ACKID indicates the expected value of the ackID field in the next packet to be received. This field can be written to synchronize the Tsi721's expected ackID with the ackID being used by the link partner. INB_ACKID returns the instantaneous value when the LLM processes the read request.	R/W	0
8:17	Reserved	Reserved	R	0
18:23	OUTSTD_ACKID	Output port unacknowledged ackID status. Bit 18 is valid only for long control symbols. OUTSTD_ACKID indicates the ackID value expected in the next acknowledge control symbol to be received. This field is automatically updated to the value of OUTB_ACKID when OUTB_ACKID is written or when CLR_OUTSTD_ACKID is set. Note: After an S-RIO reset or PCIe Hot reset, this field is updated when the Tsi721 receives a packet acknowledge from its link partner.	RS	0

(Continued)

Bits	Name	Description	Type	Reset Value
24:25	Reserved	Reserved	R	0
26:31	OUTB_ACKID	Output port next transmitted Acknowledge ID value. Bit 26 is valid only for long control symbols. OUTB_ACKID indicates the ackID value that the Tsi721 uses for the next packet transmission. This field can be written to force re-transmission of all outstanding unacknowledged packets in order to manually implement error recovery. Notes: • If CLR_OUTSTD_ACKID is not set, all outstanding unacknowledged packets are re-transmitted. Transmission begins using the ackID written to OUTB_ACKID. • If CLR_OUTSTD_ACKID is set, all outstanding unacknowledged packets are discarded. Transmission begins using the ackID written to OUTB_ACKID. • There is some ambiguity in the value of OUTB_ACKID while packets are being transmitted because of the pipelining inside Tsi721. If the pipeline stalls, OUTB_ACKID can indicate an ackID value beyond what has been transmitted and this value may persist in the register for a significant amount of time. OUTB_ACKID provides the correct value when Tsi721's output has stopped. When the output stops, OUTB_ACKID may "back up" when the pipeline state recovers.	R/W	0

⚠ Writing to this register when there are packets being exchanged with a link partner will result in non-deterministic ackID values.

18.5.7.1 Example Operation

The initial condition is as follows:

- Packets with ackID values of 0, 1, 2, 3, 4, 5 have been transmitted
- The packet with ackID = 0 has been acknowledged with a packet-accepted control symbol.

Table 85: Example Operation of CLR_OUTSTD_ACKID, OUTSTD_ACKID, and OUTB_ACKID

CLR_OUTSTD_ACKID	OUTSTD_ACKID	OUTB_ACKID	Notes/Operation
0	1	6	• Initial condition
0	p	p	• Packets previously transmitted with ackIDs of 1–5 are retransmitted. • Packet acknowledgement with ackID=*p* is expected next • Next packet transmitted will have ackID=*p*.
1	q	q[a]	• Packets previously transmitted with ackIDs of 1–5 are discarded. • Packet acknowledgement with ackID=*q* is expected next • Next packet transmitted will have ackID=*q*.

[a] OUTB_ACKID may or may not be written by software. If it is not written, *q* will be “6” in this example: OUTB_ACKID will be unchanged and OUTSTD_ACKID will be automatically updated to 6.

18.5.8 RapidIO Port Control 2 CSR

This register indicates the supported baud rates (GB_XpXX fields) and selects the baud rate for the port (GB_XpXX_EN fields).

Only one of the supported baud rates can be enabled at any one time. As required by the *RapidIO Interconnect Specification (Revision 2.1)*, attempts to enable an unsupported baud rate will be ignored. If more than one baud rate is enabled, the highest enabled baud rate is selected. Changing the selected baud rate automatically causes the link to retrain – it is equivalent to setting the FORCE_REINIT bit in the RapidIO PLM Port Implementation Specific Control Register.

This register is accessed when a local processor or an external device wants to examine or configure the Tsi721's port baud rate information.

Register Name: RIO_SP_CTL2 Reset Value: Undefined	Register Offset: 0x154

Bits	0	1	2	3	4	5	6	7
00:07	BAUD_SEL				BAUD_DISC	Reserved	GB_1p25	GB_1p25_EN
08:15	GB_2p5	GB_2p5_EN	GB_3p125	GB_3p125_EN	GB_5p0	GB_5p0_EN	GB_6p25	GB_6p25_EN
16:23	Reserved							
24:31	Reserved				INACT_EN	D_SCRM_DIS	RTEC	RTEC_EN

Bits	Name	Description	Type	Reset Value
0:3	BAUD_SEL	Indicates the initialized baud rate of the port. 0b0000 = No rate selected 0b0001 = 1.25 GBaud 0b0010 = 2.5 GBaud 0b0011 = 3.125 GBaud 0b0100 = 5.0 GBaud 0b0101 = 6.25 GBaud Others = Reserved	RS	Undefined
4	BAUD_DISC	Indicates whether automatic baud rate discovery is supported. 0 = Automatic baud rate discovery not supported 1 = Automatic baud rate discovery is supported	RS	0
5	Reserved	Reserved	R	0
6	GB_1p25	Indicates whether port operation at 1.25 GBaud is supported. 0 = 1.25 GBaud operation is not supported 1 = 1.25 GBaud operation is supported	RS	0x1

(Continued)

Bits	Name	Description	Type	Reset Value
7	GB_1p25_EN	Controls whether port operation at 1.25 GBaud is enabled. 0 = Disable 1.25 GBaud operation 1 = Enable 1.25 GBaud operation The port must not allow this bit to be set unless GB_1p25 is 1. One of the GB_XpXX_EN bits must be set at any time.	R/WS	Undefined
8	GB_2p5	Indicates whether port operation at 2.5 GBaud is supported. 0 = 2.5 GBaud operation is not supported 1 = 2.5 GBaud operation is supported	RS	0x1
9	GB_2p5_EN	Controls whether port operation at 2.5 GBaud is enabled. 0 = Disable 2.5 GBaud operation 1 = Enable 2.5 GBaud operation The port must not allow this bit to be set unless GB_2p5 is 1. One of the GB_XpXX_EN bits must be set at any time.	R/WS	Undefined
10	GB_3p125	Indicates whether port operation at 3.125 GBaud is supported. 0 = 3.125 GBaud operation is not supported 1 = 3.125 GBaud operation is supported	RS	0x1
11	GB_3p125_EN	Controls whether port operation at 3.125 GBaud is enabled. 0 = Disable 3.125 GBaud operation 1 = Enable 3.125 GBaud operation The port must not allow this bit to be set unless GB_3p125 is 1. One of the GB_XpXX_EN bits must be set at any time.	R/WS	Undefined
12	GB_5p0	Indicates whether port operation at 5.0 GBaud is supported. 0 = 5.0 GBaud operation is not supported 1 = 5.0 GBaud operation is supported	RS	0x1
13	GB_5p0_EN	Controls whether port operation at 5.0 GBaud is enabled. 0 = Disable 5.0 GBaud operation 1 = Enable 5.0 GBaud operation The port must not allow this bit to be set unless GB_5p0 is 1. One of the GB_XpXX_EN bits must be set at any time.	R/WS	Undefined
14	GB_6p25	Indicates whether port operation at 6.25 GBaud is supported. 0 = 6.25 GBaud operation is not supported 1 = 6.25 GBaud operation is supported (Tsi721 does not support 6.25 GBaud operation)	RS	Undefined

(Continued)

Bits	Name	Description	Type	Reset Value
15	GB_6p25_EN	Controls whether port operation at 6.25 GBaud is enabled. 0 = Disable 6.25 GBaud operation 1 = Enable 6.25 GBaud operation The port must not allow this bit to be set unless GB_6p25 is 1. One of the GB_XpXX_EN bits must be set at any time.	R/WS	Undefined
16:27	Reserved	Reserved	R	0
28	INACT_EN	0 = Lanes assigned to the port but not used by the port have their outputs disabled 1 = Lanes assigned to the port but not used by the port have their transmitter and receiver enabled This RapidIO 2.1 standard bit is for test/debug purposes only.	R/WS	0
29	D_SCRM_DIS	Data scrambling disable 0 = Enable transmit data scrambler and receive data descrambler 1 = Disable transmit data scrambler and receive data descrambler for control and packet data characters. Control symbol and packet data characters are neither scrambled in the transmitter before transmission nor descrambled in the receiver when they are received. The transmit scrambler remains enabled for the generation of pseudo-random data characters for the IDLE2 random data field. This bit is for test use only and must not be asserted during normal operation.	R/WS	0
30	RTEC	Indicates whether the port can transmit commands to control the transmit emphasis in the connected port. 0 = The port does not support remote transmit emphasis adjustment in the connected port 1 = The port supports remote transmit emphasis adjustment in the connected port	RS	0
31	RTEC_EN	Controls whether the port can adjust the transmit emphasis in the connected port. 0 = Remote transmit emphasis control is disabled 1 = Remote transmit emphasis control is enabled The port must not let this bit be set unless remote transmit emphasis control is supported and the link to which the port is connected is using idle sequence 2 (IDLE2).	RS	0

18.5.9 RapidIO Port Error and Status CSR

This register is accessed when a local processor or an external device wants to examine Tsi721's port error and status information. It returns 0x0000001 if it is read when the port is powered down.

Register Name: RIO_SP_ERR_STAT Reset Value: 0x0000_0001	Register Offset: 0x158

Bits	0	1	2	3	4	5	6	7
00:07	IDLE2	IDLE2_EN	IDLE_SEQ	Reserved	TXFC	OUTPUT_DROP	OUTPUT_FAIL	OUTPUT_DEGR
08:15	Reserved			OUTPUT_RE	OUTPUT_R	OUTPUT_RS	OUTPUT_ERR_ENCTR	OUTPUT_ERR_STOP
16:23	Reserved					INPUT_RS	INPUT_ERR_ENCTR	INPUT_ERR_STOP
24:31	Reserved			PORT_W_P	PORT_UNAVL	PORT_ERR	PORT_OK	PORT_UNIT

Bits	Name	Description	Type	Reset Value
0	IDLE2	Indicates whether the port supports IDLE2 for baud rate of less than 5.5 GBaud. 0 = IDLE2 is not supported for baud rates < 5.5 GBaud 1 = IDLE2 is supported for baud rates < 5.5 GBaud	RS	0
1	IDLE2_EN	Controls whether IDLE2 is enabled for baud rates of less than 5.5 GBaud. 0 = IDLE2 is disabled for baud rates < 5.5 GBaud	R	0
2	IDLE_SEQ	Indicates which idle sequence is active. 0 = IDLE1 is active. 1 = IDLE2 is active.	R	0
3	Reserved	Reserved	R	0
4	TXFC	Indicates the physical level flow control currently in use on the link. 0 = Receiver-based flow control 1 = Transmitter-based flow control	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
5	OUTPUT_DROP	Output Packet-Dropped 1 = Output port discarded a packet. This bit is set when the port's OUTPUT_FAIL bit is set and RapidIO Port Control CSR.DROP_EN bit is set (see also Table 86). Note: A soft reset is required before clearing this field (for more information, see How to Re-enable the S-RIO Port?).	R/W1CS	0
6	OUTPUT_FAIL	Output Failed-Encountered 1 = Output port encountered a failed condition, meaning that the port's error counter in the RapidIO Port Error Rate CSR has reached the failed threshold in the RapidIO Port Error Rate Threshold CSR. The status of this bit is duplicated in RapidIO PLM Port Event Status Register.OUTPUT_FAIL.	R/W1CS	0
7	OUTPUT_DEGR	Output Degraded-Encountered 1 = Output port encountered a degraded condition, meaning that the port's error counter in the RapidIO Port Error Rate CSR has reached the degraded threshold in the RapidIO Port Error Rate Threshold CSR. The status of this bit is duplicated in RapidIO PLM Port Event Status Register.OUTPUT_DEGR.	R/W1CS	0
8:10	Reserved	Reserved	R	0
11	OUTPUT_RE	Output Retry-Encountered 1 = Outbound port encountered a retry condition. Set when Output Retry-Stopped bit is set.	R/W1CS	0
12	OUTPUT_R	Output Retried 1 = Outbound port received a packet-retry control symbol and cannot make forward progress. This bit is set when Output Retry-stopped bit is set, and cleared after receiving a packet-accepted or packet-not-accepted control symbol.	R	0
13	OUTPUT_RS	Output Retry-Stopped 1 = Outbound port received a packet-retry control symbol and is in the output retry-stopped state	R	0
14	OUTPUT_ERR_ENCTR	Output Error-Encountered 1 = Outbound port encountered (and possibly recovered from) a transmission error. This bit is set when output error-stopped bit is set.	R/W1CS	0
15	OUTPUT_ERR_STOP	Output Error-Stopped 1 = Outbound port is in the output error-stopped state.	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
16:20	Reserved	Reserved	R	0
21	INPUT_RS	Input Retry-Stopped 1 = Inbound port is in the input retry-stopped state.	R	0
22	INPUT_ERR_ENCTR	Input error-encountered 1 = Inbound port encountered (and possibly recovered from) a transmission error. This bit is set when the input error-stopped bit is set.	R/W1CS	0
23	INPUT_ERR_STOP	Input Error-Stopped 1 = Inbound port is in the input error-stopped state.	R	0
24:26	Reserved	Reserved	R	0
27	PORT_W_P	Port-write Pending 1 = Port encountered a condition that 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/W1CS	0
28	PORT_UNAVL	Indicates Port Availability 0 = The port is available for use. 1 = The port is not available for use.	RS	0
29	PORT_ERR	Port Error 1 = Inbound or Outbound port encountered an error from which hardware was unable to recover (a fatal error). Write 1 to clear this bit. This bit reports the failure of the Tsi721 to recover from following fatal errors: • Four link-request attempts each receiving a link-response with an ackID that is not outstanding • Four link-request attempts with timeout error for link-response • Dead Link detected The status of this bit is duplicated in RapidIO PLM Port Event Status Register.PORT_ERR.	R/W1CS	0

(Continued)

Bits	Name	Description	Type	Reset Value
30	PORT_OK	Port OK Inbound and Outbound ports are initialized and can exchange error-free control symbols. This bit and PORT_UNIT are mutually exclusive. This bit indicates the state variable *link_initialized* defined by the *RapidIO Interconnect Specification (Revision 2.1)*	R	0
31	PORT_UNIT	Port Uninitialized Inbound and Outbound ports are not initialized. This bit and PORT_OK are mutually exclusive. This bit is set to a 1 after reset. This bit is the complement of the state variable *port_initialized* defined by the *RapidIO Interconnect Specification (Revision 2.1).*	R	1

18.5.10 RapidIO Port Control CSR

This register returns a default value of 0x00000001 when read in power-down mode.

Register Name: RIO_SP_CTL Reset Value: Undefined	Register Offset: 0x15C

Bits	0	1	2	3	4	5	6	7
00:07	PORT_WIDTH		INIT_PWIDTH			OVER_PWIDTH		
08:15	PORT_DIS	OTP_EN	INP_EN	ERR_DIS	MULT_CS	FLOW_CTRL	ENUM_B	FLOW_ARB
16:23	OVER_PWIDTH2		PORT_WIDTH2		Reserved			
24:31	Reserved				STOP_FAIL_EN	DROP_EN	PORT_LOCKOUT	PTYP

Bits	Name	Description	Type	Reset Value
0:1	PORT_WIDTH	Indicates port width modes supported by the port. This field is used in conjunction with the PORT_WIDTH2 field of this register. The bits of these two fields collectively indicate the port width modes supported by the port. 0b00 = No support for 2x or 4x 0b01 = No support for 2x; support for 4x 0b10 = Support for 2x; no support for 4x 0b11 = Support for 2x and 4x Note: 1x is supported by all ports.	RS	0b11
2:4	INIT_PWIDTH	Width of the port after initialization. This field is set by hardware when the initialization process is complete. 0b000 = Single-lane port, lane 0 0b001 = Single-lane port, lane R (redundancy lane) 0b010 = 4x lane port 0b011 = 2x lane port Other values are reserved. This field is invalid if RapidIO Port Error and Status CSR.PORT_UNIT is set.	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
5:7	OVER_PWIDTH	Software port configuration to override the hardware size. 0b000 = No override 0b001 = Reserved 0b010 = Force single-lane port, lane 0 0b011 = Force single-lane port, lane R (redundancy port) 0b100 = Reserved 0b101 = 2x mode enabled, 4x mode disabled 0b110 = 4x mode enabled, 2x mode disabled 0b111 = 2x and 4x modes enabled The software port configuration can be completed as follows: • Power-up option during bootload process Changed during the normal mode of operation. Note that after the port width is changed, the link automatically re-initializes to ensure that the link partner is aware of the change. For more information on link initialization, see FORCE_REINIT in RapidIO PLM Port Implementation Specific Control Register.	R/WS	0b000
8	PORT_DIS	Port disable 0 = Enable port receivers/drivers 1 = Disable port receivers/drivers; they are unable to receive or transmit any packets or control symbols. When the port is disabled, no data flow to output drivers. Note: When the port is disabled, the port’s transmit lanes are electrically idle and its receive lanes are ignored. Note: A soft reset is required before clearing this field (for more information, see How to Re-enable the S-RIO Port?).	R/WS	0
9	OTP_EN	Output port transmit enable 0 = Port is stopped and unable to issue any packets except to route or respond to MAINTENANCE packets. In this mode, Tsi721 is limited to issuing only Type 8 packets. 1 = Port is enabled to issue any packets This bit is set to 0 after reset and must be set to 1 by software (1 after reset must be for host only). The Port stopped feature is supported after reset in conjunction with INP_EN = 0.	R/WS	0

(Continued)

Bits	Name	Description	Type	Reset Value
10	INP_EN	Inbound port enable 0 = Inbound port is stopped and enabled only to respond to I/O logical MAINTENANCE requests. Other packets generate packet-not-accepted control symbols to force an error condition to be signaled by the sending device. 1 = Enable inbound port to respond to any packet. This bit is set to 0 after reset and must be set to 1 by software.	R/WS	0
11	ERR_DIS	Error Checking Disable. (Physical Layer only) 0 = Enable error checking and recovery. 1 = Disable error checking and recovery. For Tsi721 to operate properly, this bit must be set to 0.	R/WS	0
12	MULT_CS	Multicast-event Participant 1 = Port can forward MECSs with *CMD* field value set to 0 (see also RapidIO PLM Port MECS Forwarding Register).	R/WS	0
13	FLOW_CTRL	Enable flow control transactions 0 = Do not route or issue flow-control transactions to the port 1 = Route or issue flow-control transactions to the port Note: This bit does not control any functionality within the Tsi721.	R/WS	0
14	ENUM_B	Enumeration boundary bit This is used in system discovery algorithms. This bit does not control any functionality within the Tsi721.	R/WS	Undefined
15	FLOW_ARB	Enable flow arbitration transactions 0 = Do not route or issue flow arbitration transactions to the port 1 = Route or issue flow arbitration transactions to the port Note: This bit does not control any functionality within the Tsi721.	R/WS	0
16:17	OVER_PWIDTH2	This bit field applies to a port that supports 8- and 16- lanes. For the Tsi721 this field is read-only.	R	0
18:19	PORT_WIDTH2	This bit field applies to a port that supports 8- and 16- lanes. For Tsi721 this field is read-only.	R	0
20:27	Reserved	Reserved	R	0
28	STOP_FAIL_EN	This bit is used with the DROP_EN bit to force certain behavior when the Error Rate Failed Threshold has been met or exceeded (see Table 86).	R/WS	0
29	DROP_EN	This bit is used with the STOP_FAIL_EN bit to force certain behavior when the Error Rate Failed Threshold has been met or exceeded. (see Table 86).	R/WS	0

(Continued)

Bits	Name	Description	Type	Reset Value
30	PORT_LOCKOUT	When cleared, the packets that can be received and issued are controlled by the state of the Output Port Transmit Enable and Input Port Enable bits. When set, the port is stopped and is not enabled to receive or issue any packets; the input port can still send and respond to link-requests; all received packets return packet-not-accepted control symbols to force an error condition to be signaled by the sending device. Note: A soft reset is required before clearing this field (for more information, see How to Re-enable the S-RIO Port?). Note: Packet-not-accepted (PNA) is returned only if the Tsi721 is not in the Input-retry-stopped or Input-error-stopped states. When the Tsi721 returns packet-not-accepted, it enters the Input-error-stopped state. The cause field of the PNA is set to "General Error" (0b11111).	R/WS	0
31	PTYP	This indicates the port type. 0b1 = Serial port	RS	1

Table 86: Port Behavior When Error Rate Threshold Has Been Reached/Exceeded

Stop on Port Failed Encountered (STOP_FAIL_EN)	Drop Packet Enable (DROP_EN)	Port Behavior
0	0	The port must continue to attempt to transmit packets to the connected device if the RapidIO Port Error and Status CSR.OUTPUT_FAIL bit is set and/or if the Error Rate Failed threshold has been met or exceeded.
0	1	The port must discard packets that receive a packet-not-accepted control symbol when the Error Rate Failed Threshold has been met or exceeded. When a packet is discarded, the port must set the Output Packet-dropped bit in the RapidIO Port Error and Status CSR. If the output port "heals", the Error Rate Counter falls below the Error Rate Failed Threshold, the output port must continue to attempt to forward all packets.
1	0	The port must stop attempting to send packets to the connected device when the RapidIO Port Error and Status CSR.OUTPUT_FAIL bit is set. The output port will congest.
1	1	The port must discard all output packets without attempting to send when the port's RapidIO Port Error and Status CSR.OUTPUT_FAIL bit is set. When a packet is discarded, the port must set Output Packet-dropped bit in the RapidIO Port Error and Status CSR.

18.6 RapidIO Extended Error Management Registers

This section describes the Extended Features block (EF_ID = 0x0007) that allows an external device to manage the error status and reporting in the Tsi721.

18.6.1 RapidIO Error Reporting Block Header

This register contains the block header information.

Register Name: RIO_ERR_RPT_BH Reset Value: 0x3000_0007	Register Offset: 0x01000

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 The reset value points to the next entry in the extended feature list, RapidIO LP-Serial Per-Lane Extended Features Block Header.	RS	0x3000
16:31	EF_ID	Hard-wired Extended Features ID 0x0007 = Extended features ID for Error Management Capability	RS	0x0007

This document is confidential and is subject to an NDA.

18.6.2 RapidIO Logical/Transport Layer Error Detect CSR

Register Name: RIO_ERR_DET Reset Value: 0x0000_0000	Register Offset: 0x01008

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

Bits	Name	Description	Type	Reset Value
0:30	Reserved	Reserved	R	0
31	IMP	IDT Implementation-Specific Error was Detected 1 = SR2PC detected an S-RIO error (S-RIO I/O error or messaging error) or PCIe data link down. Software should write this bit only during software debug. Hardware may immediately clear this bit after a software write. This bit is cleared when the following conditions are *all met*; otherwise, it is set: • If any bit in SR2PC Port-Write Enable CSR is high, its associated bit in SR2PC Channel Interrupt Register {0..7} is not set • All associated outbound message channels with S-RIO message errors and port-write enabled (see Messaging Engine Port-Write Enable Register) are reset • All associated inbound message channel with S-RIO message request timeout and port-write enabled (see Messaging Engine Port-Write Enable Register) have SRTO bit of Inbound DMA Channel Interrupt Register {0..7} cleared • If UNS_RSP_EN bit in Messaging Engine Port-Write Enable Register is high and the UNS_RSP bit in Messaging Engine Interrupt Register is cleared	R/W	0

18.6.3 RapidIO Logical/Transport Layer Error Enable CSR

This register controls whether or not an error condition locks the RapidIO Logical/Transport Layer Error Detect CSR and Capture registers (see registers starting at RapidIO Logical/Transport Layer High Address Capture CSR), and is reported to the system host through a port-write request.

Register Name: RIO_ERR_EN Reset Value: 0x0000_0000	Register Offset: 0x0100C

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

Bits	Name	Description	Type	Reset Value
0:30	Reserved	Reserved	R	0
31	IMP_EN	1 = Implementation specific enable of RapidIO Logical/Transport Layer Error Detect CSR.IMP.	R/WS	0

18.6.4 RapidIO Logical/Transport Layer High Address Capture CSR

This register contains error information. It is locked and valid on detecting the first S-RIO I/O errors from a received S-RIO packet with associated port-write enable set high, that is, when bits of registers SR2PC General Interrupt CSR and SR2PC Port-Write Enable CSR satisfy the following condition: (RSP_ERR and RSP_ERR_EN) | (ILL_DEC and ILL_DEC_EN) | (ILL_TARGET and ILL_TARGET_EN) | (UNS_RSP and UNS_RSP_EN) is high. This register is unlocked when the above condition becomes false.

The contents of this register are valid only when the Tsi721 is using 50- or 66-bit addressing (for more information, see RapidIO Logical/Transport Layer Address Capture CSR).

Register Name: RIO_H_ADDR_CAPT Reset Value: 0x0000_0000	Register Offset: 0x01010

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

Bits	Name	Description	Type	Reset Value
0:31	ADDR[0:31]	Most significant bits of address associated with the error (for requests, not applicable for responses). Address bits not used by the device are set to zero. Valid only when SR2PC has detected an S-RIO error.	R/WS	0

18.6.5 RapidIO Logical/Transport Layer Address Capture CSR

This register contains error information. It is locked and valid on detecting the first S-RIO I/O errors from a received S-RIO packet with associated port-write enable set high, that is, when bits of registers SR2PC General Interrupt CSR and SR2PC Port-Write Enable CSR satisfy the following condition: (RSP_ERR & RSP_ERR_EN) | (ILL_DEC & ILL_DEC_EN) | (ILL_TARGET & ILL_TARGET_EN) | (UNS_RSP & UNS_RSP_EN) is high. This register is unlocked when the above condition becomes false.

The RapidIO Logical/Transport Layer Error Detect CSR and the Logical/Transport Layer Error Capture Registers are writable by software to allow software debug of the system error recovery mechanisms. For software debug, software must write to the Logical/Transport Layer Error Capture registers with the desired address and deviceID information, then write to the RapidIO Logical/Transport Layer Error Detect CSR to set an error bit and lock the registers. When an error detect bit is set, hardware informs the system software of the error using its standard error reporting mechanism. After the error is reported, system software can read and clear registers as required to complete its error handling protocol testing.

Register Name: RIO_ADDR_CAPT Reset Value: 0x0000_0000	Register Offset: 0x01014

Bits	7	6	5	4	3	2	1	0
00:07	ADDR[32:39]							
08:15	ADDR[40:47]							
16:23	ADDR[48:55]							
24:31	ADDR[56:60]					Reserved	XAMSBS	

Bits	Name	Description	Type	Reset Value
0:28	ADDR[32:60]	Least significant 29 bits of address associated with the error (for requests, not applicable for responses)	R/WS	0
29	Reserved	Reserved	R	0
30:31	XAMSBS	Extended address bits of the address associated with the error (for requests, not applicable for responses)	R/WS	0

18.6.6 RapidIO Logical/Transport Layer deviceID Capture CSR

This register contains error information. It is locked and valid on detecting the first S-RIO I/O errors from a received S-RIO packet with associated port-write enable set high, that is, when bits of registers SR2PC General Interrupt CSR and SR2PC Port-Write Enable CSR satisfy the following condition: (RSP_ERR & RSP_ERR_EN) | (ILL_DEC & ILL_DEC_EN) | (ILL_TARGET & ILL_TARGET_EN) | (UNS_RSP & UNS_RSP_EN) is high. This register is unlocked when the above condition becomes false.

The RapidIO Logical/Transport Layer Error Detect CSR and the Logical/Transport Layer Error Capture Registers are writable by software to allow software debug of the system error recovery mechanisms. For software debug, software must write to the Logical/Transport Layer Error Capture registers with the desired address and deviceID information, then write to the RapidIO Logical/Transport Layer Error Detect CSR to set an error bit and lock the registers. When an error detect bit is set, hardware informs the system software of the error using its standard error reporting mechanism. After the error is reported, system software can read and clear registers as required to complete its error handling protocol testing.

Register Name: RIO_ID_CAPT Reset Value: 0x0000_0000	Register Offset: 0x01018

Bits	7	6	5	4	3	2	1	0
00:07	MSB_DEST_ID							
08:15	DEST_ID							
16:23	MSB_SRC_ID							
24:31	SRC_ID							

Bits	Name	Description	Type	Reset Value
0:7	MSB_DEST_ID	Most significant byte of destID associated with the error (Large transport system only). Valid only when SR2PC has detected an S-RIO error.	R/WS	0
8:15	DEST_ID	destID associated with the error. Valid only when SR2PC has detected an S-RIO error.	R/WS	0
16:23	MSB_SRC_ID	Most significant byte of srcID associated with the error (Large transport system only). Valid only when SR2PC has detected an S-RIO error.	R/WS	0
24:31	SRC_ID	srcID associated with the error. Valid only when SR2PC has detected an S-RIO error.	R/WS	0

18.6.7 RapidIO Logical/Transport Layer Control Capture CSR

This register contains error information. It is locked and valid on detecting the first S-RIO I/O errors from a received S-RIO packet with associated port-write enable set high, that is, when bits of registers SR2PC General Interrupt CSR and SR2PC Port-Write Enable CSR satisfy the following condition: (RSP_ERR and RSP_ERR_EN) | (ILL_DEC and ILL_DEC_EN) | (ILL_TARGET and ILL_TARGET_EN) | (UNS_RSP and UNS_RSP_EN) is high. This register is unlocked when the above condition becomes false.

The RapidIO Logical/Transport Layer Error Detect CSR and the Logical/Transport Layer Error Capture Registers are writable by software to allow software debug of the system error recovery mechanisms. For software debug, software must write to the Logical/Transport Layer Error Capture registers with the desired address and deviceID information, then write to the RapidIO Logical/Transport Layer Error Detect CSR to set an error bit and lock the registers. When an error detect bit is set, hardware informs the system software of the error using its standard error reporting mechanism. After the error is reported, system software can read and clear registers as required to complete its error handling protocol testing.

Register Name: RIO_CTRL_CAPT Reset Value: 0x0000_0000	Register Offset: 0x0101C

Bits	7	6	5	4	3	2	1	0
00:07	FTYPE				TTYPE			
08:15	MESSAGE_INFO							
16:23	Reserved							
24:31	Reserved							

Bits	Name	Description	Type	Reset Value
0:3	FTYPE	ftype associated with the error. Valid only when SR2PC has detected an S-RIO error.	R/WS	0
4:7	TTYPE	ttype associated with the error. Valid only when SR2PC has detected an S-RIO error.	R/WS	0
8:15	MESSAGE_INFO	N/A	R/WS	0
16:31	Reserved	Reserved	R	0

18.6.8 RapidIO Port-Write Target deviceID CSR

This register contains the target deviceID to be used when a device generates a maintenance port-write operation to report errors to a system host.

Register Name: RIO_PW_TGT_ID Reset Value: 0x0000_0000	Register Offset: 0x01028

Bits	0	1	2	3	4	5	6	7
00:07	MSB_PW_ID							
08:15	PW_TGT_ID							
16:23	LRG_TRANS	Reserved						
24:31	Reserved							

Bits	Name	Description	Type	Reset Value
0:7	MSB_PW_ID	Most significant byte of port-write Target deviceID (Large transport system only)	R/WS	0
8:15	PW_TGT_ID	Port-Write Target deviceID This field determines the destID of port-writes generated by the Tsi721. Note: The srcID of port-writes is controlled by the RapidIO Base deviceID CSR.	R/WS	0
16	LRG_TRANS	deviceID size to use for a port-write 0 = Use the small transport deviceID 1 = Use the large transport deviceID	R/WS	0
17:31	Reserved	Reserved	R	0

18.6.9 RapidIO Port Error Detect CSR

This register indicates the physical layer errors that have been detected by the port hardware since the register was last cleared. The register is cleared by software by writing 0x0000_0000 to the register. Note: errors reported here are not detected nor are specifics captured in the RapidIO Port Attributes Capture CSR (and related registers) unless the port is in the *link_initialized* state (see RapidIO Port Error and Status CSR.PORT_OK).

Register Name: RIO_SP_ERR_DET Reset Value: 0x0000_0000	Register Offset: 0x1040

Bits	0	1	2	3	4	5	6	7
00:07	IMP_SPEC	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	DSCRAM_LOS	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	Tsi721 Implementation-specific Error. Setting this bit allows implementation specific events to contribute to the “leaky bucket” for Port Degraded and Port Failed event detection. The implementation specific error is: • Too many packet denials event, when enabled in the RapidIO PLM Port Packet Denial Control Register. The packet that caused the too many packet denials event is captured.	R/WS	0
1:8	Reserved	Reserved	R	0
9	CS_CRC_ERR	0b1 = Received a control symbol with a bad CRC	R/WS	0
10	CS_ILL_ID	0b1 = Received an acknowledge control symbol with an unexpected ackID (packet-accepted or packet_retry)	R/WS	0
11	CS_NOT_ACC	0b1 = Received a packet-not-accepted control symbol	R/WS	0
12	PKT_ILL_ACKID	0b1 = Received a packet with an unexpected ackID	R/WS	0
13	PKT_CRC_ERR	0b1 = Received a packet with a bad CRC	R/WS	0
14	PKT_ILL_SIZE	0b1 = Received a packet that exceeded 276 bytes	R/WS	0
15:16	Reserved	Reserved	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
17	DSCRAM_LOS	0b1 = Loss of receiver descrambler synchronization when control symbol and packet data are scrambled before transmission. No information is captured in RapidIO Port Packet/Control Symbol Error Capture CSR 0, 1, 2, and 3.	R/WS	0
18:25	Reserved	Reserved	R	0
26	LR_ACKID_ILL	0b1 = Received a link-response with an ackID that is not outstanding	R/WS	0
27	PROT_ERR	Protocol Error 0b1 = Received an unexpected packet or control symbol	R/WS	0
28	Reserved	Reserved	R	0
29	DELIN_ERR	Delineation Error 0b1 = Received an unaligned /SC/or/PD/, Illegal character, or Invalid character	R/WS	0
30	CS_ACK_ILL	0b1 = Received an unexpected acknowledge control symbol	R/WS	0
31	LINK_TO	0b1 = Did not receive an acknowledge or link-response within the specified timeout interval.	R/WS	0

18.6.10 RapidIO Port Error Rate Enable CSR

This register contains bits that when set cause specific detected errors to increment the error rate counter in the RapidIO Port Error Rate CSR, and capture information about the error in and then lock the RapidIO Port Packet/Control Symbol Error Capture CSR 0 (and related CSRs). Each bit of this register controls the capture and counting of the detected error whose occurrence is indicated by the equivalent bit in the RapidIO Port Error Detect CSR.

Register Name: RIO_SP_RATE_EN Reset Value: 0x0000_0000	Register Offset: 0x1044

Bits	0	1	2	3	4	5	6	7
00:07	IMP_SPEC_EN	Reserved						
08:15	Reserved	CS_CRC_ERR_EN	CS_ILL_ID_EN	CS_NOT_ACC_EN	PKT_ILL_ACKID_EN	PKT_CRC_ERR_EN	PKT_ILL_SIZE_EN	Reserved
16:23	Reserved	DSCRAM_LOS_EN	Reserved					
24:31	Reserved		LR_ACKID_ILL_EN	PROT_ERR_EN	Reserved	DELIN_ERR_EN	CS_ACK_ILL_EN	LINK_TO_EN

Bits	Name	Description	Type	Reset Value
0	IMP_SPEC_EN	Enable error rate counting of Implementation-specific error Setting this bit allows implementation specific events to contribute to the “leaky bucket” for Port Degraded and Port Failed event detection. The implementation specific error: • Too many packet denials event, when enabled in the RapidIO PLM Port Packet Denial Control Register.	R/WS	0
1:8	Reserved	Reserved	R	0
9	CS_CRC_ERR_EN	0b1 = Enable the capture of received a corrupt control symbol, and enable error counter increment due to this error.	R/WS	0
10	CS_ILL_ID_EN	0b1 = Enable the capture of received a acknowledge control symbol with an unexpected ackID (packet-accepted or packet_retry), and enable error counter increment due to this error.	R/WS	0
11	CS_NOT_ACC_EN	0b1 = Enable the capture of received a packet-not-accepted control symbol, and enable error counter increment due to this error.	R/WS	0
12	PKT_ILL_ACKID_EN	0b1 = Enable the capture of received a packet with a bad ackID, and enable error counter increment due to this error.	R/WS	0

(Continued)

Bits	Name	Description	Type	Reset Value
13	PKT_CRC_ERR_EN	0b1 = Enable the capture of received a packet with bad CRC, and enable error counter increment due to this error.	R/WS	0
14	PKT_ILL_SIZE_EN	0b1 = Enable the capture of an illegal packet size (a packet that is longer than 276 bytes), and enable error counter increment due to this error.	R/WS	0
15:16	Reserved	Reserved	R	0
17	DSCRAM_LOS_EN	0b1 Enable error rate counting of the loss of receiver descrambler synchronization when control symbol and packet data is being scrambled before transmission.	R/WS	0
18:25	Reserved	Reserved	R	0
26	LR_ACKID_ILL_EN	0b1 = Enable the capture of a non-outstanding ackID due to a link-response, and enable error counter increment due to this error.	R/WS	0
27	PROT_ERR_EN	0b1 = Enable the capture of protocol errors (received control symbol is unexpected), and enable error counter increment due to this error.	R/WS	0
28	Reserved	Reserved	R	0
29	DELIN_ERR_EN	0b1 = Enable the capture of delineation errors, and enable error counter increment due to this error (delineated error).	R/WS	0
30	CS_ACK_ILL_EN	0b1 = Enable the capture of unsolicited acknowledgement control symbol, and enable error counter increment due to this error.	R/WS	0
31	LINK_TO_EN	0b1 = Enable the capture of link timeout errors, and enable error counter increment due to this error.	R/WS	0

18.6.11 RapidIO Port Attributes Capture CSR

This register indicates the type of information contained in the error capture registers, which start at RapidIO Port Packet Error Capture CSR 1. If multiple errors are detected during the same clock cycle, one of the errors will be indicated in the Error type field. The error that is indicated is selected in this priority order of bits set in RapidIO Port Error Detect CSR (from highest to lowest):

1. CS_CRC_ERR
2. CS_ILL_ID
3. CS_NOT_ACC
4. LP_ACKID_ILL
5. PROT_ERR
6. CS_ACK_ILL
7. PKT_ILL
8. PKT_ERR
9. PKT_ILL_SIZE
10. LINK_TO
11. DELIN

The RapidIO Port Error Detect CSR and the RapidIO Port Packet Error Capture CSR 1 are also writable by software to allow software debug of the system error recovery and threshold method. For debug, software must write to this register to set VAL_CAPT after writing the packet/control symbol information in the other capture registers. Each write of a non-zero value to the RapidIO Port Error Detect CSR must cause the Error Rate Counter to increment if the corresponding error bit is enabled in the RapidIO Port Error Rate Enable CSR. When a threshold is reached, the hardware informs the system software of the error using its standard error reporting method. After the error is reported, system software can read and clear registers as required to complete its error handling protocol testing.

Register Name: RIO_SP_ERR_ATTR_CAPT Reset Value: 0x0000_0000	Register Offset: 0x1048

Bits	0	1	2	3	4	5	6	7
00:07	INFO_TYPE			ERR_TYPE				
08:15	IMPL_DEP							
16:23	IMPL_DEP							
24:31	IMPL_DEP				Reserved			VAL_CAPT

Bits	Name	Description	Type	Reset Value
0:2	INFO_TYPE	Type of information logged 0b000 = Packet 0b001 = Reserved 0b010 = Short delimited[a] control symbol 0b011 = Long delimited[a] control symbol 0b100–0b111 = Reserved	R/WS	0

(Continued)

Bits	Name	Description	Type	Reset Value
3:7	ERR_TYPE	Encoded value of captured error bit in the RapidIO Port Error Detect CSR that describes the error captured in the RapidIO Port Packet/Control Symbol Error Capture CSR 0, 1, 2, and 3.	R/WS	0
8:27	IMPL_DEP	Implementation Dependent Error Information. Bits 8–23 indicate whether a 8b/10b character is a Control (K) character or a Data (D) character. A "control" value of 1 indicates a K character and 0 indicates a Data character. • Bits 8–11 "control" bits for RapidIO Port Packet/Control Symbol Error Capture CSR 0. • Bits 12–15 "control" bits for RapidIO Port Packet Error Capture CSR 1 • Bits 16–19 "control" bits for RapidIO Port Packet Error Capture CSR 2 • Bits 20–23 "control" bits for RapidIO Port Packet Error Capture CSR 3. • Bits 24–26 are unused. • Bit 27 indicates the type of timeout on a link-response timeout: 0 = Packet acknowledgement timeout 1 = Link-response timeout RapidIO Port Packet/Control Symbol Error Capture CSR 0, 1, 2, and 3 encode data and special characters as follows: • Data characters are captured "as is" with their IMPL_DEP bit = 0. • Special characters have their data portion captured with their IMPL_DEP bit =1. • Invalid characters are captured as 0xFF with their IMPL_DEP bit = 1. • When less than 16 bytes are captured, unused characters are captured as 0x00 with their IMPL_DEP bit = 1.	R/WS	0
28:30	Reserved	Reserved	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
31	VAL_CAPT	Capture valid info This bit is set by hardware to indicate that RapidIO Port Packet/Control Symbol Error Capture CSR 0, 1, 2, and 3 contain valid information. For some types of errors and capture information, not all of the capture registers contain useful information: • For short control symbols, only capture register 0 will contain meaningful information. • For long control symbols, only capture registers 0 and 1 will contain meaningful information. • For DELIN and DSCRAM_LOS, nothing is captured. • For IMP_SPEC-max-denial, the packet that was denied is captured • For packet capture, the capture registers are used to capture either the first 16 bytes of the packet or the full packet if it is shorter than 16 bytes. The software writes 0 to clear this bit which subsequently unlocks all the capture registers of the port.	R/WS	0

a. A “delimited” control symbol is the combination of a control symbol and its delimiting special character(s).

18.6.12 RapidIO Port Packet/Control Symbol Error Capture CSR 0

For more information about this register, see RapidIO Port Error Rate Enable CSR and RapidIO Port Attributes Capture CSR.

This register provides storage for capturing packets and long/short control symbols. For short control symbols, the full 32 bits are used because the delimited[a] control symbol is captured. RapidIO PLM Port Implementation Specific Control Register.PAYL_CAP can modify which bytes of the packet are captured.

The RapidIO Port Error Detect CSR and the RapidIO Port Packet/Control Symbol Error Capture CSR 0–3, are writable by software to allow debug of the system error recovery and threshold method. For debug, software must write the RapidIO Port Attributes Capture CSR to set the VAL_CAPT bit after writing the packet/control symbol information in the other capture registers.

Register Name: RIO_SP_ERR_CAPT_0 Reset Value: 0x0000_0000	Register Offset: 0x104C

Bits	0	1	2	3	4	5	6	7
00:07	CAPT_0							
08:15	CAPT_0							
16:23	CAPT_0							
24:31	CAPT_0							

Bits	Name	Description	Type	Reset Value
0:31	CAPT_0	Control symbol or Bytes 0 to 3 of packet header	R/WS	0

18.6.13 RapidIO Port Packet Error Capture CSR 1

For more information about this register, see RapidIO Port Error Rate Enable CSR and RapidIO Port Attributes Capture CSR.

This register provides storage for capturing packets and long control symbols. For long control symbols, the full 32 bits are used because the delimited[a] control symbol is captured.

The RapidIO Port Error Detect CSR and the Port Error Capture registers are writable by software to allow debug of the system error recovery and threshold method. For debug, software must write the RapidIO Port Attributes Capture CSR to set the VAL_CAPT bit after writing the packet/control symbol information in the other capture registers.

Register Name: RIO_SP_ERR_CAPT_1 Reset Value: 0x0000_0000	Register Offset: 0x1050

Bits	0	1	2	3	4	5	6	7
00:07	CAPT_1							
08:15	CAPT_1							
16:23	CAPT_1							
24:31	CAPT_1							

Bits	Name	Description	Type	Reset Value
0:31	CAPT_1	Bytes 4 to 7 of the packet or the long control symbol.	R/WS	0

18.6.14 RapidIO Port Packet Error Capture CSR 2

For more information about this register, see RapidIO Port Error Rate Enable CSR and RapidIO Port Attributes Capture CSR.

The RapidIO Port Error Detect CSR and the Port Error Capture registers are writable by software to allow debug of the system error recovery and threshold method. For debug, software must write the RapidIO Port Attributes Capture CSR to set the VAL_CAPT bit after writing the packet/control symbol information in the other capture registers.

Register Name: RIO_SP_ERR_CAPT_2 Reset Value: 0x0000_0000	Register Offset: 0x1054

Bits	0	1	2	3	4	5	6	7
00:07	CAPT_2							
08:15	CAPT_2							
16:23	CAPT_2							
24:31	CAPT_2							

Bits	Name	Description	Type	Reset Value
0:31	CAPT_2	Bytes 8 to 11 of the packet header	R/WS	0

18.6.15 RapidIO Port Packet Error Capture CSR 3

For more information about this register, see RapidIO Port Error Rate Enable CSR and RapidIO Port Attributes Capture CSR.

The RapidIO Port Error Detect CSR and the Port Error Capture registers are writable by software to allow debug of the system error recovery and threshold method. For debug, software must write the RapidIO Port Attributes Capture CSR to set the VAL_CAPT bit after writing the packet/control symbol information in the other capture registers.

Register Name: RIO_SP_ERR_CAPT_3 Reset Value: 0x0000_0000	Register Offset: 0x1058

Bits	0	1	2	3	4	5	6	7
00:07	CAPT_3							
08:15	CAPT_3							
16:23	CAPT_3							
24:31	CAPT_3							

Bits	Name	Description	Type	Reset Value
0:31	CAPT_3	Bytes 12 to 15 of the packet header	R/WS	0

18.6.16 RapidIO Port Error Rate CSR

This register is used with the RapidIO Port Error Rate Threshold CSR to monitor and control the reporting of port physical layer errors that are detected in RapidIO Port Error Detect CSR and enabled in RapidIO Port Error Rate Enable CSR.

Register Name: RIO_SP_ERR_RATE Reset Value: 0x8000_0000	Register Offset: 0x1068

Bits	0	1	2	3	4	5	6	7
00:07	ERR_RB							
08:15	Reserved						ERR_RR	
16:23	PEAK							
24:31	ERR_RATE_CNT							

Bits	Name	Description	Type	Reset Value
0:7	ERR_RB	Error Rate Bias value. This value is calculated from the SRV_CLK period. Bias values are provided for SRV_CLK period of 100 ns. This value sets the “leak rate” for the leaky bucket. 0x00 = Do not decrement error rate counter 0x01 = Decrement every SRV_CLK period X 10e+04 (1 ms) 0x02 = Decrement every SRV_CLK period X 10e+05 (10 ms) 0x04 = Decrement every SRV_CLK period X 10e+06 (100 ms) 0x08 = Decrement every SRV_CLK period X 10e+07 (1 s) 0x10 = Decrement every SRV_CLK period X 10e+08 (10 s) 0x20 = Decrement every SRV_CLK period X 10e+09 (100 s) 0x40 = Decrement every SRV_CLK period X 10e+10 (1,000 s) 0x80 = Decrement every SRV_CLK period X 10e+11 (10,000 s) All other values = Reserved For more information, see RapidIO Port IP Prescalar for SRV_CLK Register.	R/WS	0x80
8:13	Reserved	Reserved	R	0

(Continued)

<table>
<tr><th>Bits</th><th>Name</th><th>Description</th><th>Type</th><th>Reset Value</th></tr>
<tr><td>14:15</td><td>ERR_RR</td><td>Error Rate Recovery
These bits limit the incrementing of the error rate counter above the failed threshold trigger.
0b00 = 2 error above RapidIO Port Error Rate Threshold CSR [ERR_RFT]
0b01 = 4 error above RapidIO Port Error Rate Threshold CSR[ERR_RFT]
0b10 = 16 errors above RapidIO Port Error Rate Threshold CSR[ERR_RFT]
0b11 = Permits RapidIO Port Error Rate Threshold CSR[ERR_RFT] to saturate</td><td>R/WS</td><td>0</td></tr>
<tr><td>16:23</td><td>PEAK</td><td>This value contains the peak value attained by the error rate counter.</td><td>R/WS</td><td>0</td></tr>
<tr><td>24:31</td><td>ERR_RATE_CNT</td><td>This field contains a count of the number of physical layer errors that have been detected by the port, decremented by the Error Rate Bias method, to create an indication of the physical layer error rate.
The counter cannot overflow (saturates at the maximum value 0xFF) or underflow (decrements to a minimum value 0x0), and continues to increment or decrement as defined even if thresholds are met or exceeded, limited by the ERR_RR setting as follows:
<table>
<tr><th>ERR_RR</th><th>ERR_CNT Frozen Value</th></tr>
<tr><td>0b00</td><td>2 + RapidIO Port Error Rate Threshold CSR [ERR_RFT]</td></tr>
<tr><td>0b01</td><td>4 + RapidIO Port Error Rate Threshold CSR [ERR_RFT]</td></tr>
<tr><td>0b10</td><td>16 + RapidIO Port Error Rate Threshold CSR [ERR_RFT]</td></tr>
<tr><td>0b11</td><td>0xFF</td></tr>
</table>
If the value of the counter reaches the degraded and failed thresholds specified by the RapidIO Port Error Rate Threshold CSR, the events are reported through the RapidIO Port Error and Status CSR and the RapidIO PLM Port Event Status Register.</td><td>R/WS</td><td>0</td></tr>
</table>

18.6.17 RapidIO Port Error Rate Threshold CSR

This register and the RapidIO Port Error Rate CSR monitor and control the reporting of transmission errors.

Register Name: RIO_SP_ERR_THRESH Reset Value: 0xFFFF_0000	Register Offset: 0x106C

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

Bits	Name	Description	Type	Reset Value
0:7	ERR_RFT	Error Rate Failed Threshold Trigger These bits provide the threshold value for reporting an error condition due to a possible broken link. 0x00 = Disable the trigger 0x01 = Set the error reporting threshold to 1 0x02 = Set the error reporting threshold to 2 ... 0xFF = Set the error reporting threshold to 255 If the error rate threshold is triggered then RapidIO Port Error and Status CSR[OUTPUT_FAIL] is set to 1. Note: IDT recommends a threshold value of 4 so that link-request timeouts, which only occur four times before halting, trigger a failed link notification.	R/WS	0xFF
8:15	ERR_RDT	Error Rate Degraded Threshold Trigger These bits provide the threshold value for reporting an error condition due to a degrading link. 0x00 = Disable the trigger 0x01 = Set the error reporting threshold to 1 0x02 = Set the error reporting threshold to 2 ... 0xFF = Set the error reporting threshold to 255 If the error rate threshold is triggered then RapidIO Port Error and Status CSR[OUTPUT_DEGR] is set to 1.	R/WS	0xFF
16:31	Reserved	Reserved	R	0

18.7 LP-Serial Lane Extended Features Block

This register block contains information about each lane associated with the port. This block is optional in the *RapidIO Interconnect Specification (Revision 2.1)*.

18.7.1 RapidIO LP-Serial Per-Lane Extended Features Block Header

This register contains the block header information about the per-lane serial registers.

Register Name: RIO_PER_LANE_BH Reset Value: 0x0000_000D	Register Offset: 0x03000

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 The reset value points to the next entry in the extended feature list. A value of 0 indicates that this is the last extended feature block.	RS	0
16:31	EF_ID	Hard-wired extended features ID 0x000D = Extended features ID for LP-Serial Per-Lane Register Block	RS	0x000D

18.7.2 RapidIO Lane {0..3} Status 0 CSR

This register provides per lane status information. It is used only when IDLE2 is supported.

Register Name: RIO_LANE{0..3}_STAT0 Reset Value: Undefined	Register Offset: 0x03010 += 20

Bits	0	1	2	3	4	5	6	7
00:07	PORT_NUM							
08:15	LANE_NUM				TX_TYPE	TX_MODE	RX_TYPE	
16:23	RX_INV	RX_TRN	RX_SYNC	RX_RDY	ERR_CNT			
24:31	CHG_SYNC	CHG_TRN	Reserved		STAT1	STAT2_7		

Bits	Name	Description	Type	Reset Value
0:7	PORT_NUM	Port number associated with this lane.	RS	0
8:11	LANE_NUM	• Lane within the port that this lane supports. The lane within the path independent of lane swapping. The numbering of the lanes is: • For a 4x port, the lanes A, B, C, and D are numbered 0, 1, 2 and 3 respectively • For a 2x port using lanes A and B, lanes A and B are numbered 0 and 1 respectively. • For a 2x port using lanes C and D, lanes C and D are numbered 0 and 1 respectively. • For 1x ports, all lanes are numbered 0.	RS	Undefined
12	TX_TYPE	Transmitter Type 1 = Long reach	RS	1
13	TX_MODE	Current Transmitter Operating Mode 1 = Long reach	RS	1
14:15	RX_TYPE	Receiver Type 0b01 = Medium run	RS	0b01

(Continued)

Bits	Name	Description	Type	Reset Value
16	RX_INV	Receiver Input Inverted This bit indicates whether the lane receiver has detected that the polarity of its input signal is inverted and has inverted its receiver input to correct the polarity. 0 = Receiver input not inverted 1 = Receiver input inverted Note: Tsi721 does not perform automatic polarity detection and correction. This field is always zero. Manual polarity inversion can be configured in RapidIO PLM Port Lane Polarity Control Register.RXn_POL.	R	0
17	RX_TRN	Receiver Trained Always 1 since the Tsi721 does not support adaptive equalization.	RS	1
18	RX_SYNC	Indicates whether the receiver has achieved lane synchronization. 0 = Not in sync 1 = Receiver has achieved lane synchronization according to the *RapidIO Interconnect Specification (Revision 2.1)*.	R	0
19	RX_RDY	Receiver Ready 1 = Adaptive equalization adjustment is completed and RX_SYNC is achieved. This bit always has the same value as RX_SYNC since the Tsi721 does not support adaptive equalization.	R	0
20:23	ERR_CNT	Count of the number of 8b/10b decoding errors that have occurred since the last time this register was read. The error counter hits a limit at 15; in other words, it does not roll over.	RCS	Undefined
24	CHG_SYNC	Change in the RX_SYNC state since the last time this register was read 0 = No change in RX_SYNC state 1 = The RX_SYNC state changed since the last time this register was read.	RCS	0
25	CHG_TRN	Change in the RX_TRN value since the last time this register was read.	R	0
26:27	Reserved	Reserved	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
28	STAT1	Indicates the existence of the STAT1 register (RapidIO Lane {0..3} Status 1 CSR). The RapidIO Lane {0..3} Status 1 CSR is useful only if the application uses IDLE2/long control symbols.	RES	1
29:31	STAT2_7	Indicates the existence of the STAT2 to STAT7 registers. 0b0 = None of the Lane Status CSRs 2–7 are implemented	RES	0

18.7.3 RapidIO Lane {0..3} Status 1 CSR

This register contains information about the link partner that is collected from the CS markers and CS fields of the IDLE2 sequence received by the local lane n receiver. Only information from error-free CS markers and CS fields is reported in this register. This register is used only when IDLE2 is supported.

Register Name: RIO_LANE{0..3}_STAT1 Reset Value: 0x0000_0000	Register Offset: 0x03014 += 20

Bits	0	1	2	3	4	5	6	7
00:07	IDLE2	INFO_OK	CHG	IMPL_SPEC	LP_RX_TRN	LP_WIDTH		
08:15	LP_LANE_NUM				LP_TAP_M1		LP_TAP_P1	
16:23	LP_SCRM	Reserved						
24:31	Reserved							

Bits	Name	Description	Type	Reset Value
0	IDLE2	Indicates whether IDLE2 sequence is being received. 0 = No IDLE2 sequence has been received since the last time the bit was cleared. 1 = IDLE2 sequence has been received at some time since the bit was cleared.	R/W1CS	0
1	INFO_OK	Indicates whether the IDLE2 information latched in this register is current. 0 = Information is stale. Set when errors have been detected in the last IDLE2 sequence, or the lane_sync signal has flickered since the last CS Marker and CS Field were received. 1 = Information is current. Information is from the last IDLE2 sequence received, no errors have been detected, and the lane_sync signal has remained asserted since the last CS Marker and CS Field were received.	R	0
2	CHG	Indicates whether the value of any of the other bits in this register have changed since the last time the register was read. 0 = Values are unchanged since the last read 1 = At least one value changed since the last read	RC	0
3	IMPL_SPEC	Implementation-specific meaning, according to the link partners definition.	R	0
4	LP_RX_TRN	Indicates whether the link partner's receiver is trained 0 = Receiver is not trained 1 = Receiver is trained	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
5:7	LP_WIDTH	Link Partner Port Width 0b000 = 1x mode 0b001 = 2x mode 0b010 = 4x mode0b011–0b111 = Reserved	R	0
8:11	LP_LANE_NUM	Link Partner Lane Number status The lane within the link partner's port that this lane is associated with. 0b0000 = Lane 0 0b0001 = Lane 1 0b0010 = Lane 2 0b0011 = Lane 3	R	0x0
12:13	LP_TAP_M1	Link Partner Tap Minus 1 status 0b00 = Tap(-1) is not implemented 0b01 = Tap(-1) at minimum emphasis 0b10 = Tap(-1) at maximum emphasis 0b11 = Tap(-1) at intermediate emphasis setting	R	0
14:15	LP_TAP_P1	Link Partner Tap Plus 1 status 0b00 = Tap(+1) is not implemented 0b01 = Tap(+1) at minimum emphasis 0b10 = Tap(+1) at maximum emphasis 0b11 = Tap(+1) at intermediate emphasis setting	R	0
16	LP_SCRM	Link partner data scrambling/descrambling setting 0 = Scrambling/descrambling is disabled 1 = Scrambling/descrambling is enabled	R	0
17:31	Reserved	Reserved	R	0

18.8 RapidIO Implementation-Specific Registers

This register block is not defined in the *RapidIO Interconnect Specification (Revision 2.1)*, and is specific to Tsi721.

18.9 PLM Register Block

18.9.1 IDT-Specific Physical Layer Block Header Register

This register identifies the following registers as the *IDT PLM Register Block*.

Register Name: RIO_PLM_BH Reset Value: 0x0103_0000	Register Offset: 0x10000

Bits	0	1	2	3	4	5	6	7
00:07	NEXT_BLK_PTR							
08:15	NEXT_BLK_PTR							
16:23	BLK_REV				BLK_TYPE			
24:31	BLK_TYPE							

Bits	Name	Description	Type	Reset Value
0:15	NEXT_BLK_PTR	Pointer to the next IDT block, in units of 0x100 bytes.	RS	0x0103
16:19	BLK_REV	IDT-defined revision of the block type in this device.	RS	0
20:31	BLK_TYPE	IDT-defined identifier for the register block type.	RS	0

18.9.2 RapidIO PLM Port Implementation Specific Control Register

This register provides implementation-specific control implementation of the port.

Register Name: RIO_PLM_SP_IMP_SPEC_CTL Reset Value: Undefined	Register Offset: 0x10080

Bits	0	1	2	3	4	5	6	7
00:07	PAYL_CAP	Reserved		DLB_EN	Reserved	FORCE_REINIT	SOFT_RST_PORT	TX_BYPASS
08:15	LLB_EN	RESET_REG	PORT_SELF_RST	SELF_RST	SWAP_TX		SWAP_RX	
16:23	DLT_THRESH							
24:31	DLT_THRESH							

Bits	Name	Description	Type	Reset Value
0	PAYL_CAP	Payload Capture. Setting this bit changes the behavior of the RapidIO Port Packet/Control Symbol Error Capture CSR 0 and subsequent capture registers. 0 = Normal behavior as defined by *RapidIO Interconnect Specification (Revision 2.1)* 1 = Similar behavior as defined by *RapidIO Interconnect Specification (Revision 2.1)* except that bytes 6, 7, 8, and 9 are not captured.	R/WS	0
1:2	Reserved	Reserved	R/WS	0
3	DLB_EN	Digital Equipment Loopback Mode for Port Digital equipment loopback mode connects the Tx data flow to the Rx data flow before the 8b/10b encoder/decoder. 0 = Normal operation 1 = Enable loopback for the port	R/WS	0
4	Reserved	Reserved	R	0
5	FORCE_REINIT	Force Link Re-initialization Process This bit is set by software to force a link re-initialization, and is reset automatically when the link re-initialization process begins.	R/W1S	0
6	SOFT_RST_PORT	Software reset control for the port. This bit resets the S-RIO port logic but does not reset the configuration registers. 0 = Normal mode of operation 1 = Hold port in reset	R/WS	0
7	TX_BYPASS	Bypass the transmitter clock crossing FIFO.	R/WS	0

(Continued)

Bits	Name	Description	Type	Reset Value
8	LLB_EN	Line Loopback Mode Line Loopback mode connects the Rx data flow to the Tx data flow. 0 = Normal operation 1 = Enable loopback for the port Caution: IDT does not recommend using this function because CDR and elastic buffering from receive to transmit are not available.	R/WS	0
9	RESET_REG	When PORT_SELF_RST and SELF_RST are configured to reset a single port, this register controls whether the registers in the port are reset. 0 = Registers are not reset 1 = Registers are reset. Resetting of sticky registers is controlled by RapidIO Register Reset Control CSR.CLEAR_STICKY.	R/WS	0
10	PORT_SELF_RST	If SELF_RST is 0, PORT_SELF_RST determines the port's behavior when the port receives a reset request. 0 = Do not reset the port. Set the RapidIO PLM Port Event Status Register.RST_REQ bit. 1 = Reset the port. All state machines of the port are reset to their original power-on states, but no registers are reset. The RapidIO PLM Port Event Status Register.RST_REQ bit is not set. If SELF_RST is 0, receipt of a reset request causes the bit associated with the port to be set in the RapidIO Event Management Reset Request Port Status Register. Notification of the reset request is controlled by the RapidIO Event Management Reset Request Interrupt Enable Register and RapidIO Event Management Reset Request Port-Write Enable Register. Note: Resetting the port causes it to revert to its default port width, baud rate, and idle sequence. As a result, the external device issuing the reset request may need to adjust its settings to recover communication.	R/WS	0
11	SELF_RST	SELF_RST determines the port's behavior when the port receives a reset request. 0 = Handle the reset request according to PORT_SELF_RST 1 = Reset the Tsi721. All state machines and the non-sticky registers are reset to their original power-on states. If the RapidIO Register Reset Control CSR.CLEAR_STICKY bit is set to 1, the sticky bits are also reset.	R/WS	0

(Continued)

Bits	Name	Description	Type	Reset Value
12:13	SWAP_TX	Software control for transmitter lane swap functionality for the port. This field indicates the sampled value of the SP_TX_SWAP signal. 0b00 = Lanes 0, 1, 2, 3 steered to 0, 1, 2, 3 – no reversal 0b01 = Lanes 0, 1, 2, 3 steered to 1, 0, 3, 2 – 2x lane reversal 0b10 = Lanes 0, 1, 2, 3 steered to 3, 2, 1, 0 – 4x lane reversal 0b11 = Lanes 0, 1, 2, 3 steered to 2, 3, 0, 1 – combination reversal This field only controls the lanes associated with this register's port: • The lanes of 1x ports are unaffected by this register • The lanes of 2x ports are only reversed by the 2x lane reversal and only the two lanes of the port are reversed The lanes of 4x ports are affected by all settings of this register	R/WS	Undefined
14:15	SWAP_RX	Software control for receiver lane swap functionality for the port. This field indicates the sampled value of the SP_RX_SWAP signal. 0b00 = Lanes 0, 1, 2, 3 steered to 0, 1, 2, 3 – no reversal 0b01 = Lanes 0, 1, 2, 3 steered to 1, 0, 3, 2 – 2x lane reversal 0b10 = Lanes 0, 1, 2, 3 steered to 3, 2, 1, 0 – 4x lane reversal 0b11 = Lanes 0, 1, 2, 3 steered to 2, 3, 0, 1 – combination reversal This field controls only the lanes associated with this register's port: • The lanes of 1x ports are unaffected by this register • The lanes of 2x ports are only reversed by the 2x lane reversal and only the two lanes of the port are reversed • The lanes of 4x ports are affected by all settings of this register.	R/WS	Undefined
16:31	DLT_THRESH	Sets the threshold for the dead link timer. Each time RapidIO Port Error and Status CSR.PORT_OK is cleared, the counter is reloaded from this register and starts to count down. If the count reaches 0 and the link has not initialized (that is PORT_OK has not been set), the link is declared dead. • All packets are discarded from the transmit queue • All packets sent to the port from the fabric are silently dropped until RapidIO PLM Port Event Status Register.DLT is cleared. • Packets in the receive queue can drain. Packets can be received from the link partner. The duration of the dead link timer is computed as: 2^8 * DLT_THRESH *SRV_CLK period (for more information, see RapidIO Port IP Prescalar for SRV_CLK Register). If DLT_THRESH = 0, then the DEAD LINK TIMER event is disabled. Programming DLT_THRESH causes the counter to load and then if PORT_OK is clear, the counter starts decrementing.	R/WS	0

18.9.3 RapidIO PLM Port Event Status Register

This register reports events detected by the PLM that are not reported in registers defined by the *RapidIO Interconnect Specification (Revision 2.1)*. Any bit that is set in this register and correspondingly enabled in the RapidIO PLM Port Interrupt Enable Register causes the port's bit in the RapidIO Event Management Interrupt Port Status Register to be set. Any bit that is set in this register and correspondingly enabled in the RapidIO PLM Port Port-Write Enable Register causes the port's bit in the RapidIO Event Management Port-Write Port Status Register to be set.

A portion of this register is placed into the implementation-dependent bits of port-write packets. Note that interrupts can be asserted and port-writes can be sent for the same event in this register. In this case, software must avoid race conditions when reading this register and clearing the events.

Register Name: RIO_PLM_SP_STATUS Reset Value: 0x0000_0000	Register Offset: 0x10090

Bits	0	1	2	3	4	5	6	7
00:07	MAX_DENIAL	Reserved		LINK_INIT	DLT	PORT_ERR	OUTPUT_FAIL	OUTPUT_DEGR
08:15	Reserved							RST_REQ
16:23	PBM_PW	TLM_PW	Reserved	MECS	PBM_INT	TLM_INT	Reserved	
24:31	Reserved							

Bits	Name	Description	Type	Reset Value
0	MAX_DENIAL	Maximum Denial Error Set when number of consecutive packet acknowledgement control symbols which were only packet-retry and/or packet-not-accepted has reached DENIAL_THRESHOLD in the RapidIO PLM Port Packet Denial Control Register. An interrupt is generated if MAX_DENIAL is set in the RapidIO PLM Port Interrupt Enable Register. A port-write request can also be generated if enabled.	R/W1CS	0
1:2	Reserved	Reserved	R	0
3	LINK_INIT	Link Initialization Notification Once set, the LINK_INIT bit is cleared by writing 1 to it. When the RapidIO Port Control CSR.PORT_LOCKOUT bit is 1, and the RapidIO Port Error and Status CSR.PORT_OK bit is 1, then LINK_INIT is set to 1. To stop the LINK_INIT bit from being set when PORT_OK is 1, set PORT_LOCKOUT to 0.	R/W1CS	0

(Continued)

Bits	Name	Description	Type	Reset Value
4	DLT	Dead Link Timer Event 1 = The port detected that its link partner was removed. Note: A soft reset is required before clearing this field (for more information, see How to Re-enable the S-RIO Port?).	R/W1CS	0
5	PORT_ERR	This bit indicates the status and operation of the RapidIO Port Error and Status CSR.PORT_ERR bit. Clearing this bit also clears the RapidIO Port Error and Status CSR.PORT_ERR bit.	R/W1CS	0
6	OUTPUT_FAIL	This bit indicates the status and operation of the RapidIO Port Error and Status CSR.OUTPUT_FAIL bit. Clearing this bit also clears RapidIO Port Error and Status CSR.OUTPUT_FAIL.	R/W1CS	0
7	OUTPUT_DEGR	This bit indicates the status and operation of the RapidIO Port Error and Status CSR.OUTPUT_DEGR bit. Clearing this bit also clears RapidIO Port Error and Status CSR.OUTPUT_DEGR.	R/W1CS	0
8:14	Reserved	Reserved	R	0
15	RST_REQ	Inbound Reset Request Received The port received a reset request from the link partner, and the port is configured to not perform a reset. For more information, see SELF_RST and PORT_SELF_RST in the RapidIO PLM Port Implementation Specific Control Register.	R/W1CS	0
16	PBM_PW	The Physical Buffer Module detected an event that uses port-write notification (see RapidIO PBM Port Status Register and RapidIO PBM Port Port-Write Enable Register).	R	0
17	TLM_PW	The Transport Layer Module detected an event that uses port-write notification (see RapidIO TLM Port Status Register and RapidIO TLM Port Port-Write Enable Register).	R	0
18	Reserved	Reserved	R	0
19	MECS	The port received a Multicast-Event Control Symbol (MECS) with *CMD* field identified in the RapidIO PLM Port Received MECS Status Register. Note: This bit is informational; it does not raise an event request.	R/W1CS	0

(Continued)

Bits	Name	Description	Type	Reset Value
20	PBM_INT	The Physical Buffer Module detected an event that uses interrupt notification (see RapidIO PBM Port Status Register and RapidIO PBM Port Interrupt Enable Register).	R	0
21	TLM_INT	The Transport Layer Module detected an event that uses interrupt notification (see RapidIO TLM Port Status Register and RapidIO TLM Port Interrupt Enable Register).	R	0
22:31	Reserved	Reserved	R	0

18.9.4 RapidIO PLM Port Interrupt Enable Register

This register enables events reported in the RapidIO PLM Port Event Status Register to contribute to the port's interrupt request.

Register Name: RIO_PLM_SP_INT_ENABLE Reset Value: 0x0000_0000	Register Offset: 0x10094

Bits	0	1	2	3	4	5	6	7
00:07	MAX_DENIAL	Reserved		LINK_INIT	DLT	PORT_ERR	OUTPUT_FAIL	OUTPUT_DEGR
08:15	Reserved							
16:23	Reserved							
24:31	Reserved							

Bits	Name	Description	Type	Reset Value
0	MAX_DENIAL	1 = Enable interrupt notification for MAX_DENIAL events.	R/WS	0
1:2	Reserved	Reserved	R	0
3	LINK_INIT	1 = Enable interrupt notification for LINK_INIT events.	R/WS	0
4	DLT	1 = Enable interrupt notification for DLT events.	R/WS	0
5	PORT_ERR	1 = Enable interrupt notification for PORT_ERR events.	R/WS	0
6	OUTPUT_FAIL	1 = Enable interrupt notification for OUTPUT_FAIL events.	R/WS	0
7	OUTPUT_DEGR	1 = Enable interrupt notification for OUTPUT_DEGR events.	R/WS	0
8:31	Reserved	Reserved	R	0

18.9.5 RapidIO PLM Port Port-Write Enable Register

This register enables events reported in the RapidIO PLM Port Event Status Register to contribute to the port's port-write request.

Register Name: RIO_PLM_SP_PW_ENABLE Reset Value: 0x0000_0000	Register Offset: 0x10098

Bits	0	1	2	3	4	5	6	7
00:07	MAX_DENIAL	Reserved		LINK_INIT	DLT	PORT_ERR	OUTPUT_FAIL	OUTPUT_DEGR
08:15	Reserved							
16:23	Reserved							
24:31	Reserved							

Bits	Name	Description	Type	Reset Value
0	MAX_DENIAL	1 = Enable port-write notification for MAX_DENIAL events.	R/WS	0
1:2	Reserved	Reserved	R	0
3	LINK_INIT	1 = Enable port-write notification for LINK_INIT events.	R/WS	0
4	DLT	1 = Enable port-write notification for DLT events.	R/WS	0
5	PORT_ERR	1 = Enable port-write notification for PORT_ERR events.	R/WS	0
6	OUTPUT_FAIL	1 = Enable port-write notification for OUTPUT_FAIL events.	R/WS	0
7	OUTPUT_DEGR	1 = Enable port-write notification for OUTPUT_DEGR events.	R/WS	0
8:31	Reserved	Reserved	R	0

18.9.6 RapidIO PLM Port Event Generate Register

This register generates PLM events for software verification.

Register Name: RIO_PLM_SP_EVENT_GEN Reset Value: 0x0000_0000	Register Offset: 0x1009C

Bits	0	1	2	3	4	5	6	7
00:07	MAX_DENIAL	Reserved		LINK_INIT	DLT	PORT_ERR	OUTPUT_FAIL	OUTPUT_DEGR
08:15	Reserved							RST_REQ
16:23	Reserved							
24:31	Reserved							

Bits	Name	Description	Type	Reset Value
0	MAX_DENIAL	Writing 1 to this bit creates a MAX_DENIAL event.	R/W1S	0
1:2	Reserved	Reserved	R	0
3	LINK_INIT	Writing 1 to this bit creates a LINK_INIT event.	R/W1S	0
4	DLT	Writing 1 to this bit creates a DLT event.	R/W1S	0
5	PORT_ERR	Writing 1 to this bit creates a PORT_ERR event. Note that this sets the RapidIO Port Error and Status CSR.PORT_ERR bit. For the behavior of the port when PORT_ERR is asserted, see the definition in the RapidIO Port Error and Status CSR.	R/W1S	0
6	OUTPUT_FAIL	Writing 1 to this bit creates a OUTPUT_FAIL event. Note that this sets the RapidIO Port Error and Status CSR.OUTPUT_FAIL bit. For the behavior of the port when OUTPUT_FAIL is asserted, see the definition in RapidIO Port Error and Status CSR.	R/W1S	0
7	OUTPUT_DEGR	Writing 1 to this bit creates a OUTPUT_DEGR event.	R/W1S	0
8:14	Reserved	Reserved	R	0
15	RST_REQ	Writing 1 to this bit creates a RST_REQ event. Setting this event is equivalent to receiving a reset request from the port. The reset request is handled as programmed in the RapidIO PLM Port Implementation Specific Control Register SELF_RST and PORT_SELF_RST bits.	R/W1S	0
16:31	Reserved	Reserved	R	0

18.9.7 RapidIO PLM Port All Interrupts Enable Register

This register provides interrupt enable control for the port.

Register Name: RIO_PLM_SP_ALL_INT_EN Reset Value: 0x0000_0000	Register Offset: 0x100A0

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

Bits	Name	Description	Type	Reset Value
0:30	Reserved	Reserved	R	0
31	IRQ_EN	Interrupt Error Reporting Enable 0 = Disable event notification using interrupts for the port. 1 = Enable interrupt reporting using interrupts for the port.	R/WS	0

18.9.8 RapidIO PLM Port All Port-Writes Enable Register

This register provides port-write enable control for the port.

Register Name: RIO_PLM_SP_ALL_PW_EN Reset Value: 0x0000_0001	Register Offset: 0x100A4

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

Bits	Name	Description	Type	Reset Value
0:30	Reserved	Reserved	R	0
31	PW_EN	Port-Write Error Reporting Enable 0 = Disable event notification using port-writes for the port. 1 = Enable interrupt reporting using port-writes for the port.	R/WS	1

18.9.9 RapidIO PLM Port Discovery Timer Register

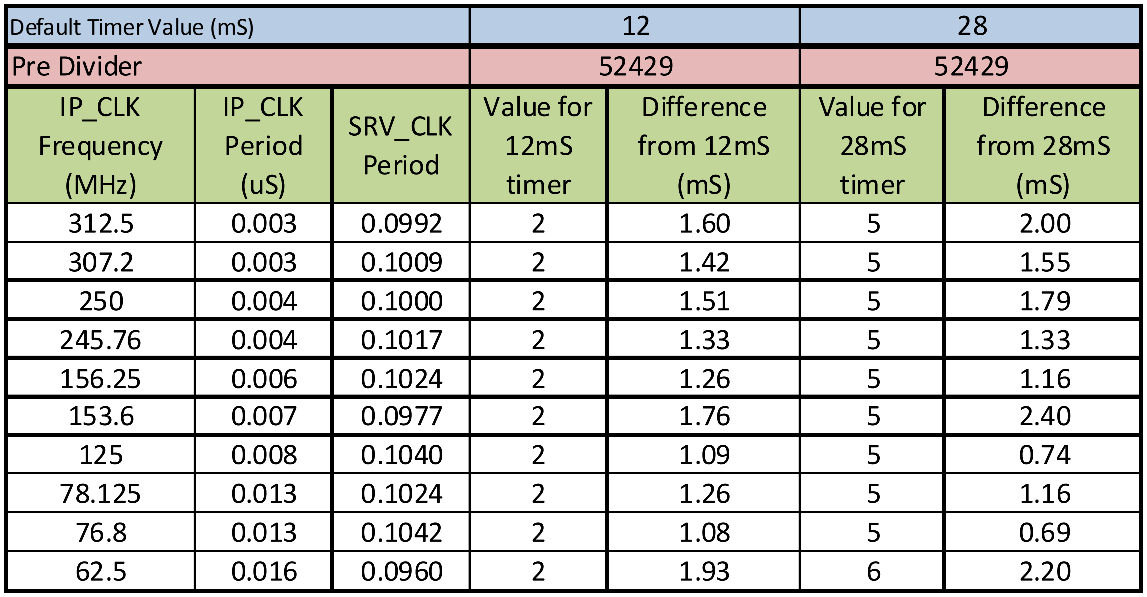
This register defines the discovery-timer value for the S-RIO port when it is configured in 4x or 2x mode.

Register Name: RIO_PLM_SP_DISCOVERY_TIMER Reset Value: 0x7000_0000	Register Offset: 0x100B4

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

Bits	Name	Description	Type	Reset Value
0:3	DISCOVERY_TIMER	Discovery Timer If the RapidIO Port Control CSR.PORT_WIDTH field indicates support for 4x or 2x mode, then this register exists for the S-RIO port. This register does not exist for the S-RIO port when it is configured in 1x mode. This field is used by the port to configure the standard discovery timer value. In the *RapidIO Interconnect Specification (Revision 2.1)*, the discovery timer is specified to be 28 msec +/- 4 msec. The 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. If the Discovery Timer expires before lane synchronization of lane alignment is achieved, the port returns to the SILENT state and begins the alignment process again. Note: In the *RapidIO Interconnect Specification (Revision 1.3)*, this timer is 12 msec +/- 4 msec. Discovery Interval is: • SRV_CLK period X 52429 X DISCOVERY_TIMER. For more information, see RapidIO PLM Port Discovery Timer Register Programming Values and RapidIO Port IP Prescalar for SRV_CLK Register. Note: Values of 0 and 1 are not legal	R/WS	7
4:31	Reserved	Reserved	R	0

Table 87: RapidIO PLM Port Discovery Timer Register Programming Values



Default Timer Value (mS)			12		28	
Pre Divider			52429		52429	
IP_CLK Frequency (MHz)	IP_CLK Period (uS)	SRV_CLK Period	Value for 12mS timer	Difference from 12mS (mS)	Value for 28mS timer	Difference from 28mS (mS)
312.5	0.003	0.0992	2	1.60	5	2.00
307.2	0.003	0.1009	2	1.42	5	1.55
250	0.004	0.1000	2	1.51	5	1.79
245.76	0.004	0.1017	2	1.33	5	1.33
156.25	0.006	0.1024	2	1.26	5	1.16
153.6	0.007	0.0977	2	1.76	5	2.40
125	0.008	0.1040	2	1.09	5	0.74
78.125	0.013	0.1024	2	1.26	5	1.16
76.8	0.013	0.1042	2	1.08	5	0.69
62.5	0.016	0.0960	2	1.93	6	2.20

Note: see Table 92 for Tsi721 IP_CLK frequency

18.9.10 RapidIO PLM Port Silence Timer Register

This register defines how long the port is in the SILENT state. The *RapidIO Interconnect Specification (Revision 2.1)* specifies the silence timer as 120 +/- 40 us.

Register Name: RIO_PLM_SP_SILENCE_TIMER Reset Value: 0x9000_0000	Register Offset: 0x100B8

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

Bits	Name	Description	Type	Reset Value
0:3	SILENCE_TIMER	Silence timer. This field defines the time that the port remains in the SILENT state. The default value of the timer depends on the frequency of the reference clock for the port. Silence Interval is • SRV_CLK period X 410 X SILENCE_TIMER For more information, see RapidIO PLM Port Silence Timer Register Programming Values and RapidIO Port IP Prescalar for SRV_CLK Register. Note: Values of 0 and 1 are not legal.	R/WS	1001
4:31	Reserved	Reserved	R	0

Table 88: RapidIO PLM Port Silence Timer Register Programming Values

Default Timer Value (uS)			120	
Pre Divider			410	
IP_CLK Frequency (MHz)	IP_CLK Period (uS)	SRV_CLK Period	Value for 120mS timer	Difference from 120mS (uS)
312.5	0.003	0.0992	3	116.06
307.2	0.003	0.1009	3	116.00
250	0.004	0.1000	3	115.08
245.76	0.004	0.1017	3	115.00
156.25	0.006	0.1024	3	112.13
153.6	0.007	0.0977	3	111.99
125	0.008	0.1040	3	110.16
78.125	0.013	0.1024	3	104.26
76.8	0.013	0.1042	3	103.98
62.5	0.016	0.0960	3	100.32

Note: see Table 92 for Tsi721 IP_CLK frequency

18.9.11 RapidIO PLM Port Vmin Exponent Register

This register is used only when IDLE2 is supported. It defines Vmin exponent. In accordance with the *RapidIO Interconnect Specification (Revision 2.1)* (Part 6, 4.12.4.2), Vmin can be specified to require $2^{VMIN\_EXP} - 1$ /VALID/ code groups with no intervening /INVALID/ code-groups before declaring synchronization.

Register Name: RIO_PLM_SP_VMIN_EXP Reset Value: 0x0003_0300	Register Offset: 0x100BC

Bits	0	1	2	3	4	5	6	7
00:07	Reserved			VMIN_EXP				
08:15	Reserved				IMAX			
16:23	Reserved				MMAX			
24:31	Reserved							

Bits	Name	Description	Type	Reset Value
0:2	Reserved	Reserved	R	0
3:7	VMIN_EXP	This field sets the number of /VALID/ code-groups required for synchronization. The number required is $2^{VMIN\_EXP} - 1$. Values for VMIN_EXP greater than 16 are reserved. The reset value of 0 is identical to the behavior specified in *RapidIO Interconnect Specification (Revision 1.3)*.	R/WS	0
8:11	Reserved	Reserved	R	0
12:15	IMAX	This field sets the maximum threshold for the *Icounter* specified by the *RapidIO Interconnect Specification (Revision 2.1)*. *Icounter* is a leaky bucket that allows the *Lane_Synchronization State Machine* to tolerate a number of bit errors before forcing the state machine to enter the NO_SYNC state. IMAX must be set to 3 or greater for receivers not using DFE, and must be set to 4 or greater for receivers using DFE. Values for IMAX less than 3 are reserved.	R/WS	3
16:19	Reserved	Reserved	R	0
20:23	MMAX	This field sets the maximum threshold for the *Mcounter* specified by the *RapidIO Interconnect Specification (Revision 2.1)*. *Mcounter* is a leaky bucket that allows the *Lane_Alignment State Machine* to tolerate a number of bit errors before forcing the state machine to enter the NOT_ALIGNED state. MMAX must be set to 3 or greater for receivers not using DFE and must be set to 4 or greater for receivers using DFE. Values for MMAX less than 3 are reserved.	R/WS	3
24:31	Reserved	Reserved	R	0

18.9.12 RapidIO PLM Port Lane Polarity Control Register

This register swaps the polarity of the differential pairs used by the port. The number of bits that are used by the port depends on the maximum port width and the port's current operating mode, as defined by the PORT_WIDTH and OVER_PWIDTH fields in the RapidIO Port Control CSR.

Register Name: RIO_PLM_SP_POL_CTL Reset Value: Undefined	Register Offset: 0x100C0

Bits	0	1	2	3	4	5	6	7
00:07	Reserved							
08:15	Reserved				TX3_POL	TX2_POL	TX1_POL	TX0_POL
16:23	Reserved							
24:31	Reserved				RX3_POL	RX2_POL	RX1_POL	RX0_POL

Bits	Name	Description	Type	Reset Value
0:11	Reserved	Reserved	R	0
12	TX3_POL	Determines whether the lines that compose a differential pair are swapped (see Table 89). 0 = Lines are not reversed 1 = Lines are reversed	R/WS	Undefined
13	TX2_POL	Determines whether the lines that compose a differential pair are swapped (see Table 89). 0 = Lines are not reversed 1 = Lines are reversed	R/WS	Undefined
14	TX1_POL	Determines whether the lines that compose a differential pair are swapped (see Table 89). 0 = Lines are not reversed 1 = Lines are reversed	R/WS	Undefined
15	TX0_POL	Determines whether the lines that compose a differential pair are swapped (see Table 89). 0 = Lines are not reversed 1 = Lines are reversed	R/WS	Undefined
16:27	Reserved	Reserved	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
28	RX3_POL	Determines whether the lines that compose a differential pair are swapped. 0 = Lines are not reversed 1 = Lines are reversed Note: RapidIO Lane {0..3} Status 0 CSR.RX_INV does not indicate the inversion set by this field (see Table 89).	R/WS	Undefined
29	RX2_POL	Determines whether the lines that compose a differential pair are swapped. 0 = Lines are not reversed 1 = Lines are reversed Note: RapidIO Lane {0..3} Status 0 CSR.RX_INV does not indicate the inversion set by this field (see Table 89).	R/WS	Undefined
30	RX1_POL	Determines whether the lines that compose a differential pair are swapped. 0 = Lines are not reversed 1 = Lines are reversed Note: RapidIO Lane {0..3} Status 0 CSR.RX_INV does not indicate the inversion set by this field (see Table 89).	R/WS	Undefined
31	RX0_POL	Determines whether the lines that compose a differential pair are swapped. 0 = Lines are not reversed 1 = Lines are reversed Note: RapidIO Lane {0..3} Status 0 CSR.RX_INV does not indicate the inversion set by this field (see Table 89).	R/WS	Undefined

18.9.13 RapidIO PLM Port Packet Denial Control Register

This register controls when a “too many retries” event is detected.

Register Name: RIO_PLM_SP_DENIAL_CTL Reset Value: 0x3000_0000	Register Offset: 0x100C8

Bits	0	1	2	3	4	5	6	7
00:07	Reserved		CNT_PNA	CNT_RTY	Reserved			
08:15	Reserved							
16:23	Reserved							
24:31	DENIAL_THRESH							

Bits	Name	Description	Type	Reset Value
0:1	Reserved	Reserved	R	0
2	CNT_PNA	Controls whether packet-not-accepted control symbols count toward the packet denial threshold value	R/WS	1
3	CNT_RTY	Controls whether Retry control symbols count toward the packet denial threshold value	R/WS	1
4:23	Reserved	Reserved	R	0
24:31	DENIAL_THRESH	Sets the threshold for reporting too many consecutive retries. 0 = Too many denials event is disabled – no denial event will be detected regardless of the number of consecutive retries. 1 = One denial triggers too many retries event 2 = Two consecutive denials triggers too many retries event ... 0xFF–255 consecutive denials triggers too many retries event For more information, see RapidIO PLM Port Event Status Register.MAX_DENIAL. Receiving consecutive retries (either PR or PNA) contribute to the “leaky bucket” for Port Degraded and Port Failed event detection (see RapidIO Port Error Detect CSR.IMP_SPEC). The leaky bucket is incremented once for each DENIAL_THRESH consecutive retries received. The denial counter is reset under two conditions: the denial counter reaches DENIAL_THRESH; or, Tsi721 receives a packet accept (PA) control symbol. Note: a Link-response which implicitly acknowledges packets does not reset the denial counter.	R/WS	0

18.9.14 RapidIO PLM Port Received MECS Status Register

Captures the MECS *CMD* field values that have been received by the S-RIO port. This register is provided for informational purposes and can only be cleared by software.

Register Name: RIO_PLM_SP_RCVD_MECS Reset Value: 0x0000_0000	Register Offset: 0x100D0

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

Bits	Name	Description	Type	Reset Value
0:23	Reserved	Reserved	R	0
24:31	CMD_STAT	This bit field indicates which MECS commands have been received. Bit 31 = MECS with CMD value of 0 has been received Bit 30 = MECS with CMD value of 1has been received Bit 29 = MECS with CMD value of 2 has been received ... Bit 24 = MECS with CMD value of 7 has been received	R/W1CS	0

18.9.15 RapidIO PLM Port MECS Forwarding Register

This register allows the port to forward MECSs with *CMD* values that match those enabled in the SUBSCRIPTION field. The MECS being forwarded by the port was received under command from the RapidIO Event Management MECS Request Register.

Register Name: RIO_PLM_SP_MECS_FWD Reset Value: 0x0000_0000	Register Offset: 0x100D8

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

Bits	Name	Description	Type	Reset Value
0:23	Reserved	Reserved	R	0
24:30	SUBSCRIPTION	Specifies which MECS CMD values the port should forward to its link. Bit 31 = Forward MECS with CMD value of 0 Bit 30 = Forward MECS with CMD value of 1 Bit 29 = Forward MECS with CMD value of 2 ... Bit 24 = Forward MECS with CMD value of 7	R/WS	0x00
31	MULT_CS	This bit is mapped to RapidIO Port Control CSR.MULT_CS and performs the same function. Either offset can access this bit.	R/WS	0

18.9.16 RapidIO PLM Port Control Symbol Transmit 1 Register

Writing to this register controls the stype0 and stype1 values of the short or long control symbol to be sent. Writing to this register triggers transmission of a short control symbol when IDLE1 is in use, and a long control symbol when IDLE2 is active.

When long control symbols are in use, this register should be written only after the RapidIO PLM Port Control Symbol Transmit 2 Register has been written. All control symbol fields are defined according to the *RapidIO Interconnect Specification (Revision 2.1)*. The control symbol's CRC field is generated by hardware.

Register Name: RIO_PLM_SP_LONG_CS_TX1 Reset Value: 0x0000_0000	Register Offset: 0x100E0

Bits	0	1	2	3	4	5	6	7
00:07	Reserved	STYPE_0			Reserved		PAR_0	
08:15	PAR_0				Reserved		PAR_1	
16:23	PAR_1				Reserved			CS_EMB
24:31	Reserved	STYPE_1			Reserved	CMD		

Bits	Name	Description	Type	Reset Value
0	Reserved	Reserved	R	0
1:3	STYPE_0	Encoding for control symbol This field uses the parameters, PAR_0 and PAR_1.	R/WS	0
4:5	Reserved	Reserved	R	0
6:11	PAR_0	Used in conjunction with stype0 encoding.	R/WS	0
12:13	Reserved	Reserved	R	0
14:19	PAR_1	Used in conjunction with stype0 encoding.	R/WS	0
20:22	Reserved	Reserved	R	0
23	CS_EMB	Embed the control symbol into a data stream 0 = Control symbol is transmitted immediately and may be embedded in a packet 1 = Control symbol is transmitted between a SOP control symbol and the control symbol that delimits the end of a packet. Note: To reliably embed the control symbol, this field should be set while traffic is stopped.	R/WS	0
24	Reserved	Reserved	R	0
25:27	STYPE_1	Encoding for the control symbol that uses the CMD parameter.	R/WS	0

(Continued)

Bits	Name	Description	Type	Reset Value
28	Reserved	Reserved	R	0
29:31	CMD	Used in conjunction with stype1 encoding to define the link maintenance commands.	R/WS	0

⚠ This register must not be written unless the port is initialized: RapidIO Port Error and Status CSR.PORT_OK is 1

⚠ Writing to this register causes control symbols to be generated that can interfere with the operation of the port. This can cause the port or its link partner to enter the input error-stopped state due to the reception of an unexpected control symbol.

18.9.17 RapidIO PLM Port Control Symbol Transmit 2 Register

This register is used only when IDLE2 is supported. Writing to this register controls the stype2 bits of a long control symbol to be sent to the link partner. To generate a long control symbol, write to this register and then write to the RapidIO PLM Port Control Symbol Transmit 1 Register.

All control symbol fields are defined according to the *RapidIO Interconnect Specification (Revision 2.1)*. The control symbol's CRC field is generated by hardware. When short control symbols/IDLE1 is active, writes to this register are ignored.

Register Name: RIO_PLM_SP_LONG_CS_TX2 Reset Value: 0x0000_0000	Register Offset: 0x100E4

Bits	0	1	2	3	4	5	6	7
00:07	Reserved	STYPE2			Reserved	PARM		
08:15	PARM							
16:23	Reserved							
24:31	Reserved							

Bits	Name	Description	Type	Reset Value
0	Reserved	Reserved	R	0
1:3	STYPE2	STYPE2 encoding, as defined in Part 12.	R/WS	0
4	Reserved	Reserved	R	0
5:15	PARM	Used in conjunction with STYPE2 encoding.	R/WS	0
16:31	Reserved	Reserved	R	0

18.10 TLM Register Block

18.10.1 IDT-Specific Transport Layer Block Header Register

This register identifies the following block of registers as the IDT Transport Layer Device Specific register block.

Register Name: RIO_TLM_BH Reset Value: 0x0106_0000	Register Offset: 0x10300

Bits	0	1	2	3	4	5	6	7
00:07	NEXT_BLK_PTR							
08:15	NEXT_BLK_PTR							
16:23	BLK_REV				BLK_TYPE			
24:31	BLK_TYPE							

Bits	Name	Description	Type	Reset Value
0:15	NEXT_BLK_PTR	Pointer to the next IDT register block, in units of 0x100 bytes. This points to the PBM Register Block.	RS	0x0106
16:19	BLK_REV	IDT-defined revision of the block type in this device.	RS	0x0
20:31	BLK_TYPE	IDT-defined identifier for the register block type.	RS	0x000

18.10.2 RapidIO TLM Port Control Register

This register provides control of the TLM for the port.

Register Name: RIO_TLM_SP_CONTROL Reset Value: 0x0030_9000	Register Offset: 0x10380

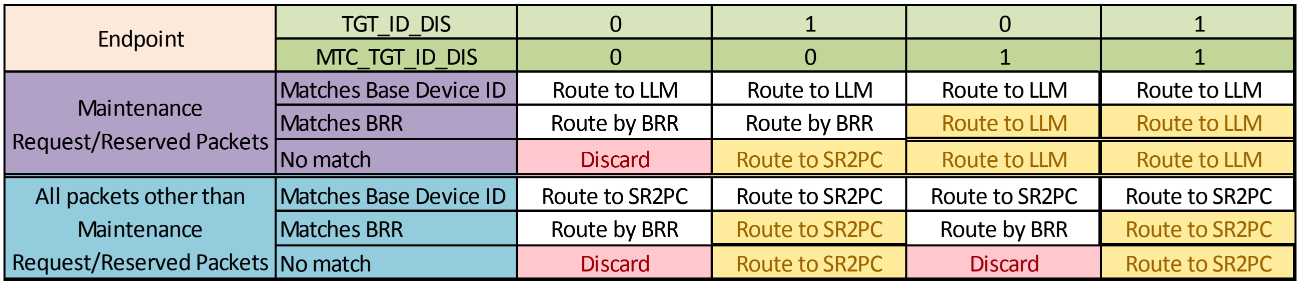
Bits	0	1	2	3	4	5	6	7
00:07	Reserved	PORTGROUP_SELECT	Reserved					
08:15	Reserved		TGT_ID_DIS	MTC_TGT_ID_DIS	Reserved			
16:23	LENGTH				Reserved			
24:31	Reserved							

Bits	Name	Description	Type	Reset Value
0	Reserved	Reserved	R	0
1	PORTGROUP_SELECT	Configures the number of Port Groups to which the TLMi directs traffic which is not routed to the LLM. 0 = One port group Other values are reserved. Within the Port Group, a VOQ is selected according to VOQ_SELECT.	R/WS	0x0
2:9	Reserved	Reserved	R	0
10	TGT_ID_DIS	Target ID Filtering Disable Used in conjunction with MTC_TGT_ID_DIS to configure destID filtering mode. 0 = Enable destID filtering 1 = Disable destID filtering TGT_ID_DIS can be configured after a fundamental reset to filter non-maintenance requests based on destID values. For more information, see destID Filtering.	R/WS	0x1
11	MTC_TGT_ID_DIS	Maintenance Target ID Filtering Disable Used in conjunction with TGT_ID_DIS to configure destID filtering mode. 0 = Enable destID filtering 1 = Disable destID filtering MTC_TGT_ID_DIS can be configured after a fundamental reset to filter maintenance requests based on destID values. For more information, see destID Filtering.	R/WS	0x1

(Continued)

Bits	Name	Description	Type	Reset Value
12:15	Reserved	Reserved	R	0x0
16:19	LENGTH	Sets the number of data nodes that must be stored before PBMi notifies the scheduler of a packet to be scheduled. A value of 9 in this field enables Store-and-Forward.	RS	0x9
20:31	Reserved	Reserved	R	0x000

Table 90: Behavior of TGT_ID_DIS and MTC_TGT_ID_DIS[1]



Endpoint	TGT_ID_DIS	0	1	0	1
	MTC_TGT_ID_DIS	0	0	1	1
Maintenance Request/Reserved Packets	Matches Base Device ID	Route to LLM	Route to LLM	Route to LLM	Route to LLM
	Matches BRR	Route by BRR	Route by BRR	Route to LLM	Route to LLM
	No match	Discard	Route to SR2PC	Route to LLM	Route to LLM
All packets other than Maintenance Request/Reserved Packets	Matches Base Device ID	Route to SR2PC	Route to SR2PC	Route to SR2PC	Route to SR2PC
	Matches BRR	Route by BRR	Route to SR2PC	Route by BRR	Route to SR2PC
	No match	Discard	Route to SR2PC	Discard	Route to SR2PC

[1] BRR means base routing register (see RapidIO TLM Base Routing Register Control Register {0..15}). "Route by BRR" means rout according to the BRR's ROUTE_MR_TO_LLM or ROUTE_ALL_TO_LLM (see same register).

18.10.3 RapidIO TLM Port Status Register

This register provides status of errors detected by the TLM. Note that interrupts can be asserted and port-writes can be sent for the same event in this register. In this case, software must avoid race conditions when reading this register and clearing the events.

Register Name: RIO_TLM_SP_STATUS Reset Value: 0x0000_0000	Register Offset: 0x10390

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

Bits	Name	Description	Type	Reset Value
0	IG_BAD_VC	Detected an inbound packet with the VC bit set. The packet that sets this bit is not logged.	R/W1CS	0x0
1:10	Reserved	Reserved	R	0
11	IG_BRR_FILTER	Discarded an inbound transaction based on the BRR programming.	R/W1CS	0x0
12:31	Reserved	Reserved	R	0x0000

18.10.4 RapidIO TLM Port Interrupt Enable Register

This register enables status of errors detected by the TLM to notify the system host by interrupt. Bits set in the RapidIO TLM Port Status Register and enabled here are summarized in TLM_INT of the RapidIO PLM Port Event Status Register.

Register Name: RIO_TLM_SP_INT_ENABLE Reset Value: 0x0000_0000	Register Offset: 0x10394

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

Bits	Name	Description	Type	Reset Value
0	IG_BAD_VC	1 = Enable IG_BAD_VC to contribute to the port's interrupt request.	R/WS	0x0
1:10	Reserved	Reserved	R	0
11	IG_BRR_FILTER	1 = Enable IG_BRR_FILTER to contribute to the port's interrupt request.	R/WS	0x0
12:31	Reserved	Reserved	R	0x00000

18.10.5 RapidIO TLM Port Port-Write Enable Register

This register enables status of errors detected by the TLM to notify the system host by a port-write. Bits set in RapidIO TLM Port Status Register and enabled here are summarized in TLM_PW of the RapidIO PLM Port Event Status Register.

Register Name: RIO_TLM_SP_PW_ENABLE Reset Value: 0x0000_0000	Register Offset: 0x10398

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

Bits	Name	Description	Type	Reset Value
0	IG_BAD_VC	1 = Enable IG_BAD_VC to contribute to the port's port-write request.	R/WS	0x0
1:10	Reserved	Reserved	R	0
11	IG_BRR_FILTER	1 = Enable IG_BRR_FILTER to contribute to the port's port-write request.	R/WS	0x0
12:31	Reserved	Reserved	R	0x00000

18.10.6 RapidIO TLM Port Event Generate Register

This register generates TLM events for software verification.

Register Name: RIO_TLM_SP_EVENT_GEN Reset Value: 0x0000_0000	Register Offset: 0x1039C

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

Bits	Name	Description	Type	Reset Value
0	IG_BAD_VC	1 = Set the corresponding bit in the RapidIO TLM Port Status Register.	R/W1S	0x0
1:10	Reserved	Reserved	R	0
11	IG_BRR_FILTER	1 = Set the corresponding bit in the RapidIO TLM Port Status Register.	R/W1S	0x0
12:31	Reserved	Reserved	R	0x00000

18.10.7 RapidIO TLM Base Routing Register Control Register {0..15}

This register controls how the Base Routing Register (BRR) is applied to inbound packets. This is BRR 0. The lower the number, the higher the priority of the BRR. When channels are combined to form wider ports, the lower the channel number the higher priority its BRRs have.

Register Name: RIO_TLM_SP_BRR_CTL{0..15} Reset Value: 0x0500_0000	Register Offset: 0x103A0, 0x103B0, 0x103C0, 0x103D0, 0x10420, 0x10430, 0x10440, 0x10450, 0x104A0, 0x104B0, 0x104C0, 0x104D0, 0x10520, 0x10530, 0x10540, 0x10550

Bits	0	1	2	3	4	5	6	7
00:07	ENABLE	RESERVED				ROUTE_MR_TO_LLM	RESERVED	PRIVATE
08:15	RESERVED							
16:23	RESERVED							
24:31	RESERVED							

Bits	Name	Description	Type	Reset Value
0	ENABLE	1 = Enable the BRR to accept and route inbound packets.	R/WS	0x0
1:4	RESERVED	Reserved.	R	0x0
5	ROUTE_MR_TO_LLM	1 = Maintenance request/reserved packets with destIDs that match this BRR are routed to the LLM; otherwise, they are routed to the Messaging Engine, Mapping Engine, or Block DMA Engine.	R/WS	0x1
6	RESERVED	Reserved.	R	0x0
7	PRIVATE	Configures the BRR to be used only by its S-RIO port. 1 = Private – used only by the S-RIO port	R/WS	0x1
8:31	RESERVED	Reserved.	R	0x0

18.10.8 RapidIO TLM Base Routing Register Pattern and Match Register 0

This register provides the pattern to which the inbound destID is compared, and which bits participate in the comparison. RapidIO TLM Base Routing Register Control Register {0..15}.PRIVATE affects the reset behavior of this register.

Register Name: RIO_TLM_SP_BRR_PATTERN_MATCH{0,4,8,12} Reset Value: 0x0000_0000	Register Offset: 0x103A4, 0x10424, 0x104A4, 0x10524

Bits	0	1	2	3	4	5	6	7
00:07	PATTERN							
08:15	PATTERN							
16:23	MATCH							
24:31	MATCH							

Bits	Name	Description	Type	Reset Value
0:15	PATTERN	This field provides the 16 bits that are compared one-for-one against the inbound destID. For comparison purposes, the upper 8 bits of the destID are processed as zeros when 8-bit destIDs are compared.	R/WS	0x0000
16:31	MATCH	This field indicates which of the 16 bits of the inbound destID are compared against PATTERN.	R/WS	0x0000

18.10.9 RapidIO TLM Base Routing Register Pattern and Match Register 1

This register provides the pattern to which the inbound destID is compared, and which bits participate in the comparison. RapidIO TLM Base Routing Register Control Register {0..15}.PRIVATE affects the reset behavior of this register.

Register Name: RIO_TLM_SP_BRR_PATTERN_MATCH{1,2,3,5,6,7,9,10,11,13,14,15} Reset Value: 0x0000_FFFF	Register Offset: 0x103B4, 0x103C4, 0x103D4, 0x10434, 0x10444, 0x10454, 0x104B4, 0x104C4, 0x104D4, 0x10534, 0x10544, 0x10554

Bits	0	1	2	3	4	5	6	7
00:07	PATTERN							
08:15	PATTERN							
16:23	MATCH							
24:31	MATCH							

Bits	Name	Description	Type	Reset Value
0:15	PATTERN	This field provides the 16 bits that are compared one-for-one against the inbound destID. For comparison purposes, the upper 8 bits of the destID are processed as zeros when 8-bit destIDs are compared.	R/WS	0x0000
16:31	MATCH	This field indicates which of the 16 bits of the inbound destID are compared against PATTERN.	R/WS	0xFFFF

18.10.10 RapidIO TLM Port ftype Filter Control Register

This register selects combinations of ftype and ttype values to which filtering should be applied

Register Name: RIO_TLM_SP_FTYPE_FILTER_CTL Reset Value: 0x0000_0000	Register Offset: 0x103E0

Bits	0	1	2	3	4	5	6	7
00:07	RESERVED	F0_IMPLEMENTATION	RESERVED	F1_ALL	F2_GSM	F2_NREAD	F2_ATOMIC	F3_RSVD
08:15	F4_RSVD	F5_GSM	F5_NWRITE	F5_NWRITE_R	F5_ATOMIC	F5_OTHER	F6_STREAMING_WRITE	F7_FLOW
16:23	F8_MR	F8_MW	F8_MRR	F8_MWR	F8_PWR	F8_OTHER	F9_DATA_STREAMING	F10_DOORBELL
24:31	F11_MESSAGE	F12_RSVD	F13_RESPONSE	F13_RESPONSE_DATA	F13_OTHER	F14_RSVD	F15_IMPLEMENTATION	RESERVED

Bits	Name	Description	Type	Reset Value
0	RESERVED	Reserved	R	0x0
1	F0_IMPLEMENTATION	0 = Inbound packets with ftype 0 are accepted. 1 = Inbound packets with ftype 0 are filtered.	R/WS	0x0
2	Reserved	Reserved	R	0x0
3	F1_ALL	1 = Inbound packets with ftype 1 are filtered.	R/WS	0x0
4	F2_GSM	1 = Inbound packets with ftype 2, ttype 0, 1, 2, 3, 5, 6, 7, 8, 9, 10, or 11 are filtered.	R/WS	0x0
5	F2_NREAD	1 = Inbound packets with ftype 2, ttype 4 are filtered.	R/WS	0x0
6	F2_ATOMIC	1 = Inbound packets with ftype 2, ttype 12, 13, 14, or 15 are filtered.	R/WS	0x0
7	F3_RSVD	1 = Inbound packets with ftype 3 are filtered.	R/WS	0x0
8	F4_RSVD	1 = Inbound packets with ftype 4 are filtered.	R/WS	0x0
9	F5_GSM	1 = Inbound packets with ftype 5, ttype 0 or 1 are filtered.	R/WS	0x0
10	F5_NWRITE	1 = Inbound packets with ftype 5, ttype 4 are filtered.	R/WS	0x0
11	F5_NWRITE_R	1 = Inbound packets with ftype 5, ttype 5 are filtered.	R/WS	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
12	F5_ATOMIC	1 = Inbound packets with ftype 5, ttype 12, 13, or 14 are filtered.	R/WS	0x0
13	F5_OTHER	1 = Inbound packets with ftype 5, ttype 2, 3, 6, 7, 8, 9, 10, 11 are filtered.	R/WS	0x0
14	F6_STREAMING_WRITE	1 = Inbound packets with ftype 6 are filtered.	R/WS	0x0
15	F7_FLOW	1 = Inbound packets with ftype 7 are filtered.	R/WS	0x0
16	F8_MR	1 = Inbound packets with ftype 8, ttype {0} are filtered. *RapidIO Interconnect Specification (Revision 2.1)* Part 3, section 2.3 System Packet Routing recommends that even if the application restricts processing requests to a list of acceptable deviceIDs, that all maintenance read requests be processed. Note: When setting this bit, the port's bit must also be set in the RapidIO Maintenance Read Request Restriction Control Register. This is required so that the LLM can distinguish between "filtered" packets and "normal" (that is, not filtered) maintenance packets.	R/WS	0x0
17	F8_MW	1 = Inbound packets with ftype 8, ttype 1 are filtered. Note: When setting this bit, the port's bit must also be set in the RapidIO Maintenance Write Request Restriction Control Register. This is required so that the LLM can distinguish between "filtered" packets and "normal" (that is, not filtered) maintenance packets.	R/WS	0x0
18	F8_MRR	1 = Inbound packets with ftype 8, ttype 2 are filtered.	R/WS	0x0
19	F8_MWR	1 = Inbound packets with ftype 8, ttype 3 are filtered.	R/WS	0x0
20	F8_PWR	1 = Inbound packets with ftype 8, ttype 4 are filtered. Note: When setting this bit, the PWR_DIS bit must also be set in the RapidIO Maintenance Port-Write Request Restriction Control Register. This is required so that the LLM can distinguish between "filtered" packets and "normal" (that is, not filtered) maintenance packets.	R/WS	0x0
21	F8_OTHER	1 = Inbound packets with ftype 8, ttype 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, or 15 are filtered.	R/WS	0x0
22	F9_DATA_STREAMING	1 = Inbound packets with ftype 9 are filtered.	R/WS	0x0
23	F10_DOORBELL	1 = Inbound packets with ftype 10 are filtered.	R/WS	0x0
24	F11_MESSAGE	1 = Inbound packets with ftype 11 are filtered.	R/WS	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
25	F12_RSVD	1 = Inbound packets with ftype 12 are filtered.	R/WS	0x0
26	F13_RESPONSE	1 = Inbound packets with ftype 13, ttype 0 are filtered.	R/WS	0x0
27	F13_RESPONSE_DATA	1 = Inbound packets with ftype 13, ttype 8 are filtered.	R/WS	0x0
28	F13_OTHER	1 = Inbound packets with ftype 13, ttype 1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 12, 13, 14, or 15 are filtered.	R/WS	0x0
29	F14_RSVD	1 = Inbound packets with ftype 14 are filtered.	R/WS	0x0
30	F15_IMPLEMENTATION	1 = Inbound packets with ftype 15 are filtered.	R/WS	0x0
31	RESERVED	Reserved	R	0x0

18.11 PBM Register Block

This section defines the registers for controlling the PBMi and PBMe, and for reporting their status.

18.11.1 IDT-Specific Packet Buffer Module Header Register

This register identifies the following block of registers as the IDT Packet Buffer Module (PBM) specific register block

Register Name: RIO_PBM_BH Reset Value: 0x0109_0000	Register Offset: 0x10600

Bits	0	1	2	3	4	5	6	7
00:07	NEXT_BLK_PTR							
08:15	NEXT_BLK_PTR							
16:23	BLK_REV				BLK_TYPE			
24:31	BLK_TYPE							

Bits	Name	Description	Type	Reset Value
0:15	NEXT_BLK_PTR	Pointer to the next IDT register block, in units of 0x100 bytes. This points to Event Management Registers.	RS	0x0109
16:19	BLK_REV	IDT-defined revision of the block type in this device.	RS	0x0
20:31	BLK_TYPE	IDT-defined identifier for the register block type.	RS	0x000

18.11.2 RapidIO PBM Port Control Register

This register must be written only during bootload process or while the port is in reset.

Register Name: RIO_PBM_SP_CONTROL Reset Value: Undefined	Register Offset: 0x10680

Bits	0	1	2	3	4	5	6	7
00:07	RESERVED							
08:15	RESERVED							IG_BACKPRESSURE_ON_FATAL
16:23	RESERVED							
24:31	RESERVED		EG_REORDER_MODE		RESERVED	EG_REORDER_STICK		

Bits	Name	Description	Type	Reset Value
0:14	RESERVED	Reserved	R	Undefined
15	IG_BACKPRESSURE_ON_FATAL	By default, when the PBMi detects a fatal error, all inbound packets are silently discarded. When IG_BACKPRESSURE_ON_FATAL is set and a fatal error occurs, the PBMi asserts back-pressure to all flowIDs regardless of fill level; when clear, the PBMi removes back-pressure from all flowIDs. Fatal errors are identified in the RapidIO PBM Port Status Register by fields containing "_FATAL" in their name.	R/WS	Undefined
16:25	RESERVED	Reserved	R	Undefined
26:27	EG_REORDER_MODE	Selects the reorder action within PBMe 0b00 = Reorder on reorder trigger, bring the oldest higher preference flowID to the front of the CRQ. Reordering is performed after each dequeue if EG_REORDER_STICK is greater than 0. 0b01 = Reorder on reorder trigger, bring the oldest highest preference flowID to front of CRQ. Reordering is performed after each dequeue if EG_REORDER_STICK is greater than 0. 0b10 = Reserved 0b11 = Reorder on each dequeue, bring oldest highest preference flowID to front of CRQ	R/WS	Undefined

(Continued)

Bits	Name	Description	Type	Reset Value
28	RESERVED	Reserved	R	Undefined
29:31	EG_REORDER_STICK	Selects the number of repeat times the CRQ is reordered after each reorder event unless the CRQ has no unsent packets: 0 = Reorder once 1 = Reorder twice ... 7 = Reorder eight times This field does not apply if EG_REORDER_MODE always selects the “oldest highest preference flowID”. Note that the following are counted as “reorders”: 1. CRQ is reordered and a new packet is brought to the head of the CRQ 2. CRQ is reordered, but the packet at the head of the CRQ remains at its head If the packet that ends up at the head of the CRQ after a reorder is marked for discard, the reorder is counted even though the packet is not transmitted.	R/WS	Undefined

18.11.3 RapidIO PBM Port Status Register

This register reports the status of various error conditions that are detected within the PBMi and PBMe. Note that interrupts can be asserted and port-writes can be sent for the same event in this register. In this case, software must avoid race conditions when reading this register and clearing the events.

ECC protection provides detection of non-fatal single-bit errors and fatal double-bit errors.

Register Name: RIO_PBM_SP_STATUS Reset Value: 0x0001_8000	Register Offset: 0x10690

Bits	0	1	2	3	4	5	6	7
00:07	IG_DATA_COR	IG_DATA_UNCOR	IG_TAG_COR	IG_TAG_FATAL	IG_TFL_COR	IG_TFL_FATAL	IG_DOH_COR	IG_DOH_FATAL
08:15	IG_DNFL_COR	IG_DNFL_FATAL	Reserved					IG_EMPTY
16:23	EG_EMPTY	Reserved		EG_DATA_COR	EG_DATA_UNCOR	Reserved		EG_DOH_COR
24:31	EG_DOH_FATAL	EG_DNFL_COR	EG_DNFL_FATAL	EG_DATA_OVERFLOW	EG_CRQ_OVERFLOW	Reserved	EG_BAD_CHANNEL	EG_BABBLE_PACKET

Bits	Name	Description	Type	Reset Value
0	IG_DATA_COR	PBMi detected and corrected a single-bit error on the data portion of a data node during dequeue (only available if ECC enabled).	R/W1CS	0x0
1	IG_DATA_UNCOR	PBMi detected a double-bit error on the data portion of a data node during dequeue (double-bit error if ECC enabled).	R/W1CS	0x0
2	IG_TAG_COR	PBMi detected and corrected a single-bit error on a Tag (only available if ECC enabled).	R/W1CS	0x0
3	IG_TAG_FATAL	PBMi detected a fatal bit error on a Tag (double-bit error if ECC enabled).	R/W1CS	0x0
4	IG_TFL_COR	PBMi detected and corrected a single-bit error in the Tag Free List (only available if ECC enabled).	R/W1CS	0x0
5	IG_TFL_FATAL	PBMi detected a fatal bit error in the Tag Free List (double-bit error if ECC enabled).	R/W1CS	0x0
6	IG_DOH_COR	PBMi detected and corrected a single-bit error on a data node overhead memory (only available if ECC enabled).	R/W1CS	0x0
7	IG_DOH_FATAL	PBMi detected a fatal bit error on a data node overhead memory (double-bit error if ECC enabled).	R/W1CS	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
8	IG_DNFL_COR	PBMi detected and corrected a single-bit error in the DNFL (only available if ECC enabled).	R/W1CS	0x0
9	IG_DNFL_FATAL	PBMi detected a fatal bit error in the DNFL (double-bit error if ECC enabled).	R/W1CS	0x0
10:14	Reserved	Reserved	R	0x0
15	IG_EMPTY	The PBMi has no packets enqueued. This field is informational only and provides the “instantaneous” status of the PBMi. No interrupt is available for this status bit.	R	0x1
16	EG_EMPTY	The PBMe has not packets enqueued. This field is informational only and provides the “instantaneous” status of the PBMe. No interrupt is available for this status bit.	R	0x1
17:18	Reserved	Reserved	R	0x0
19	EG_DATA_COR	PBMe detected and corrected a single-bit error on the data portion of a data node during dequeue (only available if ECC enabled).	R/W1CS	0x0
20	EG_DATA_UNCOR	PBMe detected a double-bit error on the data portion of a data node during dequeue (double-bit error if ECC enabled).	R/W1CS	0x0
21:22	Reserved	Reserved	R	0x0
23	EG_DOH_COR	PBMe detected and corrected a single-bit error on a data node overhead memory (only available if ECC enabled).	R/W1CS	0x0
24	EG_DOH_FATAL	PBMe detected a fatal bit error on a data node overhead memory (double-bit error if ECC enabled).	R/W1CS	0x0
25	EG_DNFL_COR	PBMe detected and corrected a single-bit error in the DNFL (only available if ECC enabled).	R/W1CS	0x0
26	EG_DNFL_FATAL	PBMe detected a fatal bit error in the DNFL (double-bit error if ECC enabled).	R/W1CS	0x0
27	EG_DATA_OVERFLOW	PBMe received a request to enqueue a packet for which it did not have enough data storage.	R/W1CS	0x0
28	EG_CRQ_OVERFLOW	PBMe received a request to enqueue a packet for which it did not have a CRQ entry.	R/W1CS	0x0
29	Reserved	Reserved	R	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
30	EG_BAD_CHANNEL	PBMe received a request to enqueue a packet on a channel Enqueue Interface that should be unused for the path’s mode. Caution: If this bit is set, one or more packets may have been lost. Note: If the path is configured with *n* channels, this error is detectable unless the path is configured as *n* 1x ports. Note: If the path is configured with *n* channels, this error is only detected on channel Enqueue Interfaces from 0 to *n*-1.	R/W1CS	0x0
31	EG_BABBLE_PACKET	PBMe detected a packet that exceeded 276 bytes on its Enqueue Interface.	R/W1CS	0x0

18.11.4 RapidIO PBM Port Interrupt Enable Register

This register enables status of errors detected by the PBM to notify the system host by interrupt. Bits that are set in the RapidIO PBM Port Status Register and also enabled here are summarized in PBM_INT of RapidIO PLM Port Event Status Register.

Register Name: RIO_PBM_SP_INT_ENABLE Reset Value: 0x0000_0000	Register Offset: 0x10694

Bits	0	1	2	3	4	5	6	7
00:07	IG_DATA_COR	IG_DATA_UNCOR	IG_TAG_COR	IG_TAG_FATAL	IG_TFL_COR	IG_TFL_FATAL	IG_DOH_COR	IG_DOH_FATAL
08:15	IG_DNFL_COR	IG_DNFL_FATAL	Reserved					
16:23	Reserved			EG_DATA_COR	EG_DATA_UNCOR	Reserved		EG_DOH_COR
24:31	EG_DOH_FATAL	EG_DNFL_COR	EG_DNFL_FATAL	EG_DATA_OVERFLOW	EG_CRQ_OVERFLOW	Reserved	EG_BAD_CHANNEL	EG_BABBLE_PACKET

Bits	Name	Description	Type	Reset Value
0	IG_DATA_COR	1 = Enable this event to contribute to the port's interrupt request.	R/WS	0x0
1	IG_DATA_UNCOR	1 = Enable this event to contribute to the port's interrupt request.	R/WS	0x0
2	IG_TAG_COR	1 = Enable this event to contribute to the port's interrupt request.	R/WS	0x0
3	IG_TAG_FATAL	1 = Enable this event to contribute to the port's interrupt request.	R/WS	0x0
4	IG_TFL_COR	1 = Enable this event to contribute to the port's interrupt request.	R/WS	0x0
5	IG_TFL_FATAL	1 = Enable this event to contribute to the port's interrupt request.	R/WS	0x0
6	IG_DOH_COR	1 = Enable this event to contribute to the port's interrupt request.	R/WS	0x0
7	IG_DOH_FATAL	1 = Enable this event to contribute to the port's interrupt request.	R/WS	0x0
8	IG_DNFL_COR	1 = Enable this event to contribute to the port's interrupt request.	R/WS	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
9	IG_DNFL_FATAL	1 = Enable this event to contribute to the port’s interrupt request.	R/WS	0x0
10:18	Reserved	Reserved	R	0x0
19	EG_DATA_COR	1 = Enable this event to contribute to the port’s interrupt request.	R/WS	0x0
20	EG_DATA_UNCOR	1 = Enable this event to contribute to the port’s interrupt request.	R/WS	0x0
21:22	Reserved	Reserved	R	0x0
23	EG_DOH_COR	1 = Enable this event to contribute to the port’s interrupt request.	R/WS	0x0
24	EG_DOH_FATAL	1 = Enable this event to contribute to the port’s interrupt request.	R/WS	0x0
25	EG_DNFL_COR	1 = Enable this event to contribute to the port’s interrupt request.	R/WS	0x0
26	EG_DNFL_FATAL	1 = Enable this event to contribute to the port’s interrupt request.	R/WS	0x0
27	EG_DATA_OVERFLOW	1 = Enable this event to contribute to the port’s interrupt request.	R/WS	0x0
28	EG_CRQ_OVERFLOW	1 = Enable this event to contribute to the port’s interrupt request.	R/WS	0x0
29	Reserved	Reserved	R	0x0
30	EG_BAD_CHANNEL	1 = Enable this event to contribute to the port’s interrupt request.	R/WS	0x0
31	EG_BABBLE_PACKET	1 = Enable this event to contribute to the port’s interrupt request.	R/WS	0x0

18.11.5 RapidIO PBM Port Port-Write Enable Register

This register enables status of errors detected by the PBM to notify the system host by port-write. Bits that are set in the RapidIO PBM Port Status Register and also enabled here are summarized in PBM_PW of RapidIO PLM Port Event Status Register.

Register Name: RIO_PBM_SP_PW_ENABLE Reset Value: 0x0000_0000	Register Offset: 0x10698

Bits	0	1	2	3	4	5	6	7
00:07	IG_DATA_COR	IG_DATA_UNCOR	IG_TAG_COR	IG_TAG_FATAL	IG_TFL_COR	IG_TFL_FATAL	IG_DOH_COR	IG_DOH_FATAL
08:15	IG_DNFL_COR	IG_DNFL_FATAL	Reserved					
16:23	Reserved			EG_DATA_COR	EG_DATA_UNCOR	Reserved		EG_DOH_COR
24:31	EG_DOH_FATAL	EG_DNFL_COR	EG_DNFL_FATAL	EG_DATA_OVERFLOW	EG_CRQ_OVERFLOW	Reserved	EG_BAD_CHANNEL	EG_BABBLE_PACKET

Bits	Name	Description	Type	Reset Value
0	IG_DATA_COR	1 = Enable this event to contribute to the port's port-write request.	R/WS	0x0
1	IG_DATA_UNCOR	1 = Enable this event to contribute to the port's port-write request.	R/WS	0x0
2	IG_TAG_COR	1 = Enable this event to contribute to the port's port-write request.	R/WS	0x0
3	IG_TAG_FATAL	1 = Enable this event to contribute to the port's port-write request.	R/WS	0x0
4	IG_TFL_COR	1 = Enable this event to contribute to the port's port-write request.	R/WS	0x0
5	IG_TFL_FATAL	1 = Enable this event to contribute to the port's port-write request.	R/WS	0x0
6	IG_DOH_COR	1 = Enable this event to contribute to the port's port-write request.	R/WS	0x0
7	IG_DOH_FATAL	1 = Enable this event to contribute to the port's port-write request.	R/WS	0x0
8	IG_DNFL_COR	1 = Enable this event to contribute to the port's port-write request.	R/WS	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
9	IG_DNFL_FATAL	1 = Enable this event to contribute to the port's port-write request.	R/WS	0x0
10:18	Reserved	Reserved	R	0x0
19	EG_DATA_COR	1 = Enable this event to contribute to the port's port-write request.	R/WS	0x0
20	EG_DATA_UNCOR	1 = Enable this event to contribute to the port's port-write request.	R/WS	0x0
21:22	Reserved	Reserved	R	0x0
23	EG_DOH_COR	1 = Enable this event to contribute to the port's port-write request.	R/WS	0x0
24	EG_DOH_FATAL	1 = Enable this event to contribute to the port's port-write request.	R/WS	0x0
25	EG_DNFL_COR	1 = Enable this event to contribute to the port's port-write request.	R/WS	0x0
26	EG_DNFL_FATAL	1 = Enable this event to contribute to the port's port-write request.	R/WS	0x0
27	EG_DATA_OVERFLOW	1 = Enable this event to contribute to the port's port-write request.	R/WS	0x0
28	EG_CRQ_OVERFLOW	1 = Enable this event to contribute to the port's port-write request.	R/WS	0x0
29	Reserved	Reserved	R	0x0
30	EG_BAD_CHANNEL	1 = Enable this event to contribute to the port's port-write request.	R/WS	0x0
31	EG_BABBLE_PACKET	1 = Enable this event to contribute to the port's port-write request.	R/WS	0x0

18.11.6 RapidIO PBM Port Event Generate Register

This register generates PBM events for software verification. Note that setting “FATAL” bits sets the equivalent bits in the status register.

Register Name: RIO_PBM_SP_EVENT_GEN Reset Value: 0x0000_0000	Register Offset: 0x1069C

Bits	0	1	2	3	4	5	6	7
00:07	IG_DATA_COR	IG_DATA_UNCOR	IG_TAG_COR	IG_TAG_FATAL	IG_TFL_COR	IG_TFL_FATAL	IG_DOH_COR	IG_DOH_FATAL
08:15	IG_DNFL_COR	IG_DNFL_FATAL	Reserved					
16:23	Reserved			EG_DATA_COR	EG_DATA_UNCOR	Reserved		EG_DOH_COR
24:31	EG_DOH_FATAL	EG_DNFL_COR	EG_DNFL_FATAL	EG_DATA_OVERFLOW	EG_CRQ_OVERFLOW	Reserved	EG_BAD_CHANNEL	EG_BABBLE_PACKET

Bits	Name	Description	Type	Reset Value
0	IG_DATA_COR	1 = Set the corresponding bit in the RapidIO PBM Port Status Register.	R/W1S	0x0
1	IG_DATA_UNCOR	1 = Set the corresponding bit in the RapidIO PBM Port Status Register.	R/W1S	0x0
2	IG_TAG_COR	1 = Set the corresponding bit in the RapidIO PBM Port Status Register.	R/W1S	0x0
3	IG_TAG_FATAL	1 = Set the corresponding bit in the RapidIO PBM Port Status Register.	R/W1S	0x0
4	IG_TFL_COR	1 = Set the corresponding bit in the RapidIO PBM Port Status Register.	R/W1S	0x0
5	IG_TFL_FATAL	1 = Set the corresponding bit in the RapidIO PBM Port Status Register.	R/W1S	0x0
6	IG_DOH_COR	1 = Set the corresponding bit in the RapidIO PBM Port Status Register.	R/W1S	0x0
7	IG_DOH_FATAL	1 = Set the corresponding bit in the RapidIO PBM Port Status Register.	R/W1S	0x0
8	IG_DNFL_COR	1 = Set the corresponding bit in the RapidIO PBM Port Status Register.	R/W1S	0x0
9	IG_DNFL_FATAL	1 = Set the corresponding bit in the RapidIO PBM Port Status Register.	R/W1S	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
10:18	Reserved	Reserved	R	0x0
19	EG_DATA_COR	1 = Set the corresponding bit in the RapidIO PBM Port Status Register.	R/W1S	0x0
20	EG_DATA_UNCOR	1 = Set the corresponding bit in the RapidIO PBM Port Status Register.	R/W1S	0x0
21:22	Reserved	Reserved	R	0x0
23	EG_DOH_COR	1 = Set the corresponding bit in the RapidIO PBM Port Status Register.	R/W1S	0x0
24	EG_DOH_FATAL	1 = Set the corresponding bit in the RapidIO PBM Port Status Register.	R/W1S	0x0
25	EG_DNFL_COR	1 = Set the corresponding bit in the RapidIO PBM Port Status Register.	R/W1S	0x0
26	EG_DNFL_FATAL	1 = Set the corresponding bit in the RapidIO PBM Port Status Register.	R/W1S	0x0
27	EG_DATA_OVERFLOW	1 = Set the corresponding bit in the RapidIO PBM Port Status Register.	R/W1S	0x0
28	EG_CRQ_OVERFLOW	1 = Set the corresponding bit in the RapidIO PBM Port Status Register.	R/W1S	0x0
29	Reserved	Reserved	R	0x0
30	EG_BAD_CHANNEL	1 = Set the corresponding bit in the RapidIO PBM Port Status Register.	R/W1S	0x0
31	EG_BABBLE_PACKET	1 = Set the corresponding bit in the RapidIO PBM Port Status Register.	R/W1S	0x0

18.11.7 RapidIO PBM Port Ingress Watermarks 0 Register

This register allocates data nodes and tags for priority 0 packets. It must be changed only during a bootload process or while the port is held in reset.

Register Name: RIO_PBM_SP_IG_WATERMARK0 Reset Value: 0x003F_0048	Register Offset: 0x106B0

Bits	0	1	2	3	4	5	6	7
00:07	RESERVED						PRIO0CRF_WM	
08:15	PRIO0CRF_WM							
16:23	RESERVED						PRIO0_WM	
24:31	PRIO0_WM							

Bits	Name	Description	Type	Reset Value
0:5	RESERVED	Reserved	R	0x00
6:15	PRIO0CRF_WM	Priority 0+CRF packets are accepted if the number of free data nodes is greater than or equal to this value. Note: It is a programming error for this value to be greater than PRIO0_WM, or less than PRIO1_WM plus the size of the largest Priority 0 or 0+CRF packet used in the system.	R/WS	0x3F
16:21	RESERVED	Reserved	R	0x00
22:31	PRIO0_WM	Priority 0 packets are accepted if the number of free data nodes is greater than or equal to this value.	R/WS	0x48

18.11.8 RapidIO PBM Port Ingress Watermarks 1 Register

This register allocates data nodes and tags for priority 1 packets. This register must be changed only during a bootload process or while the port is held in reset.

Register Name: RIO_PBM_SP_IG_WATERMARK1 Reset Value: 0x002D_0036	Register Offset: 0x106B4

Bits	0	1	2	3	4	5	6	7
00:07	RESERVED						PRIO1CRF_WM	
08:15	PRIO1CRF_WM							
16:23	RESERVED						PRIO1_WM	
24:31	PRIO1_WM							

Bits	Name	Description	Type	Reset Value
0:5	RESERVED	Reserved	R	0x00
6:15	PRIO1CRF_WM	Priority 1+CRF packets are accepted if the number of free data nodes is greater than or equal to this value. Note: It is a programming error for this value to be greater than PRIO1_WM, or less than PRIO2_WM plus the size of the largest Priority 1 or 1+CRF packet used in the system.	R/WS	0x02D
16:21	RESERVED	Reserved	R	0x00
22:31	PRIO1_WM	Priority 1 packets are accepted if the number of free data nodes is greater than or equal to this value. Note: It is a programming error for this value to be less than PRIO1CRF_WM.	R/WS	0x036

18.11.9 RapidIO PBM Port Ingress Watermarks 2 Register

This register allocates data nodes and tags for priority 2 packets. This register must be changed only during a bootload process or while the port is held in reset.

Register Name: RIO_PBM_SP_IG_WATERMARK2 Reset Value: 0x001B_0024	Register Offset:106B8

Bits	0	1	2	3	4	5	6	7
00:07	RESERVED						PRIO2CRF_WM	
08:15	PRIO2CRF_WM							
16:23	RESERVED						PRIO2_WM	
24:31	PRIO2_WM							

Bits	Name	Description	Type	Reset Value
0:5	RESERVED	Reserved	R	0x00
6:15	PRIO2CRF_WM	Priority 2+CRF packets are accepted if the number of free data nodes is greater than or equal to this value. Note: It is a programming error for this value to be greater than PRIO2_WM, or less than PRIO3_WM plus the size of the largest Priority 2 or 2+CRF packet used in the system.	R/WS	0x01B
16:21	RESERVED	Reserved	R	0x00
22:31	PRIO2_WM	Priority 2 packets are accepted if the number of free data nodes is greater than or equal to this value. Note: It is a programming error for this value to be less than PRIO2CRF_WM.	R/WS	0x24

18.11.10 RapidIO PBM Port Ingress Watermarks 3 Register

This register allocates data nodes and tags for priority 3 packets. This register must be changed only during a bootload process or while the port is held in reset.

Register Name: RIO_PBM_SP_IG_WATERMARK3 Reset Value: 0x0009_0012	Register Offset: 0x106BC

Bits	0	1	2	3	4	5	6	7
00:07	RESERVED						PRIO3CRF_WM	
08:15	PRIO3CRF_WM							
16:23	RESERVED						PRIO3_WM	
24:31	PRIO3_WM							

Bits	Name	Description	Type	Reset Value
0:05	RESERVED	Reserved	R	0x00
6:15	PRIO3CRF_WM	Priority 3+CRF packets are accepted if the number of free data nodes is greater than or equal to this value. Note: It is a programming error for this value to be greater than PRIO3_WM or less than the largest Priority 3 or 3+CRF packet used by the system.	R/WS	0x009
16:21	RESERVED	Reserved	R	0x00
22:31	PRIO3_WM	Priority 3 packets are accepted if the number of free data nodes is greater than or equal to this value. Note: It is a programming error for this value to be less than PRIO3CRF_WM.	R/WS	0x12

18.12 Event Management Registers

18.12.1 IDT-Specific Event Management Block Header

This register identifies the following registers as the IDT RapidIO Event Management register block.

Register Name: RIO_EM_BH Reset Value: 0x010a_0000	Register Offset: 0x10900

Bits	0	1	2	3	4	5	6	7
00:07	NEXT_BLK_PTR							
08:15	NEXT_BLK_PTR							
16:23	BLK_REV				BLK_TYPE			
24:31	BLK_TYPE							

Bits	Name	Description	Type	Reset Value
0:15	NEXT_BLK_PTR	Pointer to the next IDT block, in units of 0x100 bytes. This points to the Port-Write Block Registers.	RS	0x010A
16:19	BLK_REV	IDT-defined revision of the block type in this device.	RS	0
20:31	BLK_TYPE	IDT-defined identifier for the register block type.	RS	0

18.12.2 RapidIO Event Management Interrupt Status Register

This register indicates the events, which were enabled lower in the hierarchy, that caused a RapidIO interrupt to be asserted.

Register Name: RIO_EM_INT_STAT Reset Value: 0x0000_0000	Register Offset: 0x10910

Bits	0	1	2	3	4	5	6	7
00:07	Reserved		PORT	LOG	RCS	MECS	Reserved	
08:15	Reserved							PW_RX
16:23	Reserved				IG_DATA_COR	IG_DATA_UNCOR	Reserved	LOCALOG
24:31	Reserved							

Bits	Name	Description	Type	Reset Value
0:1	Reserved	Reserved	R	0
2	PORT	A port-specific interrupt event has been detected. To determine if the port detected an event that uses the interrupt notification method, see RapidIO Event Management Interrupt Port Status Register.	R	0
3	LOG	An enabled Transport/Logical Layer error has been detected and reported. This field can cleared by software writing 0x0000_0000 to the RapidIO Logical/Transport Layer Error Detect CSR.	R	0
4	RCS	A reset request has been detected. To determine if the port detected a reset request, see the RapidIO Event Management Reset Request Port Status Register.	R	0
5	MECS	An MECS with an enabled *CMD* field (see RapidIO Event Management MECS Interrupt Enable Register) has been received by the port.	R	0
6:14	Reserved	Reserved	R	0
15	PW_RX	A port-write has been received This bit is set when PW_VAL in RapidIO Port-Write Reception Status CSR is set.	R	0
16:19	Reserved	Reserved	R	0
20	IG_DATA_COR	LLM detected and corrected a single-bit error on the data portion of a data node (only available if ECC enabled).	R/W1CS	0
21	IG_DATA_UNCOR	LLM detected a double-bit error on the data portion of a data node (only available if ECCenabled).	R/W1CS	0

(Continued)

Bits	Name	Description	Type	Reset Value
22	Reserved	Reserved	R	0
23	LOCALOG	Implementation-specific Logical/Transport Layer Error. If this bit is set, then one or more of the enabled errors in Tsi721’s RapidIO Local Logical/Transport Layer Error Detect CSR is set.	RS	0
24:31	Reserved	Reserved	R	0

18.12.3 RapidIO Event Management Interrupts Enable Register

This register controls which events can assert a RapidIO interrupt. Note that interrupts must also be enabled in the RapidIO Event Management Device Interrupt Enable Register.

Register Name: RIO_EM_INT_ENABLE Reset Value: 0x0000_0000	Register Offset: 0x10914

Bits	0	1	2	3	4	5	6	7
00:07	Reserved			LOG	Reserved	MECS	Reserved	
08:15	Reserved							PW_RX
16:23	Reserved				IG_DATA_ COR	IG_DATA_ UNCOR	Reserved	LOCALOG
24:31	Reserved							

Bits	Name	Description	Type	Reset Value
0:2	Reserved	Reserved	R	0
3	LOG	1 = Enable a logical layer event cause an interrupt.	R/WS	0
4	Reserved	Reserved	R	0
5	MECS	1 = Enable an interrupt to be raised when an MECS is received with a command value that is enabled in the RapidIO Event Management MECS Interrupt Enable Register.	R/WS	0
6:14	Reserved	Reserved	R	0
15	PW_RX	1 = Enable the reception of a port-write to cause an interrupt.	R/WS	0
16:19	Reserved	Reserved	R	0
20	IG_DATA_COR	1 = Enable this event to contribute to the interrupt request.	R/WS	0
21	IG_DATA_UNCOR	1 = Enable this event to contribute to the interrupt request.	R/WS	0
22	Reserved	Reserved	R	0
23	LOCALOG	1 = Enable an implementation-specific Logical/Transport Layer Error to cause an interrupt.	R/WS	0
24:31	Reserved	Reserved	R	0

18.12.4 RapidIO Event Management Interrupt Port Status Register

This register indicates there is an interrupt pending in the port.

Register Name: RIO_EM_INT_PORT_STAT Reset Value: 0x0000_0000	Register Offset: 0x10918

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

Bits	Name	Description	Type	Reset Value
0:30	Reserved	Reserved	R	0
31	IRQ_PENDING	Interrupt Pending Status To determine the cause, see the RapidIO PLM Port Event Status Register. If this bit is set it causes the PORT bit of the RapidIO Event Management Interrupt Status Register to be set. 0 = No interrupt pending for the port 1 = Interrupt pending for the port	R	0

18.12.5 RapidIO Event Management Port-Write Status Register

This register indicates the events, which were enabled lower in the hierarchy, that caused a port-write to be sent

Register Name: RIO_EM_PW_STAT Reset Value: 0x0000_0000	Register Offset: 0x10920

Bits	0	1	2	3	4	5	6	7
00:07	Reserved		PORT	LOG	RCS	Reserved		
08:15	Reserved							
16:23	Reserved				IG_DATA_ COR	IG_DATA_ UNCOR	Reserved	LOCALOG
24:31	Reserved							

Bits	Name	Description	Type	Reset Value
0:1	Reserved	Reserved	R	0
2	PORT	A port-specific event has been detected. To determine if the port detected an event that uses the port-write notification method, see RapidIO Event Management Port-Write Port Status Register.	R	0
3	LOG	An enabled Transport/Logical Layer event has been detected and reported. This field is cleared by software writing 0x0000_0000 to the RapidIO Logical/Transport Layer Error Detect CSR.	R	0
4	RCS	A reset request has been detected by the S-RIO port.	R	0
5:19	Reserved	Reserved	R	0
20	IG_DATA_COR	LLM detected and corrected a single-bit error on the data portion of a data node (only available if ECC enabled).	R/W1CS	0
21	IG_DATA_UNCO R	LLM detected a double-bit error on the data portion of a data node (only available if ECCenabled).	R/W1CS	0
22	Reserved	Reserved	R	0
23	LOCALOG	Implementation-specific local Logical/Transport Layer Error. If this bit is set, then one or more of the enabled errors in Tsi721's RapidIO Local Logical/Transport Layer Error Detect CSR is set. This bit is reported in port-writes.	RS	0
24:31	Reserved	Reserved	R	0

18.12.6 RapidIO Event Management Port-Write Enable Register

This register controls which events can cause a port-write to be sent. Note that port-writes must also be enabled in the RapidIO Event Management Device Port-Write Enable Register.

Register Name: RIO_EM_PW_ENABLE Reset Value: 0x1000_0000	Register Offset: 0x10924

Bits	0	1	2	3	4	5	6	7
00:07	Reserved			LOG	Reserved			
08:15	Reserved							
16:23	Reserved				IG_DATA_ COR	IG_DATA_ UNCOR	Reserved	LOCALOG
24:31	Reserved							

Bits	Name	Description	Type	Reset Value
0:2	Reserved	Reserved	R	0
3	LOG	1 = Enable a logical layer event to cause a port-write to be sent.	R/WS	1
4:19	Reserved	Reserved	R	0
20	IG_DATA_COR	1 = Enable this event to contribute to the port-write request.	R/WS	0
21	IG_DATA_UNCOR	1 = Enable this event to contribute to the port-write request.	R/WS	0
22	Reserved	Reserved	R	0
23	LOCALOG	1 = Enable an implementation-specific Logical/Transport Layer Error to cause a port-write to be sent.	R/WS	0
24:31	Reserved	Reserved	R	0

18.12.7 RapidIO Event Management Port-Write Port Status Register

This register indicates whether or not the port is requesting a port-write to be sent. To determine the cause, see the RapidIO PLM Port Event Status Register.

Register Name: RIO_EM_PW_PORT_STAT Reset Value: 0x0000_0000	Register Offset: 0x10928

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

Bits	Name	Description	Type	Reset Value
0:30	Reserved	Reserved	R	0
31	PW_PENDING	Port-write pending status. 0 = No port-write pending for the port 1 = port-write pending for the port If this bit is set it causes the PORT bit of RapidIO Event Management Port-Write Status Register to be set.	R	0

18.12.8 RapidIO Event Management Device Interrupt Enable Register

This register controls whether all interrupt notification is enabled or disabled.

Register Name: RIO_EM_DEV_INT_EN Reset Value: 0x0000_0000	Register Offset: 0x10930

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

Bits	Name	Description	Type	Reset Value
0:30	Reserved	Reserved	R	0
31	INT_EN	This bit controls whether the device interrupt line can be asserted. 0 = Interrupt line is de-asserted. 1 = Interrupt line can be asserted by enabled events.	R/WS	0

18.12.9 RapidIO Event Management Device Port-Write Enable Register

This register controls whether all port-write notification is enabled or disabled.

Register Name: RIO_EM_DEV_PW_EN Reset Value: 0x0000_0001	Register Offset: 0x10934

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

Bits	Name	Description	Type	Reset Value
0:30	Reserved	Reserved	R	0
31	PW_EN	This bit controls whether any port-writes can be sent by this device. 0 = Port-writes are not sent 1 = Port-writes can be sent when an enabled event is detected	R/WS	1

18.12.10 RapidIO Event Management MECS Status Register

This register captures MECS *CMD* field values that has been received by Tsi721 from the S-RIO port. The capture function of this register is continuously enabled and thus not affected by the RapidIO Event Management MECS Interrupt Enable Register.

o determine if the port received an MECS, see the RapidIO Event Management MECS Port Status Register.

Register Name: RIO_EM_MECS_STAT Reset Value: 0x0000_0000	Register Offset: 0x1093C

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

Bits	Name	Description	Type	Reset Value
0:23	Reserved	Reserved	R	0
24:31	CMD_STAT	This field indicates which MECS command was received by the S-RIO port. Bit 31 = Received an MECS with CMD value of 0 Bit 30 = Received an MECS with CMD value of 1 Bit 29 = Received an MECS with CMD value of 2 ... Bit 24 = Received an MECS with CMD value of 7 Note: MECSs with a CMD value of greater than 0 are not supported by the *RapidIO Specification (Rev. 2.1)*.	R/W1CS	0

18.12.11 RapidIO Event Management MECS Interrupt Enable Register

This register enables a received MECS with corresponding *CMD* field value, as captured in RapidIO Event Management MECS Status Register.CMD_STAT, to raise an interrupt request if RapidIO Event Management Interrupts Enable Register.MECS is also set.

Register Name: RIO_EM_MECS_INT_EN Reset Value: 0x0000_0001	Register Offset: 0x10940

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

Bits	Name	Description	Type	Reset Value
0:23	Reserved	Reserved	R	0
24:31	CMD_EN	Enables MECS with set *CMD* to raise an interrupt request: Bit 31 = CMD value of 0 (enabled by default) Bit 30 = CMD value of 1 (disabled by default) Bit 29 = CMD value of 2 (disabled by default) ... Bit 24 = CMD value of 7 (disabled by default) Note: MECSs with a CMD value of greater than 0 are not supported by the *RapidIO Specification (Rev. 2.1)*.	R/WS	0x01

18.12.12 RapidIO Event Management MECS Capture Out Register

When an MECS with a *CMD* field that is enabled in this register is received by the port, or is requested by the RapidIO Event Management MECS Request Register, an edge is generated on the MECS signal when configured as an output (see MECS_O in the Device Control Register).

Register Name: RIO_EM_MECS_CAP_EN Reset Value: 0x0000_0000 Location: LLM	Register Offset: 0x10944

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

Bits	Name	Description	Type	Reset Value
0:23	Reserved	Reserved	R	0
24:31	CMD_EN	Bit 31 = MECS with CMD value of 0 toggles MECS signal Bit 30 = MECS with CMD value of 1 toggles MECS signal Bit 29 = MECS with CMD value of 2 toggles MECS signal ... Bit 24 = MECS with CMD value of 7 toggles MECS signal Note: MECSs with a CMD value of greater than 0 are not supported by the *RapidIO Specification (Rev. 2.1)*.	R/WS	0

18.12.13 RapidIO Event Management MECS Trigger In Register

This register is associated with the MECS signal when the signal is configured as an input. The corresponding bit in the CMD_STAT field is set when an edge detected MECS signal and the associated CMD_EN bit is set. The edge will trigger the request of a MECS with corresponding *CMD* field to the S-RIO port.

Register Name: RIO_EM_MECS_TRIG_EN Reset Value: 0x0000_0000	Register Offset: 0x10948

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

Bits	Name	Description	Type	Reset Value
0:15	Reserved	Reserved	R	0
16:23	CMD_STAT	A bit is set when an edge is detected on the MECS signal: Bit 23 = Edge detected on MECS signal ANDing CMD_EN[0] Bit 22 = Edge detected on MECS signal ANDing CMD_EN[1] Bit 21 = Edge detected on MECS signal ANDing CMD_EN[2] ... Bit 16 = Edge detected on MECS signal ANDing CMD_EN[7]	R/W1CS	0
24:31	CMD_EN	Enables the associated bit of CMD_STAT to trigger a MECS request to the S-RIO port: Bit 31 = MECS with CMD value of 0 Bit 30 = MECS with CMD value of 1 Bit 29 = MECS with CMD value of 2 ... Bit 24 = MECS with CMD value of 7 Note: MECSs with a CMD value of greater than 0 are not supported by the *RapidIO Specification (Rev. 2.1)*.	R/WS	0

18.12.14 RapidIO Event Management MECS Request Register

This register enables one or many MECS of CMD values to be requested. The request is sent to the S-RIO port and to the MECS signal.

The port decides whether or not to send an MECS based on the value of the SUBSCRIPTION field in the RapidIO PLM Port MECS Forwarding Register. The MECS signal is controlled by the RapidIO Event Management MECS Capture Out Register.

Note: An MECS must not be requested at rates exceeding 2 MHz per CMD for a 4x link operating at 5 Gbaud or above. The maximum rate of 2 MHz must be pro-rated if the link width is decreased or if the baud rate is decreased.

Register Name: RIO_EM_MECS_REQ Reset Value: 0x0000_0000	Register Offset: 0x1094C

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

Bits	Name	Description	Type	Reset Value
0:22	Reserved	Reserved	R	0
23	SEND	1 = Initiates an MECS request for each *CMD* value that is set in the CMD field. This field is automatically cleared by hardware once the request has been signaled.	R/W1S	0
24:31	CMD	Setting this field indicates which MECS command(s) are to be requested: Bit 31 = Request MECS with CMD field value of 0 Bit 30 = Request MECS with CMD field value of 1 Bit 29 = Request MECS with CMD field value of 2 ... Bit 24 = Request MECS with CMD field value of 7 Note: MECSs with a CMD value of greater than 0 are not supported by the *RapidIO Specification (Rev. 2.1)*.	R/WS	0

18.12.15 RapidIO Event Management MECS Port Status Register

This register indicates that the S-RIO port has received an MECS (enabled or not). It is provided for informational purposes and can only be cleared directly by software.

Register Name: RIO_EM_MECS_PORT_STAT Reset Value: 0x0000_0000	Register Offset: 0x10950

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

Bits	Name	Description	Type	Reset Value
0:30	Reserved	Reserved	R	0
31	PORT	MECS Received status. 1 = MECS received on the port Specifics of the MECS are described in the RapidIO PLM Port Received MECS Status Register.	R/W1CS	0

18.12.16 RapidIO Event Management MECS and ECC Event Generate Register

This register generates Event Management MECS and ECC error events for software verification.

Register Name: RIO_EM_MECS_EVENT_GEN Reset Value: 0x0000_0000	Register Offset: 0x1095C

Bits	0	1	2	3	4	5	6	7
00:07	Reserved							
08:15	Reserved							
16:23	Reserved				IG_DATA_COR	IG_DATA_UNCOR	Reserved	
24:31	CMD_STAT							

Bits	Name	Description	Type	Reset Value
0:19	Reserved	Reserved	R	0
20	IG_DATA_COR	Enable reporting of correctable ECC errors.	R/W1S	0
21	IG_DATA_UNCOR	Enable reporting of uncorrectable ECC errors.	R/W1S	0
22:23	Reserved	Reserved	R	0
24:31	CMD_STAT	Writing 1 to any bit sets the corresponding bit in the RapidIO Event Management MECS Status Register.	R/W1S	0x00

18.12.17 RapidIO Event Management Reset Request Port Status Register

This register indicates whether or not the port received a reset request:

- Asserting an interrupt – The RST_INT_EN bit is 1 in the RapidIO Event Management Reset Request Interrupt Enable Register. The RCS bit is also 1 in the RapidIO Event Management Interrupt Status Register. To clear the event, clear the RST_REQ bit in the RapidIO PLM Port Event Status Register and then the RST_REQ bit in this register.
- Sending a port-write notification – The RST_PW_EN bit is 1 in the RapidIO Event Management Reset Request Port-Write Enable Register. The RCS bit is also 1 in the RapidIO Event Management Port-Write Status Register. To clear the event, clear the RST_REQ bit in the RapidIO PLM Port Event Status Register and then the RST_REQ bit in this register.
- Resetting the port due to RapidIO PLM Port Implementation Specific Control Register.PORT_SELF_RST – The RST_REQ bit is 0 in the RapidIO PLM Port Event Status Register. To clear the event, clear the RST_REQ bit in this register.

Register Name: RIO_EM_RST_PORT_STAT Reset Value: 0x0000_0000	Register Offset: 0x10960

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

Bits	Name	Description	Type	Reset Value
0:30	Reserved	Reserved	R	0
31	RST_REQ	Reset control symbol event received status. 0 = No reset event received on the port 1 = Reset event received on the port For more information, see RapidIO PLM Port Event Status Register.	R/W1CS	0

18.12.18 RapidIO Event Management Reset Request Interrupt Enable Register

This register controls whether the port generates an interrupt when it receives a reset request. For more information, see RapidIO Event Management Reset Request Port Status Register.

Register Name: RIO_EM_RST_INT_EN Reset Value: 0x0000_0000	Register Offset: 0x10968

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

Bits	Name	Description	Type	Reset Value
0:30	Reserved	Reserved	R	0
31	RST_INT_EN	Reset Request event interrupt enable. 0 = Do not assert interrupt for reset request received on the port 1 = Assert interrupt for reset request received on the port For more information, see RapidIO Event Management Reset Request Port Status Register.	R/WS	0

18.12.19 RapidIO Event Management Reset Request Port-Write Enable Register

This register controls whether the port generates a port-write to be sent when it receives a reset request. For more information, see RapidIO Event Management Reset Request Port Status Register.

Register Name: RIO_EM_RST_PW_EN Reset Value: 0x0000_0000	Register Offset: 0x10970

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

Bits	Name	Description	Type	Reset Value
0:30	Reserved	Reserved	R	0
31	RST_PW_EN	Reset Request Port-Write enable. 0 = Do not send port-write for reset request received on the port 1 = Send port-write for reset request received on the port For more information, see RapidIO Event Management Reset Request Port Status Register.	R/WS	0

18.13 Port-Write Block Registers

The port-write block contains registers for the control of port-write transmission. This block also contains registers for the reception of port-writes.

18.13.1 IDT-Specific RapidIO Port-Write Block Header

This register identifies the following registers as the IDT RapidIO Port-Write Management register block.

Register Name: RIO_PW_BH Reset Value: 0x010D_0000	Register Offset: 0x10A00

Bits	0	1	2	3	4	5	6	7
00:07	NEXT_BLK_PTR							
08:15	NEXT_BLK_PTR							
16:23	BLK_REV				BLK_TYPE			
24:31	BLK_TYPE							

Bits	Name	Description	Type	Reset Value
0:15	NEXT_BLK_PTR	Pointer to the next IDT block, in units of 0x100 bytes. This points to the LLM Registers.	RS	0x010D
16:19	BLK_REV	IDT-defined revision of the block type in this device.	RS	0
20:31	BLK_TYPE	IDT-defined identifier for the register block type.	RS	0

18.13.2 RapidIO Port-Write Control Register

This register controls aspects of port-write transmission and composition.

Register Name: RIO_PW_CTL Reset Value: 0x0000_0000	Register Offset: 0x10A04

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

Bits	Name	Description	Type	Reset Value
0:3	PW_TIMER	Port-Write Timer. The timer defines a 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. 0b0000 = Disabled. Port-write is sent once only 0b0001 = 103 us 0b0010 = 205 us 0b0100 = 410 us 0b1000 = 820 us Other values are reserved. Accuracy is +/-10% (see also RapidIO Port IP Prescalar for SRV_CLK Register). Note: Changing the value of this field while port-writes are in flight or pending requires these steps: 1. Disable port-write generation by clearing RapidIO Event Management Device Port-Write Enable Register.PW_EN 2. Update PW_TIMER 3. Clear all pending port-write events 4. Re-enable port-write generation by setting RapidIO Event Management Device Port-Write Enable Register.PW_EN When this procedure is completed, the next port-write generated will use the new timer value.	R/WS	0b0000
4:6	Reserved	Reserved	R	0
7	PWC_MODE	0b0 = Continuous port-write capture mode 0b1 = Reliable port-write capture mode	R/WS	0
8:31	Reserved	Reserved	R	0

18.13.3 RapidIO Port-Write Routing Register

This register defines whether a port-write is transmitted. By default, port-writes are transmitted on the S-RIO port.

Register Name: RIO_PW_ROUTE Reset Value: 0x0000_0001	Register Offset: 0x10A08

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

Bits	Name	Description	Type	Reset Value
0:30	Reserved	Reserved	R	0
31	PORT	Indicates if a port-write should be sent to the port. 0 = Do not send port-writes to the S-RIO port. 1 = Send port-writes to the S-RIO port.	R/WS	1

18.13.4 RapidIO Port-Write Reception Status CSR

This register indicates whether a port-write has been captured, and if a port-write had to be discarded due to lack of resources. Note that only 16 bytes of payload capture is supported.

Register Name: RIO_PW_RX_STAT Reset Value: 0x0000_0000	Register Offset: 0x10A10

Bits	0	1	2	3	4	5	6	7
00:07	Reserved							
08:15	Reserved							
16:23	WR_SIZE				Reserved			WDPTR
24:31	Reserved				PW_SHORT	PW_TRUNC	PW_DISC	PW_VAL

Bits	Name	Description	Type	Reset Value
0:15	Reserved	Reserved	R	0
16:19	WR_SIZE	WR_SIZE, in combination with WDPTR, determine the maximum size of the port-write that has been received. For more information on the definition and use of WR_SIZE, see the *RapidIO Interconnect Specification (Revision 2.1)* Part 1: Logical I/O. This field is locked and valid when PW_VAL is set.	R/WS	0
20:22	Reserved	Reserved	R	0
23	WDPTR	WDPTR, in combination with WRSIZE, determine the maximum size of the port-write that has been received. For more information on the definition and use of WPTR, see the *RapidIO Interconnect Specification (Revision 2.1)* Part 1: Logical I/O. This field is locked and valid when PW_VAL is set.	R/WS	0
24:27	Reserved	Reserved	R	0
28	PW_SHORT	This bit is set when the captured port-write's payload is shorter than 16 bytes. The RapidIO Port-Write Reception Capture CSR {0..3} pads the payload to 16 bytes with zeros. This field is locked and valid when PW_VAL is set	R/WS	0
29	PW_TRUNC	This field is set when the captured port-write's payload is longer than 16 bytes. The RapidIO Port-Write Reception Capture CSR {0..3} truncates the payload to 16 bytes. This field is locked and valid when PW_VAL is set	R/WS	0

(Continued)

Bits	Name	Description	Type	Reset Value
30	PW_DISC	A port-write was discarded. Clearing this bit has no other effect.	R/W1CS	0
31	PW_VAL	The port-write data registers contain valid data. When set, the following fields are locked: WR_SIZE, WDPTR, PW_SHORT, and PW_TRUNC. Clearing this bit unlocks the fields but has no other effect. This bit can generate an interrupt using RapidIO Event Management Interrupt Status Register.PW_RX.	R/W1CS	0

18.13.5 RapidIO Port-Write Reception Event Generate Register

This register generates port-write reception events for software verification.

Register Name: RIO_PW_RX_EVENT_GEN Reset Value: 0x0000_0000	Register Offset: 0x10A14

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

Bits	Name	Description	Type	Reset Value
0:29	Reserved	Reserved	R	0
30	PW_DISC	1 = Set the corresponding bit in the RapidIO Port-Write Reception Status CSR.	R/W1S	0x0
31	PW_VAL	1 = Set the corresponding bit in the RapidIO Port-Write Reception Status CSR.	R/W1S	0x0

18.13.6 RapidIO Port-Write Reception Capture CSR {0..3}

When a port-write maintenance request is received, the payload is captured in these registers, and an event is captured. This register is locked and all subsequent port-write-In requests are optionally discarded. The capture registers are unlocked when software clears the PW_VAL field in the RapidIO Port-Write Reception Status CSR. If the port-write maintenance request payload is longer than 16 bytes, the payload is truncated and only the first 16 bytes are captured. This is reported in the PW_TRUNC field in the RapidIO Port-Write Reception Status CSR. If the port-write maintenance request payload is shorter than 16 bytes, it is padded with zeros. This is reported in the PW_SHORT field in the RapidIO Port-Write Reception Status CSR.

Register Name: RIO_PW_RX_CAPT{0..3} Reset Value: 0x0000_0000	Register Offset: 0x10A20 += 4

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

Bits	Name	Description	Type	Reset Value
0:31	PW_CAPT	Port-Write payload, payload offset associated with register number.	R/WS	0

18.14 LLM Registers

18.14.1 IDT-Specific LLM Module Block Header

This register identifies the following registers as the IDT Fabric Module Block.

Register Name: RIO_LLM_BH Reset Value: 0x010E_0000	Register Offset: 0x10D00

Bits	0	1	2	3	4	5	6	7
00:07	NEXT_BLK_PTR							
08:15	NEXT_BLK_PTR							
16:23	BLK_REV				BLK_TYPE			
24:31	BLK_TYPE							

Bits	Name	Description	Type	Reset Value
0:15	NEXT_BLK_PTR	Pointer to the next IDT block, in units of 0x100 bytes.	RS	0x010E
16:19	BLK_REV	IDT-defined revision of the block type in this device.	RS	0
20:31	BLK_TYPE	IDT-defined identifier for the register block type.	RS	0

18.14.2 RapidIO Maintenance Write Request Restriction Control Register

This register controls whether the S-RIO port can perform register write transactions.

Register Name: RIO_MTC_WR_RESTRICT Reset Value: 0x0000_0000	Register Offset: 0x10D10

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

Bits	Name	Description	Type	Reset Value
0:30	Reserved	Reserved	R	0
31	WR_DIS	Disable RapidIO maintenance write access to registers for the S-RIO port. 0 = Maintenance writes are processed 1 = Maintenance writes are dropped.	R/WS	0

18.14.3 RapidIO Maintenance Port-Write Request Restriction Control Register

This register controls whether the S-RIO port performs port-write transactions.

Register Name: RIO_MTC_PWR_RESTRICT Reset Value: 0x0000_0000	Register Offset: 0x10D14

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

Bits	Name	Description	Type	Reset Value
0:30	Reserved	N/A	R	0
31	PWR_DIS	Disable RapidIO maintenance port-write access to registers for the port. 0 = Maintenance port-writes are processed 1 = Maintenance port-writes are dropped.	R/WS	0

18.14.4 RapidIO Maintenance Read Request Restriction Control Register

This register controls whether the S-RIO port can perform register read transactions.

Register Name: RIO_MTC_RD_RESTRICT Reset Value: 0x0000_0000	Register Offset: 0x10D18

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

Bits	Name	Description	Type	Reset Value
0:30	Reserved	Reserved	R	0
31	RD_DIS	Disable RapidIO maintenance read access to registers for the S-RIO port. 0 = Maintenance reads are processed 1 = Maintenance reads are dropped	R/WS	0

18.14.5 RapidIO Whiteboard CSR

This register is provided for whiteboarding purposes. It does not impact the design in any way.

Register Name: RIO_WHITEBOARD Reset Value: 0x0000_0000	Register Offset: 0x10D24

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

Bits	Name	Description	Type	Reset Value
0:31	SCRATCH	The Scratch field can be read and written with any value. The value written does not affect Tsi721's operation.	R/WS	0

18.14.6 RapidIO Port IP Prescalar for SRV_CLK Register

This register defines a prescalar for “ip_clk” to handle different clock frequencies. The scaled clock provided by this register is used by the following timers:

- RapidIO Port Link Timeout Control CSR
- RapidIO PLM Port Discovery Timer Register
- RapidIO PLM Port Silence Timer Register
- RapidIO PLM Port Implementation Specific Control Register
- RapidIO Port Error Rate CSR
- RapidIO Port-Write Control Register

Register Name: RIO_PRESCALAR_SRV_CLK Reset Value: 0x0000_001F	Register Offset: 0x10D30

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

Bits	Name	Description	Type	Reset Value
0:23	Reserved	Reserved	R	0
24:31	PRESCALAR_SRV_CLK	The default value of 31 is for an ip_clk frequency of 312.5 MHz. For different frequencies of ip_clk, use the formula [ip_clk_frequency (in MHz) /10 rounded to the nearest integer.	R/WS	0x1F

Table 91: PRESCALAR_SRV_CLK Values for Various ip_clk Frequencies

IP_CLK Frequency (MHz)	IP_CLK Period (uS)	PRESCALAR_SRV_CLK	SRV_CLK period
312.5	0.003	31	0.0992
307.2	0.003	31	0.1009
250	0.004	25	0.1000
245.76	0.004	25	0.1017
156.25	0.006	16	0.1024
153.6	0.007	15	0.0977
125	0.008	13	0.1040
78.125	0.013	8	0.1024
76.8	0.013	8	0.1042
62.5	0.016	6	0.0960

Table 92: Tsi721 ip_clk Frequencies

S-RIO Link Rate (Gbaud)	ip_clk frequency (MHz)	SerDes Parallel Bus Width (bit)
1.25	125	10
2.5	125	20
3.125	156.25	20
5.0	250	20

18.14.7 RapidIO Register Reset Control CSR

This register configures register reset behavior.

Register Name: RIO_REG_RST_CTL Reset Value: 0x0000	Register Offset: 0x10D34

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

Bits	Name	Description	Type	Reset Value
0:30	Reserved	Reserved	R	0
31	CLEAR_STICKY	Allows the SELF_RST and PWDN_PORT resets to clear sticky bits in addition to the normal configuration registers. For more information, see RapidIO PLM Port Implementation Specific Control Register.	R/WS	0

18.14.8 RapidIO Local Logical/Transport Layer Error Detect CSR

This register is not to be confused with a similar register, RapidIO Logical/Transport Layer Error Detect CSR, which is implemented in the standard RapidIO offset.

When a Logical or Transport Layer error is detected *within* Tsi721, the appropriate error bit is set by the hardware in this register unless a previously enabled event has locked the register. If the corresponding enable bit is also set in the RapidIO Local Logical/Transport Layer Error Enable CSR (at the time of the event), this register locks, and the appropriate information is saved in the Local Logical/Transport Layer Error Capture Registers (see registers starting at RapidIO Local Logical/Transport Layer High Address Capture CSR) which are also locked. While the register is locked, no other error bit in this register can be set by subsequent errors or by software. Software must write a zero to the appropriate error bit(s) or enable bits to unlock this register and the Logical/Transport Layer Error Capture Registers. Multiple bits can get set in the register if simultaneous errors are detected in the same cycle that the errors are logged.

This register and the Local Logical/Transport Layer Error Capture Registers are writable by software to allow software debug of the system error recovery methods. For software debug, software must enable an error; write to the capture registers with the desired address and deviceID information; then write to this register to set an error bit which locks the registers. When the error detect bit is set by software, the hardware will inform the system software of the error using its standard error reporting method. After the error is reported, system software can read and clear this register in order to complete its error handling protocol testing.

Register Name: RIO_LOCAL_ERR_DET Reset Value: 0x0000_0000	Register Offset: 0x10D48

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

Bits	Name	Description	Type	Reset Value
0:4	Reserved	Reserved	R	0
5	ILL_ID	Illegal transaction target error Received a packet that contained a destID that is not defined for this processing element. Endpoints with multiple ports and a built-in switch function may not report this as an error (Transport) (optional) (switch or endpoint device). This bit can be set by the Tsi721 due to: • Received a packet with a TT value of either 0b10 or 0b11 • destID did not match any enabled BRR nor the RapidIO Base deviceID CSR.	R/WS	0
6:8	Reserved	Reserved	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
9	ILL_TYPE	Unsupported Transaction A transaction is received that is not supported by the Tsi721: • Tsi721 asserts this error for a maintenance packet that has a ttype between 5 and 15, inclusive. • Tsi721 also asserts this error for packets that are filtered (discarded) by the TLMi's inbound packet filter. • LLM received a maintenance read/write request from the S-RIO port when it was locked out	R/WS	0
10:31	Reserved	Reserved	R	0

18.14.9 RapidIO Local Logical/Transport Layer Error Enable CSR

This register contains the bits that control if an error condition locks the RapidIO Local Logical/Transport Layer Error Detect CSR and Capture registers (see registers starting at RapidIO Local Logical/Transport Layer High Address Capture CSR), and is reported to the system host through a port-write request or an interrupt.

Register Name: RIO_LOCAL_ERR_EN Reset Value: 0x0000_0000	Register Offset: 0x10D4C

Bits	0	1	2	3	4	5	6	7
00:07	Reserved					ILL_ID_EN	Reserved	
08:15	Reserved	ILL_TYPE_ EN	Reserved					
16:23	Reserved							
24:31	Reserved							

Bits	Name	Description	Type	Reset Value
0:4	Reserved	Reserved	R	0
5	ILL_ID_EN	Enable reporting of illegal transaction target errors.	R/WS	0
6:8	Reserved	Reserved	R	0
9	ILL_TYPE_EN	Enable reporting of unsupported transaction error enable.	R/WS	0
10:31	Reserved	Reserved	R	0

Clearing the enable bit in this register unlocks RapidIO Local Logical/Transport Layer Error Detect CSR and RapidIO Local Logical/Transport Layer High Address Capture CSR if the associated bit in the RapidIO Local Logical/Transport Layer Error Detect CSR is set.

Setting the enable bit in this register after the associated bit in the RapidIO Local Logical/Transport Layer Error Detect CSR was set – that is, after the event – will not lock either the RapidIO Local Logical/Transport Layer Error Detect CSR nor the RapidIO Local Logical/Transport Layer High Address Capture CSR. However it causes the event to be reported, if enabled, through the standard event reporting method.

18.14.10 RapidIO Local Logical/Transport Layer High Address Capture CSR

This register contains error information. It is locked when a local Logical/Transport error is detected and the corresponding enable bit is set. The contents of this register are only valid when the device is using 50- or 66-bit addressing. For more information, see RapidIO Local Logical/Transport Layer Address Capture CSR.

Register Name: RIO_LOCAL_H_ADDR_CAPT Reset Value: 0x0000_0000	Register Offset: 0x10D50

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

Bits	Name	Description	Type	Reset Value
0:31	ADDR[0:31]	Most significant bits of address associated with the error (for requests, not applicable for responses or ftype 8 transactions). Address bits not used by the device are set to zero.	R/WS	0

18.14.11 RapidIO Local Logical/Transport Layer Address Capture CSR

This register contains error information. It is locked when a local Logical/Transport error is detected and the corresponding enable bit is set.

The RapidIO Local Logical/Transport Layer Error Detect CSR and the Logical/Transport Layer Error Capture Registers are writable by software to allow software debug of the system error recovery methods. For software debug, software must write to the Logical/Transport Layer Error Capture registers with the desired address and deviceID information, then write to the RapidIO Local Logical/Transport Layer Error Detect CSR to set an error bit and lock the registers. When an error detect bit is set, hardware informs the system software of the error using its standard error reporting method. After the error is reported, system software can read and clear registers as required to complete its error handling protocol testing.

Register Name: RIO_LOCAL_ADDR_CAPT Reset Value: 0x0000_0000	Register Offset: 0x10D54

Bits	0	1	2	3	4	5	6	7
00:07	ADDR[32:60]							
08:15	ADDR[32:60]							
16:23	ADDR[32:60]							
24:31	ADDR[32:60]					Reserved	XAMSBS	

Bits	Name	Description	Type	Reset Value
0:28	ADDR[32:60]	Least significant 29 bits of address associated with the error. For ftype 2, 5, and 6, the 29 least significant bits of the received address are captured. For ftype 8, the 21 bits of the configuration offset, padded with 8 bits of zeros is captured (for ftype8, ttype 2, 3 and 4, which do not have a configuration offset, 21 bits from the reserved field which immediately follows the hop count are captured). For all other ftype and ttype combinations, this field is set to zero.	R/WS	0
29	Reserved	Reserved	R	0
30:31	XAMSBS	Extended address bits of the address associated with the error (for requests, not applicable for responses or maintenance transactions)	R/WS	0

18.14.12 RapidIO Local Logical/Transport Layer deviceID Capture CSR

This register contains error information. It is locked when an error is detected and the corresponding enable bit is set.

The RapidIO Local Logical/Transport Layer Error Detect CSR and the Logical/Transport Layer Error Capture Registers are writable by software to allow software debug of the system error recovery methods. For software debug, software must write to the Logical/Transport Layer Error Capture registers with the desired address and deviceID information, then write to the RapidIO Local Logical/Transport Layer Error Detect CSR to set an error bit and lock the registers. When an error detect bit is set, hardware informs the system software of the error using its standard error reporting method. After the error is reported, system software can read and clear registers as required to complete its error handling protocol testing.

Register Name: RIO_LOCAL_ID_CAPT Reset Value: 0x0000_0000	Register Offset: 0x10D58

Bits	0	1	2	3	4	5	6	7
00:07	MSB_DEST_ID							
08:15	DEST_ID							
16:23	MSB_SRC_ID							
24:31	SRC_ID							

Bits	Name	Description	Type	Reset Value
0:7	MSB_DEST_ID	Most significant byte of destID associated with the error (Large transport system only)	R/WS	0
8:15	DEST_ID	destID associated with the error	R/WS	0
16:23	MSB_SRC_ID	Most significant byte of srcID associated with the error (Large transport system only)	R/WS	0
24:31	SRC_ID	srcID associated with the error	R/WS	0

18.14.13 RapidIO Local Logical/Transport Layer Control Capture CSR

This register contains error information. It is locked when an error is detected and the corresponding enable bit is set.

The RapidIO Local Logical/Transport Layer Error Detect CSR and the Logical/Transport Layer Error Capture Registers are writable by software to allow software debug of the system error recovery methods. For software debug, software must write to the Logical/Transport Layer Error Capture registers with the desired address and deviceID information, then write to the RapidIO Local Logical/Transport Layer Error Detect CSR to set an error bit and lock the registers. When an error detect bit is set, hardware informs the system software of the error using its standard error reporting method. After the error is reported, system software can read and clear registers as required to complete its error handling protocol testing.

Register Name: RIO_LOCAL_CTRL_CAPT Reset Value: 0x0000_0000	Register Offset: 0x10D5C

Bits	0	1	2	3	4	5	6	7
00:07	FTYPE				TTYPE			
08:15	MESSAGE_INFO							
16:23	Reserved							
24:31	Reserved							

Bits	Name	Description	Type	Reset Value
0:3	FTYPE	ftype associated with the error	R/WS	0
4:7	TTYPE	ttype associated with the error	R/WS	0
8:15	MESSAGE_INFO	Messages are not generated by Tsi721 and message information is not logged	R/WS	0
16:31	Reserved	Reserved	R	0

18.15 Fabric Module Registers

18.15.1 IDT-Specific Fabric Module Block Header

This register identifies the following registers as the IDT Fabric Module Block. This register is not used by the Tsi721.

Register Name: RIO_FABRIC_BH Reset Value: 0x0120_0000	Register Offset: 0x10E00

Bits	0	1	2	3	4	5	6	7
00:07	NEXT_BLK_PTR							
08:15	NEXT_BLK_PTR							
16:23	BLK_REV				BLK_TYPE			
24:31	BLK_TYPE							

Bits	Name	Description	Type	Reset Value
0:15	NEXT_BLK_PTR	Pointer to the next IDT block, in units of 0x100 bytes. This points to the PRBS/BERT Control Registers.	RS	0x0120
16:19	BLK_REV	IDT-defined revision of the block type in this device.	RS	0
20:31	BLK_TYPE	IDT-defined identifier for the register block type.	RS	0

18.16 PRBS/BERT Control Registers

18.16.1 IDT-Specific PRBS/BERT Block Header

This register identifies the following registers as the PRBS/BERT block header.

Register Name: RIO_PRBS_BH Reset Value: 0x0000_0000	Register Offset: 0x12000

Bits	0	1	2	3	4	5	6	7
00:07	NEXT_BLK_PTR							
08:15	NEXT_BLK_PTR							
16:23	BLK_REV				BLK_TYPE			
24:31	BLK_TYPE							

Bits	Name	Description	Type	Reset Value
0:15	NEXT_BLK_PTR	Pointer to the next IDT block, in units of 0x100 bytes.	RS	0x0000
16:19	BLK_REV	IDT-defined revision of the block type in this device.	RS	0x0
20:31	BLK_TYPE	IDT-defined identifier for the register block type.	RS	0x000

18.16.2 RapidIO PRBS Lane {0..3} Control Register

For more information on the use of this register, see Bit Error Rate Testing (BERT).

Register Name:RIO_PRBS_LANE{0..3}_CTRL Reset Value: 0x0000_0000	Register Offset: 0x12004+= 10

Bits	0	1	2	3	4	5	6	7
00:07	PATTERN				UNI	TRAIN	ENABLE	TRANSMIT
08:15	Reserved							
16:23	Reserved							
24:31	Reserved							

Bits	Name	Description	Type	Reset Value
0:3	PATTERN	Channel 3 Transmitter Pattern Select 0b0000 = $x^{23}+x^{18}+1$ 0b0001 = $x^{31}+x^{28}+1$ 0b0011 = $x^{10}+x^{7}+1$ 0b0100 = $x^{15}+x^{14}+1$ 0b0101 = $x^{7}+x^{6}+1$ 0b0110 = 10-bit pattern from RapidIO PRBS Lane {0..3} PRBS Seed Register.SEED[22:31] 0b0111 = Balanced pattern – fixed 10-bit pattern and its inverse from RapidIO PRBS Lane {0..3} PRBS Seed Register.SEED[22:31] 0b1000 = 40-bit pattern (10'b0, 10-bit fixed pattern, 10'b3FF, 10-bit fixed pattern) using RapidIO PRBS Lane {0..3} PRBS Seed Register.SEED[22:31] and its inverse for the two 10-bit fixed patterns 0b1001 = Alternating pattern of two 10-bit code groups using RapidIO PRBS Lane {0..3} PRBS Seed Register.SEED[22:31] and SEED[12:21] as the two 10-bit code groups All other values are reserved. Note: Error detection is not provided for fixed-patterns.	R/WS	0
4	UNI	PRBS Unidirectional BERT Mode When PRBS is enabled, this bit enables testing with a non S-RIO based BERT function. 10-bit PRBS will be used instead of 8-bit PRBS and the 8b/10b encoder will be bypassed 0 = Enable 8-bit PRBS with 8b/10b encoding 1 = Enable 10-bit PRBS without 8b/10b encoding	R/WS	0

(Continued)

Bits	Name	Description	Type	Reset Value
5	TRAIN	PRBS Receive Training Mode 0 = Disable PRBS training mode. 1 = Enable PRBS receive training mode. In training mode, no errors are reported.	R/WS	0
6	ENABLE	Enable PRBS transmitter and receiver and select data source for transmission. 0 = Normal data – PRBS transmitter and receiver disabled 1 = PRBS data – PRBS transmitter and receiver enabled	R/WS	0
7	TRANSMIT	Enable load of initial transmitter PRBS seed from CSR and enable PRBS receiver pattern checker. 0 to 1 transition = Load transmitter PRBS seed from CSR. 0 = Disable PRBS receiver pattern checker. 1 = Enable PRBS receiver pattern checker.	R/WS	0
8:31	Reserved	Reserved	R	0

18.16.3 RapidIO PRBS Lane {0..3} PRBS Seed Register

Register Name:RIO_PRBS_LANE{0..3}_SEED Reset Value: 0xFFFF_FFFF	Register Offset: 0x12008+= 10

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

Bits	Name	Description	Type	Reset Value
0:31	SEED	Seed value for PRBS generation. The same is used for the polynomial seed and as the actual data as selected by the value of RapidIO PRBS Lane {0..3} Control Register.PATTERN.	R/WS	0xFFFF_FFFF

18.16.4 RapidIO PRBS Lane {0..3} Error Count Register

Register Name:RIO_PRBS_LANE{0..3}_ERR_COUNT Reset Value: 0x0000_0000	Register Offset: 0x1200C+= 10

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

Bits	Name	Description	Type	Reset Value
0:31	COUNT	PRBS error counter for this lane. This counter is non-saturating and clear on read. Each bit detected to be in error increments the counter. Note: Errors are not counted for fixed pattern BERT.-	R	0x0000_0000

19. Mapping Engine Registers

Topics discussed include the following:

- PC2SR Register Map
- SR2PC Register Map
- PC2SR Registers
- SR2PC Registers

19.1 PC2SR Register Map

For registers defined for multiple channels such as Outbound Window Lower Base Register {0..7}, the register offset between two adjacent channels is 32; that is, 0x40000 +=20, register for channel 0 has address 0x40000, += denotes the register offset between channels, and 20 is in hexidecimal.

Table 93: PC2SR Register Map

Offset	Register Name	See
0x40000 +=20	OBWINLB{0..7}	Outbound Window Lower Base Register {0..7}
0x40004 +=20	OBWINUB{0..7}	Outbound Window Upper Base Register {0..7}
0x40008 +=20	OBWINSZ{0..7}	Outbound Window Size Register {0..7}
0x41300	ZONE_SEL	Outbound Window Lookup Table Zone Select Register
0x41304	LUT_DATA0	Outbound Window Lookup Table Data 0 Register
0x41308	LUT_DATA1	Outbound Window Lookup Table Data 1 Register
0x4130C	LUT_DATA2	Outbound Window Lookup Table Data 2 Register
0x41310	PC2SR_INTE	PC2SR Interrupt Enable CSR
0x41314	PC2SR_INT	PC2SR Interrupt CSR
0x41318	PC2SR_INTSET	PC2SR Interrupt Set Register
0x4131C	PC2SR_ECC_LOG	PC2SR ECC Log Register
0x41404	CPL_SMSG_CNT	Received Completion Count for Messaging Engine Register
0x41408	CPL_BDMA_CNT	Received Completion Count for Block DMA Engine Register
0x4140C	RXTLP_BRG_CNT	Received Bridging TLP Count Register
0x41410	TXPKT_SMSG_CNT	Sent Packet Count of Messaging Engine Register

Table 93: PC2SR Register Map *(Continued)*

Offset	Register Name	See
0x41414	TXPKT_BDMA_CNT	Sent Packet Count of Block DMA Engine Register
0x41418	TXPKT_BRG_CNT	Sent Bridging Packet Count Register

19.2 SR2PC Register Map

For registers defined for multiple MSI-X vectors such as the MSI-X Pending Bit Array Upper Register, the register offset between two adjacent vectors is 16; that is, in 0x2C000 +=10, register for vector 0 has address 0x2C000, += denotes the register offset between vectors, 10 is in hexidecimal.

For registers defined for multiple channels, the register address is defined similar to the Inbound Doorbell Queue Control Register {0..7}, where 0x20000 +=1000 means register for channel 0 has address 0x20000, += denotes the register offset between two adjacent channels, 1000 is in hexidecimal.

Table 94: SR2PC Register Map

Offset	Register Name	See
0x20000 +=1000	IDQ_CTL{0..7}	Inbound Doorbell Queue Control Register {0..7}
0x20004 +=1000	IDQ_STS{0..7}	Inbound Doorbell Queue Status Register {0..7}
0x20008 +=1000	IDQ_MASK{0..7}	Inbound Doorbell Classification Register {0..7}
0x2000C +=1000	IDQ_RP{0..7}	Inbound Doorbell Queue Read Pointer Register {0..7}
0x20010 +=1000	IDQ_WP{0..7}	Inbound Doorbell Queue Write Pointer Register {0..7}
0x20014 +=1000	IDQ_BASEL{0..7}	Inbound Doorbell Queue Lower Base Register {0..7}
0x20018 +=1000	IDQ_BASEU{0..7}	Inbound Doorbell Queue Upper Base Register {0..7}
0x2001C +=1000	IDQ_SIZE{0..7}	Inbound Doorbell Queue Size Register {0..7}
0x20040 +=1000	SR_CH{0..7}INT	SR2PC Channel Interrupt Register {0..7}
0x20044 +=1000	SR_CH{0..7}INTE	SR2PC Channel Interrupt Enable Register {0..7}
0x20048 +=1000	SR_CH{0..7}INSET	SR2PC Channel Interrupt Set Register {0..7}
0x20100 +=1000	ODB_CNT{0..7}	Outbound Doorbell Count {0..7} Register
0x20104 +=1000	ODB_LOG_DAT0{0..7}	Outbound Doorbell Response Log Buffer Data 0 Register {0..7}
0x20108 +=1000	ODB_LOG_DAT1{0..7}	Outbound Doorbell Response Log Buffer Data 1 Register {0..7}
0x2010C +=1000	ODB_LOG_DAT2{0..7}	Outbound Doorbell Response Log Buffer Data 2 Register {0..7}
0x20120 +=1000	ODB_LOG_DAT3{0..7}	Outbound Doorbell Response Log Buffer Data 3 Register {0..7}
0x20124 +=1000	ODB{0..7}LOGSTS	SR2PC Outbound Doorbell Response Log Buffer Status Register {0..7}

Table 94: SR2PC Register Map *(Continued)*

Offset	Register Name	See
0x29000 +=20	IBWIN_LB{0..7}	Inbound Window Lower Base Register{0..7}
0x29004 +=20	IBWIN_UB{0..7}	Inbound Window Upper Base Register{0..7}
0x29008 +=20	IBWIN_SZ{0..7}	Inbound Window Size Register{0..7}
0x2900C +=20	IBWIN_TLA {0..7}	Inbound Window Translated Lower Address Register {0..7}
0x29010 +=20	IBWIN_TUA {0..7}	Inbound Window Translated Upper Address {0..7}
0x29800	SR2PC_GEN_INTE	SR2PC General Interrupt Enable CSR
0x29804	SR2PC_PWE	SR2PC Port-Write Enable CSR
0x29808	SR2PC_GEN_INT	SR2PC General Interrupt CSR
0x2980C	SR2PC_GEN_INTSET	SR2PC General Interrupt Set CSR
0x29810	SR2PC_CORR_ECC_LOG	SR2PC Correctable ECC Log Register
0x29814	SR2PC_UNCORR_ECC_LOG	SR2PC Uncorrectable ECC Log Register
0x29820	SR2PC_PCIE_PS	SR2PC PCIe Port State Register
0x29824	LOGBUF_STS	SR2PC Log Buffer Status Register
0x29840	DEV_INTE	Tsi721 Interrupt Enable CSR
0x29844	DEV_INT	Tsi721 Interrupt CSR
0x2984C	DEV_CH_INTE	Tsi721 Interrupt CSR
0x29850	DEV_CH_INT	Tsi721 Channel Interrupt CSR
0x29858	INT_MOD	Interrupt Moderation Register
0x29900	RXPKT_SMSG_CNT	Received Packet Count for Messaging Engine Register
0x29904	RXRSP_BDMA_CNT	Received Response Count for Block DMA Engine Register
0x29908	RXPKT_BRG_CNT	Received Bridging Packet Count Register
0x2990C	TXTLP_SMSG_CNT	Sent TLP Count of Messaging Engine Register
0x29910	TXTLP_BDMA_CNT	Sent TLP Count of Block DMA Engine Register
0x29914	TXTLP_BRG_CNT	Sent Bridging TLP Count Register
0x2991C	BRG_PKT_ERR_CNT	Received Bridging Packet Error Count Register
0x29A00	MWR_CNT	Maintenance Write Count Register
0x29A04	NWR_CNT	NWRITE_R Count Register
0x29A08	MWR_LOG_DAT0	Maintenance Write Response Log Buffer Data 0 Register

Table 94: SR2PC Register Map *(Continued)*

Offset	Register Name	See
0x29A0C	MWR_LOG_DAT1	Maintenance Write Response Log Buffer Data 1 Register
0x29A10	MWR_LOG_DAT2	Maintenance Write Response Log Buffer Data 2 Register
0x29A14	MWR_LOG_DAT3	Maintenance Write Response Log Buffer Data 3 Register
0x29A18	NWR_LOG_DAT0	NWRITE_R Response Log Buffer Data 0 Register
0x29A1C	NWR_LOG_DAT1	NWRITE_R Response Log Buffer Data 1 Register
0x29A20	NWR_LOG_DAT2	NWRITE_R Response Log Buffer Data 2 Register
0x29A24	NWR_LOG_DAT3	NWRITE_R Response Log Buffer Data 3 Register
0x29B00	PC_ARB	MSI-X Pending Bit Array Lower Register
0x2A000	MSIX_PBAL	MSI-X Pending Bit Array Lower Register
0x2A004	MSIX_PBAM	MSI-X Pending Bit Array Middle Register
0x2A008	MSIX_PBAU	MSI-X Pending Bit Array Upper Register
0x2C000 +=10	MSIX_TAB_ADDRL	MSI-X Pending Bit Array Upper Register
0x2C004 +=10	MSIX_TAB_ADDRU{0..69}	MSI-X Table Entry Message Upper Address Register {0..69}
0x2C008 +=10	MSIX_TAB_DATA{0..69}	MSI-X Table Entry Message Data Register {0..69}
0x2C00C +=10	MSIX_TAB_MSK{0..69}	MSI-X Table Entry Mask Register {0..69}

19.3 PC2SR Registers

19.3.1 Outbound Window Lower Base Register {0..7}

This register defines the lower 17 bits of the base address of an outbound window. Window Y (Y = 0–7) is assigned to register Y.

Register name: OBWINLB{0..7} Reset value: 0x0000_0000	Register offset: 0x40000 += 20

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

Bits	Name	Description	Type	Reset value
31:15	ADD	Bits 31:15 of this windows base address (PCIe address with bit 31 as MSB).	R/WS	0x0
14:1	RESERVED	RESERVED	R	0x0
0	WIN_EN	1 = This window is enabled	R/WS	0x0

19.3.2 Outbound Window Upper Base Register {0..7}

This register defines the upper 32 bits of the base address of an outbound window. Window Y (Y = 0–7) is assigned to register Y.

Register name: OBWINUB{0..7} Reset value: 0x0000_0000	Register offset: 0x40004 += 20

Bits	7	6	5	4	3	2	1	0
31:24	ADD[63:56]							
23:16	ADD[55:48]							
15:8	ADD[47:40]							
7:0	ADD[39:32]							

Bits	Name	Description	Type	Reset value
31:0	ADD	32 MSB bits of outbound window base address (PCIe address with bit 63 as MSB)	R/WS	0x0

19.3.3 Outbound Window Size Register {0..7}

This register defines the size of an outbound window. Window Y (Y = 0–7) is assigned to register Y.

Register name: OBWINSZ{0..7} Reset value: 0x0000_0000	Register offset: 0x40008 += 20

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

Bits	Name	Description	Type	Reset value
31:13	RESERVED	RESERVED	R	0x0
12:8	SIZE	Defines the window size to be 2^(15+SIZE) bytes (including 3 zone bits). Values from 0 to 19 are valid. Other values are reserved. The minimum window size is 32 KB and the maximum window size is 16 GB.	R/WS	0x0
7:0	RESERVED	RESERVED	R	0x0

19.3.4 Outbound Window Lookup Table Zone Select Register

This register selects a zone for setting up its lookup table. To write a lookup table, this register should be configured after Outbound Window Lookup Table Data 0 Register, Outbound Window Lookup Table Data 1 Register, and Outbound Window Lookup Table Data 2 Register have been configured. To read a lookup table, this register should be configured before reading Outbound Window Lookup Table Data 0 Register, Outbound Window Lookup Table Data 1 Register, and Outbound Window Lookup Table Data 2 Register.

Upon power-up, all lookup table locations must be initialized to some value by software prior to reading. Reading from uninitialized lookup table locations may result in uncorrectable ECC errors.

Register name: ZONE_SEL Reset value: 0x0000_0000	Register offset: 0x41300

Bits	7	6	5	4	3	2	1	0
31:24	RESERVED							
23:16	RESERVED						RD_WRB	ZONE_GO
15:8	RESERVED							
7:0	RESERVED		WIN_SEL[2:0]			ZONE_SEL[2:0]		

Bits	Name	Description	Type	Reset value
31:18	RESERVED	RESERVED	R	0x0
17	RD_WRB	Outbound window lookup table read write select 0 = Write 1 = Read	R/WS	0x0

(Continued)

Bits	Name	Description	Type	Reset value
16	ZONE_GO	Zone configuration go bit When this bit toggles from 0 to 1, PC2SR performs an address lookup table access for a zone selected by field win_sel[2:0] and zone_sel[2:0] using Outbound Window Lookup Table Data 0 Register, Outbound Window Lookup Table Data 1 Register, and Outbound Window Lookup Table Data 2 Register. Tsi721 automatically clears this bit after finishing the table access. To write a lookup table entry, software must poll this bit to be 0 before configuring Outbound Window Lookup Table Data 0 Register, Outbound Window Lookup Table Data 1 Register, and Outbound Window Lookup Table Data 2 Register. Then software should configure this register. To read a lookup table entry, software must poll this bit to be 0 before configuring this register. Then, software must poll this bit to be 0 before reading Outbound Window Lookup Table Data 0 Register, Outbound Window Lookup Table Data 1 Register, and Outbound Window Lookup Table Data 2 Register	R/W	0x0
15:6	RESERVED	RESERVED	R	0x0
5:3	WIN_SEL	Window select Select a window to configure address translation table.	R/WS	0x0
2:0	ZONE_SEL	Zone select Select a zone to configure address translation table. It selects a zone within the selected outbound window by win_SEL[2:0].	R/WS	0x0

19.3.5 Outbound Window Lookup Table Data 0 Register

This register defines the translated S-RIO type and lower address for a selected zone.

Upon power-up, all lookup table locations must be initialized to some value by software prior to reading. Reading from uninitialized lookup table locations may result in uncorrectable ECC errors.

Register name: LUT_DATA0 Reset value: 0x0000_0000	Register offset: 0x41304

Bits	7	6	5	4	3	2	1	0
31:24	ADD[31:24]							
23:16	ADD[23:16]/ CONFIG_OFFSET[23:16]							
15:8	ADD[15:12]/ CONFIG_OFFSET[15:12]				RD_TYPE[3:0]			
7:0	RESERVED		RD_CRF	WR_CRF	WR_TYPE[3:0]			

Bits	Name	Description	Type	Reset value
31:12	ADD/ CONFIG_OFFSET	If translated into non-maintenance packets, these are translated S-RIO address bits with ADD[65] as MSB. LSB bits may be unused during address translation. Together with other ADD bits of lut_data0/1/2 registers, ADD[65:12] maps to S-RIO header field as displayed below: • ADD[31:12] maps to S-RIO header address bits 0:19, with ADD[31] to address bit 0 • In 34-bit addressing mode, ADD[33:32] maps to S-RIO header field xamsb[0:1] (ADD[33] to xamsb[0]) • In 50-bit addressing mode, ADD[47:32] maps to S-RIO header extended address bits 0:15 • In 50-bit addressing mode, ADD[49:48] maps to S-RIO header field xamsb[0:1] (ADD[49] to xamsb[0]) • In 66-bit addressing mode, ADD[63:32] maps to S-RIO header extended address bits 0:31 • In 66-bit addressing mode, ADD[65:64] maps to S-RIO header field xamsb[0:1] (ADD[65] to xamsb[0]) If translated into maintenance packets, these are translated S-RIO config_offset bits [23:12] with ADD[23] as MSB (bits 0:8 in S-RIO bit numbering). LSB bits can be unused during address translation. If the outbound window size is 27 bits or larger, however, this field is unused.	R/WS	0x0

(Continued)

Bits	Name	Description	Type	Reset value
11:8	RD_TYPE	It encodes the translated S-RIO packet type from a PCIe MRd for this zone: • 0b0001 = NREAD • 0b0010 = Maintenance read • Others = Reserved A MRd is translated into a NREAD or maintenance read with fixed S-RIO prio of 0. However, translated S-RIO CRF is configurable per zone.	R/WS	0x0
7:6	RESERVED	RESERVED	R	0x0
5	RD_CRF	It encodes the S-RIO CRF bit of a translated S-RIO packet type from a PCIe MRd hitting this zone.	R/WS	0x0
4	WR_CRF	It encodes the S-RIO CRF bit of a translated S-RIO packet type from a PCIe MWr hitting this zone.	R/WS	0x0
3:0	WR_TYPE	It encodes the translated S-RIO packet type from a PCIe MWr for this zone: • 0b0001 = NWRITE or SWRITE • 0b0010 = Maintenance write • 0b0100 = NWRITE_R • Others = Reserved A MWr is translated into fixed S-RIO prio of 2; however, the translated S-RIO CRF is configurable per zone.	R/WS	0x0

19.3.6 Outbound Window Lookup Table Data 1 Register

This register defines the translated upper address for a selected zone.

⚠ Upon power-up, all lookup table locations must be initialized to some value by software prior to reading. Reading from uninitialized lookup table locations may result in uncorrectable ECC errors.

Register name: LUT_DATA1 Reset value: 0x0000_0000	Register offset: 0x41308

Bits	7	6	5	4	3	2	1	0
31:24	ADD[63:56]							
23:16	ADD[55:48]							
15:8	ADD[47:40]							
7:0	ADD[39:32]							

Bits	Name	Description	Type	Reset value
31:0	ADD	Translated S-RIO address bits 63:32 with bit 63 as MSB. LSB bit may be unused during address translation. For more information about the ADD field, see similar field in Outbound Window Lookup Table Data 1 Register.	R/WS	0x0

19.3.7 Outbound Window Lookup Table Data 2 Register

This register defines the translated deviceID for the selected zone.

Upon power-up, all lookup table locations must be initialized to some value by software prior to reading. Reading from uninitialized lookup table locations may result in uncorrectable ECC errors.

Register name: LUT_DATA2 Reset value: 0x0000_0000	Register offset: 0x4130C

Bits	7	6	5	4	3	2	1	0
31:24	HOP_CNT[7:0]							
23:16	RESERVED		RESERVED		ADD[65:64]		TT[1:0]	
15:8	DEVICEID[15:8]							
7:0	DEVICEID[7:0]							

Bits	Name	Description	Type	Reset value
31:24	HOP_CNT	S-RIO hop count if this zone is configured to translate PCIe MWr/MRd into S-RIO maintenance write/ read	R/WS	0x0
23:20	RESERVED	RESERVED	R	0x0
19:18	ADD	translated S-RIO address bits 65:64 in little endian	R/WS	0x0
17:16	TT	translated S-RIO TT • 0b00 = 8-bit S-RIO deviceID is used • 0b01 = 16-bit S-RIO deviceID is used • others: reserved	R/WS	0x0
15:0	DEVICEID	translated S-RIO destination deviceID with bit 15 as MSB	R/WS	0x0

19.3.8 PC2SR Interrupt Enable CSR

This register defines the PC2SR interrupt enable function.

Register name: PC2SR_INTE Reset value: 0x0000_0000	Register offset: 0x41310

Bits	7	6	5	4	3	2	1	0
31:24	RESERVED							
23:16	RESERVED				RESERVED			
15:8	RESERVED							
7:0	RESERVED					ECC_UNCORR_EN	ECC_CORR_EN	RESERVED

Bits	Name	Description	Type	Reset value
31:3	RESERVED	RESERVED	R	0x0
2	ECC_UNCORR_EN	Uncorrectable ECC Error interrupt enable 1 = Enable ECC_UNCORR in PC2SR Interrupt CSR.	R/WS	0x0
1	ECC_CORR_EN	Correctable ECC Error interrupt enable 1 = Enable ECC_CORR in PC2SR Interrupt CSR.	R/WS	0x0
0	RESERVED	RESERVED	R	0x0

19.3.9 PC2SR Interrupt CSR

This register defines PC2SR interrupts.

Register name: PC2SR_INT Reset value: 0x0000_0000	Register offset: 0x41314

Bits	7	6	5	4	3	2	1	0
31:24	RESERVED							
23:16	RESERVED				RESERVED			
15:8	RESERVED				RESERVED			
7:0	RESERVED					ECC_UNCORR	ECC_CORR	RESERVED

Bits	Name	Description	Type	Reset value
31:3	RESERVED	RESERVED	R	0x0
2	ECC_UNCORR	Uncorrectable ECC interrupt 1 = An uncorrectable ECC error occurred. It is enabled by ECC_UNCORR_EN in PC2SR Interrupt Enable CSR.	R/W1CS	0x0
1	ECC_CORR	Correctable ECC interrupt 1 = A correctable ECC error occurred. It is enabled by ECC_CORR_EN in PC2SR Interrupt Enable CSR.	R/W1CS	0x0
0	RESERVED	RESERVED.	R	0x0

19.3.10 PC2SR Interrupt Set Register

This register defines the PC2SR interrupt set function for software debug.

Register name: PC2SR_INTSET Reset value: 0x0000_0000	Register offset: 0x41318

Bits	7	6	5	4	3	2	1	0
31:24	RESERVED							
23:16	RESERVED				RESERVED			
15:8	RESERVED							
7:0	RESERVED					ECC_UNCORR_SET	ECC_CORR_SET	RESERVED

Bits	Name	Description	Type	Reset value
31:3	RESERVED	RESERVED	R	0x0
2	ECC_UNCORR_SET	Uncorrectable ECC Error interrupt set When ECC_UNCORR_SET is set to 1, it forces uncorrectable ECC error interrupt ECC_UNCORR of PC2SR Interrupt CSR to 1.	R/W1S	0x0
1	ECC_CORR_SET	Correctable ECC Error interrupt set When ECC_CORR_SET is set to 1, it forces correctable ECC error interrupt ECC_CORR of PC2SR Interrupt CSR to 1.	R/W1S	0x0
0	RESERVED	RESERVED	R	0x0

19.3.11 PC2SR ECC Log Register

This register defines the PC2SR ECC log.

Register name: PC2SR_ECC_LOG Reset value: 0x0000_0000	Register offset: 0x4131C

Bits	7	6	5	4	3	2	1	0
31:24	RESERVED		ECC_UNCORR_MEM[13:8]					
23:16	ECC_UNCORR_MEM[7:0]							
15:8	RESERVED		ECC_CORR_MEM[13:8]					
7:0	ECC_CORR_MEM[7:0]							

Bits	Name	Description	Type	Reset value
31:30	RESERVED	RESERVED	R	0x0
29:16	ECC_UNCORR_MEM	When bit x is set to 1, an uncorrectable ECC error has occurred to SRAM instance x since it was cleared last time.	R/W1CS	0x0
15:14	RESERVED	RESERVED	R	0x0
13:0	ECC_CORR_MEM	When bit x set to 1, a correctable ECC error has occurred to SRAM instance x since it was cleared last time.	R/W1CS	0x0

19.3.12 Received Completion Count for Messaging Engine Register

This register defines the number of received PCIe completions for the Messaging Engine.

Register name: CPL_SMSG_CNT Reset value: 0x0000_0000	Register offset: 0x41404

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

Bits	Name	Description	Type	Reset value
31:0	CPL_SMSG_CNT	Received PCIe completion count for Messaging Engine It is a saturating counter that counts the number of received PCIe completions for both inbound and outbound Messaging Engine.	RC	0x0

19.3.13 Received Completion Count for Block DMA Engine Register

This register defines the number of received PCIe completions for the Block DMA Engine.

Register name: CPL_BDMA_CNT Reset value: 0x0000_0000	Register offset: 0x41408

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

Bits	Name	Description	Type	Reset value
31:0	CPL_BDMA_CNT	Received PCIe completion count for Block DMA Engine It is a saturating counter that counts the number of received PCIe completions for Block DMA Engine.	RC	0x0

19.3.14 Received Bridging TLP Count Register

This register defines the number of received bridging (without going through BDMA/SMSG) PCIe TLPs.

Register name: RXTLP_BRG_CNT Reset value: 0x0000_0000	Register offset: 0x4140C

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

Bits	Name	Description	Type	Reset value
31:0	RXTLP_BRG_CNT	Received bridging PCIe TLP count It is a saturating counter that counts the number of received bridging PCIe TLPs (including those MWr/ MRd for register access or MWr bridges to doorbells). The following bridging PCIe TLPs are not counted: • TLPs with unsupported request errors, that is, with type/fmt other than MWr/ MRd/ Cpl/ CplD • TLPs with unexpected completion errors, that is, Cpl/ CplD with unexpected tag values • All zero-length MWr	RC	0x0

19.3.15 Sent Packet Count of Messaging Engine Register

This register defines the number of S-RIO message segments or responses sent by the Messaging Engine.

Register name: TXPKT_SMSG_CNT Reset value: 0x0000_0000	Register offset: 0x41410

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

Bits	Name	Description	Type	Reset value
31:0	TXPKT_SMSG_CNT	Sent packet count by Messaging Engine It is a saturating counter that counts the number of S-RIO message segments or responses sent by the Messaging Engine.	RC	0x0

19.3.16 Sent Packet Count of Block DMA Engine Register

This register defines the number of S-RIO packets sent by the Block DMA Engine.

Register name: TXPKT_BDMA_CNT Reset value: 0x0000_0000	Register offset: 0x41414

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

Bits	Name	Description	Type	Reset value
31:0	TXPKT_BDMA_CNT	Sent packet count by Block DMA Engine It is a saturating counter that counts the number of S-RIO packets sent by the Block DMA Engine.	RC	0x0

19.3.17 Sent Bridging Packet Count Register

This register defines the number of bridging (not from SMSG/BDMA) S-RIO packets sent by the Mapping Engine.

Register name: TXPKT_BRG_CNT Reset value: 0x0000_0000	Register offset: 0x41418

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

Bits	Name	Description	Type	Reset value
31:0	TXPKT_BRG_CNT	Sent bridging packet count It is a saturating counter that counts the number of bridging S-RIO packets sent by Mapping Engine.	RC	0x0

19.3.18 Received Bridging TLP Error Count Register

This register defines a saturating counter for received bridging TLPs with a specific type of error.

Register name: BRG_TLP_ERR_CNT Reset value: 0x0000_0000	Register offset: 0x4141C

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

Bits	Name	Description	Type	Reset value
31:0	BRG_TLP_ERR_CNT	Bridging TLP Error Count It is a saturating counter that counts the number of received bridging TLPs with error.	RC	0x0

19.4 SR2PC Registers

19.4.1 Inbound Doorbell Queue Control Register {0..7}

This register defines the inbound doorbell queue control function. Inbound doorbell queue Y (Y = 0–7) is assigned to register Y.

Register name: IDQ_CTL{0..7} Reset value: 0x0000_0000	Register offset: 0x20000 += 1000

Bits	7	6	5	4	3	2	1	0
31:24	RESERVED							
23:16	RESERVED							
15:8	RESERVED							
7:0	RESERVED						SUSPEND	INIT

Bits	Name	Description	Type	Reset value
31:2	RESERVED	RESERVED	R	0x0
1	SUSPEND	Inbound doorbell queue suspend (tear down). 1 = Tsi721 suspends inbound doorbell processing for this Inbound doorbell queue. The Tsi721 automatically clears this bit to 0 after it has finished the suspend procedure (SUSPENDED bit of the SR2PC Channel Interrupt Register {0..7} is set to 1). Note: Software must not set this bit to 1 while the INIT bit is 1; otherwise, device behavior will be undefined.	R/W	0x0
0	INIT	Inbound doorbell queue initialization. 1 = Tsi721 starts inbound doorbell processing for this inbound doorbell queue and resets the Inbound Doorbell Queue Read Pointer Register {0..7}, and Inbound Doorbell Queue Write Pointer Register {0..7}. Software can set INIT to 1 only when this queue is idle; that is, Inbound Doorbell Queue Status Register {0..7} RUN bit is 0. Tsi721 clears this bit when software writes to the Inbound Doorbell Queue Read Pointer Register {0..7}.	R/W	0x0

19.4.2 Inbound Doorbell Queue Status Register {0..7}

This register defines the inbound doorbell queue status function. Inbound doorbell queue channel Y (Y = 0–7) is assigned to register Y.

Register name: IDQ_STS{0..7} Reset value: 0x0000_0000	Register offset: 0x20004 += 1000

Bits	7	6	5	4	3	2	1	0
31:24	RESERVED							
23:16	RESERVED		RUN	RESERVED				
15:8	RESERVED							
7:0	RESERVED							

Bits	Name	Description	Type	Reset value
31:22	RESERVED	RESERVED	R	0x0
21	RUN	0 = The inbound doorbell queue is uninitialized or Tsi721 has suspended the queue after software sets the SUSPEND bit in Inbound Doorbell Queue Control Register {0..7}. 1 = Otherwise During inbound doorbell queue initialization, the RUN bit is set to 1 when software writes to Inbound Doorbell Queue Read Pointer Register {0..7}.	R	0x0
20:0	RESERVED	RESERVED	R	0x0

19.4.3 Inbound Doorbell Classification Register {0..7}

These registers define how inbound doorbells are classified. Register Y (Y = 0–7) is assigned to doorbell queue Y. The search order is MASK/PATTERN 0 ->7.

Register name: IDQ_MASK{0..7} Reset value: 0x0000_0000	Register offset: 0x20008 += 1000

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

Bits	Name	Description	Type	Reset value
31:16	MASK	MASK. This defines masks for doorbell classification.	R/WS	0x0
15:0	PATTERN	Defines masks for doorbell classification. Doorbell Info bits are ANDed with MASK first, then compared with PATTERN. If equal, the doorbell belongs to the doorbell queue assigned to this register.	R/WS	0x0

19.4.4 Inbound Doorbell Queue Read Pointer Register {0..7}

This register defines the read pointer of an inbound doorbell queue. Inbound doorbell queue Y (Y = 0–7) is assigned to register Y.

Register name: IDQ_RP{0..7} Reset value: 0x0000_0000	Register offset: 0x2000C += 1000

Bits	7	6	5	4	3	2	1	0
31:24	RESERVED							
23:16	RESERVED					RD_PTR[18:16]		
15:8	RD_PTR[15:8]							
7:0	RD_PTR[7:0]							

Bits	Name	Description	Type	Reset value
31:19	RESERVED	RESERVED	R	0x0
18:0	RD_PTR	Read pointer to an entry of inbound doorbell queue relative to doorbell queue base address. Software is responsible for updating this register. When a doorbell is received for this queue, the Tsi721 issues a doorbell MWr to the doorbell queue when RD_PTR is not equal to the WR_PTR. The Tsi721 queue is declared full when (WR_PTR + 1) MOD Inbound Doorbell Queue Size Register {0..7} = RD_PTR, and leaves one queue entry unused when the queue is full.	R/W	0x0

19.4.5 Inbound Doorbell Queue Write Pointer Register {0..7}

This register defines the write pointer of an inbound doorbell queue. Inbound doorbell queue Y (Y = 0–7) is assigned to register Y.

Register name: IDQ_WP{0..7} Reset value: 0x0000_0000	Register offset: 0x20010 += 1000

Bits	7	6	5	4	3	2	1	0
31:24	RESERVED							
23:16	RESERVED					WR_PTR[18:16]		
15:8	WR_PTR[15:8]							
7:0	WR_PTR[7:0]							

Bits	Name	Description	Type	Reset value
31:19	RESERVED	RESERVED	R	0x0
18:0	WR_PTR	Write pointer to an entry of an inbound doorbell queue relative to doorbell queue base address. The Tsi721 updates WR_PTR after it issues a doorbell MWr.	R/W	0x0

19.4.6 Inbound Doorbell Queue Lower Base Register {0..7}

This register defines the lower 26 bits of the base address of an inbound doorbell queue. Inbound doorbell queue Y (Y = 0–7) is assigned to register Y.

Register name: IDQ_BASEL{0..7} Reset value: 0x0000_0000	Register offset: 0x20014 += 1000

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

Bits	Name	Description	Type	Reset value
31:6	ADD	26 LSB bits of Inbound doorbell queue base address. The base address must be 64-byte aligned.	R/WS	0x0
5:0	Reserved	RESERVED	R	0x0

19.4.7 Inbound Doorbell Queue Upper Base Register {0..7}

This register defines the upper 32 bits of the base address of an inbound doorbell queue. Inbound doorbell queue Y (Y = 0–7) is assigned to register Y.

Register name: IDQ_BASEU{0..7} Reset value: 0x0000_0000	Register offset: 0x20018 += 1000

Bits	7	6	5	4	3	2	1	0
31:24	ADD[63:56]							
23:16	ADD[55:48]							
15:8	ADD[47:40]							
7:0	ADD[39:32]							

Bits	Name	Description	Type	Reset value
31:0	ADD	32 MSB bits of inbound doorbell queue base address.	R/WS	0x0

19.4.8 Inbound Doorbell Queue Size Register {0..7}

This register defines the size of an inbound doorbell queue. Inbound doorbell queue Y (Y = 0–7) is assigned to register Y.

Register name: IDQ_SIZE{0..7} Reset value: 0x0000_0005	Register offset: 0x2001C += 1000

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

Bits	Name	Description	Type	Reset value
31:4	RESERVED	RESERVED	R	0x0
3:0	SIZE	SIZE: • 0b0000 = Reserved • 0b0101 = 512 entries • 0b0110 = 1K entries • 0b0111 = 2K entries • 0b1000 = 4K entries • 0b1001 = 8K entries • 0b1010 = 16K entries • 0b1011 = 32K entries • 0b1100 = 64K entries • 0b1101 = 128K entries • 0b1110 = 256K entries • 0b1111 = 512K entries • Others = Reserved	R/WS	0x5

19.4.9 SR2PC Channel Interrupt Register {0..7}

This register defines SR2PC channel interrupts. Channel Y (Y = 0–7) is assigned to register Y.

Register name: SR_CH{0..7}INT Reset value: 0x0000_0000	Register offset: 0x20040 += 1000

Bits	7	6	5	4	3	2	1	0
31:24	RESERVED							
23:16	RESERVED							
15:8	RESERVED							
7:0	RESERVED		ODB_OK	IDBQ_RCV	SUSPENDED	ODB_TO	ODB_RETRY	ODB_ERR

Bits	Name	Description	Type	Reset value
31:6	RESERVED	RESERVED	R	0x0
5	ODB_OK	1 = This outbound doorbell channel has received an OK response since this bit was last cleared.	R/W1CS	0x0
4	IDBQ_RCV	1 = This inbound doorbell queue sent a new inbound doorbell since this bit was last cleared.	R/W1CS	0x0
3	SUSPENDED	1 = Tsi721 suspended at least once the processing of inbound doorbells for this inbound doorbell queue triggered by SUSPEND bit of Inbound Doorbell Queue Control Register {0..7} since this bit was last cleared.	R/W1CS	0x0
2	ODB_TO	1 = This outbound doorbell channel has a response timeout since this bit was last cleared.	R/W1CS	0x0
1	ODB_RETRY	1 = This outbound doorbell channel received a RETRY response since this bit was last cleared.	R/W1CS	0x0
0	ODB_ERR	1 = This outbound doorbell channel received an ERROR response since this bit was last cleared.	R/W1CS	0x0

19.4.10 SR2PC Channel Interrupt Enable Register {0..7}

This register defines the SR2PC channel interrupt enable. Channel Y (Y = 0–7) is assigned to register Y.

Register name: SR_CH{0..7}INTE Reset value: 0x0000_0010	Register offset: 0x20044 += 1000

Bits	7	6	5	4	3	2	1	0
31:24	RESERVED							
23:16	RESERVED							
15:8	RESERVED							
7:0	RESERVED		ODB_OK_EN	IDBQ_RCV_EN	SUSPENDED_EN	ODB_TO_EN	ODB_RETRY_EN	ODB_ERR_EN

Bits	Name	Description	Type	Reset value
31:6	RESERVED	RESERVED	R	0x0
5	ODB_OK_EN	1 = Enable interrupt generation for ODB_OK bit of SR2PC Channel Interrupt Register {0..7}.	R/WS	0x0
4	IDBQ_RCV_EN	1 = Enable interrupt generation for IDBQ_RCV bit of SR2PC Channel Interrupt Register {0..7}. When MSI-X is used, this bit must be set to 1.	R/WS	0x1
3	SUSPENDED_EN	1 = Enable interrupt generation for SUSPENDED bit of SR2PC Channel Interrupt Register {0..7}.	R/WS	0x0
2	ODB_TO_EN	1 = Enable interrupt generation for ODB_TO bit of SR2PC Channel Interrupt Register {0..7}.	R/WS	0x0
1	ODB_RETRY_EN	1 = Enable interrupt generation for ODB_RETRY bit of SR2PC Channel Interrupt Register {0..7}.	R/WS	0x0
0	ODB_ERR_EN	1 = Enable interrupt generation for ODB_ERR bit of SR2PC Channel Interrupt Register {0..7}.	R/WS	0x0

19.4.11 SR2PC Channel Interrupt Set Register {0..7}

This register defines SR2PC channel interrupt set. Channel Y (Y = 0–7) is assigned to register Y.

Register name: SR_CH{0..7}INTSET Reset value: 0x0000_0000	Register offset: 0x20048 += 1000

Bits	7	6	5	4	3	2	1	0
31:24	RESERVED							
23:16	RESERVED							
15:8	RESERVED							
7:0	RESERVED		ODB_OK_SET	IDBQ_RCV_SET	SUSPENDED_SET	ODB_TO_SET	ODB_RETRY_SET	ODB_ERR_SET

Bits	Name	Description	Type	Reset value
31:6	RESERVED	RESERVED	R	0x0
5	ODB_OK_SET	1 = Set ODB_OK bit in SR2PC Channel Interrupt Register {0..7} to 1.	R/W1S	0x0
4	IDBQ_RCV_SET	1 = Set IDBQ_RCV bit in SR2PC Channel Interrupt Register {0..7} to 1.	R/W1S	0x0
3	SUSPENDED_SET	1 = Set SUSPENDED bit of SR2PC Channel Interrupt Register {0..7} to 1.	R/W1S	0x0
2	ODB_TO_SET	1 = Set ODB_TO bit of SR2PC Channel Interrupt Register {0..7} to 1.	R/W1S	0x0
1	ODB_RETRY_SET	1 = Set ODB_RETRY bit of SR2PC Channel Interrupt Register {0..7} to 1.	R/W1S	0x0
0	ODB_ERR_SET	1 = Set ODB_ERR bit of SR2PC Channel Interrupt Register {0..7} to 1.	R/W1S	0x0

19.4.12 Outbound Doorbell Count {0..7} Register

Outbound doorbell channel Y is assigned to register Y.

Register name: ODB_CNT{0..7} Reset value: 0x0000_0000	Register offset: 0x20100 += 1000

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

Bits	Name	Description	Type	Reset value
31:16	ODB_TOT_CNT	Counts the total number of doorbells sent to the S-RIO network, including those with OK/RETRY/ERROR responses and those that encountered a timeout. It is a saturating counter.	RC	0x0
15:0	ODB_OK_CNT	Counts the number of doorbells with OK response. It is a saturating counter.	RC	0x0

19.4.13 Outbound Doorbell Response Log Buffer Data 0 Register {0..7}

This register is the PCIe header 1st dword for log buffer of an outbound doorbell channel. Outbound doorbell channel Y is assigned to register Y.

Register name: ODB_LOG_DAT0{0..7} Reset value: 0x0000_0000	Register offset: 0x20104 += 1000

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

Bits	Name	Description	Type	Reset value
31:0	DATA	PCIe header 1st dword of the outbound doorbell response log buffer, with PCIe byte 0 mapped to DATA[7:0]. Valid when ODB_TO or ODB_RETRY or ODB_ERR of SR2PC Channel Interrupt Register {0..7} is 1.	R/WS	0x0

19.4.14 Outbound Doorbell Response Log Buffer Data 1 Register {0..7}

This register is the PCIe header second dword for log buffer of an outbound doorbell channel. Outbound doorbell channel X is assigned to register X.

Register name: ODB_LOG_DAT1{0..7} Reset value: 0x0000_0000	Register offset: 0x20108 += 1000

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

Bits	Name	Description	Type	Reset value
31:0	DATA	PCIe header 2nd dword of the outbound doorbell response log buffer. Valid when ODB_TO or ODB_RETRY or ODB_ERR of SR2PC Channel Interrupt Register {0..7} is 1.	R/WS	0x0

19.4.15 Outbound Doorbell Response Log Buffer Data 2 Register {0..7}

This register is the PCIe header 3rd dword for log buffer of an outbound doorbell channel. Outbound doorbell channel Y is assigned to register Y.

Register name: ODB_LOG_DAT2{0..7} Reset value: 0x0000_0000	Register offset: 0x2010C += 1000

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

Bits	Name	Description	Type	Reset value
31:0	DATA	PCIe header 3rd dword of the outbound doorbell response log buffer. Valid when ODB_TO or ODB_RETRY or ODB_ERR of SR2PC Channel Interrupt Register {0..7} is 1.	R/WS	0x0

19.4.16 Outbound Doorbell Response Log Buffer Data 3 Register {0..7}

This register is the PCIe header 4th dword for log buffer of an outbound doorbell channel. Outbound doorbell channel Y is assigned to register Y.

Register name: ODB_LOG_DAT3{0..7} Reset value: 0x0000_0000	Register offset: 0x20120 += 1000

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

Bits	Name	Description	Type	Reset value
31:0	DATA	PCIe header 4th dword of the outbound doorbell response log buffer. Valid when ODB_TO or ODB_RETRY or ODB_ERR of SR2PC Channel Interrupt Register {0..7} is 1.	R/WS	0x0

19.4.17 SR2PC Outbound Doorbell Response Log Buffer Status Register {0..7}

This register defines the SR2PC outbound doorbell response log buffer status. Outbound doorbell channel Y (Y = 0–7) is assigned to register Y.

Register name: ODB{0..7}LOGSTS Reset value: 0x0000_0000	Register offset: 0x20124 += 1000

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

Bits	Name	Description	Type	Reset value
31:2	RESERVED	RESERVED	R	0x0
1:0	LOG_BUF_ERR	Valid when ODB_TO, ODB_RETRY, or ODB_ERR of SR2PC Channel Interrupt Register {0..7} is set to 1. It encodes the error type with which the PCIe header in the log buffer for this outbound doorbell channel is associated: • 0b00 = For a doorbell ERROR response • 0b01 = For a doorbell RETRY response • 0b10 = For a doorbell response timeout • Others = Reserved	R	0x0

19.4.18 Inbound Window Lower Base Register{0..7}

This register defines the lower 32 bits of the base address of an inbound window. Window Y (Y = 0–7) is assigned to register Y.

Register name: IBWIN_LB{0..7} Reset value: 0x0000_0000	Register offset: 0x29000 += 20

Bits	7	6	5	4	3	2	1	0
31:24	ADD[31:24]							
23:16	ADD[23:16]							
15:8	ADD[15:12]				RESERVED			
7:0	RESERVED							WIN_EN

Bits	Name	Description	Type	Reset value
31:12	ADD	Address Bits 31:12 of this window's base address (S-RIO address) is in little endian. Together with other ADD bits of IBWIN_LB/IBWIN_UB/IBWIN_SZ registers, ADD[65:12] maps to S-RIO header field as displayed below: • ADD[31:12] maps to S-RIO header address bits 0:19, with ADD[31] to address bit 0, • In 34-bit addressing mode, ADD[33:32] maps to S-RIO header field xamsb[0:1] (ADD[33] to xamsb[0]) • In 50-bit addressing mode, ADD[47:32] maps to S-RIO header extended address bits 0:15 • In 50-bit addressing mode, ADD[49:48] maps to S-RIO header field xamsb[0:1] (ADD[49] to xamsb[0]) • In 66-bit addressing mode, ADD[63:32] maps to S-RIO header extended address bits 0:31 • In 66-bit addressing mode, ADD[65:64] maps to S-RIO header field xamsb[0:1] (ADD[65] to xamsb[0])	R/WS	0x0
11:1	RESERVED	RESERVED	R	0x0
0	WIN_EN	1 = Enable the window	R/WS	0x0

19.4.19 Inbound Window Upper Base Register{0..7}

This register defines upper 32 bits of the base address of an inbound window. Window Y (Y = 0–7) is assigned to register Y.

Register name: IBWIN_UB{0..7} Reset value: 0x0000_0000	Register offset: 0x29004 += 20

Bits	7	6	5	4	3	2	1	0
31:24	ADD[63:56]							
23:16	ADD[55:48]							
15:8	ADD[47:40]							
7:0	ADD[39:32]							

Bits	Name	Description	Type	Reset value
31:0	ADD	32 MSB bits of inbound window base address (S-RIO address) in little endian. For more information about ADD, see the similar field in the Inbound Window Lower Base Register{0..7}.	R/WS	0x0

19.4.20 Inbound Window Size Register{0..7}

This register defines the size of an inbound window. Window Y (Y = 0–7) is assigned to register Y.

Register name: IBWIN_SZ{0..7} Reset value: 0x0000_0000	Register offset: 0x29008 += 20

Bits	7	6	5	4	3	2	1	0
31:24	RESERVED						ADD[65:64]	
23:16	RESERVED							
15:8	RESERVED			SIZE[4:0]				
7:0	RESERVED							

Bits	Name	Description	Type	Reset value
31:26	RESERVED	RESERVED	R	0x0
25:24	ADD	Bits 65:64 of this window base address (S-RIO address) in little endian.	R/WS	0x0
23:13	RESERVED	RESERVED	R	0x0
12:8	SIZE	Defines the window size to be 2^(12+SIZE) bytes. Values from 0 to 22 are valid. Other values are reserved. The minimum window size is 4 KB and the maximum window size is 16 GB.	R/WS	0x0
7:0	RESERVED	RESERVED	R	0x0

19.4.21 Inbound Window Translated Lower Address Register {0..7}

This register defines translated address for an inbound window. Window Y (Y=0–7) maps to register Y.

Register name: IBWIN_TLA {0..7} Reset value: 0x0000_0000	Register offset: 0x2900C += 20

Bits	7	6	5	4	3	2	1	0
31:24	ADD[31:24]							
23:16	ADD[23:16]							
15:8	ADD[15:12]				Reserved			
7:0	Reserved							

Bits	Name	Description	Type	Reset value
31:12	ADD	Translated PCIe address bits 31:12. LSB bits may be unused during address translation.	R/WS	0x0
11:0	Reserved	Reserved	R	0x0

19.4.22 Inbound Window Translated Upper Address {0..7}

This register defines translated address for an inbound window. Window Y (Y=0–7) maps to register Y.

Register name: IBWIN_TUA{0..7} Reset value: 0x0000_0000	Register offset: 0x29010 += 20

Bits	7	6	5	4	3	2	1	0
31:24	ADD[63:56]							
23:16	ADD[55:48]							
15:8	ADD[47:40]							
7:0	ADD[39:32]							

Bits	Name	Description	Type	Reset value
31:0	ADD	Translated PCIe address bits 63:32. LSB bits may be unused during address translation.	R/WS	0x0

19.4.23 SR2PC General Interrupt Enable CSR

This register defines SR2PC general interrupt enable.

Register name: SR2PC_GEN_INTE Reset value: 0x0000_0000	Register offset: 0x29800

Bits	7	6	5	4	3	2	1	0
31:24	ERR_RSP_EN	Reserved			ILL_DEC_EN	ILL_TARGET_EN	Reserved	RSP_TO_EN
23:16	UNS_RSP_EN	Reserved						
15:8	RESERVED	Reserved				NW_RSP_OK_EN	MW_RSP_OK_EN	DL_DOWN_EN
7:0	RESERVED	ECC_UNCORR_EN	ECC_CORR_EN	DB_MISS_EN	NW_RSP_TO_EN	MW_RSP_TO_EN	NW_RSP_ERR_EN	MW_RSP_ERR_EN

Bits	Name	Description	Type	Reset value
31	ERR_RSP_EN	I/O error response interrupt enable	R/WS	0
30:28	RESERVED	RESERVED	R	0x0
27	ILL_DEC_EN	Illegal transaction decode interrupt enable	R/WS	0
26	ILL_TARGET_EN	Illegal transaction target interrupt enable	R/WS	0
25	Reserved	Reserved	R	0
24	RSP_TO_EN	Response packet timeout interrupt enable	R/WS	0
23	UNS_RSP_EN	Unsolicited/Unexpected response packet interrupt enable	R/WS	0
22:11	RESERVED	RESERVED	R	0x0
10	NW_RSP_OK_EN	SR2PC NWRITE_R OK response interrupt enable 1 = Enable NW_RSP_OK bit of SR2PC General Interrupt CSR.	R/WS	0x0
9	MW_RSP_OK_EN	SR2PC maintenance write OK response interrupt enable 1 = Enable MW_RSP_OK bit of SR2PC General Interrupt CSR	R/WS	0x0
8	DL_DOWN_EN	PCIe link down port-write 1 = Enable DL_DOWN bit of SR2PC General Interrupt CSR	R/WS	0x0
7	RESERVED	RESERVED	R	0x0

(Continued)

Bits	Name	Description	Type	Reset value
6	ECC_UNCORR_EN	Uncorrectable ECC Error interrupt enable 1 = Enable ECC_UNCORR bit of SR2PC General Interrupt CSR.	R/WS	0x0
5	ECC_CORR_EN	Correctable ECC Error interrupt enable 1 = Enable ECC_CORR bit of SR2PC General Interrupt CSR.	R/WS	0x0
4	DB_MISS_EN	SR2PC inbound doorbell classification miss interrupt enable 1 = Enable DB_MISS bit of SR2PC General Interrupt CSR	R/WS	0x0
3	NW_RSP_TO_EN	SR2PC NWRITE_R response timeout interrupt enable 1 = Enable NW_RSP_TO bit of SR2PC General Interrupt CSR	R/WS	0x0
2	MW_RSP_TO_EN	SR2PC maintenance write response timeout interrupt enable 1 = Enable MW_RSP_TO bit of SR2PC General Interrupt CSR	R/WS	0x0
1	NW_RSP_ERR_EN	SR2PC NWRITE_R ERROR response interrupt enable 1 = Enable NW_RSP_ERR bit of SR2PC General Interrupt CSR	R/WS	0x0
0	MW_RSP_ERR_EN	SR2PC Maintenance Write ERROR response interrupt enable 1 = Enable MW_RSP_ERR bit of SR2PC General Interrupt CSR	R/WS	0x0

19.4.24 SR2PC Port-Write Enable CSR

This register defines SR2PC port-write generation enable.

Register name: SR2PC_PWE Reset value: 0x0000_0000	Register offset: 0x29804

Bits	7	6	5	4	3	2	1	0
31:24	ERR_RSP_EN	Reserved			ILL_DEC_EN	ILL_TARGET_EN	Reserved	RSP_TO_EN
23:16	UNS_RSP_EN	Reserved						
15:8	RESERVED	Reserved						DL_DOWN_EN
7:0	Reserved							

Bits	Name	Description	Type	Reset value
31	ERR_RSP_EN	I/O error response port-write generation enable 1 = Enable port-write generation for ERR_RSP bit of SR2PC General Interrupt CSR	R/WS	0
30:28	RESERVED	RESERVED	R	0x0
27	ILL_DEC_EN	Illegal transaction decode port-write generation enable 1 = Enable port-write generation for ILL_DEC bit of SR2PC General Interrupt CSR	R/WS	0
26	ILL_TARGET_EN	Illegal transaction target port-write generation enable 1 = Enable port-write generation for ILL_TARGET bit of SR2PC General Interrupt CSR	R/WS	0
25	Reserved	Reserved	R	0
24	RSP_TO_EN	Response packet timeout port-write generation enable 1 = Enable port-write generation for RSP_TO bit of SR2PC General Interrupt CSR	R/WS	0
23	UNS_RSP_EN	Unsolicited/Unexpected response packet port-write generation enable. 1 = Enable port-write generation for UNS_RSP bit of SR2PC General Interrupt CSR	R/WS	0
22:9	RESERVED	RESERVED	R	0x0
8	DL_DOWN_EN	PCIe link down port-write enable 1 = Enable port-write generation for DL_DOWN bit of SR2PC General Interrupt CSR.	R/WS	0x0
7:0	RESERVED	RESERVED	R	0x0

19.4.25 SR2PC General Interrupt CSR

This register defines SR2PC general interrupts.

Register name: SR2PC_GEN_INT Reset value: 0x0000_0000	Register offset: 0x29808

Bits	7	6	5	4	3	2	1	0
31:24	ERR_RSP	Reserved			ILL_DEC	ILL_TARGET	Reserved	RSP_TO
23:16	UNS_RSP	Reserved						
15:8	RESERVED					MW_RSP_OK	NW_RSP_OK	DL_DOWN
7:0	RESERVED	ECC_UNCORR	ECC_CORR	DB_MISS	NW_RSP_TO	MW_RSP_TO	NW_RSP_ERR	MW_RSP_ERR

Bits	Name	Description	Type	Reset value
31	ERR_RSP	S-RIO I/O error response received	R/W1CS	0
30:28	RESERVED	RESERVED	R	0x0
27	ILL_DEC	S-RIO Illegal transaction decode	R/W1CS	0
26	ILL_TARGET	S-RIO illegal transaction target (when a received S-RIO I/O request miss all inbound windows)	R/W1CS	0
25	Reserved	Reserved	R	0
24	RSP_TO	S-RIO response packet timeout	R/W1CS	0
23	UNS_RSP	S-RIO Unsolicited/Unexpected response packet was received.	R/W1CS	0
22:11	Reserved	Reserved	R	0
10	MW_RSP_OK	SR2PC maintenance write OK response interrupt 1 = Maintenance write OK response has been received since this bit was cleared. It is enabled by MW_RSP_TO_EN bit of SR2PC General Interrupt Enable CSR	R/W1CS	0x0
9	NW_RSP_OK	SR2PC NWRITE_R OK response interrupt 1 = All NWRITE_R OK responses to an associated MWr have been received since this bit was cleared. It is enabled by NW_RSP_TO_EN bit of SR2PC General Interrupt Enable CSR	R/W1CS	0x0

(Continued)

Bits	Name	Description	Type	Reset value
8	DL_DOWN	PCIe link down interrupt 1 = PCIe link is down. It is enabled by DL_DOWN_EN of SR2PC General Interrupt Enable CSR.	R/W1CS	0x0
7	RESERVED	RESERVED.	R	0x0
6	ECC_UNCORR	Uncorrectable ECC interrupt 1 = An uncorrectable ECC error has occurred. It is enabled by ECC_UNCORR_EN of SR2PC General Interrupt Enable CSR.	R/W1CS	0x0
5	ECC_CORR	Correctable ECC interrupt 1 = A correctable ECC error has occurred. It is enabled by ECC_CORR_EN of SR2PC General Interrupt Enable CSR.	R/W1CS	0x0
4	DB_MISS	SR2PC inbound doorbell classification miss interrupt 1 = A received inbound doorbell has missed all Inbound Doorbell Classification Register {0..7} since this bit was last cleared. It is enabled by DB_MISS_EN bit of SR2PC General Interrupt Enable CSR.	R/W1CS	0x0
3	NW_RSP_TO	SR2PC NWRITE_R response timeout interrupt 1 = An NWRITE_R response timeout has occurred since this bit was last cleared. It is enabled by NW_RSP_TO_EN bit of SR2PC General Interrupt Enable CSR	R/W1CS	0x0
2	MW_RSP_TO	SR2PC maintenance write response timeout interrupt 1 = A maintenance write response timeout has occurred since this bit was last cleared. It is enabled by MW_RSP_TO_EN bit of SR2PC General Interrupt Enable CSR	R/W1CS	0x0
1	NW_RSP_ERR	SR2PC NWRITE_R ERROR response interrupt 1 = An NWRITE_R ERROR response has been received since this bit was last cleared. It is enabled by NW_RSP_ERR bit of SR2PC General Interrupt Enable CSR.	R/W1CS	0x0
0	MW_RSP_ERR	SR2PC Maintenance Write ERROR response interrupt 1 = A maintenance write ERROR response has been received since this bit was last cleared. It is enabled by MW_RSP_ERR_EN bit of SR2PC General Interrupt Enable CSR.	R/W1CS	0x0

19.4.26 SR2PC General Interrupt Set CSR

This register defines SR2PC general interrupt set.

Register name: SR2PC_GEN_INTSET Reset value: 0x0000_0000	Register offset: 0x2980C

Bits	7	6	5	4	3	2	1	0
31:24	ERR_RSP_SET	RESERVED			ILL_DEC_SET	ILL_TARGET_SET	Reserved	RSP_TO_SET
23:16	UNS_RSP_SET	RESERVED						
15:8	RESERVED					MW_RSP_OK_SET	NW_RSP_OK_SET	RESERVED
7:0	RESERVED	ECC_UNCORR_SET	ECC_CORR_SET	DB_MISS_SET	NW_RSP_TO_SET	MW_RSP_TO_SET	NW_RSP_ERR_SET	MW_RSP_ERR_SET

Bits	Name	Description	Type	Reset value
31	ERR_RSP_SET	1 = Set ERR_RSP bit of SR2PC General Interrupt CSR to 1.	R/W1S	0
30:28	RESERVED	RESERVED	R	0x0
27	ILL_DEC_SET	1 = Set ILL_DEC bit of SR2PC General Interrupt CSR to 1.	R/W1S	0
26	ILL_TARGET_SET	1 = Set ILL_TARGET bit of SR2PC General Interrupt CSR to 1.	R/W1S	0
25	Reserved	Reserved	R	0
24	RSP_TO_SET	1 = Set RSP_TO bit of SR2PC General Interrupt CSR to 1.	R/W1S	0
23	UNS_RSP_SET	1 = Set UNS_RSP bit of SR2PC General Interrupt CSR to 1.	R/W1S	0
22:11	RESERVED	RESERVED	R	0x0
10	MW_RSP_OK_SET	SR2PC maintenance write OK response interrupt set 1 = Set MW_RSP_OK bit of SR2PC General Interrupt CSR to 1.	R/W1S	0x0
9	NW_RSP_OK_SET	SR2PC NWRITE_R OK response interrupt set 1 = Set NW_RSP_OK bit of SR2PC General Interrupt CSR to 1.	R/W1S	0x0
8:7	RESERVED	RESERVED	R	0x0

(Continued)

Bits	Name	Description	Type	Reset value
6	ECC_UNCORR_SET	Uncorrectable ECC Error interrupt set 1 = Set ECC_UNCORR bit of SR2PC General Interrupt CSR to 1.	R/W1S	0x0
5	ECC_CORR_SET	Correctable ECC Error interrupt set 1 = Set ECC_CORR bit of SR2PC General Interrupt CSR to 1.	R/W1S	0x0
4	DB_MISS_SET	SR2PC inbound doorbell classification miss interrupt set 1 = Set DB_MISS bit of SR2PC General Interrupt CSR to 1.	R/W1S	0x0
3	NW_RSP_TO_SET	SR2PC NWRITE_R response timeout interrupt set 1 = Set NW_RSP_TO bit of SR2PC General Interrupt CSR to 1.	R/W1S	0x0
2	MW_RSP_TO_SET	SR2PC maintenance write response timeout interrupt set 1 = Set MW_RSP_TO bit of SR2PC General Interrupt CSR to 1.	R/W1S	0x0
1	NW_RSP_ERR_SET	SR2PC NWRITE_R ERROR response interrupt enable 1 = Set NW_RSP_ERR bit of SR2PC General Interrupt CSR to 1.	R/W1S	0x0
0	MW_RSP_ERR_SET	SR2PC Maintenance Write ERROR response interrupt set 1 = Set MW_RSP_ERR bit of SR2PC General Interrupt CSR to 1.	R/W1S	0x0

19.4.27 SR2PC Correctable ECC Log Register

This register defines SR2PC correctable ECC log.

Register name: SR2PC_CORR_ECC_LOG Reset value: 0x0000_0000	Register offset: 0x29810

Bits	7	6	5	4	3	2	1	0
31:24	RESERVED							
23:16	RESERVED					ECC_CORR_MEM[18:16]		
15:8	ECC_CORR_MEM[15:8]							
7:0	ECC_CORR_MEM[7:0]							

Bits	Name	Description	Type	Reset value
31:19	RESERVED	RESERVED	R	0x0
18:0	ECC_CORR_MEM	When bit x is set to 1, a correctable ECC error occurred to SRAM instance x since it was last cleared.	R/W1CS	0x0

19.4.28 SR2PC Uncorrectable ECC Log Register

This register defines SR2PC uncorrectable ECC log.

Register name: SR2PC_UNCORR_ECC_LOG Reset value: 0x0000_0000	Register offset: 0x29814

Bits	7	6	5	4	3	2	1	0
31:24	RESERVED							
23:16	RESERVED					ECC_UNCORR_MEM[18:16]		
15:8	ECC_UNCORR_MEM[15:8]							
7:0	ECC_UNCORR_MEM[7:0]							

Bits	Name	Description	Type	Reset value
31:19	RESERVED	RESERVED	R	0x0
18:0	ECC_UNCORR_MEM	When bit x is set to 1, an uncorrectable ECC error occurred to SRAM instance x since it was last cleared.	R/W1CS	0x0

19.4.29 SR2PC PCIe Port State Register

This register defines SR2PC PCIe port status.

Register name: SR2PC_PCIE_PS Reset value: 0x0000_0000	Register offset: 0x29820

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

Bits	Name	Description	Type	Reset value
31:2	RESERVED	RESERVED	R	0x0
1:0	DSTATE	PCIe power management state: • 0b00 = D0 • 0b01 = D1 • 0b10 = D2 • 0b11 = D3	R	0x0

19.4.30 SR2PC Log Buffer Status Register

This register defines SR2PC log buffer status for NWRITE_R and maintenance response log buffer.

Register name: LOGBUF_STS Reset value: 0x0000_0000	Register offset: 0x29824

Bits	7	6	5	4	3	2	1	0
31:24	RESERVED							
23:16	RESERVED							
15:8	RESERVED							
7:0	RESERVED						MWR_LOG_BUF_ERR	NWR_LOG_BUF_ERR

Bits	Name	Description	Type	Reset value
31:2	RESERVED	RESERVED	R	0x0
1	MWR_LOG_BUF_ERR	Valid when MW_RSP_ERR or MW_RSP_TO of SR2PC General Interrupt CSR is set to 1. 0 = Maintenance write response log buffer has stored PCIe header for a maintenance write response timeout. 1 = Maintenance write response log buffer has stored PCIe header for a maintenance write error response.	R	0x0
0	NWR_LOG_BUF_ERR	Valid when NW_RSP_ERR or NW_RSP_TO SR2PC General Interrupt CSR is set to 1. 0 = NWRITE_R response log buffer has stored PCIe header for a NWRITE_R response timeout. 1 = NWRITE_R response log buffer has stored PCIe header for a NWRITE_R error response.	R	0x0

19.4.31 Tsi721 Interrupt Enable CSR

This register defines Tsi721 interrupt enable.

Register name: DEV_INTE Reset value: 0x0000_0000	Register offset: 0x29840

Bits	7	6	5	4	3	2	1	0
31:24	RESERVED							
23:16	RESERVED							
15:8	RESERVED		INT_BDMA_CH_EN	INT_BDMA_NONCH_EN	INT_SMSG_CH_EN	INT_SMSG_NONCH_EN	INT_SR2PC_CH_EN	INT_SR2PC_NONCH_EN
7:0	RESERVED	RESERVED	INT_SRIO_EN	INT_I2C_EN	RESERVED	INT_PC2SR_EN	RESERVED	RESERVED

Bits	Name	Description	Type	Reset value
31:14	RESERVED	RESERVED	R	0x0
13	INT_BDMA_CH_EN	1 = Enable INT_BDMA_CH bit of Tsi721 Interrupt CSR.	R/WS	0x0
12	INT_BDMA_NONCH_EN	1 = Enable INT_BDMA_NONCH bit of Tsi721 Interrupt CSR.	R/WS	0x0
11	INT_SMSG_CH_EN	1 = Enable INT_SMSG_CH bit of Tsi721 Interrupt CSR.	R/WS	0x0
10	INT_SMSG_NONCH_EN	1 = Enable INT_SMSG_NONCH bit of Tsi721 Interrupt CSR.	R/WS	0x0
9	INT_SR2PC_CH_EN	1 = Enable INT_SR2PC_CH bit of Tsi721 Interrupt CSR.	R/WS	0x0
8	INT_SR2PC_NONCH_EN	1 = Enable INT_SR2PC_NONCH bit of Tsi721 Interrupt CSR.	R/WS	0x0
7:6	RESERVED	RESERVED	R	0x0
5	INT_SRIO_EN	1 = Enable INT_SRIO bit of Tsi721 Interrupt CSR.	R/WS	0x0
4	INT_I2C_EN	1 = Enable INT_I2C bit of Tsi721 Interrupt CSR.	R/WS	0x0
3	RESERVED	RESERVED	R	0x0
2	INT_PC2SR_EN	1 = Enable INT_SMSG bit of Tsi721 Interrupt CSR.	R/WS	0x0
1:0	RESERVED	RESERVED	R	0x0

19.4.32 Tsi721 Interrupt CSR

This register defines Tsi721 interrupts.

Register name: DEV_INT Reset value: 0x0000_0000	Register offset: 0x29844

Bits	7	6	5	4	3	2	1	0
31:24	RESERVED							
23:16	RESERVED							
15:8	RESERVED		INT_BDMA_CH	INT_BDMA_NONCH	INT_SMSG_CH	INT_SMSG_NONCH	INT_SR2PC_CH	INT_SR2PC_NONCH
7:0	RESERVED	RESERVED	INT_SRIO	INT_I2C	RESERVED	INT_PC2SR	RESERVED	RESERVED

Bits	Name	Description	Type	Reset value
31:14	RESERVED	RESERVED	R	0x0
13	INT_BDMA_CH	1 = BDMA has an active channelized interrupt	RS	0x0
12	INT_BDMA_NONCH	1 = BDMA has an active non-channelized interrupt	RS	0x0
11	INT_SMSG_CH	1 = SMSG has an active channelized interrupt	RS	0x0
10	INT_SMSG_NONCH	1 = SMSG has an active non-channelized interrupt	RS	0x0
9	INT_SR2PC_CH	1 = SR2PC has an active channelized interrupt	RS	0x0
8	INT_SR2PC_NONCH	1 = SR2PC has an active non-channelized interrupt	RS	0x0
7:6	RESERVED	RESERVED	R	0x0
5	INT_SRIO	1 = S-RIO MAC has an active interrupt	RS	0x0
4	INT_I2C	1 = I2C has an active interrupt	RS	0x0
3	RESERVED	RESERVED	R	0x0
2	INT_PC2SR	1 = PC2SR has an interrupt	RS	0x0
1:0	RESERVED	RESERVED	R	0x0

19.4.33 Tsi721 Channel Interrupt Enable CSR

This register defines Tsi721 channel interrupt enable.

Register name: DEV_CHAN_INTE Reset value: 0x0000_0000	Register offset: 0x2984C

Bits	7	6	5	4	3	2	1	0
31:24	INT_SR2PC_CHAN_EN[7:0]							
23:16	INT_IBMSG_CHAN_EN[7:0]							
15:8	INT_OBMSG_CHAN_EN[7:0]							
7:0	INT_BDMA_CHAN_EN[7:0]							

Bits	Name	Description	Type	Reset value
31:24	INT_SR2PC_CHAN_EN	SR2PCchannel interrupt enable When INT_SR2PC_CHAN_EN[x] is set to 1, it enables INT_SR2PC_CHAN[x] bit of Tsi721 Channel Interrupt CSR.	R/WS	0x0
23:16	INT_IBMSG_CHAN_EN	Messaging Engine inbound DMA channel interrupt enable When INT_IBMSG_CHAN_EN[x] is set to 1, it enables INT_IBMSG_CHAN[x] bit of Tsi721 Channel Interrupt CSR.	R/WS	0x0
15:8	INT_OBMSG_CHAN_EN	Messaging Engine Outbound DMA channel interrupt enable When INT_OBMSG_CHAN_EN[x] is set to 1, it enables INT_OBMSG_CHAN[x] bit of Tsi721 Channel Interrupt CSR.	R/WS	0x0
7:0	INT_BDMA_CHAN_EN	Block DMA Engine channel interrupt enable When INT_BDMA_CHAN_EN[x] is set to 1, it enables INT_BDMA_CHAN[x] bit of Tsi721 Channel Interrupt CSR.	R/WS	0x0

19.4.34 Tsi721 Channel Interrupt CSR

This register defines Tsi721 channel interrupts.

Register name: DEV_CHAN_INT Reset value: 0x0000_0000	Register offset: 0x29850

Bits	7	6	5	4	3	2	1	0
31:24	INT_SR2PC_CHAN[7:0]							
23:16	INT_IBMSG_CHAN[7:0]							
15:8	INT_OBMSG_CHAN[7:0]							
7:0	INT_BDMA_CHAN[7:0]							

Bits	Name	Description	Type	Reset value
31:24	INT_SR2PC_CHAN	SR2PCchannel interrupt When INT_SR2PC_CHAN[x] is set to 1, SR2PC has an interrupt on outbound doorbell channel x or inbound doorbell queue x.	RS	0x0
23:16	INT_IBMSG_CHAN	Messaging Engine inbound DMA channel interrupt When INT_IBMSG_CHAN[x] is set to 1, Messaging Engine inbound DMA channel x has an interrupt.	RS	0x0
15:8	INT_OBMSG_CHAN	Messaging Engine Outbound DMA channel interrupt When INT_OBMSG_CHAN[x] is set to 1, Messaging Engine outbound DMA channel x has an interrupt.	RS	0x0
7:0	INT_BDMA_CHAN	Block DMA Engine channel interrupt When INT_BDMA_CHAN[x] is set to 1, Block DMA Engine channel x has an interrupt.	RS	0x0

19.4.35 Interrupt Moderation Register

These register defines the interrupt moderation timer.

Register name: INT_MOD Reset value: 0x0000_0000	Register offset: 0x29858

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

Bits	Name	Description	Type	Reset value
31:0	INT_MOD	Interrupt moderation timeout in 250-MHz SYS_CLK clock cycles. It defines the timeout period for sending a new INTx, MSI, or MSI-X. A value of all zeroes disables interrupt moderation.	R/WS	0x0

19.4.36 Received Packet Count for Messaging Engine Register

This register defines the number of received S-RIO packets for the Messaging Engine.

Register name: RXPKT_SMSG_CNT Reset value: 0x0000_0000	Register offset: 0x29900

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

Bits	Name	Description	Type	Reset value
31:0	RXPKT_SMSG_CNT	Received S-RIO packet count for Messaging Engine It is a saturating counter that counts the number of received S-RIO packets for the Messaging Engine (both inbound and outbound).	RC	0x0

19.4.37 Received Response Count for Block DMA Engine Register

This register defines the number of received S-RIO responses for the Block DMA Engine.

Register name: RXRSP_BDMA_CNT Reset value: 0x0000_0000	Register offset: 0x29904

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

Bits	Name	Description	Type	Reset value
31:0	RXRSP_BDMA_CNT	Received S-RIO response count for the Block DMA Engine. It is a saturating counter that counts the number of received S-RIO responses for Block DMA Engine.	RC	0x0

19.4.38 Received Bridging Packet Count Register

This register defines the number of received bridging (without going through BDMA/SMSG) S-RIO packets.

Register name: RXPKT_BRG_CNT Reset value: 0x0000_0000	Register offset: 0x29908

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

Bits	Name	Description	Type	Reset value
31:0	RXPKT_BRG_CNT	Received bridging S-RIO packet count. It is a saturating counter that counts the number of received bridging S-RIO packets. The following bridging S-RIO packets are not counted: • Maintenance read responses packet with TID larger than 191 (this is an error with out of range TID) • All NWRITE_R responses • All doorbell responses • All maintenance write responses	RC	0x0

19.4.39 Sent TLP Count of Messaging Engine Register

This register defines the number of PCIe TLPs sent by the Messaging Engine.

Register name: TXTLP_SMSG_CNT Reset value: 0x0000_0000	Register offset: 0x2990C

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

Bits	Name	Description	Type	Reset value
31:0	TXTLP_SMSG_CNT	Sent TLP count. It is a saturating counter that counts the number of PCIe TLPs sent by the Messaging Engine.	RC	0x0

19.4.40 Sent TLP Count of Block DMA Engine Register

This register defines the number of PCIe TLPs sent by the Block DMA Engine.

Register name: TXTLP_BDMA_CNT Reset value: 0x0000_0000	Register offset: 0x29910

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

Bits	Name	Description	Type	Reset value
31:0	TXTLP_BDMA_CNT	Sent TLP count. It is a saturating counter that counts the number of PCIe TLPs sent by the Block DMA Engine.	RC	0x0

19.4.41 Sent Bridging TLP Count Register

This register defines the number of bridging (not from SMSG/BDMA) PCIe TLPs sent by the Mapping Engine.

Register name: TXTLP_BRG_CNT Reset value: 0x0000_0000	Register offset: 0x29914

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

Bits	Name	Description	Type	Reset value
31:0	TXTLP_BRG_CNT	Sent bridging TLP count. It is a saturating counter that counts the number of bridging PCIe TLPs sent by Mapping Engine. The only bridging TLPs not counted are MWr used for MSI or MSI-X interrupts.	RC	0x0

19.4.42 Received Bridging Packet Error Count Register

This register defines a saturating counter for received bridging S-RIO packet with errors.

Register name: BRG_PKT_ERR_CNT Reset value: 0x0000_0000	Register offset: 0x2991C

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

Bits	Name	Description	Type	Reset value
31:0	BRG_PKT_ERR_CNT	Bridging Packet Error Count It is a saturating counter that counts the number of received bridging S-RIO packets with error.	RC	0x0

19.4.43 Maintenance Write Count Register

Register name: MWR_CNT Reset value: 0x0000_0000	Register offset: 0x29A00

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

Bits	Name	Description	Type	Reset value
31:16	MW_TOT_CNT	Counts the total number of maintenance writes sent to the S-RIO network, including those with OK/ERROR responses and those that encountered a timeout. It is a saturating counter.	RC	0x0
15:0	MW_OK_CNT	Counts the number of maintenance writes with OK response. It is a saturating counter.	RC	0x0

19.4.44 NWRITE_R Count Register

Register name: NWR_CNT Reset value: 0x0000_0000	Register offset: 0x29A04

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

Bits	Name	Description	Type	Reset value
31:16	NW_TOT_CNT	Counts the number of PCIe MWr bridged into NWRITE_R, including those with OK/ERROR responses and those that encountered a timeout. It is a saturating counter.	RC	0x0
15:0	NW_OK_CNT	Counts the number of PCIe MWr bridged into NWRITE_R with OK responses only. It is a saturating counter.	RC	0x0

19.4.45 Maintenance Write Response Log Buffer Data 0 Register

This register is the PCIe header 1st dword for log buffer of maintenance write responses.

Register name: MWR_LOG_DAT0 Reset value: 0x0000_0000	Register offset: 0x29A08

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

Bits	Name	Description	Type	Reset value
31:0	DATA	PCIe header 1st dword of the maintenance write response log buffer with PCIe byte 0 mapped to DATA[7:0]. Valid when MW_TO or MW_ERR of SR2PC General Interrupt CSR is 1.	R/WS	0x0

19.4.46 Maintenance Write Response Log Buffer Data 1 Register

This register is the PCIe header 2nd dword for log buffer of maintenance write responses.

Register name: MWR_LOG_DAT1 Reset value: 0x0000_0000	Register offset: 0x29A0C

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

Bits	Name	Description	Type	Reset value
31:0	DATA	PCIe header 2nd dword of the maintenance write response log buffer. Valid when MW_TO or MW_ERR of SR2PC General Interrupt CSR is 1.	R/WS	0x0

19.4.47 Maintenance Write Response Log Buffer Data 2 Register

This register is the PCIe header 3rd dword for log buffer of maintenance write responses.

Register name: MWR_LOG_DAT2 Reset value: 0x0000_0000	Register offset: 0x29A10

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

Bits	Name	Description	Type	Reset value
31:0	DATA	PCIe header 3rd dword of the maintenance write response log buffer. Valid when MW_TO or MW_ERR of SR2PC General Interrupt CSR is 1.	R/WS	0x0

19.4.48 Maintenance Write Response Log Buffer Data 3 Register

This register is the PCIe header 4th dword for log buffer of maintenance write responses.

Register name: MWR_LOG_DAT3 Reset value: 0x0000_0000	Register offset: 0x29A14

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

Bits	Name	Description	Type	Reset value
31:0	DATA	PCIe header 4th dword of maintenance write response log buffer. Valid when MW_TO or MW_ERR of SR2PC General Interrupt CSR is 1.	R/WS	0x0

19.4.49 NWRITE_R Response Log Buffer Data 0 Register

This register is the PCIe header 1st dword for log buffer of NWRITE_R responses.

Register name: NWR_LOG_DAT0 Reset value: 0x0000_0000	Register offset: 0x29A18

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

Bits	Name	Description	Type	Reset value
31:0	DATA	PCIe header 1st dword of the NWRITE_R response log buffer with PCIe byte 0 mapped to DATA[7:0]. Valid when NW_TO or NW_ERR of SR2PC General Interrupt CSR is 1.	R/WS	0x0

19.4.50 NWRITE_R Response Log Buffer Data 1 Register

This register is the PCIe header 2nd dword for log buffer of NWRITE_R responses.

Register name: NWR_LOG_DAT1 Reset value: 0x0000_0000	Register offset: 0x29A1C

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

Bits	Name	Description	Type	Reset value
31:0	DATA	PCIe header 2nd dword of the NWRITE_R response log buffer. Valid when NW_TO or NW_ERR of SR2PC General Interrupt CSR is 1.	R/WS	0x0

19.4.51 NWRITE_R Response Log Buffer Data 2 Register

This register is the PCIe header 3rd dword for log buffer of NWRITE_R responses.

Register name: NWR_LOG_DAT2 Reset value: 0x0000_0000	Register offset: 0x29A20

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

Bits	Name	Description	Type	Reset value
31:0	DATA	PCIe header 3rd dword of the NWRITE_R response log buffer. Valid when NW_TO or NW_ERR of SR2PC General Interrupt CSR is 1.	R/WS	0x0

19.4.52 NWRITE_R Response Log Buffer Data 3 Register

This register is the PCIe header 4th dword for log buffer of NWRITE_R responses.

Register name: NWR_LOG_DAT3 Reset value: 0x0000_0000	Register offset: 0x29A24

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

Bits	Name	Description	Type	Reset value
31:0	DATA	PCIe header 4th dword of NWRITE_R response log buffer. Valid when NW_TO or NW_ERR of SR2PC General Interrupt CSR is 1.	R/WS	0x0

19.4.53 MSI-X Pending Bit Array Lower Register

This register defines the lower 32 bits of the MSI-X pending bit array (for more information, see MSI-X for vector assignment).

Register name: MSIX_PBAL Reset value: 0x0000_0000	Register offset: 0x2A000

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

Bits	Name	Description	Type	Reset value
31:0	PENDING	MSI-X pending bits 31:0. Valid in MSI-X mode.	R	0x0

19.4.54 MSI-X Pending Bit Array Middle Register

This register defines the middle 32 bits of the MSI-X pending bit array (for more information, see MSI-X for vector assignment).

Register name: MSIX_PBAM Reset value: 0x0000_0000	Register offset: 0x2A004

Bits	7	6	5	4	3	2	1	0
31:24	PENDING[63:56]							
23:16	PENDING[55:48]							
15:8	PENDING[47:40]							
7:0	PENDING[39:32]							

Bits	Name	Description	Type	Reset value
31:0	PENDING	MSI-X pending bits 63:32. Valid in MSI-X mode.	R	0x0

19.4.55 MSI-X Pending Bit Array Upper Register

This register defines the upper 6 bits of the MSI-X pending bit array (for more information, see MSI-X for vector assignment).

Register name: MSIX_PBAU Reset value: 0x0000_0000	Register offset: 0x2A008

Bits	7	6	5	4	3	2	1	0
31:24	RESERVED							
23:16	RESERVED							
15:8	RESERVED							
7:0	RESERVED		PENDING[69:64]					

Bits	Name	Description	Type	Reset value
31:6	Reserved	Reserved	R	0x0
5:0	PENDING	MSI-X pending bits 69:64. Valid in MSI-X mode.	R	0x0

19.4.56 MSI-X Table Entry Message Lower Address Register {0..69}

This register defines the message lower address of a MSI-X table entry. Register Y is for MSI table entry Y (for more information, see MSI-X for vector assignment).

Register name: MSIX_TAB_ADDRL{0..69} Reset value: 0x0000_0000	Register offset: 0x2C000 += 10

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

Bits	Name	Description	Type	Reset value
31:0	ADDR	MSI-X address bits 31:0	R/W	0x0

19.4.57 MSI-X Table Entry Message Upper Address Register {0..69}

This register defines the message upper address of a MSI-X table entry. Register Y is for MSI table entry Y (for more information, see MSI-X for vector assignment).

Register name: MSIX_TAB_ADDRU{0..69} Reset value: 0x0000_0000	Register offset: 0x2C004 += 10

Bits	7	6	5	4	3	2	1	0
31:24	ADDR[63:56]							
23:16	ADDR[55:48]							
15:8	ADDR[47:40]							
7:0	ADDR[39:32]							

Bits	Name	Description	Type	Reset value
31:0	ADDR	MSI-X address bits 63:32	R/W	0x0

19.4.58 MSI-X Table Entry Message Data Register {0..69}

This register defines the message data of a MSI-X table entry. Register Y is for MSI table entry Y (for more information, see MSI-X for vector assignment).

Register name: MSIX_TAB_DATA{0..69} Reset value: 0x0000_0000	Register offset: 0x2C008 += 10

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

Bits	Name	Description	Type	Reset value
31:0	DATA	MSI-X DATA bits 31:0	R/W	0x0

19.4.59 MSI-X Table Entry Mask Register {0..69}

This register defines the mask of a MSI-X table entry. Register Y is for MSI table entry Y (for more information, see MSI-X for vector assignment).

Register name: MSIX_TAB_MSK{0..69} Reset value: 0x0000_0001	Register offset: 0x2C00C += 10

Bits	7	6	5	4	3	2	1	0
31:24	RESERVED							
23:16	RESERVED							
15:8	RESERVED							
7:0	RESERVED							MASK

Bits	Name	Description	Type	Reset value
31:1	RESERVED	RESERVED	R	0x0
0	MASK	MSI-X mask bit	R/W	0x1

20. Messaging Engine Registers

Topics discussed include the following:

- Register Map
- Register Descriptions

20.1 Register Map

For registers defined for multiple channels such as Outbound DMA Descriptor Write Count Register {0..7}, the register offset between two adjacent channel is 4 KB; that is, in 0x61000 += 1000, register for channel 0 has address 0x61000, += denotes the register offset between channels, and 1000 is in hexidecimal.

Table 95: Messaging Engine Register Map

Offset	Register Name	See
0x60010	RQRPTO	S-RIO Message Request/Response Timeout Register
0x60020	IB_DEVID	Inbound Messaging Engine deviceID Register
0x61000 += 1000	OBDMAC{0..7}DWRCNT	Outbound DMA Descriptor Write Count Register {0..7}
0x61004 += 1000	OBDMAC{0..7}DRDCNT	Outbound DMA Descriptor Read Count Register {0..7}
0x61008 += 1000	OBDMAC{0..7}CTL	Outbound DMA Channel Control Register {0..7}
0x6100C += 1000	OBDMAC{0..7}INT	Outbound DMA Channel Interrupt Register {0..7}
0x61010 += 1000	OBDMAC{0..7}INTSET	Outbound DMA Channel Interrupt Set Register {0..7}
0x61014 += 1000	OBDMAC{0..7}STS	Outbound DMA Channel Status Register {0..7}
0x61018 += 1000	OBDMAC{0..7}INTE	Outbound DMA Channel Interrupt Enable Register {0..7}
0x6101C += 1000	OBDMAC{0..7}PWE	Outbound DMA Channel Port-Write Enable Register {0..7}
0x61020 += 1000	OBDMAC{0..7}DPTRL	Outbound DMA Channel Descriptor Pointer Low Register {0..7}
0x61024 += 1000	OBDMAC{0..7}DPTRH	Outbound DMA Channel Descriptor Pointer High Register {0..7}
0x61040 += 1000	OBDMAC{0..7}DSBL	Outbound DMA Descriptor Status FIFO Lower Base Register {0..7}
0x61044 += 1000	OBDMAC{0..7}DSBH	Outbound DMA Descriptor Status FIFO Upper Base Register {0..7}
0x61048 += 1000	OBDMAC{0..7}DSSZ	Outbound DMA Descriptor Status FIFO Size Register {0..7}
0x6104C += 1000	OBDMAC{0..7}DSRP	Outbound DMA Descriptor Status FIFO Read Pointer Register {0..7}
0x61050 += 1000	OBDMAC{0..7}DSWP	Outbound DMA Descriptor Status FIFO Write Pointer Register {0..7}

Table 95: Messaging Engine Register Map *(Continued)*

Offset	Register Name	See
0x61200 += 1000	IBDMAC{0..7}FQBL	Inbound Circular Free Queue Lower Base Register {0..7}
0x61204 += 1000	IBDMAC{0..7}FQBH	Inbound Circular Free Queue Upper Base Register {0..7}
0x61208 += 1000	IBDMAC{0..7}FQSZ	Inbound Circular Free Queue Size Register {0..7}
0x6120C += 1000	IBDMAC{0..7}FQRP	Inbound Circular Free Queue Read Pointer Register {0..7}
0x61210 += 1000	IBDMAC{0..7}FQWP	Inbound Circular Free Queue Write Pointer Register {0..7}
0x61214 += 1000	IBDMAC{0..7}FQTH	Inbound Circular Free Queue Threshold Register {0..7}
0x61240 += 1000	IBDMAC{0..7}CTL	Inbound DMA Channel Control Register {0..7}
0x61244 += 1000	IBDMAC{0..7}STS	Inbound DMA Channel Status Register {0..7}
0x61248 += 1000	IBDMAC{0..7}INT	Inbound DMA Channel Interrupt Register {0..7}
0x6124C += 1000	IBDMAC{0..7}INTSET	Inbound DMA Channel Interrupt Set Register {0..7}
0x61250 += 1000	IBDMAC{0..7}INTE	Inbound DMA Channel Interrupt Enable Register {0..7}
0x61254 += 1000	IBDMAC{0..7}PWE	Inbound DMA Channel Port-Write Enable Register {0..7}
0x61300 += 1000	IBDMAC{0..7}DQBL	Inbound Descriptor Queue Lower Base Register {0..7}
0x61304 += 1000	IBDMAC{0..7}DQBH	Inbound Descriptor Queue Upper Base Register {0..7}
0x61308+= 1000	IBDMAC{0..7}DQRP	Inbound Circular Descriptor Queue Read Pointer Register {0..7}
0x6130C += 1000	IBDMAC{0..7}DQWP	Inbound Circular Descriptor Queue Write Pointer Register {0..7}
0x61314 += 1000	IBDMAC{0..7}DQSZ	Inbound Circular Descriptor Queue Size Register {0..7}
0x6A000	SMSG_INTE	Messaging Engine Interrupt Enable Register
0x6A004	SMSG_PWE	Messaging Engine Port-Write Enable Register
0x6A008	SMSG_INT	Messaging Engine Interrupt Register
0x6A00C	SMSG_PW	Messaging Engine Port-Write Register
0x6A010	SMSG_INTSET	Messaging Engine Interrupt Set Register
0x6A014	SMSG_ECC_LOG	Messaging Engine non-channelized ECC LOG Register
0x6A100	RETRY_GEN_CNT	Generated Message Segment Retry Count Register
0x6A104	RETRY_RX_CNT	Received Retry Message Response Count Register
0x6A300 += 4	SMSG_ECC_CORR{0..7}LOG	Messaging Engine Channelized Correctable ECC LOG Register {0..7}
0x6A340 += 4	SMSG_ECC_UNCORR{0..7}LOG	Messaging Engine Channelized Uncorrectable ECC LOG Register {0..7}

20.2 Register Descriptions

20.2.1 Outbound DMA Descriptor Write Count Register {0..7}

This register defines the Outbound DMA descriptor write count function. Register Y (Y = 0–7) is assigned to Outbound DMA channel Y.

Register name: OBDMAC{0..7}DWRCNT Reset value: 0x0000_0000	Register offset: 0x61000 += 1000

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

Bits	Name	Description	Type	Reset value
31:0	DWRCNT	It defines the 32-bit descriptor write count. Software must update this register in a timely manner in order not to stall Tsi721 descriptor processing. Under normal conditions, the Tsi721 continues descriptor prefetching until DRDCNT of Outbound DMA Descriptor Read Count Register {0..7} equals DWRCNT. The Tsi721 resets DWRCNT when the INIT bit in Outbound DMA Channel Control Register {0..7} toggles from 0 to 1.	R/W	0x0

20.2.2 Outbound DMA Descriptor Read Count Register {0..7}

This register defines the Outbound DMA descriptor read count function. Register Y (Y = 0–7) is assigned to Outbound DMA channel Y.

Register name: OBDMAC{0..7}DRDCNT Reset value: 0x0000_0000	Register offset: 0x61004 += 1000

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

Bits	Name	Description	Type	Reset value
31:0	DRDCNT	It defines the 32-bit descriptor read count. The Tsi721 updates this register after a descriptor prefetch. It resets DRDCNT when the INIT bit in Outbound DMA Channel Control Register {0..7} toggles from 0 to 1.	R/W	0x0

20.2.3 Outbound DMA Channel Control Register {0..7}

This register defines the Outbound DMA channel control function. Outbound DMA channel Y (Y = 0–7) is assigned to register Y.

Register name: OBDMAC{0..7}CTL Reset value: 0x0000_0000	Register offset: 0x61008 += 1000

Bits	7	6	5	4	3	2	1	0
31:24	RESERVED							
23:16	RESERVED							
15:8	RESERVED							
7:0	RESERVED					RETRY_THR	SUSPEND	INIT

Bits	Name	Description	Type	Reset value
31:3	RESERVED	RESERVED	R	0x0
2	RETRY_THR	Excessive Message Retry Error Threshold 0 = Declare excessive retry error on 257th message segment retry of a message 1 = Allow infinite retries for a message; that is, never declare excessive retry errors	R/WS	0x0
1	SUSPEND	Outbound DMA channel suspend. 1 = Tsi721 suspends Outbound DMA descriptor processing for this Outbound DMA channel. Tsi721 automatically clears this bit to 0 after it has finished the suspend procedure (SUSPENDED bit of Outbound DMA Channel Interrupt Register {0..7} is set to 1). Tsi721 resumes DMA descriptor processing for this DMA channel when it detects a non-zero Outbound DMA Descriptor Write Count Register {0..7} after this bit has been cleared to zero. Note: Software must not set this bit to 1 while the INIT bit is 1; otherwise, device behavior will be undefined.	R/W	0x0

(Continued)

Bits	Name	Description	Type	Reset value
0	INIT	Outbound DMA channel initialization. 1 = Tsi721 resets this DMA channel, including internal logic and registers, Outbound DMA Descriptor Read Count Register {0..7} Outbound DMA Descriptor Write Count Register {0..7}, and Outbound DMA Channel Status Register {0..7} for this channel. Outbound DMA Channel Status Register {0..7} can only be accessed at least 40 ns after the Tsi721 clears INIT from 1 to 0. Software can set INIT to 1 only when this Outbound DMA channel is idle; that is, Outbound DMA Channel Status Register {0..7} RUN bit is 0. The Tsi721 resets the INIT bit when software writes to Outbound DMA Descriptor Write Count Register {0..7}.	R/W	0x0

20.2.4 Outbound DMA Channel Interrupt Register {0..7}

This register defines the Outbound DMA channel interrupt function. Outbound DMA channel Y (Y = 0–7) is assigned to register Y.

Register name: OBDMAC{0..7}INT Reset value: 0x0000_0000	Register offset: 0x6100C += 1000

Bits	7	6	5	4	3	2	1	0
31:24	RESERVED							
23:16	RESERVED							
15:8	RESERVED							
7:0	RESERVED	RESERVED	RESERVED	ST_FULL	DONE	SUSPENDED	ERROR	IOF_DONE

Bits	Name	Description	Type	Reset value
31:5	RESERVED	RESERVED	R	0x0
4	ST_FULL	1 = Tsi721 detected that the descriptor status FIFO in the PCIe side main memory for this DMA channel has become full at least once since this bit was last cleared.	R/W1CS	0x0
3	DONE	1 = Tsi721 reached the end of the descriptor list without errors (finished processing all descriptors within the descriptor list of this DMA channel without any errors) at least once since this bit was last cleared.	R/W1CS	0x0
2	SUSPENDED	1 = Tsi721 suspended the processing of descriptors for this DMA channel as triggered by SUSPEND bit of Outbound DMA Channel Control Register {0..7} at least once since this bit was last cleared.	R/W1CS	0x0
1	ERROR	1 = Tsi721 detected a PCIe or S-RIO error for this DMA channel at least once since this bit was last cleared. For information about the specific error, see CS in Outbound DMA Channel Status Register {0..7}.	R/W1CS	0x0
0	IOF_DONE	1 = Tsi721 finished processing a descriptor with the IOF bit set for this DMA channel at least once since this bit was last cleared.	R/W1CS	0x0

20.2.5 Outbound DMA Channel Interrupt Set Register {0..7}

This register defines the DMA channel interrupt debug function that sets individual interrupt bits directly. DMA channel Y (Y = 0–7) is assigned to register Y.

Register name: OBDMAC{0..7}INTSET Reset value: 0x0000_0000	Register offset: 0x61010 += 1000

Bits	7	6	5	4	3	2	1	0
31:24	RESERVED							
23:16	RESERVED							
15:8	RESERVED							
7:0	RESERVED	RESERVED	RESERVED	ST_FULL_SET	DONE_SET	SUSPENDED_SET	ERROR_SET	IOF_DONE_SET

Bits	Name	Description	Type	Reset value
31:5	RESERVED	RESERVED	R	0x0
4	ST_FULL_SET	1 = Set the ST_FULL bit of Outbound DMA Channel Interrupt Register {0..7} to 1.	R/W1S	0x0
3	DONE_SET	1 = Set the DONE bit of Outbound DMA Channel Interrupt Register {0..7} is forced to 1.	R/W1S	0x0
2	SUSPENDED_SET	1 = Set the SUSPENDED bit of Outbound DMA Channel Interrupt Register {0..7} to 1.	R/W1S	0x0
1	ERROR_SET	1 = Set the ERROR bit of Outbound DMA Channel Interrupt Register {0..7} to 1.	R/W1S	0x0
0	IOF_DONE_SET	1 = Set the IOF_DONE bit of Outbound DMA Channel Interrupt Register {0..7} to 1.	R/W1S	0x0

20.2.6 Outbound DMA Channel Status Register {0..7}

This register defines the Outbound DMA channel status function. Outbound DMA channel Y (Y = 0–7) is assigned to register Y.

Register name: OBDMAC{0..7}STS Reset value: 0x0000_0000	Register offset: 0x61014 += 1000

Bits	7	6	5	4	3	2	1	0
31:24	RESERVED							
23:16	RESERVED	ABORT	RUN	CS[4:0]				
15:8	RESERVED							
7:0	RESERVED							

Bits	Name	Description	Type	Reset value
31:23	RESERVED	RESERVED	R	0x0
22	ABORT	1 = Tsi721 aborted processing descriptors for this outbound DMA channel due to an error condition. Software should write this bit only during software debug.	R/W	0x0
21	RUN	0 = The DMA channel is aborted or suspended or has processed the last descriptor of its descriptor list. 1 = Tsi721 is processing descriptors for this DMA channel. During outbound DMA channel initialization, the RUN bit is set to 1 when software writes to Outbound DMA Descriptor Write Count Register {0..7}.	R	0x0
20:16	CS	Error completion status for this Outbound DMA channel: • 00100 = S-RIO excessive message retry has occurred • 00101 = S-RIO response timeout occurred • 11111 = S-RIO message ERROR response received • 01000 = PCIe cpl/clpD timeout occurred • 01001 = PCIe poisoned cpl/clpD received • 01011 = PCIe cpl/clpD received is malformed • 01100 = PCIe cpl/clpD received has completion status of UR • 01101 = PCIe cpl/clpD received has completion status of CA • Others: Reserved Software should write this bit only during software debug.	R/WS	0x0
15:0	RESERVED	RESERVED	R	0x0

20.2.7 Outbound DMA Channel Interrupt Enable Register {0..7}

This register defines the Outbound DMA channel interrupt enable function. Outbound DMA channel Y (Y = 0–7) is assigned to register Y.

Register name: OBDMAC{0..7}INTE Reset value: 0x0000_0001	Register offset: 0x61018 += 1000

Bits	7	6	5	4	3	2	1	0
31:24	RESERVED							
23:16	RESERVED							
15:8	RESERVED							
7:0	RESERVED	RESERVED	RESERVED	ST_FULL_EN	DONE_EN	SUSPENDED_EN	ERROR_EN	IOF_DONE_EN

Bits	Name	Description	Type	Reset value
31:5	RESERVED	RESERVED	R	0x0
4	ST_FULL_EN	1 = Enable interrupt generation for ST_FULL bit of Outbound DMA Channel Interrupt Register {0..7}.	R/WS	0x0
3	DONE_EN	1 = Enable interrupt generation for DONE bit of Outbound DMA Channel Interrupt Register {0..7}.	R/WS	0x0
2	SUSPENDED_EN	1 = Enable interrupt generation for SUSPENDED bit of Outbound DMA Channel Interrupt Register {0..7}.	R/WS	0x0
1	ERROR_EN	1 = Enable interrupt generation for ERROR bit of Outbound DMA Channel Interrupt Register {0..7}.	R/WS	0x0
0	IOF_DONE_EN	1 = Enable interrupt generation for IOF_DONE bit of Outbound DMA Channel Interrupt Register {0..7}. When using MSI-X, this bit must be set to 1.	R/WS	0x1

20.2.8 Outbound DMA Channel Port-Write Enable Register {0..7}

This register defines the Outbound DMA channel port-write enable function. Outbound DMA channel Y (Y = 0–7) is assigned to register Y.

Register name: OBDMAC{0..7}PWE Reset value: 0x0000_0000	Register offset: 0x6101C += 1000

Bits	7	6	5	4	3	2	1	0
31:24	RESERVED							
23:16	RESERVED							
15:8	RESERVED							
7:0	RESERVED						ERROR_EN	RESERVED

Bits	Name	Description	Type	Reset value
31:2	RESERVED	RESERVED	R	0x0
1	ERROR_EN	1 = Enable port-write generation for CS[4:0] bits of Outbound DMA Channel Status Register {0..7} if CS[4:0] has a value of 0b00100, 0b00101, or 0b11111.	R/WS	0x0
0	RESERVED	RESERVED	R	0x0

20.2.9 Outbound DMA Channel Descriptor Pointer Low Register {0..7}

This register defines the lower 8 bits of the Outbound DMA channel descriptor pointer. Register Y (Y = 0–7) is assigned to Outbound DMA channel Y.

Register name: OBDMAC{0..7}DPTRL Reset value: 0x0000_0000	Register offset: 0x61020 += 1000

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

Bits	Name	Description	Type	Reset value
31:4	DPTRL	This field defines the bits 31:4 of the 16-byte aligned descriptor pointer for this DMA channel. Together with DPTRH in Outbound DMA Channel Descriptor Pointer High Register {0..7}, they define the 64-bit DMA channel descriptor pointer to the next descriptor to be processed. Software can read/write DPTRL and DPTRH only when the RUN bit is 0 in Outbound DMA Channel Status Register {0..7}; otherwise, since the Tsi721 is constantly updating DPTRL and DPTRH while software can only read DPTRL or DPTRH at a time, software will read inconsistent values. When the INIT bit of Outbound DMA Channel Control Register {0..7} toggles from 0 to 1, the descriptor pointer points to the first descriptor of the descriptor list. When the ABORT bit of Outbound DMA Channel Status Register {0..7} is set to 1, the descriptor pointer is undefined; otherwise, the descriptor pointer points to the next descriptor to be fetched.	R/WS	0x0
3:0	RESERVED	RESERVED	R	0x0

20.2.10 Outbound DMA Channel Descriptor Pointer High Register {0..7}

This register defines the upper 32 bits of the Outbound DMA channel descriptor pointer. Register Y (Y = 0–7) is assigned to Outbound DMA channel Y.

Register name: OBDMAC{0..7}DPTRH Reset value: 0x0000_0000	Register offset: 0x61024 += 1000

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

Bits	Name	Description	Type	Reset value
31:0	DPTRH	This field defines the upper 32 bits of the descriptor pointer for this Outbound DMA channel (see Outbound DMA Channel Interrupt Set Register {0..7}). Software should read/write DPTRL and DPTRH only when the RUN bit in Outbound DMA Channel Status Register {0..7} is 0.	R/WS	0x0

20.2.11 Outbound DMA Descriptor Status FIFO Lower Base Register {0..7}

This register defines the lower 26 bits of the base address of the Outbound DMA descriptor status FIFO. Outbound DMA channel Y (Y = 0–7) is assigned to register Y.

Register name: OBDMAC{0..7}DSBL Reset value: 0x0000_0000	Register offset: 0x61040 += 1000

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

Bits	Name	Description	Type	Reset value
31:6	ADD	26 LSB bits of Outbound DMA descriptor status base address. The base address must be 64-byte aligned.	R/WS	0x0
5:0	Reserved	RESERVED	R	0x0

20.2.12 Outbound DMA Descriptor Status FIFO Upper Base Register {0..7}

This register defines the upper 32 bits of the base address of the Outbound DMA descriptor status FIFO. Outbound DMA channel Y (Y = 0–7) is assigned to register Y.

Register name: OBDMAC{0..7}DSBH Reset value: 0x0000_0000	Register offset: 0x61044 += 1000

Bits	7	6	5	4	3	2	1	0
31:24	ADD[63:56]							
23:16	ADD[55:48]							
15:8	ADD[47:40]							
7:0	ADD[39:32]							

Bits	Name	Description	Type	Reset value
31:0	ADD	32 MSB bits of descriptor status FIFO base address	R/WS	0x0

20.2.13 Outbound DMA Descriptor Status FIFO Size Register {0..7}

This register defines the size of the Outbound DMA descriptor status FIFO. Outbound DMA channel Y (Y = 0–7) is assigned to register Y.

Register name: OBDMAC{0..7}DSSZ Reset value: 0x0000_0005	Register offset: 0x61048 += 1000

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

Bits	Name	Description	Type	Reset value
31:4	RESERVED	RESERVED	R	0x0
3:0	SIZE	SIZE 0b0000 = Reserved 0b0001 = 32 entries 0b0010 = 64 entries 0b0011 = 128 entries 0b0100 = 256 entries 0b0101 = 512 entries 0b0110 = 1K entries 0b0111 = 2K entries 0b1000 = 4K entries 0b1001 = 8K entries 0b1010 = 16K entries 0b1011 = 32K entries 0b1100 = 64K entries 0b1101 = 128K entries 0b1110 = 256K entries 0b1111 = 512K entries Others = Reserved	R/WS	0x5

20.2.14 Outbound DMA Descriptor Status FIFO Read Pointer Register {0..7}

This register defines the read pointer of the Outbound DMA descriptor status FIFO. Outbound DMA channel Y (Y = 0–7) is assigned to register Y.

Register name: OBDMAC{0..7}DSRP Reset value: 0x0000_0000	Register offset: 0x6104C += 1000

Bits	7	6	5	4	3	2	1	0
31:24	RESERVED							
23:16	RESERVED					RD_PTR[18:16]		
15:8	RD_PTR[15:8]							
7:0	RD_PTR[7:0]							

Bits	Name	Description	Type	Reset value
31:19	RESERVED	RESERVED	R	0x0
18:0	RD_PTR	Read pointer to an entry of the descriptor status FIFO relative to descriptor status FIFO base address. Software must update this register in a timely manner in order not to stall Tsi721 descriptor processing. Tsi721 launches descriptor processing only if WR_PTR in Outbound DMA Descriptor Status FIFO Write Pointer Register {0..7} does not equal RD_PTR; that is, the Tsi721 sees the queue full when (WR_PTR+1) MOD Outbound DMA Descriptor Status FIFO Size Register {0..7} = RD_PTR.	R/W	0x0

20.2.15 Outbound DMA Descriptor Status FIFO Write Pointer Register {0..7}

This register defines the write pointer of the Outbound DMA descriptor status FIFO. Outbound DMA channel Y (Y = 0–7) is assigned to register Y.

Register name: OBDMAC{0..7}DSWP Reset value: 0x0000_0000	Register offset: 0x61050 += 1000

Bits	7	6	5	4	3	2	1	0
31:24	RESERVED							
23:16	RESERVED					WR_PTR[18:16]		
15:8	WR_PTR[15:8]							
7:0	WR_PTR[7:0]							

Bits	Name	Description	Type	Reset value
31:19	RESERVED	RESERVED	R	0x0
18:0	WR_PTR	Write pointer to an entry of the descriptor status FIFO relative to the descriptor status FIFO base address. It is updated by Tsi721 after issuing descriptor status MWr to PCIe MAC.	R/W	0x0

20.2.16 S-RIO Message Request/Response Timeout Register

This register defines the request and response timeout for outbound/inbound messages for all Tx/Rx queues.

Register name: RQRPTO Reset value: 0x00FF_FFFF	Register offset: 0x60010

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

Bits	Name	Description	Type	Reset value
31:24	RESERVED	RESERVED	R	0x0
23:0	REQ_RSP_TO	Message request and response timeout threshold. The actual timeout range is between 8*REQ_RSP_TO 250-MHz SYS_CLK cycles to 64*RSP_TO 250-MHz SYS_CLK cycles. A value of 0 disables request and response timeout.	R/WS	0xFFFFFF

20.2.17 Inbound Messaging Engine deviceID Register

This register defines the S-RIO deviceID for mapping inbound messages.

Register name: IB_DEVID Reset value: 0x0000_0000	Register offset: 0x60020

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

Bits	Name	Description	Type	Reset value
31:16	RESERVED	RESERVED	R	0x0
15:0	DEVID	Messages with destID matching DEVID are mapped to Rx queues 4/5/6/7 based on its mbox header field value of 0/1/2/3. Messages with destID mismatching DEVID are mapped to Rx queues 0/1/2/3 based on its mbox header field value of 0/1/2/3. If a message’s S-RIO header tt field indicates an 8-bit deviceID, then only DEVID[7:0] is used for destID matching.	R/WS	0x0

20.2.18 Inbound Circular Free Queue Lower Base Register {0..7}

This register defines the lower 32 bits of the base address of the inbound circular free queue. Rx queue X (X = 0–7) is assigned to register X.

Register name: IBDMAC{0..7}FQBL Reset value: 0x0000_0000	Register offset: 0x61200 += 1000

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

Bits	Name	Description	Type	Reset value
31:6	ADD	Bits 31:6 of the base address of the inbound circular free queue. The base address must be 64-byte aligned.	R/WS	0x0
5:0	RESERVED	RESERVED	R	0x0

20.2.19 Inbound Circular Free Queue Upper Base Register {0..7}

This register defines the upper 32 bits of the base address of the inbound circular free queue. Rx queue Y (Y = 0–7) is assigned to register Y.

Register name: IBDMAC{0..7}FQBH Reset value: 0x0000_0000	Register offset: 0x61204 += 1000

Bits	7	6	5	4	3	2	1	0
31:24	ADD[63:56]							
23:16	ADD[55:48]							
15:8	ADD[47:40]							
7:0	ADD[39:32]							

Bits	Name	Description	Type	Reset value
31:0	ADD	Upper 32 bits of the base address of the inbound circular free queue.	R/WS	0x0

20.2.20 Inbound Circular Free Queue Size Register {0..7}

This register defines the size of the inbound circular free queue. Rx queue Y (Y = 0–7) is assigned to register Y.

Register name: IBDMAC{0..7}FQSZ Reset value: 0x0000_0005	Register offset: 0x61208 += 1000

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

Bits	Name	Description	Type	Reset value
31:4	RESERVED	RESERVED	R	0x0
3:0	SIZE	0b0000 = 16 entries 0b0001 = 32 entries 0b0010 = 64 entries 0b0011 = 128 entries 0b0100 = 256 entries 0b0101 = 512 entries 0b0110 = 1K entries 0b0111 = 2K entries 0b1000 = 4K entries 0b1001 = 8K entries 0b1010 = 16K entries 0b1011 = 32K entries 0b1100 = 64K entries 0b1101 = 128K entries 0b1110 = 256K entries 0b1111 = 512K entries	R/WS	0x5

20.2.21 Inbound Circular Free Queue Read Pointer Register {0..7}

This register defines the read pointer of the inbound circular free queue. Rx queue Y (Y = 0–7) is assigned to register Y.

Register name: IBDMAC{0..7}FQRP Reset value: 0x0000_0000	Register offset: 0x6120C += 1000

Bits	7	6	5	4	3	2	1	0
31:24	RESERVED							
23:16	RESERVED					RD_PTR[18:16]		
15:8	RD_PTR[15:8]							
7:0	RD_PTR[7:0]							

Bits	Name	Description	Type	Reset value
31:19	RESERVED	RESERVED	R	0x0
18:0	RD_PTR	Read pointer to an entry of the inbound circular free queue relative to the inbound circular free queue base address. Under normal operations, software can only read this register; Tsi721 is responsible for updating it.	R/W	0x0

This document is confidential and is subject to an NDA.

20.2.22 Inbound Circular Free Queue Write Pointer Register {0..7}

This register defines the write pointer of the inbound circular free queue. Rx queue Y (Y = 0–7) is assigned register Y.

Register name: IBDMAC{0..7}FQWP Reset value: 0x0000_0000	Register offset: 0x61210 += 1000

Bits	7	6	5	4	3	2	1	0
31:24	RESERVED							
23:16	RESERVED					WR_PTR[18:16]		
15:8	WR_PTR[15:8]							
7:0	WR_PTR[7:0]							

Bits	Name	Description	Type	Reset value
31:19	RESERVED	RESERVED	R	0x0
18:0	WR_PTR	Write pointer to an entry of the inbound circular free queue relative to the inbound circular free queue base address. It is updated by software. Tsi721 initiated Rx queue messaging when WR_PTR does not equal RD_PTR in the Inbound Circular Free Queue Read Pointer Register {0..7}. Once started performing messaging, Tsi721 reads the inbound circular descriptor queue until RD_PTR in the Inbound Circular Free Queue Read Pointer Register {0..7} equals WR_PTR.	R/W	0x0

20.2.23 Inbound Circular Free Queue Threshold Register {0..7}

This register defines the half-empty threshold of the inbound circular free queue. Rx queue Y (Y = 0–7) is assigned register Y.

Register name: IBDMAC{0..7}FQTH Reset value: 0x0000_0000	Register offset: 0x61214 += 1000

Bits	7	6	5	4	3	2	1	0
31:24	RESERVED							
23:16	RESERVED					IBFQ_TH[18:16]		
15:8	IBFQ_TH[15:8]							
7:0	IBFQ_TH[7:0]							

Bits	Name	Description	Type	Reset value
31:19	RESERVED	RESERVED	R	0x0
18:0	IBFQ_TH	Software configured half-empty threshold of the inbound free queue. When the Tsi721 detects that hardware own level of the inbound free queue falls below this threshold, it sets the associated interrupt FQ_LOW in the Inbound DMA Channel Interrupt Register {0..7}.	R/WS	0x0

20.2.24 Inbound DMA Channel Control Register {0..7}

This register defines the inbound DMA channel control function. Inbound DMA channel Y (Y = 0–7) is assigned to register Y.

Register name: IBDMAC{0..7}CTL Reset value: 0x0000_0000	Register offset: 0x61240 += 1000

Bits	7	6	5	4	3	2	1	0
31:24	RESERVED							
23:16	RESERVED							
15:8	RESERVED							
7:0	RESERVED						SUSPEND	INIT

Bits	Name	Description	Type	Reset value
31:2	RESERVED	RESERVED	R	0x0
1	SUSPEND	DMA channel suspend (tear down). 1 = Suspend DMA descriptor processing for this DMA channel. Tsi721 automatically clears this bit to 0 after it finishes the suspend procedure (SUSPENDED bit of the Inbound DMA Channel Interrupt Register {0..7} is set to 1). Note: Software must not set this bit to 1 while the INIT bit is 1; otherwise, device behavior will be undefined.	R/W	0x0
0	INIT	DMA channel initialization 1 = Start DMA descriptor processing for this DMA channel and reset the Inbound DMA Channel Status Register {0..7}, Inbound Circular Free Queue Read Pointer Register {0..7}, Inbound Circular Free Queue Write Pointer Register {0..7}, Inbound Circular Descriptor Queue Read Pointer Register {0..7}, and Inbound Circular Descriptor Queue Write Pointer Register {0..7}. The Inbound DMA Channel Status Register {0..7} can only be accessed at least 40 ns after Tsi721 has cleared INIT from 1 to 0. Software can set INIT to 1 only when this DMA channel is idle; that is, Inbound DMA Channel Status Register {0..7} RUN bit is 0. The Tsi721 clears this bit when software writes to the Inbound Circular Free Queue Write Pointer Register {0..7}.	R/W	0x0

20.2.25 Inbound DMA Channel Status Register {0..7}

This register defines the inbound DMA channel status function. Inbound DMA channel Y (Y = 0–7) is assigned to register Y.

Register name: IBDMAC{0..7}STS Reset value: 0x0000_0000	Register offset: 0x61244 += 1000

Bits	7	6	5	4	3	2	1	0
31:24	RESERVED							
23:16	RESERVED	ABORT	RUN	CS[4:0]				
15:8	RESERVED							
7:0	RESERVED							

Bits	Name	Description	Type	Reset value
31:23	RESERVED	RESERVED	R	0x0
22	ABORT	1 = Tsi721 aborted processing inbound messages for this inbound DMA channel due to encountering a PCIe error condition.	R/W	0x0
21	RUN	0 = The DMA channel is uninitialized and Tsi721 has suspended the channel after software set the SUSPEND bit of Inbound DMA Channel Control Register {0..7}. 1 = Tsi721 is in the middle of processing descriptors for this DMA channel. During inbound DMA channel initialization, the RUN bit is set to 1 when software writes to the Inbound Circular Free Queue Write Pointer Register {0..7}.	R	0x0
20:16	CS	Completion Error status for this DMA channel: • 01000 = PCIe free pointer cpl/clpD timeout occurred • 01001 = PCIe free pointer poisoned cpl/clpD received • 01011 = PCIe free pointer cpl/clpD received is malformed • 01100 = PCIe free pointer cpl/clpD received has completion status of UR • 01101 = PCIe free pointer cpl/clpD received has completion status of CA • Others = Reserved	R/WS	0x0
15:0	RESERVED	RESERVED	R	0x0

20.2.26 Inbound DMA Channel Interrupt Register {0..7}

This register defines the inbound DMA channel interrupt function. Inbound DMA channel Y (Y = 0–7) is assigned to register Y.

Register name: IBDMAC{0..7}INT Reset value: 0x0000_0000	Register offset: 0x61248 += 1000

Bits	7	6	5	4	3	2	1	0
31:24	RESERVED							
23:16	RESERVED							
15:8	RESERVED			SRTO	RESERVED			
7:0	RESERVED	RESERVED	RESERVED	RESERVED	SUSPENDED	PC_ERROR	FQ_LOW	DQ_RCV

Bits	Name	Description	Type	Reset value
31:13	RESERVED	RESERVED	R	0x0
12	SRTO	S-RIO Message request timeout 1 = A message request timeout occurred at least once to the inbound DMA channel since this bit was last cleared.	R/W1CS	0x0
11:4	RESERVED	RESERVED	R	0x0
3	SUSPENDED	1 = Tsi721 suspended the processing of descriptors for this DMA channel – triggered by SUSPEND bit of Inbound DMA Channel Control Register {0..7} – at least once since this bit was last cleared.	R/W1CS	0x0
2	PC_ERROR	1 = Tsi721 detected one or more PCIe errors for this DMA channel since this bit was last cleared.	R/W1CS	0x0
1	FQ_LOW	Inbound circular free queue low 1 = Hardware own level of inbound free queue has fallen at least once below the threshold defined by Inbound Circular Free Queue Threshold Register {0..7} since this bit was last cleared.	R/W1CS	0x0
0	DQ_RCV	Inbound message received interrupt 1 = At least one inbound message was sent since this bit was last cleared.	R/W1CS	0x0

20.2.27 Inbound DMA Channel Interrupt Set Register {0..7}

This register defines the debug function that sets the inbound DMA channel interrupt. Inbound DMA channel Y (Y = 0–7) is assigned to register Y.

Register name: IBDMAC{0..7}INTSET Reset value: 0x0000_0000	Register offset: 0x6124C += 1000

Bits	7	6	5	4	3	2	1	0
31:24	RESERVED							
23:16	RESERVED							
15:8	RESERVED			SRTO_SET	RESERVED			
7:0	RESERVED	RESERVED	RESERVED	RESERVED	SUSPENDED_SET	PC_ERROR_SET	FQ_LOW_SET	DQ_RCV_SET

Bits	Name	Description	Type	Reset value
31:13	RESERVED	RESERVED	R	0x0
12	SRTO_SET	S-RIO Message request timeout interrupt set 1 = Sets the SRTO bit of Inbound DMA Channel Interrupt Register {0..7} to 1.	R/W1S	0x0
11:4	RESERVED	RESERVED	R	0x0
3	SUSPENDED_SET	1 = Sets the SUSPEND bit of Inbound DMA Channel Interrupt Register {0..7} to 1.	R/W1S	0x0
2	PC_ERROR_SET	1 = Sets the PC_ERROR bit of Inbound DMA Channel Interrupt Register {0..7} to 1.	R/W1S	0x0
1	FQ_LOW_SET	Inbound circular free queue low interrupt set 1 = Sets the FQ_LOW bit of Inbound DMA Channel Interrupt Register {0..7} to 1.	R/W1S	0x0
0	DQ_RCV_SET	Inbound circular descriptor queue almost full interrupt set 1 = Sets the DQ_RCV bit of Inbound DMA Channel Interrupt Register {0..7} to 1.	R/W1S	0x0

20.2.28 Inbound DMA Channel Interrupt Enable Register {0..7}

This register defines the inbound DMA channel interrupt enable function. Inbound DMA channel Y (Y = 0–7) is assigned to register Y.

Register name: IBDMAC{0..7}INTE Reset value: 0x0000_0001	Register offset: 0x61250 += 1000

Bits	7	6	5	4	3	2	1	0
31:24	RESERVED							
23:16	RESERVED							
15:8	RESERVED			SRTO_EN	RESERVED			
7:0	RESERVED	RESERVED	RESERVED	RESERVED	SUSPENDED_EN	PC_ERROR_EN	FQ_LOW_EN	DQ_RCV_EN

Bits	Name	Description	Type	Reset value
31:13	RESERVED	RESERVED	R	0x0
12	SRTO_EN	S-RIO Message request timeout interrupt enable 1 = Enable interrupt generation for SRTO bit of Inbound DMA Channel Interrupt Register {0..7}.	R/WS	0x0
11:4	RESERVED	RESERVED	R	0x0
3	SUSPENDED_EN	1 = Enable interrupt generation for SUSPENDED bit of Inbound DMA Channel Interrupt Register {0..7}.	R/WS	0x0
2	PC_ERROR_EN	1 = Enable interrupt generation for PC_ERROR bit of Inbound DMA Channel Interrupt Register {0..7}.	R/WS	0x0
1	FQ_LOW_EN	Inbound circular free queue low interrupt enable 1 = Enable interrupt generation for FQ_LOW bit of Inbound DMA Channel Interrupt Register {0..7}.	R/WS	0x0
0	DQ_RCV_EN	Inbound circular descriptor queue almost full interrupt enable 1 = Enable interrupt generation for DQ_RCV bit of Inbound DMA Channel Interrupt Register {0..7}. When MSI-X is used, this bit must be set to 1.	R/WS	0x1

20.2.29 Inbound DMA Channel Port-Write Enable Register {0..7}

This register defines the inbound DMA channel port-write enable function. Inbound DMA channel Y (Y = 0–7) is assigned to register Y.

Register name: IBDMAC{0..7}PWE Reset value: 0x0000_0000	Register offset: 0x61254 += 1000

Bits	7	6	5	4	3	2	1	0
31:24	RESERVED							
23:16	RESERVED							
15:8	RESERVED			SRTO_EN	RESERVED			
7:0	RESERVED							

Bits	Name	Description	Type	Reset value
31:13	RESERVED	RESERVED	R	0x0
12	SRTO_EN	S-RIO Message request timeout port-write enable 1 = Enable S-RIO port-write generation for SRTO bit of Inbound DMA Channel Interrupt Register {0..7}.	R/WS	0x0
11:0	RESERVED	RESERVED	R	0x0

20.2.30 Inbound Descriptor Queue Lower Base Register {0..7}

This register defines the lower 26 bits of the base address of the inbound circular descriptor queue. Rx queue Y (Y = 0–7) is assigned to register Y.

Register name: IBDMAC{0..7}DQBL Reset value: 0x0000_0000	Register offset: 0x61300 += 1000

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

Bits	Name	Description	Type	Reset value
31:6	ADD	Bits 31:8 of inbound descriptor queue base address. The base address must be 64-byte aligned.	R/WS	0x0
5:0	Reserved	RESERVED	R	0x0

20.2.31 Inbound Descriptor Queue Upper Base Register {0..7}

This register defines the upper 32 bits of the base address of the inbound circular descriptor queue. Rx queue Y (Y = 0–7) is assigned to register Y.

Register name: IBDMAC{0..7}DQBH Reset value: 0x0000_0000	Register offset: 0x61304 += 1000

Bits	7	6	5	4	3	2	1	0
31:24	ADD[63:56]							
23:16	ADD[55:48]							
15:8	ADD[47:40]							
7:0	ADD[39:8]							

Bits	Name	Description	Type	Reset value
31:0	ADD	32 MSB bits of outbound descriptor queue base address.	R/WS	0x0

20.2.32 Inbound Circular Descriptor Queue Read Pointer Register {0..7}

This register defines the read pointer of the inbound circular descriptor queue. Rx queue Y (Y = 0–7) is assigned to register Y.

Register name: IBDMAC{0..7}DQRP Reset value: 0x0000_0000	Register offset: 0x61308 += 1000

Bits	7	6	5	4	3	2	1	0
31:24	RESERVED							
23:16	RESERVED					RD_PTR[18:16]		
15:8	RD_PTR[15:8]							
7:0	RD_PTR[7:0]							

Bits	Name	Description	Type	Reset value
31:19	RESERVED	RESERVED	R	0x0
18:0	RD_PTR	Read pointer to an entry of inbound descriptor queue relative to inbound descriptor queue base address. Under normal operations, software updates this register. When a message is completed, the Tsi721 issue descriptor MWr to the inbound descriptor queue when RD_PTR is not equal to the WR_PTR of Inbound Circular Descriptor Queue Write Pointer Register {0..7}; that is, the Tsi721 sees the queue full when (WR_PTR+1) MOD Inbound Circular Descriptor Queue Size Register {0..7} = RD_PTR.	R/W	0x0

20.2.33 Inbound Circular Descriptor Queue Write Pointer Register {0..7}

This register defines the write pointer of the inbound circular descriptor queue. Rx queue Y (Y = 0–7) is assigned register Y.

Register name: IBDMAC{0..7}DQWP Reset value: 0x0000_0000	Register offset: 0x6130C += 1000

Bits	7	6	5	4	3	2	1	0
31:24	RESERVED							
23:16	RESERVED					WR_PTR[18:16]		
15:8	WR_PTR[15:8]							
7:0	WR_PTR[7:0]							

Bits	Name	Description	Type	Reset value
31:19	RESERVED	RESERVED	R	0x0
18:0	WR_PTR	Write pointer to an entry of inbound descriptor queue relative to inbound descriptor queue base address. Software should use Hardware own (HO) bit (see Figure 37) of inbound message descriptors instead of WR_PTR to decide valid entries within the descriptor queue.	R/W	0x0

20.2.34 Inbound Circular Descriptor Queue Size Register {0..7}

This register defines the size of the inbound circular descriptor queue. Rx queue Y (Y = 0–7) is assigned to register Y.

Register name: IBDMAC{0..7}DQSZ Reset value: 0x0000_0005	Register offset: 0x61314 += 1000

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

Bits	Name	Description	Type	Reset value
31:4	RESERVED	RESERVED	R	0x0
3:0	SIZE	0b0000 = Reserved 0b0001 = 32 entries 0b0010 = 64 entries 0b0011 = 128 entries 0b0100 = 256 entries 0b0101 = 512 entries 0b0110 = 1K entries 0b0111 = 2K entries 0b1000 = 4K entries 0b1001 = 8K entries 0b1010 = 16K entries 0b1011 = 32K entries 0b1100 = 64K entries 0b1101–0b1111 = Reserved	R/WS	0x5

20.2.35 Messaging Engine Interrupt Enable Register

This register contains the bits that control if an error condition detected by the Messaging Engine Interrupt Register will generate an interrupt.

Register Name: SMSG_INTE Reset Value: 0x0000_0000	Register Offset: 0x6A000

Bits	7	6	5	4	3	2	1	0
31:24	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
23:16	UNS_RSP_EN	Reserved				ECC_UNCORR_EN	ECC_CORR_EN	RESERVED
15:8	ECC_UNCORR_CH_EN[7:0]							
7:0	ECC_CORR_CH_EN[7:0]							

Bits	Name	Description	Type	Reset Value
31:24	Reserved	Reserved	R	0
23	UNS_RSP_EN	1 = Enable UNS_RSP interrupt of Messaging Engine Interrupt Register..	R/WS	0
22:19	RESERVED	RESERVED	R	0x0
18	ECC_UNCORR_EN	Uncorrectable ECC Error interrupt enable for non-channelized memory instances 1 = Enable ECC_UNCORR of Messaging Engine Interrupt Register.	R/WS	0x0
17	ECC_CORR_EN	Correctable ECC Error interrupt enable for non-channelized memory instances 1 = Enable ECC_CORR of Messaging Engine Interrupt Register.	R/WS	0x0
16	RESERVED	RESERVED	R	0x0
15:8	ECC_UNCORR_CH_EN[7:0]	Uncorrectable ECC Error interrupt enable for channelized memory instances bit x = 1: Enable ECC_UNCORR_CH[x] of Messaging Engine Interrupt Register.	R/WS	0x0
7:0	ECC_CORR_CH_EN[7:0]	Correctable ECC Error interrupt enable for channelized memory instances bit x = 1: Enable ECC_CORR_CH[x] of Messaging Engine Interrupt Register.	R/WS	0x0

20.2.36 Messaging Engine Port-Write Enable Register

This register contains the bits that control if an error condition detected by the Messaging Engine Interrupt Register will generate S-RIO port-writes.

Register Name: SMSG_PWE Reset Value: 0x0000_0000	Register Offset: 0x6A004

Bits	7	6	5	4	3	2	1	0
31:24	Reserved							
23:16	UNS_RSP_EN	Reserved						
15:8	IBDMA_PW_EN[7:0]							
7:0	OBDMA_PW_EN[7:0]							

Bits	Name	Description	Type	Reset Value
31:24	Reserved	Reserved	R	0x0
23	UNS_RSP_EN	Unsolicited/Unexpected response packet port-write generation enable.	R/WS	0
22:16	Reserved	Reserved	R	0x0
15:8	IBDMA_PW_EN[7:0]	Inbound message DMA engine S-RIO error port-write enable When IBDMA_PW_EN[x] (x = 0–7) is set to 1, it enables IBDMA_PW[x] bit of the Messaging Engine Port-Write Register to generate S-RIO port-writes.	R/WS	0x00
7:0	OBDMA_PW_EN[7:0]	Outbound message DMA engine S-RIO error port-write enable When OBDMA_PW_EN[x] (x = 0–7) is set to 1, it enables OBDMA_PW[x] bit of the Messaging Engine Port-Write Register to generate S-RIO port-writes.	R/WS	0x00

20.2.37 Messaging Engine Interrupt Register

Register Name: SMSG_INT Reset Value: 0x0000_0000	Register Offset: 0x6A008

Bits	7	6	5	4	3	2	1	0
31:24	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
23:16	UNS_RSP	Reserved				ECC_UNCORR	ECC_CORR	RESERVED
15:8	ECC_UNCORR_CH[7:0]							
7:0	ECC_CORR_CH[7:0]							

Bits	Name	Description	Type	Reset Value
31:24	Reserved	Reserved	R	0
23	UNS_RSP	1 = An Unsolicited/Unexpected response packet was received.	R/W1CS	0
22:19	Reserved	Reserved	R	0x0
18	ECC_UNCORR	Uncorrectable ECC interrupt for non-channelized memory instances 1 = An uncorrectable ECC error occurred. It is enabled by ECC_UNCORR_EN of Messaging Engine Interrupt Enable Register.	R/W1CS	0x0
17	ECC_CORR	Correctable ECC interrupt for non-channelized memory instances 1 = A correctable ECC error occurred. It is enabled by ECC_CORR_EN of Messaging Engine Interrupt Enable Register.	R/W1CS	0x0
16	Reserved	Reserved	R	0x0
15:8	ECC_UNCORR_CH[7:0]	Uncorrectable ECC interrupt for channelized memory instances bit x = 1: An uncorrectable ECC error occurred for one of channel x's memory instances. It is enabled by ECC_UNCORR_CH_EN of Messaging Engine Interrupt Enable Register.	R/W1CS	0x0
7:0	ECC_CORR_CH[7:0]	Correctable ECC interrupt for channelized memory instances bit x = 1: A correctable ECC error occurred for one of channel x's memory instances. It is enabled by ECC_CORR_CH_EN of Messaging Engine Interrupt Enable Register.	R/W1CS	0x0

20.2.38 Messaging Engine Port-Write Register

Register Name: SMSG_PW Reset Value: 0x0000_0000	Register Offset: 0x6A00C

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

Bits	Name	Description	Type	Reset Value
31:16	Reserved	Reserved	R	0x0
15:8	IBDMA_PW[7:0]	Inbound message DMA engine S-RIO error port-write When IBDMA_PW[x] (x = 0–7) is set to 1, an S-RIO error has occurred to inbound message DMA channel x as indicated by S-RIO errors in the Inbound DMA Channel Interrupt Register {0..7}. It is enabled by IBDMA_PW_EN[x] in the Messaging Engine Port-Write Enable Register to generate S-RIO port-writes.	RS	0x0
7:0	OBDMA_PW[7:0]	Outbound message DMA engine S-RIO error port-write When OBDMA_PW[x] (x = 0–7) is set to 1, an S-RIO error has occurred to outbound message DMA channel x as indicated by S-RIO errors in the Outbound DMA Channel Status Register {0..7}. It is enabled by OBDMA_PW_EN[x] in the Messaging Engine Port-Write Enable Register to generate S-RIO port-writes.	RS	0x0

20.2.39 Messaging Engine Interrupt Set Register

This register defines the Messaging Engine interrupt options set for software debug.

Register Name: SMSG_INTSET Reset Value: 0x0000_0000	Register Offset: 0x6A010

Bits	7	6	5	4	3	2	1	0
31:24	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
23:16	UNS_RSP_SET	Reserved				ECC_UNCORR_SET	ECC_CORR_SET	Reserved
15:8	ECC_UNCORR_CH_SET[7:0]							
7:0	ECC_CORR_CH_SET[7:0]							

Bits	Name	Description	Type	Reset Value
31:24	Reserved	Reserved	R	0
23	UNS_RSP_SET	1 = Set the UNS_RSP_ERR bit of Messaging Engine Interrupt Register to 1.	R/W1S	0
22:19	Reserved	Reserved	R	0
18	ECC_UNCORR_SET	Uncorrectable ECC Error Interrupt Set for non-channelized memory instances 1 = Force uncorrectable ECC error interrupt ECC_UNCORR of Messaging Engine Interrupt Register to 1.	R/W1S	0x0
17	ECC_CORR_SET	Correctable ECC Error interrupt set for non-channelized memory instances 1 = Force correctable ECC error interrupt ECC_CORR of Messaging Engine Interrupt Register to 1.	R/W1S	0x0
16	RESERVED	RESERVED	R	0x0
15:8	ECC_UNCORR_CH_SET[7:0]	Uncorrectable ECC Error Interrupt Set for channelized memory instances 1 = Force uncorrectable ECC error interrupt ECC_UNCORR_CH[x] of Messaging Engine Interrupt Register to 1.	R/W1S	0x0
7:0	ECC_CORR_CH_SET[7:0]	Correctable ECC Error interrupt set for channelized memory instances 1 = Force correctable ECC error interrupt ECC_CORR_CH[x] of Messaging Engine Interrupt Register to 1.	R/W1S	0x0

20.2.40 Messaging Engine non-channelized ECC LOG Register

This register defines the SMSG non-channelized ECC log.

Register name: SMSG_ECC_LOG Reset value: 0x0000_0000	Register offset: 0x6A014

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

Bits	Name	Description	Type	Reset value
31:19	RESERVED	RESERVED	R	0x0
18:16	ECC_UNCORR_MEM[2:0]	When bit x is set to 1, an uncorrectable ECC error has occurred to non-channelized SRAM instance x since it was cleared last time.	R/W1CS	0x0
15:3	RESERVED	RESERVED	R	0x0
2:0	ECC_CORR_MEM[2:0]	When bit x is set to 1, a correctable ECC error has occurred to non-channelized SRAM instance x since it was cleared last time.	R/W1CS	0x0

20.2.41 Generated Message Segment Retry Count Register

This register defines the number of received message segments retried by Tsi721.

Register name: RETRY_GEN_CNT Reset value: 0x0000_0000	Register offset: 0x6A100

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

Bits	Name	Description	Type	Reset value
31:0	RETRY_GEN_CNT	Generated Retry Message Segment Count It is a saturating counter that counts the number of received message segments retried by Tsi721.	RC	0x0

20.2.42 Received Retry Message Response Count Register

This register defines the number of received message responses with retry status by Tsi721.

Register name: RETRY_RX_CNT Reset value: 0x0000_0000	Register offset: 0x6A104

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

Bits	Name	Description	Type	Reset value
31:0	RETRY_RX_CNT	Received Message Response Count It is a saturating counter that counts the number of received message segment responses with retry status by Tsi721.	RC	0x0

20.2.43 Messaging Engine Channelized Correctable ECC LOG Register {0..7}

This register defines the SMSG channelized correctable ECC log. Outbound DMA channel Y (Y = 0–7) is assigned to register Y.

Register name: SMSG_ECC_CORR{0..7}LOG Reset value: 0x0000_0000	Register offset: 0x6A300 += 4

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

Bits	Name	Description	Type	Reset value
31:8	RESERVED	RESERVED	R	0x0
7:0	ECC_CORR_MEM[7:0]	When bit x is set to 1, a correctable ECC error occurred to SRAM instance x of this channel since it was last cleared.	R/W1CS	0x0

20.2.44 Messaging Engine Channelized Uncorrectable ECC LOG Register {0..7}

This register defines the SMSG channelized uncorrectable ECC log. Outbound DMA channel Y (Y = 0–7) is assigned to register Y.

Register name: SMSG_ECC_UNCORR{0..7}LOG Reset value: 0x0000_0000	Register offset: 0x6A340 += 4

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

Bits	Name	Description	Type	Reset value
31:8	RESERVED	RESERVED	R	0x0
7:0	ECC_UNCORR_MEM	When bit x is set to 1, an uncorrectable ECC error occurred to SRAM instance x of this channel since it was last cleared.	R/W1CS	0x0

21. Block DMA Engine Registers

Topics discussed include the following:

- Register Map
- Register Descriptions

21.1 Register Map

For registers defined for multiple DMA channels such as DMA Descriptor Write Count Register {0..7}, register offsets between two adjacent channels is 4 KB; that is, in 0x51000 += 1000, register for channel 0 has address 0x51000, += denotes the register offset between channels and 1000 is in hexidecimal.

Table 96: Block DMA Engine Register Map

Offset	Register Name	See
0x51000 += 1000	DMAC{0..7}DWRCNT	DMA Descriptor Write Count Register {0..7}
0x51004 += 1000	DMAC{0..7}DRDCNT	DMA Descriptor Read Count Register {0..7}
0x51008 += 1000	DMAC{0..7}CTL	DMA Channel Control Register {0..7}
0x5100C += 1000	DMAC{0..7}INT	DMA Channel Interrupt Register {0..7}
0x51010 += 1000	DMAC{0..7}INTSET	DMA Channel Interrupt Set Register {0..7}
0x51014 += 1000	DMAC{0..7}STS	DMA Channel Status Register {0..7}
0x51018 += 1000	DMAC{0..7}INTE	DMA Channel Interrupt Enable Register {0..7}
0x51024 += 1000	DMAC{0..7}DPTRL	DMA Channel Descriptor Pointer Low Register {0..7}
0x51028 += 1000	DMAC{0..7}DPTRH	DMA Channel Descriptor Pointer High Register {0..7}
0x5102C += 1000	DMAC{0..7}DSBL	DMA Descriptor Status FIFO Lower Base Register {0..7}
0x51030 += 1000	DMAC{0..7}DSBH	DMA Descriptor Status FIFO Upper Base Register {0..7}
0x51034 += 1000	DMAC{0..7}DSSZ	DMA Descriptor Status FIFO Size Register {0..7}
0x51038 += 1000	DMAC{0..7}DSRP	DMA Descriptor Status FIFO Read Pointer Register {0..7}
0x5103C += 1000	DMAC{0..7}DSWP	DMA Descriptor Status FIFO Write Pointer Register {0..7}
0x5F000	BDMA_INTE	BDMA Interrupt Enable CSR
0x5F004	BDMA_INT	BDMA Interrupt CSR
0x5F008	BDMA_INTSET	BDMA Interrupt Set Register

Table 96: Block DMA Engine Register Map *(Continued)*

Offset	Register Name	See
0x5F00C	BDMA_ECC_LOG	BDMA ECC Log Register
0x5F300 += 4	BDMA_ECC_CORR{0..7}LOG	BDMA Channelized Correctable ECC LOG Register {0..7}
0x5F340 += 4	BDMA_ECC_UNCORR{0..7}LOG	BDMA Channelized Uncorrectable ECC LOG Register {0..7}

21.2 Register Descriptions

21.2.1 DMA Descriptor Write Count Register {0..7}

This register defines the DMA descriptor write count function. Register Y (Y = 0–7) is assigned to DMA channel Y.

Register name: DMAC{0..7}DWRCNT Reset value: 0x0000_0000	Register offset: 0x51000 += 1000

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

Bits	Name	Description	Type	Reset value
31:0	DWRCNT	32-bit descriptor write count. Software must update this register in a timely manner in order not to stall Tsi721 descriptor processing. Under normal conditions, Tsi721 continues descriptor prefetching until DRDCNT of DMA Descriptor Read Count Register {0..7} equals DWRCNT. If the RUN bit of DMA Channel Status Register {0..7} and the ERROR bit of DMA Channel Interrupt Register {0..7} are both 0 when software writes to this register, Tsi721 starts descriptor processing for this DMA channel. Tsi721 resets this field when INIT bit in DMA Channel Control Register {0..7} toggles from 0 to 1.	R/W	0x0

21.2.2 DMA Descriptor Read Count Register {0..7}

This register defines the DMA descriptor read count function. Register Y (Y = 0–7) is assigned to DMA channel Y.

Register name: DMAC{0..7}DRDCNT Reset value: 0x0000_0000	Register offset: 0x51004 += 1000

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

Bits	Name	Description	Type	Reset value
31:0	DRDCNT	32-bit descriptor read count. The Tsi721 updates this field after a descriptor prefetch. It resets the field when INIT bit in DMA Channel Control Register {0..7} toggles from 0 to 1.	R/W	0x0

21.2.3 DMA Channel Control Register {0..7}

This register defines DMA channel control function. DMA channel Y (Y = 0–7) is assigned registers Y.

Register name: DMAC{0..7}CTL Reset value: 0x0000_0000	Register offset: 0x51008 += 1000

Bits	7	6	5	4	3	2	1	0
31:24	RESERVED							
23:16	RESERVED							
15:8	RESERVED							
7:0	RESERVED						SUSPEND	INIT

Bits	Name	Description	Type	Reset value
31:2	RESERVED	RESERVED	R	0x0
1	SUSPEND	DMA channel suspend. 1 = Tsi721 suspends DMA descriptor processing for this DMA channel. Tsi721 automatically clears this bit to 0 after it finishes the suspend procedure (SUSPENDED bit in DMA Channel Interrupt Register {0..7} is set to 1). Tsi721 resumes DMA descriptor processing for this DMA channel when it detects a write to the DMA Descriptor Write Count Register {0..7} after this bit is cleared to 0. Note: Software must not set this bit to 1 while the INIT bit is 1; otherwise, device behavior will be undefined.	R/W	0x0
0	INIT	DMA channel initialization. 1 = Tsi721 resets the DMA channel, including internal logic and DMA Descriptor Read Count Register {0..7}, DMA Descriptor Write Count Register {0..7}, and DMA Channel Status Register {0..7} for this channel. The DMA Channel Status Register {0..7} registers can only be accessed at least 40 ns after Tsi721 has cleared INIT from 1 to 0. Software can set INIT to 1 only when this DMA channel is idle; that is, DMA Channel Status Register {0..7} RUN bit is 0. Tsi721 resets this bit when software writes to the DMA Descriptor Write Count Register {0..7}.	R/W	0x0

21.2.4 DMA Channel Interrupt Register {0..7}

This register defines the DMA channel interrupt function. DMA channel Y (Y = 0–7) is assigned registers Y.

Register name: DMAC{0..7}INT Reset value: 0x0000_0000	Register offset: 0x5100C += 1000

Bits	7	6	5	4	3	2	1	0
31:24	RESERVED							
23:16	RESERVED							
15:8	RESERVED							
7:0	RESERVED	RESERVED	RESERVED	ST_FULL	DONE	SUSPENDED	ERROR	IOF_DONE

Bits	Name	Description	Type	Reset value
31:5	RESERVED	RESERVED	R	0x0
4	ST_FULL	1 = Tsi721 detected that the descriptor status FIFO in the PCIe side main memory for this DMA channel has become full at least once since this bit was last cleared.	R/W1CS	0x0
3	DONE	1 = Tsi721 reached the end of the descriptor list without errors (finished processing all descriptors within the descriptor list of this DMA channel without any errors) at least once since this bit was last cleared.	R/W1CS	0x0
2	SUSPENDED	1 = Tsi721 suspended the processing of descriptors for this DMA channel – triggered by SUSPEND bit of DMA Channel Control Register {0..7} – at least once since this bit was last cleared.	R/W1CS	0x0
1	ERROR	1 = Tsi721 detected a PCIe or S-RIO error for this DMA channel at least once since this bit was last cleared. For information about the specific error, see CS in DMA Channel Status Register {0..7}.	R/W1CS	0x0
0	IOF_DONE	1 = Tsi721 finished processing a descriptor with the IOF bit set for this DMA channel at least once since this bit was last cleared.	R/W1CS	0x0

21.2.5 DMA Channel Interrupt Set Register {0..7}

This register defines the DMA channel interrupt debug function that sets individual interrupt bits. DMA channel Y (Y = 0–7) is assigned registers Y.

Register name: DMAC{0..7}INTSET Reset value: 0x0000_0000	Register offset: 0x51010 += 1000

Bits	7	6	5	4	3	2	1	0
31:24	RESERVED							
23:16	RESERVED							
15:8	RESERVED							
7:0	RESERVED	RESERVED	RESERVED	ST_FULL_SET	DONE_SET	SUSPENDED_SET	ERROR_SET	IOF_DONE_SET

Bits	Name	Description	Type	Reset value
31:5	RESERVED	RESERVED	R	0x0
4	ST_FULL_SET	1 = Set the ST_FULL bit in the DMA Channel Interrupt Register {0..7} to 1.	R/W1S	0x0
3	DONE_SET	1 = Set the DONE bit in the DMA Channel Interrupt Register {0..7} to 1.	R/W1S	0x0
2	SUSPENDED_SET	1 = Set the SUSPENDED in the DMA Channel Interrupt Register {0..7} to 1.	R/W1S	0x0
1	ERROR_SET	1 = Set the ERROR bit in the DMA Channel Interrupt Register {0..7} to 1.	R/W1S	0x0
0	IOF_DONE_SET	1 = Set the IOF_DONE bit in the DMA Channel Interrupt Register {0..7} to 1.	R/W1S	0x0

21.2.6 DMA Channel Status Register {0..7}

This register defines the DMA channel status function. DMA channel Y (Y = 0–7) is assigned registers Y.

Register name: DMAC{0..7}STS Reset value: 0x0000_0000	Register offset: 0x51014 += 1000

Bits	7	6	5	4	3	2	1	0
31:24	RESERVED							
23:16	RESERVED	ABORT	RUN	CS[4:0]				
15:8	RESERVED							
7:0	RESERVED							

Bits	Name	Description	Type	Reset value
31:23	RESERVED	RESERVED	R	0x0
22	ABORT	1 = Tsi721 aborted processing descriptors for this DMA channel due to an error condition. Software should write this bit only during software debug.	R/W	0x0
21	RUN	0 = The DMA channel is aborted or suspended or has processed the last descriptor of its descriptor list. 1 = Tsi721 is processing descriptors for this DMA channel. During DMA channel initialization, the RUN bit is set to 1 when software writes to DMA Descriptor Write Count Register {0..7}.	R	0x0

(Continued)

Bits	Name	Description	Type	Reset value
20:16	CS	Error completion status for this DMA channel when ERROR bit in the DMA Channel Interrupt Register {0..7} is 1: • 00101 = S-RIO response timeout occurred • 00110 = S-RIO I/O ERROR response received • 00111 = S-RIO implementation specific error occurred, that is, response payload size mismatches expected from request's wdptr/ rdsize • 01000 = PCIe cpl/clpD timeout occurred • 01001 = PCIe poisoned cpl/clpD received • 01011 = PCIe cpl/clpD received is malformed • 01100 = PCIe cpl/clpD received has completion status of UR • 01101 = PCIe cpl/clpD received has completion status of CA • Others = Reserved Software should write this bit only during software debug.	R/WS	0x0
15:0	RESERVED	RESERVED	R	0x0

21.2.7 DMA Channel Interrupt Enable Register {0..7}

This register defines the DMA channel interrupt enable function. DMA channel Y (Y = 0–7) is assigned registers Y.

Register name: DMAC{0..7}INTE Reset value: 0x0000_0001	Register offset: 0x51018 += 1000

Bits	7	6	5	4	3	2	1	0
31:24	RESERVED							
23:16	RESERVED							
15:8	RESERVED							
7:0	RESERVED	RESERVED	RESERVED	ST_FULL_EN	DONE_EN	SUSPENDED_EN	ERROR_EN	IOF_DONE_EN

Bits	Name	Description	Type	Reset value
31:5	RESERVED	RESERVED	R	0x0
4	ST_FULL_EN	1 = Enable interrupt generation for ST_FULL bit of DMA Channel Interrupt Register {0..7}.	R/WS	0x0
3	DONE_EN	1 = Enable interrupt generation for DONE bit of DMA Channel Interrupt Register {0..7}.	R/WS	0x0
2	SUSPENDED_EN	1 = Enable interrupt generation for SUSPENDED bit of DMA Channel Interrupt Register {0..7}.	R/WS	0x0
1	ERROR_EN	1 = Enable interrupt generation for ERROR bit of DMA Channel Interrupt Register {0..7}.	R/WS	0x0
0	IOF_DONE_EN	1 = Enable interrupt generation for IOF_DONE bit of DMA Channel Interrupt Register {0..7}. When using MSI-X, this bit must be set to 1.	R/WS	0x1

21.2.8 DMA Channel Descriptor Pointer Low Register {0..7}

This register defines the lower 32 bits of the DMA channel descriptor pointer. Register Y (Y = 0–7) is assigned to DMA channel Y.

Register name: DMAC{0..7}DPTRL Reset value: 0x0000_0000	Register offset: 0x51024 += 1000

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

Bits	Name	Description	Type	Reset value
31:5	DPTRL	Descriptor Pointer Low This field defines bits 31:5 of the 32-byte aligned descriptor pointer for this DMA channel. Together with DPTRH in DMA Channel Descriptor Pointer High Register {0..7}, they define the 64-bit DMA channel descriptor pointer to the next descriptor to be processed. Software can read/write DPTRL and DPTRH only when the RUN bit in DMA Channel Status Register {0..7} is 0; otherwise, since the Tsi721 is constantly updating DPTRL and DPTRH while software can only read DPTRL or DPTRH one at a time, software will read inconsistent values. When the INIT bit in DMA Channel Interrupt Register {0..7} toggles from 0 to 1, the descriptor pointer points to the first descriptor of the descriptor list. When the ERROR bit in DMA Channel Interrupt Register {0..7} is set to 1, the descriptor pointer is undefined; otherwise, the descriptor pointer points to the next descriptor to be fetched.	R/WS	0x0
4:0	RESERVED	RESERVED	R	0x0

21.2.9 DMA Channel Descriptor Pointer High Register {0..7}

This register defines the upper 32 bits of the DMA channel descriptor pointer. Register Y (Y = 0–7) is assigned to DMA channel Y.

Register name: DMAC{0..7}DPTRH Reset value: 0x0000_0000	Register offset: 0x51028 += 1000

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

Bits	Name	Description	Type	Reset value
31:0	DPTRH	Descriptor Pointer High This field defines the upper 32 bits of the descriptor pointer for this DMA channel (see DMA Channel Descriptor Pointer Low Register {0..7}. Software should read/write DPTRL and DPTRH only when the RUN bit is 0.	R/WS	0x0

21.2.10 DMA Descriptor Status FIFO Lower Base Register {0..7}

This register defines the lower 32 bits of the base address of a DMA descriptor status FIFO. DMA channel Y (Y = 0–7) is assigned registers Y.

Register name: DMAC{0..7}DSBL Reset value: 0x0000_0000	Register offset: 0x5102C += 1000

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

Bits	Name	Description	Type	Reset value
31:6	ADD	26 LSB bits of DMA descriptor status base address. The base address must be 64-byte aligned.	R/WS	0x0
5:0	Reserved	RESERVED	R	0x0

21.2.11 DMA Descriptor Status FIFO Upper Base Register {0..7}

This register defines the upper 32 bits of the base address of a DMA descriptor status FIFO. DMA channel Y (Y = 0–7) is assigned registers Y.

Register name: DMAC{0..7}DSBH Reset value: 0x0000_0000	Register offset: 0x51030 += 1000

Bits	7	6	5	4	3	2	1	0
31:24	ADD[63:56]							
23:16	ADD[55:48]							
15:8	ADD[47:40]							
7:0	ADD[39:32]							

Bits	Name	Description	Type	Reset value
31:0	ADD	32 MSB bits of descriptor status FIFO base address.	R/WS	0x0

21.2.12 DMA Descriptor Status FIFO Size Register {0..7}

This register defines the size of a DMA descriptor status FIFO. DMA channel Y (Y = 0–7) is assigned registers Y.

Register name: DMAC{0..7}DSSZ Reset value: 0x0000_0005	Register offset: 0x51034 += 1000

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

Bits	Name	Description	Type	Reset value
31:4	RESERVED	RESERVED	R	0x0
3:0	SIZE	SIZE 0b0000 = Reserved 0b0001 = 32 entries 0b0010 = 64 entries 0b0011 = 128 entries 0b0100 = 256 entries 0b0101 = 512 entries 0b0110 = 1K entries 0b0111 = 2K entries 0b1000 = 4K entries 0b1001 = 8K entries 0b1010 = 16K entries 0b1011 = 32K entries 0b1100 = 64K entries 0b1101 = 128K entries 0b1110 = 256K entries 0b1111 = 512K entries	R/WS	0x5

21.2.13 DMA Descriptor Status FIFO Read Pointer Register {0..7}

This register defines the read pointer of the DMA descriptor status FIFO. DMA channel Y (Y = 0–7) is assigned registers Y.

Register name: DMAC{0..7}DSRP Reset value: 0x0000_0000	Register offset: 0x51038 += 1000

Bits	7	6	5	4	3	2	1	0
31:24	RESERVED							
23:16	RESERVED					RD_PTR[18:16]		
15:8	RD_PTR[15:8]							
7:0	RD_PTR[7:0]							

Bits	Name	Description	Type	Reset value
31:19	RESERVED	RESERVED	R	0x0
18:0	RD_PTR	Read pointer to an entry of the descriptor status FIFO relative to the descriptor status FIFO base address. Software must update this register in a timely manner in order not to stall Tsi721 descriptor processing. The device resets this register upon a DMA channel initialization (see INIT in DMA Channel Interrupt Register {0..7}). The device launches descriptor processing only if WR_PTR in DMA Descriptor Status FIFO Write Pointer Register {0..7} does not equal RD_PTR; that is, it sees the queue full when (WR_PTR+1) MOD DMA Descriptor Status FIFO Size Register {0..7} =RD_PTR.	R/W	0x0

21.2.14 DMA Descriptor Status FIFO Write Pointer Register {0..7}

This register defines the write pointer of the DMA descriptor status FIFO. DMA channel Y (Y = 0–7) is assigned registers Y.

Register name: DMAC{0..7}DSWP Reset value: 0x0000_0000	Register offset: 0x5103C += 1000

Bits	7	6	5	4	3	2	1	0
31:24	RESERVED							
23:16	RESERVED					WR_PTR[18:16]		
15:8	WR_PTR[15:8]							
7:0	WR_PTR[7:0]							

Bits	Name	Description	Type	Reset value
31:19	RESERVED	RESERVED	R	0x0
18:0	WR_PTR	Write pointer to an entry of the descriptor status FIFO relative to the descriptor status FIFO base address.It is updated by the Tsi721 after issuing descriptor status MWr to PCIe MAC. The device resets this pointer upon a DMA channel initialization (see INIT in DMA Channel Interrupt Register {0..7}). Instead of using WR_PTR, software can only read this register. Software should decide if a descriptor status FIFO entry is valid by checking if the entry is all zeroes.	R/W	0x0

21.2.15 BDMA Interrupt Enable CSR

This register defines the BDMA interrupt enable function.

Register name: BDMA_INTE Reset value: 0x0000_0000	Register offset: 0x5F000

Bits	7	6	5	4	3	2	1	0
31:24	RESERVED							
23:16	Reserved					ECC_UNCORR_EN	ECC_CORR_EN	RESERVED
15:8	ECC_UNCORR_CH_EN[7:0]							
7:0	ECC_CORR_CH_EN[7:0]							

Bits	Name	Description	Type	Reset value
31:19	RESERVED	RESERVED	R	0x0
18	ECC_UNCORR_EN	Uncorrectable ECC Error interrupt enable 1 = Enable ECC_UNCORR in BDMA Interrupt CSR.	R/WS	0x0
17	ECC_CORR_EN	Correctable ECC Error interrupt enable 1 = Enable ECC_CORR in BDMA Interrupt CSR.	R/WS	0x0
16	RESERVED	RESERVED	R	0x0
15:8	ECC_UNCORR_CH_EN[7:0]	Uncorrectable ECC Error interrupt enable for channelized memory instances bit x = 1: Enable ECC_UNCORR_CH[x] of BDMA Interrupt CSR.	R/WS	0x0
7:0	ECC_CORR_CH_EN[7:0]	Correctable ECC Error interrupt enable for channelized memory instances bit x = 1: Enable ECC_CORR_CH[x] of BDMA Interrupt CSR.	R/WS	0x0

21.2.16 BDMA Interrupt CSR

This register defines BDMA interrupts.

Register name: BDMA_INT Reset value: 0x0000_0000	Register offset: 0x5F004

Bits	7	6	5	4	3	2	1	0
31:24	RESERVED							
23:16	Reserved					ECC_UNCORR	ECC_CORR	RESERVED
15:8	ECC_UNCORR_CH[7:0]							
7:0	ECC_CORR_CH[7:0]							

Bits	Name	Description	Type	Reset value
31:19	RESERVED	RESERVED	R	0x0
18	ECC_UNCORR	Uncorrectable ECC interrupt for non-channelized memory instances 1 = An uncorrectable ECC error occurred. It is enabled by ECC_UNCORR_EN of BDMA Interrupt Enable CSR.	R/W1CS	0x0
17	ECC_CORR	Correctable ECC interrupt for non-channelized memory instances 1 = A correctable ECC error occurred. It is enabled by ECC_CORR_EN of BDMA Interrupt Enable CSR.	R/W1CS	0x0
16	RESERVED	RESERVED	R	0x0
15:8	ECC_UNCORR_CH[7:0]	Uncorrectable ECC interrupt for channelized memory instances bit x = 1: An uncorrectable ECC error occurred for one of channel x's memory instances. It is enabled by ECC_UNCORR_CH_EN of BDMA Interrupt Enable CSR.	R/W1CS	0x0
7:0	ECC_CORR_CH[7:0]	Correctable ECC interrupt for channelized memory instances bit x = 1: A correctable ECC error occurred for one of channel x's memory instances. It is enabled by ECC_CORR_CH_EN of BDMA Interrupt Enable CSR.	R/W1CS	0x0

21.2.17 BDMA Interrupt Set Register

This register defines the BDMA interrupt set function for software debug.

Register name: BDMA_INTSET Reset value: 0x0000_0000	Register offset: 0x5F008

Bits	7	6	5	4	3	2	1	0
31:24	RESERVED							
23:16	Reserved					ECC_UNCORR_SET	ECC_CORR_SET	Reserved
15:8	ECC_UNCORR_CH_SET[7:0]							
7:0	ECC_CORR_CH_SET[7:0]							

Bits	Name	Description	Type	Reset value
31:19	RESERVED	RESERVED	R	0x0
18	ECC_UNCORR_SET	Uncorrectable ECC Error interrupt set 1 = Force uncorrectable ECC error interrupt ECC_UNCORR in BDMA Interrupt CSR to be set to 1.	R/W1S	0x0
17	ECC_CORR_SET	Correctable ECC Error interrupt set 1 = Force correctable ECC error interrupt ECC_CORR in BDMA Interrupt CSR to be set to 1.	R/W1S	0x0
16	RESERVED	RESERVED	R	0x0
15:8	ECC_UNCORR_CH_SET[7:0]	Uncorrectable ECC Error Interrupt Set for channelized memory instances 1 = Force uncorrectable ECC error interrupt ECC_UNCORR_CH[x] of BDMA Interrupt CSR to 1.	R/W1S	0x0
7:0	ECC_CORR_CH_SET[7:0]	Correctable ECC Error interrupt set for channelized memory instances 1 = Force correctable ECC error interrupt ECC_CORR_CH[x] of BDMA Interrupt CSR to 1.	R/W1S	0x0

21.2.18 BDMA ECC Log Register

This register defines the BDMA ECC log.

Register name: BDMA_ECC_LOG Reset value: 0x0000_0000	Register offset: 0x5F00C

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

Bits	Name	Description	Type	Reset value
31:24	RESERVED	RESERVED	R	0x0
23:16	ECC_UNCORR_MEM[7:0]	When bit x is set to 1, an uncorrectable ECC error occurred to non-channelized SRAM instance x since it was last cleared.	R/W1CS	0x0
15:8	RESERVED	RESERVED	R	0x0
7:0	ECC_CORR_MEM[7:0]	When bit x is set to 1, a correctable ECC error occurred to non-channelized SRAM instance x since it was last cleared.	R/W1CS	0x0

21.2.19 BDMA Channelized Correctable ECC LOG Register {0..7}

This register defines the BDMA channelized correctable ECC log. DMA channel Y (Y = 0–7) is assigned to register Y.

Register name: BDMA_ECC_CORR{0..7}LOG Reset value: 0x0000_0000	Register offset: 0x5F300 += 4

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

Bits	Name	Description	Type	Reset value
31:17	RESERVED	RESERVED	R	0x0
16:0	ECC_CORR_MEM[16:0]	When bit x is set to 1, a correctable ECC error occurred to SRAM instance x of this channel since it was last cleared.	R/W1CS	0x0

21.2.20 BDMA Channelized Uncorrectable ECC LOG Register {0..7}

This register defines the BDMA channelized uncorrectable ECC log. DMA channel Y (Y = 0–7) is assigned to register Y.

Register name: BDMA_ECC_UNCORR{0..7}LOG Reset value: 0x0000_0000	Register offset: 0x5F340 += 4

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

Bits	Name	Description	Type	Reset value
31:17	RESERVED	RESERVED	R	0x0
16:0	ECC_UNCORR_MEM[16:0]	When bit x is set to 1, an uncorrectable ECC error occurred to SRAM instance x of this channel since it was last cleared.	R/W1CS	0x0

22. I²C Registers

Topics discussed include the following:

- Register Map
- Register Descriptions

22.1 Register Map

All registers can be accessed through the internal register bus. A portion of the register space is visible to an external I²C master through the slave interface. The registers in this portion have a *peripheral address* in addition to their internal register bus address, and are called the *externally visible* I²C registers, meaning they can be accessed by the external I²C master using I²C reads and writes. The peripheral address equates to a 4-byte range within a consecutive 256-byte address space. For peripheral addresses, the lowest address maps to the least significant byte of the internal register (LSB), while the highest address maps to the most significant byte of the internal register (MSB).

The following table lists the register map for the I²C registers.

Table 97: I2C Register Map

Internal Address	Peripheral Address	Register Name	See
0x49000– 0x490FC	N/A	Reserved	
0x49100	N/A	I2C_DEVID	I2C Device ID Register
0x49104	N/A	I2C_RESET	I2C Reset Register
0x49108	N/A	I2C_MST_CFG	I2C Master Configuration Register
0x4910C	N/A	I2C_MST_CNTRL	I2C Master Control Register
0x49110	N/A	I2C_MST_RDATA	I2C Master Receive Data Register
0x49114	N/A	I2C_MST_TDATA	I2C Master Transmit Data Register
0x49118	N/A	I2C_ACC_STAT	I2C Access Status Register
0x4911C	N/A	I2C_INT_STAT	I2C Interrupt Status Register
0x49120	N/A	I2C_INT_ENABLE	I2C Interrupt Enable Register
0x49124	N/A	I2C_INT_SET	I2C Interrupt Set Register
0x4912C	N/A	I2C_SLV_CFG	I2C Slave Configuration Register

Table 97: I2C Register Map *(Continued)*

Internal Address	Peripheral Address	Register Name	See
0x49130– 0x4913C	N/A	Reserved	
0x49140	N/A	I2C_BOOT_CNTRL	I2C Boot Control Register
0x49144– 0x491FC	N/A	Reserved	
0x49200	0x00–0x03	EXI2C_REG_WADDR	Externally Visible I2C Internal Write Address Register
0x49204	0x04–0x07	EXI2C_REG_WDATA	Externally Visible I2C Internal Write Data Register
0x49208– 0x4920C	0x08–0x0F	Reserved	
0x49210	0x10–0x13	EXI2C_REG_RADDR	Externally Visible I2C Internal Read Address Register
0x49214	0x14–0x17	EXI2C_REG_RDATA	Externally Visible I2C Internal Read Data Register
0x49218– 0x4921C	0x18–0x1F	Reserved	
0x49220	0x20–0x23	EXI2C_ACC_STAT	Externally Visible I2C Slave Access Status Register
0x49224	0x24–0x27	EXI2C_ACC_CNTRL	Externally Visible I2C Internal Access Control Register
0x49228– 0x4927C	0x28–0x7F	Reserved	
0x49280	0x80–0x83	EXI2C_STAT	Externally Visible I2C Status Register
0x49284	0x84–0x87	EXI2C_STAT_ENABLE	Externally Visible I2C Enable Register
0x49288– 0x4928C	0x88–0x8F	Reserved	
0x49290	0x90–0x93	EXI2C_MBOX_OUT	Externally Visible I2C Outgoing Mailbox Register
0x49294	0x94–0x97	EXI2C_MBOX_IN	Externally Visible I2C Incoming Mailbox Register
0x49298– 0x492FC	0x98–0xFF	Reserved	
0x49300	N/A	I2C_EVENT	I2C Event and Event Snapshot Registers
0x49304	N/A	I2C_SNAP_EVENT	I2C Event and Event Snapshot Registers
0x49308	N/A	I2C_NEW_EVENT	I2C New Event Register
0x4930C	N/A	I2C_EVENT_ENB	I2C Enable Event Register

Table 97: I2C Register Map *(Continued)*

Internal Address	Peripheral Address	Register Name	See
0x49310– 0x4931C	N/A	Reserved	
0x49320	N/A	I2C_DIVIDER	I2C Time Period Divider Register
0x49324– 0x4933C	N/A	Reserved	
0x49340	N/A	I2C_START_SETUP_HOLD	I2C Start Condition Setup/Hold Timing Register
0x49344	N/A	I2C_STOP_IDLE	I2C Stop/Idle Timing Register
0x49348	N/A	I2C_SDA_SETUP_HOLD	I2C_SDA Setup and Hold Timing Register
0x4934C	N/A	I2C_SCL_PERIOD	I2C_SCL High and Low Timing Register
0x49350	N/A	I2C_SCL_MIN_PERIOD	I2C_SCL Minimum High and Low Timing Register
0x49354	N/A	I2C_SCL_ARB_TIMEOUT	I2C_SCL Low and Arbitration Timeout Register
0x49358	N/A	I2C_BYTE_TRAN_TIMEOUT	I2C Byte/Transaction Timeout Register
0x4935C	N/A	I2C_BOOT_DIAG_TIMER	I2C Boot and Diagnostic Timer
0x49360– 0x493B4	N/A	Reserved	
0x493B8	N/A	I2C_BOOT_DIAG_PROGRESS	I2C Boot Load Diagnostic Progress Register
0x493BC	N/A	I2C_BOOT_DIAG_CFG	I2C Boot Load Diagnostic Configuration Register
0x493C0– 0x493FC	N/A	Reserved	

22.2 Register Descriptions

This section describes the I^{2}C registers. These registers are reset by a fundamental reset.

22.2.1 I2C Device ID Register

This register identifies the version of the I^{2}C module in this device.

Register name: I2C_DEVID Reset value: 0x0000_0001	Register offset: 0x49100

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

Bits	Name	Description	Type	Reset Value
31:04	Reserved	Reserved	R	0x000_0000
03:00	REV	Revision Indicates the revision ID for the I^{2}C module.	R	0x1

22.2.2 I2C Reset Register

This register completes a “soft reset” of the I^{2}C Interface. A soft reset returns the logic to its idle, non-transacting state while retaining all configuration registers, such that the module does not have to be reprogrammed. This is provided for exceptional conditions. A reset while the module is involved in a transaction as a master or slave could leave the bus in an unexpected state relative to any external I^{2}C masters or slaves, and thus should be used with caution and only as a last solution if the module seems unresponsive.

I^{2}C registers that are affected by a soft reset are indicated in the description of that register. All other registers should be assumed to be unaffected by the soft reset.

Register name: I2C_RESET Reset value: 0x0000_0000	Register offset: 0x49104

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

Bits	Name	Description	Type	Reset Value
31	SRESET	Soft Reset Setting this bit resets the I^{2}C module. The R/W fields of configuration and control registers are not affected (nor is this register affected). While in reset neither the Master nor slave interface are operational: the I2C_SCL and I2C_SDA signals are undriven so as to not obstruct the bus. This bit must be written to 0 to bring the module out of reset. Any active bus transactions at the time of reset are aborted. All status and events are returned to the reset state. The boot load sequence is not invoked when the device exits from reset, although the bus idle detect sequence is invoked. Power-up latch values are not re-latched, and the fields remain at their pre-soft-reset value.	R/W	0
30:00	Reserved	Reserved	R	0

22.2.3 I2C Master Configuration Register

This register contains options that apply to master operations initiated through the I2C Master Control Register. The configuration specifies the properties of the external slave device to which a read or write transaction is directed.

Register name: I2C_MST_CFG Reset value: Undefined	Register offset: 0x49108

Bits	7	6	5	4	3	2	1	0
31:24	Reserved							
23:16	DORDER	Reserved					PA_SIZE	
15:08	Reserved							
07:00	Reserved	DEV_ADDR						

Bits	Name	Description	Type	Reset Value
31:24	Reserved	Reserved	R	0x00
23	DORDER	Data Order 0 = Data from/to data registers is ordered (processed) from MSB to LSB within an I^{2}C transaction. 1 = Data from/to data registers is ordered (processed) from LSB to MSB within an I^{2}C transaction. Data registers are I2C Master Transmit Data Register and I2C Master Receive Data Register.	R/W	0
22:18	Reserved	Reserved	R	0x00
17:16	PA_SIZE	Peripheral Address Size 00 = No peripheral address used 01 = 8-bit peripheral device addressing using LSB of PADDR 10 = 16-bit peripheral device addressing using MSB and LSB of PADDR (in that order) 11 = Reserved (handled as 00) This field selects the number of bytes in the peripheral address for master transactions. The peripheral address itself is specified in the I2C Master Control Register.	R/W	Undefined
15:7	Reserved	Reserved	R	0x000
6:0	DEV_ADDR	Device Address Specifies the 7-bit device address to select the I^{2}C device for a read or write transaction initiated through the I2C Master Control Register.	R/W	Undefined

Do not change this register while a master operation is active. The effect on the transaction cannot be determined.

22.2.4 I2C Master Control Register

This register sets the peripheral address and to start an I^{2}C transaction. The transaction is directed to the device defined in the I2C Master Configuration Register.

Note: Software must make sure not to set the peripheral address and the SIZE parameters so that page wrap-arounds occur in the target device. The Tsi721 does not force repeated start conditions within a single software initiated access.

Register name: I2C_MST_CNTRL Reset value: 0x0000_0000	Register offset: 0x4910C

Bits	7	6	5	4	3	2	1	0
31:24	START	WRITE	Reserved			SIZE		
23:16	Reserved							
15:08	PADDR							
07:00	PADDR							

Bits	Name	Description	Type	Reset Value
31	START	Start Operation 0 = I^{2}C operation is not active (self clears) 1 = Start of an I^{2}C operation. Clears itself back to zero when the initiated operation has completed. This bit cannot be cleared by software once set, except through a soft reset operation. This bit is cleared by a soft reset.	R/W1S	0
30	WRITE	I^{2}C Read or Write 0 = Read from I^{2}C memory 1 = Write to I^{2}C memory For a read, data is returned to the I2C Master Receive Data Register. For a write, data is taken from the I2C Master Transmit Data Register.	R/W	0
29:27	Reserved	Reserved	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
26:24	SIZE	Number of bytes in an I^{2}C operation (read or write) 000 = 0 bytes 001 = 1 bytes 010 = 2 bytes 011 = 3 bytes 100 = 4 bytes 101 = Reserved (equivalent of 0 bytes) 110 = Reserved (equivalent of 0 bytes) 111 = Reserved (equivalent of 0 bytes) A value of 000 should not be normally used for a Read operation because any device put into read mode assumes at least one byte will be read. An exception would be the SMBus Quick Command protocol to a device that is known to not hold the bus following the slave address phase.	R/W	000
23:16	Reserved	Reserved	R	0
15:00	PADDR	Peripheral address for master operation If PA_SIZE in the I2C Master Configuration Register is 10, then all 16 bits are used, with the MSB 8 bits placed on I^{2}C bus first, followed by the LSB 8 bits. If PA_SIZE is 01, then only LSB 8 bits are used. If PA_SIZE is 00, this field is not used for the transaction.	R/W	0x0000

A master operation is initiated by writing this register and setting the START bit to 1. The same write should set the WRITE, SIZE, and PADDR fields as required by the transaction. Other information for the transaction must have been pre-loaded into the I2C Master Configuration Register.

Do not change this register while a master operation is active. The effect on the transaction cannot be determined.

The following is the sequence triggered by setting the START bit:

Table 98: Master Operation Sequence

Phase	Description	Outcome
1. Begin transaction	Start arbitration timer.	-
2. Address slave	Detect bus idle. Send START. Send I2C_MST_CFG.[DEV_ADDR]. Send Read or Write (normally Write unless WRITE=0 and PA_SIZE=0). Wait for ACK/NACK. Disable arbitration timer.	Any loss of arbitration repeats this phase. Phase completes with ACK. NACK terminates operation with a MNACK event (issues STOP). Expiration of arbitration timer aborts operation with MARBTO.
3. Peripheral Address	If PA_SIZE = 10, send PADDR[15:8] and wait for ACK. If PA_SIZE = 01 or 10, send PADDR[7:0] and wait for ACK.	If PA_SIZE is 00 or 11, this phase is skipped. Any loss of arbitration will abort with MCOL. Any NACK will abort with MNACK.
4. Send Data (WRITE = 1)	If SIZE > 0, send SIZE bytes from I2C_MST_TDATA based on DORDER. Wait for ACK from each.	This phase is skipped if WRITE=1. Any loss of arbitration will abort with MCOL. Any NACK will abort with MNACK.
5. Read Data Setup (WRITE = 0)	Send RESTART. Repeat the “Address Slave” process with last bit a Read. Wait for ACK/NACK.	This phase is skipped if WRITE=1 or SIZE=0. NACK aborts with MNACK. Loss of arbitration aborts with MCOL (because bus was never released).
6. Read Data (WRITE = 0)	Read SIZE bytes and place in I2C_MST_RDATA based on DORDER. Respond with ACK to each, except for final byte respond with a NACK.	This phase is skipped if WRITE=1 or SIZE=0. Any loss of arbitration aborts with MCOL.
7. Complete	Issue STOP. Clear I2C_MST_CNTRL[START]. Set MSD event.	Master tries to force a STOP condition, except for arbitration loss.

22.2.5 I2C Master Receive Data Register

This register contains the data read from an external slave device following a read operation initiated using the I2C Master Control Register.

As bytes are read from the I^{2}C bus, they are placed in this register depending on DORDER in the I2C Master Configuration Register. If DORDER is 0, bytes are loaded from MSB to LSB, in order: RBYTE3, RBYTE2, RBYTE1, RBYTE0. If DORDER is 1, bytes are loaded from LSB to MSB, in order: RBYTE0, RBYTE1, RBYTE2, RBYTE3. If the transaction size is less than four (4) bytes (that is, SIZE in the I2C Master Control Register < 4) then any remaining bytes in the register are left unchanged (that is, they retain the values they had from the prior read operation).

Register name: I2C_MST_RDATA Reset value: 0x0000_0000	Register offset: 0x49110

Bits	7	6	5	4	3	2	1	0
31:24	RBYTE3							
23:16	RBYTE2							
15:08	RBYTE1							
07:00	RBYTE0							

Bits	Name	Description	Type	Reset value
31:24	RBYTE3	Received I^{2}C data — Byte 3 (most significant)	R	0x00
23:16	RBYTE2	Received I^{2}C data — Byte 2	R	0x00
15:08	RBYTE1	Received I^{2}C data — Byte 1	R	0x00
07:00	RBYTE0	Received I^{2}C data — Byte 0 (least significant)	R	0x00

22.2.6 I2C Master Transmit Data Register

This register contains the data to be written (transmitted) to an external slave when a write operation is initiated using the I2C Master Control Register. This register should be written with data to be sent before setting the START bit in that register.

As bytes are written to the I^{2}C bus, they are taken from this register depending on DORDER in the I2C Master Configuration Register. If DORDER is 0, bytes are taken from MSB to LSB, in order: TBYTE3, TBYTE2, TBYTE1, TBYTE0. If DORDER is 1, bytes are taken from LSB to MSB, in order: TBYTE0, TBYTE1, TBYTE2, TBYTE3. If the transaction size is less than 4 bytes (that is, SIZE in the I2C Master Control Register <4) then any remaining bytes in the register are unused. The contents of this register are not affected by the transaction.

Register name: I2C_MST_TDATA Reset value: 0x0000_0000	Register offset: 0x49114

Bits	7	6	5	4	3	2	1	0
31:24	TBYTE3							
23:16	TBYTE2							
15:08	TBYTE1							
07:00	TBYTE0							

Bits	Name	Description	Type	Reset value
31:24	TBYTE3	Transmitted I^{2}C data — Byte 3 (most significant)	R/W	0x00
23:16	TBYTE2	Transmitted I^{2}C data — Byte 2	R/W	0x00
15:08	TBYTE1	Transmitted I^{2}C data — Byte 1	R/W	0x00
07:00	TBYTE0	Transmitted I^{2}C data — Byte 0 (least significant)	R/W	0x00

⚠ Do not change this register while a master operation is active. The effect on the transaction cannot be determined.

22.2.7 I2C Access Status Register

This register indicates the status of the I^{2}C module. Fields in this register change dynamically as operations are initiated or progress.

Register name: I2C_ACC_STAT Reset value: 0x0000_0000	Register offset: 0x49118

Bits	7	6	5	4	3	2	1	0
31:24	SLV_ ACTIVE	BUS_ ACTIVE	Reserved		SLV_WAIT	SLV_PHASE		SLV_AN
23:16	SLV_PA							
15:08	MST_ ACTIVE	Reserved			MST_PHASE			MST_AN
07:00	Reserved				MST_NBYTES			

Bits	Name	Description	Type	Reset Value
31	SLV_ACTIVE	Slave Active 0 = Slave is not addressed 1 = Slave has been addressed by external master and a read or write is active This bit is set following the slave address phase if the address matched the SLV_ADDR or Alert Response Address and the slave interface was enabled. This bit is zeroed on a soft reset.	R	0
30	BUS_ACTIVE	Bus Active 0 = I^{2}C bus is not active 1 = I^{2}C bus is active: a START bit has been seen (and no subsequent STOP) This bit is zeroed on a soft reset, and is not set to 1 until a START condition is seen after soft reset is de-asserted.	R	0
29:28	Reserved	Reserved	R	00
27	SLV_WAIT	Slave Wait 0 = Slave is not waiting for a STOP or RESTART 1 = Slave is waiting for a STOP or RESTART This bit is clear if the bus is not active or the slave address is being received or the slave is active. This bit is set if the bus is active but the slave is not active and the slave address is not being received. This bit is zeroed on a soft reset.	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
26:25	SLV_PHASE	Slave Phase 00 = Slave address being received (even if slave interface is disabled using SLV_EN). 01 = Peripheral address being received 10 = Data incoming (write from external master) 11 = Data outgoing (read by external master) At the end of a slave operation, this field holds its value until the next START/RESTART. If a slave operation aborts, this field indicates where in the transaction the error occurred.	R	0x0
24	SLV_AN	Slave Ack/Nack 0 = Slave transaction is not in the ACK/NACK bit of a byte 1 = Slave transaction is in the ACK/NACK bit of a byte This qualifies the SLV_PHASE field.	R	0
23:16	SLV_PA	Slave Peripheral Address This field indicates the current peripheral address that is used when the Tsi721 is accessed by an external master.	R	0x00
15	MST_ACTIVE	Master Active 0 = No active master operation 1 = Master operation is active This status is the same as the START bit in the I2C Master Control Register. This bit is zeroed on a soft reset.	R	0
14:12	Reserved	Reserved	R	0
11:09	MST_PHASE	Master Phase 000 = START condition being sent 001 = External slave address being transmitted 010 = Peripheral address being transmitted 011 = RESTART condition being sent 100 = Data incoming (read operation) 101 = Data outgoing (write operation) 110 = STOP condition being sent 111 = Reserved At the end of a master operation, this field holds its value until the next master operation is started. If a master operation aborts, this field indicates where in the transaction the error occurred.	R	000

(Continued)

Bits	Name	Description	Type	Reset Value
08	MST_AN	Master Ack/Nack 0 = Master transaction is not in the ACK/NACK bit of a byte 1 = Master transaction is in the ACK/NACK bit of a byte This qualifies the MST_PHASE field.	R	0
07:04	Reserved	Reserved	R	0x0
03:00	MST_NBYTES	Master Number of Bytes This is the running count of the number of data bytes transferred in the current master operation (read or write). At the end of an operation, if successfully completed, the field will equal the SIZE field from the I2C Master Control Register. If an operation aborts prematurely, this field indicates the number of bytes transferred before the error occurred.	R	0x0

This document is confidential and is subject to an NDA.

22.2.8 I2C Interrupt Status Register

This register indicates the status of the I^{2}C interrupts. When an interrupt status bit is set, an interrupt is generated to the Interrupt Controller if the corresponding bit is enabled in the I2C Interrupt Enable Register. If the corresponding enable is not set, the interrupt status bit still asserts but does assert an interrupt to the Interrupt Controller. This register can only be accessed through the register bus.

Note: This register is affected by a soft reset. All interrupts are cleared and no interrupt will assert until an event occurs after soft reset is de-asserted.

Register name: I2C_INT_STAT Reset value: 0x0000_0000	Register offset: 0x4911C

Bits	7	6	5	4	3	2	1	0
31:24	Reserved						OMB_ EMPTY	IMB_FULL
23:16	Reserved						BL_FAIL	BL_OK
15:08	Reserved				SA_FAIL	SA_WRITE	SA_READ	SA_OK
07:00	MA_DIAG	Reserved		MA_COL	MA_TMO	MA_NACK	MA_ATMO	MA_OK

Bits	Name	Description	Type	Reset Value
31:26	Reserved	Reserved	R	0x00
25	OMB_EMPTY	Outgoing Mailbox Empty 0 = Interrupt status not asserted 1 = Outgoing mailbox is empty Set when an external I^{2}C master reads data from the mailbox (see Externally Visible I2C Outgoing Mailbox Register), if data had been previously been written to the mailbox by software.	R/W1C	0
24	IMB_FULL	Incoming Mailbox Full 0 = Interrupt status not asserted 1 = Incoming mailbox is full Set when an external I^{2}C master writes data into the Externally Visible I2C Incoming Mailbox Register.	R/W1C	0
23:18	Reserved	Reserved	R	0x00
17	BL_FAIL	Boot Load Failed 0 = Interrupt status not asserted 1 = Boot load sequence failed to complete	R/W1C	0

(Continued)

Bits	Name	Description	Type	Reset Value
16	BL_OK	Boot Load OK 0 = Interrupt status not asserted 1 = Boot load sequence completed successfully This bit is also set if the boot loading was disabled at reset.	R/W1C	0
15:12	Reserved	Reserved	R	0x0
11	SA_FAIL	Slave Access Failed 0 = Interrupt status not asserted 1 = Error detected during slave access transaction A slave transaction addressed to the Tsi721 aborted.	R/W1C	0
10	SA_WRITE	Slave Write 0 = Interrupt status not asserted 1 = Internal register write performed The Externally Visible I2C Internal Write Data Register was written by an external master, invoking a write to an internal register. This will not assert if slave writes are disabled (WR_EN in the I2C Slave Configuration Register = 0).	R/W1C	0
09	SA_READ	Slave Read 0 = Interrupt status not asserted 1 = Internal register read performed The Externally Visible I2C Internal Read Data Register was read by an external master, invoking a read to an internal register. This will not assert if slave reads are disabled (RD_EN = 0 in the I2C Slave Configuration Register).	R/W1C	0
08	SA_OK	Slave Access OK 0 = Interrupt status not asserted 1 = Access completed successfully The Tsi721 was addressed as a slave device and the transaction completed without error.	R/W1C	0
07	MA_DIAG	Master Diagnostic Event 0 = Interrupt status not asserted 1 = Diagnostic event	R/W1C	0
06:05	Reserved	Reserved	R	00

(Continued)

Bits	Name	Description	Type	Reset Value
04	MA_COL	Master Collision 0 = Interrupt status not asserted 1 = Collision (arbitration loss) occurred following the device address phase A transaction initiated using the I2C Master Control Register aborted due to loss of arbitration after the slave device address phase. This indicates multiple masters tried to access the same slave. This can also be set at the end of boot load due to BL_FAIL.	R/W1C	0
03	MA_TMO	Master Timeout 0 = Interrupt status not asserted 1 = Transaction aborted due to timeout expiration A transaction initiated using the I2C Master Control Register aborted due to expiration of the clock low, byte, or transaction timeouts. This can also be set at the end of boot load due to BL_FAIL.	R/W1C	0
02	MA_NACK	Master NACK 0 = Interrupt status not asserted 1 = NACK received during transaction A transaction initiated through the I2C Master Control Register aborted due to receipt of a NACK in response to slave address, peripheral address, or a written byte. This can also be set at the end of boot load due to BL_FAIL.	R/W1C	0
01	MA_ATMO	Master Arbitration Timeout 0 = Interrupt status not asserted 1 = Bus arbitration timeout expired A transaction initiated through the I2C Master Control Register aborted due to expiration of the arbitration timeout (for more information, see ARB_TO in I2C_SCL Low and Arbitration Timeout Register). This indicates the bus is in use by other masters.	R/W1C	0
00	MA_OK	Master Transaction OK 0 = Interrupt status not asserted 1 = Access completed and successful A transaction initiated through the I2C Master Control Register completed without error.	R/W1C	0

The write-1-to-clear (W1C) operation requires that this register first be read to create an event snapshot.

22.2.9 I2C Interrupt Enable Register

This register controls which of the interrupt status bits in the I2C Interrupt Status Register will generate an interrupt to the Interrupt Controller. It can only be accessed from the register bus.

Register name: I2C_INT_ENABLE Reset value: 0x0000_0000	Register offset: 0x49120

Bits	7	6	5	4	3	2	1	0
31:24	Reserved						OMB_ EMPTY	IMB_FULL
23:16	Reserved						BL_FAIL	BL_OK
15:08	Reserved				SA_ FAIL	SA_ WRITE	SA_READ	SA_OK
07:00	MA_DIAG	Reserved		MA_COL	MA_TMO	MA_NACK	MA_ATMO	MA_OK

Bits	Name	Description	Type	Reset Value
31:26	Reserved	Reserved	R	0x00
25	OMB_EMPTY	Enable OMB_EMPTY Interrupt 0 = Interrupt is disabled 1 = Interrupt is enabled	R/W	0
24	IMB_FULL	Enable IMB_FULL Interrupt 0 = Interrupt is disabled 1 = Interrupt is enabled	R/W	0
23:18	Reserved	Reserved	R	0x00
17	BL_FAIL	Enable BL_FAIL Interrupt 0 = Interrupt is disabled 1 = Interrupt is enabled	R/W	0
16	BL_OK	Enable BL_OK Interrupt 0 = Interrupt is disabled 1 = Interrupt is enabled	R/W	0
15:12	Reserved	Reserved	R	0x0
11	SA_FAIL	Enable SA_FAIL Interrupt 0 = Interrupt is disabled 1 = Interrupt is enabled	R/W	0

(Continued)

Bits	Name	Description	Type	Reset Value
10	SA_WRITE	Enable SA_WRITE Interrupt 0 = Interrupt is disabled 1 = Interrupt is enabled	R/W	0
09	SA_READ	Enable SA_READ Interrupt 0 = Interrupt is disabled 1 = Interrupt is enabled	R/W	0
08	SA_OK	Enable SA_OK Interrupt 0 = Interrupt is disabled 1 = Interrupt is enabled	R/W	0
07	MA_DIAG	Enable MA_DIAG Interrupt 0 = Interrupt is disabled 1 = Interrupt is enabled	R/W	0
06:05	Reserved	Reserved	R	00
04	MA_COL	Enable MA_COL Interrupt 0 = Interrupt is disabled 1 = Interrupt is enabled	R/W	0
03	MA_TMO	Enable MA_TMO Interrupt 0 = Interrupt is disabled 1 = Interrupt is enabled	R/W	0
02	MA_NACK	Enable MA_NACK Interrupt 0 = Interrupt is disabled 1 = Interrupt is enabled	R/W	0
01	MA_ATMO	Enable MA_ATMO Interrupt 0 = Interrupt is disabled 1 = Interrupt is enabled	R/W	0
00	MA_OK	Enable MA_OK Interrupt 0 = Interrupt is disabled 1 = Interrupt is enabled	R/W	0

22.2.10 I2C Interrupt Set Register

This register sets the status of the I^{2}C blocks interrupts. It can only be accessed from the register bus.

Note: Setting an interrupt sets all related underlying events in the I2C New Event Register. This is significant in that if all underlying events are disabled for a specific interrupt bit, this register will not appear to work for that bit.

Register name: I2C_INT_SET Reset value: 0x0000_0000	Register offset: 0x49124

Bits	7	6	5	4	3	2	1	0
31:24	Reserved						OMB_ EMPTY	IMB_ FULL
23:16	Reserved						BL_FAIL	BL_OK
15:08	Reserved				SA_FAIL	SA_ WRITE	SA_READ	SA_OK
07:00	MA_DIAG	Reserved		MA_COL	MA_TMO	MA_NACK	MA_ATMO	MA_OK

Bits	Name	Description	Type	Reset Value
31:26	Reserved	Reserved	R	0x00
25	OMB_EMPTY	Set OMB_EMPTY Interrupt 0 = No effect 1 = Interrupt is set	R/W1S	0
24	IMB_FULL	Set IMB_FULL Interrupt 0 = No effect 1 = Interrupt is set	R/W1S	0
23:18	Reserved	Reserved	R	0x00
17	BL_FAIL	Set BL_FAIL Interrupt 0 = No effect 1 = Interrupt is set	R/W1S	0
16	BL_OK	Set BL_OK Interrupt 0 = No effect 1 = Interrupt is set	R/W1S	0
15:12	Reserved	Reserved	R	0x0
11	SA_FAIL	Set SA_FAIL Interrupt 0 = No effect 1 = Interrupt is set	R/W1S	0

(Continued)

Bits	Name	Description	Type	Reset Value
10	SA_WRITE	Set SA_WRITE Interrupt 0 = No effect 1 = Interrupt is set	R/W1S	0
09	SA_READ	Set SA_READ Interrupt 0 = No effect 1 = Interrupt is set	R/W1S	0
08	SA_OK	Set SA_OK Interrupt 0 = No effect 1 = Interrupt is set	R/W1S	0
07	MA_DIAG	Set MA_DIAG Interrupt 0 = No effect 1 = Interrupt is set	R/W1S	0
06:05	Reserved	Reserved	R	00
04	MA_COL	Set MA_COL Interrupt 0 = No effect 1 = Interrupt is set	R/W1S	0
03	MA_TMO	Set MA_TMO Interrupt 0 = No effect 1 = Interrupt is set	R/W1S	0
02	MA_NACK	Set MA_NACK Interrupt 0 = No effect 1 = Interrupt is set	R/W1S	0
01	MA_ATMO	Set MA_ATMO Interrupt 0 = No effect 1 = Interrupt is set	R/W1S	0
00	MA_OK	Set MA_OK Interrupt 0 = No effect 1 = Interrupt is set	R/W1S	0

22.2.11 I2C Slave Configuration Register

This register configures the slave interface portion of the I^{2}C module. The slave interface is the logic that responds to transactions from an external master on the I^{2}C bus.

Register name: I2C_SLV_CFG Reset value: undefined	Register offset: 0x4912C

Bits	7	6	5	4	3	2	1	0
31:24	RD_EN	WR_EN	ALRT_EN	SLV_EN	Reserved			SLV_UNLK
23:16	Reserved							
15:08	Reserved							
07:00	Reserved	SLV_ADDR						

Bits	Name	Description	Type	Reset Value
31	RD_EN	Register Bus Read Enable This bit controls whether external masters can read registers internal to the Tsi721. The SLV_EN bit must also be set for this option to have any effect. 0 = Transactions that read the Externally Visible I2C Internal Read Data Register will not invoke reads of the internal registers. 1 = Transactions that read the Externally Visible I2C Internal Read Data Register will trigger reads of the internal register whose address is in the Externally Visible I2C Internal Read Address Register.	R/W	1
30	WR_EN	Register Bus Write Enable This bit controls whether external masters can write to Tsi721’s internal registers. The SLV_EN bit must also be set for this option to have any effect. 0 = Transactions that write the Externally Visible I2C Internal Write Data Register will not invoke writes of the internal registers. 1 = Transactions that write the Externally Visible I2C Internal Write Data Register will trigger writes of the internal register whose address is in the Externally Visible I2C Internal Write Address Register.	R/W	1

(Continued)

Bits	Name	Description	Type	Reset Value
29	ALRT_EN	Alert Address Enable 0 = Do not respond to read of the Alert Response Address of 0001100 1 = Respond to read of the Alert Response Address of 0001100 if any bits are set in the Externally Visible I2C Status Register. If enabled, the slave interface will respond to the Alert Response Address for a read transaction if the ALERT_FLAG is set in the Externally Visible I2C Slave Access Status Register. The response is to return the SLV_ADDR field followed by a 0 to the external master, then clear the ALERT_FLAG. If ALRT_EN is 0, then the Alert Response address can be used as a SLV_ADDR. If ALRT_EN is 1 and SLV_ADDR is also set to the alert response address, then the alert response behavior takes precedence.	R/W	0
28	SLV_EN	Slave Enable 0 = Slave is not enabled; SLV_ADDR is ignored. 1 = Slave interface is enabled; SLV_ADDR is responded to when transaction started by external master. When enabled, the slave interface acknowledges transactions to the SLV_ADDR from an external master. If not enabled, then all transactions are NACK'd, except the Alert Response Address read, if ALRT_EN is 1. This bit controls access to the peripheral address space of the Tsi721. Access to the internal register space is also controlled by the RD_EN and WR_EN bits. If SLV_EN is 0 then internal register access is also disabled.	R/W	1
27:25	Reserved	Reserved	R	000
24	SLV_UNLK	Slave Address Unlock 0 = The LSB 4 bits of the SLV_ADDR are locked for writing (a write will leave those bits unchanged). 1 = The LSB 4 bits of the SLV_ADDR are unlocked, and can be changed by a write. This bit controls a write-protect on the 4 LSBs of the SLV_ADDR field. When 0 those bits are not writeable, which protects the power-up latch value of those bits. To change the bits, this SLV_UNLK bit must be written to 1 on the write performed to this register that is changing the SLV_ADDR[28:31] bits.	R/W	0

(Continued)

Bits	Name	Description	Type	Reset Value
23:07	Reserved	Reserved	R	0x0000
06:00	SLV_ADDR	Slave Address This is the device address for the Tsi721 as an I^{2}C slave. An external master uses this address to access the Tsi721 peripheral register space. For the slave interface to respond to this address, the SLV_EN bit must be set. The bits of this field are latched at power-up from the state of input signals. The LSB 4 bits are then locked for writing until the SLV_UNLK bit is set to 1 during a write. This feature allows the boot load process to write a slave address value to this register without changing the power-up latch field. A SLV_ADDR of 0x00 is never valid, as that value is used for the I2C START_BYTE and General Call functions. Specific functions are not supported by the Tsi721 and are ignored and NACK'd. The power on reset value of this register is set to {0b011, I2C_SA[3:0]}.	R/W	undefined

22.2.12 I2C Boot Control Register

This register controls the boot load sequence that is initiated following a fundamental reset of the Tsi721. The initial boot load operation is controlled by the reset state of this register. Some of the fields are also latched from device signals at power-up.

Once boot loading is active, the data read from the EEPROM can modify the contents of this register and redirect the loading to another EEPROM, or to another address within the same EEPROM. This process is called “chaining.” The progress of a boot load operation can be monitored using the I2C Boot Load Diagnostic Progress Register and I2C Boot Load Diagnostic Configuration Register.

This register can be read and written after boot loading is complete, but has no further effect on module operation.

Register name: I2C_BOOT_CNTRL Reset value: undefined	Register offset: 0x49140

Bits	7	6	5	4	3	2	1	0
31:24	CHAIN	PSIZE	BINC	BUNLK	Reserved			
23:16	Reserved	BOOT_ADDR						
15:08	PAGE_MODE			PADDR				
07:00	PADDR							

Bits	Name	Description	Type	Reset Value
31	CHAIN	Chain During Boot 0 = No chain 1 = Chain to new device This bit is set to invoke a chain operation during boot load. To chain, this bit must be set and the register load count must be at zero; that is, the write to this register must be the last one in the boot sequence within this EEPROM if chaining were not continuing the boot. Except for the BUNLK and PAGE_MODE fields, modifications to the remaining fields in this register have no effect unless this bit is set. The fields will change value, but they will not affect the boot load sequence. Once boot load is complete, this register has no further effect on module operation.	R/W	0

(Continued)

Bits	Name	Description	Type	Reset Value
30	PSIZE	Peripheral Address Size 0 = Use 1 byte for peripheral address 1 = Use 2 bytes for peripheral address This selects the number of bytes in the peripheral address. If 0 then only LSB 5 bits of PADDR are used (+ 3 LSBs of 000). If 1 then all 13 bits of PADDR are used (+ 3 LSBs of 000). For 2-byte addressing, the MSB of the address is transmitted first on the I^{2}C bus (for example, see the PADDR field). This field can be changed during the boot load, in conjunction with setting the CHAIN bit, to jump the boot load to a new boot device with different address size. The power on reset value of this register is set by I2C_MA.	R/W	undefined
29	BINC	Boot Address Increment 0 = Do not increment boot address when peripheral address overflows 1 = Increment LSB 3 bits of the internal boot address when peripheral address overflows, then re-address device This option is valid only when PSIZE is 0, and accesses devices that use the LSB 3 bits of their device address as a 256-byte page select (usually 2K EEPROMs). When enabled, and the 1-byte peripheral address wraps back to zero, the LSB 3 bits of the device address is incremented, followed by a Restart and a new device address cycle. The device address starts as the value of the BOOT_ADDR field, and is copied internally at boot start or during a chain operation. It is the internal value that is incremented to simulate addressing a 2K EEPROM. This field can be changed during the boot load, in conjunction with setting the CHAIN bit, to jump the boot load to a new boot device with new page properties.	R/W	1
28	BUNLK	Boot Address Unlock 0 = The LSB 2 bits of the BOOT_ADDR are locked for writing (a write will leave those bits unchanged). 1 = The LSB 2 bits of the BOOT_ADDR are unlocked, and can be changed by a write. This bit controls a write-protect on the two LSBs of the BOOT_ADDR field. When 0, those bits are not writeable, which protects the power-up latch value of those bits. To change the bits, this BUNLK bit must be written as 1 on the write performed to this register that is changing the BOOT_ADDR[14:15] bits.	R/W	0

(Continued)

Bits	Name	Description	Type	Reset Value
27:23	Reserved	Reserved	R	0x00
22:16	BOOT_ADDR	Boot Device Address This is the device address that the Tsi721 accesses during the boot load sequence. The LSB two bits [14:15] of this field are latched at power-up from the state of input signals. This allows board configuration of up to four unique Tsi721 devices on the I2C bus, each of which access a different EEPROM during boot load. These two bits are then locked for writing until the BUNLK bit is set to 1 during a write. This feature allows the boot load process to change the boot address value to this register without changing the power-up latch field. This field can be changed during the boot load in conjunction with setting the CHAIN bit, to jump the boot load to a new boot device. This starting address is copied internally at boot start or boot load, and it is the internal value that is incremented, as explained in the BINC field. The power on reset value of this register is set by I2C_SEL and I2C_SA[1:0].	R/W	undefined

(Continued)

Bits	Name	Description	Type	Reset Value
15:13	PAGE_MODE	Page Mode 000 = 8 bytes 001 = 32 bytes 010 = 64 bytes 011 = 128 bytes 100 = 256 bytes 101 = 512 bytes 110 = 1024 bytes 111 = Infinite This field modifies the boot load process to adjust the boundary at which the boot device is re-addressed. In the default case, the boot load sequence reads 8 bytes, then does a Restart followed by the device and peripheral address phases. By changing this field, the device is re-addressed only when the peripheral address crosses the indicated boundary, thus saving a considerable number of clock cycles during the boot. It is up to the programmer of the EEPROM to ensure that the addressed device can support consecutive reads up to the selected boundary. Some devices wrap at certain page boundaries, and this setting must be consistent with such limitations. A setting of 111 causes the entire boot load to be read sequentially, with two exceptions. No matter what the setting of this field, if the boot address is incremented due to the BINC mode being enabled, or if chaining occurs, then the device is readdressed. Changing this field during boot load immediately affects the boot sequence.	R/W	000

(Continued)

Bits	Name	Description	Type	Reset Value
12:00	PADDR	Peripheral Address This is the MSB 5 or 13 bits of the peripheral address (depending on PSIZE setting). The LSB 3 bits are not programmable and are assumed 000; that is, the peripheral address must be aligned to a multiple of 8 address in the EEPROM. To form the peripheral address, this field is shifted left by 3 and then copied internally when a boot start or a chain operation occurs. The internal address is then incremented as the boot load progresses. For 2-byte addressing, the MSB of the peripheral address is sent first. For example, setting this field to 0x0127 gives a peripheral address of (0x0127 << 3) = 0x0938. The first byte sent to the external device is 0x09 and the second byte is 0x38. This field can be changed during the boot load, in conjunction with setting the CHAIN bit, to jump the boot load to a new peripheral address.	R/W	0x0000

22.2.13 Externally Visible I2C Internal Write Address Register

This register contains the internal register address set by an external I^{2}C master to be used for internal register writes when the Externally Visible I2C Internal Write Data Register is written. The address is forced to be 4-byte aligned (the 2 lowest bits are read-only).

This register is read-only from the register bus, and R/W from the I^{2}C bus through the slave interface. This register corresponds to the I^{2}C peripheral address 0x00 through 0x03.

Note: This register is also used during the boot load process to accumulate the address read from the EEPROM for each address/data pair. Therefore, at the end of the boot load process, this register will contain the last register address read from the EEPROM, or the first four bytes of the register count.

Register name: EXI2C_REG_WADDR Reset value: 0x0000_0000	Register offset: 0x49200

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

Bits	Name	Description	Type	Reset Value
31:02	ADDR	Internal Register Write Address Register address to be used when a write to the Externally Visible I2C Internal Write Data Register invokes an internal register write. This address is 4-byte aligned. The specific byte accessed is controlled by the peripheral address within the data register. Only the LSB 24 bits are significant to the Tsi721. The MSB 8 bits can be written but will not have any effect. This address auto-increments by 4 if WINC in the Externally Visible I2C Internal Access Control Register is set, and the MSB of the data (peripheral address 0x07) is written.	R	0x0000_ 0000
01:00	Reserved	Reserved	R	00

22.2.14 Externally Visible I2C Internal Write Data Register

This register contains the internal register data last written by an external I^2C master through the slave interface. The register is read-only from the register bus, and R/W from the I^2C bus through the slave interface.

This register corresponds to the I^2C peripheral addresses 0x04 through 0x07.

Note: This register is also used during the boot load process to accumulate the data read from the EEPROM for each address/data pair. Therefore, at the end of the boot load process, this register will contain the last register data read from the EEPROM, or the last four bytes of the register count.

Register name: EXI2C_REG_WDATA Reset value: 0x0000_0000	Register offset: 0x49204

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

Bits	Name	Description	Type	Reset Value
31:00	WDATA	Internal Register Write Data Data written by the external I^2C master to be used for an internal register write. When WSIZE is configured for 4-byte access in the Externally Visible I2C Internal Access Control Register, the contents of this register are written to an internal register when the MSB is written by an external I^2C master (peripheral address 0x07 = WDATA[00:07]). The register bus address is taken from the Externally Visible I2C Internal Write Address Register. Writes of other bytes (peripheral addresses 0x04-06) do not invoke internal accesses. Note: When the MSB of this register is written (peripheral address 0x07), the slave peripheral address wraps to 0x04 (the LSB of this register) instead of incrementing to 0x08. This allows an external master to write a module of internal registers without having to change the slave peripheral address, assuming WINC in the Externally Visible I2C Internal Access Control Register is set to auto-increment the WADDR. When 0x07 is read, the peripheral address increments to 0x08.	R	0

22.2.15 Externally Visible I2C Internal Read Address Register

This register contains the internal register address set by an external I^{2}C master to be used for internal register reads when the Externally Visible I2C Internal Read Data Register is read. The address is forced to be 4-byte aligned (the 2 lowest bits are read-only).

This register is read-only from the register bus, and R/W from the I^{2}C bus through the slave interface. This register corresponds to the I^{2}C peripheral address 0x10 through 0x13.

Register name: EXI2C_REG_RADDR Reset value: 0x0000_0000	Register offset: 0x49210

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

Bits	Name	Description	Type	Reset Value
31:02	ADDR	Internal Register Read Address Register address to be used when a read to the Externally Visible I2C Internal Read Data Register invokes an internal register read. This address is 4-byte aligned. The specific byte accessed is controlled by the peripheral address within the data register. Only the LSB 24 bits are significant to the Tsi721. The MSB 8 bits can be written, but will not have any effect. This address auto-increments by 4 if RINC in the Externally Visible I2C Internal Access Control Register is set, and the LSB of the data (peripheral address 0x14) is read.	R	0x0000_ 0000
01:00	Reserved	Reserved	R	00

22.2.16 Externally Visible I2C Internal Read Data Register

This register contains the internal register data last read by an external I^{2}C master through the slave interface. The register is read-only from both the register bus and the I^{2}C bus through the slave interface.

This register corresponds to the I^{2}C peripheral addresses 0x14 through 0x17.

Register name: EXI2C_REG_RDATA Reset value: 0x0000_0000	Register offset: 0x49214

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

Bits	Name	Description	Type	Reset Value
31:00	RDATA	Internal Register Read Data Data read by the external I^{2}C master as a side effect of reading this register. When RSIZE is configured for 4-byte access in the Externally Visible I2C Internal Access Control Register, this register is updated by the read of an internal register when the LSB is read by an external I^{2}C master (peripheral address 0x14 = RDATA[24:31]). The register bus address is taken from the Externally Visible I2C Internal Read Address Register. Reads of subsequent bytes (peripheral addresses 0x15-17) do not invoke internal accesses. Note: When the MSB of this register is read (peripheral address 0x17), the slave peripheral address wraps to 0x14 (the LSB of this register) instead of incrementing to 0x18. This allows an external master to read a module of internal registers without having to change the slave peripheral address, assuming RINC in the Externally Visible I2C Internal Access Control Register is set to auto-increment the RADDR. When 0x17 is written, the peripheral address increments to 0x18.	R	0

22.2.17 Externally Visible I2C Slave Access Status Register

This register provides status indications to an external I^2C master. It is read-only from both the register bus and the I^2C bus through the slave interface. This register corresponds to the I^2C peripheral addresses 0x20 through 0x23.

Note: This register is affected by a soft reset. All status will be cleared.

Register name: EXI2C_ACC_STAT Reset value: 0x0000_0000	Register offset: 0x49220

Bits	7	6	5	4	3	2	1	0
31:24	Reserved							
23:16	Reserved							
15:08	Reserved							
07:00	ACC_OK	Reserved			OMB_ FLAG	IMB_ FLAG	Reserved	ALERT_ FLAG

Bits	Name	Description	Type	Reset Value
31:08	Reserved	Reserved	R	0x00_0000
07	ACC_OK	Internal Register Access OK 0 = No access, or access in progress 1 = Access was successful This bit is set when a slave access successfully reads or writes data to an internal register through the Externally Visible I2C Internal Write Data Register or Externally Visible I2C Internal Read Data Register. Reading this bit returns the last status of the bit. If read through the slave interface (through peripheral address 0x20), the bit is then cleared to 0. The bit is not cleared to 0 when read by a host or indirectly through the EXI2C_REG_RADDR / EXI2C_REG_RDATA mechanism.	R	0
06:04	Reserved	Reserved	R	000

(Continued)

Bits	Name	Description	Type	Reset Value
03	OMB_FLAG	Outgoing Mailbox Flag 0 = Outgoing mailbox empty 1 = New data in the outgoing mailbox This bit is set when data is written to the outgoing mailbox register (Externally Visible I2C Outgoing Mailbox Register) by software. This bit remains set (flag up) until an external I^{2}C master reads the outgoing mailbox register, and the bit is then cleared (flag down). When the mailbox is read, the OMB_EMPTY interrupt is asserted. A mailbox read is considered complete when the external master issues a STOP condition to end the transaction during which any bytes in the mailbox were written.	R	0
02	IMB_FLAG	Incoming Mailbox Flag 0 = Incoming mailbox empty 1 = New data in the incoming mailbox This bit is set when data is written to the incoming mailbox register (Externally Visible I2C Incoming Mailbox Register) by an external I^{2}C master. This bit remains set (flag up) until software reads the incoming mailbox register, and the bit is then cleared (flag down). When the mailbox is written and the flag is set, the IMB_FULL interrupt is asserted. A mailbox read is considered complete when the external master issues a STOP condition to end the transaction during which any bytes in the mailbox were written.	R	0
01	Reserved	Reserved	R	0
00	ALERT_FLAG	Alert Response Flag 0 = No alert 1 = Alert response active This bit is set when the Alert Response would trigger, as defined in the Externally Visible I2C Status Register. It is cleared by a successful response to the Alert Response Address or if the global status no longer requires the alert to be asserted. On a hard reset, this flag will assert immediately due to Externally Visible I2C Status Register[RESET] asserting.	R	0

This document is confidential and is subject to an NDA.

22.2.18 Externally Visible I2C Internal Access Control Register

This register allows an external I^{2}C master to configure the functionality for internal register accesses through the slave interface. This register is read-only from the register bus and R/W from the I^{2}C bus through the slave interface.

The fields in this register control the size and auto-increment functions when internal register accesses are performed by an external master through the slave interface.

This register corresponds to the I^{2}C peripheral addresses 0x24 through 0x27.

Register name: EXI2C_ACC_CNTRL Reset value: 0x0000_00A0	Register offset: 0x49224

Bits	7	6	5	4	3	2	1	0
31:24	Reserved							
23:16	Reserved							
15:08	Reserved							
07:00	RSIZE		WSIZE		RINC	WINC	Reserved	

Bits	Name	Description	Type	Reset Value
31:08	Reserved	Reserved	R	0x00_0000
07:06	RSIZE	Internal Register Read Access Size 00 = 1 byte (Reserved) 01 = 2 bytes (Reserved) 10 = 4 bytes – An internal register read is invoked once for each internal register, loading all 4 bytes in the Externally Visible I2C Internal Read Data Register. The read is performed when the LSB of the data register is read (peripheral address 0x14). 11 = 8 bytes (Reserved) All Reserved settings will result in internal read accesses being disabled.	R	10

(Continued)

Bits	Name	Description	Type	Reset Value
05:04	WSIZE	Internal Register Write Access Size 00 = 1 byte (Reserved) 01 = 2 bytes (Reserved) 10 = 4 bytes – An internal register write is invoked once for each internal register, writing all 4 bytes from the Externally Visible I2C Internal Write Data Register. The write is performed when the MSB of the data register is written (peripheral address 0x07). 11 = 8 bytes (Reserved) All Reserved settings will result in internal writes accesses being disabled.	R	10
03	RINC	Enable Auto-Incrementing on Internal Register Reads 0 = The Externally Visible I2C Internal Read Address Register is unchanged after reads performed to the Externally Visible I2C Internal Read Data Register 1 = The Externally Visible I2C Internal Read Address Register is incremented by 4 after reads performed to the LSB of the Externally Visible I2C Internal Read Data Register (peripheral address 0x14) so that the address points to the next internal register address. When auto-incrementing is on, consecutive internal registers can be read in one I^{2}C transaction without the need to reset the peripheral address because the peripheral address wraps from 0x17 back to 0x14. If auto-incrementing is off, then the same internal register can be read multiple times in a single I^{2}C transaction. The latter could be useful for polling a status register.	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
02	WINC	Enable Auto-Incrementing on Internal Register Writes 0 = The Externally Visible I2C Internal Write Address Register is unchanged after reads performed to the Externally Visible I2C Internal Write Data Register. 1 = The Externally Visible I2C Internal Write Address Register is incremented by 4 after writes performed to the MSB of the Externally Visible I2C Internal Write Data Register (peripheral address 0x07) so that the address points to the next internal register address. When auto-incrementing is on, consecutive internal registers can be written in one I^2C transaction with the need to reset the peripheral address because the peripheral address wraps from 0x07 back to 0x04. If auto-incrementing is off, then the same internal register can be written multiple times in a single I^2C transaction.	R	0
01:00	Reserved	Reserved	R	00

22.2.19 Externally Visible I2C Status Register

This register provides a summary view of status of the Tsi721. It can be polled by an external system management device. Any bit masked by its related enable, changing from 0 to 1, will cause ALERT_FLAG to be set in the Externally Visible I2C Slave Access Status Register, and the Tsi721 to respond to the Alert Response Address if the ALRT_EN bit is set in the I2C Slave Configuration Register. The related enables are present in the Externally Visible I2C Enable Register. If all masked status bits are 0, then the ALERT_FLAG clears. The ALERT_FLAG also clears when the slave responds to the Alert Response Address, and not set again until there is a change in the status.

The register is read only from the register bus, but R/W1C from the I^{2}C bus through the slave interface. They are set when the corresponding event occurs within the Tsi721, and held asserted until an external I^{2}C master writes a 1 to that position to clear the event. If an event is still asserting when the W1C occurs, the bit remains set.

The software status bits SW_STAT{0..2} are R/W from the register bus. They can be set or cleared by software, and thereby used for any system purpose. An external I^{2}C master can write 1 to those bits to clear them. If the W1C occurs at the same time as software is writing the bit, the software written value will take precedence.

This register corresponds to the I^{2}C slave addresses 0x80 through 0x83.

Note: This register is affected by a soft reset. All status will be cleared, including the software status bits. Chip status will re-assert after a soft reset is released only if that chip event occurs again.

Register name: EXI2C_STAT Reset value: 0x0000_0000	Register offset: 0x49280

Bits	7	6	5	4	3	2	1	0
31:24	RESET	SW_ STAT2	SW_ STAT1	SW_ STAT0	OMBW	IMBR	I2C	Reserved
23:16	Reserved							
15:08	Reserved							
07:00	Reserved	ECC_UNCORR	SRIO_MAC	DL_DOWN	SMSG_NONCH	PC2SR	SR2PC_NONCH	BDMA_NONCH

Bits	Name	Description	Type	Reset Value
31	RESET	Reset Status 0 = No reset has occurred since the last time this bit was cleared. 1 = A reset has occurred since the last time this bit was cleared. This indicates a hardware reset of the Tsi721. This bit is set to 1 on a hard reset, and cleared to 0 on a soft reset of the I^{2}C logic.	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
30	SW_STAT2	Software Status Bit 2 0 = Status value 0 1 = Status value 1 This bit can be set or cleared by software using a register write for any system specific purpose.	R/W	0
29	SW_STAT1	Software Status Bit 1 0 = Status value 0 1 = Status value 1 This bit can be set or cleared by software using a register write for any system specific purpose.	R/W	0
28	SW_STAT0	Software Status Bit 0 0 = Status value 0 1 = Status value 1 This bit can be set or cleared by software using a register write for any system specific purpose.	R/W	0
27	OMBW	Outgoing Mailbox Written 0 = Outgoing mailbox not filled since last clear 1 = Outgoing mailbox has been filled This bit asserted indicates that software has written to the outgoing mailbox since this bit was last cleared.	R	0
26	IMBR	Incoming Mailbox Read 0 = Incoming mailbox not read since last clear 1 = Incoming mailbox has been emptied This bit asserted indicates that software has read the incoming mailbox, when the mailbox was full, since this bit was last cleared.	R	0
25	I2C	I^{2}C Interrupt 0 = I^{2}C is not asserting an interrupt to processor 1 = I^{2}C is asserting an interrupt to the processor	R	0
24:7	Reserved	Reserved	R	0x00
06	ECC_UNCORR	Non-correctable ECC interrupt When ECC_UNCORR is set to 1, SMSG/BDMA/SR2PC/PC2SR has an active non-correctable ECC interrupt or S-RIO MAC has a active non-data memory non-correctable ECC interrupt	R	0
05	SRIO_MAC	S-RIO MAC interrupt When SRIO_MAC is set to 1, S-RIO MAC has an active interrupt	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
04	DL_DOWN	PCIe link down interrupt When DL_DOWN is set to 1, PCIe link is down	R	0
03	SMSG_NONCH	SMSG non-channelized interrupt When SMSG is set to 1, SMSG has a non-channelized interrupt	R	0
02	PC2SR	PC2SR interrupt When PC2SR is set to 1, PC2SR has an interrupt	R	0
01	SR2PC_NONCH	SR2PC non-channelized interrupt When SR2PC is set to 1, SR2PC has a non-channelized interrupt	R	0
00	BDMA_NONCH	Block DMA Engine non-channelized interrupt When set to 1, the Block DMA Engine has a non-channelized interrupt	R	0

22.2.20 Externally Visible I2C Enable Register

Any bit set in this register will enable the equivalent bit in the Externally Visible I2C Status Register to set the ALERT_FLAG. These enables do not affect whether events are set in the global status register, only whether the asserted events can set the ALERT_FLAG when changing from 0 to 1. If an event is already asserted in the status when the related enable is changed from 0 to 1, this is equivalent to the event asserting, and the ALERT_FLAG will be set.

This register is R/W from either the register bus or from the I^{2}C bus through the slave interface. If the register is written by both at the same time, the register bus interface will take precedence.

This register corresponds to the I^{2}C peripheral addresses 0x84 through 0x87.

Register name: EXI2C_STAT_ENABLE Reset value: 0xFFFF_FFFF	Register offset: 0x49284

Bits	7	6	5	4	3	2	1	0
31:24	RESET	SW_ STAT2	SW_ STAT1	SW_ STAT0	OMBW	IMBR	I2C	Reserved
23:16	Reserved							
15:08	Reserved							
07:00	Reserved	ECC_UNCORR	SRIO_MAC	DL_DOWN	SMSG_NONCH	PC2SR	SR2PC_NONCH	BDMA_NONCH

Bits	Name	Description	Type	Reset Value
31	RESET	Enable RESET Alert Response 0 = Status asserted will not enable setting ALERT_FLAG 1 = Status asserted will enable setting ALERT_FLAG	R/W	1
30	SW_STAT2	Enable SW_STAT2 Alert Response 0 = Status asserted will not enable setting ALERT_FLAG 1 = Status asserted will enable setting ALERT_FLAG	R/W	1
29	SW_STAT1	Enable SW_STAT1 Alert Response 0 = Status asserted will not enable setting ALERT_FLAG 1 = Status asserted will enable setting ALERT_FLAG	R/W	1
28	SW_STAT0	Enable SW_STAT0 Alert Response 0 = Status asserted will not enable setting ALERT_FLAG 1 = Status asserted will enable setting ALERT_FLAG	R/W	1
27	OMBW	Enable Outgoing Mailbox Written 0 = Status asserted will not enable setting ALERT_FLAG 1 = Status asserted will enable setting ALERT_FLAG	R/W	1

(Continued)

Bits	Name	Description	Type	Reset Value
26	IMBR	Enable Incoming Mailbox Read 0 = Status asserted will not enable setting ALERT_FLAG 1 = Status asserted will enable setting ALERT_FLAG	R/W	1
25	I2C	Enable I2C Alert Response 0 = Status asserted will not enable setting ALERT_FLAG 1 = Status asserted will enable setting ALERT_FLAG	R/W	1
24:7	Reserved	Reserved These bits are unused in the Tsi721. The enables can be changed, but have no effect.	R/W	0x3FFFF
06	ECC_UNCORR	Enable ECC_UNCORR Alert Response 0 = Status asserted will not enable setting ALERT_FLAG 1 = Status asserted will enable setting ALERT_FLAG	R/W	1
05	SRIO_MAC	Enable S-RIO Alert Response 0 = Status asserted will not enable setting ALERT_FLAG 1 = Status asserted will enable setting ALERT_FLAG	R/W	1
04	DL_DOWN	Enable PCIe link down Alert Response 0 = Status asserted will not enable setting ALERT_FLAG 1 = Status asserted will enable setting ALERT_FLAG	R/W	1
03	SMSG_NONCH	Enable SMSG_NONCH Alert Response 0 = Status asserted will not enable setting ALERT_FLAG 1 = Status asserted will enable setting ALERT_FLAG	R/W	1
02	PC2SR	Enable PC2SR Alert Response 0 = Status asserted will not enable setting ALERT_FLAG 1 = Status asserted will enable setting ALERT_FLAG	R/W	1
01	SR2PC_NONCH	Enable SR2PC_NONCH Alert Response 0 = Status asserted will not enable setting ALERT_FLAG 1 = Status asserted will enable setting ALERT_FLAG	R/W	1
00	BDMA_NONCH	Enable BDMA_NONCH Alert Response 0 = Status asserted will not enable setting ALERT_FLAG 1 = Status asserted will enable setting ALERT_FLAG	R/W	1

22.2.21 Externally Visible I2C Outgoing Mailbox Register

This register is the outgoing mailbox, allowing the processor to communicate data to an external I^{2}C master. The register is R/W from the register bus, and read-only from the I^{2}C bus through the slave interface.

This register corresponds to the I^{2}C peripheral addresses 0x90 through 0x93.

Register name: EXI2C_MBOX_OUT Reset value: 0x0000_0000	Register offset: 0x49290

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

Bits	Name	Description	Type	Reset Value
31:00	DATA	Mailbox data to be transferred to an external I^{2}C master. Every write to this register by software sets the OMB_FLAG bit in the Externally Visible I2C Slave Access Status Register, indicating data is available in the outgoing mailbox. When this register is read by an external master, the OMB_FLAG bit is cleared, and an OMB_EMPTY interrupt is asserted if the OMB_FLAG bit was set. This register is read-only through the I^{2}C slave interface. Note: A read is considered complete when the STOP condition is seen on the I^{2}C bus, and one or more bytes in this register were read by the external master since the preceding START condition.	R/W	0

22.2.22 Externally Visible I2C Incoming Mailbox Register

This register is the incoming mailbox, allowing an external I^{2}C master to communicate data to the processor. The register is read-only from the register bus, and R/W from the I^{2}C bus through the slave interface.

This register corresponds to the I^{2}C peripheral addresses 0x94 through 0x97.

Register name: EXI2C_MBOX_IN Reset value: 0x0000_0000	Register offset: 0x49294

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

Bits	Name	Description	Type	Reset value
31:00	DATA	Mailbox data transferred from an external I^{2}C master. Every write to this register sets the IMB_FLAG in the Externally Visible I2C Slave Access Status Register, and asserts an IMB_FULL interrupt. When software reads this register, the IMB_FLAG is cleared. Note: This register is writable only through the I2C slave interface.	R	0

22.2.23 I2C Event and Event Snapshot Registers

These registers indicate events that occur within the I^{2}C module. For the I2C_EVENT register, each bit is an “or” of the corresponding bit in the I2C New Event Register and the I2C_SNAP_EVENT register. The I2C_SNAP_EVENT register contains those events that were asserted before the last snapshot. A snapshot is taken when the I2C Interrupt Status Register is read. Each bit in these registers are write-one-to-clear. Writing a 1 to a bit position in the I2C_EVENT register will clear the event in both the snapshot and new event registers. Writing a 1 to a bit position in the I2C_SNAP_EVENT register will clear the bit only in that register. Writing a 1 to a bit position in the I2C Interrupt Status Register will clear all related event bits in the I2C_SNAP_EVENT register, provided those events are enabled in the I2C_EVENT_ENB register. Bits from the I2C_EVENT register are masked (enabled) by the corresponding bits in the I2C Enable Event Register, and then determine whether a related bit in the I2C Interrupt Status Register is set.

Note: These registers are affected by a soft reset. All events will be cleared and will not assert while a soft reset is asserted.

Register name: I2C_{EVENT, SNAP_EVENT} Reset value: 0x0000_0000	Register offset: 0x49300 += 0x4

Bits	7	6	5	4	3	2	1	0
31:24	Reserved	SDW	SDR	SD	Reserved	DTIMER	DHIST	DCMDD
23:16	IMBW	OMBR	Reserved	SCOL	STRTO	SBTTO	SSCLTO	Reserved
15:08	Reserved	MTD	Reserved	BLTO	BLERR	BLSZ	BLNOD	BLOK
07:00	Reserved		MNACK	MCOL	MTRTO	MBTTO	MSCLTO	MARBTO

Bits	Name	Description	Type	Reset Value
31	Reserved	Reserved	R	0
30	SDW	Slave Internal Register Write Done Event 0 = Event not asserted 1 = Slave interface completed a transaction for an external master that resulted in a write to an internal register	R/W1C	0
29	SDR	Slave Internal Register Read Done Event 0 = Event not asserted 1 = Slave interface completed a transaction for an external master that resulted in a read to an internal register	R/W1C	0
28	SD	Slave Transaction Done Event 0 = Event not asserted 1 = Slave interface completed an I^{2}C transaction for an external master with no detectable error	R/W1C	0
27	Reserved	Reserved	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
26	DTIMER	Diagnostic Timer Expired Event 0 = Event not asserted 1 = Diagnostic timer has expired This event does not assert during the boot load sequence. The BLTO will assert instead.	R/W1C	0
25	DHIST	Diagnostic History Filling Event 0 = Event not asserted 1 = Diagnostic history recorded the 8th event	R/W1C	0
24	DCMDD	Diagnostic Command Done Event 0 = Event not asserted 1 = Master interface completed the diagnostic command	R/W1C	0
23	IMBW	Incoming Mailbox Write Event 0 = Event not asserted 1 = Slave interface completed a write transaction to the incoming mailbox	R/W1C	0
22	OMBR	Outgoing Mailbox Read Event 0 = Event not asserted 1 = Slave interface completed a read transaction to the outgoing mailbox when the OMB_FLAG was set The event is asserted only if the mailbox was full.	R/W1C	0
21	Reserved	Reserved	R	0
20	SCOL	Slave Collision Detect Event 0 = Event not asserted 1 = Slave interface detected a bit collision on the I^2C bus during a slave transaction initiated by an external master. The slave interface was not able to successfully assert a 1 for a data bit or for a NACK.	R/W1C	0
19	STRTO	Slave Transaction Timeout Event 0 = Event not asserted 1 = Transaction timer expired during a slave transaction initiated by an external master	R/W1C	0
18	SBTTO	Slave Byte Timeout Event 0 = Event not asserted 1 = Byte timer expired during a slave transaction initiated by an external master	R/W1C	0

(Continued)

Bits	Name	Description	Type	Reset Value
17	SSCLTO	Slave I2C_SCL Low Timeout Event 0 = Event not asserted 1 = I2C_SCL low timer expired during a slave transaction initiated by an external master	R/W1C	0
16:15	Reserved	Reserved	R	00
14	MTD	Master Transaction Done Event 0 = Event not asserted 1 = Master interface completed the I^{2}C transaction initiated through the I2C Master Control Register	R/W1C	0
13	Reserved	Reserved	R	0
12	BLTO	Boot Load Timeout Event 0 = Event not asserted 1 = Boot load timer expired This event only asserts during the boot load sequence if the Boot/Diagnostic timer expires. During normal operation, the DTIMER event will assert.	R/W1C	0
11	BLERR	Boot Load Error Event 0 = Event not asserted 1 = The boot load sequence failed due to an error during register reading: a protocol error (NACK when ACK expected), an I2C_SCL low timer, collision after the device addressing phase, or the last six bytes of a register count field not being 0xFF. This error will be qualified by the MNACK, MCOL or MSCLTO event. The last data read from the EEPROM is visible in the EXI2C_REG_WADDR and EXI2C_REG_WDATA registers.	R/W1C	0
10	BLSZ	Boot Load Size Error Event 0 = Event not asserted 1 = The boot load sequence aborted because the count field is incorrect, indicating an improperly loaded boot EEPROM. The register count is visible in the EXI2C_REG_WADDR register.	R/W1C	0
09	BLNOD	Boot Load No Device Event 0 = Event not asserted 1 = The boot load sequencer received a NACK 6 times when trying to first address the slave device. No device is responding to the boot load device address.	R/W1C	0

(Continued)

Bits	Name	Description	Type	Reset Value
08	BLOK	Boot Load OK Event 0 = Event not asserted 1 = The boot load sequence completed with no detectable errors. This bit is also asserted if boot load is disabled during power up.	R/W1C	0
07:06	Reserved	Reserved	R	00
05	MNACK	Master NACK Received Event 0 = Event not asserted 1 = Master interface received a NACK from a slave device during a transaction initiated through the I2C Master Control Register. This event can also assert during boot load and provide more information on the source of a BLERR event.	R/W1C	0
04	MCOL	Master Collision Detect Event 0 = Event not asserted 1 = Master interface lost arbitration after it addressed the slave device during a transaction initiated through the I2C Master Control Register. Another master is competing for access to the same slave device. This event can also assert during boot load and provide more information on the source of a BLERR event.	R/W1C	0
03	MTRTO	Master Transaction Timeout Event 0 = Event not asserted 1 = Transaction timeout timer expired during a transaction initiated through the I2C Master Control Register	R/W1C	0
02	MBTTO	Master Byte Timeout Event 0 = Event not asserted 1 = Byte timeout timer expired during a transaction initiated through the I2C Master Control Register	R/W1C	0

(Continued)

Bits	Name	Description	Type	Reset Value
01	MSCLTO	Master I2C_SCL Low Timeout Event 0 = Event not asserted 1 = SCL_TO timeout timer expired during a transaction initiated through the I2C Master Control Register. Another device is holding the I2C_SCL signal low. This event can also assert during boot load and provide more detail on the source of a BLERR event.	R/W1C	0
00	MARBTO	Master Arbitration Timeout Event 0 = Event not asserted 1 = Arbitration timeout timer expired during a transaction initiated through the I2C Master Control Register. Another master has control of the I^2C bus.	R/W1C	0

22.2.24 I2C New Event Register

This register indicates events that occurred since the last snapshot. This register is write-one-to-set. Writing a 1 to a bit position will set the event for diagnostic purposes. The register is cleared by writing to the I2C_EVENT register (see I2C Event and Event Snapshot Registers) or by creating a snapshot by reading the I2C Interrupt Status Register. For individual event descriptions, see the I2C_EVENT register.

Note: This register is affected by a soft reset. All events will be cleared and will not assert while a soft reset is asserted.

Register name: I2C_NEW_EVENT Reset value: 0x0000_0000	Register offset: 0x49308

Bits	7	6	5	4	3	2	1	0
31:24	Reserved	SDW	SDR	SD	Reserved	DTIMER	DHIST	DCMDD
23:16	IMBW	OMBR	Reserved	SCOL	STRTO	SBTTO	SSCLTO	Reserved
15:08	Reserved	MTD	Reserved	BLTO	BLERR	BLSZ	BLNOD	BLOK
07:00	Reserved		MNACK	MCOL	MTRTO	MBTTO	MSCLTO	MARBTO

Bits	Name	Description	Type	Reset Value
31	Reserved	Reserved	R	0
30	SDW	Slave Internal Register Write Done Event 0 = Event not asserted 1 = Event asserted	R/W1S	0
29	SDR	Slave Internal Register Read Done Event 0 = Event not asserted 1 = Event asserted	R/W1S	0
28	SD	Slave Transaction Done Event 0 = Event not asserted 1 = Event asserted	R/W1S	0
27	Reserved	Reserved	R	0
26	DTIMER	Diagnostic Timer Expired Event 0 = Event not asserted 1 = Event asserted	R/W1S	0
25	DHIST	Diagnostic History Filling Event 0 = Event not asserted 1 = Event asserted	R/W1S	0

(Continued)

Bits	Name	Description	Type	Reset Value
24	DCMDD	Diagnostic Command Done Event 0 = Event not asserted 1 = Event asserted	R/W1S	0
23	IMBW	Incoming Mailbox Write Event 0 = Event not asserted 1 = Event asserted	R/W1S	0
22	OMBR	Outgoing Mailbox Read Event 0 = Event not asserted 1 = Event asserted	R/W1S	0
21	Reserved	Reserved	R	0
20	SCOL	Slave Collision Detect Event 0 = Event not asserted 1 = Event asserted	R/W1S	0
19	STRTO	Slave Transaction Timeout Event 0 = Event not asserted 1 = Event asserted	R/W1S	0
18	SBTTO	Slave Byte Timeout Event 0 = Event not asserted 1 = Event asserted	R/W1S	0
17	SSCLTO	Slave I2C_SCL Low Timeout Event 0 = Event not asserted 1 = Event asserted	R/W1S	0
16:15	Reserved	Reserved	R	00
14	MTD	Master Transaction Done Event 0 = Event not asserted 1 = Event asserted	R/W1S	0
13	Reserved	Reserved	R	0
12	BLTO	Boot Load Timeout Event 0 = Event not asserted 1 = Event asserted	R/W1S	0
11	BLERR	Boot Load Error Event 0 = Event not asserted 1 = Event asserted	R/W1S	0

This document is confidential and is subject to an NDA.

(Continued)

Bits	Name	Description	Type	Reset Value
10	BLSZ	Boot Load Size Error Event 0 = Event not asserted 1 = Event asserted	R/W1S	0
09	BLNOD	Boot Load No Device Event 0 = Event not asserted 1 = Event asserted	R/W1S	0
08	BLOK	Boot Load OK Event 0 = Event not asserted 1 = Event asserted	R/W1S	0
07:06	Reserved	Reserved	R	00
05	MNACK	Master NACK Received Event 0 = Event not asserted 1 = Event asserted	R/W1S	0
04	MCOL	Master Collision Detect Event 0 = Event not asserted 1 = Event asserted	R/W1S	0
03	MTRTO	Master Transaction Timeout Event 0 = Event not asserted 1 = Event asserted	R/W1S	0
02	MBTTO	Master Byte Timeout Event 0 = Event not asserted 1 = Event asserted	R/W1S	0
01	MSCLTO	Master I2C_SCL Low Timeout Event 0 = Event not asserted 1 = Event asserted	R/W1S	0
00	MARBTO	Master Arbitration Timeout Event 0 = Event not asserted 1 = Event asserted	R/W1S	0

22.2.25 I2C Enable Event Register

This register modifies the function of the I2C_EVENT register (see I2C Event and Event Snapshot Registers). Each bit in this register enables (1) or disables (0) the corresponding event in the I2C_EVENT register from asserting in the I2C Interrupt Status Register.

Register name: I2C_EVENT_ENB Reset value: 0x74DE_5F3F	Register offset: 0x4930C

Bits	7	6	5	4	3	2	1	0
31:24	Reserved	SDW	SDR	SD	Reserved	DTIMER	DHIST	DCMDD
23:16	IMBW	OMBR	Reserved	SCOL	STRTO	SBTTO	SSCLTO	Reserved
15:08	Reserved	MTD	Reserved	BLTO	BLERR	BLSZ	BLNOD	BLOK
07:00	Reserved		MNACK	MCOL	MTRTO	MBTTO	MSCLTO	MARBTO

Bits	Name	Description	Type	Reset Value
31	Reserved	Reserved	R	0
30	SDW	Slave Internal Register Write Done Enable 0 = Event does not assert to interrupt status 1 = Event will assert in interrupt status	R/W	1
29	SDR	Slave Internal Register Read Done Enable 0 = Event does not assert to interrupt status 1 = Event will assert in the interrupt status	R/W	1
28	SD	Slave Transaction Done Enable 0 = Event does not assert to interrupt status 1 = Event will assert in the interrupt status	R/W	1
27	Reserved	Reserved	R	0
26	DTIMER	Diagnostic Timer Expired Enable 0 = Event does not assert to interrupt status 1 = Event will assert in the interrupt status	R/W	1
25	DHIST	Diagnostic History Filling Enable 0 = Event does not assert to interrupt status 1 = Event will assert in the interrupt status	R/W	0
24	DCMDD	Diagnostic Command Done Enable 0 = Event does not assert to interrupt status 1 = Event will assert in the interrupt status	R/W	0

(Continued)

Bits	Name	Description	Type	Reset Value
23	IMBW	Incoming Mailbox Write Enable 0 = Event does not assert to interrupt status 1 = Event will assert in the interrupt status	R/W	1
22	OMBR	Outgoing Mailbox Read Enable 0 = Event does not assert to interrupt status 1 = Event will assert in the interrupt status	R/W	1
21	Reserved	Reserved	R	0
20	SCOL	Slave Collision Detect Enable 0 = Event does not assert to interrupt status 1 = Event will assert in the interrupt status	R/W	1
19	STRTO	Slave Transaction Timeout Enable 0 = Event does not assert to interrupt status 1 = Event will assert in the interrupt status	R/W	1
18	SBTTO	Slave Byte Timeout Enable 0 = Event does not assert to interrupt status 1 = Event will assert in the interrupt status	R/W	1
17	SSCLTO	Slave I2C_SCL Low Timeout Enable 0 = Event does not assert to interrupt status 1 = Event will assert in the interrupt status	R/W	1
16:15	Reserved	Reserved	R	00
14	MTD	Master Transaction Done Enable 0 = Event does not assert to interrupt status 1 = Event will assert in the interrupt status	R/W	1
13	Reserved	Reserved	R	0
12	BLTO	Boot Load Timeout Enable 0 = Event does not assert to interrupt status 1 = Event will assert in the interrupt status	R/W	1
11	BLERR	Boot Load Error Enable 0 = Event does not assert to interrupt status 1 = Event will assert in the interrupt status	R/W	1
10	BLSZ	Boot Load Size Error Enable 0 = Event does not assert to interrupt status 1 = Event will assert in the interrupt status	R/W	1

(Continued)

Bits	Name	Description	Type	Reset Value
09	BLNOD	Boot Load No Device Enable 0 = Event does not assert to interrupt status 1 = Event will assert in the interrupt status	R/W	1
08	BLOK	Boot Load OK Enable 0 = Event does not assert to interrupt status 1 = Event will assert in the interrupt status	R/W	1
07:06	Reserved	Reserved	R	00
05	MNACK	Master NACK Received Enable 0 = Event does not assert to interrupt status 1 = Event will assert in the interrupt status	R/W	1
04	MCOL	Master Collision Detect Enable 0 = Event does not assert to interrupt status 1 = Event will assert in the interrupt status	R/W	1
03	MTRTO	Master Transaction Timeout Enable 0 = Event does not assert to interrupt status 1 = Event will assert in the interrupt status	R/W	1
02	MBTTO	Master Byte Timeout Enable 0 = Event does not assert to interrupt status 1 = Event will assert in the interrupt status	R/W	1
01	MSCLTO	Master I2C_SCL Low Timeout Enable 0 = Event does not assert to interrupt status 1 = Event will assert in the interrupt status	R/W	1
00	MARBTO	Master Arbitration Timeout Enable 0 = Event does not assert to interrupt status 1 = Event will assert in the interrupt status	R/W	1

22.2.26 I2C Time Period Divider Register

This register provides programmable extension of the reference clock period into longer periods used by the timeout and idle detect timers.

Register name: I2C_DIVIDER Reset value: 0x00F9_03E7	Register offset: 0x49320

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

Bits	Name	Description	Type	Reset Value
31:28	Reserved	Reserved	R	0x0
27:16	USDIV	Period Divider for Micro-Second Based Timers This field divides the reference clock down for use by the Idle Detect Timer, the Byte Timeout Timer, the I2C_SCL Low Timeout Timer, and the Milli-Second Period Divider. Period(USDIV) = Period(P_CLK) * (USDIV + 1), where P_CLK is 4 ns. Reset period is 1 microsecond.	R/W	0x0F9
15:12	Reserved	Reserved	R	0x0
11:00	MSDIV	Period Divider for Milli-Second Based Timers This field divides the USDIV period down further for use by the Arbitration Timeout Timer, the Transaction Timeout Timer, and the Boot/Diag Timeout Timer. Period(MSDIV) = Period(USDIV) * (MSDIV + 1). Reset period is 1 millisecond.	R/W	0x3E7

22.2.27 I2C Start Condition Setup/Hold Timing Register

This register programs the setup and hold timing for the Start condition when generated by the master control logic. The timer periods are relative to the reference clock. This register is shadowed during boot loading, and can be reprogrammed before a chain operation without affecting the bus timing for the current EEPROM.

Register name: I2C_START_SETUP_HOLD Reset value: 0x0498_03E9	Register offset: 0x49340

Bits	7	6	5	4	3	2	1	0
31:24	START_SETUP							
23:16	START_SETUP							
15:08	START_HOLD							
07:00	START_HOLD							

Bits	Name	Description	Type	Reset Value
31:16	START_SETUP	Count for the START Condition Setup Period Defines the minimum setup time for the START condition; that is, both I2C_SCL and I2C_SDA seen high before I2C_SDA pulled low. This is a master-only timing parameter. This value also doubles as the effective Stop Hold time. Period(START_SETUP) = (START_SETUP * Period(PCLK)), where PCLK is 4 ns. Reset time is 4.704 microseconds.	R/W	0x0498
15:00	START_HOLD	Count for the START Condition Hold Period Defines the minimum hold time for the START condition; that is, from I2C_SDA seen low to I2C_SCL pulled low. This is a master only timing parameter. Period(START_HOLD) = (START_HOLD * Period(P_CLK)), where P_CLK is 4 ns. Reset time is 4.004 microseconds.	R/W	0x03E9

22.2.28 I2C Stop/Idle Timing Register

This register programs the setup timing for the Stop condition when generated by the master control logic, and the Idle Detect timer. The Start Setup time doubles as the Stop Hold. The timer period for the Stop setup is relative to the reference clock. The timer period for the Idle Detect is relative to the USDIV period. The STOP setup time is shadowed during boot loading, and can be reprogrammed before a chain operation without affecting the bus timing for the current EEPROM.

Register name: I2C_STOP_IDLE Reset value: 0x03E9_0033	Register offset: 0x49344

Bits	7	6	5	4	3	2	1	0
31:24	STOP_SETUP							
23:16	STOP_SETUP							
15:08	IDLE_DET							
07:00	IDLE_DET							

Bits	Name	Description	Type	Reset Value
31:16	STOP_SETUP	Count for STOP Condition Setup Period Defines the minimum setup time for the STOP condition; that is, both I2C_SCL seen high and I2C_SDA seen low before I2C_SDA released high. This is a master-only timing parameter. Period(STOP_SETUP) = (STOP_SETUP * Period(P_CLK)), where P_CLK is 4 ns. Reset time is 4.004 microseconds.	R/W	0x03E9
15:00	IDLE_DET	Count for Idle Detect Period Used in two cases. First, defines the period after reset during which the I2C_SCL signal must be seen high to call the bus idle. This period is needed to avoid interfering with an ongoing transaction after reset. Second, defines the period before a master transaction during which the I2C_SCL and I2C_SDA signals must both be seen high to call the bus idle. This period is a protection against external master devices not correctly idling the bus. Period(IDLE_DET) = (IDLE_DET * Period(USDIV)), where USDIV is the microsecond time defined in the I2C Time Period Divider Register. A value of zero causes no idle detect period, meaning the bus will be sensed as idle immediately. Reset time is 51 microseconds.	R/W	0x0033

22.2.29 I2C_SDA Setup and Hold Timing Register

This register programs the setup and hold times for the I2C_SDA signal when output by either the master or slave interface. It is shadowed during boot loading, and can be reprogrammed before a chain operation without affecting the bus timing for the current EEPROM.

Register name: I2C_SDA_SETUP_HOLD Reset value: 0x013A_004C	Register offset: 0x49348

Bits	7	6	5	4	3	2	1	0
31:24	SDA_SETUP							
23:16	SDA_SETUP							
15:08	SDA_HOLD							
07:00	SDA_HOLD							

Bits	Name	Description	Type	Reset Value
31:16	SDA_SETUP	Count for the I2C_SDA Setup Period Defines the minimum setup time for the I2C_SDA signal; that is, I2C_SDA set to desired value before rising edge of I2C_SCL. This applies to both slave and master interface. Note: This value should be set to the sum of the I2C_SDA setup time and the maximum rise/fall time of the I2C_SDA signal to ensure that the signal is valid on the output at the correct time. This time is different than the raw I2C_SDA setup time in the *I^2C Specification*. Period(SDA_SETUP) = (SDA_SETUP * Period(P_CLK)), where P_CLK is 4 ns. Reset time is 1256 nanoseconds.	R/W	0x013A
15:00	SDA_HOLD	Count for I2C_SDA Hold Period Defines the minimum hold time for the I2C_SDA signal; that is, I2C_SDA valid past the falling edge of I2C_SCL. This applies to both slave and master interface. Period(SDA_HOLD) = (SDA_HOLD * Period(P_CLK)), where P_CLK is 4 ns. Reset time is 314 nanoseconds.	R/W	0x004C

22.2.30 I2C_SCL High and Low Timing Register

This register programs the nominal high and low periods of the I2C_SCL signal when generated by the master interface. It is shadowed during boot loading, and can be reprogrammed before a chain operation without affecting the bus timing for the current EEPROM.

Register name: I2C_SCL_PERIOD Reset value: 0x04E2_04E2	Register offset: 0x4934C

Bits	7	6	5	4	3	2	1	0
31:24	SCL_HIGH							
23:16	SCL_HIGH							
15:08	SCL_LOW							
07:00	SCL_LOW							

Bits	Name	Description	Type	Reset Value
31:16	SCL_HIGH	Count for I2C_SCL High Period Defines the nominal high period of the clock, from rising edge to falling edge of I2C_SCL. This is a master-only parameter. The observed period can be shorter if other devices pull the clock low. Period(SCL_HIGH) = (SCL_HIGH * Period(P_CLK)), where P_CLK is 4 ns. Reset time is 5.00 microseconds (100 kHz).	R/W	0x04E2
15:00	SCL_LOW	Count for I2C_SCL Low Period Defines the nominal low period of the clock, from falling edge to rising edge of I2C_SCL. This is a master-only parameter. The observed period can be longer if other devices pull the clock low. Period(SCL_LOW) = (SCL_LOW * Period(P_CLK)), where P_CLK is 4 ns. Reset time is 5.00 microseconds (100 kHz).	R/W	0x04E2

22.2.31 I2C_SCL Minimum High and Low Timing Register

This register programs the minimum high and low periods of the I2C_SCL signal when generated by the master interface. It is shadowed during boot loading, and can be reprogrammed before a chain operation without affecting the bus timing for the current EEPROM.

Register name: I2C_SCL_MIN_PERIOD Reset value: 0x03E8_0497	Register offset: 0x49350

Bits	7	6	5	4	3	2	1	0
31:24	SCL_MINH							
23:16	SCL_MINH							
15:08	SCL_MINL							
07:00	SCL_MINL							

Bits	Name	Description	Type	Reset Value
31:16	SCL_MINH	Count for I2C_SCL High Minimum Period Defines the minimum high period of the clock, from rising edge seen high to falling edge of I2C_SCL. This is a master-only parameter. The observed period can be shorter if other devices pull the clock low. Period(SCL_MINH) = (SCL_MINH * Period(P_CLK)), where P_CLK is 4 ns. Reset time is 4.00 microseconds.	R/W	0x03E8
15:00	SCL_MINL	Count for I2C_SCL Low Minimum Period Defines the minimum low period of the clock, from falling edge seen low to rising edge of I2C_SCL. This is a master-only parameter. The observed period can be longer if other devices pull the clock low. Period(SCL_MINL) = (SCL_MINL * Period(P_CLK)), where P_CLK is 4 ns. Reset time is 4.70 microseconds.	R/W	0x0497

22.2.32 I2C_SCL Low and Arbitration Timeout Register

This register programs the I2C_SCL low timeout and the Arbitration timeout. The arbitration timer period is relative to the MSDIV period, and the I2C_SCL low timeout period is relative to the USDIV period.

Register name: I2C_SCL_ARB_TIMEOUT Reset value: 0x6590_0033	Register offset: 0x49354

Bits	7	6	5	4	3	2	1	0
31:24	SCL_TO							
23:16	SCL_TO							
15:08	ARB_TO							
07:00	ARB_TO							

Bits	Name	Description	Type	Reset Value
31:16	SCL_TO	Count for I2C_SCL Low Timeout Period Defines the maximum amount of time for a slave device holding the I2C_SCL signal low. This timeout covers the period from I2C_SCL falling edge to the next I2C_SCL rising edge. Value 0x0 disables the timeout. Period(SCL_TO) = (SCL_TO * Period(USDIV)) where USDIV is the microsecond time defined in the I2C Time Period Divider Register. The reset value of this timeout is 26,000 microseconds (26 milliseconds).	R/W	0x6590
15:00	ARB_TO	Count for Arbitration Timeout Period Defines the maximum amount of time for the master interface to arbitrate for the bus before aborting the transaction. This timeout covers the period from master operation start (setting the START bit in the I2C Master Control Register) until the ACK/NACK is received from the external slave for the slave device address. A value of 0 disables the timeout. Period(ARB_TO) = (ARB_TO * Period(MSDIV)) where MSDIV is the millisecond time defined in I2C Time Period Divider Register. The reset value of this timeout is 51 milliseconds. However, this timeout is not active during the boot load sequence.	R/W	0x0033

22.2.33 I2C Byte/Transaction Timeout Register

This register programs the Transaction and Byte timeouts. The timer periods are relative to the USDIV period for the byte timeout, and relative to the MSDIV period for the transaction timeout.

Register name: I2C_BYTE_TRAN_TIMEOUT Reset value: 0x0000_0000	Register offset: 0x49358

Bits	7	6	5	4	3	2	1	0
31:24	BYTE_TO							
23:16	BYTE_TO							
15:08	TRAN_TO							
07:00	TRAN_TO							

Bits	Name	Description	Type	Reset Value
31:16	BYTE_TO	Count for Byte Timeout Period Defines the maximum amount of time for a byte to be transferred on the I2C bus. This covers the period from Start condition to next ACK/NACK, between two successive ACK/NACK bits, or from ACK/NACK to Stop/Restart condition. A value of 0 disables the timeout. Period(BYTE_TO) = (BYTE_TO * Period(USDIV)) where USDIV is the microsecond time defined in I2C Time Period Divider Register. This timeout is disabled on reset, and is not used during boot load.	R/W	0x0000
15:00	TRAN_TO	Count for Transaction Timeout Period Defines the maximum amount of time for a transaction on the I2C bus. This covers the period from Start to Stop. A value of 0 disables the timeout. Period(TRAN_TO) = (TRAN_TO * Period(MSDIV)) where MSDIV is the millisecond time defined in I2C Time Period Divider Register. This timeout is disabled on reset, and is not used during boot load.	R/W	0x0000

22.2.34 I2C Boot and Diagnostic Timer

This register programs a timer to timeout the boot load sequence, and can be used after boot load as a general purpose timer.

Register name: I2C_BOOT_DIAG_TIMER Reset value: 0x0000_0FA0	Register offset: 0x4935C

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

Bits	Name	Description	Type	Reset Value
31	FREERUN	Free Running Timer 0 = Timer is not automatically restarted when expires 1 = When timer expires, timer is restarted with same COUNT	R/W	0
30:16	Reserved	Reserved	R	0x0000
15:00	COUNT	Count for Timer Period Defines period for timer. Initial reset value is used for overall boot load timeout. During normal operation, this timer can be used for any general purpose timing. A value of 0 disables the timeout. Period(DTIMER) = (COUNT * Period(MSDIV)), where MSDIV is the millisecond period define in I2C Time Period Divider Register. Timer begins counting when this register is written. If this register is written while the counter is running, the timer is immediately restarted with the new COUNT, and the DTIMER/BLTO event is not generated. When the timer expires, either the BLTO or DTIMER event is generated, depending on whether the boot load sequence is active. If FREERUN is set to 1 when timer expires, then the timer is restarted immediately (the event is still generated), providing a periodic interrupt capability. The reset value for the boot load timeout is 4 seconds. If the boot load completes before the timer expires, the timer is set to zero (disabled).	R/W	0x0FA0

22.2.35 I2C Boot Load Diagnostic Progress Register

This register provides visibility of the register count and peripheral address during the boot load sequence.

Register name: I2C_BOOT_DIAG_PROGRESS Reset value: 0x0000_0000	Register offset: 0x493B8

Bits	7	6	5	4	3	2	1	0
31:24	REGCNT							
23:16	REGCNT							
15:08	PADDR							
07:00	PADDR							

Bits	Name	Description	Type	Reset Value
31:16	REGCNT	Register Count The number of registers remaining to load from the current EEPROM during the boot load sequence. This register is initialized to the count read from the first two bytes of the EEPROM after reset, or to the first two byte read after a boot chaining operation. The field counts down as each register address/data pair is read.	R	0x0000_0000
15:00	PADDR	Peripheral Address Value of current peripheral address used by the boot load sequence. This field is initialized to zero at reset, and increments as the boot load sequence progresses. If a chain operation is performed, this field is loaded with the new peripheral address from the I2C Boot Control Register (with the 3 LSBs set to zero).	R	0x0000_0000

This is a diagnostic register. Documentation is provided for informational purposes only. The function of this register is not guaranteed in future versions and usage thereof is not supported.

22.2.36 I2C Boot Load Diagnostic Configuration Register

This register provides visibility of boot sequence information.

Register name: I2C_BOOT_DIAG_CFG Reset value: 0x0000_0000	Register offset: 0x493BC

Bits	7	6	5	4	3	2	1	0
31:24	BOOTING	BDIS	PASIZE	PINC	Reserved			
23:16	Reserved							
15:08	Reserved							
07:00	Reserved	BOOT_ADDR						

Bits	Name	Description	Type	Reset Value
31	BOOTING	Booting 0 = Boot sequence not active 1 = Boot sequence in progress	R	0
30	BDIS	Boot Disabled 0 = Boot enabled 1 = Boot disabled	R	0
29	PASIZE	Peripheral Address Size 0 = 1-byte peripheral address 1 = 2-byte peripheral address Note: This is the state of the I2C_BOOT_CNTRL.PSIZE field at boot load start or after a chain operation.	R	0
28	PINC	Page Increment 0 = Page increment disabled 1 = Page increment enabled	R	0
27:07	Reserved	Reserved	R	0x000
06:00	BOOT_ADDR	Boot Device Address Current value of the boot device address in use by the boot load sequence. This value is incremented during the bootload if the page increment feature is enabled.	R	0x00

⚠ This is a diagnostic register. Documentation is provided for informational purposes only. The function of this register is not guaranteed in future versions and usage thereof is not supported.

23. GPIO Registers

23.1 Register Map

Table 99: GPIO Register Map

Offset	Register Name	See
0x4A000	GPIO0_DATA	GPIO 0 Data Register
0x4A004	GPIO0_CNTRL	GPIO 0 Control Register

23.2 Register Descriptions

This section describes the GPIO registers. These registers are reset by a fundamental reset.

23.2.1 GPIO 0 Data Register

This register presents the data on the GPIO Interface when configured as outputs, and defines the data on the GPIO Interface when configured as inputs.

Register name: GPIO0_DATA Reset value: Undefined	Register offset: 0x4A000

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

Bits	Name	Description	Type	Reset Value
0:15	GPIO_DATA_IN[15:0]	Value from the input port. This field indicates the current data on the GPIO Interface. This may be the same data that is being driven on the output port. Note: GPIO_DATA_IN[0] represents GPIO[0] data.	R	Undefined
16:31	GPIO_DATA_OUT[15:0]	Value driven on the output port. This field defines each bit of the output port. If the port is configured as an output with open-drain capability, when these bits are set to one the output is tristated (open-drain); when set to 0 the output is driven low. If the port is configured as an output, the value written to this field is driven on the output port. Note: When the GPIO port is configured as an output, only GPIO[15:3] can be programmed through software; GPIO[2:0] are used as interrupts (for more information, see GPIO Signals). • Note: GPIO_DATA_OUT[15] controls GPIO[15] data.	R/WS	0xFFFF

23.2.2 GPIO 0 Control Register

This register controls the direction and output configuration of the GPIO signals. The direction and configuration of each GPIO signal can be configured individually.

Register name: GPIO0_CNTRL Reset value: 0x0000_0000	Register offset: 0x4A004

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

Bits	Name	Description	Type	Reset Value
0:15	GPIO_DIR [15:0]	Direction of the GPIO signals 0 = Input, tri-stated output 1 = Output with GPIO_CFG = 0	R/WS	0x0000
16:31	GPIO_CFG [15:0]	0 = Standard output 1 = Reserved	R/WS	0x0000

24. SerDes Control Registers

24.1 Register Map

There are two sets of SerDes control registers: one for the PCIe SerDes, and the other for the S-RIO SerDes. Therefore, each row in Table 101 is associated with two registers, and these two registers have the same offset but different base address, where the base address is listed in Table 100.

Table 100: SerDes Register Base Address

PCIe SerDes Register Base Address	S-RIO SerDes Register Base Address
0x4C000	0x4E000

Table 101 summarizes the SerDes control registers.

Table 101: SerDes Register Map[a]

Offset	Register Name	See
0x1000 +=0x400	SERDES_LANEn_DIG_TX_OVRD_IN	SerDes Tx Control Register
0x100C +=0x400	SERDES_LANEn_DIG_RX_OVRD_IN	SerDes Rx Control Register

[a] Offsets relative to base address in Table 100.

24.2 Register Descriptions

This section describes the SerDes control registers. These registers are reset by a device reset.

24.2.1 SerDes Tx Control Register

This register provides SerDes override options. For more information on programming this register, see Bit Error Rate Testing (BERT).

Register name: SERDES_LANE{0..3}_LANEn_DIG_TX_OVRD_IN Reset value: 0x0000_0000	Register offset: 0x1000+=0x400

Bits	7	6	5	4	3	2	1	0
31:24	Reserved							
23:16	Reserved							
15:08	Reserved						EN_L	Reserved
07:00	Reserved	CM_EN	TX_EN	DATA_EN	Reserved	INVERT	LOOPBK_EN	HALF_RATE

Bits	Name	Description	Type	Reset value
31:10	Reserved	Reserved	R	0x0
9	EN_L	Enable low override values for all inputs controlled by bits [8:0].	R/W	0x0
8:7	Reserved	Reserved	R	0x0
6	CM_EN	Override value for transmitter common mode enable.	R/W	0x0
5	TX_EN	Override value for transmitter datapath enable.	R/W	0x0
4	DATA_EN	Override value for transmitter driver enable.	R/W	0x0
3	Reserved	Reserved	R	0x0
2	INVERT	Override value for transmit data polarity inversion enable.	R/W	0x0
1	LOOPBK_EN	Override value for Tx/Rx loopback mode enable.	R/W	0x0
0	HALF_RATE	Override value for half rate mode. 0 = All other speeds 1 = Use PCIe 2.5 Gbaud and S-RIO 1.25 Gbaud link speeds	R/W	0x0

24.2.2 SerDes Rx Control Register

This register provides SerDes override options. For more information on programming this register, see PCIe 10-bit PCS Slave Loopback or S-RIO Line Loopback.

Register name: SERDES_LANE{0..3}_LANEn_DIG_RX_OVRD_IN Reset value: 0x0000_0000	Register offset: 0x100C+=0x400

Bits	7	6	5	4	3	2	1	0
31:24	Reserved							
23:16	Reserved							
15:08	Reserved					EN	Reserved	
07:00	Reserved	TERM_EN	Reserved	ALIGN_EN	DATA_EN	PLL_EN	Reserved	INVERT

Bits	Name	Description	Type	Reset value
31:11	Reserved	Reserved	R	0x0
10	EN	Enable override values for all inputs controlled by this register.	R/W	0x0
9:7	Reserved	Reserved	R	0x0
6	TERM_EN	Override value for receiver termination enable.	R/W	0x0
5	Reserved	Reserved	R	0x0
4	ALIGN_EN	Override value for receiver symbol alignment enable.	R/W	0x0
3	DATA_EN	Override value for receiver datapath enable.	R/W	0x0
2	PLL_EN	Override value for receiver PLL enable.	R/W	0x0
1	Reserved	Reserved	R	0x0
0	INVERT	Override value for receive data polarity inversion enable.	R/W	0x0

25. Top-Level Registers

Topics discussed include the following:

- Register Map
- Register Descriptions

25.1 Register Map

For registers defined for multiple SerDes lanes, such as PCIe SerDes Transmitter Control, register offset between two adjacent lane is 32; that is, in 0x48040 += 2; register offset for lane 0 has address 0x48040, += denotes the register offset between lanes and 20 is in hexidecimal.

Table 102: SerDes Register Map

Offset	Register Name	See
0x48000	DEVSTAT	Device Status Register
0x48004	DEVCTL	Device Control Register
0x48008	CLK_GATE	Clock Gating Register
0x4800C	JTAG_ID	JTAG ID Register
0x48200	PC_TX_CTL	PCIe SerDes Transmitter Control
0x48220	PC_TX_CTL_2	PCIe SerDes Transmitter Control 2
0x48800 += 20	SR_TX_CTL{0..3}	S-RIO SerDes Transmitter Control of Lanes {0..3}

25.2 Register Descriptions

25.2.1 Device Status Register

Register name: DEVSTAT Reset value: Undefined	Register offset: 0x48000

Bits	7	6	5	4	3	2	1	0
31:24	SP_DEVID	SP_HOST	SP_SWAP_TX	SP_SWAP_RX	CLKMOD	I2C_MA	SR_BOOT	I2C_SEL
23:16	RESERVED			I2C_SA[3:0]				I2C_DISABLE
15:8	RESERVED		CLKSEL[1:0]		STRAP_RATE[2:0]			RESERVED
7:0	RESERVED						PCRDY	RESERVED

Bits	Name	Description	Type	Reset value
31	SP_DEVID	S-RIO deviceID This is the sampled value of the SP_DEVID power-up signal after a fundamental reset.	R	Undefined
30	SP_HOST	S-RIO host This is the sampled value of the SP_HOST power-up signal after a fundamental reset.	R	Undefined
29	SP_SWAP_TX	S-RIO SerDes transmitter lane swap This is the sampled value of the SP_SWAP_TX power-up signal after a fundamental reset.	R	Undefined
28	SP_SWAP_RX	S-RIO SerDes receiver lane swap This is the sampled value of the SP_SWAP_RX power-up signal after a fundamental reset.	R	Undefined
27	CLKMOD	Clock Mode This is the sampled value of the CLKMOD power-up signal after a fundamental reset.	R	Undefined
26	I2C_MA	I^{2}C Mode This is the sampled value of the I2C_MA power-up signal after a fundamental reset.	R	Undefined
25	SR_BOOT	S-RIO Boot This is the sampled value of the SR_BOOT power-up signal after a fundamental reset.	R	Undefined

(Continued)

Bits	Name	Description	Type	Reset value
24	I2C_SEL	I^{2}C Pin Select This is the sampled value of the I2C_SEL power-up signal after a fundamental reset.	R	Undefined
23:21	RESERVED	RESERVED	R	0x0
20:17	I2C_SA	I2C Slave Address This is the sampled value of the I2C_SA power-up signal after a fundamental reset.	R	Undefined
16	I2C_DISABLE	Disable I^{2}C register loading after reset This is the sampled value of the I2C_DISABLE power-up signal after a fundamental reset.	R	Undefined
15:14	RESERVED	RESERVED	R	0x0
13:12	CLKSEL	Clock Frequency Select This is the sampled value of the CLKSEL[1:0] power-up signals after a fundamental reset.	R	Undefined
11:9	STRAP_RATE	S-RIO rates This is the sampled value of the STRAP_RATE[2:0] power-up signals after a fundamental reset.	R	Undefined
8:2	RESERVED	RESERVED	R	Undefined
1	PCRDY	PCIe MAC Ready During a fundamental reset, PCIe MAC sets this bit when it is ready for a Tsi721 PCIe configuration space register update. When boot from I2C or S-RIO, they must poll this bit before starting configuration. When boot from EEPROM, Tsi721 starts the EEPROM download after this bit is set to 1.	R	Undefined
0	RESERVED	RESERVED	R	0x0

25.2.2 Device Control Register

Register name: DEVCTL Reset value: Undefined	Register offset: 0x48004

Bits	7	6	5	4	3	2	1	0
31:24	RESERVED							
23:16	RESERVED				SR_RST_MODE[3:0]			
15:8	RESERVED							
7:0	RESERVED			MECS_O	RESERVED	SRBOOT_CMPL	PCBOOT_CMPL	FRST

Bits	Name	Description	Type	Reset value
31:20	RESERVED	RESERVED	R	0x0
19:16	SR_RST_MODE	S-RIO MAC Reset Control when receiving four consecutive S-RIO reset control symbols, or GB_1p25_EN, GB_2p5_EN, GB_3p125_EN, GB_5p0_EN, or GB_6p25_EN in the RapidIO Port Control 2 CSR • 0b0000 = Perform a device-wide hot reset. • 0b0001 = Perform a S-RIO MAC core logic reset. All device registers (including those from S-RIO MAC) retain their values without being reset. • All other values are reserved. Note: When SR_RST_MODE is 0b0001, a reset request can be issued to the Tsi721 only when there are no packets being exchanged on the RapidIO link.	R/WS	0x0
15:5	RESERVED	RESERVED	R	0x0
4	MECS_O	MECS signal direction control 0 = MECS signal is an input 1 = MECS signal is an output	R/WS	0x0
3	RESERVED	RESERVED	R	0x0
2	SRBOOT_CMPL	S-RIO boot load completed When boot from S-RIO (SR_BOOT signal high), the Tsi721 automatically toggles this bit from 0 to 1 after a fundamental reset; otherwise, software is responsible to toggle this bit. When this bit transitions from 0 to 1, all RE type bits within the S-RIO MAC become read only, Tsi721 starts S-RIO link training and enters normal operation after training.	R/WS	Undefined

(Continued)

Bits	Name	Description	Type	Reset value
1	PCBOOT_CMPL	PCIe boot load completed 0 = All of Tsi721's logic except the I2C interface remains in a quasi-reset state, and Tsi721 responds to all type 0 configuration requests with configuration-request-retry-status completions. When boot Tsi721 from EEPROM, Tsi721 resets this bit to 0 after a fundamental reset, and automatically sets this bit after it finishes a boot whether EEPROM boot succeeded or failed. When boot Tsi721 from I2C master, Tsi721 resets this bit to 0 after a fundamental reset, and software should set this bit after it finishes a Tsi721 boot procedure. When boot Tsi721 from PCIe root complex, Tsi721 resets this bit to 0 after a fundamental reset, and Tsi721 automatically sets this bit to 1 when PCRDY of Device Status Register is 1.	R/WS	Undefined
0	FRST	Fundamental Reset 1 = Initiate a device fundamental reset. Tsi721 automatically clears this bit to zero.	R/W1S	0x0

25.2.3 Clock Gating Register

Register name: CLK_GATE Reset value: 0x0000_0000	Register offset: 0x48008

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

Bits	Name	Description	Type	Reset value
31:16	RESERVED	RESERVED	R	0x0
15:8	CLK_GATE_SMSGCH	Tsi721 Channel Clock Gating 1 = Logic for the Messaging Engine inbound DMA channel x and Messaging Engine outbound DMA channel x are all clock gated.	R/WS	0x0
7:0	CLK_GATE_BDMACH	Tsi721 Channel Clock Gating 1 = Logic for BDMA channel x are all clock gated.	R/WS	0x0

25.2.4 JTAG ID Register

Register name: JTAG_ID Reset value: 0x180A_B067	Register offset: 0x4800C

Bits	7	6	5	4	3	2	1	0
31:24	VERSION[3:0]				PART[15:12]			
23:16	PART[11:4]							
15:8	PART[3:0]				MANU_ID[10:7]			
7:0	MANU_ID[6:0]							RESERVED

Bits	Name	Description	Type	Reset value
31:28	VERSION	Version Reset value 0x1 = Tsi721 Version (Revision) 1 0x0 = Tsi721 Version (Revision) 0	R	0x1
27:12	PART	Part number 0x80AB = Tsi721	R	0x80AB
11:1	MANU_ID	Manufacture ID	R	0x33
0	RESERVED	RESERVED	R	0x1

25.2.5 PCIe SerDes Transmitter Control

Register name: PC_TX_CTL Reset value: 0x0015_1F33	Register offset: 0x48200

Bits	7	6	5	4	3	2	1	0
31:24	RESERVED							
23:16	RESERVED			TX_COEF35_G2[4:0]				
15:8	RESERVED			TX_COEF60_G2[4:0]				
7:0	RESERVED		TX_AMP_FULL[5:0]					

Bits	Name	Description	Type	Reset value
31:21	RESERVED	RESERVED	R	0x0
20:16	TX_COEF35_G2	This field controls the PCIe SerDes transmitter pre-emphasis level adjustment when -3.5dB adjustment is used in Gen2 mode.	R/WS	0x15
15:13	RESERVED	RESERVED	R	0x0
12:8	TX_COEF60_G2	This field controls the PCIe SerDes transmitter pre-emphasis level adjustment when -6dB adjustment is used in Gen2 mode.	R/WS	0x1F
7:6	RESERVED	RESERVED	R	0x0
5:0	TX_AMP_FULL	This field controls the adjustment of the PCIe SerDes transmitter output amplitude when full swing mode is used. The peak-to-peak differential amplitude before taking into account of package loss is equal to 1.33 x TX_AMP/64 V. Amplitude below 400 mV peak-to-peak (TX_AMP =19) is recommended for margin only, and should not be used for normal operation.	R/WS	0x33

25.2.6 PCIe SerDes Transmitter Control 2

Register name: PC_TX_CTL_2 Reset value: 0x0015_0033	Register offset: 0x48220

Bits	7	6	5	4	3	2	1	0
31:24	RESERVED							
23:16	RESERVED			TX_COEF35_G1[4:0]				
15:8	RESERVED							
7:0	RESERVED		TX_AMP_LOW[5:0]					

Bits	Name	Description	Type	Reset value
31:21	RESERVED	RESERVED	R	0x0
20:16	TX_COEF35_G1	This field controls the PCIe SerDes transmitter pre-emphasis level adjustment when -3.5dB adjustment is used in Gen1 mode.	R/WS	0x15
15:6	RESERVED	RESERVED	R	0x0
5:0	TX_AMP_LOW	This field controls the adjustment of the PCIe SerDes transmitter output amplitude in low swing mode. The peak-to-peak differential amplitude before taking into account of package loss is equal to 1.33 x TX_AMP/64 V. Amplitude below 400mV peak-to-peak (TX_AMP =19) is recommended for margin only, and should not be used for normal operation.	R/WS	0x33

25.2.7 S-RIO SerDes Transmitter Control of Lanes {0..3}

This register defines S-RIO SerDes transmitter control. Register Y (Y = 0 – 3) is assigned to lane Y.

Register name: SR_TX_CTL{0..3} Reset value: 0x0000_1F33	Register offset: 0x48800 += 20

Bits	7	6	5	4	3	2	1	0
31:24	LB_EN	RESERVED						
23:16	RESERVED							
15:8	RESERVED			TX_COEF[4:0]				
7:0	RESERVED		TX_AMP[5:0]					

Bits	Name	Description	Type	Reset value
31	LB_EN	This bit controls S-RIO SerDes loopback. 1 = SerDes transmitter output is looped back to SerDes receiver input.	R/WS	0x0
30:13	RESERVED	RESERVED	R	0x0
12:8	TX_COEF	This field controls S-RIO SerDes transmitter pre-emphasis level adjustment. The adjustment before taking into account of package loss is expressed as (TX_COEF+1)/4 dB, in incremental of 0.25dB with a maximum of 8dB.	R/WS	0x1F
7:6	RESERVED	RESERVED	R	0x0
5:0	TX_AMP	This field controls the adjustment of the S-RIO SerDes transmitter output amplitude. The peak-to-peak differential amplitude before taking into account of package loss is equal to 1.33 x TX_AMP/64 V. Amplitude below 400mV peak-to-peak (TX_AMP =19) is recommended for margin only, and should not be used for normal operation.	R/WS	0x33

Appendices

A. ackID Synchronization

Topics discussed include the following:

- Problem Overview
- ackID Resynchronization Processes

A.1 Problem Overview

The issue occurs when an ackID mismatch occurs between two switches; for example, one switch on a motherboard and another on a mezzanine card. After a hot extraction and then an insertion of another card, the expected ackIDs of the two connected ports may be different. The switch on the motherboard must be synchronized with the switch on the mezzanine card. In this example, the ackID used by the mezzanine board will be zero (as it is immediately after reset); however, the ackIDs of the two switches usually can be arbitrary values.

The *RapidIO Specification (Rev. 2.1)* offers this explanation:

> "...the system host must make the ackID values for both link partners match in order to begin sending packets. In order to accomplish this, the system host generates a link-request/link-status to the attached device to obtain its expected receiver value using the Port n Link Maintenance Request and Response CSRs. It can then set its transmit ackID value to match. Next, the system host generates a Maintenance write operation to set the attached device's Port n Local ackID Status CSR to set the transmit ackID value to match the receive ackID value in the system host. Upon receipt of the maintenance write, the attached device sets its transmit ackID value as instructed, and generates the maintenance response using the new value. Packet transmission can now proceed normally."

The issue that occurs with this method is that the motherboard may not know which port of the mezzanine switch it is connected. Without this information, the host cannot set the ackID of the mezzanine switch without potentially resetting the ackID of each port and impacting traffic that is flowing on the ports.

A.2 ackID Resynchronization Processes

A.2.1 Solution with Tsi721 as Host

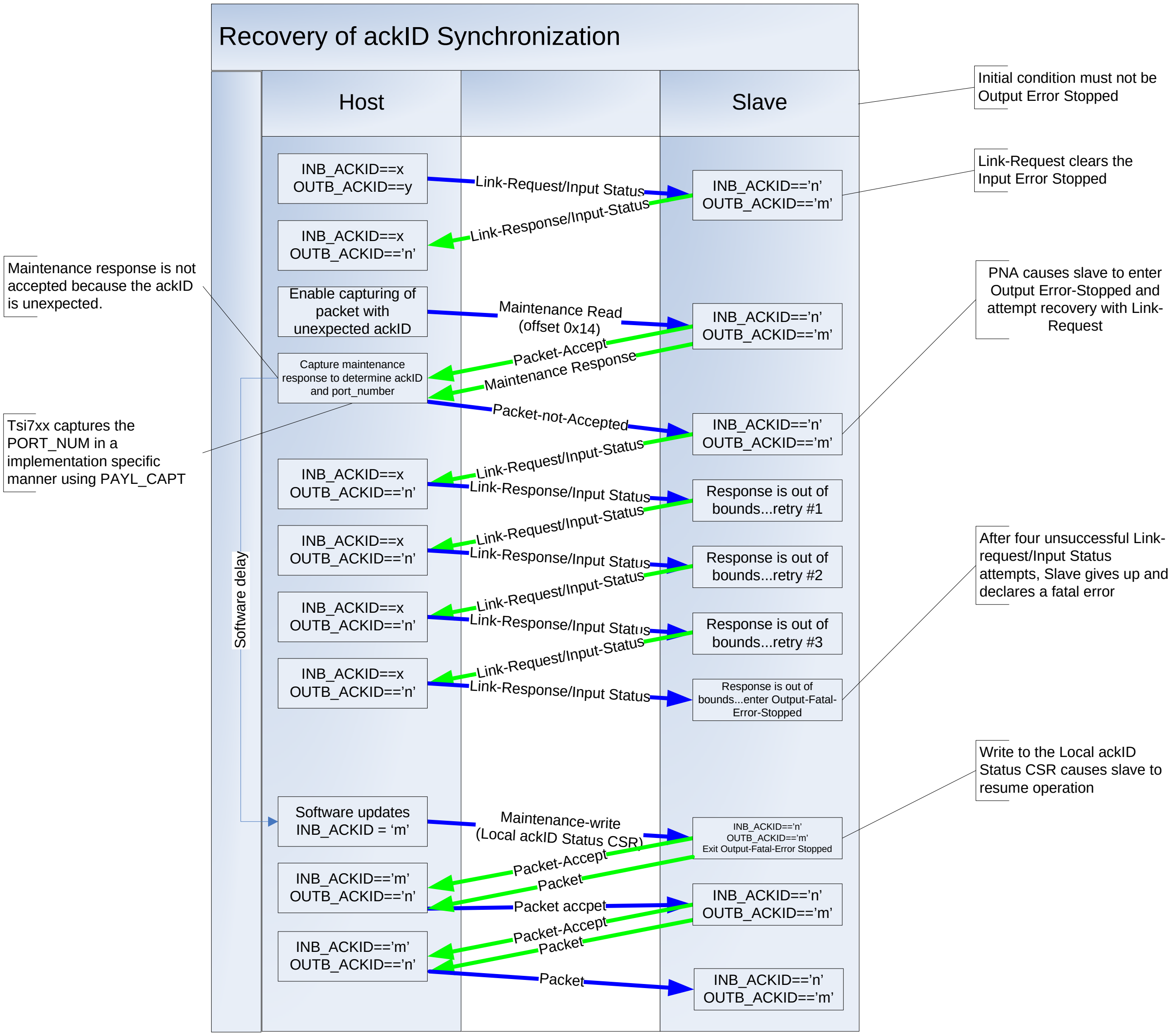
As the host, the Tsi721 takes these steps when the ackID of the link partner is unknown:

1. Software issues a link-request/input-status control symbol to its link partner using the RapidIO Port Link Maintenance Request CSR and writes 0b100 to the CMD field.
2. Tsi721 captures the received link-response, which indicates the ackID the link partner expects in RapidIO Port Link Maintenance Response CSR.ACK_ID_STAT.
3. Software sets Tsi721's Outbound ackID to the ackID received in the link-response in RapidIO Port Local ackID Status CSR by writing to the RapidIO Port Local ackID Status CSR.OUTB_ACKID.
4. Software sets RapidIO PLM Port Implementation Specific Control Register.PAYL_CAPT so that the RapidIO Port Packet/Control Symbol Error Capture CSR 0 capture both the ackID and payload of a maintenance read.

5. Software enables the “Received packet with unexpected ackID enable” in the RapidIO Port Error Rate Enable CSR.PKT_ILL_ACKID_EN and unlocks the RapidIO Port Packet/Control Symbol Error Capture CSR 0 (and related CSRs) by writing zero to the RapidIO Port Error Detect CSR.
6. Software sends a maintenance read request to its link partner to offset 0x14 (RapidIO Source Operation CAR). The returned response may be discarded due to its ackID, but if it is, the RapidIO Port Error Detect CSR.PKT_ILL_ACKID is set and the packet is captured in the RapidIO Port Packet/Control Symbol Error Capture CSR 0 (and subsequent registers).
7. Software obtains the ackID from the RapidIO Port Packet/Control Symbol Error Capture CSR 0 (bits 0–4 or 5 depending on IDLE1/IDLE2 selection), and sets the expected inbound ackID in the RapidIO Port Local ackID Status CSR.INB_ACKID.
8. Software obtains the port number to which it is connected from the captured packet in RapidIO Port Packet/Control Symbol Error Capture CSR 0. The location of the PORT_NUM field depends on the Transport Type field:
   a. For small transport systems (that is, srcID and destID are one byte each), the PORT_NUM field is reported in bytes 0 and 1 of RapidIO Port Packet Error Capture CSR 3.
   b. For large transport systems (that is, srcID and destID are two bytes each), the PORT_NUM field is reported in bytes 2 and 3 of RapidIO Port Packet Error Capture CSR 3.
9. Using the port number information, software calculates the address of the RapidIO Port Local ackID Status CSRand writes to that register to clear the port-error state of the link partner which will have been triggered by the packet-not-accepted and subsequent link-request/link-response failures that occurred due to mismatched ackIDs.

At this point, the ackIDs are in sync.

Figure 64: Recovery with Tsi721 Acting as Host


Recovery of ackID Synchronization
Host
Slave
Software delay
INB_ACKID==x
OUTB_ACKID==y
Link-Request/Input Status
INB_ACKID=='n'
OUTB_ACKID=='m'
Link-Response/Input-Status
INB_ACKID==x
OUTB_ACKID=='n'
Enable capturing of packet with unexpected ackID
Maintenance Read (offset 0x14)
INB_ACKID=='n'
OUTB_ACKID=='m'
Packet-Accept
Maintenance Response
Capture maintenance response to determine ackID and port_number
Packet-not-Accepted
INB_ACKID=='n'
OUTB_ACKID=='m'
Link-Request/Input-Status
Link-Response/Input Status
INB_ACKID==x
OUTB_ACKID=='n'
Response is out of bounds...retry #1
Link-Request/Input-Status
Link-Response/Input Status
INB_ACKID==x
OUTB_ACKID=='n'
Response is out of bounds...retry #2
Link-Request/Input-Status
Link-Response/Input Status
INB_ACKID==x
OUTB_ACKID=='n'
Response is out of bounds...retry #3
Link-Request/Input-Status
Link-Response/Input Status
INB_ACKID==x
OUTB_ACKID=='n'
Response is out of bounds...enter Output-Fatal-Error-Stopped
Software updates INB_ACKID = 'm'
Maintenance-write (Local ackID Status CSR)
INB_ACKID=='n'
OUTB_ACKID=='m'
Exit Output-Fatal-Error Stopped
Packet-Accept
Packet
INB_ACKID=='m'
OUTB_ACKID=='n'
Packet accpet
INB_ACKID=='n'
OUTB_ACKID=='m'
Packet-Accept
Packet
INB_ACKID=='m'
OUTB_ACKID=='n'
Packet
INB_ACKID=='n'
OUTB_ACKID=='m'

Initial condition must not be Output Error Stopped

Link-Request clears the Input Error Stopped

Maintenance response is not accepted because the ackID is unexpected.

PNA causes slave to enter Output Error-Stopped and attempt recovery with Link-Request

Tsi7xx captures the PORT_NUM in a implementation specific manner using PAYL_CAPT

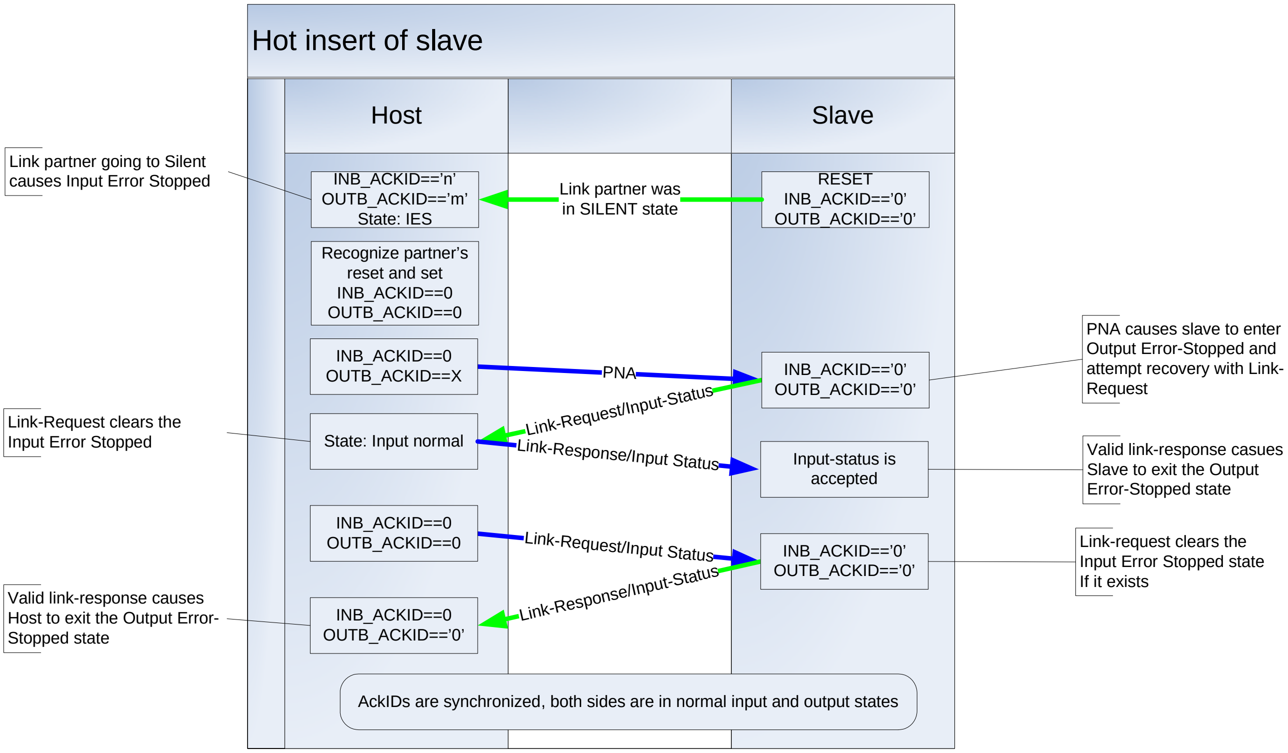
After four unsuccessful Link-request/Input Status attempts, Slave gives up and declares a fatal error

Write to the Local ackID Status CSR causes slave to resume operation

If the Slave is in Output Error Stopped (OES) state at the beginning of the process, the Tsi721 will not be able to read offset 0x14 because the Slave will be unable to return the response. To clear the OES state, Tsi721 must send a link-response to the Slave. This can be completed by having Tsi721 send a packet-not-accepted (PNA) control symbol. However, because the ackIDs are not in synchronization, the Slave will reject the ackID_status provided by the Tsi721 in the link-response and will proceed to the Port-Error state from which the only recovery is to write to the RapidIO Port Local ackID Status CSR or to force the slave to reset. Although writing the register may be preferable, the address of that register is unknown unless the Host knows the port of the Slave to which it is connected.

If the link partner's ackIDs are known to be zero (for example if it is known that the link partner was reset, its ackIDs are guaranteed to be zero), the solution is simpler as displayed in Figure 65.

Figure 65: Recovery with Tsi721 Acting as Host and ackIDs are Known


Hot insert of slave
Host
Slave
Link partner going to Silent causes Input Error Stopped
INB_ACKID=='n' OUTB_ACKID=='m' State: IES
Link partner was in SILENT state
RESET INB_ACKID=='0' OUTB_ACKID=='0'
Recognize partner's reset and set INB_ACKID==0 OUTB_ACKID==0
INB_ACKID==0 OUTB_ACKID==X
PNA
INB_ACKID=='0' OUTB_ACKID=='0'
PNA causes slave to enter Output Error-Stopped and attempt recovery with Link-Request
Link-Request/Input-Status
Link-Request clears the Input Error Stopped
State: Input normal
Link-Response/Input Status
Input-status is accepted
Valid link-response casues Slave to exit the Output Error-Stopped state
INB_ACKID==0 OUTB_ACKID==0
Link-Request/Input Status
INB_ACKID=='0' OUTB_ACKID=='0'
Link-request clears the Input Error Stopped state If it exists
Valid link-response causes Host to exit the Output Error-Stopped state
Link-Response/Input-Status
INB_ACKID==0 OUTB_ACKID=='0'
AckIDs are synchronized, both sides are in normal input and output states

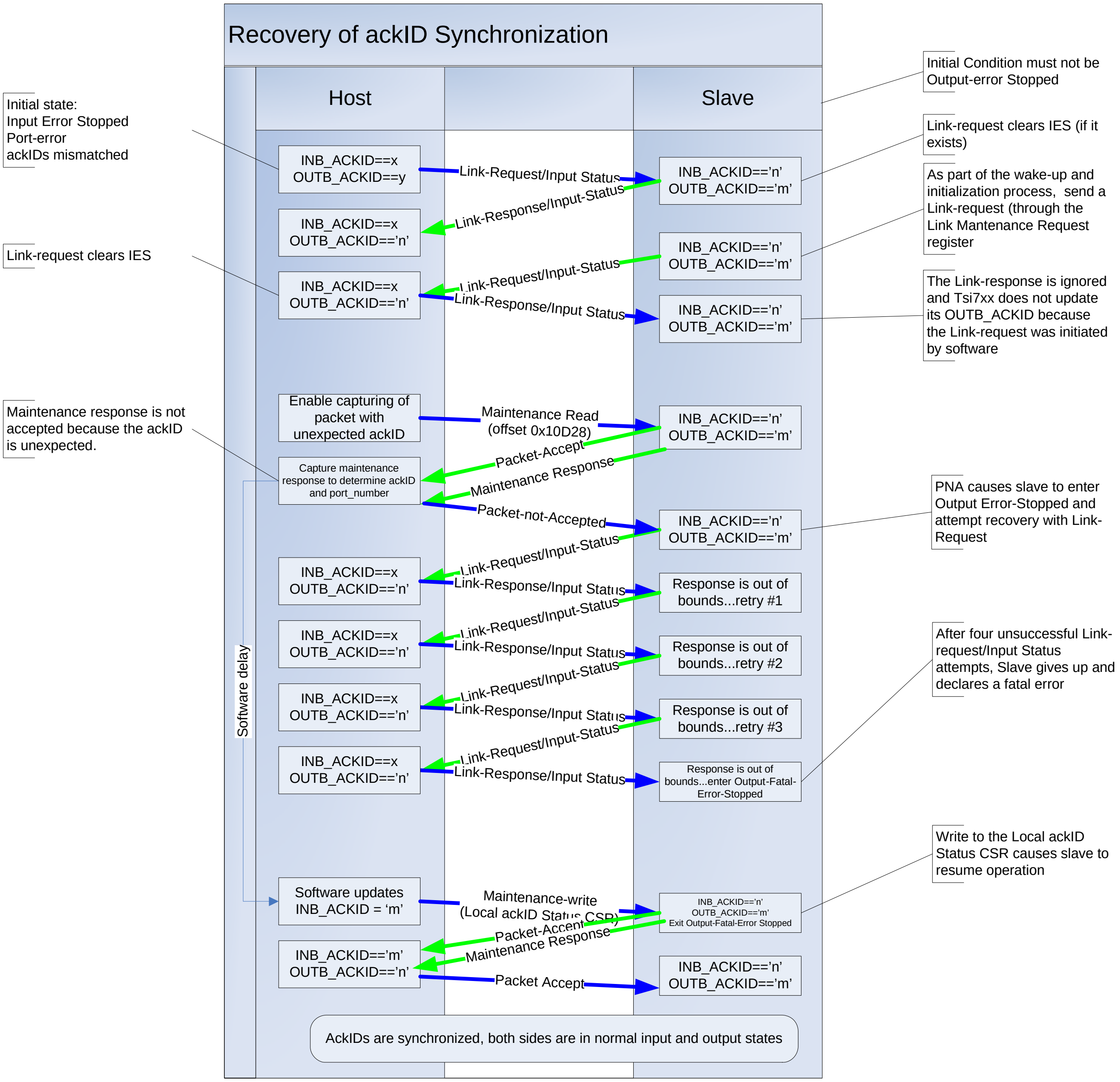
A.2.2 Solution with Tsi721 Not as Host

A.2.2.1 Host Has Software Error Recovery

Uses the same process as described in Solution with Tsi721 as Host except for these differences:

1. The Maintenance read is issued to RapidIO Port IP Prescalar for SRV_CLK Register which provides the port-number of the Tsi721 to which the host is attached in a way that allows capture registers to operate in the mode specified by *RapidIO Specification (Rev. 2.1)*.
2. Tsi721 as part of its software initialization must issue a link-request/input-status command through its RapidIO Port Link Maintenance Request CSR by writing 0b100 to the CMD field.

Figure 66: Recovery with Tsi721 Acting as Slave


Recovery of ackID Synchronization
Host
Slave
Software delay
INB_ACKID==x
OUTB_ACKID==y
Link-Request/Input Status
INB_ACKID=='n'
OUTB_ACKID=='m'
Link-Response/Input-Status
INB_ACKID==x
OUTB_ACKID=='n'
INB_ACKID=='n'
OUTB_ACKID=='m'
Link-Request/Input-Status
INB_ACKID==x
OUTB_ACKID=='n'
Link-Response/Input Status
INB_ACKID=='n'
OUTB_ACKID=='m'
Enable capturing of packet with unexpected ackID
Maintenance Read (offset 0x10D28)
INB_ACKID=='n'
OUTB_ACKID=='m'
Packet-Accept
Maintenance Response
Capture maintenance response to determine ackID and port_number
Packet-not-Accepted
INB_ACKID=='n'
OUTB_ACKID=='m'
Link-Request/Input-Status
INB_ACKID==x
OUTB_ACKID=='n'
Link-Response/Input Status
Response is out of bounds...retry #1
Link-Request/Input-Status
INB_ACKID==x
OUTB_ACKID=='n'
Link-Response/Input Status
Response is out of bounds...retry #2
Link-Request/Input-Status
INB_ACKID==x
OUTB_ACKID=='n'
Link-Response/Input Status
Response is out of bounds...retry #3
Link-Request/Input-Status
INB_ACKID==x
OUTB_ACKID=='n'
Link-Response/Input Status
Response is out of bounds...enter Output-Fatal-Error-Stopped
Software updates INB_ACKID = 'm'
Maintenance-write (Local ackID Status CSR)
INB_ACKID=='n'
OUTB_ACKID=='m'
Exit Output-Fatal-Error Stopped
Packet-Accept
Maintenance Response
INB_ACKID=='m'
OUTB_ACKID=='n'
Packet Accept
INB_ACKID=='n'
OUTB_ACKID=='m'
AckIDs are synchronized, both sides are in normal input and output states

Initial state:
Input Error Stopped
Port-error
ackIDs mismatched

Link-request clears IES

Maintenance response is not accepted because the ackID is unexpected.

Initial Condition must not be Output-error Stopped

Link-request clears IES (if it exists)

As part of the wake-up and initialization process, send a Link-request (through the Link Mantenance Request register

The Link-response is ignored and Tsi7xx does not update its OUTB_ACKID because the Link-request was initiated by software

PNA causes slave to enter Output Error-Stopped and attempt recovery with Link-Request

After four unsuccessful Link-request/Input Status attempts, Slave gives up and declares a fatal error

Write to the Local ackID Status CSR causes slave to resume operation

Similar to the Solution with Tsi721 as Host process, the Slave (in this case Tsi721) cannot be in the Output Error Stopped (OES) state when ackID synchronization recovery begins. If Tsi721 is OES, once the link is initialized, the Tsi721 will resume the link-response timers which will eventually expire and cause Tsi721 to transmit a link-request. Because the Host's ackIDs are out of synchronization, Tsi721 will reject the ackID_status in the link-response and proceed to the Port Error state from which the only recovery is to write to the RapidIO Port Local ackID Status CSR or to force the slave to reset. Although writing the register may be preferable, the address of that register is unknown unless the Host knows the port of the Slave to which it is connected or the link is operating in 4x mode in which case the port number must be zero.

A.2.2.2 Host Does Not Have Software Error Recovery

The host takes the following actions:

1. Host sends four link-request/reset control symbols which cause the Tsi721 to take one of these actions.
   - Reset the port which returns the inbound and outbound ackIDs to zero.
   - Reset the Tsi721 which returns the inbound and outbound ackIDs to zero.
   - Generate an interrupt which must cause another host to reset the port to return the inbound and outbound ackIDs to zero.
   - Ignore the reset request; this is an incorrect system configuration if loss of ackID synchronization may occur.
2. Host resets its port which returns its ackIDs to zero.

   At this point, the ackIDs are in sync.

B. Disabling the S-RIO Port

Topics discussed include the following:

B.1 Methods of Disabling the S-RIO Port

The S-RIO port can be disabled and packets discarded using the following registers:

- RapidIO Port Control CSR.PORT_LOCKOUT
- RapidIO Port Control CSR.PORT_DIS
- RapidIO PLM Port Event Status Register.DLT
- RapidIO Port Error and Status CSR.OUTPUT_FAIL

If the S-RIO port is disabled using one of the above register options, a soft reset is required to re-enable the port (for more information, see How to Re-enable the S-RIO Port?).

B.2 What Happens When the S-RIO Port is Disabled?

If the port is disabled using one of the methods listed above, Tsi721 takes the following actions:

- The PBMe discards all outstanding unacknowledged packets.
- The PBMe discards all packets enqueued from the Messaging Engine, Block DMA Engine, Mapping Engine, or LLMe and continues to discard new packets while the port is disabled.
- The ingress state machines operate as follows:
  - While in the "normal" state, a packet-not-accepted (PNA) control symbol is transmitted for any received packet and the state machine transitions to the input error-stopped state and waits for a link-request.
  - While in the input error-stopped state, received packets are ignored until a link-request is received that causes the state machine to return to the "normal" state and transmit a link-response.
  - While in Input Retry-stopped state, received packets are ignored until a Restart-from-Retry control symbol is received that causes the state machine to return to the "normal" state.
- The egress state machines operate as follows:
  - Packet Acknowledgement timers are disabled to prevent timeouts.
  - Received Packet Acknowledgements are handled as an Unexpected packet acknowledge because there are no outstanding unacknowledged packets once the port is disabled. This causes the state machine to enter the output error-stopped state and if no link-request is outstanding, a link-request is transmitted. Tsi721 remains in the output error-stopped state until the reset occurs.

- If the port is disabled while a packet is being transmitted, the packet's data stream is abruptly halted and the packet is terminated with a link-request control symbol. Any Link-response received for this link-request is ignored and if no link-response is received the link-request is not resent.
- If the port is disabled while the link is idle and there is no outstanding link-request, a link-request is sent. Any link-response received for this link-request is ignored and if no link-response is received the link-request is not resent.
- For link-requests sent before the port was disabled:
  - Received link-responses have their ackID compared against the "most-recent+1" ackID and if they are not equal, a link-request is resent. After four attempts, the RapidIO Port Error and Status CSR.PORT_ERR status bit is set.
  - If a link-response is not received before the response timer expires, the link-request is resent. After four attempts, the RapidIO Port Error and Status CSR.PORT_ERR status bit is set.

B.3 How to Re-enable the S-RIO Port?

To re-enable the S-RIO port after it has been disabled by the setting of one of the control or status bits listed in Methods of Disabling the S-RIO Port, complete the following steps:

1. Set Device Control Register.SR_RST_MODE to the desired reset mode.
2. Hold the port in reset by setting RapidIO PLM Port Implementation Specific Control Register.SOFT_RST_PORT.
3. Clear the control or status bit that disabled the port (see Methods of Disabling the S-RIO Port).
4. Remove the port reset by clearing RapidIO PLM Port Implementation Specific Control Register.SOFT_RST_PORT.
5. ackID Synchronization may be required.

Glossary

BDMA	Block DMA Engine inside Tsi721
CA	PCIe Completer Abort error
EFB	Egress Frame Buffer submodule inside Tsi721 PCIe MAC
GPIO	General Purpose I/O
IFB	Ingress Frame Buffer submodule inside Tsi721 PCIe MAC
Inbound	Direction from S-RIO to PCIe
LLM	Local Logical Layer submodule inside Tsi721 S-RIO MAC
Outbound	Direction from PCIe to S-RIO
PBMe	Egress Packet Buffer submodule inside Tsi721 S-RIO MAC
PBMi	Ingress Packet Buffer submodule inside Tsi721 S-RIO MAC
PCIe Ingress	Direction for Tsi721 to receive PCIe TLPs from PCIe root complex
PCIe Egress	Direction for Tsi721 to send PCIe TLPs to PCIe root complex
PCIe MAC	PCIe protocol processing module inside Tsi721
PC2SR	PCIe to S-RIO mapping module inside Tsi721
PCS	Physical Coding Sublayer
PLM	Physical Layer subModule inside Tsi721 S-RIO MAC
PMA	Physical Media Attachment Sublayer
Quasi-reset State	The PCIe Interface enters a quasi-reset state during a boot load process. In this state, the interface responds to all type 0 configuration request TLPs with a configuration-request-retry-status completion. All other TLPs are ignored (that is, flow control credits are returned but the TLP is discarded).
RC	Root complex
SMSG	S-RIO Messaging Engine module inside Tsi721
SR2PC	S-RIO to PCIe Mapping module inside Tsi721
S-RIO Ingress	Direction for Tsi721 to receive S-RIO packets from S-RIO network
S-RIO Egress	Direction for Tsi721 to send S-RIO packets to S-RIO network
S-RIO MAC	S-RIO protocol processing module inside Tsi721
TLP	PCIe Transaction Layer Packet
UR	PCIe Unsupported Request error

CORPORATE HEADQUARTERS
6024 Silver Creek Valley Road
San Jose, CA 95138

for SALES:
800-345-7015 or 408-284-8200
fax: 408-284-2775
www.idt.com

for Tech Support:
email: srio@idt.com

DISCLAIMER Integrated Device Technology, Inc. (IDT) and its subsidiaries reserve the right to modify the products and/or specifications described herein at any time and at IDT's sole discretion. Performance specifications and the operating parameters of the described products are determined in the independent state and are not guaranteed to perform the same way when installed in customer products. The information contained herein is provided without representation or warranty of any kind, whether express or implied, including, but not limited to, the suitability of IDT's products for any particular purpose, an implied warranty of merchantability, or non-infringement of the intellectual property rights of others. This document is presented only as a guide and does not convey any license under intellectual property rights of IDT or any third parties.

IDT's products are not intended for use in life support systems or similar devices where the failure or malfunction of an IDT product can be reasonably expected to significantly affect the health or safety of users. Anyone using an IDT product in such a manner does so at their own risk, absent an express, written agreement by IDT.

Integrated Device Technology, IDT and the IDT logo are registered trademarks of IDT. Other trademarks and service marks used herein, including protected names, logos and designs, are the property of IDT or their respective third party owners.

July 18, 2013