SG1010
StarFabric Switch

Hardware Reference Manual

Revision Information: Revision 2.0 15 October 2004

StarGen, Inc., 225 Cedar Hill Street, Suite 22, Marlborough, MA 01752
www.stargen.com

October 2004

StarGen, Inc. believes the information in this publication is correct; however, the information is subject to change without notice.

StarGen, Inc. does not claim that the use of its products in the manner described in this publication will not infringe on any existing or future patent rights, nor do the descriptions contained in this publication imply the granting of licenses to make, use, or sell equipment or software in accordance with the description.

STARGEN

Contents

Figures

Tables

Preface

This manual describes the StarGen SG1010 Fabric Switch device.

Audience

This manual is written for hardware and software developers.

Overview

This manual contains the following chapters and appendices, a glossary, and an index:

Chapter 1 Introduction – Provides and overview of the SG1010

Chapter 3 Operation – Describes the functional operation of the SG1010.

Chapter 4 Registers – Describes all of the SG1010's registers and their mappings.

Chapter 5 Signal Pin Descriptions – Defines the signals on the SG1010's pins.

Chapter 6 Signal Pin List – Pin lists

Conventions

Register Access Abbreviations

Access	Definition
R	Read only
RTC	Read to clear
RTS	Read to set
RTDEC	Read to decrement
RTINC	Read to increment
R/W	Read, write
R/W1TC	Read, write 1 to clear
R/W1TS	Read, write 1 to set
WRZ	Write, read returns zero

Associated Documents

StarFabric Architecture Specification

SG1010 StarFabric Switch Datasheet

SG1010 StarFabric Switch Hardware Implementation Guide

SG2010 PCI-to-StarFabric Bridge Hardware Reference Manual

Revision History.

Revision Number	Date yy/mm/dd	Description
2.0	**04/09/09**	Incorportates changes for Silicon Revision 3 Removed Electrical Specs - placed in datasheet Removed Internal Pull-ups section - placed in datasheet
0.9b	**01/07/26**	Incorporates changes for Silicon Revision 1.
0.9a	**00/10/13**	Incorporates engineering comments to Rev. 0.9 dated 00/10/04.
0.9	**00/10/04**	Second Draft. Corresponds to SG1010 Functional Specification, Draft Revision 0.9, 04 October 2000. Added placeholders for electrical, mechanical, and thermal specifications, and application notes.
0.4a	**00/09/26**	Incorporates engineering comments to Rev. 0.4 dated 00/09/06.
0.4	**00/09/06**	First Draft. Corresponds to SG1010 Functional Specification, Draft Revision 0.4, 30 August 2000.

1 Introduction

StarGen's SG1010 StarFabric Switch provides high-speed serial switching functionality within StarFabric, a universal switch fabric. The SG1010 provides 30 Gbps of aggregate, non-blocking, full-duplex switching capacity through six links. A link is composed of a transmitter and a receiver, which provides 2.5Gbps bandwidth in both directions simultaneously. Each transmitter and receiver consists of four aggregated 622 Mbps LVDS differential pairs. These links can be used to connect to other StarFabric switch and edge nodes. In addition, two SG1010 links can be connected to another StarFabric node to create a 5Gbps full-duplex path between the nodes. By connecting the SG1010's serial interfaces to other StarFabric switches or StarFabric edge nodes such as StarGen's SG2010 PCI-to-StarFabric bridge, flexible topologies can be designed to fit specific application requirements for bandwidth, reliability, and scalability.

The SG1010 is compliant with the IEEE 1596.3 and TIA/EIA-644 Low-Voltage Differential Signaling standards. The SG1010 supports legacy address-routed traffic, which provides 100% compatibility with existing PCI software including configuration, BIOS, OS and drivers. The SG1010 also supports enhanced StarFabric-native path and multicast routed traffic.

Example applications for the SG1010 include:

- Multi-Service Access Platforms
- DSLAMs-DSL Access Multiplexers
- Voice Over IP Gateways (VOIP)
- Edge Routers/Switches
- Wireless Basestations
- Computer Telephony Integration platforms

2 Feature Summary

The SG1010 StarFabric switch has the following features:

2.1 Scalability and Performance

- 6 StarFabric interface links, 2.5Gbps, full duplex each link
- Links can be bundled to create 5.0Gbps, full duplex point-to-point connection
- 30Gbps switching capacity

2.2 Compatibility

- Support for three routing methods: standard PCI addressing (address routing), path routing, and multi-cast routing
- Standard PCI addressing supports 100% PCI software compatibility
- Software compliant with the PCI Local Bus Specification Revision 2.2, and the PCI-to-PCI Bridge Architecture Specification Revision 1.1
- Physical layer interface is compliant with the IEEE 1596.3 and TIA/EIA-644 Low-Voltage Differential Signaling (LVDS) standards

2.3 Quality of Service

- Credit based flow control: path-based (next-turn) credits and class-of-service (CoS) credits
- 4 classes of service: asynchronous (uses address-routing credits), isochronous, multicast, and provisioning
- Design limits head-of-line blocking
- Dynamic bandwidth reservation protocol

2.4 Reliability, Availability, Serviceability

- Link-by-link CRC checking on all traffic
- Fault detection and isolation
- Path protection capability on register accesses for secure operation

- Hot-pluggable links

2.5 Additional Features

- Flexible event dispatch functionality
- In-band PCI interrupt routing with legacy “swizzle” based on device number
- Supports software generated StarFabric Special frames
- Supports two semaphores with six operations per semaphore
- Diagnostic interface for register read and write access
- SROM interfaces for register preload and vital product data
- 24 LED indicators - either under software control or reflects differential pair or link status
- Eight general purpose I/O pins

3 Operation

This chapter describes the functional operation of the SG1010 StarFabric Switch ASIC.

3.1 Routing and Address Decoding

3.1.1 Address Models

The SG1010 supports two addressing models – a StarFabric address model and a PCI address model. In the StarFabric address model, the SG1010 appears as a switch. To support the PCI address model, the SG1010 appears as a PCI-to-PCI bridge to configuration software. Because there are no bridging functions in the SG1010, the two address models are orthogonal.

3.1.1.1 StarFabric Address Model

The StarFabric address model utilizes a path or multicast ID, a channel, and an offset. Typically, the SG1010 is only interested in the path or the multicast ID, as its primary function is to forward the frame from one port to another. Edge nodes make use of the channel and the offset. Exceptions are frames that address the SG1010 registers.

StarFabric protocol identifies a standard destination channel for device registers, Channel 255. The SG1010 contains a standard set of StarFabric registers, called the StarFabric Component (SFC) Header. These registers are accessible through Channel 255 starting at offset 0. The SG1010 control and status registers (CSRs) are also accessible through Channel 255. The SG1010 PCI configuration registers are dual-mapped in Channel 255 space.

3.1.1.2 PCI Addressing Model

The SG1010 comprises part of a PCI hierarchy and forwards address-routed frames through the fabric. An address-routed frame is directed through the fabric by decoding the frame's address against a set of address ranges at each node. The address ranges are defined at each node by the standard PCI-to-PCI bridge base, limit, and bus number registers.

In the PCI addressing model, the SG1010 is modeled as a transparent PCI-to-PCI bridge. It implements a Type1 (PCI-to-PCI bridge) configuration header. The SG1010 can function with only standard PCI plug-and-play initialization code. The SG1010 is configured like any transparent PCI-to-PCI bridge and can forward PCI configuration, I/O, and memory transactions to upstream and downstream devices through the fabric.

The SG1010 root port represents the primary bus of the PCI-to-PCI bridge model. The other SG1010 ports that are part of the PCI hierarchy collectively represent the secondary bus of the PCI-to-PCI bridge. Each of these individual output ports, referred to as downstream ports, is represented as a secondary bus device. Fabric enumeration determines the root port and the downstream ports of the SG1010. The remaining ports are not part of the PCI hierarchy, but may be enabled through software to perform address routing.

Downstream frames flow from the root port to a downstream port. Upstream frames flow from a downstream port to the root port. A peer-to-peer frame flows from one downstream port to another downstream port.

SG1010 PCI configuration registers are accessible using a Type0 configuration frame from the root port. SG1010 configuration registers cannot be accessed with a configuration frame from any other port.

3.1.2 Path Routing

One of the SG1010's primary functions is to steer path-routed frames from an input port to an output port. A path-routed frame is recognized through its class-of-service. Path-routed classes of service are asynchronous, high priority (HP) asynchronous, isochronous, HP-isochronous, and provisioning. For more information about path routing, see the *StarFabric Architecture Specification.* Output port selection depends on SG1010's link and port numbering scheme.

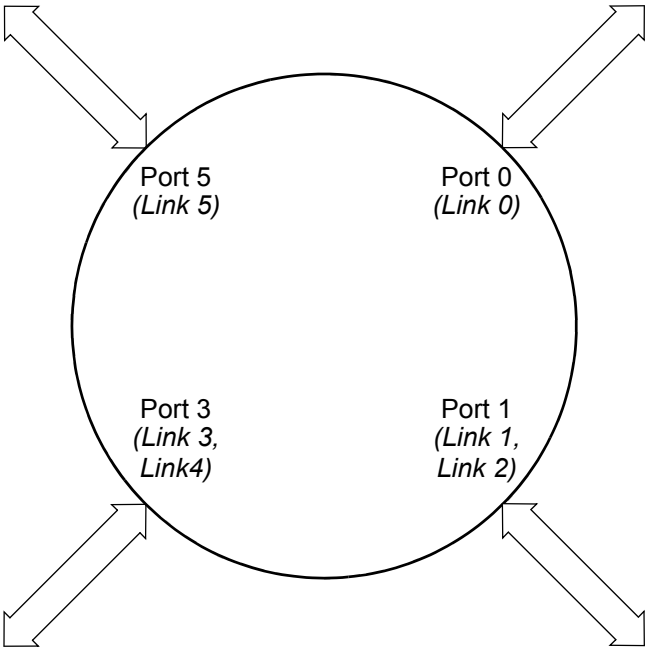
3.1.2.1 Link and Port Configurations

The SG1010 implements six links and supports ports composed of one or two links. Each link is physically numbered, from Link0 to Link5. Each port is numbered based on the lowest numbered link that is a part of that port, as shown in Figure 3–1.

The SG1010 supports a maximum of six ports (one link per port) and a minimum of two ports (some links disabled, some can be bundled). The two port configuration is not very interesting.

Path-routing is a port statement – if a port is comprised of two links, it does not matter which link is used to send the frame as long as the incoming frames on the bundled links are ordered properly.

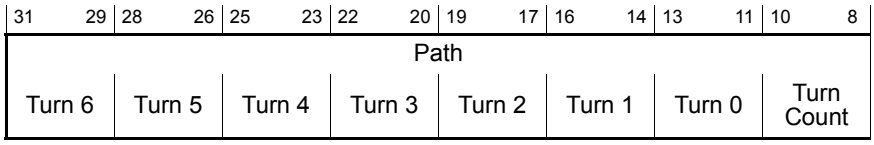
Figure 3–1 SG1010 Link and Port Configuration Example



3.1.2.2 Path Routing through Turn Specifications

A path-routed frame has a 24-bit path field in its header, consisting of one three-bit turn count field and seven three-bit turns (Figure 3–2). A turn is the output port position relative to the input port of the frame, with the first relative position starting at 0. The SG1010 supports up to six ports; that is, there are up to five possible output ports (fewer if the links are bundled). Turn specifications assume that all of the nine ports supported by the StarFabric protocol are present. The SG1010 uses the turn count of the incoming frame to index the turn value. When the SG1010 sends the frame to the next node, it increments the turn count.

Figure 3–2 Path Field



If a path-routed frame has a turn specification that directs it to an output port that is not present or is down, a path event is generated. Because turns are relative to the input port, the illegal turns in a node vary depending on the input port. Figure 3–3 shows how turns are used to route frames, based on the example in Figure 3–1.

Figure 3–3 SG1010 Turn Example

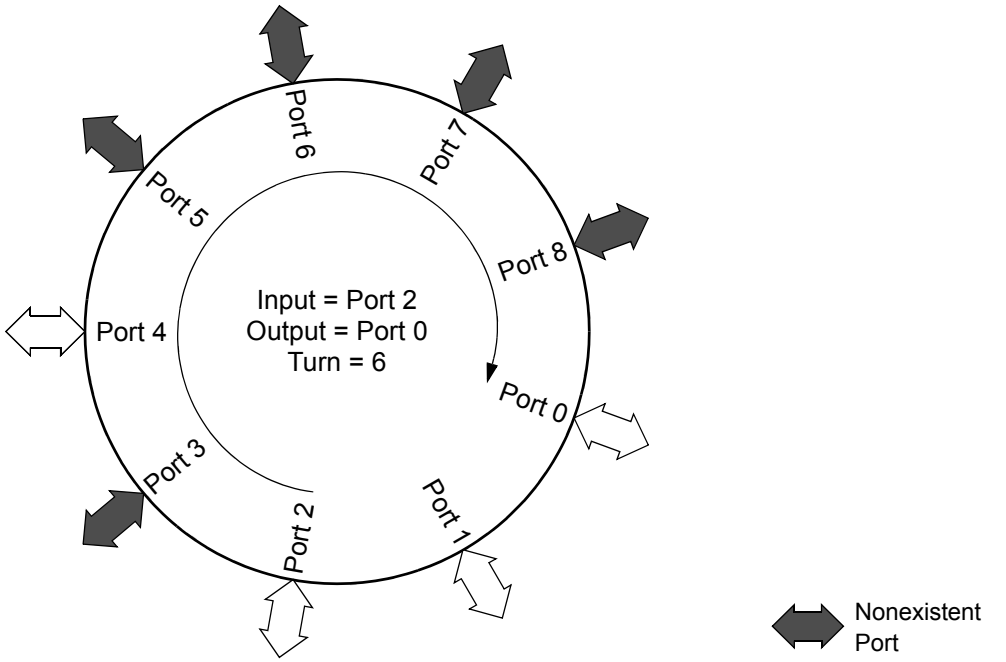


Table 3–1 shows which turns are valid and which are not, based on input port for the switch shown in Figure 3–3.

Table 3–1 Example Valid Turns per Input Port

	Output Port for Given Input Port			
Turn	**Input Port0**	**Input Port1**	**Input Port2**	**Input Port4**
0	Port1	Port2	Invalid	Invalid
1	Port2	Invalid	Port4	Invalid
2	Invalid	Port4	Invalid	Invalid
3	Port4	Invalid	Invalid	Invalid
4	Invalid	Invalid	Invalid	Port0
5	Invalid	Invalid	Invalid	Port1
6	Invalid	Invalid	Port0	Port2
7	Invalid	Port0	Port1	Invalid

SG1010 uses three conditions to determine when to transmit the frame:

- SG1010 has enough line credits for either the turn or class-of-service
- Ordering rules allow the frame to be sent
- SG1010's frame arbiter selects that frame to be sent

For more information about line credits, see Section 3.2. For more information about ordering and frame arbitration, see Section 3.3.

3.1.2.3 Path-Routing Errors

A path event occurs when the SG1010 cannot forward a path-routed frame. A path event occurs when:

- Frame is not sent due to port down (or non-existent)
- Bad path – path ends at a switch (incoming turn count = 7)

When the SG1010 detects one of these conditions, it generates a path event frame as described in Section 3.4.

3.1.2.4 Path-routed CSR Frames to SG1010

Typically, a frame that has a path that ends at a switch (turn count = 7) is an error. Frames that target SG1010 CSRs is the only exception. When the SG1010 detects an incoming frame with a turn count of 7, a provisioning class-of-service, and a Channel ID equal to 255 (FFh), then the SG1010 treats the frame as a CSR access. The SG1010 performs the read or write access, and if required, generates a response frame. Any other incoming path-routed frame with a turn count of 7 results in a path event.

3.1.2.4.1 Register Path Protection

Channel 255 frames targeting SG1010 CSRs are subject to path protection and address range checks. Register access and register access checks are described in Sections 4.1 and 4.2.

A multicast write cannot be used to access registers. Multicast writes that specify Channel 255 will result in a failure type of Software Routing Failure if a response frame is required, and a Software Multicast Distribution Failure event.

3.1.3 Multicast Routing

Multicast frames can be sent to multiple destinations. Multicast frames are restricted to writes; there is no multicast read. For more information about multicast routing, see the *StarFabric Architecture Specification.*

A multicast frame is detected when a frame is received with its class-of-service (CoS) field set to multicast.

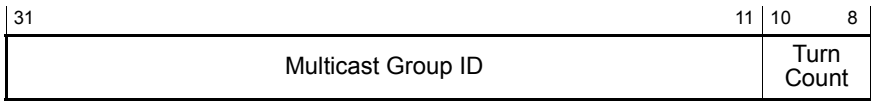
3.1.3.1 Sending Multicast Frames

Multicast frames are routed based on a table lookup at each node. The Multicast Table identifies which ports belong to a given multicast group. Multicast frames are sent out all ports that belong to a multicast group.

A frame's multicast group is identified by the Multicast Group ID field, which is stored in the same 24-bit location as the path in path-routed frames (see Figure 3–4). This field also includes a turn count that the SG1010 increments in the same way that it does when it sends path-routed frames. Multicast frames, like all other frames, are subject to the Turn Count=7 check at each switch.

The SG1010 supports 32 multicast groups, with multicast group IDs from 0 to 31.

Figure 3–4 Multicast Group ID Field



When the SG1010 receives a multicast frame, it determines the frame's multicast group. The SG1010 uses the Multicast Group ID as an index into a 32-entry Multicast Table to select a multicast group entry. Each Multicast Table entry has a per-port bitmap that identifies all the output ports belonging to that multicast group. It also specifies the input port corresponding to that multicast group.

An incoming multicast frame is subject to several checks before it can be forwarded. Table 3–2 summarizes the multicast error reports resulting from these checks, which are described in the following paragraphs. The table lists the event bits that are set, and if the multicast used a write with acknowledge operation, the Failure Type used in the write acknowledge frame. If multiple failures occur, the Failure Type used in the write acknowledge frame is listed from highest to lowest priority in the table. Multiple event bits may be set.

Table 3–2 Multicast Error Reporting Summary

Error	Event	Response Frame Failure Type
Turn Count = 7	Software Multicast Distribution Failure	Software Routing
Multicast group ID > 31	Unsupported Multicast Group ID	Channel Inactive
Input port mismatch	Software Multicast Distribution Failure	Software Routing
Output port down	Hardware Multicast Distribution Failure	N/A
No members in group	Software Multicast Distribution Failure	Software Routing

If the turn count of the incoming multicast frame is 7, then a S/W Multicast Distribution Failure event is signaled. If a write acknowledge is required, a Software Routing Failure is indicated and the frame is discarded. Unlike path-routed frames, a path event is not signaled for multicast frames.

If the SG1010 receives a multicast frame with a Multicast Group ID greater than 31, the SG1010 signals an Unsupported Multicast Group ID event and discards the multicast frame. If a write acknowledge is requested, the SG1010 responds with a Failure Type of Channel Inactive.

If the input port in the Multicast Table entry does not match the input port of the frame, the multicast frame is discarded and a S/W Multicast Distribution Failure event is signaled. If a write acknowledge is required, a Failure Type of Software Routing Failure is indicated.

If an output port is inoperative (down) a Hardware Multicast Distribution Failure event is signaled. Response frames are not returned. The SG1010 still forwards the multicast frame to the output ports of any other group members. A path event is not signaled.

If the SG1010 receives a multicast frame and there are no members in that multicast group, the SG1010 signals a S/W Multicast Distribution Failure event and discards the multicast frame. If a write acknowledge is requested, the SG1010 responds with a Failure Type of Software Routing Failure.

If the frame passes these checks, the multicast frame is forwarded out each port that is a member of the multicast group. The SG1010 sends multicast frames under the same conditions as path-routed frames; that is, line credits must be available, ordering rules must be satisfied, and the frame arbiter must select that frame for transmission out that port.

3.1.3.2 Multicast Acknowledges

The SG1010 supports aggregation of multicast write acknowledge frames. Because a multicast frame can spawn many other multicast frames depending on the number of destinations, a multicast write with acknowledge can generate many acknowledge frames. The SG1010 tracks and aggregates incoming acknowledges for each multicast group. The SG1010 does not forward a multicast acknowledge back to the origin until multicast acknowledges for each member of that multicast group are received. The last acknowledge received is always forwarded to the input port for that multicast group. A multicast acknowledge with a failure type other than Normal is always forwarded, regardless of whether it is the last acknowledge for the group. The Final Multicast Ack bit in the header distinguishes between the last aggregated write acknowledge frame coming from a port and intermediate non-aggregated acknowledge-with-error frames.

SG1010 Multicast Table entries include a field that tracks received multicast acknowledge frames. As described previously, each entry corresponds to a multicast group. The multicast acknowledge field has one bit per port.

If the SG1010 receives a multicast acknowledge frame with the Final Multicast Ack bit set (indicating that it is the last acknowledge from that port), the SG1010 sets the bit corresponding to the port in the entry corresponding to that multicast group. The SG1010 then compares the multicast acknowledge field of that entry to the multicast output port field. If a multicast acknowledge bit is set for each member of that multicast group, then all multicast acknowledges have been received for that group. The SG1010 forwards the aggregated multicast acknowledge to the next node with the Final Multicast Ack bit set. The SG1010 then clears all the bits in the multicast acknowledge field for that multicast group.

If the SG1010 forwards a multicast acknowledge frame that has a Failure Type other than normal, the SG1010 must clear the Final Multicast Ack bit in the header of the frame if it is not the last acknowledge frame for that group.

If the SG1010 generates a multicast write acknowledge due to the failures described in the previous section, the SG1010 sets the multicast acknowledge bit(s) for that multicast group for the affected ports. If all expected acknowledge bits are then set, the SG1010 sets the Final Multicast Acknowledge bit in the acknowledge frame that it generates.

If the SG1010 receives a multicast write acknowledge frame for either a group that has not been set up, or on a port that is not a member of an existing group, then the SG1010 discards the write acknowledge frame. If the SG1010 receives a multicast write acknowledge frame for an unsupported Multicast Group ID, or if the Turn Count is equal to 1, then the SG1010 discards the write acknowledge frame and sets a S/W Multicast Distribution Event bit.

The multicast write acknowledge aggregation function supports only one outstanding multicast write with acknowledge operation per group. When the SG1010 receives a multicast write with acknowledge, it clears Multicast Ack bit-field for that Multicast Group ID. Software may also clear the Multicast Ack bit-field by writing with a 1 the Clear Multicast Ack bit located in the same entry.

3.1.3.3 Tracking Multicast Acknowledge with Failures (Multicast Nacks)

The SG1010 supports multicast nack tracking, which provides additional tracking information for multicast acknowledge frames that contain a failure indication (Failure Type is not Normal). These frames are also referred to as multicast nacks. The multicast nack tracking mechanism allows software to determine where a multicast distribution failure occurred.

The Nack Tracking Support (NTS) bit in each Multicast Table entry indicates whether the node tracks multicast nacks for the multicast group corresponding to the entry. The SG1010 always returns 1 indicating that it supports this function.

The Nack Select (NSL) bit in each Multicast Table entry indicates whether the Multicast Acknowledge bitmap in the entry reflects the multicast acknowledge tracking results or the multicast nack tracking results. If the Nack Select bit is a 0, multicast acknowledge tracking is shown in the bitmap, otherwise if the bit is a 1, multicast nack tracking results are shown.

When multicast nack tracking is enabled, the node tracks the ports where multicast nack frames are received, regardless of the state of the Final Multicast Ack bit. Nack tracking results are reflected once all the multicast acknowledges are received for a multicast group, as described in Section 3.1.3.2, and the final multicast acknowledge for that group is sent. When multicast aggregation is complete, and if there were any multicast nacks received for that group, the node sets the Nack Select bit and places the nack tracking results in the Multicast Acknowledge bitfield. When software is diagnosing the failure, it may read the Nack Select bit as 1, and then determine which ports received multicast nacks. If multicast aggregation is not complete, or no multicast nacks were received, the Nack Select bit is 0, and the Multicast Acknowledge bitfield reflects multicast Acknowledge frames that were received with the Final Multicast Ack bit set.

As with the default Multicast Acknowledge bitfield, the Multicast Nack bitfield is cleared when a new Multicast Write with Acknowledge is received for that group, or when software writes the Clear Multicast Ack bit.

3.1.3.4 Configuring Multicast Groups and Bandwidth Allocation

The SG1010 manages multicast group membership based on multicast join group, exit group, and tear down group commands carried in Bandwidth Request provisioning frames. Because these same frames also carry bandwidth allocation commands, bandwidth allocation is also described in this section.

A join group, exit group, and tear down group operation must not be initiated by the origin while a multicast write with acknowledge is outstanding for that same multicast group. Similarly, a multicast write with acknowledge must not be initiated by the origin while a join group operation for that same multicast group is outstanding.

3.1.3.4.1 Bandwidth Allocation

Bandwidth allocation allows the dynamic reservation of bandwidth on a link for multicast and isochronous traffic. Before a multicast group or isochronous stream is set up, the bandwidth allocation mechanism ensures that the links can handle the additional traffic demands. The SG1010 manages bandwidth allocation on its output links using provisioning CoS frames with bandwidth allocation and bandwidth free commands.

Each link has a Bandwidth Count register in its Link State Table to track the amount of bandwidth reserved by traffic streams using that link. When two links are bundled, both link counters are used to allocate bandwidth for the port. The counters are reset to a non-zero value that correlates to the speed of the link. The number is calculated to be no more than 80% of the full link bandwidth. The Link Speed field in the Link State Table Control and Status register indirectly provide the reset value for the Bandwidth Counter. The link speed for the the SG1010 is 2 Gb/sec (after removing the 8B/10B overhead), which results in a Bandwidth Counter reset value of 5FFh according to the *StarFabric Architecture Specification*.

When bandwidth is allocated, the bandwidth requested is subtracted from the counter. When bandwidth is freed, the specified bandwidth is added to the counter. When two links are bundled, the SG1010 requires that all of the bandwidth requested in a frame come from any one, but only one, of the two link bandwidth counters. The SG1010 does not combine the available bandwidth from more than one counter to satisfy a bandwidth request. Additionally, freed bandwidth is restored to one, and only one, bandwidth counter. Reading the Bandwidth Count register returns the current value of the counter.

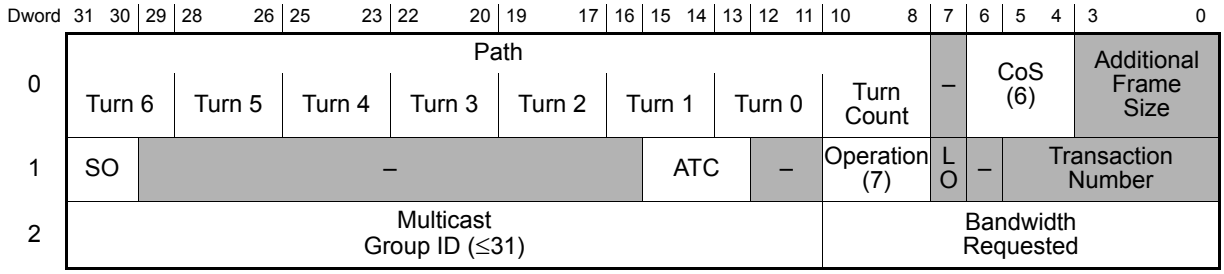
3.1.3.4.2 Joining Multicast Groups and Allocating Bandwidth

Ports become members of a multicast group through the Join Group command. The Join Group command also allows for bandwidth allocation. The join group operation is specified by a Bandwidth Reservation frame (path-routed, provisioning CoS) frame with its Secondary Operation specified as Join Group/Allocate (see Figure 3–5). This frame includes a Multicast Group ID field to specify the desired multicast group, and a Bandwidth Requested field to specify the requested bandwidth. Multicast group 1FFFFFh (7777777 octal) is used when a bandwidth allocation without a multicast operation is desired; Multicast ID 1FFFFFh is a null ID and never contains any mem-

bers. When the SG1010 detects a Bandwidth Reservation frame with a Multicast Group ID of 1FFFFFh, it performs the bandwidth allocation operation but does not perform the join group operation.

The bandwidth allocation operation is performed as the Bandwidth Reservation frame moves from the origin to the terminus. The join group operation is performed Bandwidth Response frame moves from the terminus back to the origin. Bandwidth allocation is only performed on those ports which are on a unique path to the terminus, that is, the output port is not already a member of the multicast group. If the port is already a member of the group, it is assumed the bandwidth for that group has already been allocated at that port. The SG1010 determines whether a port is already a member of the group by checking the Multicast Table entry corresponding to the group; if the Multicast Group bit corresponding to that port is not set, it is not already a member.

Figure 3–5 Bandwidth Reservation Frame Header



SO = Secondary Operation
ATC = Activating Turn Count

The SG1010 performs tests to verify the following:

- The output port is operational (up).
- The turn count is less than 7.
- Sufficient bandwidth exists to accommodate the bandwidth requested in at least one bandwidth counter corresponding to the output port.
- The multicast group is supported (ID ≤ 31, or ID=1FFFFFh for a null join operation).

If any of these tests fail, the SG1010 generates a provisioning Bandwidth Response frame back to the origin with the Secondary Operation set to the appropriate failure type. In the case of multiple failures, the Failure Type is chosen based on the following fixed priority:

1. Port failure or turn count=7
2. Unsupported multicast group
3. Bandwidth unavailable

If the port is down or unavailable, the Bandwidth Available field is set to 0; otherwise it is set to the value of the bandwidth counter for that link. When this response frame is sent back to the origin, intermediate nodes must perform cleanup operations as described in Section 3.1.3.4.2.1. The original Bandwidth Reservation frame is discarded.

If all the tests are successful, the SG1010 subtracts the bandwidth requested from one of the bandwidth counters corresponding to the port. The actual Join Group operation is performed when the response frame is returned by the terminus or, in the event of a subsequent failure, another switch.

When the SG1010 detects a Bandwidth Response frame with a Positive Acknowledge, it performs the Join Group operation. The SG1010 uses the Multicast Group ID in the header to index the Multicast Group Table and sets the bit corresponding to the input port of the response frame (which was the output port for the Bandwidth Reservation frame).

3.1.3.4.2.1 Bandwidth Response Cleanup

When an error occurs in the Join Group/Allocate operation, the node that detected the error generates a provisioning Bandwidth Response frame and sends it to the origin. Along each of the switches on the path to the origin, but only if the path is unique for that multicast group, the bandwidth must be restored. When the response frame indicates that an error has occurred, the Join Group operation is not performed.

The SG1010 determines whether it must restore the bandwidth in a failed Join Group operation checking to see if the input port of the Bandwidth Response frame was already a member of that multicast group. The SG1010 performs a lookup of the Multicast Table entry corresponding to that multicast group. If the bit corresponding to the port is set, the port was already a member and bandwidth restoration is not performed. If the bit is clear, this port was on a unique path to the terminus and bandwidth must be restored. Bandwidth is always restored if the Multicast Group of the frame is assigned to 1FFFFFh.

To restore bandwidth, the SG1010 adds the requested bandwidth to the bandwidth counter of any link in that port. The SG1010 then forwards the response frame to the next node.

3.1.3.4.3 Exiting Multicast Groups and Restoring Bandwidth

The multicast Exit Group/Free Bandwidth operations use a Bandwidth Reservation frame (path-routed, provisioning CoS) with the Secondary Operation specified as Exit Group/Free. This frame includes a Multicast Group ID field that specifies the multicast group, and a Bandwidth Requested field that, in this case, specifies the amount of bandwidth to be restored.[1] Multicast group 1FFFFFh is used when bandwidth restoration is desired without performing a multicast operation. When the SG1010 detects a Multicast Group ID of 1FFFFFh, it performs the bandwidth free operation but does not perform the exit group operation.

The Activating Turn Count field in the Bandwidth Reservation frame header specifies which nodes along the path are to perform the exit group operation. When the SG1010 receives this frame, it compares the Activating Turn Count field with the Turn Count

1. For proper bandwidth-allocation operation, the amount of bandwidth restored should be the same as the amount requested when that multicast group was formed.

field. If the Activating Turn Count is less than or equal to the incoming turn count of the frame, then the SG1010 performs the exit group operation. Otherwise, the SG1010 forwards the frame to the next node.

If the Activating Turn Count comparison is successful and the Bandwidth Requested field is non-zero, the SG1010 restores (adds) that bandwidth to a bandwidth counter for any link in the output port. If the Bandwidth Requested field is zero, then no bandwidth restoration is performed and only the exit group operation is done.

To perform the exit group operation, the SG1010 uses the Multicast Group ID in the header to index the Multicast Group Table and clears the bit corresponding to the output port for the frame. This port can no longer be used as an output port for outgoing multicast frames having that Multicast Group ID. The SG1010 then forwards the frame to the next node.

Bandwidth Response frames are not generated in response to an Exit Group operation.

3.1.3.4.4 Tearing Down Multicast Groups

A multicast origin may decide that it will no longer support that multicast group. In this case, the node sends a provisioning Bandwidth Reservation frame with the Secondary Operation specifying Tear Down Group to clear the all members in the multicast group in all nodes. This frame is multicast-routed to all nodes that have ports that are members of the multicast group. The Activating Turn Count field is ignored during Tear Down Operations; that is, the SG1010 performs a Tear Down operation regardless of the value of this header field.

When the SG1010 receives a Bandwidth Reservation frame with a Tear Down Group command, the bandwidth specified in the frame is added to the bandwidth counter of any link in each port that belonged to that multicast group before it was cleared.[1]

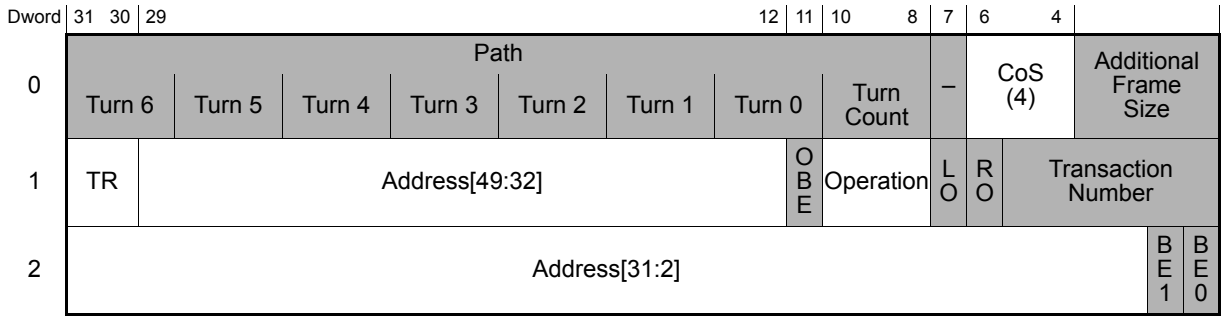
The SG1010 then clears all bits in the corresponding entry in the Multicast Group Table. When the operation is complete, there are no members in that multicast group. the SG1010 forwards the frame out all ports that were members of the multicast group before the tear down operation was performed.

Bandwidth Response frames are not generated in response to a tear down group operation.

3.1.4 Address Routing

An address-routed frame is identified by its address-routed CoS. The SG1010 selects the output port for an address-routed frame by decoding and comparing the frame's Address field against a set of address ranges, which are enabled by control bits. The frame's Target Region field identifies the address region as 32-bit memory, 64-bit memory, I/O, or configuration. Figure 3–6 shows the typical fields in an address-routed frame header.

Figure 3–6 Address Routed Frame Header



TR = Target Region

The SG1010 implements the standard set of configuration registers for a PCI-to-PCI bridge, which include registers that define memory, I/O, and configuration address ranges. These ranges are used for upstream address decoding, as well as downstream Type1-to-Type0 configuration translations.

The SG1010 also implements a Port Map Table in its CSR space. Each port has a Port Map Table entry. Each entry contains copies of the SG1010 link partner configuration registers defining these ranges. The Port Map Table is used for downstream and peer-to-peer address decoding.

The registers in the Port Map Table include copies of:

- Memory base and limit configuration registers
- Prefetchable memory base and limit configuration registers, including upper 32 bit registers
- I/O base and limit configuration registers, including upper 16 bit registers
- Primary, secondary, and subordinate bus number configuration registers
- Command configuration register, including memory enable, I/O enable, master enable, and VGA snoop control bits
- Bridge control configuration register, including ISA mode and VGA mode bits, and, if enabled, the VGA 16-bit decode control bit
  - The VGA 16-bit Decode bit is visible only if the VGA 16-bit Decode Enable bit is set in the Chip Control and Status 0 register in Channel 255 register space

During fabric enumeration, the SG1010 enables the Port Map Tables that are associated with downstream ports in the PCI hierarchy by setting the Port Map Table Enable bit in the Port State Table. Software can also enable the Port Map Tables by setting this bit using a register write.

When an address-routed frame is routed through the fabric, a path is built in the frame header as it is forwarded from node to node. When the SG1010 selects the output port for the address-routed frame based on address decodes, it increments the turn count and inserts the turn values corresponding to that turn count. The turn value is the location of the output port relative to the input port.

3.1.4.1 Smart Address Routing

Smart address routing allows shorter address-routed paths for memory and I/O peer-to-peer traffic. Smart address routing allows routing along a link that is not a part of the PCI hierarchy, bypassing the PCI hierarchical path.

Software can enable a port for smart address routing by setting the Smart Address Enable bit in the Port State Table. Both the Port Map Table Enable and the Smart Address Enable bits (see the Port State *x* CSR[4:5], Section 4.6.2.2) must be set to allow smart address routing. Setting the Smart Address Enable bit disables the decoding of configuration transactions against the Port Map Table, while still allowing memory and I/O transactions to be decoded. All address-routed frames are decoded against the Port Map Tables based on these bits.

3.1.4.2 Downstream Address Routing

Downstream address-routed frames are received from the root port. When a downstream address-routed frame is received, it has already been decoded against the SG1010 address ranges through its link partner's Port Map Table.

The SG1010 performs a similar decode against its own Port Map Tables to select the output port for the frame. The frame's Address field is decoded against all Port Map Tables that have the Port Map Table Enable bit set. This Port Map decode is essentially the PCI-to-PCI (P2P) address decode for all devices at the next level of hierarchy. The Port Map Table decode is a positive decode against all the ranges defined for that type of operation – SAC memory (32-bit only), DAC memory (64-bit only), I/O, or configuration. If none of the Port Map Table address decodes are successful, an Address-routing Failure occurs and the frame is discarded. The SG1010 sets the Master Abort Received bit in its Secondary Status PCI configuration register (described in Section 4.7.2.7). If the frame was memory or I/O, the SG1010 signals a Address-routing Failure event. Response frames use a Failure Type of Software Routing Failure.

3.1.4.3 Upstream and Peer-to-Peer Address Routing

Upstream and peer-to-peer address-routed frames are received on a non-root port. Upstream address-routed frames continue upstream towards the root, while peer-to-peer frames are forwarded to another non-root port.

When the SG1010 receives an address-routed frame from a non-root port, it positively decodes the frame's address against all Port Map Tables that have their Port Map Table Enable bits set. The Port Map Table decode is a positive decode against all the ranges defined for that type of operation (SAC memory, DAC memory, I/O or configuration). If a Port Map Table positive decode is successful, the frame is sent out the corresponding port (and is a peer-to-peer frame).

If none of the Port Map Table decodes are successful, the frame's address is inversely decoded against the appropriate address ranges in the SG1010 configuration registers. The Master Enable bit in the SG1010 Command register must be set to enable a successful inverse decode of memory and I/O frames. The inverse decode detects whether the address falls outside of the address range. If the inverse decode is successful, the frame is sent out the root port (and is an upstream frame).

If neither the Port Map Table decode nor the inverse decode are successful, an Address-routing Failure occurs and the frame is discarded. If the frame is a memory or I/O frame, the SG1010 signals an Address-routing Failure event. Response frames use a Failure Type of Software Routing Failure.

3.1.4.4 PCI Configuration Routing

This section describes the decoding and routing of PCI configuration frames. Configuration frames are forwarded through the fabric as address-routed frames based on bus number decodes.

3.1.4.4.1 Type0 Configuration Register Accesses

The SG1010 configuration registers are accessed through a Type0 configuration frame from the root port. A Type0 configuration frame is distinguished from a Type1 configuration frame through Offset[49]. If Offset[49] of a configuration frame is 0, the configuration frame is Type0, if it is a 1, it is Type1.

The SG1010 does not support configuration register accesses from any port other than the root port. If it receives a Type0 configuration frame from another port, an Address-routing Failure occurs. If a response frame is returned, a Failure Type of Software Routing Failure is used. No event is signaled.

The SG1010 performs the register write regardless of whether the operation is a write with or without error.

3.1.4.4.2 Type1 Downstream Configuration Frames

Downstream Type1 configuration frames enter the SG1010 from the root port and exit through a downstream port in the PCI hierarchy.

The SG1010 converts a Type1 configuration frame into a Type0 configuration frame if the bus number in the frame's Address field matches the value in the Secondary Bus Number register in SG1010's configuration space. If the compare is successful, the SG1010 clears Address[49]. The output port is selected based on the device number carried in the frame's Address field. The PCI device number maps to a StarFabric port number, i.e., Device 0 is sent out Port 0. The Type0 frame is forwarded to a port if:

- The device number in the frame matches the port number
- The port's Port Map Table Enable bit is set
- The port's Smart Address Enable bit is not set
- The port's Secondary and Subordinate Bus Number registers are non-zero

If there are no port number matches, or if the port's enable bits are not set as described above, then an Address-routing Failure occurs. If a response frame is required, the SG1010 uses a Failure Type of Software Routing Failure. The SG1010 also sets the Master Abort Received bit in the Secondary Status register.

The SG1010 signals a Fabric Special Cycle event if the bus number in the frame's Address field matches the value in the Secondary Bus Number register in SG1010's configuration space and the following encoding is detected:

- Register number is all 0s
- Device number is all 1s
- Function number is all 1s
- Must be a write operation (including write with acknowledge, and/or with error)

In this case, if a write acknowledge frame is required, a Failure Type of normal is used.

The SG1010 does not convert the frame to a Type0 if its bus number does not match SG1010's secondary bus number. The output port of the Type1 frame is then selected based on Port Map Table decodes against the bus number. The SG1010 performs the bus number comparison against all Port Map Tables that have

- The Port Map Table Enable bit set
- The Smart Address Enable bit not set
- Non-zero values in the Secondary and Subordinate Bus Number registers

If the bus number is greater then the secondary bus number and less than or equal to the subordinate bus number in the Port Map Table, then the positive decode is successful. The SG1010 sends the Type1 configuration frame out that port. If there are no successful decodes, an Address-routing Failure occurs. If a response frame is required, a Failure Type of Software Routing Failure is used. The SG1010 sets the Master Abort Received bit in the Secondary Status register.

3.1.4.4.3 Upstream and Peer-to-Peer Type1 Configuration Frames

The SG1010 receives upstream configuration frames from a non-root port. Upstream Type1 configuration frames are only forwarded if they are to be converted to a special cycle on another PCI bus. StarFabric protocol does not support the generation of PCI Special Cycles in the fabric, but does support the propagation of these Type1 frames through the fabric for conversion to a PCI Special Cycle on a PCI bus. The Type1 configuration frame must have the following encoding:

- Register number is all 0s
- Device number is all 1s
- Function number is all 1s
- Must be a write operation (including write with acknowledge, and or with error)

If the received Type1 configuration frame does not have this encoding, an Address-routing Failure occurs. If a response frame is returned, a Failure Type of Software Routing Failure is used.

If the special cycle encoding is detected, the SG1010 performs the following bus number comparisons:

1. Direct compare against the Primary Bus Number register in SG1010 configuration space
2. Positive decode against the range defined by the Secondary Bus Number (inclusive) and the Subordinate Bus Number (inclusive) in SG1010's Port Map Tables
   a. The Port Map Table Enable bit must be set
   b. The Smart Address Enable must be clear
   c. The bus number registers must be non-zero
3. Inverse decode against the range defined by the Secondary Bus Number (inclusive) and the Subordinate Bus Number (inclusive) in SG1010's configuration space

Case 1 tests to see if the Type1 configuration frame is decoded such that it would be converted to a PCI Special Cycle on its root port. If this decode is successful, the SG1010 discards the frame. If a response frame is returned, a Failure Type of normal is used. The SG1010 signals a Fabric Special Cycle event.

Cases 2 and 3 select the output port for the Type1 configuration frame. If any of the positive decodes against the Port Map Tables are successful, the frame is forwarded out the corresponding port. Otherwise, if the inverse decode is successful, the frame is forwarded out the root port. If none of the decodes are successful, an Address-routing Failure occurs. If a response frame is returned, a Failure Type of Software Routing Failure is used.

3.1.4.5 Address-routed Memory Frame Decoding and Routing

This section describes the decoding and routing of PCI memory frames in the SG1010. Address-routed memory frames are forwarded through the fabric based on base and limit address decodes at each node. The SG1010 is never the target of address-routed memory frames; it only forwards them.

The SG1010 performs address decodes against ranges defined by the following registers in either SG1010's configuration space or Port Map Tables:

- Memory Base and Limit Registers, defining a 32-bit address range in non-prefetchable space
- Prefetchable Memory Base and Limit Registers, defining a 64-bit address range in prefetchable space
- If the VGA Enable bit is set in the Bridge Control register, 32-bit memory addresses 000A0000h – 000BFFFFh

For downstream or peer-to-peer decodes, the Memory Enable bit must be set in the Command register (described in Section 4.7.1.3) in the corresponding Port Map Table to enable a successful decode. For upstream decodes, the Master Enable bit must be set in the configuration Command register to enable a successful decode. If the appropriate bit is not set, an Address-routing Failure occurs.

3.1.4.5.1 Downstream Memory Frames

Downstream memory address-routed frames enter the SG1010 from the root port and exit through a non-root port. To select the output port, the SG1010 positively decodes the Address field of memory address-routed frames against the memory ranges in the Port Map Tables that have the Port Map Table Enable bit set.

If there are no successful decodes against the Port Map Tables, an Address-routing Failure occurs. If a response frame is returned, a Failure Type of Software Routing Failure is used. The SG1010 sets the Master Abort Received bit in the Secondary Status configuration register and signals an Address-routing Failure event.

3.1.4.5.2 Upstream and Peer-to-Peer Memory Frames

Upstream memory address-routed frames enter through a non-root port, and are positively decoded against the Port Map Table ranges to select a port for peer-to-peer forwarding and inversely decoded against configuration register ranges to select the root port. The Port Map Table Enable bit must be set to enable a port map decode for peer-to-peer forwarding.

If a Port Map Table decode is successful, the frame is forwarded out that port. Otherwise, if the inverse decode is successful, the frame is forwarded out the root port. If no decodes are successful, an Address-routing Failure occurs. If a response frame is returned, a Failure Type of Software Routing Failure is used. The SG1010 signals a Address-routing Failure event.

3.1.4.5.3 64-bit Addressing

Address-routed frames support 50 address bits. In order to support 64-bit addressing, the system composed of the StarFabric and its endpoints must be mapped in the same 50-bit address region. However, this region may be located anywhere in 64-bit address space. In other words, bits [63:50] may be any value, either zero or non-zero, but all nodes in the StarFabric must use this value.

In order to resolve address decoding ambiguity for address-routed frames, the frame header Target Region field indicates whether the original address was a 32-bit address (SAC) or a 64-bit address (DAC).

A frame address with a 64-bit address (DAC) specified in its Target Region field is decoded only against only the 64-bit prefetchable memory base and limit address registers; all other decodes (32-bit memory and VGA) are considered unsuccessful.

A frame offset with a 32-bit address (SAC) specified in its Target Region field is assumed to have all zeros in its upper address bits [63:32] for the purposes of comparing against the 64-bit prefetchable memory base and limit address registers. These frames are also decoded against the 32-bit memory range and, if enabled, the VGA memory range.

3.1.4.6 PCI I/O Decoding

This section describes the decoding and routing of I/O address-routed frames through the SG1010.

I/O address-routed frames are forwarded through the fabric based on base and limit address decodes. The SG1010 is never the target of an I/O address-routed frame.

When forwarding I/O frames, the SG1010 decodes I/O addresses against the following registers, either in SG1010's configuration space or Port Map Table:

- I/O base and limit registers, defining a 32-bit address range.
- Bridge Control register (described in Section 4.7.3.4):
  - If the ISA Mode bit is set, only the low 256 bytes in each 1KB chunk is positively decoded; that is, bits [9:8] must be 00b (this applies only to I/O addresses in the first 64KB).
- If the VGA Enable bit is set, VGA I/O address bits [9:0] = 3B0:3BBh and 3C0:3DFh.
  - If the VGA 16-bit Decode bit is set, address bits [15:10] must be 0, otherwise they can be any value. The VGA 16-bit Decode bit is only visible when the VGA 16-bit Decode Enable bit is set in the Chip Control and Status 0 register in Channel 255 register space. By default VGA 16-bit decode is not visible, and therefore not enabled.
  - Address bits [63:16] must be zero.
- If the VGA Snoop bit in the Command register (described in Section 4.7.1.3) is set, VGA I/O address bits [9:0] = 3C6h, 3C8h, and 3C9h
  - If the VGA 16-Bit Decode bit in the Bridge Control register is set, address bits [15:10] must be 0, otherwise they can be any value. See the note above on enabling the VGA 16-bit Decode bit.
  - Address bits [63:16] must be zero.

For downstream or peer-to-peer decodes, the I/O Enable bit must be set in the Command register in the corresponding Port Map Table. For upstream decoding, the Master Enable bit must be set in the configuration Command register. Otherwise, an Address-routing Failure occurs.

3.1.4.6.1 Downstream I/O Frames

Downstream I/O address-routed frames enter the SG1010 from the root port and exit through a non-root port. To select the output port, the SG1010 positively decodes the offset of I/O address-routed frames against the I/O ranges in the Port Map Tables that have the Port Map Table Enable bit set. The Smart Address Enable bit is ignored for I/O address decoding.

If there are no successful decodes against the Port Map Tables, an Address-routing Failure occurs. If a response frame is returned, a Failure Type of Software Routing Failure is used. The SG1010 sets the Master Abort Received bit in the Secondary Status configuration register and signals an Address-routing Failure event.

3.1.4.6.2 Upstream and Peer-to-Peer I/O Frames

Upstream I/O address-routed frames enter through a non-root port, and are positively decoded against the Port Map Table ranges to select a peer-to-peer port and inversely decoded against configuration register ranges to select the root port. The Port Map Table Enable bit must be set to enable a Port Map decode. The Master Enable bit in the Command configuration register must be set to enable the inverse decode.

If a Port Map Table decode is successful, the frame is forwarded out that port. Otherwise, if the inverse decode is successful, the frame is forwarded out the root port. If no decodes are successful, an Address-routing Failure occurs. If a response frame is returned, a Failure Type of Software Routing Failure is used. SG1010 signals a Address-routing Failure event.

3.1.5 Generating Address-Routed Response Frames

When an address-routed read or write with acknowledge frame encounters a Address-routing Failure (address decode failure) in the SG1010, then the SG1010 generates a response frame (write acknowledge or read completion) to the origin of the read or write frame.

Response frames are generated in response to the following types of operations:

- Address-routed configuration read or write with acknowledge
- Address-routed I/O read or write with acknowledge
- Address-routed memory read or write with acknowledge

3.2 Line Credits

The SG1010 and all StarFabric-compliant devices use a credit algorithm to manage data buffer resources. This line credit algorithm provides a guaranteed delivery service; that is, a frame is not sent unless there is buffer space available in its link partner to accommodate it. For more information about the line credit mechanism, see the *StarFabric Architecture Specification.*

3.2.1 Line Credit Types

Line credit types are composed of CoS credits and turn credits. This allows a frame to be forwarded based on either the type of frame, or the path that the frame is taking.

Although the protocol allows seven possible classes–of-service, the SG1010 supports four CoS line credit types. The SG1010 aliases the remaining three classes-of-service for line credit and ordering purposes. The SG1010's four CoS credit types are:

- address-routed/asynchronous

- isochronous/HP-isochronous
- multicast
- provisioning/HP-asynchronous.

The first CoS in each pair is the CoS used in credit debiting, returning credits, and receiving credits.

The SG1010 supports eight sets of turn credits per link for each of the eight possible turn values in the next node. Turns are numbered relative to the input port used for the frame on that component.

The CoS line credit types must be used only for frames with the corresponding CoS, but can be used with any turn value. The turn line credits can be used with any CoS but are limited to frames using that turn value in the next node.

Turn credits and CoS credits are subdivided into request credits and write/completion credits – for more information, see Section 3.3.1.

3.2.2 Line Credit Initialization

The SG1010 allocates a number of line credits to represent its own buffers. Part of the initialization process involves providing these line credit values to its link partner(s), so that the link partners can initialize their credit counters. The SG1010 initializes its credit counters with the line credit information that it receives from its link partners. For more information about the initialization process, including a detailed description of line credit initialization, see Section 3.5.4.4.

3.2.3 Line Credit Reallocation

Software may use the software generated frame mechanism to generate a turn or CoS credit update special frame to be sent to a link partner. This allows software to reallocate the number of line credits at the link partner's disposal for sending frames to the SG1010. The SG1010 allows credit reallocation between different CoS credits, between different turn credits, or between turn and CoS credits. Request and write credits cannot be exchanged; this is not permitted by the StarFabric protocol.

Software may also send a line credit update from a link partner to the SG1010 to reallocated credits in the SG1010's credit counters. In this case, credit reallocation is subject to the restrictions imposed by the link partner.

The Update Credit special frame indicates the amount by which the corresponding line credit counter(s) is to be incremented. Line credit counters may be increased at any time. Line credit counter values may be decreased by adding a large enough amount so that the counter wraps. The SG1010 write credit counters are seven bits wide, allowing a maximum credit amount of 127. If a credit counter is set to 12 credits, for example, it may be incremented by 128-12=116 to reduce the line credit amount to 0. Because a line credit reduction depends on the current value of the counter, this operation should only be performed when the counter is not actively being used.

3.2.4 Using Line Credits

In order for the SG1010 to send a frame, enough line credits must exist to ensure that the frame can be buffered in the appropriate link partner.[2] The amount of line credits required to send a frame is equal to the size of the frame in lines (16 bytes). If there are not enough line credits to send the frame, the SG1010 continues to buffer the frame and does not send it until enough line credits are returned from its link partner to accommodate the frame. The SG1010 may also internally generate frames to send – software generated frames, event frames, or CSR read completion frames. Again, the appropriate amount of line credits for the link partner must be available before the SG1010 can send these frames.

Either turn credits or CoS credits may be used to send a frame. However, turn and CoS credits cannot be combined to send a given frame, there must be sufficient credits of one type or the other.

When the SG1010 has a frame to send, it first checks the line credits allocated to the appropriate turn credit counter for the output link.[3] If there are sufficient turn credits available, the SG1010 sends the frame and subtracts the amount of credit from that output link's line credit counter corresponding to that turn value in the next component.

If there are insufficient credits in the turn credit counter to accommodate the frame, then the SG1010 checks the appropriate CoS credit counter for that link. If there are sufficient CoS credits, then the SG1010 sends the frame and subtracts the amount of credit from that line credit counter.

Because address-routed frames are routed based on the address field and not a path composed of turns, the turn at the next node is not known. The SG1010 cannot use turn credits to send address-routed frames and must use CoS credits.

The following frames use request credits:

- All read request frames
- I/O or configuration address-routed write frames

All other frames types use write/completion credits:

- Write frames, except address-routed I/O and configuration write frames
- Read completion frames
- Write acknowledge frames
- Bandwidth reservation and response provisioning frames

2. The decision to send a frame does not rely solely on line credits. The frame can be sent only if ordering rules allow. If line credit conditions and ordering rules are satisfied, SG1010's frame arbiter operates on all frames that are eligible to be sent to that same port to determine which frame is sent first. The frame arbiter must select that frame as the next to be sent. Ordering and frame arbitration are described in Section 3.3.
3. If there are seven turns in the path, on the final turn there is no "turn" in the next component. In this case, turn 0 credits can be used.

Special frames do not require line credits to be sent. Special frames are used for initialization, fabric enumeration and line credit updates between link partners.

3.2.5 Returning Line Credits to Link Partners

SG1010's link partners also use line credits to send frames to the SG1010. The SG1010 returns accumulated line credits to its link partners when the SG1010 frees buffer space used by a frame. Line credits can be returned as soon as the frame is successfully transmitted without error to the target link partner.

To return the appropriate type of line credit, the SG1010 implements accumulators for each turn and CoS for each link. The accumulator is incremented based on the link on which the frame was received, the credit type used by the link partner (turn or CoS), the CoS or path (depending on the credit type), and the operation (write/completion or request). If the CoS is asynchronous, high-priority isochronous, or high-priority asynchronous, then these credits are aliased to address-routed, isochronous, and provisioning, respectively, for line credit return. The Line Debit Type bit in the prepended link overhead field of the frame indicates the type of credit (turn or CoS) that was used by the link partner. If the bit is 0, then CoS credits were used; if the bit is 1, then turn credits were used.

Line credits can be returned in the link overhead portion of any frame containing a sequence number that is targeted for that link partner (with the exception of the special bulk credit update frames, Section 3.2.5.1).

3.2.5.1 Bulk Credit Frames

Line credits can also be returned in generated special frames that allows bulk crediting of all line credit types at once. The two types of bulk credit special frames – one dedicated to returning CoS line credits (Special CoS Credit frame) and one dedicated to returning turn line credits (Special Turn Credit frame). This type of special frame must have a sequence number. The SG1010 uses the bulk Special CoS Credit frame to return CoS line credits when the accumulated number of frame credits reaches eight or more credits for four or more CoS or turn categories. The SG1010 returns bulk credit frames in two ways:

- If multiple credit counters have exceeded their thresholds (8 write credits or 4 request credits) a bulk update frame is sent at the next frame boundary provided at least 38 line cycles have elapsed since the last bulk credit frame was sent.
- If there are no frames pending for transmission out a link, the SG1010 sends a bulk credit update frame with whatever credits it has accumulated. It sends these frames at most one out of 38 lines. The SG1010 alternates between CoS and turn credit bulk update frames unless it has no CoS or turn credits, respectively, to return.

Additionally, the SG1010 sends a bulk credit special frame after 15 empty frames have been sent on a given link, as long as there are credits to return. Again, the SG1010 alternates between sending CoS bulk credit special frames and turn bulk credit special frames.

3.3 Frame Ordering and Arbitration

Because the SG1010 may have multiple frames to be sent out any given port, it uses ordering rules and frame arbitration to determine the order in which frames are sent.

3.3.1 Frame Ordering Rules

Frames adhere to the following ordering rules:

- Frames in different classes-of-service are not required to be ordered with respect to each other. However, the SG1010 does deliver in order frames with aliased classes-of-service with the same input and output ports:
  - > Address-routed and asynchronous
  - > Isochronous and HP-Isochronous
  - > Provisioning and HP-Asynchronous
- Frames with different output ports are not ordered with respect to each other.
- Frames arriving on different input ports are not required to be ordered with respect to each other, however the the SG1010 orders them based on the order of arrival.
- Frames with different turns in the next node are not ordered with respect to each other.
- Frames with different Multicast Group IDs are not ordered with respect to each other.
- Configuration read request and write frames are not ordered with respect to other frames.
- For frames within the same CoS using the same output port and turn or Multicast Group ID:
  - – Write frames and response frames are sent in the order that they were received.
  - – Request frames are sent in the order that they were received.
  - – Write frames and response frames may bypass request frames
  - – Request frames cannot pass write or response frames.

3.3.2 Frame Arbitration

At any point in time, the SG1010 may have frames to send to one of its ports from a variety of sources. These sources include:

- Frames received from other nodes to be routed to an output port
- Response frames resulting from SG1010 register accesses
- Path event frames
- Chip event frames
- Frames in the sent list that must be re-sent due to a transmission error
- Software-generated special frames

- Line credit bulk update special frames

Frames in the sent list of a non-bundled link have the highest priority of all routed frames and do not go through the frame arbiter.

3.3.2.1 Frame Arbitration Groups

The SG1010 implements a frame arbiter that chooses which frame to send. SG1010 organizes the arbitration criteria into the following groups for each output port to select from.

- Groups produced by the cross product of three aliased path-routed CoS groups, and eight possible next turns, and whether the frame can be bypassed
  - CoS: Asynchronous, Isochronous/High-priority isochronous, Provisioning/ High-priority asynchronous
  - Turns 0 through 7
- 31 multicast groups
- Two address-routed groups
  - Frames to be transmitted in their order of arrival
  - Frames that have been bypassed by other frames

3.3.2.2 Line Credit Filtering

The first step of frame arbitration is to determine which frames have enough line credits to be sent out a link. If a frame does not have enough line credits to be sent, it will not be eligible for frame arbitration for that link during that time slot. For more information about line credits, see Section 3.2.

3.3.2.3 CoS Arbitration

The second step of frame arbitration is based on the class-of-service. The CoS arbitration uses rotating priority, with each CoS having a different number of entries in the arbiter. There are nine total entries in the CoS Arbiter. Table 3–3 shows the arbiter entries and their relative order of service.

Table 3–3 CoS Arbitration Entries

CoS Arbiter Entry	CoS
0	Provisioning/HP Asynchronous
1	Multicast
2	Provisioning/HP Asynchronous
3	Multicast
4	Isochronous/HP Isochronous
5	Provisioning/HP Asynchronous
6	Multicast
7	Isochronous/HP Isochronous
8	Address-Routed/Asynchronous

Provisioning/HP Asynchronous has three entries in the arbiter, Isochronous/HP Isochronous has two entries, and Address-routed/Asynchronous has one entry. Each slot in the arbitration sequence corresponds to an average of 9 lines of bandwidth. The exact number of lines given to a slot at any time is dependent on traffic patterns, but over time the bandwidth allocation averages to 9 lines per slot. The arbiter changes priority using the following rules:

- If no frames are pending for CoS arbitration, the priorities remain unchanged.
- When frames are pending for CoS arbitration and the current CoS has sent less than its allocated bandwidth (9 lines on average), the priority remains at the current CoS
- When frames are pending for CoS arbitration and the current CoS has sent all of its allocated bandwidth, the priority advances to the next highest numbered entry that has a pending frame. The priority wraps from entry 8 to entry 0

For example, if the CoS arbiter previously selected entry 7 (Isochronous/HP Isochronous), all of the bandwidth has been used for this slot, and there are currently only multiple Multicast and Isochronous frames to send, the CoS arbiter selects entries 1 (Multicast), 3 (Multicast) and 4 (Isochronous/HP isochronous) during the next three arbitration slots.

A typical system with multicast, isochronous, and asynchronous traffic will see the bandwidth allocated such that multicast traffic receives 50%, isochronous traffic receives 33%, and asynchronous traffic receives 17% of the bandwidth (it assumed that provisioning and high priority asynchronous traffic will be rare). In a system without multicast traffic, isochronous traffic receives 67% and asynchronous traffic receives 33% of the bandwidth. These figures also assume that there are enough credits allocated such that credits are not a limiting factor on the bandwidth.

3.3.2.4 Path-routed Turn and Multicast Group Arbitration

If the selected CoS is path routed, then the SG1010 performs turn arbitration based on a simple rotating priority algorithm. The turn that was chosen in the last arbitration for that port is the lowest priority turn for the current arbitration. The turn arbitration is performed separately for each link (time slot).

If the selected CoS is multicast, then the SG1010 arbitrates among the 32 multicast groups based on a similar rotating priority algorithm. The multicast group that was chosen in the last arbitration for that port is the lowest priority multicast group for the current arbitration. The multicast group arbitration is performed separately for each link (time slot).

If the selected CoS is address routed, no further arbitration is performed during that time slot.

3.4 Events

Events in the SG1010 can result from a frame moving through the fabric (*path event*) or a chip event. A *chip event* can indicate that an error has occurred, or it can indicate a notification (information event). The SG1010 does not generate signal events, but may

perform operations on them as they pass through the SG1010 (for more information, see Section 3.4.5). Signal event frames are typically generated by edge nodes and are caused by input signal assertions and deassertions.

The SG1010 generates a *path event* when a path-routed frame encounters a port down, a non-existent output port, or has a path that does not end at an edge node (turn count of 7). A path event indicates to the origin that a path invalidation should be performed. Path events are always directed to the origin of the frame that caused the event.

A *routing event* is a type of chip event that occurs when a frame encounters an error that is not a path event. A routing event is generated in a manner similar to a chip event, however, the path event frame format is used that includes the offending path in the event frame. The SG1010 generates routing events for events such as multicast distribution failures, Address-routing Failures, Channel 255 path protection errors, and Channel 255 range errors.

The SG1010 supports event detection, dispatch, and propagation. The SG1010 does not support *event handling* (this is typically done at edge nodes). *Event detection* involves the detection and capture of the event in status bits at their source. *Event dispatch* is the process of determining where to send the event for handling, and if necessary, generating and sending an event frame. *Event propagation* is the routing of an event frame through the fabric.

All events are identified with a two-tiered coding system consisting of a Primary Event Code and a Secondary Event Code. The five-bit Primary Event Code gives high-level information about the type of event, while the 14-bit Secondary Event Code provides finer level detail on the event. Multiple events can be assigned to one Primary Event Code, and distinguished by their Secondary Event Codes. The SG1010 uses a subset of the Primary Event Codes.

The event codes are used as lookup table indexes during event dispatch. Information obtained from the lookup tables specify the path, CoS, EMU address, and output port of the event frame.

3.4.1 Event Tables

The SG1010 implements two tables that provide information for event frame dispatch. These tables are the Chip Event Table (described in Section 4.6.3.1) and the Event Path Table (described in Section 4.6.3.2). When a chip event is detected, the Primary Event Code is used to index the Chip Event Table. Each entry in the Chip Event Table contains the information described in Table 3–4.

Table 3–4 Chip Event Table Entries

Field	Size	Description
Destination Index	2 bits	An index into the 4-entry Event Path Table
Send Mode	1 bit	When 0, selects polled mode for send event frames to a remote destination. When 1, selects list mode for a chip event frame when a remote destination is specified.
EMU Address	7 bits	Event Message Unit (EMU) address. Bits [6:1] select one of many possible event message units at the destination. Bit [0] selects the operation to be performed by that EMU.

The Destination Index selects one entry in the Event Path Table. The Event Path Table entry specifies where the event is delivered. Each entry in the Event Path Table contains the information described in Table 3–5

Table 3–5 Event Path Table Entries

Field	Size	Description
Path	21 bits	Path specification for the event frame
CoS	3 bits	The class-of-service of the event frame
Output Port	3 bits	The output port used for the event frame
Valid	1 bit	Valid path indication

If the Event Path Table entry is invalid, then a frame is not sent. The Primary Event Code, the EMU Address, and a Secondary Event Code are all part of the event frame header. For more information about generating chip event frames, see Section 3.4.2.

3.4.2 Chip Events

The SG1010 supports two modes of event dispatch – polled and list. These modes are specified on a per event basis in the Chip Event Table. In polled mode, an event frame is sent for every enabled (not masked) chip event, but only if the Event Status bit for that event bit is clear. A read of the device and clearing of the event bit is required for subsequent events of that type to generate frames.

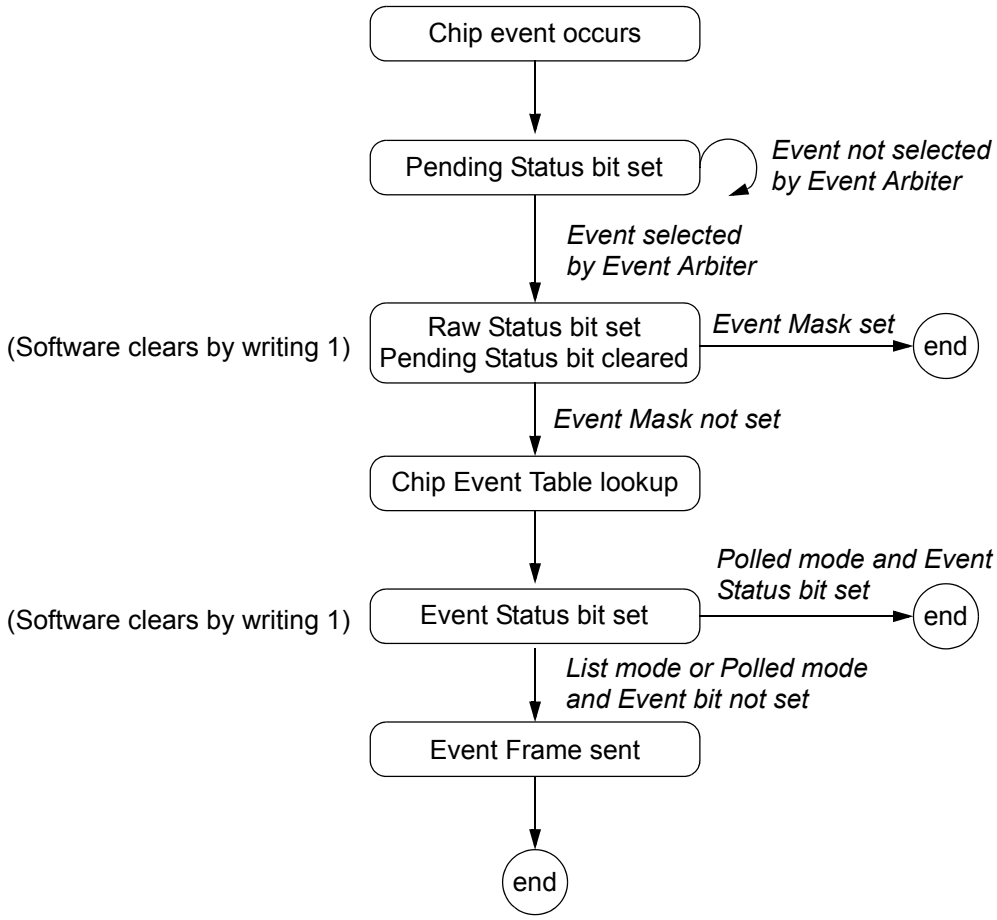
In list mode, an event frame is sent for every enabled chip event, regardless of the state of the Event Status bit. The Event Status bits have no effect on the sending of event frames in list mode, and neither reading the device nor clearing the event bits is required.

Figure 3–7 shows a summary of the process for generating and sending chip events. The following paragraphs describe each step.

3.4.2.1 Pending Events

The SG1010 tracks pending event state for every chip event to determine which events have occurred but have not yet been dispatched. The SG1010 clears the bit when the event arbiter selects the event for dispatch. These bits are not visible to software through accessible register state. If an event occurs more than once before it is dispatched, it is only dispatched once. After an event is dispatched, a subsequent occurrence of that event causes it to be pending again, and dispatched when the event arbiter selects it.

Figure 3–7 Chip Event Dispatch Flow



3.4.2.2 Event Mask

When the event arbiter selects a chip event for dispatch, the SG1010 checks the Event Mask register (described in Section 4.6.3.3) to determine whether to perform a Chip Event Table lookup. This register contains a bit for each chip event. The Event Mask has two register locations. One location is used to set a mask bit through a write-1-to-set operation; the other location is used to clear a mask bit through a write-1-to-clear operation. A read of either location returns the value of the mask register. When a mask bit is set for an event, a Chip Event Table lookup is not performed, an Event Status bit is not set, and an event frame is not sent.

3.4.2.3 Raw Event Status

When the event arbiter selects a chip event for dispatch, the SG1010 sets a corresponding bit in the Raw Event Status register (described in Section 4.6.3.4) regardless of the state of the Event Mask. The Raw Event Status register contains a bit for every chip event. A Raw Event Status bit is cleared by a software write-1-to-clear operation. The state of the bits in the Raw Event Status register do not affect the sending of event frames or the setting of bits in the Event Status register.

3.4.2.4 Chip Event Table Lookup and Event Status Register

When the event arbiter selects a chip event for dispatch, and the corresponding mask bit is not set, a Chip Event Table lookup is performed. This lookup provides the Destination Table index used to select an entry in the Event Path Table. If the entry selected in the Event Path Table specifies list mode, then an event frame is sent; if the entry specifies polled mode, an event frame is sent only if the Event Status bit is a 0 (before being set because of this event).

The Event Status register (described in Section 4.6.3.5) contains a bit for every chip event. When a frame is sent, the Event Status bit corresponding to that event is set.

3.4.2.5 Using Polled vs. List Modes

The list mode may be more desirable when the terminus of the event frame writes information from each event frame to a list in its local memory, and the device is not polled. The polled mode may be desirable when the event causes an interrupt wire to assert at the event frame terminus and the processor reads SG1010 state to determine which events have occurred. Polled mode results in fewer event frames, because setting the Event Status bit blocks any additional frames for that event; clearing the bit arms the dispatcher to send a frame for that event.

3.4.2.6 Routing Events

Routing events are a type of chip event caused by an offset, channel, or path check on a frame. The frame format of a routing event uses the same format as the path event, in that included in the event frame is the path and the relative position of the event frame output port with respect to the input port of the frame that encountered the error. If the output port of the event frame and the input port of the frame causing the event are the same, the Input Port Flag bit is set. The width of the Secondary Event field code for routing events is reduced to three bits.

When routing events are pending, the path of each event must be saved. The SG1010 implements an four-entry stack for each routing event. Each entry contains the path and input port information for the routing event. When a routing event is dispatched, the next entry in the stack is used. If a routing event occurs at the SG1010, and the stack is full for that type of routing event, then the SG1010 sets the Event Overrun status bit and does not generate a routing event for that case.

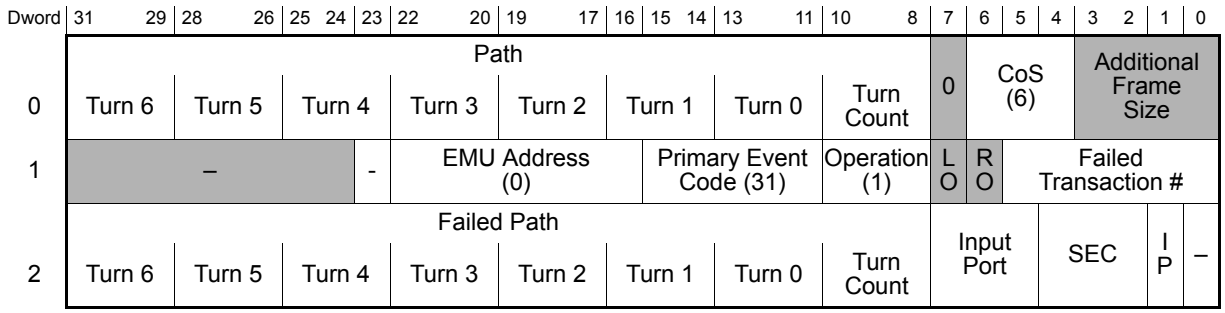
3.4.3 Path Events

A path event is a result of a path-related error during a path-routed frame transmission. A path event can result in path invalidation at the originating node. Multicast frames, address-routed frames, and response frames (write acknowledge, read completion, event frames, bandwidth response frames) cannot cause path events to be generated. When a path event occurs, an event frame is sent to the origin of the frame that caused the error. The Chip Event Table and Event Path Tables are not used. Path events are:

- Bad Path – the SG1010 receives a path-routed frame with a turn count of 7 (except for Channel 255 provisioning frames, which access SG1010's register space)
- Port Down – a path-routed frame encounters an inoperative port; no traffic can be sent or received using that port

The SG1010 creates an event frame with a Primary Event Code of 31. The EMU Address is set to 0. The Secondary Event Code indicates the type of path event that was encountered and is shown in Table 3–7. The event frame also includes the path (as received) and the input port where the frame that encountered the error was received. The SG1010 performs a path transform on the path of the original frame to generate the path to the origin (the path transform includes the bad turn as an active turn).

Figure 3–8 Path Event Notification Frame Header



EMU = Event Message Unit
SEC = Secondary Event Code
IP= Input Port flag

3.4.4 Event Dispatch without Event Tables

The Chip Event Table is used to dispatch chip event frames only if the Chip Event Table Enable bit (Event Dispatch Control CSR [0], Section 4.6.3.7) is set by software. When the Chip Event Table Enable bit is not set, the root is assumed to be the destination for all events, and the provisioning class-of-service is used for all event frames. Polled mode is assumed when sending event frames when the Chip Event Table Enable bit is not set.

The default EMU address is 02h. One exception is the "Master abort received on write without acknowledge" event, which uses EMU address 0Eh if the SERR# Enable bit in the PCI Command configuration register is set; otherwise, 02h is used.

3.4.5 Signal Events

The SG1010 does not generate or handle signal events. However, to support PCI compatibility for interrupts generated by devices behind a P2P bridge, the SG1010 detects incoming signal event frames and conditionally modifies the EMU address. This modification supports the interrupt wire-OR mechanism described in the *PCI-to-PCI Bridge Architecture Specification, Rev 1.1*. For each level of bridge hierarchy, the four INTA#, INTB#, INTC# and INTD# signals coming from different devices are wire-ORed together in a staggered scheme based on the PCI device number, called a “swizzle”.

Primary event codes 2 and 3 identify a signal event assertion and deassertion, respectively. The secondary event code identifies the signal that was asserted or deasserted. Secondary event codes 8, 9, A, B are assigned to INTA#, INTB#, INTC# and INTD# when a swizzle operation is to be performed. The SG1010 detects an event frame with these event codes and modifies the EMU address to make it appear like an interrupt swizzle has occurred. Because the PCI device number of the downstream link partner is equal to the input port number, the EMU modification is based on the input port of the signal event frame. This modification only occurs when the EMU address is one preassigned to the PCI INT*x*# signals (EMU addresses 2 through 9). Table 3–6 shows the EMU address operation performed by SG1010.

Table 3–6 PCI INT*x*# EMU Address Swizzle Modification

Incoming EMU Address	Outgoing EMU Address* Input Port (PCI Device Number):			
	0,4	1,5	2	3
0	0	0	0	0
1	1	1	1	1
2	2	4	6	8
3	3	5	7	9
4	4	6	8	2
5	5	7	9	3
6	6	8	2	4
7	7	9	3	5
8	8	2	4	6
9	9	3	5	7
Others = X	X			

* only for Secondary Event Codes 8 – B

3.4.6 Event Codes

Table 3–7 defines the Primary Event Codes and Secondary Event Codes for SG1010 events. Primary Event Codes 0 through 26, and code 31 are predefined, while Primary Event Codes 27 through 30 are device-specific assignments.

Table 3–7 Primary Event Code and Secondary Event Code Assignments

Event Type	Primary Event Code	Event	Secondary Event Code
Write Acknowledge	0	None	N/A
Write Message	1	None	N/A
Architected StarFabric Events			
Input Signal	2	Input signal assertion (not used by SG1010)	
	3	Input signal deassertion (not used by SG1010)	
Link Events	4	Link x Down	[8:0] = 0 [13:9] = Link #
		Link x Fragile	[8:0] = 1 [13:9] = Link #
		Link x Up	[8:0] = 2 [13:9] = Link #
		Link x CRC Counter Wrap	[8:0] = 3 [13:9] = Link #
		Link x 8B/10B Counter Wrap	[8:0] = 4 [13:9] = Link #
		Link x Frame Count Wrap	[8:0] = 6 [13:9] = Link #
		Link x Line Count Wrap	[8:0] = 7 [13:9] = Link #
		Link x Empty Line Wrap	[8:0] = 8 [13:9] = Link #
Port Events	5	Port x Down	[9:0] = 0 [13:10] = Port #
Routing Events	6	Channel address range error	[2:0] = 0
		Channel path protection error	[2:0] = 1
		Address-routing Failure (non-cfg)	[2:0] = 3
		Fabric Special Cycle	[2:0] = 4
		S/W Multicast Distribution Failure	[2:0] = 5
		H/W Multicast Distribution Failure	[2:0] = 6
Multicast Errors	7	Unsupported Multicast Group ID	0
SGF	8	SGF Done	0
Events	9	Event Overrun	0
SERR# Condition at node	10	Address-routing Failure on write w/o ack	1
PCI Status	11	Received Master Abort	2
Reserved	12 – 26	Reserved	–

Table 3–7 Primary Event Code and Secondary Event Code Assignments *(Continued)*

Event Type	Primary Event Code	Event	Secondary Event Code
Device-Specific Events			
Reserved	27 – 30	Reserved	–
Path Events			
Path Invalidation	31	Down port	0
		Reserved	1
		Bad path (turn count = 7)	2

3.5 Reset and Initialization

SG1010's initialization process consists of the following components:

- Reset
- Link synchronization
- Reset propagation
- Serial preload
- Fabric Enumeration
  - Component identification
  - Fabric ID assignment
  - Link to port mapping
- Line credit initialization

3.5.1 Reset

The SG1010 supports the following reset mechanisms:

- NRST_L input signal – power-up node reset.
- Fabric Reset bit in the Fabric Reset register – resets the chip and propagates an unmaskable reset.
- Node Reset bit in the SFC Control register – resets the chip
  - If Propagate Maskable Reset bit in the same register is also written with a 1, a maskable reset comma is sent out all links prior to the chip reset
- Secondary Reset bit in Bridge Control configuration register – resets the Port Map Tables and propagates an address routing reset
- Power Management state transition from $D3_{hot}$ to D0 – resets standard PCI registers and Port Map Tables and propagates an address routed reset
  - standard PCI registers are all configuration R/W and R/W1TC registers with configuration offsets between 00h - 3Fh
- Maskable reset comma character – propagated reset received from the link interface

  - resets the chip if the Reset Disable bit is clear for the input port

- Unmaskable reset comma character – propagated reset received from the link interface
  - unconditionally resets the chip
- Address routed reset comma character – resets standard PCI registers and Port Map Tables

Table 3–8 shows which initialization components are performed as a result of each of SG1010's reset mechanisms.

Table 3–8 SG1010's Reset Mechanisms

Mechanism		Link Sync and Credit Init	Propagation	Serial Preload
H/W	NRST_L	Yes	No	Yes
	Maskable Reset Comma	Yes	Maskable Comma	Yes
	Unmaskable Reset Comma	Yes	Unmaskable Comma	Yes
	Address routed reset comma	No	Address Routed Comma	No
S/W	Fabric Reset	Yes	Unmaskable Comma	Yes
	Node Reset	Yes	No*	Yes
	Secondary Reset	No	Address Routed Comma	No
	PM D3 → D0	No	Address Routed Comma	No

* Unless the Propagate Maskable Reset bit is also written with a 1

The above reset mechanisms, except Secondary Reset, Power Management reset, and address routed reset, clear all of SG1010's state and data buffers.

NRST_L is a local hardware chip reset and is not propagated into the fabric.

The Secondary Reset bit, located in the Bridge Control configuration register, resets standard PCI registers in downstream nodes and resets downstream PCI devices. Because the SG1010 Port Map Tables contain address decoding data for downstream nodes, when the Secondary Reset bit is set, the SG1010 clears all the Port Map Table registers. The Port Map Table Enable and Smart Address Enable bits are not reset.

The Power Management reset occurs when software changes the power management state from the D3 low power state to the D0 high power state. When this power management state transition occurs, the SG1010 clears standard PCI registers with R/W and R/W1TC access. Standard PCI registers are those located within configuration offsets 00h and 3Fh, as well as the Power Management enhanced capability function. The SG1010 transmits an address routed comma out both links.

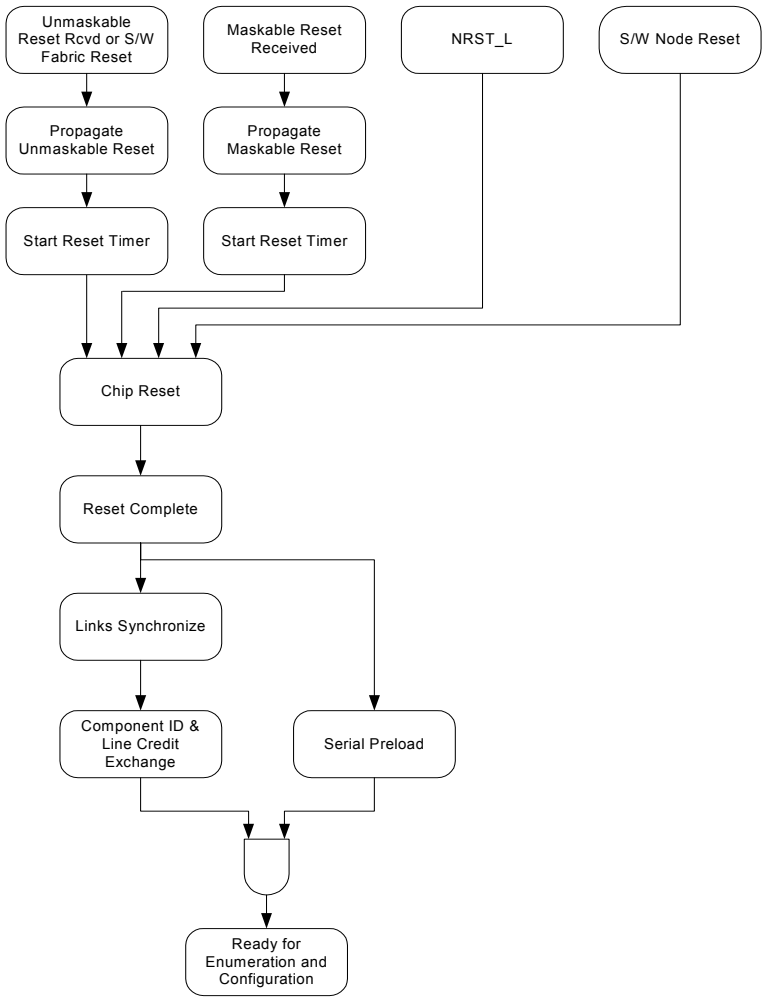
If an address routed comma is received by the SG1010, and the Reset Disable bit in the Port State Control and Status register for that port is clear, then the standard PCI registers with R/W and R/W1TC access are reset to their default values and an address routed reset is propagated out all ports. If the Reset Disable bit is set, the address routed reset comma is ignored.

A node reset propagates a maskable reset comma if the Propagate Maskable Reset bit in the SFC Control register is written with a 1 with the same access. Reset propagation is described in Section 3.5.1.1. If a reset does not propagate, only the SG1010 is reset. All state and data buffers are cleared. SG1010 link partners participate in link resynchronization and line credit exchange as a part of the SG1010 initialization flow.

The SG1010 performs serial preload, link synchronization, component identification, and line credit initialization after every type of reset except for Secondary Reset, Power Management reset, and address routed reset.

Figure 3–9 is a summary of the initialization flow. Subsequent sections describe each step in more detail.

Figure 3–9 SG1010 Initialization Flow



3.5.1.1 Reset Propagation

Reset propagation through the StarFabric relies on the transmission and reception of comma characters (well-defined 10-bit encodings) between link partners. StarFabric protocol defines two comma characters for reset propagation, a maskable reset comma character, and an unmaskable comma character. Links must be synchronized to propagate a reset, but the Traffic Enable bit does not need to be set. StarFabric protocol defines two comma characters for reset propagation: a maskable reset comma and an unmaskable reset comma. Received unmaskable comma characters unconditionally reset the node. Received maskable comma characters reset the node only if the Reset Disable bit (Port State Table, Port State *x* Control and Status CSR [6], Section 4.6.2.2) corresponding to that link is clear. If the Reset Disable bit is set, the maskable reset comma character is ignored and not propagated.

Address routed reset commas are also masked with the Reset Disable bit. If the bit is clear, the SG1010 propagates the address routed reset comma out all links when one is received.

3.5.1.1.1 Sending Propagating Resets

To propagate a maskable reset comma, an unmaskable reset comma, or an address routed reset, the SG1010 sends four comma characters in succession out each of its differential pairs for all links. When this transmission occurs, the SG1010 starts a Reset Mask Timer. The Reset Mask Timer prevents infinite reset loops caused by reset propagation by filtering all received reset comma characters. Until the Reset Mask Timer expires, the SG1010 ignores all subsequent reset comma characters.

3.5.1.1.2 Receiving Propagating Resets

To detect an incoming propagating reset on a link, the SG1010 must detect at least four comma characters on at least one of its operational differential pairs. Otherwise, the comma characters are ignored.

When SG1010 receives a unmaskable reset, the SG1010 transmits the unmaskable reset out all of its differential pairs and starts the Reset Mask Timer as described in Section 3.5.1.1.1. It then performs a chip reset followed by the rest of initialization flow.

When the SG1010 receives a maskable reset on a link, the SG1010 first checks the Reset Disable bit (Port State *x* Control and Status CSR [6], Section 4.6.2.2) in the corresponding Port State Table. If the Reset Disable bit is 1, the SG1010 ignores the maskable reset and does not propagate it. If the Reset Disable bit is 0, the SG1010 sends a maskable reset out all of its differential pairs and starts the Reset Mask Timer as described in Section 3.5.1.1.1. It then performs a chip reset followed by the rest of the initialization flow.

The Reset Disable bit is reset to 0, which allows any link partner to reset the SG1010. Fabric enumeration then sets all Reset Disable bits except for the one(s) corresponding to the root port. This allows a propagating reset to flow downstream through the tree hierarchy but not in any other direction.

Software may also modify the Reset Disable bit for any link.

3.5.2 Serial ROM Preload

The SG1010 optionally performs a serial preload operation to write register state in SG1010 after reset. All registers with write access, and selected read-only registers, can be written through the serial preload operation. Preload register access is described in Section 4.3.

The first byte read during the first part of preload contains the preload enable sequence. If a valid preload enable sequence is not read, then the preload operation is not performed.

The SG1010 can forward frames during preload and buffers register accesses until preload is complete. However, the diagnostic port cannot be used to read or write registers until serial preload is complete.

3.5.3 Link Synchronization

After the SG1010 is reset, it attempts to synchronize its six links with their respective link partners. Each link has its own synchronization state machine. The receiver controls these synchronization states. The transmitter uses this state information when sending synchronization frames.

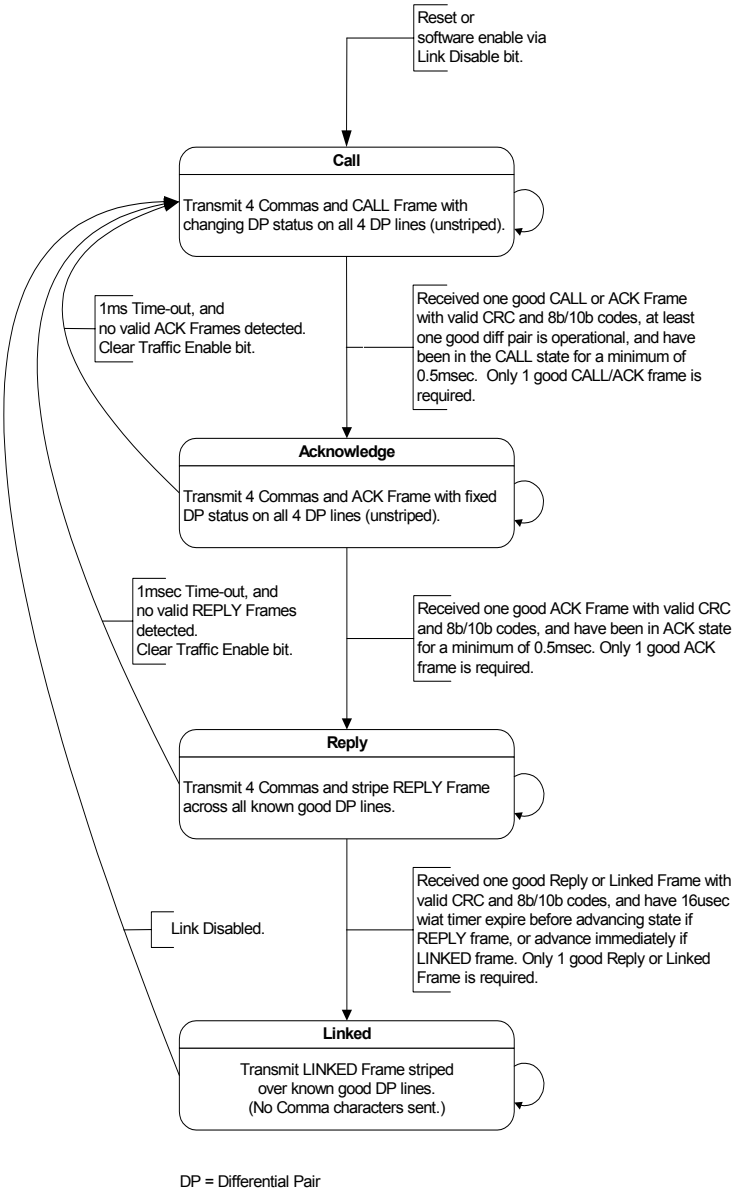
The four link synchronization states are Call, Acknowledge, Reply, and Linked. When in one of these states, a Special Link Synchronization frame is transmitted with the proper Behavior field bits set to indicate the current synchronization state. These special frames do not require sequence numbers. Figure 3–10 shows the fields in the frame header.

Figure 3–10 Special Link Synchronization Frame Header

DP = Differential Pair

Figure 3–11 shows the link synchronization flow, which is described in the following paragraphs.

Figure 3–11 Link Synchronization State Flow



DP = Differential Pair

3.5.3.1 Call State

The Call state achieves byte alignment on each of the four 622Mbs differential pairs in a link. The Call state is entered during reset, or the transmitter turns on. In the Call state, each 622Mbs differential pair comprising the link continuously sends a Special Link Synchronization frame indicating Call State, separated by four K28.5 symbols. SG1010 continues this transmission for a minimum of 31250 cycles, where a cycle is a single 62.208MHz clock period. If the receiver detects either Call or Acknowledge status with good CRC on any of the four 622Mbs differential pairs in the time period between 31250 and 62500 cycles, then the link state machine transitions to the Acknowledge state. If none of the pairs are receiving Call or Acknowledge status with good CRC in this time period, then the transmitter stops transmitting.

3.5.3.2 Acknowledge State

The Acknowledge state verifies reception of good CRC before the transmitter begins to stripe the data over all of the operational differential channels, and detects which of the differential pairs are working.

Data is still transmitted independently on each differential pair. During the Acknowledge state, the transmitter sends four bytes of K28.5 symbol between Special Link Synchronization frames. The Behavior field in the special frame is set to Acknowledge. The special frame also contains information about which differential pairs are successfully transmitting good CRC.

Again the minimum transmission time is 31250 cycles and the maximum time of reception of the acknowledge state from its link partner is 62500 cycles. If an Acknowledge state is not received from it link partner than the transmitter stops transmitting.

If the Acknowledge state with good CRC is detected on the link through the received Special frame, then the link state machine transitions to the Reply state. If the acknowledge is received earlier than 31250 cycles, then the transition to the Reply state occurs when the 31250 cycle point is reached, but not before, regardless of what is received at that timer expiration point.

3.5.3.3 Reply State

In the Reply state, the transmitter stripes data across 1, 2, or 4 working differential pairs. The transmitter sends four K28.5 bytes between special frames with the Reply state indicated in the Behavior field. The link waits for 62500 cycles to receive the Reply special frame from its link partner with good CRC. If the timer expires before a Reply special frame is received, then the link stops transmitting. If a Reply frame is received, then the transmitter starts a 1000 cycle timer and moves to the Linked state when it expires. This delay allows sufficient time for its link partner to achieve alignment across its striped channels before the K28.5 characters are removed. If a Linked special frame is received from the link partner, the link state machine moves directly from Reply to Linked without a delay.

3.5.3.4 Linked State

In the Linked state, the transmitter sends special frames with the Behavior field set to Linked. In this case, the Special Link Synchronization frame is also known as the Empty frame. This is the state of normal operation for the link, which can transmit data at any time.

3.5.3.5 Disabling a Link

Software can enable or turn off a link through the Link Disable bit in the Link Control and Status register in CSR space (Section 4.6.1.1). The transmitter is always disabled when software brings a link down. It will not transmit even if the receiver detects an incoming link synchronization frame.

However, when the Link Disable bit is 0, the transmitter may be stopped or started depending on the results of link synchronization. If the link fails to synchronize, the transmitter stops, but periodically turns on to attempt to synchronize. The transmitter attempts to synchronize for 1ms every 500ms. In addition, the transmitter will turn on and attempt to synchronize if a good Call frame is received from the link partner. If the subsequent synchronization is successful, the link remains enabled. If the synchronization is not successful, it turns off again.

After reset the link is automatically enabled and begins the link synchronization sequence.

3.5.3.6 Clock Comma Propagation

The SG1010 supports the propagation of a clock through the fabric using the reception and transmission of a special comma character. The clock comma is transmitted and received in the same way as an unmaskable reset comma.

The SG1010 detects the clock comma when four clock comma characters are received on at least one of its operational differential pairs. Otherwise, the comma characters are ignored. When the SG1010 receives a clock comma, the SG1010 transmits the clock comma out all of its differential pairs on every link and starts the Clock Mask Timer. The SG1010 ignores any clock comma characters received on any link while the Clock Mask Timer is counting. The Clock Mask Timer expires after 72.8µsec. Once the Clock Mask Timer expires, the SG1010 is enabled to receive and propagate a clock comma from any link.

3.5.4 Fabric Enumeration

Fabric enumeration assigns the port to link mapping, fabric ID (FID), root assignment and reset disables. Fabric enumeration can occur during or after serial preload – there is no timing relationship between these two initialization processes.

Fabric enumeration is performed by the transmission and reception of special frames. These special frames are:

- I Am frames for identification of the attached component types, bundled links, and ports that are a part of the tree hierarchy
- You Are frames for assignment of the Fabric ID (FID)
- Set Credit frames initialize the credit counters of a node's link partners.

These special frames do not require sequence numbers, nor are credits used to send them. The SG1010 uses transmission timers in order to ensure the transmission of these special frames – one per link for I Am frames, one per link for set credit frames, and one for all links for You Are frames. Whenever one of these special frames is sent out any link, the corresponding transmission timer resets and then starts to count. If a transmission error occurs before the timer expires, then the appropriate special frames of that type are re-sent.

There is no ordering relationship between special frames sent out different links.

The following sections describe the various components of fabric enumeration in more detail.

3.5.4.1 Component Identification

Component identification indicates the component type of a node's link partners, as well as the current FID of the device. Component identification primarily involves the sending of an I Am frame.

The SG1010 sends an I Am frame out each link when any of the following conditions occur:

- After the link synchronize after a reset deassertion
- When the product of the Traffic Enable bit and the Link Partner Traffic Enable bit transitions from a 0 to a 1, or from a 1 to a 0.
- If the link state transitions to the Linked state after it was in the Call state, and the product of the Traffic Enable bit and the Link Partner Traffic Enable bit is 0 (link up)
- After receiving a You Are frame on any link

When the SG1010 sends an I Am frame, it includes the component type (Edge Node=No, Switch=Yes) and current FID. The Address-Routing Support bit in the I Am frame is set to 0 (indicating address routing support). The Traffic Disable (TDIS) bit reflects the inverted state of the Traffic Enable bit. After reset, the Traffic Enable bit is 1, and the TDIS bit is 0.

The SG1010 performs several updates when an I Am special frame is received on a link. The attached component type value is copied into the Link State Table Control and Status register. The Fabric ID and inverted Traffic Disable[4] values are copied into the Link Partner FID register in the corresponding Link State Table entry.

If the FID received in the frame matches an FID received on another link, then the SG1010 bundles the two links, and updates the Port State and Link State tables to reflect the new link to port mapping. The SG1010 uses only enumerated FIDs (i.e., not equal to 7/7/7777777) to determine bundles.

The SG1010 performs a dynamic bundled link check to make sure that the link mapping is not changed after the fabric has been enumerated. If the two links were not previously bundled, the Link Partner FIDs are not 7/7/7777777, and an I Am frame is received on one link that has an FID matching the other link's Link Partner FID, then a bundling error has occurred (formerly unbundled links were bundled). A bundling error code of 001b is indicated in the Link Partner FID register. Similarly, if the two links were bundled, the Link Partner FIDs are not 7/7/7777777, and an I Am frame is received on one link that has an FID that does not match the other link's Link Partner FID, then a bundling error has occurred (formerly bundled links were unbundled). A

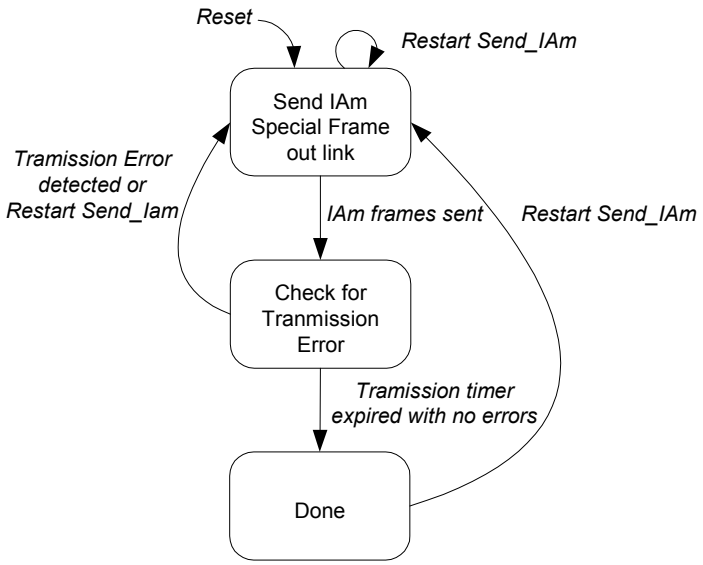
4. This inverted value is the Link Partner Traffic Enable bit.

bundling error code of 010b is indicated in the Link Partner FID register. If a bundling error occurs, the Link Partner FID and the attached component type fields are not updated, but the Traffic Disable value is updated.

If the FID in the I Am frame matches a FID assigned by the SG1010 in a previously sent You Are frame, then that link, and any bundled links, are considered to be a child of the SG1010 in the StarFabric tree hierarchy and the Port Map Table Enable bit for that port is set.

Figure 3–12 shows the component identification flow for a given link. This flow is performed for each synchronized link.

Figure 3–12 Send I Am Component Identification Flow



3.5.4.2 Fabric ID Assignment

This part of fabric enumeration assigns a FID to every node and also constructs a tree hierarchy within the fabric to be used for address-routed traffic. Fabric enumeration is initiated by the root or through a software generated frame to a link partner and can occur at any time after link synchronization. The root determines whether enumeration is performed; and if so, the timing of fabric enumeration.

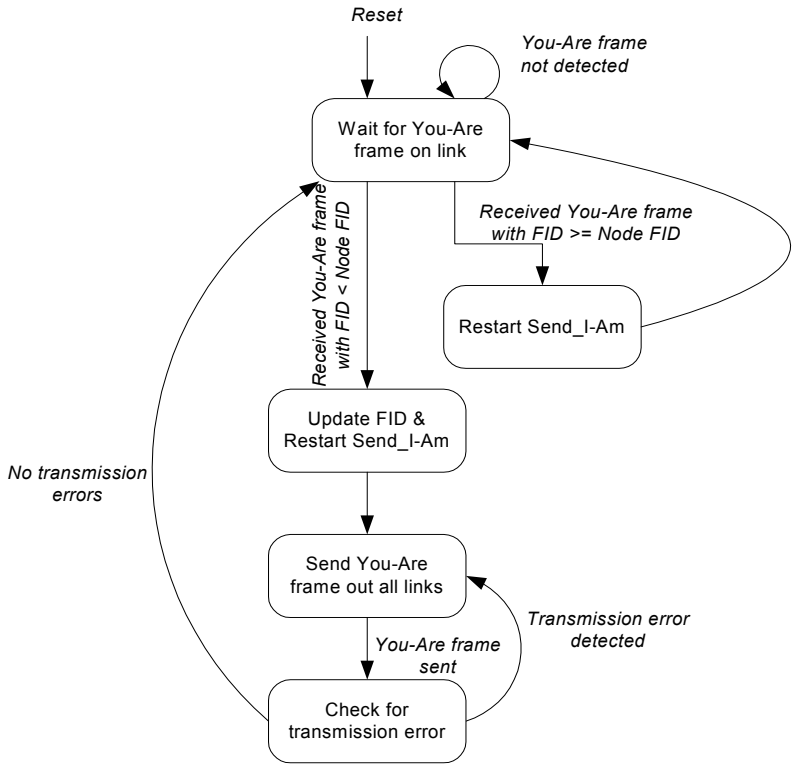
The fabric ID has the following components:

- Parallel fabric number (PFN) – used when two or more nodes are connected to the root edge node, forming parallel fabrics
- Turn count – incremented at each switch similar to the turn count of a path. All devices with the same turn count will also have the same PCI bus number (although the turn count is not necessarily equal to the bus number). The turn count indicates the number of valid turns in the FID.
- Turns – uniquely identifies the downstream link partners with the same turn count

These components of the FID are identified using an **PFN/turn counts/turns** nomenclature. During reset, the SG1010 initializes its FID to 7/7/7777777 (octal).

Figure 3–13 shows the flow for FID assignment.

Figure 3–13 Fabric ID Assignment Flow



The FID assignment starts when a You Are special frame is received on one of SG1010's links. The SG1010 compares the FID received in the frame to its current FID. If the received FID is less than the current FID, then the SG1010 updates its FID with the new one. The SG1010 performs the comparison and conditional FID update in the following order:

- PFN comparison – if the new PFN is:
  - less than the current PFN, the FID is updated and no additional comparisons are necessary.
  - greater than the current PFN, the FID is not updated and no additional comparisons are necessary.
  - equal to the current PFN, then the Turn Count is compared.
- Turn count comparison – if the new turn count is:
  - less than the current turn count, the FID is updated and no additional comparisons are necessary.
  - greater than the current turn count, the FID is not updated and no additional comparisons are necessary.

  - equal to the current turn count, then the Turn value is compared.
- Turn comparison – if the new turn is:
  - less than the current turn, the FID is updated.
  - greater than or equal to the current turn, the FID is not updated.

Regardless of whether the FID is updated, the SG1010 restarts the Component Identification state machine to inform all of its link partners of its current identity (FID).

The SG1010 updates the following registers when a You Are frame is received:

- Fabric ID
- Root Port
- Reset Disable – set for all ports except the root port

If the SG1010 has updated its own FID, it then attempts to assign new FIDs to its link partners by sending You Are frames. The product of the Traffic Enable bit (Link State Table Control and Status register) and the Link Partner Traffic Enable bit (Link Partner FID register) must be 1 in order to send a You Are frame. If an update was not performed, a You Are frame is not sent. After restarting the Component Identification state machine, the SG1010 sends a You Are special frame out each of its links. These frames contain FIDs that SG1010's link partners will use to compare and potentially update their FIDs. The algorithm for creating the FIDs to be sent is:

- The PFN is the same as SG1010's PFN.
- SG1010's turn count field is incremented to create the new turn count.
- The relative turn for the outgoing link from the newly assigned root link is placed in the turn position indexed by SG1010's current turn count of the newly assigned FID. If the root link is bundled, the lowest numbered link is used.

This algorithm imposes a restriction on how the links in a bundle are connected in order for the FIDs to be assigned properly, such that a path to the root can be constructed. When a bundled link is connected, the lowest link number of the bundle on the SG1010 must be connected to the lowest link number of the bundle on the link partner. For example, if links 2 and 4 are connected to links 3 and 5 at the link partner, the SG1010's link 2 must be connected to the link partner's link 3, and the SG1010's link 4 must be connected to the link partner's link 5. If link 2 is connected to link 5, and link 4 is connected to link 3, the path transform of the FID assigned to the link partner may be an invalid path.

Note: *This manual covers SG1010's implementation of the hardware enumeration process. For a complete description of the hardware enumeration process, see the StarFabric Architecture Specification.*

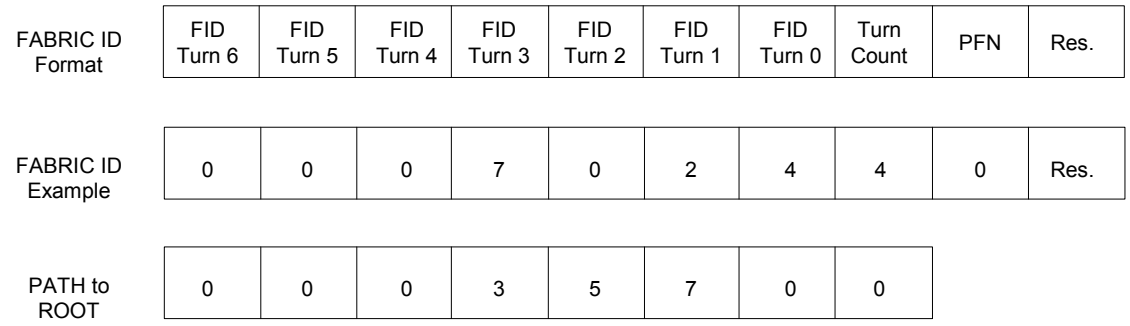
3.5.4.3 Using the Fabric ID as a Path to the Root

The FID can be manipulated to obtain a path to the root. The Event Dispatch logic performs this transform when generating event frames in default mode. The transform uses the turn count and seven turn values of the FID. Each of these components is a three-bit

octal number. To derive the path back to the root, the active turns are reversed and inverted (path transform). The turn count indicates the number of active turns, where an FID turn count of 4 indicates four active turns. The output port is not reflected in the path transform. The Root bit in the Port State Table determines the output port towards the root.

Figure 3–14 is an example of how a path to the root is created from the FID. In this example, there are four active turns. A path transform is performed on the first four turns (FID turns 7 0 2 4 are transformed into 3 5 7 0). The remaining inactive turns remain as 0.

Figure 3–14 Root Path Example

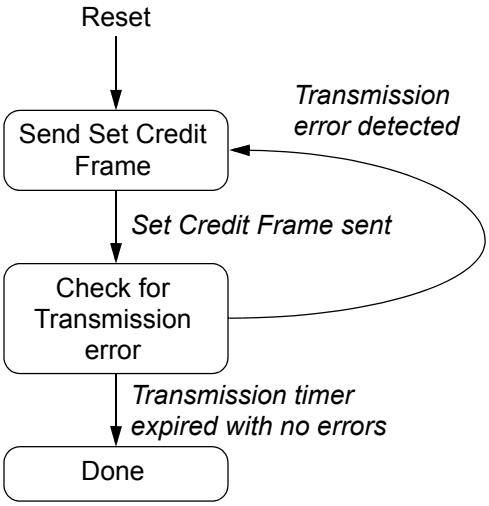


3.5.4.4 Line Credit Initialization

After reset and link synchronization, the SG1010 initializes the line credit counters of its link partners. SG1010's link partners use these credits when sending credited frames to the SG1010. The SG1010 sends a Cos Set Credit special frame containing class-of-service line credit information out each newly synchronized link after a chip reset.The SG1010 also sends a Set Credit frame whenever the product of the Traffic Enable bit (Link State Table Control and Status register) and the Link Partner Traffic Enable bit (Link Partner FID register) transitions from a 0 to a 1.

Figure 3–15 shows the set credit flow used for each link.

Figure 3–15 Set Credit Flow



The link partner loads its line credit counters with the values contained in the CoS Set Credit frame. The SG1010 does not send a Turn Set Credit frame, so those counters remain at 0 for the link partner. If use of turn credits is desired, software must perform a line credit reallocation. Table 3–9 shows the number of credits per link assigned at reset.

Table 3–9 Initial Line Credit Values for the SG1010

	Credits	
CoS Credit Type	**Write/Completion**	**Request**
Asynchronous	0	0
Isochronous	60	7
HP-Asynchronous	0	0
Multicast	60	–
Address-routed	90	7
HP-Isochronous	0	0
Provisioning	60	7

The SG1010 also receives Set Credit special frames from its link partner(s) on each synchronizing link. The link partner must always send a CoS Set Credit frame, and may optionally send a Turn Set Credit frame. The SG1010 initializes its line credit counters for each link with this information. Because the SG1010 performs line credit aliasing, when the SG1010 receives the set credit information it adds the credits from the following CoS pairs together and loads only the counter of the first CoS:

- Address-routed plus asynchronous – loaded into address-routed credit counters
- Isochronous plus HP-isochronous – loaded into isochronous credit counters
- Provisioning plus HP-asynchronous – loaded into provisioning credit counters

If the total credits added together exceed the maximum value for that credit counter, then the maximum value (127 for write credits, 15 for request credits) is used.

3.5.5 PCI Configuration

When using standard PCI initialization, a host performs standard PCI configuration of the fabric as it would any PCI hierarchy. Configuration transactions are propagated through the fabric as address-routed read request and write frames. To forward a Type 0 configuration frame to another node, a node's Port Map Table Enable bit must be set and its Smart Address Enable bit must be clear.

3.5.5.1 Port Map Table Initialization

The SG1010 Port Map Table contains PCI address decode information for its link partners. The Port Map Table is implemented on a per-port basis. The SG1010 requires this information to determine which exit port to use when forwarding a downstream or peer-to-peer PCI address-routed frame through the PCI hierarchy. The Port Map Table consists of copies of PCI configuration registers used for address decoding from its link-partners.

The SG1010 snoops configuration writes to these registers when one of its downstream link partners is the target. This occurs when the SG1010 sends a Type0 configuration write frame out a port. The SG1010 captures the write data and copies it into the corresponding Port Map Table register.

If the above configuration registers are updated through another means, such as Channel 255 write, or PCI memory or I/O write, snooping is not performed. In this case software must insure that the Port Map Tables are consistent with the port partner's configuration registers.

3.5.6 Root Port Modification

The SG1010 allows software to modify which port is the root port. The Root bit in the Port State Table can be written by software. The Root bit in the corresponding Link State Table cannot be written, however hardware modifies this bit to be consistent with the value in the Port State Table.

When the root port is changed, the output port for inversely decoded address-routed frames changed accordingly. Inversely decoded address-routed frames are forwarded out the current root port. Only one port at a time can be designated as the root port. Changing the root port while address-routed traffic is flowing can cause unpredictable results.

Software is responsible for managing all other side effects of changing the root port, which may include:

- Modifying the Reset Disable bit in the Port State Table
- Modifying the Port Map Table, and the Port Map Table Enable bit in the Port State Table
- Modifying the Event Table and Event Path Table

3.6 Link and Port Operation

3.6.1 Link and Port Conditions

Port and link state is reflected in the Port State Table (Section 4.6.2) and Link State Table (Section 4.6.1) registers, respectively. Port state is a logical condition and indicates whether a port can transmit credit-based traffic. Link state is a physical condition and indicates how many of its transmit and receive differential pairs are synchronized.

In order for traffic to be sent out a link, both the Traffic Enable (TEN) and the Link Partner Traffic Enable (LP_TEN) bits for that link must be 1. In other words, the product of TEN and LP_TEN must be 1. The TEN bit is located in the Link State Table Control and Status register. The LP_TEN bit is located in the Link Partner Fabric ID register. Only link synchronization frames, I Am frames, and Set State frames may be sent when the product of TEN and LP_TEN is 0.

3.6.1.1 Link Up

A link is up when at least one transmit differential pair and one receive differential pair are synchronized; but the link is also fragile if less than four transmit and receive pairs are synchronized (see Section 3.6.1.3).

A link up event occurs when a link goes from the Call synchronization state to the Linked synchronization state, and the product of the TEN bit and the LP_TEN bit is 0. When a link up event happens, the following flow occurs:

- The SG1010 sends an I Am special frame with the TDIS (Traffic Disable = !Traffic Enable) bit = 1
- The SG1010 receives an I Am frame from its link partner
  - The Fabric ID in the frame is stored in the Link Partner Fabric ID register
  - The TDIS bit is inverted and stored in the Link Partner Traffic Enable (LP_TEN) bit in the Link Partner Fabric ID register
  - Reception of the I Am frame does not affect the current values of the Port Map Table Enable in the Port State Table if the TEN bit is clear. Software is responsible for making sure these values are appropriate.
- After sending the I Am frame, the SG1010 signals a Link Up event

Software must then intervene to bring the link the rest of the way up by writing the Traffic Enable bit in either the SG1010 or the link partner. If software writes the Traffic Enable bit in the SG1010, the SG1010 sends a special Set State frame to its link partner with Set_TEN=1. Otherwise, if software writes the Traffic Enable bit in the link partner, the link partner sends the special Set State frame to the SG1010.

If the SG1010 receives a Set State frame with Set_TEN=1, then the SG1010 sets both its Traffic Enable bit, and the LP_TEN bit in the Link Partner FID register. Upon reception of the Set State frame and setting of these bits, the SG1010 returns an I Am frame to the link partner with TDIS=0 (that is, Traffic Enable is 1).

If the SG1010 sent the Set State frame, then the link partner sends an I Am frame back to the SG1010. The TDIS bit should be 0, indicating that the link partner's Traffic Enable bit is set. The SG1010 then updates the Link Partner Fabric ID register with both the Fabric ID of the frame and the inverted TDIS bit to indicate the state of LP_TEN.

When both the Traffic Enable bit and the LP_TEN bits are 1, the SG1010 sends a Set Credit frame to its link partner with default credit values. The SG1010 also resets both its receive and transmit sequence numbers. The link is now ready to send and receive traffic. Software must ensure that both Traffic Enable and LP_TEN bits are 1 before it considers the link to be fully up.

When links are bundled, and software writes the TEN bit of one link, the TEN bit of the other link is also set. Set State and Set Credit frames are sent out both links in the bundle as described above.

Software may attempt to bring a link up from a down state by writing a 0 to the Link Disable bit, which turns on the link's transmitter. If the link synchronizes successfully, a Link Up event is signaled and the link up process described above must be followed.

3.6.1.2 Link Down

A link down occurs when the product of the Traffic Enable bit (Link State Table Control and Status register) and the Link Partner Traffic Enable bit (Link Partner FID register) transitions from a 1 to a 0. This may happen in the SG1010 for the following reasons:

- The link is attempting to synchronize, and synchronization fails during either the Call, Ack, or Reply state (See Figure 3–11)
- An I Am frame is received where TDIS=1, causing the Link Partner Fabric ID bit to be cleared
- An I Am frame is received with an FID of 7/7/7777777, and the current Link Partner FID register is not 7/7/7777777 (indicates a link partner reset)
- Software clears the Traffic Enable bit
  - When links are bundled, clearing one Traffic Enable bit causes the other link's Traffic Enable bit to also be cleared, causing the entire port to go down
- Software sets the Link Disable bit, turning off the link's transmitter. This causes hardware to clear the Traffic Enable bit due to loss of synchronization

When a link down occurs, a Link Down chip event is signaled. If the link down is not due to failed synchronization, the SG1010 forces the link state to transition to the Call state. The link then attempts to resynchronize. If the resynchronization is successful, the SG1010 signals a Link Up event as described in Section 3.6.1.1. Additionally, SG1010 hardware resets the credit counters to their default values in the corresponding Link State Table. The SG1010 also resets the sequence numbers to 0 for that link. If the links resynchronize, the SG1010 sends an I Am frame to its link partners.

3.6.1.3 Fragile Links

Fragile link state is a physical statement about the number of working differential pairs in a given link. A fully operational link comprises four differential transmit pairs and four differential receive pairs (also called four transmit and receive pairs). A link is fragile when one or more of its differential receive pairs or differential transmit pairs loses or cannot achieve synchronization, but at least one differential transmit pair and one differential receive pair (one transmit and receive pair) is working. The SG1010 supports configurations of 1, 2 or 4 working transmit and receive pairs. If there are three working differential pairs, the SG1010 uses only two. The Differential Pair State register in the Link State Table shows which differential transmit pairs and differential receive pairs are operational and which are not.

When a link goes fragile, a transmission error is detected that causes the link to resynchronize. Upon resynchronization, the SG1010 detects that the number of working differential pairs has been reduced and signals a Fragile Link chip event. However, traffic is still transmitted and received across that link using the remaining synchronized pairs.

When a link is fragile, the receiver for the down differential pairs remains on. The transmitter attempts to synchronize for 1ms every 500ms. If the SG1010 detects that a differential pair has become operational, it forces a resynchronization on the link, to include the new differential pair as a part of the link. No event is signaled if a fragile link becomes less fragile.

If two links are bundled and one of the links goes fragile while the other link is operating at full speed, the SG1010 clears the Traffic Enable bit of the fragile link so that it cannot transmit routed frames. If both links go fragile, the SG1010 takes down the most fragile link, or the lowest numbered link if they are at the same level of fragility.

3.6.1.4 Port Up

A port is up when at least one of its links has its corresponding Traffic Enable bit set. A port is not considered to be up until it can transmit credit-based traffic. Because software must intervene to bring a port from a down state to an up state, an event is not signaled when a port goes up. Software intervention is not required, nor is an event signaled, when a port comes up after chip reset.

3.6.1.5 Port Down

A port is down when the Traffic Enable bits of all of the links that comprise the port are clear. When a port is down, none of its links can transmit traffic. The Port State Table contains a Port Up/Down State bit that reflects the state of the corresponding port.

If a port is down but at least one of its links is synchronized, it can still transmit and receive I Am and Set State special frames even if the Traffic Enable bits are off.[5] However, it cannot transmit other types of frames.

5. The SG1010 can only receive one line every 1024ns in this case.

If a frame encounters a down port, a path event is generated and returned to the originator of the frame.

3.6.2 Sending Frames

A link can transmit frames only when it is in the Linked synchronization state. Additionally, the Traffic Enable bit and the LP_TEN (Link Partner Traffic Enable) bits must both be set in order to send traffic, with the exception of I Am and Set State special frames. These frames must be used to bring a link up to an operational state after the link goes down.

When a frame is sent, the link overhead Frame Sequence Number byte is sent first, followed by the frame itself, then the link overhead Line Credit byte and two CRC bytes. Frame lines are sent in order starting with line0. Within a line, byte0 is sent first. Within a byte, the most significant bit of the 8B/10B encoded byte is sent first and the least significant bit is sent last.

3.6.2.1 Empty Frames

Empty frames (link synchronization special frames) are sent when there are no other frames to send. The empty frame contains the synchronization status of the link as well as the operational state of each differential pair.

3.6.2.2 Frame Sequence Numbers

Frame sequence numbers are used to order frames received on a bundled port. Additionally, they are used to indicate which frames need to be re-sent when an 8B/10B or CRC transmission error occurs.

All credited frames and Line Credit Update special frames use sequence numbers. Other special frames do not use sequence numbers.

A sequence number is a seven-bit field that is prepended to a frame as a part of its link overhead. Bit [7] contains the Thread ID, bits [6:1] contain the Frame Number, and bit [0] contains the Line Debit Type, the latter indicating whether CoS or turn credits were used to send the frame. If a port has only one link, the Thread ID is always 0; if it has two links, the Thread ID can be 0 or 1. A Thread ID of 0 indicates that the frame is dependent on the frame previously sent on the same link. When the Thread ID is 0, the Frame Number is the current value of the frame counter for that link (also called the raw frame number). A Thread ID of 1 indicates that the frame is dependent on a frame sent out the other link in the bundle; in this case, the Frame Number is the raw frame number of the frame it is dependent on.

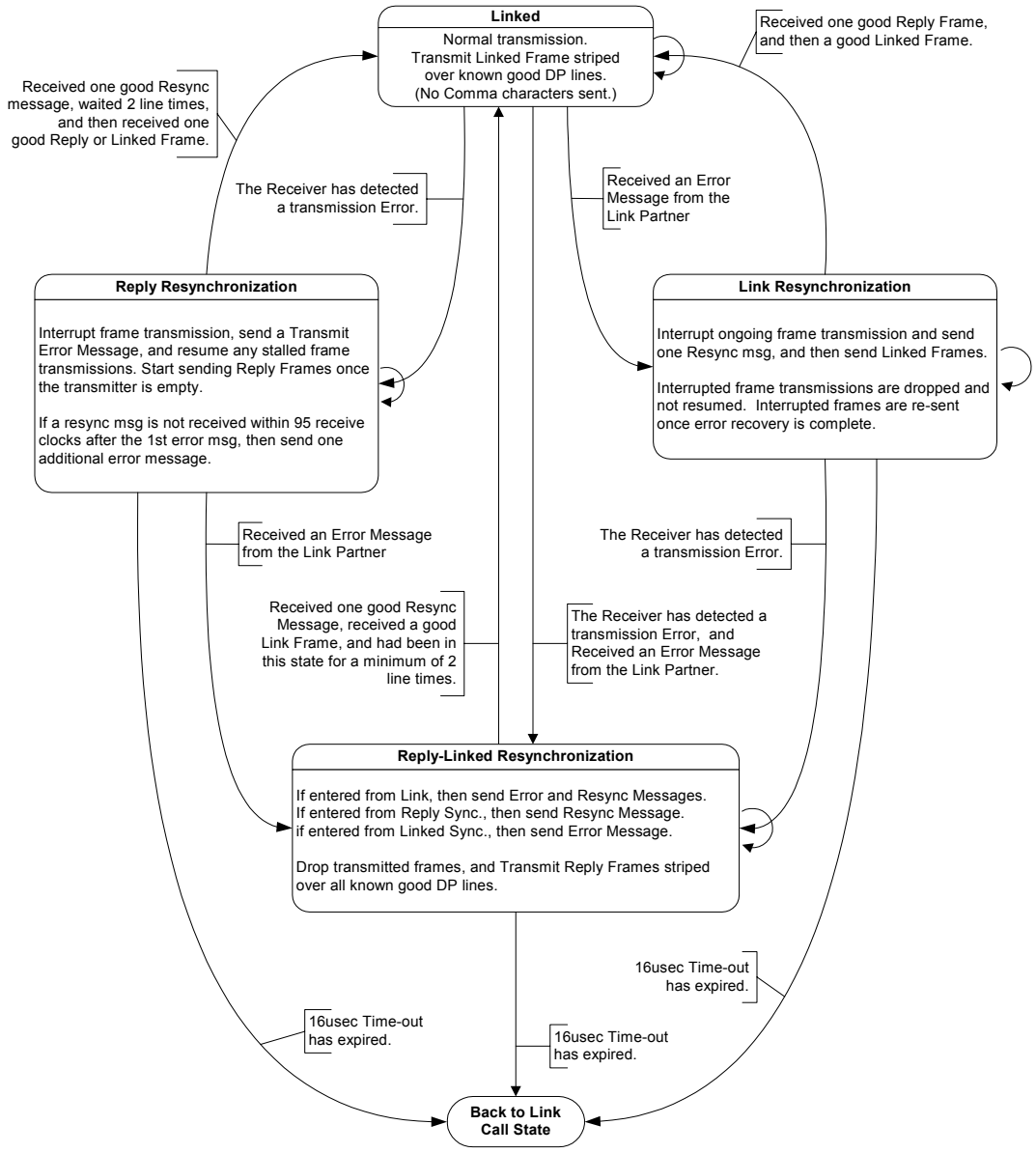
The SG1010 uses these sequence numbers to order frames received on different links in a bundled port. If the port is not bundled, the frames are always received in order. For more information about frame sequence numbers, see the *StarFabric Architecture Specification*.

When a link goes down and comes up, the link partners must reset their sequence numbers in order for the sending and receiving of frame traffic to resume. There are two sets of frame sequence numbers for each link - the transmit sequence numbers which are used in the link overhead of sent frames, and receive sequence numbers, which are the expected sequence numbers of frames that come into the link. Both must be reset. Sequence numbers are reset when both the Traffic Enable bit and the LP_TEN bit are both 1, where one or both were previously 0.

3.6.2.3 Transmission Errors

Data integrity is tracked using two mechanisms – 8B/10B encoding and CRC error checking. 8B/10B encoding is applied to each byte of data sent. CRC checking covers the entire frame, including the link overhead sequence number and line credit bytes. StarFabric protocol uses a 16-bit CRC polynomial. 8B/10B and CRC transmission error recovery are treated in the same way. The transmission error recovery flow is shown in Figure 3–16 and is described in the following sections.

Figure 3–16 Transmission Error Recovery State Flow



DP = Differential Pair

An *error message* is 4 bytes of data sent by the node detecting the transmission error to the node that sent the erroneous frame. The error message contains a K28.5 comma character, followed by a byte containing the sequence number of the frame in error and the error type indication (8B/10B or CRC), followed by a repeat of that byte, followed by another K28.5 comma character. The error message is sent on all working differential pairs; it is not striped. An error message is successfully received when a node detects the error message on at least one working pair. The error message is not covered by CRC, however the receiver of the error message must determine that the two data bytes in the message are identical.

A *resync message* consists of four K28.5 comma characters in a row transmitted across each differential pair. A resync message is sent by a node that detects an error message, and is used to acknowledge the reception of the error message.

3.6.2.3.1 Detecting a Transmission Error

When the SG1010 detects an 8B/10B or CRC transmission error on received data, it marks the current line as end of frame. Data received on that link after the error is detected is dropped. The SG1010 sends an error message to its link partner indicating the type of error that was detected (8B/10B or CRC), whether the error occurred on that link or on a bundled link, and the current received frame count (this count is initialized by the link partner when an I Am frame is received on the link and incremented every time a frame is received). If the port is bundled, the error message is sent out the other link in the port. Otherwise, it is sent out the same link that received the error. The SG1010 may interrupt frame transmission to send the error message, and then resume that frame transmission after the error message is successfully sent.

The SG1010 transitions the link that received the frame with error to the Reply state. If a resync message is not received within 95 62.208MHz clock cycles, then the SG1010 sends another copy of the error message and starts a 16sec timer. While the SG1010 is waiting for a resync message it ignores any other data received on that link. When the resync message is received, the transmitter transitions the link back into the Linked state and resumes normal operation. If the 16sec timer expires and the resync message was not detected, then the SG1010 transitions that link into the Call state for a full resynchronization as described in Section 3.8.3.

3.6.2.3.2 Receiving a Transmission Error

The SG1010 receives a transmission error message from a link partner when one of the frames it send to the link partner contained an 8B/10B or CRC error when it was received. Because error messages are not striped across differential pairs, identical error messages are sent on each working differential pair. The SG1010 must detect a valid error message on one or more differential pairs to start error recovery. A valid error message has two identical copies of the error byte and no 8b/10b errors in any of the four bytes. The receiver drops to the Call state only if it receives invalid error messages on all operational differential pairs.

The error message interrupts the regular stream of frames. The SG1010 does not drop any frames received during or after the reception of an error message.

When the SG1010 receives an error message from its link partner indicating that an 8B/10B or CRC error was detected, it sends a resync message out the link where the error message was received to acknowledge reception of the error message and to transition the link partner into the Linked state. Once the SG1010 received Linked state special frames from its link partner, it may re-send the frames.

The SG1010 determines on which link the error occurred from the error message. It detects the first frame that must be re-sent from the sequence number received in the error message. If the port is bundled, the SG1010 re-sends the frames out the other link

as described in Section 3.6.2.4. Otherwise, if there is only one link in the port, the frames must be re-sent out the same link. All frames from the frame in error to the last frame sent are re-transmitted.

3.6.2.4 Using Change Thread Frames When a Link Goes Down

When a bundled link loses synchronization or there is a transmission error, SG1010 re-sends all the frames affected by the error out the other link. Before the SG1010 re-sends these frames, it first waits until the frames transmitted on the other link have been received without error, and then sends a Change Thread special frame out the working link to tell the receiver that the thread has been temporarily changed to correspond to the link that had the error, and sends the same frame number as the first frame to be re-sent. The SG1010 uses a timer to determine that the Change Thread special frame is successfully sent. After this timer expires and no transmission errors have been detected, the SG1010 then re-sends the frames. After special frame to change the thread back to its original state. After the second Change Thread frame is received, normal traffic (without transmission error) resumes.

3.6.2.5 Kill Frames

The line credit byte appended to a transmitted frame as part of the link overhead has a "Kill Frame" encoding as one of its possible meanings. When this encoding is received, the frame that is attached to that line credit byte is discarded, much as if a transmission error was encountered. When the SG1010 discards a frame due to a Kill Frame encoding, the SG1010 expects the next frame that is transmitted to have the same sequence number as the dropped frame.

No error message is sent to the node that transmitted the frame in this case, and no event is signaled.

For more information, see the *StarFabric Architecture Specification*.

3.7 Software Generated Frames

The SG1010 software generated frame (SGF) function allows a processor to generate a special frame and send it to a link partner. The SGF registers are located in Channel 255 CSR space. The following registers are used:

- SGF Frame – a 12-byte register that contains three-Dword Special frames.
- SGF Control and Status – bits to initiate and check the status of an SGF.

When software is sending an SGF, it must write the frame to the SGF Frame register (described in Section 4.6.5.1). The SG1010 generates any required link overhead (sequence number, CRC, and line credit return information). The remainder of the frame is sent exactly as it is written to the register. Software must ensure that a valid frame format and consistent frame field values are used; the SG1010 does not check or fix bad SGF header formats, nor does it perform a chip operation based on the type of frame that is generated – the frame is simply sent.

Note: *The SG1010 does not support SGFs other than Special frames. If any other type of frame is generated, results are unpredictable.*

After the SGF Frame register is written, the SGF Control and Status register is written to select the output link (0 through 5) and send the frame. The output information is link-based, not port-based. This is necessary for special line credit update frames. Either the Link State table or the Port State table may be used to obtain a link-to-port mapping. When the SEND_SGF bit is written with a 1, the SG1010 sends the information contained in the SGF Frame register to the selected output link.

The SG1010 clears the SEND_SGF bit after the frame is sent or discarded due to an error. If the frame was not successfully sent, the SG1010 sets the SGF Not Sent (SGF_NSNT) state bit. This bit remains set until the SEND_SGF bit is written to send the next SGF. After the frame is sent, the SG1010 sets the SGF_DONE state bit in the SGF Control and Status register and also sends an SGF Done event to the event logic. The SGF_DONE state bit is cleared when software clears the SGF Done event bit in the Raw Event Status register or when the next SGF is sent by writing a 1 to the SEND_SGF control bit.

3.8 Semaphore Registers

The SG1010 implements two eight-bit semaphores that software can use to reserve resources. SG1010's hardware does not associate a semaphore with a specific resource; this is done by software. All software using the semaphores must have a common understanding of the semaphore's purpose.

Each semaphore consists of eight Dword-aligned registers, one register per semaphore operation. A semaphore is altered atomically whenever a read operation is performed to one of its registers. When a semaphore register is read, the current value of the semaphore is returned, and then the operation associated with that register is performed.

The following semaphore operations are supported:

- Clear – the semaphore is cleared to 00h
- Set – the semaphore is set to 01h
- Decrement – the semaphore is decremented by 1
- Increment – the semaphore is incremented by 1
- Increment if 0 – the semaphore is incremented if the current value is 0
- Increment if not 0 - the semaphore is incremented if the current value is non-zero
- Two reserved operations – no semaphore operation is performed

The current semaphore value before the operation is returned in the read. Writes to the semaphore registers are discarded and have no effect. The semaphores are sticky for decrements at 0, and for increments at FFh; that is, they do not wrap.

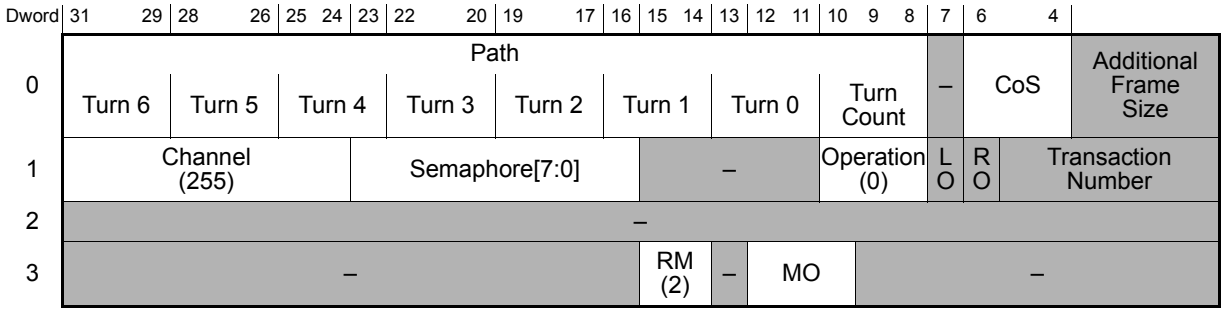
If a semaphore Dword location is read and all of the byte enables for that Dword are off, the semaphore operation is not performed.

3.8.1 Semaphore Frames

The SG1010 performs a semaphore operation when a read request frame accesses a semaphore register location. However, the SG1010 also supports semaphore register access through a Read-Modify-Write frame, which is a path-routed read request frame to Channel 255 (the header fields are shown in Figure 3–17). When the SG1010 receives one of these frames, it extracts the eight-bit semaphore number and the three-bit semaphore operation from the header Modify Operation field. The SG1010 uses the semaphore number and the semaphore operation number to construct the semaphore Channel 255 offset. The SG1010 then reads the appropriate semaphore register, returns the current value in a Read Completion frame, and performs the semaphore operation.

If an unsupported semaphore number is used (any number except 0 or 1) then a Read Completion frame with a Range failure type is returned. If a Channel Number other than 255 is specified, a Channel Inactive failure type is returned.

Figure 3–17 Read–Modify–Write Frame Header



RM = Request Mode
MO = Modify Operation

3.9 General Purpose I/O Interface

The SG1010 general purpose I/O (GPIO) interface allows software to control a signal pin interface. The SG1010 implements eight dedicated GPIO signal pins, GPIO[7:0]. The GPIO pins are controlled through a GPIO register interface (described in Section 4.7.5) in SG1010 configuration registers. The register interface controls both the direction of each GPIO pin (input only or bidirectional), as well as the value on the GPIO signal pin.

The Set GPIO Direction and Clear GPIO Direction registers are used to configure a GPIO pin to be an input only or a bidirectional pin. Each bit in the GPIO direction registers corresponds to a GPIO signal. When a GPIO direction bit is a 1, the corresponding GPIO pin is configured to be bidirectional and the SG1010 drives a value onto the signal. When a GPIO direction bit is 0, the corresponding GPIO pin is configured to be input-only and the SG1010 can only sample the value of the GPIO signal. Writing a 1 to a Set GPIO Direction register bit sets the bit to a 1, assigning the signal as bidirectional. Writing a 1 to a GPIO Clear Direction bit clears the bit to a 0, assigning the signal as an input. If a 1 is written to the same bit in both registers with the same write operation, the set operation is dominant over the clear operation. Writing a 0 to either register has no effect. Reading either register returns the current value of the GPIO direction register.

The Set GPIO Data and Clear GPIO Data registers are used to assign a value to a GPIO pin that is configured to be bidirectional, and also to return the current value on the GPIO pins. Each bit in the GPIO data registers corresponds to a GPIO signal. When the Set GPIO Data register is read, it returns the current value on the GPIO signal pins, regardless of the direction of the pin. When the Clear GPIO Data register is read, it returns the value written to the GPIO data register. Note that these values are not necessarily the same if the GPIO pin is configured as an input.

When a GPIO data bit is a 1 and the corresponding GPIO direction bit is a 1, the GPIO pin is driven high. When a GPIO data bit is 0 and the corresponding GPIO direction bit is a 1, the GPIO pin is driven low.

Writing a 1 to a Set GPIO Data register bit sets the corresponding GPIO data bit. Writing a 1 to a Clear GPIO Data register bit clears the corresponding GPIO data bit. When a 1 is written to the same bit in both registers with the same write operation, the set operation dominates the clear operation. Writing a 0 to either register has no effect.

3.10 Serial ROM Interface

3.10.1 ROM Programming Interface

The serial ROM (SROM) is programmed through SG1010's ROM register interface (described in Section 4.7.4), located in the configuration registers. SG1010's ROM register interface consists of:

- A 32-bit ROM Data register
- A 32-bit ROM Address register, consisting of
  - 16 bits of ROM Address
  - Start/Busy Flag bit

3.10.1.1 Writing through the ROM Register Interface

The ROM register interface supports an aligned four-byte write to the SROM. To perform a write, four bytes of data are first written to the ROM Data register. A second write to the ROM Address register writes the Dword-aligned ROM address and sets the Start/Busy flag to a 1 to select and start the write operation.

When the write operation is complete and the ROM interface is ready for another operation, the SG1010 clears the Start/Busy flag to a 0.

3.10.1.2 Reading through the ROM Register Interface

The ROM register interface supports an aligned four-byte read from the SROM. To initiate the read, a write to the ROM Address register writes the Dword-aligned ROM address, and clears the Start/Busy flag to a 0 to select and start the read operation. When the read operation is complete and the ROM interface is ready for another operation, the SG1010 sets the Start/Busy flag to a 1. The read data can then be read from the ROM Data register.

3.10.2 Vital Product Data (VPD)

The SG1010 supports the VPD enhanced capability function. This function allows a processor to access vital product data through a standard configuration register interface. The VPD configuration registers control the serial ROM interface. VPD read and write accesses are always four bytes, Dword-aligned.

The VPD registers are:

- The 32-bit VPD Data register (described in Section 4.7.9.4)
- The 16-bit VPD Address register (described in Section 4.7.9.3)
  - A seven-bit VPD address field
  - Operation/Status flag (VPDFLAG)

The VPD interface accesses only that portion of the SROM designated to be VPD space. VPD space is defined to be a 256-byte space starting at SROM offset 0. VPD space is partitioned further into a read-only VPD portion and a read/write portion. The read only portion consumes the first 128 bytes of VPD space. The read/write portion consumes the remainder of VPD space. VPD read-only space cannot be written from the VPD register interface.

To write VPD read-only space, or to access an SROM location outside of VPD space, the SG1010 ROM programming interface must be used. SG1010's ROM programming interface can read or write any SROM location.

The VPD read and write operations are similar to the SROM programming operations described in Section 3.10.1. To perform a VPD read, the VPD Address register is written with the Dword-aligned VPD address offset and a VPDFLAG value of 0. Writing the VPDFLAG bit with a 0 causes the SG1010 to initiate the VPD read. SG1010 appends the VPD offset onto the VPD base address, which is 0. When the SROM returns the read data, the SG1010 places it in the VPD Data register and sets the VPDFLAG to 1.

To perform a VPD write, the VPD write data is first written to the VPD Data register. The VPD Address register is then written with the Dword-aligned VPD address offset and a VPDFLAG value of 1. Writing the VPDFLAG bit with a 1 causes the SG1010 to initiate the VPD write. When the SROM write is completed, the SG1010 clears the VPDFLAG to 0. If a write is attempted to the read-only VPD space, the SG1010 does not perform a SROM operation but immediately clears the VPDFLAG to 0.

3.10.3 Serial ROM Interface

The SROM interface supports serial peripheral interface (SPI) compatible serial ROMs. The serial data format uses an eight-bit OPCODE and a 16-bit address. Example compatible SROMs with this format include the Microchip 25AA160/25LC160/25C160 family.

The SROM pin interface consists of the following signals:

- SR_DI – serial data ROM input
- SR_DO – serial data ROM output
- SR_CS_L – serial ROM chip select
- SR_CK – serial ROM clock

The register interface specifies only whether a write or a read operation is to be performed. The SG1010 always performs a write enable before every SROM write. The SROM performs its own write disable operation after each write.

3.10.3.1 Serial Address and Data Organization

The ordering of bits sent to and received from the SROM is from most significant bit to least significant bit for both address and data. For the 16-bit address, the specific *bit* sequence is AD15, AD14, AD13 … AD0. For each byte of data, the specific sequence is D7, D6, D5 … D0.

When multiple bytes are read from the SROM, the order of bytes is least significant to most significant. For a four-byte operation, the specific byte sequence is Byte 0, Byte 1, Byte 2 and Byte 3. The specific 32-bit sequence for such an operation is as follows:

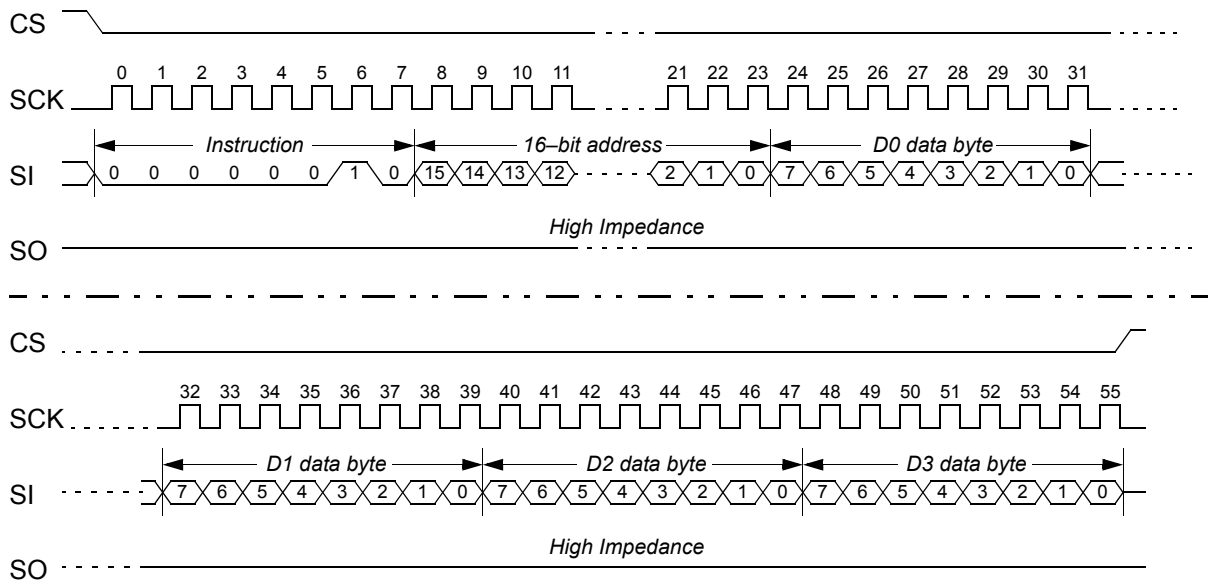
Byte0-D7, Byte0-D6, Byte0-D5,… Byte0-D0, Byte1-D7,… Byte3-D0.

The SG1010 supports four-byte writes and four-byte reads.

3.10.3.2 Serial ROM Write Operation

Figure 3–18 shows the four-byte serial ROM sequence.

Figure 3–18 Serial ROM Write Timing Diagram



The SG1010 executes a write enable (WREN) instruction prior to each SROM write operation to enable the SROM Write latch. The SROM sequencer deasserts the SR_CS_L signal when the WREN instruction completes. The WREN operation is described in Section 3.10.3.4.

The write operation comprises the following steps

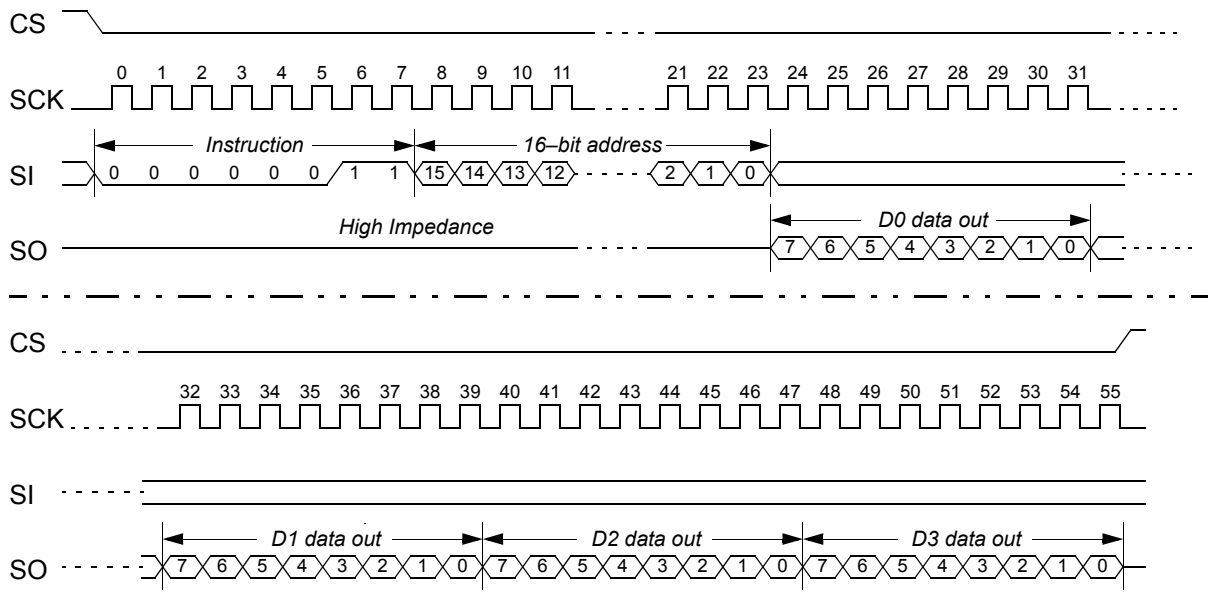
1. The SG1010 asserts SR_CS_L and shifts the instruction, address and data sequence to the serial ROM as shown in Figure 3–18. This causes the serial ROM to write:
   a. Write the 16-bit address into the two-byte address field in its CSR.[6]
   b. Write data into the D0, D1, D2 and D3 bytes in its CSR.
   c. Write OPCODE 0000.0010h into the OPCODE byte in its CSR.
   d. Write the Busy bit in the its CSR status register to a 1. This event triggers the write operation internally to the SROM.
2. The SG1010 completes the write operation and deasserts SR_CS_L. The write operation consumes 55 clock cycles.
3. The SG1010 then enters a polling routine using the RDSR (Read Status register) operation described in Section 3.10.3.5.

3.10.3.3 Serial ROM Read Operation

Figure 3–19 shows the Serial ROM read timing diagram.

6. A 16-bit address assumes a 64K SROM (8K × 8 bit). For a 16K SROM (2K × 8 bit), only the lower 12 address bits are used; the upper four bits should be written to 0.

Figure 3–19 Serial ROM Read Timing Diagram



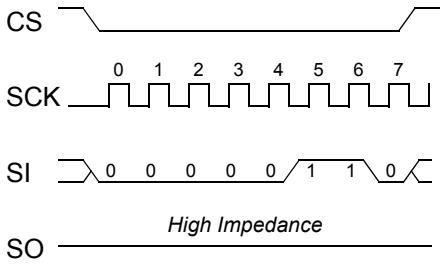
The read operation consists of the following steps:

1. The SG1010 asserts SR_CS_L and shifts the instruction and address sequence to the serial ROM. This causes the serial ROM to
   a. Write the 16-bit address into the two-byte address field in its CSR.
   b. Write OPCODE 0000.0011 into the OPCODE byte in its CSR.
   c. Writes the Busy bit in the SROM CSR status register to a 1. This event triggers the read operation internally to the SROM.
2. The SG1010 completes the read operation. The read operation consumes 55 clock cycles.
3. The SG1010 receives the serial data and places it in the ROM Data Register.
4. The SG1010 sets the Start/Busy flag to a 1 in the ROM Address register to indicate that the read operation is complete.

3.10.3.4 Serial ROM Write Enable Operation

Figure 3–20 shows the Serial ROM write enable timing diagram.

Figure 3–20 Serial ROM Write Enable Timing Diagram



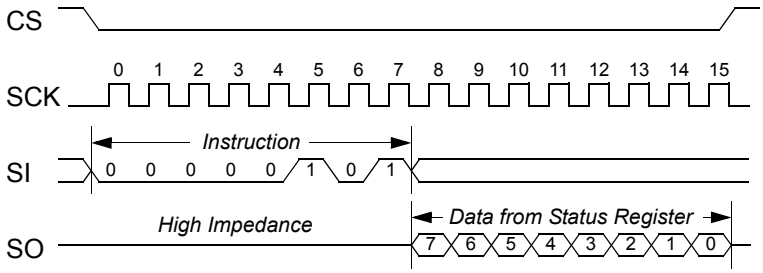
This SROM transaction enables the Write Enable latch internal to the SROM. This latch must be enabled in order for the SG1010 to perform a serial ROM write. The SG1010 automatically performs the write enable operation before any serial ROM write. The following sequence is used for the write enable operation:

1. The SG1010 asserts SR_CS_L and shifts the instruction sequence to the serial ROM. This causes the serial ROM to write OPCODE 0000.0110 into the OPCODE byte in its CSR.
2. The SG1010 deasserts SR_CS_L. The write enable operation consumes eight clock cycles.
3. When the SROM sequencer completes the write enable sequence, it then proceeds with the serial ROM write operation.

3.10.3.5 Serial ROM Read Status Register Operation

Figure 3–21 shows the Serial ROM read status timing diagram.

Figure 3–21 Serial ROM Read Status Timing Diagram



This SROM transaction reads the SROM status register. The SG1010 automatically performs the read status operation immediately after any serial ROM write to determine when the write is complete internally to the SROM. The SG1010 performs the read status operation every 2ms until the status indicates that the write is complete. The following sequence is used for the read status operation:

1. The SG1010 asserts SR_CS_L and shifts the instruction sequence to the serial ROM. This causes the serial ROM to write OPCODE 0000.0101 into the OPCODE byte in its CSR.

2. The serial ROM shifts out the contents of its status register.
3. The SG1010 deasserts SR_CS_L. The read status operation consumes 16 clock cycles.
4. The SG1010 checks bit [0] (BUSY) of the status register to determine whether the write has completed. If the write has completed, the SG1010 clears the Start/Busy bit in the ROM Address register to indicate that the serial ROM write operation is complete. If the write has not completed, the SG1010 repeats this operation 2ms later.

3.11 Diagnostic Interfaces

3.11.1 JTAG

The SG1010 is fully compliant with the IEEE 1149.1a-1993 Boundary Scan Specification (known informally as "JTAG"). The SG1010 includes the following pins for IEEE 1149.1 support: TRST_L (Test Reset), TDI (Test Data In), TMS (Test Mode Select), TCK (Test Clock) and TDO (Test Data Out). The first four pins are input-only, while TDO is output-only.

All of the input pins are pulled high internally. TDO is an output-only and drives low when TRST is high.

The SG1010's implementation of IEEE 1149.1 includes the required instructions BYPASS, EXTEST, and SAMPLE/PRELOAD, and the optional instructions IDCODE and RUNBIST.

For more information, see the IEEE Standard Test Access Port and Boundary Scan Architecture 1149.1-1990, IEEE Std. 1149.1a-1993, and IEEE Std. 1149.1b-1994.

3.11.2 Register Access Diagnostic Port

The SG1010 diagnostic port is used to read and write CSRs and other internal state during operation. It is intended primarily for debug, but it can also be used for other purposes, such as gathering statistics without consuming fabric bandwidth. The diagnostic port uses a simple, asynchronous, register indirect protocol. The SG1010 is always the target of the transactions. A device such as a microcontroller or a PC parallel port can be used to interface to the diagnostic port.

The diagnostic port presents 16 registers to the user, each one byte wide. Four registers (A3–A0) are used to accumulate a 32-bit address, and four registers (D3–D0) are used to hold a Dword of data for the write or read. Registers A3–A0 reference a Channel 255 register offset within the SG1010. Two registers, TW and TR, are used to trigger the read or write to Channel 255 register space.

Note: *The other diagnostic port registers are used internally; accessing them will cause unpredictable results.*

Table 3–10 defines the registers and their addresses.

Table 3–10 Diagnostic Interface Register Addresses

Register Name	Address	Type	Register Name	Address	Type
A0	0	R/W	TW	8	W
A1	1	R/W	TR	9	W
A2	2	R/W	Reserved	A	–
A3	3	R/W	Reserved	B	–
D0	4	R/W	Reserved	C	–
D1	5	R/W	Reserved	D	–
D2	6	R/W	Reserved	E	–
D3	7	R/W	Reserved	F	–

Table 3–11 describes the 12 signal pins on the diagnostic port interface.

Table 3–11 Diagnostic Port Signal Pins

Signal Name	Descriptive Name	Description
AD[7:0]	Address, Data, and Byte Enables	Used by the master to drive four bits of address on AD[3:0] for a read or write transaction, and eight bits of data for a write transaction. SG1010 drives eight bits of read data on these signals. The master drives four bits of byte enables on AD[7:4] during the trigger transaction at the same time as the address.
AS_L	Address Strobe	Asserted by the master when the address is valid on AD[3:0] and the byte enables are valid on AD[7:4]
RD_L	Read Strobe	Asserted by the master to begin a read to a diagnostic register
WR_L	Write Strobe	Asserted by the master when write data is valid on AD[7:0]
RDY_L	Ready	Asserted by the SG1010 when it is ready to complete the current read or write access. The SG1010 is normally not ready; after a read, write, or trigger operation, the master must check RDY_L and wait for it to be asserted before continuing the transaction. This allows the SG1010 to insert unlimited wait states if the resource is busy.

The diagnostic interface pins are multiplexed with the LED_L pins. To enable the diagnostic interface, the DIAG_EN input must be driven high. To determine which pins are used for the diagnostic interface, see Table 5–7.

The diagnostic port supports three types of transactions: Read, Write, and Trigger. A trigger transaction is a write to the TW or TR registers.

Figure 3–22 shows the timing diagram for a diagnostic port read transaction. The AC timing for these signals is found in the SG1010 Datasheet.

Figure 3–22 Diagnostic Port Read Transaction

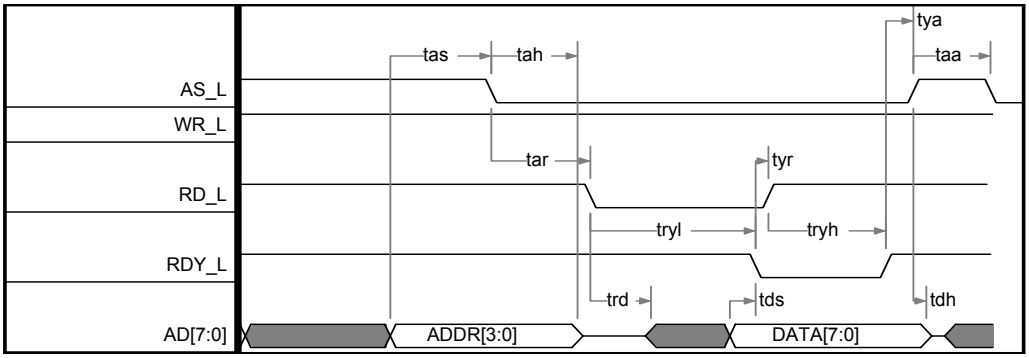
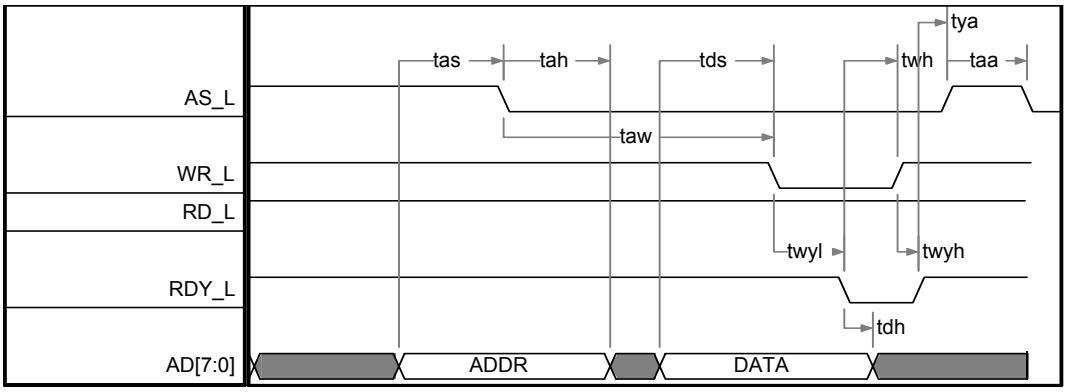


Figure 3–23 shows the timing diagram for a diagnostic port write transaction.

Figure 3–23 Diagnostic Port Write Transaction



3.11.2.1 Trigger Transaction

A trigger transaction is similar to a write transaction to registers TW or TR. However, the trigger transaction address phase also carries byte enable information for the CSR access.

The master must drive byte enables on AD[7:4] at the same time as the address is driven on AD[3:0]. Byte enables are active high and determine which bytes of the CSR are accessed. AD[4] corresponds to byte 0, AD[3] corresponds to byte 1, etc.

A write to TW causes the data in D3–D0 to be written to the CSR located at the Channel 255 offset specified by registers A3–A0. The SG1010 deasserts RDY_L when the CSR write is complete.

A write to TR causes a CSR read to occur. The SG1010 reads the register at the Channel 255 offset specified by registers A3–A0. The read data is placed in registers D3–D0. The SG1010 deasserts RDY_L when the CSR read is complete.

It takes longer to deassert RDY_L for a trigger transaction than it does for a normal read or write transaction. Trigger transactions initiated before the end of serial ROM preload will not complete until serial ROM preload is complete.

3.11.3 LED Signals

The SG1010 implements 24 signals that can be used for LED control based on link state or software control. There are six sets of four signals, one set per link: LED0_L[3:0] through LED5_L[3:0]. The four signals per link are allocated to each of the four differential pair receivers per link.

Ten of the signals (LED2_L[1:0], LED1_L[3:0], LED0_L[3:0]) are shared with the TESTMUX[9:0] signals. Twelve of the signals (LED5_L[3:0], LED4_L[3:0], LED3_L[3:0]) are shared with the Register Access Diagnostic Port. Use of the TESTMUX[9:0] signals or Register Access Diagnostic Port automatically overrides use of these respective signals for LED control.

The LED*x*_L signals are controlled using the LED Control register (described in Section 4.6.8.2), located in Channel 255 register space. Six LED Mode bits, one for each link, select whether the four LED signals corresponding to that link are controlled by software, or controlled by link state. If the LED signals are controlled by software, then the LED State bits control the value that is driven on those LED signals.

There are two modes of behavior for LED signals driven by link state. The mode is selected by sampling the GPIO[0] signal either high or low at the deasserting edge of NRST_L. If GPIO[0] is sampled low, the LED signals reflect the receiver state of each differential pair. If GPIO[0] is sampled high, the least significant LED signal in each group of four reflects the state of the link. The remaining LEDs are driven high. Table 3–12 shows the hardware controlled LED signal state.

Table 3–12 Hardware-Controlled LED Signal State

LED Signal Pin State	LED Control S/W Mode	Differential Receiver Mode (GPIO[0] sampled 0)	Link Mode – LED*x*[0] only (GPIO[0] sampled 1)
High (LED off)	0	Differential receiver not synchronized or link not in Linked state	Link Down (LED*x*[0] only)
Low (LED on)	1	Differential receiver synchronized, link in Linked state, and Traffic Enable set	Current link state is Linked and Traffic Enable set
≈ 500ms cycle (LED blinking)	–	Differential receiver synchronized, link in Linked state, and Traffic Enable not set	Current link state is Linked and Traffic Enable not set

Diagnostic Interfaces

4 Registers

The SG1010 implements the following classes of registers:

- StarFabric Component (SFC) Header registers: Fabric identity, control, and extended function list pointer (ELP) registers.
- CSRs: Control and status registers for fabric and device-specific functions.
- PCI-to-PCI Bridge PCI configuration registers: PCI control, status, and address decoding registers for address routing and PCI compatibility.

The SG1010 permits multiple Dword write and read request frames to its registers. However, unpredictable results can occur if multiple Dword reads are used to access the Configuration Index registers or the Semaphore registers since these registers have read side effects.

The SG1010 supports byte writes through the byte mask mechanism in write frames. The SG1010 also supports byte reads, although all bytes are returned for reads that have no side effects. For semaphore registers, which have read side effects, the semaphore operation is not performed if all bytes in the dword request are masked.

If a register is reserved, it cannot be written and returns zero when read.

Register access can occur through a Type0 configuration read or write frame, a Channel 255 read or write, or a serial ROM write. Table 4–6 describes the mechanisms for accessing the SG1010's registers.

Table 4–1 SG1010 Register Access Mechanisms

	Type0 Configuration	Channel 255	SROM Preload*	Diagnostic Port
Bridge Configuration	Root port only	Mapped at 0A00h	Mapped at 0A00h	Mapped at 0A00h
SFC Header	Index mechanism	Mapped at 0000h	Mapped at 0000h	Mapped at 0000h
CSRs	Index mechanism	Mapped at 0200h	Mapped at 0200h	Mapped at 0200h

* Access restricted to writes of writable and selected read-only registers using Channel 255 offsets. No read access.

Indexed registers are used in configuration space to allow access to all CSRs using Channel 255 offsets. In this case, offset and data registers are used in configuration space to get to the CSR locations.

4.1 Generating Register Access Response Frames

When the SG1010 receives a frame that indicates a register read operation or a register write with acknowledge operation, it generates one or more response frames to the origin of the read or write frame.

Response frames may be generated in response to the following types of operations:

- Address-routed configuration (Cfg) read or write with acknowledge
- Address-routed I/O read or write with acknowledge
- Path-routed Channel 255 read or write with acknowledge

The SG1010 may generate multiple read completion frames if a prescriptive Channel 255 read is requested. Otherwise, the read completion frame is restricted to a single-line frame with one Dword of read data, assuming no failures.

The Failure Type header field contains error information, if any. Register protection errors are described in Section 4.2. If no errors are encountered during the delivery of the write, the SG1010 sets the Failure Type to Fh (normal termination). Otherwise, the Failure Type field indicates the type of error encountered. Table 4–2 describes the possible Failure Types for a write transaction.

Table 4–2 Failure Types for Write Acknowledge

Failure Type	Description of Failure	Failure Type Value
Range	Channel 255 address compare failure	0h
Channel Lock	Channel 255 path protection failure	1h

4.2 Register Protection

The SG1010 has register access protection mechanisms to enable register reads and writes from only certain remote destinations. The SG1010 implements four path registers that can be enabled for register path protection. These path registers contain transformed paths, that is, the path from the SG1010 to the permissible origins of the frame, and an input port. If register path protection is enabled, any register access, whether it uses Channel 255 path routing or address routing, is subject to path protection checks. The path of the incoming frame is transformed and compared against all four path registers and their input ports. If any of the paths match, then the register access is performed if the range check is successful. If there are no matches among the enabled registers, then the SG1010 does not perform the register access, and signals a Channel Path Protection chip event. If a response frame is required, the Channel Lock failure type is used.

Each path protection register has an associated enable bit that enables the path protection compare against that registers. If none of the enable bits are set, then path protection is not performed for register accesses.

Access of registers through a multicast write is not permitted. Multicast writes that specify Channel 255 will result in a Software Routing Failure failure type if a response frame is required, and a Software Multicast Distribution Failure event.

If a Channel 255 read request or write frame specifies an offset at or above the 4KB boundary (0000 1000h or higher), a range error occurs. If a response frame is required, it contains the Range failure type. The SG1010 also signals a Channel Range event.

4.3 SROM Preload

After chip reset is completed, the SG1010 performs an optional serial ROM preload for register initialization.

Register preload data must start at byte 256 (100h) of the serial ROM. The SG1010 preloads the first byte of the preload data to detect the preload sequence enable 10000b in bits [7:3] of the first byte read. If the preload enable sequence is detected, the SG1010 continues the preload operation. Otherwise, the serial ROM read is terminated and a preload is not performed.

The serial preload data provides a list of preload operations. Each preload operation consists of a Channel 255 offset, followed by a size field and a number of data bytes, where the number of data bytes is the same as the size field. The SG1010 preloads CSR registers starting at the Channel 255 byte offset provided and continuing until all the data bytes for that region are written. Subsequent preload operations in the list are then performed based on the Channel 255 offsets, sizes, and data byte provided. The SG1010 terminates the serial preload operation when a size field of 0 is detected.

Table 4–3 shows the format for serial ROM preload data.

Table 4–3 SROM Preload Data Format

Byte	Bits	Contents
0	7:3	Preload sequence 10000b
	2:0	Reserved. Must be 0.
1	7:0	Channel 255 offset [7:0]
2	3:0	Channel 255 offset [11:8]
	7:4	Reserved. Must be 0.
3	7:0	Data field size N in bytes
Next N bytes	–	Data0 through DataN-1 Repeat starting with Channel 255 offset [7:0] until data field size = 0 detected

Any register bit that has write access can be written through the SROM preload. Additionally, selected read-only registers can also be overloaded through the preload.

Table 4–4 shows the read-only registers that can be overloaded.

Table 4–4 Read–only Registers with Preload Allowed

Register Space	Register Name	Bits Name	Dword Offset	Bits Offset
Configuration	Slot Numbering Expansion Slot	–	7Ah	5:0
SFC Header	OEM Device Driver	–	8h	–
	StarFabric Protocol Revision	–	10h	–
	Programming Interface	–	1Ch	–

4.4 Register Maps

Table 4–5 maps the addresses for the StarFabric Component (SFC) Header registers.

Table 4–5 SFC Header Register Map

Byte 3	Byte 2	Byte 1	Byte 0	CH255 Byte Offset
SFC Vendor ID				0000h
SFC Device ID				0004h
SFC OEM Vendor Driver ID				0008h
Silicon Revision ID				000Ch
StarFabric Protocol Revision				0010h
SFC Base Class ID				0014h
Reserved				0018h
SFC Programming Interface				001Ch
Fabric ID				0020h
SFC Capabilities Register				0024h
Extended Function List Pointer (ELP)				0028h
SFC Control				002Ch
SFC Fabric Reset				0030h
Reserved				0034h – 003Fh
Semaphore ELP ID				0040h
Semaphore Next ELP				0044h
Semaphore Revision ID				0048h
Semaphore Offset Pointer				004Ch
Semaphore Number of Entries				0050h
Reserved				0054h – 005Fh
SGF ELP ID				0060h
SGF Next ELP				0064h
SGF Revision ID				0068h
SGF Offset Pointer				006Ch
Reserved				0070h – 007Fh
Port State Table ELP ID				0080h

Table 4–5 SFC Header Register Map *(Continued)*

Byte 3	Byte 2	Byte 1	Byte 0	CH255 Byte Offset
Port State Table Next ELP				0084h
Port State Table Revision ID				0088h
Port State Table Offset Pointer				008Ch
Port State Table Number of Entries				0090h
Port State Table Entry Size:				0094h
Reserved				0098h – 009Fh
Link State Table ELP ID				00A0h
Link State Table Next ELP				00A4h
Link State Table Revision ID				00A8h
Link State Table Offset Pointer				00ACh
Link State Table Number of Entries				00B0h
Link State Table Entry Size				00B4h
Reserved				00B8h – 00BFh
Event Table ELP ID				00C0h
Event Table Next ELP				00C4h
Event Table Revision ID				00C8h
Event Table Offset Pointer				00CCh
Reserved				00D0h – 00DFh
Port Map Table ELP ID				00E0h
Port Map Table Next ELP				00E4h
Port Map Table Revision ID				00E8h
Port Map Table Offset Pointer				00ECh
Port Map Table Number of Entries				00F0h
Port Map Table Entry Size:				00F4h
Reserved				00F8h – 00FFh
Multicast ELP ID				0100h
Multicast Next ELP				0104h
Multicast Revision ID				0108h
Multicast Offset Pointer				010Ch
Multicast Number of Entries				0110h
Multicast Entry Size:				0114h
Reserved				0118h – 011Fh
Channel 255 Path Protection ELP ID				0120h
Channel 255 Path Protection Next ELP				0124h
Channel 255 Path Protection Revision ID				0128h
Channel 255 Path Protection Offset Pointer				012Ch
Channel 255 Path Protection Number of Entries				0130h
Reserved				134h – 013Fh

Table 4–5 SFC Header Register Map *(Continued)*

Byte 3	Byte 2	Byte 1	Byte 0	CH255 Byte Offset
Channel 255 Register ELP ID				0140h
Channel 255 Registers Next ELP				0144h
Channel 255 Registers Revision ID				0148h
Channel 255 Registers Offset Pointer				014Ch
Channel 255 Registers Number of Entries				0150h
Channel 255 Registers Entry Size				0154h
Reserved				0158h – 015Fh
Scratchpad ELP ID				0160h
Scratchpad Next ELP				0164h
Scratchpad Revision ID				0168h
Scratchpad Offset Pointer				016Ch
Scratchpad Number of Entries				0170h
Scratchpad Entry Size				0174h
Reserved				0178h – 017Fh
Vital Product Data (VPD) ELP ID				0180h
VPD Next ELP				0184h
VPD Revision ID				0188h
VPD Offset Pointer				018Ch
Reserved				0190h - 01FFh

Table 4–6 maps the addresses for the Functional Control and Status registers (CSRs).

Table 4–6 CSR Register Map

Dword	Byte 3	Byte 2	Byte 1	Byte 0	CH255 Byte Offset
Link 0 State Table					
0	Link control and status				0200h
1	Link Partner Fabric ID				0204h
2	8B/10B Error Count		CRC Error Count		0208h
3	Frame Count				020Ch
4	Line Count				0210h
5	Empty Frame Count				0214h
6	Multicast Write Credit	HP-Asynchronous Write Credit	Isochronous Write Credit	Asynchronous Write Credit	0218h
7	Reserved	Provisioning Write Credit	HP-Isochronous Write Credit	Address Routed Write Credit	021Ch
8	T3 Write Credit	T2 Write Credit	T1 Write Credit	T0 Write Credit	0220h
9	T7 Write Credit	T6 Write Credit	T5 Write Credit	T4 Write Credit	0224h
10	Provisioning Request Credit	HP-Isoc/Addr Request Credit	HP-Asynchronous Request Credit	Isoc/Async Request Credit	0228h
11	T7/6 Request Credit	T5/4 Request Credit	T3/2 Request Credit	T01/0 Request Credit	022Ch
12	Reserved	Diff. Pair State	Bandwidth Count		0230h
12	Default CoS Credit Bytes 0-3				0234h
13	Default CoS Credit Bytes 4-7				0238h
12	Default Turn Request Credit	Default Turn Write Credit	Default CoS Credit Bytes 8-9		023Ch
Link 1 State Table	(Same format as Link 0 State Table)				0240h – 027Fh
Link 2 State Table	(Same format as Link 0 State Table)				0280h – 02BFh
Link 3 State Table	(Same format as Link 0 State Table)				02C0h – 02FFh
Link 4 State Table	(Same format as Link 0 State Table)				0300h – 033Fh
Link 5 State Table	(Same format as Link 0 State Table)				0340h – 037Fh
Port 0 State Table					
0	Port to Link Map				0380h
1	Port Status				0384h
Port 1 State Table	(Same format as Port 0 State Table)				0388h – 038Ch
Port 2 State Table	(Same format as Port 0 State Table)				0390h – 0394h
Port 3 State Table	(Same format as Port 0 State Table)				0398h – 039Ch
Port 4 State Table	(Same format as Port 0 State Table)				03A0h – 03A4h
Port 5 State Table	(Same format as Port 0 State Table)				03A8h – 03ACh
	Reserved				03B0h – 03FFh

Table 4–6 CSR Register Map *(Continued)*

Dword	Byte 3	Byte 2	Byte 1	Byte 0	CH255 Byte Offset
Chip Event Table					
0 … 31	Entry 1 … Entry 31		Entry 0 … Entry30		0400h … 043Fh
Signal Event Table	(Reserved)				
0 … 6	Entry 1 … Entry 33		Entry 0 … Entry12		0440h … 045Bh
	Reserved				045Ch – 047Fh
Event Path Table					
0	Entry0				0480h
1	Entry 1				0484h
2	Entry 2				0488h
3	Entry 3				048Ch
	Reserved				0490h – 049Fh
Event Mask					
0	Write-1-to-Clear				04A0h
1	Write-1-to-Clear				04A4h
2	Write-1-to-Clear				04A8h
	Reserved				04ACh – 04BFh
0	Write-1-to-Set				04C0h
1	Write-1-to-Set				04C4h
2	Write-1-to-Set				04C8h
	Reserved				04CCh – 04DFh
Raw Event Status					
0	Raw Event Status				04E0h
1	Raw Event Status				04E4h
2	Raw Event Status				04E8h
	Reserved				04ECh – 04FFh
Event Status					
0	Event Status				0500h
1	Event Status				0504h
2	Event Status				0508h
	Reserved				050Ch – 051Fh
	Event Dispatch Control				0520h
	Reserved				0524h – 07FFh

Table 4–6 CSR Register Map *(Continued)*

Dword	Byte 3	Byte 2	Byte 1	Byte 0	CH255 Byte Offset
Port Map Table Entry 0					
0	Bridge Control		Command		0800h
1	Reserved	Subordinate Bus #	Secondary Bus #	Primary Bus #	0804h
2	Reserved		I/O Limit	I/O Base	0808h
3	Memory Limit		Memory base		080Ch
4	PF Memory Limit		PF Memory Base		0810h
5	PF Memory Base Upper 32 Bits				0814h
6	PF Memory Limit Upper 32 Bits				0818h
7	I/O Limit Upper 16 Bits		I/O Base Upper 16 Bits		081Ch
Port Map Table Entry 1	(Same format as Port Map Table Entry 0)				0820h – 083Fh
Port Map Table Entry 2	(Same format as Port Map Table Entry 0)				0840h – 085Fh
Port Map Table Entry 3	(Same format as Port Map Table Entry 0)				0860h – 087Fh
Port Map Table Entry 4	(Same format as Port Map Table Entry 0)				0880h – 089Fh
Port Map Table Entry 5	(Same format as Port Map Table Entry 0)				08A0h – 08BFh
0 – 2	SGF Frame Register Dwords 0 through 2				08C0h – 08CFh
	Reserved				08D0h – 08DFh
	SGF Control and Status				08E0h
	Reserved				08E4h
0 – 1	Scratchpad 0 – Scratchpad 1				08E8h – 08EFh
	Reserved				08F0h – 090Fh
Register Path Protection					
0	Path 0				0910h
1	Path 1				0914h
2	Path 2				0918h
3	Path 3				091Ch
	Reserved				0920h – 092Bh
	Chip Control Status 0				092Ch
	LED Control				0930h
	Reserved				0934h – 09FFh
	Bridge Configuration Registers (Dual-map)				0A00h – 0AFFh
Multicast Group Table					
0 – 31	Entries 0 through 31				0B00h – 0B7Fh
	Reserved				0B7Fh – 0BFFh

Table 4–6 CSR Register Map *(Continued)*

Dword	Byte 3	Byte 2	Byte 1	Byte 0	CH255 Byte Offset
Semaphore 0					
	Clear				0C00h
	Set				0C04h
	Decrement				0C08h
	Increment				0C0Ch
	Reserved 0				0C10h
	Increment if 0				0C14h
	Reserved 1				0C18h
	Increment if not 0				0C1Ch
Semaphore 1	(Same format as Semaphore 0)				0C20h – 0C3Fh
Reserved					0C40h – 0EFFh
Manufacturing/Diagnostic Registers (not documented)					0F00h – 0FFFh

Table 4–7 maps the address for the Bridge Configuration registers.

Table 4–7 Bridge Configuration Register Map

Byte 3	Byte 2	Byte 1	Byte 0	Bridge Cfg Offset
Device ID		Vendor ID		00h
Status		Command		04h
Class Code			Revision ID	08h
Reserved	Header Type	PMLT	Cache Line Size:	0Ch
Reserved				10h
Reserved				14h
SMLT	Subordinate Bus #	Secondary Bus #	Primary Bus #	18h
Secondary Status		I/O Limit	I/O Base	1Ch
Memory Limit		Memory Base		20h
PF Memory Limit		PF Memory Base		24h
PF Memory Base Upper 32 Bits				28h
PF Memory Limit Upper 32 Bits				2Ch
I/O Limit Upper 16 Bits		I/O Base Upper 16 Bits		30h
Reserved			ECP	34h
Reserved				38h
Bridge Control		Interrupt Pin	Interrupt Line	3Ch
Reserved				40h – 47h
ROM Address				48h
ROM Data				4Ch
Reserved				50h
GPIO Dir Clear	GPIO Data Clear	GPIO Dir Set	GPIO Data Set	54h

Table 4–7 Bridge Configuration Register Map *(Continued)*

Byte 3	Byte 2	Byte 1	Byte 0	Bridge Cfg Offset
Reserved				58h – 5Fh
Register Index 0				60h
Register Data 0				64h
Register Index 1				68h
Register Data 1				6Ch
PM Capabilities		PM Next ECP	PM ECP ID	70h
PM Data	PM P2P Support	PM Control and Status		74h
Slot # Chassis #	Slot # Exp Slot	Slot # Next ECP	Slot # ECP ID	78h
VPD Address		VPD Next ECP	VPD ECP ID	7Ch
VPD Data				80h
Reserved				84h - FFh

4.5 StarFabric Component (SFC) Header Registers

The SFC Header Registers are mapped in Table 4–5 and described in Sections 4.5.1 through 4.5.2.8.5.

4.5.1 SFC Header

These registers are defined for every StarFabric device and must be mapped at the specified Channel 255 offsets.

Note: *Register Access field abbreviations are defined in the Conventions section of the Preface.*

4.5.1.1 SFC Vendor ID

Ch. 255 Byte Offset: 0000h:0003h
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
31:0	FVENDORID	R	1h	Returns 01h, the StarFabric Vendor ID for StarGen.

4.5.1.2 SFC Device ID

Ch. 255 Byte Offset: 0004h:0007h
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
31:0	FDEVICEID	R	2h	Returns 02h, the StarFabric Device ID for the SG1010.

4.5.1.3 SFC OEM Vendor Driver ID

Ch. 255 Byte Offset: 0008h:000Bh
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
31:0	FOEMDRID	R	0h	Returns the OEM Driver ID for the SG1010, which is initialized to 0, but loadable through serial ROM pre-load.

4.5.1.4 Silicon Revision

Ch. 255 Byte Offset: 000Ch:000Fh
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
31:0	SILREVID	R	H/W	Returns the silicon revision ID for the SG1010.

4.5.1.5 StarFabric Protocol Revision

Ch. 255 Byte Offset: 0010h:0013h
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
31:0	FPREVID	R	1h	Returns the Protocol Revision ID for the SG1010, which is initialized to 1, but loadable through serial ROM preload.

4.5.1.6 SFC Base Class ID

Ch. 255 Byte Offset: 0014h:0017h
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
7:0	SBASECL	R	1h	Identifies the switch class. Reads as 1 to indicate that this device is a SG1010-class switch.
15:8	EBASECL	R	0h	Identifies the edge class. Reads as 0 to indicate that this device does not have edge-node functionality.
31:16	ESUBCL	R	0h	Identifies the edge subclass. Reads as 0 because this device has no edge functionality.

4.5.1.7 SFC Programming Interface ID

Ch. 255 Byte Offset: 001Ch:001Fh
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
31:0	FPROGIF	R	0h	Identifies the programming interface for the device. Initialized to 0, but loadable through serial ROM.

4.5.1.8 Fabric ID (FID)

Fabric ID for the device. This value is determined during fabric enumeration and is also the path from the root node.

Ch. 255 Byte Offset: 0020h:0023h
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
4:0	RES	R	0b	Reserved
7:5	PFN	R	111b	Fabric ID Parallel Fabric Number.
10:8	TC	R	111b	Fabric ID Turn Count. Identifies the number of turns from the root note to this device.
13:11	TURN0	R	111b	Fabric ID Turn 0. Identifies Turn 0 on the path from the root node to this node.
16:14	TURN1	R	111b	Fabric ID Turn 1. Identifies Turn 1 on the path from the root node to this node.
19:17	TURN2	R	111b	Fabric ID Turn 2. Identifies Turn 2 on the path from the root node to this node.
22:20	TURN3	R	111b	Fabric ID Turn 3. Identifies Turn 3 on the path from the root node to this node.
25:23	TURN4	R	111b	Fabric ID Turn 4. Identifies Turn 4 on the path from the root node to this node.
28:26	TURN5	R	111b	Fabric ID Turn 5. Identifies Turn 5 on the path from the root node to this node.
31:29	TURN6	R	111b	Fabric ID Turn 6. Identifies Turn 6 on the path from the root node to this node.

4.5.1.9 SFC Capabilities

Ch. 255 Byte Offset: 0024h:0027h
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
0	ARCAP	R	1	Reads as 1 to indicate that the SG1010 has address-routing capability.

7:1	RES	R	0	Reserved
15:8	CRSHR	R	0	Credit sharing capabilities. Reads as 0 to indicate that credits may be reallocated between different CoS and between CoS and turn credits.
31:1	RES	R	0	Reserved.

4.5.1.10 Extended Function List Pointer (ELP)

Ch. 255 Byte Offset: 0028h:002Bh
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
31:0	ELPPTR	R	40h	Pointer to the first ELP list element at Channel 255 offset 40h.

4.5.1.11 SFC Control

Ch. 255 Byte Offset: 002Ch:002Fh
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
0	NRESET	R/W	0h	Node reset. When written with 1, the chip is reset. This reset does not propagate to the fabric. The SG1010 clears this bit to 0 when chip reset is complete.
1	FRSTENA	W1TC	1h	Fabric reset enable. When a 1, enables a fabric reset through the Fabric Reset bit (FRESET). When a 0, a write to the Fabric Reset bit has no effect. Once this bit is written with a 1, it can only be cleared with a chip reset.
2	PRPRST	R/W	0h	Propagate maskable reset. When written with a 1, a maskable reset comma is propagated out all the links. The chip is subsequently reset if the Node Reset bit is also written with a 1. If the Node Reset bit is written with a 0 when this bit is written with a 1, then a chip reset is not performed after the reset propagation.
31:2	RES	R	0	Reserved

4.5.1.12 Fabric Reset

Ch. 255 Byte Offset: 0030h:0033h
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
0	FRESET	R/W	0h	Fabric reset. When written with a 1, and if the Fabric Reset Enable bit (FRSTENA) is 1, the SG1010 initiates an unmaskable reset into the fabric and then resets. The SG1010 clears this bit to 0 when chip reset is complete.
31:1	RES	R	0	Reserved

4.5.2 Extended Function List Elements

These registers comprise a linked list or pointers to function CSRs. Table 4–8 provides a summary of the ELP registers and their values for each function.

Table 4–8 Extended List Pointer (ELP) Summary

Function	ELP Offset	ELP ID	Offset Pointer	Number of Entries	Entry Size	Next ELP Offset
Semaphores	0040h	08h	0C00h	02h	N/A	0060h
SGF	0060h	(1)07h*	08C0h	N/A	N/A	0080h
Port State Table	0080h	0Ch	0380h	06h	08h	00A0h
Link State Table	00A0h	05h	0200h	06h	40h	00C0h
Events	00C0h	06h	0400h	N/A	N/A	00E0h
Port Map Table	00E0h	0Dh	0800h	06h	20h	0100h
Multicast Table	0100h	0Ah	0B00h	20h	04h	0120h
Ch255 Path Protection	0120h	09h	0910h	04h	N/A	0140h
Channel 255 Registers	0140h	00h	0000h	01h	1000h	0160h
Scratchpad	0160h	0Eh	08E8h	02h	04h	0180h
VPD	0180h	0Fh	0A7Ch	N/A	N/A	0000h

* (1) indicates that bit 31 is set, specifying an implementation-specific ELP.

4.5.2.1 Semaphore ELP

4.5.2.1.1 Semaphore ELP ID

Ch. 255 Byte Offset: 0040h:0043h
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
30:0	ELPID8	R	8h	Returns the Semaphore ELP ID.
31	ELPIDM8	R	0h	Reads as a 0 to identify that this ELP is defined by the StarFabric protocol.

4.5.2.1.2 Semaphore Next ELP

Ch. 255 Byte Offset: 0044h:0047h
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
31:0	ELPNXT8	R	60h	Returns the pointer to the next ELP offset.

4.5.2.1.3 Semaphore ELP Revision ID

Ch. 255 Byte Offset: 0048h:004Bh
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
31:0	ELPREV8	R	0h	Returns the revision number of this Semaphore implementation.

4.5.2.1.4 Semaphore ELP Offset Pointer

Ch. 255 Byte Offset: 004Ch:004Fh
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
31:0	ELPOFF8	R	0C00h	Returns the Channel 255 offset of the Semaphore registers.

4.5.2.1.5 Semaphore ELP Number

Ch. 255 Byte Offset: 0050h:0053h
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
31:0	ELPNUM8	R	2h	Returns the number of semaphores implemented by the SG1010.

4.5.2.2 SGF ELP

4.5.2.2.1 SGF ELP ID

Ch. 255 Byte Offset: 0060h:0063h
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
30:0	ELPID7	R	7h	Returns the SGF ELP ID.
31	ELPIDM7	R	1h	Reads as a 1 to indicate that this ELP refers to a device specific function.

4.5.2.2.2 SGF Next ELP

Ch. 255 Byte Offset: 0064h:0067h
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
31:0	ELPNXT7	R	80h	Returns the pointer to the next ELP offset.

4.5.2.2.3 SGF ELP Revision ID

Ch. 255 Byte Offset: 0068h:006Bh
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
31:0	ELPREV7	R	1h	Returns the revision number of this SGF implementation.

4.5.2.2.4 SGF ELP Offset Pointer

Ch. 255 Byte Offset: 006Ch:006Fh
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
31:0	ELPOFF7	R	08C0h	Returns the Channel 255 offset of the SGF registers.

4.5.2.3 Port State Table ELP

4.5.2.3.1 Port State ELP ID

Ch. 255 Byte Offset: 0080h:0083h
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
30:0	ELPIDC	R	Ch	Returns the Port State ELP ID.
31	ELPIDMC	R	0h	Reads as a 0 to identify that this ELP is defined by the StarFabric protocol.

4.5.2.3.2 Port State Next ELP

Ch. 255 Byte Offset: 0084h:0087h
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
31:0	ELPNXTC	R	A0h	Returns the pointer to the next ELP offset.

4.5.2.3.3 Port State ELP Revision ID

Ch. 255 Byte Offset: 0088h:008Bh
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
31:0	ELPREVC	R	0h	Returns the revision number of this Port State implementation.

4.5.2.3.4 Port State ELP Offset Pointer

Ch. 255 Byte Offset: 008Ch:008Fh
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
31:0	ELPOFFC	R	380h	Returns the Channel 255 offset of the Port State registers.

4.5.2.3.5 Port State ELP Number

Ch. 255 Byte Offset: 0090h:0093h
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
31:0	ELPNUMC	R	6h	Returns the number of port state entries implemented by the SG1010.

4.5.2.3.6 Port State ELP Entry Size:

Ch. 255 Byte Offset: 0094h:0097h
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
31:0	ELPSIZC	R	8h	Returns the size in bytes of each port state entry.

4.5.2.4 Link State Table ELP

4.5.2.4.1 Link State ELP ID

Ch. 255 Byte Offset: 00A0h:00A3h
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
30:0	ELPID5	R	5h	Returns the Link State ELP ID.
31	ELPIDM5	R	0h	Reads as a 0 to identify that this ELP is defined by the StarFabric protocol.

4.5.2.4.2 Link State Next ELP

Ch. 255 Byte Offset: 00A4h:00A7h
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
31:0	ELPNXT5	R	C0h	Returns the pointer to the next ELP offset.

4.5.2.4.3 Link State ELP Revision ID

Ch. 255 Byte Offset: 00A8h:00ABh
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
31:0	ELPREV5	R	0h	Returns the revision number of this link state implementation.

4.5.2.4.4 Link State ELP Offset Pointer

Ch. 255 Byte Offset: 00ACh:00AFh
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
31:0	ELPOFF5	R	0200h	Returns the Channel 255 offset of the Link State registers.

4.5.2.4.5 Link State ELP Number

Ch. 255 Byte Offset: 00B0h:00B3h
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
31:0	ELPNUM5	R	6h	Returns the number of link state entries implemented by the SG1010.

4.5.2.4.6 Link State ELP Entry Size:

Ch. 255 Byte Offset: 00B4h:00B7h
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
31:0	ELPSIZ5	R	40h	Returns the size in bytes of each link state entry.

4.5.2.5 Event ELP

4.5.2.5.1 Event ELP ID

Ch. 255 Byte Offset: 00C0h:00C3h
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
30:0	ELPID6	R	6h	Returns the Event ELP ID.
31	ELPIDM6	R	0h	Reads as a 0 to identify that this ELP is defined by the StarFabric protocol.

4.5.2.5.2 Event Next ELP

Ch. 255 Byte Offset: 00C4h:00C7h
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
31:0	ELPNXT6	R	E0h	Returns the pointer to the next ELP offset.

4.5.2.5.3 Event ELP Revision ID

Ch. 255 Byte Offset: 00C8h:00CBh
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
31:0	ELPREV6	R	1h	Returns the revision number of this event implementation.

4.5.2.5.4 Event ELP Offset Pointer

Ch. 255 Byte Offset: 00CCh:00CFh
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
31:0	ELPOFF6	R	0400h	Returns the Channel 255 offset of the Event registers.

4.5.2.6 Port Map ELP

4.5.2.6.1 Port Map ELP ID

Ch. 255 Byte Offset: 00E0h:00E3h
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
30:0	ELPIDD	R	Dh	Returns the Port Map ELP ID.
31	ELPIDMD	R	0h	Reads as a 0 to identify that this ELP is defined by the StarFabric protocol.

4.5.2.6.2 Port Map Next ELP

Ch. 255 Byte Offset: 00E4h:00E7h
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
31:0	ELPNXTD	R	100h	Returns the pointer to the next ELP offset.

4.5.2.6.3 Port Map ELP Revision ID

Ch. 255 Byte Offset: 00E8h:00EBh
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
31:0	ELPREVD	R	0h	Returns the revision number of this Port Map implementation.

4.5.2.6.4 Port Map ELP Offset Pointer

Ch. 255 Byte Offset: 00ECh:00EFh
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
31:0	ELPOFFD	R	0800h	Returns the Channel 255 offset of the Port Map registers.

4.5.2.6.5 Port Map ELP Number

Ch. 255 Byte Offset: 00F0h:00F3h
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
31:0	ELPNUMD	R	6h	Returns the number of Port Map entries implemented by SG1010.

4.5.2.6.6 Port Map ELP Entry Size:

Ch. 255 Byte Offset: 00F4h:00F7h
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
31:0	ELPSIZD	R	20h	Returns the size in bytes of each Port Map entry.

4.5.2.7 Multicast ELP

4.5.2.7.1 Multicast ELP ID

Ch. 255 Byte Offset: 0100h:0103h
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
30:0	ELPIDA	R	Ah	Returns the Multicast ELP ID.
31	ELPIDMA	R	0h	Reads as a 0 to identify that this ELP is defined by the StarFabric protocol.

4.5.2.7.2 Multicast Next ELP

Ch. 255 Byte Offset: 0104h:0107h
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
31:0	ELPNXTA	R	0120h	Returns the pointer to the next ELP offset.

4.5.2.7.3 Multicast ELP Revision ID

Ch. 255 Byte Offset: 0108h:010Bh
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
31:0	ELPREVA	R	0h	Returns the revision number of this Multicast implementation.

4.5.2.7.4 Multicast ELP Offset Pointer

Ch. 255 Byte Offset: 010Ch:010Fh
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
31:0	ELPOFFA	R	0B00h	Returns the Channel 255 offset of the Multicast registers.

4.5.2.7.5 Multicast ELP Number

Ch. 255 Byte Offset: 0110h:0113h
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
31:0	ELPNUMA	R	20h	Returns the number of Multicast entries implemented by the SG1010, including Multicast Group 0 which is reserved for isochronous traffic bandwidth reservation.

4.5.2.7.6 Multicast ELP Entry Size:

Ch. 255 Byte Offset: 0114h:0117h
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
31:0	ELPSIZA	R	4h	Returns the size in bytes of each Multicast entry.

4.5.2.8 Channel 255 Path Protection ELP

4.5.2.8.1 Channel 255 Path Protection ELP ID

Ch. 255 Byte Offset: 0120h:0123h
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
30:0	ELPID9	R	9h	Returns the Channel 255 Path Protection ELP ID.
31	ELPIDM9	R	0h	Reads as a 0 to identify that this ELP is defined by the StarFabric protocol.

4.5.2.8.2 Channel 255 Path Protection Next ELP

Ch. 255 Byte Offset: 0124h:0127h
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
31:0	ELPNXT9	R	0140h	Returns the pointer to the next ELP offset.

4.5.2.8.3 Channel 255 Path Protection ELP Revision ID

Ch. 255 Byte Offset: 0128h:012Bh
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
31:0	ELPREV9	R	0h	Returns the revision number of this Channel 255 Path Protection implementation.

4.5.2.8.4 Channel 255 Path Protection ELP Offset Pointer

Ch. 255 Byte Offset: 012Ch:012Fh
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
31:0	ELPOFF9	R	0910h	Returns the Channel 255 offset of the Channel 255 Path Protection registers.

4.5.2.8.5 Channel 255 Path Protection ELP Number

Ch. 255 Byte Offset: 0130h:0133h
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
31:0	ELPNUM9	R	4h	Returns the number of Channel 255 Path Protection entries implemented by the SG1010.

4.5.2.9 Channel 255 Registers ELP

4.5.2.9.1 Channel 255 Registers ELP ID

Ch. 255 Byte Offset: 0140h:0143h
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
30:0	ELPID0	R	0h	Returns the Channel 255 Registers ELP ID.
31	ELPIDM0	R	0h	Reads as a 0 to identify that this ELP is defined by the StarFabric protocol.

4.5.2.9.2 Channel 255 Registers Next ELP

Ch. 255 Byte Offset: 0144h:0147h
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
31:0	ELPNXT0	R	0160h	Returns the pointer to the next ELP offset.

4.5.2.9.3 Channel 255 Registers ELP Revision ID

Ch. 255 Byte Offset: 0148h:014Bh
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
31:0	ELPREV0	R	0h	Returns the revision number of this Channel 255 Register set.

4.5.2.9.4 Channel 255 Path Protection ELP Offset Pointer

Ch. 255 Byte Offset: 014Ch:014Fh
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
31:0	ELPOFF0	R	0h	Returns the first Channel 255 offset (always offset 0).

4.5.2.9.5 Channel 255 Registers ELP Number

Ch. 255 Byte Offset: 0150h:0153h
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
31:0	ELPNUM0	R	1h	Returns 1.

4.5.2.9.6 Channel 255 Registers ELP Size

Ch. 255 Byte Offset: 0154h:0157h
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
31:0	ELPSIZ0	R	1000h	The size of the Channel 255 register set to the nearest power of 2.

4.5.2.10 Scratchpad ELP

4.5.2.10.1 Scratchpad ELP ID

Ch. 255 Byte Offset: 0160h:0163h
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
30:0	ELPIDE	R	Eh	Returns the Scratchpad ELP ID.
31	ELPIDME	R	0h	Reads as a 0 to identify that this ELP is defined by the StarFabric Protocol.

4.5.2.10.2 Scratchpad Next ELP

Ch. 255 Byte Offset: 0164h:0167h
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
31:0	ELPNXTE	R	0180h	Returns the pointer to the next ELP offset.

4.5.2.10.3 Scratchpad ELP Revision ID

Ch. 255 Byte Offset: 0168h:016Bh
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
31:0	ELPREVE	R	0h	Returns the revision number of the scratchpad registers.

4.5.2.10.4 Scratchpad ELP Offset Pointer

Ch. 255 Byte Offset: 016Ch:016Fh
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
31:0	ELPOFFE	R	8E8h	Returns the offset of the Scratchpad registers.

4.5.2.10.5 Scratchpad ELP Number

Ch. 255 Byte Offset: 0170h:0173h
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
31:0	ELPNUME	R	2h	Returns the number of 32-bit scratchpad registers.

4.5.2.10.6 Scratchpad ELP Size

Ch. 255 Byte Offset: 0174h:0177h
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
31:0	ELPSIZE	R	4h	Returns the size of each scratchpad register in bytes.

4.5.2.11 Vital Product Data (VPD) ELP

4.5.2.11.1 VPD ELP ID

Ch. 255 Byte Offset: 0180h:0183h
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
30:0	ELPIDF	R	Fh	Returns the VPD ELP ID.
31	ELPIDMF	R	0h	Reads as a 0to indicate that this ELP refers to a function defined by the StarFabric protocol.

4.5.2.11.2 VPD Next ELP

Ch. 255 Byte Offset: 0184h:0187h
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
31:0	ELPNXTF	R	0h	Returns the pointer to the next ELP offset. This is the last entry.

4.5.2.11.3 VPD ELP Revision ID

Ch. 255 Byte Offset: 0188h:018Bh
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
31:0	ELPREVF	R	0h	Returns the revision number of this VPD implementation.

4.5.2.11.4 VPD ELP Offset Pointer

Ch. 255 Byte Offset: 018Ch:018Fh
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
31:0	ELPOFFF	R	0A7Ch	Returns the Channel 255 offset of the VPD registers.

4.6 Control and Status Registers (CSRs)

The SG1010 CSRs are mapped into fabric Channel 255 space. They can also be indexed indirectly through PCI configuration space.

4.6.1 Link State Table

These registers contain link-related control and status information. Each register is instantiated once per link.

4.6.1.1 Link Control and Status Register

Link 0 Ch 255 Byte Offset: 200h:203h
Link 1 Ch 255 Byte Offset: 240h:243h
Link 2 Ch 255 Byte Offset: 280h:283h
Link 3 Ch 255 Byte Offset: 2C0h:2C3h
Link 4 Ch 255 Byte Offset: 300h:303h
Link 5 Ch 255 Byte Offset: 340h:343h
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
3:0	PORTNUM	R	0	Specifies the Port State table and, if used, the Port Map table set used by this link. Initialized by hardware during fabric enumeration. This value is based on the Port State Table Link Map. Not meaningful if the link is down.
4	LDIS	R/W	0	Link Disable. When written with a 1, causes the SG1010 to disable this link's transmitters and set the Link Down event bit. When written with a 0, the SG1010 attempts to synchronize the link, and sets the Link Up event bit.
5	ROOT	R	0	When a 1, the port that this link belongs to is the root port. When a 0, the port that this link belongs to is a non-root port. This bit is initialized during fabric enumeration. Additionally, if software modifies the Root bit in the corresponding Port State Table, this Root bit is modified by hardware to be consistent.

Bits	Name	Access	Reset Value	Description
6	F8B10B	WRZ	0	Force 8B/10B error. When written with a 1, the SG1010 forces one 8B/10B error to occur on the link. This register always returns 0 when read.
7	FCRC	WRZ	0	Force CRC error. When written with a 1, the SG1010 forces one CRC error to occur on the link. This register always returns 0 when read.
9:8	LINKST	R	H/W	Current link state. This field reflects the current synchronization state of the link, where: 00b Call state 01b Acknowledge state 10b Reply state 11b Linked state
11:10	HLINKST	R/W1TC	0	Highest link state. Reflects the highest link state transmitted by the link since these bits were last cleared either by software or by reset. These bit are defined as follows: 00b Call state 01b Acknowledge state 10b Reply state 11b Linked state
13:12	RLINKST	R	H/W	Current link state of the receiver (link partner). This field reflects the current synchronization state of the link from the receiver point of view, as follows: 00b Call state 01b Acknowledge state 10b Reply state 11b Linked state
15:14	RHLINKST	R/W1TC	0	Highest link state. Reflects the highest link state received by the link since these bits were last cleared either by software or by reset. These bit are defined as follows: 00b Call state 01b Acknowledge state 10b Reply state 11b Linked state
19:16	COMTYP	R	0	Attached component type. Assigned during fabric enumeration. Currently assigned values: Bit [16]: When 1, has edge node functionality Bit [17]: When 1, has switch functionality Bits [19:18]: Reserved
23:20	RES	R	0	Reserved

Bits	Name	Access	Reset Value	Description
24	TENA	R/W	1	Traffic Enable. Enables credit-based frames to be sent on this link. Set by hardware after the chip is reset, or when a Set State frame with Set TEN=1 is received. Cleared by hardware if the link goes down as described in Section 3.6.1.2. Software may also set or clear this bit.
27:25	LNKSPD	R	000b	Indicates the link speed of this link, and dictates the reset value of the Bandwidth Count register. Reads as 000b to indicate that the full speed of this link is 2 Gbit/second in each direction.
31:28	RES	R	0	Reserved

4.6.1.2 Link Partner Fabric ID

Contains the FID of the node connected to this link, received in the last I Am frame.

Link 0 Ch 255 Byte Offset: 204h:207h
Link 1 Ch 255 Byte Offset: 244h:247h
Link 2 Ch 255 Byte Offset: 284h:287h
Link 3 Ch 255 Byte Offset: 2C4h:2C7h
Link 4 Ch 255 Byte Offset: 304h:307h
Link 5 Ch 255 Byte Offset: 344h:347h
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
2:0	RES	R	0	Reserved
3	LP_TEN	R	1	Link Partner Traffic Enable. Contains the inverse of the TDIS bit in a received I Am frame.
4	RES	R	0	Reserved
7:5	LPPFN	R	111	Contains the Parallel Fabric Number (PFN) of the link partner
10:8	LPTC	R	111b	Contains the FID Turn Count of the link partner
13:11	LPT0	R	111b	Contains the FID Turn 0 of the link partner
16:14	LPT1	R	111b	Contains the FID Turn 1 of the link partner
19:17	LPT2	R	111b	Contains the FID Turn 2 of the link partner
22:20	LPT3	R	111b	Contains the FID Turn 3 of the link partner
25:23	LPT4	R	111b	Contains the FID Turn 4 of the link partner
28:26	LPT5	R	111b	Contains the FID Turn 5 of the link partner
31:29	LPT6	R	111b	Contains the FID Turn 6 of the link partner

4.6.1.3 8B/10B and CRC Error Count

Link 0 Ch 255 Byte Offset: 208h:20Bh
Link 1 Ch 255 Byte Offset: 248h:24Bh
Link 2 Ch 255 Byte Offset: 288h:28Bh
Link 3 Ch 255 Byte Offset: 2C8h:2CBh

Link 4 Ch 255 Byte Offset: 308h:30Bh
Link 5 Ch 255 Byte Offset: 348h:34Bh
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
15:0	CRC	R/W	0	The CRC counter increments by 1 every time the link detects a CRC error. When the counter reaches its maximum value, it wraps and starts counting from 0. A CRC Counter Wrap event is signaled on overflow.
31:16	8B10B	R/W	0	The 8B/10B counter increments by 1 every time an 8B/10B error is detected by this link. When the counter reaches its maximum value, it wraps and starts counting from 0. An 8B/10B Counter Wrap event is signaled on overflow.

4.6.1.4 Frame Counter

Link 0 Ch 255 Byte Offset: 20Ch:20Fh
Link 1 Ch 255 Byte Offset: 24Ch:24Fh
Link 2 Ch 255 Byte Offset: 28Ch:28Fh
Link 3 Ch 255 Byte Offset: 2CCh:2CFh
Link 4 Ch 255 Byte Offset: 30Ch:30Fh
Link 5 Ch 255 Byte Offset: 34Ch:34Fh
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
31:0	FRAMEC	R/W	0	The frame counter increments by 1 every time a non-empty frame is transmitted by this link. When the counter reaches its maximum value, it wraps and starts counting from 0. A Frame Counter Wrap event is signaled on overflow.

4.6.1.5 Line Counter

Link 0 Ch 255 Byte Offset: 210h:213h
Link 1 Ch 255 Byte Offset: 250h:253h
Link 2 Ch 255 Byte Offset: 290h:293h
Link 3 Ch 255 Byte Offset: 2D0h:2D3h
Link 4 Ch 255 Byte Offset: 310h:313h
Link 5 Ch 255 Byte Offset: 350h:353h
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
31:0	LINEC	R/W	0	The line counter increments by 1 every time a 128-bit non-empty line is transmitted by this link. When the counter reaches its maximum value, it wraps and starts counting from 0. A Line Counter Wrap event is signaled on overflow.

4.6.1.6 Empty Frame Counter

Link 0 Ch 255 Byte Offset: 214h:217h
Link 1 Ch 255 Byte Offset: 254h:257h
Link 2 Ch 255 Byte Offset: 294h:297h
Link 3 Ch 255 Byte Offset: 2D4h:2D7h
Link 4 Ch 255 Byte Offset: 314h:317h
Link 5 Ch 255 Byte Offset: 354h:357h
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
31:0	ELINEC	R/W	0	The empty frame counter increments by 1 every time a 128-bit empty frame is transmitted by this link. When the counter reaches its maximum value, it wraps and starts counting from 0. An Empty Frame Counter Wrap event is signaled on overflow.

4.6.1.7 Asynchronous Write Credit Count

Link 0 Ch 255 Byte Offset: 218h
Link 1 Ch 255 Byte Offset: 258h
Link 2 Ch 255 Byte Offset: 298h
Link 3 Ch 255 Byte Offset: 2D8h
Link 4 Ch 255 Byte Offset: 318h
Link 5 Ch 255 Byte Offset: 358h
Size: 1 byte

Bits	Name	Access	Reset Value	Description
7:0	AWCC	R	0	This register reflects the current value of this link's counter for asynchronous write credits reserving buffer space in the link partner. The SG1010 does not implement this counter, and it reads as 0. Asynchronous credits are aliased with address-routed credits.

4.6.1.8 Isochronous Write Credit Count

Link 0 Ch 255 Byte Offset: 219h
Link 1 Ch 255 Byte Offset: 259h
Link 2 Ch 255 Byte Offset: 299h
Link 3 Ch 255 Byte Offset: 2D9h
Link 4 Ch 255 Byte Offset: 319h
Link 5 Ch 255 Byte Offset: 359h
Size: 1 byte

Bits	Name	Access	Reset Value	Description
7:0	IWCC	R	0	This register reflects the current value of this link's counter for isochronous write credits reserving buffer space in the link partner. This counter is reset when the link goes down.

4.6.1.9 HP-asynchronous Write Credit Count

Link 0 Ch 255 Byte Offset: 21Ah
Link 1 Ch 255 Byte Offset: 25Ah
Link 2 Ch 255 Byte Offset: 29Ah
Link 3 Ch 255 Byte Offset: 2Dah
Link 4 Ch 255 Byte Offset: 31Ah
Link 5 Ch 255 Byte Offset: 35Ah
Size: 1 byte

Bits	Name	Access	Reset Value	Description
7:0	HPWCC	R	0	This register reflects the current value of this link's counter for high-priority asynchronous write credits reserving buffer space in the link partner. The SG1010 does not implement this counter, and it reads as 0. HP-asynchronous credits are aliased with provisioning credits.

4.6.1.10 Multicast Write Credit Count

Link 0 Ch 255 Byte Offset: 21Bh
Link 1 Ch 255 Byte Offset: 25Bh
Link 2 Ch 255 Byte Offset: 29Bh
Link 3 Ch 255 Byte Offset: 2DBh
Link 4 Ch 255 Byte Offset: 31Bh
Link 5 Ch 255 Byte Offset: 35Bh
Size: 1 byte

Bits	Name	Access	Reset Value	Description
7:0	MCCC	R	0	This register reflects the current value of this link's counter for multicast credits reserving buffer space in the link partner. This counter is reset when the link goes down.

4.6.1.11 Address-Routed Write Credit Count

Link 0 Ch 255 Byte Offset: 21Ch
Link 1 Ch 255 Byte Offset: 25Ch
Link 2 Ch 255 Byte Offset: 29Ch
Link 3 Ch 255 Byte Offset: 2DCh
Link 4 Ch 255 Byte Offset: 31Ch
Link 5 Ch 255 Byte Offset: 35Ch
Size: 1 byte

Bits	Name	Access	Reset Value	Description
7:0	ARWCC	R	0	This register reflects the current value of this link's counter for address-routed write credits reserving buffer space in the link partner. This counter is reset when the link goes down.

4.6.1.12 HP-Isochronous Write Credit Count

Link 0 Ch 255 Byte Offset: 21Dh
Link 1 Ch 255 Byte Offset: 25Dh
Link 2 Ch 255 Byte Offset: 29Dh
Link 3 Ch 255 Byte Offset: 2DDh
Link 4 Ch 255 Byte Offset: 31Dh
Link 5 Ch 255 Byte Offset: 35Dh
Size: 1 byte

Bits	Name	Access	Reset Value	Description
7:0	HIWCC	R	0	This register reflects the current value of this link's counter for high-priority isochronous write credits reserving buffer space in the link partner. The SG1010 does not implement this counter, and it reads as 0. HP-Isochronous credits are aliased with Isochronous credits.

4.6.1.13 Provisioning Write Credit Count

Link 0 Ch 255 Byte Offset: 21Eh
Link 1 Ch 255 Byte Offset: 25Eh
Link 2 Ch 255 Byte Offset: 29Eh
Link 3 Ch 255 Byte Offset: 2DEh
Link 4 Ch 255 Byte Offset: 31Eh
Link 5 Ch 255 Byte Offset: 35Eh
Size: 1 byte

Bits	Name	Access	Reset Value	Description
7:0	PWCC	R	0	This register reflects the current value of this link's counter for provisioning write credits reserving buffer space in the link partner. This counter is reset when the link goes down.

4.6.1.14 Turn *N* Write Credit Count

Link 0 Turn 0 Ch 255 Byte Offset: 0220h
Link 0 Turn 1 Ch 255 Byte Offset: 0221h
Link 0 Turn 2 Ch 255 Byte Offset: 0222h
Link 0 Turn 3 Ch 255 Byte Offset: 0223h
Link 0 Turn 4 Ch 255 Byte Offset: 0224h
Link 0 Turn 5 Ch 255 Byte Offset: 0225h
Link 0 Turn 6 Ch 255 Byte Offset: 0226h
Link 0 Turn 7 Ch 255 Byte Offset: 0227h
Link 1 Turn 0 Ch 255 Byte Offset: 0260h
Link 1 Turn 1 Ch 255 Byte Offset: 0261h
Link 1 Turn 2 Ch 255 Byte Offset: 0262h
Link 1 Turn 3 Ch 255 Byte Offset: 0263h
Link 1 Turn 4 Ch 255 Byte Offset: 0264h
Link 1 Turn 5 Ch 255 Byte Offset: 0265h
Link 1 Turn 6 Ch 255 Byte Offset: 0266h

Link 1 Turn 7 Ch 255 Byte Offset: 0267h
Link 2 Turn 0 Ch 255 Byte Offset: 02A0h
Link 2 Turn 1 Ch 255 Byte Offset: 02A1h
Link 2 Turn 2 Ch 255 Byte Offset: 02A2h
Link 2 Turn 3 Ch 255 Byte Offset: 02A3h
Link 2 Turn 4 Ch 255 Byte Offset: 02A4h
Link 2 Turn 5 Ch 255 Byte Offset: 02A5h
Link 2 Turn 6 Ch 255 Byte Offset: 02A6h
Link 2 Turn 7 Ch 255 Byte Offset: 02A7h
Link 3 Turn 0 Ch 255 Byte Offset: 02E0h
Link 3 Turn 1 Ch 255 Byte Offset: 02E1h
Link 3 Turn 2 Ch 255 Byte Offset: 02E2h
Link 3 Turn 3 Ch 255 Byte Offset: 02E3h
Link 3 Turn 4 Ch 255 Byte Offset: 02E4h
Link 3 Turn 5 Ch 255 Byte Offset: 02E5h
Link 3 Turn 6 Ch 255 Byte Offset: 02E6h
Link 3 Turn 7 Ch 255 Byte Offset: 02E7h
Link 4 Turn 0 Ch 255 Byte Offset: 0320h
Link 4 Turn 1 Ch 255 Byte Offset: 0321h
Link 4 Turn 2 Ch 255 Byte Offset: 0322h
Link 4 Turn 3 Ch 255 Byte Offset: 0323h
Link 4 Turn 4 Ch 255 Byte Offset: 0324h
Link 4 Turn 5 Ch 255 Byte Offset: 0325h
Link 4 Turn 6 Ch 255 Byte Offset: 0326h
Link 4 Turn 7 Ch 255 Byte Offset: 0327h
Link 5 Turn 0 Ch 255 Byte Offset: 0360h
Link 5 Turn 1 Ch 255 Byte Offset: 0361h
Link 5 Turn 2 Ch 255 Byte Offset: 0362h
Link 5 Turn 3 Ch 255 Byte Offset: 0363h
Link 5 Turn 4 Ch 255 Byte Offset: 0364h
Link 5 Turn 5 Ch 255 Byte Offset: 0365h
Link 5 Turn 6 Ch 255 Byte Offset: 0366h
Link 5 Turn 7 Ch 255 Byte Offset: 0367h
Size: 1 byte

Bits	Name	Access	Reset Value	Description
7:0	T*n*WCC	R	0	This register reflects the current value of this link's counter for turn write credits reserving buffer space in the link partner. There are eight turn credit counters per link. This counter is reset when the link goes down.

4.6.1.15 Asynchronous Request/Isochronous Request Credit Count

Link 0 Ch 255 Byte Offset: 228h
Link 1 Ch 255 Byte Offset: 268h
Link 2 Ch 255 Byte Offset: 2A8h
Link 3 Ch 255 Byte Offset: 2E8h

Link 4 Ch 255 Byte Offset: 328h
Link 5 Ch 255 Byte Offset: 368h
Size: 1 byte

Bits	Name	Access	Reset Value	Description
3:0	ARCC	R	0	This register reflects the current value of this link's counter for asynchronous request credits reserving buffer space in the link partner. The SG1010 does not implement this counter, and it reads as 0. Asynchronous credits are aliased with address-routed credits.
7:4	IRCC	R	0	This register reflects the current value of this link's counter for isochronous request credits reserving buffer space in the link partner. This counter is reset when the link goes down.

4.6.1.16 HP-Asynchronous Request Credit Count

Link 0 Ch 255 Byte Offset: 229h
Link 1 Ch 255 Byte Offset: 269h
Link 2 Ch 255 Byte Offset: 2A9h
Link 3 Ch 255 Byte Offset: 2E9h
Link 4 Ch 255 Byte Offset: 329h
Link 5 Ch 255 Byte Offset: 369h
Size: 1 byte

Bits	Name	Access	Reset Value	Description
3:0	HPRCC	R	0	This register reflects the current value of this link's counter for the high-priority asynchronous request credits reserving buffer space in the link partner. The SG1010 does not implement this counter, and it reads as 0. HP-asynchronous credits are aliased with provisioning credits.
7:4	RES	R	0	Reserved

4.6.1.17 Address-Routed Request/HP-Isochronous Request Credit Count

Link 0 Ch 255 Byte Offset: 22Ah
Link 1 Ch 255 Byte Offset: 26Ah
Link 2 Ch 255 Byte Offset: 2AAh
Link 3 Ch 255 Byte Offset: 2EAh

Link 4 Ch 255 Byte Offset: 32Ah
Link 5 Ch 255 Byte Offset: 36Ah
Size: 1 byte

Bits	Name	Access	Reset Value	Description
3:0	ARRCC	R	0	This register reflects the current value of this link's counter for address-routed request credits reserving buffer space in the link partner. This counter is reset when the link goes down.
7:4	HIRCC	R	0	This register reflects the current value of this link's counter for high-priority isochronous request credits reserving buffer space in the link partner. The SG1010 does not implement this counter, and it reads as 0. HP-isochronous credits are aliased with isochronous credits.

4.6.1.18 Provisioning Request Credit Count

Link 0 Ch 255 Byte Offset: 22Bh
Link 1 Ch 255 Byte Offset: 26Bh
Link 2 Ch 255 Byte Offset: 2ABh
Link 3 Ch 255 Byte Offset: 2EBh
Link 4 Ch 255 Byte Offset: 32Bh
Link 5 Ch 255 Byte Offset: 36Bh
Size: 1 byte

Bits	Name	Access	Reset Value	Description
3:0	PRCC	R	0	This register reflects the current value of this link's counter for provisioning request credits reserving buffer space in the link partner. This counter is reset when the link goes down.
7:4	RES	R	0	Reserved

4.6.1.19 Turn *N* Request Credit Count

Link 0 Turn 0/1 Ch 255 Byte Offset: 022Ch
Link 0 Turn 2/3 Ch 255 Byte Offset: 022Dh
Link 0 Turn 4/5 Ch 255 Byte Offset: 022Eh
Link 0 Turn 6/7 Ch 255 Byte Offset: 022Fh
Link 1 Turn 0/1 Ch 255 Byte Offset: 026Ch
Link 1 Turn 2/3 Ch 255 Byte Offset: 026Dh
Link 1 Turn 4/5 Ch 255 Byte Offset: 026Eh
Link 1 Turn 6/7 Ch 255 Byte Offset: 026Fh
Link 2 Turn 0/1 Ch 255 Byte Offset: 02ACh
Link 2 Turn 2/3 Ch 255 Byte Offset: 02ADh
Link 2 Turn 4/5 Ch 255 Byte Offset: 02AEh
Link 2 Turn 6/7 Ch 255 Byte Offset: 02AFh
Link 3 Turn 0/1 Ch 255 Byte Offset: 02ECh
Link 3 Turn 2/3 Ch 255 Byte Offset: 02EDh
Link 3 Turn 4/5 Ch 255 Byte Offset: 02EEh

Link 3 Turn 6/7 Ch 255 Byte Offset: 02EFh
Link 4 Turn 0/1 Ch 255 Byte Offset: 032Ch
Link 4 Turn 2/3 Ch 255 Byte Offset: 032Dh
Link 4 Turn 4/5 Ch 255 Byte Offset: 032Eh
Link 4 Turn 6/7 Ch 255 Byte Offset: 032Fh
Link 5 Turn 0/1 Ch 255 Byte Offset: 036Ch
Link 5 Turn 2/3 Ch 255 Byte Offset: 036Dh
Link 5 Turn 4/5 Ch 255 Byte Offset: 036Eh
Link 5 Turn 6/7 Ch 255 Byte Offset: 036Fh
Size: 1 byte

Bits	Name	Access	Reset Value	Description
3:0	T*n*RCC	R	0	This register reflects the current value of this link's counter for turn *n* request credits reserving buffer space in the link partner. There are eight four-bit turn request credit counters per link. This counter is reset when the link goes down.
7:4	T*n*RCC	R	0	This register reflects the current value of this link's counter for turn *n+1* request credits reserving buffer space in the link partner. There are eight four-bit turn request credit counters per link. This counter is reset when the link goes down.

4.6.1.20 Bandwidth Count

Link 0 Ch 255 Byte Offset: 0230h:0231h
Link 1 Ch 255 Byte Offset: 0270h:0271h
Link 2 Ch 255 Byte Offset: 02B0h:02B1h
Link 3 Ch 255 Byte Offset: 02F0h:02F1h
Link 4 Ch 255 Byte Offset: 0330h:0331h
Link 5 Ch 255 Byte Offset: 0370h:0371h
Size: 2 bytes

Bits	Name	Access	Reset Value	Description
11:0	BWCNT	R/W	5FFh	This register reflects the current value of the bandwidth counter for this link.
12	BWOVFL	R/W	0	Bandwidth counter bundled link overflow bit. This is an extra bit in the bandwidth counter to accommodate bandwidth replacement when the link is bundled.
15:13	RES	R	0	Reserved

4.6.1.21 Differential Pair State

Link 0 Ch 255 Byte Offset: 0232h
Link 1 Ch 255 Byte Offset: 0272h
Link 2 Ch 255 Byte Offset: 02B2h
Link 3 Ch 255 Byte Offset: 02F2h

Link 4 Ch 255 Byte Offset: 0332h
Link 5 Ch 255 Byte Offset: 0372h
Size: 1 byte per entry, 2 bytes total

Bits	Name	Access	Reset Value	Description
3:0	TXST	R	0	This register reflects the operational transmit differential pairs for this link. Each bit corresponds to a differential pair.This state is extracted from the differential pair state in last valid synchronization special frame received (Call, Ack, Reply, or Linked frame). If the link is not receiving valid frames, this state may not be accurate. Bits [3:0] correspond to TX*n*P/TX*n*N[3:0]. When a bit is: 0: A differential pair is not operational. 1: A differential pair is operational.
7:4	RXST	R	0	This register reflects the operational receive differential pairs for this link, that is, which pairs are receiving good 8B/10B data. Each bit corresponds to a differential pair. Bits [3:0] correspond to RX*n*P/RX*n*N[3:0]. When a bit is: 0: A differential pair is not operational. 1: A differential pair is operational.

4.6.1.22 Default CoS Credit

The Default CoS Credit is the number of CoS credits the SG1010 uses when initializing its link partner's CoS write and request credit counters after link synchronization.

Link 0 Ch 255 Byte Offset: 0234h:023Dh
Link 1 Ch 255 Byte Offset: 0274h:027Dh
Link 2 Ch 255 Byte Offset: 02B4h:02BDh
Link 3 Ch 255 Byte Offset: 02F4h:02FDh
Link 4 Ch 255 Byte Offset: 0334h:033Dh
Link 5 Ch 255 Byte Offset: 0374h:037Dh
Size: 10 bytes per entry

Bits	Name	Access	Reset Value	Description
3:0	DCARR	R	0h	This field reflects the default value of the SG1010's asynchronous request line credits.
10:4	DCARW	R	0h	This field reflects the default value of the SG1010's asynchronous write line credits.
14:11	DCIR	R	7h (7)	This field reflects the default value of the SG1010's isochronous request line credits.
21:15	DCIW	R	3Ch (60)	This field reflects the default value of the SG1010's isochronous write line credits.
25:22	DCHPR	R	0h	This field reflects the default value of the SG1010's high-priority asynchronous request line credits.

Bits	Name	Access	Reset Value	Description
32:26	DCHPW	R	0h	This field reflects the default value of the SG1010's high-priority asynchronous write line credits. The Dword bit breakdown is as follows: [5:0] Dword 0 bits[31:26] [6] Dword 1 bit[0]
36:33	RES	R	0h	Reserved. The Dword bit breakdown is Dword 1 bits[4:1].
43:37	DCMCST	R	3Ch (60)	This field reflects the default value of the SG1010's multicast line credits. The Dword bit breakdown is Dword 1 bits[11:5].
47:44	DCASR	R	7h (7)	This field reflects the default value of the SG1010's address-routed request line credits. The Dword bit breakdown is Dword 1 bits[15:12].
54:48	DCASW	R	5Ah (90)	This field reflects the default value of the SG1010's address-routed write line credits. The Dword bit breakdown is Dword 1 bits[22:16].
58:55	DCRR	R	0h	This field reflects the default value of the SG1010's high-priority isochronous request line credits. The Dword bit breakdown is Dword 1 bits[26:23].
65:59	DCRW	R	0h	This field reflects the default value of the SG1010's high-priority isochronous write line credits. The Dword bit breakdown is as follows: [63:59] Dword 1 bits[31:27] [65:64] Dword 2 bits[1:0]
69:66	DCPRR	R	7h (7)	This field reflects the default value of the SG1010's provisioning request line credits. The Dword bit breakdown is Dword 2 bits[5:2].
76:70	DCPRW	R	3Ch (60)	This field reflects the default value of the SG1010's provisioning write line credits. The Dword bit breakdown is Dword 2 bits[12:6].
79:77	RES	R	0	Reserved

4.6.1.23 Default Turn Credit

Link 0 Ch 255 Byte Offset: 023Eh:023Fh
Link 1 Ch 255 Byte Offset: 027Eh:027Fh
Link 2 Ch 255 Byte Offset: 02BEh:02BFh
Link 3 Ch 255 Byte Offset: 02FEh:02FFh

Link 4 Ch 255 Byte Offset: 033Eh:033Fh
Link 5 Ch 255 Byte Offset: 037Eh:037Fh
Size: 2 bytes per entry

Bit	Name	Access	Reset Value	Description
7:0	DCTW	R	0	This register reflects the SG1010's default turn write credits. Since the SG1010 does not support default turn credits, this register returns 0.
15:8	DCTR	R	0	This register reflects the SG1010's default turn request credits. Since the SG1010 does not support default turn credits, this register returns 0.

4.6.2 Port State Table

4.6.2.1 Port State *x* Link Map

Port 0 Ch 255 Byte Offset: 0380h:0383h
Port 1 Ch 255 Byte Offset: 0388h:038Bh
Port 2 Ch 255 Byte Offset: 0390h:0393h
Port 3 Ch 255 Byte Offset: 0398h:039Bh
Port 4 Ch 255 Byte Offset: 03A0h:03A3h
Port 5 Ch 255 Byte Offset: 03A8h:03ABh
Size: 1 Dword per port, 6 Dwords total

Bits	Name	Access	Reset Value	Description
5:0	P2LMAP*x*	R	0	These bits reflect the links that belong to Port *x*. A one in bit [0] indicates that Link 0 belongs to Port *x;* A one in bit [1] indicates that Link 1 belongs to Port *x,* and so on. A 0 indicates that the link does not belong to that port. Initialized by SG1010 hardware after fabric enumeration.
31:6	RES	R	0	Reserved

4.6.2.2 Port State *x* Control and Status

Port 0 Ch 255 Byte Offset: 0384h:0387h
Port 1 Ch 255 Byte Offset: 038Ch:038Fh
Port 2 Ch 255 Byte Offset: 0394h:0397h
Port 3 Ch 255 Byte Offset: 039Ch:039Fh

Port 4 Ch 255 Byte Offset: 03A4h:03A7h
Port 5 Ch 255 Byte Offset: 03ACh:03AFh
Size: 1 Dword per port, 6 Dwords total

Bits	Name	Access	Reset Value	Description
0	PORTUP	R	1	Port Up/Down State. This bit reflects the state of the Traffic Enable bits for the port's links. When a 0, the port is down (the Traffic Enable bits for all links in the port are clear). When a 1, the port is up (the Traffic Enable bit is set for at least one link in the port).
1	RES	R	0	Reserved.
2	ROOT	R/W	0	When a 0, this port is not the root port. When a 1, this port is the root port. Initialized by hardware during fabric enumeration. Software may also modify this bit. The Root bit in the Link State Table is modified by hardware to be consistent with this value.
3	RES	R	0	Reserved
4	PMTENA	R/W	0	Port Map Table Enable. When a 0, the Port Map Table is disabled for address routing, and therefore no positively decoded address-routed frames can exit this port. When a 1, the Port Map Table is enabled for address decoding for address-routed frames. After fabric enumeration, the downstream ports in the PCI hierarchy are enabled by hardware. This bit may also be programmed by software, unless the port is down; in this case the bit is read only as 0.
5	SARENA	R/W	0	Smart Address Enable. When a 1 and the Port Map Table Enable bit (PMTENA) is set, disables the decoding of configuration addresses against the Port Map Table. When a 0, and the Port Map Table Enable bit is set, all address types are decoded. If the Port Map Table Enable bit is not set, this bit has no effect on address decoding.
6	RESETDIS	R/W	0	Maskable Reset Disable. When a 0, if the SG1010 responds to a maskable reset received on the corresponding port. When a 1, the SG1010 ignores maskable resets received on the corresponding port. This bit is written during fabric enumeration and can also be modified by software.
31:7	RES	R	0	Reserved.

4.6.3 Event Registers

4.6.3.1 Chip Event Table

Note: *Currently only event table entries 4 through 11 and 27 and 28 are implemented. The unused entries are reserved.*

Entry 0 Ch 255 Byte Offset: 0400h:0401h
Entry 1 Ch 255 Byte Offset: 0402h:0403h
Entry 2 Ch 255 Byte Offset: 0404h:0405h
Entry 3 Ch 255 Byte Offset: 0406h:0407h
Entry 4 Ch 255 Byte Offset: 0408h:0409h
Entry 5 Ch 255 Byte Offset: 040Ah:040Bh
Entry 6 Ch 255 Byte Offset: 040Ch:040Dh
Entry 7 Ch 255 Byte Offset: 040Eh:040Fh
Entry 8 Ch 255 Byte Offset: 0410h:0411h
Entry 9 Ch 255 Byte Offset: 0412h:0413h
Entry 10 Ch 255 Byte Offset: 0414h:0415h
Entry 11 Ch 255 Byte Offset: 0416h:0417h
Entry 12 Ch 255 Byte Offset: 0418h:0419h
Entry 13 Ch 255 Byte Offset: 041Ah:041Bh
Entry 14 Ch 255 Byte Offset: 041Ch:041Dh
Entry 15 Ch 255 Byte Offset: 041Eh:041Fh
Entry 16 Ch 255 Byte Offset: 0420h:0421h
Entry 17 Ch 255 Byte Offset: 0422h:0423h
Entry 18 Ch 255 Byte Offset: 0424h:0425h
Entry 19 Ch 255 Byte Offset: 0426h:0427h
Entry 20 Ch 255 Byte Offset: 0428h:0429h
Entry 21 Ch 255 Byte Offset: 042Ah:042Bh
Entry 22 Ch 255 Byte Offset: 042Ch:042Dh
Entry 23 Ch 255 Byte Offset: 042Eh:042Fh
Entry 24 Ch 255 Byte Offset: 0430h:0431h
Entry 25 Ch 255 Byte Offset: 0432h:0433h
Entry 26 Ch 255 Byte Offset: 0434h:0435h
Entry 27 Ch 255 Byte Offset: 0436h:0437h
Entry 28 Ch 255 Byte Offset: 0438h:0439h
Entry 29 Ch 255 Byte Offset: 043Ah:043Bh
Entry 30 Ch 255 Byte Offset: 043Ch:043Dh
Entry 31 Ch 255 Byte Offset: 043Eh:043Fh
Size: 2 bytes per entry, 32 entries

Bits	Name	Access	Rest Value	Description
1:0	EPIND	R/W	0	Event Path Table Index. Selects one of four entries in the Event Path Table (see Section 4.6.3.2).
5:2	RES	R	0	Reserved
6	ESEND	R/W	0	Event Frame Send Mode. When a 0, list mode is used for dispatching events. When a 1, polled mode is used for dispatching events.
7	RES	R	0	Reserved
14:8	EMUADR	R/W	0	EMU Address. The seven-bit EMU address used for chip event frames. Bits [6:1] select the EMU to be used at the event frame terminus and bit [0] selects the operation to be performed by the EMU.
15	RES	R	0	Reserved

4.6.3.2 Event Path Table

Entry 0 Ch 255 Byte Offset: 0480h:0483h
Entry 1 Ch 255 Byte Offset: 0484h:0487h
Entry 2 Ch 255 Byte Offset: 0488h:048Bh
Entry 3 Ch 255 Byte Offset: 048Ch:048Fh
Size: 4 bytes per entry, 4 entries

Bits	Name	Access	Reset Value	Description
20:0	EPATH	R/W	0	Path specification for the event frame
23:21	ECOS	R/W	0	Class of service for the event frame
26:24	EOUTPORT	R/W	0	Output port for the event frame.
30:27	RES	R	0	Reserved
31	VALID	R/W	0	Entry valid. When a 0, this entry is not valid and an event frame is not sent for those events using this entry. When a 1, the entry is valid.

4.6.3.3 Event Mask

4.6.3.3.1 Event Mask W1TC

Event Mask WTC 0 Ch 255 Byte Offset: 04A0h:04A3h
Event Mask WTC 1 Ch 255 Byte Offset: 04A4h:04A7h
Event Mask WTC 2 Ch 255 Byte Offset: 04A8h:04ABh
Size: 12 bytes

Bits	Name	Access	Reset Value	Description
31:0	EMSKC	R/W1TC	FFFF FFFFh	Event Mask Write-1-to-Clear. Each bit in this register is an event mask for the corresponding bit in the Event Status register (see Section 4.6.3.4). When a 0, the event is not masked. When a 1, the event is masked. When a bit is written with a 1, the mask bit is cleared. When this register is read, it returns the value of the mask register.

4.6.3.3.2 Event Status Mask Write-1-to-Set

Event Mask WTS 0 Ch 255 Byte Offset: 04C0h:04C3h
Event Mask WTS 1 Ch 255 Byte Offset: 04C4h:04C7h
Event Mask WTS 2 Ch 255 Byte Offset: 04C8h:04CBh
Size: 12 bytes

Bits	Name	Access	Reset Value	Description
31:0	EMSKS	R/W1TS	FFFF FFFFh	Event Mask Write-1-to-set. Each bit in this register is an event mask for the corresponding bit in the Event Status register (see Section 4.6.3.4). When a 0, the event is not masked. When a 1, the event is masked. When a bit is written with a 1, the mask bit is set. When this register is read, it returns the value of the mask register.

4.6.3.4 Raw Event Status

Raw Event Status 0 Ch 255 Byte Offset: 04E0h:04E3h
Raw Event Status 1 Ch 255 Byte Offset: 04E4h:04E7h
Raw Event Status 2 Ch 255 Byte Offset: 04E8h:04Ebh
Size: 12 bytes

Bits	Name	Access	Reset Value	Description
31:0	RAWST	R/W1TC	0	Raw Event Status bits. Each bit in this register represents a different event. The SG1010 sets a bit when the event arbiter selects the event from pending event status. Software clears a bit by writing a 1 to it. See the individual event bit assignments in Section 4.6.3.6.

4.6.3.5 Event Status

Event Status 0 Ch 255 Byte Offset: 0500h:0503h
Event Status 1 Ch 255 Byte Offset: 0504h:0507h
Event Status 2 Ch 255 Byte Offset: 0508h:050Bh
Size: 4 bytes per register, 3 registers

Bits	Name	Access	Reset Value	Description
31:0	EDSTEV	R/W1TC	0	Event Status register. Each event is represented with a bit in this register. The SG1010 sets an event bit when the event occurs, the corresponding event mask bit is clear, and the event arbiter selects the event for dispatch. Each bit is cleared by writing a 1. The individual bit assignments can be found in Section 4.6.3.6.

4.6.3.6 Event Bit Assignments

Table 4–1 specifies the bit assignments used in the SG1010's Raw Event Status, Event Mask, and Event Status registers.

Table 4–1 Chip Event Status Bit Assignments

Dword	Dword Bit	Status	Set When…
0	0	Port 0 Down	All links in Port 0 are down
0	1	Port 1 Down	All links in Port 1 are down
0	2	Port 2 Down	All links in Port 2 are down
0	3	Port 3 Down	All links in Port 3 are down
0	4	Port 4 Down	All links in Port 4 are down
0	5	Port 5 Down	All links in Port 5 are down
0	6	Link 0 Down	Link 0 loses and cannot regain sync
0	7	Link 1 Down	Link 1 loses and cannot regain sync
0	8	Link 2 Down	Link 2 loses and cannot regain sync
0	9	Link 3 Down	Link 3 loses and cannot regain sync
0	10	Link 4 Down	Link 4 loses and cannot regain sync
0	11	Link 5 Down	Link 5 loses and cannot regain sync
0	12	Link 0 Fragile	Link 0 has only one pair working
0	13	Link 1 Fragile	Link 1 has only one pair working
0	14	Link 2 Fragile	Link 2 has only one pair working
0	15	Link 3 Fragile	Link 3 has only one pair working
0	16	Link 4 Fragile	Link 4 has only one pair working
0	17	Link 5 Fragile	Link 5 has only one pair working
0	18	Link 0 Up	Link 0 was down and just re-synced
0	19	Link 1 Up	Link 1 was down and just re-synced
0	20	Link 2 Up	Link 2 was down and just re-synced
0	21	Link 3 Up	Link 3 was down and just re-synced
0	22	Link 4 Up	Link 4 was down and just re-synced
0	23	Link 5 Up	Link 5 was down and just re-synced
0	24	Link 0 CRC Counter Wrap	
0	25	Link 0 8B/10B Counter Wrap	
0	26	Reserved	
0	27	Link 0 Frame Count Wrap	
0	28	Link 0 Line Count Wrap	
0	29	Link 0 Empty Line Wrap	
0	30	Reserved	
0	31	Reserved	
1	0	Link 1 CRC Counter Wrap	
1	1	Link 1 8B/10B Counter Wrap	

Table 4–1 Chip Event Status Bit Assignments *(Continued)*

Dword	Dword Bit	Status	Set When…
1	2	Reserved	
1	3	Link 1 Frame Count Wrap	
1	4	Link 1 Line Count Wrap	
1	5	Link 1 Empty Line Wrap	
1	6	Link 2 CRC Counter Wrap	
1	7	Link 2 8B/10B Counter Wrap	
1	8	Reserved	
1	9	Link 2 Frame Count Wrap	
1	10	Link 2 Line Count Wrap	
1	11	Link 2 Empty Line Wrap	
1	12	Link 3 CRC Counter Wrap	
1	13	Link 3 8B/10B Counter Wrap	
1	14	Reserved	
1	15	Link 3 Frame Count Wrap	
1	16	Link 3 Line Count Wrap	
1	17	Link 3 Empty Line Wrap	
1	18	Link 4 CRC Counter Wrap	
1	19	Link 4 8B/10B Counter Wrap	
1	20	Reserved	
1	21	Link 4 Frame Count Wrap	
1	22	Link 4 Line Count Wrap	
1	23	Link 4 Empty Line Wrap	
1	24	Link 5 CRC Counter Wrap	
1	25	Link 5 8B/10B Counter Wrap	
1	26	Reserved	
1	27	Link 5 Frame Count Wrap	
1	28	Link 5 Line Count Wrap	
1	29	Link 5 Empty Line Wrap	
1	30	Reserved	
1	31	Reserved	
2	0	Reserved	
2	1	Reserved	
2	2	Reserved	
2	3	Reserved	
2	4	Destination channel 255 Range error	Ch 255 offset out of range
2	5	Destination channel 255 path protection error	Ch 255 path protection check failed
2	6	Address-routing Failure (non-cfg)	Memory or I/O address-routed decode failure (port map or inverse decode)

Table 4–1 Chip Event Status Bit Assignments *(Continued)*

Dword	Dword Bit	Status	Set When…
2	7	Fabric PCI special cycle	Cfg T1 to PCI Special Cycle detected in fabric
2	8	S/W Multicast Distribution Failure	Multicast send failed due to (1) no members in group (3) Input port does not match (4) turn count = 7
2	9	H/W Multicast Distribution Failure	Multicast send failed due to port down
2	10	Unsupported Multicast Group ID	Multicast Group ID > 31
2	11	SGF Done	
2	12	Event Overrun	Routing event cache full – cannot accept event
2	13	Address-routing Failure on Write w/o Ack	
2	14	PCI Status: Received Master Abort (Address-routing Failure)	Received Master Abort bit set in Secondary Status register due downstream Port Map decode failure
2	15-31	Reserved	

4.6.3.7 Event Dispatch Control

Ch 255 Byte Offset: 0520h:0523h
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
0	CETENA	R/W	0	Chip Event Table Enable. When a 0, the SG1010 does not use the event tables to generate event frames but instead uses the default target values. When a 1, the SG1010 uses the event tables to generate event frames.
31:1	RES	R	0	Reserved

4.6.4 Port Address Map Table

These registers hold the configuration, memory, and I/O PCI address regions for the link partners of the six possible ports, and are used to select the output port for address-routed frames.

4.6.4.1 Port Map Command and Bridge Control

Port 0 Ch 255 Byte Offset: 0800h:0803h
Port 1 Ch 255 Byte Offset: 0820h:0823h
Port 2 Ch 255 Byte Offset: 0840h:0843h
Port 3 Ch 255 Byte Offset: 0860h:0863h

Port 4 Ch 255 Byte Offset: 0880h:0883h
Port 5 Ch 255 Byte Offset: 08A0h:08A3h
Size: 24 bytes total, 4 bytes per port

Bits	Name	Access	Reset Value	Description
15:0	CMDP*x*	R/W	0	Port Map *x* Command register. Contains a copy of the Command configuration register for the link partner connected to port *x*.
31:16	BCP*x*	R/W	0	Port Map *x* Bridge Control register. Contains a copy of the Bridge Control configuration register for the link partner connected to port *x*.

4.6.4.2 Port Map Bus Numbers

The Port Map Bus Numbers contain the PCI bus numbers defining the configuration ranges for ports 0 through port 5.

Port 0 Ch 255 Byte Offset: 0804h:0807h
Port 1 Ch 255 Byte Offset: 0824h:0827h
Port 2 Ch 255 Byte Offset: 0844h:0847h
Port 3 Ch 255 Byte Offset: 0864h:0867h
Port 4 Ch 255 Byte Offset: 0884h:0887h
Port 5 Ch 255 Byte Offset: 08A4h:08A7h
Size: 24 bytes total, 4 bytes per port

Bits	Name	Access	Reset Value	Description
7:0	PRBUS*x*	R/W	0	Port Map *x* Primary Bus Number. Contains a copy of the Primary Bus Number register for the link partner connected to port *x*.
15:8	SECBUS*x*	R/W	0	Port Map *x* Secondary Bus Number. Contains a copy of the Secondary Bus Number register for the link partner connected to port *x*.
23:16	SUBBUS*x*	R/W	0	Port Map *x* Subordinate Bus Number. Contains a copy of the Subordinate Bus Number register for the link partner connected to port *x*.
31:24	RES	R	0	Reserved

4.6.4.3 Port Map I/O Base and Limit

The Port Map I/O Base and Limit contains the lower I/O base and limit bits for ports 0 through port 5.

Port 0 Ch 255 Byte Offset: 0808h:080Bh
Port 1 Ch 255 Byte Offset: 0828h:082Bh
Port 2 Ch 255 Byte Offset: 0848h:084Bh
Port 3 Ch 255 Byte Offset: 0868h:086Bh

Port 4 Ch 255 Byte Offset: 0888h:088Bh
Port 5 Ch 255 Byte Offset: 08A8h:08ABh
Size: 24 bytes total, 4 bytes per port

Bits	Name	Access	Reset Value	Description
3:0	RES	R	0	Reserved
7:4	PIOB*x*	R/W	0	Port Map *x* I/O Base. Contains a copy of the I/O Base register for the link partner connected to port *x*. This field holds address bits [15:12].
11:8	RES	R	0	Reserved
15:12	PIOL*x*	R/W	0	Port Map *x* I/O Limit. Contains a copy of the I/O Limit register for the link partner connected to port *x*. This field holds address bits [15:12].
31:16	RES	R	0	Reserved.

4.6.4.4 Port Map Memory Base and Limit

The Port Map Memory Base and Limit define the memory address ranges for ports 0 through port 5.

Port 0 Ch 255 Byte Offset: 080Ch:080Fh
Port 1 Ch 255 Byte Offset: 082Ch:082Fh
Port 2 Ch 255 Byte Offset: 084Ch:084Fh
Port 3 Ch 255 Byte Offset: 086Ch:086Fh
Port 4 Ch 255 Byte Offset: 088Ch:088Fh
Port 5 Ch 255 Byte Offset: 08ACh:08AFh
Size: 24 bytes total, 4 bytes per port

Bits	Name	Access	Reset Value	Description
3:0	RES	R	0	Reserved
15:4	PMB*x*	R/W	0	Port Map *x* Memory Base. Contains a copy of the Memory Base register for the link partner connected to port *x*. This field holds address bits [31:20].
19:16	RES	R	0	Reserved
31:20	PML*x*	R/W	0	Port Map *x* Memory Limit. Contains a copy of the Memory Limit register for the link partner connected to port *x*. This field holds address bits [31:20].

4.6.4.5 Port Map Prefetchable Memory Base and Limit

The Port Map Prefetchable Memory Base and Limit define the lower bits of the prefetchable memory address ranges for ports 0 through port 5.

Port 0 Ch 255 Byte Offset: 0810h:0813h
Port 1 Ch 255 Byte Offset: 0830h:0833h
Port 2 Ch 255 Byte Offset: 0850h:0853h
Port 3 Ch 255 Byte Offset: 0870h:0873h

Port 4 Ch 255 Byte Offset: 0890h:0893h
Port 5 Ch 255 Byte Offset: 08B0h:08B3h
Size: 24 bytes total, 4 bytes per port

Bits	Name	Access	Reset Value	Description
3:0	RES	R	0	Reserved
15:4	PPMB*x*	R/W	0	Port Map *x* Prefetchable Memory Base. Contains a copy of the Prefetchable Memory Base register for the link partner connected to port *x*. This field holds address bits [31:20].
19:16	RES	R	0	Reserved
31:20	PPML*x*	R/W	0	Port Map *x* Prefetchable Memory Limit. Contains a copy of the Prefetchable Memory Limit register for the link partner connected to port *x*. This field holds address bits [31:20].

4.6.4.6 Port Map Prefetchable Memory Base Upper 32 Bits

The Port Map Prefetchable Memory Base and Limit define the upper bits of the prefetchable memory address ranges for port 0 through port 5.

Port 0 Ch 255 Byte Offset: 0814h:0817h
Port 1 Ch 255 Byte Offset: 0834h:0837h
Port 2 Ch 255 Byte Offset: 0854h:0857h
Port 3 Ch 255 Byte Offset: 0874h:0877h
Port 4 Ch 255 Byte Offset: 0894h:0897h
Port 5 Ch 255 Byte Offset: 08B4h:08B7h
Size: 24 bytes total, 4 bytes per port

Bits	Name	Access	Reset Value	Description
31:0	PPMBU*x*	R/W	0	Port Map *x* Prefetchable Memory Base Upper 32 Bits. Contains a copy of the Prefetchable Memory Base Upper 32 Bits register for the link partner connected to port *x*. This field holds address bits [63:32].

4.6.4.7 Port Map Prefetchable Memory Limit Upper 32 Bits

The Port Map Prefetchable Memory Base and Limit define the upper bits of the prefetchable memory address ranges for port 0 through port 5.

Port 0 Ch 255 Byte Offset: 0818h:081Bh
Port 1 Ch 255 Byte Offset: 0838h:083Bh
Port 2 Ch 255 Byte Offset: 0858h:085Bh
Port 3 Ch 255 Byte Offset: 0878h:087Bh

Port 4 Ch 255 Byte Offset: 0898h:089Bh
Port 5 Ch 255 Byte Offset: 08B8h:08BBh
Size: 24 bytes total, 4 bytes per port

Bits	Name	Access	Reset Value	Description
31:0	PPMLU*x*	R/W	0	Port Map *x* Prefetchable Memory Limit Upper 32 Bits. Contains a copy of the Prefetchable Memory Limit Upper 32 Bits register for the link partner connected to port *x*. This field holds address bits [63:32].

4.6.4.8 Port Map I/O Base and Limit Upper 16 Bits

The Port Map I/O Base and Limit Upper 16 Bits contains the upper I/O base and limit bits for port 0 through port 5.

Port 0 Ch 255 Byte Offset: 081Ch:081Fh
Port 1 Ch 255 Byte Offset: 083Ch:083Fh
Port 2 Ch 255 Byte Offset: 085Ch:085Fh
Port 3 Ch 255 Byte Offset: 087Ch:087Fh
Port 4 Ch 255 Byte Offset: 089Ch:089Fh
Port 5 Ch 255 Byte Offset: 08BCh:08BFh
Size: 24 bytes total, 4 bytes per port

Bits	Name	Access	Reset Value	Description
15:0	PIOBU*x*	R/W	0	Port Map *x* I/O Base Upper 16 bits. Contains a copy of the I/O Base Upper 16 Bits register for the link partner connected to port *x*. This field holds address bits [31:16].
31:16	PIOLU*x*	R/W	0	Port Map *x* I/O Limit Upper 16 bits. Contains a copy of the I/O Limit Upper 16 Bits register for the link partner connected to port *x*. This field holds address bits [31:16].

4.6.5 Software Generated Frame (SGF) Registers

These registers provide the ability to generate StarFabric special frames to be sent to a link partner.

4.6.5.1 Frame

The SG1010 SGF function can only support single-line special frames. Only three Dwords are required to specify a special frame. This frame is sent as is, starting with bit 0 of byte 0 at offset 8C0h. The SG1010 adds the appropriate link overhead fields.

Line 0 Dword 0 Ch 255 Byte Offset: 08C0h:08C3h
Line 0 Dword 1 Ch 255 Byte Offset: 08C4h:08C7h
Line 0 Dword 2 Ch 255 Byte Offset: 08C8h:08CBh
Size: 12 bytes

Bits	Name	Access	Reset Value	Description
31:0	FD*x*L*x*	R/W	0	Contains a Dword of frame data to be sent as a software generated frame.

4.6.5.2 SGF Control and Status

Ch 255 Byte Offset: 08E0h:08E3h
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
0	SEND_SGF	R/W	0	When set to a 1, the SG1010 sends the data in the Frame registers out of the appropriate link. The SG1010 clears this bit after the frame has been sent or discarded.
1	RES	R	0	Reserved.
2	SGF_DONE	R	0	The SG1010 sets this bit to a 1 when the frame is sent. This bit is also set if the SGF is discarded and not sent due to an error. The SG1010 clears this bit when software writes the SGF_Done bit in the Raw Event Status register (Section 4.6.3.4) with a 1 or when the SEND_SGF bit is set to send another SGF.
3	SGF_NSNT	R	0	The SG1010 sets this bit to a 1 when the SEND_SGF bit is written with a 1 but the selected output link is down. The SG1010 clears this bit when the SEND_SGF bit is set for the next SGF.
8:4	RES	R	0	Reserved.
11:9	SGFOUT	R/W	0	SGF Output Link. Selects the output link 0 through 5 for the SGF.
31:12	RES	R	0	Reserved.

4.6.6 Scratchpad Registers

These registers provide read/write state for software and are not associated with any other chip functionality.

Ch 255 Byte Offset: 08E8h:08EFh
Size: 8 bytes

The following description applies to each byte:

Bits	Name	Access	Reset Value	Description
7:0	SCRATCH	R/W	0	Byte-accessible scratchpad state, used for reading and writing application-specific software messages/information.

4.6.7 Register Path Protection

4.6.7.1 Register Path Protection Table

There are four entries in the Register Path Protection Table. Any incoming frame used for register access is compared against all enabled paths. If all enabled path comparisons fail, the register access is not performed and a Channel Path Protection event is signaled. Otherwise, the register access is allowed if the Range check is also successful.

Path 0 Ch 255 Byte Offset: 0910:0913
Path 1 Ch 255 Byte Offset: 0914:0917
Path 2 Ch 255 Byte Offset: 0918:091B
Path 3 Ch 255 Byte Offset: 091C:091F
Size: 4 bytes per entry, 16 bytes total

Bits	Name	Access	Reset Value	Description
20:0	RPP*x*	R/W	0	This field contains a transformed (inverted and reversed) path specification used for path protection checks against incoming frames accessing any SG1010 register from any space. The transformed path is actually the path from the SG1010 to the origin.
23:21	INPUT	R/W	0	This field contains the input port that accompanies the path specification. The input port used by the frame must match the input port specified in this field.
30:24	RES	R	0	Reserved
31	PPEN*x*	R/W	0	Enables path protection using the path in bits [20:0]. If 0, path protection is not performed against this register. If 1, path protection is performed against this register.

4.6.8 Chip Control Registers

4.6.8.1 Chip Control Status 0

Ch 255 Byte Offset: 092Ch:092Fh
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
7:0	REGWCC	R/W	24h (36)	Credit count for SG1010 register write accesses. The SG1010 maintains an internal credit-based system for register writes. After reset, the SG1010 loads this register with the default credit values. When read, this register returns the current write credit count. When written, the write data in this byte is added to the current credit count. The counter wraps if the addition causes an overflow. StarGen recommends that this register be written only through SROM preload; otherwise, the results may be unexpected.
15:8	REGDWC	R	24h (36)	Default credit count for SG1010 register write accesses. When read, returns the value assigned by hardware to the register credit counter.
16	VGA16E	R/W	0	VGA 16-bit Decode Enable. When 0, the VGA 16-bit Decode bit is read-only as 0 in the Bridge Control register, both in configuration space and in the Port Map Table. When a, the VGA 16-bit Decode bit is read/write and can be used to perform a 16-bit decode on VGA I/O transactions.
31:17	RES	R	0	Reserved

4.6.8.2 LED Control

Ch 255 Byte Offset: 0930h:0933h
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
3:0	LED0S	R/W	0	LED0[3:0] S/W state. When the LED0 Mode bit is 1 and a bit is: 0: The corresponding LED0_L[3:0] signal is driven high, turning off the LED. 1: The corresponding LED0_L[3:0] signal is driven low, turning on the LED.
7:4	LED1S	R/W	0	LED1[3:0] S/W state. When the LED1 Mode bit is 1 and a bit is: 0: The corresponding LED1_L[3:0] signal is driven high, turning off the LED. 1: The corresponding LED1_L[3:0] signal is driven low, turning on the LED.
11:8	LED2S	R/W	0	LED2[3:0] S/W state. When the LED2 Mode bit is 1 and a bit is: 0: The corresponding LED2_L[3:0] signal is driven high, turning off the LED. 1: The corresponding LED2_L[3:0] signal is driven low, turning on the LED.
15:12	LED3S	R/W	0	LED3[3:0] S/W state. When the LED3 Mode bit is 1 and a bit is: 0: The corresponding LED3_L[3:0] signal is driven high, turning off the LED. 1: The corresponding LED3_L[3:0] signal is driven low, turning on the LED.
19:16	LED4S	R/W	0	LED4[3:0] S/W state. When the LED4 Mode bit is 1 and a bit is: 0: The corresponding LED4_L[3:0] signal is driven high, turning off the LED. 1: The corresponding LED4_L[3:0] signal is driven low, turning on the LED.
23:20	LED5S	R/W	0	LED5[3:0] S/W state. When the LED5 Mode bit is 1 and a bit is: 0: The corresponding LED5_L[3:0] signal is driven high, turning off the LED. 1: The corresponding LED5_L[3:0] signal is driven low, turning on the LED.
24	LED0C	R/W	0	LED0 Mode. When a 0, the SG1010 drives the LED0_L[3:0] signal pins with either link state or differential pair receiver state. When a 1, the SG1010 drives the LED0_L[3:0] signal pins with the values contained in the LED0S field.

Bits	Name	Access	Reset Value	Description
25	LED1C	R/W	0	LED1 Mode. When a 0, the SG1010 drives the LED1_L[3:0] signal pins with either link state or differential pair receiver state. When a 1, the SG1010 drives the LED1_L[3:0] signal pins with the values contained in the LED1S field.
26	LED2C	R/W	0	LED2 Mode. When a 0, the SG1010 drives the LED2_L[3:0] signal pins with either link state or differential pair receiver state. When a 1, the SG1010 drives the LED2_L[3:0] signal pins with the values contained in the LED2S field.
27	LED3C	R/W	0	LED3 Mode. When a 0, the SG1010 drives the LED3_L[3:0] signal pins with either link state or differential pair receiver state. When a 1, the SG1010 drives the LED3_L[3:0] signal pins with the values contained in the LED3S field.
28	LED4C	R/W	0	LED4 Mode. When a 0, the SG1010 drives the LED4_L[3:0] signal pins with either link state or differential pair receiver state. When a 1, the SG1010 drives the LED4_L[3:0] signal pins with the values contained in the LED4S field.
29	LED5C	R/W	0	LED5 Mode. When a 0, the SG1010 drives the LED5_L[3:0] signal pins with either link state or differential pair receiver state. When a 1, the SG1010 drives the LED5_L[3:0] signal pins with the values contained in the LED5S field.
30	LEDHSS	R/W1TS	H/W	LED Hardware State Set. When read, this bit reflects the value of the LED H/W State. When a 0, differential pair receiver state is driven on all 24 LED signals if the corresponding LEDx Mode bit is 0. When a 1, link state is driven on LEDx_L[0] if the corresponding LEDx Mode bit is 0. The reset value is determined by the value of GPIO[0] during the deasserting edge of NRST_L. When software writes this bit with a 1, the LED Hardware State is set.
31	LEDHSC	R/W1TC	H/W	LED Hardware State Clear. When read, this bit reflects the value of the LED H/W State. When a 0, differential pair receiver state is driven on all 24 LED signals if the corresponding LEDx Mode bit is 0. When a 1, link state is driven on LEDx_L[0] if the corresponding LEDx Mode bit is 0. The reset value is determined by the value of GPIO[0] during the deasserting edge of NRST_L. When software writes this bit with a 1, the LED Hardware State is cleared.

4.6.9 Configuration Registers

These registers are dual-mapped configuration registers. These registers have the same access rules as at their configuration-mapped addresses. See Section 4.7 for a description of these registers

Ch 255 Byte Offset:	0A00h:0AFFh
Size:	256 bytes

4.6.10 Multicast Registers

Multicast Group 0 Ch 255 Byte Offset:	B00h:B03h
Multicast Group 1 Ch 255 Byte Offset:	B04h:B07h
Multicast Group 2 Ch 255 Byte Offset:	B08h:B0Bh
Multicast Group 3 Ch 255 Byte Offset:	B0Ch:B0Fh
Multicast Group 4 Ch 255 Byte Offset:	B10h:B13h
Multicast Group 5 Ch 255 Byte Offset:	B14h:B17h
Multicast Group 6 Ch 255 Byte Offset:	B18h:B1Bh
Multicast Group 7 Ch 255 Byte Offset:	B1Ch:B1Fh
Multicast Group 8 Ch 255 Byte Offset:	B20h:B23h
Multicast Group 9 Ch 255 Byte Offset:	B24h:B27h
Multicast Group 10 Ch 255 Byte Offset:	B28h:B2Bh
Multicast Group 11 Ch 255 Byte Offset:	B2Ch:B2Fh
Multicast Group 12 Ch 255 Byte Offset:	B30h:B33h
Multicast Group 13 Ch 255 Byte Offset:	B34h:B37h
Multicast Group 14 Ch 255 Byte Offset:	B38h:B3Bh
Multicast Group 15 Ch 255 Byte Offset:	B3Ch:B3Fh
Multicast Group 16 Ch 255 Byte Offset:	B40h:B43h
Multicast Group 17 Ch 255 Byte Offset:	B44h:B47h
Multicast Group 18 Ch 255 Byte Offset:	B48h:B4Bh
Multicast Group 19 Ch 255 Byte Offset:	B4Ch:B4Fh
Multicast Group 20 Ch 255 Byte Offset:	B50h:B53h
Multicast Group 21 Ch 255 Byte Offset:	B54h:B57h
Multicast Group 22 Ch 255 Byte Offset:	B58h:B5Bh
Multicast Group 23 Ch 255 Byte Offset:	B5Ch:B5Fh
Multicast Group 24 Ch 255 Byte Offset:	B60h:B63h
Multicast Group 25 Ch 255 Byte Offset:	B64h:B67h
Multicast Group 26 Ch 255 Byte Offset:	B68h:B6Bh
Multicast Group 27 Ch 255 Byte Offset:	B6Ch:B6Fh
Multicast Group 28 Ch 255 Byte Offset:	B70h:B73h
Multicast Group 29 Ch 255 Byte Offset:	B74h:B77h
Multicast Group 30 Ch 255 Byte Offset:	B78h:B7Bh

Multicast Group 31 Ch 255 Byte Offset: B7Ch:B7Fh
Size: 1 Dword per group, 32 Dwords total

Bits	Name	Access	Reset Value	Description
5:0	MG*x*PORT	R/W	0	These bits indicate the output ports selected for multicast group *x*. Bit 0 corresponds to Port0, bit 1 corresponds to Port1, and so on. When a bit is a 0, the corresponding output port is not a multicast group member. When a bit is a 1, the corresponding output port is a multicast group member.
15:6	RES	R	0	Reserved
21:16	MG*x*ACK	R	0	Multicast Acknowledge Bitmap. These bits indicate the write acknowledge frames that have been received in response to a multicast frame for Multicast Group *x*. Bit 0 corresponds to Port0, bit 1 correspond to Port1, etc. When a bit is a 1, a multicast acknowledge with the Final Multicast Ack bit set has been received on the corresponding port for this multicast group. When a bit is a 0, a multicast acknowledge has not been received on the corresponding port for this multicast group. If Nack Select is a 1, this field shows which ports have received multicast nack frames (Failure Type is not Normal) for this group. Hardware clears these bits when all expected multicast acknowledges are received or when a new multicast write with acknowledge for this group is forwarded. Additionally, software may clear these bits by writing a 1 to the Clear Multicast Ack bit.
24:22	RES	R	0	Reserved
25	NTS	R	1	Multicast Nack Tracking Support.Reads as 1 to indicate that the SG1010 uses multicast nack tracking.
26	NSL	R	0	Multicast Nack Select. If a 0, the Multicast Acknowledge Bitmap reflects which ports have received a multicast acknowledge with the Final Multicast Acknowledge bit set. If a 1, the Multicast Acknowledge Bitmap reflects which ports have received a multicast nack (Failure Type not Normal), regardless of the Final Multicast Ack bit.
27	CLRACK	R/W1TC	0	When written with a 1, clears all multicast acknowledge bits in this group. Writing a 0 has no effect. Reads of this bit always return 0.
30:28	MGINPUT	R/W	0	Returns the input port number for this multicast group.
31	RES	R	0	Reserved

4.6.11 Semaphore Registers

The SG1010 implements two semaphores. Each semaphore has eight possible operations. Each operation is performed by reading the respective operation register for that semaphore. Write data is discarded and has no effect.

If a semaphore Dword location is read and all of the byte enables for that location are turned off, the semaphore operation is not performed.

4.6.11.1 Semaphore N Clear

Semaphore 0 Ch 255 Byte Offset: 0C00h:0C03h
Semaphore 1 Ch 255 Byte Offset: 0C20h:0C23h
Size: 8 bytes total, 4 bytes per semaphore

Bits	Name	Access	Reset Value	Description
7:0	SEM*x*_CLR	RTC	0	When this byte is read, the SG1010 returns the value of Semaphore *N*, and then clears the value of Semaphore *N* to 0.
31:8	RES	R	0	The semaphore is eight bits wide and this field always reads as zero.

4.6.11.2 Semaphore N Set

Semaphore 0 Ch 255 Byte Offset: 0C04h:0C07h
Semaphore 1 Ch 255 Byte Offset: 0C24h:0C27h
Size: 8 bytes total, 4 bytes per semaphore

Bits	Name	Access	Reset Value	Description
7:0	SEM*x*_SET	RTS	0	When this byte is read, the SG1010 returns the value of Semaphore *N*, and then sets the value of Semaphore *N* to FFh.
31:8	RES	R	0	The semaphore is eight bits wide and this field always reads as zero.

4.6.11.3 Semaphore N Decrement

Semaphore 0 Ch 255 Byte Offset: 0C08h:0C0Bh
Semaphore 1 Ch 255 Byte Offset: 0C28h:0C2Bh
Size: 8 bytes total, 4 bytes per semaphore

Bits	Name	Access	Reset Value	Description
7:0	SEM*x*_DEC	RTDEC	0	When this byte is read, the SG1010 returns the value of Semaphore *N*, and then decrements the value of Semaphore *N*. The decrement operation sticks at 0.
31:8	RES	R	0	The semaphore is eight bits wide and this field always reads as zero.

4.6.11.4 Semaphore N Increment

Semaphore 0 Ch 255 Byte Offset: 0C0Ch:0C0Fh
Semaphore 1 Ch 255 Byte Offset: 0C2Ch:0C2Fh
Size: 8 bytes total, 4 bytes per semaphore

Bits	Name	Access	Reset Value	Description
7:0	SEM*x*_INC	RTINC	0	When this byte is read, the SG1010 returns the value of Semaphore *N*, and then increments the value of Semaphore *N*. The increment operation sticks at FFh.
31:8	RES	R	0	The semaphore is eight bits wide and this field always reads as zero.

4.6.11.5 Semaphore N Reserved 0

Semaphore 0 Ch 255 Byte Offset: 0C10h:0C13h
Semaphore 1 Ch 255 Byte Offset: 0C30h:0C33h
Size: 8 bytes total, 4 bytes per semaphore

Bits	Name	Access	Reset Value	Description
7:0	SEM*x*_RES0	R	0	When this byte is read, the SG1010 returns the value of Semaphore *N*. No operation is performed on the semphore.
31:8	RES	R	0	The semaphore is eight bits wide and this field always reads as zero.

4.6.11.6 Semaphore N Increment if 0

Semaphore 0 Ch 255 Byte Offset: 0C14h:0C17h
Semaphore 1 Ch 255 Byte Offset: 0C34h:0C37h
Size: 8 bytes total, 4 bytes per semaphore

Bits	Name	Access	Reset Value	Description
7:0	SEM*x*_INC0	RTINC	0	When this byte is read, the SG1010 returns the value of Semaphore *N*, and then increments the value of Semaphore *N* if the previous value was 0.
31:8	RES	R	0	The semaphore is eight bits wide and this field always reads as zero.

4.6.11.7 Semaphore N Reserved 1

Semaphore 0 Ch 255 Byte Offset: 0C18h:0C1Bh
Semaphore 1 Ch 255 Byte Offset: 0C38h:0C3Bh
Size: 8 bytes total, 4 bytes per semaphore

Bits	Name	Access	Reset Value	Description
7:0	SEM*x*_RES1	R	0	When this byte is read, the SG1010 returns the value of Semaphore *N*. No operation is performed on the semaphore.
31:8	RES	R	0	The semaphore is eight bits wide and this field always reads as zero.

4.6.11.8 Semaphore N Increment if not 0

Semaphore 0 Ch 255 Byte Offset: 0C1Ch:0C1Fh
Semaphore 1 Ch 255 Byte Offset: 0C3Ch:0C3Fh
Size: 8 bytes total, 4 bytes per semaphore

Bits	Name	Access	Reset Value	Description
7:0	SEM*x*_INCN0	RTINC	0	When this byte is read, the SG1010 returns the value of Semaphore *N*, and then increments the value of Semaphore *N* if the previous value was not 0. This operation sticks at FFh.
31:8	RES	R	0	The semaphore is eight bits wide and this field always reads as zero.

4.7 Configuration Registers

The configuration registers are accessible from a Type0 configuration frame received from the root port. Configuration space is not accessible from a non-root port. Configuration registers are also accessible in Channel 255 register space through path-routed reads and writes.

4.7.1 PCI Header Registers

These registers make up the first 16 bytes of the configuration space of every PCI device.

4.7.1.1 Vendor ID

PCI Configuration Byte Offset: 00h:01h
Size: 2 bytes

Bits	Name	Access	Reset Value	Description
15:0	VENDORID	R	9902h	Identifies the vendor of this device as StarGen. Returns 9902h when read.

4.7.1.2 Device ID

PCI Configuration Byte Offset: 02h:03h
Size: 2 bytes

Bits	Name	Access	Reset Value	Description
15:0	DEVID	R	03h	PCI device identification number. Returns 03h when read.

4.7.1.3 Command

PCI Configuration Byte Offset: 04h:05h
Size: 2 bytes

Bits	Name	Access	Reset Value	Description
0	PIO	R/W	0	I/O enable for downstream I/O frames. Used by the SG1010's upstream link partner (in its Port Map Table copy) to determine whether to forward an address-routed I/O frame to this device. When a 1, the I/O range is enabled for address decoding. When a 0, I/O frames are not forwarded to the SG1010.
1	PMEM	R/W	0	Memory enable for downstream address-routed memory frames. Used by the SG1010's upstream link partner (in its Port Map Table copy) to determine whether to forward a PCI memory frame to this device. When a 1, the memory range is enabled for address decoding. When a 0, memory frames are not forwarded to the SG1010.
2	PBM	R/W	0	Bus Master Enable. This bit controls target response to upstream I/O and memory transactions. When 0, inversely decoded upstream address-routed memory and I/O transactions result in a Address-routing Failure. When 1, upstream forwarding is enabled for I/O and memory frames.
3	PSC	R	0	Special Cycles. Reads as 0 to indicate that the SG1010 does not detect special cycles.
4	PMWI	R/W	0	MWI Enable. Because the SG1010 does not receive or generate MWI transactions, but can propagate transactions that result in MWIs, this bit is R/W but does not control any operations in the SG1010.
5	VGASN	R/W	0	VGA Snoop. Used by the SG1010's upstream link partner (in its Port Map Table copy) to determine whether to forward downstream transactions addressing the VGA 1/O locations 3C6h, 3C8h, or 3C9h. When written with 1, enables VGA decoding. Bits [15:0] are not decoded if the VGA 16-bit Decode Enable bit is 0; otherwise, bits [15:0] are decoded as 0.
6	PPER	R/W	0	Primary Parity Error Response. Read/write for compatibility, but does not control any functionality because parity errors are not reported or generated in the fabric.

Bits	Name	Access	Reset Value	Description
7	STEP	R	0	Stepping Control. Reads as 0. The SG1010 does not perform stepping.
8	SERREN	R/W	0	SERR# Enable. When a 0, the SG1010 cannot generate a SERR# event, and uses the standard interrupt EMU instead. When a 1, the SG1010 is enabled to generate a SERR# event.
9	FBBENP	R/W	0	Primary Fast Back-to-back Enable. Read/write for compatibility, but does not control any functionality because the fabric has no sense of fast-back-to-back protocol.
15:10	RES	R	0	Reserved

4.7.1.4 Status

PCI Configuration Byte Offset: 06h:07h
Size: 2 bytes

Bits	Name	Access	Reset Value	Description
3:0	RES	R	0	Reserved
4	BCAP	R	1	Capabilities List. Reads as a 1 to indicate that the SG1010 supports a capabilities list.
5	P66CAP	R	1	Primary 66 MHz Capable. The SG1010 has no sense of 33/66 PCI bus operation and this bit has no meaning, but reads as 1.
6	RES	R	0	Reserved
7	PFBBC	R	1	Primary Fast Back-to-Back Capable. This bit has no meaning for the SG1010, but reads as 1.
8	PMPERR	R	0	Primary Master Data Parity Error. This bit has no meaning for the SG1010 and is not set.
10:9	PDEVTIM	R	01b	Primary DEVSEL# Timing. This field has no meaning for the SG1010 because it has no PCI bus, but is set to indicate medium timing.
11	PSTA	R	0	Primary Signaled Target Abort. The SG1010 does not generates a response frame with target abort. This bit is not set.
12	PRTA	R	0	Primary Received Target Abort. This bit has no meaning for the SG1010 because it does not generate any frames that would cause it to receive a target abort response.
13	PRMA	R/W1TC	0	Primary Received Master Abort. This bit has no meaning for the SG1010 because upstream frames that are inversely decoded are automatically forwarded. This bit is never set.
14	PSSERR	R	0	Signaled System Error. This bit has no meaning for the SG1010 and is not set.
15	PDPE	R	0	Primary Detected Parity Error. This bit has no meaning for the SG1010 and is not set.

4.7.1.5 Revision ID

PCI Configuration Byte Offset: 08h
Size: 1 byte

Bits	Name	Access	Reset Value	Description
7:0	REVID	R		Identifies the revision of this device. Initial revision is 0.

4.7.1.6 Class Code

PCI Configuration Byte Offset: 09h:0Bh
Size: 3 bytes

Bits	Name	Access	Reset Value	Description
7:0	PROGIF	R	0	Identifies the device's programming interface. Reads as 0 to indicate that there is no programming interface.
15:8	SUBCL	R	04h	Identifies the device's subclass. Reads as 04h to indicate that this is a PCI-to-PCI bridge device.
23:16	BASECL	R	06h	Identifies the device's base class. Reads as 06h to indicate that this is a bridge device.

4.7.1.7 Cache Line Size:

PCI Configuration Byte Offset: 0Ch
Size: 1 byte

Bits	Name	Access	Reset Value	Description
7:0	CLS	R/W	0	Indicates the cache line size in Dwords. Not used by the SG1010, but is R/W for software compatibility.

4.7.1.8 Master Latency Timer

PCI Configuration Byte Offset: 0Dh
Size: 1 byte

Bits	Name	Access	Reset Value	Description
7:0	MLT	R/W	0	PCI master latency timer. Not used by the SG1010, but is R/W for software compatibility.

4.7.1.9 Header Type

PCI Configuration Byte Offset: 0Eh
Size: 1 byte

Bits	Name	Access	Reset Value	Description
7:0	HDRTYPE	R	01h	Indicates a single function device with a PCI-to-PCI Bridge header.

4.7.2 PCI Address and Secondary Bus Registers

4.7.2.1 Primary Bus Number

PCI Configuration Byte Offset: 18h
Size: 1 byte

Bits	Name	Access	Reset Value	Description
7:0	PBUS	R/W	0h	Identifies the bus number of the primary PCI bus. Used for upstream decoding of Type1 configuration write transactions to determine whether they are to be translated to special cycles (which signals an event within the StarFabric).

4.7.2.2 Secondary Bus Number

PCI Configuration Byte Offset: 19h
Size: 1 byte

Bits	Name	Access	Reset Value	Description
7:0	SBUS	R/W	0h	Identifies the bus number of the secondary PCI bus. Used to define the base bus number of a window used for bus number decoding of Type1 configuration transactions. Additionally, used to decode downstream transactions to determine whether they are translated to Type0. The upstream link partner uses a copy of this value to decode downstream configuration addresses. The SG1010 uses the register for inverse (upstream) decoding and for downstream Type1 to Type 0 conversions.

4.7.2.3 Subordinate Bus Number

PCI Configuration Byte Offset: 1Ah
Size: 1 byte

Bits	Name	Access	Reset Value	Description
7:0	SUBBUS	R/W	0h	Identifies the bus number of the subordinate PCI bus. Used to define the upper limit (inclusive) of a window for decoding bus numbers of Type1 configuration transactions. The upstream link partner uses a copy of this value in its Port Map Table to decode downstream configuration addresses. The SG1010 uses the register for inverse (upstream) decoding.

4.7.2.4 Secondary Latency Timer

PCI Configuration Byte Offset: 1Bh
Size: 1 byte

Bits	Name	Access	Reset Value	Description
7:0	SMLT	R/W	0h	Secondary master latency timer. This field has no meaning for the SG1010, but is R/W for software compatibility.

4.7.2.5 I/O Base

PCI Configuration Byte Offset: 1Ch
Size: 1 byte

Bits	Name	Access	Reset Value	Description
0	IO32	R	1	Reads as 1 to indicate that the SG1010 supports a 32-bit I/O address range as a PCI-to-PCI bridge.
3:1	RES	R	0	Reserved.
7:4	IOBASE	R/W	0	Defines address bits [15:12] of the low end of the downstream I/O address range for the SG1010. Bits [11:0] are assumed to be 0, giving a minimum size and alignment granularity of 4KB. Used for upstream address decoding; the link partner uses a copy in its Port map Table for downstream decoding.

4.7.2.6 I/O Limit

PCI Configuration Byte Offset: 1Dh
Size: 1 byte

Bits	Name	Access	Reset Value	Description
0	IO32	R	1	Reads as 1 to indicate that the SG1010 supports a 32-bit I/O address range as a PCI-to-PCI bridge.
3:1	RES	R	0	Reserved.
7:4	IOLIMIT	R/W	0	Defines address bits [15:12] of the high end of the down-stream I/O address range. Bits [11:0] are assumed to be 0, giving a minimum size and alignment granularity of 4KB. Used for upstream address decoding; the link partner uses a copy in its Port map Table for downstream decoding.

4.7.2.7 Secondary Status

PCI Configuration Byte Offset: 1Eh:1Fh
Size: 2 bytes

Bits	Name	Access	Reset Value	Description
4:0	RES	R	0	Reserved
5	S66CAP	R	1	Secondary 66 MHz Capable. The SG1010 has no sense of 33/66 PCI bus operation and this bit has no meaning, but reads as 1.
6	RES	R	0	Reserved
7	SFBBC	R	1	Secondary Fast Back-to-Back Capable. This bit has no meaning for the SG1010, but reads as 1.
8	SMPERR	R	0	Secondary Master Data Parity Error. This bit has no meaning for the SG1010 and is not set.
10:9	SDEVTIM	R	01b	Secondary DEVSEL# Timing. This field has no meaning for the SG1010, because it has no PCI bus, but is set to indicate medium timing.
11	SSTA	R	0	Secondary Signaled Target Abort. The SG1010 never generates a response frame with target abort. This bit is not set.
12	SRTA	R	0	Secondary Received Target Abort. This bit has no meaning for the SG1010 because does not generate any frames that would cause it to receive a target abort response
13	SRMA	R/W1TC	0	Secondary Received Master Abort. Set by the SG1010 when it detects an Address-Routing Failure in response to a downstream frame.
14	SRSERR	R	0	Received System Error. This bit has no meaning for the SG1010 and is not set.
15	SDPE	R	0	Secondary Detected Parity Error. This bit has no meaning for the SG1010 and is not set.

4.7.2.8 Memory Base

PCI Configuration Byte Offset: 20h:21h
Size: 2 bytes

Bits	Name	Access	Reset Value	Description
3:0	RES	R	0	Reserved.
15:4	MEMBASE	R/W	0	Defines address bits [31:20] of the low end of the downstream memory address range. Bits [19:0] are assumed to be 0, giving a minimum size and alignment granularity of 1MB. Used for upstream address decoding; the link partner uses a copy in its Port map Table for downstream decoding.

4.7.2.9 Memory Limit

PCI Configuration Byte Offset: 22h:23h
Size: 2 bytes

Bits	Name	Access	Reset Value	Description
3:0	RES	R	0	Reserved.
15:4	MEMLIMIT	R/W	0	Defines address bits [31:20] of the high end of the downstream memory address range. Bits [19:0] are assumed to be 0, giving a minimum size and alignment granularity of 1MB. Used for upstream address decoding; the link partner uses a copy in its Port map Table for downstream decoding.

4.7.2.10 Prefetchable Memory Base

PCI Configuration Byte Offset: 24h:25h
Size: 2 bytes

Bits	Name	Access	Reset Value	Description
0	MEM64	R	1	Reads as 1 to indicate that the SG1010 supports a 64-bit prefetchable memory address range as a PCI-to-PCI bridge.
3:1	RES	R	0	Reserved.
15:4	PFMBASE	R/W	0	Defines address bits [31:20] of the low end of the downstream prefetchable memory address range. Bits [19:0] are assumed to be 0, giving a minimum size and alignment granularity of 1MB. Used for upstream address decoding; the link partner uses a copy in its Port map Table for downstream decoding.

4.7.2.11 Prefetchable Memory Limit

PCI Configuration Byte Offset: 26h:27h
Size: 2 bytes

Bits	Name	Access	Reset Value	Description
0	MEM64	R	1	Reads as 1 to indicate that the SG1010 supports a 64-bit prefetchable memory address range as a PCI-to-PCI bridge.
3:1	RES	R	0	Reserved.
15:4	PFMLIMIT	R/W	0	Defines address bits [31:20] of the high end of the downstream prefetchable memory address range. Bits [19:0] of the address are assumed to be 0, giving a minimum size and alignment granularity of 1 Mbyte. Used for upstream address decoding; the link partner uses a copy in its Port map Table for downstream decoding.

4.7.2.12 Prefetchable Memory Base Upper 32 Bits

PCI Configuration Byte Offset: 28h:2Bh
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
31:0	PFMB64	R/W	0	Defines address bits [63:32] of the low end of the downstream prefetchable memory address range. Used for upstream address decoding; the link partner uses a copy in its Port map Table for downstream decoding.

4.7.2.13 Prefetchable Memory Limit Upper 32 Bits

PCI Configuration Byte Offset: 2Ch:2Fh
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
31:0	PFML64	R/W	0	Defines address bits [63:32] of the high end of the downstream prefetchable memory address range. Used for upstream address decoding; the link partner uses a copy in its Port map Table for downstream decoding.

4.7.2.14 I/O Base Upper 16 Bits

PCI Configuration Byte Offset: 30h:31h
Size: 2 bytes

Bits	Name	Access	Reset Value	Description
15:0	IOB32	R/W	0	Defines address bits [31:16] of the low end of the downstream I/O address range. Used for upstream address decoding; the link partner uses a copy in its Port map Table for downstream decoding.

4.7.2.15 I/O Limit Upper 16 Bits

PCI Configuration Byte Offset: 32h:33h
Size: 2 bytes

Bits	Name	Access	Reset Value	Description
15:0	IOL32	R/W	0	Defines address bits [31:16] of the high end of the downstream I/O address range. Used for upstream address decoding; the link partner uses a copy in its Port map Table for downstream decoding.

4.7.3 Other PCI Registers

4.7.3.1 ECP

PCI Configuration Byte Offset: 34h
Size: 1 byte

Bits	Name	Access	Reset Value	Description
7:0	ECP	R	70h	Returns the configuration offset of the first ECP function, which is the Power Management ECP function at offset 70h.

4.7.3.2 Interrupt Line

PCI Configuration Byte Offset: 3Ch
Size: 1 byte

Bits	Name	Access	Reset Value	Description
7:0	INTLINE	R/W	0	Interrupt line register. The SG1010 does not have an interrupt pin. Initialization code should program this field to be FFh.

4.7.3.3 Interrupt Pin

PCI Configuration Byte Offset: 3Dh
Size: 1 byte

Bits	Name	Access	Reset Value	Description
7:0	INTPIN	R	0	Reads as 0 to indicate that the Bridge does not support an interrupt pin.

4.7.3.4 Bridge Control

PCI Configuration Byte Offset: 3Eh:3Fh
Size: 2 bytes

Bits	Name	Access	Reset Value	Description
0	SPER	R/W	0	Secondary Parity Error Response. This bit has no meaning for the SG1010, but is R/W for compatibility.
1	SERRFE	R/W`	0	SERR# Forward Enable. This bit has no meaning for the SG1010, but is R/W for compatibility.
2	ISAENA	R/W	0	ISA Enable. When a 1, the SG1010 performs ISA address filtering in the I/O Base and Limit Range. When a 0, the SG1010 does not perform ISA address filtering.
3	VGAENA	R/W	0	VGA Enable. When a 1, the SG1010 decodes VGA addresses for downstream forwarding, and inversely decodes them for upstream forwarding. When a 0, the SG1010 does not perform VGA address decoding. Used by the SG1010 for upstream decoding; its link partner uses a copy of the bit in its Port Map Table for downstream decoding.
4	VGA16	R/W R	0	VGA 16-bit Decode. Controls aliasing for VGA transactions. When a 1, a full 16-bit address decode is performed and bits [15:10] must be 0. When a 0, a 10-bit address decode is performed and address bits [15:10] may be any value to forward VGA transactions downstream. Used by the SG1010 for upstream decoding; its link partner uses a copy of the bit in its Port Map Table for downstream decoding. The use of this bit in both this register and the Port Map Tables is controlled by the VGA 16-bit Decode Enable bit in the Chip Control and Status 0 register in Channel 255 space. If the VGA 16-bit Decode Enable is 0, this bit is read-only as 0. If the VGA 16-bit Decode Enable is 0, this bit is read-write.
5	MAMODE	R/W	0	Master Abort Mode. This function is performed at edge nodes. This bit has no meaning for the SG1010, but is R/W for compatibility.

6	SECRST	R/W	0	Secondary Reset. When written with 1, a maskable reset is propagated to its link partners, but the SG1010 itself is not reset (except for Port Map Table values, which are reset).
7	FBBENS	R/W	0	Secondary Fast Back-to-back Enable. This bit has no meaning for the SG1010, but is R/W for compatibility.
8	PDISC	R/W	0	Primary Discard Timer. This bit has no meaning for the SG1010, but is R/W for compatibility.
9	SDISC	R/W	0	Secondary Discard Timer. This bit has no meaning for the SG1010, but is R/W for compatibility.
10	DISCSTAT	R	0	Discard Timer Status. This bit has no meaning for the SG1010 and is not set.
11	DISCSERR	R/W	0	Discard Timer SERR# Enable. This bit has no meaning for the SG1010, but is R/W for compatibility.
15:12	RES	R	0	Reserved.

4.7.4 ROM Control Registers

4.7.4.1 RROM Address

PCI Configuration Byte Offset: 48h:4Bh
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
15:0	ROMADDR	R/W	0	Specifies the byte address offset for an SROM access. Because only Dword-aligned accesses are performed by the ROM interface, the two LSBs are considered to be zeros.
30:16	RES	R	0	Reserved.
31	SBF	R/W	0	Start/Busy Flag. Initiates the ROM transaction. When written with a 0, the SG1010 reads four bytes from the SROM address location specified in bits [15:0]. When the read is complete and the data is available in the ROM Data register, the SG1010 sets this bit to a 1. When written with a 1, the SG1010 writes four bytes to the SROM address specified in bits [15:0]. When the write is complete, the SG1010 sets this bit to a 0.

4.7.4.2 ROM Data

PCI Configuration Byte Offset: 4Ch:4Fh
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
31:0	ROMDATA	R/W	0	Contains 32-bits of data that is either written to or has been read from the SROM. Read data is valid after the SG1010 clears the Start/Busy flag (SBF).

4.7.5 General Purpose I/O (GPIO) Registers

4.7.5.1 Set GPIO Data

PCI Configuration Byte Offset: 54h
Size: 1 byte

Bits	Name	Access	Reset Value	Description
7:0	SGPDATA	R/W1TS	0	GPIO Data Set operation. When read, returns the current value of the GPIO signal pins (which is not necessarily the same values as the GPIO Data register, depending on the signal pin direction). When written with 1, sets the corresponding bit in the GPIO Data register. If the same bit in the GPIO Direction register is 1, then the corresponding GPIO signal pin is driven high and stays high until cleared or until the direction is changed. Writing this register with a 0 has no effect.

4.7.5.2 Set GPIO Direction

PCI Configuration Byte Offset: 55h
Size: 1 byte

Bits	Name	Access	Reset Value	Description
7:0	CGPDIR	R/W1TS	0	GPIO Direction Set operation. When read, returns the current direction of each GPIO signal pin. A 0 indicates that the pin is input-only and a 1 indicates that the pin is bidirectional. When any bit is written with 1, the corresponding GPIO Direction register bit is set and the GPIO pin is configured as a bidirectional pin. The current value of the same bit in the GPIO Data register is driven onto the pin. Writing 0 to this register has no effect.

4.7.5.3 Clear GPIO Data

PCI Configuration Byte Offset: 56h
Size: 1 byte

Bits	Name	Access	Reset Value	Description
7:0	CGPDATA	R/W1TC	0	GPIO Data Clear operation. When read, returns the current value of the GPIO Data register (which is not necessarily the same value as the GPIO signal pins, depending on their direction). When written with 1, clears the corresponding bit in the GPIO Data register. If the same bit in the GPIO Direction register is 1, then the corresponding GPIO signal pin is driven low and stays low until set or until the direction is changed. Writing 0 to this register has no effect.

4.7.5.4 Clear GPIO Direction

PCI Configuration Byte Offset: 57h
Size: 1 byte

Bits	Name	Access	Reset Value	Description
7:0	SGPDIR	R/W1TC	0	GPIO Direction Clear operation. When read, returns the current direction of each GPIO signal pin, where 0 indicates that the pin is input-only and 1 indicates that the pin is bidirectional. When any bit is written with 1, the corresponding GPIO Direction bit is cleared and the GPIO pin is configured as an input-only pin. Writing 0 to this register has no effect.

4.7.6 CSR Index Registers

The CSR index registers allow indirect access to all CSR registers using Type0 configuration frames.

4.7.6.1 Register Index *x*

Register Index 0 PCI Configuration Byte Offset: 60h:63h
Register Index 1 PCI Configuration Byte Offset: 68h:6Bh
Size: 8 bytes

Bits	Name	Access	Reset Value	Description
11:0	RINDEX*x*	R/W	0	Software writes this register *x* with the Channel 255 byte offset of the register that it desires to access. Must be valid before the corresponding Register Data *x* register is accessed.
31:12	RES	R	0	Reserved

4.7.6.2 Register Data *x*

Register Data 0 PCI Configuration Byte Offset: 64h:67h
Register Data 1 PCI Configuration Byte Offset: 6Ch:6Fh
Size: 8 bytes

Bits	Name	Access	Reset Value	Description
31:0	RDATAx	R/W	0	Type0 configuration reads or writes to this register cause a read or write access to the CSR whose offset is contained in the corresponding Register Index *x* register. For both reads and writes, byte enables are used to enable byte access granularity. If this register is written with a Channel 255 operation through the Channel 255 dual-map location, then this register is treated as a reserved register – write data is discarded and zero is returned for reads.

4.7.7 Power Management ECP Registers

4.7.7.1 Power Management ECP ID

PCI Configuration Byte Offset: 70h
Size: 1 byte

Bits	Name	Access	Reset Value	Description
7:0	PMECPID	R	01h	Extended Capabilities ID for the power management function. Must read as 01h.

4.7.7.2 Power Management Next Pointer

PCI Configuration Byte Offset: 71h
Size: 1 byte

Bits	Name	Access	Reset Value	Description
7:0	PMNXTPTR	R	78h	Contains the configuration offset of the next ECP function, which is the Slot Numbering ECP function (Section 4.7.8) at offset 78h.

4.7.7.3 Power Management Capabilities

PCI Configuration Byte Offset: 72h:73h
Size: 2 bytes

Bits	Name	Access	Reset Value	Description
2:0	PMVER	R	2h	Reads as 010b to indicate that this device complies with the *PCI Power Management Specification, Version 1.1.*
3	PMECLK	R	0	PCI clock required for PME#. This function does not support PME#. This bit reads as 0.
4	RES	R	0	Reserved.
5	DSI	R	0	Device Specific Initialization required. Reads as 0 to indicate that device specific initialization is not required following a transition to D0.
8:6	AUXCURR	R	0	3.3V auxiliary current requirements. This function does not support PME#. These bits read as 0.
9	D1_SUPP	R	0	D1 Support. Reads as 0 to indicate that this function does not support the D1 power state.
10	D2_SUPP	R	0	D2 Support. Reads as 0 to indicate that this function does not support the D2 power state.
15:11	PME_SUPP	R	0	PME# Support. Reads as 0 to indicate that PME# assertion is not supported in any power state.

4.7.7.4 Power Management Control and Status

PCI Configuration Byte Offset: 74h:75h
Size: 2 bytes

Bits	Name	Access	Reset Value	Description
1:0	PWRSTATE	R/W	0	Indicates the current power state of this function and controls transitions to a new power state. Only D0 (00b) and D3 (11b) are supported, and only writes to these states are allowed. The SG1010 will remain in the current power state if a write to either D1 or D2 is attempted. A write of 00b when the current state is 11b results in a propagated maskable reset followed by a chip reset.
7:2	RES	R	0	Reserved.
8	PME_EN	R	0	PME# Enable. Reads as 0. PME# is not supported by this function.
12:9	DATA_SEL	R	0	Data Select Index. Reads as 0. The Data register is not supported.
14:13	SCALE	R	0	Data Scale. Reads as 0. The Data register is not supported.
15	PME_ST	R	0	PME# Status. Reads as 0. This function does not support PME#.

4.7.7.5 Power Management PCI-to-PCI Bridge Support

PCI Configuration Byte Offset: 76h
Size: 1 byte

Bits	Name	Access	Reset Value	Description
5:0	RES	R	0	Reserved.
6	B2_B3	R	0	B2/B3 support. Reads as 0, to indicate that this power management function does not affect PCI clocks.
7	BPCC_EN	R	0	Clock control enable. Reads as 0, to indicate that this power management function does not affect PCI clocks.

4.7.7.6 Power Management Data Register

PCI Configuration Byte Offset: 77h
Size: 1 byte

Bits	Name	Access	Reset Value	Description
7:0	PMDATA	R	0	Data register. Not supported. Reads as 0.

4.7.8 Slot Numbering ECP Registers

4.7.8.1 Slot Numbering ECP ID

PCI Configuration Byte Offset: 78h
Size: 1 byte

Bits	Name	Access	Reset Value	Description
7:0	SNECPID	R	04h	Extended Capabilities ID for the slot numbering function. Must read as 04h.

4.7.8.2 Slot Numbering Next Pointer

PCI Configuration Byte Offset: 79h
Size: 1 byte

Bits	Name	Access	Reset Value	Description
7:0	SNNEXTPTR	R	7Ch	Contains the configuration offset of the next ECP function, which is VPD (Section 4.7.9) at offset 7Ch.

4.7.8.3 Slot Numbering Expansion Slot

PCI Configuration Byte Offset: 7Ah
Size: 1 byte

Bits	Name	Access	Reset Value	Description
4:0	EXPSLTS	R	0	Indicates the number of expansion slots located on the SG1010's secondary bus. Loadable by SROM preload.
5	CHASSIS	R	0	When set to 1 through SROM preload, indicates that this is the first bridge in an expansion chassis.
7:6	RES	R	0	Reserved. Serial preload data should contain 0s (not hardware restricted).

4.7.8.4 Slot Numbering Chassis Number

PCI Configuration Byte Offset: 7Bh
Size: 1 byte

Bits	Name	Access	Reset Value	Description
7:0	CHASNUM	R/W	0	Contains the chassis number for the secondary bus.

4.7.9 Vital Product Data (VPD) ECP Registers

4.7.9.1 VPD ECP ID

PCI Configuration Byte Offset: 7Ch
Size: 1 byte

Bits	Name	Access	Reset Value	Description
7:0	VPDECPID	R	3h	Extended Capabilities ID for the VPD function. Must read as 3h.

4.7.9.2 VPD Next Pointer

PCI Configuration Byte Offset: 7Dh
Size: 1 byte

Bits	Name	Access	Reset Value	Description
7:0	VPDNXTPTR	R	0	Contains the configuration offset of the next ECP function. VPD is the last function in the list, and this register returns 0 when read.

4.7.9.3 VPD Address

PCI Configuration Byte Offset: 7Eh:7Fh
Size: 2 bytes

Bits	Name	Access	Reset Value	Description
7:0	VPDADDR	R/W	0	Dword aligned byte-address offset of the 256-byte VPD address region to be accessed (bits [1:0] are ignored and assumed to be 00b).
14:8	RES	R	0	Reserved
15	VPDFLAG	R/W	0	VPD operation/status bit. When written with a 0, the SG1010 reads 4 bytes of data from the VPD address location written to bits [7:0]. When the read is complete and data is available, the SG1010 sets this bit to 1. When written with a 1, the SG1010 writes 4 bytes of data contained in the VPD data register to the VPD address location written to bits [7:0]. When the write is complete, the SG1010 sets this bit to 0.

4.7.9.4 VPD Data

PCI Configuration Byte Offset: 80h:83h
Size: 4 bytes

Bits	Name	Access	Reset Value	Description
31:0	VPDDATA	R/W	0	Contains 4 bytes of VPD data. On a VPD read, the SG1010 will place read data in this register. Before a VPD write operation, software must write the data to be stored should be written to this register.

STARGEN 5

Signal Pin Descriptions

The tables in this chapter define the signals on the the SG1010's pins. Table 5–1 defines the abbreviations used in the Type columns of the signal pin tables.

Table 5–1 Signal Type Definitions

Signal Type	Description
I	Input only
O	Output only
TS	Tristate bidirectional
STS	Sustained tristate bidirectional
OD	Open drain output only
BOD	Bidirectional open drain
LO	LVDS output
LI	LVDS input
P	Power

Table 5–2 GPIO Signal Pins

Signal Name	Width	Type	Description
GPIO[7:1]	7	TS	General purpose I/O pins. These pins are controlled by the SG1010 GPIO register function.
GPIO[0]/ LEDHM	1	TS	Dual mode pin. General purpose I/O pin. This pin is controlled by the SG1010 GPIO register function. LEDHM is sampled on the deasserting edge of NRST_L to determine the hardware LED mode as described in Section 3.11.3.

Table 5–3 ROM Interface Signal Pins

Signal Name	Width	Type	Description
SR_DO	1	I	SROM data out. Receives read data from the SROM.
SR_DI	1	O	SROM data in. The SG1010 drives the SROM command, address, and write data on this signal.

Table 5–3 ROM Interface Signal Pins

Signal Name	Width	Type	Description
SR_CK	1	O	SROM clock input.
SR_CS_L	1	O	SROM chip select. The SG1010 drives this signal low at the beginning of an SROM operation; high at the end of the operation.

Table 5–4 Global PLL and Reset Signal Pins

Signal Name	Width	Type	Description
VDDG	1	P	VDD for 112.5MHz phase-locked loop (PLL)
VSSG	1	P	VSS for 112.5MHz PLL
TSTCLKG	1	I	Reference clock used to bypass the 112.5 MHz PLL supplying the global clock. The 112.5MHz PLL uses REFCLKL as its input clock. Bypass mode is selected through TESTMODE pins.
PLLCLKGO	1	O	112.5MHz PLL output. The SG1010 has to be put into testmode 5 in order to observe the 112.5 Mhz PLL on this pin.
NRST_L	1	I	Chip reset pin.

Table 5–5 Link Interface Signal Pins

Signal Name	Width	Type	Description
TX0P[3:0]	4	LO	Link 0 LVDS transmit positive
TX0N[3:0]	4	LO	Link 0 LVDS transmit negative
TX1P[3:0]	4	LO	Link 1 LVDS transmit positive
TX1N[3:0]	4	LO	Link 1 LVDS transmit negative
TX2P[3:0]	4	LO	Link 2 LVDS transmit positive
TX2N[3:0]	4	LO	Link 2 LVDS transmit negative
TX3P[3:0]	4	LO	Link 3 LVDS transmit positive
TX3N[3:0]	4	LO	Link 3 LVDS transmit negative
TX4P[3:0]	4	LO	Link 4 LVDS transmit positive
TX4N[3:0]	4	LO	Link 4 LVDS transmit negative
TX5P[3:0]	4	LO	Link 5 LVDS transmit positive
TX5N[3:0]	4	LO	Link 5 LVDS transmit negative
RX0P[3:0]	4	LI	Link 0 LVDS receive positive
RX0N[3:0]	4	LI	Link 0 LVDS receive negative
RX1P[3:0]	4	LI	Link 1 LVDS receive positive
RX1N[3:0]	4	LI	Link 1 LVDS receive negative
RX2P[3:0]	4	LI	Link 2 LVDS receive positive
RX2N[3:0]	4	LI	Link 2 LVDS receive negative
RX3P[3:0]	4	LI	Link 3 LVDS receive positive
RX3N[3:0]	4	LI	Link 3 LVDS receive negative

Table 5–5 Link Interface Signal Pins *(Continued)*

Signal Name	Width	Type	Description
RX4P[3:0]	4	LI	Link 4 LVDS receive positive
RX4N[3:0]	4	LI	Link 4 LVDS receive negative
RX5P[3:0]	4	LI	Link 5 LVDS receive positive
RX5N[3:0]	4	LI	Link 5 LVDS receive negative
REFCLKL	1	I	Reference clock for CDR PLL and Global PLL. Frequency is 62.208 Mhz
CTAP0[3:0]	4	I	Link 0 LVDS center taps for external reference voltages
CTAP1[3:0]	4	I	Link 1 LVDS center taps for external reference voltages
CTAP2[3:0]	4	I	Link 2 LVDS center taps for external reference voltages
CTAP3[3:0]	4	I	Link 3 LVDS center taps for external reference voltages
CTAP4[3:0]	4	I	Link 4 LVDS center taps for external reference voltages
CTAP5[3:0]	4	I	Link 5 LVDS center taps for external reference voltages
RESLO	1	I	LVDS 100Ω reference low – connects to RESHI through 100Ω 1% resistor.
RESHI	1	I	LVDS 100Ω reference high – connects to RESLO through 100Ω 1% resistor.
REF14	1	I	LVDS 1.4V reference
REF10	1	I	LVDS 1.0V reference
VDDA	4	P	Analog VDD
VSSA	4	P	Analog VSS

Table 5–6 Link Interface Test Pins

Signal Name	Width	Type	Description
TSTCLKL	1	I	Test clock during CDR PLL bypass
BYPASSL	1	I	Active-high bypass enable for CDR PLLs
RESETTX	1	I	Tx clock divide reset for CDR PLL bypass
TESTRST	1	I	CDR Test reset
TSTSHFTLD	1	I	CDR test mode shift enable
ECSEL	1	I	Manual CDR phase shift
ETOGGLE	1	I	CDR clock phase change
EXDNUP	1	I	CDR clock phase direction (1 = up, 0 = down)
TSTPHASE	1	I	Bypass phase control
LOOPBKEN	1	I	Loop back enable

Table 5–7 Test Signal Pins

Signal Name	Width	Type	Description
TESTMODE[4:0]	5	I	Defines the following test modes in the SG1010: 0h Functional/JTAG 1h Functional/PLL bypass 2h SCAN 3h IDDQ 4h Reserved 5h Global PLL test 6h - 9hReserved Ah CDR Test 1 Bh CDR Test 2 Ch Private use Dh Private use Eh Private use Fh Private use 10h - 1Fh Reserved
DIAG_EN	1	I	Diagnostic Port Enable.
LED0_L[3:0]/TESTMUX[3:0] LED1_L[3:0]/TESTMUX[7:4] LED2_L[1:0]/TESTMUX[9:8]	10	TS	Test mode output port/Transmit state LEDs for links 0, 1, 2. Also used for ASIC core/CDR interface test.
LED2_L[3:2]	2	TS	Remaining Link 2 state LEDs
LED3_L[3:0]/AD[3:0]	4	TS	Link 3 state LEDs/Diagnostic port AD[3:0] signals.
LED4_L[3:0]/AD[7:4]/SCAN_IN[3:0]	4	TS	Link 4 state LEDs/Diagnostic port AD[7:4] signals/Scan input signals.
LED5_L[3]/RDY_L/SCAN_IN[4] LED5_L[2]/AS_L/SCAN_IN[5] LED5_L[1]/WR_L/SCAN_IN[6] LED5_L[0]/RD_L/SCAN_IN[7]	4	TS	Link 5 state LEDs/Diagnostic port control signals: target ready, address strobe, write strobe, and read strobe/Scan input signals.
TCK	1	I	JTAG clock
TDI	1	I	JTAG data in
TDO	1	O	JTAG data out
TMS	1	I	JTAG mode select
TRST_L	1	I	JTAG reset
SCAN_ENA	1	I	Scan enable input
SCAN_OUT[7:0]	8	O	Scan chain outputs

STARGEN 6

Signal Pin List

The SG1010 is packaged in a 272-pin Plastic Ball Grid Array (PBGA) package. The pinout of the SG2010 is shown in Table 6–1.

Table 6–1 SG1010 PBGA Ball Assignment

INDEX	BALL	SIGNAL NAME	TYPE	INDEX	BALL	SIGNAL NAME	TYPE
1	B1	TX1P[0]	O	137	K18	RX0N[3]	I
2	C2	TX1N[0]	O	138	J19	CTAP0[3]	I
3	D2	TX1P[1]	O	139	J18	RX1P[0]	I
4	D3	TX1N[1]	O	140	J17	RX1N[0]	I
5	E4	TX1P[2]	O	141	H20	CTAP1[0]	I
6	E3	TX1N[2]	O	142	H19	RX1P[1]	I
7	D1	TX1P[3]	O	143	H18	RX1N[1]	I
8	C1	TX1N[3]	O	144	G20	CTAP1[1]	I
9	E2	TX0P[0]	O	145	G18	RX1P[2]	I
10	E1	TX0N[0]	O	146	F19	RX1N[2]	I
11	F3	TX0P[1]	O	147	E20	CTAP1[2]	I
12	G4	TX0N[1]	O	148	G17	RX1P[3]	I
13	F2	TX0P[2]	O	149	F18	RX1N[3]	I
14	F1	TX0N[2]	O	150	E18	CTAP1[3]	I
15	G3	TX0P[3]	O	151	D19	RX2P[0]	I
16	G2	TX0N[3]	O	152	C20	RX2N[0]	I
17	G1	TX5P[0]	O	153	E17	CTAP2[0]	I
18	H3	TX5N[0]	O	154	D18	RX2P[1]	I
19	H2	TX5P[1]	O	155	C19	RX2N[1]	I
20	H1	TX5N[1]	O	156	B20	CTAP2[1]	I
21	J4	TX5P[2]	O	157	C18	RX2P[2]	I
22	J3	TX5N[2]	O	158	B19	RX2N[2]	I
23	J2	TX5P[3]	O	159	A20	CTAP2[2]	I
24	J1	TX5N[3]	O	160	A19	RX2P[3]	I
25	K2	SPARE[2]	I	161	B18	RX2N[3]	I
26	K3	RESERVED[1]	IO	162	B17	CTAP2[3]	I
27	K1	RESERVED[2]	IO	163	C17	RX3P[0]	I

Table 6–1 SG1010 PBGA Ball Assignment

INDEX	BALL	SIGNAL NAME	TYPE	INDEX	BALL	SIGNAL NAME	TYPE
28	L1	RESERVED[3]	IO	164	D16	RX3N[0]	I
29	L2	RESERVED[4]	IO	165	A18	CTAP3[0]	I
30	L3	RESERVED[5]	IO	166	A17	RX3P[1]	I
31	M1	RESERVED[6]	IO	167	C16	RX3N[1]	I
32	M2	RESERVED[7]	IO	168	B16	CTAP3[1]	I
33	M3	RESERVED[8]	IO	169	A16	CTAP3[2]	I
34	M4	RESERVED[9]	IO	170	C15	RX3P[2]	I
35	N1	RESERVED[10]	IO	171	D14	RX3N[2]	I
36	N2	RESERVED[11]	IO	172	B15	RX3P[3]	I
37	N3	RESERVED[12]	IO	173	A15	RX3N[3]	I
38	P1	RESERVED[13]	IO	174	C14	CTAP3[3]	I
39	P2	RESERVED[14]	IO	175	B14	RX4P[0]	I
40	R1	RESERVED[15]	IO	176	A14	RX4N[0]	I
41	P3	RESERVED[16]	IO	177	C13	CTAP4[0]	I
42	R2	RESERVED[17]	IO	178	B13	RX4P[1]	I
43	T1	RESERVED[18]	IO	179	A13	RX4N[1]	I
44	P4	DIAG_EN	I	180	D12	CTAP4[1]	I
45	R3	TESTMODE[4]	I	181	C12	RX4P[2]	I
46	T2	TCK	I	182	B12	RX4N[2]	I
47	U1	TRST_L	I	183	A12	CTAP4[2]	I
48	T3	TMS	I	184	B11	RX4P[3]	I
49	U2	TDO	O	185	C11	RX4N[3]	I
50	V1	TDI	I	186	A11	CTAP4[3]	I
51	U3	SCAN_ENA	I	187	A10	TX4P[0]	O
52	V2	SPARE[1]	I	188	B10	TX4N[0]	O
53	W1	SCAN_OUT[0]	IO	189	C10	TX4P[1]	O
54	V3	SCAN_OUT[1]	IO	190	C9	TX4N[1]	O
55	W2	SCAN_OUT[2]	IO	191	B9	TX4P[2]	O
56	Y1	SCAN_OUT[3]	IO	192	A9	TX4N[2]	O
57	W3	SCAN_OUT[4]	IO	193	D9	TX4P[3]	O
58	Y2	SCAN_OUT[5]	IO	194	C8	TX4N[3]	O
59	W4	SCAN_OUT[6]	IO	195	B8	TX3P[0]	O
60	V4	SCAN_OUT[7]	IO	196	A8	TX3N[0]	O
61	U5	TESTMODE[3]	I	197	A7	TX3P[1]	O
62	Y3	TESTMODE[2]	I	198	B7	TX3N[1]	O
63	Y4	TESTMODE[1]	I	199	B6	TX3P[2]	O
64	V5	TESTMODE[0]	I	200	C7	TX3N[2]	O
65	W5	SR_CS_L	IO	201	A6	TX3P[3]	O
66	Y5	SR_CK	IO	202	A5	TX3N[3]	O
67	V6	SR_DI	IO	203	D7	TX2P[0]	O
68	W6	SR_DO	I	204	C6	TX2N[0]	O

Table 6–1 SG1010 PBGA Ball Assignment

INDEX	BALL	SIGNAL NAME	TYPE	INDEX	BALL	SIGNAL NAME	TYPE
69	Y6	GPIO[7]	IO	205	B5	TX2P[1]	O
70	V7	GPIO[6]	IO	206	A4	TX2N[1]	O
71	W7	GPIO[5]	IO	207	C5	TX2P[2]	O
72	Y7	GPIO[4]	IO	208	B4	TX2N[2]	O
73	V8	GPIO[3]	IO	209	A3	TX2P[3]	O
74	W8	GPIO[2]	IO	210	B3	TX2N[3]	O
75	Y8	GPIO[1]	IO	211	C4	RESLO	I
76	U9	GPIO[0]*	IO	212	B2	RESHI	I
77	V9	NRST_L	I	213	A2	REF14	I
78	W9	TSTCLKG	I	214	C3	REF10	I
79	Y9	PLLCLKGO	O	215	A1	VSS	
80	W10	LED5_L[3]*	IO	216	D4	VSS	
81	V10	LED5_L[2]*	IO	217	D8	VSS	
82	Y10	LED5_L[1]*	IO	218	D13	VSS	
83	Y11	LED5_L[0]*	IO	219	D17	VSS	
84	W11	LED4_L[3]*	IO	220	H4	VSS	
85	V11	LED4_L[2]*	IO	221	H17	VSS	
86	Y12	LED4_L[1]*	IO	222	J9	VSS	
87	W12	LED4_L[0]*	IO	223	J10	VSS	
88	V12	LED3_L[3]*	IO	224	J11	VSS	
89	U12	LED3_L[2]*	IO	225	J12	VSS	
90	Y13	LED3_L[1]*	IO	226	K9	VSS	
91	Y14	LED3_L[0]*	IO	227	K10	VSS	
92	W14	LED2_L[3]	IO	228	K11	VSS	
93	Y15	LED2_L[2]	IO	229	K12	VSS	
94	V14	LED2_L[1]*	IO	230	L9	VSS	
95	W15	LED2_L[0]*	IO	231	L10	VSS	
96	Y16	LED1_L[3]*	IO	232	L11	VSS	
97	U14	LED1_L[2]*	IO	233	L12	VSS	
98	V15	LED1_L[1]*	IO	234	M9	VSS	
99	W16	LED1_L[0]*	IO	235	M10	VSS	
100	Y17	LED0_L[3]*	IO	236	M11	VSS	
101	V16	LED0_L[2]*	IO	237	M12	VSS	
102	W17	LED0_L[1]*	IO	238	N4	VSS	
103	Y18	LED0_L[0]*	IO	239	N17	VSS	
104	U16	TSTSHFTLD	I	240	U4	VSS	
105	V17	ECSEL	I	241	U8	VSS	
106	W18	ETOGGLE	I	242	U13	VSS	
107	Y19	EXDNUP	I	243	U17	VSS	
108	V18	TSTPHASE	I	244	D11	V33	
109	W19	RESETTX	I	245	F4	V33	

Table 6–1 SG1010 PBGA Ball Assignment

INDEX	BALL	SIGNAL NAME	TYPE	INDEX	BALL	SIGNAL NAME	TYPE
110	Y20	LOOPBKEN	I	246	F17	V33	
111	W20	TESTRST	I	247	R4	V33	
112	V19	BYPASSL	I	248	R17	V33	
113	U19	TSTCLKL	I	249	U10	V33	
114	U18	REFCLKL	I	250	D10	V33	
115	T17	CTAP5[0]	I	251	T4	V33	
116	V20	RX5P[0]	I	252	U7	V33	
117	U20	RX5N[0]	I	253	D6	V15	
118	T18	RX5P[1]	I	254	D15	V15	
119	T19	RX5N[1]	I	255	K4	V15	
120	T20	CTAP5[1]	I	256	L17	V15	
121	R18	RX5P[2]	I	257	U6	V15	
122	P17	RX5N[2]	I	258	U15	V15	
123	R19	CTAP5[2]	I	259	D5	V15	
124	R20	CTAP5[3]	I	260	L4	V15	
125	P18	RX5P[3]	I	261	M17	V15	
126	P19	RX5N[3]	I	262	U11	V15	
127	P20	CTAP0[0]	I	263	L20	VDDA/V15 (CDR)	
128	N18	RX0P[0]	I	264	G19	VDDA/V15 (CDR)	
129	N19	RX0N[0]	I	265	K17	VDDA/V15 (CDR)	
130	N20	CTAP0[1]	I	266	E19	VDDA/V15 (CDR)	
131	M18	RX0P[1]	I	267	W13	VDDG/V15 (PLL)	
132	M19	RX0N[1]	I	268	V13	VSSG/VSS (PLL)	
133	M20	RX0P[2]	I	269	K20	VSSA/VSS (CDR)	
134	L19	RX0N[2]	I	270	J20	VSSA/VSS (CDR)	
135	L18	CTAP0[2]	I	271	D20	VSSA/VSS (CDR)	
136	K19	RX0P[3]	I	272	F20	VSSA/VSS (CDR)	

*

*These signal pins have multiple functions. See pin descriptions in Chapter 5 for more information.

Glossary

address routing — A mechanism used for routing frames through a fabric based on address decoding of the frame's address field at each node. PC I compatible routing through a PCI hierarchy uses address routing.

bridge — An edge node that provides protocol translation; for example, a bridge between the fabric and a PCI bus.

bundled port — A port that aggregates more than one link.

channel — A partitioned address range, used for address and path protection and address translation at edge nodes, and for StarFabric register space at all nodes.

Channel 255 — A dedicated channel used for register accesses for all StarFabric components.

chunk — A potentially non-contiguous data stream of indeterminate size whose transmission integrity is assured by sequence numbers and CRC.

DAC — Dual address cycle. A 64-bit memory region. *See also* **SAC**.

differential pair
differential signal pair — A pair of signal wires that connect a differential output buffer (Tx+ and Tx–) to a differential input buffer (Rx+ and Rx–).

disjoint route — A secondary route that shares no common switches with the primary route is said to be a disjoint route.

downstream — Used with transaction flow, frame direction, port, PCI bus, or PCI device to indicate flow away from the root.

edge node — Any node that is not a switch – either a bridge or a StarFabric native device.

fabric — The switched–serial interconnect using StarFabric, which includes switches and edge nodes, and the links that connect them.

frame — StarFabric data carriers. They carry read and write data as well as control and error information.They are made up of lines

frame header	The first 12 bytes internal to a frame. The header bytes provide frame management and are not part of the data payload.
header	*See* **frame header** *and* **StarFabric component header**.
initiator	A device that initiates a PCI transaction on a PCI bus.
input port	The port used by a node for frame reception. Used in the context of a single path or frame.
leaf	Any edge node that is not a root node. May be used in reference to edge nodes in a PCI hierarchy.
line	A unit of frame size. The smallest frame is one line, and all frames sizes are on single line granularities. A line is 16 bytes, or four Dwords.
line buffer	Storage in a node used to hold a received frame line before it is transmitted.
line credit	Allocates or frees buffer space, in line granularity, in a node.
link	A physical connection between nodes consisting of all the elements necessary for two nodes to communicate. A link includes the link interface in each node and the differential signal pairs that connect them. A link reference is used for operations involving line credits, fabric enumeration, CRC calculation, frame transmission, and frame reception.
link interface	The functional block within a node that is composed of a link transmitter and a link receiver.
link number	Refers to the absolute link numbers of a node's links.
link partner	The node that is connected to the StarFabric component in question, through a particular link, port, or differential pair.
link receiver	The subsection of a link interface that receives a frame. The link receiver includes the differential input buffers, clock recovery, serial-to-parallel converters, 8b/10b decoder, CRC logic, framing logic, and synchronization logic.
link transmitter	The subsection of a link interface that transmits a frame. The link transmitter includes the synchronization logic, framing logic, CRC logic, 8b/10b encoder, parallel-to-serial converters, and differential output buffers.
minimum turnaround latency	The minimum amount of time between the sending of a frame and the return of the line credits from the receiver of that frame.
multicast routing	A mechanism used for routing frames from a single origin to multiple edge nodes.
next turn	The turn specification for the next node in a path. Used for line credit operations and path routing.

node	Generic name for any edge node or switch that supports the StarFabric protocol. Also referred to as a StarFabric component.
origin	A node that is the generator of a frame. In the general case, the origin is an edge node, although there are specific cases where a switch can be an origin.
output port	The port on a node by which a frame is sent. Used in the context of a single path or frame.
path	The StarFabric specification for the position of a terminus relative to the origin in a fabric. The path is a field in the frame header and is used by nodes to direct a path-routed frame through the fabric.
path transform	The invert–and–reverse operation performed on a path specification to obtain the path to the origin of the frame.
path routing	A mechanism used for routing frames through a fabric based on a relative path from the origin to the terminus.
peer–to–peer	Refers to the flow of frames from one edge node to another edge node. For PCI transactions (address routing), it refers to the ability of one PCI peripheral to communicate directly with another PCI peripheral across bridges.
port	A logical connection representing the set of links that connects one node to another node. This connection representation is used for operations involving frame routing (address decoding, turn counts, multicast groups).
receiver	A node that is receiving a frame from another StarFabric component. Used in the context of a single frame transmission or a specific path.
root	The node that initiates mesh/fabric enumeration, and, for PCI-compatible fabrics, the node closest to the host processor in the PCI hierarchy.
route	The trail from an origin to a terminus. A particular route is specified by a path in StarFabric.
SAC	Single address cycle. A 32-bit memory region. *See also* **DAC**.
StarFabric component header	The set of standard registers implemented by all StarFabric components. The StarFabric component header is mapped at offset 0 of Channel 255 address space.
StarFabric component	Any edge node or switch that supports the StarFabric protocol. Also referred to as a node.
StarFabric or SF	The switched–serial protocol described in the document *StarFabric Architecture Specification*.

striping	The technique for increasing bandwidth by using multiple parallel ports to transfer a unit of data. For example, initial StarFabric devices can bundle up to four 622Mbps differential pairs into a 2.5Gbps full duplex link, for 5Gbps of total bandwidth.
switch	A node that receives a frame from one node and forwards the frame to another node.
target	A device that is the target of a PCI transaction on a PCI bus. Also referred to as a PCI target.
terminus	A node that is the intended end receiver of a frame in the fabric. In the general case, the terminus is an edge node, although there are specific cases where a switch can be a terminus.
thread	A physical link within a bundle.
thread number	Threads within a bundle are numbered for dependency and error reporting purposes. Thread numbers are bundle-specific.
transmission medium	The transmission medium is the physical material used to transport signals between two nodes. Typical transmission mediums include copper PCB etch, and copper conductors in a cable.
transmitter	A node that is sending a frame to another StarFabric component. Used in the context of a single frame transmission or a specific path.
turn	Refers to the relative direction a frame takes when traveling through a switch; that is, the position of the output port relative to the input port of a switch.
turn count	Indicates the number of valid turns in a path specification. When the path specification is in a frame header, indicates the number of valid turns that have been taken at that point in the route.
upstream	Used with transaction flow, frame direction, port, PCI bus, or PCI device to indicate flow towards the root.

Index

Acronyms

ASIC	Application-specific integrated circuit
Cfg	Configuration
CoS	Class of service
CSR	Control and status register
ECP	Extended capabilities port
ELP	Extended function list pointer
EMU	Event Message Unit
FID	Fabric ID
GPIO	General purpose I/O
H/W	Hardware
HP	High priority
P2P	PCI–to–PCI
PCI	Peripheral component interconnect
PFN	Parallel fabric number
R/W	Read/write, readable/writeable
S/W	Software
SFC	StarFabric component
SGF	Software generated frame
VPD	Vital product data
W1TS	Write 1 to set

Numerics

A

B

C

D

E

F

G–J

L

M–O

P

R

S

T–W