IBM PowerPRS™ 64G Packet Routing Switch

Datasheet

July 2, 2002

The following are trademarks of International Business Machines Corporation in the United States, or other countries, or both.

IBM	PowerNP
IBM Logo	PowerPRS

IEEE is a registered trademark of the Institute of Electrical and Electronics Engineers (IEEE).

All information contained in this document is subject to change without notice. The products described in this document are NOT intended for use in implantation, life support, space, nuclear, or military applications where malfunction may result in injury or death to persons. The information contained in this document does not affect or change IBM product specifications or warranties. Nothing in this document shall operate as an express or implied license or indemnity under the intellectual property rights of IBM or third parties. All information contained in this document was obtained in specific environments, and is presented as an illustration. The results obtained in other operating environments may vary.

IBM Microelectronics Division
1580 Route 52, Bldg. 504
Hopewell Junction, NY 12533-6351

The IBM home page can be found at
http://www.ibm.com

The IBM Microelectronics Division home page
can be found at http://www.chips.ibm.com

prs64G.03title.fm
July 2, 2002

Contents

List of Figures

List of Tables

1. General Information

1.1 Features

- Nonblocking, self-routing, single-stage switch
- 32 input ports and 32 output ports
- High performance:
    - Throughput of up to 2 Gbps per port without speed expansion
    - Aggregate throughput of up to 64 Gbps for one device or 128 Gbps for two devices
- Speed expansion:
    - Internal speed expansion (one device) doubles the port speed to up to 4 Gbps while halving the number of ports to 16
    - External speed expansion (two devices) doubles the port speed to up to 4 Gbps without changing the number of ports (at 32)
    - Combination of internal and external speed expansion (two devices) multiplies the port speed by four to up to 8 Gbps while halving the number of ports to 16
- Serial data communication of up to 500 Mbps, compliant with Electronic Industries Association/JEDEC Standard No. 8-6 regarding high-speed transceiver logic (HSTL)
- Multicast support without packet duplication in the shared memory
- Configurable number of traffic priorities (from one to four)
- Programmable output queue thresholds and shared memory thresholds
- Port paralleling that groups four ports to form one link at 8 Gbps without speed expansion or 16 Gbps with internal or external speed expansion
- Support for redundant switch-plane operation, including a scheduled switchover facility that operates without packet loss
- Serial processor interface (serial host interface)
- Configurable packet lengths
- Packet header of two to five bytes, containing destination bitmap, packet priority, and switch redundancy support information, all protected by a parity bit
- Shared memory comprised of a dynamically shared buffer with a total capacity of:
    - 1024 packets (of 64 to 80 bytes) for one device
    - 2048 packets (of 64 to 80 bytes) for two devices and external speed expansion
- Reception of control packets destined for the local processor on any input port
- Transmission of control packets from the local processor to any output port
- Detection of link liveness by reception of specific packets
- Programmable byte shuffling in egress packets
- CMOS 7SF (SA-27E) technology (L_{drawn} = 0.18 μm, L_{eff} = 0.11 μm): 1.8-V LVCMOS-compatible I/O for low-speed signals
- IEEE® Standard 1149.1 boundary scan to facilitate circuit-board testing
- Package: 1088-ball ceramic column grid array (CCGA) with direct lid attach (DLA)

1.2 Description

The IBM PowerPRS™ 64G Packet Routing Switch is one of a family of second-generation switching devices designed for high-performance, nonblocking, fixed-length packet switching. It enables the development of scalable switch fabrics with an aggregate bandwidth of 64 to 512 Gbps.

The PowerPRS 64G receives packets on up to 32 input ports and routes them to up to 32 output ports based on bitmap information contained in the packet header. To accomplish this, each PowerPRS 64G contains two 32 × 32 subswitch elements. Corresponding ports on the two subswitch elements operate in parallel to comprise a single device port.

Each PowerPRS 64G port is linked to an attached device with four differential pairs, each of which provides serial data communication of up to 500 Mbps. The PowerPRS 64G features a throughput of 2 Gbps per port, or 64 Gbps for all 32 ports. Two PowerPRS 64Gs offer a combined aggregate throughput of 128 Gbps. Internal speed expansion, external speed expansion, port expansion, and port paralleling provide options for scaling the combination of port speed and the number of ports.

Synchronization is not required between input ports. However, packets on a given port are always received or transmitted at a fixed rate according to the packet length. Four levels of packet priority provide quality-of-service support. Ingress and egress flow control are based on a grant mechanism.

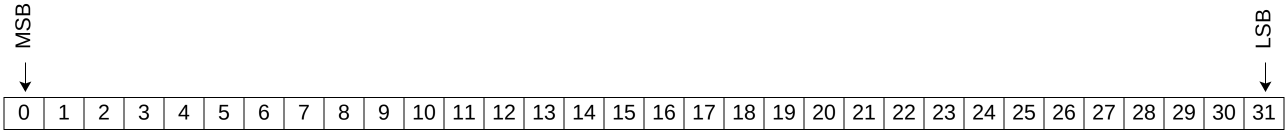
The PowerPRS 64G supports redundant switch-plane operation. It includes color-coded scheduled switch-over that operates without packet loss. Scheduled switchover is a system-level function that requires hardware and software interaction. The PowerPRS 64G performs the hardware assist for this function.

1.3 Ordering Information

Part Number	Description	Bandwidth	Throughput
IBM3288H2848	IBM Packet Routing Switch	64–512 Gbps	64 Gbps

1.4 Conventions and Notation

Throughout this document, standard IBM notation is used, meaning that bits and bytes are numbered in ascending order from left to right. For a four-byte word, bit 0 is the most significant bit (MSB) and bit 31 is the least significant bit (LSB).



0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

Notation for bit encoding is as follows:

- Hexadecimal values are preceded by an *x* and enclosed in single quotation marks. For example: x‘0A00’.
- Binary values in sentences appear in single quotation marks. For example: ‘1010’.

Differential pairs are designated by an _P for the positive signal and an _N for the negative signal at the end of the signal name. For example: DaslData[31]In[0]_P and DaslData[31]In[0]_N.

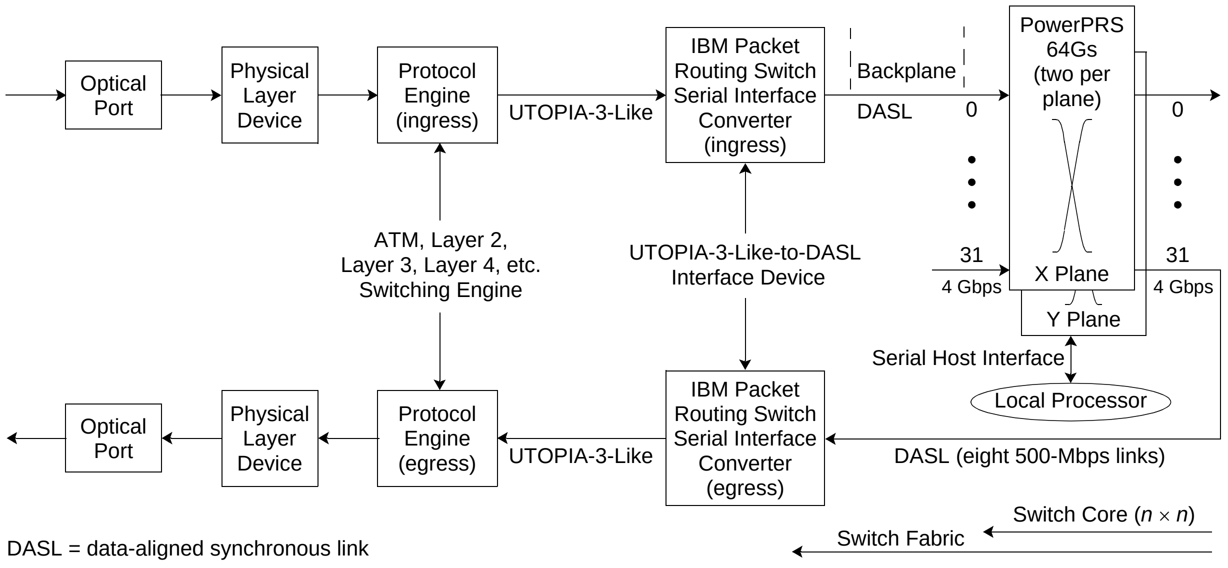
Nondifferential signals that are active low are designated by a # symbol at the end of the signal name. For example: InterruptOut#.

2. Architecture

2.1 System Application

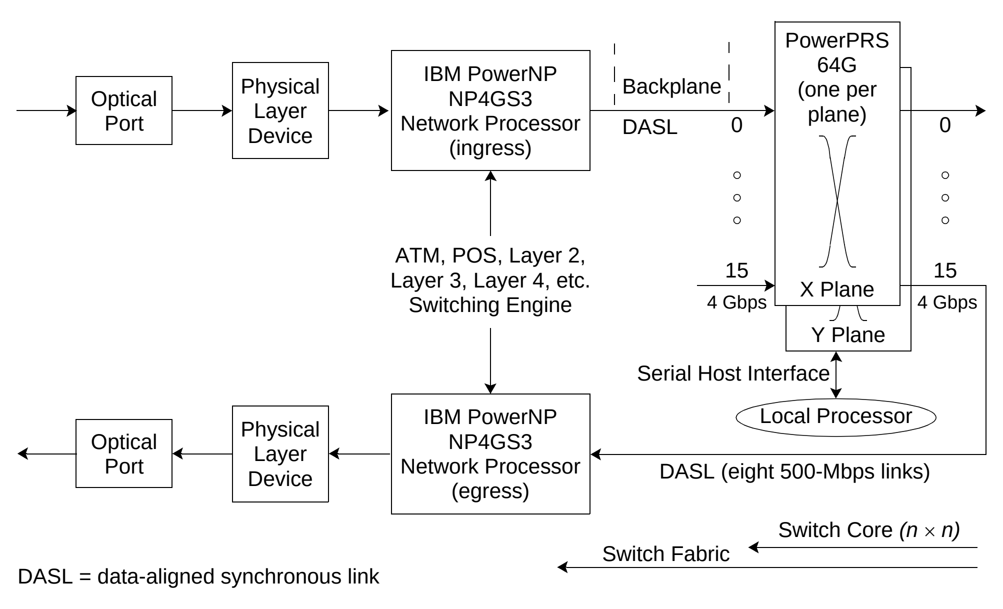
The IBM PowerPRS 64G Packet Routing Switch enables the construction of nonblocking scalable switch fabrics through repeated instances of the same switch element. It is designed for a wide variety of applications, including campus, wide-area network (WAN) edge, access, and backbone switches. When connected to the IBM Packet Routing Switch Serial Interface Converter, the PowerPRS 64G provides a complete redundant switch fabric for the attachment of a 32-bit wide interface to any protocol engine (see *Figure 2-1).* This supports such protocols as packet over SONET (POS), Gigabit Ethernet multilayer switching, and asynchronous transfer mode (ATM).

Figure 2-1. System View of the PowerPRS 64G with the IBM Packet Routing Switch Serial Interface Converter (configured with redundant 128-Gbps switch planes)



In a 16-port internal speed expansion configuration, the PowerPRS 64G connects directly to the IBM PowerNP™ NP4GS3 Network Processor, allowing a very compact and powerful switching system (see *Figure 2-2* on page 14). For more information regarding the NP4GS3, see the *IBM PowerNP NP4GS3 Databook* listing under *Related Documents* on page 157.

Figure 2-2. System View of the PowerPRS 64G with the IBM PowerNP NP4GS3 Network Processor



2.2 Internal Structure

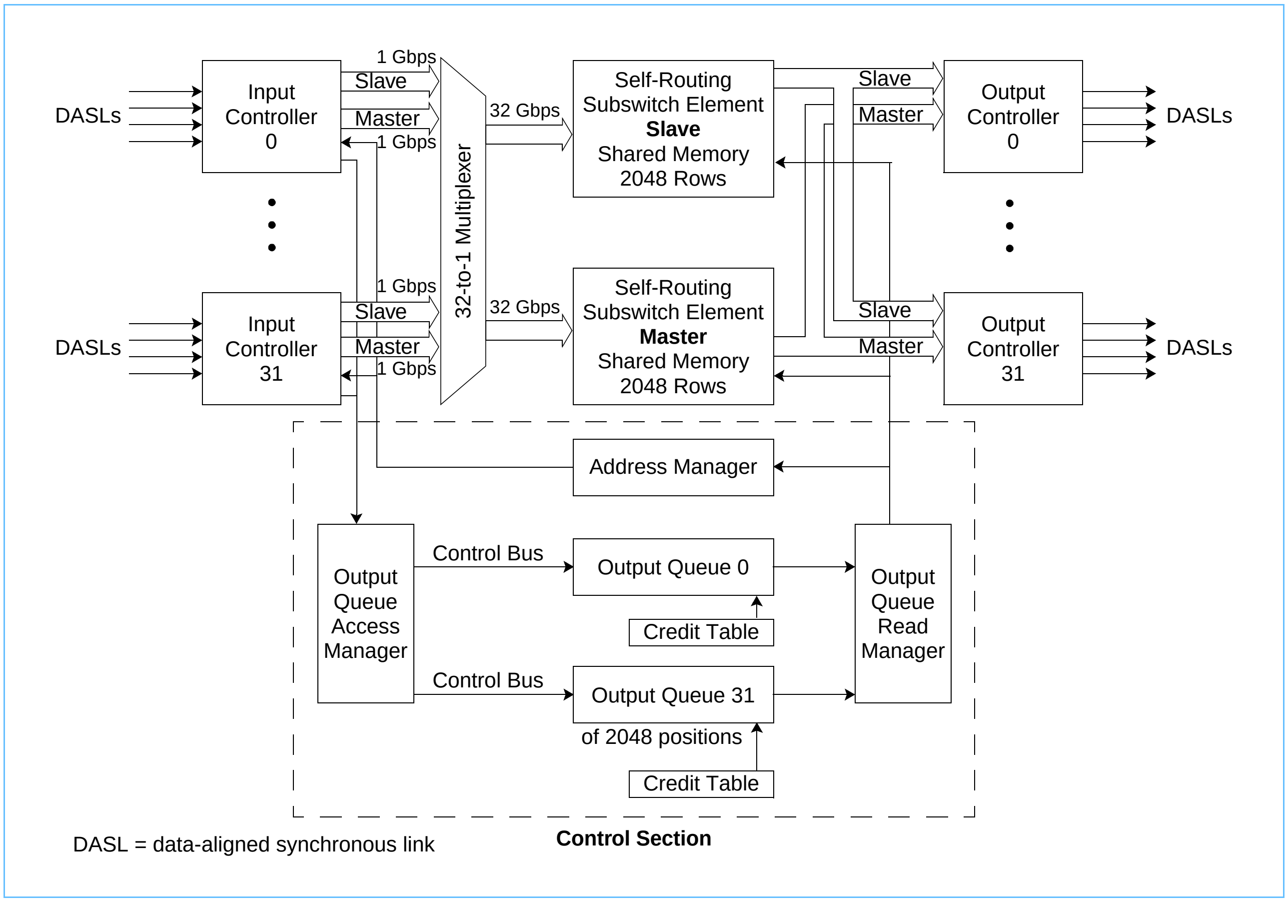
The internal structure of the PowerPRS 64G is shown in *Figure 2-3* on page 15, which depicts a single 64G operating without speed expansion.

The main components of the PowerPRS 64G are:

- Two self-routing subswitch elements, master and slave, that house the shared memory
- Thirty-two input controllers
- Thirty-two output controllers
- One control section, including:
  - Sequencer
  - Address manager
  - Output queue access manager
  - Output queues
  - Output queue read manager
  - Credit table

Data-aligned synchronous links (DASLs) provide the physical links between the PowerPRS 64G and the attached devices.

Figure 2-3. PowerPRS 64G Block Diagram



2.2.1 DASL Interface

As shown in *Figure 2-3,* the PowerPRS 64G includes two 32 × 32 self-routing subswitch elements, one designated master and the other designated slave. Like-numbered ports of the two elements are paired to form a single port for the device. Each PowerPRS 64G port is linked to an attached device with four DASLs (two per subswitch element). Each DASL is a differential pair that provides serial data communication of up to 500 Mbps. Therefore, the PowerPRS 64G features a throughput of 2 Gbps per port, or 64 Gbps for all 32 ports. Two PowerPRS 64Gs offer a combined aggregate throughput of 128 Gbps. See *Section 3.2 Physical Interface* on page 24 for more information.

Packets are transmitted in equal lengths called logical units (LUs). An LU is carried over two DASLs and stored in one subswitch element. For the basic PowerPRS 64G configuration (one device with 32 2-Gbps ports), packets are divided into two LUs: the master LU, which is transmitted to the master subswitch element; and the slave LU, which is transmitted to the slave subswitch element.

2.2.2 Shared Memory

The shared memory stores the packets that the PowerPRS 64G has received but has not yet transmitted. It is organized into two memory banks, one master and one slave. Each memory bank consists of 2048 20-byte rows, and has one write port and one read port.

2.2.3 Sequencer

The sequencer controls the PowerPRS 64G internal data flow by granting shared memory access to the input and output ports. Sequencer operation is based on time-division multiplexing (TDM). The sequencer cycles concurrently among the input and output ports, granting shared memory access to one input and one output port at a time and visiting each port once per cycle.

During each shared memory access, 16 to 20 bytes of data are processed (written or read), depending on packet length. Processing an entire LU requires either one or two shared memory accesses, depending on device configuration and packet length (see *Section 3.3.5 Master/Slave Synchronization with Two Devices* on page 27). The sequencer cycle equals the time required to process the data associated with one shared memory access. All sequencer cycles are equal in length.

The sequencer ensures that packets on a given port are always processed at a fixed interval according to their LU length; therefore, no synchronization is required between input ports.

2.2.4 Address Manager

The address manager tracks the available shared memory addresses and provides new store addresses to the input controllers. When the address manager provides a store address to an input controller, it removes that address from the available shared memory address pool. After the packet is transmitted, the output queue read manager returns the address to the address manager, which returns it to the available address pool. For multicast packets, one store address is sent to multiple output queues. The address manager tracks the number of output queues holding each store address and, when the count reaches zero, returns the address to the available shared memory address pool.

2.2.5 Input Controllers

The PowerPRS 64G has 32 input controllers, one input controller per port. When a packet arrives at a port, the input controller extracts the header information (including packet priority and destination) from the master LU. It checks the master LU header integrity using a parity bit on the header bytes. If the packet is valid, the input controller stores it in the shared memory (at the store address provided by the address manager) when access is granted by the sequencer. The input controller also forwards the shared memory address, packet priority, and packet destination to the output queue access manager. Packets arrive with a priority of 0 to 3, with 0 being the highest priority. Note that multicast packets have only one priority for all destinations.

2.2.6 Output Queue Access Manager

The output queue access manager receives the packet store address, priority, and destination from the input controllers and forwards this information to the output queues. The output queue access manager also maintains the counters that the PowerPRS 64G uses to control ingress flow. For each output queue, a counter tracks the total number of packets enqueued for that output, regardless of priority. Another counter tracks the total number of packets stored in the shared memory, regardless of output or priority.

2.2.7 Output Queues

The output queues contain the shared memory addresses of packets awaiting transmission from the PowerPRS 64G. For each port, there is one output queue per priority. An output queue stores the address of every packet enqueued for a particular output and priority. Each output queue can store up to 2048 addresses. Within an output queue, packet addresses are organized in a first-in-first-out (FIFO) queuing structure.

2.2.8 Output Queue Read Manager and Credit Table

The output queue read manager determines which packet the PowerPRS 64G transmits on each port. For each port, the output queues provide an output queue status (that is, output queue empty), one per priority, to the output queue read manager. The output queue read manager also receives the send grants that control the egress flow to the attached devices. The output queue read manager uses this information to decide which output queue to access for the next egress packet on the output port. In general, this will be the highest priority of the occupied output queues (so that high-priority packets overtake low-priority packets). However, a fixed amount of bandwidth can be assigned to low-priority packets by altering priority scheduling in the credit table. Once the output queue read manager determines which output queue to access, it retrieves the shared memory address of the next packet from that output queue and forwards the address to the shared memory.

2.2.9 Output Controllers

The PowerPRS 64G has 32 output controllers, one output controller per port. The output controller retrieves the next packet to be transmitted on a port from the address that the output queue read manager supplies to the shared memory. It inserts ingress flow control information (that is, it inserts the output queue grants) into the packet header (if that function is enabled). When the sequencer cycles to an output port, the output controller transmits the packet to the attached device.

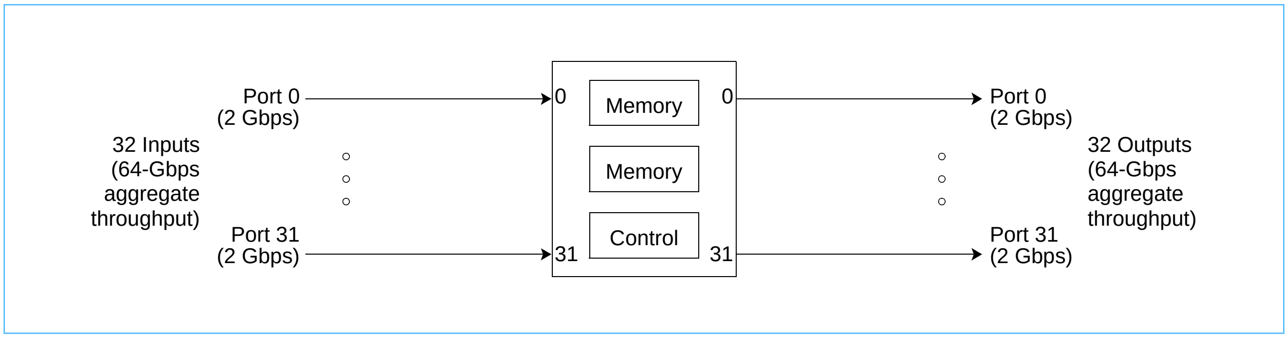
2.3 Speed Expansion Configurations

Internal speed expansion, external speed expansion, and port expansion provide options for scaling the combination of port speed and the number of ports by grouping ports or adding devices.

2.3.1 Basic Configuration: One Device without Speed Expansion

In its most basic configuration, the PowerPRS 64G operates as a single device without speed expansion. As described previously, this provides 32 ports at 2 Gbps per port, for an aggregate throughput of 64 Gbps (see *Figure 2-4* on page 18).

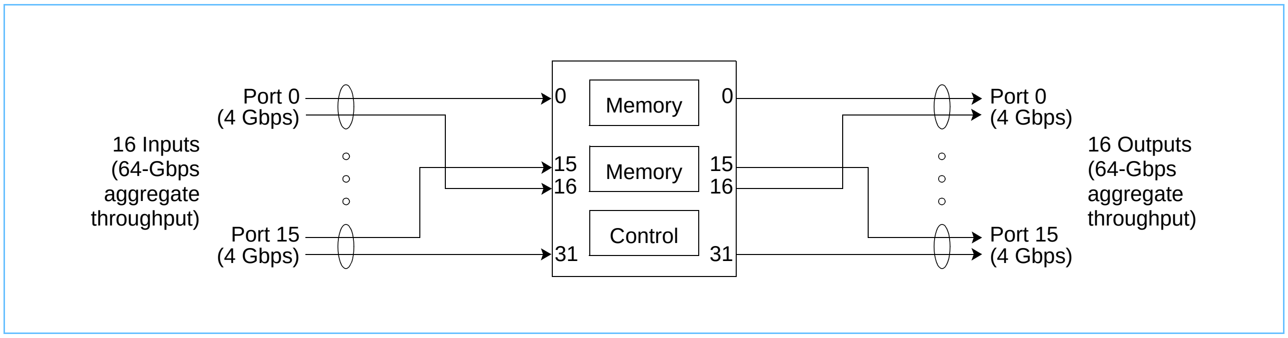
Figure 2-4. Block Diagram: One Device without Speed Expansion



2.3.2 One Device with Internal Speed Expansion

With internal speed expansion, the ports of a single PowerPRS 64G are paired. This doubles the port speed to 4 Gbps and halves the number of ports to 16 (see *Figure 2-5).* Ports are paired as follows: port 0 is paired with port 16, port 1 is paired with port 17, and so forth. Each pair of ports includes a master (ports 0 through 15) and a slave (ports 16 to 31).

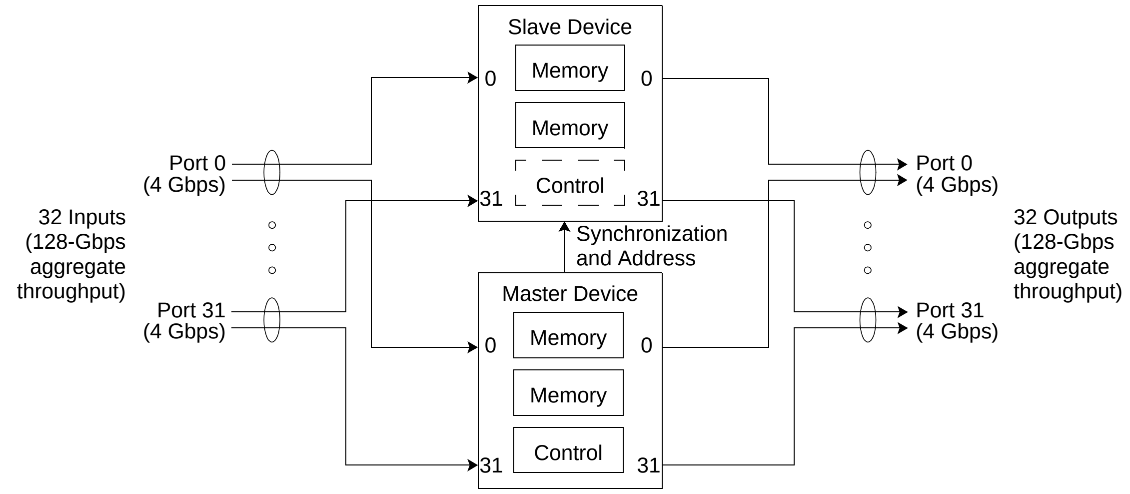
Figure 2-5. Block Diagram: One Device with Internal Speed Expansion



2.3.3 Two Devices with External Speed Expansion

With external speed expansion, two PowerPRS 64Gs are connected in parallel. Like-numbered ports between the two devices are paired, which doubles the port speed to 4 Gbps (see *Figure 2-6* on page 19). The total number of ports remains unchanged at 32. When two devices are configured for external speed expansion, one device is the master and one is the slave. The control section of the slave device is inactive. Packet synchronization and address information is passed from the master to the slave (see *Section 3.3.5 Master/Slave Synchronization with Two Devices* on page 27).

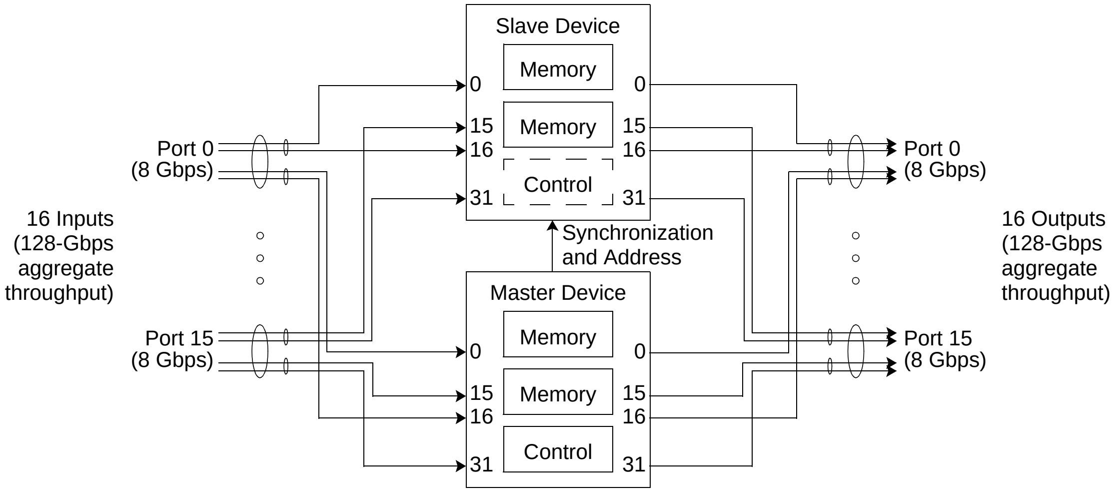
Figure 2-6. Block Diagram: Two Devices with External Speed Expansion



2.3.4 Two Devices with Internal and External Speed Expansion

With internal and external speed expansion, two PowerPRS 64Gs are connected so that the ports within each device are paired *and* the like-numbered ports between the two devices are paired (see *Figure 2-7* on page 20). Compared to a single PowerPRS 64G in the basic configuration (no speed expansion), this yields a four-fold increase in port speed, to 8 Gbps, and halves the number of ports to 16. One device is the master and one is the slave. The control section of the slave device is inactive. Packet synchronization and address information is passed from the master to the slave (see *Section 3.3.5 Master/Slave Synchronization with Two Devices* on page 27).

Figure 2-7. Block Diagram: Two Devices with Internal and External Speed Expansion



2.3.5 Speed Expansion Summary

The shared memory capacity and other device information for the various speed expansion configurations are summarized in *Table 2-1.*

Table 2-1. Speed Expansion Summary

Configuration	Number of Ports	Port Speed (Gbps)	Packet Length (bytes)	LU Size (bytes)[1]	Shared Memory Capacity (packets)
Single device without speed expansion (basic)	32 × 32	2	64–80	32–40	1024
			32–40	16–20	2048
Single device with internal speed expansion	16 × 16	4	64–80	16–20	1024
			128–160	32–40	512
Two devices with external speed expansion (no internal speed expansion)	32 × 32	4	64–80	16–20	2048
			128–160	32–40	1024
Two devices with external and internal speed expansion	16 × 16	8	128–160	16–20	1024
			256–320	32–40	512

1. The LU size in one subswitch element (either the master or the slave of one device) is equal to the packet length divided by the speed expansion factor. The speed expansion factor is the device port speed divided by the subswitch element port speed (which is 1 Gbps).

2.4 Port Expansion

With single-stage port expansion, multiple devices are interconnected in parallel to increase the number of ports without changing the port speed (see *Figure 2-8).* An external function must be provided to duplicate incoming packets and insert the correct destination bitmap used by each subswitch element and to merge traffic from different subswitch elements.

Port expansion can be combined with internal speed expansion and/or external speed expansion to simultaneously increase port speed and the number of ports.

2.5 Port Paralleling

In port paralleling, four ports are grouped to form one link. Port paralleling of four ports without speed expansion provides an 8-Gbps port (see *Figure 2-9* on page 22). With internal or external speed expansion, port paralleling provides a 16-Gbps port (see *Figure 2-10* on page 22). For more information about port paralleling, see *Section 3.11 Port Paralleling* on page 53.

Figure 2-8. Block Diagram: Four Devices with Single-Stage Port Expansion

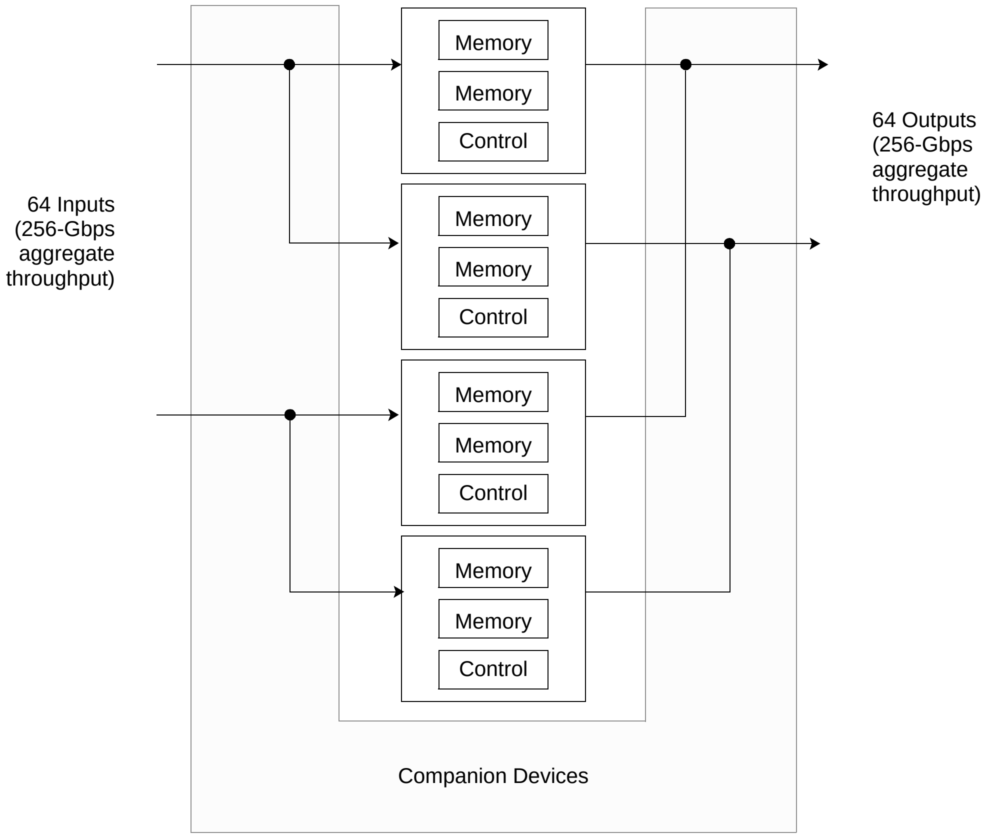


Figure 2-9. Block Diagram: One Device with Four-Port Port Paralleling

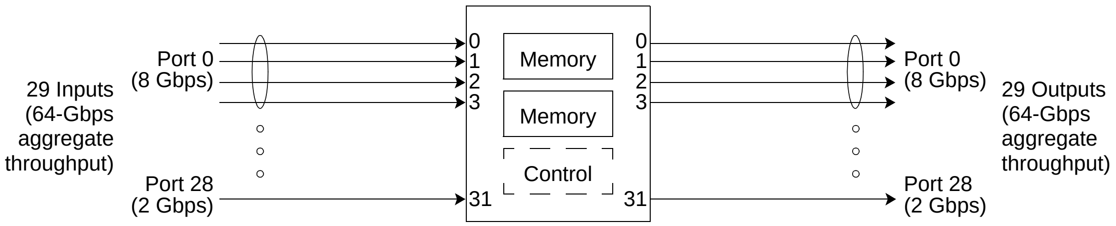
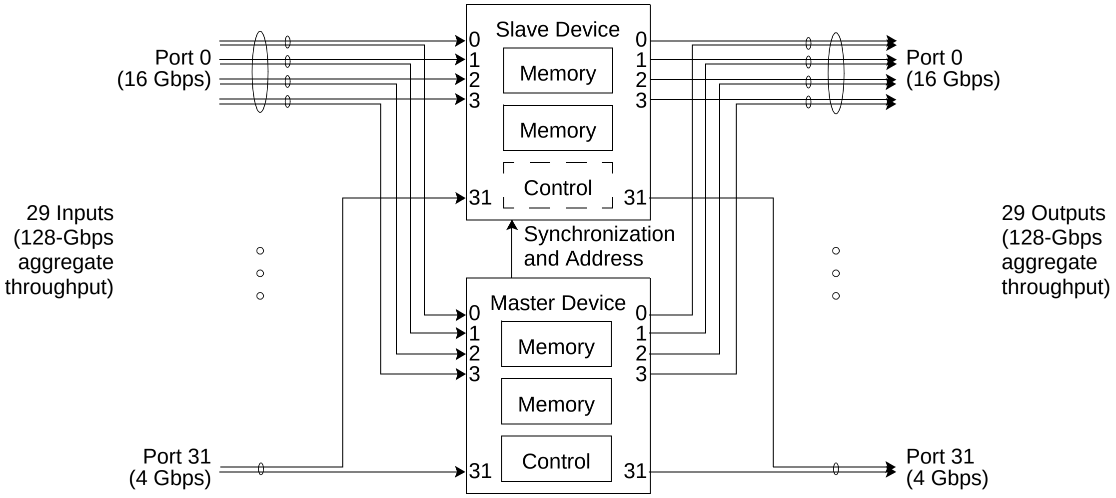


Figure 2-10. Block Diagram: Two Devices with External Speed Expansion and Port Paralleling



3. Functional Description

This section describes basic PowerPRS 64G functionality, including information about:

- Packet type
- Physical interface
- Logical unit (LU) format versus speed expansion
- Packet format according to packet type
- Ingress and egress flow control
- Packet reception and transmission
- Side communication channel (SCC)
- Switchover support
- Port paralleling

3.1 Packet Type

There are five types of packets:

- Data packets
- Control packets
- Idle packets
- Synchronization packets
- Service packets

For information about packet formats, see *Section 3.4 Packet Format According to Packet Type* on page 29.

In the basic configuration, the PowerPRS 64G output queue read manager prioritizes packet transmission for each output port in the following order:

1. Service packets
2. Control packets
3. Priority 0 data packets
4. Priority 1 data packets
5. Priority 2 data packets
6. Priority 3 data packets

3.1.1 Data Packets

Data packets carry user data to be switched from an input to one or more outputs. Data packets have a priority of 0, 1, 2, or 3, with 0 being the highest priority. They also carry routing information (destination bitmap), filtering information used for switchover support (color coding), and a “best-effort discard” flag. Egress data packets, along with egress idle packets, carry the output queue grants that control ingress flow to the PowerPRS 64G (see *Section 3.4.2 Data/Control Packet Format* on page 29).

3.1.2 Control Packets

Control packets carry the communications between the local processor and the protocol engine. Control packets do not have a specific priority.

Ingress control packets originate at the protocol engine. The PowerPRS 64G can receive control packets on any input port. When the destination bitmap of an ingress packet is all zeros, the input controller detects a control packet and stores it in the shared memory. Control packet addresses are stored in a special queue that holds up to 32 addresses.

Egress control packets originate at the local processor. The local processor stores the egress control packets in a specific shared memory location. The local processor can transmit control packets on any output port. Control packets are always transmitted on an output before any other packets in the shared memory destined for that output.

3.1.3 Idle Packets

Idle packets do not carry user data. They are transmitted on a port only when there are no data, control, or service packets available for transmission or when these packets cannot be transmitted (for example, during a flow control situation). The attached ingress devices generate ingress idle packets; the PowerPRS 64G generates egress idle packets. Egress idle and data packets carry the output queue grants used for ingress flow control. Idle packet color coding is used for switchover support.

3.1.4 Synchronization Packets

Synchronization packets provide the bit transition and packet delineation necessary for link synchronization. See *Section 3.2 Physical Interface* for information about the bit sequence used for synchronization.

3.1.5 Service Packets

Service packets are used to test link liveness. The PowerPRS 64G is capable of processing three types of service packets: yellow type 1, yellow type 2, and yellow type 3.

3.2 Physical Interface

Data-aligned synchronous links (DASLs) connect the PowerPRS 64G to the attached devices. Each PowerPRS 64G port is linked to the attached device with four DASLs (two per subswitch element). Packets are transmitted in two or more logical units (LUs). Each LU is transmitted over two DASLs in two 500-Mbps (2-ns) bit streams. One bit stream carries the even-numbered bits and the other bit stream carries the odd-numbered bits. For the basic PowerPRS 64G configuration (one device with 32 2-Gbps ports), packets consist of two LUs: the master LU, which is transmitted to the master subswitch element; and the slave LU, which is transmitted to the slave subswitch element.

Table 3-1 on page 25 presents the information and the bit order carried by each differential pair for the basic PowerPRS 64G configuration. At the device pin level, each bit stream interface is differential and complies with Electronic Industries Association/JEDEC Standard No. 8-6 regarding high-speed transceiver logic (HSTL).

Table 3-1. Bit Organization of Input and Output Ports

Input Differential Pair	Output Differential Pair	Information Carried	Bit Order
DaslData[p]In[0]_P and DaslData[p]In[0]_N	DaslData[p]Out[0]_P and DaslData[p]Out[0]_N	Even-numbered bits of **slave** byte stream	b0, b2, b4, and b6 of **slave** byte stream
DaslData[p]In[1]_P and DaslData[p]In[1]_N	DaslData[p]Out[1]_P and DaslData[p]Out[1]_N	Odd-numbered bits of **slave** byte stream	b1, b3, b5, and b7 of **slave** byte stream
DaslData[p]In[2]_P and DaslData[p]In[2]_N	DaslData[p]Out[2]_P and DaslData[p]Out[2]_N	Even-numbered bits of **master** byte stream	b0, b2, b4, and b6 of **master** byte stream
DaslData[p]In[3]_P and DaslData[p]In[3]_N	DaslData[p]Out[3]_P and DaslData[p]Out[3]_N	Odd-numbered bits of **master** byte stream	b1, b3, b5, and b7 of **master** byte stream

On the ingress path, the physical interface to the PowerPRS 64G deserializes the two 500-Mbps bit streams of an LU into one 125-MBps (8-ns) byte stream for transmission within the device. Each byte stream is routed to a subswitch element. On the egress path, the physical interface serializes the single byte stream into two bit streams.

Data bits are transferred across devices at a known frequency, so no companion clock is required. However, the DASL interfaces between the PowerPRS 64G and the attached devices must be synchronized to provide bit phase alignment and packet delineation at all data receivers.

To synchronize a link, a specific bit sequence is sent over each of the two differential pairs that carry a byte stream (either master or slave and either input or output). The sequence is a series of x‘A’ (‘1010’) characters followed by a x‘5’ (‘0101’) character (see *Figure 3-1).* This sequence guarantees the bit transition, and the x‘A’-to-x‘5’ transition allows for packet delineation. Because one differential pair carries the even-numbered bits while the other pair carries the odd-numbered bits (see *Table 3-1),* the single byte stream equivalent to the two bit streams is a series of x‘CC’ (‘1100 1100’) characters followed by a x‘33’ (‘0011 0011’) character (see *Section 3.4.4 Synchronization Packet Format* on page 35).

Figure 3-1. Link Synchronization Pattern

<table>
<tr><td>Even Bit Stream</td><td colspan="2">1</td><td colspan="2">0</td><td colspan="2">1</td><td colspan="2">0</td><td colspan="2">1</td><td colspan="2">0</td><td colspan="2">1</td><td colspan="2">0</td><td colspan="2">. . .</td><td colspan="2">0</td><td colspan="2">1</td><td colspan="2">0</td><td colspan="2">1</td></tr>
<tr><td>Odd Bit Stream</td><td colspan="2">1</td><td colspan="2">0</td><td colspan="2">1</td><td colspan="2">0</td><td colspan="2">1</td><td colspan="2">0</td><td colspan="2">1</td><td colspan="2">0</td><td colspan="2">. . .</td><td colspan="2">0</td><td colspan="2">1</td><td colspan="2">0</td><td colspan="2">1</td></tr>
<tr><td>Full Byte Stream</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td colspan="2">. . .</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td></tr>
</table>

Two internal picoprocessors, the shared DASL controllers (SDCs), control link synchronization (see *Section 6.3 DASL Synchronization and Operation* on page 111).

3.3 Logical Unit Format versus Speed Expansion Configuration

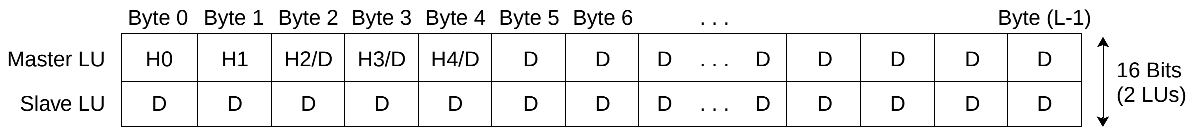
The number of LUs per packet is equal to the number of subswitch element ports (that is, the number of data streams) that comprise an aggregate port, which depends on device configuration. Packet LUs are always transmitted or received at the same time on all the data streams that comprise a port. The LUs of successive packets are transmitted one after the other, with no gap between the packets.

Regardless of how many LUs comprise a packet, only one LU is the master; the other LUs are all slaves. The master LU carries the packet header bytes followed by packet payload bytes. Slave LUs carry only payload bytes. The data packet header is two to five bytes long, depending on the header length, and includes packet control information such as routing information. In *Figures 3-2* through *3-9* (page 36), the data packet header bytes are denoted as H0 through H4.

3.3.1 Basic Configuration: One Device without Speed Expansion

As discussed in *Section 2.2.1* on page 15, in the basic configuration, a packet is divided into a master LU and a slave LU (see *Figure 3-2).* Each 2-Gbps port carries the master LU and the slave LU for one packet at a time. The master LU is transmitted to the master subswitch element and the slave LU is transmitted to the slave subswitch element.

Figure 3-2. Packet Format for a 2-Gbps Port



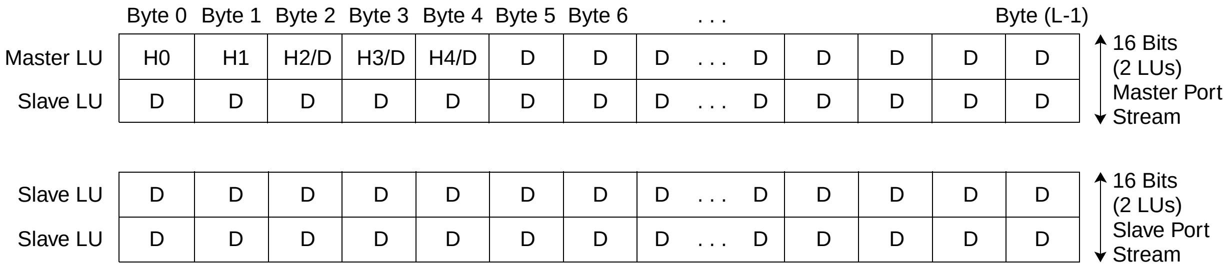
	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	. . .				Byte (L-1)
Master LU	H0	H1	H2/D	H3/D	H4/D	D	D	D . . . D	D	D	D	D
Slave LU	D	D	D	D	D	D	D	D . . . D	D	D	D	D

Notes:
1. H0 through H4 are packet header bytes. The number of packet header bytes depends on the device configuration.
2. D is user data.

3.3.2 One Device with Internal Speed Expansion or Two Devices with External Speed Expansion

For one device with internal speed expansion or two devices with external (but not internal) speed expansion, the two 2-Gbps ports are paired to form a 4-Gbps port. Because each port carries two data streams, packets are divided into four LUs. The first LU is the master and the other LUs are the slaves. In the paired ports, one port operates as the master and the other port as the slave. The master port is comprised of one master data stream and one slave data stream; the slave port is comprised of two slave data streams (see *Figure 3-3).*

Figure 3-3. Packet Format for a 4-Gbps Port



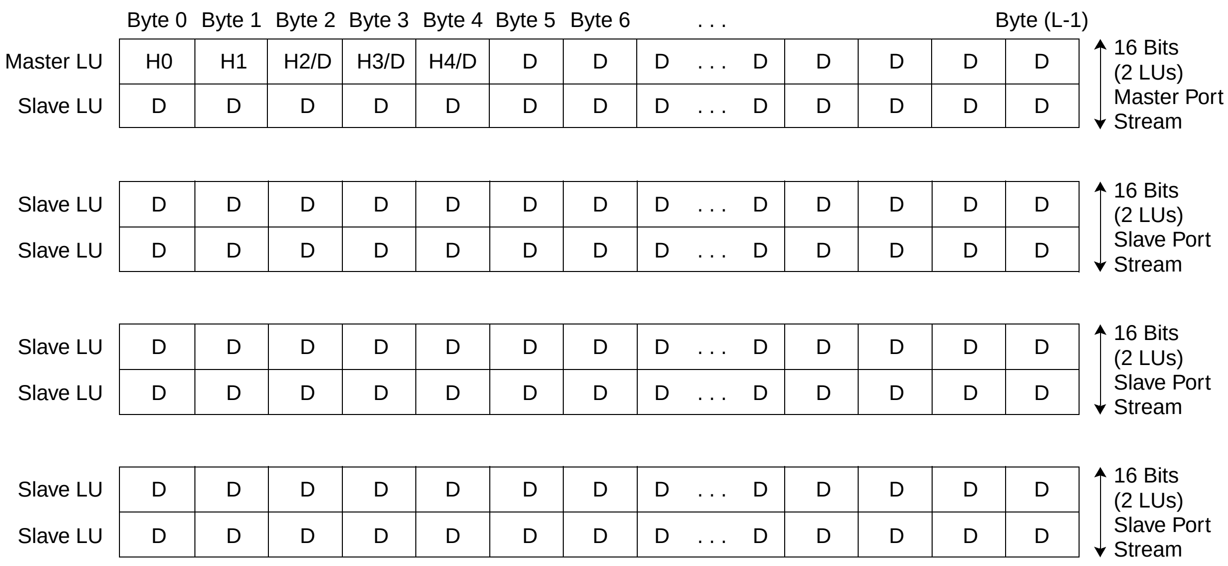
	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	. . .				Byte (L-1)
Master LU	H0	H1	H2/D	H3/D	H4/D	D	D	D . . . D	D	D	D	D
Slave LU	D	D	D	D	D	D	D	D . . . D	D	D	D	D
Slave LU	D	D	D	D	D	D	D	D . . . D	D	D	D	D
Slave LU	D	D	D	D	D	D	D	D . . . D	D	D	D	D

Notes:
1. H0 through H4 are packet header bytes. The number of packet header bytes depends on the device configuration.
2. D is user data.

3.3.3 Two Devices with Internal and External Speed Expansion

For two devices with both internal and external speed expansion, the four 2-Gbps ports are grouped to form an 8-Gbps port. Because each ungrouped port carries two data streams, packets are divided into eight LUs. The first LU is the master and the other LUs are the slaves. In the grouped port, one port operates as the master and the other three ports operate as slaves. The master port is comprised of one master data stream and one slave data stream; each slave port is comprised of two slave data streams (see *Figure 3-4).*

Figure 3-4. Packet Format for an 8-Gbps Port



	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	. . .					Byte (L-1)	
Master LU	H0	H1	H2/D	H3/D	H4/D	D	D	D . . . D	D	D	D	D		16 Bits (2 LUs) Master Port Stream
Slave LU	D	D	D	D	D	D	D	D . . . D	D	D	D	D		
Slave LU	D	D	D	D	D	D	D	D . . . D	D	D	D	D		16 Bits (2 LUs) Slave Port Stream
Slave LU	D	D	D	D	D	D	D	D . . . D	D	D	D	D		
Slave LU	D	D	D	D	D	D	D	D . . . D	D	D	D	D		16 Bits (2 LUs) Slave Port Stream
Slave LU	D	D	D	D	D	D	D	D . . . D	D	D	D	D		
Slave LU	D	D	D	D	D	D	D	D . . . D	D	D	D	D		16 Bits (2 LUs) Slave Port Stream
Slave LU	D	D	D	D	D	D	D	D . . . D	D	D	D	D		

Notes:

1. H0 through H4 are packet header bytes. The number of packet header bytes depends on the device configuration.
2. D is user data.

3.3.4 Two Devices with Port Paralleling and External Speed Expansion

For two devices with port paralleling and external speed expansion, the four 2-Gbps ports on each of the two devices are grouped to form a 16-Gbps port. Packets are divided into four LUs, and the grouped port carries four packets at a time (see *Figure 3-5* on page 28). A packet is carried on the like-numbered ports of the two devices. In each of the four paired ports, one port operates as the master and the other as the slave. The four master ports are each comprised of one master data stream and one slave data stream; the four slave ports are each comprised of two slave data streams.

3.3.5 Master/Slave Synchronization with Two Devices

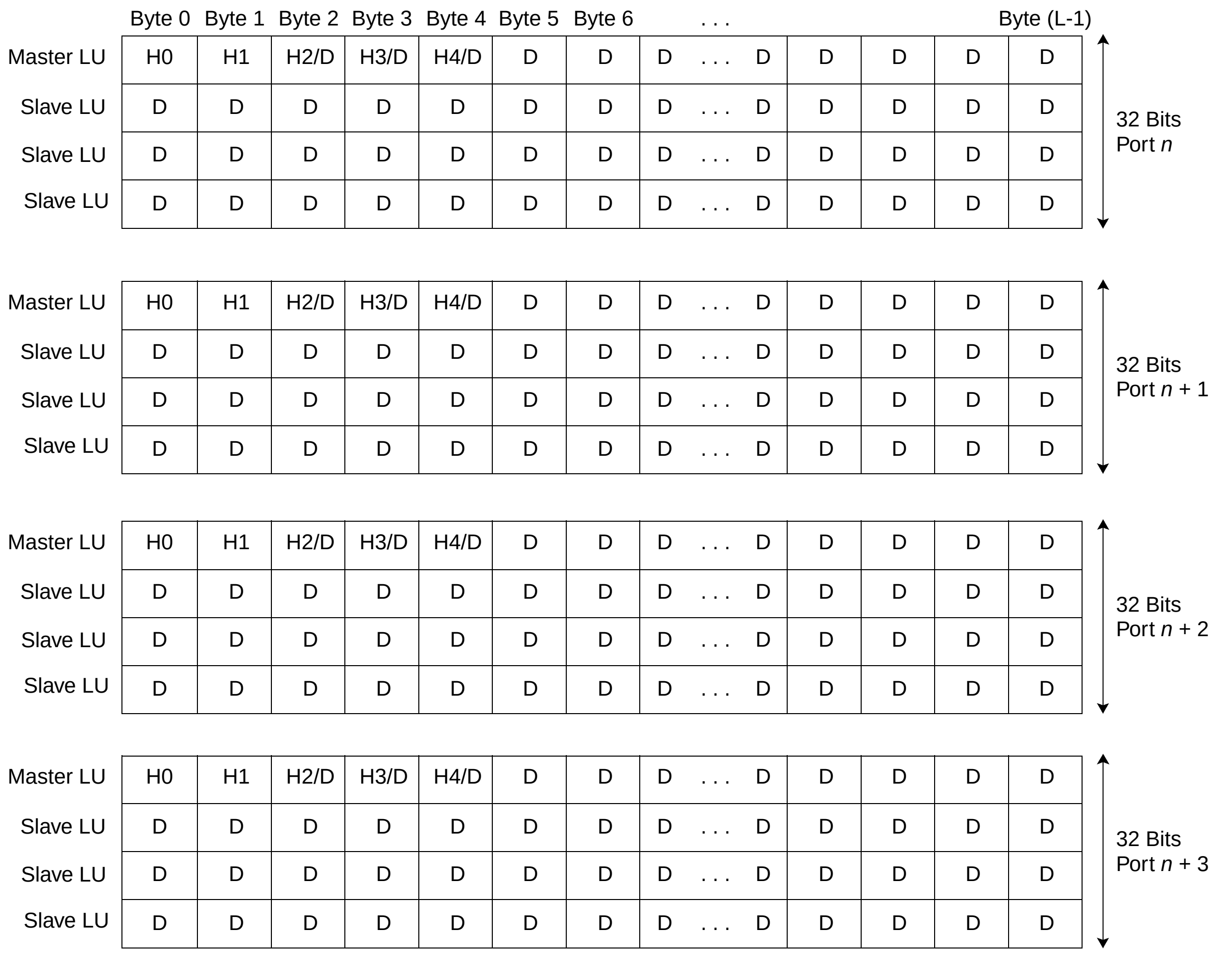
3.3.5.1 Sequencers

As discussed above, packets are processed in logical units (LUs). The number of LUs is equal to the number of data streams that comprise a port, which depends on the device configuration.

Each PowerPRS 64G contains a sequencer. When two devices are configured for external speed expansion, the slave device sequencer must be synchronized to the master device sequencer to ensure that the LUs for a particular port (or packet) are processed at the same time on both devices. The MSBusSyncOut pin and MSBusSyncIn pin conduct this synchronization. The MSBusSyncIn/Out pin mode field in the *Configuration 1 Register* (page 87) sets the pin configuration.

Note: When the packet length is greater than 128 bytes, either with or without speed expansion, LU transmission requires two sequencer cycles of equal length. When packet length is less than 128 bytes in the speed expansion configuration, LU transmission requires only one sequencer cycle.

Figure 3-5. Packet Format for a 16-Gbps Port



	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	. . .				Byte (L-1)	
Master LU	H0	H1	H2/D	H3/D	H4/D	D	D	D . . . D	D	D	D	D	32 Bits Port *n*
Slave LU	D	D	D	D	D	D	D	D . . . D	D	D	D	D	
Slave LU	D	D	D	D	D	D	D	D . . . D	D	D	D	D	
Slave LU	D	D	D	D	D	D	D	D . . . D	D	D	D	D	
Master LU	H0	H1	H2/D	H3/D	H4/D	D	D	D . . . D	D	D	D	D	32 Bits Port *n* + 1
Slave LU	D	D	D	D	D	D	D	D . . . D	D	D	D	D	
Slave LU	D	D	D	D	D	D	D	D . . . D	D	D	D	D	
Slave LU	D	D	D	D	D	D	D	D . . . D	D	D	D	D	
Master LU	H0	H1	H2/D	H3/D	H4/D	D	D	D . . . D	D	D	D	D	32 Bits Port *n* + 2
Slave LU	D	D	D	D	D	D	D	D . . . D	D	D	D	D	
Slave LU	D	D	D	D	D	D	D	D . . . D	D	D	D	D	
Slave LU	D	D	D	D	D	D	D	D . . . D	D	D	D	D	
Master LU	H0	H1	H2/D	H3/D	H4/D	D	D	D . . . D	D	D	D	D	32 Bits Port *n* + 3
Slave LU	D	D	D	D	D	D	D	D . . . D	D	D	D	D	
Slave LU	D	D	D	D	D	D	D	D . . . D	D	D	D	D	
Slave LU	D	D	D	D	D	D	D	D . . . D	D	D	D	D	

Notes:

1. H0 through H4 are packet header bytes. The number of packet header bytes depends on the device configuration.
2. D is user data.

3.3.5.2 Shared Memory Addresses

When two devices are configured for external speed expansion, the master device input and output controllers forward shared memory addresses to the slave device input and output controllers. The master device transmits this information via two speed-expansion buses: MSBusInputAddrBidi (for ingress addresses) and MSBusOutputAddrBidi (for egress addresses).

3.4 Packet Format According to Packet Type

3.4.1 Packet Signaling

The master LU carries the packet header, which contains a packet qualifier byte (the first byte, H0) and one to four bytes (H1 to H4) of additional information. Depending on the packet type, the packet qualifier byte may contain information about:

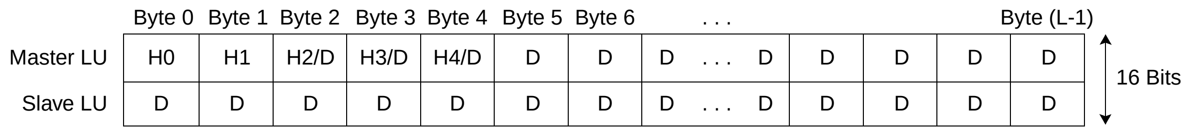
- Packet type
- Packet priority
- Packet color (for switchover support)
- Packet filtering information (for switchover support)
- Best-effort flag
- Flywheel counters
- Extended bitmap
- Header parity

The packet qualifier byte may also include reserved bits. Reserved bits pass through the device unmodified. Unless otherwise specified, reserved bits must be set to ‘0’.

3.4.2 Data/Control Packet Format

Figure 3-6 presents the format of a 2-Gbps data packet or control packet. *Tables 3-2* and *3-3* on page 30 describe the packet qualifier byte.

Figure 3-6. 2-Gbps Data/Control Packet Format



	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	. . .					Byte (L-1)
Master LU	H0	H1	H2/D	H3/D	H4/D	D	D	D . . . D	D	D	D	D	D
Slave LU	D	D	D	D	D	D	D	D . . . D	D	D	D	D	D

Notes:
1. H0 through H4 are packet header bytes. The number of packet header bytes depends on the device configuration.
2. D is user data.

Table 3-2. Data/Control Packet, Byte H0

Packet Type	Encoding per Bit Position					
	0	1	2:3	4	5	6:7
	Extended Bitmap	Parity	Protection	Best Effort	Reserved	Packet Priority
Red compliment data/control packet	0 or 1	0 or 1	01	0 or 1	0 or 1	00, 01, 10, or 11
Red direct data/control packet	0 or 1	0 or 1	10	0 or 1	0 or 1	00, 01, 10, or 11
Blue data/control packet	0 or 1	0 or 1	11	0 or 1	0 or 1	00, 01, 10, or 11

Table 3-3. Data/Control Packet, Byte H0 Field Descriptions

Bit(s)	Field Name	Description
0	Extended Bitmap	When enabled, designates the range of output ports addressed by the destination bitmap in the data packet header (see *Section 3.4.2.2 Extended Bitmap* on page 32). When enabled for a 16-port device: 0 The data packet header addresses ports 0 to 7 with a one-byte bitmap. 1 The data packet header addresses ports 8 to 15 with a one-byte bitmap. When enabled for a 32-port device: 0 The data packet header addresses ports 0 to 15 with a two-byte bitmap. 1 The data packet header addresses ports 16 to 31 with a two-byte bitmap. This function is enabled with the extended bitmap enable bit in the *Configuration 0 Register* (page 85). When the extended bitmap function is enabled, the extended flywheel function is also enabled (see *Section 3.4.2.3 Extended Flywheel* on page 32).
1	Parity	Even parity calculated on the entire packet header (H0 to H4, depending on the header length), including reserved bits. Parity calculation always includes the number of bytes defined by the header length, even if there is no information in a header byte. The parity bit is used to ensure that the packet header is valid.
2:3	Protection	Specifies the traffic type (red data packet, blue data packet, or idle packet) and the application method of the bitmap filter to the ingress packet destination bitmap (header bytes H1 through H4). The bitmap filter is specified with the *Bitmap Filter Register* (page 101). The PowerPRS 64G uses the resulting masked destination bitmap to route the packet. If the resulting bitmap is all zeros, the PowerPRS 64G ignores the packet. Note that control packets are detected before the bitmap filter is applied. Protection Field Color Bitmap Filter 00 Not applicable Not applicable (packet is an idle packet or a service packet). 01 Red (backup) Packet destination bitmap is bitwise ANDed with bitwise complement of bitmap filter. 10 Red (active) Packet destination bitmap is bitwise ANDed with bitmap filter. 11 Blue (unfiltered) Packet destination bitmap is used as is (no filter). This filtering function supports switchover and load balancing (see *Section 3.10.1 Switchover Mechanism* on page 49).
4	Best Effort	Flags packets as best-effort bandwidth traffic when the best-effort discard function is enabled. 1 The packet is flagged as best-effort bandwidth traffic, and the PowerPRS 64G can discard it, if necessary. 0 The packet is flagged as guaranteed bandwidth traffic, and the PowerPRS 64G cannot discard it. When the best-effort discard function is enabled, excess output port congestion triggers a mechanism that discards best-effort bandwidth traffic to provide output port access to guaranteed bandwidth traffic. The best-effort discard function is enabled with the best-effort discard enable bit in the *Configuration 0 Register* (page 85). See *Section 3.5.6 Best-Effort Discard* on page 38 for more information.

Table 3-3. Data/Control Packet, Byte H0 Field Descriptions

Bit(s)	Field Name	Description
5	Reserved	Reserved.
6:7	Packet Priority	Specifies the packet priority for data packets: 00 Priority 0 (highest priority) 01 Priority 1 10 Priority 2 11 Priority 3 (lowest priority) Data packets with a high priority are always transmitted before those with a lower priority. However, the credit table makes it possible to guarantee minimum bandwidth for low-priority packets (see *Section 3.6.3 Credit Table* on page 42). Service packets and control packets are transmitted before data packets, regardless of data packet priority.

For data packets, the extended bitmap bit, protection field, best-effort bit, and packet priority field of the packet qualifier byte (H0) are always transmitted without modification. The extended bitmap bit has an effect only when the extended bitmap function is enabled (see *Section 3.4.2.2 Extended Bitmap* on page 32). The best-effort bit has an effect only when the best-effort discard function is enabled (see *Section 3.5.6 Best-Effort Discard* on page 38). The protection field and packet priority field are always processed by the PowerPRS 64G, and the user must set them to the appropriate values.

For ingress data packets, packet header bytes H1 through H4 contain the packet destination bitmap; that is, they designate the output ports to which the packet is destined and the output queues into which the packet will be enqueued. Byte H1 maps ports 0 to 7, byte H2 maps ports 8 through 15, byte H3 maps ports 16 through 23, and byte H4 maps ports 24 through 31 (see *Table 3-4.)*

Table 3-4. Data/Control Packet, Bytes H1 through H4

Header Byte	Output Port Mapped by Header Bit							
	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
H1	Port 0	Port 1	Port 2	Port 3	Port 4	Port 5	Port 6	Port 7
H2	Port 8	Port 9	Port 10	Port 11	Port 12	Port 13	Port 14	Port 15
H3	Port 16	Port 17	Port 18	Port 19	Port 20	Port 21	Port 22	Port 23
H4	Port 24	Port 25	Port 26	Port 27	Port 28	Port 29	Port 30	Port 31
Note: This bitmap applies when the extended bitmap function is disabled.								

Note: For the PowerPRS 64G, the destination bitmap is a logical bitmap. Each logical port can be mapped to any physical port using the *Bitmap Mapping Register* (page 107). For example, logical bitmap 0 can address physical port 1 (rather than physical port 0).

For ingress and egress control packets, packet header bytes H1 through H4 contain only zeros.

3.4.2.1 Output Queue Grants

For egress data packets, the content of bytes H1 through H4 depends on whether the output queue grant function is enabled. When this function is enabled, bytes H1 through H4 of egress data packets and idle packets contain the output queue grants used for ingress flow control (see *Section 3.5.1 Output Queue Grants* on page 37). The output controller inserts the output queue grants into the packet header. Output queue mapping in these four bytes follows the same pattern as output port mapping in ingress data packets (see *Table 3-4).* When the output queue grant function is disabled, the content of bytes H1 through H4 (that is, the destination bitmap) remains unchanged as data packets move through the device.

The output queue grant function is enabled or disabled by setting the output queue grant insertion enable bit in the *Configuration 1 Register* (page 87). In normal operation, the output queue grant function is enabled, and egress data packet and idle packet header bytes H1 through H4 contain the output queue grants. The output queue grants indicate the status of the output queues and control whether the devices attached to the input ports can transmit packets to the PowerPRS 64G. This information is carried simultaneously for all the output queues of a single priority. Consecutive packets carry a different priority, cycling from highest to lowest priority. For example, when two priorities are enabled, two packets are necessary to transmit complete output queue grant information. When all four priorities are enabled, the complete output queue grant information is transmitted in a four-packet cycle.

If the three thresholds enable bit is set in the *Configuration 1 Register,* the cycle is reduced to three packets, and output queue grant information is only transmitted for priorities 0, 1, and 2. Once the grants for priority 2 are transmitted, the grants for priority 0 are transmitted in the next packet. Compared to four priorities, this reduces the output queue grant information update time. When the three thresholds function is enabled, the output queue thresholds for priorities 2 and 3 must be set to the same value and the shared memory thresholds for priorities 2 and 3 must be set to the same value.

3.4.2.2 Extended Bitmap

In normal PowerPRS 64G operation, every data packet header addresses every port. With the extended bitmap function enabled, each data packet header addresses only half the ports. This reduces the destination bitmap field length and increases the packet payload size. When the extended bitmap function is enabled, the bitmap addresses 16 ports within a one-byte field in each of two packets or 32 ports within a two-byte field in each of two packets (when disabled, twice as many bytes per packet are required to address all the ports).

When the extended bitmap function is enabled, only half the ports (either the low-numbered half or the high-numbered half) are addressed in a single packet cycle. Broadcast operation requires two packet cycles, and multicast operation requires either one or two packet cycles depending on whether the active bits are distributed over half the ports or all the ports.

The extended bitmap function is enabled with the extended bitmap enable bit in the *Configuration 0 Register* (page 85). Once enabled, the extended bitmap bit in the data packet qualifier byte (H0) designates the range of ports addressed by that header. Note that when the extended bitmap function is enabled, the extended flywheel function is also enabled.

3.4.2.3 Extended Flywheel

An egress data packet or idle packet header normally contains the output queue grants for a single priority and all the ports. With the extended flywheel function enabled, each packet header contains the output queue grants for a single priority for only half of the ports (either the low-numbered half or the high-numbered half) in a single packet cycle. This reduces the bitmap field length and increases the packet payload size.

The extended flywheel function is enabled when the extended bitmap function is enabled. When the extended flywheel function is enabled along with the output queue grant function, the PowerPRS 64G provides output queue grants for 16 ports within a one-byte bitmap field in each of two packets or 32 ports within a two-byte bitmap field in each of two packets (when disabled, twice as many bytes per packet are required to provide output queue grants for all the ports).

3.4.3 Idle Packet Format

Figure 3-7 presents the format of an idle packet. *Tables 3-5* and *3-6* (page 34) describe the packet qualifier byte. In egress idle packets, the content of header bytes H1 through H4 depends on whether the output queue grant function is enabled. When this function is enabled, egress idle packet header bytes contain the output queue grants used for ingress flow control (see *Section 3.5.1 Output Queue Grants* on page 37) and synchronization (see *Section 3.4.3.1 Flywheel Counters* on page 34). In egress and ingress idle packets, a one-byte side communication channel contains communications to and from the attached devices. The remaining bytes, except for the trailer (that is, the last byte of each LU), are filled with x‘CC’ characters.

Figure 3-7. Idle Packet Format

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	. . .				Byte (L-1)	
Master LU	H0	H1	H2/CC	H3/CC	H4/CC	SCC	CC	CC . . . CC	CC	CC	CC	T	16 Bits
Slave LU	CC	CC	CC	CC	CC	CC	CC	CC . . . CC	CC	CC	CC	T	

Notes:
1. H0 through H4 are packet header bytes. The number of packet header bytes depends on the device configuration.
2. SCC is side communication channel data. Bits 4:7 are copies of bits 0:3.
3. CC denotes x‘CC’ characters. T is a trailer.

Table 3-5. Idle Packet, Byte H0

Packet Type	Encoding per Bit Position				
	0	1	2:3	4:5	6:7
	Extended Flywheel	Parity	Protection	Color	Grant Priority
Blue idle packet	0 or 1	0 or 1	00	00	00, 01, 10, or 11
Red idle packet	0 or 1	0 or 1	00	01	00, 01, 10, or 11

Table 3-6. Idle Packet, Byte H0 Field Descriptions

Bit(s)	Field Name	Description
0	Extended Flywheel	For egress idle packets, when the output queue grant function (see *Section 3.4.2.1 Output Queue Grants* on page 31) and extended flywheel function (see *Section 3.4.2.3 Extended Flywheel* on page 32) are enabled, designates the ports for which the idle packet is carrying output queue grants. When enabled for a 16-port device: 0 The idle packet header provides output queue grants for ports 0 to 7 with a one-byte bitmap. 1 The idle packet header provides output queue grants for ports 8 to 15 for a one-byte bitmap. When enabled for a 32-port device: 0 The idle packet header provides output queue grants for ports 0 to 15 with a two-byte bitmap. 1 The idle packet header provides output queue grants for ports 16 to 31 with a two-byte bitmap. The extended flywheel function is enabled when the extended bitmap function is enabled. The extended bitmap function is enabled with the extended bitmap enable bit in the *Configuration 0 Register* (page 85).
1	Parity	Even parity calculated on the entire packet header (H0 to H4, depending on the header length), including reserved bits. Parity calculation always includes the number of bytes defined by the header length, even if there is no information in a header byte. The parity bit is used to ensure that the packet header is valid, and ignores any additional information carried in the idle packet.
2:3	Protection	Set to ‘00’ for idle packets.
4:5	Color	Identifies the idle packet color for switchover support: 00 Blue 01 Red Others Reserved
6:7	Grant Priority	For egress idle packets, when the output queue grant function is enabled (see *Section 3.4.2.1 Output Queue Grants*), indicates the priority of the output queue grants carried by the idle packet: 00 Priority 0 (highest priority) 01 Priority 1 10 Priority 2 11 Priority 3 (lowest priority) This field is used to synchronize the attached device output queue flywheel counter. For ingress packets, the grant priority field is ignored (its value is ‘xx’).

3.4.3.1 Flywheel Counters

Output queue grant transmission requires one packet cycle *per priority.* For each port, an internal flywheel counter determines the output queue grant priority carried by each egress packet cycle. The grant priority field of the egress idle packet qualifier byte includes the output queue grant priority determined by the flywheel counter (that is, the flywheel counter value). The attached device uses the grant priority field to synchronize its own internal flywheel counter to the PowerPRS 64G internal counter. Because complete output queue grant transmission requires a fixed number of consecutive packet cycles, continuous synchronization is not required. Synchronization occurs only upon transmission of egress idle packets, which carry the output queue grant priority in the packet qualifier byte (see *Table 3-7* on page 35 for an example). The number of priorities for which the PowerPRS 64G transmits output queue grants is set in the priority enable field of the *Configuration 1 Register* (page 87).

Table 3-7. Example of Egress Idle Packets Carrying the Output Queue Grant Priority

Flywheel Counter Value/ Output Queue Grant Priority	3	0	1	2	3	0	1
Packet Type	Data	Idle (grant priority field = ‘00’)	Data	Data	Idle (grant priority field = ‘11’)	Data	Data

3.4.3.2 CRC Checksum

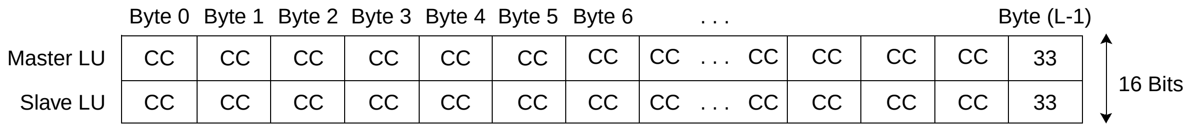
The trailer byte of each ingress and egress idle packet contains an eight-bit cyclic redundancy check (CRC) checksum value calculated over all the bytes sent over a stream since the last idle packet trailer byte. Ingress data packet and control packet trailers also contain user data. Depending on how the device is programmed, the CRC checksum value may also be calculated on ingress service packets to insure compatibility with the IBM Packet Routing Switch PRS28.4G.

CRC encoding is defined by the generating polynomial $X^8 + X^4 + X^3 + X^2 + 1$. The initialization value for CRC checksum calculation is programmable via the eight-bit trailer CRC initialization field in the *Configuration 1 Register* (page 87). This value is set so that the resulting CRC trailer of a synchronization packet is equal to x‘33’ for all LU lengths. When an idle packet is received, the CRC is verified. If a CRC error is detected, the error, if not masked, is reported via a CRC error interrupt. Regardless of the mask condition, the CRC error bit is set in the *Status Register* (page 59), and the port is identified via the *CRC Error Register* (page 92). The number of CRC errors for all the ports is reported in the trailer CRC error count field in the *Error Counter Register* (page 93).

3.4.4 Synchronization Packet Format

Figure 3-8 presents the format of a synchronization packet, and *Table 3-8* describes the packet qualifier byte. The synchronization packet trailer byte, always x‘33’, guarantees the synchronization pattern. All the other bytes in a synchronization packet are filled with x‘CC’ characters.

Figure 3-8. Synchronization Packet Format



	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	. . .				Byte (L-1)
Master LU	CC	CC	CC	CC	CC	CC	CC	CC . . . CC	CC	CC	CC	33
Slave LU	CC	CC	CC	CC	CC	CC	CC	CC . . . CC	CC	CC	CC	33

Notes:
1. CC denotes x‘CC’ characters.
2. The last byte of each LU is a trailer, which is always x‘33’.

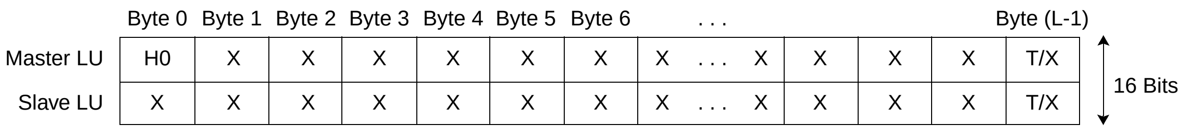
Table 3-8. Synchronization Packet, Byte H0

Packet Type	Encoding per Bit Position							
	0	1	2	3	4	5	6	7
Synchronization packet	1	1	0	0	1	1	0	0

3.4.5 Service Packet Format

Figure 3-9 presents the format of a service packet, and *Table 3-9* describes the packet qualifier byte. The parity is checked on the header length. For ingress service packets, the trailer byte is filled with either a trailer CRC (see *Section 3.4.3.2 CRC Checksum* on page 35) or any character, depending on if the option to process yellow packets as data packets is set in the *Configuration 1 Register* (page 87). All the other bytes are filled with any character. For egress service packets, all but the packet qualifier byte are filled with any character.

Figure 3-9. Service Packet Format



	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	. . .				Byte (L-1)
Master LU	H0	X	X	X	X	X	X	X . . . X	X	X	X	T/X
Slave LU	X	X	X	X	X	X	X	X . . . X	X	X	X	T/X

Notes:
1. H0 is the packet header byte.
2. X means that the byte is filled with any character.
3. The last byte of each LU is a trailer, which is filled with a trailer CRC or any character (depending on the settings).

Table 3-9. Ingress Service Packet, Byte H0

Packet Type	Encoding per Bit Position							
	0	1 (parity)	2	3	4	5	6	7
Yellow type 1 service packet	0	0 or 1	0	0	1	0	0 or 1	0 or 1
Yellow type 2 service packet	0	0 or 1	0	0	1	1	0 or 1	0 or 1
Yellow type 3 service packet	1	0 or 1	0	0	1	0	0 or 1	0 or 1

3.5 Ingress Flow Control

Ingress flow to the PowerPRS 64G is controlled by a variety of mechanisms, primarily:

- Output queue grants that reflect the output queue occupancy. There is one output queue grant per output and per priority.
- Memory grants that reflect the shared memory occupancy. There is one memory grant per priority.

The PowerPRS 64G issues output queue grants and memory grants to the ingress side of the attached devices. These grants allow the devices to transmit packets to the PowerPRS 64G. Ingress flow is also controlled by receive grants, the flow control latency function, and the best-effort discard function. Each of these flow control mechanisms is discussed below.

Note: An attached device can transmit a *unicast* packet to the PowerPRS 64G only when it has received both the memory grant and the output queue grant for the destination output. An attached device can transmit a *multicast* packet to the PowerPRS 64G when it has received a memory grant (output queue grants are not required).

3.5.1 Output Queue Grants

Output queue grants:

- Prevent packets of a single priority destined for the same output from filling the shared memory.
- Prevent low-priority packets from occupying too much shared memory.

The PowerPRS 64G issues an output queue grant for an output priority when the total number of packets in the output queue is below the output queue threshold for that priority. The PowerPRS 64G removes the grant when the number of packets is equal to or greater than the threshold (there is no hysteresis). The four programmable output queue thresholds, one for each output priority, are accessed through the *Threshold Access Register* (page 102). As described in *Section 3.4* on page 29, the PowerPRS 64G inserts the output queue grants into the egress data packet and idle packet headers by enabling the output queue grant insertion enable bit in the *Configuration 1 Register* (page 87).

To generate output queue grants, the PowerPRS 64G continuously compares the total packet count in each output queue to the four output queue thresholds. However, the PowerPRS 64G output queue grant table is refreshed only once per packet cycle. This guarantees that all the attached devices receive the same flow control information.

3.5.2 Memory Grants

The PowerPRS 64G issues a memory grant for a priority when the total number of packets in the shared memory is below the shared memory threshold for that priority. The PowerPRS 64G removes the grant when the number of packets is equal to or greater than the threshold (there is no hysteresis). The four programmable shared memory thresholds, one for each priority, are accessed through the *Threshold Access Register.* The memory grants are provided on the MemoryGrantOut pins.

To generate memory grants, the PowerPRS 64G continuously compares the total packet count to the four shared memory thresholds. The memory grants are refreshed every 32 ns so that all the attached devices receive the same flow control information.

Programming the Shared Memory Thresholds

The shared memory threshold for priority 0 should be programmed to:

$$NumberOfPackets - 64 - (32 \times MemoryGrantLatency) - ControlPacketReception - ControlPacketTransmission$$

where:

NumberOfPackets = total packet storage available in the shared memory (either 512, 1024, or 2048, depending on the speed expansion configuration; see *Table 2-1* on page 20)

64 = the addresses that the input controllers always hold in reserve

MemoryGrantLatency = transmission and processing time of the memory grant by the attached device plus one, calculated in LUs

ControlPacketReception = number of locations reserved for control packet reception, which is 32

ControlPacketTransmission = 1, for the location reserved for the local processor to send a control packet

3.5.3 Shared Memory Overrun

The PowerPRS 64G receives any incoming packet that has a store address, regardless of output queue and shared memory occupancy (unless the flow control latency function requires packet discard; see *Section 3.5.5 Flow Control Latency).* If a packet is received when the input controller does not have an available store address, the input controller discards the packet. This condition is reported in the no address interrupt bit in the *Status Register* (page 59). An interrupt is raised if the condition is not masked with the no address interrupt bit in the *Interrupt Mask Register* (page 65). This error occurs only if the shared memory thresholds are programmed incorrectly, or if the attached device is not responding to the memory grant information.

3.5.4 Receive Grants

The receive grant function permits filtering of incoming packets based on their destination. Thirty-two ReceiveGrantIn bits filter the destination bitmaps of incoming packets:

- When ReceiveGrantIn(i) is high, incoming packets can be placed in output queue *i.*
- When ReceiveGrantIn(i) is low, incoming packets cannot be placed in output queue *i,* regardless of their destination bitmaps.

For simple switching systems, the receive grant function allows switchover without packet loss. It is an extension of the bitmap filter (ReceiveGrantIn pins are ANDed with the bitmap filter), only on device-level pins, and has the exact same effect. When using the receive grant function:

- The color mechanism and the bitmap filter function should be disabled.
- The *Bitmap Filter Register* (page 101) must be set to x‘FFFF'.
- Only data packets with the packet qualifier byte (H0) protection field set to ‘10’ can be transmitted.

This function is enabled with the receive grant enable bit in the *Configuration 1 Register* (page 87). For more information about the bitmap filter, see *Table 3-3* on page 30.

3.5.5 Flow Control Latency

The input controller flow control latency function detects operational errors with the attached devices. This function is enabled with the flow control latency field in the *Configuration 0 Register* (page 85). When the flow control latency function is enabled, the input controller checks whether an incoming unicast packet is destined to an output for which neither an output queue grant nor a memory grant has been issued in the past *n* packet cycles. For a multicast packet, the input controller checks only the memory grant information. If the packet is destined for an output for which no grants have been issued, the packet is discarded. This error is reported via the flow control violation bit in the *Status Register* and, unless masked in the *Interrupt Mask Register,* the error generates a flow control violation interrupt. The violating ports are identified by the corresponding bits in the *Flow Control Violation Register* (page 94).

3.5.6 Best-Effort Discard

In some applications, certain low-priority traffic is more important than the high-priority traffic that monopolizes the output port and prevents low-priority traffic from accessing the port. The best-effort discard function attempts to correct this situation by categorizing incoming traffic as either “guaranteed bandwidth” or “best-effort bandwidth.” Best-effort bandwidth traffic is discarded at the input controllers, when necessary, to provide output port access to guaranteed traffic. The best-effort discard function helps to ensure guaranteed bandwidth traffic quality of service. This function is enabled with the best-effort discard enable bit in the *Configuration 0 Register.*

When the PowerPRS 64G is operating as a lossy switch, the best-effort discard flow control function is activated as soon as the aggregate traffic load for an output port exceeds its capacity for a given time period. The PowerPRS 64G discards only traffic flagged as best-effort; guaranteed bandwidth traffic is never discarded. Guaranteed bandwidth traffic congestion is managed through the normal flow control mechanism. Best-effort discard from the shared memory is completed in a single burst to minimize the number of affected packets.

3.5.6.1 Best-Effort Discard Counters

Operation of the lossy switch is based on a set of five counters (see *Table 3-10)* per output port that are incremented when the port's output queues receive specific packet types.

Table 3-10. Best-Effort Discard Counters

Counter	Description
Counter 1	The main counter. Counts each incoming packet.
Counter 2	Counts each incoming packet that is either guaranteed bandwidth of any priority or best-effort bandwidth of priority 0, 1, or 2.
Counter 3	Counts each incoming packet that is either guaranteed bandwidth of any priority or best-effort bandwidth of priority 0 or 1.
Counter 4	Counts each incoming packet that is either guaranteed bandwidth of any priority or best-effort bandwidth of priority 0.
Counter 5	Counts each incoming packet that is guaranteed bandwidth of any priority.

Best-effort discard counters are decremented at a packet speed equivalent to the throughput of the port (the line rate is provided by a register defined at port initialization). The counters are accessible via the *Best-Effort Resources Access Register* (page 108).

3.5.6.2 Best-Effort Discard Thresholds

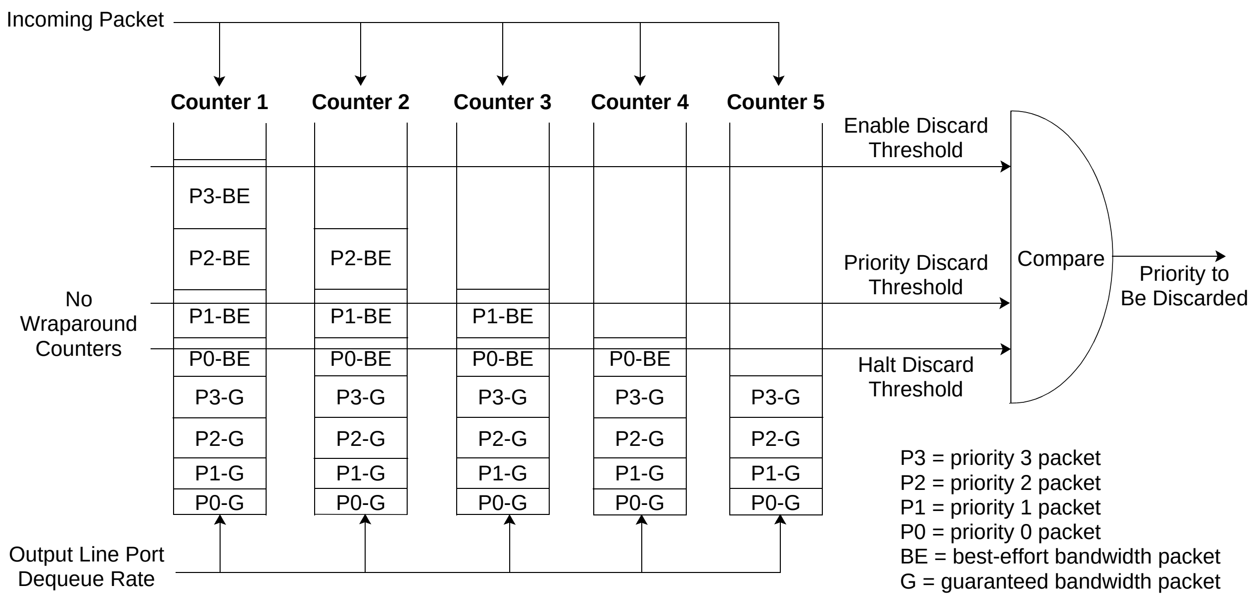
Three thresholds govern the best-effort discard process:

- *Enable discard threshold.* When counter 1 reaches this threshold, the discard process starts.
- *Halt discard threshold.* When counter 1 reaches this threshold, the discard process stops.
- *Priority discard threshold.* This threshold, in comparison to counters 1, 2, 3, and 4, determines which packets are discarded:
  - When counter 1 is above the priority discard threshold, best-effort packets of priority 3 are discarded.
  - When counter 2 is above the priority discard threshold, best-effort packets of priority 2 are discarded.
  - When counter 3 is above the priority discard threshold, best-effort packets of priority 1 are discarded.
  - When counter 4 is above the priority discard threshold, best-effort packets of priority 0 are discarded.

The priority discard threshold only determines which packets are discarded; the halt discard threshold determines when the discard process stops. The best-effort discard thresholds are accessed via the *Best-Effort Resources Access Register.*

Figure 3-10 on page 40 illustrates how the best-effort discard thresholds operate. In the figure, best-effort discard starts because counter 1 for the output port exceeds the enable discard threshold. Counters 1, 2, and 3 are all above the priority discard threshold and, consequently, the best-effort packets of priority 1, 2, and 3 destined for that output port are discarded at the input controller. Discard continues until counter 1 falls below the halt discard threshold.

Figure 3-10. Best-Effort Discard Counters and Thresholds



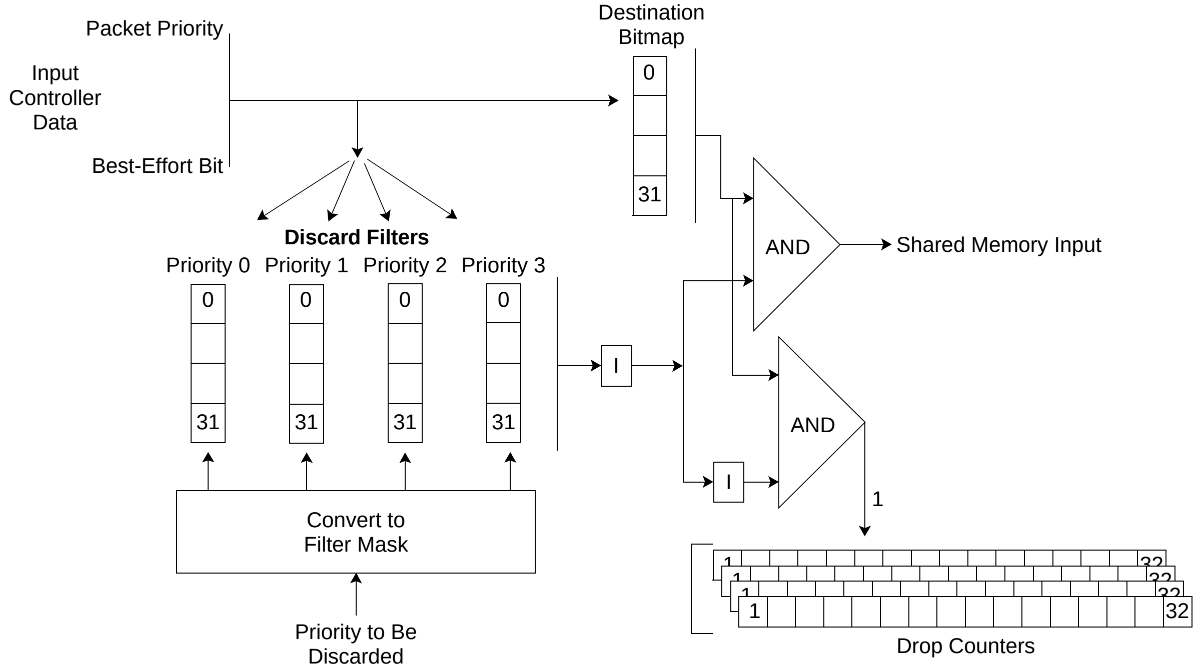
The assumption is that guaranteed bandwidth should be engineered so that it never exceeds the priority discard threshold—it dequeues traffic without the need for flow control. However, situations exist in which the traffic pattern changes before the halt discard threshold is reached. In this case, if another counter crosses the priority discard threshold, the input controllers will discard additional packets.

3.5.6.3 Best-Effort Discard Filters

Best-effort discard occurs within the input controllers, which filter the destination bitmaps of incoming packets (see *Figure 3-11* on page 41). There is one best-effort discard filter per priority. The best-effort discard filter is used as a destination bitmap mask, with the discard set to ‘0’. For each combination of destination output port and priority, there is a 20-bit counter that provides discard quantity and rate information. These drop counters are enabled with the best-effort drop counters enable bit in the *Configuration 0 Register* (page 85) and are accessible via the *Best-Effort Resources Access Register* (page 108).

When an incoming packet arrives at the PowerPRS 64G, the packet priority field, best-effort bit, and destination bitmap are provided to the best-effort filter logic. If the packet is read as best-effort, the best-effort discard filter for the packet priority is applied to the destination bitmap. If the combination is ‘0’, the packet is discarded rather than enqueued, and the corresponding drop counter (or counters for a multicast packet) is updated.

Figure 3-11. Best-Effort Discard Filters



3.6 Egress Flow Control

Egress flow from the PowerPRS 64G is controlled by a variety of mechanisms, including send grants, the send grant antistreaming function, and the credit table.

3.6.1 Send Grants

Egress flow from the PowerPRS 64G to an attached device is controlled primarily by a grant mechanism. The PowerPRS 64G transmits a packet on an output when the attached device issues a send grant for that output. When the send grant is removed, the PowerPRS 64G generates idle packets. Send grants are provided on the SendGrantIn pins. In normal operation, the attached device issues send grants without packet priority restrictions.

In send-grant-per-priority mode, the attached device determines which packet priority is allowed to exit the PowerPRS 64G and issues send grants for packets of that priority. This mode is enabled with the *Send Grant per Priority Register* (page 83).

In send-grant-per-priority mode, the output SendGrantIn pins are multiplexed over one line according to priority. The serial information is timed by nine 16-ns clock intervals derived from half the LU-byte clock. The serial information begins with a unique packet header followed by the priority bits:

0	0	0	0	1	P0	P1	P2	P3

The ‘00001’ pattern cannot be reproduced by any combination of the SendGrantIn signals. Phase alignment and pattern delineation are performed by edge detection on the first ‘1’ following four ‘0’s by oversampling the SendGrantIn line. Alignment is performed for every new set of SendGrantIn values, which ignores long-term clock phase jitter. This bit pattern can be padded by ‘0’s if its length needs to be adjusted (for example, to the LU size).

Note: In send-grant-per-priority mode, control packets and service packets are sent to an output port if the send grant for that output is active for at least one priority.

3.6.2 Send Grant Antistreaming

The optional send grant antistreaming function is intended to prevent a defective attached device from indefinitely removing the send grant. This function is enabled with the send grant antistreaming enable bit in the *Configuration 0 Register* (page 85). If a send grant is not issued for any priority for a given number of contiguous packet cycles, the antistreaming function (if enabled) internally forces the send grant to active for all priorities until the attached device issues another send grant. The number of contiguous packet cycles (from 16 to 2048) is programmable using the send grant antistreaming threshold field in the *Configuration 0 Register*. When the number of packet cycles is exceeded, a violation is reported via the send grant violation bit in the *Status Register* (page 59), and an interrupt is raised if the violation is not masked with the send grant violation bit in the *Interrupt Mask Register* (page 65). The port with the violation is reported in the *Send Grant Violation Register* (page 97).

When the send grant antistreaming function is enabled, the output queue read manager also considers the send grant active and sets an internal send grant to avoid congestion inside the PowerPRS 64G. The internal send grant is removed when the attached device resumes the normal send grant.

3.6.3 Credit Table

In the basic PowerPRS 64G configuration, the output queue read manager schedules packet transmission for each output port in the following order:

1. Service packets
2. Control packets
3. Priority 0 data packets
4. Priority 1 data packets
5. Priority 2 data packets
6. Priority 3 data packets

The credit table provides a weighted cycling mechanism that can be programmed to guarantee minimum bandwidth for low-priority packets. For each output port, the credit table indicates the packet priority to be transmitted during each packet cycle. The credit table includes 256 entries, or credits, per port. The credit table is read once per packet cycle; that is, the priority entry is read for the current credit number, which generates a credit for that priority. The credit number is incremented by one for every packet cycle. After credit number 255, the credit number returns to 0.

When a credit is generated for a priority, a packet of that priority is sent on the output port if there is a packet in the output queue and the send grant is active for that priority. If either the output queue is empty or the send grant is inactive, the basic algorithm applies.

Use of the credit table is specified by the credit table enable bit in the *Configuration 1 Register* (page 87). Indirect access to the credit table is provided via the *Credit Table Access Register* (page 104).

3.7 Packet Reception

Packets are received on an input port asynchronously with packets received on other input ports. When a packet arrives at an input port, the input controller analyzes the packet header parity and takes the following actions:

- If the header parity is incorrect, the input controller discards the entire packet. The error is reported via the header parity error bit in the *Status Register* (page 59), the affected port is identified in the *Header Parity Error Register* (page 93), and the header parity error count field is incremented in the *Error Counter Register* (page 93). The header parity error generates an interrupt unless it is masked with the header parity error bit in the *Interrupt Mask Register* (page 65).
- If the header parity is correct, the input controller analyzes the packet type and extracts the flow control information. If the packet matches the color specified in the color select field in the *Command Register* (page 100), packet reception is reported in the *Color Packet Received Register* (page 96). Further action depends on the packet type and is discussed below.

When two devices are configured for speed expansion, the input controller on the master device conducts the packet header analysis and extraction, then forwards the packet control information to the input controller on the slave device.

Note: A multicast packet is stored only once in the shared memory, but its shared memory address is enqueued in each output queue indicated by its destination bitmap. Each output port transmits a multicast packet according to the first-in-first-out structure of its output queue; consequently, a multicast packet is not necessarily transmitted at the same time on every port. A multicast packet has only one priority that applies to all its destinations.

3.7.1 Idle Packet Reception

When an idle packet is received on an input port, the input controller verifies the CRC in the trailer byte. If a CRC error is detected and not masked, the error is reported via a CRC error interrupt. Also, regardless of the mask condition, the CRC error bit is set in the *Status Register* and the port is identified in the *CRC Error Register* (page 92). The number of CRC errors for all the ports is reported in the trailer CRC error count field in the *Error Counter Register.* When two devices are configured for speed expansion, both the master and the slave device verify the CRC in the trailer byte.

3.7.2 Data Packet Reception

The input controllers discard an ingress data packet when:

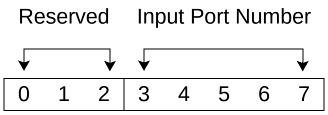
- All packet destination output ports are disabled.
- The data packet is received when the input controller does not have an available store address, and the packet cannot be stored in the shared memory. This flow control error occurs only if the shared memory thresholds are programmed incorrectly or if the attached device is not responding to the memory grant information. This error is reported via the no address interrupt bit in the *Status Register* and generates an interrupt unless it is masked with the no address interrupt bit in the *Interrupt Mask Register.*
- Required output queue grants or memory grants have not been issued. This flow control error occurs only if the attached device does not follow the ingress flow control information. This error is reported via the flow control violation bit in the *Status Register* and generates an interrupt unless it is masked with the flow control violation bit in the *Interrupt Mask Register.*

- After packet filtering, the bitmap is all zeros.
- The best-effort discard mechanism within the input controllers is enabled, and a discard condition exists (see *Section 3.5.6.3 Best-Effort Discard Filters* on page 40).

3.7.3 Control Packet Reception

The input controller recognizes an ingress data packet with an all-zero destination bitmap as a control packet. Control packet reception is reported via the control packet received bit in the *Status Register* (page 59) and generates an interrupt to the local processor unless it is masked with the control packet received bit in the *Interrupt Mask Register* (page 65).

The input controller inserts the input port number into five bits of the control packet header byte, H1, as follows:



If the force address insertion bit of the *Configuration 1 Register* (page 87) is set to ‘1’, the input controller also inserts the input port number into ingress data packet headers. This feature is used for testing.

An ingress control packet is stored in the shared memory, and its shared memory address is placed in the control packet queue. The local processor reads the content of the shared memory address according to the process described in *Section 3.7.4 Reading an Ingress Control Packet* on page 45.

A control packet occupies either one or two rows in the shared memory depending on the packet size and the speed expansion configuration (see *Table 3-11).*

Table 3-11. Shared Memory Addresses for Ingress Control Packets

External Speed Expansion	Internal Speed Expansion	Packet Size (bytes)	Addresses to Read in Master and Slave Shared Memory Banks
No	No	32 to 40	A
No	Yes	32 to 40	A
Yes	No	32 to 40	A
No	No	64 to 80	A and A + 1
No	Yes	64 to 80	A and A + 1024
Yes	No	64 to 80	A
Yes	No	128 to 160	A and A + 1
No	Yes	128 to 160	A and A + 1 A + 1024 and A + 1025
Yes	Yes	128 to 160	A and A + 1024
Yes	Yes	256 to 320	A and A + 1 A + 1024 and A + 1025

The control packet queue contains a maximum of 32 packets. If the queue is full when a control packet arrives, the packet is discarded. This event is reported via the control packet discard bit in the *Status Register* (page 59) and generates an interrupt unless it is masked with the control packet discard bit in the *Interrupt Mask Register* (page 65). The *Control Packet Counter Register* (page 94) provides the number of control packets currently enqueued.

Note: When two devices are configured for speed expansion, the input controller on the master device conducts the control packet reception activities, and only the master device generates control packet received interrupts.

3.7.4 Reading an Ingress Control Packet

When accessing the shared memory to read a control packet in a speed expansion configuration, the sequencer guarantees master and slave device shared memory access time of up to one LU length for LUs of lengths other than 16 and 32 bytes. For LUs of 16 and 32 bytes, the control packet access priority enable bit in the *Configuration 1 Register* (page 87) must be set to ‘1’ to guarantee shared memory access time of up to three LU lengths (the actual access time depends on the latency of the attached device in reacting to the memory grant mechanism). This setting applies only to single device configurations or to the master device in a speed expansion configuration. In a speed expansion configuration, a master operation follows a slave operation as described below, and the slave shared memory access is complete when the master shared memory access is complete.

When the local processor receives a control packet received interrupt, it initiates the following sequence:

1. The local processor issues a “load next control packet address” command via the *Command Register* (page 100). This command loads the shared memory content from the first address in the control packet queue into the internal memory row register. (For more information about the internal memory row register, see *Section 5.4.18 Shared Memory Access Registers* on page 98.)
2. The local processor reads the first row (or only row; see *Table 3-11* on page 44) of the control packet using the *Shared Memory Pointer Register* (page 99) and the *Shared Memory Data Register* (page 99).
3. If necessary, the local processor reads the second row of the control packet using the *Shared Memory Pointer Register* and the *Shared Memory Data Register.* As shown in *Table 3-11,* if the first row occupies shared memory address A, then the second row occupies shared memory address $A + 1$ or $A + 1024$.
4. The local processor issues a “free control packet address” command via the *Command Register* to free the first address in the control packet queue.
5. If the control packet counter value is greater than zero or if another control packet received interrupt has been issued, the local processor repeats steps 1 through 4.

In external speed expansion, the shared memory start address for a control packet is the same in both the slave and the master devices. After the local processor issues the “load next control packet address” command to the master device, the local processor must read the master *Shared Memory Pointer Register* and write its contents to the slave *Shared Memory Pointer Register.* If the control packet access priority enable bit in the *Configuration 1 Register* is set (when the LU size is 16 or 32 bytes and the device is loaded at 100 percent), the local processor must issue a read command to the slave device and then issue a read command to the master device before the control packet data from the slave and master devices can be read. Once the first row of data has been completely read in both the master and slave devices, the same slave-master command sequence is required to read the second row in both devices. This sequence guarantees access time to the slave shared memory. All the rows in both the master and slave devices must be read before the “free control packet address” command is issued to the master device.

Note: When two devices are configured for speed expansion, the “load next control packet address” command and the “free control packet address” command are issued (via the *Command Register* [page 100]) to the master device. These commands are ignored when issued to the slave device.

3.8 Packet Transmission

3.8.1 Output Port Servicing

PowerPRS 64G output ports are designed for continuous packet transmission, and packet transmission starts at a fixed point in time (see *Table 3-12).* If neither a control packet nor a data packet is available for transmission on an output port, the port will transmit an idle packet. Control packets are always transmitted on an output port before any other packets in the shared memory destined for that port, but they do not affect the performance of high-priority traffic. Control packet transmission is relatively infrequent because the local processor access is slow compared to the data packet traffic rate. When two devices are configured for external speed expansion, packet transmission starts at the same time on both the slave and master devices.

Table 3-12. Packet Transmission Time

Output Port Number	8-ns Byte Cycle	Output Port Number	8-ns Byte Cycle
0 (reference), 16	0	8, 24	2
1, 17	4	9, 25	6
2, 18	8	10, 26	10
3, 19	12	11, 27	14
4, 20	1	12, 28	3
5, 21	5	13, 29	7
6, 22	9	14, 30	11
7, 23	13	15, 31	15

3.8.2 Look-Up Tables

Two look-up tables allow the byte transmission sequence of egress packets to be rearranged. One look-up table designates the byte transmission sequence for the master data stream and the other look-up table designates the byte transmission sequence for the slave data stream. Only the first 16 data bytes of each data row of a byte stream can be rearranged. These 16 data bytes correspond to the 16 entries in each look-up table that identify if and when a data byte will be sent and if and when a data byte will be repeated (see *Table 3-13* on page 47 for an example). All the output ports use the same look-up tables. The *Look-Up Table Access Register* (page 106) provides indirect access to these tables.

Table 3-13. Example of Byte Reordering Using Look-Up Tables

Byte Sequence before Ordering	Look-Up Table Entry Sequence	Byte Sequence after Ordering
Byte 0	3	Byte 3
Byte 1	4	Byte 4
Byte 2	5	Byte 5
Byte 3	3	Byte 3
Byte 4	4	Byte 4
Byte 5	5	Byte 5
Byte 6	15	Byte 15
Byte 7	14	Byte 14
Byte 8	13	Byte 13
Byte 9	12	Byte 12
Byte 10	11	Byte 11
Byte 11	10	Byte 10
Byte 12	9	Byte 9
Byte 13	8	Byte 8
Byte 14	7	Byte 7
Byte 15	6	Byte 6
Byte 16	-	Byte 16

3.8.3 Control Packet Transmission

When accessing the shared memory to write a control packet in a speed expansion configuration, the sequencer guarantees master and slave device shared memory access time of up to one LU length for LUs of lengths other than 16 and 32 bytes. For LUs of 16 and 32 bytes, the control packet access priority enable bit in the *Configuration 1 Register* (page 87) must be set to ‘1’ to guarantee shared memory access time of up to three LU lengths (the actual access time depends on the latency of the attached device in reacting to the memory grant mechanism). This setting applies only to single device configurations or to the master device in a speed expansion configuration. In a speed expansion configuration, a master operation follows a slave operation as described below, and the slave shared memory access is complete when the master shared memory access is complete.

Depending on the speed expansion configuration and packet length, either one or two shared memory addresses are reserved for the local processor to write a control packet (see *Table 3-14* on page 48). The local processor has write access to the entire packet memory and controls the transmission of the row content.

Table 3-14. Reserved Shared Memory Addresses for Egress Control Packets

External Speed Expansion	Internal Speed Expansion	Packet Size (number of bytes)	Reserved Shared Memory Address
No	No	32 to 40	0
No	No	64 to 80	0 and 1
No	Yes	64 to 80	0 and 1024
Yes	No	64 to 80	0
Yes	No	128 to 160	0 and 1

Control packets are transmitted sequentially. Note that control packets can start only at shared memory location zero. To transmit a control packet:

1. The local processor builds the first (or only) row of the control packet in the internal memory row register via the *Shared Memory Data Register* (page 99).
2. The local processor issues an internal memory row register “write at shared memory address 0” command via the *Shared Memory Pointer Register* (page 99).
3. If necessary, the local processor repeats steps 1 and 2 for the second row of the control packet. The shared memory address specified in step 2 will be either 1 or 1024.
4. The local processor specifies the output ports from which the control packet must be transmitted by loading the *Control Packet Destination Register* (page 101). This step initiates control packet transmission, if the send grant is active.
5. The local processor waits for an “all control packets transmitted” interrupt in the *Status Register* (page 59). This interrupt occurs when all rows of the control packet have been transmitted.

When two devices are configured for speed expansion, the local processor must write a row to the slave device and then to the master device before transmitting the control packet. This slave-master sequence guarantees access time to the slave shared memory. The local processor transmits the control packet by loading the *Control Packet Destination Register* for the master device.

3.8.4 Idle Packet Transmission

An output port transmits idle packets when neither control packets nor data packets are available, or when the SendGrantIn signal is low (that is, when the attached device has not issued a send grant to the PowerPRS 64G). Idle packet transmission occurs as follows:

- If the idle color force bit in the *Configuration 0 Register* (page 85) is set to ‘1’, the port transmits idle packets of the color specified by the idle color bit in the *Configuration 0 Register.*
- If the idle color force bit in the *Configuration 0 Register* is set to ‘0', the port transmits idle packets of the color specified by the expected color bit in the *Configuration 0 Register,* if both the following conditions have been met:
  - A packet of the same color as the expected color has been received on all active inputs since a color clear command was last sent (via the *Command Register* [page 100]).
  - The corresponding output queue is empty.

  Otherwise, the port transmits idle packets of the opposite color.

3.8.5 Service Packet Transmission

Service packets are generated by the local processor. The local processor builds the service packet in the shared memory and transmits it to the desired outputs as if it were a control packet. Like control packets, egress service packets do not carry a trailer CRC byte. The send grant must be active for service packet transmission.

3.9 Side Communication Channel

A four-bit side communication channel (SCC) allows communication between the attached devices and the PowerPRS 64G. SCC information is transferred in-band in the idle packet master LU (in byte 6, bits 0:3 and 4:7).

On the path from an attached device to the PowerPRS 64G, the attached device writes SCC information into all ingress idle packets. An attached device may generate an idle packet to guarantee that an information change is propagated in a minimum amount of time. When the PowerPRS 64G receives an idle packet, it extracts and compares bits 0:3 and 4:7. If the values are identical, an internal register that contains this information is refreshed, and the information is made available through the read-only *Side Communication Channel Input Reporting Register* (page 109).

On the path from the PowerPRS 64G to the attached devices, the PowerPRS 64G inserts SCC information from four input pins (SCCIn[0:3]) into all egress idle packets. All the output ports send the same SCC information. The PowerPRS 64G automatically generates an idle packet to all the ports as soon as an edge is detected on the SCCIn pins to guarantee that the information change is propagated in a minimum amount of time.

3.10 Switchover Support

3.10.1 Switchover Mechanism

In redundant switch-plane operation, PowerPRS 64G switchover support is provided through a color mechanism. This mechanism conducts scheduled switchovers without packet loss. During normal operation, data packets and idle packets are coded red. Red traffic includes data packets with direct filtering, and link-liveness packets with either direct filtering or reverse filtering according to the mask set in the *Bitmap Filter Register* (page 101). By setting reverse filtering in the packet protection field, attached devices can send link-liveness packets to the ports on the backup switch plane (and thereby supervise the backup path). The PowerPRS 64G registers and bits involved in the switchover process are described in *Table 3-15* on page 50 and in the associated register descriptions in *Section 5.*

Table 3-15. Registers and Bits Used for Switchover Support

Register or Register Field	Description
Idle color force bit in the *Configuration 0 Register* (page 85)	1 Transmits all egress idle packets with the color specified by the idle color bit, regardless of the expected color bit setting. When the color mechanism is not used, this bit must be set to ‘1’. 0 Allows the switchover mechanism to determine the color of egress idle packets by setting the expected color bit. The PowerPRS 64G sends idle packets of the color specified by the expected color bit on output port *n*, if an expected color packet has been received on all active inputs since the color clear command was last issued and output queue *n* is empty. Otherwise, the PowerPRS 64G sends idle packets of the opposite color on output port *n*. The color clear command is sent through the *Command Register* (page 100).
Idle color bit in the *Configuration 0 Register*	Specifies the color assigned to all idle packets when the idle color force bit is set to ‘1’: 0 Blue idle packets 1 Red idle packets In this case, the PowerPRS 64G-generated idle packets will not change color during normal operation.
Expected color bit in the *Configuration 0 Register*	Specifies the expected color of incoming packets after a color clear command is initiated through the *Command Register:* 0 Blue packets 1 Red packets
Color clear bit in the *Command Register* (page 100)	Processed as a command (action is taken on the rising edge) to clear the idle packet color state machine in preparation for packet color-change detection.
Color Detection Disable Register (page 90)	When a bit is active, disables the input port color detection mechanism and sets the corresponding bit in the *Expected Color Received Register.* This mask indicates if an input port is enabled and active during the color-based switchover process.
Expected Color Received Register (page 92)	1 Either the expected color has been received on the input port since the last color clear command or the corresponding bit is set in the *Color Detection Disable Register.* 0 The opposite color is still being received on the input port.
Bitmap Filter Register (page 101)	Specifies the mask applied to the ingress packet destination bitmap for switchover support and load balancing in redundant switch-plane operation. Application of the bitmap filter depends on the packet protection field (bits 2:3) of the packet qualifier byte, H0. The incoming packet bitmap is logically ANDed with either a specified mask (red active packets) or its complement value (red backup packets), or it is left unfiltered (blue packets; see *Table 3-16).* For more information about operating this mask, see *Table 3-3* in *Section 3.4.2 Data/Control Packet Format* on page 29. Note that this register cannot be written to while the device is in standby.

Table 3-16. Ingress Data Packet Protection Field

Protection Field Value	Packet Color	Filtering
01	Red (backup)	Packet destination bitmap is bitwise ANDed with the bitwise complement of the bitmap filter.
10	Red (active)	Packet destination bitmap is bitwise ANDed with the bitmap filter.
11	Blue (unfiltered)	Packet destination bitmap is used unfiltered.

3.10.2 Scheduled Switchover Process

Redundant switch planes operate under one of two conditions:

- When one switch plane is active and the other is a backup. Data traffic flows only through the active switch path.
- When both switch planes are operating under load balancing. Data traffic is split between the two switch planes, which have complementary values in their *Bitmap Filter Registers* (page 101).

When two switch planes are initially operating under load balancing, the switchover process includes three phases:

1. Rerouting traffic to one switch plane
2. Modifying the load-balancing configuration
3. Resuming traffic on both switch planes

The scheduled switchover process for the active/backup initial operating condition is very similar to this one. Minor differences are identified below.

3.10.2.1 Phase 1: Rerouting Traffic to One Switch Plane

Before the scheduled switchover begins, data traffic is routed through both switch planes under the load-balancing configuration specified in the *Bitmap Filter Registers.* During phase 1 of the switchover process, traffic is rerouted so that it flows through only one switch plane. For this discussion, the Y switch plane is dropped and the X switch plane remains active. Phase 1 is initiated on red traffic and is complete when all traffic is blue.

To reroute traffic to one switch plane:

1. Because all current traffic (idle and data) is red, the local processor configures each PowerPRS 64G to detect the color blue by changing the expected color bit and issuing a color clear command.
2. All ingress devices change the packet qualifier byte of their incoming packets to change the color from red (normal traffic) to blue (no filtering). In addition:
    - On the Y path, the ingress devices send all their buffered red data packets to the switch core, and then start generating blue idle packets to the switch core.
    - On the X path, all ingress devices send all their buffered red packets (regardless of priority) to the switch core, and then begin sending their blue data packets to the switch core.

    Simultaneously, the egress devices block data packet reception from the X switch plane by locking their peer buffer (which connects the X and Y paths). This prevents blue traffic reception before red traffic is fully exhausted. (If the switch planes were operating in an active/backup condition, this switch plane would be the backup, and the only traffic would be link-liveness packets.)
3. When at least one blue packet (idle or data) has been received on each active input port of the X or Y switch plane, then all the red data packets have been delivered to that switch plane and all the active input ports will be receiving only blue packets (either idle or data). Each of the two local processors attached to the serial host interface (SHI) may be informed, through polling, that its switch core is detecting only blue packets.

    Note: The *Expected Color Received Register* (page 92) indicates the receipt of a blue packet since the last color clear command on each port that has not been tagged as inactive by the *Color Detection Disable Register* (page 90). The color blue was set by the expected color bit. When all bits are set in the *Expected Color Received Register,* the switch core is detecting only blue packets.

4. On the Y switch plane, when at least one blue packet has been detected on each active input port and output queue *n* is empty, the PowerPRS 64G begins to continuously generate blue (rather than red) idle packets to port *n.* On the X switch plane, when at least one blue packet has been detected on each active input port, the PowerPRS 64G begins generating blue (rather than red) idle packets, as necessary.

   At this point, all egress packets from both switch planes are blue (idle packets or data packets on the X path, and idle packets on the Y path). All data traffic is blue (unfiltered) and carried on the X path.

5. When all active egress devices have detected the arrival of a blue idle packet from the Y switch core and these devices have no more packets to send from their packet buffer queue for that switch plane to the attached traffic manager, then they unlock their peer buffer. This step unblocks traffic transmission from the X switch plane, which will have blue packets waiting for transmission to the attached devices.

   Phase 1 of the switchover is complete for the entire switch fabric. The egress devices may convey this status to their attached processor.

3.10.2.2 Phase 2: Modifying the Load-Balancing Configuration

When phase 1 of the scheduled switchover is complete, all traffic through both switch planes is blue and data traffic flows through only one plane. Both local processors can now safely modify the content of the *Bitmap Filter Register* (page 101) in accordance with the new configuration parameters, which may specify new port assignments for a different load-balancing configuration.

3.10.2.3 Phase 3: Resuming Traffic on Both Switch Planes

Phase 3 of the switchover starts the new load-balancing configuration. This phase is similar to phase 1, except it is initiated on blue traffic and is complete when all traffic is red. For this discussion, the Y switch plane is dropped and the X switch plane remains active. During phase 3, split traffic is resumed on both switch planes.

To resume traffic on both switch planes:

1. Because all current traffic (idle and data) is blue, the local processor configures each PowerPRS 64G to detect the color red by changing the expected color bit and issuing a color clear command.
2. All ingress devices stop changing the packet qualifier byte of their incoming packets so that the packet color remains red. In addition:
   - On the Y path, all ingress devices start generating red (rather than blue) idle packets to the switch core.
   - On the X path, all ingress devices send all their buffered blue packets (regardless of priority) to the switch core, and then begin sending their red data packets to the switch core.

   Simultaneously, the egress devices block data packet reception from the Y switch plane by locking their peer buffer. This step prevents red traffic reception before blue traffic is fully exhausted.
3. When at least one red packet (idle or data) has been received on each active input port of the X or Y switch plane, then all the blue data packets have been delivered to that switch plane and all the active input ports will be receiving only red packets (either idle or data). Each of the two local processors attached to the SHI may be informed, through polling, that its switch core is detecting only red packets.
4. On each switch plane, when at least one red packet has been detected on each active input port, the PowerPRS 64G on that switch plane begins to generate red (rather than blue) idle packets, as necessary.

   At this point, all egress packets from both switch planes are red.

5. When all active egress devices have detected the arrival of a red idle packet from the X switch core and these devices have no more packets to send from their packet buffer queue for that switch plane to the attached traffic manager, then they unlock their peer buffer. This step unblocks traffic transmission from the Y switch plane, which will have red packets waiting for transmission to the attached devices.

   Switchover is complete for the entire switch fabric. The egress devices may convey this status to their attached processor.

3.11 Port Paralleling

In port paralleling, four ports are grouped to form one link (see *Figure 2-9* on page 22). Port paralleling is possible only when the ports are grouped as follows:

- Ports 0, 1, 2, and 3
- Ports 4, 5, 6, and 7
- Ports 8, 9, 10, and 11
- Ports 12, 13, 14, and 15
- Ports 16, 17, 18, and 19
- Ports 20, 21, 22, and 23
- Ports 24, 25, 26, and 27
- Ports 28, 29, 30, and 31

One or more of these groups may be used at a time. The desired groups are specified in the input port paralleling and output port paralleling fields in the *Configuration 0 Register* (page 85).

3.11.1 Packet Processing with Port Paralleling

Within a group of ports configured for port paralleling, each port transmits packets independently. A packet received on a lower-numbered port is processed before a packet received at the same time on a higher-numbered port. For example, when ports 0, 1, 2, and 3 are grouped, a packet received on port 0 is processed (enqueued) before the packets received on ports 1, 2, or 3.

For packets within the same flow (that is, packets with the same combination of input, output, and priority), packet sequence must be maintained through the switch fabric and packet ordering must be guaranteed during packet transmission and reception. This is accomplished by offsetting packet transmission on each port (within the group) by four clock cycles (32 ns). For example, when ports 0, 1, 2, and 3 are grouped, a packet is transmitted on port 1 32 ns after a packet is transmitted on port 0.

3.11.2 Bitmap Mapping with Port Paralleling

When four ports are grouped for port paralleling, the ingress packet destination bitmap requires only one bit to address the entire group of ports (compared to the four bits required to address four single ports). When all 32 ports are grouped for port paralleling, the PowerPRS 64G operates like an 8 × 8 port device and only requires an 8-bit destination bitmap. A shorter bitmap header increases the total payload bandwidth available on each port. The required destination bitmap length for ports grouped for port paralleling is as follows:

- 32 bits for up to two groups
- 24 bits for three to five groups
- 16 bits for six or seven groups
- 8 bits for eight groups

Any port numbered as a multiple of four (that is, 0, 4, 8, 12,16, 20, 24, or 28) can be configured for four-way port paralleling. The lowest-numbered ports in each of the fixed groups of four are used to identify the group. For example, when ports 0, 1, 2, and 3 are grouped, the group is identified as port 0.

Table 3-17 shows how 32 PowerPRS 64G ports can be configured into eight groups (four ports each) for port paralleling. Note that each group of ports is always referenced by the lowest-numbered port in the group. In the table, the output queues associated with each group are shown in bold, and the physical output queues "consumed" (no longer available) by the port paralleling configuration are shown in parentheses. The mapping between the logical bit in the destination bitmap and the associated physical output queue is also included.

Table 3-17. Port Paralleling Configuration Examples

Logical Bitmap	Physical Output Queue Number According to Number of Groups Configured for Port Paralleling								
	0 Groups [32 Ports]	1 Group [29 Ports]	2 Groups [26 Ports]	3 Groups [23 Ports]	4 Groups [20 Ports]	5 Groups [17 Ports]	6 Groups [14 Ports]	7 Groups [11 Ports]	8 Groups [8 Ports]
0	00	**00** (01, 02, 03)	**00** (01, 02, 03)	**00** (01, 02, 03)	**00** (01, 02, 03)	**00** (01, 02, 03)	**00** (01, 02, 03)	**00** (01, 02, 03)	**00** (01, 02, 03)
1	01	04	**04** (05, 06, 07)	**04** (05, 06, 07)	**04** (05, 06, 07)	**04** (05, 06, 07)	**04** (05, 06, 07)	**04** (05, 06, 07)	**04** (05, 06, 07)
2	02	05	08	**08** (09, 10, 11)	**08** (09, 10, 11)	**08** (09, 10, 11)	**08** (09, 10, 11)	**08** (09, 10, 11)	**08** (09, 10, 11)
3	03	06	09	12	**12** (13, 14, 15)	**12** (13, 14, 15)	**12** (13, 14, 15)	**12** (13, 14, 15)	**12** (13, 14, 15)
4	04	07	10	13	16	**16** (17, 18, 19)	**16** (17, 18, 19)	**16** (17, 18, 19)	**16** (17, 18, 19)
5	05	08	11	14	17	20	**20** (21, 22, 23)	**20** (21, 22, 23)	**20** (21, 22, 23)
6	06	09	12	15	18	21	24	**24** (25, 26, 27)	**24** (25, 26, 27)
7	07	10	13	16	19	22	25	28	**28** (29, 30, 31)
8	08	11	14	17	20	23	26	29	
9	09	12	15	18	21	24	27	30	
10	10	13	16	19	22	25	28	31	
11	11	14	17	20	23	26	29		
12	12	15	18	21	24	27	30		
13	13	16	19	22	25	28			
14	14	17	20	23	26	29			
15	15	18	21	24	27	30			
16	16	19	22	25	28				
17	17	20	23	26	29				
18	18	21	24	27	30				
19	19	22	25	28	31				

Note: Although these examples use the lowest possible numbered ports to configure the groups for port paralleling, this is *not* required. The column headings show the number of four-port groups and the total number of ports (in brackets). The physical output queues associated with each group are shown in bold, and the physical output queues "consumed" (no longer available) by the port paralleling configuration are shown in parentheses.

Table 3-17. Port Paralleling Configuration Examples

Logical Bitmap	Physical Output Queue Number According to Number of Groups Configured for Port Paralleling								
	0 Groups [32 Ports]	1 Group [29 Ports]	2 Groups [26 Ports]	3 Groups [23 Ports]	4 Groups [20 Ports]	5 Groups [17 Ports]	6 Groups [14 Ports]	7 Groups [11 Ports]	8 Groups [8 Ports]
20	20	23	26	29					
21	21	24	27	30					
22	22	25	28	31					
23	23	26	29						
24	24	27	30						
25	25	28	31						
26	26	29							
27	27	30							
28	28	31							
29	29								
30	30								
31	31								

Note: Although these examples use the lowest possible numbered ports to configure the groups for port paralleling, this is *not* required. The column headings show the number of four-port groups and the total number of ports (in brackets). The physical output queues associated with each group are shown in bold, and the physical output queues “consumed” (no longer available) by the port paralleling configuration are shown in parentheses.

4. Programming Interface

The serial host interface (SHI) is the programming interface between the local processor and the PowerPRS 64G. It provides access to all PowerPRS 64G internal resources through four signals:

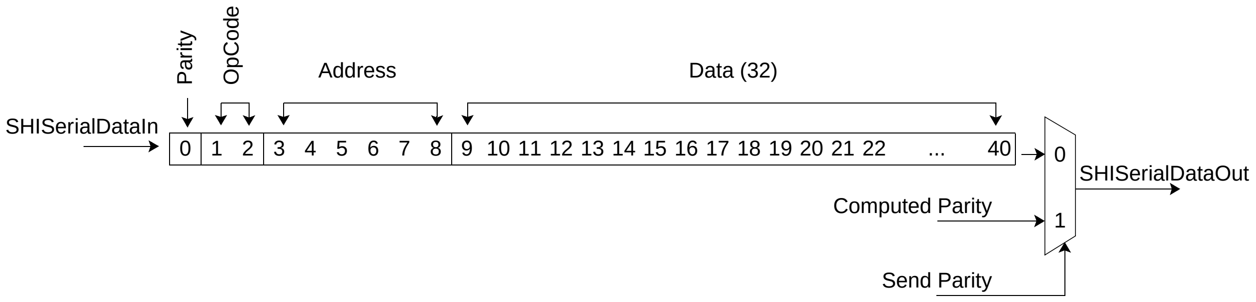
- SHIClockIn
- SHISelectIn#
- SHISerialDataIn
- SHISerialDataOut

The SHI and the SHI internal logic are synchronized to the SHI clock (see *Section 7.2.2 SHI Signals* on page 120). The SHI clock operates at a lower frequency than the system clock.

4.1 SHI Instruction Register

An instruction scanned into the SHI is decoded into four parts:

- Data field (32 bits)
- Address field (6 bits)
- OpCode field (2 bits)
- Parity bit



Bit(s)	Field Name	Description
0	Parity	If the parity is correct, executes the instruction. If the parity is incorrect, inhibits the instruction.
1:2	OpCode	Specifies the SHI command to be executed. See *Table 4-1* for descriptions of these commands.
3:8	Address	Specifies the internal register to be read or written.
9:40	Data (32)	Contains the value to be written or the value that has been returned.

Table 4-1. SHI OpCode Commands

OpCode	Operation	Type	Description
00	No-Op	33-bit	Required after a read status or read register command. Clears the data out of the *SHI Instruction Register.*
01	Read Status	3-bit	Loads the content of the *Status Register* (page 59) into the data field in the *SHI Instruction Register* and simultaneously clears the *Status Register.* This command requires the no-op command (OpCode '00') to send the *Status Register* content and parity over the SHISerialDataOut signal and clear the data field in the *SHI Instruction Register.*
10	Write Register	41-bit	Writes the value of the data field in the *SHI Instruction Register* into the register specified by the address field in the *SHI Instruction Register.*
11	Read Register	9-bit	Loads the content of the register specified by the address field in the *SHI Instruction Register* into the data field in the *SHI Instruction Register.* This command requires the no-op command (OpCode '00') to send the read result and parity over the SHISerialDataOut signal and clear the data field in the *SHI Instruction Register.*
Note: Each read status and read register command must be followed by the no-op command.			

4.2 SHI Instruction Execution

An SHI instruction is invoked when the SHISelectIn# input is set to '0'. During execution, data transmitted over SHISerialDataIn is shifted into the *SHI Instruction Register.* Shifted serial data must begin with the least significant bit and end with the most significant bit of the instruction to be executed. This scan operation is synchronized with SHIClockIn. Instructions always execute one SHI clock cycle after the SHISelectIn# signal changes from an active to an inactive state.

4.3 SHI Parity Checking

Each instruction scanned into the *SHI Instruction Register* has one bit of parity protection. The parity bit is the most significant bit (bit 0) of the *SHI Instruction Register,* and is the last bit scanned.

The *SHI Instruction Register* checks whether an incoming instruction has the required odd parity. If a parity error is detected on a received instruction, the execution of that instruction is inhibited, and the SHI parity error bit is set in the *Status Register* (page 59). All SHI command bits are protected by the parity bit (that is, if SHISelectIn# is active during *n* SHI clock cycles, the parity is checked on *n* bits).

4.4 SHI Parity Generation

Both incoming and outgoing data carry odd parity. This parity is computed for each SHI clock cycle when the SHISelectIn# signal is active. The computed parity is sent on SHISerialDataOut when the SHISelectIn# signal is deactivated.

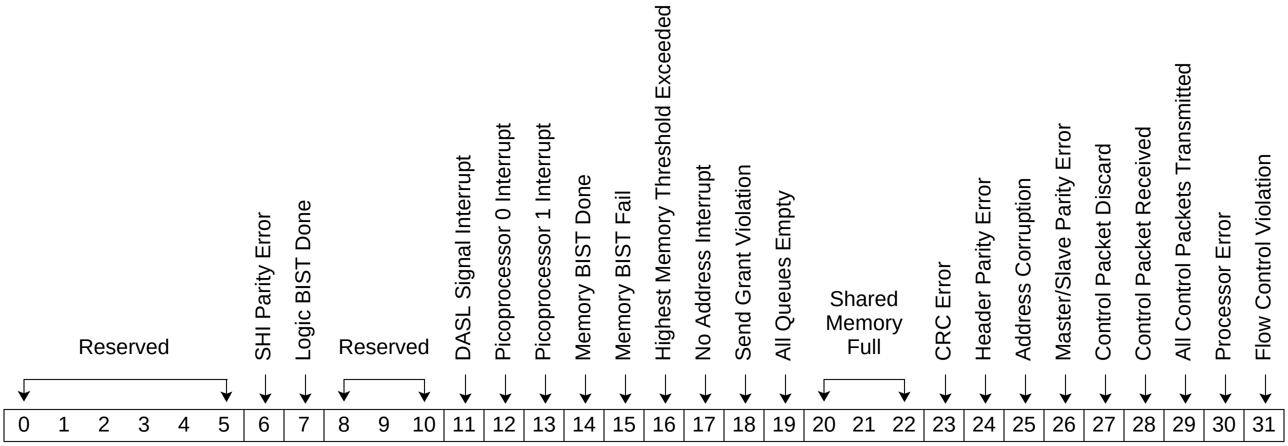
4.5 Status Register

The PowerPRS 64G *Status Register* reports self-test status and error-related events. The SHI “read status” command (see *Section 4.1 SHI Instruction Register* on page 57) can access the contents of the *Status Register.*

Status Register bits generate an interrupt to the local processor unless otherwise noted in the descriptions below. However, an interrupt can be masked by setting the corresponding bit in the *Interrupt Mask Register* (page 65). The occurrence of an event for which the interrupt mask bit is set to '1' sets the corresponding bit in the *Status Register* but does not activate the InterruptOut# signal. The InterruptOut# signal, which interrupts the local processor, is activated only when the global interrupt mask bit *is not* set and the output driver enable bit *is* set in the *Reset Register* (page 64).

Reset Value ‘0000 0000 0000 0000 0000 0000 0000 0000’

Access Type Read/Clear (except as indicated below)



Bit(s)	Field Name	Description
0:5	Reserved	Reserved.
6	SHI Parity Error	Set to ‘1’ when the serial host interface (SHI) detects a parity error in the SHI instruction.
7	Logic BIST Done (Read Only)	Set to ‘1’ when the built-in self-test (BIST) controller completes internal processing after a logic BIST request command. This command is issued by setting the logic BIST requested bit in the *Reset Register* (page 64). *Status Register* bits 8:31 are forced to ‘0’ while the BIST is running. This bit does not generate an interrupt.
8:10	Reserved	Reserved.
11	DASL Signal Interrupt	Set to ‘1’ when an interrupt is generated because a bit in the *DASL Signal Lost Register* (page 74) has changed.
12	Picoprocessor 0 Interrupt	Set to ‘1’ when the internal processor for ports 0 to 15 generates an interrupt.
13	Picoprocessor 1 Interrupt	Set to ‘1’ when the internal processor for ports 16 to 31 generates an interrupt.
14	Memory BIST Done (Read Only)	Set to ‘1’ when the BIST controller completes internal processing after a memory BIST request command. This command is issued by setting the memory BIST requested bit in the *Reset Register.* This bit does not generate an interrupt.

Bit(s)	Field Name	Description
15	Memory BIST Fail (Read Only)	Set to ‘1’ when, after completion of the memory BIST process, at least one memory BIST check failed on at least one RAM. This bit is valid only when the memory BIST done bit is asserted. This bit does not generate an interrupt.
16	Highest Memory Threshold Exceeded	Set to ‘1’ when shared memory occupancy is equal to or greater than the priority 0 threshold. This is an event, not a status; therefore, it occurs when shared memory occupancy exceeds the threshold and does not change when occupancy falls below the threshold.
17	No Address Interrupt	Set to ‘1’ when an interrupt is generated because a packet is received on an input port when no store address is available.
18	Send Grant Violation	Set to ‘1’ when an attached device removes the send grant for all priorities for the number of packet cycles defined in the send grant antistreaming threshold field in the *Configuration 0 Register* (page 85). The port is identified in the *Send Grant Violation Register* (page 97). This function is enabled only when the link is synchronized and the output queue is enabled.
19	All Queues Empty	Set to ‘1’ by the edge detection that occurs as soon as all the output queues are empty.
20:22	Shared Memory Full	Indicates that the total number of packets in the shared memory has exceeded the threshold for that priority: 000 No priority is full 001 Priority 3 is full 010 Priorities 2 and 3 are full 011 Priorities 1, 2, and 3 are full 100 Priorities 0, 1, 2, and 3 are full Others Reserved This is an event, not a status; therefore, it occurs when the number of packets exceeds the threshold and does not change when the number of packets falls below the threshold.
23	CRC Error	Set to ‘1’ when a data trailer CRC error is detected on an input port. The port is identified via the *CRC Error Register* (page 92). The number of CRC errors for all the ports is reported in the trailer CRC error count field in the *Error Counter Register* (page 93).
24	Header Parity Error	Set to ‘1’ when a parity error is detected in an incoming packet header. The port is identified via the *Header Parity Error Register* (page 93). The number of parity errors for all the ports is reported in the header parity error count field in the *Error Counter Register.*
25	Address Corruption	Detected by the address manager. Set to ‘1’ when a counter is initialized to a value when its current value *is not* zero or when an address is freed by an output controller when its counter value *is* zero. When an address is corrupted, one or more addresses are lost from the pool of addresses available to store packets. A reset is required to recover lost addresses.
26	Master/Slave Parity Error	Set to ‘1’ when a parity error is detected on the speed expansion bus that connects the master device to the slave device.
27	Control Packet Discard	Set to ‘1’ when an incoming control packet is discarded because the control packet receive queue is full.
28	Control Packet Received	Set to ‘1’ when a new control packet is received. The number of control packets currently in the control packet receive queue is indicated by the *Control Packet Counter Register* (page 94).
29	All Control Packets Transmitted	Set to ‘1’ when a control packet has been successfully transmitted to all destinations.
30	Processor Error	Set to ‘1’ when an error is generated because the local processor initiated a new command or operation before the PowerPRS 64G internal logic was ready. This interrupt is generated when: • There is a write to the *Shared Memory Pointer Register* (page 99) while a read or write operation is pending. • There is a read from or write to the *Shared Memory Data Register* (page 99) while a read or write operation is pending.
31	Flow Control Violation	Set to ‘1’ when a flow control violation interrupt is generated. For unicast packets, this interrupt is generated when a packet is destined to an output for which neither an output queue grant nor a memory grant has been issued in the past *n* packet cycles. For multicast packets, only memory grant information is used to detect violations. The flow control violation function is enabled, and *n* is set, in the flow control latency field in the *Configuration 0 Register.* The violating ports are identified by the corresponding bits in the *Flow Control Violation Register* (page 94).

5. Register Descriptions

This section describes the registers, including field definitions, that provide the mechanism for PowerPRS 64G configuration specification and status reporting.

Table 5-1 identifies each register and provides the page number where the corresponding description is located. In the register descriptions:

- *Reserved* bits return ‘0’ when read and ignore all write values.
- *Spare* bits can be read or written but do not affect device operation.

Table 5-1. Register Map (Page 1 of 2)

Register Name	Address	Access	Page
SHI Internal Registers: x‘00’ to x‘07’			
PLL Programming Register	x‘00’	Read/Write	63
Reset Register	x‘01’	Read/Write	64
Interrupt Mask Register	x‘02’	Read/Write	65
BIST Counter Register	x‘03’	Read/Write	66
BIST Data Register	x‘04’	Read/Write	67
BIST Select Register	x‘05’	Read/Write	68
Debug Bus Select Register	x‘06’	Read/Write	69
Reserved	x‘07’		
DASL Programming Registers: x‘08’ to x‘1C’			
DASL Output Driver Enable Register	x‘08’	Read/Write	72
Output Port Enable Register	x‘09’	Read/Write	72
Synchronization Packet Transmit Register	x‘0A’	Read/Write	73
Input Port Enable Register	x‘0B’	Read/Write	73
DASL Signal Lost Register	x‘0C’	Read Only	74
SDC RLOS Enable Register	x‘0D’	Read/Write	74
DASL Synchronization Hunt Register	x‘0E’	Read/Write	75
DASL Synchronization Status Register	x‘0F’	Read Only	75
Picoprocessor Instruction Memory Access Register	x‘10’	Read/Write	76
DASL Configuration Register	x‘11’	Read/Write	77
DASL Port Error Register	x‘12’	Read Only	78
DASL Port Quality Mask Register	x‘13’	Read/Write	78
DASL Port Quality Register	x‘14’	Read Only	78
SDC Resource Address Registers	x‘15’ and x‘19’	Read/Write	79
SDC Resource Control Registers	x‘16’ and x‘1A’	Read/Write	80
SDC Resource Data Registers	x‘17’ and x‘1B’	Read/Write	81
SDC Status Registers	x‘18’ and x‘1C’	Read Only	82

Table 5-1. Register Map (Page 2 of 2)

Register Name	Address	Access	Page
Flow Control Pin Status and Setting Registers: x‘1D’ to x‘1F’			
Send Grant per Priority Register	x‘1D’	Read/Write	83
Send Grant Status Register	x‘1E’	Read Only	83
Receive Grant Status Register	x‘1F’	Read Only	84
Functional Registers: x‘20’ to x‘3F’			
Configuration 0 Register	x‘20’	Read/Write	85
Configuration 1 Register	x‘21’	Read/Write	87
Output Queue Enable Register	x‘22’	Read/Write	89
Input Controller Enable Register	x‘23’	Read/Write	90
Color Detection Disable Register	x‘24’	Read/Write	90
Send Grant Enable Register	x‘25’	Read/Write	91
Force Send Grant Register	x‘26’	Read/Write	91
Expected Color Received Register	x‘27’	Read Only	92
CRC Error Register	x‘28’	Read/Clear	92
Header Parity Error Register	x‘29’	Read/Clear	93
Error Counter Register	x‘2A’	Read/Clear	93
Flow Control Violation Register	x‘2B’	Read/Clear	94
Control Packet Counter Register	x‘2C’	Read Only	94
Output Queue Status Registers	x‘2D’ to x‘30’	Read/Clear	95
Color Packet Received Register	x‘31’	Read/Clear	96
Send Grant Violation Register	x‘32’	Read/Clear	97
Occupancy Counter Register	x‘33’	Read Only	97
Shared Memory Pointer Register	x‘34’	Read/Write	99
Shared Memory Data Register	x‘35’	Read/Write	99
Command Register	x‘36’	Read/Write	100
Control Packet Destination Register	x‘37’	Read/Write	101
Bitmap Filter Register	x‘38’	Read/Write	101
Threshold Access Register	x‘39’	Read/Write	102
Credit Table Access Register	x‘3A’	Read/Write	104
Look-Up Table Access Register	x‘3B’	Read/Write	106
Bitmap Mapping Register	x‘3C’	Read/Write	107
Best-Effort Resources Access Register	x‘3D’	Read/Write	108
Best-Effort Discard Alarm Register	x‘3E’	Read Only	109
Side Communication Channel Input Reporting Register	x‘3F’	Read Only	109

Note: Registers x‘00’ to x‘06’ are implemented in the serial host interface (SHI) logic and are reset by activating the PowerOnResetIn# input signal. These registers are accessible before the phase-locked loop (PLL) is started or the flush is complete. All the bits in the remaining registers are set to ‘0’ during a flush, unless otherwise specified.

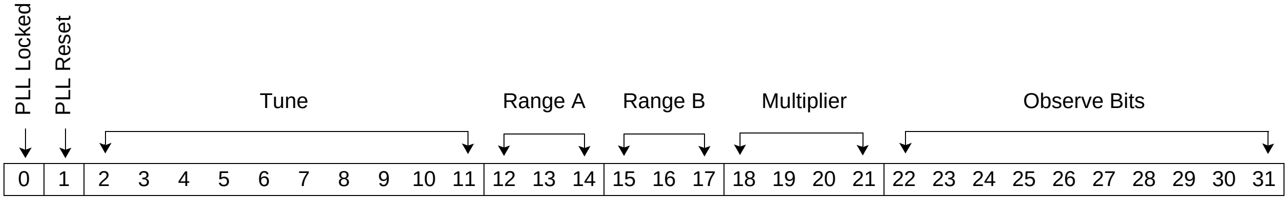
5.1 SHI Internal Registers

5.1.1 PLL Programming Register

Address x‘00’

Access Type Read/Write

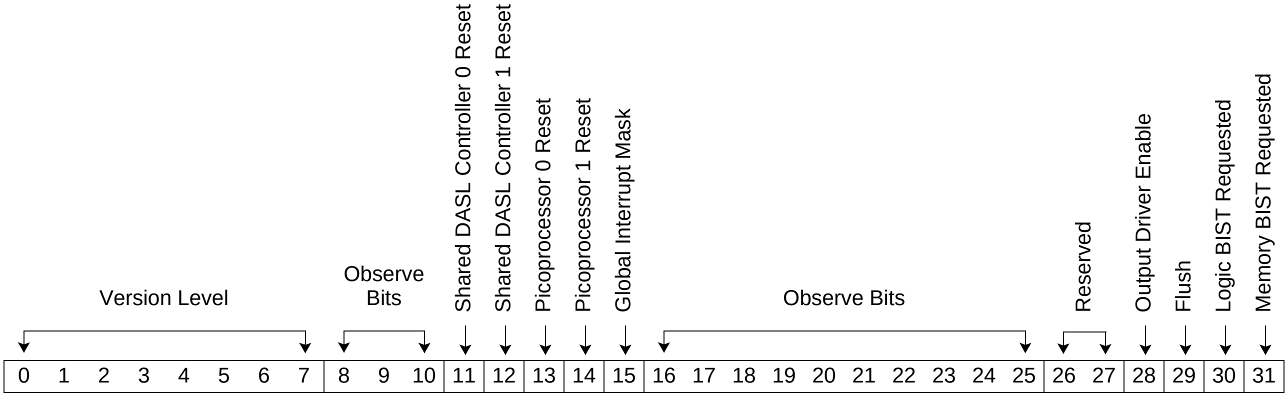
Reset Value ‘u100 0000 0000 0000 0000 00uu uuuu uuuu’, where ‘u’ = undefined



Bit(s)	Field Name	Description
0	PLL Locked (read only)	When set to ‘1’, the feedback clock is in phase with the reference clock.
1	PLL Reset	When set to ‘1’, holds the PLL in a reset state. Cannot be released until the reference clock is stable and the PLL is correctly programmed.
2:11	Tune	Used to optimize PLL stability and jitter. Must be set to ‘10 1011 1110’.
12:14	Range A	Used to select the PLL output frequency. Must be set to ‘110’.
15:17	Range B	Not used. Must be set to ‘110’.
18:21	Multiplier	Defines the PLL feedback divider. Must be set to ‘0010’.
22:31	Observe Bits (read only)	Used for testing (ten bits [20:11]).

5.1.2 Reset Register

Address	x‘01’
Access Type	Read/Write
Reset Value	‘0000 0011 uuu1 1111 uuuu uuuu uu00 0100’, where ‘u’ = undefined



Bit(s)	Field Name	Description
0:7	Version Level (read only)	Indicates the current version of the device.
8:10	Observe Bits (read only)	Used for testing (three bits [0, 21, 22]).
11	Shared DASL Controller 0 Reset	When set to ‘1’, forces the DASL logic attached to ports 0 to 15 into a reset state. This bit must remain asserted until the *DASL Configuration Register* (page 77) is fully programmed.
12	Shared DASL Controller 1 Reset	When set to ‘1’, forces the DASL logic attached to ports 16 to 31 into a reset state. This bit must remain asserted until the *DASL Configuration Register* is fully programmed.
13	Picoprocessor 0 Reset	When set to ‘1’, forces the internal processor attached to ports 0 to 15 into a reset state. This bit must remain asserted until the corresponding instruction memory is fully programmed.
14	Picoprocessor 1 Reset	When set to ‘1’, forces the internal processor attached to ports 16 to 31 into a reset state. This bit must remain asserted until the corresponding instruction memory is fully programmed.
15	Global Interrupt Mask	1 Disables event- and error-generated interrupts to the local processor. The device interrupt signal (active low) is tristated and pulled up with an external resistor. The *Status Register* (page 59) bits are asserted when the corresponding events or errors occur. 0 Enables event- and error-generated interrupts to the local processor.
16:25	Observe Bits (read only)	Used for testing (ten bits [3, 2, 1, 6, 5, 4, 10, 9, 8, 7]).
26:27	Reserved	Reserved.
28	Output Driver Enable	1 Enables all device drivers until another configuration disables them. 0 Disables (tristates) all drivers except for the SHISerialDataOut driver.
29	Flush	When set to ‘1’, keeps all the device logic except the SHI internal logic in the reset state.
30	Logic BIST Requested	When set to ‘1’, enables the built-in self-test (BIST) controller to start executing the internal logic BIST as soon as the flush bit is deactivated. This bit can be asserted only while the flush bit is active. BIST completion is reported in the *Status Register.* See *Section 6.4 Logic BIST Execution Sequence* on page 113 for more information.

Bit(s)	Field Name	Description
31	Memory BIST Requested	When set to ‘1’, enables the BIST controller to start executing the memory BIST as soon as the flush bit is deactivated. This bit can be asserted only while the flush bit is active. Memory BIST completion and results are reported in the *Status Register* (page 59). See *Section 6.5 Memory BIST Execution Sequence* on page 113 for more information.

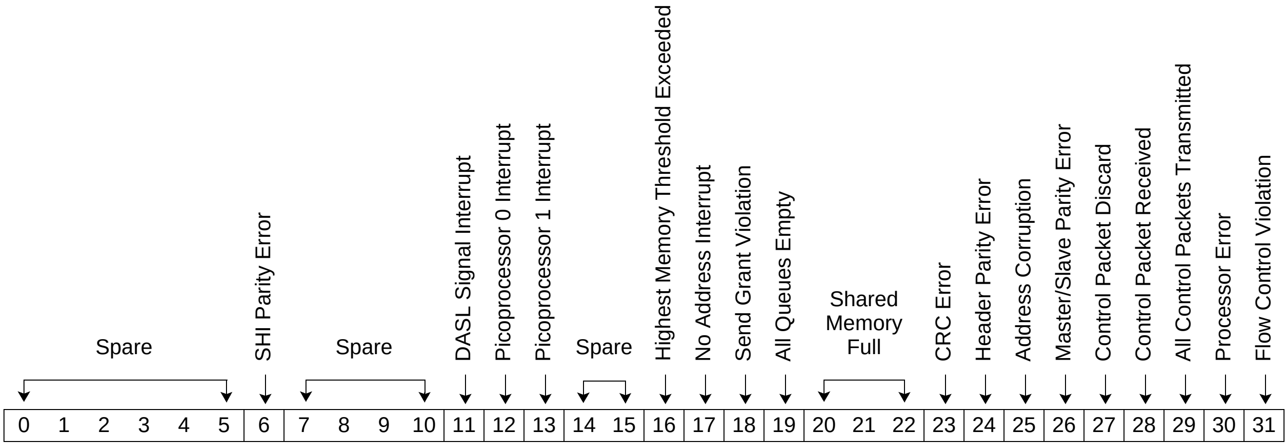
5.1.3 Interrupt Mask Register

This register sets masks for the *Status Register* (page 59) application bits. Note that the occurrence of an event for which the mask bit is set to ‘1’ sets the corresponding bit in the *Status Register* but does not generate an interrupt. For information about an event or error masked here, see the *Status Register* bit descriptions.

Address x‘02’

Access Type Read/Write

Reset Value ‘0000 0000 0000 0000 0000 0000 0000 0000’



Bit(s)	Field Name	Description
0:5	Spare	Spare.
6	SHI Parity Error	When set to ‘1’, masks this interrupt.
7:10	Spare	Spare.
11	DASL Signal Interrupt	When set to ‘1’, masks this interrupt.
12	Picoprocessor 0 Interrupt	When set to ‘1’, masks this interrupt.
13	Picoprocessor 1 Interrupt	When set to ‘1’, masks this interrupt.
14:15	Spare	Spare.
16	Highest Memory Threshold Exceeded	When set to ‘1’, masks this interrupt.
17	No Address Interrupt	When set to ‘1’, masks this interrupt.
18	Send Grant Violation	When set to ‘1’, masks this interrupt.

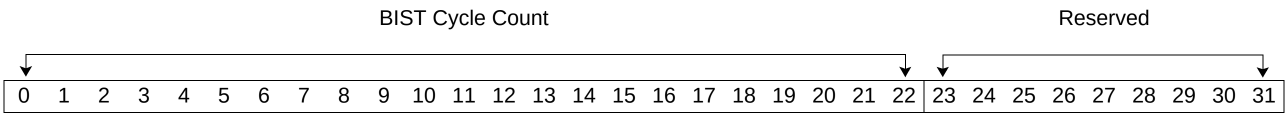
Bit(s)	Field Name	Description
19	All Queues Empty	When set to ‘1’, masks this interrupt.
20:22	Shared Memory Full	When set to ‘1’, masks this interrupt.
23	CRC Error	When set to ‘1’, masks this interrupt.
24	Header Parity Error	When set to ‘1’, masks this interrupt.
25	Address Corruption	When set to ‘1’, masks this interrupt.
26	Master/Slave Parity Error	When set to ‘1’, masks this interrupt.
27	Control Packet Discard	When set to ‘1’, masks this interrupt.
28	Control Packet Received	When set to ‘1’, masks this interrupt.
29	All Control Packets Transmitted	When set to ‘1’, masks this interrupt.
30	Processor Error	When set to ‘1’, masks this interrupt.
31	Flow Control Violation	When set to ‘1’, masks this interrupt.

5.1.4 BIST Counter Register

Address x‘03’

Access Type Read/Write

Reset Value ‘0000 0000 0000 0000 0000 0000 0000 0000’

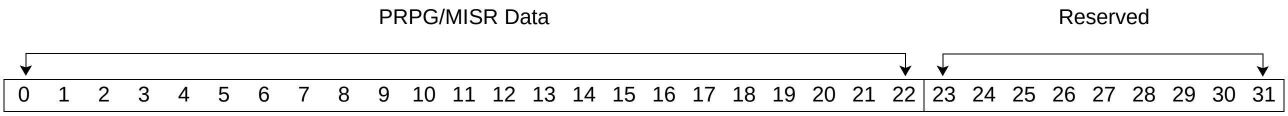


Bit(s)	Field Name	Description
0:22	BIST Cycle Count	Specifies the number of BIST cycles to be performed.
23:31	Reserved	Reserved.

5.1.5 BIST Data Register

This register, along with the *BIST Select Register* (page 68), provides indirect access to the internal pseudo-random pattern generator (PRPG) and multiple-input signature (MISR) registers.

Address	x‘04’
Access Type	Read/Write
Reset Value	Undefined



Bit(s)	Field Name	Description
0:22	PRPG/MISR Data	Contains the data that has been or will be exchanged using the settings provided in the *BIST Select Register.*
23:31	Reserved	Reserved.

5.1.6 BIST Select Register

This register, along with the *BIST Data Register* (page 67), provides indirect access to the internal PRPG and MISR registers.

Write access to an internal PRPG or MISR register requires two SHI commands:

1. Write the BIST register select field in the *BIST Select Register* with the value specifying which internal PRPG or MISR register is to be accessed.
2. Write the *BIST Data Register* with the value desired for the internal PRPG or MISR register specified in step 1. The internal PRPG or MISR register is loaded.

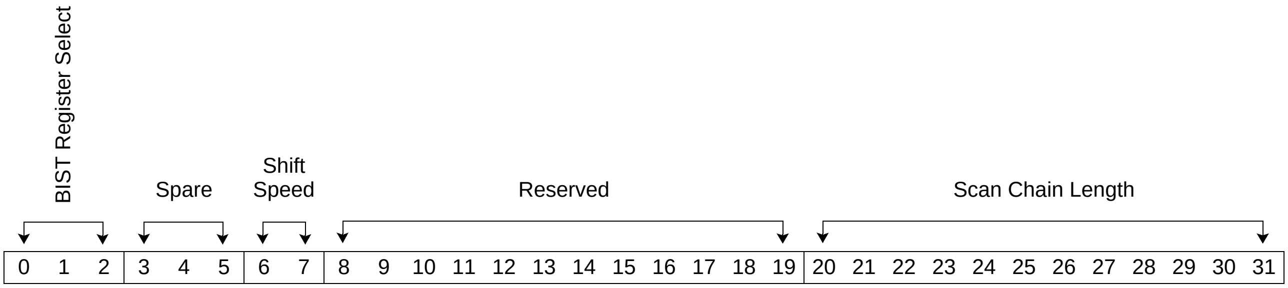
Read access to an internal PRPG or MISR register requires two SHI commands:

1. Write the BIST register select field in the *BIST Select Register* with the value specifying which internal PRPG or MISR register is to be accessed.
2. Read the *BIST Data Register.* The value for the internal PRPG or MISR register specified in step 1 is returned.

Address x‘05’

Access Type Read/Write

Reset Value ‘0000 0000 0000 0000 0000 0000 0000 0000’



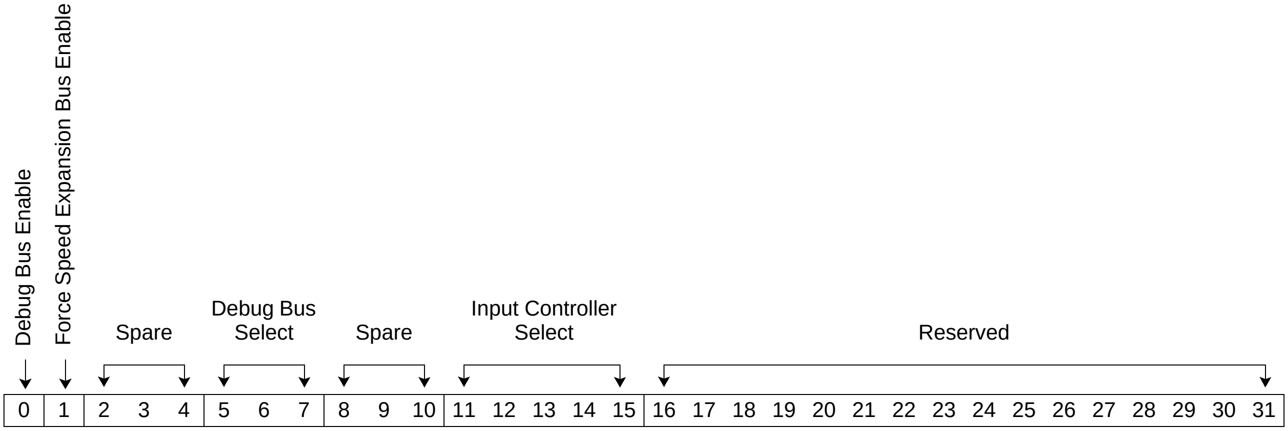
Bit(s)	Field Name	Description
0:2	BIST Register Select	Specifies the BIST register: 000 PRPG0 001 PRPG1 010 PRPG2 011 PRPG3 100 MISR0 101 MISR1 110 MISR2 111 MISR3
3:5	Spare	Spare.
6:7	Shift Speed	Defines the delay between the A and B clock pulses while shifting occurs during the BIST: 00 8 ns 01 16 ns 10 24 ns 11 32 ns
8:19	Reserved	Reserved.
20:31	Scan Chain Length	Specifies the scan chain length.

5.1.7 Debug Bus Select Register

Address x‘06’

Access Type Read/Write

Reset Value ‘0000 0000 0000 0000 0000 0000 0000 0000’



Bit(s)	Field Name	Description
0	Debug Bus Enable	When set to ‘1’, enables the debug bus drivers as well as OSC125MhzOut, OSC250MhzOut, and OSC500MhzOut.
1	Force Speed Expansion Bus Enable	When set to ‘1’, enables the master/slave bus drivers regardless of the device configuration. **Note:** Use this bit with caution to avoid destruction of the device drivers when the device is configured for external speed expansion.
2:4	Spare	Spare.
5:7	Debug Bus Select	Specifies the device element for which the DebugDataOut[0:15] pins provide information: 000 Sequencer information 001 Address manager information 010 Packet routing switch general information 011 Input controller (see bits 11:15) information 100 Picoprocessor 0 instruction address and companion clock information 101 Picoprocessor 1 instruction address and companion clock information 110 Picoprocessor 0 internal information 111 Picoprocessor 1 internal information *Table 5-2* presents the DebugDataOut[0:15] pin information for each of the device elements.
8:10	Spare	Spare.
11:15	Input Controller Select	Specifies the input controller.
16:31	Reserved	Reserved.

Table 5-2. DebugDataOut[0:15] Pin Information by Debug Bus Select Field Value

Debug Bus Select Field Value	DebugDataOut[0:15] Pin(s)	Signal Carried
‘000’	**Sequencer Information**	
	DebugDataOut[0:3]	SEQ_T_TXARAToSlaves
	DebugDataOut[4:7]	SEQ_T_TXASAToSlaves
	DebugDataOut[8]	SEQ_MSSync
	DebugDataOut[9]	SPINP_SeqClk
	DebugDataOut[10]	SEQ_SeqClk
	DebugDataOut[11]	ARG_NotPGSelectToSEQ
	DebugDataOut[12:15]	SEQ_T_OQAStart
‘001’	**Address Manager Information**	
	DebugDataOut[0]	OQA_ASACountVal
	DebugDataOut[1]	ADM_MemFreeAck
	DebugDataOut[2]	ADM_Regenerating
	DebugDataOut[3]	ARG_NotPGSelectToADM
	DebugDataOut[4]	ADM_FreeARACtrlPktAck
	DebugDataOut[5:15]	OQA_ASAToADM
‘010’	**Packet Routing Switch General Information**	
	DebugDataOut[0]	B_CLK_8ns
	DebugDataOut[1]	C_CLK_8ns
	DebugDataOut[2]	CT_RAM_NC_CLK
	DebugDataOut[3]	OCM_B_CLK
	DebugDataOut[4]	DASL_TB1
	DebugDataOut[5]	DASL_TC1
	DebugDataOut[6]	DASL_TC2
	DebugDataOut[7]	MABIST_BCLK
	DebugDataOut[8]	MABIST_CCLK
	DebugDataOut[9]	MABIST_STCLK
	DebugDataOut[10]	MABIST_LBIST
	DebugDataOut[11]	PLL_Lock
	DebugDataOut[12]	PLL_Reset
	DebugDataOut[13]	SynchronousFlush
	DebugDataOut[14]	MabistRequest
	DebugDataOut[15]	BistRequest

Table 5-2. DebugDataOut[0:15] Pin Information by Debug Bus Select Field Value (Continued)

Debug Bus Select Field Value	DebugDataOut[0:15] Pin(s)	Signal Carried
‘011’ (input controller select field selecting 1 of 32 input controllers)	**Input Controller Information**	
	DebugDataOut[0:4]	ByteCounter
	DebugDataOut[5]	RowCounter
	DebugDataOut[6]	HdrPtyError
	DebugDataOut[7]	CRCError
	DebugDataOut[8]	Receiving
	DebugDataOut[9]	IdleCell
	DebugDataOut[10]	ControlCell
	DebugDataOut[11]	DataCell
	DebugDataOut[12]	ASAValid
	DebugDataOut[13]	NSAValid
	DebugDataOut[14]	AddrInTime
	DebugDataOut[15]	MasterByteCounterVal
‘100’ *or* ‘101’	**Picoprocessor 0 *or* Picoprocessor 1 Instruction Address and Companion Clock**	
	DebugDataOut[0:10]	Picoprocessor address
	DebugDataOut[11:14]	Reserved
	DebugDataOut[15]	Picoprocessor clock
‘110’ *or* ‘111’	**Picoprocessor 0 *or* Picoprocessor 1 Internal Information**	
	DebugDataOut[0:1]	from IDCD_CC unit
	DebugDataOut[2]	MUXR_CNTL
	DebugDataOut[3]	MUXQ_CNTL
	DebugDataOut[4]	MUXA_CNTL
	DebugDataOut[5]	ImmediateDataFromInstruction
	DebugDataOut[6:7]	DataWidth
	DebugDataOut[8:9]	ALU_B_Select
	DebugDataOut[10:11]	ALU_A_Select
	DebugDataOut[12]	AccessPY
	DebugDataOut[13]	PY_AutoIncrement
	DebugDataOut[14]	AccessPX
	DebugDataOut[15]	PX_AutoIncrement

5.2 DASL Programming Registers

These registers provide data-aligned synchronous link (DASL) logic access and control. The DASL logic contains 32 DASL receivers, 32 DASL transmitters, and 2 shared DASL controllers (SDCs). SDC 0 controls ports 0 to 15, and SDC 1 controls ports 16 to 31. Each SDC contains:

- A picoprocessor with instruction memory and local memory
- A set of I/O registers and hardware for DASL receiver control assistance

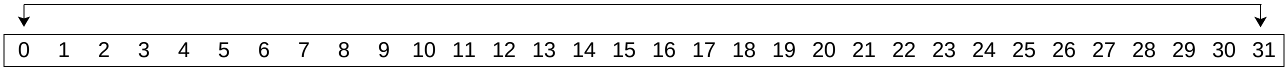
5.2.1 DASL Output Driver Enable Register

Address x‘08’

Access Type Read/Write

Reset Value ‘0000 0000 0000 0000 0000 0000 0000 0000’

DASL Output Driver Enable (for port *n* = bit *n)*



Bit(s)	Field Name	Description
0:31	DASL Output Driver Enable (for port *n* = bit *n)*	1 Enables the DASL output driver for the port, if the output driver enable bit is set in the *Reset Register* (page 64) and the FullyInsertedIn# signal is active (low level). 0 Disables (tristates) the DASL output driver.

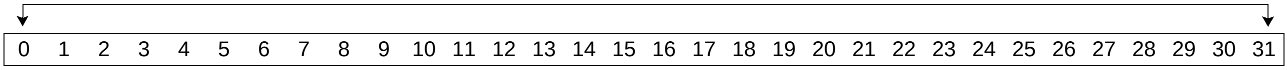
5.2.2 Output Port Enable Register

Address x‘09’

Access Type Read/Write

Reset Value ‘0000 0000 0000 0000 0000 0000 0000 0000’

Output Port Enable (for port *n* = bit *n)*



Bit(s)	Field Name	Description
0:31	Output Port Enable (for port *n* = bit *n)*	1 Enables normal data transmission on the output port. 0 Forces the data sent by the port to ‘0’.

5.2.3 Synchronization Packet Transmit Register

Address	x‘0A’
Access Type	Read/Write
Reset Value	‘0000 0000 0000 0000 0000 0000 0000 0000’

Synchronization Packet Transmit (for port *n* = bit *n)*

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

Bit(s)	Field Name	Description	
0:31	Synchronization Packet Transmit (for port *n* = bit *n)*	1	Enables the output port to transmit synchronization packets. While a port is transmitting synchronization packets, the data packets destined to that port are discarded at the rate they would have been transmitted. When a port is not transmitting synchronization packets, it transmits other packet types normally.
		0	Disables the output port from transmitting synchronization packets.

5.2.4 Input Port Enable Register

Address	x‘0B’
Access Type	Read/Write
Reset Value	‘0000 0000 0000 0000 0000 0000 0000 0000’

Input Port Enable (for port *n* = bit *n)*

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

Bit(s)	Field Name	Description	
0:31	Input Port Enable (for port *n* = bit *n)*	1	Enables the input port. The picoprocessor is allowed to align and synchronize the data stream.
		0	Disables the input port. The picoprocessor is not allowed to align and synchronize the data stream. Packets cannot be received.

5.2.5 DASL Signal Lost Register

Address x‘0C’

Access Type Read Only

Reset Value Undefined

DASL Signal Lost (for port *n* = bit *n*)

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

Bit(s)	Field Name	Description
0:31	DASL Signal Lost (for port *n* = bit *n)*	1 Does not detect a signal on at least one DASL to the corresponding input port. 0 Detects a signal on each link to the corresponding input port. If not masked, a change in any of these bits generates a DASL signal interrupt (see the *Status Register* [page 59]).

5.2.6 SDC RLOS Enable Register

Address x‘0D’

Access Type Read/Write

Reset Value ‘0000 0000 0000 0000 0000 0000 0000 0000’

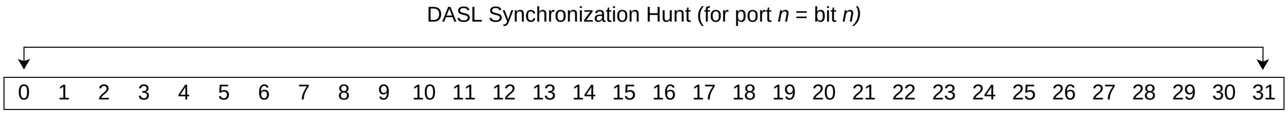
SDC RLOS Enable (for port *n* = bit *n)*

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

Bit(s)	Field Name	Description
0:31	SDC RLOS Enable (for port *n* = bit *n)*	1 Enables the SDC to react to a receiver loss-of-signal (RLOS) assertion on the port receiver by resetting the port phase-alignment state machine. In this case, the port receivers must be resynchronized. 0 Disables the SDC from reacting to an RLOS condition on the port receiver. The *DASL Signal Lost Register* (page 74) always reports an RLOS condition.

5.2.7 DASL Synchronization Hunt Register

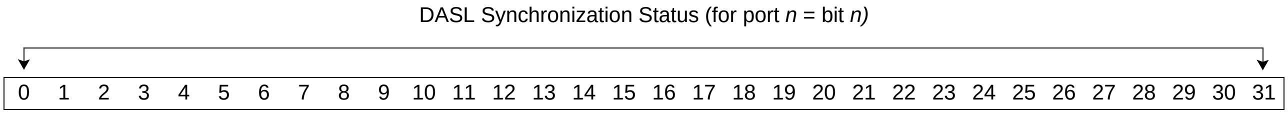
Address	x‘0E’
Access Type	Read/Write
Reset Value	‘0000 0000 0000 0000 0000 0000 0000 0000’



Bit(s)	Field Name	Description
0:31	DASL Synchronization Hunt (for port *n* = bit *n)*	When a bit in this register is set to ‘1’, enables the picoprocessor to start the data stream synchronization sequence (which includes bit phase alignment and packet delineation) for the corresponding input port. The picoprocessor sets the corresponding bit in the *DASL Synchronization Status Register* at the beginning of this process; when the process is complete, the picoprocessor clears the bit. When two PowerPRS 64Gs are configured for external speed expansion, the local processor must set the *DASL Synchronization Hunt Register* and check the *DASL Synchronization Status Register* on the two devices separately. Similarly, the local processor must direct both the master device and the slave device to transmit synchronization packets according to the *Synchronization Packet Transmit Register* (page 73) setting.

5.2.8 DASL Synchronization Status Register

Address	x‘0F’
Access Type	Read Only
Reset Value	‘0000 0000 0000 0000 0000 0000 0000 0000’



Bit(s)	Field Name	Description
0:31	DASL Synchronization Status (for port *n* = bit *n)*	This register indicates the status of the link synchronization sequence (which includes bit phase alignment and packet delineation) initiated by a synchronization hunt command issued via the *DASL Synchronization Hunt Register:* 1 The synchronization sequence is complete, and normal packet reception is possible on the port. 0 The synchronization sequence is incomplete or the port is disabled. When two PowerPRS 64Gs are configured for external speed expansion, the local processor must set the *DASL Synchronization Hunt Register* and check the *DASL Synchronization Status Register* on the two devices separately. Similarly, the local processor must direct both the master device and the slave device to transmit synchronization packets according to the *Synchronization Packet Transmit Register* (page 73) setting.

5.2.9 Picoprocessor Instruction Memory Access Register

This register provides access to the picoprocessor 0 (ports 0 to 15) and picoprocessor 1 (ports 16 to 31) instruction memory.

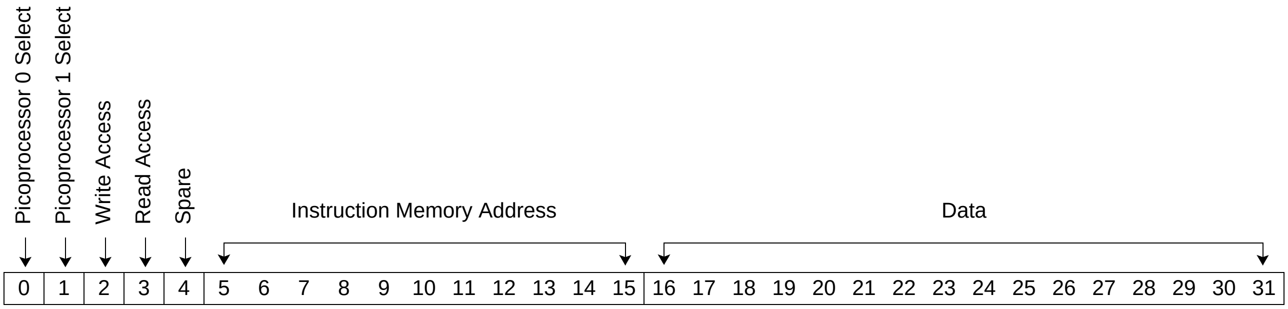
Write access to the *Picoprocessor Instruction Memory Access Register* requires one SHI command:

1. Write the register with either the picoprocessor 0 select bit or the picoprocessor 1 select bit (not both) set to ‘1’, the write bit set to ‘1’, the read bit set to ‘0’, and the instruction memory address field specifying the instruction memory address.

Read access to the *Picoprocessor Instruction Memory Access Register* requires two SHI commands:

1. Write the register with either the picoprocessor 0 select bit or the picoprocessor 1 select bit (not both) set to ‘1’, the write bit set to ‘0’, the read bit set to ‘1’, and the instruction memory address field specifying the instruction memory address to be read.
2. Read the register to return the corresponding data field value.

Address	x‘10’
Access Type	Read/Write
Reset Value	‘0000 0000 0000 0000 0000 0000 0000 0000’



Bit(s)	Field Name	Description
0	Picoprocessor 0 Select	Set to ‘1’ for access requests to and from picoprocessor 0 instruction memory.
1	Picoprocessor 1 Select	Set to ‘1’ for access requests to and from picoprocessor 1 instruction memory.
2	Write Access	Set to ‘1’ to write to the picoprocessor instruction memory.
3	Read Access	Set to ‘1’ to read from the picoprocessor instruction memory.
4	Spare	Spare.
5:15	Instruction Memory Address	Specifies the instruction memory address (between 0 and 2000) to be accessed.
16:31	Data	Specifies the data written to, or the value read from, the specified picoprocessor instruction memory address.

5.2.10 DASL Configuration Register

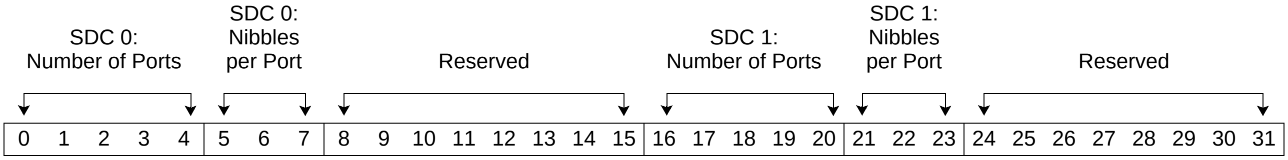
This register must be loaded before the shared DASL controller 0 reset and shared DASL controller 1 reset bits are released in the *Reset Register* (page 64). In addition, this register must be loaded before the instruction memory is loaded. The required loading process is as follows:

1. In the *Reset Register,* set the shared DASL controller 0 reset, shared DASL controller 1 reset, picoprocessor 0 reset, and picoprocessor 1 reset bits.
2. Load the *DASL Configuration Register.*
3. Release the shared DASL controller 0 reset and shared DASL controller 1 reset bits.
4. Load the instruction memory.
5. Release the picoprocessor 0 reset and picoprocessor 1 reset bits.

Address x‘11’

Access Type Read/Write

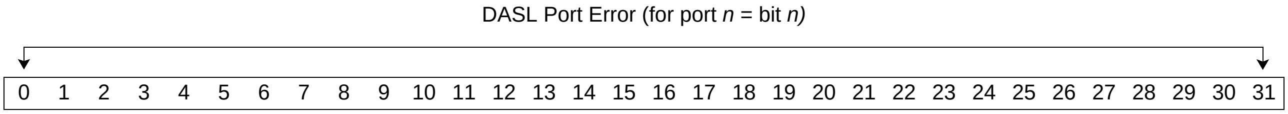
Reset Value ‘0000 0000 0000 0000 0000 0000 0000 0000’



Bit(s)	Field Name	Description
0:4	SDC 0: Number of Ports	The number of ports handled by picoprocessor 0. Must be set to x‘0F’.
5:7	SDC 0: Nibbles per Port	The number of DASLs per port. Must be set to 3.
8:15	Reserved	Reserved. Must be set to 1.
16:20	SDC 1: Number of Ports	The number of ports handled by picoprocessor 1. Must be set to x‘0F’.
21:23	SDC 1: Nibbles per Port	The number of DASLs per port. Must be set to 3.
24:31	Reserved	Reserved. Must be set to 1.

5.2.11 DASL Port Error Register

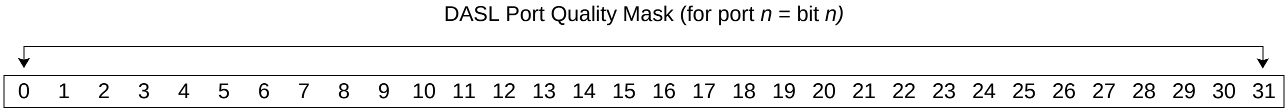
Address	x‘12’
Access Type	Read Only
Reset Value	‘0000 0000 0000 0000 0000 0000 0000 0000’



Bit(s)	Field Name	Description
0:31	DASL Port Error (for port *n* = bit *n)*	When set to ‘1’, the DASL-receiver state machine has detected a port error.

5.2.12 DASL Port Quality Mask Register

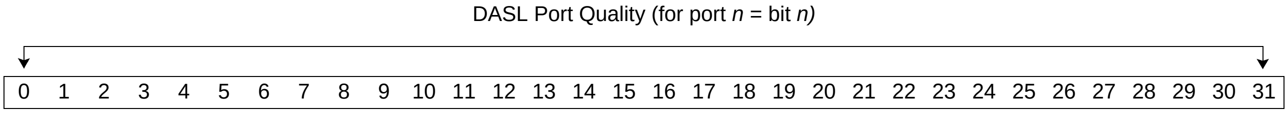
Address	x‘13’
Access Type	Read/Write
Reset Value	‘0000 0000 0000 0000 0000 0000 0000 0000’



Bit(s)	Field Name	Description
0:31	DASL Port Quality Mask (for port *n* = bit *n)*	When set to ‘1’, resets the corresponding bit in the *DASL Port Quality Register.*

5.2.13 DASL Port Quality Register

Address	x‘14’
Access Type	Read Only
Reset Value	‘0000 0000 0000 0000 0000 0000 0000 0000’



Bit(s)	Field Name	Description
0:31	DASL Port Quality (for port *n* = bit *n)*	When set to ‘1’, the picoprocessor has detected a minimum-eye violation on the port. A minimum-eye violation indicates that one or more of the port’s DASLs are operating outside of device specifications (see the *Glossary* on page 151).

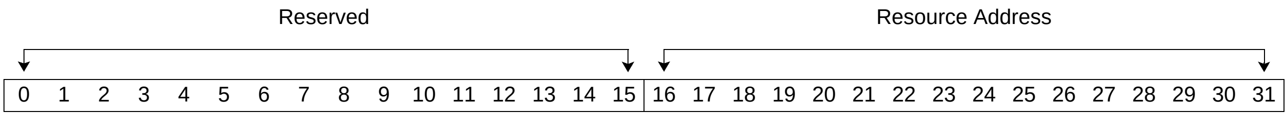
5.2.14 SDC Resource Address Registers

These registers are not used in normal operating mode.

Address x‘15’ SDC 0 (ports 0 to 15)
x‘19’ SDC 1 (ports 16 to 31)

Access Type Read/Write

Reset Value ‘0000 0000 0000 0000 0000 0000 0000 0000’

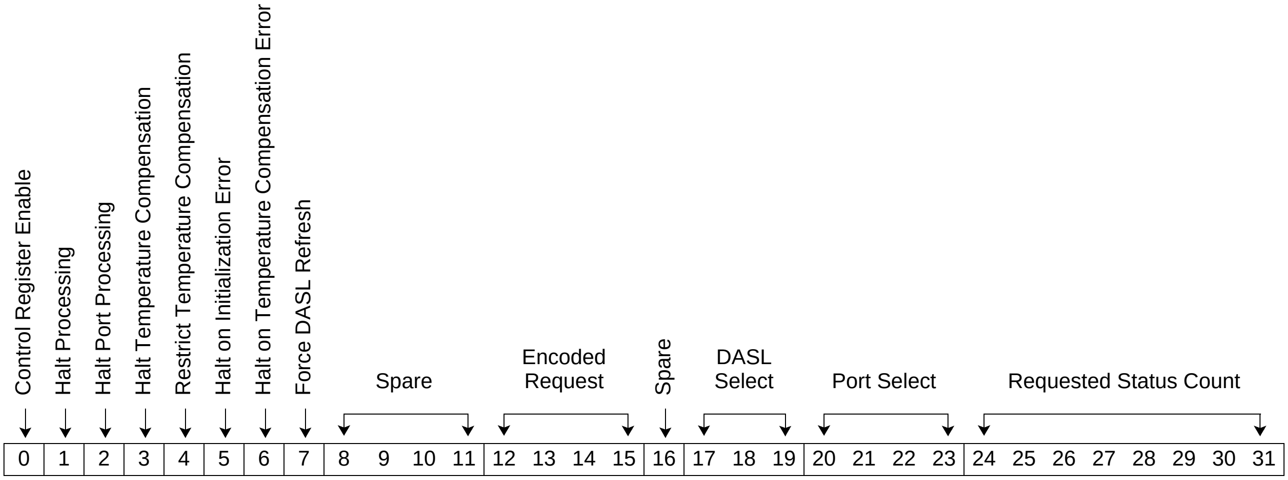


Bit(s)	Field Name	Description
0:15	Reserved	Reserved.
16:31	Resource Address	Specifies the resource address for a read or write operation, allowing indirect access to the local store and picoprocessor I/O registers.

5.2.15 SDC Resource Control Registers

These registers are not used in normal operating mode.

Address	x‘16’	SDC 0 (ports 0 to 15)
	x‘1A’	SDC 1 (ports 16 to 31)
Access Type	Read/Write	
Reset Value	‘0000 0000 0000 0000 0000 0000 0000 0000’	



Bit(s)	Field Name	Description
0	Control Register Enable	1 Enables the register content. 0 Ignores all the other bits of this register except the requested status count bit.
1	Halt Processing	When set to ‘1’, the picoprocessor stops all processing.
2	Halt Port Processing	When set to ‘1’, the picoprocessor stops all processing but continues to update the status and respond to debug requests.
3	Halt Temperature Compensation	When set to ‘1’, the picoprocessor does not perform any port processing functions beyond initialization.
4	Restrict Temperature Compensation	When set to ‘1’, the picoprocessor performs all port processing functions, but the multiplexer setting is not adjusted beyond initialization.
5	Halt on Initialization Error	When set to ‘1’, the picoprocessor stops all processing after the next initialization error.
6	Halt on Temperature Compensation Error	When set to ‘1’, the picoprocessor stops all processing after the next data mode.
7	Force DASL Refresh	When set to ‘1’, the picoprocessor writes the current DASL receiver settings to all ports that have completed initialization but does not perform data mode compensation.
8:11	Spare	Spare.

Bit(s)	Field Name	Description
12:15	Encoded Request	Specifies which services the picocode should perform: 0000 No operation 0001 Requests read access from the address in the corresponding *SDC Resource Address Register* (page 79) 0010 Requests write access from the address in the corresponding *SDC Resource Address Register* 0101 Loads sample memory 1000 Requests a delayed line sample from the DASL specified by the port select and DASL select fields 1001 Updates the DASL data structure with the corresponding *SDC Resource Data Register* (page 81) input for the DASL receiver specified by the port select and DASL select fields Others Reserved
16	Spare	Spare.
17:19	DASL Select	Specifies a specific port DASL for a given action.
20:23	Port Select	Specifies the port for a given action. A port select field request is processed at the next service opportunity for that port. If the designated port value is higher than the number of existing ports, the request is processed for port 0.
24:31	Requested Status Count	Read by the picoprocessor. The value of this field is incremented by one and written to the updated status count field in the corresponding *SDC Status Register* (page 82). This mechanism allows the application to verify that the picoprocessor is operating correctly.

5.2.16 SDC Resource Data Registers

These registers are not used in normal operating mode.

Address	x‘17’	SDC 0 (ports 0 to 15)
	x‘1B’	SDC 1 (ports 16 to 31)
Access Type	Read/Write	
Reset Value	‘0000 0000 0000 0000 0000 0000 0000 0000’	

Data

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

Bit(s)	Field Name	Description
0:31	Data	Contains the SDC resource data for read and write operations.

5.2.17 SDC Status Registers

An *SDC Status Register* is loaded by the functional picocode when the corresponding picoprocessor reset bit is released in the *Reset Register* (page 64).

Address	x‘18’	SDC 0 (ports 0 to 15)
	x‘1C’	SDC 1 (ports 16 to 31)
Access Type	Read Only	
Reset Value	‘0000 0000 0000 0000 0000 0000 0000 0000’	

DASL Control Code Version	DASL Control Code Revision	Reserved	Current Port Identification	Updated Status Count
0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19	20 21 22 23	24 25 26 27 28 29 30 31

Bit(s)	Field Name	Description
0:7	DASL Control Code Version	The DASL control code version that is running.
8:15	DASL Control Code Revision	The DASL control code revision that is running.
16:19	Reserved	Reserved.
20:23	Current Port Identification	The port number that is currently being processed by the SDC.
24:31	Updated Status Count	Written by the picoprocessor. The value of this field is equal to the value of the requested status count field in the corresponding *SDC Resource Control Register* (page 80) plus one. This mechanism allows the application to verify that the picoprocessor is operating correctly.

5.3 Flow Control Pin Status and Setting Registers

5.3.1 Send Grant per Priority Register

Address	x‘1D’
Access Type	Read/Write
Reset Value	Undefined

Send Grant per Priority (for port *n* = bit *n)*

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

Bit(s)	Field Name	Description
0:31	Send Grant per Priority (for port *n* = bit *n)*	1 Enables the send-grant-per-priority mode on the port. When this mode is enabled, the attached device issues send grants based on packet priority. 0 Disables the send-grant-per-priority mode. When this mode is disabled, the attached device issues send grants without packet priority restrictions. See *Section 3.6.1 Send Grants* on page 41 for more information.

5.3.2 Send Grant Status Register

Address	x‘1E’
Access Type	Read Only
Reset Value	Undefined

Send Grant Status (for port *n* = bit *n)*

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

Bit(s)	Field Name	Description
0:31	Send Grant Status (for port *n* = bit *n)*	If the send-grant-per-priority mode is enabled for the port (with the *Send Grant per Priority Register*), a ‘1’ in this bit indicates that the incoming signal contains the send-grant-per-priority framing ‘00001’ and the send grant for at least one priority. If the send-grant-per-priority mode is disabled for the port (see the *Send Grant per Priority Register),* this bit indicates the current level of the incoming send grant input signals from the attached device.

5.3.3 Receive Grant Status Register

Address	x‘1F’
Access Type	Read Only
Reset Value	Undefined

Receive Grant Status (for port *n* = bit *n)*

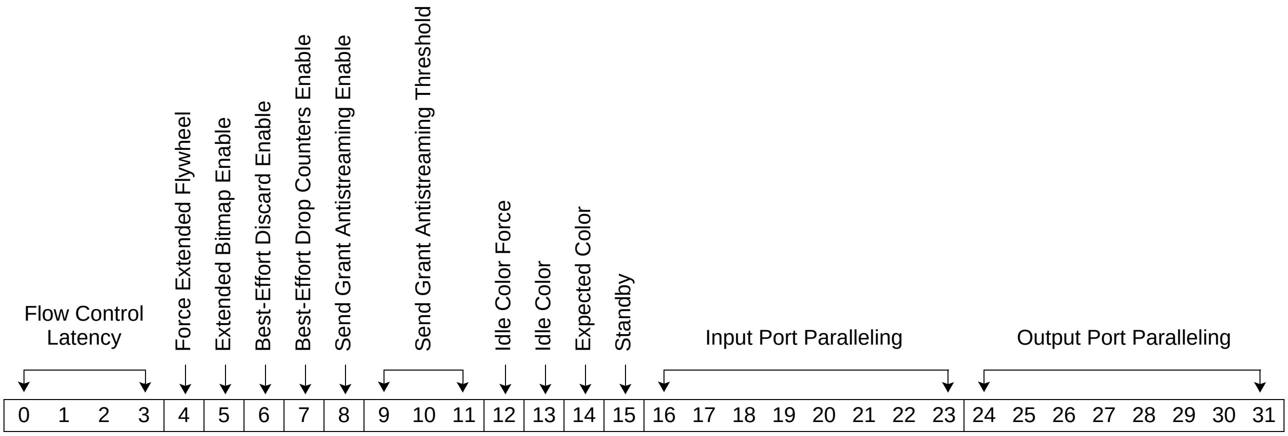
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

Bit(s)	Field Name	Description
0:31	Receive Grant Status (for port *n* = bit *n)*	Indicates the current level of the port’s receive grant input signals.

5.4 Functional Registers

5.4.1 Configuration 0 Register

Address	x‘20’
Access Type	Read/Write
Reset Value	‘0000 0000 0000 0001 0000 0000 0000 0000’



Bit(s)	Field Name	Description
0:3	Flow Control Latency	When enabled, the input controllers check flow control. For a unicast packet, the input controller checks whether an incoming packet is destined to an output for which neither an output queue grant nor a memory grant has been issued in the past *n* packet cycles. For a multicast packet, the input controller checks only the memory grant information. If the packet is destined for an output lacking grants, the packet is discarded. The error is reported via the flow control violation bit in the *Status Register* (page 59) and, unless masked in the *Interrupt Mask Register* (page 65)*,* the error generates a flow control violation interrupt. The violating ports are identified by the corresponding bits in the *Flow Control Violation Register* (page 94). 0000 Function is disabled 0001 Function is enabled with *n* = 8 0010 Function is enabled with *n* = 10 0011 Function is enabled with *n* = 12 0100 Function is enabled with *n* = 14 0101 Function is enabled with *n* = 16 0110 Function is enabled with *n* = 18 0111 Function is enabled with *n* = 20 1000 Function is enabled with *n* = 22 1001 Function is enabled with *n* = 24 1010 Function is enabled with *n* = 26 1011 Function is enabled with *n* = 28 1100 Function is enabled with *n* = 30 1101 Function is enabled with *n* = 32 1110 Function is enabled with *n* = 34 1111 Function is enabled with *n* = 36
4	Force Extended Flywheel	When set to ‘1’, forces the extended flywheel function to active (bit 0 of the packet qualifier byte [H0] on egress idle packets is forced to ‘1’). This bit is used for testing.
5	Extended Bitmap Enable	When set to ‘1’, enables the extended bitmap function (see *Section 3.4.2.2 Extended Bitmap* on page 32). This bit also enables the extended flywheel function (see *Section 3.4.2.3 Extended Flywheel* on page 32).
6	Best-Effort Discard Enable	1 Enables the input controllers to discard best-effort traffic, depending on the best-effort discard thresholds and the best-effort counter values. See *Section 3.5.6 Best-Effort Discard* on page 38. 0 Disables and clears the best-effort discard counters.
7	Best-Effort Drop Counters Enable	1 Enables the best-effort drop counters to count the discarded best-effort packets. 0 Disables and clears the best-effort drop counters.

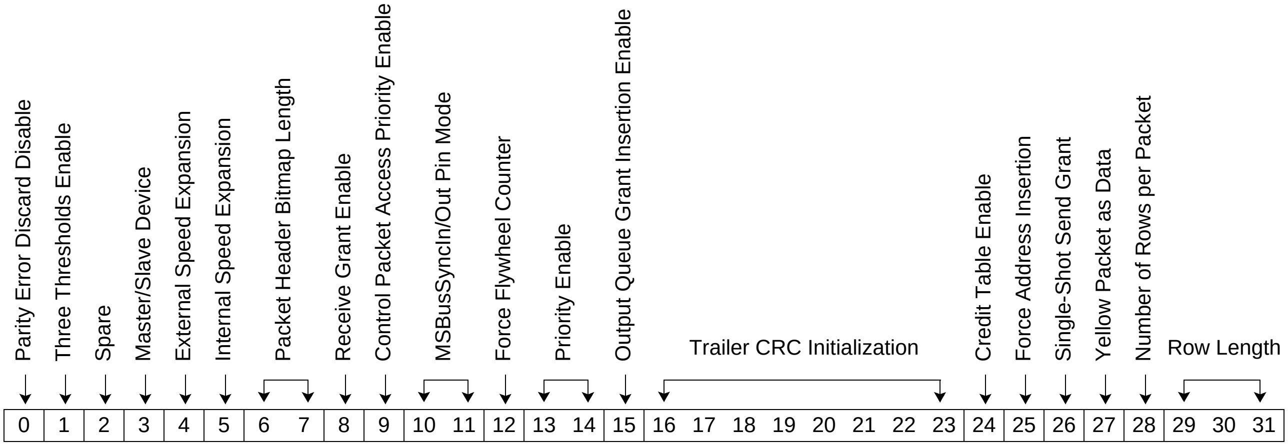
Bit(s)	Field Name	Description
8	Send Grant Antistreaming Enable	When set to ‘1’, enables the send grant antistreaming function under the parameters defined in the send grant antistreaming threshold field.
9:11	Send Grant Antistreaming Threshold	Provides protection when the attached device removes the send grant for an extended period. If the send grant is not issued for any priority for *n* contiguous packet cycles, this bit is internally forced to active for all priorities until the attached device issues another send grant: 000 *n* = 16 001 *n* = 32 010 *n* = 64 011 *n* = 128 100 *n* = 256 101 *n* = 512 110 *n* = 1024 111 *n* = 2048 The *Send Grant Violation Register* (page 97) indicates the ports for which send grant antistreaming is in progress.
12	Idle Color Force	1 Transmits all output idle packets with the color specified by the idle color bit, regardless of the expected color bit setting. When the color mechanism is not used, this bit must be set to ‘1’. 0 Allows the switchover mechanism to determine the color of output idle packets.
13	Idle Color	Specifies the color given to all idle packets when the idle color force bit is set: 0 Blue idle packets 1 Red idle packets
14	Expected Color	Specifies the expected color of incoming packets after a color clear command is initiated through the *Command Register* (page 100): 0 Blue packets 1 Red packets
15	Standby	Freezes the device while the configuration is reset. This bit is asserted after the power-on reset is completed and is released as described in *Section 6.2 Reset Sequence* on page 111.
16:23	Input Port Paralleling	For port paralleling, specifies which four input ports are grouped for single-link processing: Bit 16 set to ‘1’ Ports 0 through 3 are grouped Bit 17 set to ‘1’ Ports 4 through 7 are grouped Bit 18 set to ‘1’ Ports 8 through 11 are grouped Bit 19 set to ‘1’ Ports 12 through 15 are grouped Bit 20 set to ‘1’ Ports 16 through 19 are grouped Bit 21 set to ‘1’ Ports 20 through 23 are grouped Bit 22 set to ‘1’ Ports 24 through 27 are grouped Bit 23 set to ‘1’ Ports 28 through 31 are grouped
24:31	Output Port Paralleling	For port paralleling, specifies which four output ports are grouped for single-link processing. Encoding is identical to the input port-paralleling field.

5.4.2 Configuration 1 Register

Address x‘21’

Access Type Read/Write

Reset Value ‘0000 0000 0000 0000 0000 0000 0000 0000’



Bit(s)	Field Name	Description
0	Parity Error Discard Disable	When set to ‘1’, disables the input controllers from discarding the incoming packets with invalid header parity. The input controllers receive them as normal packets, but a header parity error interrupt is raised and the header parity error count field in the *Error Counter Register* (page 93) is incremented.
1	Three Thresholds Enable	1 Transmits the egress packet output queue grant information for priorities 0, 1, and 2 only. Once the grants for priority 2 are transmitted, the grants for priority 0 are transmitted in the next packet. Compared to four priorities, this reduces the output queue grant information update time. When this bit is set, the output queue thresholds for priorities 2 and 3 must be set to the same value and the shared memory thresholds for priorities 2 and 3 must be set to the same value. 0 Transmits the output queue grant information for all four priorities, if all the priorities are enabled (that is, if the priority enable field is set to ‘11’).
2	Spare	Spare.
3	Master/Slave Device	When two PowerPRS 64Gs are configured for external speed expansion, designates whether a device is the master or the slave: 1 Master device 0 Slave device
4	External Speed Expansion	When set to ‘1’, enables external speed expansion. This configuration bit must be specified for both the master device and the slave device.
5	Internal Speed Expansion	When set to ‘1’, enables internal speed expansion.
6:7	Packet Header Bitmap Length	Defines the length of the packet header bitmap: 00 One byte 01 Two bytes 10 Three bytes 11 Four bytes
8	Receive Grant Enable	When set to ‘1’, enables the receive grant mechanism.

Bit(s)	Field Name	Description
9	Control Packet Access Priority Enable	Guarantees that shared memory access is available to the local processor for read and write operations. When set to '1', the local processor has access to the shared memory for up to three LU lengths after the command is issued. This bit must be set only for row lengths of 16 or 32 bytes. For other row lengths, the sequencer guarantees shared memory access time to the local processor independent of this bit setting.
10:11	MSBusSyncIn/Out Pin Mode	Specifies MSBusSyncOut pin and MSBusSyncIn pin operation: 00 The MSBusSyncOut pin is tristated and the MSBusSyncIn pin is tristated. 01 The MSBusSyncOut pin is enabled and the MSBusSyncIn pin is tristated. 10 The MSBusSyncOut pin is tristated and the MSBusSyncIn pin is enabled. 11 Reserved The MSBusSyncIn pin is used to synchronize the internal sequencers of the master and slave devices configured for external speed expansion. Both the master and slave sequencers are fully synchronized and the MSBusSyncIn and MSBusSyncOut pins are fully synchronous to the internal logic. Note that: • When the PowerPRS 64G is configured for single device operation, all three possible settings are valid. • When the PowerPRS 64G is configured as a master device in external speed expansion, only the '01' and '10' settings are valid. • When the PowerPRS 64G is configured as a slave device in external speed expansion, only the '10' setting is valid.
12	Force Flywheel Counter	When set to '1', forces the flywheel counter to the value defined in the priority enable field. This bit is used for testing.
13:14	Priority Enable	Defines the number of priorities for which output queue grants are issued: 00 Priority 0 only is enabled 01 Priorities 0 and 1 are enabled 10 Priorities 0, 1, and 2 are enabled 11 Priorities 0, 1, 2, and 3 are enabled This bit controls the cycling of flywheels used to transmit grants to the attached devices (see *Section 3.4.2.3 Extended Flywheel* on page 32).
15	Output Queue Grant Insertion Enable	1 Inserts output queue grants in egress packet headers. 0 Does not insert output queue grants in egress packet headers. Egress data packet headers still contain the destination bitmap.
16:23	Trailer CRC Initialization	Specifies the initialization value to be used for the eight-bit idle packet trailer CRC check: LU Length — Trailer CRC Initialization Register in Hex 16 — D0 17 — DD 18 — 04 19 — FA 20 — 07 32 — 9B 34 — 67 36 — 2E 38 — 61 40 — 5A
24	Credit Table Enable	1 Enables the PowerPRS 64G output control logic to scan the credit table. 0 Disables the PowerPRS 64G output control logic from scanning the credit table. This allows the application to bypass credit table initialization, if the credit table is not used. For information about the credit table, see *Section 3.6.3 Credit Table* on page 42.
25	Force Address Insertion	When set to '1', enables the input controller to insert the input port number into the H1 header byte of ingress data packets. This bit is used for testing.
26	Single-Shot Send Grant	When set to '1', enables the *Force Send Grant Register* (page 91; when set to '1') to generate an internal send grant for a single packet of any priority. This bit is used for testing.
27	Yellow Packet as Data	1 Enables an incoming yellow packet to be processed as a data packet. The LU does not contain a CRC trailer byte. 0 Disables an incoming yellow packet from being processed as a data packet; the yellow packet is processed as an idle packet. The LU contains the CRC trailer byte, and the CRC checksum value is checked.

Bit(s)	Field Name	Description
28	Number of Rows per Packet	Specifies the number of rows contained in a logical unit: 0 One row 1 Two rows
29:31	Row Length	Specifies the row length in the shared memory: ‘000’ 16 bytes ‘001’ 17 bytes ‘010’ 18 bytes ‘011’ 19 bytes ‘100’ 20 bytes Others Reserved

5.4.3 Output Queue Enable Register

Address x‘22’

Access Type Read/Write

Reset Value ‘0000 0000 0000 0000 0000 0000 0000 0000’

Output Queue Enable (for port *n* = bit *n)*

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

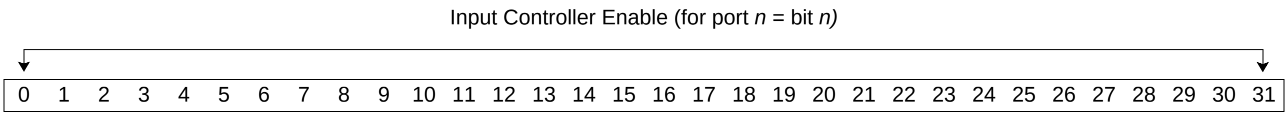
Bit(s)	Field Name	Description
0:31	Output Queue Enable (for port *n* = bit *n)*	1 Enables the output queue for each priority, and the packets destined for that output are enqueued. 0 Disables the output queue for each priority, and the following actions occur: 1. Unicast packets destined to a disabled output queue are discarded. Multicast packets are enqueued only in enabled output queues. 2. A disabled output queue is “slow flushed.” Addresses are dequeued and recycled as in normal operation. The slow flush takes place regardless of the SendGrantIn signal, as long as the queue is disabled. 3. Output queue grants corresponding to a disabled queue are forced to ‘1’. 4. Idle packets are transmitted if the corresponding bits are set in the *DASL Output Driver Enable Register* (page 72) and the *Output Port Enable Register* (page 72).

5.4.4 Input Controller Enable Register

Address x‘23’

Access Type Read/Write

Reset Value ‘0000 0000 0000 0000 0000 0000 0000 0000’



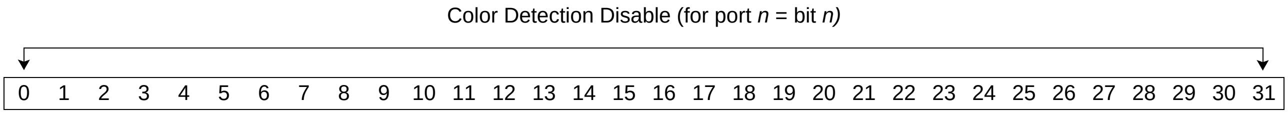
Bit(s)	Field Name	Description
0:31	Input Controller Enable (for port *n* = bit *n)*	1 Enables the port to receive data packets and control packets (normal setting). 0 Disables the port from receiving data packets and control packets. **Note:** This bit is required only for the master device.

5.4.5 Color Detection Disable Register

Address x‘24’

Access Type Read/Write

Reset Value ‘0000 0000 0000 0000 0000 0000 0000 0000’



Bit(s)	Field Name	Description
0:31	Color Detection Disable (for port *n* = bit *n)*	When set to ‘1’, disables the input port color detection mechanism and sets the corresponding bit in the *Expected Color Received Register* (page 92). See *Section 3.10 Switchover Support* on page 49.

5.4.6 Send Grant Enable Register

Address x‘25’

Access Type Read/Write

Reset Value ‘0000 0000 0000 0000 0000 0000 0000 0000’

Send Grant Enable (for port *n* = bit *n)*

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

Bit(s)	Field Name	Description
0:31	Send Grant Enable (for port *n* = bit *n)*	1 Enables normal send grant signal decoding and packet transmission on the port if the retrieved send grant information is active. 0 Disables packet transmission on the port regardless of the value of the send grant signal, unless the corresponding bit is set in the *Force Send Grant Register.*

5.4.7 Force Send Grant Register

Address x‘26’

Access Type Read/Write

Reset Value ‘0000 0000 0000 0000 0000 0000 0000 0000’

Force Send Grant (for port *n* = bit *n)*

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

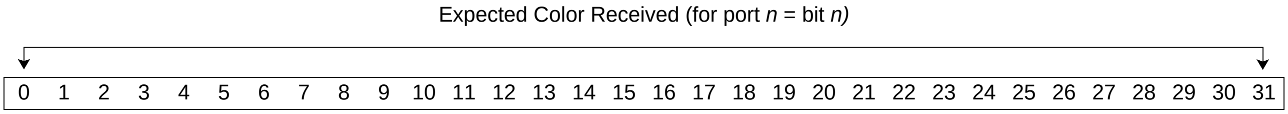
Bit(s)	Field Name	Description
0:31	Force Send Grant (for port *n* = bit *n)*	When set to ‘1’, forces the send grant pins to active for all the priorities on the port regardless of the SendGrantIn pin values. This register is used only when the corresponding bit is set to ‘0’ in the *Send Grant Enable Register.*

5.4.8 Expected Color Received Register

Address x‘27’

Access Type Read Only

Reset Value Undefined



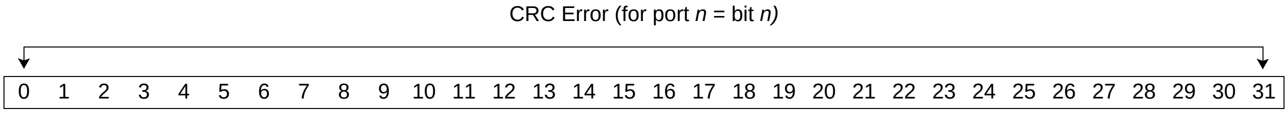
Bit(s)	Field Name	Description
0:31	Expected Color Received (for port *n* = bit *n)*	1 Either the expected color has been received on the input port since the last color clear command or the corresponding bit is set in the *Color Detection Disable Register* (page 90). 0 The color opposite of the expected color is being received on the input port.

5.4.9 CRC Error Register

Address x‘28’

Access Type Read/Clear

Reset Value ‘0000 0000 0000 0000 0000 0000 0000 000’



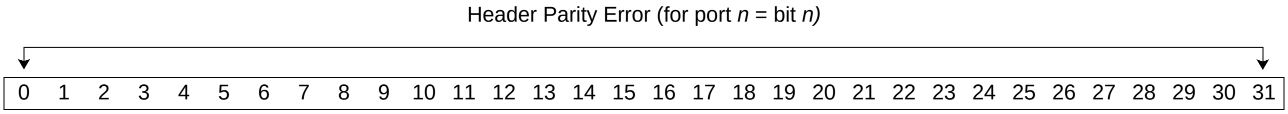
Bit(s)	Field Name	Description
0:31	CRC Error (for port *n* = bit *n)*	When set to ‘1’, the input port has detected a trailer CRC error within an idle packet. This register is cleared when read, but the errors detected during the read/clear are not lost.

5.4.10 Header Parity Error Register

Address x‘29’

Access Type Read/Clear

Reset Value ‘0000 0000 0000 0000 0000 0000 0000 0000’



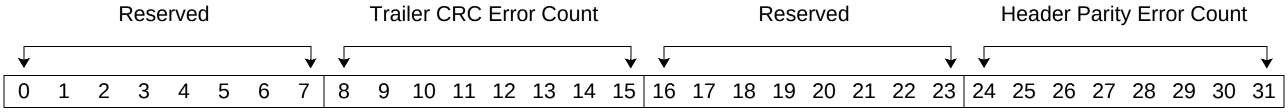
Bit(s)	Field Name	Description
0:31	Header Parity Error (for port *n* = bit *n)*	When set to ‘1’, the input port has detected a packet header parity error. This register is cleared when read, but the errors detected during the read/clear are not lost.

5.4.11 Error Counter Register

Address x‘2A’

Access Type Read/Clear

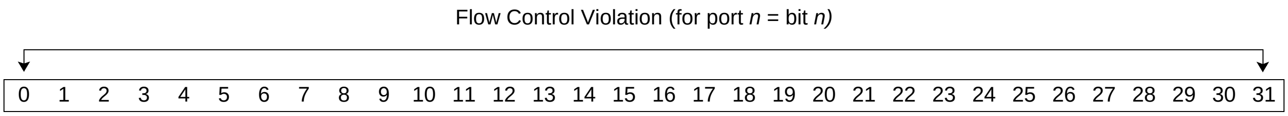
Reset Value ‘0000 0000 0000 0000 0000 0000 0000 0000’



Bit(s)	Field Name	Description
0:7	Reserved	Reserved.
8:15	Trailer CRC Error Count	Counts the CRC errors for all the input ports since the previous read and freezes when it reaches x‘FF’. This counter is cleared when read, but the errors detected during the read/clear are not lost.
16:23	Reserved	Reserved.
24:31	Header Parity Error Count	Counts the packet header parity errors for all the input ports since the previous read and freezes when it reaches x‘FF’. This counter is cleared when read, but the errors detected during the read/clear are not lost.

5.4.12 Flow Control Violation Register

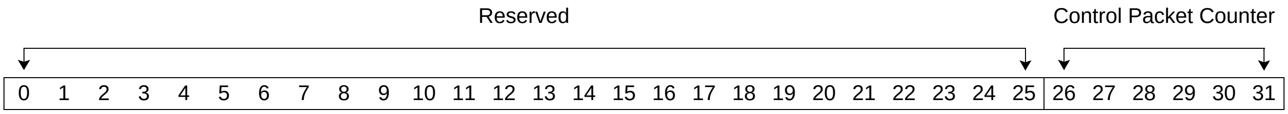
Address	x'2B'
Access Type	Read/Clear
Reset Value	'0000 0000 0000 0000 0000 0000 0000 0000'



Bit(s)	Field Name	Description
0:31	Flow Control Violation (for port *n* = bit *n*)	When set to '1', the input port has generated a flow control violation interrupt. This register is cleared when read, but the violations detected during the read/clear are not lost.

5.4.13 Control Packet Counter Register

Address	x'2C'
Access Type	Read Only
Reset Value	'0000 0000 0000 0000 0000 0000 0000 0000'

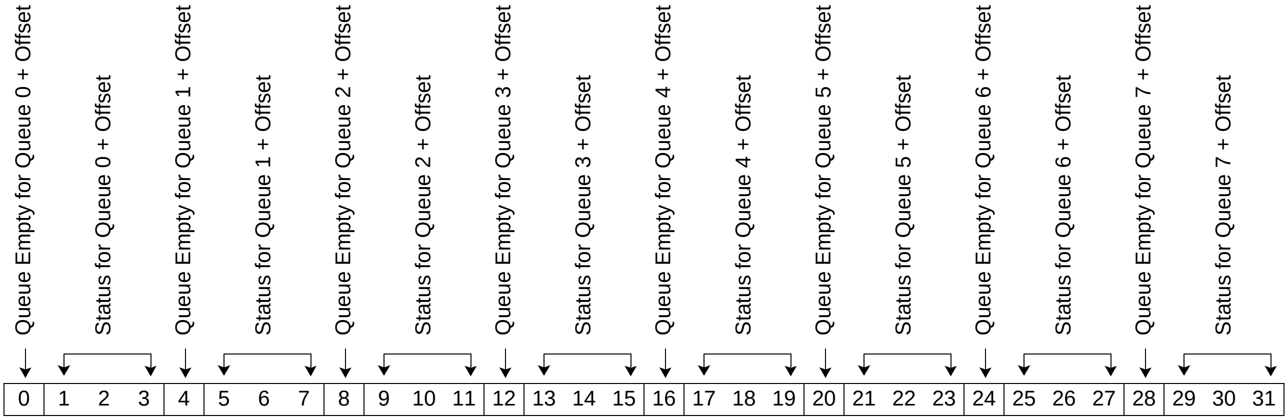


Bit(s)	Field Name	Description
0:25	Reserved	Reserved.
26:31	Control Packet Counter	Indicates the number of control packets currently in the control packet receive queue. The value ranges from 0 to 32.

5.4.14 Output Queue Status Registers

Four registers provide the status of the 32 output queues. Each of the four registers is composed of eight four-bit ranges; each four-bit range describes a single output queue status. Within the four-bit range for an output queue, the first bit is the output queue empty bit, which indicates whether the output queue has been emptied since the last read operation, and the other three bits comprise the status field, which indicates the highest occupancy level reached by the output queue since the last read operation.

Address	x‘2D’	Output queues 0 to 7	Offset = 0
	x‘2E’	Output queues 8 to 15	Offset = 8
	x‘2F’	Output queues 16 to 23	Offset = 16
	x‘30’	Output queues 24 to 31	Offset = 24
Access Type	Read/Clear		
Reset Value	‘1000 1000 1000 1000 1000 1000 1000 1000’		

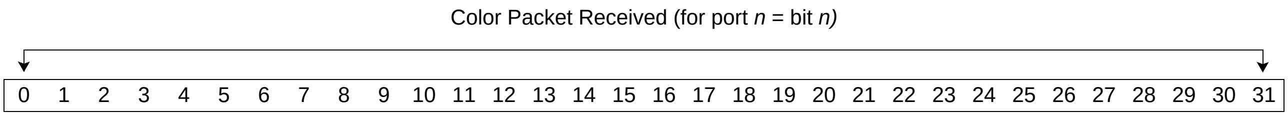


Bit(s)	Field Name	Description
0	Queue Empty for Queue 0 + Offset	When set to ‘1’, the output queue has been emptied since the last read.
1:3	Status for Queue 0 + Offset	Indicates that the total number of packets in the output queue has exceeded the threshold for the specified priority(ies). Each time a priority threshold of the output queue is exceeded, the status field for that output queue is updated. Note that the status field does not change when the number of packets of a particular priority falls below the threshold; this field continues to show the output queue status for the highest priority that has exceeded the threshold since the last read. 000 Output queue is not full for any priority 001 Output queue is full for priority 3 010 Output queue is full for priorities 2 and 3 011 Output queue is full for priorities 1, 2, and 3 100 Output queue is full for priorities 0, 1, 2, and 3 Others Reserved
4	Queue Empty for Queue 1 + Offset	See description for bit 0.
5:7	Status for Queue 1 + Offset	See description for bits 1:3.
8	Queue Empty for Queue 2 + Offset	See description for bit 0.
9:11	Status for Queue 2 + Offset	See description for bits 1:3.

Bit(s)	Field Name	Description
12	Queue Empty for Queue 3 + Offset	See description for bit 0.
13:15	Status for Queue 3 + Offset	See description for bits 1:3.
16	Queue Empty for Queue 4 + Offset	See description for bit 0.
17:19	Status for Queue 4 + Offset	See description for bits 1:3.
20	Queue Empty for Queue 5 + Offset	See description for bit 0.
21:23	Status for Queue 5 + Offset	See description for bits 1:3.
24	Queue Empty for Queue 6 + Offset	See description for bit 0.
25:27	Status for Queue 6 + Offset	See description for bits 1:3.
28	Queue Empty for Queue 7 + Offset	See description for bit 0.
29:31	Status for Queue 7 + Offset	See description for bits 1:3.

5.4.15 Color Packet Received Register

Address	x‘31’
Access Type	Read/Clear
Reset Value	Undefined



Bit(s)	Field Name	Description
0:31	Color Packet Received (for port *n* = bit *n)*	When set to ‘1’, indicates that the port has received at least one packet of the color specified in the color select field in the *Command Register* (page 100). This register is cleared when read, but the information detected during the read/clear is not lost.

5.4.16 Send Grant Violation Register

Address	x‘32’
Access Type	Read/Clear
Reset Value	‘0000 0000 0000 0000 0000 0000 0000 0000’

Send Grant Violation (for port *n* = bit *n)*
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

Bit(s)	Field Name	Description
0:31	Send Grant Violation (for port *n* = bit *n)*	When set to ‘1’, indicates that the port has removed the send grant for too long and send grant anti-streaming is in progress (see the send grant antistreaming enable and send grant antistreaming threshold fields in the *Configuration 0 Register* [page 85]). This register is cleared when read, but the violations detected during the read/clear are not lost.

5.4.17 Occupancy Counter Register

Address	x‘33’
Access Type	Read Only
Reset Value	‘0000 0000 0000 0000 0000 0000 0000 0000’

Reserved	Total Packet Count in Shared Memory	Reserved	Total Address Count in Free Queue
0 1 2 3 4	5 6 7 8 9 10 11 12 13 14 15	16 17 18 19 20	21 22 23 24 25 26 27 28 29 30 31

Bit(s)	Field Name	Description
0:4	Reserved	Reserved.
5:15	Total Packet Count in Shared Memory	Indicates the number of packets that currently occupy the shared memory. This field is continuously refreshed.
16:20	Reserved	Reserved.
21:31	Total Address Count in Free Queue	Indicates the current number of available addresses. This field is continuously refreshed.

5.4.18 Shared Memory Access Registers

The internal memory row register provides read/write access to any location in the shared memory. The internal memory row register is an internal five-word (32 bits per word) register accessed via the *Shared Memory Pointer Register* (page 99) and the *Shared Memory Data Register* (page 99).

To *write* a row to the shared memory:

1. Build the row to be loaded into the shared memory with five four-byte writes to the *Shared Memory Data Register:*
    a. The first write loads bytes 0 to 3 of the internal memory row register.
    b. The second write loads bytes 4 to 7 of the internal memory row register.
    c. The third write loads bytes 8 to 11of the internal memory row register.
    d. The fourth write loads 12 to 15 of the internal memory row register.
    e. The fifth write loads bytes 16 to 19 of the internal memory row register.
2. Write the *Shared Memory Pointer Register* with:
    - The write bit set to ‘1’,
    - The slave bank/master bank bit specifying the shared memory bank, and
    - Bits 21 to 31 specifying the row address.

    The internal memory row register content is loaded into the specified shared memory address.

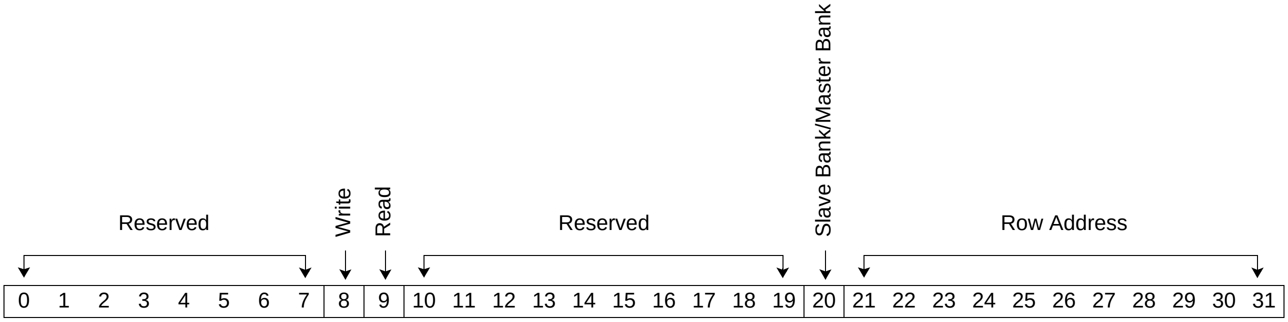
To *read* a row from the shared memory:

1. Write the *Shared Memory Pointer Register* with:
    - The read bit set to ‘1’,
    - The slave bank/master bank bit specifying the shared memory bank, and
    - Bits 21 to 31 specifying the row address.

    The shared memory content for the address is loaded into the internal memory row register.
2. Read the *Shared Memory Data Register* five times:
    a. The first read returns bytes 0 to 3 of the internal memory row register.
    b. The second read returns bytes 4 to 7 of the internal memory row register.
    c. The third read returns bytes 8 to 11 of the internal memory row register.
    d. The fourth read returns bytes 12 to 15 of the internal memory row register.
    e. The fifth read returns bytes 16 to 19 of the internal memory row register.

5.4.19 Shared Memory Pointer Register

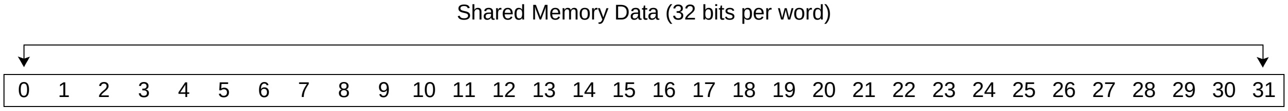
Address	x‘34’
Access Type	Read/Write
Reset Value	‘0000 0000 0000 0000 0000 0000 0000 0000’



Bit(s)	Field Name	Description
0:7	Reserved	Reserved.
8	Write	When set to ‘1’, specifies a write to the internal memory row register.
9	Read	When set to ‘1’, specifies a read from the internal memory row register.
10:19	Reserved	Reserved.
20	Slave Bank/ Master Bank	1 Specifies the slave shared memory bank. 0 Specifies the master shared memory bank.
21:31	Row Address	Specifies the row address (of the 2048 row addresses in the shared memory).

5.4.20 Shared Memory Data Register

Address	x‘35’
Access Type	Read/Write
Reset Value	‘0000 0000 0000 0000 0000 0000 0000 0000’

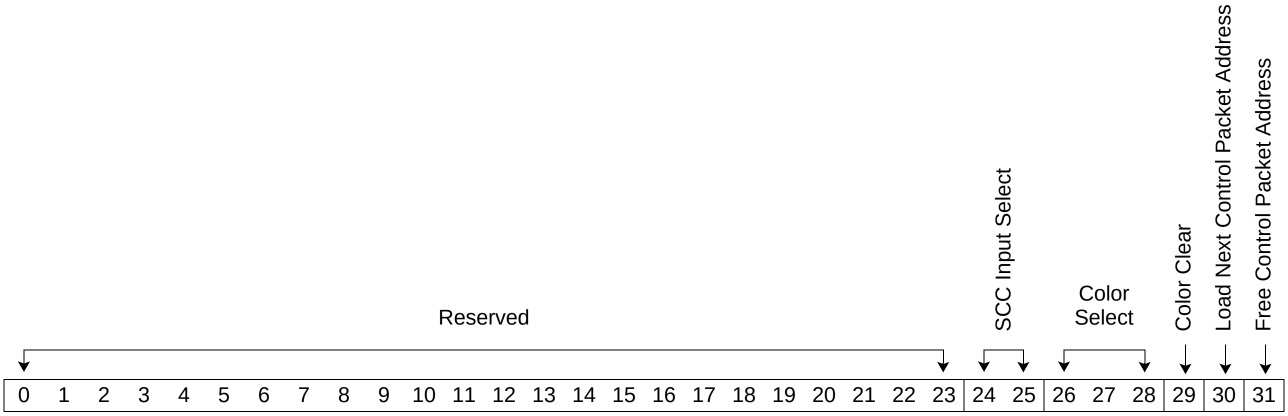


Bit(s)	Field Name	Description
0:31	Shared Memory Data (32 bits per word)	Contains the data that will be or has been exchanged with the internal memory row register.

5.4.21 Command Register

Fields in this register prompt the software to initiate specific actions. Note that bits 29 through 31 are interpreted as pulses; that is, a command is executed only when the register is written to. These bits (29:31) are not cleared after the command is executed.

Address	x‘36’
Access Type	Read/Write
Reset Value	‘0000 0000 0000 0000 0000 0000 0000 0000’



Bit(s)	Field Name	Description
0:23	Reserved	Reserved.
24:25	SCC Input Select	Selects the incoming SCC input bit reported in the *Side Communication Channel Input Reporting Register* (page 109): 00 SCC input bit 0 is reported 01 SCC input bit 1 is reported 10 SCC input bit 2 is reported 11 SCC input bit 3 is reported
26:28	Color Select	Selects the color reported in the *Color Packet Received Register* (page 96): 000 Blue 001 Red 010 Yellow type 1 011 Yellow type 2 100 Yellow type 3 101 Synchronization packet 110 Reserved 111 Reserved
29	Color Clear	When set to ‘1’, clears the idle packet color state machine. After this command is initiated, idle packets are transmitted with the color opposite that specified in the expected color bit in the *Configuration 0 Register* (page 85). Transmission on a given output port takes place only when both of the following conditions are satisfied: • At least one packet of the expected color has been received on all inputs. • The corresponding output queue is empty. If these conditions are not met, idle packets of the opposite color are transmitted on the output port.
30	Load Next Control Packet Address	When set to ‘1’, loads the content of the shared memory address at the top of the control packet queue into the internal memory row register. Note that the read address stays at the top of the control packet queue.

Bit(s)	Field Name	Description
31	Free Control Packet Address	When set to ‘1’, frees the shared memory address at the top of the control packet queue. The next available address moves to the top.

5.4.22 Control Packet Destination Register

This register provides a bitmap of the output ports to which the current control packet is destined.

Address	x‘37’
Access Type	Read/Write
Reset Value	‘0000 0000 0000 0000 0000 0000 0000 0000’

Control Packet Destination (for port *n* = bit *n)*

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

Bit(s)	Field Name	Description
0:31	Control Packet Destination (for port *n* = bit *n)*	When set to ‘1’, indicates that the control packet must be sent to the output port. When the control packet is sent over a port, the corresponding bit is cleared in the *Control Packet Destination Register.* When the control packet is sent to the last port, the register value returns to x‘0000 0000’, and an “all control packets transmitted” interrupt is generated. If for any reason the control packet is not sent to a port, the application can reset the corresponding register bit.

5.4.23 Bitmap Filter Register

Address	x‘38’
Access Type	Read/Write
Reset Value	‘1111 1111 1111 1111 1111 1111 1111 1111’

Bitmap Filter (for port *n* = bit *n)*

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

Bit(s)	Field Name	Description
0:31	Bitmap Filter (for port *n* = bit *n)*	Specifies the mask to apply to the received packet destination bitmap for switchover support and load balancing in redundant switch-plane operation. Application of the bitmap filter depends on the packet protection field (see *Table 3-3* on page 30). Note that this register cannot be written to while the device is in standby.

5.4.24 Threshold Access Register

This register provides indirect access to the internal threshold registers.

Write access to an internal threshold register requires one SHI command:

1. Write the *Threshold Access Register* with the write bit set to '1', the threshold select field specifying the internal threshold register to be written, and the threshold value field specifying the value.

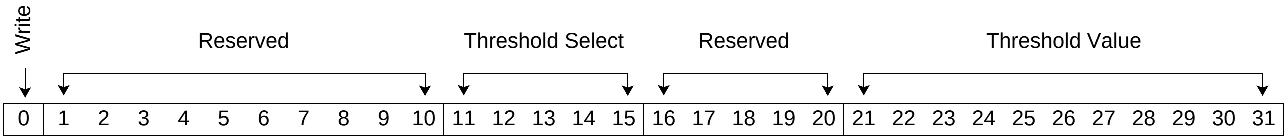
Read access to an internal threshold register requires two SHI commands:

1. Write the *Threshold Access Register* with the write bit set to '0' and the threshold select field specifying the internal threshold register to be read.
2. Read the *Threshold Access Register* to return the corresponding threshold value field value.

There are three types of internal thresholds:

- *Shared memory thresholds.* There are four shared memory thresholds, one per priority. Each shared memory threshold is composed of four 11-bit fields and has a value ranging from 0 to 2047. When the number of packets in the shared memory becomes equal to or greater than the threshold value for a given priority, the corresponding memory grant is low (‘0’); otherwise, the memory grant is high (‘1’).
- *Output queue thresholds.* There are four output queue thresholds, one per priority. Each output queue threshold is composed of four 11-bit fields and has a value ranging from 0 to 2047. When the number of packets in an output queue becomes equal to or greater than the threshold value for a given priority, the corresponding output queue grant is low (‘0’); otherwise, the output queue grant is high (‘1’).
- *Output queue thresholds for port paralleling.* These thresholds are identical to the output queue thresholds above but are used only for ports configured for port paralleling.

Address	x‘39’
Access Type	Read/Write
Reset Value	‘0000 0000 0000 0000 0000 0000 0000 0000’



Bit(s)	Field Name	Description
0	Write	1 Specifies a write to the internal threshold register. 0 Specifies a read from the internal threshold register.
1:10	Reserved	Reserved.

Bit(s)	Field Name	Description
11:15	Threshold Select	Specifies the internal threshold register to be read from or written to. Encoding to specify a shared memory threshold: 00000 Shared memory threshold register for priority 0 00001 Shared memory threshold register for priority 1 00010 Shared memory threshold register for priority 2 00011 Shared memory threshold register for priority 3 Encoding to specify an output queue threshold: 00100 Output queue threshold register for priority 0 00101 Output queue threshold register for priority 1 00110 Output queue threshold register for priority 2 00111 Output queue threshold register for priority 3 Encoding to specify an output queue threshold for a port configured for port paralleling: 01000 Output queue threshold register for priority 0 01001 Output queue threshold register for priority 1 01010 Output queue threshold register for priority 2 01011 Output queue threshold register for priority 3 Other values are reserved.
16:20	Reserved	Reserved.
21:31	Threshold Value	Contains the value written to, or the value read from, the specified internal threshold register.

5.4.25 Credit Table Access Register

This register provides indirect access to the credit table (see *Section 3.6.3 Credit Table* on page 42). There are 256 credits per port. Each port has 32 16-bit addresses, and each address contains eight credits. The one-credit fields designate the priority for which a credit is generated. Credits are generated during the packet cycle that corresponds to the credit number.

The credit table is not accessible while the PowerPRS 64G is in standby (that is, when the standby bit of the *Configuration 0 Register* [page 85] is set). Credit table access is allowed only while the corresponding port is active (that is, when the port is receiving/transmitting data).

Write access to the credit table requires one SHI command:

1. Write the *Credit Table Access Register* with the write bit set to ‘1’, the port number field specifying the output port number, the credit table address field specifying the credit table address to be written, and the eight one-credit fields set with eight credits.

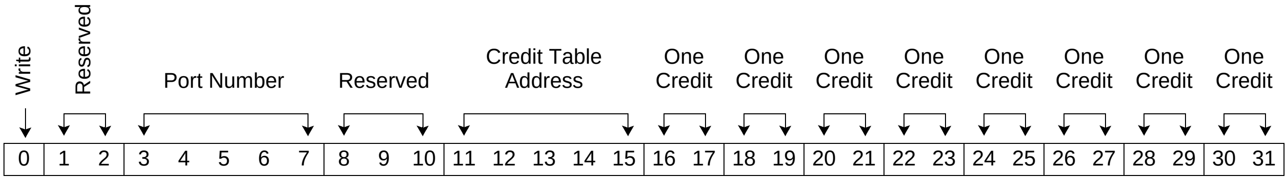
Read access to the credit table requires two SHI commands:

1. Write the *Credit Table Access Register* with the write bit set to ‘0’, the port number field specifying the port number, and the credit table address field specifying the credit table address to be read.
2. Read the *Credit Table Access Register* to return the eight corresponding one-credit field values.

Address x‘3A’

Access Type Read/Write

Reset Value ‘0000 0000 0000 0000 0000 0000 0000 0000’



Bit(s)	Field Name	Description
0	Write	1 Specifies a write to the credit table. 0 Specifies a read from the credit table.
1:2	Reserved	Reserved.
3:7	Port Number	Specifies the port number.
8:10	Reserved	Reserved.
11:15	Credit Table Address	Specifies the credit table address.
16:17	One Credit	Specifies the priority.
18:19	One Credit	Specifies the priority.
20:21	One Credit	Specifies the priority.
22:23	One Credit	Specifies the priority.
24:25	One Credit	Specifies the priority.
26:27	One Credit	Specifies the priority.

Bit(s)	Field Name	Description
28:29	One Credit	Specifies the priority.
30:31	One Credit	Specifies the priority.

5.4.26 Look-Up Table Access Register

This register provides indirect access to two look-up tables: the look-up table for the master data stream and the look-up table for the slave data stream. Each look-up table has 16 entries that specify how the first 16 data bytes of each data row of a byte stream are rearranged before transmission. The entry at location *n* of a look-up table specifies the data byte sent as the n^{th} byte in the data stream. The *Look-Up Table Access Register* reset value specifies the normal byte order, from 0 to 15 (no rearranging). The look-up tables cannot be written while the device is in standby.

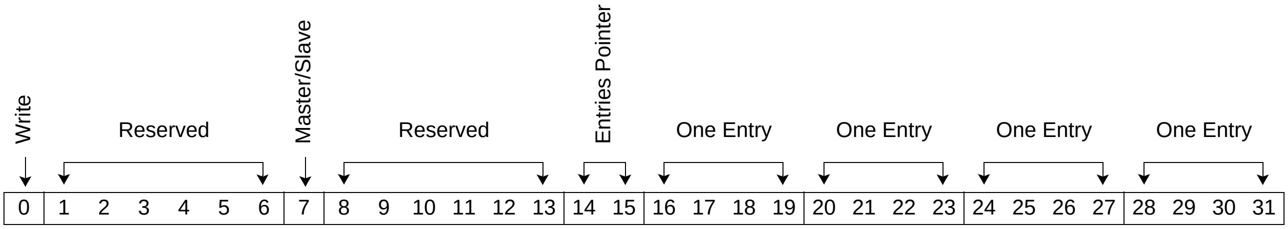
Write access to a look-up table requires one SHI command:

1. Write the *Look-Up Table Access Register* with the write bit 0 set to ‘1’, the master/slave field specifying the look-up table, the entries pointer field specifying the look-up table address, and the four one-entry fields set with the eight entries.

Read access to a look-up table requires two SHI commands:

1. Write the *Look-Up Table Access Register* with the write bit set to ‘0’, the master/slave field specifying the look-up table, and the entries pointer field specifying the look-up table address.
2. Read the *Look-Up Table Access Register* to return the four corresponding one-entry field values.

Address	x‘3B’
Access Type	Read/Write
Reset Value	‘0000 0000 0000 0000 0000 0001 0010 0011’



Bit(s)	Field Name	Description
0	Write	1 Specifies a write to the look-up table. 0 Specifies a read from the look-up table.
1:6	Reserved	Reserved.
7	Master/Slave	1 Specifies the master look-up table. 0 Specifies the slave look-up table.
8:13	Reserved	Reserved.
14:15	Entries Pointer	Specifies the look-up table address: 00 Bytes 0 to 3 01 Bytes 4 to 7 10 Bytes 8 to 11 11 Bytes 12 to 15 The values of the four selected entries are defined in bits 16:31.
16:19	One Entry	Look-up table byte 0, 4, 8, or 12, depending on the entries pointer field.
20:23	One Entry	Look-up table byte 1, 5, 9, or 13, depending on the entries pointer field.
24:27	One Entry	Look-up table byte 2, 6, 10, or 14, depending on the entries pointer field.

Bit(s)	Field Name	Description
28:31	One Entry	Look-up table byte 3, 7, 11, or 15, depending on the entries pointer field.

5.4.27 Bitmap Mapping Register

This register defines the mapping between a bit position in the packet header (bytes H1 to H4) and a physical output queue. That is, all the packets received with the bitmap bit specified by the logical bitmap bit position field set to ‘1’ are routed to the physical output queue specified by the physical queue field. For example, if the *Bitmap Mapping Register* is written with the logical bitmap bit position field set to 3 and the physical queue field set to 7, then all the packets received with bitmap bit 3 set to ‘1’ will be routed to physical output queue 7.

The *Bitmap Mapping Register* reset value is not rearranged (that is, bitmap bit *n* points to physical output queue *n).* This register cannot be written while the device is in standby.

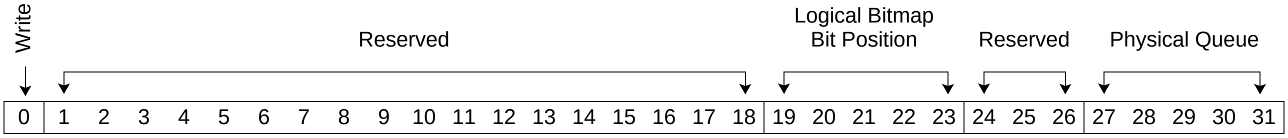
Write access to the *Bitmap Mapping Register* requires one SHI command:

1. Write the register with the write bit set to ‘1’, the logical bitmap bit position field specified, and the physical queue field specified.

Read access to the *Bitmap Mapping Register* requires two SHI commands:

1. Write the register with the write bit cleared to ‘0’ and the logical bitmap bit position field specified.
2. Read the register to return the corresponding physical queue field value.

Address	x‘3C’
Access Type	Read/Write
Reset Value	‘0000 0000 0000 0000 0000 0000 0000 0000’



Bit(s)	Field Name	Description
0	Write	1 Specifies a write to the *Bitmap Mapping Register.* 0 Specifies a read from the *Bitmap Mapping Register.*
1:18	Reserved	Reserved.
19:23	Logical Bitmap Bit Position	Specifies the logical bitmap bit position.
24:26	Reserved	Reserved.
27:31	Physical Queue	Specifies the physical output queue.

5.4.28 Best-Effort Resources Access Register

This register provides indirect access to the resources associated with the best-effort discard function (see *Section 3.5.6 Best-Effort Discard* on page 38).

Write access to the *Best-Effort Resources Access Register* requires one SHI command:

1. Write the register with the write bit set to ‘1’, the output port number field specified, the resource select field specified, and the resource value field specified.

Read access to the *Best-Effort Resources Access Register* requires two SHI commands:

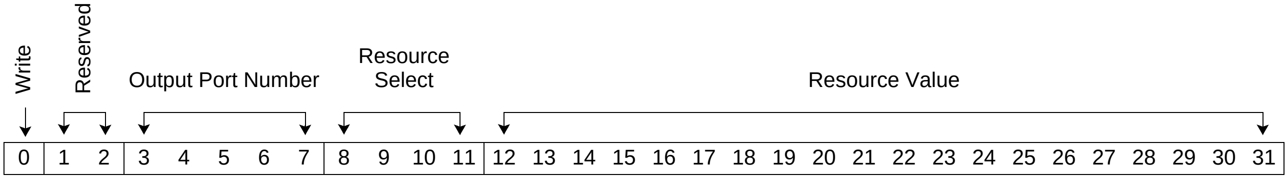
1. Write the register with the write bit cleared to ‘0’, the output port number field specified, and the resource select field specified.
2. Read the register to return the corresponding resource value field value.

Note: To write to this register, the best-effort discard function must be enabled (using the best-effort discard enable bit of the *Configuration 0 Register* [page 85]).

Address x‘3D’

Access Type Read/Write

Reset Value ‘0000 0000 0000 0000 0000 0000 0000 0000’



Bit(s)	Field Name	Description
0	Write	1 Specifies a write to the *Best-Effort Resources Access Register.* 0 Specifies a read from the *Best-Effort Resources Access Register*.
1:2	Reserved	Reserved.
3:7	Output Port Number	Specifies the output port number.
8:11	Resource Select	Specifies the resource to be accessed: 0000 Best-effort discard counter 1 0001 Best-effort discard counter 2 0010 Best-effort discard counter 3 0011 Best-effort discard counter 4 0100 Best-effort discard counter 5 0101 Best-effort enable discard threshold 0110 Best-effort priority discard threshold 0111 Best-effort halt discard threshold 1000 Best-effort drop counter for priority 0 1001 Best-effort drop counter for priority 1 1010 Best-effort drop counter for priority 2 1011 Best-effort drop counter for priority 3 1100 Protocol engine virtual packet clock (in this case, only bits 22:31 are valid) Others Reserved

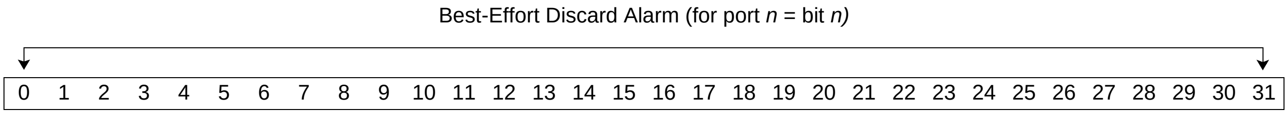
Bit(s)	Field Name	Description
12:31	Resource Value	When the resource select field specifies a counter, indicates the counter value. When the resource select field specifies the protocol engine virtual packet clock, the encoded value of this field is: x'00000' No protocol engine packet clock (protocol engine packet clock frequency is equal to 0). x‘00001' Single-shot protocol engine packet clock. This generates a pulse on the internal protocol engine packet clock (used for testing). x‘00002’ Reserved. x‘00003’ Reserved. Any other value x‘*nnnnn’* The protocol engine packet clock has a period of *“nnnnn”* × 8 ns.

5.4.29 Best-Effort Discard Alarm Register

Address x‘3E’

Access Type Read Only

Reset Value ‘0000 0000 0000 0000 0000 0000 0000 0000’



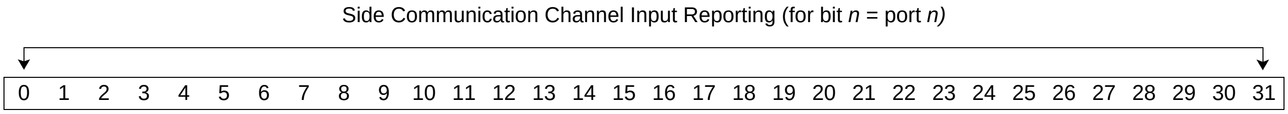
Bit(s)	Field Name	Description
0:31	Best-Effort Discard Alarm (for port *n* = bit *n)*	When set to ‘1’, the best-effort discard logic is in a phase that allows best-effort packets to be discarded for the port.

5.4.30 Side Communication Channel Input Reporting Register

Address x‘3F’

Access Type Read Only

Reset Value Undefined



Bit(s)	Field Name	Description
0:31	Side Communication Channel Input Reporting (for bit *n* = port *n)*	Indicates the port information extracted from idle packet SCC input bit *y,* where *y* equals 0, 1, 2, or 3 and is set with the SCC input select field in the *Command Register* (page 100).

6. Reset, Initialization, and Operation

6.1 Clock and PLL

A reference clock must be provided on the OscillatorIn input pin. The internal byte clock frequency will be twice the reference clock frequency, based on the *PLL Programming Register* (page 63) setting.

6.2 Reset Sequence

This sequence must be executed after system power-up.

1. Activate the PowerOnResetIn# input pin and start the serial host interface (SHI) clock.
2. Deactivate the PowerOnResetIn# input pin after at least three SHI clock cycles.
3. Write the tune, range A, range B, and multiplier fields in the *PLL Programming Register* with the required values, and then release the phase-locked loop (PLL) reset bit.
4. Wait 500 µs.
5. Release the flush bit in the *Reset Register* (page 64).
6. Release the shared data-aligned synchronous link (DASL) controller 0 reset bit and the shared DASL controller 1 reset bit in the *Reset Register.*
7. Load the picocode.
8. Release the picoprocessor 0 and 1 reset bits in the *Reset Register.*
9. Write the *Configuration 1 Register* (page 87) to specify part of the device configuration.
10. Write the *Configuration 0 Register* (page 85) to specify the rest of the device configuration, and then release the standby bit.
11. Read the *Status Register* (page 59) to clear all the interrupts.
12. If necessary, release the global interrupt mask bit in the *Reset Register.*
13. Set the output driver enable bit in the *Reset Register.*

6.3 DASL Synchronization and Operation

After the PowerPRS 64G has been fully configured, the DASL interfaces between the 64G and the attached devices must be synchronized to provide bit phase alignment and packet delineation at the data receivers in both directions. Data traffic cannot be exchanged between the PowerPRS 64G and the attached devices until synchronization is complete.

Port synchronization is under the overall control of the system control processor, which coordinates operations between the switch core and the attached devices. For a single PowerPRS 64G, port synchronization is handled by the local processor (which runs the switch control microcode and is connected to the 64G via the SHI). Port synchronization can also be performed directly through interface lines between the local processor and the attached devices.

Input ports must be synchronized:

- After device reset and initialization
- When the local processor resynchronizes a link due to data errors on the incoming packets

The following registers are required for this activity:

- *DASL Output Driver Enable Register*
- *DASL Signal Lost Register*
- *Synchronization Packet Transmit Register*
- *Output Port Enable Register*
- *Input Port Enable Register*
- *DASL Synchronization Hunt Register*
- *DASL Synchronization Status Register*
- *Output Queue Enable Register*
- *Input Controller Enable Register*

6.3.1 Synchronizing DASL Ports

Synchronization steps are required for both the PowerPRS 64G port and the attached device. Note that synchronization may be performed on one port at a time or on multiple ports simultaneously.

To synchronize an input port and enable a connection:

1. Enable the DASL output driver by writing a ‘1’ in the *DASL Output Driver Enable Register* (page 72).
2. To ensure the integrity of the serial links, use the *DASL Signal Lost Register* (page 74) to check the validity of the connection between the receiver and a differential transmitter.
3. Enable synchronization packet transmission by writing a ‘1’ in the *Synchronization Packet Transmit Register* (page 73).
4. Enable the output port by writing a ‘1’ in the *Output Port Enable Register* (page 72).
5. Enable the input port by writing a ‘1’ in the *Input Port Enable Register* (page 73).
6. Start synchronization on the enabled ports by writing a ‘1’ in the *DASL Synchronization Hunt Register* (page 75).
7. After a synchronization time-out period, poll the *DASL Synchronization Status Register* (page 75) to verify that synchronization has been attained. The *DASL Synchronization Status Register* bit will be set to ‘0’ for any port that failed to synchronize.
8. When synchronization has been attained, the local processor and the remote device report to the system control processor that they are ready for data transfer. Clear the *DASL Synchronization Hunt Register.*
9. Read the *CRC Error Register* (page 92) and the *Error Counter Register* (page 93) to clear any cyclic redundancy check (CRC) error indications that might have been set during synchronization.
10. Enable the output queue by writing a ‘1’ in the *Output Queue Enable Register* (page 89).
11. Enable the input controller by writing a ‘1’ in the *Input Controller Enable Register* (page 90).
12. Because the link is now in data mode, the local processor must poll the *CRC Error Register* and the *DASL Signal Lost Register* to check for error-free operation.

6.3.2 Stopping DASL Port Synchronization

To stop input synchronization and disable a connection:

1. Disable the input controller by writing a ‘0’ in the *Input Controller Enable Register* (page 90).
2. Disable the input port by writing a ‘0’ in the *Input Port Enable Register* (page 73).
3. Poll the *DASL Synchronization Status Register* (page 75) to verify that synchronization was lost after the input port was disabled.
4. Disable the output queue by writing a ‘0’ in the *Output Queue Enable Register* (page 89).
5. Disable the DASL output driver by writing a ‘0’ in the *DASL Output Driver Enable Register* (page 72).
6. Disable the output port by writing a ‘0’ in the *Output Port Enable Register* (page 72).

6.4 Logic BIST Execution Sequence

1. Activate the PowerOnResetIn# input pin and start the SHI clock.
2. Deactivate the PowerOnResetIn# input pin after at least three SHI clock cycles.
3. Write the tune, range A, range B, and multiplier fields in the *PLL Programming Register* (page 63) with the required values, and then release the PLL reset bit.
4. Wait 500 μs.
5. Set the logic built-in self-test (BIST) requested bit in the *Reset Register* (page 64).
6. Set the pseudorandom pattern generator (PRPG)/multiple-input signature register (MISR) data field in the *BIST Data Register* (page 67) with a defined value.
7. Load the *BIST Counter Register* (page 66).
8. Set the shift speed and scan chain length fields in the *BIST Select Register* (page 68).
9. Release the flush bit in the *Reset Register.*
10. Poll the *Status Register* (page 59) until the logic BIST done bit is set.
11. Read the PRPG/MISR data field in the *BIST Data Register.*
12. Set the flush bit and release the logic BIST requested bit in the *Reset Register.*

6.5 Memory BIST Execution Sequence

1. Activate the PowerOnResetIn# input pin and start the SHI clock.
2. Deactivate the PowerOnResetIn# input pin after at least three SHI clock cycles.
3. Write the tune, range A, range B, and multiplier fields in the *PLL Programming Register* with the required values, and then release the PLL reset bit.
4. Wait 500 µs.
5. Set the memory BIST requested bit in the *Reset Register.*
6. Release the flush bit in the *Reset Register.*

7. Poll the *Status Register* (page 59) until the memory BIST done bit is set.
8. Verify that the memory BIST fail bit in the *Status Register* is off.
9. Set the flush bit and release the memory BIST requested bit in the *Reset Register* (page 64).

7. I/O Definitions and I/O Timing

7.1 I/O Definitions

Note: Nondifferential signals are active high unless there is a # symbol at the end of the signal name, in which case the signal is active low. Differential pairs are designated by an _P for the positive signal and an _N for the negative signal at the end of the signal name.

Table 7-1. Signal Definitions (Page 1 of 4)

Signal Name	Type	I/O Cell Name	Description
Clock and Reset Signals			
OscillatorIn	Input	AINST2_PM_A	System clock used for the internal clock generation network. OscillatorIn frequency is 62.5 MHz (half the internal byte clock frequency) with a duty cycle of 50 percent.
Osc125MhzOut_P Osc125MhzOut_N	Output Differential	ODASL_PM_A	Enabled by the debug bus enable field in the *Debug Bus Select Register* (page 69). Free-running B clock phase of the internal byte clock at the output of the internal clock tree. The correct frequency of the clock output is twice that of OscillatorIn.
Osc250MhzOut_P Osc250MhzOut_N	Output Differential	ODASL_PM_A	Enabled by the debug bus enable field in the *Debug Bus Select Register.* Free-running 250-MHz clock.
Osc500MhzOut_P Osc500MhzOut_N	Output Differential	ODASL_PM_A	Enabled by the debug bus enable field in the *Debug Bus Select Register.* Data-aligned synchronous link (DASL) phase-locked loop (PLL) bit clock.
PowerOnResetIn#	Input	BP2550_PM_A	Must be active for at least four serial host interface (SHI) clock cycles. Asserting this pin causes a reset of all internal logic, except the IEEE Standard 1149.1 (JTAG) logic block. For information about the reset sequence, see *Section 6. Reset, Initialization, and Operation* on page 111.
MSBusPacketClockBidi	Bidirectional	BC2550_PM_C	Low-frequency clock used to synchronize the internal sequencers of different devices. When two PowerPRS 64Gs are configured for external speed expansion, this is a synchronous signal generated by the master device and fed to the slave device. This clock period is equal to the logical unit (LU) time with a duty cycle of 50 percent. The operating mode of this pin is programmable via the MSBusSyncIn/Out pin mode field in the *Configuration 1 Register* (page 87). This pin can be tristated for single-device configuration, or generated or received by the device.
FullyInsertedIn#	Input	BP2550_PM_A	Used to set the DASL drivers in high impedance until the board housing the PowerPRS 64G is fully inserted. This ensures that both ends of the board are fully inserted. An external pullup resistor is required to force the inactive state when the board is correctly inserted.
Data Signals			
DaslData[00:31]In[0:3]_P DaslData[00:31]In[0:3]_N	Input Differential	IDASL_PM_A	DaslData[*p*]In[*n*]_P and DaslData[*p*]In[*n*]_N form one of the four 500-Mbps differential signal pairs for input port *p*. For each port, pairs 0 and 1 carry the slave byte stream, and pairs 2 and 3 carry the master byte stream.

Table 7-1. Signal Definitions (Page 2 of 4)

Signal Name	Type	I/O Cell Name	Description
DaslData[00:31]Out[0:3]_P DaslData[00:31]Out[0:3]_N	Output Differential	ODASL_PM_A	DaslData[*p*]Out[*n*]_P and DaslData[*p*]Out[*n*]_N form one of the four 500-Mbps differential signal pairs for output port *p*. For each port, pairs 0 and 1 carry the slave byte stream, and pairs 2 and 3 carry the master byte stream.
MemoryGrantOut[0:3]	Output	BP2550_PM_A	Provides the status of the memory grant for priority *n*. The MemoryGrantOut[*n*] pins are updated every four clock cycles (clock cycles are 8 to 10 ns). The device pin encoding and corresponding grant definition are as follows: 0000 No grant 1000 Priority 0 or control packets 1100 Priority 0 or 1 or control packets 1110 Priority 0, 1, or 2 or control packets 1111 Priority 0, 1, 2, or 3 or control packets Others Reserved See *Section 3.5.2 Memory Grants* on page 37 for more information. When two PowerPRS 64Gs are configured for external speed expansion, only the master device MemoryGrantOut[0:3] bus is used.
SendGrantIn[0:31]	Input	BP2550_PM_B	Grants output ports the opportunity to transmit packets. When SendGrantIn[*n*] is active, packets may be transmitted on port *n.* When SendGrantIn[*n*] is inactive, only idle packets (and no data packets) may be transmitted on port *n.* During data packet transmission, the highest priority available packet is transmitted unless the credit table specifies otherwise. When two PowerPRS 64Gs are configured for external speed expansion, the SendGrantIn[0:31] bus of the slave device is not used and is disconnected.
ReceiveGrantIn[0:31]	Input	BP2550_PM_B	Grants output ports the opportunity to receive packets. Incoming packets are received for output *n* only if ReceiveGrantIn[*n*] is active. If ReceiveGrantIn[*n*] is inactive, packets intended for output *n* are not enqueued. When two PowerPRS 64Gs are configured for external speed expansion, the ReceiveGrantIn[0:31] bus of the slave device is not used and is disconnected.
Master/Slave Speed Expansion Signals These I/Os are connected only when two PowerPRS 64Gs are configured for external speed expansion. Note that these signals are synchronous to the internal master/slave clocks and point to point between the two devices.			
MSBusSyncOut	Output	BC2550_PM_C	A synchronous one-clock-cycle (8-ns to 10-ns) pulse that frames the information multiplexed on the MSBusInputAddrBidi[0:25] and MSBusOutputAddrBidi[0:25] buses. This pin connects directly to the MSBusSyncIn pin of the other device. For the basic PowerPRS 64G configuration (one device with no external speed expansion), this signal is tristated. The MSBusSyncOut/MSBusSyncIn operating mode is selected by the MSBusSyncIn/Out pin mode field in the *Configuration 1 Register* (page 87).
MSBusSyncIn	Input	BC2550_PM_C	Connects directly to the MSBusSyncOut pin of the other PowerPRS 64G, or to another external device.

Table 7-1. Signal Definitions (Page 3 of 4)

Signal Name	Type	I/O Cell Name	Description
MSBusInputAddrBidi[0:25]	Bidirectional	BC2550_PM_C	Multiplexed bus that connects the master device and the slave device to provide packet store addresses to the slave input controllers. MSBusInputAddrBidi[0:12] transmits store addresses for ports 1 to 15; MSBusInputAddrBidi[13:25] transmits store addresses for ports 16 to 32: MSBusInputAddrBidi[0:10] Eleven-bit address value field MSBusInputAddrBidi[11] Valid bit of the address MSBusInputAddrBidi[12] Odd parity bit over MSBusInputAddrBidi[0:11] MSBusInputAddrBidi[13:23] Eleven-bit address value field MSBusInputAddrBidi[24] Valid bit of the address MSBusInputAddrBidi[25] Odd parity bit over MSBusInputAddrBidi[13:24] MSBusInputAddrBidi is synchronized with MSBusSyncOut. The addresses for ports 0 and 16 are carried when MSBusSyncOut is high; the addresses for ports *n* and *n* + 16 are carried *n* clock cycles after MSBusSync is high. The master port uses the address and valid bit values to convey incoming packet handling instructions to the slave ports. <u>Bits 0:10 (Address)</u> <u>Bit 11 (Valid Bit)</u> <u>Slave Action</u> Zero value 1 Ignores the data packet Non-zero value 1 Receives the data packet Any value 0 Treats the incoming packet as an idle packet For the basic PowerPRS 64G configuration (one device with no external speed expansion), this bus is tristated.
MSBusOutputAddrBidi[0:25]	Bidirectional	BC2550_PM_C	Multiplexed bus that connects the master device and the slave device to provide packet read addresses to the slave output controllers. MSBusOutputAddrBidi[0:12] transmits read addresses for ports 1 to 15; MSBusOutputAddrBidi[13:25] transmits read addresses for ports 16 to 32. MSBusOutputAddrBidi[0:10] Eleven-bit address value field MSBusOutputAddrBidi[11] Valid bit of the address MSBusOutputAddrBidi[12] Odd parity bit over MSBusOutputAddrBidi[0:11] MSBusOutputAddrBidi[13:23] Eleven-bit address value field MSBusOutputAddrBidi[24] Valid bit of the address MSBusOutputAddrBidi[25] Odd parity bit over MSBusOutputAddrBidi[13:24] MSBusOutputAddrBidi is synchronized with MSBusSyncOut. The addresses for ports 0 and 16 are carried when MSBusSyncOut is high; the addresses for ports *n* and *n* + 16 are carried *n* clock cycles after MSBusSyncOut is high. For the basic PowerPRS 64G configuration (one device with no external speed expansion), this bus is tristated.
SHI Interface Signals			
SHIClockIn	Input	BP2550_PM_A	Free-running clock line that generates the SHI clock.
InterruptOut#	Output	BP2550_PM_A	Used to generate interrupts to the local processor. The signal remains asserted until an SHI read status command is executed. To support a wired-OR configuration, InterruptOut# uses an open-drain driver and is in the high-impedance state when inactive.
SHISerialDataIn	Input	BP2550_PM_A	Serial data line that shifts into the *SHI Instruction Register* (page 57).

Table 7-1. Signal Definitions (Page 4 of 4)

Signal Name	Type	I/O Cell Name	Description
SHISerialDataOut	Output	BP2550_PM_A	Serial data line that shifts out of the *SHI Instruction Register* (page 57). SHISerialDataOut is placed in high-impedance state when the SHI is not in shift state. The SHI is in shift state one SHI clock cycle after SHISelectIn# becomes inactive.
SHISelectIn#	Input	BP2550_PM_A	Enables SHI operation. One SHI clock cycle after the SHISelectIn# signal becomes active, the instruction is serially shifted into the *SHI Instruction Register.*
Debug Bus Signals			
DebugDataOut[0:15]	Output	BC2550_PM_C	Sixteen-bit bus that provides direct I/O access (logic analyzer) to the debug bus specified by the *Debug Bus Select Register* (page 69).
Digital Inputs			
SCCIn[0:3]	Input	BP2550_PM_B	The side communication channel (SCC) signal line that allows communication between the attached devices and the PowerPRS 64G.

Table 7-2. Test Signals (Page 1 of 2)

Signal Name	I/O Cell Name	Description
LSSD_SCAN_IN[0:20]	IC18PDT_PM_A	Scan chain inputs.
LSSD_SCAN_IN[21:23]	IC18PUT_PM_A	Scan chain inputs: • JTAG Test Data Input (TDI) = LSSD_SCAN_IN[21] • JTAG Test Mode Select (TMS) = LSSD_SCAN_IN[22] • JTAG Test Clock (TCK) = LSSD_SCAN_IN[23]
LSSD_SCAN_OUT[0:23]	BP2550T_PM_A	Scan chain outputs: • Memory BIST DIAG_OUT = LSSD_SCAN_OUT[20] • PLL_LOCK = LSSD_SCAN_OUT[21] • PLL_TESTOUT = LSSD_SCAN_OUT[22] • JTAG Test Data Output (TDO) = LSSD_SCAN_OUT[23]
LSSD_SCAN_MODE	IC18PDT_PM_A	Allows all clocks to be controlled from the primary inputs, and connects all scan chains.
LSSD_A_CLK	IC18PUT_PM_A	Used as an external source for the internal set/reset latch (SRL) A clock.
LSSD_B1_CLK	IC18PUT_PM_A	Used as an external source for the internal SRL B clock.
LSSD_C1_CLK	IC18PUT_PM_A	Used as an external source for the internal SRL C clock.
LSSD_C2_CLK	IC18PDT_PM_A	Used as an external source for the internal static random access memory (SRAM) C clock.
LSSD_C3_CLK	IC18PUT_PM_A	Used as an external source for the internal global register array (GRA) C clock.
LSSD_C4_CLK	IC18PUT_PM_A	Used as an external source for the internal GRA C clock (four ports are GRA only).
DI1#	IP25D1T_PM_A	Inhibits the driver for all nontest outputs. Active low.
DI2#	IC18D2PUT_PM_A	Inhibits the driver for all test outputs. Active low.

Table 7-2. Test Signals (Page 2 of 2)

Signal Name	I/O Cell Name	Description
RI#	IC18RIT_PM_A	Inhibits the receiver for all inputs. An external pullup resistor to 1.8 V is required on this pin. Active low.
TRST	IC18PDT_PM_A	JTAG test reset.
IOTEST	IC18PDT_PM_A	Used for reduced pin count testing. Allows all level-sensitive scan design (LSSD) boundary inputs to drive out signals: • LSSD_SCAN_GATE = IOTEST • BISTTESTM1 = IOTEST BISTGATE = IOTEST
LSSD_TAP_C1	IC18PUT_PM_A	JTAG controller C1 clock.
LSSD_TAP_C2	IC18PUT_PM_A	JTAG controller C2 clock.
BISTTESTM3	IC18PDT_PM_A	Used to handle the internal memory built-in self-test (BIST) controllers. The internal signal PLL_TESTIN is the logical AND of IOTEST and the inverse of BISTTESTM3.
LeakageTest	IC18LTPUT_PM_A	Used during the leakage test.
PLL_VDDA	AINSD2_PM_A	PLL analog voltage.
DELAYIn	BP2550_PM_A	The internal delay element input used for process measurement. The internal delay element is built with a chain of 300 INVERT_O gates.
DELAYOut	BP2550_PM_A	The internal delay element output used for process measurement: • 11 ns minimum (process = -3 sigma, temperature = 0 °C, voltage = 2.7 V) • 28 ns maximum (process = +3 sigma, temperature = 125 °C, voltage = 2.3 V)
PLL_GNDA	AINSD2_PM_A	PLL analog ground. (This is a new pin for this version of the device.)

7.2 I/O Timing

7.2.1 DASL Signals

The skew requirements for the data-aligned synchronous links (DASLs) are presented in *Table 7-3.*

Table 7-3. DASL Skew

Parameter	Rating	Units	Notes
Maximum skew between the two lines of a differential pair	±130	ps	
Maximum skew between two 500-Mbps links to the same port (this also applies to any two ports configured for speed expansion)	±2	clock cycles (8 to 10 ns)	1
1. Clock cycle = 8 ns (125-MHz operation) to 10 ns (100-MHz operation).			

7.2.2 SHI Signals

Figure 7-1. SHI Signal Timing Diagram

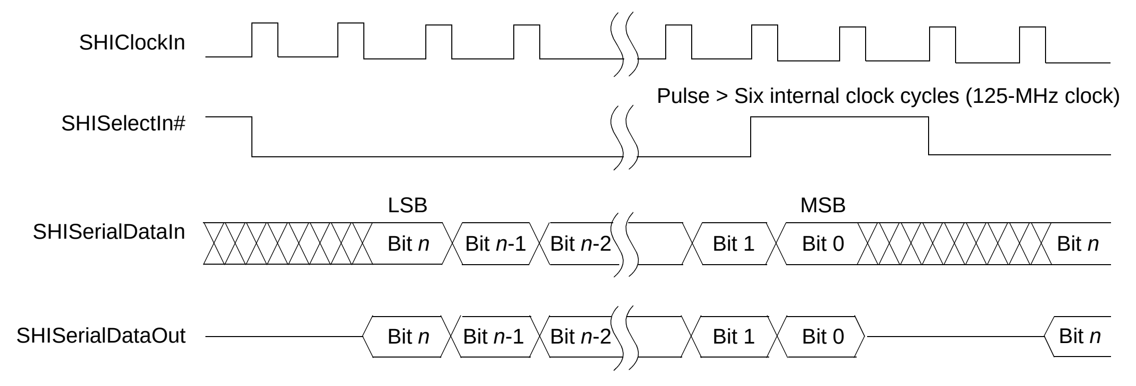


Figure 7-2. SHI Signal-to-Clock Timing Diagram

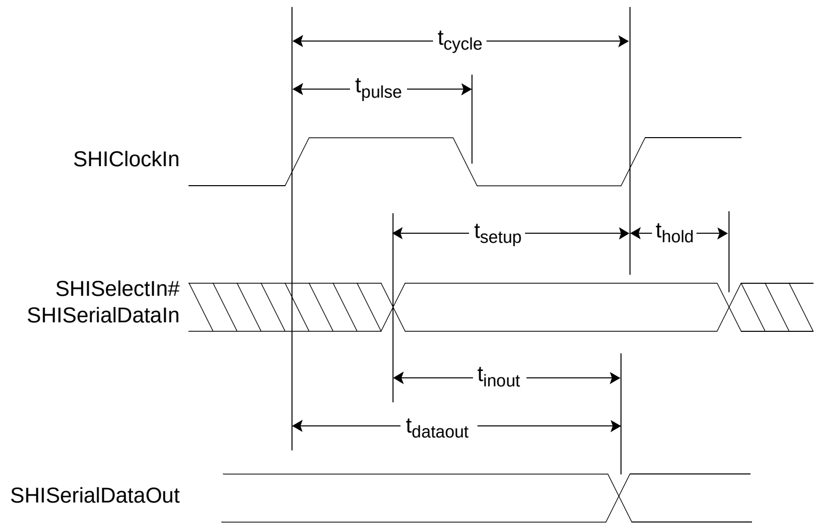


Table 7-4. SHI Signal Timing Values

Symbol	Parameter	Rating		Units
		Minimum	Maximum	
t_{cycle}	Cycle time	16		ns
t_{pulse}	Pulse width	4		ns
t_{setup}	Setup time	4		ns
t_{hold}	Hold time	5		ns
t_{inout}	SHISelectIn# to SHISerialDataOut	2	6	ns
$t_{dataout}$	SHIClockIn to SHISerialDataOut	3	8	ns

8. Data and Flow Control Latencies

8.1 Data Packet Transmission

Table 8-1 shows the difference between the time an ingress data-aligned synchronous link (DASL) *receives* the first bit of a packet and the time an egress DASL *sends* the first bit of the same packet, assuming the packet is not enqueued for flow control.

Table 8-1. Data Packet Transmission

Shared Memory Required	Time (clock cycles)	
	Minimum	Maximum
One row per packet	3 × row size	5 × row size
Two rows per packet	3 × row size	6 × row size
Note: Row size = 16 to 20 clock cycles (8 ns).		

8.2 Send Grant Off to Egress Idle Packet

Table 8-2 shows the difference between the time an attached device removes the send grant and the time the first idle packet exits the PowerPRS 64G according to this flow control mechanism.

Table 8-2. Send Grant Off to Egress Idle Packet

Shared Memory Required	Time (clock cycles)	
	Minimum	Maximum
One row per packet	3	3 + row size
Two rows per packet	3	3 + (2 × row size)
Note: Row size = 16 to 20 clock cycles, and clock cycle = 8 ns (125-MHz operation).		

8.3 Ingress Data Packet Received to Output Queue Grant Off

Table 8-3 shows the difference between the time an ingress data packet that causes the output queue occupancy to exceed the output queue threshold is received (and requires the output queue grant to be turned off) and the time the attached device receives the updated output queue grant status (which turns off the output queue grant).

Table 8-3. Ingress Data Packet Received to Output Queue Grant Off

Shared Memory Required	Time (clock cycles)	
	Minimum	Maximum
One row per packet	27	27 + (6 × row size)
Note: Row size = 16 to 20 clock cycles, and clock cycle = 8 ns (125-MHz operation). This must take into account the flywheel counters used for output queue grant synchronization.		

Table 8-3. Ingress Data Packet Received to Output Queue Grant Off

Shared Memory Required	Time (clock cycles)	
	Minimum	Maximum
Two rows per packet	27	27 + (11 × row size)
Note: Row size = 16 to 20 clock cycles, and clock cycle = 8 ns (125-MHz operation). This must take into account the flywheel counters used for output queue grant synchronization.		

8.4 Ingress Data Packet Received to Memory Grant Off

Table 8-4 shows the difference between the time an ingress data packet that causes the shared memory occupancy to exceed the shared memory threshold is received (and requires the memory grant to be turned off) and the time the attached device receives the updated memory grant status (which turns off the memory grant).

Table 8-4. Ingress Data Packet to Memory Grant Off

Shared Memory Required	Time (clock cycles)	
	Minimum	Maximum
One or two rows per packet	30	30 + row size
Note: Row size = 16 to 20 clock cycles, and clock cycle = 8 ns (125-MHz operation).		

9. Pin Information

Figure 9-1. Pinout (1088-ball ceramic column grid array [CCGA] package, bottom view)

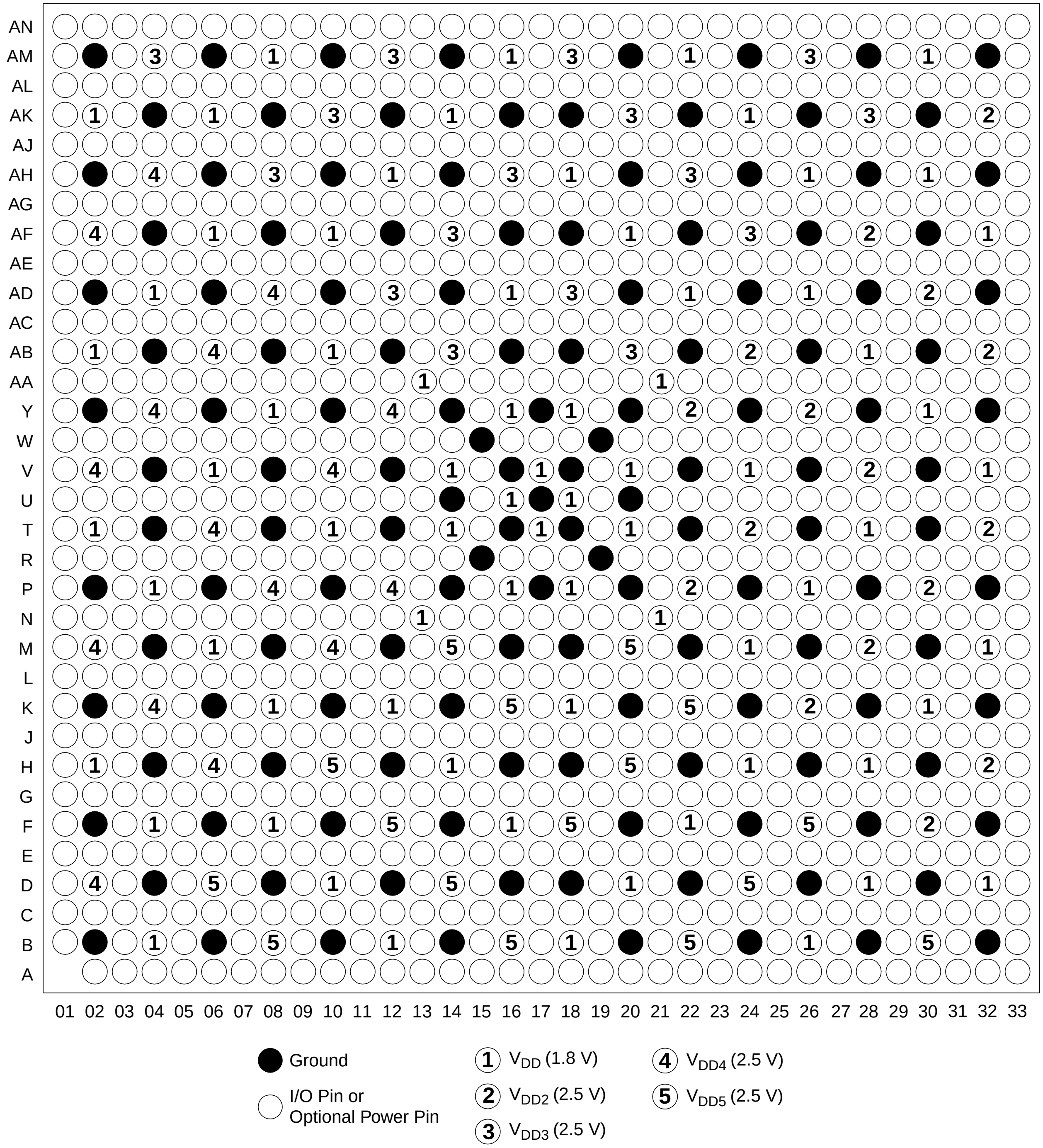


Table 9-1. Ground and V_{DD} Pin Locations

Pin Function	Pin Locations
GND	AB04, AB08, AB12, AB16, AB18, AB22, AB26, AB30, AD02, AD06, AD10, AD14, AD20, AD24, AD28, AD32, AF04, AF08, AF12, AF16, AF18, AF22, AF26, AF30, AH02, AH06, AH10, AH14, AH20, AH24, AH28, AH32, AK04, AK08, AK12, AK16, AK18, AK22, AK26, AK30, AM02, AM06, AM10, AM14, AM20, AM24, AM28, AM32, B02, B06, B10, B14, B20, B24, B28, B32, D04, D08, D12, D16, D18, D22, D26, D30, F02, F06, F10, F14, F20, F24, F28, F32, H04, H08, H12, H16, H18, H22, H26, H30, K02, K06, K10, K14, K20, K24, K28, K32, M04, M08, M12, M16, M18, M22, M26, M30, P02, P06, P10, P14, P17, P20, P24, P28, P32, R15, R19, T04, T08, T12, T16, T18, T22, T26, T30, U14, U17, U20, V04, V08, V12, V16, V18, V22, V26, V30, W15, W19, Y02, Y06, Y10, Y14, Y17, Y20, Y24, Y28, Y32
V_{DD}	AA13, AA21, AB02, AB10, AB28, AD04, AD16, AD22, AD26, AF06, AF10, AF20, AF32, AH12, AH18, AH26, AH30, AK02, AK06, AK14, AK24, AM08, AM16, AM22, AM30, B04, B12, B18, B26, D10, D20, D28, D32, F04, F08, F16, F22, H02, H14, H24, H28, K08, K12, K18, K30, M06, M24, M32, N13, N21, P04, P16, P18, P26, T02, T10, T14, T17, T20, T28, U16, U18, V06, V14, V17, V20, V24, V32, Y08, Y16, Y18, Y30
V_{DD2}	AB24, AB32, AD30, AF28, AK32, F30, H32, K26, M28, P22, P30, T24, T32, V28, Y22, Y26
V_{DD3}	AB14, AB20, AD12, AD18, AF14, AF24, AH08, AH16, AH22, AK10, AK20, AK28, AM04, AM12, AM18, AM26
V_{DD4}	AB06, AD08, AF02, AH04, D02, H06, K04, M02, M10, P08, P12, T06, V02, V10, Y04, Y12
V_{DD5}	B08, B16, B22, B30, D06, D14, D24, F12, F18, F26, H10, H20, K16, K22, M14, M20

Table 9-2. Optional Power Pins (Recommended)

Pin Function	Pin Locations
GND_A	AA15, AA16, AA19, AA29, AB21, AB31, AC19, AE13, AE21, AE25, AE27, AN33, G07, K13, K17, L20, M11, M13, M21, M23, N15, N17, N19, N23, P21, P29, R11, R14, R16, R23, U11, U21, V13, W17, W31, Y11, Y13, Y23
VDD250_A	AC13, AC20, AE07, AG27, AL13, C13, G25, L14, L19, P11, P23, R13, R18, U12, W12, W18, Y21
VDD_A (1.8 V)	AA22, AB29, AC22, AD17, AN01, G09, J25, J27, L11, L21, N09, N16, R17, R21, R22, R27, U15, U24, U28, W25, W29, Y19, Y27

Table 9-3. I/O Signal List, Sorted by Signal Name (Page 1 of 8)

Signal Name	Pin Location	Signal Name	Pin Location	Signal Name	Pin Location
BISTTESTM3	AJ01	DaslData[01]Out[3]_N	AE28	DaslData[03]Out[2]_P	AJ24
DaslData[00]In[0]_N	AD33	DaslData[01]Out[3]_P	AE29	DaslData[03]Out[3]_N	AJ25
DaslData[00]In[0]_P	AE33	DaslData[02]In[0]_N	AK29	DaslData[03]Out[3]_P	AH25
DaslData[00]In[1]_N	AD31	DaslData[02]In[0]_P	AL30	DaslData[04]In[0]_N	AN25
DaslData[00]In[1]_P	AE31	DaslData[02]In[1]_N	AM33	DaslData[04]In[0]_P	AN24
DaslData[00]In[2]_N	AE32	DaslData[02]In[1]_P	AL32	DaslData[04]In[1]_N	AL25
DaslData[00]In[2]_P	AF33	DaslData[02]In[2]_N	AM29	DaslData[04]In[1]_P	AL24
DaslData[00]In[3]_N	AF31	DaslData[02]In[2]_P	AN30	DaslData[04]In[2]_N	AM23
DaslData[00]In[3]_P	AE30	DaslData[02]In[3]_N	AM31	DaslData[04]In[2]_P	AN23
DaslData[00]Out[0]_N	AA26	DaslData[02]In[3]_P	AN32	DaslData[04]In[3]_N	AL23
DaslData[00]Out[0]_P	AB27	DaslData[02]Out[0]_N	AH29	DaslData[04]In[3]_P	AK23
DaslData[00]Out[1]_N	AB25	DaslData[02]Out[0]_P	AG28	DaslData[04]Out[0]_N	AG23
DaslData[00]Out[1]_P	AC27	DaslData[02]Out[1]_N	AG29	DaslData[04]Out[0]_P	AE22
DaslData[00]Out[2]_N	AC26	DaslData[02]Out[1]_P	AF29	DaslData[04]Out[1]_N	AE23
DaslData[00]Out[2]_P	AC25	DaslData[02]Out[2]_N	AJ26	DaslData[04]Out[1]_P	AF23
DaslData[00]Out[3]_N	AC29	DaslData[02]Out[2]_P	AJ27	DaslData[04]Out[2]_N	AG22
DaslData[00]Out[3]_P	AC28	DaslData[02]Out[3]_N	AH27	DaslData[04]Out[2]_P	AF21
DaslData[01]In[0]_N	AG30	DaslData[02]Out[3]_P	AJ28	DaslData[04]Out[3]_N	AH23
DaslData[01]In[0]_P	AH31	DaslData[03]In[0]_N	AM27	DaslData[04]Out[3]_P	AJ23
DaslData[01]In[1]_N	AH33	DaslData[03]In[0]_P	AN28	DaslData[05]In[0]_N	AN22
DaslData[01]In[1]_P	AG32	DaslData[03]In[1]_N	AL28	DaslData[05]In[0]_P	AM21
DaslData[01]In[2]_N	AK31	DaslData[03]In[1]_P	AK27	DaslData[05]In[1]_N	AK21
DaslData[01]In[2]_P	AJ30	DaslData[03]In[2]_N	AN26	DaslData[05]In[1]_P	AL22
DaslData[01]In[3]_N	AK33	DaslData[03]In[2]_P	AM25	DaslData[05]In[2]_N	AN21
DaslData[01]In[3]_P	AJ32	DaslData[03]In[3]_N	AK25	DaslData[05]In[2]_P	AN20
DaslData[01]Out[0]_N	AD25	DaslData[03]In[3]_P	AL26	DaslData[05]In[3]_N	AL21
DaslData[01]Out[0]_P	AC24	DaslData[03]Out[0]_N	AD23	DaslData[05]In[3]_P	AL20
DaslData[01]Out[1]_N	AD29	DaslData[03]Out[0]_P	AE24	DaslData[05]Out[0]_N	AG21
DaslData[01]Out[1]_P	AD27	DaslData[03]Out[1]_N	AG26	DaslData[05]Out[0]_P	AJ22
DaslData[01]Out[2]_N	AE26	DaslData[03]Out[1]_P	AF25	DaslData[05]Out[1]_N	AH21
DaslData[01]Out[2]_P	AF27	DaslData[03]Out[2]_N	AG24	DaslData[05]Out[1]_P	AJ21

Table 9-3. I/O Signal List, Sorted by Signal Name (Page 2 of 8)

Signal Name	Pin Location	Signal Name	Pin Location	Signal Name	Pin Location
DaslData[05]Out[2]_N	AG20	DaslData[07]Out[2]_N	AF15	DaslData[09]Out[2]_N	AE12
DaslData[05]Out[2]_P	AE20	DaslData[07]Out[2]_P	AG15	DaslData[09]Out[2]_P	AG11
DaslData[05]Out[3]_N	AJ20	DaslData[07]Out[3]_N	AJ16	DaslData[09]Out[3]_N	AF11
DaslData[05]Out[3]_P	AH19	DaslData[07]Out[3]_P	AJ15	DaslData[09]Out[3]_P	AE11
DaslData[06]In[0]_N	AM19	DaslData[08]In[0]_N	AM13	DaslData[10]In[0]_N	AN06
DaslData[06]In[0]_P	AN19	DaslData[08]In[0]_P	AN12	DaslData[10]In[0]_P	AM07
DaslData[06]In[1]_N	AK19	DaslData[08]In[1]_N	AL12	DaslData[10]In[1]_N	AK07
DaslData[06]In[1]_P	AL19	DaslData[08]In[1]_P	AK13	DaslData[10]In[1]_P	AL06
DaslData[06]In[2]_N	AL18	DaslData[08]In[2]_N	AN11	DaslData[10]In[2]_N	AN04
DaslData[06]In[2]_P	AN18	DaslData[08]In[2]_P	AM11	DaslData[10]In[2]_P	AM05
DaslData[06]In[3]_N	AL17	DaslData[08]In[3]_N	AK11	DaslData[10]In[3]_N	AL04
DaslData[06]In[3]_P	AN17	DaslData[08]In[3]_P	AL11	DaslData[10]In[3]_P	AK05
DaslData[06]Out[0]_N	AG19	DaslData[08]Out[0]_N	AE14	DaslData[10]Out[0]_N	AJ10
DaslData[06]Out[0]_P	AF19	DaslData[08]Out[0]_P	AG14	DaslData[10]Out[0]_P	AG10
DaslData[06]Out[1]_N	AJ19	DaslData[08]Out[1]_N	AH15	DaslData[10]Out[1]_N	AH09
DaslData[06]Out[1]_P	AJ18	DaslData[08]Out[1]_P	AJ14	DaslData[10]Out[1]_P	AJ09
DaslData[06]Out[2]_N	AD19	DaslData[08]Out[2]_N	AJ12	DaslData[10]Out[2]_N	AE10
DaslData[06]Out[2]_P	AE19	DaslData[08]Out[2]_P	AG13	DaslData[10]Out[2]_P	AD11
DaslData[06]Out[3]_N	AF17	DaslData[08]Out[3]_N	AJ13	DaslData[10]Out[3]_N	AF09
DaslData[06]Out[3]_P	AJ17	DaslData[08]Out[3]_P	AH13	DaslData[10]Out[3]_P	AG08
DaslData[07]In[0]_N	AM17	DaslData[09]In[0]_N	AN10	DaslData[11]In[0]_N	AJ04
DaslData[07]In[0]_P	AK17	DaslData[09]In[0]_P	AN09	DaslData[11]In[0]_P	AK03
DaslData[07]In[1]_N	AN16	DaslData[09]In[1]_N	AL10	DaslData[11]In[1]_N	AN02
DaslData[07]In[1]_P	AL16	DaslData[09]In[1]_P	AL09	DaslData[11]In[1]_P	AM03
DaslData[07]In[2]_N	AN15	DaslData[09]In[2]_N	AM09	DaslData[11]In[2]_N	AJ02
DaslData[07]In[2]_P	AM15	DaslData[09]In[2]_P	AN08	DaslData[11]In[2]_P	AK01
DaslData[07]In[3]_N	AL15	DaslData[09]In[3]_N	AL08	DaslData[11]In[3]_N	AL02
DaslData[07]In[3]_P	AK15	DaslData[09]In[3]_P	AK09	DaslData[11]In[3]_P	AM01
DaslData[07]Out[0]_N	AE15	DaslData[09]Out[0]_N	AF13	DaslData[11]Out[0]_N	AJ07
DaslData[07]Out[0]_P	AD15	DaslData[09]Out[0]_P	AG12	DaslData[11]Out[0]_P	AJ08
DaslData[07]Out[1]_N	AH17	DaslData[09]Out[1]_N	AJ11	DaslData[11]Out[1]_N	AJ06
DaslData[07]Out[1]_P	AG17	DaslData[09]Out[1]_P	AH11	DaslData[11]Out[1]_P	AH07

Table 9-3. I/O Signal List, Sorted by Signal Name (Page 3 of 8)

Signal Name	Pin Location	Signal Name	Pin Location	Signal Name	Pin Location
DaslData[11]Out[2]_N	AG06	DaslData[13]Out[2]_N	AB07	DaslData[15]Out[2]_N	W10
DaslData[11]Out[2]_P	AH05	DaslData[13]Out[2]_P	AA08	DaslData[15]Out[2]_P	W09
DaslData[11]Out[3]_N	AF05	DaslData[13]Out[3]_N	AC06	DaslData[15]Out[3]_N	U06
DaslData[11]Out[3]_P	AG05	DaslData[13]Out[3]_P	AC05	DaslData[15]Out[3]_P	U08
DaslData[12]In[0]_N	AG02	DaslData[14]In[0]_N	AB01	DaslData[16]In[0]_N	U02
DaslData[12]In[0]_P	AH01	DaslData[14]In[0]_P	AA02	DaslData[16]In[0]_P	U04
DaslData[12]In[1]_N	AH03	DaslData[14]In[1]_N	AA04	DaslData[16]In[1]_N	T01
DaslData[12]In[1]_P	AG04	DaslData[14]In[1]_P	AB03	DaslData[16]In[1]_P	T03
DaslData[12]In[2]_N	AF01	DaslData[14]In[2]_N	AA01	DaslData[16]In[2]_N	R01
DaslData[12]In[2]_P	AE02	DaslData[14]In[2]_P	Y01	DaslData[16]In[2]_P	R02
DaslData[12]In[3]_N	AE04	DaslData[14]In[3]_N	AA03	DaslData[16]In[3]_N	R03
DaslData[12]In[3]_P	AF03	DaslData[14]In[3]_P	Y03	DaslData[16]In[3]_P	R04
DaslData[12]Out[0]_N	AF07	DaslData[14]Out[0]_N	AA07	DaslData[16]Out[0]_N	R09
DaslData[12]Out[0]_P	AE08	DaslData[14]Out[0]_P	AB05	DaslData[16]Out[0]_P	R10
DaslData[12]Out[1]_N	AE05	DaslData[14]Out[1]_N	AA06	DaslData[16]Out[1]_N	U07
DaslData[12]Out[1]_P	AE06	DaslData[14]Out[1]_P	AA05	DaslData[16]Out[1]_P	U05
DaslData[12]Out[2]_N	AC10	DaslData[14]Out[2]_N	Y07	DaslData[16]Out[2]_N	R08
DaslData[12]Out[2]_P	AD09	DaslData[14]Out[2]_P	Y09	DaslData[16]Out[2]_P	R07
DaslData[12]Out[3]_N	AD07	DaslData[14]Out[3]_N	Y05	DaslData[16]Out[3]_N	T05
DaslData[12]Out[3]_P	AD05	DaslData[14]Out[3]_P	W06	DaslData[16]Out[3]_P	R05
DaslData[13]In[0]_N	AE01	DaslData[15]In[0]_N	W02	DaslData[17]In[0]_N	P01
DaslData[13]In[0]_P	AD01	DaslData[15]In[0]_P	W01	DaslData[17]In[0]_P	N01
DaslData[13]In[1]_N	AE03	DaslData[15]In[1]_N	W04	DaslData[17]In[1]_N	P03
DaslData[13]In[1]_P	AD03	DaslData[15]In[1]_P	W03	DaslData[17]In[1]_P	N03
DaslData[13]In[2]_N	AC02	DaslData[15]In[2]_N	V03	DaslData[17]In[2]_N	N02
DaslData[13]In[2]_P	AC01	DaslData[15]In[2]_P	V01	DaslData[17]In[2]_P	M01
DaslData[13]In[3]_N	AC03	DaslData[15]In[3]_N	U03	DaslData[17]In[3]_N	M03
DaslData[13]In[3]_P	AC04	DaslData[15]In[3]_P	U01	DaslData[17]In[3]_P	N04
DaslData[13]Out[0]_N	AC07	DaslData[15]Out[0]_N	W07	DaslData[17]Out[0]_N	P09
DaslData[13]Out[0]_P	AB09	DaslData[15]Out[0]_P	W08	DaslData[17]Out[0]_P	P07
DaslData[13]Out[1]_N	AC09	DaslData[15]Out[1]_N	W05	DaslData[17]Out[1]_N	R06
DaslData[13]Out[1]_P	AC08	DaslData[15]Out[1]_P	V05	DaslData[17]Out[1]_P	P05

Table 9-3. I/O Signal List, Sorted by Signal Name (Page 4 of 8)

Signal Name	Pin Location	Signal Name	Pin Location	Signal Name	Pin Location
DaslData[17]Out[2]_N	M05	DaslData[19]Out[2]_N	J08	DaslData[21]Out[2]_N	G10
DaslData[17]Out[2]_P	N07	DaslData[19]Out[2]_P	H07	DaslData[21]Out[2]_P	E10
DaslData[17]Out[3]_N	N05	DaslData[19]Out[3]_N	J06	DaslData[21]Out[3]_N	E09
DaslData[17]Out[3]_P	N06	DaslData[19]Out[3]_P	J05	DaslData[21]Out[3]_P	F09
DaslData[18]In[0]_N	L01	DaslData[20]In[0]_N	D01	DaslData[22]In[0]_N	A08
DaslData[18]In[0]_P	L02	DaslData[20]In[0]_P	E02	DaslData[22]In[0]_P	B09
DaslData[18]In[1]_N	L04	DaslData[20]In[1]_N	B01	DaslData[22]In[1]_N	D09
DaslData[18]In[1]_P	L03	DaslData[20]In[1]_P	C02	DaslData[22]In[1]_P	C08
DaslData[18]In[2]_N	K01	DaslData[20]In[2]_N	D03	DaslData[22]In[2]_N	A09
DaslData[18]In[2]_P	J01	DaslData[20]In[2]_P	E04	DaslData[22]In[2]_P	A10
DaslData[18]In[3]_N	K03	DaslData[20]In[3]_N	B03	DaslData[22]In[3]_N	C09
DaslData[18]In[3]_P	J03	DaslData[20]In[3]_P	A02	DaslData[22]In[3]_P	C10
DaslData[18]Out[0]_N	N08	DaslData[20]Out[0]_N	F05	DaslData[22]Out[0]_N	G11
DaslData[18]Out[0]_P	M07	DaslData[20]Out[0]_P	G06	DaslData[22]Out[0]_P	J12
DaslData[18]Out[1]_N	L05	DaslData[20]Out[1]_N	G05	DaslData[22]Out[1]_N	J11
DaslData[18]Out[1]_P	L06	DaslData[20]Out[1]_P	H05	DaslData[22]Out[1]_P	H11
DaslData[18]Out[2]_N	M09	DaslData[20]Out[2]_N	E08	DaslData[22]Out[2]_N	G12
DaslData[18]Out[2]_P	L07	DaslData[20]Out[2]_P	E07	DaslData[22]Out[2]_P	H13
DaslData[18]Out[3]_N	L08	DaslData[20]Out[3]_N	F07	DaslData[22]Out[3]_N	F11
DaslData[18]Out[3]_P	L09	DaslData[20]Out[3]_P	E06	DaslData[22]Out[3]_P	E11
DaslData[19]In[0]_N	J02	DaslData[21]In[0]_N	B05	DaslData[23]In[0]_N	B11
DaslData[19]In[0]_P	H01	DaslData[21]In[0]_P	A04	DaslData[23]In[0]_P	A11
DaslData[19]In[1]_N	H03	DaslData[21]In[1]_N	D05	DaslData[23]In[1]_N	C11
DaslData[19]In[1]_P	J04	DaslData[21]In[1]_P	C04	DaslData[23]In[1]_P	D11
DaslData[19]In[2]_N	F01	DaslData[21]In[2]_N	B07	DaslData[23]In[2]_N	A12
DaslData[19]In[2]_P	G02	DaslData[21]In[2]_P	A06	DaslData[23]In[2]_P	B13
DaslData[19]In[3]_N	G04	DaslData[21]In[3]_N	C06	DaslData[23]In[3]_N	D13
DaslData[19]In[3]_P	F03	DaslData[21]In[3]_P	D07	DaslData[23]In[3]_P	C12
DaslData[19]Out[0]_N	K09	DaslData[21]Out[0]_N	K11	DaslData[23]Out[0]_N	G13
DaslData[19]Out[0]_P	L10	DaslData[21]Out[0]_P	J10	DaslData[23]Out[0]_P	E12
DaslData[19]Out[1]_N	K05	DaslData[21]Out[1]_N	G08	DaslData[23]Out[1]_N	F13
DaslData[19]Out[1]_P	K07	DaslData[21]Out[1]_P	H09	DaslData[23]Out[1]_P	E13

Table 9-3. I/O Signal List, Sorted by Signal Name (Page 5 of 8)

Signal Name	Pin Location	Signal Name	Pin Location	Signal Name	Pin Location
DaslData[23]Out[2]_N	G14	DaslData[25]Out[2]_N	H19	DaslData[27]Out[2]_N	J22
DaslData[23]Out[2]_P	J14	DaslData[25]Out[2]_P	G19	DaslData[27]Out[2]_P	G23
DaslData[23]Out[3]_N	E14	DaslData[25]Out[3]_N	E18	DaslData[27]Out[3]_N	H23
DaslData[23]Out[3]_P	F15	DaslData[25]Out[3]_P	E19	DaslData[27]Out[3]_P	J23
DaslData[24]In[0]_N	B15	DaslData[26]In[0]_N	A20	DaslData[28]In[0]_N	B25
DaslData[24]In[0]_P	A15	DaslData[26]In[0]_P	A21	DaslData[28]In[0]_P	A26
DaslData[24]In[1]_N	D15	DaslData[26]In[1]_N	C20	DaslData[28]In[1]_N	C26
DaslData[24]In[1]_P	C15	DaslData[26]In[1]_P	C21	DaslData[28]In[1]_P	D25
DaslData[24]In[2]_N	C16	DaslData[26]In[2]_N	B21	DaslData[28]In[2]_N	A28
DaslData[24]In[2]_P	A16	DaslData[26]In[2]_P	A22	DaslData[28]In[2]_P	B27
DaslData[24]In[3]_N	A17	DaslData[26]In[3]_N	C22	DaslData[28]In[3]_N	D27
DaslData[24]In[3]_P	C17	DaslData[26]In[3]_P	D21	DaslData[28]In[3]_P	C28
DaslData[24]Out[0]_N	G15	DaslData[26]Out[0]_N	J20	DaslData[28]Out[0]_N	E24
DaslData[24]Out[0]_P	H15	DaslData[26]Out[0]_P	G20	DaslData[28]Out[0]_P	G24
DaslData[24]Out[1]_N	E15	DaslData[26]Out[1]_N	F19	DaslData[28]Out[1]_N	F25
DaslData[24]Out[1]_P	E16	DaslData[26]Out[1]_P	E20	DaslData[28]Out[1]_P	E25
DaslData[24]Out[2]_N	K15	DaslData[26]Out[2]_N	E22	DaslData[28]Out[2]_N	J24
DaslData[24]Out[2]_P	J15	DaslData[26]Out[2]_P	G21	DaslData[28]Out[2]_P	K23
DaslData[24]Out[3]_N	E17	DaslData[26]Out[3]_N	E21	DaslData[28]Out[3]_N	H25
DaslData[24]Out[3]_P	G17	DaslData[26]Out[3]_P	F21	DaslData[28]Out[3]_P	G26
DaslData[25]In[0]_N	D17	DaslData[27]In[0]_N	A23	DaslData[29]In[0]_N	A30
DaslData[25]In[0]_P	B17	DaslData[27]In[0]_P	B23	DaslData[29]In[0]_P	B29
DaslData[25]In[1]_N	A18	DaslData[27]In[1]_N	D23	DaslData[29]In[1]_N	A32
DaslData[25]In[1]_P	C18	DaslData[27]In[1]_P	C23	DaslData[29]In[1]_P	B31
DaslData[25]In[2]_N	A19	DaslData[27]In[2]_N	A24	DaslData[29]In[2]_N	C30
DaslData[25]In[2]_P	B19	DaslData[27]In[2]_P	A25	DaslData[29]In[2]_P	D29
DaslData[25]In[3]_N	C19	DaslData[27]In[3]_N	C24	DaslData[29]In[3]_N	C32
DaslData[25]In[3]_P	D19	DaslData[27]In[3]_P	C25	DaslData[29]In[3]_P	B33
DaslData[25]Out[0]_N	J19	DaslData[27]Out[0]_N	H21	DaslData[29]Out[0]_N	E27
DaslData[25]Out[0]_P	K19	DaslData[27]Out[0]_P	G22	DaslData[29]Out[0]_P	E26
DaslData[25]Out[1]_N	H17	DaslData[27]Out[1]_N	E23	DaslData[29]Out[1]_N	E28
DaslData[25]Out[1]_P	F17	DaslData[27]Out[1]_P	F23	DaslData[29]Out[1]_P	F27

Table 9-3. I/O Signal List, Sorted by Signal Name (Page 6 of 8)

Signal Name	Pin Location	Signal Name	Pin Location	Signal Name	Pin Location
DaslData[29]Out[2]_N	G28	DaslData[31]Out[2]_N	M27	LSSD_C3_CLK	AJ31
DaslData[29]Out[2]_P	F29	DaslData[31]Out[2]_P	N26	LSSD_C4_CLK	C29
DaslData[29]Out[3]_N	H29	DaslData[31]Out[3]_N	L27	LSSD_SCAN_IN[0]	T09
DaslData[29]Out[3]_P	G29	DaslData[31]Out[3]_P	M25	LSSD_SCAN_IN[1]	AE16
DaslData[30]In[0]_N	E30	DebugDataOut[0]	L23	LSSD_SCAN_IN[2]	AJ33
DaslData[30]In[0]_P	D31	DebugDataOut[1]	N22	LSSD_SCAN_IN[3]	AN29
DaslData[30]In[1]_N	E32	DebugDataOut[2]	G27	LSSD_SCAN_IN[4]	AN27
DaslData[30]In[1]_P	D33	DebugDataOut[3]	N20	LSSD_SCAN_IN[5]	AL05
DaslData[30]In[2]_N	F31	DebugDataOut[4]	N24	LSSD_SCAN_IN[6]	V27
DaslData[30]In[2]_P	G30	DebugDataOut[5]	N25	LSSD_SCAN_IN[7]	AL31
DaslData[30]In[3]_N	G32	DebugDataOut[6]	L33	LSSD_SCAN_IN[8]	V25
DaslData[30]In[3]_P	F33	DebugDataOut[7]	L32	LSSD_SCAN_IN[9]	AG33
DaslData[30]Out[0]_N	H27	DebugDataOut[8]	M29	LSSD_SCAN_IN[10]	AN07
DaslData[30]Out[0]_P	J26	DebugDataOut[9]	N27	LSSD_SCAN_IN[11]	E05
DaslData[30]Out[1]_N	J29	DebugDataOut[10]	L30	LSSD_SCAN_IN[12]	AL33
DaslData[30]Out[1]_P	J28	DebugDataOut[11]	L31	LSSD_SCAN_IN[13]	C01
DaslData[30]Out[2]_N	L24	DebugDataOut[12]	N29	LSSD_SCAN_IN[14]	G33
DaslData[30]Out[2]_P	K25	DebugDataOut[13]	N28	LSSD_SCAN_IN[15]	C05
DaslData[30]Out[3]_N	K27	DebugDataOut[14]	P25	LSSD_SCAN_IN[16]	AE18
DaslData[30]Out[3]_P	K29	DebugDataOut[15]	P27	LSSD_SCAN_IN[17]	AJ29
DaslData[31]In[0]_N	H33	DELAYIn	AA09	LSSD_SCAN_IN[18]	AL27
DaslData[31]In[0]_P	J32	DELAYOut	M17	LSSD_SCAN_IN[19]	AN03
DaslData[31]In[1]_N	J30	DI1#	AJ05	LSSD_SCAN_IN[20]	AL03
DaslData[31]In[1]_P	H31	DI2#	A03	LSSD_SCAN_IN[21]	AL01
DaslData[31]In[2]_N	J33	FullyInsertedIn#	M19	LSSD_SCAN_IN[22]	G03
DaslData[31]In[2]_P	K33	InterruptOut#	L18	LSSD_SCAN_IN[23]	C07
DaslData[31]In[3]_N	J31	IOTEST	V07	LSSD_SCAN_MODE	A27
DaslData[31]In[3]_P	K31	LeakageTest	U31	LSSD_SCAN_OUT[0]	G01
DaslData[31]Out[0]_N	L25	LSSD_A_CLK	E31	LSSD_SCAN_OUT[1]	C33
DaslData[31]Out[0]_P	L26	LSSD_B1_CLK	A29	LSSD_SCAN_OUT[2]	C31
DaslData[31]Out[1]_N	L28	LSSD_C1_CLK	T27	LSSD_SCAN_OUT[3]	A05
DaslData[31]Out[1]_P	L29	LSSD_C2_CLK	G31	LSSD_SCAN_OUT[4]	C03

Table 9-3. I/O Signal List, Sorted by Signal Name (Page 7 of 8)

Signal Name	Pin Location	Signal Name	Pin Location	Signal Name	Pin Location
LSSD_SCAN_OUT[5]	AN31	MSBusInputAddrBidi[7]	V21	MSBusOutputAddrBidi[13]	T23
LSSD_SCAN_OUT[6]	E01	MSBusInputAddrBidi[8]	W30	MSBusOutputAddrBidi[14]	T29
LSSD_SCAN_OUT[7]	E29	MSBusInputAddrBidi[9]	W26	MSBusOutputAddrBidi[15]	R29
LSSD_SCAN_OUT[8]	G18	MSBusInputAddrBidi[10]	W27	MSBusOutputAddrBidi[16]	T19
LSSD_SCAN_OUT[9]	A31	MSBusInputAddrBidi[11]	W20	MSBusOutputAddrBidi[17]	T33
LSSD_SCAN_OUT[10]	G16	MSBusInputAddrBidi[12]	W24	MSBusOutputAddrBidi[18]	U19
LSSD_SCAN_OUT[11]	AG31	MSBusInputAddrBidi[13]	Y29	MSBusOutputAddrBidi[19]	U22
LSSD_SCAN_OUT[12]	E03	MSBusInputAddrBidi[14]	W28	MSBusOutputAddrBidi[20]	U33
LSSD_SCAN_OUT[13]	J16	MSBusInputAddrBidi[15]	AA30	MSBusOutputAddrBidi[21]	U27
LSSD_SCAN_OUT[14]	AG01	MSBusInputAddrBidi[16]	W21	MSBusOutputAddrBidi[22]	U29
LSSD_SCAN_OUT[15]	AG03	MSBusInputAddrBidi[17]	AA31	MSBusOutputAddrBidi[23]	U26
LSSD_SCAN_OUT[16]	C27	MSBusInputAddrBidi[18]	Y31	MSBusOutputAddrBidi[24]	U32
LSSD_SCAN_OUT[17]	A07	MSBusInputAddrBidi[19]	Y25	MSBusOutputAddrBidi[25]	U30
LSSD_SCAN_OUT[18]	AL29	MSBusInputAddrBidi[20]	AA28	MSBusPacketClockBidi	AB33
LSSD_SCAN_OUT[19]	AL07	MSBusInputAddrBidi[21]	AC31	MSBusSyncIn	AC32
LSSD_SCAN_OUT[20]	AN05	MSBusInputAddrBidi[22]	AC30	MSBusSyncOut	AA25
LSSD_SCAN_OUT[21]	E33	MSBusInputAddrBidi[23]	W22	Osc125MhzOut_N	W33
LSSD_SCAN_OUT[22]	V09	MSBusInputAddrBidi[24]	AA23	Osc125MhzOut_P	W32
LSSD_SCAN_OUT[23]	J18	MSBusInputAddrBidi[25]	AA27	Osc250MhzOut_N	N33
LSSD_TAP_C1	T07	MSBusOutputAddrBidi[0]	P31	Osc250MhzOut_P	P33
LSSD_TAP_C2	AG16	MSBusOutputAddrBidi[1]	N31	Osc500MhzOut_N	M33
MemoryGrantOut[0]	A33	MSBusOutputAddrBidi[2]	M31	Osc500MhzOut_P	N32
MemoryGrantOut[1]	P19	MSBusOutputAddrBidi[3]	N30	OscillatorIn	T25
MemoryGrantOut[2]	K21	MSBusOutputAddrBidi[4]	R28	PLL_GNDA	T21
MemoryGrantOut[3]	J21	MSBusOutputAddrBidi[5]	R25	PLL_VDDA	T31
MSBusInputAddrBidi[0]	U23	MSBusOutputAddrBidi[6]	R24	PowerOnResetIn#	N18
MSBusInputAddrBidi[1]	V31	MSBusOutputAddrBidi[7]	R20	ReceiveGrantIn[0]	AA24
MSBusInputAddrBidi[2]	V33	MSBusOutputAddrBidi[8]	R26	ReceiveGrantIn[1]	AA20
MSBusInputAddrBidi[3]	U25	MSBusOutputAddrBidi[9]	R33	ReceiveGrantIn[2]	AB23
MSBusInputAddrBidi[4]	V19	MSBusOutputAddrBidi[10]	R32	ReceiveGrantIn[3]	AC23
MSBusInputAddrBidi[5]	V29	MSBusOutputAddrBidi[11]	R31	ReceiveGrantIn[4]	AG25
MSBusInputAddrBidi[6]	V23	MSBusOutputAddrBidi[12]	R30	ReceiveGrantIn[5]	AC21

Table 9-3. I/O Signal List, Sorted by Signal Name (Page 8 of 8)

Signal Name	Pin Location	Signal Name	Pin Location	Signal Name	Pin Location
ReceiveGrantIn[6]	AD21	ReceiveGrantIn[29]	AG07	SendGrantIn[15]	N14
ReceiveGrantIn[7]	AB19	ReceiveGrantIn[30]	AA14	SendGrantIn[16]	N11
ReceiveGrantIn[8]	AA18	ReceiveGrantIn[31]	AA11	SendGrantIn[17]	N10
ReceiveGrantIn[9]	AC18	RI#	AJ03	SendGrantIn[18]	R12
ReceiveGrantIn[10]	AB17	SCCIn[0]	W23	SendGrantIn[19]	P13
ReceiveGrantIn[11]	AA17	SCCIn[1]	L22	SendGrantIn[20]	T13
ReceiveGrantIn[12]	AC17	SCCIn[2]	AN14	SendGrantIn[21]	T11
ReceiveGrantIn[13]	AE17	SCCIn[3]	Y15	SendGrantIn[22]	T15
ReceiveGrantIn[14]	AC16	SendGrantIn[0]	L17	SendGrantIn[23]	U09
ReceiveGrantIn[15]	W16	SendGrantIn[1]	J17	SendGrantIn[24]	U13
ReceiveGrantIn[16]	AL14	SendGrantIn[2]	L16	SendGrantIn[25]	U10
ReceiveGrantIn[17]	AN13	SendGrantIn[3]	C14	SendGrantIn[26]	V15
ReceiveGrantIn[18]	AB15	SendGrantIn[4]	A13	SendGrantIn[27]	V11
ReceiveGrantIn[19]	AC15	SendGrantIn[5]	A14	SendGrantIn[28]	W14
ReceiveGrantIn[20]	AD13	SendGrantIn[6]	M15	SendGrantIn[29]	W13
ReceiveGrantIn[21]	AC14	SendGrantIn[7]	L15	SendGrantIn[30]	W11
ReceiveGrantIn[22]	AG09	SendGrantIn[8]	J13	SendGrantIn[31]	AA10
ReceiveGrantIn[23]	AB13	SendGrantIn[9]	L13	SHIClockIn	AC33
ReceiveGrantIn[24]	AC12	SendGrantIn[10]	P15	SHISelectIn#	AA32
ReceiveGrantIn[25]	AE09	SendGrantIn[11]	L12	SHISerialDataIn	AA33
ReceiveGrantIn[26]	AC11	SendGrantIn[12]	J09	SHISerialDataOut	Y33
ReceiveGrantIn[27]	AB11	SendGrantIn[13]	N12	TRST	AG18
ReceiveGrantIn[28]	AA12	SendGrantIn[14]	J07		

Table 9-4. I/O Signal List, Sorted by Grid Position (Page 1 of 8)

Pin Location	Signal Name	Pin Location	Signal Name	Pin Location	Signal Name
A02	DaslData[20]In[3]_P	A33	MemoryGrantOut[0]	AB09	DaslData[13]Out[0]_P
A03	DI2#	AA01	DaslData[14]In[2]_N	AB11	ReceiveGrantIn[27]
A04	DaslData[21]In[0]_P	AA02	DaslData[14]In[0]_P	AB13	ReceiveGrantIn[23]
A05	LSSD_SCAN_OUT[3]	AA03	DaslData[14]In[3]_N	AB15	ReceiveGrantIn[18]
A06	DaslData[21]In[2]_P	AA04	DaslData[14]In[1]_N	AB17	ReceiveGrantIn[10]
A07	LSSD_SCAN_OUT[17]	AA05	DaslData[14]Out[1]_P	AB19	ReceiveGrantIn[7]
A08	DaslData[22]In[0]_N	AA06	DaslData[14]Out[1]_N	AB23	ReceiveGrantIn[2]
A09	DaslData[22]In[2]_N	AA07	DaslData[14]Out[0]_N	AB25	DaslData[00]Out[1]_N
A10	DaslData[22]In[2]_P	AA08	DaslData[13]Out[2]_P	AB27	DaslData[00]Out[0]_P
A11	DaslData[23]In[0]_P	AA09	DELAYIn	AB33	MSBusPacketClockBidi
A12	DaslData[23]In[2]_N	AA10	SendGrantIn[31]	AC01	DaslData[13]In[2]_P
A13	SendGrantIn[4]	AA11	ReceiveGrantIn[31]	AC02	DaslData[13]In[2]_N
A14	SendGrantIn[5]	AA12	ReceiveGrantIn[28]	AC03	DaslData[13]In[3]_N
A15	DaslData[24]In[0]_P	AA14	ReceiveGrantIn[30]	AC04	DaslData[13]In[3]_P
A16	DaslData[24]In[2]_P	AA17	ReceiveGrantIn[11]	AC05	DaslData[13]Out[3]_P
A17	DaslData[24]In[3]_N	AA18	ReceiveGrantIn[8]	AC06	DaslData[13]Out[3]_N
A18	DaslData[25]In[1]_N	AA20	ReceiveGrantIn[1]	AC07	DaslData[13]Out[0]_N
A19	DaslData[25]In[2]_N	AA23	MSBusInputAddrBidi[24]	AC08	DaslData[13]Out[1]_P
A20	DaslData[26]In[0]_N	AA24	ReceiveGrantIn[0]	AC09	DaslData[13]Out[1]_N
A21	DaslData[26]In[0]_P	AA25	MSBusSyncOut	AC10	DaslData[12]Out[2]_N
A22	DaslData[26]In[2]_P	AA26	DaslData[00]Out[0]_N	AC11	ReceiveGrantIn[26]
A23	DaslData[27]In[0]_N	AA27	MSBusInputAddrBidi[25]	AC12	ReceiveGrantIn[24]
A24	DaslData[27]In[2]_N	AA28	MSBusInputAddrBidi[20]	AC14	ReceiveGrantIn[21]
A25	DaslData[27]In[2]_P	AA30	MSBusInputAddrBidi[15]	AC15	ReceiveGrantIn[19]
A26	DaslData[28]In[0]_P	AA31	MSBusInputAddrBidi[17]	AC16	ReceiveGrantIn[14]
A27	LSSD_SCAN_MODE	AA32	SHISelectIn#	AC17	ReceiveGrantIn[12]
A28	DaslData[28]In[2]_N	AA33	SHISerialDataIn	AC18	ReceiveGrantIn[9]
A29	LSSD_B1_CLK	AB01	DaslData[14]In[0]_N	AC21	ReceiveGrantIn[5]
A30	DaslData[29]In[0]_N	AB03	DaslData[14]In[1]_P	AC23	ReceiveGrantIn[3]
A31	LSSD_SCAN_OUT[9]	AB05	DaslData[14]Out[0]_P	AC24	DaslData[01]Out[0]_P
A32	DaslData[29]In[1]_N	AB07	DaslData[13]Out[2]_N	AC25	DaslData[00]Out[2]_P

Table 9-4. I/O Signal List, Sorted by Grid Position (Page 2 of 8)

Pin Location	Signal Name	Pin Location	Signal Name	Pin Location	Signal Name
AC26	DaslData[00]Out[2]_N	AE09	ReceiveGrantIn[25]	AF21	DaslData[04]Out[2]_P
AC27	DaslData[00]Out[1]_P	AE10	DaslData[10]Out[2]_N	AF23	DaslData[04]Out[1]_P
AC28	DaslData[00]Out[3]_P	AE11	DaslData[09]Out[3]_P	AF25	DaslData[03]Out[1]_P
AC29	DaslData[00]Out[3]_N	AE12	DaslData[09]Out[2]_N	AF27	DaslData[01]Out[2]_P
AC30	MSBusInputAddrBidi[22]	AE14	DaslData[08]Out[0]_N	AF29	DaslData[02]Out[1]_P
AC31	MSBusInputAddrBidi[21]	AE15	DaslData[07]Out[0]_N	AF31	DaslData[00]In[3]_N
AC32	MSBusSyncIn	AE16	LSSD_SCAN_IN[1]	AF33	DaslData[00]In[2]_P
AC33	SHIClockIn	AE17	ReceiveGrantIn[13]	AG01	LSSD_SCAN_OUT[14]
AD01	DaslData[13]In[0]_P	AE18	LSSD_SCAN_IN[16]	AG02	DaslData[12]In[0]_N
AD03	DaslData[13]In[1]_P	AE19	DaslData[06]Out[2]_P	AG03	LSSD_SCAN_OUT[15]
AD05	DaslData[12]Out[3]_P	AE20	DaslData[05]Out[2]_P	AG04	DaslData[12]In[1]_P
AD07	DaslData[12]Out[3]_N	AE22	DaslData[04]Out[0]_P	AG05	DaslData[11]Out[3]_P
AD09	DaslData[12]Out[2]_P	AE23	DaslData[04]Out[1]_N	AG06	DaslData[11]Out[2]_N
AD11	DaslData[10]Out[2]_P	AE24	DaslData[03]Out[0]_P	AG07	ReceiveGrantIn[29]
AD13	ReceiveGrantIn[20]	AE26	DaslData[01]Out[2]_N	AG08	DaslData[10]Out[3]_P
AD15	DaslData[07]Out[0]_P	AE28	DaslData[01]Out[3]_N	AG09	ReceiveGrantIn[22]
AD19	DaslData[06]Out[2]_N	AE29	DaslData[01]Out[3]_P	AG10	DaslData[10]Out[0]_P
AD21	ReceiveGrantIn[6]	AE30	DaslData[00]In[3]_P	AG11	DaslData[09]Out[2]_P
AD23	DaslData[03]Out[0]_N	AE31	DaslData[00]In[1]_P	AG12	DaslData[09]Out[0]_P
AD25	DaslData[01]Out[0]_N	AE32	DaslData[00]In[2]_N	AG13	DaslData[08]Out[2]_P
AD27	DaslData[01]Out[1]_P	AE33	DaslData[00]In[0]_P	AG14	DaslData[08]Out[0]_P
AD29	DaslData[01]Out[1]_N	AF01	DaslData[12]In[2]_N	AG15	DaslData[07]Out[2]_P
AD31	DaslData[00]In[1]_N	AF03	DaslData[12]In[3]_P	AG16	LSSD_TAP_C2
AD33	DaslData[00]In[0]_N	AF05	DaslData[11]Out[3]_N	AG17	DaslData[07]Out[1]_P
AE01	DaslData[13]In[0]_N	AF07	DaslData[12]Out[0]_N	AG18	TRST
AE02	DaslData[12]In[2]_P	AF09	DaslData[10]Out[3]_N	AG19	DaslData[06]Out[0]_N
AE03	DaslData[13]In[1]_N	AF11	DaslData[09]Out[3]_N	AG20	DaslData[05]Out[2]_N
AE04	DaslData[12]In[3]_N	AF13	DaslData[09]Out[0]_N	AG21	DaslData[05]Out[0]_N
AE05	DaslData[12]Out[1]_N	AF15	DaslData[07]Out[2]_N	AG22	DaslData[04]Out[2]_N
AE06	DaslData[12]Out[1]_P	AF17	DaslData[06]Out[3]_N	AG23	DaslData[04]Out[0]_N
AE08	DaslData[12]Out[0]_P	AF19	DaslData[06]Out[0]_P	AG24	DaslData[03]Out[2]_N

Table 9-4. I/O Signal List, Sorted by Grid Position (Page 3 of 8)

Pin Location	Signal Name	Pin Location	Signal Name	Pin Location	Signal Name
AG25	ReceiveGrantIn[4]	AJ08	DaslData[11]Out[0]_P	AK13	DaslData[08]In[1]_P
AG26	DaslData[03]Out[1]_N	AJ09	DaslData[10]Out[1]_P	AK15	DaslData[07]In[3]_P
AG28	DaslData[02]Out[0]_P	AJ10	DaslData[10]Out[0]_N	AK17	DaslData[07]In[0]_P
AG29	DaslData[02]Out[1]_N	AJ11	DaslData[09]Out[1]_N	AK19	DaslData[06]In[1]_N
AG30	DaslData[01]In[0]_N	AJ12	DaslData[08]Out[2]_N	AK21	DaslData[05]In[1]_N
AG31	LSSD_SCAN_OUT[11]	AJ13	DaslData[08]Out[3]_N	AK23	DaslData[04]In[3]_P
AG32	DaslData[01]In[1]_P	AJ14	DaslData[08]Out[1]_P	AK25	DaslData[03]In[3]_N
AG33	LSSD_SCAN_IN[9]	AJ15	DaslData[07]Out[3]_P	AK27	DaslData[03]In[1]_P
AH01	DaslData[12]In[0]_P	AJ16	DaslData[07]Out[3]_N	AK29	DaslData[02]In[0]_N
AH03	DaslData[12]In[1]_N	AJ17	DaslData[06]Out[3]_P	AK31	DaslData[01]In[2]_N
AH05	DaslData[11]Out[2]_P	AJ18	DaslData[06]Out[1]_P	AK33	DaslData[01]In[3]_N
AH07	DaslData[11]Out[1]_P	AJ19	DaslData[06]Out[1]_N	AL01	LSSD_SCAN_IN[21]
AH09	DaslData[10]Out[1]_N	AJ20	DaslData[05]Out[3]_N	AL02	DaslData[11]In[3]_N
AH11	DaslData[09]Out[1]_P	AJ21	DaslData[05]Out[1]_P	AL03	LSSD_SCAN_IN[20]
AH13	DaslData[08]Out[3]_P	AJ22	DaslData[05]Out[0]_P	AL04	DaslData[10]In[3]_N
AH15	DaslData[08]Out[1]_N	AJ23	DaslData[04]Out[3]_P	AL05	LSSD_SCAN_IN[5]
AH17	DaslData[07]Out[1]_N	AJ24	DaslData[03]Out[2]_P	AL06	DaslData[10]In[1]_P
AH19	DaslData[05]Out[3]_P	AJ25	DaslData[03]Out[3]_N	AL07	LSSD_SCAN_OUT[19]
AH21	DaslData[05]Out[1]_N	AJ26	DaslData[02]Out[2]_N	AL08	DaslData[09]In[3]_N
AH23	DaslData[04]Out[3]_N	AJ27	DaslData[02]Out[2]_P	AL09	DaslData[09]In[1]_P
AH25	DaslData[03]Out[3]_P	AJ28	DaslData[02]Out[3]_P	AL10	DaslData[09]In[1]_N
AH27	DaslData[02]Out[3]_N	AJ29	LSSD_SCAN_IN[17]	AL11	DaslData[08]In[3]_P
AH29	DaslData[02]Out[0]_N	AJ30	DaslData[01]In[2]_P	AL12	DaslData[08]In[1]_N
AH31	DaslData[01]In[0]_P	AJ31	LSSD_C3_CLK	AL14	ReceiveGrantIn[16]
AH33	DaslData[01]In[1]_N	AJ32	DaslData[01]In[3]_P	AL15	DaslData[07]In[3]_N
AJ01	BISTTESTM3	AJ33	LSSD_SCAN_IN[2]	AL16	DaslData[07]In[1]_P
AJ02	DaslData[11]In[2]_N	AK01	DaslData[11]In[2]_P	AL17	DaslData[06]In[3]_N
AJ03	RI#	AK03	DaslData[11]In[0]_P	AL18	DaslData[06]In[2]_N
AJ04	DaslData[11]In[0]_N	AK05	DaslData[10]In[3]_P	AL19	DaslData[06]In[1]_P
AJ05	DI1#	AK07	DaslData[10]In[1]_N	AL20	DaslData[05]In[3]_P
AJ06	DaslData[11]Out[1]_N	AK09	DaslData[09]In[3]_P	AL21	DaslData[05]In[3]_N
AJ07	DaslData[11]Out[0]_N	AK11	DaslData[08]In[3]_N	AL22	DaslData[05]In[1]_P

Table 9-4. I/O Signal List, Sorted by Grid Position (Page 4 of 8)

Pin Location	Signal Name	Pin Location	Signal Name	Pin Location	Signal Name
AL23	DaslData[04]In[3]_N	AN06	DaslData[10]In[0]_N	B11	DaslData[23]In[0]_N
AL24	DaslData[04]In[1]_P	AN07	LSSD_SCAN_IN[10]	B13	DaslData[23]In[2]_P
AL25	DaslData[04]In[1]_N	AN08	DaslData[09]In[2]_P	B15	DaslData[24]In[0]_N
AL26	DaslData[03]In[3]_P	AN09	DaslData[09]In[0]_P	B17	DaslData[25]In[0]_P
AL27	LSSD_SCAN_IN[18]	AN10	DaslData[09]In[0]_N	B19	DaslData[25]In[2]_P
AL28	DaslData[03]In[1]_N	AN11	DaslData[08]In[2]_N	B21	DaslData[26]In[2]_N
AL29	LSSD_SCAN_OUT[18]	AN12	DaslData[08]In[0]_P	B23	DaslData[27]In[0]_P
AL30	DaslData[02]In[0]_P	AN13	ReceiveGrantIn[17]	B25	DaslData[28]In[0]_N
AL31	LSSD_SCAN_IN[7]	AN14	SCCIn[2]	B27	DaslData[28]In[2]_P
AL32	DaslData[02]In[1]_P	AN15	DaslData[07]In[2]_N	B29	DaslData[29]In[0]_P
AL33	LSSD_SCAN_IN[12]	AN16	DaslData[07]In[1]_N	B31	DaslData[29]In[1]_P
AM01	DaslData[11]In[3]_P	AN17	DaslData[06]In[3]_P	B33	DaslData[29]In[3]_P
AM03	DaslData[11]In[1]_P	AN18	DaslData[06]In[2]_P	C01	LSSD_SCAN_IN[13]
AM05	DaslData[10]In[2]_P	AN19	DaslData[06]In[0]_P	C02	DaslData[20]In[1]_P
AM07	DaslData[10]In[0]_P	AN20	DaslData[05]In[2]_P	C03	LSSD_SCAN_OUT[4]
AM09	DaslData[09]In[2]_N	AN21	DaslData[05]In[2]_N	C04	DaslData[21]In[1]_P
AM11	DaslData[08]In[2]_P	AN22	DaslData[05]In[0]_N	C05	LSSD_SCAN_IN[15]
AM13	DaslData[08]In[0]_N	AN23	DaslData[04]In[2]_P	C06	DaslData[21]In[3]_N
AM15	DaslData[07]In[2]_P	AN24	DaslData[04]In[0]_P	C07	LSSD_SCAN_IN[23]
AM17	DaslData[07]In[0]_N	AN25	DaslData[04]In[0]_N	C08	DaslData[22]In[1]_P
AM19	DaslData[06]In[0]_N	AN26	DaslData[03]In[2]_N	C09	DaslData[22]In[3]_N
AM21	DaslData[05]In[0]_P	AN27	LSSD_SCAN_IN[4]	C10	DaslData[22]In[3]_P
AM23	DaslData[04]In[2]_N	AN28	DaslData[03]In[0]_P	C11	DaslData[23]In[1]_N
AM25	DaslData[03]In[2]_P	AN29	LSSD_SCAN_IN[3]	C12	DaslData[23]In[3]_P
AM27	DaslData[03]In[0]_N	AN30	DaslData[02]In[2]_P	C14	SendGrantIn[3]
AM29	DaslData[02]In[2]_N	AN31	LSSD_SCAN_OUT[5]	C15	DaslData[24]In[1]_P
AM31	DaslData[02]In[3]_N	AN32	DaslData[02]In[3]_P	C16	DaslData[24]In[2]_N
AM33	DaslData[02]In[1]_N	B01	DaslData[20]In[1]_N	C17	DaslData[24]In[3]_P
AN02	DaslData[11]In[1]_N	B03	DaslData[20]In[3]_N	C18	DaslData[25]In[1]_P
AN03	LSSD_SCAN_IN[19]	B05	DaslData[21]In[0]_N	C19	DaslData[25]In[3]_N
AN04	DaslData[10]In[2]_N	B07	DaslData[21]In[2]_N	C20	DaslData[26]In[1]_N
AN05	LSSD_SCAN_OUT[20]	B09	DaslData[22]In[0]_P	C21	DaslData[26]In[1]_P

Table 9-4. I/O Signal List, Sorted by Grid Position (Page 5 of 8)

Pin Location	Signal Name	Pin Location	Signal Name	Pin Location	Signal Name
C22	DaslData[26]In[3]_N	E04	DaslData[20]In[2]_P	F05	DaslData[20]Out[0]_N
C23	DaslData[27]In[1]_P	E05	LSSD_SCAN_IN[11]	F07	DaslData[20]Out[3]_N
C24	DaslData[27]In[3]_N	E06	DaslData[20]Out[3]_P	F09	DaslData[21]Out[3]_P
C25	DaslData[27]In[3]_P	E07	DaslData[20]Out[2]_P	F11	DaslData[22]Out[3]_N
C26	DaslData[28]In[1]_N	E08	DaslData[20]Out[2]_N	F13	DaslData[23]Out[1]_N
C27	LSSD_SCAN_OUT[16]	E09	DaslData[21]Out[3]_N	F15	DaslData[23]Out[3]_P
C28	DaslData[28]In[3]_P	E10	DaslData[21]Out[2]_P	F17	DaslData[25]Out[1]_P
C29	LSSD_C4_CLK	E11	DaslData[22]Out[3]_P	F19	DaslData[26]Out[1]_N
C30	DaslData[29]In[2]_N	E12	DaslData[23]Out[0]_P	F21	DaslData[26]Out[3]_P
C31	LSSD_SCAN_OUT[2]	E13	DaslData[23]Out[1]_P	F23	DaslData[27]Out[1]_P
C32	DaslData[29]In[3]_N	E14	DaslData[23]Out[3]_N	F25	DaslData[28]Out[1]_N
C33	LSSD_SCAN_OUT[1]	E15	DaslData[24]Out[1]_N	F27	DaslData[29]Out[1]_P
D01	DaslData[20]In[0]_N	E16	DaslData[24]Out[1]_P	F29	DaslData[29]Out[2]_P
D03	DaslData[20]In[2]_N	E17	DaslData[24]Out[3]_N	F31	DaslData[30]In[2]_N
D05	DaslData[21]In[1]_N	E18	DaslData[25]Out[3]_N	F33	DaslData[30]In[3]_P
D07	DaslData[21]In[3]_P	E19	DaslData[25]Out[3]_P	G01	LSSD_SCAN_OUT[0]
D09	DaslData[22]In[1]_N	E20	DaslData[26]Out[1]_P	G02	DaslData[19]In[2]_P
D11	DaslData[23]In[1]_P	E21	DaslData[26]Out[3]_N	G03	LSSD_SCAN_IN[22]
D13	DaslData[23]In[3]_N	E22	DaslData[26]Out[2]_N	G04	DaslData[19]In[3]_N
D15	DaslData[24]In[1]_N	E23	DaslData[27]Out[1]_N	G05	DaslData[20]Out[1]_N
D17	DaslData[25]In[0]_N	E24	DaslData[28]Out[0]_N	G06	DaslData[20]Out[0]_P
D19	DaslData[25]In[3]_P	E25	DaslData[28]Out[1]_P	G08	DaslData[21]Out[1]_N
D21	DaslData[26]In[3]_P	E26	DaslData[29]Out[0]_P	G10	DaslData[21]Out[2]_N
D23	DaslData[27]In[1]_N	E27	DaslData[29]Out[0]_N	G11	DaslData[22]Out[0]_N
D25	DaslData[28]In[1]_P	E28	DaslData[29]Out[1]_N	G12	DaslData[22]Out[2]_N
D27	DaslData[28]In[3]_N	E29	LSSD_SCAN_OUT[7]	G13	DaslData[23]Out[0]_N
D29	DaslData[29]In[2]_P	E30	DaslData[30]In[0]_N	G14	DaslData[23]Out[2]_N
D31	DaslData[30]In[0]_P	E31	LSSD_A_CLK	G15	DaslData[24]Out[0]_N
D33	DaslData[30]In[1]_P	E32	DaslData[30]In[1]_N	G16	LSSD_SCAN_OUT[10]
E01	LSSD_SCAN_OUT[6]	E33	LSSD_SCAN_OUT[21]	G17	DaslData[24]Out[3]_P
E02	DaslData[20]In[0]_P	F01	DaslData[19]In[2]_N	G18	LSSD_SCAN_OUT[8]
E03	LSSD_SCAN_OUT[12]	F03	DaslData[19]In[3]_P	G19	DaslData[25]Out[2]_P

Table 9-4. I/O Signal List, Sorted by Grid Position (Page 6 of 8)

Pin Location	Signal Name	Pin Location	Signal Name	Pin Location	Signal Name
G20	DaslData[26]Out[0]_P	J03	DaslData[18]In[3]_P	K07	DaslData[19]Out[1]_P
G21	DaslData[26]Out[2]_P	J04	DaslData[19]In[1]_P	K09	DaslData[19]Out[0]_N
G22	DaslData[27]Out[0]_P	J05	DaslData[19]Out[3]_P	K11	DaslData[21]Out[0]_N
G23	DaslData[27]Out[2]_P	J06	DaslData[19]Out[3]_N	K15	DaslData[24]Out[2]_N
G24	DaslData[28]Out[0]_P	J07	SendGrantIn[14]	K19	DaslData[25]Out[0]_P
G26	DaslData[28]Out[3]_P	J08	DaslData[19]Out[2]_N	K21	MemoryGrantOut[2]
G27	DebugDataOut[2]	J09	SendGrantIn[12]	K23	DaslData[28]Out[2]_P
G28	DaslData[29]Out[2]_N	J10	DaslData[21]Out[0]_P	K25	DaslData[30]Out[2]_P
G29	DaslData[29]Out[3]_P	J11	DaslData[22]Out[1]_N	K27	DaslData[30]Out[3]_N
G30	DaslData[30]In[2]_P	J12	DaslData[22]Out[0]_P	K29	DaslData[30]Out[3]_P
G31	LSSD_C2_CLK	J13	SendGrantIn[8]	K31	DaslData[31]In[3]_P
G32	DaslData[30]In[3]_N	J14	DaslData[23]Out[2]_P	K33	DaslData[31]In[2]_P
G33	LSSD_SCAN_IN[14]	J15	DaslData[24]Out[2]_P	L01	DaslData[18]In[0]_N
H01	DaslData[19]In[0]_P	J16	LSSD_SCAN_OUT[13]	L02	DaslData[18]In[0]_P
H03	DaslData[19]In[1]_N	J17	SendGrantIn[1]	L03	DaslData[18]In[1]_P
H05	DaslData[20]Out[1]_P	J18	LSSD_SCAN_OUT[23]	L04	DaslData[18]In[1]_N
H07	DaslData[19]Out[2]_P	J19	DaslData[25]Out[0]_N	L05	DaslData[18]Out[1]_N
H09	DaslData[21]Out[1]_P	J20	DaslData[26]Out[0]_N	L06	DaslData[18]Out[1]_P
H11	DaslData[22]Out[1]_P	J21	MemoryGrantOut[3]	L07	DaslData[18]Out[2]_P
H13	DaslData[22]Out[2]_P	J22	DaslData[27]Out[2]_N	L08	DaslData[18]Out[3]_N
H15	DaslData[24]Out[0]_P	J23	DaslData[27]Out[3]_P	L09	DaslData[18]Out[3]_P
H17	DaslData[25]Out[1]_N	J24	DaslData[28]Out[2]_N	L10	DaslData[19]Out[0]_P
H19	DaslData[25]Out[2]_N	J26	DaslData[30]Out[0]_P	L12	SendGrantIn[11]
H21	DaslData[27]Out[0]_N	J28	DaslData[30]Out[1]_P	L13	SendGrantIn[9]
H23	DaslData[27]Out[3]_N	J29	DaslData[30]Out[1]_N	L15	SendGrantIn[7]
H25	DaslData[28]Out[3]_N	J30	DaslData[31]In[1]_N	L16	SendGrantIn[2]
H27	DaslData[30]Out[0]_N	J31	DaslData[31]In[3]_N	L17	SendGrantIn[0]
H29	DaslData[29]Out[3]_N	J32	DaslData[31]In[0]_P	L18	InterruptOut#
H31	DaslData[31]In[1]_P	J33	DaslData[31]In[2]_N	L22	SCCIn[1]
H33	DaslData[31]In[0]_N	K01	DaslData[18]In[2]_N	L23	DebugDataOut [0]
J01	DaslData[18]In[2]_P	K03	DaslData[18]In[3]_N	L24	DaslData[30]Out[2]_N
J02	DaslData[19]In[0]_N	K05	DaslData[19]Out[1]_N	L25	DaslData[31]Out[0]_N

Table 9-4. I/O Signal List, Sorted by Grid Position (Page 7 of 8)

Pin Location	Signal Name	Pin Location	Signal Name	Pin Location	Signal Name
L26	DaslData[31]Out[0]_P	N14	SendGrantIn[15]	R07	DaslData[16]Out[2]_P
L27	DaslData[31]Out[3]_N	N18	PowerOnResetIn#	R08	DaslData[16]Out[2]_N
L28	DaslData[31]Out[1]_N	N20	DebugDataOut[3]	R09	DaslData[16]Out[0]_N
L29	DaslData[31]Out[1]_P	N22	DebugDataOut[1]	R10	DaslData[16]Out[0]_P
L30	DebugDataOut[10]	N24	DebugDataOut[4]	R12	SendGrantIn[18]
L31	DebugDataOut[11]	N25	DebugDataOut[5]	R20	MSBusOutputAddrBidi[7]
L32	DebugDataOut[7]	N26	DaslData[31]Out[2]_P	R24	MSBusOutputAddrBidi[6]
L33	DebugDataOut[6]	N27	DebugDataOut[9]	R25	MSBusOutputAddrBidi[5]
M01	DaslData[17]In[2]_P	N28	DebugDataOut[13]	R26	MSBusOutputAddrBidi[8]
M03	DaslData[17]In[3]_N	N29	DebugDataOut[12]	R28	MSBusOutputAddrBidi[4]
M05	DaslData[17]Out[2]_N	N30	MSBusOutputAddrBidi[3]	R29	MSBusOutputAddrBidi[15]
M07	DaslData[18]Out[0]_P	N31	MSBusOutputAddrBidi[1]	R30	MSBusOutputAddrBidi[12]
M09	DaslData[18]Out[2]_N	N32	Osc500MhzOut_P	R31	MSBusOutputAddrBidi[11]
M15	SendGrantIn[6]	N33	Osc250MhzOut_N	R32	MSBusOutputAddrBidi[10]
M17	DELAYOut	P01	DaslData[17]In[0]_N	R33	MSBusOutputAddrBidi[9]
M19	FullyInsertedIn#	P03	DaslData[17]In[1]_N	T01	DaslData[16]In[1]_N
M25	DaslData[31]Out[3]_P	P05	DaslData[17]Out[1]_P	T03	DaslData[16]In[1]_P
M27	DaslData[31]Out[2]_N	P07	DaslData[17]Out[0]_P	T05	DaslData[16]Out[3]_N
M29	DebugDataOut[8]	P09	DaslData[17]Out[0]_N	T07	LSSD_TAP_C1
M31	MSBusOutputAddrBidi[2]	P13	SendGrantIn[19]	T09	LSSD_SCAN_IN[0]
M33	Osc500MhzOut_N	P15	SendGrantIn[10]	T11	SendGrantIn[21]
N01	DaslData[17]In[0]_P	P19	MemoryGrantOut[1]	T13	SendGrantIn[20]
N02	DaslData[17]In[2]_N	P25	DebugDataOut[14]	T15	SendGrantIn[22]
N03	DaslData[17]In[1]_P	P27	DebugDataOut[15]	T19	MSBusOutputAddrBidi[16]
N04	DaslData[17]In[3]_P	P31	MSBusOutputAddrBidi[0]	T21	PLL_GNDA
N05	DaslData[17]Out[3]_N	P33	Osc250MhzOut_P	T23	MSBusOutputAddrBidi[13]
N06	DaslData[17]Out[3]_P	R01	DaslData[16]In[2]_N	T25	OscillatorIn
N07	DaslData[17]Out[2]_P	R02	DaslData[16]In[2]_P	T27	LSSD_C1_CLK
N08	DaslData[18]Out[0]_N	R03	DaslData[16]In[3]_N	T29	MSBusOutputAddrBidi[14]
N10	SendGrantIn[17]	R04	DaslData[16]In[3]_P	T31	PLL_VDDA
N11	SendGrantIn[16]	R05	DaslData[16]Out[3]_P	T33	MSBusOutputAddrBidi[17]
N12	SendGrantIn[13]	R06	DaslData[17]Out[1]_N	U01	DaslData[15]In[3]_P

Table 9-4. I/O Signal List, Sorted by Grid Position (Page 8 of 8)

Pin Location	Signal Name	Pin Location	Signal Name	Pin Location	Signal Name
U02	DaslData[16]In[0]_N	V07	IOTEST	W14	SendGrantIn[28]
U03	DaslData[15]In[3]_N	V09	LSSD_SCAN_OUT[22]	W16	ReceiveGrantIn[15]
U04	DaslData[16]In[0]_P	V11	SendGrantIn[27]	W20	MSBusInputAddrBidi[11]
U05	DaslData[16]Out[1]_P	V15	SendGrantIn[26]	W21	MSBusInputAddrBidi[16]
U06	DaslData[15]Out[3]_N	V19	MSBusInputAddrBidi[4]	W22	MSBusInputAddrBidi[23]
U07	DaslData[16]Out[1]_N	V21	MSBusInputAddrBidi[7]	W23	SCCIn[0]
U08	DaslData[15]Out[3]_P	V23	MSBusInputAddrBidi[6]	W24	MSBusInputAddrBidi[12]
U09	SendGrantIn[23]	V25	LSSD_SCAN_IN[8]	W26	MSBusInputAddrBidi[9]
U10	SendGrantIn[25]	V27	LSSD_SCAN_IN[6]	W27	MSBusInputAddrBidi[10]
U13	SendGrantIn[24]	V29	MSBusInputAddrBidi[5]	W28	MSBusInputAddrBidi[14]
U19	MSBusOutputAddrBidi[18]	V31	MSBusInputAddrBidi[1]	W30	MSBusInputAddrBidi[8]
U22	MSBusOutputAddrBidi[19]	V33	MSBusInputAddrBidi[2]	W32	Osc125MhzOut_P
U23	MSBusInputAddrBidi[0]	W01	DaslData[15]In[0]_P	W33	Osc125MhzOut_N
U25	MSBusInputAddrBidi[3]	W02	DaslData[15]In[0]_N	Y01	DaslData[14]In[2]_P
U26	MSBusOutputAddrBidi[23]	W03	DaslData[15]In[1]_P	Y03	DaslData[14]In[3]_P
U27	MSBusOutputAddrBidi[21]	W04	DaslData[15]In[1]_N	Y05	DaslData[14]Out[3]_N
U29	MSBusOutputAddrBidi[22]	W05	DaslData[15]Out[1]_N	Y07	DaslData[14]Out[2]_N
U30	MSBusOutputAddrBidi[25]	W06	DaslData[14]Out[3]_P	Y09	DaslData[14]Out[2]_P
U31	LeakageTest	W07	DaslData[15]Out[0]_N	Y15	SCCIn[3]
U32	MSBusOutputAddrBidi[24]	W08	DaslData[15]Out[0]_P	Y25	MSBusInputAddrBidi[19]
U33	MSBusOutputAddrBidi[20]	W09	DaslData[15]Out[2]_P	Y29	MSBusInputAddrBidi[13]
V01	DaslData[15]In[2]_P	W10	DaslData[15]Out[2]_N	Y31	MSBusInputAddrBidi[18]
V03	DaslData[15]In[2]_N	W11	SendGrantIn[30]	Y33	SHISerialDataOut
V05	DaslData[15]Out[1]_P	W13	SendGrantIn[29]		

10. Electrical Characteristics

Table 10-1. Absolute Maximum Ratings

Symbol	Parameter	Rating			Units	Notes
		Minimum	Typical	Maximum		
V_{DD}	Supply voltage		1.8	1.98	V	1
V_{IN}	Input voltage	-0.6		V_{DD} + 0.45	V	1, 2
V_{OUT}	Output voltage	-0.6		V_{DD}	V	1, 2
Θ_{JA}	Thermal impedance junction to ambient package, airflow = 0 feet per minute		10.8		°C/W	3
Θ_{JA}	Thermal impedance junction to ambient package, airflow = 200 feet per minute		8.4		°C/W	3
Θ_{JA}	Thermal impedance junction to ambient package, airflow = 400 feet per minute		6.6		°C/W	3
Θ_{JC}	Thermal impedance junction to case package		0.20		°C/W	4
T_S	Storage temperature	-65		150	°C	1
T_A	Operating junction temperature range	0		125	°C	1
	Electrostatic discharge	-3000	6000	3000	V	1

1. Permanent device damage may occur if the above absolute maximum ratings are exceeded. Extended exposure to absolute maximum rating conditions may affect device reliability. Normal operation should be restricted to the conditions listed in *Table 10-2.*
2. For $V_{DD} \leq 1.95$ V.
3. For devices mounted to a 2S2P card (1-ounce copper, size 63.5 mm × 76.2 mm), with flow on both sides of the card in a vertical orientation.
4. Θ_{JC} represents the temperature difference between the junction and the top center of the outside surface of the component package, divided by the power applied to the component mounted to the test card.

Table 10-2. Recommended Operating Conditions

Symbol	Parameter (for TTL-compatible I/Os)		Rating			Units	Notes
			Minimum	Typical	Maximum		
V_{DD}	Supply voltage		1.71	1.8	1.89	V	1, 2
V_{DD2}	Supply voltage		2.375	2.5	2.25	V	1, 2
V_{IH}	Input up level (HSTL)		V_{REF} + 0.1		V_{DD} + 0.3	V	3
V_{IL}	Input down level (HSTL)		-0.3		V_{REF} - 0.1	V	3
V_{OH}	High-level output voltage (I_{OH} = -8 mA)		V_{DD} - 0.4			V	
V_{OL}	Low-level output voltage (I_{OL} = 8 mA)				0.4	V	
I_{IL}	Receiver maximum input leakage, low-level input current at least-positive down level	No pullup or pulldown			0	µA	4
		With pullup			0	µA	
		With pulldown			-150	µA	
I_{IH}	Receiver maximum input leakage, high-level input current at most-positive up level	No pullup or pulldown			0	µA	4
		With pullup			200	µA	
		With pulldown			0	µA	
C_I	Input capacitance (V_{DD} = nominal)				5	pF	

1. For power-up, the +2.5 V supply is activated either prior to or concurrently with the +1.8 V supply.
2. For power-down, the +2.5 V supply is deactivated either after or concurrently with the +1.8 V supply.
3. $V_{REF} = V_{DD} \div 2$
4. Applies to bidirectional receivers without pullup or pulldown resistors.

Table 10-3. Power Dissipation

Supply	Power		Current	
	Nominal	Maximum	Nominal	Maximum
2.5 V	0.464 W	0.512 W	0.185 A	0.205 A
1.8 V	21.6 W	23.9 W	12 A	13.3 A

Table 10-4. Electrical Characteristics for DASL I/Os

Parameter	Reference Signal	Rating			Units
		Minimum	Typical	Maximum	
Rising transition rate of the output	DASL driver	1.8	2.09	2.28	V/ns
Falling transition rate of the output	DASL driver	1.54	1.92	2.08	V/ns
Maximum input pin capacitance	DASL receiver		2.5		pF

Note: The DASL I/Os are compatible with the high-speed transceiver logic (HSTL) differential interface defined in Electronic Industries Association/JEDEC Standard No. 8-6.

Table 10-5. Clocks

Parameter		Rating		Units	Notes
		Minimum	Maximum		
Internal byte clock frequency		106.25	125	MHz	
OscillatorIn frequency	Internal PLL	53.125	62.5	MHz	1, 2

1. The OscillatorIn input must have a tolerance of ±100 ppm (0.01%), a duty cycle of 40 to 60 percent, and a phase jitter of ±150 ps (cycle to cycle) maximum.
2. The skew between the master and slave OscillatorIn input pins is ±250 ps maximum.

11. DASL and Reference Clock Line Termination

The data-aligned synchronous link (DASL) data inputs (DaslData[*p*]In[*n*]_P and DaslData[*p*]In[*n*]_N) and the reference clock (Osc125MhzOut_P and Osc125MhzOut_N, Osc250MhzOut_P and Osc250MhzOut_N, *or* Osc500MhzOut_P and Osc500MhzOut_N) use differential I/O pairs compliant with the high-speed transceiver logic (HSTL) interface defined in Electronic Industries Association/JEDEC Standard No. 8-6. For the reference clock differential pair, both termination resistors are connected directly to ground, without the use of a capacitor. The recommended termination for the DASL data input receiver lines is shown in *Figure 11-1.* The termination network must be placed within 2.5 cm of the receiver device (for DASL and reference clock lines).

Figure 11-1. DASL Termination

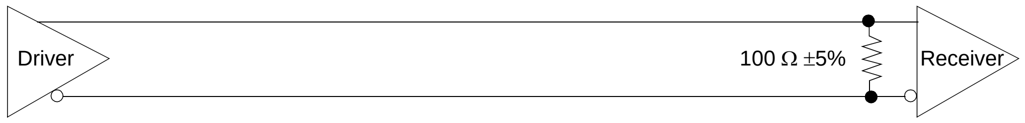
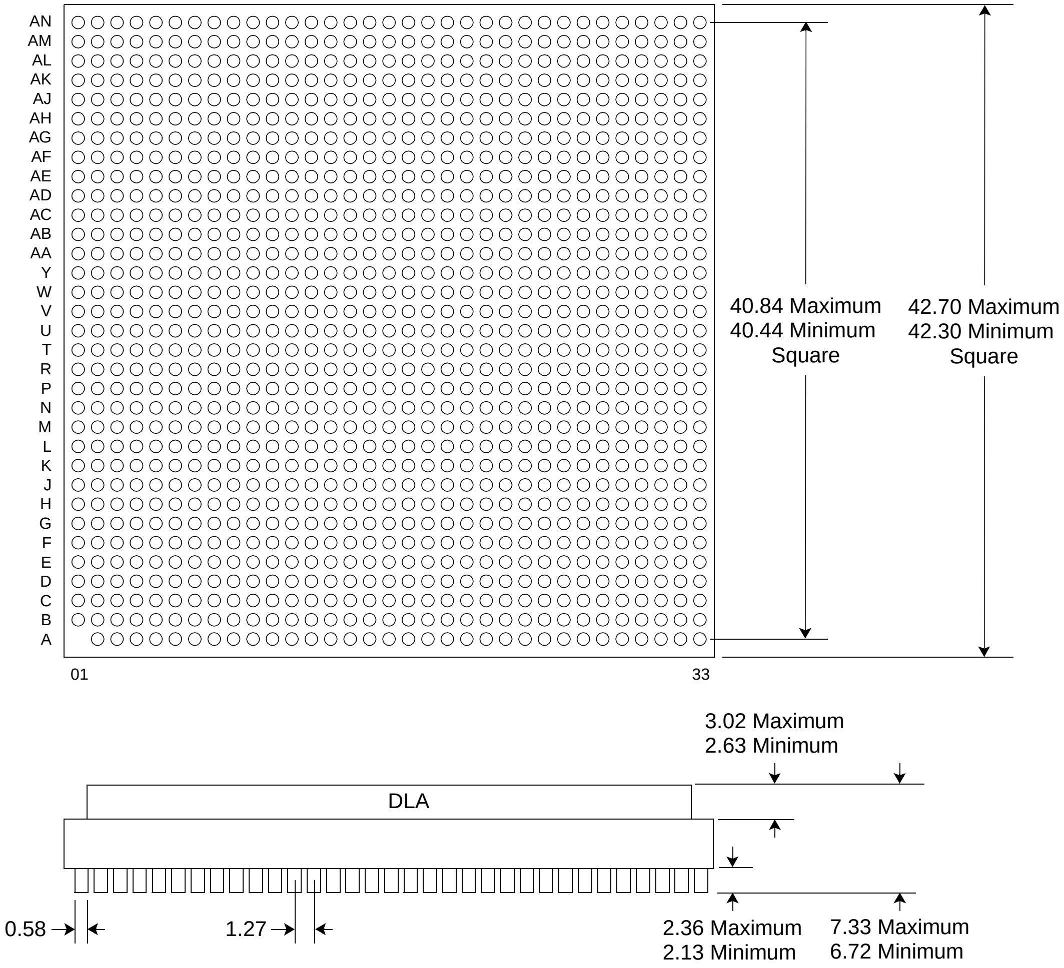


Figure 11-2. DASL Termination for Master LUs, Bits 2 and 3

12. Mechanical Information

Figure 12-1. Package Mechanical



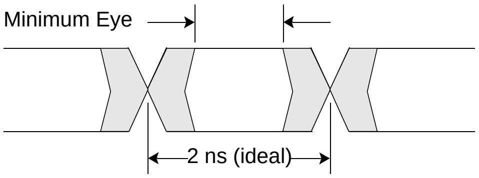
Package type is 1088-ball ceramic column grid array (CCGA) with direct lid attach (DLA), package size is 42.5 × 42.5 mm, and array size is 33 × 33. All dimensions are in millimeters.

13. Glossary

absolute maximum rating	The highest value a quantity can have before malfunction or damage occurs.
address	A number designating a particular memory location.
array	An ordered arrangement of data elements.
ATM	asynchronous transfer mode
bandwidth	The minimum capacity required for effective transmission and reception of a data packet.
best-effort delivery	The delivery of packets with no bandwidth guarantee and an unspecified quality of service. Packets may be discarded during periods of traffic congestion.
bidirectional	The ability to transmit in both directions.
BIST	built-in self-test
bitmap	A binary representation in which a bit or set of bits corresponds to an assigned value or condition.
buffer	A memory bank used for temporary storage.
bus	A common pathway over which input and output signals are routed.
CCGA	ceramic column grid array
checksum	A calculated value used to ensure that data is stored or transmitted without error. Checksums detect single-bit errors and some multiple-bit errors, but are not as effective as the CRC method for detecting errors.
clock	An internal timing device that synchronizes the data pulses between the transmitter and receiver.
clock cycle	One “tick” of the clock. For example, a 100-MHz clock has 100 million ticks per second.
clock frequency	The reciprocal of the time period of a single clock cycle.
CMOS	complementary metal-oxide semiconductor
configuration	The arrangement of speed expansion, port expansion, and port-paralleling options that create a custom switching device.
control packet	The packet that carries the communications between the local processor and the protocol engine. Control packets do not have a specific priority.
counter	A circuit that counts pulses and generates an output at a specified time.
CPU	central processing unit
CRC	cyclic redundancy check

credit table	A weighted cycling mechanism that can be programmed to guarantee minimum bandwidth for low-priority packets. The credit table includes 256 entries, or credits, per port.
DASL	data-aligned synchronous link
data packet	See “packet.”
differential pair	Two wires of opposite polarity configured as a pair to reduce signal noise and crosstalk.
DLA	direct lid attach
driver	Also called a “device driver.” A routine that links a peripheral device to the operating system and performs internal functions, or a functional unit that increases the output current, power, or voltage of another functional unit.
egress flow	Data flow from the PowerPRS 64G to the attached device.
FIFO	first-in-first-out
filter	A pattern or mask through which only selected data is passed (for example, bitmap filter, best-effort discard filter, and so forth).
Gbps	gigabits per second
GRA	global register array
HSTL	high-speed transceiver logic
hysteresis	The lag between making a change and the response or effect of the change.
I/O	input/output
idle packet	The packet that is transmitted when the other packet types are either unavailable for transmission or prevented from transmission due to a flow control situation. Idle packets carry the output queue grants used for ingress flow control, but they do not carry user data.
IEEE	Institute of Electrical and Electronics Engineers
ingress flow	Data flow from the attached device to the PowerPRS 64G.
interrupt	A signal that gains the attention of the CPU and is usually generated when input or output is required.
jitter	A flicker or fluctuation in a transmission signal caused by a bit arriving either ahead or behind the standard clock cycle or, more generally, the variable arrival of packets.
JTAG	Joint Test Action Group. IEEE Standard 1149.1 (regarding boundary-scan architecture) is also referred to as the JTAG standard, after the group that developed it.
junction	The region of contact between opposite types of semiconductor materials.

latency	The lag between initiating a data request and starting the actual data transfer.
line	An electronic communications channel such as a wire.
load balancing	The fine tuning of the network to more evenly distribute the data and/or processing across all available resources.
local processor	The microprocessor connected to the PowerPRS 64G with the serial host interface (SHI) and used to control device configuration.
look-up table	One of two look-up tables that allow the byte transmission sequence of egress packets to be rearranged. One look-up table designates the byte transmission sequence for the master data stream and the other look-up table designates the byte transmission sequence for the slave data stream.
LSB	least significant bit
LSSD	level-sensitive scan design
LU	logical unit. The part of a packet that is processed by one PowerPRS 64G subswitch element.
LVCMOS	low-voltage complementary metal-oxide semiconductor
mask	A bit pattern used to change (reject) or extract (accept) bit positions in another bit pattern. For example, when the Boolean AND operation is used to match a mask of ‘0’s and ‘1’s with a string of data bits and a ‘1’ occurs in both the mask and the data, the resulting bit will contain a ‘1’ in that position. Hardware interrupts are enabled and disabled in this manner, with each interrupt assigned a bit position in the *Interrupt Mask Register* (page 65).
Mbps	megabits per second
MBps	megabytes per second
memory bank	A physical section of a device memory.
minimum eye	The part of the DASL receiver error-detection function used to detect bit synchronization errors caused by excessive transient noise (that is, electrostatic discharge). These errors are characterized by eye closures that are detected by an edge-sampling circuit. An error exists when an edge is detected within the minimum-eye region of the data sample.



MISR	multiple-input signature register
mode	An operational state of at least two possible conditions to which a system can be switched.

MSB	most significant bit
multicast	A one-to-many transmission, where “many” is specifically defined (contrast with broadcast).
multiplexer	A device that merges several low-speed transmissions into one high-speed transmission and vice versa.
network processor	A programmable CPU chip optimized to perform the packet processing supported by the PowerPRS 64G.
oversampling	The process of creating a more accurate digital representation of an analog signal by sampling the analog signals some number of times per second (frequency) and converting them into binary signals. Sampling requires at least two times the bandwidth of the sampled frequency.
packet	The user data element that the PowerPRS 64G processes in equal lengths called logical units (LUs). Depending on the device configuration, one packet is made of two, four, or eight LUs. Data packets are prioritized from zero to three, with zero being the highest priority.
packet header	The first two to five bytes in a packet that contain destination bitmap, packet priority, and switch redundancy support information, all protected by a parity bit.
packet priority	Four levels of data packet priority provide quality-of-service support. Data packets are prioritized from zero to three, with zero being the highest priority.
packet qualifier byte	The first byte (H0) of the packet header. This byte contains important information about the packet, such as packet type, packet priority, and so forth.
parity	The number of bits (or the number of similar bits) are even or odd, as intended.
payload	The part of the packet that carries the message data (contrast with packet header).
pin	The male lead on a chip that is plugged into its female counterpart to complete the circuit. The number of pins reflects the number of wires, or pathways, that can carry signals.
pinout	A diagram of the integrated circuit that shows the locations of the pins for various functions.
PLL	phase-locked loop
port paralleling	An optional PowerPRS 64G configuration for reducing the number of device ports. Four ports are grouped to form one link, up to a maximum of eight links (or groups).
POS	packet over SONET (synchronous optical network)
PowerPRS 64G	The IBM-approved product nickname for the IBM PowerPRS 64G Packet Routing Switch.

PRPG	pseudorandom pattern generator
pulldown	A circuit that lowers the value of a connected device.
pullup	A circuit that raises the value of a connected device.
pulse	A transient signal of short duration, constant amplitude, and one polarity.
quality of service	The ability to define a level of performance.
queue	A temporary holding place for egress data.
RAM	random access memory
read/clear	A register field in which the value is cleared immediately after a read.
receiver	An electronic device that accepts signals, and processes or converts them for internal use.
register	A small, high-speed circuit that stores internal operation values, such as the address of the instruction being executed and the data being processed.
resistor	An electronic component that resists the flow of current in a device.
RLOS	receiver loss of signal
SCC	side communication channel
SDC	shared DASL controller
sequencer	The PowerPRS 64G component that controls the internal data flow by granting shared memory access to the input and output ports.
service packet	The packet that tests link liveness. Also referred to as a yellow packet.
SHI	serial host interface
skew	A timing change in a transmission signal.
SRAM	static random access memory
SRL	set/reset latch
standby	The state in which the PowerPRS 64G is not in operation, but can be immediately activated.
stream	A contiguous flow of bits, bytes, or data from one place to another.
subswitch element	One of two 32 × 32 device components, designated either master or slave, that house the PowerPRS 64G shared memory.
switch fabric	The PowerPRS 64G internal interconnect architecture that redirects the ingress and egress data flow.
switchover	The process of redirecting the data flow between the two redundant switch planes.

synchronization packet	The packet that provides the bit transition and packet delineation necessary for link synchronization.
TDM	time-division multiplexing
tolerance	The amount of error allowed in a value, rating, dimension, and so forth.
traffic	Data crossing the network.
trailer	The last byte of each LU.
transceiver	A read/write data terminal capable of transmitting and receiving analog or digital signals.
transmitter	An electronic device that generates signals.
unicast	A single transmission.
UTOPIA-3-Like	An interface similar to the universal test and operations physical layer interface used in ATM network technology.
WAN	wide-area network
yellow packet	See “service packet.”

14. Related Documents

ASIC SA-27E Databook, Part I: Base Library and I/Os, IBM Corporation, January 2002 (contact your IBM representative for access to this document).

ASIC SA-27E Databook, Part II: Macros, IBM Corporation, January 2002 (contact your IBM representative for access to this document).

High-Speed Transceiver Logic (HSTL): A 1.5-V Output Buffer Supply Voltage-Based Interface Standard for Digital Integrated Circuits, Electronic Industries Association/JEDEC Standard No. 8-6, August 1995.

IBM Packet Routing Switch PRS28.4G Version 1.7 Datasheet, IBM Corporation, February 2001 (see http://www-3.ibm.com/chips/techlib/techlib.nsf/productfamilies/Networking_Technology).

IBM Packet Routing Switch Serial Interface Converter Datasheet, Preliminary, IBM Corporation, January 2002 (see http://www-3.ibm.com/chips/techlib/techlib.nsf/products/Packet_Routing_Switch_Serial_Interface_Converter).

IBM PowerNP NP4GS3 Network Processor Datasheet, Preliminary, IBM Corporation, February 2002 (see http://www-3.ibm.com/chips/techlib/techlib.nsf/products/IBM_PowerNP_NP4GS3).

IEEE Standard Test Access Port and Boundary-Scan Architecture, IEEE Standard 1149.1-1990, IEEE Standards Association, 1990.

Revision Log

Revision Date	Contents of Modification
06/01/00	Initial release (revision 00).
08/31/00	First release (01). Changed reference to document type from Databook to Datasheet. Changed page header labels from IBM Packet Routing Switch PRS64G to IBM Packet Routing Switch and from IBM32SW0640DSLCDA250 to PRS64G. Changed references to IBM64G Packet Routing Switch to IBM Packet Routing Switch PRS64G or PRS64G, as appropriate.
07/10/01	Second revision (02). Incorporated significant technical content changes throughout the document. Edited the document in its entirety for internal consistency and consistency with related product datasheets. Updated and expanded the *Glossary.*
07/02/02	Third revision (03). Updated the product name to follow PowerPRS product naming conventions. Corrected *Figure 2-2* on page 14; changed the PowerPRS 64G port configuration from 32 to 16. Edited the text that references the figure (page 13) accordingly. Made minor editorial changes to improve readability. Updated *Related Documents* on page 157.