PowerSpan II™

PowerPC-to-PCI Bus Switch

User Manual

Document Number: 80A1010_MA001_06
Document Status: Final
Release Date: December 2002

This document discusses the features, capabilities, and configuration requirements of the PowerSpan II. It is intended for hardware and software engineers who are designing system interconnect applications with the PowerSpan II.

TUNDRA

Tundra Semiconductor Corporation

Trademarks

Copyright

Disclaimer

Tundra assumes no responsibility for the accuracy or completeness of the information presented which is subject to change without notice. In no event will Tundra be liable for any direct, indirect, special, incidental or consequential damages, including lost profits, lost business or lost data, resulting from the use of or reliance upon the information, whether or not Tundra has been advised of the possibility of such damages.

Mention of non-Tundra products or services is for information purposes only and constitutes neither an endorsement nor a recommendation.

Corporate Profile

Tundra Semiconductor Corporation

Tundra Semiconductor Corporation (TSE:TUN) designs, develops, and markets advanced System Interconnect for use by the world’s leading Internet and communications infrastructure vendors. Tundra chips provide the latest interface and throughput features to help these vendors design and deliver more powerful equipment in shorter timeframes. Tundra products are essential to a range of applications, including telecommunications, data communications, wireless communications, industrial automation, and ruggedized systems. Tundra headquarters are located in Kanata, Ontario, Canada, and sales offices are based in Mountain View, California and Maidenhead, U.K. Tundra sells its products worldwide through a network of direct sales personnel, independent distributors, and manufacturers’ representatives. More information is available online at www.tundra.com.

Greater Demand, Greater Opportunity

The increasingly complex requirements placed on the Internet, intranets and extranets have created an insatiable demand for higher speed and greater capacity in communications networks. The evolution of converging communications networks requires higher levels of security and increasingly sophisticated network intelligence. These network demands, and the user expectations that drive them, have created a global need for well-managed and ever-increasing bandwidth.

Tundra helps meet those demands by creating underlying technology that enables the accelerated flow of voice, data, and video information over communications networks. Tundra products can be found in a broad range of applications, including telecommunications, data communications, wireless communications, industrial automation, and avionics. Communications infrastructure vendors rely on Tundra for off-the-shelf, standards-based, easy-to-deploy and highly scalable System Interconnect products.

Tundra System Interconnect

Tundra is *System Interconnect*. Tundra uses the term System Interconnect to refer to the technology used to connect all the components and sub-systems in almost any embedded system. This concept applies to the interfacing of functional elements (CPU, memory, I/O complexes, etc.) within a single-board system, and the interfacing of multiple boards in a larger system.

System Interconnect is a vital enabling technology for the networked world. The convergence of voice, video, and data traffic, the need for more secure communications, and the exploding demand for high-speed network access are putting communications infrastructure vendors under intense pressure to provide faster, well-managed bandwidth that also integrates smoothly with existing technology. Tundra System Interconnect helps these vendors address their customer needs. It enables them to build standards-based network equipment that can scale to multi-gigahertz speeds and also integrate with existing infrastructure.

Partnerships

Fundamental to the success of Tundra is its partnerships with leading manufacturers, including Motorola, Compaq and Texas Instruments. As a result of these alliances, Tundra devices greatly influence the design of customers' architectures. Customers are changing their designs to incorporate Tundra products. This highlights the commitment Tundra holds to be a significant part of its customers' success.

The Tundra design philosophy is one in which a number of strategic customers are invited to participate in the definition, design, test, and early silicon supply phases of product development. Close working relationships with customers and clear product roadmaps ensure that Tundra can anticipate and meet the future directions and needs of communications systems designers and manufacturers.

Tundra Customers

Tundra semiconductor products are used by the world's leading communications infrastructure vendors, including Cisco, Motorola, Ericsson, Nortel, Lucent, IBM, Xerox, Hewlett-Packard, 3Com, Nokia, Siemens, Alcatel, Matsushita, OKI, Fujitsu, Samsung, and LGS.

Tundra Customer Support

Tundra is respected throughout the industry for its outstanding commitment to customer support. Tundra ensures that its customers can take immediate advantage of the company's products through its Applications Engineering Group, unmatched Design Support Tools (DST), and full documentation. Customer support also includes Web-based and telephone access to in-house technical resources.

Tundra System Interconnect … Silicon Behind the Network ™

Contact Information

Tundra is dedicated to providing its customers with superior technical documentation and support. The following types of support are available:

Webpages

Product information

www.tundra.com/PowerSpan II describes PowerSpan II's features, benefits, typical applications, and block diagram. This webpage also provides links to other product-related information located on the Tundra website.

Design Support Tools (DST)

www.tundra.com/dst contains an extensive collection of technical documents that explain PowerSpan II's features and how to implement them. Some of the DST resources include the device manual, manual addenda, application notes, design notes, and device errata.

Once you register for access to the Design Support Tools webpage you can opt to receive email notification when a resource is added or changed.

FAQ database

www.tundra.com/faq is a support database that contains answers to common technical questions fielded by our knowledgeable Technical Support team.

Sales support

www.tundra.com/sales contains information that will help you locate a Tundra sales representative nearest you.

Email

Technical support	Use support@tundra.com to send technical questions and feedback to our Technical Support team. Please include PowerSpan II in the subject header of your message.
Document feedback	Use docfeedback@tundra.com to provide feedback on the *PowerSpan II Bus Switch User Manual*.
Document ordering	Use docs@tundra.com to order printed copies of Tundra product manuals (Final status only). Please include PowerSpan II in the subject header of your message.

Phone/Fax

Phone	613-592-0714 or 1-800-267-7231
Fax	613-592-1320

Mail

Tundra headquarters

Tundra Semiconductor Corporation
603 March Road
Kanata, ON
K2K 2M5

Contents

List of Figures

List of Tables

About this Document

This chapter discusses general document information about the PowerSpan II *User Manual*. The following topics are described:

- "Revision History" on page 25
- "Document Conventions" on page 26
- "Related Documents" on page 28

Revision History

80A1010_MA001_05, Final, November 2002

The Single PCI PowerSpan II has reached production status. This manual represents the production information for the Single PCI PowerSpan II.

80A1010_MA001_04, Preliminary, April 2002

80A1010_MA001_03, Advance, July 2001

80A1010_MA001_02, Advance, May 2001

80A1010_MA001_01, Advance, May 2001

Document Conventions

This section explains the document conventions used in this manual.

Signal Notation

Signals are either active high or active low. Active low signals are defined as true (asserted) when they are at a logic low. Similarly, active high signals are defined as true at a logic high. Signals are considered asserted when active and negated when inactive, irrespective of voltage levels. For voltage levels, the use of 0 indicates a low voltage while a 1 indicates a high voltage.

For voltage levels, the use of 0 indicates a low voltage while a 1 indicates a high voltage. For voltage levels, the use of 0 indicates a low voltage while a 1 indicates a high voltage.

Each signal that assumes a logic low state when asserted is followed by an underscore sign, "_". For example, SIGNAL_ is asserted low to indicate an active low signal. Signals that are not followed by an underscore are asserted when they assume the logic high state. For example, SIGNAL is asserted high to indicate an active high signal.

Bit Ordering Notation

This document adopts the convention that the most significant bit is always the largest number (also referred to as *Little-Endian* bit ordering). For example, the PCI address/data bus consists of AD[31:0], where AD[31] is the most significant bit and AD[0] is the least-significant bit of the field. However, this device enables endian conversion and endian-ness is a programmable feature.

Object Size Notation

The following object size conventions are used:

- A *byte* is an 8-bit object.
- A *word* is a 16-bit (2 byte) object.
- A *doubleword* is a 32-bit (4 byte) object.
- A *quadword* is a 64-bit (8 byte) object.
- A *Kword* is 1024 16-bit words.

Numeric Notation

The following numeric conventions are used:

- Hexadecimal numbers are denoted by the prefix *0x*. For example, 0x04.
- Binary numbers are denoted by the suffix *b*. For example, 10b.

Typographic Notation

The following typographic conventions are used in this manual:

- *Italic* type is used for the following purposes:
  - — Book titles: For example, *PCI Local Bus Specification.*
  - — Important terms: For example, when a device is granted access to the PCI bus it is called the *bus master.*
  - — Undefined values: For example, the device supports four channels depending on the setting of the PCI_D*x* register.
- `Courier` type is used to represent a file name or text that appears on a computer display. For example, “run `load.exe` by typing it at a command prompt.”

Symbols Used

The following symbols are used in this manual.

Tip	This symbol indicates a basic design concept or information considered helpful.
📝	This symbol indicates important configuration information or suggestions.
⚠	This symbol indicates procedures or operating levels that may result in misuse or damage to the device.

Document Status Information

Tundra technical documentation is classified as either Advance, Preliminary, or Final:

- Advance: The Advance manual contains information that is subject to change and exists until prototypes are available. This type of manual can be downloaded from our website at www.tundra.com.
- Preliminary: The Preliminary manual contains information about a product that is near production-ready, and is revised as required. The Preliminary manual exists until the product is released to production. This type of manual can be downloaded from our website at www.tundra.com.
- Final: The Final manual contains information about a final, customer-ready product. This type of manual can be downloaded from our website. It can also be ordered in print format by calling 613-592-0714 or 1-800-267-7231 (please ask for customer service), or by email at docs@tundra.com.

Related Documents

The following documents are useful for reference purposes when using this manual.

PCI Local Bus Specification (Revision 2.2)

This specification defines the PCI hardware environment including the protocol, electrical, mechanical and configuration specification for the PCI local bus components and expansion boards. For more information, see www.pcisig.com.

PCI-to-PCI Bridge Architecture Specification (Revision 1.1)

This specification establishes the minimum behavioral requirements that PCI-to-PCI bridges must meet to be compliant to the PCI Local Bus Specification. Recommendations and guidance on optional PCI-to PCI bridge features are also provided by this specification. For more information, see www.pcisig.com.

PCI Bus Power Management Interface Specification (Revision 1.1)

This specification defines power management capabilities that enhance the base PCI architecture. For more information, see www.pcisig.com.

CompactPCI Hot Swap Specification (Revision 2.0)

This specification defines a process for installing and removing CompactPCI boards without adversely effecting a running system. For more information, see www.picmg.com.

I^2C Specification

This specification defines the standard I2C bus interface, including specifications for all the enhancements. For more information, see www-eu2.semiconductors.com/i2c.

1. Functional Overview

This chapter describes the PowerSpan II’s architecture. The following topics are discussed:

- “PCI Interface” on page 35
- “Processor Bus Interface” on page 37
- “DMA Controller” on page 38
- “I2C / EEPROM” on page 38
- “Concurrent Reads” on page 38

1.1 Overview

The Tundra PowerSpan II is a PowerPC-to-PCI Bus Switch that bridges PCI to the PowerQUICC II (MPC8260, and MPC8260A), MPC7400, MPC7410, PowerPC 7400, PowerPC 750, PowerPC 740, and PowerPC 603e processors. PowerSpan II is available in either single PCI or dual PCI variants. PowerSpan II defines a new level of PCI bus switch flexibility.

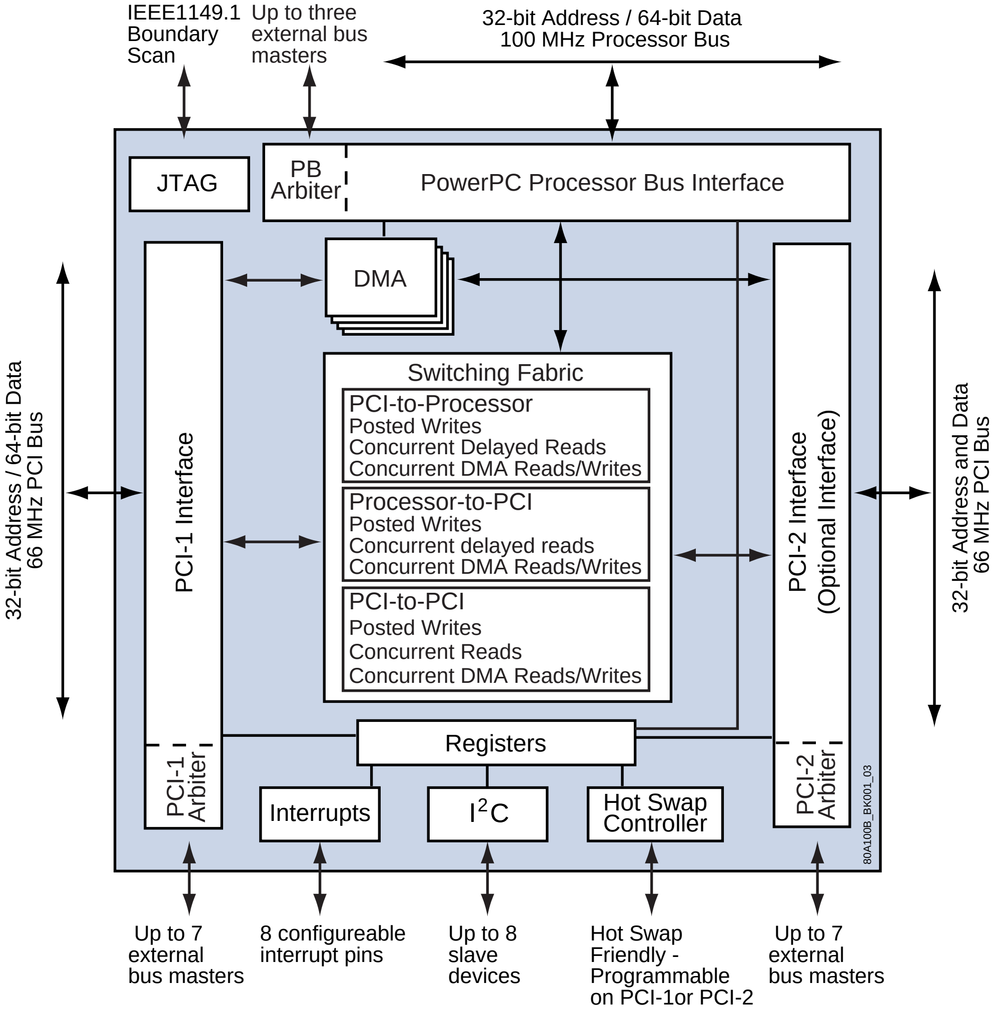
The integrated, non-transparent PCI-to-PCI bridge in the Dual PCI PowerSpan II provides a significant opportunity for designers to reduce component count and increase overall system performance.

PowerSpan II offers a flexible package design. The design is available in both the original PowerSpan II package dimensions and newly designed, smaller packages.

The high level of performance and flexibility of PowerSpan II is made possible through *Switched PCI* - unique to PowerSpan II. Switched PCI uses a switching fabric to enable data streams to pass from port-to-port across the multi-ported PowerSpan II without collision. This improves the burst performance and decreases latency on the PCI and processor busses — a key element of I/O processor performance.

PowerSpan II is also part of a chipset offering with the Tundra PowerPC Memory Controller — PowerPro. When PowerSpan II is in a application with both PowerPro and the MPC8260, PowerPro can be configured to implement ECC protection in the system. Please refer to the *PowerPro PowerPC Memory Controller Manual* for more information

Figure 1: PowerSpan II Block Diagram


IEEE1149.1 Boundary Scan
Up to three external bus masters
32-bit Address / 64-bit Data 100 MHz Processor Bus
JTAG
PB Arbiter
PowerPC Processor Bus Interface
DMA
32-bit Address / 64-bit Data 66 MHz PCI Bus
PCI-1 Interface
Switching Fabric
PCI-to-Processor
Posted Writes
Concurrent Delayed Reads
Concurrent DMA Reads/Writes
Processor-to-PCI
Posted Writes
Concurrent delayed reads
Concurrent DMA Reads/Writes
PCI-to-PCI
Posted Writes
Concurrent Reads
Concurrent DMA Reads/Writes
PCI-2 Interface (Optional Interface)
32-bit Address and Data 66 MHz PCI Bus
PCI-1 Arbiter
Registers
PCI-2 Arbiter
Interrupts
I2C
Hot Swap Controller
80A100B_BK001_03
Up to 7 external bus masters
8 configureable interrupt pins
Up to 8 slave devices
Hot Swap Friendly - Programmable on PCI-1or PCI-2
Up to 7 external bus masters

1.1.1 PowerSpan II Features

PowerSpan II has the following features:

1.1.1.1 Processor Support

- Direct connect interface for embedded PowerPC processors MPC8260, MPC8260A, MPC74xx, PowerPC 7400, PowerPC 740, and PowerPC 750
- 25 MHz-to-100 MHz bus frequency
- Programmable endian conversion
- MPC8260 Configuration Slave support for power-up options
- Eight programmable memory maps to PCI from the processor
- Processor bus arbiter with support for three requesters

1.1.1.2 PCI Support

- Dual PCI PowerSpan II:
  - One 32-bit or 64-bit PCI interface
  - One 32-bit interface
  - 66 MHz operation
- Single PCI PowerSpan II:
  - One 32-bit or 64-bit PCI interface
  - 66 MHz operation
- Integrated, non-transparent PCI-to-PCI bridge in the Dual PCI PowerSpan II
- PCI arbiters on each PCI interface
- CompactPCI Hot Swap Friendly
- *PCI 2.2 Specification* compliant

1.1.1.3 Packaging options

- Single PCI PowerSpan II (CA91L8260B)
  - 64-bit/66MHz
  - 420 HSBGA: 1.27mm ball pitch, 35mm body size
  - 484 PBGA: 1.0mm ball pitch, 23mm body size
- Dual PCI PowerSpan II (CA91L8200B)
  - 32-bit/66MHz and 64-bit/66MHz

- — 480 HSBGA: 1.27mm ball pitch, 37.5mm body size
- — 504 HSBGA: 1.0mm ball pitch, 27mm body size

1.1.2 PowerSpan II Benefits

PowerSpan II offers the following benefits to designers:

- Smaller packages reduce board area required for system design.
- Integrated PCI bus, processor bus arbiters decrease individual component count on boards.
- Flexible PCI interfaces enable PowerSpan II to meet many different application requirements.
- Integrated, non-transparent PCI-to-PCI bridge connects traffic between the two PCI interfaces. This decreases individual component count and simplifies conventional CompactPCI board architecture.
- Supports reads from multiple I/O devices in parallel, non-blocking streams which decreases bus latency.
- The Tundra PowerQUICC II and PowerPC architecture expertise is passed to customers with the Design Support Tools. Tundra offers a complete package of Design Support Tools including software, schematics, and evaluation board. For more information refer to www.tundra.com.
- Detailed design documentation including device manuals, and application notes.

1.1.3 Typical Applications

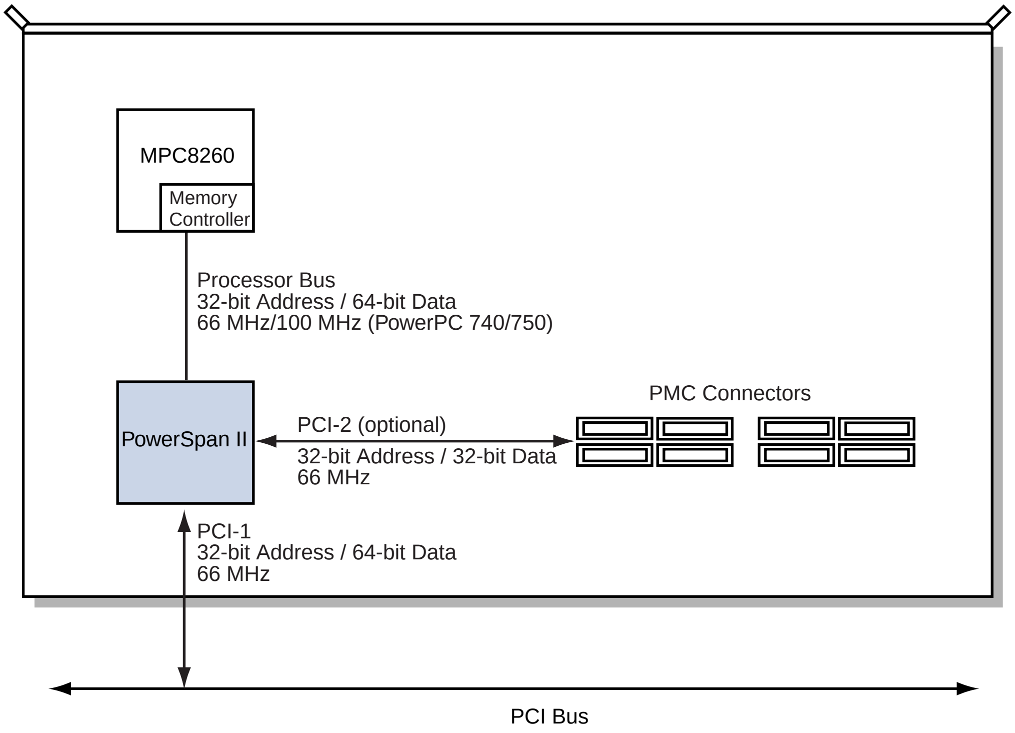
Tundra understands vendors' needs to increase performance throughout today's communications networks. From premise equipment to local carrier gear to high-end switches, designer's need to deliver ever-faster traffic through the same or smaller footprint at a reduced cost. Tundra System Interconnect helps in that effort by providing features and benefits across all areas of the network. PowerSpan II helps designers working on infrastructure equipment in the following areas:

Table 1: PowerSpan II Applications

LAN/WAN	Remote/Local Access	Wireless
Exchange Carrier Switching Equipment	ADSL Concentrators	Third Generation (3G) Base Stations
Ethernet Switches	VoIP Gateways	
MPEG 2 Encoders	VPN Equipment	

PowerSpan II is a very flexible device. The following diagram shows a typical PowerPC system architecture using PowerQUICC II and the
Dual PCI PowerSpan II.

Figure 2: Typical PowerSpan II Application


MPC8260
Memory Controller
Processor Bus
32-bit Address / 64-bit Data
66 MHz/100 MHz (PowerPC 740/750)
PMC Connectors
PowerSpan II
PCI-2 (optional)
32-bit Address / 32-bit Data
66 MHz
PCI-1
32-bit Address / 64-bit Data
66 MHz
PCI Bus

1.1.4 PowerSpan II and PowerSpan Differences Summary

The following table summarizes the main PowerSpan II programmable features that were unavailable in the PowerSpan device. All functional enhancements are programmable in order to make sure that all original PowerSpan functionality can be exercised. For detaill information, please refer to the PowerSpan II *and PowerSpan Differences Summary* available on the Tundra website at www.tundra.com..

Table 2: PowerSpan II Functional Enhancements

Functional Enhancement Descriptions	See
Packaging Change Packaging has been changed from HPBGA packages to HSBGA packages. Four variants are available for PowerSpan II: two variants for the Single PCI PowerSpan II and two variants for the Dual PCI PowerSpan II. Both the Single and Dual PCI PowerSpan II have packages, signals, and pins that are backwards compatible with the original PowerSpan device.	“Electrical and Signal Characteristics” on page 277 and “Package Information” on page 443
New Revision ID PowerSpan II has a new ID.	“Register Descriptions” on page 283
Read implementation PowerSpan II supports 4 byte transactions.	“PCI Interface” on page 43, “Processor Bus Interface” on page 101, and “Register Descriptions” on page 283
True Little-endian Mode A new endian mode was developed for PowerSpan II	“Endian Mapping” on page 213, and “Register Descriptions” on page 283
Base Address Implementation PowerSpan II supports a PCI base address of 0x00000.	“Register Descriptions” on page 283
Maximum Retry Counter Modification The maximum retry counter is programmable in PowerSpan II	“Register Descriptions” on page 283

Table 2: PowerSpan II Functional Enhancements

Functional Enhancement Descriptions	See
Arbitration Timing for Masters PowerSpan II measures the length of time it takes a master to respond to the GNT# signal.	“Arbitration” on page 163 and “Register Descriptions” on page 283
PowerPC 7400 Transaction Support PowerSpan II has been designed to support specific PowerPC 7400 misaligned transactions.	“Processor Bus Interface” on page 101 and “Register Descriptions” on page 283
Delay Sampling of Transaction Start Signal The PowerSpan II PB arbiter can be programmed to sample requests two clocks after the PB_TS_ signal is asserted.	“Arbitration” on page 163 and “Register Descriptions” on page 283
Programmable DMA Block Size PowerSpan II enables programmable DMA block sizes.	“DMA” on page 137 and “Register Descriptions” on page 283
PB Arbiter Qualifies Bus Grants The PowerSpan II PB Arbiter can be programmed to qualify data bus grants before issuing data bus grants.	“Arbitration” on page 163 and “Register Descriptions” on page 283
Target Fast Back to Back Capable (TFBBC) The default setting of this bit was changed to 0 in PowerSpan II; the device does not support fast back-to-back transactions.	“Register Descriptions” on page 283

1.2 PCI Interface

PowerSpan II is available as a Single PCI PowerSpan II or Dual PCI PowerSpan II. A 64-bit PCI Interface is available on both variants; the Dual PCI PowerSpan II has a 32-bit PCI Interface in addition to the 64-bit PCI Interface. In both cases, the PCI Interfaces on the PowerSpan II support 66MHz operation and are asynchronous to the other interfaces on the device.

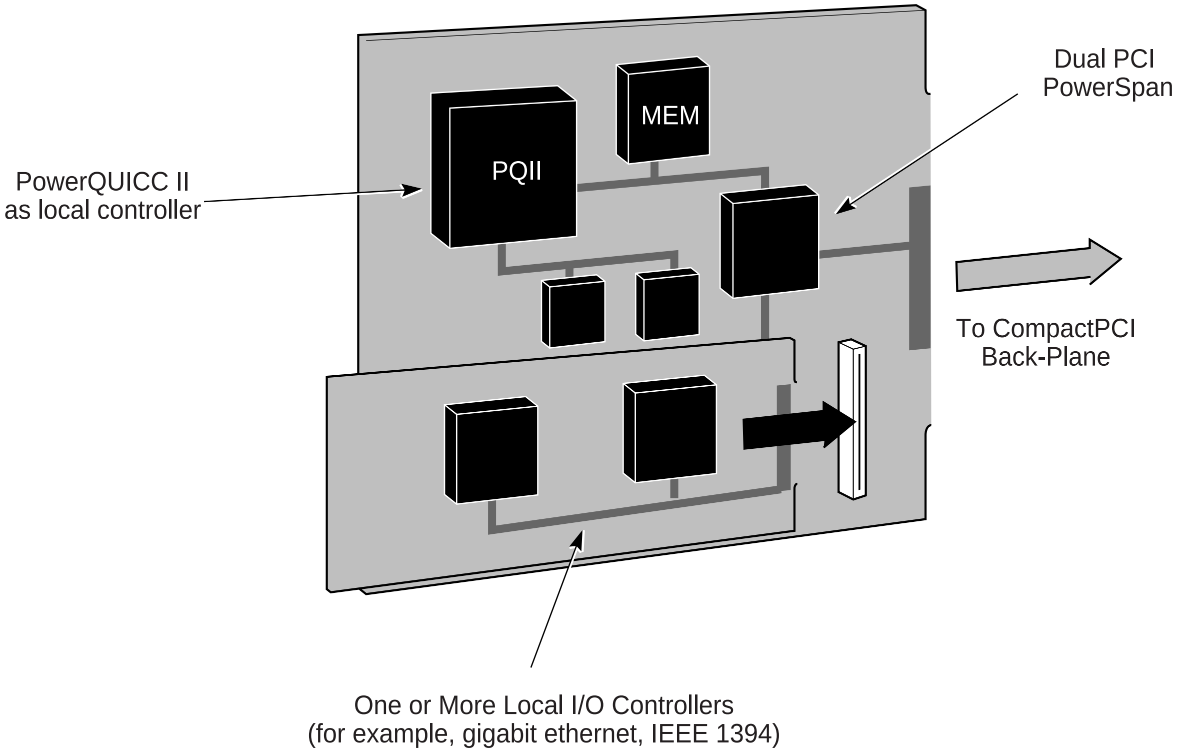
The PCI interfaces are *PCI 2.2 Specification* compliant.

1.2.1 PCI-to-PCI Bridge

The Dual PCI PowerSpan II is a PCI-to-PCI bridge. It connects traffic between the two PCI interfaces. This PCI-to-PCI bridging function is “non-transparent”. In a non-transparent bridge one PCI bus is hidden from system BIOS running in the other PCI domain. Memory and I/O transfers pass freely between the PCI interfaces, but Configuration accesses are filtered.

The application is shown in Figure 3.

Figure 3: Non-transparent PCI-to-PCI in CompactPCI Application


Dual PCI PowerSpan
MEM
PQII
PowerQUICC II as local controller
To CompactPCI Back-Plane
One or More Local I/O Controllers (for example, gigabit ethernet, IEEE 1394)

Because of the non-transparent PCI-to-PCI bridging, the MPC8260 on the CompactPCI adapter card acts as local host without the local PCI devices being configured by the CompactPCI system host.

1.2.2 Primary PCI Interface

The PowerSpan II provides extra functionality for one of the PCI interfaces. The PCI Interface assigned extra functionality must be specified as Primary PCI Interface through a power-up option.

The Primary PCI Interface functions are:

- CompactPCI Hot Swap Friendly support

- I_2O *2.0 Specification* compliant messaging
- Vital Product Data support.

This extra functionality is available for the Single PCI PowerSpan II and the Dual PCI PowerSpan II.

1.2.3 PCI Host Bridge

PowerSpan II is designed for host bridge applications. The PowerPC processor generates configuration cycles on the PCI bus in the same way as that found in conventional PCI host bridges. In addition, with concurrent reads and low device latency, the PCI Target Interface on PowerSpan II is specifically designed to allow low latency access to host packet memory for I/O controllers on either of the PCI buses.

1.2.4 PCI Bus Arbitration

Each PCI Interface has an integrated PCI bus arbiter. Each arbiter supports four external bus requesters. An additional three bus requesters can be assigned between the two PCI arbiters.

The PCI arbiters implement a fairness algorithm, two round robin priority levels and flexible bus parking options.

1.3 Processor Bus Interface

The PowerSpan II provides a direct-connect 64-bit interface for the PowerQUICC II (MPC8260, and MPC8260A), MPC7400, MPC7410, PowerPC 7400, PowerPC 750, PowerPC 740, and PowerPC 603e processors. . The direct-connect support for these interfaces has been extensively verified during product development with processor functional models as well as with a hardware emulation methodology. This verification ensures any potential interface issues are identified and resolved by Tundra Semiconductor before PowerSpan II customers begin to design their own systems.

PowerSpan II supports MPC8260 extended cycles on the Processor Bus Interface. Extended cycle support on the MPC8260 means more flexible bursting and more efficient use of the processor bandwidth.

1.3.1 Address Decoding

Instead of consuming chip selects from the MPC8260, PowerSpan II performs its own address decoding for up to eight memory (slave) images to the PCI bus from the processor bus. This allows a flexible mapping of processor transactions to PCI cycle types.

1.3.2 Processor Bus Arbitration

The Processor Bus Interface has an integrated bus arbiter. The Processor Bus Interface supports three external bus masters for applications involving multiple processors. The processor bus arbiter implements two levels of priority, where devices programmed into a specific priority level operate in a round robin fashion in that level.

1.4 DMA Controller

PowerSpan II provides four independent, bidirectional DMA channels. Each DMA channel is capable of Linked-List or Direct mode transfers.

Each DMA channel transfers data from any-port to any-port. For example, from PCI-1 to PCI-2, Processor Bus to PCI-1, or Processor Bus and Processor Bus. High throughput data transfer is coupled by flexible endian mapping and a range of status bits mappable to external interrupts.

1.5 I^{2}C / EEPROM

1.5.1 EEPROM

PowerSpan II registers can be programmed by data in an EEPROM at system reset. This enables board designers to set unique identifiers for their cards on the PCI bus at reset, and set various image parameters and addresses. Configuring PowerSpan II with the EEPROM allows PowerSpan II to boot-up as a Plug and Play compatible device. PowerSpan II supports reads from, and writes to, the EEPROM.

1.5.2 I^{2}C

PowerSpan II has a master only I^{2}C bus compatible interface which supports up to eight I^{2}C slave devices. This interface is used by PowerSpan II for the initialization of registers and for reading and writing PCI Vital Product Data (VPD).

PowerSpan II also provides a mechanism to perform master read and write operations to EEPROMs or other I^{2}C compatible slave devices.

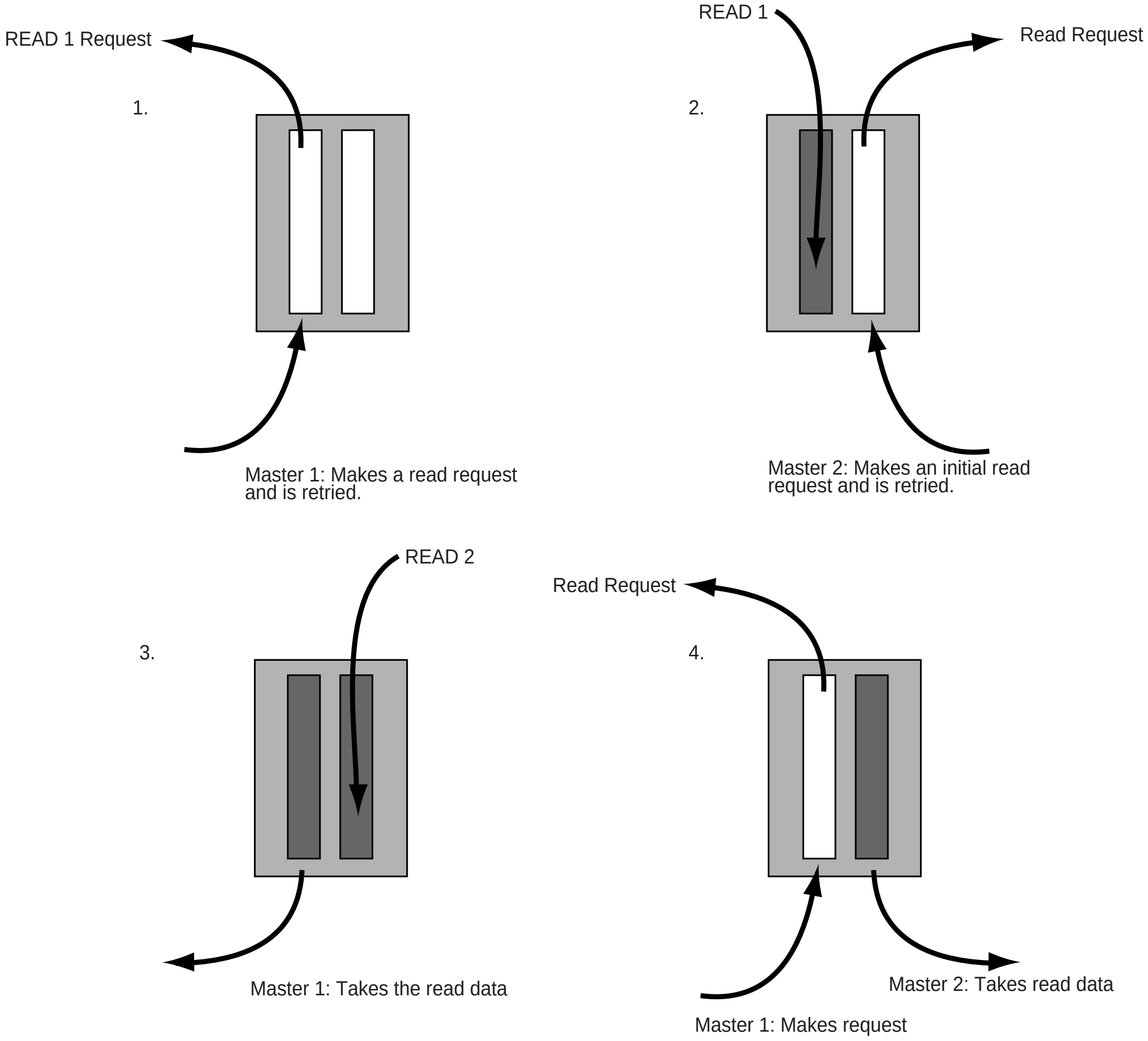
1.6 Concurrent Reads

PowerSpan II’s Switched PCI architecture enables concurrent reads through a single channel. This ability greatly reduces read latency, which is often the limiting factor in PCI performance.

1.6.1 PowerSpan II’s Concurrent Read Solution

With PowerSpan II’s concurrent reads, read requests are accepted even while the current read is in progress. Figure 4 illustrates the concurrent read process with the PowerSpan II.

Figure 4: Concurrent Read Process with PowerSpan II


READ 1 Request
1.
Master 1: Makes a read request and is retried.
READ 1
Read Request
2.
Master 2: Makes an initial read request and is retried.
READ 2
3.
Master 1: Takes the read data
Read Request
4.
Master 2: Takes read data
Master 1: Makes request

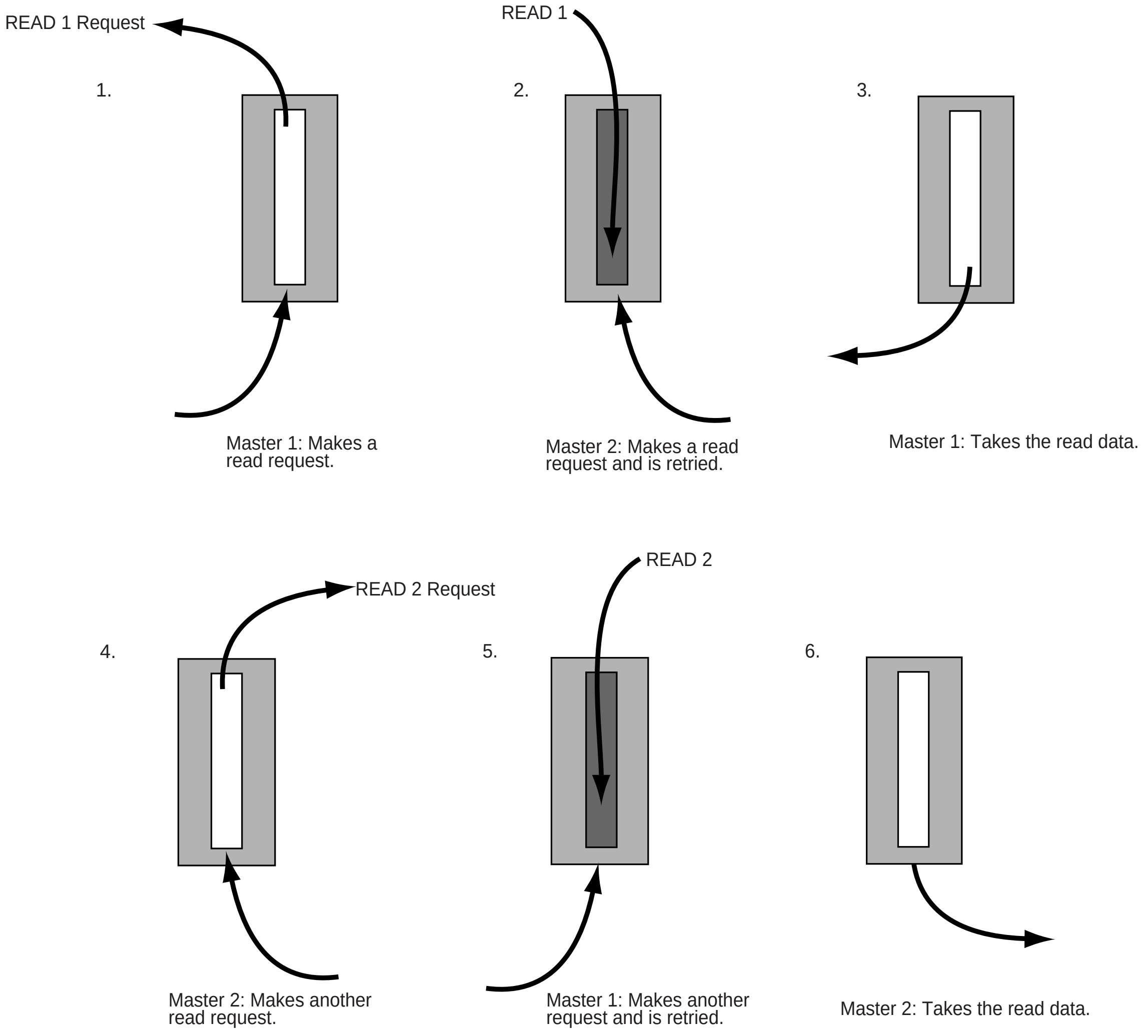
When Master 2 makes its first read request in Step 2, it is retried but information about the read request is latched and initiates a read on the other bus. This occurs even though a read is in progress for Master 1.

PowerSpan II can simultaneously support two reads to the Processor Bus and two reads to the PCI bus.

1.6.1.1 Conventional Reads and Retries

In conventional FIFO-based bridge architectures, bus masters must take turns for read opportunities and incur multiple retries while waiting. Figure illustrates the read process for subsequent reads where retries are incurred while a pending read is completed.

Figure 5: Reads with Conventional FIFO-Based Bridges


READ 1 Request
1.
Master 1: Makes a read request.
READ 1
2.
Master 2: Makes a read request and is retried.
3.
Master 1: Takes the read data.
READ 2 Request
4.
Master 2: Makes another read request.
READ 2
5.
Master 1: Makes another request and is retried.
6.
Master 2: Takes the read data.

When Master 2 is retried in Step 2, no information is latched about the read request. When Master 2 returns for a subsequent read request in Step 4, it is treated by the bridge as the first read request.

1.6.2 PowerSpan II’s Concurrent Read Applications

1.6.2.1 PCI Host Bridge

In a PCI host bridge application, all of the PCI masters — for example, I/O controllers — potentially receive only one retry before receiving read data. Even with another read pending, when the PCI Target Interface of the PCI host device receives a read request, it latches the information and begins another burst read prefetch on the processor bus. The PCI host bridge latches the addresses and delivers the data to each master using separate, dedicated buffering. This approach greatly reduces the overall system latency and allows for a more scalable I/O subsystem.

1.6.2.2 Adapter Card

In an adapter card application, the read latency problem is a mirror image of a PCI host bridge application. In an adapter card, the MPC8260 serial ports (FCCs) may be expected to transfer bit streams through the MPC8260/PCI to host memory across the PCI bus. In this case, there can be eight separate FCCs potentially contending for the processor slave interface in the PCI bridge — assuming there are two MPC8260s on the local bus. This architecture adds considerable latencies to read transactions because of FCCs attempting reads to host memory across the PCI bus. Ideally, each FCC would have a dedicated channel to the PCI bus so they do not have to share resources.

PowerSpan II supports this ideal situation through its concurrent reads in a flexible switching architecture. The PCI bridge latches information about the local read as it receives the read request even with reads pending. The FCCs can now receive transmit data from system host memory with far lower latencies.

2. PCI Interface

Peripheral Component Interconnect (PCI) is a bus protocol that defines how devices communicate on a peripheral bus and with a host processor. If a device is referred to as PCI compliant it must be compliant with the *PCI Local Bus Specification (Revision 2.2)*. A PCI bus supports frequencies up to 66 MHz, and 32-bit or 64-bit transfers.

This chapter describes the PCI Interface of the Dual PCI PowerSpan II. The following topics are discussed:

2.1 Overview

This chapter describes the functionality of the Dual PCI PowerSpan II. The Single PCI PowerSpan II is identified when its functionality or settings differ from the Dual PCI PowerSpan II.

The Single PCI PowerSpan II and the Dual PCI PowerSpan II have different characteristics. The features of each device are shown in the following list.

- Dual PCI PowerSpan II:
  - One 32-bit or 64-bit PCI interface

  - One 32-bit interface
  - 66 MHz operation
- Single PCI PowerSpan II:
  - One 32-bit or 64-bit PCI interface
  - 66 MHz operation

2.1.1 Primary PCI

The Dual PCI PowerSpan II has two PCI interfaces: the PCI-1 Interface and the PCI-2 Interface. PCI-1 interface is 32-bit or 64-bit capable, while the PCI-2 Interface is 32-bit. Both PCI interfaces have 66 MHz capability.

There are two settings available for the Dual PCI PowerSpan II: Primary PCI Interface and Secondary PCI Interface. The Primary PCI Interface adds extra functionality to the PCI Interface that is designated as the Primary PCI Interface. The Secondary PCI Interface has no extra functionality.

The following features are associated with the Primary PCI Interface:

- CompactPCI Hot Swap Support (see “CompactPCI Hot Swap Silicon Support” on page 67)
- Vital Product Data (see “Vital Product Data” on page 75)
- I_2O Shell Interface (see “I2O Shell Interface” on page 77)

The Primary PCI Bus (PRI_PCI) bit in the “Reset Control and Status Register” on page 385 is always 0 in the Single PCI PowerSpan II.

Either the PCI-1 Interface (64-bit) or the PCI-2 Interface (32-bit) can be configured as the Primary Interface. The selected PCI interface is assigned as the Primary PCI Interface through the Primary PCI Select (PWRUP_PRI_PCI) power-up option (see “Resets, Clocks and Power-up Options” on page 201 for more information). Primary PCI functionality is shown in the value of the Primary PCI Bus (PRI_PCI) bit in the “Reset Control and Status Register” on page 385. The PRI_PCI is a status bit and only shows which bus is primary. It does not enable a bus as the Primary PCI Interface. The Primary PCI interface is enabled with a power-up option (see 9. “Resets, Clocks and Power-up Options” on page 201).

2.1.1.1 Clock Frequencies

Each of the PCI interfaces, PCI-1 and PCI-2, run at frequencies from 25 MHz to 66 MHz. The DEV66 bit in the “PCI 1 Control and Status Register” on page 301 indicates that PowerSpan II is a 66 MHz-capable device.

The speed of these buses is determined through a power-up option (see "Clocks" on page 205 and "Power-Up Options" on page 205) using the corresponding P1_M66EN pins.

Both PCI interfaces run asynchronously to one another, and asynchronously to the Processor Bus Interface.

2.1.2 PCI Data Width

The PCI-1 Interface is a 64-bit data interface that supports 32-bit addressing. The PCI-2 Interface is a 32-bit data interface that supports 32-bit addressing.

2.1.2.1 PowerSpan II in non-Hot Swap and PCI Host Applications

The PCI-1 Interface can be programmed to assert P1_REQ64# to indicate the data width of the PCI-1 bus at reset. This feature is controlled by the PWRUP_P1_R64_EN power-up option (see "Power-Up Options" on page 205) and minimizes required external logic. A logic low applied to P1_64EN# enables this feature. PowerSpan II drives P1_REQ64# when PWRUP_P1_R64_EN is selected and P1_64EN# is set to 0.

When P1_64EN# is at a logic low, and PWRUP_P1_R64_EN is selected, P1_REQ64# is asserted low during reset. The status of PWRUP_P1_R64_EN is reflected in the P1_R64_EN bit in the "Reset Control and Status Register" on page 385.

This feature must only be used in systems where PowerSpan II controls both P1_REQ64# and P1_RST#. In this scenario, PowerSpan II is the Central Resource in the system and can ensure that timing parameters are satisfied.

2.1.2.2 PowerSpan II in non-Hot Swap and PCI Peripheral Applications

The PCI-1 Interface supports the following mechanisms for determining the width of the PCI-1 datapath:

- sampling P1_REQ64# at the negation of P1_RST#
- logic level on P1_64EN#

In non-Hot Swap applications, the P1_64EN# signal must be pulled high in order to enable sampling of P1_REQ64# to determine the width of the data path. The result of the sampling of P1_REQ64# is or'd with the logic level on P1_64EN# to determine data path width (see Table 3).

Table 3: Signals Involved in PCI Data Width Determination

Signal		
P1_REQ64#	**P1_64EN#**	**Result**
0	0	64-bit bus
1	0	64-bit bus
0	1	64-bit bus
1	1	32-bit bus

2.1.2.3 PowerSpan II in Hot Swap Applications

In Hot Swap applications the P1_64EN# signal is the only signal sampled to indicate the PCI data width. The following scenarios can be used for determining the proper implementation of the P1_REQ64# and P1_64EN# signals:

- PCI bus is currently a 32-bit slot and the Hot Swap board is 64-bit capable. In this case, P1_REQ64# is pulled up in the slot and P1_64EN# is OPEN and the card will initialize in 32-bit mode.
- PCI bus is currently a 32-bit slot and the Hot Swap board is 32-bit capable. In this case, P1_REQ64# is not sampled and P1_64EN# does not exist on the board so initialization would be 32-bit mode.
- PCI bus is currently a 64-bit slot and the Hot Swap board is 64-bit capable. In this case, P1_REQ64# could be anything but P1_64EN# is GND and the card will initialize in 64-bit mode.
- PCI bus is currently a 64-bit slot and the Hot Swap board is 32-bit capable. In this case, P1_REQ64# is not sampled and P1_64EN# does not exist on the board so initialization would be 32-bit mode.

2.1.2.4 PowerSpan II Drives PCI 64-bit Extension Signal in 32-bit Environment

When PowerSpan II's 64-bit PCI interface is programmed to operate in 32-bit mode, the 64-bit extension PCI bus signals can be left open. PowerSpan II actively drives the following the input signals:

- Driven Low
  — P1_AD[63:32]

  - P1_CBE[7:4]
  - P1_REQ64#
  - P1_ACK64#
  - P1_PAR64
  - P1_AD[63:32]
  - P1_CBE[7:4]
  - P1_PAR64

- Driven High
  - P1_REQ64#
  - P1_ACK64#

This insures the signals do not oscillate and that there is not a significant power drain through the input buffer.

2.1.3 PCI Interface Descriptions

The PowerSpan II PCI interfaces are described in terms of its PCI master and PCI target functions. This description is largely independent of PCI-1 versus PCI-2, or the assignment of the Primary PCI Interface functions. Exceptions to these rules are noted as required.

Cross-references to PCI registers are shown as PCI-1 whenever the cross-references apply equally to PCI-1 or PCI-2 registers.

2.1.4 Transaction Ordering

PowerSpan II implements a set of ordering rules for transactions initiated by master(s) connected to PCI Interface Px, that are destined for targets and/or slaves connected to PCI Interface Py.

Transactions initiated by master(s) connected to PCI Interface Px, but with different PowerSpan II destination interfaces, are independent from an ordering perspective. Transactions initiated by PowerSpan II DMA and PowerSpan II generated interrupt events have no ordering relationship to externally initiated transactions processed by PowerSpan II.

2.1.4.1 Transactions Between Px and Py

PowerSpan II implements the following transaction ordering rules for transactions flowing between PCI Interface Px and PCI Interface Py:

- The order in which delayed read requests are latched on the source bus, and posted memory write transactions are presented on the source bus, is the order in which they appear on the destination bus.
- Writes flowing from Px to Py have no ordering relationship to writes flowing from Py to Px.
- The acceptance of a posted write as a target or slave is not contingent on the completion of a transaction by the master of the same interface. PowerSpan II master and target/slave modules are independent.

2.1.4.2 Transactions Between the PB Interface and the PCI Interfaces

When there are transactions to the PB Interface from both PCI-1 and PCI-2, there is a possibility that a transaction from PCI-2 can be queued ahead of a transaction from
PCI-. This is caused by the fact there is no transaction ordering between the two independent PCI interfaces. For example, if transactions to the PB Interface arrive in the following order from PCI-1 and PCI-2:

- PCI-1 Write 1
- PCI-2 Write 1
- PCI-2 Write 2
- PCI-1 Write 2

The transactions can be completed to the PB Interface in the following order even though PCI-2 Write 2 entered PowerSpan II before PCI-1 Write 2:

- PCI-1 Write 1
- PCI-2 Write 1
- PCI-1 Write 2
- PCI-2 Write 2

This is caused by the fact that PCI-1 to PB Interface transactions and PCI-2 to PB Interface transactions arbitrate in a round robin fashion. When a PowerSpan II decision is required on whether to service a transaction from PCI-1 or PCI-2, writes are available at both even though at one point a write is only available from PCI-2.

2.1.4.3 DMA Transactions

DMA transactions and regular write/read transactions arbitrate for the use of a master interface in a round robin scheme. There are no special priorities for DMA transactions and regular write/read transactions.

Writes and reads from one source are queued and arbitrated for the use of the master interface with DMA in a round robin design. A DMA transaction can be given a lower priority by programming the DMA Channel Off Counter (OFF) bit in the DMA Control and Status register (see page 372). The OFF bit provides programmable control over the amount of source bus traffic generated by the DMA channel. The channel interleaves source bus transfers with a period of idle processor bus clocks where no source bus requests are generated. When source and destination interfaces are different, 256 bytes of source bus traffic occurs before the idle period. If source and destination interfaces are the same, 64 bytes of source bus traffic occur before the idle period. This helps prevent PowerSpan II from interfering with processor bus instruction fetches.

All transactions (writes/reads/DMA) from two source interfaces arbitrate in a round robin scheme on a per interface basis. Refer to “Transactions Between the PB Interface and the PCI Interfaces” on page 48 for more information.

2.1.4.4 PCI Transaction Ordering Rules

The *PCI 2.2 Specification* outlines transaction ordering rules for PCI transactions. PowerSpan II does not comply with the following PCI transaction ordering rules:

- PowerSpan II only completes the writes that are destined for the same bus as the initiated read when it is processing a read request. It does not complete writes in both directions before processing a read request. PowerSpan II does not prioritize writes over reads.
- PowerSpan II does not allow posted memory writes to pass delayed read requests. This implies that deadlock conditions may occur when the customer uses bridges that do not support delayed transactions. Deadlock conditions are broken by the PowerSpan II maximum retry counter.

2.2 PCI Target Interface

2.2.1 PowerSpan II as PCI Target

PowerSpan II participates in a transaction as a PCI target when a PCI master initiates one of the following actions:

- attempts to access the alternate PCI Interface
- attempts to access processor bus memory
- accesses PowerSpan II registers

This chapter describes only the first two conditions listed above. Transactions targeted for the PowerSpan II’s 4 Kbytes of device control and status registers are discussed in “Register Access” on page 283.

The operation of the PCI Target is described by dividing the PCI transaction into the following phases:

- **Address Phase**: This section discusses the decoding of PCI accesses.
- **Data Transfer**: This section describes control of burst length and byte lane management.
- **Terminations**: This section describes the terminations supported by the PowerSpan II, how they are mapped from the destination port to the PCI Target, and exception handling.

2.2.2 Address Phase

2.2.2.1 Transaction Decoding

Transaction decoding on the PCI Target operates in both normal decode mode and Master-based decode mode. Only memory and configuration cycles are decoded. I/O cycles are not decoded.

During normal decode mode, a PCI device monitors the Px_AD and Px_C/BE# lines to decode an access to some programmed PCI physical address range — through positive decoding.

A PCI target image is defined as the range of PCI physical address space to decode a PCI transaction. A PCI target image location and size is controlled using a Base Address field in the “PCI 1 Target Image X Base Address Register” on page 312, and a Block Size field in the “PCI 1 Target Image x Control Register” on page 322.

Normal address decoding only applies to memory cycles.

Table 4 illustrates the command encoding for PowerSpan II as PCI target.

Table 4: Command Encoding for Transaction Type—PowerSpan II as PCI Target

Px_C/BE#[3:0]	Transaction Type	PowerSpan II Capable
0000	Interrupt Acknowledge	No
0001	Special Cycle	No
0010	I/O Read	No
0011	I/O Write	No
0100	Reserved	N/A
0101	Reserved	N/A
0110	Memory Read	Yes
0111	Memory Write	Yes
1000	Reserved	N/A
1001	Reserved	N/A
1010	Configuration Read	Yes (Type 0 only)
1011	Configuration Write	Yes (Type 0 only)
1100	Memory Read Multiple	Yes
1101	Dual Address Cycle	No
1110	Memory Read Line	Yes
1111	Memory Write and Invalidate	Aliased to Memory Write

The PCI target image decodes and claims PCI transactions and controls how these incoming PCI transactions are mapped to the destination port on PowerSpan II.

Table 5 describes the programming model for a PCI Target Image Control register.

Table 5: Programming Model for PCI Target Image Control Register

Bits	Type	Description	Default Setting
IMG_EN	R/W	Enables the PCI target image to decode in the specified physical address range of memory space.	Disabled
TA_EN	R/W	Enables address translation (see "PCI 1 Target Image x Translation Address Register" on page 328).	Disabled
BAR_EN	R/W	Enables the PCI Base Address register. When this bit is set, the Px_BSTx Register is R/W and visible to Processor Bus access and PCI memory cycles. When this bit is cleared, the Px_BSTx register returns only zeros on a read. Writes will have no effect on Px_BSTx when this bit is cleared.	Enabled or configurable through EEPROM
BS[3:0]	R/W	Sets the block size of the PCI target image. The size of the image is 64Kbyte * 2^{BS}.	Default value is 0, can be programmed through any port after reset or loaded through EEPROM.
MODE	R/W	Maps the incoming PCI transaction to either memory or I/O space on the alternate PCI bus.	Default value is 0 (Memory command generation)
DEST	R/W	Directs the incoming PCI transaction to either the processor bus or the alternate PCI interface.	Defaults to processor bus
MEM_IO	R/W	Commands to the corresponding image generates Memory Read commands on the destination PCI bus (Py) with the same byte enables latched from the source bus transaction PowerSpan II is capable of performing 1,2,3, or 4 byte memory transfers on the PCI bus(es).	Default value is 0 (Regular I/O mode)
RTT[4:0]	R/W	A 5-bit value, defined in the processor bus protocol, is generated on the PB_TT lines during a read on the processor bus.	Defaults to Read
GBL	R/W	Controls the assertion of the PB_GBL_ cache control signal.	Asserts PB_GBL_
CI_	R/W	Controls the assertion of the PB_CI_ cache control signal.	Asserts PB_CI_

Table 5: Programming Model for PCI Target Image Control Register

Bits	Type	Description	Default Setting
WTT[4:0]	R/W	A 5-bit value, defined in the processor bus protocol, is generated on the PB_TT lines during a write on the processor bus.	Defaults to Write with Flush
PRKEEP	R/W	Enables PowerSpan II to keep prefetch read data over subsequent transactions (see "Reads" on page 55).	Disabled
END[1:0]	R/W	Sets endian mapping to little-endian, PowerPC little-endian, or big-endian (see "Endian Mapping" on page 213).	Big-endian is the default mode.
MRA	R/W	Aliases a Memory Read Command to Memory Read Multiple Command. This causes PowerSpan II to prefetch read data on the destination bus (processor bus or PCI) up to the amount programmed in the RD_AMT[2:0] field.	Disabled
RD_AMT[2:0]	R/W	Controls the prefetch read amount for a Memory Read when MRA is enabled. Memory Read Multiple always causes prefetch up to the value in RD_AMT[2:0]. This can be programmed up to a maximum of 128 bytes.	8 bytes is the default prefetch read amount

Master-based Decode

The PCI Target supports Master-based decode when the PowerSpan II PCI arbiter is enabled (see "Arbitration" on page 163). With Master-based decode enabled, a PCI target image only claims a transaction decoded for its specified physical address space if it originates from a specific PCI master.

External bus masters are selected for a specific target image by setting the corresponding bits in the "PCI 1 Target Image x Translation Address Register" on page 328.

PowerSpan II behavior is undefined if more than one overlapping target image claims a transaction. For example, if two target image have the same base address and size, then they must have unique master bits set in the "PCI 1 Target Image x Translation Address Register" on page 328.

2.2.2.2 Address Translation

The address generated on the destination port is dependent on the use of address translation in the source target image. For more information, see the Translation Address Enable (TA_EN) bit in the "PCI 1 Target Image x Control Register" on page 322. When address translation is enabled — by setting the TA_EN bit— the address generated on the destination bus is derived from the following three inputs:

- incoming address on the PCI Target
- block size of the target image
  BS[3:0] in the "PCI 1 Target Image x Control Register" on page 322
- translation address
  TADDR in the "PCI 1 Target Image x Translation Address Register" on page 328

When address translation is disabled the address on the destination bus is the same as the address on the source bus.

2.2.2.3 Transaction Type Mapping

Mapping to the Processor Bus Interface

The PCI Target Image controls the transaction type on the processor bus through the use of the PB Read Transfer Type (RTT[4:0]) and PB Write Transfer Type (WTT[4:0]) bits in the "PCI 1 Target Image x Control Register" on page 322. By default, these bit fields assign reads as read operations on the processor bus, and assign incoming writes as Write with Flush on the Processor Bus.

Mapping to a PCI Interface

The PCI target image determines the address space on the destination PCI bus through the use of the Image Mode (MODE) bit in the "PCI 1 Target Image x Control Register" on page 322. By default, incoming PCI transactions are mapped to Memory Space on the alternate PCI Interface. Setting the MODE bit maps incoming PCI transactions to
I/O Space on the alternate PCI Interface.

2.2.2.4 Address Parity

The PCI target image monitors parity during the address phase of decoded transactions. Address parity errors are reported on Px_SERR# when both the Parity Error Response (PERESP) and SERR Enable (SERR_EN) bits are set in the "PCI 1 Control and Status Register" on page 301. Assertion of the Px_SERR# signal can be disabled by clearing the SERR_EN bit.

PowerSpan II records an error condition in the event of an address parity error (see "Error Handling" on page 187). PowerSpan II claims the errored transaction and forwards the transaction to the destination bus.

2.2.3 Data Phase

2.2.3.1 Writes

PowerSpan II accepts single beat or burst transactions in memory space. I/O accesses are not decoded. All writes to the PCI Target are posted writes.

Burst writes are linear bursts. A Target-Disconnect is issued if a buffer fills while a burst write is in progress (see "Termination Phase" on page 58). PowerSpan II can manage arbitrary PCI byte enable combinations during PCI burst writes.

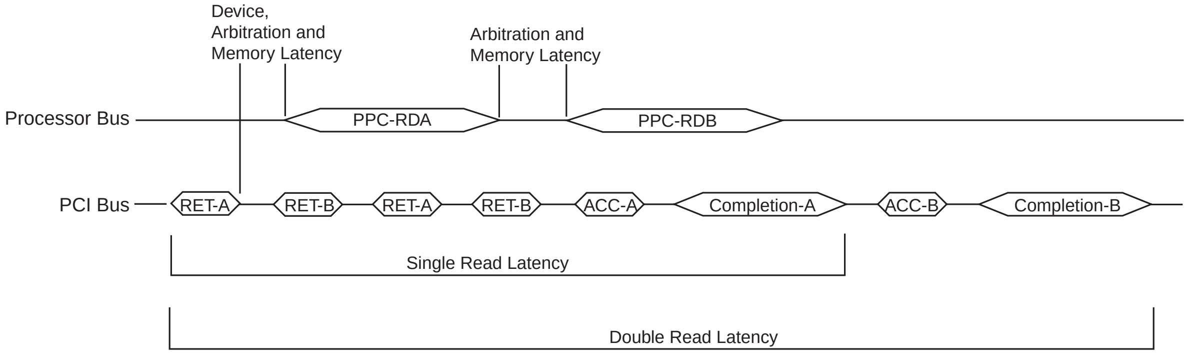
PowerSpan II does not support delayed write transactions as described in the *PCI 2.2 Specification*.

2.2.3.2 Reads

PowerSpan II supports up to four concurrent reads from external PCI masters. All four reads are treated equally and have the same prefetch capacity, but have individually programmable values.

An example of PowerSpan II's concurrent read capability is illustrated in Figure 6. The concurrent reads in the figure are represented by Read A and Read B. In the figure, both Read A and Read B are retried. Once Read A is completed on the processor bus, Read B is initiated on the processor bus while the contents of Read A are returned to the PCI master. Because Read B is completed on the processor bus while Read A data is returned to the PCI bus, read latency is significantly reduced with concurrent reads.

Figure 6: Concurrent Read Waveform



See "Concurrent Reads" on page 38 for a general discussion of read pipelining in PowerSpan II.

Concurrent Read Phases

The delayed, concurrent reads on the PCI Target consist of the following phases:

1. Delayed Read Request

  - The PCI Target latches the transaction parameters and issues a retry.

2. Delayed Read
   - The PCI Target obtains the requested data. The destination bus master retries requested data.
3. Delayed Read Completion
   - The master repeats the transaction with the same parameters used for the initial request and data is provided by PowerSpan II.

Read line buffers are allocated on a first come, first serve basis. When an external master makes the initial memory request, the PowerSpan II PCI Target captures the PCI address in an available delayed read request latch. This initiates a read on the destination bus specified by the Destination Bus (DEST) bit in the “PCI 1 Target Image x Control Register” on page 322.

Prefetch Reads

All PowerSpan II PCI target memory reads are considered prefetchable to 8-byte boundaries by default. Setting the MEM_IO bit in the “PCI 1 Target Image x Control Register” on page 322 enables 1,2,3, or 4 byte memory reads on the PCI bus and 4 byte reads on the processor bus. When PowerSpan II is programmed to support 4 byte transactions, only 4 byte transactions are supported. Burst transactions are not supported while the MEM_IO bit is set to 1.

In order to program PowerSpan II to complete 4 byte reads on the PCI bus, both the MEM_IO bit and the MODE bit must be set to 1 in the PCI *x* Target Image *x* Control register.

In order to perform a 4-byte read from the PCI bus to the processor (60x) bus, the following bits must be programmed:

- MEM_IO bit set to 1
- MODE bit set to 1 or 0
- END bit, in the “PCI 1 Target Image x Control Register” on page 322, must not be set to little-endian mode (00). It can be set to PowerPC little-endian (01), or big-endian (10).

When the Target Image Control register is programmed for 4 byte read transactions, requesting 8 byte reads causes undefined results in the system.

PowerSpan II prefetch behavior on the destination bus when claiming Memory reads on the originating bus is controlled by the PCI Memory Read Alias (MRA) bit and the Prefetch Size (RD_AMT[2:0]) field in the "PCI 1 Target Image x Control Register" on page 322. If the MRA bit is set when PowerSpan II claims a memory read, PowerSpan II prefetches the amount programmed into the RD_AMT[2:0] field — up to 128 bytes.

The Memory Read Line command results in a prefetch of the value programmed into Cache Line (CLINE) bit. When the MRA bit is cleared, the target image prefetches 8 bytes when a PCI Memory Read command is decoded.

The Memory Read Multiple command results in a prefetch read of a minimum of 32 bytes or the value programmed into the RD_AMT[2:0] field— independent of the MRA bit setting.

The PowerSpan II PCI target read watermarks are defined in Table 6.

Table 6: PowerSpan II PCI Target Read Watermarks

PCI Command	Prefetch Amount
Memory Read	8 bytes (default) or 1,2,3, or 4 bytes Depending in the setting in the MEM_IO bit in the PCI Target Image Control Register (see page 322)
Memory Read Line	Minimum of CLINE in the Px_MISC0 register
Memory Read Multiple	Minimum of 32 bytes or RD_AMT.

PowerSpan II never prefetches data beyond a 4-Kbyte address boundary regardless of the value programmed in the RD_AMT field. This boundary corresponds to the processor bus memory management page size.

The PowerSpan II PCI Target can be configured to keep prefetch data over multiple read accesses for any master that provides the correct address — by setting the PRKEEP bit in the "PCI 1 Target Image x Control Register" on page 322. PowerSpan II increments its latched address for the read transaction based on the amount of data removed by the PCI master during the read transaction. If the PCI master returns with an address that matches the incremented address held by PowerSpan II, then PowerSpan II provides data already held in the prefetch line buffer.

Writes do not invalidate read buffer contents.

2.2.3.3 Data Parity

PowerSpan II monitors Px_PAR#/Px_PAR64# when it accepts data as a PCI target during a write. PowerSpan II drives Px_PAR#/Px_PAR64# when it provides data as a PCI target during a read. In both cases, the Px_PAR#/Px_PAR64# signal provides even parity for Px_C/BE#[3:0] and Px_AD[31:0] — or Px_C/BE#[7:4] and Px_AD[63:32] for the PCI-1 Interface in 64-bit mode.

The PERESP bit in the "PCI 1 Control and Status Register" on page 301 determines whether or not PowerSpan II responds to parity errors as a PCI target. Data parity errors are reported through the assertion of Px_PERR# when the PERESP bit is set. The Detected Parity Error (D_PE) bit in the "PCI 1 Control and Status Register" on page 301 is set when PowerSpan II encounters a parity error as a PCI target on any transaction. PowerSpan II records an error condition when a parity error occurs (see "Error Handling" on page 187).

2.2.4 Termination Phase

2.2.4.1 PCI Target Terminations

The PCI Target Interface generates the following terminations:

1. Target-Disconnect (with data): A termination is requested by the PCI target — by asserting Px_STOP# and Px_TRDY# — when it requires a new address phase. Target-Disconnect means the transaction is terminated after one or more valid data transfers.

   The PCI target requests a Target-Disconnect in the following cases:

   - PowerSpan II is unable to buffer an incoming write or provide data from a read buffer during a read.
   - PowerSpan II reaches the 4-Kbyte address boundary on reads and writes to the processor bus.
   - One data phase for PowerSpan II register accesses
   - One data phase for I_2O shell accesses
   - Detection of a transaction with non-linear addressing

2. Target-Retry: A termination is requested — by asserting Px_STOP# and Px_DEVSEL# while Px_TRDY# is high — by the PCI Target because it cannot currently process the transaction. Retry means the transaction is terminated after the address phase without any data transfer. PowerSpan II retries read requests while it fetches data from the destination bus. Any attempt by a PCI master to complete the memory read transaction is retried by the PCI target until at least an 8-byte quantity is available in the line buffer. If a PCI master does not retry the transaction within 2^{15} clocks after a read request has been latched, the delayed read request latch and line buffer are de-allocated. This prevents deadlock conditions.

3. Target-Abort: The PCI target requests a termination of a transaction — by negating Px_DEVSEL# and Px_TRDY# and asserting Px_STOP# on the same clock edge — when it cannot respond to the transaction, or during a fatal error. A fatal error occurs when: a bus error is experienced on the processor bus, the maximum retry count is exceeded, a Target-Abort occurs on the alternate PCI bus during a read, or a Master-Abort occurs on the alternate PCI bus during a read.

   Although there may be a fatal error for the initiating application, the transaction completes gracefully, ensuring normal PCI operation for other PCI resources. PowerSpan II sets the signaled Target-Abort (S_TA) bit in the "PCI 1 Control and Status Register" on page 301, and records an error condition in the event of a Target-Abort (see "Error Handling" on page 187)

Error Logging and Interrupts

The PowerSpan II PCI Target records errors under the following conditions:

- address parity error
- data parity error on writes
- Target-Abort

See "Error Handling" on page 187 and "Interrupt Handling" on page 173 for a full description of error logging support and associated interrupt mapping options.

2.3 PCI Master Interface

2.3.1 PowerSpan II as PCI Master

In order for PowerSpan II to be a PCI master in a transaction the Bus Master (BM) bit, in the "PCI 1 Control and Status Register" on page 301, must be set. With this bit set, PowerSpan II is PCI Master in a transaction in the following instances:

- Servicing a request by:
  - the processor bus: PowerSpan II is accessed as a PB slave
  - the alternate PCI Interface: PowerSpan II is accessed as a PCI target
- processing a transfer by one of the four PowerSpan II DMA channels
- generating a configuration or IACK cycle because of a PowerSpan II register access

This section discusses only the first three conditions listed above. Configuration and IACK cycles are discussed in "Configuration and IACK Cycle Generation" on page 295.

The operation of the PCI Master is described by dividing the PCI master transaction into the following phases:

- **Arbitration Phase**: This section describes how PowerSpan II requests the PCI bus and its response to bus parking.
- **Address Phase**: This section discusses the generation of the PCI address and command encoding.
- **Data Transfer**: This section describes control of burst length.
- **Terminations**: This section explains the terminations supported by PowerSpan II, how they are mapped to the source port (Processor Bus Interface or the alternate PCI Interface), and exception handling.

PowerSpan II cannot be both master and target on a PCI bus at the same time.

2.3.2 Arbitration for PCI Bus

PowerSpan II issues a bus request on the PCI bus when it requires access to the PCI bus. When the PowerSpan II PCI arbiter is active, this request is internal. When it is not enabled the request appears externally (see “PCI Interface Arbitration” on page 163 for more information).

The internal PowerSpan II PCI arbiter parks the bus on a PCI master by asserting Px_GNT# to the PCI master. Bus parking improves the performance of the PowerSpan II PCI Master by reducing arbitration latency.

2.3.3 Address Phase

2.3.3.1 Command Encoding

The encoding on the Px_C/BE# lines indicate the transaction type on the PCI bus. The PCI command encoding supported by PowerSpan II, and their corresponding transaction types, are shown in Table 7.

Table 7: Command Encoding for Transaction Type (PowerSpan II as PCI Master)

Px_C/BE# [3:0]	Transaction Type	PowerSpan II Capable
0000	Interrupt Acknowledge	Yes (see "Configuration and IACK Cycle Generation" on page 295)
0001	Special Cycle	No
0010	I/O Read	Yes
0011	I/O Write	Yes
0100	Reserved	N/A
0101	Reserved	N/A
0110	Memory Read	Yes
0111	Memory Write	Yes
1000	Reserved	N/A
1001	Reserved	N/A
1010	Configuration Read	Yes (see "Configuration and IACK Cycle Generation" on page 295)
1011	Configuration Write	Yes (see "Configuration and IACK Cycle Generation" on page 295)
1100	Memory Read Multiple	Yes
1101	Dual Address Cycle	No
1110	Memory Read Line	Yes
1111	Memory Write and Invalidate (The MWI_EN bit is hard-wired to "0" in the "PCI 1 Control and Status Register" on page 301)	No

A new request for access to the bus is generated by the PowerSpan II PCI Master when it requires access to the PCI bus to service a request from the Processor Bus Interface or the other PCI interface (Py). After the request is generated by PowerSpan II, it successfully arbitrates for access to the PCI bus when it receives GNT_ from the arbiter. PowerSpan II then asserts Px_FRAME# to indicate the beginning of a transaction.

2.3.3.2 Address Translation

The address generated by the PCI Master is dependent on the use of address translation in the source target image (see "PCI 1 Target Image x Control Register" on page 322) or slave image (see "Processor Bus Slave Image x Control Register" on page 342). When address translation is enabled — by setting the TA_EN bit in PCI Target or PB Slave Image Control Register — PowerSpan II produces the PCI address using the following inputs:

- the incoming address from the source bus
- the block size of the slave or target image
- the translation offset

For address translation going from the processor bus to PCI, see "Processor Bus Interface" on page 101. For an example of address translation control going from PCI to PCI, see "PCI 1 Target Image x Translation Address Register" on page 328.

When address translation is disabled, the address generated by the PCI Master is the same as the address on the source bus.

2.3.4 Data Phase

2.3.4.1 Writes

Non-DMA Writes

For non-DMA writes, the length of the PCI write transaction is dependent on the length of the transaction delivered from the source bus. Writes originating from the processor bus can be either single cycle writes or burst writes. Burst writes from the processor bus are always 32 bytes in length. This burst is converted to an 8 byte burst on a 32-bit PCI bus. Either PCI-1 or PCI-2 can be configured as 32-bit. Single cycle writes from the 64-bit processor bus are translated into two 8 byte burst writes on the 32-bit PCI bus. This information is summarized in Table 8

Table 8: PB Writes and Their Corresponding PCI Writes

PB Write	64-bit PCI Write	32-bit PCI Write
32-byte Line Write	4-beat 32-byte Burst Write	8-beat 32-byte Burst Write
8-byte Single Write	Single Beat 8-byte Write	2-beat 8-byte Burst Write

The PB Master can also generate MPC8260 extended cycles. Extended cycles are either 16 byte or 24 byte transactions. These cycles are enabled by setting the Extended Cycle (EXTCYC) bit to 1 in the Processor Bus Miscellaneous control Register (see page 361).

Incoming PCI writes are executed as similar writes on the alternate PCI Interface. For example, a 64-byte burst write to memory space is executed as a 64-byte burst write to the memory space on the alternate bus, provided the target does not disconnect.

Write transactions intended for I/O space on the alternate PCI bus must be single beat writes. Bursting is not supported for a target image programmed to generate an I/O access on the alternate PCI bus. A burst write directed at such a target image results in a Target-Disconnect after every data beat.

DMA Writes

The PowerSpan II DMA channels always attempt to perform the longest possible burst — up to 128-bytes — on the PCI bus.

2.3.4.2 Reads

The minimum memory read prefetch quantity is 8 bytes (default). Setting the MEM_IO bit in the “PCI 1 Target Image x Control Register” on page 322 enables 1,2,3, or 4 byte memory reads on the PCI bus.

The PowerSpan II PCI Master generates a Memory Read command selection according to the rules in Table 9.

Table 9: PowerSpan II PCI Master Read Commands

Internal Request of Transaction Length	PCI Memory Read Command
<= 8 bytes	Memory Read
<= CLINE[7:0] in Px_MISC0	Memory Read Line
> CLINE[7:0] in Px_MISC0	Memory Read Multiple

The read amount presented to the PCI Master determines the command used. A Memory Read Line command uses the burst length programmed into the CLINE[7:0] field in the "PCI 1 Miscellaneous 0 Register" on page 307. It is programmable to 16-, 32-, 64-, or 128 bytes.

If the PCI Master does not complete the burst read transaction before a target termination, it completes the read with subsequent PCI read transactions at the appropriate address.

2.3.4.3 Parity Monitoring and Generation

PowerSpan II monitors Px_PAR#/Px_PAR64# when it accepts data as a PCI master during a read, and drives Px_PAR#/Px_PAR64# when it provides data as a PCI master during a write. PowerSpan II also drives Px_PAR#/Px_PAR64# during the address phase of a transaction when it is a PCI master. In both address and data phases, the Px_PAR#/Px_PAR64# signal provides even parity for Px_C/BE#[3:0] and Px_AD[31:0]. Even parity is enabled Px_C/BE#[7:4] and Px_AD[63:32] for PCI-1 in 64-bit mode.

PowerSpan II parity response is enabled through the Parity Error Response (PERESP) bit in the "PCI 1 Control and Status Register" on page 301. Data parity errors are reported through the assertion of Px_PERR# when the PERESP bit is set. The Detected Parity Error (D_PE) bit in the "PCI 1 Control and Status Register" on page 301 is set when PowerSpan II encounters a parity error as a PCI master on any transaction. PowerSpan II records an error condition in the event of a parity error (see "Error Handling" on page 187).

The Master Data Parity Detected (MDP_D) bit in the "PCI 1 Control and Status Register" on page 301 is set if the PERESP bit is enabled and either PowerSpan II is the master of the transaction where it asserts PERR#, or the addressed target asserts PERR#. If the transfer originated from the Processor Bus Interface, then PowerSpan II sets the MDP_D bit and the Px_PB_ERR_EN bit in the "Interrupt Enable Register 1" on page 395. PowerSpan II then asserts an interrupt (see "Interrupt Handling" on page 173).

PowerSpan II continues with the transaction independent of any parity errors reported during the transaction.

2.3.5 Termination Phase

2.3.5.1 PCI Master Terminations

The PCI Master supports all four types of PCI terminations:

1. Master-Abort: The PCI Master negates Px_FRAME# and then negates Px_IRDY# on the following clock edge when no target responds with Px_DEVSEL# asserted on the fifth positive edge of clock after Px_FRAME# is asserted. PowerSpan II sets R_MA in Px_CSR and records an error condition in the event of a Master-Abort (see "Error Handling" on page 187)
2. Target-Disconnect (with data): A termination is requested by the target — by asserting Px_STOP, Px_DEVSEL# and Px_TRDY# — because it is unable to respond within the latency requirements of the *PCI 2.2 Specification* or it requires a new address phase. Target-Disconnect means the transaction is terminated after data is transferred. PowerSpan II negates Px_REQ# for at least two clock cycles if it receives Px_STOP# from the PCI target.
3. Target-Retry: Termination is requested — by asserting Px_STOP# and Px_DEVSEL# while Px_TRDY# is high — by the target because it cannot currently process the transaction. Retry means the transaction is terminated after the address phase without data transfer. PowerSpan II has a Maximum Retry Counter (MAX_RETRY) in the "PCI 1 Miscellaneous Control and Status Register" on page 337 which is used to record an error condition if the number of retries exceed the programmed amount (see "Error Handling" on page 187).
4. Target-Abort: The target requests a termination of a transaction — by negating Px_DEVSEL# and asserting Px_STOP# on the same clock edge — when it cannot respond to the transaction, or during a fatal error. Although there may be a fatal error for the initiating application, the transaction completes gracefully, ensuring normal PCI operation for other PCI resources. PowerSpan II sets R_TA in Px_CSR and records an error condition in the event of a Target-abort (see "Error Handling" on page 187).

2.3.5.2 Error Logging and Interrupts

The PowerSpan II PCI Master records errors under the following conditions:

- Data Parity on reads (when the PERSP bit, in the "PCI 1 Control and Status Register" on page 301, is set)
- Master-Abort
- Target-Abort
- Expiration of Maximum Retry Counter (when the MAX_RETRY field, in the "PCI 1 Miscellaneous Control and Status Register" on page 337, is set.

See "Error Handling" on page 187 and "Interrupt Handling" on page 173 for a full description of error logging support and associated interrupt mapping options.

2.4 CompactPCI Hot Swap Silicon Support

CompactPCI's Hot Swap Specification defines the process for installing and removing adapter boards without adversely affecting a running system. It provides a programmatic access to Hot Swap services. This enables system re-configuration and fault recovery to take place with no system down time and minimum operator interaction.

PowerSpan II is compliant with the *CompactPCI's Hot Swap Specification, Revision 2.0* and is a Hot Swap Silicon device. Hot Swap Silicon support includes the following:

- Open drain output pin ENUM# is used to indicate Hot Swap insertion and extraction events.
- 5V tolerant input pin ES for sensing the state of the ejector switch used to insert or extract a CompactPCI board.
- 5V tolerant open drain output pin LED# for controlling the blue Light Emitting (LED) required to indicate status of the software connection process.

For the different levels of Hot Swap support, refer to the *CompactPCI Hot Swap Specification*.

To simplify the design of CompactPCI Hot Swap adapter cards, PowerSpan II has additional support. This support includes:

- A 5V tolerant input pin HEALTHY# for sensing the status of the Back End power on the card.
- An input pin P1_64EN# that enables Hot Swap adapter cards to sense the presence of a 64-bit PCI backplane.

2.4.1 LED Support

The LED can be controlled by hardware and software. PowerSpan II drives the LED# signal low to turn on the LED during the Physical and Hardware Connection process (when HEALTHY# is negated). A blue LED with an internal resistor can be directly connected between the 5V rail and the LED# signal. Software controls the LED by setting the LED On/Off (LOO) bit in the "PCI 1 Compact PCI Hot Swap Control and Status Register" on page 317.

2.4.2 ES Input

The *CompactPCI Hot Swap Specification* defines a switch located in the ejector handle that indicates to PowerSpan II if the ejector handle is open or closed. A low value on ES input indicates that the ejector latch is open. A high value on ES indicates that the ejector latch is closed and is in operation mode.

2.4.3 HEALTHY# Signal

PowerSpan II manages the electrical board level issues involved in the Hot Swap process with the HEALTHY# signal. The negation of HEALTHY# indicates only some of the components on the Hot Swap card are powered. To operate in this environment and minimize long term reliability issues, the HEALTHY# signal controls the electrical behavior of PowerSpan II I/O buffers.

During the negation of HEALTHY#, PowerSpan II disables its output and bidirectional pins (except for LED#) to avoid applying power to non-powered components on the card. The signals connected between PowerSpan II and these non-powered components result in floating pins on PowerSpan II. PowerSpan II uses HEALTHY# to inhibit the input receivers. An inhibited receiver has no static current path between supply and ground that could be activated by a voltage level near the switching point.

See “Resets, Clocks and Power-up Options” on page 201 for more details on HEALTHY# and PowerSpan II reset.

2.4.4 CompactPCI Hot Swap Card Insertion and Extraction

A CompactPCI board has a staggered pin arrangement (long/medium/short) to allow power and ground, signal and a Board Inserted Indicator (BD_SEL#) to be connected and disconnected in stages. A limited number of power and ground pins are long. The rest of the power, ground, and signal pins are of medium length. BD_SEL# is a short pin. When BD_SEL# connects, the physical connection process is complete.

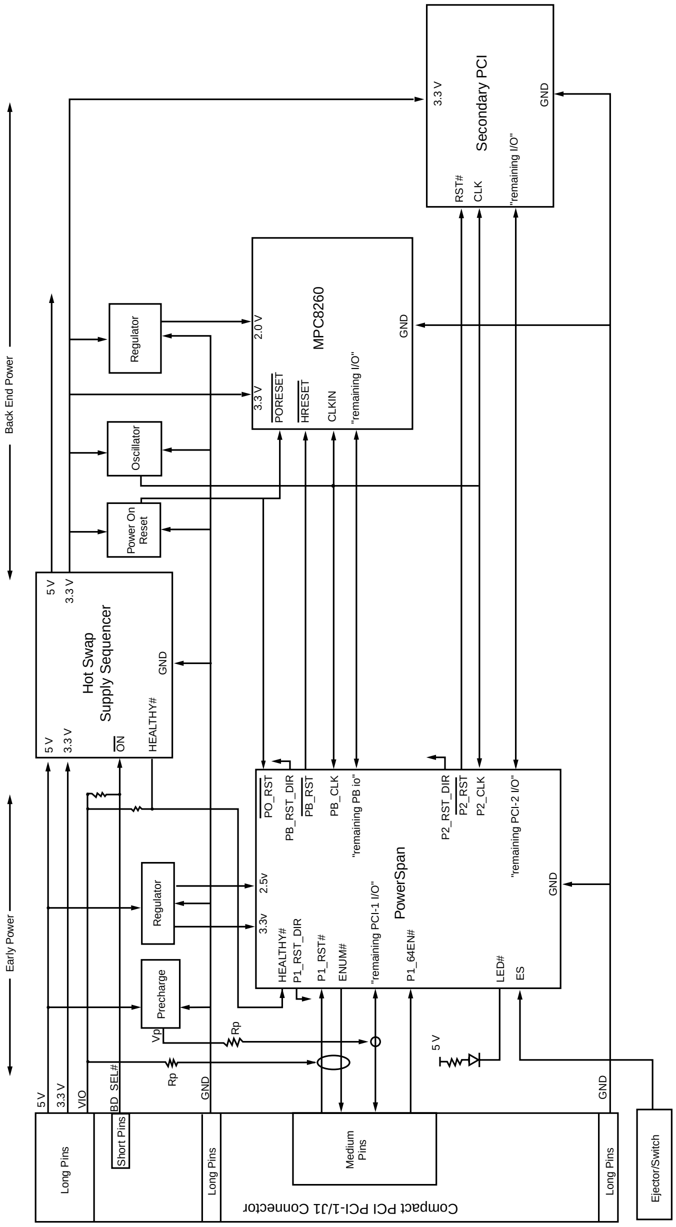
2.4.4.1 CompactPCI Hot Swap Process

A CompactPCI Hot Swap board is divided into two power regions: Early Power and Back End Power. Early Power is provided by the long pins on the CompactPCI connector. Back End Power is controlled by a sequencer on the card. The sequencer begins to power the Back End of the card when the short CompactPCI signal BD_SEL# engages on insertion, or when host software enables the process as in a High Availability system.

In Figure 7, PowerSpan II is designed into a CompactPCI adapter card.

Figure 7 assumes the CompactPCI backplane is not in reset during the insertion and extraction process. For example, PowerSpan II’s P1_RST# is negated.

Figure 7: PowerSpan II in a CompactPCI Adapter Card



Ensure that PB_CLK and P2_CLK are within specification before the release of Back End power-up reset.

2.4.5 Hot Swap Insertion Process

Use the application illustrated in Figure 7 as a point of reference in the Hot Swap insertion process outlined below.

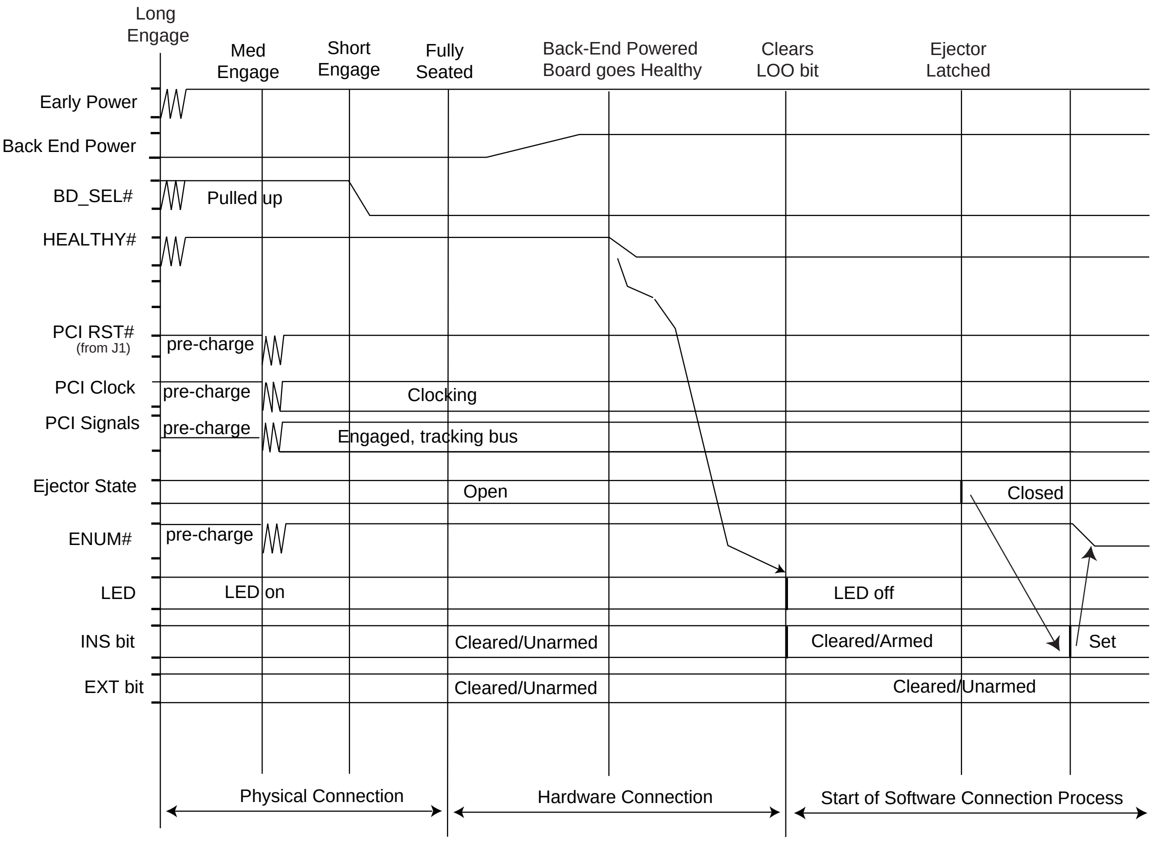
1. Long pins contact for Early Power:
   - HEALTHY# negated
     - PowerSpan II resources are in reset
     - LED# pin enabled, status diode turned on
     - PowerSpan II output pins disabled, input pins inhibited
   - Card’s PCI signals pre-charge
2. Medium pins contact PCI backplane signals:
   - PowerSpan II’s Primary PCI Interface, in this case PCI-1, connects to the PCI pins on the backplane
   - PowerSpan II P1_CLK is within specification
3. Short pins contact, BD_SEL# asserted:
   - Back End Power ramps
   - Back End Power-up reset asserted
     - PowerSpan II PO_RST_ asserted
     - MPC8260 PORESET_ asserted, MPC8260 asserts HRESET_
   - Clock generator begins oscillation
     - PowerSpan II PB_CLK and P2_CLK begin to oscillate
   - Ejector switch closes sometime after short pins contact
4. Back End power is within specification:
   - HEALTHY# asserted
     - LED# pin disabled
     - PowerSpan II outputs enabled, PB_RST_ and P2_RST# asserted
   - MPC8260, Secondary PCI clocks are within specification
5. Back End Power-up reset negation:
   - PowerSpan II PLLs released from reset and begin to lock on to P1_CLK, PB_CLK, P2_CLK

- MPC8260 completes its configuration master transactions
  - PowerSpan II power-up options are loaded
- MPC8260 HRESET_ times out
- PowerSpan II PLL locking complete
  - All PowerSpan II resources out of reset, PB_RST_ and P2_RST# negated
  - PowerSpan II executes EEPROM load or waits to be initialized by the processor

6. PowerSpan II waits for the closed ejector switch and responds by:
   - Setting INS bit in the HS_CSR register
   - Asserting ENUM#
7. PowerSpan II is now able to accept Configuration cycles on PCI-1 from the CompactPCI Host

Since Px_LOCKOUT bit in the PCI-1 Miscellaneous Control and Status (MISC_CSR) register (see page 337) defaults to 1, PowerSpan II retries the Host Configuration accesses on the PCI-1 Interface until Px_LOCKOUT is cleared. The Host then negates ENUM# by clearing the Insertion (INS) bit in the PCI-1 CompactPCI Hot Swap Control and Status (P1_HS_CSR) register (see page 317) and configures the card. The Px_LOCKOUT bit is cleared by an EEPROM load or by access from the Processor Bus Interface. It is automatically cleared by PowerSpan II when the PWRUP_BOOT option is set to PCI.

Figure 8: Hot Swap Insertion



2.4.6 Hot Swap Extraction Process

Use the application illustrated in Figure 7 as a point of reference in the Hot Swap extraction process outlined below.

1. Ejector switch opens

   — The Extraction (EXT) bit in the P1_HS_CSR register is set, causing the assertion of ENUM#

2. Compact PCI Host:

   — Reads the P1_HS_CSR of each agent to determine which card is being extracted

   — Clears the PowerSpan II EXT bit. This causes the negation of ENUM# and arms the INS bit

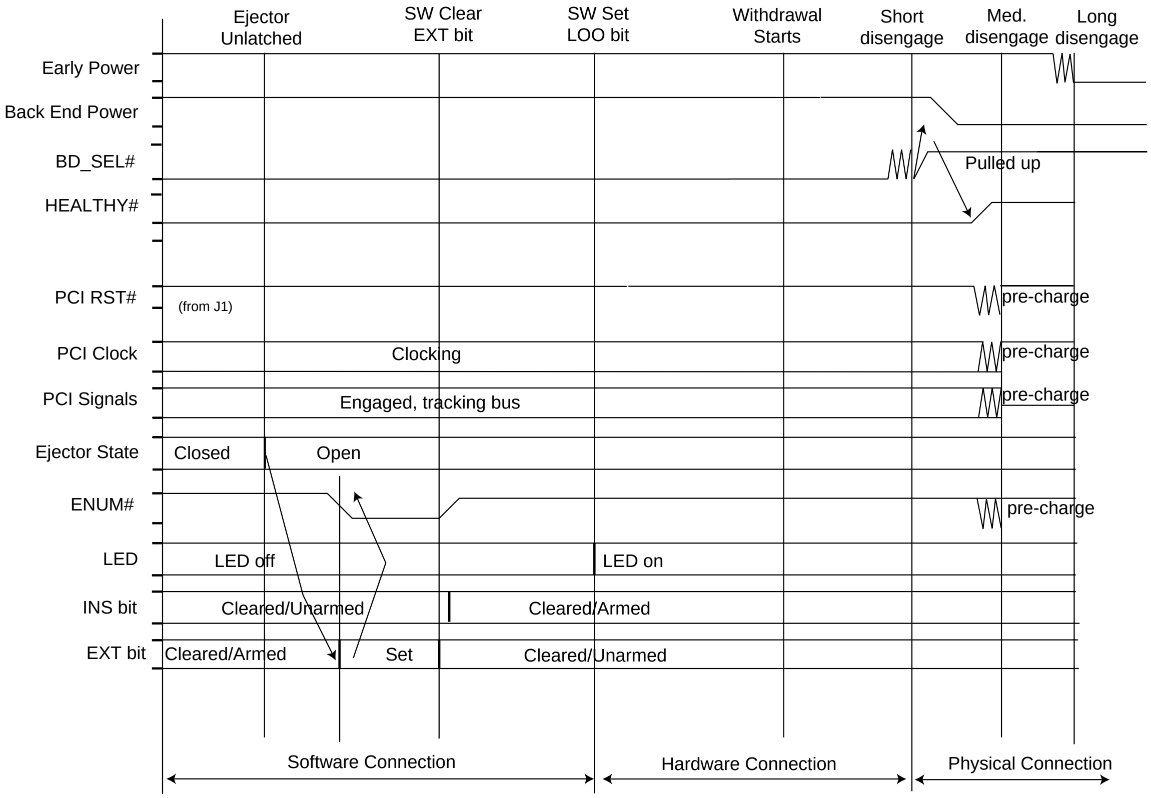
   — Places the card in a software dormant state

- Sets the LED On/Off (LOO) bit in the P1_HS_CSR register. This causes the assertion of LED# which turns the light emitting diode to signal the operator

At this point the operator can close the ejector switch and reenter the insertion process.

3. Operator begins extracting the card.
4. Short pins break, BD_SEL# is negated
   - Back End power goes out of specification
   - HEALTHY# negated
     - All PowerSpan II resources reset
     - LED# pin enabled, status diode turned on
     - PowerSpan II output pins disabled, input pins inhibited
   - PCI pre-charge reapplied
5. Medium pins break.
6. Long pins break.

Figure 9: Hot Swap Extraction



After the status LED# is illuminated by the host, the operator can close the ejector switch, rather than extracting the card. If the closure or the extraction occurs, a PowerSpan II register reload from EEPROM does not occur.

2.5 Vital Product Data

Vital Product Data (VPD) is information which uniquely defines items of a system. These items include the hardware, software and microcode elements of a system. VPD also provides a mechanism for storing information, such as performance data on a device. VPD resides in a local storage device.

PowerSpan II supports VPD through the I^2C Interface to serial EEPROM. The Vital Product Data Extended Capabilities Pointer and supporting registers reside in the configuration space of the PowerSpan II Primary PCI Interface. The VPD feature requires the VPD_EN bit, in the "Miscellaneous Control and Status Register" on page 378, to be set and an available external EEPROM.

2.5.1 VPD Access

VPD accesses through PowerSpan II default to the I^2C serial EEPROM device zero (VPD EEPROM Chip Select (VPD_CS) = 0b000 in the "Miscellaneous Control and Status Register" on page 378). This is also used for EEPROM loading of the registers after reset. Since the lower bytes in the EEPROM contain data for setting up PowerSpan II before software initialization, the lower portion of the EEPROM (the first 64 bytes) are not visible through the VPD registers. The upper 192 bytes of the 256 byte EEPROM are visible through the VPD registers. Of these bytes, the first 64 bytes are VPD-Read Only and the remaining 128 bytes are VPD-Read/Write. When VPD_CS = 0b000, VPD addresses are translated upward by 64 bytes before being presented to the EEPROM.

PowerSpan II can be programmed with an alternate chip select for VPD access if more than the 192 accessible bytes is required. Programming of the I^2C chip select is done in the PowerSpan II "Miscellaneous Control and Status Register" on page 378. If an alternate I^2C chip select is used then the first 64 bytes of the VPD EEPROM is designated as VPD-Read Only and the upper 192 bytes are designated as VPD-Read/Write.

The VPD access to the EEPROM is similar to the EEPROM access implemented in PowerSpan II through the I2C_CSR register, except that it uses the "PCI 1 Vital Product Data Capability Register" on page 319 and the "PCI 1 Vital Product Data Register" on page 321. Since they both access the same resource, a PowerSpan II semaphore register SEMAx must be used to acquire exclusive access of the I^2C Interface before software initiates VPD accesses.

2.5.2 Reading VPD Data

PowerSpan II implements 8-bits of address for accessing the EEPROM up to a maximum of 256 bytes. The VPD address must be DWORD-aligned. A single read access reads four consecutive bytes starting from the VPD address from the EEPROM. If I^{2}C chip select zero is used for VPD, then 192 bytes (address 0x00-BF) of VPD are accessible through the VPD read. Using another I^{2}C chip select, the VPD Read can access the entire 256-byte EEPROM address range.

During a read access, the VPD Address (VPDA) field and the VPD Flag (F) bit are written in the "PCI 1 Vital Product Data Capability Register" on page 319. The F bit must be set to 0 to indicate a VPD read access. PowerSpan II sets the F bit to 1 when it completes reading the 4 bytes from the EEPROM. The F bit must be polled to determine when the read is complete. Byte 0 (bits 7 through 0) of the "PCI 1 Vital Product Data Register" on page 321 contains the data referenced by the VPD Address — bytes 1 through 3 contain the successive bytes.

If the Px_VPDD register or the I2C_CSR register is written to prior to the flag bit being set to 1, the results of the original read operation are unpredictable.

2.5.3 Writing VPD Data

A write can only occur to the upper 128 bytes of the EEPROM or, potentially, the upper 192 bytes if I^{2}C chip select is non-zero. Similar to the read operation, the write operation always writes four consecutive bytes starting from the VPD address to the EEPROM.

The "PCI 1 Vital Product Data Register" on page 321 is written with the 4 bytes of data. Byte 0 (register bits 7 - 0) contains the data to be written to the location referenced by the VPD Address. Bytes 1-3 contain the data for the successive bytes. The VPDA field and the F bit is then written. The F bit must be set to 1 to indicate a VPD write. The F bit is polled to determine when the write to the EEPROM is completed. PowerSpan II sets the F bit to 0 when the write is completed.

When a write is attempted to the lower 64 bytes of the VPD area of the EEPROM, PowerSpan II does not perform the write operation and clears the F bit.

The Px_VPDD or I2C_CSR register must not be written while a write operation is occurring.

2.6 I_2O Shell Interface

PowerSpan II provides portions of the I_2O Shell Interface for the platform it is connecting to the Primary PCI bus. The I_2O Shell Interface defined in the *I_2O 2.0 Specification* is comprised of three main sections:

- messaging interface
- protocol for exchanging messages
- executive class messages

PowerSpan II implements the I_2O messaging interface and, in conjunction with the Input/Output Processor (IOP), enables the message passing protocol.

2.6.1 I_2O Target Image

There are three registers which enable Memory access to the I_2O Shell Interface and local IOP Message Frames. The supporting registers include the following:

- "PCI 1 I2O Target Image Base Address Register" on page 309
- "PCI I2O Target Image Control Register" on page 415
- "PCI I2O Target Image Translation Address Register" on page 420

The I_2O target image does not support Master-Based Decode.

The I_2O Shell Interface consists of Inbound and Outbound Queues and supporting I_2O Host interrupt registers. The queues contain Message Frame Addresses (MFAs). These MFAs specify the starting address of Message Frames relative to the base address of the memory window in PowerPC memory. PowerSpan II implements I_2O support with the first Memory Base Address Register in PCI configuration space.

The I_2O target image is divided into an I_2O Shell Interface and a processor bus memory window intended for IOP Message Frame accesses. The I_2O Shell Interface is accessed through the lower 4 Kbytes of the I_2O target image. I_2O Shell Interface accesses are limited to 32-bit single data phase PCI transactions. Accesses through the I_2O target image memory window to IOP Message Frames are burstable up to 64-bits wide for PCI-1, but limited to 32-bit wide for PCI-2.

PowerSpan II does not support posting of more than one write transaction to the Inbound or Outbound Queue. Attempts to write to the Inbound or Outbound Queue are retried until the currently active write completes on the Processor Bus Interface.

2.6.2 IOP Functionality

A number of configuration steps are required before PowerSpan II and the embedded processor bus are enabled to provide IOP functionality. The following example assumes PCI-1 is the Primary PCI Interface.

The steps required to implement IOP functionality are listed below.

1. In order to identify PowerSpan II as an I_2O Controller the “PCI 1 Class Register” on page 305 must be programmed as follows:
   - Base Class Code (BASE) = 0x0E
   - Sub Class Code (SUB) = 0x00
   - Programming Interface (PROG) = 0x01

   Programming values other than the ones listed above do not affect the behavior of PowerSpan II as an I2O device.

2. The Inbound and Outbound Queues’ location and size in IOP memory must be programmed in PowerSpan II. This is accomplished by programming the “I2O Queue Base Address Register” on page 423:
   - Processor Bus I_2O Base Address (PB_I2O_BS): specifies base address of the Queues
   - FIFO Size (FIFO_SIZE): specifies the size of the Queues

3. The PCI I_2O target image must be configured to claim I_2O Shell and Message Frame accesses from PCI. The following registers must be programmed:
   - Configure I_2O image size with the Block size (BS) bit in “PCI I2O Target Image Control Register” on page 415 (PCI_TI2O_CTL).
   - Enable Base Address Register (BAR) visibility in configuration space.
     - Set BAR_EN in the PCI_TI2O_CTL register.
   - Program PCI Base Address Register “PCI 1 I2O Target Image Base Address Register” on page 309.
     - Set Image Enable (IMG_EN) in “PCI I2O Target Image Control Register” on page 415 to enable decode. Note that this occurs if a non-zero value is written to the PCI Base Address Register.
   - Configure Processor Bus Master transaction parameters.
     - Write Transfer Type (WTT) in the PCI_TI2O_CTL register.
     - Read Transfer Type (RTT) in the PCI_TI2O_CTL register.

    - Global Command (GBL) in the PCI_TI2O_CTL register.
    - Cache Inhibit (CI) in the PCI_TI2O_CTL register.
  - Select endian conversion mechanism with the Endian Conversion (END) bit in the PCI_TI2O_CTL register
  - Configure address translation
    - Translation Address Enable (TA_EN) bit in the PCI_TI2O_CTL register.
    - Translation Address (TADDR) in the "PCI I2O Target Image Translation Address Register" on page 420 (PCI_TI2O_TADDR)
  - Enable decode in PCI memory space.
    - Set memory Space (MS) bit in the "PCI 1 Control and Status Register" on page 301.

  At this stage, the I_2O image is defined but all accesses to the PCI I_2O target image are retried.

4. The IOP is required to initialize all Top and Bottom Pointer registers and initialize all the MFAs in the Inbound Free List FIFO. At this point, the IOP enables PCI accesses with the following step:
  - Set the I2O Enable (I2O_EN) bit in the "I2O Control and Status Register" on page 421.

2.6.3 Messaging Interface

The *I_2O 2.0 Specification* defines a mechanism for connecting an I/O Platform (IOP) to an I_2O system through a memory-based system, such as PCI, which has no inherent message passing capability. An IOP which is connected to a memory-based system is said to be locally attached. The PowerSpan II implements four I_2O defined memory mapped registers on PCI to enable the physical and logical connection of the IOP to the system. Two of these memory-mapped registers provide the interface for the external Host platform and other IOPs to exchange messages with the local IOP sitting behind the PowerSpan II. These two registers are the Inbound Queue and Outbound Queue interfaces. The other two registers are used as I_2O specific Interrupt Status and Enable registers for the local IOP to signal the Host platform. Additional PowerSpan II specific registers are implemented to support the messaging interface.

2.6.3.1 Inbound Queue

The I_2O Inbound Queue register is the messaging interface used by the Host or external IOP to post messages to the local IOP.

The I$_2$0 Inbound Queue Register Interface is located at offset 0x040 of the PowerSpan II PCI I$_2$0 target image in PCI Memory space. The Inbound Queue has a Free List FIFO and a Post List FIFO, both of which reside in the IOP local memory.

2.6.3.2 Inbound Free List FIFO and Post List FIFO

The Free List contains the Message Frame Address (MFAs) of Message Frames (MFs) in the IOP's local memory, which are available to the Host or other IOPs for writing inbound messages.

The Post List contains the MFAs of MFs in the local IOPs memory which contain inbound messages for the IOP to process.

The Inbound MFAs are 32-bit offsets from the translated PowerSpan II I$_2$0 target image window base address in local IOP memory. When the Host platform or an external IOP wishes to send a message to the local IOP it must first obtain an MFA from the Inbound Free List. The external platform is then free to place a message in the associated MF. The MFA is then placed into the Inbound Post List for the Local IOP to process.

All I$_2$O Inbound Queue MFAs must be offsets of greater than 4 Kbytes.

2.6.3.3 Outbound Queue

The Outbound Queue Register is the messaging interface used by the local IOP to post messages to the Host. The I$_2$O Outbound Queue Register Interface is located at offset 0x044 of the PowerSpan II PCI I$_2$O target image in PCI memory space. The Outbound Queue has a Free List FIFO and a Post List FIFO, both of which reside in the IOP local memory. The Free List contains the Message Frame Address (MFAs) of Message Frames (MFs) in the Host system memory, which are available to the local IOP for writing outbound messages. Outbound MFAs are absolute addresses of a Message Frame in Host memory. The Post List contains the MFAs of MFs in the Host system memory which contain outbound messages for the Host to process. When the local IOP wishes to send a message to the Host platform it must first obtain an MFA from the Outbound Free List. The local IOP is then free to place a message in the associated MF. The MFA is then placed into the Outbound Post List for the Host to process. All Outbound messages are targeted for the Host platform. If the local IOP wishes to send a message to another IOP (peer-to-peer communication) it uses the external IOPs Inbound Queue to post the Message.

2.6.3.4 Protocol for Exchanging Messages

PowerSpan II I_2O Registers

The PowerSpan II PCI I_2O Shell Interface implements the following I_2O defined registers:

- I_2O Outbound Post List Interrupt Status Register
- I_2O Outbound Post List Interrupt Mask Register
- I_2O Inbound Queue
- I_2O Outbound Queue
- I_2O Host Outbound Index Register (used for Outbound Option)

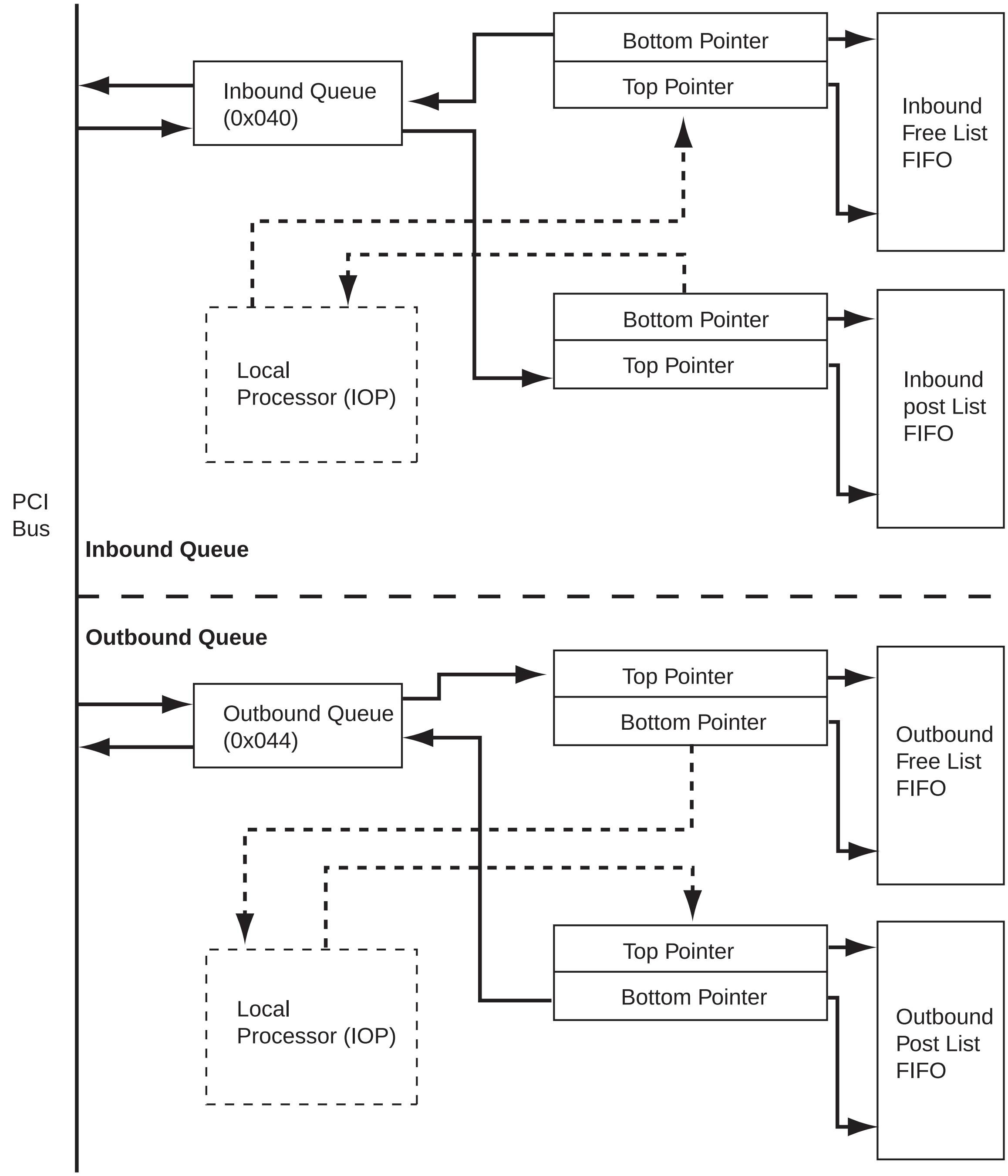
In addition to the registers defined in the *I_2O 2.0 Specification*, PowerSpan II implements a number of registers to support the I_2O message passing protocol of the Shell Interface.

- PCI I_2O Target Image Control Register (PCI_TI2O_ CTL)
- PCI I_2O Target Image Translation Address Register (PCI_TI2O_ TADDR)
- I_2O Queue Base Address Register (I20_ QUEUE_BS)
  - Processor Bus I_2O Base Address Field: Base Address of the block of IOP memory that contains the four FIFOs (two Inbound and two Outbound). The Base Address alignment is 1 Mbyte.
  - FIFO Size Field: Indicates the number of bytes required for each of the Inbound Queue and Outbound Queue FIFOs implemented in local memory
- I_2O Control and Status Register (I20_ CSR)
  - Host Outbound Post List Size Field: Indicates the number of entries in the Host Outbound Post List FIFO in Host memory, used for the Outbound Option.
  - I_2O Enable Field: Enables/Disables PowerSpan II I_2O Interface
    - PowerSpan II Primary PCI target retries I_2O accesses until enabled
  - XI_2O Enable Field**:** Enables/Disables PowerSpan II Outbound Option
    - IPL: Inbound Post List is set when the Inbound Post list FIFO is not empty
    - OFL: Outbound Free List is set when the Outbound Free List FIFO is not empty.

- Inbound Free List Bottom/Top/Top Increment Pointer Registers: (IFL_BOT/IFL_TOP/IFL_TOP_INC)
    - — Manages the Inbound Free List circular FIFO implemented in local memory
- Inbound Post List Bottom/Bottom Increment/Top Pointer Registers: (IPL_BOT/IPL_BOT_INC/IPL_TOP)
    - — Used to manage the Inbound Post List circular FIFO implemented in local memory
- Outbound Free List Bottom/Bottom Increment/Top Pointer Registers: (OFL_BOT/IPL_BOT_INC/OFL_TOP)
    - — Used to manage the Outbound Free List circular FIFO implemented in local memory
- Outbound Post List Bottom/Top/Top Increment Pointer Registers: (IPL_BOT/IPL_TOP_INC/IPL_TOP)
    - — Used to manage the Outbound Post List circular FIFO implemented in local memory
- IOP Outbound Index/Increment Registers: (IOP_OI/IOP_OI_INC)
    - — Used to manage the Host Outbound FIFO
- Host Outbound Index/Index Alias Registers: (HOST_OI/HOST_OIA)
    - — Used to manage the Host Outbound FIFO
- Host Outbound Index Offset Registers: (HOST_OIO)
    - — Determines offset of the I_2O target image at which the Host Processor can access the I_2O Host Outbound Index Register

Interactions between the IOP and Host platforms during the I_2O message passing protocols are displayed in Figure 3.4. The solid lines indicate pointers which are maintained and incremented by the PowerSpan II. The dashed lines indicate pointers which are incremented by the IOP. The IOP writes one to increment to PowerSpan II increment register associated with the pointer.

Figure 10: PowerSpan II I_2O Message Passing



The Top and Bottom pointers manage external FIFOs to determine the full and/or empty status of the FIFOs. After a FIFO write, the Top pointer is incremented. If the Top pointer then equals the Bottom pointer, a FIFO full condition exists. After a FIFO read, the Bottom pointer is incremented. If the Bottom pointer then equals the Top pointer, a FIFO empty condition exists.

2.6.4 Inbound Messages

The Inbound Free and Post List FIFOs are implemented as circular queues using Bottom and Top pointers. The PowerSpan II implements the Bottom and Top pointers for the Inbound Free List FIFO and the Inbound Post List FIFO. The FIFOs reside in the local memory. The Inbound posted messages also reside in local processor memory.

When the Host platform or external IOP wants to post a message to the local IOP, it must first acquire an MFA from the Inbound Free List. This is accomplished through a PCI read transaction to the PowerSpan II Inbound Queue Register at offset 0x040 of the I_2O target image. PowerSpan II provides the next available MFA from the Inbound Free List FIFO pointed to by the Inbound Free List Bottom Pointer Register. PowerSpan II increments the Inbound Free List Bottom Pointer Register to point to the next entry of the FIFO. A read from the Inbound Queue Register when the Inbound Free List FIFO is empty (Bottom Pointer equal to Top pointer) returns 0xffff_ffff to the requesting PCI master.

Once the Host or external IOP obtains an MFA, it is then to write a message to the IOP's local MF at the address offset from the Px_BSI2O specified by the MFA. Once the message is transferred the Host or external IOP writes the MFA back to the same I_2O target image offset (0x040). PowerSpan II accepts the write transaction on PCI and generate a write to the Inbound Post List FIFO at the local IOP memory address pointed to by the Inbound Post List Top Pointer Register. PowerSpan II then increments the Inbound Post List Top Pointer Register and asserts the I2O_IOP Interrupt Status bit in the ISR0 register to notify the local processor of MFAs in the Inbound Post list FIFO. The IPL bit in the "I2O Control and Status Register" on page 421 is set while the Inbound Post List FIFO is not empty, indicating that Inbound Message Frames need to be processed.

2.6.4.1 Local Processor Functions

For Inbound Messaging, the local processor performs the following:

- detects the interrupt
- reads the PowerSpan II ISR0 Register
- determines the source of the interrupt through the I2O_IOP register
- clears the I2O_IOP interrupt (write 1 to clear)
- reads the Inbound Post List FIFO Bottom Pointer Register to access the Inbound Post List FIFO to get the MFA
- increments the Inbound Post List Bottom Pointer Register by writing the Inbound Post List Bottom Pointer Increment Register
- reads and processes the MF pointed to by the MFA

- writes the MFA back to the Top of the Inbound Free List FIFO
- writes to the PowerSpan II's Inbound Free List Top Pointer Increment Register to increment the address by four
- reads the IPL bit, in the I2O_CSR, to determine if the Inbound Post List is empty

The interrupt can be masked, leaving it to the processor to poll the ISR Register. A read from the Inbound Post List Bottom Pointer Register by the IOP when the Inbound Post List FIFO is empty returns 0xffff_ffff to the processor if the EMTR field of the I2O_CSR register is set to one.

2.6.5 Outbound Messages

The Outbound Free and Post List FIFOs are implemented as circular queues using Bottom and Top pointers. PowerSpan II implements the Bottom and Top pointers for the Outbound Free List FIFO and the Outbound Post List FIFO.

When the local IOP wants to post a message to the Host, it must first acquire an MFA from the Outbound Free List. The IOP reads the MFA pointed to by the Outbound Free List Bottom Pointer Register. The processor then increments the Outbound Free List Bottom Pointer Register by four to point to the next entry of the FIFO.

The IOP, having obtained a Host MFA, is then free to write a message through the PowerSpan II to the Host MF at the Host memory address specified by the MFA. Once the message is transferred, the IOP writes the MFA to the Outbound Post List FIFO at the address pointed to by the Outbound Post List Top pointer maintained by PowerSpan II. The processor then increments the Outbound Post List Top Pointer Register by four.

While the Outbound Post List FIFO is non-empty PowerSpan II sets an interrupt status bit in the PowerSpan II I_2O Outbound Post List Interrupt Status Register of the I_2O target image (0x030). If the Interrupt is not masked by the PowerSpan II Outbound Post List Interrupt Mask Register of the I_2O target image (0x034), PowerSpan II drives an interrupt to notify the Host processor of MFAs in the Outbound Post list FIFO. PowerSpan II determines the Outbound Post List FIFO to be non-empty when the Outbound Post List FIFO Bottom and Top pointers do not point to the same FIFO address.

2.6.5.1 Host Processor Functions

For Outbound Messaging, the host processor performs the following:

- detects the interrupt.
- reads the I_2O Outbound Post List Interrupt Status Register (0x030).

- reads the Outbound Queue Register at offset 0x044 of the PowerSpan II I_2O target image map to obtain the next Outbound Post List MFA.
- processes the Message pointed to by the MFA.

The Outbound Interrupt Status and Mask bits are aliased in I2O_HOST in the "Interrupt Status Register 0" on page 388 and I2O_HOST_MASK in the "Interrupt Status Register 0" on page 388. The IOP must program I2O_HOST_MAP in the "Interrupt Map Register Miscellaneous" on page 409 in order for the Outbound Interrupt to be routed to PowerSpan II's Primary PCI interrupt pin.

2.6.5.2 Outbound Message Frame Addresses (MFA)

PowerSpan II provides the MFA at the Bottom of the Outbound Post List FIFO by performing a delayed read from the processor bus. The PowerSpan II increments the Outbound Post List Bottom Pointer Register and compares the value with the Outbound Post List Top Pointer to determine if the Outbound Post List FIFO is empty. When the Bottom and Top pointers contain the same value the I_2O Outbound Post Queue Interrupt Status bit is cleared by the PowerSpan II. Alternatively, the interrupt can be masked out, leaving it to the Host processor to poll the Outbound Queue Register. When the Outbound Post List FIFO is empty, the PowerSpan II returns 0xffff_ffff when the Host processor reads the Outbound Queue Register.

Once the Host processor has processed the MFA, it writes the MFA back to the Outbound Queue Register (0x044) to place it back in the Outbound Free List FIFO. PowerSpan II accepts the write transaction and takes responsibility for replacing the MFA in the Outbound Free List FIFO at the address pointed to by the Outbound Free List Top Pointer Register. PowerSpan II then increments the Outbound Free List Top Pointer Register by four.

2.6.6 Pull Capability

The *I_2O 2.0 Specification* defines an enhancement that allows the IOP to provide a capability to pull the I_2O Inbound messages from the Host memory. In this configuration the Host places the I_2O Inbound messages in MFs located in the Host memory as opposed to the IOP local memory. The Host must also implement a Host Free List FIFO in Host memory. This FIFO does not replace the IOP Inbound Free List FIFO, which must still be implemented in the IOP local memory to support normal inbound message passing, or peer to-peer message passing. This capability increases server performance by virtue of the Host CPU and server platforms being optimized for memory access rather than I/O access. Under this option, the Host can post Inbound messages to the IOP with a single write to the IOP. The IOP pulls the MF from the Host memory and releases a MF to the Host by generating a single write to Host memory. The Pull capability applies only to the IOP's Inbound Queue and to the posting of messages by the Host.

The Pull model requires 16 byte alignment of the message frames, therefore, the least significant four bits of the MFA are always zero. The Pull options use these four bits to create an Extended MFA (XMFA). The Pull model uses the least significant bit of the XMFA to indicate a pull request. This bit is the Pull Indicator or the P bit. Bits 3:1 of the XMFA indicate the number of data transfers required to copy the message. This number is system specific and has no effect on the PowerSpan II's behavior.

To prevent overflow of the local Inbound Post List FIFO the IOP reports an Inbound Post List Headroom to the Host, which is the difference between the size of the Inbound Post List FIFO and the total number of IOP Inbound Message Frames allocated by the IOP in local memory. This is the number of XMFAs the Host can Post to the I_2O Inbound Post List FIFO and guarantee not to overflow the Inbound Post List FIFO. The IOP must not allocate more MFAs in the Inbound Free List than can be accepted in the Inbound Post List (along with XMFAs) without causing overflow.

Equation

#Inbound Free MFAs + #XMFAs <= FIFO_SIZE (see "I2O Queue Base Address Register" on page 423 for more information)

2.6.6.1 Host Posting

The Host can post a message to the IOP using the Pull Capability by using the following methods:

- reading an XMFA from the Host Free List FIFO
- writing an Inbound Message to the MF in Host memory indicated by the XMFA
- writing an XMFA to the Inbound Queue Register at offset 0x040 of the I_2O target image on PCI

The XMFA is processed by the PowerSpan II in the same way as a normal MFA posted to the local IOP by the Host or external IOP. The IOP can determine if the XMFA is posted using the Pull Capability and to pull the message from the system memory. To release an XMFA back to the Host platform the IOP writes the XMFA back to the Host Free List FIFO which resides in system memory.

2.6.6.2 Host Free List Address

The address and size of the Host Free List FIFO is provided to the IOP by the I_2O System Host in an I_2O defined “IOP Message Pull Extensions” Message. The Host Free List FIFO structure is located at a memory boundary equal to its size to enable the IOP to know when it has reached the end of the FIFO. When the IOP returns XMFAs to the Host Free List FIFO sets the P bit to 1. When the IOP reaches the end of the FIFO resets the FIFO index to the base address and this time through write the P bit to 0. This allows the Host to track the progress of the local IOP in returning XMFAs.

Figure 11: PowerSpan II I_2O Pull Capability

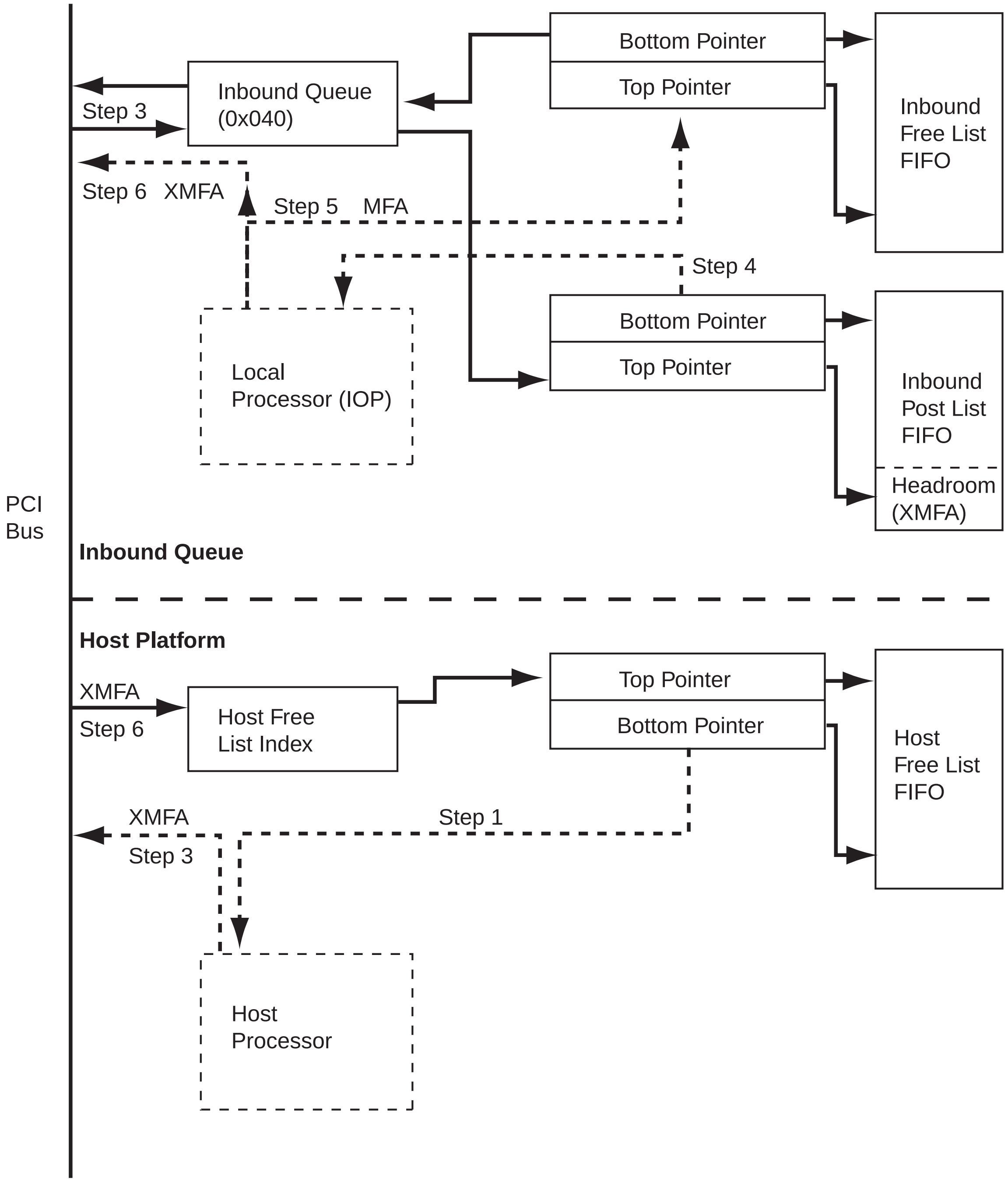


Figure 11 illustrates the following steps in PowerSpan II $I_2 0$ pull capability:

1. Host reads XMFA from Host Free List
2. Host writes message to MF in Host memory
3. Host writes XMFA to Inbound Queue

4. Local processor reads XMFA from the Inbound Post List FIFO
5. Local processor copies MF from Host memory
6. Local processor writes XMFA to Host Free List Index

2.6.7 Outbound Option

The *I_2O 2.0 Specification* allows for the IOP to provide an enhanced capability to post reply messages to the Host. This mechanism is independent of the Pull Capability of the previous section. This capability reduces the number of reads that the Host must perform to the IOP. Under the Outbound Option Operation, the local IOP copies out the reply message to the Host system memory and then posts the message by performing a single write to the Host memory. The Host need only to write to the PowerSpan II to return the MFA.

The Outbound option requires 16-byte alignment of the message frames and thus the least significant four bits of the MFA are always zero. The Outbound option uses these four bits to create and Extended MFA (XMFA). The least significant bit of the outbound XMFA is the Cycle Indicator bit or the C bit.

2.6.7.1 Host Posting

To post a message to the Host, the IOP completes the following:

1. Obtains an Outbound MFA from the Outbound Free List FIFO.
2. Copies out the reply message to the MF indicated by the Host allocated Outbound MFA.
3. Posts the Outbound MFA to the HostPostList FIFO pointed to by the IOP Outbound Index Register, setting the least significant bit of the MFA to 1, and increment the IOP Outbound Index Register by writing to the IOP Outbound Index Increment Register.

The PowerSpan II IOP Outbound Index Register is initialized by the IOP with a value received along with the Host Outbound Post List FIFO Size through an “IOP Message Outbound Extensions” message from the Host. The size of the Host Outbound Post List FIFO is specified in the HOPL_SIZE bit in the I2O_CSR register.

The PowerSpan II IOP Outbound Index Register points to the Top of the Host Outbound Post List FIFO implemented in Host memory. When it reaches the end of the FIFO the IOP resets the IOP Outbound Index Register to the base of the FIFO. The IOP writes XMFAs to the FIFO with the C bit set to 0, and continues to alternate this pattern. This allows the Host to determine where the IOP processor has last written to the FIFO.

PowerSpan II also implements a Host Outbound Index Register where the Host will write its Host Outbound Post List FIFO Index after servicing Outbound reply messages posted using the Outbound Option. The Host Outbound Index Register points to the Bottom of the Host Outbound Post List FIFO. PowerSpan II maps this register into the PowerSpan II I_2O Target Image Shell Interface at the offset specified in the I_2O Host Outbound Index Offset Register. This register is initialized by the IOP with an offset provided by the Host through the IOP Message Outbound Extensions message.

When I_2O Extended capabilities are enabled with I2O_CSR[XI2O_EN], PowerSpan II will set an interrupt status bit in the I_2O Outbound Post List Interrupt Status register when the I_2O Host Outbound Index Register is not equal to the I_2O IOP Outbound Index Register. This indicates that the Host Outbound Post List FIFO is non-empty.

PowerSpan II compares the value of the Host Outbound Index Register to the IOP Outbound Index Register. If they are identical the Interrupt is cleared by the PowerSpan II. If these registers differ, then it is assumed that the PowerSpan II has posted additional Outbound reply messages which have not yet been serviced by the Host, and therefore, the PowerSpan II continues to assert the Interrupt to the Host. The Host will post empty MFAs back to the IOP by writing to the PowerSpan II’s Outbound Queue Register (0x044), with the C bit set to zero. PowerSpan II services the written MFA the same as a normal Outbound MFA being returned to the IOP.

Figure 12: PowerSpan II I_2O Outbound Capability

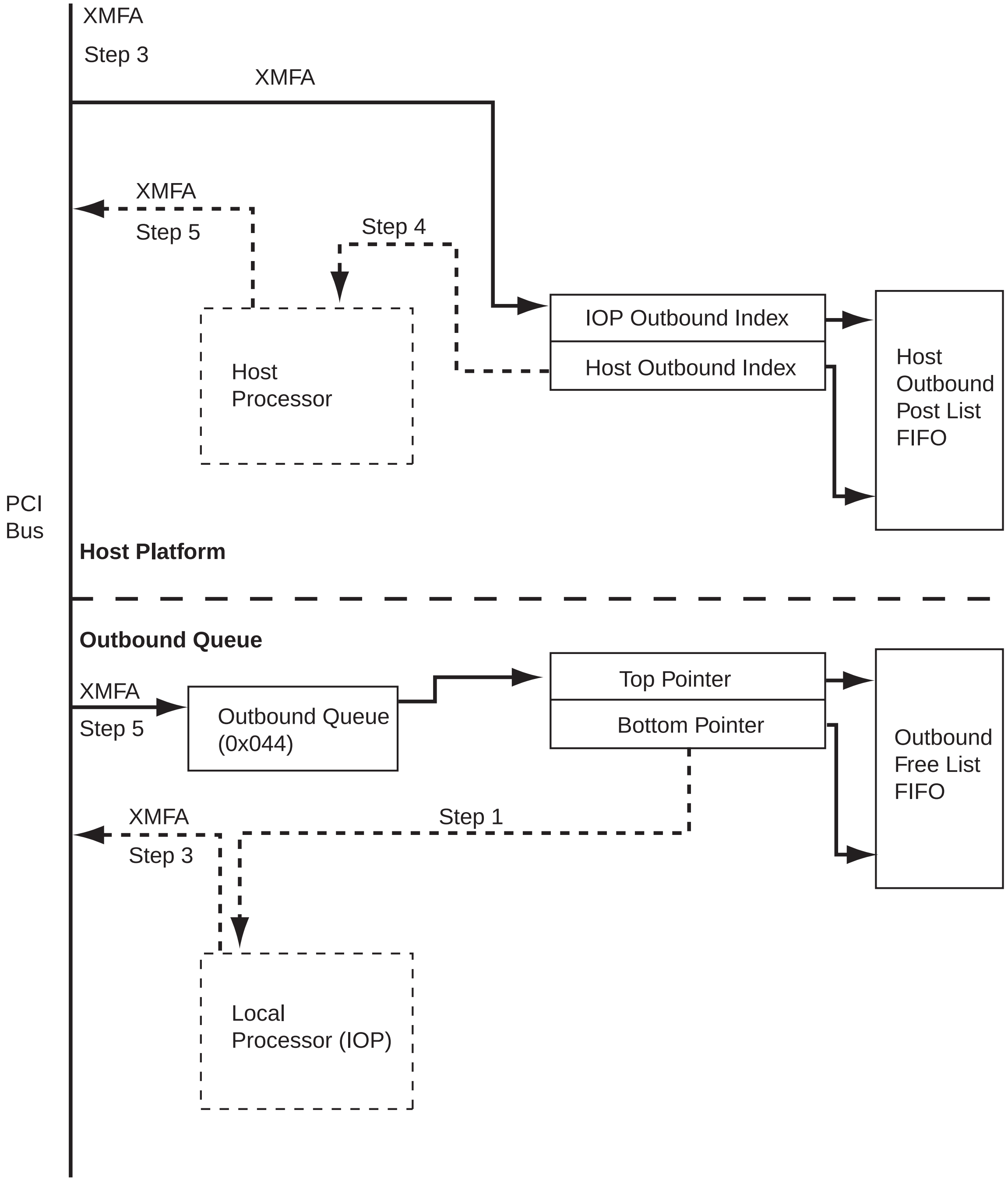


Figure 12 illustrates the following steps in PowerSpan II I_20 outbound capability:

1. Local processor reads the Outbound Free List to obtain an MFA
2. Local processor writes the MF in the Host memory
3. Local processor writes the MFA to the Host Outbound Post List FIFO, setting the P bit

4. Host processor reads the XMFAs from the Host Outbound Post List
5. Host writes the XMFA to the Outbound Queue (0x044)

2.6.8 I$_2$O Standard Registers

This section defines the standard I$_2$O register set supported by PowerSpan II. These registers are accessible within the PowerSpan II I$_2$O target image. In Table 10, all standards-based registers are in italics.

Table 10: PowerSpan II I20 Target Image Map

Offset (HEX)	Register Mnemonic	Register Name
0x000-028	PowerSpan II Reserved	
0x030	OPL_IS	*I$_2$0 Outbound Post List Interrupt Status Register*
0x034	OPL_IM	*I$_2$0 Outbound Post List Interrupt Mask Register*
0x038	PowerSpan II Reserved	
0x040	IN_Q	*I$_2$0 Inbound Queue*
0x044	OUT_Q	*I$_2$0 Outbound Queue*
0x048-[HOST_OIO]-4	PowerSpan II Reserved	
[HOST_OIO]	HOST_OI	*I$_2$O Host Outbound Index Register*
[HOST_OIO]+4-0xFF	PowerSpan II Reserved	
0x100-xxx	I$_2$0 Inbound Message Frames	

The I$_2$O Shell Interface is located in the first 4 Kbytes of the PowerSpan II I$_2$O target image. The I$_2$O Inbound Message Frames occupies offsets above the 4 Kbyte point of the PowerSpan II I$_2$O target image. The upper limit of the I$_2$O Inbound Message Frames is determined by the size of the PowerSpan II I$_2$O target image, as defined by the PCI_I2O_CTL[BS] register.

The offset of the I$_2$O Host Outbound Index Register is programmed in the I$_2$O Host Outbound Index Offset Register (HOST_OIO) of the PowerSpan II Register Map.

The following tables show the I$_2$O register definitions.

Table 11: I2O Outbound Post List Interrupt Status Register

Register Name: OPL_IS	**Register Offset: 030**

<table>
<tr><th>PCI
Bits</th><th colspan="3">Function</th><th>PPC
Bits</th></tr>
<tr><td>31-24</td><td colspan="3">I_2O Reserved</td><td>0-7</td></tr>
<tr><td>23-16</td><td colspan="3">I_2O Reserved</td><td>8-15</td></tr>
<tr><td>15-08</td><td colspan="3">I_2O Reserved</td><td>16-23</td></tr>
<tr><td>07-00</td><td>I_2O Reserved</td><td>OPL_
ISR</td><td>I_2O Reserved</td><td>24-31</td></tr>
</table>

OPL_IS Description

Name	Type	Reset By	Reset State	Function
OPL_ISR	R	Px_RST	0	Outbound Post List Interrupt Service Request 0 = Outbound Post_List FIFO is empty 1 = Outbound Post_List FIFO is not empty. The value of the interrupt mask bit does not affect this bit.

The *$I_2$0 2.0 Specification* requires the Outbound Post_List Interrupt Status register to be located at offset 0x30 in the *Memory* region specified by the first base address register ($I_2$0 Base Address Register - Px_BSI2O).

When the $I_2$0 messaging unit in PowerSpan II is enabled (I2O_CSR[I2O_EN] = 1), a Memory access from PCI to offset 0x30 from Px_BSI2O is destined for OPL_IS.

When the $I_2$0 messaging unit in PowerSpan II is not enabled, the OPL_IS register is not visible to read or write access. The register essentially disappears from all PowerSpan II memory maps.

OPL_ISR is bit three of this register.

Table 12: I2O Outbound Post List Interrupt Mask Register

Register Name: OPL_IM	Register Offset: 034

PCI Bits	Function			PPC Bits
31-24	I_2O Reserved			0-7
23-16	I_2O Reserved			8-15
15-08	I_2O Reserved			16-23
07-00	I2O Reserved	OP_ISM	I_2O Reserved	24-31

OPL_IM Description

Name	Type	Reset By	Reset State	Function
OP_ISM	R/W	Px_RST	0	Outbound Post_List Interrupt Mask 0 = Outbound Post_List Interrupt is enabled 1 = Outbound Post_List Interrupt is masked

The *$I_2$0 2.0 Specification* requires the Outbound Post_List Interrupt Mask register to be located at offset 0x34 in the memory region specified by the first base address register ($I_2$0 Base Address Register - Px_BSI2O).

When the $I_2$0 messaging unit in PowerSpan II is enabled (I2O_CSR[I2O_EN] = 1), a memory access from PCI to offset 034h from Px_BSI2O is destined for OPL_IM.

When the $I_2$0 messaging unit in PowerSpan II is not enabled, the OPL_IM register is not visible to read or write access. The register essentially disappears from all PowerSpan II memory maps.

OP_ISM is bit 3 of this register.

Table 13: I2O Inbound Queue

Register Name: IN_Q	Register Offset: 040

PCI Bits	Function	PPC Bits
31-24	MFA	0-7
23-16	MFA	8-15
15-08	MFA	16-23
07-00	MFA	24-31

IN_Q Description

Name	Type	Reset By	Reset State	Function
MFA[31:0]	R/W	Px_RST	0	Inbound Message Frame Address

MFA: The Inbound Message Frame Address specifies locations in the IOP memory map where Inbound Message Frames reside.

The Message Frame Address is the offset from the beginning of the I_2O target image window in the destination bus memory map and the destination address where the Message Frame begins.

A read from the I_2O Inbound Queue returns the next available MFA from the I_2O Inbound Free List FIFO. This is a destructive read.

A write to this offset is used to place a MFA into the I_2O Inbound Post List FIFO. The PowerSpan II accepts the write cycle as a posted write and is responsible for completing the cycle on the destination bus.

When the I_20 Interface in PowerSpan II is not enabled, the IN_Q register is not visible to read or write access. The register essentially disappears from all PowerSpan II memory maps.

Table 14: I2O Outbound Queue

Register Name: OUT_Q	Register Offset: 044

PCI Bits	Function	PPC Bits
31-24	MFA	0-7
23-16	MFA	8-15
15-08	MFA	16-23
07-00	MFA	24-31

OUT_Q Description

Name	Type	Reset By	Reset State	Function
MFA[31:0]	R/W	Px_RST	0	Outbound Message Frame Address

MFA: The Outbound Message Frame Address specify locations in the Host memory map where Outbound Message Frames reside.

The Message Frame Address is the Host memory address of the Message Frame.

A read from the I_2O Outbound Queue returns the next MFA from the I_2O Outbound Post List FIFO. This is a destructive read.

A write to this offset places a Free Host MFA into the I_2O Outbound Free List FIFO. PowerSpan II accepts the write cycle as a posted write and is responsible for completing the cycle on the destination bus.

When the I_20 Interface in PowerSpan II is not enabled, the OUT_Q register is not visible to read or write access. The register essentially disappears from all PowerSpan II memory maps.

Table 15: I2O Host Outbound Index Register

Register Name: HOST_OI	Register Offset: [HOST_OIO]

PCI Bits	Function			PB Bits
31-24	OI			0-7
23-16	OI			8-15
15-08	OI			16-23
07-00	OI	0	0	24-31

HOST_OI Description

Name	Type	Reset By	Reset State	Function
OI[29:0]	R/W	Px_RST	0	Host Outbound Index

OI: This register indicates the address in Host memory from which the Host is to retrieve the next Outbound XMFA. This register is initialized by the IOP with an index received from the Host in an I_2O message. The register is written by the Host during I_2O Outbound Option message passing.

When the I_2O Host Outbound Index Register and the I_2O IOP Outbound Index Register differ, the Outbound Post List Interrupt Status bit is set in the OPL_IS register at offset 0x30 of the PCI I_2O target Image. When these registers contain the same Host memory address, the Interrupt is cleared.

This feature is only supported if the I_2O Outbound Option is enabled with the XI2O_EN bit in the I2O_CSR register and I2O_EN.

The HOPL_SIZE bit in the I2O_CSR register determines the alignment of this Index register.

The Register Offset is specified in the I_2O Host Outbound Index Offset Register at offset 0x548 of the PowerSpan II Register Map. The I_2O Host Outbound Index Register must be located in the lower 4 Kbytes of the PCI I_2O target image map.

When the I_20 Interface in PowerSpan II is not enabled, the HOST_OI register is not visible to read or write access. The register essentially disappears from all PowerSpan II memory maps.

3. Processor Bus Interface

This chapter describes the functionality of the Processor Bus Interface. Both the Single PCI PowerSpan II and Dual PCI PowerSpan II have a Processor Bus Interface. The following topics are discussed:

- "Overview" on page 101
- "PB Slave Interface" on page 103
- "PB Master Interface" on page 123

3.1 Overview

The PowerSpan II Processor Bus (PB) Interface directly connects with a wide range of processors in order to meet the demands of high end systems, the PB Interface operates up to100 MHz and has a 64-bit data bus.

3.2 Interface Support

The PowerSpan II Processor Bus Interface supports the following PowerPC devices:

- PowerQUICC II (MPC8260, and MPC8260A)
- MPC74xx
- PowerPC 7400
- PowerPC 750
- PowerPC 740
- PowerPC 603e processors

These interfaces are not identical, but for the most part, the processor interface on the PowerSpan II is referred to simply as the Processor Bus (PB). The interface sections in this chapter highlight where the PowerSpan II operates differently to address specific processor requirements as the need arises. An example of this different operation is the extended cycles with the MPC8260.

⚠ The MPC8260 and PowerPC 7400 must operate in 60x compatible bus mode to be used with PowerSpan II. In single MPC8260 mode the processor cannot share the bus with other external masters.

3.2.1 Terminology

The following terms are used in the Processor Bus Interface descriptions:

- **address retry window**: refers to the clock following the assertion of AACK_, which is the latest a snooping master can request for an address tenure re-run.
- **window of opportunity***:* refers to the clock following the assertion of ARTRY_. The retrying master has to request the bus on this clock to ensure that it is the next bus owner. This enables it to perform the transactions required to maintain cache coherency.

3.2.2 PB Bus Interface Descriptions

The PB Bus Interface is described in terms of its master and slave functions. The PCI interfaces on PowerSpan II are described in terms of its PCI master and PCI target functions. This description is largely independent of PCI-1 versus PCI-2, or the assignment of the Primary PCI Interface functions. Exceptions to these rules are noted in the manual.

3.2.2.1 Transaction Ordering

For information on PowerSpan II's PCI transaction ordering refer to "Transaction Ordering" on page 47.

3.3 PB Slave Interface

3.3.1 PowerSpan II as PB Slave

PowerSpan II becomes active as a PB slave when one of the following conditions occurs:

- A processor bus master accesses a PCI resource, generating a memory or I/O space access
- A processor bus master accesses a PCI resource, generating a configuration or IACK access
- A processor bus master accesses PowerSpan II registers

This section covers the first two of these conditions. See "Register Access" on page 283 for a discussion of the last two items in the bullet list above.

The operation of the PB Slave is described below by dividing the PB Slave transaction into the following different phases:

- **Address Phase**: This section discusses the decoding of processor bus accesses.

- **Data Transfer**: This section describes control of transaction length.
- **Terminations**: This section describes the terminations supported by PowerSpan II, and exception handling.

The PowerSpan II PB Slave supports cacheable accesses to PCI, but it does not guarantee coherency if more than one processor accesses a given range of memory. In order to address this issue, operating system pages mapped to PowerSpan II must have the Memory Coherency Attribute (M) set to zero. PowerSpan II performs PCI read prefetches. These reads can be cached in an internal queueing memory within PowerSpan II — if PRKEEP is set to 1. When a write is performed to a prefetched address, a subsequent read yields stale data. Prefetching attributes for each image map must meet the systems cache coherency requirements.

Pull-up resistors are not required on the processor bus address (PB_A[0:31]) and data (PB_D[0:63]) signals to guarantee functional operation of PowerSpan II. However, adding resistors to the address and data signals minimizes the current drawn by the PowerSpan II's tristated buffers when the bus is in an idle condition. The system designer must decide whether to add these resistors to the address and data bus.

3.3.2 Address Phase

3.3.2.1 Transaction Decoding

Transaction decoding on the PB Slave operates in both normal decode mode and Master-based decode mode.

When PowerSpan II is in normal decode mode, each PB slave monitors the Processor Bus Address (PB_A[]). When the address falls into one of the programmed windows, and the Transfer Type (PB_TT[]) is supported, PowerSpan II claims the address tenure.

A PB slave image is defined as the range of processor bus physical address space that decodes a PowerSpan II access.

PB slave image location is controlled by setting the Base Address (BA) field in the “Processor Bus Slave Image x Base Address Register” on page 350. PB slave image size is controlled by setting the Block Size (BS) field in the “Processor Bus Slave Image x Control Register” on page 342.

PowerSpan II supports eight general purpose slave images and four specialty slave images. A general purpose slave image generates memory or I/O reads and writes to the PCI bus. For example, the eight general purpose slave images can support the local bus traffic of four MPC8260 SCCs, two threads of CPU traffic destined for PCI-1, and two threads destined for PCI-2. The specialty images are used for the generation of PCI Configuration cycles on PCI-1 and PCI-2, IACK reads on PCI-1, IACK reads on PCI-2 and PowerSpan II register accesses.

The PB slave image also controls how an incoming PB transaction is mapped to the destination port on PowerSpan II. For example, there are bits for endian mapping, prefetch behavior, etc. Table 16 describes the programming model for a PB Slave Image Control register.

The PB slave image only claims a transaction when all of the following conditions are met:

- the external address matches the slave image
- the transaction codes are supported

⚠ In normal decoding mode (see "Transaction Decoding" on page 104), the PB slave image claims transactions initiated by the PowerSpan II PB Master Interface if the transaction meets the two conditions listed above. In order to avoid the PB slave from claiming transactions from the a transaction PowerSpan II PB Master Interface, the Master-based Decode functionality can be enabled.

Table 16 describes the bits and default settings of the PB Slave Image Control register (see page 342).

Table 16: Programming Model for PB Slave Image Control Register

Bits	Type	Description	Default Setting
IMG_EN	R/W	Enables the PB slave image to decode in the specified physical address range.	Disabled
TA_EN	R/W	Enables address translation (see “Processor Bus Slave Image x Translation Address Register” on page 348).	Disabled
BS[4:0]	R/W	Sets the block size of the PB slave image. The size of the image is 4 Kbyte * 2^{BS}.	Default value is 0. It can be programmed through any port after reset, or loaded through EEPROM.
MODE	R/W	Maps the incoming PB transaction to either Memory or I/O space on the PCI bus.	Defaults to Memory space.
DEST	R/W	Directs the incoming PB transaction to either of PCI-1 or PCI-2	Defaults to PCI-1
MEM_IO	R/W	Enables 1,2,3, or 4 byte memory reads on the PCI bus(es).	Regular I/O mode
PRKEEP	R/W	Enables PowerSpan II to keep prefetch read data over subsequent transactions.	Disabled
END[1:0]	R/W	Sets endian mapping to little-endian, PowerPC little-endian, or big-endian	Big-endian is the default mode.
RD_AMT[2:0]	R/W	Controls the prefetch read amount. Can be programmed up to a maximum of 128 bytes.	8 bytes is the default prefetch read amount

PB memory management supports a variety of memory/cache access attributes: write through (W), caching-inhibited (I), and memory coherency (M). Although PowerSpan II does not decode these attributes — external pins PB_GBL_ and PB_CI_ are output only— specific guidelines must be followed to ensure correct system operation. These guidelines are shown in Table 17.

Table 17: Recommended Memory/Cache Attribute Settings

PowerSpan II Resource	Memory Coherency	Caching Inhibited
Registers	M=0	I=1
PCI I/O space	M=0	I=1
PCI Memory space	M=0	External L2 cache: I=1 No External L2 cache: I=0 or 1

Register and PCI I/O space accesses require I=1 because PowerSpan II does not accept burst transactions to these resources.

Master-based Decode Mode

The PB Slave Interface supports Master-based decode mode when the internal PowerSpan II processor bus arbiter is enabled (see “Processor Bus Arbitration” on page 168) and the Master Decode Enable (MD_EN) bit is set in the “Processor Bus Slave Image x Control Register” on page 342. When Master-based decode is enabled, a PB slave image only claims a transaction decoded for its specified physical address space if it originates from specific processor bus master or masters.

External bus masters are selected for a specific target by setting one or more of the M1 to M3 bits in the “Processor Bus Slave Image x Translation Address Register” on page 348.

The PB slave image only claims a transaction when all of the following conditions are met:

- the address matches the slave image
- the transaction codes are supported
- Mx is set and the identified master is requesting a transaction

PowerSpan II behavior is undefined if more than one identically programmed, or overlapping, slave image claims a transaction. For example, if two slave image have the same base address and size, then they must have unique master bits set in the “Processor Bus Slave Image x Translation Address Register” on page 348.

3.3.2.2 Transfer Types

The PB Slave only claims processor bus transactions with specific transfer types. The supported transfer types consist of address only, read, and write. They are defined in Table 18.

All reads are treated as delayed reads and can be single cycle, extended or bursts. All writes are treated as posted writes and can be single cycle, extended or bursts. PowerSpan II handles address only cycles by asserting PB_AACK_ — no data transfer occurs.

Address only transfer types are claimed to ensure PowerSpan II does not negatively impact cache control, reservation, or ordering transactions on the processor bus.

Table 18: PowerSpan II PB Slave Transfer Types

TT[0:4]	Name
Address Only	
00000	Clean Block
00100	Flush Block
01000	Sync Block
01100	Kill Block
10000	eieio
11000	tlb invalidate
00001	lwarx
01001	tlb sync
01101	icbi
Reads	
01010	Read
01110	Read with intent to modify
11010	Read Atomic
11110	Read with intent to modify atomic
01011	Read with no intent to cache
Writes	
00010	Write with flush
00110	Write with kill
10010	Write with flush atomic

Because PowerSpan II does not have a cache, all read and write transfer types are treated the same. For example, a Read with Intent to Modify command (PB_TT= 01110) is handled the same way as a Read Atomic command (PB_TT= 11010).

PowerSpan II performs PCI read prefetches and stores read data in an internal buffer when the Prefetch Keep (PRKEEP) bit is set to 1. The purpose of a prefetch read is to fetch read information before the master requests the information. If the master then requests the information the target can respond immediately with the prefetched information. This ability protects the master from slow access times for information it requires. However, when a write is performed to a prefetched address, a subsequent read could yield stale data. In order to guarantee there is no stale data, set the PRKEEP bit to 0. This function disables the internal buffer to ensure there is no stale data. By setting this PRKEEP bit to 0 PowerSpan II is unable to perform PCI read prefetches and read performance may be decreased in the system.

Prefetching attributes for each image map must meet the system's cache coherency requirements

3.3.2.3 Address Tenure

Each slave on the PB Interface is responsible for the following:

- decoding the address broadcast by the master
- claiming the address tenure with PB_AACK_ assertion
- managing the data termination signals during the data tenure

The PB Slave uses PB_AACK_ to limit the level of address pipelining to one. The earliest the PB slave can assert PB_AACK_ is two clocks after PB_TS_.

The PB Slave does not acknowledge subsequent address phases until it finishes its participation in the current data tenure. If the previous address phase was claimed by another slave, the PB slave does not acknowledge the current address phase until the previous slave completes its data tenure.

The use of PB_ARTRY_ by the PB Slave is enabled by the Address Retry Enable (ARTRY_EN) bit in the "Processor Bus Miscellaneous Control and Status Register" on page 361. If the ARTRY_EN bit is set to 0, the PB_ARTRY_ signal is not asserted and the PB slave retains ownership of the bus. The PB Slave retains ownership after the assertion of PB_AACK_ and until it is able to assert PB_TA.

When ARTRY_EN has a value of 1, the PB Slave can assert PB_ARTRY_. The default setting is 0 (ARTRY_EN is disabled). The PB Interface has higher performance if the ARTRY_EN bit is enabled. PowerSpan II's PB Master or another external master can gain access to the bus when PowerSpan II cannot assert PB_TA.

When ARTRY_EN is enabled, the PB Slave asserts PB_ARTRY_ in the following situations:

- a write destined for PCI cannot be internally buffered
- when a read request has been latched and read data is being fetched from PCI
- a register access when a load from EEPROM is in progress
- writing to registers when another bus (PCI-1, PCI-2) is also writing to the register block

If the assertion of PB_ARTRY_ is enabled, it occurs the clock after PB_AACK_ within the *address retry window.*

3.3.2.4 Address Translation

The incoming address on the PB Interface can have a translation offset applied to it using the TADDR[19:0] field of the "Processor Bus Slave Image x Translation Address Register" on page 348. When the translation offset is applied to the incoming PB address, the translated address appears on the destination bus (PCI-1 or PCI-2). The translation offset replaces the PB address, up to the size of the image. TADDR[19:0] replaces PB address lines PB_A[0:19].

For example, if TADDR[19:0] = 0x12345 and the BS bit in the PB_SIx_CTL register equals 0 (4-Kbyte image) and the address on the processor bus is PB_A[0:31] = 0x78563412, then the PCI address becomes 0x12345412. Table 19 summarizes the relationship between translation offset, processor bus address, and block size of the image.

Table 19: Translation Address Mapping

PB_SIx_TADDR	Processor Bus Address (PB_A)	BS bit (PB_SIx_CTL register)	Block Size
31	0	10011	2G
31:30	0:1	10010	1G
31:29	0:2	10001	512M
31:28	0:3	10000	256M
31:27	0:4	01111	128M
31:26	0:5	01110	64M
31:25	0:6	01101	32M
31:24	0:7	01100	16M
31:23	0:8	01011	8M
31:22	0:9	01010	4M
31:21	0:10	01001	2M
31:20	0:11	01000	1M
31:19	0:12	00111	512k
31:18	0:13	00110	256k
31:17	0:14	00101	128k
31:16	0:15	00100	64k
31:15	0:16	00011	32k
31:14	0:17	00010	16k
31:13	0:18	00001	8k
31:12	0:19	00000	4k

3.3.2.5 Address Parity

Address parity checking is provided on each byte of the address bus. Address parity bit assignments are defined in Table 20.

Table 20: PowerSpan II PB Address Parity Assignments

Address Bus	Address Parity
PB_A[0:7]	PB_AP[0]
PB_A[8:15]	PB_AP[1]
PB_A[16:23]	PB_AP[2]
PB_A[24:31]	PB_AP[3]

When the PB Slave detects an address parity error during its decode process it does not assert Address Acknowledge (PB_AACK_). Address parity checking is enabled with the Address Parity Enable (AP_EN) bit in the "Processor Bus Miscellaneous Control and Status Register" on page 361. Odd parity versus even parity is configured with the PARITY bit in the same register.

Special Parity Requirements with the MPC8260

Address parity and data parity must be specially programmed in a joint PowerSpan II and MPC8260 application.

In a joint application all memory accesses from the MPC8260 to PowerSpan II must be routed through the internal memory controller on the MPC8260. When the data is passed through the memory controller both address parity and data parity can be used in the system.

If accesses do not pass through the memory controller of the MPC8260 before reaching PowerSpan II, and PowerSpan II has either or both address and data parity enabled, then PowerSpan II reports parity errors on the transaction.

To enable or disable address parity in PowerSpan II, set the Address Parity Enable (AP_EN) bit in the Processor Bus Miscellaneous Control and Status (PB_MISC_CSR) register (see page 361).

To enable or disable data parity in PowerSpan II, set the Data Parity Enable (AP_EN) bit in the Processor Bus Miscellaneous Control and Status (PB_MISC_CSR) register (see page 361).

3.3.3 Data Phase

3.3.3.1 Transaction Length

The PB Slave supports a set of the data transfer sizes supported by the embedded PowerPC family. All data transfer sizes supported by the PowerSpan II PB Slave are illustrated in Table 21. Burst transfers are indicated by the assertion of Processor Bus Transfer Burst (PB_TBST_). The shaded regions indicate transaction sizes unique to the MPC8260.

Table 21: PowerSpan II PB Transfer Sizes

Transfer Size	Bytes	PB_TBST	PB_TSIZ[0]	PB_TSIZ[1:3]
Byte	1	1	0	001
Half-word	2	1	0	010
Tri-byte	3	1	0	011
Word	4	1	0	100
Five bytes	5	1	0	101
Six bytes	6	1	0	110
Seven bytes	7	1	0	111
Double Word (DW)	8	1	0	000
Extended Double (MPC8260 only)	16	1	1	001
Extended Triple (MPC8260 only)	24	1	1	010
Burst (Quad DW)	32	0	0	010

3.3.3.2 Data Alignment

Embedded processor bus transfer sizes and alignments, defined in Table 21 and Table 22, are supported by the PB Slave for transaction accesses. The size and alignment combinations defined in Table 22 are supported by the MPC8260, PowerPC 740, and PowerPC 750 processors. The shaded table cells show transactions that support the PowerPC 7400 processor.

Table 22 lists the size and alignment transactions less than or equal to 8 bytes. PowerSpan II register accesses are limited to 4 bytes or less.

The PowerSpan II port size is 64-bit.

Table 22: PowerSpan II Processor Bus Single Beat Data Transfers

Size	TSIZ[0:3]	A[29:31]	Data Bus Byte Lanes							
			0	1	2	3	4	5	6	7
Byte	0001	000	D0							
	0001	001		D1						
	0001	010			D2					
	0001	011				D3				
	0001	100					D4			
	0001	101						D5		
	0001	110							D6	
	0001	111								D7
Half word	0010	000	D0	D1						
	0010	001		D1	D2					
	0010	010			D2	D3				
	0010	011				D3	D4			
	0010	100					D4	D5		
	0010	101						D5	D6	
	0010	110							D6	D7
Tri-byte	0011	000	D0	D1	D2					
	0011	001		D1	D2	D3				
	0011	010			D2	D3	D4			
	0011	011				D3	D4	D5		
	0011	100					D4	D5	D6	
	0011	101						D5	D6	D7

Table 22: PowerSpan II Processor Bus Single Beat Data Transfers

Size	TSIZ[0:3]	A[29:31]	Data Bus Byte Lanes							
Word	0100	000	D0	D1	D2	D3				
	0100	001		D1	D2	D3	D4			
	0100	010			D2	D3	D4	D5		
	0100	011				D3	D4	D5	D6	
	0100	100					D4	D5	D6	D7
Five bytes	0101	000	D0	D1	D2	D3	D4			
	0101	001		D1	D2	D3	D4	D5		
	0101	010			D2	D3	D4	D5	D6	
	0101	011				D3	D4	D5	D6	D7
Six bytes	0110	000	D0	D1	D2	D3	D4	D5		
	0110	001		D1	D2	D3	D4	D5	D6	
	0110	010			D2	D3	D4	D5	D6	D7
Seven bytes	0111	000	D0	D1	D2	D3	D4	D5	D6	
	0111	001		D1	D2	D3	D4	D5	D6	D7
Double word	0000	000	D0	D1	D2	D3	D4	D5	D6	D7

The information in Table 22 is independent of endian considerations and pertains to byte lane control on the processor bus. For endian considerations, please consult “Endian Mapping” on page 213.

PowerPC 7400 Transaction Support

The PowerPC 7400 processors supports misaligned transactions within a double word (64-bit aligned) boundary. As long as the transaction does not cross the double word boundary, the PowerPC 7400 can transfer data on the misaligned address.

PowerSpan II supports a specific types of the PowerPC 7400 misaligned transactions (shown in Table 22) when the MODE_7400 bit is set in the Processor Bus Miscellaneous Control and Status register (page 361). Any misaligned transaction between PowerSpan II and the MPC7400 that is a single word (32-bit) or less must be within a single word aligned boundary. Any transfer greater than a single word must start or end on a word boundary.

⚠ Software must make sure that the PowerPC 7400 does not initiate unsupported misaligned transactions to PowerSpan II

3.3.3.3 Cache Line Size

The supported embedded PowerPC processors implement a 32-byte cache line size. Cache wrap reads are supported by the PB slave for burst and extended transactions.

PowerPC processors do not generate cache wrap writes.

3.3.3.4 Reads

Address Retry Enable

The PB slave supports up to eight concurrent delayed reads when the Address Retry Enable (ARTRY_EN) bit in the "Processor Bus Miscellaneous Control and Status Register" on page 361 is set to 1. Refer to "Concurrent Reads" on page 38 for more information on read pipelining in PowerSpan II.

When an external master makes an initial read request, the PowerSpan II PB slave latches the address. This initiates a read on the destination bus. The destination bus is specified by the Destination Bus (DEST) bit in the "Processor Bus Slave Image x Control Register" on page 342.

Delayed Reads

The outstanding read is referred to as a delayed read. Delayed reads consist of the following phases:

1. Delayed Read Request
   — PowerSpan II PB Slave latches transaction parameters and issues a retry
2. Delayed Read Completion
   — The PB Slave obtains the requested data and completion status on the destination bus
3. Read Completion
   — The master repeats the transaction with the same parameters used for the initial request

Any attempt by a processor bus master to complete the read transaction is retried by the PowerSpan II PB Slave until the following byte quantities are available in the line buffer:

- 32 bytes
- 8 bytes if the RD_AMT=0 (see "Processor Bus Slave Image x Control Register" on page 342)
- 16 bytes if the RD_AMT=1

Read Amount

All PowerSpan II PB slave reads destined for PCI Memory space are considered prefetchable to 8-byte boundaries by default. Setting the MEM_IO bit in the Processor Bus Slave Image *x* Control register (see page 342) enables 1,2,3, or 4 byte reads from the PCI bus(es).

In order to program PowerSpan II to complete 4 byte reads on the PB bus, both the MEM_IO bit and the MODE bit must be set to 1 in the Processor Bus Slave Image *x* Control register.

In order to perform a 4-byte read from the processor (60x) bus to PCI, the following bits must be programmed:

- MEM_IO bit set to 1
- MODE bit set to 1
- END bit, in the "Processor Bus Slave Image x Control Register" on page 342, must not be set to little-endian mode (00). It can be set to PowerPC little-endian (01), or big-endian (10).

⚠ When the Slave Image Control register is programmed for 4 byte read transactions, requesting 8 byte reads causes undefined results in the system.

The amount of data prefetched on the destination bus is specified using the Prefetch Read Amount (RD_AMT[2:0]) field in the "Processor Bus Slave Image x Control Register" on page 342. If the Prefetch Keep (PRKEEP) bit is set, then PowerSpan II automatically increments the latched address every time the processor bus master returns for read data. This PRKEEP function enables a burst read by the PowerSpan II PCI Master to be unpacked as smaller transfers on the processor bus.

The PB Interface can generate a 32-byte burst read with a starting address at the second, third or fourth 8-byte quantity. A cache wrap read always causes the PB slave to make a 32-byte read request from the destination PCI bus. In other words, PRKEEP and RD_AMT[2:0] have no effect.

There are instances where a read requires more data than that specified by RD_AMT. Since PB slaves cannot terminate transactions, PowerSpan II compensates for a potential hang situation — for example, not having enough read data — by over-riding the programming of RD_AMT. PowerSpan II prefetches the larger data value. This enables the PowerSpan II to accommodate the byte count specified by the transaction. Alternatively, it initiates a new read transaction on the destination if it does not have enough data to satisfy the transaction.

The read amount values that can be programmed in the RD_AMT field are shown in Table 23. The read amount setting determines different values to prefetch from the destination bus.

Table 23: Read Amount settings

RD_AMT[2:0]	Data Fetched
000	8 bytes
001	16 bytes
010	32 bytes
011	64 bytes
100	128 bytes
101-111	Reserved

Discard Timer

Each PB slave image has a discard timer. If an external master does not claim data within 2^{15} clocks after data is read from the destination bus, the Delayed Read Request latch is de-allocated. This prevents deadlock conditions. Read buffer contents are flushed but there is no error recorded and no interrupts are generated.

Posted Writes

Posted writes have dedicated line buffers and are treated independently of reads. A write to an image does not invalidate the contents of the read line buffer currently in use.

Address Retry Disabled

The PB Slave supports a single read at a time when ARTRY_EN is disabled. ARTRY_EN is disabled by setting the bit to 0. The PB slave acknowledges the address tenure with the PB_AACK_ signal and captures the address in the Delayed Read latch. However, when ARTRY_EN is disabled, the PB slave does not acknowledge the data transfer until the Read Amount (RD_AMT) field in the "Processor Bus Slave Image x Control Register" on page 342 is read. The Delayed Read Request latch is de-allocated when the external processor bus master completes the transaction.

PRKEEP has no affect when PKEEP is set to 1 and ARTRY_EN is disabled. A maximum of 32 bytes can be programmed in the RD_AMT field.

3.3.3.5 Writes

All writes are posted and are buffered separately from read data. The transaction length of the PB write is directly translated to the PCI bus with no address phase deletion. For example, a single cycle write on the PB results in a single cycle write on the PCI bus.

3.3.3.6 Data Parity

Data parity is enabled by setting the Data Parity Enable (DP_EN) bit in the "Processor Bus Miscellaneous Control and Status Register" on page 361. Even parity or odd parity is enabled by setting the Parity (PARITY) bit in the same register.

Parity generation and checking is provided for each byte of the data bus and for each data beat of the data tenure. Data parity bit assignments are as defined in Table 24.

Table 24: PowerSpan II PB Data Parity Assignments

Data Bus	Data Parity
PB_D[0:7]	PB_DP[0]
PB_D[8:15]	PB_DP[1]
PB_D[16:23]	PB_DP[2]
PB_D[24:31]	PB_DP[3]
PB_D[32:39]	PB_DP[4]
PB_D[40:47]	PB_DP[5]
PB_D[48:55]	PB_DP[6]
PB_D[56:63]	PB_DP[7]

The data parity bits, PB_DP[0:7], are driven to the correct values for even or odd parity by the PB slave during reads and checked during writes.

The detection of a data parity error does not affect the transaction and data is still forwarded to the destination bus. See "Error Handling" on page 187 and "Interrupt Handling" on page 173 for a full description of error logging support and associated interrupt mapping options.

Special Parity Requirements with the MPC8260

Address parity and data parity must be specially programmed in a joint PowerSpan II and MPC8260 application.

In a joint application all memory accesses from the MPC8260 to PowerSpan II must be routed through the internal memory controller on the MPC8260. When the data is passed through the memory controller both address parity and data parity can be used in the system.

If accesses do not pass through the memory controller of the MPC8260 before reaching PowerSpan II, and PowerSpan II has either or both address and data parity enabled, then PowerSpan II reports parity errors on the transaction.

To enable or disable address parity in PowerSpan II, set the Address Parity Enable (AP_EN) bit in the Processor Bus Miscellaneous Control and Status (PB_MISC_CSR) register (see page 361).

To enable or disable data parity in PowerSpan II, set the Data Parity Enable (DP_EN) bit in the Processor Bus Miscellaneous Control and Status (PB_MISC_CSR) register (see page 361).

3.3.4 Terminations

3.3.4.1 PB Slave Termination

The PB slave uses the following pins to indicate termination of individual data beats and/or data tenure:

- Address Retry (PB_ARTRY_): This signal terminates the entire address and data tenure and schedules the transaction to be rerun. No data is transferred, even if asserted coincidentally with PB_TA/PB_DVAL_, as in the case of a third party address retry.
- Transfer Acknowledge (PB_TA_): This signal is asserted by the PowerSpan II PB Slave to indicate the successful transfer of a single beat transaction, or each 8-byte quantity transferred for a burst.

- Data Valid (PB_DVAL_): This signal is asserted by the PB slave to indicate the successful transfer of an 8-byte quantity within an extended transfer of 16 or 24 bytes. PB_TA_ is asserted together with PB_DVAL_ on the transfer of the last 8-byte quantity.
- Transfer Error Acknowledge (PB_TEA_): This signal indicates an unrecoverable error and causes the external master to immediately terminate the data tenure.

The PB Slave does not assert a data termination signal earlier than the *address retry window.*

3.3.4.2 Assertion of PB_TEA_

PowerSpan II asserts PB_TEA_ when a particular slave image cannot handle transactions involving more than 4 bytes. This applies to the following:

- register accesses (see "Register Access" on page 283)
- accesses to general purpose slave image configured for PCI I/O space
- access to registers designed to generate PCI Configuration or IACK commands (see "Configuration and IACK Cycle Generation" on page 295)

PowerSpan II also asserts PB_TEA_ if a read from PCI generates a Master-Abort or Target-Abort.

The assertion PB_TEA_ is enabled or disabled with the TEA Enable (TEA_EN) bit in the "Processor Bus Miscellaneous Control and Status Register" on page 361.

In a development environment, the TEA_EN bit is set to allow the assertion of PB_TEA_ to support the debug of software. In a production environment, customers may find it useful to disable the assertion of PB_TEA_.

3.3.4.3 Errors

The PowerSpan II PB Slave detects the following error conditions:

- address parity
- data parity on writes
- illegal accesses

See "Error Handling" on page 187 and "Interrupt Handling" on page 173 for a full description of error logging support and associated interrupt mapping options.

3.4 PB Master Interface

3.4.1 PowerSpan II as PB Master

The PowerSpan II becomes active as PB Master when:

- PowerSpan II is accessed as a PCI target
- one of the PowerSpan II DMA engines is processing a transfer

The operation of the PB Master is described by dividing a transaction into three different phases:

- Address Phase: This section discusses the arbitration for the address bus, and generation of the PB address and transfer types.
- Data Transfer: This section describes arbitration for the data bus, and control of transaction size and length.
- Terminations: This section describes the terminations supported by PowerSpan II, and exception handling.

3.4.2 Address Phase

3.4.2.1 Address Bus Arbitration and Tenure

The PB Master asserts Address Bus Busy (PB_ABB_) to indicate address bus ownership after it receives a qualified bus grant for its address bus request. A qualified bus grant assumes the following:

- address bus grant asserted
- PB_ARTRY_ negated
- address bus not busy

The PB Master negates PB_ABB_ for at least one clock after Address Acknowledge (PB_AACK_) has been asserted by the slave. This is true even if the arbiter parked the bus on PowerSpan II. For example, in Figure 13 on page 128 the bus is parked at the PowerSpan II (PB_BG[1]_ is asserted throughout), PB_ABB_ is negated the first positive clock edge after sampling PB_AACK_.

The PowerSpan II PB Master derives equivalent Address Bus Busy information from processor bus control signals. This allows the PowerSpan II processor bus arbiter to operate in 60x environments that do not implement ABB. The MPC8260 uses ABB to qualify address bus grants generated by the system arbiter.

The PB Master operates in a multi-processor, cache-coherent PowerPC environment that requires correct implementation of the *window of opportunity*. The following PB Master behavior supports the *window of opportunity*:

- respond to PB_ARTRY_ in the address retry window
- snoop PB_ARTRY_

Negating Address Bus Requests

When the PB Master (as current address bus owner) detects Address Retry (PB_ARTRY_) asserted during the address retry window, it negates its bus request for at least one clock. This guarantees the snooping master that retried the cycle an opportunity to request and be granted the bus before the PowerSpan II PB Master can restart its transaction. Once the bus is re-acquired, the PB Master restarts the transaction.

Cache Coherency

The Global (PB_GBL_) and Cache Inhibit (PB_CI_) parameters are programmable for each PCI target image and DMA channel (GBL and CI in the "PCI 1 Target Image x Control Register" on page 322 and the "DMA x Attributes Register" on page 376). Assertion of PB_GBL_ during a PB master transaction instructs all processors on the bus to snoop the transaction. Control of this parameter allows the user to implement non-coherent accesses in specific areas of memory. Assertion of PB_CI_ prohibits external agents from caching the transaction. This ability is useful in a system with an L2 look aside cache.

The PB Master, along with all other bus masters, are required to snoop ARTRY_ when they are not the bus owner. If ARTRY_ is asserted, the masters must ensure the following actions are taken:

- release bus request, if it is asserted, for at least one clock
- do not acquire the bus if presently granted
- do not assert bus request during the *window of opportunity*

To ensure a transaction is retried, systems assert PB_ARTRY_ at, or before, the first assertion of PB_TA_. This timing avoids a data tenure being terminated after data is transferred between bus agents. Normally, a retry scenario implies PB_ARTRY_ assertion one clock after assertion of the PB_AACK_ — in the address retry window. In certain systems however, the first assertion of PB_TA can occur before PB_AACK_. If this situation occurs, PB_ARTRY_ must be asserted at the same time as the first assertion of PB_TA_ and must be held until the clock after PB_AACK_ assertion.

Address Pipelining

The PB Master can operate in a system that implements up to one level of address pipelining. The PB Master does not prohibit other bus agents from pipelining transactions.

When mastering the bus, the PB Master can begin a new address tenure before the current data tenure completes.

Internal and External Arbitration

When the PowerSpan II processor bus arbiter is enabled (see “Arbitration” on page 163) all processor bus master address bus requests and grants are internal to PowerSpan II. When an external arbiter is used, the PB Master requests the address bus on PB_BR[1]_ and receives grants on PB_BG[1]_. For example, PowerSpan II’s internal arbiter is disabled in Figure 13 on page 128.

3.4.2.2 Address Translation

The address generated by the PB Master is dependent on the use of address translation in the source target image (see “PCI 1 Target Image x Control Register” on page 322. When address translation is enabled — TA_EN bit is set in the PCI target image— PowerSpan II produces the processor bus address using three inputs:

- the incoming address from the source bus
- the block size of the target image
- the translation offset

This does not apply to DMA transfers because the destination address is assumed to have the necessary offset by design.

For more information, see “DMA x Destination Address Register” on page 368.

3.4.2.3 Transaction type

The transfer type parameter of a PB master transaction, PB_TT, is specified with the PCI target image or DMA channel registers. The following registers control the parameter for write transactions:

- WTT[4:0] field in the “PCI 1 Target Image x Control Register” on page 322
- WTT[4:0] field in the “DMA x Attributes Register” on page 376

The following registers control the parameter for read transactions:

- RTT[4:0] field in the “PCI 1 Target Image x Control Register” on page 322
- RTT[4:0] field in the “DMA x Attributes Register” on page 376

The default transfer type generated by the PowerSpan II PB Interface master is shown in Table 25.

Table 25: Default PowerSpan II PB Master Transfer Type

PB Master Transaction	PB_TT[0:4]	Processor Bus Command
Writes	00010	Write with flush
Reads	01010	Read

3.4.2.4 Address Parity

Address parity generation is provided on each byte of the address bus. Address parity bit assignments are defined in Table 26

Table 26: PowerSpan II PB Address Parity Assignments

Address Bus	Address Parity
PB_A[0:7]	PB_AP[0]
PB_A[8:15]	PB_AP[1]
PB_A[16:23]	PB_AP[2]
PB_A[24:31]	PB_AP[3]

During PB master transactions, PB_AP[0:3] is driven to the correct values for either even parity or odd parity. Odd versus even parity is controlled with the PARITY bit in the "Processor Bus Miscellaneous Control and Status Register" on page 361.

3.4.3 Data Phase

3.4.3.1 Data Bus Arbitration and Tenure

The PB Master generates a data bus request by driving Transfer Type (PB_TT[3]) high during assertion of Transfer Start (PB_TS_). The PB Master asserts Data Bus Busy (PB_DBB_) to indicate data bus ownership when it receives a qualified bus grant (see Figure 13 on page 128). A qualified bus grant includes:

- Data Bus Grant (DBG) signal asserted

If the DBG signal is asserted past the data tenure of a transaction, the PB Master sees the assertion of the DBG signal as a new data tenure and re-asserts PB_DBB_.

- PB_ARTRY_ negated
- Data bus not busy

External slaves must not indicate a successful data transfer with the assertion of PB_TA_ and/or PB_DVAL_ earlier than two clocks after the assertion of PB_TS_. To ensure MPC8260 compliance with this rule, the MPC8260 register BCR[APD] must be programed to a value greater than one. This parameter specifies the earliest time after assertion of PB_TS_ that the MPC8260 slave asserts PB_TA_ to complete a data transfer. The BCR[APD] parameter is supported by the MPC8260 to accommodate processor bus agents with a range of snoop response times. BCR[APD] must be programmed to accommodate the slowest snooping device in the system.

The PB Master negates PB_DBB_ for at least one clock after the final data termination signal is asserted by the slave.

The PowerSpan II PB Master derives equivalent Data Bus Busy information from PB control signals. This allows the PowerSpan II processor bus arbiter to operate in processor bus environments that do not implement DBB. The MPC8260 uses DBB to qualify data bus grants generated by the system arbiter.

3.4.3.2 Transaction Length

The PB Master can generate a super-set of the data transfer sizes supported by the embedded PowerPC family. The user can disable certain data transfer sizes that are unique to the MPC8260. All data transfer sizes supported by the PowerSpan II PB master are illustrated in Table 27 below. Burst transfers are indicated by the assertion of Processor Bus Transfer Burst (PB_TBST_, see Figure 13 and Figure 14). PB_TBST_ is negated during single cycle transactions (see Figure 15 and Figure 16 below).

The shaded regions in Table 27 indicate transaction sizes that are unique to the MPC8260. The extended cycles supported by the MPC8260 are identified with an additional size pin, Processor Bus Transfer Size (PB_TSIZ[0]). Extended cycles are enabled using the EXTCYC bit in the "Processor Bus Miscellaneous Control and Status Register" on page 361.

PowerSpan II only interfaces to 64-bit slaves.

Table 27: PowerSpan II PB Transfer Sizes

Transfer Size	Bytes	PB_TBST	PB_TSIZ[0]	PB_TSIZ[1:3]
Byte	1	1	0	001
Half-word	2	1	0	010
Tri-byte	3	1	0	011
Word	4	1	0	100
Five bytes	5	1	0	101
Six bytes	6	1	0	110
Seven bytes	7	1	0	111
Double Word (DW)	8	1	0	000
Extended Double (MPC8260 only)	16	1	1	001
Extended Triple (MPC8260 only)	24	1	1	010
Burst (Quad DW)	32	0	0	010

The following figures, Figure 13 and Figure 14, illustrate burst reads and burst writes on the PB Master.

Figure 13: PB Master Interface Burst Read

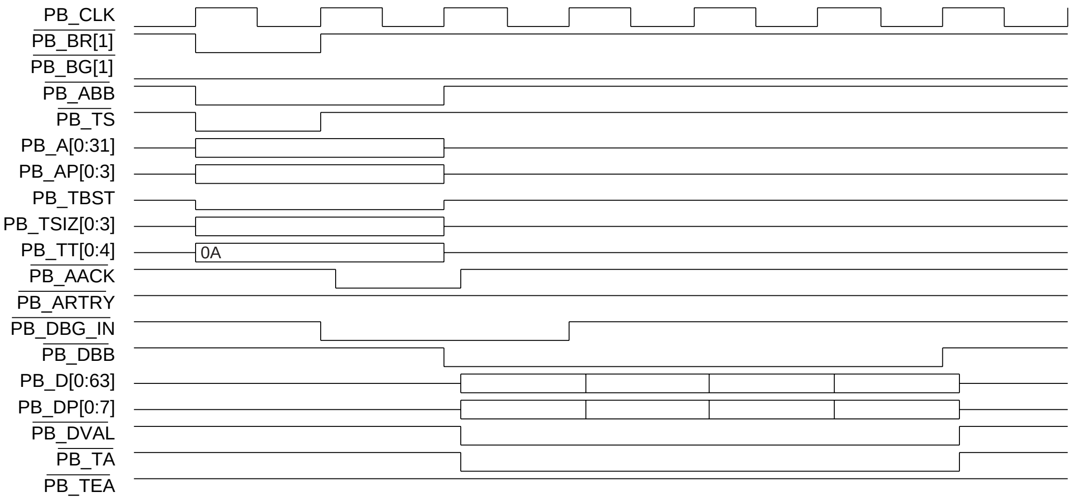
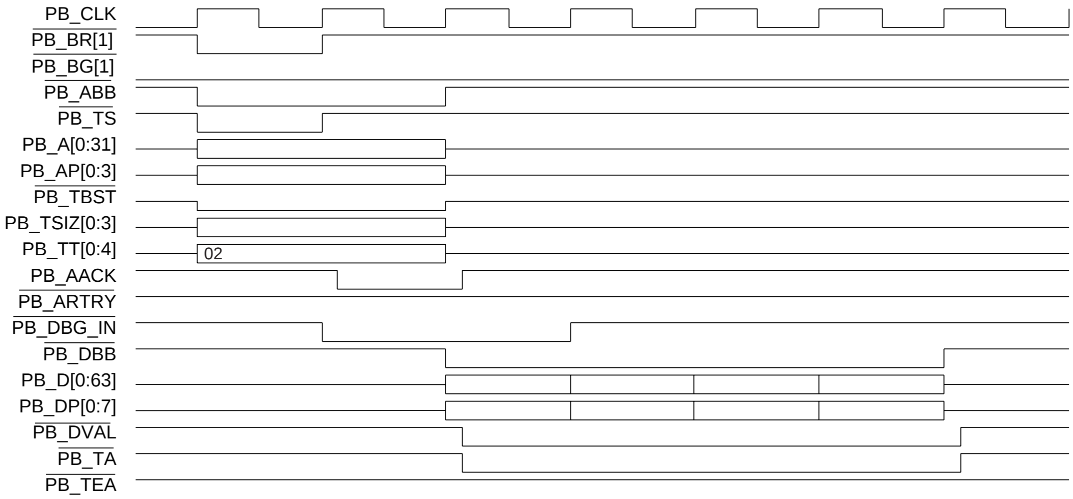


Figure 14: PB Master Interface Burst Write



The following figures, Figure 15 and Figure 16, illustrate single cycle read and single cycle write transfers on the PB Master Interface.

Figure 15: PB Master Interface Single Cycle Read

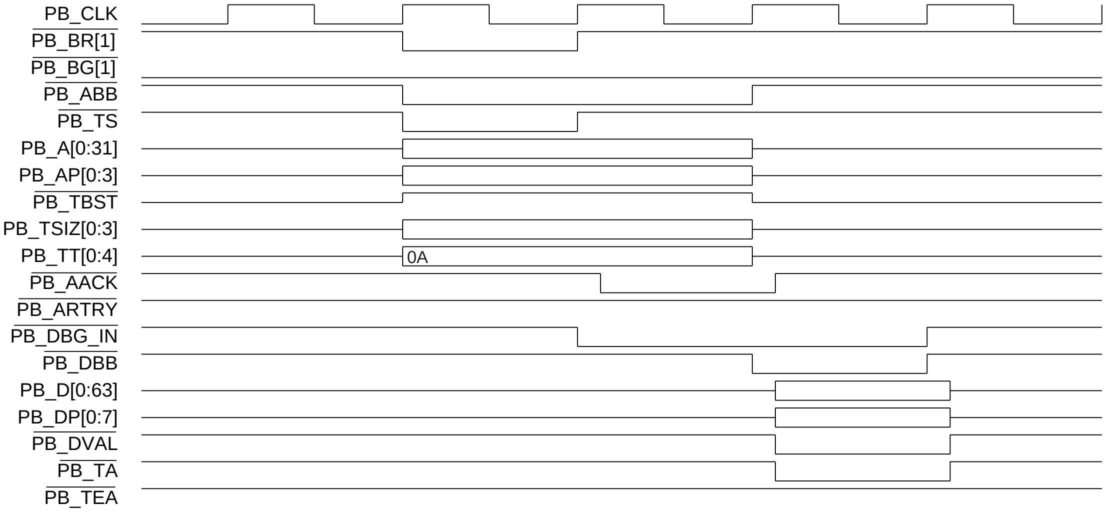
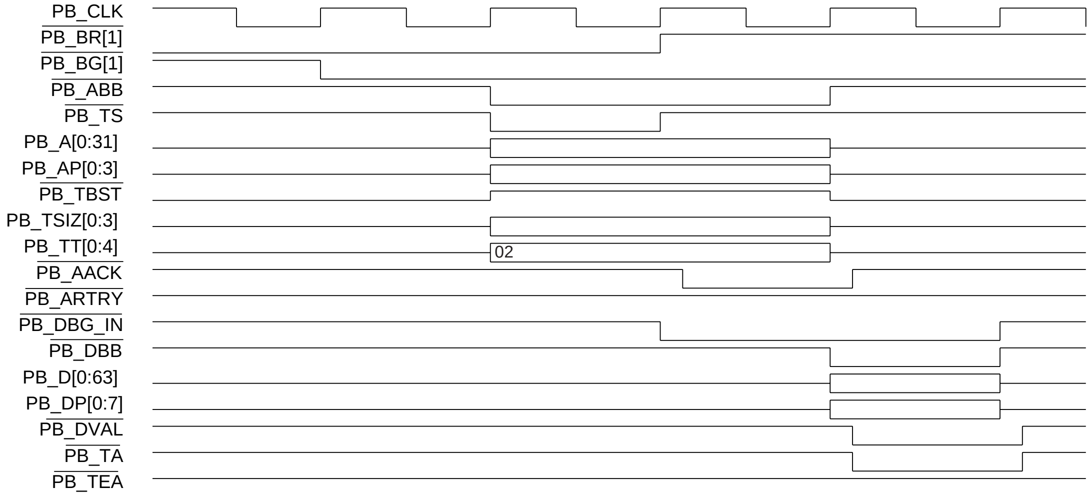


Figure 16: PB Master Interface Single Cycle Write



3.4.3.3 Data Alignment

Embedded processor bus transfer sizes and alignments are supported by the PowerSpan II PB Master for transaction accesses. The PB master creates the necessary sequence of transactions from a set of processor bus data size and alignment options. The size and alignment combinations defined in Table 29 are supported by the MPC8260, PowerPC 740, and PowerPC 750 processors. This set includes:

- transactions less than or equal to 8-bytes (single beat transactions)
- specific misaligned transactions
- extended transactions of 16 or 24-bytes
- burst of 32-bytes

Table 28 illustrates the lanes used to carry each byte of a multi-byte structure on a 64-bit processor data bus.

Table 28: 64-bit PB Data Bus Byte Lane Definitions

Byte Address	PB Byte Lanes			
PB_A[29:31]	Lane Number	PowerSpan II Pins	MPC8260 Pins	MPC740 Pins
000	0	PB_D[0:7]	D[0:7]	DH[0:7]
001	1	PB_D[8:15]	D[8:15]	DH[8:15]
010	2	PB_D[16:23]	D[16:23]	DH[16:23]
011	3	PB_D[24:31]	D[24:31]	DH[24:31]

Table 28: 64-bit PB Data Bus Byte Lane Definitions

Byte Address	PB Byte Lanes			
100	4	PB_D[32:39]	D[32:39]	DL[0:7]
101	5	PB_D[40:47]	D[40:47]	DL[8:15]
110	6	PB_D[48:55]	D[48:55]	DL[16:23]
111	7	PB_D[56:63]	D[56:63]	DL[24:31]

Table 29 lists the size and alignment transactions less than or equal to 8-bytes. The shaded table cells show transactions that support the PowerPC 7400 processor.

Table 29: PowerSpan II Processor Bus Single Beat Data Transfers

Size	TSIZ[0:3]	A[29:31]	Data Bus Byte Lanes							
			0	1	2	3	4	5	6	7
Byte	0001	000	D0							
	0001	001		D1						
	0001	010			D2					
	0001	011				D3				
	0001	100					D4			
	0001	101						D5		
	0001	110							D6	
	0001	111								D7
Half word	0010	000	D0	D1						
	0010	001		D1	D2					
	0010	010			D2	D3				
	0010	011				D3	D4			
	0010	100					D4	D5		
	0010	101						D5	D6	
	0010	110							D6	D7

Table 29: PowerSpan II Processor Bus Single Beat Data Transfers

Size	TSIZ[0:3]	A[29:31]	Data Bus Byte Lanes							
Tri-byte	0011	000	D0	D1	D2					
	0011	001		D1	D2	D3				
	0011	010			D2	D3	D4			
	0011	011				D3	D4	D5		
	0011	100					D4	D5	D6	
	0011	101						D5	D6	D7
Word	0100	000	D0	D1	D2	D3				
	0100	001		D1	D2	D3	D4			
	0100	010			D2	D3	D4	D5		
	0100	011				D3	D4	D5	D6	
	0100	100					D4	D5	D6	D7
Five bytes	0101	000	D0	D1	D2	D3	D4			
	0101	001		D1	D2	D3	D4	D5		
	0101	010			D2	D3	D4	D5	D6	
	0101	011				D3	D4	D5	D6	D7
Six bytes	0110	000	D0	D1	D2	D3	D4	D5		
	0110	001		D1	D2	D3	D4	D5	D6	
	0110	010			D2	D3	D4	D5	D6	D7
Seven bytes	0111	000	D0	D1	D2	D3	D4	D5	D6	
	0111	001		D1	D2	D3	D4	D5	D6	D7
Double word	0000	000	D0	D1	D2	D3	D4	D5	D6	D7

The information in Table 29 is independent of endian considerations and pertains to byte lane control on the processor bus. For endian considerations, please consult “Endian Mapping” on page 213.

PowerPC 7400 Transaction Support

The PowerPC 7400 processors supports misaligned transactions within a double word (64-bit aligned) boundary. As long as the transaction does not cross the double word boundary, the PowerPC 7400 can transfer data on the misaligned address.

PowerSpan II supports a specific types of the PowerPC 7400 misaligned transactions (shown in Table 29) when the MODE_7400 bit is set in the Processor Bus Miscellaneous Control and Status register (page 361). Any misaligned transaction between PowerSpan II and the MPC7400 that is a single word (32-bit) or less must be within a single word aligned boundary. Any transfer greater than a single word must start or end on a word boundary.

⚠ Software must make sure that the PowerPC 7400 does not initiate unsupported misaligned transactions to PowerSpan II.

3.4.3.4 Cache Line Size

The PowerPC processors supported by PowerSpan II implement a 32-byte cache line size (8 words). Cache wrap bursts are not generated because the PB master starts a burst transaction at a 32-byte aligned address. For a transaction that is not 32-byte aligned, the PB master utilizes one or more single beat or extended transaction size, to align to the cache line boundary, before generating the required burst transaction or transactions.

The PowerSpan II PB Master assumes all external slaves can accept burst transactions.

3.4.3.5 Data Parity

Data Parity is enabled by setting the DP_EN bit in the Processor Bus Control and Status register (see “Processor Bus Miscellaneous Control and Status Register” on page 361). Even or odd parity can be enabled by setting the PARITY bit on the Processor Bus Control and Status register.

Parity generation and checking is provided for each byte of the data bus and for each data beat of the data tenure. Data parity bit assignments are as defined in Table 30.

Table 30: PowerSpan II PB Data Parity Assignments

Data Bus	Data Parity
PB_D[0:7]	PB_DP[0]
PB_D[8:15]	PB_DP[1]
PB_D[16:23]	PB_DP[2]

Table 30: PowerSpan II PB Data Parity Assignments

Data Bus	Data Parity
PB_D[24:31]	PB_DP[3]
PB_D[32:39]	PB_DP[4]
PB_D[40:47]	PB_DP[5]
PB_D[48:55]	PB_DP[6]
PB_D[56:63]	PB_DP[7]

The data parity bits, PB_DP[0:7], are driven to the correct values for even or odd parity by the PB Master during writes. If checking is enabled (by setting the DP_EN bit) the data parity bits, PB_DP[0:7], are checked by the PB Master during reads. The detection of a data parity error does not affect the transaction, and data is still forwarded to the destination.

See “Error Handling” on page 187 and “Interrupt Handling” on page 173 for a full description of error logging support and associated interrupt mapping options.

3.4.4 Terminations

The PB master uses the following pins as termination signals for individual data beats and data tenure:

- Address Retry (PB_ARTRY_): This signal terminates the entire data tenure and schedules the transaction to be rerun. No data is transferred.
- Transfer Acknowledge (PB_TA_): This signal is asserted by the external slave to indicate the successful transfer of a single beat transaction, or each 8 byte quantity transferred for a burst.
- Data Valid (PB_DVAL_): This signal is asserted by the external slave to indicate the successful transfer of a quantity of data. The MPC8260 provides this pin to support the termination of extended cycles. The external slave asserts this pin once for each successful 8 byte transfer. PB_TA_ is asserted, with PB_DVAL_, on the final transfer of the transaction. The slave uses PB_TA_ and/or PB_DVAL_ to insert wait states. The PB master ignores PB_DVAL_ when the EXTCYC bit cleared in the “Processor Bus Miscellaneous Control and Status Register” on page 361.
- Transfer Error Acknowledge (PB_TEA): This signal indicates an unrecoverable error and causes the PB master to immediately terminate the data tenure.

3.4.4.1 Errors

The PB master detects three error conditions:

- data parity on reads
- assertion of PB_TEA_ by external slave
- expiration of maximum retry counter (MAX_RETRY bit in the "Processor Bus Miscellaneous Control and Status Register" on page 361).

See "Error Handling" on page 187 and "Interrupt Handling" on page 173 for a full description of error logging support and associated interrupt mapping options.

4. DMA

A direct memory access (DMA) channel allows a transaction to occur between two devices without involving the host processor (for example, a read transaction between a peripheral device and host processor memory). Because less time is required to complete transactions, applications that contain one or more DMA channels support faster read and write transfers than applications that support only host-assisted transactions.

discusses the following topics about the PowerSpan II DMA:

- "DMA Operation" on page 138
- "DMA Interrupts" on page 149
- "DMA Error Handling" on page 150

4.1 Overview

PowerSpan II has four identical Direct Memory Access (DMA) channels for independent data transfer between the three ports of the Dual PCI PowerSpan II: Processor Bus Interface (PB), PCI Interface 1 (PCI-1) and PCI Interface 2 (PCI-2). The programming and operation of the four DMAs are the same. This chapter discusses DMA operation within the context of a single channel. In addition, since the DMAs are able to transfer data from any port to any port, the DMA discussion refers to "source" bus and "destination" bus with no reference to bus type. Exceptions to this guideline are noted in the manual.

In the Single PCI PowerSpan II, the PCI-2 specific DMA bits must not be programmed. DMA transfers must not be directed to the PCI-2 Interface.

There are two modes of operation for the PowerSpan II DMA: Direct mode and Linked-List mode. In Direct mode, the DMA control registers are directly programmed for each DMA transfer — one start address and transfer size. In Linked-List mode, the PowerSpan II loads its DMA registers from a linked-list of “command packets”. The packets are essentially pre-programmed register contents for a PowerSpan II DMA channel.

4.2 DMA Operation

4.2.1 DMA Register Description

The DMA registers are the same for each DMA channel. DMA registers are described in Table 31. The registers for DMA1 begin at offset 0x300 and their organization in PowerSpan II register space is described in “Register Descriptions” on page 283.

Table 31: DMA Register Description

Register	Register Description and Operation
DMAx_SRC_ADDR	The Source Address Register can be programmed for an address on any one of the three PowerSpan II buses. This register can be programmed in Direct mode or automatically loaded in Linked-List mode. Writing to this register while the DMA is in operation has no effect. While the DMA is active, this register provides the current status of the source address. This address is byte-aligned.
DMAx_DST_ADDR	The Destination Address Register can be programmed for an address on any one of the three PowerSpan II buses — even the same bus as that used for the Source Address. This register can be programmed in Direct mode or automatically loaded in Linked-List mode. Writing to this register while the DMA is in operation has no effect. While the DMA is active, this register provides the status of the current destination address. This address is byte-aligned. The lower bits on the destination address are the same as the lower bits on the source address.
DMAx_TCR	The DMA Transfer Control Register specifies the source and destination buses, the endian conversion mode of a transfer involving the processor bus and a PCI bus (see “Endian Mapping” on page 213), and specifies the byte count from any remaining Direct mode operation.
DMAx_CPP	The DMA Command Packet Pointer register specifies the 32-byte aligned address of the next command packet in the linked-list. This is programmed by PowerSpan II as it loads a command packet. There is a LAST flag in this register to indicate the end of the linked-list.
DMAx_GCSR	The DMA General Control and Status Register controls DMA activity, reflects operational status and enables DMA-specific interrupts (see Table 32)
DMAx_ATTR	The DMA Attributes Register controls the transfer type and cache-specific behavior of processor bus transactions. It also selects the command packet port.

Most DMA channel registers are locked against any changes by the user while the channel is active. However, both the Stop Request (STOP_REQ) and Halt Request (HALT_REQ) bits, in the “DMA x General Control and Status Register” on page 372, are not locked.

4.2.1.1 Source and Destination Addresses

The lower three bits of DMA Destination Address register are taken directly from the lower three bits of the Source Address register. This enforces 8-byte alignment of the starting source and destination addresses. The source and destination address registers are part of the command packet contents (“Linked-List Mode DMA Operation” on page 145).

The starting byte address on the source port is specified in “DMA x Source Address Register” on page 367. The starting byte address on the Destination port is specified in “DMA x Destination Address Register” on page 368.

4.2.1.2 Transfer Control Register

The “DMA x Transfer Control Register” on page 369 details the programming options for this register. It controls the direction of the transfer, the endian conversion between the processor bus and the PCI bus and specifies the transfer byte count. Note that the maximum byte count is 16 Mbytes. The DMA Transfer Control Register is part of the command packet contents (“Linked-List Mode DMA Operation” on page 145).

In the Single PCI PowerSpan II, the PCI-2 specific DMA bits must not be programmed. DMA transfers must not be directed to the PCI-2 Interface.

4.2.1.3 Command Packet Addressing

The “DMA x Command Packet Pointer Register” on page 371 specifies the address for the command packets in Linked-List mode. See “Linked-List Mode DMA Operation” on page 145 for more details on command packet processing.

4.2.1.4 Address Retry

The Address Retry Enable (ARTRY_EN) bit in the “Processor Bus Miscellaneous Control and Status Register” on page 361 controls PowerSpan II’s assertion of PB_ARTRY_ during the servicing of transactions. When the ARTRY_EN bit is set to 0, the PB Slave is disabled from generating address retries.

DMA Addresses and Retries

If a PowerSpan II DMA transaction is retried enough times the its retry counter may expire. When the retry timer expires, the DMA transaction does not try to restart the transaction at the original address; it jumps the address. The new address starts at the nearest address boundary. The nearest address boundary depends on the value programmed in the DBS field (see "DMA x General Control and Status Register" on page 372). For example, the nearest address boundary for a incremented address when the DMA block size is set to 128 bytes is 0x80. In this case, the equation for the incremented address value is: original address + 0x80.

By advancing the address, PowerSpan II provides a method to step-out of the error condition.

4.2.1.5 General DMA Control and Status

The "DMA x General Control and Status Register" on page 372 is not part of the command packet contents and is set up prior to any DMA operation (Direct or Linked-List mode). The contents of the DMA General Control and Status Register are described in Table 32 below.

Table 32: Programming Model for DMA General Control and Status Register

Bits	Type	Description	Default Setting
GO	Write 1 to set	Initiates DMA activity	Clear
CHAIN	R/W	Enables the Linked-List mode of operation.	Disabled
STOP_REQ	Write 1 to set	Stops DMA operation after the internally buffered data is written out to the destination bus.	Clear
HALT_REQ	Write 1 to set	Halts DMA operation after the completion of the current command packet.	Clear
DACT	R	Provides status of DMA activity (active or inactive).	Clear
DBS[1:0]	R/W	Controls the byte size of DMA transactions when DBS_EN is set to 1.	Clear
DBS_EN	R/W	Enables byte size control of transactions generated by the DMA Channel. Transaction size is based on the setting of the DBS field.	Clear
OFF	R/W	DMA Channel Off Counter (number of PB clocks) Controls the number of Processor clocks between sequential PB tenures.	Clear

Table 32: Programming Model for DMA General Control and Status Register

Bits	Type	Description	Default Setting
P1_ERR	R/ Write 1 to clear	A status bit indicating an error has occurred on PCI-1.	Clear
P2_ERR	R/ Write 1 to clear	A status bit indicating an error has occurred on PCI-2. Disregard this bit with the Single PCI PowerSpan II.	Clear
PB_ERR	R/ Write 1 to clear	A status bit indicating an error has occurred on the processor bus.	Clear
STOP	R/ Write 1 to clear	A status bit indicating if the DMA has been stopped (STOP_REQ was set)	Clear
HALT	R/ Write 1 to clear	A status bit indicating if the DMA has been halted (HALT_REQ was set).	Clear
DONE	R/ Write 1 to clear	A status bit indicating if the DMA has been completed its Direct mode or Linked-List mode.	Clear
P1_ERR_EN	R/W	Enables an interrupt if an error occurs on PCI-1.	Disabled
P2_ERR_EN	R/W	Enables an interrupt if an error occurs on PCI-2. Do not program this bit if using the Single PCI PowerSpan II.	Disabled
PB_ERR_EN	R/W	Enables an interrupt if an error occurs on the processor bus.	Disabled
STOP_EN	R/W	Enables an interrupt if the DMA has been stopped (STOP_REQ bit was set).	Disabled
HALT_EN	R/W	Enables an interrupt if the DMA has been halted (HALT_REQ bit was set).	Disabled
DONE_EN	R/W	Enables an interrupt if the DMA completes its Direct mode or Linked-List mode.	Disabled

4.2.1.6 Processor Bus Transfer Attributes

The “DMA x Attributes Register” on page 376 controls the read and write transactions generated by the DMA as a processor bus master.

The default transfer type generated by the processor bus master is shown in Table 33.

Table 33: Default PowerSpan II PB Master Transfer Type

PB Master Transfer	PB_TT[0:4]	60x Command
Writes	00010	Write with flush
Reads	01010	Read

The Global (PB_GBL_) and Cache Inhibit (PB_CI_) parameters are programmable for each DMA in the“DMA x Attributes Register” on page 376. Assertion of PB_GBL_ during a processor bus master transaction instructs all processors on the bus to snoop the transaction. Control of this parameter enables the user to implement non-coherent accesses in specific areas of memory. Assertion of PB_CI_ prohibits external agents from caching the transaction.

4.3 Direct Mode DMA Operation

In Direct mode, the contents for all of a DMA channel registers are directly programmed into PowerSpan II before every DMA operation (see Table 31). This results in higher software overhead than in Linked-List mode since PowerSpan II register accesses are required for every DMA block transfer.

4.3.1 Initializing a Direct Mode Operation

The GO bit in the “DMA x General Control and Status Register” on page 372 is used when the following conditions are met:

- The CHAIN bit is zero which indicates a Direct mode operation
- All status bits in the DMA General Control and Status Register are cleared, including:

  P1_ERR

  P2_ERR

  PB_ERR

  STOP

  HALT

  DONE

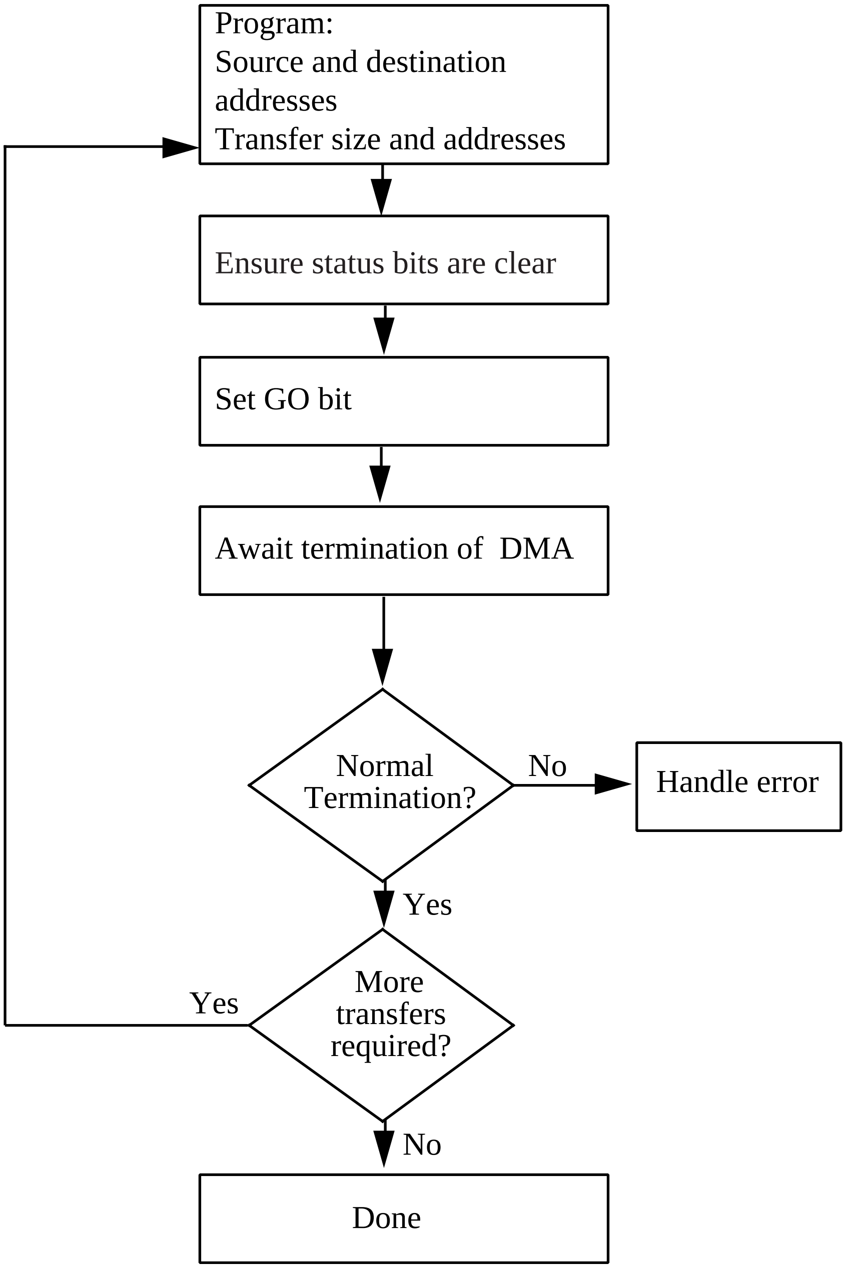
The CHAIN bit and status bits can be properly configured on the same register write which sets the GO bit.

The DMA channel delivers data from the source port to the destination port until:

- DMA is stopped by setting the STOP_REQ bit
- DMA encounters an error on one of the buses
- transfer byte count decrements to zero

When the Direct mode operation completes the programmed transfer, PowerSpan II sets the DONE bit. The operation completes when the transfer byte count decrements to zero. The operation of a Direct mode operation is illustrated in Figure 17.

Figure 17: Direct Mode DMA Transfers



4.3.1.1 Direct Mode Transfer Acknowledgment

The following registers are updated during a transfer and can be used to monitor status during DMA channel activity:

- DMA Source Address (DMAx_SRC_ADDR) in the "DMA x Source Address Register" on page 367
- DMA Destination Address (DMAx_DST_ADDR) in the "DMA x Destination Address Register" on page 368
- Byte Count (BC[23:0]) field in the "DMA x Destination Address Register" on page 368.

4.3.1.2 Terminating a Direct Mode Transfer

The current Direct mode transfer can be stopped by writing 1 to the STOP_REQ bit in the "DMA x General Control and Status Register" on page 372. When this occurs, the channel stops attempting to buffer data from the source bus. When the remaining buffered source data is written to the destination bus, the STOP status bit is set.

The channel can be restarted by clearing the STOP status bit (along with any other status bits) and then writing a 1 to the GO bit.

Due to the pipelined nature of DMA channel requests, up to 256-bytes can be transferred after the user programmed the initial stop request.

4.4 Linked-List Mode DMA Operation

In Linked-List (scatter-gather) mode, PowerSpan II steps through a linked series of command packets in external memory. The DMA is configured with the starting address of this list and independently reads command packets and executes the transfers specified.

Each command packet is 32-byte aligned. If the command packets are resident in PCI memory, the byte ordering is little-endian. If the command packets are resident in processor bus memory, the byte ordering is big-endian. command packets can reside on any one of the three PowerSpan II interfaces. The contents of a command packet are described, with the associated DMA register, in Table 34.

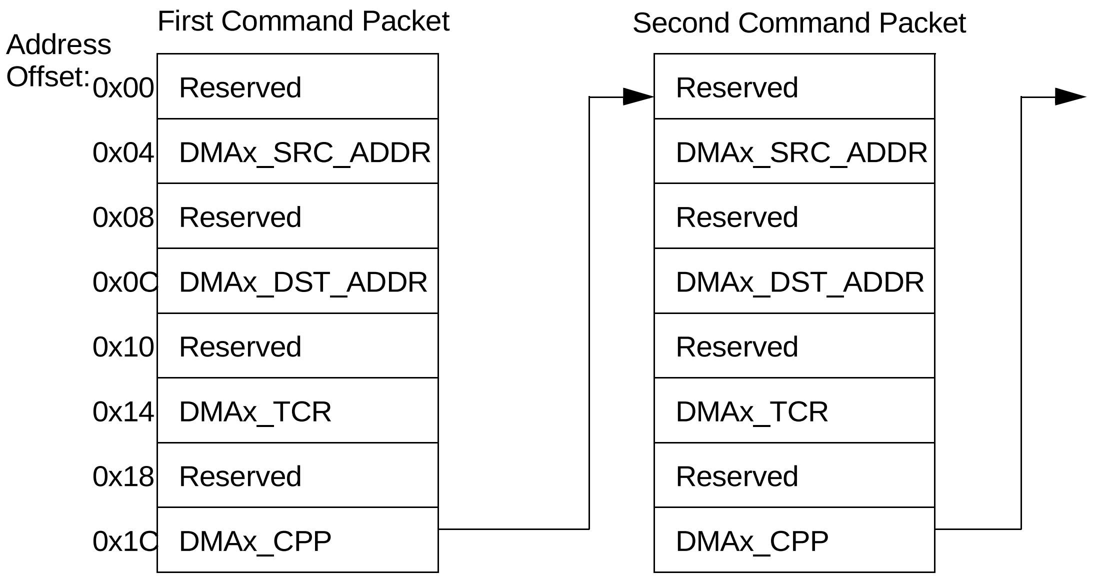
Table 34: Programming Model for the Command Packet Contents

Register	Register Description and Operation
DMAx_SRC_ADDR	The Source Address Register can be programmed for an address on any one of the three PowerSpan II buses. This register can be programmed in Direct mode or automatically loaded in Linked-List mode. Writing to this register while the DMA is in operation has no effect. While the DMA is active, this register provides the status on the current source address. This address is byte-aligned.
DMAx_DST_ADDR	The Destination Address Register can be programmed for an address on any one of the three PowerSpan II buses (including the same bus as that used for the Source Address). This register can be programmed in Direct mode or automatically loaded in Linked-List mode. Writing to this register while the DMA is in operation has no effect. While the DMA is active, this register provides the status on the current destination address. This address is byte-aligned. The lower bits on the destination address are the same as the lower bits on the source address.
DMAx_TCR	The DMA Transfer Control Register specifies the source and destination buses, the endian conversion mode of a transfer involving the Processor Bus and a PCI bus (see "Endian Mapping" on page 213), and specifies the byte count from any remaining Direct mode operation.
DMAx_CPP	The DMA Command Packet Pointer register specifies the 32-byte aligned address of the next command packet in the linked-list. This is programmed by PowerSpan II as it loads a command packet. There is a LAST flag in this register to indicate the end of the linked-list.
DMAx_ATTR	The DMA Attributes Register specifies to the channel the location of the linked-list port.

The Command Packet Pointer register (DMAx_CPP) contains two elements: the Next Command Packet Pointer (NCP[31:5]) and the LAST bit. The NCP[31:5] field directs the PowerSpan II DMA to the next command packet in the linked-list. The LAST bit indicates the end of the linked-list.

The chaining of the command packets is illustrated in Figure 18.

Figure 18: DMA Command Packet Linked-List



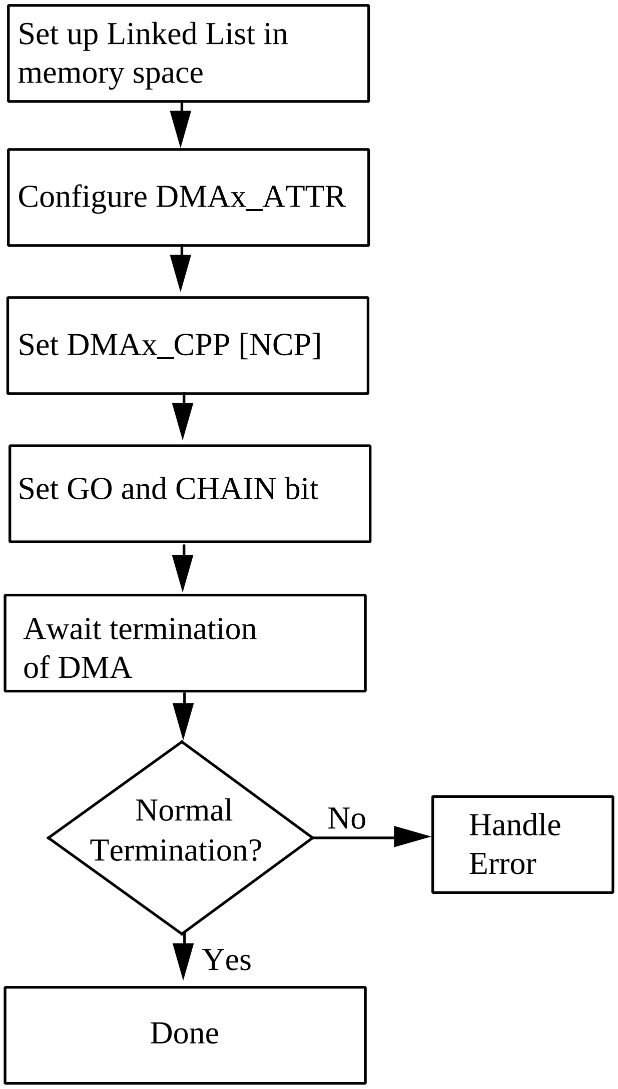
4.4.1 Initializing a Linked-List Mode Transfer

A Linked-List mode DMA transfer is configured using the following steps:

1. Set-up the command packet linked-list in memory accessible to any one of the PowerSpan II’s three ports. The command packet port selection is independent of the port selected as the source or destination port.
2. Configure DMAx_ATTR parameters and DMAx_GCSR.
3. Set-up the NCP[31:5] field to point to the first command packet.
4. Ensure the BC[23:0] field in the DMA Transfer Control Register is 0.
5. Clear all status bits in the DMA General Control and Status Register.
6. Set the GO bit.

The steps to configure a Linked-List mode DMA transfer are illustrated in Figure 19.

Figure 19: Sequence of Operations in a Linked-List Transfer



The DMA walks through the linked-list of command packets until it executes the last packet. When the operation programmed with that last Command Packet is completed, the DMA sets the DONE bit. The LAST bit indicates the end of the linked-list.

If the Linked-List mode is started with a non-zero byte count in the DMA Transfer Control Register, a Direct mode DMA transfer is initiated by PowerSpan II to clear the remaining byte count value. Once that Direct mode transfer is complete, the DMA then processes the linked-list pointed to in the DMA Command Packet Pointer Register. This mechanism allows the restart of a linked-list transfer that has been stopped with the STOP_REQ bit in the "DMA x General Control and Status Register" on page 372.

4.4.1.1 Terminating Linked-List Mode

Linked-List mode is terminated in two ways: setting the STOP_REQ bit or the HALT_REQ bit in the "DMA x General Control and Status Register" on page 372. When the STOP_REQ bit is set, the DMA stops making source port requests. When all outstanding transactions are completed on the destination bus, the STOP status bit is set. The channel can be restarted by clearing the STOP status bit (along with any other status bits) and then writing a one to the GO bit.

Processing of the current linked-list can also be halted by setting the HALT_REQ bit. If this bit is set, transfers specified by the current command packet are completed and then the DMA sets the HALT status bit. Since DMAx_CPP contains the address of the next command packet, the channel can be restarted by writing 1 to HALT bit (to clear the HALT state), the CHAIN bit (to re-initiate the Linked-List mode), and the GO bit (to re-activate the DMA).

4.5 DMA Interrupts

The PowerSpan II DMA supports a number of interrupt sources for each channel. Individual enable and status bits exist for each source. The status and enable bits are contained in the "DMA x General Control and Status Register" on page 372:

Table 35: DMA Channel Interrupt Sources and Enables

Interrupt Source	Enable bit
DONE	DONE_EN
P1_ERR	P1_ERR_EN
P2_ERR	P2_ERR_EN
PB_ERR	PB_ERR_EN
HALT	HALT_EN
STOP	STOP_EN

See "Interrupt Handling" on page 173 for a complete description of the mapping and status bits for each of these interrupt sources.

4.6 DMA Error Handling

PowerSpan II can encounter external bus errors while mastering the source, destination or command packet ports on behalf of a DMA channel. Each DMA channel provides the following status bits in the "DMA x General Control and Status Register" on page 372 that indicate an error condition occurred during DMA bus master activity:

- P1_ERR
- P2_ERR
- PB_ERR

In addition to the reporting provided by a DMA channel, the participating Master also reports the error. See "Error Handling" on page 187.

4.6.1 PCI Error Bits

The error bits for PCI-1 and PCI-2 are set when the corresponding PowerSpan II PCI Master encounters one of the following conditions while servicing a DMA channel:

- Master-Abort
- Target-Abort
- Maximum retry limit is reached

4.6.2 Processor Bus Error Bit

The error bit for the processor bus is set if the PowerSpan II PB Master encounters one of the following conditions while servicing a DMA channel.

- assertion of PB_TEA_
- maximum retry limit is reached

The occurrence of data parity error does not affect DMA channel behavior but is captured by the appropriate interface.

4.6.3 Source Port Errors

When an error occurs on the source port, transactions initiated by the source port are terminated. Any source data buffered in the PowerSpan II is written to the destination port and the appropriate DMAx_GCSR error bit is set.

Due to the pipelined nature of DMA channel requests, additional Source port transaction activity may occur until all outstanding channel requests are completed.

4.6.4 Destination Port Errors

When an error occurs on the destination port transactions associated with any buffered data are terminated, and the appropriate DMAx_GCSR error bit is set.

Due to the pipelined nature of DMA channel requests, additional destination port transaction activity can occur until all outstanding channel requests are completed.

4.6.5 Command Port Errors

When an error occurs on the command port the appropriate DMAx_GCSR error bit is set. The DMA channel registers are not updated with command packet data.

Each PowerSpan II external port has error log registers that provides additional diagnostic information to assist in error recovery. These error log registers indicate when multiple errors occur due to the pipelined nature of DMA channel requests.

See “Error Handling” on page 187 and “Interrupt Handling” on page 173 for a full description of error logging support and associated interrupt mapping options.

5. I^{2}C/EEPROM

The I^{2}C (Inter-IC) bus is a bi-directional, two-wire serial data and serial clock bus that provides communication links between integrated circuits (ICs) in an embedded application. Each device is recognized by a unique address and can operate as either a receiver device (for example, an LCD driver), or a transmitter device (for example, EEPROM) with the capability to both receive and send information. Transmitters and receivers can operate in either master or slave mode, depending on whether the IC initiates data transfers.

This chapter discusses the following topics about the PowerSpan II I^{2}C/EEPROM Interface:

- “Power-Up Configuration” on page 154
- “Bus Master I^{2}C Transactions” on page 161
- “PCI Vital Product Data (VPD)” on page 161

5.1 Overview

PowerSpan II has a master only, I^{2}C bus compatible interface which supports up to eight I^{2}C slave devices. This interface is primarily used by PowerSpan II for the initialization of registers and for reading and writing PCI Vital Product Data (VPD). However, PowerSpan II also provides a mechanism for processor bus and PCI masters to access the I^{2}C devices.

PowerSpan II does not support multiple masters on the same I^{2}C bus.

PowerSpan II I^{2}C Interface supports the following features:

- I^{2}C 7-bit device addressing

- Standard mode (up to 100 Kbits/s)
- Single read/write (random read, byte write)
- Sequential read during power-up configuration

The interface consists of two pins: I2C_SDA and I2C_SCLK. I2C_SDA is a bidirectional open drain signal for transferring address, control, and data bits. I2C_SCLK is the clock output for the I^2C slave devices. I2C_SCL is derived from the processor bus clock. For example, at the maximum Processor Bus Clock (PB_CLK) frequency of 100 MHz, the I2C_SCLK clock rate is 100 kHz.

5.2 Power-Up Configuration

At the end of each PowerSpan II reset sequence, the I^2C Interface initiates a sequential read with device select code 0b1010000. If no response is detected, the read is terminated and the EEPROM Load bit (ELOAD), in the “Reset Control and Status Register” on page 385 (RST_CSR), is cleared to indicate the absence of an external EEPROM.

When an EEPROM is not used in the system for initialization, the initialization occurs from the processor bus. Once initialization is complete, the P1_LOCKOUT and P2_LOCKOUT bits must be cleared in the “Miscellaneous Control and Status Register” on page 378 (MISC_CSR) to enable the host processor to assign memory space.

When a EEPROM is used in a system, the EEPROM device responds and a number of PowerSpan II register bits are loaded from the external device and the ELOAD bit is set. During this loading process, all accesses to PowerSpan II’s external interfaces are retried.

5.2.1 EEPROM Loading

When the reset sequence is initiated by assertion of PO_RST_ — a power-up reset — the register loading process is defined by Table 36. The first byte read from the EEPROM defines the loading option and is reflected in the EEPROM Load Option (ELOAD_OPT) field, in the “Miscellaneous Control and Status Register” on page 378, at the conclusion of the loading process.

The loading options for EEPROM are short loading and long loading. The short load consists of 29 bytes and is designed to provide a PowerSpan II configuration to support the absence of a processor on the PB Interface. The long load is 61 bytes in length and provides additional configuration convenience. The upper 192 bytes of the EEPROM are reserved for PCI Vital Product Data (see “PCI Vital Product Data (VPD)” on page 161).

Table 36 defines the power-up EEPROM load sequence. The shaded areas indicate registers not visible in the Single PCI PowerSpan II.

Table 36 assumes PCI little-endian bit ordering. Consult the register tables for each of the registers listed in the table to obtain the corresponding PowerPC big-endian bit ordering.

Table 36: Power-up EEPROM Load Sequence

Byte Offset	Bit	Name	Description
0x00	7-0	MISC_CSR[ELOAD_OPT]	0b00000001=short load 0b00000010=long load 0b00000100=reserved others=do not load
0x01	7-0	PowerSpan II Reserved	
0x02	7-0	PowerSpan II Reserved	
0x03	7-0	PowerSpan II Reserved	
Start of short load			
0x04	7-0	PowerSpan II Reserved	
0x05	7-4	PowerSpan II Reserved	
	3	P1_CSR[BM]	PCI-1 Bus Master Enable
	2	P1_CSR[MS]	PCI-1 Memory Space Enable
	1	P2_CSR[BM]	PCI-2 Bus Master Enable
	0	P2_CSR[MS]	PCI-2 Memory Space Enable
0x06	7-5	PowerSpan II Reserved	
	4	P1_BSI2O[PRFTCH]	PCI-1 I2O target image prefetch indicator
	3	P1_BST0[PRFTCH]	PCI-1 Target image 0 prefetch indicator
	2	P1_BST1[PRFTCH]	PCI-1 Target image 1 prefetch indicator
	1	P1_BST2[PRFTCH]	PCI-1 Target image 2 prefetch indicator
	0	P1_BST3[PRFTCH]	PCI-1 Target image 3 prefetch indicator
0x07	7-0	P1_SID[SID[15:8]]	PCI-1 Subsystem ID bits 15-8
0x08	7-0	P1_SID[SID[7:0]]	PCI-1 Subsystem ID bits 7-0

Table 36: Power-up EEPROM Load Sequence

Byte Offset	Bit	Name	Description
0x09	7-0	P1_SID[SVID[15:8]]	PCI-1 Subsystem vendor ID bits 15-8
0x0A	7-0	P1_SID[SVID[7:0]]	PCI-1 Subsystem vendor ID bits 7-0
0x0B	7-2	PowerSpan II Reserved	
	1	P1_MISC1[INT_PIN[0]]	PCI-1 Interrupt pin bit 0
	0	P2_MISC1[INT_PIN[0]]	PCI-2 Interrupt pin bit 0
0x0C	7-5	PowerSpan II Reserved	
	4	P1_MISC_CSR[BSREG_BAR_EN]	PCI-1 Register image base address register enable
	3	P1_TI0_CTL[BAR_EN]	PCI-1 Target image 0 base address register enable
	2	P1_TI1_CTL[BAR_EN]	PCI-1 Target image 1 base address register enable
	1	P1_TI2_CTL[BAR_EN]	PCI-1 Target image 2 base address register enable
	0	P1_TI3_CTL[BAR_EN]	PCI-1 Target image 3 base address register enable
0x0D	7-4	P1_TI0_CTL[BS]	PCI-1 Target image 0 block size
	3-0	P1_TI1_CTL[BS]	PCI-1 Target image 1 block size
0x0E	7-4	P1_TI2_CTL[BS]	PCI-1 Target image 2 block size
	3-0	P1_TI3_CTL[BS]	PCI-1 Target image 3 block size
0x0F	7	MISC_CSR[VPD_EN]	PCI Vital Product Data enable
	6-4	MISC_CSR[VPD_CS[2:0]]	PCI Vital Product Data chip select
	3-0	PowerSpan II Reserved	

Table 36: Power-up EEPROM Load Sequence

Byte Offset	Bit	Name	Description
0x10	7	MISC_CSR[P1_LOCKOUT]	PCI-1 Lockout
	6	MISC_CSR[P2_LOCKOUT]	PCI-2 Lockout
	5-4	PowerSpan II Reserved	
	3	MISC_CSR[PCI_ARB_CFG]	PCI Arbiter Configuration Complete
	2	MISC_CSR[PCI_M7]	PCI Arbiter Master 7
	1	MISC_CSR[PCI_M6]	PCI Arbiter Master 6
	0	MISC_CSR[PCI_M5]	PCI Arbiter Master 5
0x11	7	IDR[P2_HW_DIR]	$\overline{\text{P2\_INTA}}$ Direction
	6	IDR[P1_HW_DIR]	$\overline{\text{P1\_INTA}}$ Direction
	5	IDR[INT5_HW_DIR]	$\overline{\text{INT}}$[5] Direction
	4	IDR[INT4_HW_DIR]	$\overline{\text{INT}}$[4] Direction
	3	IDR[INT3_HW_DIR]	$\overline{\text{INT}}$[3] Direction
	2	IDR[INT2_HW_DIR]	$\overline{\text{INT}}$[2] Direction
	1	IDR[INT1_HW_DIR]	$\overline{\text{INT}}$[1] Direction
	0	IDR[INT0_HW_DIR]	$\overline{\text{INT}}$[0] Direction
0x12	7-6	PowerSpan II Reserved	
	5	PCI_I2O_CTL[BAR_EN]	PCI I$_2$O Target image base address register enable
	4	PowerSpan II Reserved	
	3-0	PCI_I2O_CTL[BS]	PCI I$_2$O Target image block size
0x13	7-5	PowerSpan II Reserved	
	4	P2_BSI2O[PRFTCH]	PCI-2 I2O target image prefetch indicator
	3	P2_BST0[PRFTCH]	PCI-2 Target image 0 prefetch indicator
	2	P2_BST1[PRFTCH]	PCI-2 Target image 1 prefetch indicator
	1	P2_BST2[PRFTCH]	PCI-2 Target image 2 prefetch indicator
	0	P2_BST3[PRFTCH]	PCI-2 Target image 3 prefetch indicator

Table 36: Power-up EEPROM Load Sequence

Byte Offset	Bit	Name	Description
0x14	7-0	P2_SID[SID[15:8]]	PCI-2 Subsystem ID bits 15-8
0x15	7-0	P2_SID[SID[7:0]]	PCI-2 Subsystem ID bits 7-0
0x16	7-0	P2_SID[SVID[15:8]]	PCI-2 Subsystem vendor ID bits 15-8
0x17	7-0	P2_SID[SVID[7:0]]	PCI-2 Subsystem vendor ID bits 7-0
0x18	7-5	PowerSpan II Reserved	
	4	P2_MISC_CSR[BSREG_BAR_EN]	PCI-2 Register image base address register enable
	3	P2_TI0_CTL[BAR_EN]	PCI-2 Target image 0 base address register enable
	2	P2_TI1_CTL[BAR_EN]	PCI-2 Target image 1 base address register enable
	1	P2_TI2_CTL[BAR_EN]	PCI-2 Target image 2 base address register enable
	0	P2_TI3_CTL[BAR_EN]	PCI-2 Target image 3 base address register enable
0x19	3-0	P2_TI0_CTL[BS]	PCI-2 Target image 0 block size
	3-0	P2_TI1_CTL[BS]	PCI-2 Target image 1 block size
0x1A	3-0	P2_TI2_CTL[BS]	PCI-2 Target image 2 block size
	3-0	P2_TI3_CTL[BS]	PCI-2 Target image 3 block size
0x1B-0x1F		PowerSpan II Reserved	
End of short load, long load continues			
0x20	7-0	P1_ID[DID[15:8]]	PCI-1 Device ID bits 15-8
0x21	7-0	P1_ID[DID[7:0]]	PCI-1 Device ID bits 7-0
0x22	7-0	P1_ID[VID[15:8]]	PCI-1 Vendor ID bits 15-8
0x23	7-0	P1_ID[VID[7:0]]	PCI-1 Vendor ID bits 7-0
0x24	7-0	P1_CLASS[BASE]	PCI-1 Base Class Code
0x25	7-0	P1_CLASS[SUB]	PCI-1 Sub Class Code
0x26	7-0	P1_CLASS[PROG]	PCI-1 Programming Interface

Table 36: Power-up EEPROM Load Sequence

Byte Offset	Bit	Name	Description
0x27	7-0	P1_CLASS[RID]	PCI-1 Revision ID
0x28		PowerSpan II Reserved	
	6	PB_SI0_CTL[TA_EN]	PB Slave image 0 translation enable
	5	PB_SI0_CTL[MD_EN]	PB Slave image 0 Master decode enable
	4-0	PB_SI0_CTL[BS]	PB Slave image 0 block size
0x29	7	PB_SI0_CTL[MODE]	PB Slave image 0 image select
	6	PB_SI0_CTL[DEST]	PB Slave image 0 destination
	5-0	PowerSpan II Reserved	
0x2A	7	PB_SI0_CTL[PRKEEP]	PB Slave image 0 image prefetch read keep
	6-5	PB_SI0_CTL[END]	PB Slave image 0 image endian conversion
	4-3	PowerSpan II Reserved	
	2-0	PB_SI0_CTL[RD_AMT]	PB Slave image 0 Read Prefetch Amount
0x2B	7-0	PB_SI0_TADDR[31:24]	PB Slave image 0 translation address bits 31-24
0x2C	7-0	PB_SI0_TADDR[23:16]	PB Slave image 0 translation address bits 23:16
0x2D	7-4	PB_SI0_TADDR[15:12]	PB Slave image 0 translation address bits 15-12
	3	PB_SI0_TADDR[M3]	PB Slave image 0 master 3 select
	2	PB_SI0_TADDR[M2]	PB Slave image 0 master 2 select
	1	PB_SI0_TADDR[M1]	PB Slave image 0 master 1 select
	0	PowerSpan II Reserved	
0x2E	7-0	PB_SI0_BADDR[31:24]	PB Slave image 0 base address bits 31-24
0x2F	7-0	PB_SI0_BADDR[23:16]	PB Slave image 0 base address bits 23-16
0x30	7-4	PB_SI0_BADDR[15:12]	PB Slave image 0 base address bits 15-12
	3-0	PowerSpan II Reserved	
0x31	7-0	PB_REG_ADDR[31:24]	PB Slave register image base address bits 31-24

Table 36: Power-up EEPROM Load Sequence

Byte Offset	Bit	Name	Description
0x32	7-0	PB_REG_ADDR[23:16]	PB Slave register image base address bits 23-16
0x33	7-4	PB_REG_ADDR[15:12]	PB Slave register image base address bits 15-12
	3-2	PowerSpan II Reserved	
	0	PB_REG_ADDR[END]	PB Slave register image endian conversion
0x34	7-0	P2_ID[DID[15:8]]	PCI-2 Device ID bits 15-8
0x35	7-0	P2_ID[DID[7:0]]	PCI-2 Device ID bits 7-0
0x36	7-0	P2_ID[VID[15:8]]	PCI-2 Vendor ID bits 15-8
0x37	7-0	P2_ID[VID[7:0]]	PCI-2 Vendor ID bits 7-0
0x38	7-0	P2_CLASS[BASE]	PCI-2 Base Class Code
0x39	7-0	P2_CLASS[SUB]	PCI-2 Sub Class Code
0x3A	7-0	P2_CLASS[PROG]	PCI-2 Programming Interface
0x3B	7-0	P2_CLASS[RID]	PCI-2 Revision ID
0x3C-0x3F	7-0	PowerSpan II Reserved	
End of load sequence			
0x40-0xFF	7-0	Reserved for PCI Vital Product Data (VPD)	

⚠ When a long EEPROM load is executed, the PB Slave Image 0 is enabled automatically. The IMG_EN bit is set to 1 in the PB_S1_CTL register.

When the reset sequence is initiated by assertion of PB_RST_, P1_RST# or P2_RST#, the first byte of the EEPROM is read to determine the loading sequence desired. All bytes for the selected load option are read from the EEPROM, but only a subset of PowerSpan II registers are updated. This subset is defined by the external reset pin that initiated the reset sequence. Only those register bits affected by the active reset pin(s) are updated with EEPROM contents. See "Register Descriptions" on page 283 for more information.

5.3 Bus Master I^{2}C Transactions

I^{2}C master reads and writes can be performed from any one of the PowerSpan II's three interfaces — PB, PCI-1 or PCI-2. These I^{2}C transactions are generated by accessing the "I2C/EEPROM Interface Control and Status Register" on page 383. This register can be used to access EEPROMs or perform arbitrary single byte transfers to other I^{2}C compatible devices. Since the I^{2}C Interface is a shared resource, software must use a PowerSpan II register semaphore, SEMAx, to acquire exclusive access to interface before initiating transactions with I2C_CSR. The I2C_CSR register contains the following fields:

- EEPROM Address (ADDR): The 8-bit EEPROM address specifies the address for byte writes and random reads.
- Data (DATA): The 8-bit data field is the source for writes and destination for reads.
- Device Code (DEV_CODE): Device code is the 4-bit field that specifies the I^{2}C device type. The default is 1010b which is the code for EEPROMs.
- Chip Select (CS): Chip select is the 3-bit field use to select one of the eight slaves on the I^{2}C bus. The device code and chip select fields together form the I^{2}C 7-bit device address.
- Read/Write (RW)
- Active (ACT): When the active bit is set, a transfer is in progress and the register is in read-only mode. After performing a write or read access, the user must poll the active bit until it is negated before performing other transfers. The active bit is also asserted during power-up EEPROM load and when a PCI Vital Product Data transfer is in progress.
- Error (ERR): If the PowerSpan II is unable to complete an I^{2}C access, the ERR bit is set when the ACT is negated. The ERR bit must be cleared before attempting another access.

5.4 PCI Vital Product Data (VPD)

Vital Product Data (VPD) is the information that uniquely defines items such as the hardware, software, and microcode elements of a system. VPD also provides a mechanism for storing information such as performance and failure data on a device.

VPD resides in a local storage device. PowerSpan II supports VPD through the serial EEPROM. If an external EEPROM is not used, the VPD feature is disabled.

There are four bits in Table 36 associated with PCI Vital Product Data: VPD_EN and VPD_CS[2:0]. These bits may also be programmed in the “Miscellaneous Control and Status Register” on page 378.

When VPD_EN is set, PowerSpan II supports PCI Vital Product Data through the VPD capabilities registers in the PCI Configuration Space of the designated Primary PCI Interface (see “Primary PCI” on page 44). The VPD may be located in two different places: the upper 192 bytes of the first EEPROM (VPD_CS=000) or the entire 256 bytes of a second EEPROM with chip select VPD_CS. The VPD Enable and VPD Chip Select fields in the “Miscellaneous Control and Status Register” on page 378 are initialized as part of the short load post reset sequence.

If VPD is located in the first EEPROM, the first byte is located at offset 0x40. On every VPD transfer, PowerSpan II adds the offset 0x40 to the address in the P1_VPDC or P2_VPDC register. If VPD is not located in the first EEPROM, then the address in the P1_VPDC or P2_VPDC register is used directly as the 8 bit EEPROM address.

Accesses to Vital Product Data in external EEPROM is performed in the manner described in “Bus Master I^{2}C Transactions” on page 161.

The bit ordering of the data returned from EEPROM in the “PCI 1 Vital Product Data Register” on page 321 can be addressed according to little-endian or big-endian conventions. See the bit ordering information in the register table to obtain the necessary bit ordering information.

6. Arbitration

Arbitration is a process used by multi-drop bus protocols, such as PCI, to support read and write access on a peripheral bus. A bus arbiter is a logic module that controls access to the bus by the devices residing on it. For example, when a device requires access to the bus it sends a request signal to the bus arbiter. If the bus is not active, the arbiter grants the device access; otherwise, the device must continue to request access until it is successful in obtaining the bus.

This chapter discusses the following topics about PowerSpan II’s Processor Bus and PCI arbitration capabilities:

- “PCI Interface Arbitration” on page 163
- “Processor Bus Arbitration” on page 168

6.1 Overview

PowerSpan II has three arbiters. There is an arbiter on each PCI interface: PCI-1 and PCI-2. There is also one arbiter for the Processor Bus Interface.

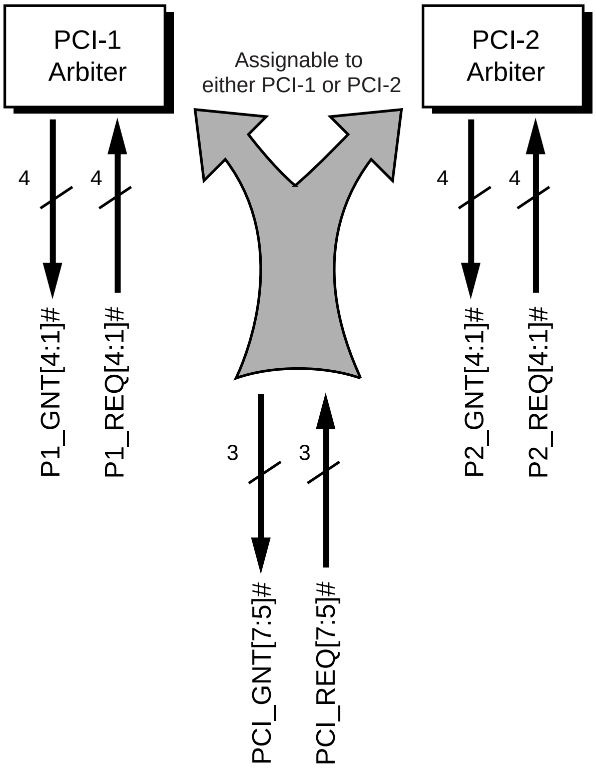
6.2 PCI Interface Arbitration

Each PowerSpan II PCI Interface supports a PCI central arbiter. Each arbiter has dedicated support for the PowerSpan II PCI Master — with internal request and grant signaling and up to four external PCI masters.

PowerSpan II provides external pins to support three additional external PCI masters — PCI_REQ[7:5]#/PCI_GNT[7:5]#. Pairs of these additional arbitration pins can be individually assigned to the PCI-1 arbiter or the PCI-2 arbiter (see Figure 20). Assignment of these pins is accomplished by initializing the PCI Arbiter Master (PCI_Mx) bits and PCI Arbiter Pins Configuration (PCI_ARB_CFG) bit in the “Miscellaneous Control and Status Register” on page 378. These bits can be configured either through EEPROM load at reset (see “I2C/EEPROM” on page 153) or direct PowerSpan II register access. Requests on PCI_REQ#[7:5] are ignored until these bits are initialized.

Signals PCI_REQ[7:5]# / PCI_GNT[7:5]# operate at a maximum of 33 MHz.

Figure 20: Assignment of Additional Bus Requesters with PCI Arbiters



In the Single PCI PowerSpan II, PCI_REQ#[7:5]/PCI_GNT#[7:5] are assigned to the PCI-1 arbiter

6.2.1 Arbitration Levels

The PowerSpan II PCI arbiter implements a fairness algorithm in order to prevent deadlocks. There are two priority levels signed to the PCI master agents. Fairness is defined by the *PCI 2.2 Specification* as an algorithm that grants all potential PCI masters access to the bus, independent of other requests.

6.2.1.1 High and Low Priority PCI Agents

There are two priority levels assigned to the PCI Master Agents: high priority and low priority. Each priority level performs a round-robin arbitration algorithm among the PCI masters assigned to each level. For example, all the PCI masters assigned to the lower priority level represent one entry in the higher priority round-robin arbitration. For every turn of the high priority round-robin arbitration, high priority PCI masters asserting Px_REQ# are granted access to the PCI bus. At the same time, only one lower priority level PCI master asserting Px_REQ# is granted access to the PCI bus.

Arbitration on PowerSpan II is hidden. Hidden arbitration means it occurs during the previous access so that no PCI cycles are consumed due to arbitration — except when the bus is in an idle state.

6.2.1.2 Requesting the PCI Bus

When the bus is idle a master requesting the bus has 16 clocks from the detection of Px_GNT# asserted to drive Px_FRAME# asserted. If the 16 clocks is exceeded, the arbiter assumes the master is unable to drive the bus and re-arbitrates the bus to another requesting master. PCI masters unable to assert Px_FRAME# within 16 clocks of detecting Px_GNT# asserted lose their turn to access the PCI bus.

The PCI bus is idle when both Px_FRAME# and Px_IRDY# are negated.

Functioning and Non-functioning PCI Masters

A master that does not respond to the Px_GNT# signal in 16 clocks is considered a non-functioning master by the PowerSpan II PCI*x* Arbiter when the Status enable (STATUS_EN) bit is set to 1 in the PCI*x* Arbiter Control register (see page 339). The STATUS_EN bit enables an internal monitor in the PowerSpan II PCI*x* Arbiter that checks that no PCI Master waits longer than 16 PCI clock cycles before starting a transaction.

When a master takes longer than 16 clocks before starting a transaction, the STATUS bit is set to 1 in the PCI*x* Arbiter Control register (see page 339). When the STATUS bit is set to 1 by the PowerSpan II PCI*x* arbiter, the PowerSpan II arbiter does not include the non-functioning PCI Master in its arbitration algorithm. When the bit is set to 0, the operating status of the PCI Master is considered by the arbiter to be functioning and the PCI Master is included in the arbitration algorithm used by PowerSpan II.

When PowerSpan II is reset, all masters are considered functioning — the STATUS bit is set to 0.

Refer to “Bus Parking on a Non-functioning Master” on page 168 for more information on bus parking on a master that is non-functioning.

6.2.1.3 PCI Master Driving the PCI Bus

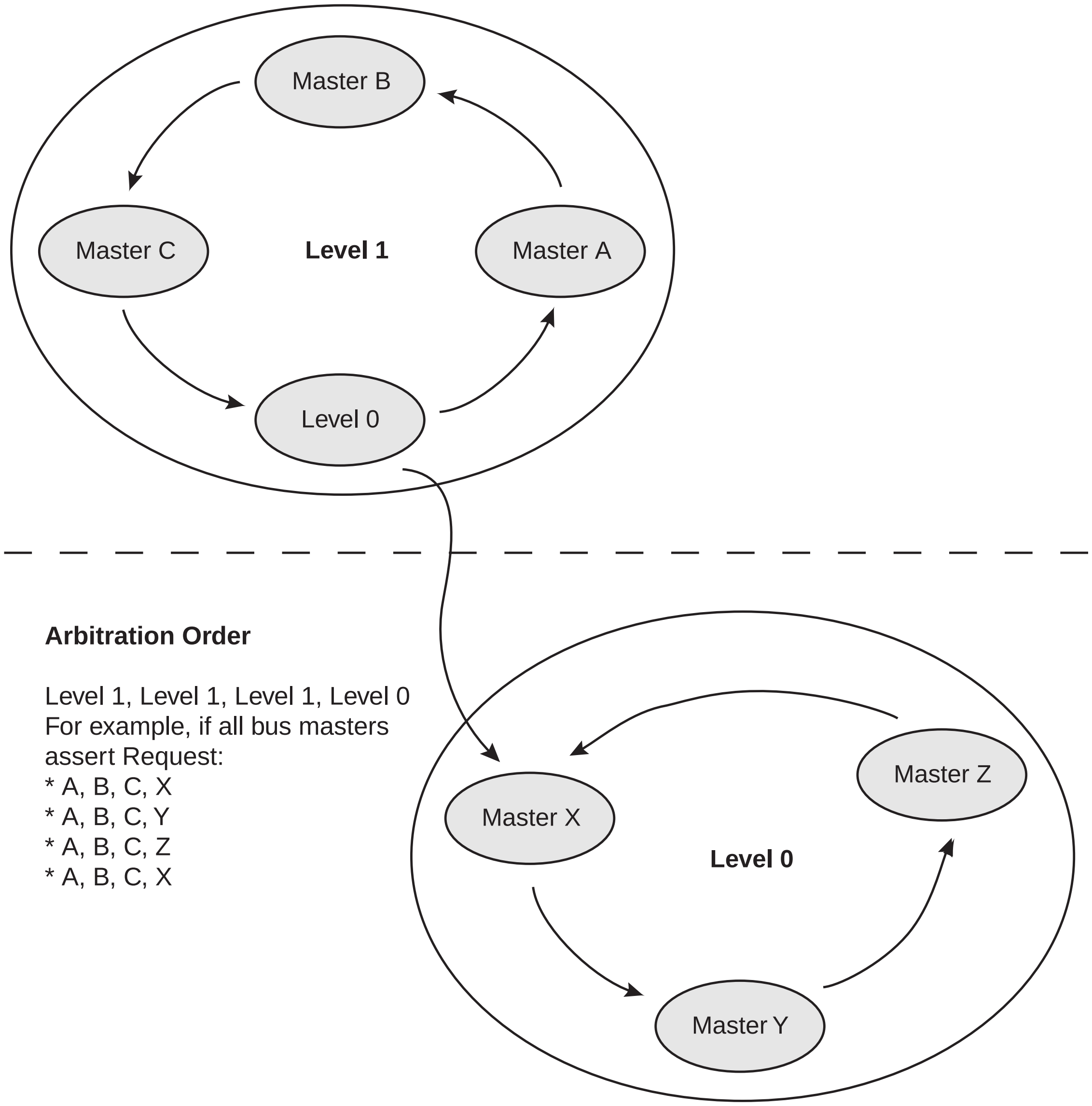
A PCI master accessing the PCI bus has extended assertion of Px_GNT# by the arbiter if no other masters are attempting to access the bus. The arbiter keeps Px_GNT# asserted for the PCI master actively driving the bus. This enables the PCI master to extend its PCI access beyond the Latency Timer.

The Px_GNT# to the driving PCI master remains asserted for the duration of the transaction, regardless of the state of the master’s Px_REQ# signal. The arbiter does not try to park the bus on another master while the present master is actively driving the bus with Px_REQ# negated.

The arbiter negates all Px_GNT# lines for all masters except the PCI master accessing the PCI bus if another PCI master asserts Px_REQ# to gain the bus. Px_GNT# is negated for the duration of the active access. The PowerSpan II arbiter updates the arbitration when it detects Px_FRAME# negated and Px_IRDY# asserted — which occurs in the last data phase of the transaction. The arbitration update is designed to minimize the latency to higher priority PCI masters which may have asserted their Px_REQ# while the present transaction was active.

The arbitration algorithm is illustrated in Figure 21.

Figure 21: Arbitration Algorithm



Arbitration Order

Level 1, Level 1, Level 1, Level 0
For example, if all bus masters assert Request:
* A, B, C, X
* A, B, C, Y
* A, B, C, Z
* A, B, C, X

Each PowerSpan II PCI arbiter is programmable with the corresponding arbiter control register (see “PCI 1 Bus Arbiter Control Register” on page 339) and enabled through power-up option PCI x Arbiter enable (PWRUP_Px_ARB_EN) (see “Resets, Clocks and Power-up Options” on page 201). Each master has a arbitration level for PCI Master Device x (Mx_PRI) bit in the “PCI 1 Bus Arbiter Control Register” on page 339 (Px_ARB_CTRL) to determine its arbitration level.

External Arbitration

When an external arbiter is used, the PowerSpan II PCI Master uses Px_REQ#[1]/ Px_GNT#[1] to acquire the bus.

6.2.2 Bus Parking

The PowerSpan II PCI arbiter provides a flexible address bus parking scheme. When no master is requesting the address bus, the PCI arbiter can park on either the:

- last bus master
- specific bus master

The bus parking mode is determined by PARK bit in the Px_ARB_CTRL register. When Specific Master mode is selected by setting the PARK bit to 0, the BM_PARK[2:0] field selects the specific bus master for bus parking. The parked master must enable its drivers for the following PCI signals:

- AD[31:0]
- Px_C/BE#[3:0]
- PAR

Bus parking does not occur until the PCI bus is idle. When a PCI master is accessing the bus when no Px_REQ# signals are asserted to the PowerSpan II PCI arbiter, Px_GNT# remains asserted to the master until the bus becomes idle.

6.2.2.1 Bus Parking on a Non-functioning Master

It is possible for PowerSpan II to park the bus on a master that is considered non-functioning or to park the bus on the last master that has a status that has changed to non-functioning by the STATUS bit is set to 1 in the PCI*x* Arbiter Control register (see page 339). Refer to "Functioning and Non-functioning PCI Masters" on page 165 for a detailed description of functioning and non-functioning PCI Masters.

When PowerSpan II parks the bus on a non-functioning PCI Master, the PowerSpan II PCI*x* Arbiter waits for another master to request the bus. Once another master request the bus the PCI*x* Arbiter then ignores the non-functioning master until the master is considered functioning. The master status is considered functioning after PowerSpan II is reset (default setting) or the STATUS bit is cleared.

6.3 Processor Bus Arbitration

PowerSpan II's internal processor bus arbiter is enabled through a power-up option (PWRUP_PB_ARB_EN, see "Resets, Clocks and Power-up Options" on page 201). When the internal arbiter is enabled, PowerSpan II's PB Master uses an internal arbitration mechanism to acquire the processor bus. When the internal arbiter is disabled an external arbiter is implemented. The PB Master uses Address Bus Request (PB_BR[1]_), Address Bus Grant (PB_BG[1]_) and Data Bus Grant (PB_DBG[1]_) to gain external bus ownership.

PowerSpan II’s internal processor bus arbiter supports three external processor bus masters and implements internal request and grant lines for the PowerSpan II itself — four processor bus masters in total. The external masters are enabled with the External Master x Enable (Mx_EN) bits in the “Processor Bus Arbiter Control Register” on page 364.

The processor bus arbiter implements two levels of priority. Devices programmed into a specific priority level operate in a round robin fashion. Each master has a External Master x Priority Level (Mx_PRI) bit in the PB_ARB_CTRL register to determine its arbitration level for the address bus. The arbitration level for each master can be reconfigured during system run-time.

6.3.1 Address Bus Arbitration

The PB_BG_ pins change state under the following conditions:

- The assertion of PB_REQ_ when the bus is idle.
- After the assertion of PB Address Acknowledge (PB_AACK_), Bus Grant (PB_BG_) changes to the next requesting master or the parked master.

Requesting masters are required to qualify bus grants before beginning an address tenure.

The MPC8260 and other processor bus agents require the system signal ABB_ to qualify address bus grants. The PowerSpan II PB Master Interface does not require ABB_ to qualify data bus grants.

6.3.1.1 Bus Parking

The PowerSpan II processor bus arbiter provides a flexible address bus parking scheme. When no master is requesting the address bus, the processor bus arbiter can park on either the:

- last bus master
- specific bus master

The bus parking mode is determined by the PARK bit in the PB_ARB_CTRL Register. When specific master mode is selected (PARK = 0), the BM_PARK[1:0] field selects the specific bus master for address parking.

The parked master does not drive any address bus signals until it generates a request to use the address bus.

6.3.2 Data Bus Arbitration

The arbiter samples PB_TT[3] when PB_TS_ is asserted to generate data bus requests. The arbiter grants the data bus to the current address bus owner by asserting one of PB_DBG[1:3]_ signals. The signal is asserted, by default, one clock after PB_TS_. The PB arbiter can be programmed to sample requests two clocks after the PB_TS_ signals is asserted. The arbiter is programmed through the **TS_DLY** bit in the Processor Bus Arbiter Control register (see page 364).

An example application for this feature is some L2 caches hold the BR_ signal after the TS_ signal starts. The PowerSpan II arbiter could see this as a valid request and give the bus to the L2 cache when the bus was not requested. This bit delays when the PB arbiter samples the signal so a false bus request is not granted.

The current data bus grant is negated when the requesting master has qualified the grant.

Requesting masters are required to qualify bus grants before beginning an data tenure.

The MPC8260 and other processor bus agents require the system signal DBB_ to qualify data bus grants. The PowerSpan II PB Master does not require DBB_ to qualify data bus grants.

6.3.2.1 Qualifying Data Bus Grant

Some processors, specifically the MPC7400, must have the data bus grant qualified by the arbiter before it is issued to the master. PowerSpan II, by default, does not qualify the data bus grant by the PowerSpan II PB Arbiter and requires that the requesting master qualify bus grants before beginning an data tenure.

The PowerSpan II PB Arbiter can be programmed to qualify data bus grants before issuing them by setting the 7400_MODE bit in the Reset control and Status register (see page 385). When the 7400_MODE bit is set to 1, the PB arbiter qualifies data bus grants before issuing them to a processor bus master. When the 7400_MODE bit is disabled (default setting) the PB arbiter issues a data bus grant to the processor bus master and expects that the processor bus master the at receives the grant qualifies the grant.

The 7400_MODE bit is a power-up option.

6.3.3 Address Only Cycles

The arbiter supports address only cycles. If Transfer Type (PB_TT[3]) is sampled low during PB_TS_, the arbiter does not grant the data bus. The use of PB_TT[3] as a data bus request means that the PowerSpan II PB arbiter does not support the processor bus instructions eciwx and ecowx.

6.3.4 PowerSpan II Arbiter and System Boot

System boot from the PCI bus can be selected by configuring the processor bus arbiter at power-up to ignore all external requests on PB_BR[3:1]_. This allows an external PCI master, with the PowerSpan II PB Master, to configure the MPC8260 memory controller and load boot code before enabling recognition of requests on PB_BR[3:1]_.

Alternatively, at power-up the processor bus arbiter can be configured to recognize requests on PB_BR[1]_ and ignore requests on PB_BR[3:2]_. In this case the processor connected to PB_BR[1]_ can enable recognition of requests from other masters when its system configuration tasks are complete.

The PowerSpan II processor bus arbiter controls system boot with the M3_EN, M2_EN and M1_EN bits in the "PCI 1 Bus Arbiter Control Register" on page 339 (PB_ARB_CTRL), as well as the power-up option PWRUP_BOOT.

The default value of these Mx_EN bits in the PB_ARB_CTRL register are set using the PWRUP_BOOT option shown in Table 37. When PWRUP_BOOT is selected to boot from PCI, both Px_LOCKOUT bits in the MISC_CSR register are cleared automatically, even if an EEPROM is not present.

Table 37: Mx_EN Default State

PWRUP_BOOT Selection	RST_CSR Register	M1_EN	M2_EN	M3_EN
Boot PCI	PCI_BOOT=1	0	0	0
Boot PB	PCI_BOOT=0	1	0	0

The processor bus arbiter does not have to be enabled to select either PCI or processor bus boot. The PWRUP_BOOT option sets the M1_EN bit in the PB_ARB_CTL register and the P1_LOCKOUT and P2_LOCKOUT bits in the MISC_CSR register. Setting the Px_LOCKOUT bits means any configuration cycles for PowerSpan II on the PCI bus are retried until the Px_LOCKOUT bits are cleared from the processor bus or the EEPROM. When PCI_BOOT is set to 1 (boot is from PCI) the Px_LOCKOUT bits are not set

For more information on power-up options and boot selection, refer to "PowerSpan II Power-up Options" on page 207.

7. Interrupt Handling

An interrupt is a signal informing a program that an event (for example, an error) has occurred. When a program receives an interrupt signal, it temporarily suspends normal processing and diverts the execution of instructions to a sub-routine handled by an interrupt controller. The controller communicates with the host processor and the device that initiated the interrupt to determine how to handle the interrupt.

This chapter discusses the following topics about the PowerSpan II interrupt features:

- "Interrupt Sources" on page 173
- "Interrupt Registers" on page 175
- "Interrupt Pins" on page 183
- "Mailboxes" on page 185
- "Doorbells" on page 185

7.1 Overview

PowerSpan II handles interrupts both from normal device operation and from exceptions. These interrupts are programmed through certain register settings and are signaled through both input and output signal pins.

The following sections describes PowerSpan II interrupt handling.

7.2 Interrupt Sources

Interrupt sources are classified as originating from normal device operation or conditions generated from an exception.

These classifications are discussed in the following sections.

7.2.1 Interrupts from Normal Operations

Interrupt sources associated with normal device operations are:

- Eight bidirectional, configurable interrupts pins:

  P1_INTA#, P2_INTA#, INT[5:0]_
- DMA channels (see "DMA Interrupts" on page 184 for DMA interrupt sources)
- Doorbell interrupts (see "Interrupt Enable Register 0" on page 393 for doorbell interrupt generation)
- Mailbox interrupts (see "Mailbox x Register" on page 412)

7.2.2 Interrupts from Transaction Exceptions

Bus transaction exceptions can occur on any one of the PowerSpan II interfaces — PCI-1, PCI-2 or Processor Bus (PB) — because of bus errors, address parity errors, or data parity errors. When an error occurs, PowerSpan II tracks the direction of the transaction through the interrupt enabling and status function.

Interrupt sources associated with exceptions are:

1. PB Interface errors
   - PB_P1_ERR
   - PB_P2_ERR
   - PB_A_PAR
   - PB_P1_D_PAR
   - PB_P2_D_PAR
   - PB_P1_RETRY
   - PB_P2_RETRY
   - PB_PB_ERR
   - PB_PB_D_PAR
   - PB_PB_RETRY
2. PCI-1 Interface errors
   - P1_PB_ERR
   - P1_P2_ERR
   - P1_A_PAR

- P1_PB_RETRY
- P1_P2_RETRY
- P1_P1_ERR
- P1_P1_RETRY

3. PCI-2 Interface errors
   - P2_PB_ERR
   - P2_P1_ERR
   - P2_A_PAR
   - P2_PB_RETRY
   - P2_P1_RETRY
   - P2_P2_ERR
   - P2_P2_RETRY

See “Error Handling” on page 187 for information on how these interrupts for bus transaction exceptions are associated with error logging functionality.

7.3 Interrupt Registers

PowerSpan II interrupt status and enabling, as well as message passing through mailboxes and doorbells are controlled by the interrupt registers. Table 38 provides a description of PowerSpan II registers controlling these functions.

Table 38: Interrupt Register Description

Register Type	Register Description and Operation
Status	The status register bits cover all of the interrupt sources supported PowerSpan II and indicate active interrupt sources when set (see “Interrupt Status” on page 177). With a some exceptions, all bits in these registers are read and cleared by setting (“R/Write 1 to Clear”)
Enable	The enable register bits cover all of the interrupt sources supported by PowerSpan II and allow status bits to assert an external pin (see “Interrupt Enable” on page 179). With some exceptions, all bits in these registers are Read/Write.

Table 38: Interrupt Register Description

Register Type	Register Description and Operation
Mapping	This series of registers allow each interrupt source to be mapped to a specific interrupt output pin. The mapping definitions are provided in Table 43 (see “Interrupt Mapping” on page 181)
Direction	Interrupt Direction refers to the ability to control the input/output characteristics of the PowerSpan II interrupt pins. Each pin has a corresponding bit that configures it as either an input-only or an output-only (see “Interrupt Pins” on page 183).
Mailbox	The Mailbox registers are a series of eight 32-bit Read/Write registers available for message passing between PowerSpan II interfaces (see “Mailboxes” on page 185)

7.3.1 Interrupt Status

When an interrupt source becomes active, the relevant status bit is set in one of the interrupt status registers. Interrupt Status is reported through two registers: "Interrupt Enable Register 0" on page 393 and "Interrupt Status Register 1" on page 390. Interrupt Status Register 0 provides status for interrupts resulting from normal device operation. This includes I^{2}O, DMA, hardware, doorbell and mailbox interrupts. A register description for ISR0 is provided in Table 39.

All status bits are clear by default.

Table 39: Register Description for Interrupt Status Register 0

Bits	Type	Description
ISR1_ACTV	R	This bit indicates an active status bit in ISR1. This enables software to monitor activity of the other interrupt status register while observing this interrupt status register.
I2O_HOST	R	Indicates to the Host that there are outstanding Message Frame Addresses in the Outbound Post List FIFO.
I2O_IOP	R/ Write 1 to Clear	Indicates to the IOP that there are outstanding Message Frame Addresses in the Inbound Post List FIFO.
DMAx	R/ Write 1 to Clear	Status bit is set when DMAx generates an interrupt. See "DMA x General Control and Status Register" on page 372 for details of DMAx interrupt sources.
x_HW	R/ Write 1 to Clear	An interrupt is outstanding on an interrupt input (one of eight interrupt pins, see "Interrupt Pins" on page 183).
DBx	R/ Write 1 to Clear	Set when a doorbell register is written to in the corresponding IER0 bit.
MBOXx	R/ Write 1 to Clear	Set when there is a write to a mailbox.

Interrupt Status Register 1 provides status for interrupts resulting from exceptions occurring during device operation. This includes maximum retry errors, bus errors, and parity error. A register description for ISR1 is provided in Table 40.

Table 40: Register Description for Interrupt Status Register 1

Bits	Type	Description
ISR0_ACTV	R	This bit allows software to monitor activity of the other interrupt status register while observing this interrupt status register. It essentially chains the two registers so both are only read if necessary.
PB_x_RETRY	R/ Write 1 to Clear	The PowerSpan II PB Master Interface has detected more than the maximum allowable retries.
PB_x_ERR	R/ Write 1 to Clear	The PowerSpan II PB Interface asserted (as slave) or received (as master) PB_TEA_. The PB slave detects illegal conditions, while the PB master receives PB_TEA_.
PB_A_PAR	R/ Write 1 to Clear	An address parity error was detected on the PB.
PB_x_D_PAR	R/ Write 1 to Clear	A data parity error was detected on the PB.
P2_x_ERR	R/ Write 1 to Clear	The PowerSpan II PCI-2 Interface detected an error. The corresponding PCI Control and Status Register must be checked for the error.
P2_A_PAR	R/ Write 1 to Clear	The PowerSpan II PCI-2 Interface detected an address parity error.
P2_x_RETRY	R/ Write 1 to Clear	The PowerSpan II PCI-2 Master has detected more than the maximum allowable retries.

Table 40: Register Description for Interrupt Status Register 1

Bits	Type	Description
P1_x_ERR	R/ Write 1 to Clear	The PowerSpan II PCI-1 Interface detected an error. The corresponding PCI Control and Status Register must be checked for the error.
P1_A_PAR	R/ Write 1 to Clear	The PowerSpan II PCI-1 Interface detected an address parity error.
P1_x_RETRY	R/ Write 1 to Clear	The PowerSpan II PCI-1 Master has detected more than the maximum allowable retries.

The description in Table 40 groups several bits under one name. For example, P1_x_RETRY actually corresponds to P1_P2_RETRY, P1_PB_RETRY, and P1_P1_RETRY.

PowerSpan II has the following conventions:

- For errors detected by a master, PowerSpan II has separate reporting mechanisms for each source interface. For example, if the PowerSpan II PCI-2 master detects an address parity error on a transaction claimed by the PB slave, the P2_PB_A_PAR bit in the ISR1 register is set.
- For errors detected by a target/slave, PowerSpan II has separate reporting mechanisms for each destination port. For example, if the PowerSpan II PB slave detects a data parity error on a transaction destined for an agent connected to the PCI-1 external interface, the P1_PB_A_PAR bit in the ISR1 register is set.

7.3.2 Interrupt Enable

Each interrupt enable bit allows an active source status bit to assert one of the external interrupt pins. Interrupt enabling is controlled through two registers:“Interrupt Enable Register 0” on page 393 and “Interrupt Status Register 1” on page 390. Interrupt Enable Register 0 enables interrupts resulting from normal device operation. This includes I^{2}O, DMA, hardware, doorbell and mailbox interrupts. A register description for IER0 is in Table 41.

All interrupts are disabled by default.

Table 41: Register Description for Interrupt Enable Register 0

Bits	Type	Description
I2O_HOST_MASK	R/W	Masks an interrupt to the Host that there are outstanding MFAs in the Outbound Post List FIFO.
I2O_IOP_EN	R/W	Enables an interrupt to the IOP indicating that there are outstanding MFAs in the Inbound Post List FIFO.
DMAx_EN	R/W	Enables the DMAx interrupt
x_HW_EN	R/W	Enables the corresponding hardware interrupt source t (one of eight interrupt pins, see "Interrupt Pins" on page 183).
DBx_EN	Write 1 to Set	Sets the corresponding status bit
MBOXx_EN	R/W	Enables the Mailbox interrupt source

Interrupt Enable Register 1 enables interrupts resulting from errors occurring during the device operation. This includes maximum retry errors, bus errors, and parity errors. A register description for IER1 is in Table 42.

Table 42: Register Description for Interrupt Enable Register 1

Bits	Type	Description
PB_x_RETRY_EN	R/W	Enables interrupt if the PowerSpan II PB Master has detected more than the maximum allowable retries.
PB_x_ERR_EN	R/W	Enables interrupt if the PowerSpan II PB Interface asserted (as slave) or received (as master) PB_TEA_.
PB_A_PAR_EN	R/W	Enables interrupt if an address parity error was detected on the PB.
PB_x_D_PAR_EN	R/W	Enables interrupt if a data parity error was detected on the PB.
P2_x_ERR_EN	R/W	Enables interrupt if the PowerSpan II PCI-2 Interface detected an error. The corresponding PCI Control and Status Register must be checked for the error.
P2_A_PAR_EN	R/W	Enables interrupt if the PowerSpan II PCI-2 Interface detected an address parity error.
P2_x_RETRY_EN	R/W	Enables interrupt if the PowerSpan II PCI-2 Master has detected more than the maximum allowable retries.

Table 42: Register Description for Interrupt Enable Register 1

Bits	Type	Description
P1_x_ERR_EN	R/W	Enables interrupt if the PowerSpan II PCI-1 Interface detected an error. The corresponding PCI Control and Status Register must be checked for the error.
P1_A_PAR_EN	R/W	Enables interrupt if the PowerSpan II PCI-1 Interface detected an address parity error.
P1_x_RETRY_EN	R/W	Enables interrupt if the PowerSpan II PCI-1 Master has detected more than the maximum allowable retries.

The descriptions in Table 40, Table 41,and Table 42 groups several bits under one name. For example, P1_x_RETRY actually corresponds to P1_P2_RETRY, P1_PB_RETRY, and P1_P1_RETRY.

For errors detected by a master, PowerSpan II has separate reporting mechanisms for each source port. For example, if the PCI-2 master detects an address parity error on a transaction claimed by the PB slave, the P2_PB_A_PAR bit in the ISR1 register is set.

For errors detected by a target/slave, PowerSpan II has separate reporting mechanisms for each destination port. For example, if the PowerSpan II PB slave detects a data parity error on a transaction destined for an agent connected to the PCI-1 external interface, the P1_PB_A_PAR bit in the ISR1 register is set.

7.3.3 Interrupt Mapping

The following registers contain mapping bits for PowerSpan II interrupt sources:

- IMR_MBOX (mailbox sources)
- IMR_DB (doorbell sources)
- IMR_DMA (DMA channel sources)
- IMR_HW (external pin sources)
- IMR_P1 (PCI-1 sources)
- IMR_P2 (PCI-2 sources)
- IMR_PB (Processor Bus sources)
- IMR2_PB (Processor Bus sources)
- IMR_MISC (I_2O sources)

Each interrupt source contains a three bit field in an IMR_x register. This mapping field determines which external pin to assert when the source is active and enabled.

Table 43 details the mapping scheme. The shaded area in the table denotes the shaded map field and interrupt pin information apply only to the Dual PCI PowerSpan II

Table 43: Mapping Definition

Map Field	Interrupt Pin
000	P1_INTA#
001	P2_INTA#
010	INT[0]_
011	INT[1]_
100	INT[2]_
101	INT[3]_
110	INT[4]_
111	INT[5]_

7.4 Interrupt Pins

PowerSpan II has the following interrupt pins:

- P1_INTA#
- P2_INTA#
- INT[5:0]_

Pins INT[5:0]_ are 5V tolerant and general purpose interrupt pins. Interrupt pins are active low and, when configured as input, are sampled on three successive processor bus clock edges to ensure appropriate setting of a status bit.

Each pin is bidirectional, open drain, active low and level sensitive. The input/output character of each interrupt pin is controlled through a corresponding bit in the "Interrupt Direction Register" on page 410. Each pin can be configured as either an input or output. All pins are configured as inputs by default.

P1_INTA# and P2_INTA# are intended to be used with PCI interfaces PCI-1 and PCI-2. They are electrically PCI compliant. To configure PCI interface Px with interrupt capability, the following register settings are required:

- INT_PIN = 0x01, in the "PCI 1 Miscellaneous 1 Register" on page 315 (Px_MISC1). This setting enables a single function PCI device INTA#
- Px_HW_DIR = 0x01, in the "Interrupt Direction Register" on page 410 (ID). Px_INTA# is configured as an output pin.

If the PCI interface Px does not require interrupt capability, the following register settings are necessary:

- INT_PIN = 0x00, in the Px_MISC1 register. This setting enables a single function PCI device that is using no interrupts.
- Px_INTA = user defined, in the ID register. Px_INTA# is used as general purpose pin.

PowerSpan II provides an EEPROM load feature to automatically control the interrupt capabilities of PCI-1 and PCI-2 (see "I2C/EEPROM" on page 153).

7.5 DMA Interrupts

The PowerSpan II DMA supports a number of interrupt sources for each channel. Individual enable and status bits exist for each source. The status and enable bits are contained in the "DMA x General Control and Status Register" on page 372:

Table 44: DMA Channel Interrupt Sources and Enables

Interrupt Source	Enable bit
DONE	DONE_EN
P1_ERR	P1_ERR_EN
P2_ERR	P2_ERR_EN
PB_ERR	PB_ERR_EN
HALT	HALT_EN
STOP	STOP_EN

The following programming steps route done, halt and stop interrupts on DMA channel two onto INT[3]_:

- Set the DMA2_EN bit in the IER0 register
- Program the DMA2_MAP bit to 0bx101 in the IMR_DMA register
- Set the DONE_EN, HALT_EN, STOP_EN bits in the "DMA x General Control and Status Register" on page 372 (DMAx_GCSR)

7.5.1 DMA Interrupt Servicing

To service a DMA interrupt, the following steps must be taken:

- Read "Interrupt Enable Register 0" on page 393 (ISR0) to determine which DMA channel caused the interrupt
- Read DMAx_GCSR to determine which DMA source caused the interrupt
- Service the interrupt
- Write 1 to clear DMAx_GCSR[*status_bit*] and allow a restart of the DMA channel
- Write 1 to clear the DMAx_EN bit in the ISR0 register and negate the interrupt signal

7.6 Mailboxes

PowerSpan II provides eight 32-bit general Mailbox registers for passing messages between processes. Each Mailbox has an associated interrupt enable and status bit. When enabled, an interrupt is generated whenever there is a write to the Mailbox register.

7.7 Doorbells

The Doorbell interrupts are generated by writing 1 to the corresponding Doorbell x Enable (DBx_EN) bit in the "Interrupt Enable Register 0" on page 393. The Doorbell interrupt is cleared by writing a 1 to the corresponding Doorbell x (DBx) bit in the "Interrupt Status Register 0" on page 388.

8. Error Handling

Errors occur in a system as a result of parity, bus, or internal problems. In order to handle errors so that they have minimum effects on an application, devices have a logic module called an error handler. The error handler logs data about the error then communicates the information to another device (for example, a host processor) that is capable of resolving the error condition.

This chapter discusses the following topics about PowerSpan II's error handling features:

- "PB Interface Errors" on page 188
- "PCI Interface Errors" on page 192
- "DMA Errors" on page 198

8.1 Overview

PowerSpan II has error detection, error reporting and error recovery mechanisms for each of the major interfaces — Processor Bus (PB), PCI-1 and PCI-2.

The master and target/slave of each interface provides error detection for transactions where they participate. The types of errors identified are:

- address parity
- data parity
- bus errors (Target-Abort, Master-Abort, and PB_TEA_ assertion)
- maximum retry errors

Each of PowerSpan II's interfaces has a mechanism for reporting detected errors to hardware and/or software. The reporting mechanisms include:

- interrupt status bits in the "Interrupt Status Register 1" on page 390 — the error is reported through PowerSpan II's interrupt generation mechanisms
- PCI standard error reporting mechanisms
- error logging registers that capture parameters from the transaction that caused the error
- assertion of external bus protocol pins

PowerSpan II has separate reporting mechanisms for each source port when errors are detected by a master. For example, if the PC1-2 Master detects a address parity error on a transaction claimed by the PB slave, the P2_PB_A_PAR bit in the ISR1 register is set.

For errors detected by a target/slave, PowerSpan II provides separate reporting tools for each destination port. For example, if the PowerSpan II PB slave detects a data parity error on a transaction destined for an agent connected to the PCI-1 external interface, the PB_P1_D_PAR bit in the ISR1 is set.

Each PowerSpan II DMA channel provides an additional reporting mechanism (see "DMA Errors" on page 198).

8.2 PB Interface Errors

The PB master and slave detect error conditions while participating in PB transactions. In addition to Interrupt Status Register 1, the PB Interface has the following error reporting mechanisms:

- Assertion of PB_TEA_ — provided the Transfer Error Acknowledge Enable (TEA_EN) bit in the "Processor Bus Miscellaneous Control and Status Register" on page 361 (PB_MISC_CSR) is set to 1
- Capture of specific parameters from the transaction that caused the error

  a. "Processor Bus Error Control and Status Register" on page 359 (PB_ERRCS) logs:

     – PB_TT signals

     – PB_TSIZ signals

  b. "Processor Bus Address Error Log" on page 360 (PB_AERR) logs:

     – PB_A signals

Table 45 itemizes the error cases detected and reported by the PB master and the PB slave. Error logging in PB_ERRCS and PB_AERR is triggered for each of the error cases outlined in Table 45.

Table 45: PB Interface Errors

Interface	Error	Destination/Source	Conditions	Reporting
PB slave	Address parity	PCI-1, PCI-2, Registers	Address only, Write, Read	PB_A_PAR in the ISR1 register
	Data parity	PCI-1, Registers	Write	PB_P1_D_PAR in the ISR1 register
		PCI-2	Write	PB_P2_D_PAR in the ISR1 register
	Illegal access	PCI-1 (Memory)	Unaligned access in *PPC little-endian* mode	PB_TEA if TEA_EN=1, PB_P1_ERR in the ISR1 register
		PCI-1 (Configuration, IO, IACK) Registers	Unaligned access in *PPC little-endian* mode, Transaction Size > 4 bytes or burst	
		PCI-2 (Memory)	Unaligned access in *PPC little-endian* mode	PB_TEA if TEA_EN=1, PB_P2_ERR in the ISR1 register
		PCI-2 (Configuration, IO, IACK)	Unaligned access in *PPC little-endian* mode, Transaction Size > 4 bytes or burst	
	Propagation of error from destination master	PCI-1	Read	PB_TEA if TEA_EN=1
		PCI-2	Read	PB_TEA if TEA_EN=1

Table 45: PB Interface Errors

Interface	Error	Destination/Source	Conditions	Reporting
PB Master	Data parity	External PCI-1 agent PB to PCI-1 DMA	Read	PB_P1_D_PAR in the ISR1 register
		External PCI-2 agent PB to PCI-2 DMA	Read	PB_P2_D_PAR in the ISR1 register
		DMA PB Linked-List PB to PB DMA	Read	PB_PB_D_PAR in the ISR1 register
	External agent asserts PB_TEA_	External PCI-1 agent PB to PCI-1 DMA PCI-1 to PB DMA	Read/Write	PB_P1_ERR in the ISR1 register
		External PCI-2 agent PB to PCI-2 DMA PCI-2 to PB DMA	Read/Write	PB_P2_ERR in the ISR1 register
		DMA PB Linked-List PB to PB DMA		PB_PB_ERR in the ISR1 register
	Max retry expires	External PCI-1 agent PB to PCI-1 DMA PCI-1 to PB DMA		PB_P1_RETRY in the ISR1 register
		External PCI-2 agent PB to PCI-2 DMA PCI-2 to PB DMA		PB_P2_RETRY in the ISR1 register

The shaded row from the PB slave section of Table 45 indicates the PB slave asserts PB_TEA_ and sets a bit in the ISR1 register when an external PB master attempts a register access or a PCI-1 Configuration, IO or IACK transaction with any of the following characteristics:

- not naturally aligned — if Endian (END) bit in the PB_REG_BADDR register is programmed for PowerPC little-endian mode
- Transfer Size, PB_TSIZ, indicates a transfer greater than 4 bytes

When a PowerSpan II PCI master is performing a read and encounters a Target-Abort, or generates a Master-Abort, an error indication is latched. When the Address Retry Enable (ARTRY_EN) bit, in the "Processor Bus Miscellaneous Control and Status Register" on page 361, is set to 0 the error is immediately signaled by the PB slave and the transaction terminates. If ARTRY_EN is set to 1, the PB slave propagates this error to the initiating the processor bus agent when it returns to retrieve the read data it requested.

The assertion of the PB_TEA_ signal is controlled with assertion the Transfer Error Acknowledge Enable (TEA_EN) bit in the PB_MISC_CSR register. If TEA_EN is set, the PB slave reports error scenarios as defined in Table 45. If TEA_EN is cleared, transactions determined to be in error are not forwarded to the intended interface or registers. The appropriate ISR1 status bits are set.

The PB slave propagation of PCI Master-Abort for Configuration commands is controlled with the Master-Abort Configuration Error Mapping (MAC_TEA) bit in the PB_MISC_CSR register. When MAC_TEA is set, the PB slave returns all ones on a PCI Configuration read which terminates with Master-Abort. If MAC_TEA is cleared, the PB slave asserts PB_TEA_.

The shaded row from the PB master section of Table 45 indicates that the PB master sets a bit in the ISR1 register if its transaction is terminated with PB_TEA_. The sources for such a transaction are:

- external PCI-1 agent read or write
- DMA channel moving data to/from PCI-1

A typical interrupt service routine for a PB Interface error — as illustrated for in Table 45 — executes the following steps:

1. Read ISR1 to determine which interface reported the error.
2. Read error logs PB_ERRCS and PB_AERR to obtain diagnostic information if the PB Interface reported the error.
3. Clear the Error Status (ES) bit in the PB_ERRCS to enable future error logging.
4. Clear the status bit in ISR1— this negates external interrupt pin.
5. Fix the configuration issue that caused the error.
6. Retry the transaction that caused the error.

The flow of transactions through the PowerSpan II interfaces is independent of error status bits in ISR1 and Error Status bit in the "Processor Bus Error Control and Status Register" on page 359. If PowerSpan II detects an error while processing a transaction, subsequent transactions are not affected.

The transaction response for a PB slave error is as follows:

- address parity: do not claim the transaction
- data parity: transaction proceeds normally to its destination
- illegal access (see Table 45)

The transaction response for a PB master error is as follows:

- data parity on reads:

  transaction proceeds normally back to the source

  correct data parity is calculated internally and propagated back to the source
- assertion of PB_TEA_, expiration of max retry counter:

  all writes

  – stop the transaction

  – purge the entire source transaction from the Switching Fabric

  – for DMA writes, error status sent to DMA channel registers

  all reads

  – stop the transaction

  – latch error condition for propagation back to source

8.3 PCI Interface Errors

In the following discussion Px refers to the PCI Interface that detected the error and Py refers to the alternate PCI Interface.

The Px master and target detect error conditions while participating in PCI bus transactions. In addition to the “Interrupt Status Register 1” on page 390 (ISR1), the Px Interface provides the following reporting mechanisms:

- External signaling of the following signals:

  Target-Abort

  Master-Abort

  address parity errors

  data parity errors
- Detection of Target-Abort.

- Standard PCI error reporting in “PCI 1 Control and Status Register” on page 301 (Px_CSR).
- Capture of specific parameters from the transaction that caused the error:

  Error Control and Status register Px_ERRCS, which logs PCI command

  Address Error Log register Px_AERR, which logs PCI Address (Px_AD)

Table 46 itemizes the error cases detected and reported by the Px master and the Px target. Error logging in Px_ERRCS and Px_AERR is triggered for each of these error cases

Table 46: PCI Interface Errors

Interface	Error	Destination/ Source	Conditions	Reporting

Px
target
	Address parity	PB, Registers, Py	Write, Read	Px_SERR if
				PERESP=1 and
				SERR_EN=1,
				S_SERR in the Px_CSR register if
				PERESP=1 and
				SERR_EN=1,
				D_PE in the Px_CSR register,
				Px_A_PAR in the ISR1 register

	Data parity	PB, Registers	Write	Px_PERR if PERESP=1, D_PE in the Px_CSR register, Px_PB_ERR in the ISR1 register
		Py	Write	Px_PERR if PERESP=1, D_PE in the Px_CSR register, Px_Py_ERR in the ISR1 register
	Propagation of error from destination master	PB	Read	Target-Abort, S_TA in the Px_CSR register, Px_PB_ERR in the ISR1 register
		Py	Read	Target-Abort, S_TA in the Px_CSR register, Px_Py_ERR in the ISR1 register

Table 46: PCI Interface Errors

Interface	Error	Destination/ Source	Conditions	Reporting
Px master	Data parity	External PB agent Px to PB DMA	Read	Px_PERR if PERESP=1, MDP_D in the Px_CSR register if Px_PERR, D_PE in the Px_CSR register, Px_PB_ERR in the ISR1 register
		External Py agent Px to Py DMA	Read	Px_PERR if PERESP=1, Px_CSR[MDP_D] if Px_PERR_, D_PE in the Px_CSR register, Px_Py_ERR in the ISR1 register
		DMA Px Linked-List Px to Px DMA	Read/Write	Px_PERR if PERESP=1, MDP_D in the Px_CSR register if Px_PERR_, D_PE in the Px_CSR register, Px_Px_ERR in the ISR1 register
	External agent generates Target-Abort	External PB agent Px to PB DMA PB to Px DMA	Read/Write	R_TA in the Px_CSR register, Px_PB_ERR in the ISR1 register
		External Py agent Px to Py DMA Py to Px DMA	Read/Write	R_TA in the Px_CSR register, Px_Py_ERR in the ISR1 register
		DMA Px Linked-List Px to Px DMA		R_TA in the Px_CSR register, Px_Px_ERR in the ISR1 register

Table 46: PCI Interface Errors

Interface	Error	Destination/ Source	Conditions	Reporting
Px master	Px master generates Master-Abort	External PB agent Px to PB DMA PB to Px DMA	Read/Write	R_MA in the Px_CSR register, Px_PB_ERR in the ISR1 register
		External Py agent Px to Py DMA Py to Px DMA		R_MA in the Px_CSR register, Px_Py_ERR in the ISR1 register
		DMA Px Linked-List Px to Px DMA		R_MA in the Px_CSR register, Px_Px_ERR in the ISR1 register
Px master	Max retry expires	External PB agent Px to PB DMA PB to Px DMA		Px_PB_RETRY in the ISR1 register
		External Py agent Px to Py DMA Py to Px DMA		Px_Py_RETRY in the ISR1 register
		DMA Px Linked-List Px to Px DMA		Px_Px_RETRY in the ISR1 register

When the PB master or Py master are performing a read and encounter an error condition, an error indication is latched. The Px target propagates this error to the initiating Px external master when it comes back to acquire the read data it requested. This scenario is indicated by the shaded row in the Px target section of Table 46. The Px target signals a Target-Abort on the bus and sets the Px Processor Bus Error (Px_PB_ERR) bit in the ISR1and Signaled Target-Abort (S_TA) bit in the Px_CSR. In this case the PowerSpan II PB Master or Py master and the Px target reports the error.

The shaded row from the Px master section of Table 46 indicates that the Px master sets the Px_PB_ERR bit in the ISR1 register and the R_TA bit in the Px_CSR register if its transaction terminates with a Target-Abort. The sources for such a transaction are:

- an external PB agent read or write
- a DMA channel moving data to/from PB

The MDP_D bit in the Px_CSR register is also set for data parity errors detected by an external target during write transactions. This condition was not included in the Px master section of Table 46 because the master does not detect the error.

The assertion of Px_PERR# is controlled with the Parity Error Response (PERESP) bit in the Px_CSR. The assertion of Px_SERR# is controlled with the PERESP bit and SERR# Enable (SERR_EN) bit in the Px_CSR.

The user controls the Px target propagation of PCI Master-Abort Configuration command initiated on Py with MAC_ERR. When the Master-Abort Configuration Error Mapping (MAC_ERR) bit is set in the Px_MISC_CSR, the Px target returns all ones on a Py Configuration read that terminates with Master-Abort. When the MAC_ERR bit is cleared, the Px target responds with Target-Abort.

A typical interrupt service routine for a PB Interface error — as illustrated for in Table 46 — executes the following steps:

1. Read ISR1 to determine which interface reported the error.
2. If the PCI-1 Interface reported the error:
   - Read error logs Px_ERRCS and Px_AERR to obtain diagnostic information.
   - Read Px_CSR to distinguish address parity, data parity, Target-Abort, Master-Abort scenarios.
3. Clear Px_ERRCS[ES] to enable future error logging.
4. Clear the status bit in ISR1.

   Negates external interrupt pin.
5. Clear the error bits in Px_CSR.
6. Fix the configuration issue that caused the error.
7. Retry the transaction that caused the error.

The flow of transactions through PowerSpan II is independent of error status bits in ISR1, error status bits in P1_CSR and the error log status bit P1_ERRCS[ES].

The transaction response for a Px target error is:

- address parity: claim and complete as normal
- data parity: transaction proceeds normally to its destination

The transaction response for a Px master error is:

- data parity on reads

  transaction proceeds normally to its source

  correct data parity is calculated internally and propagated back to the source
- detection of Target-Abort, generation of Master-Abort, expiration of max retry counter:

  all writes:

  – stop the transaction
  – purge the entire source transaction from the Switching Fabric
  – for DMA writes, error status sent to DMA channel registers

  all reads:

  – stop the transaction
  – latch error condition for propagation back to source

8.4 DMA Errors

A PowerSpan II DMA channel requires a PowerSpan II master to service source activity and a second PowerSpan II master to service destination activity. These masters provide error detection and reporting services as described in the previous sections. The DMA channel provides the following additional status bits to indicate an error condition on an interface currently in use:

- Processor Bus Error (PB_ERR) bit in the “DMA x General Control and Status Register” on page 372 (DMAx_GCSR)
- PCI-1 Bus Error (P1_ERR) in the DMAx_GCSR register
- PCI-2 Bus Error (P2_ERR) in the DMAx_GCSR register

These status bits can be used to cause the assertion of a PowerSpan II interrupt pin according to “Interrupt Handling” on page 173.

Assume that an error occurred at the PCI-1 master using DMA-2. A typical interrupt service routine executes the following steps:

1. ISR1 read to determine which interface reported the error.
2. If PCI-1 reports the error:
    - error logs P1_ERRCS and P1_AERR read to obtain diagnostic information.
    - P1_CSR read to distinguish address parity, data parity, target abort, master abort scenarios.
3. ISR0 read to determine if a DMA2 status bit is set.
4. DMA2_GCSR read to determine which condition caused the channel to interrupt.
5. The ES bit is cleared in the P1_ERRCS register to enable future error logging.
6. The status bit in ISR is cleared — this negates external interrupt pin.
7. The status bit in P1_CSR is cleared.
8. Configuration issue that caused the error is corrected.
9. P1_ERR bit in the DMA2_GCSR is cleared to allow DMA channel two to restart.
10. DMA channel two is restarted.

9. Resets, Clocks and Power-up Options

This chapter describes the resets, clocks and power-up options implemented by PowerSpan II. The following topics are discussed:

- "Reset" on page 201
- "Clocks" on page 205
- "Power-Up Options" on page 205
- "MPC8260 Configuration Slave Mode" on page 209

9.1 Reset

PowerSpan II provides reset capabilities for Host, Adapter and Hot Swap applications. Host and Adapter situations are described in "Typical Applications" on page 481, while Hot Swap information is in "PCI Interface" on page 43.

9.1.1 Reset Pins

PowerSpan II reset pins are listed in Table 47..

Table 47: PowerSpan II Reset Pins

Pin Name	Direction	Description
PO_RST_	Input	Power-On Reset
HEALTHY_	Input	Board Status
PB_RST_	Bidirectional Open Drain	Processor Bus Hard Reset

Table 47: PowerSpan II Reset Pins

Pin Name	Direction	Description
P1_RST#	Bidirectional	PCI-1 Bus Reset
P2_RST#	Bidirectional	PCI-2 Bus Reset
TRST_	Input	JTAG Reset

All pins indicate a reset condition when driven low, except for HEALTHY# signal.

9.1.1.1 Reset Direction Control Pins

Each bidirectional reset pin has a dedicated direction control pin. The assertion of a reset pin configured as input propagates to all reset pins configured as output.

The relationship between the reset and direction control pins is defined in Table 48.

Table 48: Reset Direction Control Pins

Control Pin	Associated Reset Pin	Description
PB_RST_DIR	PB_RST	Direction of PB_RST_ 0=Input 1=Output
P1_RST_DIR	P1_RST	Direction of P1_RST# 0=Input 1=Output
P2_RST_DIR	P2_RST	Direction of P2_RST# 0=Input 1=Output

The dedicated direction control pins must either be pulled up or down at all times. At least one of the bidirectional reset pins must be configured as an input.

9.1.1.2 Reset Response

The assertion of an external reset pin elicits a specific response from PowerSpan II. Table 49 defines how various PowerSpan II resources are affected by active reset pins.

Table 49: PowerSpan II Reset Response

	PowerSpan II Resource					
Reset Pin	**PLLs**	**PCI-1 Registers**	**PCI-2 Registers**	**PB Registers**	**PowerSpan II Device Specific Registers**	**Finite State Machines**
PO_RST_ = 0	Yes	Yes	Yes	Yes	Yes	Yes
HEALTHY# = 1	Yes	Yes	Yes	Yes	Yes	Yes
PB_RST_ = 0	No	No	No	Yes	Yes	Yes
P1_RST# = 0	No	Yes	No	No	Yes	Yes
P2_RST# = 0	No	No	Yes	No	Yes	Yes

PowerSpan II's response to the assertion of a bidirectional reset pin is independent of the direction of that pin.

PowerSpan II's PB_RST_, P1_RST# and P2_RST# are considered bidirectional. When they are configured as outputs, they still sense logic lows on the PowerSpan II reset signal. If these signals sense the logic low on their particular reset signal, the signal stays asserted and PowerSpan II is in a reset state. For example, if PB_RST_ is asserted low PowerSpan II enters a reset state.

PowerSpan II's response to the assertion of each pin is asynchronous — it does not require the presence of a clock. PowerSpan II synchronizes the negation of reset. The de-assertion edge of all input signals is required to be monotonic (bounce free) through the input switching range.

The Phase Locked Loops (PLLs) in PowerSpan II are reset by the assertion of PO_RST_ or negation of HEALTHY#. The assertion of PO_RST_ or negation of HEALTHY# causes all other PowerSpan II resources to be reset. These resources are not released from reset until all PLLs are locked.

⚠ Applications that use both HEALTHY# and PO_RST_ must assert HEALTHY# before negating PO_RST_.

The HEALTHY# pin affects the tristate enable control of PowerSpan II outputs, as well as the inhibit control of PowerSpan II input buffers. See "CompactPCI Hot Swap Silicon Support" on page 67 for more details on the use of HEALTHY#.

The assertion of TRST_ resets the JTAG controller and configures the Boundary Scan Register for normal system operation.

Customers must assert TRST_ concurrently with PO_RST_ as part of the power-up reset sequence.

9.1.1.3 Reset Generation

Each of PowerSpan II's three interfaces have bidirectional reset pins that are used to reset the hardware on the associated bus.

PowerSpan II assertion of PB_RST_ occurs if PB_RST_DIR is pulled high and one of the following occurs:

- PO_RST_ asserted
- P1_RST_DIR is pulled low and P1_RST# is asserted
- P2_RST_DIR is pulled low and P2_RST# is asserted

PowerSpan II assertion of P1_RST# occurs if P1_RST_DIR is pulled high and on of the following occurs:

- PO_RST_ asserted
- PB_RST_DIR is pulled low and PB_RST# is asserted
- P2_RST_DIR is pulled low and P2_RST# is asserted

PowerSpan II assertion of P2_RST_ occurs if P2_RST_DIR is pulled high and on of the following occurs:

- PO_RST_ asserted
- PB_RST_DIR is pulled low and PB_RST# is asserted
- P1_RST_DIR is pulled low and P1_RST# is asserted

The negation of HEALTHY# tristates all PowerSpan II output pins, including the reset outputs. PowerSpan II reset outputs do not respond immediately to the negation of PO_RST_ because they are negated once all PLLs are locked.

9.2 Clocks

Each of the PowerSpan II external ports has a clock input pin. The pins are:

- PB_CLK
- P1_CLK
- P2_CLK

The clock input for each port enables PowerSpan II's master/target state machines to be synchronized to the external bus. Each interface has a dedicated PLL designed to eliminate clock tree insertion delay. PowerSpan II requires the input clock to be at the specified frequency before the PLL is removed from reset. PowerSpan II PLLs are reset during the assertion of PO_RST_ or the negation of HEALTHY#. The PLLs are not locked until after the negation of PO_RST_ or HEALTHY#.

Each PLL has a dedicated configuration pin to indicate the desired operating frequency range. The following configuration pins are used by the PLL:

- PB_FAST
- P1_M66EN
- P2_M66EN

The input clocks are not required to maintain specific phase relationships. However, there is a limitation on the range of input clock periods. The ratio of the maximum period to minimum period, for all three clock inputs, must be less than four. For example, if the period of PB_CLK is 10 ns, the periods of P1_CLK and P2_CLK must be less than, but not equal to, 40 ns.

PowerSpan II has power-up options for bypassing all three PLLs and observing the locking status of each PLL in system operation. Refer to "Power-Up Options" on page 205 for more information on power-up options.

9.3 Power-Up Options

To ensure proper operation, a number of PowerSpan II features must be configured by completion of the power-up reset sequence. PowerSpan II has the following modes to configure these power-up options:

- Multiplexed System Pins mode
- MPC8260 Configuration Slave mode

The Multiplexed System Pins mode multiplexes a system pin during the power-up reset sequence to configure each of the power-up options.

In the MPC8260 Configuration Slave mode, the power-up options are latched from PB_D with PB_RSTCONF_.

The PB_RSTCONF_ signal determines the configuration mode used by PowerSpan II. If PB_RSTCONF_ is asserted low, then the MPC8260 Configuration Slave mode is used to configure PowerSpan II. At all other times the Multiplexed System Pins mode is used by PowerSpan II. The Multiplexed System Pins mode determines the value of the power-up options.

Power-up option status can be confirmed by reading the Reset Control and Status (RST_CSR) register (see "MPC8260 Configuration Slave Mode" on page 209 for more information).

Power-up options are not affected by reset events on the PB_RST_, P1_RST# or P2_RST# pins.

Table 50 lists PowerSpan II power-up options and directions for their configuration with PowerSpan II system pins and the processor data bus in configuration slave mode.

Table 50: PowerSpan II Power-up Options

Power-up Option	Selection	System Pin[a]	PB_D Pin[b]	RST_CSR Register
PB Arbiter Enable (PWRUP_PB_ARB_EN)	Enable PB Arbiter	PB_FAST=1	PB_D[0]=1	PB_ARB_EN=1
	Disable PB Arbiter	PB_FAST=0	PB_D[0]=0	PB_ARB_EN=0
PCI-1 Arbiter Enable (PWRUP_P1_ARB_EN)	Enable PCI-1 Arbiter	P1_M66EN=1	PB_D[1]=1	P1_ARB_EN=1
	Disable PCI-1 Arbiter	P1_M66EN=0	PB_D[1]=0	P1_ARB_EN=0
PCI-2 Arbiter Enable (PWRUP_P2_ARB_EN)	Enable PCI-2 Arbiter	P2_M66EN=1	PB_D[2]=1	P2_ARB_EN=1
	Disable PCI-2 Arbiter	P2_M66EN=0	PB_D[2]=0	P2_ARB_EN=0
Primary PCI Select (PWRUP_PRI_PCI)	PCI-1 is Primary	INT[5]_=1	PB_D[3]=0	PRI_PCI=0
	PCI-2 is Primary	INT[5]_=0	PB_D[3]=1	PRI_PCI=1
PCI-1 REQ64 Enable (PWRUP_P1_R64_EN)	Disable P1_REQ64_	INT[4]_=1	PB_D[4]=0	P1_R64_EN=0
	Enable P1_REQ64_	INT[4]_=0	PB_D[4]=1	P1_R64_EN=1
Boot select (PWRUP_BOOT)	PB Boot	INT[3]_=1	PB_D[5]=0	PCI_BOOT=0
	PCI Boot	INT[3]_=0	PB_D[5]=1	PCI_BOOT=1
7400 Mode Enable (PWRUP_7400_MODE)	Disable 7400_Mode	INT[2]_=1	PB_D[6]=0	7400_MODE=0
	Enable 7400_Mode	INT[2]_=0	PB_D[6]=1	7400_MODE=1
PLL Bypass Enable (PWRUP_BYPASS_EN)	Disable PLL Bypass	INT[1]_=1	PB_D[7]=0	BYPASS_EN=0
	Enable PLL Bypass	INT[1]_=0	PB_D[7]=1	BYPASS_EN=1

[a] The information in the System Pin column is used when PowerSpan II is in Multiplexed System Pin mode (see page 208)

[b] The information in the PB_D Pin column is used when PowerSpan II is in MPC8260 Configuration Slave mode (page 209).

The options PWRUP_PB_ARB_EN, PWRUP_P1_ARB_EN, and PWRUP_P2_ARB_EN are used to select between an external arbiter or a PowerSpan II arbiter for each interface.

PWRUP_PRI_PCI designates PCI-1 or PCI-2 as being connected to the Primary PCI Interface segment in the system (see “Primary PCI” on page 44 for more details on Primary PCI Interface functionality).

When PWRUP_P1_R64_EN is enabled, the PowerSpan II PCI-1 Interface pulses P1_REQ64# during assertion of P1_RST# to signal the presence of a 64-bit data path to all agents on the bus segment. PowerSpan II must be the Central Resource in the system and control both P1_REQ64# and P1_RST#. By meeting these requirements, PowerSpan II ensures that the timing parameters a for a 64-bit data width are satisfied.

The option PWRUP_BOOT enables the system designer to control boot from PCI or from the processor bus. Refer to “Arbitration” on page 163 for more information

By enabling option PWRUP_BYPASS_EN, all PLLs in the design are bypassed.

In the Single PCI PowerSpan II, the following power-up options are not configurable:

- PWRUP_P2_ARB_EN
  - — PCI-2 arbiter is always disabled
- PWRUP_PRI_PCI
  - — PCI-1 is always the Primary Interface

9.3.1 Multiplexed System Pin Mode

PowerSpan II multiplexes a number of system pins to provide power-up support (see Table 50). The multiplexed pins are used to communicate system or power-up information. The multiplexed system pins mode is the default mode for PowerSpan II power-up options.

During the assertion of PO_RST_, the following pins are in system mode:

- PB_FAST

- P1_M66EN
- P2_M66EN

During the assertion of PO_RST_, the following pin is in power-up option mode:

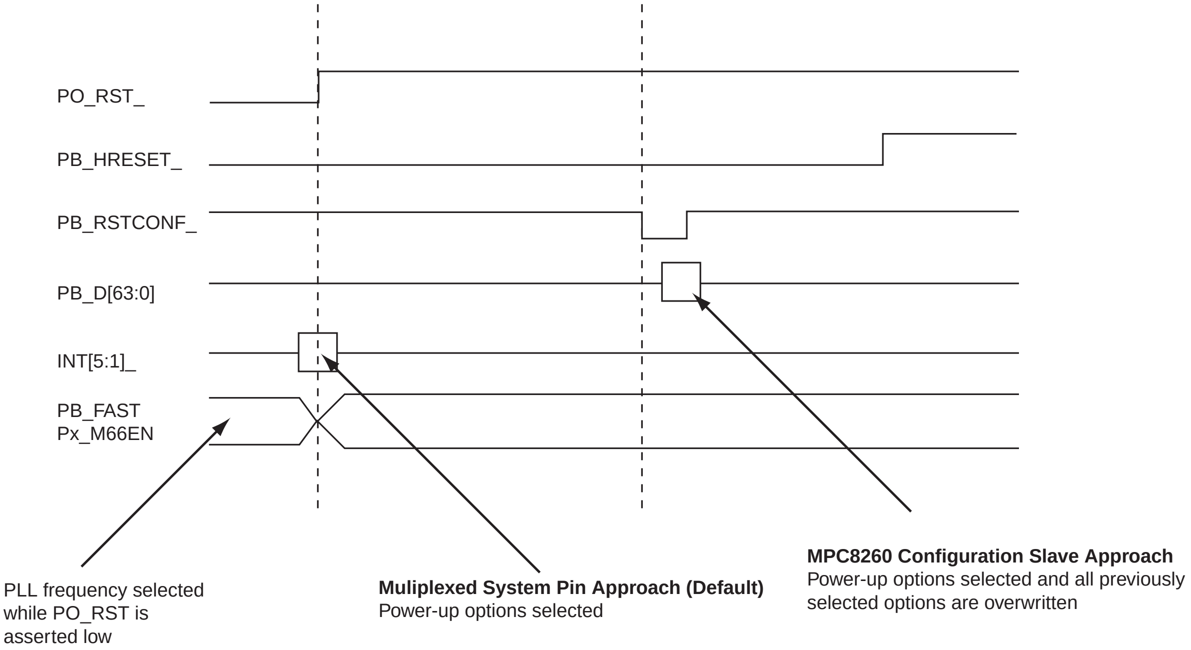
- INT[5:1]_

When PO_RST_ is asserted, the PLL frequency is selected.

The required system levels are latched by PowerSpan II at the negation of PO_RST_ (see Figure 22). The level on these pins determines the power-up options settings. The system levels are typically provided by external transceiver or FPGA.

System operation of these pins requires external pull-ups. Default values for power-up options loaded by INT[5:1]_ are highlighted in Table 50 and are selected to require a minimum number of transceivers in most applications.

Figure 22: PowerSpan II Power-up Waveform



PLL frequency selected while PO_RST is asserted low

Muliplexed System Pin Approach (Default)
Power-up options selected

MPC8260 Configuration Slave Approach
Power-up options selected and all previously selected options are overwritten

9.3.2 MPC8260 Configuration Slave Mode

When there is a configuration master in a PowerSpan II system, for example the MPC8260, the MPC8260 Configuration Slave mode overrides the default power-up option — the Multiplexed System Pins mode. The MPC8260's slave mode power-up options overwrite the multiplex system pin mode power-up options that were sampled at PO_RST_ (see Figure 22).

MPC8260 systems provide support for a single configuration master and up to seven configuration slaves. During the assertion of HRESET_, the configuration master MPC8260 reads configuration words from memory and writes them to the configuration slaves. A total of seven 64-bit words are transferred over the data bus. One of A[0:6] lines is strobed to transfer each word.

When PowerSpan II is the configuration slave and the MP8260 is the configuration master, the MP8260 drives the HRESET_ signal.

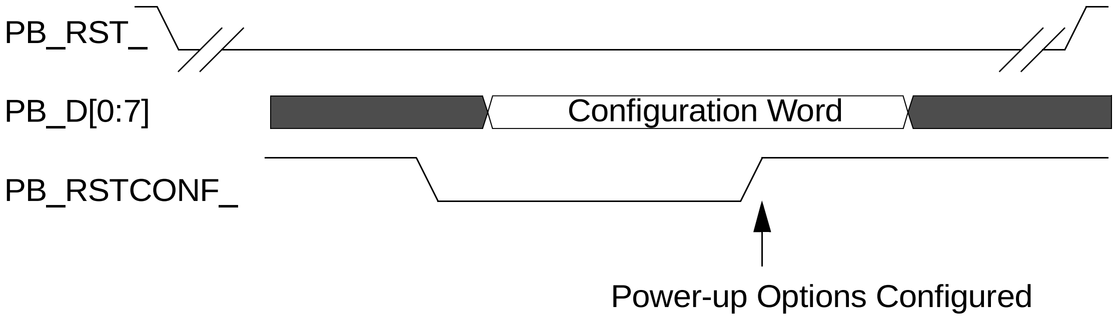
PowerSpan II acts as a configuration slave under the following conditions:

- PB_RSTCONF_ is connected to one of the MPC8260 A[0:6] lines
- PB_RST_ is connected to the MPC8260 HRESET_ signal
- PB_D is connected to the MPC8260 Data (D) line

When the MPC8260 Configuration Slave mode is enabled, the multiplexed pins are sampled first then the configuration word overrides the original setting. In this case, the power-up options are configured by PB_D as defined in Table 50.

PowerSpan II configuration slave timing is illustrated in Figure 23.

Figure 23: PowerSpan II Configuration Slave Timing



The MPC8260 configuration master updates all configuration slaves for each HRESET_ sequence. PowerSpan II receives the same configuration word after each sequence.

9.3.3 Assertion of P1_REQ64#

When the PWRUP_P1_REQ64_EN bit is set to 1, in the RST_CSR register (see page 385), PowerSpan II does not assert P1_REQ64# signal until the configuration word is latched. A 10 clock setup time for the P1_REQ64# signal is required by the *PCI 2.2 Specification*. In order to meet this requirement, PowerSpan II must not be the last four MPC8260 slaves. PowerSpan II must be the Central Resource in the system and control both P1_REQ64# and P1_RST#. By meeting these requirements, PowerSpan II ensures that the timing parameters for a 64-bit data width are satisfied.

10. Endian Mapping

Big-endian refers to a method of formatting data where address 0 (or the smallest address referencing to the data) points to the most significant byte of the data.

Little-endian refers to a method of formatting data where address 0 (or the smallest address referencing the data) points to the least significant byte of the data.

Data in a system must be consistent; that is, the system must be entirely big-endian or little-endian.

This chapter describes the endian mapping system used in PowerSpan II. The following topics are discussed:

- “Conventions” on page 214
- “Processor Bus and PowerSpan II Register Transfers” on page 215
- “Processor Bus and PCI Transfers” on page 219

10.1 Overview

PowerSpan II supports a flexible endian conversion scheme for the following transactions involving the Processor Bus (PB) Interface:

- Access of PowerSpan II registers from the PB Interface
- Transfers between the processor bus and PCI — both externally initiated and PowerSpan II DMA initiated

No endian conversion is performed for transactions mapped between the two PCI interfaces: PCI-1 and PCI-2.

10.2 Conventions

Table 51 illustrates the data bus lanes used to carry each byte of a multi-byte structure on PCI. PCI stores multi-byte structures with little-endian byte ordering.

Table 51: PCI Byte Lane Definitions

Byte Address	PCI Byte Lanes				
AD[2:0]	**64-bit Transaction**		**32-bit Transaction**		
	Lane Number	**Pins**	**AD[2]**	**Lane Number**	**Pins**
000	0	P1_AD[7:0]	0	0	Px_AD[7:0]
001	1	P1_AD[15:8]	0	1	Px_AD[15:8]
010	2	P1_AD[23:16]	0	2	Px_AD[23:16]
011	3	P1_AD[31:24]	0	3	Px_AD[31:24]
100	4	P1_AD[39:32]	1	0	Px_AD[7:0]
101	5	P1_AD[47:40]	1	1	Px_AD[15:8]
110	6	P1_AD[55:48]	1	2	Px_AD[23:16]
111	7	P1_AD[63:56]	1	3	Px_AD[31:24]

Table 52 illustrates the lanes used to carry each byte of a multi-byte structure on a 64-bit PB Interface data bus.

Table 52: 64-bit PB Data Bus Byte Lane Definitions

Byte Address	Processor Bus Byte Lanes			
PB_A[29:31]	**Lane Number**	**PowerSpan II Pins**	**MPC8260 Pins**	**MPC740 Pins**
000	0	PB_D[0:7]	D[0:7]	DH[0:7]
001	1	PB_D[8:15]	D[8:15]	DH[8:15]
010	2	PB_D[16:23]	D[16:23]	DH[16:23]
011	3	PB_D[24:31]	D[24:31]	DH[24:31]
100	4	PB_D[32:39]	D[32:39]	DL[0:7]

Table 52: 64-bit PB Data Bus Byte Lane Definitions

Byte Address	Processor Bus Byte Lanes			
101	5	PB_D[40:47]	D[40:47]	DL[8:15]
110	6	PB_D[48:55]	D[48:55]	DL[16:23]
111	7	PB_D[56:63]	D[56:63]	DL[24:31]

PowerSpan II supports both big-endian and PowerPC little-endian byte ordering. Endian selection with PowerPC is performed with the processor register MSR[LE] and defaults to big-endian. PowerPC little-endian mode allows a PowerPC and Pentium processor to share a data structure in memory.

10.3 Processor Bus and PowerSpan II Register Transfers

The PowerSpan II PB Slave supports register accesses from a PowerPC operating in big-endian or PowerPC little-endian mode. The endian conversion mode for processor access to PowerSpan II registers is selected by programming the END bit in the “Processor Bus Register Image Base Address Register” on page 351. The default mode is big-endian, which matches the default mode of the processor bus.

PowerSpan II registers are little-endian structures. The endian conversion process provided by PowerSpan II for processor bus accesses to its registers is designed to preserve the significance of the programmer’s multi-byte structures or scalars. Endian conversion for access to PowerSpan II registers from the processor is data invariant.

When the processor bus is operating in big-endian mode, the END bit must be set to big-endian mode. In this case, the PowerSpan II PB slave maps the processor bus byte lanes to PowerSpan II register addresses according to Table 53.

Table 53: PowerSpan II Big-Endian PB Register Accesses

		PowerPC Byte Lanes								PowerSpan II Register Address				
											A[1:0]			
Transfer Size	**Starting Address PB_A [29:31]**	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**A[2]**	**11**	**10**	**01**	**00**
Byte	000	D0								0	D0			
	001		D1							0		D1		
	010			D2						0			D2	
	011				D3					0				D3
	100					D4				1	D4			
	101						D5			1		D5		
	110							D6		1			D6	
	111								D7	1				D7
Two bytes	000	D0	D1							0	D0	D1		
	001		D1	D2						0		D1	D2	
	010			D2	D3					0			D2	D3
	100					D4	D5			1	D4	D5		
	101						D5	D6		1		D5	D6	
	110							D6	D7	1			D6	D7
Tri-byte	000	D0	D1	D2						0	D0	D1	D2	
	001		D1	D2	D3					0		D1	D2	D3
	100					D4	D5	D6		1	D4	D5	D6	
	101						D5	D6	D7	1		D5	D6	D7
Word	000	D0	D1	D2	D3					0	D0	D1	D2	D3
	100					D4	D5	D6	D7	1	D4	D5	D6	D7

In PowerPC little-endian mode, the processor munges the address and places the scalar on the external byte lanes starting at this modified address. The scalar is still in big-endian order. This operation is only defined for starting addresses that are a multiple of the size of the scalar. In PowerPC literature, this is referred to as being naturally aligned.

Munging the address makes the address appear to the processor bus that individual aligned scalars are stored as little-endian values when they are actually stored in big-endian order. They are stored at different byte addresses with a double word.

The munging performed by the processor is illustrated in Table 54.

Table 54: Processor Bus Address Munging

Transfer Size	Address modification
4 bytes	XOR with 0b100
2 bytes	XOR with 0b110
1 byte	XOR with 0b111

When the processor bus is operating in PowerPC little-endian mode, END bit must be set to PowerPC little-endian mode. In this case, the PB Slave munges the processor bus address, and maps byte lanes to register addresses to preserve the significance of the scalar.

The byte lane to register address mapping is shown in Table 55. Only munged cases are illustrated. The PB slave asserts PB_TEA_ in response to an unaligned access to a register if the END bit is set for PowerPC little-endian mode.

Table 55: PowerSpan II PowerPC Little-Endian PB Register Accesses

Transfer Size	Starting Address (Munged) PB_A [29:31]	PowerPC Byte Lanes								PowerSpan II Register Address				
											A[1:0]			
		0	1	2	3	4	5	6	7	A[2]	11	10	01	00
Byte	000	D0								1				D0
	001		D1							1			D1	
	010			D2						1		D2		
	011				D3					1	D3			
	100					D4				0				D4
	101						D5			0			D5	
	110							D6		0		D6		
	111								D7	0	D7			
Two bytes	000	D0	D1							1			D0	D1
	010			D2	D3					1	D2	D3		
	100					D4	D5			0			D4	D5
	110							D6	D7	0	D6	D7		
Word	000	D0	D1	D2	D3					1	D0	D1	D2	D3
	100					D4	D5	D6	D7	0	D4	D5	D6	D7

A PCI transaction is generated by accessing the following registers:

- Processor Bus Configuration Cycle Data register (PB_CONF_DATA)
- Processor Bus to PCI-1 Interrupt Acknowledge Cycle Generation register (PB_P1_IACK)
- Processor Bus to PCI-2 Interrupt Acknowledge Cycle Generation register (PB_P2_IACK)

The endian conversion scheme applied for processor bus access to these registers is controlled by the END bit, but the endian mapping scheme in this case is described in "Processor Bus and PCI Transfers" on page 219.

10.4 Processor Bus and PCI Transfers

The following endian conversion modes are provided for transactions involving the processor bus and a PCI Interface:

- big-endian (swap or address invariance)
- little-endian (no swap or data invariance)
- PowerPC little-endian (no swap and address munge)
- true little-endian (swap or address invariance)

The following PowerSpan II register bits are used to control the endian conversion for transactions involving the PB Interface and PCI:

- END [1:0] field in the Py_TIx_CTL Registers
- END [1:0] field in the PCI_TI2O_CTL Registers
- END [1:0] field in the PB_SIx_CTL Registers
- END [1:0] field in the DMAx_TCR Registers

The endian conversion mode of a DMA channel can be updated for each direct mode transaction or for each element in a linked-list.

The following sections describe each of the endian conversion modes.

10.4.1 Big-endian Mode

When operating in big-endian mode, PowerSpan II uses an address invariant scheme for mapping processor bus byte lanes. In this mode, all elements of a multi-byte structure or scalar appear at the same address in both PCI and processor bus spaces, but their relative significance is not preserved.

If the processor bus is programmed to be big-endian, PowerSpan II big-endian mode must be used for processor bus/PCI transactions.

PowerSpan II byte lane mappings for big-endian mode support are illustrated in Table 56. Byte lane number references are defined in Table 51 and Table 52.

Table 56: PowerSpan II Big-endian Mode Byte Lane Mapping

Transfer Size	Start Address	PowerPC Byte Lanes								PCI Byte Lanes							
		0	1	2	3	4	5	6	7	7	6	5	4	3	2	1	0
Byte	000	D0															D0
	001		D1													D1	
	010			D2											D2		
	011				D3									D3			
	100					D4							D4				
	101						D5					D5					
	110							D6			D6						
	111								D7	D7							
Two bytes	000	D0	D1													D1	D0
	001		D1	D2											D2	D1	
	010			D2	D3									D3	D2		
	100					D4	D5					D5	D4				
	101						D5	D6			D6	D5					
	110							D6	D7	D7	D6						
Tri-byte	000	D0	D1	D2											D2	D1	D0
	001		D1	D2	D3									D3	D2	D1	
	100					D4	D5	D6			D6	D5	D4				
	101						D5	D6	D7	D7	D6	D5					
Word	000	D0	D1	D2	D3									D3	D2	D1	D0
	100					D4	D5	D6	D7	D7	D6	D5	D4				
Five bytes	000	D0	D1	D2	D3	D4							D4	D3	D2	D1	D0
	011				D3	D4	D5	D6	D7	D7	D6	D5	D4	D3			
Six bytes	000	D0	D1	D2	D3	D4	D5					D5	D4	D3	D2	D1	D0
	010			D2	D3	D4	D5	D6	D7	D7	D6	D5	D4	D3	D2		

Table 56: PowerSpan II Big-endian Mode Byte Lane Mapping

Transfer Size	Start Address	PowerPC Byte Lanes								PCI Byte Lanes							
		0	1	2	3	4	5	6	7	7	6	5	4	3	2	1	0
Seven bytes	000	D0	D1	D2	D3	D4	D5	D6			D6	D5	D4	D3	D2	D1	D0
	001		D1	D2	D3	D4	D5	D6	D7	D7	D6	D5	D4	D3	D2	D1	
Double	000	D0	D1	D2	D3	D4	D5	D6	D7	D7	D6	D5	D4	D3	D2	D1	D0

10.4.2 Little-endian Mode

When operating in little-endian mode, PowerSpan II uses a data invariant scheme for mapping PowerPC byte lanes. Data invariance preserves the relative byte significance of a structure in both PCI and PowerPC spaces, but translates the byte addressing.

In order to access PCI device registers from the processor bus in little-endian mode, there are certain addressing rules which must be followed. In PowerSpan II when little-endian mode is selected, no address swapping takes place (refer to Table 57 on page 222). This means that the MSB on the processor bus goes to the MSB on PCI. However, the MSB on processor bus is the low address and MSB on PCI is the high address.

10.4.2.1 4 Byte Transactions

When performing 4 byte transactions to the PCI bus in little-endian mode the intended address must XOR the address with 0x4. This creates the address for PCI which is used in the transaction.

In little-endian mode for 4 byte transfers, the following changes must be made:

- change a register on PCI at offset 0x0 using address 0x4
- change a register on PCI at offset 0x4 using address 0x0
- change a register on PCI at offset 0x8 using address 0xC
- change a register on PCI at offset 0xC using address 0x8

These rules enable the transactions to reach the intended targets without manual code changes.

PowerSpan II byte lane mappings for little-endian mode support are illustrated in Table 57 on page 222.

Table 57: PowerSpan II Little-endian Mode Byte Lane Mapping

Transfer Size	Start Address	PowerPC Byte Lanes								PCI Byte Lanes							
		0	1	2	3	4	5	6	7	7	6	5	4	3	2	1	0
Byte	000	D0								D0							
	001		D1								D1						
	010			D2								D2					
	011				D3								D3				
	100					D4								D4			
	101						D5								D5		
	110							D6								D6	
	111								D7								D7
Two bytes	000	D0	D1							D0	D1						
	001		D1	D2							D1	D2					
	010			D2	D3							D2	D3				
	100					D4	D5							D4	D5		
	101						D5	D6							D5	D6	
	110							D6	D7							D6	D7
Tri-byte	000	D0	D1	D2						D0	D1	D2					
	001		D1	D2	D3						D1	D2	D3				
	100					D4	D5	D6						D4	D5	D6	
	101						D5	D6	D7						D5	D6	D7
Word	000	D0	D1	D2	D3					D0	D1	D2	D3				
	100					D4	D5	D6	D7					D4	D5	D6	D7
Five bytes	000	D0	D1	D2	D3	D4				D0	D1	D2	D3	D4			
	011				D3	D4	D5	D6	D7				D3	D4	D5	D6	D7
Six bytes	000	D0	D1	D2	D3	D4	D5			D0	D1	D2	D3	D4	D5		
	010			D2	D3	D4	D5	D6	D7			D2	D3	D4	D5	D6	D7

Table 57: PowerSpan II Little-endian Mode Byte Lane Mapping

Transfer Size	Start Address	PowerPC Byte Lanes								PCI Byte Lanes							
		0	1	2	3	4	5	6	7	7	6	5	4	3	2	1	0
Seven bytes	000	D0	D1	D2	D3	D4	D5	D6		D0	D1	D2	D3	D4	D5	D6	
	001		D1	D2	D3	D4	D5	D6	D7		D1	D2	D3	D4	D5	D6	D7
Double	000	D0	D1	D2	D3	D4	D5	D6	D7	D0	D1	D2	D3	D4	D5	D6	D7

10.4.3 PowerPC Little-endian Mode

In PowerPC little-endian mode, the PB Master swaps byte lanes according to Table 57 and munges outgoing addresses PB_A[29:31] according to Table 54. Address munging does not occur for burst and extended cycles.

In PowerPC little-endian mode, the PB Master is restricted to transferring naturally aligned quantities. External PCI masters or the PowerSpan II's DMA channels can request transactions that are not naturally aligned. The PB Master breaks up these requests into single byte transactions on the processor bus, with a performance penalty.

The PB Slave asserts PB_TEA_ in response to a transaction that is not naturally aligned. These cases are as follows:

- PB_TSIZ = 3, 5, 6, 7 bytes
- PB_TSIZ = 2 bytes and PB_A[31] = 1

For DMA transactions between the processor (60x) bus and the PCI-1 bus, the END bit in the DMAx_TCR registers (page 369) must be set to 11. For all other PowerPC little-endian transfers, the END bit must be set to 01.

10.4.4 True Little-endian Mode

When operating in true little-endian mode, PowerSpan II uses a data invariant scheme for mapping PowerPC byte lanes. Data invariance preserves the relative byte significance of a structure in both PCI and PowerPC spaces, but translates the byte addressing.

In order to access PCI device registers from the processor bus in true little-endian mode, there are certain addressing rules which must be followed. In PowerSpan II when true little-endian mode is selected, no address swapping takes place (refer to Table 58). This means that the MSB on the processor bus goes to the MSB on PCI. However, the MSB on processor bus is the low address and MSB on PCI is the high address.

True little-endian mode cannot be used with the 4 byte read implementation in the PowerSpan II design. The MEM_IO bit must be set to 0 when the END field is set to 11. Refer to “Reads” on page 55 and “Reads” on page 117 for a detailed explanation of the 4 byte read through the PCI Interfaces and PB Interface.

The 4 byte read implementation can be used with the other types of endian conversion.

Table 58: PowerSpan II True Little-Endian Byte Lane Mappings

	Starting Address (Munged)	PowerPC Byte Lanes								PowerSpan II PCI Address				
											A[1:0]			
Transfer Size	PB_A [29:31]	0	1	2	3	4	5	6	7	A[2]	11	10	01	00
Byte	000	D0								0	D0			
	001		D1							0		D1		
	010			D2						0			D2	
	011				D3					0				D3
	100					D4				1	D4			
	101						D5			1		D5		
	110							D6		1			D6	
	111								D7	1				D7
Two bytes	000	D0	D1							0	D0	D1		
	010			D2	D3					0			D2	D3
	100					D4	D5			1	D4	D5		
	110							D6	D7	1			D6	D7
Tri bytes	000	D0	D1	D2						0	D0	D1	D2	
	001		D1	D2	D3					0		D1	D2	D3
	100					D4	D5	D6		1	D4	D5	D6	
	101						D5	D6	D7	1		D5	D6	D7
Word	000	D0	D1	D2	D3					0	D0	D1	D2	D3
	100					D4	D5	D6	D7	1	D4	D5	D6	D7
Five Bytes	000	D0	D1	D2	D3	D4				0	D0	D1	D2	D3
										1	D4			
	011				D3	D4	D5	D6	D7	0				D3
										1	D4	D5	D6	D7

Table 58: PowerSpan II True Little-Endian Byte Lane Mappings

Transfer Size	Starting Address (Munged) PB_A [29:31]	PowerPC Byte Lanes								PowerSpan II PCI Address				
											A[1:0]			
		0	1	2	3	4	5	6	7	A[2]	11	10	01	00
Six Bytes	000	D0	D1	D2	D3	D4	D5			0	D0	D1	D2	D3
										1	D4	D5		
	010			D2	D3	D4	D5	D6	D7	0			D2	D3
										1	D4	D5	D6	D7
Seven Bytes	000	D0	D1	D2	D3	D4	D5	D6		0	D0	D1	D2	D3
										1	D4	D5	D6	
	001		D1	D2	D3	D4	D5	D6	D7	0		D1	D2	D3
										1	D4	D5	D6	D7
Double	000	D0	D1	D2	D3	D4	D5	D6	D7	0	D0	D1	D2	D3
										1	D4	D5	D6	D7

11. Signals and Pinout

This chapter describes the Processor Bus (PB) Interface, Single PCI PowerSpan II and Dual PCI PowerSpan II signals. Signals the differ between the Single PCI PowerSpan II and Dual PCI PowerSpan II are identified in the signal tables. The following topics are discussed:

- "Signal Description" on page 227

11.1 Signal Description

This section organizes the PowerSpan II signals along the following functional groups:

- Processor Bus
- PCI-1
- PCI-2
- Miscellaneous
- Test

The Dual PCI PowerSpan II contains all five of these signal groupings.

The Single PCI PowerSpan II device does not implement the PCI-2 signal group.

11.1.1 Signal Types

Signals are classified according to the types defined in Table 59.

Table 59: Signal Type Definitions

Signal type	Signal type definition
Input	Standard input only signal.
Output	Standard output only signal.
Tristate output	Standard tristate output only signal.
Open drain	Open drain output that allows multiple devices to share as a wire-OR
Tristate bidirectional	Tristate input/output signal.
Bidirectional open drain	Open drain input/output which allows multiple devices to share as a wire or when it is used as output.

⚠ All arbitration signals — REQ# and GNT# — must be weakly pulled-up when using the PowerSpan II’s arbiters. This is true for all of PowerSpan II’s arbiters: Processor Bus, PCI-1 and PCI-2.

11.1.2 Processor Bus Signals

This section describes PowerSpan II PB Interface signals used to interface to the PowerPC 603/740/750 or the MPC8260. Signals in this group are 3.3V LVTTL compatible. The signals are not 5V tolerant.

Table 60 summarizes the signals in this grouping. Signals with electrical characteristics different from the remainder of the group are placed at the end of the table.

Table 60: Processor Bus Signals

Pin Name	Pin Type	Reset State	Recommended Termination	Description
PB_AACK_	Tristate bidirectional	Hi-Z	Pull-up resistor	**Address Acknowledge:** A processor bus slave asserts this signal to indicate that it identified the address tenure. Assertion of this signal terminates the address tenure.
PB_ABB_	Tristate output	Hi-Z	Pull-up resistor	**Address Bus Busy:** Indicates ownership of the processor address bus.
PB_AP[0:3]	Tristate bidirectional	Hi-Z	Pull-up resistor	**Address Parity:** The processor address bus master drives this signal to indicate the parity of the address bus.
PB_ARTRY_	Tristate bidirectional	Hi-Z	Pull-up resistor	**Address Retry:** Assertion of this signal indicates that the bus transaction must be retried by the processor bus master.

Table 60: Processor Bus Signals

Pin Name	Pin Type	Reset State	Recommended Termination	Description
PB_A[0:31]	Tristate bidirectional	Hi-Z	No requirement[a]	**Address Bus:** Address for the current bus cycle. It is driven by PowerSpan II when it is the 603 bus master. At all other times it is an input to PowerSpan II.
PB_BG[1]_	Tristate bidirectional	Hi-Z	Pull-up resistor	**Address Bus Grant:** This is an input when an external arbiter is used and an output when the internal arbiter is used. As input it is used by an external arbiter to grant the processor address bus to PowerSpan II. As output it is used by the internal arbiter to grant the processor address bus to an external bus master. This pin must be weakly pulled high.
PB_BG[2:3]_	Tristate output	Hi-Z	Pull-up resistor	**Address Bus Grant:** It is used by the internal arbiter to grant the processor address bus to the external bus masters. These pins must be weakly pulled high.
PB_BR[1]_	Tristate bidirectional	Hi-Z	Pull-up resistor	**Address Bus Request:** This is an output when an external arbiter is used and an input when an internal arbiter is used. As output it indicates that PowerSpan II requests the ownership of the processor address bus. As input an external master should assert this signal to request the ownership of the processor address bus from PowerSpan II's internal arbiter. This pin must be weakly pulled high.

Table 60: Processor Bus Signals

Pin Name	Pin Type	Reset State	Recommended Termination	Description
PB_BR[2:3]_	Input	Hi-Z	Pull-up resistor	**Address Bus Request:** These are inputs only. They are used by external masters to request the processor address bus from the internal arbiter. These pins must be weakly pulled high.
PB_CI_	Tristate output	Hi-Z	Pull-up resistor	**Cache Inhibit:** It is used for L2 cache control. It indicates whether the transaction should be cached or not.
PB_CLK	Input	-	-	**Processor Bus Clock:** All devices intended to interface with the bus processor side of the PowerSpan II must be synchronized to this clock. The PB_CLK can operate up to 100 MHz.
PB_DBB_	Tristate output	Hi-Z	Pull-up resistor	**Data Bus Busy:** Indicates the ownership of the data bus. The master who owns the processor data bus asserts this signal.

Table 60: Processor Bus Signals

Pin Name	Pin Type	Reset State	Recommended Termination	Description
PB_DBG[1]_	Tristate bidirectional	Hi-Z	Pull-up resistor	**Data Bus Grant:** This is an input when an external arbiter is used and an output when the internal arbiter is used. As input it is used by an external arbiter to grant the processor data bus to PowerSpan II. As output it is used by the internal arbiter to grant the processor data bus to an external bus master. This pin must be weakly pulled high.

Table 60: Processor Bus Signals

Pin Name	Pin Type	Reset State	Recommended Termination	Description
PB_DBG[2:3]_	Tristate output	Hi-Z	Pull-up resistor	**Data Bus Grant:** This is an output only. It is used by the internal arbiter to grant the processor data bus to external bus masters. These pins must be weakly pulled high.
PB_DP[0:7]	Tristate bidirectional	Hi-Z	No requirement[a]	**Data Parity:** The processor data bus slave drives on reads, master drives on write to indicate the parity of the data bus.
PB_DVAL_	Tristate bidirectional	Hi-Z	Pull-up resistor	**Data Valid:** Indicates if the data beat is valid on PB_D[0:63].
PB_D[0:63]	Tristate bidirectional	Hi-Z	No requirement[a]	**Data Bus**
PB_FAST	Input	-	Power-up option	**PLL Configuration:** If the signal is pulled low, it configures the PB Interface PLL to operate with input frequencies between 25 and 50 MHz. If the signal is pulled high, it configures the PB Interface PLL to operate with input frequencies above 50 MHz to a maximum of 100 MHz.

Table 60: Processor Bus Signals

Pin Name	Pin Type	Reset State	Recommended Termination	Description
PB_GBL_	Tristate output	Hi-Z	Pull-up resistor	**Global:** Indicates that the transfer is coherent and it should be snooped by bus masters.
PB_RSTCONF_	Input (Schmitt trigger)	-	-	**Reset Configuration:** Asserted by MPC8260 master to indicate to PowerSpan II to load power-up options. This pin must be pulled high if the multiplexed system pin mechanism is used to load the power-up options.
PB_RST_	Bidirectional open drain (Schmitt trigger)	Low (if PB_RST_DIR=1, otherwise the signal is an input)	Pull-up resistor	**Reset:** Asynchronous active low reset. This is connected to HRESET_ pin of the MPC8260
PB_RST_DIR	Input	-	Power-up option	**Processor Bus Reset Direction**
PB_TA_	Tristate bidirectional	Hi-Z	Pull-up resistor	**Transfer Acknowledge:** Indicates that a data beat is valid on the data bus. For single beat transfers, it indicates the termination of the transfer. For burst transfers, it will be asserted four times to indicate the transfer of four data beats with the last assertion indicating the termination of the burst transfer.

Table 60: Processor Bus Signals

Pin Name	Pin Type	Reset State	Recommended Termination	Description
PB_TBST_	Tristate bidirectional	Hi-Z	Pull-up resistor	**Transfer Burst:** The bus master asserts this pin to indicate that the current transaction is a burst transaction
PB_TEA_	Tristate bidirectional	Hi-Z	Pull-up resistor	**Transfer Error Acknowledge:** Indicates a bus error.
PB_TSIZ[0:3]	Tristate bidirectional	Hi-Z	Pull-down resistor on TSIZ[0][b]	**Transfer Size:** Indicates the number of bytes to be transferred during a bus cycle.
PB_TS_	Tristate bidirectional	Hi-Z	Pull-up resistor	**Transfer Start:** Indicates the beginning of a new address bus tenure.
PB_TT[0:4]	Tristate bidirectional	Hi-Z	No requirement[a]	**Transfer Type:** The bus master drives these pins to specify the type of the transaction.
PB_VDDA	Supply	-	-	**PB Analog VDD:** Voltage supply pin to the analog circuits in the PB Phase Locked Loop (nominally 2.5V).
PB_DVDD	Supply	-	-	**PB Digital VDD:** Voltage supply pin to the digital circuits in the PB Phase Locked Loop (nominally 2.5V).
PB_DVSS	Ground	-	-	**PB Digital VSS:** Ground pin to the digital circuits in the PB Phase Locked Loop.
PB_AVSS	Ground	-	-	**PB Analog VSS:** Ground pin to the digital circuits in the PB Phase Locked Loop.

a. Pull-up resistors are not required on the processor bus address (PB_A[0:31]) and data (PB_D[0:63]) signals to guarantee functional operation of the PowerSpan II. However, adding resistors to the address and data signals minimizes the current drawn by the PowerSpan II's tristated buffers when the bus is in an idle condition. The system designer must decide whether to add these resistors to the address and data bus.

b. A pull-up resistor must be added to the signal if a external master that does not support MPC8260 extended cycles is in the system. If this type of external master is in the system, TSIZ[0] must be disconnected and a pull-down resistor must be used on the signal. Refer to E. "Typical Applications" on page 481 for a description and illustration of this type of system.

11.1.3 PCI-1 Signals

This section describes PowerSpan II signals used to interface to PCI-1. Signals in this group are compatible with both 3V and 5V signaling environments — as defined by the *PCI 2.2 Specification*.

Table 61 summarizes the signals in this grouping. Signals with electrical characteristics different from the remainder of the group are placed at the end of the table.

Table 61: PCI-1 Signals[a]

Pin Name	Pin Type	Description
P1_AD [63:0]	Tristate bidirectional	**PCI-1 Address/Data Bus:** Address and data are multiplexed over these pins providing a 64-bit address/data bus. [b]
P1_ACK64#	Tristate bidirectional	**PCI-1 Acknowledge 64-bit Transaction:** Active low signal asserted by a target to indicate its willingness to participate in a 64-bit transaction. Driven by the target; sampled by the master. Rescinded by the target at the end of the transaction.
P1_CBE[7:0]#	Tristate bidirectional	**PCI-1 Bus Command and Byte Enable Lines:** Command and byte enable information is multiplexed over all eight CBE lines.
P1_DEVSEL#	Tristate bidirectional	**PCI-1 Device Select:** An active low indication from an agent that is the target of the current transaction. Driven by the target; sampled by the master. Rescinded by the target at the end of the transaction.
P1_FRAME#	Tristate bidirectional	**PCI-1 Cycle Frame for PCI Bus:** An active low indication from the current bus master of the beginning and end of a transaction. Driven by the bus master; sampled by the selected target. Rescinded by the bus master at the end of the transaction.
P1_GNT[1]#	Tristate bidirectional	**PCI-1 Grant:** This is an input when an external arbiter is used and an output when the internal arbiter is used. As input it is used by the external arbiter to grant the bus to PowerSpan II. As output it is used by the internal arbiter to grant the bus to an external master. This pin must be weakly pulled high.

Table 61: PCI-1 Signals[a]

Pin Name	Pin Type	Description
P1_GNT [4:2]#	Tristate output	**PCI-1 Grant:** These are outputs only. They are used by the PCI-1 internal arbiter to grant the bus to external masters.
PCI_GNT [7:5]#	Tristate output	**PCI-1 Grant:** These outputs may be driven by the PCI-1 or PCI-2 internal arbiter to grant the bus to external masters. They are assigned to PCI-1 or PCI-2 by software. These pins should be weakly pulled high in a system.
P1_IDSEL	Input	**PCI-1 Initialization Device Select:** Used as a chip select during Configuration read and write transactions.
P1_INTA#	Bidirectional open drain	**PCI-1 Interrupt A:** An active low level sensitive indication of an interrupt. Asynchronous to P1_CLK.
P1_IRDY#	Tristate bidirectional	**PCI-1 Initiator Ready:** An active low indication of the current bus master's ability to complete the current dataphase. Driven by the master; sampled by the selected target.
P1_PAR	Tristate bidirectional	**PCI-1 Parity:** Carries even parity across P1_AD[31:0] and P1_C/BE[3:0]. Driven by the master for the address and write dataphases. Driven by the target for read dataphases.
P1_PAR64	Tristate bidirectional	**PCI-1 Parity Upper Dword:** Carries even parity across P1_AD[63:32] and P1_CBE[7:4]. Driven by the master for address and write dataphases. Driven by the target for read dataphases.
P1_CLK	Input	**PCI-1 Clock: Clock input for the PCI-1 Interface:** P1_CLK operates between 25 and 66MHz.
P1_M66EN	Input	**PCI-1 66 MHz Enable:** When pulled low, configures the PCI-1 PLL for operation between 25 and 33 MHz. When pulled high, configures the PCI-1 Interface PLL for operation above 33 MHz to a maximum of 66 MHz.
P1_PERR#	Tristate bidirectional	**PCI-1 Parity Error:** An active low indication of a data parity error. Driven by the target receiving data. Rescinded by that agent at the end of the transaction.
P1_REQ[1]#	Tristate bidirectional	**PCI-1 Bus Request:** This is an output when an external arbiter is used and an input when the PCI-1 internal arbiter is used. As input it is used by an external master to request the bus. As output it is used by PowerSpan II to request the bus. This pin must be weakly pulled high.
P1_REQ[4:2]#	Input	**PCI-1 Bus Request:** These are inputs only. They can be used by external masters to request the bus through the PCI-1 arbiter. These pins should be weakly pulled high in a system.

Table 61: PCI-1 Signals[a]

Pin Name	Pin Type	Description
PCI_REQ [7:5]#	Input	**PCI-1 Bus Request:** These inputs are used by external masters to request the bus from the PCI-1 or PCI-2 arbiter. They are assigned to PCI-1 or PCI-2 by software. These pins must be weakly pulled high in a system.
P1_REQ64#	Tristate bidirectional	**PCI-1 Request 64-bit Transfer:** An active low indication from the current master of its choice to perform 64-bit transactions. Rescinded by the bus master at the end of the transaction.
P1_RST#	Tristate bidirectional	**PCI-1 Reset:** Asynchronous active low reset for PCI-1 Interface
P1_SERR#	Open drain	**PCI-1 System Error:** An active low indication of address parity error.
P1_STOP#	Tristate bidirectional	**PCI-1 Stop:** An active low indication from the target of its desire to stop the current transition. Sampled by the master. Rescinded by the target at the end of the transaction.
P1_TRDY#	Tristate bidirectional	**PCI-1 Target Ready:** An active low indication of the current target's ability to complete the dataphase. Driven by the target; sampled by the current bus master. Rescinded by the target at the end of the transaction.
P1_64EN#	Input	**PCI-1 64-bit Enable:** An active low indication that a CompactPCI Hot Swap board is in a 64-bit slot. This signal must be pulled high in a non-Hot Swap environment.
P1_RST_DIR	Input (LVTTL)	**PCI-1 Bus Reset Direction**
P1_VDDA	Supply	**PCI-1 Analog VDD:** Voltage supply pin to the analog circuits in the PCI-1 Phase Locked Loop (nominally 2.5V).
P1_DVDD	Supply	**PCI-1 Digital VDD:** Voltage supply pin to the digital circuits in the PCI-1 Phase Locked Loop (nominally 2.5V).
P1_DVSS	Ground	**PCI-1 Digital VSS:** Ground pin to the digital circuits in the PCI-1 Phase Locked Loop.
P1_AVSS	Ground	**PCI-1 Analog VSS:** Ground pin to the digital circuits in the PCI-1 Phase Locked Loop.

[a] Refer to the *PCI Local Bus Specification* for reset states and recommended terminations of these PCI signals.

[b] To use the PowerSpan II Dual PCI in a 32-bit environment, add a pull-up resistor to P1_AD[32:63].

11.1.4 PCI-2 Signals

This section describes PowerSpan II signals used to interface to PCI-2. Signals in this group are compatible with both 3V and 5V signaling environments — as defined by the *PCI 2.2 Specification*.

These signals are not implemented in the Single PCI PowerSpan II.

Table 62 below summarizes the signals in this grouping. Signals with electrical characteristics different from the remainder of the group are placed at the end of the table.

Table 62: PCI-2 Signals[a]

Pin Name	Pin Type	Description
P2_AD[31:0]	Tristate bidirectional	**PCI-2 Address/Data Bus:** Address and data are multiplexed over these pins providing a 32-bit address/data bus.
P2_CBE[3:0]#	Tristate bidirectional	**PCI-2 Bus Command and Byte Enable Lines:** Command and byte enable information is multiplexed over all four CBE lines.
P2_DEVSEL#	Tristate bidirectional	**PCI-2 Device Select:** An active low indication from an agent that is the target of the current transaction. Driven by the target; sampled by the master. Rescinded by the target at the end of the transaction.
P2_FRAME#	Tristate bidirectional	**PCI-2 Cycle Frame for PCI Bus:** An active low indication from the current bus master of the beginning and end of a transaction. Driven by the bus master, sampled by the selected target. Rescinded by the bus master at the end of the transaction.
P2_GNT[1]#	Tristate bidirectional	**PCI-2 Grant:** This is an input when an external arbiter is used and an output when the PCI-2 internal arbiter is used. As input it is used by the external arbiter to grant the bus to PowerSpan II. As output it is used by the PCI-2 internal arbiter to grant the bus to an external master. This pin must be weakly pulled high in a system.
P2_GNT [4:2]#	Tristate output	**PCI-2 Grant:** These are outputs only. They are used by the PCI-2 internal arbiter to grant the bus to external masters. These pins must be weakly pulled high in a system.
P2_IDSEL	Input	**PCI-2 Initialization Device Select:** Used as a chip select during *Configuration* read and write transactions
P2_INTA#	Bidirectional open drain	**PCI -2 Interrupt A:** An active low level sensitive indication of an interrupt. Asynchronous to P2_CLK

Table 62: PCI-2 Signals[a]

Pin Name	Pin Type	Description
P2_IRDY#	Tristate bidirectional	**PCI-2 Initiator Ready:** An active low indication of the current bus master's ability to complete the current dataphase. Driven by the master; sampled by the selected target.
P2_PAR	Tristate bidirectional	**PCI-2 Parity:** Carries even parity across P2_AD[31:0] and P2_C/BE[3:0]. Driven by the master for the address and write dataphases. Driven by the target for read dataphases.
P2_CLK	Input	**PCI-2 Clock:** Clock input for the PCI-2 Interface. P2_CLK operates between 25 and 66MHz.
P2_M66EN	Input	**PCI-2 66 MHz Enable:** When pulled low, configures the PCI-2 PLL for operation between 25 and 33 MHz. When pulled high, configures the PCI-2 Interface PLL for operation above 33 MHz to a maximum of 66 MHz.
P2_PERR#	Tristate bidirectional	**PCI-2 Parity Error:** An active low indication of a data parity error. Driven by the target receiving data. Rescinded by that agent at the end of the transaction.
P2_REQ[1]	Tristate bidirectional	**PCI-2 Bus Request:** This is an output when an external arbiter is used and an input when the PCI-2 Interface internal arbiter is used. As input it is used by an external master to request the bus. As output it is used by PowerSpan II to request the bus. This pin must be weakly pulled high.
P2_REQ[4:2]	Input	**PCI-2 Bus Request:** These are inputs only. They can be used by external masters to request the bus from the PCI-2 arbiter. These pins must be weakly pulled high in a system.
P2_RST#	Tristate bidirectional	**PCI-2 Reset:** Asynchronous active low reset for PCI-2 Interface.
P2_SERR#	Open drain	**PCI-2 System Error:** An active low indication of address parity error.
P2_STOP#	Tristate bidirectional	**PCI-2 Stop:** An active low indication from the target of its desire to stop the current transition. Sampled by the master. Rescinded by the target at the end of the transaction.
P2_TRDY#	Tristate bidirectional	**PCI-2 Target Ready:** An active low indication of the current target's ability to complete the dataphase. Driven by the target; sampled by the current bus master. Rescinded by the target at the end of the transaction.

Table 62: PCI-2 Signals[a]

Pin Name	Pin Type	Description
P2_RST_DIR	Input (LVTTL)	**PCI-2 Bus Reset Direction**
P2_VDDA	Supply	**PCI-2 Analog VDD:** Voltage supply pin to the analog circuits in the PCI-2 Phase Locked Loop (nominally 2.5V).
P2_DVDD	Supply	**PCI-2 Digital VDD:** Voltage supply pin to the digital circuits in the PCI-1 Phase Locked Loop (nominally 2.5V).
P2_DVSS	Ground	**PCI-2 Digital VSS:** Ground pin to the digital circuits in the PCI-1 Phase Locked Loop.
P2_AVSS	Ground	**PCI-2 Analog VSS:** Ground pin to the digital circuits in the PCI-1 Phase Locked Loop.

[a] Refer to the *PCI Local Bus Specification* for reset states and recommended terminations of these PCI signals.

11.1.5 Miscellaneous Signals

Table 63 below lists PowerSpan II signals which are not necessarily dedicated to the PB, PCI-1 or PCI-2 Interfaces. They have a variety of electrical capabilities.

Table 63: Miscellaneous Signals

Pin Name	Pin Type	Reset State	Recommended Termination	Description
INT[5:0]_	Bidirectional open drain (5V tolerant LVTTL) (Schmitt trigger)	Hi-Z	Pull-up resistor	**Interrupt:** General purpose interrupt pins
ENUM#	Open drain output (PCI)	Hi-Z	Pull-up resistor if the application is a system host. Otherwise there is no resistor requirement on the signal.	**System Enumeration:** Used to notify system host that a board has been freshly inserted or extracted from the system.
ES	Input (5V tolerant LVTTL) (Schmitt trigger)	-	Pull-down resistor in non-Hot Swap environment[a]	**Ejector Switch:** Indicates the status of Hot Swap board ejector switch. A logic high value indicates the switch is closed and it is in operation mode. This signal must be pulled low in a non-Hot Swap environment.
LED#	Open drain output (5V tolerant LVTTL)	Low	Pull-up resistor if the application is a system host. Otherwise there is no resistor requirement on the signal.[a]	**LED:** Controls the Hot Swap status LED.

Table 63: Miscellaneous Signals

Pin Name	Pin Type	Reset State	Recommended Termination	Description
HEALTHY#	Input (5 V tolerant LVTTL) (Schmitt trigger)	-	Pull-down resistor in non-Hot Swap environment[a]	**Board Healthy:** In a Hot Swap environment, indicates the board is ready to be released from reset and become an active agent on PCI. Negation of this signal resets all PowerSpan II resources, including PLL's. Additionally, all PowerSpan II outputs are tristated when this pin is negated; inputs and bidirects are inhibited. This signal must be pulled low in a non-Hot Swap application.
PO_RST_	Input (5 V tolerant LVTTL) (Schmitt trigger)	-	-	**Power On Reset:** Assertion of this signal resets all PowerSpan II resources, including PLL's.
I2C_SCLK	Open drain output (5 V tolerant LVTTL)	Hi-Z	Pull-up resistor	**Serial Clock:** EEPROM Serial clock. This pin must be pulled high even if an EEPROM is not installed on the board.
I2C_SDA	Bidirectional open drain (5 V tolerant LVTTL)	Hi-Z	Pull-up resistor	**Serial Data:** EPROM Serial data line. This pin must be pulled high even if an EEPROM is not installed on the board.
Vdd CORE	Supply	-	-	**Core Vdd:** Nominally 2.5V
Vdd I/O	Supply	-	-	**IO Vdd:** Nominally 3.3V
VSS	Supply	-	-	**Ground**

a. Refer to the *CompactPCI Hot Swap Specification* for information on these signals.

11.1.6 Test Signals

Table 64 lists PowerSpan II signals used to support silicon or board level testing.

Table 64: Test Signals

Pin Name	Pin Type	Reset State	Recommended Termination	Description
PI_TEST1	Input	Internal pull-down resistor	Pull-down resistor	**PLL Test 1:** Internal PLL test signal. This is for internal Tundra use.
PI_TEST2	Input	Internal pull-down resistor	Pull-down resistor	**PLL Test 2:** Internal PLL test signal. This is for internal Tundra use.
P2_TEST1	Input **Note:** This signal is present in both the Single PCI PowerSpan II and the Dual PCI PowerSpan II. The signal is used for both PCI-1 and PCI-2 internal testing.	Internal pull-down resistor	Pull-down resistor	**PLL Test 1:** Internal PLL test signal. This is for internal Tundra use.
P2_TEST2	Input	Internal pull-down resistor	Pull-down resistor	**PLL Test 2:** Internal PLL test signal. This is for internal Tundra use.
PB_TEST1	Input	Internal pull-down resistor	Pull-down resistor	**PLL Test 1:** Internal PLL test signal. This is for internal Tundra use.
PB_TEST2	Input	Internal pull-down resistor	Pull-down resistor	**PLL Test 2:** Internal PLL test signal. This is for internal Tundra use.
TCK	Input (LVTTL)	Hi-Z	-	**Test Clock (JTAG):** Used to clock state information and data into and out of the device during boundary scan.

Table 64: Test Signals

Pin Name	Pin Type	Reset State	Recommended Termination	Description
TMS	Input (LVTTL)	Internal pull-up resistor	-	**Test Mode Select (JTAG)**: Used to control the state of the Test Access Port controller
TDI	Input (LVTTL)	Internal pull-up resistor	-	**Test Data Input (JTAG):** Used (in conjunction with TCK) to shift data and instructions into the Test Access Port (TAP) in a serial bit stream.
TDO	Tristate output (LVTTL)	Hi-Z	-	**Test Data Output (JTAG):** Used (in conjunction with TCK) to shift data and instructions into the Test Access Port (TAP) in a serial bit stream.
TRST_	Input (LVTTL) (Schmitt trigger)	Internal pull-up resistor	Pull-down resistor if JTAG is not used in the system. Otherwise the signal must be toggled with the PO_RST_ signal.	**Test Reset (JTAG):** Asynchronous reset for the JTAG controller. This pin must be asserted during the power-up reset sequence to ensure that the Boundary Scan Register elements are configured for normal system operation. Customers must assert TRST _concurrently with PO_RST_ as part of the power-up reset sequence.
TE	Input	Internal pull-down resistor	Pull-down resistor	**Test Enable:** Enables manufacturing test. Tundra recommends that system designers pull this signal low.

11.2 Dual PCI PowerSpan II Pinout

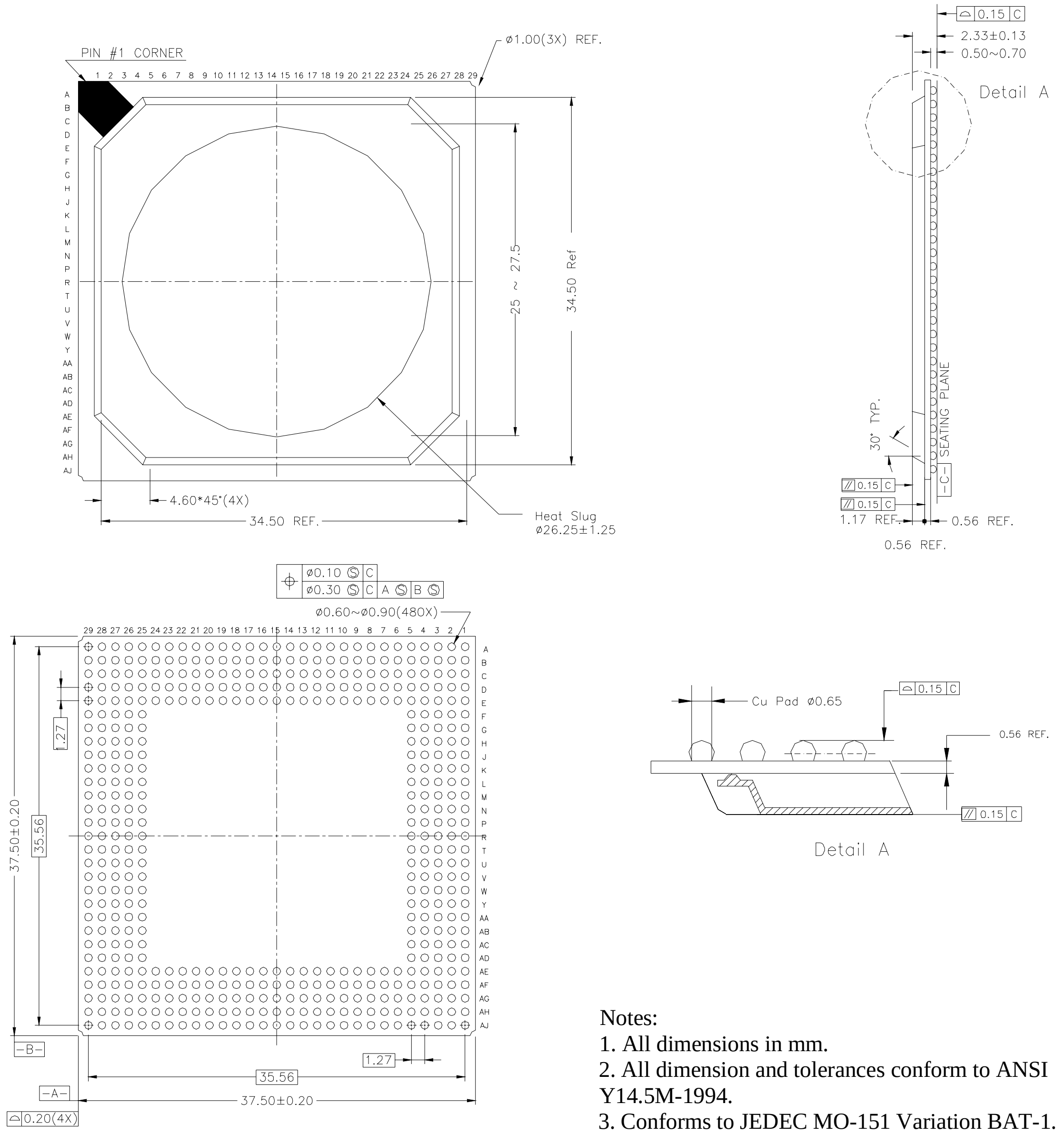
11.2.1 Dual PCI PowerSpan II 480 HSBGA

Figure 24 illustrates the top, side, and bottom views of the PowerSpan II package.

Table 65: Package Characteristics

Feature	Description
Package Type	480 HSBGA
Package Body Size	37.5mm
JEDEC Specification	JEDEC MO-151 Variation BAT-1

Figure 24: 480 HSBGA



Notes:
1. All dimensions in mm.
2. All dimension and tolerances conform to ANSI Y14.5M-1994.
3. Conforms to JEDEC MO-151 Variation BAT-1.

11.2.1.1 Package Notes

1. All dimensions in mm
2. All dimensions and tolerance conform to ANSI Y14.5M - 1994
3. Conforms to JEDEC MO-151 Variation BAT-1

11.2.2 480 HSBGA Pin Information

The following table shows the PowerSpan II 480 HSBGA, 1.27 mm package, pin information. This package is backwards compatible with the original PowerSpan's 480 HPBGA device.

A1. VSS_IO

A2. VSS_IO

A3. PB_A[12]

A4. PB_A[14]

A5. PB_A[16]

A6. PB_A[18]

A7. VSS_IO

A8. PB_A[21]

A9. PB_A[24]

A10. PB_A[27]

A11. PB_A[31]

A12. VSS_IO

A13. P1_AD[34]

A14. P1_AD[38]

A15. P1_AD[41]

A16. VSS_IO

A17. P1_AD[48]

A18. VSS_IO

A19. P1_AD[53]

A20. P1_AD[56]

A21. P1_AD[59]

A22. P1_AD[62]

A23. VSS_IO

A24. P1_CBE[6]_

A25. P1_VDDA

A26. P1_REQ64_

G25. VDD25

G26. P1_AD[9]

G27. PCI_GNT[5]_

G28. P1_GNT[4]_

G29. VSS_IO

H1. INT[4]_

H2. PB_A[0]

H3. PB_A[1]

H4. PB_BR2_

H5. VDD25

H25. VDD25

H26. VSS

H27. P1_AD[12]

H28. P1_AD[11]

H29. P1_AD[10]

J1. PB_TT[1]

J2. HEALTHY_

J3. PB_TT[2]

J4. PB_TT[3]

J5. VSS

J25. VSS

J26. VSS

J27. P1_INTA_

J28. PCI_GNT[6]_

J29. P1_AD[13]

K1. PB_AACK_

AC25. VDD25

AC26. P2_TEST1

AC27. P2_AD[9]

AC28. P2_AD[8]

AC29. VSS_IO

AD1. PB_D[60]

AD2. PB_D[52]

AD3. PB_D[44]

AD4. JT_TRST_

AD5. VDD25

AD25. VDD25

AD26. PCI_GNT[7]_

AD27. P2_AD[12]

AD28. P2_AD[11]

AD29. P2_AD[10]

AE1. PB_D[20]

AE2. PB_TEST1

AE3. PB_D[12]

AE4. PB_BG2_

AE5. PB_AVSS

AE6. PB_DVDD

AE7. VDD25

AE8. VDD25

AE9. VSS

AE10. VDD33

AE11. VDD33

A27. P1_AD[0]

A28. VSS_IO

A29. VSS_IO

B1. VSS_IO

B2. VSS_IO

B3. PB_A[13]

B4. PB_A[15]

B5. VSS_IO

B6. JT_TMS

B7. P1_RST_DIR

B8. PB_A[22]

B9. PB_A[26]

B10. PB_A[28]

B11. PB_CI_

B12. P1_AD[32]

B13. P1_AD[35]

B14. VSS_IO

B15. P1_AD[42]

B16. P1_AD[45]

B17. P1_AD[49]

B18. P1_AD[50]

B19. P1_AD[54]

B20. P1_AD[57]

B21. P1_IDSEL

B22. P1_AD[63]

B23. P1_CBE[4]_

B24. P1_CBE[7]_

B25. P1_TEST2

B26. P1_ACK64_

K2. PB_TT[4]

K3. VSS

K4. PB_TT[0]

K5. VDD33

K25. VDD33

K26. P1_PAR

K27. P1_CBE[1]_

K28. P1_AD[15]

K29. P1_AD[14]

L1. PB_ARTRY_

L2. PB_TSIZ[3]

L3. PB_BG1_

L4. PB_BR3_

L5. VDD33

L25. VDD33

L26. P1_TRDY_

L27. P1_DEVSEL_

L28. P1_STOP_

L29. P1_PERR_

M1. VSS_IO

M2. PB_TSIZ[2]

M3. PB_TSIZ[1]

M4. PB_TS_

M5. VDD33

M25. VDD33

M26. P1_CBE[2]_

M27. P1_FRAME_

M28. P1_IRDY_

M29. VSS_IO

AE12. VDD33

AE13. VSS

AE14. VDD25

AE15. VDD25

AE16. VDD25

AE17. VSS

AE18. VDD33

AE19. VDD33

AE20. VDD33

AE21. VSS

AE22. VDD25

AE23. VDD25

AE24. P2_DVDD

AE25. P2_AVSS

AE26. P2_AD[14]

AE27. P2_AD[13]

AE28. P2_TEST2

AE29. P2_IDSEL

AF1. PB_D[36]

AF2. PB_D[28]

AF3. INT[3]_

AF4. PB_DVSS

AF5. PB_CLK

AF6. INT[1]_

AF7. VSS

AF8. PB_VDDA

AF9. PB_D[26]

AF10. PB_D[57]

AF11. INT[0]_

B27. P1_CLK

B28. VSS_IO

B29. VSS_IO

C1. PB_A[11]

C2. TE

C3. VSS

C4. JT_TCK

C5. PB_A[17]

C6. PB_A[19]

C7. PB_DBG2_

C8. PB_A[23]

C9. PB_A[25]

C10. PB_A[29]

C11. PB_DBG1_

C12. P1_AD[33]

C13. P1_AD[36]

C14. P1_AD[39]

C15. P1_AD[43]

C16. P1_AD[46]

C17. P1_SERR_

C18. P1_AD[51]

C19. P1_AD[55]

C20. P1_REQ[4]_

C21. P1_AD[60]

C22. P1_PAR64

C23. P1_CBE[5]_

C24. P1_REQ1_

C25. P1_GNT1_

C26. P1_64EN_

N1. PB_AP[3]

N2. PB_TSIZ[0]

N3. I2C_SCLK

N4. PB_TBST_

N5. VSS

N25. VSS

N26. P1_AD[19]

N27. P1_AD[18]

N28. P1_AD[17]

N29. P1_AD[16]

P1. VSS_IO

P2. PB_AP[1]

P3. PB_AP[2]

P4. I2C_SDA

P5. VDD25

P25. VDD25

P26. P1_AD[22]

P27. P1_AD[21]

P28. VSS_IO

P29. P1_AD[20]

R1. PB_TA_

R2. PB_DVAL_

R3. PB_TEA_

R4. PB_AP[0]

R5. VDD25

R25. VDD25

R26. P1_AD[25]

R27. P1_AD[24]

R28. P1_CBE[3]_

AF12. PB_D[25]

AF13. PB_D[1]

AF14. PB_FAST

AF15. PB_D[24]

AF16. PB_DP[7]

AF17. PB_DP[4]

AF18. PB_DP[1]

AF19. P2_RST_

AF20. P2_AD[28]

AF21. P2_AD[24]

AF22. VSS

AF23. P2_AD[19]

AF24. P2_SERR_

AF25. P2_CLK

AF26. P2_DVSS

AF27. P2_INTA_

AF28. P2_CBE[1]_

AF29. P2_AD[15]

AG1. PB_D[43]

AG2. PB_D[35]

AG3. VSS

AG4. INT[2]_

AG5. PB_D[3]

AG6. PB_D[11]

AG7. PB_D[42]

AG8. PB_D[58]

AG9. PB_D[18]

AG10. PB_ABB_

AG11. PB_RSTCONF_

C27. VSS

C28. P1_AD[2]

C29. P1_AD[1]

D1. PB_A[8]

D2. PB_A[9]

D3. PB_A[10]

D4. VSS

D5. JT_TDI

D6. JT_TDO

D7. PB_A[20]

D8. VSS

D9. VSS

D10. PB_A[30]

D11. PB_BR1_

D12. PB_GBL_

D13. P1_AD[37]

D14. P1_AD[40]

D15. P1_AD[44]

D16. P1_AD[47]

D17. P1_M66EN

D18. P1_AD[52]

D19. P1_REQ[3]_

D20. P1_AD[58]

D21. P1_AD[61]

D22. ENUM_

D23. P1_TEST1

D24. P1_GNT[2]_

D25. P1_GNT[3]_

D26. P1_DVSS

R29. P1_AD[23]

T1. PB_D[15]

T2. VSS_IO

T3. PB_D[30]

T4. PB_D[39]

T5. VDD25

T25. VDD25

T26. P1_AD[28]

T27. P1_AD[27]

T28. P1_AD[26]

T29. VSS_IO

U1. LED_

U2. PB_D[7]

U3. PB_D[22]

U4. PB_D[47]

U5. VSS

U25. VSS

U26. P1_AD[31]

U27. VSS

U28. P1_AD[30]

U29. P1_AD[29]

V1. VSS_IO

V2. PB_D[6]

V3. PB_D[55]

V4. PB_D[23]

V5. VDD33

V25. VDD33

V26. P2_GNT[3]_

V27. P2_GNT[4]_

AG12. PB_D[17]

AG13. PB_RST_

AG14. PB_D[40]

AG15. PB_D[16]

AG16. PB_DP[6]

AG17. PB_DP[3]

AG18. PB_DP[0]

AG19. P2_AD[31]

AG20. P2_AD[27]

AG21. P2_CBE[3]_

AG22. P2_AD[22]

AG23. P2_AD[18]

AG24. P2_AD[17]

AG25. P2_FRAME_

AG26. P2_REQ[2]_

AG27. VSS

AG28. P2_REQ[3]_

AG29. P2_PAR

AH1. VSS_IO

AH2. VSS_IO

AH3. NC

AH4. PB_D[59]

AH5. PB_TEST2

AH6. PB_D[19]

AH7. PB_D[50]

AH8. PB_D[34]

AH9. VSS_IO

AH10. PB_D[49]

AH11. PB_D[41]

D27. P1_REQ[2]_

D28. P1_AD[4]

D29. P1_AD[3]

E1. PB_DBG3_

E2. VSS_IO

E3. PB_A[7]

E4. ES

E5. VSS

E6. VDD25

E7. VDD25

E8. VDD25

E9. VSS

E10. VDD33

E11. VDD33

E12. VDD33

E13. VSS

E14. VDD25

E15. VDD25

E16. VDD25

E17. VSS

E18. VDD33

E19. VDD33

E20. VDD33

E21. VSS

E22. VDD25

E23. VDD25

E24. P1_DVDD

E25. P1_AVSS

E26. PCI_REQ[5]_

V28. P1_RST_

V29. VSS_IO

W1. PB_D[31]

W2. PB_D[62]

W3. PB_D[54]

W4. PB_D[46]

W5. VDD33

W25. VDD33

W26. P2_AD[0]

W27. P2_REQ1_

W28. P2_GNT1_

W29. P2_GNT[2]_

Y1. PB_D[21]

Y2. PB_D[38]

Y3. PB_D[14]

Y4. PB_D[53]

Y5. VDD33

Y25. VDD33

Y26. P2_AD[3]

Y27. PCI_REQ[7]_

Y28. P2_AD[2]

Y29. P2_AD[1]

AA1. PB_D[37]

AA2. NC

AA3. PB_D[29]

AA4. VSS

AA5. VSS

AA25. VSS

AA26. P2_AD[5]

AH12. PB_D[9]

AH13. PB_D[56]

AH14. PB_D[32]

AH15. PB_D[8]

AH16. VSS_IO

AH17. PB_DP[2]

AH18. PB_D[63]

AH19. P2_AD[30]

AH20. P2_AD[26]

AH21. VSS_IO

AH22. P2_AD[21]

AH23. P2_VDDA

AH24. P2_AD[16]

AH25. VSS_IO

AH26. P2_TRDY_

AH27. P2_STOP_

AH28. VSS_IO

AH29. VSS_IO

AJ1. VSS_IO

AJ2. VSS_IO

AJ3. PB_D[51]

AJ4. PB_D[4]

AJ5. PB_DBB_

AJ6. PB_D[27]

AJ7. VSS_IO

AJ8. PB_D[10]

AJ9. PB_D[2]

AJ10. PO_RST_

AJ11. PB_D[33]

E27. P1_AD[6]

E28. VSS_IO

E29. P1_AD[5]

F1. PB_A[4]

F2. PB_A[5]

F3. PB_A[6]

F4. PB_RST_DIR

F5. VDD25

F25. VDD25

F26. PCI_REQ[6]_

F27. P1_AD[8]

F28. P1_CBE[0]_

F29. P1_AD[7]

G1. VSS_IO

G2. PB_A[2]

G3. PB_A[3]

G4. VSS

G5. VDD25

AA27. P2_AD[4]

AA28. VSS_IO

AA29. P2_M66EN

AB1. PB_D[5]

AB2. PB_D[61]

AB3. PB_D[45]

AB4. VSS

AB5. VDD25

AB25. VDD25

AB26. P2_REQ[4]_

AB27. P2_CBE[0]_

AB28. P2_AD[7]

AB29. P2_AD[6]

AC1. VSS_IO

AC2. PB_BG3_

AC3. INT[5]_

AC4. PB_D[13]

AC5. VDD25

AJ12. VSS_IO

AJ13. PB_D[48]

AJ14. VSS_IO

AJ15. PB_D[0]

AJ16. PB_DP[5]

AJ17. P2_RST_DIR

AJ18. VSS_IO

AJ19. P2_AD[29]

AJ20. P2_AD[25]

AJ21. P2_AD[23]

AJ22. P2_AD[20]

AJ23. VSS_IO

AJ24. P2_CBE[2]_

AJ25. P2_IRDY_

AJ26. P2_DEVSEL_

AJ27. P2_PERR_

AJ28. VSS_IO

AJ29. VSS_IO

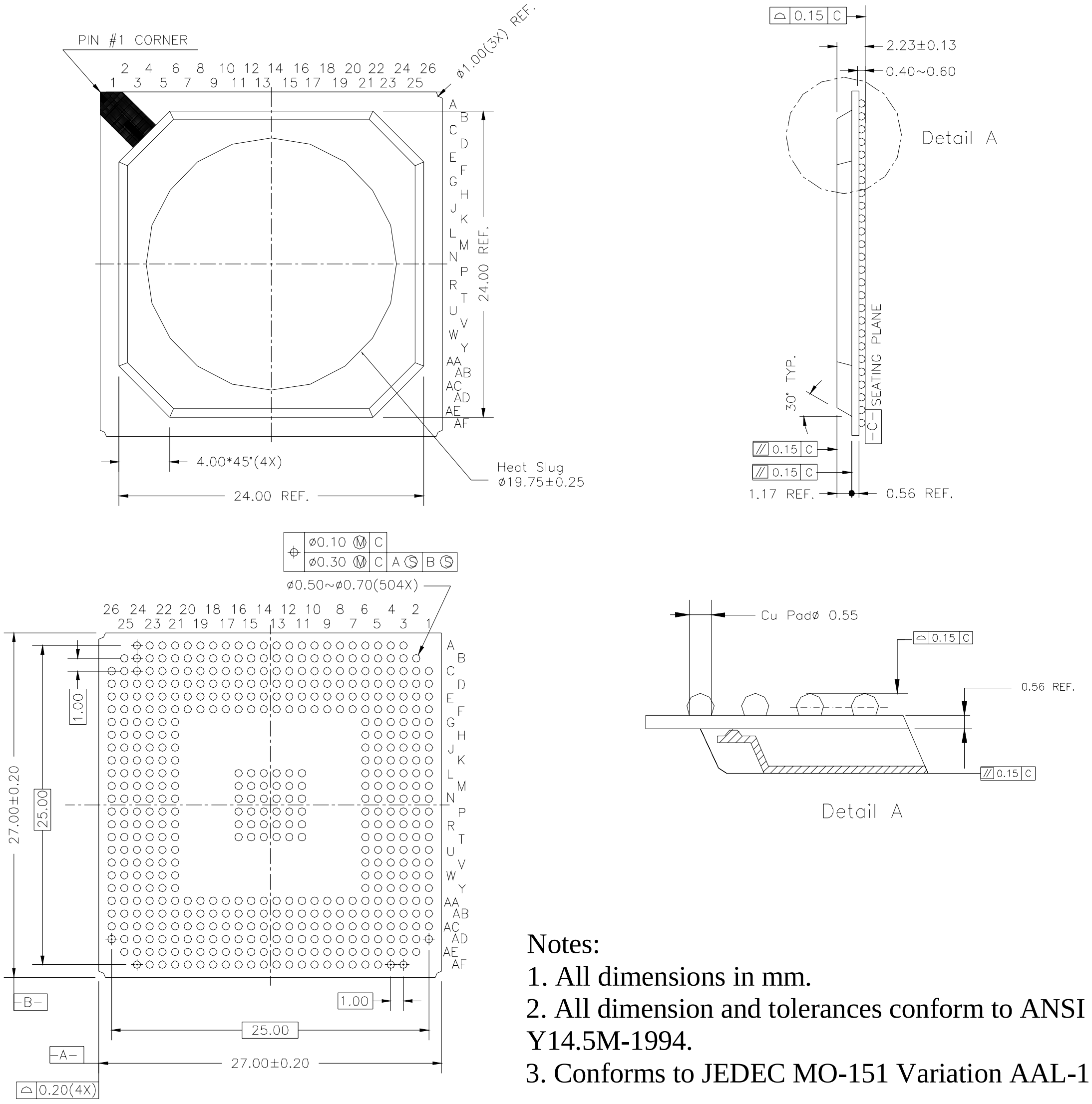
11.2.3 Dual PCI PowerSpan II 504 HSBGA

Figure 25 illustrates the top, side, and bottom views of the PowerSpan II package.

Table 66: Package Characteristics

Feature	Description
Package Type	504 HSBGA
Package Body Size	27mm
JEDEC Specification	JEDEC MO-151 Variation AAL-1

Figure 25: 504 HSBGA



Notes:
1. All dimensions in mm.
2. All dimension and tolerances conform to ANSI Y14.5M-1994.
3. Conforms to JEDEC MO-151 Variation AAL-1.

11.2.3.1 Package Notes

1. All dimensions in mm
2. All dimensions and tolerance conform to ANSI Y14.5M - 1994
3. Conforms to JEDEC MO-151 Variation AAL-1

11.2.4 504 HSBGA Pin Information

The following table shows the PowerSpan II 504 HSBG, 1.00 mm package, pin information.

A3. JT_TDO

A4. PB_A[19]

A5. JT_TMS

A6. PB_A[21]

A7. PB_A[24]

A8. PB_CI_

A9. P1_AD[33]

A10. P1_AD[32]

A11. P1_AD[35]

A12. P1_AD[40]

A13. P1_AD[42]

A14. P1_AD[43]

A15. P1_AD[45]

A16. P1_SERR_

A17. P1_AD[52]

A18. P1_AD[53]

A19. P1_AD[56]

A20. P1_AD[58]

A21. P1_AD[61]

A22. P1_CBE[7]_

A23. P1_GNT[2]_

A24. P1_ACK64_

B2. VSS_IO

B3. VSS_IO

B4. PB_A[15]

B5. PB_A[20]

H21. VDD25

H22. VDD25

H23. P1_CBE[0]_

H24. P1_INTA_

H25. P1_AD[15]

H26. P1_CBE[2]_

J1. PB_ARTRY_

J2. PB_BR3_

J3. INT[4]_

J4. PB_BR2_

J5. VDD25

J6. VDD25

J21. VDD25

J22. VDD25

J23. P1_GNT[4]_

J24. P1_AD[13]

J25. P1_DEVSEL_

J26. P1_IRDY_

K1. PB_TS_

K2. PB_TSIZ[3]

K3. PB_TT[1]

K4. PB_TT[2]

K5. VDD33

K6. VDD33

K21. VDD33

K22. VDD33

W21. VDD25

W22. VDD25

W23. P2_IDSEL

W24. P2_AD[5]

W25. PCI_REQ[7]_

W26. P2_REQ[1]_

Y1. PB_D[53]

Y2. PB_D[61]

Y3. PB_D[60]

Y4. JT_TRST_

Y5. PB_D[35]

Y6. VDD33

Y21. VDD33

Y22. P2_PAR

Y23. P2_AD[15]

Y24. P2_AD[8]

Y25. P2_AD[6]

Y26. P2_M66EN

AA1. PB_D[45]

AA2. INT[5]_

AA3. PB_D[20]

AA4. PB_D[28]

AA5. PB_BG2_

AA6. PB_DVDD

AA7. VDD33

AA8. VDD25

B6. PB_A[23]

B7. PB_A[25]

B8. PB_A[29]

B9. PB_DBG1_

B10. PB_GBL_

B11. P1_AD[36]

B12. P1_AD[34]

B13. P1_AD[41]

B14. P1_AD[44]

B15. P1_AD[49]

B16. P1_M66EN

B17. P1_AD[54]

B18. P1_REQ[3]_

B19. P1_AD[59]

B20. P1_AD[63]

B21. P1_CBE[6]_

B22. P1_REQ64_

B23. P1_64EN_

B24. VSS_IO

B25. VSS_IO

C1. PB_A[9]

C2. VSS_IO

C3. VSS

C4. PB_A[13]

C5. PB_A[12]

C6. PB_DBG2_

C7. P1_RST_DIR

C8. PB_A[22]

C9. PB_A[30]

K23. PCI_GNT[6]_

K24. P1_TRDY_

K25. P1_FRAME_

K26. P1_AD[19]

L1. I2C_SCLK

L2. PB_TBST_

L3. PB_BG1_

L4. PB_TT[4]

L5. VDD33

L6. VDD33

L11. VSS_IO

L12. VSS_IO

L13. VSS_IO

L14. VSS_IO

L15. VSS_IO

L16. VSS_IO

L21. VDD33

L22. VDD33

L23. P1_AD[14]

L24. P1_PERR_

L25. P1_AD[16]

L26. P1_AD[22]

M1. PB_AP[1]

M2. PB_AP[3]

M3. PB_TSIZ[1]

M4. PB_TSIZ[0]

M5. PB_TSIZ[2]

M6. VSS

M11. VSS_IO

AA9. VDD25

AA10. VDD33

AA11. VDD33

AA12. VSS

AA13. VSS

AA14. VSS

AA15. VSS

AA16. VDD33

AA17. VDD33

AA18. VDD25

AA19. VDD25

AA20. VDD33

AA21. P2_DVDD

AA22. P2_INTA_

AA23. P2_AD[13]

AA24. P2_AD[9]

AA25. P2_CBE[0]_

AA26. P2_AD[7]

AB1. PB_D[13]

AB2. PB_D[12]

AB3. INT[3]_

AB4. PB_TEST1

AB5. VSS_IO

AB6. INT[2]_

AB7. PB_TEST2

AB8. VDD25

AB9. VDD25

AB10. VDD33

AB11. VDD33

C10. PB_A[28]

C11. PB_A[31]

C12. P1_AD[37]

C13. P1_AD[39]

C14. P1_AD[47]

C15. P1_AD[50]

C16. P1_AD[55]

C17. P1_REQ[4]_

C18. P1_AD[62]

C19. P1_PAR64

C20. P1_CBE[4]_

C21. P1_CBE[5]_

C22. P1_CLK

C23. P1_AVSS

C24. P1_VDDA

C25. VSS_IO

C26. P1_AD[4]

D1. PB_A[6]

D2. PB_A[11]

D3. TE

D4. VSS

D5. JT_TDI

D6. PB_A[17]

D7. PB_A[14]

D8. PB_A[16]

D9. PB_A[18]

D10. PB_A[26]

D11. PB_A[27]

D12. P1_AD[38]

M12. VSS_IO

M13. VSS_IO

M14. VSS_IO

M15. VSS_IO

M16. VSS_IO

M21. VSS

M22. P1_STOP_

M23. P1_AD[17]

M24. P1_AD[18]

M25. P1_AD[21]

M26. P1_AD[23]

N1. PB_DVAL_

N2. PB_TA_

N3. PB_AP[2]

N4. I2C_SDA

N5. VSS

N6. VSS

N11. VSS_IO

N12. VSS_IO

N13. VSS_IO

N14. VSS_IO

N15. VSS_IO

N16. VSS_IO

N21. VSS

N22. VSS

N23. PCI_REQ[5]_

N24. P1_AD[20]

N25. P1_CBE[3]_

N26. P1_AD[24]

AB12. PB_D[49]

AB13. VSS

AB14. VSS

AB15. P2_CBE[3]_

AB16. VDD33

AB17. VDD33

AB18. VDD25

AB19. VDD25

AB20. P2_REQ[3]_

AB21. P2_TEST1

AB22. VSS_IO

AB23. P2_TEST2

AB24. P2_AD[14]

AB25. P2_AD[12]

AB26. P2_AD[10]

AC1. PB_D[36]

AC2. VSS_IO

AC3. VSS_IO

AC4. VSS_IO

AC5. PB_DVSS

AC6. INT[1]_

AC7. PB_D[3]

AC8. PB_D[11]

AC9. PB_D[42]

AC10. PB_D[10]

AC11. PO_RST_

AC12. PB_D[25]

AC13. PB_D[40]

AC14. PB_DP[5]

D13. P1_AD[46]
D14. P1_AD[48]
D15. P1_AD[51]
D16. P1_AD[57]
D17. P1_AD[60]
D18. ENUM_
D19. P1_REQ[1]_
D20. P1_GNT[1]_
D21. P1_GNT[3]_
D22. P1_DVSS
D23. VSS_IO
D24. VSS_IO
D25. VSS_IO
D26. P1_AD[8]
E1. PB_A[3]
E2. PB_RST_DIR
E3. PB_A[10]
E4. ES
E5. VSS
E6. VSS
E7. JT_TCK
E8. VDD25
E9. VDD25
E10. VDD33
E11. VDD33
E12. PB_BR1_
E13. VSS
E14. VSS
E15. P1_IDSEL

P1. PB_TEA_
P2. PB_D[15]
P3. PB_D[39]
P4. PB_AP[0]
P5. VSS
P6. VSS
P11. VSS_IO
P12. VSS_IO
P13. VSS_IO
P14. VSS_IO
P15. VSS_IO
P16. VSS_IO
P21. VSS
P22. VSS
P23. P1_AD[28]
P24. PCI_GNT[7]_
P25. PCI_REQ[6]_
P26. P1_AD[25]
R1. PB_D[30]
R2. LED_
R3. PB_D[6]
R4. PB_D[47]
R5. PB_D[37]
R6. VSS
R11. VSS_IO
R12. VSS_IO
R13. VSS_IO
R14. VSS_IO
R15. VSS_IO

AC15. PB_DP[4]
AC16. P2_AD[26]
AC17. P2_AD[21]
AC18. P2_IRDY_
AC19. P2_DEVSEL_
AC20. P2_TRDY_
AC21. P2_PERR_
AC22. P2_DVSS
AC23. VSS_IO
AC24. VSS_IO
AC25. VSS_IO
AC26. P2_AD[11]
AD1. PB_D[43]
AD2. VSS_IO
AD3. PB_VDDA
AD4. PB_AVSS
AD5. PB_CLK
AD6. PB_D[19]
AD7. PB_D[50]
AD8. PB_D[26]
AD9. PB_D[57]
AD10. INT[0]_
AD11. PB_D[33]
AD12. PB_RST_
AD13. PB_D[32]
AD14. P2_RST_DIR
AD15. PB_D[63]
AD16. P2_AD[31]
AD17. P2_AD[27]

E16. VDD33

E17. VDD33

E18. VDD25

E19. VDD25

E20. P1_TEST2

E21. P1_TEST1

E22. VSS_IO

E23. P1_AD[0]

E24. P1_AD[2]

E25. P1_AD[9]

E26. P1_AD[7]

F1. PB_A[0]

F2. PB_A[2]

F3. PB_DBG3_

F4. PB_A[7]

F5. VSS

F6. VDD33

F7. VDD33

F8. VDD25

F9. VDD25

F10. VDD33

F11. VDD33

F12. VSS

F13. VSS

F14. VSS

F15. VSS

F16. VDD33

F17. VDD33

F18. VDD25

R16. VSS_IO

R21. VSS

R22. P2_AD[1]

R23. P1_AD[29]

R24. P1_AD[31]

R25. P1_AD[27]

R26. PCI_GNT[5]_

T1. PB_D[7]

T2. PB_D[22]

T3. PB_D[62]

T4. PB_D[21]

T5. VDD33

T6. VDD33

T11. VSS_IO

T12. VSS_IO

T13. VSS_IO

T14. VSS_IO

T15. VSS_IO

T16. VSS_IO

T21. VDD33

T22. VDD33

T23. P2_AD[0]

T24. P2_GNT[2]_

T25. P1_AD[30]

T26. P1_AD[26]

U1. PB_D[55]

U2. PB_D[31]

U3. PB_D[38]

U4. PB_D[5]

AD18. P2_AD[20]

AD19. P2_CBE[2]_

AD20. P2_AD[18]

AD21. P2_AD[19]

AD22. P2_CLK

AD23. P2_AVSS

AD24. P2_VDDA

AD25. VSS_IO

AD26. P2_CBE[1]_

AE2. VSS_IO

AE3. VSS_IO

AE4. PB_D[51]

AE5. PB_D[4]

AE6. PB_D[58]

AE7. PB_D[18]

AE8. PB_ABB_

AE9. PB_RSTCONF_

AE10. PB_D[17]

AE11. PB_D[56]

AE12. PB_FAST

AE13. PB_D[16]

AE14. PB_DP[6]

AE15. PB_DP[2]

AE16. PB_DP[0]

AE17. P2_AD[30]

AE18. P2_RST_

AE19. P2_AD[23]

AE20. P2_AD[24]

AE21. P2_AD[16]

F19. VDD25

F20. VDD33

F21. P1_DVDD

F22. P1_REQ[2]_

F23. P1_AD[1]

F24. P1_AD[5]

F25. P1_AD[11]

F26. P1_AD[10]

G1. HEALTHY_

G2. PB_TT[3]

G3. PB_A[4]

G4. PB_A[8]

G5. VSS

G6. VDD33

G21. VDD33

G22. P1_AD[6]

G23. P1_AD[3]

G24. P1_AD[12]

G25. P1_PAR

G26. P1_CBE[1]_

H1. PB_AACK_

H2. PB_TT[0]

H3. PB_A[1]

H4. PB_A[5]

H5. VDD25

H6. VDD25

U5. VDD33

U6. VDD33

U21. VDD33

U22. VDD33

U23. P2_AD[4]

U24. P2_AD[2]

U25. P2_GNT[3]_

U26. P1_RST_

V1. PB_D[23]

V2. PB_D[46]

V3. PB_D[29]

V4. PB_D[52]

V5. VDD25

V6. VDD25

V21. VDD25

V22. VDD25

V23. P2_REQ[4]_

V24. P2_AD[3]

V25. P2_GNT[1]_

V26. P2_GNT[4]_

W1. PB_D[54]

W2. PB_D[14]

W3. PB_BG3_

W4. PB_D[44]

W5. VDD25

W6. VDD25

AE22. P2_SERR_

AE23. P2_REQ[2]_

AE24. VSS_IO

AE25. VSS_IO

AF3. PB_D[59]

AF4. PB_DBB_

AF5. PB_D[27]

AF6. PB_D[34]

AF7. PB_D[2]

AF8. PB_D[41]

AF9. PB_D[9]

AF10. PB_D[1]

AF11. PB_D[48]

AF12. PB_D[24]

AF13. PB_D[8]

AF14. PB_D[0]

AF15. PB_DP[7]

AF16. PB_DP[3]

AF17. PB_DP[1]

AF18. P2_AD[29]

AF19. P2_AD[25]

AF20. P2_AD[28]

AF21. P2_AD[22]

AF22. P2_AD[17]

AF23. P2_FRAME_

AF24. P2_STOP_

11.3 Single PCI PowerSpan II Pin Information

The PowerSpan II Single PCI device is offered in two packages. The 484 PBGA package is offered with a 23 mm body size and 1.00 mm ball pitch.The 420 HSBGA package is offered with a 35 mm body size and 1.27 mm ball pitch. The 35 mm body size is the same as the original PowerSpan package offering.

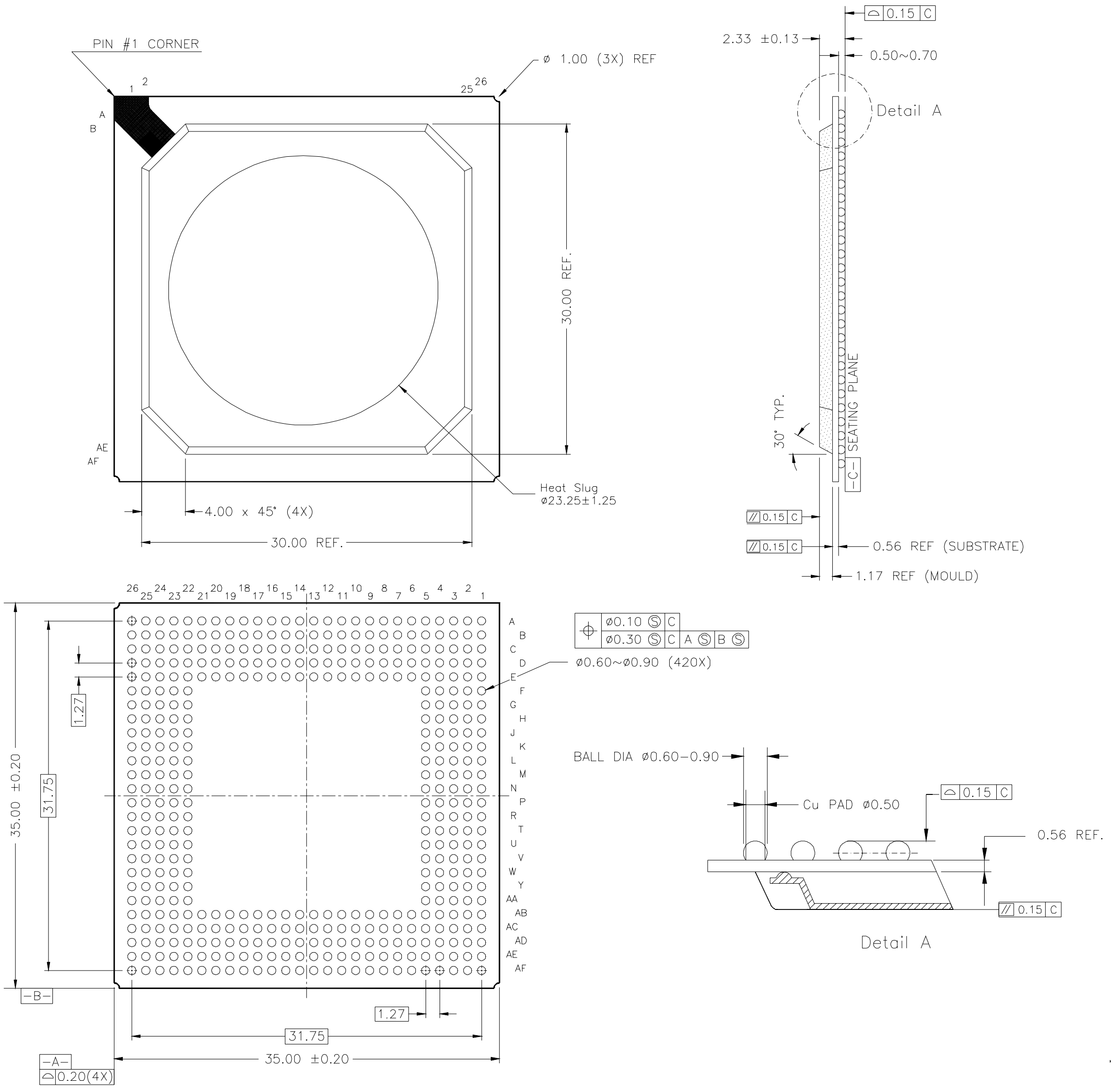
11.3.1 Single PCI PowerSpan II 420 HSBGA

Figure 26 illustrates the top, side, and bottom views of the PowerSpan II package.

Table 67: Package Characteristics

Feature	Description
Package Type	420 HSBGA
Package Body Size	35mm
JEDEC Specification	JEDEC MO-151 Variation BAT-1

Figure 26: 420 HSBGA



11.3.1.1 Package Notes

1. All dimensions in mm
2. All dimensions and tolerance conform to ANSI Y14.5M - 1994
3. Conforms to JEDEC MS-034 Variation BAR-1

11.3.2 420 HSBGA Pin Information

The following table shows the PowerSpan II 420 HSBGA, 1.27 mm package, pin information. This package is backwards compatible with the original PowerSpan's 420 HPBGA device.

A1. VSS_IO

A2. VSS_IO

A3. JT_TDI

A4. JT_TDO

A5. JT_TCK

A6. VSS_IO

A7. PB_A[30]

A8. P1_RST_DIR

A9. VSS_IO

A10. PB_DBG1_

A11. VSS_IO

A12. P1_AD[34]

A13. P1_AD[40]

A14. P1_AD[44]

A15. P1_AD[47]

A16. VSS_IO

A17. P1_AD[50]

A18. P1_AD[51]

A19. P1_REQ[4]_

A20. P1_AD[59]

A21. VSS_IO

A22. P1_VDDA

A23. P1_CBE[7]_

A24. P1_GNT[3]_

A25. P1_CLK

A26. VSS_IO

G1. PB_TT[3]

G2. PB_A[1]

G3. PB_A[5]

G4. PB_A[3]

G5. VDD33

G22. VDD33

G23. P1_AD[6]

G24. P1_AD[1]

G25. P1_AD[4]

G26. P1_AD[3]

H1. PB_BR3_

H2. PB_TT[2]

H3. PB_A[4]

H4. PB_A[2]

H5. VDD33

H22. VDD33

H23. P1_AD[8]

H24. P1_AD[9]

H25. PCI_GNT[5]_

H26. P1_TEST2

J1. HEALTHY_

J2. VSS_IO

J3. PB_A[0]

J4. INT[4]_

J5. VSS

J22. VSS

AA1. VSS_IO

AA2. PB_D[36]

AA3. PB_D[28]

AA4. PB_D[12]

AA5. VDD33

AA22. VDD33

AA23. VSS

AA24. VSS

AA25. VSS_IO

AA26. VSS_IO

AB1. INT[5]_

AB2. PB_D[44]

AB3. PB_D[43]

AB4. PB_D[35]

AB5. VSS

AB6. VDD33

AB7. VDD33

AB8. VDD33

AB9. VSS

AB10. PB_DVDD

AB11. VDD25

AB12. VDD25

AB13. VSS_IO

AB14. VSS_IO

AB15. VDD25

AB16. VDD25

B1. TE

B2. VSS_IO

B3. PB_A[13]

B4. PB_A[19]

B5. PB_A[20]

B6. PB_A[18]

B7. PB_A[21]

B8. PB_A[26]

B9. PB_DBG2_

B10. PB_BR1_

B11. P1_AD[33]

B12. P1_AD[32]

B13. P1_AD[39]

B14. P1_AD[41]

B15. P1_AD[48]

B16. P1_AD[49]

B17. P1_AD[52]

B18. P1_REQ[3]_

B19. P1_AD[56]

B20. P1_IDSEL

B21. ENUM_

B22. NC

B23. P1_GNT[1]_

B24. P1_64EN_

B25. VSS_IO

B26. VSS_IO

C1. PB_RST_DIR

C2. PB_A[10]

C3. VSS_IO

J23. P1_CBE[0]_

J24. P1_AD[7]

J25. P1_GNT[4]_

J26. PCI_GNT[6]_

K1. PB_AACK_

K2. PB_TT[4]

K3. PB_TT[0]

K4. PB_TT[1]

K5. VDD25

K22. P1_DVDD

K23. P1_AD[12]

K24. P1_AD[11]

K25. P1_AD[10]

K26. P1_AD[13]

L1. VSS_IO

L2. PB_ARTRY_

L3. PB_TSIZ[3]

L4. PB_BG1_

L5. VDD25

L22. VDD25

L23. P1_INTA_

L24. P1_PAR

L25. P1_CBE[1]_

L26. VSS_IO

M1. PB_TS_

M2. PB_TSIZ[1]

M3. PB_TBST_

M4. PB_TSIZ[2]

M5. VDD25

AB17. VDD25

AB18. VSS

AB19. VDD33

AB20. VDD33

AB21. VDD33

AB22. VSS

AB23. PCI_GNT[7]_

AB24. VSS_IO

AB25. VSS_IO

AB26. NC

AC1. JT_TRST_

AC2. PB_D[53]

AC3. PB_TEA_

AC4. PB_AVSS

AC5. NC

AC6. PB_D[3]

AC7. LED_

AC8. VSS

AC9. PB_D[19]

AC10. PB_D[27]

AC11. PB_D[18]

AC12. PB_D[57]

AC13. PB_D[41]

AC14. PB_D[1]

AC15. PB_FAST

AC16. PB_D[24]

AC17. PB_DP[6]

AC18. PB_DP[3]

AC19. PB_D[63]

C4. PB_A[17]

C5. PB_A[15]

C6. PB_A[16]

C7. PB_A[22]

C8. PB_A[25]

C9. PB_A[29]

C10. PB_A[27]

C11. PB_GBL_

C12. P1_AD[37]

C13. VSS_IO

C14. NC

C15. P1_SERR_

C16. P1_AD[53]

C17. P1_AD[57]

C18. P1_AD[60]

C19. P1_AD[61]

C20. P1_CBE[6]_

C21. P1_CBE[5]_

C22. P1_REQ[1]_

C23. P1_AD[46]

C24. VSS_IO

C25. PB_CI_

C26. PCI_REQ[5]_

D1. PB_A[8]

D2. PB_A[11]

D3. PB_A[7]

D4. VSS

D5. PB_A[12]

D6. PB_A[14]

M22. VDD25

M23. P1_AD[15]

M24. P1_AD[14]

M25. P1_AD[5]

M26. P1_AD[2]

N1. I2C_SCLK

N2. PB_TSIZ[0]

N3. PB_AP[3]

N4. PB_AP[2]

N5. VSS_IO

N22. VSS_IO

N23. P1_FRAME_

N24. P1_CBE[2]_

N25. P1_PERR_

N26. P1_STOP_

P1. PB_AP[1]

P2. NC

P3. PB_TA_

P4. PB_AP[0]

P5. VSS_IO

P22. VSS_IO

P23. P1_AD[17]

P24. P1_AD[18]

P25. P1_AD[19]

P26. P1_IRDY_

R1. VSS_IO

R2. PB_D[47]

R3. PB_D[22]

R4. PB_D[7]

AC20. PB_DP[5]

AC21. VSS

AC22. PB_D[8]

AC23. VSS

AC24. VSS_IO

AC25. P1_RST_

AC26. P1_AD[31]

AD1. INT[3]_

AD2. PB_D[37]

AD3. PB_DVSS

AD4. PB_D[31]

AD5. PB_D[38]

AD6. PB_D[14]

AD7. PB_D[11]

AD8. PB_D[51]

AD9. PB_D[42]

AD10. PB_D[50]

AD11. PB_D[34]

AD12. PB_D[2]

AD13. PB_D[33]

AD14. PB_D[9]

AD15. PB_D[48]

AD16. P2_TEST1

AD17. PB_DP[7]

AD18. PB_DP[2]

AD19. PB_DP[4]

AD20. PB_DP[1]

AD21. PB_D[0]

AD22. VSS

D7. JT_TMS

D8. PB_A[23]

D9. PB_A[24]

D10. PB_A[28]

D11. PB_A[31]

D12. P1_AD[35]

D13. P1_AD[43]

D14. P1_AD[45]

D15. P1_M66EN

D16. P1_AD[54]

D17. P1_AD[58]

D18. P1_AD[63]

D19. P1_PAR64

D20. P1_CBE[4]_

D21. VSS

D22. P1_GNT[2]_

D23. P1_AVSS

D24. P1_AD[42]

D25. P1_AD[36]

D26. P1_REQ[2]_

E1. PB_DBG3_

E2. VSS_IO

E3. PB_A[9]

E4. ES

E5. VSS

E6. VDD33

E7. VDD33

E8. VDD33

E9. VSS

R5. VDD25

R22. VDD25

R23. P1_TEST1

R24. P1_AD[21]

R25. P1_AD[22]

R26. P1_AD[16]

T1. VSS_IO

T2. PB_D[6]

T3. PB_D[55]

T4. PB_D[23]

T5. VDD25

T22. VDD25

T23. P1_AD[24]

T24. P1_CBE[3]_

T25. P1_AD[20]

T26. VSS_IO

U1. PB_D[30]

U2. PB_D[54]

U3. PB_D[46]

U4. PB_D[21]

U5. VDD25

U22. VDD25

U23. NC

U24. P1_AD[23]

U25. P1_AD[25]

U26. VSS_IO

V1. PB_BG3_

V2. NC

V3. PB_D[29]

AD23. VSS_IO

AD24. VSS_IO

AD25. NC

AD26. VSS_IO

AE1. VSS_IO

AE2. VSS_IO

AE3. INT[2]_

AE4. PB_D[39]

AE5. PB_DBB_

AE6. NC

AE7. PB_D[15]

AE8. PB_D[59]

AE9. PB_VDDA

AE10. PB_D[58]

AE11. PB_D[10]

AE12. VSS_IO

AE13. PB_RSTCONF_

AE14. PB_D[17]

AE15. PB_D[56]

AE16. PB_D[40]

AE17. NC

AE18. NC

AE19. VSS_IO

AE20. PB_DP[0]

AE21. PB_D[32]

AE22. VSS_IO

AE23. VSS_IO

AE24. VSS_IO

AE25. VSS_IO

E10. VDD25

E11. VDD25

E12. VDD25

E13. VSS_IO

E14. VSS_IO

E15. VDD25

E16. VDD25

E17. VDD25

E18. VSS

E19. VDD33

E20. VDD33

E21. VDD33

E22. P1_DVSS

E23. P1_AD[62]

E24. P1_AD[38]

E25. P1_AD[55]

E26. PCI_REQ[6]_

F1. VSS_IO

F2. PB_BR2_

F3. NC

F4. PB_A[6]

F5. VDD33

F22. VDD33

F23. P1_ACK64_

F24. P1_AD[0]

F25. P1_REQ64_

F26. VSS_IO

V4. PB_D[5]

V5. VSS

V22. VSS

V23. P1_AD[27]

V24. P1_AD[26]

V25. VSS_IO

V26. VSS_IO

W1. PB_D[62]

W2. PB_TEST2

W3. PB_D[45]

W4. PB_D[60]

W5. VDD33

W22. VDD33

W23. P1_AD[30]

W24. P1_AD[29]

W25. P1_AD[28]

W26. PCI_REQ[7]_

Y1. PB_D[52]

Y2. PB_D[20]

Y3. PB_TEST1

Y4. PB_D[13]

Y5. VDD33

Y22. VDD33

Y23. VSS

Y24. P1_DEVSEL_

Y25. P1_TRDY_

Y26. VSS_IO

AE26. NC

AF1. VSS_IO

AF2. PB_BG2_

AF3. INT[1]_

AF4. PB_D[61]

AF5. PB_CLK

AF6. VSS_IO

AF7. PB_DVAL_

AF8. PB_ABB_

AF9. I2C_SDA

AF10. PB_D[26]

AF11. VSS_IO

AF12. PB_D[49]

AF13. PO_RST_

AF14. PB_D[25]

AF15. PB_RST_

AF16. VSS_IO

AF17. PB_D[16]

AF18. INT[0]_

AF19. VSS_IO

AF20. PB_D[4]

AF21. VSS_IO

AF22. NC

AF23. NC

AF24. NC

AF25. VSS_IO

AF26. VSS_IO

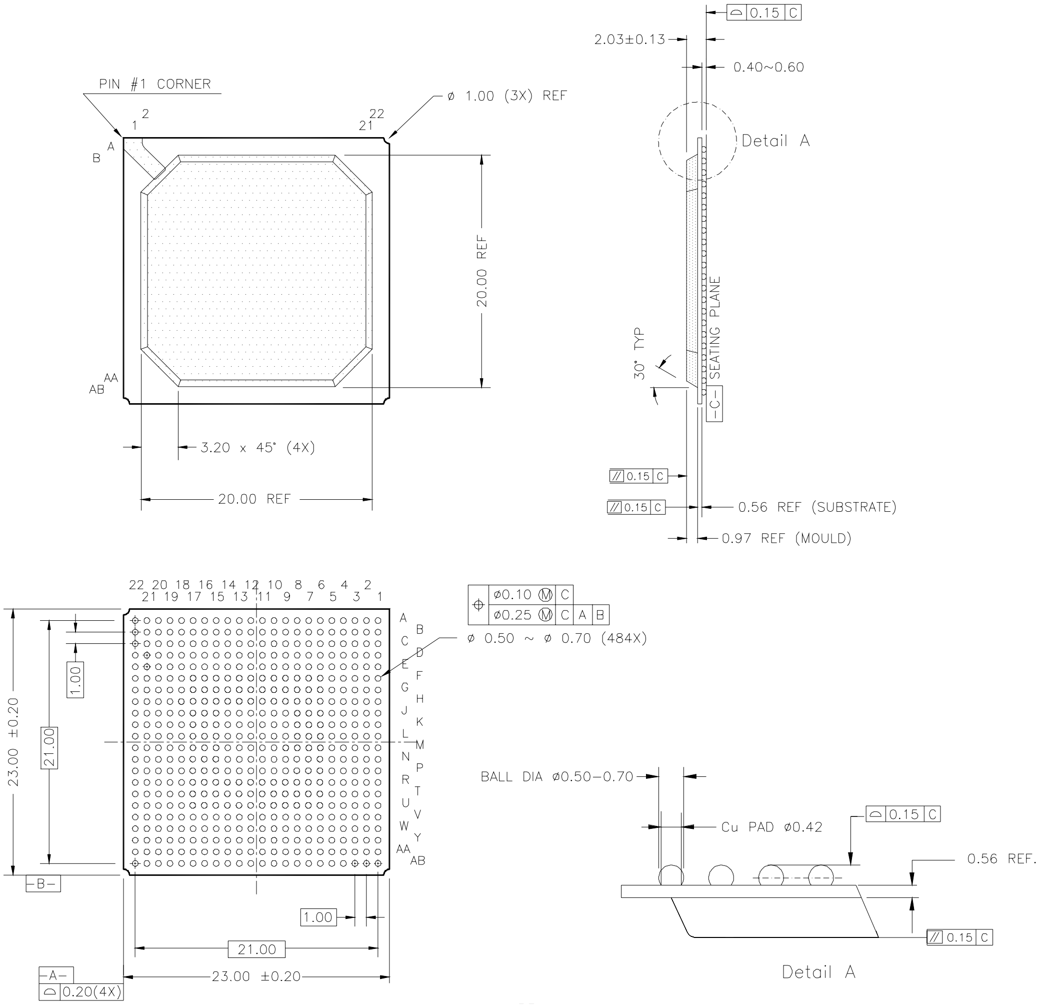
11.3.3 Single PCI PowerSpan II 484 HSBGA

Figure 27 illustrates the top, side, and bottom views of the PowerSpan II package.

Table 68: Package Characteristics

Feature	Description
Package Type	484 HSBGA
Package Body Size	23mm
JEDEC Specification	JEDEC MS-034 Variation AAJ-1

Figure 27: 484 PBGA



11.3.3.1 Package Notes

1. All dimensions in mm
2. All dimensions and tolerance conform to ANSI Y14.5M - 1994
3. Conforms to JEDEC MS-034 Variation AAJ-1

11.3.3.2 484 PBGA Pin Information

The following table shows the PowerSpan II 484 PBGA, 1.00 mm package, pin information.

A1. PB_A[6]

A2. PB_A[13]

A3. TE

A4. JT_TMS

A5. PB_A[20]

A6. JT_TCK

A7. PB_A[24]

A8. PB_A[30]

A9. PB_A[27]

A10. PB_DBG2_

A11. PB_GBL_

A12. P1_AD[34]

A13. P1_AD[32]

A14. P1_AD[40]

A15. P1_AD[48]

A16. P1_AD[53]

A17. P1_AD[52]

A18. P1_AD[58]

A19. P1_AD[60]

A20. P1_PAR64

A21. ENUM_

A22. VSS_IO

B1. PB_RST_DIR

B2. ES

B3. PB_A[14]

B4. PB_A[12]

H9. VSS_IO

H10. VSS_IO

H11. VSS_IO

H12. VSS_IO

H13. VSS_IO

H14. VSS_IO

H15. VSS_IO

H16. VSS

H17. VDD33

H18. VDD25

H19. PCI_REQ[5]_

H20. P1_AD[0]

H21. P1_REQ64_

H22. P1_AD[8]

J1. PB_TSIZ[2]

J2. PB_TBST_

J3. PB_ARTRY_

J4. PB_BG1_

J5. VDD25

J6. VDD33

J7. VSS

J8. VSS_IO

J9. VSS_IO

J10. VSS_IO

J11. VSS_IO

J12. VSS_IO

R17. VDD33

R18. VDD25

R19. P1_AD[22]

R20. P1_AD[17]

R21. P1_AD[5]

R22. P1_PERR_

T1. PB_TEST2

T2. PB_D[21]

T3. PB_D[52]

T4. JT_TRST_

T5. VDD25

T6. VDD33

T7. VSS

T8. VSS

T9. VSS

T10. VSS

T11. VSS

T12. VSS

T13. VSS

T14. VSS

T15. VSS

T16. VSS

T17. VDD33

T18. VDD25

T19. P1_AD[25]

T20. P1_AD[18]

B5. JT_TDO

B6. PB_A[22]

B7. PB_A[21]

B8. PB_A[28]

B9. PB_A[31]

B10. PB_BR1_

B11. P1_AD[35]

B12. P1_AD[37]

B13. P1_AD[43]

B14. P1_AD[39]

B15. P1_AD[47]

B16. P1_AD[49]

B17. P1_AD[57]

B18. P1_AD[56]

B19. P1_CBE[6]_

B20. P1_AD[59]

B21. VSS_IO

B22. P1_VDDA

C1. PB_A[11]

C2. PB_BR2_

C3. VSS_IO

C4. JT_TDI

C5. PB_A[15]

C6. PB_A[16]

C7. PB_A[18]

C8. PB_A[25]

C9. P1_RST_DIR

C10. PB_DBG1_

C11. P1_AD[33]

J13. VSS_IO

J14. VSS_IO

J15. VSS_IO

J16. VSS

J17. VDD33

J18. VDD25

J19. PCI_REQ[6]_

J20. P1_CBE[0]_

J21. P1_AD[4]

J22. P1_AD[9]

K1. PB_AP[2]

K2. PB_AP[3]

K3. PB_TSIZ[0]

K4. PB_AACK_

K5. VDD25

K6. VDD33

K7. VSS

K8. VSS_IO

K9. VSS_IO

K10. VSS_IO

K11. VSS_IO

K12. VSS_IO

K13. VSS_IO

K14. VSS_IO

K15. VSS_IO

K16. VSS

K17. VDD33

K18. VDD25

K19. P1_AD[1]

T21. P1_STOP_

T22. P1_AD[19]

U1. PB_D[20]

U2. PB_D[45]

U3. PB_D[36]

U4. PB_D[43]

U5. VDD25

U6. VDD33

U7. VDD33

U8. VDD33

U9. VDD33

U10. VDD33

U11. VDD33

U12. VDD33

U13. VDD33

U14. VDD33

U15. VDD33

U16. VDD33

U17. VDD33

U18. VDD25

U19. PCI_REQ[7]_

U20. P1_AD[21]

U21. P1_AD[16]

U22. P1_IRDY_

V1. PB_D[5]

V2. INT[5]_

V3. PB_D[44]

V4. PB_D[37]

V5. PB_TEA_

C12. P1_AD[41]

C13. P1_AD[44]

C14. P1_SERR_

C15. P1_AD[50]

C16. P1_AD[51]

C17. P1_REQ[3]_

C18. P1_AD[63]

C19. P1_AD[61]

C20. VSS_IO

C21. P1_DVSS

C22. P1_AVSS

D1. PB_A[2]

D2. PB_A[8]

D3. PB_DBG3_

D4. VSS_IO

D5. PB_A[10]

D6. PB_A[17]

D7. PB_A[19]

D8. PB_A[23]

D9. PB_A[26]

D10. PB_A[29]

D11. VDD25

D12. P1_AD[45]

D13. P1_M66EN

D14. P1_AD[54]

D15. P1_REQ[4]_

D16. P1_IDSEL

D17. P1_CBE[7]_

D18. P1_CBE[5]_

K20. P1_AD[7]

K21. P1_AD[12]

K22. P1_AD[3]

L1. I2C_SCLK

L2. PB_AP[1]

L3. PB_TSIZ[1]

L4. PB_TS_

L5. VDD25

L6. VDD33

L7. VSS

L8. VSS_IO

L9. VSS_IO

L10. VSS_IO

L11. VSS_IO

L12. VSS_IO

L13. VSS_IO

L14. VSS_IO

L15. VSS_IO

L16. VSS

L17. VDD33

L18. VDD25

L19. VDD25

L20. P1_TEST2

L21. P1_INTA_

L22. PCI_GNT[5]_

M1. PB_D[47]

M2. PB_TA_

M3. PB_AP[0]

M4. VDD25

V6. PB_D[35]

V7. VDD25

V8. VDD25

V9. VDD25

V10. VDD25

V11. VDD25

V12. VDD25

V13. VDD25

V14. VDD25

V15. VDD25

V16. VDD25

V17. VDD25

V18. VDD25

V19. P1_DEVSEL_

V20. P1_CBE[3]_

V21. P1_AD[20]

V22. P1_TEST1

W1. PB_D[60]

W2. PB_D[53]

W3. PB_D[13]

W4. VSS_IO

W5. PB_BG2_

W6. INT[1]_

W7. INT[2]_

W8. PB_D[61]

W9. PB_D[42]

W10. PB_D[59]

W11. PB_D[18]

W12. VDD25

D19. VSS_IO

D20. P1_CLK

D21. P1_DVDD

D22. PB_CI_

E1. INT[4]_

E2. PB_A[1]

E3. PB_A[5]

E4. PB_A[7]

E5. VSS_IO

E6. VDD25

E7. VDD25

E8. VDD25

E9. VDD25

E10. VDD25

E11. VDD25

E12. VDD25

E13. VDD25

E14. VDD25

E15. VDD25

E16. VDD25

E17. VDD25

E18. P1_CBE[4]_

E19. P1_GNT[1]_

E20. P1_AD[46]

E21. P1_AD[42]

E22. P1_ACK64_

F1. PB_A[0]

F2. PB_TT[2]

F3. PB_A[4]

M5. VDD25

M6. VDD33

M7. VSS

M8. VSS_IO

M9. VSS_IO

M10. VSS_IO

M11. VSS_IO

M12. VSS_IO

M13. VSS_IO

M14. VSS_IO

M15. VSS_IO

M16. VSS

M17. VDD33

M18. VDD25

M19. P1_AD[10]

M20. P1_AD[13]

M21. P1_AD[11]

M22. P1_GNT[4]_

N1. PB_D[7]

N2. PB_D[6]

N3. PB_D[22]

N4. PB_D[46]

N5. VDD25

N6. VDD33

N7. VSS

N8. VSS_IO

N9. VSS_IO

N10. VSS_IO

N11. VSS_IO

W13. PB_D[48]

W14. PB_D[56]

W15. PB_DP[2]

W16. PB_D[63]

W17. PB_D[8]

W18. PCI_GNT[7]_

W19. VDD25

W20. P1_AD[27]

W21. P1_AD[26]

W22. P1_AD[24]

Y1. PB_D[12]

Y2. PB_D[28]

Y3. VSS_IO

Y4. PB_DVSS

Y5. PB_D[31]

Y6. PB_D[3]

Y7. PB_D[11]

Y8. PB_D[51]

Y9. PB_D[27]

Y10. PB_D[10]

Y11. PB_D[58]

Y12. PB_D[41]

Y13. PB_RSTCONF_

Y14. PB_D[1]

Y15. P2_TEST1

Y16. PB_D[16]

Y17. PB_DP[3]

Y18. PB_DP[0]

Y19. PB_DP[1]

F4. PB_A[9]

F5. VDD25

F6. VSS_IO

F7. VDD33

F8. VDD33

F9. VDD33

F10. VDD33

F11. VDD33

F12. VDD33

F13. VDD33

F14. VDD33

F15. VDD33

F16. VDD33

F17. VDD33

F18. P1_REQ[1]_

F19. P1_GNT[2]_

F20. P1_AD[62]

F21. P1_AD[36]

F22. P1_AD[38]

G1. PB_TT[1]

G2. PB_TT[0]

G3. PB_BR3_

G4. PB_A[3]

G5. VDD25

G6. VDD33

G7. VSS_IO

G8. VSS

G9. VSS

G10. VSS

N12. VSS_IO

N13. VSS_IO

N14. VSS_IO

N15. VSS_IO

N16. VSS

N17. VDD33

N18. VDD25

N19. P1_CBE[2]_

N20. PCI_GNT[6]_

N21. P1_AD[15]

N22. P1_PAR

P1. PB_D[30]

P2. PB_D[55]

P3. PB_D[54]

P4. PB_D[23]

P5. VDD25

P6. VDD33

P7. VSS

P8. VSS_IO

P9. VSS_IO

P10. VSS_IO

P11. VSS_IO

P12. VSS_IO

P13. VSS_IO

P14. VSS_IO

P15. VSS_IO

P16. VSS

P17. VDD33

P18. VDD25

Y20. P1_TRDY_

Y21. P1_AD[23]

Y22. P1_AD[28]

AA1. INT[3]_

AA2. VSS_IO

AA3. PB_AVSS

AA4. PB_CLK

AA5. PB_D[38]

AA6. PB_D[14]

AA7. PB_DVAL_

AA8. PB_ABB_

AA9. PB_D[50]

AA10. PB_D[34]

AA11. PB_D[2]

AA12. PB_D[49]

AA13. PO_RST_

AA14. PB_D[17]

AA15. PB_RST_

AA16. PB_D[40]

AA17. INT[0]_

AA18. PB_DP[4]

AA19. PB_DP[5]

AA20. PB_D[32]

AA21. P1_AD[30]

AA22. P1_AD[29]

AB1. VSS_IO

AB2. PB_VDDA

AB3. PB_DVDD

AB4. PB_D[39]

G11. VSS

G12. VSS

G13. VSS

G14. VSS

G15. VSS

G16. VSS

G17. VDD33

G18. P1_GNT[3]_

G19. P1_64EN_

G20. P1_AD[6]

G21. P1_AD[55]

G22. P1_REQ[2]_

H1. PB_TT[4]

H2. PB_TSIZ[3]

H3. HEALTHY_

H4. PB_TT[3]

H5. VDD25

H6. VDD33

H7. VSS

H8. VSS_IO

P19. P1_AD[2]

P20. P1_CBE[1]_

P21. P1_FRAME_

P22. P1_AD[14]

R1. PB_D[62]

R2. PB_BG3_

R3. PB_D[29]

R4. PB_TEST1

R5. VDD25

R6. VDD33

R7. VSS

R8. VSS_IO

R9. VSS_IO

R10. VSS_IO

R11. VSS_IO

R12. VSS_IO

R13. VSS_IO

R14. VSS_IO

R15. VSS_IO

R16. VSS

AB5. PB_DBB_

AB6. LED_

AB7. PB_D[15]

AB8. PB_D[19]

AB9. I2C_SDA

AB10. PB_D[26]

AB11. PB_D[57]

AB12. PB_D[33]

AB13. PB_D[25]

AB14. PB_D[9]

AB15. PB_FAST

AB16. PB_D[24]

AB17. PB_DP[7]

AB18. PB_DP[6]

AB19. PB_D[4]

AB20. PB_D[0]

AB21. P1_RST_

AB22. P1_AD[31]

12. Electrical and Signal Characteristics

This chapter describes the electrical characteristics of the PowerSpan II device. It also details the pin-outs of both the Single PCI PowerSpan II and Dual PCI PowerSpan II. The following topics are discussed:

- "Electrical Characteristics" on page 277
- "Power Dissipation" on page 279
- "Operating Conditions" on page 280

12.1 Electrical Characteristics

PowerSpan II's electrical characteristics are defined by PCI electrical characteristics and non-PCI electrical characteristics.

12.1.1 PCI Electrical Characteristics

PowerSpan II's PCI interfaces are electrically compatible with the 3.3V and the 5.0V signaling interfaces as defined by the *PCI 2.2 Specification*.

PowerSpan II supports the *CompactPCI Hot Swap Specification Revision 2.0* and is classified as Hot Swap Silicon. PowerSpan II is compliant with the *PCI Local Bus Specification Revision 2.2* regarding device accessibility after release of LOCAL_PCI_RST_ through Initially Retrying. Optionally devices can choose to Initially Not Respond after release.

12.1.2 Non-PCI Electrical Characteristics

The following table, Table 69, specifies the required DC characteristics of all non-PCI PowerSpan II signal pins.

Table 69: HBGA Electrical Characteristics (non-PCI)[a]

Symbol	Parameter	Condition	Min	Max	Units
V_{IL}	Input low voltage	$V_{OUT} \geq V_{OH}$(min) or $V_{OUT} \leq V_{OL}$(max)	- 0.3	0.8	V
V_{IH}	Input high voltage (5 V tolerant LVTTL)		2.0	V_{DD} + 0.3	V
Iin	Input leakage Current	V_{IN} = 0 V or $V_{IN} = V_{DD}$		5	µA
Iin	Input leakage Current (internal pull-up)	V_{IN} = 0 V or $V_{IN} = V_{DD}$	-2.0	-100	µA
Iin	Input leakage Current (internal pull-down)	V_{IN} = 0 V or $V_{IN} = V_{DD}$	2.0	100	µA
V_{OH}[b]	Output high voltage	V_{DD} = min, I_{OH} = -2mA	2.4		V
V_{OL}[b]	Output low voltage	V_{DD} = min, I_{OH} = 2mA		0.4	V
C_{IN}	Input Capacitance			10	pF
I_{OL}[c]	Output Low Current (65 ohm output)	V_{OL}=1.5V	25	100	mA

a. Non-PCI DC Electrical Characteristics (Ta= -40°C to 85°C)

b. These electrical characteristics comply with *JESD8B Interface Standard for Nominal 3 V/3.3 V Supply Digital Integrated Circuits*

c. CompactPCI Hot Swap LED pin

12.2 Power Dissipation

Table 70 shows the Single PCI PowerSpan II power dissipation..

Table 70: Single PCI PowerSpan II Power Dissipation

Processor Bus Clock	PCI-1 Clock	Maximum
50 MHz	33 MHz	1.2 W
66 MHz	33 MHz	1.3 W
100 MHz	66 MHz	2.3 W

Table 71 shows the Dual PCI PowerSpan II power dissipation

Table 71: Dual PCI PowerSpan II Power Dissipation

Processor Bus Clock	PCI-1 Clock	PCI-2 Clock	Maximum
50 MHz	33 MHz	25 MHz	1.2 W
66 MHz	33 MHz	33 MHz	1.3 W
100 MHz	66 MHz	66 MHz	2.3 W

12.3 Operating Conditions

12.3.1 Recommended Operating Conditions

The following table, Table 72, specifies the recommended operating conditions of the PowerSpan II.

Table 72: Operating and Storage Conditions

Symbol	Parameter	Min	Max	Units
Vdd I/O	I/O DC Supply Voltage	3.15	3.45	V
Vdd Core	Core Supply Voltage	2.38	2.63	V
Px_VDDA	PLL Supply Voltage	2.38	2.63	V
Ta	Ambient Temperature	-40	+85	°C
H	Humidity	0	80	% Relative Humidity

12.3.2 Handling and Storage Specifications

After encapsulation cure cavity down assemblies are JESD22-A112 Moisture Sensitivity Category 3. From that point on, parts are to be handled to the following criteria:

1. From encapsulation cure to BGA/PGA attach, Class 3 modules can be exposed for a maximum cumulative time of eight days to an environment of no more than 30 OC and/or 60% RH. If this condition is exceeded, modules are to be baked at
   125 O C +/- 10 OC for 24 hours minimum or at other qualified bake parameters.
2. Modules shall be placed in an ESD carrier. Each module shall be orientated in the same way. When preparing for shipment to stock location, modules are to be baked at 125 O C +/- 10 O C for 24 hours minimum or at other qualified bake parameters. Modules are to be sealed within 24 hours for Class 3 modules in a moisture barrier ESD bag with desiccant. This bakeout procedure can be repeated once to remain in compliance.
3. Failure to comply with this requirement can result in die to die pad, encapsulant to soldermask or encapsulant to die delamination during reflow.

12.3.3 Absolute Maximum Ratings

The following table, Table 73, specifies the absolute maximum ratings of PowerSpan II.

Table 73: Absolute Maximum Ratings

Symbol	Parameter	Limits	Units
Vdd Core[a b]	Core Supply Voltage	-0.3 to 2.7	V
Vdd I/O[a c]	I/O Supply Voltage	-0.3 to 3.6	V
Px_VDDA[a b]	PLL Supply Voltage	-0.3 to 2.7	V
Vin[a d e]	DC Input Voltage (LVTTL)	-0.3 to Vdd + 0.3	V
Vin[a d e]	DC Input Voltage (5 V tolerant LVTTL)	-0.6 to 5.5	V
Tstg	Storage Temperature	-65 to 150	°C

[a] Functional operation at the maximums is not guaranteed. Stress beyond those listed can affect device reliability or cause permanent damage to PowerSpan II.

[b] Vdd Core/ Px_VDDA must not exceed Vdd I/O by more than 0.4 V. This includes during power-on reset.

[c] Vdd I/O must not exceed Vdd Core/ Px_VDDA by more than 1.6 V. This includes during power-on reset.

[d] These limits only apply to overshoot and undershoot. Cell functionality is not implied.

[e] Vin must not exceed Vdd I/O by more than 2.5 V at any time. This includes during power-on reset.

13. Register Descriptions

This chapter describes the registers used in PowerSpan II. It describes the register settings and bits which enable PowerSpan II features and functionality. The following topics are discussed:

- "Register Access" on page 283
- "Register Reset" on page 295
- "Configuration and IACK Cycle Generation" on page 295
- "Register Descriptions" on page 298

13.1 Register Access

The PowerSpan II registers can be accessed from both PCI and the processor bus. PowerSpan II allows reads to its registers from all of its bus interfaces at the same time. However, writes may occur from only one bus interface at a time.

13.1.1 Register Map

The 4 Kbytes of PowerSpan II Control and Status Registers (PCSR) are used for PCI Control and Status Registers (CSRs), and for overall PowerSpan II operation. The PCSR space is functionally divided into two areas: the PCI CSR space and the PowerSpan II PCSR space. PSCR space is accessible from the Processor Bus, PCI-1 or PCI-2 interfaces.

Table 74 is a detailed memory map for PCSR space and shows the PowerSpan II register map for the Dual PCI PowerSpan II. PowerSpan II is available as both the Single PCI PowerSpan II and Dual PCI PowerSpan II.

All registers that are based on standards — for example, required PCI registers — are in italics. The shaded registers under PCI-1 Configuration and PCI-2 Configuration registers exist only if the associated PCI Interface is configured as the Primary Interface. A interface is configured as Primary using a power-up option (see "Resets, Clocks and Power-up Options" on page 201 for more information). The PCI Interface that is designated as Primary has added functionality which includes CompactPCI Hot Swap support, Vital Product Data support and an I^2C Interface. Refer to"PCI Interface" on page 43 for more information on Primary Interface functionality.

Table 74: PowerSpan II Register Map

Offset (HEX)	Register Mnemonic	Register Name	Page
PCI-1 *Configuration* Registers			
000	P1_ID	*PCI-1 ID Register*	page 300
004	P1_CSR	*PCI-1 Control and Status Register*	page 301
008	P1_CLASS	*PCI-1 Class Register*	page 305
00C	P1_MISC0	*PCI-1 Miscellaneous 0 Register*	page 307
010	P1_BSI2O	*PCI-1 I_2O Target Image Base Address Register*	page 309
014	P1_BSREG	*PCI-1 Register Image Base Address Register*	page 311
018	P1_BST0	*PCI-1 Target Image 0 Base Address Register*	page 312
01C	P1_BST1	*PCI-1 Target Image 1 Base Address Register*	page 312
020	P1_BST2	*PCI-1 Target Image 2 Base Address Register*	page 312
024	P1_BST3	*PCI-1 Target Image 3 Base Address Register*	page 312
028	PCI Unimplemented		
02C	P1_SID	*PCI-1 Subsystem ID Register*	page 313
030	PCI Unimplemented		
034	P1_CAP	*PCI-1 Capability Pointer Register*	page 314
038	PCI Unimplemented		
03C	P1_MISC1	*PCI-1 Miscellaneous 1 Register*	page 315
040-0E0	PCI Unimplemented		
0E4	P1_HS_CSR	*PCI-1 Compact PCI Hot Swap Control and Status Register*	page 317

Table 74: PowerSpan II Register Map

Offset (HEX)	Register Mnemonic	Register Name	Page
0E8	P1_VPDC	*PCI-1 Vital Product Data Capability Register*	page 319
0EC	P1_VPDD	*PCI-1 Vital Product Data Register*	page 321
0F0-0FC	PCI Unimplemented		
PCI-1 Registers			
100	P1_TI0_CTL	PCI-1 Target Image 0 Control Register	page 322
104	P1_TI0_TADDR	PCI-1 Target Image 0 Translation Address Register	page 328
108-10C	PowerSpan II Reserved		
110	P1_TI1_CTL	PCI-1 Target Image 1 Control Register	page 322
114	P1_TI1_TADDR	PCI-1 Target Image 1 Translation Address Register	page 328
118-11C	PowerSpan II Reserved		
120	P1_TI2_CTL	PCI-1 Target Image 2 Control Register	page 322
124	P1_TI2_TADDR	PCI-1 Target Image 2 Translation Address Register	page 328
128-12C	PowerSpan II Reserved		
130	P1_TI3_CTL	PCI-1 Target Image 3 Control Register	page 322
134	P1_TI3_TADDR	PCI-1 Target Image 3 Translation Address Register	page 328
138-140	PowerSpan II Reserved		
144	P1_CONF_INFO	PCI-1 to PCI-2 Configuration Cycle Information Register	page 330
148	P1_CONF_DATA	PCI-1 to PCI-2 Configuration Cycle Data Register	page 333
14C	P1_IACK	PCI-1 to PCI-2 Interrupt Acknowledge Cycle Generation Register	page 334
150	P1_ERRCS	PCI-1 Bus Error Control and Status Register	page 335
154	P1_AERR	PCI-1 Address Error Log Register	page 336
158-15C	PowerSpan II Reserved		
160	P1_MISC_CSR	PCI-1 Miscellaneous Control and Status Register	page 337
164	P1_ARB_CTRL	PCI-1 Bus Arbiter Control Register	page 339
168-1FC	PowerSpan II Reserved		

Table 74: PowerSpan II Register Map

Offset (HEX)	Register Mnemonic	Register Name	Page
Processor Bus Registers			
200	PB_SI0_CTL	Processor Bus Slave Image 0 Control Register	page 342
204	PB_SI0_TADDR	Processor Bus Slave Image 0 Translation Address Register	page 348
208	PB_SI0_BADDR	Processor Bus Slave Image 0 Base Address Register	page 350
20C	PowerSpan II Reserved		
210	PB_SI1_CTL	Processor Bus Slave Image 1 Control Register	page 342
214	PB_SI1_TADDR	Processor Bus Slave Image 1 Translation Address Register	page 348
218	PB_SI1_BADDR	Processor Bus Slave Image 1 Base Address Register	page 350
21C	PowerSpan II Reserved		
220	PB_SI2_CTL	Processor Bus Slave Image 2 Control Register	page 342
224	PB_SI2_TADDR	Processor Bus Slave Image 2 Translation Address Register	page 348
228	PB_SI2_BADDR	Processor Bus Slave Image 2 Base Address Register	page 350
22C	PowerSpan II Reserved		
230	PB_SI3_CTL	Processor Bus Slave Image 3 Control Register	page 342
234	PB_SI3_TADDR	Processor Bus Slave Image 3 Translation Address Register	page 348
238	PB_SI3_BADDR	Processor Bus Slave Image 3 Base Address Register	page 350
23C	PowerSpan II Reserved		
240	PB_SI4_CTL	Processor Bus Slave Image 4 Control Register	page 342
244	PB_SI4_TADDR	Processor Bus Slave Image 4 Translation Address Register	page 348
248	PB_SI4_BADDR	Processor Bus Slave Image 4 Base Address Register	page 350
24C	PowerSpan II Reserved		
250	PB_SI5_CTL	Processor Bus Slave Image 5 Control Register	page 342

Table 74: PowerSpan II Register Map

Offset (HEX)	Register Mnemonic	Register Name	Page
254	PB_SI5_TADDR	Processor Bus Slave Image 5 Translation Address Register	page 348
258	PB_SI5_BADDR	Processor Bus Slave Image 5 Base Address Register	page 350
25C	PowerSpan II Reserved		
260	PB_SI6_CTL	Processor Bus Slave Image 6 Control Register	page 342
264	PB_SI6_TADDR	Processor Bus Slave Image 6 Translation Address Register	page 348
268	PB_SI6_BADDR	Processor Bus Slave Image 6 Base Address Register	page 350
26C	PowerSpan II Reserved		
270	PB_SI7_CTL	Processor Bus Slave Image 7 Control Register	page 342
274	PB_SI7_TADDR	Processor Bus Slave Image 7 Translation Address Register	page 348
278	PB_SI7_BADDR	Processor Bus Slave Image 7 Base Address Register	page 350
27C	PowerSpan II Reserved		
280	PB_REG_BADDR	Processor Bus Register Image Base Address Register	page 351
284-28C	PowerSpan II Reserved		
290	PB_CONF_INFO	Processor Bus PCI Configuration Cycle Information Register	page 352
294	PB_CONF_DATA	Processor Bus PCI Configuration Cycle Data Register	page 355
298-29C	PowerSpan II Reserved		
2A0	PB_P1_IACK	Processor Bus to PCI-1 Interrupt Acknowledge Cycle Generation Register	page 356
2A4	PB_P2_IACK	Processor Bus to PCI-2 Interrupt Acknowledge Cycle Generation Register	page 357
2A8-2AC	PowerSpan II Reserved		
2B0	PB_ERRCS	Processor Bus Error Control and Status Register	page 359
2B4	PB_AERR	Processor Bus Address Error Log Register	page 360

Table 74: PowerSpan II Register Map

Offset (HEX)	Register Mnemonic	Register Name	Page
2B8- 2BC	PowerSpan II Reserved		
2C0	PB_MISC_CSR	Processor Bus Miscellaneous Control and Status Register	page 361
2C4- 2CC	PowerSpan II Reserved		
2D0	PB_ARB_CTRL	Processor Bus Arbiter Control Register	page 364
2D4- 2FC	PowerSpan II Reserved		
DMA Registers			
300	PowerSpan II Reserved		
304	DMA0_SRC_ADDR	DMA 0 Source Address Register	page 367
308	PowerSpan II Reserved		
30C	DMA0_DST_ADDR	DMA 0 Destination Address Register	page 368
310	PowerSpan II Reserved		
314	DMA0_TCR	DMA 0 Transfer Control Register	page 369
318	PowerSpan II Reserved		
31C	DMA0_CPP	DMA 0 Command Packet Pointer Register	page 371
320	DMA0_GCSR	DMA 0 General Control Register	page 372
324	DMA0_ATTR	DMA 0 Attributes Register	page 376
328-330	PowerSpan II Reserved		
334	DMA1_SRC_ADDR	DMA 1 Source Address Register	page 367
338	PowerSpan II Reserved		
33C	DMA1_DST_ADDR	DMA 1 Destination Address Register	page 368
340	PowerSpan II Reserved		
344	DMA1_TCR	DMA 1 Transfer Control Register	page 369
348	PowerSpan II Reserved		
34C	DMA1_CPP	DMA 1 Command Packet Pointer Register	page 371

Table 74: PowerSpan II Register Map

Offset (HEX)	Register Mnemonic	Register Name	Page
350	DMA1_GCSR	DMA 1 General Control and Status Register	page 372
354	DMA1_ATTR	DMA 1 Attributes Register	page 376
358-360	PowerSpan II Reserved		
364	DMA2_SRC_ADDR	DMA 2 Source Address Register	page 367
368	PowerSpan II Reserved		
36C	DMA2_DST_ADDR	DMA 2 Destination Address Register	page 368
370	PowerSpan II Reserved		
374	DMA2_TCR	DMA 2 Transfer Control Register	page 369
378	PowerSpan II Reserved		
37C	DMA2_CPP	DMA 2 Command Packet Pointer Register	page 371
380	DMA2_GCSR	DMA 2 General Control and Status Register	page 372
384	DMA2_ATTR	DMA 2 Attributes Register	page 376
388-390	PowerSpan II Reserved		
394	DMA3_SRC_ADDR	DMA 3 Source Address Register	page 367
398	PowerSpan II Reserved		
39C	DMA3_DST_ADDR	DMA 3 Destination Address Register	page 368
3A0	PowerSpan II Reserved		
3A4	DMA3_TCR	DMA 3 Transfer Control Register	page 369
3A8	PowerSpan II Reserved		
3AC	DMA3_CPP	DMA 3 Command Packet Pointer Register	page 371
3B0	DMA3_GCSR	DMA 3 General Control and Status Register	page 372
3B4	DMA3_ATTR	DMA 3 Attributes Register	page 376
3B8-3FC	PowerSpan II Reserved		
Miscellaneous Registers			
400	MISC_CSR	Miscellaneous Control and Status Register	page 378
404	CLOCK_CTL	Clock Control Register	page 381

Table 74: PowerSpan II Register Map

Offset (HEX)	Register Mnemonic	Register Name	Page
408	I2C_CSR	I^{2}C Interface Control and Status Register	page 383
40C	RST_CSR	Reset Control and Status Register	page 385
410	ISR0	Interrupt Status Register 0	page 388
414	ISR1	Interrupt Status Register 1	page 390
418	IER0	Interrupt Enable Register 0	page 393
41C	IER1	Interrupt Enable Register 1	page 395
420	IMR_MBOX	Interrupt Map Register: Mailbox	page 398
424	IMR_DB	Interrupt Map Register: Doorbell	page 400
428	IMR_DMA	Interrupt Map Register: DMA	page 401
42C	IMR_HW	Interrupt Map Register: Hardware	page 402
430	IMR_P1	Interrupt Map Register: PCI-1	page 404
434	IMR_P2	Interrupt Map Register: PCI-2	page 405
438	IMR_PB	Interrupt Map Register: Processor Bus	page 406
43C	IMR2_PB	Interrupt Map Register Two: Processor Bus	page 408
440	IMR_MISC	Interrupt Map Register: Miscellaneous	page 409
444	IDR	Interrupt Direction Register	page 410
448-44C	PowerSpan II Reserved		
450	MBOX0	Mailbox 0 Register	page 412
454	MBOX1	Mailbox 1 Register	page 412
458	MBOX2	Mailbox 2 Register	page 412
45C	MBOX3	Mailbox 3 Register	page 412
460	MBOX4	Mailbox 4 Register	page 412
464	MBOX5	Mailbox 5 Register	page 412
468	MBOX6	Mailbox 6 Register	page 412
46C	MBOX7	Mailbox 7 Register	page 412
470	SEMA0	Semaphore 0 Register	page 413

Table 74: PowerSpan II Register Map

Offset (HEX)	Register Mnemonic	Register Name	Page
474	SEMA1	Semaphore 1 Register	page 414
478-4FC	PowerSpan II Reserved		
I_2O Registers			
500	PCI_TI2O_CTL	PCI I_2O Target Image Control Register	page 415
504	PCI_TI2O_TADDR	PCI I_2O Target Image Translation Address Register	page 420
508	I2O_CSR	I_2O Control and Status Register	page 421
50C	I2O_QUEUE_BS	I_2O Queue Base Address Register	page 423
510	IFL_BOT	I_2O Inbound Free List Bottom Pointer Register	page 425
514	IFL_TOP	I_2O Inbound Free List Top Pointer Register	page 426
518	IFL_TOP_INC	I_2O Inbound Free List Top Pointer Increment Register	page 427
51C	IPL_BOT	I_2O Inbound Post List Bottom Pointer Register	page 428
520	IPL_BOT_INC	I_2O Inbound Post List Bottom Pointer Increment Register	page 429
524	IPL_TOP	I_2O Inbound Post List Top Pointer Register	page 430
528	OFL_BOT	I_2O Outbound Free List Bottom Pointer Register	page 431
52C	OFL_BOT_INC	I_2O Outbound Free List Bottom Pointer Increment Register	page 432
530	OFL_TOP	I_2O Outbound Free List Top Pointer Register	page 433
534	OPL_BOT	I_2O Outbound Post List Bottom Pointer Register	page 434
538	OPL_TOP	I_2O Outbound Post List Top Pointer Register	page 435
53C	OPL_TOP_INC	I_2O Outbound Post List Top Pointer Increment Register	page 436
540	HOST_OIO	I_2O Host Outbound Index Offset Register	page 437
544	HOST_OIA	I_2O Host Outbound Index Alias Register	page 438
548	IOP_OI	I_2O IOP Outbound Index Register	page 439
54C	IOP_OI_INC	I_2O IOP Outbound Index Increment Register	page 440
550-7FC	PowerSpan II Reserved		

Table 74: PowerSpan II Register Map

Offset (HEX)	Register Mnemonic	Register Name	Page
PCI-2 *Configuration* Registers (Dual PCI PowerSpan II)			
800	P2_ID	*PCI-2 ID Register*	page 441
804	P2_CSR	*PCI-2 Control and Status Register*	page 442
808	P2_CLASS	*PCI-2 Class Register*	refer to PCI-1
80C	P2_MISC0	*PCI-2 Miscellaneous 0 Register*	refer to PCI-1
810	P2_BSI2O	*PCI-2 I_2O Target Image Base Address Register*	refer to PCI-1
814	P2_BSREG	*PCI-2 Register Image Base Address Register*	refer to PCI-1
818	P2_BST0	*PCI-2 Target Image 0 Base Address Register*	refer to PCI-1
81C	P2_BST1	*PCI-2 Target Image 1 Base Address Register*	refer to PCI-1
820	P2_BST2	*PCI-2 Target Image 2 Base Address Register*	refer to PCI-1
824	P2_BST3	*PCI-2 Target Image 3 Base Address Register*	refer to PCI-1
828	PCI Unimplemented		
82C	P2_SID	*PCI-2 Subsystem ID Register*	refer to PCI-1
830	PCI Unimplemented		
834	P2_CAP	*PCI-2 Capability Pointer Register*	refer to PCI-1
838	PCI Unimplemented		
83C	P2_MISC1	*PCI-2 Miscellaneous 1 Register*	refer to PCI-1
840-8E0	PCI Unimplemented		
8E4	P2_HS_CSR	*PCI-2 Compact PCI Hot Swap Control and Status Register*	refer to PCI-1
8E8	P2_VPDC	*PCI-2 Vital Product Data Capability Register*	refer to PCI-1
8EC	P2_VPDD	*PCI-2 Vital Product Data Register*	refer to PCI-1
8F0-8FC	PCI Unimplemented		
PCI-2 Registers (Dual PCI PowerSpan II)			
900	P2_TI0_CTL	PCI-2 Target Image 0 Control Register	refer to PCI-1
904	P2_TI0_TADDR	PCI-2 Target Image 0 Translation Address Register	refer to PCI-1

Table 74: PowerSpan II Register Map

Offset (HEX)	Register Mnemonic	Register Name	Page
908-90C	PowerSpan II Reserved		
910	P2_TI1_CTL	PCI-2 Target Image 1 Control Register	refer to PCI-1
914	P2_TI1_TADDR	PCI-2 Target Image 1 Translation Address Register	refer to PCI-1
918-91C	PowerSpan II Reserved		
920	P2_TI2_CTL	PCI-2 Target Image 2 Control Register	refer to PCI-1
924	P2_TI2_TADDR	PCI-2 Target Image 2 Translation Address Register	refer to PCI-1
928-92C	PowerSpan II Reserved		
930	P2_TI3_CTL	PCI-2 Target Image 3 Control Register	refer to PCI-1
934	P2_TI3_TADDR	PCI-2 Target Image 3 Translation Address Register	refer to PCI-1
938-940	PowerSpan II Reserved		
944	P2_CONF_INFO	PCI-2 to PCI 1 Configuration Cycle Information Register	refer to PCI-1
948	P2_CONF_DATA	PCI-2 to PCI 1 Configuration Cycle Data Register	refer to PCI-1
94C	P2_IACK	PCI-2 to PCI 1 Interrupt Acknowledge Cycle Generation Register	refer to PCI-1
950	P2_ERRCS	PCI-2 Bus Error Control and Status Register	refer to PCI-1
954	P2_AERR	PCI-2 Address Error Log Register	refer to PCI-1
958-95C	PowerSpan II Reserved		
960	P2_MISC_CSR	PCI-2 Miscellaneous Control and Status Register	refer to PCI-1
964	P2_ARB_CTRL	PCI-2 Bus Arbiter Control Register	refer to PCI-1
968-FFC	PowerSpan II Reserved		

13.1.2 Access from PCI

The PCI-1 Register Image Base Address Register specifies the 4-Kbyte aligned base address for the PowerSpan II Control and Status Registers (PCSRs) in PCI Memory Space. The base address for PCSR space is enabled by:

1. Setting the BSREG_BAR_EN bit in the P1_MISC_CS
2. Writing to the P1_BSREG register either with a PCI Configuration write access or by writing to it from the Processor Bus (PB).

Once enabled, the PCSR space can be accessed in PCI Memory Space with single-beat 32-bit accesses.

13.1.3 Access from the Processor Bus

The PB_REG_BADDR register specifies the 4-Kbyte aligned base address for PCSR space on the processor bus. This register is programmed through any register interface or through EEPROM. Register accesses through the Processor Bus Interface can be big-endian or PowerPC little-endian (see “Processor Bus and PowerSpan II Register Transfers” on page 215). The endian conversion for register accesses from the Processor Bus Interface is controlled with the END bit in the PB_REG_BADDR Register. The default mode is big-endian.

The reset state for the base address for PCSR space on the processor bus is 0x3000_0000. Refer to Table 112 for more information.

13.1.4 Access from Multiple Interfaces

PowerSpan II allows reads to its registers from all of its bus interfaces at the same time. However, writes may occur from only one bus interface at a time. This prevents data corruption if two or more interfaces try to write to the same register simultaneously.

Register writes to “write 1 to set/clear” status bits may not be reflected by an immediate register read.

PowerSpan II uses an internal round robin arbitration mechanism for register access from the different bus interfaces. Register writes are retried until the interface doing the write has successfully arbitrated for register access.

Register accesses from all interfaces are retried during EEPROM load.

Each PowerSpan II PCI Target has a Px Lockout (Px_LOCKOUT) bit in the “Miscellaneous Control and Status Register” on page 378 (MISC_CSR). While a lockout bit is set, the corresponding PCI Target retries all Configuration Type 0 transactions. When the Base Address registers have been configured, memory transactions are claimed, but they are retried until the lockout bit is cleared. By default the Px_LOCKOUT bits are set. The lockout bits can either be cleared by EEPROM load, or by an access from the Processor Bus Interface. The lockout bits are automatically cleared by PowerSpan II when the PWRUP_BOOT bit in the “Processor Bus Arbiter Control Register” on page 364 is set to PCI.

13.2 Register Reset

The PCSR space is divided into four reset domains:

- PCI-1 CSR space
- PCI-2 CSR space
- Processor Bus Interface registers
- Device Specific registers

See “Reset Response” on page 203 for a detailed description of register reset partitioning.

When an EEPROM is detected by PowerSpan II’s I^{2}C Interface after reset, certain registers are initialized with the contents of the EEPROM. See “I2C/EEPROM” on page 153 for details on which register fields are loaded through EEPROM.

13.3 Configuration and IACK Cycle Generation

PowerSpan II has registers that must be programmed in order for a PCI master to generate configuration (Type 1 or 0) and IACK transactions on the alternate PCI Interface and for the processor bus to generate configuration (Type 1 or 0) and IACK transactions on either PCI bus.

13.3.1 From PCI-to-PCI

The following PowerSpan II registers are used by a PCI master to generate configuration (Type 1 or 0) and IACK transactions on the alternate PCI Interface:

- Px_CONF_INFO/Px_CONF_DATA
- Px_IACK

13.3.1.1 PCI Configuration Data

Generating a Configuration transaction on PCI requires the programming of the “PCI 1 to PCI-2 Configuration Cycle Information Register” on page 330 (Px_CONF_INFO) in order set-up the address of the Configuration cycle. The PCI transaction is generated when a register access occurs on the“PCI 1 to PCI-2 Configuration Cycle Data Register” on page 333 (Px_CONF_DATA).

When a register write is performed to Px_CONF_DATA, the address and data parameters in Px_CONF_DATA are used to generate a configuration transaction on the alternate PCI bus.

When a register read is performed to Px_CONF_DATA, the read is retried while a configuration read transaction is generated on the alternate PCI bus. The address for the read transaction is defined by Px_CONF_INFO. While the data is being retrieved, register accesses to Px_CONF_DATA is retried.

The Px_CONF_INFO and Px_CONF_DATA registers must be treated as shared resources for applications that require more than one agent to generate configuration transactions on PCI. A semaphore is used to control access.

13.3.1.2 Interrupt Acknowledge Generation

The Px_IACK register is used to generate IACK reads on the alternate PCI bus. If a register read is performed to Px_IACK, then the read is retried while an IACK cycle is generated on the alternate PCI bus. The address for the IACK cycle is taken directly from the originating PCI bus.

The Px_IACK register must be treated as shared resources for applications that require more than one agent to generate IACK transactions on PCI. A semaphore is used to control access.

Writes to Px_IACK have no effect.

13.3.2 From the Processor Bus to PCI

The following PowerSpan II registers are used to generate Configuration (Type 1 or 0) and IACK transactions going from the Processor Bus Interface to either of the PCI Interfaces:

- PB_CONF_INFO/PB_CONF_DATA
- PB_Px_IACK

13.3.2.1 Processor Bus Configuration Data

Generating a Configuration transaction on PCI requires the programming of the "Processor Bus PCI Configuration Cycle Information Register" on page 352 (PB_CONF_INFO) to set-up the address of the Configuration cycle. The DEST bit selects the PCI bus for the configuration access. The PCI transaction is generated when the user performs a register access to the "Processor Bus Configuration Cycle Data Register" on page 355 (PB_CONF_DATA).

When a register write is performed to PB_CONF_DATA, the address and data parameters in PB_CONF_DATA are used to generate a Configuration transaction on the selected PCI bus.

The Processor Bus Slave response to the read of PB_CONF_DATA is dependent on the state the Address Retry Enable (ARTRY_EN) bit of the "Processor Bus Miscellaneous Control and Status Register" on page 361. If ARTRY_EN is disabled, the Processor Bus slave claims the read of PB_CONF_DATA. The Processor Bus slave only asserts PB_TA_ to complete the transaction when the read data is returned from PCI. If ARTRY_EN is enabled, the read of PB_CONF_DATA is retried immediately. Subsequent register accesses to PB_CONF_DATA will be retried until the data is returned from PCI.

13.3.2.2 Interrupt Acknowledge Generation

The PB_Px_IACK registers are used to generate IACK reads on the PCI interfaces.The address for the IACK cycle is taken directly from the processor bus.

The Processor Bus slave response to the read of PB_Px_IACK is dependent on the state of the Address Retry Enable (ARTRY_EN) bit of the "Processor Bus Miscellaneous Control and Status Register" on page 361. If ARTRY_EN is disabled, the Processor Bus slave claims the read of PB_Px_IACK. The Processor Bus slave only asserts PB_TA_ to complete the transaction when the read data is returned from PCI. If ARTRY_EN is enabled, the read of PB_Px_IACK is retried immediately. Subsequent register accesses to PB_Px_IACK are retried until the data is returned from PCI.

Applications that require more than one agent to generate IACK transactions on PCI must use a semaphore to control PB_P1_IACK and PB_P2_IACK.

Writes to PB_Px_IACK have no effect.

13.4 Bit Ordering and Endian Ordering

The register tables in "Register Descriptions" on page 298 provide bit ordering in both PCI little-endian and PowerPC big-endian. The register table from the "PCI 1 Control and Status Register" on page 301 is repeated here. The "PCI Bits" in the left hand column give the addressing of the register bits when the register is accessed from the PCI bus in little-endian mode. The "PB Bits" in the far right hand column give the addressing of the register bits when the register is accessed from the Processor Bus in big-endian mode.

Please consult “Endian Mapping” on page 213 for a full endian discussion.

PCI Bits	Function								PB Bits
31-24	D_PE	S_SERR	R_MA	R_TA	S_TA	DEVSEL		MDP_D	0-7
23-16	TFBBC	0	DEV66	CAP_L	PCI Reserved				8-15
15-08	PCI Reserved						MFBBC	SERR_EN	16-23
07-00	WAIT	PERESP	VGAPS	MWI_EN	SC	BM	MS	IOS	24-31

Throughout the manual, register fields are given assuming PCI little-endian bit ordering. The user must consult the register table to obtain the corresponding PowerPC big-endian bit ordering.

13.5 Register Descriptions

In the following detailed descriptions of each register, the shaded register bits are different for the Dual PCI PowerSpan II and Single PCI PowerSpan II

Table 75 describes the abbreviations used in the register descriptions.

Table 75: Abbreviations used in Register Descriptions

Abbreviation	Description
G_RST	General Reset (Active when either PB_RST, P1_RST, or P2_RST is asserted)
PB_RST	Processor Bus Reset
P1_RST	PCI-1 (P1) Reset
P2_RST	PCI-2 (P2) Reset
Px_RST	PCI-1 or PCI-2 Reset
PRI_RST	Primary PCI Reset
R/W	Read/Write
R	Read Only
R/Write 1 to Clear	Read/Write 1 to Clear

Table 75: Abbreviations used in Register Descriptions

Abbreviation	Description
Write 1 to Set	Read 0/Write 1 to Set (Writing a 1 triggers an event)
R/WPB	Read Only from PCI, Read/Write from Processor Bus
0 EEPROM	Reset value is 0. Register bit may be loaded by EEPROM after reset
1 EEPROM	Reset value is 1. Register bit may be loaded by EEPROM after reset
PWRUP	Register bit loaded as a power-up option
PCI Reserved	Do not write. Read back 0
PCI Unimplemented	Do not write. Read back 0
PowerSpan II Reserved	Do not write. Read back undefined
Reserved	Do not write. Read back undefined
Single PCI PowerSpan II	Single PCI PowerSpan II (PCI-1 and Processor Bus)

13.5.1 PCI-1 ID Register

This register is Read Only from PCI-1 and Read/Write from the Processor Bus Interface.

Table 76: PCI 1 ID Register

Register Name: P1_ID	Register Offset: 0x000

PCI Bits	Function	PB Bits
31-24	DID	0-7
23-16	DID	8-15
15-08	VID	16-23
07-00	VID	24-31

P1_ID Description

Name	Type	Reset By	Reset State	Function
DID[15:0]	R/Write from processor bus	P1_RST	0x8260 EEPROM	Device ID Tundra allocated Device Identifier
DID[15:0]	R/Write from processor bus	P1_RST	0x8261	Single PCI PowerSpan II
VID[15:0]	R/Write from processor bus	P1_RST	0x10E3 EEPROM	Vendor ID PCI SIG allocated Vendor Identifier

13.5.2 PCI-1 Control and Status Register.

Table 77: PCI 1 Control and Status Register

Register Name: P1_CSR	**Register Offset: 0x004**

PCI Bits	Function								PB Bits
31-24	D_PE	S_SERR	R_MA	R_TA	S_TA	DEVSEL		MDP_D	0-7
23-16	TFBBC	0	DEV66	CAP_L	PCI Reserved				8-15
15-08	PCI Reserved						MFBBC	SERR_EN	16-23
07-00	WAIT	PERESP	VGAPS	MWI_EN	SC	BM	MS	IOS	24-31

P1_CSR Description

Name	Type	Reset By	Reset State	Function
D_PE	R/W 1 to clear	P1_RST	0	Detected Parity Error 0 = No Parity Error, 1 = Parity Error This bit is set by the device whenever the Master Module detects a data parity error or the Target Module detects a data or address parity error.
S_SERR	R/W 1 to clear	P1_RST	0	Signaled SERR# 0 = SERR# not asserted, 1 = SERR# asserted The device as PCI Target sets this bit when it asserts SERR# to signal an address parity error. SERR_EN and PERESP must be set before SERR# can be asserted.
R_MA	R/W 1 to clear	P1_RST	0	Received Master Abort 0 = device did not generate Master-Abort 1 = device generated Master Abort The device sets this bit when a transaction it initiated had to be terminated with a Master-Abort.

P1_CSR Description (Continued)

Name	Type	Reset By	Reset State	Function
R_TA	R/W 1 to clear	P1_RST	0	Received Target Abort 0 = device did not detect Target-Abort 1 = device detected Target-Abort The device sets this bit when a transaction it initiated was terminated with a Target-Abort.
S_TA	R/W 1 to clear	P1_RST	0	Signaled Target-Abort 0 = device Target Module did not terminate transaction with Target-Abort 1 = device Target Module terminated transaction with Target-Abort.
DEVSEL [1:0]	R	P1_RST	01	Device Select Timing The device is a medium speed device.
MDP_D	R/W 1 to clear	P1_RST	0	Master Data Parity Detected 0 = Master Module did not detect/generate data parity error 1 = Master Module detected/generated data parity error The device sets this bit if the PERESP bit is set and either (a) it is the master of the transaction in which it asserts PERR#, or (b) the addressed target asserts PERR#
TFBBC	R	P1_RST	0	Target Fast Back to Back Capable The device cannot accept fast back to back transactions
DEV66	R	P1_RST	1	Device 66 MHz The device is a 66 MHz capable device
CAP_L	R	P1_RST	PWRUP	Capabilities List 0 = capabilities list unsupported 1 = capabilities list supported
MFBBC	R	P1_RST	0	Master Fast Back to Back Enable The device does not generate fast back to back transactions

P1_CSR Description (Continued)

Name	Type	Reset By	Reset State	Function
SERR_EN	R/W	P1_RST	0	SERR# Enable 0 = Disable SERR# driver 1 = Enable SERR# driver Setting this and PERESP allows the device to report address parity errors with SERR# as PCI target.
WAIT	R	P1_RST	0	Wait Cycle Control 0 = No address/data stepping
PERESP	R/W	P1_RST	0	Parity Error Response 0 = Disable 1 = Enable Controls the device's response to address and data parity errors. When enabled, PERR# is asserted and the MDP_D bit is set in response to data parity errors. When this bit and SERR_EN are set, the device reports address parity errors on SERR#. This bit does not affect the device's parity generation.
VGAPS	R	P1_RST	0	VGA Palette Snoop 0 = Disable
MWI_EN	R	P1_RST	0	Memory Write and Invalidate Enable 0 = Disable PowerSpan II does not generate MWI transactions.
SC	R	P1_RST	0	Special Cycles 0 = Disable The device does not respond to Special cycles as a PCI target.
BM	R/W	P1_RST	0 EEPROM	Bus Master Enables the device to generate cycles as a PCI Master. 0 = Disable, 1 = Enable

P1_CSR Description (Continued)

Name	Type	Reset By	Reset State	Function
MS	R/W	P1_RST	0 EEPROM	Memory Space 0 = Disable, 1 = Enable Enables the device to accept Memory cycles as a PCI target.
IOS	R	P1_RST	0	IO Space 0 = Disable The device does not respond to I/O cycles as a PCI target.

CAP_L: The capabilities list is only supported by the Primary PCI Interface. When PCI-1 is the Primary Interface, CAP_L in PCI-1 is set and CAP_L in PCI-2 is cleared. The opposite is true when PCI-2 is the Primary Interface. The Primary PCI Interface is determined by the PWRUP_PRI_PCI power-up option.

13.5.3 PCI-1 Class Register

Table 78: PCI 1 Class Register

Register Name: P1_CLASS	**Register Offset: 0x008**

PCI Bits	**Function**	**PB Bits**
31-24	BASE	0-7
23-16	SUB	8-15
15-08	PROG	16-23
07-00	RID	24-31

P1_CLASS Description

Name	**Type**	**Reset By**	**Reset State**	**Function**
BASE[7:0]	R/WPB	P1_RST	0x06 EEPROM	Base Class Code 0x06 = Bridge Device (default) 0x0E = I2O controller
SUB[7:0]	R/WPB	P1_RST	0x80 EEPROM	Sub Class Code 0x80 = Other bridge device (if BASE = 0x06) 0x00 = I2O Device (if BASE = 0x0E)
PROG[7:0]	R/WPB	P1_RST	0x00 EEPROM	Programming Interface 0x00 = Other Bridge Device (if BASE = 06) 0x01 = I2O Inbound and Outbound Queues mapped to offsets 0x40 and 0x44 respectively, and I2O Interrupt Status and Mask registers at offsets 0x30 and 0x34 (if BASE = 0x0E)
RID[7:0]	R/WPB	P1_RST	0x01 EEPROM	Revision ID PowerSpan II - 0x01

BASE[7:0]: When PowerSpan II is an I_2O controller, this field must be programmed with 0x0E either from the Processor Bus or by EEPROM.

SUB[7:0]: When PowerSpan II is an I_2O controller, this field must be programmed with 0x00 either from the Processor Bus or by EEPROM.

PROG[7:0]: When PowerSpan II is an I_2O controller, this field must be programmed with 0x01 either from the Processor Bus or by EEPROM.

13.5.4 PCI-1 Miscellaneous 0 Register

Table 79: PCI 1 Miscellaneous 0 Register

Register Name: P1_MISC0	**Register Offset: 0x00C**

<table>
<tr><th>PCI
Bits</th><th colspan="4">Function</th><th>PB
Bits</th></tr>
<tr><td>31-24</td><td>BISTC</td><td>SBIST</td><td>PCI Reserved</td><td>CCODE</td><td>0-7</td></tr>
<tr><td>23-16</td><td>MFUNCT</td><td colspan="3">LAYOUT</td><td>8-15</td></tr>
<tr><td>15-08</td><td colspan="4">LTIMER</td><td>16-23</td></tr>
<tr><td>07-00</td><td colspan="4">CLINE</td><td>24-31</td></tr>
</table>

P1_MISC0 Description

Name	Type	Reset By	Reset State	Function
BIST	R	P1_RST	0	BIST Capable 0 = device is not BIST capable
SBIST	R	P1_RST	0	Start BIST 0 = device is not BIST capable
CCODE [3:0]	R	P1_RST	0	Completion Code 0 = device is not BIST capable
MFUNCT	R	P1_RST	0	Multifunction Device 0 = device is not a multifunction device
LAYOUT [6:0]	R	P1_RST	0	Configuration Space Layout

P1_MISC0 Description

Name	Type	Reset By	Reset State	Function
LTIMER [7:0]	R/W	P1_RST	0	Latency Timer Number of PCI bus clocks before the device must initiate termination of transaction as a master. Resolution of one clock.
CLINE[7:0]	R/W	P1_RST	0	Cacheline Size Specifies the cacheline size for this interface, in number of 32-bit words. Valid settings are 4, 8, 16 or 32 words. Default setting is 8 words. All other settings default to 8 words. 0x00 = 8 x 32-bit words 0x04 = 4 x 32 bit words 0x08 = 8 x 32-bit words 0x10 = 16 x 32-bit words 0x20 = 32 x 32 bit words others = 8 x 32-bit words

LTIMER[7:0]: This field specifies the value of the Latency Timer for the PCI-1 Master in units of PCI bus clocks. The latency timer provides a resolution of one PCI bus clock. This timer always has a minimum value of eight PCI bus clocks. The values 000b-111b correspond to eight clock cycles.

CLINE[7:0] The CLINE Size specifies the system cacheline size in units of 32-bit words. The CLINE is used by the PowerSpan II PCI Master in determining which PCI Read cycle it generates on PCI (MR, MRL, MRM). Table 80 shows the relationship between the read amount and the read command.

Table 80: Read Amount Versus Read Command

Read Amount	Read Command
< 8 bytes	*Memory* Read
<= CLINE	*Memory* Read Line
> CLINE	*Memory* Read Multiple

13.5.5 PCI-1 I2O Target Image Base Address Register

This register specifies the 64 KByte aligned base address of the device’s PCI I_2O Target Image in PCI Memory space. PowerSpan II only supports the I_2O Target Image on the Primary PCI Interface.

Table 81: PCI 1 I2O Target Image Base Address Register

Register Name: P1_BSI2O	**Register Offset: 0x010**

PCI Bits	**Function**								**PB Bits**
31-24	BA								0-7
23-16	BA								8-15
15-08	0	0	0	0	0	0	0	0	16-23
07-00	0	0	0	0	PRFTCH	TYPE		SPACE	24-31

P1_BSI2O Description

Name	**Type**	**Reset By**	**Reset State**	**Function**
BA[15:0]	R/W	P1_RST	0	Base Address
PRFTCH	R/WPB	P1_RST	1 EEPROM	Prefetchable Memory is prefetchable
TYPE [1:0]	R	P1_RST	0	Type 00 = locate anywhere in 32-bit address space
SPACE	R	P1_RST	0	PCI Bus Address Space 0 = Memory

The first 4 Kbytes of this image provides the I_2O Shell Interface Inbound and Outbound Queues and the Host Interrupt Status and Mask Registers. Cycles claimed by the PowerSpan II I_2O Target Image with offsets greater than 4 Kbytes is passed on to the Processor Bus. The control information for the PowerSpan II I_2O Target Image is fully defined in the PCI_TI2O_CTL and PCI_TI2O_TADDR registers.

A write must occur to this register before the device’s I_2O Target Image is accessed through PCI Memory transactions. This write can be performed with either a PCI configuration transaction or a register access by the local processor.

A Base Address of 0x00000 is not a supported base address and the register image does not respond to PCI transactions as a target device when 0x00000 is written to this field — the image is disabled. PowerSpan II supports a Base Address of 0x00000 if the BAR_EQ_0 bit is set in the MISC_CSR register (see page 378).

The BS field in the PCI_TI2O_CTL register determines the size of the image requested in PCI Memory space for the PCI I_2O Target Image.

Writes are enabled to this register only if the BAR_EN bit in the PCI_TI2O_CTL register is set.

This register is not implemented in the Secondary PCI Interface

13.5.6 PCI-1 Register Image Base Address Register

This register specifies the 4-KByte aligned base address of the device’s register space in PCI Memory Space. The Register space is only 4 KByte, therefore the PCI address lines [11:0] are used to select the register.

Table 82: PCI 1 Register Image Base Address Register

Register Name: P1_BSREG	Register Offset: 0x014

PCI Bits	Function								PB Bits
31-24	BA								0-7
23-16	BA								8-15
15-08	BA				0	0	0	0	16-23
07-00	0	0	0	0	PRFTCH	TYPE		SPACE	24-31

P1_BSREG Description

Name	Type	Reset By	Reset State	Function
BA[19:0]	R/W	P1_RST	0	Base Address
PRFTCH	R	P1_RST	0	Prefetchable memory is not prefetchable
TYPE [1:0]	R	P1_RST	0	Type 00 = locate anywhere in 32-bit address space
SPACE	R	P1_RST	0	PCI Bus Address Space 0 = Memory

A write must occur to this register before the device’s registers can be accessed through PCI memory transactions. This write can be performed with a PCI Configuration transaction or a register access by the local processor.

A Base Address of 0x00000 is not a supported base address and the register image does not respond to PCI transactions as a target device when 0x00000 is written to this field — the image is disabled. PowerSpan II supports a Base Address of 0x00000 if the BAR_EQ_0 bit is set in the MISC_CSR register (see page 378).

Writes are enabled to this register only when the BSREG_BAR_EN bit in the P1_MISC_CSR register is set.

Table 83: PCI 1 Target Image X Base Address Register

Register Name: P1_BSTx	**Register Offset: 0x018, 0x01C, 0x020, 0x024**

PCI Bits	**Function**								**PB Bits**
31-24	BA								0-7
23-16	BA								8-15
15-08	0	0	0	0	0	0	0	0	16-23
07-00	0	0	0	0	PRFTC H	TYPE		SPACE	24-31

P1_BSTx Description

Name	**Type**	**Reset By**	**Reset State**	**Function**
BA[15:0]	R/W	P1_RST	0	Base Address
PRFTCH	R/WPB	P1_RST	1 EEPROM	Prefetchable memory is prefetchable
TYPE [1:0]	R	P1_RST	0	Type 00 = locate anywhere in 32-bit address space
SPACE	R	P1_RST	0	PCI Bus Address Space 0 = Memory Space

This register specifies the 64-KByte aligned base address of the device’s PCI Target Image *x* in PCI Memory space.

A write must occur to this register before the device’s PCI Target Image X is accessed through PCI Memory transactions. This write is performed with a PCI Configuration transaction or a register access by the local processor.

A Base Address of 0x00000 is not a supported base address and the register image does not respond to PCI transactions as a target device when 0x00000 is written to this field — the image is disabled. PowerSpan II supports a Base Address of 0x00000 if the BAR_EQ_0 bit is set in the MISC_CSR register (see page 378).

The BS field of the P1_TIx_CTL register will determine the size of the image requested in PCI memory space for PCI Target Image X.

Writes are enabled to this register only when the BAR_EN bit in the P1_TIx_CTL register is set.

Reads from this image are treated as prefetchable. The PRFTCH field is programmable to provide flexibility for the BIOS.

Table 84: PCI 1 Subsystem ID Register

Register Name: P1_SID	Register Offset: 0x02C

PCI Bits	Function	PB Bits
31-24	SID	0-7
23-16	SID	8-15
15-08	SVID	16-23
07-00	SVID	24-31

P1_SID Description

Name	Type	Reset By	Reset State	Function
SID[15:0]	R/WPB	P1_RST	0 EEPROM	Subsystem ID Values for subsystem ID are vendor specific
SVID[15:0]	R/WPB	P1_RST	0 EEPROM	Subsystem Vendor ID Subsystem Vendor IDs are obtained from the PCI SIG and used to identify the vendor of the add-in board or subsystem.

Writes to the PCI_SID register from the processor propagates to its contents. Writes to the P1_SID register from the PCI bus have no effect on its contents.

13.5.7 PCI-1 Capability Pointer Register

Table 85: PCI 1 Capability Pointer Register

Register Name: P1_CAP	Register Offset: 0x034

PCI Bits	Function			PB Bits
31-24	PCI Reserved			0-7
23-16	PCI Reserved			8-15
15-08	PCI Reserved			16-23
07-00	CAP_PTR	0	0	24-31

P1_CAP Description

Name	Type	Reset By	Reset State	Function
CAP_PTR [7:0]	R	P1_RST	0xE4	Capabilities Pointer

The CAP_PTR indicates the register offset in PCI configuration space of the first capabilities pointer in the capabilities linked-list.

This register is not implemented in the Secondary PCI Interface

13.5.8 PCI-1 Miscellaneous 1 Register

Table 86: PCI 1 Miscellaneous 1 Register

Register Name: P1_MISC1	Register Offset: 0x03C

PCI Bits	Function	PB Bits
31-24	MAX_LAT	0-7
23-16	MIN_GNT	8-15
15-08	INT_PIN	16-23
07-00	INT_LINE	24-31

P1_MISC1 Description

Name	Type	Reset By	Reset State	Function
MAX_LAT [7:0]	R/W	P1_RST	0	Maximum Latency No special latency requirements
MIN_GNT [7:0]	R/W	P1_RST	0	Minimum Grant No special requirements
INT_PIN [7:1]	R	P1_RST	0	Interrupt Pin (7 to 1)
INT_PIN [0]	R/WPB	P1_RST	1 EEPROM	Interrupt Pin 0 = The device does not use any PCI interrupts 1 = The device uses $\overline{\text{INTA}}$
INT_LINE [7:0]	R/W	P1_RST	0	Interrupt Line

MAX_LAT[7:0]: This field specifies how often the device needs access to the PCI bus.

MIN_GNT[7:0]: This field indicates how long a master wants to retain bus ownership whenever it initiates a transaction.

INT_PIN [7:1]: This field represents general purpose interrupt pins. Interrupt pins are active low and, when configured as input, are sampled on three successive processor bus clock edges to ensure appropriate setting of a status bit.

Each pin is bidirectional, open drain, active low and level sensitive. The input/ output character of each interrupt pin is controlled through a corresponding bit in the "Interrupt Direction Register" on page 410. Each pin can be configured as either an input or output. All pins are configured as inputs by default.

INT_PIN[0]: This interrupt pin is used to enable PCI interrupts. If this bit is not set, PowerSpan II does not use PCI interrupts. Setting this bit enables a single function PCI device to use INTA#.

INT_LINE[7:0]: This read/write interrupt line field is used to identify which of the system interrupt request lines on the interrupt controller the device's interrupt request pin is routed to.

13.5.9 PCI-1 Compact PCI Hot Swap Control and Status Register

PowerSpan II supports *CompactPCI Hot Swap Specification Revision 2.0* and is a Hot Swap Silicon Device.

This register controls CompactPCI Hot Swap support in PowerSpan II. The Hot Swap functionality is enabled in the Primary PCI Interface of PowerSpan II. In the Single PCI PowerSpan II the lone PCI Interface is enabled as Primary, but in the Dual PCI PowerSpan II only one of the two ports can be enabled as Primary.

Table 87: PCI 1 Compact PCI Hot Swap Control and Status Register

Register Name: P1_HS_CSR	Register Offset: 0x0E4

<table>
<tr><th>PCI
Bits</th><th colspan="7">Function</th><th>PB
Bits</th></tr>
<tr><td>31-24</td><td colspan="7">PCI Reserved</td><td>0-7</td></tr>
<tr><td>23-16</td><td>INS</td><td>EXT</td><td colspan="2">PI</td><td>LOO</td><td>0</td><td>EIM</td><td>0</td><td>8-15</td></tr>
<tr><td>15-08</td><td colspan="7">NXT_PTR</td><td>16-23</td></tr>
<tr><td>07-00</td><td colspan="7">CAP_ID</td><td>24-31</td></tr>
</table>

P1_HS_CSR Description

Name	Type	Reset By	Reset State	Function
INS	R/Write 1 to clear	P1_RST	0	ENUM# Status - Insertion 1 = ENUM# Asserted 0 = ENUM# Negated
EXT	R/Write 1 to clear	P1_RST	0	ENUM_ Status - Extraction 1 = ENUM# Asserted 0 = ENUM# Negated
PI	R	P1_RST	0	Programming Interface 00 = INS, EXT, LOO, EIM supported 01,10,11 = Reserved
LOO	R/W	P1_RST	0	LED ON/OFF 1 = LED On 0 = LED Off

P1_HS_CSR Description

Name	Type	Reset By	Reset State	Function
EIM	R/W	P1_RST	0	ENUM# Signal Mask 1 = Mask Signal 0 = Enable Signal
NXT_PTR [7:0]	R	P1_RST	0xE8 or 0 (See Below)	Next Pointer
CAP_ID [7:0]	R	P1_RST	0x06	Capability ID

PI: Programming interface bit indicates the programming interface supported by the board. PowerSpan II implements a bit value of 0, which means INS, EXT, LOO, EIM are supported. Refer to the *CompactPCI Hot Swap Specification Revision 2.0* for more information

NEXT_PTR: If MISC_CSR[VPD_EN] is set and an external EEPROM is detected, then this field reads back 0xE8. When the VPD_EN bit in the MISC_CSR register is cleared or an external EEPROM is not detected, this field reads back 0.

This register is not implemented in the Secondary PCI Interface.

13.5.10 PCI-1 Vital Product Data Capability Register

PowerSpan II only supports VPD access from the Primary PCI Interface. The Secondary PCI Interface reads zero for VPD accesses. VPD writes have no effect.

Table 88: PCI 1 Vital Product Data Capability Register

Register Name: P1_VPDC	Register Offset: 0E8

PCI Bits	Function		PB Bits
31-24	F	Reserved	0-7
23-16	VPDA		8-15
15-08	NXT_PTR		16-23
07-00	CAP_ID		24-31

P1_VPDC Description

Name	Type	Reset By	Reset State	Function
F	R/W	P1_RST	0	Data Transfer Complete Flag
VPDA [7:0]	R/W	P1_RST	0x00	Vital Product Data Address
NXT_PTR [7:0]	R	P1_RST	0x00	Next Pointer VPD is the last Extended Capabilities Pointer
CAP_ID [7:0]	R	P1_RST	0x03	Capability ID

F: Indicates when the transfer between the VPD Data register and the EEPROM is complete. Software clears the bit to initiate a read and PowerSpan II sets the bit when the read data is available in the VPD Data register. Software sets the bit to initiate a write and PowerSpan II clears the bit to indicate when the data has been transferred.

VPDA: The 8-bit address specifies the VPD address offset for the VPD-Read or VPD-Write to the serial EEPROM. When I^2C chip select 0 is used for the VPD EEPROM the VPD address translates a maximum of 64 bytes and 192 bytes are available for VPD. The first 64 bytes of VPD is VPD-Read Only, and the remaining 128 bytes — 192 bytes if separate 256 byte EEPROM used for VPD — is VPD-Read/Write.

This register is not implemented in the Secondary PCI Interface. It is also disabled when the NXT_PTR bit in the P1_HS_CSR register is 0.

13.5.11 PCI-1 Vital Product Data Register

This register is enabled when the VPD_EN bit in the MISC_CSR is set to 1. If it is disabled the register always reads zero.

PowerSpan II only supports VPD access from the Primary PCI Interface. The Secondary PCI Interface always reads zero for VPD accesses. VPD writes will have no effect.

This register is not implemented in the Secondary PCI Interface. It is also disabled when the NXT_PTR bit in the P1_HS_CSR register is 0

Table 89: PCI 1 Vital Product Data Register

Register Name: P1_VPDD	Register Offset: 0x0EC

PCI Bits	Function	PB Bits
31-24	VPD_DATA	0-7
23-16	VPD_DATA	8-15
15-08	VPD_DATA	16-23
07-00	VPD_DATA	24-31

P1_VPDD Description

Name	Type	Reset By	Reset State	Function
VPD_DATA [31:0]	R/W	P1_RST	0	VPD Data

13.5.12 PCI-1 Target Image x Control Register

This register contains the control information for the PowerSpan II PCI 1 Target Image x. The Image is enabled for decode when both IMG_EN and BAR_EN are set.

The bits in this register are not dynamic. Do not alter these settings while transactions are being processed through PowerSpan II. Refer to "Translation Address Mapping" on page 349 for more information on dynamic address translation.

Table 90: PCI 1 Target Image x Control Register

Register Name: P1_TIx_CTL	**Register Offset: 0x100, 0x110, 0x120, 0x130**

PCI Bits	**Function**								**PB Bits**
31-24	IMG_EN	TA_EN	BAR_EN	MD_EN	BS				0-7
23-16	MODE	DEST	MEM_IO	RTT					8-15
15-08	GBL	CI	0	WTT					16-23
07-00	PR KEEP	END		MRA	0	RD_AMT			24-31

P1_TIx_CTL Description

Name	**Type**	**Reset By**	**Reset State**	**Function**
IMG_EN	R/W	P1_RST	0	Image Enable 0 = Disable 1 = Enable
TA_EN	R/W	P1_RST	0	Translation Address Enable 0 = Disable 1 = Enable
BAR_EN	R/W	P1_RST	1 EEPROM	PCI Base Address Register Enable 0 = Disable 1 = Enable

P1_TIx_CTL Description

Name	Type	Reset By	Reset State	Function
MD_EN	R/W	P1_RST	0	Master Decode Enable 0=Disable 1=Enable
BS[3:0]	R/W	P1_RST	0 EEPROM	Block Size (64 Kbyte * 2^{BS})
MODE	R/W	P1_RST	0	Image Mode 0 = Memory command generation 1 = I/O command generation or 4 byte memory read (see Table 92 on page 326)
DEST	R/W	P1_RST	0	Destination Bus 0 = Processor Bus 1 = PCI-2 Bus
				Single PCI PowerSpan II: Reserved Processor Bus is the only destination.
MEM_IO	R/W	P1_RST	0	MEM_IO mode 0 = Regular IO mode 1 = Enables 4 byte reads on the processor (60x) bus or 1,2,3 or 4 byte memory reads on the PCI bus(es). The bus that the read occurs on is controlled by the DEST bit.
RTT[4:0]	R/W	P1_RST	0b01010	Processor Bus Read Transfer Type (PB_TT[0:4]) 01010 = Read
GBL	R/W	P1_RST	0	Global 0 = Assert PB_GBL_ 1 = Negate PB_GBL_
CI	R/W	P1_RST	0	Cache Inhibit 0 = Assert PB_CI_ 1 = Negate PB_CI_
WTT[4:0]	R/W	P1_RST	0b00010	Processor Bus Write Transfer Type (PB_TT[0:4]) 00010 = Write with flush

P1_TIx_CTL Description

Name	Type	Reset By	Reset State	Function
PRKEEP	R/W	P1_RST	0	Prefetch Read Keep Data 0 = Disable 1 = Enable
END[1:0]	R/W	P1_RST	0b10	Endian Conversion Mode 00 = Little-endian 01 = PowerPC little-endian 10 = Big-endian 11 = True little-endian
MRA	R/W	P1_RST	0	PCI Memory Read Alias to Memory Read Multiple 0 = Disabled 1 = Enabled
RD_AMT[2:0]	R/W	P1_RST	0	Prefetch Size Specifies the number of bytes the device prefetches for PCI Memory Read Multiple transactions claimed by the target image

IMG_EN: The image enable bit is set by the following:

- Non-Zero write to the P1_BSTx register
- Register write to IMG_EN

The Image Enable is cleared by writing 0 to the IMG_EN bit or writing a 0 the P1_BSTx register (see page 312). IMG_EN will always read zero if P1_BSTx is zero.

TA_EN: When set, the Translation Address (P1_TIx_TADDR) replaces the upper bits of the PCI x bus address. The new address is used on the destination bus. Clearing the enable bit results in no address translation.

BAR_EN: When this bit is set, the P1_BSTx register is Read/Write and visible to PCI BIOS Configuration cycles. When this bit is disabled, the P1_BSTx register is not visible in PCI-1 Configuration space and is read zero only.

Writes to P1_BSTx have no effect when this bit is cleared. This effectively disables the PowerSpan II P1_BSTx Image and PowerSpan II does not request PCI Memory space for the image. If the user is clearing this bit, they must also clear P1_BSTx.

MD_EN: Enables master decode when the internal PCI arbiter is in use —when the P1_ARB_EN bit in the RST_CSR register is set. If MD_EN is cleared, only the PCI Address and Command are used for transaction decode. If MD_EN is set, the originating master is included in the transaction decode. A transaction is claimed only if it originates from the master(s) specified in P1_TIx_TADDR.

MRA: When set, the PCI x Target Image X alias a PCI Memory Read cycle to a PCI Memory Read Multiple cycle and prefetches the number of bytes specified in the RD_AMT[2:0] field. When MRA is cleared the Target Image prefetches 8 bytes when a PCI Memory Read command is decoded.

BS: The block size specifies the size of the image, address lines compared and address lines translated.

Table 91: Block Size

BS[3:0]	Block Size	Address Lines Compared/Translated
0000	64k	AD31-AD16
0001	128K	AD31-AD17
0010	256K	AD31-AD18
0011	512K	AD31-AD19
0100	1M	AD31-AD20
0101	2M	AD31-AD21
0110	4M	AD31-AD22
0111	8M	AD31-AD23
1000	16M	AD31-AD24
1001	32M	AD31-AD25
1010	64M	AD31-AD26
1011	128M	AD31-AD27
1100	256M	AD31-AD28
1101	512M	AD31-AD29
1110	1G	AD31-AD30
1111	2G	AD31

MODE: Determines if the image is used to generate Memory or IO commands on PCI. The MODE is only applicable if the destination is the alternate PCI bus.

- Memory Command Mode:

  PCI Memory commands generated on PCI-2.

  Bursting is supported.

  PRKEEP and RD_AMT[2:0] are only applicable in Memory Command Mode.

- IO Command Mode

  Causes PCI IO commands to be generated on PCI-2. When the image is selected to perform IO commands, transactions are limited to 4 bytes or less. A PCI Master initiated cycle attempting to burst to the image in this mode will be terminated with a Target Disconnect (Retry) after every data beat.

The MODE bit and the MEM_IO bit work together to control the size of the transaction (see Table 92)..

Table 92: Setting for MODE and MEM_IO Bits

MODE Setting	MEM_IO setting	Transaction size
0	0	Memory cycle (minimum 8 byte rmemory read)
0	1	Memory cycle (minimum 8 byte memory read)
1	0	I/O Cycle
1	1	Memory cycle (minimum 4 byte memory read)

END: This bit sets the endian conversion mode. This field is only applicable if the destination is the Processor Bus.

DEST: Selects the destination bus for the transaction.

MEM_IO: PowerSpan II supports 4-byte reads. When this bit is set, I/O commands to the corresponding image generates Memory Read commands on the destination PCI bus (Py) with the same byte enables latched from the source bus transaction. If the destination of the transaction is the PB Interface, a minimum 32-bit aligned, 4-byte read is generated on the processor bus.

The MODE bit and the MEM_IO bit work together to control the size of the transaction (see Table 92).

RTT/WTT: Selects the Transfer Type on the Processor Bus. The register bits RTT[4:0]/WTT[4:0] are mapped to pins PB_TT[0:4]

PRKEEP: Used to hold read data fetched beyond the initial PCI read cycle. When set, subsequent read requests to the same image at the next address retrieves the read data directly from the Switching Fabric instead of causing the destination bus to fetch more data. The read data is invalidated when a read with a non-matching address occurs.

RD_AMT[2:0]: The read amount setting determines different values to prefetch from the destination bus.

Table 93: Read Amount

RD_AMT[2:0]	Data Fetched
000	8 bytes
001	16 bytes
010	32 bytes
011	64 bytes
100	128 bytes
101-111	Reserved

13.5.13 PCI-1 Target Image x Translation Address Register

Table 94: PCI 1 Target Image x Translation Address Register

Register Name: P1_TIx_TADDR	Register Offset: 0x104, 0x114, 0x124, 0x134

PCI Bits	Function								PB Bits
31-24	TADDR								0-7
23-16	TADDR								8-15
15-08	PowerSpan II Reserved								16-23
07-00	M7	M6	M5	M4	M3	M2	M1	0	24-31

P1_TIx_TADDR Description

Name	Type	Reset By	Reset State	Function
TADDR[15:0]	R/W	P1_RST	0	Translation Address (through substitution)
M7-M1	R/W	P1_RST	0	Master Select 0=Do not claim transactions generated by this master 1=Claim transactions generated by this master

TADDR[15:0]: When the TA_EN bit in the P1_TIx_CTL register is set, TADDR[15:0] replaces the PCI-1 bus upper address bits. It replaces the upper address bits up to the size of the image.

The TADDR[15:0] field can be changed while transactions are being processed by PowerSpan II. This is the only parameter that can be changed during a transaction. All other programmable parameters must stay constant during a transaction

M7-M1: These bits indicate which external master(s) are qualified to access the image. The image supports master decode if the PCI Arbiter is enabled (the P1_ARB_EN bit in the RST_CSR register is set) and when the MD_EN bit in the P1_TIx_CTL register is set.

Table 95 details external arbitration pins associated with bits M7-M1. The shaded combinations in the table identify external arbitration pins which can be used for PCI-1, depending on the programming of the PCI_M7, PCI_M6,PCI_M5 bits in the MISC_CSR register.

Table 95: Arbitration Pin Mapping

Register Bit	External Arbitration Pins
M1	P1_REQ#[1]/P1_GNT#[1]
M2	P1_REQ#[2]/P1_GNT#[2]
M3	P1_REQ#[3]/P1_GNT#[3]
M4	P1_REQ#[4]/P1_GNT#[4]
M5	PCI_REQ#[5]/PCI_GNT#[5]
M6	PCI_REQ#[6]/PCI_GNT#[6]
M7	PCI_REQ#[7]/PCI_GNT#[7]

13.5.14 PCI-1 to PCI-2 Configuration Cycle Information Register

This register is used to set up the address phase of a PCI configuration cycle on PCI-2.

Table 96: PCI 1 to PCI-2 Configuration Cycle Information Register

Register Name: P1_CONF_INFO	**Register Offset: 0x144**

<table>
<tr><th>PCI
Bits</th><th colspan="4">Function</th><th>PB
Bits</th></tr>
<tr><td>31-24</td><td colspan="4">PowerSpan II Reserved</td><td>0-7</td></tr>
<tr><td>23-16</td><td colspan="4">BUS_NUM</td><td>8-15</td></tr>
<tr><td>15-08</td><td>DEV_NUM</td><td colspan="3">FUNC_NUM</td><td>16-23</td></tr>
<tr><td>07-00</td><td colspan="2">REG_NUM</td><td>0</td><td>TYPE</td><td>24-31</td></tr>
</table>

P1_CONF_INFO Description

Name	Type	Reset By	Reset State	Function
BUS_NUM [7:0]	R/W	P1_RST	0	Bus Number
DEV_NUM [4:0]	R/W	P1_RST	0	Device Number
FUNC_ NUM [2:0]	R/W	P1_RST	0	Function Number
REG_NUM [5:0]	R/W	P1_RST	0	Register Offset
TYPE	R/W	P1_RST	0	Configuration Cycle Type 0 = Type 0 1 = Type 1

For a Configuration Type 1 cycle — with the TYPE bit set to 1— an access of the PCI-1 Configuration Data register performs a corresponding Configuration Type 1 cycle on the PCI-2 Interface. During the address phase of the Configuration Type 1 cycle, the PCI-2 address lines carry the values encoded in the P1_CONF_INFO register (P2_AD[31:0] = P1_CONF_INFO[31:0]).

For a Configuration Type 0 cycle — with the TYPE bit set to 0 — an access of the PCI Configuration Data register performs a corresponding Configuration Type 0 cycle on the PCI-2 Interface. Programming the Device Number causes one of the upper address lines, P2_AD[31:11], to be asserted during the address phase of the Configuration
Type 0 cycle as defined in Table 97.

Table 97: PCI-2 AD[31:11] lines asserted during Configuration Type 0 cycles

DEV_NUM[4:0]	P2_AD[31:11]
00000	0000 0000 0000 0001 0000 0
00001	0000 0000 0000 0010 0000 0
00010	0000 0000 0000 0100 0000 0
00011	0000 0000 0000 1000 0000 0
00100	0000 0000 0001 0000 0000 0
00101	0000 0000 0010 0000 0000 0
00110	0000 0000 0100 0000 0000 0
00111	0000 0000 1000 0000 0000 0
01000	0000 0001 0000 0000 0000 0
01001	0000 0010 0000 0000 0000 0
01010	0000 0100 0000 0000 0000 0
01011	0000 1000 0000 0000 0000 0
01100	0001 0000 0000 0000 0000 0
01101	0010 0000 0000 0000 0000 0
01110	0100 0000 0000 0000 0000 0
01111	1000 0000 0000 0000 0000 0
10000	0000 0000 0000 0000 0000 1
10001	0000 0000 0000 0000 0001 0
10010	0000 0000 0000 0000 0010 0
10011	0000 0000 0000 0000 0100 0
10100	0000 0000 0000 0000 1000 0
10101-11111	0000 0000 0000 0000 0000 0

The remaining address lines are:

- P2_AD[10:8] = FUNC_NUM[2:0]
- P2_AD[7:2] = REG_NUM[5:0]
- P2_AD[1:0] = 00

This register is not implemented in the Single PCI PowerSpan II and must be treated as reserved.

13.5.15 PCI-1 to PCI-2 Configuration Cycle Data Register

A write to the Configuration Data register from the PCI-1 bus causes a Configuration Write Cycle to be generated on the PCI-2 Interface. This is defined by the Configuration Address register (P1_CONF_ADDR). A read of this register from the PCI-1 bus causes a Configuration Read Cycle to be generated on the PCI-2 Interface. The PCI Bus Configuration Cycles generated by accessing the Configuration Data register are handled as a posted write or delayed read.

A write to the PCI Configuration Data register from the PCI-2 Interface or the Processor Bus has no effect. A read from PCI-2 Interface or the Processor Bus returns undefined data.

Table 98: PCI 1 to PCI-2 Configuration Cycle Data Register

Register Name: P1_CONF_DATA	Register Offset: 0x148

PCI Bits	Function	PB Bits
31-24	CDATA	0-7
23-16	CDATA	8-15
15-08	CDATA	16-23
07-00	CDATA	24-31

P1_CON_DAT Description

Name	Type	Reset By	Reset State	Function
CDATA [31:0]	R/W	P1_RST	0	Configuration data

This register is not implemented in the Single PCI PowerSpan II and must be treated as reserved.

13.5.16 PCI-1 to PCI-2 Interrupt Acknowledge Cycle Generation Register

This register generates an Interrupt Acknowledge cycle on PCI-2.

Table 99: PCI 1 to PCI-2 Interrupt Acknowledge Cycle Generation Register

Register Name: P1_IACK	Register Offset: 0x14C

PCI Bits	Function	PB Bits
31-24	IACK_VEC	0-7
23-16	IACK_VEC	8-15
15-08	IACK_VEC	16-23
07-00	IACK_VEC	24-31

P1_ACK Description

Name	Type	Reset By	Reset State	Function
IACK_VEC [31:0]	R	P1_RST	0	PCI IACK Cycle Vector

Reading this register from the PCI-1 bus causes an IACK cycle to be generated on the PCI-2 Interface. The byte lanes enabled on the PCI-2 bus are determined by P1_CBE#[3:0] of the PCI-1 Memory Read cycle. The address on the PCI-1 bus used to access the P1_IACK register is passed directly over to the PCI-2 bus during the PCI IACK cycle. However, address information is ignored during PCI IACK cycles and has no effect.

Reads from this register behave as delayed transfers. This means that the PCI-1 bus master is retried until the read data is latched from the PCI-2 target. When the IACK cycle completes on the PCI-2 bus, the IACK_VEC[31:0] field is returned as read data when the PCI-1 bus master returns after the retry.

Writing to this register from the Processor Bus or either PCI bus has no effect. Reads from the PCI-2 Interface and Processor Bus return all zeros.

This register is not implemented in the Single PCI PowerSpan II and must be treated as reserved.

13.5.17 PCI-1 Bus Error Control and Status Register

The PCI-1 bus interface logs errors when it detects a Parity Error, Master-Abort, Target-Abort or Maximum Retry conditions

Table 100: PCI 1 Bus Error Control and Status Register

Register Name: P1_ERRCS	**Register Offset: 150**

<table>
<tr><th>PCI
Bits</th><th colspan="3">Function</th><th>PB
Bits</th></tr>
<tr><td>31-24</td><td>PowerSpan II Reserved</td><td>MES</td><td>ES</td><td>0-7</td></tr>
<tr><td>23-16</td><td colspan="3">PowerSpan II Reserved</td><td>8-15</td></tr>
<tr><td>15-08</td><td colspan="3">PowerSpan II Reserved</td><td>16-23</td></tr>
<tr><td>07-00</td><td>CMDERR</td><td colspan="2">PowerSpan II Reserved</td><td>24-31</td></tr>
</table>

P1_ERRCS Description

Name	Type	Reset By	Reset State	Function
MES	R	P1_RST	0	Multiple Error Status 1 = a second error occurred before the first error could be cleared.
ES	R/Write 1 to clear	P1_RST	0	Error Status 0 = no error currently logged 1 = error currently logged
CMDERR [3:0]	R	P1_RST	0	PCI Command Error Log

When the ES bit is set, it means an error has been logged and the contents of the CMDERR[3:0] and PAERR[31:0] of the P1_AERR register are valid. Information in the log cannot be changed while ES is set. Clearing the ES by writing 1 to the bit allows the error log registers to capture future errors.

MES is useful in determining if multiple errors occur. The original error logging is not overwritten when MES is set. Clearing ES also clears the MES bit.

13.5.18 PCI-1 Address Error Log Register

The PCI-1 Interface logs errors when it detects a Parity Error, Master-Abort, Target-Abort or Maximum Retry conditions.

Table 101: PCI 1 Address Error Log Register

Register Name: P1_AERR	Register Offset: 0x154

PCI Bits	Function	PB Bits
31-24	PAERR	0-7
23-16	PAERR	8-15
15-08	PAERR	16-23
07-00	PAERR	24-31

P1_AERR Description

Name	Type	Reset By	Reset State	Function
PAERR [31:0]	R	P1_RST	0	PCI Address Error Log

The address of a PCI-1 bus transaction that generates an error condition is logged in this register. When the error occurs, the ES bit in the P1_ERRCS register is set, qualifying and freezing the contents of this register. This register logs additional errors only after the ES bit in the P1_ERRCS register is cleared.

13.5.19 PCI-1 Miscellaneous Control and Status Register

Table 102: PCI 1 Miscellaneous Control and Status Register

Register Name: P1_MISC_CSR	**Register Offset: 0x160**

PCI Bits	Function			PB Bits
31-24	PowerSpan II Reserved			0-7
23-16	PowerSpan II Reserved			8-15
15-08	BSREG_BAR_EN	PowerSpan II Reserved	MAX_RETRY	16-23
07-00	MAC_ERR	PowerSpan II Reserved		24-31

P1_MISC Description

Name	Type	Reset By	Reset State	Function
BSREG_BAR_EN	R/W	P1_RST	1 EEPROM	PCI-1 Registers Image Base Address Register enable. 0=disable 1=enable
MAX_RETRY[3:0]	R/W	P1_RST	0	Maximum number of PCI Retry Terminations 0000 = retry forever 0001 = 64 retries other - 2^{24} retries
MAC_ERR	R/W	P1_RST	1	Master Abort Configuration Error Mapping 0 = generate target abort when master abort occurs on PCI-2 configuration cycles 1 = return all ones when Master-Abort occurs on PCI-2 configuration cycles
				Single PCI PowerSpan II: Reserved

BSREG_BAR_EN: When this bit is cleared, the P1_BSREG register is not visible in PCI-1 Configuration space and reads zero. Also, when this bit is cleared writes have no effect when this bit is cleared.

When the P1_BSREG register is not visible in PCI-1 Configuration space, the PowerSpan II PCI-1 register image is disabled and PowerSpan II does not request PCI Memory space for the image.

13.5.20 PCI-1 Bus Arbiter Control Register

PowerSpan II's PCI-1 interface has dedicated support for four external PCI masters. The user can assign up to three additional PCI masters to the PCI-1 arbiter by configuring the PCI_M7 bit, the PCI_M6 bit, and the PCI_M5 in the MISC_CSR register (see page 378).

Depending on the number of external masters supported, some of bits M4-M7 and combinations of BM_PARK are not applicable. Programming these combinations result in unpredictable PowerSpan II behavior.

The PowerSpan II PCI-1 internal arbiter is enabled by a power-up option. When disabled, an external arbiter is used. The signals P1_REQ[1]_/P1_GNT[1]_ are used by the PowerSpan II PCI-1 Master to arbitrate for access to the bus.

The P1_ARB_EN bit in the RST_CSR register (see page 385) specifies if the PCI-1 arbiter is enabled or disabled.

Table 103: PCI 1 Bus Arbiter Control Register

Register Name: P1_ARB_CTRL	**Register Offset: 0x164**

PCI Bits	**Function**								**PB Bits**
31-24	PowerSpan II Reserved								0-7
23-16	STATUS_BITS								8-15
15-08	M7_PRI	M6_PRI	M5_PRI	M4_PRI	M3_PRI	M2_PRI	M1_PRI	PS_PRI	16-23
07-00	PowerSpan II Reserved			STATUS_EN	PARK	BM_PARK			24-31

P1_ARB_CTRL Description

Name	**Type**	**Reset By**	**Reset State**	**Function**
STATUS_BITS	R/W	P1_RST	0	Operational status of PCI Master Device *x* 0 = functioning 1 = non-functioning
Mx_PRI	R/W	P1_RST	0	Arbitration Level for PCI Master Device *x* 0 = low priority 1 = high priority

P1_ARB_CTRL Description

Name	Type	Reset By	Reset State	Function
PS_PRI	R/W	P1_RST	0	Arbitration Level for PowerSpan II 0 = low priority 1 = high priority
STATUS_EN	R/W	P1_RST	0	Enable monitoring of master by arbiter 0 = disabled 1 = enabled
PARK	R/W	P1_RST	0	PCI-1 Bus Parking Algorithm 0 = last master 1 = select master
BM_PARK [2:0]	R/W	P1_RST	0	Parked Master

STATUS_BITS: These series of bits are separated per master. There is one bit designated for each master and is separate from the others, but all eight are called STATUS_BITS[7:0]. The individual bits are set when a PCI Master does not respond to a grant given by the PowerSpan II arbiter for 16 clock cycles. Once this bit is set to 1 by the PowerSpan II arbiter, the PowerSpan II arbiter does not include the non-functioning PCI Master in the arbitration algorithm used by PowerSpan II. When the bit is set to 0, the operating status of the PCI Master is functioning and it is included in the arbitration algorithm used by PowerSpan II.

Mx_PRI: Determines the arbitration priority level for PCI Master agents assigned to the PCI-1 arbiter.

STATUS_EN: Enables internal monitor of the PowerSpan II PCI arbiter. The monitor checks that no PCI Master waits longer than 16 PCI clock cycles before starting a transaction. The default setting is 0.

PARK: When this bit is set the arbiter parks the PCI-1 bus on the PCI master programmed in the BM_PARK[2:0] field. When cleared the arbiter parks the PCI-1 bus on the last PCI master to be granted the bus.

BM_PARK[2:0]: This field instructs the arbiter where to park the bus. The shaded combinations in Table 104 identifies potential PCI-1 external masters. Their presence depends on the programming of the PCI_M7,PCI_M6,PCI_M5 bits in the MISC_CSR register.

Table 104: Parked PCI Master

BM_PARK [2:0]	Parked PCI Master	External Pins
000	PowerSpan II	None
001	M1	P1_REQ#[1]/P1_GNT#[1]
010	M2	P1_REQ#[2]/P1_GNT#[2]
011	M3	P1_REQ#[3]/P1_GNT#[3]
100	M4	P1_REQ#[4]/P1_GNT#[4]
101	M5	PCI_REQ#[5]/PCI_GNT#[5]
110	M6	PCI_REQ#[6]/PCI_GNT#[6]
111	M7	PCI_REQ#[7]/PCI_GNT#[7]

13.5.21 Processor Bus Slave Image x Control Register

This register contains the control information for the "Processor Bus Slave Image x Base Address Register" on page 350.

The bits in this register are not dynamic. Do not alter these settings while transactions are being processed through PowerSpan II. Refer to "Processor Bus Slave Image x Translation Address Register" on page 348 for more information on dynamic address translation.

Table 105: Processor Bus Slave Image x Control Register

Register Name: PB_SIx_CTL	Register Offset: 0x200, 0x210, 0x220, 0x230, 0x240, 0x250, 0x260, 0x270

PCI Bits	Function						PB Bits
31-24	IMG_EN	TA_EN	MD_EN	BS			0-7
23-16	MODE	DEST	MEM_IO	PowerSpan II Reserved			8-15
15-08	PowerSpan II Reserved						16-23
07-00	PRKEEP	END		0	0	RD_AMT	24-31

PB_SIx_CTL Description

Name	Type	Reset By	Reset State	Function
IMG_EN	R/W	PB_RST	0 EEPROM	Image Enable 0=Disable 1=Enable
TA_EN	R/W	PB_RST	0 EEPROM	Translation Address Enable 0=Disable 1=Enable
MD_EN	R/W	PB_RST	0 EEPROM	Master Decode Enable 0=Disable 1=Enable
BS[4:0]	R/W	PB_RST	0 EEPROM	Block Size (4 Kbyte*2^{BS})

PB_SIx_CTL Description

Name	Type	Reset By	Reset State	Function
MODE	R/W	PB_RST	0 EEPROM	Image Mode 0 = Memory command generation 1 = I/O command generation or 4 byte memeory read (see Table 107 on page 346)
DEST	R/W	PB_RST	0 EEPROM	Destination Bus 0 = PCI 1 bus 1 = PCI-2 bus
				Single PCI PowerSpan II: Reserved PCI-1 Bus is the only destination.
MEM_IO	R/W	PB_RST	0	MEM_IO mode 0 = Regular I/O mode 1 = Enables 1,2,3, or 4 byte memory reads on the PCI bus(es)
PRKEEP	R/W	PB_RST	0 EEPROM	Prefetch Read Keep 0 = purge read data at end of transfer 1 = keep read data
END[1:0]	R/W	P1_RST	10b EEPROM	Endian Conversion Mode 00 = Little-endian 01 = PowerPC little-endian 10 = Big-endian 11 = True little-endian
RD_AMT [2:0]	R/W	PB_RST	0 EEPROM	Read Prefetch Amount Amount of read data fetched from PCI

IMG_EN: The Image Enable bit is changed in the following ways:

- EEPROM initialization
- register write to IMG_EN

The IMG_EN is cleared by writing a zero to the bit.

TA_EN: When set, the Translation Address (PB_SIx_TADDR) replaces the upper bits of the Processor Bus address. Clearing the enable results in no address translation.

MD_EN: Enables Master Decode when the Processor Bus arbiter is in use — the Processor Bus Arbiter Enable bit, in the "Reset Control and Status Register" on page 385, is set. If MD_EN is cleared, only the Processor Bus Address and Transaction Type are used for transaction decode. If MD_EN is set, the originating master is included in the transaction decode. A transaction is claimed only if it originates from the master(s) specified in PB_SIx_TADDR.

BS: The Block Size specifies the size of the image, address lines compared and address lines translated.

Table 106: Block Size

BS[4:0]	Block Size	Address Lines Compared/ Translated
00000	4k	A0-A19
00001	8k	A0-A18
00010	16k	A0-A17
00011	32k	A0-A16
00100	64k	A0-A15
00101	128k	A0-A14
00110	256k	A0-A13
00111	512k	A0-A12
01000	1M	A0-A11
01001	2M	A0-A10
01010	4M	A0-A9
01011	8M	A0-A8
01100	16M	A0-A7
01101	32M	A0-A6
01110	64M	A0-A5
01111	128M	A0-A4
10000	256M	A0-A3
10001	512M	A0-A2

Table 106: Block Size

BS[4:0]	Block Size	Address Lines Compared/ Translated
10010	1G	A0-A1
10011	2G	A0
10100-11111	Reserved	Reserved

MODE: Determines if the image is used to generate Memory or IO commands on PCI.

- Memory Command Mode

  Causes PCI Memory commands to be generated on PCI. Bursting is supported.

  PRKEEP and RD_AMT[2:0] are only applicable in Memory Command Mode.

- IO Command Mode

  Causes PCI IO commands to be generated on PCI. When the image is selected to perform IO commands, transactions are limited to 4 bytes or less. A transaction attempting to move more than 4 bytes will cause a TEA_ response. The TEA_ can be suppressed by setting the PB_MISC_CSR[TEA_EN] bit.

The MODE bit and the MEM_IO bit work together to control the size of the transaction (see Table 107)..

Table 107: Setting for MODE and MEM_IO Bits

MODE Setting	MEM_IO setting	Transaction size
0	0	Memory cycle (minimum 8 byte memory read)
0	1	Memory cycle (minimum 8 byte memeory read)
1	0	I/O Cycle
1	1	Memory cycle (minimum 4 byte memeory read)

DEST: Selects the destination bus for the transaction.

MEM_IO: PowerSpan II supports 4-byte reads. When this bit is set, the Memory Read command to the corresponding image generates the Memory Read command on the destination PCI bus with a minimum 32 bit aligned 4-byte read.

The MODE bit and the MEM_IO bit work together to control the size of the transaction (see Table 107).

PRKEEP: Prefetch Read Keep stores prefetch data beyond an initial read. When set, subsequent read requests to the same image at the next address retrieves the read data directly from the switching fabric instead of causing either PCI bus to fetch more data. The read data is invalidated when a read with a non-matching address occurs.

The ARTRY_EN bit must be set to 1 in order for the PowerSpan II Prefetch Keep feature to keep prefetched data. The ARTRY_EN bit is in the Processor Bus Miscellaneous Control and Status register (see page 361).

END: Selects the endian mapping.

RD_AMT[2:0]: The Read Amount setting determines different values to prefetch from PCI. If PRKEEP is not set, it is recommended limiting the RD_AMT to 32-bytes.

Table 108: Read Amount

RD_AMT[2:0]	Data Fetched
000	8 bytes
001	16 bytes
010	32 bytes
011	64 bytes
100	128 bytes
101-111	Reserved

If the slave image is programmed to be in IO mode (the MODE bit in the PB_SIx_CTL register set to 1) then the RD_AMT is not used and a maximum of 4 bytes will be read from the PCI bus.

The EEPROM load capability allows a processor on the Processor Bus to boot directly from a device on PCI. Only the control registers for Processor Bus Slave Image 0 are loaded from EEPROM.

13.5.22 Processor Bus Slave Image x Translation Address Register

Table 109: Processor Bus Slave Image x Translation Address Register

Register Name: PB_SIx_TADDR	Register Offset: 0x204, 0x214, 0x224, 0x234, 0x244, 0x254, 0x264, 0x274

<table>
<tr><th>PCI
Bits</th><th colspan="4">Function</th><th>PB
Bits</th></tr>
<tr><td>31-24</td><td colspan="4">TADDR</td><td>0-7</td></tr>
<tr><td>23-16</td><td colspan="4">TADDR</td><td>8-15</td></tr>
<tr><td>15-08</td><td>TADDR</td><td colspan="3">PowerSpan II Reserved</td><td>16-23</td></tr>
<tr><td>07-00</td><td>PowerSpan II Reserved</td><td>M3</td><td>M2</td><td>M1</td><td>0</td><td>24-31</td></tr>
</table>

PB_SIx_TADDR Description

Name	Type	Reset By	Reset State	Function
TADDR[19:0]	R/W	PB_RST	0 EEPROM	Translation Address
M3-M1	R/W	PB_RST	0 EEPROM	Master Select 0=Do not claim transactions generated by this master 1=Claim transactions generated by this master

TADDR: The Translation Address register replaces the Processor Bus address, up to the size of the image. TADDR[31:12] replace the Processor Bus PB_A[0:19].

For example, if TADDR[31:12] = 0x12345 and PB_SIx_CTL[BS]=0 (4 K image) and the address on the Processor Bus is PB_A[0:31] = 0x78563412, then the PCI address becomes 0x12345412

The TADDR[19:0] field can be changed while transactions are being processed by PowerSpan II. This is the only parameter that can be changed during a transaction. All other programmable parameters must stay constant during a transaction

.

Table 110: Translation Address Mapping

PB_SIx_ TADDR[]	Processor Bus Address PB_A	PB_SIx_CTL[BS]	Block Size
31	0	10011	2G
31:30	0:1	10010	1G
31:29	0:2	10001	512M
31:28	0:3	10000	256M
31:27	0:4	01111	128M
31:26	0:5	01110	64M
31:25	0:6	01101	32M
31:24	0:7	01100	16M
31:23	0:8	01011	8M
31:22	0:9	01010	4M
31:21	0:10	01001	2M
31:20	0:11	01000	1M
31:19	0:12	00111	512k
31:18	0:13	00110	256k
31:17	0:14	00101	128k
31:16	0:15	00100	64k
31:15	0:16	00011	32k
31:14	0:17	00010	16k
31:13	0:18	00001	8k
31:12	0:19	00000	4k

M3-M1: These bits indicate which external master(s) are qualified to access the image. The image supports master decode if the Processor Bus arbiter is enabled — the Processor Bus Arbiter Enable bit, in the “Reset Control and Status Register” on page 385, is set and when MD_EN bit in the PB_SIx_CTL is set. Bit M3 represents the external master connected to PB_BG[3]_ and M1 represents the external master connected to PB_BG[1]_.

13.5.23 Processor Bus Slave Image x Base Address Register

This register defines the lowest address of the slave image. The minimum image size is 4 Kbyte as defined in BS field in the PB_SIx_CTL.

The initial write to this register sets the IMG_EN bit in the PB_SIx_CTL register. Subsequent writes to this register will have no effect on the IMG_EN bit.

A base address of 0 is valid.

Table 111: Processor Bus Slave Image x Base Address Register

Register Name: PB_SIx_BADDR	**Register Offset: 0x208, 0x218, 0x228, 0x238, 0x248,0x 258, 0x268, 0x278**

<table>
<tr><th>PCI
Bits</th><th colspan="5">Function</th><th>PB
Bits</th></tr>
<tr><td>31-24</td><td colspan="5">BA</td><td>0-7</td></tr>
<tr><td>23-16</td><td colspan="5">BA</td><td>8-15</td></tr>
<tr><td>15-08</td><td>BA</td><td>0</td><td>0</td><td>0</td><td>0</td><td>16-23</td></tr>
<tr><td>07-00</td><td colspan="5">PowerSpan II Reserved</td><td>24-31</td></tr>
</table>

PB_SIx_BADDR Description

Name	Type	Reset By	Reset State	Function
BA[19:0]	R/W	PB_RST	0 EEPROM	Processor Bus Base Address

13.5.24 Processor Bus Register Image Base Address Register

This register defines the Processor Bus address offset for PowerSpan II internal registers. The register can be loaded by an external EEPROM.

A base address of 0 is valid.

Table 112: Processor Bus Register Image Base Address Register

Register Name: PB_REG_BADDR	**Register Offset: 0x280**

<table>
<tr><th>PCI
Bits</th><th colspan="3">Function</th><th>PB
Bits</th></tr>
<tr><td>31-24</td><td colspan="3">BA</td><td>0-7</td></tr>
<tr><td>23-16</td><td colspan="3">BA</td><td>8-15</td></tr>
<tr><td>15-08</td><td>BA</td><td colspan="2">PowerSpan II Reserved</td><td>16-23</td></tr>
<tr><td>07-00</td><td colspan="2">PowerSpan II Reserved</td><td>END</td><td>24-31</td></tr>
</table>

PB_REG_BADDR Description

Name	Type	Reset By	Reset State	Function
BA[19:0]	R/W	PB_RST	0x30000 EEPROM	Processor Bus Register Base Address
END	R/W	PB_RST	0 EEPROM	Endian Conversion Mode 0 = Big-endian 1 = PowerPC little-endian

BA: The base address for the Processor Bus Base Address image represent the upper address bits (A[31:12]). The base address for the processor address bus at reset is 0x3000_0000.

END[1:0]: The endian conversion mode for processor access to PowerSpan II registers.

13.5.25 Processor Bus PCI Configuration Cycle Information Register

This register is used to set up the address phase of a PCI Configuration cycle.

Table 113: Processor Bus PCI Configuration Cycle Information Register

Register Name: PB_CONF_INFO	Register Offset: 0x290

PCI Bits	Function								PB Bits
31-24	0	0	0	0	0	0	0	DEST	0-7
23-16	BUS_NUM								8-15
15-08	DEV_NUM					FUNC_NUM			16-23
07-00	REG_NUM						0	TYPE	24-31

PB_CONF_INFO Description

Name	Type	Reset By	Reset State	Function
DEST	R/W	PB_RST	0	Destination Bus 0 = PCI 1 1 = PCI-2
DEST	R/W	PB_RST	0	Single PCI PowerSpan II: Reserved PCI-1 Bus is the only destination.
BUS_NUM[7:0]	R/W	PB_RST	0	Bus Number
DEV_NUM[4:0]	R/W	PB_RST	0	Device Number
FUNC_NUM[2:0]	R/W	PB_RST	0	Function Number
REG_NUM[5:0]	R/W	PB_RST	0	Register Number
TYPE	R/W	PB_RST	0	Configuration Cycle Type 0 = Type 0 1 = Type 1

For a Configuration Type 1 cycle — with the TYPE bit set to 1— an access of the PCI Configuration Data register performs a corresponding Configuration Type 1 cycle on either PCI bus. During the address phase of the Configuration Type 1 cycle, the PCI address lines carry the values encoded in the PB_CONF_INFO register (AD[31:0] = PB_CONF_INFO[31:0]). The Destination (DEST) field, in the PB_CONF_INFO register, is an exception to this because it is zero on AD[24].

For a Configuration Type 0 cycle — with the TYPE bit set to 0 — an access of the PCI Configuration Data register performs a corresponding Configuration Type 0 cycle on either PCI bus. Programming the Device Number causes the assertion of one of the upper address lines, AD[31:11], during the address phase of the Configuration Type 0 cycle. This is shown in Table 114.

Table 114: PCI AD[31:11] lines asserted during Configuration Type 0 cycles

DEV_NUM[4:0]	AD[31:11]
00000	0000 0000 0000 0001 0000 0
00001	0000 0000 0000 0010 0000 0
00010	0000 0000 0000 0100 0000 0
00011	0000 0000 0000 1000 0000 0
00100	0000 0000 0001 0000 0000 0
00101	0000 0000 0010 0000 0000 0
00110	0000 0000 0100 0000 0000 0
00111	0000 0000 1000 0000 0000 0
01000	0000 0001 0000 0000 0000 0
01001	0000 0010 0000 0000 0000 0
01010	0000 0100 0000 0000 0000 0
01011	0000 1000 0000 0000 0000 0
01100	0001 0000 0000 0000 0000 0
01101	0010 0000 0000 0000 0000 0
01110	0100 0000 0000 0000 0000 0
01111	1000 0000 0000 0000 0000 0
10000	0000 0000 0000 0000 0000 1
10001	0000 0000 0000 0000 0001 0

Table 114: PCI AD[31:11] lines asserted during Configuration Type 0 cycles

DEV_NUM[4:0]	AD[31:11]
10010	0000 0000 0000 0000 0010 0
10011	0000 0000 0000 0000 0100 0
10100	0000 0000 0000 0000 1000 0
10101-11111	0000 0000 0000 0000 0000 0

The remaining address lines are:

- AD[10:8] = FUNC_NUM[2:0]
- AD[7:2] = REG_NUM[5:0]
- AD[1:0] = 00

PowerSpan II does not generate configuration cycles to devices connected to AD[15:11].

13.5.26 Processor Bus Configuration Cycle Data Register

Table 115: Processor Bus Configuration Cycle Data Register

Register Name: PB_CONF_DATA	Register Offset: 0x294

PCI Bits	Function	PB Bits
31-24	CDATA	0-7
23-16	CDATA	8-15
15-08	CDATA	16-23
07-00	CDATA	24-31

PB_CONF_DATA Description

Name	Type	Reset By	Reset State	Function
CDATA[31:0]	R/W	PB_RST	0	Configuration Data

A write to the Configuration Data register from the Processor Bus causes a Configuration Write Cycle to be generated on either PCI bus as defined by the Configuration Info register (PB_CONF_INFO). A read of this register from the Processor Bus causes a Configuration Read Cycle to be generated on either PCI bus. The PCI Bus Configuration Cycles generated by accessing the Configuration Data register is handled as a posted write or delayed read.

The byte lanes enabled on the PCI bus are determined by PB_SIZ[0:3] and PB_A[30:31] of the Processor Bus read or write cycle.

A write to the PCI Configuration Data register from the either PCI bus has no effect. A read from either PCI bus is undefined.

The END bit in the PB_REG_BADDR selects the endian conversion scheme used for accesses to PCI through this register. The definition of endian conversion scheme is for PCI accesses, not register accesses.

13.5.27 Processor Bus to PCI-1 Interrupt Acknowledge Cycle Generation Register

This register is used to generate an Interrupt Acknowledge cycle originating on the Processor Bus and destined for the PCI-1 bus. Reading this register from the Processor Bus causes an IACK cycle to be generated on the PCI bus. The byte lanes enabled on the PCI bus are determined by PB_SIZ[0:3] and PB_A[30:31] of the Processor Bus read cycle. The address on the Processor Bus used to access the PB_P1_IACK register is passed directly over to the PCI bus during the PCI IACK cycle. However, address information is ignored during PCI IACK cycles, so this has no effect.

Table 116: Processor Bus to PCI-1 Interrupt Acknowledge Cycle Generation Register

Register Name: PB_P1_IACK	Register Offset: 0x2A0

PCI Bits	Function	PB Bits
31-24	IACK_VEC	0-7
23-16	IACK_VEC	8-15
15-08	IACK_VEC	16-23
07-00	IACK_VEC	24-31

PB_P1_IACK Description

Name	Type	Reset By	Reset State	Function
IACK_VEC[31:0]	R	PB_RST	0	PCI IACK Cycle Vector

If the Address Retry Enable (ARTRY_EN) bit is set, in the“PCI 1 Miscellaneous 1 Register” on page 315, the Processor Bus Master is retried until the read data is latched from the PCI target. When the IACK cycle completes on the PCI-1 bus, the IACK_VEC[31:0] field is returned as read data when the Processor Bus Master returns after the retry.

Writing to this register from the Processor Bus or either PCI bus has no effect. Reads from the PCI bus return all zeros.

The END bit in the PB_REG_BADDR selects the endian conversion scheme used for accesses to PCI through this register. The definition of endian conversion scheme is for PCI accesses, not register accesses.

13.5.28 Processor Bus to PCI-2 Interrupt Acknowledge Cycle Generation Register

This register is used to generate an Interrupt Acknowledge cycle originating on the Processor Bus and destined for the PCI-2 bus. Reading this register from the Processor Bus causes an IACK cycle to be generated on the PCI bus. The byte lanes enabled on the PCI bus are determined by PB_SIZ[0:3] and PB_A[30:31] of the Processor Bus read cycle. The address on the Processor Bus used to access the PB_P1_IACK register is passed directly over to the PCI bus during the PCI IACK cycle. However this has no effect because address information is ignored during PCI IACK cycles.

Table 117: Processor Bus to PCI-2 Interrupt Acknowledge Cycle Generation Register

Register Name: PB_P2_IACK	Register Offset: 0x2A4

PCI Bits	Function	PB Bits
31-24	IACK_VEC	0-7
23-16	IACK_VEC	8-15
15-08	IACK_VEC	16-23
07-00	IACK_VEC	24-31

PB_P2_IACK Description

Name	Type	Reset By	Reset State	Function
IACK_VEC[31:0]	R	PB_RST	0	PCI IACK Cycle Vector

If the Address Retry Enable (ARTRY_EN)) bit is set, in the“PCI 1 Miscellaneous 1 Register” on page 315, the Processor Bus Master is retried until the read data is latched from the PCI target. When the IACK cycle completes on the PCI-2 bus, the IACK_VEC[31:0] field is returned as read data when the Processor Bus Master returns after the retry.

Writing to this register from the Processor Bus, or either PCI bus, has no effect. Reads from the PCI bus return all zeros.

The END bit in the PB_REG_BADDR selects the endian conversion scheme used for accesses to PCI through this register. The definition of endian conversion scheme is for PCI accesses, not register accesses.

This register is not implemented in the Single PCI PowerSpan II and must be treated as reserved.

13.5.29 Processor Bus Error Control and Status Register

The Processor Bus Interface logs errors when it detects a maximum retry error, parity error or assertion of PB_TEA_ conditions.

Table 118: Processor Bus Error Control and Status Register

Register Name: PB_ERRCS	**Register Offset: 0x2B0**

<table>
<tr><th>PCI
Bits</th><th colspan="5">Function</th><th>PB
Bits</th></tr>
<tr><td>31-24</td><td colspan="3">PowerSpan II Reserved</td><td>MES</td><td>ES</td><td>0-7</td></tr>
<tr><td>23-16</td><td colspan="5">PowerSpan II Reserved</td><td>8-15</td></tr>
<tr><td>15-08</td><td colspan="2">TT_ERR</td><td colspan="3">PowerSpan II Reserved</td><td>16-23</td></tr>
<tr><td>07-00</td><td>SIZ_ERR</td><td colspan="4">PowerSpan II Reserved</td><td>24-31</td></tr>
</table>

PB_ERRCS Description

Name	Type	Reset By	Reset State	Function
MES	R	PB_RST	0	Multiple Error Status 1 = a second error occurred before the first error could be cleared.
ES	R/Write 1 to Clear	PB_RST	0	Error Status 0 = no error currently logged 1 = error currently logged
TT_ERR[4:0]	R	PB_RST	0	Processor Bus Transaction Type Error Log
SIZ_ERR[3:0]	R	PB_RST	0	Processor Bus SIZ field Error Log

When the ES bit is set, it means an error has been logged and the contents of the TT_ERR, SIZ_ERR and AERR are valid. Information in the log cannot be changed while ES is set. Clearing the ES by writing a one to the bit allows the error log registers to capture future errors.

MES is useful for determining if multiple errors occur. The Processor Bus error logs are not overwritten when MES is set. Clearing ES also clears MES.

13.5.30 Processor Bus Address Error Log

The Processor Bus Interface logs errors when it detects a maximum retry error, parity error or assertion of PB_TEA_ conditions.

Table 119: Processor Bus Address Error Log

Register Name: PB_AERR	Register Offset: 0x2B4

PCI Bits	Function	PB Bits
31-24	AERR	0-7
23-16	AERR	8-15
15-08	AERR	16-23
07-00	AERR	24-31

PB_AERR Description

Name	Type	Reset By	Reset State	Function
AERR[31:0]	R	PB_RST	0	Processor Bus Error Log

The address of a processor bus transaction that generates an error condition is logged in this register. When the error occurs, the ES bit in the PB_ERRCS is set, qualifying and freezing the contents of this register. This register logs additional errors only after the ES bit (see Table 135 on page 388) is cleared.

13.5.31 Processor Bus Miscellaneous Control and Status Register

Table 120: Processor Bus Miscellaneous Control and Status Register

Register Name: PB_MISC_CSR	**Register Offset: 0x2C0**

PCI Bits	**Function**								**PB Bits**
31-24	PowerSpan II Reserved								0-7
23-16	PowerSpan II Reserved								8-15
15-08	PowerSpan II Reserved				MAX_RETRY				16-23
07-00	EXT CYC	MAC_ TEA	MODE_ 7400	TEA_EN	ARTRY_ EN	DP_EN	AP_EN	PARITY	24-31

PB_MISC_CSR Description

Name	**Type**	**Reset By**	**Reset State**	**Function**
MAX_RETRY [3:0]	R/W	PB_RST	0	Maximum number of retries. Except for 0000, all entries are multiples of 64 retries 0000 = retry forever 0001 = 64 retries 0010 = 128 retries 0011 = 192 retries, etc.
EXTCYC	R/W	PB_RST	0	Determines if the PowerSpan II PB master is enabled to generate extended cycles (16 byte or 24 byte) 0 = Cannot generate extended cycle 1= Can generate extended cycle
MAC_TEA	R/W	PB_RST	1	Master-Abort Configuration Error Mapping 0 = Assert PB_TEA_ when Master-Abort occurs on PCI configuration cycles 1 = Return all “1s” when Master-Abort occurs on PCI configuration cycles
MODE_7400	R/W	PB_RST	1	Determines if PowerSpan II Processor Bus Slave can accept misaligned data transfers defined for MPC7400 0 = cannot accept MPC7400 misaligned transfers 1 = can accept MPC7400 misaligned transfers

PB_MISC_CSR Description

Name	Type	Reset By	Reset State	Function
TEA_EN	R/W	PB_RST	1	Suppress PB_TEA_ generation 0 = PowerSpan II does not assert PB_TEA_ 1 = PowerSpan II asserts PB_TEA_
ARTRY_EN	R/W	PB_RST	0	Address Retry Enable 0 = PB Slave never asserts PB_ARTRY_ 1 = PB Slave asserts PB_ARTRY_ as required
DP_EN	R/W	PB_RST	0	Data Parity Enable 0 = Data parity checking disabled 1 = Data parity checking enabled
AP_EN	R/W	PB_RST	0	Address Parity Enable 0 = Address parity checking disabled 1 = Address parity checking enabled
PARITY	R/W	PB_RST	0	Parity 0 = Odd Parity 1 = Even Parity

EXTCYC: PowerSpan II generates 16 byte and 24 byte extended cycles when EXTCYC is set. This ability improves performance of PowerQUICC II systems. To maintain compatibility with other PowerPC devices, clearing EXTCYC ensures PowerSpan II never generates extended cycles.

MAC_TEA: This bit controls the handling of a Master-Abort while a PowerSpan II PCI Master is generating a configuration transaction initiated by a processor bus master. If MAC_TEA is cleared, the processor bus slave returns all ones to the initiating processor bus master. If MAC_TEA is cleared and TEA_EN is set to 1, the processor bus slave asserts PB_TEA_ to terminate the transaction initiated by the processor.

MODE_7400: PowerSpan II supports a specific set of PowerPC 7400 misaligned transactions when this bit is set to 1. Refer to Table 29 on page 131 for a complete list of data transfers supported by PowerSpan II.

TEA_EN: When this bit is cleared, PowerSpan II never asserts TEA_. Error conditions are signalled exclusively with interrupts.

ARTRY_EN: Controls PowerSpan II's use of PB_ARTRY_ during the servicing of transactions. When ARTRY_EN is set, the Processor Bus Slave retries a processor (60x) bus master under the following conditions:

- Register write while an external master connected to another PowerSpan II interface is doing a register write
- Register read during I^2C load
- Posted write when no buffers are available
- Read from PCI-1 or PCI-2

ARTRY_EN is cleared by default. The user will see improved Processor Bus Interface utilization by setting ARTRY_EN.

⚠ The ARTRY_EN bit must be set to 1 in order for the PowerSpan II Prefetch Keep feature to keep prefetched data. Prefetch Keep is enabled by setting the PRKEEP bit in the Processor Bus Slave Image *x* Base Address Control register (see page 350).

DP_EN, AP_EN: When cleared, the PowerSpan II does not check the parity pins for the proper parity value. PowerSpan II still drives out parity on master writes and slave read cycles. Parity checking is disabled by default.

13.5.32 Processor Bus Arbiter Control Register

The arbitration control register is used to control the parameters of the on-chip Processor Bus arbiter.

Table 121: Processor Bus Arbiter Control Register

Register Name: PB_ARB_CTRL	**Register Offset: 0x2D0**

PCI Bits	**Function**					**PPC Bits**
31-24	PowerSpan II Reserved					0-7
23-16	PowerSpan II Reserved	M3_EN	M2_EN	M1_EN	0	8-15
15-08	PowerSpan II Reserved	M3_PRI	M2_PRI	M1_PRI	PS_PRI	16-23
07-00	PowerSpan II Reserved	TS_DLY	PARK	BM_PARK		24-31

PB_ARB_CTRL Description

Name	**Type**	**Reset By**	**Reset State**	**Function**
Mx_EN	R/W	PB_RST	PWRUP	External Master x Enable 0=External requests ignored 1=External requests recognized
Mx_PRI	R/W	PB_RST	0	External Master x Priority Level 0 = Low Priority 1 = High Priority
PS_PRI	R/W	PB_RST	0	PowerSpan II Priority Level 0 = Low Priority 1 = High Priority
TS_DLY	R/W	PB_RST	0	Controls when arbiter samples requests 0 = sample clock after TS_ 1 = sample 2 clocks after TS_

PB_ARB_CTRL Description

Name	Type	Reset By	Reset State	Function
PARK	R/W	PB_RST	0	Bus Park Mode 0 = Park on last bus master 1 = Park on specific master
BM_PARK	R/W	PB_RST	0	Bus Master to be Parked 00 = PowerSpan II 01 = External Master 1 10 = External Master 2 11 = External Master 3

Mx_EN: When set, the arbiter recognizes address bus requests for this master. When cleared, the arbiter ignores address bus requests from this master. The default state for these bits is determined by the PWRUP_BOOT option as defined by the following table:

Table 122: Mx_EN Default State

PWRUP_BOOT Selection	RST_CSR Register	M1_EN	M2_EN	M3_EN
Boot PCI	PCI_BOOT=1	0	0	0
Boot Processor Bus	PCI_BOOT=0	1	0	0

Mx_PRI: Determines the arbitration priority for external masters.

TS_DLY: When set, the PB arbiter samples incoming requests two clocks after a TS_ signal is received. When cleared, the arbiter samples requests one clock after a TS_ signal is received. The default state is 0.

An example application for this feature is some L2 caches hold the BR_ signal after the TS_ signal starts. The PowerSpan II arbiter could see this as a valid request and give the bus to the L2 cache when the bus was not requested. This bit delays when the PB arbiter samples the signal so a false bus request is not granted.

PARK: When set, the arbiter parks the address bus on the Processor Bus master programmed in the BM_PARK field. When cleared, the arbiter parks the address bus on the last Processor Bus master to be granted the bus.

BM_PARK: Identifies the master to be parked (see Table 123).

Table 123: Parked Processor Bus Master

BM_PARK [1:0]	Parked Processor Bus Master	External Pins
00	PowerSpan II	None
01	M1	PB_BR[1]_/PB_BG[1]_
10	M2	PB_BR[2]_/PB_BG[2]_
11	M3	PB_BR[3]_/PB_BG[3]_

13.5.33 DMA x Source Address Register

This register specifies the starting byte address on the source port for channel DMAx. The register is programmed for Direct mode DMA or updated by the Linked-list when loading the command packet.

Table 124: DMA x Source Address Register

Register Name: DMAx_SRC_ADDR	Register Offset: 0x304, 0x334, 0x 364, 0x394

PCI Bits	Function	PB Bits
31-24	SADDR	0-7
23-16	SADDR	8-15
15-08	SADDR	16-23
07-00	SADDR	24-31

DMAx_SRC_ADDR Description

Name	Type	Reset By	Reset State	Function
SADDR[31:0]	R/W	G_RST	0	Starting byte address on the source bus for the port defined by DMAx_TCR[SRC_PORT]

The DMAx_SRC_ADDR register is updated during the DMA transaction. Writing to this register while the DMA is active has no effect. While the DMA is active, this register provide status information on the progress of the transfer

13.5.34 DMA x Destination Address Register

This register specifies the starting byte address on the destination port for channel DMAx. This register is programmed for a Direct mode DMA or programmed by the Linked-list when loading the command packet.

Table 125: DMA x Destination Address Register

Register Name: DMAx_DST_ADDR	**Register Offset: 0x30C, 0x33C, 0x36C, 0x39C**

PCI Bits	Function				PB Bits
31-24	DADDR				0-7
23-16	DADDR				8-15
15-08	DADDR				16-23
07-00	DADDR	0	0	0	24-31

DMAx_DST_ADDR Description

Name	Type	Reset By	Reset State	Function
DADDR[31:3]	R/W	G_RST	0	Starting byte address on the destination bus for the port defined by DMAx_TCR[DST_PORT]

The DMAx_DST_ADDR register is updated during the DMA transaction. Writing to this register while the DMA is active has no effect. While the DMA is active, this register provides status information on the progress of the transfer.

DADDR: The lower three bits of the destination address is identical to the lower three bits of the source address (DMAx_SRC_ADDR).

13.5.35 DMA x Transfer Control Register

This register is used to specify parameters for channel DMAx. It is programmed directly for direct mode DMA or programmed by the Linked-list when loading the command packet.

Writing to this register while the DMA is active has no effect. While the DMA is active, this register provides status information on the progress of the transfer.

Table 126: DMA x Transfer Control Register

Register Name: DMAx_TCR	Register Offset: 0x314, 0x344, 0x374, 0x3A4

PCI Bits	Function				PB Bits
31-24	SRC_PORT	DST_PORT	END	0	0-7
23-16	BC				8-15
15-08	BC				16-23
07-00	BC				24-31

DMAx_TCR Description

Name	Type	Reset By	Reset State	Function
SRC_PORT [1:0]	R/W	G_RST	0	Source Port for DMA transfer 00 = PCI-1 01 = PCI-2 10 = PB 11 = reserved
				Single PCI PowerSpan II: 00 = PCI-1 10 = PB 01, 11 = reserved
DST_PORT [1:0]	R/W	G_RST	0	Destination Port for DMA transfer 00 = PCI-1 01 = PCI-2 10 = PB 11 = reserved

DMAx_TCR Description

Name	Type	Reset By	Reset State	Function
				Single PCI PowerSpan II: 00 = PCI-1 10 = PB 01, 11 = reserved
END[1:0]	R/W	P1_RST	10	Endian Conversion Mode 00 = Little-endian 01 = PowerPC little-endian 10 = Big-endian 11 = True little-endian
BC[23:0]	R/W	G_RST	0	Byte Count

END[1:0]: Selects the endian conversion mode for DMA activity involving the Processor Bus and a PCI Interface. When the source and destination ports are the same, then the conversion mode is little-endian, regardless of the value of this bit.

BC: When the initial value of the byte count is non-zero in Linked-List mode, the DMA starts with a Direct mode transfer. After the direct mode transfer has completed, the DMA channel begins processing the linked-list.

The field is updated during the DMA transaction.

13.5.36 DMA x Command Packet Pointer Register

This register specifies the 32-byte aligned address of the next command packet in the Linked-List for channel DMAx. It is programmed by PowerSpan II from the Linked-list when loading the command packet.

The DMAx_CPP register is updated at the start of a Linked-list transfer and remains constant throughout the transfer. Writing to this register while the DMA is active has no effect.

For a Direct mode DMA transfer, this register does not need to be programmed.

Table 127: DMA x Command Packet Pointer Register

Register Name: DMAx_CPP	Register Offset: 0x31C, 0x34C, 0x37C, 0x3AC

PCI Bits	Function			PB Bits
31-24	NCP			0-7
23-16	NCP			8-15
15-08	NCP			16-23
07-00	NCP	PowerSpan II Reserved	LAST	24-31

DMAx_CPP Description

Name	Type	Reset By	Reset State	Function
NCP[31:5]	R/W	G_RST	0	Next Command Packet Address. Points to a 32-byte aligned memory location of a linked-list on the port specified by the CP_PORT bit in the DMAx_ATTR register.
LAST	R/W	G_RST	0	Last Item 0 = more items in linked list 1 = last item in linked list

13.5.37 DMA x General Control and Status Register

This register contains general control and status information for channel DMA*x*. This register is not part of a linked-list Command Packet..

Writing to the CHAIN and OFF bits while the DMA is active has no effect.

Table 128: DMA x General Control and Status Register

Register Name: DMAx_GCSR	**Register Offset: 0x320, 0x350, 0x380, 0x3B0**

PCI Bits	**Function**								**PB Bits**
31-24	GO	CHAIN	0	0	0	STOP_REQ	HALT_REQ	0	0-7
23-16	DACT	DBS		DBS_EN		OFF			8-15
15-08	0	0	P1_ERR	P2_ERR	PB_ERR	STOP	HALT	DONE	16-23
07-00	0	0	P1_ERR_EN	P2_ERR_EN	PB_ERR_EN	STOP_EN	HALT_EN	DONE_EN	24-31

DMAx_GCSR Description

Name	**Type**	**Reset By**	**Reset State**	**Function**
GO	Write 1 to Set	G_RST	0	DMA Go bit 0 = no effect, 1 = Begin DMA transfer
CHAIN	R/W	G_RST	0	DMA Chaining 0 = DMA Direct mode 1 = DMA Linked-List mode
STOP_REQ	Write 1 to Set	G_RST	0	DMA Stop Request 0 = no effect 1 = Stop DMA when all buffered data has been written

DMAx_GCSR Description

Name	Type	Reset By	Reset State	Function
HALT_REQ	Write 1 to Set	G_RST	0	DMA Halt Request 0 = no effect 1 = Halt DMA at completion of current command packet
DACT	R	G_RST	0	DMA Active 0 = not active 1 = active
DBS[1:0]	R/W	G_RST	0	DMA Block Size (when DBS_EN is set to 1) 00=32 bytes 01=16 bytes 10=8 bytes 11=4 bytes
DBS_EN	R/W	G_RST	0	DMA Block Size Enable 0 = not active 1 = active
OFF	R/W	G_RST	0	DMA Channel Off Counter (number of PB clocks) 000 = 0 001 = 128 010 = 256 011 = 512 100 = 1024 101 = 2048 110 = 4096 111 = 8192
P1_ERR	R/Write 1 to Clear	G_RST	0	PCI-1 Bus Error 0 = no error 1 = error
P2_ERR	R/Write 1 to Clear	G_RST	0	PCI-2 Bus Error 0 = no error 1 = error
				Single PCI PowerSpan II: Reserved

DMAx_GCSR Description

Name	Type	Reset By	Reset State	Function
PB_ERR	R/Write 1 to Clear	G_RST	0	Processor Bus Error 0 = no error 1 = error
STOP	R/Write 1 to Clear	G_RST	0	DMA Stopped Flag 0 = not stopped 1 = stopped
HALT	R/Write 1 to Clear	G_RST	0	DMA Halted Flag 0 = not halted 1 = halted
DONE	R/Write 1 to Clear	G_RST	0	DMA Done Flag 0 = transfer not done 1 = transfer done
P1_ERR_EN	R/W	G_RST	0	Primary PCI Error Interrupt Enable 0 = no interrupt 1 = enable interrupt
P2_ERR_EN	R/W	G_RST	0	Normal PCI Error Interrupt Enable 0 = no interrupt 1 = enable interrupt
				Single PCI PowerSpan II: Reserved
PB_ERR_EN	R/W	G_RST	0	Processor Bus Error Interrupt Enable 0 = no interrupt 1 = enable interrupt
STOP_EN	R/W	G_RST	0	DMA Stop Interrupt Enable 0 = no interrupt 1 = enable interrupt
HALT_EN	R/W	G_RST	0	DMA Halt Interrupt Enable 0 = no interrupt 1 = enable interrupt
DONE_EN	R/W	G_RST	0	DMA Done Interrupt Enable 0 = no interrupt 1 = enable interrupt

DBS[1:0]: Controls the byte size of transactions generated by the DMA channel. The DBS_EN bit must be set to 1 in order for the DBS functionality to be enabled.

DBS_EN: Provides programmable control over the byte size of transactions generated by the DMA channel. The byte size is based on values programmed into DBS[1:0].

OFF: Provides programmable control over the amount of source bus traffic generated by the DMA channel. The channel will interleave source bus transfers with a period of idle Processor Bus clocks where no source bus requests are generated. When source and destination ports are different, 256 bytes of source bus traffic occur before the idle period. If source and destination ports are the same, 64 bytes of source bus traffic occur before the idle period. This helps prevent PowerSpan II from interfering with Processor Bus instruction fetches.

DONE: The DONE bit is set in the following cases:

- completion of Direct Mode DMA
- completion of Linked-List DMA

The DMA will not proceed until the DONE, and all other status bits, are cleared.

13.5.38 DMA x Attributes Register

This register contains additional parameters for DMA channel x. It is not part of a Linked-List Command Packet.

Table 129: DMA x Attributes Register

Register Name: DMAx_ATTR	**Register Offset: 0x324, 0x354, 0x384, 0x3B4**

PCI Bits	Function					PB Bits
31-24	CP_PORT	0	GBL_	CI_	PowerSpan II Reserved	0-7
23-16	PowerSpan II Reserved		RTT			8-15
15-08	PowerSpan II Reserved		WTT			16-23
07-00	PowerSpan II Reserved					24-31

DMAx_ATTR Description

Name	Type	Reset By	Reset State	Function
CP_PORT [1:0]	R/W	G_RST	0	Command Packet Port 00 = PCI-1 01 = PCI-2, 10 = PB 11 = reserved
				Single PCI PowerSpan II: 00 = PCI-1 10 = PB 01, 11 = reserved
PB_GBL_	R/W	G_RST	0	Processor Bus Global 0 = Assert PB_GBL_ 1 = Negate PB_GBL_
PB_CI_	R/W	G_RST	0	Processor Bus Cache Inhibit 0 = Assert PB_CI_ 1 = Negate PB_CI_

DMAx_ATTR Description

Name	Type	Reset By	Reset State	Function
RTT[4:0]	R/W	G_RST	01010	Processor Bus Read Transfer Type PB_TT[0:4] 01010 = Read
WTT[4:0]	R/W	G_RST	00010	Processor Bus Write Transfer Type PB_TT[0:4] 00010 = Write with flush

RTT/WTT: Selects the Transfer Type on the Processor Bus. The register bits RTT[4:0]/WTT[4:0] are mapped to pins PB_TT[0:4].

13.5.39 Miscellaneous Control and Status Register

Table 130: Miscellaneous Control and Status Register

Register Name: MISC_CSR	Register Offset: 0x400

PCI Bits	Function								PB Bits
31-24	TUNDRA_DEV_ID								0-7
23-16	TUNDRA_VER_ID								8-15
15-08	VPD_EN	VPD_CS			BAR_EQ_0	Reserved	ELOAD_OPT		16-23
07-00	P1_LOCKOUT	P2_LOCKOUT	PowerSpan II Reserved		PCI_ARB_CFG	PCI_M7	PCI_M6	PCI_M5	24-31

MISC_CSR Description

Name	Type	Reset By	Reset State	Function
TUNDRA_DEV_ID[7:0]	R	G_RST	0x00	Tundra Internal Device ID
			0x01	Single PCI PowerSpan II
TUNDRA_VER_ID[7:0]	R	G_RST	0x02	Tundra Internal Version ID PowerSpan II = 02 (Original PowerSpan = 01)
VPD_EN	R/W	G_RST	0 EEPROM	PCI Vital Product Data. 0=disabled 1=enabled
VPD_CS[2:0]	R/W	G_RST	0 EEPROM	PCI Vital Product Data EEPROM Chip Select.
BAR_EQ_0	R/W	G_RST	0 EEPROM	Base Address Equivalent to 0x00000
ELOAD_OPT[1:0]	R	G_RST	0 EEPROM	EEPROM load option 00=do not load 01=short load 10=long load 11=reserved

MISC_CSR Description

Name	Type	Reset By	Reset State	Function
P1_LOCKOUT	R/Write 1 to Clear	G_RST	1 EEPROM	PCI-1 lockout 0=not set 1=set
P2_LOCKOUT	R/Write 1 to Clear	G_RST	1 EEPROM	PCI-2 lockout 0=not set 1=set
				Single PCI PowerSpan II: Reserved
PCI_ARB_CFG	Write 1 to set	G_RST	0 EEPROM	PCI Arbiter Pins Configured 0=Floating PCI Arbiter pins not yet configured 1=Floating PCI Arbiter pins configured
				Single PCI PowerSpan II: Reserved
PCI_M7	R/W	G_RST	0 EEPROM	PCI Arbiter Master 7 0=PowerSpan II PCI-1 arbiter 1=PowerSpan II PCI-2 arbiter
				Single PCI PowerSpan II: Reserved
PCI_M6	R/W	G_RST	0 EEPROM	PCI Arbiter Master 6 0=PowerSpan II PCI-1 arbiter 1=PowerSpan II PCI-2 arbiter
				Single PCI PowerSpan II: Reserved
PCI_M5	R/W	G_RST	0 EEPROM	PCI Arbiter Master 5 0=PowerSpan II PCI-1 arbiter 1=PowerSpan II PCI-2 arbiter
				Single PCI PowerSpan II: Reserved

VPD_EN: Enables PCI Vital Product Data (VPD) as described in the “I2C/EEPROM” on page 153. When enabled, the VPD registers in the PCI Interface that has been designated as primary are used to access PCI Vital Product Data.

BAR_EQ_0: This bit enables a value of 0x00000 for Px Base Address registers.

ELOAD_OPT: Identifies the load option selected in the first byte of the power-up EEPROM.

Px_LOCKOUT: When set, all configuration and memory register space accesses from PCI are retried. The Px_LOCKOUT bit must be cleared for all memory space accesses to the PowerSpan II's PCI target images.

⚠ PowerSpan II does not terminate the cycle when the Px_LOCKOUT bit is not cleared during a memory space access to the PCI target images. If PowerSpan II does not terminate the cycle, the PCI bus experiences a deadlock condition.

The Px_LOCKOUT bit must be cleared before the corresponding PCI Target Image claims a transaction. The bit is cleared by an agent on the Processor Bus or by EEPROM load. The bit is cleared automatically by PowerSpan II when the PWRUP_BOOT option is set to PCI.

PCI_ARB_CFG: When set, this bit enables recognition of external master requests on PCI_REQ#[7:5]. The user must set this bit after completing configuration all of the PowerSpan II floating PCI arbitration pins with bits PCI_M7, PCI_M6 and PCI_M5. When PCI_ARB_CFG is not set, requests from external masters connected to PCI_REQ#[7:5] are ignored.

Initialization of PCI_ARB_CFG is not required for the Single PCI PowerSpan II because PCI_REQ#[7:5]/PCI_GNT#[7:5] are dedicated to the PCI-1 Interface.

PCI_Mx: Each of these PCI Master bits must be explicitly initialized by the user to indicate which PowerSpan II PCI arbiter should service the pair of PCI_REQ#/PCI_GNT# pins. Initialization occurs through EEPROM load or a register write.

Table 131 indicates register bit to arbitration pin mappings:

Table 131: Arbitration Pin Mappings

Bit	Arbitration Pins
PCI_M5	PCI_REQ#[5]/PCI_GNT#[5]
PCI_M6	PCI_REQ#[6]/PCI_GNT#[6]
PCI_M7	PCI_REQ#[7]/PCI_GNT#[7]

The PCI_Mx bits do not affect the behavior of the Single PCI PowerSpan II because PCI_REQ#[7:5]#/PCI_GNT#[7:5] are dedicated to the PCI-1 Interface.

13.5.40 Clock Control Register

PowerSpan II does not use the TUNE bits for adjusting the PLL parameters. This register does not effect PLL performance.

This register does not effect the functionality or performance of PowerSpan II. This register makes the device backwards compatible with the PowerSpan II device.

Table 132: Clock Control Register

Register Name: CLOCK_CTL	**Register Offset: 0x404**

PCI Bits	Function	PB Bits
31-24	PB_TUNE	0-7
23-16	P1_TUNE	8-15
15-08	P2_TUNE	16-23
07-00	PowerSpan II Reserved	24-31

CLOCK_CTL Description

Name	Type	Reset By	Reset State	Function
PB_TUNE[7:0]	R	G_RST	(See Below) EEPROM	PB PLL Tune Bits
P1_TUNE[7:0]	R	G_RST	(See Below) EEPROM	PCI-1 PLL Tune Bits
P2_TUNE[7:0]	R	G_RST	(See Below) EEPROM	PCI-2 PLL Tune Bits

PB_TUNE: Tune bits for the Processor Bus PLL. The reset value is a function of the system level applied to the PB_FAST external pin. The reset values are:

- PB_TUNE[7:2] = 000100
- PB_TUNE[1] = ~PB_FAST
- PB_TUNE[0] = 1

P1_TUNE: Tune bits for the PCI-1 PLL. The reset value is a function of the system level applied to the P1_M66EN external pin. The reset values are:

- P1_TUNE[7:1] = 0001001
- P1_TUNE[0] = ~P1_M66EN

P2_TUNE: Tune bits for the PCI-2 PLL. The reset value is a function of the system level applied to the P2_M66EN external pin. The reset values are:

- P2_TUNE[7:1] = 0001001
- P2_TUNE[0] = ~P2_M66EN

13.5.41 I^{2}C/EEPROM Interface Control and Status Register

This register supports the PowerSpan II I^{2}C/EEPROM interface.

Table 133: I^{2}C/EEPROM Interface Control and Status Register

Register Name: I2C_CSR	Register Offset: 0x408

<table>
<tr><th>PCI
Bits</th><th colspan="5">Function</th><th>PB
Bits</th></tr>
<tr><td>31-24</td><td colspan="5">ADDR</td><td>0-7</td></tr>
<tr><td>23-16</td><td colspan="5">DATA</td><td>8-15</td></tr>
<tr><td>15-08</td><td colspan="3">DEV_CODE</td><td>CS</td><td>RW</td><td>16-23</td></tr>
<tr><td>07-00</td><td>ACT</td><td>ERR</td><td colspan="3">PowerSpan II Reserved</td><td>24-31</td></tr>
</table>

I2C_CSR Description

Name	Type	Reset By	Reset State	Function
ADDR[7:0]	R/W	G_RST	0	Specifies I^{2}C device address to be accessed.
DATA[7:0]	R/W	G_RST	0	Specifies the required data for a write. Holds the data at the end of a read.
DEV_CODE[3:0]	R/W	G_RST	1010	Device select. I^{2}C 4-bit device code.
CS[2:0]	R/W	G_RST	0	Chip Select
RW	R/W	G_RST	0	0=read 1=write
ACT	R	G_RST	0	I^{2}C interface active 0=not active 1=active
ERR	R/Write 1 to Clear	G_RST	0	Error 0=no error 1=error condition

The user initiates a I^{2}C bus cycle by writing to this register. Software must wait for the ACT bit to be zero before starting a new I^{2}C cycle. When the ACT bit is 1, writes to this register have no effect and the DATA field is undefined.

The ACT bit is set under of the following conditions:

- I^{2}C interface is busy servicing a read or write as a result of a write to this register
- I^{2}C interface is busy loading registers at the end of reset
- I^{2}C interface is busy accessing PCI Vital Product Data

The PCI VPD EEPROM Chip Select (VPD_CS) bit, in the MISC_CSR, selects the EEPROM where VPD resides. If VPD_CS is 000b, then VPD starts at address offset 0x40 of the first EEPROM. For all other values of VPD_CS, VPD starts at address offset 0x00 of the specified EEPROM.

ERR: The register updates the ERR bit five clock cycles after a clean transaction has completed.

13.5.42 Reset Control and Status Register

This register contains the read-only bits that specify all PowerSpan II power-up options and status of a number of pins that are normally fixed for each application.

Table 134: Reset Control and Status Register

Register Name: RST_CSR	**Register Offset: 0x40C**

PCI Bits	**Function**								**PB Bits**
31-24	PB_RST_DIR	PB_ARB_EN	PB_FAST	PCI_BOOT	PowerSpan II Reserved				0-7
23-16	P1_RST_DIR	P1_ARB_EN	P1_M66EN	PowerSpan II Reserved			P1_R64_EN	P1_D64	8-15
15-08	P2_RST_DIR	P2_ARB_EN	P2_M66EN	PowerSpan II Reserved				PRI_PCI	16-23
07-00	PowerSpan II Rsvd	7400_MODE	BYPASS_EN	ELOAD	PowerSpan II Reserved				24-31

RST_CSR Description

Name	**Type**	**Reset By**	**Reset State**	**Function**
PB_RST_DIR	R	G_RST	PWRUP	Status of PB_RST_DIR pin.
PB_ARB_EN	R	G_RST	PWRUP	Processor bus arbiter enable. 0=Disabled power-up option 1=Enabled power-up option
PB_FAST	R	G_RST	PWRUP	Processor Bus Clock Frequency Selection 0=25 MHz to 50 MHz 1=50 MHz to 100 MHz
PCI_BOOT	R	G_RST	PWRUP	PCI Boot 0=Boot from Processor Bus 1=Boot from PCI
P1_RST_DIR	R	G_RST	PWRUP	Status of P1_RST_DIR pin.

RST_CSR Description

		Reset	Reset	
Name	**Type**	**By**	**State**	**Function**
P1_ARB_EN	R	G_RST	PWRUP	PCI-1 arbiter enable 0=Disabled power-up option 1=Enabled power-up option
P1_M66EN	R	G_RST	PWRUP	PCI-1 Clock Frequency Selection 0=25 MHz to 33 MHz 1=33 MHz to 66 MHz
P1_R64_EN	R	G_RST	PWRUP	P1_REQ64# output enable. 0=PowerSpan II does not assert P1_REQ64# at reset 1=PowerSpan II does assert P1_REQ64# at reset to indicate the presence of a 64-bit P1_AD[] bus
P1_D64	R	G_RST	PWRUP	PCI-1 Databus Width 0=connected to 32-bit AD bus 1=connected to 64-bit AD bus See Table 3 on page 46
P2_RST_DIR	R	G_RST	PWRUP	Status of P2_RST_DIR pin.
				Single PCI PowerSpan II: Reserved
P2_ARB_EN	R	G_RST	PWRUP	PCI-2 arbiter enable. 0=Disabled power-up option 1=Enabled power-up option
				Single PCI PowerSpan II: Reserved
P2_M66EN	R	G_RST	PWRUP	PCI-2 Clock Frequency Selection 0=25 MHz to 50 MHz 1=33 MHz to 66 MHz
				Single PCI PowerSpan II: Reserved
PRI_PCI	R	G_RST	PWRUP	Designated Primary PCI Bus. 0=PCI-1 is Primary 1=PCI-2 is Primary
				Single PCI PowerSpan II: Reserved

RST_CSR Description

Name	Type	Reset By	Reset State	Function
7400_MODE	R	G_RST	PWRUP	7400 Mode Enable 0=Disabled power-up option 1=Enabled power-up option
BYPASS_EN	R	G_RST	PWRUP	Phase Locked Loop Bypass Enable 0=Disabled power-up option 1=Enabled power-up option
ELOAD	R	G_RST	PWRUP	EEPROM load after reset. 0=EEPROM load not enabled 1=EEPROM load enabled

P1_D64: Indicates the width of the databus to which the PCI-1 Interface is connected. This is determined by the level on P1_REQ64# at the negation of P1_RST#, or by the level on P1_64EN#.

PB_FAST: Indicates the latched value of the PB_FAST pin. This bit is used to optimally configure the Processor Bus Interface PLL for the desired operating frequency.

P1_M66EN: Indicates the latched value of the P1_M66EN pin. This bit is used to optimally configure the PCI-1 interface PLL for the desired operating frequency.

P2_M66EN: Indicates the latched value of the P2_M66EN pin. This bit is used to optimally configure the PCI-2 interface PLL for the desired operating frequency.

7400_MODE: When enabled, the PB arbiter qualifies bus grants before issuing a grant to a PB Master. When disabled, the PB arbiter issues a grant to a PB Master and it is expected that the PB Master receiving the grant qualifies the grant.

BYPASS_EN: Indicates the setting of this power-up option. If this bit is set, the user has elected to bypass all PowerSpan II PLL’s. This bit supports slow speed emulation of a PowerSpan II based system.

13.5.43 Interrupt Status Register 0

This register is one of two interrupt status registers. ISR0 is used primarily for normal operating status.

When set, each bit of this register indicates the corresponding interrupt source is active.

Table 135: Interrupt Status Register 0

Register Name: ISR0	Register Offset: 0x410

PCI Bits	Function								PB Bits
31-24	ISR1_ACTV	0	I2O_HOST	I2O_IOP	DMA3	DMA2	DMA1	DMA0	0-7
23-16	P2_HW	P1_HW	INT5_HW	INT4_HW	INT3_HW	INT2_HW	INT1_HW	INT0_HW	8-15
15-08	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	16-23
07-00	MBOX7	MBOX6	MBOX5	MBOX4	MBOX3	MBOX2	MBOX1	MBOX0	24-31

ISR0 Description

Name	Type	Reset By	Reset State	Function
ISR1_ACTV	R	G_RST	0	Indicates an interrupt status bit is set in ISR1 register.
I2O_HOST	R	G_RST	0	Interrupt asserted to the I2O Host to indicate that the Outbound Post List FIFO contains MFAs of messages for the Host to process.
I2O_IOP	R/Write 1 to Clear	G_RST	0	Interrupt to embedded PowerPC to indicate that the Inbound Post List FIFO contains MFAs of messages for the embedded PowerPC to process.
DMAx	R/Write 1 to Clear	G_RST	0	Set when DMAx generates an interrupt. See DMAx_GCSR register for details.
P1_HW	R/Write 1 to Clear	G_RST	0	PCI-1 hardware interrupt. Set when a level interrupt is detected on the PCI-1 INTA# pin.
P2_HW	R/Write 1 to Clear	G_RST	0	PCI-2 hardware interrupt. Set when a level interrupt is detected on the PCI-2 INTA# pin.

ISR0 Description

Name	Type	Reset By	Reset State	Function
				Single PCI PowerSpan II: Reserved
INT0_HW	R/Write 1 to Clear	G_RST	0	Hardware interrupt. Set when a level interrupt is detected on the INT[0]_ pin.
INT1_HW	R/Write 1 to Clear	G_RST	0	Hardware interrupt. Set when a level interrupt is detected on the INT[1]_ pin.
INT2_HW	R/Write 1 to Clear	G_RST	0	Hardware interrupt. Set when a level interrupt is detected on the INT[2]_ pin.
INT3_HW	R/Write 1 to Clear	G_RST	0	Hardware interrupt. Set when a level interrupt is detected on the INT[3]_ pin.
INT4_HW	R/Write 1 to Clear	G_RST	0	Hardware interrupt. Set when a level interrupt is detected on the INT[4]_ pin.
INT5_HW	R/Write 1 to Clear	G_RST	0	Hardware interrupt. Set when a level interrupt is detected on the INT[5]_ pin.
DB7-DB0	R/Write 1 to Clear	G_RST	0	Set when a doorbell register is written to in the IER register.
MBOX[7:0]	R/Write 1 to Clear	G_RST	0	Set when a mailbox is written to.

ISR1_ACTV: This bit is a logical OR of all the status bits in the ISR1 register. When any status bit in ISR1 is set, ISR1_ACTV is set. When all bits of the ISR1 register are cleared, ISR1_ACTV is cleared. This bit is useful in determining whether or not to read the ISR1 register to determine the source of the interrupt.

I2O_HOST: This bit is an alias for the I2O Outbound Post List Status Register located at offset 0x030 of the I2O Target Image.

13.5.44 Interrupt Status Register 1

This register is one of two interrupt status registers. ISR1 is organized with error conditions in PowerSpan II.

Table 136: Interrupt Status Register 1

Register Name: ISR1	Register Offset: 0x414

PCI Bits	Function								PB Bits
31-24	ISR0_ACTV	PowerSpan II Reserved			PB_P1_RETRY	PB_P2_RETRY	PB_PB_RETRY	0	0-7
23-16	PB_P1_ERR	PB_P2_ERR	PB_PB_ERR	PB_A_PAR	PB_P1_D_PAR	PB_P2_D_PAR	PB_PB_D_PAR	0	8-15
15-08	P2_P1_ERR	P2_PB_ERR	P2_P2_ERR	P2_A_PAR	P2_P1_RETRY	P2_PB_RETRY	P2_P2_RETRY	0	16-23
07-00	P1_P2_ERR	P1_PB_ERR	P1_P1_ERR	P1_A_PAR	P1_P2_RETRY	P1_PB_RETRY	P1_P1_RETRY	0	24-31

ISR1 Description

Name	Type	Reset By	Reset State	Function
ISR0_ACTV	R	G_RST	0	Indicates an interrupt status bit is set in ISR0 register.
PB_P1_RETRY	R/Write 1 to Clear	G_RST	0	Processor Bus Max Retry Error. Maximum number of retries detected. The cycle was initiated/ destined to the PCI 1 bus.
PB_P2_RETRY	R/Write 1 to Clear	G_RST	0	Processor Bus Max Retry Error. Maximum number of retries detected. The cycle was initiated/ destined to the PCI-2 bus.
				Single PCI PowerSpan II: Reserved
PB_PB_RETRY	R/Write 1 to Clear	G_RST	0	Processor Bus Max Retry Error. Maximum number of retries detected during Processor Bus to Processor Bus DMA.
PB_P1_ERR	R/Write 1 to Clear	G_RST	0	Processor Bus interface asserted/received PB_TEA_. The cycle was initiated/destined to the PCI-1 bus.

ISR1 Description

Name	Type	Reset By	Reset State	Function
PB_P2_ERR	R/Write 1 to Clear	G_RST	0	Processor Bus interface asserted/received PB_TEA_. The cycle was initiated/destined to the PCI-2 bus.
				Single PCI PowerSpan II: Reserved
PB_PB_ERR	R/Write 1 to Clear	G_RST	0	Processor Bus interface asserted/received PB_TEA_ during Processor Bus to Processor Bus DMA.
PB_A_PAR	R/Write 1 to Clear	G_RST	0	Processor Bus Address Parity Error detected.
PB_P1_D_PAR	R/Write 1 to Clear	G_RST	0	Processor Bus Data Parity Error detected. The cycle was initiated/destined to the PCI 1 bus.
PB_P2_D_PAR	R/Write 1 to Clear	G_RST	0	Processor Bus Data Parity Error detected. The cycle was initiated/destined to the PCI-2 bus.
				Single PCI PowerSpan II: Reserved
PB_PB_D_PAR	R/Write 1 to Clear	G_RST	0	Processor Bus Data Parity Error detected during Processor Bus to Processor Bus DMA.
P2_P1_ERR	R/Write 1 to Clear	G_RST	0	PCI-2 interface detected an error. The P2_CSR error bits must be checked for the source of the error. The cycle was initiated/destined to the PCI 1 bus.
				Single PCI PowerSpan II: Reserved
P2_PB_ERR	R/Write 1 to Clear	G_RST	0	PCI-2 interface detected an error. The P2_CSR error bits must be checked for the source of the error. The cycle was initiated/destined to the Processor Bus.
				Single PCI PowerSpan II: Reserved
P2_P2_ERR	R/Write 1 to Clear	G_RST	0	PCI-2 interface detected an error during P2 to P2 DMA.
				2P: Reserved
P2_A_PAR	R/Write 1 to Clear	G_RST	0	PCI-2 interface detected an address parity error.
				2P: Reserved

ISR1 Description

Name	Type	Reset By	Reset State	Function
P2_P1_RETRY	R/Write 1 to Clear	G_RST	0	PCI-2 Master received too many retries. The cycle was initiated from the PCI 1 bus.
				2P: Reserved
P2_PB_RETRY	R/Write 1 to Clear	G_RST	0	PCI-2 Master received too many retries. The cycle was initiated from the Processor Bus.
				2P: Reserved
P2_P2_RETRY	R/Write 1 to Clear	G_RST	0	PCI-2 Master received too many retries during P2 to P2 DMA.
				2P: Reserved
P1_P2_ERR	R/Write 1 to Clear	G_RST	0	PCI-1 interface detected an error. The P1_CSR error bits must be checked for the source of the error. The cycle was initiated/destined to the PCI-2 bus.
				2P: Reserved
P1_PB_ERR	R/Write 1 to Clear	G_RST	0	PCI-1 interface detected an error. The P1_CSR error bits must be checked for the source of the error. The cycle was initiated/destined to the Processor Bus.
P1_P1_ERR	R/Write 1 to Clear	G_RST	0	PCI-1 interface detected an error during P1 to P1 DMA.
P1_A_PAR	R/Write 1 to Clear	G_RST	0	PCI-1 interface detected an address parity error.
P1_P2_RETRY	R/Write 1 to Clear	G_RST	0	PCI-1 Master received too many retries. The cycle was initiated from the PCI-2 bus.
				2P: Reserved
P1_PB_RETRY	R/Write 1 to Clear	G_RST	0	PCI-1 Master received too many retries. The cycle was initiated from the Processor Bus.
P1_P1_RETRY	R/Write 1 to Clear	G_RST	0	PCI-1 Master received too many retries during PCI-1 to PCI-1 DMA.

ISR0_ACTV: This bit is a logical OR of all the status bits in the ISR0 register. If any register is set, ISR0_ACTV is set. When all bits of the ISR0 register are cleared, ISR0_ACTV is cleared.

13.5.45 Interrupt Enable Register 0

Each bit, when set, allows the corresponding active status bit in ISR0 to generate an interrupt on an external pin. The external pin is determined by the Interrupt Mapping Registers and the Interrupt Direction Register.

Table 137: Interrupt Enable Register 0

Register Name: IER0	Register Offset: 0x418

PCI Bits	Function								PB Bits
31-24	PowerSpan II Reserved		I2O_HOST_MASK	I2O_IOP_EN	DMA3_EN	DMA2_EN	DMA1_EN	DMA0_EN	0-7
23-16	P2_HW_EN	P1_HW_EN	INT5_HW_EN	INT4_HW_EN	INT3_HW_EN	INT2_HW_EN	INT1_HW_EN	INT0_HW_EN	8-15
15-08	DB7_EN	DB6_EN	DB5_EN	DB4_EN	DB3_EN	DB2_EN	DB1_EN	DB0_EN	16-23
07-00	MBOX7_EN	MBOX6_EN	MBOX5_EN	MBOX4_EN	MBOX3_EN	MBOX2_EN	MBOX1_EN	MBOX0_EN	24-31

IER0 Description

Name	Type	Reset By	Reset State	Function
I2O_HOST_MASK	R/W	G_RST	0	I2O_HOST interrupt mask 0=interrupt enabled, 1=interrupt masked
I2O_IOP_EN	R/W	G_RST	0	I2O_IOP interrupt enable
DMAx_EN	R/W	G_RST	0	DMAx interrupt enable
P1_HW_EN	R/W	G_RST	0	PCI 1 hardware interrupt enable
P2_HW_EN	R/W	G_RST	0	PCI-2 hardware interrupt enable
				2P: Reserved
INT0_HW_EN	R/W	G_RST	0	INT[0]_ hardware interrupt enable
INT1_HW_EN	R/W	G_RST	0	INT[1]_ hardware interrupt enable
INT2_HW_EN	R/W	G_RST	0	INT[2]_ hardware interrupt enable
INT3_HW_EN	R/W	G_RST	0	INT[3]_ hardware interrupt enable

IER0 Description

Name	Type	Reset By	Reset State	Function
INT4_HW_EN	R/W	G_RST	0	INT[4]_ hardware interrupt enable
INT5_HW_EN	R/W	G_RST	0	INT[5]_ hardware interrupt enable
DBx_EN	Write 1 to set	G_RST	0	Writing a one to this register sets the doorbell register in the ISR0 register.
MBOXx_EN	R/W	G_RST	0	Mailbox interrupt enable

I2O_HOST_MASK: This bit is an alias for the I2O register OPL_IM[OP_ISM] used to mask interrupts associated with the I2O Outbound Queue.

The Doorbell registers generate an interrupt when writing 1 to the register. This causes the corresponding doorbell bit in the ISR0 register to be set. In order to clear the doorbell interrupt, the ISR0 status bit must be cleared.

13.5.46 Interrupt Enable Register 1

Each bit, when set, allows the corresponding active status bit in ISR1 to generate an interrupt on an external pin. The external pin is determined by the Interrupt Mapping Registers and the Interrupt Direction Register.

Table 138: Interrupt Enable Register 1

Register Name: IER1	Register Offset: 0x41C

PCI Bits	Function								PB Bits
31-24	PowerSpan II Reserved				PB_P1_RETRY_EN	PB_P2_RETRY_EN	PB_PB_RETRY_EN	0	0-7
23-16	PB_P1_ERR_EN	PB_P2_ERR_EN	PB_PB_ERR_EN	PB_A_PAR_EN	PB_P1_D_PAR_EN	PB_P2_D_PAR_EN	PB_PB_D_PAR_EN	0	8-15
15-08	P2_P1_ERR_EN	P2_PB_ERR_EN	P2_P2_ERR_EN	P2_A_PAR_EN	P2_P1_RETRY_EN	P2_PB_RETRY_EN	P2_P2_RETRY_EN	0	16-23
07-00	P1_P2_ERR_N	P1_PB_ERR_EN	P1_P1_ERR_EN	P1_A_PAR_EN	P1_P2_RETRY_EN	P1_PB_RETRY_EN	P1_P1_RETRY_EN	0	24-31

IER1 Description

Name	Type	Reset By	Reset State	Function
PB_P1_RETRY_EN	R/W	G_RST	0	Processor Bus Max Retry Counter enable. The cycle was initiated/destined to the PCI-1 bus.
PB_P2_ RETRY_EN	R/W	G_RST	0	Processor Bus Max Retry Error enable. The cycle was initiated/destined to the PCI-2 bus.
				2P: Reserved
PB_PB_RETRY	R/W	G_RST	0	Processor Bus Max Retry Counter enable. Processor Bus to Processor Bus DMA.
PB_P1_ERR_EN	R/W	G_RST	0	Processor Bus Error Enable. The cycle was initiated/destined to the PCI-1 bus.
PB_P2_ERR_EN	R/W	G_RST	0	Processor Bus Error enable. The cycle was initiated/destined to the PCI-2 bus.

IER1 Description

Name	Type	Reset By	Reset State	Function
				2P: Reserved
PB_PB_ERR_EN	R/W	G_RST	0	Processor Bus Error enable. Processor Bus to Processor Bus DMA.
PB_A_PAR_EN	R/W	G_RST	0	Processor Bus Address Parity Error enable
PB_P1_D_PAR_EN	R/W	G_RST	0	Processor Bus Data Parity Error enable. The cycle was initiated/destined to the PCI-1 bus.
PB_P2_D_PAR_EN	R/W	G_RST	0	Processor Bus Data Parity Error enable. The cycle was initiated/destined to the PCI-2 bus.
				2P: Reserved
PB_PB_D_PAR_EN	R/W	G_RST	0	Processor Bus Data Parity Error enable. Processor Bus to Processor Bus DMA.
P2_P1_ERR_EN	R/W	G_RST	0	PCI-2 error enable. The cycle was initiated/ destined to the PCI 1 bus.
				2P: Reserved
P2_PB_ERR_EN	R/W	G_RST	0	PCI-2 error enable. The cycle was initiated/ destined to the Processor Bus.
				2P: Reserved
P2_P2_ERR_EN	R/W	G_RST	0	PCI-2 error enable.PCI-2 to PCI-2 DMA.
				2P: Reserved
P2_A_PAR_EN	R/W	G_RST	0	PCI-2 address parity error enable.
				2P: Reserved
P2_P1_RETRY_EN	R/W	G_RST	0	PCI-2 max retry enable. The cycle was initiated/ destined to the PCI-1 bus.
				2P: Reserved
P2_PB_RETRY_EN	R/W	G_RST	0	PCI-2 max retry enable. The cycle was initiated/ destined to the Processor Bus.
				2P: Reserved

IER1 Description

Name	Type	Reset By	Reset State	Function
P2_P2_ RETRY_EN	R/W	G_RST	0	PCI-2 max retry enable. PCI-2 to PCI-2 DMA.
				2P: Reserved
P1_P2_ERR_ EN	R/W	G_RST	0	PCI-1 error enable. The cycle was initiated/ destined to the PCI-2 bus.
				2P: Reserved
P1_PB_ERR_ EN	R/W	G_RST	0	PCI-1 error enable. The cycle was initiated/ destined to the Processor Bus.
P1_P1_ERR_ EN	R/W	G_RST	0	PCI-1 error enable. PCI-1 to PCI-1 DMA.
P1_A_PAR_E N	R/W	G_RST	0	PCI-1 address parity error enable.
P1_P2_ RETRY_EN	R/W	G_RST	0	PCI-1 max retry enable. The cycle was initiated/ destined to the PCI-2 bus.
				2P: Reserved
P1_PB_RET RY_EN	R/W	G_RST	0	PCI-1 max retry enable. The cycle was initiated/ destined to the Processor Bus.
P1_P1_RETR Y_EN	R/W	G_RST	0	PCI-1 max retry enable. PCI-1 to PCI-1 DMA.

13.5.47 Interrupt Map Register Mail Box

Each map field assigns an interrupt output pin to the corresponding interrupt source.

Table 139: Interrupt Map Register Mail Box

Register Name: IMR_MBOX	Register Offset: 0x420

PCI Bits	Function				PB Bits
31-24	MBOX7_MAP	0	MBOX6_MAP	0	0-7
23-16	MBOX5_MAP	0	MBOX4_MAP	0	8-15
15-08	MBOX3_MAP	0	MBOX2_MAP	0	16-23
07-00	MBOX1_MAP	0	MBOX0_MAP	0	24-31

IMR_MBOX Description

Name	Type	Reset By	Reset State	Function
MBOX7_MAP [2:0]	R/W	G_RST	0	Map Mailbox #7 to an interrupt pin.
MBOX6_MAP [2:0]	R/W	G_RST	0	Map Mailbox #6 to an interrupt pin.
MBOX5_MAP [2:0]	R/W	G_RST	0	Map Mailbox #5 to an interrupt pin.
MBOX4_MAP [2:0]	R/W	G_RST	0	Map Mailbox #4 to an interrupt pin.
MBOX3_MAP [2:0]	R/W	G_RST	0	Map Mailbox #3 to an interrupt pin.
MBOX2_MAP [2:0]	R/W	G_RST	0	Map Mailbox #2 to an interrupt pin.
MBOX1_MAP [2:0]	R/W	G_RST	0	Map Mailbox #1 to an interrupt pin.
MBOX0_MAP [2:0]	R/W	G_RST	0	Map Mailbox #0 to an interrupt pin.

Table 140 describes the mapping of interrupt sources to the external interrupt pins. The shaded entries indicate unsupported combinations for the Single PCI PowerSpan II.

Table 140: Mapping Definition

Map Field	Interrupt Pin
000	P1_INTA#
001	P2_INTA#
010	INT[0]_
011	INT[1]_
100	INT[2]_
101	INT[3]_
110	INT[4]_
111	INT[5_

13.5.48 Interrupt Map Register Doorbell

Each map field assigns an interrupt output pin to the corresponding interrupt source. Table 140 on page 399 defines the mapping definitions.

Table 141: Interrupt Map Register Doorbell

Register Name: IMR_DB	Register Offset: 0x424

PCI Bits	Function				PB Bits
31-24	DB7_MAP	0	DB6_MAP	0	0-7
23-16	DB5_MAP	0	DB4_MAP	0	8-15
15-08	DB3_MAP	0	DB2_MAP	0	16-23
07-00	DB1_MAP	0	DB0_MAP	0	24-31

IMR_DB Description

Name	Type	Reset By	Reset State	Function
DB7_MAP[2:0]	R/W	G_RST	0	Map doorbell #7 to an interrupt pin
DB6_MAP[2:0]	R/W	G_RST	0	Map doorbell #6 to an interrupt pin
DB5_MAP[2:0]	R/W	G_RST	0	Map doorbell #5 to an interrupt pin
DB4_MAP[2:0]	R/W	G_RST	0	Map doorbell #4 to an interrupt pin
DB3_MAP[2:0]	R/W	G_RST	0	Map doorbell #3 to an interrupt pin
DB2_MAP[2:0]	R/W	G_RST	0	Map doorbell #2 to an interrupt pin
DB1_MAP[2:0]	R/W	G_RST	0	Map doorbell #1 to an interrupt pin
DB0_MAP[2:0]	R/W	G_RST	0	Map doorbell #0 to an interrupt pin

13.5.49 Interrupt Map Register DMA

Each map field assigns an interrupt output pin to the corresponding interrupt source. Table 140 on page 399 defines the mapping definitions.

Table 142: Interrupt Map Register DMA

Register Name: IMR_DMA	Register Offset: 0x428

PCI Bits	Function				PB Bits
31-24	PowerSpan II Reserved				0-7
23-16	PowerSpan II Reserved				8-15
15-08	DMA3_MAP	0	DMA2_MAP	0	16-23
07-00	DMA1_MAP	0	DMA0_MAP	0	24-31

IMR_DMA_TIMER Description

Name	Type	Reset By	Reset State	Function
DMA3_MAP[2:0]	R/W	G_RST	0	Map DMA #3 to an interrupt pin
DMA2_MAP[2:0]	R/W	G_RST	0	Map DMA #2 to an interrupt pin
DMA1_MAP[2:0]	R/W	G_RST	0	Map DMA #1 to an interrupt pin
DMA0_MAP[2:0]	R/W	G_RST	0	Map DMA #0 to an interrupt pin

13.5.50 Interrupt Map Register Hardware

This register assigns an interrupt output pin to the corresponding interrupt source. All sources are associated with errors detected by the PCI-1 Interface.

Each map field assigns an interrupt output pin to the corresponding interrupt source. Table 140 on page 399 defines the mapping definitions

Table 143: Interrupt Map Register Hardware

Register Name: IMR_HW	Register Offset: 0x42C

PCI					PB
Bits	**Function**				**Bits**
31-24	P2_HW_MAP	0	P1_HW_MAP	0	0-7
23-16	INT5_HW_MAP	0	INT4_HW_MAP	0	8-15
15-08	INT3_HW_MAP	0	INT2_HW_MAP	0	16-23
07-00	INT1_HW_MAP	0	INT0_HW_MAP	0	24-31

IMR_MISC Description

		Reset	Reset	
Name	**Type**	**By**	**State**	**Function**
P1_HW_MAP[2:0]	R/W	G_RST	0	Map PCI-1 hardware interrupt to an interrupt pin
P2_HW_MAP[2:0]	R/W	G_RST	0	Map PCI-2 hardware interrupt to an interrupt pin
				2P: Reserved
INT5_HW_MAP[2:0]	R/W	G_RST	0	Map INT[5]_ hardware interrupt to an interrupt pin
INT4_HW_MAP[2:0]	R/W	G_RST	0	Map INT[4]_ hardware interrupt to an interrupt pin
INT3_HW_MAP[2:0]	R/W	G_RST	0	Map INT[3]_ hardware interrupt to an interrupt pin
INT2_HW_MAP[2:0]	R/W	G_RST	0	Map INT[2]_ hardware interrupt to an interrupt pin

IMR_MISC Description

Name	Type	Reset By	Reset State	Function
INT1_HW_MAP[2:0]	R/W	G_RST	0	Map INT[1]_ hardware interrupt to an interrupt pin
INT0_HW_MAP[2:0]	R/W	G_RST	0	Map INT[0]_ hardware interrupt to an interrupt pin

13.5.51 Interrupt Map Register PCI-1

Each map field assigns an interrupt output pin to the corresponding interrupt source. Table 140 on page 399 defines the mapping definitions.

Table 144: Interrupt Map Register PCI 1

Register Name: IMR_P1	Register Offset: 0x430

PCI Bits	Function				PB Bits
31-24	P1_P2_ERR_MAP	0	P1_PB_ERR_MAP	0	0-7
23-16	P1_P1_ERR_MAP	0	P1_A_PAR_MAP	0	8-15
15-08	P1_P2_RETRY_MAP	0	P1_PB_RETRY_MAP	0	16-23
07-00	P1_P1_RETRY_MAP	PowerSpan II Reserved			24-31

IMR_P1 Description

Name	Type	Reset By	Reset State	Function
P1_P2_ERR_MAP [2:0]	R/W	G_RST	0	Map PCI-1 errors to an interrupt pin
				2P: Reserved
P1_PB_ERR_MAP [2:0]	R/W	G_RST	0	Map PCI-1 errors to an interrupt pin
P1_P1_ERR_MAP [2:0]	R/W	G_RST	0	Map PCI-1 errors to an interrupt pin. PCI-1 to PCI-1 DMA.
P1_A_PAR_MAP [2:0]	R/W	G_RST	0	Map PCI-11 address parity errors to an interrupt pin
P1_P2_RETRY_ MAP[2:0]	R/W	G_RST	0	Map PCI-1 max retry error to an interrupt pin
				2P: Reserved
P1_PB_RETRY_ MAP[2:0]	R/W	G_RST	0	Map PCI-1 max retry error to an interrupt pin
P1_P1_RETRY_ MAP[2:0]	R/W	G_RST	0	Map PCI-1 max retry error to an interrupt pin. PCI-1 to PCI-1 DMA.

13.5.52 Interrupt Map Register PCI-2

This register assigns an interrupt output pin to the corresponding interrupt source. All sources are associated with errors detected by the PCI-2 Interface. Table 140 on page 399 defines the mapping definitions.

This register is not implemented in the Single PCI PowerSpan II and must be treated as reserved.

Table 145: Interrupt Map Register PCI-2

Register Name: IMR_P2	Register Offset: 434

PCI Bits	Function				PB Bits
31-24	P2_P1_ERR_MAP	0	P2_PB_ERR_MAP	0	0-7
23-16	P2_P2_ERR_MAP	0	P2_A_PAR_MAP	0	8-15
15-08	P2_P1_RETRY_MAP	0	P2_PB_RETRY_MAP	0	16-23
07-00	P2_P2_RETRY_MAP	PowerSpan II Reserved			24-31

IMR_P2 Description

Name	Type	Reset By	Reset State	Function
P2_P1_ERR[2:0]	R/W	G_RST	0	Map PCI-2 errors to an interrupt pin
P2_PB_ERR[2:0]	R/W	G_RST	0	Map PCI-2 errors to an interrupt pin
P2_P2_ERR_MAP[2:0]	R/W	G_RST	0	Map PCI-2 errors to an interrupt pin. PCI-2 to PCI-2 DMA.
P2_A_PAR_MAP[2:0]	R/W	G_RST	0	Map PCI-2 address parity errors to an interrupt pin
P2_P1_RETRY_MAP[2:0]	R/W	G_RST	0	Map PCI-2 max retry error to an interrupt pin
P2_PB_RETRY_MAP[2:0]	R/W	G_RST	0	Map PCI-2 max retry error to an interrupt pin
P2_P2_RETRY_MAP[2:0]	R/W	G_RST	0	Map PCI-2 max retry error to an interrupt pin. PCI-2 to PCI-2 DMA.

13.5.53 Interrupt Map Register Processor Bus

This register assigns an interrupt output pin to the corresponding interrupt source. All sources are associated with errors detected by the Processor Bus Interface. Table 140 on page 399 defines the mapping definitions.

Table 146: Interrupt Map Register Processor Bus

Register Name: IMR_PB	Register Offset: 0x438

PCI Bits	Function				PB Bits
31-24	PB_P1_ERR_MAP	0	PB_P2_ERR_MAP	0	0-7
23-16	PB_PB_ERR_MAP	0	PB_A_PAR_MAP	0	8-15
15-08	PB_P1_D_PAR_MAP	0	PB_P2_D_PAR_MAP	0	16-23
07-00	PB_PB_D_PAR_MAP	PowerSpan II Reserved			24-31

IMR_PB Description

Name	Type	Reset By	Reset State	Function
PB_P1_ERR_MAP[2:0]	R/W	G_RST	0	Map Processor Bus error to an interrupt pin
PB_P2_ERR_MAP[2:0]	R/W	G_RST	0	Map Processor Bus error to an interrupt pin
				2P: Reserved
PB_PB_ERR_MAP[2:0]	R/W	G_RST	0	Map Processor Bus error to an interrupt pin. Processor Bus to Processor Bus DMA.
PB_A_PAR_MAP[2:0]	R/W	G_RST	0	Map Processor Bus address parity error to an interrupt pin
PB_P1_D_PAR_MAP[2:0]	R/W	G_RST	0	Map Processor Bus data parity error to an interrupt pin
PB_P2_D_PAR_MAP[2:0]	R/W	G_RST	0	Map Processor Bus data parity error to an interrupt pin

IMR_PB Description

Name	Type	Reset By	Reset State	Function
				2P: Reserved
PB_PB_D_PAR_MAP[2:0]	R/W	G_RST	0	Map Processor Bus data parity error to an interrupt pin. Processor Bus to Processor Bus DMA.

13.5.54 Interrupt Map Register Two Processor Bus

This register maps Processor Bus maximum retry errors to interrupt pins. Max retry errors that are mapped include PCI-1, PCI-2 and Processor Bus.

Table 147: Interrupt Map Register Two Processor Bus

Register Name: IMR2_PB	**Register Offset: 0x43C**

PCI Bits	Function				PB Bits
31-24	PB_P1_RETRY_MAP	0	PB_P2_RETRY_MAP	0	0-7
23-16	PB_PB_RETRY_MAP	PowerSpan II Reserved			8-15
15-08	PowerSpan II Reserved				16-23
07-00	PowerSpan II Reserved				24-31

Name	Type	Reset By	Reset State	Function
PB_P1_RETRY_ MAP[2:0]	R/W	G_RST	0	Map Processor Bus max retry errors to an interrupt pin
PB_P2_RETRY_ MAP[2:0]	R/W	G_RST	0	Map Processor Bus max retry errors to an interrupt pin
				Single PCI PowerSpan II Reserved
PB_PB_RETRY_ MAP[2:0]	R/W	G_RST	0	Map Processor Bus max retry errors to an interrupt pin. Processor Bus to Processor Bus DMA.

13.5.55 Interrupt Map Register Miscellaneous

Each map field assigns an interrupt output pin to the corresponding interrupt source. Table 140 on page 399 defines the mapping definitions.

Table 148: Interrupt Map Register Miscellaneous

Register Name: IMR_MISC	**Register Offset: 0x440**

PCI Bits	**Function**				**PB Bits**
31-24	I2O_IOP_MAP	0	I2O_HOST_MAP	0	0-7
23-16	PowerSpan II Reserved				8-15
15-08	PowerSpan II Reserved				16-23
07-00	PowerSpan II Reserved				24-31

IMR_MISC Description

Name	**Type**	**Reset By**	**Reset State**	**Function**
I2O_HOST_MAP [2:0]	R/W	G_RST	0	Map I2O Host interrupt to an interrupt pin
I2O_IOP_MAP[2:0]	R/W	G_RST	0	Map I2O IOP interrupt to an interrupt pin

I2O_HOST_MAP: This field must be configured to route the interrupt source to the interrupt pin on PowerSpan II's Primary PCI Interface

13.5.56 Interrupt Direction Register

This register controls the direction of the corresponding interrupt pin. The direction can be to be an input or output.

Table 149: Interrupt Direction Register

Register Name: IDR	Register Offset: 0x444

PCI Bits	Function								PB Bits
31-24	P2_HW_DIR	P1_HW_DIR	INT5_HW_DIR	INT4_HW_DIR	INT3_HW_DIR	INT2_HW_DIR	INT1_HW_DIR	INT0_HW_DIR	0-7
23-16	PowerSpan II Reserved								8-15
15-08	PowerSpan II Reserved								16-23
07-00	PowerSpan II Reserved								24-31

IDR Description

Name	Type	Reset By	Reset State	Function
P2_HW_DIR	R/W	G_RST	0 EEPROM	P2_INTA _ Direction 0 = Input 1 = Output
				2P: Reserved
P1_HW_DIR	R/W	G_RST	0 EEPROM	P1_INTA_ Direction 0 = Input 1 = Output
INT5_HW_DIR	R/W	G_RST	0 EEPROM	INT[5]_ Interrupt Direction 0 = Input 1 = Output
INT4_HW_DIR	R/W	G_RST	0 EEPROM	INT[4]_ Interrupt Direction 0 = Input 1 = Output
INT3_HW_DIR	R/W	G_RST	0 EEPROM	INT[3]_ Interrupt Direction 0 = Input 1 = Output

IDR Description

Name	Type	Reset By	Reset State	Function
INT2_HW_DIR	R/W	G_RST	0 EEPROM	INT[2]_ Interrupt Direction 0 = Input 1 = Output
INT1_HW_DIR	R/W	G_RST	0 EEPROM	INT[1]_ Interrupt Direction 0 = Input 1 = Output
INT0_HW_DIR	R/W	G_RST	0 EEPROM	INT[0]_ Interrupt Direction 0 = Input 1 = Output

13.5.57 Mailbox x Register

This register is the General Purpose Mailbox register. When interrupts are enabled in the IER0 register, writes to any byte of this register cause an interrupt. The interrupt can be mapped to any of PowerSpan II's interrupt pins. This mapping is set in the IMR_MBOX register.

Table 150: Mailbox x Register

Register Name: MBOXx	Register Offset: 0x450, 0x454, 0x458, 0x45C, 0x460, 0x464, 0x468, 0x46C

PCI Bits	Function	PB Bits
31-24	MBOXx	0-7
23-16	MBOXx	8-15
15-08	MBOXx	16-23
07-00	MBOXx	24-31

MBOXx Description

Name	Type	Reset By	Reset State	Function
MBOXx [31:0]	R/W	G_RST	0	Mailbox x

13.5.58 Semaphore 0 Register

When a SEMx is 0, this semaphore can be obtained by writing a 1 to the SEMx bit with a unique tag TAGx. If on a subsequent read, the SEMx bit is set and the TAGx field contains the same unique tag, then the semaphore has been obtained successfully.

To release a semaphore, write a 0 to the SEMx bit and the same tag that was used to obtain the semaphore. If the tag is different from the tag that is in the register, then the write will have no effect.

Access to a single semaphore in this register requires a byte-wide transaction.

Table 151: Semaphore 0 Register

Register Name: SEMA0	**Register Offset: 0x470**

PCI Bits	**Function**		**PB Bits**
31-24	SEM3	TAG3	0-7
23-16	SEM2	TAG2	8-15
15-08	SEM1	TAG1	16-23
07-00	SEM0	TAG0	24-31

SEMA0 Description

Name	**Type**	**Reset By**	**Reset State**	**Function**
SEM3	R/W	G_RST	0	Semaphore 3
TAG3[6:0]	R/W	G_RST	0	Tag 3
SEM2	R/W	G_RST	0	Semaphore 2
TAG2[6:0]	R/W	G_RST	0	Tag 2
SEM1	R/W	G_RST	0	Semaphore 1
TAG1[6:0]	R/W	G_RST	0	Tag 1
SEM0	R/W	G_RST	0	Semaphore 0
TAG0[6:0]	R/W	G_RST	0	Tag 0

13.5.59 Semaphore 1 Register

If a SEMx is 0, this semaphore can be obtained by writing a 1 to the SEMx bit with a unique tag TAGx. If on a subsequent read, the SEMx bit is set and the TAGx field contains the same unique tag, then the semaphore has been obtained successfully.

To release a semaphore, write a 0 to the SEMx bit and the same tag that was used to obtain the semaphore. If the tag is different from the tag that is in the register, then the write will have no effect.

Access to a single semaphore in this register requires a byte-wide transaction.

Table 152: Semaphore 1 Register

Register Name: SEMA1	Register Offset: 0x474

PCI Bits	Function		PB Bits
31-24	SEM7	TAG7	0-7
23-16	SEM6	TAG6	8-15
15-08	SEM5	TAG5	16-23
07-00	SEM4	TAG4	24-31

SEMA1 Description

Name	Type	Reset By	Reset State	Function
SEM7	R/W	G_RST	0	Semaphore 7
TAG7[6:0]	R/W	G_RST	0	Tag 7
SEM6	R/W	G_RST	0	Semaphore 6
TAG6[6:0]	R/W	G_RST	0	Tag 6
SEM5	R/W	G_RST	0	Semaphore 5
TAG5[6:0]	R/W	G_RST	0	Tag 5
SEM4	R/W	G_RST	0	Semaphore 4
TAG4[6:0]	R/W	G_RST	0	Tag 4

13.5.60 PCI I2O Target Image Control Register

This register contains the control information for the PowerSpan II PCI I^2O Target Image. The lower 4 Kbytes of the image provide the I^2O Shell Interface - Inbound and Outbound Queues and the Host Interrupt Status and Mask Registers. I^2O Message Frames are accessible above the 4-KByte boundary. The Queues and the Message Frames reside in memory connected to the processor bus.

All PCI transactions claimed by the image are destined for the Processor Bus.

Table 153: PCI I2O Target Image Control Register

Register Name: PCI_TI2O_CTL	**Register Offset: 0x500**

PCI Bits	**Function**						**PB Bits**
31-24	IMG_EN	TA_EN	BAR_EN	0	BS		0-7
23-16	PowerSpan II Reserved			RTT			8-15
15-08	GBL	CI	0	WTT			16-23
07-00	PRKEEP	END		MRA	0	RD_AMT	24-31

PCI_TI2O_CTL Description

Name	**Type**	**Reset By**	**Reset State**	**Function**
IMG_EN	R/W	PRI_RST	0	Image Enable 0 = Disable 1 = Enable
TA_EN	R/W	PRI_RST	0	Translation Address Enable 0 = Disable 1 = Enable
BAR_EN	R/W	PRI_RST	0 EEPROM	PCI Base Address Register Enable 0 = Disable 1 = Enable
BS[3:0]	R/W	PRI_RST	0 EEPROM	Block Size (64 Kbyte * 2^{BS})

PCI_TI2O_CTL Description

Name	Type	Reset By	Reset State	Function
RTT[4:0]	R/W	PRI_RST	0b01010	Processor Bus Read Transaction Type (PB_TT[0:4]) 01010 = Read
GBL	R/W	PRI_RST	0	Global 0=Assert PB_GBL_ 1=Negate PB_GBL_
CI	R/W	PRI_RST	0	Cache Inhibit 0=Assert PB_CI_ 1=Negate PB_CI_
WTT[4:0]	R/W	PRI_RST	0b00010	Processor Bus Write Transaction Type (PB_TT[0:4]) 00010=Write with flush
PRKEEP	R/W	PRI_RST	0	Prefetch Read Keep Data 0 = Disable 1 = Enable
END[1:0]	R/W	PRI_RST	10b	Endian Conversion Mode 00 = Little-endian 01 = PowerPC little-endian 10 = Big-endian 11 = True little-endian
MRA	R/W	PRI_RST	0	PCI Memory Read Alias to MRM 0 = Disabled 1 = Enabled
RD_AMT[2:0]	R/W	PRI_RST	0	Prefetch Size Specifies the number of bytes the device will prefetch for PCI Memory Read Multiple transactions claimed by the target image

The following parameters do not affect Processor Bus transactions generated by I_2O Shell accesses:

- TA_EN

  no address translation for I^2O Shell accesses

- PRKEEP

  no read keep for I^{2}O Shell accesses
- END
- RD_AMT

  prefetch amount fixed at 8 bytes for I^{2}O Shell accesses

IMG_EN: The Image Enable bit can be changed with the following actions:

- initial write to Px_BSI2O[BA]
- register write to IMG_EN

The image enable is cleared by writing a zero to IMG_EN or writing zero to the Px_BSI2O[BA] field. This effectively disables I2O functionality.

TA_EN: When set, the Translation Address (TADDR[15:0]) field, in the PCI_TI2O_TADDR replaces the upper bits of the PCI bus address. The new address is used on the Processor Bus. Clearing the enable will result in no address translation.

BAR_EN: When this bit is enabled the Px_BSI2O register is Read/Write. When this bit is disabled the Px_BSI2O register is not visible and read zero only. Writes to Px_BSI2O have no effect when this bit is cleared. This bit must be enabled for PCI BIOS configuration in order to map PowerSpan II PCI I^{2}O Target Image into memory space.

The Image is enabled for decode when IMG_EN and BAR_EN are set.

MRA: When set, the PCI I2O Target Image will alias a PCI Memory Read cycle to a PCI Memory Read Multiple cycle and prefetches the number of bytes specified in the RD_AMT[2:0] field. When MRA is the Target Image prefetches 8 bytes when a PCI Memory Read cycle is decoded and claimed.

BS: The Block Size specifies the size of the image, address lines compared and address lines translated.

Table 154: Block Size

BS[3:0]	Block Size	Address Lines Compared/Translated
0000	64k	AD31-AD16
0001	128K	AD31-AD17
0010	256K	AD31-AD18

Table 154: Block Size

BS[3:0]	Block Size	Address Lines Compared/Translated
0011	512K	AD31-AD19
0100	1M	AD31-AD20
0101	2M	AD31-AD21
0110	4M	AD31-AD22
0111	8M	AD31-AD23
1000	16M	AD31-AD24
1001	32M	AD31-AD25
1010	64M	AD31-AD26
1011	128M	AD31-AD27
1100	256M	AD31-AD28
1101	512M	AD31-AD29
1110	1G	AD31-AD30
1111	2G	AD31
10100-11111	Reserved	Reserved

END: This selects the endian conversion mode.

PRKEEP: This bit is used to hold read data is fetched beyond the initial PCI read cycle. When set, subsequent read requests to the same image at the next address retrieves the read data directly from the switching fabric instead of causing the destination bus to fetch more data. The read data is invalidated when a read with a non-matching address occurs.

RD_AMT[2:0]: The read amount setting determines different values to prefetch from the destination bus.

Table 155: Read Amount

RD_AMT[2:0]	Data Fetched
000	8bytes
001	16 bytes
010	32 bytes

Table 155: Read Amount

RD_AMT[2:0]	Data Fetched
011	64 bytes
100	128 bytes
101-111	Reserved

13.5.61 PCI I2O Target Image Translation Address Register

Address translation does not occur for I^{2}O Shell Interface accesses.

Table 156: PCI I2O Target Image Translation Address Register

Register Name: PCI_TI2O_TADDR	Register Offset: 0x504

PCI Bits	Function	PB Bits
31-24	TADDR	0-7
23-16	TADDR	8-15
15-08	PowerSpan II Reserved	16-23
07-00	PowerSpan II Reserved	24-31

PCI_TI2O_TADDR Description

Name	Type	Reset By	Reset State	Function
TADDR[15:0]	R/W	PRI_RST	0	Translation Address (through substitution)

TADDR[15:0]: When the TA_EN bit, in the PCI_TI2O_CTL register, is set TADDR[15:0] replaces the PCI bus upper address bits, up to the size of the image.

13.5.62 I2O Control and Status Register

Table 157: I2O Control and Status Register

Register Name: I2O_CSR	Register Offset: 0x508

PCI Bits	Function						PB Bits
31-24	PowerSpan II Reserved						0-7
23-16	PowerSpan II Reserved						8-15
15-08	PowerSpan II Reserved						16-23
07-00	HOPL_SIZE	EMTR	OFL	IPL	XI$_2$O_EN	I$_2$O_EN	24-31

I2O_CSR Description

Name	Type	Reset By	Reset State	Function
HOPL_SIZE [2:0]	R/W	PRI_RST	0	Host Outbound Post List Size
EMTR	R/W	PRI_RST	0	Empty FIFO Read Response 0 = return pointer on read when FIFO empty 1 = return 0xFFFF_FFFF on read when FIFO empty
OFL	R	PRI_RST	0	Outbound Free List 0 = empty 1 = not empty
IPL	R	PRI_RST	0	Inbound Post List 0 = empty 1 = not empty
XI$_2$O_EN	R/W	PRI_RST	0	Extended MFA Enabled
I$_2$O_EN	R/W	PRI_RST	0	I$_2$O Enabled 0 = I2O disabled 1 = I2O enabled

HOPL_SIZE: This field specifies the size of the Host Outbound Post List circular FIFO in the Host memory. The IOP must program this field when PowerSpan II extended Outbound Option support is enabled.

Table 158: Host Outbound Post List Size

HOPL_SIZE [2:0]	Max No. of MFAs per FIFO	Memory Required per FIFO (Kbytes)	PowerSpan II IOP Host Outbound Index Register bits incremented
000 001	256	1	IOP_OI [9:2]
010	1K	4	IOP_OI [11:2]
100	4K	16	IOP_OI [13:2]

EMTR: The Empty FIFO Read Response bit determines the PowerSpan II response to an IOP read of the Inbound Post List Bottom Pointer Register or the Outbound Free List Bottom Pointer Register. If the EMTR bit is set, a read from either of these registers when their corresponding FIFO is empty will return 0xffff_ffff as read data to the IOP. If the bit is not set, the contents of the corresponding Pointer Register will be returned as read data.

OFL: Indicates status of the Outbound Free List FIFO. If this bit is set, at least one Outbound Message Frame is available in Host memory.

IPL: Indicates status of the Inbound Post List FIFO. If this bit is set, there are Inbound Message Frames for the IOP to process.

XI2O_EN: The IOP programs this bit to enable the PowerSpan II I^{2}O Extended Capabilities support for the Outbound Option. The Host Outbound Index Offset Register needs to be programmed with the offset in the PCI I2O target Image where the Host Outbound Index Register can be located for the Outbound Option Support. This can be accomplished through the Host Outbound Index Alias register.

The IOP will need to program the following registers to support I^{2}O extended capabilities:

- I2O IOP Outbound Index register
- I2O Host Outbound Index Offset register
- I2O Host Outbound Index Alias register

I2O_EN: The local processor sets this bit to enable the PowerSpan II I2O Shell Interface support. The IOP must initialize the I2O Inbound Free list and Post list FIFO’s before enabling the PowerSpan II I^{2}O Shell Interface. When this bit is cleared, all Px_BSI2O accesses on Primary PCI are retried.

13.5.63 I2O Queue Base Address Register

This register specifies the location and size of the Inbound and Outbound Queues in processor memory space. The IOP must program this register before enabling the PowerSpan II I2O Shell Interface.

Table 159: I2O Queue Base Address Register

Register Name: I2O_QUEUE_BS	**Register Offset: 0x50C**

<table>
<tr><th>PCI
Bits</th><th colspan="3">Function</th><th>PB
Bits</th></tr>
<tr><td>31-24</td><td colspan="3">PB_I2O_BS</td><td>0-7</td></tr>
<tr><td>23-16</td><td>PB_I2O_BS</td><td colspan="2">PowerSpan II Reserved</td><td>8-15</td></tr>
<tr><td>15-08</td><td colspan="3">PowerSpan II Reserved</td><td>16-23</td></tr>
<tr><td>07-00</td><td colspan="2">PowerSpan II Reserved</td><td>FIFO_SIZE</td><td>24-31</td></tr>
</table>

I2O_QUEUE_BS Description

Name	Type	Reset By	Reset State	Function
PB_I2O_BS [11:0]	R/W	PRI_RST	0	Processor Bus I2O Base Address
FIFO_SIZE [2:0]	R/W	PRI_RST	0	FIFO Size

PB_I2O_BS: The PB_I2O_BS field specifies the base address of the 1 MB block of embedded PowerPC memory that contains the four FIFOs (Inbound Free List, Inbound Post List, Outbound Free List, Outbound Post List). The four FIFOs are of equal size, but do not need to be in contiguous memory locations. The PB_I2O_BS field is aliased in the most significant 12 bits of each of the PowerSpan II I^{2}O Bottom and Top Pointer Registers.

FIFO_SIZE: This field specifies the size of the circular FIFOs in the IOP local memory. Total FIFO memory allocation is four times the single FIFO size.

Table 160: I2O FIFO Sizes

FIFO_SIZE [2:0]	Max No. of MFAs per FIFO	Memory Required per FIFO (Kbytes)	PowerSpan II I2O Pointer bits incremented
000	256	1	I2O_PTR [9:2]
001	1K	4	I2O_PTR [11:2]
010	4K	16	I2O_PTR [13:2]
011	16K	64	I2O_PTR [15:2]
100	64K	256	I2O_PTR [17:2]

I2O_PTR is one of the following:

- IFL_BOT
- IFL_TOP
- IPL_BOT
- IPL_TOP
- OFL_BOT
- OFL_TOP
- OPL_BOT
- OPL_TOP

13.5.64 I2O Inbound Free List Bottom Pointer Register

Table 161: I2O Inbound Free List Bottom Pointer Register

Register Name: IFL_BOT	**Register Offset: 0x510**

<table>
<tr><th>PCI
Bits</th><th colspan="4">Function</th><th>PB
Bits</th></tr>
<tr><td>31-24</td><td colspan="4">PB_I2O_BS</td><td>0-7</td></tr>
<tr><td>23-16</td><td>PB_I2O_BS</td><td colspan="3">BOT</td><td>8-15</td></tr>
<tr><td>15-08</td><td colspan="4">BOT</td><td>16-23</td></tr>
<tr><td>07-00</td><td colspan="2">BOT</td><td>0</td><td>0</td><td>24-31</td></tr>
</table>

IFL_BOT Description

Name	Type	Reset By	Reset State	Function
PB_I2O_BS [11:0]	R	PRI_RST	0	Processor Bus I2O Base Address
BOT [17:0]	R/W	PRI_RST	0	Inbound Free List Bottom Pointer

BOT: This pointer gives the address offset for the Inbound Free List Bottom Pointer from PB_I2O_BS. This pointer is initialized by the IOP and maintained by PowerSpan II. This pointer is incremented by four for each PCI read from the Inbound Queue.

⚠ If the initial values of the Inbound Free List Bottom and Top pointers are the same, the inbound free list is empty. The user can program the Top pointer to be four less than the Bottom pointer and then write to the INCR bit in the IFL_TOP_INC register to make the inbound free list full.

13.5.65 I2O Inbound Free List Top Pointer Register

Table 162: I2O Inbound Free List Top Pointer Register

Register Name: IFL_TOP	**Register Offset: 0x514**

<table>
<tr><th>PCI
Bits</th><th colspan="4">Function</th><th>PB
Bits</th></tr>
<tr><td>31-24</td><td colspan="4">PB_I2O_BS</td><td>0-7</td></tr>
<tr><td>23-16</td><td>PB_I2O_BS</td><td colspan="3">TOP</td><td>8-15</td></tr>
<tr><td>15-08</td><td colspan="4">TOP</td><td>16-23</td></tr>
<tr><td>07-00</td><td colspan="2">TOP</td><td>0</td><td>0</td><td>24-31</td></tr>
</table>

IFL_TOP Description

Name	Type	Reset By	Reset State	Function
PB_I2O_BS [11:0]	R	PRI_RST	0	Processor Bus I2O Base Address
TOP [17:0]	R/W	PRI_RST	0	Inbound Free List Top Pointer

TOP: This pointer gives the address offset for the Inbound Free List Top Pointer from PB_I2O_BS. This pointer is initialized by the IOP and can be incremented by four by writing a 1 to the INCR bit in the IFL_TOP_INC register.

If the initial values of the Inbound Free List Bottom and Top pointers are the same, the inbound free list is empty. The user can program the Top pointer to be four less than the Bottom pointer and then set the INCR bit in the IFL_TOP_INC register to make the inbound free list full.

13.5.66 Inbound Free List Top Pointer Increment Register

Table 163: I2O Inbound Free List Top Pointer Increment Register

Register Name: IFL_TOP_INC	**Register Offset: 0x518**

PCI Bits	**Function**		**PB Bits**
31-24	PowerSpan II Reserved		0-7
23-16	PowerSpan II Reserved		8-15
15-08	PowerSpan II Reserved		16-23
07-00	PowerSpan II Reserved	INCR	24-31

IFL_TOP_INC Description

Name	**Type**	**Reset By**	**Reset State**	**Function**
INCR	Write 1 to set	PRI_RST	0	Inbound Free List Top Pointer Increment Write 1 to increment the pointer by four.

13.5.67 I2O Inbound Post List Bottom Pointer Register

Table 164: I2O Inbound Post List Bottom Pointer Register

Register Name: IPL_BOT	Register Offset: 0x51C

<table>
<tr><th>PCI
Bits</th><th colspan="4">Function</th><th>PB
Bits</th></tr>
<tr><td>31-24</td><td colspan="4">PB_I2O_BS</td><td>0-7</td></tr>
<tr><td>23-16</td><td>PB_I2O_BS</td><td colspan="3">BOT</td><td>8-15</td></tr>
<tr><td>15-08</td><td colspan="4">BOT</td><td>16-23</td></tr>
<tr><td>07-00</td><td colspan="2">BOT</td><td>0</td><td>0</td><td>24-31</td></tr>
</table>

IPL_BOT Description

Name	Type	Reset By	Reset State	Function
PB_I2O_BS [11:0]	R	PRI_RST	0	Processor Bus I2O Base Address
BOT [17:0]	R/W	PRI_RST	0	Inbound Post List Bottom Pointer

BOT: This pointer gives the address offset for the Inbound Post List Bottom Pointer from PB_I2O_BS. This pointer is initialized by the IOP and can be incremented by four by setting the INCR bit in the IPL_BOT_INC register.

The initial values of the Inbound Post List Bottom and Top pointers must be the same. After these pointers are initialized, the inbound post list is empty.

13.5.68 I2O Inbound Post List Bottom Pointer Increment Register

Table 165: I2O Inbound Post List Bottom Pointer Increment Register

Register Name: IPL_BOT_INC	**Register Offset: 0x520**

PCI Bits	**Function**		**PB Bits**
31-24	PowerSpan II Reserved		0-7
23-16	PowerSpan II Reserved		8-15
15-08	PowerSpan II Reserved		16-23
07-00	PowerSpan II Reserved	INCR	24-31

IPL_BOT_INC Description

Name	**Type**	**Reset By**	**Reset State**	**Function**
INCR	Write 1 to set	PRI_RST	0	Inbound Post List Bottom Pointer Increment Write 1 to increment the pointer by four.

13.5.69 I2O Inbound Post List Top Pointer Register

Table 166: I2O Inbound Post List Top Pointer Register

Register Name: IPL_TOP	**Register Offset: 0x524**

<table>
<tr><th>PCI
Bits</th><th colspan="4">Function</th><th>PB
Bits</th></tr>
<tr><td>31-24</td><td colspan="4">PB_I2O_BS</td><td>0-7</td></tr>
<tr><td>23-16</td><td>PB_I2O_BS</td><td colspan="3">TOP</td><td>8-15</td></tr>
<tr><td>15-08</td><td colspan="4">TOP</td><td>16-23</td></tr>
<tr><td>07-00</td><td colspan="2">TOP</td><td>0</td><td>0</td><td>24-31</td></tr>
</table>

IPL_TOP Description

Name	Type	Reset By	Reset State	Function
PB_I2O_BS [11:0]	R	PRI_RST	0	Processor Bus I2O Base Address
TOP [17:0]	R/W	PRI_RST	0	Inbound Post List Top Pointer

TOP: This pointer gives the address offset for the Inbound Post List Top Pointer from PB_I2O_BS. This pointer is initialized by the IOP and maintained by PowerSpan II. This pointer is incremented by four for each PCI write to the Inbound Queue.

The initial values of the Inbound Post List Bottom and Top pointers should be the same. After these pointers are initialized, the inbound post list is empty.

13.5.70 I2O Outbound Free List Bottom Pointer Register

Table 167: I2O Outbound Free List Bottom Pointer Register

Register Name: OFL_BOT	Register Offset: 0x528

PCI Bits	Function				PB Bits
31-24	PB_I2O_BS				0-7
23-16	PB_I2O_BS	BOT			8-15
15-08	BOT				16-23
07-00	BOT		0	0	24-31

OFL_BOT Description

Name	Type	Reset By	Reset State	Function
PB_I2O_BS [11:0]	R	PRI_RST	0	Processor Bus I2O Base Address
BOT [17:0]	R/W	PRI_RST	0	Outbound Free List Bottom Pointer

BOT: This pointer gives the address offset for the Outbound Free List Bottom Pointer from PB_I2O_BS. This pointer is initialized by the IOP and can be incremented by four by writing a 1 to the INCR bit in the OFL_BOT_INC register.

⚠ The initial values of the Outbound Free List Bottom and Top pointers must be the same. After these pointers are initialized, the outbound free list is empty.

13.5.71 I2O Outbound Free List Bottom Pointer Increment Register

Table 168: I2O Outbound Free List Bottom Pointer Increment Register

Register Name: OFL_BOT_INC	**Register Offset: 0x52C**

PCI **Bits**	**Function**		**PB** **Bits**
31-24	PowerSpan II Reserved		0-7
23-16	PowerSpan II Reserved		8-15
15-08	PowerSpan II Reserved		16-23
07-00	PowerSpan II Reserved	INCR	24-31

OFL_BOT_INC Description

Name	**Type**	**Reset By**	**Reset State**	**Function**
INCR	Write 1 to set	PRI_RST	0	Outbound Free List Bottom Pointer Increment Write 1 to increment the pointer by four.

13.5.72 I2O Outbound Free List Top Pointer Register

Table 169: I2O Outbound Free List Top Pointer Register

Register Name: OFL_TOP	**Register Offset: 0x530**

PCI Bits	Function				PB Bits
31-24	PB_I2O_BS				0-7
23-16	PB_I2O_BS	TOP			8-15
15-08	TOP				16-23
07-00	TOP		0	0	24-31

OFL_TOP Description

Name	Type	Reset By	Reset State	Function
PB_I2O_BS [11:0]	R	PRI_RST	0	Processor Bus I2O Base Address
TOP [17:0]	R/W	PRI_RST	0	Outbound Free List Top Pointer

TOP: This pointer gives the address offset for the Outbound Free List Top Pointer from PB_I2O_BS. This pointer is initialized by the IOP and maintained by PowerSpan II. This pointer is incremented by four for each PCI write to the Outbound Queue.

The initial values of the Outbound Free List Bottom and Top pointers must be the same. After these pointers are initialized, the outbound free list is empty.

13.5.73 I2O Outbound Post List Bottom Pointer Register

Table 170: I2O Outbound Post List Bottom Pointer Register

Register Name: OPL_BOT	**Register Offset: 0x534**

<table>
<tr><th>PCI
Bits</th><th colspan="4">Function</th><th>PB
Bits</th></tr>
<tr><td>31-24</td><td colspan="4">PB_I2O_BS</td><td>0-7</td></tr>
<tr><td>23-16</td><td>PB_I2O_BS</td><td colspan="3">BOT</td><td>8-15</td></tr>
<tr><td>15-08</td><td colspan="4">BOT</td><td>16-23</td></tr>
<tr><td>07-00</td><td colspan="2">BOT</td><td>0</td><td>0</td><td>24-31</td></tr>
</table>

OPL_BOT Description

Name	Type	Reset By	Reset State	Function
PB_I2O_BS [11:0]	R	PRI_RST	0	Processor Bus I2O Base Address
BOT [17:0]	R/W	PRI_RST	0	Outbound Post List Bottom Pointer

BOT: This pointer gives the address offset for the Outbound Post List Bottom Pointer from PB_I2O_BS. This pointer is initialized by the IOP and maintained by PowerSpan II. This pointer is incremented by four for each PCI read from the Outbound Queue.

The initial values of the Outbound Post List Bottom and Top pointers must be the same. After these pointers are initialized, the outbound post list is empty.

13.5.74 I2O Outbound Post List Top Pointer Register

Table 171: I2O Outbound Post List Top Pointer Register

Register Name: OPL_TOP	**Register Offset: 0x538**

<table>
<tr><th>PCI
Bits</th><th colspan="4">Function</th><th>PB
Bits</th></tr>
<tr><td>31-24</td><td colspan="4">PB_I2O_BS</td><td>0-7</td></tr>
<tr><td>23-16</td><td>PB_I2O_BS</td><td colspan="3">TOP</td><td>8-15</td></tr>
<tr><td>15-08</td><td colspan="4">TOP</td><td>16-23</td></tr>
<tr><td>07-00</td><td colspan="2">TOP</td><td>0</td><td>0</td><td>24-31</td></tr>
</table>

OPL_TOP Description

Name	Type	Reset By	Reset State	Function
PB_I2O_BS [11:0]	R	PRI_RST	0	Processor Bus I2O Base Address
TOP [17:0]	R/W	PRI_RST	0	Outbound Post List Top Pointer

TOP: This pointer gives the address offset for the Outbound Post List Top Pointer from PB_I2O_BS. This pointer is initialized by the IOP and can be incremented by four by writing 1 to the INCR bit in the OPL_TOP_INC register.

⚠ The initial values of the Outbound Post List Bottom and Top pointers must be the same. After these pointers are initialized, the outbound post list is empty.

13.5.75 I2O Outbound Post List Top Pointer Increment Register

Table 172: I2O Outbound Post List Top Pointer Increment Register

Register Name: OPL_TOP_INC	**Register Offset: 0x53C**

PCI Bits	**Function**		**PB Bits**
31-24	PowerSpan II Reserved		0-7
23-16	PowerSpan II Reserved		8-15
15-08	PowerSpan II Reserved		16-23
07-00	PowerSpan II Reserved	INCR	24-31

OPL_TOP_INC Description

Name	**Type**	**Reset By**	**Reset State**	**Function**
INCR	Write 1 to set	PRI_RST	0	Outbound Post List Top Pointer Increment Write 1 to increment the pointer by four.

13.5.76 I2O Host Outbound Index Offset Register

Table 173: I2O Host Outbound Index Offset Register

Register Name: HOST_OIO	**Register Offset: 0x540**

<table>
<tr><th>PCI
Bits</th><th colspan="4">Function</th><th>PB
Bits</th></tr>
<tr><td>31-24</td><td colspan="4">PowerSpan II Reserved</td><td>0-7</td></tr>
<tr><td>23-16</td><td colspan="4">PowerSpan II Reserved</td><td>8-15</td></tr>
<tr><td>15-08</td><td colspan="2">PowerSpan II Reserved</td><td colspan="2">OIO</td><td>16-23</td></tr>
<tr><td>07-00</td><td colspan="2">OIO</td><td>0</td><td>0</td><td>24-31</td></tr>
</table>

HOST_OIO Description

Name	Type	Reset By	Reset State	Function
OIO[9:0]	R/W	PRI_RST	0	Host Outbound Index Offset

OIO[9:0]: Specifies the I2O Target Image Offset where the I2O Host Outbound Index Register is located within the PowerSpan II I2O Target Image. The I2O Host Outbound Index register must be in the first 4 Kbytes of the PowerSpan II I2O Target Image and be aligned to a 4-byte boundary.

This register must not be programmed with the following values: 0x030, 0x034, 0x040, 0x044.

13.5.77 I2O Host Outbound Index Alias Register

This register is required for PowerSpan II I^{2}O Outbound Option support. This is an alias to the I2O Host Outbound Index Register in the PowerSpan II I2O Target Image. The Host maintains this register.

This register indicates the address in Host memory from which the Host is to retrieve the next Outbound XMFA. This register is initialized by the IOP with an index received from the Host in an I2O message. The register will be written by the Host during I2O Outbound Option message passing.

Table 174: I2O Host Outbound Index Alias Register

Register Name: HOST_OIA	Register Offset: 0x544

PCI Bits	Function			PB Bits
31-24	OIA			0-7
23-16	OIA			8-15
15-08	OIA			16-23
07-00	OIA	0	0	24-31

HOST_OIA Description

Name	Type	Reset By	Reset State	Function
OIA[29:0]	R/W	PRI_RST	0	Host Outbound Index Alias Register

If the I2O Host Outbound Index Register and the I2O IOP Outbound Index Register differ, then the Outbound Post List Interrupt Status bit is set in the OPL_IS register at offset 0x30 of the PCI I2O target Image. When these registers contain the same Host memory address, the Interrupt is cleared.

This feature is only supported when the I2O Outbound Option is enabled with I2O_CSR[XI2O_EN].

I2O_CSR[HOPL_SIZE] determines the alignment of this Index Register.

13.5.78 I2O IOP Outbound Index Register

This register is required for PowerSpan II I2O Outbound Option support. This register indicates the address in Host memory to which the IOP is to post the next Outbound XMFA. The IOP maintains this register.

Table 175: I2O IOP Outbound Index Register

Register Name: IOP_OI	**Register Offset: 0x548**

PCI Bits	Function			PB Bits
31-24	OI			0-7
23-16	OI			8-15
15-08	OI			16-23
07-00	OI	0	0	24-31

IOP_OI Description

Name	Type	Reset By	Reset State	Function
OI[29:0]	R/W	PRI_RST	0	IOP Outbound Index

If the I2O Host Outbound Index Register and the I2O IOP Outbound Index Register differ, then the Outbound Post List Interrupt Status bit is set in the OPL_IS register at offset 0x30 of the PCI I2O target Image. When these registers contain the same Host memory address, the Interrupt is cleared.

This feature is only supported when the I2O Outbound Option is enabled with the XI2O_EN bit in the I2O_CSR register.

The HOPL_SIZE bit in the I2O_CSR register determines the alignment of this Index Register

13.5.79 I2O IOP Outbound Index Increment Register

The IOP writes 1 to INCR to increment the IOP Outbound Index Register.

Table 176: I2O IOP Outbound Index Increment Register

Register Name: IOP_OI_INC	**Register Offset: 0x54C**

PCI Bits	Function		PB Bits
31-24	PowerSpan II Reserved		0-7
23-16	PowerSpan II Reserved		8-15
15-08	PowerSpan II Reserved		16-23
07-00	PowerSpan II Reserved	INCR	24-31

IOP_OI_INC Description

Name	Type	Reset By	Reset State	Function
INCR	Write 1 to set	PRI_RST	0	IOP Outbound Index Increment Write 1 to increment the pointer by four.

13.5.80 PCI-2 Configuration Registers

Table 177: PCI-2 Configuration Registers

Register Offset: 0x800-0x8FC

PCI-2 Configuration

Function
The PCI-2 Configuration Registers are functionally identical to the PCI-1 Configuration Registers from offsets 0x000-0FC. Documentation of the PCI-2 Configuration Space is the same as the PCI-1 Interface, shifting the register offsets up by 0x800 and swapping PCI-1 and PCI-2 everywhere.

13.5.81 PCI-2 Target Image Control and Status Registers

Table 178: PCI-2 Target Image Control and Status Registers

Register Offset: 0x900-0x9FC

PCI-2 Target Image

Function
The PCI-2 Target Image Control and Status Registers are functionally identical to the PCI-1 Target Image Control and Status Registers from offsets 0x100-1FC. Documentation of the PCI-2 Target Images is the same as the PCI-1 Images, shifting the register offsets up by 0x800 and swapping PCI-1 and PCI-2 everywhere.

A. Package Information

This appendix discusses PowerSpan II's packaging (mechanical) features. The following topics are discussed:

- "Package Characteristics" on page 443
- "Thermal Characteristics" on page 450

A.1 Package Characteristics

PowerSpan II's package characteristics are summarized in the following sections.

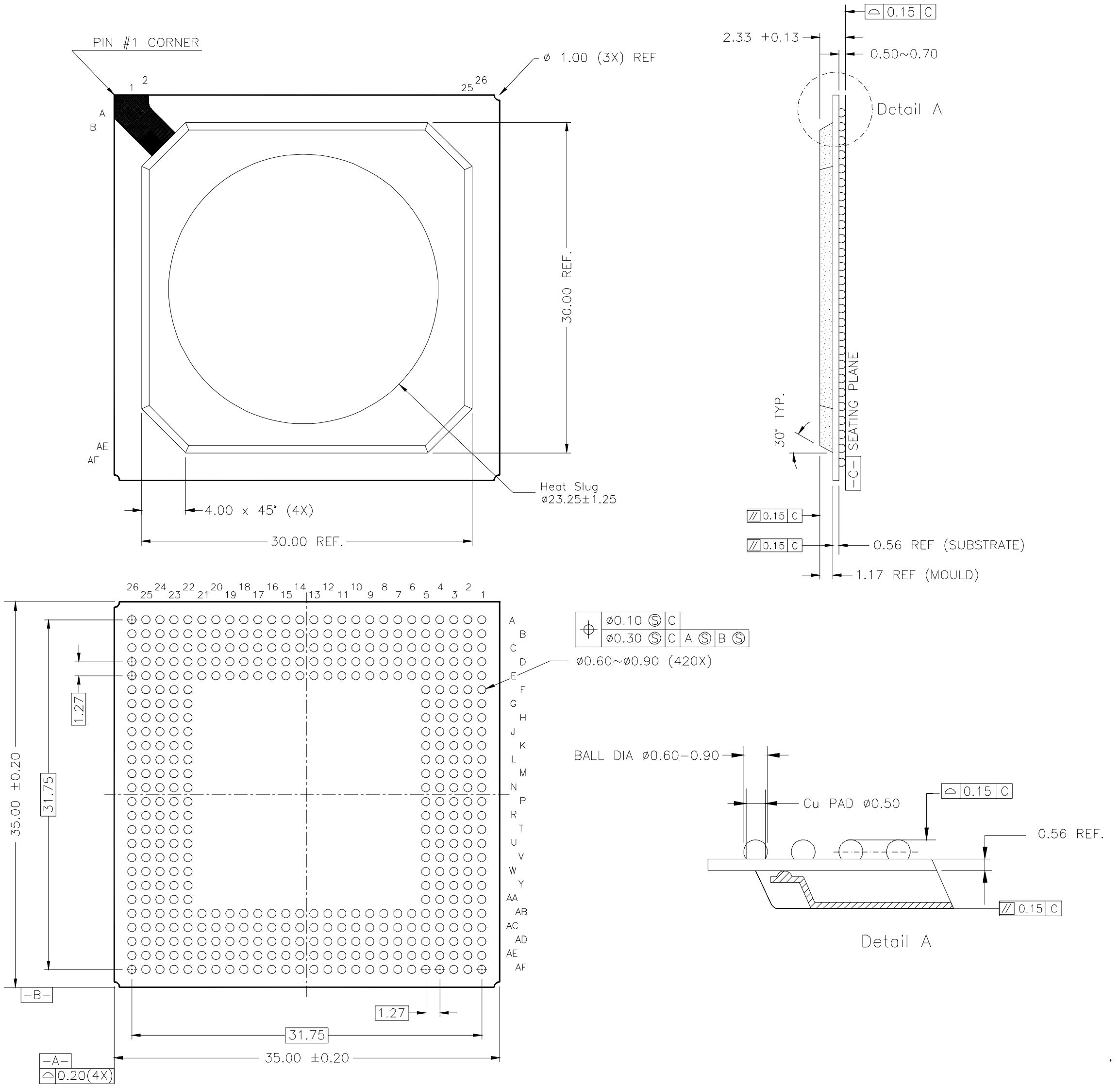
A.1.1 Single PCI PowerSpan II 420 HSBGA

Figure 28 illustrates the top, side, and bottom views of the PowerSpan II package.

Table 179: Package Characteristics

Feature	Description
Package Type	420 HSBGA
Package Body Size	35mm
JEDEC Specification	JEDEC MO-151 Variation BAT-1

Figure 28: 420 HSBGA



A.1.1.1 Package Notes

1. All dimensions in mm
2. All dimensions and tolerance conform to ANSI Y14.5M - 1994
3. Conforms to JEDEC MS-034 Variation BAR-1

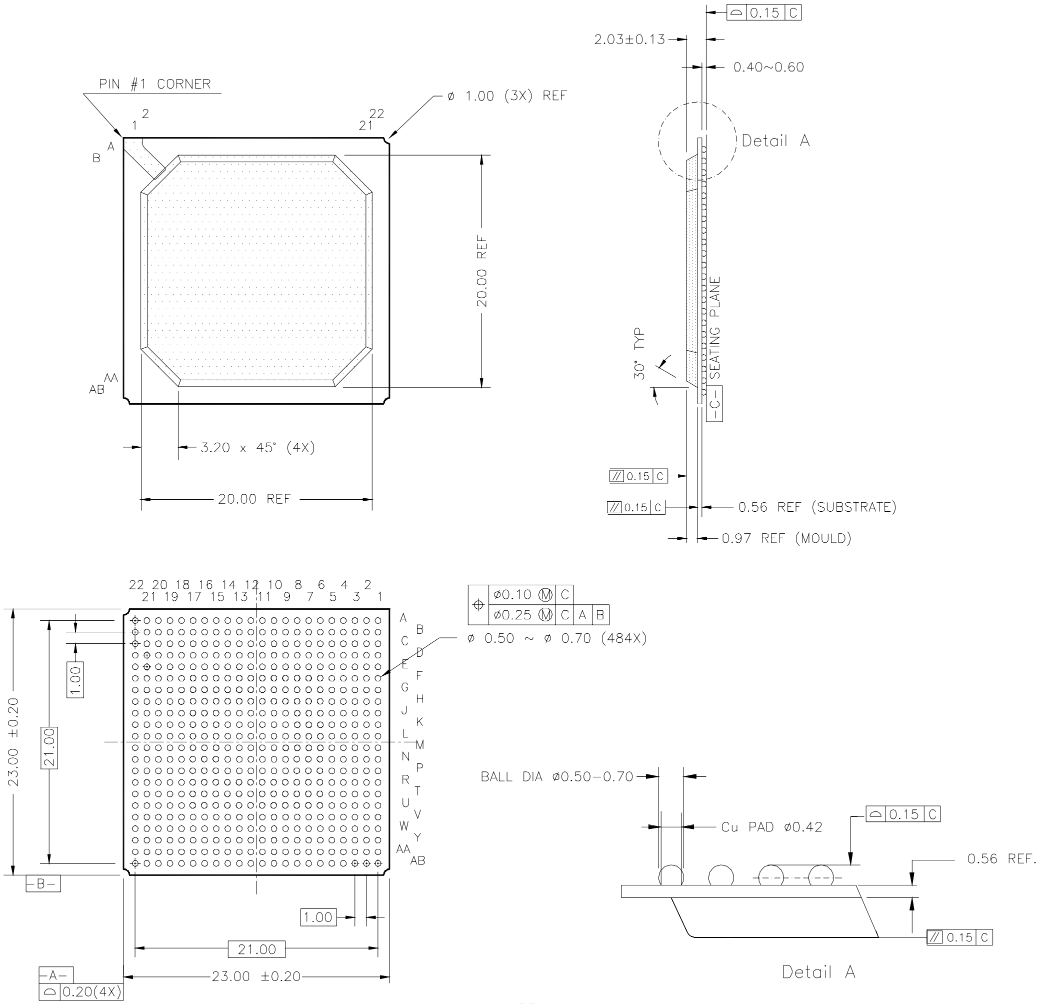
A.1.2 Single PCI PowerSpan II 484 HSBGA

Figure 29 illustrates the top, side, and bottom views of the PowerSpan II package.

Table 180: Package Characteristics

Feature	Description
Package Type	484 HSBGA
Package Body Size	23mm
JEDEC Specification	JEDEC MS-034 Variation AAJ-1

Figure 29: 484 PBGA



A.1.2.1 Package Notes

1. All dimensions in mm
2. All dimensions and tolerance conform to ANSI Y14.5M - 1994
3. Conforms to JEDEC MS-034 Variation AAJ-1

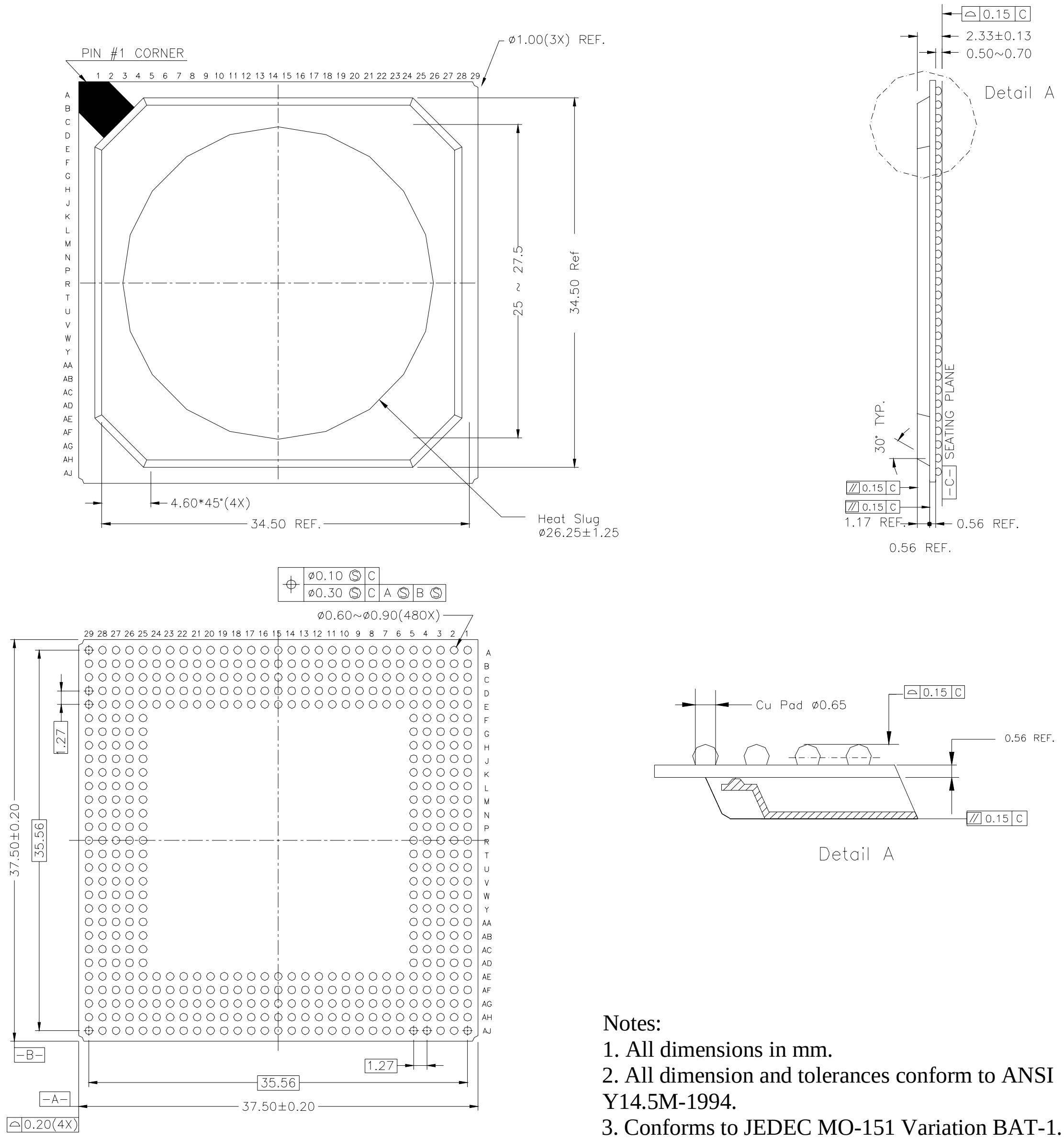
A.1.3 Dual PCI PowerSpan II 480 HSBGA

Figure 30 illustrates the top, side, and bottom views of the PowerSpan II package.

Table 181: Package Characteristics

Feature	Description
Package Type	480 HSBGA
Package Body Size	37.5mm
JEDEC Specification	JEDEC MO-151 Variation BAT-1

Figure 30: 480 HSBGA



Notes:
1. All dimensions in mm.
2. All dimension and tolerances conform to ANSI Y14.5M-1994.
3. Conforms to JEDEC MO-151 Variation BAT-1.

A.1.3.1 Package Notes

1. All dimensions in mm
2. All dimensions and tolerance conform to ANSI Y14.5M - 1994
3. Conforms to JEDEC MO-151 Variation BAT-1

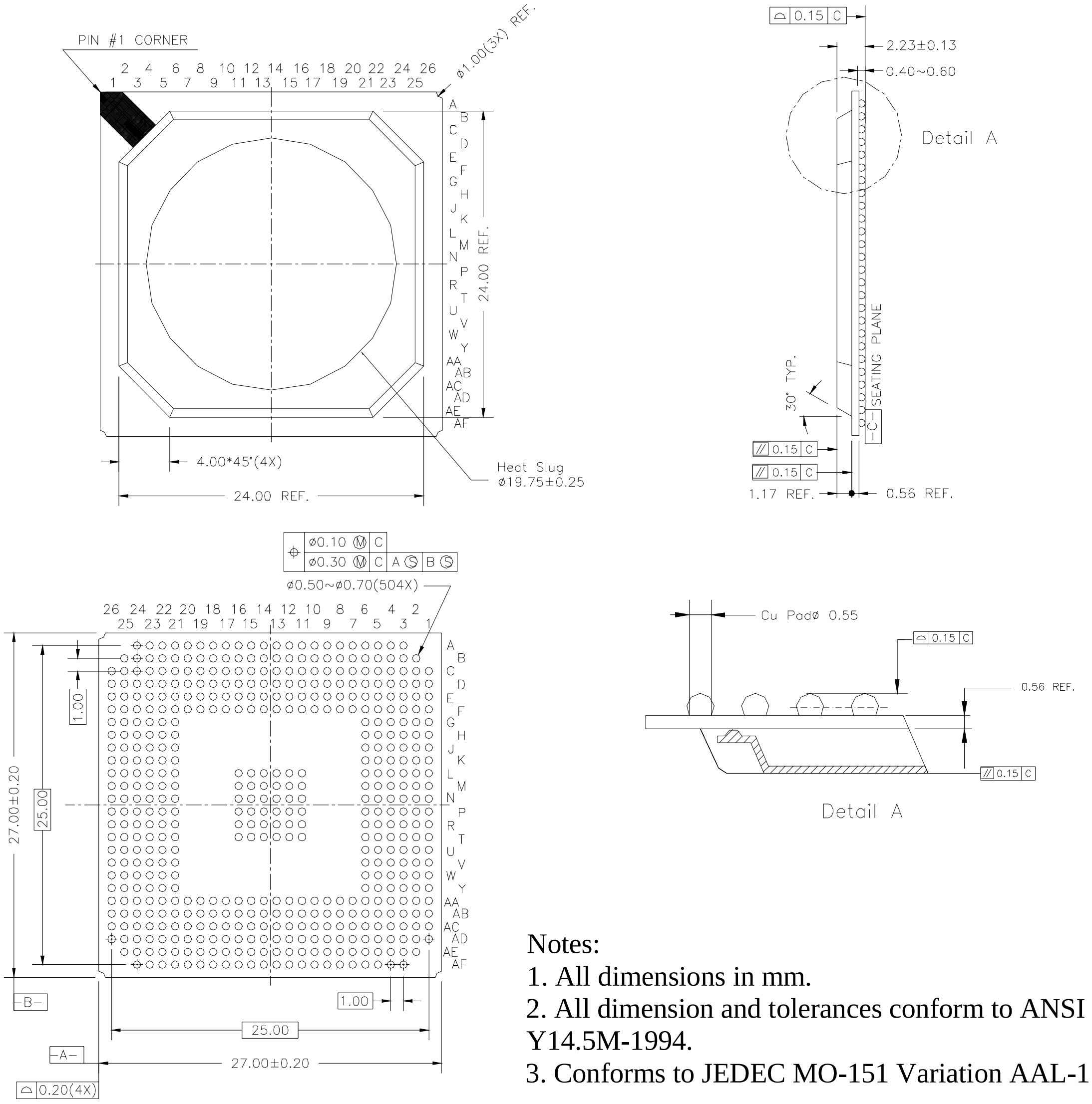
A.1.4 Dual PCI PowerSpan II 504 HSBGA

Figure 31 illustrates the top, side, and bottom views of the PowerSpan II package.

Table 182: Package Characteristics

Feature	Description
Package Type	504 HSBGA
Package Body Size	27mm
JEDEC Specification	JEDEC MO-151 Variation AAL-1

Figure 31: 504 HSBGA



Notes:
1. All dimensions in mm.
2. All dimension and tolerances conform to ANSI Y14.5M-1994.
3. Conforms to JEDEC MO-151 Variation AAL-1.

A.1.4.1 Package Notes

1. All dimensions in mm
2. All dimensions and tolerance conform to ANSI Y14.5M - 1994
3. Conforms to JEDEC MO-151 Variation AAL-1

A.2 Thermal Characteristics

The thermal performance of PowerSpan II package is represented by the following parameters:

1. θ_{JA}, Thermal resistance from junction to ambient

   $\theta_{JA} = (T_J - T_A) / P$

   Where,

   T_J is the junction temperature

   T_A is the ambient temperature

   P is the power dissipation

   θ_{JA} represents the resistance to the heat flows from the chip to ambient air. It is an index of heat dissipation capability. Lower θ_{JA} means better thermal performance.

2. ψ_{JT}, Thermal characterization parameter from junction-to-top center

   $\psi_{JT} = (T_J - T_T) / P$

   Where T_T is the temperature of the top-center of the package

   ψ_{JT} is used to estimate junction temperature by measuring T_T in actual environment.

3. 3. θ_{JC}, Thermal resistance from junction to case

   $\theta_{JC} = (T_J – T_C) / P$

   Where,

   T_C is the case temperature

   θ_{JC} is a measure of package internal thermal resistance from chip to package exterior.

   The value is dependent upon package material and package geometry.

   $\theta_{JA,}$ θ_{JC} and ψ_{JT} simulation are carried out to show the thermal performance of the PowerSpan II.

A.2.1 Single PCI 420 HSBGA Package

The thermal characteristic estimates for the 420 package are based on the parameters in Table 183.

Table 183: Thermal Estimate Parameters

Package Conditions	
Package type	HSBGA 420
Package size	35 x 35 x 2.33 mm^3
Pitch	1.27 mm
Pad size	318 x 318 mil^2
Chip size	232 x 232 mil^2
Substrate (layers)	4 L
Substrate thickness	0.56 mm
PCB Conditions (JEDEC JESD51-7)	
PCB Layers	4L
PCB dimensions	101.6 x 114.3 mm
PCB thickness	1.6 mm
Simulation conditions	
Power dissipation	1.9 watts
Ambient temperature	85 °C

Table 184 shows the thermal estimates and the thermal characterization parameters from junction-to-top center (Ψ_{JT}) and the thermal resistance from junction to case for the 420 package. These estimates are based on the parameters described in Table 183.

Table 184: 420 HSBGA Package Performance

Theta ja (C/W)			Psi$_{jt}$ (C/W)	Theta jc (C/W)
0 m/s	1 m/s	2 m/s		
17.1	14.5	13.1	5.35	6.60

A.2.2 Single PCI 484 PBGA Package

The thermal characteristic estimates for the 484 package are based on the parameters in Table 185.

Table 185: Thermal Estimate Parameters

Package Conditions	
Package type	HSBGA 484
Package size	23 x 23 x 2.03 mm^3
Pitch	1.00 mm
Pad size	357 x 357 mil^2
Chip size	207 x 207 mil^2
Substrate (layers)	4 L
Substrate thickness	1.60 mm
PCB Conditions (JEDEC JESD51-7)	
PCB Layers	4L
PCB dimensions	101.6 x 114.3 mm
PCB thickness	1.6 mm
Simulation conditions	
Simulation Conditions	
Power dissipation	1.9 watts
Ambient temperature	85 °C

Table 186 shows the thermal estimates and the thermal characterization parameters from junction-to-top center (Ψ_{JT}) and the thermal resistance from junction to case for the 484 package. These estimates are based on the parameters described in Table 185.

Table 186: 484 PBGA Package Performance

Theta ja (C/W)			Psi$_{jt}$ (C/W)	Theta jc (C/W)
0 m/s	1 m/s	2 m/s		
21.4	18.7	17.0	4.87	6.00

A.2.3 Dual PCI 480 HSBGA Package

The thermal characteristic estimates for the 480 package are based on the parameters in Table 187.

Table 187: Thermal Estimate Parameters

Package Conditions	
Package type	HSBGA 480L
Package size	37.5 x 37.5 x 2.33 mm^3
Pitch	1.27 mm
Pad size	303 x 303 mil^2
Chip size	253 x 253 mil^2
Substrate (layers)	4 L
Substrate thickness	0.56 mm
PCB Conditions (JEDEC JESD51-7)	
PCB Layers	4L
PCB dimensions	101.6 x 114.3 mm
PCB thickness	1.6 mm
Simulation Conditions	
Power dissipation	3.0 watts
Ambient temperature	85 °C

Table 188 shows the thermal estimates and the thermal characterization parameters from junction-to-top center (Ψ_{JT}) and the thermal resistance from junction to case for the 480 package. These estimates are based on the parameters described in Table 187.

Table 188: 480 PBGA Package Performance

Theta ja (C/W)			Psi$_{jt}$ (C/W)	Theta jc (C/W)
0 m/s	1 m/s	2 m/s		
15.1	13.2	11.8	4.87	6.00

A.2.4 Dual PCI 504 HSBGA Package.

The Thermal Characteristic estimates for the 504 package are based on the parameters in Table 189.

Table 189: Thermal Estimate Parameters

Package Conditions	
Package type	HSBGA 504L
Package size	27 x 27 x 2.33 mm^3
Pitch	1.00 mm
Pad size	309 x 309 mil^2
Chip size	259 x 259 mil^2
Substrate (layers)	4 L
Substrate thickness	0.56 mm
PCB Conditions (JEDEC JESD51-7)	
PCB Layers	4L
PCB dimensions	101.6 x 114.3 mm
PCB thickness	1.6 mm
Simulation Conditions	
Power dissipation	3.0 watts
Ambient temperature	85 °C

Table 190 shows the thermal estimates and the thermal characterization parameters from junction-to-top center (Ψ_{JT}) and the thermal resistance from junction to case for the 504 package. These estimates are based on the parameters described in Table 189.

Table 190: 504 PBGA Package Performance

Theta ja (C/W)				
0 m/s	1 m/s	2 m/s	Psi$_{jt}$ (C/W)	Theta jc (C/W)
16.4	14.7	11.8	4.10	5.90

B. Ordering Information

This appendix discusses PowerSpan II's ordering information.

B.1 Ordering Information

Tundra products are designated by a part number. When ordering, refer to products by their full part number. For detailed mechanical drawings or alternative packaging requirements, please contact Tundra Semiconductor directly (see "Contact Information" on page 5).

Table 191: Standard Ordering Information

Part Number	Description	Frequency	Voltage (IO/ CORE)	Temperature	Package	Diameter (mm)
CA91L8200B-100CEZ2	Dual PCI PowerSpan II	100MHz	3.3/2.5	0° to 70°C	480 HSBGA	37.5 x 37.5 x 1.27
CA91L8200B-100CLZ2	Dual PCI PowerSpan II	100MHz	3.3/2.5	0° to 70°C	504 HSBGA	27 x 27 x 1.00
CA91L8200B-100IEZ2	Dual PCI PowerSpan II	100MHz	3.3/2.5	-40° to 85°C	480 HSBGA	37.5 x 37.5 x 1.27
CA91L8200B-100ILZ2	Dual PCI PowerSpan II	100MHz	3.3/2.5	-40° to 85°C	504 HSBGA	27 x 27 x 1.00

Table 191: Standard Ordering Information

Part Number	Description	Frequency	Voltage (IO/ CORE)	Temperature	Package	Diameter (mm)
CA91L8260B-100CE	Single PCI PowerSpan II	100MHz	3.3/2.5	0° to 70°C	420 HSBGA	35 x 35 x 1.27
CA91L8260B-100CL	Single PCI PowerSpan II	100MHz	3.3/2.5	0° to 70°C	484 PBGA	23 x 23 x 1.00
CA91L8260B-100IE	Single PCI PowerSpan II	100MHz	3.3/2.5	-40° to 85°C	420 HSBGA	35 x 35 x 1.27
CA91L8260B-100IL	Single PCI PowerSpan II	100MHz	3.3/2.5	-40° to 85°C	484 PBGA	23 x 23 x 1.00

C. AC Timing

This chapter describes PowerSpan II’s AC Timing parameters. The following topics are discussed:

- “Single PCI PowerSpan II Timing Parameters” on page 458
- “Dual PCI PowerSpan II Timing Parameters” on page 464
- “Timing Diagrams” on page 470

C.1 Overview

This chapter describes the timing information for the PowerSpan II device. The timing for the both the Single and the Dual PCI PowerSpan II’s Processor Bus Interface is 100 MHz, while the PCI Interface(s) can operate either at 33 MHz or 66 MHz.

C.2 Single PCI PowerSpan II Timing Parameters

The timing parameters specified in this document are guaranteed by design. Test conditions for timing parameters in Table 192 to Table 196 are:

- Commercial (C): 0ºC to 70ºC, 3.15 - 3.45V, 2.38 - 2.63V
- Industrial (I): -40ºC to 85ºC, 3.15 - 3.45V, 2.38 - 2.63V

Table 192: Reset, and Clock Timing Parameters

Timing Parameter	Description	CE/IE		Units	Note
		Min	Max		
Reset Timing					
t_{100}	Power-up reset pulse width	500		ns	1
t_{101}	Back end power stable to back end power-up reset released.	500		ns	
t_{102}	Clock frequency stable before release of power-up reset	0		ns	2
t_{103}	PLL lock time	100	400	us	3
t_{104}	Reset propagation delay		20	ns	
t_{428}	TRST_ pulse width	500		ns	4
Power-up Option Timing					
t_{110}	Power-up option setup time on multiplexed system pins	10		ns	
t_{111}	Power-up option hold time on multiplexed system pins	3.0	10	ns	5
t_{112}	Power-up option setup time to PB_RSTCONF_	10		ns	
t_{113}	Power-up option hold time from PB_RSTCONF_	5		ns	
t_{114}	PB_RSTCONF_ pulse width	1		PB_CLKs	
Processor Clock Timing					
t_{120}	PB_CLK period	10	40	ns	
	PB_CLK frequency	25	100	MHz	
t_{121}	PB_CLK high time	4		ns	
t_{122}	PB_CLK low time	4		ns	

Table 192: Reset, and Clock Timing Parameters

Timing Parameter	Description	CE/IE		Units	Note
		Min	Max		
t_{123}	PB_CLK slew rate	2		V/ns	
t_{124}	PB_CLK cycle to cycle jitter		300	ps	
PCI Clock Timing					
t_{130}	P1_CLK, P2_CLK period	15	40	ns	
	P1_CLK, P2_CLK frequency	25	66	MHz	
t_{131}	P1_CLK, P2_CLK high time	6		ns	
t_{132}	P1_CLK, P2_CLK low time	6		ns	
t_{133}	P1_CLK, P2_CLK slew rate	2		V/ns	
t_{134}	P1_CLK, P2_CLK cycle to cycle jitter		300	ps	
Clock to Clock Relationships					
t_{140}	Clock period ratio	1	< 4	-	6

1. Pulse width measured from Vdd Core (2.5V), Vdd I/O (3.3V), and Px_VDDA supplies in specification
2. Required for PB_CLK, P1_CLK and P2_CLK. This parameter ensures that each PLL locks. If a frequency change is required, a new power-up sequence must be initiated.
3. This parameter is a function of the slowest frequency of PB_CLK, P1_CLK, and P2_CLK. The minimum occurs at 100 MHz, the maximum at 25 MHz. After this time, PowerSpan II is synchronized to external buses and able to participate in transactions once externally applied resets are released.
4. Assertion of TRST_ is required at power-up to initialize the JTAG controller and configure the Boundary Scan Register for normal system operation.
5. The maximum specification ensures correct power-up levels on PB_FAST, P1_M66EN and P2_M66EN and ensures stable system levels on INT[5:1]_ before the power-up reset sequence completes.
6. The ratio of largest to smallest clock period for PB_CLK, P1_CLK, P2_CLK must be strictly less than four. For example, if PB_CLK period is 12 ns, the periods of P1_CLK and P2_CLK must be each less than 48 ns.

PowerSpan II's PCI interface can be configured for operating frequencies between 25 and 33 MHz by ensuring that the pin Px_M66EN is connected to logic zero. Table 193 summarizes the timing behavior of a PowerSpan II PCI interface configured in this way. This table is valid for operation in 3.3V or 5.0V signaling environments.

Table 193: PCI 33 MHz Timing Parameters

Timing Parameter	Description	CE/IE		Units	Note
		Min	Max		
t_{200}	Float to active delay	2		ns	
t_{201}	Active to float delay		28	ns	
t_{202}	Signal valid delay				
	Bussed signals	2	11	ns	
	Point to point signals	2	12	ns	1
t_{203}	Input setup time				
	Bussed signals	7		ns	
	Point to point signals	10		ns	2
t_{204}	Input hold time	0		ns	
t_{205}	P1_REQ64# to P1_RST# setup time				
	Adapter scenario	10		PB_CLKs	3
	Host scenario	10		PB_CLKs	5
t_{206}	P1_RST# to P1_REQ64# hold time				
	Adapter scenario	2.3		ns	3, 4
	Host scenario	0	50	ns	5
t_{207}	Reset to float		40	ns	

1. This group of point to point signals include: P1_REQ[1]#, P2_REQ[1]#, P1_GNT[4:1]#, P2_GNT[4:1]# and PCI_GNT[7:5]#.
2. This group of point to point signals include: P1_GNT[1]#, P2_GNT[1]#, P1_REQ[4:1]#, P2_REQ[4:1#], and PCI_REQ[7:5]#.
3. In the adapter scenario, an external agent controls both P1_REQ64# and P1_RST#.
4. In the *PCI Local Bus Specification (Revision 2.2)* this value is required to be 0 ns.
5. In the host scenario, PowerSpan II controls both P1_REQ64# and P1_RST#.

Each PowerSpan II PCI interface can be configured for 66 MHz operating frequencies by ensuring that the pin Px_M66EN is connected to logic one. Table 194 summarizes the timing behavior of a PowerSpan II PCI interface configured in this way. This table is valid for operation in a 3V signaling environment.

Table 194: PCI 66 MHz Timing Parameters

Timing Parameter	Description	CE/IE		Units	Note
		Min	Max		
t_{200}	Float to active delay	1		ns	
t_{201}	Active to float delay		14	ns	
t_{202}	Signal valid delay				
	Bussed signals	1	6.0	ns	
	Point to point signals	1	6.0	ns	1
t_{203}	Input setup time				
	Bussed signals	3.0		ns	
	Point to point signals	5		ns	2
t_{204}	Input hold time	0		ns	
t_{205}	P1_REQ64# to P1_RST# setup time				
	Adapter scenario	10		PB_CLKs	3
	Host scenario	10		PB_CLKs	4
t_{206}	P1_RST# to P1_REQ64# hold time				
	Adapter scenario	2.3		ns	3
	Host scenario	0	50	ns	4
t_{207}	Reset to float		40	ns	

1. This group of point to point signals include: P1_REQ[1]#, P2_REQ[1]#, P1_GNT[4:1]#, P2_GNT[4:1]# and PCI_GNT[7:5]#.
2. This group of point to point signals include: P1_GNT[1]#, P2_GNT[1]#, P1_REQ[4:1]#, P2_REQ[4:1]#, and PCI_REQ[7:5]#.
3. In the adapter scenario, an external agent controls both P1_REQ64# and P1_RST#.
4. In the host scenario, PowerSpan II controls both P1_REQ64# and P1_RST#.

Table 195: PB Timing Parameters

Timing Parameter	Description	CE/IE		Units	Note
		Min	Max		
t_{302}	PB_CLK to output valid delay				
	Parameter group outputs	1.0	5.0	ns	1,2
	Control group outputs	1.0	5.0	ns	1,3
	PB_ARTRY_	1.0	5.0	ns	1
	Arbitration group outputs	1.0	5.0	ns	1, 4
t_{303}	Input setup time				
	Bussed signals	3.0		ns	
	PB_AP	2.0		ns	
	Point to point signals	3.0		ns	5
t_{304}	Input hold time	0.3		ns	

1. Numbers measured into lumped 35 pF load.
2. The transaction parameter group of signals includes: PB_A[0:31], PB_AP[0:3], PB_TSIZ[0:3], PB_TT[0:4], PB_TBST_, PB_GBL_, PB_CI_, PB_D[0:63], PB_DP[0:7].
3. The transaction control group of signals includes: PB_TS_, PB_ABB_, PB_DBB_, PB_TA_, PB_DVA_L, PB_TEA_, PB_AACK_.
4. The transaction arbitration group outputs includes: PB_BR[1]_, PB_BG[1:3]_, PB_DBG[1:3]_
5. The point to point signals include: PB_BG[1]_, PB_BR[1:3]_, PB_DBG[1]_

Table 196: Miscellaneous Timing Parameters

Timing Parameter	Description	CE/IE		Units	Note
		Min	Max		
Interrupt Timing					
t_{400}	Float to active delay	2	15	ns	1
t_{401}	Active to float delay	2	15	ns	1
t_{402}	Input setup time	3		ns	2
t_{403}	Input hold time	0.5		ns	2
t_{404}	Pulse width	4		PB_CLKs	3
I^2C Timing					
t_{410}	I2C_SCLK period	1024	1024	PB_CLKs	
t_{411}	I2C_SCLK high time	512	512	PB_CLKs	
t_{412}	I2C_SCLK low time	512	512	PB_CLKs	
t_{413}	STOP condition setup time	512	512	PB_CLKs	
t_{414}	Bus free time	512		PB_CLKs	
t_{415}	START condition setup time	1024		PB_CLKs	
t_{416}	START condition hold time	512	512	PB_CLKs	
t_{417}	Data output valid time	256	256	PB_CLKs	
t_{418}	Data output hold time	256	256	PB_CLKs	
t_{419}	Data sample time	256	256	PB_CLKs	

[1] Numbers measured into 35 pF load.

[2] PowerSpan II synchronizes these inputs before using them. This parameter must be met for deterministic response time.

[3] PowerSpan II filters these inputs to ensure spurious low going pulses are not recognized as active interrupts. An interrupt pin is considered valid if three PB_CLK samples yield the same result.

C.3 Dual PCI PowerSpan II Timing Parameters

The timing parameters specified in this document are guaranteed by design. Test conditions for timing parameters in Table 197 to Table 201 are:

- Commercial (C): 0ºC to 70ºC, 3.15 - 3.45V, 2.38 - 2.63V
- Industrial (I): -40ºC to 85ºC, 3.15 - 3.45V, 2.38 - 2.63V

Table 197: Reset, and Clock Timing Parameters

Timing Parameter	Description	CE/IE		Units	Note
		Min	Max		
Reset Timing					
t_{100}	Power-up reset pulse width	500		ns	1
t_{101}	Back end power stable to back end power-up reset released.	500		ns	
t_{102}	Clock frequency stable before release of power-up reset	0		ns	2
t_{103}	PLL lock time	100	400	us	3
t_{104}	Reset propagation delay		20	ns	
t_{428}	TRST_ pulse width	500		ns	4
Power-up Option Timing					
t_{110}	Power-up option setup time on multiplexed system pins	10		ns	
t_{111}	Power-up option hold time on multiplexed system pins	3.0	10	ns	5
t_{112}	Power-up option setup time to PB_RSTCONF_	10		ns	
t_{113}	Power-up option hold time from PB_RSTCONF_	5		ns	
t_{114}	PB_RSTCONF_ pulse width	1		PB_CLKs	
Processor Clock Timing					
t_{120}	PB_CLK period	10	40	ns	
	PB_CLK frequency	25	100	MHz	
t_{121}	PB_CLK high time	4		ns	
t_{122}	PB_CLK low time	4		ns	

Table 197: Reset, and Clock Timing Parameters

Timing Parameter	Description	CE/IE		Units	Note
		Min	Max		
t_{123}	PB_CLK slew rate	2		V/ns	
t_{124}	PB_CLK cycle to cycle jitter		300	ps	
PCI Clock Timing					
t_{130}	P1_CLK, P2_CLK period	15	40	ns	
	P1_CLK, P2_CLK frequency	25	66	MHz	
t_{131}	P1_CLK, P2_CLK high time	6		ns	
t_{132}	P1_CLK, P2_CLK low time	6		ns	
t_{133}	P1_CLK, P2_CLK slew rate	2		V/ns	
t_{134}	P1_CLK, P2_CLK cycle to cycle jitter		300	ps	
Clock to Clock Relationships					
t_{140}	Clock period ratio	1	< 4	-	6

1. Pulse width measured from Vdd Core (2.5V), Vdd I/O (3.3V), and Px_VDDA supplies in specification
2. Required for PB_CLK, P1_CLK and P2_CLK. This parameter ensures that each PLL locks. If a frequency change is required, a new power-up sequence must be initiated.
3. This parameter is a function of the slowest frequency of PB_CLK, P1_CLK, and P2_CLK. The minimum occurs at 100 MHz, the maximum at 25 MHz. After this time, PowerSpan II is synchronized to external buses and able to participate in transactions once externally applied resets are released.
4. Assertion of TRST_ is required at power-up to initialize the JTAG controller and configure the Boundary Scan Register for normal system operation.
5. The maximum specification ensures correct power-up levels on PB_FAST, P1_M66EN and P2_M66EN and ensures stable system levels on INT[5:1]_ before the power-up reset sequence completes.
6. The ratio of largest to smallest clock period for PB_CLK, P1_CLK, P2_CLK must be strictly less than four. For example, if PB_CLK period is 12 ns, the periods of P1_CLK and P2_CLK must be each less than 48 ns.

PowerSpan II's PCI interface can be configured for operating frequencies between 25 and 33 MHz by ensuring that the pin Px_M66EN is connected to logic zero. Table 198 summarizes the timing behavior of a PowerSpan II PCI interface configured in this way. This table is valid for operation in 3.3V or 5.0V signaling environments.

Table 198: PCI 33 MHz Timing Parameters

Timing Parameter	Description	CE/IE		Units	Note
		Min	Max		
t_{200}	Float to active delay	2		ns	
t_{201}	Active to float delay		28	ns	
t_{202}	Signal valid delay				
	Bussed signals	2	11	ns	
	Point to point signals	2	12	ns	1
t_{203}	Input setup time				
	Bussed signals	7		ns	
	Point to point signals	10		ns	2
t_{204}	Input hold time	0		ns	
t_{205}	P1_REQ64# to P1_RST# setup time				
	Adapter scenario	10		PB_CLKs	3
	Host scenario	10		PB_CLKs	4
t_{206}	P1_RST# to P1_REQ64# hold time				
	Adapter scenario	2.3		ns	3
	Host scenario	0	50	ns	4
t_{207}	Reset to float		40	ns	

1. This group of point to point signals include: P1_REQ[1]#, P2_REQ[1]#, P1_GNT[4:1]#, P2_GNT[4:1]#, and PCI_GNT[7:5]#.
2. This group of point to point signals include: P1_GNT[1]#, P2_GNT[1]#, P1_REQ[4:1]#, P2_REQ[4:1]#, and PCI_REQ[7:5]#.
3. In the adapter scenario, an external agent controls both P1_REQ64# and P1_RST#.
4. In the host scenario, PowerSpan II controls both P1_REQ64# and P1_RST#.

Each PowerSpan II PCI interface can be configured for 66 MHz operating frequencies by ensuring that the pin Px_M66EN is connected to logic one. Table 199 summarizes the timing behavior of a PowerSpan II PCI interface configured in this way. This table is valid for operation in a 3V signaling environment.

Table 199: PCI 66 MHz Timing Parameters

Timing Parameter	Description	CE/IE		Units	Note
		Min	Max		
t_{200}	Float to active delay	1		ns	
t_{201}	Active to float delay		14	ns	
t_{202}	Signal valid delay				
	Bussed signals	1	6.0	ns	
	Point to point signals	1	6.0	ns	1
t_{203}	Input setup time				
	Bussed signals	3.0		ns	
	Point to point signals	5		ns	2
t_{204}	Input hold time	0		ns	
t_{205}	P1_REQ64# to P1_RST# setup time				
	Adapter scenario	10		PB_CLKs	3
	Host scenario	10		PB_CLKs	4
t_{206}	P1_RST# to P1_REQ64# hold time				
	Adapter scenario	2.3		ns	3
	Host scenario	0	50	ns	4
t_{207}	Reset to float		40	ns	

1. This group of point to point signals include: P1_REQ[1]#, P2_REQ[1]#, P1_GNT[4:1]#, P2_GNT[4:1]#, and PCI_GNT[7:5]#.
2. This group of point to point signals include: P1_GNT[1]#, P2_GNT[1]#, P1_REQ[4:1]#, P2_REQ[4:1]#, and PCI_REQ[7:5]#.
3. In the adapter scenario, an external agent controls both P1_REQ64# and P1_RST#.
4. In the host scenario, PowerSpan II controls both P1_REQ64# and P1_RST#.

Table 200: PB Timing Parameters

Timing Parameter	Description	CE/IE		Units	Note
		Min	Max		
t_{302}	PB_CLK to output valid delay				
	Parameter group outputs	1.0	5.0	ns	1,2
	Control group outputs	1.0	5.0	ns	1,3
	PB_ARTRY_	1.0	5.0	ns	1
	Arbitration group outputs	1.0	5.0	ns	1, 4
t_{303}	Input setup time				
	Bussed signals	3.0		ns	
	PB_AP	2.0		ns	
	Point to point signals	3.0		ns	5
	PB_BG[1]_	3.5		ns	
t_{304}	Input hold time				
	PB_DBG[1]_	0.2		ns	
	All other inputs	0.3		ns	

1. Numbers measured into lumped 35 pF load.
2. The transaction parameter group of signals includes: PB_A[0:31], PB_AP[0:3], PB_TSIZ[0:3], PB_TT[0:4], PB_TBST_, PB_GBL_, PB_CI_, PB_D[0:63], PB_DP[0:7].
3. The transaction control group of signals includes: PB_TS_, PB_ABB_, PB_DBB_, PB_TA_, PB_DVA_L, PB_TEA_, PB_AACK_.
4. The transaction arbitration group outputs includes: PB_BR[1]_, PB_BG[1:3]_, PB_DBG[1:3]_
5. The point to point signals include: PB_BR[1:3]_, and PB_DBG[1]_

Table 201: Miscellaneous Timing Parameters

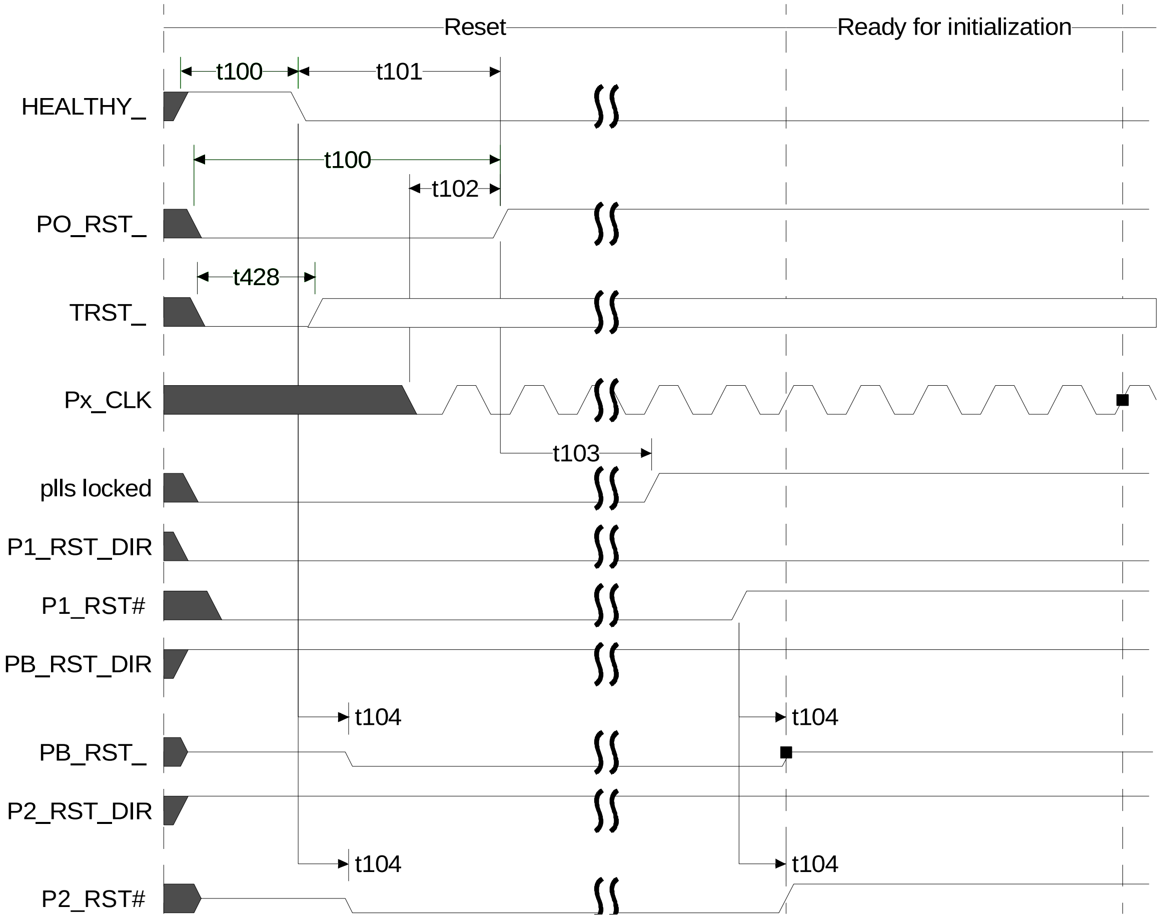
Timing Parameter	Description	CE/IE		Units	Note
		Min	Max		
Interrupt Timing					
t_{400}	Float to active delay	2	15	ns	1
t_{401}	Active to float delay	2	15	ns	1
t_{402}	Input setup time	3		ns	2
t_{403}	Input hold time	0.5		ns	2
t_{404}	Pulse width	4		PB_CLKs	3
I^2C Timing					
t_{410}	I2C_SCLK period	1024	1024	PB_CLKs	
t_{411}	I2C_SCLK high time	512	512	PB_CLKs	
t_{412}	I2C_SCLK low time	512	512	PB_CLKs	
t_{413}	STOP condition setup time	512	512	PB_CLKs	
t_{414}	Bus free time	512		PB_CLKs	
t_{415}	START condition setup time	1024		PB_CLKs	
t_{416}	START condition hold time	512	512	PB_CLKs	
t_{417}	Data output valid time	256	256	PB_CLKs	
t_{418}	Data output hold time	256	256	PB_CLKs	
t_{419}	Data sample time	256	256	PB_CLKs	

1. Numbers measured into 35 pF load.
2. PowerSpan II synchronizes these inputs before using them. This parameter must be met for deterministic response time.
3. PowerSpan II filters these inputs to ensure spurious low going pulses are not recognized as active interrupts. An interrupt pin is considered valid if three PB_CLK samples yield the same result.

C.4 Timing Diagrams

The timing diagrams in this section apply to both the Single PCI PowerSpan II and the Dual PCI PowerSpan II.

Figure 32: Power-up Reset: Compact PCI Adapter Scenario



Notes:

1. P1_RST# configured as input
2. PB_RST_ and P2_RST# configured as output
3. If JTAG is not used, the TRST_ signal can be pulled low through a resistor (~2.5 KOhm).

Figure 33: Power-up Options: Multiplexed System Pin Approach

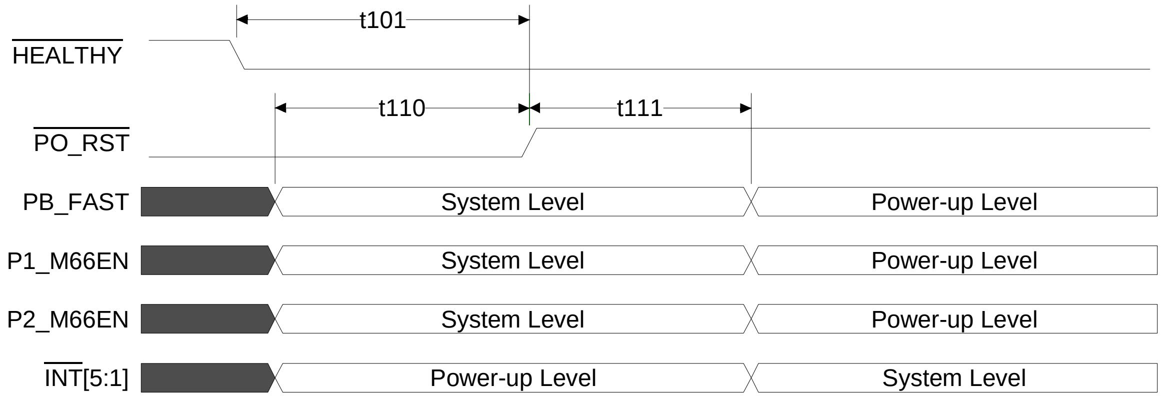
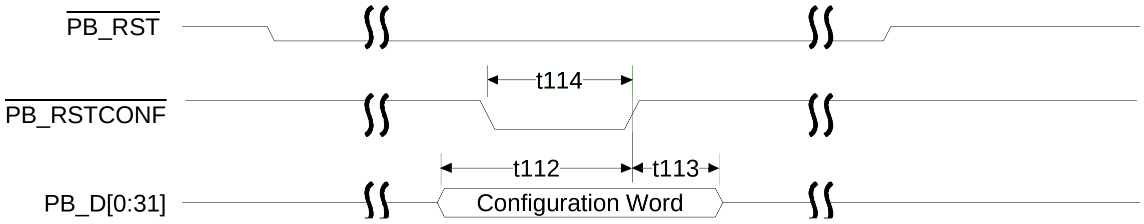


Figure 34: Power-up Options: MPC8260 Configuration Slave Approach



Notes:

1. The power-up options latched by the MPC8260 Configuration Slave mode take precedence over those latched by the Multiplexed System Pins mode.
2. The MPC8260 configuration master runs configuration cycles as part of each HRESET_ sequence.

E. Typical Applications

This chapter describes typical applications for PowerSpan II as well as PowerPC direct connection to PowerSpan II. The following topics are discussed:

E.1 Overview

Motorola’s family of embedded PowerPC processors are widely used in the deployment of communications products. The introduction of the MPC8260, with its unsurpassed levels of integration and performance, enhances the role of PowerPC in communications systems. PowerSpan II has a general purpose Processor Bus (PB) Interface to Motorola’s PowerPC embedded processor family, which enables the design of PCI based communication products.

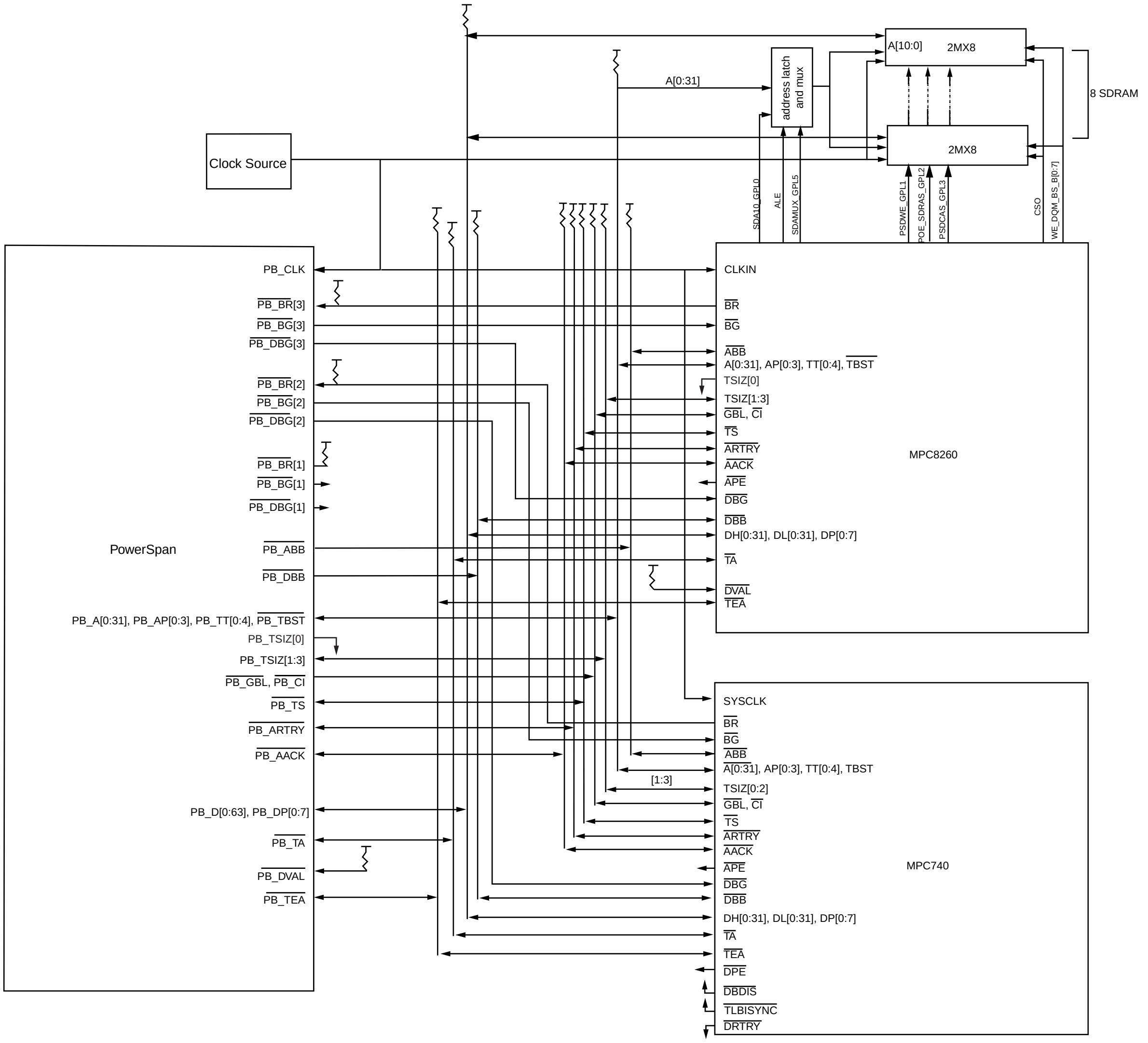
This section highlights the use of PowerSpan II in PowerPC and CompactPCI applications.

E.2 PowerSpan II and the Processor Bus

The PowerSpan II PB Interface provides direct connect support for a number of PowerPC embedded processors. The block diagram in Figure 45 illustrates a system where the PowerSpan II and the MPC8260 and MPC740 are directly connected.

The MPC740 is a 64-bit bus master in this system. It can only interact with agents that have a 64-bit port size. Since the MPC740 does not generate the extended transaction types of the MPC8260, it needs to be configured to meet MPC8260 constraints.

Figure 45: PowerSpan II in Multi-processor 60x system



The MPC8260 is the system memory controller being used in this application in order to manage 64-bit wide SDRAM. The MPC8260 has processor bus master and slave capability. As a bus master in this system it can access SDRAM and PCI. The address latch and multiplexor allow the external processor bus agents, the MPC740, and PowerSpan II to access MPC8260 controlled memory. Additionally, the MPC740 and PowerSpan II can program MPC8260 registers and master the MPC8260 local bus.

Figure 45 illustrates the system where PowerSpan II provides the following support:

- Processor bus

  Address and data bus arbitration

  Processor bus Master/Slave capability
- Single or dual PCI access
- Four channel DMA

The PowerSpan II PB Interface fully supports the multi processing cache coherent aspects of the processor bus. The PowerSpan II PB Interface can only interact with 64-bit port size agents.

The presence of the MPC740 limits the extent of extended cycle support in the system depicted in Figure 45.

The SIZ[0] pin is hardwired on PowerSpan II and the MPC8260. PowerSpan II must operate with extended cycles disabled. It is still possible for the MPC8260 bus master to target SDRAM with extended cycles.

The PowerSpan II processor bus arbiter controls system boot. Boot can be selected from PCI by configuring the arbiter at power-up to ignore all external requests on Bus Request (PB_BR[3:1]_). This allows an external PCI master to configure the MPC8260 memory controller and load system boot code before enabling recognition of requests on PB_BR[3:1]_.

Alternatively, at power-up the processor bus arbiter is configured to recognize requests on PB_BR[1]_ and ignore requests on PB_BR[3:2]_. In this case the processor connected to PB_BR[1]_ enables recognition of requests from other masters when its system configuration tasks are complete.

E.3 CompactPCI Adapter Card

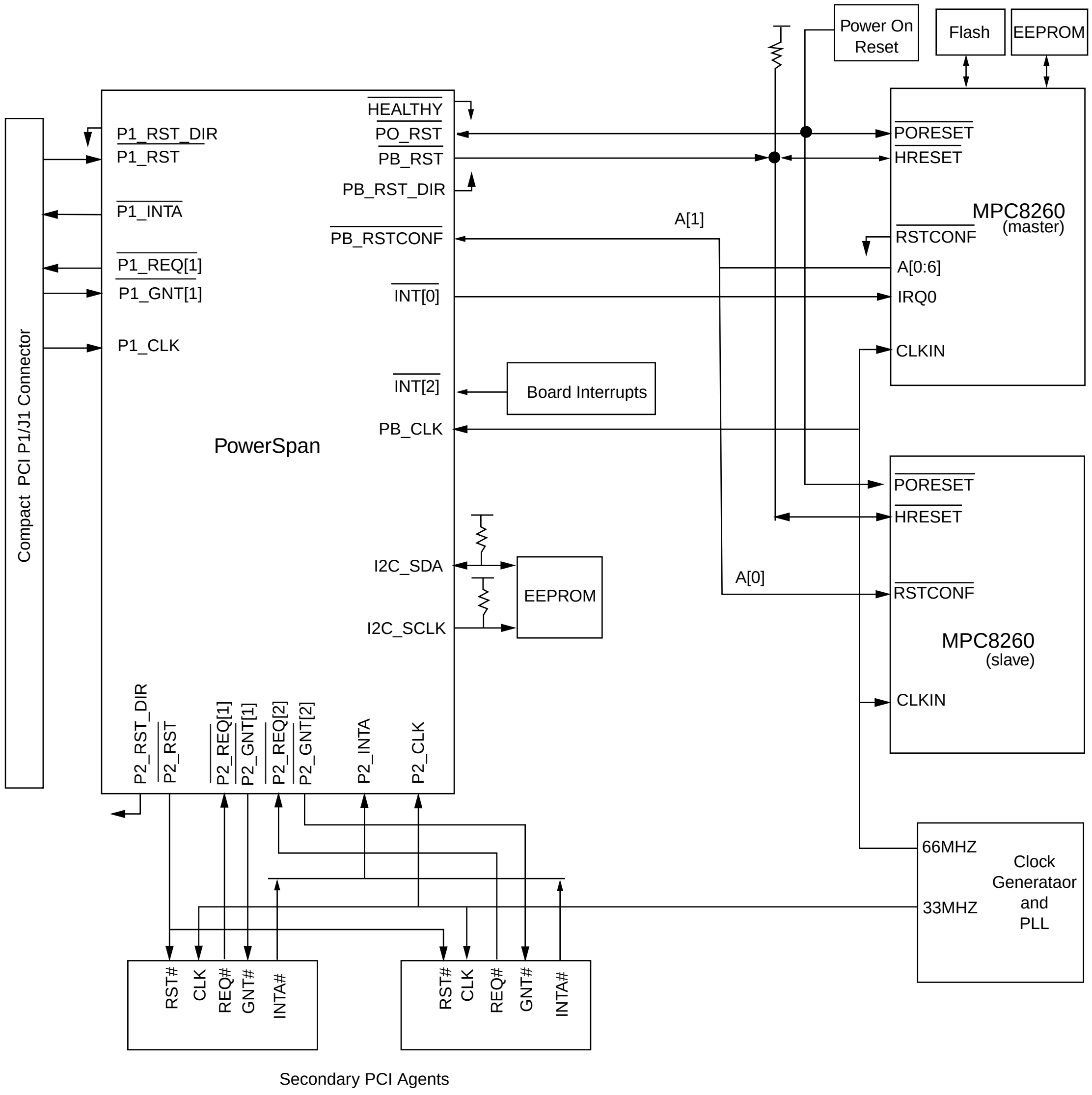
A common PowerSpan II application is the support of MPC8260 based CompactPCI adapter cards. These cards are installed in peripheral slots of the CompactPCI chassis.

In Figure 46 PowerSpan II is in a dual MPC8260 application. One processor is selected to be the Configuration master (RSTCONF_ is tied low) while the second processor, and PowerSpan II, are configuration slaves.

Optionally, the second processor could have the PowerPC core disabled and be used strictly to provide more serial interface capability.

PowerSpan II’s PCI-1 Interface is designated as the Primary Interface, through power-up option, and is connected to the CompactPCI backplane. It is possible to designate either PCI-1 or PCI-2 as the Primary Interface with a power-up option. The backplane supplies reset, clock and central resource functionality. The Secondary PCI Interface, PCI- 2, connects to a secondary PCI system on the card and provides reset and arbitration for the secondary bus.

Figure 46: PowerSpan II in CompactPCI Peripheral Slot



All PowerSpan II resources are reset when PO_RST_ is asserted by the card's power on reset logic. PB_RST_ and P2_RST# are configured as outputs and are asserted during the power on reset sequence (PO_RST_) or during a CompactPCI reset (P1_RST#). The connection between PowerSpan II PB_RST_ and MPC8260 HRESET _is required for PowerSpan II to load its power-up options during Configuration activity generated by the Configuration master.

The adapter card has two basic configuration scenarios — MPC8260 system boots from local FLASH or from PCI.

When the MPC8260 system boots from local FLASH there are two possible scenarios which can occur. In the first case, after reset, PowerSpan II retries all accesses to its Primary PCI target. The MPC8260 configures the PowerSpan II Primary and Secondary Base Address Registers and then configures all agents on the Secondary bus. The MPC8260 then enables accesses to the PowerSpan II Primary PCI target. The CompactPCI Host then completes the configuration of all Primary PCI agents.

In the second scenario, an agent on the PCI secondary bus configures all agents there. The PowerSpan II Secondary PCI target retries all accesses until the MPC8260 completed configuration of PowerSpan II Secondary Base Address Registers.

PowerSpan II supports system boot from PCI with the processor bus arbiter.

E.4 CompactPCI Host Card

PowerSpan II supports the MPC8260 as a Host in a CompactPCI system. The application illustration, Figure 47, shows a directly connected PowerSpan II in a MPC8260 system, which is supported by the MPC740.

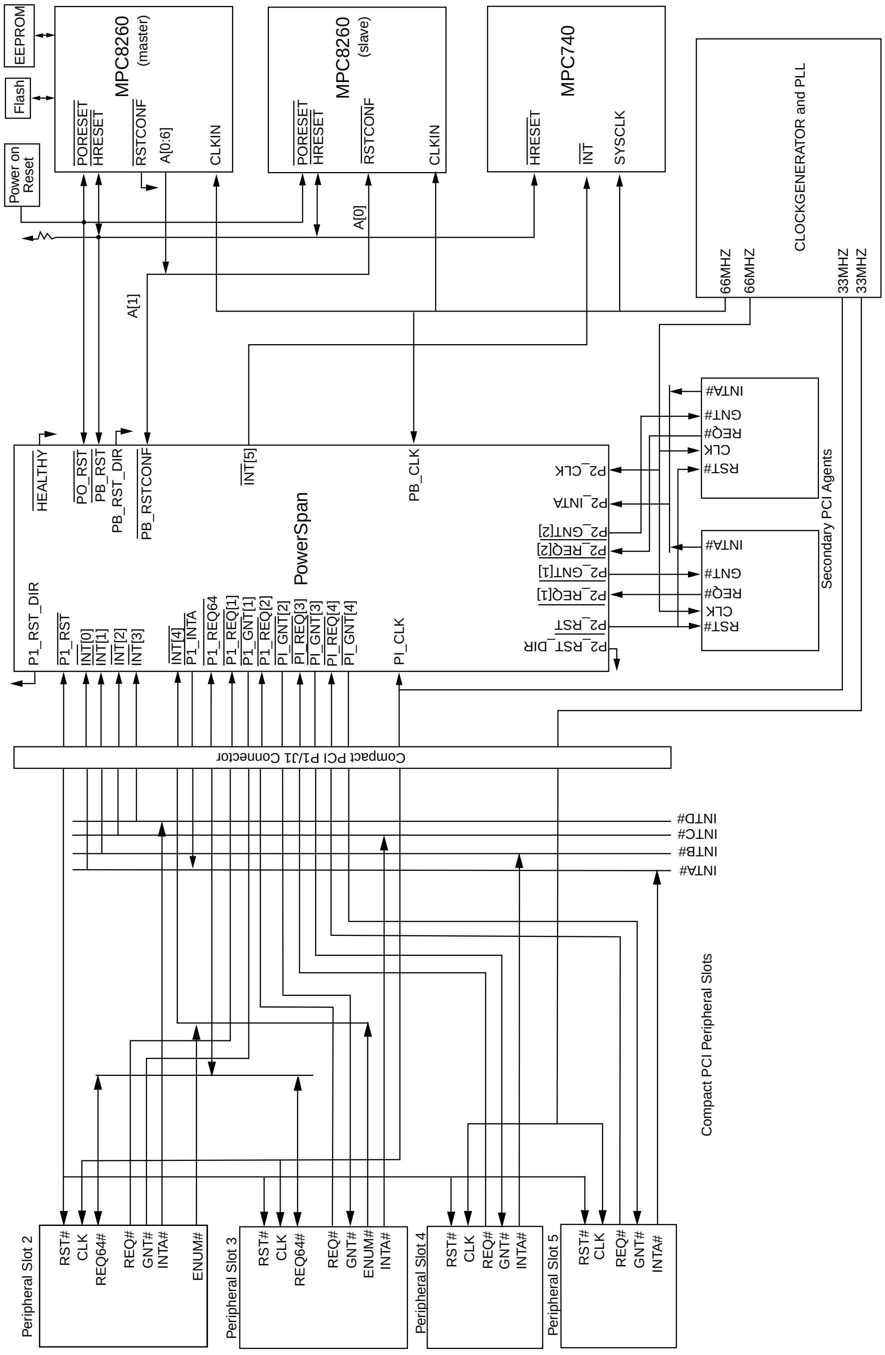
PowerSpan II has reset and arbitration functionality for the Primary and Secondary PCI bus segments. PB_RST_ is configured as an input while both P1_RST# and P2_RST# are outputs. A processor power-on reset or hard reset resets both Primary and Secondary PCI bus segments.

The card provides clocks for the embedded PowerPCs, PowerSpan II's PB Interface, all secondary PCI agents, as well as the multiple clocks required for the CompactPCI backplane.

PowerSpan II's bidirectional interrupt pins are used to handle all four CompactPCI interrupts and the Hot Swap system enumeration interrupt. All interrupts are routed to the MPC740 through PowerSpan II pin INT[5]_.

The MPC8260 system boots from local FLASH on the card. The MPC8260 uses PowerSpan II to configure all PCI agents on the Primary and Secondary PCI buses.

Figure 47: PowerSpan II in CompactPCI System Slot



Glossary

ATPG Automatic Test Pattern Generation.

BAR Base Address Register.

BD Buffer Descriptor. Each serial port in the MPC8260 uses BDs to indicate to the SDMA channel the location of packet data in system memory.

Big-endian A byte ordering method in memory where the address of a word corresponds to the most significant byte.

BIST Built-in Self Test.

CompactPCI CompactPCI is an adaptation of the PCI Specification for Industrial and/or embedded applications requiring a more robust mechanical form factor than desktop PCI.

Cycle Cycle refers to a single data beat; a transaction is composed of one or more cycles.

DDM Device Driver Module. A module that abstracts the service of an I/O device and registers it as an I2O Device.

Device An I/O object that refers to an I/O facility or service. Adapters are the objects of hardware configuration, while logical devices are the objects of software configuration.

DMA Direct Memory Access. A process for transferring data from main memory to a device without passing it through the Host processor.

DRAM Dynamic Random Access Memory.

Dual PCI PowerSpan II PowerSpan II variant with dual PCI interfaces.

FLASH Writable non-volatile memory, often used to store code in embedded systems.

Host Node A node composed of one or more application processors and their associated resources. Host nodes execute a single homogeneous operating system and are dedicated to processing applications. The host node is responsible for configuring and initializing the IOP into the system.

Host Operating System The control program executing on the host. This may be the BIOS code, the host bootstrap code, or the final operating system for application programs. Also called the host or OS.

Hot Swap Refers to the orderly insertion and extraction of CompactPCI boards without adversely affecting system operation.

I2C Inter-IC Bus

I2O Intelligent I/O

Inbound Queue A message queue of a particular I/O platform that receives messages from any sender (host or another IOP).

IOP I/O platform. A platform consisting of processor, memory, I/O adapters and I/O devices. They are managed independently from other processors within the system, solely for processing I/O transactions.

Little-endian A byte ordering method in memory where the address of a word corresponds to the least significant byte

Master Master (initiator) is the owner of the PCI bus. It is used for both the processor (60x) bus and the PCI bus.

MF Message Frame

MFA Message Frame Address

MPC8260 Motorola Integrated Communications Controller with 603ev PowerPC core and external bus

Outbound Queue A message queue for a specific I/O platform for posting messages to the local host, in lieu of the host’s Inbound Queue.

PowerPC 740 Embedded PowerPC processor.

PowerPC 750 Embedded PowerPC processor.

PowerPC 7400 Embedded PowerPC processor.

Prefetchable A range of *Memory* space is prefetchable if there are no side effects on reads.

Peripheral Slot Slots on a CompactPCI bus segment that may contain simple boards, intelligent slaves, or PCI bus masters.

PMC PCI Mezzanine Card. A mechanical format for adding PCI based mezzanine cards to a VMEbus card.

Primary PCI Interface Adds extra functionality to the PCI Interface that is designated as the Primary PCI Interface.The features are associated with the Primary PCI Interface include: CompactPCI Hot Swap Support, Vital Product Data support and I_2O Shell Interface.

SDMA Serial DMA. Used to support the movement of serial data to/from the serial ports of the MPC8260.

Secondary PCI Interface In the Dual PCI PowerSpan II, this interface is the interface that is not designated as the Primary PCI Interface.

Single PCI PowerSpan II PowerSpan II variant with a single PCI interface.

Slave Slave (target) is the device which is accessed by the bus master. It is used to refer to the address accessed by the master on the processor (60x) bus.

Strong ordering A memory access model that requires exclusive access to an address before making an update, to prevent another device from using stale data.

System Slot The slot on a CompactPCI bus segment that provides arbitration, clock distribution, and reset functions for all boards on the segment

Target Target (slave) is the device which is accessed by the bus master. It is used to refer to the address accessed by the master on the PCI bus.

Target image Target image is a memory range which is mapped between the PCI bus and the processor (60x) bus.

Weak ordering A memory access model that allows bus operations to be reordered dynamically. This improves overall performance and reduces the effect of memory latency on instruction throughput.

Index