PLX TECHNOLOGY®
THE I/O INTERCONNECT SOLUTION

PCI 6150BB Data Book

TECHNOLOGY®
THE I/O INTERCONNECT SOLUTION

PCI 6150BB Data Book

Version 2.11

February 2005

Website:	http://www.plxtech.com
Technical Support:	http://www.plxtech.com/support/
Phone:	408 774-9060
	800 759-3735
FAX:	408 774-2169

Order Number: PCI 6150-SIL-DB-P1-2.11; Former HiNT Part Number: HB4

Printed in the USA, February 2005

CONTENTS

FIGURES

TABLES

REGISTERS

PREFACE

The information contained in this document is subject to change without notice. Although an effort has been made maintain accurate information, there may be misleading or even incorrect statements made herein.

Supplemental Documentation

The following is a list of documentation to provide further details:

- *PCI Local Bus Specification, Revision 2.1*, June 1, 1995
  PCI Special Interest Group (PCI-SIG)
  5440 SW Westgate Drive #217, Portland, OR 97221 USA
  Tel: 503 291-2569, Fax: 503 297-1090, http://www.pcisig.com/home
- *PCI Local Bus Specification, Revision 2.3*
  PCI Special Interest Group (PCI-SIG)
  5440 SW Westgate Drive #217, Portland, OR 97221 USA
  Tel: 503 291-2569, Fax: 503 297-1090, http://www.pcisig.com/home
- *PCI to PCI Bridge Architecture Specification*, *Revision 1.1*
  PCI Special Interest Group (PCI-SIG)
  5440 SW Westgate Drive #217, Portland, OR 97221 USA
  Tel: 503 291-2569, Fax: 503 297-1090, http://www.pcisig.com/home
- *PCI Bus Power Management Interface Specification, Revision 1.1*, June 30, 1997
  PCI Special Interest Group (PCI-SIG)
  5440 SW Westgate Drive #217, Portland, OR 97221 USA
  Tel: 503 291-2569, Fax: 503 297-1090, http://www.pcisig.com/home
- *PICMG 2.1, R2.0, CompactPCI Hot Swap Specification,* January 2001
  PCI Industrial Computer Manufacturers Group (PICMG)
  c/o Virtual Inc., 401 Edgewater Place, Suite 500, Wakefield, MA 01880, USA
  Tel: 781 246-9318, Fax: 781 224-1239, http://www.picmg.org
- IEEE Standard 1149.1-1990, *IEEE Standard Test Access Port and Boundary-Scan Architecture*, 1990
  The Institute of Electrical and Electronics Engineers, Inc.
  445 Hoes Lane, PO Box 1331, Piscataway, NJ 08855-1331, USA
  Tel: 800 678-4333 (domestic only) or 732 981-0060, Fax: 732 981-1721, http://www.ieee.org/portal/index.jsp

Note: *In this data book, shortened titles are provided to the previously listed documents. The following table lists these abbreviations.*

Supplemental Documentation Abbreviations

Abbreviation	Document
PCI r2.3	*PCI Local Bus Specification, Revision 2.3*
P-to-P Bridge r1.1	*PCI to PCI Bridge Architecture Specification, Revision 1.1*
PCI Power Mgmt. r1.1	*PCI Bus Power Management Interface Specification, Revision 1.1*
PICMG 2.1 R2.0	*PICMG 2.1 R2.0 CompactPCI Hot Swap Specification*

DATA ASSIGNMENT CONVENTIONS

Data Assignment Conventions

Data Width	PCI 6150 Convention
1 byte (8 bits)	Byte
2 bytes (16 bits)	Word
4 bytes (32 bits)	DWORD/Dword
8 bytes (64 bits)	QWORD/Qword

REVISION HISTORY

Date	Version	Comments
5/02	1.2	Production Release, Silicon Revision BA1.
6/02	1.21	Added JTAG external pull up/low resistor requirement.
8/02	1.22	Updated DC characteristic table.
5/03	2.0	Production Release, Silicon Revision BB. This release reflects PLX part numbering. Added Silicon Revision BB to Section 2. Section 5, changed pin type in tables for P_SERR#, S_SERR#, S_REQ0#, S_GNT[8:1]#, S_M66EN, S_CLKO[9:0], and S_RSTOUT#. Removed Section 6.2, “Extended Register Map”. Removed Section 6.3.3, “Extended Registers”. Updated Configuration Map in Section 6.1 to reflect deletion of Extended Registers. Updated Register DEh, bits [15:11]. Added three notes to table in Section 14.5, “Frequency Division Options.” Updated Section 20.3.1, 24-3Fh.
9/04	2.1	Sampling Release, Silicon Revision BC. General enhancements to text to PLX standard. Update signal names to standard PLX signal names. Removed references to Non-Transparent mode. Included missing Subsystem Vendor and Device ID registers.
2/05	2.11	Changed Silicon Revision from BC back to BB. Updated to reflect *PCI Bus Power Management Interface Specification, Revision 1.1 compliance.* Removed “Address Translation” from Features section block diagram. Changed pin name of PIN_LED to PIN_LED/EJECT. Included eject switch-related information to pin description. Updated Pin Type column information in Section 3 pin tables for EEPCLK, ENUM#, GPIO3FN#, P_GNT#, P_IDSEL, PIN_LED/EJECT, P_LOCK#, P_M66EN, P_REQ#, P_RSTIN#, P_SERR#, S_GNT0#, S_GNT[8:1]#, S_RSTOUT#, and S_SERR#. Changed Bridge Control register, bit 4 (BCNTRL; PCI:3Eh) to ***Reserved***. Removed Subsystem Vendor and Device ID registers, as they are not supported in this silicon revision. Added Lead-Free ROHS-Compliant packaging information to Section B.2, “Package Ordering.”

THE I/O INTERCONNECT SOLUTION

PCI 6150

PCI-to-PCI Bridge

February 2005
Version 2.11

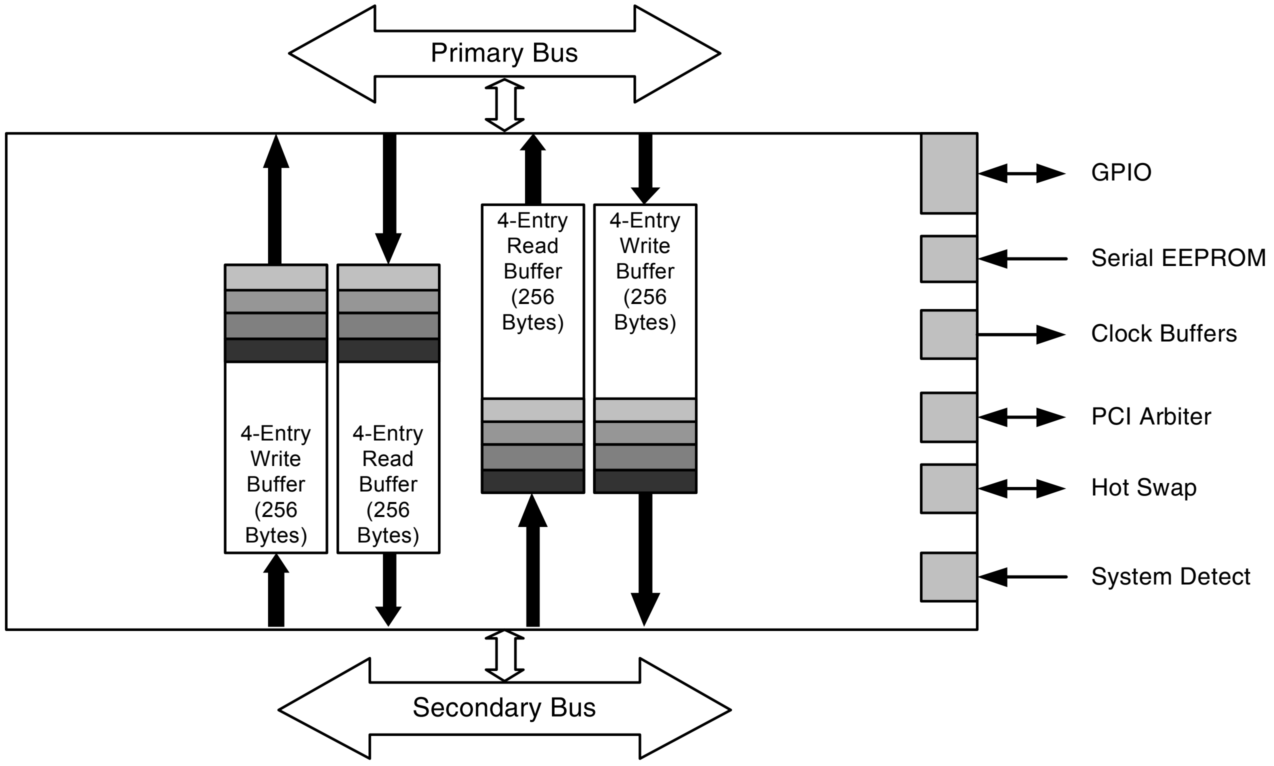
High-Performance, Asynchronous 32-Bit, 66 MHz PCI-to-PCI Bridge for Servers, Storage, Telecommunication, Networking and Embedded Applications

FEATURE SUMMARY

The PLX FastLane™ PCI 6150 32-bit, 66 MHz PCI-to-PCI bridge is designed for high-performance, programmable data rates and PLLs that can divide input clock frequencies up and down, which makes the device useful for bus expansion systems. The PCI 6150 also has Hot Swap capability, which makes it useful in high-availability systems. In addition, the PCI 6150 offers the largest data FIFO among all 32-bit PCI-to-PCI bridges in today's market, and sophisticated buffer management and configuration options, all of which are designed to customize performance optimization.

The PCI 6150 provides the following features and applications:

- *PCI r2.3* compliant
- 3.3V signaling, including 5V input signal tolerance and fast PCI buffers
- Provides 1 KB of buffering (data FIFO) to maximize performance
  - Upstream and downstream Posted Write buffers (256 bytes each)
  - Upstream and downstream Read Data buffers (256 bytes each)
  - Supports up to four simultaneous Posted Write transactions and four simultaneous Delayed transactions in each direction
  - Programmable prefetch of up to 256 bytes for maximum read performance optimization
  - Flow-through zero wait state burst up to 4 KB for large volume data transfer
  - Optional flow-through enable allows for customization
  - Fast back-to-back enable—Read-only supported
- Asynchronous design supports standard 66-to-33 MHz and faster secondary port speed, *such as* 33-to-66 MHz conversion
- Out-of-order Delayed transactions
- Enhanced address decoding
  - 32-bit I/O Address range
  - 32-bit Memory-Mapped I/O Address range
  - ISA Aware mode for legacy support in the first 64 KB of I/O Address range
  - VGA addressing and palette snooping support
- Address Stepping hardcoded to two clocks
- Ten secondary Clock outputs with pin-controlled enable and individual maskable control to nine bus masters on secondary interface support
- External arbiter or programmable arbitration for up to nine bus masters on secondary interface support
- Hot Swap *Ready*
- *PICMG 2.1 R2.0* with PI=1
  - Support for device hiding, eliminating mid-transaction extraction problems
- PCI Mobile Design Guide and Power Management D_{3cold} Wakeup capable with PME# support
- Four GPIO pins with output control and power-up status latch capable
- Serial EEPROM loadable and programmable PCI Read-Only register configurations
  - Serial EEPROM Load modification and recheck
  - VPD support
- IEEE Standard 1149.1-1990 JTAG interface for boundary scan test
- Multiple IDs check all Device and Revision IDs
- Industry-standard 208-pin Plastic Quad Flat Pack (PQFP) or 256-pin (ball) Plastic Ball Grid Array (PBGA) package


Primary Bus
4-Entry Write Buffer (256 Bytes)
4-Entry Read Buffer (256 Bytes)
4-Entry Read Buffer (256 Bytes)
4-Entry Write Buffer (256 Bytes)
GPIO
Serial EEPROM
Clock Buffers
PCI Arbiter
Hot Swap
System Detect
Secondary Bus

PCI 6150 Block Diagram

1 INTRODUCTION

This section provides information about PLX Technology, Inc., and its products, the FastLane™ PCI 6000 Bridge Series, and PCI 6150 features and applications.

1.1 COMPANY AND PRODUCT INFORMATION

PLX Technology, Inc., is the leading supplier of standard interconnect silicon to the storage, communications, server, and embedded-control industries. PLX's comprehensive I/O interconnect product offering ranges from I/O accelerators, PCI-to-PCI bridges, PCI-X-to-PCI-X bridges, and HyperTransport™ bridges to the PLX PCI Express-based family of switches and bridges currently under development.

In addition to a broad product offering, PLX provides development tool support through Software Development Kits (SDKs), hardware Rapid Development Kits (RDKs), and third-party tool support through the PLX Partner Program. Our complete tool offering, combined with leadership PLX silicon, enables system designers to maximize system throughput, lower development costs, minimize system design risk, and provide faster time to market.

The PLX commitment to meeting customer requirements extends beyond complete product solutions, and includes active participation in industry associations. PLX contributes to the key standard-setting bodies in our industry, including PCI-SIG™ (the special interest group responsible for the creation and release of all PCI specifications), PICMG® (the organization responsible for the new AdvancedTCA™ standard for fabrics), HyperTransport™ Consortium, and Blade Systems Alliance (BladeS). Furthermore, PLX is a key developer for PCI Express technology and a member of the Intel® Developers Network for PCI Express Technology.

Founded in 1986, PLX has been developing products based on the PCI industry standard since 1994. PLX is publicly traded (NASDAQ:PLXT) and headquartered in Sunnyvale, CA, USA, with other domestic offices in Utah and Southern California. PLX European operations are based in the United Kingdom and Asian operations are based in China and Japan.

1.2 FASTLANE PCI 6000 BRIDGE SERIES

The PLX FastLane PCI 6000 series offers the industry's broadest set of PCI-to-PCI and PCI-X-to-PCI-X bridges. These bridges allow additional devices to be attached to the PCI Bus, and provide the ability to include intelligent adapters on a PCI Bus. In addition, these bridges allow PCI Buses of different speeds to be part of the same subsystem. (Refer to Table 1-1 and Figure 1-1.)

The PLX PCI and PCI-X family of interconnect products include both PCI-to-PCI and PCI-X-to-PCI-X bridging devices, offering system designers innovative features along with improved I/O performance. The PLX FastLane PCI 6000 series of PCI-to-PCI bridging products provide support for the entire range of current PCI Bus data widths and speeds, including 32-bit 33 MHz, 64-bit 66 MHz, and the latest 64-bit 133 MHz PCI-X variety of the standard.

The FastLane PCI 6000 product line is distinguished by featuring the widest range of options, lowest power requirements, highest performance, and smallest footprint in the industry. The product line includes features such as the ability to clock the PCI Bus segments asynchronously to one another.

The entire line of PLX bridging products are designed to provide high-performance interconnect for servers, storage, telecommunications, networking, and embedded applications. Like all PLX interconnect chips, the FastLane PCI 6000 series products are supported by PLX comprehensive reference design tools and the industry-recognized PLX support infrastructure.

Table 1-1. FastLane PCI 6000 Series PCI and PCI-X Bridge Product Comparison

Features	PCI 6140-AA33PC	PCI 6150-BB66BC PCI 6150-BB66PC	PCI 6152-CC33BC PCI 6152-CC33PC	PCI 6152-CC66BC	PCI 6156-DA33PC
PCI Bus Type	32-bit 33 MHz PCI	32-bit 66 MHz PCI	32-bit 33 MHz PCI	32-bit 66 MHz PCI	32-bit 33 MHz PCI
PCI Local Bus Support	r2.1 compliant	r2.3 compliant	r2.2 compliant	r2.2 compliant	r2.2 compliant
3.3 and 5V Tolerant I/O	Yes	Yes	Yes	Yes	Yes
Asynchronous Operation	No	25 to 66 MHz	No	No	No
Power Dissipation	200 mW	1.8W	300 mW	300 mW	300 mW
GPIO Interface	No	Four GPIO Pins	Four GPIO Pins	Four GPIO Pins	No
Transparency Modes	Transparent only	Transparent only	Transparent only	Transparent only	Transparent only
CompactPCI-Compatible Hot Swap	*Friendly*	r2.0 with PI=1	*Friendly*	*Friendly*	—
Data FIFO	—	1 KB	—	—	—
Number of Bus Masters on Secondary Bus	Up to 4	Up to 9	Up to 4	Up to 4	Up to 10
Retry Architecture	Standard	Standard	Performance-Optimized	Performance-Optimized	Performance-Optimized
Programmable Flow-Through	—	Yes	—	—	—
Programmable Prefetch	Not specified	Up to 4 KB	N/A	N/A	N/A
Zero Wait State Burst	Up to 1 KB	Up to 1 KB	Up to 1 KB	Up to 1 KB	Up to 1 KB
Serial EEPROM Support	—	Yes	Yes	Yes	Yes
Vital Product Data Registers	—	Yes	Yes	Yes	Yes
D_3 Wakeup Power Management	Yes	Yes	Yes	Yes	Yes
Secondary Clock Outputs	Yes	Yes	Yes	Yes	Yes
JTAG Support	—	IEEE 1149.1 compliant	—	—	—
Packaging	PQFP-128	PBGA-256 PQFP-208	Tiny BGA-160 PQFP-160	Tiny BGA-160	PQFP-208
Package Size	23 x 17 mm	17 x 17 mm 31 x 31 mm	15 x 15 mm 32 x 32 mm	15 x 15 mm	31 x 31 mm
Rapid Development Kit	PCI 6140RDK	PCI 6150RDK	PCI 6152RDK	PCI 6152RDK	PCI 6156RDK

Table 1-1. FastLane PCI 6000 Series PCI and PCI-X Bridge Product Comparison (Continued)

Features	PCI 6350-AA66PC	PCI 6154-BB66BC	PCI 6254-BB66BC	PCI 6520-XX	PCI 6540-XX
PCI Bus Type	32-bit 66 MHz PCI	64-bit 66 MHz PCI	64-bit 66 MHz	64-bit 133 MHz PCI-X	64-bit 133 MHz PCI-X
PCI Local Bus Support	r2.3 compliant	r2.3 compliant	r2.3 compliant	r2.3 compliant	r2.3 compliant
3.3 and 5V Tolerant I/O	Yes	Yes	Yes	Yes	Yes
Asynchronous Operation	Yes	25 to 66 MHz	25 to 66 MHz	33 to 133 MHz	25 to 133 MHz
Power Dissipation	1.47W	2.0W	2.0W	1.0W	1.0W
GPIO Interface	Four GPIO Pins	Four GPIO Pins	16 GPIO Pins	8 GPIO Pins	16 GPIO Pins
Transparency Modes	Transparent only	Transparent only	Transparent, Non-Transparent and Universal modes	Transparent only	Transparent, Non-Transparent and Universal modes
CompactPCI-Compatible Hot Swap	—	—	r2.0 with PI=1	—	r2.0 with PI=1
Data FIFO	192 byte	1 KB	1 KB	10 KB	10 KB
Number of Bus Masters on Secondary Bus	Up to 9	Up to 9	Up to 9	Up to 8	Up to 8
Retry Architecture	Standard	Standard	Standard	Standard	Standard
Programmable Flow-Through	Yes	Yes	Yes	Yes	Yes
Programmable Prefetch	Up to 2 KB	Up to 4 KB	Up to 4 KB	Up to 4 KB	Up to 4 KB
Zero Wait State Burst	Up to 4 KB	Up to 1 KB	Up to 1 KB	Up to 4 KB	Up to 4 KB
Serial EEPROM Support	Yes	Yes	Yes	Yes	Yes
Vital Product Data Registers	Yes	Yes	Yes	Yes	Yes
D_3 Wakeup Power Management	Yes	Yes	Yes	Yes	Yes
Secondary Clock Outputs	Yes	Yes	Yes	Yes	Yes
JTAG Support	IEEE 1149.1 compliant	IEEE 1149.1 compliant	IEEE 1149.1 compliant	IEEE 1149.1 compliant	IEEE 1149.1 compliant
Packaging	PBGA-256 PQFP-208	PBGA-304	PBGA-365	PBGA-380	PBGA-380
Package Size	17 x 17 mm 31 x 31 mm	31 x 31 mm	31 x 31 mm	27 x 27 mm	27 x 27 mm
Rapid Development Kit	PCI 6350RDK	PCI 6154RDK	PCI 6254RDK	PCI 6520RDK	PCI 6540RDK

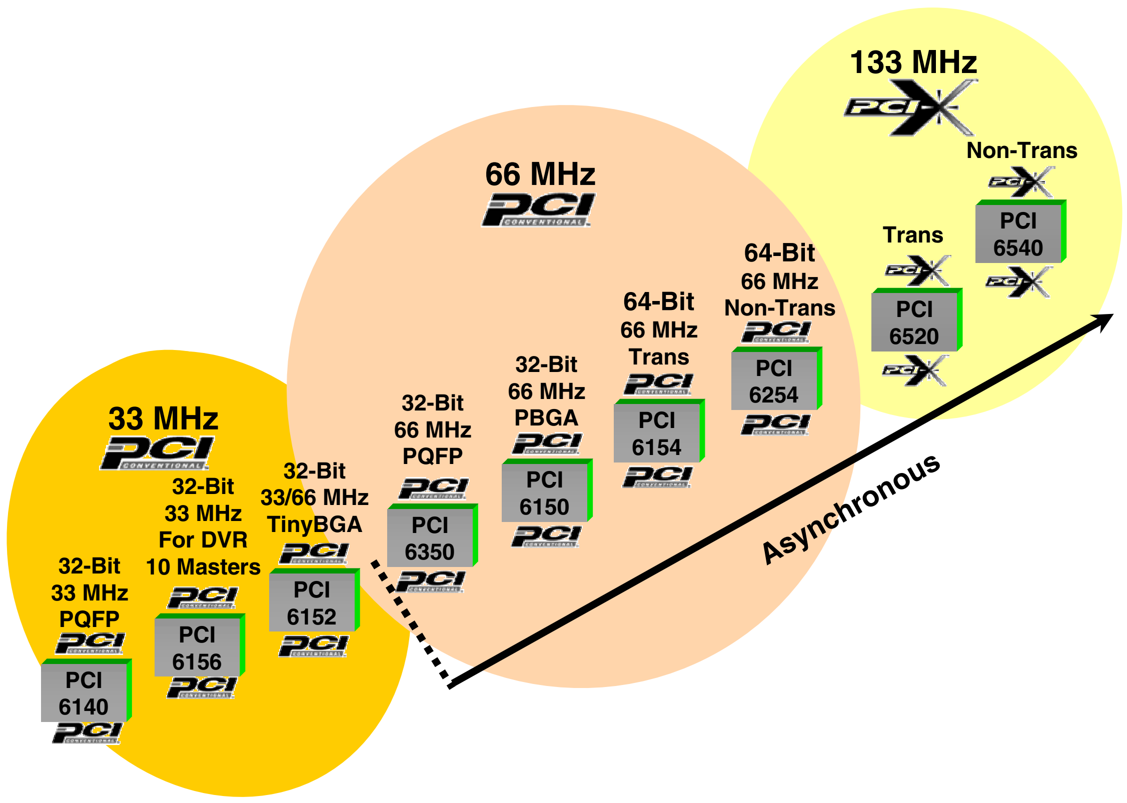


Figure 1-1. FastLane PCI 6000 Bridge Series

1.2.1 PCI 6150

The PCI 6150 is the most powerful PCI-to-PCI bridging device offered in the industry. As illustrated in Figure 1-2, the PCI 6150 is a two-port device providing a*synchronous* operation between the *primary* and *secondary* ports.

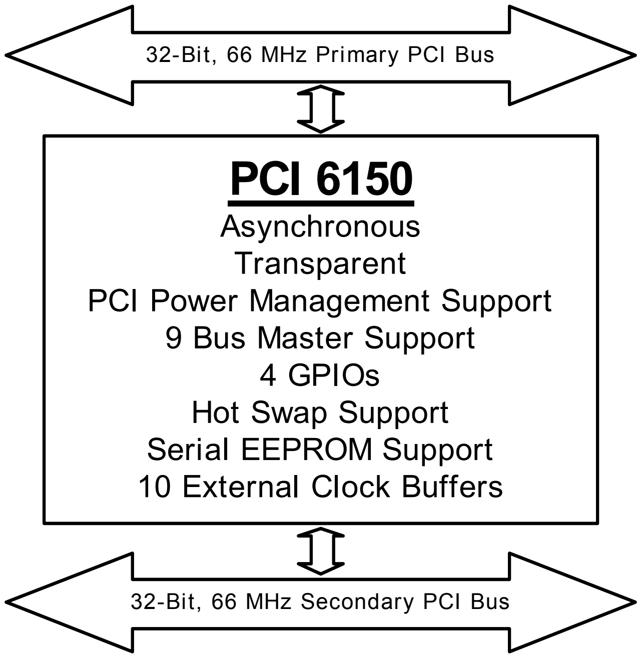

32-Bit, 66 MHz Primary PCI Bus
PCI 6150
Asynchronous
Transparent
PCI Power Management Support
9 Bus Master Support
4 GPIOs
Hot Swap Support
Serial EEPROM Support
10 External Clock Buffers
32-Bit, 66 MHz Secondary PCI Bus

Figure 1-2. PCI 6150 PCI-to-PCI Bridge

1.3 FEATURE DESCRIPTION

The PCI 6150 is built upon the powerful PLX PCI-to-PCI Bridge Architecture, and offers the largest data FIFO among all 32-bit PCI-to-PCI bridges in today's market. The PCI 6150 provides the following features and applications:

- *PCI r2.3* compliant
- 3.3V signaling, including 5V input signal tolerance and fast PCI buffers
- Provides 1 KB of buffering (data FIFO) to maximize performance
    - Upstream and downstream Posted Write buffers (256 bytes each)
    - Upstream and downstream Read Data buffers (256 bytes each)
    - Supports up to four simultaneous Posted Write transactions and four simultaneous Delayed transactions in each direction
    - Programmable prefetch of up to 256 bytes for maximum read performance optimization
    - Flow-through zero wait state burst up to 4 KB for large volume data transfer
    - Optional flow-through enable allows for customization
    - Fast back-to-back enable—Read-only supported
- Asynchronous design supports standard 66-to-33 MHz and faster secondary port speed, *such as* 33-to-66 MHz conversion
- Out-of-order Delayed transactions
- Enhanced address decoding
    - 32-bit I/O Address range
    - 32-bit Memory-Mapped I/O Address range
    - ISA Aware mode for legacy support in the first 64 KB of I/O Address range
    - VGA addressing and palette snooping support
- Address Stepping hardcoded to two clocks
- Ten secondary Clock outputs with pin-controlled enable and individual maskable control to nine bus masters on secondary interface support
- External arbiter or programmable arbitration for up to nine bus masters on secondary interface support
- Hot Swap *Ready*
- *PICMG 2.1 R2.0* with PI=1
    - Support for device hiding, eliminating mid-transaction extraction problems
- PCI Mobile Design Guide and Power Management D_{3cold} Wakeup capable with PME# support
- Four GPIO pins with output control and power-up status latch capable
- Serial EEPROM loadable and programmable PCI Read-Only register configurations
    - Serial EEPROM Load modification and recheck
    - VPD support
- IEEE Standard 1149.1-1990 JTAG interface for boundary scan test
- Multiple IDs check all Device and Revision IDs
- Industry-standard 208-pin Plastic Quad Flat Pack (PQFP) or 256-pin (ball) Plastic Ball Grid Array (PBGA) package

1.4 APPLICATION—MULTIPLE DEVICE EXPANSION

Figure 1-3 illustrates the PCI 6150 being used to provide electrical isolation to the PCI Bus. This is necessary because PCI slots restrict the number of loads that can be accommodated. The devices on the secondary port must be *PCI*, and the bus must operate at 32-bit, up to 66 MHz. This configuration is a common mechanism for providing multiple PCI devices on a single bus without exceeding the bus load limitation as defined in *PCI r2.3*.

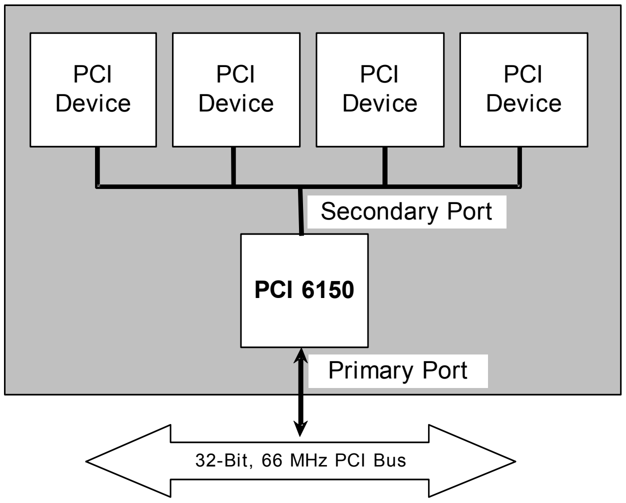


Figure 1-3. Multiple Device Expansion

2 FUNCTIONAL OVERVIEW

This section describes general operation of the PCI 6150 bridge, and provides an overview of write and read transactions.

2.1 GENERAL OPERATION

Each PCI port can run at different (asynchronous) frequencies, up to 66 MHz, which allows the designer to optimize the performance of each bus.

The PCI 6150 provides an Internal Arbiter function on the secondary bus, for up to nine secondary bus masters. However, the Internal Arbiter may be disabled if an External Arbiter is used. The PCI 6150 also sources ten secondary PCI clock outputs.

The PCI 6150 supports a serial EEPROM device for register configuration data. This allows the PCI 6150 to automatically load custom configuration upon power-up, which minimizes the software overhead of configuring the bridge through a host processor.

The PCI 6150 provides features satisfying the requirements of *PCI Power Mgmt. r1.1*, supporting Power Management states D_0 through D_{3cold} and D_{3hot}. (Refer to Section 18, "Power Management," for further details.)

The PCI 6150 is CompactPCI Hot Swap *Ready*, and complies with *PICMG 2.1 R2.0* with *High Availability* Programming Interface level 1 (PI=1). (Refer to Section 19, "Hot Swap," for further details.)

The PCI 6150 fully supports Vital Product Data (VPD) by providing the Address, Data, and Control registers (PVPDAD; PCI:EAh, PVPDATA; PCI:ECh, PVPDID; PCI:E8h, and PVPD_NEXT; PCI:E9h) for accessing VPD stored in the unused portion of the serial EEPROM. VPD allows reading or writing of user data to the upper 192 bytes of serial EEPROM space, and that data can contain information such as board serial number, software revision, firmware revision, or other data required for non-volatile storage. (Refer to Section 20, "VPD," for further details.)

2.2 WRITE TRANSACTIONS

The primary or secondary bus accomplishes a Write operation by placing the address and data into the *Write buffer*. This initiates a PCI Write operation on the other bus. The Write operation is called a *Posted Write* operation, because the initiating bus performs the write, then moves on without waiting for the operation to complete.

In addition, the PCI 6150 has the capability to start a Write operation before receiving all Write data. In this case, the Write operation begins when there is sufficient Write data to begin the burst, providing a Flow-Through operation as the balance of the Write data arrives in the device.

2.3 READ TRANSACTIONS

When the downstream or upstream bus needs to read data from the other bus, the bus places the Read request into the *Read Command queue*. This initiates a Read operation on the other bus, and the data is placed into the associated *Read buffer* as it returns.

For PCI transactions, there is an additional prefetch mechanism when returning the requested Read data. In this mode, the PCI 6150 can be programmed to prefetch up to 1 KB of data at a time. This data is stored in the Read buffer and is not flushed until the buffer times out. If requested, prefetched data can be delivered to the PCI Bus without the normal read on the other bus.

3 PIN DESCRIPTION

This section describes the PCI 6150 pins, including pin summary, pull-up and pull-down resistor recommendations, and pinout listings.

Note: *In this data book, the PBGA balls are also referred to as pins.*

3.1 PIN SUMMARY

Tables 3-4 through Table 3-8 and Table 3-10 through Table 3-13 describe each PCI 6150 pin:

- Primary PCI Bus Interface
- Secondary PCI Bus Interface
- Clock-Related
- Reset
- CompactPCI Hot Swap
- JTAG
- Serial EEPROM Interface
- Miscellaneous
- Power, Ground, and Reserved

For a visual view of the PCI 6150 pinout, refer to Section 22, “Mechanical Specs.”

Table 3-1 lists abbreviations used in Section 3 to represent various pin types.

Table 3-1. Pin Type Abbreviations

Abbreviation	Pin Type
I	CMOS Input (5V input tolerant, I/O V_{DD}=3.3V).
I/O	CMOS Bi-Directional Input Output (5V input tolerant, I/O V_{DD}=3.3V).
O	CMOS Output.
PCI	PCI Compliant.
PD	Internally pulled down.
PU	Internally pulled up.
PI	PCI Input (5V input tolerant, I/O V_{DD}=3.3V).
PO	PCI Output.
PSTS	PCI Sustained Three-State Output. Active low signal which must be driven inactive for one cycle before being three-stated to ensure high performance on a shared signal line.
PTS	PCI Three-State Bi-Directional (5V input tolerant, I/O V_{DD}=3.3V).

3.2 PULL-UP AND PULL-DOWN RESISTOR RECOMMENDATIONS

Pull-up and pull-down resistor values are not critical. With the exception of those mentioned in Section 3.2.1, a 10K-Ohm resistor is recommended unless stated otherwise.

3.2.1 PCI Bus Interface Pins

The pins detailed in Table 3-2 are generic primary and secondary PCI interface pins. When producing motherboards, system slot cards, adapter cards, backplanes, and so forth, the termination of these pins should follow the guidelines detailed in *PCI r2.3*.

Table 3-2. Generic PCI Bus Interface Pins that follow *PCI r2.3* Layout Guidelines

Bus	Pin Name
Primary	P_AD[31:0], P_CBE[3:0]#, P_DEVSEL#, P_FRAME#, P_GNT#, P_IDSEL, P_IRDY#, P_LOCK#, P_M66EN, P_PAR, P_PERR#, P_REQ#, P_SERR#, P_STOP#, P_TRDY#
Secondary	S_AD[31:0], S_CBE[3:0]#, S_DEVSEL#, S_FRAME#, S_GNT[8:0]#, S_IRDY#, S_LOCK#, S_M66EN, S_PAR, S_PERR#, S_REQ[8:0]#, S_SERR#, S_STOP#, S_TRDY#

The following guidelines are not exhaustive and should be read in conjunction with the appropriate sections of *PCI r2.3*.

PCI control signals require a pull-up resistor on the motherboard to ensure that these signals are always at valid values when a PCI Bus agent is not driving the bus. These control signals include DEVSEL#, FRAME#, IRDY#, LOCK#, PERR#, SERR#, STOP#, and TRDY#. The point-to-point and shared bus signals do ***not*** require pull-up resistors, as bus parking ensures that these signals remain stable. The value of these pull-up resistors depends on the bus loading. *PCI r2.3* provides formulas for calculating these resistors.

When making adapter cards in which the PCI 6150 primary port is wired to the PCI connector, pull-up resistors are not required because they are pre-installed on the motherboard.

Based on the above, in an embedded design, pull-up resistors may be required for PCI control signals on the primary and secondary buses. Whereas, for a PCI adapter card design, pull-up resistors are required only on the PCI 6150 port that is not connected to the motherboard or host system.

S_M66EN ***must*** be pulled high or low with a 10K-Ohm resistor.

The S_REQ[8:1]# inputs must be pulled high with a 10K-Ohm pull-up resistor to V_{DD}. S_REQ0# also requires a 10K-Ohm pull-up resistor to V_{DD} if S_CFN#=0.

Pull S_GNT[8:1]# high if S_CFN#=1.

3.2.2 Clock-Related Pins

Clock routing is detailed in Section 4, “Clocking.” Pull-up resistors are not required on the S_CLKO[9:0] pins; however, a series termination resistor is required when using these pins. S_CLKO0 may require a pull-up resistor when this pin is disabled (SCLKCNTRL[1:0]=11b; PCI:68h). S_CLKO[9:0] may also require pull-up resistors if they are disabled by pulling MSK_IN high. Table 3-3 delineates the remaining clock pin resistor requirements.

Table 3-3. Clock Pin Pull-Up/Pull-Down Resistor Requirements

Resistor Requirements	Pin Name
Pull high or low if unused	OSCIN
Optionally pull high or low	MSK_IN**, OSCSEL#
Pull-up or pull-down resistor not required	P_CLKIN, S_CLKIN, S_CLKO[9:0]*

Notes: ** Refer also to the text preceding this table.*

*** MSK_IN is used in the PQFP package only.*

3.2.3 Reset Pins

The P_RSTIN# Reset signal may require a pull-up resistor, depending on the application. The S_RSTOUT# Reset signal does ***not*** require a pull-up nor pull-down resistor.

3.2.4 CompactPCI Hot Swap Pins

If Hot Swap is used, pull the EJECT_EN# and GPIO3FN# pins low. The GPIO3 pin is then used as an EJECT input for Hot Swap. PIN_LED/EJECT is then connected to an external LED. If EJECT_EN# is not used, it must be at logic 0 and pulled low.

If Hot Swap is not used, tie GPIO3FN# high or GPIO3 low. EJECT_EN# is don't care.

3.2.5 JTAG Pins

The TCK, TDI, and TMS JTAG signals must be pulled high or low to a known state, using an external resistor. TRST# must be pulled low, using a 330-Ohm resistor.

The TDO signal does not require a pull-up nor pull-down resistor.

3.2.6 Serial EEPROM Pins

If a serial EEPROM is used, EE_EN# requires a pull-down resistor. If a serial EEPROM is not used, pull up EE_EN# to disable serial EEPROM autoload during system boot-up.

EEPCLK does ***not*** require a pull-up nor pull-down resistor. EEPDATA requires an external pull-up resistor.

3.2.7 Miscellaneous Pins

The BPCC_EN signal may optionally be pulled high or low. S_CFN# may also optionally be pulled high or low, but must be tied low to use the Internal Arbiter.

When programmed as outputs, the GPIO[3:0] pins do not require external pull-up nor pull-down resistors. If configured as inputs, pull the GPIO[3:0] pins high or low, depending on the application. If Hot Swap is not used, tie GPIO3FN# high or GPIO3 low. When pulled high, CFG66 enables the PCI 6150 to declare 66 MHz capability.

3.2.7.1 System Voltage Pins

For designs and add-in cards that have an independent V_{IO} voltage source, and for which proper power sequencing cannot be guaranteed, the current between the V_{IO} voltage source and PCI 6150 V_{IO} pins must be limited to protect the device from long-term undue stress resulting in damage (*such as* from resistor insertion).

Note: *By their nature, add-in cards cannot assume proper power sequencing and requirements must be met by system power supplies.*

Use the following guidelines to determine the required resistance value for the P_V_{IO} and S_V_{IO} pins:

- **3.3V signaling environments**—40 to 200-Ohm resistance between the V_{IO} voltage source and the PCI 6150 V_{IO} pins is recommended if V_{IO} is a maximum of 3.6V
- **3.3 or 5V signaling environments**—40 to 70-Ohm resistance is recommended

A single resistor can be used if the V_{IO} pins are bused, or multiple parallel resistors can be used between the V_{IO} voltage source and V_{IO} pins. The resistor power dissipation rating depends upon the resistance size and signaling environment. *For example,* if a single 50-Ohm resistor is used in a 5V signaling environment, the worst-case power dissipation would result in 480 mW. (Refer to Figure 3-1.)

If four, 200-Ohm resistors are used in parallel, each would be required to dissipate 120 mW.

Any resistance value within the recommended ranges prevents the device from being damaged, while providing sufficient clamping action to keep the Input Voltage (V_{IN}) below its maximum rating. A resistance value at the lower end of the range is recommended to provide preferable clamping action, and a sufficient V_{IN} margin.

$$480\text{ mW} = \frac{(V \bullet V) / R\ (5.5V\text{ (maximum signal amplitude, plus 10\%)} - 0.6V\text{ (1 diode drop)})^2}{50\text{ Ohms}}$$

Figure 3-1. Worst-Case Power Dissipation Example

3.3 PINOUT

Note: *Refer to Section 3.2 for pull-up and pull-down resistor recommendations not specifically stated in these tables.*

Table 3-4. Primary PCI Bus Interface Pins

Symbol	Signal Name	Total Pins	Pin Type	PQFP Pin Number	PBGA Pin Number	Function
P_AD[31:0]	Primary Address and Data	32	I/O PTS PCI	49, 50, 55, 57, 58, 60, 61, 63, 67, 68, 70, 71, 73, 74, 76, 77, 93, 95, 96, 98, 99, 101, 107, 109, 112, 113, 115, 116, 118, 119, 121, 122	N3, T2, T4, N5, P5, T5, N6, R5, T6, P7, T7, R7, T8, P8, R8, T9, R12, P12, T14, R13, N12, T15, P16, N15, M14, M13, M15, L13, M16, L14, L15, L16	Multiplexed Address and Data Bus. Address is indicated by P_FRAME# assertion during PCI transactions. Write data is stable and valid when P_IRDY# is asserted and Read data is stable and valid when P_TRDY# is asserted. Data is transferred on rising clock edges when P_IRDY# and P_TRDY# are asserted. During bus idle, the PCI 6150 drives P_AD[31:0] to valid logic levels when P_GNT# is asserted. (Refer to Section 13, "PCI Bus Arbitration," for further details.)
P_CBE[3:0]#	Primary Command and Byte Enables	4	I/O PTS PCI	64, 79, 92, 110	R6, R9, T13, N16	Multiplexed Command and Byte Enable fields. Provides the transaction type during the PCI Address phase. In the Data phase of PCI Memory Write transactions, P_CBE[3:0]# provide Byte Enables. During bus idle, the PCI 6150 drives P_CBE[3:0]# to valid logic levels when P_GNT# is asserted. (Refer to Section 13, "PCI Bus Arbitration," for further details.)
P_DEVSEL#	Primary Device Select	1	I/O PSTS PCI	84	P10	Asserted by the target, indicating that the device is accepting the transaction. As a master, the PCI 6150 waits for P_DEVSEL# assertion within five cycles of P_FRAME# assertion; otherwise, the transaction terminates with a Master Abort. Before being placed into a high-impedance state, P_DEVSEL# is driven to a high state for one cycle.
P_FRAME#	Primary Frame	1	I/O PSTS PCI	80	P9	Driven by the initiator of a transaction to indicate the beginning and duration of an access. P_FRAME# de-assertion indicates the final Data phase requested by the initiator. Before being placed into a high-impedance state, P_FRAME# is driven to a high state for one cycle.
P_GNT#	Primary Grant	1	I PI	46	R1	When asserted, the PCI 6150 can access the primary bus. During bus idle with P_GNT# asserted, the PCI 6150 drives P_AD[31:0], P_CBE[3:0]#, and P_PAR to valid logic levels. (Refer to Section 13, "PCI Bus Arbitration," for further details.)

Table 3-4. Primary PCI Bus Interface Pins (Continued)

Symbol	Signal Name	Total Pins	Pin Type	PQFP Pin Number	PBGA Pin Number	Function
P_IDSEL	Primary Initialization Device Select	1	I PI	65	P6	Used as a Chip Select line for Type 0 Configuration accesses to PCI 6150 Configuration space.
P_IRDY#	Primary Initiator Ready	1	I/O PSTS PCI	82	T10	Driven by the initiator of a transaction to indicate its ability to complete the current Data phase on the primary bus. Once asserted in a Data phase, P_IRDY# is not de-asserted until the end of the Data phase. Before being placed into a high-impedance state, P_IRDY# is driven to a de-asserted state for one cycle.
P_LOCK#	Primary Lock	1	I/O PSTS	87	R11	Asserted by the bus master, indicating an atomic operation that may require multiple transactions to complete. Primary lock asserted by master for multiple transactions to complete. If lock function is not needed, when no secondary PCI devices support lock, pull high and do not connect to the PCI Bus. Can be disabled by setting MSCOPT[13]=0; PCI:46h.
P_M66EN	Primary 66 MHz Enable	1	I PI	102	R14	Set high to allow 66 MHz primary bus operation. Along with S_M66EN, controls the frequency output to the S_CLKO[9:0] pins. (Refer to Section 4, “Clocking,” for further details.)
P_PAR	Primary Parity	1	I/O PTS PCI	90	N11	Parity is even across P_AD[31:0], P_CBE[3:0]#, and P_PAR [*that is,* an even number of ones (1)]. P_PAR is an input, and is valid and stable for one cycle after the Address phase (indicated by P_FRAME# assertion) for address parity. For Write Data phases, P_PAR is an input and valid one clock after P_IRDY# assertion. For Read Data phases, P_PAR is an output and valid one clock after P_TRDY# assertion. P_PAR is placed into a high-impedance state one cycle after the P_AD[31:0] lines are placed into a high-impedance state. During bus idle, the PCI 6150 drives P_PAR to a valid logic level when P_GNT# is asserted.
P_PERR#	Primary Parity Error	1	I/O PSTS PCI	88	T12	Asserted when a Data Parity error is detected for data received on the primary interface. Before being placed into a high-impedance state, P_PERR# is driven to a de-asserted state for one cycle.

Table 3-4. Primary PCI Bus Interface Pins (Continued)

Symbol	Signal Name	Total Pins	Pin Type	PQFP Pin Number	PBGA Pin Number	Function
P_REQ#	Primary Request	1	O PO	47	P2	Asserted by the PCI 6150 to request ownership of the primary bus to perform a transaction. The PCI 6150 de-asserts P_REQ# for at least two PCI Clock cycles before re-asserting it. (Refer to Section 13.2, “Primary PCI Bus Arbitration,” for further details.)
P_SERR#	Primary System Error	1	I/O PTS	89	P11	P_SERR# can be driven low by any device to indicate a System error condition. The PCI 6150 drives P_SERR# if one of the following conditions is met: • Address Parity error • Posted Write Data Parity error on target bus • S_SERR# is asserted • Master Abort during Posted Write transaction • Target Abort during Posted Write transaction • Posted Write transaction discarded • Delayed Write request discarded • Delayed Read request discarded • Delayed transaction Master Timeout Pull-up P_SERR# through an external resistor.
P_STOP#	Primary Stop	1	I/O PSTS PCI	85	T11	Asserted by the target to end the transaction on the current Data phase. Before being placed into a high-impedance state, P_STOP# is driven to a de-asserted state for one cycle.
P_TRDY#	Primary Target Ready	1	I/O PSTS PCI	83	R10	Driven by the target of a transaction to indicate its ability to complete the current Data phase on the primary bus. Before being placed into a high-impedance state, P_TRDY# is driven to a de-asserted state for one cycle.
Total		49				

Table 3-5. Secondary PCI Bus Interface Pins

Symbol	Signal Name	Total Pins	Pin Type	PQFP Pin Number	PBGA Pin Number	Function
S_AD[31:0]	Secondary Address and Data	32	I/O PTS PCI	206, 204, 203, 201, 200, 198, 197, 195, 192, 191, 189, 188, 186, 185, 183, 182, 165, 164, 162, 161, 159, 154, 152, 150, 147, 146, 144, 143, 141, 140, 138, 137	A4, D5, C5, A5, B5, D6, A6, C6, C7, A7, B7, C8, A8, B8, A9, C9, C12, D12, A14, B13, A15, B16, E13, C16, E14, D16, F13, E16, F14, F15, F16, G16	Multiplexed Address and Data Bus. Address is indicated by S_FRAME# assertion during PCI transactions. Write data is stable and valid when S_IRDY# is asserted and Read data is stable and valid when S_TRDY# is asserted. Data is transferred on rising clock edges when S_IRDY# and S_TRDY# are asserted. During bus idle, the PCI 6150 drives S_AD[31:0] to valid logic levels when S_GNT[8:0]# are not asserted. (Refer to Section 13, "PCI Bus Arbitration," for further details.)
S_CBE[3:0]#	Secondary Command and Byte Enables	4	I/O PTS PCI	194, 180, 167, 149	B6, B9, B12, E15	Multiplexed Command and Byte Enable fields. Provides the transaction type during the PCI Address phase. In the Data phase of PCI Memory Write transactions, S_CBE[3:0]# provide the Byte Enables. During bus idle, PCI 6150 drives S_CBE[3:0]# to valid logic levels when S_GNT[8:0]# are not asserted when external arbitration is not activated. (Refer to Section 13, "PCI Bus Arbitration," for further details.)
S_DEVSEL#	Secondary Device Select	1	I/O PSTS PCI	175	A11	Asserted by the target, indicating that the device is accepting the transaction. As a master, the PCI 6150 waits for S_DEVSEL# assertion within five cycles of S_FRAME# assertion; otherwise, the transaction terminates with a Master Abort. Before being placed into a high-impedance state, S_DEVSEL# is driven to a high state for one cycle.
S_FRAME#	Secondary Frame	1	I/O PSTS PCI	179	A10	Driven by the initiator of a transaction to indicate the beginning and duration of an access. S_FRAME# de-assertion indicates the final Data phase requested by the initiator. Before being placed into a high-impedance state, S_FRAME# is driven to a high state for one cycle.
S_GNT0#	Secondary Grant 0	1	I/O PTS	10	D1	Behaves as S_GNT[8:1]# when external arbitration is not activated. When external arbitration is activated, becomes the External Bus Request output from the PCI 6150.

Table 3-5. Secondary PCI Bus Interface Pins (Continued)

Symbol	Signal Name	Total Pins	Pin Type	PQFP Pin Number	PBGA Pin Number	Function
S_GNT[8:1]#	Secondary Grants 8 through 1	8	O PO	19, 18, 17, 16, 15, 14, 13, 11	G1, F1, F2, G3, F4, E1, E2, F3	Asserted by the PCI 6150 to access the secondary bus. The PCI 6150 de-asserts S_GNT[8:1]# for at least two PCI Clock cycles before re-asserting them. During bus idle, with S_GNT[8:1]# asserted, the PCI 6150 drives S_AD[31:0], S_CBE[3:0]#, and S_PAR to valid logic levels. (Refer to Section 13, “PCI Bus Arbitration,” for further details.) Pull S_GNT[8:1]# high if S_CFN#=1.
S_IRDY#	Secondary Initiator Ready	1	I/O PSTS PCI	177	B10	Driven by the initiator of a transaction to indicate its ability to complete the current Data phase on the secondary bus. Once asserted in a data phase, it is not de-asserted until end of the Data phase. Before being placed into a high-impedance state, S_IRDY# is driven to a de-asserted state for one cycle.
S_LOCK#	Secondary Lock	1	I/O PSTS	172	C11	Asserted by the bus master, indicating an atomic operation that may require multiple transactions to complete.
S_M66EN	Secondary 66 MHz Enable	1	I/O PTS	153	D15	Driven low if P_M66EN is low; otherwise, driven from outside to select 66 or 33 MHz. S_M66EN ***must*** be pulled high or low with a 10K-Ohm resistor. Along with P_M66EN, controls the frequency output to the S_CLKO[9:0] pins. (Refer to Section 4, “Clocking,” for further details.)
S_PAR	Secondary Parity	1	I/O PTS PCI	168	A13	Parity is even across S_AD[31:0], S_CBE[3:0]#, and S_PAR [*that is,* an even number of ones (1)]. S_PAR is an input, and is valid and stable for one cycle after the Address phase (indicated by S_FRAME# assertion) for address parity. For Write Data phases, S_PAR is an input and valid one clock after S_IRDY# assertion. For Read Data phases, S_PAR is an output and valid one clock after S_TRDY# assertion. S_PAR is placed into a high-impedance state one cycle after the S_AD[31:0] lines are placed into a high-impedance state. During bus idle, the PCI 6150 drives S_PAR to a valid logic level when S_GNT[8:1]# are asserted. (Refer to Section 13, “PCI Bus Arbitration,” for further details.)

Table 3-5. Secondary PCI Bus Interface Pins (Continued)

Symbol	Signal Name	Total Pins	Pin Type	PQFP Pin Number	PBGA Pin Number	Function
S_PERR#	Secondary Parity Error	1	I/O PSTS PCI	171	A12	Asserted when a Data Parity error is detected for data received on the secondary interface. Before being placed into a high-impedance state, S_PERR# is driven to a de-asserted state for one cycle.
S_REQ0#	Secondary Request 0	1	I/O PTS	207	B4	Asserted by an external device to request to start a transaction on the secondary bus. ***Must*** be externally pulled up through resistors to V_{DD}. When external arbitration is activated, becomes the External Bus Grant input from the PCI 6150.
S_REQ[8:1]#	Secondary Requests 8 through 1	8	I PI	9, 8, 7, 6, 5, 4, 3, 2	E4, E3, D2, C1, C2, D3, A2, B3	Asserted by an external device to request secondary bus ownership to perform a transaction. S_REQ[8:1]# ***must*** be externally pulled up through 10K-Ohm resistors to V_{DD}.
S_SERR#	Secondary System Error	1	I/O PTS	169	D11	S_SERR# can be driven low by any device to indicate a System error condition. The PCI 6150 drives S_SERR# if the following conditions are met: • Address Parity error • Posted Write Data Parity error on Target Bus • Master Abort during Posted Write transaction • Target Abort during Posted Write transaction • Posted Write transaction discarded • Delayed Write request discarded • Delayed Read request discarded • Delayed Transaction Master timeout Pull-up S_SERR# through an external resistor.
S_STOP#	Secondary Stop	1	I/O PSTS PCI	173	B11	Asserted by the secondary target to end the transaction on the current Data phase. Before being placed into a high-impedance state, S_STOP# is driven to a de-asserted state for one cycle.

Table 3-5. Secondary PCI Bus Interface Pins (Continued)

Symbol	Signal Name	Total Pins	Pin Type	PQFP Pin Number	PBGA Pin Number	Function
S_TRDY#	Secondary Target Ready	1	I/O PSTS PCI	176	C10	Driven by the target of a transaction to indicate its ability to complete the current Data phase on the secondary bus. Once asserted in a data phase, it is not de-asserted until end of the data phase. Before being placed into a high-impedance state, S_TRDY# is driven to a de-asserted state for one cycle.
Total		64				

Table 3-6. Clock-Related Pins

Symbol	Signal Name	Total Pins	Pin Type	PQFP Pin Number	PBGA Pin Number	Function
MSK_IN	Secondary Clock Disable Serial Input	PQFP: 1 PBGA: 0	I	126	—	Used by hardware mechanism to disable secondary clock outputs. The serial stream is received by MSK_IN, starting when P_RSTIN# is detected de-asserted and S_RSTOUT# is detected asserted. The serial data is used for selectively disabling secondary clock outputs and is shifted into the Secondary Clock Control Configuration register (SCLKCNTRL; PCI:68h). When tied low, enables all secondary clock outputs. Tied high, the clocks become active until high after reset. After ones (1) shift in, the clocks are driven high. ***Note:*** *Used in the PQFP package only. If using the PBGA package, use software to disable unused Secondary Clock buffers through the SCLKCNTRL; PCI:68h register.*
OSCSEL#	External Oscillator Enable	1	I	51	K16	Enables external clock connection for the secondary interface. If low, the secondary bus clock outputs use the clock signal from OSCIN, instead of P_CLKIN, to generate S_CLKO[9:0]. May optionally be pulled high or low. If high, P_CLKIN is used. OSCSEL# must ***not*** remain unconnected. ***Note:*** *When OSCSEL# input is V_{DD}, the external Clock function is disabled and OSC_IN input is ignored,*
OSCIN	External Oscillator Input	1	I	54	K15	External clock input used to generate secondary output clocks when enabled through the OSCSEL# pin. Pull high or low if unused. ***Note:*** *When OSCSEL# input is V_{DD}, the external Clock function is disabled and OSC_IN input is ignored,*
P_CLKIN	Primary Clock Input	1	I	45	M4	Provides timing for primary interface transactions.
S_CLKIN	Secondary Clock Input	1	I	21	H3	Provides timing for secondary interface transactions.

Table 3-6. Clock-Related Pins (Continued)

Symbol	Signal Name	Total Pins	Pin Type	PQFP Pin Number	PBGA Pin Number	Function
S_CLKO[9:0]	Secondary Clock Output	10	O	42, 41, 39, 38, 36, 35, 33, 32, 30, 29	M3, M2, N1, L4, L3, M1, L2, L1, K3, K2	Provides S_CLKIN or OSCIN (if enabled) phase synchronous output clocks. Pull-up resistors are not required on S_CLKO[9:0]; however, a series termination resistor is required when using these pins. S_CLKO0 drives the CompactPCI backplane.
Total—PQFP ***Total—PBGA***		15 14				

Table 3-7. Reset Pins

Symbol	Signal Name	Total Pins	Pin Type	PQFP Pin Number	PBGA Pin Number	Function
P_RSTIN#	Primary Reset Input	1	I PI	43	P1	When P_RSTIN# is active, asynchronously place outputs in a high-impedance state, and float P_SERR# and P_GNT#. All primary port PCI standard Configuration registers at offsets 00h to 3Fh revert to their default state. When asserted, all primary PCI signals are placed into a high-impedance state. May require a pull-up resistor, depending on the application.
S_RSTOUT#	Secondary Reset Output	1	O PO	22	H1	Asserted when one of the following conditions is met: • P_RSTIN# is asserted S_RSTOUT# remains asserted if P_RSTIN# is asserted and does not de-assert until P_RSTIN# is de-asserted. • Bridge Control register Secondary Reset bit in Configuration space is set (BCNTRL[6]=1; PCI:3Eh). S_RSTOUT# remains asserted until BCNTRL[6]=0. When asserted, all control signals are placed into a high-impedance state and zeros (0) are driven on S_AD[31:0], S_CBE[3:0]# and S_PAR.
Total		2				

Table 3-8. CompactPCI Hot Swap Pins

Symbol	Signal Name	Total Pins	Pin Type	PQFP Pin Number	PBGA Pin Number	Function
EJECT_EN#	Ejector Pin Use Enable	1	I	106	R16	Used to enable the GPIO3 pin as EJECT input. If this pin is 1, GPIO3 functions as a GPIO pin. GPIO3 only functions as EJECT input when both GPIO3FN# and EJECT_EN# are tied low, which also enables Hot Swap capability. (Refer to Table 3-9.) If not used, EJECT_EN# ***must*** be at logic 0 and pulled low.
ENUM#	Enumeration	1	O OD PTS	127	J14	Indicates an open-drain bused signal asserted when an adapter was inserted or is ready to be extracted from a PCI slot. Asserted through the Hot Swap registers (HS_CNTL; PCI:E4h, HS_CSR; PCI:E6h, and HS_NEXT; PCI:E5h). If used, ENUM# requires a pull-up resistor.
GPIO3FN#	GPIO3 Function Select	1	i PI	155	B14	When GPIO3FN# is tied high, GPIO3 functions as a GPIO pin regardless of EJECT_EN# state. GPIO3 functions as Ejector input only when both GPIO3FN# and EJECT_EN# are tied low (Hot Swap enabled). (Refer to Table 3-9.)
PIN_LED/ EJECT	Status Blue LED	1	I/O	128	J16	Active high signal that allows other circuits to drive the Blue Hot Swap LED. Turns ON LED if RSTIN# is asserted, or when the LOO bit is set (HS_CSR[3]=1; PCI:E6h) and RSTIN# is de-asserted. After RSTIN# de-assertion, the LED remains ON until the eject switch (handle) is closed, then the PCI 6150 turns OFF the LED. If used, PIN_LED/EJECT does not require a pull-up nor pull-down resistor. However, if unused, PIN_LED/EJECT must be pulled high.
Total		4				

Note: *If the Hot Swap function is not used, pull GPIO3FN# high or GPIO3 low to disable the function.*

Table 3-9. EJECT_EN# and GPIO3FN# Settings for Enabling Hot Swap Capability

EJECT_EN#	GPIO3FN#	Hot Swap	Eject Input
0	0	Enabled	GPIO3
Don’t Care	1	Disabled	—

Table 3-10. JTAG Pins

Symbol	Signal Name	Total Pins	Pin Type	PQFP Pin Number	PBGA Pin Number	Function
TCK	Test Clock Input	1	I PU	133	H15	Used to clock state information and test data into and out of the PCI 6150 during Test Access Port (TAP) operation. Pull TCK high or low to a known state, using an external resistor.
TDI	Test Data Input	1	I PU	129	J15	Used to serially shift test data and test instructions into the PCI 6150 during TAP operation. Pull TDI high or low to a known state, using an external resistor.
TDO	Test Data Output	1	O	130	H16	Used to serially shift test data and test instructions out of the PCI 6150 during TAP operation. Pull TDO high using an external resistor.
TMS	Test Mode Select	1	I PU	132	H14	Used to control the PCI 6150 TAP controller state. Pull TMS high or low to a known state, using an external resistor.
TRST#	Test Reset	1	I	134	G15	Asynchronous JTAG logic reset. Provides asynchronous initialization of the TAP controller. TRST# ***must*** be externally pulled low with a 330-Ohm resistor.
Total		5				

Note: *The JTAG interface is described in Section 21, "Testability/Debug."*

Table 3-11. Serial EEPROM Pins

Symbol	Signal Name	Total Pins	Pin Type	PQFP Pin Number	PBGA Pin Number	Function
EE_EN#	Serial EEPROM Enable	1	I	103	C15	To enable serial EEPROM use, EE_EN# should be 0. Otherwise, connect to logic 1 state. If a serial EEPROM is used, EE_EN# requires a pull-down resistor. If a serial EEPROM is not used, pull up EE_EN# to disable serial EEPROM autoload during system boot-up.
EEPCLK	Serial EEPROM Clock	1	O	158	C14	Clock signal to the serial EEPROM interface. Used during autoload and for VPD functions. EEPCLK is placed into a high-impedance state if EE_EN#=1.
EEPDATA	Serial EEPROM Data	1	I/O	160	D14	Serial data interface to the serial EEPROM. Requires an external pull-up resistor. EEPDATA is placed into a high-impedance state if EE_EN#=1.
Total		3				

Note: *When input to EE_EN# is V_{DD}, the serial EEPROM function is disabled and the EEPCLK and EEPDATA pins are ignored.*

Table 3-12. Miscellaneous Pins

Symbol	Signal Name	Total Pins	Pin Type	PQFP Pin Number	PBGA Pin Number	Function
BPCC_EN	Bus/Power Click Control	1	I	44	N2	When tied high and the PCI 6150 is placed into the D_{3hot} power state, the PCI 6150 places the secondary bus into the B_2 power state. The PCI 6150 disables the secondary clocks and drives them to 0. When tied low, placing the PCI 6150 into the D_{3hot} power state has no effect on the secondary bus clocks.
CFG66	Primary Configuration 66 MHz	PQFP: 1 PBGA: 0	I	125	—	Pin state is reflected in the Primary Status register (PCISR[5]; PCI:06h). When 1, CFG66 enables the PCI 6150 to declare 66 MHz capability. Otherwise, CFG66 has no effect on PCI 6150 operation. ***Note:*** *Used in the PQFP package only. In the PBGA package, the 66 MHz-Capable bits are hardwired to 1 (PCISR[5]=1; PCI:06h and PCISSR[5]=1; PCI:1Eh) to indicate 66 MHz capability.*
GPIO[3:0]	General Purpose Input/ Output 3 to 0	4	I/O PU	24, 25, 27, 28	J3, J2, J1, K1	General purpose signals, programmable as input-only or bi-directional by writing to the GPIO Output Enable register (GPIOOE; PCI:66h). During P_RSTIN# assertion, GPIO[3:0] are used to shift in the Clock Disable serial data. If configured as input, pull high or low, depending on application. When Hot Swap is enabled, GPIO3 functions as Ejector input only when both GPIO3FN# and EJECT_EN# are tied low.
P_V_{IO}	Primary Interface I/O Voltage	1	I	124	K14	Must be tied to 3.3 or 5V, depending on the primary interface signaling voltage.

Table 3-12. Miscellaneous Pins (Continued)

Symbol	Signal Name	Total Pins	Pin Type	PQFP Pin Number	PBGA Pin Number	Function
S_CFN#	Internal Arbiter Enable	1	I	23	H2	Values: 0 = Uses Internal Arbiter. 1 = Uses External Arbiter. May optionally be pulled high or low; however, S_CFN# ***must*** be tied low to use the Internal Arbiter.
S_V_{IO}	Secondary Interface I/O Voltage	1	I	135	G14	Must be tied to 3.3 or 5V, depending on the secondary interface signaling voltage.
Total—PQFP ***Total—PBGA***		9 8				

Table 3-13. Power, Ground, and Reserved Pins

Symbol	Signal Name	Total Pins	Pin Type	PQFP Pin Number	PBGA Pin Number	Function
RESERVED	Reserved	PQFP: 1 PBGA: 0	—	151	—	The PCI 6150 does not use this pin.
V_{DD}	Power	PQFP: 28 PBGA: 46	I	1, 26, 34, 40, 53, 56, 62, 69, 75, 81, 91, 97, 105, 108, 114, 120, 131, 139, 145, 157, 163, 170, 178, 184, 190, 196, 202, 208	A3, C4, D7, D8, D9, D10, E6, E7, E8, E9, E10, E11, F5, F12, G4, G5, G12, G13, H4, H5, H12, H13, J4, J5, J12, J13, K4, K5, K12, K13, L5, L12, M6, M7, M8, M9, M10, M11, N7, N8, N9, N10, P13, P15, R3, T3	+3.3V power supply.
V_{SS}	Ground	PQFP: 28 PBGA: 61	I	12, 20, 31, 37, 48, 52, 59, 66, 72, 78, 86, 94, 100, 104, 111, 117, 123, 136, 142, 148, 156, 166, 174, 181, 187, 193, 199, 205	A1, A16, B1, B2, B15, C3, C13, D4, D13, E5, E12, F6, F7, F8, F9, F10, F11, G2, G6, G7, G8, G9, G10, G11, H6, H7, H8, H9, H10, H11, J6, J7, J8, J9, J10, J11, K6, K7, K8, K9, K10, K11, L6, L7, L8, L9, L10, L11, M5, M12, N4, N13, N14, P3, P4, P14, R2, R4, R15, T1, T16	Ground.
Total—PQFP ***Total—PBGA***		57 107				

4 CLOCKING

This section describes the PCI 6150 clocking requirements.

To correctly operate, the PCI 6150 requires both a primary and secondary clock.

4.1 PRIMARY AND SECONDARY CLOCK INPUTS

The PCI 6150 implements a separate clock input for each PCI interface. The primary interface is synchronized to the primary Clock input, P_CLKIN. The secondary interface is synchronized to the Secondary Clock input, S_CLKIN.

The PCI 6150 operates at a maximum frequency of 66 MHz. Output clocks S_CLKO[9:0] can be derived from P_CLKIN, P_CLKIN/2, or an external asynchronous clock source.

The PCI 6150 primary and Secondary Clock inputs can be asynchronous. There are no skew constraints between these Clock inputs; however, the maximum ratio between the primary and secondary clock frequencies are 1:2.5 or 2.5:1.

4.2 SECONDARY CLOCK OUTPUTS

The PCI 6150 has ten Secondary Clock outputs that can be used to drive up to nine external secondary bus devices, Typically, S_CLKO0 or S_CLKO4 is used to drive the PCI 6150 S_CLKIN signal.

The rules for using secondary clocks are as follows:

- Each secondary clock output is limited to no more than one PCI load at 66 MHz
- Each clock trace length, including the feedback clock to the PCI 6150 S_CLKIN signal, must have equal length and impedance
- Terminate or disable unused secondary clock outputs to reduce power dissipation and noise in the system

4.3 DISABLING UNUSED SECONDARY CLOCK OUTPUTS

Note: *MSK_IN is used in the PQFP package only. If using the PBGA package, use software to disable unused Secondary Clock buffers through the SCLKCNTRL; PCI:68h register.*

When Secondary Clock outputs are not used, GPIO[2, 0] and MSK_IN can be used to clock in a serial mask that selectively three-states secondary clock outputs. Refer to Section 14, "GPIO Interface," for details in this application.

After the serial mask is shifted into the PCI 6150, the mask value is readable and can be changed in the Clock Disable bits (SCLKCNTRL[13:0]; PCI:68h). When the mask is modified by a Configuration Write operation to this register, the new clock mask disables the appropriate secondary clock outputs within a few cycles. This feature allows software to disable or enable Secondary Clock outputs based on the presence of option cards, and so forth.

The PCI 6150 delays de-asserting S_RSTOUT#, until the Serial Clock mask has completely shifted in and the secondary clocks are disabled or enabled, according to the mask. The delay between P_RSTIN# assertion and S_RSTOUT# de-assertion is 16 to 32 clocks.

4.3.1 Secondary Clock Control

Note: *MSK_IN is used in the PQFP package only. If using the PBGA package, use software to disable unused Secondary Clock buffers through the SCLKCNTRL; PCI:68h register.*

The PCI 6150 uses the GPIO[2, 0] pins and MSK_IN signal to input a 16-bit Serial Data stream. This data stream is shifted into the Secondary Clock Control register, as soon as P_RSTIN# is detected de-asserted and S_RSTOUT# is detected, and is used for selectively disabling S_CLKO[9:0] (SCLKCNTRL [13:0]; PCI:68h). S_RSTOUT# de-assertion is delayed until the PCI 6150 completes shifting in the Clock Mask data, taking 16 Clock cycles (32 cycles if operating at 66 MHz). After that, the GPIO[2, 0] pins can be used as general purpose I/O pins.

An External Shift register should be used to load and shift the data. (Refer to Figure 4-1.) The GPIO[2, 0] pins are used for Shift register control and serial data input, which occurs by way of a dedicated input signal, MSK_IN. The Shift register circuitry is unnecessary for correct PCI 6150 operation. The Shift registers may be eliminated, and MSK_IN can be tied low to enable all S_CLKO[9:0] signals, or tied high to force all S_CLKO[9:0] signals high. Table 4-1 delineates GPIO[2, 0] pin Shift register operation and Table 4-2 delineates serial data formatting, based on a design where the PCI 6150 secondary bus is used to drive up to four PCI adapter card slots or nine devices in an embedded system.

As noted in Table 4-2, the first eight bits contain the Philips 74F166 PRSNT*x*[2:1]# signal (refer to Figure 4-1) values for four slots, and control S_CLKO[3:0]. If one or both of the PRSNT*x*[2:1]# signals are 0, a card is present in the slot and the secondary clock for that slot is not masked. If these clocks are connected to devices and not to slots, tie one or both of the bits low, to enable the clock. The next five bits are the clock device masks (*that is*, each bit enables or disables the clock for one device). These bits control S_CLKO[8:4]—a value of 0 enables the clock, and 1 disables the clock. Bit 13 is the S_CLKO9 Clock Enable bit, which is connected to the PCI 6150 S_CLKIN.

Table 4-1. GPIO Shift Register Operation

Pin	Operation
GPIO0	Shift register Clock output at 66 MHz maximum frequency.
GPIO1	***Not used.***
GPIO2	Shift Register Control. Values: 0 = Load 1 = Shift
GPIO3	***Not used.***

If desired, the assignment of S_CLKO*x* to slots, devices, and PCI 6150 S_CLKIN input can be re-arranged from the assignment noted here. However, it is important that the Serial Data Stream format match the assignment of S_CLKO*x*. The GPIO[2, 0] pin serial protocol is designed to work with two Philips 74F166, 8-bit Bi-Directional Universal Shift registers.

The eight least significant bits, SCLKCNTRL[7:0], are connected to the 74F166 PRSNT*x*[2:1]# pins for the slots. The SCLKCNTRL[12:8] are tied high to disable their respective secondary clocks because those clocks are not connected. SCLKCNTRL[13] is tied high because S_CLKO9 is connected to the PCI 6150 S_CLKIN signal.

Figure 4-1 illustrates an example application where the PCI 6150 is connected to four PCI adapter card slots. The PRSNT*x*[2:1]# pin values on the 74F166 devices are shifted into SCLKCNTRL[7:0]. The PRSNT0[1]# value is shifted into SCLKCNTRL[0], PRSNT0[2]# value is shifted into SCLKCNTRL[1], and so forth. Bit 0 in the upper 74F166 is tied low, and thus enables S_CLKO4. In this application, S_CLKO4 may be used as the feedback to S_CLKIN.

When S_RSTOUT# is detected asserted and P_RSTIN# is detected de-asserted, the PCI 6150 drives GPIO2 low for one cycle to load the clock mask inputs into the Shift register. On the next cycle, the PCI 6150 drives GPIO2 high to perform a Shift operation. This shifts the clock mask into MSK_IN; the most significant bit is shifted in first, and the least significant bit is shifted in last. (Refer to Figure 4-2.)

After the Shift operation is complete, the PCI 6150 places GPIO[2, 0] into a high-impedance state and can de-assert S_RSTOUT# if the Secondary Reset bit is clear (BCNTRL[6]=0; PCI:3Eh). The PCI 6150 then ignores MSK_IN, and GPIO signal control reverts to the PCI 6150 GPIO Control registers. The Clock Disable bits can be subsequently modified through a Configuration Write command to the Secondary Clock Control register (SCLKCNTRL; PCI:68h) in device-specific Configuration space.

Table 4-2. GPIO Serial Data Format

SCLKCNTRL[15:0]	Description	S_CLKO[9:0]
1:0	Slot 0 74F166 PRSNT0[2:1]# or Device 0	0
3:2	Slot 1 74F166 PRSNT1[2:1]# or Device 1	1
5:4	Slot 2 74F166 PRSNT2[2:1]# or Device 2	2
7:6	Slot 3 74F166 PRSNT3[2:1]# or Device 3	3
8	Device 4	4
9	Device 5	5
10	Device 6	6
11	Device 7	7
12	Device 8	8
13	Device 9	9
15:14	***Reserved***	—

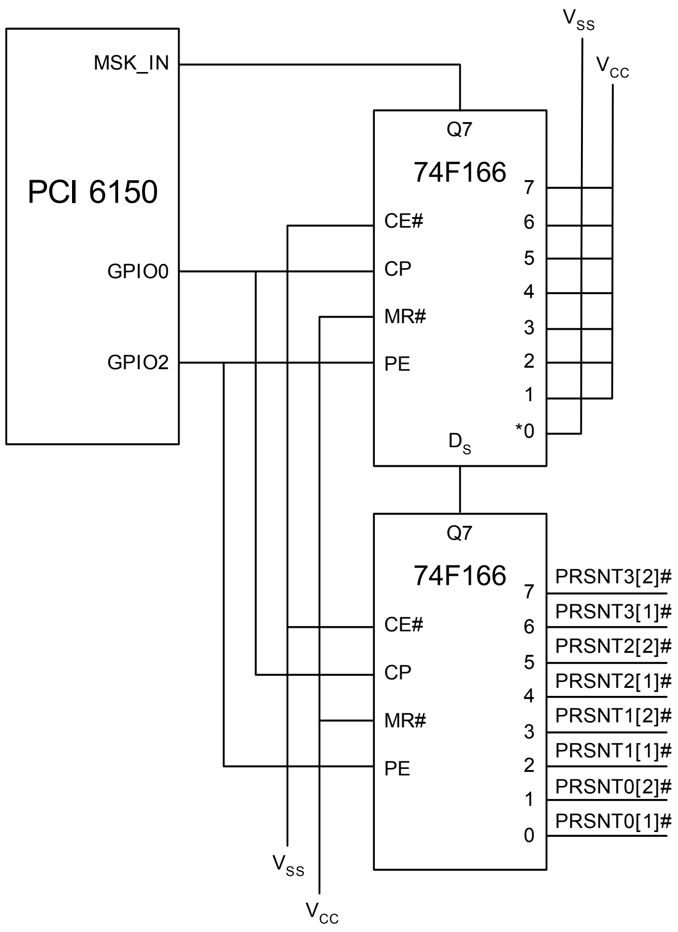


Figure 4-1. GPIO Clock Mask Implementation on System Board Example

Notes: ** Pulling the upper 74F166 bit 0 low enables S_CLKO4.*

In the Philips 74F166 PRSNTx# signals, x indicates the slot number, and the number in brackets indicates the appropriate PRSNT# signal (for example, PRSNT0[1]# is signal PRSNT1# of slot 0).

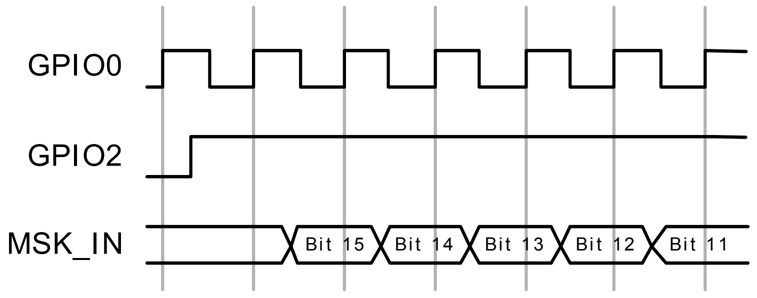


Figure 4-2. Clock Mask and Load Shift Timing

4.4 FREQUENCY DIVISION OPTIONS

The PCI 6150 has built-in frequency division options to automatically adjust the S_CLKO[9:0] clocks for PCI 33 or 66 MHz operation. Table 4-3 lists the clock division ratios used, depending on the P_M66EN and S_M66EN signal states.

Table 4-3. PCI Clock Frequency Division Ratios

P_M66EN Value	S_M66EN Value	PCI Clock Frequency Division Ratio
1	1	1/1
1	0	1/2
0	1	1/1
0	0	1/1

Note: *S_M66EN* ***cannot*** *be floating.*

4.5 USING AN EXTERNAL CLOCK SOURCE

The PCI 6150 uses two signals—OSCSEL# and OSCIN—when connecting an external clock source to the PCI 6150. During normal operation, the PCI 6150 generates S_CLKO[9:0], based on the PCI clock source (P_CLKIN). If OSCSEL# is asserted (low), then the PCI 6150 derives S_CLKO[9:0] from the OSCIN signal instead. Clock division is performed on the OSCIN and P_CLKIN clocks, depending on the P_M66EN and S_M66EN signal states.

4.6 RUNNING SECONDARY PORT FASTER THAN PRIMARY PORT

The PCI 6150 allows the secondary port to use a higher clock frequency than that of the primary port. In this case, a secondary clock source, using an external oscillator or clock generator, must be provided.

If the external oscillator is connected to OSCIN and OSCSEL# is asserted (low), then the output generated by S_CLKO[9:0] is divided, as per Table 4-3. Division control can be disabled by pulling S_M66EN high and not connecting this pin to a PCI slot (which may be on the secondary bus). If the S_CLKO[9:0] outputs are not required, then the external clock can be fed directly into the S_CLKIN signal.

5 RESET AND INITIALIZATION

This section describes 66 MHz operation, primary, secondary, and power management reset, and register initialization.

Note: *JTAG reset is discussed in Section 21.1.4, "JTAG Reset Input TRST#."*

5.1 66 MHZ OPERATION

Note: *CFG66 is used in the PQFP package only. In the PBGA package, the 66 MHz-Capable bits are hardwired to 1 (PCISR[5]=1; PCI:06h and PCISSR[5]=1; PCI:1Eh) to indicate 66 MHz capability.*

The PCI 6150 supports up to 66 MHz operation. The CFG66 and P_M66EN pin inputs should be high for 66 MHz operation.

The CFG66 signal must be tied high on the board to enable 66 MHz operation and to set the Status register 66 MHz Capable bits in Configuration space (PCISR[5]=1; PCI:06h and PCISSR[5]=1; PCI:1Eh).

The P_M66EN and S_M66EN signals indicate whether the primary and secondary interfaces, respectively, are operating at 66 MHz. This information is needed to control the secondary bus frequency. Per *PCI r2.3*, for clock frequencies between 33 and 66 MHz, the clock frequency may not change except while P_RSTIN# is asserted, or when Spread Spectrum Clocking (SSC) is used to reduce EMI emissions.

The following primary and secondary bus frequency combinations are supported when using the primary P_CLKIN signal to generate secondary clock outputs:

- 66 MHz primary bus, 66 MHz secondary bus
- 66 MHz primary bus, 33 MHz secondary bus
- 33 MHz primary bus, 33 MHz secondary bus

If P_M66EN is low (*for example*, the primary bus runs at 33 MHz), the PCI 6150 drives S_M66EN low to indicate that the secondary bus is operating at 33 MHz. If the secondary bus is set to run faster than the primary bus, S_M66EN need not be connected to secondary PCI devices.

The PCI 6150 can also generate S_CLKO[9:0] from OSCIN, if enabled. When the PCI 6150 is running with external clock input that is not generated from S_CLKO[9:0], the P_M66EN- and S_M66EN-controlled clock division does not apply.

When OSCIN or other external clock inputs are used for the secondary port, the PCI 6150 can run with a maximum ratio of 1:2.5 or 2.5:1 between the primary and secondary bus clocks.

For further details, refer to Section 4.4, "Frequency Division Options," and Section 4.6, "Running Secondary Port Faster than Primary Port."

5.2 RESET

This subsection describes the primary and secondary interface and chip reset mechanisms. The PCI 6150 has two Reset mechanisms and two Reset pins—P_RSTIN# and S_RSTOUT#. (Refer to Table 5-1.) In addition, the PCI 6150 can respond to Power Management-initiated internal resets.

After the Reset signals are de-asserted, the PCI 6150 requires 256 clocks to initialize bridge functions. During this initialization, Type 0 accesses can be accepted. However, no Memory nor I/O transactions are allowed through the bridge during this time.

Table 5-1. Reset Input Sources

Reset Inputs	Function
P_RSTIN#	• Resets primary and secondary ports • Causes S_RSTOUT# to be active • Causes serial EEPROM load
S_RSTOUT#	Not used as input
Chip Reset (DCNTRL[0]=1; PCI:41h)	Resets internal state machines
Secondary Reset (BCNTRL[6]=1; PCI:3Eh)	• Resets only secondary port • Causes S_RSTOUT# to be active

5.2.1 Primary Reset Input

To properly reset, the PCI 6150 requires at least two clocks before the P_RSTIN# rising edge.

When P_RSTIN# is asserted, the following events occur:

1. PCI 6150 immediately places all primary PCI interface signals into a high-impedance state.
2. All registers are reset.
3. P_RSTIN# assertion automatically causes a secondary port reset. Forty-three clocks after P_RSTIN# goes high, S_RSTOUT# goes high.
4. Clock Disable bits begin shifting in at the rising edge of P_RSTIN#.

The asserting and de-asserting edges of P_RSTIN# can be asynchronous to P_CLKIN and S_CLKIN. The P_RSTIN# asserting and de-asserting edges can be asynchronous to P_CLKIN and S_CLKIN. The PCI 6150 requires 256 primary port PCI clocks after P_RSTIN# rising edge to reset its internal logic.

When P_RSTIN# is asserted, all primary PCI interface signals, including the primary Request output, are immediately placed into a high-impedance state. All Posted Write and Delayed Transaction Data buffers are reset. Therefore, transactions residing in the buffers are discarded upon P_RSTIN# assertion.

5.2.2 Secondary Reset Output

The PCI 6150 is responsible for driving the secondary bus Reset signal, S_RSTOUT#. The PCI 6150 asserts S_RSTOUT# when any of the following conditions are met:

- P_RSTIN# asserted

  S_RSTOUT# remains asserted if P_RSTIN# is asserted and does not de-assert until P_RSTIN# is de-asserted.
- Bridge Control register Secondary Reset bit is set (BCNTRL[6]=1; PCI:3Eh)

  S_RSTOUT# remains asserted until BCNTRL[6]=0.

When S_RSTOUT# is asserted, all secondary PCI interface control signals, including S_GNT[8:0]#, are immediately placed into a high-impedance state. S_AD[31:0], S_CBE[3:0]#, and S_PAR are driven low for the duration of S_RSTOUT# assertion. All Posted Write and Delayed Transaction Data buffers are reset; therefore, any transactions residing in buffers at the time of secondary reset are discarded.

When S_RSTOUT# is asserted by means of the Secondary Reset bit, the PCI 6150 remains accessible during secondary interface reset and continues to respond to Configuration Space accesses from the primary interface.

5.2.3 JTAG Reset

Refer to Section 21.1.4, “JTAG Reset Input TRST#.”

5.2.4 Software Resets

The Diagnostic Control register Chip Reset bit can be used to reset the PCI 6150 (DCNTRL[0]=1; PCI:41h). This action causes S_RSTOUT# assertion; however, the signals are ***not*** placed into a high-impedance state.

When the Chip Reset bit is set, all registers and chip states are reset. When chip reset completes, within four PCI Clock cycles after completion of the Configuration Write operation that sets the Chip Reset bit, the Chip Reset bit automatically clears and the PCI 6150 is ready for configuration. During chip reset, the PCI 6150 is inaccessible.

5.2.5 Power Management Internal Reset

.When there is a D_{3hot}-to-D_0 transition with the Power Management Control/Status register Power State bits programmed to D_0 (PMCSR[1:0]=00b; PCI:E0h), an internal reset equivalent to P_RSTIN# is generated and all relevant registers are reset. However, S_RSTOUT# is ***not*** asserted.

5.3 REGISTER INITIALIZATION

The PCI 6150 Configuration registers may be initialized in one of three ways:

- Default values
- Serial EEPROM contents
- Host initialization

5.3.1 Default Initialization

After P_RSTIN# de-assertion, the PCI 6150 automatically checks for a valid a serial EEPROM. If the serial EEPROM is not valid nor present, the PCI 6150 automatically loads default values into the Configuration registers. (Refer to the “Value after Reset” column of the register tables in Section 6, “Registers.”)

5.3.2 Serial EEPROM Initialization

After P_RSTIN# de-assertion, if the PCI 6150 finds a valid serial EEPROM, register values are loaded from the serial EEPROM and overwrite the default values. (Refer to Section 7.3, “Serial EEPROM Autoload Mode at Reset.”)

5.3.3 Host Initialization

When device initialization is complete, the host system may access the appropriate registers to configure them according to system requirements.

Typically, registers are accessed by performing Type 0 Configuration accesses from the appropriate bus.

For details regarding register access, refer to Section 6, “Registers.”

Note: *Not all registers may be written to nor available from both sides of the bridge.*

6 REGISTERS

This section describes the PCI 6150 PCI registers. The PCI 6150 includes the standard Type 01h Configuration Space header, as defined in *P-to-P Bridge r1.1*.

Note: *Registers listed with a PCI offset or address are accessed by standard PCI Type 0 Configuration accesses.*

6.1 PCI CONFIGURATION REGISTER ADDRESS MAPPING

Table 6-1. PCI Configuration Register Address Mapping

PCI Configuration Register Address	To ensure software compatibility with other versions of the PCI 6150 family and to ensure compatibility with future enhancements, write 0 to all unused bits.				PCI Writable	Serial EEPROM Writable
	31 24	23 16	15 8	7 0		
00h	Device ID*		Vendor ID*		Yes	Yes
04h	Primary Status		Primary Command		Yes	No
08h	Class Code*			Revision ID	Yes	Yes
0Ch	Built-In Self-Test*	Header Type*	Primary Latency Timer	Cache Line Size	Yes	Yes
10h – 17h	***Reserved***				No	No
18h	Secondary Latency Timer	Subordinate Bus Number	Secondary Bus Number	Primary Bus Number	Yes	No
1Ch	Secondary Status		I/O Limit	I/O Base	Yes	No
20h	Memory Limit		Memory Base		Yes	No
24h	Prefetchable Memory Limit		Prefetchable Memory Base		Yes	No
28h	Prefetchable Memory Base Upper 32 Bits				Yes	No
2Ch	Prefetchable Memory Limit Upper 32 Bits				Yes	No
30h	I/O Limit Upper 16 Bits		I/O Base Upper 16 Bits		Yes	No
34h	***Reserved***			New Capability Pointer (DCh if Power Management Support; otherwise, E4h)	No	No
38h	***Reserved***				No	No
3Ch	Bridge Control		Interrupt Pin	***Reserved***	Yes	No
40h	Arbiter Control		Diagnostic Control	Chip Control	Yes	No
44h	Miscellaneous Options		Timeout Control	Primary Flow-Through Control	Yes	Yes
48h	Secondary Incremental Prefetch Count	Primary Incremental Prefetch Count	Secondary Prefetch Line Count	Primary Prefetch Line Count	Yes	Yes
4Ch	***Reserved***	Secondary Flow-Through Control	Secondary Maximum Prefetch Count	Primary Maximum Prefetch Count	Yes	Yes

Table 6-1. PCI Configuration Register Address Mapping (Continued)

PCI Configuration Register Address	To ensure software compatibility with other versions of the PCI 6150 family and to ensure compatibility with future enhancements, write 0 to all unused bits.				PCI Writable	Serial EEPROM Writable
	31 24	23 16	15 8	7 0		
50h	***Reserved***	Test	Internal Arbiter Control		Yes	No
54h	Serial EEPROM Data		Serial EEPROM Address	Serial EEPROM Control	Yes	No
58h – 63h	***Reserved***				No	No
64h	GPIO[3:0] Input Data	GPIO[3:0] Output Enable	GPIO[3:0] Output Data	P_SERR# Event Disable	Yes	No
68h	***Reserved***	P_SERR# Status	Secondary Clock Control		Yes	No
6Ch – 98h	***Reserved***				No	No
9Ch	***Reserved***			Read-Only Register Control	Yes	No
A0h – D8h	***Reserved***				Yes	No
DCh	Power Management Capabilities*		Power Management Next Capability Pointer (E4h)	Power Management Capability ID (01h)	Yes	Yes
E0h	Power Management Data*	PMCSR Bridge Supports Extensions	Power Management Control/Status*		Yes	Yes
E4h	***Reserved***	Hot Swap Control/ Status (0h)	Hot Swap Next Capability Pointer (E8h)	Hot Swap Control (Capability ID) (06h)	Yes	No
E8h	VPD Address (0h)		VPD Next Capability Pointer (00h)	VPD Capability ID (03h)	Yes	No
ECh	VPD Data (0h)				Yes	No

Notes: ** Writable only when the Read-Only Registers Write Enable bit is set (RRC[7]=1; PCI:9Ch).*

.Refer to the individual register descriptions to determine which bits are writable.

6.1.1 PCI Type 1 Header

Register 6-1. (PCIIDR; PCI:00h) PCI Configuration ID

Bit	Description	Read	Write	Value after Reset
15:0	**Vendor ID.** Identifies PCI 6150 manufacturer. Defaults to the PCI-SIG-issued PLX Vendor ID (3388h), if a blank or no serial EEPROM is present.	Yes	Only if RRC[7]=1; Serial EEPROM	3388h
31:16	**Device ID.** Identifies the particular device. Defaults to PLX PCI 6150 part number (0022h), if a blank or no serial EEPROM is present.	Yes	Only if RRC[7]=1; Serial EEPROM	0022h

Register 6-2. (PCICR; PCI:04h) Primary PCI Command

Bit	Description	Read	Write	Value after Reset
0	**I/O Space Enable.** Controls bridge response to I/O accesses on primary interface. Values: 0 = Ignores I/O transactions 1 = Enables response to I/O transactions	Yes	Yes	0
1	**Memory Space Enable.** Controls bridge response to Memory accesses on primary interface. Values: 0 = Ignores Memory transactions 1 = Enables response to Memory transactions	Yes	Yes	0
2	**Bus Master Enable.** Controls bridge ability to operate as a master on primary interface. Values: 0 = Does not initiate transactions on primary interface and disables response to Memory or I/O transactions on secondary interface 1 = Enables bridge to operate as a master on primary interface	Yes	Yes	0
3	**Special Cycle Enable.** ***Not Supported.***	Yes	No	0
4	**Memory Write and Invalidate Enable.** ***Not Supported.***	Yes	No	0
5	**VGA Palette Snoop Enable.** Controls bridge response to VGA-compatible Palette accesses. Values: 0 = Ignores VGA Palette accesses on primary interface 1 = Enables response to VGA Palette writes on primary interface (I/O address AD[9:0]=3C6h, 3C8h, and 3C9h) ***Note:*** *If BCNTRL[3]=1; PCI:3Eh (VGA Enable bit), then VGA Palette accesses are forwarded, regardless of the PCICR[5] value.*	Yes	Yes	0
6	**Parity Error Response Enable.** Controls bridge response to Parity errors. Values: 0 = Ignores Parity errors 1 = Performs normal parity checking	Yes	Yes	0
7	**Wait Cycle Control.** If set to 1, the PCI 6150 performs address/data stepping.	Yes	Yes	1
8	**P_SERR# Enable.** Controls the primary System Error (P_SERR#) pin enable. Values: 0 = Disables P_SERR# driver 1 = Enables P_SERR# driver	Yes	Yes	0
9	**Fast Back-to-Back Enable.** Controls bridge ability to generate Fast Back-to-Back transactions to various devices on secondary interface. Values: 0 = No Fast Back-to-Back transactions 1 = Enables Fast Back-to-Back transactions	Yes	Yes	0
15:10	***Reserved.***	Yes	No	0h

Register 6-3. (PCISR; PCI:06h) Primary PCI Status

Bit	Description	Read	Write	Value after Reset
3:0	***Reserved.***	Yes	No	0h
4	**New Capability Functions Support.** Writing 1 supports New Capabilities Functions. The New Capability Function ID is located at the PCI Configuration space offset, determined by the New Capabilities linked list pointer value at CAP_PTR; PCI:34h.	Yes	No	1
5	**66 MHz-Capable.** If set to 1, this device supports a 66 MHz PCI clock environment. Reflects CFG66 pin state. ***Note:*** *Hardwired to 1 in the PBGA package.*	Yes	No	1
6	**UDF.** No User-Definable Features.	Yes	No	0
7	**Fast Back-to-Back Capable.** Fast Back-to-Back write capable on primary port. Set to 1.	Yes	No	0
8	**Data Parity Error Detected.** Set when the following conditions are met: • P_PERR# is asserted, and • Command register Parity Error Response Enable bit is set (PCICR[6]=1; PCI:04h) Writing 1 clears bit to 0.	Yes	Yes/Clr	0
10:9	**DEVSEL# Timing**. Reads as 01b to indicate PCI 6150 responds no slower than with medium timing.	Yes	No	01b
11	**Signaled Target Abort.** Set by a target device when a Target Abort cycle occurs. Writing 1 clears bit to 0.	Yes	Yes/Clr	0
12	**Received Target Abort.** Set to 1 by the PCI 6150 when transactions are terminated with Target Abort. Writing 1 clears bit to 0.	Yes	Yes/Clr	0
13	**Received Master Abort.** Set to 1 by the PCI 6150 when transactions are terminated with Master Abort. Writing 1 clears bit to 0.	Yes	Yes/Clr	0
14	**Signaled System Error.** Set when P_SERR# is asserted. Writing 1 clears bit to 0.	Yes	Yes/Clr	0
15	**Parity Error Detected.** Set when a Parity error is detected, regardless of the Parity Error Response Enable bit state (PCICR[6]=x; PCI:04h). Writing 1 clears bit to 0.	Yes	Yes/Clr	0

Register 6-4. (PCIREV; PCI:08h) PCI Revision ID

Bit	Description	Read	Write	Value after Reset
7:0	**Revision ID.** PCI 6150 silicon revision.	Yes	No	04h

Register 6-5. (PCICCR; PCI:09h – 0Bh) PCI Class Code

Bit	Description	Read	Write	Value after Reset
7:0	**Register Level Programming Interface.** None defined.	Yes	Only if RRC[7]=1; Serial EEPROM; Serial EEPROM	0h
15:8	**Subclass Code.** PCI-to-PCI bridge or other bridge device.	Yes	Only if RRC[7]=1; Serial EEPROM; Serial EEPROM	04h
23:16	**Base Class Code.** Bridge device.	Yes	Only if RRC[7]=1; Serial EEPROM; Serial EEPROM	06h

Register 6-6. (PCICLSR; PCI:0Ch) PCI Cache Line Size

Bit	Description	Read	Write	Value after Reset
7:0	**System Cache Line Size.** Specified in units of 32-bit words (Dwords). Only cache line sizes of a power of two are valid. Maximum value is 20h. For values greater than 20h, PCI 6150 operates as if PCICLSR is programmed with value of 08h. Used when terminating Memory Write and Invalidate transactions and prefetching. Memory read prefetching is controlled by the Prefetch Count registers. ***Note:*** *Only one bit can be set in this register.*	Yes	Yes	0h

Register 6-7. (PCILTR; PCI:0Dh) Primary PCI Bus Latency Timer

Bit	Description	Read	Write	Value after Reset
7:0	**Primary PCI Bus Latency Timer.** Specifies amount of time (in units of PCI Bus clocks) the PCI 6150, as a bus master, can burst data on the primary PCI Bus. Time counting begins when the master asserts P_FRAME#.	Yes	Yes	0h

Register 6-8. (PCIHTR; PCI:0Eh) PCI Header Type

Bit	Description	Read	Write	Value after Reset
6:0	**Configuration Layout Type.** Specifies register layout at offsets 10h to 3Fh in Configuration space. Header Type 0 is defined for PCI devices other than PCI-to-PCI bridges (Header Type 1) and Cardbus bridges (Header Type 2).	Yes	Only if RRC[7]=1; Serial EEPROM	1h
7	**Multi-Function Device.** Value of 1 indicates multiple (up to eight) functions (logical devices), each containing its own, individually addressable Configuration space, 64 Dwords in size.	Yes	Only if RRC[7]=1; Serial EEPROM	0

Note: *PCIHTR is hardcoded to 01h.*

Register 6-9. (PCIBISTR; PCI:0Fh) PCI Built-In Self-Test

Bit	Description	Read	Write	Value after Reset
7:0	**Built-In Self-Test (BIST).** Can only be programmed by serial EEPROM.	Yes	Only if RRC[7]=1; Serial EEPROM	0h

Register 6-10. (PCIPBNO; PCI:18h) PCI Primary Bus Number

Bit	Description	Read	Write	Value after Reset
7:0	**Primary Bus Number.** Programmed with the PCI Bus number to which the primary bridge interface is connected. Value is set with Configuration software.	Yes	Yes	0h

Register 6-11. (PCISBNO; PCI:19h) PCI Secondary Bus Number

Bit	Description	Read	Write	Value after Reset
7:0	**Secondary Bus Number.** Programmed with the PCI Bus number to which the secondary bridge interface is connected. Value is set with Configuration software.	Yes	Yes	0h

Register 6-12. (PCISUBNO; PCI:1Ah) PCI Subordinate Bus Number

Bit	Description	Read	Write	Value after Reset
7:0	**Subordinate Bus Number.** Programmed with the PCI Bus Number with the highest number subordinate to the bridge. Value is set with Configuration software.	Yes	Yes	0h

Register 6-13. (PCISLTR; PCI:1Bh) Secondary PCI Bus Latency Timer

Bit	Description	Read	Write	Value after Reset
7:0	**Secondary PCI Bus Latency Timer.** Specifies the amount of time (in units of PCI Bus clocks) the PCI 6150, as a bus master, can burst data on the secondary PCI Bus. Latency Timer checks for Master accesses on the secondary bus that remain unclaimed by targets.	Yes	Yes	0h

Register 6-14. (PCIIOBAR; PCI:1Ch) I/O Base

Bit	Description	Read	Write	Value after Reset
7:0	**I/O Base.** Specifies the I/O Base Address Range bits [15:12] for forwarding the cycle through the bridge (Base Address bits [11:0] are assumed to be 0h). Used in conjunction with the I/O Limit, I/O Base Upper 16 Bits, and I/O Limit Upper 16 Bits registers (PCIIOLMT; PCI:1Dh, PCIIOBARU16; PCI:30h, and PCIIOLMTU16; PCI:32h, respectively) to specify a range of 32-bit addresses supported for PCI Bus I/O transactions. The lower four Read-Only bits [3:0] are hardcoded to 0001b to indicate 32-bit I/O addressing support.	Yes	Yes [7:4]	1h

Register 6-15. (PCIIOLMT; PCI:1Dh) I/O Limit

Bit	Description	Read	Write	Value after Reset
7:0	**I/O Limit.** Specifies the Upper I/O Limit Address Range bits [15:12] for forwarding the cycle through the bridge (Limit Address bits [11:0] are assumed to be FFFh). Used in conjunction with the I/O Base, I/O Base Upper 16 Bits, and I/O Limit Upper 16 Bits registers (PCIIOBAR; PCI:1Ch, PCIIOBARU16; PCI:30h, and PCIIOLMTU16; PCI:32h, respectively) to specify a range of 32-bit addresses supported for PCI Bus I/O transactions. The lower four Read-Only bits [3:0] are hardcoded to 0001b to indicate 32-bit I/O addressing support.	Yes	Yes [7:4]	1h

Register 6-16. (PCISSR; PCI:1Eh) Secondary PCI Status

Bit	Description	Read	Write	Value after Reset
4:0	***Reserved.***	Yes	No	0h
5	**66 MHz-Capable.** If set to 1, the PCI 6150 supports a 66 MHz PCI clock environment. Reflects CFG66 pin state. ***Note:*** *Hardwired to 1 in the PBGA package.*	Yes	No	1
6	**UDF.** No User-definable features.	Yes	No	0
7	**Fast Back-to-Back Capable.** Fast Back-to-Back write capable on secondary port. Set to 1.	Yes	No	0
8	**Data Parity Error Detected.** Set when the following conditions are met: • S_PERR# is asserted, and • Command register Parity Error Response Enable bit is set (PCICR[6]=1; PCI:04h) Writing 1 clears bit to 0.	Yes	Yes/Clr	0
10:9	**DEVSEL# Timing**. Reads as 01b to indicate PCI 6150 responds no slower than with medium timing.	Yes	No	01b
11	**Signaled Target Abort.** Set by a target device when a Target Abort cycle occurs. Writing 1 clears bit to 0.	Yes	Yes/Clr	0
12	**Received Target Abort.** Set to 1 by PCI 6150 when transactions are terminated with Target Abort. Writing 1 clears bit to 0.	Yes	Yes/Clr	0
13	**Received Master Abort.** Set to 1 by PCI 6150 when transactions are terminated with Master Abort. Writing 1 clears bit to 0.	Yes	Yes/Clr	0
14	**Signaled System Error.** Set when S_SERR# is asserted. Writing 1 clears bit to 0.	Yes	Yes/Clr	0
15	**Parity Error Detected.** Set when a Parity error is detected, regardless of the Parity Error Response Enable bit state (PCICR[6]=x; PCI:04h). Writing 1 clears bit to 0.	Yes	Yes/Clr	0

Register 6-17. (PCIMBAR; PCI:20h) Memory Base

Bit	Description	Read	Write	Value after Reset
15:0	**Memory Base.** Specifies the Memory-Mapped I/O Base Address Range bits [31:20] for forwarding the cycle through the bridge. The upper 12 bits corresponding to [31:20] are writable. The lower 20 Address bits [19:0] are assumed to be 0h. Used in conjunction with the Memory Limit register (PCIMLMT; PCI:22h) to specify a range of 32-bit addresses supported for PCI Bus Memory-Mapped I/O transactions. The lower four Read-Only bits [3:0] are hardcoded to 0h.	Yes	Yes [15:4]	0h

Register 6-18. (PCIMLMT; PCI:22h) Memory Limit

Bit	Description	Read	Write	Value after Reset
15:0	**Memory Limit.** Specifies the Upper Memory-Mapped I/O Limit Address Range bits [31:20] for forwarding the cycle through the bridge. The upper 12 bits corresponding to [31:20] are writable. The lower 20 Address bits [19:0] are assumed to be F_FFFFh. Used in conjunction with the Memory Base register (PCIMBAR; PCI:20h) to specify a range of 32-bit addresses supported for PCI Bus Memory-Mapped I/O transactions. The lower four Read-Only bits [3:0] are hardcoded to 0h.	Yes	Yes [15:4]	0h

Register 6-19. (PCIPMBAR; PCI:24h) Prefetchable Memory Base

Bit	Description	Read	Write	Value after Reset
15:0	**Prefetchable Memory Base.** Specifies the Prefetchable Memory-Mapped Base Address Range bits [31:20] for forwarding the cycle through the bridge. The upper 12 bits corresponding to [31:20] are writable. The lower 20 Address bits [19:0] are assumed to be 0h. Used in conjunction with the Prefetchable Memory Limit, Prefetchable Memory Base Upper 32 Bits, and Prefetchable Memory Limit Upper 32 Bits registers (PCIPMLMT; PCI:26h, PCIPMBARU32; PCI:28h, and PCIPMLMTU32; PCI:2Ch, respectively) to specify a range of 64-bit addresses supported for Prefetchable Memory transactions on the PCI Bus. The lower four Read-Only bits [3:0] are hardcoded to 01h, indicating 64-bit address support.	Yes	Yes [15:4]	1h

Register 6-20. (PCIPMLMT; PCI:26h) Prefetchable Memory Limit

Bit	Description	Read	Write	Value after Reset
15:0	**Prefetchable Memory Limit.** Specifies the Upper Prefetchable Memory-Mapped Limit Address Range bits [31:20] for forwarding the cycle through the bridge. The lower 20 Address bits [19:0] are assumed to be F_FFFFh. Used in conjunction with the Prefetchable Memory Base, Prefetchable Memory Base Upper 32 Bits, and Prefetchable Memory Limit Upper 32 Bits registers (PCIPMBAR; PCI:24h, PCIPMBARU32; PCI:28h, and PCIPMLMTU32; PCI:2Ch, respectively) to specify a range of 64-bit addresses supported for Prefetchable Memory transactions on the PCI Bus. The lower four Read-Only bits [3:0] are hardcoded to 01h, indicating 64-bit address support.	Yes	Yes [15:4]	1h

Register 6-21. (PCIPMBARU32; PCI:28h) Prefetchable Memory Base Upper 32 Bits

Bit	Description	Read	Write	Value after Reset
31:0	**Prefetchable Memory Base Upper 32 Bits.** Specifies the Upper Prefetchable Memory-Mapped Base Address Range bits [63:32] for forwarding the cycle through the bridge. The lower 20 Address bits [19:0] are assumed to be 0h. Used in conjunction with the Prefetchable Memory Base, Prefetchable Memory Limit, and Prefetchable Memory Limit Upper 32 Bits registers (PCIPMBAR; PCI:24h, PCIPMLMT; PCI:26h, and PCIPMLMTU32; PCI:2Ch, respectively) to specify a range of 64-bit addresses supported for Prefetchable Memory transactions on the PCI Bus.	Yes	Yes	0h

Register 6-22. (PCIPMLMTU32; PCI:2Ch) Prefetchable Memory Limit Upper 32 Bits

Bit	Description	Read	Write	Value after Reset
31:0	**Prefetchable Memory Limit Upper 32 Bits.** Specifies the Upper Prefetchable Memory-Mapped Limit Address Range bits [63:32] for forwarding the cycle through the bridge. The lower 20 Address bits [19:0] are assumed to be F_FFFFh. Used in conjunction with the Prefetchable Memory Base, Prefetchable Memory Limit, and Prefetchable Memory Base Upper 32 Bits registers (PCIPMBAR; PCI:24h, PCIPMLMT; PCI:26h, and PCIPMBARU32; PCI:28h, respectively) to specify a range of 64-bit addresses supported for Prefetchable Memory transactions on the PCI Bus.	Yes	Yes	0h

Register 6-23. (PCIIOBARU16; PCI:30h) I/O Base Upper 16 Bits

Bit	Description	Read	Write	Value after Reset
15:0	**I/O Base Upper 16 Bits.** Specifies the Upper I/O Base Address Range bits [31:16] for forwarding the cycle through the bridge. Base Address bits [11:0] are assumed to be 0h. Used in conjunction with the I/O Base, I/O Limit, and I/O Limit Upper 16 Bits registers (PCIIOBAR; PCI:1Ch, PCIIOLMT; PCI:1Dh, and PCIIOLMTU16; PCI:32h, respectively) to specify a range of 32-bit addresses supported for PCI Bus I/O transactions.	Yes	Yes	0h

Register 6-24. (PCIIOLMTU16; PCI:32h) I/O Limit Upper 16 Bits

Bit	Description	Read	Write	Value after Reset
15:0	**I/O Limit Upper 16 Bits.** Specifies the Upper I/O Limit Address Range bits [31:16] for forwarding the cycle through the bridge. Limit Address bits [11:0] are assumed to be FFFh. Used in conjunction with the I/O Base, I/O Limit, and I/O Base Upper 16 Bits registers (PCIIOBAR; PCI:1Ch, PCIIOLMT; PCI:1Dh, and PCIIOBARU16; PCI:30h, respectively) to specify a range of 32-bit addresses supported for PCI Bus I/O transactions.	Yes	Yes	0h

Register 6-25. (CAP_PTR; PCI:34h) New Capability Pointer

Bit	Description	Read	Write	Value after Reset
7:0	**New Capability Pointer.** Provides an offset into PCI Configuration space for the Next capability location in the New Capabilities Linked List. If the selected Device ID supports Power Management, the value defaults to DCh. Otherwise, the value defaults to E4h (for Hot Swap). .	Yes	No	DCh (PM) or E4h (Hot Swap)
31:8	***Reserved.***	Yes	No	0h

Register 6-26. (PCIIPR; PCI:3Dh) PCI Interrupt Pin

Bit	Description	Read	Write	Value after Reset
7:0	**Interrupt Pin.** Reads as 0h to indicate that PCI 6150 does not use interrupt pins.	Yes	No	0h

Register 6-27. (BCNTRL; PCI:3Eh) Bridge Control

Bit	Description	Read	Write	Value after Reset
0	**Parity Error Response Enable.** Controls bridge response to Parity errors on secondary interface. Values: 0 = Ignores Address and Data Parity errors on secondary interface 1 = Enables Parity error reporting and detection on secondary interface	Yes	Yes	0
1	**S_SERR# Enable.** Controls forwarding of S_SERR# to primary interface. Values: 0 = Disables S_SERR# forwarding to primary 1 = Enables S_SERR# forwarding to primary	Yes	Yes	0
2	**ISA Enable.** Controls bridge response to ISA I/O addresses, which is limited to the first 64 KB. Values: 0 = Forwards I/O addresses in the range defined by the I/O Base and Limit registers (PCIIOBAR; PCI:1Ch and PCIIOLMT; PCI:1Dh, respectively). 1 = Blocks forwarding of ISA I/O addresses in the range defined by the I/O Base and Limit registers in the first 64 KB of I/O space that address the last 768 bytes in each 1-KB block. Secondary I/O transactions are forwarded upstream, if the address falls within the last 768 bytes in each 1-KB block. Command Configuration register Master Enable bit must also be set (PCICR[2]=1; PCI:04h) to enable ISA. ***Note:*** *There is an ISA Enable Control bit Write Protect mechanism controlled by serial EEPROM. When set in serial EEPROM, and serial EEPROM initialization is enabled, PCI 6150 changes this bit to Read-Only and the ISA-Enable feature is* ***not*** *available.*	Yes	Yes	0
3	**VGA Enable.** Controls bridge response to VGA-compatible addresses. Values: 0 = Does not forward VGA-compatible Memory nor I/O addresses from primary to secondary 1 = Forwards VGA-compatible Memory and I/O addresses from primary to secondary, regardless of other settings ***Note:*** *If set to 1, then I/O addresses in the range of 3B0h to 3BBh and 3C0h to 3DFh are forwarded, regardless of the PCICR[5]; PCI:04h or BCNTRL[2] values.*	Yes	Yes	0

Register 6-27. (BCNTRL; PCI:3Eh) Bridge Control (Continued)

Bit	Description	Read	Write	Value after Reset
4	***Reserved***	Yes	No	0
5	**Master Abort Mode.** Controls bridge behavior in response to Master Aborts on secondary interface. Values: 0 = Does not report Master Aborts (return FFFF_FFFFh on reads or discard data on writes). 1 = Reports Master Aborts by signaling Target Abort. If the Master Abort is the result of a primary-to-secondary Posted Write cycle, P_SERR# is asserted (PCICR[8]=1; PCI:04h). ***Note:*** *During Lock cycles, PCI 6150 ignores this bit, and completes the cycle as a Target Abort.*	Yes	Yes	0
6	**Secondary Reset.** Forces S_RSTOUT# assertion on secondary interface. Values: 0 = Does not force S_RSTOUT# assertion 1 = Forces S_RSTOUT# assertion	Yes	Yes	0
7	**Fast Back-to-Back Enable.** Controls bridge ability to generate Fast Back-to-Back transactions to various devices on secondary interface. Values: 0 = No Fast Back-to-Back transactions 1 = Enables Fast Back-to-Back transactions	Yes	Yes	0
8	**Primary Master Timeout (Discard Timer).** Sets the maximum number of PCI clocks for an initiator on the primary bus to repeat the Delayed transaction request. Values: 0 = Timeout after 2^{15} PCI clocks 1 = Timeout after 2^{10} PCI clocks	Yes	Yes	0
9	**Secondary Master Timeout (Discard Timer).** Sets the maximum number of PCI clocks for an initiator on the secondary bus to repeat the Delayed transaction request. Values: 0 = Timeout after 2^{15} PCI clocks 1 = Timeout after 2^{10} PCI clocks	Yes	Yes	0
10	**Master Timeout Status.** Set to 1 when primary or secondary Master Timeout occurs. Writing 1 clears bit to 0.	Yes	Yes/Clr	0
11	**Master Timeout P_SERR# Enable.** Enable P_SERR# assertion during Master Timeout. Values: 0 = P_SERR# not asserted on Master Timeout 1 = P_SERR# asserted on primary or secondary Master Timeout	Yes	Yes	0
15:12	***Reserved.***	Yes	No	0h

6.1.2 Device-Specific

6.1.2.1 Chip, Diagnostic, and Arbiter Control

Register 6-28. (CCNTRL; PCI:40h) Chip Control

Bit	Description	Read	Write	Value after Reset
0	***Reserved.***	Yes	No	0
1	**Memory Write Disconnect Control.** Controls when PCI 6150, as a target, Disconnects Memory transactions. Values: 0 = Disconnects on queue full or on a 4-KB boundary 1 = Disconnects on a Cache Line boundary, when the queue fills, or on a 4-KB boundary	Yes	Yes	0
2	**I/O 1-KB Decode.** Values: 0 = I/O decodes to 1-KB resolution 1 = I/O decodes to 4-KB resolution	Yes	Yes	0
3	***Reserved.***	Yes	No	0
4	**Secondary Bus Prefetch Disable.** Controls PCI 6150 ability to prefetch during upstream Memory Read transactions. Values: 0 = Prefetches and does not forward Byte Enables during Memory Read transactions. 1 = Requests only 1 Dword from the target during Memory Read transactions and forwards Byte Enables. PCI 6150 returns a Target Disconnect to the requesting master on the first Data transfer. Memory Read Line and Memory Read Multiple transactions remain prefetchable.	Yes	Yes	0
7:5	***Reserved.***	Yes	No	000b

Register 6-29. (DCNTRL; PCI:41h) Diagnostic Control

Bit	Description	Read	Write	Value after Reset
0	**Chip Reset.** Chip and secondary bus reset. Setting bit activates full chip reset, asserts S_RSTOUT#, and forces the Bridge Control register Secondary Reset bit to be set (BCNTRL[6]=1; PCI:3Eh). After resetting the PCI 6150 registers, bit is cleared; however, BCNTRL[6] remains set to 1. Writing 0 has no effect.	Yes	Yes	0
2:1	***Reserved and must be set to 00b.***	Yes	Yes	00b
3	**Secondary Reset Output Mask.** ***Not Supported***.	Yes	No	0
7:4	***Reserved.***	Yes	No	0h

Register 6-30. (ACNTRL; PCI:42h) Arbiter Control

Bit	Description	Read	Write	Value after Reset
8:0	**Arbiter Control.** Each bit controls whether a secondary bus master is assigned to the high- or low-priority group. Bits [8:0] correspond to request inputs S_REQ[8:0]#, respectively. Value of 1h assigns the bus master to the high-priority group. ***Note:*** *S_REQ0# is an I/O pin.*	Yes	Yes	0
11:9	***Reserved.***	Yes	No	0
12	**Primary Port Ordering Rule.** ***Reserved and must be set to 0.***	Yes	Yes	0
13	**Secondary Port Ordering Rule.** ***Reserved and must be set to 0.***	Yes	Yes	0
15:14	***Reserved and must be set to 0.***	Yes	No	0h

6.1.2.2 Primary Flow-Through Control

Register 6-31. (PFTCR; PCI:44h) Primary Flow-Through Control

Bit	Description	Read	Write	Value after Reset
2:0	**Primary Posted Write Completion Wait Count.** Maximum number of clocks the PCI 6150 waits for Posted Write data from the initiator if delivering Write data in Flow-Through mode and the Internal Post Write queues are almost empty. If the count is exceeded without additional data from the initiator, the cycle to the target is terminated and later completed. Values: 000b = Terminates the cycle if there is only one data entry remaining in the Internal Write queue 001b = De-asserts S_IRDY# and waits one clock for source data on the primary bus, before terminating cycle … 111b = De-asserts S_IRDY# and waits seven clocks for source data on the primary bus, before terminating cycle	Yes	Yes; Serial EEPROM	111b
3	***Reserved.*** Returns 00b when read.	Yes	No	0
6:4	**Primary Delayed Read Completion Wait Count.** Maximum number of clocks the PCI 6150 waits for Delayed Read data from the target, if returning Read data in Flow-Through mode and the Internal Delayed Read queue is almost full. If the count is exceeded without additional space in the queue, the cycle to target is terminated, and completed when the initiator Retries the remainder of the cycle. Values: 000b = Terminates the cycle if there is only one data entry remaining in the Read queue 001b = De-asserts S_TRDY# and waits one clock for source data on the primary bus, before terminating cycle … 111b = De-asserts S_TRDY# and waits seven clocks for source data on the primary bus, before terminating cycle	Yes	Yes; Serial EEPROM	111b
7	***Reserved.*** Returns 00b when read.	Yes	No	0

6.1.2.3 Timeout Control

Register 6-32. (TOCNTRL; PCI:45h) Timeout Control

Bit	Description	Read	Write	Value after Reset
2:0	**Maximum Retry Counter Control.** Controls the maximum number of times the PCI 6150 Retries a cycle before signaling a timeout. Timeout applies to Read/Write Retries and can be enabled to trigger SERR# on the primary or secondary port, depending on the SERR# events enabled. Maximum number of Retries to timeout: 000b = 2^{24} 001b = 2^{18} 010b = 2^{12} 011b = 2^{6} 111b = 2^{0}	Yes	Yes; Serial EEPROM	000b
3	***Reserved.***	Yes	No	0
5:4	**Primary Master Timeout Divider.** Provides additional options for the primary Master Timeout. In addition to its original setting in the Bridge Control register (BCNTRL[8]; PCI:3Eh), the Timeout Counter can optionally be divided by up to 256: 00b = Counter—Primary Master Timeout / 1 01b = Timeout Counter—Primary Master Timeout / 8 10b = Timeout Counter—Primary Master Timeout / 16 11b = Timeout Counter—Primary Master Timeout / 256 BCNTRL[8] can set the primary Master Timeout to 32K (default) or 1K Clock cycles.	Yes	Yes; Serial EEPROM	00b
7:6	**Secondary Master Timeout Divider.** Provides additional options for the secondary Master Timeout. In addition to its original setting in the Bridge Control register (BCNTRL[9]; PCI:3Eh), the Timeout Counter can optionally be divided by up to 256: 00b = Counter—Secondary Master Timeout / 1 01b = Timeout Counter—Secondary Master Timeout / 8 10b = Timeout Counter—Secondary Master Timeout / 16 11b = Timeout Counter—Secondary Master Timeout / 256 BCNTRL[9] can set the secondary Master Timeout to 32K (default) or 1K Clock cycles.	Yes	Yes; Serial EEPROM	00b

6.1.2.4 Miscellaneous Options

Register 6-33. (MSCOPT; PCI:46h) Miscellaneous Options

Bit	Description	Read	Write	Value after Reset
0	**Write Completion Wait for PERR#.** If set to 1, PCI 6150 waits for target PERR# status before completing a Delayed Write transaction to the initiator.	Yes	Yes; Serial EEPROM	0
1	**Read Completion Wait for PAR.** If set to 1, PCI 6150 waits for target PAR status before completing a Delayed Read transaction to the initiator.	Yes	Yes; Serial EEPROM	0
2	**Delayed Read Transaction (DRT) Out-of-Order Enable.** If set to 1, PCI 6150 may return Delayed Read transactions in a different order than requested. Otherwise, Delayed Read transactions are returned in the same order as requested.	Yes	Yes; Serial EEPROM	0
3	**Generate Parity Enable.** If set to 1, PCI 6150 (as a master) generates PAR to cycles going across the bridge. Otherwise, PCI 6150 passes along the PAR of the cycle as stored in the internal buffers.	Yes	Yes	0
6:4	**Address Step Control.** During Type 0 Configuration cycles, PCI 6150 drives the address for the number of clocks specified by these bits, before asserting FRAME#. Defaults to 001b in Conventional PCI mode. Values: 000b = Concurrently asserts FRAME# and drives the address on the bus 001b = Asserts FRAME# one clock after driving the address on the bus … 111b = Asserts FRAME# seven clocks after driving the address on the bus	Yes	Yes; Serial EEPROM	001b
8:7	***Reserved.***	Yes	Yes	00b
9	**Prefetch Early Termination.** Values: 0 = Terminates prefetching at the Initial Prefetch Count if Flow Through is not achieved, and another Prefetching Read cycle is accepted by the PCI 6150 1 = Completes prefetching as programmed by the Prefetch Count registers, regardless of other outstanding prefetchable reads in the Transaction queue	Yes	Yes; Serial EEPROM	0
10	**Read Minimum Enable.** If set to 1, PCI 6150 initiates Read cycles only if there is sufficient space available in the FIFO, as required by the Prefetch Count registers.	Yes	Yes; Serial EEPROM	0
11	***Reserved.***	Yes	No; Serial EEPROM	0

Register 6-33. (MSCOPT; PCI:46h) Miscellaneous Options (Continued)

Bit	Description	Read	Write	Value after Reset
12	**Memory Write and Invalidate Control.** Values: 0 = Retries Memory Write and Invalidate commands if there is insufficient space for one cache line of data in the internal queues. 1 = Passes Memory Write and Invalidate commands if there are one or more cache lines of FIFO space available. If there is insufficient space, completes as a Memory Write cycle.	Yes	Yes; Serial EEPROM	0
13	**Primary Lock Enable.** If set to 1, PCI 6150 follows the LOCK protocol on primary interface; otherwise, LOCK is ignored.	Yes	Yes; Serial EEPROM	1
14	**Secondary Lock Enable.** If set to 1, PCI 6150 follows the LOCK protocol on secondary interface; otherwise, LOCK is ignored.	Yes	Yes; Serial EEPROM	1
15	***Reserved.***	Yes	No; Serial EEPROM	0

6.1.2.5 Prefetch Control

Registers 48h to 4Eh are the Flow-Through Prefetch Control registers, which are used to fine-tune the PCI 6150 Memory Read prefetch behavior. (Refer to Section 17, "PCI Flow-Through Optimization," for further details regarding these registers.)

Register 6-34. (PITLPCNT; PCI:48h) Primary Initial Prefetch Count

Bit	Description	Read	Write	Value after Reset
5:0	**Primary Initial Prefetch Count.** Controls the Initial Prefetch Count on the primary bus during reads to Prefetchable Memory space. Value must be a power of two (only one bit should be set to 1 at any time). Value is a number of Dwords. Bit 0 is Read-Only and always 0.	Yes	Yes [5:1]; Serial EEPROM	10h
7:6	***Reserved.*** Returns 00b when read.	Yes	No	00b

Register 6-35. (SITLPCNT; PCI:49h) Secondary Initial Prefetch Count

Bit	Description	Read	Write	Value after Reset
5:0	**Secondary Initial Prefetch Count.** Controls the Initial Prefetch Count on the secondary bus during reads to Prefetchable Memory space. Value must be a power of two (only one bit should be set to 1 at any time). Value is a number of Dwords. Bit 0 is Read-Only and always 0.	Yes	Yes [5:1]; Serial EEPROM	10h
7:6	***Reserved.*** Returns 00b when read.	Yes	No	00b

Register 6-36. (PINCPCNT; PCI:4Ah) Primary Incremental Prefetch Count

Bit	Description	Read	Write	Value after Reset
5:0	**Primary Incremental Prefetch Count.** Controls the Incremental Read Prefetch Count. When an entry's remaining prefetch Dword count falls below this value, the bridge prefetches an additional "Primary Incremental Prefetch Count" set of Dwords. Value must be a power of two (only one bit should be set to 1 at any time). Value is a number of Dwords. Bit 0 is Read-Only and always 0. Value must not exceed half the value programmed in the Primary Maximum Prefetch Count register (PMAXPCNT; PCI:4Ch). Otherwise, no incremental prefetch is performed.	Yes	Yes [5:1]; Serial EEPROM	10h
7:6	***Reserved.*** Returns 00b when read.	Yes	No	00b

Register 6-37. (SINCPCNT; PCI:4Bh) Secondary Incremental Prefetch Count

Bit	Description	Read	Write	Value after Reset
5:0	**Secondary Incremental Prefetch Count.** Controls the Incremental Read Prefetch Count. When an entry's remaining prefetch Dword count falls below this value, the bridge prefetches an additional "Secondary Incremental Prefetch Count" set of Dwords. Value must be a power of two (only one bit should be set to 1 at any time). Value is a number of Dwords. Bit 0 is Read-Only and always 0. Value must not exceed half the value programmed in the Secondary Maximum Prefetch Count register (SMAXPCNT; PCI:4Dh). Otherwise, no incremental prefetch is performed.	Yes	Yes [5:1]; Serial EEPROM	10h
7:6	***Reserved.*** Returns 00b when read.	Yes	No	00b

Register 6-38. (PMAXPCNT; PCI:4Ch) Primary Maximum Prefetch Count

Bit	Description	Read	Write	Value after Reset
5:0	**Primary Maximum Prefetch Count.** Applies only to PCI-to-PCI bridging. Limits the cumulative maximum count of prefetchable Dwords allocated to one entry on the primary bus when Flow Through for that entry is not achieved. Value must be an even number. Bit 0 is Read-Only and always 0. Value is specified in Dwords, *except* if 0h value is programmed, which sets the Primary Maximum Prefetch Count to its maximum value of 256 bytes. A PCI Read cycle causes a PCI request for the Maximum Count data.	Yes	Yes [5:1]; Serial EEPROM	20h
7:6	***Reserved.*** Returns 00b when read.	Yes	No	00b

Register 6-39. (SMAXPCNT; PCI:4Dh) Secondary Maximum Prefetch Count

Bit	Description	Read	Write	Value after Reset
5:0	**Secondary Maximum Prefetch Count.** Applies only to PCI-to-PCI bridging. Limits the cumulative maximum count of prefetchable Dwords allocated to one entry on the secondary bus when Flow Through for that entry is not achieved. Value must be an even number. Bit 0 is Read-Only and always 0. Value is specified in Dwords, *except* if 0h value is programmed, which sets the Secondary Maximum Prefetch Count to its maximum value of 256 bytes. A PCI Read cycle causes a PCI request for the Maximum Count data.	Yes	Yes [5:1]; Serial EEPROM	20h
7:6	***Reserved.*** Returns 00b when read.	Yes	No	00b

Register 6-40. (SFTCR; PCI:4Eh) Secondary Flow-Through Control

Bit	Description	Read	Write	Value after Reset
2:0	**Secondary Posted Write Completion Wait Count.** Maximum number of clocks the PCI 6150 waits for Posted Write data from the initiator if delivering Write data in Flow-Through mode and the Internal Post Write queues are almost empty. If the count is exceeded without additional data from the initiator, the cycle to the target is terminated and later completed. Values: 000b = Terminates the cycle if there is only one data entry remaining in the Internal Write queue 001b = De-asserts P_IRDY# and waits one clock for source data on the secondary bus, before terminating cycle … 111b = De-asserts P_IRDY# and waits seven clocks for source data on the secondary bus, before terminating cycle	Yes	Yes; Serial EEPROM	111b
3	***Reserved.*** Returns 00b when read.	Yes	No	0
6:4	**Secondary Delayed Read Completion Wait Count.** Maximum number of clocks the PCI 6150 waits for Delayed Read data from the target, if returning Read data in Flow-Through mode and the Internal Delayed Read queue is almost full. If the count is exceeded without additional space in the queue, the cycle to target is terminated, and completed when the initiator Retries the remainder of the cycle. Values: 000b = Terminates the cycle if there is only one data entry remaining in the Read queue 001b = De-asserts P_TRDY# and waits one clock for source data on the secondary bus, before terminating cycle … 111b = De-asserts P_TRDY# and waits seven clocks for source data on the secondary bus, before terminating cycle	Yes	Yes; Serial EEPROM	111b
7	***Reserved.*** Returns 00b when read.	Yes	No	0

6.1.2.6 Internal Arbiter Control

Register 6-41. (IACNTRL; PCI:50h) Internal Arbiter Control

Bit	Description	Read	Write	Value after Reset
0	**Low-Priority Group Fixed Arbitration.** If set to 1, the low-priority group uses fixed-priority arbitration; otherwise, rotating-priority arbitration is used.	Yes	Yes	0
1	**Low-Priority Group Arbitration Order.** Valid only when the low-priority arbitration group is set to a fixed arbitration scheme. Values: 0 = Priority decreases with bus master number. (*For example*, assuming Master 2 is set as the highest priority master, Master 3 retains higher priority than Master 4.) 1 = Priority increases with bus master number. (*For example*, assuming Master 2 is set as the highest priority master, Master 4 retains higher priority than Master 3. This order is relative to the master with the highest priority for this group, as specified in IACNTRL[7:4].	Yes	Yes	0
2	**High-Priority Group Fixed Arbitration.** If set to 1, the high-priority group uses the fixed-priority arbitration; otherwise, rotating-priority arbitration is used**.**	Yes	Yes	0
3	**High-Priority Group Arbitration Order.** Valid only when the high-priority arbitration group is set to a fixed arbitration scheme. Values: 0 = Priority decreases with bus master number. (*For example*, assuming Master 2 is set as the highest priority master, Master 3 retains higher priority than Master 4.) 1 = Priority increases with bus master number. (*For example,* assuming Master 2 is set as the highest priority master, Master 4 retains higher priority than Master 3.) This order is relative to the master with the highest priority for this group, as specified in IACNTRL[11:8].	Yes	Yes	0
7:4	**Highest Priority Master in Low-Priority Group.** Controls which master in the low-priority group retain the highest priority. Valid only if the group uses the fixed arbitration scheme. Values: 0000b = Master 0 retains highest priority 0001b = Master 1 retains highest priority … 1001b = PCI 6150 retains highest priority 1010b – 1111b = ***Reserved***	Yes	Yes	0000b

Register 6-41. (IACNTRL; PCI:50h) Internal Arbiter Control (Continued)

Bit	Description	Read	Write	Value after Reset
11:8	**Highest Priority Master in High-Priority Group.** Controls which master in the high-priority group retains the highest priority. Valid only if the group uses the fixed arbitration scheme. Values: 0000b = Master 0 retains highest priority 0001b = Master 1 retains highest priority … 1001b = PCI 6150 retains highest priority 1010b – 1111b = ***Reserved***	Yes	Yes	0000b
15:12	**Bus Grant Parking Control.** Controls bus grant behavior during idle. Values: 0000b = Last master granted is parked 0001b = Master 0 is parked … 1001b = Master 8 is parked 1010b = PCI 6150 is parked All other values de-assert the grant (no parking).	Yes	Yes	0000b

6.1.2.7 Test and Serial EEPROM

Register 6-42. (TEST; PCI:52h) Test

Bit	Description	Read	Write	Value after Reset
0	**Serial EEPROM Autoload Control.** If set to 1, disables serial EEPROM autoload. ***For Test Use Only.*** To stop serial EEPROM load, write 1 to this bit within 1200 clocks after P_RSTIN# goes high.	Yes	Yes	0
1	**Fast Serial EEPROM Autoload.** If set to 1, speeds up serial EEPROM autoload by 32 times. ***For Test Use Only.*** To enable Fast serial EEPROM load, write 1 to this bit within 1200 clocks after P_RSTIN# goes high.	Yes	Yes	0
2	**Serial EEPROM Autoload Status.** Serial EEPROM autoload status is set to 1 during autoload.	Yes	No	Serial EEPROM Autoload Status
7:3	***Reserved.***	Yes	No	0h

Register 6-43. (EEPCNTRL; PCI:54h) Serial EEPROM Control

Bit	Description	Read	Write	Value after Reset
0	**Start.** Starts serial EEPROM Read or Write cycle. Bit is cleared when serial EEPROM load completes.	Yes	Yes	0
1	**Serial EEPROM Command.** Controls commands sent to the serial EEPROM. Values: 0 = Read 1 = Write	Yes	Yes	0
2	**Serial EEPROM Error.** Set to 1 if serial EEPROM ACK was not received during serial EEPROM cycle.	Yes	No	—
3	**Serial EEPROM Autoload Successful.** Set to 1 if serial EEPROM autoload successfully occurred after reset, with appropriate Configuration registers loaded with the values programmed in the serial EEPROM. If 0, the serial EEPROM autoload was unsuccessful or disabled.	Yes	No	—
5:4	***Reserved.*** Returns 00b when read.	Yes	No	00b
7:6	**Serial EEPROM Clock Rate.** Controls the serial EEPROM clock frequency. The serial EEPROM clock is derived from the primary PCI clock. Values: 00 = PCLK / 1024 (Used for 66 Mhz PCI) 01 = PCLK / 512 10 = PCLK / 256 11 = PCLK / 32 (Test mode use only)	Yes	Yes	00b

Register 6-44. (EEPADDR; PCI:55h) Serial EEPROM Address

Bit	Description	Read	Write	Value after Reset
0	***Reserved.***	Yes	No	—
7:1	**Serial EEPROM Address.** Word address for the serial EEPROM cycle.	Yes	Yes	—

Register 6-45. (EEPDATA; PCI:56h) Serial EEPROM Data

Bit	Description	Read	Write	Value after Reset
15:0	**Serial EEPROM Data.** Contains data to be written to the serial EEPROM. During reads, contains data received from the serial EEPROM after a Read cycle completes.	Yes	Yes	—

6.1.2.8 Primary System Error Event

Register 6-46. (PSERRED; PCI:64h) P_SERR# Event Disable

Bit	Description	Read	Write	Value after Reset
0	***Reserved.***	Yes	No	0
1	**Posted Write Parity Error.** Controls PCI 6150 ability to assert P_SERR# when a Data Parity error is detected on the target bus during a Posted Write transaction. P_SERR# is asserted if this event occurs when bit is 0 and Command register P_SERR# Enable bit is set (PCICR[8]=1; PCI:04h).	Yes	Yes	0
2	**Posted Memory Write Non-Delivery.** Controls PCI 6150 ability to assert P_SERR# when it is unable to deliver Posted Write data after 2^{24} attempts [or programmed Maximum Retry count (TOCNTRL[2:0]; PCI:45h)]. P_SERR# is asserted if this event occurs when bit is 0 and Command register P_SERR# Enable bit is set (PCICR[8]=1; PCI:04h).	Yes	Yes	0
3	**Target Abort during Posted Write.** Controls PCI 6150 ability to assert P_SERR# when it receives a Target Abort while attempting to deliver Posted Write data. P_SERR# is asserted if this event occurs when bit is 0 and Command register P_SERR# Enable bit is set (PCICR[8]=1; PCI:04h)	Yes	Yes	0
4	**Master Abort on Posted Write.** Controls PCI 6150 ability to assert P_SERR# when it receives a Master Abort while attempting to deliver Posted Write data. P_SERR# is asserted if this event occurs when bit is 0 and Command register P_SERR# Enable bit is set (PCICR[8]=1; PCI:04h).	Yes	Yes	0
5	**Delayed Configuration or I/O Write Non-Delivery.** Controls PCI 6150 ability to assert P_SERR# when it is unable to deliver Delayed Write data after 2^{24} attempts [or programmed Maximum Retry count (TOCNTRL[2:0]; PCI:45h)]. P_SERR# is asserted if this event occurs when bit is 0 and Command register P_SERR# Enable bit is set (PCICR[8]=1; PCI:04h).	Yes	Yes	0
6	**Delayed Read-No Data from Target.** Controls PCI 6150 ability to assert P_SERR# when it is unable to transfer Read data from the target after 2^{24} attempts [or programmed Maximum Retry count (TOCNTRL[2:0]; PCI:45h)]. P_SERR# is asserted if this event occurs when bit is 0 and Command register P_SERR# Enable bit is set (PCICR[8]=1; PCI:04h).	Yes	Yes	0
7	***Reserved.*** Returns 0 when read.	Yes	No	0

6.1.2.9 GPIO

Register 6-47. (GPIOOD; PCI:65h) GPIO[3:0] Output Data

Bit	Description	Read	Write	Value after Reset
3:0	**GPIO[3:0] Output Data Write 1 to Clear.** Writing 1 to these bits drives the corresponding signal low on the GPIO[3:0] bus, if the signal is programmed as an output. Writing 0 has no effect. Read returns last written value.	Yes	Yes/Set Low	0h
7:4	**GPIO[3:0] Output Data Write 1 to Set.** Writing 1 to these bits drives the corresponding signal high on the GPIO[3:0] bus, if the signal is programmed as an output. Writing 0 has no effect. Read returns last written value.	Yes	Yes/Set High	0h

Register 6-48. (GPIOOE; PCI:66h) GPIO[3:0] Output Enable

Bit	Description	Read	Write	Value after Reset
3:0	**GPIO[3:0] Output Enable Write 1 to Clear.** Writing 1 to these bits configures the corresponding signal on the GPIO[3:0] bus as an input. Writing 0 has no effect. Read returns last written value.	Yes	Yes/Set Low	0h
7:4	**GPIO[3:0] Output Enable Write 1 to Set.** Writing 1 to these bits configures the corresponding signal on the GPIO[3:0] bus as an output. Writing 0 has no effect. Read returns last written value.	Yes	Yes/Set High	0h

Register 6-49. (GPIOID; PCI:67h) GPIO[3:0] Input Data

Bit	Description	Read	Write	Value after Reset
3:0	***Reserved.***	Yes	No	0h
7:4	**GPIO[3:0] Input Data.** Reads the GPIO[3:0] pin state. The state is updated on the primary PCI Clock cycle, following a change in GPIO[3:0] state.	Yes	No	—

6.1.2.10 Secondary Clock Control

Register 6-50. (SCLKCNTRL; PCI:68h) Secondary Clock Control

Bit	Description	Read	Write	Value after Reset
1:0	**Clock 0 Disable**. If either bit is 0, S_CLKO0 is enabled. When both bits are 1, S_CLKO0 is disabled. Upon secondary bus reset, shifting in a serial data stream initializes this bit. These bits are assigned to correspond to the Philips 74F166 P_RSNT0[2:1]# slot 0 pins.	Yes	Yes	00b
3:2	**Clock 1 Disable.** If either bit is 0, S_CLKO1 is enabled. When both bits are 1, S_CLKO1 is disabled. Upon secondary bus reset, shifting in a serial data stream initializes this bit. These bits are assigned to correspond to the Philips 74F166 P_RSNT1[2:1]# slot 1 pins.	Yes	Yes	00b
5:4	**Clock 2 Disable.** If either bit is 0, S_CLKO2 is enabled. When both bits are 1, S_CLKO2 is disabled. Upon secondary bus reset, shifting in a serial data stream initializes this bit. These bits are assigned to correspond to the Philips 74F166 P_RSNT2[2:1]# slot 2 pins.	Yes	Yes	00b
7:6	**Clock 3 Disable.** If either bit is 0, S_CLKO3 is enabled. When both bits are 1, S_CLKO3 is disabled. Upon secondary bus reset, shifting in a serial data stream initializes this bit. These bits are assigned to correspond to the Philips 74F166 P_RSNT3[2:1]# slot 3 pins.	Yes	Yes	00b
8	**Clock 4 Disable.** If 0, S_CLKO4 is enabled. When 1, S_CLKO4 is disabled. Upon secondary bus reset, shifting in a serial data stream initializes this bit.	Yes	Yes	0
9	**Clock 5 Disable.** If 0, S_CLKO5 is enabled. When 1, S_CLKO5 is disabled. Upon secondary bus reset, shifting in a serial data stream initializes this bit.	Yes	Yes	0
10	**Clock 6 Disable.** If 0, S_CLKO6 is enabled. When 1, S_CLKO6 is disabled. Upon secondary bus reset, shifting in a serial data stream initializes this bit.	Yes	Yes	0
11	**Clock 7 Disable.** If 0, S_CLKO7 is enabled. When 1, S_CLKO7 is disabled. Upon secondary bus reset, shifting in a serial data stream initializes this bit.	Yes	Yes	0
12	**Clock 8 Disable.** If 0, S_CLKO8 is enabled. When 1, S_CLKO8 is disabled. Upon secondary bus reset, shifting in a serial data stream initializes this bit.	Yes	Yes	0
13	**Clock 9 Disable.** If 0, S_CLKO9 is enabled. When 1, S_CLKO9 is disabled. Upon secondary bus reset, shifting in a serial data stream initializes this bit.	Yes	Yes	0
15:14	***Reserved.***	Yes	No	00b

6.1.2.11 Primary System Error Status

Register 6-51. (PSERRSR; PCI:6Ah) P_SERR# Status

Bit	Description	Read	Write	Value after Reset
0	**Address Parity Error.** P_SERR# is asserted because an Address Parity error occurred on either side of the bridge.	Yes	Yes/Clr	0
1	**Posted Write Data Parity Error.** P_SERR# is asserted because a Posted Write Data Parity error occurred on the target bus.	Yes	Yes/Clr	0
2	**Posted Write Non-Delivery.** P_SERR# is asserted because PCI 6150 was unable to deliver Posted Write data to the target before the Timeout Counter expired.	Yes	Yes/Clr	0
3	**Target Abort during Posted Write.** P_SERR# is asserted because PCI 6150 received a Target Abort when delivering Posted Write data.	Yes	Yes/Clr	0
4	**Master Abort during Posted Write.** P_SERR# is asserted because PCI 6150 received a Master Abort when delivering Posted Write data.	Yes	Yes/Clr	0
5	**Delayed Write Non-Delivery.** P_SERR# is asserted because PCI 6150 was unable to deliver Delayed Write data before the Timeout Counter expired.	Yes	Yes/Clr	0
6	**Delayed Read Failed.** P_SERR# is asserted because PCI 6150 was unable to read data from the target before the Timeout Counter expired.	Yes	Yes/Clr	0
7	**Delayed Transaction Master Timeout.** P_SERR# is asserted because a master did not repeat a Read or Write transaction before the initiator bus Master Timeout Counter expired.	Yes	Yes/Clr	0

6.1.2.12 Read-Only Register Control

Register 6-52. (RRC; PCI:9Ch) Read-Only Register Control

Bit	Description	Read	Write	Value after Reset
6:0	***Reserved.***	Yes	No	0h
7	**Read-Only Registers Write Enable.** Setting this bit to 1 enables writes to specific bits within these normally Read-Only registers (refer to the listed registers for further details): • Vendor and Device IDs (PCIIDR; PCI:00h) • PCI Class Code (PCICCR; PCI:09h – 0Bh) • PCI Header Type (PCIHTR; PCI:0Eh) • PCI Built-In Self-Test (PCIBISTR; PCI:0Fh) • Power Management Capabilities (PMC; PCI:DEh) • Power Management Control/Status (PMCSR; PCI:E0h) • Power Management Data (PMCDATA; PCI:E3h) Bit must be cleared after the values are modified in these Read-Only registers.	Yes	Yes	0

6.1.2.13 Power Management Capability

Specific bits in the PMC; PCI:DEh, PMCDATA; PCI:E3h, and PMCSR; PCI:E0h Power Management registers are normally Read-Only. However, their default values can be changed by firmware or software by setting the Read-Only Registers Write Enable bit (RRC[7]=1; PCI:9Ch). After modifying these registers, the Write Enable bit must be cleared to preserve the Read-Only nature of these registers. It should be noted that the RRC[7] state does ***not*** affect Write accesses to PMCSR[15, 8].

.

Register 6-53. (PMCAPID; PCI:DCh) Power Management Capability ID

Bit	Description	Read	Write	Value after Reset
7:0	**Power Management Capability ID.** PCI-SIG-issued Capability ID for Power Management is 1h.	Yes	No	1h

Register 6-54. (PMNEXT; PCI:DDh) Power Management Next Capability Pointer

Bit	Description	Read	Write	Value after Reset
7:0	**Next_Cap Pointer.** Provides an offset into PCI Configuration space for the Hot Swap capability location in the New Capabilities Linked List (E4h).	Yes	No	E4h

Register 6-55. (PMC; PCI:DEh) Power Management Capabilities

Bit	Description	Read	Write	Value after Reset
2:0	**Version.** Set to 001b, which indicates that this function complies with *PCI Power Mgmt. r1.1*.	Yes	Only if RRC[7]=1; Serial EEPROM	001b
3	**PME Clock.** Set to 0, because PCI 6150 does not support PME# signaling.	Yes	Only if RRC[7]=1; Serial EEPROM	0
4	**Auxiliary Power Source.** Set to 0, because PCI 6150 does not support PME# signaling.	Yes	Only if RRC[7]=1; Serial EEPROM	0
5	**Device-Specific Initialization (DSI).** Returns 0, indicating PCI 6150 does not require special initialization.	Yes	Only if RRC[7]=1; Serial EEPROM	0
8:6	***Reserved.***	Yes	No	000b
9	**D_1 Support.** Returns 1, indicating that PCI 6150 supports the D_1 device power state.	Yes	Only if RRC[7]=1; Serial EEPROM	1
10	**D_2 Support.** Returns 1, indicating that PCI 6150 supports the D_2 device power state.	Yes	Only if RRC[7]=1; Serial EEPROM	1
15:11	**PME Support.** Set to 0h, indicating that PME# is not supported.	Yes	Only if RRC[7]=1; Serial EEPROM	0h

Register 6-56. (PMCSR; PCI:E0h) Power Management Control/Status

Bit	Description	Read	Write	Value after Reset
1:0	**Power State.** Used to determine the current power state of a function and to set the function into a new power state. Values: 00b = D_0 (default) 01b = D_1; valid only if PMC[9]=1; PCI:82h 10b = D_2; valid only if PMC[10]=1; PCI:82h 11b = D_{3hot}; if BPCC_EN=1, S_CLKO[9:0] are stopped	Yes	Yes; Serial EEPROM	00b
7:2	***Reserved.***	Yes	No	0h
8	**PME Enable.** Set to 0, because PCI 6150 does not support PME# signaling.	Yes	Yes Serial EEPROM	0
12:9	**Data Select.** Returns 0h, indicating PCI 6150 does not return dynamic data.	Yes	Only if RRC[7]=1; Serial EEPROM	0h
14:13	**Data Scale.** Returns 00b when read, as the PCI 6150 does not return dynamic data.	Yes	No; Serial EEPROM	00b
15	**PME Status.** Set to 0, because PCI 6150 does not support PME# signaling.	Yes	Yes; Serial EEPROM	0

Register 6-57. (PMCSR_BSE; PCI:E2h) PMCSR Bridge Supports Extensions

Bit	Description	Read	Write	Value after Reset
5:0	***Reserved.***	Yes	No	0h
6	**B_2/B_3 Support for D_{3hot}.** Reflects BPCC_EN input pin state. Value of 1 indicates that when PCI 6150 is programmed to D_{3hot} state, the secondary bus clock is stopped.	Yes	No	—
7	**Bus Power Control Enable.** Reflects BPCC_EN input pin state. Value of 1 indicates that the secondary bus Power Management state follows that of PCI 6150, with one exception—D_{3hot}.	Yes	No	—

Register 6-58. (PMCDATA; PCI:E3h) Power Management Data

Bit	Description	Read	Write	Value after Reset
7:0	**Power Management Data.** Used to report the state-dependent data requested by PMCSR[12:9]. Value is scaled by the value reported by PMCSR[14:13]; PCI:E0h.	Yes	Only if RRC[7]=1; Serial EEPROM	0h

6.1.2.14 Hot Swap Capability

Register 6-59. (HS_CNTL; PCI:E4h) Hot Swap Control

Bit	Description	Read	Write	Value after Reset
7:0	**Hot Swap Capability ID.** PCI-SIG-issued Capability ID for Hot Swap is 06h.	Yes	No	06h

Register 6-60. (HS_NEXT; PCI:E5h) Hot Swap Next Capability Pointer

Bit	Description	Read	Write	Value after Reset
7:0	**Next_Cap Pointer.** Provides an offset into PCI Configuration space for the VPD capability location in the New Capabilities Linked List (E8h).	Yes	No	E8h

Register 6-61. (HS_CSR; PCI:E6h) Hot Swap Control/Status

Bit	Description	Read	Write	Value after Reset
0	**Device Hiding Arm (DHA).** DHA is set to 1 by hardware when the Hot Swap port PCI RSTIN# becomes inactive and the handle switch remains unlocked. Handle locking clears this bit. Values: 0 = Disarm Device Hiding 1 = Arm Device Hiding	Yes	Yes	0
1	**ENUM# Mask Status (EIM).** Enables or disables ENUM# assertion. Values: 0 = Enables ENUM# assertion 1 = Masks ENUM# assertion	Yes	Yes	0
2	**Pending INSert or EXTract (PIE).** Set when INS or EXT is 1 or INS is armed (write 1 to EXT bit). Values: 0 = Neither is pending 1 = In insertion or extraction is in progress	Yes	No	—
3	**LED Status (LOO).** Indicates whether LED is ON or OFF. Values: 0 = LED OFF 1 = LED ON	Yes	Yes	0
5:4	**Programming Interface (PI).** Hardcoded at 01b—INS, EST, LOO, EIM, PIE, and Device Hiding supported. Upon RSTIN# assertion, the PCI 6150 turns ON the LED. After RSTIN# de-assertion, the LED remains ON until the eject switch (handle) is closed, then the PCI 6150 turns OFF the LED.	Yes	No	01b
6	**Extraction State (EXT).** Set by hardware when the ejector handle is unlocked and INS=0.	Yes	Yes/Clr	—
7	**Insertion State (INS).** Set by hardware when the Hot Swap port RSTIN# is de-asserted, serial EEPROM autoload is completed, and ejector handle is locked. Writing 1 to EXT bit also arms INS.	Yes	Yes/Clr	—
15:8	***Reserved.***	Yes	No	0h

6.1.2.15 VPD Capability

Register 6-62. (PVPDID; PCI:E8h) Vital Product Data Capability ID

Bit	Description	Read	Write	Value after Reset
7:0	**Vital Product Data Capability ID.** PCI-SIG-issued Capability ID for VPD is 03h.	Yes	No	03h

Register 6-63. (PVPD_NEXT; PCI:E9h) Vital Product Data Next Capability Pointer

Bit	Description	Read	Write	Value after Reset
7:0	**Next_Cap Pointer**. Provides offset into PCI Configuration space for the Next Capability location in the New Capabilities Linked List (00h). ***Note:*** *00h indicates the end of the New Capabilities Linked List.*	Yes	No	00h

Register 6-64. (PVPDAD; PCI:EAh) Vital Product Data Address

Bit	Description	Read	Write	Value after Reset
1:0	***Reserved.***	Yes	No	00b
7:2	**VPD Address.** Offset into the serial EEPROM to location where data is written and read. PCI 6150 accesses the serial EEPROM at address PVPDAD[7:2]+40h. The 40h offset ensures that VPD accesses do not overwrite the PCI 6150 serial EEPROM Configuration data stored in serial EEPROM locations 00h to 3Fh.	Yes	Yes	0
14:8	***Reserved.***	Yes	No	0h
15	**VPD Operation**. Writing 0 generates a Read cycle from the serial EEPROM at the VPD address specified in PVPDAD[7:2]. This bit remains at logic 0 until the serial EEPROM cycle is complete, at which time the bit is set to 1. Data for reads is available in the VPD Data register (PVPDATA; PCI:ECh). Writing 1 generates a Write cycle to the serial EEPROM at the VPD address specified in PVPDAD[7:2]. Remains at logic 1, until the serial EEPROM cycle is completed, at which time the bit is cleared to 0. Place data for writes into the VPD Data register.	Yes	Yes	0

Register 6-65. (PVPDATA; PCI:ECh) VPD Data

Bit	Description	Read	Write	Value after Reset
31:0	**VPD Data (Serial EEPROM Data).** The least significant byte of this register corresponds to the byte of VPD at the address specified by the VPD Address register (PVPDAD[7:2]; PCI:A2h). Data is read from or written to PVPDATA, using standard Configuration accesses.	Yes	Yes	0h

7 SERIAL EEPROM

This section describes information specific to the PCI 6150 serial EEPROM interface and use—access, Autoload mode, and data structure.

7.1 OVERVIEW

Important Note: *Erroneous serial EEPROM data can cause the PCI 6150 to lock the system. Provide an optional switch or jumper to disable the serial EEPROM in board designs.*

The PCI 6150 provides a two-wire interface to a serial EEPROM device. The interface can control an ISSI IS24C02 or compatible part, which is organized as 256 x 8 bits. The serial EEPROM is used to initialize the internal PCI 6150 registers, and alleviates the need for user software to configure the PCI 6150. If a programmed serial EEPROM is connected, the PCI 6150 automatically loads data from the serial EEPROM after P_RSTIN# de-assertion.

The serial EEPROM data structure is defined in Section 7.4.1. The serial EEPROM interface is organized on a 16-bit base in Little Endian format, and the PCI 6150 supplies a 7-bit serial EEPROM Word address.

The following pins are used for the serial EEPROM interface:

- **EEPCLK**—Serial EEPROM clock output
- **EEPDATA**—Serial EEPROM bi-directional serial data
- **EE_EN#—**Low input enables serial EEPROM access

Note: *The PCI 6150 does not control the serial EEPROM A0 to A2 address inputs. It can only access serial EEPROM addresses set to 0.*

7.2 SERIAL EEPROM ACCESS

The PCI 6150 can access the serial EEPROM on a Word basis, using the hardware sequencer. Users access one Word data by way of the PCI 6150 Serial EEPROM Control register:

- Serial EEPROM Start/Read/Write Control (EEPCNTRL; PCI:54h)
- Serial EEPROM Address (EEPADDR; PCI:55h)
- Serial EEPROM Data (EEPDATA; PCI:56h)

Before each access, software should check the Auto Mode Cycle in Progress status (EEPCNTRL[0]; PCI:54h, same bit as Start) before issuing the next Start. The following is the general procedure for Read/ Write Serial EEPROM accesses:

1. Program the Serial EEPROM Address register (EEPADDR; PCI:55h) with the Word address.
2. **Writes**—Program Word data to the Serial EEPROM Data register (EEPDATA; PCI:56h).

   Reads—Proceed to the next step.
3. **Writes**—Set the Serial EEPROM Command and Start bits (EEPCNTRL[1:0]=11b; PCI:54h, respectively) to start the Serial EEPROM Sequencer.

   Reads—Set the Start bit (EEPCNTRL[1:0]=01b; PCI:54h) to start the Serial EEPROM Sequencer.
4. When the serial EEPROM read/write is complete, as indicated by the bit value of 0 (Serial EEPROM Control register, EEPCNTRL[0]=0; PCI:54h):

   Writes—Data was successfully written to the serial EEPROM.

   Reads—Data was loaded into the Serial EEPROM Data register (EEPDATA; PCI:56h) by the serial EEPROM sequencer.

7.3 SERIAL EEPROM AUTOLOAD MODE AT RESET

Upon P_RSTIN# going high at Reset, the PCI 6150 autoloads input for the serial EEPROM autoload condition if EE_EN#=0.

The PCI 6150 initially reads the first offset in the serial EEPROM, which should contain a valid signature value of 1516h. If the signature is correct, register autoload immediately commences after reset. During autoload, the PCI 6150 reads sequential words from the serial EEPROM and writes to the appropriate registers. If a blank serial EEPROM is connected, the PCI 6150 stops loading the serial EEPROM contents after reading the first word, as the serial EEPROM's signature is not valid. Likewise, if no serial EEPROM is connected, the PCI 6150 also stops loading the serial EEPROM contents after attempting to read the first word.

Before the PCI 6150 registers can be accessed by way of the host, check the auto-load condition by reading the EEPAUTO bit. Host access is allowed only after the EEPAUTO status becomes 0, which means that the auto load initialization sequence is complete.

The serial EEPROM initialized value is cleared by an active P_RSTIN# or Power Management-initiated internal reset.

7.4 SERIAL EEPROM DATA STRUCTURE

Following reset and the previously described conditions, the PCI 6150 autoloads the registers with serial EEPROM data. Figure 7-1 illustrates the serial EEPROM data structure.

The PCI 6150 accesses the serial EEPROM, one word at a time. **It is important to note that in the Data phase, bit orders are the reverse of that in the Address phase. The PCI 6150 supports only Serial EEPROM Device Address 0.**

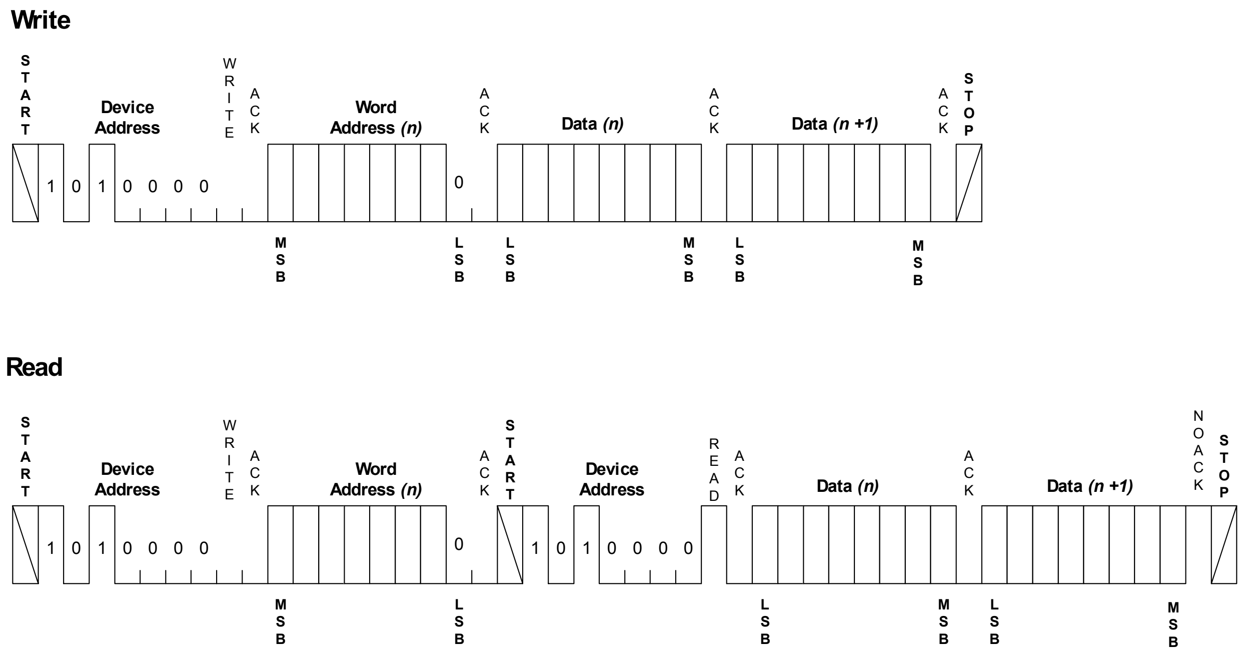


Figure 7-1. Serial EEPROM Data Structure

7.4.1 Serial EEPROM Address and Corresponding PCI 6150 Registers

Table 7-1. Serial EEPROM Address and Corresponding PCI 6150 Registers

Serial EEPROM Byte Address	PCI Configuration Offset	Description
00h – 01h	—	**Serial EEPROM Signature.** Autoload proceeds only if it reads a value of 1516h on the first word loaded. Value: 1516h = Valid signature; otherwise, disables autoloading.
02h	—	**Region Enable.** Enables or disables certain regions of the PCI Configuration space from being loaded from the serial EEPROM. Valid combinations are: Bit 0 = ***Reserved.*** Bits [4:1] = 0000b = Stops autoload at serial EEPROM offset 03h = Group 1. 0001b = Stops autoload at serial EEPROM offset 13h = Group 2. 0011b = Stops autoload at serial EEPROM offset 23h = Group 3. 0111b, 1111b = ***Reserved.*** Other combinations are undefined. Bits [7:5] = ***Reserved.***
03h	—	**Enable Miscellaneous Functions.** Bit 0 = ISA Enable Control bit Write Protect. When set, PCI 6150 changes the standard PCI-to-PCI Bridge Control register (BCNTRL[2]; PCI:3Eh) to Read-Only. The ISA Enable feature is then ***not*** available. Bits [7:1] = ***Reserved.***
End of Group 1		
04h – 05h	00h – 01h	**Vendor ID (PCIIDR[15:0]).**
06h – 07h	02h – 03h	**Device ID (PCIIDR[31:16]).**
08h	—	***Reserved.***
09h	09h	**Class Code.** Contains low byte of Class Code register (PCICCR[7:0]).
0Ah – 0Bh	0Ah – 0Bh	**Class Code Higher Bytes.** Contains upper bytes of Class Code register (PCICCR[23:8]).
0Ch	0Eh	**Header Type (PCIHTR).**
0Dh	09h	***Reserved.***
0Eh – 0Fh	0Ah – 0Bh	***Reserved.***
10h	0Eh	***Reserved.***
11h	0Fh	**Built-In Self Test (BIST) (PCIBISTR).** Set to 0.
12h – 13h	50h	**Internal Arbiter Control (IACNTRL).**
End of Group 2		

Table 7-1. Serial EEPROM Address and Corresponding PCI 6150 Registers (Continued)

Serial EEPROM Byte Address	PCI Configuration Offset	Description
14h	44h	**Primary Flow-Through Control (PFTCR).**
15h	45h	**Timeout Control (TOCNTRL).**
16h – 17h	46h – 47h	**Miscellaneous Options (MSCOPT).**
18h	48h	**Primary Initial Prefetch Count (PITLPCNT).**
19h	49h	**Secondary Initial Prefetch Count (SITLPCNT).**
1Ah	4Ah	**Primary Incremental Prefetch Count (PINCPCNT).**
1Bh	4Bh	**Secondary Incremental Prefetch Count (PINCPCNT).**
1Ch	4Ch	**Primary Maximum Prefetch Count (PMAXPCNT).**
1Dh	4Dh	**Secondary Maximum Prefetch Count (PMAXPCNT).**
1Eh	4Eh	**Secondary Flow-Through Control (SFTCR).**
1Fh	E3h	**Power Management Data (PMCDATA).**
20h – 21h	E0h	**Power Management Control/Status (PMCSR).**
22h – 23h	DEh	**Power Management Capabilities (PMC).**
End of Group 3		
26h – 3Fh	—	***Reserved.* Must be set to 0.**

Note: ** These addresses will be defined in the next databook release.*

8 PCI BUS OPERATION

This section describes PCI transactions to which the PCI 6150 responds and those it initiates when operating with one or both of its interfaces.

8.1 TRANSACTIONS

Table 8-1 lists the PCI command codes and transaction types to which the PCI 6150 responds and initiates. The *Master* and *Target* columns indicate support for transactions wherein the PCI 6150 initiates transactions as a master, and responds to transactions as a target, on the primary and secondary buses.

Table 8-1. PCI Transactions

CBE[3:0]#	Transaction Type	Initiates as Master		Responds as Target	
		Primary	Secondary	Primary	Secondary
0000b	Interrupt Acknowledge ***(Not Supported)***	N	N	N	N
0001b	Special Cycle ***(Not Supported)***	Y	Y	N	N
0010b	I/O Read	Y	Y	Y	Y
0011b	I/O Write	Y	Y	Y	Y
0100b	***Reserved***	N	N	N	N
0101b	***Reserved***	N	N	N	N
0110b	Memory Read	Y	Y	Y	Y
0111b	Memory Write	Y	Y	Y	Y
1000b	***Reserved***	N	N	N	N
1001b	***Reserved***	N	N	N	N
1010b	Configuration Read	N	Y	Y	N
1011b	Configuration Write	Type 1	Y	Y	Type 1
1100b	Memory Read Multiple	Y	Y	Y	Y
1101b	Dual Address Cycle (DAC)	Y	Y	Y	Y
1110b	Memory Read Line	Y	Y	Y	Y
1111b	Memory Write and Invalidate	Y	Y	Y	Y

As indicated in Table 8-1, the PCI 6150 does not support the following PCI commands—it ignores them and reacts to these commands as follows:

- ***Reserved***—The PCI 6150 does not generate ***reserved*** command codes.
- **Interrupt Acknowledge**—The PCI 6150 never initiates an Interrupt Acknowledge transaction and, as a target, it ignores Interrupt Acknowledge transactions. Interrupt Acknowledge transactions are expected to reside entirely on the primary PCI Bus closest to the host bridge.
- **Special Cycle**—The PCI 6150 does not respond to Special Cycle transactions. To generate Special Cycle transactions on other PCI Buses (downstream or upstream), use a Type 1 Configuration command.
- **Type 0 Configuration Write**—The PCI 6150 does not generate Type 0 Configuration Write transactions on the primary interface.

8.2 SINGLE ADDRESS PHASE

The PCI 6150 32-bit address uses a single Address phase. This address is driven on AD[31:0], and the bus command is driven on P_CBE[3:0]#.

The PCI 6150 supports only the linear increment Address mode, which is indicated when the lower two Address bits equal 00b. If either of the lower two Address bits is equal to a non-zero value, the PCI 6150 automatically Disconnects the transaction after the first Data transfer.

8.3 DUAL ADDRESS PHASE

The PCI 6150 supports the Dual Address Cycle (DAC) bus command to transfer 64-bit addresses. In DAC transactions, the first Address phase occurs during the initial FRAME# assertion, and the second Address phase occurs one clock later. During the first Address phase, the DAC command is presented on CBE[3:0]#, and the lower 32 bits of the address on AD[31:0]. The second Address phase retains the cycle command on CBE[3:0]#, and the upper 32 bits of the address on AD[31:0].

DACs are used to access locations that are not in the first 4 GB of PCI Memory space. Addresses in the first 4 GB of PCI Memory space always use a Single Address Cycle (SAC).

The PCI 6150 supports DACs in the downstream and upstream directions. The PCI 6150 responds to DACs for the following commands only:

- Memory Write
- Memory Write and Invalidate
- Memory Read
- Memory Read Line
- Memory Read Multiple

The PCI 6150 forwards DACs downstream when their addresses fall within Prefetchable Memory space. DACs originating on the secondary bus, with addresses outside Prefetchable Memory space, are forwarded upstream.

8.4 DEVICE SELECT (DEVSEL#) GENERATION

The PCI 6150 performs positive address decoding when accepting transactions on the primary or secondary bus. The PCI 6150 never subtractively decodes. Medium DEVSEL# timing is used for 33 MHz operation. Slow DEVSEL# timing is used for 66 MHz operation.

8.5 DATA PHASE

Depending on the command type, the PCI 6150 can support multiple Data phase PCI transactions. Write transactions are treated as Posted Write or Delayed Write transactions.

Table 8-2 lists the forwarding method used for each type of Write operation.

Table 8-2. Write Transaction Forwarding

Transaction Type	Forwarding Method
Memory Write	Posted
Memory Write and Invalidate	
I/O Write	Delayed
Type 1 Configuration Write	

8.5.1 Posted Write Transactions

When the PCI 6150 determines that a Memory Write transaction is to be forwarded across the bridge, the PCI 6150 asserts DEVSEL# with slow timing and TRDY# in the same cycle, provided that sufficient Buffer space is available in the Posted Write Data queue, and that the queue contains fewer than four outstanding Posted transactions. The PCI 6150 can accept one dual-Dword of Write data every PCI Clock cycle (*that is*, no target wait states are inserted). Up to 256 bytes of Posted Write data are stored in internal Posted Write buffers and eventually delivered to the target.

The PCI 6150 continues to accept Write data until one of the following occurs:

- Initiator normally terminates the transaction
- Cache Line boundary or an aligned 4-KB boundary is reached, depending on transaction type
- Posted Write Data buffer fills

When one of the last two events occurs, the PCI 6150 returns a Target Disconnect to the requesting initiator on this Data phase to terminate the transaction.

After the Posted Write transaction is selected for completion, the PCI 6150 requests ownership of the target bus. This can occur while the PCI 6150 is receiving data on the initiator bus. After the PCI 6150 has ownership of the target bus, and the target bus is detected in the idle condition, the PCI 6150 initiates the Write cycle and continues to transfer Write data until all Write data corresponding to that transaction is delivered, or a Target Termination is received. If Write data exists in the queue, the PCI 6150 can drive one dual-Dword of Write data each PCI Clock cycle. If Write data is flowing through the PCI 6150 and the initiator stalls, the PCI 6150 inserts wait states on the target bus if the queue empties.

The PCI 6150 ends the transaction on the target bus when one of the following conditions is met:

- All Posted Write data was delivered to the target
- Target returns a Target Disconnect or Retry (the PCI 6150 starts another transaction to deliver the remaining Write data)
- Target returns a Target Abort (the PCI 6150 discards remaining Write data)

The Master Latency Timer expires, and the PCI 6150 no longer retains the target bus grant (the PCI 6150 starts another transaction to deliver the remaining Write data).

8.5.2 Memory Write and Invalidate Transactions

Memory Write and Invalidate transactions guarantee the transfer of entire cache lines. By default, the PCI 6150 Retries a Memory Write and Invalidate cycle until there is space for one or more cache lines of data in the internal buffers. The PCI 6150 then completes the transaction on the secondary bus as a Memory Write and Invalidate cycle. The PCI 6150 can also be programmed to accept Memory Write and Invalidate cycles under the same conditions as normal Memory Writes. In this case, if the Write buffer fills before an entire cache line is transferred, the PCI 6150 Disconnects and completes the Write cycle on the secondary bus as a normal Memory Write cycle by way of the Miscellaneous Options register Memory Write and Invalidate Control bit (MSCOPT[12]; PCI:46h). The PCI 6150 Disconnects Memory Write and Invalidate commands at Aligned Cache Line boundaries. The Cache Line Size register (PCICLSR; PCI:0Ch) cache line size value provides the number of Dwords in a cache line. For the PCI 6150 to generate Memory Write and Invalidate transactions, this cache line size value must be written to a value of 08h, 10h, or 20h Dwords. If an invalid cache line size is programmed, wherein the value is 0, not a power of two, or greater than 20h Dwords, the PCI 6150 sets the cache line size to the minimum value of 08h. The PCI 6150 always Disconnects on the Cache Line boundary.

When the Memory Write and Invalidate transaction is Disconnected before a Cache Line boundary is reached (typically because the Posted Write Data buffer fills), the transaction is converted to a Memory Write transaction.

8.5.3 Delayed Write Transactions

A Delayed Write transaction forwards I/O Write and Type 1 Configuration cycles by way of the PCI 6150, and is limited to a single Dword Data transfer.

When a Write transaction is first detected on the initiator bus, the PCI 6150 claims the access and returns a Target Retry to the initiator. During the cycle, the PCI 6150 samples the Bus Command, Address, and Address Parity bits. The PCI 6150 also samples the first data Dword, Byte Enable bits, and data parity. Cycle information is placed into the Delayed Transaction queue if there are no other existing Delayed transactions with the same cycle information, and if the Delayed Transaction queue is not full. When the PCI 6150 schedules a Delayed Write transaction to be the next cycle to complete based on its ordering constraints, the PCI 6150 initiates the transaction on the target bus. The PCI 6150 transfers the Write data to the target.

If the PCI 6150 receives a Target Retry in response to the Write transaction on the target bus, the PCI 6150 continues to repeat the Write transaction until the Data transfer is complete, or an error condition is encountered. If the PCI 6150 is unable to deliver Write data after 2^{24} attempts (programmable through the Timeout Control register Maximum Retry Counter Control bits, TOCNTRL[2:0]; PCI:45h), the PCI 6150 ceases further write attempts and returns a Target Abort to the initiator. The Delayed transaction is removed from the Delayed Transaction queue.

The PCI 6150 also asserts P_SERR# if the Command register P_SERR# Enable bit is set (PCICR[8]=1; PCI:04h). When the initiator repeats the same Write transaction (same command, address, Byte Enable bits, and data), after the PCI 6150 has completed data delivery and retains all complete cycle information in the queue, the PCI 6150 claims the access and returns TRDY# to the initiator, indicating that the Write data was transferred. If the initiator requests multiple Dwords, the PCI 6150 asserts STOP#, in conjunction with TRDY#, to signal a Target Disconnect. Only those bytes of Write data with valid Byte Enable bits are compared. If any Byte Enable bits are disabled (driven high), the corresponding byte of Write data is ***not*** compared.

If the initiator repeats the Write transaction before the data is transferred to the target, the PCI 6150 returns a Target Retry to the initiator. The PCI 6150 continues to return a Target Retry to the initiator until Write data is delivered to the target or an error condition is encountered. When the Write transaction is repeated, the PCI 6150 does not make a new entry into the Delayed Transaction queue.

The PCI 6150 implements a Discard Timer that starts counting when the Delayed Write completion is at the head of the Delayed Transaction queue. The initial value of this timer can be set to one of four values, selectable through the primary and secondary Bridge Control register Master Timeout bits (BCNTRL[8:9]; PCI:3Eh, respectively), as well as the Timeout Control register Master Timeout Divider bits (TOCNTRL[7:4]; PCI:45h). If the Discard Timer expires before the Write cycle is Retried, the PCI 6150 discards the Delayed Write transaction from the Delayed Transaction queue. The PCI 6150 also conditionally asserts P_SERR#.

8.5.4 Write Transaction Address Boundaries

The PCI 6150 imposes internal Address boundaries when accepting Write data. The Aligned Address boundaries are used to prevent the PCI 6150 from continuing a transaction over a device Address boundary and to provide an upper limit on maximum latency. When the Aligned Address boundaries are reached (per conditions listed in Table 8-3), the PCI 6150 returns a Target Disconnect to the initiator.

8.5.5 Buffering Multiple Write Transactions

The PCI 6150 continues to accept Posted Memory Write transactions if space for at least 1 Dword of data in the Posted Write Data buffer remains and there are fewer than four outstanding Posted Memory Write cycles. If the Posted Write Data buffer fills before the initiator terminates the Write transaction, the PCI 6150 returns a Target Disconnect to the initiator.

Delayed Write transactions are posted when one or more open entries exist in the Delayed Transaction queue. The PCI 6150 can queue up to four Posted Write transactions and four Delayed transactions in both the downstream and upstream directions.

8.5.6 Read Transactions

Delayed Read forwarding is used for all Read transactions that cross the PCI 6150.

Delayed Read transactions are treated as prefetchable or non-prefetchable.

Table 8-4 delineates the read behavior (prefetchable or non-prefetchable) for each type of Read operation.

Table 8-3. Write Transaction Disconnect Address Boundaries

Transaction Type	Condition	Aligned Address Boundary
Delayed Write	All	Disconnects after one Data transfer
Posted Memory Write	Memory Write Disconnect Control Bit = 0[1]	4-KB Aligned Address boundary
	Memory Write Disconnect Control Bit = 1[1]	Disconnects at Cache Line boundary
Posted Memory Write and Invalidate	Cache Line Size = 8h	8h-Dword aligned Address boundary
	Cache Line Size = 10h	10h-Dword aligned Address boundary
	Cache Line Size = 12h	12h-Dword aligned Address boundary

1. Memory Write Disconnect Control bit is located in the Chip Control register in Configuration space (CCNTRL[1]; PCI:40h).

Table 8-4. Read Transaction Prefetching

Transaction Type	Read Behavior
I/O Read	Never prefetches
Configuration Read	
Memory Read	Downstream—Prefetches if address is in prefetchable space Upstream—Prefetches if prefetch disable is off (default)
Memory Read Line	Always prefetches if request is for more than one Data transfer
Memory Read Multiple	

8.5.7 Prefetchable Read Transactions

A Prefetchable Read transaction is a Read transaction wherein the PCI 6150 performs speculative DWORD reads, transferring data from the target before the initiator requests the data. This behavior allows a Prefetchable Read transaction to consist of multiple Data transfers. Only the first Byte Enable bits can be forwarded. The PCI 6150 enables all Byte Enable bits of subsequent transfers.

Prefetchable behavior is used for Memory Read Line and Memory Read Multiple transactions, as well as Memory Read transactions that fall into Prefetchable Memory space.

The amount of prefetched data depends on the transaction type. The amount of prefetching may also be affected by the amount of free space in the PCI 6150 Read FIFO and by the Read Address boundaries encountered. In addition, there are several PCI 6150-specific registers that can be used to optimize read prefetch behavior.

Prefetching should not be used for those Read transactions that cause side effects on the target device (*that is*, Control and Status registers, FIFOs, and so forth). The target device BARs indicate whether a Memory Address region is prefetchable.

8.5.8 Non-Prefetchable Read Transactions

A Non-Prefetchable Read transaction is a Read transaction issued by the initiator into a non-prefetchable region. The transaction is used for I/O and Configuration Read transactions, as well as for Memory Reads from Non-Prefetchable Memory space. In this case, the PCI 6150 requests only 1 Dword from the target and Disconnects the initiator after delivery of the first Dword of Read data.

Use Non-Prefetchable Read transactions for regions in which extra Read transactions could have side effects (*such as* in FIFO memory or the Control registers). If it is important to retain the Byte Enable bit values during the Data phase of cycles forwarded across the bridge, use Non-Prefetchable Read transactions. If these locations are mapped into Memory space, use the Memory Read command and map the target into Non-Prefetchable (Memory-Mapped I/O) Memory space to utilize non-prefetching behavior.

8.5.9 Read Prefetch Address Boundaries

The PCI 6150 imposes internal Read Address boundaries on read prefetching. The PCI 6150 uses the Address boundary to calculate the initial amount of prefetched data. During Read transactions to Prefetchable regions, the PCI 6150 prefetches data until it reaches one of these aligned Address boundaries, unless the target signals a Target Disconnect before reaching the Read Prefetch boundary. After reaching the Aligned Address boundary, the PCI 6150 may optionally continue prefetching data, depending on certain conditions. (Refer to Section 17, “PCI Flow-Through Optimization.”) When finished transferring Read data to the initiator, the PCI 6150 returns a Target Disconnect with the last Data transfer, unless the initiator completes the transaction before delivering all the prefetched Read data. Remaining prefetched data is discarded.

Prefetchable Read transactions in Flow-Through mode prefetch to the nearest aligned 4-KB Address boundary, or until the initiator de-asserts FRAME#.

Table 8-5 delineates the Read Prefetch Address boundaries for Read transactions during Non-Flow-Through mode.

Table 8-5. Read Prefetch Address Boundaries

Transaction Type	Address Space	Prefetch Aligned Address Boundary
Configuration Read	—	1 Dword (No Prefetch)
I/O Read	—	1 Dword (No Prefetch)
Memory Read	Non-Prefetchable	1 Dword (No Prefetch)
Memory Read	Prefetchable	Configured by way of Prefetch Count registers
Memory Read Line	Prefetchable	Configured by way of Prefetch Count registers
Memory Read Multiple	Prefetchable	Configured by way of Prefetch Count registers

8.5.10 Delayed Read Requests

The PCI 6150 treats all Read transactions as Delayed Read transactions (*that is*, the Read request from the initiator is posted into a Delayed Transaction queue). Read data from the target is placed into the Read Data queue directed toward the initiator bus interface and transferred to the initiator when the initiator repeats the Read transaction.

When the PCI 6150 accepts a Delayed Read request, it first samples the Read address, Read bus command, and address parity. When IRDY# is asserted, the PCI 6150 samples the Byte Enable bits for the first Data phase. This information is entered into the Delayed Transaction queue. The PCI 6150 terminates the transaction by signaling a Target Retry to the initiator. Upon receiving the Target Retry, the initiator must to continue to repeat the same Read transaction until at least one Data transfer completes, or until it receives a target response other than a Target Retry (Master or Target Abort).

8.5.11 Delayed Read Completion with Target

When a Delayed Read request is scheduled to be executed, the PCI 6150 arbitrates for the target bus and initiates the Read transaction, using the exact Read address and Read command captured from the initiator during the initial Delayed Read request. If the Read transaction is non-prefetchable, the PCI 6150 drives the captured Byte Enable bits during the next cycle. If the transaction is a Prefetchable Read transaction, the PCI 6150 drives the captured (first) Byte Enable bits, followed by 0 for the subsequent Data phases. If the PCI 6150 receives a Target Retry in response to the Read transaction on the target bus, it repeats the Read transaction until at least one Data transfer completes or it encounters an error condition. If the transaction is terminated by way of a normal Master Termination or Target Disconnect after at least one Data transfer is complete, the PCI 6150 does not initiate further attempts to read additional data.

If the PCI 6150 is unable to obtain Read data from the target after 2^{24} attempts (default), the PCI 6150 ceases further read attempts and returns a Target Abort to the initiator. The Delayed transaction is removed from the Delayed Transaction queue. The PCI 6150 also asserts P_SERR# if the Command register P_SERR# Enable bit is set (PCICR[8]=1; PCI:04h).

After receiving DEVSEL# and TRDY# from the target, the PCI 6150 transfers the data stored in the internal Read FIFO, then terminates the transaction. The PCI 6150 can accept 1 Dword/Qword of Read data during each PCI Clock cycle—no master wait states are inserted. The number of Dwords/Qwords transferred during a Delayed Read transaction depends on the conditions delineated in Table 8-5 (assuming no Target Disconnect is received).

8.5.12 Delayed Read Completion on Initiator Bus

When the Delayed Read transaction completes on the target bus, the Delayed Read data is at the head of the Read Data queue. When all ordering constraints with Posted Write transactions are satisfied, the PCI 6150 transfers the data to the initiator when the initiator repeats the transaction. For Memory Read transactions, the PCI 6150 aliases the Memory Read, Memory Read Line, and Memory Read Multiple bus commands when matching the bus command of the transaction to the bus command in the Delayed Transaction queue. The PCI 6150 returns a Target Disconnect along with the transfer of the last Dword of Read data to the initiator. If the PCI 6150 initiator terminates the transaction before all Read data is transferred, the remaining Read data in the Data buffers is discarded.

When the master repeats the transaction and starts transferring prefetchable Read data from the Data buffers while the Read transaction on the target bus is in progress, and before a Read boundary is reached on the target bus, the Read transaction starts operating in Flow-Through mode. Because data is flowing from the target to the initiator through the Data buffers, long Read bursts can be sustained. In this case, the Read transaction is allowed to continue until the initiator terminates the transaction, an aligned 4-KB Address boundary is reached, or the buffer fills, whichever occurs first. When the buffer empties, the PCI 6150 reflects the stalled condition to the initiator by de-asserting TRDY# for a maximum of eight clock periods until more Read data is available; otherwise, the PCI 6150 Disconnects the cycle. When the initiator terminates the transaction, the PCI 6150 de-assertion of FRAME# on the initiator bus is forwarded to the target bus. Any remaining Read data is discarded.

The PCI 6150 implements a Discard Timer that starts counting when the Delayed Write completion is at the head of the Delayed Transaction queue. The initial value of this timer can be set to one of four values, selectable through the primary and secondary Bridge Control register Master Timeout bits (BCNTRL[8:9]; PCI:3Eh, respectively), as well as the Timeout Control register Master Timeout Divider bits (TOCNTRL[7:4]; PCI:45h). If the Discard Timer expires before the Write cycle is Retried, the PCI 6150 discards the Delayed Write transaction from the Delayed Transaction queue. The PCI 6150 also conditionally asserts P_SERR#.

The PCI 6150 has the capability to post multiple Delayed Read requests, up to a maximum of four in both directions. If an initiator starts a Read transaction that matches the Address and Read command of a queued Read transaction, the current Read command is not stored because it is contained in the Delayed Transaction queue.

8.5.13 Configuration Transactions

Configuration transactions are used to initialize a PCI system. Every PCI device has a Configuration space that is accessed by Configuration commands. All registers are accessible only in Configuration space.

In addition to accepting Configuration transactions for initialization of its own Configuration space, the PCI 6150 forwards Configuration transactions for device initialization in hierarchical PCI Bus systems, as well as Special Cycle generation.

To support hierarchical PCI Bus systems, Type 0 and Type 1 Configuration transactions are specified.

Type 0 Configuration transactions are issued when the intended target resides on the same PCI Bus as the initiator. Type 0 Configuration transactions are identified by the Configuration command and the lowest two bits of the address are set to 00b.

Type 1 Configuration transactions are issued when the intended target resides on another PCI Bus, or a Special Cycle is to be generated on another PCI Bus. Type 1 Configuration commands are identified by the Configuration command and the lowest two Address bits are set to 01b.

The Register Number is found in both Type 0 and Type 1 formats and provides the Dword address of the Configuration register to be accessed. The Function Number is also included in both Type 0 and Type 1 formats, and indicates which function of a multi-function device is to be accessed. For single-function devices, this value is not decoded. Type 1 Configuration transaction addresses also include five bits, designating the Device Number that identifies the target PCI Bus device to be accessed. In addition, the Bus Number in Type 1 transactions specifies the target PCI Bus.

8.5.14 PCI 6150 Type 0 Access

Configuration space is accessed by a Type 0 Configuration transaction on the primary interface. Configuration space is ***not*** accessible from the secondary bus. The PCI 6150 responds to a Type 0 Configuration transaction by asserting P_DEVSEL# when the following conditions are met during the Address phase:

- Bus command is a Configuration Read or Write transaction.
- Lower two Address bits on P_AD[1:0] must be 01b.
- P_IDSEL must be asserted.
- PCI 6150 limits all Configuration accesses to a single DWORD Data transfer and returns a Target Disconnect with the first Data transfer if additional Data phases are requested. Because Read transactions to Configuration space do not have side effects, all bytes in the requested Dword are returned, regardless of the Byte Enable bit values.
- Type 0 Configuration Read and Write transactions do not use data buffers (*that is*, these transactions are immediately completed, regardless of the Data buffers state).

The PCI 6150 ignores all Type 0 transactions initiated on the secondary interface.

8.5.15 Type 1-to-Type 0 Translation

Type 1 Configuration transactions are specifically used for device configuration in a hierarchical PCI Bus system. A PCI-to-PCI bridge is the only type of device that should respond to a Type 1 Configuration command. Type 1 Configuration commands are used when the Configuration access is intended for a PCI device that resides on a PCI Bus other than the one where the Type 1 transaction is generated.

The PCI 6150 performs a Type 1-to-Type 0 translation when the Type 1 transaction is generated on the primary bus and is intended for a device attached directly to the secondary bus. The PCI 6150 must convert the Configuration command to a Type 0 format, enabling the secondary bus device to respond to the command. Type 1-to-Type 0 translations are performed only in the downstream direction (*that is*, the PCI 6150 generates a Type 0 transaction only on the secondary bus, and never on the primary bus).

The PCI 6150 responds to a Type 1 Configuration transaction and translates the transaction into a Type 0 transaction on the secondary bus when the following conditions are met during the Address phase:

- Lower two Address bits on P_AD[1:0] are 01b
- Bus Number in address field P_AD[23:16] is equal to the Secondary Bus Number register value in Configuration space (PCISBNO; PCI:19h)
- Bus command on P_CBE[3:0]# is a Configuration Read or Write transaction

When translating a Type 1 transaction to a Type 0 transaction on the secondary interface, the PCI 6150 performs the following translations to the address:

- Sets the lower two Address bits on S_AD[1:0] to 00b
- Decodes the Device Number and drives the bit pattern specified in Table 8-6 on S_AD[31:16] for the purpose of asserting the device’s IDSEL signal
- Sets S_AD[15:11] to 0h
- Leaves the Function and Register Number fields unchanged

The PCI 6150 asserts unique address lines, based on the Device Number. These address lines may be used as secondary IDSEL signals. Address line mapping depends on the Device Number in the Type 1 Address bits, P_AD[15:11]. The PCI 6150 uses the mapping presented in Table 8-6.

The PCI 6150 can assert up to 16 unique address lines to be used as secondary IDSEL signals for up to 16 secondary bus devices, for Device Numbers ranging from 0 to 15. Because of the PCI Bus electrical loading constraints, more than 16 IDSEL signals should not be necessary. However, if more than 15 device numbers are needed, an external method of generating IDSEL lines must be used, and the upper Address bits are ***not*** asserted. The Configuration transaction is translated and passed from primary-to-secondary bus. If an IDSEL pin is not asserted to a secondary device, the transaction terminates in a Master Abort.

The PCI 6150 forwards Type 1-to-Type 0 Configuration Read or Write transactions as Delayed transactions. Type 1-to-Type 0 Configuration Read or Write transactions are limited to a single 32-bit Data transfer. When Type 1-to-Type 0 Configuration cycles are forwarded, Address Stepping is used, and a valid address is driven on the bus before FRAME# assertion. Type 0 Configuration Address Stepping is programmable through the Miscellaneous Options register Address Step Control bits (MSCOPT[6:4]; PCI:46h).

Table 8-6. Device Number to IDSEL S_AD Pin Mapping

Device Number	P_AD[15:11]	Secondary IDSEL S_AD[31:16]	S_AD Bit
0h	00000b	0000_0000_0000_0001b	16
1h	00001b	0000_0000_0000_0010b	17
2h	00010b	0000_0000_0000_0100b	18
3h	00011b	0000_0000_0000_1000b	19
4h	00100b	0000_0000_0001_0000b	20
5h	00101b	0000_0000_0010_0000b	21
6h	00110b	0000_0000_0100_0000b	22
7h	00111b	0000_0000_1000_0000b	23
8h	01000b	0000_0001_0000_0000b	24
9h	01001b	0000_0010_0000_0000b	25
10h	01010b	0000_0100_0000_0000b	26
11h	01011b	0000_1000_0000_0000b	27
12h	01100b	0001_0000_0000_0000b	28
13h	01101b	0010_0000_0000_0000b	29
14h	01110b	0100_0000_0000_0000b	30
15h	01111b	1000_0000_0000_0000b	31
Special Cycle	1XXXXb	0000_0000_0000_0000b	—

8.5.16 Type 1-to-Type 1 Forwarding

Type 1-to-Type 1 transaction forwarding provides a hierarchical configuration mechanism when two or more levels of PCI-to-PCI bridges are used.

When the PCI 6150 detects a Type 1 Configuration transaction intended for a PCI Bus downstream from the secondary bus, the PCI 6150 forwards the transaction unchanged to the secondary bus. Ultimately, this transaction is translated to a Type 0 Configuration command or to a Special Cycle transaction by a downstream PCI-to-PCI bridge. Downstream Type 1-to-Type 1 forwarding occurs when the following conditions are met during the Address phase:

- Lower two Address bits on AD[1:0] are equal to 01b
- Bus Number falls in the range defined by the lower limit (exclusive) in the Secondary Bus Number register (PCISBNO; PCI:19h) and upper limit (inclusive) in the Subordinate Bus Number register (PCISUBNO; PCI:1Ah)
- Bus command is a Configuration Read or Write transaction

The PCI 6150 also supports Type 1-to-Type 1 upstream Configuration Write transaction forwarding to support upstream Special Cycle generation. A Type 1 Configuration command is forwarded upstream when the following conditions are met:

- Lower two Address bits on AD[1:0] are equal to 01b
- Bus Number falls outside the range defined by the lower limit (inclusive) in the Secondary Bus Number register (PCISBNO; PCI:19h) and upper limit (inclusive) in the Subordinate Bus Number register (PCISUBNO; PCI:1Ah)
- Device Number in Address bits AD[15:11] is equal to 11111b
- Function Number in Address bits AD[10:8] is equal to 111b
- Bus command is a Configuration Write transaction
- PCI 6150 forwards Type 1-to-Type 1 Configuration Write transactions as Delayed transactions, limited to a single Data transfer

8.5.17 Special Cycles

The Type 1 configuration mechanism is used to generate Special Cycle transactions in hierarchical PCI systems. Special Cycle transactions are ignored by operating as a target and are not forwarded across the bridge. Special Cycle transactions can be generated from Type 1 Configuration Write transactions in the downstream or upstream direction.

The PCI 6150 initiates a Special Cycle on the target bus when a Type 1 Configuration Write transaction is detected on the initiating bus and the following conditions are met during the Address phase:

- Lower two Address bits on AD[1:0] are equal to 01b
- Device Number in Address bits AD[15:11] is equal to 11111b
- Function Number in Address bits AD[10:8] is equal to 111b
- Register number in Address bits AD[7:2] is equal to 0h
- Bus Number is equal to the Secondary Bus Number register value in Configuration space (PCISBNO; PCI:19h) for downstream forwarding, or equal to the Primary Bus Number register value in Configuration space (PCIPBNO; PCI:18h) for upstream forwarding
- Bus command on the initiator CBE bus is a Configuration Write command

When the PCI 6150 initiates a transaction on the target interface, the bus command is changed from Configuration Write to Special Cycle. The address and data are forwarded, unchanged. Devices that use Special Cycle ignore the address and decode only the bus command. The Data phase contains the Special Cycle message. The transaction is forwarded as a Delayed transaction because Special Cycles complete as Master Aborts. After the transaction is completed on the target bus, through Master Abort condition detection, the PCI 6150 responds with TRDY# to the next attempt of the Configuration transaction from the initiator. If more than one Data transfer is requested, the PCI 6150 responds with a Target Disconnect operation during the first Data phase.

8.6 TRANSACTION TERMINATION

This subsection describes how the PCI 6150 returns transaction termination conditions to the initiator.

The initiator can terminate transactions with one of the following types of termination:

- **Normal Termination**—Occurs when the initiator de-asserts FRAME# at the beginning of the last Data phase, and de-asserts IRDY# at the end of the last Data phase in conjunction with TRDY# or STOP# assertion from the target.
- **Master Abort**—Occurs when no target response is detected. When the initiator does not detect the DEVSEL# signal from the target within five Clock cycles after asserting FRAME#, the initiator terminates the transaction with a Master Abort. If FRAME# is asserted, the initiator de-asserts FRAME# on the next cycle, then de-asserts IRDY# on the following cycle. IRDY# must be asserted in the same cycle in which FRAME# is de-asserted. If FRAME# was de-asserted, IRDY# can be de-asserted on the next Clock cycle following Master Abort condition detection.

The target can terminate transactions with one of the following types of termination:

- **Normal Termination**—TRDY# and DEVSEL# are asserted in conjunction with FRAME# de-assertion and IRDY# assertion.
- **Target Retry**—STOP# and DEVSEL# are asserted without TRDY# during the first Data phase. No data transfers during the transaction. This transaction must be repeated.
- **Target Disconnect (with Data transfer)**—DEVSEL# and STOP# are asserted with TRDY#. Indicates that this is the last Data transfer of the transaction.
- **Target Disconnect (without Data transfer)**—STOP# and DEVSEL# are asserted without TRDY# after previous Data transfers. Indicates that no further Data transfers are made during this transaction.
- **Target Abort**—STOP# is asserted without DEVSEL# and TRDY#. Indicates that the target is never able to complete this transaction. DEVSEL# must be asserted for at least one cycle during the transaction before the Target Abort is signaled.

8.6.1 PCI 6150-Initiated Master Termination

As an initiator, the PCI 6150 uses normal termination if DEVSEL# is returned by the target within five Clock cycles of PCI 6150 FRAME# assertion on the target bus. In this case, the PCI 6150 terminates a transaction when the following conditions are met:

- During Delayed Write transactions, a single Dword/Qword is delivered.
- During Non-Prefetchable Read transactions, a single Dword/Qword is transferred from the target.
- During Prefetchable Read transactions, a Prefetch boundary is reached.
- For Posted Write transactions, all Write data for the transaction is transferred from Data buffers to the target.
- For Burst transfers (*except* Memory Write and Invalidate transactions), the Master Latency Timer expires and the PCI 6150 bus grant is de-asserted.
- Target terminates the transaction with a Retry, Disconnect, or Target Abort.
- If the PCI 6150 is delivering Posted Write data when it terminates the transaction because the Master Latency Timer expired, the PCI 6150 initiates another transaction to deliver the remaining Write data. The transaction address is updated to reflect the address of the current Dword to be delivered.

If the PCI 6150 is delivering Posted Write data when it terminates the transaction because the Master Latency Timer expires, the PCI 6150 initiates another transaction to deliver the remaining Write data. The Transaction address is updated to reflect the current DWORD address to be delivered.

If the PCI 6150 is prefetching Read data when it terminates the transaction because the Master Latency Timer expired, the PCI 6150 does ***not*** repeat the transaction to obtain additional data.

8.6.2 Master Abort Received by PCI 6150

If the initiator initiates a transaction on the target bus and does not detect DEVSEL# returned by the target within five Clock cycles of FRAME# assertion, the PCI 6150 terminates the transaction, as specified in the Bridge Control register Master Abort Mode bit (BCNTRL[5]; PCI:3Eh).

For Delayed Read and Write transactions, the PCI 6150 can assert TRDY# and return FFFF_FFFFh for reads, or return a Target Abort. SERR# is also optionally asserted.

When a Master Abort is received in response to a Posted Write transaction, the PCI 6150 discards the Posted Write data and makes no further attempts to deliver the data. The PCI 6150 sets the Status register Received Master Abort bit when the Master Abort is received on the primary bus (PCISR[13]=1; PCI:06h), or the Secondary Status register Received Master Abort bit when the Master Abort is received on the secondary interface (PCISSR[13]=1; PCI:1Eh).

When the Master Abort Mode bit is set and a Master Abort is detected in response to a Posted Write transaction, the PCI 6150 also asserts P_SERR#, if enabled (PCICR[8]=1; PCI:04h), but not disabled by the device-specific P_SERR# disable for Master Aborts that occur during Posted Write transactions. (Refer to Table 8-7.)

8.6.3 Target Termination Received by PCI 6150

When the PCI 6150 initiates a transaction on the target bus and the target responds with DEVSEL#, the target can end the transaction with one of the following types of termination:

- Normal termination (upon FRAME# de-assertion)
- Target Retry
- Target Disconnect
- Target Abort

The PCI 6150 controls these terminations using various methods, depending on the type of transaction performed.

Table 8-7. P_SERR# Assertion Requirements in Response to Master Abort on Posted Write

Description	Bit
Received Master Abort	PCISR[13]=1; PCI:06h
P_SERR# Enable	PCICR[8]=1; PCI:04h
Master Abort on Posted Write	PSERRED[4]=0; PCI:64h

8.6.3.1 Posted Write Target Termination Response

When the PCI 6150 initiates a Posted Write transaction, the Target Termination ***cannot*** be returned to the initiator. Table 8-8 delineates the response to each type of Target Termination that occurs during a Posted Write transaction.

When a Target Retry or Disconnect is returned and Posted Write data associated with that transaction remains in the Write buffers, the PCI 6150 initiates another Write transaction to attempt to deliver the remaining Write data. In the case of a Target Retry, the same address is driven as for the initial Write transaction attempt. If a Target Disconnect is received, the address that is driven on a subsequent Write transaction attempt is updated to reflect the current Dword address. If the initial Write transaction is a Memory Write and Invalidate transaction, and a partial delivery of Write data to the target is performed before a Target Disconnect is received, the PCI 6150 uses the Memory Write command to deliver the remaining Write data because less than a cache line is transferred in the subsequent Write transaction attempt.

After the PCI 6150 makes 2^{24} write attempts and fails to deliver all Posted Write data associated with that transaction, the PCI 6150 asserts P_SERR#, if enabled in the Command register, ***and*** the device-specific P_SERR# Disable bit for this condition is ***not*** set. (Refer to Table 8-9.) The Write data is discarded.

Table 8-8. Response to Posted Write Target Termination

Target Termination	Response
Normal	No additional action.
Target Retry	Repeats Write transaction to target.
Target Disconnect	Initiates Write transaction to deliver remaining Posted Write data.
Target Abort	Sets target interface Status register Received Target Abort bit (primary—PCISR[12]=1, PCI:06h, secondary—PCISSR[12]=1; PCI:1Eh). Asserts P_SERR#, if enabled, and sets the Primary Status register Signaled System Error bit (PCICR[8]=1; PCI:04h and PCISR[14]=1; PCI:06h, respectively).

Table 8-9. P_SERR# Assertion Requirements in Response to Posted Write Parity Error

Description	Bit
P_SERR# Enable	PCICR[8]=0; PCI:04h
Posted Write Parity Error	PSERRED[1]=0; PCI:64h

8.6.3.2 Delayed Write Target Termination Response

When the PCI 6150 initiates a Delayed Write transaction, the type of Target Termination received from the target can be returned to the initiator. Table 8-10 delineates the response to each type of Target Termination that occurs during a Delayed Write transaction. The PCI 6150 repeats a Delayed Write transaction until the PCI 6150:

- Completes at least one Data transfer
- Receives a Master Abort
- Receives a Target Abort

The PCI 6150 makes 2^{24} write attempts (default), resulting in a response of Target Retry. After the PCI 6150 makes 2^{24} attempts of the same Delayed Write transaction on the target bus, the PCI 6150 asserts P_SERR# if the Command register P_SERR# Enable bit is set and the implementation-specific P_SERR# Disable bit for this condition is not set. (Refer to Table 8-11.) The PCI 6150 stops initiating transactions in response to that Delayed Write transaction and the Delayed Write request is discarded. Upon a subsequent Write transaction attempt by the initiator, the PCI 6150 returns a Target Abort.

Table 8-10. Response to Delayed Write Target Termination

Target Termination	Response			
Normal	Returns Disconnect to initiator with first Data transfer only if multiple Data phases are requested.			
Target Retry	Returns Target Retry to initiator. Continue write attempts to target.			
Target Disconnect	Returns Disconnect to initiator with first Data transfer only if multiple Data phases are requested.			
Target Abort	Returns Target Abort to initiator. Sets target interface Status register Received Target Abort bit. Sets initiator interface Status register Signaled Target Abort bit.			
	Initiator (Primary Bus)	**Target (Secondary Bus)**	**Initiator (Secondary Bus)**	**Target (Primary Bus)**
	PCISR[11]=1; PCI:06h	PCISSR[12]=1; PCI:1Eh	PCISR[12]=1; PCI:06h	PCISSR[11]=1; PCI:1Eh

Table 8-11. P_SERR# Assertion Requirements in Response to Delayed Write

Description	Bit
P_SERR# Enable	PCICR[8]=1; PCI:04h
Delayed Configuration or I/O Write Non-Delivery	PSERRED[5]=0; PCI:64h

8.6.3.3 Delayed Read Target Termination Response

When the PCI 6150 initiates a Delayed Read transaction, the abnormal target responses can be returned to the initiator. Other target responses depend on the amount of data the initiator requests. Table 8-12 delineates the response to each type of Target Termination that occurs during a Delayed Read transaction.

The PCI 6150 repeats a Delayed Read transaction until the PCI 6150:

- Completes at least one Data transfer
- Receives a Master Abort
- Receives a Target Abort
- Produces 2^{24} read attempts, resulting in a response of Target Retry

After the PCI 6150 produces 2^{24} attempts of the same Delayed Read transaction on the target bus, the PCI 6150 asserts P_SERR# if the Command register P_SERR# Enable bit is set and the implementation-specific P_SERR# Disable bit for this condition is ***not*** set. (Refer to Table 8-13.) The PCI 6150 stops initiating transactions in response to that Delayed Read transaction, and the Delayed Read request is discarded. Upon a subsequent Read transaction attempt by the initiator, the PCI 6150 returns a Target Abort.

Table 8-12. Response to Delayed Read Target Termination

Target Termination	Response			
Normal	If prefetchable, Target Disconnects only if initiator requests more data than read from target. If non-prefetchable, Target Disconnects on first Data phase.			
Target Retry	Re-initiates Read transaction to target.			
Target Disconnect	If initiator requests more data than read from target, returns Target Disconnect to initiator.			
Target Abort	Returns Target Abort to initiator. Sets target interface Status register Received Target Abort bit. Sets initiator interface Status register Signaled Target Abort bit.			
	Initiator (Primary Bus)	**Target (Secondary Bus)**	**Initiator (Secondary Bus)**	**Target (Primary Bus)**
	PCISR[11]=1; PCI:06h	PCISSR[12]=1; PCI:1Eh	PCISR[12]=1; PCI:06h	PCISSR[11]=1; PCI:1Eh

Table 8-13. P_SERR# Assertion Requirements in Response to Delayed Read

Description	Bit
P_SERR# Enable	PCICR[8]=1; PCI:04h
Delayed Read-No Data from Target	PSERRED[6]=0; PCI:64h

8.6.4 PCI 6150-Initiated Target Termination

The PCI 6150 can return a Target Retry, Disconnect, or Abort to an initiator for reasons other than detection of that condition at the target interface.

8.6.4.1 Target Retry

When it cannot accept Write data or return Read data as a result of internal conditions, the PCI 6150 returns a Target Retry to the initiator when any of the following conditions are met:

- **Delayed Write Transactions**
  - Transaction is in the process of entering the Delayed Transaction queue.
  - Transaction has entered the Delayed Transaction queue, but target response has not been received.
  - Target response was received, but the Posted Memory Write Ordering rule prevents the cycle from completing.
  - Delayed Transaction queue is full; therefore, transaction ***cannot*** be queued.
  - Transaction with the same address and command was queued.
  - Locked sequence is being propagated across the PCI 6150, and the Write transaction is not a Locked transaction.
  - Target bus is locked and the Write transaction is a Locked transaction.
- **Delayed Read Transactions**
  - Transaction is in the process of entering the Delayed Transaction queue.
  - Read request was queued, but Read data is not yet available.
  - Data was read from the target, but the data is not at the head of the Read Data queue, or a Posted Write transaction precedes it.
  - Delayed Transaction queue is full, and the transaction ***cannot*** be queued.
  - Delayed Read request with the same address and bus command was queued.
  - Locked sequence is being propagated across the PCI 6150, and the Read transaction is not a Locked transaction.
  - Target bus is locked and the Read transaction is a Locked transaction.
- **Posted Write Transactions**
  - Posted Write Data buffer does not contain sufficient space for the address and at least two Qwords of Write data.
  - Locked sequence is being propagated across the PCI 6150, and the Write transaction is not a Locked transaction.

When a Target Retry is returned to a Delayed transaction initiator, the initiator must repeat the transaction with the same address and bus command, as well as the data if this is a Write transaction, within the time frame specified by the Master Timeout value; otherwise, the transaction is discarded from the buffers.

8.6.4.2 Target Disconnect

The PCI 6150 returns a Target Disconnect to an initiator when the PCI 6150:

- Reaches an internal Address boundary
- Reaches a 4-KB boundary for a Posted Memory Write cycle
- Cannot accept further Write data
- Contains no further Read data to deliver

8.6.4.3 Target Abort

The PCI 6150 returns a Target Abort to an initiator when the PCI 6150:

- Returns a Target Abort from the intended target
- Detects a Master Abort on the target, and the Master Abort Mode bit is set (BCNTRL[5]=1; PCI:3Eh)
- Cannot obtain Delayed Read data from the target nor deliver Delayed Write data to the target after 2^{24} attempts

When returning a Target Abort to the initiator, the PCI 6150 sets the Status register Signaled Target Abort bit corresponding to the initiator interface (PCISR[12 or 11]=1; PCI:06h).

9 ADDRESS DECODING

This section describes address decoding, including Address ranges, Memory address decoding, ISA mode, and VGA addressing support.

9.1 OVERVIEW

The PCI 6150 uses three Address ranges to control I/O and Memory Transaction forwarding across the bridge. These address ranges are defined by Base and Limit Address registers in Configuration space.

9.2 ADDRESS RANGES

The PCI 6150 uses the following Address ranges to determine which I/O and Memory transactions are forwarded from the primary-to-secondary PCI Bus, and from the secondary-to-primary PCI Bus:

- One 32-Bit I/O Address range
- One 32-Bit Memory-Mapped I/O (non-prefetchable memory) range
- One 64-Bit Prefetchable Memory Address range

Transaction addresses falling within these ranges are forwarded downstream from the primary-to-secondary PCI Bus. Transaction addresses falling outside these ranges are forwarded upstream from the secondary-to-primary PCI Bus.

The PCI 6150 uses flat Address space (*that is*, it does not perform address translation). The Address space has no gaps; therefore, addresses that are not marked for downstream forwarding are always forwarded upstream.

9.2.1 I/O Address Decoding

The PCI 6150 uses the following mechanisms, defined in Configuration space, to specify the I/O Address space for downstream and upstream forwarding:

- I/O Base and Limit Address registers (Base—PCIIOBAR; PCI:1Ch and PCIIOBARU16; PCI:30h, Limit—PCIIOLMT; PCI:1Dh and PCIIOLMTU16; PCI:32h)
- ISA Enable bit (BCNTRL[2]; PCI:3Eh)
- VGA Enable bit (BCNTRL[3]; PCI:3Eh)
- VGA Palette Snoop Enable bit (PCICR[5]; PCI:04h)

To enable downstream I/O transaction forwarding, the Command register I/O Space Enable bit must be set (PCICR[0]=1; PCI:04h). If the I/O Space Enable bit is not set, I/O transactions initiated on the primary bus are ignored. To enable upstream I/O transaction forwarding, the Command register Master Enable bit must be set (PCICR[2]=1; PCI:04h). If the Master Enable bit is not set, the PCI 6150 ignores I/O and Memory transactions initiated on the secondary bus. Setting the Master Enable bit also allows upstream Memory transaction forwarding.

Caution: If any configuration state affecting I/O transaction forwarding is changed by a Configuration Write operation on the primary bus when there are ongoing I/O transactions on the secondary bus, the PCI 6150 response to the secondary bus I/O transactions is unpredictable. Configure the I/O Base and Limit Address registers, and ISA Enable, VGA Enable, and VGA Palette Snoop Enable bits before setting the I/O Space Enable and Master Enable bits, and subsequently change these registers only when the primary and secondary PCI Buses are idle.

9.2.1.1 I/O Base and Limit Address Registers

The PCI 6150 implements one set of I/O Base and Limit Address registers in Configuration space that define an I/O Address range for downstream forwarding. The PCI 6150 supports 32-bit I/O addressing, which allows I/O addresses downstream from the PCI 6150 to be mapped anywhere in a 4-GB I/O Address space.

I/O transactions with addresses that fall inside the I/O Base and Limit register-defined range are forwarded downstream from the primary-to-secondary PCI Bus. I/O transactions with addresses that fall outside this range are forwarded upstream from the secondary-to-primary PCI Bus. The I/O range can be disabled by setting the I/O Base address to a value greater than that of the I/O Limit address. When the I/O range is disabled, all I/O transactions are forwarded upstream (no I/O transactions are forwarded downstream).

The I/O range has a minimum granularity of 4 KB and is aligned on a 4-KB boundary. The maximum I/O range is 4 GB.

The I/O Base register consists of an 8-bit field (PCIIOBAR; PCI:1Ch) and a 16-bit field (PCIIOBARU16; PCI:30h). The upper four bits of the 8-bit field define bits [15:12] of the I/O Base address. The lower four Read-Only bits are hardcoded to 0001b to indicate 32-bit I/O addressing support. Bits [11:0] of the Base address are assumed to be 0h, which naturally aligns the Base address to a 4-KB boundary with a minimum granularity of 4 KB. The 16 bits contained in the I/O Base Upper 16 Bits register (PCIIOBARU16; PCI:30h) define AD[31:16] of the I/O Base address. All 16 bits are read/write. After a primary bus or chip reset, the I/O Base address value is initialized to 0000_0001h.

The I/O Limit register consists of an 8-bit field (PCIIOLMT; PCI:1Dh) and a 16-bit field (PCIIOLMTU16; PCI:32h). The upper four bits of the 8-bit field define bits [15:12] of the I/O Limit address. The lower four Read-Only bits are hardcoded to 0001b to indicate 32-bit I/O addressing support. Bits [11:0] of the Limit address are assumed to be FFFh, which naturally aligns the Limit address to the top of a 4-KB I/O Address block. The 16 bits contained in the I/O Limit Upper 16 Bits register (PCIIOLMTU16; PCI:32h) define AD[31:16] of the I/O Limit address. All 16 bits are read/write. After a primary bus or chip reset, the I/O Limit address value is reset to 0000_ 0FFFh.

Note: *The initial states of the I/O Base and Limit registers (PCIIOBAR; PCI:1Ch and PCIIOLMT; PCI:1Dh, respectively) define an I/O range of 0000_0000h to 0000_0FFFh, which is the lower 4 KB of I/O space. Write these registers with their appropriate values before setting the Command register Master or I/O Space Enable bit (PCICR[2 or 0]=1; PCI:04h).*

9.3 MEMORY ADDRESS DECODING

The PCI 6150 has three mechanisms for defining Memory Address ranges for Memory transaction forwarding:

- Memory-Mapped I/O Base and Limit Address registers (PCIMBAR; PCI:20h and PCIMLMT; PCI:22h, respectively)
- Prefetchable Memory Base and Limit Address registers (Base—PCIPMBAR; PCI:24h and PCIPMBARU32; PCI:28h, Limit—PCIPMLMT; PCI:26h and PCIPMLMTU32; PCI:2Ch)
- VGA mode (BCNTRL[3]=1; PCI:3Eh)

This subsection describes the first two mechanisms. VGA mode is described in Section 9.5.1.

To enable downstream Memory transaction forwarding, the Command register Memory Space Enable bit must be set (PCICR[1]=1; PCI:04h). To enable upstream Memory transaction forwarding, the Command register Master Enable bit must be set (PCICR[2]=1; PCI:04h). Setting the Master Enable bit also enables upstream I/O transaction forwarding.

Caution: If any configuration state affecting Memory transaction forwarding is changed by a Configuration Write operation on the primary bus when there are ongoing memory transactions on the secondary bus, response to the secondary bus Memory transactions is unpredictable. Configure the Memory-Mapped I/O Base and Limit Address registers, Prefetchable Memory Base and Limit Address registers, and VGA Enable bit before setting the Memory Space Enable and Master Enable bits, and subsequently change these registers only when the primary and secondary PCI Buses are idle.

9.3.1 Memory-Mapped I/O Base and Limit Address Registers

Memory-mapped I/O is also referred to as Non-Prefetchable memory. Memory addresses that cannot be automatically prefetched, but can conditionally prefetch based on command type, should be mapped into this space. Read transactions to Non-Prefetchable space may exhibit side effects—may exhibit non-memory-like behavior. The PCI 6150 prefetches in this space only if the Memory Read line or Memory Read Multiple commands are used. Transactions using the Memory Read command are limited to a single data transfer.

The Memory-Mapped I/O Base and Limit Address registers define an Address range that the PCI 6150 uses to determine when to forward Memory commands. The PCI 6150 forwards a Memory transaction from the primary-to-secondary interface if the Transaction address falls within the Memory-Mapped I/O Address range. The PCI 6150 ignores Memory transactions initiated on the secondary interface that fall into this Address range. Transactions that fall outside this Address range are ignored on the primary interface and forwarded upstream from the secondary interface (provided that the transactions do not fall into the Prefetchable Memory range, or are not forwarded downstream by the VGA mechanism).

The Memory-Mapped I/O Address range supports only 32-bit addressing. *P-to-P Bridge r1.1* does not provide for 64-bit addressing in the Memory-Mapped I/O space. The Memory-Mapped I/O Address range has a granularity and alignment of 1 MB and a maximum range of 4 GB.

The Memory-Mapped I/O Address range is defined by a 16-bit Memory-Mapped I/O Base Address register (BAR) and a 16-bit Memory-Mapped I/O Limit Address register (PCIMBAR; PCI:20h and PCIMLMT; PCI:22h, respectively). The upper 12 bits of each of these registers correspond to bits [31:20] of the Memory address. The lower four bits are hardcoded to 0h. The lower 20 bits of the Memory-Mapped I/O Base address are assumed to be 0h, which results in a natural alignment to a 1-MB boundary. The lower 20 bits of the Memory-Mapped I/O Limit address are assumed to be F_FFFFh, which results in an alignment to the top of a 1-MB block.

Note: *The initial state of the Memory-Mapped I/O Base Address register (PCIMBAR; PCI:20h) is 0000_0000h. The initial state of the Memory-Mapped I/O Limit Address register (PCIMLMT; PCI:22h) is 000F_FFFFh. The initial states of these registers define a Memory-Mapped I/O range at the lower 1-MB Memory block. Write these registers with their appropriate values before setting the Command register Master or Memory Space Enable bit (PCICR[2 or 1]=1; PCI:04h).*

To disable the Memory-Mapped I/O Address range, write the Memory-Mapped I/O Base Address register with a value greater than that of the Memory-Mapped I/O Limit Address register.

9.3.1.1 Prefetchable Memory Base and Limit Address Registers

Locations accessed in the Prefetchable Memory Address range must have true memory-like behavior and not exhibit side effects when read (*that is*, extra reads to a prefetchable memory location must ***not*** have side effects). The PCI 6150 prefetches for all types of Memory Read commands in this Address space.

The PCI 6150 Prefetchable Memory Base and Limit Address registers define an Address range that the PCI 6150 uses to determine when to forward Memory transactions. The PCI 6150 forwards a Memory transaction from the primary-to-secondary interface, if the Transaction address falls within the Prefetchable Memory Address range. The PCI 6150 ignores Memory transactions initiated on the secondary interface that fall into this address range. The PCI 6150 does not respond to transactions that fall outside this address range on the primary interface and forwards those transactions upstream from the secondary interface (provided that the transactions do not fall into the Memory-Mapped I/O Address range, or are not forwarded by the VGA mechanism).

The PCI 6150 Prefetchable Memory range supports 64-bit addressing and provides additional registers to define the upper 32 bits of the Prefetchable Memory Base and Limit addresses. For address comparison, a Single Address Cycle (SAC; 32-bit address) Prefetchable Memory transaction is treated as a 64-bit Address transaction, where the upper 32 bits of the address are equal to 0h. This upper 32-bit value of 0h is compared to the Prefetchable Memory Base and Limit Address Upper 32 Bits registers. The Prefetchable Memory Base Address Upper 32 Bits register must be 0h to pass SAC transactions downstream.

The Prefetchable Memory Address range is defined by a 16-bit Prefetchable Memory Base Address register and a 16-bit Prefetchable Memory Limit Address register (PCIPMBAR; PCI:24h and PCIPMLMT; PCI:26h, respectively). The upper 12 bits of each of these registers correspond to bits [31:20] of the Memory address. The lower four Read-Only bits are hardcoded to 1h, indicating 64-bit address support. The lower 20 bits of the Prefetchable Memory Base address are assumed to be 0h, which results in a natural alignment to a 1-MB boundary. The lower 20 bits of the Prefetchable Memory Limit address are assumed to be F_FFFFh, which results in an alignment to the top of a 1-MB block. The maximum Memory Address range is 4 GB for 32-bit addressing, and 2^{64} bytes for 64-bit addressing.

Note: *Write the PCIPMBAR and PCIPMLMT registers with their appropriate values before setting the Command register Memory Space Enable or Master Enable bit.*

To disable the Prefetchable Memory Address range, write the Prefetchable Memory Base Address register with a value greater than that of the Prefetchable Memory Limit Address register. The entire Base register value must be greater than the entire Limit register value (*that is*, the upper 32 bits must be considered). Therefore, to disable the Address range, the Upper 32 Bits registers can both be set to the same value, while the lower Base register is set to a value greater than that of the lower Limit register; otherwise, the Upper 32-bit Base register must be greater than the Upper 32-bit Limit register.

9.4 ISA MODE

The PCI 6150 supports ISA mode by providing the Bridge Control register ISA Enable bit in Configuration space (BCNTRL[2]=1; PCI:3Eh). ISA mode modifies the PCI 6150 response inside the I/O Address range to support I/O space mapping in the presence of an ISA Bus in the system. This bit only affects the PCI 6150 response when the following conditions are met:

- Transaction falls inside the Address range defined by the I/O Base and Limit Address registers, and
- Address also falls inside the first 64 KB of I/O space (Address bits [31:16]=0h)

When the ISA Enable bit is set, the PCI 6150 does ***not*** forward downstream I/O transactions that address the upper 768 bytes of each aligned 1-KB block. Only those transactions addressing the lower 256 bytes of an aligned 1-KB block inside the Base and Limit I/O Address range are forwarded downstream. Transactions above the 64-KB I/O Address boundary are forwarded, as defined by the I/O Base and Limit register Address range.

Additionally, if the ISA Enable bit is set, the PCI 6150 forwards upstream those I/O transactions that address the upper 768 bytes of each aligned 1-KB block within the first 64 KB of I/O space. The Command Configuration register Master Enable bit must also be set (PCICR[2]=1; PCI:04h) to enable upstream forwarding. All other I/O transactions initiated on the secondary bus are forwarded upstream if the transactions fall outside the I/O Address range.

When the ISA Enable bit is set, devices downstream of the PCI 6150 can have I/O space mapped into the first 256 bytes of each 1-KB segment below the 64-KB boundary, or anywhere in I/O space above the 64-KB boundary.

9.5 VGA SUPPORT

The PCI 6150 provides two modes for VGA support:

- VGA mode, supporting VGA-compatible addressing
- VGA Snoop mode, supporting VGA palette forwarding

9.5.1 VGA Mode

When a VGA-compatible device exists downstream from the PCI 6150, enable VGA mode by setting the Bridge Control register VGA Enable bit (BCNTRL[3]=1; PCI:3Eh). When operating in VGA mode, the PCI 6150 forwards downstream those transactions that address the VGA Frame Buffer Memory and VGA I/O registers, regardless of the I/O Base and Limit Address register values. The PCI 6150 ignores transactions initiated on the secondary interface addressing these locations.

The VGA Frame buffer resides in the Memory Address range—000A_0000h to 000B_FFFFh.

Read transactions to Frame Buffer memory are treated as non-prefetchable. The PCI 6150 requests only a single Data transfer from the target, and Read Byte Enable bits are forwarded to the target bus.

The VGA I/O addresses consist of I/O addresses 3B0h to 3BBh and 3C0h to 3DFh.

These I/O addresses are aliased every 1 KB throughout the first 64 KB of I/O space [*that is*, Address bits [15:10] are not decoded and can be any value, while Address bits [31:16] must be all zeros (0)].

VGA BIOS addresses starting at C_0000h are ***not*** decoded in VGA mode.

9.5.2 VGA Snoop Mode

The PCI 6150 provides VGA Snoop mode, allowing for VGA Palette Write transactions to be forwarded downstream. This mode is used when a graphics device downstream from the PCI 6150 must snoop or respond to VGA Palette Write transactions. To enable the mode, set the Command register VGA Palette Snoop Enable bit (PCICR[5]=1; PCI:04h). The PCI 6150 claims VGA Palette Write transactions by asserting DEVSEL# in VGA Snoop mode.

When the VGA Palette Snoop Enable bit is set, the PCI 6150 forwards downstream transactions with I/O addresses 3C6h, 3C8h, and 3C9h.

These addresses are also forwarded as part of the previously described VGA Compatibility mode. Again, Address bits [15:10] are ***not*** decoded, while Address bits [31:16] must be equal to 0h (*that is*, these addresses are aliased every 1 KB throughout the first 64 KB of I/O space).

Note: *If BCNTRL[3]=1; PCI:3Eh (VGA Enable bit), then VGA Palette accesses are forwarded, regardless of the PCICR[5]; PCI:04h value.*

10 TRANSACTION ORDERING

This section describes the ordering rules that control PCI transaction forwarding across the PCI 6150. To maintain data coherency and consistency, the PCI 6150 complies with *PCI r2.3* ordering rules. For a detailed discussion of transaction ordering, refer to *PCI r2.3,* Appendix E.

10.1 TRANSACTIONS GOVERNED BY ORDERING RULES

Ordering relationships are established for the following transaction classes that cross the PCI 6150:

- **Posted Write Transactions (Comprised of Memory Write, and Memory Write and Invalidate, Transactions)**—Completed at the source before completing at the destination (*that is*, data is written into intermediate Data buffers before reaching the target).
- **Delayed Write Request Transactions (Comprised of I/O Write and Configuration Write Transactions)**—Terminated by Target Retry on the initiator bus and queued in the Delayed Transaction queue. A Delayed Write transaction must complete on the target bus before completing on the initiator bus.
- **Delayed Write Completion Transactions (Comprised of I/O Write and Configuration Write Transactions)**—Completed on the target bus, with the target response queued in the buffers. A Delayed Write Completion transaction proceeds in the direction opposite to that of the original Delayed Write request (*that is*, the transaction proceeds from target-to-initiator bus).
- **Delayed Read Request Transactions (Comprised of all Memory Read, I/O Read, and Configuration Read Transactions)**—Terminated by Target Retry on the initiator bus and queued in the Delayed Transaction queue.
- **Delayed Read Completion Transactions (Comprised of all Memory Read, I/O Read, and Configuration Read Transactions)**—Completed on the target bus, and the Read data was queued in the Read Data buffers. A Delayed Read Completion transaction proceeds in the direction opposite that of the original Delayed Read request (*that is*, the transaction proceeds from target-to-initiator bus).

The PCI 6150 does ***not*** combine, merge, nor collapse Write transactions:

- **Combine separate Write transactions into a single Write transaction**—This optimization is best implemented in the originating master.
- **Merge bytes on separate Masked Write transactions to the same Dword address**—This optimization is also best implemented in the originating master.
- **Collapse sequential Write transactions to the same address into a single Write transaction**—*PCI r2.3* does not allow collapsing of transactions.

10.2 GENERAL ORDERING GUIDELINES

PCI-independent transactions on the primary and secondary buses have a relationship only when those transactions cross the PCI 6150.

The following general ordering guidelines govern transactions crossing the PCI 6150:

- Ordering relationship of a transaction, with respect to other transactions, is determined when the transaction completes (*that is*, when a transaction ends with a Termination other than Target Retry).
- Requests terminated with a Target Retry can be accepted and completed in any order with respect to other transactions terminated with a Target Retry. If the order of completion of Delayed requests is important, the initiator should not start a second Delayed transaction until the first one completes. If more than one Delayed transaction is initiated, the initiator should repeat all the Delayed transaction requests, using a fairness algorithm. Repeating a Delayed transaction ***cannot*** be contingent upon completion of another Delayed transaction; otherwise, deadlock may occur.
- Write transactions flowing in one direction have no ordering requirements with respect to Write transactions flowing in the other direction. The PCI 6150 can simultaneously accept Posted Write transactions on both interfaces, as well as simultaneously initiate Posted Write transactions on both interfaces.

- Acceptance of a Posted Memory Write transaction as a target can never be contingent on the completion of an Unlocked, Unposted transaction as a master. This is true of the PCI 6150 and must also be true of other bus agents; otherwise, deadlock may occur.
- PCI 6150 accepts Posted Write transactions, regardless of the state of completion of Delayed transactions being forwarded across the PCI 6150.

10.3 ORDERING RULES

The following ordering rules describe the transaction relationships. Each ordering rule is followed by an explanation, and the ordering rules are referred to by number in Table 10-1. These ordering rules apply to Posted Write transactions, Delayed Write and Read requests, and Delayed Write and Read Completion transactions crossing the PCI 6150 in the same direction. Note that Delayed Completion transactions cross the PCI 6150 in the direction opposite that of the corresponding Delayed requests.

1. **Posted Write**—Posted Write transactions must complete on the target bus in the order in which the transactions were received on the initiator bus.

   The subsequent Posted Write transaction could be setting a flag that covers the data in the first Posted Write transaction. If the second transaction were to complete before the first transaction, devices checking that flag could subsequently be using stale data.
2. **Delayed Write Request**—Delayed Write requests ***cannot*** pass previously queued Posted Write data. As in the case of Posted Memory Write transactions, the Delayed Write transaction might be setting a flag regarding data in the Posted Write transaction. If the Delayed Write request were to complete before the earlier Posted Write transaction, devices checking the flag could subsequently be using stale data.
3. **Delayed Read Request**—Delayed Read requests traveling in the same direction as previously queued Posted Write transactions must push the Posted Write data ahead of it. The Posted Write transaction must complete on the target bus before the Delayed Read request can be attempted on the target bus.

   The Read transaction might be in the same location as the Write data; therefore, if the Read transaction were to pass the Write transaction, the read would return stale data.
4. **Delayed Write Completion**—Posted Write transactions must be provided opportunities to pass Delayed Read and Write requests and completions. Otherwise, deadlock may occur when bridges that support Delayed transactions are used in the same system with bridges that do not support Delayed transactions. A fairness algorithm is used to arbitrate between the Posted Write and Delayed Transaction queues.

   The PCI 6150 can return Delayed Read transactions in a different order than requested if the DRT Out-of-Order Enable bit is set to 1 (MSCOPT[2]=1; PCI:46h). Requested cycles can execute out of order across the bridge, if all other ordering rules are satisfied. Therefore, if the PCI 6150 starts a Delayed transaction that is Retried by the target, the PCI 6150 can execute another transaction in the Delayed Transaction Request queue. Also, if there are Delayed Write and Read requests in the queue, and the Read Data FIFOs are full, the PCI 6150 may execute the Delayed Write request before the Delayed Read request.
5. **Delayed Read Completion**—Delayed Read completions must “pull” ahead of previously queued Posted Write data traveling in the same direction. In this case, the Read data is traveling in the same direction as the Write data, and the initiator of the Read transaction is on the same side of the bridge as the target of the Write transaction. The Posted Write transaction must complete on the target before Read data is returned to the initiator.

   The Read transaction could be to a Status register of the initiator of the Posted Write data and therefore should not complete until the Write transaction is complete.The PCI 6150 can generate cycles across the bridge in the same order requested if the Miscellaneous Options register DRT Out-of-Order Enable bit is set (MSCOPT[2]=1; PCI:46h). By default, requested cycles can execute out of order across the bridge if all other ordering rules are satisfied. Therefore, if the PCI 6150 begins a Delayed transaction that is Retried by the target, the PCI 6150 can execute another transaction in the Delayed Transaction Request queue. Additionally, if there is both Delayed Write and Delayed Read requests in the queue, and the Read Data FIFO is full, the PCI 6150 may execute the Delayed Write request before the Delayed Read request.

On cycle completion, the PCI 6150 may complete cycles in a different order than that requested by the initiator.

10.4 DATA SYNCHRONIZATION

Data synchronization refers to the relationship between interrupt signaling and data delivery. *PCI r2.3* provides the following alternative methods for synchronizing data and interrupts:

- Device signaling the interrupt performs a read of the data just written (software)
- Device driver performs a Read operation to any register in the interrupting device before accessing data written by the device (software)
- System hardware guarantees that Write buffers are flushed before interrupts are forwarded

The PCI 6150 does not have a hardware mechanism to guarantee data synchronization for Posted Write transactions. Therefore, all Posted Write transactions must be followed by a Read operation, from the PCI 6150 to the location recently written (or some other location along the same path), or from the device driver to a PCI 6150 register.

Table 10-1. Transaction Ordering Summary

Pass	Posted Write	Delayed Write Request	Delayed Read Request	Delayed Write Completion	Delayed Read Completion
Posted Write	N[1]	Y[4]	Y[4]	Y[4]	Y[4]
Delayed Write Request	N[5]	Y	Y	Y	Y
Delayed Read Request	N[3]	Y	Y	Y	Y
Delayed Write Completion	Y	Y	Y	Y	Y
Delayed Read Completion	N[2]	Y	Y	Y	Y

Legend:

Superscript Number = *Refers to the five applicable ordering rules listed in Section 10.3. Many entries are not governed by these ordering rules; therefore, the implementation can choose whether the transactions pass each other.*

Y = *Transactions may be completed out of order or "pass" each other.*

N = *Row transaction must* ***not*** *pass the column transaction.*

11 ERROR HANDLING

This section provides detailed information regarding PCI 6150 error management. It also describes error status reporting and error operation disabling.

11.1 OVERVIEW

The PCI 6150 checks, forwards, and generates parity on the primary and secondary interfaces. To maintain transparency, the PCI 6150 forwards the existing parity condition from one bus to the other, along with address and data, or regenerates the data parity on the other bus (MSCOPT[3]=1; PCI:46h).

To support error reporting on the PCI Bus, the PCI 6150 implements the following:

- P_PERR#, P_SERR#, S_PERR#, and S_SERR# signals
- Primary and secondary Status registers (PCISR; PCI:06h and PCISSR; PCI:1Eh, respectively)
- Device-specific P_SERR# Event Disable and Status registers (PSERRED; PCI:64h and PSERRSR; PCI:6Ah, respectively)

11.2 ADDRESS PARITY ERRORS

The PCI 6150 checks address parity for all Bus transactions, and Address and Bus commands.

When the PCI 6150 detects an Address Parity error on the primary interface, the following occurs:

1. If the Command register Parity Error Response Enable bit is set (PCICR[6]=1; PCI:04h), the PCI 6150 does not claim the transaction with P_DEVSEL#. This may allow the transaction to terminate in a Master Abort.

   If the Parity Error Response Enable bit is ***not*** set, the PCI 6150 proceeds as usual and accepts the transaction if the transaction is directed to, or across, the PCI 6150.
2. PCI 6150 sets the Status register Parity Error Detected bit (PCISR[15]=1; PCI:06h).
3. PCI 6150 asserts P_SERR# and sets the Status register Signaled System Error bit (PCISR[14]=1), if the Command register P_SERR# Enable and Parity Error Response Enable bits are set (PCICR[8, 6]=11b; PCI:04h).

When the PCI 6150 detects an Address Parity error on the secondary interface, the following occurs:

1. If the Bridge Control register Parity Error Response Enable bit is set (BCNTRL[0]=1; PCI:3Eh), the PCI 6150 does not claim the transaction with S_DEVSEL#. This may allow the transaction to terminate in a Master Abort.

   If the Parity Error Response Enable bit is ***not*** set, the PCI 6150 proceeds as usual and accepts the transaction if the transaction is directed to, or across, the PCI 6150.
2. PCI 6150 sets the Secondary Status register Parity Error Detected bit (PCISSR[15]=1; PCI:1Eh), regardless of the Parity Error Response Enable bit state (PCICR[6]=x).
3. PCI 6150 asserts S_SERR# and sets the Status register Signaled System Error bit (PCISSR[14]=1).

11.3 DATA PARITY ERRORS

When forwarding transactions, the PCI 6150 attempts to pass the data parity condition from one interface to the other unchanged, whenever possible, to allow the master and target devices to manage the error condition.

The following subsections describe, for each transaction, the sequence that occurs when a Parity error is detected and the way in which the parity condition is forwarded across the bridge.

11.3.1 Configuration Write Transactions to Configuration Space

When the PCI 6150 detects a Data Parity error during a Type 0 Configuration Write transaction to Configuration space, the following occurs:

1. If the Command register Parity Error Response Enable bit is set (PCICR[6]=1; PCI:04h), the PCI 6150 asserts P_PERR#. If the Parity Error Response Enable bit is ***not*** set, the PCI 6150 does ***not*** assert P_PERR#. In either case, the Configuration register is written.
2. PCI 6150 sets the Status register Parity Error Detected bit (PCISR[15]=1; PCI:06h), regardless of the Parity Error Response Enable bit state (PCICR[6]=x).

11.3.2 Read Transactions

When the PCI 6150 detects a Parity error during a Read transaction, the target drives data and data parity, and the initiator checks parity and conditionally asserts P_PERR# or S_PERR#.

For downstream transactions, when the PCI 6150 detects a Read Data Parity error on the secondary bus, the PCI 6150:

1. Asserts S_PERR# two cycles following the Data transfer, if the secondary interface Bridge Control register Parity Error Response Enable bit is set (BCNTRL[0]=1; PCI:3Eh).
2. Sets the secondary Status register Parity Error Detected bit (PCISSR[15]=1; PCI:1Eh), regardless of the Parity Error Response Enable bit state (PCICR[6]=x).
3. Sets the secondary Status register Data Parity Error Detected bit (PCISSR[8]=1), if BCNTRL[0]=1.
4. Returns the bad parity with the data to the initiator on the primary bus. If the data with the bad parity is prefetched and not read by the initiator on the primary bus, the data is discarded and data with bad parity is not returned to the initiator.
5. Completes the transaction as usual.

For upstream transactions, when the PCI 6150 detects a Read Data Parity error on the primary bus, the PCI 6150:

1. Asserts P_PERR# two cycles following the Data transfer, if the primary interface Command register Parity Error Response Enable bit is set (PCICR[6]=1).
2. Sets the primary Status register Parity Error Detected bit (PCISR[15]=1).
3. Sets the primary Status register Data Parity Error Detected bit (PCISR[8]=1), if PCICR[6]=1.
4. Returns the bad parity with the data to the initiator on the secondary bus. If the data with the bad parity is prefetched and not read by the initiator on the secondary bus, the data is discarded and data with bad parity is not returned to the initiator.
5. Completes the transaction as usual.

The PCI 6150 returns to the initiator the data and parity received from the target. When the initiator detects a Parity error on this Read data and is enabled to report the error, the initiator asserts its PERR# signal (which is then connected to the PCI 6150 P_PERR# or S_PERR# signal, depending on the initiator bus) two cycles after the Data transfer. It is assumed that the initiator is responsible for handling Parity error conditions; therefore, when the PCI 6150 detects the initiator's PERR# assertion while returning Read data to the initiator, the PCI 6150 takes no further action and completes the transaction as usual.

11.3.3 Posted Write Transactions

During downstream Posted Write transactions, when the PCI 6150 is responding as a target and detects a Data Parity error on the initiator (primary) bus, it:

1. Asserts P_PERR# two cycles after the Data transfer, if the primary interface Command register Parity Error Response Enable bit is set (PCICR[6]=1).
2. Sets the primary interface Status register Parity Error Detected bit (PCISR[15]=1).
3. Captures and forwards the bad parity condition to the secondary bus.
4. Completes the transaction as usual.

Similarly, during upstream Posted Write transactions, when the PCI 6150 is responding as a target and detects a Data Parity error on the initiator (secondary) bus, it:

1. Asserts S_PERR# two cycles after the Data transfer, if the secondary interface Bridge Control register Parity Error Response Enable bit is set (BCNTRL[0]=1).
2. Sets the secondary interface Status register Parity Error Detected bit (PCISSR[15]=1), regardless of the Parity Error Response Enable bit state (PCICR[6]=x).
3. Captures and forwards the bad parity condition to the primary bus.
4. Completes the transaction as usual.

During downstream Write transactions, when a Data Parity error is reported on the target (secondary) bus by the target's assertion of S_PERR#, the PCI 6150:

1. Sets the secondary Status register Data Parity Error Detected bit (PCISSR[8]=1), if the secondary interface Bridge Control register Parity Error Response Enable bit is set (BCNTRL[0]=1).
2. Asserts P_SERR# and sets the Status register Signaled System Error bit (PCISR[14]=1), if the following conditions are met:
   - Primary interface Command register P_SERR# Enable and Parity Error Response Enable bits are set (PCICR[8, 6]=11b, respectively), and
   - Device-specific P_SERR# Disable bit for Posted Write Parity errors is ***not*** set (PSERRED[1]=0; PCI:64h), and
   - Secondary interface Bridge Control register Parity Error Response Enable bit is set (BCNTRL[0]=1), and
   - PCI 6150 did not detect the Parity error on the initiator (primary) bus (*that is*, the Parity error was not forwarded from the primary bus)

During upstream Write transactions, when a Data Parity error is reported on the target (primary) bus by the target's assertion of P_PERR#, the PCI 6150:

1. Sets the Status register Data Parity Error Detected bit (PCISR[8]=1), if the primary interface Command register Parity Error Response Enable bit is set (PCICR[6]=1).
2. Asserts P_SERR# and sets the Status register Signaled System Error bit (PCISR[14]=1), if the following conditions are met:
   - Primary interface Command register P_SERR# Enable and Parity Error Response Enable bits are set (PCICR[8, 6]=11b, respectively), and
   - Secondary interface Bridge Control register Parity Error Response Enable bit is set (BCNTRL[0]=1), and
   - PCI 6150 did not detect the Parity error on the initiator (secondary) bus (*that is*, the Parity error was not forwarded from the secondary bus)

P_SERR# assertion signals the Parity error condition when the initiator is not sent information about an error having occurred. Because the data is delivered with no errors, there is no other way to signal this information to the initiator.

If a Parity error is forwarded from the initiator bus to the target bus, P_SERR# is ***not*** asserted.

11.3.4 Delayed Write Transactions

When the PCI 6150 detects a Data Parity error during a Delayed Write transaction, it conditionally asserts PERR#. The PCI 6150 passes or regenerates data parity to the target bus (MSCOPT[3]=1; PCI:46h). A Parity error can occur:

- During the original Delayed Write Request transaction
- When the initiator repeats the Delayed Write Request transaction
- When the PCI 6150 completes the Delayed Write transaction to the target

When a Delayed Write transaction is queued, the Address, Command, Address and Data Parity, Data, and Byte Enable bits are captured and a Target Retry is returned to the initiator. When the PCI 6150 detects a Parity error on the Write data for the initial Delayed Write Request transaction, the following occurs:

1. If the Parity Error Response Enable bit corresponding to the initiator bus is set (primary—PCICR[6]=1, secondary—BCNTRL[0]=1), the PCI 6150 asserts P_PERR# or S_PERR# two clocks after the data. The PCI 6150 always accepts the cycle, and can optionally pass the incorrect parity to the other bus, or regenerate data parity on the other bus (MSCOPT[3]=1; PCI:46h).
2. PCI 6150 sets the Status register Parity Error Detected bit corresponding to the initiator bus (primary—PCISR[15]=1, secondary—PCISSR[15]=1), regardless of the Parity Error Response Enable bit state (PCICR[6]=x).

Following the initiating transaction (the first PCI 6150 Retry), the subsequent Data Parity error of a similar transaction on the initiating bus is detected as usual; however, the Data Parity error no longer affects FIFO operation. The cycles are considered similar if they have the same Address, Command, Byte Enables and Write data. The Parity bit is not part of this "similar" detection operation. Therefore, if a Data Parity error occurs only in the Parity bit (same data as before), the cycle operates as usual. Conversely, if a Data Parity error occurs in the data segment (different data from the initiating Write data), the PCI 6150 treats the error as a new Delayed Write transaction.

11.4 DATA PARITY ERROR REPORTING SUMMARY

In the previous subsections, the PCI 6150 responses to Data Parity errors are presented according to transaction type in progress. This subsection organizes the PCI 6150 responses to Data Parity errors according to the Status bits set by the PCI 6150 and the signals asserted.

Table 11-1 delineates the primary interface Status register Parity Error Detected bit status. This bit is set when the PCI 6150 detects a Parity error on the primary interface.

Table 11-2 delineates the secondary interface Status register Parity Error Detected bit status. This bit is set when the PCI 6150 detects a Parity error on the secondary interface.

Table 11-3 delineates the primary interface Status register Data Parity Error Detected bit status. This bit is set under the following conditions:

- PCI 6150 must be a master on the primary bus, and
- Primary interface Command register Parity Error Response Enable bit must be set (PCICR[6]=1), and
- P_PERR# is detected asserted, or a Parity error is detected on the primary bus

Table 11-4 delineates the secondary interface Status register Data Parity Error Detected bit status. This bit is set under the following conditions:

- PCI 6150 must be a master on the secondary bus, and
- Secondary interface Bridge Control register Parity Error Response Enable bit must be set (BCNTRL[0]=1), and
- S_PERR# is detected asserted, or a Parity error is detected on the secondary bus

Table 11-5 delineates P_PERR# assertion. This signal is set under the following conditions:

- PCI 6150 is either the target of a Write transaction or the initiator of a Read transaction on the primary bus, and
- Primary interface Command register Parity Error Response Enable bit must be set (PCICR[6]=1), and
- PCI 6150 detects a Data Parity error on the primary bus, or detects S_PERR# asserted during the Completion phase of a downstream Delayed Write transaction on the target (secondary) bus

Table 11-6 delineates S_PERR# assertion. This signal is set under the following conditions:

- PCI 6150 is either the target of a Write transaction or the initiator of a Read transaction on the secondary bus, and
- Secondary interface Bridge Control register Parity Error Response Enable bit must be set (BCNTRL[0]=1), and
- PCI 6150 detects a Data Parity error on the secondary bus, or detects P_PERR# asserted during the Completion phase of an upstream Delayed Write transaction on the target (primary) bus

Table 11-7 delineates P_SERR# or S_SERR# assertion. This signal is set under the following conditions:

- Command register P_SERR# Enable and Parity Error Response Enable bits must be set (PCICR[8, 6]=11b, respectively), and
- Bridge Control register Parity Error Response Enable bit must be set (BCNTRL[0]=1), and
- PCI 6150 detects S_PERR# asserted on a downstream Posted Write transaction, or P_PERR# asserted on an upstream Posted Write transaction, and
- PCI 6150 did not detect the Parity error as a target of the Posted Write transaction

Table 11-1. Primary Interface Parity Error Detected Bit Status

Primary Parity Error Detected Bit (PCISR[15])	Transaction Type	Direction	Bus on which Error Detected	Primary Parity Error Response Enable Bit (PCICR[6])	Secondary Parity Error Response Enable Bit (BCNTRL[0])
0	Read	Downstream	Primary	x	x
0			Secondary	x	x
1		Upstream	Primary	x	x
0			Secondary	x	x
1	Posted Write	Downstream	Primary	x	x
0			Secondary	x	x
0		Upstream	Primary	x	x
0			Secondary	x	x
1	Delayed Write	Downstream	Primary	x	x
0			Secondary	x	x
0		Upstream	Primary	x	x
0			Secondary	x	x

Note: *x = Don't care.*

Table 11-2. Secondary Interface Parity Error Detected Bit Status

Secondary Parity Error Detected Bit (PCISSR[15])	Transaction Type	Direction	Bus on which Error Detected	Primary Parity Error Response Enable Bit (PCICR[6])	Secondary Parity Error Response Enable Bit (BCNTRL[0])
0	Read	Downstream	Primary	x	x
1			Secondary	x	x
0		Upstream	Primary	x	x
0			Secondary	x	x
0	Posted Write	Downstream	Primary	x	x
0			Secondary	x	x
0		Upstream	Primary	x	x
1			Secondary	x	x
0	Delayed Write	Downstream	Primary	x	x
0			Secondary	x	x
0		Upstream	Primary	x	x
1			Secondary	x	x

Note: *x = Don’t care.*

Table 11-3. Primary Interface Data Parity Error Detected Bit Status

Primary Data Parity Error Detected Bit (PCISR[8])	Transaction Type	Direction	Bus on which Error Detected	Primary Parity Error Response Enable Bit (PCICR[6])	Secondary Parity Error Response Enable Bit (BCNTRL[0])
0	Read	Downstream	Primary	x	x
0			Secondary	x	x
1		Upstream	Primary	1	x
0			Secondary	x	x
0	Posted Write	Downstream	Primary	x	x
0			Secondary	x	x
1		Upstream	Primary	1	x
0			Secondary	x	x
0	Delayed Write	Downstream	Primary	x	x
0			Secondary	x	x
1		Upstream	Primary	1	x
0			Secondary	x	x

Note: *x = Don't care.*

Table 11-4. Secondary Interface Data Parity Error Detected Bit Status

Secondary Data Parity Error Detected Bit (PCISSR[8])	Transaction Type	Direction	Bus on which Error Detected	Primary Parity Error Response Enable Bit (PCICR[6])	Secondary Parity Error Response Enable Bit (BCNTRL[0])
0	Read	Downstream	Primary	x	x
1			Secondary	x	1
0		Upstream	Primary	x	x
0			Secondary	x	x
0	Posted Write	Downstream	Primary	x	x
1			Secondary	x	1
0		Upstream	Primary	x	x
0			Secondary	x	x
0	Delayed Write	Downstream	Primary	x	x
1			Secondary	x	1
0		Upstream	Primary	x	x
0			Secondary	x	x

Note: *x = Don't care.*

Table 11-5. P_PERR# Assertion

P_PERR#	Transaction Type	Direction	Bus on which Error Detected	Primary Parity Error Response Enable Bit (PCICR[6])	Secondary Parity Error Response Enable Bit (BCNTRL[0])
1 (De-asserted)	Read	Downstream	Primary	x	x
1			Secondary	x	x
0 (Asserted)		Upstream	Primary	1	x
1			Secondary	x	x
0	Posted Write	Downstream	Primary	1	x
1			Secondary	x	x
1		Upstream	Primary	x	x
1			Secondary	x	x
0	Delayed Write	Downstream	Primary	1	x
0*			Secondary	1	1
1		Upstream	Primary	x	x
1			Secondary	x	x

Notes: *x = Don't care.*

** Parity error detected on the target (secondary) bus, but not on the initiator (primary) bus.*

Table 11-6. S_PERR# Assertion

S_PERR#	Transaction Type	Direction	Bus on which Error Detected	Primary Parity Error Response Enable Bit (PCICR[6])	Secondary Parity Error Response Enable Bit (BCNTRL[0])
1 (De-asserted)	Read	Downstream	Primary	x	x
0 (Asserted)			Secondary	x	1
1		Upstream	Primary	x	x
1			Secondary	x	x
1	Posted Write	Downstream	Primary	x	x
1			Secondary	x	x
1		Upstream	Primary	x	x
0			Secondary	x	1
1	Delayed Write	Downstream	Primary	x	x
1			Secondary	x	x
0*		Upstream	Primary	1	1
0			Secondary	x	1

Notes: *x = Don't care.*

** Parity error detected on the target (secondary) bus, but not on the initiator (primary) bus.*

Table 11-7. P_SERR# or S_SERR# for Data Parity Error Assertion

P_SERR# or S_SERR#	Transaction Type	Direction	Bus on which Error Detected	Primary Parity Error Response Enable Bit (PCICR[6])	Secondary Parity Error Response Enable Bit (BCNTRL[0])
1 (De-asserted)	Read	Downstream	Primary	x	x
1			Secondary	x	x
1		Upstream	Primary	x	x
1			Secondary	x	x
1	Posted Write	Downstream	Primary	x	x
0* (Asserted)			Secondary	1	1
0**		Upstream	Primary	1	1
1			Secondary	x	x
1	Delayed Write	Downstream	Primary	x	x
1			Secondary	x	x
1		Upstream	Primary	x	x
1			Secondary	x	x

Notes: *x = Don’t care.*

** Parity error detected on the target (secondary) bus, but not on the initiator (primary) bus.*

*** Parity error detected on the target (primary) bus, but not on the initiator (secondary) bus*

11.5 SYSTEM ERROR (P_SERR#) REPORTING

The PCI 6150 uses the P_SERR# signal to conditionally report a number of System error conditions in addition to the special case Parity error conditions.

In this data book, when P_SERR# assertion is discussed, the following conditions are assumed:

- For the PCI 6150 to assert P_SERR#, the Command register P_SERR# Enable bit must be set (PCICR[8]=1)
- When the PCI 6150 asserts P_SERR#, the PCI 6150 must also set the Status register Signaled System Error bit (PCISR[14]=1)

In compliance with *P-to-P Bridge r1.1*, the PCI 6150 asserts P_SERR# when it detects S_SERR# input asserted and the Bridge Control register S_SERR# Enable bit is set (BCNTRL[1]=1). In addition, the PCI 6150 also sets the secondary Status register Signaled System Error bit (PCISSR(14]=1).

Note: *S_SERR# is an I/O pin.*

The PCI 6150 also conditionally asserts P_SERR# for the following conditions:

- Master Abort detected during Posted Write transaction (on the secondary bus)
- Target Abort detected during Posted Write transaction (on the secondary bus)
- Posted Write data discarded after 2^{24} delivery attempts (2^{24} Target Retries received)
- S_PERR# reported on the target bus during a Posted Write transaction (refer to Section 11.4)
- Delayed Write data discarded after 2^{24} delivery attempts (2^{24} Target Retries received)
- Delayed Read data ***cannot*** be transferred from the target after 2^{24} attempts (2^{24} Target Retries received)
- Master Timeout on Delayed transaction

The device-specific P_SERR# Status register reports the reason for P_SERR# assertion.

Most of these events have additional device-specific Disable bits in the P_SERR# Event Disable register that can mask P_SERR# assertion for specific events. The Master Timeout condition has S_SERR# and P_SERR# Enable bits for that event in the Bridge Control register (BCNTRL[12:11], respectively), and therefore does not have a device-specific Disable bit.

12 EXCLUSIVE ACCESS

This section describes P_LOCK# and S_LOCK# signal use to implement exclusive access to a target for transactions crossing the PCI 6150, including concurrent locks, and acquiring and ending exclusive access.

12.1 CONCURRENT LOCKS

The primary and secondary bus Lock mechanisms concurrently operate, *except* when a Locked transaction is crossing the PCI 6150. A primary master can lock a primary target without affecting the Lock status on the secondary bus, and vice versa. This means that a primary master can lock a primary target concurrent with a secondary master locking a secondary target.

12.2 ACQUIRING EXCLUSIVE ACCESS ACROSS PCI 6150

For a PCI Bus, before acquiring access to the P_LOCK# and/or S_LOCK# signal and starting a series of Locked transactions, the initiator must first verify whether the following conditions are met:

- PCI Bus is idle, and
- P_LOCK# and/or S_LOCK# is de-asserted

The initiator leaves P_LOCK# and/or S_LOCK# de-asserted during the Address phase and asserts P_LOCK# and/or S_LOCK# one Clock cycle later. Target lock is achieved after the target completes a Data transfer.

Locked transactions can cross the PCI 6150 in the downstream and upstream directions, from the primary-to-secondary bus and vice versa.

When the target resides on another PCI Bus, the master must acquire not only the lock on its own PCI Bus, but also the lock on every bus between its bus and the target bus. When the PCI 6150 detects an initial Locked transaction on the primary bus that is intended for a target on the secondary bus, the PCI 6150 samples the Address, Transaction Type, Byte Enable, and Parity bits, and the S_LOCK# signal. Because a Target Retry is signaled to the initiator, the initiator must relinquish the lock on the primary bus, and therefore the lock is not yet established.

The first Locked transaction must be a Read transaction. Subsequent Locked transactions can be Write or Read transactions. Posted Memory Write transactions that are part of the Locked-transaction sequence are nevertheless posted. Memory Read transactions that are part of the Locked-transaction sequence are ***not*** prefetched.

When the Locked Delayed Read request is queued, the PCI 6150 does ***not*** queue further transactions until the locked sequence is complete. The PCI 6150 signals a Target Retry to all transactions initiated subsequent to the Locked Read transaction that are intended for targets on the opposite side of the PCI 6150. The PCI 6150 allows transactions queued before the Locked transaction to complete before initiating the Locked transaction.

When the Locked Delayed Read request moves to the head of the Delayed Transaction queue, the PCI 6150 initiates the request as a Locked Read transaction by de-asserting S_LOCK# on the target bus during the first Address phase, then re-asserting S_LOCK# one cycle later. If S_LOCK# was previously asserted (used by another initiator), the PCI 6150 waits to request access to the secondary bus until S_LOCK# is sampled de-asserted when the target bus is idle. Note that the existing lock on the target bus did not cross the PCI 6150; otherwise, the pending queued Locked transaction would not have queued. When the PCI 6150 is able to complete a Data transfer with the Locked Read transaction, the lock is established on the secondary bus.

When the initiator repeats the Locked Read transaction on the primary bus with the same Address, Transaction Type, Byte Enable, and Parity bits, the PCI 6150 transfers the Read data back to the initiator, and the lock is also established on the primary bus.

For the PCI 6150 to recognize and respond to the initiator, the initiator's subsequent Read transaction attempts must use the Locked-transaction sequence (de-assert P_LOCK# during the Address phase, then re-assert P_LOCK# one cycle later). If the P_LOCK# sequence is not used in subsequent attempts, a Master Timeout condition may result. When a Master Timeout condition occurs, P_SERR# is conditionally asserted, the Read data and queued Read transaction are discarded, and S_LOCK# is de-asserted on the target bus.

After the intended target is locked, subsequent Locked transactions initiated on the initiator bus that are forwarded by the PCI 6150 are driven as Locked transactions on the target bus.

When the PCI 6150 receives a Master or Target Abort in response to the Delayed Locked Read transaction, this status is passed back to the initiator, and no locks are established on the initiator or target bus. The PCI 6150 resumes Unlocked transaction forwarding in both directions.

12.3 ENDING EXCLUSIVE ACCESS

After the lock is acquired on the initiator and target buses, the PCI 6150 must maintain the lock on the target bus for subsequent Locked transactions until the initiator relinquishes the lock.

The only time a Target Retry causes the lock to be relinquished is on the first transaction of a Locked sequence. On subsequent transactions in the sequence, the Target Retry has no effect on the P_LOCK# and/or S_LOCK# signal status.

An established target lock is maintained until the initiator relinquishes the lock. The PCI 6150 does not recognize whether the current transaction is the last in a sequence of Locked transactions until the initiator de-asserts P_LOCK# and/or S_LOCK# at the end of the transaction.

When the last Locked transaction is a Delayed transaction, the PCI 6150 previously completed the transaction on the secondary bus. In this case, when the PCI 6150 detects that the initiator has relinquished the P_LOCK# and/or S_LOCK# signal by sampling the signal de-asserted while P_FRAME# or S_FRAME# is de-asserted, the PCI 6150 de-asserts P_LOCK# and/or S_LOCK# on the target bus when possible. Because of this behavior, P_LOCK# and/or S_LOCK# may not be de-asserted until several cycles after the last Locked transaction completes on the target bus. After de-asserting P_LOCK# and/or S_LOCK# to indicate the end of a sequence of Locked transactions, the PCI 6150 resumes Unlocked transaction forwarding.

When the last Locked transaction is a Posted Write, the PCI 6150 de-asserts P_LOCK# and/or S_LOCK# on the target bus at the end of the transaction because the lock was relinquished at the end of the Write transaction on the initiator bus.

When the PCI 6150 receives a Master or Target Abort in response to a Locked Delayed transaction, the PCI 6150 returns a Master or Target Abort when the initiator repeats the Locked transaction. The initiator must then de-assert P_LOCK# and/or S_LOCK# at the end of the transaction. The PCI 6150 sets the appropriate Status bits, flagging the abnormal Target Termination condition, and normal forwarding of Unlocked Posted and Delayed transactions resumes.

When the PCI 6150 receives a Master or Target Abort in response to a Locked Posted Write transaction, the PCI 6150 ***cannot*** communicate that status to the initiator. The PCI 6150 asserts P_SERR# on the initiator bus when a Master or Target Abort is received during a Locked Posted Write transaction, if the Command register P_SERR# Enable bit is set (PCICR[8]=1; PCI:04h). P_SERR# is asserted for the Master Abort condition if the Bridge Control register Master Abort Mode bit is set (BCNTRL[5]=1; PCI:3Eh).

Note: *The PCI 6150 has an option to ignore the Lock protocol, by clearing the Secondary and/or Primary Lock Enable bits (MSCOPT[14:13]=00b; PCI:46h, respectively).*

13 PCI BUS ARBITRATION

This section describes primary and secondary bus arbitration and bus parking.

13.1 OVERVIEW

The PCI 6150 must arbitrate for use of the secondary bus when forwarding downstream transactions, and for the primary bus when forwarding upstream transactions. The primary bus Arbiter is external to the PCI 6150 (typically located on the motherboard). For the secondary PCI Bus, the PCI 6150 has a built-in Internal Arbiter. The Internal Arbiter can be disabled, allowing use of an External Arbiter for secondary bus arbitration.

13.2 PRIMARY PCI BUS ARBITRATION

The PCI 6150 uses one Request output pin and one Grant input pin (P_REQ# and P_GNT#, respectively) for primary PCI Bus arbitration. The PCI 6150 asserts P_REQ# when forwarding transactions upstream (*that is*, when operating as an initiator on the primary PCI Bus). When there are one or more pending transactions in the upstream direction queues—Posted Write data or Delayed transaction requests—the PCI 6150 maintains P_REQ# assertion. However, if a Target Retry, Disconnect, or Abort is received in response to a PCI 6150-initiated transaction on the primary PCI Bus, the PCI 6150 de-asserts P_REQ# for two PCI Clock cycles. For all cycles passing through the bridge, P_REQ# is not asserted until the complete transaction request is queued.

When P_GNT# is asserted low by the primary bus Arbiter after the PCI 6150 asserts P_REQ#, the PCI 6150 initiates a transaction on the primary bus on behalf of the secondary master.

If the primary bus External Arbiter asserts the PCI 6150 P_GNT# signal when P_REQ# is not asserted, the PCI 6150 parks P_AD[31:0], P_CBE[3:0]#, and P_PAR by driving these signals to valid logic levels. If the primary bus is parked on the PCI 6150 and the PCI 6150 has a transaction to initiate on the primary bus, the PCI 6150 initiates the transaction if P_GNT# remained asserted during the cycle prior to the start of the transfer.

13.3 SECONDARY PCI BUS ARBITRATION

The PCI 6150 implements a secondary PCI Bus Internal Arbiter, which supports up to nine external bus masters in addition to the PCI 6150. If required, the Internal Arbiter can be disabled, allowing use of an External Arbiter for secondary bus arbitration.

13.3.1 Secondary Bus Arbitration Using Internal Arbiter

To use the Internal Arbiter, the secondary bus Internal Arbiter Enable pin, S_CFN#, must be tied low. The PCI 6150 has nine secondary bus Request input and Grant output pins (S_REQ[8:0]# and S_GNT[8:0]#, respectively) to support external secondary bus masters. If S_CFN# is high, S_REQ0# and S_GNT0# are reconfigured as output and input, respectively, and S_GNT[8:1]# are driven high.

Note: *S_REQ0# and S_GNT0# are I/O pins.*

The PCI 6150 uses a two-level arbitration scheme, whereby arbitration is divided into two groups—low- and high-priority. The low-priority group represents a single entry in the high-priority group. Therefore, if the high-priority group consists of *n* masters, the highest priority is assigned to the low-priority group at least once every *n*+1 transactions. Priority changes evenly among the low-priority group. Therefore, assuming all masters request the bus, members of the high-priority group are serviced *n* transactions out of *n*+1, while one member of the low-priority group is serviced once every *n*+1 transactions.

Each master can be assigned to a low- or high-priority group, through the Arbiter Control register (ACNTRL; PCI:42h).

Each group can be programmed to use a rotating- or fixed-priority scheme, through the Internal Arbiter Control register Group Fixed Arbitration bits (IACNTRL[2, 0]; PCI:50h).

13.3.2 Rotating-Priority Scheme

The secondary Arbiter supports a programmable two-level rotating algorithm that enables the nine request/grant pairs to control up to nine external bus masters. In addition, there is a request/grant pair internal to the PCI 6150, which allows the device to request and be granted access to the secondary bus. Figure 13-1 is an example of the Internal Arbiter wherein four masters, including the PCI 6150, are in the high-priority group, and five masters are in the low-priority group. Using this example, if all requests are always asserted, the highest priority rotates among the masters in the following way (the PCI 6150 is denoted as *B;* high-priority members are provided in *italic* type, and low-priority members in **boldface** type):

B, m0, m1, m2, **m3**, *B, m0, m1, m2,* **m4**,
B, m0, m1, m2, **m5**, and so forth

If all masters are assigned to one group, the algorithm defaults to a rotating-priority scheme among all masters. After reset, all external masters are assigned to the low-priority group, and the PCI 6150 to the high-priority group. Therefore, by default, the PCI 6150 receives highest priority on the secondary bus every other transaction and priority rotates evenly among the other masters.

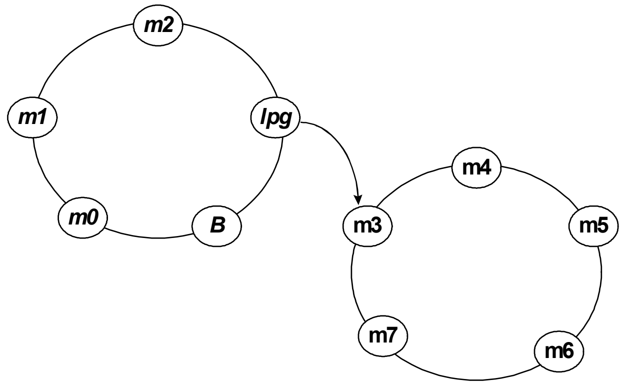


Figure 13-1. Secondary Bus Arbiter Example

Note: *In Figure 13-1, “lpg” denotes “low-priority group.”*

Priorities are re-evaluated upon S_FRAME# assertion (*that is*, at the start of each new transaction on the secondary PCI Bus). From this point, until the next transaction starts, the Arbiter asserts the Grant signal corresponding to the highest priority request asserted. If a Grant signal for a particular request is asserted, and a higher priority request subsequently asserts, the Arbiter de-asserts the asserted Grant signal and asserts the Grant signal corresponding to the new higher priority request on the next PCI Clock cycle. When priorities are re-evaluated, the highest priority is assigned to the next highest priority master, relative to the master that initiated the previous transaction. The master that initiated the last transaction now has the lowest priority within its group. Priority is also re-evaluated if the requesting agent de-asserts its request without generating cycles while the request was granted.

If the PCI 6150 detects that an initiator has failed to assert S_FRAME# after 16 cycles of Grant signal assertion and a secondary bus idle condition, the Arbiter re-evaluates grant assignment. If another initiator asserts REQ# to request the bus, the PCI 6150 switches the grant to the new initiator; otherwise, the same grant is asserted to the same initiator, even if the PCI 6150 does not assert S_FRAME# within 16 cycles.

13.3.3 Fixed-Priority Scheme

The PCI 6150 also supports a fixed-priority scheme within the low- and high-priority groups. In this case, the Internal Arbiter Control register controls whether the low- or high-priority group uses the fixed- or rotating-priority scheme (IACNTRL[2, 0]; PCI:50h). If using a fixed-priority scheme, a master within the group is assigned the highest priority within its group, and an option is set to control the priority of other masters relative to the highest priority master. This is controlled through the Internal Arbiter Control register Highest Priority Master and Group Arbitration Order bits (IACNTRL [11:4, 3, 1]; PCI:50h).

Using the example provided in Figure 13-1, but with the groups at fixed-priority, suppose that:

- Master 7 (m7) has the highest priority of the low-priority group (IACNTRL[7:4]=0111b)
- PCI 6150 (B) has the highest priority of the high-priority group (IACNTRL[11:8]=1000b)
- Priority decreases in ascending order of masters for both groups (IACNTRL[3, 1]=00b)

The order of priority with the highest first is as follows:

B, m0, m1, m2, **m7**, **m3**, **m4***,* **m5***,* **m6**

If IACNTRL[3, 1]=11b, priority increases with ascending order of bus master and the order becomes:

B, *m2*, *m1*, *m0*, **m7**, **m6**, **m5**, **m4**, **m3**

Take care when using fixed arbitration in the low-priority group. As previously noted, the low-priority group receives the grant only when there are no high-priority group requests. When the Arbiter switches to the low-priority group, the highest priority master requesting the bus within that group receives the grant. If there are several requests issued by the high-priority group members and the high-priority master in the low-priority group, then lower priority devices in the low-priority group may have to wait before receiving the grant.

To prevent bus contention, if the secondary PCI Bus is idle, the Arbiter waits at least one Clock cycle between the S_GNT*x*# de-assertion and assertion of the next S_REQ*x*#. If the secondary PCI Bus is busy (*that is*, S_FRAME# or S_IRDY# is asserted) when another bus master requests the bus, the Arbiter can de-assert one grant and assert the next grant during the same PCI Clock cycle.

13.3.4 Secondary Bus Arbitration Using External Arbiter

The Internal Arbiter can be disabled by pulling the secondary bus Internal Arbiter Enable pin (S_CFN#) high. An External Arbiter must be used if more than one bus master is required to initiate cycles on the secondary bus.

When S_CFN# is tied high, the PCI 6150 re-configures two pins to be external Request and Grant pins. S_REQ0# is re-configured to be the external Request output from the PCI 6150 and is used by the PCI 6150 to request the secondary bus. S_GNT0# is reconfigured to be the PCI 6150 external Grant input from the External Arbiter.

If the PCI 6150 requests the secondary PCI Bus (S_REQ0# asserted) and the External Arbiter grants the bus to the PCI 6150 (S_GNT0# asserted), the PCI 6150 initiates a transaction on the secondary bus one Clock cycle later.

If the secondary bus External Arbiter asserts S_GNT0# when S_REQ0# is not asserted, the PCI 6150 parks S_AD[31:0], S_CBE[3:0]#, and S_PAR by driving these signals to valid logic levels.

When using an External Arbiter, the unused secondary bus Grant outputs (S_GNT[8:1]#) are driven high. Unused secondary bus Request inputs (S_REQ[8:1]#) ***must*** be pulled high.

13.4 ARBITRATION BUS PARKING

Bus parking refers to driving the AD[31:0], CBE[3:0]#, and PAR lines to a known value while the bus is idle. The PCI Bus is parked on the PCI 6150 primary or secondary bus when one or both buses are idle. Bus parking occurs when the bus grant to the PCI 6150 on the parked bus is being asserted, and the PCI 6150 request for that bus is not asserted. The AD[31:0] and CBE[3:0]# signals are first driven low (0), then the PAR signals are driven low (0) one cycle later.

When the GNT# signal for the parked bus is de-asserted, the PCI 6150 places the AD[31:0], CBE[3:0]#, and PAR signals into a high-impedance state on the next PCI clock cycle. If the PCI 6150 is parking and wants to initiate a transaction on that bus, the PCI 6150 can start the transaction on the next PCI Clock cycle by asserting FRAME# if GNT# remains asserted.

If the secondary bus Internal Arbiter is enabled, the secondary arbiter can be optionally parked at the last active slot, or on any of the designated slots, and it can also be disabled.

The PCI 6150 has the following options related to arbitration parking, selectable through Internal Arbiter Control register Bus Grant Parking Control bits (IACNTRL[15:12]; PCI:50h):

- **No parking**—All grants are de-asserted if there are no asserted requests
- **Fixed parking**—Grant can be assigned to a specific master
- **Last master granted**—Grant is assigned to the last granted master

14 GPIO INTERFACE

This section describes the GPIO interface pins, control registers, and serial stream.

14.1 GPIO INTERFACE PINS

The PCI 6150 provides four, general-purpose I/O interface pins (GPIO[3:0]). (Refer to Table 14-1.) During normal operation, the Configuration registers control the GPIO interface. During Secondary reset, the GPIO[2:0] and MSK_IN can be used to shift in a 16-bit serial stream that serves as a secondary bus Clock Disable Mask.

The GPIO[3:0] pins have weak internal pull-up resistors. External pull-up or pull-down resistors are recommended.

Note: *MSK_IN is used in the PQFP package only. If using the PBGA package, use software to disable unused Secondary Clock buffers through the SCLKCNTRL; PCI:68h register.*

Table 14-1. GPIO Pin Operation

GPIO Pin	Alternate Function
GPIO0—Pull-up	Functions as Secondary Bus Clock Mask Shift register clock output when P_RSTIN# is asserted at 66 MHz maximum frequency.
GPIO1—Pull-up	***No alternate function.***
GPIO2 —Pull-up	Functions as Shift/Load Control Output to Shift register when P_RSTIN# is asserted. Values: 0 = Load 1 = Shift
GPIO3 —Pull-up	When GPIO3FN# is tied high, GPIO3 functions as a GPIO pin regardless of EJECT_EN# state. GPIO3 functions as Ejector input only when both GPIO3FN# and EJECT_EN# are tied low (Hot Swap enabled).

14.2 GPIO SERIAL STREAM

Refer to Section 4.3.1, “Secondary Clock Control,” on page 4-1.

14.3 GPIO CONTROL REGISTERS

The GPIO registers can be accessed from both sides of the bus. During normal operation, the GPIO interface is controlled by the following three GPIO Configuration registers:

- Output Enable (GPIOOE)
- Output Data (GPIOOD)
- Input Data (GPIOID)

The GPIO Configuration registers consist of five 8-bit fields:

- Output Enable Write 1 to Set (GPIOOE[7:4])
- Output Enable Write 1 to Clear (GPIOOE[3:0])
- Output Data Write 1 to Set (GPIOOD[7:4])
- Output Data Write 1 to Clear (GPIOOD[3:0])
- Input Data (GPIOID[7:4])

The Output Enable fields control whether the GPIO signals are inputs or outputs. Each signal is independently controlled by a bit in each Output Enable field. If 1 is written to the Write 1 to Set field, the corresponding pin is activated as an output. If 1 is written to the Write 1 to Clear field, the output driver is placed into a high-impedance state, and the pin is input only. Writing zeros (0) to these registers has no effect. The reset state for these signals is input only.

The Output Data fields also use the Write 1 to Set and Clear methods. If 1 is written to the Write 1 to Set field and the pin is enabled as an output, the corresponding GPIO output is driven high. If 1 is written to the Write 1 to Clear field and the pin is enabled as an output, the corresponding GPIO output is driven low. Writing zeros (0) to these registers has no effect. The value written to the Output Data register is driven only when the GPIO signal is configured as output. A Type 0 Configuration Write operation is used to program these registers. The reset value for the output is 0.

The Input Data field is Read-Only and reflects the current value of the GPIO[3:0] pins. A Type 0 Configuration Read operation to the Input Data register returns the values of these pins. The GPIO[3:0] pins can be read at any time, whether configured as input only or bi-directional.

15 SUPPORTED COMMANDS

This section discusses the PCI 6150 PCI command set.

15.1 PRIMARY INTERFACE COMMAND SET

Table 15-1 delineates the PCI 6150 primary interface command set.

Table 15-1. Primary Interface Supported Commands

P_CBE[3:0]#	PCI Command	Support
0000b	Interrupt Acknowledge	***Not Supported.***
0001b	Special Cycle	
0010b	I/O Read	If the address is within pass-through I/O range, the transaction is claimed and passed through. If the address points to an I/O-mapped internal bridge register, the transaction is claimed. Otherwise, the transaction is ignored.
0011b	I/O Write	Same as I/O Read (P_CBE[3:0]#=0010b).
0100b — 0101b	***Reserved***	—
0110b	Memory Read	If the address is within pass-through Memory range, the transaction is claimed and passed through. If the address points to a memory-mapped internal bridge register, the transaction is claimed. Otherwise, the transaction is ignored.
0111b	Memory Write	Same as Memory Read (P_CBE[3:0]#=0110b).
1000b – 1001b	***Reserved***	***Not Supported.***
1010b	Configuration Read	Type 0 Configuration Read, claimed if the P_IDSEL line is asserted; otherwise, the read is ignored. If claimed, the target internal register(s) is read. Never passed through. Type 1 Configuration Read, claimed if the P_IDSEL line is asserted; otherwise, the read is ignored. If the target bus is the bridge's secondary bus, the transaction is claimed and passed through as a Type 0 Configuration Read. If the target bus is a subordinate bus that exists behind the bridge (but not equal to the secondary bus), the transaction is claimed and passed through as a Type 1 Configuration Read.
1011b	Configuration Write	Type 0 Configuration Write, same as Configuration Read (P_CBE[3:0]#=1010b). Type 1 Configuration Write (not Special Cycle request), same as Configuration Read (P_CBE[3:0]#=1010b). Configuration Write as Special Cycle request (Device = 1Fh, Function = 7h). If the target bus is the bridge's secondary bus, the transaction is claimed and passed through as a Special Cycle. If the target bus is a subordinate bus that exists behind the bridge (but not equal to the secondary bus), the transaction is claimed and passed through unchanged as a Type 1 Configuration Write.

Table 15-1. Primary Interface Supported Commands (Continued)

P_CBE[3:0]#	PCI Command	Support
1100b	Memory Read Multiple	Treated as a Memory Read (P_CBE[3:0]#=0110b).
1101b	DAC	***Not Supported.***
1110b	Memory Read Line	Treated as a Memory Read (P_CBE[3:0]#=0110b).
1111b	Memory Write and Invalidate	Treated as a Memory Write (P_CBE[3:0]#=0111b).

15.2 SECONDARY INTERFACE COMMAND SET

Table 15-2 delineates the PCI 6150 secondary interface PCI command set.

Table 15-2. Secondary Interface Supported Commands

S_CBE[3:0]#	PCI Command	Support
0000b	Interrupt Acknowledge	***Not Supported.***
0001b	Special Cycle	
0010b	I/O Read	If the address is within pass-through I/O range, the transaction is claimed and passed through. If the address points to an I/O-mapped internal bridge register, the transaction is claimed. Otherwise, the transaction is ignored.
0011b	I/O Write	Same as I/O Read (S_CBE[3:0]#=0010b).
0100b — 0101b	***Reserved***	—
0110b	Memory Read	If the address is within pass-through Memory range, the transaction is claimed and passed through. If the address points to a memory-mapped internal bridge register, the transaction is claimed. Otherwise, the transaction is ignored.
0111b	Memory Write	Same as Memory Read (S_CBE[3:0]#=0110b).
1000b — 1001b	***Reserved***	***Not Supported.***
1010b	Configuration Read	Upstream Configuration Read cycles. ***Not Supported.***
1011b	Configuration Write	Type 0 Configuration Write. ***Not Supported.*** Type 1 Configuration Write (not a Special Cycle request). ***Not Supported.*** Configuration Write as Special Cycle request (Device = 1Fh, Function = 7h). If the target bus is the bridge's primary bus, the transaction is claimed and passed through as a Special Cycle. If the target bus is neither the primary bus nor in the range of buses defined by the bridge's secondary and subordinate bus registers, the transaction is claimed and passed through unchanged as a Type 1 Configuration Write. If the target bus is not the bridge's primary bus, but is within the range of buses defined by the bridge's secondary and subordinate bus registers, the transaction is ignored.
1100b	Memory Read Multiple	Treated as a Memory Read (S_CBE[3:0]#=0110b).
1101b	DAC	***Not Supported.***
1110b	Memory Read Line	Treated as a Memory Read (S_CBE[3:0]#=0110b).
1111b	Memory Write and Invalidate	Treated as a Memory Write (S_CBE[3:0]#=0111b).

16 BRIDGE BEHAVIOR

This section presents various bridge behavior scenarios that occur when the target responds to a cycle generated by the PCI 6150, on behalf of the initiating master.

16.1 BRIDGE ACTIONS FOR VARIOUS CYCLE TYPES

A PCI cycle is initiated by FRAME# assertion. In a bridge, there are several possibilities for this to occur. Table 16-1 summarizes these possibilities, and delineates the PCI 6150 action for various cycle types.

After the PCI cycle is initiated, a target then has up to three clocks to respond before subtractive decoding, or four clocks before a Master Abort, is initiated. If the target detects an address hit, it asserts DEVSEL# in the cycle corresponding to the Configuration Status register DEVSEL# Timing bits (PCISR[10:9]; PCI:06h or PCISSR[10:9]; PCI:1Eh).

PCI cycle termination can occur in a number of ways. Normal termination begins by the initiator (master) de-asserting FRAME#, with IRDY# being asserted (or remaining asserted) on the same cycle. The cycle completes when TRDY# and IRDY# are simultaneously asserted. The target should de-assert TRDY# for one cycle following final assertion (sustained three-state signal).

Table 16-1. Bridge Actions for Various Cycle Types

Initiator	Target	PCI 6150 Response
Master on primary port	Target on the same primary port	Does not respond. This situation is detected by decoding the address and monitoring P_DEVSEL# for other fast and medium-speed devices on the primary port.
	Target on secondary port	Asserts P_DEVSEL# and normally terminates the cycle if posted; otherwise, returns with a Retry. Next, passes the cycle to the appropriate port. When the cycle completes on the target port, the PCI 6150 waits for the initiator to repeat the same cycle and end with normal termination.
	Target not on primary nor secondary port	Does not respond and the cycle terminates as a Master Abort.
Master on secondary port	Target on the same secondary port	Does not respond.
	Target on primary or other secondary port	Asserts S_DEVSEL# and normally terminates the cycle if posted; otherwise, returns with a Retry. Next, passes the cycle to the appropriate port. When the cycle completes on the target port, the PCI 6150 waits for the initiator to repeat the same cycle and end with normal termination.
	Target not on primary nor other secondary port	Does not respond.

16.2 ABNORMAL TERMINATION (MASTER ABORT, INITIATED BY BRIDGE MASTER)

A Master Abort indicates that the PCI 6150, operating as a master, receives no response from a target (*that is*, no target asserts P_DEVSEL# or S_DEVSEL#). The bridge de-asserts FRAME#, then de-asserts IRDY#.

16.3 PARITY AND ERROR REPORTING

Parity must be checked for all addresses and Write data. Parity is defined on the P_PAR and S_PAR signals. Parity should be even [*that is*, an even number of ones (1)] across AD[31:0], CBE[3:0]#, and PAR. Parity information on PAR is valid the cycle after AD[31:0] and CBE[3:0]#, are valid.

For all Address phases, if a Parity error is detected, the error is reported on the P_SERR# signal by asserting P_SERR# for one cycle, then placing two cycles into a high-impedance state after the bad address. P_SERR# can be asserted only if the Command register P_SERR# and Parity Error Response bits are both set to 1 (PCICR[8, 6]=11b; PCI:04h, respectively). For Write Data phases, a Parity error is reported by asserting P_PERR# two cycles after the Data phase and remains asserted for one cycle when PCICR[8]=1. The target reports any type of Data Parity errors during Write cycles, while the master reports Data Parity errors during Read cycles.

Address Parity error detection causes the PCI bridge target to not claim the bus (P_DEVSEL# remains inactive). The cycle then terminates with a Master Abort. When the bridge is operating as master, a Data Parity error during a Read cycle results in the bridge master initiating a Master Abort.

17 PCI FLOW-THROUGH OPTIMIZATION

This section describes Flow-Through optimization, including precautions when using non-optimized PCI master devices, Posted Write and Delayed Read Flow Through, Read cycle optimization, and Read Prefetch boundaries.

17.1 OVERVIEW

The PCI 6150 operates in Flow-Through mode when data from the same transaction is simultaneously transferred on both sides of the bridge (*that is*, data on one side of the bridge "flows through" to the other side of the bridge). The PCI 6150 has several options to optimize PCI transfers after Flow-Though mode is achieved by way of the bridge.

The purpose of Flow-Through mode is to improve PCI Bus utilization and efficiency. If Data transfers on one side of the bridge are broken into several transactions on the other side of the bridge, poor bus efficiency results. By using Flow-Through mode, the PCI 6150 improves bus efficiency for Posted writes, Delayed Reads, and reads to prefetchable spaces.

17.2 PRECAUTIONS WHEN USING NON-OPTIMIZED PCI MASTER DEVICES

The PCI 6150 is capable of high-performance prefetching. However, some PCI masters may be unable to prefetch a large amount of data. This may be due to a small internal buffer size or other limiting factors. *For example*, if data is being read from a register or FIFO-based architecture, valuable data may be lost if the host prematurely terminates a Prefetch cycle (ideally such spaces would not be listed as prefetchable). Under these circumstances the default values for prefetching may be overly aggressive and affect overall performance. In this case, tune default prefetching by reprogramming the Prefetch registers, as listed in Table 17-1. (Refer to Section 6, "Registers," for a detailed description of these registers.)

The serial EEPROM can also be used to program the Configuration space upon reset.

17.3 POSTED WRITE FLOW THROUGH

During Flow Through of Posted Write cycles, if there is only one Data transfer pending in the Internal Post Memory Write queue, the PCI 6150 can be programmed to wait for a specified number of clocks before Disconnecting. The PCI 6150 de-asserts IRDY# on the target side and waits up to seven clocks for additional data from the initiator. If new Write data is received from the initiator during this period, the PCI 6150 re-asserts IRDY# and continues with the Write cycle. If new Write data is not received during this period, the PCI 6150 terminates the cycle to the target with the last data from the queue and later finishes the cycle.

The Flow-Through Control registers for Posted writes are detailed in Section 6, "Registers." (Refer to PFTCR[2:0]; PCI:44h and SFTCR[2:0]; PCI:4Eh.)

17.4 DELAYED READ FLOW THROUGH

For Flow Through of Delayed Read cycles, if the Internal Read Queue is almost full, the PCI 6150 can be programmed to insert wait states to delay Read data from the target for a specified number of clocks before Disconnecting. During this time, the PCI 6150 de-asserts IRDY# on the target bus and waits up to seven clocks. If additional space becomes available in the Internal Read queue before the end of the IRDY# inactive period, the PCI 6150 re-asserts IRDY# and proceeds with the next Read Data phase. If no additional space becomes available in the Internal Read queue, the current Data phase becomes the last (IRDY# is asserted) and the cycle Disconnects at the end of the Data phase.

The Flow-Through Control registers for Delayed Reads are detailed in Section 6, "Registers." (Refer to PFTCR[6:4]; PCI:44h and SFTCR[6:4]; PCI:4Eh.)

Table 17-1. Reprogramming Prefetch Registers

Configuration Space Register	Data
Primary Initial Prefetch Count (PITLPCNT; PCI:48h)	Same value as Cache Line Size register (PCICLSR; PCI:0Ch). ***Note:*** *Most PCs set this value to 08h.*
Secondary Initial Prefetch Count (SITLPCNT; PCI:49h)	
Primary Incremental Prefetch Count (PINCPCNT; PCI:4Ah)	0h
Secondary Incremental Prefetch Count (SINCPCNT; PCI:4Bh)	
Primary Maximum Prefetch Count (PMAXPCNT; PCI:4Ch)	
Secondary Maximum Prefetch Count (SMAXPCNT; PCI:4Dh)	

17.5 READ CYCLE OPTIMIZATION

The main function of Read Cycle optimization is to increase the probability of Flow Through occurring during Read accesses to Prefetchable Memory regions. To improve the probability of Flow Through, the amount of data to be prefetched must be correctly configured.

If the PCI 6150 prefetches insufficient data, Flow Through does not occur because prefetching on the target side completes before the Initiator Retries the Read access. Under these circumstances, the Read cycles become divided into multiple cycles.

If the PCI 6150 prefetches excessive data and the internal FIFOs fill, the PCI 6150 must wait for the initiator to Retry the previous Read cycle and then flush the unclaimed data before queuing subsequent cycles.

The initial count is normally equivalent to the cache-line size. This assumes that a master usually requires at least one cache line of data. The incremental count is used only when the PCI 6150 does not detect Flow Through for the current cycle being prefetched during the Initial Prefetch Count. The PCI 6150 continues prefetching in increments until it reaches the maximum count, then Disconnects the cycle.

- For Read prefetching, the PCI 6150 implements several registers that control the amount of data prefetched on the primary and secondary PCI Buses. The Prefetch registers listed in Table 17-1 can be used to optimize PCI 6150 performance during Read cycles.

The PCI 6150 prefetches until Flow Through occurs or prefetching must stop, based on the following conditions:

Prefetch continues while:

$$(IPMC + IPC + IPC + \ldots + IPC) < MPC$$

where:

IPMC = Initial Prefetch Maximum Count

IPC = Incremental Prefetch Count, < ½ MPC

MPC = Maximum Prefetch Count

If the Prefetch Count did not reach MPC and Flow Through was achieved, the PCI 6150 continues prefetching until the requesting master terminates the Prefetch request. Otherwise, when MPC is reached, the PCI 6150 stops prefetching data.

Incremental Prefetch can be disabled by setting IPC ≥ MPC.

17.5.1 Primary and Secondary Initial Prefetch Count

Assuming that there is sufficient space in the internal FIFO, the Primary and Secondary Initial Prefetch Count registers (PITLPCNT; PCI:48h and SITLPCNT; PCI:49h, respectively) control the amount of data initially prefetched by the PCI 6150 on the primary or secondary bus during reads to the Prefetchable Memory region. If Flow Through is achieved during this initial prefetch, the PCI 6150 continues prefetching beyond this count.

17.5.2 Primary and Secondary Incremental Prefetch Count

The Primary and Secondary Incremental Prefetch Count registers (PINCPCNT; PCI:4Ah and SINCPCNT; PCI:4Bh, respectively) control the amount of prefetching after the initial prefetch. If Flow Through is not achieved during the initial prefetch, the PCI 6150 attempts to prefetch further data, until the FIFO fills, or until the Maximum Prefetch Count is reached. Each subsequent prefetch is equal to the Incremental Prefetch Count.

17.5.3 Primary and Secondary Maximum Prefetch Count

The Primary and Secondary Maximum Prefetch Count registers (PMAXPCNT; PCI:4Ch and SMAXPCNT; PCI:4Dh, respectively) limit the amount of prefetched data for a single entry available in the internal FIFO at any time. During Read Prefetch cycles, the PCI 6150 Disconnects the cycle if the data count in the FIFO for the current cycle reaches this value, and Flow Through has not been achieved.

17.6 READ PREFETCH BOUNDARIES

For Memory Read and Memory Read Line commands, the PCI 6150 prefetches from the starting address up to an address with an offset that is a multiple of the Initial Prefetch Count. *For example*, if the starting address is 10h and the Initial Prefetch Count equals 20h, the PCI 6150 prefetches only a 10h (20h to 10h) count. After this, the PCI 6150 begins incremental prefetch until the Maximum Prefetch Count is reached, or Flow Through is achieved. The exception to this is in the case of a 64-bit request and six or fewer Dwords from the boundary, or a 32-bit request and four or fewer Dwords from the boundary, in which the PCI 6150 does not activate Incremental Prefetch.

For Memory Read Multiple commands, if the starting address is not 0, the PCI 6150 first prefetches from the starting address up to the address with an offset equal to that of the Initial Prefetch Count. After this, the PCI 6150 prefetches one additional Initial Prefetch Count. *For example*, if the starting address is 10h and the Initial Prefetch Count equals 20h, the PCI 6150 first prefetches a 10h (20h to 10h) count, then continues to prefetch another 20h count. Subsequent to this, Incremental Prefetch is invoked until the Maximum Prefetch Count is reached or Flow Through is achieved.

18 POWER MANAGEMENT

This section describes the Power Management feature.

18.1 OVERVIEW

The PCI 6150 incorporates functionality that meets the requirements of *PCI Power Mgmt. r1.1*. These features include:

- PCI Power Management registers, using the Enhanced Capabilities Port (ECP) address mechanism
- Support for D_0, D_{3cold}, and D_{3hot} power management states
- Support for D_0, D_1, D_2, D_{3cold}, and D_{3hot} power management states for devices behind the bridge
- Support for B_2 secondary bus power state when in the D_{3hot} power management state

18.2 POWER MANAGEMENT TRANSITIONS

Table 18-1 delineates the states and related actions the PCI 6150 performs during Power Management transitions. (No other transactions are allowed.)

PME# signals are routed from downstream devices around PCI-to-PCI bridges. PME# signals do ***not*** pass through PCI-to-PCI bridges.

Table 18-1. States and Related Actions during Power Management Transitions

Current State	Next State	Action
D_0	D_1	Unimplemented power state. The PCI 6150 ignores the write to the Power State bits (power state remains at D_0, PMCSR[1:0]=00b; PCI:E0h).
D_0	D_2	
D_0	D_{3hot}	If enabled by the BPCC_EN pin, the PCI 6150 disables the secondary clocks and drives them low.
D_0	D_{3cold}	Power is removed from the PCI 6150. A power-up reset must be performed to bring the PCI 6150 to D_0.
D_{3cold}	D_0	Power-up reset. The PCI 6150 performs the standard power-up reset functions.
D_{3hot}	D_0	The PCI 6150 enables secondary clock outputs and performs an internal chip reset. S_RSTOUT# is ***not*** asserted. All registers are returned to the reset values and buffers are cleared.
D_{3hot}	D_{3cold}	Power is removed from the PCI 6150. A power-up reset must be performed to bring the PCI 6150 to D_0.

19 HOT SWAP

This section describes the Hot Swap feature and its use.

19.1 OVERVIEW

The PCI 6150 incorporates functionality that meets *PICMG 2.1 R2.0* requirements with High-Availability Programming Interface level 1 (PI=1). The CompactPCI Hot Swap register block is located at PCI Configuration offset E4h. Refer to *PICMG 2.1 R2.0* for detailed implementation guidelines. The Hot Insertion Power-Up sequence recommendation is illustrated in Figure 19-1.

Notes: *To use the Hot Swap function, EJECT_EN# and GPIO3FN# must be connected to 0. (Refer to Table 19-1.)*

If the Hot Swap function is not used, pull GPIO3FN# high or GPIO3 low to disable the function.

19.2 LED ON/OFF (PI=1)

For PI=1 support, upon RSTIN# assertion, the PCI 6150 turns ON the LED. After RSTIN# de-assertion, the LED remains ON until the eject switch (handle) is closed, then the PCI 6150 turns OFF the LED.

Table 19-1. EJECT_EN# and GPIO3FN# Settings for Enabling Hot Swap Capability

EJECT_EN#	GPIO3FN#	Hot Swap	Eject Input
0	0	Enabled	GPIO3
Don't Care	1	Disabled	—

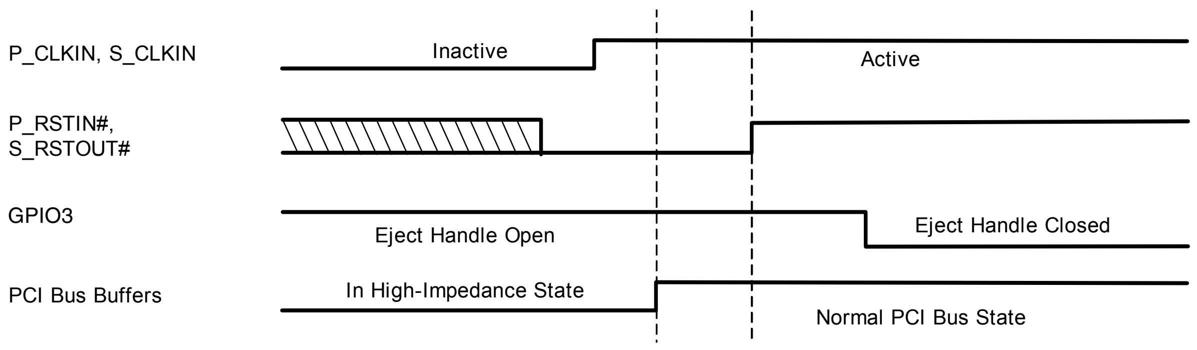


Figure 19-1. Hot Insertion Power-Up Sequence Recommendation

19.3 HOT SWAP SIGNALS

The PCI 6150 uses the following Hot Swap-related pins:

- **EJECT_EN#**—Ejector Pin Use Enable. Used to enable the GPIO3 pin as EJECT input. If this pin is 1, GPIO3 functions as a GPIO pin. GPIO3 only functions as EJECT input when both GPIO3FN# and EJECT_EN# are tied low, which also enables Hot Swap capability.
- **ENUM#**—Indicates an open-drain bused signal asserted when an adapter was inserted or is ready to be extracted from a PCI slot. Asserted through the Hot Swap registers (HS_CNTL; PCI:E4h, HS_CSR; PCI:E6h, and HS_NEXT; PCI:E5h).
- **GPIO3FN#**—When GPIO3FN# is tied high, GPIO3 functions as a GPIO pin regardless of the EJECT_EN# pin state. GPIO3 functions as Ejector input only when both GPIO3FN# and EJECT_EN# are tied low. To enable Hot Swap capability, both the GPIO3FN# and EJECT_EN# inputs must be low.
- **PIN_LED/EJECT**—Active high signal that allows other circuits to drive the Blue Hot Swap LED. Turns ON LED if RSTIN# is asserted, or when the LOO bit is set (HS_CSR[3]=1; PCI:E6h) and RSTIN# is de-asserted.

19.4 HOT SWAP REGISTER CONTROL AND STATUS

The PCI 6150 Hot Swap Control/Status register (HS_CSR) is located at PCI offset E6h.

19.5 DEVICE HIDING

The PCI 6150 implements Device Hiding to eliminate mid-transaction extractions. This invokes Device Hiding by hardware from the Hot Swap port after RSTIN# becomes inactive and the ejector handle remains unlocked.

Software quiesces the PCI 6150 when Device Hiding is invoked. The current transaction is completed as early as possible. The PCI 6150 does not initiate a transaction as a master, respond as a target to I/O transactions, nor signal interrupts.

When Device Hiding is invoked, the PCI 6150 terminates the current Configuration transaction by signaling a Disconnect. After the current transaction completes (is Disconnected), the PCI 6150 does not respond as a target to any subsequent transactions until Device Hiding is canceled.

If not participating in a transaction when Device Hiding is invoked, the PCI 6150 does not respond as a target to subsequent transactions until Device Hiding is canceled.

Device Hiding is canceled when the handle switch is relocked.

20 VPD

This section describes the VPD feature.

The PCI 6150 contains the Vital Product Data (VPD) registers, as specified in *PCI r2.3*. VPD information is stored in the serial EEPROM device, along with Autoload information.

The PCI 6150 provides storage of 192 bytes of VPD data in the serial EEPROM device.

The VPD register block is located at offsets E8h to EFh in PCI Configuration space. (Refer to Section 6.1.2.15, “VPD Capability.”) VPD also uses the Enhanced Capabilities Port Address mechanism.

21 TESTABILITY/DEBUG

This section describes the JTAG interface for use in testing and debugging the PCI 6150.

21.1 JTAG INTERFACE

The PCI 6150 provides a JTAG Boundary Scan interface, which can be utilized to debug a pin's board connectivity.

21.1.1 IEEE 1149.1 Test Access Port

The IEEE 1149.1 Test Access Port (TAP), commonly called the JTAG (Joint Test Action Group) debug port, is an architectural standard described in IEEE Standard 1149.1-1990, *IEEE Standard Test Access Port and Boundary-Scan Architecture.* The standard describes a method for accessing internal chip facilities using a four- or five-signal interface.

The JTAG debug port, originally designed to support scan-based board testing, is enhanced to support the attachment of debug tools. The enhancements, which comply with IEEE Standard 1149.1-1990 specifications for vendor-specific extensions, are compatible with standard JTAG hardware for boundary-scan system testing.

- **JTAG Signals**—JTAG debug port implements the four required JTAG signals—TCK, TDI, TDO, TMS—and the optional TRST# signal. (Refer to Table 3-10, "JTAG Pins," on page 3-15 for signal descriptions.)
- **JTAG Clock Requirements**—TCK signal frequency can range from DC to 10 MHz.
- **JTAG Reset Requirements**—JTAG debug port logic and system simultaneously reset. The two methods for placing the PCI 6150 JTAG TAP controller into the Test-Logic-Reset state are as follows:
  - Upon receiving TRST#, the JTAG TAP controller returns to the Test-Logic Reset state
  - Hold the PCI 6150 TMS pin high while transitioning the PCI 6150 TCK pin five times

21.1.2 JTAG Instructions

The JTAG debug port provides the standard **EXTEST**, **SAMPLE/PRELOAD**, and **BYPASS** instructions. Invalid instructions behave as **BYPASS** instructions.

Table 21-1 lists the JTAG instructions, along with their input codes.

Table 21-1. JTAG Instructions (IEEE Standard 1149.1-1990)

Instruction	Input Code
EXTEST	00000b
SAMPLE/PRELOAD	00001b
BYPASS	11111b

21.1.3 JTAG Boundary Scan

Boundary Scan Description Language (BSDL), IEEE 1149.1b-1994, is a supplement to IEEE Standard 1149.1-1990 and IEEE 1149.1a-1993, *IEEE Standard Test Access Port and Boundary-Scan Architecture*. BSDL, a subset of the IEEE 1076-1993 Standard VHSIC Hardware Description Language (VHDL), allows a rigorous description of testability features in components that comply with the standard. Automated test pattern generation tools use BDSL for package interconnect tests and Electronic Design Automation (EDA) tools for synthesized test logic and verification. BSDL supports robust extensions that can be used for internal test generation and to write software for hardware debug and diagnostics.

The primary components of BSDL include the logical port description, physical pin map, instruction set, and Boundary register description.

The logical port description assigns symbolic names to the PCI 6150 pins. Each pin has a logical type of in, out, in out, buffer, or linkage that defines the logical signal flow direction.

The physical pin map correlates the PCI 6150 logical ports to the physical pins of a specific package. A BSDL description can have several physical pin maps; each map is provided a unique name.

Instruction set statements describe the bit patterns that must be shifted into the Instruction register to place the PCI 6150 in the various Test modes defined by the standard. Instruction set statements also support instruction descriptions unique to the PCI 6150.

The Boundary register description lists each of its cells or shift stages. Each cell has a unique number—the cell numbered 0 is the closest to the Test Data Out (TDO) pin and the cell with the highest number is closest to the Test Data In (TDI) pin. Each cell contains additional information, including:

- Cell type
- Logical port associated with the cell
- Logical function of the cell
- Safe value
- Control cell number
- Disable value
- Result value

21.1.4 JTAG Reset Input TRST#

The TRST# input pin is the asynchronous JTAG logic reset. TRST# assertion causes the PCI 6150 TAP controller to initialize. In addition, when the TAP controller is initialized, it selects the PCI 6150 normal logic path (core-to-I/O). Consider the following when implementing the asynchronous JTAG logic reset on a board:

- If JTAG functionality is required, one of the following should be considered:
    - Use the TRST# input signal low-to-high transition once.
    - Hold the PCI 6150 TMS pin high while transitioning the PCI 6150 TCK pin five times.
- If JTAG functionality is not required, the TRST# signal must be directly connected to ground.

Note: *IEEE Standard 1149.1-1990 requires pull-up resistors on the TDI, TMS, and TRST# pins. To remain PCI r2.3-compliant, no internal pull-up resistors are provided on JTAG pins in the PCI 6150; therefore, the pull-up resistors must be externally added to the PCI 6150 when implementing JTAG.*

22 MECHANICAL SPECS

This section provides the PCI 6150 mechanical dimensions and pinout. The PCI 6150 is available as an industry standard 208-pin PQFP or 256-pin PBGA package.

22.1 208-PIN PQFP

22.1.1 Mechanical Dimensions—208-Pin PQFP

Figure 22-1 illustrates the mechanical dimensions of the 208-pin PQFP package. Table 22-1 lists the mechanical dimensions, in millimeters, unless specified otherwise.

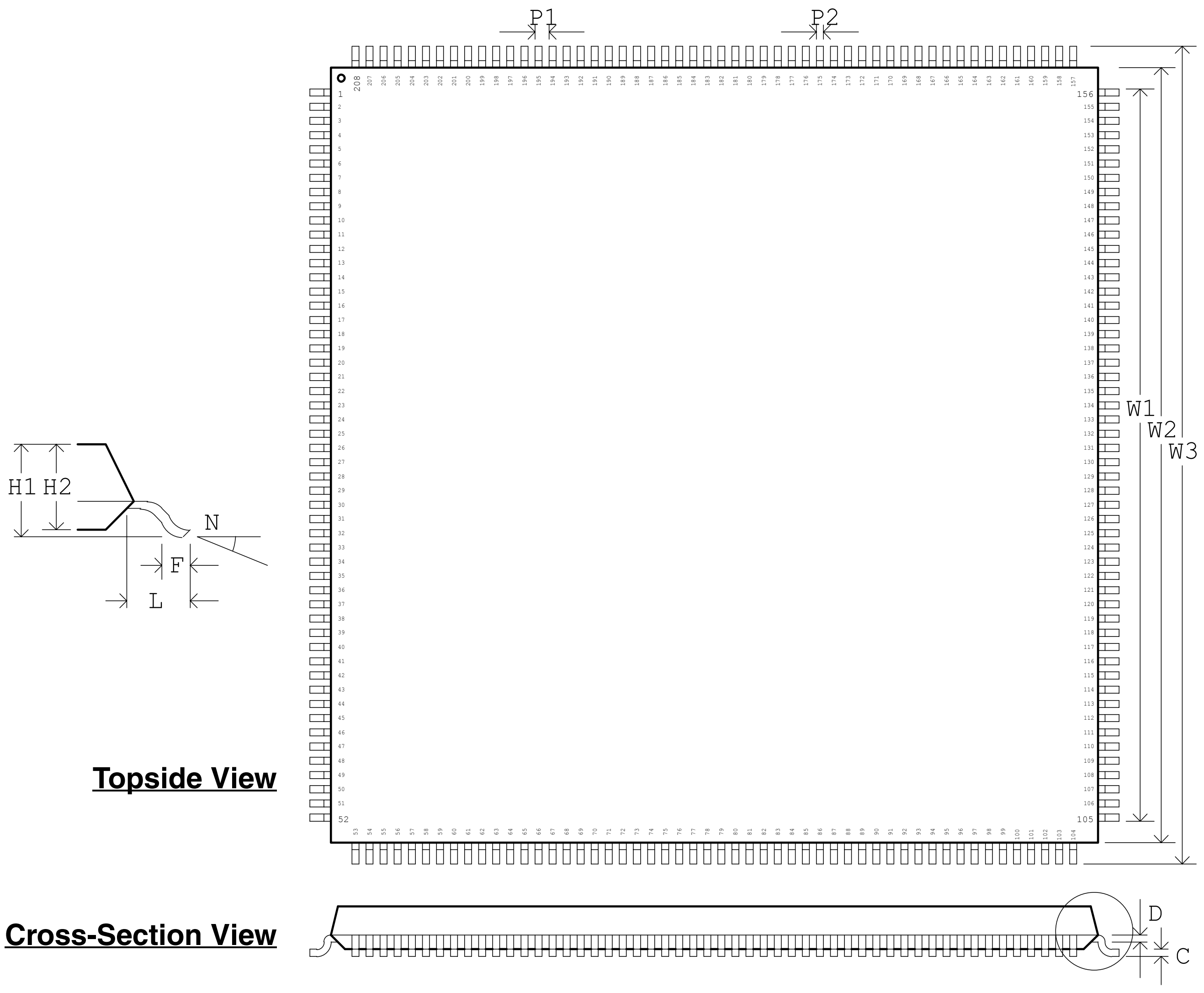


Figure 22-1. PCI 6150 Mechanical Dimensions—208-Pin PQFP

Table 22-1. PCI 6150 Mechanical Dimensions for Figure 22-1 Symbols (in Millimeters)—208-Pin PQFP

Symbol	Dimension	Minimum	Nominal	Maximum
W1	—	—	—	—
W2	Package width (length)	27.95	28.00	28.05
W3	Package overall width (length)	—	30.60	—
P1	Lead pitch	—	0.50	—
P2	Lead width	0.17	—	0.27
C	Lead thickness	0.09	—	0.20
D	—	—	0.13	—
H1	Package overall height	—	4.20	—
H2	Package thickness	3.17	—	3.95
L	Lead length	—	1.30	—
F	Foot length	0.45	0.60	0.75
N	Foot angle	0	—	7

22.1.2 Physical Layout with Pinout—208-Pin PQFP

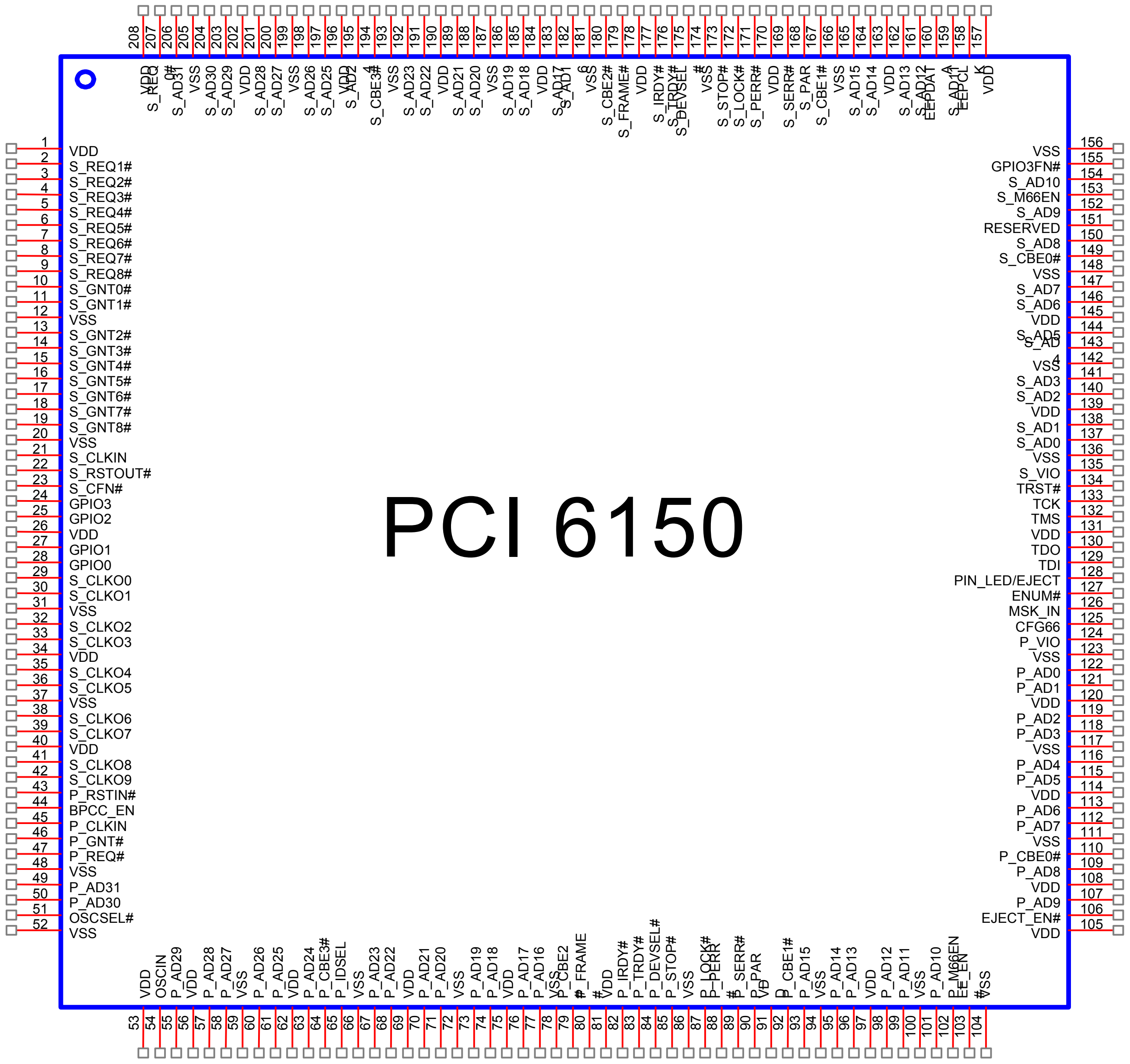


Figure 22-2. PCI 6150 Top View—208-Pin PQFP

22.2 256-PIN PBGA

22.2.1 Mechanical Dimensions—256-Pin PBGA

Figure 22-3 illustrates the mechanical dimensions of the 256-pin PBGA package.

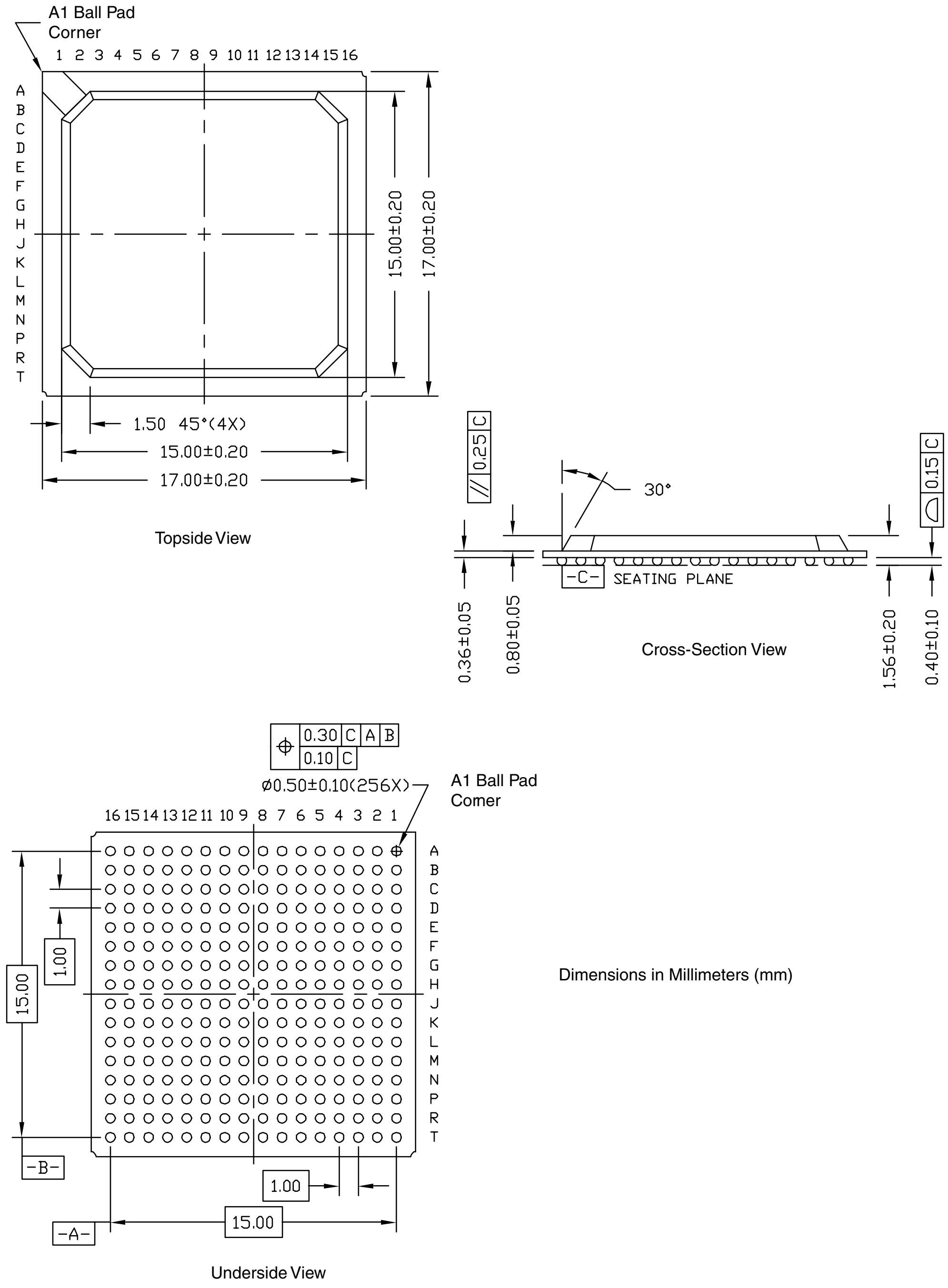


Figure 22-3. PCI 6150 Mechanical Dimensions—256-Pin PBGA

22.2.2 Physical Layout with Pinout—256-Pin PBGA

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
A	VSS	S_REQ2#	VDD	S_AD31	S_AD28	S_AD25	S_AD22	S_AD19	S_AD17	S_FRAME#	S_DEVSEL#	S_PERR#	S_PAR	S_AD13	S_AD11	VSS	A
B	VSS	VSS	S_REQ1#	S_REQ0#	S_AD27	S_CBE3#	S_AD21	S_AD18	S_CBE2#	S_IRDY#	S_STOP#	S_CBE1#	S_AD12	GPIO3FN#	VSS	S_AD10	B
C	S_REQ5#	S_REQ4#	VSS	VDD	S_AD29	S_AD24	S_AD23	S_AD20	S_AD16	S_TRDY#	S_LOCK#	S_AD15	VSS	EEPCLK	EE_EN#	S_AD8	C
D	S_GNT0#	S_REQ6#	S_REQ3#	VSS	S_AD30	S_AD26	VDD	VDD	VDD	VDD	S_SERR#	S_AD14	VSS	EEPDATA	S_M66EN	S_AD6	D
E	S_GNT3#	S_GNT2#	S_REQ7#	S_REQ8#	VSS	VDD	VDD	VDD	VDD	VDD	VDD	VSS	S_AD9	S_AD7	S_CBE0#	S_AD4	E
F	S_GNT7#	S_GNT6#	S_GNT1#	S_GNT4#	VDD	VSS	VSS	VSS	VSS	VSS	VSS	VDD	S_AD5	S_AD3	S_AD2	S_AD1	F
G	S_GNT8#	VSS	S_GNT5#	VDD	VDD	VSS	VSS	VSS	VSS	VSS	VSS	VDD	VDD	S_VIO	TRST#	S_AD0	G
H	S_RSTOUT#	S_CFN#	S_CLKIN	VDD	VDD	VSS	VSS	VSS	VSS	VSS	VSS	VDD	VDD	TMS	TCK	TDO	H
J	GPIO1	GPIO2	GPIO3	VDD	VDD	VSS	VSS	VSS	VSS	VSS	VSS	VDD	VDD	ENUM#	TDI	PIN_LED/ EJECT	J
K	GPIO0	S_CLKO0	S_CLKO1	VDD	VDD	VSS	VSS	VSS	VSS	VSS	VSS	VDD	VDD	P_VIO	OSCIN	OSCSEL#	K
L	S_CLKO2	S_CLKO3	S_CLKO5	S_CLKO6	VDD	VSS	VSS	VSS	VSS	VSS	VSS	VDD	P_AD4	P_AD2	P_AD1	P_AD0	L
M	S_CLKO4	S_CLKO8	S_CLKO9	P_CLKIN	VSS	VDD	VDD	VDD	VDD	VDD	VDD	VSS	P_AD6	P_AD7	P_AD5	P_AD3	M
N	S_CLKO7	BPCC_EN	P_AD31	VSS	P_AD28	P_AD25	VDD	VDD	VDD	VDD	P_PAR	P_AD11	VSS	VSS	P_AD8	P_CBE0#	N
P	P_RSTIN#	P_REQ#	VSS	VSS	P_AD27	P_IDSEL	P_AD22	P_AD18	P_FRAME#	P_DEVSEL#	P_SERR#	P_AD14	VDD	VSS	VDD	P_AD9	P
R	P_GNT#	VSS	VDD	VSS	P_AD24	P_CBE3#	P_AD20	P_AD17	P_CBE2#	P_TRDY#	P_LOCK#	P_AD15	P_AD12	P_M66EN	VSS	EJECT_EN#	R
T	VSS	P_AD30	VDD	P_AD29	P_AD26	P_AD23	P_AD21	P_AD19	P_AD16	P_IRDY#	P_STOP#	P_PERR#	P_CBE1#	P_AD13	P_AD10	VSS	T
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	

Figure 22-4. PCI 6150 Top View—256-Pin PBGA

23 ELECTRICAL SPECS

This section presents the PCI 6150 electrical specifications.

23.1 GENERAL ELECTRICAL SPECIFICATIONS

The ratings provided in this subsection are those above which the useful life of the PCI 6150 may be impaired.

Table 23-1 lists the PCI 6150 maximum ratings. Table 23-2 lists the PCI 6150 functional operating range. Table 23-3 lists the PCI 6150 DC electrical characteristics.

Caution:* *Stresses greater than the maximums listed in Table 23-1 cause permanent damage to the PCI 6150. This is a stress rating only and functional operation of the PCI 6150 at or above those indicated in the operational sections of this data book is not implied. Exposure to absolute maximum rating conditions for extended periods of time may affect reliability.

Note: *The power consumption for V_{DD} is dependent on bus frequency, data traffic, and device loading.*

Table 23-1. Maximum Ratings

Parameter	Minimum	Maximum
Storage Temperature Range	-55 °C	+125 °C
Junction Temperature	—	+125 °C
V_{DD} Supply Voltage	—	3.9V
Maximum Voltage to Signal Pins	—	5.5V
Maximum Power	—	1.8W

Table 23-2. Functional Operating Range

Parameter	Minimum	Maximum
V_{DD} Supply Voltage	3.0V	3.6V
Operating Ambient Temperature	0 °C	70 °C

Table 23-3. DC Electrical Characteristics

Symbol	Parameter	Condition	Minimum	Maximum	Unit	Notes
V_{DD}	V_{DD} Supply Voltage	—	3.0	3.6	V	—
V_{IO}	P_V_{IO}, S_V_{IO} Pin Interface I/O Voltage	—	3.0	5.5	V	—
V_{IH}	Input High Voltage	—	0.5 V_{DD}	V_{IO}	V	—
V_{IL}	Input Low Voltage	—	-0.5	+0.3 V_{DD}	V	—
V_{OL}	Output Low Voltage	I_{IOUT} = +1500 µA	—	+0.1 V_{DD}	V	—
V_{OH}	Output High Voltage	I_{IOUT} = -500 µA	0.9 V_{DD}	—	V	—
I_{IL}	Input Leakage Current	$0 < V_{IN} < V_{DD}$	—	±2	µA	—
C_{IN}	Input Pin Capacitance	—	—	7.0	pF	—

23.2 PCI SIGNAL TIMING SPECIFICATION

Figure 23-1 illustrates the PCI 6150 signal timing specifications. Table 23-4 delineates the minimum and maximum values, for the symbols that appear in Figure 23-1.

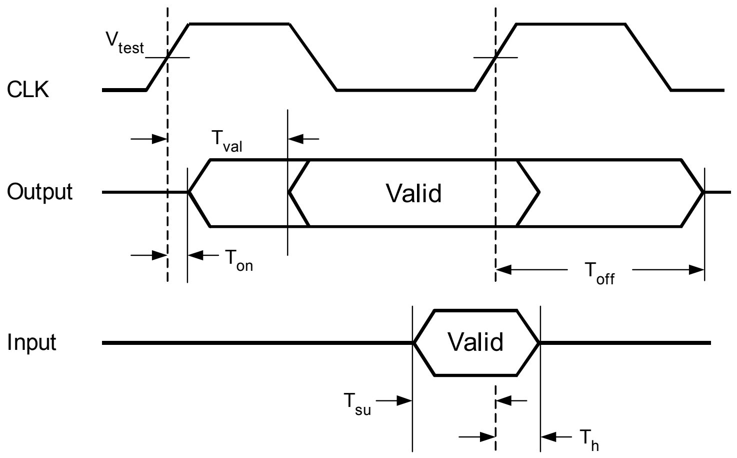


Figure 23-1. PCI Signal Timing Specification

Table 23-4. 66 MHz PCI Signal Timing for Figure 23-1

Symbol	Parameter	Minimum	Maximum	Symbol	Parameter	Minimum	Maximum
T_{val}	CLK to Signal Valid Delay— Bused Signals	2 ns	6 ns	T_{su}	Input Setup Time to CLK— Bused signals	3	—
$T_{val(ptp)}$	CLK to Signal Valid Delay— Point to Point	2 ns	6 ns	$T_{su(ptp)}$	Input Setup Time to CLK— Point to Point	5	—
T_{on}	Float to Active Delay	2 ns	—	T_h	Input Signal Hold Time from CLK	0.5	—
T_{off}	Active to Float Delay	—	14 ns	V_{test}	Voltage Test	—	0.4 V

A USING PCI 6150

Because the PCI 6150 primary and secondary ports are asynchronous to one another, these two independent systems can run at differing frequencies. The secondary bus can be run faster than the primary bus, and vice versa.

The PCI 6150 controls powerful programmable buffers, which can be used to regulate data throughput for multiple PCI masters on the secondary port. The data prefetch size can be programmed to up to 256 bytes.

The host system PCI bus is connected to the PCI 6150 primary port. The secondary PCI port can use a custom-designed External Arbiter or the PCI 6150 Internal Arbiter. To provide clocks to secondary PCI devices and PCI 6150 S_CLKIN, use custom-designed clock generations, PCI 6150 S_CLKO[9:0] outputs (derived out of the primary port PCI clock input), or an external oscillator.

Figure A-1 provides basic optimization design.

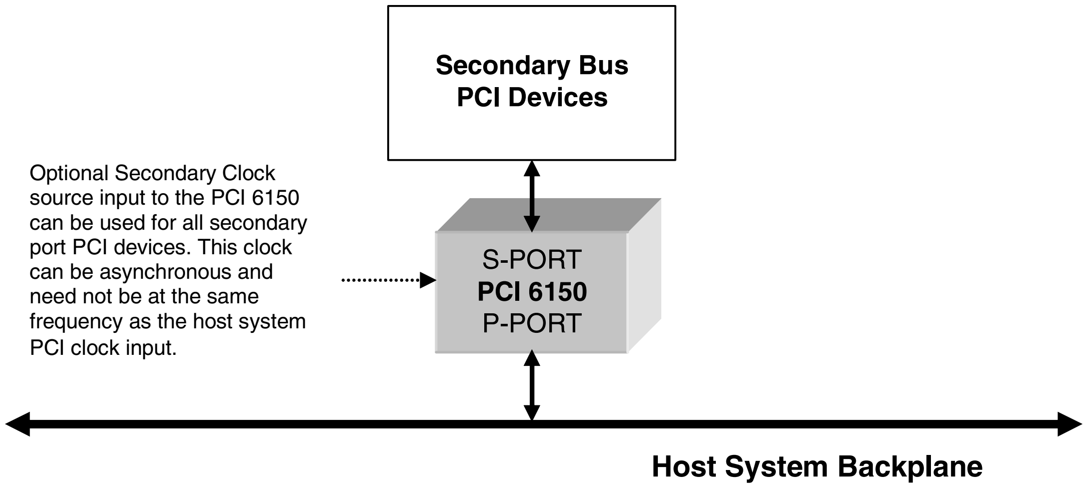

Secondary Bus
PCI Devices
Optional Secondary Clock source input to the PCI 6150 can be used for all secondary port PCI devices. This clock can be asynchronous and need not be at the same frequency as the host system PCI clock input.
S-PORT
PCI 6150
P-PORT
Host System Backplane

Figure A-1. PCI 6150 Basic Optimization Design

B GENERAL INFORMATION

The PLX FastLane™ PCI 6150 32-bit, 66 MHz PCI-to-PCI bridge is designed for high-performance, high-availability applications in Hot Swap, bus expansions, programmable data transfer rate control, frequency conversions from slower-to-faster or faster-to-slower PCI Buses. The PCI 6150 provides sophisticated buffer management and configuration options designed to customize performance optimization.

The PCI 6150 offers the largest data FIFO among all 32-bit PCI-to-PCI bridges in today's market. The PCI 6150 provides the following features and applications:

- *PCI r2.3* compliant
- 3.3V signaling, including 5V input signal tolerance and fast PCI buffers
- Provides 1 KB of buffering (data FIFO) to maximize performance
  - Upstream and downstream Posted Write buffers (256 bytes each)
  - Upstream and downstream Read Data buffers (256 bytes each)
  - Supports up to four simultaneous Posted Write transactions and four simultaneous Delayed transactions in each direction
  - Programmable prefetch of up to 256 bytes for maximum read performance optimization
  - Flow-through zero wait state burst up to 4 KB for large volume data transfer
  - Optional flow-through enable allows for customization
  - Fast back-to-back enable—Read-only supported
- Asynchronous design supports standard 66-to-33 MHz and faster secondary port speed, *such as* 33-to-66 MHz conversion
- Out-of-order Delayed transactions
- Enhanced address decoding
  - 32-bit I/O Address range
  - 32-bit Memory-Mapped I/O Address range
  - ISA Aware mode for legacy support in the first 64 KB of I/O Address range
  - VGA addressing and palette snooping support
- Address Stepping hardcoded to two clocks
- Ten secondary Clock outputs with pin-controlled enable and individual maskable control to nine bus masters on secondary interface support
- External arbiter or programmable arbitration for up to nine bus masters on secondary interface support
- Hot Swap *Ready*
- *PICMG 2.1 R2.0* with PI=1
  - Support for device hiding, eliminating mid-transaction extraction problems
- PCI Mobile Design Guide and Power Management D_{3cold} Wakeup capable with PME# support
- Four GPIO pins with output control and power-up status latch capable
- Serial EEPROM loadable and programmable PCI Read-Only register configurations
  - Serial EEPROM Load modification and recheck
  - VPD support
- IEEE Standard 1149.1-1990 JTAG interface for boundary scan test
- Multiple IDs check all Device and Revision IDs
- Industry-standard 208-pin Plastic Quad Flat Pack (PQFP) or 256-pin (ball) Plastic Ball Grid Array (PBGA) package

B.1 HINT/PLX PART NUMBER CONVERSION

Table B-1. Hint/PLX Part Number Conversion

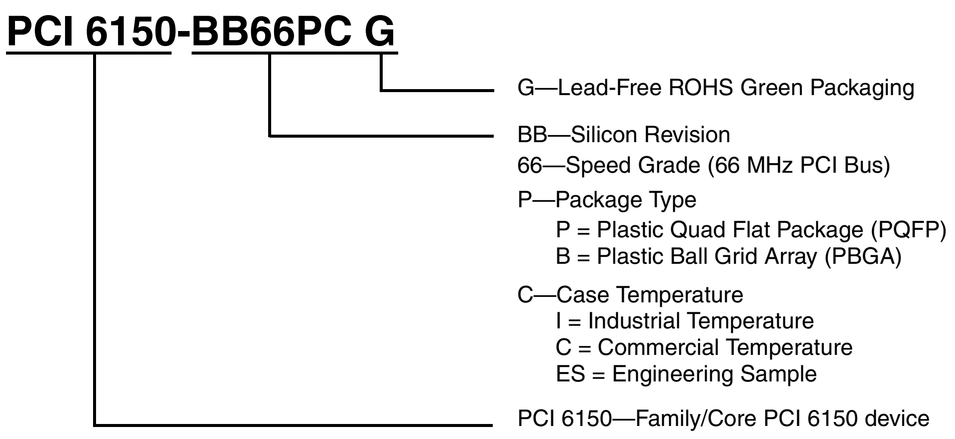
HiNT Part Number	PLX Part Number
HB4	PCI 6150

B.2 PACKAGE ORDERING

The PCI 6150 is available in standard leaded packaging and lead-free ROHS packaging. Ordering information is delineated in Table B-2.

Table B-2. Available Packages

Package Type	Ordering Part Numbers
Standard Leaded PQFP Package	PCI6150-BB66PC
Lead-Free ROHS Green PQFP Packaging	PCI6150-BB66PC G
Standard Leaded PBGA Package	PCI6150-BB66BC
Lead-Free ROHS Green PBGA Packaging	PCI6150-BB66BC G


PCI 6150-BB66PC G
G—Lead-Free ROHS Green Packaging
BB—Silicon Revision
66—Speed Grade (66 MHz PCI Bus)
P—Package Type
P = Plastic Quad Flat Package (PQFP)
B = Plastic Ball Grid Array (PBGA)
C—Case Temperature
I = Industrial Temperature
C = Commercial Temperature
ES = Engineering Sample
PCI 6150—Family/Core PCI 6150 device

B.3 UNITED STATES AND INTERNATIONAL REPRESENTATIVES, AND DISTRIBUTORS

A list of PLX Technology, Inc., representatives and distributors can be found at http://www.plxtech.com.

B.4 TECHNICAL SUPPORT

PLX Technology, Inc., technical support information is listed at http://www.plxtech.com/support/, or call 408 774-9060 or 800 759-3735.

C PCI 6150BB AND PCI 6350AA PIN COMPARISONS AND SIGNAL DIFFERENCES

C.1 PIN ASSIGNMENT COMPARISONS

Table C-1 lists the PQFP pin differences, and Table C-2 lists the PBGA pin differences, between the PCI 6150BB and PCI 6350AA.

Table C-1. PCI 6150BB Versus PCI 6350AA Pin Assignment Comparison—PQFP Package

PQFP Pin Location	PCI 6150BB	PCI 6350AA
51	OSCSEL#	V_{DD}
54	OSCIN	V_{SS}
103	EE_EN#	V_{DD}
106	EJECT_EN#	V_{SS}
124	P_V_{IO}	PME_EN#
127	ENUM#	NC
128	PIN_LED/EJECT	NC
135	S_V_{IO}	NC
151	RESERVED	EE_EN#
155	GPIO3FN#	V_{DD}

Table C-2. PCI 6150BB Versus PCI 6350AA Pin Assignment Comparison—PBGA Package

PBGA Pin Location	PCI 6150BB	PCI 6350AA
B14	GPIO3FN#	NC
G14	S_V_{IO}	NC
J14	ENUM#	NC
J16	PIN_LED/EJECT	NC
K14	P_V_{IO}	NC
K15	OSCIN	MSK_IN
K16	OSCSEL#	CFG66
R16	EJECT_EN#	NC

C.2 PACKAGE SIGNAL DIFFERENCES

Table C-3 lists the signals that exist in one PCI 6150BB or PCI 6350AA package type, but not the other (*that is,* in the PQFP, but not the PBGA, or, in the PBGA, but not the PQFP).

Table C-3. Signal Differences between PCI 6150BB and PCI 6350AA PQFP and PBGA Packages

Signal Name	**PCI 6150BB**		**PCI 6350AA**	
	PQFP	**PBGA**	**PQFP**	**PBGA**
CFG66[1]	Yes	No	Yes	Yes
MSK_IN[2]	Yes	No	Yes	Yes
PME_EN#[3]	N/A	N/A	Yes	No
RESERVED	Yes	No	N/A	N/A

1. *Used only in the PCI 6150BB PQFP and PCI 6350AA PQFP and PBGA packages. In the PCI 6150BB PBGA package, the 66 MHz-Capable bits are hardwired to 1 (PCISR[5]=1; PCI:06h and PCISSR[5]=1; PCI:1Eh) to indicate 66 MHz capability.*
2. *Used only in the PCI 6150BB PQFP and PCI 6350AA PQFP and PBGA packages. If using the PCI 6150BB PBGA package, use software to disable unused Secondary Clock buffers through the SCLKCNTRL; PCI:68h register.*
3. *Used only in the PCI 6350AA PQFP package. In the PCI 6150BB PQFP and PBGA and PCI 6350AA PBGA packages, the Power Management feature is internally bonded as enabled.*

INDEX

Index