TECHNOLOGY

THE I/O INTERCONNECT SOLUTION

PCI 6156 Data Book

PCI 6156 Data Book

Version 1.0

February 2004

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 6156-SIL-DB-P1-1.0

Printed in the USA, February 2004

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 2.2*, December 18, 1998
  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 1.0*, 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

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 r2.1	*PCI Local Bus Specification, Revision 2.1*
PCI r2.2	*PCI Local Bus Specification, Revision 2.2*
P-to-P Bridge r1.1	*PCI to PCI Bridge Architecture Specification, Revision 1.1*
PCI Power Mgmt. r1.0	*PCI Bus Power Management Interface Specification, Revision 1.0*

DATA ASSIGNMENT CONVENTIONS

Data Assignment Conventions

Data Width	PCI 6156 Convention
1 byte (8 bits)	Byte
2 bytes (16 bits)	Word
4 bytes (32 bits)	DWORD/Dword

REVISION HISTORY

Date	Version	Comments
1/04	1.0	Production Release, Silicon Revision DA.

PCI 6156

PCI-to-PCI Bridge

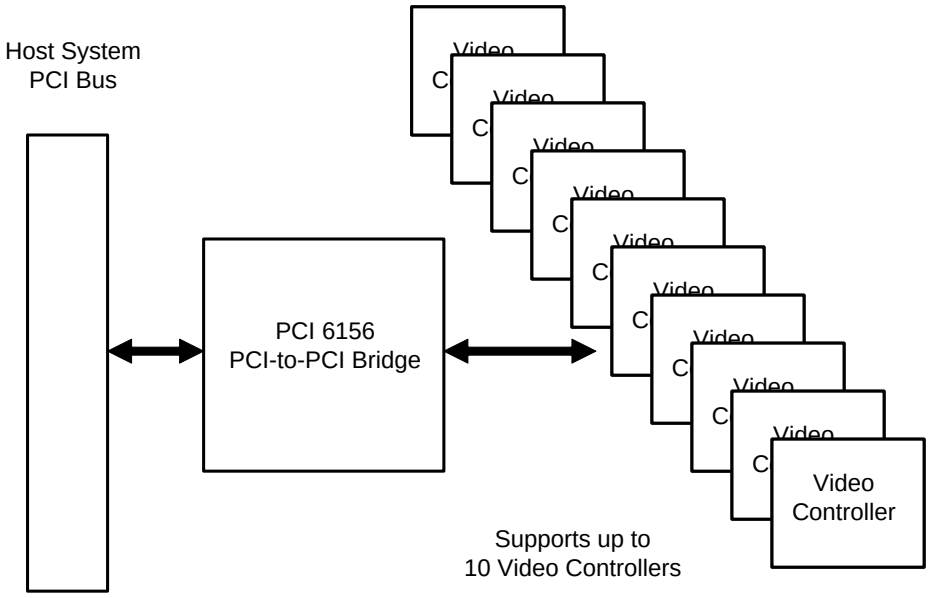
February 2004
Version 1.0

Synchronous 32-Bit, 33 MHz PCI-to-PCI Bridge for Video Capture Applications

FEATURE SUMMARY

The PCI 6156 is built upon the powerful PLX PCI-to-PCI Bridge Architecture. It addresses the needs of the surveillance market by offering a system-level performance advantage. In particular, the PCI 6156 can support up to ten video capture chips by providing ten pairs of PCI request and grant lines. The following is a brief summary of its features and applications:

- Ideal architecture for video capture applications
- *PCI r2.2* with VPD support
- Lowest power bridge, supporting up to ten PCI masters
- Synchronous primary and secondary PCI Bus operation
- High performance, No Retry penalty flow with uninterrupted zero wait state burst up to 1 KB
- Concurrent primary and secondary port operation supports traffic isolation
- Provides programmable arbitration support for up to ten bus masters on the secondary interface
- Five buffered secondary PCI clock outputs
- Enhanced address decoding
  - Supports 32-bit I/O Address range
  - Supports 64-bit Memory Address range
  - ISA-Aware mode for legacy support in the first 64 KB of I/O Address range
  - VGA addressing and VGA palette snooping support
- Supports 3.3V PCI with 5V tolerant I/O
- Industry-standard 208-pin Plastic Quad Flat Pack (PQFP) and 160-pin Tiny Ball Grid Array (TinyBGA) packages



PCI 6156 Application Example

1 INTRODUCTION

This section provides information about PLX Technology, Inc., and its products, the PCI 6000 Bridge Series, and PCI 6156 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 PCI 6000 BRIDGE SERIES

The PLX 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.

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 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 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 PCI 6000 series products are supported by PLX comprehensive reference design tools and the industry-recognized PLX support infrastructure.

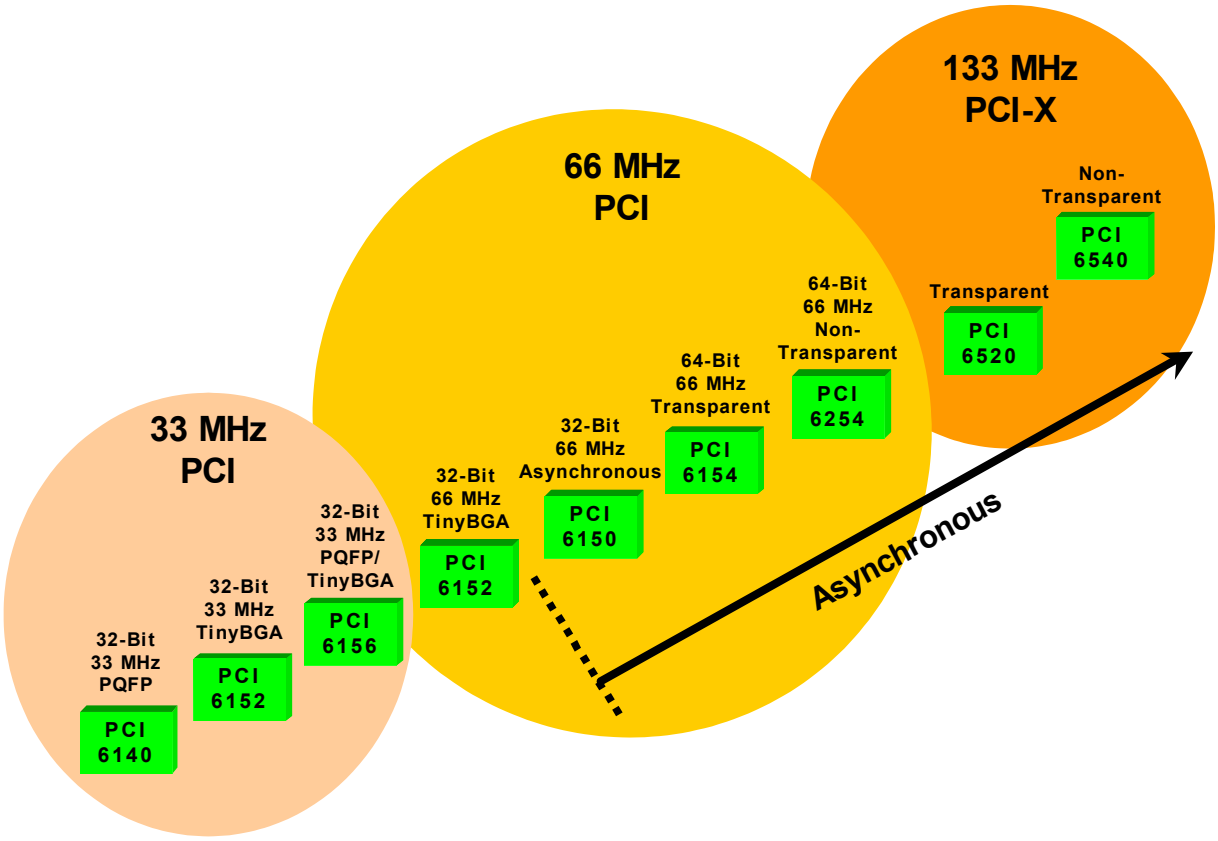


Figure 1-1. PCI 6000 Bridge Series

1.2.1 PCI 6156

As illustrated in Figure 1-2, the PCI 6156 is a two-port device providing *synchronous* operation between the *primary* and *secondary* ports.

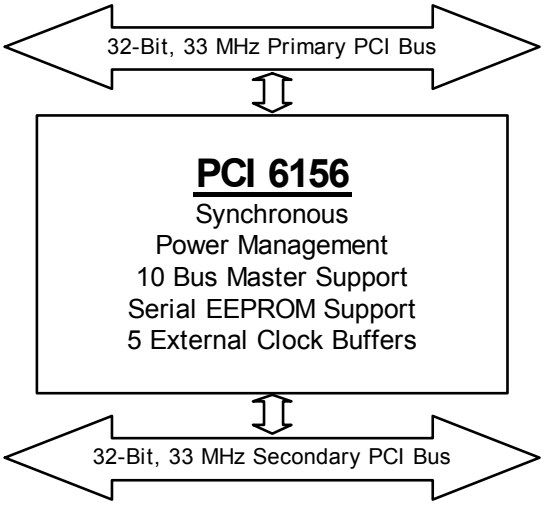


Figure 1-2. PCI 6156 PCI-to-PCI Bridge

1.3 FEATURE DESCRIPTION

The PCI 6156 is built upon the powerful PLX PCI-to-PCI Bridge Architecture. It addresses the needs of the surveillance market by offering a system-level performance advantage. In particular, the PCI 6156 can support up to ten video capture chips by providing ten pairs of PCI request and grant lines. Features include:

- Ideal architecture for video capture applications
- *PCI r2.2* with VPD support
- Lowest power bridge, supporting up to ten PCI masters
- Synchronous primary and secondary PCI Bus operation
- High performance, no Retry penalty flow with uninterrupted zero wait state burst up to 1 KB
- Concurrent primary and secondary port operation supports traffic isolation
- Provides programmable arbitration support for up to ten bus masters on the secondary interface
- Five buffered secondary PCI clock outputs
- Enhanced address decoding
- Supports 32-bit I/O address range
- Supports 64-bit memory address range
- ISA-Aware mode for legacy support in the first 64 KB of I/O address range
- VGA addressing and VGA palette snooping support
- Supports 3.3V PCI with 5V tolerant I/O
- Industry-standard 208-pin Plastic Quad Flat Pack (PQFP) and 160-pin Tiny Ball Grid Array (TinyBGA) packages

1.4 APPLICATIONS

1.4.1 Multiple Device Expansion

Figure 1-3 illustrates the PCI 6156 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, 33 MHz. This configuration is a common mechanism for providing multiple PCI devices on a single bus without exceeding the bus load limitation defined in *PCI r2.2*.

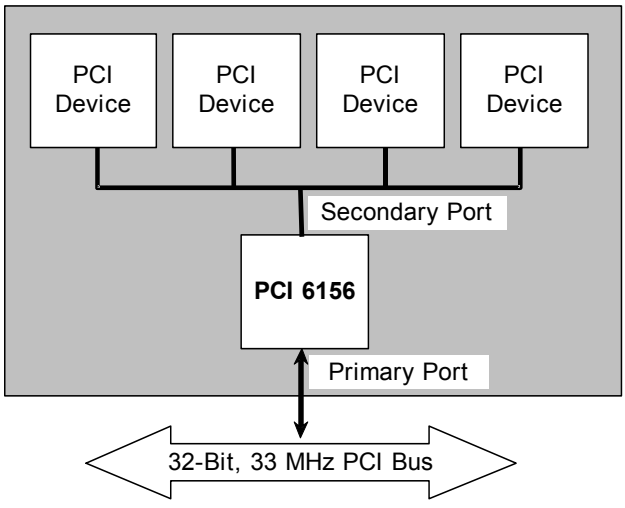


Figure 1-3. Multiple Device Expansion

2 FUNCTIONAL OVERVIEW

This section describes general operation of the PCI 6156 bridge, and provides an overview of write and read transactions.

2.1 GENERAL OPERATION

The primary and secondary PCI port runs at the same (synchronous) frequency, 33 MHz, operate at a 32-bit bus data width.

The PCI 6156 provides an Internal Arbiter function on the secondary bus, for up to ten secondary bus masters. The PCI 6156 also sources five secondary PCI clock outputs.

The PCI 6156 provides features satisfying the requirements of *PCI Power Mgmt. r1.0*, supporting Power Management states D_0 through D_{3cold} and D_{3hot}. (Refer to Section 15, "Power Management," for further details.)

The PCI 6156 supports a serial EEPROM device for register configuration data. This allows the PCI 6156 to automatically load custom configuration upon power-up, which minimizes the software overhead of configuring the bridge through a host processor.

The PCI 6156 fully supports Vital Product Data (VPD) by providing the Address, Data, and Control registers (PVPDAD; PCI:A2h, PVPDATA; PCI:A4h, PVPDID; PCI:A0h, and PVPD_NEXT; PCI:A1h) for accessing VPD stored in the unused portion of the serial EEPROM. VPD allows reading or writing of user data to the upper 224 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 21, "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.

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.

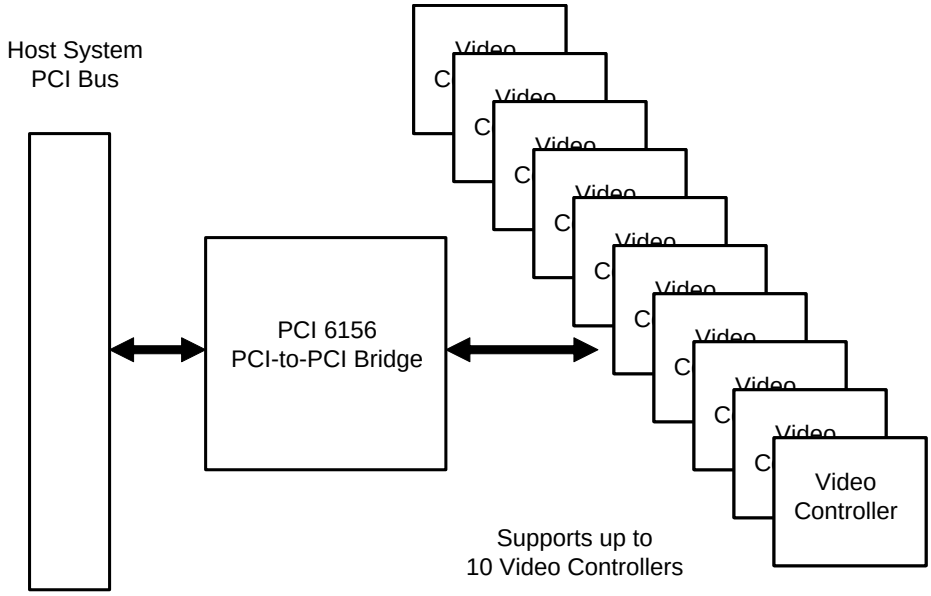


Figure 2-1. PCI 6156 Application Example

3 PIN DESCRIPTION

This section describes the PCI 6156 pins, including pin summary, pull-up and pull-down resistor recommendations, and pinout listings.

3.1 PIN SUMMARY

Tables 3-3 through Table 3-9 describe each PCI 6156 pin:

- PCI Primary Bus Interface
- PCI Secondary Bus Interface
- Clock Related
- Reset
- Serial EEPROM Interface
- Power and Ground
- Reserved and No Connect

For a visual view of the PCI 6156 pinout, refer to Section 17, “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.
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 r2.2*.

Table 3-2. Generic PCI Bus Interface Pins that follow *PCI r2.2* 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_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[9:0]#, S_IRDY#, S_PAR, S_PERR#, S_REQ[9:0]#, S_SERR#, S_STOP#, S_TRDY#

The following guidelines are not exhaustive and should be read in conjunction with the appropriate sections of *PCI r2.2*.

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#, PERR#, SERR#, STOP#, and TRDY#. The point-to-point and shared bus signals require pull-up resistors. The value of these pull-up resistors depends on the bus loading. *PCI r2.2* provides formulas for calculating these resistors.

When making adapter card devices in which the PCI 6156 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 6156 port that is not connected to the motherboard or host system.

The S_REQ[9:0]# inputs must be pulled high with a 10K-Ohm pull-up resistor.

3.2.2 Clock-Related Pins

Clock routing is detailed in Section 4, “Clocking.” Pull-up resistors are not required on the S_CLKO[4:0] pins; however, a series termination resistor is required when using these pins. Neither pull-up nor pull-down resistors are required on P_CLKIN or S_CLKIN.

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 pull-up nor pull-down resistors.

3.2.4 Serial EEPROM Pins

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

3.2.5 RESERVED Pin

The RESERVED pin is available only in the 208-pin PQFP package. It is suggested that an option be provided to add a pull-up resistor to this pin in all PCB layouts.

3.3 PINOUT

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

Table 3-3. Primary PCI Bus Interface Pins

Symbol	Signal Name	Total Pins	Pin Type	PQFP Pin Number	TinyBGA Pin Number	Function
P_AD[31:0]	Primary Address and Data	32	I/O PTS PCI	87, 89, 90, 91, 93, 94, 95, 99, 111, 112, 114, 115, 116, 118, 120, 122, 141, 142, 143, 145, 147, 148, 150, 151, 161, 162, 163, 167, 169, 170, 173, 174	P10, N10, M10, P11, N11, M11, P12, N12, M14, L12, L13, L14, K12, K13, K14, J12, E14, E13, E12, D14, D13, D12, C13, B14, B12, A12, C11, B11, A11, C10, A10, C9	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.
P_CBE[3:0]#	Primary Command and Byte Enables	4	I/O PTS PCI	108, 124, 139, 160	P14, J13, F12, A13	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.
P_DEVSEL#	Primary Device Select	1	I/O PSTS PCI	129	H14	Asserted by the target, indicating that the device is accepting the transaction. As a master, the PCI 6156 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	125	J14	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	PI	83	N9	When asserted, the PCI 6156 can access the primary bus.
P_IDSEL	Primary Initialization Device Select	1	PI	109	N14	Used as a Chip Select line for Type 0 Configuration accesses to PCI 6156 Configuration space.

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

Symbol	Signal Name	Total Pins	Pin Type	PQFP Pin Number	TinyBGA Pin Number	Function
P_IRDY#	Primary Initiator Ready	1	I/O PSTS PCI	126	H12	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_PAR	Primary Parity	1	I/O PTS PCI	138	F13	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.
P_PERR#	Primary Parity Error	1	I/O PSTS PCI	136	G12	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.
P_REQ#	Primary Request	1	PTS	85	M9	Asserted by the PCI 6156 to request ownership of the primary bus to perform a transaction. The PCI 6156 de-asserts P_REQ# for at least two PCI Clock cycles before re-asserting it. (Refer to Section 12.2, “Primary PCI Bus Arbitration,” for further details.)
P_SERR#	Primary System Error	1	PTS	137	F14	P_SERR# can be driven low by any device to indicate a System error condition. The PCI 6156 drives P_SERR# if one of the following conditions is met: • Address Parity error • S_SERR# is asserted P_SERR# is pulled up through an external resistor.

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

Symbol	Signal Name	Total Pins	Pin Type	PQFP Pin Number	TinyBGA Pin Number	Function
P_STOP#	Primary Stop	1	I/O PSTS PCI	130	G14	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	128	H13	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		47				

Table 3-4. Secondary PCI Bus Interface Pins

Symbol	Signal Name	Total Pins	Pin Type	PQFP Pin Number	TinyBGA Pin Number	Function
S_AD[31:0]	Secondary Address and Data	32	I/O PTS PCI	45, 44, 40, 39, 38, 35, 34, 32, 30, 28, 27, 26, 24, 23, 22, 20, 198, 197, 194, 193, 191, 189, 188, 187, 185, 183, 182, 181, 179, 178, 177, 175	M1, L3, L2, L1, K3, K1, J3, J2, H3, H2, H1, G1, G2, G3, F1, F2, A3, C4, A4, C5, B5, A5, C6, B6, C7, B7, A7, A8, B8, C8, A9, B9	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.
S_CBE[3:0]#	Secondary Command and Byte Enables	4	I/O PTS PCI	31, 18, 203, 186	J1, F3, A2, A6	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.
S_DEVSEL#	Secondary Device Select	1	I/O PSTS PCI	10	D2	Asserted by the target, indicating that the device is accepting the transaction. As a master, the PCI 6156 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	14	E2	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_GNT[9:0]#	Secondary Grant	10	PO	202, 195, 149, 110, 79, 76, 63, 61, 59, 58	B3, B4, C14, M13, N8, N7, N4, M4, P3, N3	Asserted by the PCI 6156 to access the secondary bus. The PCI 6156 de-asserts S_GNT[9:0]# for at least two PCI Clock cycles before re-asserting them.
S_IRDY#	Secondary Initiator Ready	1	I/O PSTS PCI	13	E3	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.

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

Symbol	Signal Name	Total Pins	Pin Type	PQFP Pin Number	TinyBGA Pin Number	Function
S_PAR	Secondary Parity	1	I/O PTS PCI	6	B1	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.
S_PERR#	Secondary Parity Error	1	I/O PSTS PCI	8	C1	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_REQ[9:0]#	Secondary Request	10	PI	204, 171, 152, 100, 36, 77, 57, 48, 47, 46	A1, B10, A14, P13, K2, P7, P2, P1, N1, M2	Asserted by an external device to request secondary bus ownership to perform a transaction. S_REQ[9:0]# ***must*** be externally pulled up through 10K-Ohm resistors to V_{DD}.
S_SERR#	Secondary System Error	1	PI	7	C2	S_SERR# can be driven low by any device to indicate a System error condition.
S_STOP#	Secondary Stop	1	I/O PSTS PCI	9	D3	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.
S_TRDY#	Secondary Target Ready	1	I/O PSTS PCI	12	D1	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 3-5. Clock-Related Pins

Symbol	Signal Name	Total Pins	Pin Type	PQFP Pin Number	TinyBGA Pin Number	Function
P_CLKIN	Primary Clock Input	1	PI	81	M8	Provides timing for primary interface transactions.
S_CLKIN	Secondary Clock Input	1	PI	65	M5	Provides timing for secondary interface transactions.
S_CLKO[4:0]	Secondary Clock Output	5	O	75, 73, 71, 69, 67	M7, P6, N6, M6, P5	Provides Secondary Clock phase synchronous with the PCI clock. Pull-up resistors are not required on S_CLKO[4:0]; however, a series termination resistor is required when using these pins. ***Note:*** *When supporting more than four PCI devices behind the bridge, consider driving two devices using one S_CLKOx signal if the two devices are close to one another, and do not run faster than 33 MHz.*
Total		7				

Table 3-6. Reset Pins

Symbol	Signal Name	Total Pins	Pin Type	PQFP Pin Number	TinyBGA Pin Number	Function
P_RSTIN#	Primary Reset Input	1	PI	78	P8	When P_RSTIN# is active, asynchronously place outputs in a high-impedance state, and float P_SERR# and P_GNT#. May require a pull-up resistor, depending on the application.
S_RSTOUT#	Secondary Reset Output	1	PO	64	P4	Asserted when any 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 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 3-7. Serial EEPROM Pins

Symbol	Signal Name	Total Pins	Pin Type	PQFP Pin Number	TinyBGA Pin Number	Function
EEPCLK	Serial EEPROM Clock	1	O	135	G13	Clock signal to the serial EEPROM interface. Used during autoload and for VPD functions.
EEPDATA	Serial EEPROM Data	1	I/O	16	E1	Serial data interface to the serial EEPROM. Requires an external pull-up resistor.
Total		2				

Table 3-8. Power and Ground Pins

Symbol	Signal Name	Total PQFP Pins	Total TinyBGA Pins	Pin Type	PQFP Pin Number	TinyBGA Pin Number	Function
V_{DD}	Power	25	16	I	11, 21, 29, 37, 51, 60, 62, 66, 70, 82, 86, 92, 98, 103, 117, 127, 132, 140, 155, 166, 172, 190, 196, 201, 207	D6, D7, D8, D9, F4, F11, G4, G11, H4, H11, J4, J11, L6, L7, L8, L9	+3.3V power supply.
V_{SS}	Ground	30	20	I	2, 15, 25, 33, 41, 42, 54, 68, 72, 74, 80, 84, 88, 96, 105, 113, 119, 121, 123, 134, 144, 146, 158, 164, 168, 176, 180, 184, 192, 199	B2, B13, C3, C12, D4, D5, D10, D11, E4, E11, K4, K11, L4, L5, L10, L11, M3, M12, N2, N13	Ground.
Total		55	36				

Table 3-9. Reserved and No Connect Pins

Symbol	Signal Name	Total PQFP Pins	Total TinyBGA Pins	Pin Type	PQFP Pin Number	TinyBGA Pin Number	Function
RESERVED	***Reserved***	1	—	—	43	—	Available only in the 208-pin PQFP package. It is suggested that an option be provided to add a pull-up resistor to this pin in all PCB layouts.
NC	No Connect	30	2	—	1, 3, 4, 5, 17, 19, 49, 50, 52, 53, 55, 56, 97, 101, 102, 104, 106, 107, 131, 133, 153, 154, 156, 157, 159, 165, 200, 205, 206, 208	N5, P9	No connect pins, which are not to be connected or used as routing channels. May be used in future PCI 6156 revisions.
Total		31	2				

4 CLOCKING

This section describes the PCI 6156 clocking requirements.

To correctly operate, the PCI 6156 requires both a primary and secondary clock. Additionally, a serial resistor must be added as close as possible to the source. Because one clock source drives two devices, the trace length must be the same after the clock trace splits in two.

4.1 PRIMARY AND SECONDARY CLOCK INPUTS

The PCI 6156 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.

4.2 SECONDARY CLOCK OUTPUTS

The PCI 6156 has five secondary clock outputs that can be used to drive up to ten external secondary bus devices, with one feedback to S_CLKIN. Two to three devices may share one clock source.

The PCI 6156 is a synchronous design, and supports only 1:1 frequency ratio on the primary and secondary bus interfaces.

The rules for using secondary clocks are as follows:

- Each secondary clock output is limited to no more than three loads at 33 MHz
- One of the secondary clock outputs must be used for PCI 6156 S_CLKIN input
- To minimize skew between secondary clocks and allow a maximum etch delay of 2 ns, use an equivalent amount of etch on the board for all secondary clocks
- Each clock trace length, including the feedback clock to the PCI 6156 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

5 RESET AND INITIALIZATION

This section describes primary, secondary, and Power Management reset, as well as register initialization.

5.1 RESET

This subsection describes the primary and secondary interface reset mechanisms. The PCI 6156 has two reset mechanisms and two reset pins—P_RSTIN# and S_RSTOUT#. In addition, the PCI 6156 can respond to Power Management-initiated internal resets.

After the Reset signals are de-asserted, the PCI 6156 requires 512 clocks to initialize bridge functions. During this initialization, Type 0 accesses can be accepted.

5.1.1 Primary Reset Input

When P_RSTIN# is asserted, the following events occur:

1. PCI 6156 immediately places all primary and secondary PCI interface signals into a high-impedance state.
2. All registers are reset.
3. P_RSTIN# assertion automatically causes a secondary port reset and S_RSTOUT# assertion.
4. Forty-three clocks after P_RSTIN# goes high, S_RSTOUT# goes high.

The asserting and de-asserting edges of P_RSTIN# can be asynchronous to P_CLKIN and S_CLKIN.

When P_RSTIN# is asserted, all 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.

5.1.2 Secondary Reset Output

The PCI 6156 is responsible for driving the secondary bus reset signal, S_RSTOUT#. The PCI 6156 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[9:0]#, 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 6156 remains accessible during secondary interface reset and continues to respond to Configuration Space accesses from the primary interface.

5.1.3 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:84h), an internal reset equivalent to P_RSTIN# is generated and all relevant registers are reset. However, S_RSTOUT# is ***not*** asserted.

5.2 REGISTER INITIALIZATION

The PCI 6156 Configuration registers may be initialized in one of three ways:

- Default values
- Serial EEPROM contents
- Host initialization

5.2.1 Default Initialization

After P_RSTIN# de-assertion, the PCI 6156 automatically checks for a valid a serial EEPROM. If the serial EEPROM is not valid nor present, the PCI 6156 automatically loads default values into the Configuration registers. (Refer to the "Value after Reset" column of the register tables in Section 6, "Registers.")

5.2.2 Serial EEPROM Initialization

After P_RSTIN# de-assertion, if the PCI 6156 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.")

5.2.3 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 6156 registers.

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 6156 family and to ensure compatibility with future enhancements, write 0 to all unused bits. 31 24 23 16 15 8 7 0	PCI Writable	Serial EEPROM Writable
00h	Device ID | Vendor ID	Yes	Yes
04h	Primary Status | Primary Command	Yes	No
08h	Class Code | Revision ID	Yes	Yes
0Ch	***Reserved*** | 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	No	No
38h	***Reserved***	No	No
3Ch	Bridge Control | Interrupt Pin | ***Reserved***	Yes	No
40h	Arbiter Control | Chip Control	Yes	No
44h – 67h	***Reserved***	No	No
68h	***Reserved*** | Secondary Clock Control	Yes	No
72h – 79h	***Reserved***	No	No
80h	Power Management Capabilities | Power Management Next Capability Pointer (A0h) | Power Management Capability ID (01h)	Yes	Yes
84h	Power Management Data | PMCSR Bridge Supports Extensions ***(Reserved)*** | Power Management Control/Status	Yes	Yes
88h – 9Fh	***Reserved***	No	No
A0h	VPD Address (0h) | VPD Next Capability Pointer (0h) | VPD Capability ID (03h)	Yes	No
A4h	VPD Data (0h)	Yes	No
A8h – BFh	***Reserved***	No	No
C0h	Arbiter Control | ***Reserved*** | Miscellaneous Control 1 | ***Reserved***	Yes	No
C4h	***Reserved*** | Miscellaneous Control 2	Yes	No
C8h	Serial EEPROM Data | Serial EEPROM Address | Serial EEPROM Control	Yes	Yes
CCh	Test | ***Reserved***	Yes	Yes
D0h – FFh	***Reserved***	No	No

Note: *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 6156 manufacturer. Defaults to the PCI-SIG-issued PLX Vendor ID (3388h), if a blank or no serial EEPROM is present.	Yes	No; Serial EEPROM	3388h
31:16	**Device ID.** Identifies the particular device. Defaults to PLX PCI 6156 part number (0031h), if a blank or no serial EEPROM is present.	Yes	No; Serial EEPROM	0031h

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 set to 1 and BCNTRL[3]=1; PCI:3Eh (VGA mode enabled), PCI 6156 behaves as if only the VGA Enable bit is set.*	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 6156 performs address/data stepping.	Yes	No	0
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 primary 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.	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	1
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 6156 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 6156 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 6156 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 6156 revision.	Yes	No	1h

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	No; Serial EEPROM	0h
15:8	**Subclass Code.** PCI-to-PCI bridge or other bridge device.	Yes	No; Serial EEPROM	04h
23:16	**Base Class Code.** Bridge device.	Yes	No; 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 (in units of 32-bit words) of a power of two are valid. Maximum value is 20h. For values greater than 20h, PCI 6156 operates as if PCICLSR is programmed with value of 08h. Used when terminating Memory Write and Invalidate transactions and prefetching. ***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 6156, 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	No; 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	No; Serial EEPROM	0

Note: *PCIHTR is hardcoded to 01h.*

Register 6-9. (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 6-10. (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 6-11. (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 6-12. (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 6156, 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 6-13. (PCIIOBAR; PCI:1Ch) I/O Base

Bit	Description	Read	Write	Value after Reset
7:0	**I/O Base.** Specifies the Base I/O 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 bits [3:0] are Read-Only and hardcoded to 01h to indicate 32-bit I/O addressing support.	Yes	Yes [7:4]	1h

Register 6-14. (PCIIOLMT; PCI:1Dh) I/O Limit

Bit	Description	Read	Write	Value after Reset
7:0	**I/O Limit.** Specifies the Upper Limit I/O 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 bits [3:0] are Read-Only and hardcoded to 01h to indicate 32-bit I/O addressing support.	Yes	Yes [7:4]	1h

Register 6-15. (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 6156 supports a 66 MHz PCI clock environment.	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 6156 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 6156 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 6156 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 6-16. (PCIMBAR; PCI:20h) Memory Base

Bit	Description	Read	Write	Value after Reset
15:0	**Memory Base.** Specifies the Base Memory-Mapped I/O 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 bits [3:0] are Read-Only and hardcoded to 0h.	Yes	Yes [15:4]	0h

Register 6-17. (PCIMLMT; PCI:22h) Memory Limit

Bit	Description	Read	Write	Value after Reset
15:0	**Memory Limit.** Specifies the Upper Limit Memory-Mapped I/O 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 bits [3:0] are Read-Only and hardcoded to 0h.	Yes	Yes [15:4]	0h

Register 6-18. (PCIPMBAR; PCI:24h) Prefetchable Memory Base

Bit	Description	Read	Write	Value after Reset
15:0	**Prefetchable Memory Base.** Specifies the Base Prefetchable Memory-Mapped 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 bits [3:0] are Read-Only and hardcoded to 01h, indicating 64-bit address support.	Yes	Yes [15:4]	1h

Register 6-19. (PCIPMLMT; PCI:26h) Prefetchable Memory Limit

Bit	Description	Read	Write	Value after Reset
15:0	**Prefetchable Memory Limit.** Specifies the Upper Limit Prefetchable Memory-Mapped 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 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 bits [3:0] are Read-Only and hardcoded to 01h, indicating 64-bit address support.	Yes	Yes [15:4]	1h

Register 6-20. (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 Base Prefetchable Memory-Mapped 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 6-21. (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 Limit Prefetchable Memory-Mapped 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 6-22. (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 Base I/O 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 6-23. (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 Limit I/O 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 6-24. (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 Power Management capability location in the New Capabilities Linked List (80h).	Yes	No	80h
31:8	***Reserved.***	Yes	No	0h

Register 6-25. (PCIIPR; PCI:3Dh) PCI Interrupt Pin

Bit	Description	Read	Write	Value after Reset
7:0	**Interrupt Pin.** Reads as 0h to indicate that PCI 6156 does not use interrupt pins.	Yes	No	0h

Register 6-26. (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.	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 and PCICR[3]=1 (VGA Palette Snoop mode enabled), PCI 6156 behaves as if only the VGA Enable bit is set.*	Yes	Yes	0

Register 6-26. (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).	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 = Enable Fast Back-to-Back transactions	Yes	Yes	0
11:8	***Reserved.*** Can be used as a Software register.	Yes	Yes	0h
15:12	***Reserved.***	Yes	No	0h

6.1.2 Device-Specific

6.1.2.1 Chip and Arbiter Control

Register 6-27. (CCNTRL; PCI:40h) Chip Control

Bit	Description	Read	Write	Value after Reset
3:0	***Reserved.*** Set to 0.	Yes	No	0h
4	**Secondary Bus Prefetch Disable.** Controls PCI 6156 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 6156 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
15:5	***Reserved.***	Yes	No	0h

Register 6-28. (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_REQ9# is always in the low-priority group.*	Yes	Yes	0h
9	**PCI 6156 Priority.** Defines whether PCI 6156 secondary port is in the high- or low-priority group. 0 = Low-priority group 1 = High-priority group	Yes	Yes	1
15:10	***Reserved.*** Set to 0h.	Yes	No	0h

6.1.2.2 Secondary Clock Control

Register 6-29. (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.	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.	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.	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.	Yes	Yes	00b
8	**Clock 4 Disable.** If 0, S_CLKO4 is enabled. When 1, S_CLKO4 is disabled.	Yes	Yes	0
9	**Primary Order Control.** Value of 1 enforces primary port completion order. This bit can also be controlled by serial EEPROM address 02h, bit 6.	Yes	Yes	0
10	**Secondary Order Control.** Value of 1 enforces secondary port completion order. This bit can also be controlled by serial EEPROM address 02h, bit 7.	Yes	Yes	0
11	**Fast Secondary Grant.** Value of 1 enables Fast S_GNT[9:0]# timing. This bit can also be controlled by serial EEPROM address 02h, bit 5. ***Caution: Use of this feature must be strictly qualified according to customer application environment.***	Yes	Yes	0
14:12	**Secondary Grant Count.** Values: 000b = Disable (default) Non-0 = Minimum number of clocks S_GNT# is valid Bit 14 can also be controlled by serial EEPROM address 03h, bit 6. Bits [13:12] can also be controlled by serial EEPROM address 03h, bit 5.	Yes	Yes	000b
15	**Primary Request Control.** Value of 1 causes P_REQ# to enforce two-clock inactive time. This bit can also be controlled by serial EEPROM address 03h, bit 7.	Yes	Yes	0

6.1.2.3 Power Management Capability

Register 6-30. (PMCAPID; PCI:80h) 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 6-31. (PMNEXT; PCI:81h) 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 VPD capability location in the New Capabilities Linked List (A0h).	Yes	No	A0h

Register 6-32. (PMC; PCI:82h) Power Management Capabilities

Bit	Description	Read	Write	Value after Reset
2:0	**Version.** Set to 001b, indicates that this function complies with *PCI Power Mgmt. r1.0*.	Yes	No; Serial EEPROM	001b
3	**PME Clock.** Set to 0, because PCI 6156 does not support PME# signaling.	Yes	No; Serial EEPROM	0
4	**Auxiliary Power Source.** Set to 0, because PCI 6156 does not support PME# signaling.	Yes	No; Serial EEPROM	0
5	**Device-Specific Initialization (DSI).** Returns 0, indicating PCI 6156 does not require special initialization.	Yes	No; Serial EEPROM	0
8:6	***Reserved.***	Yes	No	000b
9	**D_1 Support.** Returns 1, indicating that PCI 6156 supports the D_1 device power state.	Yes	No; Serial EEPROM	1
10	**D_2 Support.** Returns 1, indicating that PCI 6156 supports the D_2 device power state.	Yes	No; Serial EEPROM	1
15:11	**PME Support.** Set to 7E02b.	Yes	No	0h

Register 6-33. (PMCSR; PCI:84h) 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}	Yes	Yes; Serial EEPROM	00b
7:2	***Reserved.***	Yes	No	0h
8	**PME Enable.** Set to 0, because PCI 6156 does not support PME# signaling.	Yes	No; Serial EEPROM	0
12:9	**Data Select.** Used to select which data is reported through PMCDATA; PCI:87h and PMCSR[14:13].	Yes	Yes; Serial EEPROM	0h
14:13	**Data Scale.** Reports the state-dependent data requested by PMCSR[12:9]. Register value is scaled according to the value reported by PMCSR[14:13].	Yes	No; Serial EEPROM	00b
15	**PME Status.** Set to 0, because PCI 6156 does not support PME# signaling.	Yes	No; Serial EEPROM	0

Register 6-34. (PMCSR_BSE; PCI:86h) PMCSR Bridge Supports Extensions

Bit	Description	Read	Write	Value after Reset
7:0	***Reserved.***	Yes	No	0h

Register 6-35. (PMCDATA; PCI:87h) 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]; PCI:84h. Register value is scaled by the value reported by PMCSR[14:13].	Yes	No; Serial EEPROM	0h

6.1.2.4 VPD Capability

Register 6-36. (PVPDID; PCI:A0h) 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 6-37. (PVPD_NEXT; PCI:A1h) 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 (0h). ***Note:*** *0h indicates the end of the Capabilities list.*	Yes	No	0h

Register 6-38. (PVPDAD; PCI:A2h) 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 6156 accesses the serial EEPROM at address PVPDAD[7:2]+20h. The 20h offset ensures that VPD accesses do not overwrite the PCI 6156 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:A4h). 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 6-39. (PVPDATA; PCI:A4h) 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

6.1.2.5 Miscellaneous and Internal Arbiter Control

Register 6-40. (MSCCNTRL1; PCI:C1h) Miscellaneous Control 1

Bit	Description	Read	Write	Value after Reset
0	**ISA I/O.** Enables several I/O addresses to be located behind the PCI 6156. If set, the following I/O addresses belong to the secondary bus: • 0207h – 0200h = Game port • 0233h – 0220h = Audio • 0331h – 0330h = MIDI • 038Bh – 0388h = FM	Yes	Yes	0
1	**Memory Read Line Control.** If 1, PCI 6156 always stops prefetch on Cache Line boundaries on Memory Read Line transactions.	Yes	Yes	0
2	**Read Byte Enable Control.** If 1, PCI 6156 forces all Byte Enables to be active during Read Burst cycles.	Yes	Yes	0
3	***Reserved.***	Yes	Yes	0
4	**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
5	**Low-Priority Group Arbitration Order.** Valid only when the low-priority arbitration group is set to a fixed-priority 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[3:0]; PCI:C3h.	Yes	Yes	0
6	**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
7	**High-Priority Group Arbitration Order.** Valid only when the high-priority arbitration group is set to a fixed-priority 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]; PCI:C3h.	Yes	Yes	0

Register 6-41. (IACNTRL; PCI:C3h) Internal Arbiter Control

Bit	Description	Read	Write	Value after Reset
3:0	**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 6156 retains highest priority 1010b = Master 9 retains highest priority 1011b – 1111b = ***Reserved***	Yes	Yes	0000b
7:4	**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 6156 retains highest priority 1010b – 1111b = ***Reserved*** ***Note:*** *S_REQ9# is always in the low-priority group.*	Yes	Yes	0000b

Register 6-42. (MSCCNTRL2; PCI:C4h) Miscellaneous Control 2

Bit	Description	Read	Write	Value after Reset
2:0	***Reserved.***	Yes	No	000b
3	**S_GNT[9:0]# De-Assertion.** If 1, PCI 6156 de-asserts S_GNT[9:0]# one clock after P_GNT# is de-asserted. Otherwise, S_GNT[9:0]# are de-asserted at the same time as P_GNT#.	Yes	Yes	0
4	**Secondary-to-Primary Transaction Delay.** Specifies delay for transactions going from secondary to primary PCI interface. Values: 0 = Delay secondary-to-primary bus transfer by two PCI clocks 1 = Delay secondary-to-primary bus transfer by one PCI clock	Yes	Yes	0
5	**Primary-to-Secondary Transaction Delay.** Specifies delay for transactions going from primary to secondary PCI interface. Values: 0 = Delay primary-to-secondary bus transfer by two PCI clocks 1 = Delay primary-to-secondary bus transfer by one PCI clock	Yes	Yes	0
6	**Retry Secondary Master.** If 0, and PCI 6156 was granted access to the primary bus, and a secondary master initiates a cycle to access the primary bus while it remains busy, the PCI 6156 waits for the primary bus to become idle instead of immediately Retrying the secondary master. If 1, PCI 6156 immediately Retrys the secondary master if granted access to the primary bus while the primary bus is busy.	Yes	Yes	0
7	**Back-to-Back Cycle Enable.** Enables back-to-back cycles on the primary interface. Fast Back-to-Back is enabled (PCICR[9]=1; PCI:04h).	Yes	Yes	0

6.1.2.6 Serial EEPROM and Test

Register 6-43. (EEPCNTRL; PCI:C8h) 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: 00b = ***Reserved*** 01b = PCI clock / 256 (Used for 33 MHz PCI) 10b = PCI clock / 128 11b = PCI clock (Test mode), read returns last written value	Yes	Yes	01b

Register 6-44. (EEPADDR; PCI:C9h) 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:CAh) 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	—

Register 6-46. (TEST; PCI:CFh) Test

Bit	Description	Read	Write	Value after Reset
0	**Serial EEPROM Autoload Control.** If set to 1, disables serial EEPROM autoload.	Yes	Yes	0
1	**Fast Serial EEPROM Autoload.** If set to 1, speeds up serial EEPROM autoload.	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

7 SERIAL EEPROM

This section describes information specific to the PCI 6156 serial EEPROM interface and use—access, Autoload mode, reset, data structure. and how it associates with VPD.

7.1 OVERVIEW

Important Note: *Erroneous serial EEPROM data can cause the PCI 6156 to lock the system. Provide an optional switch or jumper to disable the serial EEPROM in board designs.*

The PCI 6156 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 6156 registers, and alleviates the need for user software to configure the PCI 6156. If a programmed serial EEPROM is connected, the PCI 6156 automatically loads data from the serial EEPROM after P_RSTIN# de-assertion.

The serial EEPROM data structure is defined in Section 7.5.1. The serial EEPROM interface is organized on a 16-bit base in Little Endian format, and the PCI 6156 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

Note: *The PCI 6156 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 6156 can access the serial EEPROM on a Word basis, using the hardware sequencer. Users access one Word data by way of the PCI 6156 Serial EEPROM Control register:

- Serial EEPROM Start/Read/Write Control (EEPCNTRL; PCI:C8h)
- Serial EEPROM Address (EEPADDR; PCI:C9h)
- Serial EEPROM Data (EEPDATA; PCI:CAh)

Before each access, software should check the Auto Mode Cycle in Progress status (EEPCNTRL[0]; PCI:C8h, 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:C9h) with the Word address.
2. **Writes**—Program Word data to the Serial EEPROM Data register (EEPDATA; PCI:CAh).

   Reads—Proceed to the next step.
3. **Writes**—Set the Serial EEPROM Command and Start bits (EEPCNTRL[1:0]=11b; PCI:C8h, respectively) to start the Serial EEPROM Sequencer.

   Reads—Set the Start bit (EEPCNTRL[1:0]=01b; PCI:C8h) 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:C8h):

   Writes—Data was successfully written to the serial EEPROM.

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

7.3 SERIAL EEPROM AUTOLOAD MODE AT RESET

Upon P_RSTIN# going high, the PCI 6156 autoloads the serial EEPROM data into the internal PCI 6156 registers.

The PCI 6156 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 commences immediately commences after reset. During autoload, the PCI 6156 reads sequential words from the serial EEPROM and writes to the appropriate registers. If a blank serial EEPROM is connected, the PCI 6156 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 6156 also stops loading the serial EEPROM contents after attempting to read the first word.

7.3.1 Autoload Mode Serial EEPROM Access

Using Autoload mode, the PCI 6156 can access the serial EEPROM on a word basis, by way of a hardware sequencer. Access word data by way of the PCI 6156 configuration registers for serial EEPROM Read/Write Command, Start, Control, and Address (EEPCNTRL[1:0]; PCI:C8h, TEST[0]; PCI:CFh, and EEPADDR[7:1]; PCI:C9h, respectively). Before each access, check the Serial EEPROM Autoload Status bit (TEST[2]; PCI:CDh) with software before issuing the next Start.

7.4 SERIAL EEPROM MODE AT RESET

During Reset, the PCI 6156 autoloads input for the serial EEPROM automatic load condition. The first serial EEPROM offset contains a signature (1516h). If the signature is recognized, register autoload commences immediately after reset. During autoload, the PCI 6156 reads sequential words from the serial EEPROM and writes to the appropriate registers.

Before the PCI 6156 registers can be accessed through the host, check the autoload condition by reading the Serial EEPROM Autoload Status bit (TEST[2]; PCI:CFh). Host access is allowed only after TEST[2]=0, which indicates that the autoload initialization sequence is complete.

7.5 SERIAL EEPROM DATA STRUCTURE

Following reset and the previously described conditions, the PCI 6156 autoloads the registers with serial EEPROM data. Figure 7-1 illustrates the serial EEPROM data structure.

The PCI 6156 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 6156 supports only Serial EEPROM Device Address 0.**

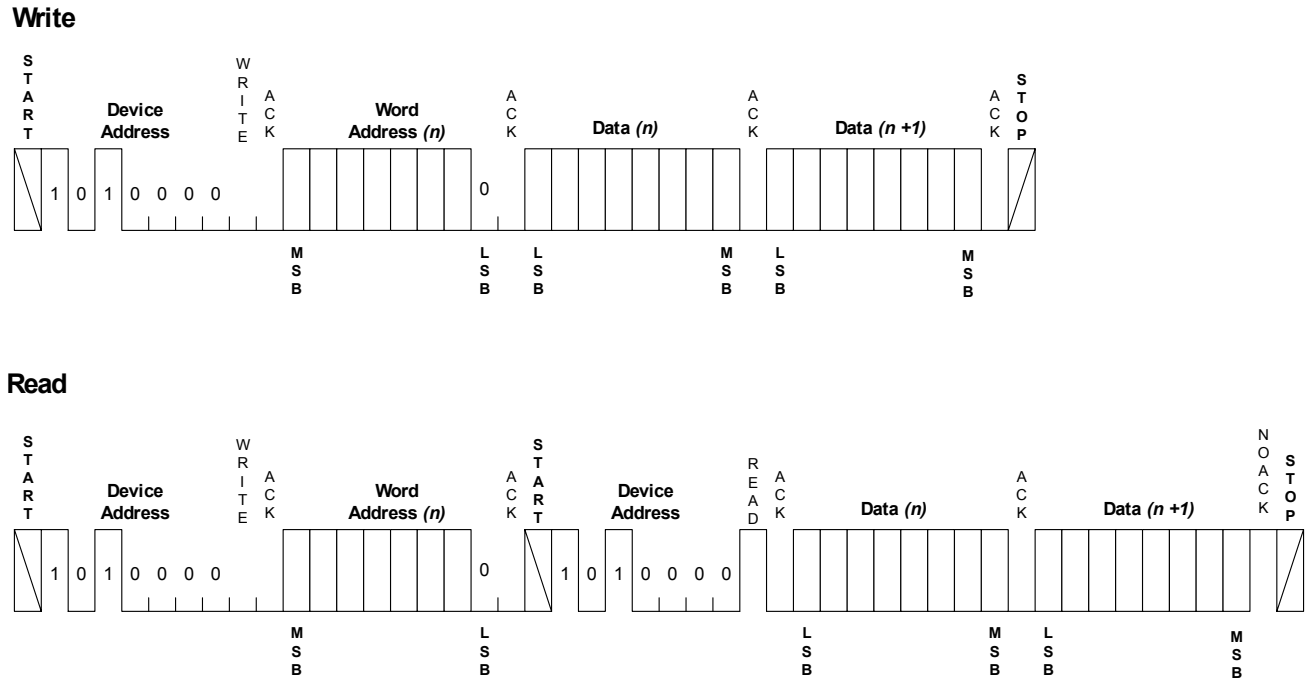


Figure 7-1. Serial EEPROM Data Structure

7.5.1 Serial EEPROM Address and Corresponding PCI 6156 Registers

Table 7-1. Serial EEPROM Address and Corresponding PCI 6156 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 02h. 0001b = Stops autoload at serial EEPROM offset 04h. 0011b = Stops autoload at serial EEPROM offset 07h. 0111b = Stops autoload at serial EEPROM offset 11h. 1111b = Autoloads all serial EEPROM loadable registers 11h. Other combinations are undefined. Bit 5 = SCLKCNTRL[11]; PCI:68h. Bit 6 = SCLKCNTRL[9]; PCI:68h. Bit 7 = SCLKCNTRL[10]; PCI:68h.
03h	—	**Secondary Clock Enable.** Valid combinations are: Bit 0 = 1 = Disables S_CLKO0. Bit 1 = 1= Disables S_CLKO1. Bit 2 = 1 = Disables S_CLKO2. Bit 3 = 1 = Disables S_CLKO3. Bit 4 = 1 = Disables S_CLKO4. Bit 5 = SCLKCNTRL[13:12]; PCI:68h use this serial EEPROM bit. Bit 6 = SCLKCNTRL[14]; PCI:68h. Bit 7 = SCLKCNTRL[15]; PCI:68h.
04h – 07h	F0h – F3h	***Reserved.***
08h – 09h	C0h – C4h	**Miscellaneous.** Valid combinations are: Bit 0 = ***Reserved.*** Bits [7:1] = MSCCNTRL2[7:1]; PCI:C4h. Bit 8 = MSCCNTRL1[0]; PCI:C1h. Bits [15:9] = MSCCNTRL1[7:1]; PCI:C1h.
0Ah – 0Bh	00h – 01h	**Vendor ID (PCIIDR[15:0]; PCI:00h).**
0Ch – 0Dh	02h – 03h	**Device ID (PCIIDR[31:16]; PCI:00h).**
0Eh – 0Fh	C3h, 42h	**Miscellaneous.** Valid combinations are: Bits [7:0] = ACNTRL; PCI:42h. Bits [15:8] = IACNTRL; PCI:C3h.
10h – 11h	82h	**Power Management Capability (PMC; PCI:82h).**
12h – 13h	87h	**Power Management Data.** Valid combinations are: Bits [7:0] = ***Reserved.*** Bits [15:8] = PMCDATA; PCI:87h.
14h – 3Fh	CFh	***Reserved.***

7.6 VITAL PRODUCT DATA

The PCI 6156 contains VPD registers, as specified in *PCI r2.2*. The VPD information is stored in the serial EEPROM device along with the autoload information. Refer to Section 16, "VPD," for further information regarding this feature.

8 PCI BUS OPERATION

This section describes PCI transactions to which the PCI 6156 responds and those it initiates when operating with one or both of its interfaces.

The PCI 6156 provides complete PCI-to-PCI bridge capability, allowing a PCI master and PCI slave on either side. It passes control and data between the primary and secondary bus to guarantee visibility from each side. The PCI 6156 is designed to behave as an intelligent buffer.

The PCI 6156 achieves zero wait-state bridging by controlling the direction of control and data. Control and data are divided into three signal groups:

1. FRAME#/IRDY#/CBE[3:0]#
2. DEVSEL#/TRDY#/STOP#
3. AD[31:0]

P_GNT# determines the FRAME#/IRDY#/CBE[3:0]# direction. If P_GNT# is asserted when FRAME# is active, address decode determines the DEVSEL#/TRDY#/STOP# direction. (Refer to Section 9, “Address Decoding.”) Address decode and slave location determine the AD[31:0] direction.

8.1 PCI TRANSACTION TYPES

Table 8-1 lists the PCI command codes and transaction types to which the PCI 6156 responds and initiates. The *Master* and *Target* columns indicate support for transactions wherein the PCI 6156 initiates transactions as a master, and responds to transactions as a target, respectively, on the primary and secondary buses.

As indicated in Table 8-1, the PCI 6156 does not support the following PCI commands—it ignores them and reacts to these commands as follows:

- ***Reserved***—PCI 6156 never initiates a PCI transaction with a ***reserved*** command code and, as a target, the PCI 6156 ignores ***reserved*** command codes.
- **Interrupt Acknowledge**—PCI 6156 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**—PCI 6156 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**—PCI 6156 does not generate Type 0 Configuration transactions on the primary interface, nor does it respond to Type 0 Configuration transactions on the secondary interface. *P-to-P Bridge r1.1* does ***not*** support configuration from the secondary bus.
- **Dual Address Cycles (DAC)**—PCI 6156 does not respond to, nor initiate, DAC transactions.

Table 8-1. PCI Transaction Types

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 – 0101b	***Reserved***	N	N	N	N
0110b	Memory Read	Y	Y	Y	Y
0111b	Memory Write	Y	Y	Y	Y
1000b – 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) ***(Not Supported)***	Y	Y	Y	Y
1110b	Memory Read Line	Y	Y	Y	Y
1111b	Memory Write and Invalidate	Y	Y	Y	Y

8.2 SINGLE ADDRESS PHASE

A 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]#.

8.3 DEVICE SELECT (DEVSEL#) GENERATION

The PCI 6156 performs positive address decoding when accepting transactions on the primary or secondary bus. The PCI 6156 never subtractively decodes. Medium DEVSEL# timing is used for both interfaces.

8.4 DATA PHASE

The Address phase, or phases of a PCI transaction, are followed by one or more Data phases. A Data phase is completed when IRDY# and either TRDY# or STOP# are asserted. Data transfer occurs only when both IRDY# and TRDY# are asserted during the same PCI clock cycle. The last Data phase of a transaction is indicated when FRAME# is de-asserted and both TRDY# and IRDY# are asserted, or when IRDY# and STOP# are asserted.

8.5 POSTED WRITE TRANSACTIONS

When the PCI 6156 determines that a Memory Write transaction is to be forwarded across the bridge, the PCI 6156 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 6156 can accept one 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 6156 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 6156 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 6156 requests ownership of the target bus. This can occur while the PCI 6156 is receiving data on the initiator bus. After the PCI 6156 has ownership of the target bus, and the target bus is detected in the idle condition, the PCI 6156 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 6156 can drive one Dword of Write data each PCI Clock cycle. If Write data is flowing through the PCI 6156 and the initiator stalls, the PCI 6156 inserts wait states on the target bus if the queue empties.

The PCI 6156 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 6156 starts another transaction to deliver the remaining Write data)
- Target returns a Target Abort (the PCI 6156 discards remaining Write data)

The Master Latency Timer expires, and the PCI 6156 no longer retains the target bus grant (the PCI 6156 starts another transaction to deliver the remaining Write data).

8.6 WRITE TRANSACTIONS

Acting as a PCI Bus extender, the PCI 6156 responds differently according to the address and initiator. The following examples delineate this process:

- **Primary Master Access Device on Primary Bus**—PCI 6156 forwards all PCI signals from the primary-to-secondary bus to allow any device thereon to track the PCI Bus.
- **Primary Master Access Device on Secondary Bus**—PCI 6156 forwards address, command, data, Byte Enables, and P_IRDY# to the secondary bus, while forwarding S_DEVSEL#, S_TRDY#, and S_STOP# to the primary bus.
- **Secondary Master Access Device on Secondary Bus**—PCI 6156 forwards all PCI signals from the secondary-to-primary bus to allow any device thereon to track the PCI bus.
- **Secondary Master Access Device on Primary Bus**—PCI 6156 forwards address, command, data, Byte Enables, and S_IRDY# to the primary bus, while forwarding P_DEVSEL#, P_TRDY#, and P_STOP# to the secondary bus.

8.7 READ TRANSACTIONS

The PCI 6156 responds according to the address and initiator of the Read command. The following examples delineate this process:

- **Primary Master Access Device on Primary Bus**—PCI 6156 does ***not*** forward PCI signals from the primary-to-secondary bus.
- **Primary Master Access Device on Secondary Bus**—PCI 6156 forwards address, command, Byte Enables, and P_IRDY# to the secondary bus, while forwarding data, S_DEVSEL#, S_TRDY#, and S_STOP# to primary bus.
- **Secondary Master Access Device on Secondary Bus**—PCI 6156 does ***not*** forward PCI signals from the secondary-to-primary bus, except in the case of dummy arbitration.
- **Secondary Master Access Device on Primary Bus**—PCI 6156 forwards address, command, Byte Enables, and S_IRDY# to the primary bus, while forwarding data, P_DEVSEL#, P_TRDY#, and P_STOP# to the secondary bus.

There is no buffer inside the PCI 6156 to accommodate reads.

8.8 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 6156 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 multifunction 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.

8.8.1 Type 0 Access to PCI 6156

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 6156 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 00b
- P_IDSEL must be asserted
- Function Code is 0

The PCI 6156 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 6156 ignores all Type 0 transactions initiated on the secondary interface.

8.8.2 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 6156 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 6156 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 6156 generates a Type 0 transaction only on the secondary bus, and never on the primary bus).

The PCI 6156 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 6156 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 8-2 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 6156 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 6156 uses the mapping presented in Table 8-2.

The PCI 6156 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 6156 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.

Table 8-2. 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
Ah	01010b	0000_0100_0000_0000b	26
Bh	01011b	0000_1000_0000_0000b	27
Ch	01100b	0001_0000_0000_0000b	28
Dh	01101b	0010_0000_0000_0000b	29
Eh	01110b	0100_0000_0000_0000b	30
Fh	01111b	1000_0000_0000_0000b	31
1Fh	11111b	Generate Special Cycle (P_AD[7:2] = 00h) 0000_0000_0000_0000b (P_AD[7:2] ≠ 00h)	None

8.8.3 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 6156 detects a Type 1 Configuration transaction intended for a PCI Bus downstream from the secondary bus, the PCI 6156 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 6156 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

The PCI 6156 forwards Type 1-to-Type 1 Configuration Write transactions as Delayed transactions, limited to a single Data transfer.

8.8.4 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 either the downstream or upstream direction.

The PCI 6156 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 6156 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 6156 responds with TRDY# to the next attempt of the Configuration transaction from the initiator. If more than one Data transfer is requested, the PCI 6156 responds with a Target Disconnect operation during the first Data phase.

8.9 TRANSACTION TERMINATION

This subsection describes how the PCI 6156 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)**—DEVSEL# and STOP# 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.

8.9.1 PCI 6156-Initiated Master Termination

As an initiator, the PCI 6156 uses normal termination if DEVSEL# is returned by the target within five Clock cycles of PCI 6156 FRAME# assertion on the target bus. The PCI 6156 terminates a transaction when the target terminates the transaction with last data transfer, Retry, Disconnect, or Target Abort.

8.9.2 Master Abort Received by PCI 6156

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 6156 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 6156 is able to reflect the Master Abort condition to the initiator. When the PCI 6156 detects a Master Abort in response to a Delayed transaction, and the initiator repeats the transaction, the PCI 6156 does not respond to the transaction with DEVSEL#. This returns the Master Abort condition to the initiator.

Note: *When the PCI 6156 performs a Type 1-to-Special Cycle translation, a Master Abort is the expected termination for the Special Cycle on the target bus. In this case, the Received Master Abort bit is not set (PCISR[13]=0; PCI:06h), and the Type 1 Configuration transaction is disconnected after the first Data phase.*

8.9.3 Target Termination Received by PCI 6156

When the PCI 6156 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 6156 controls these terminations using various methods, depending on the type of transaction performed.

8.9.3.1 Posted Write Target Termination Response

When the PCI 6156 initiates a Posted Write transaction, the Target Termination ***cannot*** be returned to the initiator. Table 8-3 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 6156 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 6156 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 6156 makes 2^{24} write attempts and fails to deliver all Posted Write data associated with that transaction, the PCI 6156 asserts P_SERR#, if enabled in the Command register, ***and*** the device-specific P_SERR# Disable bit for this condition is ***not*** set (PCICR[8]=0; PCI:04h). The Write data is discarded.

Table 8-3. 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).

8.9.3.2 Delayed Write Target Termination Response

When the PCI 6156 initiates a Delayed Write transaction, the type of Target Termination received from the target can be returned to the initiator. Table 8-4 delineates the response to each type of Target Termination that occurs during a Delayed Write transaction. The PCI 6156 repeats a Delayed Write transaction until it meets one of the following conditions:

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

The PCI 6156 makes 2^{24} write attempts (default), resulting in a response of Target Retry.

Table 8-4. Response to Delayed Write Target Termination

<table>
<tr><th>Target Termination</th><th colspan="4">Response</th></tr>
<tr><td>Normal</td><td colspan="4">Returns Disconnect to initiator with first Data transfer only if multiple Data phases are requested.</td></tr>
<tr><td>Target Retry</td><td colspan="4">Returns Target Retry to initiator. Continue write attempts to target.</td></tr>
<tr><td>Target Disconnect</td><td colspan="4">Returns Disconnect to initiator with first Data transfer only if multiple Data phases are requested.</td></tr>
<tr><td rowspan="4">Target Abort</td><td colspan="4">Returns Target Abort to initiator.
Sets target interface Status register Received Target Abort bit.
Sets initiator interface Status register Signaled Target Abort bit.</td></tr>
<tr><td colspan="2">Initiator</td><td colspan="2">Target</td></tr>
<tr><td>Primary</td><td>Secondary</td><td>Primary</td><td>Secondary</td></tr>
<tr><td>PCISR[11]=1;
PCI:06h</td><td>PCISR[12]=1;
PCI:06h</td><td>PCISSR[11]=1;
PCI:1Eh</td><td>PCISSR[12]=1;
PCI:1Eh</td></tr>
</table>

8.9.4 PCI 6156-Initiated Target Termination

The PCI 6156 can return a Target Retry, Disconnect, or Abort to an initiator for reasons other than detection of that condition at the target interface.

8.9.4.1 Target Retry

When it cannot accept Write data or return Read data as a result of internal conditions, the PCI 6156 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
  - Uses more than 16 clocks to accept this 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
  - Uses more than 16 clocks to accept this transaction

- **Posted Write Transactions**
  - Posted Write Data buffer does not contain sufficient space for the address

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.

8.9.4.2 Target Disconnect

The PCI 6156 returns a Target Disconnect to an initiator when the PCI 6156:

- Reaches an internal Address boundary
- Reaches a 4-KB boundary for a Posted Memory Write cycle
- Cannot accept further Write data
- When the target returns a Target Disconnect

8.9.4.3 Target Abort

The PCI 6156 returns a Target Abort to an initiator when the PCI 6156:

- 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 6156 sets the Status register Signaled Target Abort bit corresponding to the secondary or primary 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 6156 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 6156 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 32-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 6156 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 6156 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 6156 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 6156 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 6156 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 6156 supports 32-bit I/O addressing, which allows I/O addresses downstream from the PCI 6156 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 01h 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 01h 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 6156 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 6156 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 6156 uses to determine when to forward Memory commands. The PCI 6156 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 6156 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 6156 prefetches for all types of Memory Read commands in this Address space.

The PCI 6156 Prefetchable Memory Base and Limit Address registers define an Address range that the PCI 6156 uses to determine when to forward Memory transactions. The PCI 6156 forwards a Memory transaction from the primary-to-secondary interface, if the Transaction address falls within the Prefetchable Memory Address range. The PCI 6156 ignores Memory transactions initiated on the secondary interface that fall into this address range. The PCI 6156 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 6156 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 6156 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 6156 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 6156 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 6156 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 6156 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 6156 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 6156 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 6156, enable VGA mode by setting the Bridge Control register VGA Enable bit (BCNTRL[3]=1; PCI:3Eh). When operating in VGA mode, the PCI 6156 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 6156 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 6156 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 6156 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 6156 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 6156 claims VGA Palette Write transactions by asserting DEVSEL# in VGA Snoop mode.

When the VGA Palette Snoop Enable bit is set, the PCI 6156 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 both the VGA Enable and VGA Palette Snoop Enable bits are set (BCNTRL[3]=1 and PCICR[5]=1, respectively), the PCI 6156 behaves as if only the VGA Enable bit is set.*

10 TRANSACTION DELAY

This section describes transaction delay.

PCI 6156 transaction delay, from one interface to another, can be controlled through the appropriate Miscellaneous Control 2 register bits (MSCCNTRL2[5:4]; PCI:C4h). Primary-to-secondary transaction delay and secondary-to-primary transaction delay can be separately configured with a one or two PCI clock delay.

10.1 ONE-CLOCK LATENCY MODE

The PCI 6156 supports One-Clock Latency mode to minimize the delay. In this mode, there is strictly a one-clock delay between the originating FRAME# and bridge FRAME# signals. The PCI 6156 passes all Memory and I/O cycles, regardless of address. On the second Clock cycle, the PCI 6156 checks to which side of the bridge the cycle belongs. If it is determined to be the incorrect side, the PCI 6156 self-terminates its own cycle by asserting STOP#.

One-Clock Latency mode can be independently enabled for primary-to-secondary and secondary-to-primary transfers. It is primarily used to minimize the delay between P_GNT# assertion on the primary bus and P_FRAME# assertion resulting from the transaction initiated on the secondary bus. The PCI 6156 is designed to restrict S_GNT[9:0]#, based on P_GNT# assertion.

The control flow is as follows:

1. Primary bus asserts P_GNT# in response to P_REQ# asserted by the PCI 6156.
2. Secondary bus asserts S_GNT[9:0]# in response to P_GNT#.
3. Secondary master asserts S_FRAME# in response to S_GNT[9:0]#.
4. PCI 6156 passes through S_FRAME# to P_FRAME#.

There is a possibility of up to a four-clock delay between steps 1 to 4 by not using One-Clock Latency mode; however, this mode decreases the delay to three clocks.

10.2 TWO-CLOCK LATENCY MODE

If Two-Clock Latency mode is enabled (default), then the delay for step 4 in the previous section is two clocks instead of one. Also, the address is checked before FRAME# is passed to the other side of the bridge. All else remains the same.

11 ERROR HANDLING

This section provides detailed information regarding PCI 6156 error management. It also describes error status reporting and error operation disabling.

11.1 OVERVIEW

The PCI 6156 checks, forwards, and generates parity on the primary and secondary interfaces. To maintain transparency, the PCI 6156 tries to forward the existing parity condition from one bus to the other, along with address and data. The PCI 6156 always attempts to be transparent when reporting errors, but this is not always possible because of the presence of Posted data and Delayed transactions.

To support error reporting on the PCI Bus, the PCI 6156 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)

11.2 ADDRESS PARITY ERRORS

The PCI 6156 checks address parity for all Bus transactions, and Address and Bus commands.

When the PCI 6156 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 6156 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 6156 proceeds as usual and accepts the transaction if the transaction is directed to, or across, the PCI 6156.
2. PCI 6156 sets the Status register Parity Error Detected bit (PCISR[15]=1; PCI:06h).
3. PCI 6156 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).

When the PCI 6156 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 6156 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 6156 proceeds as usual and accepts the transaction if the transaction is directed to, or across, the PCI 6156.
2. PCI 6156 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 6156 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 6156 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 6156 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 6156 asserts P_PERR#. If the Parity Error Response Enable bit is ***not*** set, the PCI 6156 does ***not*** assert P_PERR#. In either case, the Configuration register is written.
2. PCI 6156 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 6156 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 6156 detects a Read Data Parity error on the secondary bus, the PCI 6156:

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 6156 detects a Read Data Parity error on the primary bus, the PCI 6156:

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 6156 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 6156 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 6156 detects the initiator's PERR# assertion while returning Read data to the initiator, the PCI 6156 takes no further action and completes the transaction as usual.

11.3.3 Posted Write Transactions

During downstream Posted Write transactions, when the PCI 6156 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 6156 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 6156:

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
   - Secondary interface Bridge Control register Parity Error Response Enable bit is set (BCNTRL[0]=1), and
   - PCI 6156 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 6156:

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 6156 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 6156 detects a Data Parity error during a Delayed Write transaction, it conditionally asserts PERR#. The PCI 6156 either passes or regenerates data parity to the target bus. A Parity error can occur:

- During the original Delayed Write Request transaction
- When the initiator repeats the Delayed Write Request transaction
- When the PCI 6156 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 6156 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 6156 asserts P_PERR# or S_PERR# two clocks after the data. The PCI 6156 always accepts the cycle, and can optionally pass the incorrect parity to the other bus, or regenerate the Parity bit on the other bus.
2. PCI 6156 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 6156 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 6156 treats the error as a new Delayed Write transaction.

11.4 DATA PARITY ERROR REPORTING SUMMARY

In the previous subsections, the PCI 6156 responses to Data Parity errors are presented according to transaction type in progress. This subsection organizes the PCI 6156 responses to Data Parity errors according to the Status bits set by the PCI 6156 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 6156 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 6156 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 6156 must be a master on the primary bus, and
- Primary interface Command register Parity Error Response Enable bit must be set (PCICR[6]=1)

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

- PCI 6156 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)

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

- PCI 6156 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 6156 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 6156 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 6156 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)

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_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
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 6156 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 6156 to assert P_SERR#, the Command register P_SERR# Enable bit must be set (PCICR[8]=1; PCI:04h)
- When the PCI 6156 asserts P_SERR#, the PCI 6156 must also set the Status register Signaled System Error bit (PCISR[14]=1; PCI:06h)

In compliance with *P-to-P Bridge r1.1*, the PCI 6156 asserts P_SERR# when it detects S_SERR# input asserted and the Bridge Control register S_SERR# Enable bit is set (BCNTRL[1]=1; PCI:3Eh). In addition, the PCI 6156 also sets the secondary Status register Signaled System Error bit (PCISSR(14]=1; PCI:1Eh).

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 PCI BUS ARBITRATION

This section describes primary and secondary bus arbitration.

12.1 OVERVIEW

The PCI 6156 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 6156 (typically located on the motherboard). For the secondary PCI Bus, the PCI 6156 has a built-in Internal Arbiter.

12.2 PRIMARY PCI BUS ARBITRATION

The PCI 6156 uses one Request output pin and one Grant input pin (P_REQ# and P_GNT#, respectively) for primary PCI Bus arbitration. The PCI 6156 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 6156 maintains P_REQ# assertion. However, if a Target Retry, Disconnect, or Abort is received in response to a PCI 6156-initiated transaction on the primary PCI Bus, the PCI 6156 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 6156 asserts P_REQ#, the PCI 6156 initiates a transaction on the primary bus on behalf of the secondary master.

12.3 SECONDARY PCI BUS ARBITRATION

The PCI 6156 implements a secondary PCI Bus Internal Arbiter, which supports up to ten external bus masters in addition to the PCI 6156.

12.3.1 Secondary Bus Arbitration Using Internal Arbiter

The PCI 6156 has ten secondary bus Request input and Grant output pins (S_REQ[9:0]# and S_GNT[9:0]#, respectively) to support external secondary bus masters.

The PCI 6156 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 (MSCCNTRL1[6, 4]; PCI:C1h).

12.3.2 Rotating-Priority Scheme

The secondary Arbiter supports a programmable two-level rotating algorithm that enables the ten request/grant pairs to control up to ten external bus masters. In addition, there is a request/grant pair internal to the PCI 6156, which allows the device to request and be granted access to the secondary bus. Figure 12-1 is an example of the Internal Arbiter wherein four masters, including the PCI 6156, 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 6156 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 6156 to the high-priority group. Therefore, by default, the PCI 6156 receives highest priority on the secondary

bus every other transaction and priority rotates evenly among the other masters.

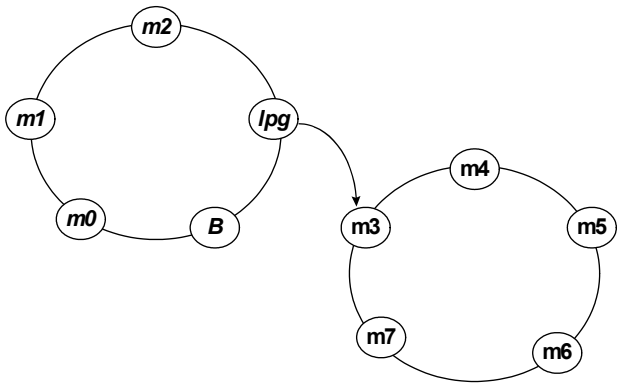


Figure 12-1. Secondary Bus Arbiter Example

Note: *In Figure 12-1, "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 6156 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.

12.3.3 Fixed-Priority Scheme

The PCI 6156 also supports a fixed-priority scheme within the low- and high-priority groups. In this case, the Miscellaneous Control 1 register controls whether the low- or high-priority group uses the fixed- or rotating-priority scheme (MSCCNTRL1[6, 4]; PCI:C1h). 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; PCI:C3h and MSCCNTRL1[7, 5]; PCI:C1h, respectively).

Using the example provided in Figure 12-1, but with the groups at fixed priority, suppose that:

- Master 7 (m7) has the highest priority of the low-priority group (IACNTRL[3:0]=0111b)
- PCI 6156 (B) has the highest priority of the high-priority group (IACNTRL[7:4]=1001b)
- Priority decreases in ascending order of masters for both groups (MSCCNTRL1[7, 5]=00b)

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

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

If MSCCNTRL1[7, 5]=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 SUPPORTED COMMANDS

This section discusses the PCI 6156 PCI command set.

13.1 PRIMARY INTERFACE COMMAND SET

Table 13-1 delineates the PCI 6156 primary interface command set.

Table 13-1. 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 13-1. 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).

13.2 SECONDARY INTERFACE COMMAND SET

Table 13-2 delineates the PCI 6156 secondary interface PCI command set.

Table 13-2. 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).

14 BRIDGE BEHAVIOR

This section presents various bridge behavior scenarios that occur when the target responds to a cycle generated by the PCI 6156, on behalf of the initiating master.

14.1 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 14-1 summarizes these possibilities, and delineates the PCI 6156 action for various cycle types.

After the PCI cycle is initiated, a target then has up to three cycles to respond before subtractive decoding 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 14-1. Bridge Actions for Various Cycle Types

Initiator	Target	PCI 6156 Response
Master on primary port	Target on the same primary port	Forwards all signals to the secondary port.
	Target on secondary port	Asserts P_DEVSEL#, then passes the cycle to the secondary port. When the cycle completes on the target port, the PCI 6156 waits for the initiator to 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	The PCI 6156 forwards all signals to the primary port. The PCI 6156 detects this situation by decoding the address, and monitoring S_DEVSEL# for other fast- and medium-speed devices on the secondary port.
	Target on primary or other secondary port	Asserts S_DEVSEL#, then passes the cycle to the appropriate port. When the cycle completes on the target port, the PCI 6156 waits for the initiator to end with normal termination.
	Target not on primary nor other secondary port	Does not respond.

14.2 ABNORMAL TERMINATION (BRIDGE MASTER-INITIATED)

14.2.1 Master Abort

A Master Abort indicates that the PCI 6156, 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#.

14.2.2 Address and Data Parity Errors Resulting in Master Abort

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.

14.3 PCI MASTER ON PRIMARY BUS

Table 14-2 delineates the PCI Control/Data path direction when a PCI transaction is initiated by a PCI master residing on the primary bus. The path direction guarantees cycle integrity, viewed from the primary and secondary buses.

The PCI 6156 is designed to pass most primary cycles to the secondary bus, except Configuration cycles in the one-clock delay case, as described in Table 14-2. The PCI 6156 performs Configuration Type 1-to-Type 0 conversion on the cycle with a matched Bus Number. It passes the Type 1 Configuration cycle, which is located on the secondary bus.

Table 14-2. PCI Control/Data Path Direction during PCI Master-Initiated PCI Transaction

Slave Location	Command	FRAME#/CBE[3:0]#/ IRDY#	DEVSEL#/TRDY# STOP#	AD[31:0]
Primary Bus	Read	Secondary -> Primary	Primary -> Secondary	Primary -> Secondary
	Write	Secondary -> Primary	Primary -> Secondary	Primary <- Secondary
Secondary Bus	Read	Secondary -> Primary	Primary <- Secondary	—
	Write	Secondary -> Primary	Primary <- Secondary	—

14.4 CONFIGURATION TYPE 1 TO-TYPE 0 CONVERSION

When a Type 1 Configuration cycle appears on the primary bus with a Bus Number equal to the PCI 6156 bridge Bus Number, the PCI 6156 performs a Type 1-to-Type 0 Conversion cycle.

The PCI 6156 first Retries all subsequent primary cycles, until their completion. It reverses the grant to the secondary bus to block the secondary master from accessing the bus. The Conversion cycle appears on the secondary bus, without being reflected to the primary bus.

The PCI 6156 then issues the converted Type 0 Configuration cycle on the secondary bus. The cycle termination can be normal, Master Abort, or Target Abort. In the case of a read, the PCI 6156 latches the data, and waits for the same Type 1 Configuration cycle on the primary bus.

14.5 CONFIGURATION TYPE 1-TO-TYPE 1 BY-PASSING

When a Type 1 Configuration cycle appears on the primary bus with a Bus Number greater than the PCI 6156 bridge Bus Number, but smaller than the secondary Subordinate Bus Number, the same Type 1 Configuration cycle appears on the secondary bus. Otherwise, the bypassing process is similar to the Type 1-to-Type 0 Conversion process. (Refer to Section 14.4.) The PCI 6156 internal state machine generates the secondary cycle, Retries all primary cycles, and blocks any secondary masters from accessing the secondary bus.

14.6 TYPE 0 CONFIGURATION CYCLE FILTER MODE

In Type 0 Configuration Cycle Filter mode, the PCI 6156 filters out all Primary Type 0 Configuration cycles by delaying passing of P_FRAME# by one PCI clock. In the case of a Type 1 Configuration cycle through the bridge, the PCI 6156 returns a Retry and relies on the internal state machine to generate Type 0 or Type 1 Configuration cycles on the secondary bus.

14.7 DECODING

The PCI 6156 uses a decoding circuit to determine the Slave device location.

During the memory cycle, the PCI 6156 uses Memory Base/Limit and Prefetch Memory Base/Limit registers. The Slave is on the secondary bus if one of the following conditions is met:

- PCIMBAR[15:0]; PCI:20h ≤Address ≤ PCIMLMT[15:0]; PCI:22h
- PCIPMBAR[15:0]; PCI:24h ≤Address ≤ PCIPMLMT[15:0]; PCI:26h
- VGA enabled (BCNTRL[3]=1; PCI:3Eh)—A_0000h ≤ Address ≤B_FFFFh

During the I/O cycle, the PCI 6156 uses the I/O Base/ Limit registers. The Slave device is on the secondary bus if one of the following conditions is met:

- PCIIOBAR[7:0]; PCI:1Ch ≤Address ≤ PCIIOLMT[7:0]; PCI:1Dh (refer to Note)
- VGA enabled (BCNTRL[3]=1; PCI:3Eh)— 3B0h ≤Address ≤3BBh
- VGA enabled (BCNTRL[3]=1; PCI:3Eh)— 3C0h ≤Address ≤3DFh

Note: *When ISA is enabled (BCNTRL[2]=1; PCI:3Eh), the I/O space between address 0h to 256h is always* ***reserved*** *at every 1-KB boundary.*

During the Type 1 Configuration cycle, the PCI 6156 uses the Device Bus Number and Subordinate Bus Number. The Slave device is on the secondary bus if one of the following conditions is met:

- PCIPBNO[7:0]; PCI:18h ≤Address ≤ PCISUBNO[7:0]; PCI:1Ah
- PCISBNO[7:0]; PCI:19h ≤Address ≤ PCISUBNO[7:0]; PCI:1Ah

All Type 0 Configuration cycles, Interrupt Acknowledge cycles, and the Special Cycle appearing on the primary bus are considered to have a Slave on the primary bus. Likewise, all similar cycles appearing on the secondary bus are considered to have a Slave on the secondary bus.

14.8 SECONDARY MASTER

The secondary master issues S_REQ[9:0]# to request the bus. The PCI 6156 generates P_REQ# on the primary bus. When P_GNT# is active, the PCI 6156 uses a round-robin algorithm to grant one secondary master using S_GNT[9:0]#.

Table 14-3 delineates the control/data path. The PCI 6156 passes all cycles from secondary-to-primary bus.

Table 14-3. Secondary Master Round-Robin Algorithm Control/Data Path

Slave Location	Read/Write	FRAME#/CBE[3:0]#/ IRDY#	DEVSEL#/TRDY# STOP#	AD[31:0]
Primary Bus	Read	Secondary -> Primary	Primary -> Secondary	Primary -> Secondary
	Write	Secondary -> Primary	Primary -> Secondary	Primary <- Secondary
Secondary Bus	Read	Secondary -> Primary	Primary <- Secondary	—
	Write	Secondary -> Primary	Primary <- Secondary	—

15 POWER MANAGEMENT

This section describes the Power Management feature.

15.1 OVERVIEW

The PCI 6156 incorporates functionality that meets the requirements of *PCI Power Mgmt. r1.0*. These features include:

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

15.2 POWER MANAGEMENT TRANSITIONS

Table 15-1 delineates the states and related actions the PCI 6156 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 15-1. States and Related Actions during Power Management Transitions

Current State	Next State	Action
D_0	D_1	Unimplemented power state. The PCI 6156 ignores the write to the Power State bits (power state remains at D_0, PMCSR[1:0]=00b; PCI:84h).
	D_2	
	D_{3hot}	***Reserved.***
	D_{3cold}	Power is removed from the PCI 6156. A power-up reset must be performed to bring the PCI 6156 to D_0.
D_{3hot}	D_0	The PCI 6156 performs an internal chip reset. S_RSTOUT# is ***not*** asserted. All registers are returned to the reset values and buffers are cleared.
	D_{3cold}	Power is removed from the PCI 6156. A power-up reset must be performed to bring the PCI 6156 to D_0.
D_{3cold}	D_0	Power-up reset. The PCI 6156 performs the standard power-up reset functions.

16 VPD

This section describes the VPD feature.

The PCI 6156 contains the Vital Product Data (VPD) registers, as specified in *PCI r2.2*. VPD information is stored in the serial EEPROM device, along with Autoload information.

The PCI 6156 provides storage of 224 bytes of VPD data in the serial EEPROM device.

The VPD register block is located at offsets A0h to A7h in PCI Configuration space. (Refer to Section 6.1.2.4, “VPD Capability.”) VPD also uses the Enhanced Capabilities Port Address mechanism.

17 MECHANICAL SPECS

This section provides the PCI 6156 mechanical dimensions and pinout.

The PCI 6156 is provided in two industry-standard packages—208-pin PQFP or 15 x 15 mm 160-pin (ball) TinyBGA.

17.1 208-PIN PQFP PACKAGE

This section provides the mechanical dimensions and pinout for the 208-pin PQFP package.

17.1.1 208-Pin PQFP Package Mechanical Dimensions

Figure 17-1 illustrates the PQFP package mechanical dimensions. Table 17-1 lists the mechanical dimensions, in millimeters, unless specified otherwise.

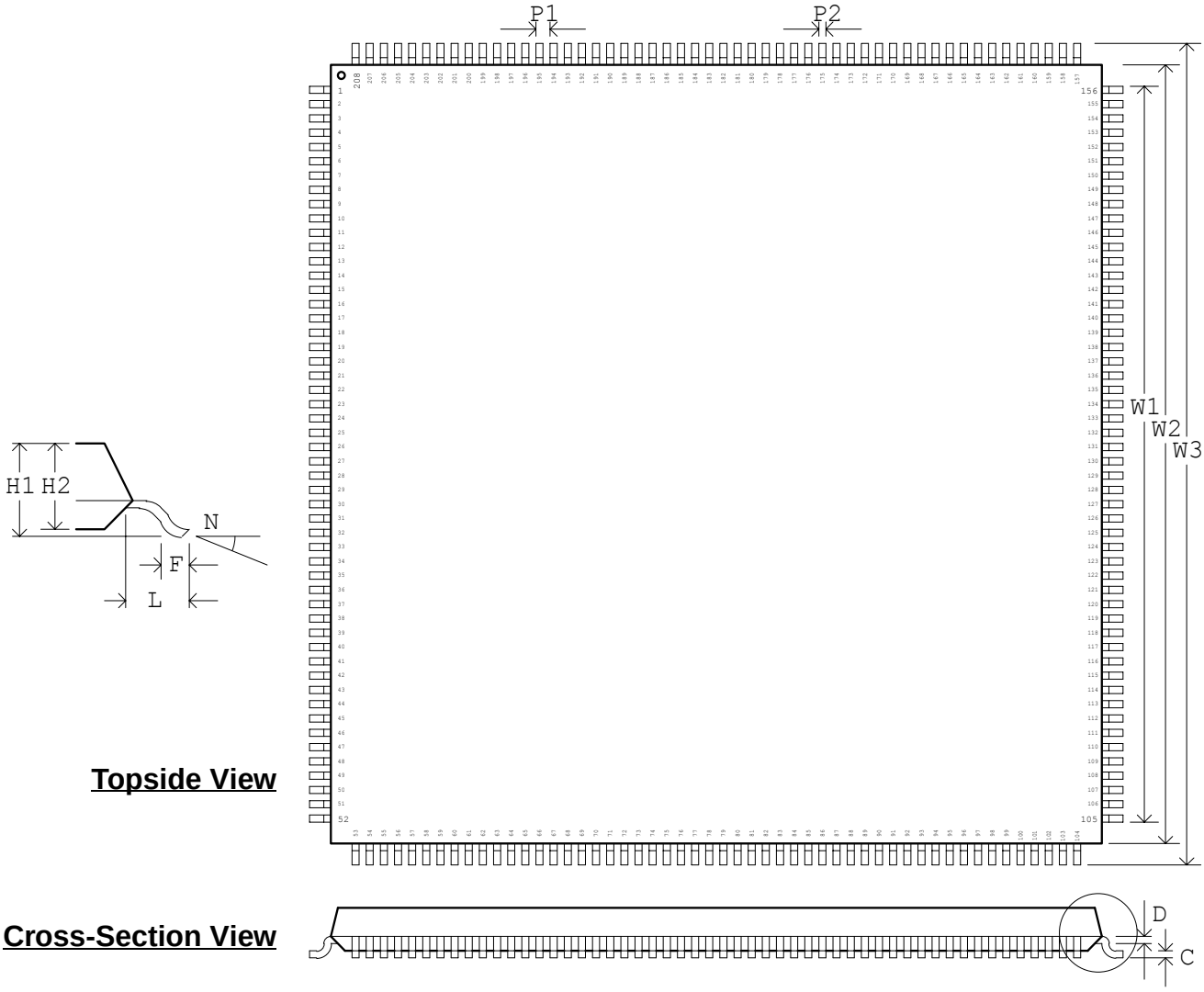


Figure 17-1. PCI 6156 208-Pin PQFP Mechanical Dimensions

Table 17-1. PCI 6156 208-Pin PQFP Mechanical Dimensions for Figure 17-1 Symbols (in Millimeters)

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

17.1.2 208-Pin PQFP Package Physical Layout with Pinout

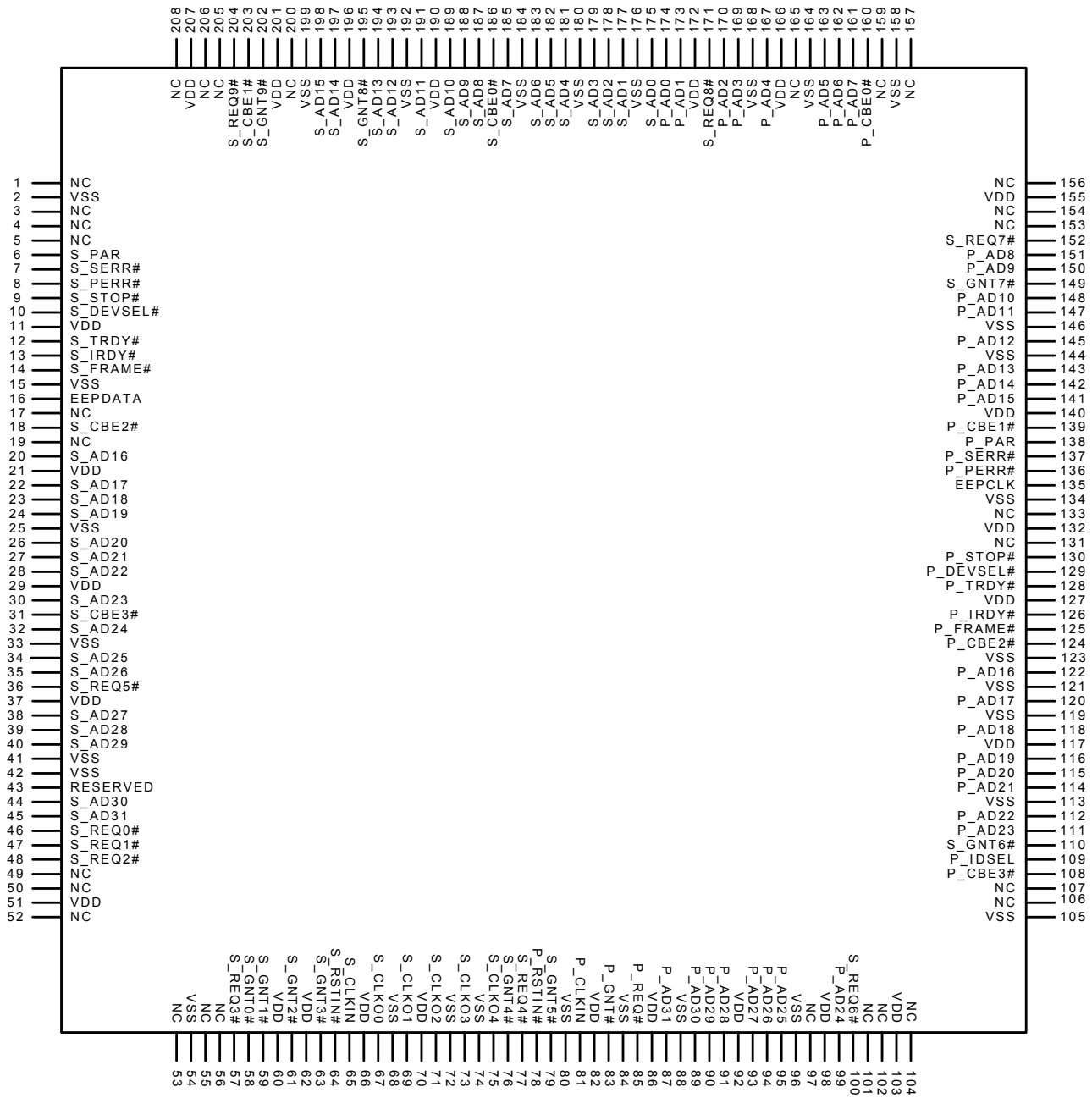


Figure 17-2. PCI 6156 208-Pin PQFP Top View

This page intentionally left blank.

17.2 160-PIN TINYBGA PACKAGE

This section provides the mechanical dimensions and pinout for the industry-standard 15 x 15 mm 160-pin (ball) TinyBGA package.

17.2.1 160-Pin TinyBGA Package Mechanical Dimensions

Figure 17-3 illustrates the TinyBGA package mechanical dimensions. Table 17-1 lists the mechanical dimensions, in millimeters, unless specified otherwise.

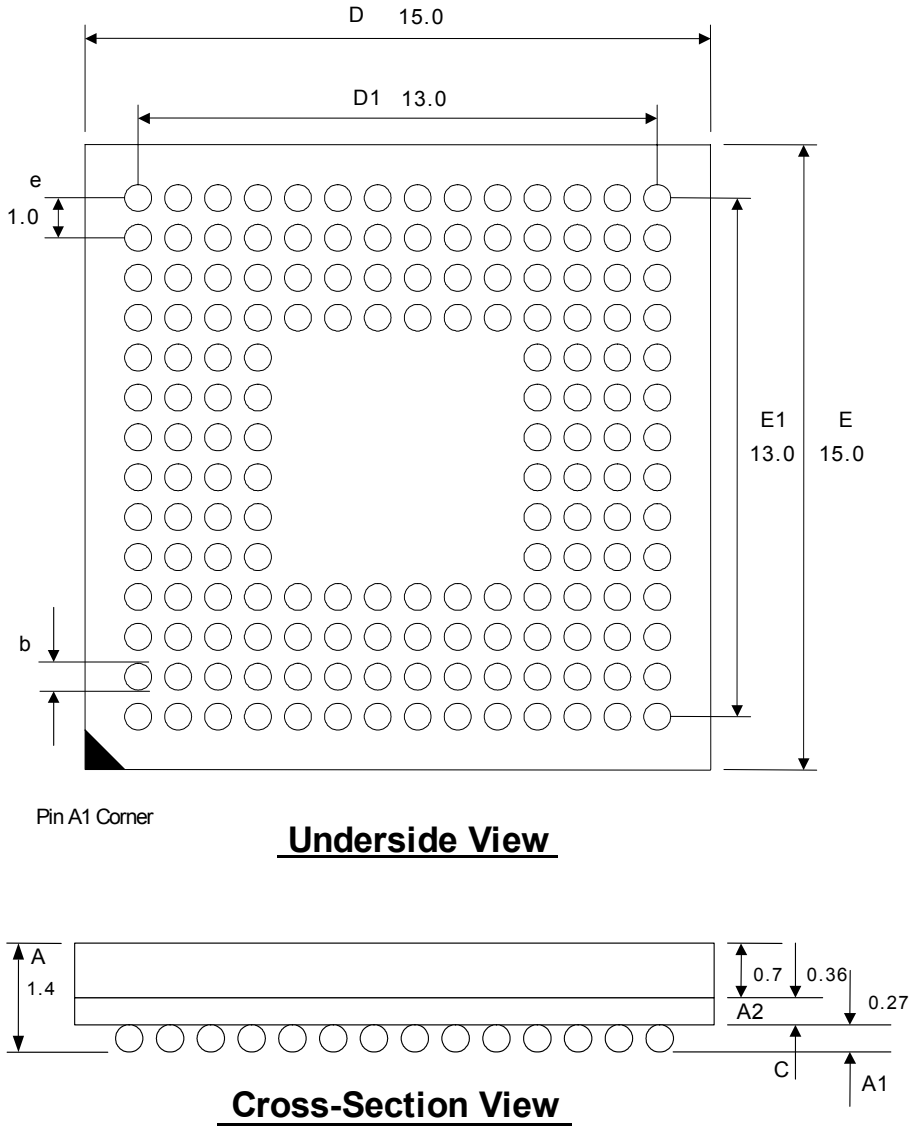


Figure 17-3. PCI 6156 160-Pin TinyBGA Mechanical Dimensions

Table 17-2. PCI 6156 160-Pin TinyBGA Mechanical Dimensions for Figure 17-3 Symbols (in Millimeters)

Symbol	Dimension	Minimum	Nominal	Maximum
A	Overall package height	—	—	1.4
A1	Package standoff height	—	0.27	—
A2	Encapsulation thickness	—	0.70	—
b	Ball diameter	0.35	0.40	0.45
C	Substrate thickness	—	0.36	—
e	Ball pitch	—	1.0	—
D, E	Overall package width	—	15.0	—
D1, E1	Overall encapsulation width	—	13.0	—

17.2.2 160-Pin TinyBGA Package Physical Layout with Pinout

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
A	S_REQ9#	S_CBE1#	S_AD15	S_AD13	S_AD10	S_CBE0#	S_AD5	S_AD4	S_AD1	P_AD1	P_AD3	P_AD6	P_CBE0#	S_REQ7#	A
B	S_PAR	VSS	S_GNT9#	S_GNT8#	S_AD11	S_AD8	S_AD6	S_AD3	S_AD0	S_REQ8#	P_AD4	P_AD7	VSS	P_AD8	B
C	S_PERR#	S_SERR#	VSS	S_AD14	S_AD12	S_AD9	S_AD7	S_AD2	P_AD0	P_AD2	P_AD5	VSS	P_AD9	S_GNT7#	C
D	S_TRDY#	S_DEVSEL#	S_STOP#	VSS	VSS	VDD	VDD	VDD	VDD	VSS	VSS	P_AD10	P_AD11	P_AD12	D
E	EEPDATA	S_FRAME#	S_IRDY#	VSS							VSS	P_AD13	P_AD14	P_AD15	E
F	S_AD17	S_AD16	S_CBE2#	VDD							VDD	P_CBE1#	P_PAR	P_SERR#	F
G	S_AD20	S_AD19	S_AD18	VDD							VDD	P_PERR#	EEPCLK	P_STOP#	G
H	S_AD21	S_AD22	S_AD23	VDD							VDD	P_IRDY#	P_TRDY#	P_DEVSEL#	H
J	S_CBE3#	S_AD24	S_AD25	VDD							VDD	P_AD16	P_CBE2#	P_FRAME#	J
K	S_AD26	S_REQ5#	S_AD27	VSS							VSS	P_AD19	P_AD18	P_AD17	K
L	S_AD28	S_AD29	S_AD30	VSS	VSS	VDD	VDD	VDD	VDD	VSS	VSS	P_AD22	P_AD21	P_AD20	L
M	S_AD31	S_REQ0#	VSS	S_GNT2#	S_CLKIN	S_CLKO1	S_CLKO4	P_CLKIN	P_REQ#	P_AD29	P_AD26	VSS	S_GNT6#	P_AD23	M
N	S_REQ1#	VSS	S_GNT0#	S_GNT3#	NC	S_CLKO2	S_GNT4#	S_GNT5#	P_GNT#	P_AD30	P_AD27	P_AD24	VSS	P_IDSEL	N
P	S_REQ2#	S_REQ3#	S_GNT1#	S_RSTOUT#	S_CLKO0	S_CLKO3	S_REQ4#	P_RSTIN#	NC	P_AD31	P_AD28	P_AD25	S_REQ6#	P_CBE3#	P
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	

Figure 17-4. PCI 6156 160-Pin TinyBGA Top View

18 ELECTRICAL SPECS

This section presents the PCI 6156 electrical specifications.

18.1 GENERAL ELECTRICAL SPECIFICATIONS

The ratings provided in this subsection are those above which the useful life of the PCI 6156 may be impaired.

Table 18-1 lists the PCI 6156 maximum ratings. Table 18-2 lists the PCI 6156 functional operating range. Table 18-3 lists the PCI 6156 DC electrical characteristics.

Caution: Stresses greater than the maximums listed in Table 18-1 cause permanent damage to the PCI 6156. This is a stress rating only and functional operation of the PCI 6156 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 18-1. 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	—	300 mW

Table 18-2. Functional Operating Range

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

Table 18-3. DC Electrical Characteristics

Symbol	Parameter	Condition	Minimum	Maximum	Unit	Notes
V_{DD}	V_{DD} Supply Voltage	—	3.0	3.6	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	$0 < V_{in} < V_{DD}$	—	±10	µA	—
C_{in}	Input Pin Capacitance	—	—	10.0	pF	—

18.2 PCI SIGNAL TIMING SPECIFICATION

Figure 18-1 illustrates the PCI 6156 signal timing specifications. Table 18-4 delineates the minimum and maximum values, for the symbols that appear in Figure 18-1.

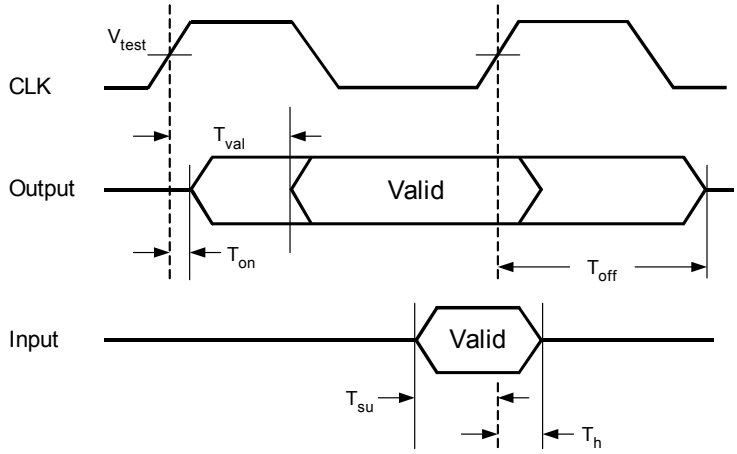


Figure 18-1. PCI Signal Timing Specification

Table 18-4. 66 MHz PCI Signal Timing for Figure 18-1

Symbol	Parameter	Minimum	Maximum	Symbol	Parameter	Minimum	Maximum
T_{val}	CLK to Signal Valid Delay— Bused Signals	2 ns	11 ns	T_{su}	Input Setup Time to CLK— Bused signals	7 ns	—
$T_{val(ptp)}$	CLK to Signal Valid Delay— Point to Point	2 ns	12 ns	$T_{su(ptp)}$	Input Setup Time to CLK— Point to Point	10, 12 ns	—
T_{on}	Float to Active Delay	2 ns	—	T_h	Input Signal Hold Time from CLK	0 ns	—
T_{off}	Active to Float Delay	—	28 ns	V_{test}	Voltage Test	—	0.4 V

A GENERAL INFORMATION

The PCI 6156 is built upon the powerful PLX PCI-to-PCI Bridge Architecture. It addresses the needs of the surveillance market by offering a system-level performance advantage. In particular, the PCI 6156 can support up to ten video capture chips by providing ten pairs of PCI request and grant lines.

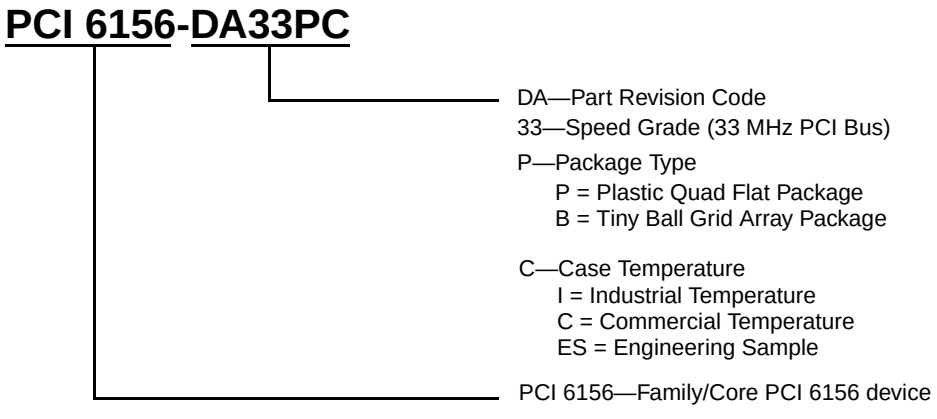
The following is a brief summary of its features and applications:

- Ideal architecture for video capture applications
- *PCI r2.2* with VPD support
- Lowest power bridge, supporting up to ten PCI masters
- Synchronous primary and secondary PCI Bus operation
- High performance, No Retry penalty flow with uninterrupted zero wait state burst up to 1 KB
- Concurrent primary and secondary port operation supports traffic isolation
- Provides programmable arbitration support for ten bus masters on secondary interface
- Five buffered secondary PCI clock outputs
- Enhanced address decoding
  - Supports 32-bit I/O Address range
  - Supports 64-bit Memory Address range
  - ISA-Aware mode for legacy support in the first 64 KB of I/O Address range
  - VGA addressing and VGA palette snooping support
- Supports 3.3V PCI with 5V tolerant I/O
- Industry-standard 208-pin Plastic Quad Flat Pack (PQFP) and 160-pin (ball) Tiny Ball Grid Array (TinyBGA) packages

A.1 PACKAGE ORDERING

Table A-1. Available Package

Package	Ordering Part Number
208-pin PQFP	PCI6156-DA33PC
160-pin (ball) TinyBGA	PCI6156-DA33BC



A.2 UNITED STATES AND INTERNATIONAL REPRESENTATIVES, AND DISTRIBUTORS

A list of PLX Technology, Inc., representatives and distributors can be found at http://www.plxtech.com.

A.3 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.

INDEX

R

S

T

V