TEXAS INSTRUMENTS

PCI6050

Serial PCI-to-PCI Bridge

Data Manual

June 2000

MSDS Bus Solutions

SLLS434A

IMPORTANT NOTICE

Texas Instruments and its subsidiaries (TI) reserve the right to make changes to their products or to discontinue any product or service without notice, and advise customers to obtain the latest version of relevant information to verify, before placing orders, that information being relied on is current and complete. All products are sold subject to the terms and conditions of sale supplied at the time of order acknowledgment, including those pertaining to warranty, patent infringement, and limitation of liability.

TI warrants performance of its semiconductor products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Customers are responsible for their applications using TI components.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards must be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance or customer product design. TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used. TI's publication of information regarding any third party's products or services does not constitute TI's approval, warranty or endorsement thereof.

Contents

Section	Title	Page
1	**Introduction**	**1–1**
1.1	Features	1–1
1.2	Related Documents	1–1
1.3	Trademarks	1–2
1.4	Ordering Information	1–2
2	**Terminal Descriptions**	**2–1**
3	**Feature/Protocol Descriptions**	**3–1**
3.1	Introduction to the PCI6050	3–1
3.1	PCI6050 System Diagram	3–2
3.2	Protocol Fundamentals	3–2
3.2.1	Packet Format	3–2
3.2.2	Start Up Procedure	3–3
3.2.3	Packet Summary	3–3
3.3	PCI Commands	3–4
3.4	Configuration Cycles	3–4
3.5	Secondary PCI Clocking	3–6
3.6	Special Cycle Generation	3–7
3.7	Bus Arbitration	3–7
3.7.1	Primary Bus Arbitration	3–7
3.7.2	Secondary Bus Arbitration	3–7
3.8	Decode Options	3–7
3.9	System Error Handling	3–7
3.9.1	Posted Write Parity Error	3–8
3.9.2	Posted Write Timeout	3–8
3.9.3	Target Abort on Posted Writes	3–8
3.9.4	Master Abort on Posted Writes	3–8
3.9.5	Master Delayed Write Timeout	3–8
3.9.6	Master Delayed Read Timeout	3–8
3.9.7	Secondary $\overline{\text{SERR}}$	3–8
3.10	Parity Handling and Parity Error Reporting	3–8
3.11	Master and Target Abort Handling	3–9
3.12	Discard Timer	3–9
3.13	Delayed Transactions	3–9
3.14	CompactPCI Hot-Swap Support	3–10
3.15	JTAG Support	3–10
3.16	GPIO Interface	3–14
3.17	PCI Power Management	3–15

3.17.1 Behavior in Low Power States 3–15
3.18 Serial ROM Implementation 3–15
4 Bridge Configuration Header 4–1
4.1 Vendor Register 4–2
4.2 Device ID Register 4–2
4.3 PCI Command Register 4–3
4.4 PCI Status Register 4–4
4.5 Revision ID Register 4–5
4.6 Class Code Register 4–5
4.7 Cache Line Size Register 4–5
4.8 Primary Latency Timer Register 4–6
4.9 Header Type Register 4–6
4.10 BIST Register 4–6
4.11 Primary Bus Number Register 4–7
4.12 Secondary Bus Number Register 4–7
4.13 Subordinate Bus Number Register 4–7
4.14 Secondary Bus Latency Timer Register 4–8
4.15 I/O Base Register 4–8
4.16 I/O Limit Register 4–8
4.17 Secondary Status Register 4–9
4.18 Memory Base Register 4–10
4.19 Memory Limit Register 4–10
4.20 Prefetchable Memory Base Register 4–10
4.21 Prefetchable Memory Limit Register 4–11
4.22 Prefetchable Base Upper 32 Bits Register 4–11
4.23 Prefetchable Limit Upper 32 Bits Register 4–12
4.24 I/O Base Upper 16 Bits Register 4–12
4.25 I/O Limit Upper 16 Bits Register 4–12
4.26 Capability Pointer Register 4–13
4.27 Interrupt Line Register 4–13
4.28 Interrupt Pin Register 4–13
4.29 Bridge Control Register 4–14
5 Extension Registers 5–1
5.1 Chip Control Register 5–1
5.2 Diagnostic Control Register 5–2
5.3 Arbiter Control Register 5–3
5.4 Primary $\overline{\text{SERR}}$ Event Disable Register 5–4
5.5 Primary GPIO Output Data Register 5–5
5.6 Primary GPIO Output Enable Register 5–5
5.7 Primary GPIO Input Data Register 5–6
5.8 Primary CLKOUT Control Register 5–7
5.9 Primary $\overline{\text{SERR}}$ Status Register 5–8
5.10 Secondary CLKOUT Control Register 5–9
5.11 Secondary $\overline{\text{SERR}}$ Status Register 5–10

5.12 General Purpose Event Register 5–11
5.13 Extended Diagnostic Status Register 5–12
5.14 Secondary GPIO Output Data Register 5–13
5.15 Secondary GPIO Output Enable Register 5–13
5.16 Secondary GPIO Input Data Register 5–14
5.17 Sequence Error Count Register 5–14
5.18 CRC Error Count Register 5–14
5.19 Receive Error Count Register 5–15
5.20 Transceiver Test Control and Status Register 5–15
5.21 PM Capability ID Register 5–16
5.22 PM Next Item Pointer Register 5–16
5.23 Power Management Capabilities Register 5–17
5.24 Power Management Control/Status Register 5–18
5.25 PMCSR Bridge Support Register 5–18
5.26 PM Data Register 5–19
5.27 HS Capability ID Register 5–19
5.28 HS Next Item Pointer Register 5–19
5.29 Primary CPCI Hot Swap Control and Status Register 5–20
5.30 Subsystem Vendor ID Register 5–20
5.31 Subsystem ID Register 5–21
5.32 FIFO BIST Register 5–21

List of Illustrations

Figure *Title* *Page*
2–1 PCI6050 Terminal Diagram 2–1
3–1 The PCI-to-PCI Bridge Implementation With PCI6050 Device 3–1
3–2 PCI6050 System Diagram 3–2
3–3 Packet Format and Bit Ordering 3–3
3–4 AD[31–0] During Address Phase of a Type 0 Configuration Cycle 3–4
3–5 AD[31–0] During Address Phase of a Type 1 Configuration Cycle 3–5
3–6 Bus Hierarchy and Numbering 3–6
3–7 Secondary PCI Clock Block Diagram 3–6
3–8 Serial ROM Data Format 3–16

List of Tables

Table	*Title*	*Page*
2–1	Terminal Signal Names Sorted by Terminal Number	2–2
2–2	Terminal Signal Names Sorted Alphabetically to Terminal Number	2–3
2–3	PCI System	2–4
2–4	PCI Address and Data	2–4
2–5	PCI Interface Control	2–5
2–6	PCI Clock Distribution	2–6
2–7	Secondary PCI Arbitration and Control	2–6
2–8	PCI6060 Clock and Control Interface	2–7
2–9	PCI6060 Transmit Data Path	2–7
2–10	PCI6060 Receive Data Path	2–8
2–11	Miscellaneous Terminals	2–8
2–12	JTAG Interface Terminals	2–9
2–13	Power Supply	2–9
3–1	PCI Command Definition	3–4
3–2	AD[31–16] During Address Phase of a Type 0 Configuration Cycle	3–5
3–3	JTAG Instructions and Op Codes	3–10
3–4	Boundary Scan Pin Order	3–11
4–1	PCI Command Register Description	4–3
4–2	PCI Status Register Description	4–4
4–3	Secondary Status Register Description	4–9
4–4	Bridge Control Register Description	4–14
5–1	Chip Control Register Description	5–1
5–2	Diagnostic Control Register Description	5–2
5–3	Arbiter Control Register Description	5–3
5–4	Primary $\overline{\text{SERR}}$ Event Disable Register Description	5–4
5–5	Primary GPIO Output Data Register Description	5–5
5–6	Primary GPIO Output Enable Register Description	5–5
5–7	Primary GPIO Input Data Register Description	5–6
5–8	Primary CLKOUT Control Register Description	5–7
5–9	Primary $\overline{\text{SERR}}$ Status Register Description	5–8
5–10	Secondary CLKOUT Control Register Description	5–9
5–11	Secondary $\overline{\text{SERR}}$ Status Register Description	5–10
5–12	General Purpose Event Register Description	5–11
5–13	Extended Diagnostic Status Register Description	5–12
5–14	Secondary GPIO Output Data Register Description	5–13
5–15	Secondary GPIO Output Enable Register Description	5–13
5–16	Secondary GPIO Input Data Register Description	5–14

5–17 Transceiver Test Control and Status Register Description 5–15
5–18 Power Management Capabilities Register Description 5–17
5–19 Power Management Control/Status Register Description 5–18
5–20 PMCSR Bridge Support Register Description 5–18
5–21 Primary CPCI Hot Swap Control and Status Register Description 5–20
5–22 FIFO BIST Register Description 5–21

1 Introduction

The Texas Instruments PCI6050 symmetric serialized PCI-to-PCI bridge provides a high-performance connection path between two peripheral component interconnect (PCI) buses. Transactions occur between masters on one PCI bus and targets on another PCI bus, and the PCI6050 device allows bridged transactions to occur concurrently on both buses. The PCI6050 pipeline architecture supports burst-mode transfers to maximize data throughput, and the two bus traffic paths through the bridge act independently.

A PCI6050-based PCI-to-PCI bridge is compliant with the *PCI Local Bus Specification* and can be used to overcome trace length limitations by allowing PCI-to-PCI implementations to span across separate system boards, and extend the electrical loading limits of 10 devices per PCI bus and 1 PCI device per expansion slot by creating hierarchical buses. The PCI6050 device provides two-tier internal arbitration for up to nine secondary bus masters and may be implemented with an external secondary PCI bus arbiter.

The compact-PCI hot-swap extended PCI capability is provided, which makes the PCI6050 device an ideal solution for multifunction compact PCI cards, remote chassis-to-chassis interconnects, and the adaptation of single function cards for hot-swap compliance.

1.1 Features

The PCI6050 device supports the following features:

- Configurable for *PCI Bus Power Management Interface Specification*
- Provides CompactPCI hot-swap functionality
- 3.3-V PCI core logic with universal PCI interface compatible with 3.3-V and 5-V PCI signaling environments
- 2.5-V serial link core logic compatible with the PCI6060 transceiver
- Supports 32-bit, 33-MHz PCI buses
- Provides internal two-tier arbitration for up to nine secondary bus masters and supports an external secondary bus arbiter
- Performs burst data transfers with pipeline architecture to maximize data throughput in both directions
- Independent read and write buffers for each direction
- Allows up to three delayed transactions in both directions
- Provides 10 secondary PCI clock outputs
- Predictable latency per *PCI Local Bus Specification*
- Propagates bus locking
- Secondary bus is driven low during reset
- Provides VGA/palette memory and I/O, and subtractive decoding options
- Advanced submicron, low-power CMOS technology

1.2 Related Documents

- *Advanced Configuration and Power Interface (ACPI) Specification (Revision 1.0)*
- *PCI Local Bus Specification (Revision 2.2)*
- *PCI-to-PCI Bridge Architecture Specification (Revision 1.1)*
- *PCI Bus Power Management Interface Specification (Revision 1.1)*

- PICMG *Compact-PCI Hot Swap Specification (Revision 1.0)*
- Texas Instruments *PCI6060 Product Data Manual*

1.3 Trademarks

1.4 Ordering Information

DEVICE	NAME	VOLTAGE	PACKAGE
PCI6050	Serial PCI-to-PCI Bridge		208-Terminal QFP

2 Terminal Descriptions

Figure 2–1 illustrates the terminals in the PCI6050 device.

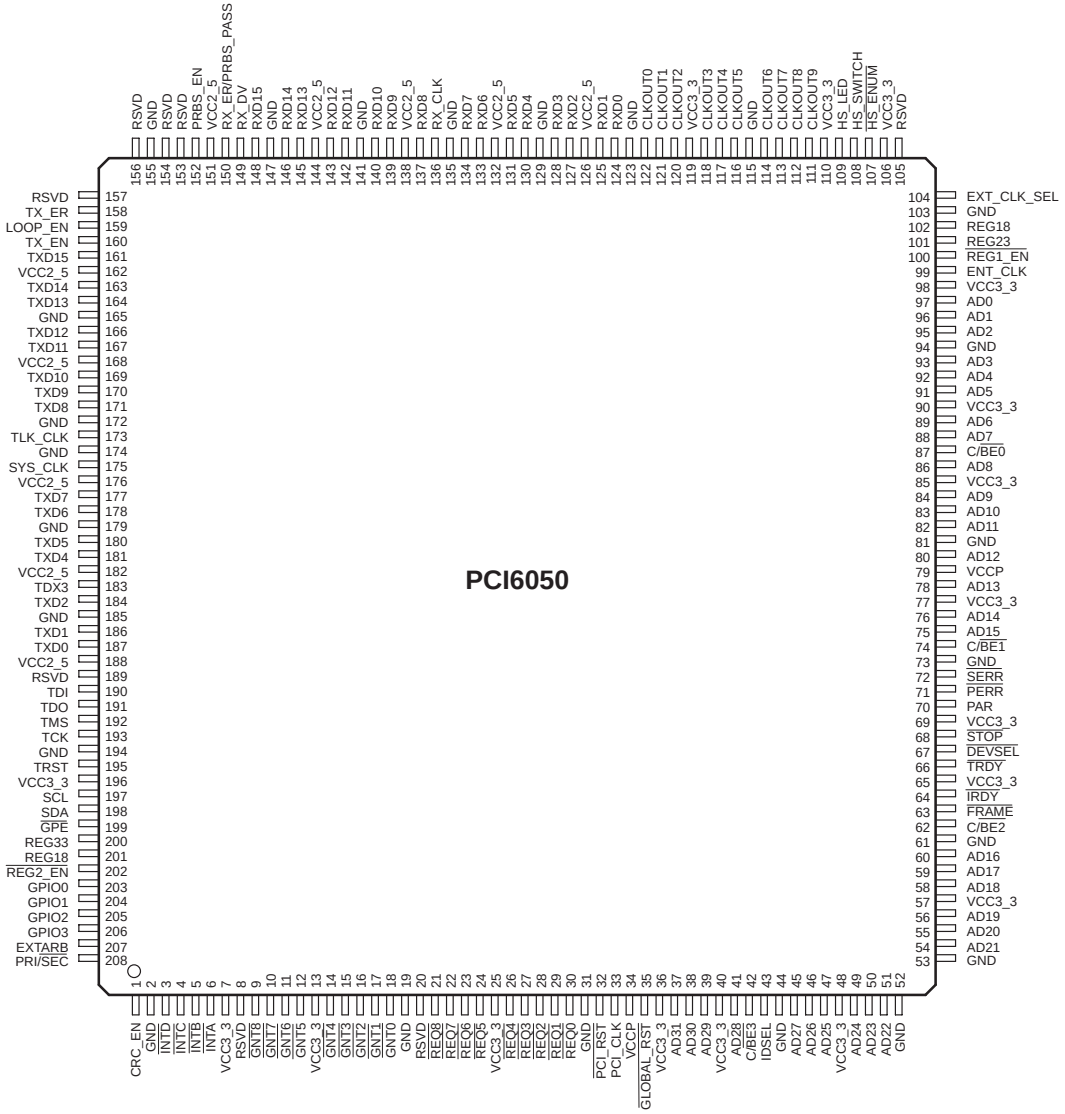

Figure 2–1. PCI6050 Terminal Diagram

Table 2–1 lists the PCI6050 device terminals in numeric order; Table 2–2 presents them in alphabetic order with their corresponding terminal number.

ADVANCE INFORMATION

Table 2–1. Terminal Signal Names Sorted by Terminal Number

NO.	TERMINAL NAME	NO.	TERMINAL NAME	NO.	TERMINAL NAME	NO.	TERMINAL NAME
1	CRC_EN	53	GND	105	RSVD	157	RSVD
2	GND	54	AD21	106	VCC3_3	158	TX_ER
3	$\overline{\text{INTD}}$	55	AD20	107	$\overline{\text{HS\_ENUM}}$	159	LOOP_EN
4	$\overline{\text{INTC}}$	56	AD19	108	HS_SWITCH	160	TX_EN
5	$\overline{\text{INTB}}$	57	VCC3_3	109	HS_LED	161	TXD15
6	$\overline{\text{INTA}}$	58	AD18	110	VCC3_3	162	VCC2_5
7	VCC3_3	59	AD17	111	CLKOUT9	163	TXD14
8	RSVD	60	AD16	112	CLKOUT8	164	TXD13
9	$\overline{\text{GNT8}}$	61	GND	113	CLKOUT7	165	GND
10	$\overline{\text{GNT7}}$	62	C/$\overline{\text{BE2}}$	114	CLKOUT6	166	TXD12
11	$\overline{\text{GNT6}}$	63	$\overline{\text{FRAME}}$	115	GND	167	TXD11
12	$\overline{\text{GNT5}}$	64	$\overline{\text{IRDY}}$	116	CLKOUT5	168	VCC2_5
13	VCC3_3	65	VCC3_3	117	CLKOUT4	169	TXD10
14	$\overline{\text{GNT4}}$	66	$\overline{\text{TRDY}}$	118	CLKOUT3	170	TXD9
15	$\overline{\text{GNT3}}$	67	$\overline{\text{DEVSEL}}$	119	VCC3_3	171	TXD8
16	$\overline{\text{GNT2}}$	68	$\overline{\text{STOP}}$	120	CLKOUT2	172	GND
17	$\overline{\text{GNT1}}$	69	VCC3_3	121	CLKOUT1	173	TLK_CLK
18	$\overline{\text{GNT0}}$	70	PAR	122	CLKOUT0	174	GND
19	GND	71	$\overline{\text{PERR}}$	123	GND	175	SYS_CLK
20	RSVD	72	$\overline{\text{SERR}}$	124	RXD0	176	VCC2_5
21	$\overline{\text{REQ8}}$	73	GND	125	RXD1	177	TXD7
22	$\overline{\text{REQ7}}$	74	C/$\overline{\text{BE1}}$	126	VCC2_5	178	TXD6
23	$\overline{\text{REQ6}}$	75	AD15	127	RXD2	179	GND
24	$\overline{\text{REQ5}}$	76	AD14	128	RXD3	180	TXD5
25	VCC3_3	77	VCC3_3	129	GND	181	TXD4
26	$\overline{\text{REQ4}}$	78	AD13	130	RXD4	182	VCC2_5
27	$\overline{\text{REQ3}}$	79	VCCP	131	RXD5	183	TXD3
28	$\overline{\text{REQ2}}$	80	AD12	132	VCC2_5	184	TXD2
29	$\overline{\text{REQ1}}$	81	GND	133	RXD6	185	GND
30	$\overline{\text{REQ0}}$	82	AD11	134	RXD7	186	TXD1
31	GND	83	AD10	135	GND	187	TXD0
32	$\overline{\text{PCI\_RST}}$	84	AD9	136	RX_CLK	188	VCC2_5
33	PCI_CLK	85	VCC3_3	137	RXD8	189	RSVD
34	VCCP	86	AD8	138	VCC2_5	190	TDI
35	$\overline{\text{GLOBAL\_RST}}$	87	C/$\overline{\text{BE0}}$	139	RXD9	191	TDO
36	VCC3_3	88	AD7	140	RXD10	192	TMS
37	AD31	89	AD6	141	GND	193	TCK
38	AD30	90	VCC3_3	142	RXD11	194	GND
39	AD29	91	AD5	143	RXD12	195	TRST
40	VCC3_3	92	AD4	144	VCC2_5	196	VCC3_3
41	AD28	93	AD3	145	RXD13	197	SCL
42	C/$\overline{\text{BE3}}$	94	GND	146	RXD14	198	SDA
43	IDSEL	95	AD2	147	GND	199	$\overline{\text{GPE}}$
44	GND	96	AD1	148	RXD15	200	REG33
45	AD27	97	AD0	149	RX_DV	201	REG18
46	AD26	98	VCC3_3	150	RX_ER/PRBS_PASS	202	$\overline{\text{REG2\_EN}}$
47	AD25	99	EXT_CLK	151	VCC2_5	203	GPIO0
48	VCC3_3	100	$\overline{\text{REG1\_EN}}$	152	PRBSEN	204	GPIO1
49	AD24	101	REG33	153	RSVD	205	GPIO2
50	AD23	102	REG18	154	RSVD	206	GPIO3
51	AD22	103	GND	155	GND	207	EXTARB
52	GND	104	EXT_CLK_SEL	156	RSVD	208	PRI/$\overline{\text{SEC}}$

Table 2–2. Terminal Signal Names Sorted Alphabetically to Terminal Number

TERMINAL NAME	NO.	TERMINAL NAME	NO.	TERMINAL NAME	NO.	TERMINAL NAME	NO.
AD0	97	GLOBAL_RST	35	PERR	71	TDO	191
AD1	96	GND	2	PRBSEN	152	TLK_CLK	173
AD2	95	GND	19	PRI/SEC	208	TMS	192
AD3	93	GND	31	REG1_EN	100	TRDY	66
AD4	92	GND	44	REG18	102	TRST	195
AD5	91	GND	52	REG18	201	TXD0	187
AD6	89	GND	53	REG2_EN	202	TXD1	186
AD7	88	GND	61	REG33	101	TXD2	184
AD8	86	GND	73	REG33	200	TXD3	183
AD9	84	GND	81	REQ0	30	TXD4	181
AD10	83	GND	94	REQ1	29	TXD5	180
AD11	82	GND	103	REQ2	28	TXD6	178
AD12	80	GND	115	REQ3	27	TXD7	177
AD13	78	GND	123	REQ4	26	TXD8	171
AD14	76	GND	129	REQ5	24	TXD9	170
AD15	75	GND	135	REQ6	23	TXD10	169
AD16	60	GND	141	REQ7	22	TXD11	167
AD17	59	GND	147	REQ8	21	TXD12	166
AD18	58	GND	155	RSVD	8	TXD13	164
AD19	56	GND	165	RSVD	20	TXD14	163
AD20	55	GND	172	RSVD	105	TXD15	161
AD21	54	GND	174	RSVD	153	TX_EN	160
AD22	51	GND	179	RSVD	154	TX_ER	158
AD23	50	GND	185	RSVD	156	VCCP	34
AD24	49	GND	194	RSVD	157	VCCP	79
AD25	47	GNT0	18	RSVD	189	VCC2_5	126
AD26	46	GNT1	17	RX_CLK	136	VCC2_5	132
AD27	45	GNT2	16	RXD0	124	VCC2_5	138
AD28	41	GNT3	15	RXD1	125	VCC2_5	144
AD29	39	GNT4	14	RXD2	127	VCC2_5	151
AD30	38	GNT5	12	RXD3	128	VCC2_5	162
AD31	37	GNT6	11	RXD4	130	VCC2_5	168
C/BE0	87	GNT7	10	RXD5	131	VCC2_5	176
C/BE1	74	GNT8	9	RXD6	133	VCC2_5	182
C/BE2	62	GPE	199	RXD7	134	VCC2_5	188
C/BE3	42	GPIO0	203	RXD8	137	VCC3_3	7
CLKOUT0	122	GPIO1	204	RXD9	139	VCC3_3	13
CLKOUT1	121	GPIO2	205	RXD10	140	VCC3_3	25
CLKOUT2	120	GPIO3	206	RXD11	142	VCC3_3	36
CLKOUT3	118	HS_ENUM	107	RXD12	143	VCC3_3	40
CLKOUT4	117	HS_LED	109	RXD13	145	VCC3_3	48
CLKOUT5	116	HS_SWITCH	108	RXD14	146	VCC3_3	57
CLKOUT6	114	IDSEL	43	RXD15	148	VCC3_3	65
CLKOUT7	113	INTA	6	RX_DV	149	VCC3_3	69
CLKOUT8	112	INTB	5	RX_ER/PRBS_PASS	150	VCC3_3	77
CLKOUT9	111	INTC	4	SCL	197	VCC3_3	85
CRC_EN	1	INTD	3	SDA	198	VCC3_3	90
DEVSEL	67	IRDY	64	SERR	72	VCC3_3	98
EXTARB	207	LOOP_EN	159	STOP	68	VCC3_3	106
EXT_CLK	99	PAR	70	SYS_CLK	175	VCC3_3	110
EXT_CLK_SEL	104	PCI_CLK	33	TCK	193	VCC3_3	119
FRAME	63	PCI_RST	32	TDI	190	VCC3_3	196

Table 2–3 lists the PCI system terminals and describes their function.

Table 2–3. PCI System

TERMINAL NAME	TERMINAL NO.	TYPE	FUNCTION
GLOBAL_RST	35	I	Global reset. GLOBAL_RST is used to initialize the PCI6050 device to its default state after power is applied. This reset must be supplied to both the primary and secondary nodes in order for the serial link to be established. Note that GLOBAL_RST serves as the only reset input to a secondary PCI6050 node.
PCI_CLK	33	I	PCI clock. PCI_CLK provides timing for all PCI transactions with a maximum frequency of 33 MHz. All PCI bus signals are sampled at rising edge of PCI_CLK.
PCI_RST	32	I/O	PCI reset. For a primary PCI6050 node, PCI_RST is an input that causes the bridge to put all output buffers in a high-impedance state and reset all internal registers. For a secondary PCI6050 node, PCI_RST is the output to the secondary PCI bus that initiates a bus reset.

Table 2–4 lists the PCI address and data terminals and describes their function.

Table 2–4. PCI Address and Data

TERMINAL NAME	TERMINAL NO.	TYPE	FUNCTION
AD31 AD30 AD29 AD28 AD27 AD26 AD25 AD24 AD23 AD22 AD21 AD20 AD19 AD18 AD17 AD16 AD15 AD14 AD13 AD12 AD11 AD10 AD9 AD8 AD7 AD6 AD5 AD4 AD3 AD2 AD1 AD0	37 38 39 41 45 46 47 49 50 51 54 55 56 58 59 60 75 76 78 80 82 83 84 86 88 89 91 92 93 95 96 97	I/O	PCI address and data bus. The AD[31–0] signals make up the multiplexed PCI address and data bus on the PCI interface. During the address phase of a PCI bus cycle, AD[31–0] contain a 32-bit address or other destination information. During each data phase of the transaction, AD[31–0] contain data.
C/BE3 C/BE2 C/BE1 C/BE0	42 62 74 87	I/O	PCI command and byte enables. During the address phase of a PCI bus cycle, C/ BE3–C/BE0 defines the bus command for the transaction. During the data phase, this 4-bit bus determines which byte lanes of the full 32-bit data bus carries meaningful data. C/BE0 applies to byte 0 (AD[7–0]), C/BE1 applies to byte 1 (AD[15–8]), C/BE2 applies to byte 2 (AD[23–16]), and C/BE3 applies to byte 3 (AD[31–24]).

Table 2–5 describes the PCI interface control signals.

Table 2–5. PCI Interface Control

TERMINAL NAME	TERMINAL NO.	TYPE	FUNCTION
$\overline{\text{DEVSEL}}$	67	I/O	PCI device select. $\overline{\text{DEVSEL}}$ is driven by the target of a PCI transaction to claim the cycle. As a PCI master, the PCI6050 device monitors $\overline{\text{DEVSEL}}$ until a target responds. If no target responds before the time-out condition occurs, then the bridge terminates the cycle with a master abort.
$\overline{\text{FRAME}}$	63	I/O	PCI cycle frame. $\overline{\text{FRAME}}$ is driven by the master of a PCI bus cycle. $\overline{\text{FRAME}}$ is asserted to indicate that a bus transaction is beginning, and data transfers continue while this signal is asserted. When $\overline{\text{FRAME}}$ is deasserted, the primary bus transaction is in the final data phase.
$\overline{\text{GNT0}}$	18	I/O	PCI bus grant. For a primary PCI6050 node, $\overline{\text{GNT0}}$ is the PCI bus grant input and is driven by the primary PCI bus arbiter to grant the bridge access to the primary PCI bus after the current data transaction has completed. For a secondary PCI6050 node, $\overline{\text{GNT0}}$ is an output from the internal bus arbiter to another secondary bus master. If the internal arbiter is disabled (external arbiter), then this terminal is the secondary PCI6050 PCI bus grant input.
IDSEL	43	I	PCI initialization and device select. IDSEL selects a primary PCI6050 node during configuration space accesses. IDSEL can be connected to one of the upper 24 PCI address lines on the primary PCI bus. This terminal is not used in secondary PCI6050 node applications and should be tied to a valid logic level. Note: The configuration space of the PCI6050 device can only be accessed from the primary bus.
$\overline{\text{INTD}}$ $\overline{\text{INTC}}$ $\overline{\text{INTB}}$ $\overline{\text{INTA}}$	3 4 5 6	I/O	PCI interrupts. The PCI6050 device provides PCI interrupt routing from the secondary PCI bus to the primary PCI bus. The INTx terminals are inputs to a secondary PCI6050 node, which are transmitted to the primary PCI6050 node by way of the serial link. The primary PCI6050 node provides the interrupt information to the host system by way of the INTx terminal outputs.
$\overline{\text{IRDY}}$	64	I/O	PCI initiator ready. $\overline{\text{IRDY}}$ indicates ability of the PCI bus master to complete the current data phase of the transaction. A data phase is completed on a rising edge of PCI_CLK where both $\overline{\text{IRDY}}$ and $\overline{\text{TRDY}}$ are asserted. Until $\overline{\text{IRDY}}$ and $\overline{\text{TRDY}}$ are both sampled asserted, wait states are inserted.
PAR	70	I/O	PCI parity. In all primary bus read and write cycles, the bridge calculates even parity across the AD and C/$\overline{\text{BE}}$ buses. As a master during PCI write cycles, the bridge outputs this parity indicator with one PCI_CLK delay. As a target during PCI read cycles, the calculated parity is compared to the parity indicator driven by the master; a miscompare can result in a parity error assertion ($\overline{\text{PERR}}$).
$\overline{\text{PERR}}$	71	I/O	PCI parity error. $\overline{\text{PERR}}$ is driven by a primary PCI6050 node when enabled through the command register to indicate that the calculated parity does not match the sampled PAR or that $\overline{\text{PERR}}$ was detected by the secondary PCI6050 node. For a secondary PCI6050 node, $\overline{\text{PERR}}$ is an input asserted by a secondary bus PCI device to indicate that calculated parity does not match the sampled PAR.
$\overline{\text{REQ0}}$	30	I/O	PCI bus request. For a primary PCI6050 node, $\overline{\text{REQ0}}$ is the PCI bus request output and is driven by the PCI6050 device to request access to the primary PCI bus after the current data transaction has completed. For a secondary PCI6050 node, $\overline{\text{REQ0}}$ is an input to the internal bus arbiter from another secondary bus master. If the internal arbiter is disabled (external arbiter), then this terminal is the secondary PCI6050 PCI bus request output.
$\overline{\text{SERR}}$	72	I/O	PCI system error. $\overline{\text{SERR}}$ is driven by a primary PCI6050 node to indicate that a critical system error has occurred or that $\overline{\text{SERR}}$ was detected on one of the subordinate buses downstream from the secondary PCI6050 node. The assertion need not be from the target of the primary PCI cycle. For a secondary PCI6050 node, $\overline{\text{SERR}}$ is an input asserted by a secondary bus PCI device to indicate that a critical system error has occurred.
$\overline{\text{STOP}}$	68	I/O	PCI cycle stop. $\overline{\text{STOP}}$ is driven by a PCI target to request that the initiator stop the current PCI bus transaction. This signal is used for target disconnects and is commonly asserted by target devices which do not support multiple data transfers.
$\overline{\text{TRDY}}$	66	I/O	PCI target ready. $\overline{\text{TRDY}}$ indicates the ability of the PCI bus target to complete the current data phase of the transaction. A data phase is completed upon a rising edge of PCI_CLK where both $\overline{\text{IRDY}}$ and $\overline{\text{TRDY}}$ are asserted. Until both $\overline{\text{IRDY}}$ and $\overline{\text{TRDY}}$ are both sampled asserted, wait states are inserted.

Table 2–6 describes the PCI clock signals; Table 2–7 describes the secondary PCI arbitration and control signals.

Table 2–6. PCI Clock Distribution

TERMINAL NAME	TERMINAL NO.	TYPE	FUNCTION
CLKOUT9 CLKOUT8 CLKOUT7 CLKOUT6 CLKOUT5 CLKOUT4 CLKOUT3 CLKOUT2 CLKOUT1 CLKOUT0	111 112 113 114 116 117 118 120 121 122	O	PCI clocks outputs. CLKOUT[9–0] provide timing for all transactions on the secondary PCI bus. Each secondary bus device samples all secondary PCI signals at the rising edge of its corresponding CLKOUT input. If CLKOUT[9–0] are utilized as PCI clock sources, then CLKOUT9 is fed back to the PCI_CLK input. Note: The CLKOUT9 feedback trace should exceed the length of other CLKOUT traces by 2.5 inches.
EXT_CLK	99	I	External clock. EXT_CLK is an optional 33-MHz PCI clock input from an external oscillator or clock source. This input is multiplexed with the internal 31.25-MHz (125 MHz divided by 4) clock and may be used for driving the CLKOUT terminals. When unused, this terminal should be tied to a valid logic level.
EXT_CLK_SEL	104	I	External clock select. EXT_CLK_SEL is used to select between an external 33-MHz clock source (EXT_CLK) and the internal 31.25-MHz (125 MHz divided by 4) clock as the CLKOUT terminal source. When EXT_CLK_SEL is high, the external clock is used. When EXT_CLK_SEL is low, the internal clock is used to drive CLKOUT terminals.

Table 2–7. Secondary PCI Arbitration and Control

TERMINAL NAME	TERMINAL NO.	TYPE	FUNCTION
$\overline{\text{GNT8}}$ $\overline{\text{GNT7}}$ $\overline{\text{GNT6}}$ $\overline{\text{GNT5}}$ $\overline{\text{GNT4}}$ $\overline{\text{GNT3}}$ $\overline{\text{GNT2}}$ $\overline{\text{GNT1}}$	9 10 11 12 14 15 16 17	O	Secondary PCI grants. When the internal arbiter is used, $\overline{\text{GNT8}}$–$\overline{\text{GNT1}}$ are outputs from the secondary PCI6050 node to other secondary bus masters to indicate when they are permitted to take ownership of the bus. These terminals are not used in a primary PCI6050 node.
$\overline{\text{REQ8}}$ $\overline{\text{REQ7}}$ $\overline{\text{REQ6}}$ $\overline{\text{REQ5}}$ $\overline{\text{REQ4}}$ $\overline{\text{REQ3}}$ $\overline{\text{REQ2}}$ $\overline{\text{REQ1}}$	21 22 23 24 26 27 28 29	I	Secondary PCI requests. When the internal arbiter is used, $\overline{\text{REQ8}}$–$\overline{\text{REQ1}}$ are inputs from other secondary bus masters to indicate when they wish to take ownership of the bus. These terminals are not used in a primary PCI6050 node.
EXTARB	207	I	Secondary external arbiter. EXTARB indicates whether or not an external arbiter is used in a secondary PCI6050 application and has no usage in primary PCI6050 applications. When EXTARB is sampled high, the internal secondary bus arbiter is disabled. When EXTARB is low, the PCI6050 internal secondary bus arbiter is enabled.

Table 2–8 describes the PCI6060 device clock and control interface signals; Table 2–9 describes the transmit data signals for the PCI6060 device.

Table 2–8. PCI6060 Clock and Control Interface

TERMINAL NAME	TERMINAL NO.	TYPE	FUNCTION
SYS_CLK	175	I	125-MHz system clock. SYS_CLK is a 125-MHz clock input used to control the 16-bit transmit data path (TXD[15–0], TX_EN, and TX_ER) to the PCI6060 device. The PCI6050 device internally divides this clock by 4 to generate the 31.25-MHz clock which may be used for the CLKOUT[9–0] outputs.
TLK_CLK	173	O	125-MHz reference clock. This output clock is synchronous to the 16-bit transmit data path (TXD[15–0], TX_EN, and TX_ER) and should be used as the transmit clock source for the PCI6060 device.
LOOP_EN	159	O	Loop enable. This output from the PCI6050 device can be connected to the PCI6060 LOOPEN terminal in order to allow the PCI6050 device to enable or disable its internal loop-back path.
PRBSEN	152	O	Pseudo-random bit stream enable. This output from the PCI6050 device can be connected to the PCI6060 PRBSEN terminal in order to allow the PCI6050 device to enable or disable its internal pseudo-random bit stream generator for loopback testing.
RX_CLK	136	I	Receive clock. This 125-MHz clock is synchronous to the receive data path (RXD[15–0], RX_DV, and RX_ER) and is provided by the PCI6060 device.

Table 2–9. PCI6060 Transmit Data Path

TERMINAL NAME	TERMINAL NO.	TYPE	FUNCTION
TXD15	161	O	Transmit data. TXD[15–0] carries the 16-bit data stream from the PCI6050 device to the PCI6060 device.
TXD14	163		
TXD13	164		
TXD12	166		
TXD11	167		
TXD10	169		
TXD9	170		
TXD8	171		
TXD7	177		
TXD6	178		
TXD5	180		
TXD4	181		
TXD3	183		
TXD2	184		
TXD1	186		
TXD0	187		
TX_EN	160	O	Transmit data enable. TX_EN is driven by the PCI6050 device to indicate to the PCI6060 device when there is valid data on TXD[15–0]. This signal may also be used in conjunction with TX_ER to produce special control codes on the transceiver interface.
TX_ER	158	O	Transmit error. This signal is used in conjunction with TX_EN to produce special control codes on the transceiver interface.

Table 2–10 describes the PCI6060 device receive data signals. Table 2–11 describes miscellaneous terminals on the PCI6050 device.

Table 2–10. PCI6060 Receive Data Path

TERMINAL NAME	TERMINAL NO.	TYPE	FUNCTION
RXD15 RXD14 RXD13 RXD12 RXD11 RXD10 RXD9 RXD8 RXD7 RXD6 RXD5 RXD4 RXD3 RXD2 RXD1 RXD0	148 146 145 143 142 140 139 137 134 133 131 130 128 127 125 124	I	Receive data. RXD[15–0] carries the 16-bit data stream from the PCI6060 device to the PCI6050 device.
RX_DV	149	I	Receive data valid. RX_DV is driven by the PCI6060 device to indicate to the PCI6050 device when there is valid data on RXD[15–0]. This signal may also be used in conjunction with RX_ER to determine when there are special control codes on the receiver interface or when a receiver error has occurred.
RX_ER/ PRBS_PASS	150	I	Receive error/status of PRBS test. The RX_ER may be used along with RX_DV by the PCI6050 device to determine when a special control code or transmit error was received by the PCI6060 device. The PRBS_PASS input from the PCI6060 device indicates whether the internal loop-back of pseudo-random data passed or failed. A primary PCI6050 device provides a mechanism to enable and read the status of the PRBS loop-back test via the transceiver test control and status registers.

Table 2–11. Miscellaneous Terminals

TERMINAL NAME	TERMINAL NO.	TYPE	FUNCTION
CRC_EN	1	I	CRC enable. When CRC_EN is sampled high by a primary PCI6050 node, a 16-bit CRC is attached to each of the serial packets prior to its transmission over the serial link. The receiver compares this CRC with a calculated CRC in order to validate the incoming packet. When CRC_EN is sampled low, CRC-based packet validation is disabled.
PRI/$\overline{\text{SEC}}$	208	I	Mode select. PRI/$\overline{\text{SEC}}$ determines whether the PCI6050 device behaves as a primary or secondary serial PCI node. When sampled high, the PCI6050 device assumes the role of a primary node. When sampled low, the PCI6050 device acts as a secondary node.
$\overline{\text{GPE}}$	199	O	General purpose event. The PCI6050 device can be programmed to assert this output under various conditions that may require special service from an ACPI handler. This output provides open-drain fail-safe signaling.
GPIO3 GPIO2 GPIO1 GPIO0	206 205 204 203	I/O	General purpose inputs/output. GPIO[3–0] provide basic general-purpose input and output functionality programmable through the PCI6050 device, and are available on both a primary and secondary PCI6050 nodes.
$\overline{\text{HS\_ENUM}}$	107	I/O	Hot-swap enumeration. $\overline{\text{HS\_ENUM}}$ is an input to a secondary PCI6050 node and is communicated via the serial link to the primary PCI6050 node, which asserts this open-drain output accordingly.
HS_LED	109	O	Hot-swap LED. This output is controlled via the the HS_CSR register and is provided to indicate that a hot-swappable primary PCI6050 implementation may be safely removed. When $\overline{\text{PCI\_RST}}$ is asserted to the PCI6050 device, it drives this LED output high.
HS_SWITCH	108	I	Hot-swap handle switch. This input provides the status of the ejector handle state and may be read through the extended diagnostic status register.
SCL	197	I/O	Serial ROM clock. This terminal provides the serial ROM clock output from the PCI6050 device, and is sampled at reset to determine the presence of a serial ROM. Applications that do not support a serial ROM must place a weak pull-down on this terminal.
SDA	198	I/O	Serial ROM data. This terminal provides the serial ROM data signal, which is both an output to select the word address and an input for the data transfer to the PCI6050 device.

Table 2–12 describes the JTAG interface terminals, and Table 2–13 lists the power supply terminals for the PCI6050 device.

Table 2–12. JTAG Interface Terminals

TERMINAL NAME	TERMINAL NO.	TYPE	FUNCTION
TDI	190	I	JTAG serial data in. TDI is the serial input through which JTAG instructions and test data enter the JTAG interface. The new data on TDI is sampled on the rising edge of TCLK.
TDO	191	O	JTAG serial data out. TDO is the serial output through which test instructions and data from the test logic leave the PCI6050 device.
TMS	192	I	JTAG test mode select. TMS causes state transitions in the test access port controller.
TCK	193	I	JTAG boundary-scan clock. TCLK is the clock controlling the JTAG logic.
TRST	195	I	JTAG TAP reset. When TRST is asserted low, the TAP controller is asynchronously forced to enter a reset state and initialize the test logic.

Table 2–13. Power Supply

TERMINAL NAME	TERMINAL NO.	TYPE	FUNCTION
GND	2, 19, 31, 44, 52, 53, 61, 73, 81, 94, 103, 115, 123, 129, 135, 141, 147, 155, 165, 172, 174, 179, 185, 194	I	Device ground terminals
VCCP	34, 79	I	Primary bus-signaling environment supply. VCCP is used in protection circuitry on PCI bus I/O signals.
VCC2_5	126, 132, 138, 144, 151, 162, 168, 176, 182, 188	I	Power supply terminals for core logic (2.5 V)
VCC3_3	7, 13, 25, 36, 40, 48, 57, 65, 69, 77, 85, 90, 98, 106, 110, 119, 196	I	Power supply terminals for core logic (3.3 V)
$\overline{\text{REG2\_EN}}$ REG1_EN	202 100	I	Regulator enables. The REGx_EN terminals are used to enable internal voltage regulators and should be tied to a low logic level.
REG33	101, 200	I	Power supply terminals for internal voltage regulators (3.3 V)
REG18	102, 201	I/O	Regulated output terminals supplied by the PCI6050 device that require external bypass capacitors of at least 0.1 μF.

3 Feature/Protocol Descriptions

The Texas Instruments PCI6050 device is a symmetric serialized PCI-to-PCI bridge that provides a high-performance solution for connecting two PCI buses via a high-speed serial cable medium.

The PCI6050 device has two major interfaces: a PCI interface and a transceiver (PCI6060) interface. The PCI6050 device communicates with a PCI bus and through a generic parallel interface to the PCI6060 transceiver. Two separate PCI6050 nodes are required, each with its own PCI6060 serial link, for a complete system. This four-chip solution provides a single, transparent PCI-to-PCI bridge, as illustrated in Figure 3–1.

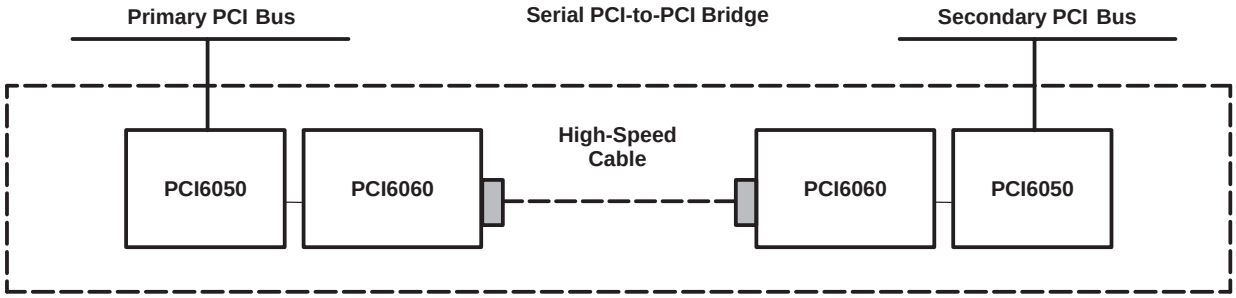

Figure 3–1. The PCI-to-PCI Bridge Implementation With PCI6050 Device

As a symmetrical device, the PCI6050 device can be implemented as either a primary or secondary serial PCI-to-PCI bridge node. The mode is determined by the level of the PRI/$\overline{\text{SEC}}$ input terminal. At a high level, a primary PCI6050 node claims downstream transactions that appear on the primary PCI bus which target either the PCI6050-based bridge or any devices located on a target bus that is subordinate to the bridge. These transactions are forwarded to the secondary PCI6050 node via the serial link so that they may be initiated and completed on the secondary PCI bus. In addition, a primary node initiates upstream transactions that are forwarded from the secondary PCI bus by the secondary node in a similar manner.

The PCI6050 device, a single-function PCI device, provides an internal PCI bus arbiter that supports up to nine secondary PCI bus masters when it is configured as a secondary node. The PCI6050 device also provides 10 CLKOUT outputs that may supplied to PCI devices by dividing the SYS_CLK input clock by 4 or through an external clock source up to a maximum of 33 MHz provided by the system. The PCI6050 device supports primary PCI clock frequencies up to a maximum of 33 MHz.

The PCI6050 serialized PCI-to-PCI bridge implementation provides parallel PCI interrupt routing from secondary devices to the primary system. In addition, the PCI6050 device supports CompactPCI hot swap on the primary interface by passing the CompactPCI hot-swap $\overline{\text{HS\_ENUM}}$ signal from the secondary bus to the primary interface, and it supports four general purpose inputs and outputs which operate and are controlled independently for each node. A two-wire serial interface is provided for register pre-loading and subsystem identification for both primary and secondary nodes.

3.1 PCI6050 System Diagram

Figure 3–2 illustrates a system diagram for a notebook/dock application using the PCI6050 device.

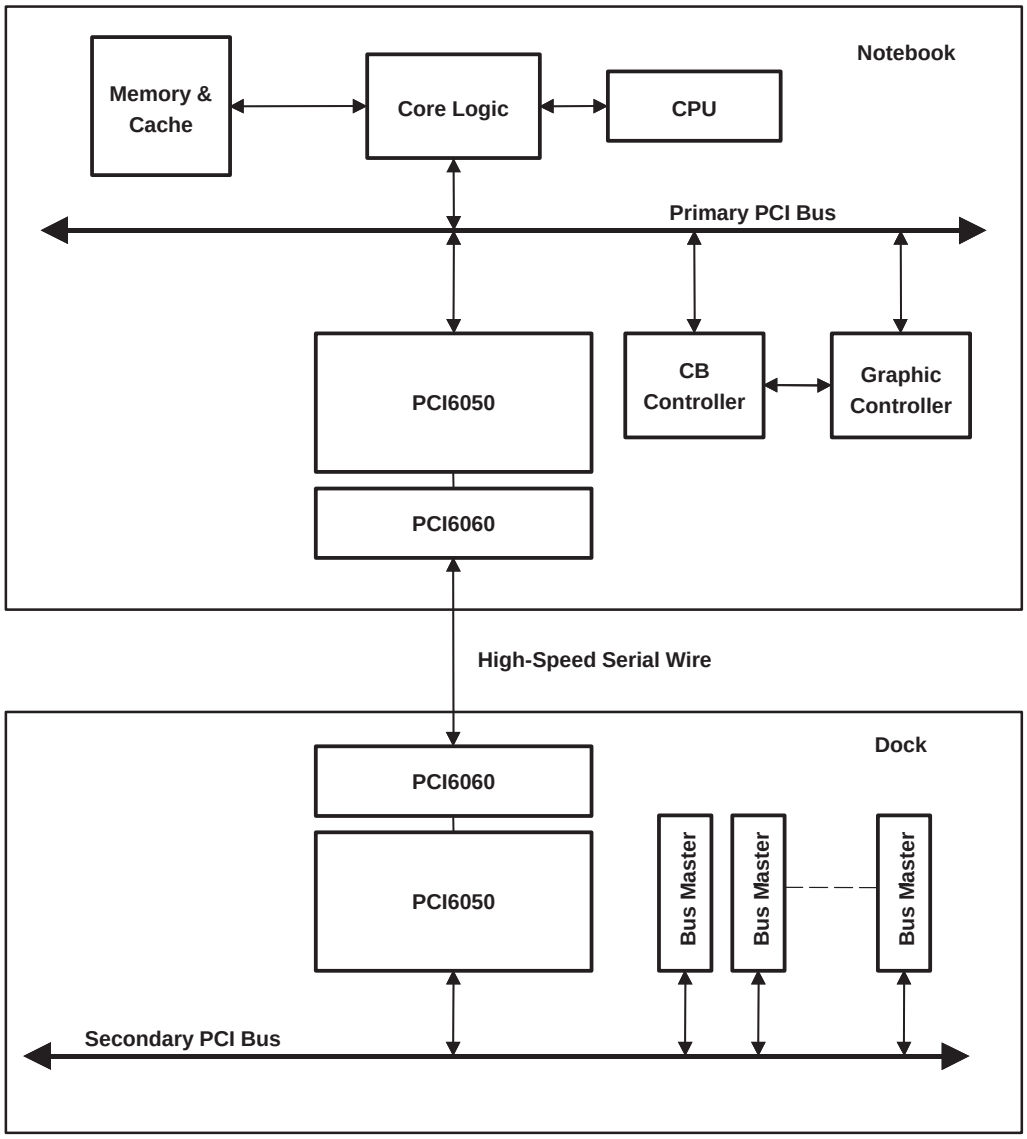

Figure 3–2. PCI6050 System Diagram

3.2 Protocol Fundamentals

The high-speed serial protocol defined here addresses a full duplex communication path with a continuous signaling rate for a point-to-point connection between two nodes. The protocol itself is not limited to any specific signaling rate, but the rate is constant ± transceiver tolerances and no speed scaling is built into the protocol. In the point-to-point connection, both nodes must implement the identical signaling rate within the transceiver tolerance.

The Texas Instruments PCI6060 transceiver provides the data rate and functionality that is ideal for the serial protocol, and the transceiver has an integrated 8B/10B encoder/decoder (2.5 GHz) to allow for clock recovery and provide some means of error detection across the cable.

3.2.1 Packet Format

The serial protocol uses 48-bit or 64-bit packets transmitted most significant bit (MSB) first. The 6 most significant bits of a packet make up the packet identifier, which includes 5 bits to indicate the packet type and a single bit to

indicate the source of the packet as being the primary or secondary node. The 16 least significant bits of a 64-bit packet contain a CRC value when CRC protection is enabled. The remaining bits contain data that specific to each valid packet type. Figure 3–3 illustrates the general packet format, and MSB as the leftmost bit is assumed for all diagrams in this specification unless otherwise noted.

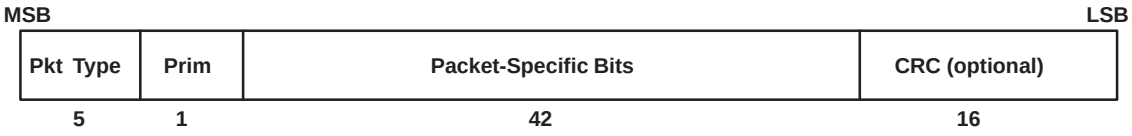

Figure 3–3. Packet Format and Bit Ordering

3.2.2 Start-Up Procedure

Before any PCI6050 node can transfer serialized PCI-to-PCI bridge packets, the devices on both ends of the serial link must be in a valid state and ready to transmit and receive the packets. A start-up handshaking mechanism is used to bring the serial link into a default state after power-up or when the link is established. During this startup process, a primary PCI6050 node acknowledges all type-0 configuration cycles by signaling a retry. All type-1 configuration cycles are terminated as indicated by the bridge control register (target abort or normal completion), and all memory and I/O cycles are terminated with a master abort. After successful completion of this startup phase, all PCI cycles are handled normally.

3.2.3 Packet Summary

This section provides a general overview of the various packets used in communication between the two PCI6050 nodes during normal operation. The following list comprises the basic set of functional packets that comprise the PCI6050 protocol. Because this represents a proprietary interface, it is beyond the scope of this data manual to fully describe the PCI6050 protocol here.

- PCI address – A PCI address packet supplies the command and address information for a given PCI transaction as well as the parity for the address phase of the transaction. In addition, this packet provides a sequence field, which is used to optimize FIFO performance and maintain the pipeline.
- PCI data – A PCI data packet supplies the byte-enable flags and transaction data information for a given PCI transaction as well as the parity for the current data phase of the transaction. In addition, this packet provides a sequence field, which is used to optimize FIFO performance and maintain the pipeline.
- Configuration register update packet – A configuration register update packet is used to synchronize the PCI configuration registers on both PCI6050 nodes. Both PCI6050 nodes have knowledge of registers and bits that need to be communicated through the serial link when configuration transactions or changes in the system status occur. These update packets are sent continuously until the destination node returns a value matching the source contents to confirm receipt.
- Completion status – Completion status packets are used to communicate the status of pending delayed transactions, including an indication if the transaction was terminated with a master abort, target abort, transaction timeout, or other erroneous condition.
- FIFO confirmation – FIFO confirmation packets are used to communicate the state of the pipeline, confirm a correctly received packet, or to notify the source node of any error that occurred during the packet transfer. This packet is also used to communicate error conditions such as a receiver error, CRC error, or sequence error condition to indicate that the packet must be retransmitted.
- Ordering tag – The ordering tag packet is used to notify the FIFO structures within the receiving node that a transaction (such as a memory write) is pending, and that the completion of that transaction should precede any other pending transactions (such as memory reads). This allows a PCI6050-based bridge to maintain the ordering rules within the PCI-to-PCI bridge specification.

3.3 PCI Commands

The bridge responds to PCI bus cycles as a PCI target device based on the decoding of each address phase and internal register settings. Table 3–1 lists the valid PCI bus cycles and their encoding on the command/byte enables (C/$\overline{BE}$) bus during the address phase of a bus cycle.

Table 3–1. PCI Command Definition

C/$\overline{BE}$[3–0]	COMMAND
0000	Interrupt acknowledge
0001	Special cycle
0010	I/O read
0011	I/O write
0100	Reserved
0101	Reserved
0110	Memory read
0111	Memory write
1000	Reserved
1001	Reserved
1010	Configuration read
1011	Configuration write
1100	Memory read multiple
1101	Dual address cycle
1110	Memory read line
1111	Memory write and invalidate

The bridge never responds as a PCI target to the interrupt acknowledge, special cycle, or reserved commands. The bridge does, however, initiate special cycles on both interfaces when a type 1 configuration cycle issues the special cycle request. The remaining PCI commands address either memory, I/O, or configuration space. The bridge accepts PCI cycles by asserting $\overline{DEVSEL}$ as a medium-speed device; that is, $\overline{DEVSEL}$ is asserted two clock cycles after the address phase.

The PCI6050 device converts memory write and invalid commands to memory write commands when forwarding transactions from either the primary or secondary side of the bridge if the bridge cannot guarantee that an entire cache line will be delivered.

3.4 Configuration Cycles

The *PCI Local Bus Specification* defines two types of PCI configuration read and write cycles: type 0 and type 1. The bridge decodes each type differently. Type 0 configuration cycles are intended for devices on the primary bus, whereas type 1 configuration cycles are intended for devices on some hierarchically subordinate bus. The difference between these two types of cycles is the encoding of the primary PCI (AD) bus during the address phase of the cycle. Figure 3–4 shows the AD bus encoding during the address phase of a type 0 configuration cycle. The 6-bit register number field represents an 8-bit address with the lower bits masked to 0, indicating a double-word boundary. This results in a 256-byte configuration address space per function, per device. Individual byte accesses may be selected within a double word by using the P_C/$\overline{BE}$ signals during the data phase of the cycle.

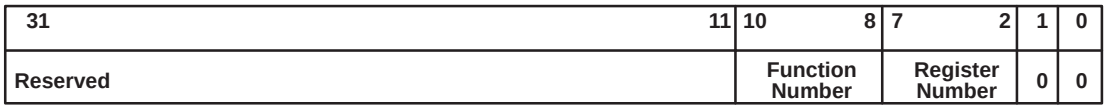

Figure 3–4. AD[31–0] During Address Phase of a Type 0 Configuration Cycle

The bridge claims only type 0 configuration cycles when its IDSEL terminal is asserted during the address phase of the cycle and the PCI function number encoded in the cycle ID is 0. If the function number is 1 or greater, then the bridge does not recognize the configuration command. In this case, the bridge does not assert $\overline{\text{DEVSEL}}$, and the configuration transaction results in a master abort. The bridge services valid type 0 configuration read or write cycles by accessing internal registers from the configuration header.

Because type 1 configuration cycles are issued to devices on the subordinate buses, the bridge claims type 1 cycles based on the bus number of the destination bus. The AD bus encoding during the address phase of a type 1 cycle is shown in Figure 3–5. The device number and bus number fields define the destination bus and device for the cycle.

31 24	23 16	15 11	10 8	7 2	1	0
Reserved	Bus Number	Device Number	Function Number	Register Number	0	1

Figure 3–5. AD[31–0] During Address Phase of a Type 1 Configuration Cycle

Several bridge configuration registers shown in the configuration register table are significant for decoding and claiming type 1 configuration cycles. The destination bus number encoded on the AD bus is compared to the values programmed in the bridge configuration registers 18h, 19h, and 1Ah, which are primary bus number, secondary bus number, and subordinate bus number registers, respectively. These registers default to 00h and are programmed by the host software to reflect the bus hierarchy in the system (see Figure 3–6 for an example of a system bus hierarchy and how the PCI6050 bus number registers are programmed in this case).

When the primary PCI6050 node claims a type 1 configuration cycle that has a bus number equal to its secondary bus number, the PCI6050 device converts the type 1 configuration cycle to a type 0 configuration cycle so that the secondary node may assert the proper AD line as the target device IDSEL (Table 3–2). All other type 1 transactions that access a bus number greater than the bridge's secondary bus number but less than or equal to its subordinate bus number are forwarded as type 1 configuration cycles.

Table 3–2. AD[31–16] During Address Phase of a Type 0 Configuration Cycle

DEVICE NUMBER	SECONDARY IDSEL AD[31–16]	AD ASSERTED
0h	0000 0000 0000 0001	16
1h	0000 0000 0000 0010	17
2h	0000 0000 0000 0100	18
3h	0000 0000 0000 1000	19
4h	0000 0000 0001 0000	20
5h	0000 0000 0010 0000	21
6h	0000 0000 0100 0000	22
7h	0000 0000 1000 0000	23
8h	0000 0001 0000 0000	24
9h	0000 0010 0000 0000	25
10h	0000 0100 0000 0000	26
11h	0000 1000 0000 0000	27
12h	0001 0000 0000 0000	28
13h	0010 0000 0000 0000	29
14h	0100 0000 0000 0000	30
15h	1000 0000 0000 0000	31
10h–1Eh	0000 0000 0000 0000	–

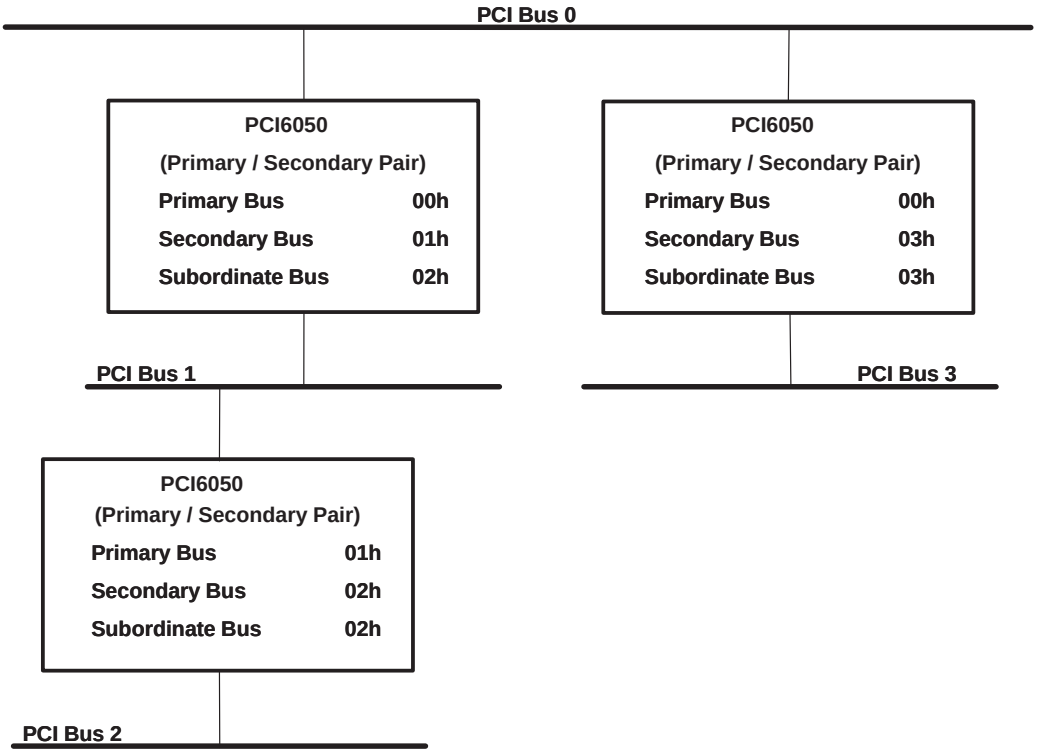

Figure 3–6. Bus Hierarchy and Numbering

3.5 Secondary PCI Clocking

The PCI6050 device provides 10 secondary clock outputs (CLKOUT[0–9]). Nine are provided for clocking secondary devices. The tenth clock (CLKOUT9) should be routed back into the secondary PCI6050 node PCI_CLK input to ensure the all secondary bus devices are synchronized to the bridge, as shown in Figure 3–7.

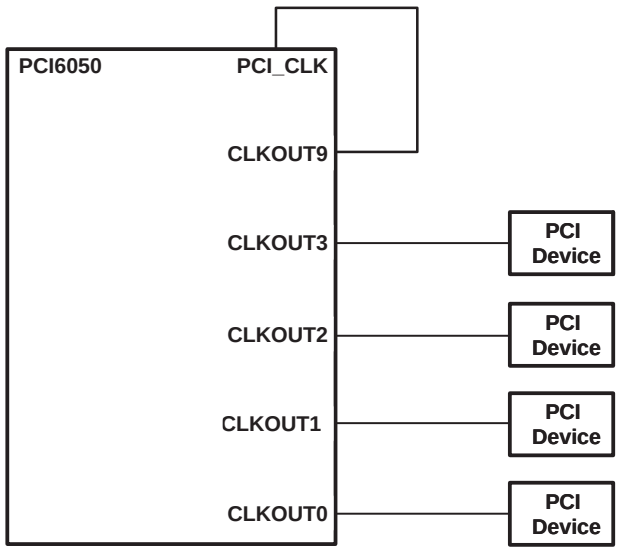

Figure 3–7. Secondary PCI Clock Block Diagram

3.6 Special Cycle Generation

The bridge is designed to generate special cycles on both buses through a type 1 conversion. During a type 1 configuration cycle, if the bus number field matches the bridge's secondary bus number, then the device number field is 1Fh, and the function number field is 07h; consequently, the bridge generates a special cycle on the secondary bus with a message that matches the type 1 configuration cycle data. If the bus number is a subordinate bus and not the secondary bus, then the bridge passes the type 1 special cycle request through to the secondary interface along with the proper message. Special cycles are never passed through the bridge, but they can be generated by the bridge through type 1 configuration cycles with a special cycle request.

3.7 Bus Arbitration

The primary PCI6050 node implements a bus request ($\overline{\text{REQ0}}$) and a bus grant ($\overline{\text{GNT0}}$) terminal for primary bus arbitration. Nine secondary bus requests and nine secondary bus grants are provided on the secondary PCI6050 node. Ten potential initiators, including the bridge, can be located on the secondary bus. The secondary PCI6050 node provides a two-tier arbitration scheme on the secondary PCI bus for priority bus-master handling. The two-tier arbitration scheme improves performance in systems in which not all master devices require the same bandwidth. Any master that requires frequent use of the bus can be programmed to be in the higher priority tier.

3.7.1 Primary Bus Arbitration

The primary PCI6050 node, acting as an initiator on the primary bus, asserts $\overline{\text{REQ0}}$ when attempting to forward transactions upstream on the primary bus. In the upstream direction, as long as a posted write or a delayed transaction request is in the queue, the primary PCI6050 node keeps $\overline{\text{REQ0}}$ asserted. If a target disconnect, a target retry, or a target abort is received in response to a transaction initiated on the primary bus by the PCI6050 device, then $\overline{\text{REQ0}}$ is deasserted for two PCI clock cycles in accordance with the *PCI Local Bus Specification*.

When the primary bus arbiter asserts $\overline{\text{GNT0}}$ in response to $\overline{\text{REQ0}}$ from the primary PCI6050 node, the device initiates a transaction on the primary bus during the next PCI clock cycle after the primary bus is sampled idle.

When $\overline{\text{REQ0}}$ is not asserted and the primary bus arbiter asserts $\overline{\text{GNT0}}$ to the PCI6050 device, it responds by parking the AD[31–0], C/$\overline{\text{BE}}$[3–0], and primary parity (PAR) (driving them to valid logic levels). If the PCI6050 device is parking the primary bus and wants to initiate a transaction on the bus, then it can start the transaction on the next PCI clock by asserting the primary cycle frame ($\overline{\text{FRAME}}$) while $\overline{\text{GNT0}}$ is still asserted. If $\overline{\text{GNT0}}$ is deasserted, then the bridge must re-arbitrate for the bus to initiate another transaction.

3.7.2 Secondary Bus Arbitration

EXTARB controls the state of the secondary internal arbiter. The internal arbiter can be enabled by pulling the EXTARB terminal low. The PCI6050 device provides nine secondary bus request terminals and nine secondary bus grant terminals. Including the bridge, there are a total of 10 potential secondary bus masters. These request and grant signals are connected to the internal arbiter.

An external secondary bus arbiter can be used instead of the PCI6050 internal arbiter. When an external arbiter is used, the internal arbiter of the PCI6050 device should be disabled by pulling the EXTARB terminal high. When an external secondary bus arbiter is used, the PCI6050 device internally reconfigures the $\overline{\text{REQ0}}$ and $\overline{\text{GNT0}}$ signals to its own $\overline{\text{REQ}}$ and $\overline{\text{GNT}}$, respectively. When an external arbiter is used, all unused secondary bus grant outputs ($\overline{\text{GNT}}$[8–1]) are placed in a high-impedance mode. Any unused secondary request inputs ($\overline{\text{REQ}}$[8–1]) should be pulled high to prevent these inputs from oscillating.

3.8 Decode Options

The PCI6050 device supports positive decoding using the standard bridge memory windows on the primary interface and negative decoding on the secondary interface.

3.9 System Error Handling

The PCI6050 device can be configured to signal a system error ($\overline{\text{SERR}}$) under a variety of conditions. The primary $\overline{\text{SERR}}$ event disable register (offset 64h, see Section 5.4) and the primary $\overline{\text{SERR}}$ status register (offset 6Ah, see

Section 5.9) provide control and status bits for each condition which can cause the bridge to signal $\overline{\text{SERR}}$. These individual bits enable $\overline{\text{SERR}}$ reporting for both downstream and upstream transactions.

By default, the PCI6050 device does not signal $\overline{\text{SERR}}$. If the PCI6050 device is configured to signal $\overline{\text{SERR}}$ by setting bit 8 of the PCI command register (offset 04h, see Section 4.3), then the bridge signals $\overline{\text{SERR}}$ if any of the error conditions in the primary $\overline{\text{SERR}}$ event disable register occur and that condition is enabled. By default all error conditions are enabled in the primary $\overline{\text{SERR}}$ event disable register. When the bridge signals $\overline{\text{SERR}}$, bit 14 of the secondary status register (offset 1Eh, see Section 4.17) is set.

3.9.1 Posted Write Parity Error

If bit 1 in the primary $\overline{\text{SERR}}$ event disable register is 0, then parity errors on the target bus during a posted write are passed to the initiating bus by the assertion of $\overline{\text{SERR}}$. When this occurs, bit 1 of the primary $\overline{\text{SERR}}$ status register is set. The status bit may be cleared by writing a 1.

3.9.2 Posted Write Timeout

If bit 2 in the primary $\overline{\text{SERR}}$ event disable register is 0 and the retry timer expires before a posted write is completed, then the PCI6050 signals $\overline{\text{SERR}}$ on the initiating bus. When this occurs, bit 2 of the primary $\overline{\text{SERR}}$ status register is set. The status bit may be cleared by writing a 1.

3.9.3 Target Abort on Posted Writes

If bit 3 in the primary $\overline{\text{SERR}}$ event disable register is 0 and the bridge gets a target abort during a posted write transaction, then the PCI6050 device signals $\overline{\text{SERR}}$ on the initiating bus. When this occurs, bit 3 of the primary $\overline{\text{SERR}}$ status register is set. The status bit may be cleared by writing a 1.

3.9.4 Master Abort on Posted Writes

If bit 4 in the primary $\overline{\text{SERR}}$ event disable register is 0 and a posted write transaction results in a master abort, then the PCI6050 device signals $\overline{\text{SERR}}$ on the initiating bus. When this occurs, bit 4 of the primary $\overline{\text{SERR}}$ status register is set. The status bit may be cleared by writing a 1.

3.9.5 Master Delayed Write Timeout

If bit 5 in the primary $\overline{\text{SERR}}$ event disable register is 0 and the retry timer expires before a delayed write is completed, then the PCI6050 device signals $\overline{\text{SERR}}$ on the initiating bus. When this occurs, bit 5 of the primary $\overline{\text{SERR}}$ status register is set. The status bit may be cleared by writing a 1.

3.9.6 Master Delayed Read Timeout

If bit 6 in the primary $\overline{\text{SERR}}$ event disable register is 0 and the retry timer expires before a delayed read is completed, then the PCI6050 device signals $\overline{\text{SERR}}$ on the initiating bus. When this occurs, bit 6 of the primary $\overline{\text{SERR}}$ status register is set. The status bit may be cleared by writing a 1.

3.9.7 Secondary $\overline{\text{SERR}}$

The PCI6050 device passes $\overline{\text{SERR}}$ from the secondary bus to the primary bus via the serial link if it is enabled for $\overline{\text{SERR}}$ response, bit 8 in the PCI command register (offset 04h, see Section 4.3) is set, and bit 1 in the bridge control register (offset 3Eh, see Section 4.29) is set.

3.10 Parity Handling and Parity Error Reporting

The PCI6050 device forwards parity errors from the initiating bus to the target bus. The following parity conditions result in the bridge signaling an error.

- Address parity error – If the parity error response enable bit (bit 6) in the PCI command register is set, then the PCI6050 device signals $\overline{\text{SERR}}$ on address parity errors and target abort transactions.
- Data parity error – If the parity error response enable bit (bit 6) in the PCI command register is set, then the PCI6050 device signals $\overline{\text{PERR}}$ when it receives bad data. When the bridge detects bad parity, bit 15 (primary parity error) in the PCI status register is set. If the bridge is configured to respond to parity errors, then the data parity error detected bit (bit 8 in the status register) is set when the bridge detects bad parity. The data parity error detected bit is also set when the bridge, as a bus master, asserts $\overline{\text{PERR}}$ or detects $\overline{\text{PERR}}$.

3.11 Master and Target Abort Handling

If the PCI6050 device receives a target abort during a write burst, then it signals target abort back to the initiator bus. If it receives a target abort during a read burst, then it provides all of the valid data on the initiator bus and disconnects. Target aborts for posted and nonposted transactions are reported as specified in the *PCI-to-PCI Bridge Specification.* Master aborts for posted and nonposted transactions are reported as specified in the *PCI-to-PCI Bridge Specification.* If a transaction is attempted on the primary bus after a secondary reset is asserted, then the PCI6050 device follows the master abort mode bit setting in the bridge control register (offset 3Eh, see Section 4.29) for reporting errors.

3.12 Discard Timer

The PCI6050 device is free to discard the data or status of a delayed transaction that was completed with a delayed transaction termination when a bus master has not repeated the request within 2^{10} or 2^{15} PCI clocks (approximately 30 μs and 993 μs, respectively). The *PCI Local Bus Specification* states that a bridge should wait 2^{15} PCI clocks before discarding the transaction data or status.

The PCI6050 device implements a discard timer for use in delayed transactions. After a delayed transaction is completed on the destination bus, the bridge may discard it under two conditions. The first condition occurs when a read transaction is made to a region of memory that the bridge knows is prefetchable, or when the command is a memory read line or a memory read multiple, implying that the memory region is prefetchable. The other condition occurs when the master originating the transaction (either a read or a write, prefetchable or nonprefetchable) has not retried the transaction within 2^{10} or 2^{15} clocks. A timer referred to as the discard timer tracks the number of clocks. When the discard timer expires, the bridge is required to discard the data. The PCI6050 device default value for the discard timer is 2^{15} clocks; however, this value can be set to 2^{10} clocks by setting bit 9 in the bridge control register. For more information on the discard timer, see *error conditions* in the *PCI Local Bus Specification.*

3.13 Delayed Transactions

The bridge supports delayed transactions as defined in the *PCI Local Bus Specification.* A target must be able to complete the initial data phase in 16 PCI clocks or less from the assertion of the cycle frame ($\overline{\text{FRAME}}$), and subsequent data phases must be completed in 8 PCI clocks or less. A delayed transaction consists of three phases:

- An initiator device issues a request.
- The target completes the request on the destination bus and signals the completion to the initiator.
- The initiator completes the request on the originating bus.

If the bridge is the target of a PCI transaction and it must access a slow device to write or read the requested data, and the transaction takes longer than 16 clocks, then the bridge must latch the address, the command, and the byte enables, and then issue a retry to the initiator. The initiator must end the transaction without any transfer of data and is required to retry the transaction later using the same address, command, and byte enables. This is the first phase of the delayed transaction.

During the second phase, if the transaction is a read cycle, then the bridge fetches the requested data on the destination bus, stores it internally, and obtains the completion status, thus completing the transaction on the destination bus. If it is a write transaction, then the bridge writes the data and obtains the completion status, thus completing the transaction on the destination bus. The bridge stores the completion status until the master on the initiating bus retries the initial request.

During the third phase, the initiator re-arbitrates for the bus. When the bridge sees the initiator retry the transaction, it compares the second request to the first request. If the address, command, and byte enables match the values latched from the first request, then the completion status (and data if the request was a read) is transferred to the initiator. At this point, the delayed transaction is complete. If the second request from the initiator does not match the first request exactly, then the bridge issues another retry to the initiator.

The PCI6050 device supports three delayed transactions in each direction at any given time.

3.14 CompactPCI Hot-Swap Support

The PCI6050 device is hot-swap friendly silicon that supports all of the hot-swap capable features, contains support for software control, and integrates circuitry required by the *CompactPCI Hot-Swap Specification*. To be hot-swap capable, the PCI6050 device supports the following:

- Compliance with *PCI Local Bus Specification*
- Tolerance of V_{CC} from early power
- Asynchronous reset
- Tolerance of precharge voltage
- I/O buffers must meet modified V/I requirements
- Limited I/O pin voltage at precharge voltage
- Hot-swap control and status programming via extended PCI capabilities linked list
- Hot-swap terminals: $\overline{\text{HS\_ENUM}}$, HS_SWITCH, and HS_LED

CPCI hot-swap defines a process for installing and removing PCI boards without adversely affecting a running system. The PCI6050 device provides this functionality so that it can be implemented on a board that can be removed and inserted in a hot-swap system.

A primary PCI6050 node provides three terminals to support hot-swap: $\overline{\text{HS\_ENUM}}$ (output), HS_SWITCH (input), and HS_LED (output). The $\overline{\text{HS\_ENUM}}$ output indicates to the system that an insertion event occurred or that a removal event is about to occur. The HS_SWITCH input indicates that state of a board ejector handle, and the HS_LED output lights a blue LED to signal insertion and removal ready status. The primary PCI6050 node hot-swap functionality is controlled via the primary CPCI hot-swap control and status register (offset E6h, see Section 5.29).

The PCI6050 device allows for a 2-ms debounce period on HS_SWITCH input to ensure that the state of the input has been stable for about 2 ms following the insertion and removal of a board implementing the PCI6050 device, before the state change is reported to the hot-swap status bits.

When the $\overline{\text{HS\_ENUM}}$ signal is received at the secondary PCI6050 node $\overline{\text{HS\_ENUM}}$ input pin, the hardware forwards the hot-swap event to the primary system by way of the serial link.

3.15 JTAG Support

As shown in Table 3–3, the PCI6050 device supports the following JTAG instructions:

- EXTEST, BYPASS, and SAMPLE
- HIGHZ and CLAMP
- Private (various private instructions are used by TI for test purposes)

Table 3–3. JTAG Instructions and Op Codes

INSTRUCTION	OP CODE	DESCRIPTION
EXTEST	00000	External test: Drives pins from the boundary scan register
SAMPLE	00001	Sample I/O pins
CLAMP	00100	Drives pins from the boundary scan register and selects the bypass register for shifts
HIGHZ	00101	3-state all outputs and I/O pins except for the TDO pin
BYPASS	11111	Selects the bypass register for shifts

Table 3–4 describes the boundary scan pin order.

Table 3–4. Boundary Scan Pin Order

BOUNDARY SCAN REGISTER NUMBER	PIN NUMBER	PIN NAME	GROUP DISABLE REGISTER	BOUNDARY SCAN CELL TYPE
1	197	SCL	–	Bidirectional
2	198	SDA	–	Bidirectional
3	199	GPE	–	Output
4	203	GPIO0	5	Bidirectional
5	–	–	–	Control
6	204	GPIO1	7	Bidirectional
7	–	–	–	Control
8	205	GPIO2	9	Bidirectional
9	–	–	–	Control
10	206	GPIO3	11	Bidirectional
11	–	–	–	Control
12	207	EXTARB	–	Input
13	208	PRI/$\overline{\text{SEC}}$	–	Input
14	1	CRC_EN	–	Input
15	3	$\overline{\text{INTD}}$	–	Bidirectional
16	4	$\overline{\text{INTC}}$	–	Bidirectional
17	5	$\overline{\text{INTB}}$	–	Bidirectional
18	6	$\overline{\text{INTA}}$	–	Bidirectional
19	9	GNT8	23	Output
20	10	GNT7	23	Output
21	11	GNT6	23	Output
22	12	GNT5	23	Output
23	–	–	–	Control
24	14	GNT4	23	Output
25	15	GNT3	23	Output
26	16	GNT2	23	Output
27	17	GNT1	23	Output
28	18	GNT0	23	Bidirectional
29	21	REQ8	–	Input
30	22	REQ7	–	Input
31	23	REQ6	–	Input
32	24	REQ5	–	Input
33	26	REQ4	–	Input
34	27	REQ3	–	Input
35	28	REQ2	–	Input
36	29	REQ1	–	Input
37	30	REQ0	38	Bidirectional
38	–	–	–	Control
39	32	$\overline{\text{PCI\_RST}}$	40	Input
40	–	–	–	Control
41	33	PCI_CLK	–	Input
42	35	$\overline{\text{GLOBAL\_RST}}$	–	Input
43	37	AD31	46	Bidirectional

Table 3–4. Boundary Scan Pin Order (Continued)

BOUNDARY SCAN REGISTER NUMBER	PIN NUMBER	PIN NAME	GROUP DISABLE REGISTER	BOUNDARY SCAN CELL TYPE
44	38	AD30	46	Bidirectional
45	39	AD29	46	Bidirectional
46	–	–	–	Control
47	41	AD28	46	Bidirectional
48	42	C/$\overline{BE3}$	49	Bidirectional
49	–	–	–	Control
50	43	IDSEL	–	Input
51	45	AD27	54	Bidirectional
52	46	AD26	54	Bidirectional
53	47	AD25	54	Bidirectional
54	–	–	–	Control
55	49	AD24	54	Bidirectional
56	50	AD23	61	Bidirectional
57	51	AD22	61	Bidirectional
58	54	AD21	61	Bidirectional
59	55	AD20	61	Bidirectional
60	56	AD19	61	Bidirectional
61	–	–	–	Control
62	58	AD18	61	Bidirectional
63	59	AD17	61	Bidirectional
64	60	AD16	61	Bidirectional
65	62	C/$\overline{BE2}$	66	Bidirectional
66	–	–	–	Control
67	63	$\overline{FRAME}$	68	Bidirectional
68	–	–	–	Control
69	64	$\overline{IRDY}$	70	Bidirectional
70	–	–	–	Control
71	66	$\overline{TRDY}$	74	Bidirectional
72	67	$\overline{DEVSEL}$	74	Bidirectional
73	68	$\overline{STOP}$	74	Bidirectional
74	–	–	–	Control
75	70	PAR	76	Bidirectional
76	–	–	–	Control
77	71	$\overline{PERR}$	78	Bidirectional
78	–	–	–	Control
79	72	$\overline{SERR}$	–	Bidirectional
80	74	C/$\overline{BE1}$	81	Bidirectional
81	–	–	–	Control
82	75	AD15	86	Bidirectional
83	76	AD14	86	Bidirectional
84	78	AD13	86	Bidirectional
85	80	AD12	86	Bidirectional
86	–	–	–	Control

Table 3–4. Boundary Scan Pin Order (Continued)

BOUNDARY SCAN REGISTER NUMBER	PIN NUMBER	PIN NAME	GROUP DISABLE REGISTER	BOUNDARY SCAN CELL TYPE
87	82	AD11	86	Bidirectional
88	83	AD10	86	Bidirectional
89	84	AD9	86	Bidirectional
90	86	AD8	86	Bidirectional
91	87	C/$\overline{BE0}$	92	Bidirectional
92	–	–	–	Control
93	88	AD7	98	Bidirectional
94	89	AD6	98	Bidirectional
95	91	AD5	98	Bidirectional
96	92	AD4	98	Bidirectional
97	93	AD3	98	Bidirectional
98	–	–	–	Control
99	95	AD2	98	Bidirectional
100	96	AD1	98	Bidirectional
101	97	AD0	98	Bidirectional
102	99	EXT_CLK	–	Input
103	104	EXT_CLK_SEL	–	Input
104	107	$\overline{HS\_ENUM}$	–	Bidirectional
105	108	HS_SWITCH	–	Input
106	109	HS_LED	–	Output
107	111	CLKOUT9	–	Output
108	112	CLKOUT8	–	Output
109	113	CLKOUT7	–	Output
110	114	CLKOUT6	–	Output
111	116	CLKOUT5	–	Output
112	117	CLKOUT4	–	Output
113	118	CLKOUT3	–	Output
114	120	CLKOUT2	–	Output
115	121	CLKOUT1	–	Output
116	122	CLKOUT0	–	Output
117	124	RXD0	–	Input
118	125	RXD1	–	Input
119	127	RXD2	–	Input
120	128	RXD3	–	Input
121	130	RXD4	–	Input
122	131	RXD5	–	Input
123	133	RXD6	–	Input
124	134	RXD7	–	Input
125	136	RX_CLK	–	Input
126	137	RXD8	–	Input
127	139	RXD9	–	Input
128	140	RXD10	–	Input

Table 3–4. Boundary Scan Pin Order (Continued)

BOUNDARY SCAN REGISTER NUMBER	PIN NUMBER	PIN NAME	GROUP DISABLE REGISTER	BOUNDARY SCAN CELL TYPE
129	142	RXD11	–	Input
130	143	RXD12	–	Input
131	145	RXD13	–	Input
132	146	RXD14	–	Input
133	148	RXD15	–	Input
134	149	RXDV	–	Input
135	150	RXER_PRBSPASS	–	Input
136	152	PRBS_EN	–	Output
137	158	TXER	–	Output
138	159	LOOP_EN	–	Output
139	160	TXEN	–	Output
140	161	TXD15	–	Output
141	163	TXD14	–	Output
142	164	TXD13	–	Output
143	166	TXD12	–	Output
144	167	TXD11	–	Output
145	169	TXD10	–	Output
146	170	TXD9	–	Output
147	171	TXD8	–	Output
148	173	TLK_CLK	–	Output
149	175	SYS_CLK	–	Input
150	177	TXD7	–	Output
151	178	TXD6	–	Output
152	180	TXD5	–	Output
153	181	TXD4	–	Output
154	183	TXD3	–	Output
155	184	TXD2	–	Output
156	186	TXD1	–	Output
157	187	TXD0	–	Output

3.16 GPIO Interface

The PCI6050 device implements a four-pin general purpose I/O interface that is controlled through software for both the primary and secondary nodes. The GPIO terminals of each node are controlled independently through two separate locations with the PCI6050 configuration space.

The primary GPIO control registers begin at PCI offset 65h. The secondary GPIO control registers begin at PCI offset A9h. Setting the GPIOTOIIC bit in the diagnostic control register (bit 3 of offset 41h, see Section 5.2) maps the control bits for GPIO1 and GPIO2 to SDA and SCL on the serial ROM interface, respectively. This allows for the system designer to implement in-circuit serial ROM programming through software.

In addition, the PCI6050 device stops forwarding I/O and memory transactions if bit 5 of the chip control register (offset 40h, see Section 5.1) is set to 1 and GPIO3 is driven high. The bridge will complete all queued posted writes and delayed requests but delayed completions will not be returned until GPIO3 is driven low and transaction forwarding is resumed. The bridge continues to accept configuration cycles in this mode.

3.17 PCI Power Management

The *PCI Power Management Interface Specification* establishes the infrastructure required to let the operating system control the power of PCI functions. This is done by defining a standard PCI interface and operations to manage the power of PCI functions on the bus. The PCI bus and the PCI functions can be assigned to one of four software visible power management states, which result in varying levels of power savings.

The four power management states of PCI functions are D0 *fully on* state, D1, D2 *intermediate* states, and D3 *Off* state. Similarly, bus power states are B0–B3. The bus power states B0–B3 are derived from the device power state of the originating device. The power state of the secondary bus is derived from the power state of the PCI6050 device.

For the operating system to power manage the device power states on the PCI bus, PCI functions support four power management operations:

- Capabilities reporting
- Power status reporting
- Setting the power state
- System wake-up

The operating system identifies the capabilities of the PCI function by traversing the new capabilities list. The presence of the new capabilities list is indicated by the PCI status register which provides access to the capabilities list.

3.17.1 Behavior in Low Power States

The PCI6050 device supports D0, D1, D2, and $D3_{hot}$ power states. The PCI6050 device is fully functional only in the D0 state. In the lower power states, the bridge does not accept any I/O or memory transactions. These transactions are master aborted. The bridge accepts type 0 configuration cycles in all power states except $D3_{cold}$. The bridge also accepts type 1 configuration cycles but does not pass these cycles to the secondary bus in any of the low power states. Type 1 configuration writes are discarded and type 1 configuration reads return all 1s. All error reporting is done in low power states. In the D2 and $D3_{hot}$ states, the bridge turns off all secondary clocks for additional power savings.

When the PCI6050 device goes from $D3_{hot}$ to D0, an internal reset is generated. This reset initializes all PCI configuration registers to their default values.

> **NOTE:** Neither the TI extension registers (40h–FFh) nor the power management registers (beginning at offset E0h) are reset.

3.18 Serial ROM Implementation

The optional serial ROM is used to pre-load primary and secondary PCI6050 configuration registers and may be used for other diagnostic purposes. The serial ROM is optional, and is not expected in most PCI6050 implementations. The PCI6050 device samples the SCL terminal following $\overline{\text{GLOBAL\_RST}}$ assertion to determine the presence of a serial ROM. When a serial ROM is not implemented, the SCL PCI6050 terminal should be pulled down with a weak resistor.

Typically, only write-accessible bits in these registers may be pre-loaded, and exceptions are noted in the bit descriptions. However, configuration writes to the registers always takes priority over the data loaded from the secondary serial ROM, regardless of whether or not the serial ROM load has occurred or whether or not any serial ROM load value was ever communicated to the primary side. Thus, the value written by a host configuration write to the secondary CLKOUT control register takes priority on the value loaded from the secondary serial ROM. Figure 3–8 illustrates the PCI6050 serial ROM data format.

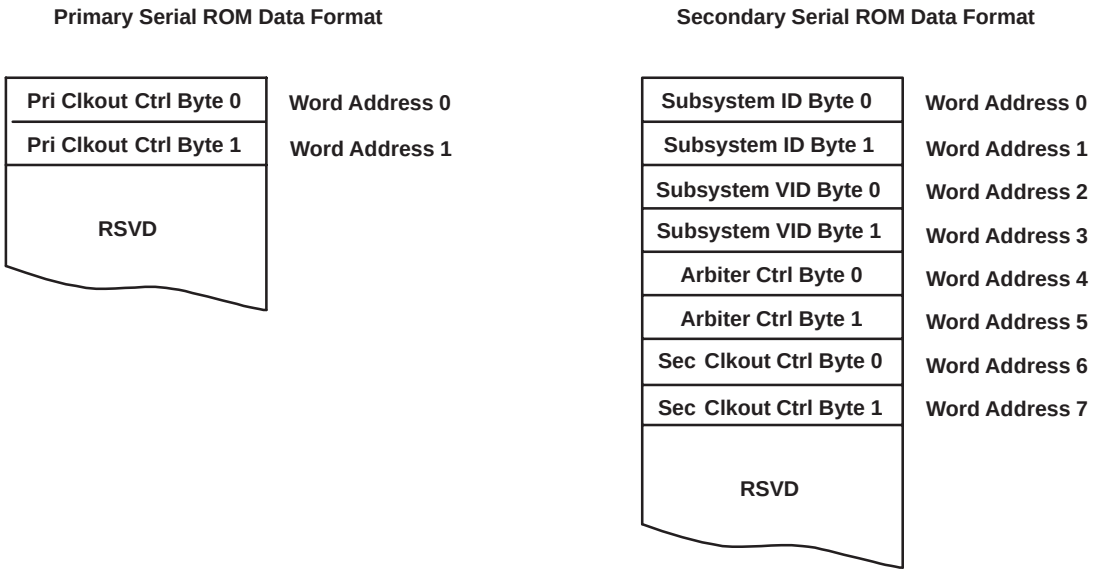

Figure 3–8. Serial ROM Data Format

If the serial ROM is implemented, then it is addressed at slave address A0h after the PCI6050 device enters the active state and is loaded in a data-burst mode. Only serial ROMs that support the address auto-increment data-burst mode can be used with the PCI6050 device.

4 Bridge Configuration Header

The PCI6050 bridge is a single-function PCI device. The configuration header is in compliance with the *PCI-to-PCI Bridge Architecture Specification*. The following table shows the PCI configuration header, which includes the predefined portion of the bridge's configuration space. The PCI configuration offset is shown in the right column under the OFFSET heading.

REGISTER NAME				OFFSET
Device ID		Vendor ID		00h
Status		Command		04h
Class code			Revision ID	08h
BIST	Header type	Latency timer	Cache line size	0Ch
Reserved				10h
Reserved				14h
Secondary bus latency timer	Subordinate bus number	Secondary bus number	Primary bus number	18h
Secondary status		I/O limit	I/O base	1Ch
Memory limit		Memory base		20h
Prefetchable memory limit		Prefetchable memory base		24h
Prefetchable base upper 32 bits				28h
Prefetchable limit upper 32 bits				2Ch
I/O limit upper 16 bits		I/O base upper 16 bits		30h
Reserved			Capability pointer	34h
Reserved				38h
Bridge control register		Interrupt pin	Interrupt line	3Ch
Arbiter control		Diagnostic control	Chip control	40h
Reserved				44h–63h
Primary GPIO input	Primary GPIO direction	Primary GPIO output	$\overline{\text{SERR}}$ control	64h
Reserved	Primary $\overline{\text{SERR}}$ status	Primary CLKOUT control		68h
Reserved	Secondary $\overline{\text{SERR}}$ status	Secondary CLKOUT control		6Ch
Reserved				70h–9Fh
Reserved		General purpose event		A0h
Extended diagnostic status		Reserved		A4h
Secondary GPIO input	Secondary GPIO direction	Secondary GPIO output	Reserved	A8h
Reserved				ACh–BBh
Receive error count	CRC error count	Sequence error count	Reserved	BCh
Reserved		Transceiver test control and status		C0h
Reserved				C4h–DBh
Power management capabilities		PM next item pointer	PM capability ID	DCh
PM data	PMCSR bridge support	Power management control/status		E0h
Reserved	Hot-swap control and status	HS next item pointer	HS capability ID	E4h
Subsystem ID		Subsystem vendor ID		E8h
Reserved		FIFO BIST		ECh
Reserved				F0h
Reserved				F4h
Reserved				F8h
Reserved				FCH

4.1 Vendor Register

This 16-bit value is allocated by the PCI Special Interest Group (SIG) and identifies TI as the manufacturer of this device. The vendor ID assigned to TI is 104Ch.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Vendor ID															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	1	0	0	0	0	0	1	0	0	1	1	0	0

Register: **Vendor ID**
Type: Read-only
Offset: 00h
Default: 104Ch

4.2 Device ID Register

This 16-bit value is allocated by the vendor and identifies the PCI device. The device ID for the PCI6050 device is AC70h.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Device ID															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	1	0	1	0	1	1	0	0	0	1	1	1	0	0	0	0

Register: **Device ID**
Type: Read-only
Offset: 02h
Default: AC70h

4.3 PCI Command Register

The command register provides control over the bridge interface to the primary PCI bus. VGA palette snooping is enabled through this register, and all other bits adhere to the definitions in the *PCI Local Bus Specification*. Table 4–1 describes the bit functions in the command register.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Command															
Type	R	R	R	R	R	R	R/W	R/W	R	R/W	R/W	R	R	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Command**
Type: Read-only, Read/Write
Offset: 04h
Default: 0000h

Table 4–1. PCI Command Register Description

BIT	ACCESS	DESCRIPTION
15–10	R	Reserved. Bits 15–10 return 0s when read.
9	R/W	Fast back-to-back enable. The PCI6050 device does not generate fast back-to-back transactions on the primary PCI bus. Bit 9 is read/write, but does not affect the bridge when set. This bit defaults to 0.
8	R/W	$\overline{\text{SERR}}$ enable. Bit 8 controls the enable for the $\overline{\text{SERR}}$ driver on the primary interface. 0 = Disable $\overline{\text{SERR}}$ driver on primary interface (default). 1 = Enable the $\overline{\text{SERR}}$ driver on primary interface.
7	R	Address/data stepping control. Bit 7 controls address/data stepping by the bridge on both interfaces. The PCI6050 device does not support address/data stepping and this bit is hardwired to 0.
6	R/W	Parity error response enable. Bit 6 controls the bridge response to parity errors. 0 = Parity error response disabled (default) 1 = Parity error response enabled
5	R/W	VGA palette snoop enable. When set, a primary PCI6050 node will pass I/O writes on the primary PCI bus with addresses 3C6h, 3C8h, and 3C9h inclusive of ISA aliases (that is, only AD[9–0] are included in decode) to the secondary PCI bus.
4	R	Memory write and invalidate enable. In a PCI-to-PCI bridge, bit 4 must be read-only and return 0 when read.
3	R	Special cycle enable. A PCI-to-PCI bridge cannot respond as a target to special cycle transactions, so bit 3 is defined as read-only and must return 0 when read.
2	R/W	Bus master enable. Bit 2 controls the ability of the primary PCI6050 node to initiate a cycle on the primary PCI bus. When bit 2 is 0, the secondary PCI6050 node does not respond to any memory or I/O transactions on the secondary interface because they cannot be forwarded to the primary PCI bus. 0 = Bus master capability disabled (default) 1 = Bus master capability enabled
1	R/W	Memory response enable. Bit 1 controls the bridge response to memory accesses for both prefetchable and nonprefetchable memory spaces on the primary PCI bus. Only when bit 1 is set does the bridge forward memory accesses to the secondary bus from a primary bus initiator. 0 = Memory space disabled (default) 1 = Memory space enabled
0	R/W	I/O space enable. Bit 0 controls the bridge response to I/O accesses on the primary interface. Only when bit 0 is set does the bridge forward I/O accesses to the secondary bus from a primary bus initiator. 0 = I/O space disabled (default) 1 = I/O space enabled

4.4 PCI Status Register

The status register provides device information regarding the primary PCI bus to the host system. Table 4–2 describes the bit functions in the PCI status register.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Status															
Type	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0

Register: **Status**
Type: Read/Write, Read-only
Offset: 06h
Default: 0210h

Table 4–2. PCI Status Register Description

BIT	ACCESS	DESCRIPTION
15	R/W	Primary parity error. Bit 15 is set when either an address or data parity error is detected. 0 = No $\overline{\text{PERR}}$ detected (default) 1 = $\overline{\text{PERR}}$ detected
14	R/W	Primary system error. Bit 14 is set if $\overline{\text{SERR}}$ is enabled (bit 8) in the PCI command register (offset 04h, see Section 4.3) and the bridge signals a system error ($\overline{\text{SERR}}$). 0 = No $\overline{\text{SERR}}$ signaled (default) 1 = $\overline{\text{SERR}}$ signaled
13	R/W	Primary master abort received. Bit 13 is set when a cycle initiated by the bridge on the primary bus has been terminated by a master abort. 0 = No master abort received (default) 1 = Master abort received
12	R/W	Primary target abort received. Bit 12 is set when a cycle initiated by the bridge on the primary bus has been terminated by a target abort. 0 = No target abort received (default) 1 = Target abort received
11	R/W	Primary target abort signaled. Bit 11 is set by the bridge when it terminates a transaction on the primary bus with a target abort. 0 = No target abort signaled by the bridge (default) 1 = Target abort signaled by the bridge
10–9	R	Primary $\overline{\text{DEVSEL}}$ timing. These read-only bits encode the timing of $\overline{\text{DEVSEL}}$ and are hardwired 01b, indicating that the bridge asserts this signal at a medium speed.
8	R/W	Primary PCI data parity error detected. Bit 8 is encoded as: 0 = The conditions for setting this bit have not been met. No parity error detected.(default). 1 = A data parity error occurred and the following conditions were met: a. $\overline{\text{PERR}}$ was asserted by any PCI device including the bridge. b. The bridge was the bus master during the data parity error. c. The parity error response enable bit (bit 6) is set in the PCI command register (offset 04h, see Section 4.3).
7	R	Fast back-to-back capable. The PCI6050 device does not support fast back-to-back transactions as a target; therefore, bit 7 is hardwired to 0.
6	R	User-definable feature (UDF) support. The PCI6050 device does not support the user-definable features; therefore, bit 6 is hardwired to 0.
5	R	66-MHz capable. The PCI6050 device operates at a maximum PCI_CLK frequency of 33 MHz; therefore, bit 5 is hardwired to 0.
4	R	Capabilities list. Bit 4 is read-only and is hardwired to 1, indicating that capabilities additional to standard PCI are implemented. The linked list of PCI power management capabilities is implemented by this function.
3–0	R	Reserved. Bits 3–0 return 0s when read.

4.5 Revision ID Register

The revision ID register indicates the silicon revision of the PCI6050 device.

Bit	7	6	5	4	3	2	1	0
Name	Revision ID							
Type	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0

Register: **Revision ID**
Type: Read-only
Offset: 08h
Default: 00h (reflects the current revision of the silicon)

4.6 Class Code Register

This register categorizes the PCI6050 device as a PCI-to-PCI bridge device (0604h) with a 00h programming interface.

Bit	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8
Name	Class code															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0

Bit	7	6	5	4	3	2	1	0
Name	Class code							
Type	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0

Register: **Class code**
Type: Read-only
Offset: 09h
Default: 060400h

4.7 Cache Line Size Register

The cache line size register is programmed by host software to indicate the system cache line size needed by the bridge on memory read line and memory read multiple transactions.

Bit	7	6	5	4	3	2	1	0
Name	Cache line size							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **Cache line size**
Type: Read/Write
Offset: 0Ch
Default: 00h

4.8 Primary Latency Timer Register

The latency timer register specifies the latency timer for the bridge in units of PCI clock cycles. When the bridge is a primary PCI bus initiator and asserts $\overline{\text{FRAME}}$, the latency timer begins counting from 0. If the latency timer expires before the bridge transaction has terminated, then the bridge terminates the transaction when its $\overline{\text{GNT}}$ is de-asserted.

Bit	7	6	5	4	3	2	1	0
Name	Latency timer							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **Latency timer**
Type: Read/Write
Offset: 0Dh
Default: 00h

4.9 Header Type Register

The header type register is read-only and returns 01h when read, indicating that the PCI6050 configuration space adheres to the PCI-to-PCI bridge configuration. Only the layout for bytes 10h–3Fh of configuration space is considered.

Bit	7	6	5	4	3	2	1	0
Name	Header type							
Type	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	1

Register: **Header type**
Type: Read-only
Offset: 0Eh
Default: 01h

4.10 BIST Register

The PCI6050 device does not support built-in self test (BIST). The BIST register is read-only and returns the value 00h when read.

Bit	7	6	5	4	3	2	1	0
Name	BIST							
Type	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0

Register: **BIST**
Type: Read-only
Offset: 0Fh
Default: 00h

4.11 Primary Bus Number Register

The primary bus number register indicates the primary bus number to which the bridge is connected. The bridge uses this register, in conjunction with the secondary bus number and subordinate bus number registers, to determine when to forward PCI configuration cycles to the secondary buses.

Bit	7	6	5	4	3	2	1	0
Name	Primary bus number							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **Primary bus number**
Type: Read/Write
Offset: 18h
Default: 00h

4.12 Secondary Bus Number Register

The secondary bus number register indicates the secondary bus number to which the bridge is connected. The PCI6050 device uses this register, in conjunction with the primary bus number and subordinate bus number registers, to determine when to forward PCI configuration cycles to the secondary buses. Configuration cycles directed to the secondary bus are converted to type 0 configuration cycles.

Bit	7	6	5	4	3	2	1	0
Name	Secondary bus number							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **Secondary bus number**
Type: Read/Write
Offset: 19h
Default: 00h

4.13 Subordinate Bus Number Register

The subordinate bus number register indicates the bus number of the highest numbered bus beyond the primary bus existing behind the bridge. The PCI6050 device uses this register, in conjunction with the primary bus number and secondary bus number registers, to determine when to forward PCI configuration cycles to the subordinate buses. Configuration cycles directed to a subordinate bus (not the secondary bus) remain type 1 cycles as the cycle crosses the bridge.

Bit	7	6	5	4	3	2	1	0
Name	Subordinate bus number							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **Subordinate bus number**
Type: Read/Write
Offset: 1Ah
Default: 00h

4.14 Secondary Bus Latency Timer Register

The secondary bus latency timer specifies the latency timer for the secondary PCI6050 node in units of PCI clock cycles. When the bridge is a secondary PCI bus initiator and asserts $\overline{\text{FRAME}}$, the latency timer begins counting from 0. If the latency timer expires before the bridge transaction has terminated, then the bridge terminates the transaction when its $\overline{\text{GNT}}$ is de-asserted. The PCI-to-PCI bridge $\overline{\text{GNT}}$ is an internal signal and is removed when another secondary bus master arbitrates for the bus.

Bit	7	6	5	4	3	2	1	0
Name	Secondary bus latency timer							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **Secondary bus latency timer**
Type: Read/Write
Offset: 1Bh
Default: 00h

4.15 I/O Base Register

The I/O base register is used in decoding I/O addresses to pass through the bridge. The bridge supports 32-bit I/O addressing; thus, bits 3–0 are read-only and default to 0001b. The upper 4 bits are writable and correspond to address bits AD15–AD12. The lower 12 address bits of the I/O base address are considered 0. Thus, the bottom of the defined I/O address range is aligned on a 4K-byte boundary. The upper 16 address bits of the 32-bit I/O base address corresponds to the contents of the I/O base upper 16 bits register (offset 30h, see Section 4.24). Primary PCI6050 nodes positively decode the window defined by the I/O base and limit registers. Secondary PCI6050 nodes negatively decode the window programmed in these registers.

Bit	7	6	5	4	3	2	1	0
Name	I/O base							
Type	R/W	R/W	R/W	R/W	R	R	R	R
Default	0	0	0	0	0	0	0	1

Register: **I/O base**
Type: Read/Write, Read-only
Offset: 1Ch
Default: 01h

4.16 I/O Limit Register

The I/O limit register is used in decoding I/O addresses to pass through the bridge. The bridge supports 32-bit I/O addressing; thus, bits 3–0 are read-only and default to 0001b. The upper 4 bits are writable and correspond to address bits AD15–AD12. The lower 12 address bits of the I/O limit address are considered FFFh. Thus, the top of the defined I/O address range is aligned on a 4K-byte boundary. The upper 16 address bits of the 32-bit I/O limit address corresponds to the contents of the I/O limit upper 16 bits register (offset 32h, see Section 4.25). Primary PCI6050 nodes positively decode the window defined by the I/O base and limit registers. Secondary PCI6050 nodes negatively decode the window programmed in these registers.

Bit	7	6	5	4	3	2	1	0
Name	I/O limit							
Type	R/W	R/W	R/W	R/W	R	R	R	R
Default	0	0	0	0	0	0	0	1

Register: **I/O limit**
Type: Read/Write, Read-only
Offset: 1Dh
Default: 01h

4.17 Secondary Status Register

The secondary status register is similar in function to the PCI status register (offset 06h, see Section 4.4); however, its bits reflect status conditions of the secondary interface. Bits in this register are cleared by writing a 1 to the respective bit.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Secondary status															
Type	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0

Register: **Secondary status**
Type: Read/Write, Read-only
Offset: 1Eh
Default: 0280h

Table 4–3. Secondary Status Register Description

BIT	ACCESS	DESCRIPTION
15	R/W	Secondary parity error detected. Bit 15 is set when a parity error is detected on the secondary PCI interface. 0 = No parity error detected on the secondary bus (default) 1 = Parity error detected on the secondary bus
14	R/W	Secondary $\overline{\text{SERR}}$ detected. Bit 14 is set when the secondary interface detects $\overline{\text{SERR}}$ asserted. Note that the secondary PCI6050 node never asserts $\overline{\text{SERR}}$. 0 = No $\overline{\text{SERR}}$ detected on the secondary bus (default) 1 = $\overline{\text{SERR}}$ detected on the secondary bus
13	R/W	Secondary master abort received. Bit 13 is set when a cycle initiated by the bridge on the secondary bus has been terminated by a master abort. 0 = No master abort received (default) 1 = Bridge master aborted the cycle
12	R/W	Secondary target abort received. Bit 12 is set when a cycle initiated by the bridge on the secondary bus has been terminated by a target abort. 0 = No target abort received (default) 1 = Bridge received a target abort
11	R/W	Secondary target abort signaled. Bit 11 is set by the PCI6050 device when it terminates a transaction on the secondary bus with a target abort. 0 = No target abort signaled (default) 1 = Bridge signaled a target abort
10–9	R	Secondary $\overline{\text{DEVSEL}}$ timing. These read-only bits encode the timing of $\overline{\text{DEVSEL}}$ and are hardwired to 01b, indicating that the bridge asserts this signal at a medium speed.
8	R/W	Secondary data parity error detected. Bit 8 is encoded as: 0 = The conditions for setting this bit have not been met. 1 = A data parity error occurred and the following conditions were met: a. $\overline{\text{PERR}}$ was asserted by any PCI device including the bridge. b. The bridge was the bus master during the data parity error. c. The secondary parity error response enable bit (bit 0) is set in the bridge control register (offset 3Eh, see Section 4.29).
7	R	Fast back-to-back capable. Hardwired to 1 to indicate that the PCI6050 device is able to respond to fast back-to-back transactions on the secondary interface.
6	R	User-definable feature (UDF) support. The PCI6050 device does not support the user-definable features; therefore, bit 6 is hardwired to 0.
5	R	66-MHz capable. The PCI6050 device operates at a maximum PCI_CLK frequency of 33 MHz; therefore, bit 5 is hardwired to 0.
4–0	R	Reserved. Bits 4–0 return 0s when read.

4.18 Memory Base Register

The memory base register defines the base address of a memory-mapped I/O address range used by the bridge to determine when to forward memory transactions from one interface to the other. The upper 12 bits of this register are read/write and correspond to the address bits AD31–AD20. The lower 20 address bits are considered 0s; thus, the address range is aligned to a 1M-byte boundary. The bottom 4 bits are read-only and return 0s when read.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Memory base															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Memory base**
Type: Read/Write, Read-only
Offset: 20h
Default: 0000h

4.19 Memory Limit Register

The memory limit register defines the upper-limit address of a memory-mapped I/O address range used to determine when to forward memory transactions from one interface to the other. The upper 12 bits of this register are read/write and correspond to the address bits AD31–AD20. The lower 20 address bits are considered 1s; thus, the address range is aligned to a 1M-byte boundary. The bottom 4 bits are read-only and return 0s when read.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Memory limit															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Memory limit**
Type: Read/Write, Read-only
Offset: 22h
Default: 0000h

4.20 Prefetchable Memory Base Register

The prefetchable memory base register defines the base address of a prefetchable memory address range used by the bridge to determine when to forward memory transactions from one interface to the other. The upper 12 bits of this register are read/write and correspond to the address bits AD31–AD20. The lower 20 address bits are considered 0s; thus, the address range is aligned to a 1M-byte boundary. The bottom 4 bits are read-only and return 0s when read. The prefetchable memory window may be mapped anywhere in the 64-bit address space as indicated by bit 0 returning a value of 1 when read.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Prefetchable memory base															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Prefetchable memory base**
Type: Read/Write, Read-only
Offset: 24h
Default: 0000h

4.21 Prefetchable Memory Limit Register

The prefetchable memory limit register defines the upper-limit address of a prefetchable memory address range used to determine when to forward memory transactions from one interface to the other. The upper 12 bits of this register are read/write and correspond to the address bits AD31–AD20. The lower 20 address bits are considered 1s; thus, the address range is aligned to a 1M-byte boundary. The bottom 4 bits are read-only and return 0s when read. The prefetchable memory window may be mapped anywhere in the 64-bit address space as indicated by bit 0 returning a value of 1 when read.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Prefetchable memory limit															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Prefetchable memory limit**
Type: Read/Write, Read-only
Offset: 26h
Default: 0000h

4.22 Prefetchable Base Upper 32 Bits Register

The prefetchable base upper 32 bits register plus the prefetchable memory base register define the base address of the 64-bit prefetchable memory address range used by the bridge to determine when to forward memory transactions from one interface to the other. The prefetchable base upper 32 bits register should be programmed to all 0s when 32-bit addressing is being used.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Prefetchable base upper 32 bits															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Prefetchable base upper 32 bits															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Prefetchable base upper 32 bits**
Type: Read/Write
Offset: 28h
Default: 0000 0000h

4.23 Prefetchable Limit Upper 32 Bits Register

The prefetchable limit upper 32 bits register plus the prefetchable memory limit register define the base address of the 64-bit prefetchable memory address range used by the bridge to determine when to forward memory transactions from one interface to the other. The prefetchable limit upper 32 bits register should be programmed to all 0s when 32-bit addressing is being used.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Prefetchable limit upper 32 bits															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Prefetchable limit upper 32 bits															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Prefetchable limit upper 32 bits**
Type: Read/Write
Offset: 2Ch
Default: 0000 0000h

4.24 I/O Base Upper 16 Bits Register

The I/O base upper 16 bits register specifies the upper 16 bits corresponding to AD31–AD16 of the 32-bit address that specifies the base of the I/O range to forward from the primary PCI bus to the secondary PCI bus.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	I/O base upper 16 bits															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **I/O base upper 16 bits**
Type: Read/Write
Offset: 30h
Default: 0000h

4.25 I/O Limit Upper 16 Bits Register

The I/O limit upper 16 bits register specifies the upper 16 bits corresponding to AD31–AD16 of the 32-bit address that specifies the upper limit of the I/O range to forward from the primary PCI bus to the secondary PCI bus.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	I/O limit upper 16 bits															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **I/O limit upper 16 bits**
Type: Read/Write
Offset: 32h
Default: 0000h

4.26 Capability Pointer Register

The capability pointer register provides the pointer to the PCI configuration header where the PCI power management register block resides. The capability pointer provides access to the first item in the linked list of capabilities. The capability pointer register is read-only and returns DCh when read, indicating the power management registers are located at PCI header offset DCh.

Bit	7	6	5	4	3	2	1	0
Name	Capability pointer							
Type	R	R	R	R	R	R	R	R
Default	1	1	0	1	1	1	0	0

Register: **Capability pointer**
Type: Read-only
Offset: 34h
Default: DCh

4.27 Interrupt Line Register

The interrupt line register is read/write and is used to communicate interrupt line routing information. Because the bridge does not implement an interrupt signal terminal, this register defaults to FFh.

Bit	7	6	5	4	3	2	1	0
Name	Interrupt line							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	1	1	1	1	1	1	1	1

Register: **Interrupt line**
Type: Read/Write
Offset: 3Ch
Default: FFh

4.28 Interrupt Pin Register

The bridge default state does not implement any interrupt terminals. Reads from bits 7–0 of this register return 0s.

Bit	7	6	5	4	3	2	1	0
Name	Interrupt pin							
Type	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0

Register: **Interrupt pin**
Type: Read-only
Offset: 3Dh
Default: 00h

4.29 Bridge Control Register

The bridge control register (see Table 4–4) provides many of the same controls for the secondary interface that are provided by the command register for the primary interface. Some bits affect the operation of both interfaces.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Bridge control															
Type	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Bridge control**
Type: Read/Write, Read-only
Offset: 3Eh
Default: 0000h

Table 4–4. Bridge Control Register Description

BIT	ACCESS	DESCRIPTION
15–12	R	Reserved. Bits 15–12 return 0s when read.
11	R/W	Completion discard $\overline{\text{SERR}}$ enable. 0 = $\overline{\text{SERR}}$ signaling disabled for primary discard timeouts (default) 1 = $\overline{\text{SERR}}$ signaling enabled for primary discard timeouts
10	R/W	Completion discard timeout status. When set, this bit must be cleared by writing 1 to it. 0 = No discard timer error (default) 1 = Discard timer error. Either primary or secondary discard timer expired and a delayed transaction was discarded from the queue in the bridge.
9	R/W	Secondary completion discard timer. Selects the number of PCI clocks that the secondary PCI6050 node will wait for a master on the secondary interface to repeat a delayed transaction request. 0 = The secondary discard timer counts 2^{15} PCI clock cycles (default). 1 = The secondary discard timer counts 2^{10} PCI clock cycles.
8	R/W	Primary completion discard timer. Selects the number of PCI clocks that the primary PCI6050 node will wait for a master on the primary interface to repeat a delayed transaction request. 0 = The primary discard timer counts 2^{15} PCI clock cycles (default). 1 = The primary discard timer counts 2^{10} PCI clock cycles.
7	R/W	Secondary FBB enable. Controls the ability of the PCI6050 device to generate fast back-to-back transactions on the secondary interface. 0 = PCI6050 device does not generate fast back-to-back transactions on the secondary PIC bus. 1 = PIC6050 device generates fast back-to-back transactions on the secondary PIC bus.
6	R/W	Secondary bus reset. When bit 6 is set, the secondary reset signal ($\overline{\text{PCI\_RST}}$) is asserted. $\overline{\text{PCI\_RST}}$ is de-asserted by resetting this bit. Bit 6 is encoded as: 0 = Do not force the assertion of $\overline{\text{PCI\_RST}}$ (default). 1 = Force the assertion of $\overline{\text{PCI\_RST}}$.
5	R/W	Master abort mode. Bit 5 controls how the bridge responds to a master abort that occurs on either interface when the bridge is the master. If this bit is set, a posted write transaction has completed on the requesting interface, and $\overline{\text{SERR}}$ enable (bit 8) of the PCI command register (offset 04h, see Section 4.3) is 1, then $\overline{\text{SERR}}$ is asserted when a master abort occurs. If the transaction has not completed, then a target abort is signaled. If the bit is cleared, then all 1s are returned on reads and write data is accepted and discarded when a transaction that crosses the bridge is terminated with master abort. The default state of bit 5 after a reset is 0. 0 = Do not report master aborts (return FFFF FFFFh on reads and discard data on writes) (default). 1 = Report master aborts by signaling target abort if possible, or if $\overline{\text{SERR}}$ is enabled via bit 1 of this register, by asserting $\overline{\text{SERR}}$.
4	R	Reserved. Bit 4 returns 0 when read.
3	R/W	VGA enable. When bit 3 is set, the bridge positively decodes and forwards VGA-compatible memory addresses in the video frame buffer range 000A 0000h–000B FFFFh, I/O addresses in the range 03B0h–03BBh, and 03C0–03DFh from the primary to the secondary interface, independent of the I/O and memory address ranges. When this bit is set, the bridge blocks forwarding of these addresses from the secondary to the primary. Reset clears this bit. Bit 3 is encoded as: 0 = Do not forward VGA-compatible memory and I/O addresses from the primary to the secondary interface (default). 1 = Forward VGA-compatible memory and I/O addresses from the primary to the secondary, independent of the I/O and memory address ranges and independent of the ISA enable bit.

Table 4–4. Bridge Control Register Description (Continued)

BIT	ACCESS	DESCRIPTION
2	R/W	ISA enable. When bit 2 is set, the bridge blocks the forwarding of ISA I/O transactions from the primary to the secondary, addressing the last 768 bytes in each 1K-byte block. This applies only to the addresses (defined by the I/O window registers) that are located in the first 64K bytes of PCI I/O address space. From the secondary to the primary, I/O transactions are forwarded if they address the last 768 bytes in each 1K-byte block in the address range specified in the I/O window registers. Bit 2 is encoded as: 0 = Forward all I/O addresses in the address range defined by the I/O base and I/O limit registers (default). 1 = Block forwarding of ISA I/O addresses in the address range defined by the I/O base and I/O limit registers when these I/O addresses are in the first 64K bytes of PCI I/O address space and address the top 768 bytes of each 1K-byte block.
1	R/W	$\overline{\text{SERR}}$ forwarding enable. Bit 1 controls the forwarding of secondary interface $\overline{\text{SERR}}$ assertions to the primary interface. Only when this bit is set does the bridge forward $\overline{\text{SERR}}$ to the primary bus signal $\overline{\text{SERR}}$. For the primary interface to assert $\overline{\text{SERR}}$, bit 8 of the PCI command register (offset 04h, see Section 4.3) must be set. 0 = $\overline{\text{SERR}}$ disabled (default) 1 = $\overline{\text{SERR}}$ enabled
0	R/W	Secondary parity error response enable. Bit 0 controls the bridge response to parity errors on the secondary interface. When this bit is set, the secondary PCI6050 node asserts $\overline{\text{PERR}}$ to report parity errors on the secondary interface. 0 = Ignore address and parity errors on the secondary interface (default). 1 = Enable parity error reporting and detection on the secondary interface.

5 Extension Registers

The TI extension registers are those registers that lie outside the standard PCI-to-PCI bridge device configuration space (that is, registers 40h–FFh in PCI configuration space in the PCI6050 device). These registers can be accessed through configuration reads and writes. The TI extension registers add flexibility and performance benefits to the standard PCI-to-PCI bridge. Table 5–1 describes the bit functions in the chip control register.

5.1 Chip Control Register

The chip control register contains read/write and read-only bits and has a default value of 00h. This register is used to control the functionality of certain PCI transactions.

Bit	7	6	5	4	3	2	1	0
Name	Chip control							
Type	R	R	R/W	R/W	R	R	R/W	R
Default	0	0	0	0	0	0	0	0

Register: **Chip control**
Type: Read/Write, Read only
Offset: 40h
Default: 00h

Table 5–1. Chip Control Register Description

BIT	ACCESS	DESCRIPTION
7–6	R	Reserved. Bits 7–5 return 0s when read.
5	R/W	Transaction forwarding control for I/O and memory cycles. 0 = Transaction forwarding is controlled by bits 0 and 1 of the PCI command register (offset 04h, see Section 4.3) (default). 1 = Transaction forwarding is disabled if GPIO3 is driven high.
4	R/W	Memory read prefetch disable. When bit 4 is set, memory read prefetching is disabled. 0 = Upstream memory reads are enabled (default). 1 = Upstream memory reads are disabled.
3–2	R	Reserved. Bits 3–2 return 0s when read.
1	R/W	Memory write and memory write and invalidate disconnect control. 0 = Disconnects on queue full or 4-KB boundaries (default) 1 = Disconnects on queue full, 4-KB boundaries, and cache line boundaries
0	R	Reserved. Bit 0 returns 0 when read.

5.2 Diagnostic Control Register

The diagnostic control register is used to control or check the functionality of the serial PCI link or the serial ROM implementation. Table 5–2 describes the bit functions in the diagnostic control register.

Bit	7	6	5	4	3	2	1	0
Name	Diagnostic control							
Type	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	x	0	0	0	0	0

Register: **Diagnostic control**
Type: Read-only, Read/Write
Offset: 41h
Default: 00h

Table 5–2. Diagnostic Control Register Description

BIT	ACCESS	DESCRIPTION
7	R	Reserved. Bit 7 returns 0 when read.
6	R/W	Serial ROM test mode. 0 = Serial ROM clock is PCI clock divided by 256 (default). 1 = Serial ROM clock is PCI clock divided by 4.
5	R/W	CRC enable. Bit 5 reflects the PCI6050 mode of operation (CRC or non-CRC). Hardware updates this bit with the value of CRC_EN pin (pin 1) latched during reset. After reset, this may be used to switch modes. 0 = The PCI6050 device does not use a 16-bit CRC to protect the serial link packets. 1 = The PCI6050 device uses a 16-bit CRC to protect the serial link packets.
4	R/W	Link dead. A primary PCI6050 device sets this bit if the start-up timer expires without receiving a confirmation packet from the secondary node. This bit can be cleared by a write of 1, which means that the node repeats the handshaking mechanism to establish the link. 0 = The serial link is functional and the secondary node is available. 1 = The serial link is nonfunctional and/or the secondary node is unavailable.
3	R/W	GPIOTOIIC. Allows software to route the GPIO control registers to the SDA and SCL terminals for in-circuit serial ROM programming. 0 = GPIO1 and GPIO2 registers control their respective terminals. 1 = GPIO1 and GPIO2 registers control SDA and SCL, respectively.
2–1	R/W	Internal counter test. These bits are used for chip validation only and should not be modified during operation in an actual system.
0	R/W	Chip and secondary bus reset control. Writing a 1 to this bit causes the PCI6050 device to set bit 6 of the bridge control register (offset 3Eh, see Section 4.29) and internally reset the configuration registers. Bit 6 of the bridge control register is not reset by the internal reset. Bit 0 is self-clearing.

5.3 Arbiter Control Register

The arbiter control register is used for the bridge's internal arbiter. The arbitration scheme used is a two-tier rotational arbitration. The secondary PCI6050 node is the only secondary bus initiator that defaults to the higher priority arbitration tier. This internal arbiter is used in a secondary PCI6050 node if EXTARB (pin 207) is low. If EXTARB is high, then the PCI6050 device communicates to the external secondary PCI bus arbiter via $\overline{\text{REQ0}}$ and $\overline{\text{GNT0}}$. When PCI6050 device internal arbiter is used, a master may be connected to $\overline{\text{REQ0}}$ and $\overline{\text{GNT0}}$. Table 5–3 describes the bit functions in the arbiter control register.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Arbiter control															
Type	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Register: **Arbiter control**
Type: Read/Write, Read-only
Offset: 42h
Default: 0200h

Table 5–3. Arbiter Control Register Description

BIT	ACCESS	DESCRIPTION
15–10	R	Reserved. Bits 15-10 return 0s when read.
9	R/W	Bridge tier select. This bit determines the priority of the PCI6050 bridge in the two-tier arbitration scheme. 0 = Lowest priority tier 1 = Highest priority tier (default)
8	R/W	$\overline{\text{GNT8}}$ tier select. This bit determines the priority of the $\overline{\text{GNT8}}$ device in the two-tier arbitration scheme. 0 = Lowest priority tier (default) 1 = Highest priority tier
7	R/W	$\overline{\text{GNT7}}$ tier select. This bit determines the priority of the $\overline{\text{GNT7}}$ device in the two-tier arbitration scheme. 0 = Lowest priority tier (default) 1 = Highest priority tier
6	R/W	$\overline{\text{GNT6}}$ tier select. This bit determines the priority of the $\overline{\text{GNT6}}$ device in the two-tier arbitration scheme. 0 = Lowest priority tier (default) 1 = Highest priority tier
5	R/W	$\overline{\text{GNT5}}$ tier select. This bit determines the priority of the $\overline{\text{GNT5}}$ device in the two-tier arbitration scheme. 0 = Lowest priority tier (default) 1 = Highest priority tier
4	R/W	$\overline{\text{GNT4}}$ tier select. This bit determines the priority of the $\overline{\text{GNT4}}$ device in the two-tier arbitration scheme. 0 = Lowest priority tier (default) 1 = Highest priority tier
3	R/W	$\overline{\text{GNT3}}$ tier select. This bit determines the priority of the $\overline{\text{GNT3}}$ device in the two-tier arbitration scheme. 0 = Lowest priority tier (default) 1 = Highest priority tier
2	R/W	$\overline{\text{GNT2}}$ tier select. This bit determines the priority of the $\overline{\text{GNT2}}$ device in the two-tier arbitration scheme. 0 = Lowest priority tier (default) 1 = Highest priority tier
1	R/W	$\overline{\text{GNT1}}$ tier select. This bit determines the priority of the $\overline{\text{GNT1}}$ device in the two-tier arbitration scheme. 0 = Lowest priority tier (default) 1 = Highest priority tier
0	R/W	$\overline{\text{GNT0}}$ tier select. This bit determines the priority of the $\overline{\text{GNT0}}$ device in the two-tier arbitration scheme. 0 = Lowest priority tier (default) 1 = Highest priority tier

5.4 Primary $\overline{\text{SERR}}$ Event Disable Register

The primary $\overline{\text{SERR}}$ event disable register is used to enable/disable $\overline{\text{SERR}}$ event on the primary interface. All events are enabled by default. Table 5–4 describes the bit functions in the primary $\overline{\text{SERR}}$ event disable register.

Bit	7	6	5	4	3	2	1	0
Name	Primary $\overline{\text{SERR}}$ event disable							
Type	R	R/W	R/W	R/W	R/W	R/W	R/W	R
Default	0	0	0	0	0	0	0	0

Register: **Primary $\overline{\text{SERR}}$ event disable**
Type: Read/Write, Read-only
Offset: 64h
Default: 00h

Table 5–4. Primary $\overline{\text{SERR}}$ Event Disable Register Description

BIT	ACCESS	DESCRIPTION
7	R	Reserved. Bit 7 returns 0 when read.
6	R/W	Master delayed read timeout 0 = $\overline{\text{SERR}}$ is signaled on a master timeout after 2^{24} retries on a delayed read (default). 1 = $\overline{\text{SERR}}$ is not signaled on a master timeout.
5	R/W	Master delayed write timeout 0 = $\overline{\text{SERR}}$ is signaled on a master timeout after 2^{24} retries on a delayed write (default). 1 = $\overline{\text{SERR}}$ is not signaled on a master timeout.
4	R/W	Master abort on posted write transactions. When set, bit 4 enables $\overline{\text{SERR}}$ reporting on master aborts on posted write transactions. 0 = Master aborts on posted writes are enabled (default). 1 = Master aborts on posted writes are disabled.
3	R/W	Target abort on posted writes. When set, bit 3 enables $\overline{\text{SERR}}$ reporting on target aborts on posted write transactions. 0 = Target aborts on posted writes are enabled (default). 1 = Target aborts on posted writes are disabled.
2	R/W	Master posted write timeout 0 = $\overline{\text{SERR}}$ is signaled on a master timeout after 2^{24} retries on a posted write (default). 1 = $\overline{\text{SERR}}$ is not signaled on a master timeout.
1	R/W	Posted write parity error 0 = $\overline{\text{SERR}}$ is signaled on a posted write parity error (default). 1 = $\overline{\text{SERR}}$ is not signaled on a posted write parity error.
0	R	Reserved. Bit 0 returns 0 when read.

5.5 Primary GPIO Output Data Register

The primary GPIO output data register controls the data driven on the GPIO terminals of the primary node when configured as outputs. Table 5–5 describes the bit functions in the primary GPIO output data register.

Bit	7	6	5	4	3	2	1	0
Name	Primary GPIO output data							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **Primary GPIO output data**
Type: ReadWrite
Offset: 65h
Default: 00h

Table 5–5. Primary GPIO Output Data Register Description

BIT	ACCESS	DESCRIPTION
7–4	R/W	GPIO3–GPIO0 output high. Writing a 1 to any of these bits causes the corresponding GPIO signal to be driven high. Writing a 0 has no effect. GPIO terminals programmed as inputs are not affected by these bits.
3–0	R/W	GPIO3–GPIO0 output low. Writing a 1 to any of these bits causes the corresponding GPIO signal to be driven low. Writing a 0 has no effect. GPIO terminals programmed as inputs are not affected by these bits.

5.6 Primary GPIO Output Enable Register

The primary GPIO output enable register controls the direction of the primary node GPIO signals. By default all GPIO terminals are inputs. Table 5–6 describes the bit functions in the primary GPIO output enable register.

Bit	7	6	5	4	3	2	1	0
Name	Primary GPIO output enable							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **Primary GPIO output enable**
Type: Read/Write
Offset: 66h
Default: 00h

Table 5–6. Primary GPIO Output Enable Register Description

BIT	ACCESS	DESCRIPTION
7–4	R/W	GPIO3–GPIO0 output enable. Writing a 1 to any of these bits causes the corresponding GPIO signal to be configured as an output. Writing a 0 has no effect.
3–0	R/W	GPIO3–GPIO0 input enable. Writing a 1 to any of these bits causes the corresponding GPIO signal to be configured as an input. Writing a 0 has no effect.

5.7 Primary GPIO Input Data Register

The primary GPIO input data register returns the current state of the primary node GPIO terminals when read. Table 5–7 describes the bit functions in the GPIO input data register.

Bit	7	6	5	4	3	2	1	0
Name	Primary GPIO input data							
Type	R	R	R	R	R	R	R	R
Default	x	x	x	x	0	0	0	0

Register: **Primary GPIO input data**
Type: Read-only
Offset: 67h
Default: 00h

Table 5–7. Primary GPIO Input Data Register Description

BIT	ACCESS	DESCRIPTION
7–4	R	GPIO3–GPIO0 input data. These four bits return the current state of the GPIO terminals.
3–0	R	Reserved. Bits 3–0 return 0s when read.

5.8 Primary CLKOUT Control Register

The primary CLKOUT control register is used to control the primary node clock outputs. Table 5–8 describes the bit functions in the primary clock control register.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Primary CLKOUT control															
Type	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Register: **Primary CLKOUT control**
Type: Read/Write, Read-only
Offset: 68h
Default: 3FFFh

Table 5–8. Primary CLKOUT Control Register Description

BIT	ACCESS	DESCRIPTION
15–14	R	Reserved. Bits 15–14 return 0s when read.
13	R/W	CLKOUT9 disable 0 = CLKOUT9 enabled 1 = CLKOUT9 disabled and driven high (default)
12	R/W	CLKOUT8 disable 0 = CLKOUT8 enabled 1 = CLKOUT8 disabled and driven high (default)
11	R/W	CLKOUT7 disable 0 = CLKOUT7 enabled 1 = CLKOUT7 disabled and driven high (default)
10	R/W	CLKOUT6 disable 0 = CLKOUT6 enabled 1 = CLKOUT6 disabled and driven high (default)
9	R/W	CLKOUT5 disable 0 = CLKOUT5 enabled 1 = CLKOUT5 disabled and driven high (default)
8	R/W	CLKOUT4 disable 0 = CLKOUT4 enabled 1 = CLKOUT4 disabled and driven high (default)
7–6	R/W	CLKOUT3 disable 00, 01, 10 = CLKOUT3 enabled 11 = CLKOUT3 disabled and driven high (11 = default)
5–4	R/W	CLKOUT2 disable 00, 01, 10 = CLKOUT2 enabled 11 = CLKOUT2 disabled and driven high (11 = default)
3–2	R/W	CLKOUT1 disable 00, 01, 10 = CLKOUT1 enabled 11 = CLKOUT1 disabled and driven high (11 = default)
1–0	R/W	CLKOUT0 disable 00, 01, 10 = CLKOUT0 enabled 11 = CLKOUT0 disabled and driven high (11 = default)

5.9 Primary $\overline{\text{SERR}}$ Status Register

The primary $\overline{\text{SERR}}$ status register indicates the cause of an $\overline{\text{SERR}}$ event on the primary interface. Table 5–9 describes the bit functions in the primary $\overline{\text{SERR}}$ status register.

Bit	7	6	5	4	3	2	1	0
Name	Primary $\overline{\text{SERR}}$ status							
Type	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0

Register: **Primary $\overline{\text{SERR}}$ status**
Type: Read-only
Offset: 6Ah
Default: 00h

Table 5–9. Primary $\overline{\text{SERR}}$ Status Register Description

BIT	ACCESS	DESCRIPTION
7	R/W	Delayed transaction master timeout $\overline{\text{SERR}}$ status. $\overline{\text{SERR}}$ condition has occurred because the master did not repeat a read or write transaction before the master timeout counter expired on the initiator's PCI bus. Write 1 to clear this bit.
6	R/W	Delay read no data from target $\overline{\text{SERR}}$ status. $\overline{\text{SERR}}$ condition has occurred because the PCI6050 device was unable to read any data from the target after 2^{24} attempts. Write 1 to clear this bit.
5	R/W	Delayed write non-delivery $\overline{\text{SERR}}$ status. $\overline{\text{SERR}}$ condition has occurred because the PCI6050 device was unable to deliver delayed write data after 2^{24} attempts. Write 1 to clear this bit.
4	R/W	Master abort on posted writes $\overline{\text{SERR}}$ status. This bit indicates that an $\overline{\text{SERR}}$ condition occurred because a posted write request bound for either the primary or secondary bus was master-aborted when initiated by the PCI6050 device.
3	R/W	Target abort on posted writes $\overline{\text{SERR}}$ status. This bit indicates that an $\overline{\text{SERR}}$ condition occurred because a posted write request bound for either the primary or secondary bus was target-aborted when initiated by the PCI6050 device.
2	R/W	Posted write nondelivery $\overline{\text{SERR}}$ status. This bit indicates that an $\overline{\text{SERR}}$ condition occurred because the PCI6050 device was unable to deliver the posted write data to the target after 2^{24} attempts. Write 1 to clear this bit.
1	R/W	Parity error on posted writes $\overline{\text{SERR}}$ Status. This bit indicates that an $\overline{\text{SERR}}$ condition occurred because a posted write request bound for either the primary or secondary bus resulted in a parity error when the PCI6050 device initiated the transaction on the remote bus. This bit can be cleared by a write of 1.
0	R/W	Address parity error status. This bit indicates the occurrence of address parity $\overline{\text{SERR}}$ condition has occurred. This bit can be cleared by a write of 1.

5.10 Secondary CLKOUT Control Register

The secondary CLKOUT control register is used to control the secondary node clock outputs. Table 5–10 describes the bit functions in the secondary clock control register.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Secondary CLKOUT control															
Type	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Secondary CLKOUT control**
Type: Read/Write, Read-only
Offset: 6Ch
Default: 0000h

Table 5–10. Secondary CLKOUT Control Register Description

BIT	ACCESS	DESCRIPTION
15–14	R	Reserved. Bits 15–14 return 0s when read.
13	R/W	CLKOUT9 disable 0 = CLKOUT9 enabled (default) 1 = CLKOUT9 disabled and driven high
12	R/W	CLKOUT8 disable 0 = CLKOUT8 enabled (default) 1 = CLKOUT8 disabled and driven high
11	R/W	CLKOUT7 disable 0 = CLKOUT7 enabled (default) 1 = CLKOUT7 disabled and driven high
10	R/W	CLKOUT6 disable 0 = CLKOUT6 enabled (default) 1 = CLKOUT6 disabled and driven high
9	R/W	CLKOUT5 disable 0 = CLKOUT5 enabled (default) 1 = CLKOUT5 disabled and driven high
8	R/W	CLKOUT4 disable 0 = CLKOUT4 enabled (default) 1 = CLKOUT4 disabled and driven high
7–6	R/W	CLKOUT3 disable 00, 01, 10 = CLKOUT3 enabled (00 = default) 11 = CLKOUT3 disabled and driven high
5–4	R/W	CLKOUT2 disable 00, 01, 10 = CLKOUT2 enabled (00 = default) 11 = CLKOUT2 disabled and driven high
3–2	R/W	CLKOUT1 disable 00, 01, 10 = CLKOUT1 enabled (00 = default) 11 = CLKOUT1 disabled and driven high
1–0	R/W	CLKOUT0 disable 00, 01, 10 = CLKOUT0 enabled (00 = default) 11 = CLKOUT0 disabled and driven high

5.11 Secondary $\overline{\text{SERR}}$ Status Register

The secondary $\overline{\text{SERR}}$ status register indicates the cause of an $\overline{\text{SERR}}$ event on the secondary interface. Table 5–11 describes the bit functions in the secondary $\overline{\text{SERR}}$ status register.

Bit	7	6	5	4	3	2	1	0
Name	Secondary $\overline{\text{SERR}}$ status							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **Secondary $\overline{\text{SERR}}$ status**
Type: Read/Write
Offset: 6Eh
Default: 00h

Table 5–11. Secondary $\overline{\text{SERR}}$ Status Register Description

BIT	ACCESS	DESCRIPTION
7	R/W	Delayed transaction master timeout $\overline{\text{SERR}}$ status. $\overline{\text{SERR}}$ condition has occurred because master did not repeat a read or write transaction before the master timeout counter expired on the initiator's PCI bus. Write 1 to clear this bit.
6	R/W	Delay read no data from target $\overline{\text{SERR}}$ status. $\overline{\text{SERR}}$ condition has occurred because the PCI6050 device was unable to read any data from the target after 2^{24} attempts. Write 1 to clear this bit.
5	R/W	Delayed write nondelivery $\overline{\text{SERR}}$ status. $\overline{\text{SERR}}$ condition has occurred because the PCI6050 device was unable to deliver delayed write data after 2^{24} attempts. Write 1 to clear this bit.
4	R/W	Master abort on posted writes $\overline{\text{SERR}}$ status. This bit indicates that an $\overline{\text{SERR}}$ condition occurred because a posted write request bound for either the primary or secondary bus was master aborted when initiated by the PCI6050 device.
3	R/W	Target abort on posted writes $\overline{\text{SERR}}$ status. This bit indicates that an $\overline{\text{SERR}}$ condition occurred because a posted write request bound for either the primary or secondary bus was target aborted when initiated by the PCI6050 device.
2	R/W	Posted write non-delivery $\overline{\text{SERR}}$ status. $\overline{\text{SERR}}$ condition has occurred because the PCI6050 device was unable to deliver the posted write data to the target after 2^{24} attempts. Write 1 to clear this bit.
1	R/W	Parity error on posted writes $\overline{\text{SERR}}$ status. This bit indicates that an $\overline{\text{SERR}}$ condition occurred because a posted write request bound for either the primary or secondary bus resulted in a parity error when the PCI6050 device initiated the transaction on the remote bus. This bit can be cleared by a write of 1.
0	R/W	Address parity error status. This bit indicates the occurrence of address parity $\overline{\text{SERR}}$ condition has occurred. This bit can be cleared by a write of 1.

5.12 General Purpose Event Register

This register is provided for control and status of the general purpose event signal, $\overline{\text{GPE}}$. The $\overline{\text{GPE}}$, among other things, is used to indicate that there is a change of accessibility for devices on the system (for example, those behind the PCI6050 device PCI-to-PCI bridge implementation), and the $\overline{\text{GPE}}$ software handler requests appropriate PCI re-enumeration from the operating system. Table 5–12 describes the bit functions in the general purpose event register.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	General purpose event															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W
Default	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1

Register: **General purpose event**
Type: Read-only, Read/Write
Offset: A0h
Default: 1001h

Table 5–12. General Purpose Event Register Description

BIT	ACCESS	DESCRIPTION
15	R/W	Start-up complete status. This bit is set when the start-up complete bit (bit 4) in the extended diagnostic status register (offset A6h, see Section 5.13) is set. This bit reflects the status of the start-up complete bit.
14	R/W	$\overline{\text{GPE}}$ on start-up complete status. This bit enables the start-up complete bit (bit 4) in the extended diagnostic status register (offset A6h, see Section 5.13) to generate a $\overline{\text{GPE}}$ event.
13	R/W	Unexpected secondary start-up status. This bit is set when an unexpected secondary start-up packet has been received.
12	R/W	$\overline{\text{GPE}}$ on unexpected secondary start-up status. This bit enables an unexpected secondary start-up packet to generate a $\overline{\text{GPE}}$ event.
11	R/W	$\overline{\text{GPE}}$ on receiver error. This bit enables the receive error count register (offset BFh, see Section 5.19).
10	R/W	$\overline{\text{GPE}}$ on active state status. This bit enables the PCI6050 device to generate a $\overline{\text{GPE}}$ event when the serial link goes active.
9	R/W	Force $\overline{\text{GPE}}$. When software writes a 1 to this bit, a $\overline{\text{GPE}}$ is generated, and this bit is automatically cleared, returning 0 when read.
8	R/W	$\overline{\text{GPE}}$ service scratch bit. This bit is implemented as read/write, and provides no additional functionality.
7–3	R	Reserved. Bits 7–3 return 0s when read.
2	R/W	Active state status. This bit is set when the PCI6050 node is active to start the transaction and the link active bit (bit 3) in the extended diagnostic status register (offset A6h, see Section 5.13) has transitioned to 1.
1	R/W	$\overline{\text{GPE}}$ status. This bit is set when any of the potential $\overline{\text{GPE}}$ events occur and the associated enable bit is set. When this bit is set, a $\overline{\text{GPE}}$ is signaled if the $\overline{\text{GPE}}$ enable bit (bit 0) is set.
0	R/W	$\overline{\text{GPE}}$ enable. This bit is the output enable for the $\overline{\text{GPE}}$ output signal. When this bit and an enabled event occurs, a general-purpose event is signaled through the $\overline{\text{GPE}}$ output.

5.13 Extended Diagnostic Status Register

This register is provided for diagnostic purposes. Table 5–13 describes the bit functions in the extended diagnostic status register.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Extended diagnostic status															
Type	R/W	R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x

Register: **Extended diagnostic status**
Type: Read/Write, Read-only
Offset: A6h
Default: 0000h

Table 5–13. Extended Diagnostic Status Register Description

BIT	ACCESS	DESCRIPTION
15	R/W	Hot-swap switch status. Returns the logical value of HS_SWITCH input (pin 108). 0 = Hot-swap handle closed 1 = Hot-swap handle open
14	R/W	External arbiter status. This bit indicates the status of the EXTARB input (pin 207) on the secondary node. 0 = Internal arbitration in use 1 = External arbiter present and in use
13–5	R	Reserved. Bits 13–5 return 0s when read.
4	R/W	Start-up complete. This bit reflects the current state of the serial link start-up process and is set once the start-up sequence has completed.
3	R/W	Link active. This bit is set when the serial link start-up procedure has been completed and the link is ready to transmit data.
2	R/W	Serial ROM error status. This bit is set when an error occurs on the serial ROM interface.
1	R/W	Serial ROM busy status. This bit is set when the serial ROM interface is active, or busy, and is cleared when the serial ROM interface is idle.
0	R/W	Serial ROM detect status. This bit indicates the presence of a serial ROM as determined by the state of the SCL terminal (pin 197) during reset. 0 = No serial ROM was detected. 1 = A serial ROM is present.

5.14 Secondary GPIO Output Data Register

The secondary GPIO output data register controls the data driven on the secondary node GPIO terminals configured as outputs. Table 5–14 describes the bit functions in the secondary GPIO output data register.

Bit	7	6	5	4	3	2	1	0
Name	Secondary GPIO output data							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **Secondary GPIO output data**
Type: Read/Write
Offset: A9h
Default: 00h

Table 5–14. Secondary GPIO Output Data Register Description

BIT	ACCESS	DESCRIPTION
7–4	R/W	GPIO3–GPIO0 output high. Writing a 1 to any of these bits causes the corresponding GPIO signal to be driven high. Writing a 0 has no effect. GPIO terminals programmed as inputs are not affected by these bits.
3–0	R/W	GPIO3–GPIO0 output low. Writing a 1 to any of these bits causes the corresponding GPIO signal to be driven low. Writing a 0 has no effect. GPIO terminals programmed as inputs are not affected by these bits.

5.15 Secondary GPIO Output Enable Register

The secondary GPIO output enable register controls the direction of the secondary node GPIO signal. By default all GPIO terminals are inputs. Table 5–15 describes the bit functions in the secondary GPIO output enable register.

Bit	7	6	5	4	3	2	1	0
Name	Secondary GPIO output data							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **Secondary GPIO output data**
Type: Read/Write
Offset: AAh
Default: 00h

Table 5–15. Secondary GPIO Output Enable Register Description

BIT	ACCESS	DESCRIPTION
7–4	R/W	GPIO3–GPIO0 output enable. Writing a 1 to any of these bits causes the corresponding GPIO signal to be configured as an output. Writing a 0 has no effect.
3–0	R/W	GPIO3–GPIO0 input enable. Writing a 1 to any of these bits causes the corresponding GPIO signal to be configured as an input. Writing a 0 has no effect.

5.16 Secondary GPIO Input Data Register

The secondary GPIO input data register returns the current state of the secondary node GPIO terminals when read. Table 5–16 describes the bit functions in the secondary GPIO input data register.

Bit	7	6	5	4	3	2	1	0
Name	Secondary GPIO input data							
Type	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0

Register: **Secondary GPIO input data**
Type: Read-only
Offset: ABh
Default: 00h

Table 5–16. Secondary GPIO Input Data Register Description

BIT	ACCESS	DESCRIPTION
7–4	R	GPIO3–GPIO0 input data. These four bits return the current state of the GPIO terminals.
3–0	R	Reserved. Bits 3–0 return 0s when read.

5.17 Sequence Error Count Register

The sequence error count register provides the number of packet sequence errors that have occurred on the serial interface. The counter stops after 255 errors are encountered and can be cleared by writing FFh to this location.

Bit	7	6	5	4	3	2	1	0
Name	Sequence error count							
Type	RC	RC	RC	RC	RC	RC	RC	RC
Default	0	0	0	0	0	0	0	0

Register: **Sequence error count**
Type: Read-clear
Offset: BDh
Default: 00h

5.18 CRC Error Count Register

The CRC error count register provides the number of CRC errors that have occurred on the serial interface. The counter stops after 255 errors are encountered and can be cleared by writing FFh to this location.

Bit	7	6	5	4	3	2	1	0
Name	CRC error count							
Type	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0

Register: **CRC error count**
Type: Read-only
Offset: BEh
Default: 00h

5.19 Receive Error Count Register

The receive error count register provides the number of receive errors that were encountered by the PCI6060 on the serial interface. The counter stops after 255 errors are encountered and can be cleared by writing FFh to this location.

Bit	7	6	5	4	3	2	1	0
Name	Receive error count							
Type	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0

Register: **Receive error count**
Type: Read-only
Offset: BFh
Default: 00h

5.20 Transceiver Test Control and Status Register

The transceiver test control and status register is provided for TI diagnostic purposes only and should never be used in an actual system. Table 5–17 describes the bit functions in the transceiver test control and status register.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Transceiver test control and status															
Type	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Transceiver test control and status**
Type: Read/Write, Read-only
Offset: C0h
Default: 0000h

Table 5–17. Transceiver Test Control and Status Register Description

BIT	ACCESS	DESCRIPTION
15	R/W	Primary PRBS_PASS status. This bit reflects the state of the PRBS_PASS terminal. A value of 1 indicates that the PRBS test passed.
14	R/W	Primary PRBS_FAIL status. This bit reflects the state of the PRBS_PASS terminal. A value of 1 indicates that the PRBS test failed.
13–3	R	Reserved. Bits 13–3 return 0s when read.
2	R/W	Primary loop-back test enable. A primary PCI6050 node drives the LOOP_EN signal to a logic level consistent with the value of this bit. When this bit is set in a primary PCI6050 node, the loop-back test is enabled until this bit is cleared.
1	R/W	Primary PRBS test enable. A primary PCI6050 node drives the PRBS_EN signals to a logic level consistent with the value of this bit. When this bit is set in a primary PCI6050 node, the PRBS test is enabled until this bit is cleared.
0	R	Reserved. Bit 0 returns 0 when read.

5.21 PM Capability ID Register

The capability ID register identifies the linked list item as the register for PCI power management. The capability ID register returns 01h when read, which is the unique ID assigned by the PCI SIG for the PCI location of the capabilities pointer and the value.

Bit	7	6	5	4	3	2	1	0
Name	Capability ID							
Type	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	1

Register: **Capability ID**
Type: Read-only
Offset: DCh
Default: 01h

5.22 PM Next Item Pointer Register

The PM next item pointer register is used to indicate the next item in the linked list of PCI power management (PM) capabilities. The next item pointer returns E4h indicating that the PCI6050 device supports more than one extended capability.

Bit	7	6	5	4	3	2	1	0
Name	Next item pointer							
Type	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	1

Register: **Next item pointer**
Type: Read-only
Offset: DDh
Default: 00h

5.23 Power Management Capabilities Register

The power management capabilities register contains information on the capabilities of the PCI6050 functions related to power management. The PCI6050 function supports D0, D1, D2, and D3 power states. Table 5–18 describes the bit functions in the power management capabilities register.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Power management capabilities															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	0

Register: **Power management capabilities**
Type: Read-only
Offset: DEh
Default: 0602h

Table 5–18. Power Management Capabilities Register Description

BIT	ACCESS	DESCRIPTION
15–11	R	$\overline{\text{PME}}$ support. This five-bit field indicates the power states that the device supports asserting $\overline{\text{PME}}$. A 0 for any of these bits indicates that the PCI6050 device cannot assert the $\overline{\text{PME}}$ signal from that power state. For the PCI6050 device, these five bits return 00000b when read indicating that $\overline{\text{PME}}$ is not supported.
10	R	D2 support. Bit 10 returns 1 when read indicating that the bridge function supports the D2 device power state.
9	R	D1 support. Bit 9 returns 1 when read indicating that the bridge function supports the D1 device power state.
8–6	R	Auxiliary current. Returns 000b when read. $\overline{\text{PME}}$ generation from $D3_{cold}$ is not supported.
5	R	Device-specific initialization. This bit returns 0 when read, indicating that the bridge function does not require special initialization (beyond the standard PCI configuration header) before the generic class device driver is able to use it.
4	R	Auxiliary power source. Bit 4 returns 0 when read indicating that the PCI6050 device does not require an auxiliary power supply.
3	R	$\overline{\text{PME}}$ clock. This bit is implemented as read-only 0, because the $\overline{\text{PME}}$ signaling is not supported.
2–0	R	Version. By returning 010b when read, this field indicates that the PCI6050 device is compliant with Revision 1.1 of the *PCI Bus Power Management Specification.*

5.24 Power Management Control/Status Register

The power management control/status register determines and changes the current power state of the PCI6050 device. The contents of this register are not affected by the internally generated reset caused by the transition from the $D3_{hot}$ to D0 state. Table 5–19 describes the bit functions in the power management control/status register.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Power management control/status															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Power management control/status**
Type: Read/Write, Read-only
Offset: E0h
Default: 0000h

Table 5–19. Power Management Control/Status Register Description

BIT	ACCESS	DESCRIPTION
15	R	$\overline{PME}$ status. This bit returns a 0 when read because the PCI6050 device does not support $\overline{PME}$.
14–13	R	Data scale. This two-bit read-only field indicates the scaling factor to be used when interpreting the value of the data register. These bits return only 00b because the data register is not implemented.
12–9	R	Data select. This 4-bit field is used to select which data is to be reported through the data register and data scale field. These bits return only 0000b because the data register is not implemented.
8	R	$\overline{PME}$ enable. This bit returns 0 when read because the PCI6050 device does not support $\overline{PME}$ signaling.
7–2	R	Reserved. Bits 7–2 return 0s when read.
1–0	R/W	Power state. This two-bit field is used both to determine the current power state of a function and to set the function into a new power state. The definition of this is given below: 00 = D0 01 = D1 10 = D2 11 = $D3_{hot}$

5.25 PMCSR Bridge Support Register

The PMCSR bridge support register is required for all PCI bridges and supports PCI bridge-specific functionality. Table 5–20 describes the bit functions in the PMCSR bridge support register.

Bit	7	6	5	4	3	2	1	0
Name	PMCSR bridge support							
Type	R	R	R	R	R	R	R	R
Default	1	1	0	0	0	0	0	0

Register: **PMCSR bridge support**
Type: Read/Write, Read-only
Offset: E2h
Default: C0h

Table 5–20. PMCSR Bridge Support Register Description

BIT	ACCESS	DESCRIPTION
7	R	Bus power control enable. This bit is implemented as read-only 1, indicating the bus power state follows the PCI6050 device power state.
6	R	B2/B3 support for $D3_{hot}$. This bit is implemented as read-only 1, indicating that when the PCI6050 device is programmed to $D3_{hot}$ the secondary bus clock is stopped. The PCI6050 device stops the CLKOUT[8–0] outputs at a low state when programmed to D3. CLKOUT9 cannot be stopped through PCI power management.
5–0	R	Reserved. Bits 5–0 return 0s when read.

5.26 PM Data Register

The data register is an optional, 8-bit read-only register that provides a mechanism for the function to report state-dependent operating data such as power consumed or heat dissipation. The PCI6050 device does not implement the data register.

Bit	7	6	5	4	3	2	1	0
Name	PM data							
Type	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0

Register: **PM data**
Type: Read-only
Offset: E3h
Default: 00h

5.27 HS Capability ID Register

The HS capability ID register identifies the linked list item as the register for CPCI hot-swap (HS) capabilities. The register returns 06h when read, which is the unique ID assigned by the PICMG for PCI location of the capabilities pointer and the value.

Bit	7	6	5	4	3	2	1	0
Name	HS capability ID							
Type	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	1	1	0

Register: **HS capability ID**
Type: Read-only
Offset: E4h
Default: 06h

5.28 HS Next Item Pointer Register

The HS next item pointer register is used to indicate the next item in the linked list of CPCI hot-swap capabilities. Because the PCI6050 functions include only two capabilities list items, this register returns 0s when read.

Bit	7	6	5	4	3	2	1	0
Name	HS next item pointer							
Type	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0

Register: **HS next item pointer**
Type: Read-only
Offset: E5h
Default: 00h

5.29 Primary CPCI Hot-Swap Control and Status Register

The hot-swap control status register contains control and status information for CPCI hot-swap resources. Table 5–21 describes the bit functions in the hot-swap control status register.

Bit	7	6	5	4	3	2	1	0
Name	Primary PCI hot-swap control status							
Type	R/W	R/W	R	R	R/W	R	R/W	R
Default	0	0	0	0	0	0	0	0

Register: **Primary PCI hot-swap control status**
Type: Read/Write, Read-only
Offset: E6h
Default: 00h

Table 5–21. Primary CPCI Hot-Swap Control and Status Register Description

BIT	ACCESS	DESCRIPTION
7	R/W	Primary $\overline{\text{HS\_ENUM}}$ insertion status. When set, the $\overline{\text{HS\_ENUM}}$ output is driven by the PCI6050 device. This bit defaults to 0, and is set after a PCI reset occurs, the pre-load of serial ROM is complete, the ejector handle is closed, and bit 6 is 0. Thus, this bit is set following an insertion when the board implementing the PCI6050 device is ready for configuration. This bit cannot be set under software control.
6	R/W	Primary $\overline{\text{HS\_ENUM}}$ extraction status. When set, the $\overline{\text{HS\_ENUM}}$ output is driven by the primary PCI6050 node. This bit defaults to 0, and is set when the ejector handle is opened and bit 7 is 0. Thus, this bit is set when the board implementing the primary PCI6050 node is about to be removed. This bit cannot be set under software control.
5–4	R	Reserved. Bits 5–4 return 0s when read.
3	R/W	Primary LED ON/OFF. This bit defaults to 0, and controls the external LED indicator (HS_LED) under normal conditions. However, for a duration following a $\overline{\text{PCI\_RST}}$, the HS_LED output is driven high by the PCI6050 device and this bit will be ignored. When this bit is interpreted, a 1 drives HS_LED high, and a 0 drives HS_LED low. Following a $\overline{\text{PCI\_RST}}$, the HS_LED output is driven high by the PCI6050 device until the ejector handle is closed. When these conditions are met, the HS_LED is under software control via this bit.
2	R	Reserved. Bit 2 returns 0 when read.
1	R/W	Primary $\overline{\text{HS\_ENUM}}$ interrupt mask. This bit allows the $\overline{\text{HS\_ENUM}}$ output to be masked by software. Bits 6 and 7 are set independently from this bit. 0 = Enable $\overline{\text{HS\_ENUM}}$ output 1 = Mask $\overline{\text{HS\_ENUM}}$ output
0	R	Reserved. Bit 0 returns 0 when read.

5.30 Subsystem Vendor ID Register

The subsystem vendor ID register, used for system and option card identification purposes, may be required for certain operating systems.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Subsystem vendor ID															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Subsystem vendor ID**
Type: Read/Write
Offset: E8h
Default: 0000h

5.31 Subsystem ID Register

The subsystem ID register, used for system and option card identification purposes, may be required for certain operating systems.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Subsystem ID															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Subsystem ID**
Type: Read/Write
Offset: EAh
Default: 0000h

5.32 FIFO BIST Register

The FIFO BIST register is used for internal TI purposes only and should not be used in an actual system. Table 5–22 describes the bit functions in the FIFO BIST register.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	FIFO BIST															
Type	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FIFO BIST															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **FIFO BIST**
Offset: ECh
Type: Read/Write, Read-only
Default: 0000h

Table 5–22. FIFO BIST Register Description

BIT	ACCESS	DESCRIPTION
31	R/W	Built-in self-test enable. When this bit is set, the BIST function is enabled. This bit is self-clearing when the BIST completes.
30–26	R	Reserved. Bits 30–26 return 0s when read.
25–24	R	Pass fail code. This two-bit field is used to communicate the results of BIST. 00 = Pass 01 = Failed initialization pattern 10 = Failed test pattern write verification 11 = Failed inverted test pattern verification
23–11	R	Reserved. Bits 23–11 return 0s when read.
10–8	R	SRAM select. This three-bit field is used to communicate which of the four SRAMs have failed BIST. 000 = Transmit posted write FIFO 001 = Transmit completion 0 FIFO 010 = Transmit completion 1 FIFO 011 = Transmit completion 2 FIFO 100 = Receive posted write FIFO 101 = Receive completion 0 FIFO 110 = Receive completion 1 FIFO 111 = Receive completion 2 FIFO
7–0	R	Offset address. This field contains the 2K-byte address that is currently being accessed by the BIST logic. After a failure, this field contains the address of the failing DWORD.

IMPORTANT NOTICE

Texas Instruments and its subsidiaries (TI) reserve the right to make changes to their products or to discontinue any product or service without notice, and advise customers to obtain the latest version of relevant information to verify, before placing orders, that information being relied on is current and complete. All products are sold subject to the terms and conditions of sale supplied at the time of order acknowledgment, including those pertaining to warranty, patent infringement, and limitation of liability.

TI warrants performance of its semiconductor products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Customers are responsible for their applications using TI components.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards must be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance or customer product design. TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used. TI's publication of information regarding any third party's products or services does not constitute TI's approval, warranty or endorsement thereof.