Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

PI7C8150A

2-PORT PCI-to-PCI BRIDGE

REVISION 1.1

3545 North 1st Street, San Jose, CA 95134
Telephone: 1-877-PERICOM, (1-877-737-4266)
Fax: 408-435-1100
Internet: http://www.pericom.com

LIFE SUPPORT POLICY

Pericom Semiconductor Corporation's products are not authorized for use as critical components in life support devices or systems unless a specific written agreement pertaining to such intended use is executed between the manufacturer and an officer of PSC.

1) Life support devices or system are devices or systems which:
   a) Are intended for surgical implant into the body or
   b) Support or sustain life and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2) A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness. Pericom Semiconductor Corporation reserves the right to make changes to its products or specifications at any time, without notice, in order to improve design or performance and to supply the best possible product. Pericom Semiconductor does not assume any responsibility for use of any circuitry described other than the circuitry embodied in a Pericom Semiconductor product. The Company makes no representations that circuitry described herein is free from patent infringement or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent, patent rights or other rights, of Pericom Semiconductor Corporation.

REVISION HISTORY

Date	Revision Number	Description
10/15/03	1.00	First Release of Data Sheet
05/20/05	1.01	Corrected VDD and VSS pin assignments in Section 2.2.7. Removed pins 106 and 155 (R16 and B14) as these should be MS1 and MS0 respectively. Added Pb-free parts in the Ordering Information
04/20/06	1.1	Removed 'Advance Information' from header Removed 'solutions@pericom.com' email link Changed logos Updated compliance to *PCI Local Bus Specification*, revision 2.3

This page intentionally left blank.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

This page intentionally left blank.

1 INTRODUCTION

Product Description

The PI7C8150A is an enhanced PCI-to-PCI Bridge is designed to be fully compliant with the *PCI Local Bus Specification* Revision 2.3. Both the primary and secondary interfaces are specified to run at 32-bits and up to 66MHz (33MHz for PI7C8150A-33).

Product Features

- 32-bit Primary and Secondary Ports run up to 66MHz (33MHz for PI7C8150A-33)
- Compliant with the *PCI Local Bus Specification, Revision 2.3*
- Compliant with PCI-to-PCI Bridge Architecture Specification, Revision 1.1.
- All I/O and memory commands
  - Type 1 to Type 0 configuration conversion
  - Type 1 to Type 1 configuration forwarding
  - Type 1 configuration write to special cycle conversion
- Compliant with the *Advanced Configuration Power Interface (ACPI) Specification.*
- Compliant with the *PCI Power Management Specification, Revision 1.0.*
- Synchronous operation support
- Provides internal arbitration for one set of nine secondary bus masters
  - Programmable 2-level priority arbiter
  - Disable control for use of external arbiter
- Supports posted write buffers in all directions
- Four 128 byte FIFO's for delay transactions
- Two 128 byte FIFO's for posted memory transactions
- Enhanced address decoding
- Extended Commercial temperature range 0°C to 85°C
- IEEE 1149.1 JTAG interface support
- 3.3V core; 3.3V and 5V signaling
- 208-pin FQFP and 256-pin PBGA
  - Pb-free & Green available

2 SIGNAL DEFINITIONS

2.1 Signal Types

Signal Type	Description
I	Input Only
O	Output Only
P	Power
TS	Tri-State bi-directional
STS	Sustained Tri-State. Active LOW signal must be pulled HIGH for 1 cycle when deasserting.
OD	Open Drain

2.2 Signals

Note: Signal names that end with "_L" are active LOW.

2.2.1 PRIMARY BUS INTERFACE SIGNALS

Name	Pin #	Pin #	Type	Description
P_AD[31:0]	49, 50, 55, 57, 58, 60, 61, 63, 67, 68, 70, 71, 73, 74, 76, 77, 93, 95, 96, 98, 99, 101, 107, 109, 112, 113, 115, 116, 118, 119, 121, 122	N3, T2, T4, N5, P5, T5, N6, R5, T6, P7, T7, R7, T8, P8, R8, T9, R12, P12, T14, R13, N12, T15, P16, N15, M14, M13, M15, L13, M16, L14, L15, L16	TS	**Primary Address / Data:** Multiplexed address and data bus. Address is indicated by P_FRAME_L assertion. Write data is stable and valid when P_IRDY_L is asserted and read data is stable and valid when P_TRDY_L is asserted. Data is transferred on rising clock edges when both P_IRDY_L and P_TRDY_L are asserted. During bus idle, PI7C8150A drives P_AD to a valid logic level when P_GNT_L is asserted.
P_CBE[3:0]	64, 79, 92, 110	R6, R9, T13, N16	TS	**Primary Command/Byte Enables:** Multiplexed command field and byte enable field. During address phase, the initiator drives the transaction type on these pins. After that, the initiator drives the byte enables during data phases. During bus idle, PI7C8150A drives P_CBE[3:0] to a valid logic level when P_GNT_L is asserted.
P_PAR	90	N11	TS	**Primary Parity.** Parity is even across P_AD[31:0], P_CBE[3:0], and P_PAR (i.e. an even number of 1's). P_PAR is an input and is valid and stable one cycle after the address phase (indicated by assertion of P_FRAME_L) for address parity. For write data phases, P_PAR is an input and is valid one clock after P_IRDY_L is asserted. For read data phase, P_PAR is an output and is valid one clock after P_TRDY_L is asserted. Signal P_PAR is tri-stated one cycle after the P_AD lines are tri-stated. During bus idle, PI7C8150A drives P_PAR to a valid logic level when P_GNT_L is asserted.

Name	Pin #	Pin #	Type	Description
P_FRAME_L	80	P9	STS	**Primary FRAME (Active LOW).** Driven by the initiator of a transaction to indicate the beginning and duration of an access. The de-assertion of P_FRAME_L indicates the final data phase requested by the initiator. Before being tri-stated, it is driven to a de-asserted state for one cycle.
P_IRDY_L	82	T10	STS	**Primary IRDY (Active LOW).** Driven by the initiator of a transaction to indicate its ability to complete current data phase on the primary side. Once asserted in a data phase, it is not de-asserted until the end of the data phase. Before tri-stated, it is driven to a de-asserted state for one cycle.
P_TRDY_L	83	R10	STS	**Primary TRDY (Active LOW).** Driven by the target of a transaction to indicate its ability to complete current data phase on the primary side. Once asserted in a data phase, it is not de-asserted until the end of the data phase. Before tri-stated, it is driven to a de-asserted state for one cycle.
P_DEVSEL_L	84	P10	STS	**Primary Device Select (Active LOW).** Asserted by the target indicating that the device is accepting the transaction. As a master, PI7C8150A waits for the assertion of this signal within 5 cycles of P_FRAME_L assertion; otherwise, terminate with master abort. Before tri-stated, it is driven to a de-asserted state for one cycle.
P_STOP_L	85	T11	STS	**Primary STOP (Active LOW).** Asserted by the target indicating that the target is requesting the initiator to stop the current transaction. Before tri-stated, it is driven to a de-asserted state for one cycle.
P_LOCK_L	87	R11	STS	**Primary LOCK (Active LOW).** Asserted by the master for multiple transactions to complete.
P_IDSEL	65	P6	I	**Primary ID Select.** Used as a chip select line for Type 0 configuration access to PI7C8150A configuration space.
P_PERR_L	88	T12	STS	**Primary Parity Error (Active LOW).** Asserted when a data parity error is detected for data received on the primary interface. Before being tri-stated, it is driven to a de-asserted state for one cycle.
P_SERR_L	89	P11	OD	**Primary System Error (Active LOW).** Can be driven LOW by any device to indicate a system error condition. PI7C8150A drives this pin on: ▪ Address parity error ▪ Posted write data parity error on target bus ▪ Secondary S_SERR_L asserted ▪ Master abort during posted write transaction ▪ Target abort during posted write transaction ▪ Posted write transaction discarded ▪ Delayed write request discarded ▪ Delayed read request discarded ▪ Delayed transaction master timeout This signal requires an external pull-up resistor for proper operation.
P_REQ_L	47	P2	TS	**Primary Request (Active LOW):** This is asserted by PI7C8150A to indicate that it wants to start a transaction on the primary bus. PI7C8150A de-asserts this pin for at least 2 PCI clock cycles before asserting it again.
P_GNT_L	46	R1	I	**Primary Grant (Active LOW):** When asserted, PI7C8150A can access the primary bus. During idle and P_GNT_L asserted, PI7C8150A will drive P_AD, P_CBE, and P_PAR to valid logic levels.
P_RESET_L	43	P1	I	**Primary RESET (Active LOW):** When P_RESET_L is active, all PCI signals should be asynchronously tri-stated.

Name	Pin #	Pin #	Type	Description
P_M66EN	102	R14	I	**Primary Interface 66MHz Operation.** This input is used to specify if PI7C8150A is capable of running at 66MHz. For 66MHz operation on the Primary bus, this signal should be pulled "HIGH". For 33MHz operation on the Primary bus, this signal should be pulled LOW. In this condition, S_M66EN will be driven LOW, forcing the secondary bus to run at 33MHz also.

2.2.2 SECONDARY BUS INTERFACE SIGNALS

Name	Pin #	Pin #	Type	Description
S_AD[31:0]	206, 204, 203, 201, 200, 198, 197, 195, 192, 191, 189, 188, 186, 185, 183, 182, 165, 164, 162, 161, 159, 154, 152, 150, 147, 146, 144, 143, 141, 140, 138, 137	A4, D5, C5, A5, B5, D6, A6, C6, C7, A7, B7, C8, A8, B8, A9, C9, C12, D12, A14, B13, A15, B16, E13, C16, E14, D16, F13, E16, F14, F15, F16, G16	TS	**Secondary Address/Data:** Multiplexed address and data bus. Address is indicated by S_FRAME_L assertion. Write data is stable and valid when S_IRDY_L is asserted and read data is stable and valid when S_IRDY_L is asserted. Data is transferred on rising clock edges when both S_IRDY_L and S_TRDY_L are asserted. During bus idle, PI7C8150A drives S_AD to a valid logic level when S_GNT_L is asserted respectively.
S_CBE[3:0]	194, 180, 167, 149	B6, B9, B12, E15	TS	**Secondary Command/Byte Enables:** Multiplexed command field and byte enable field. During address phase, the initiator drives the transaction type on these pins. The initiator then drives the byte enables during data phases. During bus idle, PI7C8150A drives S_CBE[3:0] to a valid logic level when the internal grant is asserted.
S_PAR	168	A13	TS	**Secondary Parity:** Parity is even across S_AD[31:0], S_CBE[3:0], and S_PAR (i.e. an even number of 1's). S_PAR is an input and is valid and stable one cycle after the address phase (indicated by assertion of S_FRAME_L) for address parity. For write data phases, S_PAR is an input and is valid one clock after S_IRDY_L is asserted. For read data phase, S_PAR is an output and is valid one clock after S_TRDY_L is asserted. Signal S_PAR is tri-stated one cycle after the S_AD lines are tri-stated. During bus idle, PI7C8150A drives S_PAR to a valid logic level when the internal grant is asserted.
S_FRAME_L	179	A10	STS	**Secondary FRAME (Active LOW):** Driven by the initiator of a transaction to indicate the beginning and duration of an access. The de-assertion of S_FRAME_L indicates the final data phase requested by the initiator. Before being tri-stated, it is driven to a de-asserted state for one cycle.
S_IRDY_L	177	B10	STS	**Secondary IRDY (Active LOW):** Driven by the initiator of a transaction to indicate its ability to complete current data phase on the secondary side. Once asserted in a data phase, it is not de-asserted until the end of the data phase. Before tri-stated, it is driven to a de-asserted state for one cycle.
S_TRDY_L	176	C10	STS	**Secondary TRDY (Active LOW):** Driven by the target of a transaction to indicate its ability to complete current data phase on the secondary side. Once asserted in a data phase, it is not de-asserted until the end of the data phase. Before tri-stated, it is driven to a de-asserted state for one cycle.

Name	Pin #	Pin #	Type	Description
S_DEVSEL_L	175	A11	STS	**Secondary Device Select (Active LOW):** Asserted by the target indicating that the device is accepting the transaction. As a master, PI7C8150A waits for the assertion of this signal within 5 cycles of S_FRAME_L assertion; otherwise, terminate with master abort. Before tri-stated, it is driven to a de-asserted state for one cycle.
S_STOP_L	173	B11	STS	**Secondary STOP (Active LOW):** Asserted by the target indicating that the target is requesting the initiator to stop the current transaction. Before tri-stated, it is driven to a de-asserted state for one cycle.
S_LOCK_L	172	C11	STS	**Secondary LOCK (Active LOW):** Asserted by the master for multiple transactions to complete.
S_PERR_L	171	A12	STS	**Secondary Parity Error (Active LOW):** Asserted when a data parity error is detected for data received on the secondary interface. Before being tri-stated, it is driven to a de-asserted state for one cycle.
S_SERR_L	169	D11	I	**Secondary System Error (Active LOW):** Can be driven LOW by any device to indicate a system error condition.
S_REQ_L[8:0]	9, 8, 7, 6, 5, 4, 3, 2, 207	E4, E3, D2, C1, C2, D3, A2,B3, B4	I	**Secondary Request (Active LOW):** This is asserted by an external device to indicate that it wants to start a transaction on the secondary bus. The input is externally pulled up through a resistor to VDD.
S_GNT_L[8:0]	19, 18, 17, 16, 15, 14, 13, 11, 10	G1, F1, F2, G3, F4, E1, E2,F3, D1	TS	**Secondary Grant (Active LOW):** PI7C8150A asserts this pin to access the secondary bus. PI7C8150A de-asserts this pin for at least 2 PCI clock cycles before asserting it again. During idle and S_GNT_L asserted, PI7C8150A will drive S_AD, S_CBE, and S_PAR.
S_RESET_L	22	H1	O	**Secondary RESET (Active LOW):** Asserted when any of the following conditions are met: 1. Signal P_RESET_L is asserted. 2. Secondary reset bit in bridge control register in configuration space is set. When asserted, all control signals are tri-stated and zeroes are driven on S_AD, S_CBE, and S_PAR.
S_M66EN	153	D15	I/OD	**Secondary Interface 66MHz Operation:** This input is used to specify if PI7C8150A is running at 66MHz on the secondary side. When HIGH, the Secondary bus may run at 66MHz. When LOW, the Secondary bus may only run at 33MHz. If P_M66EN is pulled LOW, the S_M66EN is also driven LOW.
S_CFN_L	23	H2	I	**Secondary Bus Central Function Control Pin:** When tied LOW, it enables the internal arbiter. When tied HIGH, an external arbiter must be used. S_REQ_L[0] is reconfigured to be the secondary bus grant input, and S_GNT_L[0] is reconfigured to be the secondary bus request output. S_CFN_L has a weak internal pull-down resistor.

2.2.3 CLOCK SIGNALS

Name	Pin #	Pin #	Type	Description
P_CLK	45	M4	I	**Primary Clock Input:** Provides timing for all transactions on the primary interface.
S_CLKIN	21	H3	I	**Secondary Clock Input:** Provides timing for all transactions on the secondary interface.

Name	Pin #	Pin #	Type	Description
S_CLKOUT[9:0]	42, 41, 39, 38, 36, 35, 33, 32, 30, 29	M3, M2, N1, L4, L3, M1, L2, L1, K3, K2	O	**Secondary Clock Output:** Provides secondary clocks phase synchronous with the P_CLK in synchronous mode. When these clocks are used, one of the clock outputs must be fed back to S_CLKIN. Unused outputs may be disabled by: 1. Writing the secondary clock disable bits in the configuration space 2. Using the serial disable mask using the GPIO pins and MSK_IN 3. Terminating them electrically.

2.2.4 MISCELLANEOUS SIGNALS

Name	Pin #	Pin #	Type	Description
MSK_IN	126	K15	I	**MSK_IN - Secondary Clock Disable Serial Input (synchronous mode):** This pin is used by PI7C8150A to disable secondary clock outputs. The serial stream is received by MSK_IN, starting when P_RESET is detected deasserted and S_RESET_L is detected as being asserted. The serial data is used for selectively disabling secondary clock outputs and is shifted into the secondary clock control configuration register. This pin can be tied LOW to enable all secondary clock outputs or tied HIGH to drive all the secondary clock outputs HIGH.
P_VIO	124	K14	I	**Primary I/O Voltage:** This pin is used to determine either 3.3V or 5V signaling on the primary bus. P_VIO must be tied to 3.3V only when all devices on the primary bus use 3.3V signaling. Otherwise, P_VIO is tied to 5V.
S_VIO	135	G14	I	**Secondary I/O Voltage:** This pin is used to determine either 3.3V or 5V signaling on the secondary bus. S_VIO must be tied to 3.3V only when all devices on the secondary bus use 3.3V signaling. Otherwise, S_VIO is tied to 5V.
BPCCE	44	N2	I	**Bus/Power Clock Control Management Pin:** When this pin is tied HIGH and the PI7C8150A is placed in the $D3_{HOT}$ power state, it enables the PI7C8150A to place the secondary bus in the B2 power state. The secondary clocks are disabled and driven to 0. When this pin is tied LOW, there is no effect on the secondary bus clocks when the PI7C8150A enters the $D3_{HOT}$ power state.
CFG66 / SCAN_EN_H	125	K16	I	This is a multiplexed pin that has 2 functions. **CFG66 - 66MHz Configuration:** This pin is used to designate 66MHz operation. Tie HIGH to enable 66MHz operation or tie LOW to designate 33MHz operation. **SCAN_EN_H - Full-Scan Enable Control (synchronous mode):** When SCAN_EN_H is LOW, full-scan is in shift operation. When SCAN_EN_H is HIGH, full-scan is in parallel operation. *Note: Valid only in test mode. Pin is CFG66 in normal operation.*

MS0, MS1	155, 106	B14, R16	I	**Mode Selection:** Reserved for future features. MS0 / MS1 / Description 0 / 0 / RESERVED 0 / 1 / RESERVED 1 / 0 / Normal operation 1 / 1 / RESERVED MS0 should be set to 1 and MS1 should be set to 0 for normal operation.

2.2.5 GENERAL PURPOSE I/O INTERFACE SIGNALS

Name	Pin #	Pin #	Type	Description
GPIO[3:0]	24, 25, 27, 28	J3, J2, J1, K1	TS	**General Purpose I/O Data Pins:** The 4 general-purpose signals are programmable as either input-only or bi-directional signals by writing the GPIO output enable control register in the configuration space.

2.2.6 JTAG BOUNDARY SCAN SIGNALS

Name	Pin #	Pin #	Type	Description
TCK	133	H15	I	**Test Clock.** Used to clock state information and data into and out of the PI7C8150A during boundary scan.
TMS	132	H14	I	**Test Mode Select.** Used to control the state of the Test Access Port controller.
TDO	130	H16	O	**Test Data Output.** When SCAN_EN_H is HIGH, it is used (in conjunction with TCK) to shift data out of the Test Access Port (TAP) in a serial bit stream.
TDI	129	J15	I	**Test Data Input.** When SCAN_EN_H is HIGH, it is used (in conjunction with TCK) to shift data and instructions into the Test Access Port (TAP) in a serial bit stream.
TRST_L	134	G15	I	**Test Reset.** Active LOW signal to reset the Test Access Port (TAP) controller into an initialized state.

2.2.7 POWER AND GROUND

Name	Pin #	Pin #	Type	Description
VDD	1, 26, 34, 40, 51, 53, 56, 62, 69, 75, 81, 91, 97, 103, 105, 108, 114, 120, 131, 139, 145, 151, 157, 163, 170, 178, 184, 190, 196, 202, 208	A3, C4, C15, D7, D8, D9, D10, E6, E7, E8, E9, E10, E11, F5, F12, G4, G5, G12, G13, H4, H5, H12, H13, J4, J5, J12, J13, K4, K5, K12, K13, L5, L12, M6, M7, M8, M9, M10, M11, N7, N8, N9, N10, P13, P15, R3, T3	P	**Power:** +3.3V Digital power.
VSS	12, 20, 31, 37, 48, 52, 54, 59, 66, 72,	A1, A16, B1, B2, B15, C3,	P	**Ground:** Digital ground.

Name	Pin #	Pin #	Type	Description
	78, 86, 94, 100, 104, 111, 117, 123, 136, 142, 148, 156, 158, 160, 166, 174, 181, 187, 193, 199, 205	C13, C14, D4, D13, D14, E5, E12, F6, F7, F8, F9, F10, F11, G2, G6, G7, G8, G9, G10, G11, H6, H7, H8, H9, H10, H11, J6, J7, J8, J9, J10, J11, K6, K7, K8, K9, K10, K11, L6, L7, L8, L9, L10, L11, M5, M12, N4, N13, N14, P3, P4, P14, R2, R4, R15, T1, T16		

2.3 PIN LIST – 208-PIN FQFP

Table 2-1. Pin List – 208-pin FQFP

Pin Number	Name	Type	Pin Number	Name	Type
1	VDD	P	2	S_REQ_L[1]	I
3	S_REQ_L[2]	I	4	S_REQ_L[3]	I
5	S_REQ_L[4]	I	6	S_REQ_L[5]	I
7	S_REQ_L[6]	I	8	S_REQ_L[7]	I
9	S_REQ_L[8]	I	10	S_GNT_L[0]	TS
11	S_GNT_L[1]	TS	12	VSS	P
13	S_GNT_L[2]	TS	14	S_GNT_L[3]	TS
15	S_GNT_L[4]	TS	16	S_GNT_L[5]	TS
17	S_GNT_L[6]	TS	18	S_GNT_L[7]	TS
19	S_GNT_L[8]	TS	20	VSS	P
21	S_CLKIN	I	22	S_RESET_L	O
23	S_CFN_L	I	24	GPIO[3]	TS
25	GPIO[2]	TS	26	VDD	P
27	GPIO[1]	TS	28	GPIO[0]	TS
29	S_CLKOUT[0]	O	30	S_CLKOUT[1]	O
31	VSS	P	32	S_CLKOUT[2]	O
33	S_CLKOUT[3]	O	34	VDD	P
35	S_CLKOUT[4]	O	36	S_CLKOUT[5]	O
37	VSS	P	38	S_CLKOUT[6]	O
39	S_CLKOUT[7]	O	40	VDD	P
41	S_CLKOUT[8]	O	42	S_CLKOUT[9]	O
43	P_RESET_L	I	44	BPCCE	I
45	P_CLK	I	46	P_GNT_L	I
47	P_REQ_L	TS	48	VSS	P
49	P_AD[31]	TS	50	P_AD[30]	TS
51	VDD	P	52	VSS	P
53	VDD	P	54	VSS	P
55	P_AD[29]	TS	56	VDD	P
57	P_AD[28]	TS	58	P_AD[27]	TS
59	VSS	P	60	P_AD[26]	TS
61	P_AD[25]	TS	62	VDD	P
63	P_AD[24]	TS	64	P_CBE[3]	TS
65	P_IDSEL	I	66	VSS	P

Pin Number	Name	Type	Pin Number	Name	Type
67	P_AD[23]	TS	68	P_AD[22]	TS
69	VDD	P	70	P_AD[21]	TS
71	P_AD[20]	TS	72	VSS	P
73	P_AD[19]	TS	74	P_AD[18]	TS
75	VDD	P	76	P_AD[17]	TS
77	P_AD[16]	TS	78	VSS	P
79	P_CBE[2]	TS	80	P_FRAME_L	STS
81	VDD	P	82	P_IRDY_L	STS
83	P_TRDY_L	STS	84	P_DEVSEL_L	STS
85	P_STOP_L	STS	86	VSS	P
87	P_LOCK_L	STS	88	P_PERR_L	STS
89	P_SERR_L	STS	90	P_PAR	STS
91	VDD	P	92	P_CBE[1]	TS
93	P_AD[15]	TS	94	VSS	P
95	P_AD[14]	TS	96	P_AD[13]	TS
97	VDD	P	98	P_AD[12]	TS
99	P_AD[11]	TS	100	VSS	P
101	P_AD[10]	TS	102	P_M66EN	I
103	VDD	P	104	VSS	P
105	VDD	P	106	MS1	I
107	P_AD[9]	TS	108	VDD	P
109	P_AD[8]	TS	110	P_CBE[0]	TS
111	VSS	P	112	P_AD[7]	TS
113	P_AD[6]	TS	114	VDD	P
115	P_AD[5]	TS	116	P_AD[4]	TS
117	VSS	P	118	P_AD[3]	TS
119	P_AD[2]	TS	120	VDD	P
121	P_AD[1]	TS	122	P_AD[0]	TS
123	VSS	P	124	P_VIO	I
125	CFG66 / SCAN_EN_H	I	126	MSK_IN	I
127	RESERVED	-	128	RESERVED	-
129	TDI	I	130	TDO	O
131	VDD	P	132	TMS	I
133	TCK	I	134	TRST_L	I
135	S_VIO	I	136	VSS	P
137	S_AD[0]	TS	138	S_AD[1]	TS
139	VDD	P	140	S_AD[2]	TS
141	S_AD[3]	TS	142	VSS	P
143	S_ADD[4]	TS	144	S_AD[5]	TS
145	VDD	P	146	S_AD[6]	TS
147	S_AD[7]	TS	148	VSS	P
149	S_CBE[0]	TS	150	S_AD[8]	TS
151	VDD	P	152	S_AD[9]	TS
153	S_M66EN	I/OD	154	S_AD[10]	TS
155	MS0	I	156	VSS	P
157	VDD	P	158	VSS	P
159	S_AD[11]	TS	160	VSS	P
161	S_AD[12]	TS	162	S_AD[13]	TS
163	VDD	P	164	S_AD[14]	TS
165	S_AD[15]	TS	166	VSS	P
167	S_CBE[1]	TS	168	S_PAR	TS
169	S_SERR_L	I	170	VDD	P
171	S_PERR_L	STS	172	S_LOCK_L	STS
173	S_STOP_L	STS	174	VSS	P
175	S_DEVSEL_L	STS	176	S_TRDY_L	STS
177	S_IRDY_L	STS	178	VDD	P
179	S_FRAME_L	STS	180	S_CBE[2]	TS
181	VSS	P	182	S_AD[16]	TS
183	S_AD[17]	TS	184	VDD	P
185	S_AD[18]	TS	186	S_AD[19]	TS
187	VSS	P	188	S_AD[20]	TS
189	S_AD[21]	TS	190	VDD	P

Pin Number	Name	Type	Pin Number	Name	Type
191	S_AD[22]	TS	192	S_AD[23]	TS
193	VSS	P	194	S_CBE[3]	TS
195	S_AD[24]	TS	196	VDD	P
197	S_AD[25]	TS	198	S_AD[26]	TS
199	VSS	P	200	S_AD[27]	TS
201	S_AD[28]	TS	202	VDD	P
203	S_AD[29]	TS	204	S_AD[30]	TS
205	VSS	P	206	S_AD[31]	TS
207	S_REQ_L[0]	I	208	VDD	P

2.4 PIN LIST – 256-BALL PBGA

Table 2-2. Pin List – 256-pin PBGA

Pin Number	Name	Type	Pin Number	Name	Type	Pin Number	Name	Type
A1	VSS	P	A2	S_REQ_L[2]	I	A3	VDD	P
A4	S_AD[31]	TS	A5	S_AD[28]	TS	A6	S_AD[25]	TS
A7	S_AD[22]	TS	A8	S_AD[19]	TS	A9	S_AD[17]	TS
A10	S_FRAME_L	STS	A11	S_DEVSEL_L	STS	A12	S_PERR_L	STS
A13	S_PAR	TS	A14	S_AD[13]	TS	A15	S_AD[11]	TS
A16	VSS	P	B1	VSS	P	B2	VSS	P
B3	S_REQ_L[1]	I	B4	S_REQ_L[0]	I	B5	S_AD[27]	TS
B6	S_CBE_L[3]	TS	B7	S_AD[21]	TS	B8	S_AD[18]	TS
B9	S_CBE_L[2]	TS	B10	S_IRDY_L	STS	B11	S_STOP_L	STS
B12	S_CBE_L[1]	TS	B13	S_AD[12]	TS	B14	MS0	P
B15	VSS	P	B16	S_AD[10]	TS	C1	S_REQ_L[5]	I
C2	S_REQ_L[4]	I	C3	VSS	P	C4	VDD	P
C5	S_AD[29]	TS	C6	S_AD[24]	TS	C7	S_AD[23]	TS
C8	S_AD[20]	TS	C9	S_AD[16]	TS	C10	S_TRDY_L	STS
C11	S_LOCK_L	STS	C12	S_AD[15]	TS	C13	VSS	P
C14	VSS	P	C15	VDD	P	C16	S_AD[8]	TS
D1	S_GNT_L[0]	TS	D2	S_REQ_L[6]	I	D3	S_REQ_L[3]	I
D4	VSS	P	D5	S_AD[30]	TS	D6	S_AD[26]	TS
D7	VDD	P	D8	VDD	P	D9	VDD	P
D10	VDD	P	D11	S_SERR_L	I	D12	S_AD[14]	TS
D13	VSS	P	D14	VSS	P	D15	S_M66EN	I/OD
D16	S_AD[6]	TS	E1	S_GNT_L[3]	TS	E2	S_GNT_L[2]	TS
E3	S_REQ_L[7]	I	E4	S_REQ_L[8]	I	E5	VSS	P
E6	VDD	P	E7	VDD	P	E8	VDD	P
E9	VDD	P	E10	VDD	P	E11	VDD	P
E12	VSS	P	E13	S_AD[9]	TS	E14	S_AD[7]	TS
E15	S_CBE_L[0]	TS	E16	S_AD[4]	TS	F1	S_GNT_L[7]	TS
F2	S_GNT_L[6]	TS	F3	S_GNT_L[1]	TS	F4	S_GNT_L[4]	TS
F5	VDD	P	F6	VSS	P	F7	VSS	P
F8	VSS	P	F9	VSS	P	F10	VSS	P
F11	VSS	P	F12	VDD	P	F13	S_AD[5]	TS
F14	S_AD[3]	TS	F15	S_AD[2]	TS	F16	S_AD[1]	TS
G1	S_GNT_L[8]	TS	G2	VSS	P	G3	S_GNT_L[5]	TS
G4	VDD	P	G5	VDD	P	G6	VSS	P
G7	VSS	P	G8	VSS	P	G9	VSS	P
G10	VSS	P	G11	VSS	P	G12	VDD	P
G13	VDD	P	G14	S_VIO	I	G15	TRST_L	I
G16	S_AD[0]	TS	H1	S_RESET_L	O	H2	S_CFN_L	I
H3	S_CLKIN	I	H4	VDD	P	H5	VDD	P
H6	VSS	P	H7	VSS	P	H8	VSS	P
H9	VSS	P	H10	VSS	P	H11	VSS	P
H12	VDD	P	H13	VDD	P	H14	TMS	I
H15	TCK	I	H16	TDO	O	J1	GPIO[1]	TS

Pin Number	Name	Type	Pin Number	Name	Type	Pin Number	Name	Type
J2	GPIO[2]	TS	J3	GPIO[3]	TS	J4	VDD	P
J5	VDD	P	J6	VSS	P	J7	VSS	P
J8	VSS	P	J9	VSS	P	J10	VSS	P
J11	VSS	P	J12	VDD	P	J13	VDD	P
J14	RESERVED	-	J15	TDI	I	J16	RESERVED	-
K1	GPIO[0]	TS	K2	S_CLKOUT[0]	O	K3	S_CLKOUT[1]	O
K4	VDD	P	K5	VDD	P	K6	VSS	P
K7	VSS	P	K8	VSS	P	K9	VSS	P
K10	VSS	P	K11	VSS	P	K12	VDD	P
K13	VDD	P	K14	P_VIO	I	K15	MSK_IN	I
K16	CFG66 SCAN_EN_H	I	L1	S_CLKOUT[2]	O	L2	S_CLKOUT[3]	O
L3	S_CLKOUT[5]	O	L4	S_CLKOUT[6]	O	L5	VDD	P
L6	VSS	P	L7	VSS	P	L8	VSS	P
L9	VSS	P						
L10	VSS	P	L11	VSS	P	L12	VDD	P
L13	P_AD[4]	TS	L14	P_AD[2]	TS	L15	P_AD[1]	TS
L16	P_AD[0]	TS	M1	S_CLKOUT[4]	O	M2	S_CLKOUT[8]	O
M3	S_CLKOUT[9]	O	M4	P_CLK	I	M5	VSS	P
M6	VDD	P	M7	VDD	P	M8	VDD	P
M9	VDD	P	M10	VDD	P	M11	VDD	P
M12	VSS	P	M13	P_AD[6]	TS	M14	P_AD[7]	TS
M15	P_AD[5]	TS	M16	P_AD[3]	TS	N1	S_CLKOUT[7]	O
N2	BPCCE	I	N3	P_AD[31]	TS	N4	VSS	P
N5	P_AD[28]	TS	N6	P_AD[25]	TS	N7	VDD	P
N8	VDD	P	N9	VDD	P	N10	VDD	P
N11	P_PAR	TS	N12	P_AD[11]	TS	N13	VSS	P
N14	VSS	P	N15	P_AD[8]	TS	N16	P_CBE_L[0]	TS
P1	P_RESET_L	I	P2	P_REQ_L	TS	P3	VSS	P
P4	VSS	P	P5	P_AD[27]	TS	P6	P_IDSEL	I
P7	P_AD[22]	TS	P8	P_AD[18]	TS	P9	P_FRAME_L	STS
P10	P_DEVSEL_L	STS	P11	P_SERR_L	OD	P12	P_AD[14]	TS
P13	VDD	P	P14	VSS	P	P15	VDD	P
P16	P_AD[9]	TS	R1	P_GNT_L	I	R2	VSS	P
R3	VDD	P	R4	VSS	P	R5	P_AD[24]	TS
R6	P_CBE_L[3]	TS	R7	P_AD[20]	TS	R8	P_AD[17]	TS
R9	P_CBE_L[2]	TS	R10	P_TRDY_L	STS	R11	P_LOCK_L	STS
R12	P_AD[15]	TS	R13	P_AD[12]	TS	R14	P_M66EN	I
R15	VSS	P	R16	MS1	P	T1	VSS	P
T2	P_AD[30]	TS	T3	VDD	P	T4	P_AD[29]	TS
T5	P_AD[26]	TS	T6	P_AD[23]	TS	T7	P_AD[21]	TS
T8	P_AD[19]	TS	T9	P_AD[16]	TS	T10	P_IRDY_L	STS
T11	P_STOP_L	STS	T12	P_PERR_L	STS	T13	P_CBE_L[1]	TS
T14	P_AD[13]	TS	T15	P_AD[10]	TS	T16	VSS	P

3 PCI BUS OPERATION

This Chapter offers information about PCI transactions, transaction forwarding across PI7C8150A, and transaction termination. The PI7C8150A has two 128-byte FIFO's for buffering of upstream and downstream transactions. These hold addresses, data, commands, and byte enables that are used for write transactions. The PI7C8150A also has an additional four 128-byte FIFO's that hold addresses, data, commands, and byte enables for read transactions.

3.1 TYPES OF TRANSACTIONS

This section provides a summary of PCI transactions performed by PI7C8150A. Table 3-1 lists the command code and name of each PCI transaction. The Master and Target columns indicate support for each transaction when PI7C8150A initiates transactions as a master, on the primary (P) and secondary (S) buses, and when PI7C8150A responds to transactions as a target, on the primary (P) and secondary (S) buses.

Table 3-1. PCI Transactions

Types of Transactions		Initiates as Master		Responds as Target	
		Primary	Secondary	Primary	Secondary
0000	Interrupt Acknowledge	N	N	N	N
0001	Special Cycle	Y	Y	N	N
0010	I/O Read	Y	Y	Y	Y
0011	I/O Write	Y	Y	Y	Y
0100	Reserved	N	N	N	N
0101	Reserved	N	N	N	N
0110	Memory Read	Y	Y	Y	Y
0111	Memory Write	Y	Y	Y	Y
1000	Reserved	N	N	N	N
1001	Reserved	N	N	N	N
1010	Configuration Read	N	Y	Y	N
1011	Configuration Write	Y (Type 1 only)	Y	Y	Y (Type 1 only)
1100	Memory Read Multiple	Y	Y	Y	Y
1101	Dual Address Cycle	Y	Y	Y	Y
1110	Memory Read Line	Y	Y	Y	Y
1111	Memory Write and Invalidate	Y	Y	Y	Y

As indicated in Table 3-1, the following PCI commands are not supported by PI7C8150A:

- PI7C8150A never initiates a PCI transaction with a reserved command code and, as a target, PI7C8150A ignores reserved command codes.
- PI7C8150A does not generate interrupt acknowledge transactions. PI7C8150A ignores interrupt acknowledge transactions as a target.
- PI7C8150A does not respond to special cycle transactions. PI7C8150A cannot guarantee delivery of a special cycle transaction to downstream buses because of the broadcast nature of the special cycle command and the inability to control the transaction as a target. To generate special cycle transactions on other PCI buses, either upstream or downstream, Type 1 configuration write must be used.
- PI7C8150A neither generates Type 0 configuration transactions on the primary PCI bus nor responds to Type 0 configuration transactions on the secondary PCI buses.

3.2 SINGLE ADDRESS PHASE

A 32-bit address uses a single address phase. This address is driven on P_AD[31:0], and the bus command is driven on P_CBE[3:0]. PI7C8150A supports the linear increment address mode only, which is indicated when the lowest two address bits are equal to zero.

If either of the lowest two address bits is non-zero, PI7C8150A automatically disconnects the transaction after the first data transfer.

3.3 DEVICE SELECT (DEVSEL_L) GENERATION

PI7C8150A always performs positive address decoding (medium decode) when accepting transactions on either the primary or secondary buses. PI7C8150A never does subtractive decode.

3.4 DATA PHASE

The address phase of a PCI transaction is followed by one or more data phases. A data phase is completed when IRDY_L and either TRDY_L or STOP_L are asserted. A transfer of data occurs only when both IRDY_L and TRDY_L are asserted during the same PCI clock cycle. The last data phase of a transaction is indicated when FRAME_L is de-asserted and both TRDY_L and IRDY_L are asserted, or when IRDY_L and STOP_L are asserted. See Section 3.8 for further discussion of transaction termination.

Depending on the command type, PI7C8150A can support multiple data phase PCI transactions. For detailed descriptions of how PI7C8150A imposes disconnect boundaries, see Section 3.5.4 for write address boundaries and Section 3.6.3 read address boundaries.

3.5 WRITE TRANSACTIONS

Write transactions are treated as either posted write or delayed write transactions. Table 3-2 shows the method of forwarding used for each type of write operation.

Table 3-2. Write Transaction Forwarding

Type of Transaction	Type of Forwarding
Memory Write	Posted (except VGA memory)
Memory Write and Invalidate	Posted
Memory Write to VGA memory	Delayed
I/O Write	Delayed
Type 1 Configuration Write	Delayed

3.5.1 MEMORY WRITE TRANSACTIONS

Posted write forwarding is used for “Memory Write” and “Memory Write and Invalidate” transactions.

When PI7C8150A determines that a memory write transaction is to be forwarded across the bridge, PI7C8150A asserts DEVSEL_L with medium timing and TRDY_L in the next cycle, provided that enough buffer space is available in the posted memory write queue for the address and at least one DWORD of data. Under this condition, PI7C8150A accepts write data without obtaining access to the target bus.

The PI7C8150A can accept one DWORD of write data every PCI clock cycle. That is, no target wait state is inserted. The write data is stored in an internal posted write buffers and is subsequently delivered to the target.
The PI7C8150A continues to accept write data until one of the following events occurs:

- The initiator terminates the transaction by de-asserting FRAME# and IRDY#.
- An internal write address boundary is reached, such as a cache line boundary or an aligned 4KB boundary, depending on the transaction type.
- The posted write data buffer fills up.

When one of the last two events occurs, the PI7C8150A returns a target disconnect to the requesting initiator on this data phase to terminate the transaction.

Once the posted write data moves to the head of the posted data queue, PI7C8150A asserts its request on the target bus. This can occur while PI7C8150A is still receiving data on the initiator bus. When the grant for the target bus is received and the target bus is detected in the idle condition, PI7C8150A asserts FRAME_L and drives the stored write address out on the target bus. On the following cycle, PI7C8150A drives the first DWORD of write data and continues to transfer write data until all write data corresponding to that transaction is delivered, or until a target termination is received. As long as write data exists in the queue, PI7C8150A can drive one DWORD of write data each PCI clock cycle; that is, no master wait states are inserted. If write data is flowing through PI7C8150A and the initiator stalls, PI7C8150A will signal the last data phase for the current transaction at the target bus if the queue empties. PI7C8150A will restart the follow-on transactions if the queue has new data.

PI7C8150A ends the transaction on the target bus when one of the following conditions is met:

- All posted write data has been delivered to the target.
- The target returns a target disconnect or target retry (PI7C8150A starts another transaction to deliver the rest of the write data).
- The target returns a target abort (PI7C8150A discards remaining write data).
- The master latency timer expires, and PI7C8150A no longer has the target bus grant (PI7C8150A starts another transaction to deliver remaining write data).

Section 3.8.3.2 provides detailed information about how PI7C8150A responds to target termination during posted write transactions.

3.5.2 MEMORY WRITE AND INVALIDATE

Posted write forwarding is used for Memory Write and Invalidate transactions.

If offset 74h bits [8:7] = 11, the PI7C8150A disconnects Memory Write and Invalidate commands at aligned cache line boundaries. The cache line size value in the cache line size register gives the number of DWORD in a cache line.

If offset 74h bits [8:7] = 00, the PI7C8150A converts Memory Write and Invalidate transactions to Memory Write transactions at the destination.

If the value in the cache line size register does meet the memory write and invalidate conditions, the PI7C8150A returns a target disconnect to the initiator on a cache line boundary.

3.5.3 DELAYED WRITE TRANSACTIONS

Delayed write forwarding is used for I/O write transactions and Type 1 configuration write transactions.

A delayed write transaction guarantees that the actual target response is returned back to the initiator without holding the initiating bus in wait states.
A delayed write transaction is limited to a single DWORD data transfer.

When a write transaction is first detected on the initiator bus, and PI7C8150A forwards it as a delayed transaction, PI7C8150A claims the access by asserting DEVSEL_L and returns a target retry to the initiator. During the address phase, PI7C8150A samples the bus command, address, and address parity one cycle later. After IRDY_L is asserted, PI7C8150A also samples the first data DWORD, byte enable bits, and data parity. This information is placed into the delayed transaction queue. The transaction is queued only if no other existing delayed transactions have the same address and command, and if the delayed transaction queue is not full. When the delayed write transaction moves to the head of the delayed transaction queue and all ordering constraints with posted data are satisfied. The PI7C8150A initiates the transaction on the target bus. PI7C8150A transfers the write data to the target. If PI7C8150A receives a target retry in response to the write transaction on the target bus, it continues to repeat the write transaction until the data transfer is completed, or until an error condition is encountered.

If PI7C8150A is unable to deliver write data after 2^{24} (default) or 2^{32} (maximum) attempts, PI7C8150A will report a system error. PI7C8150A also asserts P_SERR_L if the primary SERR_L enable bit is set in the command register. See Section 6.4 for information on the assertion of P_SERR_L. When the initiator repeats the same write transaction (same command, address, byte enable bits, and data), and the completed delayed transaction is at the head of the queue, the PI7C8150A claims the access by asserting DEVSEL_L and returns TRDY_L to the initiator, to indicate that the write data was transferred. If the initiator requests multiple DWORD, PI7C8150A also asserts STOP_L in conjunction with TRDY_L to signal a target disconnect. Note that only those bytes of write data with valid byte enable bits are compared. If any of the byte enable bits are turned off (driven HIGH), the corresponding byte of write data is not compared.

If the initiator repeats the write transaction before the data has been transferred to the target, PI7C8150A returns a target retry to the initiator. PI7C8150A continues to return a target retry to the initiator until write data is delivered to the target, or until an error condition is encountered. When the write transaction is repeated, PI7C8150A does not make a new entry into the delayed transaction queue. Section 3.8.3.1 provides detailed information about how PI7C8150A responds to target termination during delayed write transactions.