



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



**PI7C7300D**  
**3-PORT PCI-to-PCI BRIDGE**  
Revision 1.01



3545 North 1<sup>ST</sup> 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 systems 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.

All other trademarks are of their respective companies.

---

**REVISION HISTORY**

<b>Revision</b>	<b>Date</b>	<b>Description</b>
1.00	09/21/2004	Initial release of datasheet
1.01	11/21/2005	Removed "Advance Information" from datasheet Renamed pin Y4 from BYPASS to BY_PASS

---

*This page intentionally left blank.*



## TABLE OF CONTENTS

<b>1</b>	<b>INTRODUCTION</b> .....	<b>11</b>
<b>2</b>	<b>BLOCK DIAGRAM</b> .....	<b>12</b>
<b>3</b>	<b>SIGNAL DEFINITIONS</b> .....	<b>13</b>
3.1	SIGNAL TYPES .....	13
3.2	PRIMARY BUS INTERFACE SIGNALS .....	13
3.3	SECONDARY BUS INTERFACE SIGNALS .....	15
3.4	CLOCK SIGNALS.....	17
3.5	MISCELLANEOUS SIGNALS .....	17
3.6	COMPACT PCI HOT-SWAP SIGNALS .....	17
3.7	JTAG BOUNDARY SCAN SIGNALS .....	18
3.8	POWER AND GROUND .....	18
3.9	PI7C7300D PBGA PIN LIST .....	18
<b>4</b>	<b>PCI BUS OPERATION</b> .....	<b>21</b>
4.1	TYPES OF TRANSACTIONS .....	21
4.2	SINGLE ADDRESS PHASE .....	22
4.3	DUAL ADDRESS PHASE .....	22
4.4	DEVICE SELECT (DEVSEL#) GENERATION .....	23
4.5	DATA PHASE .....	23
4.6	WRITE TRANSACTIONS .....	23
4.6.1	<i>MEMORY WRITE TRANSACTIONS</i> .....	24
4.6.2	<i>MEMORY WRITE AND INVALIDATE TRANSACTIONS</i> .....	25
4.6.3	<i>DELAYED WRITE TRANSACTIONS</i> .....	25
4.6.4	<i>WRITE TRANSACTION ADDRESS BOUNDARIES</i> .....	26
4.6.5	<i>BUFFERING MULTIPLE WRITE TRANSACTIONS</i> .....	26
4.6.6	<i>FAST BACK-TO-BACK WRITE TRANSACTIONS</i> .....	27
4.7	READ TRANSACTIONS.....	27
4.7.1	<i>PREFETCHABLE READ TRANSACTIONS</i> .....	27
4.7.2	<i>NON-PREFETCHABLE READ TRANSACTIONS</i> .....	27
4.7.3	<i>READ PREFETCH ADDRESS BOUNDARIES</i> .....	28
4.7.4	<i>DELAYED READ REQUESTS</i> .....	30
4.7.5	<i>DELAYED READ COMPLETION WITH TARGET</i> .....	30
4.7.6	<i>DELAYED READ COMPLETION ON INITIATOR BUS</i> .....	32
4.7.7	<i>FAST BACK-TO-BACK READ TRANSACTION</i> .....	32
4.8	CONFIGURATION TRANSACTIONS .....	33
4.8.1	<i>TYPE 0 ACCESS TO PI7C7300D</i> .....	33
4.8.2	<i>TYPE 1 TO TYPE 0 CONVERSION</i> .....	34
4.8.3	<i>TYPE 1 TO TYPE 1 FORWARDING</i> .....	35
4.8.4	<i>SPECIAL CYCLES</i> .....	36
4.9	TRANSACTION TERMINATION .....	37
4.9.1	<i>MASTER TERMINATION INITIATED BY PI7C7300D</i> .....	37
4.9.2	<i>MASTER ABORT RECEIVED BY PI7C7300D</i> .....	38
4.9.3	<i>TARGET TERMINATION RECEIVED BY PI7C7300D</i> .....	38
4.9.4	<i>TARGET TERMINATION INITIATED BY PI7C7300D</i> .....	41
4.10	CONCURRENT MODE OPERATION.....	42
<b>5</b>	<b>ADDRESS DECODING</b> .....	<b>42</b>

---

5.1	ADDRESS RANGES.....	43
5.2	I/O ADDRESS DECODING.....	43
5.2.1	I/O BASE AND LIMIT ADDRESS REGISTER.....	44
5.2.2	ISA MODE.....	45
5.3	MEMORY ADDRESS DECODING.....	45
5.3.1	MEMORY-MAPPED I/O BASE AND LIMIT ADDRESS REGISTERS.....	46
5.3.2	PREFETCHABLE MEMORY BASE AND LIMIT ADDRESS REGISTERS.....	47
5.4	VGA SUPPORT.....	48
5.4.1	VGA MODE.....	48
5.4.2	VGA SNOOP MODE.....	48
<b>6</b>	<b>TRANSACTION ORDERING.....</b>	<b>49</b>
6.1	TRANSACTIONS GOVERNED BY ORDERING RULES.....	49
6.2	GENERAL ORDERING GUIDELINES.....	50
6.3	ORDERING RULES.....	50
6.4	DATA SYNCHRONIZATION.....	52
<b>7</b>	<b>ERROR HANDLING.....</b>	<b>52</b>
7.1	ADDRESS PARITY ERRORS.....	52
7.2	DATA PARITY ERRORS.....	53
7.2.1	CONFIGURATION WRITE TRANSACTIONS TO CONFIGURATION SPACE.....	53
7.2.2	READ TRANSACTIONS.....	53
7.2.3	DELAYED WRITE TRANSACTIONS.....	54
7.2.4	POSTED WRITE TRANSACTIONS.....	56
7.3	DATA PARITY ERROR REPORTING SUMMARY.....	58
7.4	SYSTEM ERROR (SERR#) REPORTING.....	63
<b>8</b>	<b>EXCLUSIVE ACCESS.....</b>	<b>63</b>
8.1	CONCURRENT LOCKS.....	64
8.2	ACQUIRING EXCLUSIVE ACCESS ACROSS PI7C7300D.....	64
8.2.1	LOCKED TRANSACTIONS IN DOWSTREAM DIRECTION.....	64
8.2.2	LOCKED TRANSACTION IN UPSTREAM DIRECTION.....	65
8.3	ENDING EXCLUSIVE ACCESS.....	65
<b>9</b>	<b>PCI BUS ARBITRATION.....</b>	<b>66</b>
9.1	PRIMARY PCI BUS ARBITRATION.....	67
9.2	SECONDARY PCI BUS ARBITRATION.....	67
9.2.1	SECONDARY BUSARBITRATION USING THE INTERNAL ARBITER.....	67
9.2.2	PREEMPTION.....	69
9.2.3	SECONDARY BUS ARBITRATION USING AN EXTERNAL ARBITER.....	69
9.2.4	BUS PARKING.....	69
<b>10</b>	<b>COMPACT PCI HOT SWAP.....</b>	<b>70</b>
<b>11</b>	<b>CLOCKS.....</b>	<b>70</b>
11.1	PRIMARY CLOCK INPUTS.....	70
11.2	SECONDARY CLOCK OUTPUTS.....	70
<b>12</b>	<b>RESET.....</b>	<b>71</b>
12.1	PRIMARY INTERFACE RESET.....	71
12.2	SECONDARY INTERFACE RESET.....	71

---

---

<b>13</b>	<b>SUPPORTED COMMANDS .....</b>	<b>72</b>
13.1	PRIMARY INTERFACE.....	72
13.2	SECONDARY INTERFACE.....	73
<b>14</b>	<b>CONFIGURATION REGISTERS .....</b>	<b>73</b>
14.1	CONFIGURATION REGISTER 1 AND 2.....	75
14.1.1	VENDOR ID REGISTER – OFFSET 00h.....	75
14.1.2	DEVICE ID REGISTER – OFFSET 00h.....	76
14.1.3	COMMAND REGISTER – OFFSET 04h.....	76
14.1.4	STATUS REGISTER – OFFSET 04h.....	77
14.1.5	REVISION ID REGISTER – OFFSET 08h.....	78
14.1.6	CLASS CODE REGISTER – OFFEST 08h.....	78
14.1.7	CACHE LINE SIZE REGISTER – OFFSET 0Ch.....	78
14.1.8	PRIMARY LATENCY TIMER REGISTER – OFFSET 0Ch.....	78
14.1.9	HEADER TYPE REGISTER – OFFSET 0Ch.....	79
14.1.10	PRIMARY BUS NUMBER REGISTER – OFFSET 18h.....	79
14.1.11	SECONDARY (S1 or S2) BUS NUMBER REGISTER – OFFSET 18h.....	79
14.1.12	SUBORDINATE (S1 or S2) BUS NUMBER REGISTER – OFFSET 18h.....	79
14.1.13	SECONDARY LATENCY TIMER REGISTER – OFFSET 18h.....	79
14.1.14	I/O BASE REGISTER – OFFSET 1Ch.....	80
14.1.15	I/O LIMIT REGISTER – OFFSET 1Ch.....	80
14.1.16	SECONDARY STATUS REGISTER – OFFSET 1Ch.....	80
14.1.17	MEMORY BASE REGISTER – OFFSET 20h.....	81
14.1.18	MEMORY LIMIT REGISTER – OFFSET 20h.....	81
14.1.19	PREFETCHABLE MEMORY BASE REGISTER – OFFSET 24h.....	82
14.1.20	PREFETCHABLE MEMORY LIMIT REGISTER – OFFSET 24h.....	82
14.1.21	PREFETCHABLE MEMORY BASE ADDRESS UPPER 32-BITS REGISTER – OFFSET 28h.....	82
14.1.22	PREFETCHABLE MEMORY LIMIT ADDRESS UPPER 32-BITS REGISTER – OFFSET 2Ch.....	82
14.1.23	I/O BASE ADDRESS UPPER 16-BITS REGISTER – Offset 30h.....	83
14.1.24	I/O LIMIT ADDRESS UPPER 16-BITS REGISTER – OFFSET 30h.....	83
14.1.25	ECP POINTER REGISTER – OFFSET 34h.....	83
14.1.26	BRIDGE CONTROL REGISTER – OFFSET 3Ch.....	83
14.1.27	DIAGNOSTIC / CHIP CONTROL REGISTER – OFFSET 40h.....	85
14.1.28	ARBITER CONTROL REGISTER – OFFSET 40h.....	86
14.1.29	UPSTREAM MEMORY CONTROL REGISTER – OFFSET 48h.....	86
14.1.30	HOT SWAP SWITCH TIME SLOT REGISTER – OFFSET 4Ch.....	86
14.1.31	UPSTREAM (S1 or S2 to P) MEMORY BASE REGISTER – OFFSET 50h.....	87
14.1.32	UPSTREAM (S1 or S2 to P) MEMORY LIMIT REGISTER – OFFSET 50h.....	87
14.1.33	UPSTREAM (S1 or S2 to P) MEMORY BASE UPPER 32-BITS REGISTER – OFFSET 54h.....	87
14.1.34	UPSTREAM (S1 or S2 to P) MEMORY LIMIT UPPER 32 BITS REGISTER – OFFSET 58h.....	88
14.1.35	P_SERR# EVENT DISABLE REGISTER – OFFSET 64h.....	88
14.1.36	SECONDARY CLOCK CONTROL REGISTER – OFFSET 68h.....	89
14.1.37	PORT OPTION REGISTER – OFFSET 74h.....	89
14.1.38	MASTER TIMEOUT COUNTER REGISTER – OFFSET 74h.....	91
14.1.39	RETRY COUNTER REGISTER – OFFSET 78h.....	91
14.1.40	SAMPLING TIMER REGISTER – OFFSET 7Ch.....	91
14.1.41	SECONDARY SUCCESSFUL I/O READ COUNTER REGISTER – OFFSET 80h.....	91

---



14.1.42	<i>SECONDARY SUCCESSFUL I/O WRITE COUNTER REGISTER – OFFSET 84h.....</i>	91
14.1.43	<i>SECONDARY SUCCESSFUL MEMORY READ COUNTER REGISTER – Offset 88h</i>	92
14.1.44	<i>SECONDARY SUCCESSFUL MEMORY WRITE COUNTER REGISTER – OFFSET 8Ch</i>	92
14.1.45	<i>PRIMARY SUCCESSFUL I/O READ COUNTER REGISTER – OFFSET 90h.....</i>	92
14.1.46	<i>PRIMARY SUCCESSFUL I/O WRITE COUNTER REGISTER – OFFSET 94h.....</i>	92
14.1.47	<i>PRIMARY SUCCESSFUL MEMORY READ COUNTER REGISTER – OFFSET 98h..</i>	92
14.1.48	<i>PRIMARY SUCCESSFUL MEMORY WRITE COUNTER REGISTER – OFFSET 9Ch</i>	93
14.1.49	<i>CAPABILITY ID REGISTER – OFFSET B0h.....</i>	93
14.1.50	<i>NEXT POINTER REGISTER – OFFSET B0h.....</i>	93
14.1.51	<i>SLOT NUMBER REGISTER – OFFSET B0h.....</i>	93
14.1.52	<i>CHASSIS NUMBER REGISTER – OFFSET B0h.....</i>	93
14.1.53	<i>CAPABILITY ID REGISTER – OFFSET C0h.....</i>	94
14.1.54	<i>NEXT POINTER REGISTER – OFFSET C0h.....</i>	94
14.1.55	<i>HOT SWAP CONTROL AND STATUS REGISTER – OFFSET C0h.....</i>	94
<b>15</b>	<b>BRIDGE BEHAVIOR.....</b>	<b>94</b>
15.1	BRIDGE ACTIONS FOR VARIOUS CYCLE TYPES .....	95
15.2	TRANSACTION ORDERING .....	95
15.3	ABNORMAL TERMINATION (INITIATED BY BRIDGE MASTER) .....	96
15.3.1	<i>MASTER ABORT.....</i>	96
15.3.2	<i>PARITY AND ERROR REPORTING.....</i>	96
15.3.3	<i>REPORTING PARITY ERRORS.....</i>	96
15.3.4	<i>SECONDARY IDSEL MAPPING .....</i>	97
<b>16</b>	<b>IEEE 1149.1 COMPATIBLE JTAG CONTROLLER .....</b>	<b>97</b>
16.1	BOUNDARY SCAN ARCHITECTURE .....	97
16.1.1	<i>TAP PINS.....</i>	98
16.1.2	<i>INSTRUCTION REGISTER.....</i>	98
16.2	BOUNDARY-SCAN INSTRUCTION SET .....	99
16.3	TAP TEST DATA REGISTERS.....	99
16.4	BYPASS REGISTER.....	100
16.5	BOUNDARY-SCAN REGISTER .....	100
16.6	TAP CONTROLLER .....	100
<b>17</b>	<b>ELECTRICAL AND TIMING SPECIFICATIONS.....</b>	<b>103</b>
17.1	MAXIMUM RATINGS .....	103
17.2	3.3V DC SPECIFICATIONS.....	104
17.3	3.3V AC SPECIFICATIONS.....	105
17.4	PRIMARY AND SECONDARY BUSES AT 66MHz CLOCK TIMING .....	106
17.5	PRIMARY AND SECONDARY BUSES AT 33MHz CLOCK TIMING .....	106
17.6	POWER CONSUMPTION .....	106
<b>18</b>	<b>272-PIN PBGA PACKAGE FIGURE .....</b>	<b>107</b>
18.1	PART NUMBER ORDERING INFORMATION .....	107

## LIST OF TABLES

TABLE 4-1 PCI TRANSACTIONS .....	21
TABLE 4-2 WRITE TRANSACTION FORWARDING .....	23
TABLE 4-3 WRITE TRANSACTION DISCONNECT ADDRESS BOUNDARIES .....	26
TABLE 4-4 READ PREFETCH ADDRESS BOUNDARIES .....	29
TABLE 4-5 READ TRANSACTION PREFETCHING .....	30
TABLE 4-6 DEVICE NUMBER TO IDSEL S1_AD OR S2_AD PIN MAPPING .....	34
TABLE 4-7 DELAYED WRITE TARGET TERMINATION RESPONSE .....	39
TABLE 4-8 RESPONSE TO POSTED WRITE TARGET TERMINATION .....	39
TABLE 4-9 RESPONSE TO DELAYED READ TARGET TERMINATION .....	40
TABLE 6-1 SUMMARY OF TRANSACTION ORDERING .....	50
TABLE 7-1 SETTING THE PRIMARY INTERFACE DETECTED PARITY ERROR BIT .....	58
TABLE 7-2 SETTING SECONDARY INTERFACE DETECTED PARITY ERROR BIT .....	58
TABLE 7-3 SETTING PRIMARY INTERFACE DATA PARITY ERROR DETECTED BIT .....	59
TABLE 7-4 SETTING SECONDARY INTERFACE DATA PARITY ERROR DETECTED BIT .....	59
TABLE 7-5 ASSERTION OF P_PERR# .....	60
TABLE 7-6 ASSERTION OF S_PERR# .....	62
TABLE 7-7 ASSERTION OF P_SERR# FOR DATA PARITY ERRORS .....	62
TABLE 16-1 TAP PINS .....	99
TABLE 16-2 JTAG BOUNDARY REGISTER ORDER .....	101

## LIST OF FIGURES

FIGURE 9-1 SECONDARY ARBITER EXAMPLE .....	68
FIGURE 16-1 TEST ACCESS PORT BLOCK DIAGRAM .....	98
FIGURE 17-1 PCI SIGNAL TIMING MEASUREMENT CONDITIONS .....	105
FIGURE 18-1 272-PIN PBGA PACKAGE .....	107

---

*This page intentionally left blank.*

# 1 INTRODUCTION

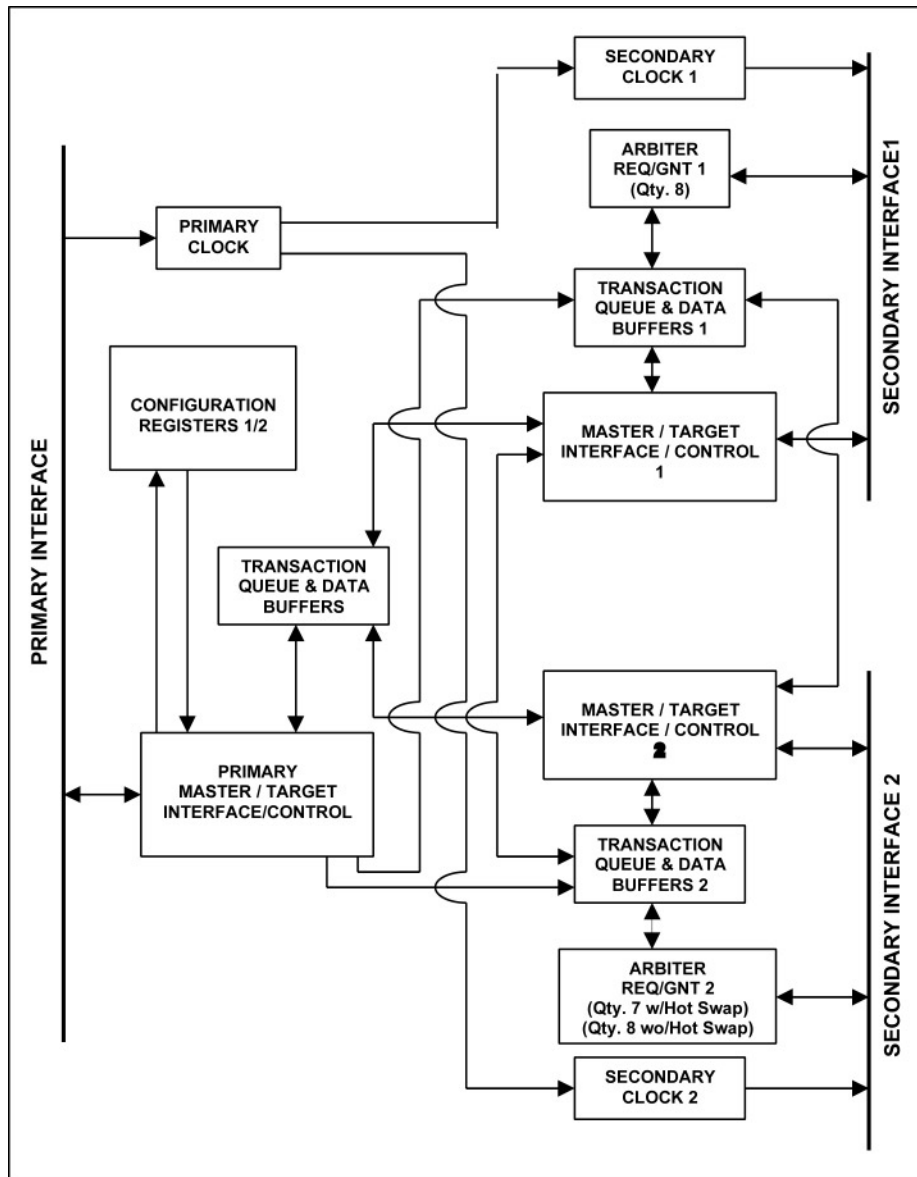
## PRODUCT DESCRIPTION

The PI7C7300D is Pericom Semiconductor's second-generation PCI-PCI Bridge and is an updated revision to the PI7C7300A. It is designed to be fully compliant with the 32-bit, 66MHz implementation of the *PCI Local Bus Specification, Revision 2.2*. The PI7C7300D supports only synchronous bus transactions between devices on the Primary Bus running at 33MHz to 66MHz and the Secondary Buses operating at either 33MHz or 66MHz. The Primary and Secondary Buses can also operate in concurrent mode, resulting in added increase in system performance. Concurrent bus operation off-loads and isolates unnecessary traffic from the Primary Bus; thereby enabling a master and a target device on the same Secondary PCI Bus to communicate even while the Primary Bus is busy. In addition, the Secondary Buses have load balancing capability, allowing faster devices to be isolated away from slower devices. Among the other features supported by the PI7C7300D are: support for up to 15 devices on the Secondary Buses, Compact PCI Hot Swap (PICMG 2.1, R1.0) *Friendly* Support and Dual Addressing Cycle.

## PRODUCT FEATURES

- 32-bit Primary and Two Secondary Ports run up to 66MHz
- All 3 ports compliant with the *PCI Local Bus Specification, Revision 2.2*
- 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
- Concurrent Primary to Secondary Bus operation and independent intra-Secondary Port channel to reduce traffic on the Primary Port
- Provides internal arbitration for one set of eight secondary bus masters (S1 bus) and one set of seven (eight if Hot Swap is disabled) secondary bus masters (S2 bus)
  - Programmable 2-level priority arbiter
  - Disable control for use of external arbiter
- Supports posted write buffers in all directions
- Three 128 byte FIFO's for delay transactions
- Three 128 byte FIFO's for posted memory transactions
- Enhanced address decoding
  - 32-bit I/O address range
  - 32-bit memory-mapped I/O address range
  - VGA addressing and VGA palette snooping
  - ISA-aware mode for legacy support in the first 64KB of I/O address range
- Dual Addressing cycle (64-bit)
- Supports system transaction ordering rules
- Tri-state control of output buffers on secondary buses
- Compact PCI Hot Swap (PICMG 2.1, R1.0) *Friendly* Support
- Industrial Temperature range -40°C to 85°C
- IEEE 1149.1 JTAG interface support
- 3.3V core; 3.3V PCI I/O interface with 5V I/O tolerance
- 272-pin plastic BGA package

**2 BLOCK DIAGRAM**





### 3 SIGNAL DEFINITIONS

#### 3.1 SIGNAL TYPES

Signal Type	Description
PI	PCI input (3.3V, 5V tolerant)
PIU	PCI input (3.3V, 5V tolerant) with weak pull-up
PID	PCI input (3.3V, 5V tolerant) with weak pull-down
PO	PCI output (3.3V)
PB	PCI tri-state bidirectional (3.3V, 5V tolerant)
PSTS	PCI sustained tri-state bi-directional (Active LOW signal which must be driven inactive for one cycle before being tri-stated to ensure HIGH performance on a shared signal line)
PTS	PCI tri-state output
POD	PCI output which either drives LOW (active state) or tri-state

#### 3.2 PRIMARY BUS INTERFACE SIGNALS

Name	Pin #	Type	Description
P_AD[31:0]	Y7, W7, Y8, W8, V8, U8, Y9, W9, W10, V10, Y11, V11, U11, Y12, W12, V12, V16, W16, Y16, W17, Y17, U18, W18, Y18, U19, W19, Y19, U20, V20, Y20, T17, R17	PB	<b>Primary Address/Data.</b> Multiplexed address and data bus. Address is indicated by P_FRAME# assertion. Write data is stable and valid when P_IRDY# is asserted and read data is stable and valid when P_TRDY# is asserted. Data is transferred on rising clock edges when both P_IRDY# and P_TRDY# are asserted. During bus idle, PI7C7300D drives P_AD to a valid logic level when P_GNT# is asserted.
P_CBE[3:0]	V9, U12, U16, V19	PB	<b>Primary Command/Byte Enables.</b> 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, PI7C7300D drives P_CBE[3:0] to a valid logic level when P_GNT# is asserted.
P_PAR	U15	PB	<b>Primary Parity.</b> 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#) for address parity. For write data phases, P_PAR is an input and is valid one clock after P_IRDY# is asserted. For read data phase, P_PAR is an output and is valid one clock after P_TRDY# is asserted. Signal P_PAR is tri-stated one cycle after the P_AD lines are tri-stated. During bus idle, PI7C7300D drives P_PAR to a valid logic level when P_GNT# is asserted.
P_FRAME#	W13	PSTS	<b>Primary FRAME (Active LOW).</b> Driven by the initiator of a transaction to indicate the beginning and duration of an access. The de-assertion of P_FRAME# indicates the final data phase requested by the initiator. Before being tri-stated, it is driven to a de-asserted state for one cycle.

Name	Pin #	Type	Description
P_IRDY#	V13	PSTS	<b>Primary IRDY (Active LOW).</b> 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#	U13	PSTS	<b>Primary TRDY (Active LOW).</b> 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#	Y14	PSTS	<b>Primary Device Select (Active LOW).</b> Asserted by the target indicating that the device is accepting the transaction. As a master, PI7C7300D waits for the assertion of this signal within 5 cycles of P_FRAME# assertion; otherwise, terminate with master abort. Before tri-stated, it is driven to a de-asserted state for one cycle.
P_STOP#	W14	PSTS	<b>Primary STOP (Active LOW).</b> 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#	V14	PSTS	<b>Primary LOCK (Active LOW).</b> Asserted by the master for multiple transactions to complete.
P_IDSEL	Y10	PI	<b>Primary ID Select.</b> Used as a chip select line for Type 0 configuration accesses to PI7C7300D configuration space.
P_PERR#	Y15	PSTS	<b>Primary Parity Error (Active LOW).</b> 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#	W15	POD	<b>Primary System Error (Active LOW).</b> Can be driven LOW by any device to indicate a system error condition. PI7C7300D drives this pin on: <ul style="list-style-type: none"> <li>▪ Address parity error</li> <li>▪ Posted write data parity error on target bus</li> <li>▪ Secondary S1_SERR# or S2_SERR# asserted</li> <li>▪ Master abort during posted write transaction</li> <li>▪ Target abort during posted write transaction</li> <li>▪ Posted write transaction discarded</li> <li>▪ Delayed write request discarded</li> <li>▪ Delayed read request discarded</li> <li>▪ Delayed transaction master timeout</li> </ul> This signal requires an external pull-up resistor for proper operation.
P_REQ#	W6	PTS	<b>Primary Request (Active LOW).</b> This is asserted by PI7C7300D to indicate that it wants to start a transaction on the primary bus. PI7C7300D de-asserts this pin for at least 2 PCI clock cycles before asserting it again.
P_GNT#	U7	PI	<b>Primary Grant (Active LOW).</b> When asserted, PI7C7300D can access the primary bus. During idle and P_GNT# asserted, PI7C7300D will drive P_AD, P_CBE, and P_PAR to valid logic levels.
P_RESET#	Y5	PI	<b>Primary RESET (Active LOW).</b> When P_RESET# is active, all PCI signals should be asynchronously tri-stated.

Name	Pin #	Type	Description
P_M66EN	V18	PI	<b>Primary Interface 66MHz Operation.</b> This input is used to specify if PI7C7300D 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, S1_M66EN and S2_M66EN will both need to be "LOW", forcing both secondary buses to run at 33MHz also.

### 3.3 SECONDARY BUS INTERFACE SIGNALS

Name	Pin #	Type	Description
S1_AD[31:0],  S2_AD[31:0]	B20, B19, C20, C19, C18, D20, D19, D17, E19, E18, E17, F20, F19, F17, G20, G19, L20, L19, L18, M20, M19, M17, N20, N19, N18, N17, P17, R20, R19, R18, T20, T19  J4, H1, H2, H3, H4, G1, G3, G4, F2, F3, F4, E1, E4, D1, C1, B1, C5, B5, D6, C6, B6, A6, C7, B7, D8, C8, D9, C9, B9, A9, D10, C10	PB	<b>Secondary Address/Data.</b> Multiplexed address and data bus. Address is indicated by S1_FRAME# or S2_FRAME# assertion. Write data is stable and valid when S1_IRDY# or S2_IRDY# is asserted and read data is stable and valid when S1_IRDY# or S2_IRDY# is asserted. Data is transferred on rising clock edges when both S1_IRDY# or S2_IRDY# and S1_TRDY# or S2_TRDY# are asserted. During bus idle, PI7C7300D drives S1_AD or S2_AD to a valid logic level when S1_GNT# or S2_GNT# is asserted respectively.
S1_CBE[3:0], S2_CBE[3:0]	E20, G18, K17, P20 F1, A1, A4, A7	PB	<b>Secondary Command/Byte Enables.</b> 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, PI7C7300D drives S1_CBE[3:0] or S2_CBE[3:0] to a valid logic level when the internal grant is asserted.
S1_PAR, S2_PAR	K18, B4	PB	<b>Secondary Parity.</b> Parity is even across S1_AD[31:0], S1_CBE[3:0], and S1_PAR or S2_AD[31:0], S2_CBE[3:0], and S2_PAR (i.e. an even number of 1's). S1_PAR or S2_PAR is an input and is valid and stable one cycle after the address phase (indicated by assertion of S1_FRAME# or S2_FRAME#) for address parity. For write data phases, S1_PAR or S2_PAR is an input and is valid one clock after S1_IRDY# or S2_IRDY# is asserted. For read data phase, S1_PAR or S2_PAR is an output and is valid one clock after S1_TRDY# or S2_TRDY# is asserted. Signal S1_PAR or S2_PAR is tri-stated one cycle after the S1_AD or S2_AD lines are tri-stated. During bus idle, PI7C7300D drives S1_PAR or S2_PAR to a valid logic level when the internal grant is asserted.
S1_FRAME#, S2_FRAME#	H20, D2	PSTS	<b>Secondary FRAME (Active LOW).</b> Driven by the initiator of a transaction to indicate the beginning and duration of an access. The de-assertion of S1_FRAME# or S2_FRAME# indicates the final data phase requested by the initiator. Before being tri-stated, it is driven to a de-asserted state for one cycle.

Name	Pin #	Type	Description
S1_IRDY#, S2_IRDY#	H19, B2	PSTS	<b>Secondary IRDY (Active LOW).</b> 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.
S1_TRDY#, S2_TRDY#	H18, A2	PSTS	<b>Secondary TRDY (Active LOW).</b> 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.
S1_DEVSEL#, S2_DEVSEL#	J20, D3	PSTS	<b>Secondary Device Select (Active LOW).</b> Asserted by the target indicating that the device is accepting the transaction. As a master, PI7C7300D waits for the assertion of this signal within 5 cycles of S1_FRAME# or S2_FRAME# assertion; otherwise, terminate with master abort. Before tri-stated, it is driven to a de-asserted state for one cycle.
S1_STOP#, S2_STOP#	J19, C3	PSTS	<b>Secondary STOP (Active LOW).</b> 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.
S1_LOCK#, S2_LOCK#	J18, B3	PSTS	<b>Secondary LOCK (Active LOW).</b> Asserted by the master for multiple transactions to complete.
S1_PERR#, S2_PERR#	J17, D4	PSTS	<b>Secondary Parity Error (Active LOW).</b> 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.
S1_SERR#, S2_SERR#	K20, C4	PI	<b>Secondary System Error (Active LOW).</b> Can be driven LOW by any device to indicate a system error condition.
S1_REQ#[7:0], S2_REQ#[6:0]	B11, A12, D13, C13, C15, A16, C17, B17 R3, P2, P1, M2, M1, K1, K3	PIU	<b>Secondary Request (Active LOW).</b> 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.
S1_GNT#[7:0] S2_GNT#[6:0]	C11, B12, B13, A14, D14, B16, D16, B18 P4, R1, N4, M3, L4, L1, K2	PO	<b>Secondary Grant (Active LOW).</b> PI7C7300D asserts this pin to access the secondary bus. PI7C7300D de-asserts this pin for at least 2 PCI clock cycles before asserting it again. During idle and S1_GNT# or S2_GNT# asserted, PI7C7300D will drive S1_AD, S1_CBE, and S1_PAR or S2_AD, S2_CBE, and S2_PAR.
S1_RESET#, S2_RESET#	B10, T4	PO	<b>Secondary RESET (Active LOW).</b> Asserted when any of the following conditions are met: <ol style="list-style-type: none"> <li>Signal P_RESET# is asserted.</li> <li>Secondary reset bit in bridge control register in configuration space is set.</li> </ol> When asserted, all control signals are tri-stated and zeroes are driven on S1_AD, S1_CBE, and S1_PAR or S2_AD, S2_CBE, and S2_PAR.
S1_EN, S2_EN	W3, W4	PIU	<b>Secondary Enable (Active HIGH).</b> When S1_EN or S2_EN is inactive, secondary bus PCI S1 or PCI S2 will be asynchronously tri-stated.
S1_M66EN, S2_M66EN	D7, W5	PI	<b>Secondary Interface 66MHz Operation.</b> This input is used to specify if PI7C7300D is capable of running at 66MHz on the secondary side. When HIGH, the S1 or S2 bus may run at 66MHz. When LOW, the S1 or S2 bus may only run at 33MHz. If P_M66EN is pulled LOW, both S1_M66EN and S2_M66EN need to be LOW.

Name	Pin #	Type	Description
S_CFN#	Y2	PIU	<b>Secondary Bus Central Function Control Pin.</b> When tied LOW, it enables the internal arbiter. When tied HIGH, an external arbiter must be used. S1_REQ#[0] or S2_REQ#[0] is reconfigured to be the secondary bus grant input, and S1_GNT#[0] or S2_GNT#[0] is reconfigured to be the secondary bus request output.

### 3.4 CLOCK SIGNALS

Name	Pin #	Type	Description
P_CLK	V6	PI	<b>Primary Clock Input.</b> Provides timing for all transactions on the primary interface.
S1_CLKOUT [7:0]	A11, C12, A13, B14, B15, C16, A18, A19	PTS	<b>Secondary Clock Output.</b> Provides secondary 1 clocks phase synchronous with the P_CLK.
S2_CLKOUT [7:0]	T3, T1, P3, N3, M4, L3, L2, J1	PTS	<b>Secondary Clock Output.</b> Provides secondary 2 clocks phase synchronous with the P_CLK.

### 3.5 MISCELLANEOUS SIGNALS

Name	Pin #	Type	Description
BY_PASS	Y4	PI	<b>Reserved.</b> Reserved for future use. Must be tied HIGH.
PLL_TM	Y3	PI	<b>Reserved.</b> Reserved for future use. Must be tied LOW.
S_CLKIN	V5	PI	<b>Reserved.</b> Reserved for future use. Must be tied LOW.
SCAN_TM#	V4	PI	<b>Full-Scan Test Mode Enable (Active LOW).</b> Connect HIGH for normal operation. When SCAN_TM# is active, the ten scan chains will be enabled. The scan clock is P_CLK. The scan input and outputs are as follows: S1_REQ[6], S1_REQ[5], S1_REQ[4], S1_REQ[3], S1_REQ[2], S2_REQ#[6], S2_REQ#[5], S2_REQ#[4], S2_REQ#[3], S2_REQ#[2], and S1_GNT#[6], S1_GNT#[5], S1_GNT#[4], S1_GNT#[3], S1_GNT#[2], S2_GNT#[6], S2_GNT#[5], S2_GNT#[4], S2_GNT#[3], S2_GNT#[2]
SCAN_EN	U5	PID	<b>Full-Scan Enable Control.</b> SCAN_EN should be tied LOW in normal mode. When SCAN_EN is LOW, full-scan is in shift operation if SCAN_TM# is active. When SCAN_EN is HIGH, full-scan is in parallel operation if SCAN_TM# is active.

### 3.6 COMPACT PCI HOT-SWAP SIGNALS

Name	Pin #	Type	Description
LOO	U1	PO	<b>Hot Swap LED.</b> The output of this pin lights a blue LED to indicate insertion and removal ready status. If HS_EN is LOW, pin is S2_GNT#[7].
HS_SW#	T2	PI	<b>Hot Swap Switch.</b> When driven LOW, this signal indicates that the board ejector handle indicates an insertion or impending extraction of a board. If HS_EN is LOW, pin is S2_REQ#[7].



Name	Pin #	Type	Description
HS_EN	U6	PI	<b>Hot Swap Enable.</b> To enable Hot Swap Friendly support, this signal should be pulled HIGH.
ENUM#	R4	POD	<b>Hot Swap Status Indicator.</b> The output of ENUM# indicates to the system that an insertion has occurred of that an extraction is about to occur.

### 3.7 JTAG BOUNDARY SCAN SIGNALS

Name	Pin #	Type	Description
TCK	V2	PIU	<b>Test Clock.</b> Used to clock state information and data into and out of the PI7C7300D during boundary scan.
TMS	W1	PIU	<b>Test Mode Select.</b> Used to control the state of the Test Access Port controller.
TDO	V3	PTS	<b>Test Data Output.</b> When SCAN_EN 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	W2	PIU	<b>Test Data Input.</b> When SCAN_EN 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#	U3	PIU	<b>Test Reset.</b> Active LOW signal to reset the Test Access Port (TAP) controller into an initialized state.

### 3.8 POWER AND GROUND

Name	Pin #	Type	Description
VDD	B8, C14, D5, D11, D15, E2, F18, J3, L17, N2, P19, U10, V1, V7, V15, W20		<b>3.3V Digital Power</b>
VSS	A3, A5, A8, A10, A15, A17, A20, C2, D12, D18, E3, G2, G17, H17, J2, J9, J10, J11, J12, K4, K9, K10, K11, K12, K19, L9, L10, L11, L12, M9, M10, M11, M12, M18, N1, P18, R2, T18, U2, U9, U14, U17, V17, W11, Y6, Y13		<b>Digital Ground</b>
AVCC	Y1		<b>Analog 3.3V for PLL</b>
AGND	U4		<b>Analog Ground for PLL</b>

### 3.9 PI7C7300D PBGA PIN LIST

Pin #	Name	Type	Pin #	Name	Type
A1	S2_CBE[2]	PB	A2	S2_TRDY#	PSTS
A3	VSS	-	A4	S2_CBE[1]	PB
A5	VSS	-	A6	S2_AD[10]	PB

Pin #	Name	Type	Pin #	Name	Type
A7	S2_CBE[0]	PB	A8	VSS	-
A9	S2_AD[2]	PB	A10	VSS	-
A11	S1_CLKOUT[7]	PTS	A12	S1_REQ#[6]	PIU
A13	S1_CLKOUT[5]	PTS	A14	S1_GNT#[4]	PO
A15	VSS	-	A16	S1_REQ#[2]	PIU
A17	VSS	-	A18	S1_CLKOUT[1]	PTS
A19	S1_CLKOUT[0]	PTS	A20	VSS	-
B1	S2_AD[16]	PB	B2	S2_IRDY#	PSTS
B3	S2_LOCK#	PSTS	B4	S2_PAR	PB
B5	S2_AD[14]	PB	B6	S2_AD[11]	PB
B7	S2_AD[8]	PB	B8	VDD	-
B9	S2_AD[3]	PB	B10	S1_RESET#	PO
B11	S1_REQ#[7]	PIU	B12	S1_GNT#[6]	PO
B13	S1_GNT#[5]	PO	B14	S1_CLKOUT[4]	PTS
B15	S1_CLKOUT[3]	PTS	B16	S1_GNT#[2]	PO
B17	S1_REQ#[0]	PIU	B18	S1_GNT#[0]	PO
B19	S1_AD[30]	PB	B20	S1_AD[31]	PB
C1	S2_AD[17]	PB	C2	VSS	-
C3	S2_STOP#	PSTS	C4	S2_SERR#	PI
C5	S2_AD[15]	PB	C6	S2_AD[12]	PB
C7	S2_AD[9]	PB	C8	S2_AD[6]	PB
C9	S2_AD[4]	PB	C10	S2_AD[0]	PB
C11	S1_GNT#[7]	PO	C12	S1_CLKOUT[6]	PTS
C13	S1_REQ#[4]	PIU	C14	VDD	PTS
C15	S1_REQ#[3]	PIU	C16	S1_CLKOUT[2]	PTS
C17	S1_REQ#[1]	PIU	C18	S1_AD[27]	PB
C19	S1_AD[28]	PB	C20	S1_AD[29]	PB
D1	S2_AD[18]	PB	D2	S2_FRAME#	PSTS
D3	S2_DEVSEL#	PSTS	D4	S2_PERR#	PSTS
D5	VDD	-	D6	S2_AD[13]	PB
D7	S1_M66EN	PI	D8	S2_AD[7]	PB
D9	S2_AD[5]	PB	D10	S2_AD[1]	PB
D11	VDD	-	D12	VSS	-
D13	S1_REQ#[5]	PIU	D14	S1_GNT#[3]	PO
D15	VDD	-	D16	S1_GNT#[1]	PO
D17	S1_AD[24]	PB	D18	VSS	-
D19	S1_AD[25]	PB	D20	S1_AD[26]	PB
E1	S2_AD[20]	PB	E2	VDD	-
E3	VSS	-	E4	S2_AD[19]	PB
E17	S1_AD[21]	PB	E18	S1_AD[22]	PB
E19	S1_AD[23]	PB	E20	S1_CBE[3]	PB
F1	S2_CBE[3]	PB	F2	S2_AD[23]	PB
F3	S2_AD[22]	PB	F4	S2_AD[21]	PB
F17	S1_AD[18]	PB	F18	VDD	-
F19	S1_AD[19]	PB	F20	S1_AD[20]	PB
G1	S2_AD[26]	PB	G2	VSS	-
G3	S2_AD[25]	PB	G4	S2_AD[24]	PB
G17	VSS	-	G18	S1_CBE[2]	PB
G19	S1_AD[16]	PB	G20	S1_AD[17]	PB
H1	S2_AD[30]	PB	H2	S2_AD[29]	PB
H3	S2_AD[28]	PB	H4	S2_AD[27]	PB
H17	VSS	-	H18	S1_TRDY#	PSTS
H19	S1_IRDY#	PSTS	H20	S1_FRAME#	PSTS
J1	S2_CLKOUT[0]	PTS	J2	VSS	-
J3	VDD	-	J4	S2_AD[31]	PB
J9	VSS	-	J10	VSS	-
J11	VSS	-	J12	VSS	-
J17	S1_PERR#	PSTS	J18	S1_LOCK#	PSTS
J19	S1_STOP#	PSTS	J20	S1_DEVSEL#	PSTS
K1	S2_REQ#[1]	PIU	K2	S2_GNT#[0]	PO

Pin #	Name	Type	Pin #	Name	Type
K3	S2_REQ#[0]	PIU	K4	VSS	-
K9	VSS	-	K10	VSS	-
K11	VSS	-	K12	VSS	-
K17	S1_CBE[1]	PB	K18	S1_PAR	PB
K19	VSS	-	K20	S1_SERR#	PI
L1	S2_GNT#[1]	PO	L2	S2_CLKOUT[1]	PTS
L3	S2_CLKOUT[2]	PTS	L4	S2_GNT#[2]	PO
L9	VSS	-	L10	VSS	-
L11	VSS	-	L12	VSS	-
L17	VDD	-	L18	S1_AD[13]	PB
L19	S1_AD[14]	PB	L20	S1_AD[15]	PB
M1	S2_REQ#[2]	PIU	M2	S2_REQ#[3]	PIU
M3	S2_GNT#[3]	PO	M4	S2_CLKOUT[3]	PTS
M9	VSS	-	M10	VSS	-
M11	VSS	-	M12	VSS	-
M17	S1_AD[10]	PB	M18	VSS	-
M19	S1_AD[11]	PB	M20	S1_AD[12]	PB
N1	VSS	-	N2	VDD	-
N3	S2_CLKOUT[4]	PTS	N4	S2_GNT#[4]	PO
N17	S1_AD[6]	PB	N18	S1_AD[7]	PB
N19	S1_AD[8]	PB	N20	S1_AD[9]	PB
P1	S2_REQ#[4]	PIU	P2	S2_REQ#[5]	PIU
P3	S2_CLKOUT[5]	PTS	P4	S2_GNT#[6]	PO
P17	S1_AD[5]	PB	P18	VSS	-
P19	VDD	-	P20	S1_CBE[0]	PB
R1	S2_GNT#[5]	PO	R2	VSS	-
R3	S2_REQ#[6]	PIU	R4	ENUM#	POD
R17	P_AD[0]	PB	R18	S1_AD[2]	PB
R19	S1_AD[3]	PB	R20	S1_AD[4]	PB
T1	S2_CLKOUT[6]	PTS	T2	HS_SW	PI
T3	S2_CLKOUT[7]	PTS	T4	S2_RESET#	PO
T17	P_AD[1]	PB	T18	VSS	-
T19	S1_AD[0]	PB	T20	S1_AD[1]	PB
U1	LOO	PO	U2	VSS	-
U3	TRST#	PIU	U4	AGND	-
U5	SCAN_EN	PID	U6	HS_EN	PI
U7	P_GNT#	PI	U8	P_AD[26]	PB
U9	VSS	-	U10	VDD	-
U11	P_AD[19]	PB	U12	P_CBE[2]	PB
U13	P_TRDY#	PB	U14	VSS	-
U15	P_PAR	PB	U16	P_CBE[1]	PB
U17	VSS	-	U18	P_AD[10]	PB
U19	P_AD[7]	PB	U20	P_AD[4]	PB
V1	VDD	-	V2	TCK	PIU
V3	TDO	PTS	V4	SCAN_TM#	PI
V5	S_CLKIN	PI	V6	P_CLK	PI
V7	VDD	-	V8	P_AD[27]	PB
V9	P_CBE[3]	PB	V10	P_AD[22]	PB
V11	P_AD[20]	PB	V12	P_AD[16]	PB
V13	P_IRDY#	PB	V14	P_LOCK#	PSTS
V15	VDD	-	V16	P_AD[15]	PB
V17	VSS	-	V18	P_M66EN#	PI
V19	P_CBE[0]	PB	V20	P_AD[3]	PB
W1	TMS	PIU	W2	TDI	PIU
W3	S1_EN	PIU	W4	S2_EN	PIU
W5	S2_M66EN	PI	W6	P_REQ#	PTS
W7	P_AD[30]	PB	W8	P_AD[28]	PB
W9	P_AD[24]	PB	W10	P_AD[23]	PB
W11	VSS	-	W12	P_AD[17]	PB
W13	P_FRAME#	PB	W14	P_STOP#	PSTS

Pin #	Name	Type	Pin #	Name	Type
W15	P_SERR#	POD	W16	P_AD[14]	PB
W17	P_AD[12]	PB	W18	P_AD[9]	PB
W19	P_AD[6]	PB	W20	VDD	-
Y1	AVCC	-	Y2	S_CFN#	PIU
Y3	PLL_TM	PI	Y4	BYPASS	-
Y5	P_RESET#	PI	Y6	VSS	-
Y7	P_AD[31]	PB	Y8	P_AD[29]	PB
Y9	P_AD[25]	PB	Y10	P_IDSEL	PI
Y11	P_AD[21]	PB	Y12	P_AD[18]	PB
Y13	VSS	-	Y14	P_DEVSEL#	PSTS
Y15	P_PERR#	PSTS	Y16	P_AD[13]	PB
Y17	P_AD[11]	PB	Y18	P_AD[8]	PB
Y19	P_AD[5]	PB	Y20	P_AD[2]	PB

## 4 PCI BUS OPERATION

This Chapter offers information about PCI transactions, transaction forwarding across PI7C7300D, and transaction termination. The PI7C7300D has three 128-byte buffers for buffering of upstream and downstream transactions. These hold addresses, data, commands, and byte enables and are used for both read and write transactions.

### 4.1 TYPES OF TRANSACTIONS

This section provides a summary of PCI transactions performed by PI7C7300D. Table 4-1 lists the command code and name of each PCI transaction. The Master and Target columns indicate support for each transaction when PI7C7300D initiates transactions as a master, on the primary (P) and secondary (S1, S2) buses, and when PI7C7300D responds to transactions as a target, on the primary (P) and secondary (S1, S2) buses.

**Table 4-1 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 4-1, the following PCI commands are not supported by PI7C7300D:

- PI7C7300D never initiates a PCI transaction with a reserved command code and, as a target, PI7C7300D ignores reserved command codes.
- PI7C7300D does not generate interrupt acknowledge transactions. PI7C7300D ignores interrupt acknowledge transactions as a target.
- PI7C7300D does not respond to special cycle transactions. PI7C7300D 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.
- PI7C7300D neither generates Type 0 configuration transactions on the primary PCI bus nor responds to Type 0 configuration transactions on the secondary PCI buses.

## 4.2 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]. PI7C7300D 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 nonzero, PI7C7300D automatically disconnects the transaction after the first data transfer.

## 4.3 DUAL ADDRESS PHASE

A 64-bit address uses two address phases. The first address phase is denoted by the asserting edge of FRAME#. The second address phase always follows on the next clock cycle.

For a 32-bit interface, the first address phase contains dual address command code on the C/BE#[3:0] lines, and the low 32 address bits on the AD[31:0] lines. The second address phase consists of the specific memory transaction command code on the C/BE#[3:0] lines, and the high 32 address bits on the AD[31:0] lines. In this way, 64-bit addressing can be supported on 32-bit PCI buses.

The *PCI-to-PCI Bridge Architecture Specification* supports the use of dual address transactions in the prefetchable memory range only. See Section 5.3.2 for a discussion of prefetchable address space. The PI7C7300D supports dual address transactions in both the upstream and the downstream direction. The PI7C7300D supports a programmable 64-bit address range in prefetchable memory for downstream forwarding of dual address transactions. Dual address transactions falling outside the prefetchable address range are forwarded upstream, but not downstream. Prefetching and posting are performed in a manner consistent with the guidelines given in this specification for each type of memory transaction in prefetchable memory space.



## 4.4 DEVICE SELECT (DEVSEL#) GENERATION

PI7C7300D always performs positive address decoding (medium decode) when accepting transactions on either the primary or secondary buses. PI7C7300D never does subtractive decode.

## 4.5 DATA PHASE

The address phase of a PCI transaction is followed by one or more data phases. A data phase is completed when IRDY# and either TRDY# or STOP# are asserted. A transfer of data occurs only when both IRDY# and TRDY# are asserted during the same PCI clock cycle. The last data phase of a transaction is indicated when FRAME# is de-asserted and both TRDY# and IRDY# are asserted, or when IRDY# and STOP# are asserted. See Section 4.9 for further discussion of transaction termination.

Depending on the command type, PI7C7300D can support multiple data phase PCI transactions. For detailed descriptions of how PI7C7300D imposes disconnect boundaries, see Section 4.6.4 for write address boundaries and Section 4.7.3 read address boundaries.

## 4.6 WRITE TRANSACTIONS

Write transactions are treated as either posted write or delayed write transactions. Table 4-2 shows the method of forwarding used for each type of write operation.

**Table 4-2 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

#### 4.6.1 MEMORY WRITE TRANSACTIONS

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

When PI7C7300D determines that a memory write transaction is to be forwarded across the bridge, PI7C7300D asserts DEVSEL# with medium timing and TRDY# 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, PI7C7300D accepts write data without obtaining access to the target bus. The PI7C7300D 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 PI7C7300D 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 PI7C7300D 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, PI7C7300D asserts its request on the target bus. This can occur while PI7C7300D 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, PI7C7300D asserts FRAME# and drives the stored write address out on the target bus. On the following cycle, PI7C7300D 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, PI7C7300D 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 PI7C7300D and the initiator stalls, PI7C7300D will signal the last data phase for the current transaction at the target bus if the queue empties. PI7C7300D will restart the follow-on transactions if the queue has new data.

PI7C7300D 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 (PI7C7300D starts another transaction to deliver the rest of the write data).
- The target returns a target abort (PI7C7300D discards remaining write data).
- The master latency timer expires, and PI7C7300D no longer has the target bus grant (PI7C7300D starts another transaction to deliver remaining write data).

Section 4.9.3.2 provides detailed information about how PI7C7300D responds to target termination during posted write transactions.

---

## 4.6.2 MEMORY WRITE AND INVALIDATE TRANSACTIONS

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

The PI7C7300D 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 the value in the cache line size register does not meet the memory write and invalidate conditions, the PI7C7300D returns a target disconnect to the initiator either on a cache line boundary or when the posted write buffer fills.

When the Memory Write and Invalidate transaction is disconnected before a cache line boundary is reached, typically because the posted write buffer fills, the transaction is converted to Memory Write transaction.

## 4.6.3 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 PI7C7300D forwards it as a delayed transaction, PI7C7300D claims the access by asserting DEVSEL# and returns a target retry to the initiator. During the address phase, PI7C7300D samples the bus command, address, and address parity one cycle later. After IRDY# is asserted, PI7C7300D 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 PI7C7300D initiates the transaction on the target bus. PI7C7300D transfers the write data to the target. If PI7C7300D 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 PI7C7300D is unable to deliver write data after  $2^{24}$  (default) or  $2^{32}$  (maximum) attempts, PI7C7300D will report a system error. PI7C7300D also asserts P\_SERR# if the primary SERR# enable bit is set in the command register. See Section 7.4 for information on the assertion of P\_SERR#. 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 PI7C7300D claims the access by asserting DEVSEL# and returns TRDY# to the initiator, to indicate that the write data

was transferred. If the initiator requests multiple DWORD, PI7C7300D also asserts STOP# in conjunction with TRDY# to signal a target disconnect. Note that only those