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

PI7C8154A

2-Port PCI-to-PCI Bridge

REVISION 1.02

3545 North 1st Street, San Jose, CA 95134
Telephone: 1-877-PERICOM, (1-877-737-4266)
Fax: 408-435-1100
Email: solutions@pericom.com
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
07/10/04	0.03	Initial release of preliminary specification
07/26/04	1.00	Initial release of specification to the web Updated Power Dissipation in section 17.9 Updated T_{DELAY} in sections 17.4 and 17.5 Revised V_{IH} parameter in section 17.2
11/24/09	1.01	Updated DC Specifications Updated the Ambient Temperature to be Industrial Temp Compliant
12/03/09	1.02	Added Appendix with Intel 21154 Pinout Comparison

This page intentionally left blank.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION

Product Description

The PI7C8154A is Pericom Semiconductor's PCI-to-PCI Bridge, designed to be fully compliant with the 64-bit, 66MHz implementation of the *PCI Local Bus Specification, Revision 2.2*. The PI7C8154A supports synchronous bus transactions between devices on the Primary Bus and the Secondary Buses operating up to 66MHz. The primary and secondary buses can also operate in concurrent mode, resulting in added increase in system performance.

Product Features

- 64-bit Primary and Secondary Ports run up to 66MHz
- 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
- Compliant with the *PCI Power Management Specification,* Revision 1.1
- Provides internal arbitration for nine secondary bus masters
  - Programmable 2-level priority arbiter
- Supports serial EEPROM interface for register auto-load and VPD access
- *Dynamic Prefetching Control*
- Supports posted write buffers in all directions
- 512 byte upstream posted memory write
- 512 byte downstream posted memory write
- 1024 byte upstream read data buffer
- 1024 byte downstream read data buffer
- Enhanced address decoding
- 32-bit I/O address range
- 32-bit memory-mapped I/O address range
- 64-bit prefetchable address range
- IEEE 1149.1 JTAG interface support
- Industrial temperature range -40°C to 85°C
- 3.3V and 5V signaling
- 304-pin PBGA package
  - Pb-free & Green available

1 SIGNAL DEFINITIONS

1.1 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

1.2 SIGNALS

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

1.2.1 PRIMARY BUS INTERFACE SIGNALS

Name	Pin #	Type	Description
P_AD[31:0]	U2, U4, U1, V2, V1, V3, W2, W1, W4, Y3, AA1, AA3, Y4, AB3, AA4, Y5, AB8, AA8, AC9, AB9, AA9, AC10, AA10, Y11, AB11, AA11, AA12, AB12, AB13, AA13, Y13, AA14	TS	**Primary Address / Data:** 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, bridge drives P_AD[31:0] to a valid logic level when P_GNT# is asserted.
P_CBE[3:0]	Y2, AB4, AA7, AC11	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, bridge drives P_CBE[3:0] to a valid logic level when P_GNT# is asserted.
P_PAR	AB7	TS	**Primary Parity.** P_PAR is even parity of P_AD[31:0] and P_CBE[3:0] (i.e. an even number of 1's). P_PAR 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 valid one clock after P_IRDY# is asserted. For read data phase, P_PAR 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, BRIDGE drives P_PAR to a valid logic level when P_GNT# is asserted.
P_FRAME#	AA5	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# 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#	AC5	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#	AB5	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#	AA6	STS	**Primary Device Select (Active LOW).** Asserted by the target indicating that the device is accepting the transaction. As a master, bridge 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#	AC6	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#	AB6	I	**Primary LOCK (Active LOW).** Asserted by an initiator, one clock cycle after the first address phase of a transaction, attempting to perform an operation that may take more than one PCI transaction to complete.
P_IDSEL	Y1	I	**Primary ID Select.** Used as a chip select line for Type 0 configuration access to bridge configuration space.
P_PERR#	AC7	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#	Y7	OD	**Primary System Error (Active LOW).** Can be driven LOW by any device to indicate a system error condition. Bridge drives this pin on: ▪ Address parity error ▪ Posted write data parity error on target bus ▪ Secondary S_SERR# 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#	U3	TS	**Primary Request (Active LOW):** This is asserted by BRIDGE to indicate that it wants to start a transaction on the primary bus. Bridge de-asserts this pin for at least 2 PCI clock cycles before asserting it again.
P_GNT#	R2	I	**Primary Grant (Active LOW):** When asserted, PI7C8154A can access the primary bus. During idle and P_GNT# asserted, bridge will drive P_AD, P_CBE, and P_PAR to valid logic levels.
P_RESET#	R3	I	**Primary RESET (Active LOW):** When P_RESET# is active, all PCI signals should be asynchronously tri-stated.

Name	Pin #	Type	Description
P_M66EN	AB10	I	**Primary Interface 66MHz Operation.** This input is used to specify if bridge 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.

1.2.2 PRIMARY BUS INTERFACE SIGNALS – 64-BIT EXTENSION

Name	Pin #	Type	Description
P_AD[63:32]	AA16, AB16, AA17, AB17, Y17, AB18, AC18, AA18, AC19, AA19, AB20, Y19, AA20, AB21, AC21, AA21, Y20, AA23, Y21, W20, Y23, W21, W23, W22, V21, V23, V22, U23, U20, U22, T23, T22	TS	**Primary Upper 32-bit Address / Data:** Multiplexed address and data bus providing an additional 32 bits to the primary. When a dual address command is used and P_REQ64# is asserted, the initiator drives the upper 32 bits of the 64-bit address. Otherwise, these bits are undefined and driven to valid logic levels. During the data phase of a transaction, the initiator drives the upper 32 bits of the 64-bit write data, or the target drives the upper 32 bits of the 64-bit read data, when P_REQ64# and P_ACK64# are both asserted. Otherwise, these bits are pulled up to a valid logic level through external resistors.
P_CBE[7:4]	AA15, AB15, Y15, AC15	TS	**Primary Upper 32-bit Command/Byte Enables:** Multiplexed command field and byte enable field. During address phase, when the dual address command is used and P_REQ64# is asserted, the initiator drives the transaction type on these pins. Otherwise, these bits are undefined, and the initiator drives a valid logic level onto the pins. For read and write transactions, the initiator drives these bits for the P_AD[63:32] data bits when P_REQ64# and P_ACK64# are both asserted. When not driven, these bits are pulled up to a valid logic level through external resistors.
P_PAR64	T21	TS	**Primary Upper 32-bit Parity:** P_PAR64 carries the even parity of P_AD[63:32] and P_CBE[7:4] for both address and data phases. P_PAR64 is driven by the initiator and is valid 1 cycle after the first address phase when a dual address command is used and P_REQ64# is asserted. P_PAR64 is valid 1 clock cycle after the second address phase of a dual address transaction when P_REQ64# is asserted. P_PAR64 is valid 1 cycle after valid data is driven when both P_REQ64# and P_ACK64# are asserted for that data phase. P_PAR64 is driven by the device driving read or write data 1 cycle after the P_AD lines are driven. P_PAR64 is tri-stated 1 cycle after the P_AD lines are tri-stated. Devices receive data sample P_PAR64 as an input to check for possible parity errors during 64-bit transactions. When not driven, P_PAR64 is pulled up to a valid logic level through external resistors.

Name	Pin #	Type	Description
P_REQ64#	AC14	STS	**Primary 64-bit Transfer Request:** P_REQ64# is asserted by the initiator to indicate that the initiator is requesting a 64-bit data transfer. P_REQ64# has the same timing as P_FRAME#. When P_REQ64# is asserted LOW during reset, a 64-bit data path is supported. When P_REQ64# is HIGH during reset, bridge drives P_AD[63:32], P_CBE[7:4], and P_PAR64 to valid logic levels. When deasserting, P_REQ64# is driven to a deasserted state for 1 cycle and then sustained by an external pull-up resistor.
P_ACK64#	AB14	STS	**Primary 64-bit Transfer Acknowledge:** P_ACK64# is asserted by the target only when P_REQ64# is asserted by the initiator to indicate the target's ability to transfer data using 64 bits. P_ACK64# has the same timing as P_DEVSEL#. When deasserting, P_ACK64# is driven to a deasserted state for 1 cycle and then is sustained by an external pull-up resistor.

1.2.3 SECONDARY BUS INTERFACE SIGNALS

Name	Pin #	Type	Description
S_AD[31:0]	C3, A3, B3, C4, A4, B4, C5, B5, A6, A7, D7, B7, A8, B8, C8, A9, C13, B13, A13, D13, C14, B14, C15, B15, C16, B16, C17, B17, D17, A17, B18, A18	TS	**Secondary Address/Data:** Multiplexed address and data bus. Address is indicated by S_FRAME# assertion. Write data is stable and valid when S_IRDY# is asserted and read data is stable and valid when S_IRDY# is asserted. Data is transferred on rising clock edges when both S_IRDY# and S_TRDY# are asserted. During bus idle, bridge drives S_AD[31:0] to a valid logic level when S_GNT# is asserted respectively.
S_CBE[3:0]	C6, D9, C12, A15	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, bridge drives S_CBE[3:0] to a valid logic level when the internal grant is asserted.
S_PAR	B12	TS	**Secondary Parity:** S_PAR is an even parity of S_AD[31:0] and S_CBE[3:0] (i.e. an even number of 1's). S_PAR is valid and stable one cycle after the address phase (indicated by assertion of S_FRAME#) for address parity. For write data phases, S_PAR is valid one clock after S_IRDY# is asserted. For read data phase, S_PAR is valid one clock after S_TRDY# is asserted. Signal S_PAR is tri-stated one cycle after the S_AD lines are tri-stated. During bus idle, bridge drives S_PAR to a valid logic level when the internal grant is asserted.
S_FRAME#	B9	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# 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#	C9	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.

Name	Pin #	Type	Description
S_TRDY#	A10	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.
S_DEVSEL#	B10	STS	**Secondary Device Select (Active LOW):** Asserted by the target indicating that the device is accepting the transaction. As a master, bridge waits for the assertion of this signal within 5 cycles of S_FRAME# assertion; otherwise, terminate with master abort. Before tri-stated, it is driven to a de-asserted state for one cycle.
S_STOP#	C10	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#	A11	STS	**Secondary LOCK (Active LOW):** Asserted by an initiator, one clock cycle after the first address phase of a transaction, when it is propagating a locked transaction downstream. Bridge does not propagate locked transactions upstream.
S_PERR#	C11	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#	B11	I	**Secondary System Error (Active LOW):** Can be driven LOW by any device to indicate a system error condition.
S_REQ#[8:0]	E1, E3, D2, D1, E4, D3, C2, C1, D4	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#[8:0]	H1, G3, G2, G4, G1, F2, F1, F3, E2	TS	**Secondary Grant (Active LOW):** PI7C8154A asserts these pins to allow external masters to access the secondary bus. Bridge de-asserts these pins for at least 2 PCI clock cycles before asserting it again. During idle and S_GNT# deasserted, PI7C8154A will drive S_AD, S_CBE, and S_PAR.
S_RESET#	H2	O	**Secondary RESET (Active LOW):** Asserted when any of the following conditions are met: 1. Signal P_RESET# is asserted. 2. Secondary reset bit in bridge control register in configuration space is set. 3. The chip reset bit in the chip control register in configuration space is set. When asserted, all control signals are tri-stated and zeroes are driven on S_AD, S_CBE, S_PAR, and S_PAR64.
S_M66EN	A14	I/OD	**Secondary Interface 66MHz Operation:** This input is used to specify if bridge is capable of 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 driven LOW.
S_CFN#	K1	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#[0] is reconfigured to be the secondary bus grant input, and S_GNT#[0] is reconfigured to be the secondary bus request output.

1.2.4 SECONDARY BUS INTERFACE SIGNALS – 64-EXTENSTION

Name	Pin #	Type	Description
S_AD[63:32]	C20, A21, D20, C21, C23, C22, D21, E20, D22, E21, E23, F21, F23, F22, G20, G22, G21, H23, H22, H21, J23, J20, J22, K23, K22, K21, L23, L21, L22, M22, M23, M21	TS	**Secondary Upper 32-bit Address/Data:** Multiplexed address and data bus. Address is indicated by S_FRAME# assertion. Write data is stable and valid when S_IRDY# is asserted and read data is stable and valid when S_IRDY# is asserted. Data is transferred on rising clock edges when both S_IRDY# and S_TRDY# are asserted. During bus idle, bridge drives S_AD to a valid logic level when S_GNT# is asserted respectively.
S_CBE[7:4]	A19, C19, A20, D19	TS	**Secondary Upper 32-bit 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, bridge drives S_CBE[7:0] to a valid logic level when the internal grant is asserted.
S_PAR64	N21	TS	**Secondary Upper 32-bit Parity:** S_PAR64 carries the even parity of S_AD[63:32] and S_CBE[7:4] for both address and data phases. S_PAR64 is driven by the initiator and is valid 1 cycle after the first address phase when a dual address command is used and S_REQ64# is asserted. S_PAR64 is valid 1 clock cycle after the second address phase of a dual address transaction when S_REQ64# is asserted. S_PAR64 is valid 1 cycle after valid data is driven when both S_REQ64# and S_ACK64# are asserted for that data phase. S_PAR64 is driven by the device driving read or write data 1 cycle after the S_AD lines are driven. S_PAR64 is tri-stated 1 cycle after the S_AD lines are tri-stated. Devices receive data sample S_PAR64 as an input to check for possible parity errors during 64-bit transactions. When not driven, S_PAR64 is pulled up to a valid logic level through external resistors.
S_REQ64#	B19	STS	**Secondary 64-bit Transfer Request:** S_REQ64# is asserted by the initiator to indicate that the initiator is requesting a 64-bit data transfer. S_REQ64# has the same timing as S_FRAME#. When S_REQ64# is asserted LOW during reset, a 64-bit data path is supported. When S_REQ64# is HIGH during reset, bridge drives S_AD[63:32], S_CBE[7:4], and S_PAR64 to valid logic levels. When deasserting, S_REQ64# is driven to a deasserted state for 1 cycle and then sustained by an external pull-up resistor.
S_ACK64#	C18	STS	**Secondary 64-bit Transfer Acknowledge:** S_ACK64# is asserted by the target only when S_REQ64# is asserted by the initiator to indicate the target's ability to transfer data using 64 bits. S_ACK64# has the same timing as S_DEVSEL#. When deasserting, S_ACK64# is driven to a deasserted state for 1 cycle and then is sustained by an external pull-up resistor.

1.2.5 CLOCK SIGNALS

Name	Pin #	Type	Description
P_CLK	T3	I	**Primary Clock Input:** Provides timing for all transactions on the primary interface.

Name	Pin #	Type	Description
S_CLKIN	J4	I	**Secondary Clock Input:** Provides timing for all transactions on the secondary interface.
S_CLKOUT[9:0]	P1, P2, P3, N1, N3, M2, M1, M3, L3, L2	O	**Secondary Clock Output:** Provides secondary clocks phase synchronous with the P_CLK. 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.

1.2.6 MISCELLANEOUS SIGNALS

Name	Pin #	Type	Description
MSK_IN	R21	I	**Secondary Clock Disable Serial Input:** This pin is used by bridge to disable secondary clock outputs. The serial stream is received by MSK_IN, starting when P_RESET is detected deasserted and S_RESET# 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	R20	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	N22	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	R4	I	**Bus/Power Clock Control Management Pin:** When this pin is tied HIGH and the bridge is placed in the D2 or $D3_{HOT}$ power state, it enables the bridge 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 bridge enters the D2 or $D3_{HOT}$ power state.
CONFIG66	R22	I	**66MHz Configuration:** This pin indicates if the bridge is capable of running at 66MHz operation. Tie HIGH to set bit [21] of offset 04h of the status register.
PMEENA#	D11	I	**Power Management Enable Support:** This pin sets bits [31:27] offset DEh of the Power Management Capabilities Register. When tied LOW, bits [31:27] offset DEh are set to 11111 to indicate that the secondary devices are capable of asserting PME#. When this pin is tied HIGH, bits [31:27] offset DEh are set to 00000 to indicate that PI7C8154A does not support the PME# pin. For Intel 21154, this pin is defined as VDD. For more info, please refer to Appendix on page 113.

EEDATA	A22	I/O	**EEPROM Data:** Serial data interface to the EEPROM. For Intel 21154, this pin is defined as VDD. For more info, please refer to Appendix on page 113.
EECLK	A23	O	**EEPROM Clock:** Clock signal to the EEPROM interface used during the autoload and VPD functions. For Intel 21154, this pin is defined as VSS. For more info, please refer to Appendix on page 113.
EE_EN#	AC22	I	**EEPROM Enable:** Set to LOW to enable EEPROM interface. For Intel 21154, this pin is defined as VDD. For more info, please refer to Appendix on page 113.

1.2.7 GENERAL PURPOSE I/O INTERFACE SIGNALS

Name	Pin #	Type	Description
GPIO[3:0]	K2, K3, L4, L1	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.

1.2.8 JTAG BOUNDARY SCAN SIGNALS

Name	Pin #	Type	Description
TCK	N20	I	**Test Clock.** Used to clock state information and data into and out of the bridge during boundary scan.
TMS	P21	I	**Test Mode Select.** Used to control the state of the Test Access Port controller.
TDO	P22	O	**Test Data Output.** Used as the serial output for the test instructions and data from the test logic.
TDI	P23	I	**Test Data Input.** Serial input for the JTAG instructions and test data.
TRST#	N23	I	**Test Reset.** Active LOW signal to reset the Test Access Port (TAP) controller into an initialized state.

1.2.9 POWER AND GROUND

Name	Pin #	Type	Description
VDD	A2, B1, B6, B20, B23, D5, D6, D10, D14, D15, D18, E22, H4, H20, J1, J3, J21, M4, M20, N4, R1, R23, T1, T4, T20, W3, Y6, Y10, Y14, Y18, Y22, AB1, AB19, AB23, AC2, AC3, AC8, AC12, AC16	P	**Power:** +3.3V Digital power.
VSS	A1, A5, A12, A16, B2, B21, B22, C7, D8, D12, D16, D23, F4, F20, G23, H3, J2, K4, K20, L20, N2, P4, P20, T2, U21, V4, V20, Y8, Y9, Y12, Y16, AA2, AA22, AB2, AB22, AC1, AC4, AC13, AC17, AC20, AC23	P	**Ground:** Digital ground.

1.3 PIN LIST

BALL LOCATION	PIN NAME	TYPE	BALL LOCATION	PIN NAME	TYPE
A1	VSS	P	A2	VDD	P
A3	S_AD[30]	TS	A4	S_AD[27]	TS
A5	VSS	P	A6	S_AD[23]	TS
A7	S_AD[22]	TS	A8	S_AD[19]	TS
A9	S_AD[16]	TS	A10	S_TRDY#	STS
A11	S_LOCK#	STS	A12	VSS	P
A13	S_AD13]	TS	A14	SM66EN	I/OD
A15	S_CBE[0]	TS	A16	VSS	P
A17	S_AD[2]	TS	A18	S_AD[0]	TS
A19	S_CBE[7]	TS	A20	S_CBE[5]	TS
A21	S_AD[62]	TS	A22	EEDATA	I/O
A23	EECLK	O	-	-	-
B1	VDD	P	B2	VSS	P
B3	S_AD[29]	TS	B4	S_AD[26]	TS
B5	S_AD[24]	TS	B6	VDD	P
B7	S_AD[20]	TS	B8	S_AD[18]	TS
B9	S_FRAME#	STS	B10	S_DEVSEL#	STS
B11	S_SERR#	I	B12	S_PAR	TS
B13	S_AD[14]	TS	B14	S_AD[10]	TS
B15	S_AD[8]	TS	B16	S_AD[6]	TS
B17	S_AD[4]	TS	B18	S_AD[1]	TS
B19	S_REQ64#	STS	B20	VDD	P
B21	VSS	P	B22	VSS	P
B23	VDD	P	-	-	-
C1	S_REQ#[1]	I	C2	S_REQ#[2]	I
C3	S_AD[31]	TS	C4	S_AD[28]	TS
C5	S_AD[25]	TS	C6	S_CBE[3]	TS
C7	VSS	P	C8	S_AD[17]	TS
C9	S_IRDY#	STS	C10	S_STOP#	STS
C11	S_PERR#	STS	C12	S_CBE[1]	TS
C13	S_AD[15]	TS	C14	S_AD[11]	TS
C15	S_AD[9]	TS	C16	S_AD[7]	TS
C17	S_AD[5]	TS	C18	S_ACK64#	STS
C19	S_CBE[6]	TS	C20	S_AD[63]	TS
C21	S_AD[60]	TS	C22	S_AD[58]	TS
C23	S_AD[59]	TS	-	-	-
D1	S_REQ#[5]	I	D2	S_REQ#[6]	I
D3	S_REQ_[3]	I	D4	S_REQ#[0]	I
D5	VDD	P	D6	VDD	P
D7	S_AD[21]	TS	D8	VSS	P
D9	S_CBE[2]	TS	D10	VDD	P
D11	PMEENA#	I	D12	VSS	P
D13	S_AD[12]	TS	D14	VDD	P
D15	VDD	P	D16	VSS	P
D17	S_AD[3]	TS	D18	VDD	P
D19	S_CBE[4]	TS	D20	S_AD[61]	TS
D21	S_AD[57]	TS	D22	S_AD[55]	TS
D23	VSS	P	-	-	-
E1	S_REQ#[8]	I	E2	S_GNT#[0]	TS
E3	S_REQ#[7]	I	E4	S_REQ#[4]	I
-	-	-	E20	S_AD[56]	TS
E21	S_AD[54]	TS	E22	VDD	P
E23	S_AD[53]	TS	-	-	-
F1	S_GNT#[2]	TS	F2	S_GNT#[3]	TS
F3	S_GNT#[1]	TS	F4	VSS	P
-	-	-	F20	VSS	P

BALL LOCATION	PIN NAME	TYPE	BALL LOCATION	PIN NAME	TYPE
F21	S_AD[52]	TS	F22	S_AD[50]	TS
F23	S_AD[51]	TS	-	-	-
G1	S_GNT#[4]	TS	G2	S_GNT#[6]	TS
G3	S_GNT#[7]	TS	G4	S_GNT#[5]	TS
-	-	-	G20	S_AD[49]	TS
G21	S_AD[47]	TS	G22	S_AD[48]	TS
G23	VSS	P	-	-	-
H1	S_GNT#[8]	TS	H2	S_RESET#	O
H3	VSS	P	H4	VDD	P
-	-	-	H20	VDD	P
H21	S_AD[44]	TS	H22	S_AD[45]	TS
H23	S_AD[46]	TS	-	-	-
J1	VDD	P	J2	VSS	P
J3	VDD	P	J4	S_CLKIN	I
-	-	-	J20	S_AD[42]	TS
J21	VDD	P	J22	S_AD[41]	TS
J23	S_AD[43]	TS	-	-	-
K1	S_CFN#	I	K2	GPIO[3]	TS
K3	GPIO[2]	TS	K4	VSS	P
-	-	-	K20	VSS	P
K21	S_AD[38]	TS	K22	S_AD[39]	TS
K23	S_AD[40]	TS	-	-	-
L1	GPIO[0]	TS	L2	S_CLKOUT[0]	O
L3	S_CLKOUT[1]	O	L4	GPIO[1]	TS
-	-	-	L20	VSS	P
L21	S_AD[36]	TS	L22	S_AD[35]	TS
L23	S_AD[37]	TS	-	-	-
M1	S_CLKOUT[3]	O	M2	S_CLKOUT[4]	O
M3	S_CLKOUT[2]	O	M4	VDD	P
-	-	-	M20	VDD	P
M21	S_AD[32]	TS	M22	S_AD[34]	TS
M23	S_AD[33]	TS	-	-	-
N1	S_CLKOUT[6]	O	N2	VSS	P
N3	S_CLKOUT[5]	O	N4	VDD	P
-	-	-	N20	TCK	I
N21	S_PAR64	TS	N22	S_VIO	I
N23	TRST#	I	-	-	-
P1	S_CLKOUT[9]	O	P2	S_CLKOUT[8]	O
P3	S_CLKOUT[7]	O	P4	VSS	P
-	-	-	P20	VSS	P
P21	TMS	I	P22	TDO	O
P23	TDI	I	-	-	-
R1	VDD	P	R2	P_GNT#	I
R3	P_RESET#	I	R4	BPCCE	I
-	-	-	R20	P_VIO	I
R21	MSK_IN	I	R22	CONFIG66	I
R23	VDD	P	-	-	-
T1	VDD	P	T2	VSS	P
T3	P_CLK	I	T4	VDD	P
-	-	-	T20	VDD	P
T21	P_PAR64	TS	T22	P_AD[32]	TS
T23	P_AD[33]	TS	-	-	-
U1	P_AD[29]	TS	U2	P_AD[31]	TS
U3	P_REQ#	TS	U4	P_AD[30]	TS
-	-	-	U20	P_AD[35]	TS
U21	VSS	P	U22	P_AD[34]	TS
U23	P_AD[36]	TS	-	-	-
V1	P_AD[27]	TS	V2	P_AD[28]	TS

BALL LOCATION	PIN NAME	TYPE	BALL LOCATION	PIN NAME	TYPE
V3	P_AD[26]	TS	V4	VSS	P
-	-	-	V20	Reserved[1]	P
V21	P_AD[39]	TS	V22	P_AD[37]	TS
V23	P_AD[38]	TS	-	-	-
W1	P_AD[24]	TS	W2	P_AD[25]	TS
W3	VDD	P	W4	P_AD[23]	TS
-	-	-	W20	P_AD[44]	TS
W21	P_AD[42]	TS	W22	P_AD[40]	TS
W23	P_AD[41]	TS	-	-	-
Y1	P_IDSEL	I	Y2	P_CBE[3]	TS
Y3	P_AD[22]	TS	Y4	P_AD[19]	TS
Y5	P_AD[16]	TS	Y6	VDD	P
Y7	P_SERR#	OD	Y8	VSS	P
Y9	VSS	P	Y10	VDD	P
Y11	P_AD[8]	TS	Y12	VSS	P
Y13	P_AD[1]	TS	Y14	VDD	P
Y15	P_CBE[5]	TS	Y16	VSS	P
Y17	P_AD[59]	TS	Y18	Reserved[2]	P
Y19	P_AD[52]	TS	Y20	P_AD[47]	TS
Y21	P_AD[45]	TS	Y22	VDD	P
Y23	P_AD[43]	TS	-	-	-
AA1	P_AD[21]	TS	AA2	VSS	P
AA3	P_AD[20]	TS	AA4	P_AD[17]	TS
AA5	P_FRAME#	STS	AA6	P_DEVSEL#	STS
AA7	P_CBE[1]	TS	AA8	P_AD[14]	TS
AA9	P_AD[11]	TS	AA10	P_AD[9]	TS
AA11	P_AD[6]	TS	AA12	P_AD[5]	TS
AA13	P_AD[2]	TS	AA14	P_AD[0]	TS
AA15	P_CBE[7]	TS	AA16	P_AD[63]	TS
AA17	P_AD[61]	TS	AA18	P_AD[56]	TS
AA19	P_AD[54]	TS	AA20	P_AD[51]	TS
AA21	P_AD[48]	TS	AA22	VSS	P
AA23	P_AD[46]	TS	-	-	-
AB1	VDD	P	AB2	VSS	P
AB3	P_AD[18]	TS	AB4	P_CBE[2]	TS
AB5	P_TRDY#	STS	AB6	P_LOCK#	I
AB7	P_PAR	TS	AB8	P_AD[15]	TS
AB9	P_AD[12]	TS	AB10	P_M66EN	I
AB11	P_AD[7]	TS	AB12	P_AD[4]	TS
AB13	P_AD[3]	TS	AB14	P_ACK64#	STS
AB15	P_CBE[6]	TS	AB16	P_AD[62]	TS
AB17	P_AD[60]	TS	AB18	P_AD[58]	TS
AB19	VDD	P	AB20	P_AD[53]	TS
AB21	P_AD[50]	TS	AB22	VSS	P
AB23	VDD	P	-	-	-
AC1	VSS	P	AC2	VDD	P
AC3	VDD	P	AC4	VSS	P
AC5	P_IRDY#	STS	AC6	P_STOP#	STS
AC7	P_PERR#	STS	AC8	VDD	P
AC9	P_AD[13]	TS	AC10	P_AD[10]	TS
AC11	P_CBE[0]	TS	AC12	VDD	P
AC13	VSS	P	AC14	P_REQ64#	STS
AC15	P_CBE[4]	TS	AC16	VDD	P
AC17	VSS	P	AC18	P_AD[57]	TS
AC19	P_AD[55]	TS	AC20	VSS	P
AC21	P_AD[49]	TS	AC22	EE_EN#	I
AC23	VSS	P	-	-	-

[1] Connected to GROUND
[2] Connected to V_{DD}

2 SIGNAL DEFINITIONS

This Chapter offers information about PCI transactions, transaction forwarding across PI7C8154A, and transaction termination. The PI7C8154A has two 128-byte buffers for read data buffering of upstream and downstream transactions. Also, PI7C8154A has two 128-byte buffers for write data buffering of upstream and downstream transactions.

2.1 TYPES OF TRANSACTIONS

This section provides a summary of PCI transactions performed by PI7C8154A. Table 2-1 lists the command code and name of each PCI transaction. The Master and Target columns indicate support for each transaction when PI7C8154A initiates transactions as a master, on the primary and secondary buses, and when PI7C8154A responds to transactions as a target, on the primary and secondary buses.

Table 2-1 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 2-1, the following PCI commands are not supported by PI7C8154A:

- PI7C8154A never initiates a PCI transaction with a reserved command code and, as a target, PI7C8154A ignores reserved command codes.
- PI7C8154A does not generate interrupt acknowledge transactions. PI7C8154A ignores interrupt acknowledge transactions as a target.
- PI7C8154A does not respond to special cycle transactions. PI7C8154A 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.
- PI7C8154A neither generates Type 0 configuration transactions on the primary PCI bus nor responds to Type 0 configuration transactions on the secondary PCI bus.

2.2 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]. PI7C8154A 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, PI7C8154A automatically disconnects the transaction after the first data transfer.

2.3 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 CBE[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 CBE[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 3.3.3 for a discussion of prefetchable address space. The PI7C8154A supports dual address transactions in both the upstream and the downstream direction. The PI7C8154A 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 document for each type of memory transaction in prefetchable memory space.

2.4 DEVICE SELECT (DEVSEL#) GENERATION

PI7C8154A always performs positive address decoding (medium decode) when accepting transactions on either the primary or secondary buses. PI7C8154A never does subtractive decode.

2.5 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 2.11 for further discussion of transaction termination.

Depending on the command type, PI7C8154A can support multiple data phase PCI transactions. For detailed descriptions of how PI7C8154A imposes disconnect boundaries, see Section 2.6.4 for write address boundaries and Section 2.7.3 read address boundaries.

2.6 WRITE TRANSACTIONS

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

Table 2-2 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

2.6.1 MEMORY WRITE TRANSACTIONS

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

When PI7C8154A determines that a memory write transaction is to be forwarded across the bridge, PI7C8154A asserts DEVSEL# with medium decode 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, PI7C8154A accepts write data without obtaining access to the target bus. The PI7C8154A 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 PI7C8154A 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 PI7C8154A 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, PI7C8154A asserts its request on the target bus. This can occur while PI7C8154A 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, PI7C8154A asserts FRAME# and drives the stored write address out on the target bus. On the following cycle, PI7C8154A 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, PI7C8154A can drive one DWORD of write data in each PCI clock cycle; that is, no master wait states are inserted. If write data is flowing through PI7C8154A and the initiator stalls, PI7C8154A will signal the last data phase for the current transaction at the target bus if the queue empties. PI7C8154A will restart the follow-on transactions if the queue has new data.

PI7C8154A 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 (PI7C8154A starts another transaction to deliver the rest of the write data).