

Peripheral Component Interconnect (PCI) Host Controller Data Sheet



••••• Proven System Block (PSB) for QuickLogic Customer Specific Standard Products (CSSPs)

Features

The QuickLogic PCI Host Controller has the following features:

- 32-bit/33 MHz or 32-bit/66 MHz PCI Host/Master/Target
- PCI v2.3 compliant
- Supports up to two PCI master agents
- Type 0 and Type 1 configuration cycle generation from the host processor
- 3.3 V PCI signaling
- Supports PCI or miniPCI bus form factors
- Extremely high throughput while consuming very little power (as low as 35 mW)

Overview

PCI was widely used by personal computers as the interface between the microprocessor and the peripherals. It has been recently replaced by the serial PCI Express interface for PC motherboards. In the portable world, SDIO and USB are gaining more popularity and become more prevalent compared to PCI. However, PCI is still used in some portable applications, such as WiFi and networking, due to costs, performance and peripheral availability. Industrial and automotive applications may also choose PCI for backward compatibility reasons.

QuickLogic's PCI Host Controller PSB is PCI version 2.3 compliant and supports 32-bit/33 MHz or 32-bit/66 MHz PCI Host/Master/Target. It seamlessly adds PCI interface to application processors without native PCI support because of the high pin count requirement. It enables high data throughput at 33 MHz or 66 MHz with a 32-bit databus. The built-in DMA alleviates CPU overhead to lower CPU utilization and power consumption.

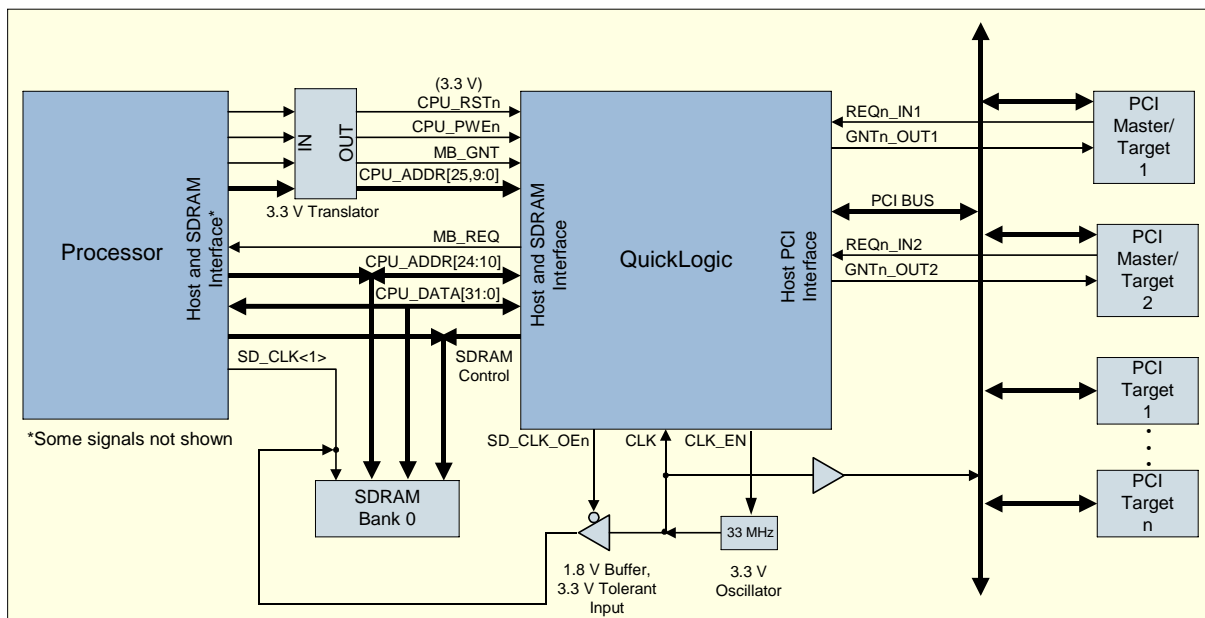
PCI Host Controller Architecture

The host processor communicates to the PCI Host Controller via its system memory interface. The QuickLogic CSSP are based on its proprietary ViaLink[®] technology and therefore can be customized to seamlessly interface with any types of host interfaces. The CSSP performs internal decoding of the local address and directs the memory data transaction towards the status/control registers or PCI interface. By using the CSSP, designers are provided with the ability to seamlessly add PCI connectivity not natively supported.

The PCI Host Controller enables the CSSP to initiate configuration and memory read/write cycles to devices on the PCI bus. These accesses are used for control purposes such as device enumeration, status updates, or DMA setup. Additionally, the CSSP acts an alternate bus master to host memory. Two PCI agents can read DMA descriptors and read/write DMA data directly to/from host memory through the embedded PCI core in the controller. A hardware handshake protocol on the CSSP is used to gain control of the host memory bus when data transfers are requested from PCI.

Figure 1 shows a system level block diagram of the PCI Host Controller.

Figure 1: PCI Host Controller System Level Block Diagram



Interface List and Description

Table 1 summarizes the PCI Host Controller interface signals.

Table 1: Pin Descriptions

Pin	Type ^a	I/O Rail Voltage	Description
PCI Interface			
CLK	I	3.3 V	PCI clock input. Also serves as SDRAM clock input when the PCI Host Controller has control of processor memory bus.
CLK_EN	O	VCCIO	PCI clock oscillator enable output.
PCI_RSTn	O	VCCIO	PCI reset output. Active low.
AD[31:0]	I/O	VCCIO	Multiplexed PCI address/data.
CBE _n [3:0]	I/O	VCCIO	Multiplexed PCI command/byte enables. Active low.
FRAME _n	S/T/S	VCCIO	Indication of beginning of PCI data transfer. Active low.
IRDY _n	S/T/S	VCCIO	PCI initiator ready. Active low.
DEVSEL _n	S/T/S	VCCIO	PCI device select. Active low.
TRDY _n	S/T/S	VCCIO	PCI target ready. Active low.
STOP _n	S/T/S	VCCIO	Active low PCI target-initiated termination. Active low.
PAR	I/O	VCCIO	Even parity result.
PERR _n	S/T/S	VCCIO	PCI parity error. Active low.
SERR _n	O/D	VCCIO	PCI system error. Active low.
CLKRUN _n	S/T/S	VCCIO	PCI control signal. Used by QuickLogic CSSP to request permission to stop or slow the PCI clock. Active low.
PME _n	I	VCCIO	PCI power bus management signal. Active low.
PCI Arbitration and Interrupts			
PCI_INT1 _n	I	VCCIO	PCI interrupt #1. Active low.
PCI_INT2 _n	I	VCCIO	PCI interrupt #2. Active low.
REQ _n _IN1	I	VCCIO	PCI bus request signal from PCI device #1. Active low.
REQ _n _IN2	I	VCCIO	PCI bus request signal from PCI device #2. Active low.
GNT _n _OUT1	O	VCCIO	PCI bus grant signal to PCI device #1. Active low.
GNT _n _OUT2	O	VCCIO	PCI bus grant signal to PCI device #2. Active low.
Host Interface			
CPU_RSTn	I	3.3 V	System reset input. Active low.
CPU_ADDR[25]	I/O	VCCIO	CPU address bus. 32-bit data accesses are used so lower two address bits are ignored.
CPU_ADDR[24:10]	I/O	VCCIO<A>	CPU address bus. 32-bit data accesses are used so lower two address bits are ignored.

Table 1: Pin Descriptions (Continued)

Pin	Type ^a	I/O Rail Voltage	Description
CPU_ADDR[9:5]	I/O	VCCIO	CPU address bus. 32-bit data accesses are used so lower two address bits are ignored.
CPU_ADDR[4:2]	I/O	3.3 V	CPU address bus. 32-bit data accesses are used so lower two address bits are ignored.
CPU_DATA[31:0]	I/O	VCCIO<A>	CPU data bus.
CPU_CS _n	I	VCCIO<A>	Chip select. Active low.
CPU_OE _n	I/O	VCCIO<A>	Output enable. Active low.
CPU_PWE _n	I	VCCIO	Write enable. Active low
CPU_RDY	O	VCCIO<A>	Ready signal.
CPU_RDWR _n	I/O	VCCIO<A>	Read/not write indicator.
CPU_INT _n	O	VCCIO<A>	Interrupt. Active low.
SD_CS _n	O	VCCIO<A>	SDRAM chip select. Active low.
SD_RAS _n	O	VCCIO<A>	SDRAM row address strobe. Active low
SD_CAS _n	O	VCCIO<A>	SDRAM column address strobe. Active low
SD_WE _n	O	VCCIO<A>	SDRAM write enable. Active low
SD_DQM[3:0]	I/O	VCCIO<A>	SDRAM/host data mask control.
SD_CLK_OE _n	O	VCCIO<A>	Clock enable. Active low. Output to external tri-state buffer driving SDRAM clock input. Enabled when PCI Host Controller has control of processor memory bus.
MB_REQ	O	VCCIO<A>	SDRAM memory alternate bus master request.
MB_GNT	I	VCCIO	SDRAM memory alternate bus master grant.

a. I = Input

O = Output

I/O = Bidirectional

S/T/S = Sustained Tri-state Signal

Register Sets

Memory Map

Table 2 shows an example of a QuickLogic CSSP memory map for the Marvell PXA2xx processor using the offset addresses. Within external memory region 5 of the Marvell PXA2xx processor, host memory is divided according to specific functions.

Table 2: Example CSSP Memory Map for a Marvell PXA2xx Processor

Offset Address	Memory Description
0x0000_0000 - 0x0000_001F	Common register space
0x0300_0000 - 0x300_00FF	PCI register space
0x0300_0400 - 0x0300_0403	PCI configuration space data port
0x0380_0000 - 0x03FF_FFFF	PCI memory space

The control/status registers are used to manage the behavior of the CSSP and report the status of various flags. There are two sets of registers: Common Register Set and PCI Register Set. Each of these register sets are memory-mapped to the processor, and are only accessible by the processor.

Writing to reserved registers has no effect, and reading from reserved registers yields undefined values.

All registers are accessed using 32-bit read or write accesses.

Table 3: Control/Status Register Memory Map

Offset Address	Register Name	Register Description
0x0000_0000	COM_ID	Identification register
0x0000_0004	COM_REV	Revision register
0x0000_0008	COM_FEATURES	Companion device features register
0x0000_0014	COM_INTR_EN	Master interrupt enable register
0x0000_0018	COM_INTR_STAT	Master interrupt status register
0x0300_0000	PCI_CFG	PCI configuration register
0x0300_0004	PCI_INTR_EN	PCI interrupt enable register
0x0300_0008	PCI_INTR_STAT	PCI interrupt status register
0x0300_0010	PCI_RD_UAB	PCI upper read address- bits register
0x0300_0014	PCI_WR_UAB	PCI upper write address - bits register
0x0300_0020	PCI_CFG_ADDR	PCI configuration address register
0x0300_0040	PCI_SDRAM	PCI SDRAM configuration register
0x0300_0400	PCI_CFG_DATA	PCI configuration data register

Register Descriptions

Common Register Set

COM_ID: Identification Register

Address Offset: 0000 0000h

Size (bytes): 0000 0004h

Access: 32-bit

The Identification Register is used to provide the companion device identification number for the host processor.

Name	Bit(s)	Type	Reset Value	Function
COM_ID	31:0	R	0x2	Identification Field This is the identification number for this companion device design.

COM_REV: Revision Register

Address Offset: 0000 0004h

Size (bytes): 0000 0004h

Access: 32-bit

The Revision Register is used to provide the companion device revision number for the host processor.

Name	Bit(s)	Type	Reset Value	Function
COM_REV	31:0	R	0x0	Revision Field This is the revision number for this companion device design.

COM_FEATURES: Companion Device Features Register

Address Offset: 0000 0008h

Size (bytes): 0000 0004h

Access: 32-bit

The Companion Device Features Register is used to indicate the features supported in this companion device design.

Name	Bit(s)	Type	Reset Value	Function
Reserved	31:5	R	0x0	N/A
TYPE1_CFG	4	R	0b1	PCI Type 1 Configuration Support This bit is set to 0b1 to indicate that PCI Type 1 configuration accesses are supported in this companion device design.
PCI_EN	3	R	0b1	PCI Interface Support This bit is set to 0b1 to indicate that the companion device supports an interface to PCI.
Reserved	2:0	R	0x0	N/A

COM_INTR_EN: Master Interrupt Enable Register

Address Offset: 0000 0014h

Size (bytes): 0000 0004h

Access: 32-bit

The Master Interrupt Enable Register is used to enable interrupt reporting through *CPU_INTn* to the host processor.

Name	Bit(s)	Type	Reset Value	Function
Reserved	31:4	R	0x0	N/A
COM_PCI_INTR_EN	3	R/W	0b0	PCI Interrupt Enable When this bit is set to 0b1, the companion device will interrupt the processor when a PCI-related function has generated an interrupt.
Reserved	2:0	R	0x0	N/A

COM_INTR_STAT: Master Interrupt Status Register

Address Offset: 0000 0018h

Size (bytes): 0000 0004h

Access: 32-bit

The Master Interrupt Status Register is used to provide interrupt status to the host processor.

Name	Bit(s)	Type	Reset Value	Function
Reserved	31:4	R	0x0	N/A
COM_PCI_INTR_STAT	3	R	0b0	PCI Interrupt Status This bit is set to 0b1 when a PCI-related function has generated an interrupt.
Reserved	2:0	R	0x0	N/A.

PCI Register Set

PCI_CFG: PCI Configuration Register

Address Offset: 0300 0000h

Size (bytes): 0000 0004h

Access: 32-bit

The PCI Configuration Register is used to control the CAS latency of the companion device's SDRAM interface.

Name	Bit(s)	Type	Reset Value	Function
PCI_CFG	31:26	R/W	0x0	QuickLogic Companion Device BAR Register The companion device responds to PCI transactions where the PCI address matches the upper address bits of this register <i>PCI_CFG[31:26]</i> .
Reserved	25:0	R	0x0	N/A

PCI_INTR_EN: PCI Interrupt Enable Register

Address Offset: 0300 0004h

Size (bytes): 0000 0004h

Access: 32-bit

The PCI Interrupt Enable Register is used to enable PCI interrupt reporting through *CPU_INTn* to the host processor.

Name	Bit(s)	Type	Reset Value	Function
Reserved	31:4	R	0x0	N/A
PCI_INTR_STAT2_EN	3	R/W	0b0	PCI Interrupt #2 Enable 1 = PCI interrupt #2 is passed to the host. 0 = PCI interrupt #2 is monitored but not passed to the host.
PCI_INTR_STAT1_EN	2	R/W	0b0	PCI Interrupt #1 Enable 1 = PCI interrupt #1 is passed to the host. 0 = PCI interrupt #1 is monitored but not passed to the host.
TGT_ABORT_DET_EN	1	R/W	0b0	Target Abort Interrupt Enable When this bit is set to 0b1, the companion device will interrupt the processor when the previous processor-to-PCI access was terminated with an abort by the PCI target.
MST_TTO_DET_EN	0	R/W	0b0	Master Time-Out Interrupt Enable When this bit is set to 0b1, the companion device will interrupt the processor when the previous processor-to-PCI access was terminated with a timeout by the companion device.

PCI_INTR_STAT: PCI Interrupt Status Register

Address Offset: 0300 0008h

Size (bytes): 0000 0004h

Access: 32-bit

The PCI Interrupt Status Register is used to provide PCI interrupt status to the host processor.

Name	Bit(s)	Type	Reset Value	Function
Reserved	31:4	R	0x0	N/A
PCI_INTR_STAT2	3	R	0b1	PCI Interrupt #2 Status This register indicates the current status of the PCI_INT2n signal.
PCI_INTR_STAT1	2	R	0b1	PCI Interrupt #1 Status This register indicates the current status of the PCI_INT1n signal.
TGT_ABORT_DET	1	R/W	0b0	Target Abort Status Flag When this bit is set to 0b1, the previous processor-to-PCI access was terminated with an abort by the PCI target. Write 0b1 to clear, write 0b0 has no effect.
MST_TTO_DET	0	R/W	0b0	Master Time-Out Status Flag When this bit is set to 0b1, the previous processor-to-PCI access was terminated with a timeout by the companion device. Write 0b1 to clear, write 0b0 has no effect.

PCI_RD_UAB: PCI Upper Read Address Bits Register

Address Offset: 0300 0010h

Size (bytes): 0000 0004h

Access: 32-bit

The PCI Upper Read Address Bits Register is used to set the upper address bits for PCI host read accesses.

Name	Bit(s)	Type	Reset Value	Function
PCI_RD_UAB	31:23	R/W	0x0	PCI Upper Read Address Bits Register This register sets the upper address bits for PCI host read accesses. $AD[31:23] = PCI_RD_UAB[31:23]$ $AD[22:2] = CPU_ADDR[22:2]$ $AD[1:0] = GND$
Reserved	22:0	R	0x0	N/A

PCI_WR_UAB: PCI Upper Write Address Bits Register

Address Offset: 0300 0014h

Size (bytes): 0000 0004h

Access: 32-bit

The PCI Upper Write Address Bits Register is used to set the upper address bits for PCI host write accesses.

Name	Bit(s)	Type	Reset Value	Function
PCI_WR_UAB	31:23	R/W	0x0	PCI Upper Write Address Bits Register This register sets the upper address bits for PCI host write accesses. $AD[31:23] = PCI_WR_UAB[31:23]$ $AD[22:2] = CPU_ADDR[22:2]$ $AD[1:0] = GND$
Reserved	22:0	R	0x0	N/A

PCI_CFG_ADDR: PCI Configuration Address Register

Address Offset: 0300 0020h

Size (bytes): 0000 0004h

Access: 32-bit

The PCI Configuration Address Register is used to enable/disable configuration space accesses at the PCI configuration space data port, and specify the bus number, device number, function number, and register number for the desired configuration read/write transaction.

Name	Bit(s)	Type	Reset Value	Function
CFG_EN	31	R/W	0x0	PCI Configuration Space Access Enable 1 = Accesses to the PCI configuration space data port are translated to configuration transactions on the PCI bus 0 = Accesses to the PCI configuration space data port are ignored for write accesses and return invalid data for read accesses
Reserved	30:24	R	0x0	N/A
BUS_NUM	23:16	R/W	0x0	PCI Bus Number This field selects a specific PCI bus in the system. When the bus number is equal to 0x0, a PCI Type 0 configuration access translation will be generated. When the bus number is not equal 0x0, a PCI Type 1 configuration access translation will be generated.

Name	Bit(s)	Type	Reset Value	Function
DEV_NUM	15:11	R/W	0x0	PCI Device Number This field selects a PCI device on the specified PCI bus. Only device numbers 0x0 to 0x14 are supported. For PCI Type 0 configuration accesses, the device number will be translated to assert the appropriate IDSEL (<i>AD[11]</i> through <i>AD[31]</i>) line.
FUNCT_NUM	10:8	R/W	0x0	PCI Device Function Number This field selects a function in the specified PCI device if multiple functions are supported.
REG_NUM	7:2	R/W	0x0	PCI Configuration Space Register Number This field selects a DWORD in the specified PCI device's configuration space.
Reserved	1:0	R	0x0	N/A

PCI_SDRAM: PCI SDRAM Configuration Register

Address Offset: 0300 0040h

Size (bytes): 0000 0004h

Access: 32-bit

The PCI SDRAM Configuration Register is used to control the CAS latency of the companion device's SDRAM interface.

Name	Bit(s)	Type	Reset Value	Function
Reserved	31:1	R	0x0	N/A
CAS_LAT	0	R/W	0b0	CAS Latency Field This field controls the CAS latency of the companion device's SDRAM interface. 0 - CAS latency is set to 2 1 - CAS latency is set to 3

PCI_CFG_DATA: PCI Configuration Data Register

Address Offset: 0300 0400h

Size (bytes): 0000 0004h

Access: 32-bit

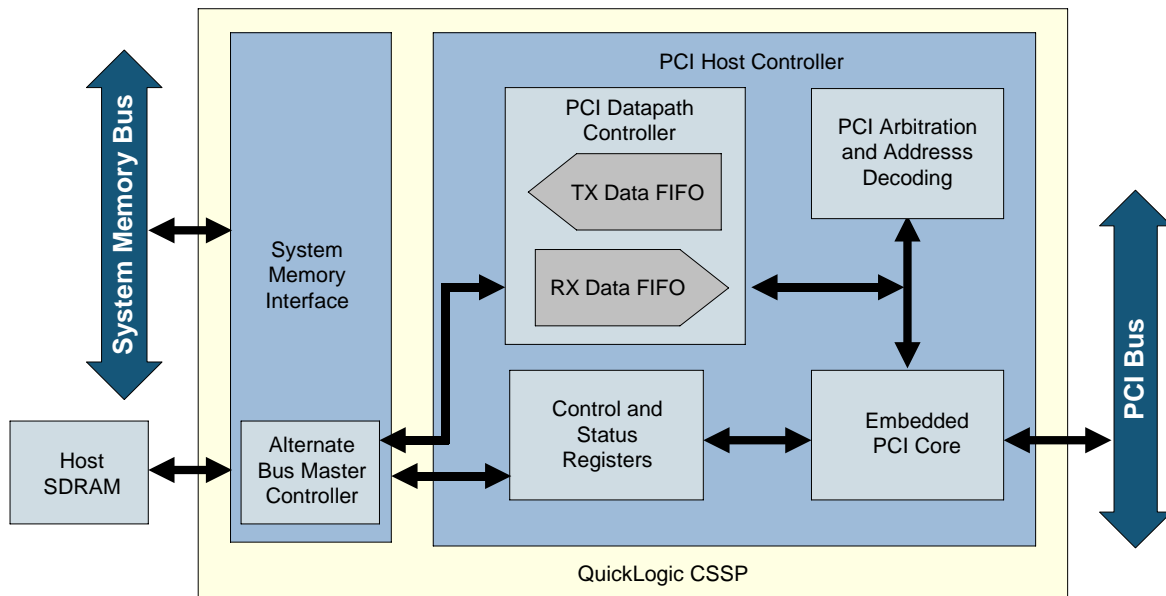
The PCI Configuration Data Register is used to access the configuration space of PCI device #1 or PCI device #2.

Name	Bit(s)	Type	Reset Value	Function
PCI_CFG_DATA	31:0	R/W	X	<p>PCI Configuration Data Register</p> <p>Accesses to the PCI configuration data register are translated to configuration read or write transactions on the PCI bus. The companion device performs the address translation based on the value previously written to the PCI_CFG_ADDR register.</p>

Functional and Module Description

The PCI Host Controller implements connectivity between the peripherals and the processor, and consists of the following functional blocks as shown in **Figure 2**.

Figure 2: PCI Host Controller Block Diagram



Memory Interface

The Memory Interface communicates to the System Memory Bus. It is responsible for decoding the local address of the transfer, and directing the transaction to the QuickLogic CSSP registers or PCI. All local bus byte lanes are passed through during PCI accesses but are otherwise assumed to be enabled at all times.

The processor can initiate PCI configuration and memory accesses via this module. Depending on the decoded local address, this module will activate the PCI data path controller to complete the transaction.

PCI Core and PCI Host Functionalities

The QuickLogic PCI Host Controller operates with a built-in arbiter. The arbiter is responsible for managing PCI bus control between the QuickLogic Companion Device and two PCI devices. The arbitration scheme is round-robin, and the bus is parked on PCI device #1 by default.

Upon power-up or reset, PCI configuration accesses are used by the processor to enumerate the PCI bus. Each PCI Host Controller has its own configuration space which is individually mapped to the PCI configuration space data port through the `PCI_CFG_ADDR` register. From the contents written to this register, the companion device performs the proper translation so that the processor can generate Type 0 or Type 1 configuration cycles to each PCI device.

PCI Data Path Controller

Each PCI agent can access the host SDRAM via the PCI data path controller. From the processor side, the processor can initiate read/write transactions to PCI memory and configuration spaces.

The PCI data path controller communicates to the embedded PCI core. This module has a PCI master function and a PCI slave function. It also supports PCI-related interrupt monitoring and reporting via the `PCI_INTR_EN` and `PCI_INTR_STAT` registers.

When requested by the processor interface, the PCI master function works with the PCI core to perform the desired transaction on the PCI bus. All four processor byte enables are passed on when reading or writing to PCI devices. The PCI slave function makes the host SDRAM memory accessible from PCI. The complete 32-bit PCI address used is constructed from `CPU_ADDR[22:2]` and the `PCI_RD_UAB` or `PCI_WR_UAB` register, depending on the transfer direction. All four PCI byte enables are passed on when writing to host memory.

The PCI data path controller contains two FIFOs to buffer the dataflow between SDRAM and PCI, so the PCI slave function can support burst transfers. PCI-to-SDRAM transfers are buffered in the Receive FIFO, and the SDRAM interface is activated to move data from the Receive FIFO into the SDRAM. SDRAM-to-PCI transfers are retried on PCI until the SDRAM interface fetches data from the SDRAM into the Transmit FIFO, then a burst transfer occurs from the Transmit FIFO to PCI. The SDRAM interface prefetches data from the SDRAM until the PCI transaction is complete.

Processor-initiated PCI transfers are intended for control purposes such as setting up the PCI Configuration Spaces and DMA engines in each PCI agent, so the PCI master function does not initiate burst transfers on PCI.

SDRAM Interface

The SDRAM interface arbitrates for the host processor memory bus, and controls the host SDRAM during transfers initiated from PCI.

When a device on PCI tries to write to host memory, this module is activated to pop data from the Receive FIFO of the PCI data path controller into host SDRAM memory. When a device on PCI tries to read from host memory, this module is activated to push data into the Transmit FIFO of the PCI data path controller from host SDRAM memory. The SDRAM controller also monitors timing during data transfers so that the memory bus is relinquished allowing the host processor to perform refresh cycles.

The processor can control the CAS latency of the SDRAM interface via the PCI_SDRAM register.

Supported Operating Systems

The PCI Host Controller PSB supports the following operating systems:

- Windows[®] CE
- Windows Mobile[®]
- Linux[®]

Contact Information

Phone: (408) 990-4000 (US)
(905) 940-4149 (Canada)
+(44) 1932-57-9011 (Europe)
+(852) 2567-5441 (Asia)

E-mail: info@quicklogic.com

Sales: America-sales@quicklogic.com
Europe-sales@quicklogic.com
Asia-sales@quicklogic.com
Japan-sales@quicklogic.com

Support: www.quicklogic.com/support

Internet: www.quicklogic.com

Revision History

Revision	Date	Originator and Comments
A	July 2008	First release.

Notice of Disclaimer

QuickLogic is providing this design, product or intellectual property "as is." By providing the design, product or intellectual property as one possible implementation of your desired system-level feature, application, or standard, QuickLogic makes no representation that this implementation is free from any claims of infringement and any implied warranties of merchantability or fitness for a particular purpose. You are responsible for obtaining any rights you may require for your system implementation. QuickLogic shall not be liable for any damages arising out of or in connection with the use of the design, product or intellectual property including liability for lost profit, business interruption, or any other damages whatsoever. QuickLogic products are not designed for use in life-support equipment or applications that would cause a life-threatening situation if any such products failed. Do not use QuickLogic products in these types of equipment or applications.

QuickLogic does not assume any liability for errors which may appear in this document. However, QuickLogic attempts to notify customers of such errors. QuickLogic retains the right to make changes to either the documentation, specification, or product without notice. Verify with QuickLogic that you have the latest specifications before finalizing a product design.

Copyright and Trademark Information

Copyright © 2008 QuickLogic Corporation. All Rights Reserved.

The information contained in this document is protected by copyright. All rights are reserved by QuickLogic Corporation. QuickLogic Corporation reserves the right to modify this document without any obligation to notify any person or entity of such revision. Copying, duplicating, selling, or otherwise distributing any part of this product without the prior written consent of an authorized representative of QuickLogic is prohibited.

QuickLogic and ViaLink are registered trademarks; and the QuickLogic logo is a trademark of QuickLogic. Other trademarks are the property of their respective companies.