LogiCORE IP 64-Bit Initiator/Target v3 & v4 for PCI

DS205 July 25, 2012 **Product Specification v3.167 & v4.17**

Features

- Fully compatible 64-bit, 66/33 MHz LogiCORE™ IP Initiator/Target core for PCI™
- Customizable, programmable, single-chip solution
- Pre-defined implementation for predictable timing
- Incorporates Xilinx Smart-IP technology
- 3.3V operation at 0–66 MHz
- 5.0V operation at 0–33 MHz
- Fully verified design tested with Xilinx proprietary test bench and hardware
- Delivered through Xilinx® CORE Generator™ tool and Vivado™ IP Catalog
- CardBus compliant
- Supported initiator functions:
  - Configuration read, configuration write
  - Memory read, memory write, MRM, MRL
  - Interrupt acknowledge, special cycles
  - I/O read, I/O write
- Supported target functions:
  - Type 0 configuration space header
  - Up to three base address registers (MEM or I/O with adjustable block size from 16 bytes to 2 GB)
  - Medium decode speed
  - Parity generation, parity error detection
  - Configuration read, configuration write
  - Memory read, memory write, MRM, MRL
  - Interrupt acknowledge
  - I/O read, I/O write
  - Target abort, target retry, target disconnect

LogiCORE IP Facts		
Core Specifics		
Supported Device Family(1)	See Table 1.	
Resource Utilization(2)	**v4 Core**	**v3 Core**
LUTs	565	724
Slice Flip-Flops	404	732
IOB Flip-Flops	94	176
IOBs	94	89
GCLK(3)	2	1
Provided with Core		
Documentation	Product Specification v3 & v4 Getting Started Guide v3 User Guide v4 User Guide v3	
Design File Formats	ISE: VHDL/Verilog Simulation Model ISE: NGC Netlist (v4 core only) ISE: NGO Netlist (v3 core only) Vivado: Encrypted RTL	
Constraints File	ISE: UCF Vivado: XDC	
Test Bench	VHDL/Verilog Example Test Bench	
Instantiation Template	VHDL/Verilog Wrapper	
Example Designs	VHDL/Verilog Example Design	
Design Tool Requirements(4)		
Xilinx Implementation Tools	ISE® Design Suite v14.2 Vivado Design Suite v2012.2(5)	
Simulation	Mentor Graphics ModelSim Cadence Incisive Enterprise Simulator (IES)	
Synthesis	Xilinx XST Vivado Synthesis	
Support		
Provided by Xilinx @ www.xilinx.com/support		

1. For a complete listing of supported devices, see the release notes for this core.
2. Resource utilization depends on core configuration and design requirements. Unused resources are trimmed by the Xilinx technology mapper. Utilization figures reported represent a maximum configuration.
3. Designs running at 66 MHz in Virtex®-4 and Virtex-5 FPGA implementations require additional BUFG for 200 MHz reference clock.
4. For the supported versions of the tools, see the Xilinx Design Tools: Release Notes Guide.
5. Supports 7 series devices only.

Xilinx provides technical support for this LogiCORE IP product when used as described. Xilinx cannot guarantee timing, functionality, or support of product if implemented in devices not listed, or if customized beyond that described in the related product documentation.

For Spartan®-6 devices, only those devices listed in Table 1 have been tested with the latest software speed files to meet PCI timing. For a part or package not listed in the data sheet, open a WebCase with Xilinx for latest available status.

Table 1: **Core Implementation**

Supported Device[1],[2]	Core Version	Signaling Environment
64/66 Core		
Virtex-5 XC5VFX70T-FF1136-2C/I[3] (regional clock)	v4	3.3V only
Virtex-5 XC5VLX50-FF1153-2C/I[3] (regional clock)	v4	3.3V only
Virtex-5 XC5VLX50T-FF1136-2C/I[3] (regional clock)	v4	3.3V only
Virtex-5 XC5VLX110-FF1153-2C/I[3] (regional clock)	v4	3.3V only
Virtex-5 XC5VLX110T-FF1136-2C/I[3] (regional clock)	v4	3.3V only
Virtex-5 XC5VSX50T-FF1136-2C/I[3] (regional clock)	v4	3.3V only
Virtex-5 XC5VSX95T-FF1136-2C/I[3] (regional clock)	v4	3.3V only
Virtex-4 XC4VFX20-FF672-11C/I[3] (regional clock)	v3	3.3V only
Virtex-4 XC4VLX25-FF668-11C/I[3],[4] (regional clock)	v3	3.3V only
Virtex-4 XC4VSX35-FF668-11C/I[3],[4] (regional clock)	v3	3.3V only
Spartan-3A XC3S400A-FG400-5C	v3	3.3V only
Spartan-3A XC3S700A-FG400-5C	v3	3.3V only
Spartan-3A XC3S700A-FG484-5C	v3	3.3V only
Spartan-3A XC3S1400A-FG484-5C	v3	3.3V only
Spartan-3A XC3S1400A-FG676-5C	v3	3.3V only
Spartan-3ADSP XC3SD1800A-FG676-5C	v3	3.3V only
Spartan-3ADSP XC3SD3400A-FG676-5C	v3	3.3V only
Spartan-3E XC3S1200E-FG400-5C[4]	v3	3.3V only
64/33 Core		
Kintex™-7 XC7K70T-SBG324-1C/I	v4	3.3V only
Kintex-7 XC7K70T-FBG484-1C/I	v4	3.3V only
Kintex-7 XC7K70T-FBG676-1C/I	v4	3.3V only
Kintex-7 XC7K160T-FBG484-1C/I	v4	3.3V only
Kintex-7 XC7K160T-FBG676-1C/I	v4	3.3V only
Kintex-7 XC7K160T-FFG676-1C/I	v4	3.3V only
Kintex-7 XC7K325T-FBG676-1C/I	v4	3.3V only
Kintex-7 XC7K325T-FBG900-1C/I	v4	3.3V only
Kintex-7 XC7K325T-FFG676-1C/I	v4	3.3V only
Kintex-7 XC7K325T-FFG900-1C/I	v4	3.3V only
Kintex-7 XC7K355T-FFG901-1C/I	v4	3.3V only
Kintex-7 XC7K410T-FBG676-1C/I	v4	3.3V only
Kintex-7 XC7K410T-FBG900-1C/I	v4	3.3V only

Table 1: **Core Implementation *(Cont'd)***

Supported Device[1],[2]	Core Version	Signaling Environment
Kintex-7 XC7K410T-FFG676-1C/I	v4	3.3V only
Kintex-7 XC7K410T-FFG900-1C/I	v4	3.3V only
Kintex-7 XC7K420T-FFG901-1C/I	v4	3.3V only
Kintex-7 XC7K420T-FFG1156-1C/I	v4	3.3V only
Kintex-7 XC7K480T-FFG901-1C/I	v4	3.3V only
Kintex-7 XC7K480T-FFG1156-1C/I	v4	3.3V only
Artix™-7 XC7A100T-CSG324-1C/I	v4	3.3V only
Artix-7 XC7A100T-FGG484-1C/I	v4	3.3V only
Artix-7 XC7A100T-FGG676-1C/I	v4	3.3V only
Artix-7 XC7A200T-FBG484-2C/I	v4	3.3V only
Artix-7 XC7A200T-FBG676-2C/I	v4	3.3V only
Artix-7 XC7A350T-CBG484-2C/I	v4	3.3V only
Artix-7 XC7A350T-FBG676-2C/I	v4	3.3V only
Virtex-5 XC5VFX70T-FF1136-1C/[3] (global clock)	v4	3.3V only
Virtex-5 XC5VFX70T-FF1136-1C/I[3] (regional clock)	v4	3.3V only
Virtex-5 XC5VLX50-FF1153-1 C/I[3] (global clock)	v4	3.3V only
Virtex-5 XC5VLX50-FF1153-1C/I[3] (regional clock)	v4	3.3V only
Virtex-5 XC5VLX50T-FF-1136-1C[3] (global clock)	v4	3.3V only
Virtex-5 XC5VLX50T-FF-1136-1C[3] (regional clock)	v4	3.3V only
Virtex-5 XC5VLX110-FF1153-1C/I[3] (global clock)	v4	3.3V only
Virtex-5 XC5VLX110-FF1153-1C/I[3] (regional clock)	v4	3.3V only
Virtex-5 XC5VLX110T-FF1136-1C/I[3] (global clock)	v4	3.3V only
Virtex-5 XC5VLX110T-FF1136-1C/I[3] (regional clock)	v4	3.3V only
Virtex-5 XC5VSX50T-FF-1136-1C[3] (global clock)	v4	3.3V only
Virtex-5 XC5VSX50T-FF-1136-1C[3] (regional clock)	v4	3.3V only
Virtex-5 XC5VSX95T-FF-1136-1C[3] (global clock)	v4	3.3V only
Virtex-5 XC5VSX95T-FF-1136-1C[3] (regional clock)	v4	3.3V only
Virtex-4 XC4VFX20-FF672-10C/I[3] (global clock)	v3	3.3V only
Virtex-4 XC4VFX20-FF672-10C/I[3] (regional clock)	v3	3.3V only
Virtex-4 XC4VLX25-FF668-10C/I[3],[4] (global clock)	v3	3.3V only
Virtex-4 XC4VLX25-FF668-10C/I[3],[4] (regional clock)	v3	3.3V only
Virtex-4 XC4VSX35-FF668-10C/I[3],[4] (global clock)	v3	3.3V only
Virtex-4 XC4VSX35-FF668-10C/I[3],[4] (regional clock)	v3	3.3V only
Spartan-6 XC6SLX9-FTG256-2C/I	v4	3.3V only
Spartan-6 XC6SLX9-CSG324-2C/I	v4	3.3V only
Spartan-6 XC6SLX16-FTG256-2C/I	v4	3.3V only
Spartan-6 XC6SLX16-CSG324-2C/I	v4	3.3V only
Spartan-6 XC6SLX25-FTG256-2C/I	v4	3.3V only
Spartan-6 XC6SLX25-CSG324-2C/I	v4	3.3V only

Table 1: **Core Implementation *(Cont'd)***

Supported Device[1],[2]	Core Version	Signaling Environment
Spartan-6 XC6SLX25-FGG484-2C/I	v4	3.3V only
Spartan-6 XC6SLX25T-FGG484-2C/I	v4	3.3V only
Spartan-6 XC6SLX45-CSG324-2C/I	v4	3.3V only
Spartan-6 XC6SLX45-FGG484-2C/I	v4	3.3V only
Spartan-6 XC6SLX45T-FGG484-2C/I	v4	3.3V only
Spartan-6 XC6SLX45-FGG676-2C/I	v4	3.3V only
Spartan-6 XC6SLX45-CSG484-2C/I	v4	3.3V only
Spartan-6 XC6SLX45T-CSG484-2C/I	v4	3.3V only
Spartan-6 XC6SLX75-CSG484-2C/I/Q	v4	3.3V only
Spartan-6 XC6SLX75T-CSG484-2C/I/Q	v4	3.3V only
Spartan-6 XC6SLX75-FGG484-2C/I	v4	3.3V only
Spartan-6 XC6SLX75T-FGG484-2C/I	v4	3.3V only
Spartan-6 XC6SLX75-FGG676-2C/I	v4	3.3V only
Spartan-6 XC6SLX75T-FGG676-2C/I	v4	3.3V only
Spartan-6 XC6SLX100-CSG484-2C/I	v4	3.3V only
Spartan-6 XC6SLX100T-CSG484-2C/I	v4	3.3V only
Spartan-6 XC6SLX100-FGG484-2C/I	v4	3.3V only
Spartan-6 XC6SLX100T-FGG484-2C/I	v4	3.3V only
Spartan-6 XC6SLX100-FGG676-2C/I	v4	3.3V only
Spartan-6 XC6SLX100T-FGG676-2C/I	v4	3.3V only
Spartan-6 XC6SLX100-FGG900-2C/I	v4	3.3V only
Spartan-6 XC6SLX100T-FGG900-2C/I	v4	3.3V only
Spartan-6 XC6SLX150-CSG484-2C/I	v4	3.3V only
Spartan-6 XC6SLX150T-CSG484-2C/I	v4	3.3V only
Spartan-6 XC6SLX150-FGG484-2C/I	v4	3.3V only
Spartan-6 XC6SLX150T-FGG484-2C/I	v4	3.3V only
Spartan-6 XC6SLX150-FGG676-2C/I	v4	3.3V only
Spartan-6 XC6SLX150T-FGG676-2C/I	v4	3.3V only
Spartan-6 XC6SLX150-FGG900-2C/I	v4	3.3V only
Spartan-6 XC6SLX150T-FGG900-2C/I	v4	3.3V only
Spartan-3A XC3S400A-FG400-4C/I	v3	3.3V only
Spartan-3A XC3S700A-FG400-4C/I	v3	3.3V only
Spartan-3A XC3S700A-FG484-4C/I	v3	3.3V only
Spartan-3A XC3S1400A-FG484-4C/I	v3	3.3V only
Spartan-3A XC3S1400A-FG676-4C/I	v3	3.3V only
Spartan-3ADSP XC3SD1800A-FG676-4C/I	v3	3.3V only
Spartan-3ADSP XC3SD3400A-FG676-4C/I	v3	3.3V only

Table 1: **Core Implementation *(Cont'd)***

Supported Device[1],[2]	Core Version	Signaling Environment
Spartan-3E XC3S1200E-FG400-4C/I[4]	v3	3.3V only
Spartan-3 XC3S1000-FG456-4C/I	v3	3.3V only

1. Packages listed are supported in both standard and lead-free variants, if available. For example, FF1136 denotes support for both FF1136 and FGG1136 packages.
2. For additional part/package combinations in Spartan-3 and older device families, see the UCF Generator located at www.xilinx.com/cgi-bin/UCFgen/UCF4PCI.cgi. For Spartan-3E, Spartan-3A, Spartan-3AN, Spartan-3A DSP, Virtex-4, and newer device families, use the UCF Generator in the CORE Generator software.
3. Virtex-4 and Virtex-5 FPGA solutions require a 200 MHz reference clock.
4. Virtex-4 (except FX) and Spartan-3E FPGA solutions require stepping silicon 1.

Applications

- Embedded applications in networking, industrial, and telecommunication systems
- Add-in for PCI boards such as frame buffers, network adapters, and data acquisition boards
- Hot swap CompactPCI boards
- CardBus compliant
- Any application requiring a PCI interface

General Description

The Initiator/Target core for PCI is a pre-implemented and fully tested module for Xilinx FPGAs. The pinout for each device and the relative placement of the internal logic are predefined. Critical paths are controlled by constraints files to ensure predictable timing, significantly reducing the engineering time required to implement your design. Resources can instead be focused on your unique user application logic in the FPGA and on the system-level design. As a result, the Xilinx products for PCI minimize product development time.

The core meets the setup, hold, and clock-to-timing requirements as specified in the PCI specification, and is verified through extensive simulation.

Other FPGA resources that can be used in conjunction with the core to enable efficient implementation include:

- Block SelectRAM™ memory. Blocks of on-chip ultra-fast RAM with synchronous write and dual-port RAM capabilities. Used in designs for PCI to implement FIFOs.
- SelectRAM memory. Distributed on-chip ultra-fast RAM with synchronous write option and dual-port RAM capabilities. Used in designs for PCI to implement FIFOs.
- Internal three-state bus capability for data multiplexing.

The interface is carefully optimized for best possible performance and utilization in Xilinx FPGAs.

Smart-IP Technology

Drawing on the architectural advantages of Xilinx FPGAs, Xilinx Smart-IP technology ensures the highest performance, predictability, repeatability, and flexibility in PCI designs. The Smart-IP technology is incorporated in every Initiator/Target core for PCI. Xilinx Smart-IP technology leverages the Xilinx architectural advantages such as look-up tables and segmented routing. Also included is floor planning information, such as logic mapping and location constraints. This technology provides the best physical layout, predictability, and performance. In addition, these features allow for significantly reduced compile times over competing architectures.

To guarantee the critical setup, hold, minimum clock-to-out, and maximum clock-to-out timing, the core is delivered with Smart-IP constraint files that are unique for a device and package combination. These constraints files guide the implementation tools so that the critical paths always are within specification.

Xilinx provides Smart-IP constraints files for many device and package combinations. Constraints files for unsupported device and package combinations can be generated using the web-based constraints file generator.

Functional Description

Figure 1 illustrates the Initiator/Target core for PCI partitioned into five major blocks and a user application.

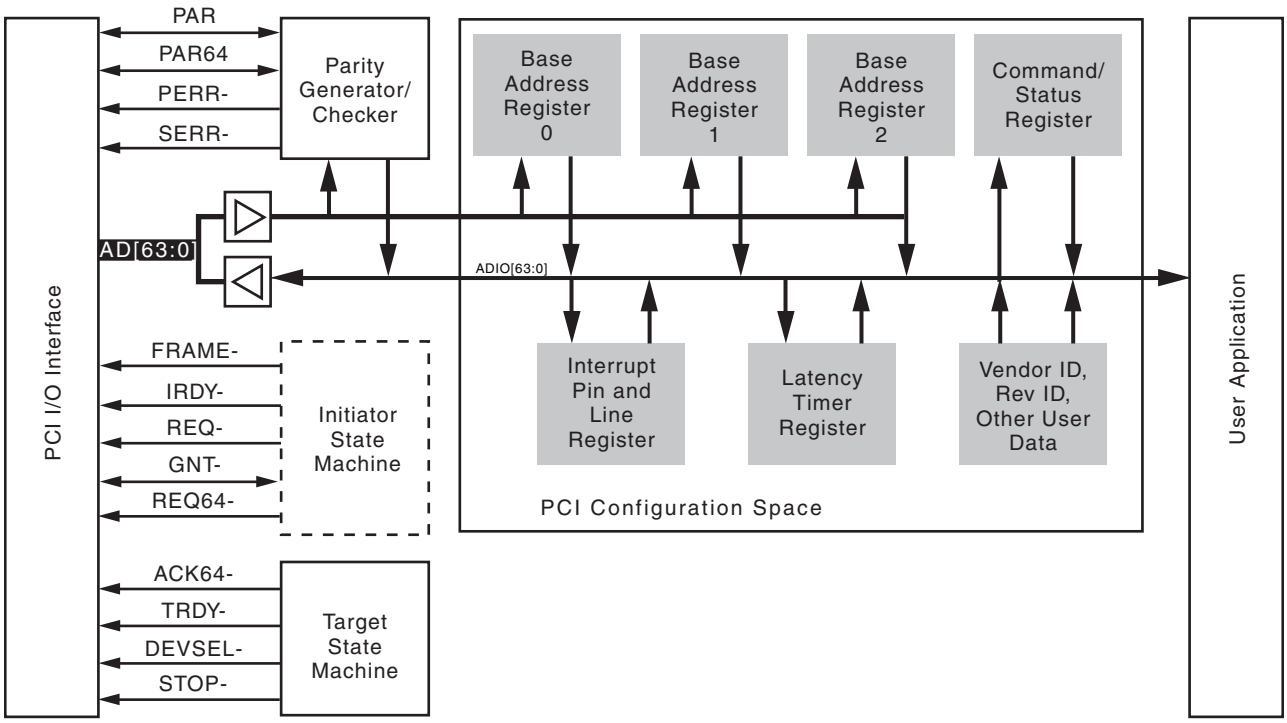

Figure 1: **Block Diagram of the Initiator/Target Core for PCI**

I/O Interface Block for PCI

The I/O interface block handles the physical connection to the PCI bus including all signaling, input and output synchronization, output 3-state controls, and all request-grant handshaking for bus mastering.

User Application

The Initiator/Target core for PCI provides a simple, general-purpose interface for a wide range of applications.

Configuration Space

This block provides the first 64 bytes of Type 0, v3.0 Configuration Space Header (Table 2) to support software-driven Plug-and-Play initialization and configuration. This includes information for Command, Status, and three Base Address Registers (BARs).

The capability to extend configuration space has been built into the user application interface. This, and the ability to implement a capabilities pointer in configuration space, allows you to implement functions such as power management and message signaled interrupts in your application.

Table 2: **Configuration Space Header for PCI**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Device ID																Vendor ID																00h
Status																Command																04h
Class Code																								Rev ID								08h
BIST								Header Type								Latency Timer								*Cache Line Size*								0Ch
Base Address Register 0 (BAR0)																																10h
Base Address Register 1 (BAR1)																																14h
Base Address Register 2 (BAR2)																																18h
Base Address Register 3 (BAR3)																																1Ch
Base Address Register 4 (BAR4)																																20h
Base Address Register 5 (BAR5)																																24h
Cardbus CIS Pointer																																28h
Subsystem ID																Subsystem Vendor ID																2Ch
Expansion ROM Base Address																																30h
Reserved																								CapPtr								34h
Reserved																																38h
Max Lat								Min Gnt								Int Pin								Int Line								3Ch
Reserved																																40h-FFh

Notes: Shaded areas are not implemented and return zero.

Parity Generator/Checker

This block generates and checks even parity across the AD bus, the CBE# lines, and the parity signals. It also reports data parity errors using PERR# and address parity errors with SERR#.

Initiator State Machine

This block controls the Initiator/Target core for PCI initiator functions. The states implemented are a subset of those defined in Appendix B of the *PCI Local Bus Specification*. The initiator control logic uses one-hot encoding for maximum performance.

Target State Machine

This block controls the Initiator/Target core for PCI target functions. The states implemented are a subset of those defined in Appendix B of the *PCI Local Bus Specification*. The target control logic uses one-hot encoding for maximum performance.

Core Configuration

The Initiator/Target core for PCI can be configured to fit unique system requirements using the Xilinx CORE Generator tool GUI. For the v3 core, the HDL configuration file can be changed to configure the core after it is generated. These customization options, among many others, are supported by the core:

- Device and vendor ID
- Base Address Registers (number, size, and type)

See the *Initiator/Target v3 for PCI User Guide* (UG159) and the *Initiator/Target v4 for PCI User Guide* (UG262).

Burst Transfer

The PCI bus derives its performance from its ability to support burst transfers. The performance of any PCI application depends largely on the size of the burst transfer. Buffers to support PCI burst transfer can efficiently be implemented using on-chip RAM resources.

Supported PCI Commands

Table 3 illustrates the PCI bus commands supported by the core.

Table 3: **Bus Commands for PCI**

CBE [3:0]	Command	PCI Initiator	PCI Target
0000	Interrupt Acknowledge	Yes	Yes
0001	Special Cycle	Yes	Ignore
0010	I/O Read	Yes	Yes
0011	I/O Write	Yes	Yes
0100	Reserved	Ignore	Ignore
0101	Reserved	Ignore	Ignore
0110	Memory Read	Yes	Yes
0111	Memory Write	Yes	Yes
1000	Reserved	Ignore	Ignore
1001	Reserved	Ignore	Ignore
1010	Configuration Read	Yes	Yes
1011	Configuration Write	Yes	Yes
1100	Memory Read Multiple	Yes	Yes
1101	Dual Address Cycle	No	Ignore
1110	Memory Read Line	Yes	Yes
1111	Memory Write Invalidate	No	Yes

Bandwidth

The Initiator/Target core for PCI supports fully compliant zero wait-state burst operations for sourcing and receiving data. This core supports a sustained bandwidth of up to 528 MBytes per second. The design can be configured for very long bursts.

The flexible user application interface, combined with support for many different PCI features, provides a solution that lends itself to use in many high-performance applications. Because you are not locked into one DMA engine, you can create an optimized design for your application.

Recommended Design Experience

The Initiator/Target core for PCI is pre-implemented, allowing engineering focus on the unique user application functions of a PCI design. However, PCI is a high-performance design that is challenging to implement in any technology. For this reason, previous experience building high-performance, pipelined FPGA designs using Xilinx implementation software and constraint files is recommended. The challenge to implement a complete PCI design including user application functions varies, depending on the configuration and functionality of your application. Contact your local Xilinx representative for a closer review and estimation for your specific requirements.

Timing Specifications

The maximum speed at which the user design runs can be affected by the size and quality of the design. Table 4 and Table 5 show the key timing parameters for the PCI interface. Table 4 lists the timing parameters in the 66 MHz implementations, and Table 5 lists the timing parameters in the 33 MHz implementations.

Table 4: **Timing Parameters, 66 MHz Implementation**

Symbol	Parameter	Min	Max
T_{cyc}	CLK Cycle Time	15[1]	30
T_{high}	CLK High Time	6	-
T_{low}	CLK Low Time	6	-
T_{val}	CLK to Signal Valid Delay (bussed signals)	2[2]	6[2]
T_{val}	CLK to Signal Valid Delay (point to point signals)	2[2]	6[2]
T_{on}	Float to Active Delay	2[2]	-
T_{off}	Active to Float Delay	-	14[1]
T_{su}	Input Setup Time to CLK (bussed signals)	3[2,3]	-
T_{su}	Input Setup Time to CLK (point to point signals)	5[2,3]	-
T_{h}	Input Hold Time from CLK	0[2,3]	-
T_{rstoff}	Reset Active to Output Float	-	40

Notes

1. Controlled by timespec constraints, included in product.
2. Controlled by SelectIO™ interface configured for PCI66_3.
3. Controlled by directed-routing constraints, included in product.

Table 5: **Timing Parameters, 33 MHz Implementations**

Symbol	Parameter	Min	Max
T_{cyc}	CLK Cycle Time	30[1]	-
T_{high}	CLK High Time	11	-
T_{low}	CLK Low Time	11	-
T_{val}	CLK to Signal Valid Delay (bussed signals)	2[2]	11[2]
T_{val}	CLK to Signal Valid Delay (point to point signals)	2[2]	11[2]
T_{on}	Float to Active Delay	2[2]	-
T_{off}	Active to Float Delay	-	28[1]
T_{su}	Input Setup Time to CLK (bussed signals)	7[2]	-
T_{su}	Input Setup Time to CLK (point to point signals)	10[2]	-
T_{h}	Input Hold Time from CLK	0[2]	-
T_{rstoff}	Reset Active to Output Float	-	40

Notes
1. Controlled by timespec constraints, included in product.
2. Controlled by SelectIO interface configured for PCI33_3 or PCI33_5.

Support

Xilinx provides technical support for this LogiCORE IP product when used as described in the product documentation. Xilinx cannot guarantee timing, functionality, or support of product if implemented in devices that are not defined in the documentation, if customized beyond that allowed in the product documentation, or if changes are made to any section of the design labeled *DO NOT MODIFY*.

Ordering Information

The Initiator/Target core for PCI is accessed through the Xilinx CORE Generator software. The CORE Generator tool is bundled with the ISE Design Suite at no additional charge. To purchase the Initiator/Target core for PCI, contact your local Xilinx sales representative.

Visit the Initiator/Target for PCI/PCI-X product offering page for more information.

Revision History

The following table shows the revision history for this document:

Date	Version	Description of Revisions
08/30/02	1.2	Style update.
12/18/02	1.3	Updated to V3.0.103; v5.Ii, 1st feature: 32-bit was 64/32-bit
3/3/03	1.4	Revised date to 3/3/03; updated to v3.0.105; v5.2i.
4/14/03	1.5	Updated to v3.0.106; Updated Facts Table PCI64/33 product listings to include Spartan-3 device support.

Date	Version	Description of Revisions
5/8/03	1.6	Updated Xilinx tools to 5.2i SP2; added note 10.
9/17/03	1.7	Updated to v3.0.113; Xilinx Tools v6.1i SP1 was v5.2i SP2; date was May 8, 2003.
11/10/03	1.8	Updated to v3.0.116; in Supported Products table, PCI64/66 section, Virtex-II Pro XC2VP7FF672-6C was -7C.
12/17/03	1.9	Updated device support information in Facts Table (added Virtex-II Pro devices).
1/7/04	1.10	Updated to v3.0.117; in Supported Products table, sections PCI64/66 and PCI64/33, added Virtex-II Pro XC2VP20...through XC2VP50; updated Xilinx tools to 5.2i SP3.
1/30/04	1.11	Updated to v3.0.122, updated copyright information to 2004.
4/9/04	1.12	Updated to v3.0.126; updated Xilinx tools to 6.2i SP1; added notes 11 and 12 to Supported devices table; added suffix /I to all Virtex-II Pro devices.
4/26/04	1.13	Updated build version to v3.0.128, updated Xilinx tools to 6.2i SP2, changed date to April 26, 2004.
7/15/04	1.14	Updated build to v3.0.129, and added support for Xilinx tools v6.2i SP3. The data sheet is updated to the new template.
11/11/04	1.15	Updated support for Xilinx tools v6.3i; updated PCI spec to v3.0; added Exemplar LeonardoSpectrum and Cadence NC-Verilog entry and verification tools.
12/8/04	1.17	Updated to build 3.0.140 and Virtex-4 support.
3/7/05	1.18	Updated to Xilinx tools 7.1i and build 3.0.145.
5/13/05	2.0	Updated build to 3.0.150, added support for Spartan-3E, addition of SP2.
8/31/05	3.0	Updated build to 3.0.151, updated SP2 to SP3 for 7.1i
9/12/05	4.0	Updated build to 3.0.152, updated SP3 to SP4 for 7.1i, removed reference to Spartan-3 as pending from note in Table 1. Moved Table 3 to follow text reference.
1/18/06	5.0	Updated build to 3.0.155, ISE v8.1i, and release date
7/13/06	6.0	Added support for Virtex-5, core v4, ISE to v8.2i, build to 160, release date
2/15/07	7.0	ISE v9.1i, Virtex-5 LXT support, build to 161, release date. Added support for Spartan-3A and Spartan-3E for 66 MHz.
5/17/07	7.5	Updated references to PCI for compliance to PCI-SIG trademark guidelines. Advanced support for Cadence IUS to v5.7.
08/08/07	8.0	Minor updates for supported tool versions for IP1 Jade Minor release, build no. to 163.
10/10/07	8.5	Updated release date.
3/24/08	9.0	Updated to support ISE v10.1.
4/25/08	9.5	V4 build 7 PCI core update only for adding Virtex-5 FXT support.
9/19/08	10.0	Updated to support ISE v10.1 Service Pack 3.
4/24/09	10.5	Updated to support ISE v11.1 and Spartan-6 FPGAs. Removed support for deprecated devices: Virtex-II, Virtex-II Pro, and Virtex-E.
6/24/09	11.0	Updated to support ISE v11.2.
9/16/09	12.0	Updated to v4.10 and ISE v11.3. Added additional Spartan-6 FPGA part and package support.
12/02/09	12.5	Updated to ISE v11.4. Added additional Spartan-6 FPGA part and package support.

Date	Version	Description of Revisions
4/19/10	13.0	Updated to support ISE v12.1. Added additional Spartan-6 FPGA part and package support.
4/19/10	13.1	Removed support for Spartan-6 64-bit LX100/100T devices.
7/23/10	14.0	Updated to support ISE v12.2. Added support for Spartan-6 64-bit LX100/100T devices.
06/22/11	14.1	Updated to support ISE v13.2. Added support for Spartan-6 FGG900, Kintex-7, and Virtex-7 devices.
10/19/11	14.2	Updated to support ISE 13.3 software for core version v4.15. • In Table 1, corrected Kintex-7 FPGA device numbers and added new Kintex-7 devices.
01/18/12	14.3	Updated to support ISE 13.4 software for core version v4.16. • In Table 1, added support for Artix-7 devices.
03/09/12	14.4	Removed XC7A30T and XC7A50T devices.
07/25/12	15.0	Updated to support Vivado 2012.2 and ISE 14.2 Design Suites for core version v4.17.

Notice of Disclaimer