32-Bit Initiator/Target v3 & v4 for PCI

DS206 June 24, 2009 **Product Specification v3.167 & v4.9**

Features

- Fully compliant 32-bit, 66/33 MHz 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 the Xilinx CORE Generator™ software v10.1 with applicable service pack
- 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

Core Facts		
Resource Utilization(1)	**v4 Core**	**v3 Core**
LUTs	506	553
Slice Flip-Flops	333	566
IOB Flip-Flops	270	97
IOBs	55	50
TBUFs(2)	0	288
GCLKs(3)	2	1
Provided with Core		
Documentation	*Getting Started Guide v4* *Getting Started Guide v3* *User Guide v4* *User Guide v3*	
Design File Formats	Verilog/VHDL Simulation Model NGC Netlist (v4 core only) NGO Netlist (v3 core only)	
Constraints Files	User Constraints File (UCF)	
Example Design	Verilog/VHDL Example Design	
Design Tool Requirements		
Xilinx Tools	ISE® v11.2	
Tested Entry and Verification Tools(4) 5	Synplicity™ Synplify Xilinx XST(6) Mentor Graphics® ModelSim® v6.4b and above Cadence® IUS v8.1 -s009 and above	
Support		
Provided by Xilinx @ www.xilinx.com/support		

1. Depends on configuration of the interface and design. Unused resources are trimmed by the Xilinx technology mapper. The utilization figures reported represent a maximum configuration.
2. Virtex®-5, Virtex-4, Spartan®-6, Spartan-3, Spartan-3A, Spartan-3AN, Spartan-3A DSP and Spartan-3E devices do not contain TBUFs. The Xilinx tools automatically translate TBUFs to LUTs and they are included in the worst-case LUT count listed.
3. Virtex-4 and Virtex-5 implementations require additional BUFG for 200 MHz reference clock.
4. See the PCI Getting Started Guide or product release notes for current supported version.
5. Requires a Verilog LRM-IEEE 1364-2005 encryption-compliant simulator. For VHDL simulation, a mixed HDL license is required.
6. XST is a command line option only.

NOTE: Xilinx provides technical support for this LogiCORE IP product when used only as described in the Getting Started and User Guides. Xilinx cannot guarantee timing, functionality, or support of product if implemented in devices not listed, or if it is customized beyond the guidelines provided in the associated product documentation.

Table 1: **Core Implementation**

Supported Devices[1],[2],[3]	Core Version	Signaling Environment
PCI32/66		
Virtex-5 XC5VFX70T-FF1136-2C/I[4] (regional clock)	v4	3.3V only
Virtex-5 XC5VLX50-FF1153-2C/I[4] (regional clock)	v4	3.3V only
Virtex-5 XC5VLX50T-FF1136-2/C/I[4] (regional clock)	v4	3.3V only
Virtex-5 XC5VLX110-FF1153-2C/I[4] (regional clock)	v4	3.3V only
Virtex-5 XC5VLX110T-FF1136-2/C/I[4] (regional clock)	v4	3.3V only
Virtex-5 XC5VSX50T-FF1136-2/C/I[4] (regional clock)	v4	3.3V only
Virtex-5 XC5VSX95T-FF1136-2/C/I[4] (regional clock)	v4	3.3V only
Virtex-4 XC4VFX20-FF672-11C/I[4] (regional clock)	v3	3.3V only
Virtex-4 XC4VLX25-FF688-11C/I[4],[5] (regional clock)	v3	3.3V only
Virtex-4 XC4VSX35-FF668-11C/I[4],[5] (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-3AN XC3S400AN-FGG400-5C	v3	3.3V only
Spartan-3AN XC3S700AN-FGG484-5C	v3	3.3V only
Spartan-3AN XC3S1400AN-FGG676-5C	v3	3.3V only
Spartan-3ADSP XC3SD1800A-FG676-5C	v3	3.3V only
Spartan-3ADSP XC3SD3400A-FG676-5C	v3	3.3V only
Spartan-3E XC3S500E-FT256-5C[5]	v3	3.3V only
Spartan-3E XC3S1200E-FG400-5C[5]	v3	3.3V only
PCI32/33		
Virtex-5 XC5VFX70T-FF1136-1C/I[4] (global clock)	v4	3.3V only
Virtex-5 XC5VFX70T-FF1136-1C/I[4] (regional clock)	v4	3.3V only
Virtex-5 XC5VLX50-FF1153-1C/I[4] (global clock)	v4	3.3V only
Virtex-5 XC5VLX50-FF1153-1C/I[4] (regional clock)	v4	3.3V only
Virtex-5 XC5VLX50T-FF1136-1C/I[4] (global clock)	v4	3.3V only
Virtex-5 XC5VLX110T-FF1136-1C/I[4] (global clock)	v4	3.3V only

Table 1: **Core Implementation *(Continued)***

Supported Devices[1],[2],[3]	Core Version	Signaling Environment
Virtex-5 XC5VLX110T-FF1136-1C/I[4] (regional clock)	v4	3.3V only
Virtex-5 XC5VSX50T-FF1136-1C/I[4] (regional clock)	v4	3.3V only
Virtex-5 XC5VSX95T-FF1136-1C/I[4] (global clock)	v4	3.3V only
Virtex-5 XC5VSX95T-FF1136-1C/I[4] (regional clock)	v4	3.3V only
Virtex-5 XC5VLX110-FF1153-1C/I[4] (global clock)	v4	3.3V only
Virtex-5 XC5VLX110-FF1153-1C/I[4] (regional clock)	v4	3.3V only
Virtex-4 XC4VFX20-FF672-10C/I[4](global clock)	v3	3.3V only
Virtex-4 XC4VFX20-FF672-10C/I[4] (regional clock)	v3	3.3V only
Virtex-4 XC4VLX25-FF668-10C/I[4],[5] (global clock)	v3	3.3V only
Virtex-4 XC4VLX25-FF668-10C/I[4],[5] (regional clock)	v3	3.3V only
Virtex-4 XC4VSX35-FF668-10C/I[4],[5] (global clock)	v3	3.3V only
Virtex-4 XC4VSX35-FF668-10C/I[4],[5] (regional clock)	v3	3.3V only
Spartan-6 XC6SLX45T-FG484-1C/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-3AN XC3S400AN-FGG400-4C/I	v3	3.3V only
Spartan-3AN XC3S700AN-FGG484-4C/I	v3	3.3V only
Spartan-3AN XC3S1400AN-FGG676-4C/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
Spartan-3E XC3S500E-FT256-4C/I[5]	v3	3.3V only
Spartan-3E XC3S1200E-FG400-4C/I[5]	v3	3.3V only
Spartan-3 XC3S1000-FG456-4C/I	v3	3.3V only

1. Virtex-5, Virtex-4, Spartan-6, Spartan-3A, Spartan-3AN, Spartan-3ADSP, Spartan-3E and Spartan-3 devices are supported over commercial and industrial temperature ranges.
2. Packages listed are supported in both standard and lead-free variants, if available. For example, FF1136 denotes support for both FF1136 and FFG1136 packages.
3. For additional part/package combinations in Spartan-3 and older device families, see the UCF Generator located in the PCI Lounge. 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.
4. Virtex-4 and Virtex-5 solutions require a 200 MHz reference clock.
5. Virtex-4 (except FX) and Spartan-3E solutions require silicon stepping 1 or later.

Applications

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

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. This significantly reduces engineering time required to implement the PCI portion of 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, Xilinx products for PCI minimize your product development time.

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

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

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

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

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, as well as floorplanning 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 constraint files guide the implementation tools so that the critical paths always are within specification.

Xilinx provides Smart-IP constraint files for many device and package combinations. Constraint files for unsupported device and package combinations may be generated using the web-based constraint file generator.

Functional Description

Figure 1 illustrates a user application and the PCI Interface partitioned into five major blocks.

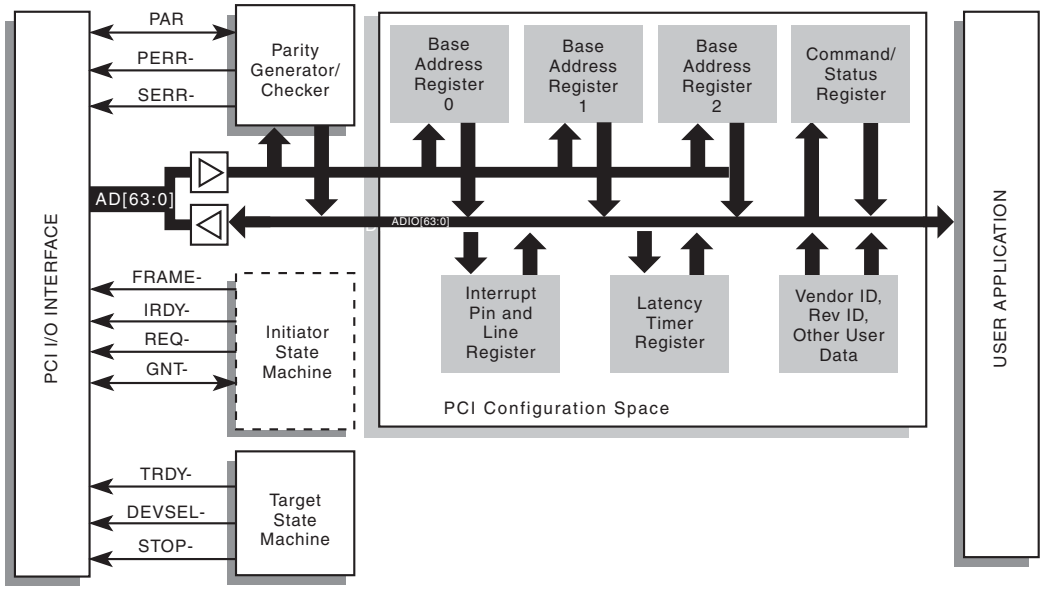

Figure 1: **Initiator/Target core for PCI Block Diagram**

I/O Interface Block

The I/O interface block handles the physical connection to the PCI bus including all signaling, input and output synchronization, output three-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, version 3.0 Configuration Space Header, as shown in Table 2, to support software-driven Plug-and-Play initialization and configuration. This includes information for Command and Status, and three Base Address Registers (BARs).

The capability for extending configuration space has been built into the user application interface. This capability, including 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	16	15	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

Note: 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 via PERR# and address parity errors via 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 core 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 core can be easily configured to fit unique system requirements using the Xilinx CORE Generator GUI or by changing the HDL configuration file. The following 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 core for PCI user guides for more information.

Burst Transfer

The PCI bus derives its performance from its ability to support burst transfers. 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 Commands for PCI

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 both sourcing and receiving data. The core supports a sustained bandwidth of up to 264 MBps, and can be configured to support very long bursts.

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

Recommended Design Experience

The Initiator/Target core for PCI is pre-implemented, allowing engineering focus on the unique user application functions of a design. Regardless, PCI is a high-performance design that is challenging to implement in any technology. Therefore, previous experience with 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 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 your user design is capable of running can be affected by the size and quality of the design. Table 4 lists the Timing Parameters in the 66 MHz Implementations and Table 5 lists Timing Parameters in the 33 MHz Implementations.

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

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

1. Controlled by timespec constraints, included in product.
2. Controlled by SelectIO 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

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

Ordering Information

The Initiator/Target core for PCI is accessed through the Xilinx CORE Generator software v11.2 or higher. The Xilinx CORE Generator software is bundled with the ISE Foundation v11.2 software at no additional charge. To purchase the core, please 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 chronicles the revision history for this document.

Date	Version	Revision
07/30/02	1.2	Style updates
12/18/02	1.3	Updated to build v3.0.103; v5.Ii, 1st feature: 32-bit was 64/32-bit
3/7/03	1.4	Updated to build v3.0.105; v5.2i
4/14/03	1.5	Updated to build v3.0.106; updated PC32/33 product listings to include Spartan-3 device support.
5/8/03	1.6	Updated Xilinx tools to 5.2i SP2; added Note 10.
9/17/03	1.7	Updated to build v3.0.113; Xilinx Tools v6.1i SP1 was v5.2i SP2; date was May 8, 2003.
10/28/03	1.8	Updated to build v3.0.116, in Supported Devices table, added XC prefix to device names.
1/30/04	1.9	Updated to build v3.0.122, updated copyright information to 2004.
4/9/04	1.10	Updated to build v3.0.126; updated Xilinx tools to 6.2i SP1; in supported devices table, added notes 11 and 12; added suffix /I to all Virtex-II Pro devices.
4/26/04	1.11	Updated to build v3.0.128, updated Xilinx tools to 6.2i SP2, changed date to April 26, 2004.
7/15/04	1.12	Updated to build v3.0.129 and to support Xilinx tools v6.2i SP3. The data sheet is updated to the new template.
11/11/04	1.13	Updated support for Xilinx tools v6.3i SP2; updated PCI spec to v3.0; added Exemplar LeonardoSpectrum and Cadence NC-Verilog entry and verification tools.
12/8/04	1.14	Updated to build 3.0.140 and Virtex-4 support.
3/7/05	1.15	Updated to Xilinx tools 7.1i and build v3.0.145.
5/13/2005	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 SP 2 to SP3 for 7.1i
9/12/05	4.0	Updated build to 3.0.152, updated to SP4 for 7.1i, updated release date, removed reference in Table 1 footnote to Spartan-3 as pending, moved placement of Table 3 to immediately follow text reference.
1/18/06	5.0	Updated build to 3.0.155, ISE software to v8.1i, and release date
2/14/06	5.5	Updated build number to 158, added SP 2 support to ISE 8.1i, release date.
7/13/06	6.0	Added v4 core, Virtex-5 support, ISE to 8.21, build number to 160, release date.
2/15/07	6.5	Updated ISE to 9.1i, updated various tool versions, added trademark and registered TM symbols, added support for Spartan-3A,Spartan-3E 66 MHz, and Virtex-5 LXT.
5/17/07	7.0	Corrected usage of PCI terminology to comply with PCI-SIG trademark guidelines. Updated Cadence IUS to v5.7.
8/08/07	7.5	Updated to build 163, minor editorial updates for IP1 Jade Minor release, added Spartan-3A DSP device support.
10/10/07	8.0	Updated trademark usage, release date, added Spartan-3AN device support.
3/24/08	8.5	Updated to support ISE v10.1.
4/25/08	9.0	V4 build 7 PCI core update only for adding Virtex-5 FXT support.

Date	Version	Revision
9/19/08	9.5	Updated to support ISE v10.1 Service Pack 3.
4/24/09	10.1	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	10.5	Updated to support ISE v11.2.

Notice of Disclaimer