XILINX®

Xilinx Inc.
2100 Logic Drive
San Jose, CA 95124
Phone: +1 408-559-7778
Fax: +1 408-377-3259
E-mail: Techsupport: support.xilinx.com
          Feedback: logicore@xilinx.com
URL: http://www.xilinx.com

Introduction

With Xilinx LogiCORE PCI32 Virtex Interface, a designer can build a customized, 32-bit, 0-33 MHz fully PCI compliant system with the highest possible sustained performance (128 Mbytes/sec), and up to 300,000 System Gates in the Virtex FPGA family.

Features

- Fully 2.2 PCI compliant 32-bit, 33 MHz PCI Initiator/Target Interface
- Zero wait-state burst operation
- Programmable single-chip solution with customizable back-end functionality
- Pre-defined implementation for predictable timing in Xilinx Virtex Series FPGAs
- Incorporates Xilinx Smart-IP Technology
- Hot Swap CompactPCI friendly
- Universal PCI support in Virtex
- 3.3 V support Virtex and Virtex-E
- 5 V support in Virtex
- Fully verified design tested with Xilinx testbench and hardware
- Configurable on-chip dual-port FIFOs can be added for maximum burst speed
- Supported Initiator functions
  - Memory Read, Memory Write, Memory Read Multiple (MRM), Memory Read Line (MRL) commands
  - I/O Read, I/O Write commands
  - Configuration Read, Configuration Write commands
  - Bus Parking
  - Special Cycles, Interrupt Acknowledge
  - Basic Host Bridging

LogiCORE™ Facts

Core Specifics	
Device Family	Virtex/Virtex-E
Slices Used[1]	230-280
IOBs Used	53
System Clock f_{max}	0-33MHz
Device Features Used	Multi-standard SelectIO SelectMAP Configuration (optional) Block SelectRAM+™ (optional user FIFO) Boundary scan

Supported Devices[2]/Percent Resources Used		
	I/O	**Slices**
XCV300-5 BG432	17%	7-8%
XCV300E-5 BG432	17%	7-8%

Provided with Core	
Documentation	*PCI Design Guide* *PCI Implementation Guide* *PCI Data Book*
Design File Formats	Verilog/VHDL Simulation Model Verilog/VHDL Instantiation Code NGO Netlist
Constraints File	User Constraints File (UCF)
Verification Tools	Verilog/VHDL Testbench
Reference designs & application notes	Example designs PING Reference Design Synthesizable PCI Bridge(SB08)

Design Tool Requirements	
Xilinx Core Tools	2.1i SP2
Tested Entry/Verification Tools[3]	For CORE instantiation: Synopsys FPGA Express Synopsys FPGA Compiler Synplicity Synplify For CORE verification: Cadence Verilog XL MTI ModelSim PE/Plus

Xilinx provides technical support for this LogiCORE™ product when used as described in the Design and Implementation Guides and in the Application Notes. Xilinx cannot guarantee timing, functionality, or support of product if implemented in devices not listed above, or if customized beyond that referenced in the product documentation, or if any changes are made in sections of design marked as "DO NOT MODIFY".

1. The exact number of Slices depends on user configuration of the core and level of resource sharing with adjacent logic. For example, a factor that can affect the size of the design are the number and size of the BARs.
2. Re-targeting the PCI core to an unlisted device or package will void the guarantee of timing. See "Smart-IP Technology - guaranteed timing" on page 11. for details.
3. See Xilinx Web Site for updates on tested design tools.

Features (cont.)

- Supported Target functions (PCI Master and Slave)
  - Type 0 Configuration Space Header
  - Up to 3 Base Address Registers (memory or I/O with adjustable block size from 16 Bytes to 2 GBytes, medium decode speed)
  - Parity Generation (PAR), Parity Error Detection (PERR# and SERR#)
  - Extended Capabilities Registers (backend module)
  - Memory Read, Memory Write, Memory Read Multiple (MRM), Memory Read Line (MRL), Memory Write Invalidate (MWI) commands
  - I/O Read, I/O Write commands
  - Configuration Read, Configuration Write commands
  - Interrupt Acknowledge
  - 32-bit data transfers, burst transfers with linear address ordering
  - Target Abort, Target Retry, Target Disconnect
  - Full Command/Status Registers
- Available for configuration and download on the web
  - Web-based configuration tool
  - Generation of proven design files
 - Instant access to new releases

Applications

- Embedded applications within telecommunication, networking, and industrial systems
- PCI add-in boards such as graphic cards, video adapters, LAN adapters and data acquisition boards
- Hot Swap CompactPCI boards
  - Other applications that need PCI

General Description

The LogiCORE™ PCI32 Master and Slave Interface has pre-implemented and fully tested modules for the Xilinx Virtex Series FPGAs. The pinout for the device and the relative placement of the internal Configurable Logic Blocks (CLBs) are pre-defined. Critical paths are controlled by TimeSpecs and placement to ensure predictable timing. This significantly reduces engineering time required to implement the PCI portion of your design. Resources can instead be focused on the unique back-end logic in the FPGA and on the system level design. As a result, LogiCORE™ PCI products can minimize your product development time.

Xilinx Virtex Series FPGAs enables designs of fully PCI-compliant systems. The devices meet all required electrical and timing parameters including AC output drive characteristics, input capacitance specifications (10pF), 3 ns setup and 0 ns hold to system clock, and 11 ns system clock to output. These devices meet all specifications for PCI 3.3 V and 5 V.
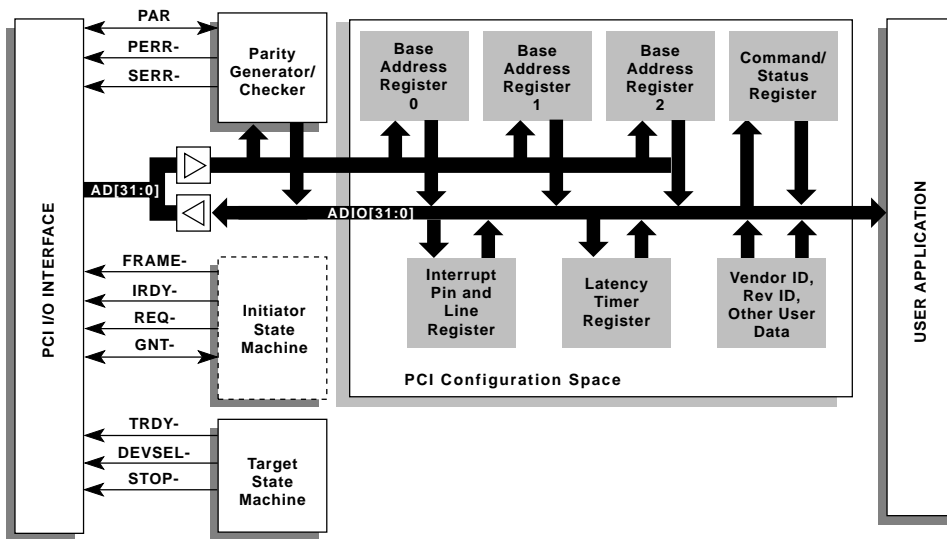
Figure 1: LogiCORE PCI32 Interface Block Diagram

The *PCI Compliance Checklist* has detailed information about electrical compliance. Other features that enable efficient implementation of a complete PCI system in the Virtex Series includes:

- Block SelectRAM+™ memory: Blocks of on-chip ultra-fast RAM with synchronous write and dual-port RAM capabilities. Used in PCI Interfaces to implement FIFO
- Select-RAM™ memory: on-chip ultra-fast RAM with synchronous write option and dual-port RAM option. Used in PCI Interfaces to implement FIFO
- Individual output enable for each I/O
- Internal 3-state bus capability
- 4 global low-skew clock or signal distribution networks
- IEEE 1149.1-compatible boundary scan logic support
- Designed for CompactPCI Hot Swap support

The Master and Slave Interface module is carefully optimized for best possible performance and utilization in the Virtex FPGA architecture. When implemented in a XCV300, 7-8% of the FPGA slices are used.

Smart-IP Technology - Guaranteed Timing

Drawing on the architectural advantages of Xilinx FPGAs, new Xilinx Smart-IP technology ensures highest performance, predictability, repeatability, and flexibility in PCI designs. The Smart-IP technology is incorporated in every LogiCORE PCI core.

Xilinx Smart-IP technology leverages the Xilinx architectural advantages, such as look-up tables (LUTs), distributed RAM, and segmented routing, as well as floorplanning information, such as logic mapping and relative location constraints. This technology provides the best physical layout, predictability, and performance. Additionally, these predetermined features allow for significantly reduced compile times over competing architectures.

PCI cores made with Smart-IP technology are unique by maintaining their performance and predictability regardless of the device size.

To guarantee critical setup, hold, and min. and max. clock-to-out timing, the PCI core is delivered with Smart-IP constraints files that are unique to a device and package combination. These constraints files guide the implementation tools such that the critical paths are always within PCI specification. Retargeting the PCI core to an unsupported device will void the guarantee of timing. Contact a Xilinx XPERTs partner for support of unlisted devices and packages. For contact information, see the XPERTs section in chapter 7 of the Xilinx PCI Data Book .

Universal PCI Support

Since Virtex FPGAs are capable of operating in either 3.3 V or 5 V PCI environments, a designer can easily build universal PCI cards. This requires loading one of the bitstreams at power up. Refer to the *PCI Implementation Guide* and *Building a Universal PCI Card using Xilinx FPGAs Application Note*. Virtex-E devices only support 3.3 V PCI.

Functional Description

The LogiCORE PCI32 Master and Slave Interface is partitioned into five major blocks and a user application as shown in Figure 1. Each block is described below.

PCI Configuration Space

This block provides the first 64 Bytes of Type 0, version 2.1 Configuration Space Header (CSH) (see Table 1) to support software-driven "Plug-and-Play" initialization and configuration. This includes information for Command, Status, and three Base Address Registers (BARs). These BARs illustrate how to implement memory or I/O mapped address spaces.

Table 1: PCI Configuration Space Header

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 (BAR5)				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	Interrupt Pin	Interrupt Line	3Ch
Reserved				40h-FFh

Note:
Italicized address areas are not implemented in the LogiCORE PCI32 Virtex Interface default configuration. These locations return zero during configuration read accesses.

Each BAR sets the base address for the interface and allows the system software to determine the addressable range required by the interface. Each BAR designated as a memory space can be made to represent a 32-bit space.

Using a combination of Configurable Logic Block (CLB) flip-flops for the read/write registers and CLB look-up tables for the read-only registers results in optimized logic mapping and placement.

The LogiCORE PCI32 Interface includes the ability to add extended configuration capabilities as defined in the V2.2 PCI specification. This capability, including the ability to implement a CapPtr in configuration space, allows the user to implement extended functions such as Power Management, Hot Swap CSR, and Message Based Interrupts in the backend design.

PCI 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.

Parity Generator/Checker

This block generates/checks even parity across the AD bus, the CBE lines, and the PAR signal. It also reports data parity errors via PERR- and address parity errors via SERR-.

Target State Machine

This block controls the PCI interface for Target functions. The states implemented are a subset of equations defined in "Appendix B" of the *PCI Local Bus Specification*. The controller is a high-performance state machine using one-hot (state-per-bit) encoding for maximum performance. State-per-bit encoding of the next-state logic functions facilitates a high performance implementation in the Xilinx FPGA architecture.

Initiator State Machine

This block controls the PCI interface for Initiator functions. The states implemented are a subset of equations defined in "Appendix B" of the *PCI Local Bus Specification*. The Initiator Control Logic also uses state-per-bit encoding for maximum performance.

User Application with Optional Burst FIFOs

The LogiCORE PCI32 Interface provides a simple, general-purpose interface with a 32-bit data path and latched address for de-multiplexing the PCI address/data bus. This user interface allows the rest of the device to be used in a wide range of 32-bit applications.

Typically, the user application contains burst FIFOs to increase PCI system performance. An on-chip read/write FIFO, built from the on-chip synchronous dual-port RAM (Block SelectRAM+™) available in Virtex Series FPGAs, supports data transfers in excess of 66 MHz.

Several synthesizable re-usable bridge designs including commonly used backend functions, such as doorbells and mailboxes, are provided with the core.

Interface Configuration

The LogiCORE PCI32 Interface can easily be configured to fit unique system requirements by using Xilinx web-based PCI configuration tool or by changing the Verilog or VHDL configuration file. The following customization options are supported by the LogiCORE product and described in product documentation.

- Initiator or target functionality
- Base Address Register configuration (1 - 3 Registers, size and mode)
- Configuration Space Header ROM
- Initiator and target state machine (e.g., termination conditions, transaction types and request/transaction arbitration)
- Burst functionality
- User Application including FIFO (back-end design)

Supported PCI Commands

Table 2 illustrates PCI bus commands supported by Logi-CORE™ PCI32 Interface. The *PCI Compliance Checklist* has more details on supported and unsupported commands.

Table 2: PCI Bus Commands

CBE [3:0]	Command	PCI Master	PCI Slave
0000	Interrupt Acknowledge	Yes	Yes
0001	Special Cycle	Yes	Ignore
0010	I/O Read	Yes	Yes
0011	I/O Write	Yes	Yes
0100	Reserved	No	Ignore
0101	Reserved	No	Ignore
0110	Memory Read	Yes	Yes
0111	Memory Write	Yes	Yes
1000	Reserved	No	Ignore
1001	Reserved	No	Ignore
1010	Configuration Read	Yes	Yes
1011	Configuration Write	Yes	Yes
1100	Memory Read Multiple	Yes	Yes
1101	Dual Address Cycle	No[1]	Ignore
1110	Memory Read Line	Yes	Yes
1111	Memory Write Invalidate	No[1]	Yes

Note: The initiator can present these commands, however, they either require additional user-application logic to support them or have not been thoroughly tested.

Timing Specification

The Virtex Series FPGA devices, together with the Logi-CORE PCI32 product enable design of fully compliant PCI systems. The maximum speed at which your back-end is capable of running can be affected by the size of the design as well as by the loading of the hot signals coming directly from the PCI bus. Table 3 shows the key timing parameters for the LogiCORE PCI32 Interface that must be met for full PCI compliance.

Table 3: 33 MHz PCI32 Transfer Rates

Parameter	Ref.	PCI Spec.		LogiCORE PCI32 XCV300-5	
		Min	Max	Min	Max
CLK Cycle Time	T_{CYC}	30	∞	30^1	∞
CLK High Time	T_{HIGH}	11		11	
CLK Low Time	T_{LOW}	11		11	
CLK to Bus Signals Valid[3]	T_{ICKOF}	2	11	2^2	11^1
CLK to REQ# Valid[3]	T_{ICKOF}	2	12	2^2	12^1
Tri-state to Active	T_{ON}	2		2^2	
CLK to Tri-state	T_{OFF}		28		28^1
Bus Signal Setup to CLK	T_{PSD}		7		7^1
GNT# Setup to CLK	T_{PSD}		10		10^1
Input Hold Time After CLK	T_{PHD}		0		0^2
RST# to Tri-state	$T_{RST-OFF}$		40		40^1

Notes:
1. Controlled by TIMESPECS, included in product
2. Verified by silicon characterization

Bandwidth

Xilinx LogiCORE PCI32 Interface supports fully compliant zero wait-state burst operations for both sourcing and receiving data. This Interface supports a sustained bandwidth of up to 128 MBytes/sec. The design can be configured to take advantage of the ability of the LogiCORE PCI32 Interface to do very long bursts. Since the FIFO is not of fixed size, bursts can go on for as long as the chipset arbiter will allow. Furthermore, since the FIFOs and DMA are decoupled from the proven core, a designer can modify these functions without affecting the critical PCI timing.

The flexible Xilinx backend, combined with support for many different PCI features, gives users a solution that lends itself to being used in many high-performance applications. Xilinx is able to support different depths of FIFOs as well as dual port FIFOs, synchronous or asynchronous FIFOs and multiple FIFOs. The user is not locked into one DMA engine, hence, a DMA that fits a specific application can be designed.

The theoretical maximum bandwidth of a 32-bit, 33 MHz PCI bus is 128 MBytes/sec. Attaining this maximum bandwidth will depend on several factors, including the PCI design used, PCI chipset, the processor's ability to keep up with your data stream, the maximum capability of your PCI design, and other traffic on the PCI bus. Older chipsets and processors will tend to allow less bandwidth than newer ones.

No additional wait-states are inserted in response to a wait-state from another agent on the bus. Either IRDY or TRDY is kept asserted until the current data phase ends, as required by the V2.2 PCI Specification. See Table 4 for PCI bus transfer rates for various operations.

Table 4: LogiCORE PCI32 Transfer Rates

Zero Wait-State Mode	
Operation	Transfer Rate
Initiator Write (PCI←LogiCORE)	3-1-1-1
Initiator Read (PC →LogiCORE)	4-1-1-1
Target Write (PCI→LogiCORE)	5-1-1-1
Target Read (PCI←LogiCORE)	6-1-1-1

Synthesizable PCI Bridge Design (SB08)

The synthesizable PCI bridge design, SB08, is an application bridge for use with the LogiCORE PCI32 Interface. It is delivered in Verilog and VHDL and has been fully tested with various devices. This example demonstrates how to interface to the PCI core and provide a modular foundation upon which to base other designs. The reference design can be easily modified to remove select portions of functionality.

This design is a general purpose data transfer engine to be used with the LogiCORE PCI32 Interface. Figure 1 presents a block diagram of the synthesizable PCI bridge design. Typically, the user will customize the local interface to conform to a particular peripheral bus (ISA, VME, i960) or attach to a memory device. The design is modular so that unused portions may be removed. The *Synthesizable*

PCI Bridge Design Data Sheet lists the set of features and specifics for the SB08 design.

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. A FIFO to support PCI burst transfer can efficiently be implemented using the Virtex on-chip RAM features, both Distributed SelectRAM and Block SelectRAM+™.

Each Virtex CLB supports four 16x1 RAM blocks. This corresponds to 64 bits of single-ported RAM or 32 bits of dual-ported RAM, with simultaneous read/write capability.

Each Virtex device has two columns of Block RAM. The V300 device has 65,536 bits of Block SelectRAM+™ that can be used to create deep, dual-ported FIFOs.
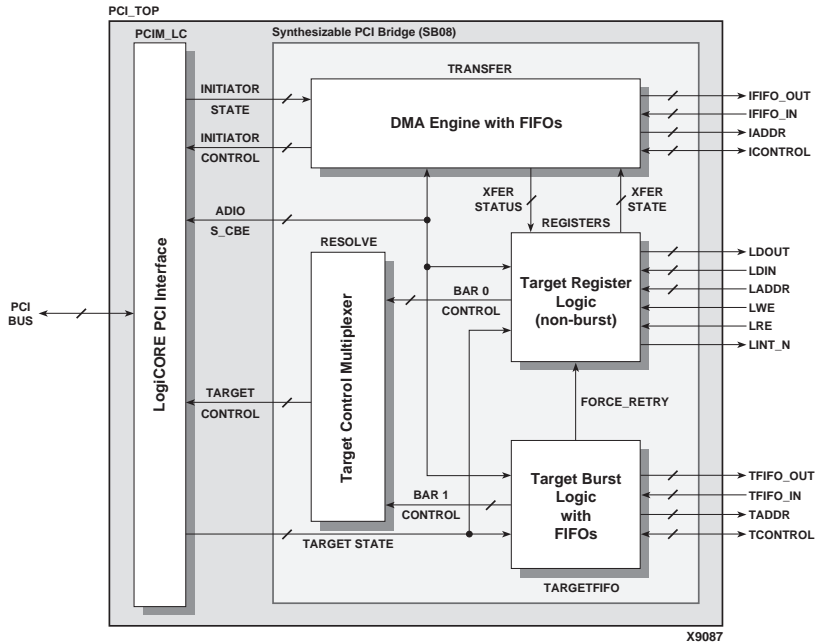

Figure 2: Block Diagram of Synthesizable Bridge Design for PCI32 LogiCORE Interface

Verification Methods

Xilinx has developed a system-level testbench that allows simulation of an open PCI environment in which a Logi-CORE-PCI-based design may be tested by itself or with other simulatable PCI agents. Included in these agents are a behavioral host and target, and several "plug-in" modules, including a PCI signal recorder and a PCI protocol monitor. Using these tools, the PCI developers can write microcode-style test scripts that can be used to verify different bus-operation scenarios, including those in the PCI Compliance Checklist.

The Xilinx PCI testbench is a powerful verification tool that is also used as the basis for verification of the PCI Logi-CORE. The PCI LogiCORE is also tested in hardware for electrical, functional, and timing compliance.

Ping Reference Design

The Xilinx "PING" Application Example, delivered in Verilog and VHDL, has been developed to provide an easy-to-understand example which demonstrates many of the principles and techniques required to successfully use a LogiCORE PCI32 Interface in a System-on-a-Chip solution. The PING design is also used as a test vehicle for PCI core verification.

Device Utilization

Target/Initiator options require a variable amount of CLB resources for the PCI32 Interface

Utilization of the device can vary slightly, depending on the configuration choices made by the designer. Factors that can affect the size of the core are:

- Number of Base Address Registers used. Turning off any unused BARs will save resources.
- Size of the BARs. Setting the BAR to a smaller size requires more flip-flops. A smaller address space requires more flip-flops to decode.
- Latency timer. Disabling the latency timer will save resources. It must be enabled for bursting.

Recommended Design Experience

The LogiCORE PCI32 Interface is pre-implemented allowing engineering focus at the unique back-end functions of a PCI design. Regardless, PCI is a high-performance system that is challenging to implement in any technology, ASIC or FPGA. Therefore, previous experience with building high-performance, pipelined FPGA designs using Xilinx implementation software, TIMESPECs, and guide files is recommended. The challenge to implement a complete PCI design including back-end 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.