OpenCores

PCI Bridge White Paper

6/29/2001

Introduction

Unlike other PCI IP cores on the market today, this IP core is not just simply a PCI interface, which leaves you with a lot of backend implementation issues still not addressed properly. This is a real bridge, with a standardized backend, which only needs to be connected to other IP cores on WISHBONE bus and functions immediately without any additional efforts by HDL designers. When bridge is implemented into the design, it becomes software configurable. System designer in conjunction with software designer can achieve fast operation by knowing just a few of the most important PCI operations. But even that is not a must.

Properly set parameters of the core before synthesis (like FIFO depths, number of images etc.), allow for choice between highly configurable and high throughput bridge, low cost bridge (in terms of technology specific device utilization) or a tradeoff between those two extremes.

Features

The following lists the main features of PCI Bridge IP core:

- Independent clock domains for PCI and WISHBONE side of the bridge
- Two possible parameterized implementations provided – HOST and GUEST. HOST implementation is used for host bridging with WISHBONE SoC bus as host bus. GUEST implementation can be used for expansion bus bridging with WISHBONE SoC bus as expansion bus.
- WISHBONE SoC bus revision B.1 compliant with separate Master and Slave interfaces
- PCI 2.2 compliant 32 bit, 66 MHz Initiator and Target interface
- Zero Wait state burst operation
- Four synthesizable, dual port FIFOs with parameterized depth
- Parameterized number of synthesizable, fully programmable images (default one, maximum 5 images) with address translation capability and image size of 4KB to 1GB.
- Programmable image address space mapping (I/O or Memory space)
- PCI transaction ordering requirements met within the core. Bridge uses posted writes and delayed reads in either direction
- Single delayed transaction support in either direction
- Extended configuration space implemented for additional software programmable features of the core.
- Fully transparent PCI bus command usage controllable on image by image basis with proper settings of configuration registers
- Supported Initiator functions:
- Memory Read, Memory Read Line, Memory Read Multiple, Memory Write commands
- IO Read and Write commands
- Configuration Read and Write commands
- Interrupt acknowledge command
- Linear burst ordering supported
- Fully transparent WISHBONE interface operation controllable on image by image basis with proper software settings of configuration registers
- Supported Target functions
- Memory Read, Memory Read Line, Memory Read Multiple, Memory Write, Memory Write and Invalidate commands
- IO Read and Write commands
- Configuration Read and Write commands
- Linear burst ordering supported

Architecture

Figure shows general architecture of PCI Bridge IP core. It consists of several building blocks:

- WISHBONE slave module
- WISHBONE Write FIFO (WBW_FIFO)
- WISHBONE Read FIFO (WBR_FIFO)
- PCI MASTER module
- PCI I/O
- PCI Target module
- PCI Write FIFO (PCIW_FIFO)
- PCI Read FIFO (PCIR_FIFO)
- WISHBONE master module
- Configuration space

Contact: Miha Dolenc (mihad@opencores.org)
Tadej Markovic (tadej@opencores.org)

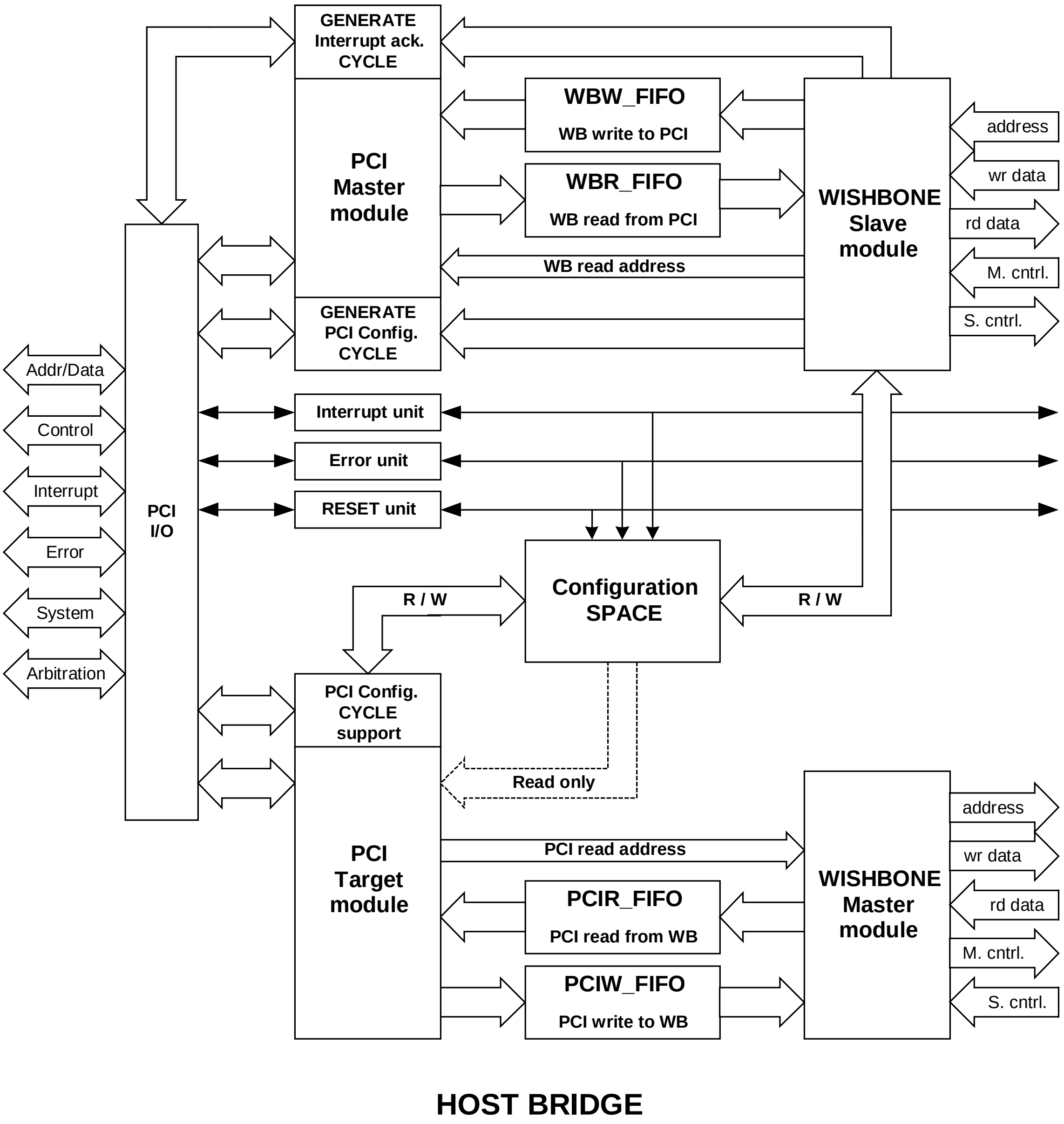

Figure 1: Core's Architecture

Bridge allows only one outstanding delayed transaction request at a time in each direction.

WISHBONE Slave Module

Slave interface is WISHBONE Rev. B.1 compliant. It accepts accesses that fall into one of implemented image's regions and responds according to that image's configuration. Image's configuration can be software programmed to perform address translation and/or use memory access optimizing commands when burst transfer is requested. Writes are handled as posted and reads as delayed. Configuration space is also accessible through WISHBONE slave interface with read/write capability for HOST and read only for GUEST bridges. Configuration cycle generation is performed as delayed transaction regardless of whether it is a read or write request. Interrupt acknowledge cycle is processed as delayed read.

WISHBONE Write FIFO

is used for storing address and data information coming from external WISHBONE master performing a write to the address that falls within one of the image's address space. On the other side, PCI master state machine takes data out and completes writes on PCI bus. Depth is parameterized.

WISHBONE Read FIFO

Data from delayed reads, which already completed on PCI bus, is stored in this FIFO. PCI master state machine performs a read when requested, stores data in the FIFO and data is provided for external WISHBONE master when it repeats the request

Contact: Miha Dolenc (mihad@opencores.org)
Tadej Markovic (tadej@opencores.org)

through WISHBONE slave interface. FIFO depth is parameterized.

PCI MASTER Module

PCI master module is PCI 2.2 compliant initiator interface. It is responsible for completing posted writes stored in WBW_FIFO or service delayed read, configuration read/write or interrupt acknowledge cycle requests. Bus command used on PCI bus depends on several factors:

- In an image access it depends on address space mapping of an image (IO or memory), whether it is a burst transfer and by configuration of an image (if it has memory access optimizing commands enabled).
- Other requests use special dedicated commands according to PCI specification.

PCI Target Module

Target state machine accepts cycles that fall within one of image's address space and configuration cycles from PCI bus. Configuration cycle accesses provide access only to Type0 predefined header portion of configuration space. Extended configuration space is accessible with memory read and write accesses through special 4KB memory mapped image. Read only access to configuration space is allowed for HOST and read/write is allowed for GUEST bridges. All but configuration accesses are passed through the bridge to WISHBONE master interface. Each implemented image can be configured to perform address translation and/or configure WISHBONE master state machine to perform burst transfers even though PCI Initiator requesting a transfer does not use memory access optimizing command.

PCI Write FIFO

is used for storing address and data information coming from external PCI initiator performing a write command to the address that falls within one of the image's address space. On the other side, WISHBONE master state machine takes data out and completes writes on WISHBONE bus. Depth is parameterized.

PCI Read FIFO

Data from delayed reads that already completed on WISHBONE bus is stored in this FIFO. WISHBONE master state machine performs a read when requested, stores data in the FIFO and data is provided for external PCI initiator when it repeats the request through PCI Target Module. FIFO depth is parameterized.

WISHBONE Master Module

responds to requests received from PCI bus and passed through the bridge. When memory-optimizing commands are used on PCI bus, WISHBONE master interface can boost performance by pre-fetching large bursts for reads and storing them in a read FIFO. Burst writes accepted from PCI bus and stored in write FIFO are performed as burst writes on WISHBONE bus also.

Configuration Space

4KB of Configuration Space is provided for software controllable features of the bridge. Besides Type0 predefined header demanded by PCI specification, there are registers provided for image's control, address translation, interrupt control and status, error reporting etc. Read/Write access is provided for WISHBONE bus in HOST implementation and to PCI bus for GUEST implementation. Opposite side has read only access to Configuration Space.

Clocks

PCI and WISHBONE clocks are independent of each other and have no special synchronization demands. PCI clock is specified in PCI specification to be 0-66 MHz, while WISHBONE clock frequency is at the designers will.

Interrupts

Bridge can trigger interrupts on certain conditions, such as Parity error, Target Abort during posted write or similar. Configuration Space provides a mechanism for enabling this interrupts and interrupt status register for reporting them. Bridge also routes and reports interrupts triggered on busses. HOST bridge implementation triggers interrupts (if enabled) to WISHBONE bus, while GUEST implementation triggers them on PCI bus.

Contact: Miha Dolenc (mihad@opencores.org)
Tadej Markovic (tadej@opencores.org)