PCI32tLite

PCI Target IP Core

Developer's Manual
User's Manual

1 Introduction

The PCI32tLite IP core provides the funtionality of a PCI target. The core has been designed to permit interface between a PCI Master and simple WHISBONE Slaves, and fitting into smallest FPGA (about 200 LC's in ALTERA CYCLONE II FPGA).

Thank you for your interest in the "pci32tlite_oc" IP Core. I'll be grateful with your feedbacks, good or bad. They will help to improve the IP Core.

Document History

Revisión	Descripción	Autor	Fecha
R00B00	First Beta release PRELIMINARY.	Peio Azkarate	20050512
R00	First release	Peio Azkarate	20060717
R01	• Added information for the R02 of the IPCore • Added "pcmip_dbg" example project.	Peio Azkarate	20070919
R01B00	• Added information for the R02 of the IPCore • Added "pcmip_dbg" example project.	Peio Azkarate	20070919
R02	• Added information for the R03 of the IPCore • Changed BURST operation • Added PCI32TLITE_HOWTO projects reference • Removed "pcmipdbg_uart" project reference • Change on PCI to WB data bus translation for BIG/16 (due to a bug fix).	Peio Azkarate	20080616
R03	• Remove PCI32TLITE_HOWTO information.	Peio Azkarate	2022-0119

2 Features

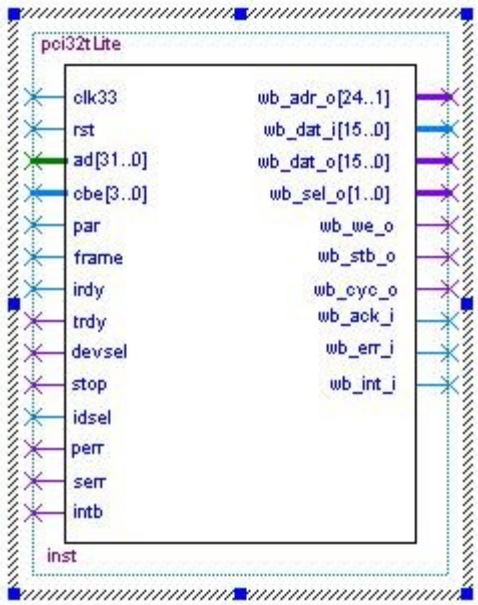
FEATURES

- WHISBONE compatible.

- 32 bits TARGET PCI Bus interface
- MASTER WHISBONE; with configurable data bus size, 32/16/8 bits and configurable endianness, "BIG"/"LITTLE".
- PCI Bus is Little endian.
- Configuration Space Registers are Little endian.
- PCI Configuration Space Register transactions could be BYTE/WORD/DWORD.
- Supported BURST transactions.
- BAR0 register, occupies 32Mbytes on PCI memory map for "1BARMEM" configuration.
- BAR0 register, occupies 512Bytes on PCI I/O map for "1BARIO" configuration.
- Supported commands are:
  - Memory Read
  - Memory Write
  - Configuration Read
  - Configuration Write
- Target initiated Termination (RETRY)
- FPGA Proven.

3 Description

INTERCONECTION



IP Interface Signals

Configuration

PCI32TLITE - Configuration Generics				
Name	**Type**		**Polarity**	**Description**
vendorID	generic	std_logic_v		PCI Vendor ID
deviceID	generic	std_logic_v		PCI Device ID
revisionID	generic	std_logic_v		PCI Revision ID
subsystemID	generic	std_logic_v		PCI Subsystem ID
subsystemvID	generic	std_logic_v		PCI Subsystem Vendor ID
BARS	generic	string		"1BARMEM"/"1BARIO"
WBSIZE	generic	integer		Bus Whisbone SIZE: 32/16/8
WBENDIAN	generic	string		Bus Whisbone Endianess: "BIG"/"LITTLE"

Control

PCI32TLITE - Control signals			
Name	**Type**	**Polarity**	**Description**
rst	input	High	Reset. Active High, ¡ be careful ! the PCI "rst" signal is active Low.

PCI Target

PCI32TLITE - PCI target signals			
Name	**Type**	**Polarity**	**Description**
clk33	input		PCI Clock 33Mhz
ad[31..0]	bidir		PCI Address/Data
cbe[3..0]	input	Low	PCI Command/Byte enable
par	bidir	Low	PCI parity. numero de 1's en ad[31..0], cbe[3..0] y par debe ser un numero "par".

frame	input	Low	PCI frame. Indicate a bus transaction begin. When deasserted, the transaction is in the final data phase or has completed.
irdy	input	Low	Initiator ready. Indicates the bus master ability to complete the current data phase of the transaction.
trdy	output	Low	Target ready. Indicates the target ablility to complete the current data phase of the transaction.
devsel	output	Low	Device select. Target indicates to master that it has decoded its address.
stop	output	Low	The target is requesting the master to stop the current transaction.
idsel	input	Low	Initialization Device Select. Chip select for configuration transactions.
perr	output	Low	PCI Parity error
serr	output	Low	PCI System error
intb	output	Low	PCI Interrupt B

Whisbone Master

PCI32TLITE - Whisbone master signals			
Name	**Type**	**Polarity**	**Description**
wb_adr_o[24..0]	output		Whisbone address
wb_dat_i[(WBSIZE1)..0]	input		Whisbone data in
wb_dat_o[(WBSIZE1)..0]	output		Whisbone data out
wb_sel_o[((WBSIZE/8)1)..0]	output	High	Whisbone data byte selection
wb_we_o	output	High	Whisbone write enable
wb_stb_o	output	High	Whisbone data strobe
wb_cyc_o	output	High	Whisbone cycle
wb_ack_i	input	High	Whisbone acknoledge
wb_rty_i	input	High	Whisbone retry
wb_err_i	input	High	Whisbone error
wb_int_i	input	High	Whisbone interrupt

Note: The reset signal "rst" is not a PCI signal. "rts" is active high while the PCI RST# is active low.

COMPONENTS

Fichero	Descripción
pci32tlite.vhd	TOP structural
pcidec.vhd	Decoder
pciwbsequ.vhd	PCI to WB sequencer
pcidmux.vhd	Data mux
pciregs.vhd	Configuration Space Registers
pcipargen.vhd	Parity generation
onalib.vhd	Small common functions.

PCIDEC

PCI decoder.

Registering address and command signals. Decode access to Configuration Space Register and to the PCI memory map.

PCIWBSEQU

PCI to WHISBONE sequencer. Translates transactions fron PCI to WHISBONE. Generate control signals for others blocks.

PCIDMUX

Multiplex WHISBONE and Configuration Space Registers data bus.
Registering WHISBONE data bus.
Transalates the 32 bits PCI data bus to 8/16/32 bits WHISBONE data bus.
PCI Little endian WB byte swapping when Big endian.

PCIREGS

PCI registers. Configuration Space registers.

PCIPARGEN

Target parity generation on read transactions.

4 Implementation

PCI-WHISBONE Address map translation

PCI (AD24..AD2) address are translated to WHISBONE wb_adr(24..2). The WHISBONE address wb_adr(1) and wb_adr(0) are a logical combination from PCI CBE signals.

PCI-WHISBONE Big Endian-16 Bits

The PCI data bus size is 32 bits and the WHISBONE data bus is configurated as 16 bits.
The PCI bus is “Little endian” and WHISBONE is configured as “Big endian”.
Configuration Space Registers are “Little endian” and BYTE/WORD/DWORD accesses are possible.
Access from PCI to WHISBONE memory map (BAR0 32MBytes) should be done in BYTE/WORD.

The PCI to WHISBONE translation is "Address invariant translation". Address invaration mapping preserves the byte ordering of a data structure in a little endian memory map an a big endian memory map.

PCI				WHISBONE				Byte lane	
CBE(3)	CBE(2)	CBE(1)	CBE(0)	A1	A0	SEL(1)	SEL(0)	PCI	WB
0	0	0	0	0	0	1	1	D(15..0)	D(15..0)
1	1	0	0	0	0	1	1	D(15..0)	D(15..0)
0	0	1	1	1	0	1	1	D(31..16)	D(15..0)
1	1	1	0	0	0	1	0	D(7..0)	D(15..8)
1	1	0	1	0	1	0	1	D(15..8)	D(7..0)
1	0	1	1	1	0	1	0	D(23..16)	D(15..8)
0	1	1	1	1	1	0	1	D(31..24)	D(7..0)

PCI-WHISBONE LITTLE ENDIAN-32 Bits Byte lane

PCI				WHISBONE						Byte lane	
CBE(3)	CBE(2)	CBE(1)	CBE(0)	A1	A0	SEL(3..0)				PCI	WB
0	0	0	0	X	X	1	1	1	1	D(31..0) D(23..16)	D(31..0) D(7..0)
1	1	0	0	X	X	0	0	1	1	D(15..8) D(7..0)	D(15..8) D(7..0)
0	0	1	1	X	X	1	1	0	0	D(31..24) D(23..16)	D(15..8) D(7..0)
1	1	1	0	X	X	0	0	0	1	D(7..0)	D(7..0)
1	1	0	1	X	X	0	0	1	0	D(15..8)	D(15..8)

1	0	1	1	X	X	0	1	0	0	D(23..16) D(7..0)
0	1	1	1	X	X	1	0	0	0	D(31..24) D(15..8)

PCI-WHISBONE LITTLE ENDIAN-8 Bits Byte lane

PCI				WHISBONE			Byte lane	
CBE(3)	CBE(2)	CBE(1)	CBE(0)	A1	A0	SEL(0)	**PCI**	**WB**
0	0	0	0	0	0	1	AD(7..0)	D(7..0)
1	1	0	0	0	0	1	AD(7..0)	D(7..0)
0	0	1	1	1	0	1	AD(23..16)	D(7..0)
1	1	1	0	0	0	1	AD(7..0)	D(7..0)
1	1	0	1	0	1	1	AD(15..8)	D(7..0)
1	0	1	1	1	0	1	AD(23..16)	D(7..0)
0	1	1	1	1	1	1	AD(31..24)	D(7..0)

Whisbone Configuration

The Whisbone data bus size and endianess can be configurated through WBSIZE and WBENDIAN VHDL generics. Allowed combination values for these generics are:

- WBSIZE=32 WBENDIAN="LITTLE"
- WBSIZE=16 WBENDIAN="BIG"
- WBSIZE=8 WBENDIAN="LITTLE"

Whisbone RETRY

When a PCI transaction is terminated (on the WB bus side) with "wb_rty" instead a normal "wb_ack" termination. On PCI bus side, the STOP# signal is activated and produce a Target disconect termination.

BARS Configuration

BARS generic is available in order to configure BAR0 decoding response. "1BARMEM" option configures BAR0 for use 32MBytes on PCI Memory map. "1BARIO" option configures BAR0 for use 512Bytes on PCI IO map.

PCI BURST

PCI BURST transactions are terminated by target with "termination with..." so burst transactions are broken into individual transactions and they aren't traslated to whisbone bus as burts. This is a transparent feature for the whisbone slaves, and it's responsability to the PCI Compatible Master Controller/Bridge convert burst into individual cicles.

CONFIGURATION SPACE REGISTERS

The Configuration Registers from **0x00** to **0x3C** conform with the PCI Device Configuration Header Format.

pci32tLite - PCI Configuration Space Registers layout

Address	Byte			
	3	**2**	**1**	**0**
0x00	Device ID (generic)		Vendor ID (generic)	
0x04	Status Register		Command Register	
0x08	Class Code (0x068000)			Revision ID (generic)
0x0C	BIST	Header Type	Latency Timer	Cache Line Size
0x10	BAR0			
0x14	This register is always 0.			
0x18	This register is always 0.			
0x1C	This register is always 0.			
0x20	This register is always 0			
0x24	This register is always 0			
0x28	Card Bus CIS Pointer			
0x2C	Subsystem ID (generic)		Subsystem Vendor ID (generic)	
0x30	Expansion ROM Base Address Register			
0x34	Reserved			
0x38	Reserved			
0x3C	Maximum Latency	Minimum Grant	(0x01 INTA)	Interrupt Line

Register	Addr	Offset	Byte Enable	Size
VENDORID	000000	00	0/1 (r)	WORD
DEVICEID	000000	02	2/3 (r)	WORD
CMD	000001	04	0/1 (r/w)	WORD
ST	000001	06	2/3 (r/w*)	WORD
REVISIONID	000010	08	0 (r)	BYTE
CLASSCODE	000010	09	1/2/3 (r)	3 BYTES
HEADERTYPE	000011	0E	2 (r)	BYTE
BAR0	000100	10	0/1/2/3 (r/w)	DWORD
INTLINE	001111	3C	0 (r/w)	BYTE
INTPIN	001111	3D	1 (r/w)	BYTE
(W) reseteable*				

VENDORID Vendor ID Register (r)

Identifies manufacturer of device. To configure by "generic".

DEVICEID Device ID Register (r)

Identifies the device. To configure by "generic".

CMD Command Register (r/w)

15	14	13	12	11	10	9	8
x	x	x	x	x	x	x	SERRENb

7	6	5	4	3	2	1	0
x	PERRENb	x	x	x	x	MEMSPENb	x

Reset value = 0x0000
SERRENb : System ERRor ENable (1 = Enable, 0 = Disable)
PERRENb : Parity ERRor ENable (1 = Enable, 0 = Disable)
MEMSPENb : MEMmory SPace ENable (1 = Enable, 0 = Disable)

ST STatus register (r/w*)

15	14	13	12	11	10	9	8
PERRDTb	SERRSIb	x	x	TABORTSIb	DEVSELTIMb[1..0]		x

7	6	5	4	3	2	1	0
x	x	x	x	x	x	x	x

Reset value = 0x0200
PERRDTb : Parity ERRor DeTected (1 = Error Detected)
PERRSIb : System ERRor Signaled (1 = Error Signaled)
TABORTSIb : Target ABORT Signaled (1 = Abort Signaled)
DEVSELTIMb : DEVSEL Timing (01 = Medium)

Reads to this register behave normally. Writes are different, that bits can be cleared, but not set. Write a 1 to clear one bit.

REVISIONID Revision ID Register (r)

Identifies a device revision.

CLASSCODEID CLASS CODE Register (r)

Identifies the generic function of the device.

HEADERTYPE HEADER TYPE Register (r)

Identifies the layout of the second part of the predefined header.

BAR0 Base Address 0 Register (r/w)

31	30	29	28	27	26	25	24
BAR032MBb[6..0]							0

23	22	21	20	19	18	17	16
0x00							

15	14	13	12	11	10	9	8
0x00							

7	6	5	4	3	2	1	0
0x00							

Reset value = 0x00
BAR032MBb : Base Address 32MBytes decode space (7 bits)

SUBSYSTEMID Subsystem ID Register (r)

Identifies the addin board or subsystem where th PCI device resides. To configure by "generic".

SUBSYSTEM VENDORID Subsystem Vendor ID Register (r)

Identifies the vendor of the addin board or subsystem where th PCI device resides. To configure by "generic".

INTLINE Interrupt Line Register (r/w)

7	6	5	4	3	2	1	0
INTLINEr[7..0]							

Reset value = 0x00
INTLINEr : Defines interrupt line routing

INTPIN Interrupt Pin Register (r)

7	6	5	4	3	2	1	0
0x01							

Reset value = 0x01
INTPINr : Tells which interrupt pin the device uses. 0x01=INTA

5 FPGA proven

The pci32tLite_oc IP core has been tested in hardware, several project have been implemented in FPGA.

6 LICENSE