Tampere University of Technology

Faculty of Computing and Electrical Engineering

Department of Computer Systems

HIBI_PE_DMA HW Reference Manual

Author:
Lasse Lehtonen

Updated:
March 2, 2012

Contents

1 REVISION HISTORY

Table 1

Revision	Author	Date	Description
1.03	Lasse Lehtonen	26.01.2012	Ported old document to Latex
1.04	Lasse Lehtonen	27.01.2012	Added streaming channels
1.05	Lasse Lehtonen	02.03.2012	Removed SW part and changed register order

2 DOCUMENT OVERVIEW

2.1 SCOPE

This documentation describes how to use HIBI PE DMA component.

2.2 AUDIENCE

For hardware integrators and software developers using this component.

2.3 RELATED DOCUMENTATION

Table 2

Document	Description
building_test_system.pptx	Descibes an example using in SOPC with NIOS II processors
hpd_sw_ref.pdf	Software reference manual
HIBI_PE_DMA.pptx	Introduction to HIBI PE DMA

2.4 DOCUMENT CONVENTIONS

- Ports: `teletype` in text
- Generics: `teletype` in text

3 INTRODUCTION

3.1 BRIEF DESCRIPTION

HIBI_PE_DMA (HPD) component allows separate processor systems with Avalon-MM compatible interface to communicate with each other via Hibi hierarchical bus. Communication is DMA based and uses either packet or stream channels for network transactions. HIBI_PE_DMA supports both polling and interrupts.

3.2 EXAMPLE SYSTEM

Example system showing three SOPC blocks with Nios processors communicating via Hibi. Each HIBI_PE_DMA component is associated with dual-port RAMs where they store received data and read the data to be sent.

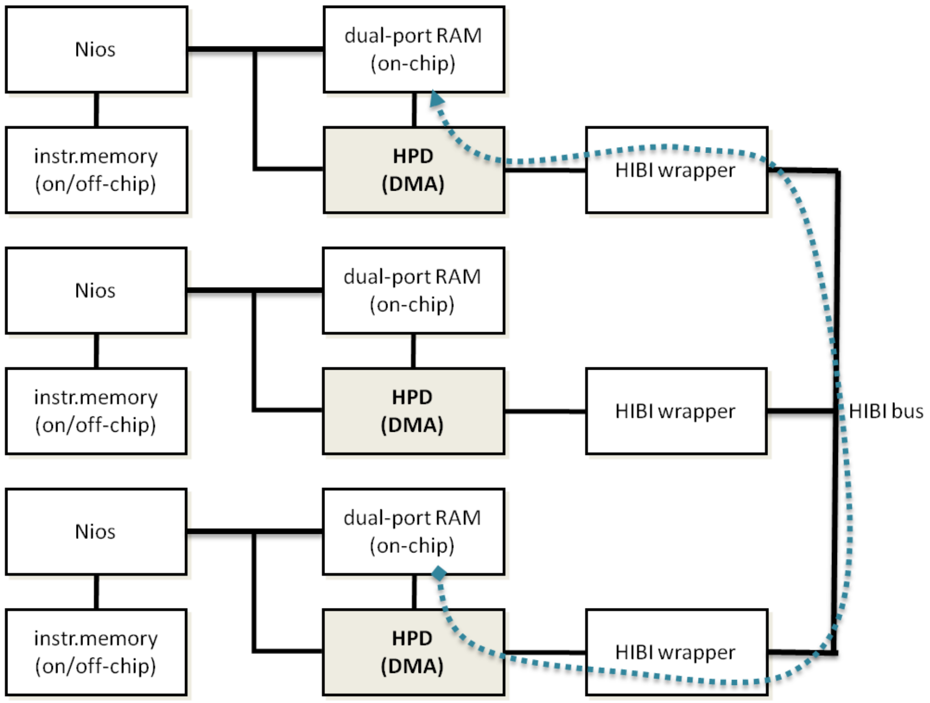

Figure 1: Example system with three Nios processors

4 HARDWARE DESIGN

4.1 HIBI_PE_DMA

4.1.1 GENERICS

Table 3

Name	Description
data_width_g	Width of the data buses
addr_width_g	Width of the Hibi address
words_width_g	Number of bits needed to represent word counters
n_stream_chans_g	Number of stream channels
n_packet_chans_g	Number of packet channels
n_chans_bits_g	Bits needed to represent n_stream_chans_g + n_packet_chans_g
hibi_addr_cmp_lo_g	Lower boundary for address comparison
hibi_addr_cmp_hi_g	Upper boundary for address comparison

4.1.2 CLOCKING AND RESET

Table 4

Port	Width	Direction	Description
clk	1	in	Clock, active on rising edge
rst_n	1	in	Reset, asynchronous, active low

4.1.3 CONFIGURATION INTERFACE

This slave interface is used by the processor for reading or writing HIBI_PE_DMA registers.

Table 5

Port	Width	Direction	Description
avalon_cfg_addr_in	n_chans_g+4	in	Register address
avalon_cfg_writedata_in	addr_width_g	in	Data in
avalon_cfg_we_in	1	in	Write enable
avalon_cfg_readdata_out	addr_width_g	out	Data out
avalon_cfg_re_in	1	in	Read enable
avalon_cfg_cs_in	1	in	Chip select
avalon_cfg_waitrequest_out	1	out	Stall

4.1.4 MEMORY WRITE INTERFACE

This master interface is used by HIBI_PE_DMA to write received data to the shared dual-port memory.

Table 6

Port	Width	Direction	Description
avalon_addr_out_rx	addr_width_g	out	Write address
avalon_we_out_rx	1	out	Write enable
avalon_be_out_rx	data_width_g / 8	out	Byte enable
avalon_writedata_out_rx	data_width_g	out	Data out
avalon_waitrequest_in_rx	1	in	Stall

4.1.5 MEMORY READ INTERFACE

This master interface is used by HIBI_PE_DMA to read data to be sent from the shared dual-port memory.

Table 7

Port	Width	Direction	Description
avalon_addr_out_tx	addr_width_g	out	Read address
avalon_re_out_tx	1	out	Read enable
avalon_readdata_in_tx	data_width_g	in	Data in
avalon_waitrequest_in_tx	1	in	Stall
avalon_readdatavalid_in_tx	1	out	Byte enable

4.1.6 HIBI RECEIVE INTERFACE

Table 8

Port	Width	Direction	Description
hibi_data_in	data_width_g	in	Data in
hibi_av_in	1	in	Address valid
hibi_empty_in	1	in	No more data
hibi_comm_in	5	in	Hibi command
hibi_re_out	1	out	Read enable

4.1.7 HIBI TRANSMIT INTERFACE

Table 9

Port	Width	Direction	Description
hibi_data_out	data_width_g	in	Data out
hibi_av_out	1	in	Address valid
hibi_full_in	1	in	Receiver fulll
hibi_comm_out	5	in	Hibi command
hibi_we_out	1	out	Write enable

4.1.8 INTERRUPT INTERFACE

Table 10

Port	Width	Direction	Description
rx_irq_out	1	out	Interrupt, active high

4.1.9 INTERFACE STRUCTURE

Figure 2 shows how HIBI_PE_DMA's interfaces are connected.

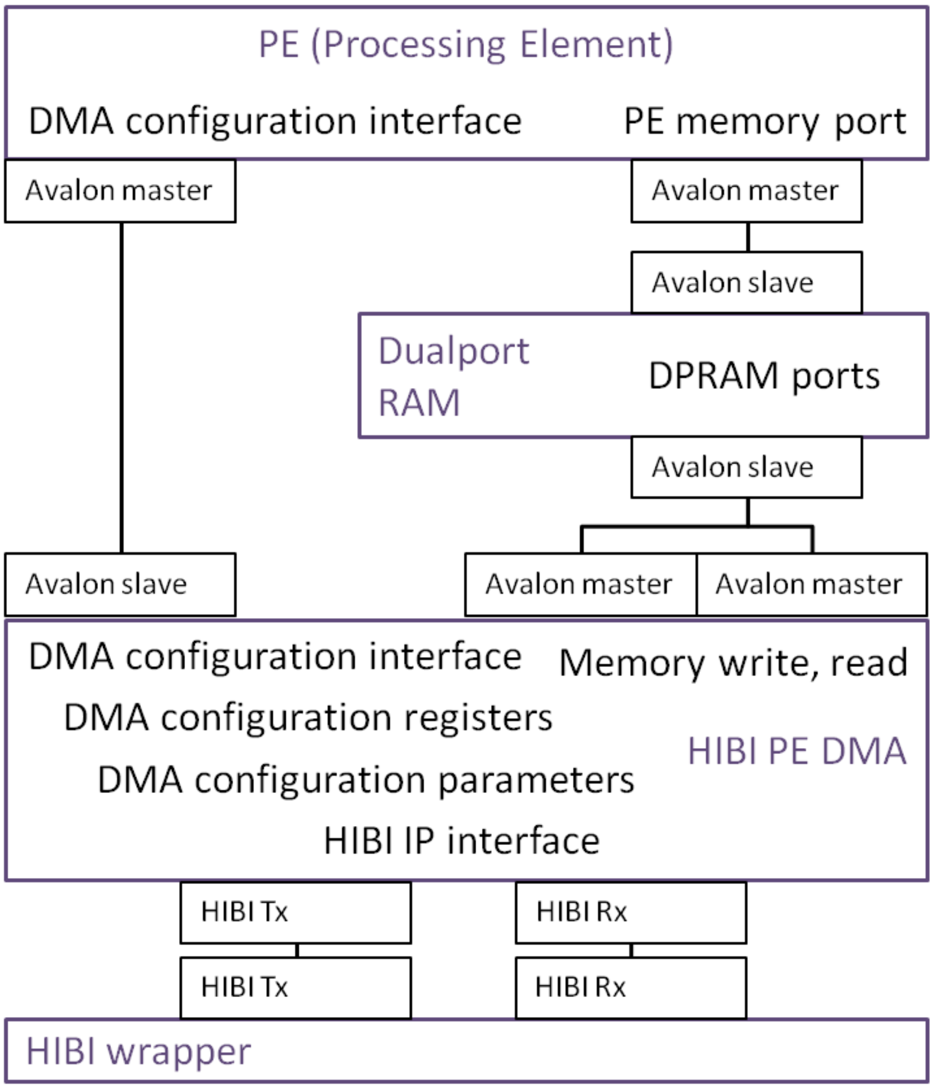

Figure 2: HIBI_PE_DMA connected to a Processing Element (PE), dual-port memory and Hibi

4.1.10 INTEGRATION

Related source files are listed in next table in the order of compilation (when applicable).

Table 11

Filename	Description
hpd_tx_control.vhd	Sending logic
hpd_rx_packet.vhd	RX packet channel
hpd_rx_stream.vhd	RX stream channel
hpd_rx_and_conf.vhd	RX channels and configuration registers
hibi_pe_dma.vhd	Top level

4.1.11 REGISTERS

Table 12

Address	Name	Access	Description
0x00	RX_INITIALIZE	W	Initializes channels
0x01	CONTROL	RW	Control register
0x02	IRQ_STATUS	RW	Read IRQ status and acknowledge interrupts
0x03	TX_MEM_ADDR	W	Address where data to be sent begins
0x04	TX_WORDS	W	How many words to send
0x05	TX_COMM	W	Hibi command to send the data with
0x06	TX_HIBI_ADDR	W	Hibi address to send the data
0x07	RX_HIBI_DATA	R	Current data on hibi rx interface
0xn8	RX_MEM_ADDR	RW	Address where channel n stores received data
0xn9	RX_WORDS	RW	How many words to receive for packet channel n or read acknowledge for stream channel n
0xnA	RX_HIBI_ADDR	RW	Hibi address for channel n to listen

RX_INITIALIZE REGISTER (0x00) initializes channels with previously set data. For every bit that is '1' corresponding channel is initializes. Eg. if bit 3 is '1' then channel 3 will be initialized. Initialization means that the channel reads in registers 0xn0-0xn3 and starts listening hibi receive interface.

Table 13

Bit	Name	Access	Description
[total_channels-1:0]	INIT	W	Initialize channels

CONTROL REGISTER (0x01) is used to enable interrupts, start sending and checking if transmission has been done.

Table 14

Bit	Name	Access	Description
0	TXE	W	Start transmission
1	IE	RW	Interrupt enable
16	TXDONE	R	Transmission completed

IRQ_STATUS REGISTER (0x02) holds the interrupt status. Channel interrupts are acknowledged by writing '1' to the corresponding bit, other interrupt goes low automatically when this register is read. IGNORED_TX is '1' if transmission was tried when previous transmission was still going on. UNKNOWN_RX is '1' when hibi_data_in line has an address that isn't listened by any channel.

Table 15

Bit	Name	Access	Description
[total_channels-1:0]	IRQ	RW	Channel interrupt statuses
addr_width_g-2	IGNORED_TX	R	Last transmission ignored
addr_width_g-1	UNKOWN_RX	R	Received unknown hibi address

TX_MEM_ADDR REGISTER (0x03) tells HIBI_PE_DMA where the beginning of the packet to be sent is in the shared memory.

Table 16

Bit	Name	Access	Description
[addr_width_g-1:0]	ADDRESS	W	Packet's starting address

TX_WORDS REGISTER (0x04) defines the amount of words to send.

Table 17

Bit	Name	Access	Description
[words_width_g-1:0]	WORDS	W	How many words to send

TX_COMM REGISTER (0x05) defines witch hibi command to use when sending the data.

Table 18

Bit	Name	Access	Description
[4:0]	CMD	W	Hibi command

TX_HIBI_ADDR REGISTER (0x06) is used as the target hibi address when sending.

Table 19

Bit	Name	Access	Description
[addr_width_g-1:0]	ADDR	W	Hibi address to use

RX_HIBI_DATA REGISTER (0x07) is used to read the unknown incoming address from hibi. When HIBI_PE_DMA receives an address from hibi bus that no channel is listening it stalls the hibi and raises an interrupt. This register can then be used to read the address.

Table 20

Bit	Name	Access	Description
[addr_width_g-1:0]	ADDR	R	Contents of hibi_data_in bus

RX_MEM_ADDR REGISTER (0xn8) holds the address of the beginning of the memory area where to store incoming data from channel n. Area should be at least as big as the amount of words to receive specified with register 0xn2. Address is updated only when channel is initialized.

Table 21

Bit	Name	Access	Description
[addr_width_g-1:0]	ADDRESS	RW	Where to store data

RX_WORDS REGISTER (0xn9) tells packet channels how many words to receive before interrupting. Amount is updated only when channel is initialized. For stream channels this register tells the length of the rx memory region when initialized and for already initialized stream channels this register acknowledges that WORDS words have been read from the rx memory and allows the channel to continue receiving. When read for returns amount of words already received.

Table 22

Bit	Name	Access	Description
[words_width_g-1:0]	WORDS	RW	Words to receive or acknowledge

RX_HIBI_ADDR REGISTER (0xnA) specifies the hibi address to listen and receive data from. Address is updated only when channel is initialized.

Table 23

Bit	Name	Access	Description
[addr_width_g-1:0]	ADDRESS	RW	Hibi address to listen