

USB3H-IP (USB3.0 Host function IP) demo reference design manual

Rev1.0E 10-March-2015

This document describes USB3H-IP (USB3.0 host function IP-Core) demo reference design using Altera or Xilinx evaluation board and USB3.0 adapter board. Refer to the host demo manual for operation procedure in the following web-site.

For Altera: http://www.dgway.com/USB3-IP_A_E.html

For Xilinx: http://www.dgway.com/USB3-IP_X_E.html

1. Recommended knowledge

User is recommended to acquire following technical knowledge, skill, or experience to understand this reference design.

1. Basic operation of Altera or Xilinx FPGA design tool and CPU firmware development tool.
2. Basic knowledge of C language.
3. Basic knowledge of USB3.0 operation.
4. FAT system knowledge (if user wants to develop FAT application system).

Because reference system use external PHY device (TUSB1310A from Texas Instruments) in the PHY layer, users are recommended to refer this datasheet.

TUSB1310A datasheet: <http://www.ti.com/product/TUSB1310A>

User can apply USB2.0 knowledge to the USB3.0 system operation understanding, and software design is almost the same between USB2.0 and USB3.0. This reference design implements host system that controls USB Mass Storage Class (Bulk-Only). Refer to the technical documents, web-site, or technical magazine for detail of USB3.0 hardware (protocol, or serial transfer technology) or difference between USB2.0 and USB3.0.

This reference design use FAT system for Mass Storage operation application, however, File System in the design uses freeware (FatFs) so that there is no operation guarantee or technical support provided from DesignGateway or original designer of FatFs.

FatFs introduction site: http://elm-chan.org/fsw/ff/00index_e.html

2. Hardware Design Description

2-1 Altera Hardware Description

Figure-1 shows block diagram of USB3.0 Host reference design system on Altera platform.

1. USB3.0-IP Host Core Includes both USB3.0 Link Layer and Protocol Layer.
2. usb_rp_c4gx Top module of FPGA design
3. PIPE_TO_IOB Connects PIPE I/F of IP-Core and IOB of FPGA for external PHY device I/F.
4. nios_proc64_c4gx This module includes NiosII processor and DDR controller, and executes IP-Core internal registers and DMA access.
5. Software For “Device Framework (USB3.0 Standard section 9 or later)” and upper Control, they are managed by software. This design implements host system that control Mass Storage Class device.
6. Demo (Adapter) board AB08-USB3HSMC adapter board (from DesignGateway) mounts External PHY device (TUSB1310A), and communicates with FPGA via HSMC I/F.

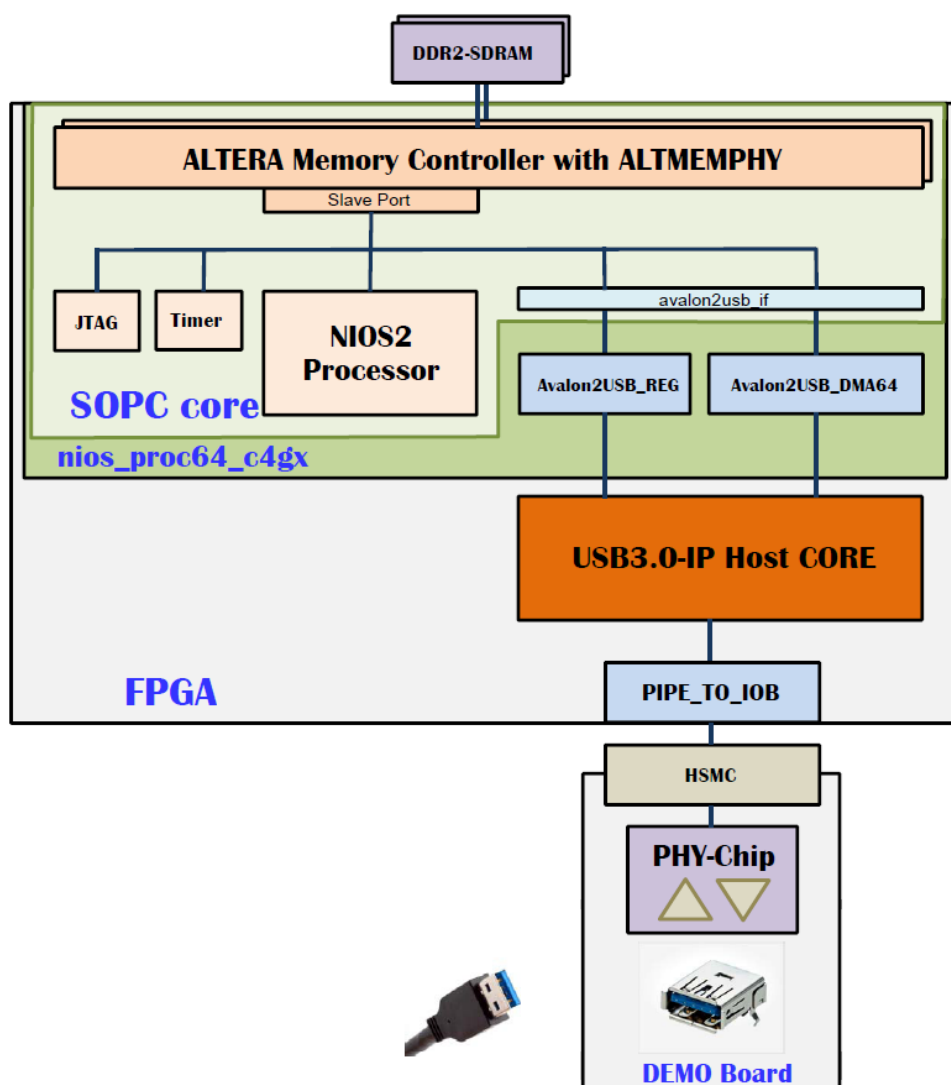


Figure-1: Host Reference Design on Altera (CycloneIV GX board) Platform

- **USB3.0-IP Host Core**

Refer to the datasheet for IP-Core details.

- **usb_rp_c4gx (usb_rp_a2gx for ArriaII GX reference design)**

This Top-module includes each module and IP-Core connection in the reference design. It also includes setting signal output for PHY device.

Because AB08-USB3HSMC demo adapter board use A-type USB connector due to the compatible usage between host and device mode, Plus(+) and Minus(-) connection of USB3.0 differential serial line connection is inverted. To invert +/- polarity in the PHY device, this design use RX polarity inversion detection feature (Lane\Polarity). So that LANE_POLARITY output from IP-Core is connected to RX_POLARITY pin of PHY device. **Note that if user system connects +/- with straight forward connection (no inversion necessary), user must set RX_POLARITY to '0'**. For TX side, +/- polarity is automatically detected and adjusted by device side in USB3.0 Standard.

To set Strapping Options in the PHY device, state switch operation between Options setting state and normal operation state is done within this module.

- **PIPE_TO_IOB**

This module connects between IP-Core and external PHY device. See figure-2 for this module structure. This module operates very high speed of 250MHz (4ns cycle time) for PIPE interface. Important point is...

- ✓ For receive signal and clock timing, use Place & Route for timing adjustment. Also use phase_shift feature of PLL if necessary.
- ✓ For TX_CLK, use phase_shift feature of PLL so that phase is 180 degree shifted from internal clock.
- ✓ For signal path from input DFF and IP-Core and from IP-Core to output DFF, use (add) pipeline DFF so that design can meet timing requirement.

For more detail of timing adjustment technique in this module, refer to the source code of this reference design.

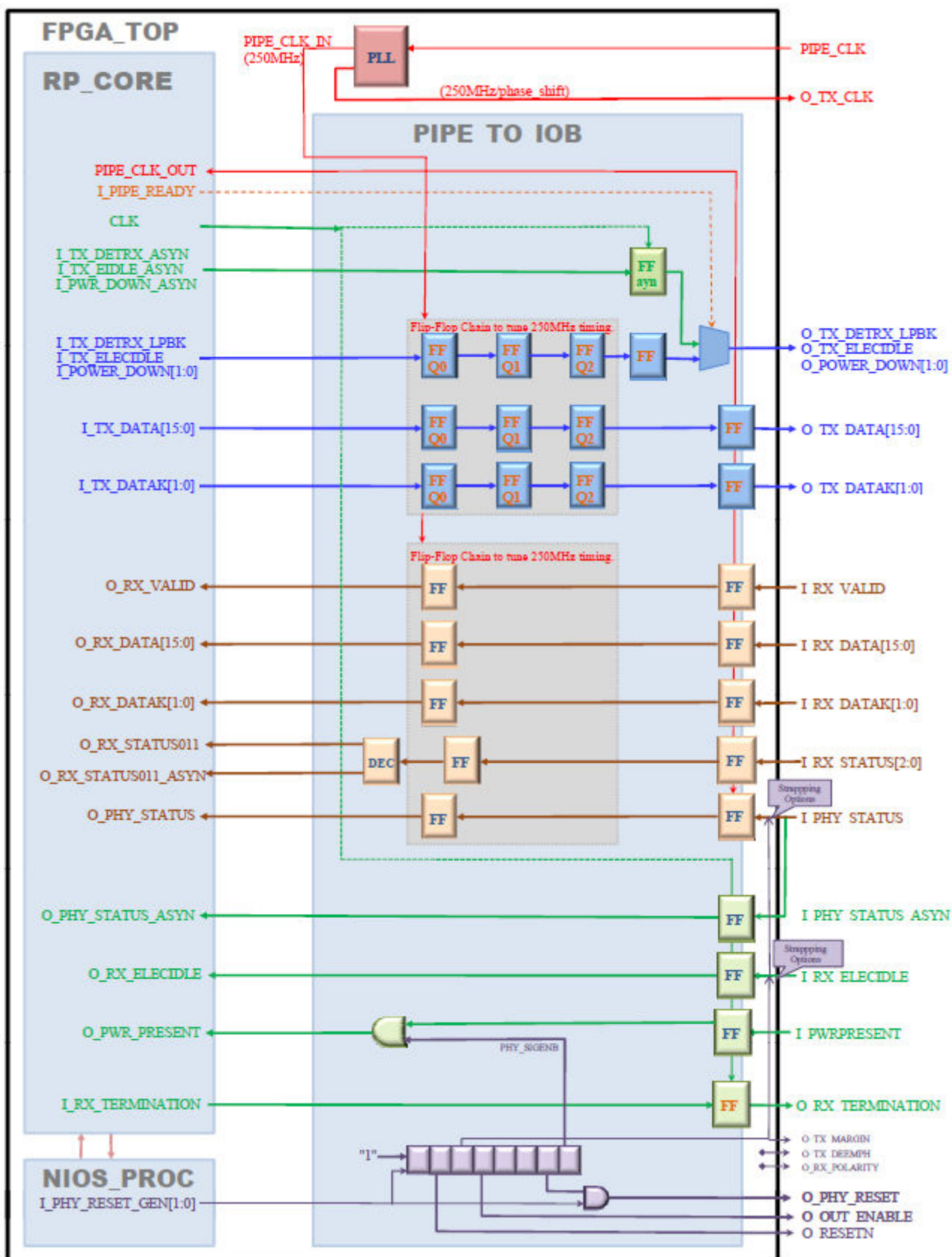


Figure-2: PIPE TO IOB structure of Altera design

- nios_proc64_c4gx (nios_proc_a2gx for ArriaII GX reference design)

This module includes NiosII processor, DDR controller (altmemddr), Avalon2USB_REG, and Avalon2USB_DMA64/Avalon2USB_DMA. Avalon2USB_REG is the connection module for register access from NiosII via Avalon bus., while Avalon2USB_DMA64 (64bit bus version for c4gx)/Avalon2USB_DMA (32bit bus version for a2gx) is connection module for DMA access from altmemddr via Avalon bus. Note that reference design uses two DDR controllers for CycloneIV GX platform design , and each DDR controller connects with 64Mbytes DDR memory so that 128Mbytes in total.

Figure-3 below shows block diagram of Avalon2USB_DMA64 module. For Avalon bus operation, refer to the technical document from Altera, and for IP-Core interface, refer to the IP-Core datasheet. In CycloneIV GX design, Avalon bus operates 64bit (@75MHz) operation while IP-Core operates 32bit (@125MHz) so this module inserts PFIFO for bus size conversion. For more detail, refer to the source code in the reference design.

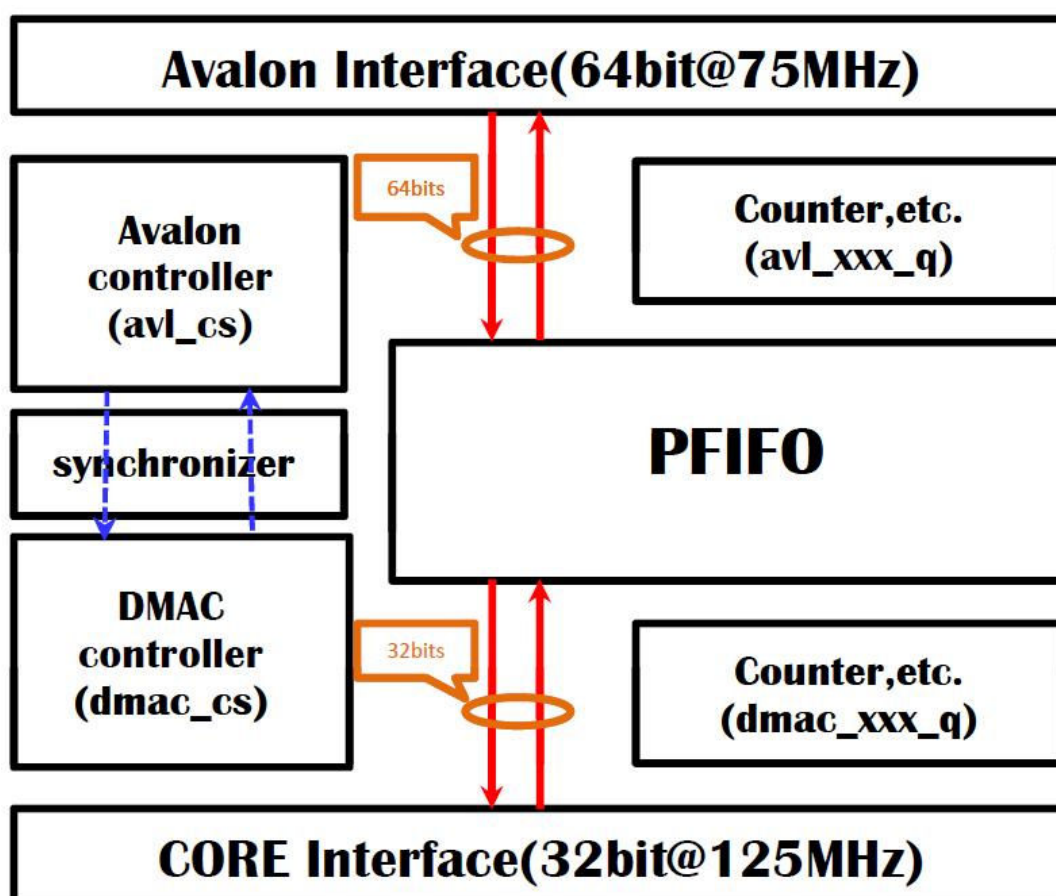


Figure-3: Avalon2USB_DMA64 structure

2-2 Xilinx Hardware Description

Figure-4 shows block diagram of USB3.0 Host reference design system on Xilinx platform.

- | | | |
|----|----------------------|---|
| 1. | USB3.0-IP Host Core | Includes both USB3.0 Link Layer and Protocol Layer. |
| 2. | usb_rp_sp605 | Top module of FPGA design |
| 3. | PIPE_TO_IOB | Connects PIPE I/F of IP-Core and IOB of FPGA for external PHY device I/F. |
| 4. | mb_proc_npi64 | This module includes MicroBlaze processor and DDR controller (MPMC), and executes IP-Core internal registers and DMA access. |
| 7. | Software | For “Device Framework (USB3.0 Standard section 9 or later)” and upper Control, they are managed by software. This design implements host system that control Mass Storage Class device. |
| 5. | Demo (Adapter) board | AB07-USB3FMC adapter board (from DesignGateway) mounts External PHY device (TUSB1310A), and communicates with FPGA via FMC I/F. |

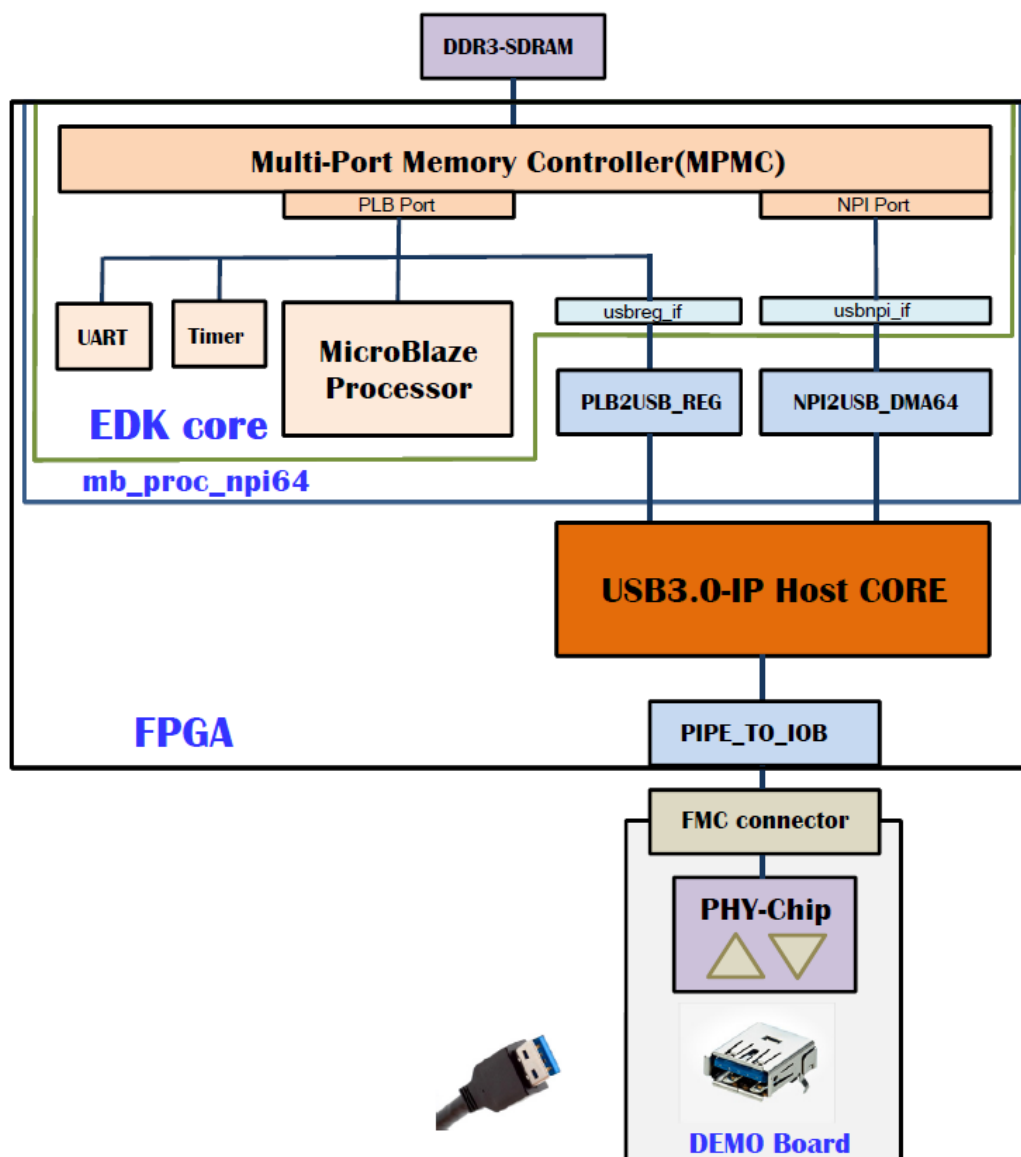


Figure-4: Host Reference Design on Xilinx (SP605 board) Platform

- **USB3.0-IP Host Core**

Refer to the datasheet for IP-Core details.

- **usb_rp_sp605**

This Top-module includes each module and IP-Core connection in the reference design. It also includes setting signal output for PHY device.

Because AB07-USB3FMC demo adapter board use A-type USB connector due to the compatibility with device mode, Plus(+) and Minus(-) connection of USB3.0 differential serial line connection is inverted. To invert +/- polarity in the PHY device, this design use RX polarity inversion detection feature (Lane\Polarity). So that LANE_POLARITY output from IP-Core is connected to RX_POLARITY pin of PHY device. **Note that if user system connects +/- with straight forward connection (no inversion necessary), user must set RX_POLARITY to '0'**. For TX side, +/- polarity is automatically detected and adjusted by device side in USB3.0 Standard.

To set Strapping Options in the PHY device, state switch operation between Options setting state and normal operation state is done within this module.

- **PIPE_TO_IOB**

This module connects between IP-Core and external PHY device. See figure-5 for this module structure. This module operates very high speed of 250MHz (4ns cycle time) for PIPE interface. Important point is...

- ✓ For receive signal and clock timing adjustment, use phase_shift feature of DCM_SP.
- ✓ For TX_CLK, use ODDR2 in IOB.
- ✓ For signal path from input DFF and IP-Core and from IP-Core to output DFF, use (add) pipeline DFF so that design can meet timing requirement.

For more detail of timing adjustment technique in this module, refer to the source code of this reference design.

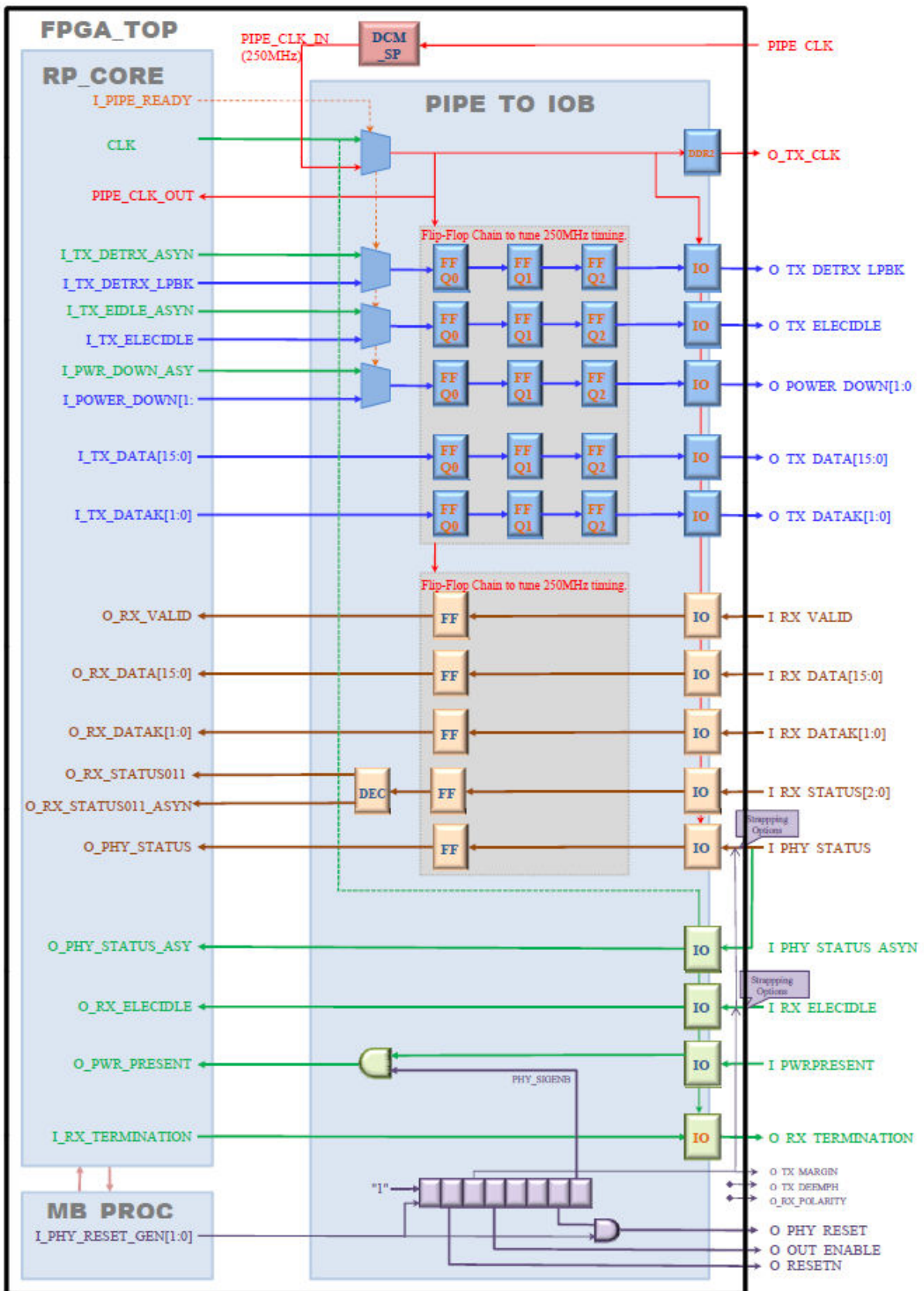


Figure-5: PIPE TO IOB structure of Xilinx design

- mb_proc_npi64

This module includes MicroBlaze processor, DDR controller (mpmc), PLB2USB_REG, and NPI2USB_REG64. PLB2USB_REG is the connection module for register access from MicroBlaze via PLB bus, while NPI2USB_REG64 is connection module for DMA access from mpmc via NPI bus.

Figure-6 below shows block diagram of NPI2USB_DMA64 module. For NPI bus operation, refer to the technical document from Xilinx, and for IP-Core interface, refer to the IP-Core datasheet. In SP-605 design, NPI bus operates 64bit (@83MHz) operation while IP-Core operates 32bit (@125MHz) so this module inserts PFIFO for bus size conversion. For more detail, refer to the source code in the reference design.

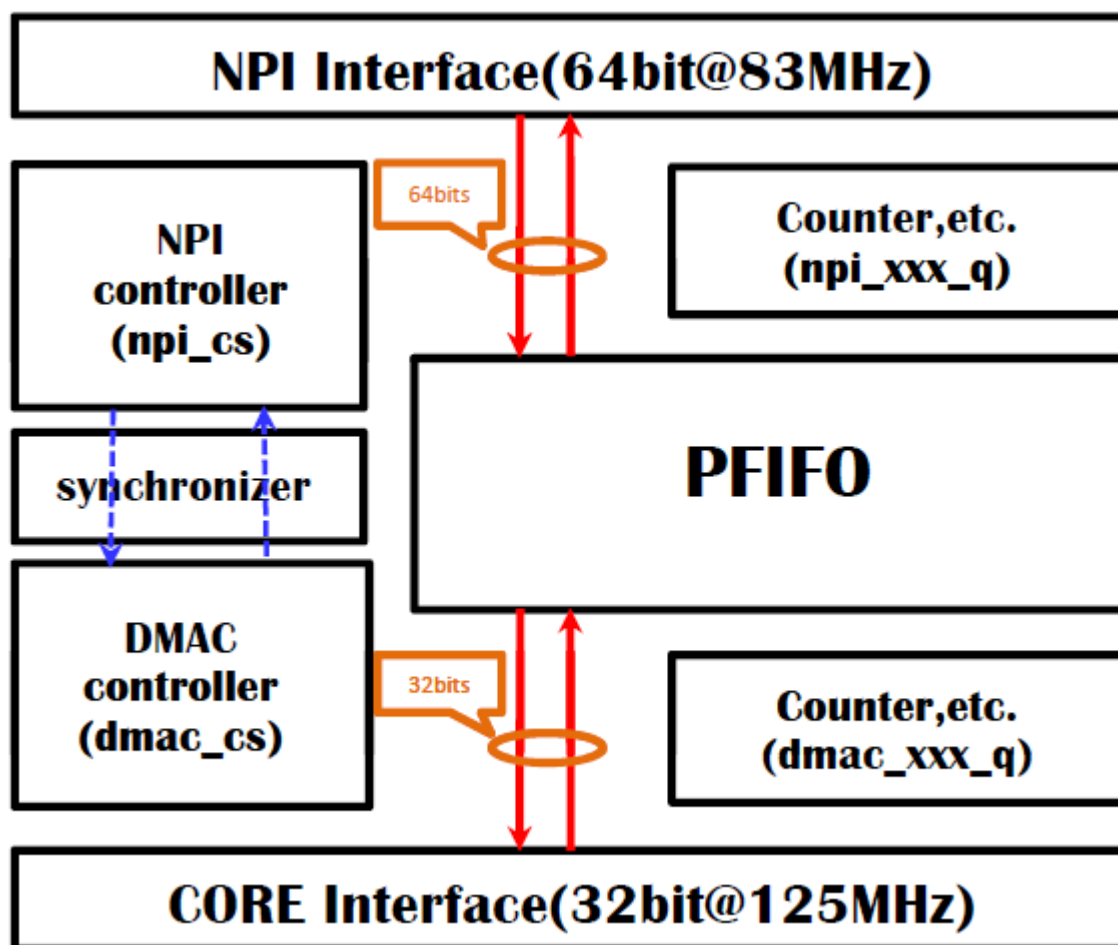


Figure-6: NPI2USB_DMA64 structure

3. Software description

Following description is software in this reference design for both Altera and Xilinx. For more detail, refer to the IP-Core datasheet and source code in this design.

- usb_core_rp.h: Common header file
- usb_com_rp.c: Command Shell
- usb_bus_ctrl_rp.c: Control of bus status polling or interrupt.
- usb_dev_ctrl_rp.c: Control of device request before device class detection.
- usb_hub_ctrl.c: Control of device request for Hub class. device
- usb_msc_ctrl_rp.c: Control of device request for Mass Storage device and SCSI (ATA) command control.
- usb_msc_ctrl_rw.c: Data read/write control in SCSI (ATA) command.
- usb_rp0_ctrl.c: Control of device request (setup) by RPO [root point zero]
- usb_rpn_ctrl.c: Control of RPO[root point out] and RPI [root point in] that generate Bulk-only transfer.
- usb_bffr_ctrl_rp.c: Control of logging and buffering feature, and UART control.

Notes of Xilinx SP605 design

Because software of this reference design has large command shell size, it exceeds maximum on-chip BRAM size (64Kbytes) that can be connected to MicroBlaze via LMB (Local Memory Bus). So that this reference design connects additional on-chip RAM to the PLB bus. To control text code position, it needs to modify Linker Script in manual. Refer to Xilinx Answer#21639 for this technique.

- *lscript.ld: Linker Script to control text code position.*

For Mass Storage device demo operation, this reference design uses FAT system modified from free-ware of FatFs. C Source code related to this FatFs is stored under /FatFs folder in the reference design project. Most part is same as original design of FatFs, while following part is modified for this reference design.

- Usb_rp_api.c: Lower layer interface of FatFs for IP-Core interface in this design.
- ffconf_h: Some 'define' is modified. (Use "Changed by Design Gateway" string to find modification in the source code)
- ff.h: In this IP-Core, read/write data must be aligned to 256 bytes boundary. So work buffer for FatFs is used as pointer variables and assigned to 256 bytes boundary position in initialization process. Use "Changed by Design Gateway" string to find this modification.

4. Revision History

Revision	Date	Description
1.0	2012/01/06	First Japanese version release
1.0E	10-March-2015	First English version release (includes both Altera and Xilinx)

Copyright: 2012 Design Gateway Co,Ltd.