USB3.0 (Device) IP Protocol & Link Layer Core

13-May-2015 Product Specification Rev 1.4E

Design Gateway Co.,Ltd

54 BB Building 13th Fl., Room No.1302 Sukhumvit
21 Rd. (Asoke), Klongtoey-Nua, Wattana,
Bangkok 10110, Thailand
Phone: (+66) 02-261-2277
Fax: (+66) 02-261-2290
E-mail: sales@design-gateway.com
URL: www.design-gateway.com

Core Facts

Provided with Core	
Documentation	Reference design manual, Demo instruction manual
Design File Formats	Encrypted Netlist
Constraints Files	User constraint file
Instantiation Templates	VHDL
Reference Designs & Application Notes	ISE/Vivado/EDK Project, See Reference Design Manual
Additional Items	Demo on SP605, ML605, KC705, ZC706 (Requires AB07-USB3FMC)
Support	
Support Provided by Design Gateway Co., Ltd.	

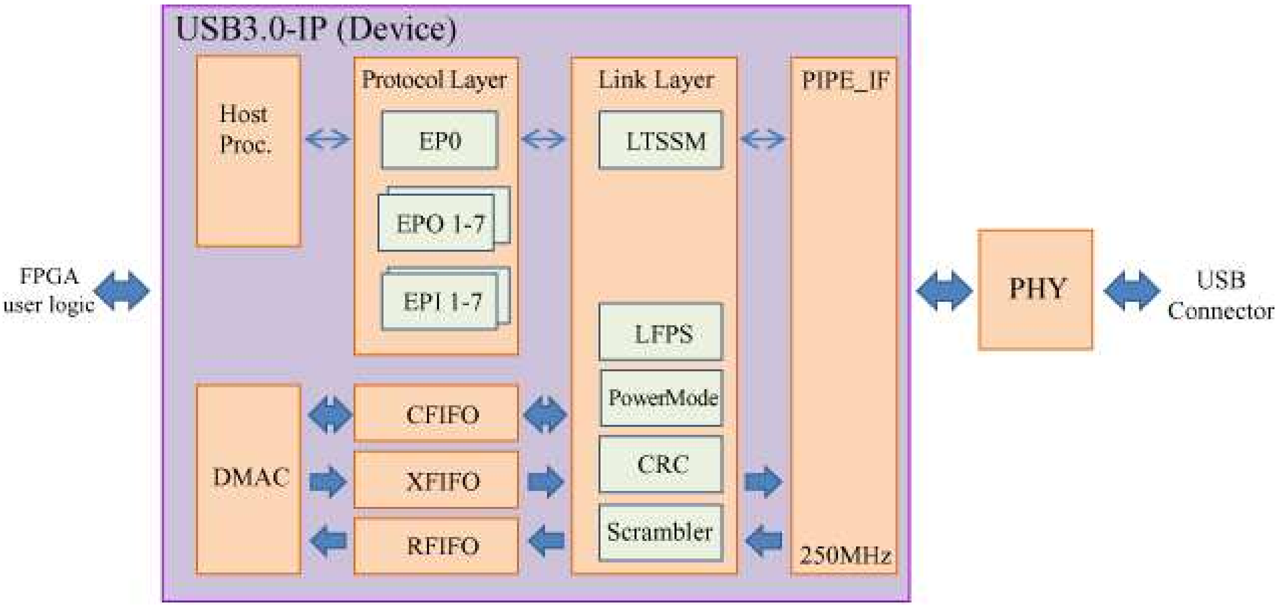
Features

- Compliant with the USB 3.0 specification revision 1.0
- USB3.0 Device Contoroller
- Imprement link layer and protocol layer
- Physical layer interfaces to PHY chip by TI (TUSB1310A)
- IP core clocks are adjustable (250MHz for PIPE I/F, more than 125MHz for internal)
- Support 16bit PIPE interface
- Support IN/OUT end point up to 15 points
  - 1 point for control
  - 7 points each for IN/OUT
- Support All transmission taps (Control, Bulk, Isochronous and Interruput transmission)
- Simple transaction interface with Host processor or DMA interface
- Reference design available on SP605, ML605, KC705, ZC706 with Design GateWay AB07-USB3FMC Card

Table 1: Example Implementation Statistics (Control x1, IN x2, OUT x2)

Family	Example Device	Fmax (MHz)	Slice Regs	Slice LUTs	Slices[1]	IOB[2]	GCLK	BRAM	DCM / CMT	Design Tools
Spartan-6	XC6SLX45T-3FGG484	156	3887	5919	3154	68	2	10	2	ISE14.6
Virtex-6	XC6VLX240T-1FF1156	208	3868	5862	2592	68	2	6	2	ISE14.6
Kintex-7	XC7K325T-2FFG900	256	3837	6292	2518	68	2	6	2	ISE14.6
Zynq-7000	XC7Z045-2FFG900	250	3837	6272	2675	68	2	6	2	ISE14.6

Notes:
1) Actual slice count dependent on percentage of unrelated logic – see Mapping Report File for details
2) Assuming all core I/Os, I/Os for TI_PHY and clocks are routed off-chip and internal signals are enclosed by FF.
3) The number of end port of the core is variable. In case of Bulk transmission, FIFO size can be reduced to one.



* LTSSM: Link Training and Status State Machine, LFPS : Low Frequency Periodic Signaling

Figure 1: USB3.0 (Device) IP Block Diagram

Applications

USB3.0 (Device) IP Core is ideal for use in a USB3.0 supported device system which require high bandwidth up to 5.0Gbps. This IP Core will process almost all USB3.0 protocols (some part of chapter 6, chapter 7 and 8 of the USB3.0 specification) by hardware. It achieves processing by low-end CPU. By setting parameter, this IP Core flexibly supports both a device achieved by minimum end point such as mass storage class and a high-end device which need multiple end points. This IP Core is optimized for saving FPGA internal logic resource by eliminating legacy USB protocol of USB2.0 (480Mbps) or earlier, so that it provides most cost-effective solution for 5Gbps super speed implementation.

General Description

The USB3.0 (Device) IP Core implements link layer and protocol layer. Only setting data address on memory prepared by Host processor, transmission length and other parameter to the register in the IP Core, IP Core will process dividing into packets, adding CRC & Scramble and flow control on USB bus, and return processing result to the register. Data receive process is also same flow.

Host interface of the IP Core consist of simple register access interface, provide simple DMA access interface for memory and able to connect to Microblaze or MPMC (Multi Port Memory Controller) easily.

For connection with PHY chip, it is compliant with standard PIPE interface. So it is just able to connect through Flip-Flop for timing adjustment with FPGA port.

Internal clock in the IP Core is more than 125MHz (125M x 4bytes = 500MB/s) and clock for connection with PHY chip is fixed to 250MHz (500MHz x 2 bytes). However Host interface and DMA interface are able to connect by low frequency clock after synchronization.

Functional Description

The USB3.0 (Device) IP Core is structured by 3 blocks.

Protocol Layer

Protocol Layer manages data on memory assigned by register from Host processor, and divides to USB3.0 packet and sends to Link layer. Receiving is opposite process. This layer manages End-to-End sequence number with host (host bus adaptor auch as PC) and credit process.

- **EP0(End point 0)**

  Process control transmession specified by USB3.0.

  It includes setup packet receiving, data transmission for control (In/Out) and status transmission.
- **EPO(End point Out 1~7)**

  Process BULK-OUT transmission, INTERRUPT-OUT transmission and ISOCHRONOUS-OUT transmission.
- **EPI(End point In 1~7)**

  Process BULK-IN transmission, INTERRUPT- IN transmission and ISOCHRONOUS-IN transmission.
- **MPP(Multi-purpose point)**

  It is used for transmission/receiving process of Port-Capabilities / Port-Configuration/Response after changing to U0 status and used for transmission/receiving process of Isochronous Time Stamp(ITS) and Link Management Packet(LMP).

- **FIFO**

  CFIFO is data sending/receiving FIFO for EP0. XFIFO is data transmission FIFO (up to 2pcs/automatic assignment) which is shared at EPI. RFIFO is data receiving FIFO (up to 2pcs/automatic assignment) which is shared at EPO.

- **DMAC, Arbiters**

  DMA request from each end points and packet send/receive request are adjusted and sequencially processed in Protocol Layer.

Link Layer

Link Layer adds CRC (CRC-5 or16 or 32) to packet from Protocol Layer, scrambles it and send it with 8B(x4) symbol to PIPE_IF. Sending and receiving between links and flow management are also processed by this layer.

- **LSSSM block**

  Manage link status specified by USB3.0, initialize and do sequential processing of power status.

- **FLOW block**
  Flow control between links. Manage transmission status (normal or abnormal), retry and credit process between links.

- **LFPS block**

  Send and receive LFPS(Low Frequecncy Periodic Signaling) when initialization or returning from power mode.

- **Power Mode block**

  Process send / receive Link Command when switching to power mode.

- **Transmission/Receiving process block**

  Add/check CRC, add/cancel scramble and add/check Link Control word.

PIPE Interface

PIPE interface sends/receives data and signal from/to Link Layer to/from PHY, synchronized with PIPE clock.

- **Transmission block**

  It converts 8B(x4) symbol to 8B(x2) symbol, executes Elastic process for the symbol and adds SKIP order set. (In case that internal is 125MHz operation, transceiver side also need Elastic procass because of frequency differential with PHY clock.)

- **Receiving block**
  It is reverse process of Transmission block.

- **Control block**

  Send necessary signal synchronized with PIPE clock, according to state transition of LinkLayer. And It sends signal from PHY to Link Layer synchronized with internal clock.

FPGA Controller

Normally host processor which executes application software is used as FPGA internal controller, and it manages control which is upper than USB device framework (Specification chapter 9 and after) by register access of USB 3.0 Device IP Core. System controller consists of Host processor, DMA interface, memory and so on.

USB3.0 PHY

Use external chip supported USB3.0, such as TUSB1310A by TI which has PIPE_IF.

Core I/O Signals

Core I/O signals are not fixed with specified device and any pins to able to connect to user logic flexibly.All core I/O signals are shown as following table 2.
Logic of these signals are active high if no specified.

Table 2: Core I/O Signals.

Signal	Signal Direction	Description
Common Interface Signal		
RST_N	In	Reset USB3.0 (device) IP core. Active low.
CLK	In	IP Core operating frequency output（125MHz）.
PCLK	In	PIPE clock (250MHz). Input same frequency (phase is able to adjust by DCM) with PIPE clock generated by PHY chip. Able to switch to the constant operating clocks (such as CLK) during PCLK stop.
I_EPO_ENB[7:1]	In	Define implementation/un-implementation of End Point Out(EPO). Able to implement up to 7 points. [1] EPO1, [7] EPO7. 1=Implement
I_EPI_ENB[7:1]	In	Define implementation/un-implementation of End Point In(EPI). Able to implement up to 7 points. [1] EPI1, [7] EPI7. 1=Implement

Signal	Signal Direction	Description
PIPE Interface Signal		
O_PIPE_READY	Out	PIPE clock (PCLK) is operating. After changing to other state except P3, detect rising of I_PHY_STATUS_ASYN and assert.
O_RX_TERMINATION	Out	Control RX termination existence (yes or no). 1 = yes, 0 = no. Change asynchronizing with PCLK (change even though during PCLK stop)
O_TX_DETRX_ASYN	Out	Control detecting RX termination existence of opposite side. Change asynchronizing with PCLK, Use it at near P3 state(O_PIPE_READY negate).
O_TX_EIDLE_ASYN	Out	Control TX signal(SSTXP/SSTXN) to electrical idle state. Change asynchronizing with PCLK, Use it at near P3 state(O_PIPE_READY negate).
O_PWR_DOWN_ASYN[1:0]	Out	Control power state of PHY. 00:P0, 01:P2, 10:P3, 11:P3 Change asynchronizing with PCLK, Use it at near P3 state(O_PIPE_READY negate).
O_TX_DETRX_LPBK	Out	Control detecting RX termination existence of opposite side. Change synchronizing with PCLK, Use it at except near P3 state(O_PIPE_READY assert).
O_TX_ELECIDLE	Out	Control TX signal(SSTXP/SSTXN) to electrical idle state. Change synchronizing with PCLK, Use it at except near P3 state(O_PIPE_READY assert).
O_POWER_DOWN [1:0]	Out	Control power state of PHY. 00:P0, 01:P2, 10:P3, 11:P3 Change synchronizing with PCLK, Use it at except near P3 state(O_PIPE_READY assert).
I_PWRPRESENT	In	Input the state of power supplying to VBUS. Asynchronizing with PCLK.
I_RX_ELECIDLE	In	Input electrical idle state of RX signal (SSRXP/SSRXN). Asynchronizing with PCLK.
I_PHY_STATUS_ASYN	In	Input the signal which control to input PHY status. Asynchronizing with PCLK. After changing from P3 state, it is used for detecting PCLK operation start.
I_RX_STATUS011_ASYN	In	Input detected result of RX termination existence of opposite side. Asynchronizing with PCLK. 1 = yes.
I_PHY_STATUS	In	Input the signal which control to input PHY status.Synchronizing with PCLK. During O_PIPE_READY is asserted, it displays detecting completion of RX terminal existence.
I_RX_STATUS011	In	Input detected result of RX termination existence of opposite side.Synchronizing with PCLK. Valid when I_PHY_STATUS = 1. 1=yes.

O_TX_DATAK[1:0]	Out	TX symbol (8B code). K code (1) or D code (0). Synchronizing with PCLK.
O_TX_DATA[15:0]	Out	Twice of TX symbol (8B code). Synchronizing with PCLK.
I_RX_VALID	In	The timing which RX symbol is valid. 1 = valid. Synchronizing with PCLK.
I_RX_DATAK[1:0]	In	RXsymbol (8B code). K code (1) or D code (0). Synchronizing with PCLK.
I_RX_DATA[15:0]	In	Twice of RX symbol (8B code). Synchronizing with PCLK.

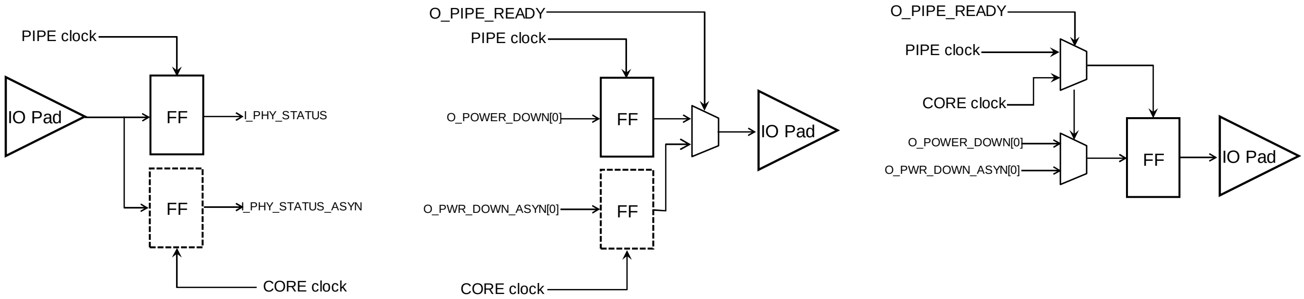
Signal	Signal Direction	Description
Host Register Interface Signal		
I_LINK_REG_RE[15:0]	In	Read signal of control register of link layer. Not all 16 lines are implemented. 1 line is 4bytes. Assert 1 cycle synchronized with CLK.
I_LINK_REG_WE[15:0]	In	Write signal of control register of link layer. Not all 16 lines are implemented. 1 line is 4bytes. Assert 1 cycle synchronized with CLK.
I_PRTE_REG_RE[15:0]	In	Read signal of control register of protocol layer. Not all 16 lines are implemented. 1 line is 4bytes. Assert 1 cycle synchronized with CLK.
I_PRTE_REG_WE[15:0]	In	Write signal of control register of protocol layer. Not all 16 lines are implemented. 1 line is 4bytes. Assert 1 cycle synchronized with CLK.
I_XPP_REG_RE[511:0]	In	Read signal of control register of each end point. Each end point has 32 lines, however not all 32 lines are implemented. 1 line is 4bytes. Assert 1 cycle synchronized with CLK.
I_XPP_REG_WE[511:0]	In	Write signal of control register of each end point. Each end point has 32 lines, however not all 32 lines are implemented. 1 line is 4bytes. Assert 1 cycle synchronized with CLK.
O_EP_REG_RD[31:0]	Out	Read data of all control register. Valid at the next cycle when any _RE is asserted.
I_EP_REG_WD0[31:0]	In	Input write data to control register. Valid at the timing when any _WE is asserted.
I_EP_REG_WD1[31:0]	In	Input write data to control register. Valid at the timing when any _WE is asserted. In case of using EPI4~7, EPO4~7, you have to connect to the signal which is same logic with I_EP_REG_WD0. Connecting to different FF is better for load balancing.
O_EP_IRQ	Out	Interrupt signal from USB core. 1= Interrupt. Level signal.
O_EXT_CNTL [1:0]	Out	USB core external control signal.Control register of link layer can switch ON/OFF.
O_LANE_POL	Out	RX Lane Porarity Inversion. 1: Invert , 0: Not Invert.
I_USB20_RESET	In	Input reset from USB2.0. 1 = reset. In case of no USB2.0 core, fix to 0.

Signal	Signal Direction	Description
DMA Access Interface Signal		
I_DMAC_IDLE	In	Next DMA access start enables.
O_DMAC_REQ	Out	DMA access start. Synchronize with CLK. Assert for 1cycle.
O_DMAC_ADR [31:0]	Out	DMA start address. Valid at O_DMAC_REQ asserted.
O_DMAC_U2M	Out	DMA direction. Valid at O_DMAC_REQ asserted. 0: Memory to USB, 1: USB to Memory.
O_DMAC_LEN[8:0]	Out	DMA length. Valid at O_DMAC_REQ asserted. 0x100: 256words(4K bytes), 0x001: 1word(4bytes)
O_DMAC_BE [3:0]	Out	Valid / Invalid of each bytes.
O_DMAC_DONE	Out	Data transmission completed. Assert 1 cycle after several cycle after sending end data completed.
I_DMAC_M2U_VLD	In	Valid timing of I_DMAC_M2U_DATA in case of DMA from Memory to USB.

I_DMAC_M2U_DATA[31:0] or [63:0]	In	Input data from memory in case of DMA from Memory to USB. Data width can adjust by “DMA64_MODE” paramerter ot top module. 4(or 8) bytes transmission complete (no wait) in the timing I_DMAC_M2U_VLD=1’b1.
I_DMAC_U2M_WAIT	In	Timing which memory cannot accept data in case of DMA from USB to Memory.
O_DMAC_U2M_OUT	Out	Valid timing of O_DMAC_U2M_DATA in case of DMA from USB to Memory.
O_DMAC_U2M_DATA[31:0] or [63:0]	Out	Output data to memory in case of DMA from USB to Memory. Data width can adjust by “DMA64_MODE” paramerter ot top module.. 4(or 8) bytes transmission complete in the timing O_DMAC_U2M_OUT=1’b1, I_DMAC_U2M_WAIT=1’b0

Notes of PIPE interface

This is general PIPE interface, but must be careful to synchronous/asynchronous of signal when it connects with external PHY chip (see figure 2).
Input signals have signal which references at PIPE clock (PCLK) from PHY is stopping. These signals is direct to pin or the re-synchronized signal (with _ASYN) by FF operated by core internal clock. In case that there is reference signal synchronized PIPE clock during PIPE clock is operating, it must be separate to 2 lines.
Some output signals are also changed even though PIPE clock (PCLK) is stopping. These signals is direct to pin or output with re-synchronized (with _ASYN) by FF operated by core internal clock. In case that there is the output signal synchronized PIPE clock during PIPE clock is operating, multiplex in front of a pin or output by synchronized with FF operated by switching clock.



(a) Input Example

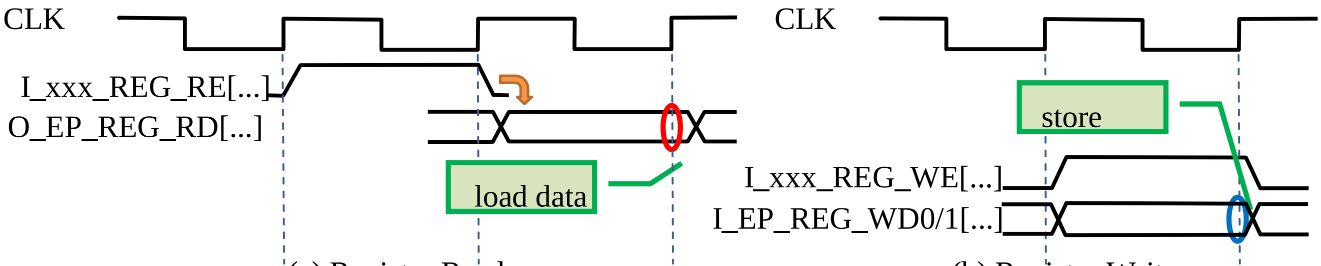
(b) Output Example

(c) Output Alternate

Figure 2: PIPE interface connection example

Timing Diagram of Host register interface

Register access is shown at Figure 3. 1cycle asserting I_xxx_REG_RE to output read value of the register to O_EP_REG_RD. And write when write value is input to I_EP_REG_WD with 1 cycle asserting I_xxx_REG_WE.



(a) Register Read

(b) Register Write

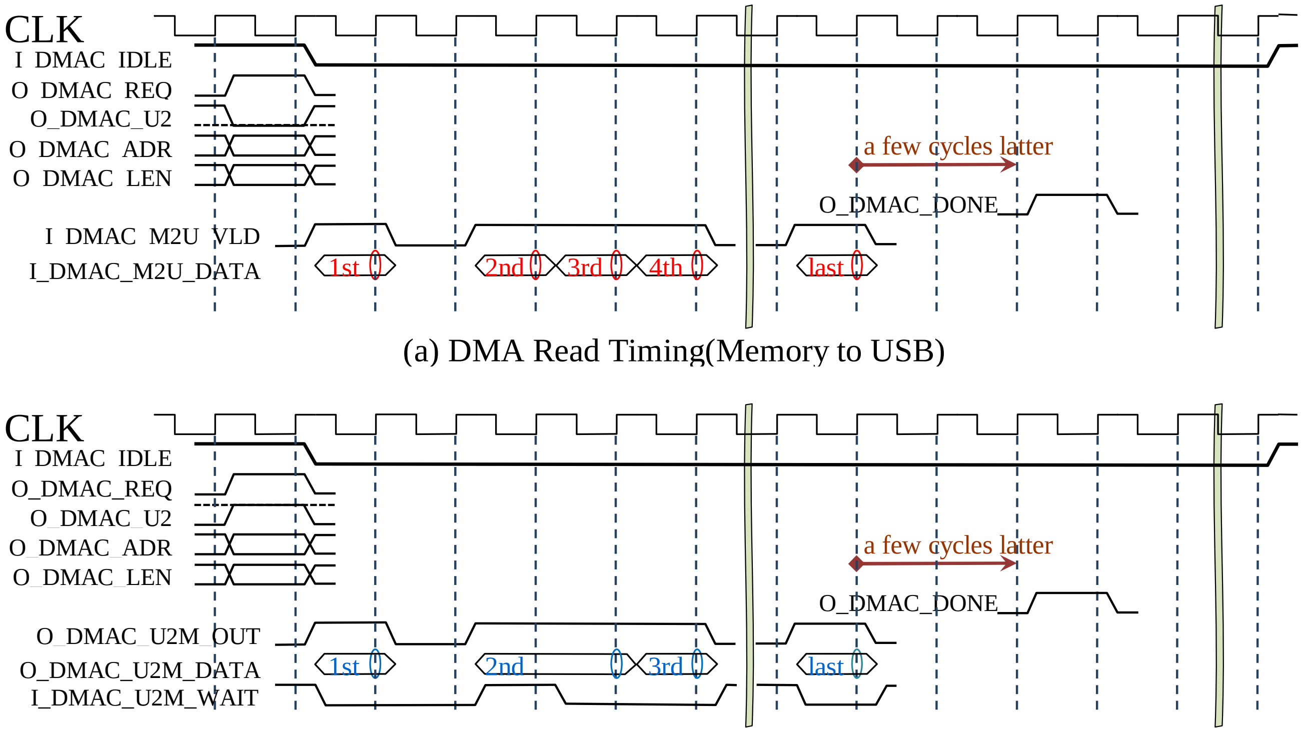
Figure 3 : Signal waveform of host register interface

Timing Diagram of DMA access interface

When I_DMAC_IDLE is asserted, O_DMAC_REQ is also asserted for 1 cycle as shown at Figure 4. In the same time, access address, direction and length are specified.

Data is transferred with flow control by VLD or WAIT signal depend on direction.

When the transmission completed and after several cycles, O_DMAC_DONE is asserted for 1 cycle then DMA access completed.



(a) DMA Read Timing(Memory to USB)

(b) DMA Write Timing (USB to Memory)

Figure 4 : Signal waveform of DMA access interface

IP Core control register

IP Core Control register which Host processor can access are shown as following Table 4.

Table 4: IP Core control register

Register Name	R/W	Offset	Description
Link Layer Register [ADDRESS=C_BASEADDR+LINK_BASE+OFFSET]			
LINK_CNFG	R/W	0h	Link Layer Configuration Register
USB_VALID		[0]	Enable USB bus (LTSSM) state transition except to SS_Disable (Operation enable state)
PMD_ENB		[1]	Enable to receive Power Mode transition request from Host.
SCRB_DIS		[2]	Disbale scramble. For debugging.
EXT_CNT		[5:4]	ON/OFF O_EXT_CNTL [1:0] (refering I/O signal).
U2_INACT_CNT		[15:8]	Set the count time (U2_Inactive_timer) transiting from U1 to U2. Value=Actual time/(USB_CLK× 0x10000)
LINK_CNTL	W	1h	Link Layer Control Register. Instructed if "1" is written. Set 1-bit only(one-hot) at same time. No described bits should be written "0".
GO_RXDET		[0]	Make USB bus state (LTSSM) transit from SS_Disable to Rx_Detect_Active (Operation start). LINK_CNFG[USB_VALID] must be ON before.
GO_RCOV		[3]	Transit from Active_U0 state to Recovery state.
GO_PMD_NUM		[9:8]	Set Power Mode level for transiting by GO_PMD. 01:U1, 10:U2, 11;U3
GO_PMD		[10]	Request transition to Power Mode. Success or Not are depend on Host side status.
GO_U0		[11]	Order to return from Power Mode (Active_U1/U2/U3).
LINK_IRQE	R/W	2h	Link Layer Interrupt Register.
IRQ		[0]	Interruption requesting. Read Only.
ENB		[1]	When IRQ=1, Interrupt signal to external (referring O_EP_IRQ。I/O signal) is ON.
LINK_LTSSM	R	3h	Link Layer Status Register.
RCOV		[6:0]	The reason why Link Layer transit to Recovery state. Kept during LINK_LTSSM[7] is ON.
IRQ_FACTOR		[15:7]	Cause of Link Layer interruption. State of Link Layer is possible to change without command from CPU. When any bit is ON, LINK_IRQE[IRQ] is also ON. Clear by 1 write to the bit.
LTSSM		[28:24]	Current state of Link Layer. The states are compliant with USB3.0 specification. Code allocation, which is implementation specific, is not disclosed in here..
DEV_ADR_OFF		[30]	USB device address is 0 (un-setting). The device address will be cleared by hardware when Link Layer status transits to SS_Disable,Rx_Detect_Reset.
VBUS_OFF		[31]	Current VBUS is OFF.

Register Name	R/W	Offset	Description
Protocol Layer Register [ADDRESS=C_BASEADDR+PRTE_BASE+OFFSET]			
PRTE_CNFG	R/W	0h	Protocol Layer Configuration Register
DEV_ADR		[6:0]	Set device address of USB bus. The device address will be cleared by hardware when Link Layer status transits to SS_Disable,Rx_Detect_Reset.
PRTE_CNTL	W	1h	Protocol control register. Instructed if "1" is written. Set 1-bit only(one-hot) at same time. No described bits should be written "0"
ARBT_RESET		[0]	Arbiter reset of Protocol Layer. For debugging.
PRTE_IRQE	R/W	2h	Protocol Layer Interrupt Enable Register.

EP0		[0]	Enable interruption from EP.
EPO		[7:1]	Enable interruption from EPO1〜7. [17]: EPO1, [23]:EPO7
MPP		[16]	Enable interruption from MPP.
EPI		[23:17]	Enable interruption from EPI1〜7. [1]: EPI1, [7]:EPI7
PRTE_IRQ	R	3h	Protocol Layer interrupt Register.
EP0		[0]	Under requesting interruption from EP. When applicable IRQE bit is 1, interrupt signal for external will be ON.
EPO		[7:1]	Under requesting interruption from EPO1~7. When applicable IRQE bit is 1, interrupt signal for external will be ON. [1]: EPO1, [7]:EPO7
MPP		[16]	Under requesting interruption from MPP. When applicable IRQE bit is 1, interrupt signal for external will be ON.
EPI		[23:17]	Under requesting interruption from EPI1~7. When applicable IRQE bit is 1, interrupt signal for external will be ON. [1]: EPI1, [7]:EPI7
DMA_BASE	R/W	4h	DMA Base Register.
BASE		[31:28]	Upper bit of memory address for DMA.

Register Name	R/W	Offset	Description
End Point Zero (EP0) Register [ADDRESS=C_BASEADDR+XP_BASE+(0x80×0)+0x000+OFFSET]			
EP0_CNFG	R/W	00h	EP0 Configuration Register.
VALID		[0]	EP operation enables. Even it is not enable, Setup Data will be received correctly, transit to "SETUP_IP" state and INVALID is ON when receiving Setup DP packet. When clearing SETU_IP" state, WRDY packet will be received.
AUTO_SETUP		[1]	If this bit is ON, Transit from SETUP without Software instruction and return SETUP acknowledge to Host.
AUTO_RESULT		[2]	If this bit is ON, Transit from RESULT without Software instruction and return to IDLE state.
AUTO_STALLED		[3]	If this bit is ON, Transit from STALLED without Software instruction and return STALL acknowledge to Host.
EP0_CNTL	R/W	01h	EP0 control register. Show status and update all bits of [31:16]. Requesting interrupt if any bit in [13:0] is ON. [23:16] show detected events. [31:24] show transitional events.
EP0_CLR	W	02h	EP0 control register clear. Clear applicable bit by 1 writing.
EP0_SET	W	03h	EP0 control register set. Set applicable bit by 1 writing.
RESULT		[0]	RESULT state. When clearing it, final ACK packet will be received. And then return to IDLE. When STALL is ON, STALL packet will be returned. When VALID is OFF, NRDY packet will be returned.
SETUP		[1]	SETUP in progress state. Setup DP packet is received. SETUP0/1 register store 8 bytes contents. ACK packet (NumP=1) will be returned when clearing. However, when RETRY is ON (Setup DP packet receiving error), retry is attached. When INVALID is ON (VALID reference), NumP=0 will be returned. After that, it will transit to specified state following INRDY/OUTRDY etc.
STALLED		[8]	STALLED state. STALL packet will be returned when clear.
DEFFERED		[9]	DEFFERED state. Transit to this state when the packet attaching Deffered bit is received. ERDY packet will be returned and transit to specified state when clearing.
WRDY		[10]	WRDY state.(waiting ERDY) After sending NRDY packet etc(flow state), become to the status. ERDY packet will be returned and transit to specified state when clearing.
WAIT_RECV		[14]	Waiting for any packet. Valid when BUSY is ON.
BUSY		[15]	It is ON except (IDLE state or [13:0] state). When WAIT_RECV is ON, Clearing this bit to return to

			IDLE. And return to IDLE state when both this bit and [14] are cleared at same time.
	STALL	[16]	STALL packet is received or some abnormal is detected.
	OVER	[17]	Traffic of Host is larger than specified in EP0_DLEN. (Device side is fewer.)
	UNDER	[18]	Traffic of Host is fewer than specified in EP0_DLE. (Device side is more.)
	SEND_STALL	[19]	Send STALL at STALLED state automatically if AUTO_STALLED is enable.
	RETRY	[24]	Error exists in received DP packet (include SetupDP packet).
	INVALID	[25]	Receive Setup DP packet when VALID is OFF.
	NRDY	[27]	Send NRDY from RESULT state.
EP0_DLEN	R/W	04h	EP0 data length setting register.
	DLEN	[20:0]	Data length.
	EOB	[29]	Add EOB (End Of Burst) bit to end of packet when IN transmission.
	OUTRDY	[30]	OUT transmission (from Host to Device) ready.
	INRDY	[31]	IN transmission (from Device to Host) ready.
EP0_PLEN	R	05h	EP0 data length result register
	PLEN	[20:0]	Length of actual transferred data.
	OUTRDY	[30]	OUT transmission (from Host to Device) ready.（copy of DLEN register）
	INRDY	[31]	IN transmission (from Device to Host) ready.（copy of DLEN register）
EP0_BFFR	R/W	06h	EP0 data length buffer (memory address) register
	BADDR	[27:8]	Start address of the memory for data input/output.
EP0_SEQN	R/W	07h	EP0 sequence number register
	SEQN	[4:0]	Current sequence number of packet. Writable, but normally no need.
EP0_SETUS0	R/W	09h	EP0 setup data register 0
EP0_SETUS1	R/W	0Ah	EP0 setup data register 1
	Setup0/1	[31:0]	Received contents by SetupDP packet.

Register Name	**R/W**	**Offset**	**Description**
End Point Out (EPO) Register [ADDRESS=C_BASEADDR+XP_BASE+(0x80×N)+0x000+OFFSET]			
EPO_CNFG	R/W	00h	EPO Configuration Register.
	VALID	[0]	EP operation valid. When invalid and DPpacket is received, NRDY packet will be sent and then WRDY state.
	ISOCHR	[1]	Isochronous mode transmission
	AGGR	[2]	Special mode that immediately sends ACK packet before IP core completes received DP packet data transfer to the main memory.
	AUTO_STALLED	[3]	If this bit is ON, Transit from STALLED without Software instruction and return STALL acknowledge to Host.
	BURST	[18:16]	Set burst length.4~1. Normally“4”, other setting for debug.
	FIFO_REQ	[30]	Arbitration request of receiving FIFO of EPO. For debugging. Read only.
	FIFO_LOC	[31]	The mode which EPO occupies FIFO. For debugging.
EPO_CNTL	R/W	01h	EPO control register. Show status and update all bits of [31:16]. Requesting interrupt if any bit in [13:0] is ON. [23:16] show detected events. [31:24] show transitional events.
EPO_CLR	W	02h	EPO control register clear. Clear applicable bit by 1 writing.
EPO_SET	W	03h	EPO control register set. Set applicable bit by 1 writing.
	COMP	[0]	COMP state (transmission completed). Clear it to return to IDLE. OUTRDY will be cleared. When extra DP packet is received, transit to NRDY.

	STALLED	[8]	STALLED state. Return STALLpacket when clearing.
	DEFFERED	[9]	DEFFERED state. Transit to this state when the packet attaching Deffered bit is received. ERDY packet will be returned and transit to specified state when clearing.
	WRDY	[10]	WRDY state. After sending NRDY packet etc (flow state), become to the status. ERDY packet will be returned and transit to specified state when clearing.
	NRDY	[11]	NRDY state. Under sending NRDYpacket. After a while, transit to WRDY state.
	WAIT_RECV	[14]	Waiting for any packet. Valid when BUSY is ON.
	BUSY	[15]	It is ON except (IDLE state or [13:0] state). When WAIT_RECV is ON, Clearing this bit to return to IDLE. And return to IDLE state when both this bit and [14] are cleared at same time.
	STALL	[16]	STALL packet is received or some abnormal is detected.
	OVER	[17]	Traffic of Host is a lot. (Device side is fewer.)
	UNDER	[18]	Traffic of Host is a few. (Device side is more.)
	RETRY	[24]	Error exists in received DP packet (include SetupDP packet).
	INVALID	[25]	When IDLE state, ACKpacket is received at the state that DPpacket cannot be received. (VALID or OTRDY is OFF)
EPO_DLEN	R/W	04h	EPO data length setting register
	DLEN	[20:0]	Data length.
	OUTRDY	[30]	OUT transmission (from Host to Device). Cleared by COMP state (passing NRDY,ERDY state).
EPO_PLEN	R	05h	EPO data length result register
	PLEN	[20:0]	Actual transferred data length. Except 0 means a data is received.
	OUTRDY	[31]	OUT transmission (from Host to Device).（Copy of DLENregister）
EPO_BFFR	R/W	06h	EPO data length buffer (memory address) register
	BADDR	[27:8]	Start address of the memory for data input/output.
EPO_SEQN	R/W	07h	EPO sequence number register
	COMP_SEQN	[4:0]	Sequence number of transferred packet (ACK sent). Writable, but normally unnecessary.
	BFR_SEQN	[12:8]	Sequence number of stored packet in buffer (memory). Writable, but normally unnecessary.
	RCV_SEQN	[20:16]	Sequence number of received packet. Writable, but normally unnecessary.

Register Name	R/W	Offset	Description
End Point In (EPI) Register [ADDRESS=C_BASEADDR+XP_BASE+(0x80×N)+0x400+OFFSET]			
EPI_CNFG	R/W	00h	EPI Configuration Register.
	VALID	[0]	EP operation valid. When invalid and ACKpacket is received, NRDY packet will be sent and then WRDY state.
	ISOCHR	[1]	Isochronous mode transmission.
	RESTART	[2]	EPI will restart and be waiting for receiving when ACK(NumP=0) packet is sent before Host does not reach to setting transmission length yet. In case of OFF, UNDER will be ON and then transit to COMP state.
	AUTO_STALLED	[3]	If this bit is ON, Transit from STALLED without Software instruction and return STALL acknowledge to Host.
	BURST	[18:16]	Set burst length. Set 4~1. Normally“4”, other setting for debug.
	FIFO_REQ	[30]	Arbitration request of sending FIFO of EPI. For debugging. Read only.
	FIFO_LOC	[31]	The mode which EPI occupies FIFO. For debugging.
EPI_CNTL	R/W	01h	EPI control register. Show status and update all bits of [31:16]. Requesting interrupt if any bit in [13:0] is ON. [23:16] show detected events. [31:24] show transitional events.
EPI_CLR	W	02h	EPI control register clear. Clear applicable bit by 1 writing.
EPI_SET	W	03h	EPI control register set. Set applicable bit by 1 writing.

	COMP	[0]	COMP state (transmission completed). Clear it to return to IDLE. INRDY will be cleared. When extra ACK packet is received, transit to NRDY. When DP packet with EOB is sent, transit to WRDY.
	STALLED	[8]	STALLED state. Clear it to return STALLpacket.
	DEFFERED	[9]	DEFFERED state. Transit to this state when the packet attaching Deffered bit is received. ERDY packet will be returned and transit to specified state when clearing.
	WRDY	[10]	WRDY state. After sending NRDY packet etc (flow state), become to the status. ERDY packet will be returned and transit to specified state when clearing.
	NRDY	[11]	NRDY state. Under sending NRDYpacket. After a while, transit to WRDY state.
	WAIT_RECV	[14]	Waiting for any packet. Valid when BUSY is ON.
	BUSY	[15]	It is ON except (IDLE state or [13:0] state). When WAIT_RECV is ON, Clearing this bit to return to IDLE. And return to IDLE state when both this bit and [14] are cleared at same time.
	STALL	[16]	STALL packet is received or some abnormal is detected.
	OVER	[17]	Traffic of Host is a lot. (Device side is fewer.)
	UNDER	[18]	Traffic of Host is a few. (Device side is more.)
	RETRY	[24]	ACKpacket with Retry is received.
	INVALID	[25]	When IDLE state, ACKpacket is received at the state that DPpacket cannot be sent. (VALID or INRDY is OFF)
	RESTART_IP	[26]	RESTART is processing.
EPI_DLEN	R/W	04h	EPI data lendth setting register
	DLEN	[20:0]	Data length.
	EOB	[29]	When IN transmission, Add EOB(End Of Burst) bit to end packet.
	INRDY	[31]	IN transmission (from Device to Host). Cleared by COMP state (passing NRDY, ERDY state).
EPI_PLEN	R	05h	EPI data length result register
	PLEN	[20:0]	Actual transferred data length. Except 0 means a data is sent.
	INRDY	[31]	IN transmission (from Device to Host).（Copy of DLENregister）
EPI_BFFR	R/W	06h	EPI data length buffer (memory address) register
	BADDR	[27:8]	Start address of the memory for data input/output.
EPI_SEQN	R/W	07h	EPI sequence number register
	COMP_SEQN	[4:0]	Sequence number of transferred packet (ACK received). Writable, but normally unnecessary.
	BFR_SEQN	[12:8]	Sequence number of taken packet from buffer (memory). Writable, but normally unnecessary.
	TRN_SEQN	[20:16]	Sequence number of transferred packet. Writable, but normally unnecessary.
	NUM_SEQN	[28:24]	Sequence number of the packet which has credit by ACK. Writable, but normally unnecessary.

Register Name	**R/W**	**Offset**	**Description**
Multi Purpose Point (MPP) Register [ADDRESS=C_BASEADDR+XP_BASE+(0x80×0)+0x400+OFFSET]			
MPP_CNFG	R/W	00h	MPP Configuration Register.
	TRNS	[0]	Send packet from MPP. It is cleared when completed. When being Active_U0, send Port Capabilities, Port Configuration Response with automatically ON/OFF.
MPP_THD0	R/W	04h	MPP sending packet 0
MPP_THD1	R/W	05h	MPP sending packet 1
	THD0/1	[31:0]	Data of sending packet
MPP_RHD0	R/W	06h	MPP receiving packet 0.
MPP_RHD1	R/W	07h	MPP receiving packet 1.
	RHD0/1	[31:0]	Data of receiving packet
MPP_RCVD	R/W	08h	EPI sequence number register
	MISC_RCNT	[3:0]	When Receiving packet, it will count up +1. After 0xF, return to 0x0. Writable .

	MISC_RCVD_H	[7]	Receive any packet. ON: interruption request. Writable
	PCFG_RCVD	[30]	After Active_U0, receive Port Configuration.
	PCAP_RCVD	[31]	After Active_U0, receive Port Capabilities.

Register Map

Core register location connecting I_xxx_REG_RE/WE[..] in sequence of address is shown as figure 5.

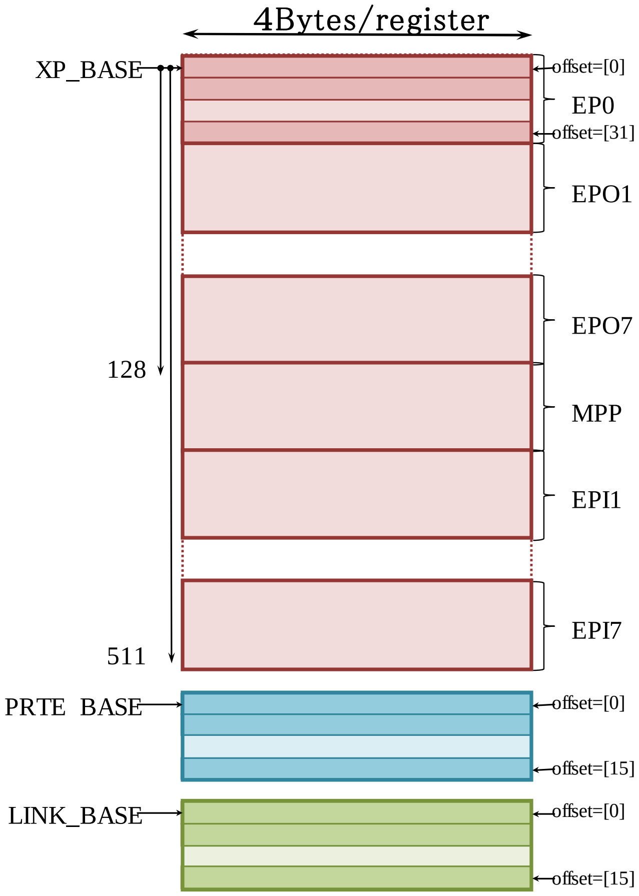


Figure 5 : Address map of IP-Core internal register

IP Core control step

This is the summary of the IP Core control step.

USB bus initialization

When device is connected with USB bus, bus operation is started. The step until U0 state is as following.

▯ Check ON of VBUS by LINK_LTSSM[VBUS_OFF]

▯ Operation ready by LINK_CNFG[USB_VALID].

▯ Order state transition to Rx_Detect_Active state by LINK_CNTL[GO_RXDET]

④ Automatic link if host side is also operation ready

If it is Active_U0 state by LINK_LTSSM[LTSSM], the linking is successful. If it returns to SS_disable, go to② and retry.

Until USB device configured

The step until USB device configured state by control transmission by EP0 after linking is successful, is as following.

▯ EP0_CNFG[VALID] makes EP0 operation ready

- □ Receive SetupDP packet to become SETUP state
- □ First SetupDP will be SET_ADDRESS and no data transmission
- □ Receive STATUS packet to be RESULT state and send final ACK packet

▯ Set device address to PRTE_CNFG[DEV_ADR]. Hardware becomes Address state

▯ After that, operate control transmission such as SetupDP packet several times

- □ In case that control transmission has data transmission, set send data to memory (IN transmission) or keep receiving buffer at memory (OUT transmission) and order to EP0_BFFR and EP0_DLEN
- □ After complete IN/OUT transmission, receive STATUS packet. After that, same as no data transmission

▯ Finally receive SET_CONFIGURATION to become Configured state

BULK_IN, BULK_OUT transmission of USB device

This is the transmission by using BULK_IN(EPI), BULK_OUT(EPO) after device is configured state. Each transmission are differenet for every device class. Following step is for BULK_OUT, but the step for BULK_IN is also nearly same.

▯ Set VALID to ON, and set burst length by EPO_CNFG

▯ Keep receiving buffer in memory and order ot EPO_BFFR and EPO_DLEN
Set estimation data length (or more) to DLEN

▯ If something are received, PLEN shows receiving data length. So check receiving buffer. Clear COMP state and return to ②.

Core verification method

USB3.0 IPdevice IP Core logic can be verified by SP605 Xilinx evaluation board together with AB07-USB3FMC adapter board that enables real operation check. Please ask AB07-USB3FMC adapter board to Design Gateway.

Recommended design skill

To implement this IP Core to user circuit, technical skill about EDK and Microblaze are required. And general knowledge about high-speed interface standard are also recommended. Moreover knowledge of the protocol specification of USB3.0 standard for hardware debugging and USB device specification for software development and debugging are required.

Ordering Information

This product is available directly from Design Gateway company. Please contact Design Gateway for pricing and additional information about this product using the contact information on the front page of this datasheet.

History

Revision	Date	Description
1.0	May-20-2011	Release 1st English version
1.1E	04-March-2015	Fixed some description
1.2E	09-Mar-2015	Added O_LAN_POL and removed I_DS_PORT (Table2 page6)
1.3E	22-Apr-2015	Added some feature in IP-core register
1.4E	13-May-2015	Add 7-series device support