# TB-6S-LX150T-IMG2
# TB-FMCL-USB3.0
# Reference Design Guide

Rev.1.10

Revision History

| Version | Date | Description | Publisher |
|---|---|---|---|
| Rev.1.00 | 2012/06/21 | Initial release | Yoshioka |
| Rev.1.10 | 2012/12/20 | Supported by Cypress FX3 SDK for Windows ver.1.2.1 | Kiguchi |
| | | | |
| | | | |

Please read this software license agreement carefully before downloading or using the Software. By downloading the Software, you are agreeing to be bound by the terms of this software license. If you do not agree to the terms of this software license, you may not download, install or use the Software.

## Software License Agreement

Tokyo Electron Device Limited (“Licensor”) and Customer (“Licensee”) agree to the terms of this software license as follows:

1. Licensor grants Licensee a nonexclusive license to use, copy, modify and distribute the Software solely for development, production and sell of Licensee’s own products.
2. Licensee acknowledges that all title to the copyright and all other intellectual property rights in and to the Software, its accompanying documentation and any copy made by Licensee are the exclusive property of and remain with Licensor.
3. The Software is provided AS IS and without any warranty of any kind. In no event will Licensor be liable to Licensee for any software problems (no obligation to render technical support), infringement of any intellectual property rights, or any direct, indirect, consequential, incidental damages or loss of profit arising out of the use of this Software by Licensee or other third parties.
4. (1)Licensee shall not disclose any confidential information relating to Licensor’s technical and marketing information on the Software (hereafter referred to as “Confidential Information”). Licensee shall not disclose the Confidential Information even within its organization except for authorized employees and the board members.

   (2)The Confidential Information does not include information which (a) is within the public domain, (b) is already known to Licensee before the time it receives the same from Licensor or disclosed by a third party as a matter of right and (c) is independently developed by Licensee.

# Table of Contents

# List of Figures

# List of Tables

# 1. Overview

This documentation describes the reference design of TB-FMCL-USB30 on TB-6S-LX150T-IMG2 platform. The reference design is structured by FPGA, Firmware and Application software.
Also, this documentation suggested how to design about USB3.0 interface.

**Related documents:**
All documents relating to this board can be downloaded from our website. Please see attached paper on the products.

# 2. Reference Design Environment

## 2.1. FPGA Design Environment

**Table 2-1 FPGA design Environment**

| Item | Details | Note |
|---|---|---|
| OS | WindowsXP Professional SP3 | - |
| CPU | Intel Pentium 3.40GHz or Higher | - |
| RAM | 2.0GB or Higher | - |
| Design Language | Verilog-HDL | - |
| Simulation Tool | ModsimSE6.5a | - |
| Synthesis Tool | ISE Foundation (ISE13.2) | - |

**Table 2-2 Platform board Environment**

| Item | | Details | Note |
|---|---|---|---|
| FPGA Device | | XC6SLX150T-3FGG900 | - |
| Logic Frequency | | GPIF II I/F 100MHz, AXI I/F 200MHz | - |
| FPGA IO Pins | | 64 / 540 | - |
| FPGA Internal Resources | BRAM | 10 / 268 | - |
| | Registers | 2,276 / 184,304 | |
| | LUT | 1,797 / 92,152 | - |

## 2.2. Software Design Environment

**Table 2-3 Software design Environment**

| Item | Details | Note |
|---|---|---|
| OS | WindowsXP Professional SP3 | - |
| CPU | Intel Pentium 3.40GHz or Higher | - |
| RAM | 2.0GB or Higher | - |
| Design Language | C Language | - |
| Application design Tool | Microsoft Visual Studio 2008 (GUI) | - |
| Library | CyAPI.lib *1 | - |
| Driver | cyusb3.sys *1 | - |
| Firmware Design Tool | Eclipse IDE / ARM GCC / FX3 SDK *2 | - |

*1) Based on Cypress FX3 SDK for Windows ver.1.21
*2) It is included on Cypress FX3 SDK for Windows ver.1.21

### 2.3. Operation Environment

**Table 2-4 Operation Environment**

| Item | Detail | Note |
|---|---|---|
| OS | WindowsXP Professional SP3 32bit<br>Windows7 Professional SP1 32bit<br>Windows7 Professional SP1 64bit | - |
| CPU | Intel(R) Core(TM) i5-2450M CPU @2.50GHz | - |
| RAM | 4.0GB | - |
| Boards | [USB3.0 Interface board]<br>TB-FMCL-USB30<br>[FPGA Evaluation Platform]<br>TB-6S-LX150T-IMG2 | - |

### 2.4. Reference Document of USB3.0 device

Please refer to following documentations with this reference design.

1) EZ-USB FX3 Software Development Kit: Download from the Cypress Web page
2) Getting Started with FX3 SDK.pdf: It is included in Cypress FX3 SDK for Windows ver.1.2.1
3) FX3 Programing Manual.pdf: It is included in Cypress FX3 SDK for Windows ver.1.2.1
4) FX3 API guide.pdf: It is included in Cypress FX3 SDK for Windows ver.1.2.1

## 3. Entire Structure

Following figure shows the system block diagram of the reference design.

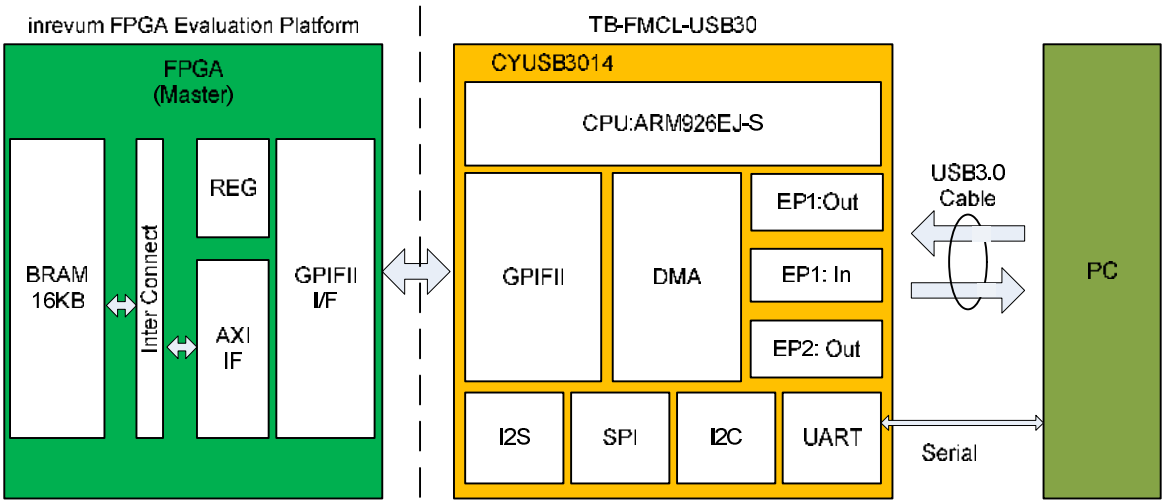



**Table 3-1 System Block Diagram**

The reference design has three design blocks

FPGA(Master): Master controller into FPGA of TB-6S-LX150T-IMG2(Spartan6 LX150T).
GPIPII I/F is supported SlaveFIFO data transfer.

USB Control: CYUSB3012 is mounted on TB-FMCL-USB30.

Host PC: Application, Library and Driver software into Host PC

In the next session describe more detail functions.

# 4. FPGA design

## 4.1. Overview of FPGA design

FPGA circuit decodes address from external USB host and access to internal memory area.
Following Figure shows a block diagram of FPGA inside circuit.

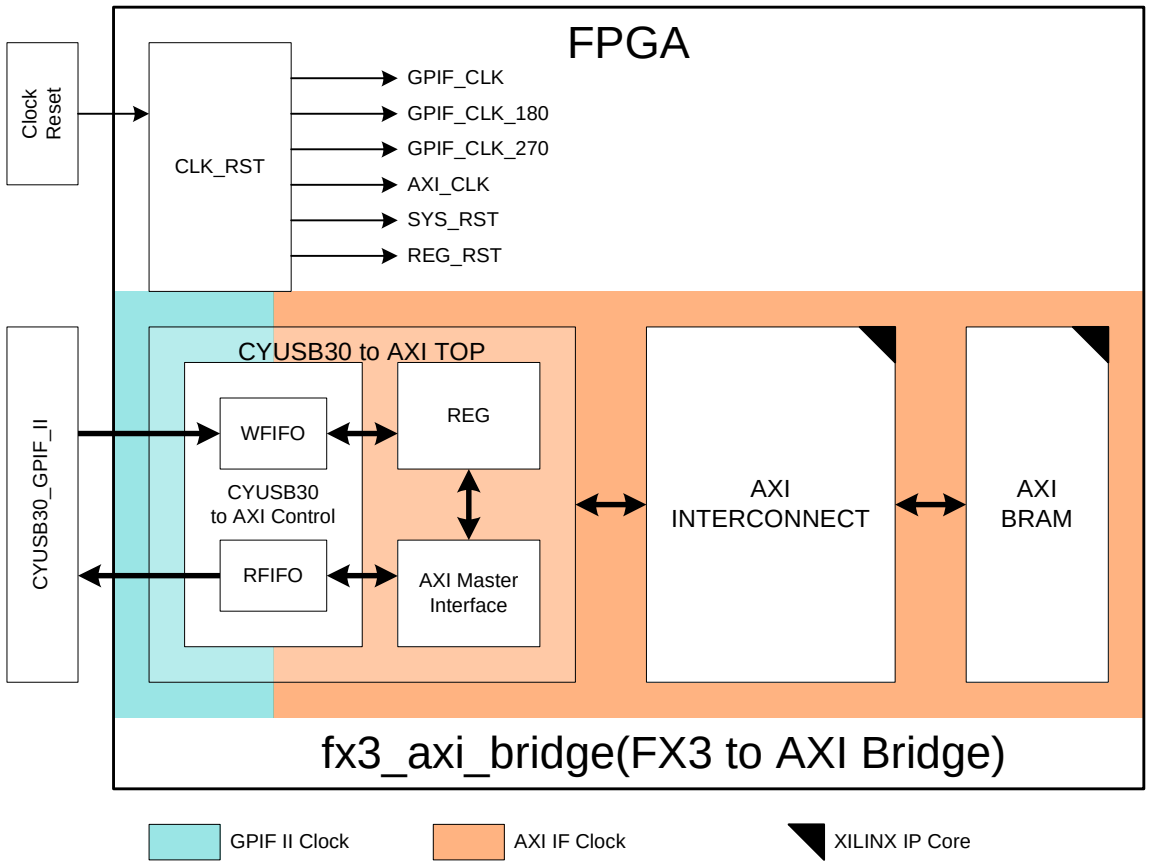



**Table 4-1 FPGA Block Diagram**

## 4.2. FPGA Pin Assign

Following Table shows a pin assign of FPGA.

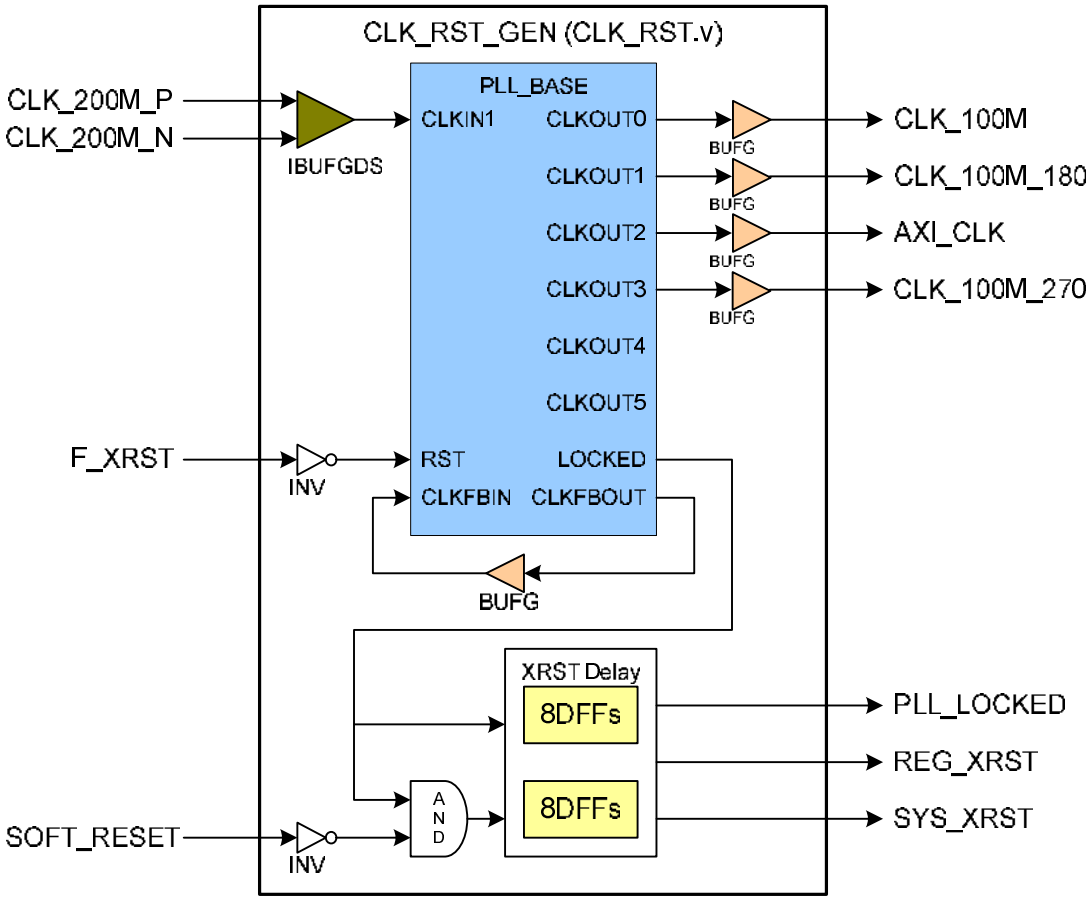
**Table 4-2 FPGA Pin Assign**

| Signal name | FX3 Pin Name | IO | Pins | Active | Initial | Function |
|---|---|---|---|---|---|---|
| Clock & Reset | | | | | | |
| CLK200M_P | - | I | 1 | ↑ | - | Differential clock 200MHz |
| CLK200M_N | - | I | 1 | ↓ | - | Differential clock 200MHz |
| FPGA_XRST | - | I | 1 | L | - | Reset signal |
| LED | - | O | 1 | H | 0 | Debug indication signal |

| GPIF II I/F | | | | | | |
|---|---|---|---|---|---|---|
| RESET_N | RESET_N_FMC | O | 1 | L | - | FX3 chip reset |
| SL_PCLK | PCLK | O | 1 | ↑ | - | Slave FIFO clock |
| SLCS_n | CTL0 | O | 1 | L | 1 | Slave FIFO chip select |
| FIFO_ADDR[1] | CTL11 | O | 1 | L | 0 | Channels address |
| FIFO_ADDR[0] | CTL12 | O | 1 | L | 0 | Channels address |
| SLRD_n | CTL3 | O | 1 | L | 1 | Read enable |
| SLOE_n | CTL2 | O | 1 | L | 1 | Output enable |
| SLWR_n | CTL1 | O | 1 | L | 1 | Write enable |
| PKTEND_n | CTL7 | O | 1 | L | 1 | Package end |
| DQ[0] | DATA0 | IO | 1 | - | Z | Data-bus |
| DQ[1] | DATA1 | IO | 1 | - | Z | Data-bus |
| DQ[2] | DATA2 | IO | 1 | - | Z | Data-bus |
| DQ[3] | DATA3 | IO | 1 | - | Z | Data-bus |
| DQ[4] | DATA4 | IO | 1 | - | Z | Data-bus |
| DQ[5] | DATA5 | IO | 1 | - | Z | Data-bus |
| DQ[6] | DATA6 | IO | 1 | - | Z | Data-bus |
| DQ[7] | DATA7 | IO | 1 | - | Z | Data-bus |
| DQ[8] | DATA8 | IO | 1 | - | Z | Data-bus |
| DQ[9] | DATA9 | IO | 1 | - | Z | Data-bus |
| DQ[10] | DATA10 | IO | 1 | - | Z | Data-bus |
| DQ[11] | DATA11 | IO | 1 | - | Z | Data-bus |
| DQ[12] | DATA12 | IO | 1 | - | Z | Data-bus |
| DQ[13] | DATA13 | IO | 1 | - | Z | Data-bus |
| DQ[14] | DATA14 | IO | 1 | - | Z | Data-bus |
| DQ[15] | DATA15 | IO | 1 | - | Z | Data-bus |
| DQ[16] | DATA16 | IO | 1 | - | Z | Data-bus |
| DQ[17] | DATA17 | IO | 1 | - | Z | Data-bus |
| DQ[18] | DATA18 | IO | 1 | - | Z | Data-bus |
| DQ[19] | DATA19 | IO | 1 | - | Z | Data-bus |
| DQ[20] | DATA20 | IO | 1 | - | Z | Data-bus |
| DQ[21] | DATA21 | IO | 1 | - | Z | Data-bus |
| DQ[22] | DATA22 | IO | 1 | - | Z | Data-bus |
| DQ[23] | DATA23 | IO | 1 | - | Z | Data-bus |
| DQ[24] | DATA24 | IO | 1 | - | Z | Data-bus |
| DQ[25] | DATA25 | IO | 1 | - | Z | Data-bus |
| DQ[26] | DATA26 | IO | 1 | - | Z | Data-bus |
| DQ[27] | DATA27 | IO | 1 | - | Z | Data-bus |
| DQ[28] | DATA28 | IO | 1 | - | Z | Data-bus |
| DQ[29] | DATA29 | IO | 1 | - | Z | Data-bus |
| DQ[30] | DATA30 | IO | 1 | - | Z | Data-bus |
| DQ[31] | DATA31 | IO | 1 | - | Z | Data-bus |
| FLGA_n | CTL4 | I | 1 | L | 1 | Slave FIFO status flags of current channel |
| FLGB_n | CTL5 | I | 1 | L | 1 | Slave FIFO status flags of current channel |

## 4.3. Detail Functions

### 4.3.1. Clock Tree




F_XRST is active High and it is connected to SW9.
REF_XRST is targeting to REG_IF module and CYUSB3014.
SYS_XRST is targeting to all component and logic without CLK_RST_GEN module.

### 4.3.2. CYUSB30_AXI_TOP function

Following figure shows block diagram of USB GPIF.

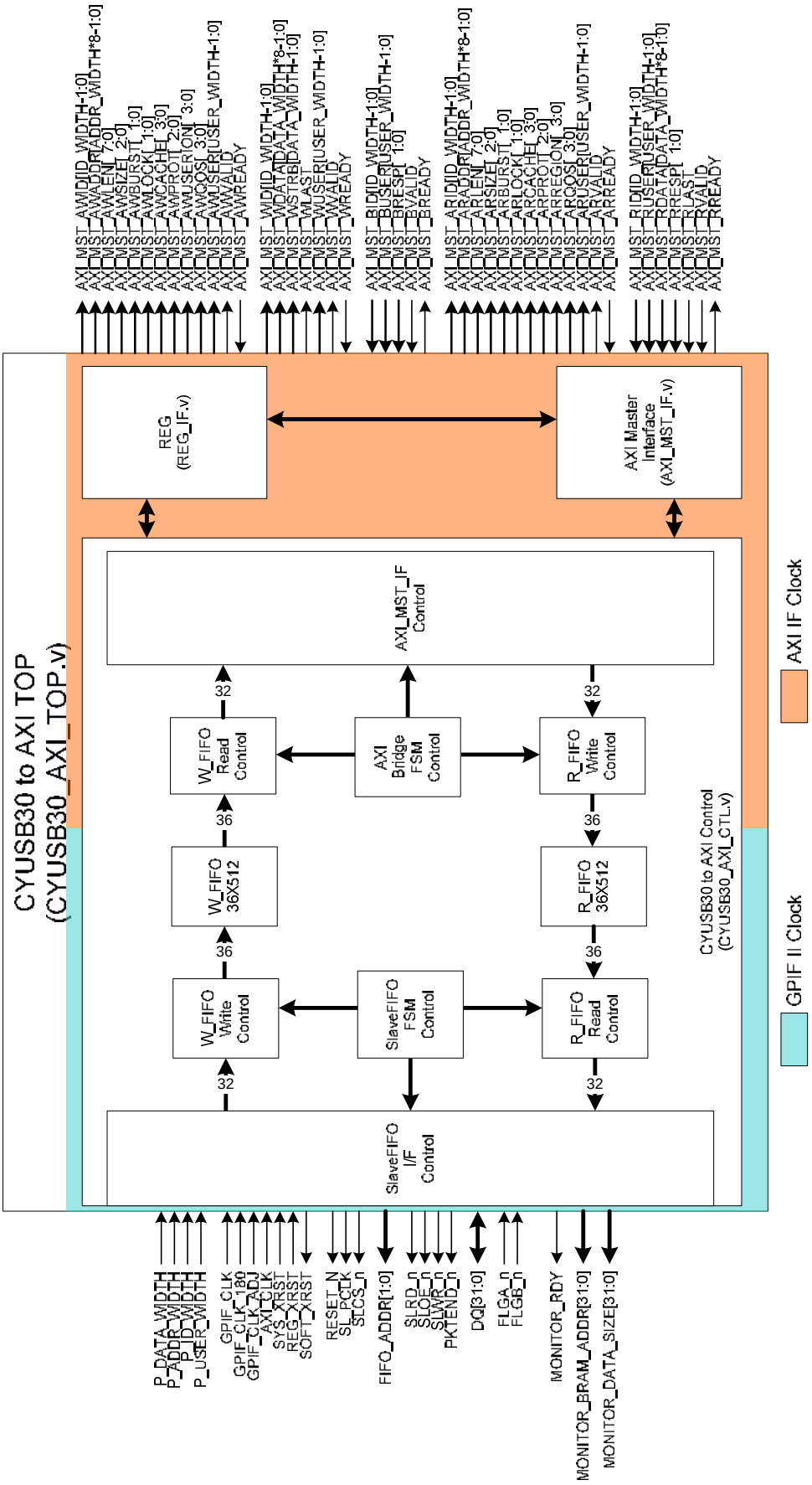



**Table 4-3 CYUSB30 AXI TOP Block Diagram**

### 4.3.3. CYUSB30 AXI TOP internal interface

**Table 4-4 CYUSB30 AXI TOP internal interface**

| Signal Name | I/O | Pins | Active | Initial | Function |
|---|---|---|---|---|---|
| System Parameter | | | | | |
| DATA_WIDTH | I | - | - | 4 | Bridge Data Bus bytes width |
| ADDR_WIDTH | I | - | - | 4 | Bridge Address Bus bytes width |
| ID_WIDTH | I | - | - | 4 | AXI I/F ID Width |
| USER_WIDTH | I | - | - | 4 | AXI I/F USER Width |
| Clock & Reset | | | | | |
| GPIF_CLK | I | 1 | ↑ | 0 | GPIF clock, 0 phase 100MHz(max) |
| GPIF_CLK_180 | I | 1 | ↑ | 1 | GPIF clock, 180 phase 100MHz(max) |
| GPIF_CLK_ADJ | I | 1 | ↑ | 1 | GPIF clock, 270 phase 100MHz(max) |
| AXI_CLK | I | 1 | ↑ | 0 | AXI clock 200MHz(max) |
| SYS_XRST | I | 1 | 0 | 1 | System sync. reset(low active) |
| REG_XRST | I | 1 | 0 | 1 | REG sync. reset(low active) |
| SOFT_RESET | O | 1 | 0 | 1 | System soft reset output |
| GPIF_II I/F | | | | | |
| RESET_N | O | 1 | L | - | FX3 chip reset |
| SL_PCLK | O | 1 | ↑ | - | Slave FIFO clock |
| SLCS_n | O | 1 | L | 1 | Slave FIFO chip select |
| FIFO_ADDR | O | 2 | L | All 0 | Channels address |
| SLRD_n | O | 1 | L | 1 | Read enable |
| SLOE_n | O | 1 | L | 1 | Output enable |
| SLWR_n | O | 1 | L | 1 | Write enable |
| PKTEND_n | O | 1 | L | 1 | Package end |
| DQ | IO | 32 | - | All Z | Data-bus |
| FLGA_n | I | 1 | L | 1 | Slave FIFO status flags of current Channel |
| FLGB_n | I | 1 | L | 1 | Slave FIFO status flags of current Channel |
| USER MONITOR I/F | | | | | |
| MONITOR_RDY | O | 1 | 1 | 0 | Monitor ready(active high) |
| MONITOR_BRAM_ADDR | O | 32 | - | - | Monitor BRAM start address(write) |
| MONITOR_DATA_SIZE | O | 32 | - | - | Monitor BRAM data size(write) |
| AXI IF | | | | | |
| AXI_MST_AWID | O | ID_WIDTH | - | All 0 | Write Address Channel Transaction ID |
| AXI_MST_AWADDR | O | ADDR_WIDTH*8 | - | All 0 | Write Address Channel Address |
| AXI_MST_AWLEN | O | 8 | - | All 0 | Write Address Channel Burst Length code |
| AXI_MST_AWSIZE | O | 3 | - | All 0 | Write Address Channel Transfer Size code |
| AXI_MST_AWBURST | O | 2 | - | All 0 | Write Address Channel Burst Type |
| AXI_MST_AWLOCK | O | 2 | - | All 0 | Write Address Channel Atomic Access Type |
| AXI_MST_AWCACHE | O | 4 | - | All 0 | Write Address Channel Cache Characteristics |
| AXI_MST_AWPROT | O | 3 | - | All 0 | Write Address Channel Protection Bits |
| AXI_MST_AWUSERION | O | 4 | - | All 0 | Write Address Channel address region index |
| AXI_MST_AWQOS | O | 4 | - | All 0 | Write Address Channel Quality of Service |
| AXI_MST_AWUSER | O | USER_WIDTH | - | All 0 | Write Address Channel USER signal |

| | | | | | |
|---|---|---|---|---|---|
| AXI_MST_AWVALID | O | 1 | 1 | 0 | Write Address Channel Valid |
| AXI_MST_AWREADY | I | 1 | 1 | 1 | Write Address Channel Ready |
| AXI_MST_WID | O | ID_WIDTH | - | All 0 | Write Data Channel Transaction ID |
| AXI_MST_WDATA | O | DATA_WIDTH*8 | - | All 0 | Write Data Channel Data |
| AXI_MST_WSTRB | O | DATA_WIDTH | - | All 0 | Write Data Channel Data Byte Strobes |
| AXI_MST_WLAST | O | 1 | 1 | - | Write Data Channel Last Data Beat |
| AXI_MST_WUSER | O | USER_WIDTH | - | All 0 | Write Data Channel USER signal |
| AXI_MST_WVALID | O | 1 | 1 | 0 | Write Data Channel Valid |
| AXI_MST_WREADY | I | 1 | 1 | 1 | Write Data Channel Ready |
| AXI_MST_BID | I | ID_WIDTH | - | All 0 | Write Response Channel Transaction ID |
| AXI_MST_BUSER | I | USER_WIDTH | - | All 0 | Write Response Channel USER signal |
| AXI_MST_BRESP | I | 2 | - | All 0 | Write Response Channel Response Code |
| AXI_MST_BVALID | I | 1 | 1 | 0 | Write Response Channel Valid |
| AXI_MST_BREADY | O | 1 | 1 | 0 | Write Response Channel Ready |
| AXI_MST_ARID | O | ID_WIDTH | - | All 0 | Read Address Channel Transaction ID |
| AXI_MST_ARADDR | O | ADDR_WIDTH*8 | - | All 0 | Read Address Channel Address |
| AXI_MST_ARLEN | O | 8 | - | All 0 | Read Address Channel Burst Length code |
| AXI_MST_ARSIZE | O | 3 | - | All 0 | Read Address Channel Transfer Size code |
| AXI_MST_ARBURST | O | 2 | - | All 0 | Read Address Channel Burst Type |
| AXI_MST_ARLOCK | O | 2 | - | All 0 | Read Address Channel Atomic Access Type |
| AXI_MST_ARCACHE | O | 4 | - | All 0 | Read Address Channel Cache Characteristics |
| AXI_MST_ARPROT | O | 3 | - | All 0 | Read Address Channel Protection Bits |
| AXI_MST_ARREGION | O | 4 | - | All 0 | Read Address Channel address region index |
| AXI_MST_ARQOS | O | 4 | - | All 0 | Read Address Channel Quality of Service |
| AXI_MST_ARUSER | O | USER_WIDTH | - | All 0 | Read Address Channel USER signal |
| AXI_MST_ARVALID | O | 1 | 1 | 0 | Read Address Channel Valid |
| AXI_MST_ARREADY | I | 1 | 1 | 0 | Read Address Channel Ready |
| AXI_MST_RID | I | ID_WIDTH | - | All 0 | Read Data Channel Transaction ID |
| AXI_MST_RUSER | I | USER_WIDTH | - | All 0 | Read Data Channel USER signal |
| AXI_MST_RDATA | I | DATA_WIDTH*8 | - | All 0 | Read Data Channel Data |
| AXI_MST_RRESP | I | 2 | - | All 0 | Read Data Channel Response Code |
| AXI_MST_RLAST | I | 1 | 1 | 0 | Read Data Channel Last Data Beat |
| AXI_MST_RVALID | I | 1 | 1 | 0 | Read Data Channel Valid |
| AXI_MST_RREADY | O | 1 | 1 | 0 | Read Data Channel Ready |

4.3.4. USB GPIF timing chart

Following figure shows a timing chart of USB GPIF.

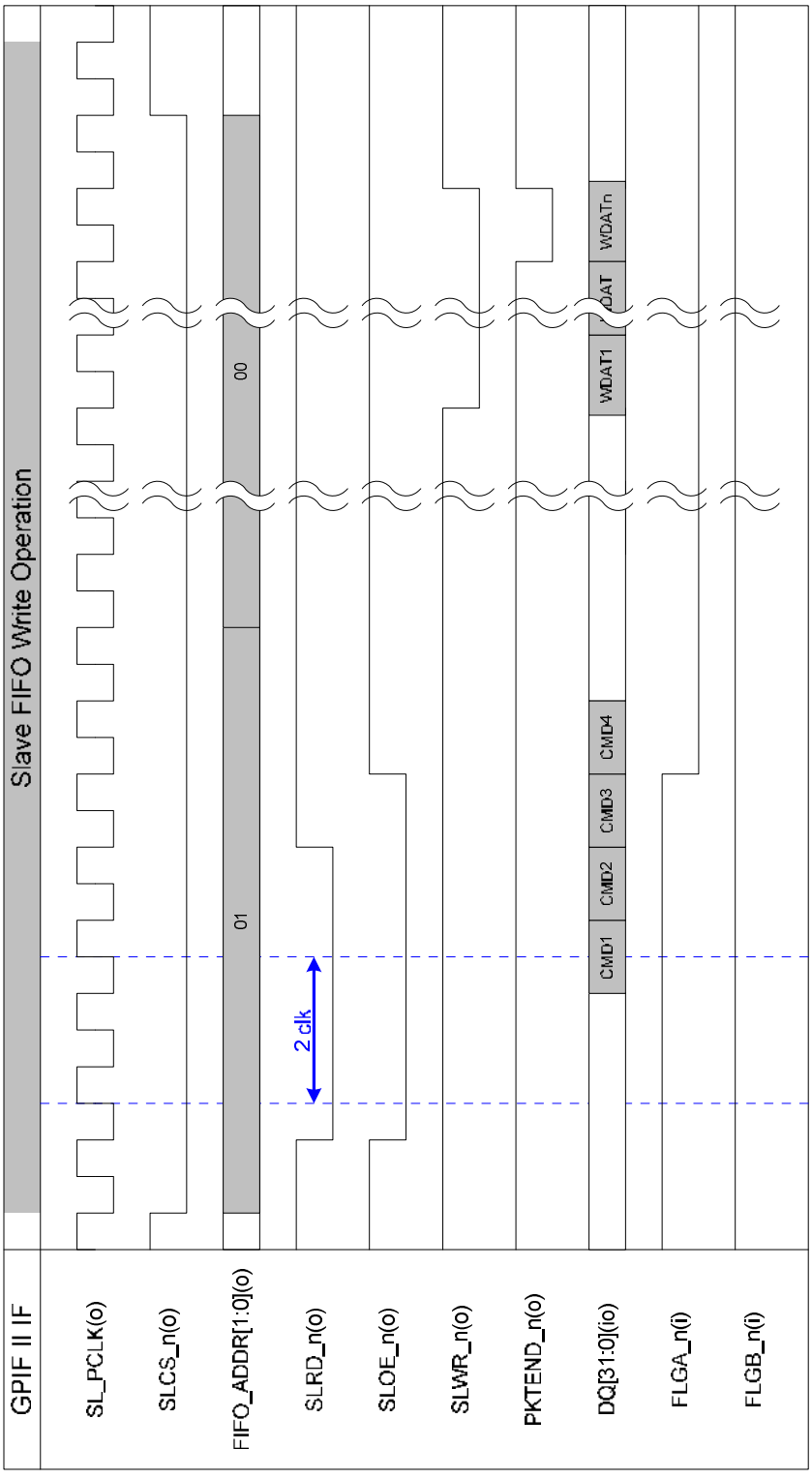



**Table 4-5 Write Access Timing Chart**

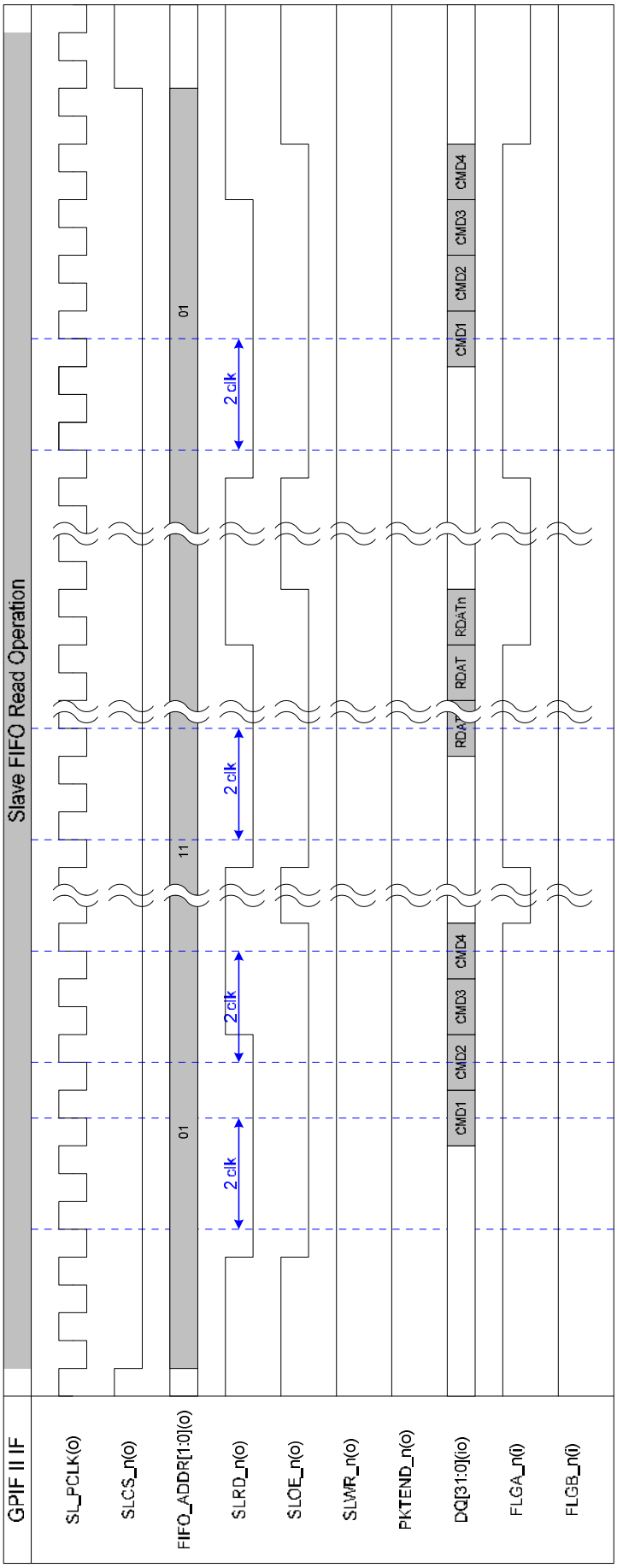



**Table 4-6 Read Access Timing Chart**

## 4.4. Register Map and Functions

### 4.4.1. Register Map and Functions

Following table shows a register map of FPGA design,

| Address | Register Name | R/W | Functions | Initial |
|---|---|---|---|---|
| System | | | | |
| 0x00 | エラー! 参照元が見つかりません。 | R | FPGA revision | 0x2012_0221 |
| 0x04 | SOFT_RESET | R/W | Soft reset | 0x0000_0000 |
| 0x08 | SYS_TEST | R/W | Debug | 0x0000_0000 |
| 0x0C | Reserved | - | - | - |
| AXI I/F | | | | |
| 0x10 | P_AXI_AW_ADDR | R/W | Address of AW channel | 0x0000_0000 |
| 0x14 | P_AXI_AW_P1 | R/W | ID,LEN,BURST,SIZE of AW channel | 0x1201_0000 |
| 0x18 | P_AXI_AW_P2 | R/W | LOCK,CACHE,PROT, REGION of AW channel | 0x0000_0000 |
| 0x1C | P_AXI_AW_P3 | R/W | QOS, USER of AW channel | 0x0000_0000 |
| 0x20 | P_AXI_AW_CTL | R/W | AW channel control | 0x0001_0000 |
| 0x24 | P_AXI_W_DATA | R/W | Data of W channel | 0x0000_0000 |
| 0x28 | P_AXI_W_P | R/W | ID,USER of W channel | 0x0000_0000 |
| 0x2C | P_AXI_W_CTL | R/W | W channel control | 0x0001_0000 |
| 0x30 | P_AXI_B_P | R | ID,USER of B channel | 0x0000_0000 |
| 0x34 | P_AXI_B_CTL | R/W | B channel control | 0x0000_0000 |
| 0x38 | P_AXI_AR_ADDR | R/W | Address of AR channel | 0x0000_0000 |
| 0x3C | P_AXI_AR_P1 | R/W | ID,LEN,BURST,SIZE of AR channel | 0x1201_0000 |
| 0x40 | P_AXI_AR_P2 | R/W | LOCK,CACHE,PROT, REGION of AR channel | 0x0000_0000 |
| 0x44 | P_AXI_AR_P3 | R/W | QOS, USER of AR channel | 0x0000_0000 |
| 0x48 | P_AXI_AR_CTL | R/W | AR channel control | 0x0001_0000 |
| 0x4C | P_AXI_R_DATA | R | Data of R channel | 0x0000_0000 |
| 0x50 | P_AXI_R_P | R/W | ID,USER of R channel | 0x0000_0000 |
| 0x54 | P_AXI_R_CTL | R/W | R channel control | 0x0000_0000 |
| 0x58 ~ 0xFF | Reserved | - | - | - |

Register Type

R: Read Only

R/W: Read and Write (some bit is R/C: Write 1 to Clear)

Address 0x58 to 0xFF: Reserved.

### 4.4.2. FPGA_RED register

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Data | Year | | | | | | | | | | | | | | | | Month | | | | | | | | Date | | | | | | | | |
| Type | R | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Init | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |

| | | | |
|---|---|---|---|
| Address | 0x00 | | |
| Initial | 0x2012_0221 | | |
| Function | FPGA Revision Register. It shows the generation date of FPGA bit file. | | |
| Data | Name | Bit | Function |
| | Year | 31:16 | Year |
| | Month | 15:8 | Month |
| | Date | 7:0 | Data |
| Note | | | |

### 4.4.3. SOFT_RESET register

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Data | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | SOFT_RESET |
| Type | R | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | R/W |
| Init | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | |
|---|---|---|---|
| Address | 0x04 | | |
| Initial | 0x0000_0000 | | |
| Function | FPGA logic reset register | | |
| Data | Name | Bit | Function |
| | - | 31:1 | |
| | SOFT_RESET | 0 | 0: Reset release<br>1: Reset set(not reset FX3 Chip) |
| Note | | | |

### 4.4.4. SYS_TEST register

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Data | | | | | | | | | | | | | | | | | | | | | | | | | | | | GPIF_RD_ERR_VALID | GPIF_RD_ERR | GPIF_WR_ERR_VALID | GPIF_WR_ERR | REG_ACCESS_EN |
| Type | R | | | | | | | | | | | | | | | | | | | | | | | | | | | R/C | R/C | R/C | R/C | R/W |
| Init | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | |
|---|---|---|---|
| Address | 0x08 | | |
| Initial | 0x0000_0000 | | |
| Function | GPIF Error control and Accessing enable of AXI-BRAM | | |
| Data | Name | Bit | Function |
| | - | 31:5 | |
| | GPIF_RD_ERR_VALID | 4 | 0: GPIF_RD_ERR un-valid<br>1: GPIF_RD_ERR valid |
| | GPIF_RD_ERR | 3 | 0: Read operation has no errors<br>1: Read operation has errors |
| | GPIF_WR_ERR_VALID | 2 | 0: GPIF_WR_ERR un-valid<br>1: GPIF_WR_ERR valid |
| | GPIF_WR_ERR | 1 | 0: Write operation has no errors<br>1: Write operation has errors |
| | REG_ACCESS_EN | 0 | 0: REG R/W AXI-BRAM access un-enable<br>1: REG R/W AXI-BRAM access enable |
| Note | | | |

### 4.4.5. AXI_QW_ADDR register

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Data | USER_AWADDR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Type | R/W | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Init | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | |
|---|---|---|---|
| Address | 0x10 | | |
| Initial | 0x0000_0000 | | |
| Function | Write address register of AXI-AW(AXI Master). | | |
| Data | Name | Bit | Function |
| | USER_AWADDR | 31:0 | Write data address input |
| Note | Only used for register access for axi interface | | |

### 4.4.6. P_AXI_AW_P1 register

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Data | | | USER_AWBURST | | | USER_AWSIZE | | | USER_AWLEN | | | | | | | | | | | | | | | | | | | | USER_AWID | | | |
| Type | R | | R/W | | R | R/W | | | R/W | | | | | | | | R | | | | | | | | | | | | R/W | | | |
| Init | 0 | 0 | 0 | 1 | 0 | 0 | **1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | |
|---|---|
| Address | 0x14 |
| Initial | 0x1201_0000 |
| Function | Parameter of AXI-AW P1 |

Data

| Name | Bit | Function |
|---|---|---|
| - | 31:30 | |
| USER_AWBURST | 29:28 | 00: FIXED, 01: INCR, 10: WRAP, 11: reserved |
| - | 27 | |
| USER_AWSIZE | 26:24 | AW channel SIZE |
| USER_AWLEN | 23:16 | AW channel LEN |
| - | 15:4 | |
| USER_AWID | 3:0 | ID tag |

Note

### 4.4.7. P_AXI_AW_P2 register

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Data | | | | | | | | | | | | | USER_AWREGION | | | | | | | | | USER_AWPROT | | | USER_AWCACHE | | | | | | USER_AWLOCK | |
| Type | R | | | | | | | | | | | | R/W | | | | R | | | | | R/W | | | R/W | | | | R | | R/W | |
| Init | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | |
|---|---|
| Address | 0x18 |
| Initial | 0x0000_0000 |
| Function | Parameter of AXI-AW P2 |

Data

| Name | Bit | Function |
|---|---|---|
| - | 31:20 | |
| USER_AWREGION | 19:16 | AW channel LEN REGION |
| - | 15:11 | |
| USER_AWPROT | 10:8 | AW channel PROT |
| USER_AWCACHE | 7:4 | AW channel CACHE |
| - | 3:2 | |
| USER_AWLOCK | 1:0 | AW channel LOCK |

Note

### 4.4.8. P_AXI_AW_P3 register

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Data | | | | | | | | | | | | | USER_AWUSER | | | | | | | | | | | | | | | | USER_AWQOS | | | |
| Type | R | | | | | | | | | | | | R/W | | | | R | | | | | | | | | | | | R/W | | | |
| Init | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | |
|---|---|
| Address | 0x1C |
| Initial | 0x0000_0000 |
| Function | Parameter of AXI-AW P3 |

| | Name | Bit | Function |
|---|---|---|---|
| Data | - | 31:20 | |
| | USER_AWUSER | 19:16 | AW channel USER |
| | - | 15:4 | |
| | USER_AWQOS | 3:0 | AW channel QOS |
| Note | | | |

### 4.4.9. P_AXI_AW_CTL register

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Data | | | | | | | | | | | | | | | | USER_AWADDR_NEXT_EN | | | | | | | | | | | | | | | | USER_AWVALID |
| Type | R | | | | | | | | | | | | | | | R/C | R | | | | | | | | | | | | | | | R/W |
| Init | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | |
|---|---|
| Address | 0x20 |
| Initial | 0x0001_0000 |
| Function | Control register of AXI-AW |

| | Name | Bit | Function |
|---|---|---|---|
| Data | - | 31:17 | |
| | USER_AWADDR_NEXT_EN | 16 | 0: Write address of AW channel un-enable<br>1: Write address of AW channel enable |
| | - | 15:1 | |
| | USER_AWVALID | 0 | 0: Write address of AW channel un-valid<br>1: Write address of AW channel valid |
| Note | Only used for register access for axi interface | | |

### 4.4.10. P_AXI_W_DATA register

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Data | P_AXI_W_DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Type | R/W | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Init | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | |
|---|---|---|---|
| Address | 0x24 | | |
| Initial | 0x0000_0000 | | |
| Function | Data register of AXI-W(AXI Master) | | |
| Data | Name | Bit | Function |
| | USER__WDATA | 31:0 | Write data input |
| Note | Only used for register access for axi interface | | |

### 4.4.11. P_AXI_W_P register

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Data | | | | | | | | | | | | | USER_WUSER | | | | | | | | | | | | | | | | USER_WID | | | |
| Type | R | | | | | | | | | | | | R/W | | | | R | | | | | | | | | | | | R/W | | | |
| Init | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | |
|---|---|---|---|
| Address | 0x28 | | |
| Initial | 0x0000_0000 | | |
| Function | Parameter register of AXI-W | | |
| Data | Name | Bit | Function |
| | - | 31:20 | |
| | USER_WUSER | 19:16 | W channel USER |
| | - | 15:4 | |
| | USER_WID | 3:0 | W channel ID |
| Note | | | |

### 4.4.12. P_AXI_W_CTL register

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Data | | | | | | | | | | | | | | | | USER_WDATA_NEXT_EN | USER_WSTRB | | | | | | | | | | | | | | USER_WLAST | USER_WVALID |
| Type | R | | | | | | | | | | | | | | | R/C | R/W | | | | | | | | R | | | | | | R/W | R/W |
| Init | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | |
|---|---|---|---|
| Address | 0x2C | | |
| Initial | 0x0001_0000 | | |
| Function | Control register of AXI-W | | |
| Data | Name | Bit | Function |
| | - | 31:17 | |
| | USER_WDATA_NEXT_EN | 16 | 0: Write data of W channel un-enable<br>1: Write data of W channel enable |
| | USER_WSTRB | 15:8 | Write data byte enable |
| | - | 7:2 | |
| | USER_WLAST | 1 | 0: Write data of W channel is not the last one<br>1: Write data of W channel is the last one |
| | USER_WVALID | 0 | 0: Write data of W channel un-valid<br>1: Write data of W channel valid |
| Note | Only used for register access for axi interface | | |

### 4.4.13. P_AXI_B_P register

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Data | | | | | | | | | | | | | USER_BUSER | | | | | | | | | | | | | | | | USER_BID | | | |
| Type | R | | | | | | | | | | | | R | | | | R | | | | | | | | | | | | R | | | |
| Init | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | |
|---|---|
| Address | 0x30 |
| Initial | 0x0000_0000 |
| Function | Parameter register of AXI-B(Write access acknowledge) |

| | Name | Bit | Function |
|---|---|---|---|
| Data | | 31:20 | |
| | USER_BUSER | 19:16 | B channel USER |
| | - | 15:4 | |
| | USER_BID | 3:0 | B channel ID |
| Note | | | |

### 4.4.14. P_AXI_B_CTL register

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Data | | | | | | | | | | | | | | | USER_BRESP | | | | | | | | | USER_BVALID | | | | | | | | USER_BREADY |
| Type | R | | | | | | | | | | | | | | R | | R | | | | | | | R/C | R | | | | | | | R/W |
| Init | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | |
|---|---|
| Address | 0x34 |
| Initial | 0x0000_0000 |
| Function | Control register of AXI-B |

| | Name | Bit | Function |
|---|---|---|---|
| Data | - | 31:18 | |
| | USER_BRESP | 17:16 | 00: OKAY, 01: EXOKAY, 10: SLVERR, 11: DECERR |
| | - | 15:9 | |
| | USER_BVALID | 8 | 0: USER_BRESP un-valid<br>1: USER_BRESP valid |
| | - | 7:1 | |
| | USER_BREADY | 0 | 0: GPIF is not ready for receive USER_BRESP<br>1: GPIF is ready for receive USER_BRESP |
| Note | | | |

### 4.4.15. P_AXI_AR_ADDR

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Data | USER_ARADDR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Type | R/W | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Init | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | |
|---|---|---|---|
| Address | 0x38 | | |
| Initial | 0x0000_0000 | | |
| Function | Read address register of AXIAR(AXI Master) | | |
| Data | Name | Bit | Function |
| | USER_ARADDR | 31:0 | Read data address input |
| Note | Only used for register access for axi interface | | |

### 4.4.16. P_AXI_AR_P1 register

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Data | | | USER_ARBURST | | | USER_ARSIZE | | | USER_ARLEN | | | | | | | | | | | | | | | | | | | | USER_ARID | | | |
| Type | R | | R/W | | R | R/W | | | R/W | | | | | | | | R | | | | | | | | | | | | R/W | | | |
| Init | 0 | 0 | 0 | 1 | 0 | 0 | **1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | |
|---|---|---|---|
| Address | 0x3C | | |
| Initial | 0x1201_0000 | | |
| Function | Parameter register of AXI-AR P1 | | |
| Data | Name | Bit | Function |
| | - | 31:30 | |
| | USER_ARBURST | 29:28 | 00: FIXED<br>01: INCR<br>10: WRAP<br>11: reserved |
| | - | 27 | |
| | USER_ARSIZE | 26:24 | AR channel SIZE |
| | USER_ARLEN | 23:16 | AR channel LEN |
| | - | 15:4 | |
| | USER_ARID | 3:0 | ID tag |
| Note | | | |

### 4.4.17. P_AXI_AR_P2 register

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Data | | | | | | | | | | | | | USER_ARREGION | | | | | | | | | USER_ARPROT | | | USER_ARCACHE | | | | | | USER_ARLOCK | |
| Type | R | | | | | | | | | | | | R/W | | | | R | | | | | R/W | | | R/W | | | | R | | R/W | |
| Init | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Address | 0x40 | | |
|---|---|---|---|
| Initial | 0x0000_0000 | | |
| Function | Parameter register of AXI-AR P2 | | |
| Data | Name | Bit | Function |
| | - | 31:20 | |
| | USER_ARREGION | 19:16 | AR channel LEN REGION |
| | - | 15:11 | |
| | USER_ARPROT | 10:8 | AR channel PROT |
| | USER_ARCACHE | 7:4 | AR channel CACHE |
| | - | 3:2 | |
| | USER_ARLOCK | 1:0 | AR channel LOCK |
| Note | | | |

### 4.4.18. P_AXI_AR_P3 register

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Data | | | | | | | | | | | | | USER_ARUSER | | | | | | | | | | | | | | | | USER_ARQOS | | | |
| Type | R | | | | | | | | | | | | R/W | | | | R | | | | | | | | | | | | R/W | | | |
| Init | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | |
|---|---|
| Address | 0x44 |
| Initial | 0x0000_0000 |
| Function | Parameter register of AXI-AR P3 |

| | Name | Bit | Function |
|---|---|---|---|
| Data | - | 31:20 | |
| | USER_ARUSER | 19:16 | AR channel USER |
| | - | 15:4 | |
| | USER_ARQOS | 3:0 | AR channel QOS |
| Note | | | |

### 4.4.19. P_AXI_AR_CTL register

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Data | | | | | | | | | | | | | | | | USER_ARADDR_NEXT_EN | | | | | | | | | | | | | | | | USER_ARVALID |
| Type | R | | | | | | | | | | | | | | | R/C | R | | | | | | | | | | | | | | | R/W |
| Init | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | |
|---|---|
| Address | 0x48 |
| Initial | 0x0001_0000 |
| Function | Control register of AXI-AR |

| | Name | Bit | Function |
|---|---|---|---|
| Data | - | 31:17 | |
| | USER_ARADDR_NEXT_EN | 16 | 0: Read address of AR channel un-enable<br>1: Read address of AR channel enable |
| | - | 15:1 | |
| | USER_ARVALID | 0 | 0: Read address of AR channel un-valid<br>1: Read address of AR channel valid |
| note | | | |

### 4.4.20. P_AXI_R_DATA register

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Data | USER_RDATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Type | R | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Init | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | |
|---|---|---|---|
| Address | 0x4C | | |
| Initial | 0x0000_0000 | | |
| Function | Data register of AXI-R(AXI Master) | | |
| Data | Name | Bit | Function |
| | USER__RDATA | 31:0 | Read data output |
| Note | Only used for register access for axi interface | | |

### 4.4.21. P_AXI_R_P register

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Data | | | | | | | | | | | | | USER_RUSER | | | | | | | | | | | | | | | | USER_RID | | | |
| Type | R | | | | | | | | | | | | R | | | | R | | | | | | | | | | | | R | | | |
| Init | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | |
|---|---|---|---|
| Address | 0x50 | | |
| Initial | 0x0000_0000 | | |
| Function | Parameter register of AXI-R | | |
| Data | Name | Bit | Function |
| | | 31:20 | |
| | USER_RUSER | 19:16 | R channel USER |
| | - | 15:4 | |
| | USER_RID | 3:0 | R channel ID |
| Note | | | |

### 4.4.22. P_AXI_R_CTL register

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Data | | | | | | | | | | | | | | | USER_RRESP | | | | | | | | USER_RDLAST | USER_RVALID | | | | | | | | USER_RREADY |
| Type | R | | | | | | | | | | | | | | R | | R | | | | | | R/C | R/C | R | | | | | | | R/W |
| Init | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | |
|---|---|---|---|
| Address | 0x54 | | |
| Initial | 0x0000_0000 | | |
| Function | Control register of AXI-R | | |
| Data | Name | Bit | Function |
| | - | 31:18 | |
| | USER_RRESP | 17:16 | 00: OKAY, 01: EXOKAY, 10: SLVERR, 11: DECERR |
| | - | 15:10 | |
| | USER_RDLAST | 9 | 0: USER_RRESP un-valid<br>1: USER_RRESP valid |
| | USER_RVALID | 8 | 0: R channel back data un-valid<br>1: R channel back data valid |
| | - | 7:1 | |
| | USER_RREADY | 0 | 0: GPIF is not ready for receive data<br>1: GPIF is ready for receive data |
| Note | Only used for register access for axi interface | | |

# 5. Software Design

## 5.1. Firmware Design

This section describes the firmware of ARM926EJ-S into Cypress CYUSB3014 and method of Descriptor, Vender Command and Download.

### 5.1.1. Descriptor

Following descriptor information is used in the reference design.

Note: This reference design provide descriptor for Full Speed but Full Speed is not tested.
Thus TB-FMCL-USB30 and the reference design are not support.

Device descriptor (USB2.0 Connection)

**Table 5-1 Device Descriptor (USB2.0)**

| Filed Name | Size (Byte) | Data (Hex) | Description (reference design) |
|---|---|---|---|
| Descriptor Size | 1 | 12 | Size of Descriptor |
| DescriptorType | 1 | 01 | Type of Descriptor (Device) |
| bcdUSB | 2 | 0200 | Device Version (USB2.0) |
| Device Class | 1 | 00 | Device Class |
| Device Sub-class | 1 | 00 | Device Sub Class |
| Device protocol | 1 | 00 | Device Type (Interface descriptor) |
| Maxpacket size for EP0 | 1 | 40 | EP0 Maximum Packet Size (64Byte) |
| Vendor ID | 2 | 04EC | Vender ID |
| Product ID | 2 | F000 | Product ID |
| Device release number | 2 | 0000 | Device release Version |
| Manufacture string index | 1 | 01 | Manufacture String Index “TED” |
| Product string index | 1 | 02 | Product String Index “TB-FMCL-USB30” |
| Serial number string index | 1 | 03 | Serial Number Index “110” |
| Number of configurations | 1 | 01 | Number of configurations(# of Descriptor) |

Device descriptor (USB3.0 Connection)

**Table 5-2 Device Descriptor (USB3.0)**

| Filed Name | Size (Byte) | Data (Hex) | Description (reference design) |
|---|---|---|---|
| Descriptor Size | 1 | 12 | Size of Descriptor |
| DescriptorType | 1 | 01 | Type of Descriptor (Device) |
| bcdUSB | 2 | 0300 | Device Version (USB3.0) |
| Device Class | 1 | 00 | Device Class |
| Device Sub-class | 1 | 00 | Device Sub Class |
| Device protocol | 1 | 00 | Device Type (Interface descriptor) |
| Maxpacket size for EP0 | 1 | 40 | EP0 Maximum Packet Size (64Byte) |
| Vendor ID | 2 | 04EC | Vender ID |
| Product ID | 2 | F001 | Product ID |
| Device release number | 2 | 0000 | Device release Version |
| Manufacture string index | 1 | 01 | Manufacture String Index “TED” |
| Product string index | 1 | 02 | Product String Index “TB-FMCL-USB30” |
| Serial number string index | 1 | 03 | Serial Number Index “110” |
| Number of configurations | 1 | 01 | Number of configurations(# of Descriptor) |

BOS Descriptor (Only for Super Speed)

**Table 5-3 BOS Descriptor**

| Filed Name | Size (Byte) | Data (Hex) | Description (reference design) |
|---|---|---|---|
| bLength | 1 | 05 | Size of Descriptor |
| bDescriptorType | 1 | 0D | Type of Descriptor (BOS) |
| wTotalLength | 2 | 0016 | Total Length.<br>It is included following device capability description. |
| bNumDeviceCaps | 1 | 02 | Number of Device Capability Descriptor |

Device Capability Descriptor (Only for Super Speed)
USB2.0 Expansion descriptor

**Table 5-4 Device Capability Descriptor(USB2.0 Expansion)**

| Filed Name | Size (Byte) | Data (Hex) | Description (reference design) |
|---|---|---|---|
| bLength | 1 | 05 | Size of Descriptor |
| bDescriptorType | 1 | 0D | Type of Descriptor (Device Capability) |
| bDevCapabilityType | 1 | 02 | Capability Type(USB2.0 Expansion) |
| bmAttributes | 1 | 02 | Capability Support Attributes |

SS Device Capability Descriptor

**Table 5-5 Device Capability Descriptor (SS device capability)**

| Filed Name | Size (Byte) | Data (Hex) | Description (reference design) |
|---|---|---|---|
| bLength | 1 | 0A | Size of Descriptor |
| bDescriptorType | 1 | 10 | Type of Descriptor (Device Capability) |
| bDevCapabilityType | 1 | 03 | Capability Type (SS capability) |
| bmAttributes | 1 | 00 | Capability Support Attributes |
| wSpeedsSupported | 2 | 000E | Supported transfer speed<br>bit0:LS, bit1:FS, bit2:HS, bit3: SS |
| bFuctionalitySupport | 1 | 03 | Bit number of available transfer rate |
| bU1DevExitLat | 1 | 00 | Maximum transition time of U1 to U0: [us] |
| wU2DevExitLat | 2 | 0000 | Maximum transition time of U2 to U1: [us] |

Configuration Descriptor (High Speed)

**Table 5-6 Configuration Descriptor (High Speed)**

| Filed Name | Size (Byte) | Data (Hex) | Description (reference design) |
|---|---|---|---|
| bLength | 1 | 09 | Size of Descriptor |
| bDescriptorType | 1 | 02 | Type of Descriptor (Configuration) |
| wTotalLength | 2 | 0027 | Total Descriptor Size |
| bNumInterfaces | 1 | 01 | Number of Interfaces |
| bConfigurationValue | 1 | 01 | Configuration Value |
| iConfiguration | 1 | 00 | String Index Number |
| bmAttributes | 1 | 80 | Configuration Attributes |
| bMaxPower | 1 | 32 | Maximum power consumption<br>HS: “bMaxPower” x 2mA |

Configuration Descriptor (Super Speed)

**Table 5-7 Configuration Descriptor (Super Speed)**

| Filed Name | Size (Byte) | Data (Hex) | Description (reference design) |
|---|---|---|---|
| bLength | 1 | 09 | Size of Descriptor |
| bDescriptorType | 1 | 02 | Type of Descriptor (Configuration) |
| wTotalLength | 2 | 0039 | Total Descriptor Size |
| bNumInterfaces | 1 | 01 | Number of Interfaces |
| bConfigurationValue | 1 | 01 | Configuration Value |
| iConfiguration | 1 | 00 | String Index Number |
| bmAttributes | 1 | 00 | Configuration Attributes |
| bMaxPower | 1 | 32 | Maximum power consumption<br>SS: “bMaxPower” x 8mA |

Interface Descriptor (High Speed)

**Table 5-8 Interface Description (High Speed)**

| Filed Name | Size (Byte) | Data (Hex) | Description (reference design) |
|---|---|---|---|
| bLength | 1 | 09 | Size of Descriptor |
| bDescriptorType | 1 | 04 | Type of Descriptor (Interface) |
| bInterfaceNumber | 1 | 00 | Interface Number |
| bAlternateSetting | 1 | 00 | Alternate Setting |
| bNumEndpoints | 1 | 03 | Endpoint Number |
| bInterfaceClass | 1 | FF | Interface Class Code |
| bInterfaceSubClass | 1 | 00 | Interface Sub Class Code |
| bInterfaceProtocol | 1 | 00 | Interface Protocol Code |
| bInterface | 1 | 00 | String Index Number |

Interface Descriptor (Super Speed)

**Table 5-9 Interface Description (Super Speed)**

| Filed Name | Size (Byte) | Data (Hex) | Description (reference design) |
|---|---|---|---|
| bLength | 1 | 09 | Size of Descriptor |
| bDescriptorType | 1 | 04 | Type of Descriptor (Interface) |
| bInterfaceNumber | 1 | 00 | Interface Number |
| bAlternateSetting | 1 | 00 | Alternate Setting |
| bNumEndpoints | 1 | 03 | Endpoint Number |
| bInterfaceClass | 1 | FF | Interface Class Code |
| bInterfaceSubClass | 1 | 00 | Interface Sub Class Code |
| bInterfaceProtocol | 1 | 00 | Interface Protocol Code |
| bInterface | 1 | 00 | String Index Number |

Endpoint Descriptor (High Speed)
Bulk Out (EP1 OUT: PC to USB), Packet Size(512Byte)

**Table 5-10 Endpoint Descriptor (High Speed: EP1 Out)**

| Filed Name | Size (Byte) | Data (Hex) | Description (reference design) |
|---|---|---|---|
| bLength | 1 | 07 | Size of Descriptor |
| bDescriptorType | 1 | 05 | Type of Descriptor (Endpoint) |
| bEndpointAddress | 1 | 01 | Endpoint number and Direction(EP1, Output) |
| bmAttributes | 1 | 02 | Transmission Type(Bulk transmission) |
| wMaxPacketSize | 2 | 0200 | Maximum Packet Size (512Byte) |
| bInterval | 1 | 00 | Interval time |

Bulk In (EP1 IN: USB to PC), Packet Size(512Byte)

**Table 5-11 Endpoint Descriptor (High Speed: EP1 In)**

| Filed Name | Size (Byte) | Data (Hex) | Description (reference design) |
|---|---|---|---|
| bLength | 1 | 07 | Size of Descriptor |
| bDescriptorType | 1 | 05 | Type of Descriptor (Endpoint) |
| bEndpointAddress | 1 | 08 | Endpoint number and Direction(EP1, Input) |
| bmAttributes | 1 | 02 | Transmission Type(Bulk transmission) |
| wMaxPacketSize | 2 | 0200 | Maximum Packet Size (512Byte) |
| bInterval | 1 | 00 | Interval time |

Bulk Out (EP2 OUT: PC to USB), Packet Size(512Byte)

**Table 5-12 Endpoint Descriptor (High Speed: EP2 Out)**

| Filed Name | Size (Byte) | Data (Hex) | Description (reference design) |
|---|---|---|---|
| bLength | 1 | 07 | Size of Descriptor |
| bDescriptorType | 1 | 05 | Type of Descriptor (Endpoint) |
| bEndpointAddress | 1 | 02 | Endpoint number and Direction(EP1, Output) |
| bmAttributes | 1 | 02 | Transmission Type(Bulk transmission) |
| wMaxPacketSize | 2 | 0200 | Maximum Packet Size (512Byte) |
| bInterval | 1 | 00 | Interval time |

Endpoint Descriptor (Super Speed)
Bulk Out (EP1 OUT: PC to USB), Packet Size(1024Byte)

**Table 5-13 Endpoint Descriptor (Super Speed: EP1 Out)**

| Filed Name | Size (Byte) | Data (Hex) | Description (reference design) |
|---|---|---|---|
| bLength | 1 | 07 | Size of Descriptor |
| bDescriptorType | 1 | 05 | Type of Descriptor (Endpoint) |
| bEndpointAddress | 1 | 01 | Endpoint number and Direction(EP1, Output) |
| bmAttributes | 1 | 02 | Transmission Type(Bulk transmission) |
| wMaxPacketSize | 2 | 0400 | Maximum Packet Size (1024Byte) |
| bInterval | 1 | 00 | Interval time |

Bulk In (EP1 IN: USB to PC), Packet Size(1024Byte)

**Table 5-14 Endpoint Descriptor (Super Speed: EP1 In)**

| Filed Name | Size (Byte) | Data (Hex) | Description (reference design) |
|---|---|---|---|
| bLength | 1 | 07 | Size of Descriptor |
| bDescriptorType | 1 | 05 | Type of Descriptor (Endpoint) |
| bEndpointAddress | 1 | 81 | Endpoint number and Direction(EP1, Input) |
| bmAttributes | 1 | 02 | Transmission Type(Bulk transmission) |
| wMaxPacketSize | 2 | 0400 | Maximum Packet Size (1024Byte) |
| bInterval | 1 | 00 | Interval time |

Bulk Out (EP2 OUT: PC to USB), Packet Size(1024Byte)

**Table 5-15 Endpoint Descriptor (Super Speed: EP2 Out)**

| Filed Name | Size (Byte) | Data (Hex) | Description (reference design) |
|---|---|---|---|
| bLength | 1 | 07 | Size of Descriptor |
| bDescriptorType | 1 | 05 | Type of Descriptor (Endpoint) |
| bEndpointAddress | 1 | 02 | Endpoint number and Direction(EP1, Output) |
| bmAttributes | 1 | 02 | Transmission Type(Bulk transmission) |
| wMaxPacketSize | 2 | 0400 | Maximum Packet Size (1024Byte) |
| bInterval | 1 | 00 | Interval time |

Super Speed Endpoint Companion Descriptor (Only Super Speed)
Bulk Out (EP1 OUT: PC to USB)

**Table 5-16 Super Speed Endpoint Companion Descriptor (EP1 Out)**

| Filed Name | Size (Byte) | Data (Hex) | Description (reference design) |
|---|---|---|---|
| bLength | 1 | 06 | Size of Descriptor |
| bDescriptorType | 1 | 30 | Type of Descriptor (Super Speed Endpoint Companion) |
| bMaxBurst | 1 | 0B | Maximum Burst Packet Number |
| bmAttributes | 1 | 00 | Maximum Stream Number |
| wBytesPerInterval | 2 | 0000 | Interval time |

Bulk In (EP1 IN: USB to PC)

**Table 5-17 Super Speed Endpoint Companion Descriptor (EP1 In)**

| Filed Name | Size (Byte) | Data (Hex) | Description (reference design) |
|---|---|---|---|
| bLength | 1 | 06 | Size of Descriptor |
| bDescriptorType | 1 | 30 | Type of Descriptor (Super Speed Endpoint Companion) |
| bMaxBurst | 1 | 0B | Maximum Burst Packet Number |
| bmAttributes | 1 | 00 | Maximum Stream Number |
| wBytesPerInterval | 2 | 0000 | Interval time |

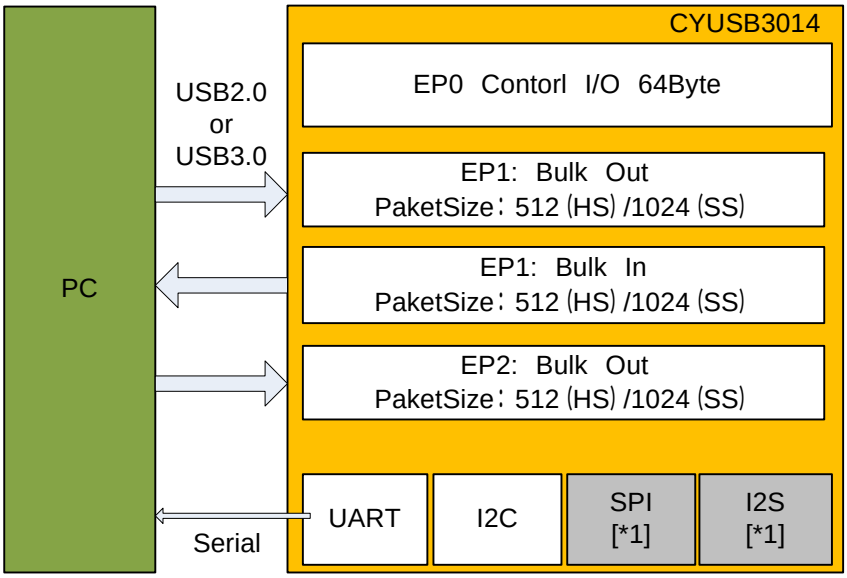
Bulk Out (EP2 OUT: PC to USB)

**Table 5-18 Super Speed Endpoint Companion Descriptor (EP2 Out)**

| Filed Name | Size (Byte) | Data (Hex) | Description (reference design) |
|---|---|---|---|
| bLength | 1 | 06 | Size of Descriptor |
| bDescriptorType | 1 | 30 | Type of Descriptor (Super Speed Endpoint Companion) |
| bMaxBurst | 1 | 0B | Maximum Burst Packet Number |
| bmAttributes | 1 | 00 | Maximum Stream Number |
| wBytesPerInterval | 2 | 0000 | Interval time |

5.1.2. End-point block diagram

Following Figure shows block diagram of end-point




Note: [*1] SPI and I2S will be invalid after the firmware-boot at this reference design.

**Table 5-19 End-Point Block Diagram**

EP0: Control each vender request command transfers.
EP1Out: Data transfer PC to FX3.
EP1in: Data transfer FX3 to PC.
EP2Out: Command packet transfer PC to FX3.

FX3 is transferring data to FPGA by GPIF interface.
FPGA is Master of GPIF interface and transfer data to SlaveFIFO mode of the FX3.

5.1.3. Vender Request Command

Following command is vender request command of this firmware.

#define VR_FIRM_GET_VERSION 0xC0 //Get_Firmware_Version

| bmRequest Type | bRequest | wValue | wIndex | wLength | data |
|---|---|---|---|---|---|
| C0h | C0h | 0000h | 0000h | 0002h | Firmware Version |

[First Byte] Minor Version
[Second Byte] Major Version

### 5.1.4. GPIFII design

Following figure shows a GPIFII connection

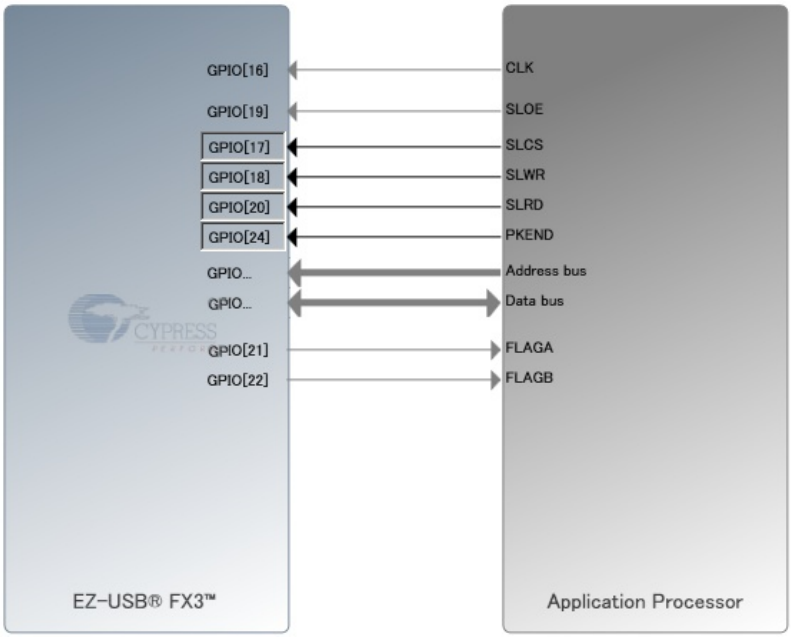



**Table 5-20 GPIFII connection**

This reference design uses the SlaveFIFO mode of the FX3. The data bus is 32bit.

Block Diagram

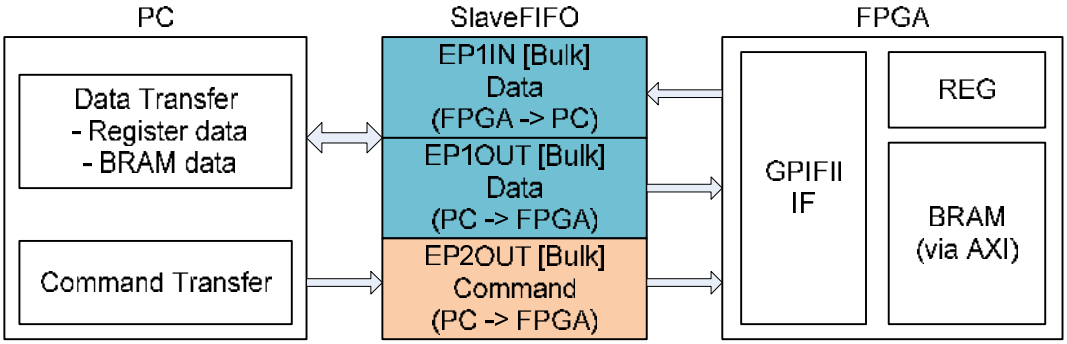



**Table 5-21 Interface block diagram**

GPIFII is Slave FIFO mode then structured by data transfer end-point EP1In/Out and EP2Out.
FPGA is waiting command from EP2Out always then processing based on received command.

Command Packet Format

<Command Packet Format>

| CMD | ADDR | LENGTH | RESERVE |
|---|---|---|---|
| Command 32Bit | Address 32Bit | Size 32Bit | Reserve 32Bit |

CMD:

Register Read: 0x0000 0004
Memory Read: 0x0000 0005
Register Write Start: 0x0000 0006
Register Write Stop: 0x0000 000E
Memory Write Start: 0x0000 0007
Memory Write Stop: 0x0000 000F

ADDR:

Write: Write to Address
Read: Read from Address

LENGTH:

Write/Read data 32 bit access count (1 = 4 Byte)

RESERVE:

## 5.2. Firmware Download

Before downloading firmware, USB device driver software should be installed in your PC.
USB device driver software is provided by Cypress Web site.

File and Tools
At first, please download EZ-USB FX3 SDK form Cypress web site and install to your PC.

Device driver file for BootLoader/BootProgram (cyusb3.inf, cyusb3.sys, WdfColnstaller01009.dll)
If install Cypress FX3 SDK for Windows ver.1.2.1, these driver software in following folder.
[Selected folder when installing SDK]/Cypress/EX-USB FX3 SDK/1.2/driver/bin
../wxp/x86: Windows XP 32bit version
../wlh/x86: Windows Vista 32 bit version *1
../wlh/x64: Windows Vista/XP 64bit *1, *2
../win7/x86: Windows 7 32bit version
../win7/x64: Windows 7 64bit version *2

Driver software of reference design(using same SDK driver software)
These files are included in reference design.
[Extract folder]/SOFT/Bin/Sys
../wxp/x86: Windows XP 32bit version
../wlh/x86: Windows Vista 32 bit version *1
../wlh/x64: Windows Vista/XP 64bit *1, *2
../win7/x86: Windows 7 32bit version
../win7/x64: Windows 7 64bit version *2

Download tool (CyControl.exe)
If install SDK Ver1.01, these driver software in following folder.
[Selected folder when installing SDK]/Cypress/EX-USB FX3 SDK/1.2/bin
Or select program from windows start menu.
[Program] – [Cypress] – [Cypress SuperSpeed USBSuite] – “control Center”

Note:
*1 This driver software is not checked with reference design.
*2 Windows mast start by “Disable Driver Signature Enforcement”. Please see Microsoft web site

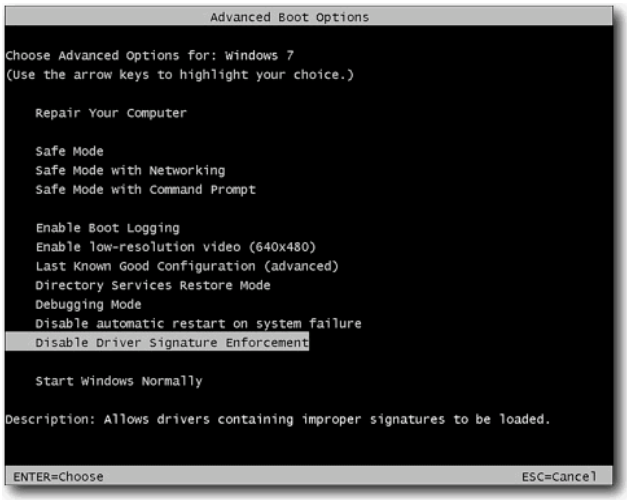


**Table 5-22 CYUSB3014 Boot up mode settings**

| PMODE[2:0] | Boot up mode | JP settings | | |
|---|---|---|---|---|
| | | JP4 | JP3 | JP2 |
| F : 0 : 0 | GPIFII Sync ADMUX (16bit) | N.C | 2-3 short | 2-3 short |
| F : 0 : 1 | GPIFII Async ADMUX (16bit) | N.C | 2-3 short | 1-2 short |
| F : 1 : 1 | USB boot | N.C | 1-2 short | 1-2 short |
| F : 0 : F | GPIFII Async SRAM (16bit) | N.C | 2-3 short | N.C |
| F : 1 : F | I2C, On Failure, USB Boot is Enabled | N.C | 1-2 short | N.C |
| 1 : F : F | I2C only | 1-2 short | N.C | N.C |
| 0 : F : 1 | SPI, On Failure, USB Boot is Enabled | 2-3 short | N.C | 1-2 short |

F:Float(N.C)　1:Connecto to Vcc　0:Connect to GND

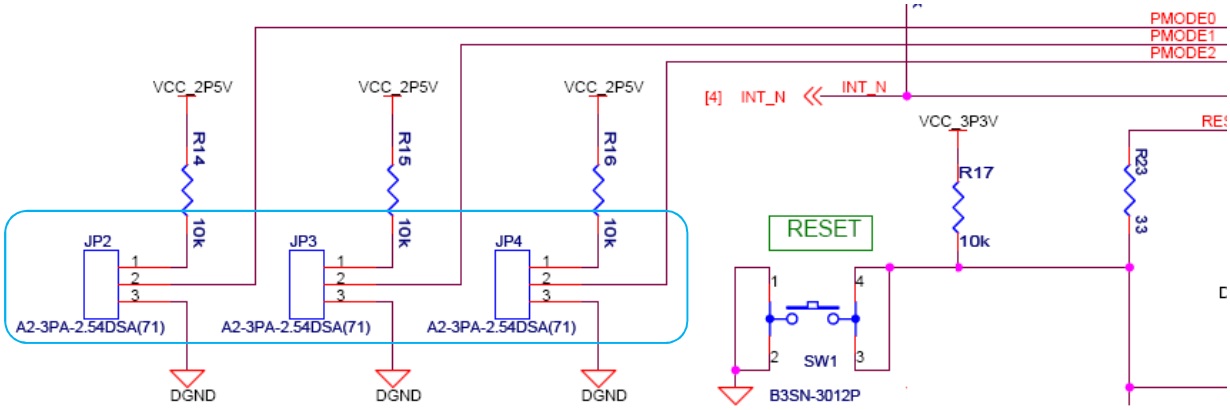



**Figure 5-1 CYUSB3014 Boot up mode setting circuit**

## 5.3. Installing Driver Software

This section describes how to install driver software on the computer connecting to the board.

### 5.3.1. Open Device Manager

Open Device Manager by selecting Windows Control Panel -> Device Manager. A device icon with mark "!" will appear (e.g. "WestBridge").
Click on this device and then **Update Driver Software**.

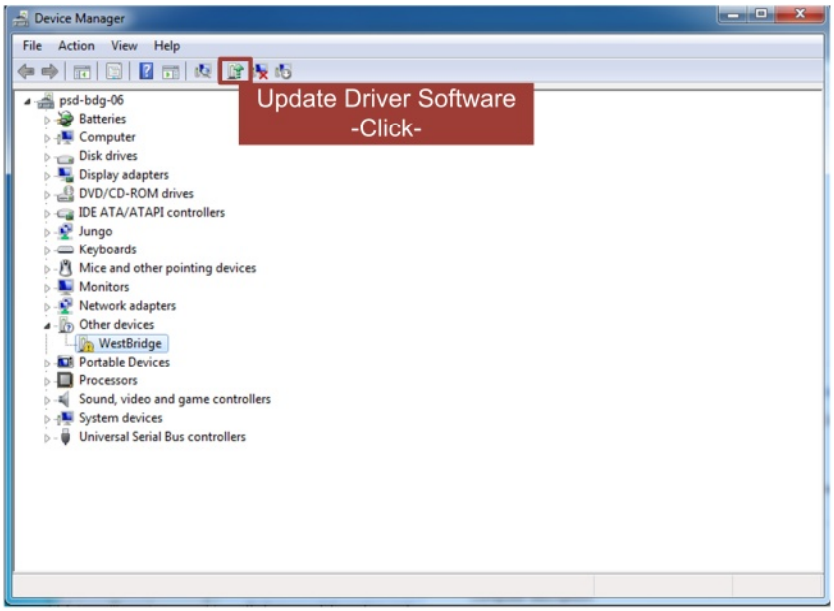



**Figure 5-2 Updating Driver Software**

### 5.3.2. Searching for Driver Software

Select **Browse my computer for driver software** to search for driver software you want.

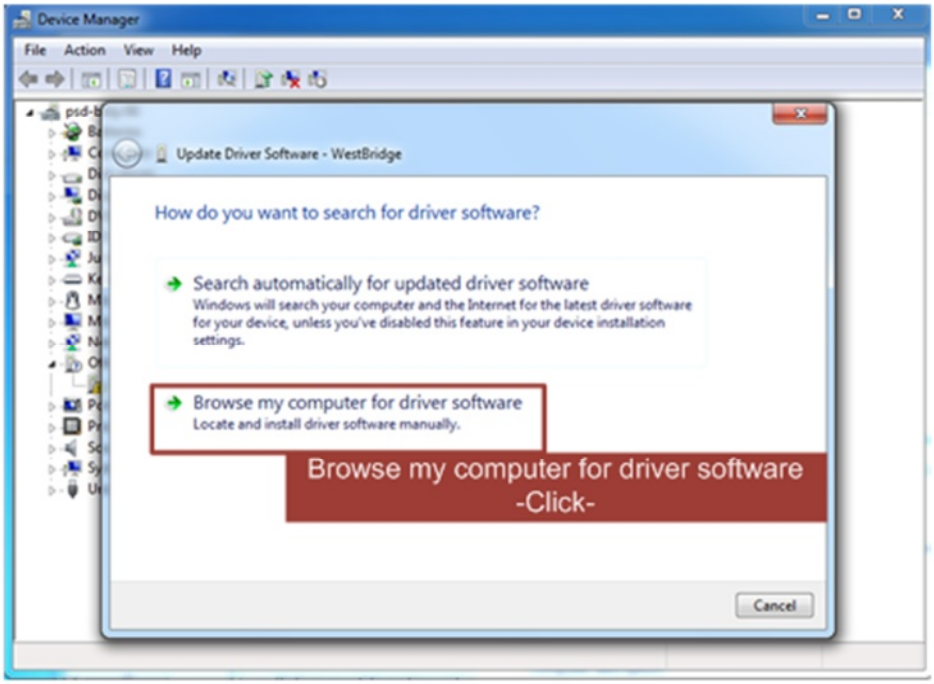



**Figure 5-3 Specifying a Search Method**

### 5.3.3. Specifying a Driver Store Folder

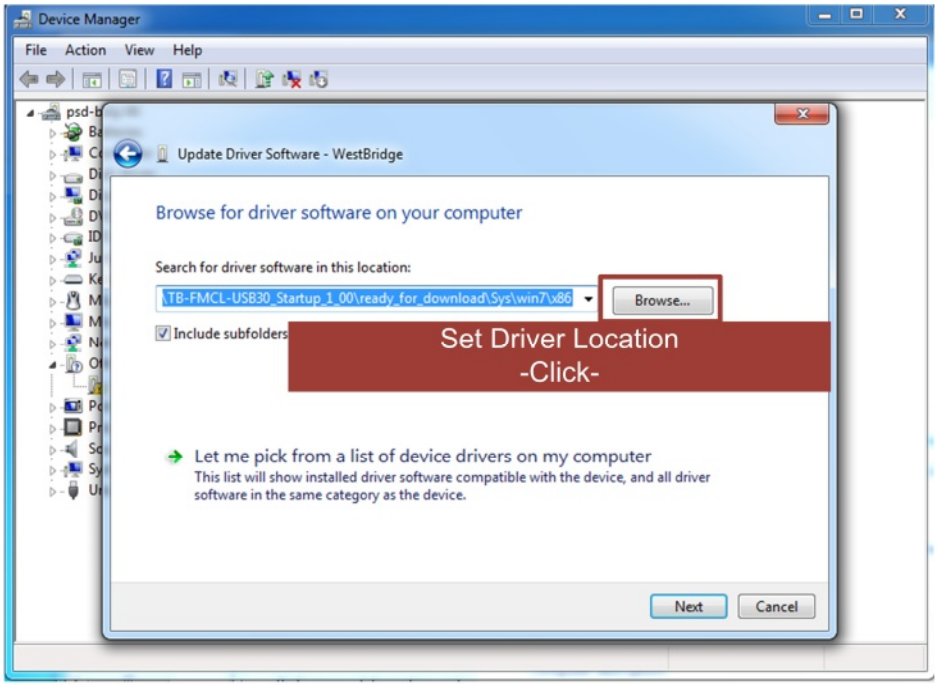



**Figure 5-4 Specifying a Driver Store Folder**

Driver Store Folder : C:/work/ready_for_download/Sys/win7/x86

### 5.3.4. Executing Driver Installation

A Windows security warning will appear. Click **Install this driver software anyway**.

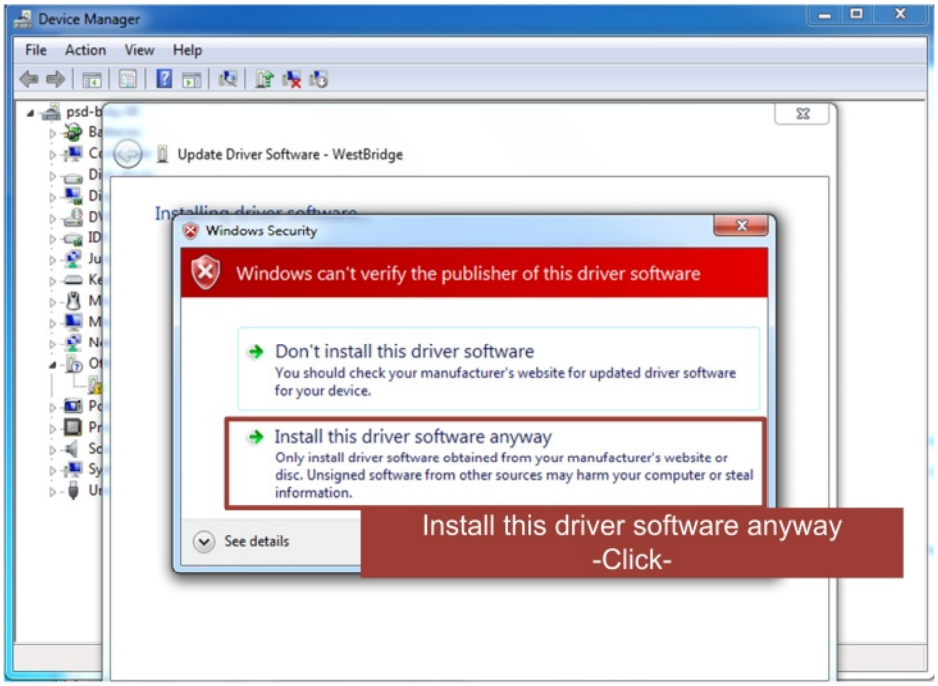



**Figure 5-5 Install Driver without Signature**

### 5.3.5. Completion of Driver Software Installation

When the software installation has been completed successfully, click on the **Close** button.

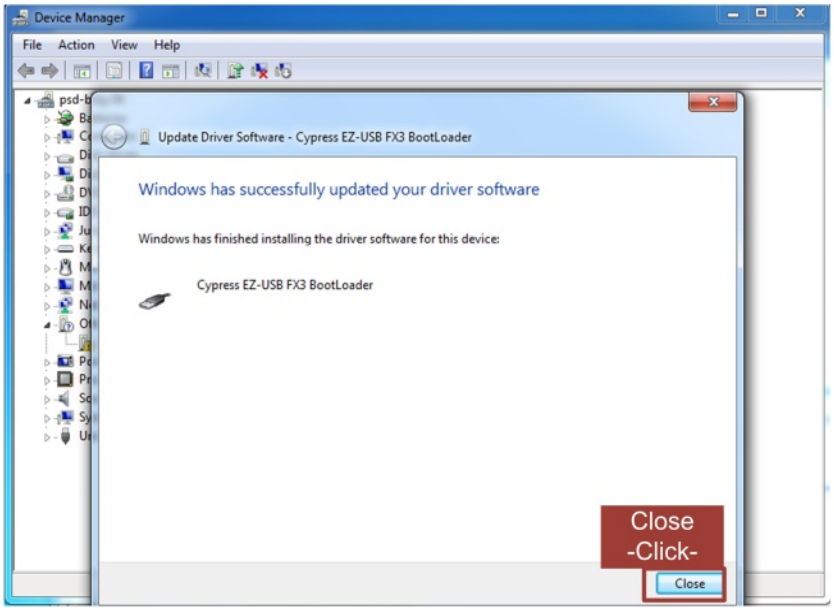



**Figure 5-6 Completion of Driver Software Installation**

## 5.4. Downloading USB3.0 Firmware

This section provides information on how to download USB3.0 Firmware.

### 5.4.1. Running CyControl.exe

Run **C:/work/ready_for_download/CyControl.exe** to download the firmware.

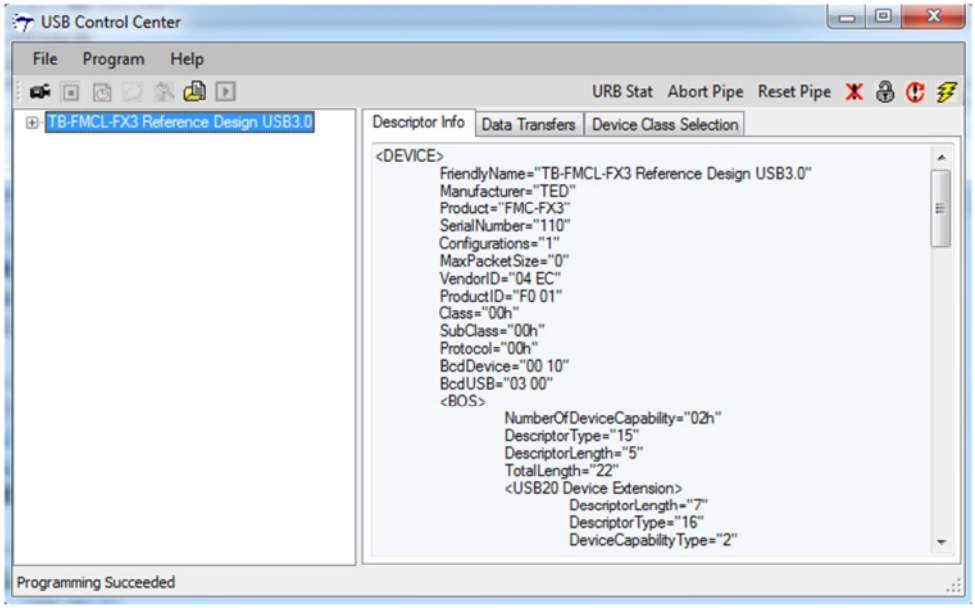



**Figure 5-7 CyControl**

### 5.4.2. Programming the Firmware

From the CyControl menu, select **Program** => **FX3** => **RAM** and specify **C:/work/ready_for_download/Firm/*.img** in the Select file to download dialog.

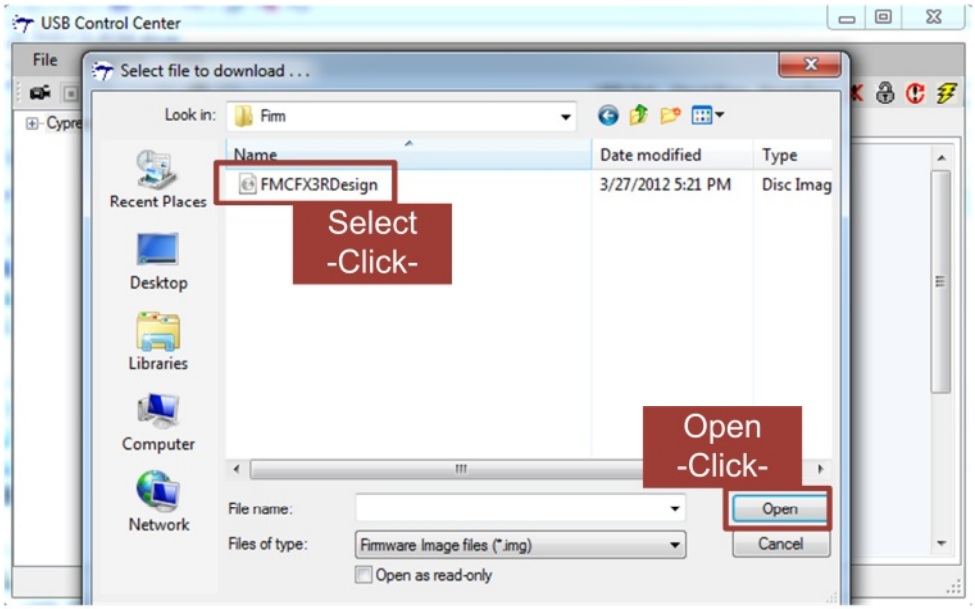



**Figure 5-8 Specifying Firmware**

### 5.4.3. Completion of Programming

When programming is successfully completed, **TB-FMCL-USB30 Reference Design USB3.0** will appear in the CyControl menu.

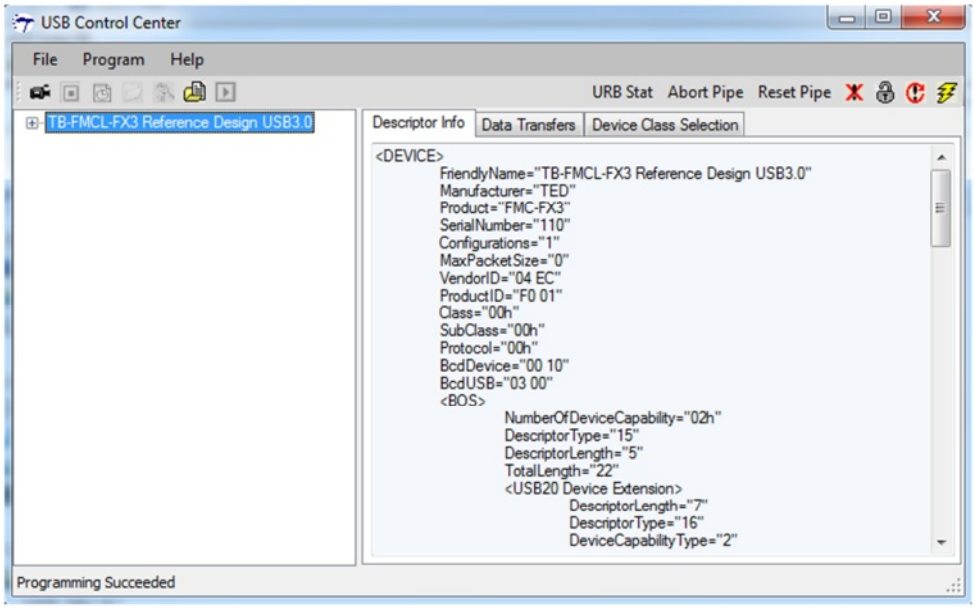



**Figure 5-9 Completion of Programming**

## 5.5. Application Designing

### 5.5.1. Detail of Function

This application is following functions.

❒PNP Event Process

When connect / disconnect USB device, automatically application open or close USB device.

❒Register/AXI BRAM Single Access(write/read)

❒Bulk Data Transfer (write/read)

❒Measuring data transfer rate

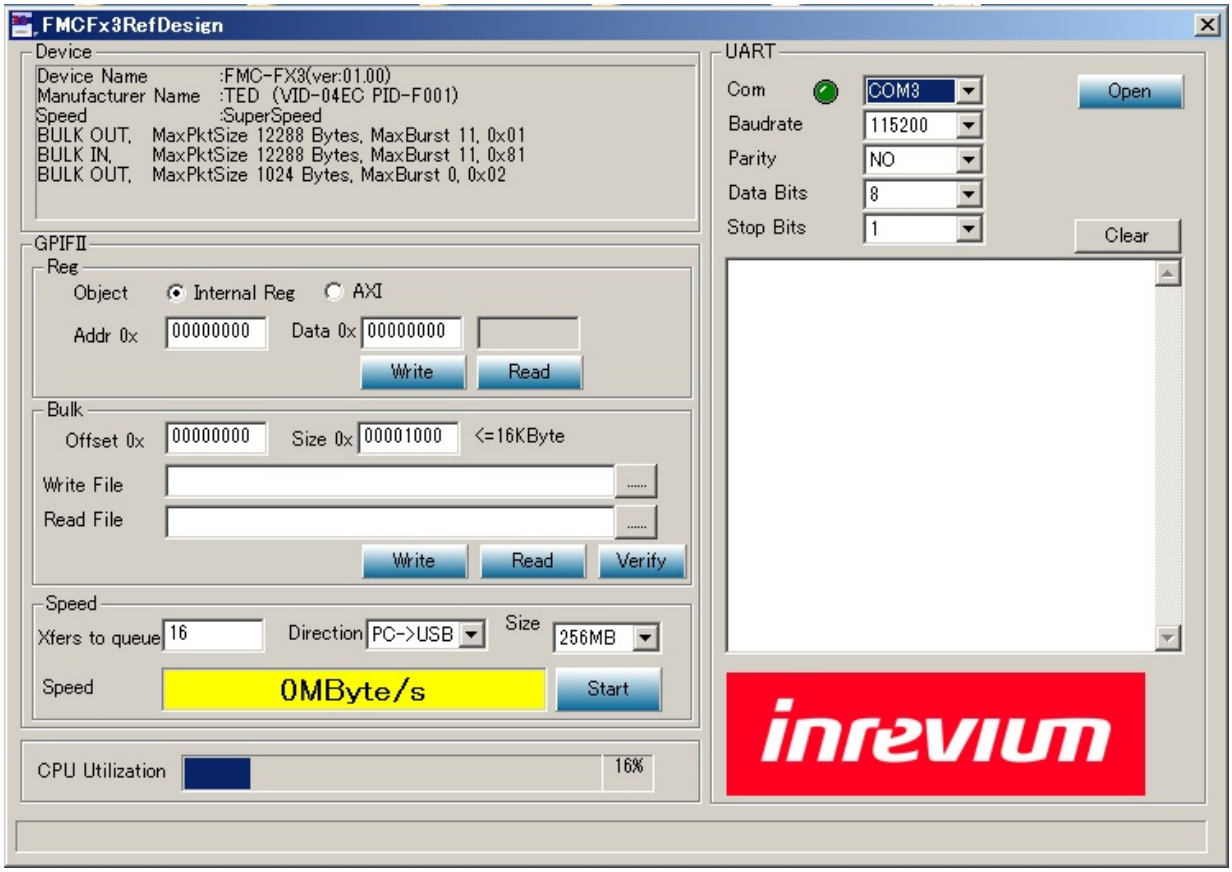
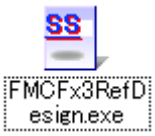



**Figure 5-10 GUI of Application**

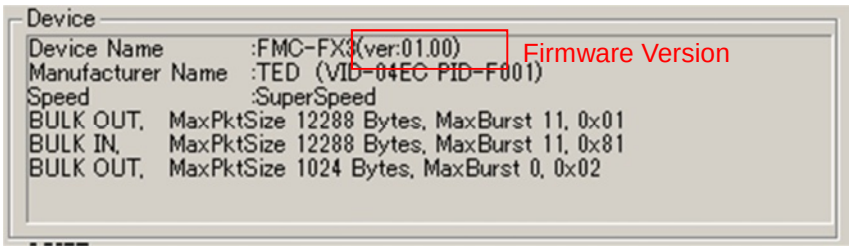
## 5.6. Operation of Application

File of application is stored in the following folder.
[Expand the ZIP file]/SOFT/Bin/App/FMCFx3RefDesign.exe



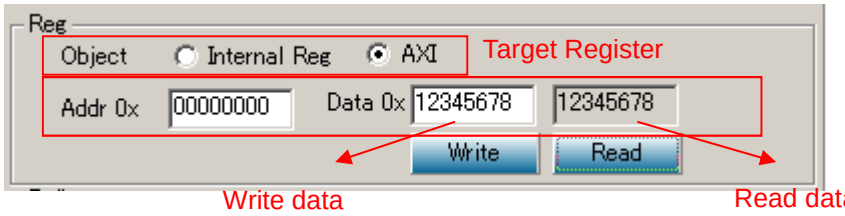

Please double clock this application.

### 5.6.1. USB Device information




- Device Name
- Manufacturer Name (Vender ID, Product ID)
- Speed: SuperSpeed(USB3.0) or HighSpeed(USB2.0)
- Endpoint information(Type / Direction / MaxPktSize, etc..)

### 5.6.2. Register / AXI BRAM Single Access



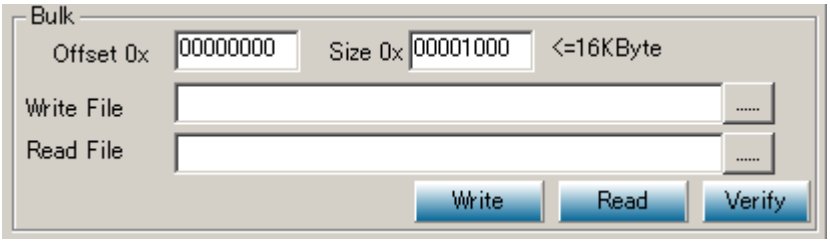

**[Object]** is selection for **“Internal Reg”** or **“AXI”**

- Write access to FPGA Register / AXI BRAM
  1, Writing target address to **[Addr]** by HEX
  2, Writing the data to **[Write Data]** by HEX
  3, Clock the **[Write]** button

- Read access to FPGA Register / AXI BRAM
  1, Writing target address to **[Addr]** by HEX
  2, Clock the **[Read]** button
  3, Read data is showed in **[Read Data]** by HEX

*Accessing Area: ”Internal Reg”: 0x00 to 0x54, “AXI”: 32bit
Data is mirroring by 16KB size.

### 5.6.3. Bulk Data Transfer

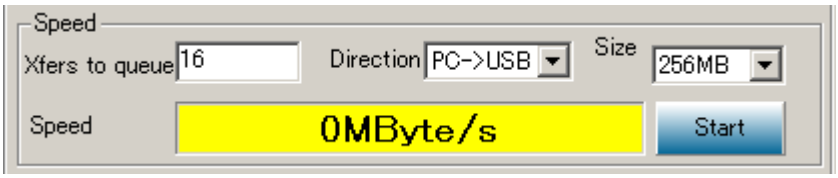



- Memory write(PC to BRAM)
  1, Writing top address of BRAM to **[Offset]** by HEX
  2, Write the size of data to **[Size]** by HEX (Alignment 4Byte, Max 16KB)
  3, Selecting the file of target data by **[Write File]**
  4, Click the **[Write]** button to starting the data transfer

- Memory red(BRAM to PC)
  1, Writing top address of BRAM to **[Offset]** by HEX
  2, Write the size of data to **[Size]** by HEX (Alignment 4Byte, Max 16KB)
  3, Selecting the file of target data by **[Read File]**
  4, Click the **[Write]** button to starting the data transfer

- Data compare
  1, Click the **[Verify]** button, the application will compare a **[Write File]** and a **[Read File]**.
  Comparing size is **[Size]**
  *Match: it shows the “Verify Successfully”
  *Mismatch: It shows the “Verify Failed at(first mismatch byte)”

### 5.6.4. Measuring the data transfer rate




This function can measure the data transfer PC to BRAM and BRAM to PC.
1, Writing the number of cueing to **[Xfers to queue]** (Max: 128)
2, Setting direction of data transfer by **[Direction]**
3, Selecting data size by **[Size]** (256MB, 512MB or 1GB)
4, Click the **[Start]** button, The data transfer is starting and measuring the data transfer rate
5, If finished the **[Size]** of data transfer, GUI shows the rate to [Speed]

### 5.6.5. Shows CPU Utilization

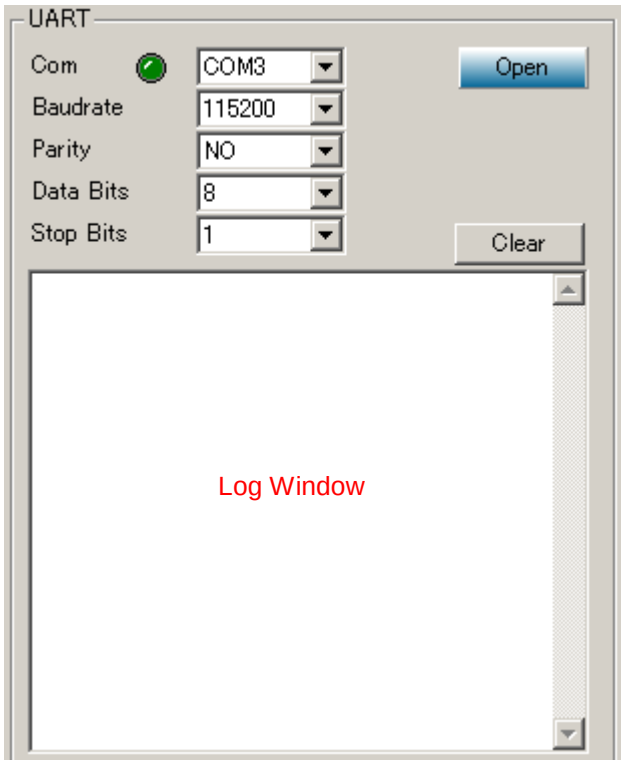

It shows CPU Utilization of OS on real time.

### 5.6.6. UART

**Notice**

**This function needs a cable for connecting PC and CN7 of TB-FMCL-USB30.**

**The cable is not attached to TB-FMCL-USB30.**




1, Setting to each parameters

**[Com]** : COM port number

**[Baudrate]** : 115200 (this reference design)

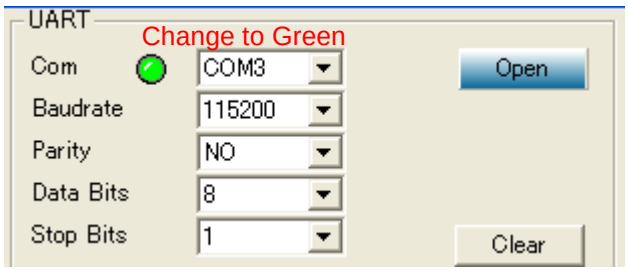
**[Parity]** : No (this reference design)

**[Data Bits]** : 8 bit (this reference design)

**[Stop Bits]** : 1 bit (this reference design)

2, Click to the **[Open]** button then open the COM port.

*If success to open the COM port, GUI shoes green at Com




3, If UART received log, GUI shows the log on windows

* Click [Clear] button then clear the log window.

### 5.6.7. Register Write Sequence

The register write sequence uses EP2OUT and EP1OUT then access to target address (4Byte).

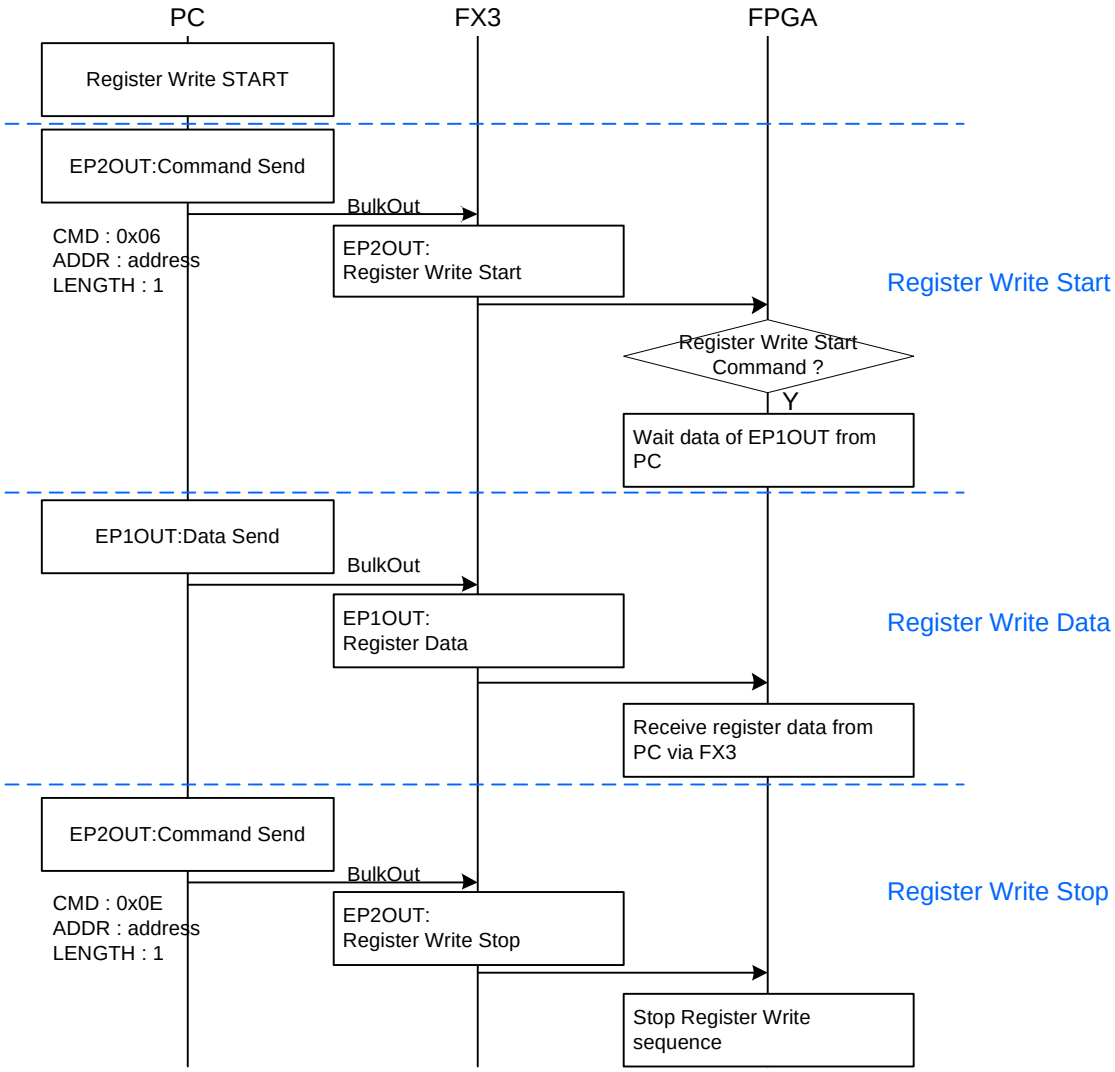



**Figure 5-11 Register write sequence**

### 5.6.8. Register Read Sequence

The register read sequence uses EP2OUT and P1IN then access to target address (4Byte).

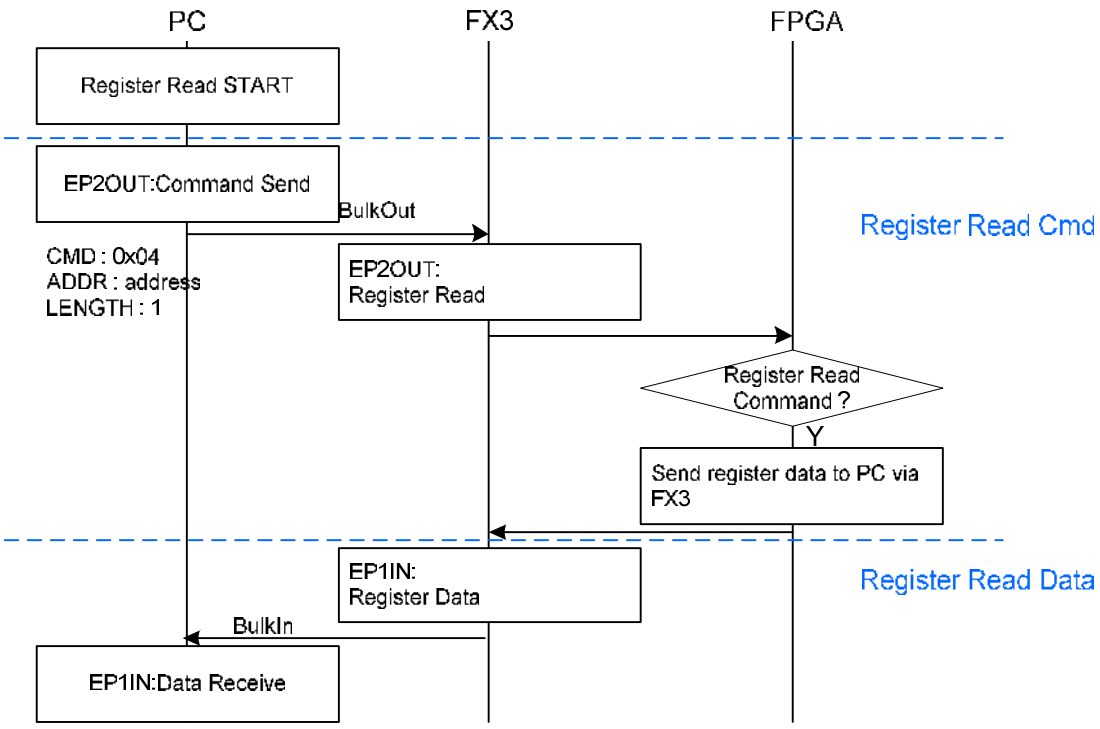



**Figure 5-12 Register read sequence**

### 5.6.9. Memory Write Sequence

The memory write sequence uses EP2OUT and EP1OUT then access to target address (Byte: LENGTH)

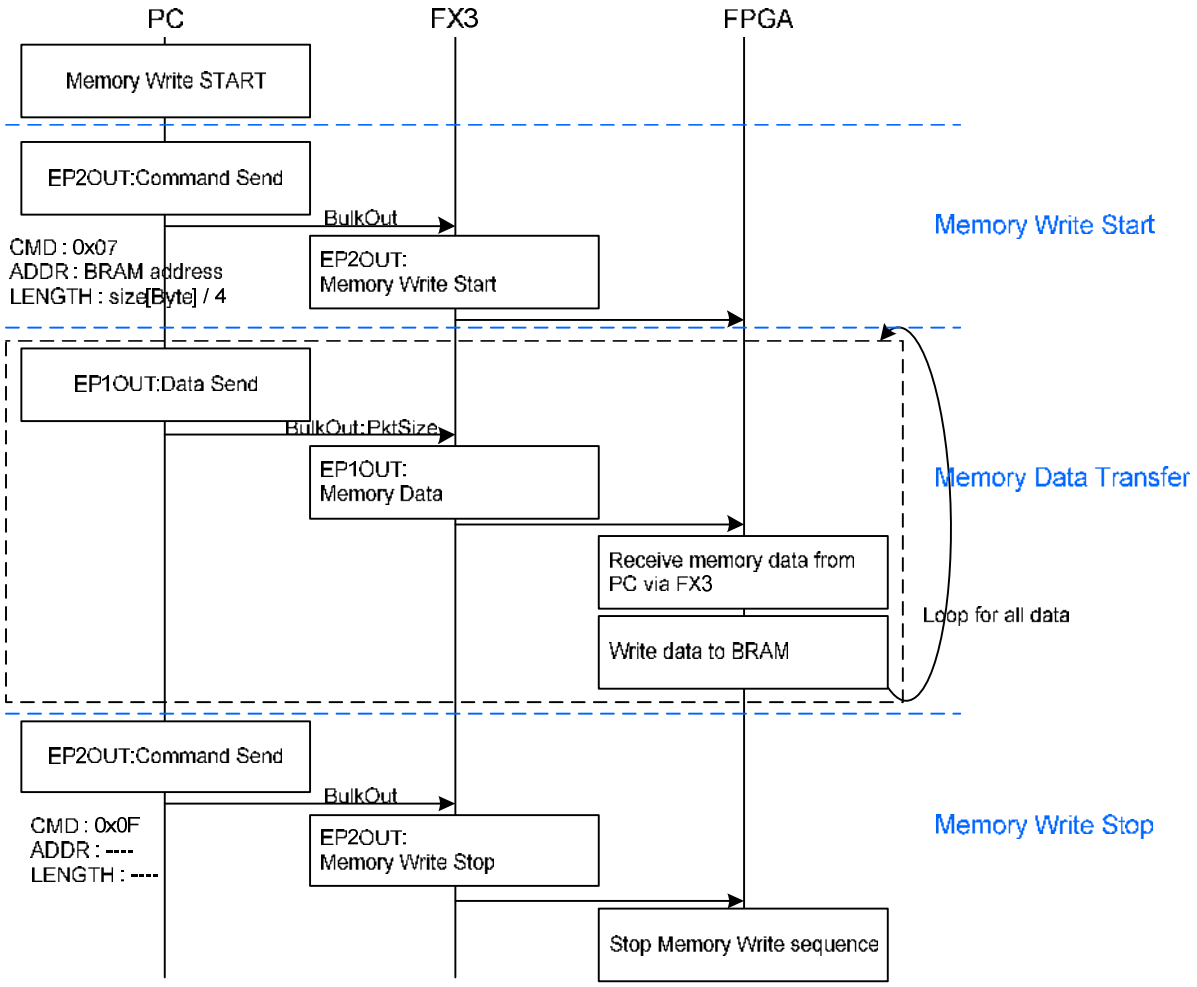



**Figure 5-13 Memory Write Sequence**

*Alignment of LENGTH: Transfer size[Byte] / 4

### 5.6.10. Memory Read Sequence

The memory read sequence uses EP2OUT and EP1OUT then access to target address (Byte: LENGTH)

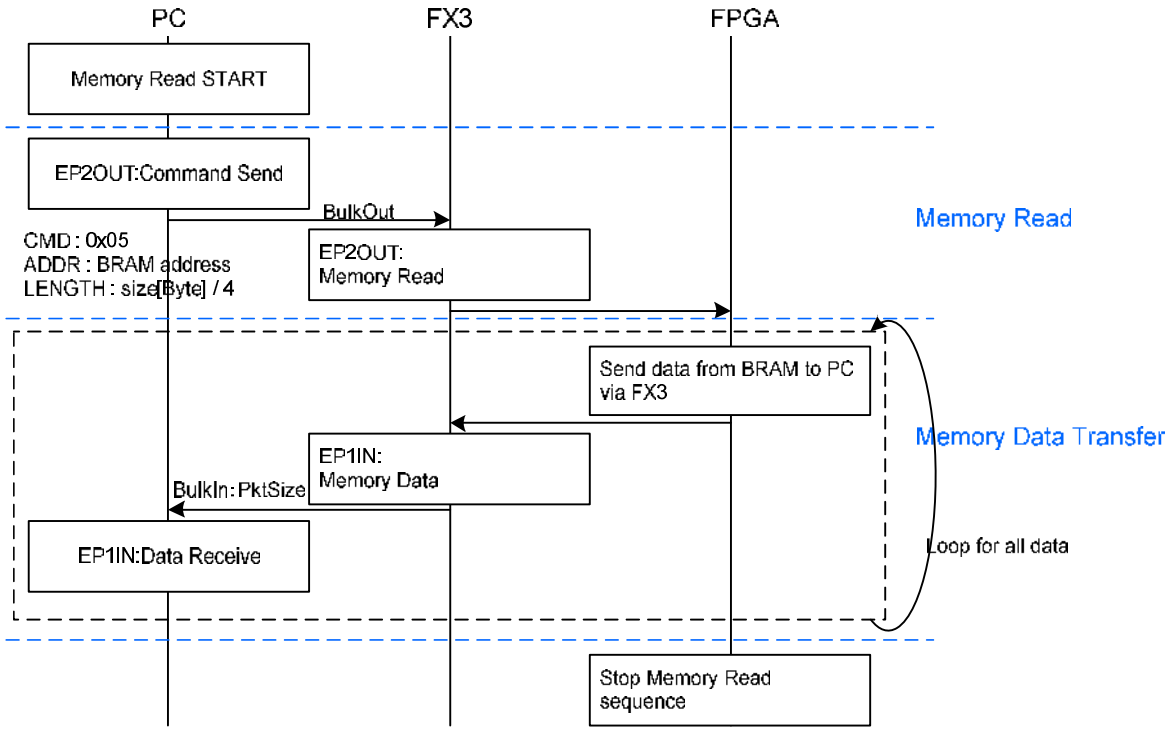



**Figure 5-14 Memory Read Sequence**

*Alignment of LENGTH: Transfer size[Byte] / 4

### 5.7. Library and Device Driver

This reference design uses the SDK from Cypress. Please refer more details to following documents

☞Library

[Expand the ZIP file] /Cypress/EZ-USB FX3 SDK/1.2/library/cpp/CyAPI.pdf

☞Device Driver

[Expand the ZIP file] /Cypress/EZ-USB FX3 SDK/1.2/driver/CyUSB.pdf

**Notice**

**Above condition is Cypress FX3 SDK for Windows ver.1.2.1.**

# 6. Appendix [Folder Structure]

Following folder structure is this reference design.

```
TB-FMCL-USB30_SampleSoftware_x.xx
└──SOFT
    └──Source
        ├──App      ··· Project file of application
        └──Firm     ··· Project file of firmware
```

TOKYO ELECTRON DEVICE

PLD Solution Dept. PLD Division
URL: **http://solutions.inrevium.com/**
E-mail: psd-support@teldevice.co.jp

HEAD Quarter: Yokohama East Square, 1-4 Kinko-cho, Kanagawa-ku, Yokohama City,
Kanagawa, Japan 221-0056
TEL: +81-45-443-4016 FAX: +81-45-443-4058