Name _____ ID _____ Due date: Wednesday, June 12

S&H Co-Design Homework #2 Handout Spring 2019

(Zynq-7010, xil_printf(), Zybo Board Interface Files)

Refer to the Zynq Book, Zybo Reference Manual (ZYBO_RM_B_V6.pdf), and Zynq-7000 Technical Reference Manual (ug585) and the other related Xilinx user guide (ugxxx) and data sheet (ds187) documents to answer the following questions. Include a pdf memo with screen captures and explanations if needed.

1 Define the following terms as used in Zynq-7000 (Most definitions are copied from the Zynq Book.)

1.1 Processing System (PS)

1.2 Programmable Logic (PL)

1.3 Multiplexed IO (MIO)

1.4 Extended Multiplexed IO (EMIO)

1.5 AXI 3.0

1.6 SelectIO

1.7 GPIO

2 The Zybo's Zynq-7010 chip bears the following marking: "XC7Z010-1CLG400C". It uses the "Ball Grid Array" (BGA) pinout for maximum packaging density. Answer the following questions about the chip. Include sources and evidences to support your answers such as document names and page numbers etc.

2.1 What is the maximum number of physical pin connections available with this particular BGA layout? (Hint: see ug865)

2.2 What is the vertical and horizontal spacing (pitch) between adjacent ball grid pins?

2.3 What are the overall physical dimensions of the Zybo's Zynq chip?

2.4 How many processing system (PS) I/O pins are available?

2.5 How many programmable logic (PL) ("Select I/O") single-ended I/O pins are available?

2.6 How many programmable logic (PL) ("Select I/O") differential I/O pairs pin pairs are available?

3 Answer the following questions about Lab #4 Part 1, i.e., ZYNQ Lab 1.

3.1 Why are DDR and FIXED_IO ports added automatically when "Run Block Automation" is executed?

3.2 What is the FIXED_IO port? What is its relationship with GPIO? What is its relationship with MIO?

3.3 What memory modules are tested?

3.4 How large are the sizes of the memory units that have been tested?

4 Describe major differences between printf() and xil_printf() and why and when xil_printf() is better than printf(). Hint: read the header of xil_printf().c source code.

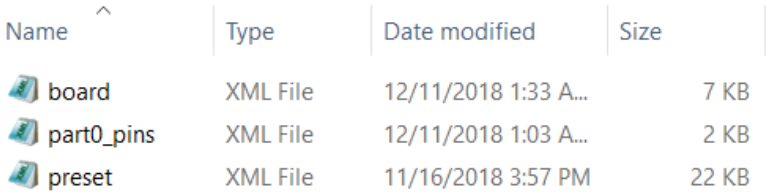
5 Read the source code for xil_printf(), describe what it does and how it is implemented to accomplish its function. Include screen captures if needed.

6 How does xil_printf() know where or which com port, to send its formatted character string? Describe the call structure and how the driver knows the base address of UART 1. Indicate how Level 1 driver of xil_printf() relates to Level 0 driver of uart device. Hint: xil_printf() uses outbyte() from uartps_3_0 library, which sends data to UART 1. Include screen captures to support your answer.

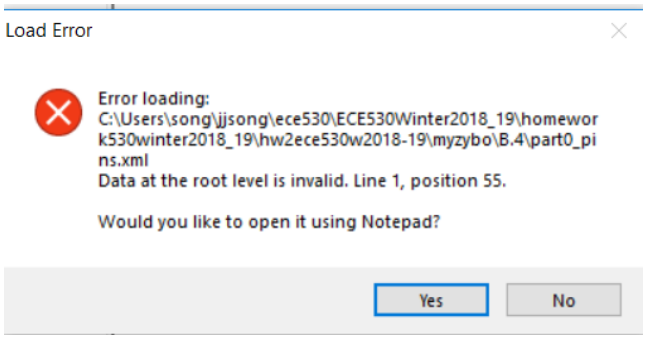
7 *Zybo Board Interface Files. Modify Version b.3 of the Zybo board interface files to include description for pMod Connector JC.*

board.xml, part0_pins.xml, preset.xml files in C: C:\Xilinx\Vivado\2015.2\data\boards\board_parts\zybo\B.3 folder is called Zybo Board Interface Files. board.xml specifies I/O pins on Zybo and part0_pins.xml specifies how the pins are connected to physical pins on Zybo. preset.xml contains reset requirements, mainly, for the ARM cores and peripherals.

The xml files can be reviewed and edited with the free Microsoft XML Notepad available from Microsoft Download Center. Or at this website: http://www.lovettsoftware.com/downloads/xmlnotepad/XmlNotepad.application.



Note: When XML Notepad is used to edit and save an XML file, characters "\r\n" are added at the end of each row so that the XML file could not be opened by XML Notepad anymore and following error will appear.



I have to open the saved xml file with Wordpad to remove all \r\n" in the file by replacing all \r\n so that I could open it again with XML Notepad.

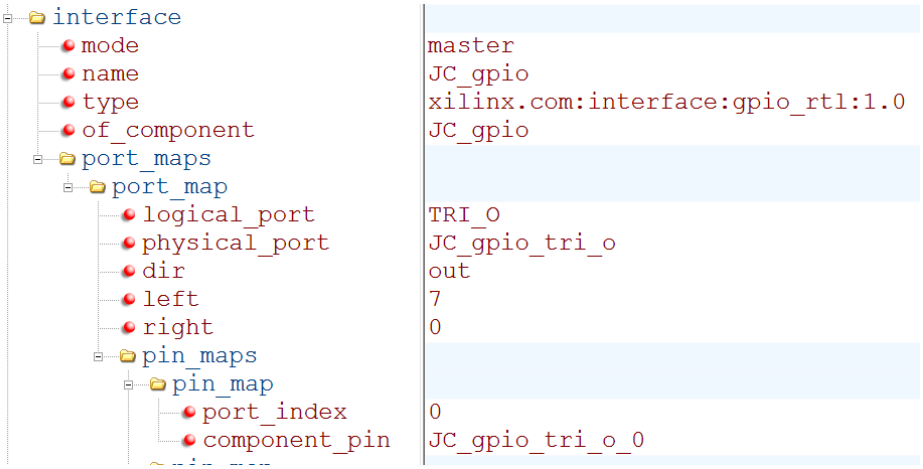
Your assignment is to edit both board.xml and part0_pins.xml to include Pmod connector JC as an 8-b bit output port.

Pmod JA (XADC)	Pmod JB (Hi-Speed)	Pmod JC (Hi-Speed)	Pmod JD (Hi-Speed)	Pmod JE (Std.)	Pmod JF (MIO)
JA1: N15	JB1: T20	JC1: V15	JD1: T14	JE1: V12	JF1: MIO-13
JA2: L14	JB2: U20	JC2: W15	JD2: T15	JE2: W16	JF2: MIO-10
JA3: K16	JB3: V20	JC3: T11	JD3: P14	JE3: J15	JF3: MIO-11
JA4: K14	JB4: W20	JC4: T10	JD4: R14	JE4: H15	JF4: MIO-12
JA7: N16	JB7: Y18	JC7: W14	JD7: U14	JE7: V13	JF7: MIO-0
JA8: L15	JB8: Y19	JC8: Y14	JD8: U15	JE8: U17	JF8: MIO-9
JA9: J16	JB9: W18	JC9: T12	JD9: V17	JE9: T17	JF9: MIO-14
JA10: J14	JB10: W19	JC10: U12	JD10: V18	JE10: Y17	JF10: MIO-15

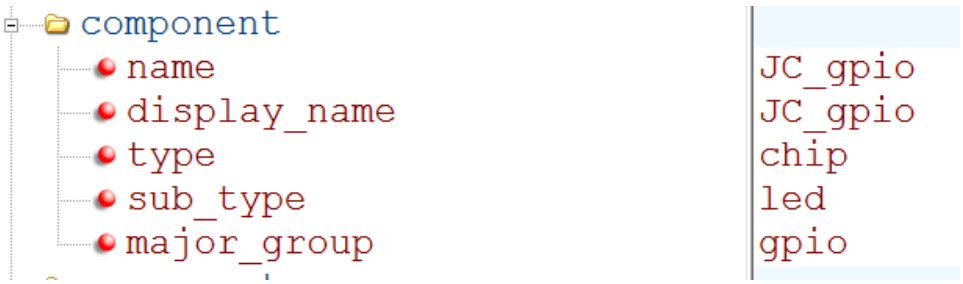
One way to make this port is to follow how leds_4bits port is made in both board_part.xml and part0_pins.xml. Open board_part.xml file with XML Notepad and choose View->Expand All under Tree View to see and edit this file. You can see interfaces under the first component.

Notice there is an interface for leds_4bits. You can duplicate this interface and change its name to JC_gpio to make it the interface for your JC port. The JC port is a 7-bit output port with indexes 7 to 0 from left to right. Use the same naming convention as that of leds_4bits.

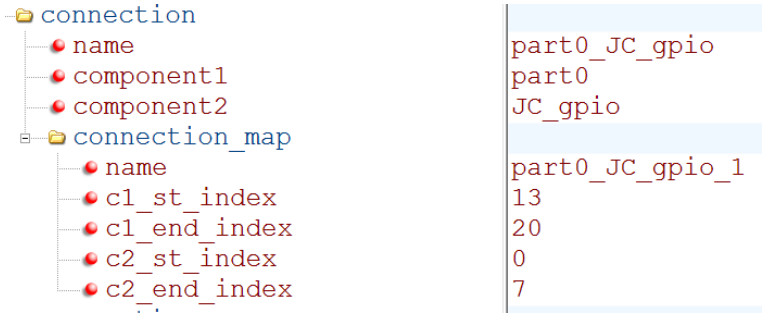
Component pin names JC_gpio_tri_o_0, … to JC_gpio_tri_o_7 are used in part0_pins.xml to match them to physical pins.



You will also need to define a component for your JC port by duplicating the component for leds_4bits and name it JC_gpio which is the same as the interface. You can keep the sub_type as led and major_group to be gpio.



You will then need to make a new connection for your JC port, named part0_JC_gpio, where component1 is part0 and component2 is JC_gpio. C1 start and end indexes are used to match pin indexes in part0_pins.xml. Since the original files have used indexes from 1 to 12. We can start c1_st_index as 13 and c1_end_indes to be 20. C2 start and end indexes are logic indexes from 0 to 7. These specifications will map pin 13 to logic pin 0 etc.



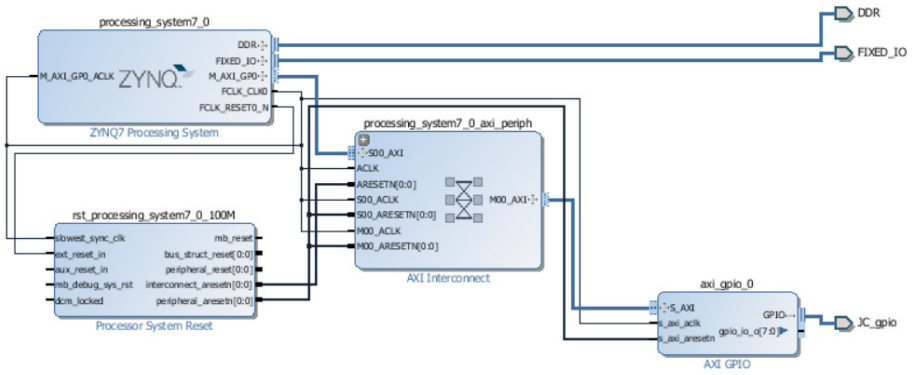
Same your board.xml file. Open it with Wordpad to remove \r\n from it. Otherwise the file cannot be opened with XML Notepad again.

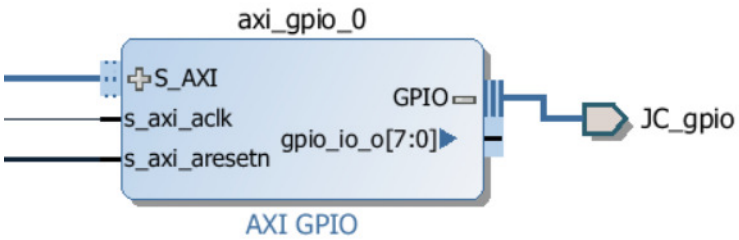
You will then need to edit part0_pins.xml to associate component pins JC_gpio_tri_o_0 etc. to physical pins. Duplicate pins eight times to add eight JC pins from JC_gpio_tri_o_0 to JC_gpio_tri_o_7 and use the correct indexes and physical pin locations for them. Same this file and open it with Wordpad to remove \r\n.

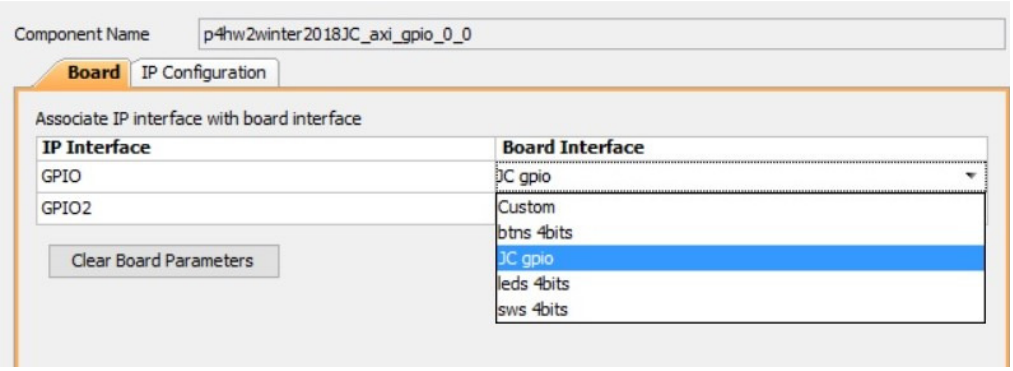
```
pin
    index       13
    name        JC_gpio_tri_o_0
    iostandard  LVCMOS33
    loc         V15
pin
    index       14
    name        JC_gpio_tri_o_1
    iostandard  LVCMOS33
    loc         W15
```

Try to open your board.xml and part0_pins.xml one more time to make sure they can be opened. Copy them to your default Zybo board file location such as C:\Xilinx\Vivado\2015.2\data\boards\board_parts\zybo\B.3.

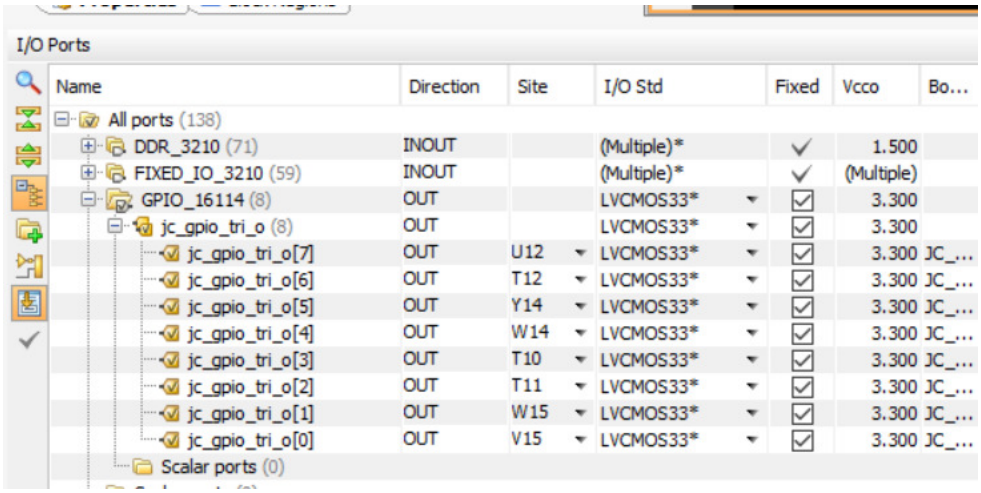
Create a project to add your JC port to a gpio interface to show you have made a correct port. Your JC port may not be visible in the board menu window, but it can be accessed when you open your gpio interface to add board interface to it.







After you synthesize your block design, you should be able to open the synthesized design and I/O port to see JC port.



Attach some screen shots of your design process and evidence to show you have created a valid JC port. Your new board and part0_pins files will work with previous labs as well as the original files.