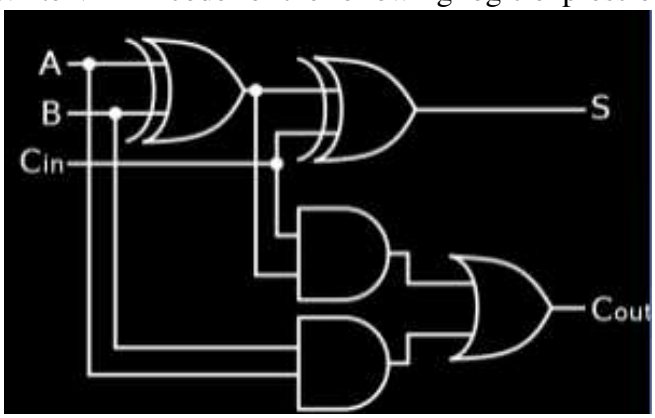
LAB Assignment #1 for ECE 443

Assigned: Mon., Aug. 31, 2015
Due: Wed., Sept. 9, 2015

Description: Write VHDL code for full_adder.vhd with inputs from switches and outputs to LEDs.

This assignment is intentionally kept simple to allow you to get familiar with the Xilinx FPGA software tools.

1) Write VHDL code for the following logic expressions for a full adder:



2) Create a block diagram with 1 instance of a GPIO with the first channel connected to the switches and the second channel set as 'custom' and labeled 'full_adder_inputs'.

3) Connect the inputs of the full adder VHDL code ('A', 'B' and 'Cin') to the low order 3 bits of the custom GPIO port in the 'design_wrapper' file.

4) Create a constaints file that maps the 's' and 'Cout' outputs to leds outputs LD0 and LD1.

5) Write a simple C program that constantly reads from switches 0, 1 and 2 through the first channel of the GPIO and writes the values to the second channel of the GPIO. See starter code.

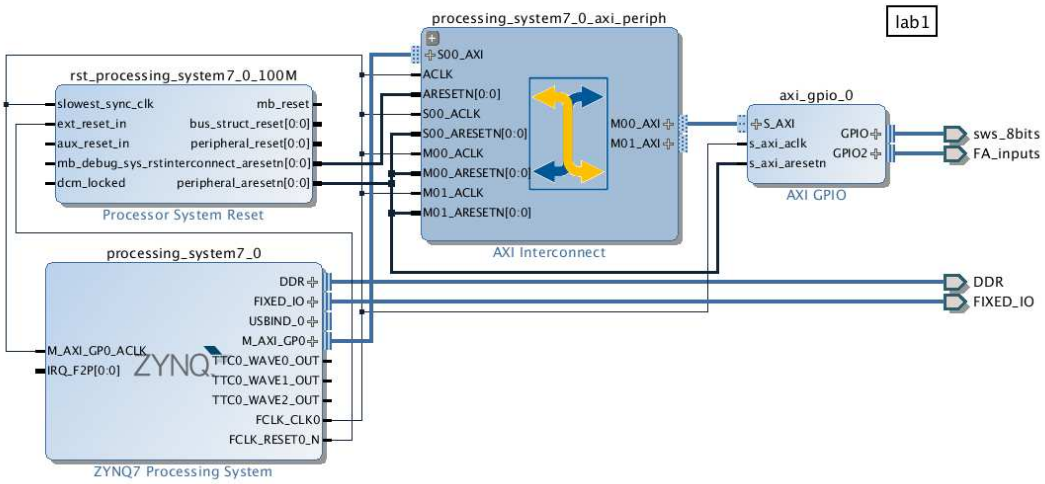
6) Switches can be changed from '0' to '1', with the 's' and 'Cout' LEDs representing the sum and carry out values.

7) Be prepared to demo your project in class.

Laboratory Report Requirements:

Grading:
The grading from this lab will be based entirely on your in-class demo. Bonus points will be given to any implementation feature that goes above and beyond the requirements.

Create this block diagram as discussed in class.



Create, regenerate layout, validate and save. Click on ‘design_1’ in the design sources window and select ‘Generate HDL wrapper’, choose ‘Copy to allow user edits’ in the pop-up.

Add this constraints file as discussed in class.

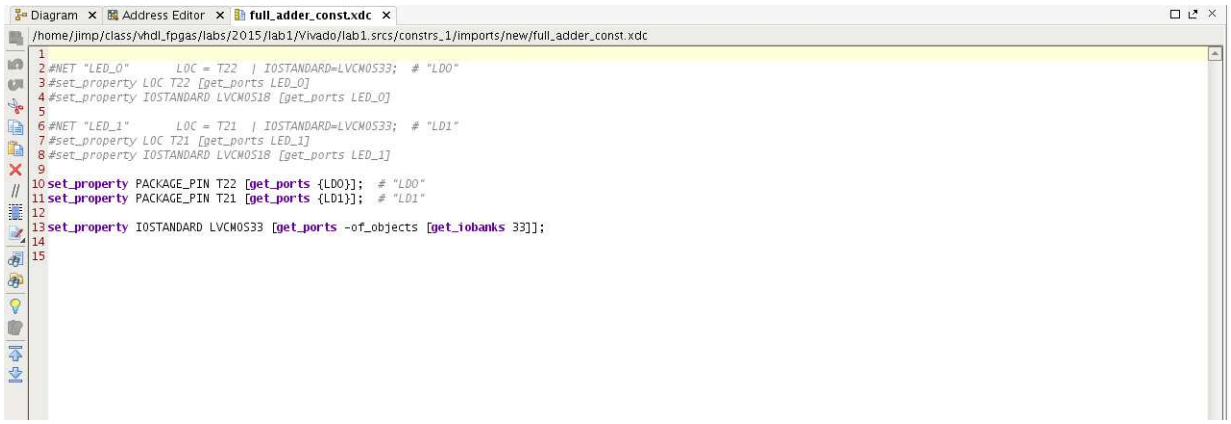


Diagram | Address Editor | full_adder_const.xdc

/home/jimp/class/vhdl_fpgas/labs/2015/lab1/Vivado/lab1.srcs/constrs_1/imports/new/full_adder_const.xdc

```
#NET "LED_0"       LOC = T22  | IOSTANDARD=LVCMOS33;  # "LD0"
#set_property LOC T22 [get_ports LED_0]
#set_property IOSTANDARD LVCMOS18 [get_ports LED_0]

#NET "LED_1"       LOC = T21  | IOSTANDARD=LVCMOS33;  # "LD1"
#set_property LOC T21 [get_ports LED_1]
#set_property IOSTANDARD LVCMOS18 [get_ports LED_1]

set_property PACKAGE_PIN T22 [get_ports {LD0}];  # "LD0"
set_property PACKAGE_PIN T21 [get_ports {LD1}];  # "LD1"

set_property IOSTANDARD LVCMOS33 [get_ports -of_objects [get_iobanks 33]];
```

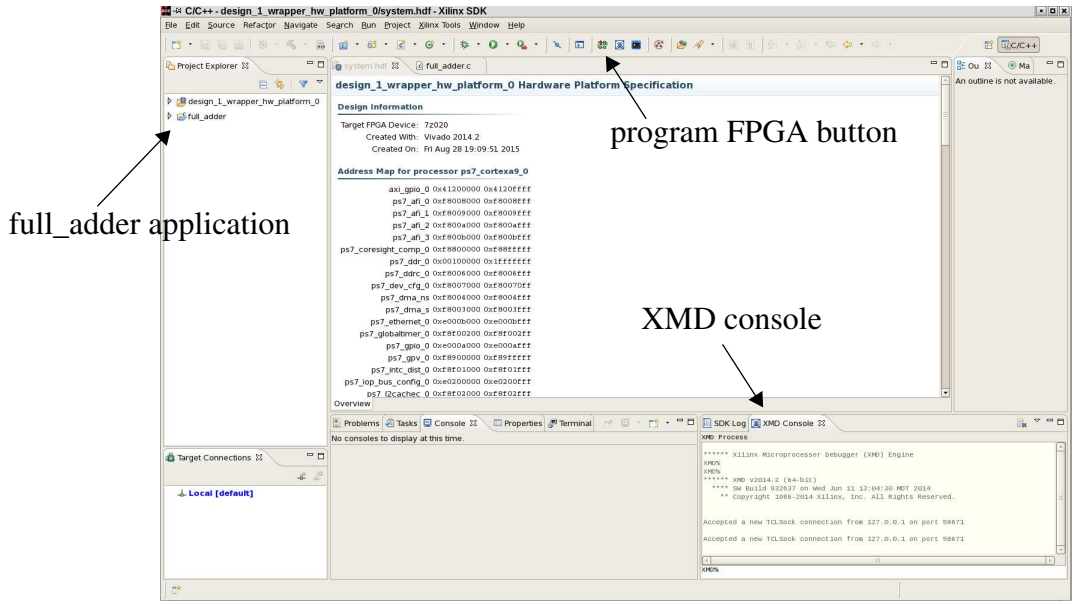
Remember to buy a USB-to-microUSB cable and a short twisted pair cable for networking.

Plug your microUSB cables into the ‘PROG’ and ‘UART’ microUSB connectors on the board (the two closest to the power supply connector).

Remember to set the default language to VHDL in the ‘Project Setting’ menu in vivado immediately after creating a new project.

SDK:

1) From vivado, File menu, run ‘Export’ and then ‘Export Hardware’, click ‘Ok’ with defaults.
2) From vivado, File menu, run ‘Launch SDK’, click ‘Ok’ with defaults.
3) SDK should run and create ‘design_1_wrapper_hw_platform_0’ component automatically.



4) From SDK, File menu, ‘New’, ‘Application Project’, type ‘full_adder’ in ‘Project name:’ field, change ‘OS platform’ to ‘linux’, click ‘Next’, choose ‘Empty application’.
5) On left, ‘full_adder’ should show up, right click and choose ‘import’. In dialog, expand ‘General’ tab and click ‘File System’, click ‘next’. Select the ‘full_adder_starter.c’ file after choosing the proper directory.
6) To force compilation, right click ‘full_adder’, click ‘Clean Project.
7) Binary is written to vivado/lab1.sdk/full_adder/Debug/full_adder.elf. This is the file you need to transfer using ‘scp’ (see below) to the Zedboard.

8) Program the FPGA with your bitstream that you synthesized in vivado using the button at the top (see figure above). Note that the serial port terminal appears to be ‘dead’ after programming the board.
9) Run the following 3 commands in the ‘XMD console’ to free up the serial port terminal (see figure above). If the XMD console is not visible, click ‘Xilinx tools’ in the menu along top and choose ‘XMD console’

connect arm hw
target 64
con

10) The serial terminal should again respond to your input commands.

Setting up the network:
Under linux, do the following:
1) Connect a twisted pair cable between ethernet ports on your laptop and Zedboard. Be sure to disable the wireless network. Also be sure the serial terminal is NOT locked up (see previous page).
2) Type 'ifconfig' to make sure 'eth0' shows up.
3) OPTIONAL: do this ONLY if 'ifconfig' does NOT show eth0 above. In an xterm on your laptop, type
    insmod /lib/modules/2.6.xx-xx.elf6.x86_64/kernel/drivers/net/xxx
    Substitute xx-xx with the linux kernel you are using and xxx with the driver for your card.

4) In the minicom window (which connects to the Zedboard through a serial channel), type
    ifconfig eth0 192.168.1.20 netmask 255.255.255.0
    Then type 'ifconfig' by itself to make sure 'eth0' is configured with the IP you just specified.

5) On your laptop, type
    ifconfig eth0 192.168.1.10 netmask 255.255.255.0

6) On your laptop, type 'route' to make sure 192.168.1.0 shows up. If not, repeat 5) above.

7) On your laptop, type
    route add default gw 192.168.1.1 eth0

8) On your laptop, type 'route' again to make sure 'default 192.168.1.1' shows up as a second entry in the routing table.

9) At this point, you should be able to type
    ssh root@192.168.1.20
    To logon to the linux system running on the Zedboard. Use 'root' as the password.

10) cd into 'vivado/lab1.sdk/full_adder/Debug/' identified on the previous page and type
    scp full_adder.elf root@192.168.1.20:/
    After type 'root' for the password, the binary is transferred to the Zedboard from your laptop.

11) In the serial channel, type 'cd /', and then 'chmod a+x full_adder.elf' as two separate commands.

12) Run the full_adder.elf binary in the background by typing:
    full_adder.elf&

13) If everything is working, you should be able to change switches 0, 1 and 2 and have the output of the full adder (sum and Cout) be displayed on LED 0 and 1.

Conceptual model: