LAB Assignment #7, for ECE 338

Assigned Oct. 31st
Due Nov. 7th

Description: Create an mixed C code/VHDL project that allows the C program to utilize your hardware-implemented multiplier.

The VHDL code that I'm supplying connects the multiplier inputs and outputs from the last lab to the GPIO registers. The C code that I am supplying shows you how to issues commands to the multiplier state machine running in the PL side and to read status and retrieve results.

Use Vivado to create a project called lab7, add the VHDL code that I have supplied (NOTE: IT IS DIFFERENT THAN THE VHDL FROM THE PREVIOUS LAB), add the block diagram as discussed in the previous lab, synthesize the project and program your FPGA with the bitstream.

NOTE: You should double click on the Zynq processor in the block diagram, click 'Clock Configuration', expand 'PL Fabric Clocks' and reset FCLK_CLK0 if necessary to ensure it is set to 50 (MHz).

Create an application in SDK and import the C file I have provided on my website for this lab. Follow the screen cast instructions.

NOTE: Please ignore the video instruction that indicates that you should change the compiler. Please use the compiler that is provided by default. Xilinx has fixed that issue.

The C program carries out several multiplications, retrieves the result and prints them to standard output.

Run the application and create a screen snapshot of the printed output from your run of the program.

This lab is worth 10 points.

IMPORTANT NOTE FOR THOSE USING ZYBO Z7:
The ZYBO kernel I have posted for ZYBO-Z7 has problems with being programmed using JTAG.

There are three ways you can program the FPGA:
1) Using the hardware manager with vivado (most of you probably use this JTAG method).
2) Running xsdb from a terminal (also uses JTAG method).
3) From a C program (or the command line) while logged into the ZYBO (FPGA manager method).

Turns out, neither of the JTAG methods work.

You can continue to use this kernel but will need to do the following:

1) From vivado, in the tcl window of your vivado project, run:
write_cfgmem -disablebitswap -force -format BIN -size 256 -interface SMAPx32 -checksum -loadbit "up 0x0 lab7/lab7.runs/impl_1/design_1_wrapper.bit" -verbose design_1_wrapper.bit.bin

Note: you may need to tune the path 'lab7/lab7.runs/impl_1/design_1_wrapper.bit' to the bitstream file depending on where you run vivado from.

This command generates an output file called 'design_1_wrapper.bit.bin' that is written to the directory where you run vivado from.

2) In a terminal window from the place where you ran vivado (and where the design_1_wrapper.bit.bin exists), copy the design_1_wrapper.bit.bin file to the ZYBO board by typing from the terminal:
scp design_1_wrapper.bit.bin root@192.168.0.10:/lib/firmware

Tune you IP address according to how you set up your network. NOTE: YOU MUST write this file to /lib/firmware on the ZYBO board.
IMPORTANT: You may need to create the 'fireware' directory first, so ssh to the ZYBO board and type:
mkdir /lib/firmware

And then scp the design_1_wrapper.bit.bin file in step 2.

3) ssh to your ZYBO board and then run the following command:
echo design_1_wrapper.bit.bin > /sys/class/fpga_manager/fpga0/firmware

You can then run the C program, e.g., lab7.elf, without the Linux OS locking up.

IMPORTANT: In future labs and the project, whenever you create a new bitstream using Vivado, run these three steps again.