## LAB Assignment #6, for ECE 338

Assigned Oct. 16
Due Oct. 23

## Description: Create a Block Diagram with the Zynq processor, 1 dual-channel GPIO and a 8 KB BRAM.

In this lab, you will begin your training on how to setup a communication channel (actually 2 32-bit registers) between the processor (PS side) to the programmable logic (PL side). The registers are called General Purpose Input/Output (GPIO). On the processor side, they will be mapped into the address space of a C program that you will use and can be read and written as if they are a normal variable in C (with a small caveat to be discussed). On the PL side, you need to create these two registers using the block diagram. We will also add a Block RAM (BRAM) so that data from the PS side can be stored and manipulated by PL side state machines (which you will do in the project).

IMPORTANT: The block diagram screencasts create the block diagram in a different fashion than the method described here. Although it is accurate with regard to how to create the IP blocks described below, it also instructs you to add 'ports' and then manually modify the design_1_wrapper. This method is valid but is a lot more work than the method described below. So please ignore components that require you to add 'ports'. The ONLY two ports that should be present in your block diagram after you are finished creating it are "DDR" and "FIXED_IO". All other connections between blocks connect to the Top.vhd block that is created when you drag-and-drop the VHDL module to the block diagram -- see below.

Create a directory called lab6, cd into that directory and create 2 subdirectories as follows:

```
mkdir VHDL
mkdir Vivado
```

Copy the Top.vhd and multiplier.vhd VHDL files that I've provided into the VHDL directory (Note: the multiplier.vhd is from the previous lab)

Change directory into Vivado, i.e. cd Vivado, run vivado and create a project called lab6

- Add the Top.vhd and multiplier.vhd files to the project

Create a block diagram.

- Click '+', search Zynq, select Zynq7 processing system
  - Run Block automation
  - Click okay (NO changes)

- Click '+', search GPIO, select AXI GPIO
  - Run Connection automation
  - Check 'axi_gpio_0', and click okay if prompted.
  - Double click on the GPIO

Configure the GPIO with 2 channels, with channel 1 configured as a 32-bit input register (ALL INPUTS) and channel 2 configured as a 32-bit output register (ALL OUTPUTS) as follows:
• Change the 'Board Interface' to 'Custom'
• Click 'IP Configuration' tab
• Leave 'All Inputs' checked
• Change GPIO Width of GPIO to 32
• Click 'Enable Dual Channel'
• For GPIO 2, Check 'All Outputs'
• Make sure GPIO Width is set to 32
• Click okay to close the configuration dialog
• Do NOT run 'Connection Automation'
• Click on pin labeled 'btns' and delete it

- Click '+', search mem, select Block Memory Generator
  Configure a memory with 8 K Bytes words, where each word is 16-bits wide as follows:
  • Double click on the BRAM_PORTA but NOT ON THE NAME -- ON THE BLOCK
  • In the Block Memory Generator dialog, set Mode to Stand Alone
  • Click 'Port A Options' tab
  • Set 'Write Width' to 16
  • Leave 'Write Depth' set to 8192
  • Select 'Always Enabled' under 'Enable Port Type'
  • Uncheck 'Primitives Output Register'
  • Click 'okay'
- Drag-and-drop the **Top.vhd** file into the block diagram
  •Click and hold on the Top.vhd file in the 'Sources' tab on the right and drag to the block diagram
  •Connect the following:
  Connect Clk from Top.vhd to the FCLK_CLK0 on the Zynq microprocessor
  Connect RESETN from Top.vhd to the FCLK_RESET_N on the Zynq microprocessor
  Expand the GPIO and GPIO2 ports by clicking on the '+' sign on the GPIO block
  Connect the GPIO_Ins from Top.vhd to the gpio_io_o port on the GPIO block
  Connect the GPIO_Outs from Top.vhd to the gpio_io_i port on the GPIO block
  Expand the BRAM_PORTA port by clicking on the '+' sign on the GPIO block
  Connect the BRAM address port from Top.vhd to the BRAM address port on the BRAM block
  Connect the BRAM write enable port from Top.vhd to the BRAM write enable port on the BRAM block
  Connect the BRAM data out (BRAM_PORTA_dout) port from Top.vhd to the BRAM data in port on the BRAM block
  Connect the BRAM data in port from Top.vhd to the BRAM data out port on the BRAM block
  Connect the clka port on the BRAM block to the FCLK_CLK0 port on the microprocessor

Click 'Validate Design' (square with check mark in it). Ignore warning about CLK_DELAY and RESETN if any show up.
Type <Ctrl-S> to save the block diagram.
Clock the Block Diagram (click 'x' in upper right hand corner.
Right click 'design_1 in Source window.

Select ‘Create HDL Wrapper’
Click ‘Copy generated wrapper to allow user edits’
Click ‘okay’.
‘design_1_wrapper.vhd’ displays

Make sure Vivado places the ‘3 point triangle’ graphical symbol next to design_1_wrapper. If it doesn’t, right clock on design_1_wrapper.vhd and click ‘Set as Top’

Note that if you make changes to the VHDL code, Vivado will ask you to ‘Refresh Changed Modules’. Click to confirm. It opens the block diagram, which you can immediately close

Run Synthesis, Implementation and Generate Bitstream and then program your FPGA

Turn in a screen snapshot of your block diagram.

This lab is worth 10 points.