Lab 16: Building a 4-bit Memory

Use four D flip flops (DFFs) to build and implement a simple 4-bit memory using VHDL and your Digilab board. You will implement the DFF in VHDL as a lower level code block and then instantiate that four times in the top level code block. See module 16 and ch. 12 of your text for information about flip flops.

Submission of the lab consists of:

1) (0pts) a cover sheet with you name and section,
2) (80 pts total) a print out of the VHDL code and lower level code blocks,
3) (20 pts total) a print of the constraints file (*.xdc)
   [Before printing, delete the unused switches, leds, etc It needs to be readable.]
4) Demonstrate the working project on the Digilab board to your instructor.

Failure to demonstrate your project will cause your grade for the above to be discounted by 50%. A circuit that does not meet the required specifications will result in your lab being discounted by a minimum of 10%.

Here are the specifications your circuit must meet.

The data are entered using sw0-sw3. led0-led3 display the position of the switches. When btnR is pressed, the data is read from the switches and appears at the outputs of the DFFs. The outputs of the DFFs are displayed using led4-led7. Pressing btnL clears the memory. The “clear” function must be asynchronous, meaning it must not depend on the clock. Hint: Check the Elab SPL lab web site for example code for a DFF with synchronous reset. You will need to modify that code to get the functionality you need here. Also you will need to include this statement in your XDC file.

```
set_property CLOCK_DEDICATED_ROUTE FALSE [get_nets btnR]
```

Be sure to save your work. You will use this project again as part of a later lab.

Switch & LED Assignments		
switch	current switch state	switch state in memory
sw0	led0	led4
sw1	led1	led5
sw2	led2	led6
sw3	led3	led7

Your instructor has a working model you can inspect.