Lab 1 Assignment

ECE 525.442 FPGA Design using VHDL

Simple Seven-Segment Display Controller

For this exercise, you will configure the Xilinx field programmable gate array (FPGA) on the Nexys-4 development board to act as a simple controller for all eight digits of the seven-segment display. The ports to the top-level design are a 16-bit `std_logic_vector` input driven by the slider switches and three push buttons to change the state of the seven-segment display. The output of the top-level design is an 8-bit `std_logic_vector` named `seg7`. This vector illuminates the seven-segment digits as well as the decimal point located with each digit. Another 8-bit `std_logic_vector` output called the anodes is used to select which of the seven-segment digits to illuminate. For this implementation, the value of the slider switches SW3-SW0 is shown on the seven-segment display using hexadecimal representation (0 through F).

Task 1

Complete Nexys4DDR_Tutorial.pdf, as this will guide you through the creation of a project and a simple seven-segment decoder which can be used in the remainder of this lab assignment.

Task 2

Create an entity/architecture pair called `seg7_hex` in a file named *seg7_hex.vhd* (NOTE: It's good practice to name the VHD file the same as the entity name)

The `seg7_hex` entity should have two ports: a 4-bit input called `digit` and an 8-bit output called `seg7`. This component is instantiated in your top-level architecture. You can take inspiration from the Task 1 Tutorial on how to implement this.

Task 3

Create a new top-level entity/architecture pair called `lab1_top` in a file named *lab1_top.vhd*. This new entity is the main entity for the design and will instantiate the 7-segment decoder that you built in Task 2. This will be the first design you build that will be synthesized and loaded to the FPGA on the Nexys-4 board. The entity for `lab1_top` should look like the following code:

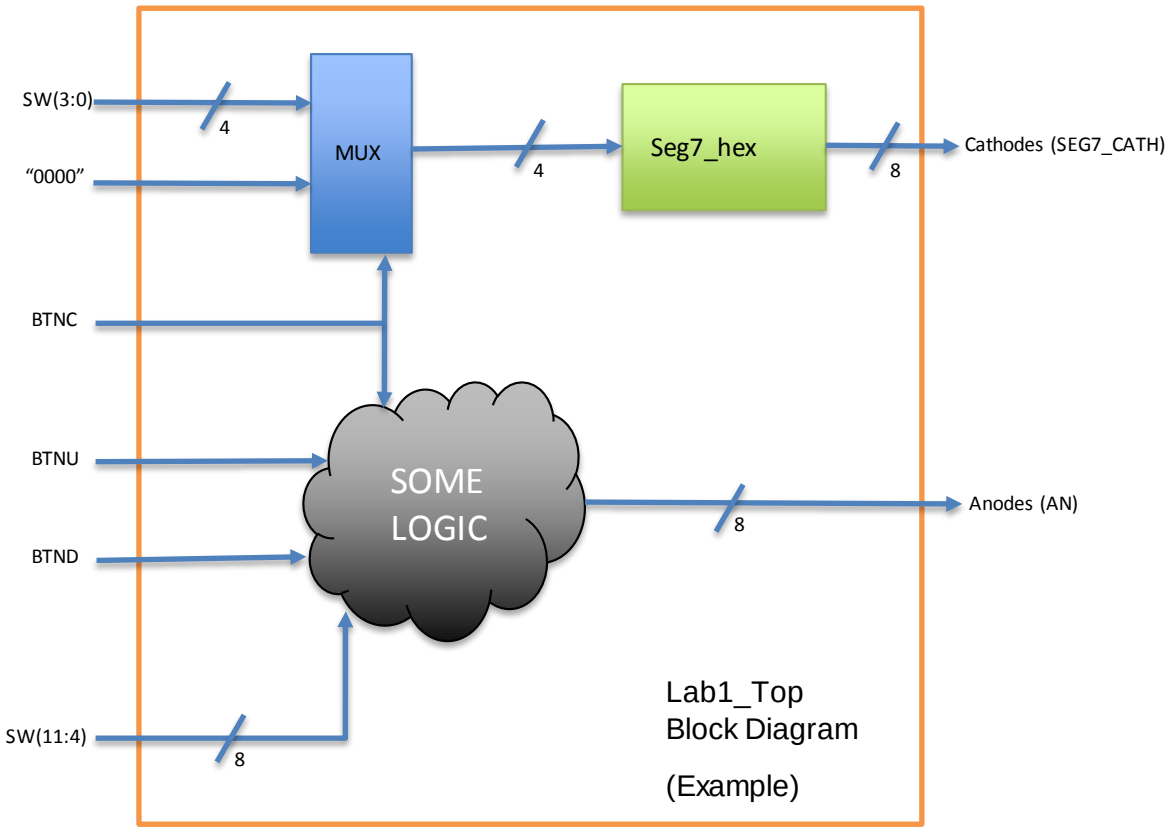
```
entity lab1_top is port (
        --Push Buttons
        BTNC : in STD_LOGIC;
        BTND : in STD_LOGIC;
        BTNU : in STD_LOGIC;
        --Switches (16 Switches)
        SW : in STD_LOGIC_VECTOR (15 downto 0);
        --LEDs (16 LEDs)
        LED : out STD_LOGIC_VECTOR (15 downto 0);
        --Seg7 Display Signals
        SEG7_CATH : out STD_LOGIC_VECTOR (7 downto 0);
        AN : out STD_LOGIC_VECTOR (7 downto 0));
end lab1_top;
```

The main component of `lab1_top`'s architecture should be an instantiation of `seg7_hex` as built in Task 2. The remaining logic contained in `lab1_top` should implement the following tasks :

1. The input to the `seg7_hex` component should be the four slider switches SW3-SW0 or all zero's (depending on the push buttons below). SW3 is considered the most significant bit (msb) of the binary number and SW0 is the least (lsb).
2. Normally, with none of the push buttons pressed, the displays will be active depending on the state of SW11 through SW4, which will form an 8-bit binary number. Each switch determines which digit on the seven segment display is active. SW11 makes the left-most digit active, SW10 makes the next digit active, and continues to SW4 which activates the right-most digit.
3. When the UP BTN is pressed (**BTNU**), only the upper four digits will be active and showing the SW3 to SW0 value, independent of the values of SW11 through SW4.
4. When the DOWN BTN is pressed (**BTND**), only the bottom four digits will be active and showing the SW3 to SW0 value, independent of the values of SW11 through SW4.
5. When the CENTER BTN is pressed (**BTNC**), all eight digits will be active and showing all zeros, independent of the values of SW11 through SW4 and independent of the digit selection of SW3 to SW0.
6. The green LEDs, labeled LD15 through LD0 should light up on the board based on which switch SW15 through SW0 is set to the HIGH position. This should occur ALL the time.

NOTE: Assume only one push button is pressed at any given time.

A simple block diagram of how your design may be implemented is illustrated below. Your design should consist of two entity/architecture pairs: `seg7_hex` and `lab1_top`, no others. In other words, a block in the diagram below doesn't necessarily mean a new VHDL component, just some combinatorial logic. The top_level entity/architecture pair (`lab1_top`) should instantiate the `seg7_hex` component.



Task 4

Set the pin assignments for the ports of `lab1_top` to the appropriate Xilinx FPGA pins and save this file as a XDC. I would recommend starting with the Nexys 4 DDR Master XDC File and comment out all the unused pins. Keeping the naming consistent with the Master XDC file will make it easier to share designs throughout this course. Synthesize, generate a programming file, and upload this file to the board using the Hardware Manager. Test your design to verify that it works.

To Submit

Submit the following files zipped up together in a file called Lab1_<last_name>.zip:

- All VHD files for the Lab
- The XDC constraints file
- BIT file for programming the FPGA
- Synthesis Results as Word or Text file

Note: The Grading sheet on the next page provides a good overview of the key points for the design and what the instructor looks for when grading. Please note that this is a guide and isn't comprehensive. Please read over the lab assignment carefully to verify the proper behavior of the entire design and all the proper documents are submitted.

Grading Sheet

Lab 1/Module 1: Simple 7-segment display

Lab	1	
Student:		

Grade	Out of	Notes
	100	

	25	Seven segment encoder (`seg7_hex`) module correctly implemented
	5	Value of SW3-SW0 shows up on seven-segment display
	10	Values displayed on seven segment display should only be hexadecimal (0-F)
	10	Green LEDs light up according to which slider switch is in HIGH position
	25	Correct Operation of Push Buttons
		Slider switches SW3-SW0 represent input digit
		1. With no buttons pressed, the digit is displayed in any display that has the corresponding slider switch set HIGH from SW11 to SW4.
		2. When the UP BTN is pressed (BTNU), only the upper four digits are active and showing the SW3 to SW0 value
		3. When the DOWN BTN is pressed (BTND), only the bottom four digits are active and showing the SW3 to SW0 value
		4. When the CENTER BTN is pressed (BTNC), all of the digits are active and showing all zeros
	20	Well documented, commented and appropriate VHDL Code
	5	Synthesis report with device resource utilization summary