FPGA Design Using VHDL
EN 525.442.81
Lab 1

Simple Seven-Segment Display Controller

For this exercise, you will configure the Xilinx field programmable gate array (FPGA) on the Nexys-2 development board to act as a simple controller for the boards seven-segment display. The inputs to the top-level design will be an 8-bit std_logic_vector input driven by the slider switches and a 2-bit std_logic_vector input driven by two of the pushbuttons. The output of the top-level design will be a seven (7) bit std_logic_vector named seg7. This vector will illuminate the segments to form the numeric representation of the value represented by the slider switches. Additionally, a 4-bit output (anodes) will be used to select which of the seven-segment digits to illuminate. For this implementation, the value of slider switches SW3-SW0 will be displayed on the seven segment display using hexadecimal representation.

Note: ISE does not correctly handle spaces in the project path – please store your project in a directory tree without spaces.

Task 1
Run through the class tutorial located on the course webpage [NEXYS2Tutorial.pdf].

Task 2
Create an entity/architecture pair called seg7_hex in a file named *seg7_hex.vhd* that performs the above function. This entity should only have two ports: a 4-bit input `digit` and a 7 bit output `seg7`. This component will be instantiated in your top-level architecture.

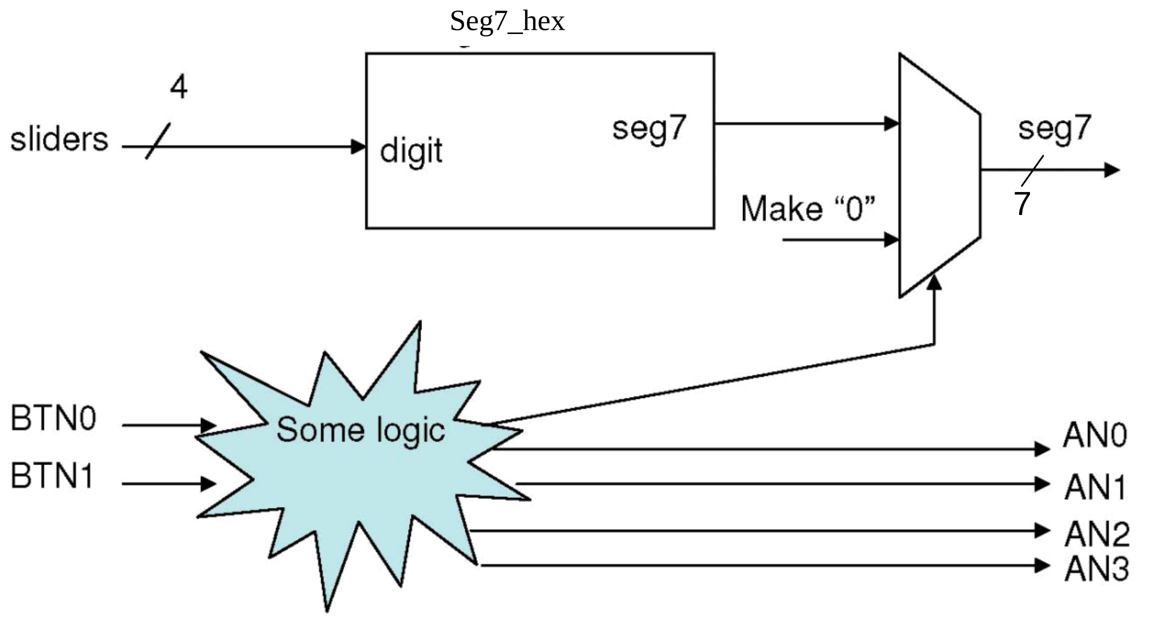
Task 3
Create a new top-level entity/architecture pair called lab1_top in a file named *lab1_top.vhd*. This new entity will **instantiate** the 7-segment controller that you built in Task 2. This will be the first design that you synthesize and load to the FPGA on the Nexys-2 board. The entity for lab1_top should look like the following code (if you would like to make the anodes a std_logic_vector, feel free):

```
Entity lab1_top is
Port (
      sliderSwitches : in std_logic_vector(7 downto 0);
      Pushbuttons : in std_logic_vector(1 downto 0);
      Seg7 : out std_logic_vector(6 downto 0);
      AN0  : out std_logic;      --low = enable rightmost digit
      AN1  : out std_logic;
      AN2  : out std_logic;
      AN3  : out std_logic;      --low = enable leftmost digit
      LEDs : out std_logic_vector(7 downto 0) – green LEDs
);
```

The main component of lab1_top's architecture should be an instantiation of seg7_hex as built in Task 2. You will hook it up to perform the tasks as described below:

1. The input to the seven-segment controller will be the four slider switches SW3-SW0. SW3 is considered to be the msb of the binary number and SW0 is the lsb.
2. Normally, with none of the pushbuttons pressed, the digit will be displayed on the rightmost LED display.
3. When **BTN0 only** is pressed, the digit will be displayed on the second from the right LED display (i.e the one controlled by AN1)
4. When **BTN1 only** is pressed, the digit will be displayed on both of the two rightmost displays.
5. When **both BTN0 and BTN1** are pressed, **the digit 0 should be displayed** on ALL FOUR LED displays regardless of the current switch values.
6. The green LEDs, labeled LD0-LD7 on your board, will light up based on which switch, SW0-SW7, is set to the HIGH position.

A small block diagram of how this might 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 on the diagram below doesn't necessarily mean a new VHDL entity, just some logic. The top_level entity/architecture pair should instantiate the seven-segment encoder.



Task 4
Set the pin assignments for the ports of `lab1_top` to the appropriate Xilinx FPGA pins and save this file as a UCF. Synthesize, generate a programming file, and upload this file to the board using the Digilent Adept tool. Test your design to verify that it works.

Submit
Submit the *vhd*, *ucf* and *bit* files via blackboard. Additionally, submit the complete synthesis results in a separate text or word file.

Note
The last page on this lab description should be treated as a guide for what aspects of the design the instructor will look at. It is not intended to be a complete list of all requirements for this lab assignment. Please read the lab description carefully to make sure you met all lab requirements.

FPGA Design Using VHDL
525.442 Grading Sheet
Lab 01

(___ / 100)

(___ / 25) Seven segment encoder 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

(___ / 10) Green LEDs light up according to which slider switch is in HIGH position

(___ / 25) Correct Operation of Push Buttons
Four slider switches SW3-SW0 used as input

- With no buttons pressed, digit displayed on the rightmost LED display
- When just BTN0 is pressed, the digit will be displayed on the second from the right LED display (i.e the one controlled by AN1)
- When just BTN1 is pressed, the digit will be displayed on both of the two rightmost displays.
- When both BTN0 and BTN1 are pressed, 0's are to be displayed on ALL FOUR LED displays.

(___ / 20) Well-documented / Appropriate VHDL Code

(___ / 5) Synthesize report with device resource utilization summary

For this lab assignment only, you will be asked to submit the synthesis report file that contains the summary of resource utilization.

For subsequent labs, you will be asked to submit the report file **AND** will be asked to **analyze the resource utilization**. In this course, we will try to emphasize that you need to have a rough idea of what your design will require resource-wise (i.e., flip-flops, multipliers, internal memory, etc.).