

## Simple Seven-Segment Display Controller

For this exercise, you will configure the Xilinx field programmable gate array (FPGA), found on the Spartan III starter board, to act as a simple controller for a seven-segment LED display, also found on the board. The device will take as an input a four (4) bit `std_logic_vector` named **digit**. The output of the module will be a seven (7) bit `std_logic_vector` named **seg7**. This vector will light the segments to form the numeric representation of the value represented by **digit**. For this implementation, if **digit** is in the range [0-9] the value specified will be displayed. If the value of **digit** is outside of this range, an 'E' will be displayed. Use the reference manual for the Nexsys2 board to assist in creating the appropriate characters.

*Note! ISE (as of version 10.1) does not correctly handle spaces in the project path – please store your project in a directory tree without spaces.*

### Task 1

Create an entity/architecture pair called **seg7\_decimal** in a file named **seg7\_decimal.vhd** that performs the above function. This entity should only have 2 ports : `digit`, and `seg7`.

### Task 2

Create a new entity/architecture pair called **lab1\_top** in a file named **lab1\_top.vhd**. This new entity will make use of the decimal 7-segment controller that you just built, and will be the first design that you actually synthesize and load to the starter kit. The entity for `lab1_top` should look like this (if you would like to make the anodes an array, feel free):

```
Entity lab1_top is
Port (
    sliderSwitches : in std_logic_vector(3 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
);
```

The main component of `lab1_top`'s architecture should be an instantiated copy of `seg7_decimal` as built in task1. You will hook it up to perform the tasks as described below:

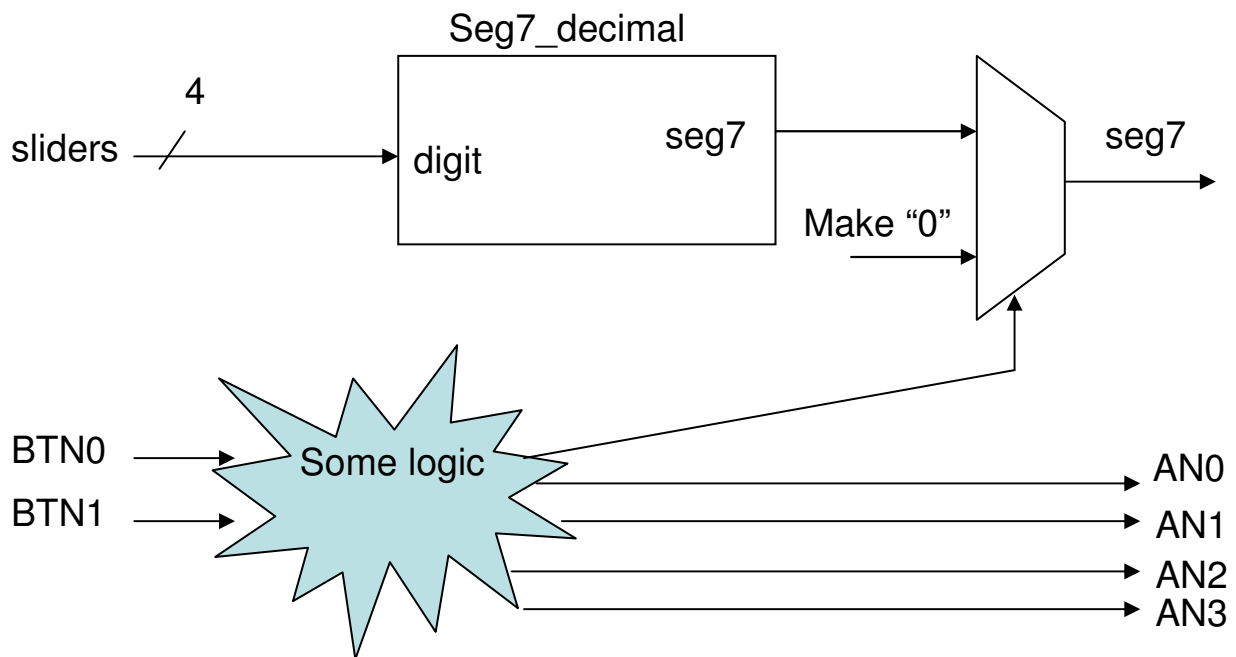
- 1) The input to the 7-segment controller will be the 4 slider switches `sw3-0`. `SW3` is considered to be the MSB of our binary number, and `SW0` the LSB.
- 2) Normally, with none of the pushbuttons pressed, the digit will be displayed on the rightmost LED display.

- 3) 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)
- 4) When just BTN1 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 LEDs displays regardless of the current switch values.

A small block diagram of how this might be implemented is attached. Your design should consist of two entity/architecture pairs: `seg7_decimal`, and `lab1_top`, no others. (In other words, a block on the block diagram doesn't necessarily mean a new VHDL entity, just some logic)

### Task 3

Using the constraints editor set the pin assignments for the ports of `lab1_top` to the appropriate Xilinx pin numbers. Synthesize, generate a programming file, and upload this file to the board using Export. Test your design to verify that it works.



### Submit

There will be no demonstration for this project, thus the only required submission is a zipped electronic archive of your entire project directory. File should be named **lab1\_yourlastname.zip**. The project should include the bitfile, and should be uploaded via the web link on the class webpage prior to 4:25 p.m. the day of class.

VHDL/FPGA Microprocessor Design  
525.442  
Grading Sheet, Assignment 1

( \_\_\_ / 100)

( \_\_\_ / 50) Seven segment decoder module correctly implemented

( \_\_\_ / 25) Correct Operation of Entire Program

- 4 Slider Switches 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) Summary of the design's synthesis results. In this class, 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.). This summary should simply seek to explain what resources were used, and why. This does not have to be elaborate, but it should not simply be a copy of the synthesis results.