Design Project Overview:

Th de Lab 4 design project is worth 2 Lab grades due to the quantity of work required. The primary purpose of this lab is to provide you practice in designing more complex systems that integrate sequential and combinatorial building blocks. In this lab, you will test and debug the basic elevator controller built during the in-class exercise by synthesizing and configuring a FPGA with your Moore state machine. You will then incrementally add features to your design that build up to a fully featured elevator controller system. In order to accomplish this, you will need to integrate modules from previous in class exercises and labs. **A key note: start early and do not delay!** Early delays will be hard to recover from.

Authorized Resources: For this lab, you may work in teams of up to two (including the prelab). You may seek help from any cadet or instructor, not limited to this course, and reference any publication in its completion. Online resources are acceptable. However, no resources containing solutions to the course homework, labs, exams, quizzes, or in class/out of class exercises are allowed. Your work (and your code) must **always** be your own. Normal documentation of all resources utilized is required.

Due Dates: Prelab – 1700 T27 via Gradescope

Minimum Functionality Demo – 1700 T30 in person or via video link

Minimum Functionality Demo (1700 T30 in person or via video link)

The initial phase of your design will be to implement MINIMUM functionality. The minimum functionality is what is required to consider the lab accomplished for passing the course. However, the minimum functionality will only get you a 60% of the total points assuming you receive all the points for the final deliverables (report, code, etc). Figure 1 below shows the BASYS3 board interface that you will use for MINIMUM functionality. As can be seen in the figure, your elevator's floor should be shown on seven-segment display 2 (Display 0 is the far right display). This should be updated according to the output of your basic elevator controller at a clock rate of **2 Hz**. Your elevator must not teleport! Three push buttons are to be used for resets as shown below. The **Master Reset** button should reset the **FSM** AND the **clock**.

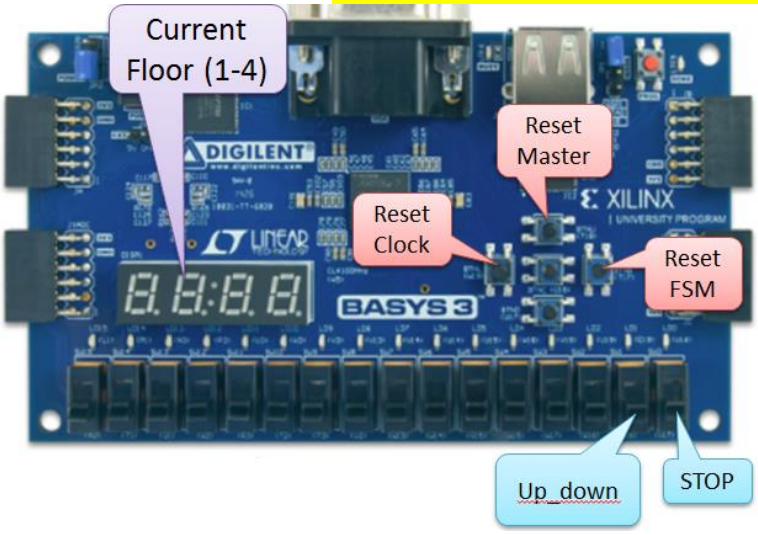

Figure 1. MINIMUM functionality interface

Prelab tasks (1700 T27 via Gradescope)

1) Provide a top-level schematic showing the architecture of how you will implement MINIMUM functionality (draw.io, PowerPoint, etc – do not hand draw). The expectation is you will update this schematic at the end of this lab to reflect your final design.

    a. Note, you will need to include several components in this schematic. Please use the top-level entity interface found in the **top_basys3.vhd** file included with the Lab 4 files (notice there are signals you will not use for minimum functionality, leave them in the file as they are included for the additional functionality, and include them in your prelab schematic). You need to include a clock divider module, and the remaining modules you need will come from your previous in-class exercises and labs (MooreElevatorController and sevenSegDecoder).

    b. Label all wires not connected to the top-level entity ports. They will be additional signals you will create in your VHDL code.

    c. Only turn on seven-segment display 2 (display 0 is the far right). Force the others OFF by disabling their anodes.

    d. Turn OFF (ground) all LEDs, since you do not need them for MINIMUM functionality, but they are in the entity for later functionality.

    e. Just like in the last lab, the clock divider uses *generics* so that you can redefine characteristics at instantiation. For instance, you can redefine the number of times the clock divider divides a clock for each instance of the module. Be sure to specify the value for your clock divider to obtain 2 Hz. For an idea on how to accomplish this, look at the generic mapping of the clock divider in ICE4 and Lab 3.

    f. All component names should match the labels given to them (ex: MooreElevatorController should be labeled MooreElevatorController)

    g. All components should have their internal ports labeled (So the reader knows what signals are being connected)

Lab Tasks

Part 1: Minimum Functionality

1) Create a Vivado project called **Lab4**.

2) Create a **code** folder and add copies of all of the code files you will need for your design
   a. Do not forget to add the files to your project

3) Using your Prelab drawing, complete the top-level module
   a. Remember this basically involves:
      i. Copying in component definitions
      ii. Creating additional wires
      iii. Instantiating your component instances and connecting all wires

   b. For the clock divider, what value for `k_DIV` do you need to produce a 2 Hz clock?
      i. If desired, you can test it with the provided testbench.

4) Create your XDC file from the XDC template, and uncomment of the hardware support you will need per your top_basys3.

5) When you synthesize and later implement your design you may receive several warnings about unconnected wires or ports. Ignore these.

6) Test that your design works as expected.

   a. Once you have verified all functionality, demo your MINIMUM functionality to an instructor.
   b. While not required, feel free to create a test bench for your top_basys3 if it doesn't appear to be working correctly. You will need to create an artificial clock (which you have done to test FSMs before)
   c. **Does your master reset work as intended? Think about how your basic elevator controller implements a reset. If you had to fix something, explain what you did in your report.**

7) If you have not already done so, create a **bitFiles** folder inside the same folder where your **code** folder resides. The bitFiles folder will store all of your .bit files.

8) Change the name of your .bit file from top_basys3.bit to **LAB4_MINIMUM_functionality.bit** and move it to your **bitFiles** folder.
   a. This will ensure your bit file doesn't get overwritten!

9) Commit and push your work!

10) Demo your functioning design to an instructor.