

Homework 1

Submission Instructions:

- You are required to submit **BOTH** demo videos and VHDL codes to Blackboard.
- Create each VHDL project with a project name based on the lab and question numbers, e.g. “**ceng2010_hw1_q1**”.
- Zip all the project folders to one single zip/rar file named with your student ID number, e.g. “**1155123456.zip**”.
- Upload the zip/rar file to Blackboard before the deadline stated in Blackboard
- Zero mark will be given if you ask for help from others such as Classmates/TA/Instructor. Hints are given on next page.
- Marks will be deducted for late submission, deduct 10 marks per every 30-minute interval

You are required to record a short mp4 video to demonstrate the answers. In the video, the following elements are required:

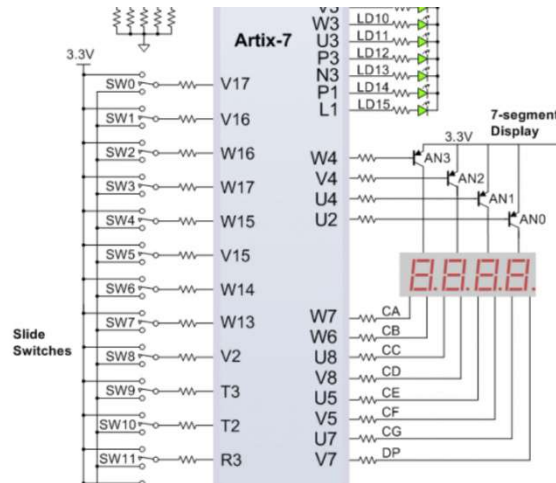
- A. Next to your FPGA board, show your full name and SID on a paper [5 marks]
- B. Voice descriptions in English/Cantonese/Mandarin on what you are doing [5 marks]
- C. Demonstrate works by presenting all possible input combinations step-by-step clearly [30 marks]

1. Using VHDL to implement a shifting 7-segment display that will rotate the digits to the left or right according to the user inputs. The requirements are as follows:

- a. Use the last digit of your SID to define your own four-digit pattern, for example: [10 marks]
 - i. If the last digit is ‘6’, the pattern is ‘6789’.
 - ii. If the last digit is ‘9’, the pattern is ‘9012’, and so on.
 - iii. When the system is being initialized/reset, this pattern will be shown.
- b. Provide control on reset and rotating the pattern to the left or right. For example, if the original pattern is ‘6789’... [10 marks]
 - i. Rotating to the left means ‘6789’ => ‘7896’ => ‘8967’ => ‘9678’ => ‘6789’ => ...
 - ii. Rotating to the right means ‘6789’ => ‘9678’ => ‘8967’ => ‘7896’ => ‘6789’ => ...
- c. There are two options of implementation requirements for you to choose. **For different options, please noted that you will get different maximum scores.** You are required to implement **only one** of the following options. [20 marks for Option 1 **OR** 30 marks for Option 2]
 - i. **Option 1**
 - 1. Use btnD to reset the pattern.
 - 2. Use btnC to trigger the rotating process. By pressing the btnC button once, the pattern will rotate for one digit.
 - 3. Use sw0 to control the rotation directions. If sw0 is 1, the pattern will rotate to the right. If sw0 is 0, the pattern will rotate to the left.
 - ii. **Option 2**
 - 1. Use btnC to reset the pattern.
 - 2. Use btnL to rotate the pattern to the left for one digit.
 - 3. Use btnR to rotate the pattern to the right for one digit.
- d. For both options, the rotations have to be clear and **stable**. This means that only one digit should be clearly rotated for one single pressing action. If it does not rotate or rotate more than one digit, marks will be deducted. [10 marks]

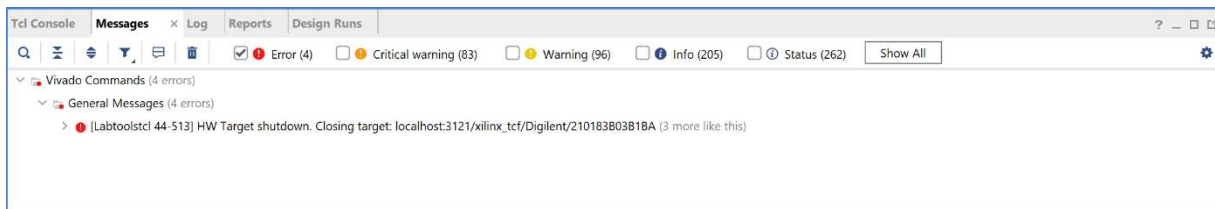
Hints:

- Include the **basys3_hw1.xdc** constraints file to your project for modifications (if necessary).
- If some arithmetic functions with Signed or Unsigned values are used, you should include the following library:
 - **use IEEE.NUMERIC_STD.ALL;**
- As shown in the following circuit diagram of Basys3 board, <https://reference.digilentinc.com/reference/programmable-logic/basys-3/reference-manual>. The data pins (i.e. CA, CB, CC, CD, CE, CF, CG, DP) are commonly connected to all four digits of the 7-segment display. To display a number onto a particular 7-segment, you should activate the anode activation pin (i.e. AN0, AN1, AN2, AN3) of that particular 7-segment, as we did in the previous labs. In other words, if we need to display different numbers on different 7-segments “at the same time”, a refresh mechanism is required. The mechanism will activate only one 7-segment at a time with a very high refresh rate, so the human eyes will treat that different numbers are displaying on different 7-segment at the same time. For more info, please refer to: <https://www.fpga4student.com/2017/09/vhdl-code-for-seven-segment-display.html>



- Please noted that the “nested clocked statements” as below are **NOT** supported in VHDL.

```
if rising_edge(clk1) then
    if rising_edge(clk2) then
        ...
    end if;
end if;
```
- If you have encountered any problems, check the **Messages** windows for **Error** messages and debug them one-by-one.



THE END