Cpr E 281 LAB11
ELECTRICAL AND COMPUTER ENGINEERING
IOWA STATE UNIVERSITY

Finite State Machines

PRELAB!

Read the entire lab, and **complete** the prelab questions (Q1-Q2) on the answer sheet **before** coming to the laboratory.

1.0 Objectives

In this lab you will create circuits that function as finite state machines. Read Chapter 6 of your textbook and complete the prelab before you come to the lab.

2.0 A Simple Counting Device

For this step, you are to design a device that simply cycles through six states. It will have an input for the clock, an input ***w*** which will keep the current state if low and advance to the next state if high, and the output will be the present state. The device is a modulo-6 counter. The design method will be similar to that for the modulo-8 counter in Section 6.7 of your textbook.

Fill in the state-assigned table on the answer sheet and use it to design the circuit.

Use D flip-flops for the device's memory. The ***w*** input is assigned to a toggle switch. The output needs to be represented on a seven-segment display.

The clock signal for the circuit comes from a 50MHz clock on the DE2-115 board which can be connected from **PIN_Y2, PIN_AG14, or PIN_AG15** on the FPGA. This clock rate is obviously too high to ever see a transition of an illuminated display with the naked eye; to fix this problem, the clock rate must be reduced down to something with a much larger period. A **clock_generator** has been provided with the lab files. This module will use the 50MHz clock signal and reduce it to a clock signal with a period of about 2.68 seconds. Double click on the **clock_generator** symbol and study it. On your answer sheet, explain how this module works (an equation to change 50MHz (freq) down to ~2.68s (period) will suffice). *Hint: there is a reason for the clock_divider_1024 having the value 1024 instead of 1000.*

Create a new project and use the DE2-115 board to verify your design. When you are convinced your design is working correctly, demonstrate your design to the TA.

3.0 A Simple Counter

You will now design a different kind of counter. Again, modeling this device as a finite state machine, create a counter that repeatedly counts 0, 2, 4, 5, 0, 2, 4, 5, and so on. This device will have an input ***w*** which, as in Sec. 2.0, will keep the current state if low and advance to the next state if high.

Fill in the state-assigned table on the answer sheet and use it to design the circuit.

Use D flip-flops for the device's memory. The ***w*** input will need to be assigned to a toggle switch. The output needs to be represented on a seven-segment display.

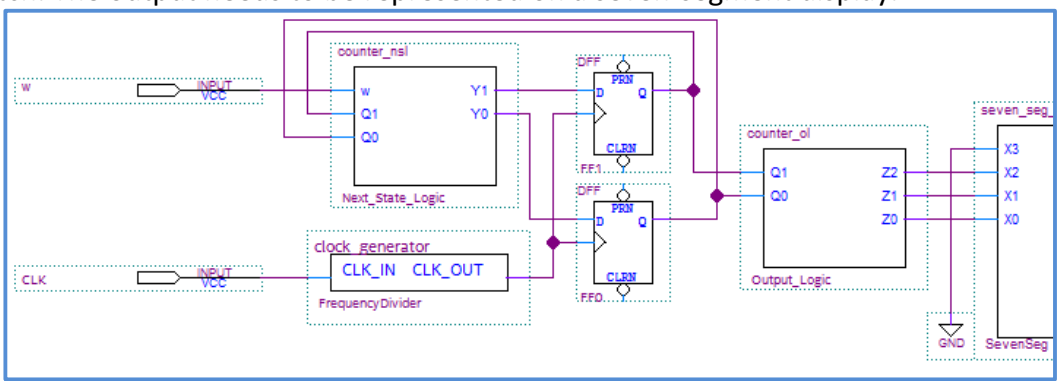

Figure 1: A simple counter

Create a new project and use the DE2-115 board to verify your design. When you are convinced your design is working correctly, demonstrate your design to the TA.

After you have completed Sec. 3.0, remove the clock_generator from the design file, connect the clock to the D Flip-Flops, and then create a symbol file for step3.

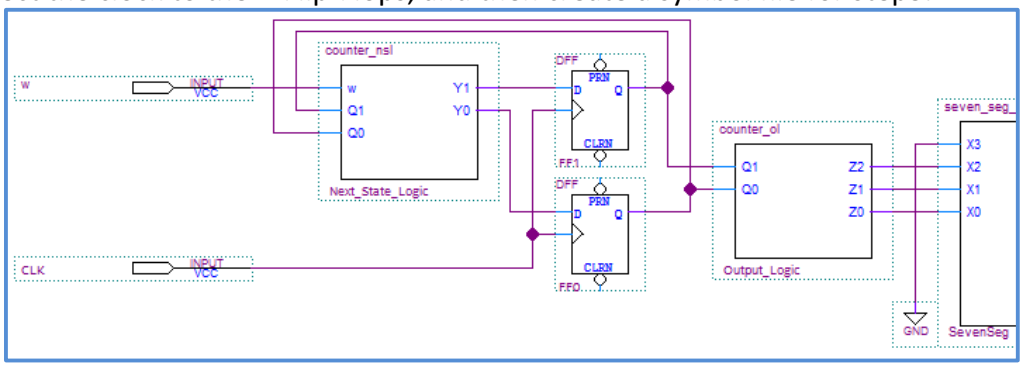

Figure 2: A simple counter with the clock_generator removed

4.0 Switch Debouncing

When working with switches, moving the switch through an On/Off transition or an Off/On transition may cause the switch to bounce, i.e., the output of the switch may not be stable for a period of time (see Figure 3 below). In this part, you will implement one way to debounce a switch using components you are familiar with. This debouncer will use the built-in board clock to create an automatic delay that is longer than the time the switches can bounce for.

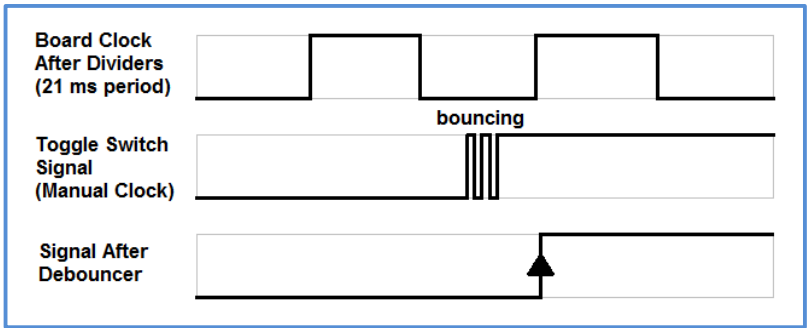

Figure 3: Bouncing of the output signal and debounced output signal

The debouncer will have two inputs (the board clock and a manual clock) and one output (the debounced signal). The board clock is the 50MHz clock that you used in the previous step. The manual clock is a clock that the user controls directly with a toggle switch, for instance. The output signal is a clock that will transition once every ~20 ms, but only if the manual clock has been toggled. Use two **clock_generator_1024** blocks to increase the board clock period to ~20 ms. Use a D flip-flop in your design to have the output signal update only once for every clock cycle. The **clock_generator_1024** blocks are used (together with the board clock) as the clock signal for the D flip-flop to store the user input value (manual clock value) in order to output an updated value for the user input every ~20 ms (debounced signal).

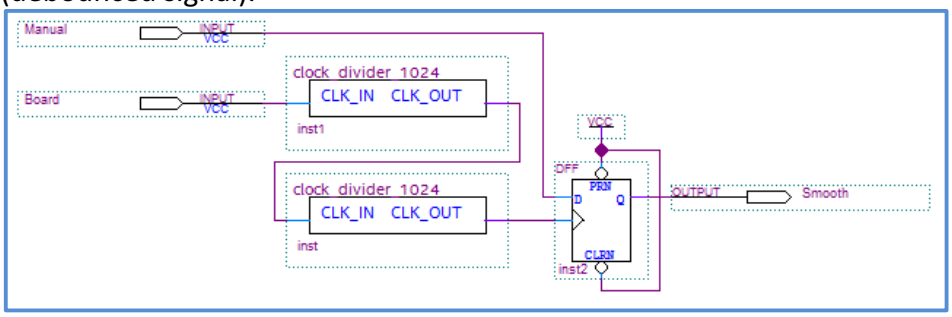

Figure 4: Debouncer Circuit

Once you have created the debouncer circuit, create a symbol for the debouncer. To test the debouncer circuit, first open a new .bdf file in a new project. Then, place two symbols of the counter you built in Sec. 3.0 along with a debouncer circuit that you built in the previous step. One of the counters will have the unaltered input from the switch and the other counter will have the debounced input. Each output should be assigned to a seven-segment display. When complete, your circuit should have one output (via the debounced circuit) that consistently makes only one transition for every positive edge from your manual clock (toggle switch), whereas the other output may sometimes skip numbers due to its bounced output signal.

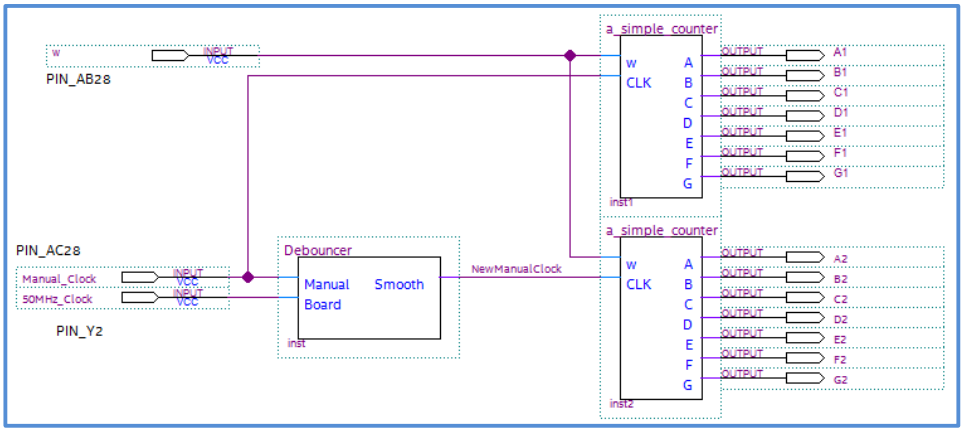

Figure 5: Debouncer In Use

Create a new project and use the DE2-115 board to verify your design. When you are convinced that your design is working correctly, demonstrate it to the TA.

5.0 Complete

You are done with this lab. Close all lab files, exit Quartus Prime, log off the computer, power down the DE2-115 board, and hand in your answer sheet. **Don't forget to write down your name and your lab section number**.