Finite State Machines

PRELAB!

Read the entire lab, and **complete** the prelab questions (Q1-Q3) 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 the 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 to **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 using the value 1024 instead of 1000.*

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 the 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.

Use the DE2-115 board to verify your design. When you are convinced your design is working correctly, demonstrate your design to the TA.

4.0 Switch Debouncing

When working with switches the On/Off transition and the Off/On transition may cause the switch to bounce, i.e., the output of the switch may not be settled for some time. In this part you will implement one way to debounce a switch. This device will use the built-in board clock to create an automatic delay that is longer than the time the switches can bounce for.

The debouncer will have two inputs (the board clock and a user input clock) and one output (the debounced signal). Use two **clock_generator_1024** blocks and a D flip-flop in your design. Use the **clock_generator_1024** blocks to delay the board clock and the D flip-flop to store the value of the user input.

To test the debouncer circuit, place two symbols of the counter built in Sec. 3.0. One will have the unaltered input from the switch and the other will have the debounced input. Each output should be assigned to a seven-segment display.

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**.