SANTA CLARA UNIVERSITY	ELEN 21 Spring 2017	Dr. Sally Wood, Dr. Radhika Grover

Laboratory #8: Counters

For lab sections May 30-June 2, 2017

I. OBJECTIVES

In this laboratory you will:

- Use D-type flip-flops to build both a ripple counter and a synchronous counter.
- Use a decoder to full decode a counter's outputs into individual count indicators.

PROBLEM STATEMENT

An n-bit binary up counter will produce a sequence of outputs in binary counting order. For example a 3-bit up counter would produce the sequence 000, 001, 010, 011,100, 101, 110, 111, 000, 001, 010 011, etc. The counter should have a clock input to determine when the count should change and an enable input to either prevent or allow counting.

A counter requires memory to operate since the next value is determined by the previous "remembered" value. A flip-flop stores one bit of data when an active clock edge appears at the clock input. The most common type of flip-flop is the D-type, which simply transfers the value of the D input to the Q output when the active clock edge occurs.

For a ripple counter, a flip-flop output Q_i will only change when the flip-flop output Q_{i-1} changes from 1 to 0. The change in value of the less significant bit controls the clock input for Q_i.

For a synchronous counter, all flip-flops have a common clock and the flip-flop output on an active clock edge is determined by its D input. For a synchronous up counter, the output of a flip-flop changes on an active clock edge only when all of the less significant bits have reached their maximum count before the clock edge.

II. PRE-LAB

(i) Draw a schematic for a Toggle flip-flop designed using a D-type flip-flop. When input T=0, the flip-flop output should not change on an active clock edge so Q(t+1) = Q(t). When T=1, the flip-flop output should change on an active clock edge so Q(t+1) = (Q(t))'.

(ii) Design and draw a schematic for a three-bit ripple up counter using positive edge triggered D-type flip-flops. Include a CountEnable signal the will prevent counting on active clock edges when it is 0 and will allow counting on active clock edges when it is 1. Clearly label your circuit.

(iii) Design and draw a schematic for a three-bit synchronous up counter using D-type flip-flops. Include a CountEnable signal as you did for the ripple counter. Clearly label your circuit.

III. PROCEDURE

Part 1: Counter Implementation and Testing

Schematic:

1. Extend your prelab ripple counter from three bits to four bits and make a Quartus schematic for a four-bit ripple counter that counts up on a rising clock edge when the CountEnable input is 1 and stays at the same value when the CountEnable input is 0. You will need four independently clocked flip-flops for your circuit. Find libraries→other→maxplus2→7474, which is a part designed to implement a standard TTL component that had two independent D-type flip-flops. Note that the D, CLCK, Q, QN, PRN and CLRN inputs have a "1" for one flip-flop and a "2" for the other.
2. Extend your prelab synchronous up counter from three bits to four bits and make a Quartus schematic for a four-bit synchronous up counter. Although you could use two 7474s as you did with the ripple counter, since this is a synchronous counter and all flip-flops see the same clock, you can also use the libraries→other→maxplus2 *8dff* component that you used as a data storage register in the minicalculator lab.

Inputs/Outputs:

3. Connect an input switch to the CountEnable input of both of your up counters.
4. Copy the slot machine inspired game 8-bit up-counter and "clock counter" components to your circuit to make a clock divider that will reduce the rate of the fast system clock50 to a slow rate so that you can view the operation of your circuit. Select one of the "slow mode" outputs, e.g.q3, as the clock input for both of your 4-bit counters. If you find this is too slow or too fast during testing, select q2 or q4 or another output from the up counter clock divider.
5. Connect the clock input of your counters to an LED for observation during debugging.
6. Connect the 4-bit output of each of your up counters to a seven-segment display.
7. To make debugging your circuit easier, it is suggested that you connect the D inputs of all of your flip-flops to LEDs for observation.
8. Be sure that all inputs to your components are controlled and are not left floating. Double check to be sure that preset and clear inputs are not active since these will override all other inputs.

Testing:

9. Download your circuit to the FPGA.
10. With the CountEnable switch set to 0, observe the seven-segment displays and the LEDs. Nothing should be changing except the clock input to your two counters.
    a. If the clock input to your two counters in NOT changing, correct that problem first since without a clock input the circuit will not operate.
    b. If the seven-segment displays are changing even though CountEnable = 0, correct that problem.
11. With the CountEnable switch set to 1, check that the counting operation is correct.
    a. If the values in the seven segment display are not changing, correct that.
    b. If the values are not changing when the clock changes from 0 to 1, correct that.
    c. If the values are changing, but not changing in the correct order, debug your circuit. Check that the flip-flop inputs are correct. Make sure that the least significant bit of your counters is also the least significant bit in the 4-but word controlling the seven-segment display.

d. Note that the counters as designed did not have a reset capability, so your counters may start at different values. However, they both should always count up and they both should change values at the same time.

12. When your counters are functioning correctly, demonstrate then to your lab assistant.

Part 2: Fully Decoded Sequencer

13. Modify your circuit to add a 3:8 decoder to decode the three least significant bits of the synchronous counter output. In the libraries find libraries→other→maxplus2→dec38 and add it to your circuit.
    a. Connect the SENSE input to a logic 1 for active high decoder outputs rather than active low outputs.
    b. Connect INHB to a logic 0 since inhibit is the complement of enable.
    c. Connect input selects s0, s1, and s2 to the three least significant outputs of your synchronous up counter.
    d. Connect outputs y0 to y7 to eight LEDs with y0 on the left and y7 on the right. (Since you are not modifying your counter circuit and it is already working correctly, you can remove the LED connections to the flip-flop D inputs if you used them earlier.)
14. Download your modified circuit to the FPGA.
15. What do you expect to see on the 8 LEDs that are the decoder outputs? If the pattern in not correct, debug your circuit and then demonstrate it to your Lab TA.

Part 3: Extra Credit

16. If time permits, for extra credit add a reset capability for both of your four-bit counters and demonstrate it.

IV. REPORT

1. Introduction, procedure, results, conclusions and references.
2. Include schematic, test plan, and results for each operation.
3. How would you modify your counters to extend to 8 bit counters? Indicate what additional components would be needed for the circuit and write the logic equations for the added flip-flop inputs.
4. How would you modify your counters count either up or down with an input switch controlling the direction? Indicate what additional components would be needed for the circuit and write the logic equations for all the flip-flop inputs.