Cpr E 281 LAB10
ELECTRICAL AND COMPUTER ENGINEERING
IOWA STATE UNIVERSITY

Designing Registers and Counters

PRELAB!

Read the entire lab, and **complete** the prelab questions (Q1-Q5) on the answer sheet **before** coming to the laboratory.

1.0 Objectives

In this Lab, you will design shift registers and counters, and you will observe their functionality.

2.0 Shift Register

A shift register is a specialized register that takes an input serially and shifts it from one bit position to the next bit position. A full explanation of shift registers is available in Chapter 5.8 of the textbook, where Figure 5.17a shows a 4-bit shift register built using D flip-flops.

For this step in the lab, you will design a 4-bit shift register using D flip-flops. You may use Figure 5.17a as a reference. You will use a D flip-flop of the **primitives** library to build a shift register. Create a new project ***lab10step1*** and open a new .bdf file and name it ***lab10step1***. Insert a symbol for the D flip-flop (*dff*). Connect the first DFF to input pin ***IN***. Set the preset ***PRN*** and the clear ***CLRN*** inputs of this flip-flop to high. One way to do this is to wire both ports to a **VCC** symbol. In addition, to observe the output, you will have to add an output pin for the output of each D flip-flop, label the first ***Q1***, second ***Q2***, third ***Q3*** and the last ***Q4***.

Import a Verilog file of your shift register into Questa ModelSim and verify your shift register is operating correctly. Observe that the values on the outputs of the DFFs may be an unknown value 'x', represented by red bars on the timing diagram. These values should return to known green values once each DFF signal receives a definite 0 or 1 on its input. If the output is unknown for too long, the input may not be connected properly, so check your circuit for connection problems. Once your waveforms are satisfactory, fill in the sequence table on the answer sheet and demonstrate your circuit to the TA.

3.0 Counters

3.1 Synchronous Up-Counters

Counters are either synchronous (common clock signal) or asynchronous. A discussion of synchronous counters may be found in Chapter 5.9.2 of your textbook. Figure 5.23 describes how a 4-bit counter can be built using D flip-flops.

Create a new project ***lab10step2a*** with a new .bdf file also named ***lab10step2a.bdf***. Use the same **D** flip-flops you used in the last step, as well as some additional gates, to build a synchronous 4-bit up-counter. Use the Figure 5.23 in your text as a reference.

As mentioned in the previous step, the output for DFFs will be unknown ('x') if their inputs are also unknown, so it is important to initialize these values. This is especially important for counters, which must use their previous value to determine their next value. To achieve a normal count, you will use the asynchronous clear to initialize the output of each DFF. After you have created your 4-bit counter, be sure to add input pins **Enable** and **Clock**.

Then, add the active-low input **CLRN**, and connect this to the CLRN input for each DFF. The purpose of **CLRN** will be to reset the up-counter back to 0. Before we can observe any meaningful output of the counter, we must first reset the counter by setting **CLRN** to 0 and then back to 1.
For preset, connect VCC to the PRN input for each DFF as we have no need to preset this clock. Finally, connect all four outputs together to a single output pin as a bus (similar to how we did this in Lab8) named **Q[3..0]**. Ensure that the least-significant bit Q0 receives the correct bit.

Use ModelSim to verify your up-counter is working properly. Once you are confident your circuit is functioning properly, demonstrate your circuit to the TA.

Create a new project ***lab10step2b*** and open a new .bdf file and name it ***lab10step2b***. Use **T** flip-flops, and additional gates, to build a synchronous 5-bit up-counter. Use Figure 5.22 in your text as a reference and extend the counter to five bits. You can get T flip-flops (*tff*) from the **primitives** library the same way you got the D flip-flops. Set the preset inputs of these TFFs to VCC and set the clear inputs to **CLRN**. Don't forget to add **Enable** and **Clock** so that you can use the same DO file in Questa to demonstrate your results.

Use ModelSim to verify your up-counter is working properly. Once you are confident your circuit is functioning properly, demonstrate your circuit to the TA.

3.2 Asynchronous Counters
In this step ***lab10step3a***, you will build a 4-bit asynchronous **up**-counter. Figure 5.19a in your textbook shows a 3-bit asynchronous up-counter using T flip-flops. Extend this circuit to build a 4-bit asynchronous up-counter using JK flip-flops. This requires that, instead of connecting your VCC to just three sets of PRN and T inputs, your VCC will now connect to the four sets of PRN, J, and K inputs of each JK flip-flop.

Note: Mechanical switches have a bad electrical property called bouncing. As the two metal conductors in the switch get close to each other an electrical charge can jump across the gap between the conductors. This can happen several times before the two conductors make complete physical contact. This may cause a digital circuit to react as if the switch was opened and closed multiple times. An engineer must remove this property (debounce the switch) in order to achieve the desired action of the switch. This can be done using hardware in the

circuit, or by adding a delay to switch reads in software. For this lab, you just need to be aware of the problem and make sure that with every manual clock pulse you are not seeing more than one action from your circuit.

Create a new project and use either the **jkff** object or the **jkff2** object (which comes with its own QN output port) from the primitives in your design. Use the DE2-115 board to verify your up-counter is operating correctly. Use a pushbutton for the ***Clock*** pin, a rocker switch for **CLRN**, and use a **seven segment** display (from Lab 5) to represent the outputs. As you press the pushbutton, observe how the output counts and how reliably the hardware successfully counts to the next number. Once your circuit is functioning properly, demonstrate your circuit to the TA.

In this step ***lab10step3b***, you will build a 4-bit asynchronous **down**-counter using JK flip-flops. Figure 5.20a in your textbook shows a 3-bit asynchronous down-counter using T flip-flops. Like the previous step, extend this circuit to build a 4-bit asynchronous down-counter using JK flip-flops. Instead of connecting VCC to PRN and a rocker switch to CLRN, you will connect VCC to CLRN and a rocker switch to PRN.

Create a project just as you did for the previous step and use the DE2-115 board to verify your down-counter is operating correctly. Use a toggle switch for the ***Clock*** pin, a rocker switch for **PRN**, and use a **seven segment** display to represent the outputs. Observe the counter reliability as you did for the up-counter. Once your circuit is functioning properly, demonstrate your circuit to the TA.

4.0 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 answers. **Don't forget to write down your name and lab section number**.