## PRELAB!

**Read the entire lab and complete the prelab questions (Q1-Q5) on the report template and submit your completed questions on Canvas before your lab time. You will submit this report again once you have completed the lab.**

## 1.0 Objectives

In this Lab you will design shift registers and counters, and 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. 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 the 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*). 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.

Use the ModelSim to verify your shift register is operating correctly. Create an output pin for the output of each D flip-flop, label the first ***Q1***, second ***Q2***, third ***Q3*** and the last ***Q4***. Fill in the sequence table on the report and include y screenshots of your circuit in your lab report.

## 3.0 Counters

**Important note (counter initializing):**

For this section (counters) you have to pay attention to the initial values of flip-flops. Counters start counting from an initial value. If they don’t have a starting point, they will not be able to count!

To set an initial value for the flip-flops, two asynchronous inputs. **CLRN** and **PRN** can be used. Remember the result from the step1 when they were connected to **vcc**.

**In this section for each step there have to be inputs for PRN and CLRN of the flip-flops to initialize them.**

Then by setting the CLRN to zero, flip-flop’s output will change to 0. And by setting PRN to zero flip-flop’s output will change to 1. **After initializing CLRN and PRN are set back to 1** so that they stop changing the output!

# 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*** and open a new .bdf file and name it ***lab10step2a***. Use the same **D** flip-flops you used in the last step, and additional gates, to build a synchronous 4-bit up-counter. Use the Figure 5.23 in your text as a reference.

Don’t forget that we have to initialize the counter. To initialize the counter, connect all the PRNs to PRN input and all the CLRNs to the CLRN input. You would have 4 inputs; **Enable, Clock, PRN, CLRN.**

Use ModelSim to verify your up-counter is operating correctly. Once you are confident your circuit is functioning properly, include it screenshots of the circuit and the simulations in your lab report.

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 4-bit up-counter. Use Figure 5.21a in your text as a reference. You can get T flip-flops (*tff*) from the **primitives** library the same way you got the D flip-flops. To initialize the counter, connect all the PRNs to PRN input and all the CLRNs to the CLRN input. You would have 4 inputs; **Enable, Clock, PRN, CLRN.**

Use ModelSim to verify your up-counter is operating correctly. Once you are confident your circuit is functioning properly, include it screenshots of the circuit and the simulations in your lab report.

# 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. You can use an inverter on the Q output to get the ~Q from the flip-flops. To initialize the counter, connect all the PRNs to PRN input and all the CLRNs to the CLRN input. Lastly, instead of a constant 1 for each tff use an input named Enable. You would have 4 inputs; **Enable, Clock, PRN, CLRN.**

Use ModelSim to verify your up-counter is operating correctly. Once you are confident your circuit is functioning properly, include it screenshots of the circuit and the simulations in your lab report.

In this step ***lab10step3b***, you will build a 4-bit asynchronous **down**-counter using T flip-flops. Figure 5.20a in your textbook shows a 3-bit asynchronous down-counter using T flip-flops. Extend this circuit to build a 4-bit asynchronous down-counter. To initialize the counter, connect all the PRNs to PRN input and all the CLRNs to the CLRN input. Lastly, instead of a constant 1 for each tff use an input named Enable. You would have 4 inputs; **Enable, Clock, PRN, CLRN.**

Use ModelSim to verify your down-counter is operating correctly. Once you are confident your circuit is functioning properly, include it screenshots of the circuit and the simulations in your lab report.

# 4.0 Complete

You are now done with this lab. Close all lab files, exit Quartus Prime, log off the computer and submit your report to Canvas. **Don’t forget to include your name and your lab section number.**