

ECE 2700 Lab 5

Due the Week of March 4, 2013 at the beginning of your session (100 points)

Objective

To understand how to build and use registers in Verilog.

1 Preparation

1. Read this document in its entirety.
2. Design a 4-bit register using D flip-flops.
3. Design the circuit in Exercise 3.19 from the main textbook. See details below.

2 D Flip-Flop

Inside your ECE2700 directory, create a new folder called **Lab4**. Start up Xilinx ISE and create a new project called **Reg** inside the **Lab4** directory.

Create a new file called **dflipflop.v**. Inside, design a D flip-flop. The inputs are the clock signal *Clk*, hold signal *Hold*, and data input *D*. The output is *Q*. The hold signal is used to ensure no change is made to the stored value even if the clock signal is high and is a custom functionality requested by, say, your client. The D flip-flop also has two **asynchronous** inputs: *Rst*, which resets the D flip-flop to 0, and *Set*, which set the D flip-flop to 1. Note that reset has priority over set and hold, while set has priority over hold. Use the following declaration: `dflipflop d1(Clk, Hold, Rst, Set, D, Q);`

Test your design using **dflipfloptest.v**, which is available on Canvas. Make sure you understand what the test cases are supposed to do. Feel free to modify the test cases. Show your working design to the TA.

3 4-bit Register

Inside a new file called **reg4bit.v**, design a 4-bit register using instances of the D flip-flop you designed earlier. **You are required to use vectors in this exercise.**

Test your design using simulation. It may be easier to follow the test cases in **dflipfloptest.v** and make changes accordingly. Show your results to the TA.

4 Using Registers in a Combinational Circuit

Complete Exercise 3.19 from the main textbook. You can assume the input is 4-bit long instead of 8-bit long. You are **required** to use the 4-bit registers you designed earlier. The 4-bit 4×1 mux is provided in Canvas (**mux4bit4x1.v**). Note that you should design your circuit in such a way that if the control inputs are 00, the oldest value is the output (and if the control inputs are 11, the newest value is the output). **Use vectors.**

Test your design. It may be helpful to declare the output of each register as an output as well so that you can see it in the waveform and debug the result.

5 TA Checkoff

- (10 points) Prelab design.
- (20 points) D flip-flop.
- (30 points) 4-bit register.
- (40 points) Register use in combinational circuit.