

## 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 the following storage elements:

- Basic SR Latch
- D Latch
- D Flip-Flop

The objective is to observe, analyze, and understand the working and the timing behavior of these devices. You will do all the designs in this lab using a schematic block diagram and simulate each in ModelSim. Refer to Sections 5.1 through 5.4, and complete the circuit diagrams before you come to the lab.

## 2.0 Basic SR Latches

Information on a Basic SR Latch can be found in Section 5.1 of the textbook. For the first step **lab9step1**, build a Basic SR Latch with NOR gates using a block diagram/schematic file named **lab9step1a.bdf**. This circuit is given in Figure 5.4a of your textbook. Name the inputs **S** and **R**; name the outputs **Q** and **QN**.

Convert your BDF file to Verilog with the filename **lab9step1a.v**. Use ModelSim with **lab9step1.do** to observe the behavior of this SR NOR Latch. Use the results to complete the characteristic table on the answer sheet. When you are convinced that the latch is performing as predicted, and you are confident you understand its behavior, demonstrate your circuit to the TA.

Create a new BDF named **lab9step1b.bdf** and copy your old SR NOR Latch over to this new file. Since latches and other memory devices can be built using either NAND or NOR gates, build another SR latch by replacing the NOR gates with NAND gates. Change the **R** input to **S** and the **S** input to **R**. Place an inverter on each input to create the  $\bar{S}\bar{R}$  case. Once your design is complete, convert this circuit to a Verilog file with the name **lab9step1b.v**. Use ModelSim and **lab9step1.do** to observe the behavior of this  $\bar{S}\bar{R}$  NAND Latch. Use the results to complete the characteristic table on the answer sheet. When you are convinced that the latch is performing as predicted, and you are confident you understand its behavior, demonstrate your circuit to the TA.

### 3.0 D Latches

In a new project *lab9step2*, build a Gated D Latch using NAND gates and a NOT gate. Refer to Section 5.3 for a detailed explanation of D Latches. Figure 5.7a shows how to build a D Latch using NAND gates and a NOT gate. Once you have built the latch, label the inputs *Clk* and *D*, and outputs *Q* and *QN*. Use ModelSim with the *lab9step2and3.do* file to observe the behavior of the latch. When you are convinced that the latch is performing as predicted, and you are confident you understand its behavior, complete the timing diagram on the answer sheet and demonstrate your circuit to the TA.

Create a **Symbol** for this Gated D Latch as you will need it in the next section.

### 4.0 D Flip-Flops

Refer to Section 5.4 for a detailed explanation of D Flip-Flops.

In a new project (*lab9step3a*), use the Gated D Latch you built to design a Negative-Edge-Triggered D Flip-Flop. Section 5.4.1 discusses this design. Refer to Figure 5.9a. Use ModelSim with *lab9step2and3.do* again to observe the behavior of the flip-flop. When you are convinced that the flip-flop is performing as predicted, and you are confident you understand its behavior, complete the timing diagram on the answer sheet and demonstrate your circuit to the TA.

In a new project (*lab9step3b*), you will design a Positive-Edge-Triggered D Flip-Flop using six NAND gates. Section 5.4.2 discusses this design. Please use Figure 5.11a as a guide for this step. Again, use ModelSim with *lab9step2and3.do* to observe the behavior of the flip-flop. When you are convinced that the flip-flop is performing as predicted, and you are confident you understand its behavior, complete the timing diagram on the answer sheet, and demonstrate your circuit 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.**