EEC 18 **DIGITAL SYSTEMS I** **Fall 2024**

University of California, Davis • Department of Electrical and Computer Engineering

Lab 4: Latches, Flip-flops and Registers

Objective: The purpose of this lab is to investigate latches, flip-flops, and registers.

Pre-lab: Read Roth Sections 11.1–11.4 and other relevant sections of Unit 11.
Analyze the cross-coupled NOR gates of the Transparent (called "Gated" by Roth) SR Latch in Figure 1 with inputs R_g and S_g and outputs Q and QN. Construct a truth table with inputs R_g(t), S_g(t) and Q(t) and outputs Q(t+Δt) and QN(t+Δt), where Δt is the time required for a change of state to occur. Ignore the case when S_g(t) = R_g(t) = 1.

I. Transparent SR Latch

Implement a transparent SR latch storage element built from gates as shown in Figure 1.

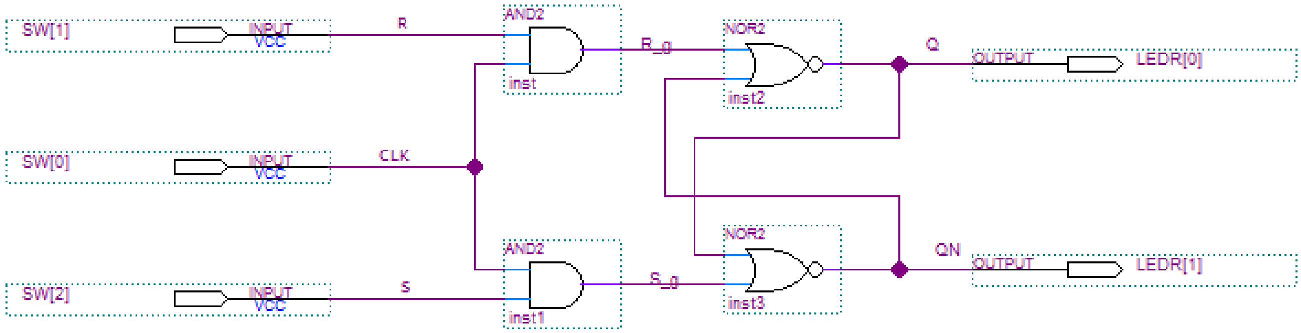

Figure 1. A transparent SR latch circuit

Design Procedure

1. Create a new Quartus project (e.g., **lab4a**) for the SR latch following the same procedure as in earlier labs.

2. Enter the schematic shown in Figure 1.

3. Compile the circuit in Quartus.

4. Download the circuit to the DE10-Lite board and answer the following questions.

   Q1. When the CLK (SW[0]) is high, does the output change when R and S are changed? If so, give examples.

   Q2. When the CLK is low, does the output change when R and S are changed? If so, give examples.

   Q3. Is there any case when Q = QN? (i.e. when LEDR[1] and LEDR[0] are either both ON or both OFF). Describe this case. Since Q and QN should always be complements, this case would be an illegal state for an SR latch.

   Q4. When R=S=0, what happens to the output when the CLK is toggled?

   Q5. When R=S=1 and CLK goes from 1 to 0, what happens to the output? Explain why this happens. Is it what you expect?

<u>Demonstrate your circuit to your TA and have him or her sign your verification sheet</u>. (Since Parts I-III are short, you should get them all checked off at once.)

II. Transparent D Latch

Figure 2 shows the circuit for a transparent D latch. Notice that this circuit has been built using cross-coupled NAND gates instead of cross-coupled NOR gates as in the transparent SR latch in Part I. The circuit in Part I could be easily modified to make a transparent D latch.

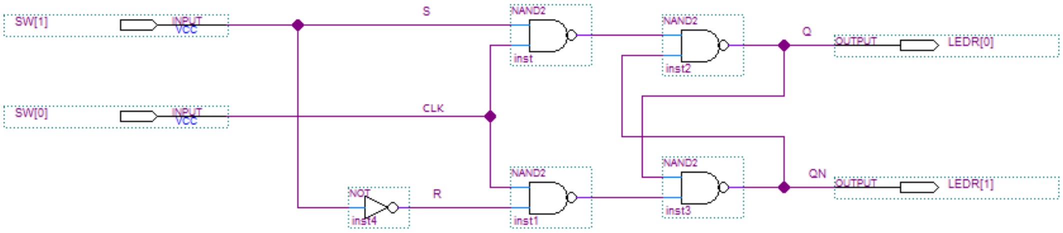

Figure 2. Circuit for a transparent D latch

Design Procedure

1. Create a new project, **lab4b**, for the transparent D latch. Enter the schematic into Quartus, compile and download the circuit to the DE10-Lite board. Answer the following questions:

Q6. When CLK is high, does changing the D input affect the output? If so, give examples.

Q7. When CLK is low, does changing the D input affect the output? If so, give examples.

Q8. Is there any case when Q = QN? (i.e. when LEDR[1] and LEDR[0] are either both ON or both OFF). Describe this case. Since Q and QN should always be complements, this case would be an illegal state for a D latch.

Demonstrate your circuit to your TA and have him or her sign your verification sheet. (Since Parts I-III are short, you should get them all checked off at once.)

III. Edge Triggered D Flip-flop

Figure 3 shows the circuit for an edge-triggered D flip-flop built using two transparent D-latches in "master-slave" configuration. The first D latch (connected to the D input) is the master latch, and the second D latch (connected to the outputs) is the slave latch.

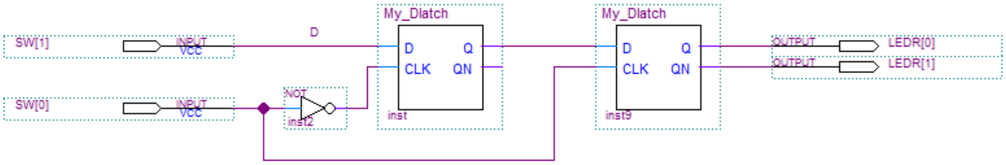

Figure 3. Edge-triggered D flip-flop

Design Procedure

1. Create a new project, **lab4c**, for the edge-triggered D flip-flop schematic.

2. Enter the schematic into Quartus. Create two separate schematics: the top-level design and the transparent D latch. Create a symbol for the transparent D latch. Do not use the name **Dlatch** or similar for the symbol, since there is already a component by this name in the Altera libraries. Use a different name, such as **My_Dlatch**, to avoid unwanted errors.

3. Compile and download the circuit to the DE10-Lite board. Answer the following questions:

Q9. When CLK is high, does changing the D input affect the output? If so, give examples.

Q10. When CLK is low, does changing the D input affect the output? If so, give examples.

Q11. When does the output change? How is this different from a transparent D latch

Demonstrate your circuit to your TA and have him or her sign your verification sheet.

IV. 12-bit Register and Hex-to-Seven-Segment Converters

A *register* is a group of memory elements (typically edge-triggered flip-flops) which have their clock signal inputs tied together, and thus act as a single memory storage block operating on multiple bits of binary data (typically closely-related data such as bits within a word) at a time.

In this part, you will implement a memory / register circuit on the Intel DE10-Lite board. The circuit has the following specifications:

- The current value of switches SW[9..0] on the DE10-Lite board should always be displayed in ***hexadecimal*** on the three seven-segment displays HEX2–HEX0. This part of the circuit will be combinational logic.

- Create a symbol for the binary to hexadecimal circuit. This combinational circuit should take a 4-bit binary input and output the seven-segment display driver signals so that the corresponding hexadecimal digit is displayed, as shown in Figure 4 below—note some patterns are non-standard. Use buses for the inputs and outputs as shown in Figure 5.

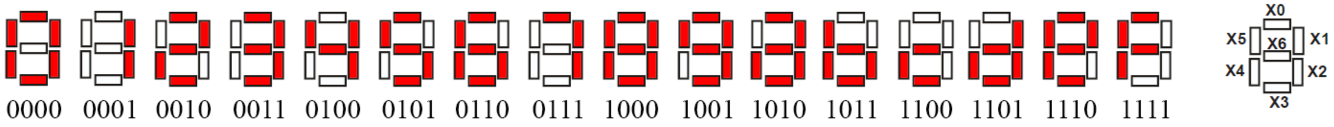

Figure 4. Hexadecimal Character Display

- Create a 12-bit positive edge triggered register using the embedded D flip-flops in the Altera FPGA. (Use the DFF component in the Altera library). Create a symbol for the register. (This can be implemented using 4-bit registers, if preferred). Again, use buses for the input and output to minimize wiring, as shown in Figure 5.

- Use KEY[0] as an active-low asynchronous reset and KEY[1] as the clock input of the 12-bit register.

- The contents of the 12-bit register should always be displayed in ***hexadecimal*** on the three seven-segment displays HEX5–HEX3.

- The SW[] input signals can be displayed on the red LEDs (LEDR[]), but this is not required. If the red LEDs are not used, they should be turned off.

Design Procedure

1. Create a new Quartus project, **lab4d**, which will be used to implement the desired circuit on the Intel DE10-Lite board.

2. Create the schematic and symbol for the combinational circuit that does the binary to hexadecimal segment display driver conversion. Create a schematic and symbol for the 4-bit or 12-bit register. Finally, create a top-level schematic that has the FPGA I/O pins and uses the register and combinational logic circuits. Import the pin assignments.

3. Compile the circuit and verify that the register is implemented using the embedded D flip-flops in the Altera FPGA. (Check the Resource Usage Summary or the Flow Control.)

4. Program the FPGA and test the functionality of the design by toggling the switches and observing the output displays. Right after the FPGA board is programmed, the values of SW[9..0] should be displayed in HEX[2..0], while HEX[5..3] are turned off or display 0s. Once the clock is triggered by pressing KEY[1] and release it, the values will be stored in the registers and displayed in HEX[5..3].

5. <u>Demonstrate your working circuit to your TA and have your TA sign your verification sheet</u>.

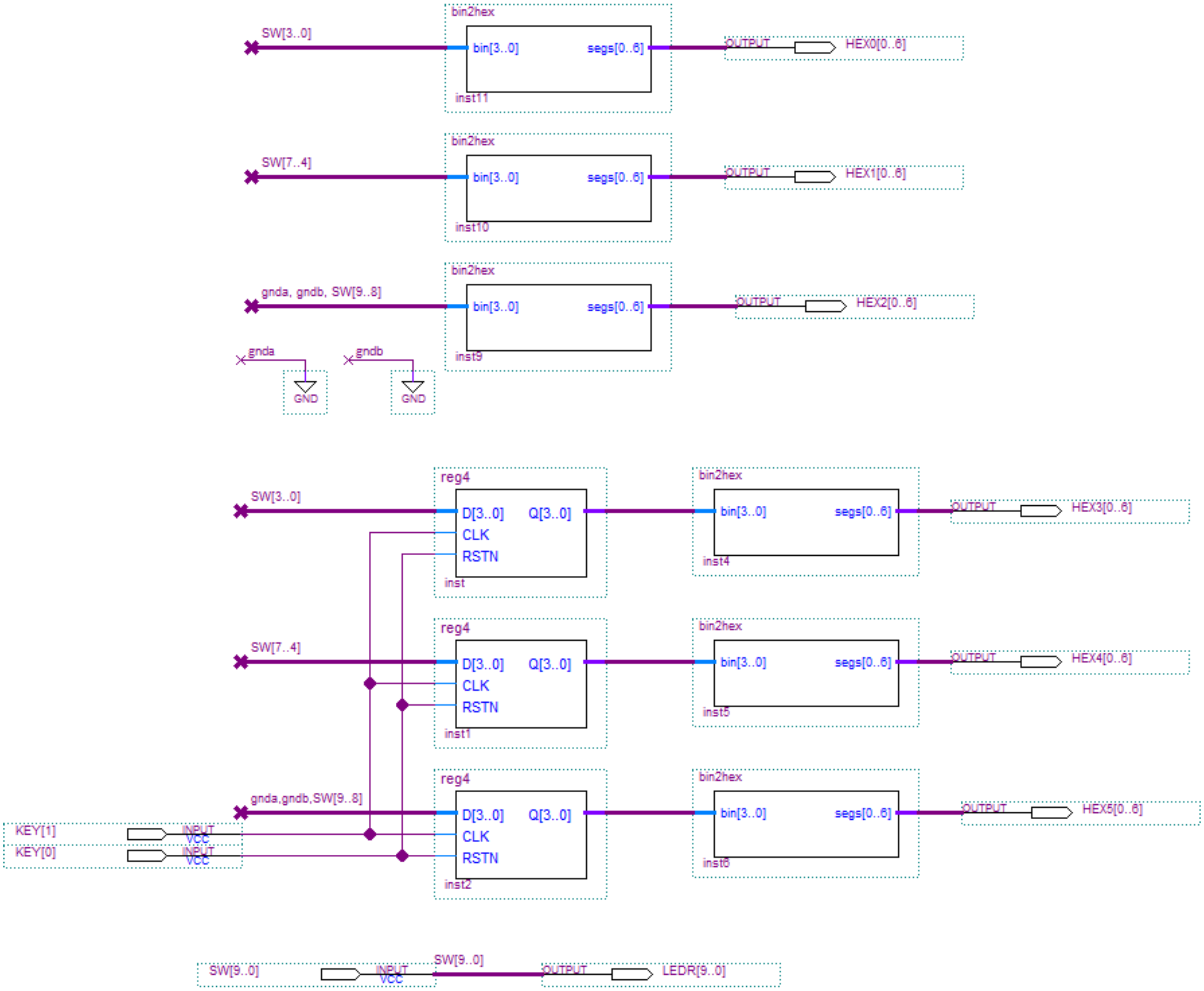

Figure 5. Top-level Schematic with Custom Symbols

V. Lab Report

Submit the following items:

1) Write your **Name**, **Section**, and **Lab number** on the first page of your report.
2) Complete Quartus schematics for all requested designs.
3) Answers to ***all*** questions asked throughout this lab.

2021/10/28 Posted

Acknowledgement: This lab is based on a Laboratory Exercise developed by Altera Corp. Used by permission.