CSC258H Lab 4: Sequential Logic

1 Introduction

This week, we'll examine the behavior of circuits which maintain internal state. These circuits are called latches and flip-flops. Designs which utilize these circuits are called "sequential", since data proceeds through them in a linear fashion with latches and flip-flops used to break the design into stages.

Before the lab, read through the lab handout so that you are familiar with the procedure. Make sure you understand the circuits you'll be implementing and have generated the requested truth tables and circuit diagrams. This lab is larger than last week, so if you encountered issues with time last week, you may wish to work ahead of the lab to implement some of your circuits. As you work, you'll find section 3.2 of the text to be particularly helpful.

2 SR Latches

Before lab, write down the truth table for a gated SR latch like the one below. How many columns of input does your table have? Clearly, it should columns for `S` and `R`. What about the `Clock`? How about the internal state (the initial values of `Q` and `NotQ`)? How many rows does the table have? How many circuit state and input combinations do you need to test to fully verify the circuit?

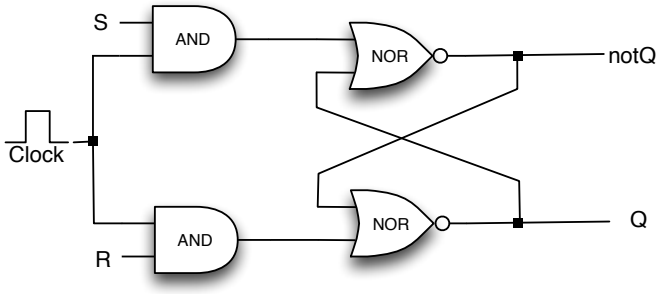

Figure 1: A gated SR latch

Use a schematic to implement the gated SR latch above. Include three input pins as input to the circuit instead of connecting the clock input to a clock component. During testing, we will want to control when clock pulses arrive, and using a third input for the clock input gives us the flexibility to connect that input to a device like a switch or push button.

Using your truth table, create a waveform file that fully verifies your circuit and then perform a functional simulation. Does the output of the simulation match your truth table? **Show your simulation results to the TA. Your simulation must demonstrate that a gated SR latch is NOT edge-triggered.**

3 D Flip-flop

The output of a latch changes whenever the clock input is high – during the positive "pulse". In contrast, the output of a flip-flop only changes when the clock input is changing. Typically, we build positive edge flip-flops – ones that change when the clock input changes from low to high. We could implement latches that change when the clock input is low (negative pulse), and similarly, a flip-flop could be implemented to change on the negative edge. The latter would be called a "negative-edge triggered flip-flop" or "falling-edge triggered flip-flop".

A D latch is very similar to an SR latch, except that, in a D latch, the S and R signals are always negations of each other (refer to the lecture slides for more details). A master-slave D flip-flop is a series of two D latches which have opposite clock signals (refer to the lecture slides for more details). **Before lab, draw the circuit diagram of a D latch using basic logic gates and an SR latch. Also draw the circuit diagram for a positive-edge triggered D flip-flop built from logic gates and D latches.**

In the lab, create a symbol for your SR latch from the last section and then use it to implement a D latch. Then, use those D-latches to implement the **positive-edge** triggered D flip-flop. Your design should take two inputs – `D` and `Clock`. It should produce a single output `Q`. Build a test vector to verify that your design works as expected. **Show your simulation result to the TA. Convince your TA that your D flip-flop is edge-triggered rather than pulse-triggered.**

4 FPGA Testing (Optional)

Optionally, after completing your D flip-flop, load your SR latch and D flip-flop onto the FPGA to verify them. For the SR latch, connect `S` and `R` to SW[0] and SW[1] (PIN_N25 and PIN_N26), and connect the clock to SW[2] (PIN_P25). Map the output to LEDR[0] (PIN_AE23). For the D flip-flop, you need one less switch. However, you may want to modify your schematic to send the internal state (the wire between the slave and master latches) to its own output. If you do so, you can map that output to LEDG[0] (PIN_AE22).

Do the latch and flip-flop operate as you expect? If you change the inputs `S`, `R`, or `D`, do you see the change immediately? Or do you need to change the clock?

5 Summary of TODOs

Below is a short summary of the steps to be completed for this lab:

1. Before the lab, read through the lab handout and prepare (a) the truth table for a gated SR latch, (b) the circuit diagram for a D latch built from an SR latch (and basic logic gates), and (c) the circuit diagram of a D flip-flop built from D latches (and basic logic gates). Bring these items to lab on paper for your TA to verify.

2. During lab, implement the gated SR latch and simulate it with test vectors that fully verify the truth table that you created. Show your TA the simulation results. You must provide an example that shows that the gated SR latch is NOT edge-triggered.

3. Save your gated SR latch as a symbol. Use your gated SR latch to build a D latch. Then, use your D latch to build a D flip-flop.

4. Create test vectors to simulate the D flip-flop. Show your simulation result to the TA. Convince your TA that you flip-flop is edge-triggered rather than pulse triggered.

Evaluation (3 marks in total): 1 mark for bringing the required truth tables and diagrams *and demonstrating your understanding of those items* to the TA; 1 mark for demonstrating a simulation result for an SR latch *and explaining why it demonstrates that the latch is not edge-triggered*; and 1 mark for demonstrating a simulation result for a D flip-flop *and comparing it to the SR latch to explain why it is an edge-triggered device*. I've added emphasis to these items: you must be able to explain your circuits and waveforms to the TA to earn credit.