Homework 2: Latches and Flip Flops

Due date: 07.04.2023 @23.59

Res. Asst. Büşranur Bülbül Demir
bulbulb17@itu.edu.tr

"Memory is the mother of all wisdom."
Aeschylus

1 Introduction

In this experiment, you will implement and examine data storage elements: latches and flip-flops using Verilog. In this experiment, you are allowed to use '&' (Bitwise AND), '|' (Bitwise OR), '∼' (Bitwise NOT), '{}' (Concatenate). Moreover, you can also use `always` block but only in the last part. That is, you are not allowed to implement latches and flip flops with this keyword. *You must simulate all parts separately with Vivado.* For this experiment, it may be useful to revise your previous knowledge from BLG231E-Digital Circuits course.

2 Preliminary

Answer the following questions in your report using your own words. Please be careful about not plagiarizing and properly cite the sources you benefit from.

1. Explain what a flip flop is and why they are useful. Use your own words.
2. Explain what are the differences between latches and flip flops with your own words.
3. Briefly explain how an SR-latch works and what the functionalities of the input variables are.
4. Construct the truth table of an SR latch which does not have an Enable input.
5. Construct the truth table of an SR latch which has an Enable input.
6. Construct the truth table of a D flip flop.
7. Construct the truth table of a JK flip flop.

3 Experiment

Part 1 - SR Latch (only NAND Gates)

Please implement an SR latch module with S and R inputs and with Q and Q_neg outputs ***without an Enable input***. You are allowed to use only 2-input NAND gates that you should implement as a separate module. Using the truth table you have constructed in the Preliminary section, write the characteristic equation of the latch as $Q(t+1) = f(S; R; Q(t))$. In your report, explain how you found the equation and how the latch behaves for disallowed inputs.

Part 2 - SR Latch with Enable input (only NAND Gates)

Please implement an SR latch module with S, R and Enable inputs and with Q and Q_neg outputs. You are allowed to use only 2-input NAND gates that you should implement as a separate module. Using the truth table you have constructed in the Preliminary section, write the characteristic function of the latch as $Q(t+1) = f(S; R; E; Q(t))$. In your report, explain how you found the equation and how the latch behaves for disallowed inputs. Discuss whether it is different from an SR Latch without an Enable input in your report.

Part 3 - D Flip-Flop from D-Latches

Please implement a negative edge triggered D flip-flop module with D input and for Q and Q_neg outputs using D latches with Enable input. You should implement the D latches with Enable input by yourselves using only 2-input NAND gates as a separate module. Show in your report that the clock is only effective at the falling edge.

Part 4 -JK Flip-Flop (only NAND gates)

Please implement an positive edge triggered JK flip-flop module from SR flip-flop with J, K and Clock inputs and for Q and Q_{neg} outputs. You should implement the JK flip flop by yourselves using only 2-input NAND gates as a separate module.

Part 5 - Asynchronous Up Counter (JK flip flop)

Please implement a 4-bit asynchronous up counter using JK flip flops. For each flip flop, input J, K and clock and get values from output Q. You should implement the counter count between 0-14 in decimal. Also reset the counter when it reaches 15 in decimal. Add the truth table of the counter you implemented to the report.

Part 6 - Synchronous Up Counter (JK flip flop)

Please implement a 4-bit synchronous up counter using JK flip flops. The counter must perform the same operation as the counter you created in Part 5. Add the truth table of the counter you implemented to the report.

Part 7

Please implement a positive edge triggered pulse generator using a circular shift register. The circuit should take 16-bit input for the loaded value, 1-bit input for the clock signal, 1-bit input for the load flag and give 1-bit output. Basically, when Load=0 (that is, shift=1), with the clock signal, circular shift operation is done; when Load=1, with the clock signal, a 16-bit input value is loaded.

Design your circuit in a way that the output of this circuit is the most significant bit (MSB) of the loaded value (so, firstly determine what the direction of the shift should be to maintain MSB as the output).

Your design should support variable pulse frequencies and duration listed below. Build the circuit and generate given signals. For each signal, observe both input and output. Explain your findings on the report.

- with the 1/2 frequency of clock signal
- with the 1/4 frequency of clock signal
- with the 1/8 frequency of clock signal
- with 1/7 pulse-gap duration rate
- with 3/13 pulse-gap duration rate
- with 11/5 pulse-gap duration rate

Hint 1: You can analyze the internal structure of 74XX165 IC for pulse-generator.

Hint 2: You can check what Pulse Width Modulation (PWM) is.

Hint 3: In order to provide different frequency ratios with your pulse generator, you should change the 16-bit input values in the simulation step. You should not change the design of the pulse generator for different frequency ratios.

4 Report

- You should show your work of Preliminary study on the report in detail.
- You can use any **software tool** for your circuit designs. You may attach them to the report as figures by properly referencing them in the text.
- Please use the table attributes of Latex. You can check out online Latex table generators (for example: https://www.tablesgenerator.com).
- Your report should contain information about the results of your simulations. If your implementations are not fully correct, discuss what the source of the errors might be in your report.
- If your implementations are not fully correct, discuss what the source of the errors might be in your report.

5 Submission

- Please do not send any document via e-mail to one of the assistants.
- Your reports must be written with Latex format. Latex report template is available on Ninova. You can use any Latex editor whichever you want. If you upload your report without Latex file, you directly get 0 as your report grade. You should upload both .pdf and .tex files of your report.
- You should submit 2 separate ".v" files for your Verilog codes. One of them should contain the modules and other one should contain the simulation codes.
- It will be sufficient for one person from each group to upload the homework.
- Be aware of the deadline. Late submissions are not accepted.
- Please do not hesitate to contact me (bulbulb17@itu.edu.tr) for any question.