CS 211, Summer 2017

Laboratory Exercise: Lab Exercise: Latches, Flip-flops and Simulation

Instructor: Professor Izidor Gertner

Due date: July 17, 2016, by 11:59PM

Objective:

This lab will allow you to understand the difference between latches and flip-flops and why this difference matters when building circuits. You will create several latches and flip-flops using block diagram design and run waveform simulations to understand and describe the behavior of these devices. The redesign all the sequential circuits using VHDL and repeat simulations.

Overview:

Latches and flip-flops are the basic elements for storing information.
One latch or flip-flop can store one bit of information.

Latches – These are sequential devices that watch their input continuously and can change their output at ANY time (some cases require clock signal or enable signal to be asserted (set to High)).

Latches are said to be LEVEL triggered. In level triggering the circuit will become active when the gating or clock pulse is on a particular level, high or low.

Flip-flops – Sequential device that samples its input and changes its output ONLY when the clock signal is changing (at the rising or falling edge of the clock). In essence, flip-flops trigger their output based on an event (clock event).

Flip-flops are built from latches and are said to be EDGE triggered. In edge triggering the circuit becomes active at negative or positive edge of the clock signal. For example if the circuit is positive edge triggered, it will take input at exactly the time in which the clock signal goes from low to high. Similarly input is taken at exactly the time in which the clock signal goes from high to low in negative edge triggering.

What do we mean by "sequential" device?
For a sequential device, its output depends not only on its current input but on the past sequence of inputs, possibly arbitrarily back in time. Therefore, we can conclude that such devices have some sort of memory feature because they must remember their previous value generation in order to compute the next output.

We will now review: SR-Latch, SR-Latch with Control, D-Latch, D Flip-flop, SR Flip-flop, JK Flip-flop and T Flip-flop. Your assignment is listed at the end of this review.

CS 211, Summer 2017

Laboratory Exercise: Lab Exercise: Latches, Flip-flops and Simulation

Instructor: Professor Izidor Gertner

Due date: July 17, 2016, by 11:59PM

SR-LATCH

Design:

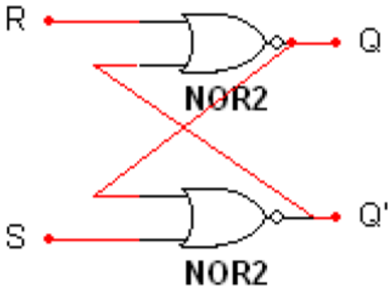
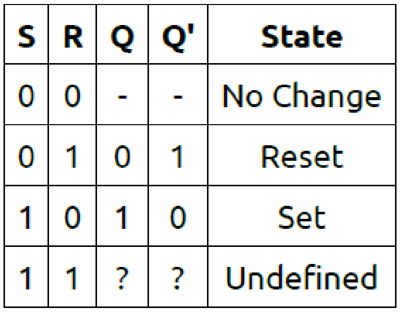


Figure 1. SR-Latch circuit using NOR gates

Functionality:

The Set/Reset Latch (SR-Latch) is a basic storage device used to store a binary bit. This latch has two functions, SET and RESET, which are controlled by the configuration of the input signals, S and R, respectively. These functions are demonstrated by the table in Figure 2. This table shows that when input S is 1 and input R is 0, the latch is in the “Set” state and its output, Q, will always be one. Likewise, when the input S is 0 and input R is 1, the latch will be in the “Reset” state and its output, Q, will always be zero. Furthermore, when both inputs are 0, the latch will hold the state of the previous output and will show no change in its output. However, when both inputs are 1, the latch will be in the undefined state; therefore, it will be impossible to predict the next state. In fact, one of the requirements of the SR latch is that the outputs Q and Q’ always be compliment of each other.



S	R	Q	Q'	State
0	0	-	-	No Change
0	1	0	1	Reset
1	0	1	0	Set
1	1	?	?	Undefined

Figure 2. Truth table for SR-latch

CS 211, Summer 2017

Laboratory Exercise: Lab Exercise: Latches, Flip-flops and Simulation

Instructor: Professor Izidor Gertner

Due date: July 17, 2016, by 11:59PM

CONTROL SR-LATCH

Design:

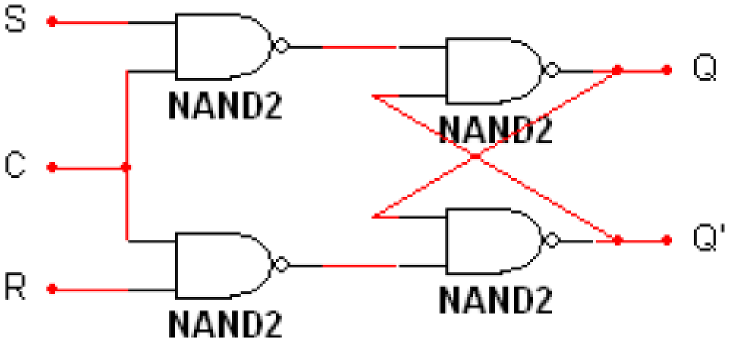
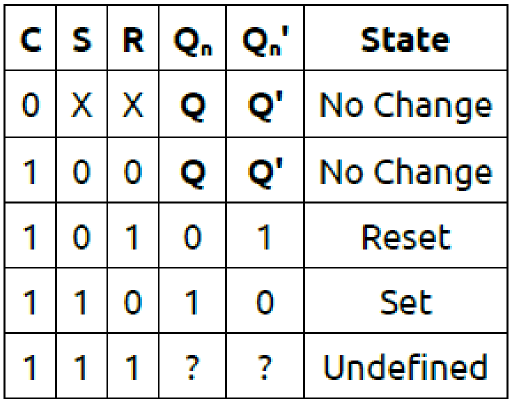


Figure 3. Control SR-Latch using NAND gates

Functionality:

This latch is similar to the SR-Latch in that it stores a binary bit, but uses an extra control but to determine state changes. For example, when the control signal, C, is 0, then the latch exhibits no changes and retains the previous state/bit. However, when the control signal is 1, the state of the latch is determined in the same manner as the regular SR-Latch using only inputs, S and R. This behavior is shown below in Figure 4, where Qn, denotes the next state of Q (i.e. Q next).



C	S	R	Q_n	Q_n'	State
0	X	X	**Q**	**Q'**	No Change
1	0	0	**Q**	**Q'**	No Change
1	0	1	0	1	Reset
1	1	0	1	0	Set
1	1	1	?	?	Undefined

Figure 4. Truth table for Control SR-Latch

CS 211, Summer 2017

Laboratory Exercise: Lab Exercise: Latches, Flip-flops and Simulation

Instructor: Professor Izidor Gertner

Due date: July 17, 2016, by 11:59PM

D-LATCH

Design:

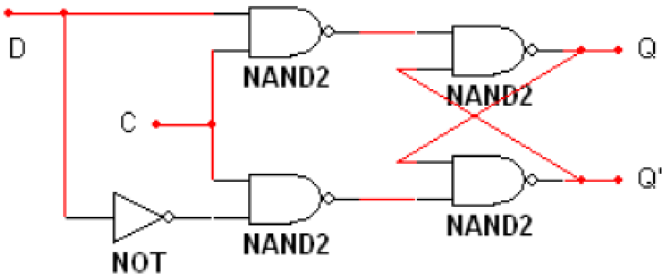
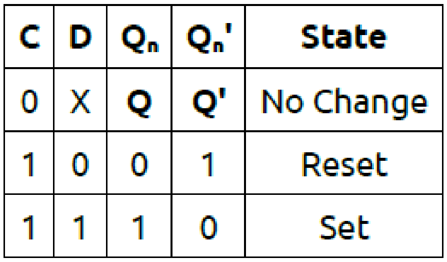


Figure 5. D-Latch using NAND gates

Functionality:

We obtain a D-Latch when we tie a NOT gate to the bottom NAND gate and only input D is used instead of independent Set and Reset. This will assure that S and R are always compliment or opposite of each other, thus eliminating falling into the undefined state when S=1, R=1.

This latch also stores a binary bit, but uses only one input signal and a clock signal. This latch is similar to the Control SR-Latch described above in that the clock signal is use in the same manner as the control signal. This relationship can be seen from the truth table of Figure 8 below. This table demonstrates that the D-Latch is a simplified version of an SR-Latch in that it has three states, Set, Reset, and No Change, but requires only one input signal (D) and a clock rate ©.



C	D	Q_n	Q_n'	State
0	X	Q	Q'	No Change
1	0	0	1	Reset
1	1	1	0	Set

Figure 6. Truth table for D-Latch

Laboratory Exercise: Lab Exercise: Latches, Flip-flops and Simulation

Instructor: Professor Izidor Gertner

Due date: July 17, 2016, by 11:59PM

MASTER-SLAVE D FLIP-FLOP

Design:

Two D-Latches are connected in series. Notice that the output, Q, of the first latch is tied to the input of the second latch, D, and both latches use the same clock signal, but the second latch inverts the clock signal.

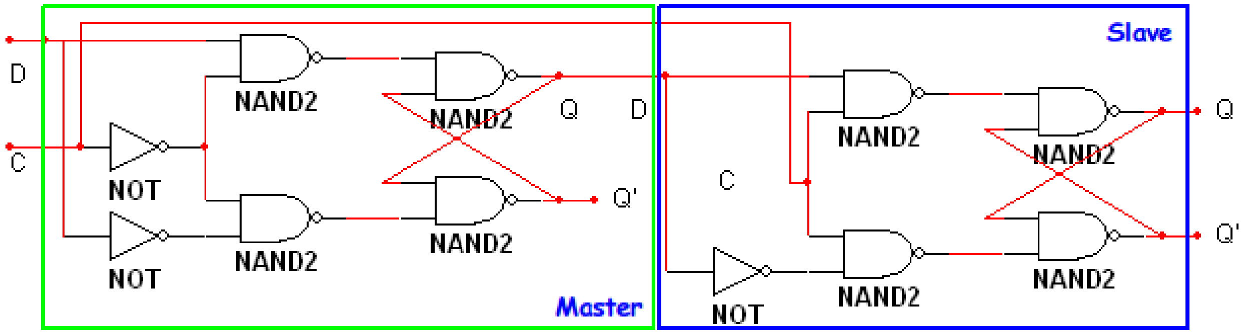


Figure 7. Master-Slave D Flip-flop

An abstracted representation of the master-slave D flip-flop from Figure7 above is shown below:

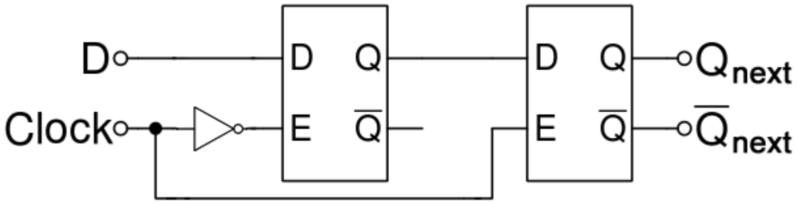


Figure 8a. Abstracted view of two D Latches connected to make positive edge-triggered Master-Slave D Flip-flop

It is important to note that this is a positive edge-triggered circuit, as opposed to the pulse-triggered design of the latches. This means that the output Q will equal the input D only on a clock edge (in this case, a raising edge). Therefore, Q can only change at the moment the clock goes from 0 to 1.
A negative edge-triggered D flip-flop can be obtaining by taking the NOT gate out from the front of master D latch and putting it in front of slave D latch, like this:

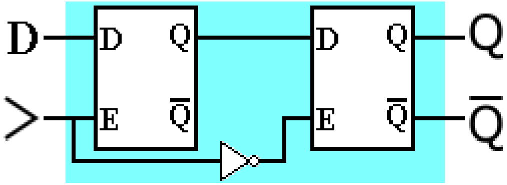


Figure 8b. Abstracted view of two D Latches connected to make negative edge-triggered Master-Slave D Flip-flop

Functionality:

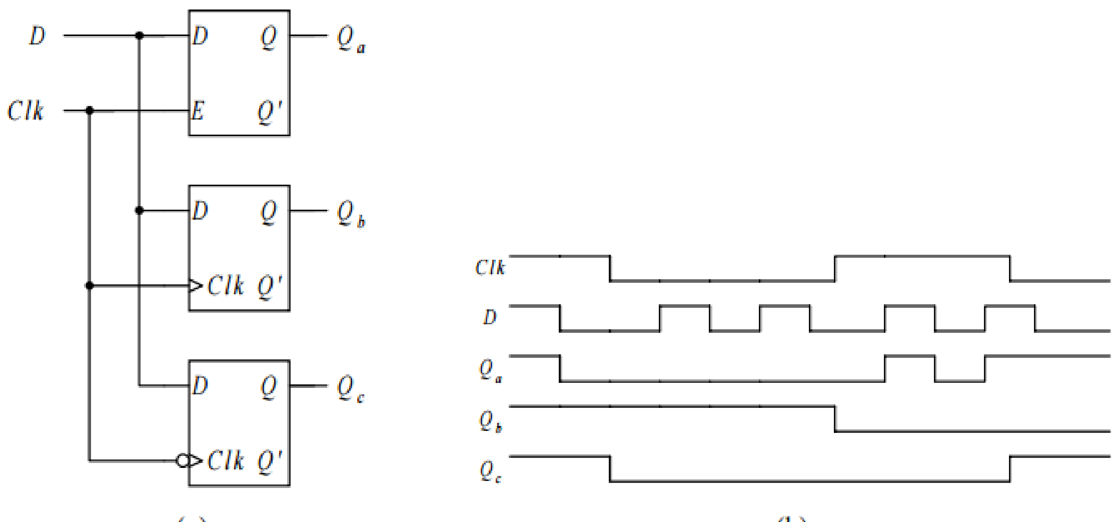
This flip-flop stores a binary bit, but uses two D-Latches in a master-slave configuration. It is different from the regular D-Latch because its state changes do not depend on the state of the clock signal, but the transition of the clock signal from zero to one. This can be seen from the truth table of Figure 9, where the left column represents the clock signal and D is input for the master D-Latch. When the clock transitions from 0 to 1 (goes from low to high), the previous input is used to generate the global output for the system. The reason for this is that at clock state 0, the slave latch exhibits no change (retaining the global output, Q0), while the master latch generates a new input for the slave latch, q1. Then, when the clock state changes to 1, the master latch exhibits no change (retaining the previous output, q1), while the slave latch generates a new output based on the previous output (output of the master latch), Q1. Likewise, this pattern continues throughout the table.

The truth table for a negative edge-triggered flip-flop is not shown but you should by now be able to infer its behavior.

Clk	D	Q	Q_{next}	Q_{next}'
0	×	0	0	1
0	×	1	1	0
1	×	0	0	1
1	×	1	1	0
↑	0	×	0	1
↑	1	×	1	0

Figure 9. Truth table for positive edge-triggered master-slave D flip-flop

Figure 10 compares the behavior between a D Latch, a positive edge-triggered D flip-flop and a negative edge-triggered D flip-flop. They all share the same input D and clock signal Clk. Figure 10(a) shows the symbol diagrams. figure 10 (b) shows a sample trace of the circuit's operations. Notice that the gated D latch Qa follows the D input as long as the clock is high. The positive-edge-triggered flip-flop Qb responds to the D input only at the rising edge of the clock while the negative-edge-triggered flip-flop Qc responds to the D input only at the falling edge of the clock.



(a) (b)

Figure 10. Comparison of a gated latch, positive edge-triggered flip-flop and a negative edge-triggered flip-flop.(a) shows the circuit diagrams. (b) shows the timing diagram.

MASTER-SLAVE SR FLIP-FLOP

Design:

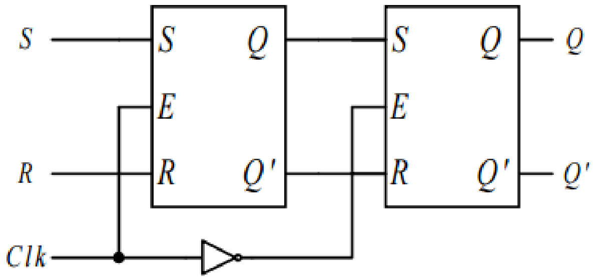


Figure 11. SR flip-flop circuit diagram

Functionality:

We can replace the D latches in the D flip-flop of Figure 8 with SR latches to get a master-slave SR flip-flop shown in Figure 11. Like SR latches, SR flip-flops are useful in control applications where we want to be able to set or reset the data bit. However, unlike SR latches, SR flip-flops change their content only at the active edge of the clock signal. Similar to SR latches, SR flip-flops can enter an undefined state when both inputs are asserted simultaneously.

The truth table for SR flip-flop is shown below in Figure 12. Notice how the values for Qnext and Qnext' are undetermined when both S and R are asserted (set to high) regardless of the value of Q, which is the value set on Q in a previous clock event and waiting to pass it on to the slave latch.

We will not be too concerned about SR flip-flops in this assignment but we add it for consistency and because it helps us understand and justify the behavior of a very useful circuit: JK flip-flop.

S	R	Q	Q_{next}	Q_{next}'
0	0	0	0	1
0	0	1	1	0
0	1	0	0	1
0	1	1	0	1
1	0	0	1	0
1	0	1	1	0
1	1	0	×	×
1	1	1	×	×

Figure 12. Truth table for SR flip-flop

CS 211, Summer 2017

Laboratory Exercise: Lab Exercise: Latches, Flip-flops and Simulation

Instructor: Professor Izidor Gertner

Due date: July 17, 2016, by 11:59PM

JK FLIP-FLOP

Design:

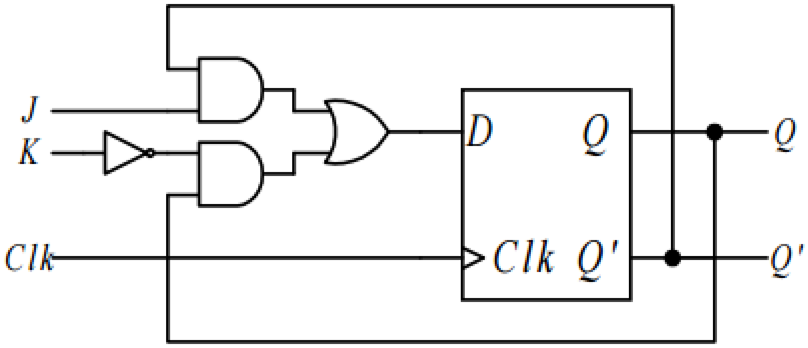


Figure 13. JK flip-flop circuit diagram

Functionality:
JK flip-flops are very similar to SR flip-flops. The J input is just like the S input in that when asserted, it sets the flip-flop. Similarly, the K input is like the R input where it clears the flip-flop when asserted. The only difference is when both inputs are asserted. For the SR flip-flop, the next state is undefined, whereas, for the JK flip-flop, the next state is the inverse of the current state. In other words, the JK flip-flop toggles its state when both inputs are asserted. The circuit for the JK flip-flop is shown in Figure 13; its truth table is shown in Figure 14 below:

J	K	Q	Q_{next}	Q_{next}'
0	0	0	0	1
0	0	1	1	0
0	1	0	0	1
0	1	1	0	1
1	0	0	1	0
1	0	1	1	0
1	1	0	1	0
1	1	1	0	1

Figure 14. Truth table for JK flip-flop – Notice that unlike SR flip-flop, the circuit does not enter an undetermined state when both inputs J and K are asserted.

The truth table for the JK flip-flop can be shortened as shown in Figure 15. This is called the characteristic table. The truth table lists all possible combinations of input signals, current state, intermediate value (in this case Q), and next state (plus its inverse). The characteristic table answers the question of what is the next state when given the inputs and the current state, and is used in the analysis of sequential circuits.

J	K	Q_{next}
0	0	Q
0	1	0
1	0	1
1	1	Q'

Figure 15. Characteristic table for JK flip-flop

T FLIP-FLOP

Design:

The T flip-flop can be constructed using a D flip-flop with the two outputs Q and Q' feedback to the D input through a multiplexer that is controlled by the T input as shown in Figure 16.

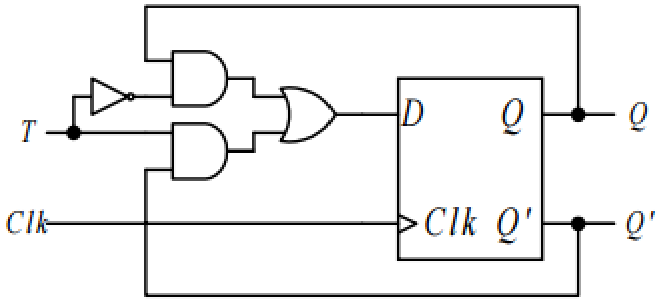
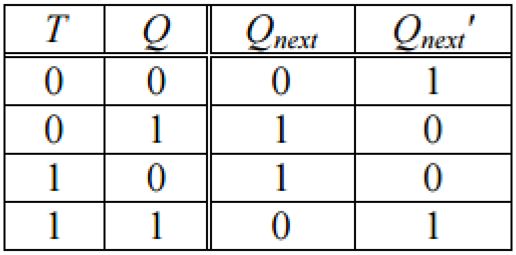


Figure 16. T flip-flop circuit diagram

Functionality:
T flip-flop's behavior is probably the easiest to understand and remember. T stands for toggle, meaning that if T is asserted (T=1), then the flip-flop state toggles or changes its value from what it was before. When T is de-asserted (T=0), the flip-flop keeps its current state.



T	Q	Q_{next}	Q_{next}'
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

Figure 14. Truth table for T flip-flop – Notice that when T=0, Qnext keeps the value of previous Q. When T=1, the value of Qnext is opposite of value of its previous Q.

YOUR TASK

List of latches and flip-flops you have to design:

Using block diagrams in Quartus you will design and explain the behavior of the following circuits:

1. SR latch
2. Control SR latch
3. D latch
4. Positive edge-triggered Master-slave D latch
5. Negative edge-triggered Master-slave D latch
6. JK flip-flop
7. T flip-flop

NOTE: If you need help with the basics of block diagram design and simulation you can review a very basic Quartus II video tutorial at http://tinyurl.com/mztraqz

Step 1: Build

- Open Quartus and create a new project.
- Create Block Diagram Files for the latches and flip-flops listed above (each latch/flip-flop in its own individual file). For your diagram designs, you can use the corresponding figures shown in the previous pages for your reference.

Step 2: Convert to Symbol

- Convert your D latch and D flip-flop (positive and negative) circuits to symbols.
  NOTE: If you need help, you can check out the following video tutorial on symbol creation in Quartus: http://tinyurl.com/m8rrkyd
- Give the symbols meaningful short names. For example, you can name your positive edge-triggered master-slave D flip flop as PosDFF.
- Put the symbols you just created in one master block diagram file. The idea is to have them lined up just like we had a D-latch and two flip-flops in Figure 10a shown earlier in this assignment. Give them a shared clock input and a shared D input.

Step 3: Waveform simulation

- Open a new Vector Waveform File for each circuit built and add the inputs and outputs.
- Give values to the waveform such that all possible input combinations are simulated.
- Run the simulation and observe the results.
- Compare results to the truth tables given previously for each latch/flip-flop.
- For the file that contains the 3 symbols created (D-latch, positive D flipflop, negative D flipflop), run a waveform simulation with the following setting for Clock and D:

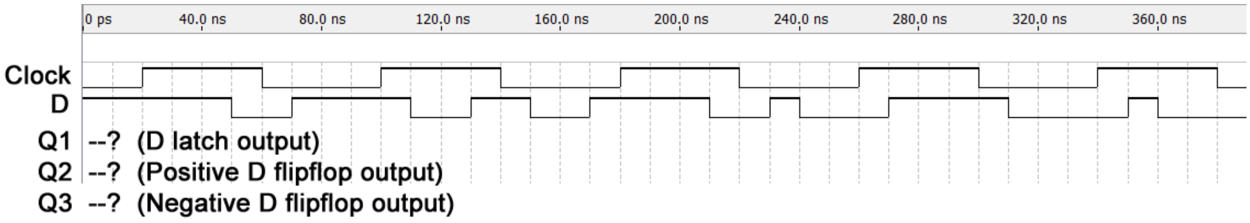


Figure 15. Waveform setting for D latch and D flip flops comparison. Can you predict the outputs?

Step 3: Report

Please write a report stating the following:

CS 211, Summer 2017

Laboratory Exercise: Lab Exercise: Latches, Flip-flops and Simulation

Instructor: Professor Izidor Gertner

Due date: July 17, 2016, by 11:59PM

Objective

- What is the goal of this assignment?

Design

- Include screenshots of your block diagram design in Quartus software of each circuit.
- Describe the inputs and outputs of each latch and flip flop you built.

Functionality

- Explain the behavior of a SR Latch. What do we mean by undetermined state (metastability)?
- Explain how the gated (control) SR-Latch is a modified SR-Latch.
- Explain how the D-Latch is a modified gated SR-Latch.
- Explain how to build a Master-Slave D flip flop.
- Explain how a JK flip-flop works.
- Explain how a T flip-flop works.

Simulation

- Include a screenshot of the vector waveform from step 3 after producing the output for EACH of the latches/flip-flops built.
- Include a screenshot of the completed vector waveform from the settings given in figure 15.

Analysis

- What is the difference between latches and flip-flops? Why does it matter?
- Why do we want to avoid the scenario when S=1 and R=1 in an SR-Latch? Could this scenario happen in a D-Latch? Why or why not?
- What is the difference between edge-triggered and level triggered devices? Is the T flip-flop you built in this lab level or edge triggered, for instance?
- Are flip-flops always required to be clocked? Why or why not?
- Compare the behavior of the D latch and D flip-flop (positive and negative) outputs in figure 15.

CS 211, Summer 2017

Laboratory Exercise: Lab Exercise: Latches, Flip-flops and Simulation

Instructor: Professor Izidor Gertner

Due date: July 17, 2016, by 11:59PM

Conclusion

- Give a brief conclusion on what you learned about this assignment. Can you give examples of what can be built with latches and flip-flops?