City College of New York

Take Home Test # 2

Zi Xuan Li

CSC 211 Fall 2022

Professor Izidor Gertner

November 13th, 2022

Objective:

The objective of this lab report is to build, simulate, and verify the correctness of both the Rising Edge D-Flipflops and Falling Edge D-Flipflops. This report will compose of Rising and Falling Edge D-Flipflops built using AND, OR, and NOT gates built in a template of 2:1 Multiplexers, as well as Rising and Falling Edge D-Flipflops built using 2:1 Multiplexers as symbols, and lastly Rising and Falling Edge D-Flipflops built using the DFF found in the library of Quartus components. Alongside the D-Flipflops, the report will also demonstrate VHDL code for the Rising and Falling D-Flipflop as well as simulation of the code using the Quartus "Create Symbol file from current VHDL File" function. Finally, the report will conclude with a simulation comparison of a level sensitive D-Latch, Rising Edge D-Flipflop, and a Falling Edge D-Flipflop.

Functionality, Specifications, and Simulation:

To verify the correctness of these digital circuits, we must first understand what a D-FF is and how a Rising Edge and Falling Edge D-FF works.

The D-FF is a flip flop that contains two inputs. Those inputs are the data (D) input and the clock (CLK) input. The D-FF is used to delay the change of state of its output signal (Q) until the next rising edge of the clock signal occurs. This means that the output of the flipflop changes with the transition of the clock pulse, either from low to high or high to low in the case of the falling edge D-FF.

The Rising Edge D-FF is a circuit that changes its output (Q) in accordance with the input (D) during the positive transition of the clock (CLK) pulse of the flipflop. This implies that the D-FF is a flip-flop in which the output can only change with the edge of the clock pulse, regardless of the change in the input. In other words, unless the Rising Edge D-FF's clock input is changing from 0 to 1, the flipflop stays in a hold state.

The Falling Edge D-FF is a circuit that changes its output (Q) in accordance with the input (D) during the negative transition of the clock (CLK) pulse of the flipflop. This implies that the D-FF is a flip-flop in which the output can only change with the edge of the clock pulse, regardless of the change in the input. But unlike the Rising Edge D-FF, unless the Falling Edge D-FF's clock input is changing from 1 to 0, the flipflop stays in a hold state.

Rising Edge D-Flipflop		
CLK	D	Q(t)
0	0	Q (t – 1)
0	1	Q (t – 1)
	0	0
	1	1

Falling Edge D-Flipflop		
CLK	D	Q(t)
	0	0
	1	1
1	0	Q (t – 1)
1	1	Q (t – 1)

Part 1: (Design and build in Quartus Rising Edge D-FF using 2:1 MUX using gates from the first lab)

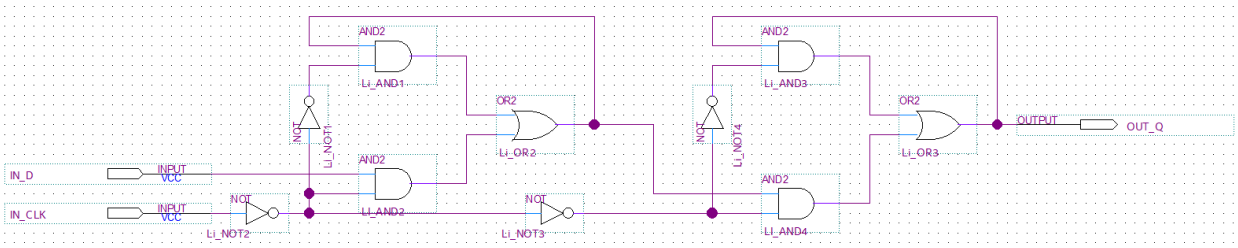

Figure 1: Rising Edge D-FF built using the 2:1 Mux as a template with AND, OR, and NOT gates.

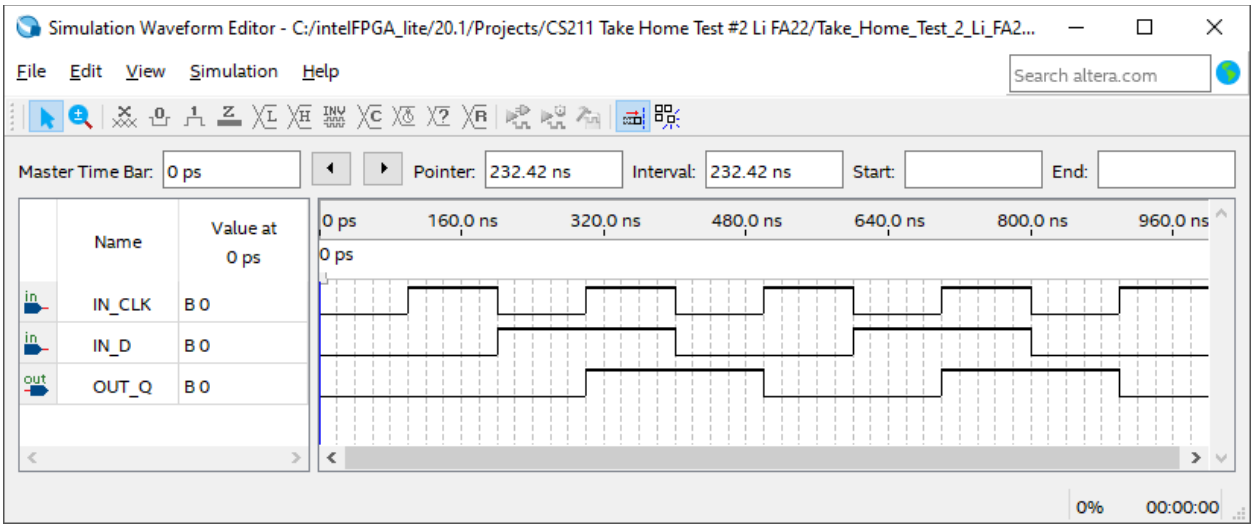

Figure 2: University VWF output for the Rising Edge D-FF built in figure 1 with input signals IN_CLK, IN_D and output signal OUT_Q. The duty cycles of the signals remain at 50%, however the period of IN_CLK is 200ns and IN_D is 400ns.

Rising Edge D-Flipflop		
CLK	D	Q(t)
0	0	Q (t – 1)
0	1	Q (t – 1)
↑	0	0
↑	1	1

To verify the correctness of the Rising Edge D-FF built using 2:1 Mux as a template with AND, OR, and NOT gates. We can compare the instances where the input IN_CLK is transitioning from (0 to 1) i.e., when a rising edge occurs in the CLK signal to the Rising Edge D-FF truth table. In this case, the first rising edge occurs between 0 ns and 160ns where IN_D is low, we can see that OUT_Q is also low. The second rising edge occurs between 160ns and 320ns where IN_D is high, and we can see that OUT_Q is also high. Looking at the timestamps between those instances, we can see that OUT_Q is holding onto the previous value of Q unless IN_D changes during the CLK's rising edge. With that information, we can map the rest of the inputs and outputs and conclude that the waveform simulation is correct.

Part 2: (Design and build in Quartus Falling Edge D-FF using 2:1 MUX using gates from the first lab)

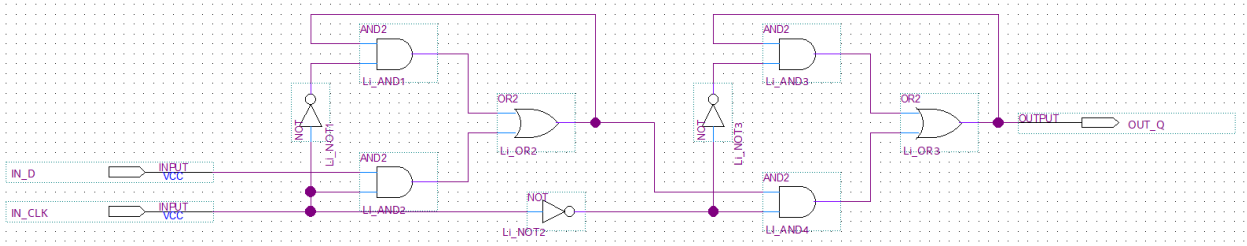

Figure 3: Falling Edge D-FF built using the 2:1 Mux as a template with AND, OR, and NOT gates.

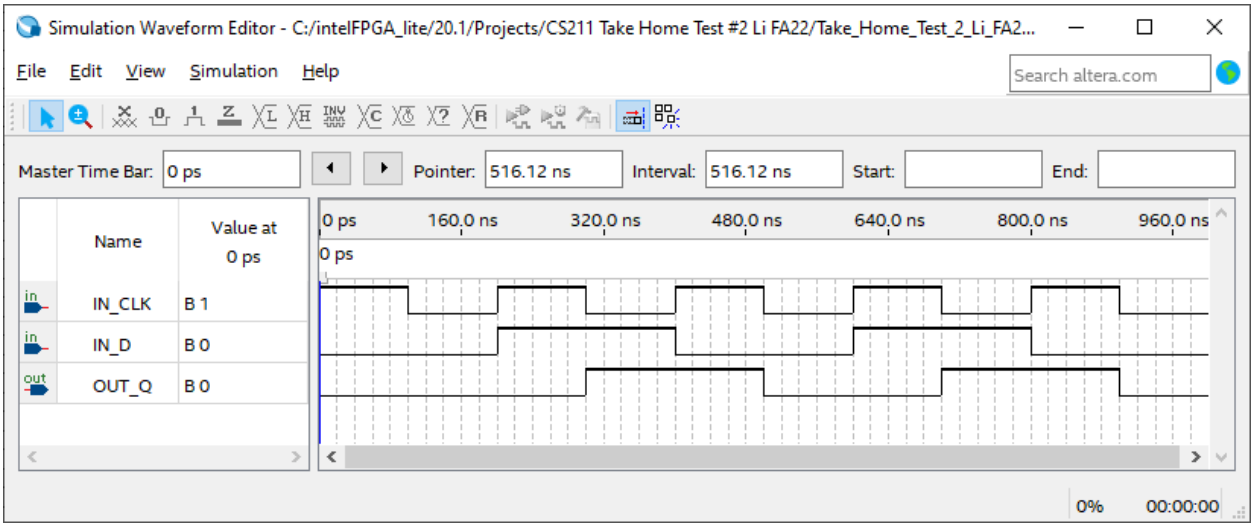

Figure 4: University VWF output for the Falling Edge D-FF built in figure 3 with input signals IN_CLK, IN_D and output signal OUT_Q. The duty cycle remains at 50%, however the period of IN_CLK is 200ns and inverted and IN_D is 400ns.

Falling Edge D-Flipflop		
CLK	D	Q(t)
↓	0	0
↓	1	1
1	0	Q (t – 1)
1	1	Q (t – 1)

To verify the correctness of the Falling Edge D-FF built using 2:1 Mux as a template with AND, OR, and NOT gates. We can compare the instances where the input IN_CLK is transitioning from (1 to 0) i.e., when a falling edge occurs in the CLK signal to the Falling Edge D-FF truth table. In this case, the first falling edge occurs between 0 ns and 160ns where IN_D is low, we can see that OUT_Q is also low. The second falling edge occurs between 160ns and 320ns where IN_D is high, and we can see that OUT_Q is also high. Looking at the timestamps between those instances, we can see that OUT_Q is holding onto the previous value of Q unless IN_D changes during the CLK's falling edge. With that information, we can map the rest of the inputs and outputs and conclude that the waveform simulation is correct.

Part 3: (Design and build in Quartus Rising Edge D-FF using 2:1 MUX using Quartus component symbol)

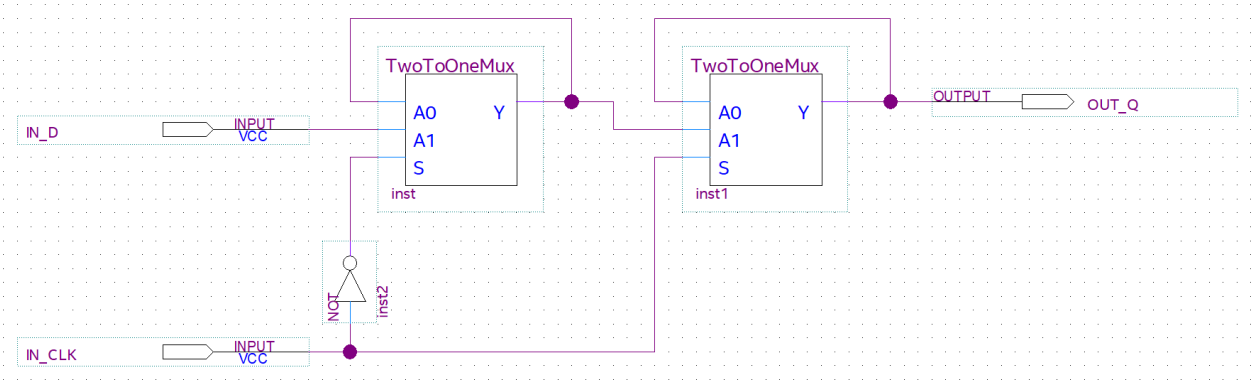

Figure 5: Rising Edge D-FF built using the 2:1 Mux symbol as a component.

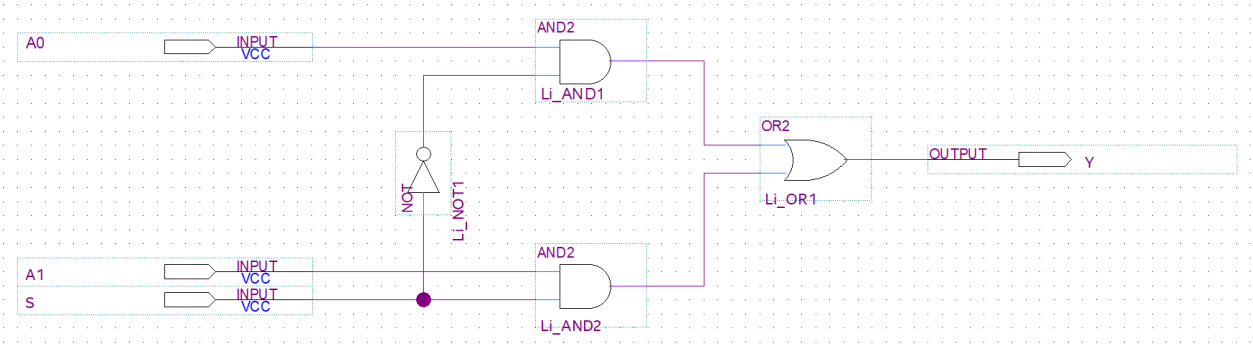

Figure 6: Circuit diagram of the 2:1 Mux symbol used in the Rising Edge D-FF in figure 5.

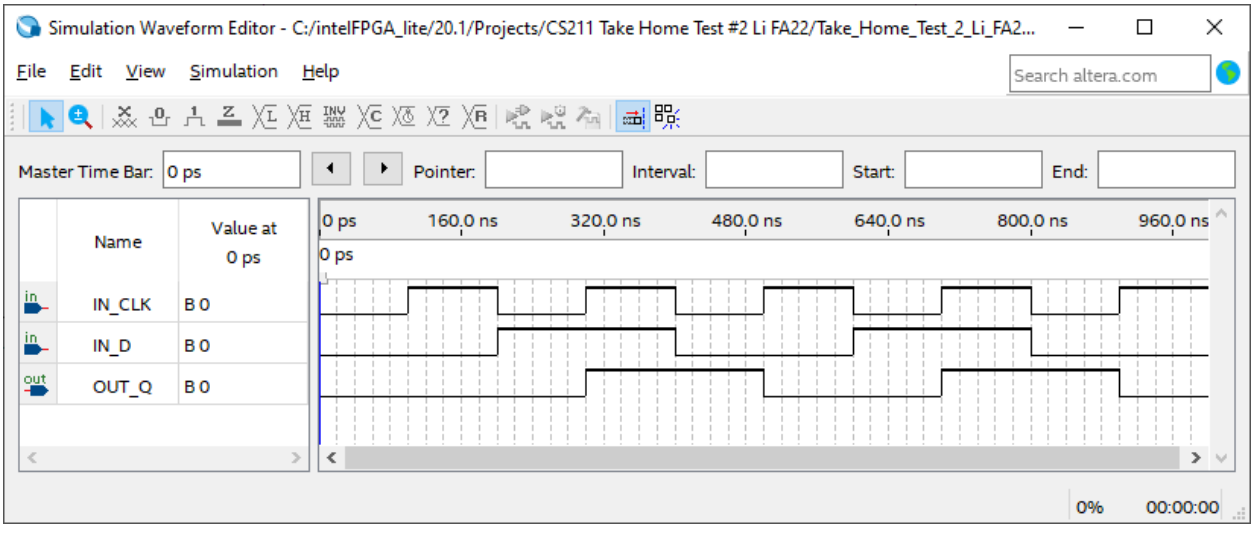

Figure 7: University VWF output for the Rising Edge D-FF built in figure 5 with input signals IN_CLK, IN_D and output signal OUT_Q. The duty cycles of the signals remain at 50%, however the period of IN_CLK is 200ns and IN_D is 400ns.

Rising Edge D-Flipflop		
CLK	D	Q(t)
0	0	Q (t – 1)
0	1	Q (t – 1)
↓	0	0
↓	1	1

To verify the correctness of the Rising Edge D-FF built using 2:1 Mux symbol as a component. We can compare the instances where the input IN_CLK is transitioning from (0 to 1) i.e., when a rising edge occurs in the CLK signal to the Rising Edge D-FF truth table. In this case, the first rising edge occurs between 0 ns and 160ns where IN_D is low, we can see that OUT_Q is also low. The second rising edge occurs between 160ns and 320ns where IN_D is high, and we can see that OUT_Q is also high. Looking at the timestamps between those instances, we can see that OUT_Q is holding onto the previous value of Q unless IN_D changes during the CLK's rising edge. With that information, we can map the rest of the inputs and outputs and conclude that the waveform simulation is correct.

Part 4: (Design and build in Quartus Falling Edge D-FF using 2:1 MUX using Quartus component symbol)

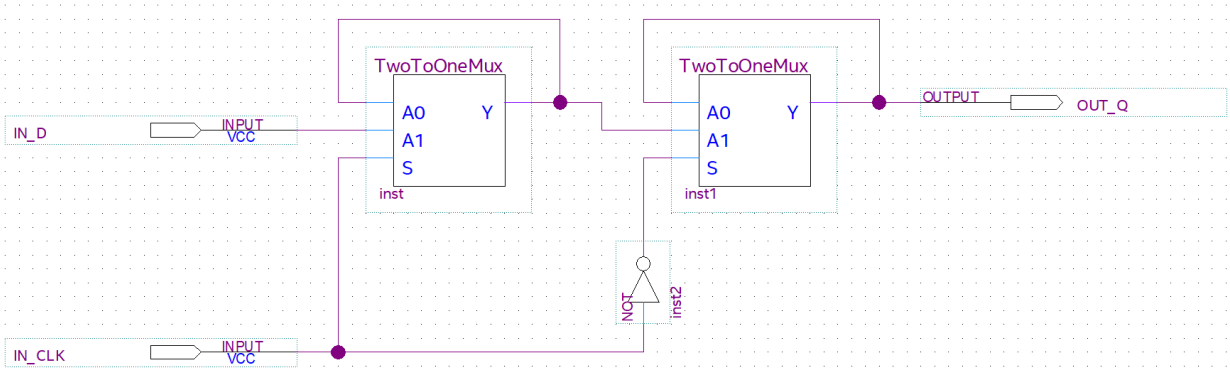

Figure 8: Falling Edge D-FF built using the 2:1 Mux symbol as a component.

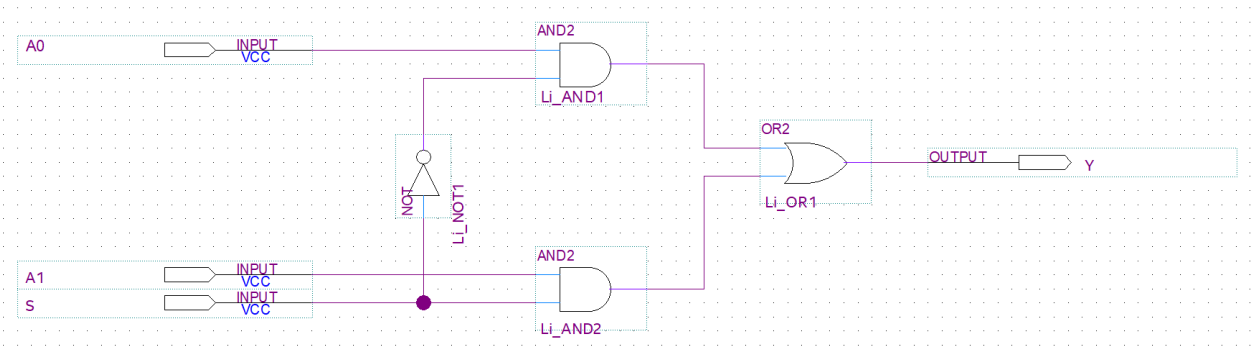

Figure 9: Circuit diagram of the 2:1 Mux symbol used in the Falling Edge D-FF in figure 8.

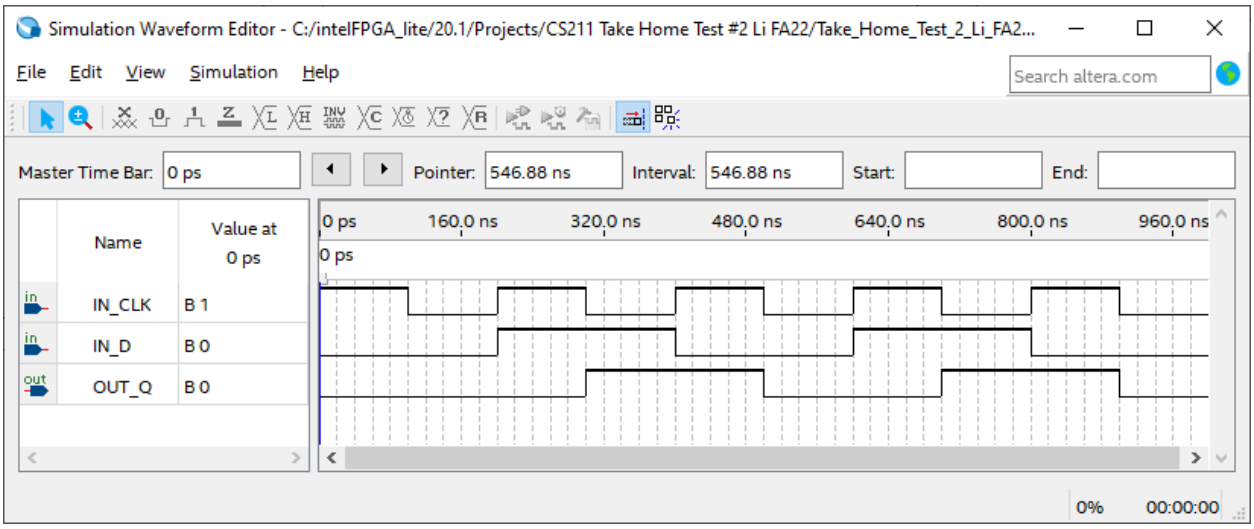

Figure 10: University VWF output for the Falling Edge D-FF built in figure 8 with input signals IN_CLK, IN_D and output signal OUT_Q. The duty cycle remains at 50%, however the period of IN_CLK is 200ns and inverted and IN_D is 400ns.

Falling Edge D-Flipflop		
CLK	D	Q(t)
↓	0	0
↓	1	1
1	0	Q (t – 1)
1	1	Q (t – 1)

To verify the correctness of the Falling Edge D-FF FF built using 2:1 Mux symbol as a component. We can compare the instances where the input IN_CLK is transitioning from (1 to 0) i.e., when a falling edge occurs in the CLK signal to the Falling Edge D-FF truth table. In this case, the first falling edge occurs between 0 ns and 160ns where IN_D is low, we can see that OUT_Q is also low. The second falling edge occurs between 160ns and 320ns where IN_D is high, and we can see that OUT_Q is also high. Looking at the timestamps between those instances, we can see that OUT_Q is holding onto the previous value of Q unless IN_D changes during the CLK's falling edge. With that information, we can map the rest of the inputs and outputs and conclude that the waveform simulation is correct.

Part 5: (Write YOUR OWN VHDL code for Rising Edge D-FF, Falling Edge D-FF. Create a separate Quartus project for each)

```
1  Library IEEE;
2  USE IEEE.Std_logic_1164.all;
3
4  entity RisingEdgeDFlipFlop is
5  port
6     (
7        Q : out std_logic;
8        Clk :in std_logic;
9        D :in  std_logic
10    );
11
12 end RisingEdgeDFlipFlop;
13
14 architecture Behavioral of RisingEdgeDFlipFlop is
15 begin
16  process(clk)
17  begin
18
19     if(rising_edge(clk)) then
20     Q <= D;
21     end if;
22  end process;
23  end Behavioral;
```

Figure 11: VHDL code of a Rising Edge D-FF.

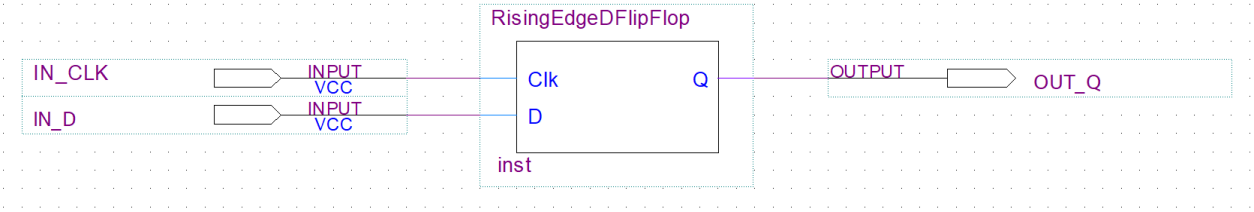

Figure 12: Circuit Diagram created using the symbol generated from the VHDL code in figure 11.

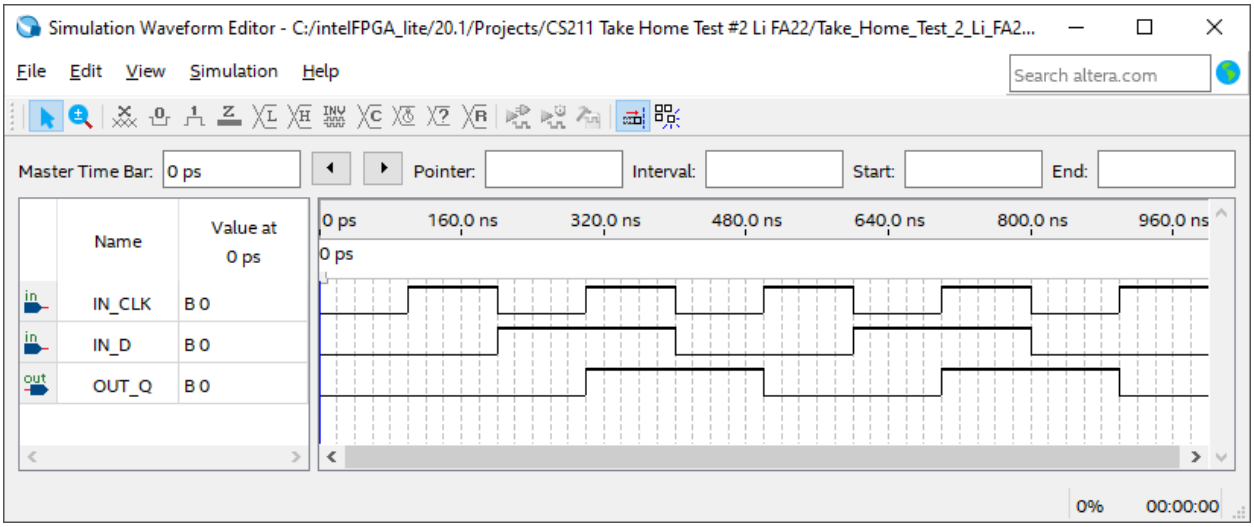

Figure 13: University VWF output for the Rising Edge D-FF built in figure 12 with input signals IN_CLK, IN_D and output signal OUT_Q. The duty cycles of the signals remain at 50%, however the period of IN_CLK is 200ns and IN_D is 400ns.

Rising Edge D-Flipflop		
CLK	D	Q(t)
0	0	Q (t – 1)
0	1	Q (t – 1)
↓	0	0
↓	1	1

To verify the correctness of the Rising Edge D-FF FF built using a symbol generated from VHDL code. We can compare the instances where the input IN_CLK is transitioning from (0 to 1) i.e., when a rising edge occurs in the CLK signal to the Rising Edge D-FF truth table. In this case, the first rising edge occurs between 0 ns and 160ns where IN_D is low, we can see that OUT_Q is also low. The second rising edge occurs between 160ns and 320ns where IN_D is high, and we can see that OUT_Q is also high. Looking at the timestamps between those instances, we can see that OUT_Q is holding onto the previous value of Q unless IN_D changes during the CLK's rising edge. With that information, we can map the rest of the inputs and outputs and conclude that the waveform simulation is correct.

```
1  Library IEEE;
2  USE IEEE.Std_logic_1164.all;
3
4  entity FallingEdgeDFlipFlop is
5  port
6     (
7        Q : out std_logic;
8        Clk :in std_logic;
9        D :in  std_logic
10    );
11
12 end FallingEdgeDFlipFlop;
13
14 architecture Behavioral of FallingEdgeDFlipFlop is
15 begin
16  process(Clk)
17  begin
18
19     if(falling_edge(Clk)) then
20     Q <= D;
21     end if;
22  end process;
23  end Behavioral;
```

Figure 14: VHDL code of a Falling Edge D-FF.

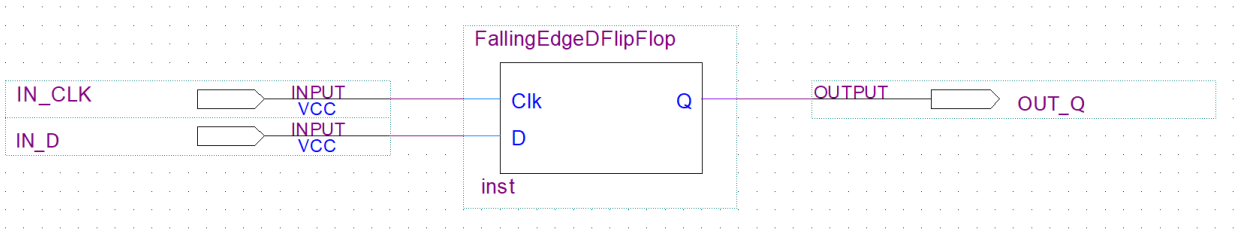

Figure 15: Circuit Diagram created using the symbol generated from the VHDL code in figure 14.

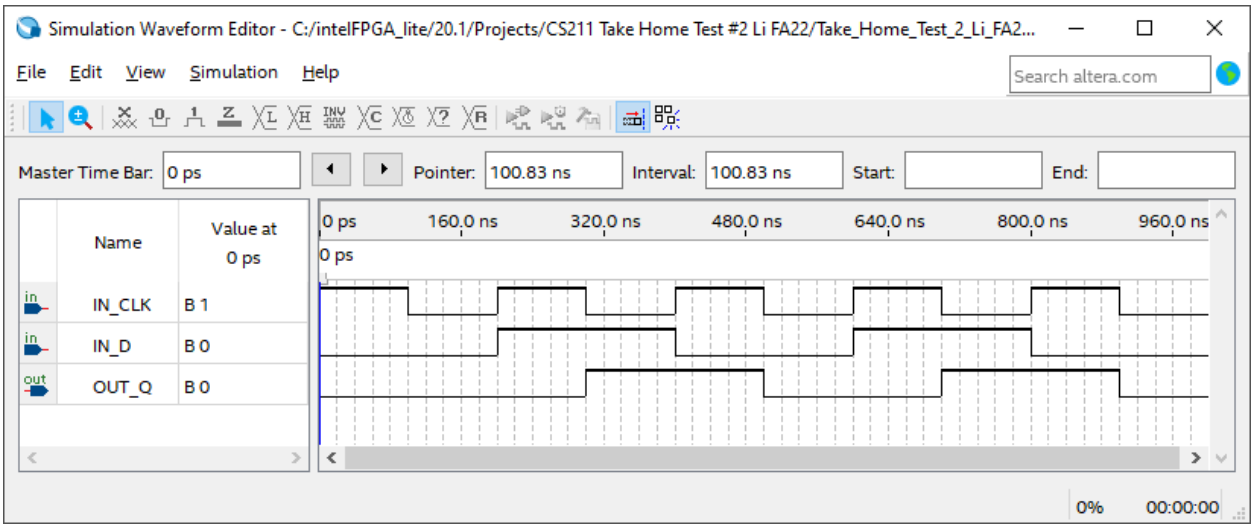

Figure 16: University VWF output for the Falling Edge D-FF built in figure 15 with input signals IN_CLK, IN_D and output signal OUT_Q. The duty cycle remains at 50%, however the period of IN_CLK is 200ns and inverted and IN_D is 400ns.

Falling Edge D-Flipflop		
CLK	D	Q(t)
↓	0	0
↓	1	1
1	0	Q (t – 1)
1	1	Q (t – 1)

To verify the correctness of the Falling Edge D-FF built using a symbol generated from VHDL code. We can compare the instances where the input IN_CLK is transitioning from (1 to 0) i.e., when a falling edge occurs in the CLK signal to the Falling Edge D-FF truth table. In this case, the first falling edge occurs between 0 ns and 160ns where IN_D is low, we can see that OUT_Q is also low. The second falling edge occurs between 160ns and 320ns where IN_D is high, and we can see that OUT_Q is also high. Looking at the timestamps between those instances, we can see that OUT_Q is holding onto the previous value of Q unless IN_D changes during the CLK's falling edge. With that information, we can map the rest of the inputs and outputs and conclude that the waveform simulation is correct.

Part 6: (Design and build in Quartus Rising Edge D-FF using Quartus component symbol for D-FlipFlop)

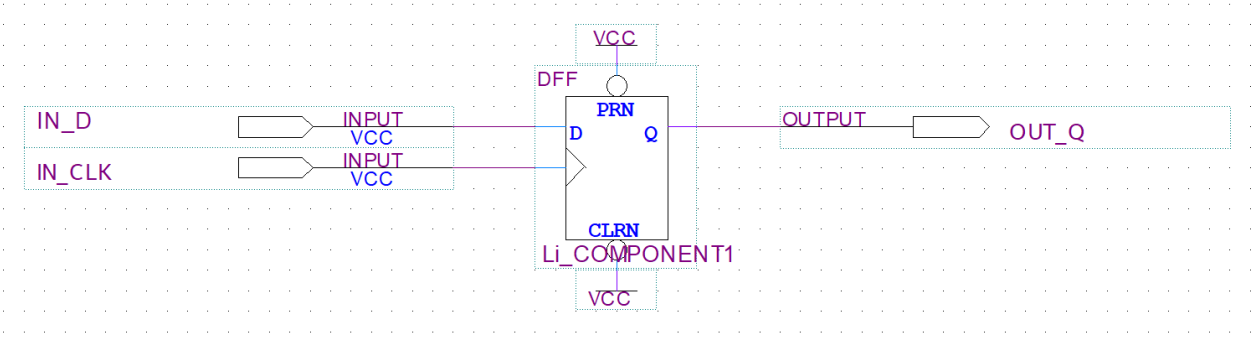

Figure 17: Rising Edge D-FF built using the DFF found in Quartus's library of components.

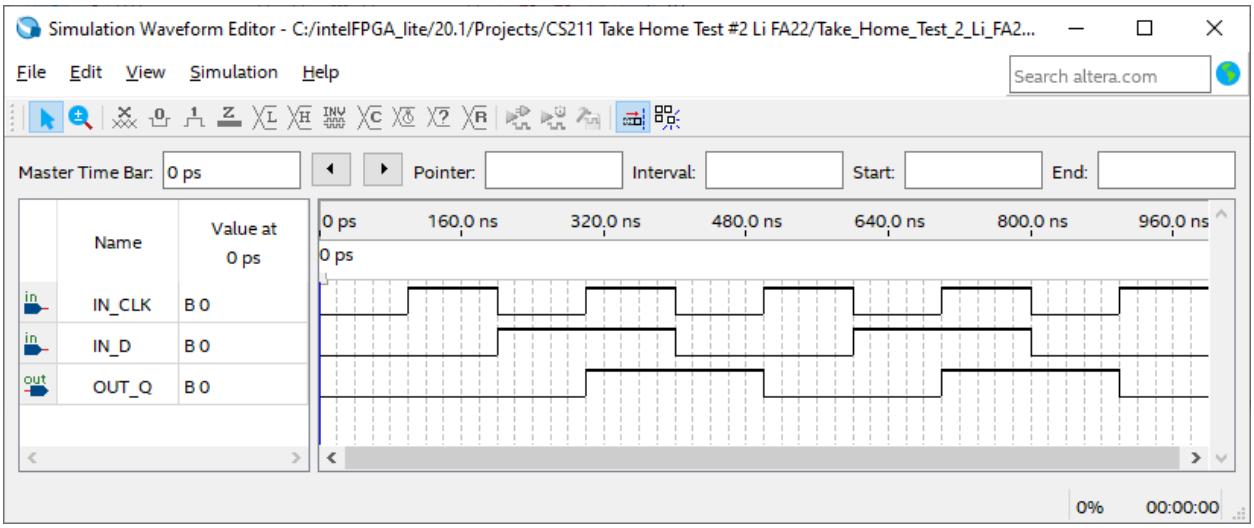

Figure 18: University VWF output for the Rising Edge D-FF built in figure 17 with input signals IN_CLK, IN_D and output signal OUT_Q. The duty cycles of the signals remain at 50%, however the period of IN_CLK is 200ns and IN_D is 400ns.

Part 7: (Design and build in Quartus Falling Edge D-FF using Quartus component symbol for D-FlipFlop)

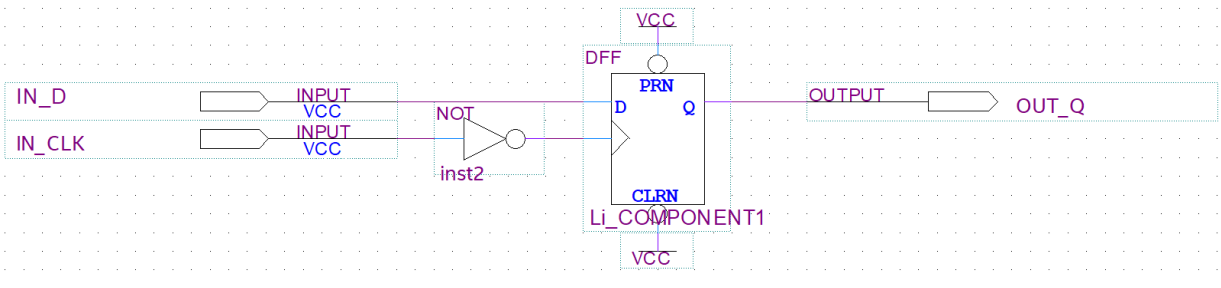

Figure 19: Falling Edge D-FF built using the DFF found in Quartus's library of components.

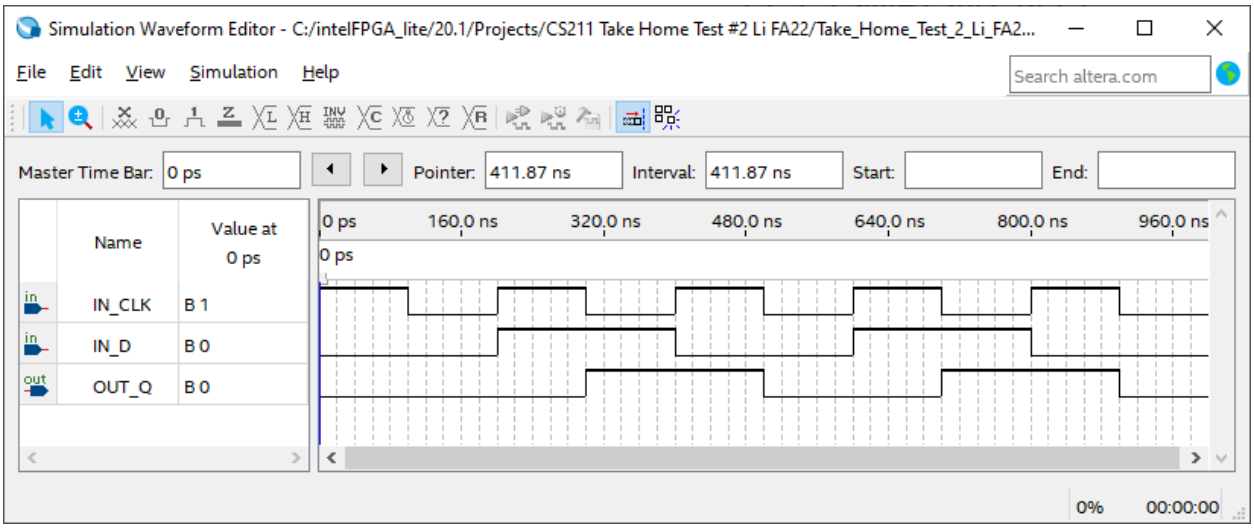

Figure 20: University VWF output for the Falling Edge D-FF built in figure 19 with input signals IN_CLK, IN_D and output signal OUT_Q. The duty cycle remains at 50%, however the period of IN_CLK is 200ns and inverted and IN_D is 400ns.

Part 8: (Compare in simulation case 5 with cases 7 and 8.)

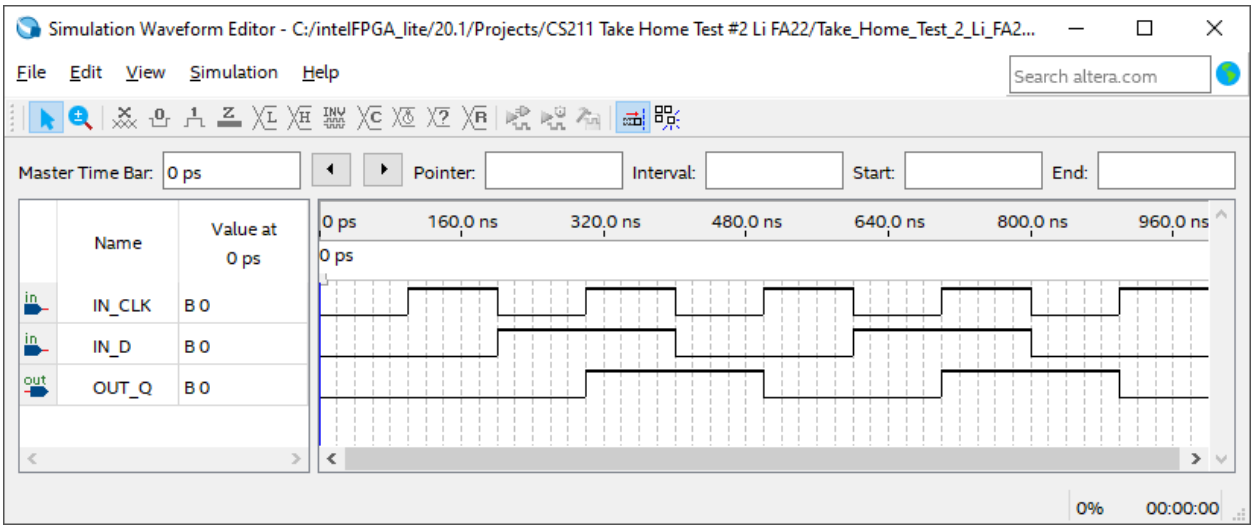

Figure 21: University VWF output for the Rising Edge D-FF built in figure 12 with input signals IN_CLK, IN_D and output signal OUT_Q. The duty cycles of the signals remain at 50%, however the period of IN_CLK is 200ns and IN_D is 400ns.

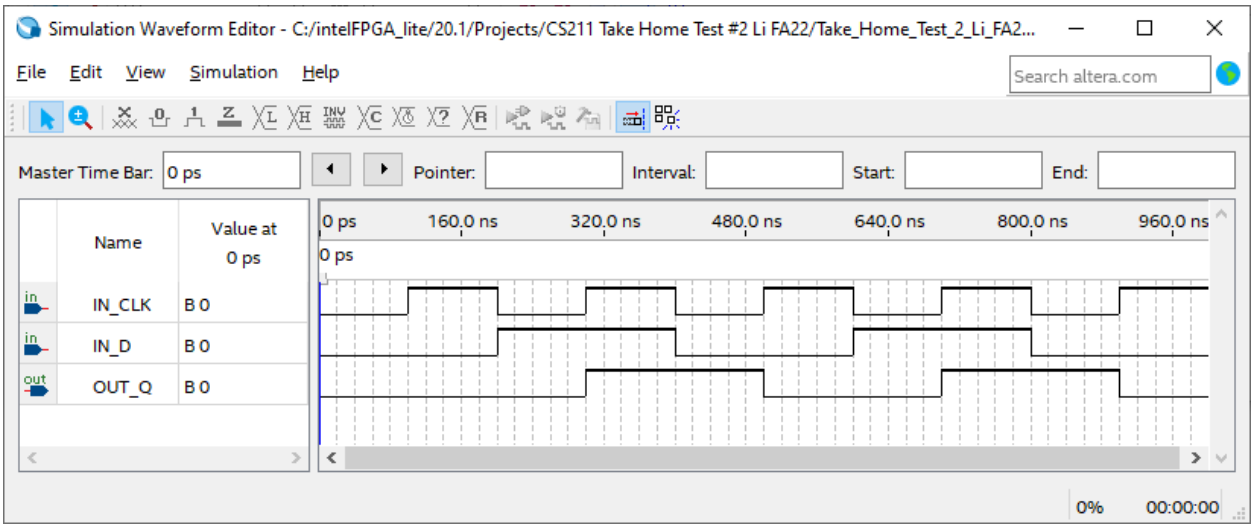

Figure 22: University VWF output for the Rising Edge D-FF built in figure 17 with input signals IN_CLK, IN_D and output signal OUT_Q. The duty cycles of the signals remain at 50%, however the period of IN_CLK is 200ns and IN_D is 400ns.

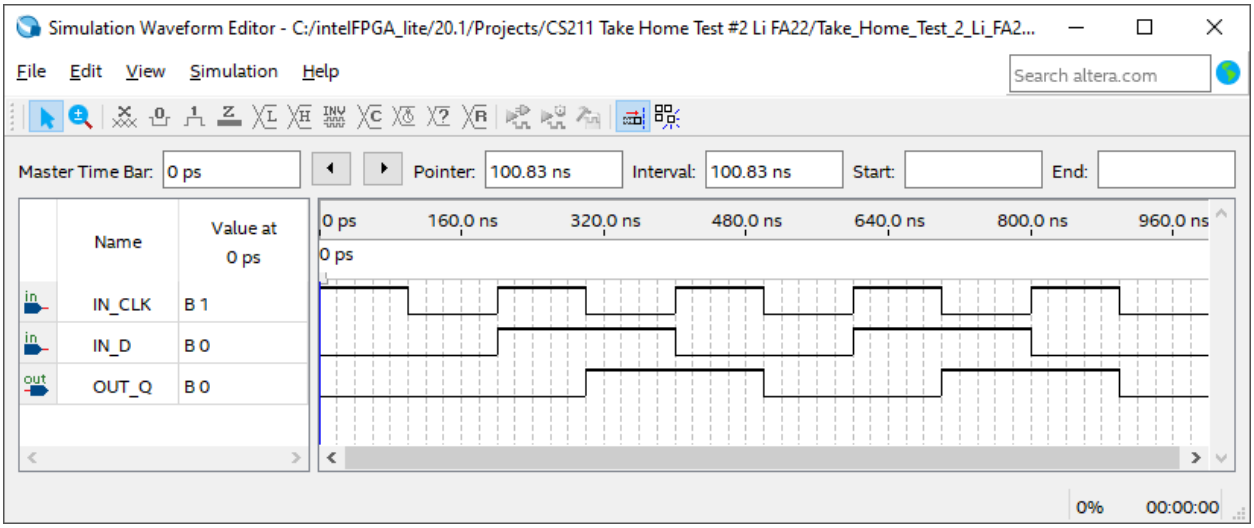

Figure 23: University VWF output for the Falling Edge D-FF built in figure 15 with input signals IN_CLK, IN_D and output signal OUT_Q. The duty cycle remains at 50%, however the period of IN_CLK is 200ns and inverted and IN_D is 400ns.

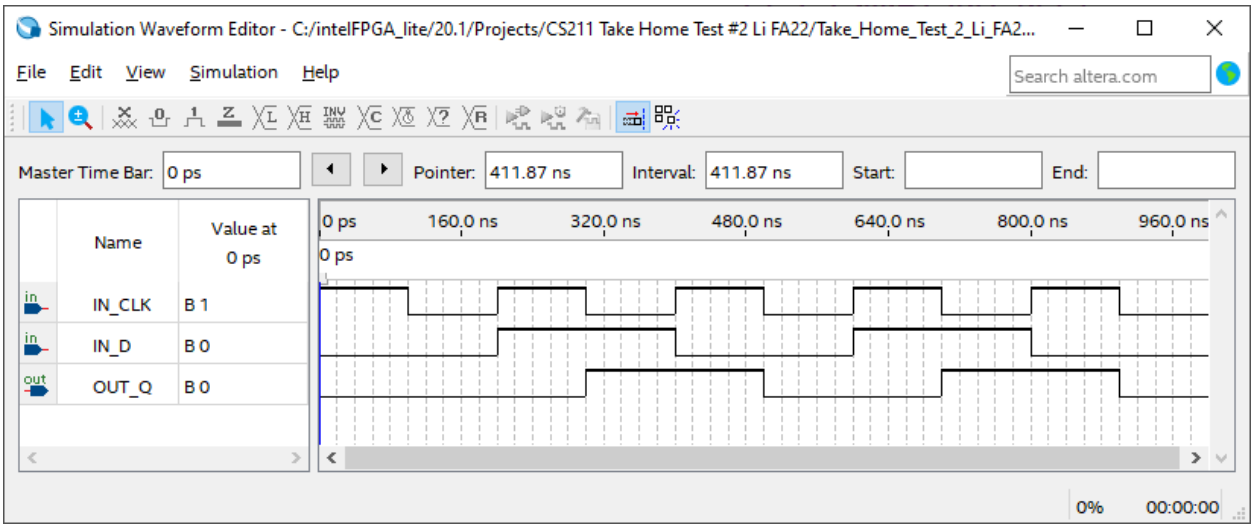

Figure 24: University VWF output for the Falling Edge D-FF built in figure 19 with input signals IN_CLK, IN_D and output signal OUT_Q. The duty cycle remains at 50%, however the period of IN_CLK is 200ns and inverted and IN_D is 400ns.

Comparing the waveforms from Part 5.a to Part 7 and Part 5.b to Part 8, we can see that there are no differences between the output OUT_Q of the digital circuits when their inputs IN_CLK and IN_D are the same. With the knowledge of the fact that Part 7 and 8 are built using Quartus library components, we know that the waveform simulation are correct. Comparing and constrasting them, we can see that they have no differences at all, therefore the waveforms built using VHDL code are correct.

Part 9: (Create in simulation timing diagram shown on the next page for all devices

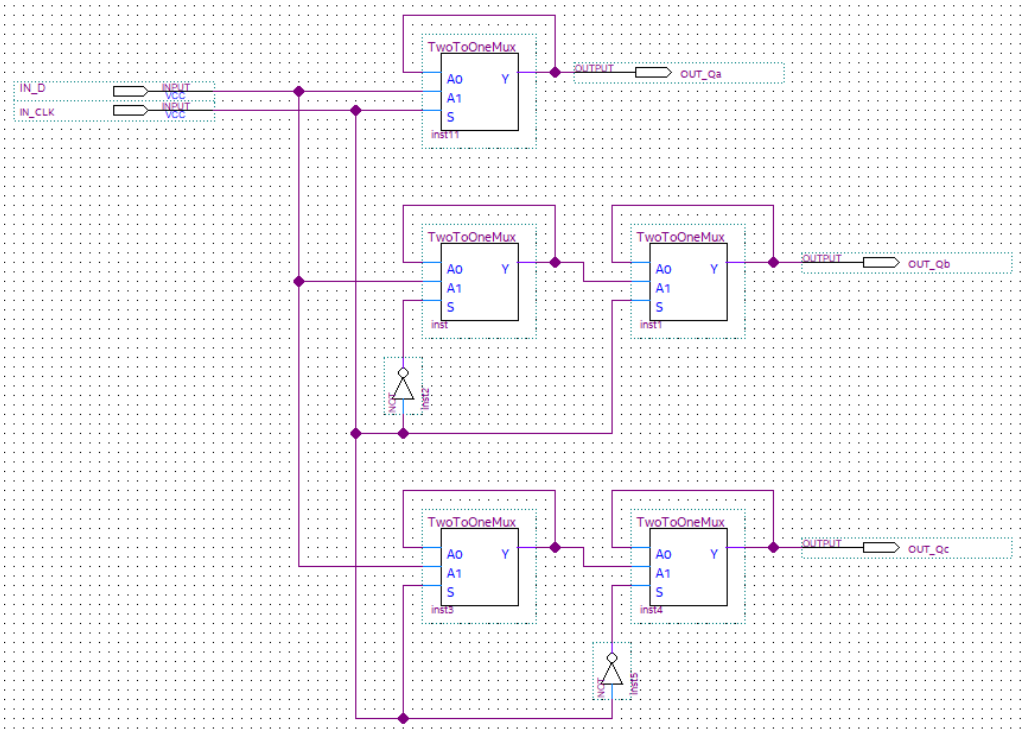

Figure 25: Circuit diagram of D-Latch, Rising Edge D-FF, Falling Edge D-FF with inputs IN_D, IN_CLK and outputs OUT_Qa, OUT_Qb, and OUT_Qc representing D-Latch, Rising Edge D-FF, and Falling Edge D-FF respectively.

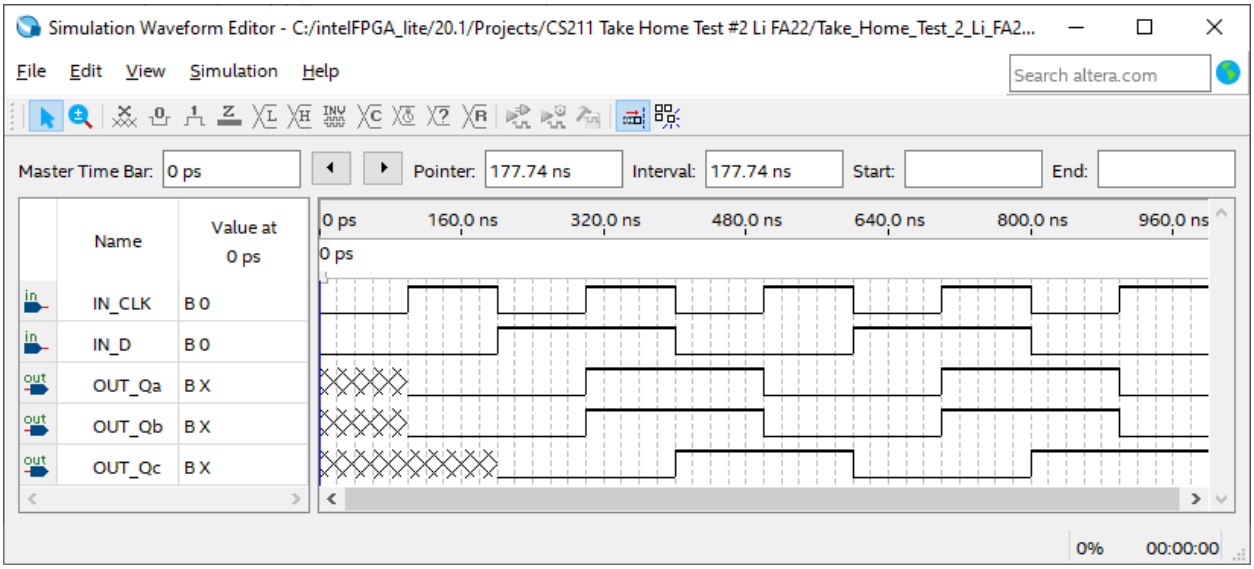

Figure 26: University VWF output for the circuit built in figure 21 with input signals IN_CLK, IN_D and output signal OUT_Qa, OUT_Qb, and OUT_Qc representing D-Latch, Rising Edge D-FF, and Falling Edge D-FF respectively. The duty cycle of the inputs remains at 50%, however the period of IN_CLK is 200ns and IN_D is 400ns.

D Latch		
CLK	D	Q(t)
0	0	Q (t – 1)
0	1	Q (t – 1)
1	0	0
1	1	1

Rising Edge D-Flipflop		
CLK	D	Q(t)
0	0	Q (t – 1)
0	1	Q (t – 1)
↑	0	0
↑	1	1

Falling Edge D-Flipflop		
CLK	D	Q(t)
↓	0	0
↓	1	1
1	0	Q (t – 1)
1	1	Q (t – 1)

To verify the correctness of the 3 different waveform simulations we can compare OUT_Qa to the truth table of the D-Latch, OUT_Qb to the truth table of the rising edge D-Flipflop, and OUT_Qc to the truth table of the falling edge D-Flipflop. For the waveform of OUT_Qa, we can see when IN_CLK is high and IN_D is low that OUT_Qa is also low. We can also see for OUT_Qa when IN_CLK is high and IN_D is high that OUT_Qa is also high. Otherwise, OUT_Qa is always holding onto the previous output of OUT_Qa until IN_CLK is high and IN_D changes state. For the waveform of OUT_Qb, we can see when IN_CLK changes from 0 to 1 and IN_D is 0 that OUT_Qb is also 0. We can also see for OUT_Qb when IN_CLK changes from 0 to 1 and IN_D is 1 that OUT_Qb is also 1. Otherwise, OUT_Qb is always holding onto the previous output of OUT_Qb unless IN_CLK transitions from 0 to 1 and IN_D changes state. For the waveform of OUT_Qc, we can see when IN_CLK changes from 1 to 0 and IN_D is 0 that OUT_Qc is also 0. We can also see for OUT_Qc when IN_CLK changes from 1 to 0 and IN_D is 1 that OUT_Qc is also 1. Otherwise, OUT_Qc is always holding onto the previous output of OUT_Qc unless IN_CLK transitions from 1 to 0 and IN_D changes state. With all this information, we can compare this to their respective truth tables and determine that the waveform simulations are correct.

Conclusions:

In this report, I learned about another application of the 2:1 Multiplexer. Previously, I learned that the 2:1 Multiplexer can be wired to act like a D-Latch. This time, I learned that two 2:1 Multiplexers can be wired in such a way to function like a D-Flipflop. By generating a latch using a mux and using back-to-back latches at different clock edges, two 2:1 Multiplexers can function like a flip flop. I also learned about Rising Edge as well as Falling edge and their applications in digital electronic circuits. Rising Edge refers to when the clock pulse transitions from low to high (0 to 1) and falling edge refers to when the clock pulse transitions from high to low (1 to 0).