City College of New York

Take Home Test # 1

Zi Xuan Li

CSC 211 Fall 2022

Professor Izidor Gertner

November 6th, 2022

Objective:

The objective of this report is to build, simulate, and verify the correctness of both Positive and Negative D–Latches that are built using 2:1 Multiplexers. This report will simulate D-Latches made using the template of a 2:1 Multiplexer symbol as well as 2:1 Multiplexer made using AND, OR, and NOT gates. Alongside that, the report will also include VHDL code for a Positive D-Latch as well as Negative D-Latch.

Functionality, Specifications, and Simulation:

To verify the correctness of these circuits we must understand how a Positive D-Latch as well as a Negative D-Latch works.

A Positive D-Latch works the exact same as a typical D-Latch. The functionality of the D-Latch is meant to capture or "latch" the logic which is present on the data signal when the CLK signal is high. That means, if the data on the D signal changes state (from 1 to 0 or from 0 to 1) while the CLK signal is high, then the output, Q, follows the input, D. When the CLK signal is low, the last state of the D signal is trapped and held in the latch and the output Q retains the previous value of D (the value before the CLK drops to 0).

A Negative D-Latch works similarly to the Positive D-Latch except that the output Q follows the input D when the CLK signal is low instead of high. That means, if the data on the D signal changes state while the CLK is low, then the output, Q, follows the input, D. When the CLK signal is high, the output Q retains the previous value of D (the value before the CLK rises to 1).

Using this information, we can form a truth table of a Positive D-Latch as well as a Negative D-Latch.

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

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

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

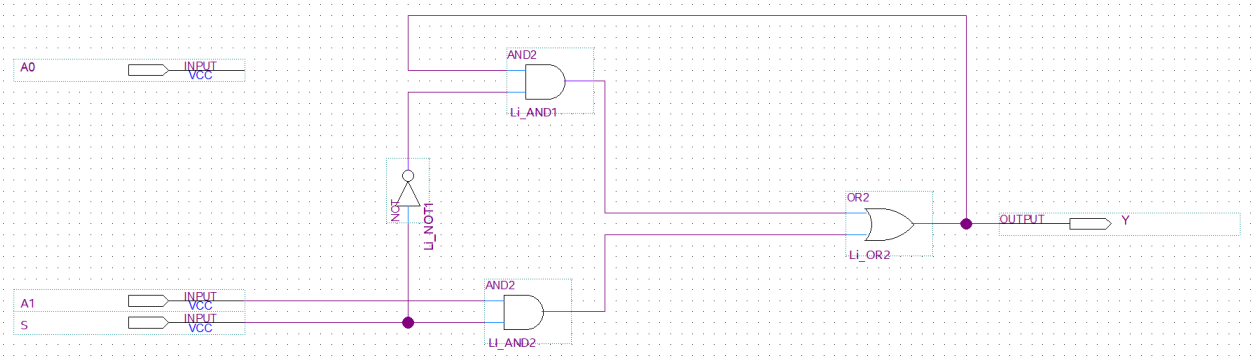

Figure 1: Positive D-Latch digital circuit built using the 2:1 Mux as a template with gates that we used in the first lab.

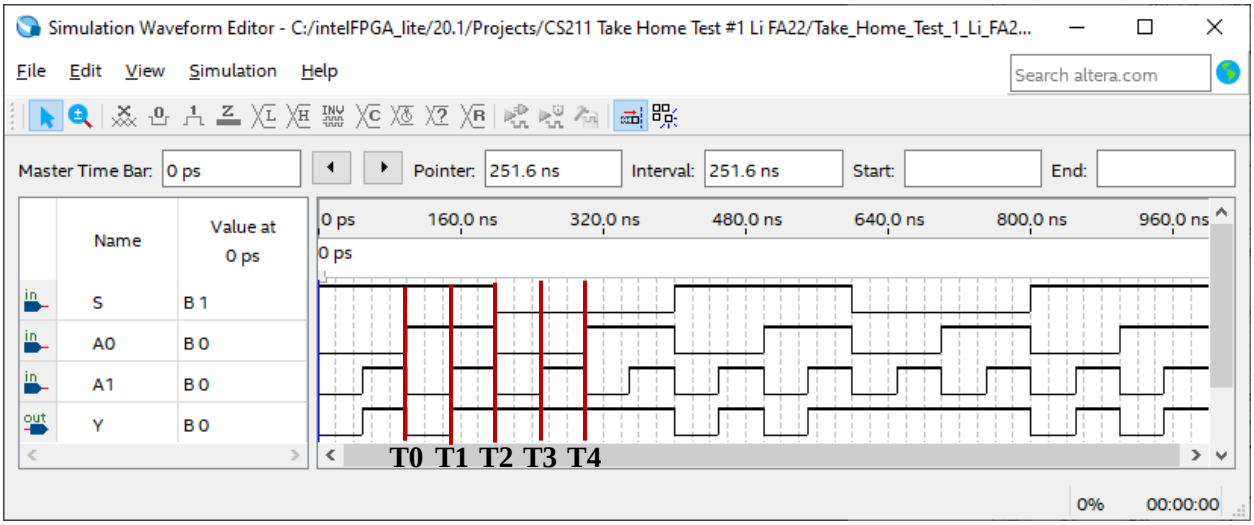

Figure 2: University Program VWF output for the D-Latch in figure 1 with input signals S, A0, A1 and output signals Y. In this waveform simulation, S represents standard input ENA/CLK with a period of 400ns and duty cycle of 50%. A0 is not used in the digital circuit. A1 represents standard input D with a period of 100 ns and duty cycle of 50%. Y represents the standard output Q of the D-Latch.

To verify the correctness of the digital circuit in Figure 1, I compared the VWF in Figure 2 with the truth table of a standard Positive D-Latch. In Figure 2, T0 to T1 shows that the output of the digital circuit when S (CLK) is HIGH and A1 (D) is LOW results in Y (Q) being LOW. T1 to T2 shows that the output when S is HIGH and A1 is HIGH results in Y being HIGH. T2 to T3 shows that the output when S is LOW and A1 is LOW results in Y being HIGH (latch case). T3 to T4 shows that the output when S is LOW and A1 is HIGH results in Y being HIGH (latch case). The results of the VWF matches all the possible cases of the Positive D-Latch truth table and thus it is correct.

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

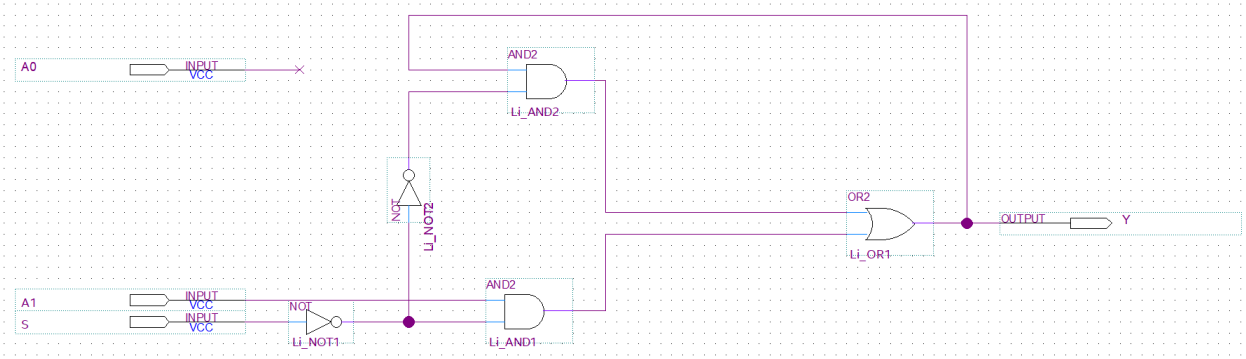

Figure 3: Negative D-Latch digital circuit built using the 2:1 Mux as a template with gates that we used in the first lab.

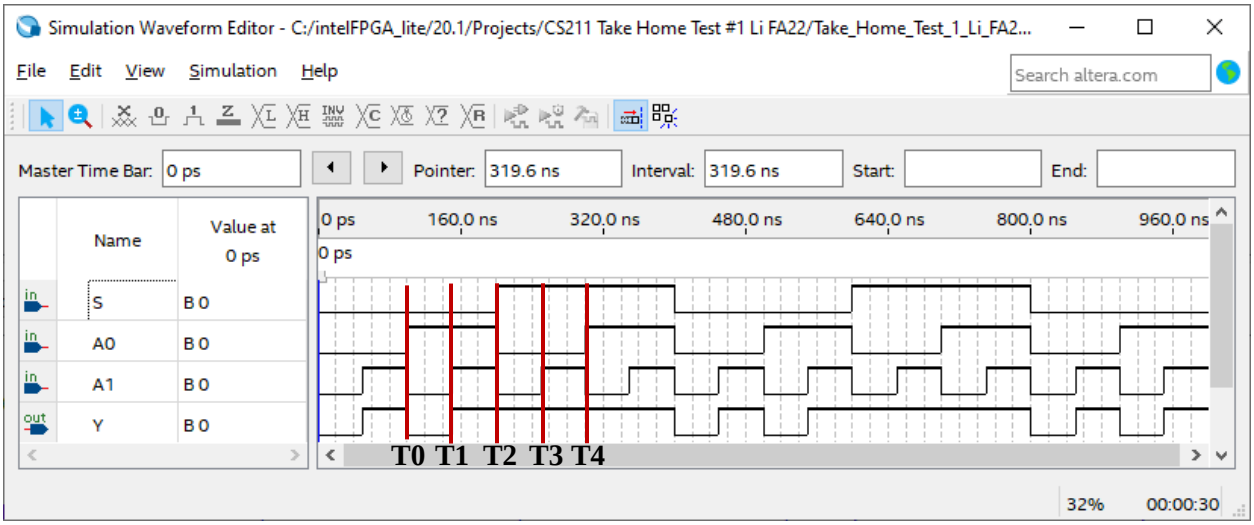

Figure 4: University Program VWF output for the D-Latch in figure 3 with input signals S, A0, A1 and output signals Y. In this waveform simulation, S represents standard input ENA/CLK with a period of 400ns and duty cycle of 50%. A0 is not used in the digital circuit. A1 represents standard input D with a period of 100 ns and duty cycle of 50%. Y represents the standard output Q of the D-Latch.

To verify the correctness of the digital circuit in Figure 3, I compared the VWF in Figure 4 with the truth table of a standard Negative D-Latch. In Figure 4, T0 to T1 shows that the output of the digital circuit when S (CLK) is LOW and A1 (D) is LOW results in Y (Q) being LOW. T1 to T2 shows that the output when S is LOW and A1 is HIGH results in Y being HIGH. T2 to T3 shows that the output when S is HIGH and A1 is LOW results in Y being HIGH (latch case). T3 to T4 shows that the output when S is HIGH and A1 is HIGH results in Y being HIGH (latch case). The results of the VWF matches all the possible cases of the Negative D-Latch truth table and thus is correct.

Part 3: (Design and build in Quartus Positive D-Latch using 2:1 MUX using Quartus component symbol)

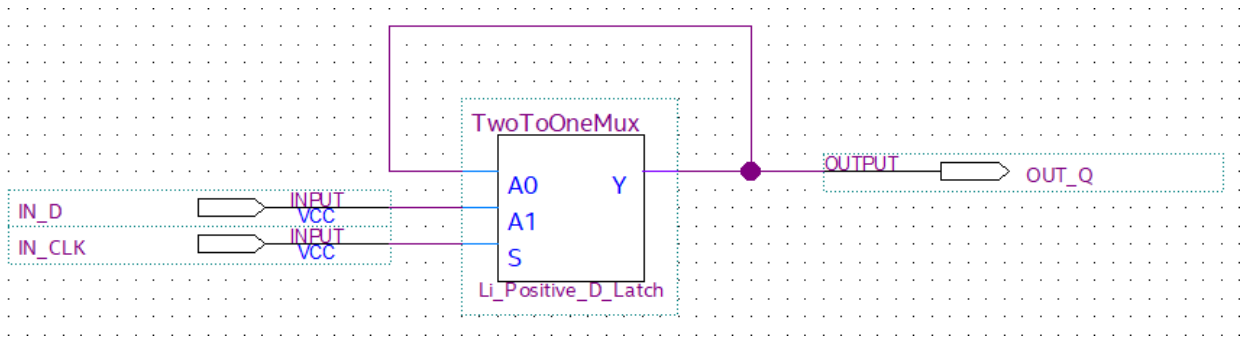

Figure 5: Positive D-Latch digital circuit built using the 2:1 Mux symbol as a component.

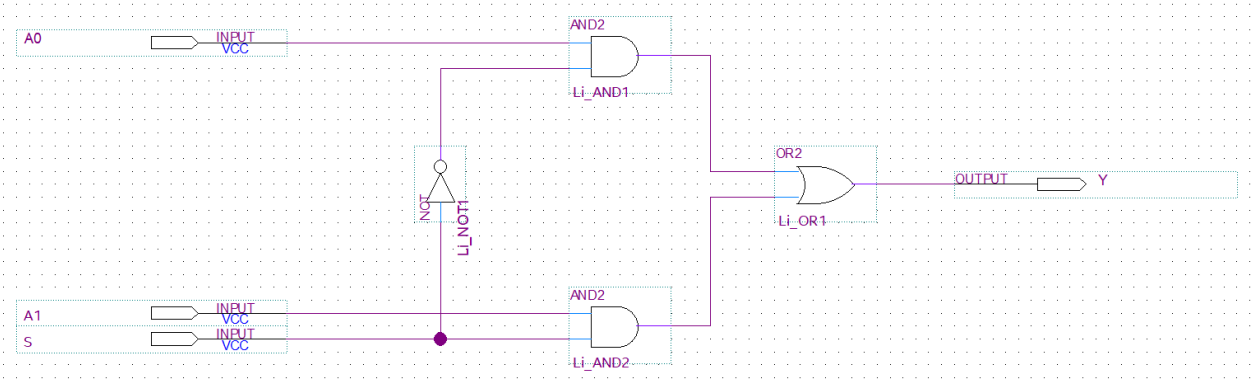

Figure 6: Digital Circuit of the 2:1 Mux Symbol being used in Figure 5.

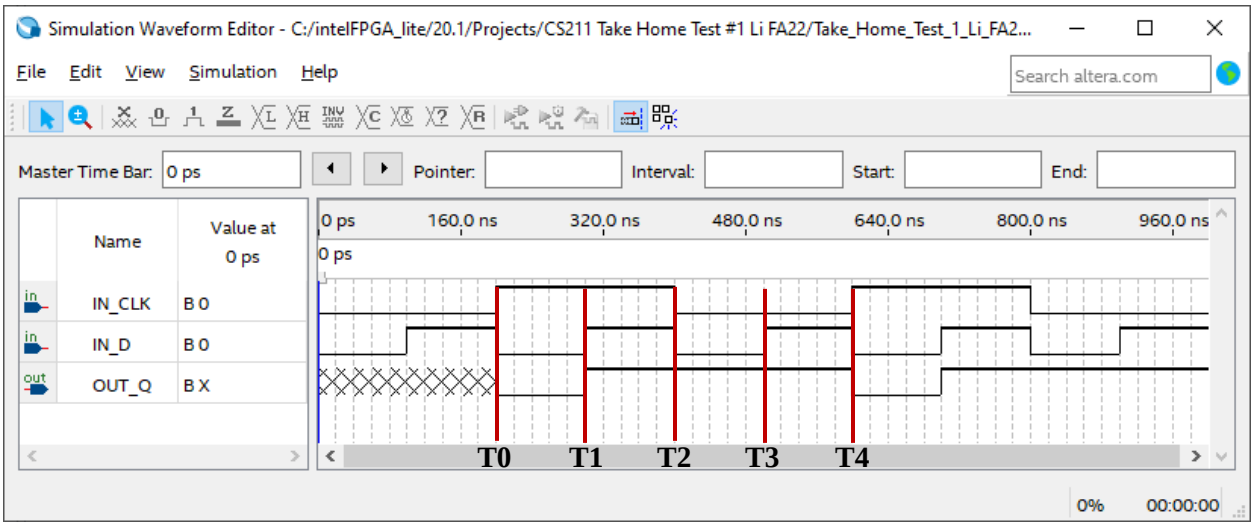

Figure 7: University Program VWF output for the D-Latch in figure 5 with input signals IN_CLK and IN_D and output signal OUT_Q. In this waveform simulation, IN_CLK has a period of 400ns and duty cycle of 50%. IN_D has a period of 200ns and duty cycle of 50%. OUT_Q represents the output.

To verify the correctness of the digital circuit in Figure 5, I compared the VWF in Figure 7 with the truth table of a standard Positive D-Latch. In Figure 7, T0 to T1 shows that the output of the digital circuit when IN_CLK is HIGH and IN_D is LOW results in OUT_Q being LOW. T1 to T2 shows that the output when IN_CLK is HIGH and IN_D is HIGH results in OUT_Q being HIGH. T2 to T3 shows that the output when IN_CLK is LOW and IN_D is LOW results in OUT_Q being HIGH (latch case). T3 to T4 shows that the output when IN_CLK is low and IN_D is HIGH results in OUT_Q being HIGH (latch case). The results of the VWF matches all the possible cases of the Positive D-Latch truth table and thus is correct.

Part 4: (Design and build in Quartus Negative D-Latch using 2:1 MUX using Quartus component symbol)

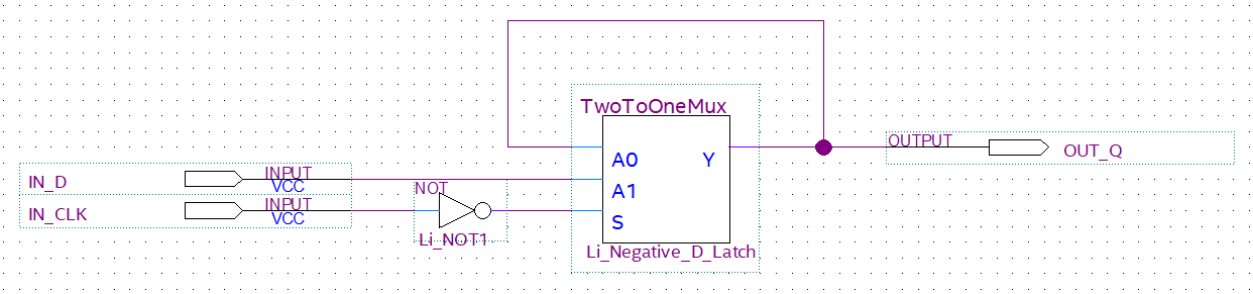

Figure 8: Negative D-Latch digital circuit built using the 2:1 Mux symbol as a component (2 to 1 Mux component is the same refer to figure 6).

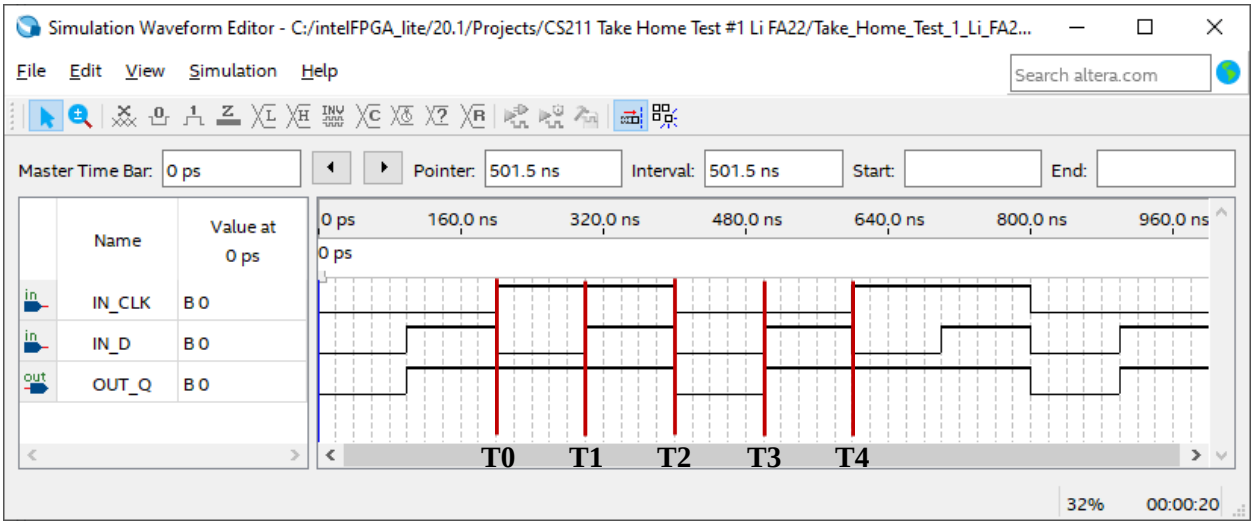

Figure 9: University Program VWF output for the D-Latch in figure 5 with input signals IN_CLK and IN_D and output signal OUT_Q. In this waveform simulation, IN_CLK has a period of 400ns and duty cycle of 50%. IN_D has a period of 200ns and duty cycle of 50%. OUT_Q represents the output.

To verify the correctness of the digital circuit in Figure 8, I compared the VWF in Figure 9 with the truth table of a standard Negative D-Latch. In Figure 9, T0 to T1 shows that the output of the digital

circuit when IN_CLK is HIGH and IN_D is LOW results in OUT_Q being HIGH (latch case). T1 to T2 shows that the output when IN_CLK is HIGH and IN_D is HIGH results in OUT_Q being HIGH (latch case). T2 to T3 shows that the output when IN_CLK is LOW and IN_D is LOW results in OUT_Q being LOW. T3 to T4 shows that the output when IN_CLK is low and IN_D is HIGH results in OUT_Q being HIGH. The results of the VWF matches all the possible cases of the Positive D-Latch truth table and thus is correct.

Part 5: (Write VHDL code for Positive D-Latch, Negative D-Latch. Create a separate Quartus project for each)

```
1   library IEEE;
2   use IEEE.STD_LOGIC_1164.ALL;
3
4   entity Positive_D_Latch is
5       Port ( D  : in  STD_LOGIC;
6              EN : in  STD_LOGIC;
7              Q  : out STD_LOGIC);
8   end Positive_D_Latch;
9
10  architecture Behavioral of Positive_D_Latch is
11      signal DATA : STD_LOGIC;
12  begin
13
14      DATA <= D when (EN = '1') else DATA;
15      Q <= DATA;
16
17  end Behavioral;
```

Figure 10: Positive D-Latch VHDL Code.

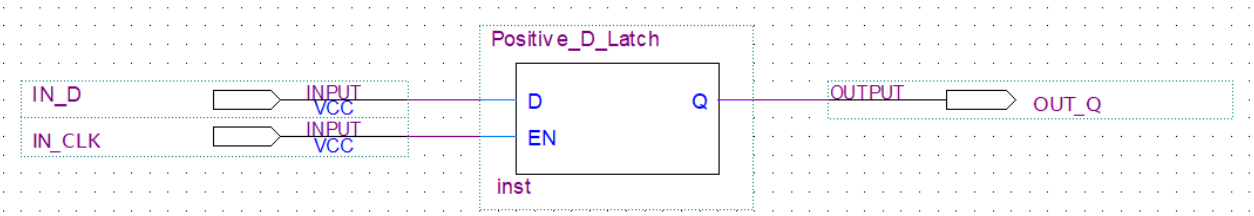

Figure 11: BDF file of the VHDL code generated from Figure 10 into a symbol with inputs and outputs.

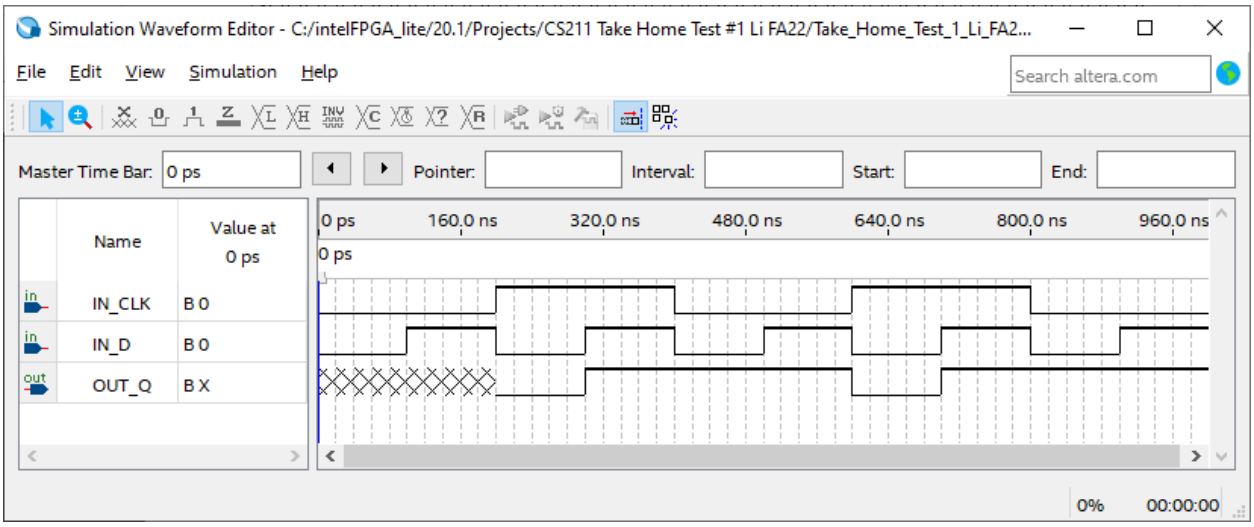

Figure 12: VWF output of the VHDL Code in Figure 10 generated as a symbol with input signals IN_CLK and IN_D and output signal OUT_Q.

```vhdl
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity Negative_D_Latch is
    Port ( D  : in  STD_LOGIC;
           EN : in  STD_LOGIC;
           Q  : out STD_LOGIC);
end Negative_D_Latch;

architecture Behavioral of Negative_D_Latch is
    signal DATA : STD_LOGIC;
begin

    DATA <= D when (EN = '0') else DATA;
    Q <= DATA;

end Behavioral;
```

Figure 13: Negative D-Latch VHDL Code.

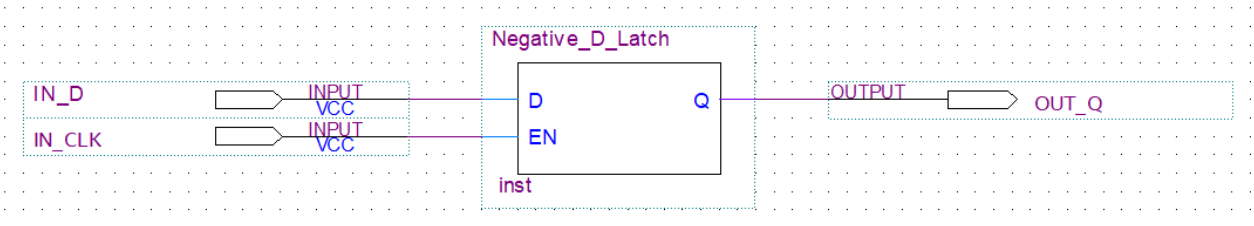

Figure 14: BDF file of the VHDL code generated from figure 13 into a symbol with inputs and outputs.

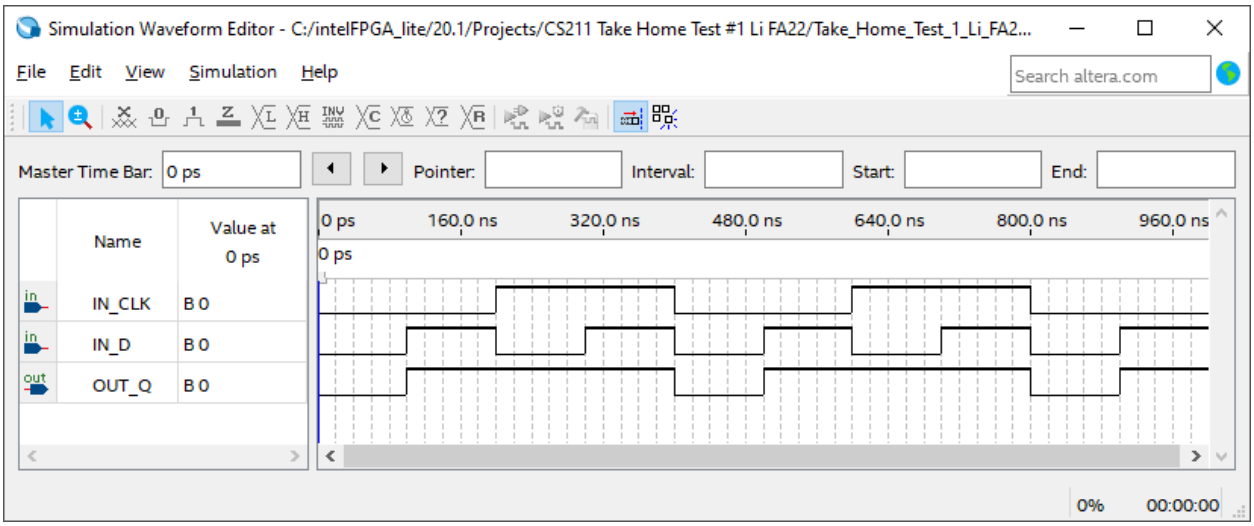

Figure 15: VWF output of the VHDL Code in Figure 13 generated as a symbol with input signals IN_CLK and IN_D and output signal OUT_Q.

```
1  -- Copyright (C) 2020  Intel Corporation. All rights reserved.
2  -- Your use of Intel Corporation's design tools, logic functions
3  -- and other software and tools, and any partner logic
4  -- functions, and any output files from any of the foregoing
5  -- (including device programming or simulation files), and any
6  -- associated documentation or information are expressly subject
7  -- to the terms and conditions of the Intel Program License
8  -- Subscription Agreement, the Intel Quartus Prime License Agreement,
9  -- the Intel FPGA IP License Agreement, or other applicable license
10 -- agreement, including, without limitation, that your use is for
11 -- the sole purpose of programming logic devices manufactured by
12 -- Intel and sold by Intel or its authorized distributors.  Please
13 -- refer to the applicable agreement for further details, at
14 -- https://fpgasoftware.intel.com/eula.
15
16 -- PROGRAM     "Quartus Prime"
17 -- VERSION     "Version 20.1.1 Build 720 11/11/2020 SJ Lite Edition"
18 -- CREATED     "Sun Nov 06 20:14:43 2022"
19
20 LIBRARY ieee;
21 USE ieee.std_logic_1164.all;
22
23 LIBRARY work;
24
25 ENTITY UpdatedPositiveDataLatchUsingGatesVHDL IS
26    PORT
27    (
28       A0 :  IN  STD_LOGIC;
29       A1 :  IN  STD_LOGIC;
30       S :  IN  STD_LOGIC;
31       Y :  OUT  STD_LOGIC
32    );
33 END UpdatedPositiveDataLatchUsingGatesVHDL;
34
35 ARCHITECTURE bdf_type OF UpdatedPositiveDataLatchUsingGatesVHDL IS
36
37 SIGNAL   SYNTHESIZED_WIRE_0 :  STD_LOGIC;
38 SIGNAL   SYNTHESIZED_WIRE_1 :  STD_LOGIC;
39 SIGNAL   SYNTHESIZED_WIRE_2 :  STD_LOGIC;
40 SIGNAL   SYNTHESIZED_WIRE_3 :  STD_LOGIC;
41
42
43 BEGIN
44 Y <= SYNTHESIZED_WIRE_0;
45
46
47
48 SYNTHESIZED_WIRE_3 <= SYNTHESIZED_WIRE_0 AND SYNTHESIZED_WIRE_1;
49
50
51 SYNTHESIZED_WIRE_2 <= A1 AND S;
52
53
54 SYNTHESIZED_WIRE_1 <= NOT(S);
55
56
57
58 SYNTHESIZED_WIRE_0 <= SYNTHESIZED_WIRE_2 OR SYNTHESIZED_WIRE_3;
59
60
61 END bdf_type;
```

Figure 16: Positive D-Latch digital VHDL Code generated using Quartus HDL Design.

```
-- Copyright (C) 2020  Intel Corporation. All rights reserved.
-- Your use of Intel Corporation's design tools, logic functions
-- and other software and tools, and any partner logic
-- functions, and any output files from any of the foregoing
-- (including device programming or simulation files), and any
-- associated documentation or information are expressly subject
-- to the terms and conditions of the Intel Program License
-- Subscription Agreement, the Intel Quartus Prime License Agreement,
-- the Intel FPGA IP License Agreement, or other applicable license
-- agreement, including, without limitation, that your use is for
-- the sole purpose of programming logic devices manufactured by
-- Intel and sold by Intel or its authorized distributors.  Please
-- refer to the applicable agreement for further details, at
-- https://fpgasoftware.intel.com/eula.

-- PROGRAM		"Quartus Prime"
-- VERSION		"Version 20.1.1 Build 720 11/11/2020 SJ Lite Edition"
-- CREATED		"Sun Nov 06 20:16:18 2022"

LIBRARY ieee;
USE ieee.std_logic_1164.all;

LIBRARY work;

ENTITY UpdatedNegativeDataLatchUsingGatesVHDL IS
    PORT
    (
        A0 :  IN  STD_LOGIC;
        A1 :  IN  STD_LOGIC;
        S :  IN  STD_LOGIC;
        Y :  OUT  STD_LOGIC
    );
END UpdatedNegativeDataLatchUsingGatesVHDL;

ARCHITECTURE bdf_type OF UpdatedNegativeDataLatchUsingGatesVHDL IS

SIGNAL	SYNTHESIZED_WIRE_6 :  STD_LOGIC;
SIGNAL	SYNTHESIZED_WIRE_1 :  STD_LOGIC;
SIGNAL	SYNTHESIZED_WIRE_2 :  STD_LOGIC;
SIGNAL	SYNTHESIZED_WIRE_4 :  STD_LOGIC;
SIGNAL	SYNTHESIZED_WIRE_5 :  STD_LOGIC;


BEGIN
Y <= SYNTHESIZED_WIRE_1;



SYNTHESIZED_WIRE_4 <= A1 AND SYNTHESIZED_WIRE_6;


SYNTHESIZED_WIRE_5 <= SYNTHESIZED_WIRE_1 AND SYNTHESIZED_WIRE_2;


SYNTHESIZED_WIRE_6 <= NOT(S);



SYNTHESIZED_WIRE_2 <= NOT(SYNTHESIZED_WIRE_6);



SYNTHESIZED_WIRE_1 <= SYNTHESIZED_WIRE_4 OR SYNTHESIZED_WIRE_5;


END bdf_type;
```

Figure 17: Negative D-Latch digital VHDL Code generated using Quartus HDL Design.

Conclusions:

In this report, I learned about one of the many applications of the 2:1 Multiplexer. One of them being a D-Latch. The equation of a 2:1 Mux is Y = SA * 0 + S' * A1 and we can rewrite that to fit the purpose of D-Latch because the logic of the D-Latch can be interpreted as if Clock is high, then Q follows D else Q follows Q which can be written as Clock * D + Clock' * Q. I also learned about the difference between positive and Negative D-Latches. Positive D-Latch refers to when the clock pulse transitions from a low to high (0 to 1) and is called a rising edge. Negative D-Latch refers to when the clock pulse transitions from high to low (1 to 0) and is called a falling edge. Through this report, I also developed a better understanding of D-Latches, Multiplexers, and writing VHDL code.