Zi Xuan Li

Professor Izidor Gertner

CS 211

October 23rd, 2022

Laboratory Exercise 5: Advanced MUX, Decoder, Encoder, and LPM Report

Objective:

- The objective of this lab was to apply everything that we learned about the Quartus application from the tutorials to build circuits such as the 8-bit 2:1 Mux, 5:1 Mux, 3-bit 5:1 Mux, 2:4 Decoder, 8:3 Encoder, and a Demultiplexer and verify correctness of said circuits using waveform simulations.

Functionality, Specifications, and Simulation:

I verified the correctness of each of the individual circuits with the truth table that was prescribed in the lab and in the circumstance that there was no truth table provided, I found a truth table of said circuit online to compare with the vector waveform output data to verify correctness.

- Part I.A (8-Bit 2 to 1 Mux)

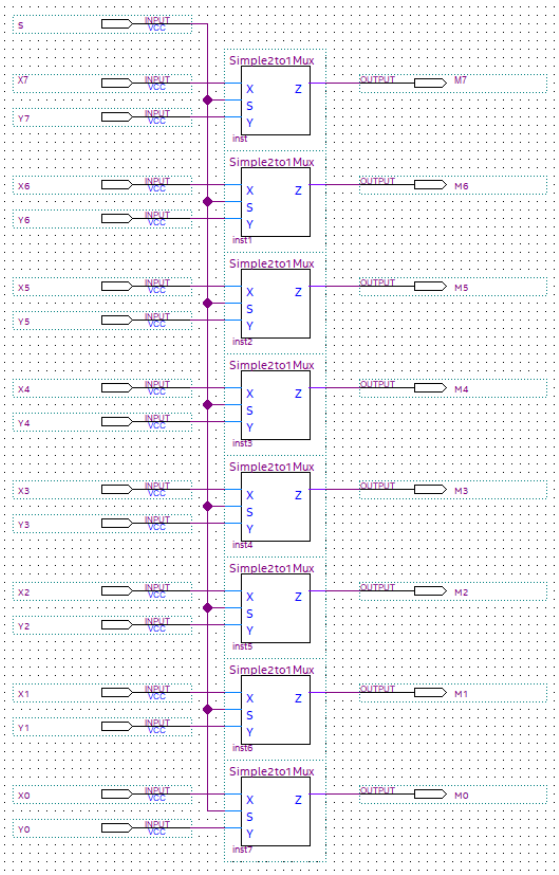

Figure 1: The figure above is a screenshot of the 8-bit 2 to 1 Mux

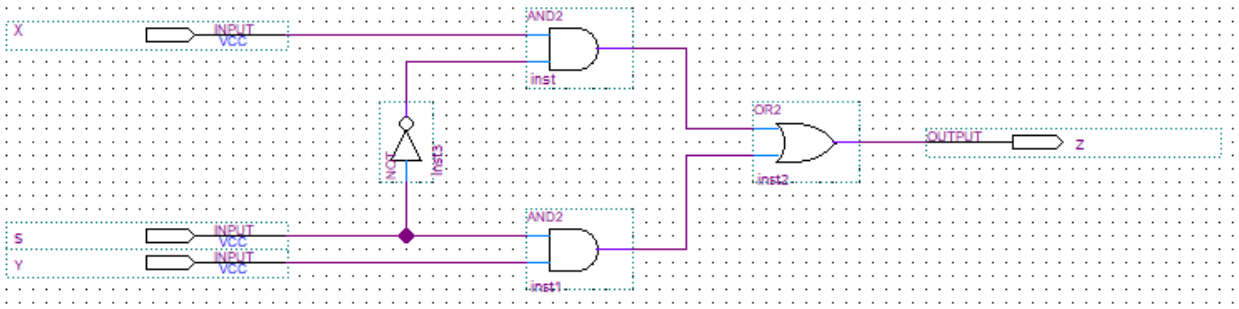

Figure 2: The figure above is a screenshot of the circuit diagram of the 2 to 1 Mux being used in the 8-bit 2 to 1 Mux circuit.

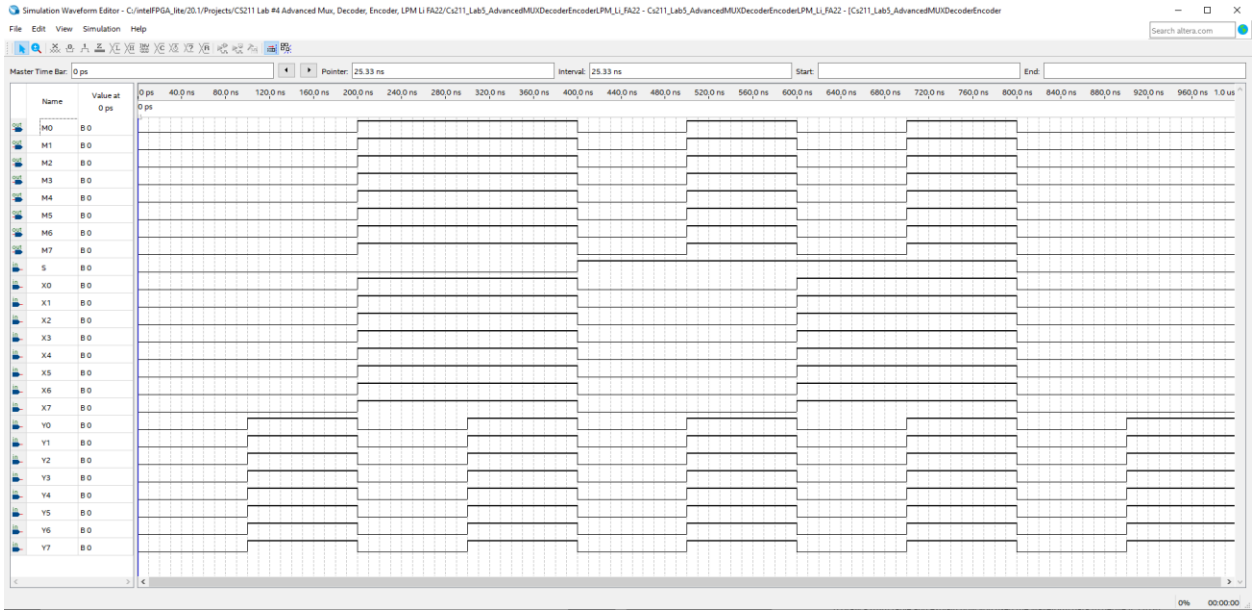

Figure 3: The figure above is a screenshot of the vector waveform output file of the 8 bit 2 to 1 Mux circuit.

- Part I.B (5 to 1 Mux)

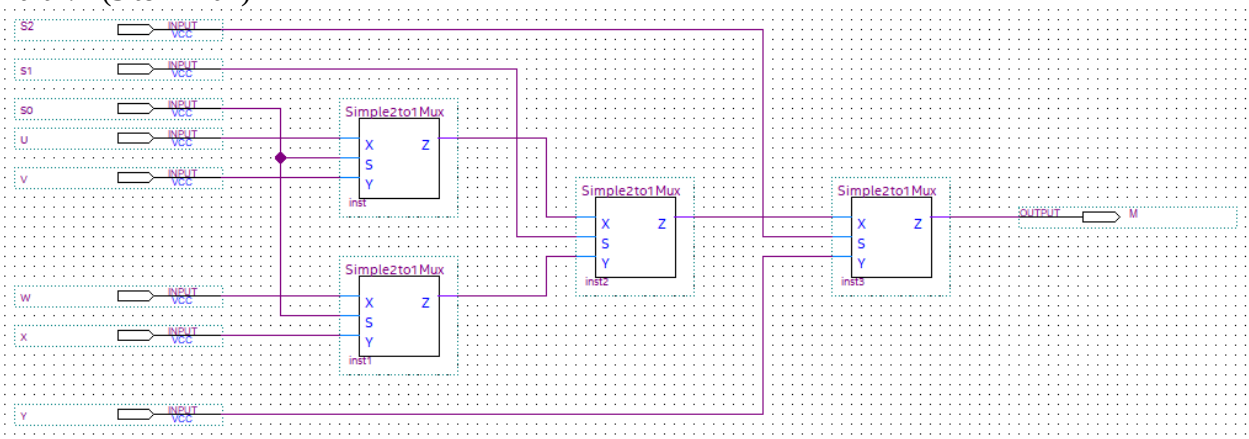

Figure 4: The figure above is a screenshot of the 1-bit 5 to 1 Mux circuit built using 2 to 1 Mux as a component. The simple 2 to 1 Mux component is the same component as figure 2.

Figure 5: The figure above is a screenshot of the vector waveform output file of the 1-bit 5 to 1 Mux circuit.

- Part I.C (3-bit 5 to 1 Mux)

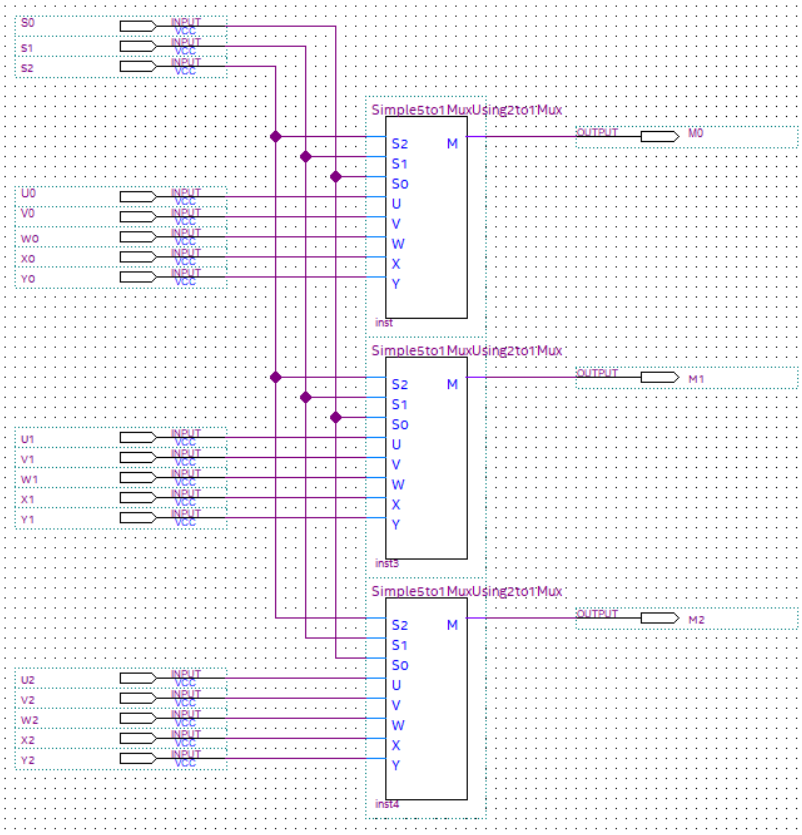

Figure 6: The figure above is a screenshot of the 3-bit 5 to 1 Mux circuit diagram. The simple 5 to 1 Mux using 2 to 1 Mux component is the same component as figure 4.

- Part II.0 (Finding the library function of the lpm_mux symbol)

No instructions were given here to submit a simulation, nor to verify correctness, nor to create a program FPGA.

- Part II.1 (2 to 4 Decoder)

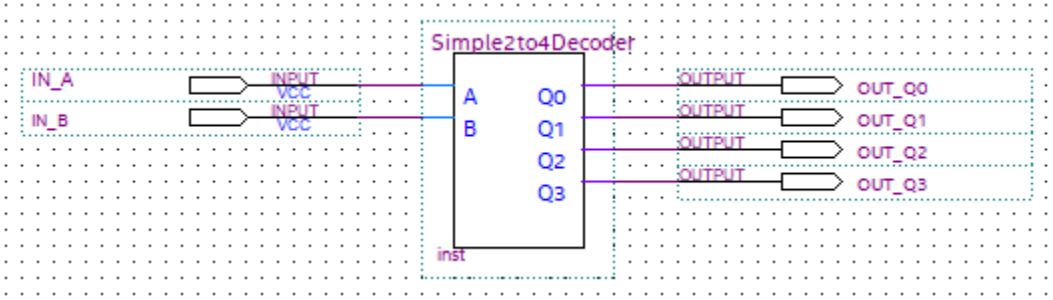

Figure 7: The figure above is a screenshot of the 2 to 4 Decoder.

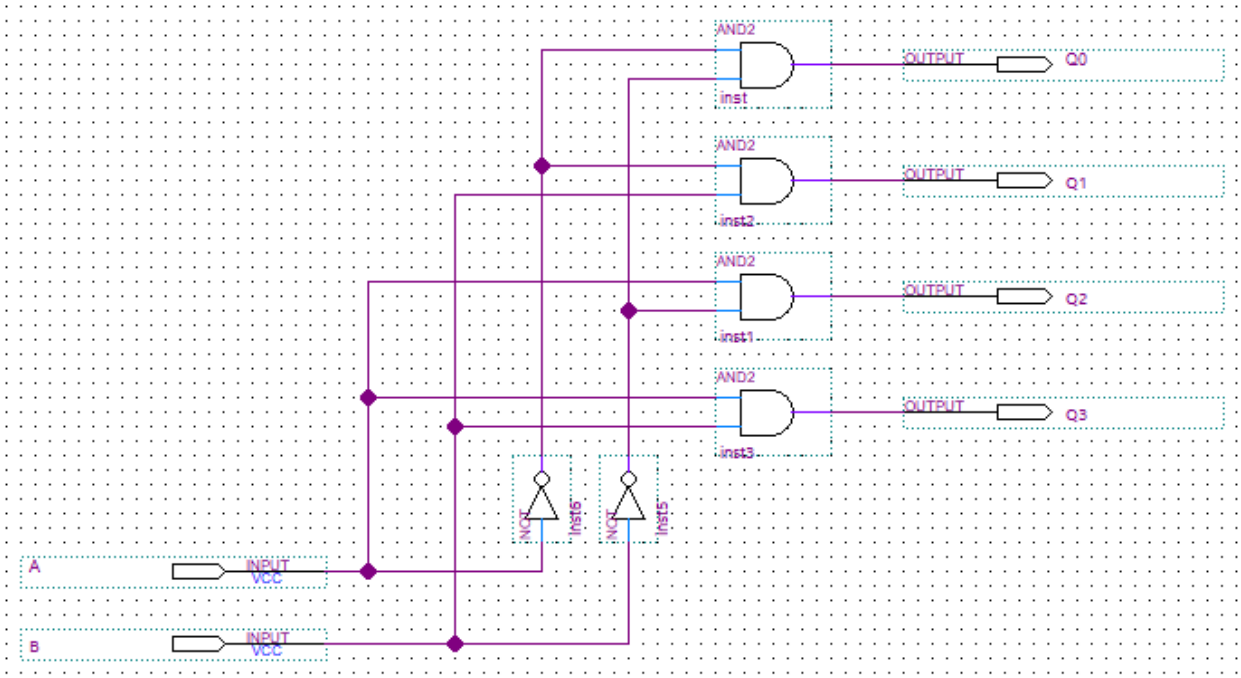

Figure 8: The figure above is a screenshot of the circuit diagram of the Simple 2 to 4 Decoder component used in 2 to 4 Decoder.

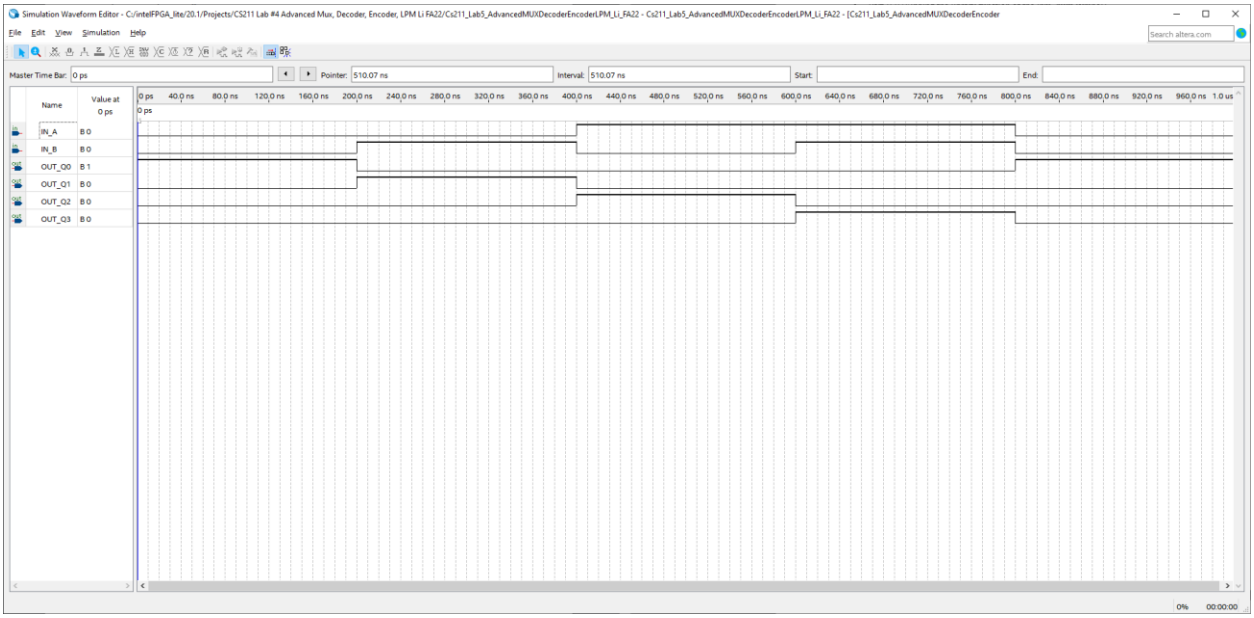

Figure 9: The figure above is a screenshot of the vector waveform output file of the 2 to 4 Decoder.

```
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     "Sat Oct 22 23:08:12 2022"
19
20 LIBRARY ieee;
21 USE ieee.std_logic_1164.all;
22
23 LIBRARY work;
24
25 ENTITY Simple2to4DecoderVHDL IS
26    PORT
27    (
28       A :  IN  STD_LOGIC;
29       B :  IN  STD_LOGIC;
30       Q0 :  OUT  STD_LOGIC;
31       Q1 :  OUT  STD_LOGIC;
32       Q2 :  OUT  STD_LOGIC;
33       Q3 :  OUT  STD_LOGIC
34    );
35 END Simple2to4DecoderVHDL;
36
37 ARCHITECTURE bdf_type OF Simple2to4DecoderVHDL IS
38
39 SIGNAL   SYNTHESIZED_WIRE_4 :  STD_LOGIC;
40 SIGNAL   SYNTHESIZED_WIRE_5 :  STD_LOGIC;
41
42
43 BEGIN
44
45
46
47 Q0 <= SYNTHESIZED_WIRE_4 AND SYNTHESIZED_WIRE_5;
48
49
50 Q2 <= A AND SYNTHESIZED_WIRE_5;
51
52
53 Q1 <= SYNTHESIZED_WIRE_4 AND B;
54
55
56 Q3 <= A AND B;
57
58
59 SYNTHESIZED_WIRE_5 <= NOT(B);
60
61
62
63 SYNTHESIZED_WIRE_4 <= NOT(A);
64
65
66
67 END bdf_type;
```

Figure 10: The figure above is a screenshot of the VHDL code of the 2 to 4 Decoder as requested in the lab exercise.

- Part II.2 (3 to 8 Decoder)

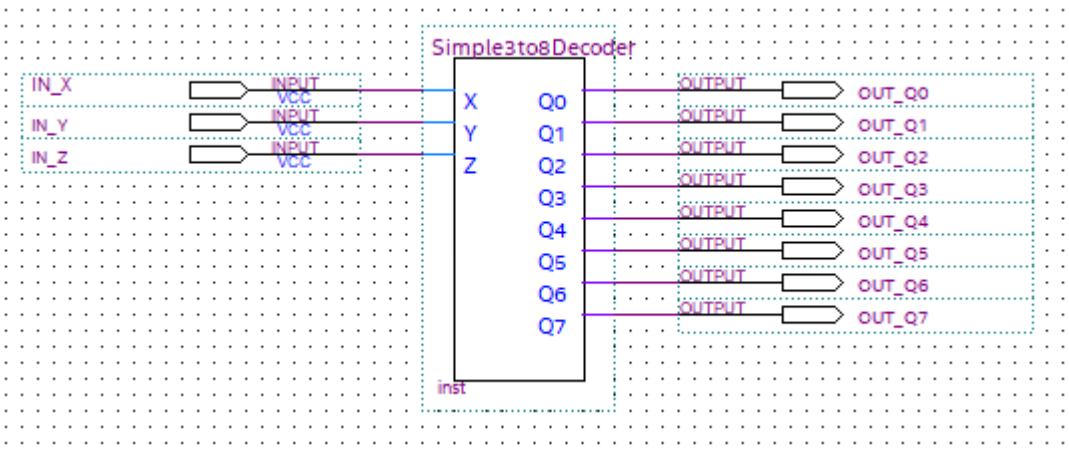

Figure 11: The figure above is a screenshot of the 3 to 8 Decoder.

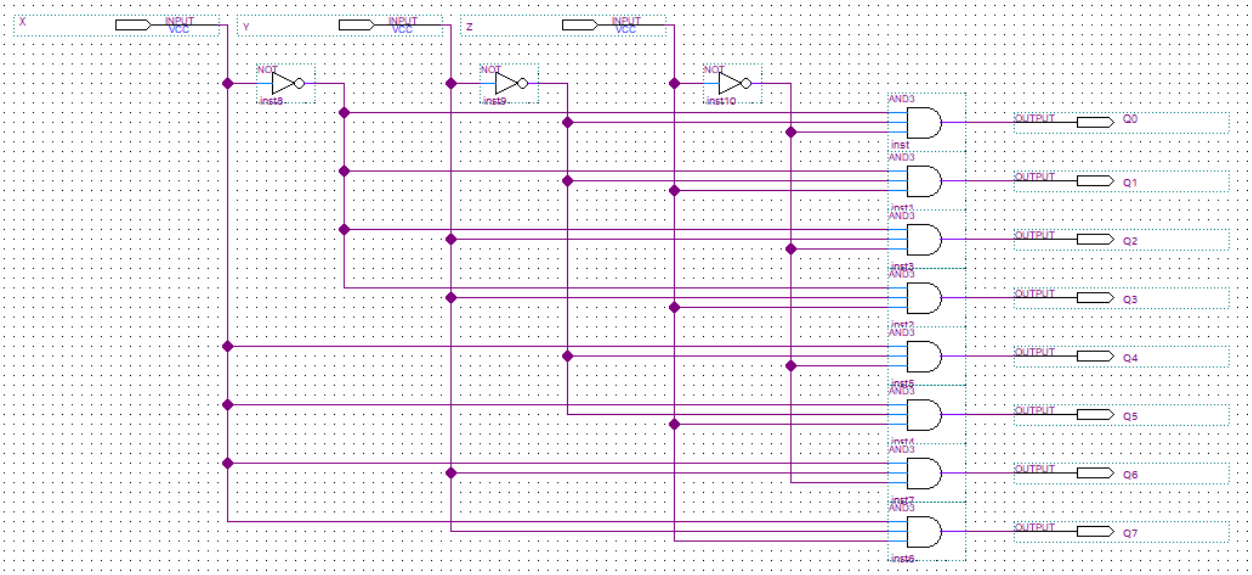

Figure 12: The figure above is a screenshot of the circuit diagram of the Simple 3 to 8 Decoder component used in 3 to 8 Decoder.

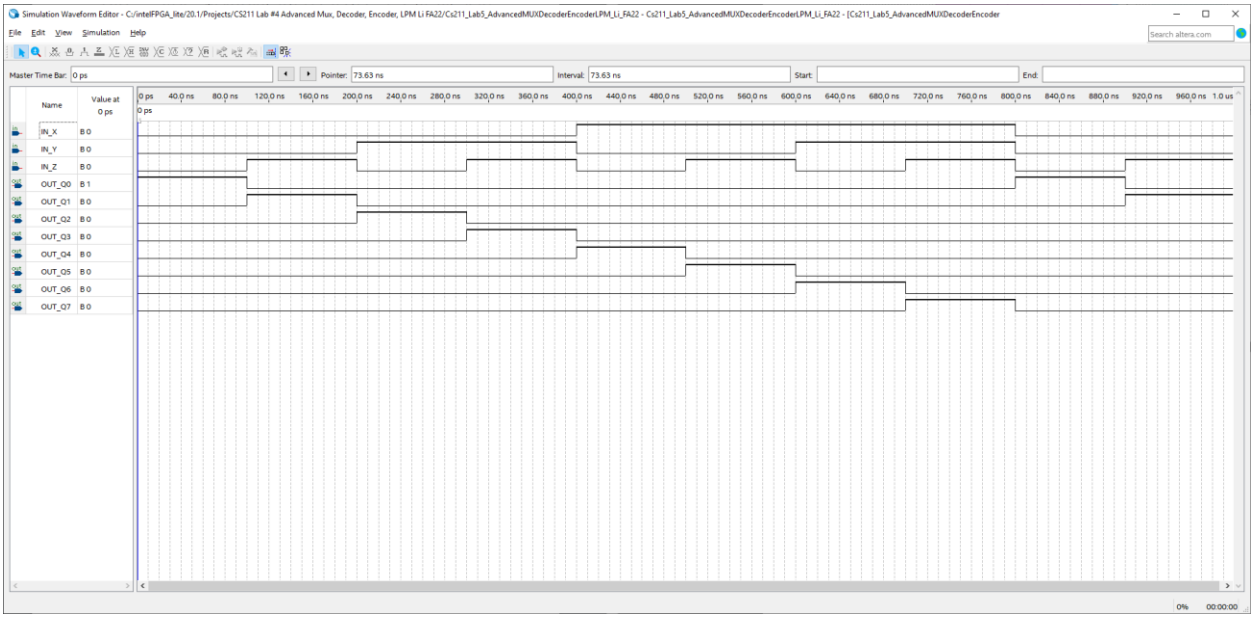

Figure 13: The figure above is a screenshot of the vector waveform output file of the 3 to 8 Decoder.

```
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 Oct 23 00:35:27 2022"
19
20 LIBRARY ieee;
21 USE ieee.std_logic_1164.all;
22
23 LIBRARY work;
24
25 ENTITY Simple3to8DecoderVHDL IS
26     PORT
27     (
28         X :  IN  STD_LOGIC;
29         Y :  IN  STD_LOGIC;
30         Z :  IN  STD_LOGIC;
31         Q0 :  OUT  STD_LOGIC;
32         Q1 :  OUT  STD_LOGIC;
33         Q2 :  OUT  STD_LOGIC;
34         Q3 :  OUT  STD_LOGIC;
35         Q4 :  OUT  STD_LOGIC;
36         Q5 :  OUT  STD_LOGIC;
37         Q6 :  OUT  STD_LOGIC;
38         Q7 :  OUT  STD_LOGIC
39     );
40 END Simple3to8DecoderVHDL;
41
42 ARCHITECTURE bdf_type OF Simple3to8DecoderVHDL IS
43
44 SIGNAL    SYNTHESIZED_WIRE_12 :  STD_LOGIC;
45 SIGNAL    SYNTHESIZED_WIRE_13 :  STD_LOGIC;
46 SIGNAL    SYNTHESIZED_WIRE_14 :  STD_LOGIC;
47
48
49 BEGIN
50
51
52
53 Q0 <= SYNTHESIZED_WIRE_12 AND SYNTHESIZED_WIRE_13 AND SYNTHESIZED_WIRE_14;
54
55
56 Q1 <= SYNTHESIZED_WIRE_12 AND SYNTHESIZED_WIRE_13 AND Z;
57
58
59 SYNTHESIZED_WIRE_14 <= NOT(Z);
60
61
62
63 Q3 <= SYNTHESIZED_WIRE_12 AND Y AND Z;
64
65
66 Q2 <= SYNTHESIZED_WIRE_12 AND Y AND SYNTHESIZED_WIRE_14;
67
68
69 Q5 <= X AND SYNTHESIZED_WIRE_13 AND Z;
70
71
72 Q4 <= X AND SYNTHESIZED_WIRE_13 AND SYNTHESIZED_WIRE_14;
73
74
75 Q7 <= X AND Y AND Z;
76
77
78 Q6 <= X AND Y AND SYNTHESIZED_WIRE_14;
79
80
81 SYNTHESIZED_WIRE_12 <= NOT(X);
82
83
84
85 SYNTHESIZED_WIRE_13 <= NOT(Y);
86
87
88
89 END bdf_type;
```

Figure 14: The figure above is a screenshot of the VHDL code of the 3 to 8 Decoder as requested in the lab exercise.

- Part II.3 (8 to 3 Encoder)

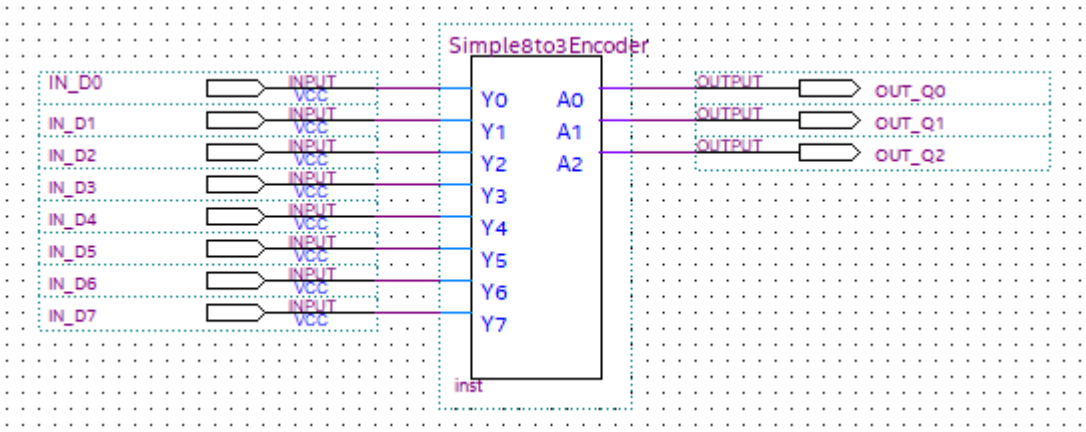

Figure 15: The figure above is a screenshot of the 8 to 3 Encoder.

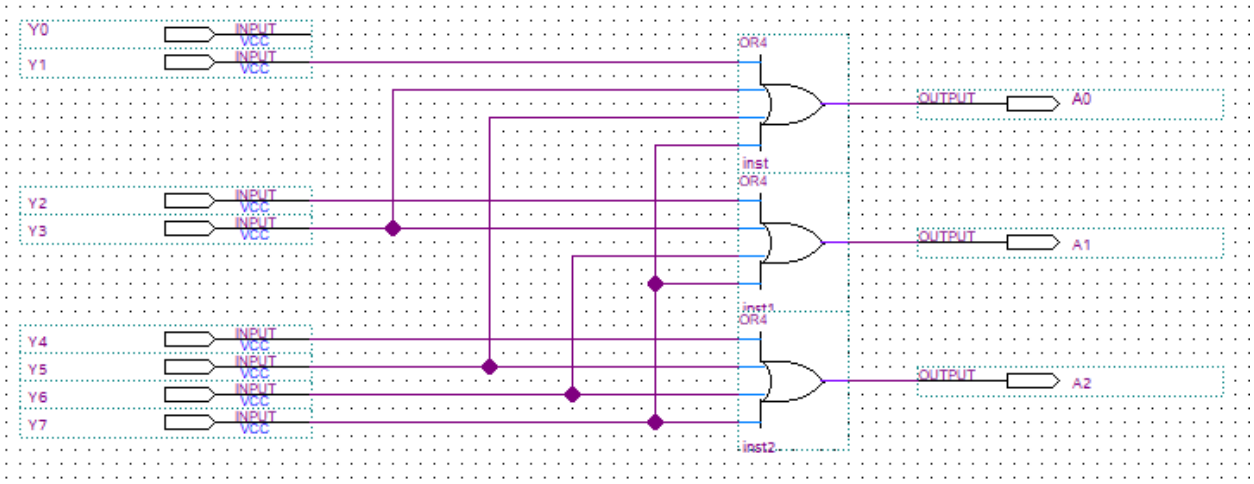

Figure 16: The figure above is a screenshot of the circuit diagram of the Simple 8 to 3 Encoder component used in 8 to 3 Encoder.

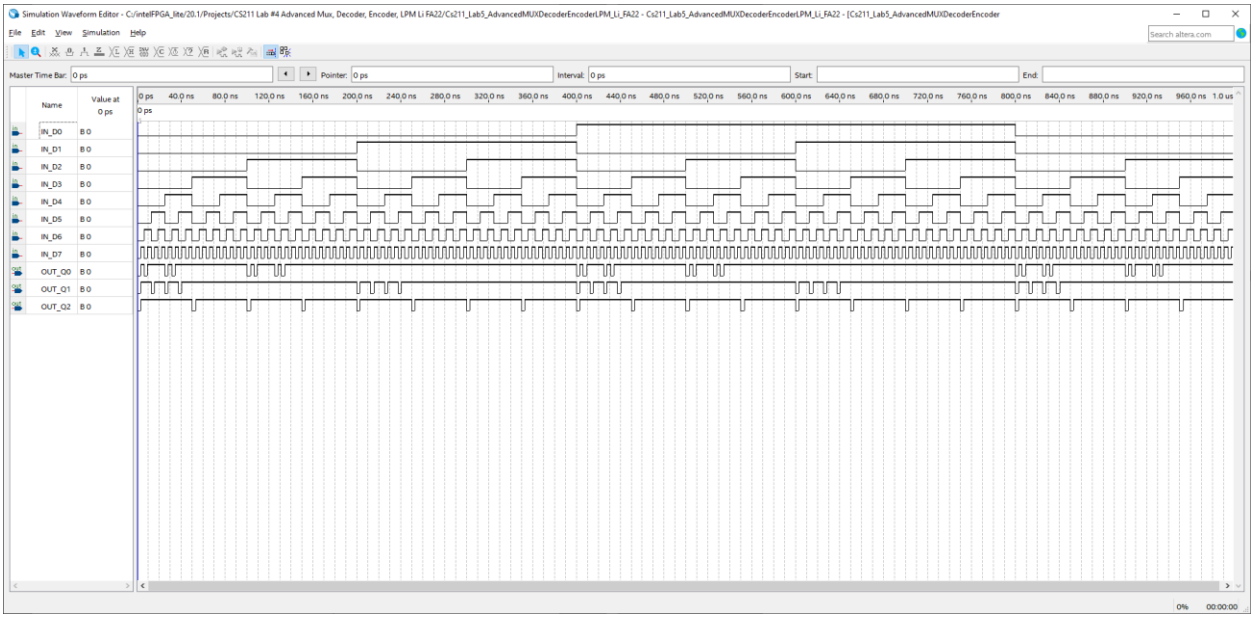

Figure 17: The figure above is a screenshot of the vector waveform output file of the 8 to 3 Encoder.

```
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 Oct 23 00:50:08 2022"
19
20 LIBRARY ieee;
21 USE ieee.std_logic_1164.all;
22
23 LIBRARY work;
24
25 ENTITY Simple8to3EncoderVHDL IS
26    PORT
27    (
28       Y0 :  IN  STD_LOGIC;
29       Y1 :  IN  STD_LOGIC;
30       Y2 :  IN  STD_LOGIC;
31       Y3 :  IN  STD_LOGIC;
32       Y4 :  IN  STD_LOGIC;
33       Y5 :  IN  STD_LOGIC;
34       Y6 :  IN  STD_LOGIC;
35       Y7 :  IN  STD_LOGIC;
36       A0 :  OUT  STD_LOGIC;
37       A1 :  OUT  STD_LOGIC;
38       A2 :  OUT  STD_LOGIC
39    );
40 END Simple8to3EncoderVHDL;
41
42 ARCHITECTURE bdf_type OF Simple8to3EncoderVHDL IS
43
44
45
46 BEGIN
47
48
49
50 A0 <= Y1 OR Y5 OR Y7 OR Y3;
51
52
53 A1 <= Y2 OR Y6 OR Y7 OR Y3;
54
55
56 A2 <= Y4 OR Y6 OR Y7 OR Y5;
57
58
59 END bdf_type;
```

Figure 18: The figure above is a screenshot of the VHDL code of the 8 to 3 Encoder as requested in the lab exercise.

- Part II.4 (Demultiplexer)

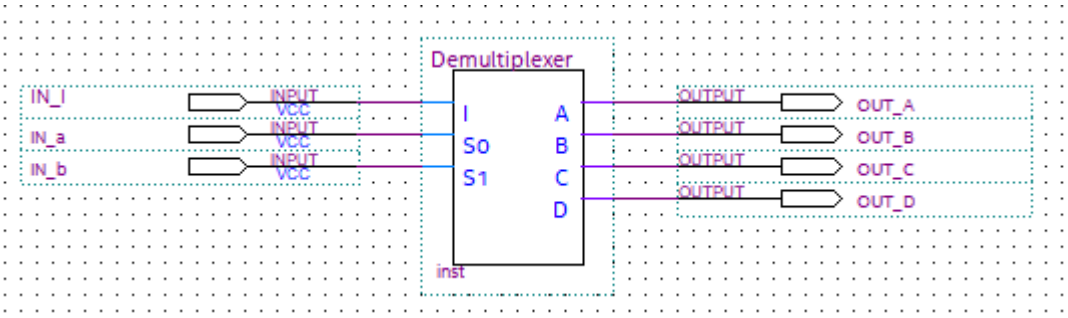

Figure 19: The figure above is a screenshot of the Demultiplexer.

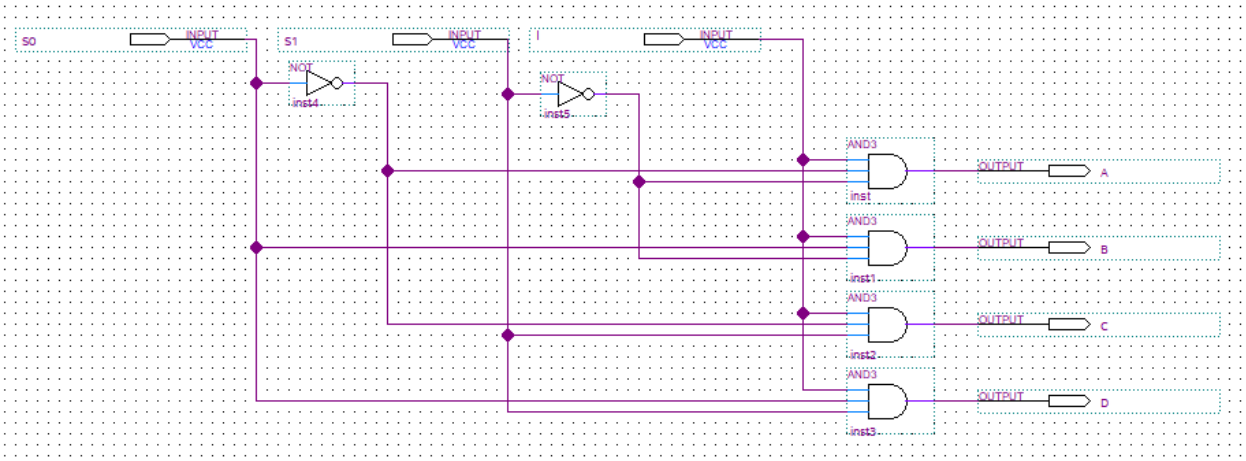

Figure 20: The figure above is a screenshot of the circuit diagram of Demultiplexer component used in the Demultiplexer.

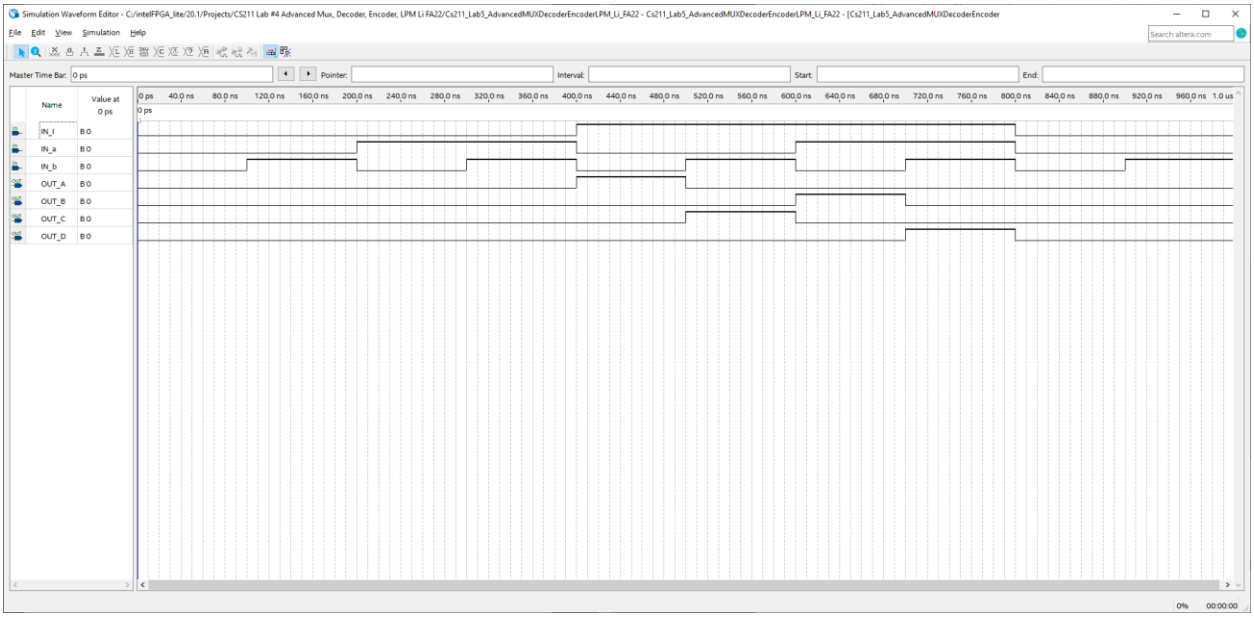

Figure 21: The figure above is a screenshot of the vector waveform output file of the Demultiplexer.

```
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 Oct 23 01:05:10 2022"
19
20 LIBRARY ieee;
21 USE ieee.std_logic_1164.all;
22
23 LIBRARY work;
24
25 ENTITY DemultiplexerVHDL IS
26    PORT
27    (
28       I :  IN  STD_LOGIC;
29       S0 :  IN  STD_LOGIC;
30       S1 :  IN  STD_LOGIC;
31       A :  OUT  STD_LOGIC;
32       B :  OUT  STD_LOGIC;
33       C :  OUT  STD_LOGIC;
34       D :  OUT  STD_LOGIC
35    );
36 END DemultiplexerVHDL;
37
38 ARCHITECTURE bdf_type OF DemultiplexerVHDL IS
39
40 SIGNAL   SYNTHESIZED_WIRE_4 :  STD_LOGIC;
41 SIGNAL   SYNTHESIZED_WIRE_5 :  STD_LOGIC;
42
43
44 BEGIN
45
46
47
48 A <= I AND SYNTHESIZED_WIRE_4 AND SYNTHESIZED_WIRE_5;
49
50
51 B <= I AND S0 AND SYNTHESIZED_WIRE_5;
52
53
54 C <= I AND SYNTHESIZED_WIRE_4 AND S1;
55
56
57 D <= I AND S0 AND S1;
58
59
60 SYNTHESIZED_WIRE_4 <= NOT(S0);
61
62
63
64 SYNTHESIZED_WIRE_5 <= NOT(S1);
65
66
67
68 END bdf_type;
```

Figure 22: The figure above is a screenshot of the VHDL code of the Demultiplexer as requested in the lab exercise.

Conclusions:

- I learned how the 5:1 Mux, 2:4 Decoder. 3:8 Decoder, 8:3 Encoder, and Demultiplexer operates and their logic diagrams along with their truth tables. In addition to that, I learned how to create VHDL code from an already existing block diagram and also how to convert VHDL code into a symbol that can be used in a new project.