

The City College of New York
Computer Organization Lab
CSC 34300
(Spring 2024)

LAB 2

Mux, Decoder, Encoder Basic

Mahir Asef

Instructor: Albi Arapi

Date: 03.06.2024

Table of Contents

Objective -----	Page 3
Functionality and Specifications-----	Page 3
Simulations-----	Page 36
Conclusion-----	Page 39

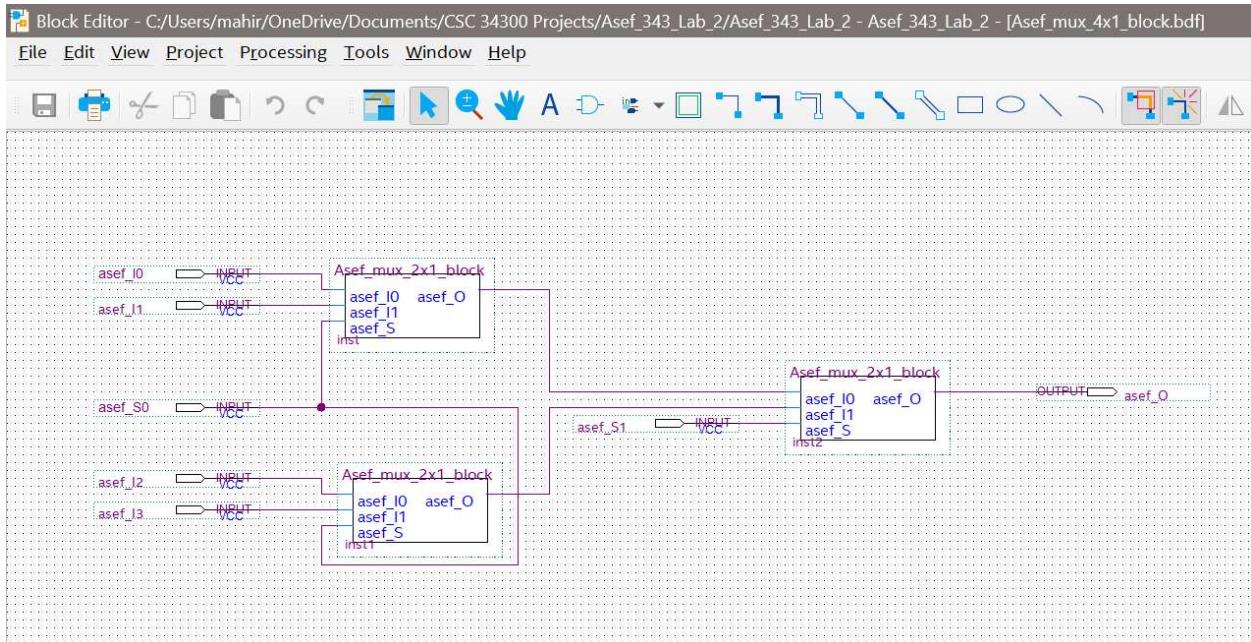
Objective:

The purpose of this lab is to experiment and design a more complicated Multiplexer by building on the idea of a 2:1 Multiplexer. In this lab, we will be recreating 2:1 Mux with 8 bits input and output as well as creating 4:1 Mux and 5:1 Mux which takes 4 and 5 dependent inputs and 2 and 3 control inputs respectively. This lab will also shift us to give us an introduction to Encoder, Decoder and Demultiplexer. We will be able to design a 3:8 Decoder, an 8:3 priority Encoder and 1:2 Demultiplexer by the end of this experiment. For each of these logic designs, we will create a schematic/block diagram, create individual symbols, write truth table, use the truth table to write our own VHDL code and compare it to the compiler generated VHDL code. We will finally create testbench VHDL code to verify all our designs by simulating them on Modelsim.

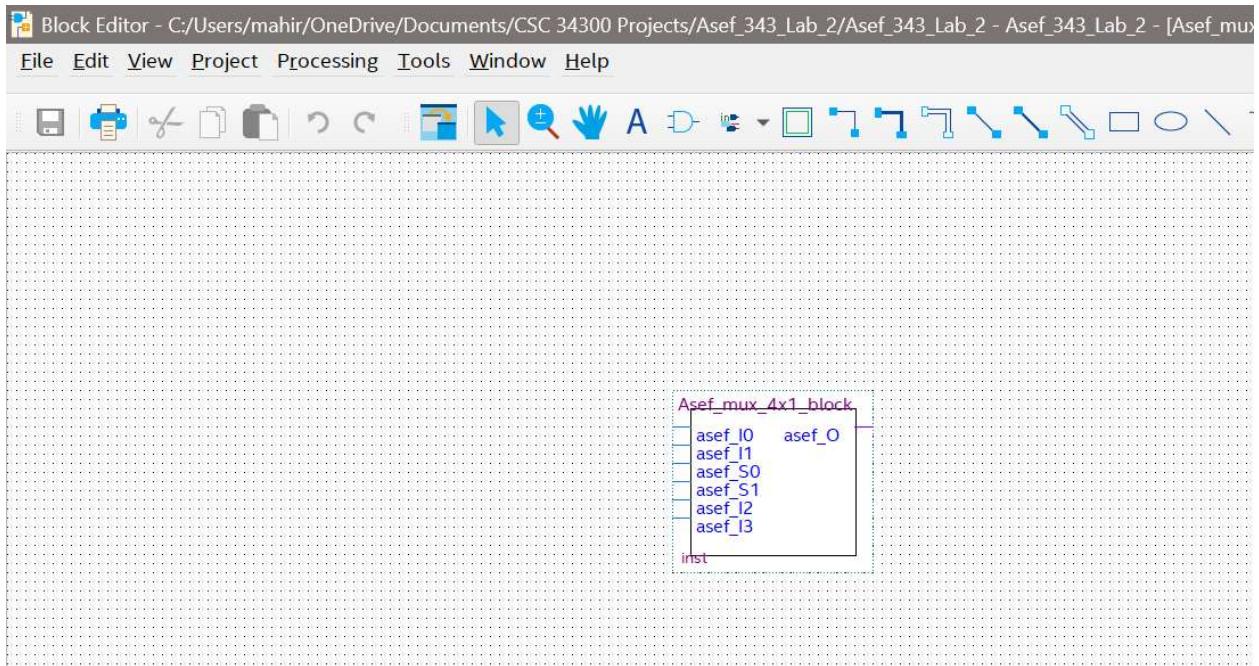
Specification and Functionality:

A. Mux 4:1

Schematic:



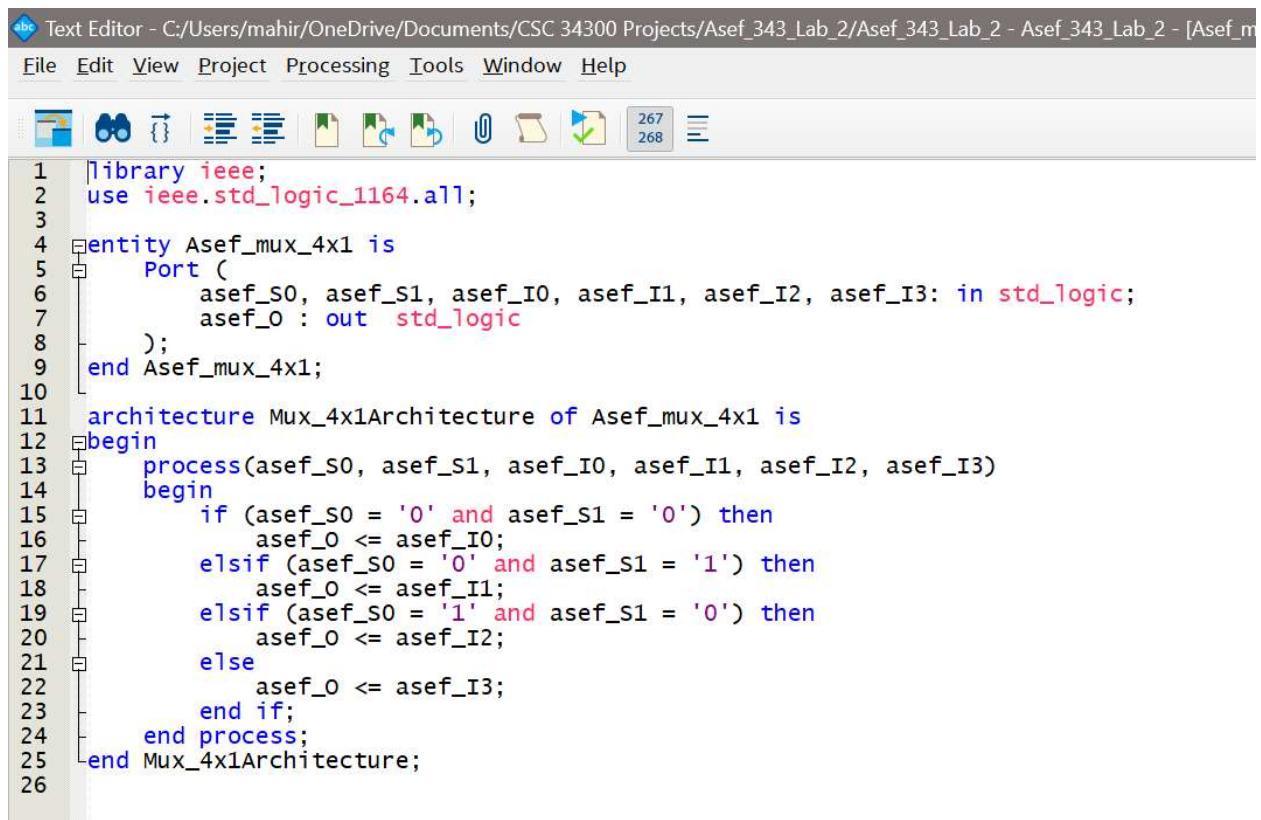
Symbol:



Truth Table:

Input		Output
S1	S0	O
0	0	I0
0	1	I1
1	0	I2
1	1	I3

VHDL Code:



The screenshot shows a text editor window with the title "Text Editor - C:/Users/mahir/OneDrive/Documents/CSC 34300 Projects/Asef_343_Lab_2/Asef_343_Lab_2 - Asef_343_Lab_2 - [Asef_m].vhd". The menu bar includes File, Edit, View, Project, Processing, Tools, Window, and Help. The toolbar has icons for file operations like Open, Save, Copy, Paste, and Find. The status bar shows "267" and "268". The code is a VHDL entity and architecture for a 4x1 multiplexer:

```
library ieee;
use ieee.std_logic_1164.all;

entity Asef_mux_4x1 is
    Port (
        asef_S0, asef_S1, asef_I0, asef_I1, asef_I2, asef_I3: in std_logic;
        asef_O : out std_logic
    );
end Asef_mux_4x1;

architecture Mux_4x1Architecture of Asef_mux_4x1 is
begin
    process(asef_S0, asef_S1, asef_I0, asef_I1, asef_I2, asef_I3)
    begin
        if (asef_S0 = '0' and asef_S1 = '0') then
            asef_O <= asef_I0;
        elsif (asef_S0 = '0' and asef_S1 = '1') then
            asef_O <= asef_I1;
        elsif (asef_S0 = '1' and asef_S1 = '0') then
            asef_O <= asef_I2;
        else
            asef_O <= asef_I3;
        end if;
    end process;
end Mux_4x1Architecture;
```

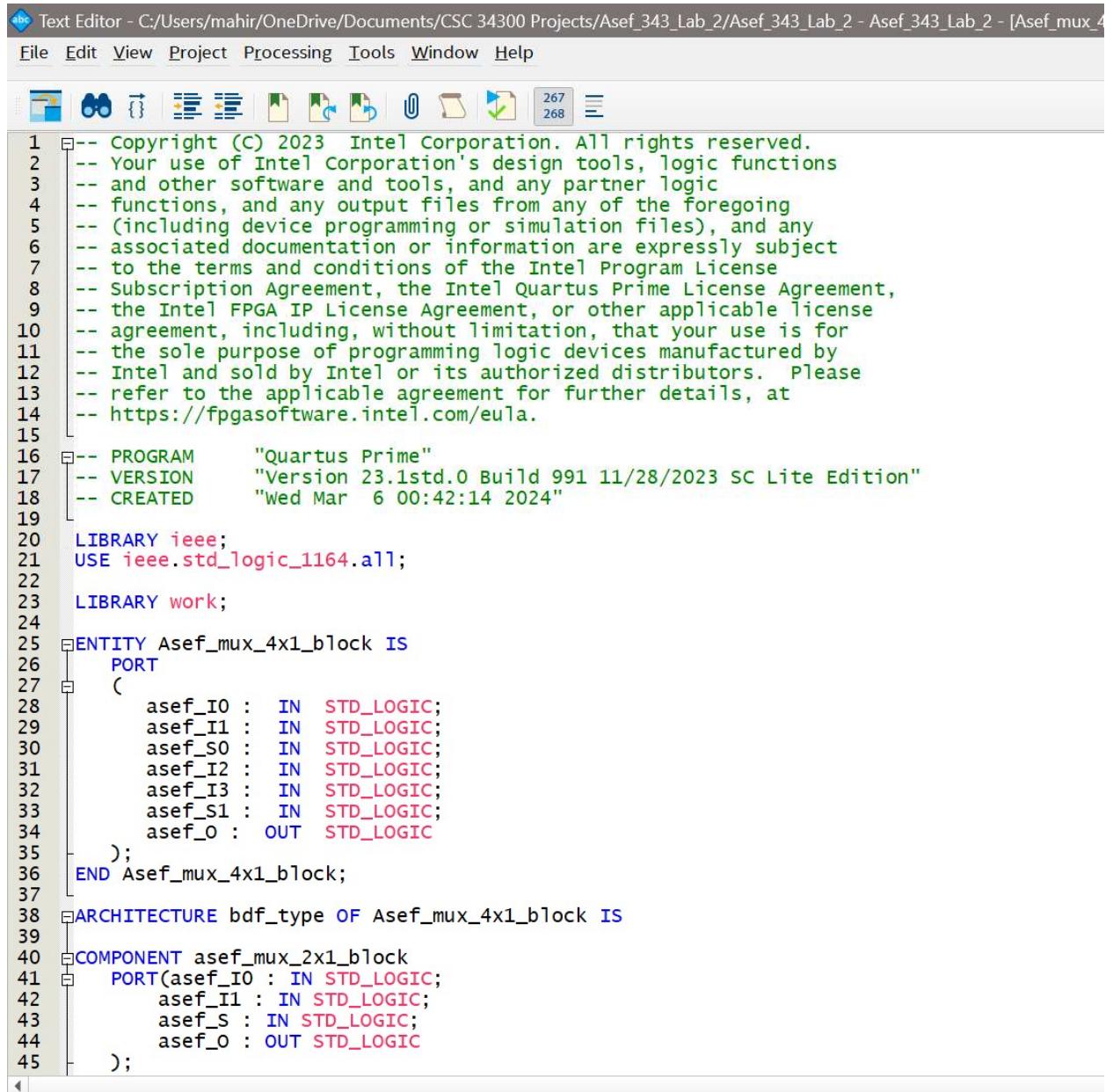
Testbench Code:

```

1 library ieee;
2 use ieee.std_logic_1164.all;
3
4 entity Asef_mux_4x1_TB is
5 end entity Asef_mux_4x1_TB;
6
7 architecture Mux_4x1_Architecture of Asef_mux_4x1_TB is
8 -- Component declaration
9 component Asef_mux_4x1
10 Port (
11     asef_S0, asef_S1, asef_I0, asef_I1, asef_I2, asef_I3: in std_logic;
12     asef_O : out std_logic
13 );
14 end component Asef_mux_4x1;
15
16 -- Signals for testbench
17 signal Asef_S0, Asef_S1, Asef_in0, Asef_in1, Asef_in2, Asef_in3, Asef_out : std_logic;
18
19 begin
20 DUT : Asef_mux_4x1
21 port map (
22     asef_S0 => Asef_S0,
23     asef_S1 => Asef_S1,
24     asef_I0 => Asef_in0,
25     asef_I1 => Asef_in1,
26     asef_I2 => Asef_in2,
27     asef_I3 => Asef_in3,
28     asef_O => Asef_out
29 );
30
31 stimulus: process
32 begin
33     Asef_S0 <= '0'; Asef_S1 <= '0';
34     Asef_in0 <= '0'; Asef_in1 <= '1';
35     Asef_in2 <= '0'; Asef_in3 <= '1';
36     wait for 20 ns;
37     assert Asef_out = '0' report "Test case 1 failed" severity error;
38
39     Asef_S0 <= '0'; Asef_S1 <= '1';
40     Asef_in0 <= '1'; Asef_in1 <= '1';
41     Asef_in2 <= '0'; Asef_in3 <= '0';
42     wait for 20 ns;
43     assert Asef_out = '1' report "Test case 2 failed" severity error;
44
45     Asef_S0 <= '1'; Asef_S1 <= '0';
46     Asef_in0 <= '1'; Asef_in1 <= '0';
47     Asef_in2 <= '1'; Asef_in3 <= '1';
48     wait for 20 ns;
49     assert Asef_out = '1' report "Test case 3 failed" severity error;
50
51     Asef_S0 <= '1'; Asef_S1 <= '1';
52     Asef_in0 <= '0'; Asef_in1 <= '1';
53     Asef_in2 <= '1'; Asef_in3 <= '0';
54     wait for 20 ns;
55     assert Asef_out = '0' report "Test case 4 failed" severity error;
56
57
58 end process stimulus;
59 end architecture Mux_4x1_Architecture;
60

```

Generated VHDL Code:



The screenshot shows a text editor window with the title "Text Editor - C:/Users/mahir/OneDrive/Documents/CSC 34300 Projects/Asef_343_Lab_2/Asef_343_Lab_2 - Asef_343_Lab_2 - [Asef_mux_4x1_block.vhd]". The menu bar includes File, Edit, View, Project, Processing, Tools, Window, and Help. The toolbar contains icons for file operations like Open, Save, Copy, Paste, and Undo. The status bar shows "267" and "268". The code itself is a VHDL entity and architecture definition:

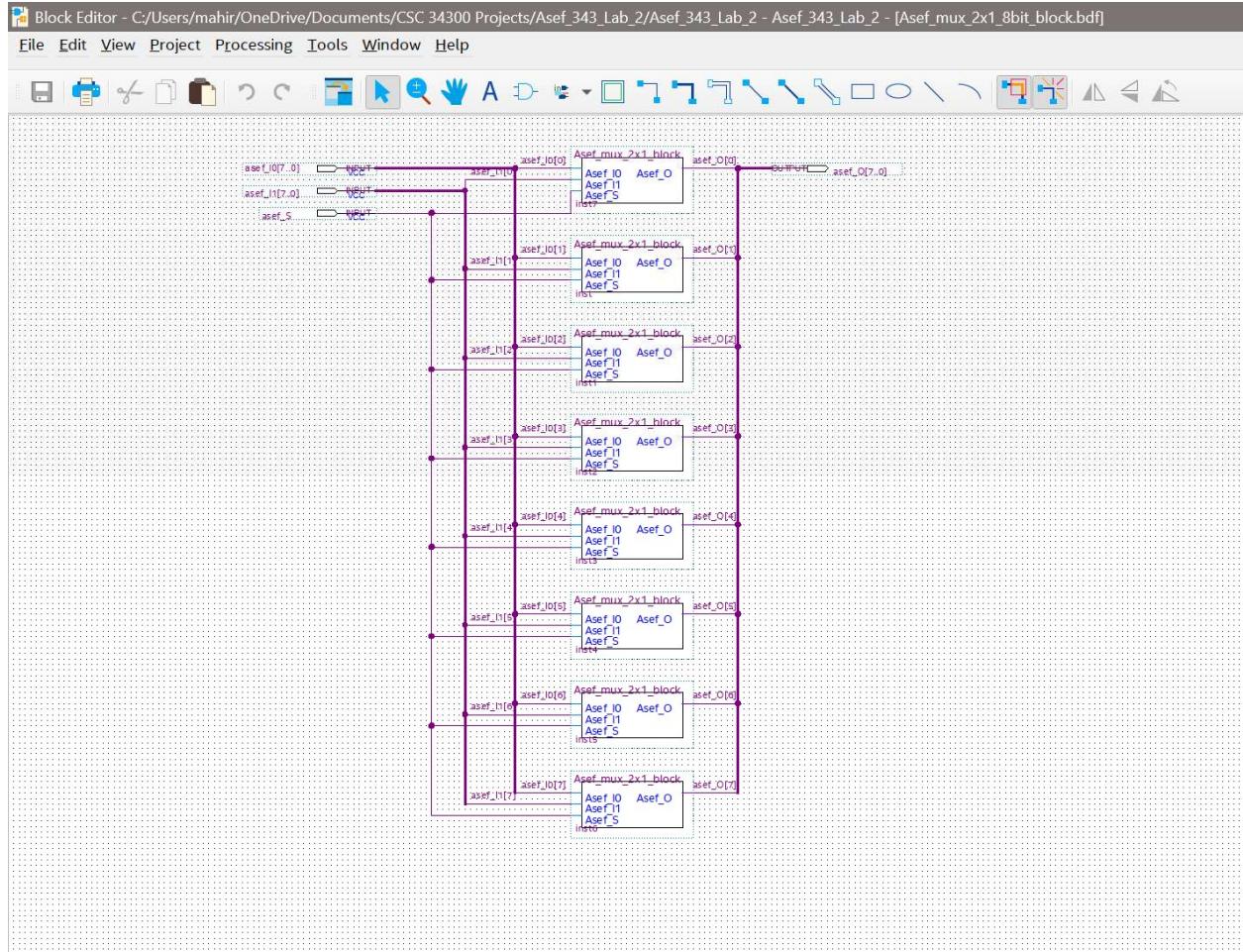
```
1  -- Copyright (C) 2023 Intel Corporation. All rights reserved.
2  -- Your use of Intel Corporation's design tools, logic functions
3  -- and other software and tools, and any partner logic
4  -- functions, and any output files from any of the foregoing
5  -- (including device programming or simulation 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 23.1std.0 Build 991 11/28/2023 SC Lite Edition"
18 -- CREATED      "Wed Mar  6 00:42:14 2024"
19
20 LIBRARY ieee;
21 USE ieee.std_logic_1164.all;
22
23 LIBRARY work;
24
25 ENTITY Asef_mux_4x1_block IS
26   PORT
27   (
28     asef_I0 : IN STD_LOGIC;
29     asef_I1 : IN STD_LOGIC;
30     asef_S0 : IN STD_LOGIC;
31     asef_I2 : IN STD_LOGIC;
32     asef_I3 : IN STD_LOGIC;
33     asef_S1 : IN STD_LOGIC;
34     asef_O : OUT STD_LOGIC
35   );
36 END Asef_mux_4x1_block;
37
38 ARCHITECTURE bdf_type OF Asef_mux_4x1_block IS
39
40 COMPONENT asef_mux_2x1_block
41   PORT(asef_I0 : IN STD_LOGIC;
42         asef_I1 : IN STD_LOGIC;
43         asef_S : IN STD_LOGIC;
44         asef_O : OUT STD_LOGIC
45   );

```

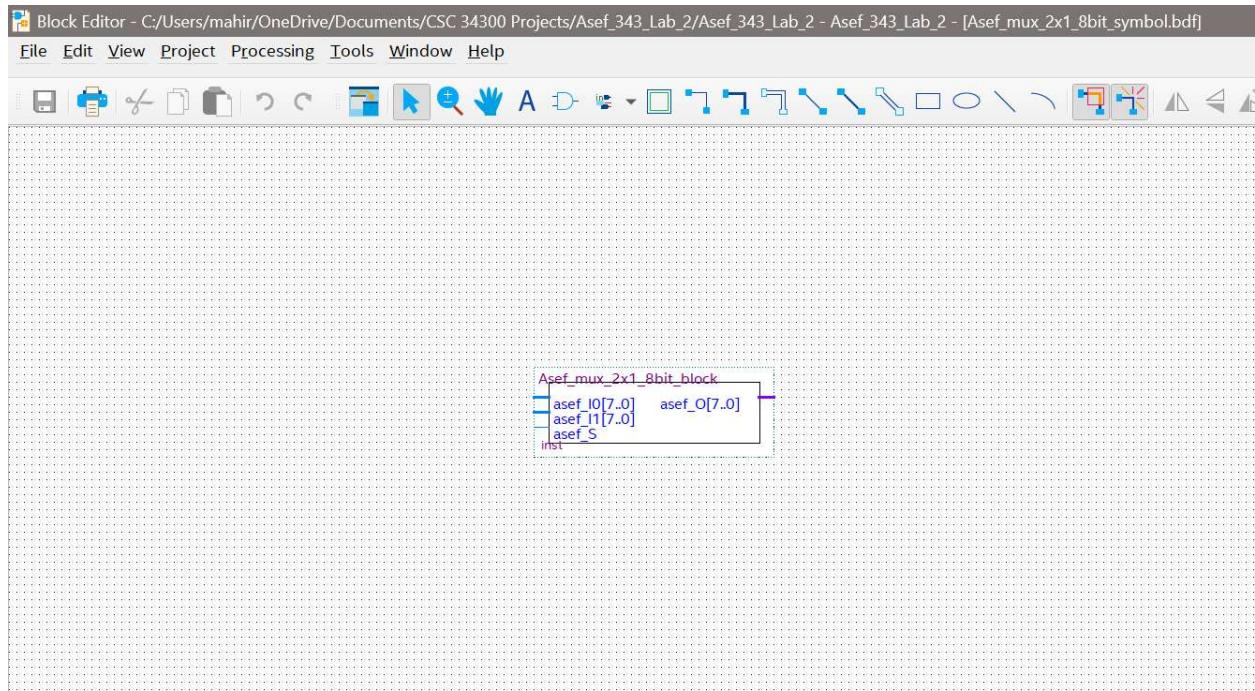
```
46      ,
47  END COMPONENT;
48
49  SIGNAL  SYNTHESIZED_WIRE_0 : STD_LOGIC;
50  SIGNAL  SYNTHESIZED_WIRE_1 : STD_LOGIC;
51
52 BEGIN
53
54
55  b2v_inst : asef_mux_2x1_block
56  PORT MAP(asef_I0 => asef_I0,
57            asef_I1 => asef_I1,
58            asef_S => asef_S0,
59            asef_O => SYNTHESIZED_WIRE_0);
60
61
62  b2v_inst1 : asef_mux_2x1_block
63  PORT MAP(asef_I0 => asef_I2,
64            asef_I1 => asef_I3,
65            asef_S => asef_S0,
66            asef_O => SYNTHESIZED_WIRE_1);
67
68
69  b2v_inst2 : asef_mux_2x1_block
70  PORT MAP(asef_I0 => SYNTHESIZED_WIRE_0,
71            asef_I1 => SYNTHESIZED_WIRE_1,
72            asef_S => asef_S1,
73            asef_O => asef_O);
74
75
76
77 END bdf_type;
```

B. Mux 2:1 8 Bits Input and Output

Schematic:



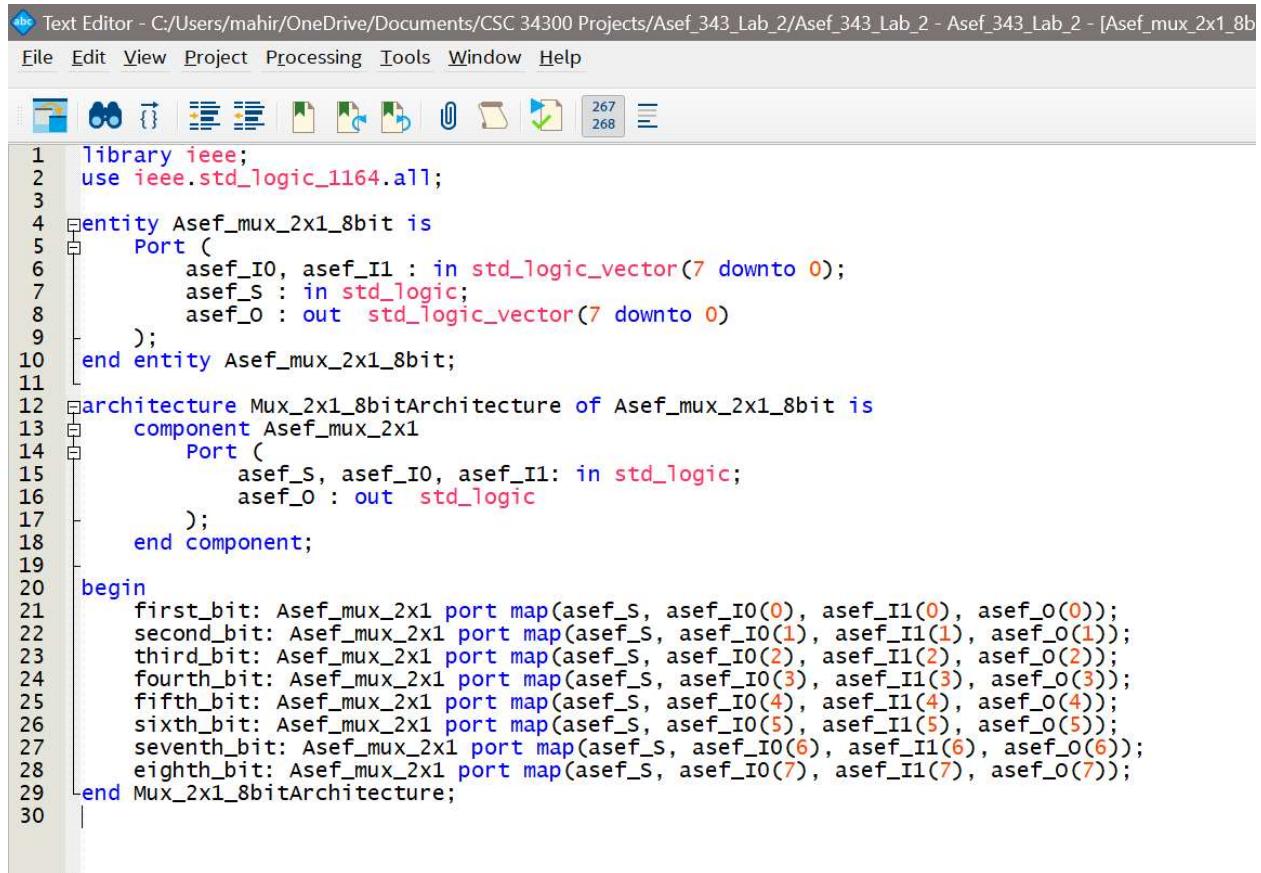
Symbol:



Truth Table:

Input		Output	
S	I0	I1	O
00000000	00000000	X	00000000
00000000	00000000	X	00000001
00000001	X	00000000	00000000
00000001	X	00000001	00000001

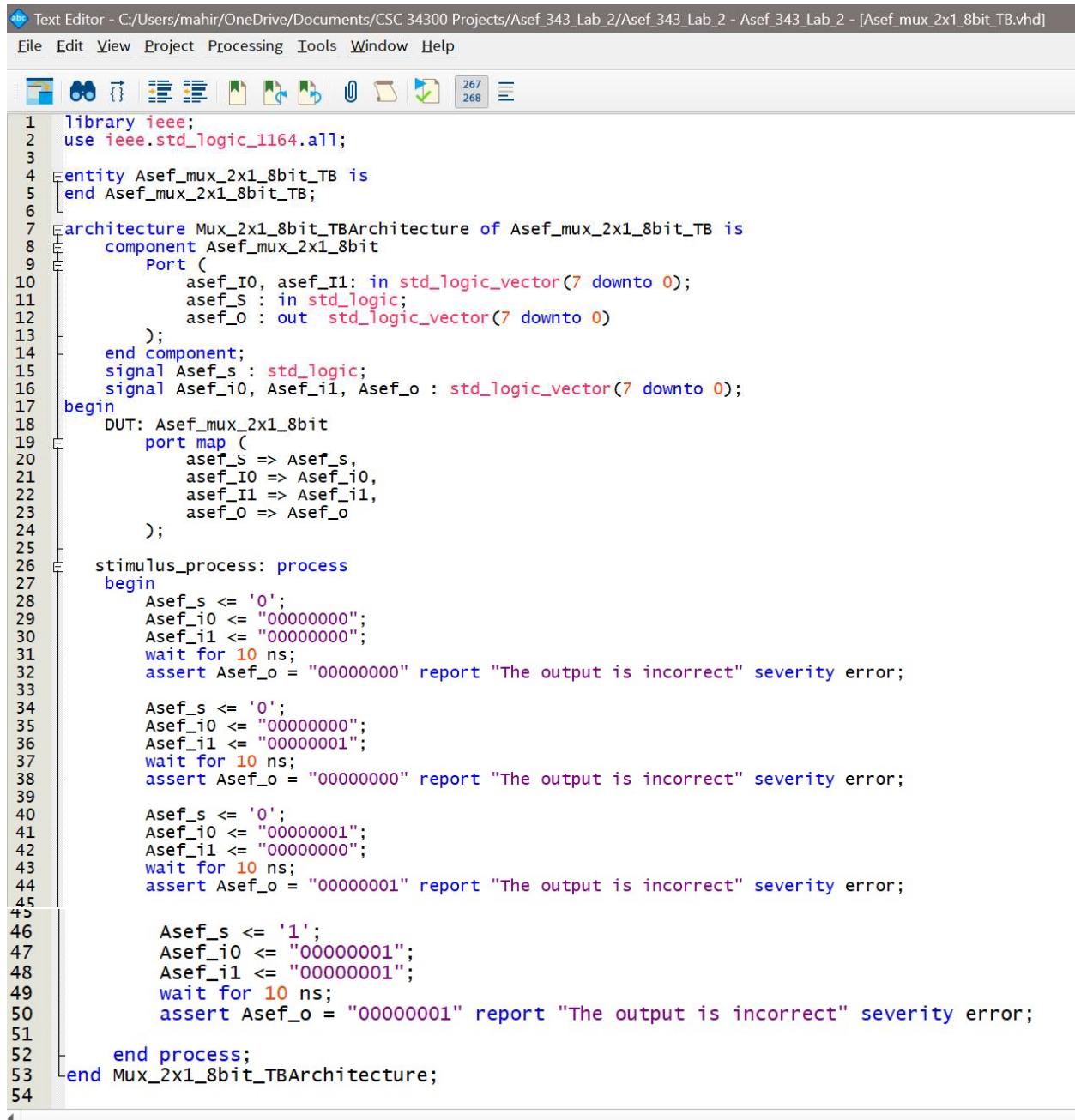
VHDL Code:



The screenshot shows a text editor window with the title "Text Editor - C:/Users/mahir/OneDrive/Documents/CSC 34300 Projects/Asef_343_Lab_2/Asef_343_Lab_2 - Asef_343_Lab_2 - [Asef_mux_2x1_8bit.vhd]". The menu bar includes File, Edit, View, Project, Processing, Tools, Window, and Help. The toolbar has icons for file operations like Open, Save, and Print. The status bar shows "267" and "268". The code itself is a VHDL entity and architecture definition:

```
1 Library ieee;
2 use ieee.std_logic_1164.all;
3
4 entity Asef_mux_2x1_8bit is
5     Port (
6         asef_I0, asef_I1 : in std_logic_vector(7 downto 0);
7         asef_S : in std_logic;
8         asef_O : out std_logic_vector(7 downto 0)
9     );
10 end entity Asef_mux_2x1_8bit;
11
12 architecture Mux_2x1_8bitArchitecture of Asef_mux_2x1_8bit is
13     component Asef_mux_2x1
14         Port (
15             asef_S, asef_I0, asef_I1: in std_logic;
16             asef_O : out std_logic
17         );
18     end component;
19
20 begin
21     first_bit: Asef_mux_2x1 port map(asef_S, asef_I0(0), asef_I1(0), asef_O(0));
22     second_bit: Asef_mux_2x1 port map(asef_S, asef_I0(1), asef_I1(1), asef_O(1));
23     third_bit: Asef_mux_2x1 port map(asef_S, asef_I0(2), asef_I1(2), asef_O(2));
24     fourth_bit: Asef_mux_2x1 port map(asef_S, asef_I0(3), asef_I1(3), asef_O(3));
25     fifth_bit: Asef_mux_2x1 port map(asef_S, asef_I0(4), asef_I1(4), asef_O(4));
26     sixth_bit: Asef_mux_2x1 port map(asef_S, asef_I0(5), asef_I1(5), asef_O(5));
27     seventh_bit: Asef_mux_2x1 port map(asef_S, asef_I0(6), asef_I1(6), asef_O(6));
28     eighth_bit: Asef_mux_2x1 port map(asef_S, asef_I0(7), asef_I1(7), asef_O(7));
29 end Mux_2x1_8bitArchitecture;
30
```

Testbench Code:



The screenshot shows a VHDL text editor interface with a menu bar (File, Edit, View, Project, Processing, Tools, Window, Help) and a toolbar with various icons. The main window displays the VHDL code for a test bench. The code defines an entity Asef_mux_2x1_8bit_TB with an architecture Mux_2x1_8bit_TBArchitecture. The architecture contains a component instantiation of Asef_mux_2x1_8bit with its port map, and a stimulus process that asserts the output is correct for various input conditions.

```

1 library ieee;
2 use ieee.std_logic_1164.all;
3
4 entity Asef_mux_2x1_8bit_TB is
5 end Asef_mux_2x1_8bit_TB;
6
7 architecture Mux_2x1_8bit_TBArchitecture of Asef_mux_2x1_8bit_TB is
8   component Asef_mux_2x1_8bit
9     Port (
10       asef_I0, asef_I1: in std_logic_vector(7 downto 0);
11       asef_S : in std_logic;
12       asef_O : out std_logic_vector(7 downto 0)
13     );
14   end component;
15   signal Asef_s : std_logic;
16   signal Asef_i0, Asef_i1, Asef_o : std_logic_vector(7 downto 0);
17 begin
18   DUT: Asef_mux_2x1_8bit
19     port map (
20       asef_S => Asef_s,
21       asef_I0 => Asef_i0,
22       asef_I1 => Asef_i1,
23       asef_O => Asef_o
24     );
25
26   stimulus_process: process
27   begin
28     Asef_s <= '0';
29     Asef_i0 <= "00000000";
30     Asef_i1 <= "00000000";
31     wait for 10 ns;
32     assert Asef_o = "00000000" report "The output is incorrect" severity error;
33
34     Asef_s <= '0';
35     Asef_i0 <= "00000000";
36     Asef_i1 <= "00000001";
37     wait for 10 ns;
38     assert Asef_o = "00000000" report "The output is incorrect" severity error;
39
40     Asef_s <= '0';
41     Asef_i0 <= "00000001";
42     Asef_i1 <= "00000000";
43     wait for 10 ns;
44     assert Asef_o = "00000001" report "The output is incorrect" severity error;
45
46     Asef_s <= '1';
47     Asef_i0 <= "00000001";
48     Asef_i1 <= "00000001";
49     wait for 10 ns;
50     assert Asef_o = "00000001" report "The output is incorrect" severity error;
51
52   end process;
53 end Mux_2x1_8bit_TBArchitecture;
54

```

Generated VHDL Code:

Text Editor - C:/Users/mahir/OneDrive/Documents/CSC 34300 Projects/Asef_343_Lab_2/Asef_343_Lab_2 - Asef_343_Lab_2 - [Asef]

File Edit View Project Processing Tools Window Help

267
268

```
1 -- Copyright (C) 2023 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 23.1std.0 Build 991 11/28/2023 SC Lite Edition"
18 -- CREATED      "Wed Mar 6 00:23:55 2024"
19
20 LIBRARY ieee;
21 USE ieee.std_logic_1164.all;
22
23 LIBRARY work;
24
25 ENTITY Asef_mux_2x1_8bit_block IS
26   PORT
27   (
28     asef_S : IN STD_LOGIC;
29     asef_I0 : IN STD_LOGIC_VECTOR(7 DOWNTO 0);
30     asef_I1 : IN STD_LOGIC_VECTOR(7 DOWNTO 0);
31     asef_O : OUT STD_LOGIC_VECTOR(7 DOWNTO 0)
32   );
33 END Asef_mux_2x1_8bit_block;
34
35 ARCHITECTURE bdf_type OF Asef_mux_2x1_8bit_block IS
36
37 COMPONENT asef_mux_2x1_block
38   PORT(Asef_I0 : IN STD_LOGIC;
39         Asef_I1 : IN STD_LOGIC;
40         Asef_S : IN STD_LOGIC;
41         Asef_O : OUT STD_LOGIC
42   );
43 END COMPONENT;
44
45 SIGNAL asef_O_ALTERA_SYNTHESIZED : STD_LOGIC_VECTOR(7 DOWNTO 0);
```

Text Editor - C:/Users/mahir/OneDrive/Documents/CSC 34300 Projects/Asef_343_Lab_2/Asef_3.vhd

File Edit View Project Processing Tools Window Help

267
268

```
46
47
48 BEGIN
49
50
51
52 b2v_inst : asef_mux_2x1_block
53 PORT MAP(Asef_I0 => asef_I0(1),
54             Asef_I1 => asef_I1(1),
55             Asef_S => asef_S,
56             Asef_O => asef_O_ALTERA_SYNTHESIZED(1));
57
58
59 b2v_inst1 : asef_mux_2x1_block
60 PORT MAP(Asef_I0 => asef_I0(2),
61             Asef_I1 => asef_I1(2),
62             Asef_S => asef_S,
63             Asef_O => asef_O_ALTERA_SYNTHESIZED(2));
64
65
66 b2v_inst2 : asef_mux_2x1_block
67 PORT MAP(Asef_I0 => asef_I0(3),
68             Asef_I1 => asef_I1(3),
69             Asef_S => asef_S,
70             Asef_O => asef_O_ALTERA_SYNTHESIZED(3));
71
72
73 b2v_inst3 : asef_mux_2x1_block
74 PORT MAP(Asef_I0 => asef_I0(4),
75             Asef_I1 => asef_I1(4),
76             Asef_S => asef_S,
77             Asef_O => asef_O_ALTERA_SYNTHESIZED(4));
78
79
80 b2v_inst4 : asef_mux_2x1_block
81 PORT MAP(Asef_I0 => asef_I0(5),
82             Asef_I1 => asef_I1(5),
83             Asef_S => asef_S,
84             Asef_O => asef_O_ALTERA_SYNTHESIZED(5));
85
86
87 b2v_inst5 : asef_mux_2x1_block
88 PORT MAP(Asef_I0 => asef_I0(6),
89             Asef_I1 => asef_I1(6),
90             Asef_S => asef_S,
```

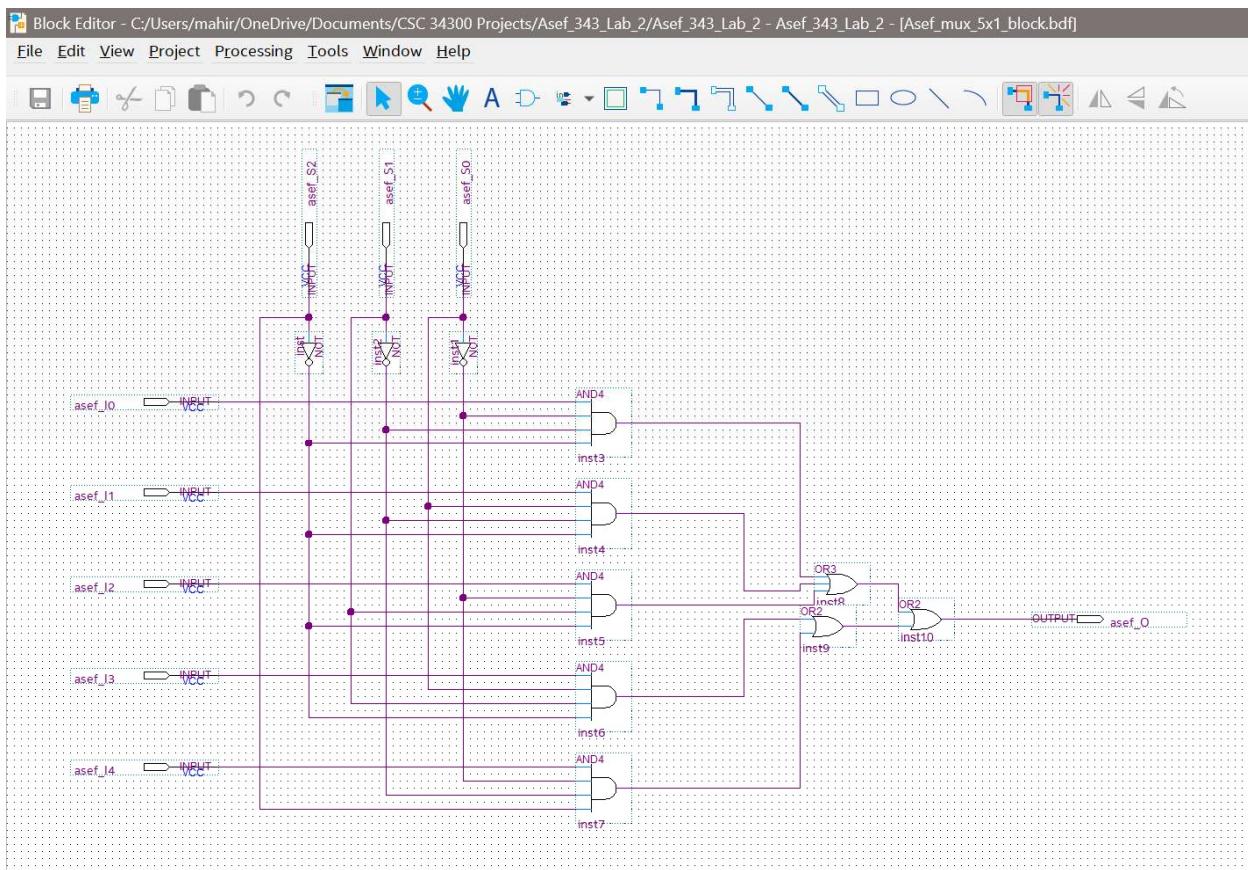
```

91      '~~~' ~~~' ~~~' ~~~',
92      Asef_O => asef_O_ALTERA_SYNTHESIZED(6));
93
94  b2v_inst6 : asef_mux_2x1_block
95  PORT MAP(Asef_I0 => asef_I0(7),
96            Asef_I1 => asef_I1(7),
97            Asef_S => asef_S,
98            Asef_O => asef_O_ALTERA_SYNTHESIZED(7));
99
100 b2v_inst7 : asef_mux_2x1_block
101 PORT MAP(Asef_I0 => asef_I0(0),
102           Asef_I1 => asef_I1(0),
103           Asef_S => asef_S,
104           Asef_O => asef_O_ALTERA_SYNTHESIZED(0));
105
106 asef_O <= asef_O_ALTERA_SYNTHESIZED;
107
108 END bdf_type;

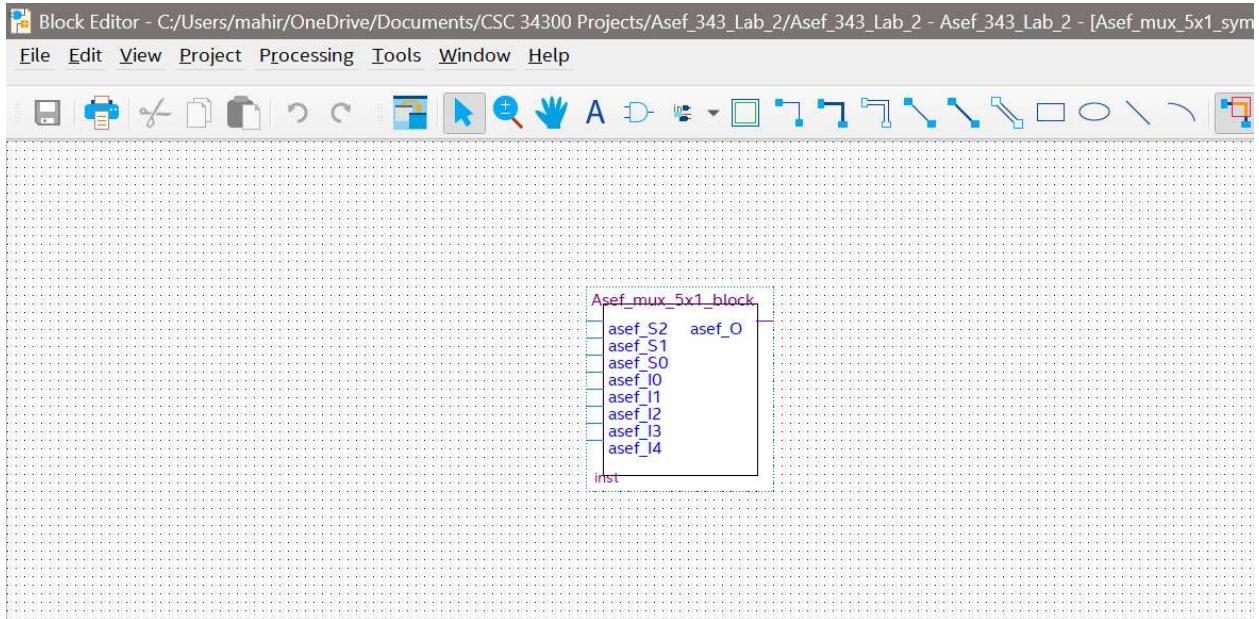
```

C. Mux 5:1

Schematic:



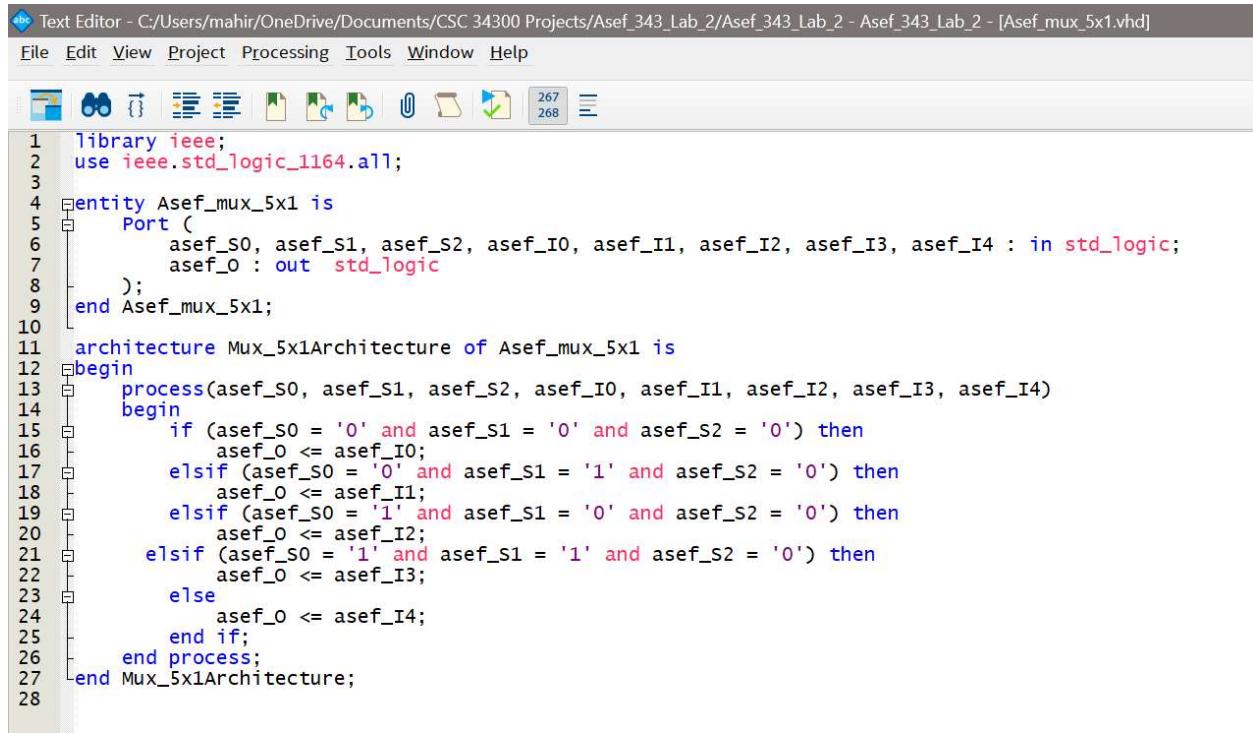
Symbol:



Truth Table:

Input			Output
S2	S1	S0	O
0	0	0	I0
0	0	1	I1
0	1	0	I2
0	1	1	I3
1	0	0	I4
1	0	1	I4
1	1	0	I4
1	1	1	I4

VHDL Code:



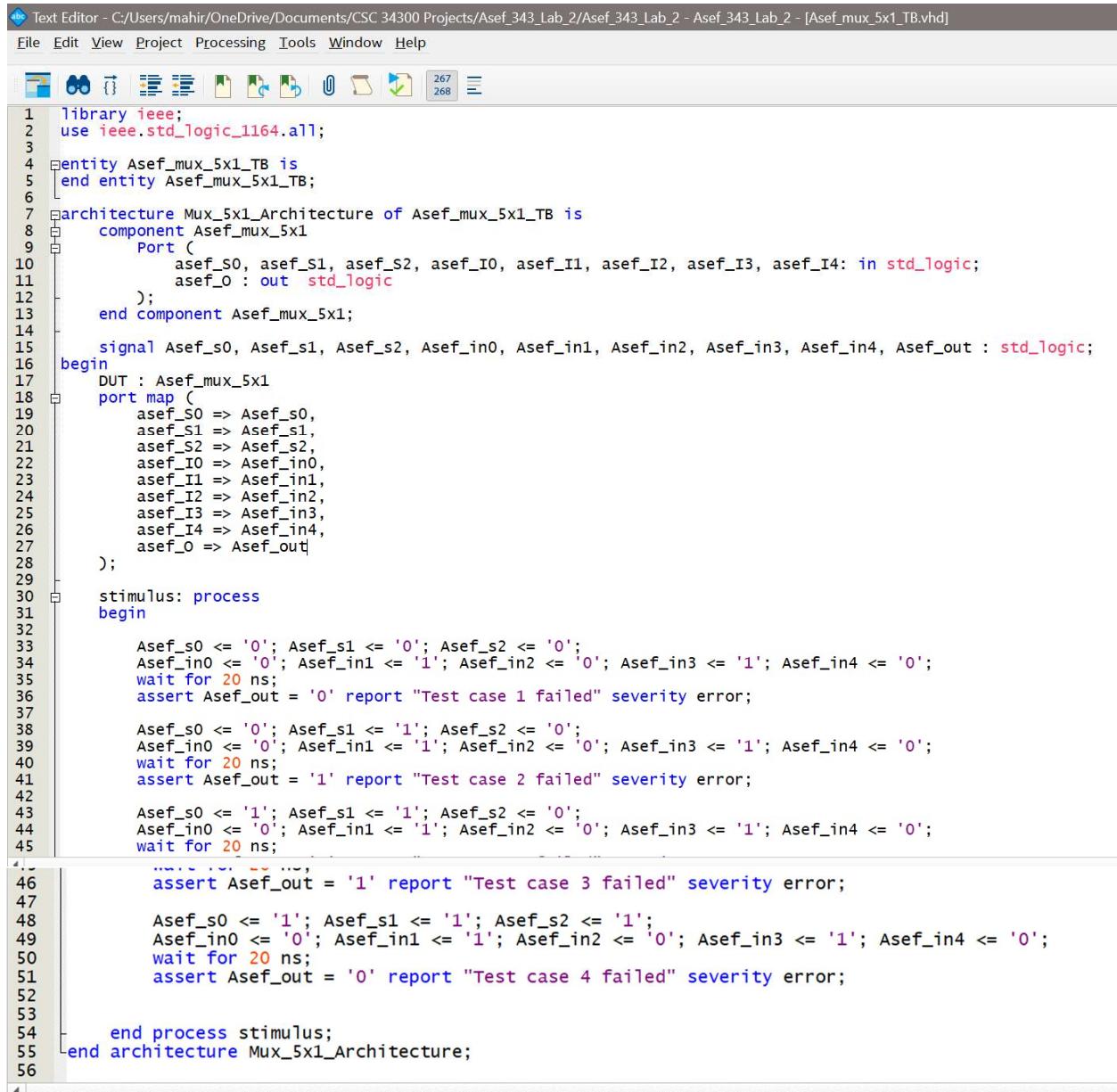
The screenshot shows a text editor window with the title "Text Editor - C:/Users/mahir/OneDrive/Documents/CSC 34300 Projects/Asef_343_Lab_2/Asef_343_Lab_2 - Asef_343_Lab_2 - [Asef_mux_5x1.vhd]". The menu bar includes File, Edit, View, Project, Processing, Tools, Window, and Help. The toolbar has icons for new file, open file, save file, copy, paste, and others. The status bar shows "267" and "268". The code is a VHDL entity and architecture for a 5x1 multiplexer:

```
library ieee;
use ieee.std_logic_1164.all;

entity Asef_mux_5x1 is
    Port (
        asef_S0, asef_S1, asef_S2, asef_I0, asef_I1, asef_I2, asef_I3, asef_I4 : in std_logic;
        asef_O : out std_logic
    );
end Asef_mux_5x1;

architecture Mux_5x1Architecture of Asef_mux_5x1 is
begin
    process(asef_S0, asef_S1, asef_S2, asef_I0, asef_I1, asef_I2, asef_I3, asef_I4)
    begin
        if (asef_S0 = '0' and asef_S1 = '0' and asef_S2 = '0') then
            asef_O <= asef_I0;
        elsif (asef_S0 = '0' and asef_S1 = '1' and asef_S2 = '0') then
            asef_O <= asef_I1;
        elsif (asef_S0 = '1' and asef_S1 = '0' and asef_S2 = '0') then
            asef_O <= asef_I2;
        elsif (asef_S0 = '1' and asef_S1 = '1' and asef_S2 = '0') then
            asef_O <= asef_I3;
        else
            asef_O <= asef_I4;
        end if;
    end process;
end Mux_5x1Architecture;
```

Testbench Code:



The screenshot shows a VHDL text editor interface with a menu bar (File, Edit, View, Project, Processing, Tools, Window, Help) and a toolbar above the code area. The code itself is a testbench for a 5x1 Mux component. It includes declarations for the IEEE library and std_logic_1164 package, the entity Asef_mux_5x1_TB, its architecture, and a stimulus process. The architecture defines a component Asef_mux_5x1 with a port mapping from its internal signals to external pins. The stimulus process contains four test cases, each asserting specific input conditions and checking the output Asef_out against expected values ('0' or '1').

```

1  library ieee;
2  use ieee.std_logic_1164.all;
3
4  entity Asef_mux_5x1_TB is
5  end entity Asef_mux_5x1_TB;
6
7  architecture Mux_5x1_Architecture of Asef_mux_5x1_TB is
8  component Asef_mux_5x1
9  Port (
10    asef_S0, asef_S1, asef_S2, asef_I0, asef_I1, asef_I2, asef_I3, asef_I4: in std_logic;
11    asef_O : out std_logic
12  );
13  end component Asef_mux_5x1;
14
15  signal Asef_S0, Asef_S1, Asef_S2, Asef_in0, Asef_in1, Asef_in2, Asef_in3, Asef_in4, Asef_out : std_logic;
16 begin
17  DUT : Asef_mux_5x1
18  port map (
19    asef_S0 => Asef_S0,
20    asef_S1 => Asef_S1,
21    asef_S2 => Asef_S2,
22    asef_I0 => Asef_in0,
23    asef_I1 => Asef_in1,
24    asef_I2 => Asef_in2,
25    asef_I3 => Asef_in3,
26    asef_I4 => Asef_in4,
27    asef_O => Asef_out
28  );
29
30  stimulus: process
31  begin
32
33    Asef_S0 <= '0'; Asef_S1 <= '0'; Asef_S2 <= '0';
34    Asef_in0 <= '0'; Asef_in1 <= '1'; Asef_in2 <= '0'; Asef_in3 <= '1'; Asef_in4 <= '0';
35    wait for 20 ns;
36    assert Asef_out = '0' report "Test case 1 failed" severity error;
37
38    Asef_S0 <= '0'; Asef_S1 <= '1'; Asef_S2 <= '0';
39    Asef_in0 <= '0'; Asef_in1 <= '1'; Asef_in2 <= '0'; Asef_in3 <= '1'; Asef_in4 <= '0';
40    wait for 20 ns;
41    assert Asef_out = '1' report "Test case 2 failed" severity error;
42
43    Asef_S0 <= '1'; Asef_S1 <= '1'; Asef_S2 <= '0';
44    Asef_in0 <= '0'; Asef_in1 <= '1'; Asef_in2 <= '0'; Asef_in3 <= '1'; Asef_in4 <= '0';
45    wait for 20 ns;
46    assert Asef_out = '1' report "Test case 3 failed" severity error;
47
48    Asef_S0 <= '1'; Asef_S1 <= '1'; Asef_S2 <= '1';
49    Asef_in0 <= '0'; Asef_in1 <= '1'; Asef_in2 <= '0'; Asef_in3 <= '1'; Asef_in4 <= '0';
50    wait for 20 ns;
51    assert Asef_out = '0' report "Test case 4 failed" severity error;
52
53
54  end process stimulus;
55 end architecture Mux_5x1_Architecture;
56

```

Generated VHDL Code:

Text Editor - C:/Users/mahir/OneDrive/Documents/CSC 34300 Projects/Asef_343_Lab_2/Asef_343_Lab_2 - Asef_343_Lab_2 - [

File Edit View Project Processing Tools Window Help

267
268

```
1  -- Copyright (C) 2023 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 23.1std.0 Build 991 11/28/2023 SC Lite Edition"
18 -- CREATED      "Wed Mar  6 00:16:37 2024"
19
20 LIBRARY ieee;
21 USE ieee.std_logic_1164.all;
22
23 LIBRARY work;
24
25 ENTITY Asef_mux_5x1_block IS
26   PORT
27   (
28     asef_I1 :  IN STD_LOGIC;
29     asef_I0 :  IN STD_LOGIC;
30     asef_I2 :  IN STD_LOGIC;
31     asef_I3 :  IN STD_LOGIC;
32     asef_I4 :  IN STD_LOGIC;
33     asef_S2 :  IN STD_LOGIC;
34     asef_S1 :  IN STD_LOGIC;
35     asef_S0 :  IN STD_LOGIC;
36     asef_O :  OUT STD_LOGIC
37   );
38 END Asef_mux_5x1_block;
39
40 ARCHITECTURE bdf_type OF Asef_mux_5x1_block IS
41
42   SIGNAL  SYNTHESIZED_WIRE_0 :  STD_LOGIC;
43   SIGNAL  SYNTHESIZED_WIRE_1 :  STD_LOGIC;
44   SIGNAL  SYNTHESIZED_WIRE_17 :  STD_LOGIC;
45   SIGNAL  SYNTHESIZED_WIRE_18 :  STD_LOGIC;
```

Text Editor - C:/Users/mahir/OneDrive/Documents/CSC 34300 Projects/Asef_343_Lab_2/Asef_343_Lab_2 - [Asef_mux_5x1_block.vhd]

File Edit View Project Processing Tools Window Help

267 268

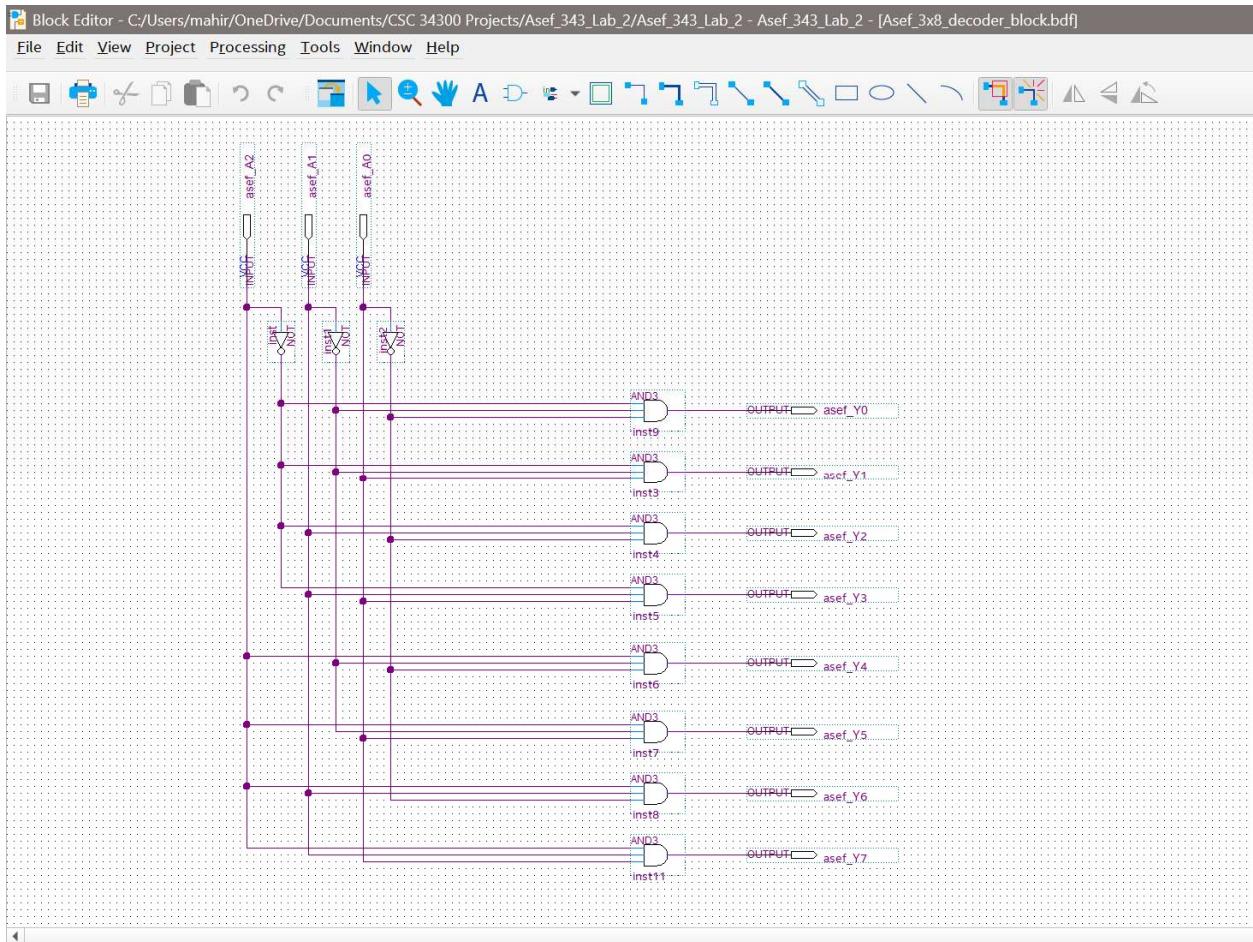
```

46 SIGNAL  SYNTHESIZED_WIRE_19 : STD_LOGIC;
47 SIGNAL  SYNTHESIZED_WIRE_12 : STD_LOGIC;
48 SIGNAL  SYNTHESIZED_WIRE_13 : STD_LOGIC;
49 SIGNAL  SYNTHESIZED_WIRE_14 : STD_LOGIC;
50 SIGNAL  SYNTHESIZED_WIRE_15 : STD_LOGIC;
51 SIGNAL  SYNTHESIZED_WIRE_16 : STD_LOGIC;
52
53
54 BEGIN
55
56
57
58   SYNTHESIZED_WIRE_19 <= NOT(asef_S2);
59
60
61   SYNTHESIZED_WIRE_17 <= NOT(asef_S0);
62
63
64
65   asef_O <= SYNTHESIZED_WIRE_0 OR SYNTHESIZED_WIRE_1;
66
67
68   SYNTHESIZED_WIRE_18 <= NOT(asef_S1);
69
70
71
72   SYNTHESIZED_WIRE_14 <= asef_I0 AND SYNTHESIZED_WIRE_17 AND SYNTHESIZED_WIRE_18 AND SYNTHESIZED_WIRE_19;
73
74
75   SYNTHESIZED_WIRE_12 <= asef_I1 AND asef_S0 AND SYNTHESIZED_WIRE_18 AND SYNTHESIZED_WIRE_19;
76
77
78   SYNTHESIZED_WIRE_13 <= asef_I2 AND SYNTHESIZED_WIRE_17 AND asef_S1 AND SYNTHESIZED_WIRE_19;
79
80
81   SYNTHESIZED_WIRE_16 <= asef_I3 AND asef_S0 AND asef_S1 AND SYNTHESIZED_WIRE_19;
82
83
84   SYNTHESIZED_WIRE_15 <= asef_I4 AND SYNTHESIZED_WIRE_17 AND SYNTHESIZED_WIRE_18 AND asef_S2;
85
86
87   SYNTHESIZED_WIRE_1 <= SYNTHESIZED_WIRE_12 OR SYNTHESIZED_WIRE_13 OR SYNTHESIZED_WIRE_14;
88
89
90
91   SYNTHESIZED_WIRE_0 <= SYNTHESIZED_WIRE_15 OR SYNTHESIZED_WIRE_16;
92
93
94 END bdf_type;

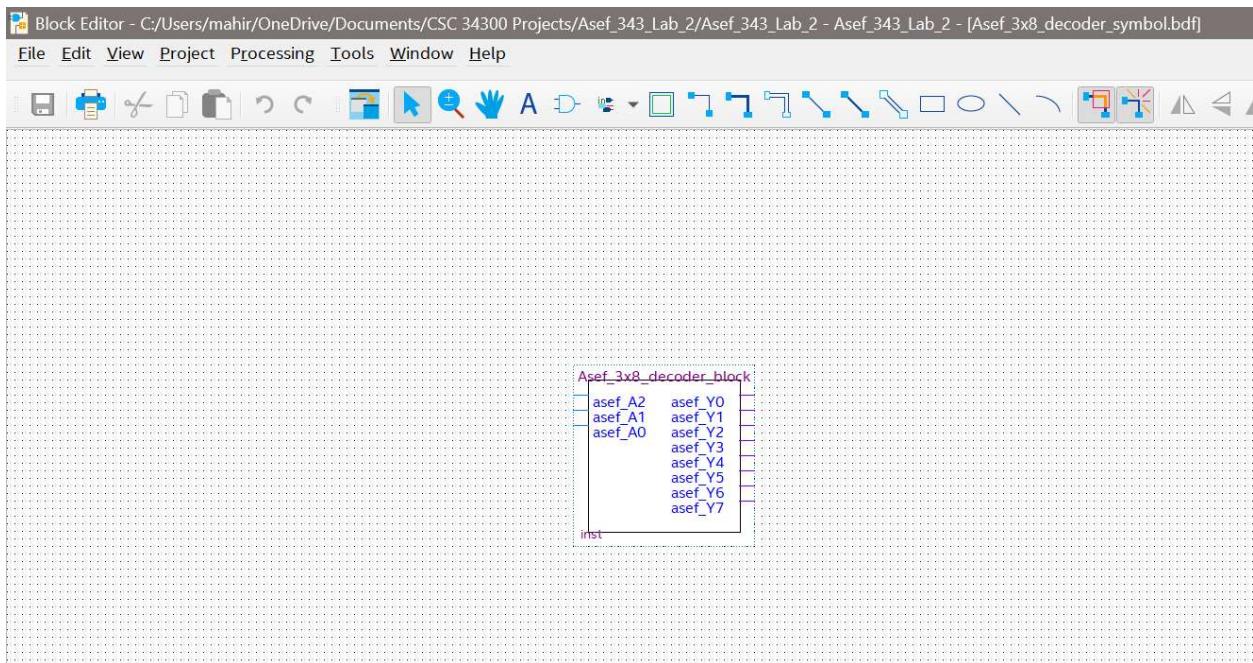
```

D. 3:8 Decoder

Schematic:



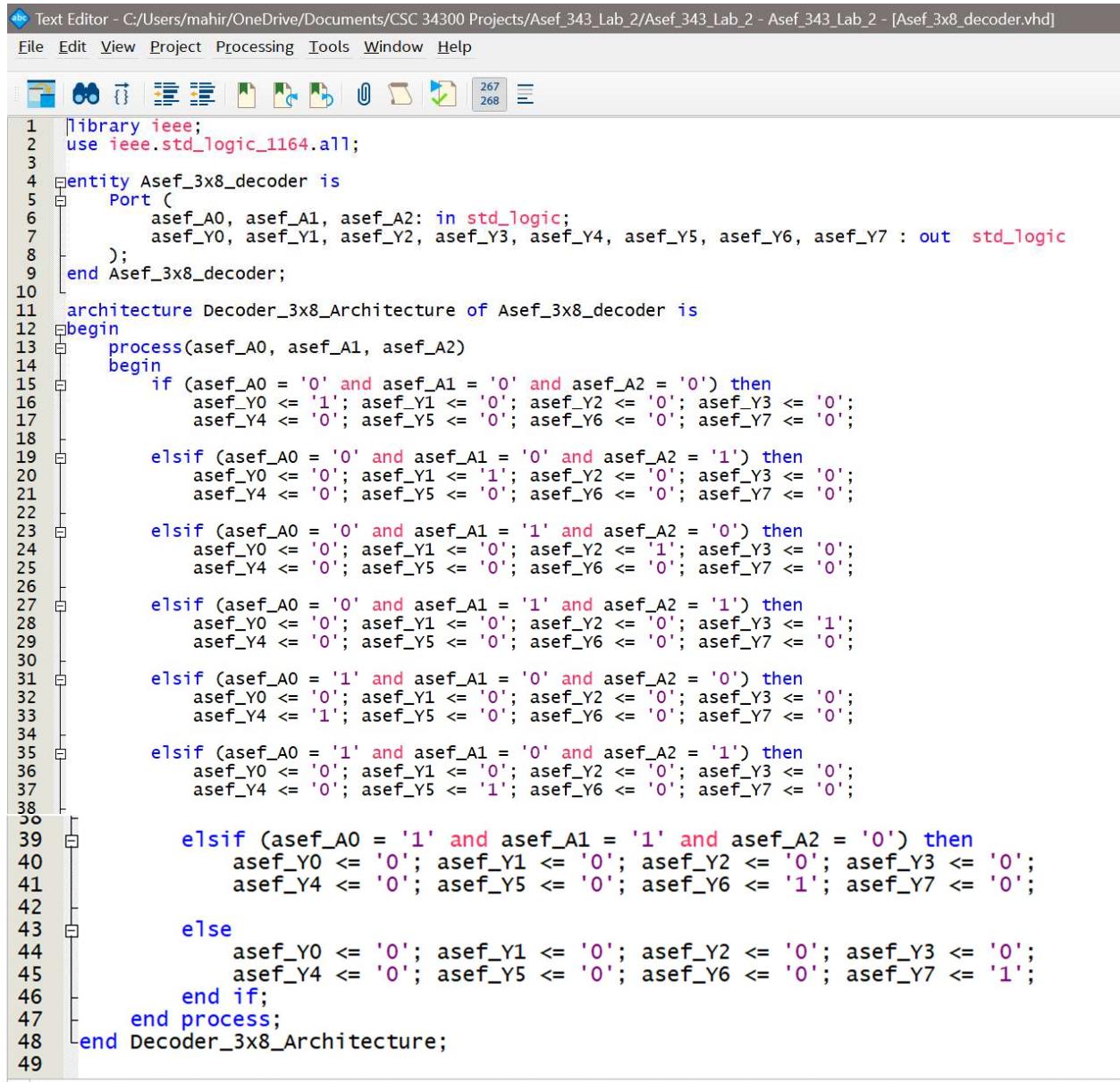
Symbol:



Truth Table:

Input			Output								
A2	A1	A0	D7	D6	D5	D4	D3	D2	D1	D0	
0	0	0	0	0	0	0	0	0	0	1	
0	0	1	0	0	0	0	0	0	1	0	
0	1	0	0	0	0	0	0	1	0	0	
0	1	1	0	0	0	0	1	0	0	0	
1	0	0	0	0	0	1	0	0	0	0	
1	0	1	0	0	1	0	0	0	0	0	
1	1	0	0	1	0	0	0	0	0	0	
1	1	1	1	0	0	0	0	0	0	0	

VHDL Code:



The screenshot shows a VHDL text editor window with the following details:

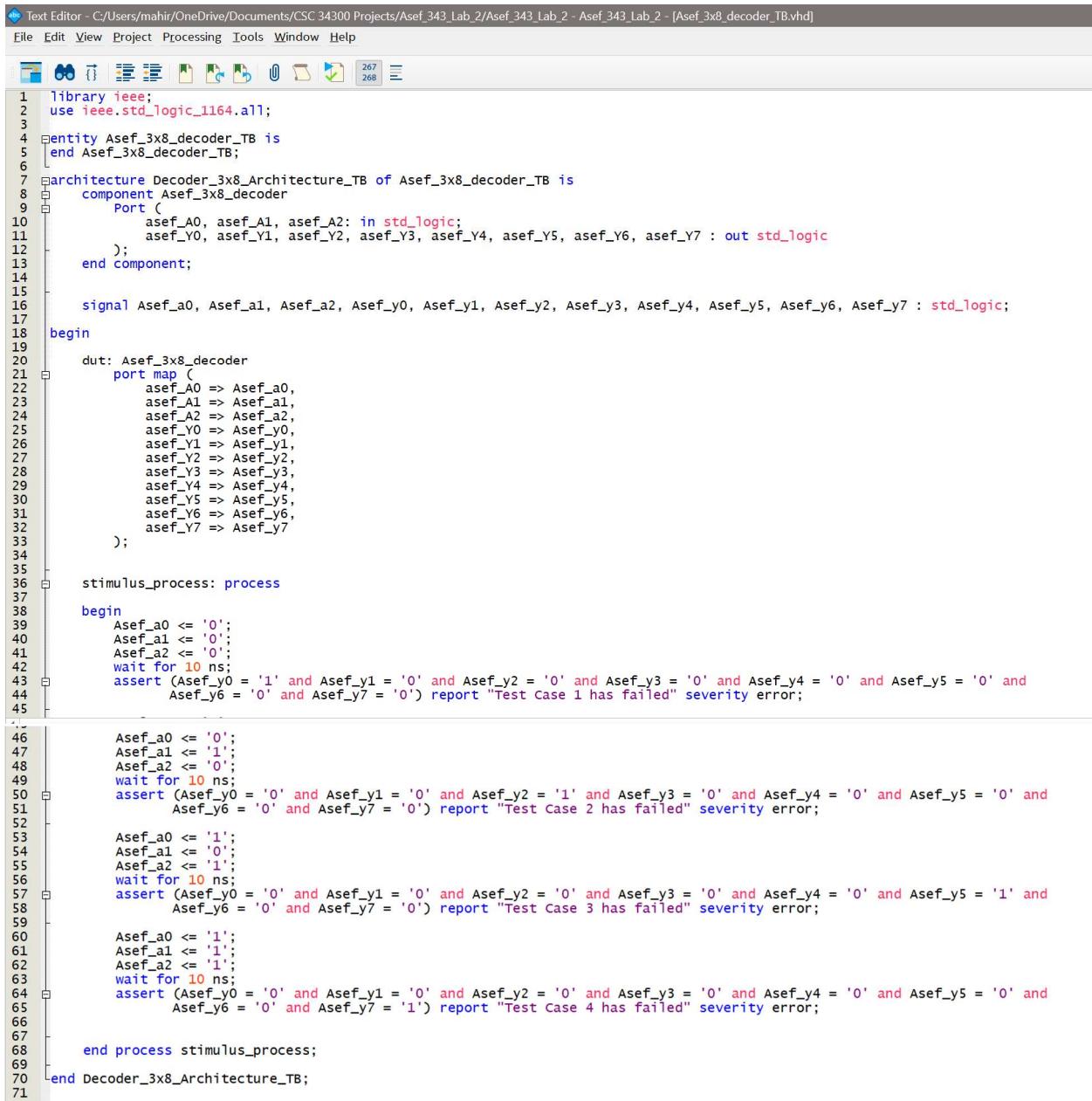
- Title Bar:** Text Editor - C:/Users/mahir/OneDrive/Documents/CSC 34300 Projects/Asef_343_Lab_2/Asef_343_Lab_2 - Asef_343_Lab_2 - [Asef_3x8_decoder.vhd]
- Menu Bar:** File Edit View Project Processing Tools Window Help
- Toolbar:** Includes icons for New, Open, Save, Copy, Paste, Find, and others.
- Status Bar:** Shows line numbers 267 and 268.
- Code Area:** Displays the VHDL code for the Asef_3x8_decoder entity. The code defines a 3-to-8 decoder with three inputs (asef_A0, asef_A1, asef_A2) and eight outputs (asef_Y0 to asef_Y7). It uses a process to map the input combinations to the output values.

```

1 library ieee;
2 use ieee.std_logic_1164.all;
3
4 entity Asef_3x8_decoder is
5   Port (
6     asef_A0, asef_A1, asef_A2: in std_logic;
7     asef_Y0, asef_Y1, asef_Y2, asef_Y3, asef_Y4, asef_Y5, asef_Y6, asef_Y7 : out std_logic
8   );
9 end Asef_3x8_decoder;
10
11 architecture Decoder_3x8_Architecture of Asef_3x8_decoder is
12 begin
13   process(asef_A0, asef_A1, asef_A2)
14   begin
15     if (asef_A0 = '0' and asef_A1 = '0' and asef_A2 = '0') then
16       asef_Y0 <= '1'; asef_Y1 <= '0'; asef_Y2 <= '0'; asef_Y3 <= '0';
17       asef_Y4 <= '0'; asef_Y5 <= '0'; asef_Y6 <= '0'; asef_Y7 <= '0';
18
19     elsif (asef_A0 = '0' and asef_A1 = '0' and asef_A2 = '1') then
20       asef_Y0 <= '0'; asef_Y1 <= '1'; asef_Y2 <= '0'; asef_Y3 <= '0';
21       asef_Y4 <= '0'; asef_Y5 <= '0'; asef_Y6 <= '0'; asef_Y7 <= '0';
22
23     elsif (asef_A0 = '0' and asef_A1 = '1' and asef_A2 = '0') then
24       asef_Y0 <= '0'; asef_Y1 <= '0'; asef_Y2 <= '1'; asef_Y3 <= '0';
25       asef_Y4 <= '0'; asef_Y5 <= '0'; asef_Y6 <= '0'; asef_Y7 <= '0';
26
27     elsif (asef_A0 = '0' and asef_A1 = '1' and asef_A2 = '1') then
28       asef_Y0 <= '0'; asef_Y1 <= '0'; asef_Y2 <= '0'; asef_Y3 <= '1';
29       asef_Y4 <= '0'; asef_Y5 <= '0'; asef_Y6 <= '0'; asef_Y7 <= '0';
30
31     elsif (asef_A0 = '1' and asef_A1 = '0' and asef_A2 = '0') then
32       asef_Y0 <= '0'; asef_Y1 <= '0'; asef_Y2 <= '0'; asef_Y3 <= '0';
33       asef_Y4 <= '1'; asef_Y5 <= '0'; asef_Y6 <= '0'; asef_Y7 <= '0';
34
35     elsif (asef_A0 = '1' and asef_A1 = '0' and asef_A2 = '1') then
36       asef_Y0 <= '0'; asef_Y1 <= '0'; asef_Y2 <= '0'; asef_Y3 <= '0';
37       asef_Y4 <= '0'; asef_Y5 <= '1'; asef_Y6 <= '0'; asef_Y7 <= '0';
38
39     elsif (asef_A0 = '1' and asef_A1 = '1' and asef_A2 = '0') then
40       asef_Y0 <= '0'; asef_Y1 <= '0'; asef_Y2 <= '0'; asef_Y3 <= '0';
41       asef_Y4 <= '0'; asef_Y5 <= '0'; asef_Y6 <= '1'; asef_Y7 <= '0';
42
43     else
44       asef_Y0 <= '0'; asef_Y1 <= '0'; asef_Y2 <= '0'; asef_Y3 <= '0';
45       asef_Y4 <= '0'; asef_Y5 <= '0'; asef_Y6 <= '0'; asef_Y7 <= '1';
46     end if;
47   end process;
48 end Decoder_3x8_Architecture;
49

```

Testbench Code:



The screenshot shows a text editor window with the title "Text Editor - C:/Users/mahir/OneDrive/Documents/CSC 34300 Projects/Asef_343_Lab_2/Asef_343_Lab_2 - [Asef_3x8_decoder_TB.vhd]". The menu bar includes File, Edit, View, Project, Processing, Tools, Window, and Help. The toolbar has icons for new, open, save, cut, copy, paste, find, and zoom. The status bar shows 267 and 268. The code is a VHDL testbench for a 3x8 decoder:

```

1 library ieee;
2 use ieee.std_logic_1164.all;
3
4 entity Asef_3x8_decoder_TB is
5 end Asef_3x8_decoder_TB;
6
7 architecture Decoder_3x8_Architecture_TB of Asef_3x8_decoder_TB is
8   component Asef_3x8_decoder
9     Port (
10       asef_A0, asef_A1, asef_A2: in std_logic;
11       asef_Y0, asef_Y1, asef_Y2, asef_Y3, asef_Y4, asef_Y5, asef_Y6, asef_Y7 : out std_logic
12     );
13   end component;
14
15   signal Asef_a0, Asef_a1, Asef_a2, Asef_y0, Asef_y1, Asef_y2, Asef_y3, Asef_y4, Asef_y5, Asef_y6, Asef_y7 : std_logic;
16
17 begin
18
19   dut: Asef_3x8_decoder
20     port map (
21       asef_A0 => Asef_a0,
22       asef_A1 => Asef_a1,
23       asef_A2 => Asef_a2,
24       asef_Y0 => Asef_y0,
25       asef_Y1 => Asef_y1,
26       asef_Y2 => Asef_y2,
27       asef_Y3 => Asef_y3,
28       asef_Y4 => Asef_y4,
29       asef_Y5 => Asef_y5,
30       asef_Y6 => Asef_y6,
31       asef_Y7 => Asef_y7
32     );
33
34
35   stimulus_process: process
36   begin
37
38     begin
39       Asef_a0 <= '0';
40       Asef_a1 <= '0';
41       Asef_a2 <= '0';
42       wait for 10 ns;
43       assert (Asef_y0 = '1' and Asef_y1 = '0' and Asef_y2 = '0' and Asef_y3 = '0' and Asef_y4 = '0' and Asef_y5 = '0' and
44         Asef_y6 = '0' and Asef_y7 = '0') report "Test Case 1 has failed" severity error;
45
46       Asef_a0 <= '0';
47       Asef_a1 <= '1';
48       Asef_a2 <= '0';
49       wait for 10 ns;
50       assert (Asef_y0 = '0' and Asef_y1 = '0' and Asef_y2 = '1' and Asef_y3 = '0' and Asef_y4 = '0' and Asef_y5 = '0' and
51         Asef_y6 = '0' and Asef_y7 = '0') report "Test Case 2 has failed" severity error;
52
53       Asef_a0 <= '1';
54       Asef_a1 <= '0';
55       Asef_a2 <= '1';
56       wait for 10 ns;
57       assert (Asef_y0 = '0' and Asef_y1 = '0' and Asef_y2 = '0' and Asef_y3 = '0' and Asef_y4 = '0' and Asef_y5 = '1' and
58         Asef_y6 = '0' and Asef_y7 = '0') report "Test Case 3 has failed" severity error;
59
60       Asef_a0 <= '1';
61       Asef_a1 <= '1';
62       Asef_a2 <= '1';
63       wait for 10 ns;
64       assert (Asef_y0 = '0' and Asef_y1 = '0' and Asef_y2 = '0' and Asef_y3 = '0' and Asef_y4 = '0' and Asef_y5 = '0' and
65         Asef_y6 = '0' and Asef_y7 = '1') report "Test Case 4 has failed" severity error;
66
67
68   end process stimulus_process;
69
70 end Decoder_3x8_Architecture_TB;
71

```

Generated VHDL Code:

Text Editor - C:/Users/mahir/OneDrive/Documents/CSC 34300 Projects/Asef_343_Lab_2/Asef_343_Lab_2 - [A]

File Edit View Project Processing Tools Window Help

267
268

```

1  -- Copyright (C) 2023 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 23.1std.0 Build 991 11/28/2023 SC Lite Edition"
18  CREATED      "Wed Mar  6 00:34:18 2024"
19
20 LIBRARY ieee;
21 USE ieee.std_logic_1164.all;
22
23 LIBRARY work;
24
25 ENTITY Asef_3x8_decoder_block IS
26   PORT
27   (
28     asef_A2 : IN STD_LOGIC;
29     asef_A1 : IN STD_LOGIC;
30     asef_A0 : IN STD_LOGIC;
31     asef_Y0 : OUT STD_LOGIC;
32     asef_Y1 : OUT STD_LOGIC;
33     asef_Y2 : OUT STD_LOGIC;
34     asef_Y3 : OUT STD_LOGIC;
35     asef_Y4 : OUT STD_LOGIC;
36     asef_Y5 : OUT STD_LOGIC;
37     asef_Y6 : OUT STD_LOGIC;
38     asef_Y7 : OUT STD_LOGIC
39   );
40 END Asef_3x8_decoder_block;
41
42 ARCHITECTURE bdf_type OF Asef_3x8_decoder_block IS
43
44   SIGNAL  SYNTHESIZED_WIRE_12 : STD_LOGIC;
45   SIGNAL  SYNTHESIZED_WIRE_13 : STD_LOGIC;

```

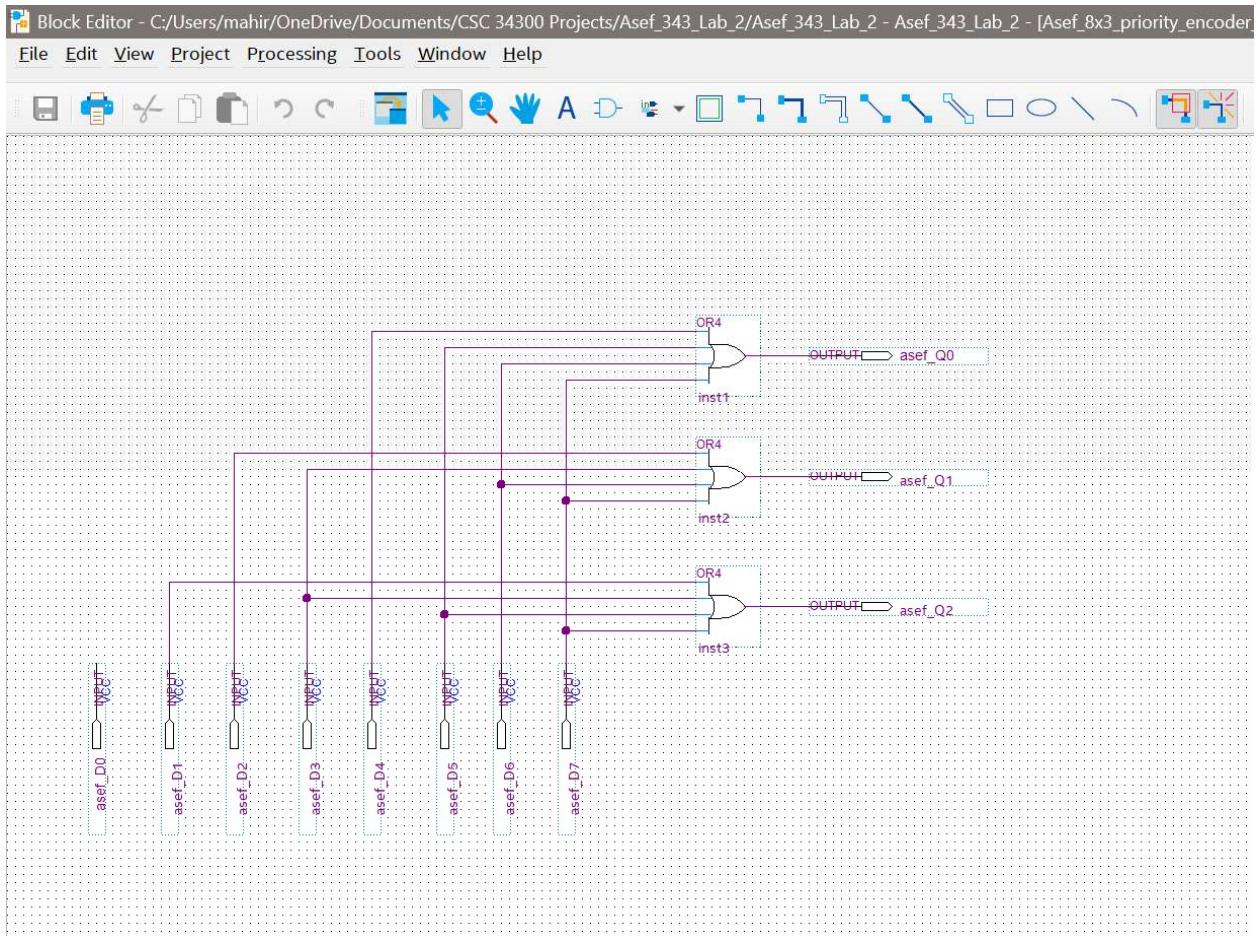
```

46  SIGNAL  SYNTHESIZED_WIRE_14 :  STD_LOGIC;
47
48 BEGIN
49
50
51
52  SYNTHESIZED_WIRE_12 <= NOT(asef_A2);
53
54
55
56  SYNTHESIZED_WIRE_13 <= NOT(asef_A1);
57
58
59
60  asef_Y7 <= asef_A2 AND asef_A1 AND asef_A0;
61
62
63  SYNTHESIZED_WIRE_14 <= NOT(asef_A0);
64
65
66
67  asef_Y1 <= SYNTHESIZED_WIRE_12 AND SYNTHESIZED_WIRE_13 AND asef_A0;
68
69
70  asef_Y2 <= SYNTHESIZED_WIRE_12 AND asef_A1 AND SYNTHESIZED_WIRE_14;
71
72
73  asef_Y3 <= SYNTHESIZED_WIRE_12 AND asef_A1 AND asef_A0;
74
75
76  asef_Y4 <= asef_A2 AND SYNTHESIZED_WIRE_13 AND SYNTHESIZED_WIRE_14;
77
78
79  asef_Y5 <= asef_A2 AND SYNTHESIZED_WIRE_13 AND asef_A0;
80
81
82  asef_Y6 <= asef_A2 AND asef_A1 AND SYNTHESIZED_WIRE_14;
83
84
85  asef_Y0 <= SYNTHESIZED_WIRE_12 AND SYNTHESIZED_WIRE_13 AND SYNTHESIZED_WIRE_14;
86
87
88
89 END bdf_type;

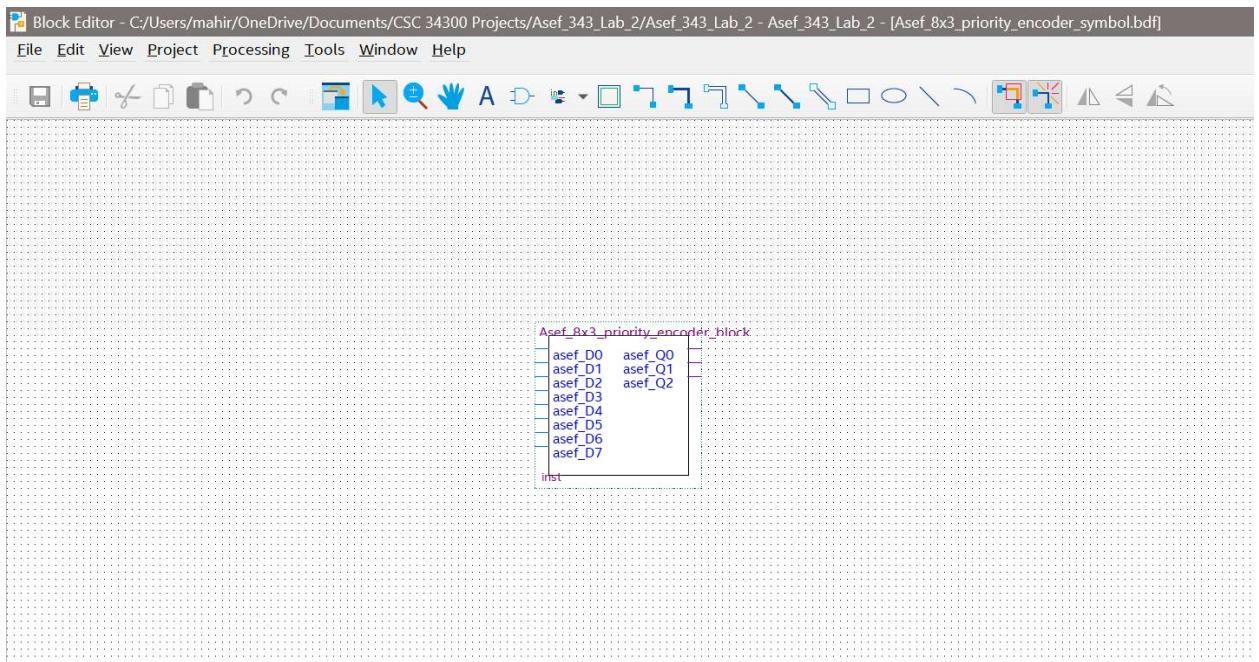
```

E. 8:3 Priority Encoder

Schematic:



Symbol:



Truth Table:

Input								Output		
D7	D6	D5	D4	D3	D2	D1	D0	Q2	Q1	Q0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	X	0	0	1
0	0	0	0	0	1	X	X	0	1	0
0	0	0	0	1	X	X	X	0	1	1
0	0	0	1	X	X	X	X	1	0	0
0	0	1	X	X	X	X	X	1	0	1
0	1	X	X	X	X	X	X	1	1	0
1	X	X	X	X	X	X	X	1	1	1

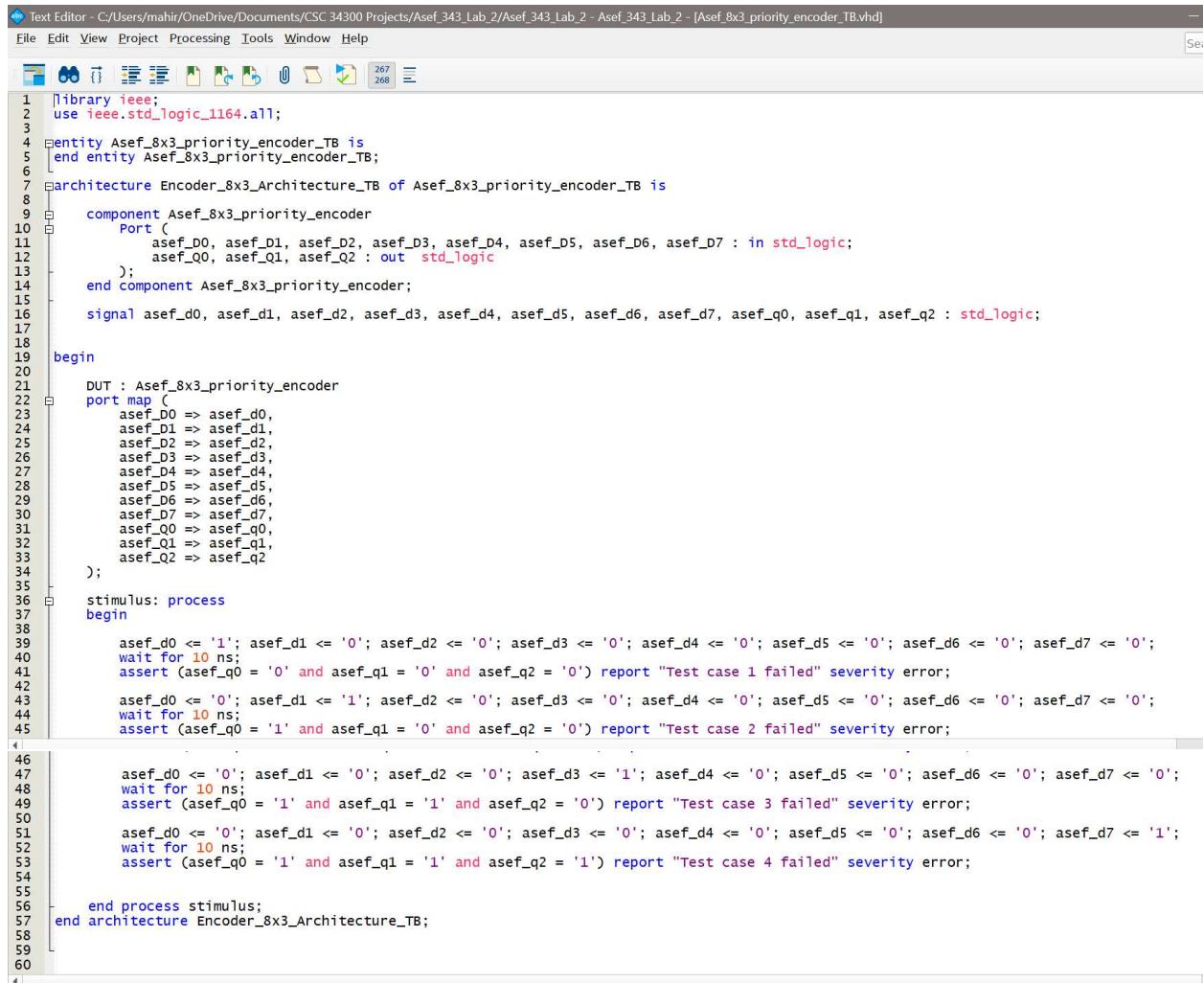
VHDL Code:

```

1 library ieee;
2 use ieee.std_logic_1164.all;
3
4 entity Asef_8x3_priority_encoder is
5   Port (
6     asef_D0, asef_D1, asef_D2, asef_D3, asef_D4, asef_D5, asef_D6, asef_D7 : in std_logic;
7     asef_Q0, asef_Q1, asef_Q2 : out std_logic
8   );
9 end Asef_8x3_priority_encoder;
10
11 architecture Encoder_8x3_Architecture of Asef_8x3_priority_encoder is
12 begin
13   process(asef_D0, asef_D1, asef_D2, asef_D3, asef_D4, asef_D5, asef_D6, asef_D7)
14   begin
15     if asef_D0 = '1' then
16       asef_Q0 <= '0'; asef_Q1 <= '0'; asef_Q2 <= '0';
17
18     elsif asef_D1 = '1' then
19       asef_Q0 <= '1'; asef_Q1 <= '0'; asef_Q2 <= '0';
20
21     elsif asef_D2 = '1' then
22       asef_Q0 <= '0'; asef_Q1 <= '1'; asef_Q2 <= '0';
23
24     elsif asef_D3 = '1' then
25       asef_Q0 <= '1'; asef_Q1 <= '1'; asef_Q2 <= '0';
26
27     elsif asef_D4 = '1' then
28       asef_Q0 <= '0'; asef_Q1 <= '0'; asef_Q2 <= '1';
29
30     elsif asef_D5 = '1' then
31       asef_Q0 <= '1'; asef_Q1 <= '0'; asef_Q2 <= '1';
32
33     elsif asef_D6 = '1' then
34       asef_Q0 <= '0'; asef_Q1 <= '1'; asef_Q2 <= '1';
35
36     elsif asef_D7 = '1' then
37       asef_Q0 <= '1'; asef_Q1 <= '1'; asef_Q2 <= '1';
38
39     else
40       asef_Q0 <= '0'; asef_Q1 <= '0'; asef_Q2 <= '0';
41     end if;
42   end process;
43 end Encoder_8x3_Architecture;
44

```

Testbench Code:



```

1 library ieee;
2 use ieee.std_logic_1164.all;
3
4 entity Asef_8x3_priority_encoder_TB is
5 end entity Asef_8x3_priority_encoder_TB;
6
7 architecture Encoder_8x3_Architecture_TB of Asef_8x3_priority_encoder_TB is
8
9 component Asef_8x3_priority_encoder
10 Port (
11     asef_D0, asef_D1, asef_D2, asef_D3, asef_D4, asef_D5, asef_D6, asef_D7 : in std_logic;
12     asef_Q0, asef_Q1, asef_Q2 : out std_logic
13 );
14 end component Asef_8x3_priority_encoder;
15
16 signal asef_d0, asef_d1, asef_d2, asef_d3, asef_d4, asef_d5, asef_d6, asef_d7, asef_q0, asef_q1, asef_q2 : std_logic;
17
18 begin
19
20     DUT : Asef_8x3_priority_encoder
21     port map (
22         asef_D0 => asef_d0,
23         asef_D1 => asef_d1,
24         asef_D2 => asef_d2,
25         asef_D3 => asef_d3,
26         asef_D4 => asef_d4,
27         asef_D5 => asef_d5,
28         asef_D6 => asef_d6,
29         asef_D7 => asef_d7,
30         asef_Q0 => asef_q0,
31         asef_Q1 => asef_q1,
32         asef_Q2 => asef_q2
33     );
34
35
36     stimulus: process
37     begin
38
39         asef_d0 <= '1'; asef_d1 <= '0'; asef_d2 <= '0'; asef_d3 <= '0'; asef_d4 <= '0'; asef_d5 <= '0'; asef_d6 <= '0'; asef_d7 <= '0';
40         wait for 10 ns;
41         assert (asef_q0 = '0' and asef_q1 = '0' and asef_q2 = '0') report "Test case 1 failed" severity error;
42
43         asef_d0 <= '0'; asef_d1 <= '1'; asef_d2 <= '0'; asef_d3 <= '0'; asef_d4 <= '0'; asef_d5 <= '0'; asef_d6 <= '0'; asef_d7 <= '0';
44         wait for 10 ns;
45         assert (asef_q0 = '1' and asef_q1 = '0' and asef_q2 = '0') report "Test case 2 failed" severity error;
46
47         asef_d0 <= '0'; asef_d1 <= '0'; asef_d2 <= '0'; asef_d3 <= '1'; asef_d4 <= '0'; asef_d5 <= '0'; asef_d6 <= '0'; asef_d7 <= '0';
48         wait for 10 ns;
49         assert (asef_q0 = '1' and asef_q1 = '1' and asef_q2 = '0') report "Test case 3 failed" severity error;
50
51         asef_d0 <= '0'; asef_d1 <= '0'; asef_d2 <= '0'; asef_d3 <= '0'; asef_d4 <= '0'; asef_d5 <= '0'; asef_d6 <= '0'; asef_d7 <= '1';
52         wait for 10 ns;
53         assert (asef_q0 = '1' and asef_q1 = '1' and asef_q2 = '1') report "Test case 4 failed" severity error;
54
55
56     end process stimulus;
57 end architecture Encoder_8x3_Architecture_TB;
58
59

```

Generated VHDL Code:

Text Editor - C:/Users/mahir/OneDrive/Documents/CSC 34300 Projects/Asef 343 Lab 2/Asef 343 Lab 2 - Asef 343 La

File Edit View Project Processing Tools Window Help

267 268

```

1  --- Copyright (C) 2023 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 23.1std.0 Build 991 11/28/2023 SC Lite Edition"
18 --- CREATED      "Wed Mar  6 00:37:52 2024"
19
20 LIBRARY ieee;
21 USE ieee.std_logic_1164.all;
22
23 LIBRARY work;
24
25 ENTITY Asef_8x3_priority_encoder_block IS
26   PORT
27   (
28     asef_D0 : IN STD_LOGIC;
29     asef_D1 : IN STD_LOGIC;
30     asef_D2 : IN STD_LOGIC;
31     asef_D3 : IN STD_LOGIC;
32     asef_D4 : IN STD_LOGIC;
33     asef_D5 : IN STD_LOGIC;
34     asef_D6 : IN STD_LOGIC;
35     asef_D7 : IN STD_LOGIC;
36     asef_Q0 : OUT STD_LOGIC;
37     asef_Q1 : OUT STD_LOGIC;
38     asef_Q2 : OUT STD_LOGIC
39   );
40 END Asef_8x3_priority_encoder_block;
41
42 ARCHITECTURE bdf_type OF Asef_8x3_priority_encoder_block IS
43

```

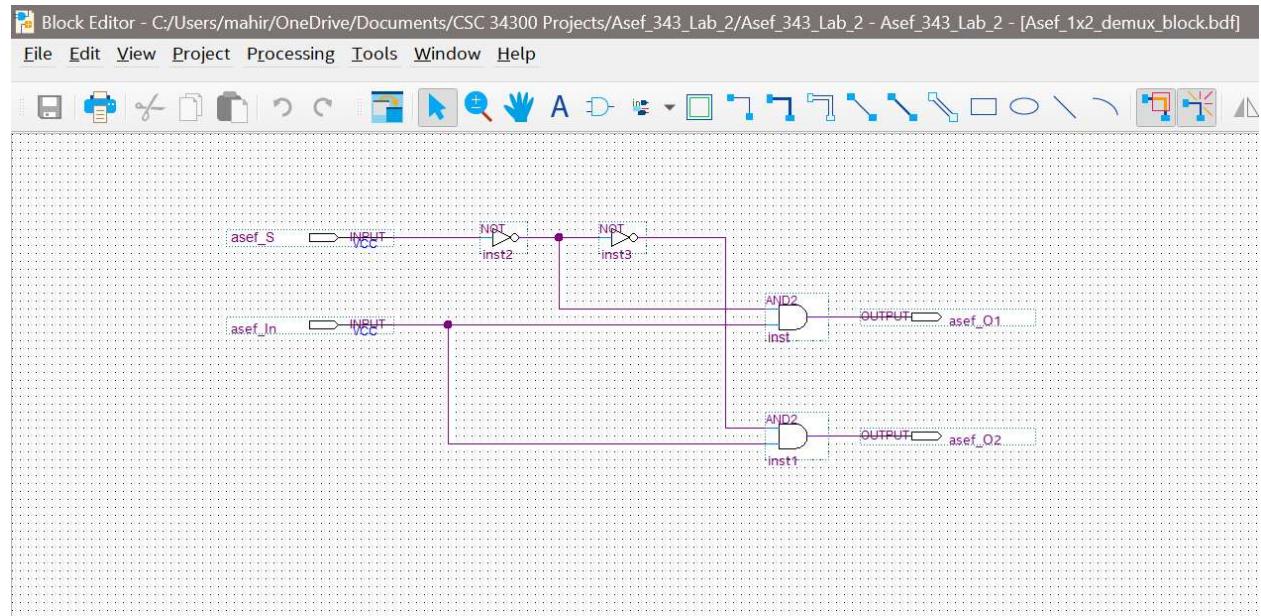
```

44
45 BEGIN
46
47
48
49 asef_Q0 <= asef_D4 OR asef_D6 OR asef_D7 OR asef_D5;
50
51 asef_Q1 <= asef_D2 OR asef_D6 OR asef_D7 OR asef_D3;
52
53 asef_Q2 <= asef_D1 OR asef_D5 OR asef_D7 OR asef_D3;
54
55
56
57
58
59 END bdf_type;

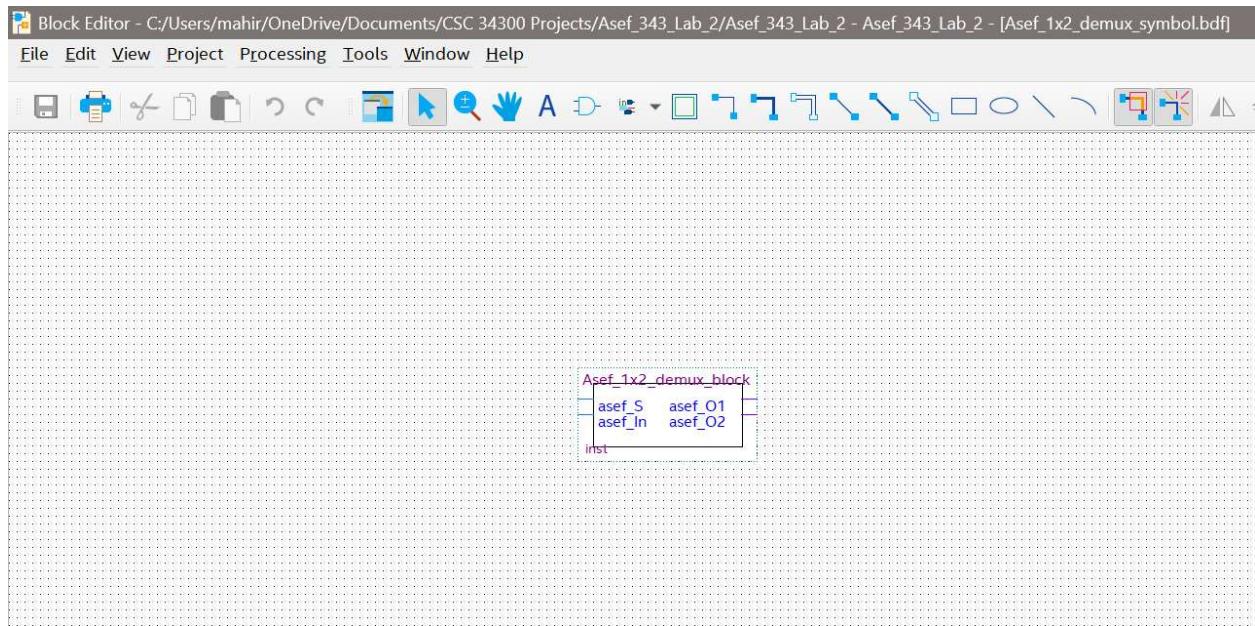
```

F. 1:2 Demultiplexer

Schematic:

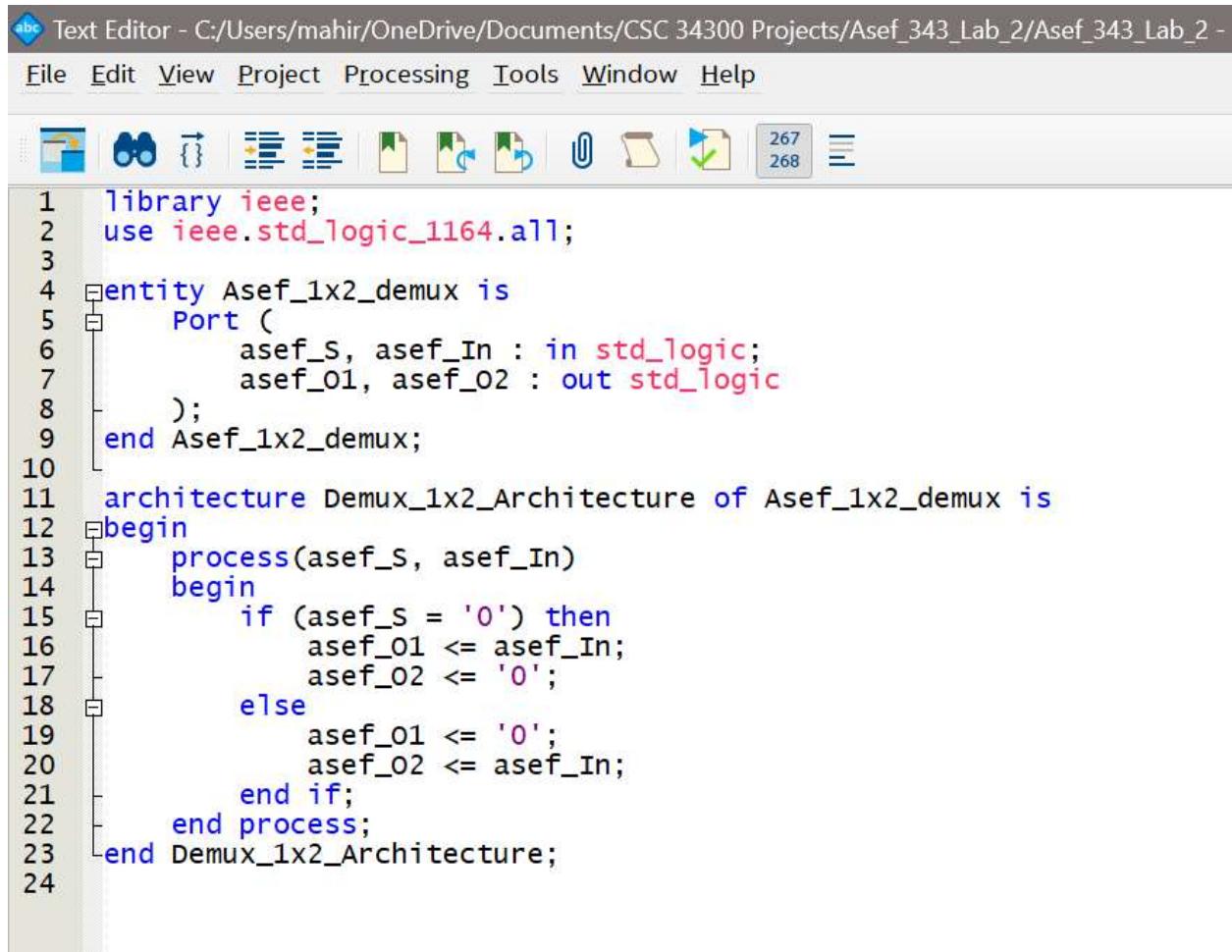


Symbol:

**Truth Table:**

Input	Output	
S	Out1	Out2
0	0	In
1	In	0

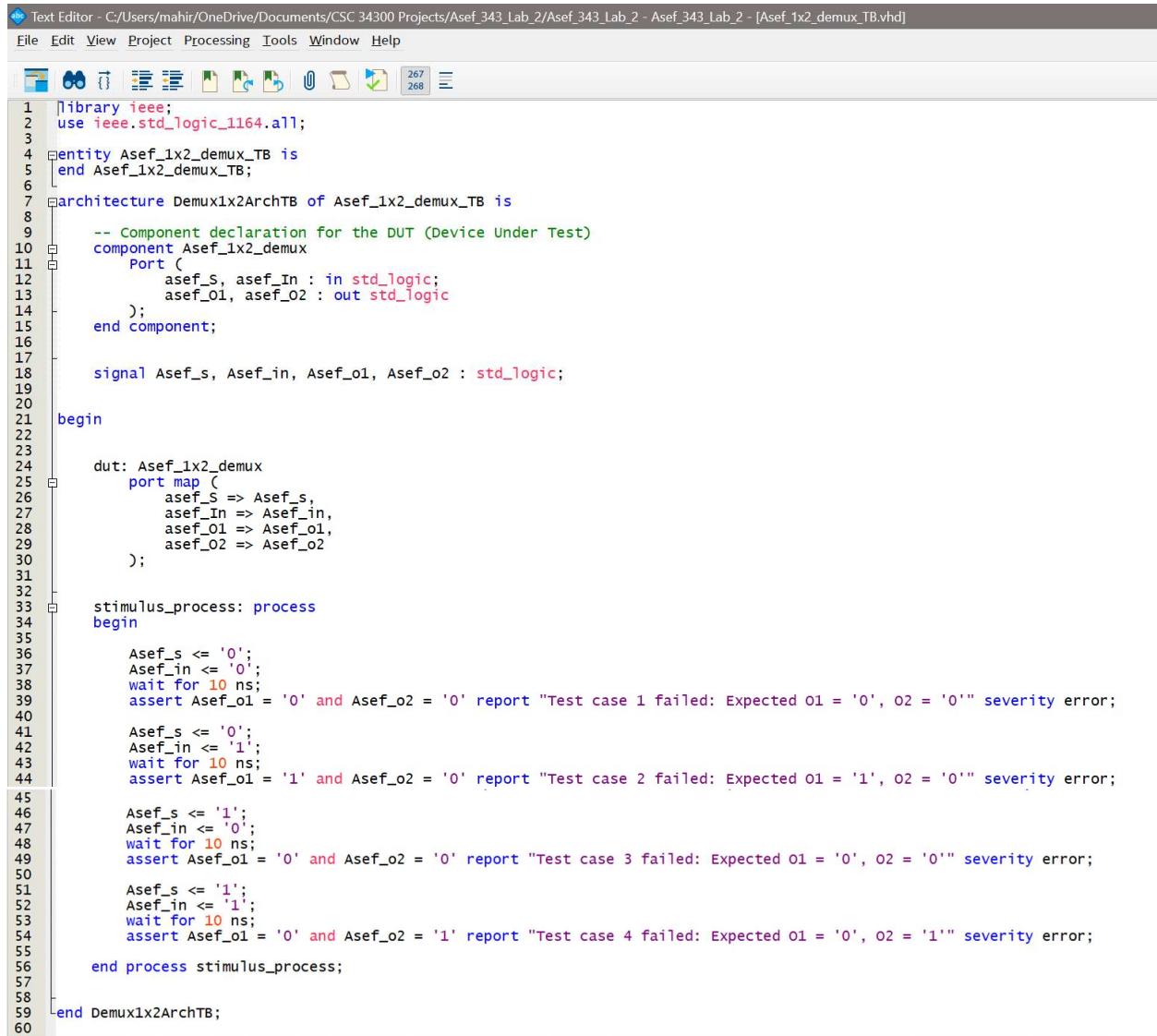
VHDL Code:



The screenshot shows a text editor window with the title "Text Editor - C:/Users/mahir/OneDrive/Documents/CSC 34300 Projects/Asef_343_Lab_2/Asef_343_Lab_2 -". The menu bar includes File, Edit, View, Project, Processing, Tools, Window, and Help. The toolbar contains icons for file operations like Open, Save, Copy, Paste, and Undo. A status bar at the bottom right shows "267 268". The code itself is a VHDL entity and architecture:

```
1 library ieee;
2 use ieee.std_logic_1164.all;
3
4 entity Asef_1x2_demux is
5     Port (
6         asef_S, asef_In : in std_logic;
7         asef_O1, asef_O2 : out std_logic
8     );
9 end Asef_1x2_demux;
10
11 architecture Demux_1x2_Architecture of Asef_1x2_demux is
12 begin
13     process(asef_S, asef_In)
14     begin
15         if (asef_S = '0') then
16             asef_O1 <= asef_In;
17             asef_O2 <= '0';
18         else
19             asef_O1 <= '0';
20             asef_O2 <= asef_In;
21         end if;
22     end process;
23 end Demux_1x2_Architecture;
24
```

Testbench Code:



```

1 library ieee;
2 use ieee.std_logic_1164.all;
3
4 entity Asef_1x2_demux_TB is
5 end Asef_1x2_demux_TB;
6
7 architecture Demux1x2ArchTB of Asef_1x2_demux_TB is
8
9   -- Component declaration for the DUT (Device Under Test)
10  component Asef_1x2_demux
11    Port (
12      asef_S, asef_In : in std_logic;
13      asef_O1, asef_O2 : out std_logic
14    );
15  end component;
16
17  signal Asef_S, Asef_in, Asef_O1, Asef_O2 : std_logic;
18
19
20 begin
21
22  dut: Asef_1x2_demux
23    port map (
24      asef_S => Asef_S,
25      asef_In => Asef_in,
26      asef_O1 => Asef_O1,
27      asef_O2 => Asef_O2
28    );
29
30
31
32  stimulus_process: process
33  begin
34
35    Asef_S <= '0';
36    Asef_in <= '0';
37    wait for 10 ns;
38    assert Asef_O1 = '0' and Asef_O2 = '0' report "Test case 1 failed: Expected O1 = '0', O2 = '0'" severity error;
39
40    Asef_S <= '0';
41    Asef_in <= '1';
42    wait for 10 ns;
43    assert Asef_O1 = '1' and Asef_O2 = '0' report "Test case 2 failed: Expected O1 = '1', O2 = '0'" severity error;
44
45    Asef_S <= '1';
46    Asef_in <= '0';
47    wait for 10 ns;
48    assert Asef_O1 = '0' and Asef_O2 = '0' report "Test case 3 failed: Expected O1 = '0', O2 = '0'" severity error;
49
50    Asef_S <= '1';
51    Asef_in <= '1';
52    wait for 10 ns;
53    assert Asef_O1 = '0' and Asef_O2 = '1' report "Test case 4 failed: Expected O1 = '0', O2 = '1'" severity error;
54
55  end process stimulus_process;
56
57
58
59 end Demux1x2ArchTB;
60

```

Generated VHDL Code:

abc Text Editor - C:/Users/mahir/OneDrive/Documents/CSC 34300 Projects/Asef_343_Lab_2/Asef_343_Lab_2 - Asef_343_Lab_2 - [As]

File Edit View Project Processing Tools Window Help

267 268

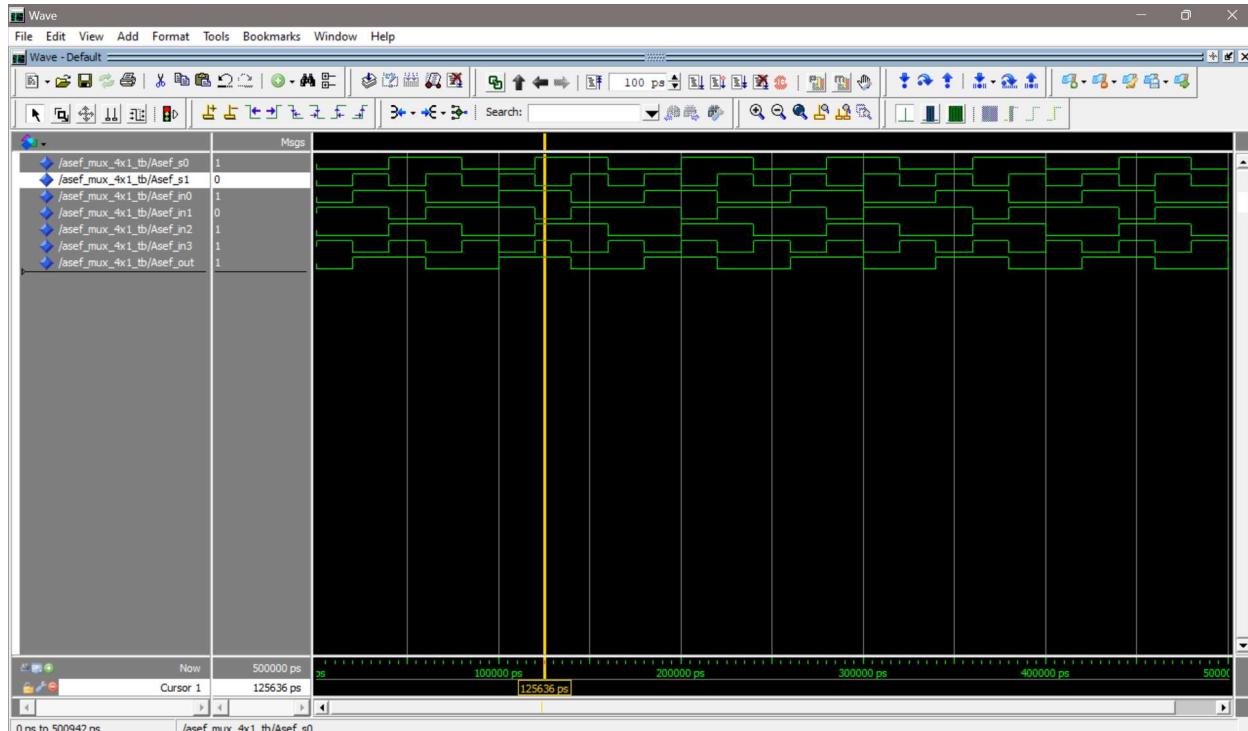
```

1  -- Copyright (C) 2023 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 23.1std.0 Build 991 11/28/2023 SC Lite Edition"
18 -- CREATED      "Wed Mar  6 00:38:28 2024"
19
20 LIBRARY ieee;
21 USE ieee.std_logic_1164.all;
22
23 LIBRARY work;
24
25 ENTITY Asef_1x2_demux_block IS
26   PORT
27   (
28     asef_S :  IN STD_LOGIC;
29     asef_In :  IN STD_LOGIC;
30     asef_O1 :  OUT STD_LOGIC;
31     asef_O2 :  OUT STD_LOGIC
32   );
33 END Asef_1x2_demux_block;
34
35 ARCHITECTURE bdf_type OF Asef_1x2_demux_block IS
36
37 SIGNAL    SYNTHESIZED_WIRE_3 :  STD_LOGIC;
38 SIGNAL    SYNTHESIZED_WIRE_1 :  STD_LOGIC;
39
40
41 BEGIN
42
43
44   asef_O1 <= SYNTHESIZED_WIRE_3 AND asef_In;
45
46
47
48   asef_O2 <= SYNTHESIZED_WIRE_1 AND asef_In;
49
50
51   SYNTHESIZED_WIRE_3 <= NOT(asef_S);
52
53
54
55   SYNTHESIZED_WIRE_1 <= NOT(SYNTHESIZED_WIRE_3);
56
57
58
59 END bdf_type;

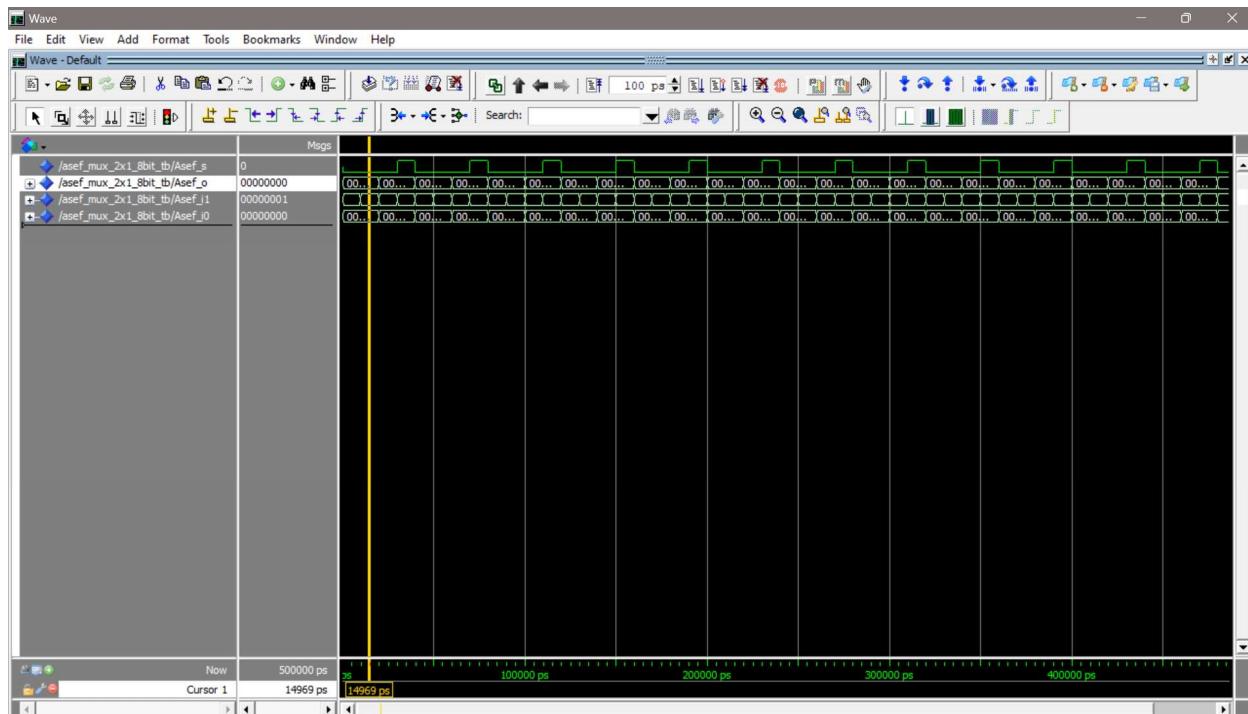
```

Simulations:

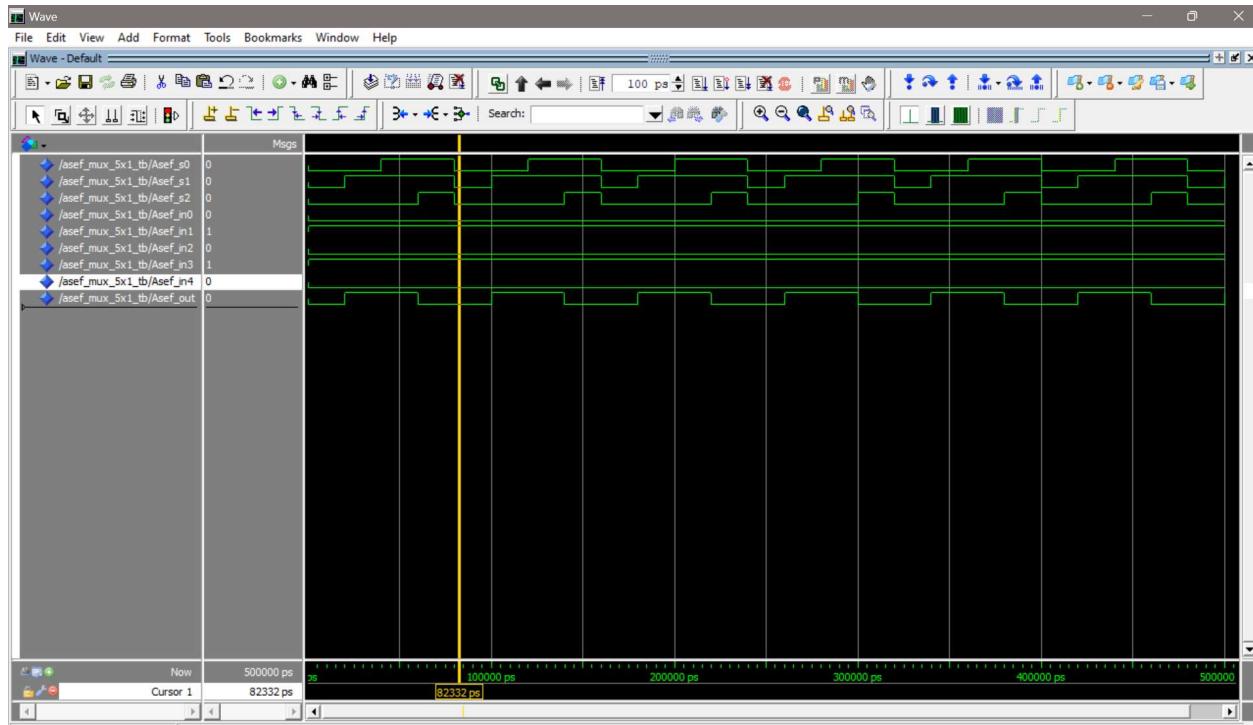
A. Mux 4:1



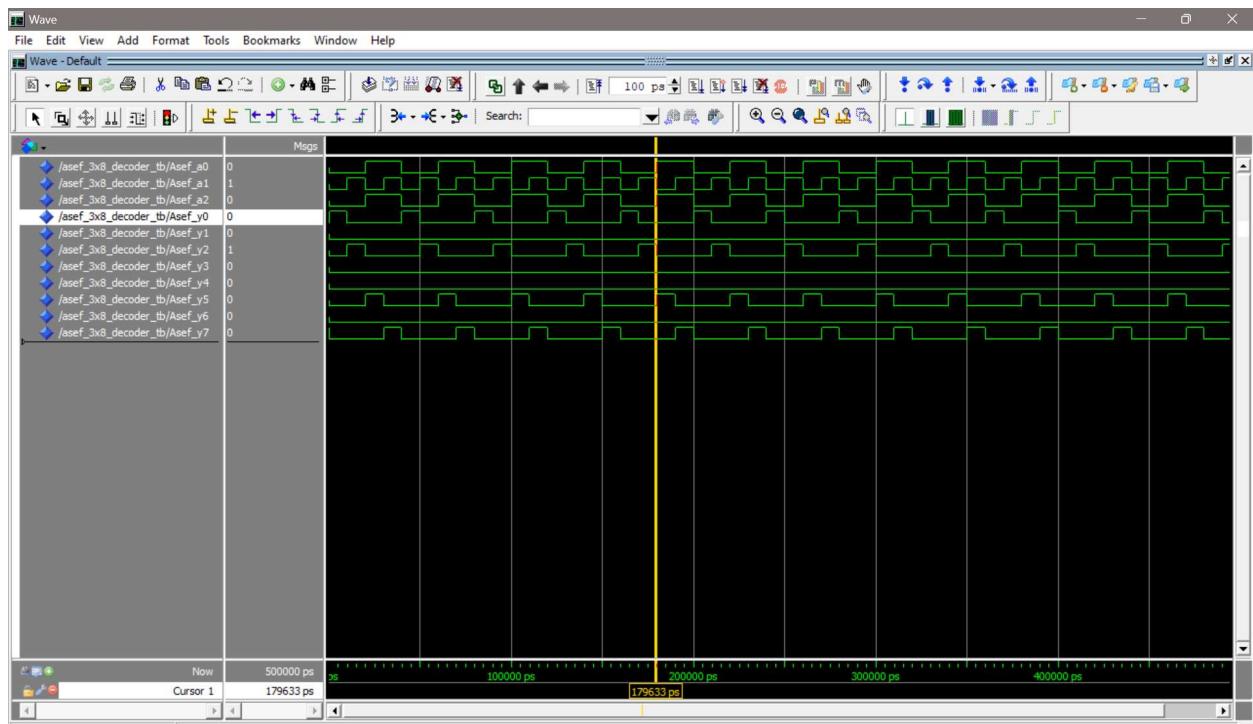
B. Mux 2:1 8 Bits Input and Output



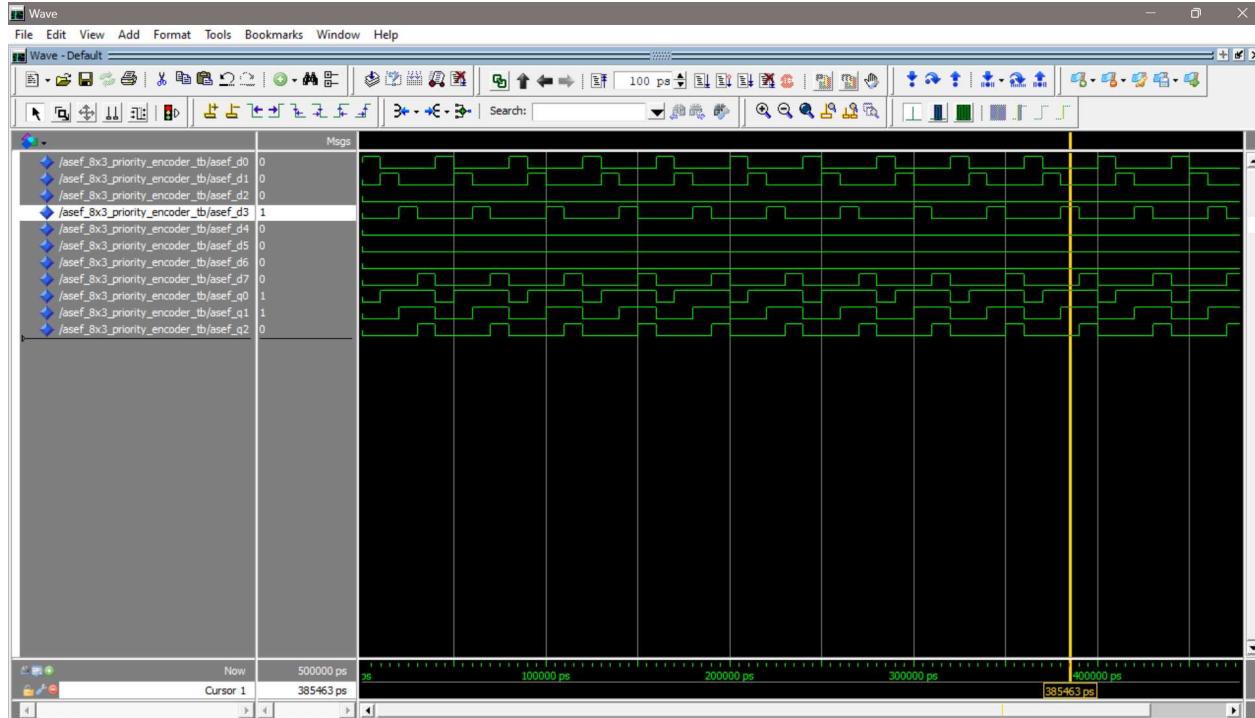
C. Mux 5:1



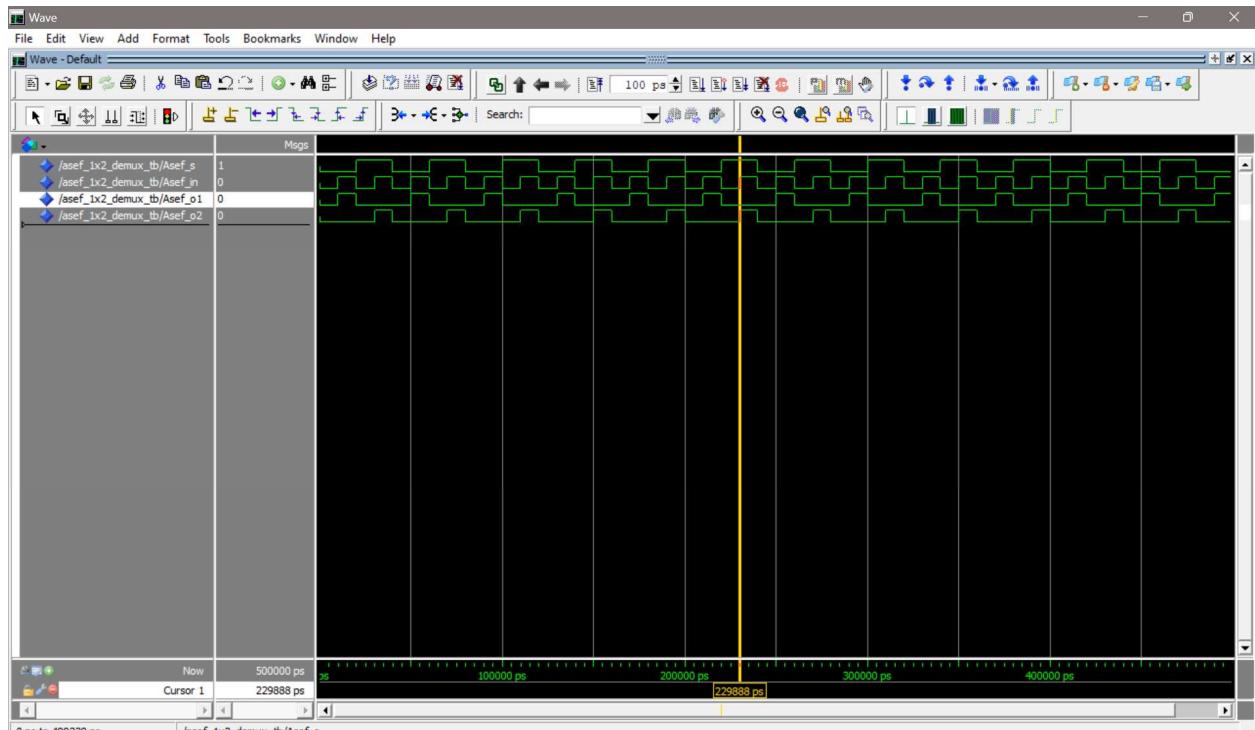
D. 3:8 Decoder



E. 8:3 Priority Encoder



F. 1:2 Demultiplexer



Conclusion:

We have reached the end of this Mux, Encoder, Decoder basic laboratory experiment. Throughout this lab, we have been able to successfully design all our logic gates using schematic in Quartus and complied them with no error. Then for each of our designs we created a symbol block and then using our schematic, we created a truth table for each design as shown above. Next, we have utilized the logic in our truth tables to write our own VHDL code for all individual design and made sure to compile them with no errors. We have also created auto generated VHDL file for all our designs from our block diagrams to compare it to our own written VHDL as also shown above. Finally, we have written testbench VHDL code for each of our projects to simulate all our designs onto Modelsim for verification. In Modelsim, we have selected each diagram's testbench and selected all inputs/outputs signal to generate waves and run them for 500 ns as shown in the pictures above. As we examine and compare our simulations with our schematic design and truth tables, we can observe that all our simulation results perfectly agree with our logic in the truth tables. Since we have compiled all our schematic designs and our VHDL code with no errors and successfully created testbench codes for each of our design to simulate in Modelsim and compared our simulation result with our truth table to find out that they indeed match; overall, this experiment has been a very informative and successful at learning about more complex Multiplexers, complex Encoders and Decoders.