VHDL

Assignment 1

Name: ***Ambarish Prashant Chandurkar***

Roll No. : BT17ECE021

```
1   ----------------------------------------------------------------------------------
2   -- Company:
3   -- Engineer:
4   --
5   -- Create Date:    14:32:25 02/09/2019
6   -- Design Name:
7   -- Module Name:    EightTo1_MUX_usingCase - Behavioral
8   ----------------------------------------------------------------------------------
9   library IEEE;
10  use IEEE.STD_LOGIC_1164.ALL;
11
12  entity EightTo1_MUX_usingCase is
13      Port ( a : in  STD_LOGIC;
14             b : in  STD_LOGIC;
15             c : in  STD_LOGIC;  -- Signal 'a' to 'h' are 8 Active High Data inputs
16             d : in  STD_LOGIC;
17             e : in  STD_LOGIC;
18             f : in  STD_LOGIC;
19             g : in  STD_LOGIC;
20             h : in  STD_LOGIC;
21             s1 : in  STD_LOGIC; -- Signals s1,s2,s3 are Active High Select Inputs
22             s2 : in  STD_LOGIC;
23             s3 : in  STD_LOGIC;
24             En : in  STD_LOGIC; -- En is Active High Enable
25             Y : out  STD_LOGIC); -- Y is Active High Output
26  end EightTo1_MUX_usingCase;
27
28  architecture Behavioral of EightTo1_MUX_usingCase is
29
30  begin
31     process (a,b,c,d,e,f,g,h,s1,s2,s3,En)
32     variable s : STD_LOGIC_VECTOR (2 downto 0); -- A Temporary Signal Vector is
    created for Select input
33     begin                                       -- Since only one identifier is
    allowed in Case statement
34        s(0) := s1;
35        s(1) := s2;
36        s(2) := s3;
37        if En = '1' then
38           case s is
39              when "000" => Y <= a; -- Different Selections for Different Select Inputs
40              when "001" => Y <= b;
41              when "010" => Y <= c;
42              when "011" => Y <= d;
43              when "100" => Y <= e;
44              when "101" => Y <= f;
45              when "110" => Y <= g;
46              when others => Y <= h;
47           end case;
48        else
49           Y <= '0';
50        end if;
51     end process;
52  end Behavioral;
```

```
1  ----------------------------------------------------------------------------------
2  -- Company:
3  -- Engineer:
4  --
5  -- Create Date:    15:42:14 02/09/2019
6  -- Design Name:
7  -- Module Name:    EightTo1_MUX_usingIfElse - Behavioral
8  ----------------------------------------------------------------------------------
9  library IEEE;
10 use IEEE.STD_LOGIC_1164.ALL;
11
12 entity EightTo1_MUX_usingIfElse is
13     Port ( a : in  STD_LOGIC;
14            b : in  STD_LOGIC;
15            c : in  STD_LOGIC;
16            d : in  STD_LOGIC;
17            e : in  STD_LOGIC;
18            f : in  STD_LOGIC;  -- Signal 'a' to 'h' are 8 Active High Data inputs
19            g : in  STD_LOGIC;
20            h : in  STD_LOGIC;
21            s1 : in  STD_LOGIC; -- Signals s1,s2,s3 are Active High Select Inputs
22            s2 : in  STD_LOGIC;
23            s3 : in  STD_LOGIC;
24            En : in  STD_LOGIC; -- En is Active High Enable
25            Y : out  STD_LOGIC); -- Y is Active High Output
26 end EightTo1_MUX_usingIfElse;
27
28 architecture Behavioral of EightTo1_MUX_usingIfElse is
29
30 begin
31 process(a,b,c,d,e,f,g,h,En,s1,s2,s3)
32    begin
33          if En = '1' then
34             if (s3 = '0' and s2 = '0' and s1 = '0') then
35                y <= a;
36             elsif (s3 = '0' and s2 = '0' and s1 = '1') then
37                y <= b;
38             elsif (s3 = '0' and s2 = '1' and s1 = '0') then
39                y <= c;
40             elsif (s3 = '0' and s2 = '1' and s1 = '1') then
41                y <= d;
42             elsif (s3 = '1' and s2 = '0' and s1 = '0') then
43                y <= e;
44             elsif (s3 = '1' and s2 = '0' and s1 = '1') then
45                y <= f;
46             elsif (s3 = '1' and s2 = '1' and s1 = '0') then
47                y <= g;
48             else
49                y <= h;
50             end if;
51          else
52             y <= '0';
53          end if;
54    end process;
55 end Behavioral;
```

```
1  ----------------------------------------------------------------------------------
2  -- Create Date:    16:54:42 02/09/2019
3  -- Design Name:
4  -- Module Name:    EightTo3_ENCODER_usingCase - Behavioral
5  ----------------------------------------------------------------------------------
6  library IEEE;
7  use IEEE.STD_LOGIC_1164.ALL;
8
9  entity EightTo3_ENCODER_usingCase is
10     Port ( D0 : in  STD_LOGIC;
11            D1 : in  STD_LOGIC; --D0 to D7 is DATA input
12            D2 : in  STD_LOGIC;
13            D3 : in  STD_LOGIC;
14            D4 : in  STD_LOGIC;
15            D5 : in  STD_LOGIC;
16            D6 : in  STD_LOGIC;
17            D7 : in  STD_LOGIC;
18            Y : out  STD_LOGIC_VECTOR (2 downto 0); -- Y is output Vector, All are
   Active High
19            En : in  STD_LOGIC); --En is Enable
20 end EightTo3_ENCODER_usingCase;
21
22 architecture Behavioral of EightTo3_ENCODER_usingCase is
23
24 begin
25 process (D0,D1,D2,D3,D4,D5,D6,D7,En)
26    variable D : STD_LOGIC_VECTOR (7 downto 0); -- A Temporary Signal Vector is
   created for Data input
27    begin                                       -- Since only one identifier is
   allowed in Case statement
28       D(0) := D0;
29       D(1) := D1;
30       D(2) := D2;
31       D(3) := D3;
32       D(4) := D4;
33       D(5) := D5;
34       D(6) := D6;
35       D(7) := D7;
36       if En = '1' then
37          case D is
38             when "00000001" => Y <= "000";
39             when "00000010" => Y <= "001";
40             when "00000100" => Y <= "010";
41             when "00001000" => Y <= "011";
42             when "00010000" => Y <= "100";
43             when "00100000" => Y <= "101";
44             when "01000000" => Y <= "110";
45             when "10000000" => Y <= "111";
46             when others => Y <= "000";
47          end case;
48       else
49          Y <= "000"; -- Output when Enable is low
50       end if;
51    end process;
52
53 end Behavioral;
54
```

```
1   -- Module Name:    EightTo3_ENCODER_usingIfElse - Behavioral
2   library IEEE;
3   use IEEE.STD_LOGIC_1164.ALL;
4
5   entity EightTo3_ENCODER_usingIfElse is
6       Port ( D0 : in  STD_LOGIC; --D0 to D7 is DATA input
7              D1 : in  STD_LOGIC;
8              D2 : in  STD_LOGIC;
9              D3 : in  STD_LOGIC;
10             D4 : in  STD_LOGIC;
11             D5 : in  STD_LOGIC;
12             D6 : in  STD_LOGIC;
13             D7 : in  STD_LOGIC;
14             Y : out  STD_LOGIC_VECTOR (2 downto 0); -- Y is output Vector, All are
    Active High
15             En : in  STD_LOGIC); --En is Enable
16  end EightTo3_ENCODER_usingIfElse;
17
18  architecture Behavioral of EightTo3_ENCODER_usingIfElse is
19  begin
20  process (D0,D1,D2,D3,D4,D5,D6,D7,En)
21     variable D : STD_LOGIC_VECTOR (7 downto 0); -- A Temporary Signal Vector is
    created for Select input
22     begin                                        -- This simplifies If - Else syntax
23        D(0) := D0;
24        D(1) := D1;
25        D(2) := D2;
26        D(3) := D3;
27        D(4) := D4;
28        D(5) := D5;
29        D(6) := D6;
30        D(7) := D7;
31        if En = '1' then
32           if (D = "00000001") then
33              Y <= "000";
34           elsif (D = "00000010") then
35              Y <= "001";
36           elsif (D = "00000100") then
37              Y <= "010";
38           elsif (D = "00001000") then
39              Y <= "011";
40           elsif (D = "00010000") then
41              Y <= "100";
42           elsif (D = "00100000") then
43              Y <= "101";
44           elsif (D = "01000000") then
45              Y <= "110";
46           elsif (D = "10000000") then
47              Y <= "111";
48           else
49              Y <= "000";
50           end if;
51        else
52           Y <= "000";
53        end if;
54     end process;
55  end Behavioral;
```

```
1  ----------------------------------------------------------------------------------
2  -- Company:
3  -- Engineer:
4  --
5  -- Create Date:    15:59:28 02/09/2019
6  -- Design Name:
7  -- Module Name:    OneTo8_DEMUX_usingCase - Behavioral
8  ----------------------------------------------------------------------------------
9  library IEEE;
10 use IEEE.STD_LOGIC_1164.ALL;
11
12 entity OneTo8_DEMUX_usingCase is
13     Port ( Inp : in  STD_LOGIC; --'Inp' is DATA input
14            s1 : in  STD_LOGIC; -- s1,s2,s3 are Select Inputs
15            s2 : in  STD_LOGIC;
16            s3 : in  STD_LOGIC;
17            En : in STD_LOGIC; --En is Enable
18            Y : out  STD_LOGIC_VECTOR (7 downto 0)); -- Y is output Vector, All are
   Active High
19 end OneTo8_DEMUX_usingCase;
20
21 architecture Behavioral of OneTo8_DEMUX_usingCase is
22
23 begin
24 process (Inp,s1,s2,s3,En)
25    variable s : STD_LOGIC_VECTOR (2 downto 0); -- A Temporary Signal Vector is
   created for Select input
26    begin                                       -- Since only one identifier is
   allowed in Case statement
27       s(0) := s1;
28       s(1) := s2;
29       s(2) := s3;
30       Y <= "00000000"; --Initialisation of Output
31       if En = '1' then
32          case s is
33             when "000" => Y(0) <= Inp;
34             when "001" => Y(1) <= Inp;
35             when "010" => Y(2) <= Inp; --Changing the required Bit Only
36             when "011" => Y(3) <= Inp;
37             when "100" => Y(4) <= Inp;
38             when "101" => Y(5) <= Inp;
39             when "110" => Y(6) <= Inp;
40             when others => Y(7) <= Inp;
41          end case;
42       else
43          Y <= "00000000";
44       end if;
45    end process;
46
47 end Behavioral;
48
49
```

```
1   ----------------------------------------------------------------------------------
2   -- Company:
3   -- Engineer:
4   --
5   -- Create Date:    16:14:30 02/09/2019
6   -- Design Name:
7   -- Module Name:    OneTo8_DEMUX_usingIfElse - Behavioral
8   ----------------------------------------------------------------------------------
9   library IEEE;
10  use IEEE.STD_LOGIC_1164.ALL;
11
12  entity OneTo8_DEMUX_usingIfElse is
13      Port ( Inp : in  STD_LOGIC; --'Inp' is DATA input
14             En : in  STD_LOGIC; --En is Enable
15             s1 : in  STD_LOGIC;
16             s2 : in  STD_LOGIC;-- s1,s2,s3 are Select Inputs
17             s3 : in  STD_LOGIC;
18             Y : out  STD_LOGIC_VECTOR (7 downto 0)); -- Y is output Vector, All are
    Active High
19  end OneTo8_DEMUX_usingIfElse;
20
21  architecture Behavioral of OneTo8_DEMUX_usingIfElse is
22
23  begin
24  process(Inp,En,s1,s2,s3)
25        begin
26        Y <= "00000000"; --Initialisation of Output
27           if En = '1' then
28              if (s3 = '0' and s2 = '0' and s1 = '0') then
29                 Y(0) <= Inp;
30              elsif (s3 = '0' and s2 = '0' and s1 = '1') then
31                 Y(1) <= Inp;
32              elsif (s3 = '0' and s2 = '1' and s1 = '0') then
33                 Y(2) <= Inp;
34              elsif (s3 = '0' and s2 = '1' and s1 = '1') then
35                 Y(3) <= Inp;
36              elsif (s3 = '1' and s2 = '0' and s1 = '0') then
37                 Y(4) <= Inp;
38              elsif (s3 = '1' and s2 = '0' and s1 = '1') then
39                 Y(5) <= Inp;
40              elsif (s3 = '1' and s2 = '1' and s1 = '0') then
41                 Y(6) <= Inp;
42              else
43                 Y(7) <= Inp;
44              end if;
45           else
46              Y <= "00000000";
47           end if;
48     end process;
49
50  end Behavioral;
51
52
```

```
1  ----------------------------------------------------------------------------------
2  -- Company:
3  -- Engineer:
4  --
5  -- Create Date:    16:28:34 02/09/2019
6  -- Design Name:
7  -- Module Name:    Threeto8_DECODER_usingCase - Behavioral
8  ----------------------------------------------------------------------------------
9  library IEEE;
10 use IEEE.STD_LOGIC_1164.ALL;
11
12 entity Threeto8_DECODER_usingCase is
13     Port ( D0 : in  STD_LOGIC; --D0,D1,D2 is DATA input
14            D1 : in  STD_LOGIC;
15            D2 : in  STD_LOGIC;
16            Y : out  STD_LOGIC_VECTOR (7 downto 0); -- Y is output Vector, All are
   Active High
17            En : in  STD_LOGIC); --En is Enable
18 end Threeto8_DECODER_usingCase;
19
20 architecture Behavioral of Threeto8_DECODER_usingCase is
21
22 begin
23 process (D0,D1,D2,En)
24    variable D : STD_LOGIC_VECTOR (2 downto 0); -- A Temporary Signal Vector is
   created for Data input
25    begin                                       -- Since only one identifier is
   allowed in Case statement
26       D(0) := D0;
27       D(1) := D1;
28       D(2) := D2;
29       if En = '1' then
30          case D is
31             when "000" => Y <= "00000001";
32             when "001" => Y <= "00000010";
33             when "010" => Y <= "00000100";
34             when "011" => Y <= "00001000";
35             when "100" => Y <= "00010000";
36             when "101" => Y <= "00100000";
37             when "110" => Y <= "01000000";
38             when others => Y <= "10000000";
39          end case;
40       else
41          Y <= "11111111"; -- Output when Enable is low
42       end if;
43    end process;
44
45 end Behavioral;
46
47
```

```
1   ----------------------------------------------------------------------------------
2   -- Company:
3   -- Engineer:
4   --
5   -- Create Date:    16:40:29 02/09/2019
6   -- Design Name:
7   -- Module Name:    Threeto8_DECODER_usingIfElse - Behavioral
8   ----------------------------------------------------------------------------------
9   library IEEE;
10  use IEEE.STD_LOGIC_1164.ALL;
11
12  entity Threeto8_DECODER_usingIfElse is
13      Port ( D0 : in  STD_LOGIC; --D0,D1,D2 is DATA input
14             D1 : in  STD_LOGIC;
15             D2 : in  STD_LOGIC;
16             En : in  STD_LOGIC; --En is Enable
17             Y : out  STD_LOGIC_VECTOR (7 downto 0)); -- Y is output Vector, All are
    Active High
18  end Threeto8_DECODER_usingIfElse;
19
20  architecture Behavioral of Threeto8_DECODER_usingIfElse is
21
22  begin
23  process(En,D0,D1,D2)
24        begin
25           if En = '1' then
26              if (D2 = '0' and D1 = '0' and D0 = '0') then
27                 Y <= "00000001";
28              elsif (D2 = '0' and D1 = '0' and D0 = '1') then
29                 Y <= "00000010";
30              elsif (D2 = '0' and D1 = '1' and D0 = '0') then
31                 Y <= "00000100";
32              elsif (D2 = '0' and D1 = '1' and D0 = '1') then
33                 Y <= "00001000";
34              elsif (D2 = '1' and D1 = '0' and D0 = '0') then
35                 Y <= "00010000";
36              elsif (D2 = '1' and D1 = '0' and D0 = '1') then
37                 Y <= "00100000";
38              elsif (D2 = '1' and D1 = '1' and D0 = '0') then
39                 Y <= "01000000";
40              else
41                 Y <= "10000000";
42              end if;
43           else
44              Y <= "11111111"; -- Output when Enable is low
45           end if;
46     end process;
47  end Behavioral;
```