

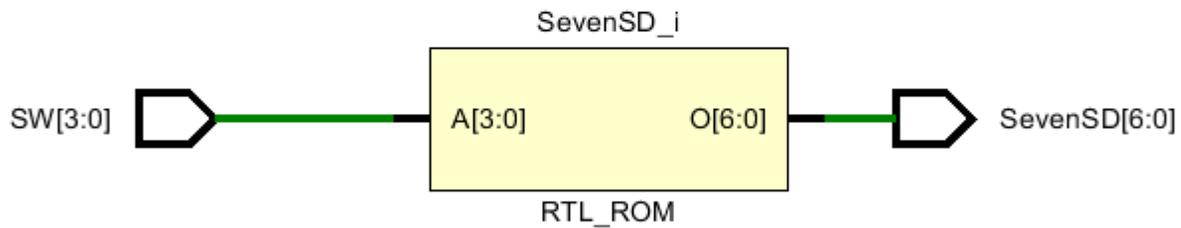
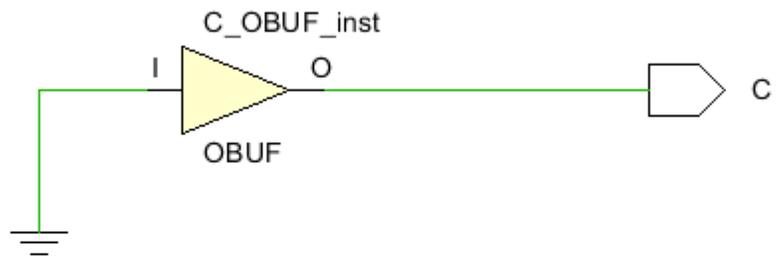
C:/Jah/EEL470/Assignment 3-Part1 project/Assignment 3-Part1 project.srscs/sources_1/new/sevenSementDis.vhd



```
2  -- Company:
3  -- Engineer:
4  --
5  -- Create Date: 06/08/2025 10:16:38 AM
6  -- Design Name:
7  -- Module Name: sevenSementDis - Behavioral
8  -- Project Name:
9  -- Target Devices:
10 -- Tool Versions:
11 -- Description:
12 --
13 -- Dependencies:
14 --
15 -- Revision:
16 -- Revision 0.01 - File Created
17 -- Additional Comments:
18 --
19 -----
20
21
22 library IEEE;
23 use IEEE.STD_LOGIC_1164.ALL;
24
25 entity sevenSementDis is
26     Port ( SW : in STD_LOGIC_VECTOR (3 downto 0);
27           C : out STD_LOGIC;
28           SevenSD : out STD_LOGIC_VECTOR (6 downto 0));
29 end sevenSementDis;
30
31 architecture Behavioral of sevenSementDis is
32
33 begin
34 C <= '0';
35 process (SW)
36 begin
37 case (SW) is
38     when "0000" => SevenSD <= "0111111";
39     when "0001" => SevenSD <= "0000110";
40     when "0010" => SevenSD <= "1011011";
41     when "0011" => SevenSD <= "1001111";
42     when "0100" => SevenSD <= "1100110";
43     when "0101" => SevenSD <= "1101101";
44     when "0110" => SevenSD <= "1111101";
45     when "0111" => SevenSD <= "0000111";
46     when "1000" => SevenSD <= "1111111";
47     when "1001" => SevenSD <= "1101111";
48     when others => SevenSD <= "1000000";
49 end case;
50 end process;
51 end Behavioral;
52
```



```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3
4  entity RippleCarryAdder4 is
5      Port ( A : in  STD_LOGIC_VECTOR(3 downto 0);
6            B : in  STD_LOGIC_VECTOR(3 downto 0);
7            Cin : in  STD_LOGIC;
8            Sum : out  STD_LOGIC_VECTOR(3 downto 0);
9            Cout : out  STD_LOGIC );
10 end RippleCarryAdder4;
11
12 architecture Structural of RippleCarryAdder4 is
13
14     component FullAdder
15         Port ( A : in  STD_LOGIC;
16               B : in  STD_LOGIC;
17               Cin : in  STD_LOGIC;
18               Sum : out  STD_LOGIC;
19               Cout : out  STD_LOGIC );
20     end component;
21
22     signal C : STD_LOGIC_VECTOR(3 downto 0);
23
24 begin
25     FA0: FullAdder port map(A => A(0), B => B(0), Cin => Cin,      Sum => Sum(0), Cout => C(0));
26     FA1: FullAdder port map(A => A(1), B => B(1), Cin => C(0),    Sum => Sum(1), Cout => C(1));
27     FA2: FullAdder port map(A => A(2), B => B(2), Cin => C(1),    Sum => Sum(2), Cout => C(2));
28     FA3: FullAdder port map(A => A(3), B => B(3), Cin => C(2),    Sum => Sum(3), Cout => Cout);
29
30 end Structural;
31
```





```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3
4  entity FullAdder is
5      Port ( A      : in  STD_LOGIC;
6            B      : in  STD_LOGIC;
7            Cin    : in  STD_LOGIC;
8            Sum    : out STD_LOGIC;
9            Cout   : out STD_LOGIC );
10 end FullAdder;
11
12 architecture Behavioral of FullAdder is
13
14     component HalfAdder
15         Port ( A : in  STD_LOGIC;
16               B : in  STD_LOGIC;
17               Sum : out STD_LOGIC;
18               Cout : out STD_LOGIC );
19     end component;
20
21     signal S1, C1, C2 : STD_LOGIC;
22
23 begin
24     HA1: HalfAdder port map(A => A, B => B, Sum => S1, Cout => C1);
25     HA2: HalfAdder port map(A => S1, B => Cin, Sum => Sum, Cout => C2);
26     Cout <= C1 OR C2;
27 end Behavioral;
28
```



```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3  use IEEE.STD_LOGIC_ARITH.ALL;
4  use IEEE.STD_LOGIC_UNSIGNED.ALL;
5
6  entity keypad_decoder is
7      Port (
8          row : in STD_LOGIC_VECTOR (2 downto 0);
9          col : in STD_LOGIC_VECTOR (3 downto 0);
10         output : out STD_LOGIC_VECTOR (3 downto 0)
11     );
12 end keypad_decoder;
13
14 architecture Behavioral of keypad_decoder is
15 begin
16     process(row, col)
17     begin
18         case (row & col) is
19             when "0001" => output <= "0000";
20             when "0010" => output <= "0001";
21             when "0011" => output <= "0010";
22             when "0100" => output <= "0011";
23             when "0101" => output <= "0100";
24             when "0110" => output <= "0101";
25             when "0111" => output <= "0110";
26             when "1000" => output <= "0111";
27             when "1001" => output <= "1000";
28             when "1010" => output <= "1001";
29             when "1011" => output <= "0000";
30             when "1100" => output <= "0000";
31             when others => output <= "0000";
32         end case;
33     end process;
34 end Behavioral;
```