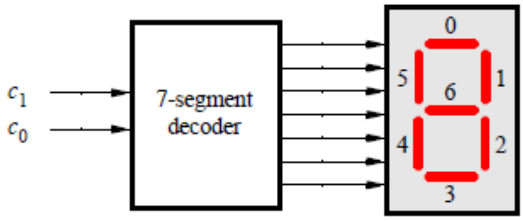
EL323//El324 Digital System Laboratory II

LAB 3 : Design using VHDL

Objective:

1. Can use the Quartus II and create the new project using VHDL
2. To know the assign pin of DE-0 in Quartus.
3. Design using VHDL

<u>LAB 3.1</u> (Altera Document: Laboratory Exercise 1, Part IV)



1. Create new Project name Lab3_1.
2. Create new entity for 7-segment display in this below

$c_1 c_0$	Character
00	d
01	E
10	0
11	

Table 1. Character codes.

- Connect the c_1 c_0 inputs to switches SW_1 and SW_0
- Connect the outputs of the decoder to the HEX_0 display on the DE_0 board. The segments in this display are called HEX_{00}, HEX_{01}, …, HEX_{06}, You should declare the 7-bit port

HEX0: OUT STD_LOGIC_VECTOR(0 TO 6);

3. Use the DE0 User Manual to define the DE0 pin.
4. Compile the code and program to DE0.

Lab3.1 Code.

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;

-- Simple module that connects the SW switches to the LEDG lights
ENTITY Lab3_1 IS
        PORT ( SW : IN STD_LOGIC_VECTOR(1 DOWNTO 0);
             HEX0_D : OUT STD_LOGIC_VECTOR(6 DOWNTO 0) );
END Lab3_1;

ARCHITECTURE Behavior OF Lab3_1 IS
        SIGNAL s : STD_LOGIC_VECTOR (1 DOWNTO 0);
BEGIN
        s <= SW(1 DOWNTO 0);
-- Recall that 7-Segments are active at LOW-level logic --
        -- character 'd'  0100001  SW 00
        -- character 'E'   _____   SW 01
        -- character '0'   _____   SW 10
        -- character ' '   _____   SW 11
        --
        --       0
        --      ---
        --     |   |
        --   5|   |1
        --     | 6 |
        --      ---
        --     |   |
        --   4|   |2
        --     |   |
        --      ---
        --       3
        --
        -- the following equations describe display functions in (inverted)
        -- cannonical SOP form

        HEX0_D(0) <= (s(1) AND  s(0)) OR ((NOT s(1)) AND (NOT s(0)));
        HEX0_D(1) <= _____
        HEX0_D(2) <= _____
        HEX0_D(3) <= _____
        HEX0_D(4) <= _____
        HEX0_D(5) <= _____
        HEX0_D(6) <= _____

        ---- some additional VHDL code as necessary ----
END Behavior;
```

Lab3.1 Answer Sheet

SW		Display 7 Segment	
SW1	SW0	Character	bits
0	0	‘d’	_ _ _ _ _ _ _
0	1	‘E’	_ _ _ _ _ _ _
1	0	‘0’	_ _ _ _ _ _ _
1	1	‘ ’	_ _ _ _ _ _ _

HEX0_D(0) <= (s(1) AND s(0)) OR ((NOT s(1)) AND (NOT s(0)));

HEX0_D (1) <= _____

HEX0_D (2) <= _____

HEX0_D (3) <= _____

HEX0_D (4) <= _____

HEX0_D (5) <= _____

HEX0_D (6) <= _____

Conclusion

……

……

……

……

……

……

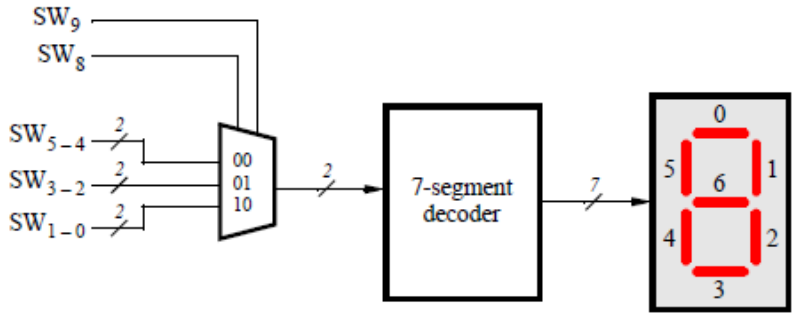
……

……

……

……

LAB 3.2 (Altera Document: Laboratory Exercise 1, Part V)



1. Create new Project name Lab3_2.
2. Create new entity for 7-segment display in this below

SW_9 SW_8	Character pattern		
00	d	E	0
01	E	0	d
10	0	d	E

Table 2. Rotating the word DE0 on three displays

- Connect the inputs switches Using SW9 and SW8 to select the display output
- SW5-0 is required to produce the patterns of characters as same as the Lab3_1 and select to show like table 2.

3. Use the DE0 User Manual to define the DE0 pin.
4. Compile the code and program to DE0.

Lab3.2 Code.

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY Lab3_2 IS
	PORT ( SW : IN STD_LOGIC_VECTOR(9 DOWNTO 0);
			HEX0_D : OUT STD_LOGIC_VECTOR(6 DOWNTO 0);
			HEX1_D : OUT STD_LOGIC_VECTOR(6 DOWNTO 0);
			HEX2_D : OUT STD_LOGIC_VECTOR(6 DOWNTO 0);
			HEX3_D : OUT STD_LOGIC_VECTOR(6 DOWNTO 0);
			LEDG : OUT STD_LOGIC_VECTOR(9 DOWNTO 0));
END Lab3_2;
ARCHITECTURE Behavior OF Lab3_2 IS
	COMPONENT mux_2bit_3to1
		PORT ( S, U, V, W : IN STD_LOGIC_VECTOR(1 DOWNTO 0);
				M : OUT STD_LOGIC_VECTOR(1 DOWNTO 0));
	END COMPONENT;

	COMPONENT char_7seg
		PORT ( C : IN STD_LOGIC_VECTOR(1 DOWNTO 0);
			Display : OUT STD_LOGIC_VECTOR(6 DOWNTO 0));
	END COMPONENT;

	SIGNAL M_0, M_1, M_2, M_3 : STD_LOGIC_VECTOR(1 DOWNTO 0);
	SIGNAL Blank : STD_LOGIC_VECTOR(1 DOWNTO 0);

BEGIN
	LEDG(9 DOWNTO 8) <= SW(9 DOWNTO 8);
	Blank <= "11";
	-- first 7-segment --
	LEDG(3 DOWNTO 2) <= M_0;
	M3: mux_2bit_3to1 PORT MAP (SW(9 DOWNTO 8), SW(5 DOWNTO 4), SW(3 DOWNTO 2),
SW(1 DOWNTO 0), M_3);
	H3: char_7seg PORT MAP (M_3, HEX3_D);
	-- second 7-segment --
	M2: mux_2bit_3to1 PORT MAP (SW(9 DOWNTO 8), SW(3 DOWNTO 2), SW(1 DOWNTO 0),
SW(5 DOWNTO 4), M_2);
	H2: char_7seg PORT MAP (M_2, HEX2_D);
	-- third 7-segment --
	M1: mux_2bit_3to1 PORT MAP (SW(9 DOWNTO 8), SW(1 DOWNTO 0), SW(5 DOWNTO 4),
SW(3 DOWNTO 2), M_1);
	H1: char_7seg PORT MAP (M_1, HEX1_D);
	-- fourth 7-segment --
	H0: char_7seg PORT MAP (Blank, HEX0_D);
END Behavior;
```

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
-- implements a 2-bit wide 3-to-1 multiplexer
ENTITY mux_2bit_3to1 IS
        PORT ( S, U, V, W : IN STD_LOGIC_VECTOR(1 DOWNTO 0);
                        M : OUT STD_LOGIC_VECTOR(1 DOWNTO 0));
END mux_2bit_3to1;
ARCHITECTURE Behavior OF mux_2bit_3to1 IS
        SIGNAL m_0, m_1 : STD_LOGIC;
BEGIN
        m_0 <= _____________________________________________
        M(0) <=_____________________________________________-- M(0)
        m_1 <= _____________________________________________
        M(1) <=_____________________________________________ -- M(1)
END Behavior;

LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY char_7seg IS
        PORT ( C : IN STD_LOGIC_VECTOR(1 DOWNTO 0);
                        Display : OUT STD_LOGIC_VECTOR(6 DOWNTO 0));
END char_7seg;
ARCHITECTURE Behavior OF char_7seg IS
        --
        --       0
        --      ---
        --     |    |
        --   5|    |1
        --     | 6  |
        --      ---
        --     |    |
        --   4|    |2
        --     |    |
        --       ---
        --       3
        --

BEGIN
        Display(0) <= (NOT(C(1)) AND NOT(C(0))) OR (C(1) AND C(0));
        Display(1) <= _____________________________________________
        Display(2) <= _____________________________________________
        Display(3) <= _____________________________________________
        Display(4) <= _____________________________________________
        Display(5) <= _____________________________________________
        Display(6) <= _____________________________________________
END Behavior;
```

Lab3.2 Answer Sheet

The Mux 2bit 3 to 1 :

m_0 <= _____

M(0) <=_____-- M(0)

m_1 <= _____

M(1) <=_____ -- M(1)

SW		Display 7 Segment	
SW1,3,5	SW0,2,4	Character	bits
0	0	‘d’	_ _ _ _ _ _ _
0	1	‘E’	_ _ _ _ _ _ _
1	0	‘0’	_ _ _ _ _ _ _
1	1	‘ ’	_ _ _ _ _ _ _

Display(0) <= (NOT(C(1)) AND NOT(C(0))) OR (C(1) AND C(0));

Display(1) <= _____

Display(2) <= _____

Display(3) <= _____

Display(4) <= _____

Display(5) <= _____

Display(6) <= _____

SW										Display 7 Segment
SW9	**SW8**	**SW7**	**SW6**	**SW5**	**SW4**	**SW3**	**SW2**	**SW1**	**SW0**	**Character**
0	0	NC	NC	1	0	0	1	0	0	_ _ _ _
0	1	NC	NC	1	0	0	1	0	0	_ _ _ _
1	0	NC	NC	1	0	0	1	0	0	_ _ _ _
1	1	NC	NC	1	0	0	1	0	0	_ _ _ _
0	0	NC	NC	1	0	0	0	0	1	_ _ _ _
0	1	NC	NC	1	0	0	0	0	1	_ _ _ _
1	0	NC	NC	1	0	0	0	0	1	_ _ _ _
1	1	NC	NC	1	0	0	0	0	1	_ _ _ _
0	0	NC	NC	0	0	1	0	0	1	_ _ _ _
0	1	NC	NC	0	0	1	0	0	1	_ _ _ _
1	0	NC	NC	0	0	1	0	0	1	_ _ _ _
1	1	NC	NC	0	0	1	0	0	1	_ _ _ _
0	0	NC	NC	0	1	1	0	0	0	_ _ _ _
0	1	NC	NC	0	1	1	0	0	0	_ _ _ _
1	0	NC	NC	0	1	1	0	0	0	_ _ _ _
1	1	NC	NC	0	1	1	0	0	0	_ _ _ _

Conclusion

……

……

……

……

……

……

……

……

……

……

Homework

1. Follow the Lab3_1 change character to this table. And Write the VHDL Code.

SW		Display 7 Segment	
SW1	SW0	Character	bits
0	0	‘A’	_ _ _ _ _ _ _
0	1	‘3’	_ _ _ _ _ _ _
1	0	‘5’	_ _ _ _ _ _ _
1	1	‘C‘	_ _ _ _ _ _ _