Link to my github

Lab 04-segment

1. Preparation tasks

Decoder truth table for common anode 7-segment display

Hex	Inputs	A	B	C	D	E	F	G
0	0000	0	0	0	0	0	0	1
1	0001	1	0	0	1	1	1	1
2	0010	0	0	1	0	0	1	0
3	0011	0	0	0	0	1	1	0
4	0100	1	0	0	1	1	0	0
5	0101	0	1	0	0	1	0	0
6	0110	0	1	0	0	0	0	0
7	0111	0	0	0	1	1	1	1
8	1000	0	0	0	0	0	0	0
9	1001	0	0	0	0	1	0	0
A	1010	0	0	0	1	0	0	0
b	1011	1	1	0	0	0	0	0
C	1100	0	1	1	0	0	0	1
d	1101	1	0	0	0	0	1	0
E	1110	0	1	1	0	0	0	0
F	1111	0	1	1	1	0	0	0

Table with connection of 7-segment displays on Nexys A7 board

SW	AN	SW PIN	AN PIN	7-seg LED	7-seg PIN
SW0	AN7	J15	U13	CA	T10

SW	AN	SW PIN	AN PIN	7-seg LED	7-seg PIN
SW1	AN6	L16	K2	CB	R10
SW2	AN5	M13	T14	CC	K16
SW3	AN4	R15	P14	CD	K13
SW4	AN3	R17	J14	CE	P15
SW5	AN2	T18	T9	CF	T11
SW6	AN1	U18	J18	CG	L18
SW7	AN0	R13	J17	DP	H15

2. Seven-segment display decoder

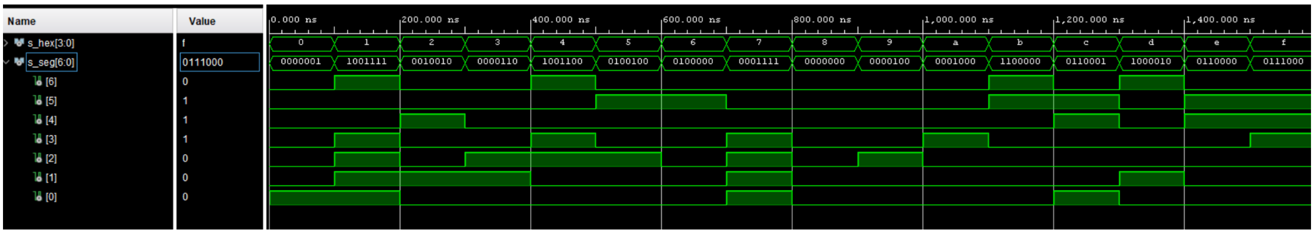
VHDL architecture from source file hex_7seg.vhd

```
architecture Behavioral of hex_7seg is
begin
    p_7seg_decoder : process(hex_i)
    begin
        case hex_i is
            when "0000" =>
                seg_o <= "0000001";  -- 0
            when "0001" =>
                seg_o <= "1001111";  -- 1
            when "0010" =>
                seg_o <= "0010010";  -- 2
            when "0011" =>
                seg_o <= "0000110";  -- 3
            when "0100" =>
                seg_o <= "1001100";  -- 4
            when "0101" =>
                seg_o <= "0100100";  -- 5
            when "0110" =>
                seg_o <= "0100000";  -- 6
            when "0111" =>
                seg_o <= "0001111";  -- 7
            when "1000" =>
                seg_o <= "0000000";  -- 8
            when "1001" =>
                seg_o <= "0000100";  -- 9
            when "1010" =>
                seg_o <= "0001000";  -- A
            when "1011" =>
                seg_o <= "1100000";  -- b
            when "1100" =>
                seg_o <= "0110001";  -- C
            when "1101" =>
                seg_o <= "1000010";  -- d
            when "1110" =>
                seg_o <= "0110000";  -- E
            when others =>
                seg_o <= "0111000";  -- F
        end case;
    end process p_7seg_decoder;
```

VHDL stimulus process from testbench file tb_hex_7seg.vhd

```
p_7seg_decoder : process
    begin
        report "Stimulus process started" severity note;
        s_hex <= "0000"; wait for 100 ns;
        s_hex <= "0001"; wait for 100 ns;
        s_hex <= "0010"; wait for 100 ns;
        s_hex <= "0011"; wait for 100 ns;
        s_hex <= "0100"; wait for 100 ns;
        s_hex <= "0101"; wait for 100 ns;
        s_hex <= "0110"; wait for 100 ns;
        s_hex <= "0111"; wait for 100 ns;
        s_hex <= "1000"; wait for 100 ns;
        s_hex <= "1001"; wait for 100 ns;
        s_hex <= "1010"; wait for 100 ns;
        s_hex <= "1011"; wait for 100 ns;
        s_hex <= "1100"; wait for 100 ns;
        s_hex <= "1101"; wait for 100 ns;
        s_hex <= "1110"; wait for 100 ns;
        s_hex <= "1111";
        report "Stimulus process finished" severity note;
        wait;
    end process p_7seg_decoder;
```

Screenshot with simulated time waveforms



VHDL code with 7-segment module instantiation from source file top.vhd

```
hex2seg : entity work.hex_7seg
        port map(
            hex_i    => SW,
            seg_o(6) => CA,   -- 6 MSB
            seg_o(5) => CB,
            seg_o(4) => CC,
            seg_o(3) => CD,
            seg_o(2) => CE,
            seg_o(1) => CF,
            seg_o(0) => CG
        );
```

3. LED(7:4) indicators

Truth table for LEDs(7:4)

Hex	Inputs	LED4	LED5	LED6	LED7
0	0000	1	0	0	0
1	0001	0	0	1	1
2	0010	0	0	0	1
3	0011	0	0	1	0
4	0100	0	0	0	1
5	0101	0	0	1	0
6	0110	0	0	0	0
7	0111	0	0	1	0
8	1000	0	0	0	1
9	1001	0	0	1	0
A	1010	0	1	0	0
b	1011	0	1	1	0
C	1100	0	1	0	0
d	1101	0	1	1	0
E	1110	0	1	0	0
F	1111	0	1	1	0

VHDL code for LEDs(7:4)

```
LED(3 downto 0) <= SW;
```

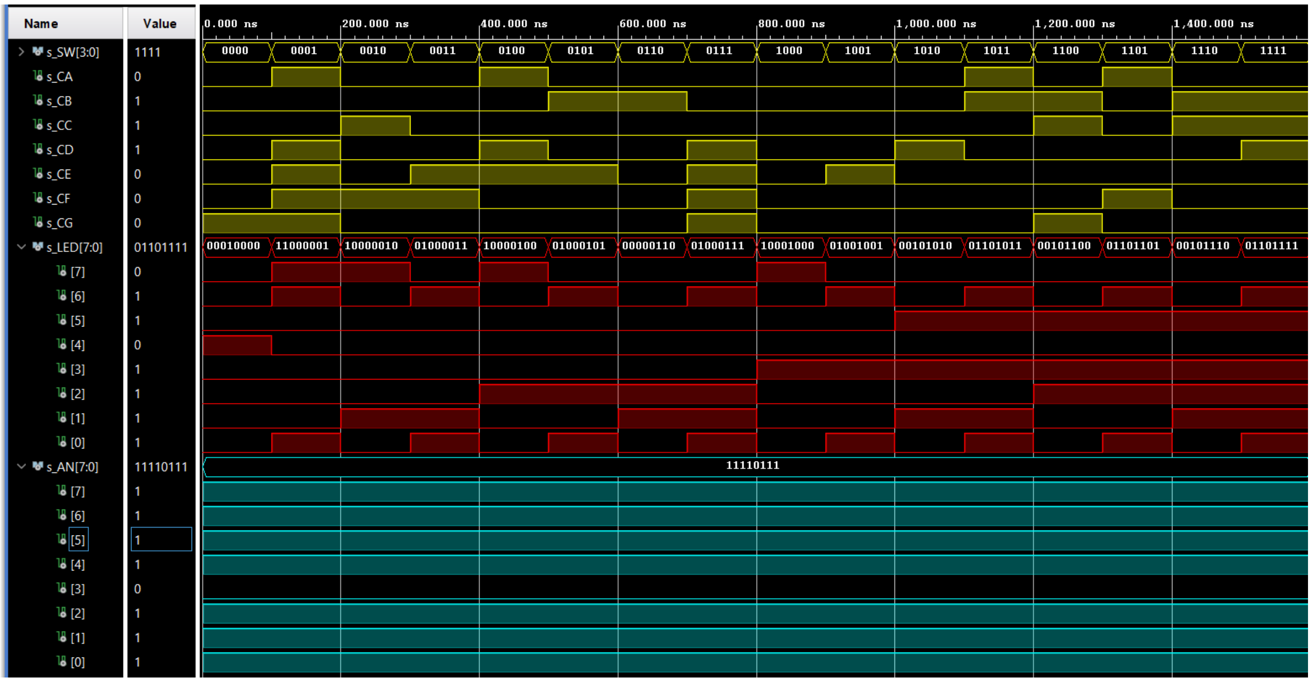
```
LED (4) <=  '1' when (SW = "0000") else '0'; -- 0

LED (5) <=  '1' when (SW > "1001") else '0'; -- > 9

LED (6) <=  '1' when (SW(0) = '1') else '0'; -- 1, 3, 5, 7, 9...

LED (7) <=  '1' when (SW = "0001") else -- 1, 2, 4, 8...
            '1' when (SW = "0010") else
            '1' when (SW = "0100") else
            '1' when (SW = "1000") else '0';
```

Screenshot with simulated time waveforms



Name
Value
s_SW[3:0] 1111
s_CA 0
s_CB 1
s_CC 1
s_CD 1
s_CE 0
s_CF 0
s_CG 0
s_LED[7:0] 01101111
[7] 0
[6] 1
[5] 1
[4] 0
[3] 1
[2] 1
[1] 1
[0] 1
s_AN[7:0] 11110111
[7] 1
[6] 1
[5] 1
[4] 1
[3] 0
[2] 1
[1] 1
[0] 1
0.000 ns
200.000 ns
400.000 ns
600.000 ns
800.000 ns
1,000.000 ns
1,200.000 ns
1,400.000 ns
0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111
00010000
11000001
10000010
01000011
10000100
01000101
00000110
01000111
10001000
01001001
00101010
01101011
00101100
01101101
00101110
01101111
11110111