Cvičenie 4

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

Hex	Inputs	A	B	C	D	E	F	G
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	0	0	0	1	1	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

Name	Port	Function
T10	CA	A
R10	CB	B
K16	CC	C
K13	CD	D
P15	CE	E
T11	CF	F
L18	CG	G
J17	AN[0]	KAT 1
J18	AN[1]	KAT 2
T9	AN[2]	KAT 3
J14	AN[3]	KAT 4
P14	AN[4]	KAT 5
T14	AN[5]	KAT 6
K2	AN[6]	KAT 7
U13	AN[7]	KAT 8

Seven-segment display decoder

VHDL architecture from source file hex_7seg.vhd

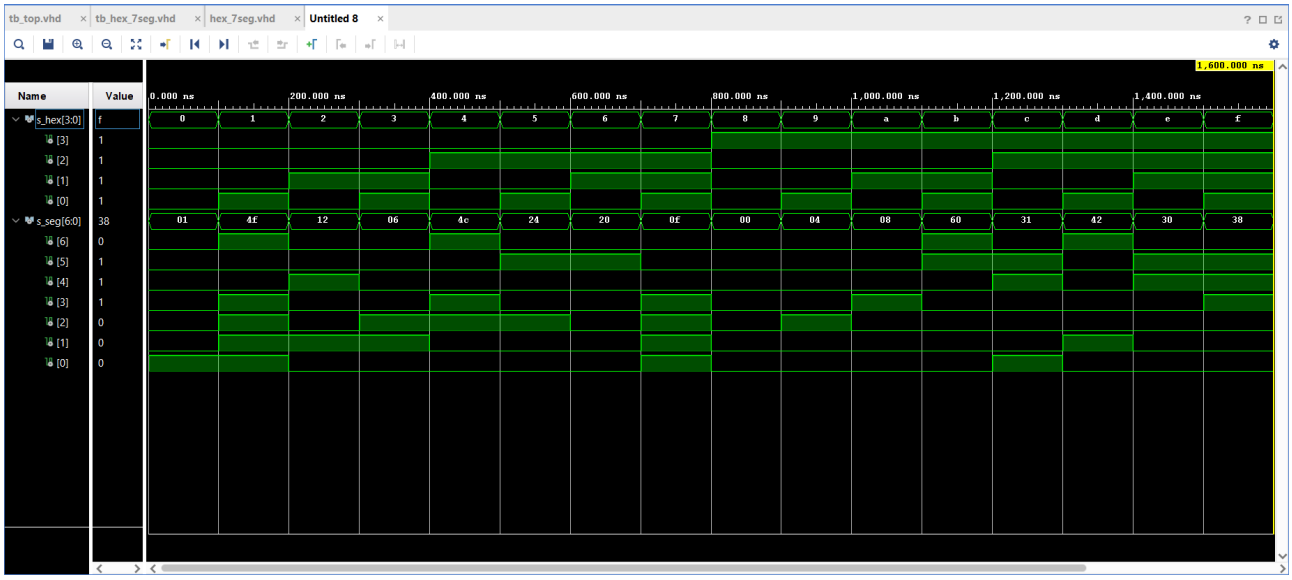
```
architecture behavioral of hex_7seg is
begin

    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;

end architecture behavioral;
```

VHDL stimulus process from testbench file tb_hex_7seg.vhd

```
p_stimulus : 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"; wait for 100 ns;

        report "Stimulus process finished" severity note;
        wait;
        end process p_stimulus;
```

Simulated time waveforms



VHDL code from source file top.vhd

```
architecture Behavioral of top1 is
begin

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

    AN <= "11110111";
    LED(3 downto 0) <= SW;
    LED(4)  <= '1' when (SW = "0000") else '0';
    LED(5)  <= '1' when (SW > "1001") else '0';
    LED(6)  <= '1' when (SW = "0001" or SW = "0011" or SW = "0101" or SW = "0111" or
    LED(7)  <= '1' when (SW = "0001" or SW = "0010" or SW = "0100" or SW = "1000") e

end Behavioral;
```

VHDL stimulus process from testbench file tb_top.vhd

```
    p_stimulus : 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";
        wait for 100 ns;

        report "Stimulus process finished" severity note;
        wait;
    end process p_stimulus;

end architecture Behavoral;
```

LED(7:4) indicators

Truth table

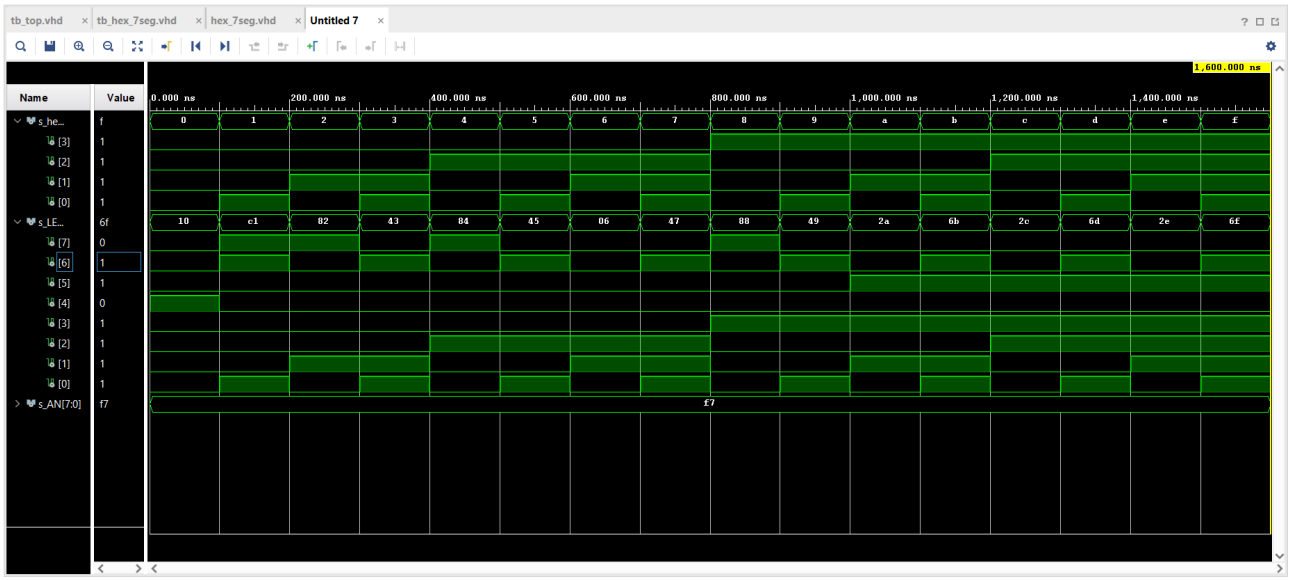
Hex	Inputs	LED4	LED5	LED6	LED7
0	0000	1	0	0	0

Hex	Inputs	LED4	LED5	LED6	LED7
1	0001	0	0	1	0
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	1
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	0	0
C	1100	0	1	0	0
d	1101	0	1	0	0
E	1110	0	1	0	0
F	1111	0	1	0	0

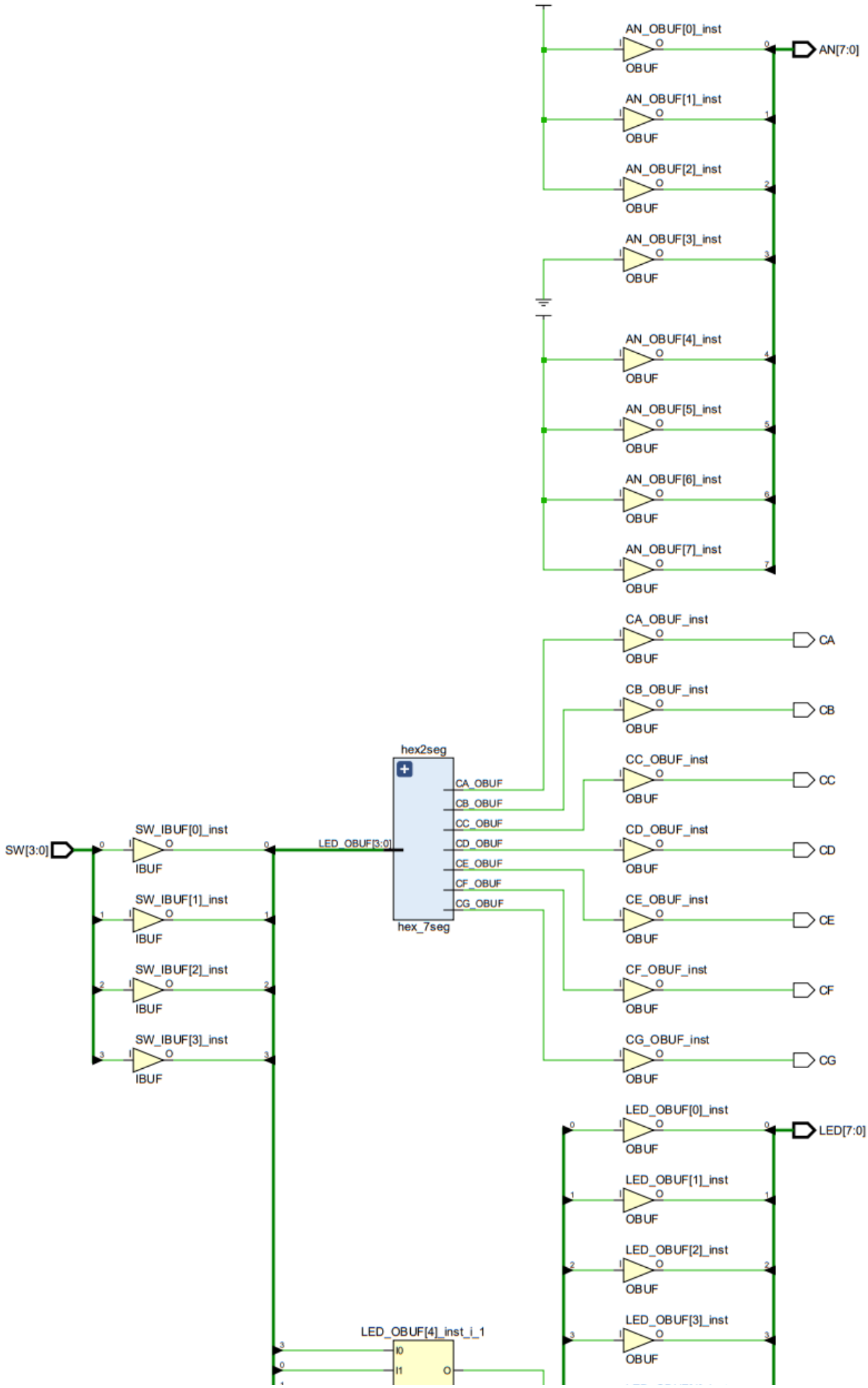
Code for LEDs(7:4)

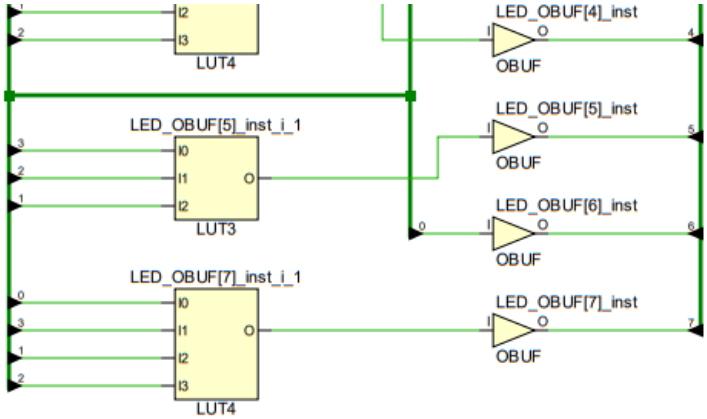
```
LED(4)  <= '1' when (SW = "0000") else '0';
    LED(5)  <= '1' when (SW > "1001") else '0';
    LED(6)  <= '1' when (SW = "0001" or SW = "0011" or SW = "0101" or SW = "0111" or
    LED(7)  <= '1' when (SW = "0001" or SW = "0010" or SW = "0100" or SW = "1000") e
```

Simulated time waveforms for tb_top


tb_top.vhd
tb_hex_7seg.vhd
hex_7seg.vhd
Untitled 7
Name
Value
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
1,600.000 ns
s_he...
s_LE...
s_AN[7:0]
f7

Schematic


AN_OBUF[0]_inst
OBUF
AN[7:0]
AN_OBUF[1]_inst
OBUF
AN_OBUF[2]_inst
OBUF
AN_OBUF[3]_inst
OBUF
AN_OBUF[4]_inst
OBUF
AN_OBUF[5]_inst
OBUF
AN_OBUF[6]_inst
OBUF
AN_OBUF[7]_inst
OBUF
CA_OBUF_inst
OBUF
CA
CB_OBUF_inst
OBUF
CB
CC_OBUF_inst
OBUF
CC
CD_OBUF_inst
OBUF
CD
CE_OBUF_inst
OBUF
CE
CF_OBUF_inst
OBUF
CF
CG_OBUF_inst
OBUF
CG
hex2seg
CA_OBUF
CB_OBUF
CC_OBUF
CD_OBUF
CE_OBUF
CF_OBUF
CG_OBUF
hex_7seg
LED_OBUF[3:0]
SW[3:0]
SW_IBUF[0]_inst
IBUF
SW_IBUF[1]_inst
IBUF
SW_IBUF[2]_inst
IBUF
SW_IBUF[3]_inst
IBUF
LED_OBUF[0]_inst
OBUF
LED[7:0]
LED_OBUF[1]_inst
OBUF
LED_OBUF[2]_inst
OBUF
LED_OBUF[3]_inst
OBUF
LED_OBUF[4]_inst_i_1
I0
I1
O


LED_OBUF[4]_inst
OBUF
LUT4
LED_OBUF[5]_inst_i_1
LED_OBUF[5]_inst
OBUF
LUT3
LED_OBUF[6]_inst
OBUF
LED_OBUF[7]_inst_i_1
LED_OBUF[7]_inst
OBUF
LUT4