Cvičenie 5

Truth table for common anode 7-segment display

Name	Port	Function
J15	SW	L=0V, H=3,3V
P17	RESET	L=0V, H=3,3V
T10	CA	A
R10	CB	B
K16	CC	C
K13	CD	D
P15	CE	E
T11	CF	F
L18	CG	G

Name	Port	Function
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

Table with calculated values

Time interval	Number of clk periods	Number of clk periods in hex	Number of clk periods in binary
2 ms	200 000	`x"3_0d40"`	`b"0011_0000_1101_0100_0000"`
4 ms	400 000	`x"6_1a80"`	`b"0110_0001_1010_1000_0000"`
10 ms	1 000 000	`x"f_4240"`	`b"1111_0100_0010_0100_0000"`
250 ms	25 000 000	`x"17d_7840"`	`b"0001_0111_1101_0111_1000_0100_0000"`
500 ms	50 000 000	`x"2fa_f080"`	`b"0010_1111_1010_1111_0000_1000_0000"`
1 sec	100 000 000	`x"5f5_e100"`	`b"0101_1111_0101_1110_0001_0000_0000"`

Bidirectional counter

Listing of VHDL code of the process p_cnt_up_down

```
    p_cnt_up_down : process(clk)
begin
    if rising_edge(clk) then
```

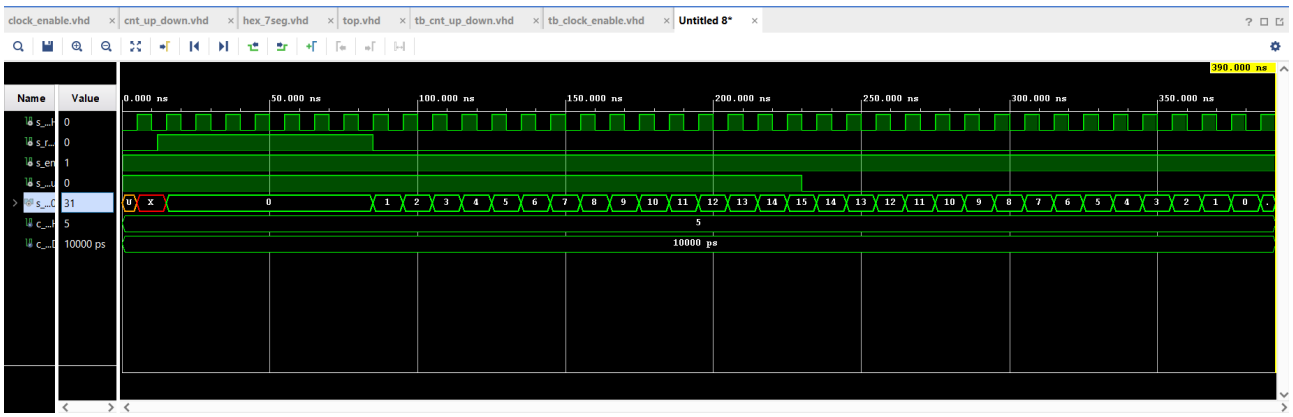
```
            if (reset = '1') then
                s_cnt_local <= (others => '0');

            elsif (en_i = '1') then

            if (cnt_up_i = '1') then
                s_cnt_local <= s_cnt_local + 1;

            elsif (cnt_up_i = '0') then
                s_cnt_local <= s_cnt_local - 1;

             end if;
            end if;
        end if;
    end process p_cnt_up_down;
```

Listing of VHDL reset and stimulus processes from testbench file

```
    p_reset_gen : process
    begin
        s_reset <= '0';
        wait for 12 ns;

        -- Reset activated
        s_reset <= '1';
        wait for 73 ns;

        s_reset <= '0';
        wait;
    end process p_reset_gen;

    p_stimulus : process
    begin
        report "Stimulus process started" severity note;

        -- Enable counting
        s_en     <= '1';

        -- Change counter direction
        s_cnt_up <= '1';
        wait for 230 ns;
        s_cnt_up <= '0';
        wait for 230 ns;

        -- Disable counting
        s_en     <= '0';

        report "Stimulus process finished" severity note;
```

```
      wait;
   end process p_stimulus;
```

Simulated time waveforms



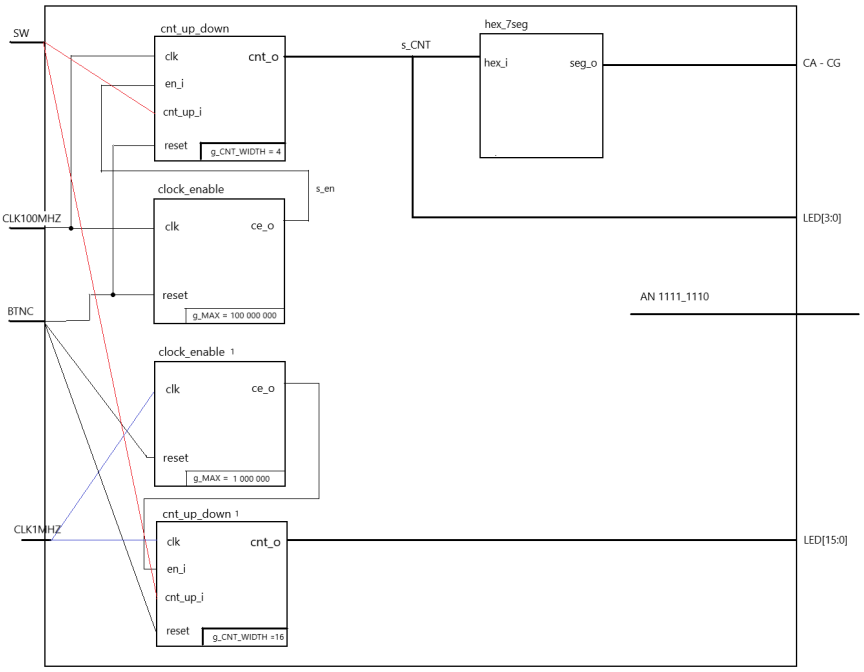
Top level

Listing of VHDL code from source file top.vhd with all instantiations for the 4-bit bidirectional counter

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity top is
    Port ( CLK100MHZ : in STD_LOGIC;
           BTNC : in STD_LOGIC;
           SW : in STD_LOGIC_VECTOR (1-1 downto 0);
           LED : out STD_LOGIC_VECTOR (4-1 downto 0);
           CA : out STD_LOGIC;
           CB : out STD_LOGIC;
           CC : out STD_LOGIC;
           CD : out STD_LOGIC;
           CE : out STD_LOGIC;
           CF : out STD_LOGIC;
           CG : out STD_LOGIC;
           AN : out STD_LOGIC_VECTOR (8 - 1 downto 0));
end top;

architecture Behavioral of top is

    -- Internal clock enable
    signal s_en  : std_logic;
    -- Internal counter
    signal s_cnt : std_logic_vector(16 - 1 downto 0);

begin
```

```
    clk_en0 : entity work.clock_enable
        generic map(
            g_MAX => 100000000
        )
        port map(
            clk     => CLK100MHZ,
            reset   => BTNC,
            ce_o    => s_en
        );

    bin_cnt0 : entity work.cnt_up_down
        generic map(
            g_CNT_WIDTH => 4
        )
        port map(
             clk     => CLK100MHZ,
            reset   => BTNC,
            en_i    => s_en,
            cnt_up_i    => SW(0),
            cnt_o   => s_cnt
        );

    LED(3 downto 0) <= s_cnt;

    hex2seg : entity work.hex_7seg
        port map(
            hex_i    => s_cnt,
            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 <= b"1111_1110";

end architecture Behavioral;
```

Image of the top layer including both counters, ie a 4-bit bidirectional counter from Part 4 and a 16-bit counter with a 10 ms time base


SW
cnt_up_down
clk
cnt_o
en_i
cnt_up_i
reset
g_CNT_WIDTH = 4
s_CNT
hex_7seg
hex_i
seg_o
CA - CG
clock_enable
s_en
CLK100MHZ
clk
ce_o
reset
g_MAX = 100 000 000
LED[3:0]
AN 1111_1110
BTNC
clock_enable 1
clk
ce_o
reset
g_MAX = 1 000 000
cnt_up_down 1
CLK1MHZ
clk
cnt_o
en_i
cnt_up_i
reset
g_CNT_WIDTH =16
LED[15:0]