

05 - Counter

1. Tables

Table with connection of 5 buttons:

Button	Pin No.	Logic Val. When Pressed	Logic Val. When Not Pressed
BTNL	P17	0	1
BTNR	M17	0	1
BTNU	M18	0	1
BTND	P18	0	1
BTNC	N17	0	1

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"110_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"1_0111_1101_0111_1000_0100_0000"
500 ms	50 000 000	x"2FA_F080"	b"10_1111_1010_1111_0000_1000_0000"
1 sec	100 000 000	x"5F5_E100"	b"0101_1111_0101_1110_0001_0000_0000"

2. 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                -- Synchronous reset
            s_cnt_local <= (others => '0'); -- Clear all bits

        elsif (en_i = '1') then              -- Test if counter is enabled
            if (cnt_up_i = '1') then
                s_cnt_local <= s_cnt_local + 1;
            else
                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

```
-----
-- Reset generation process
-----
p_reset_gen : process
begin
    s_reset <= '0';
    wait for 33 ns;

    -- Reset activated
    s_reset <= '1';
    wait for 120 ns;

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

-----
-- Data generation process
-----
p_stimulus : process
begin
    report "Stimulus process started" severity note;

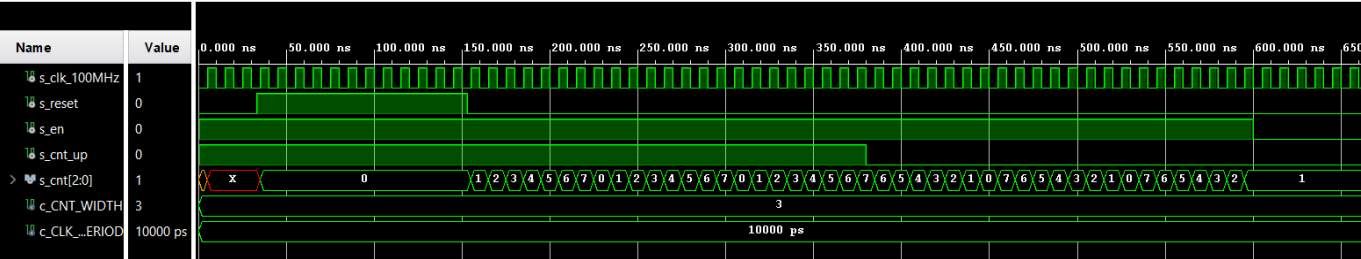
    -- Enable counting
    s_en      <= '1';

    -- Change counter direction
    s_cnt_up <= '1';
    wait for 380 ns;
    s_cnt_up <= '0';
    wait for 220 ns;

    -- Disable counting
    s_en      <= '0';

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

Screenshot with simulated time waveforms:



3. Top level

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

```
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 body for top level
-----

architecture Behavioral of top is

  -- Internal clock enable
  signal s_en  : std_logic;
  -- Internal counter
  signal s_cnt : std_logic_vector(4 - 1 downto 0);

begin

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

  -----
  -- Instance (copy) of cnt_up_down entity
  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
    );

-- Display input value on LEDs
LED(3 downto 0) <= s_cnt;

-----

-- Instance (copy) of hex_7seg entity
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
    );

-- Connect one common anode to 3.3V
AN <= b"1111_1110";

end architecture Behavioral;
```

The diagram shows the internal structure of the 'top' module. It includes two 'cnt_up_down' blocks, one 'hex_7seg' block, and two 'clock_enable' blocks. The inputs are SW(0), CLK100MHZ, and BTNC. The outputs are (CA-CG), LED(3:0), AN(7:0), and LED(15:0). The first counter (4-bit) outputs to the 7-segment display. The second counter (16-bit) outputs to the LED(15:0) display. Both counters are controlled by a common clock and reset signal, and their enable signals are generated by the 'clock_enable' blocks.