Learn Git and GitHub without any code!

Using the Hello World guide, you'll start a branch, write comments, and open a pull request.

Read the guide

FranciscaCampos / **Digital-electronics-1**

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main

Digital-electronics-1 / Labs / 05-counter / **README.md**

FranciscaCampos Update README.md

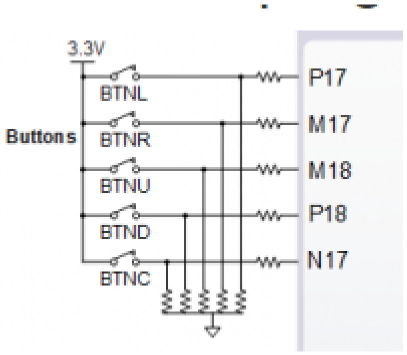
1 contributor

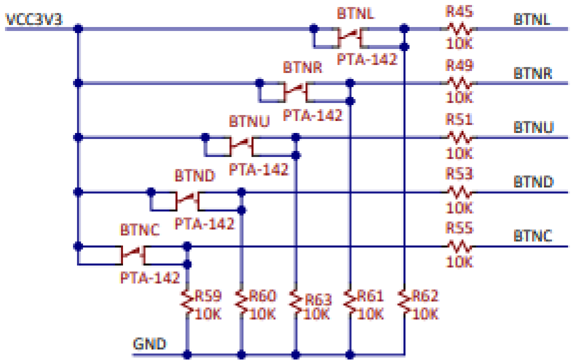
Raw Blame

157 lines (121 sloc) | 4.79 KB

Lab 5: Binary counter

1. Preparation tasks





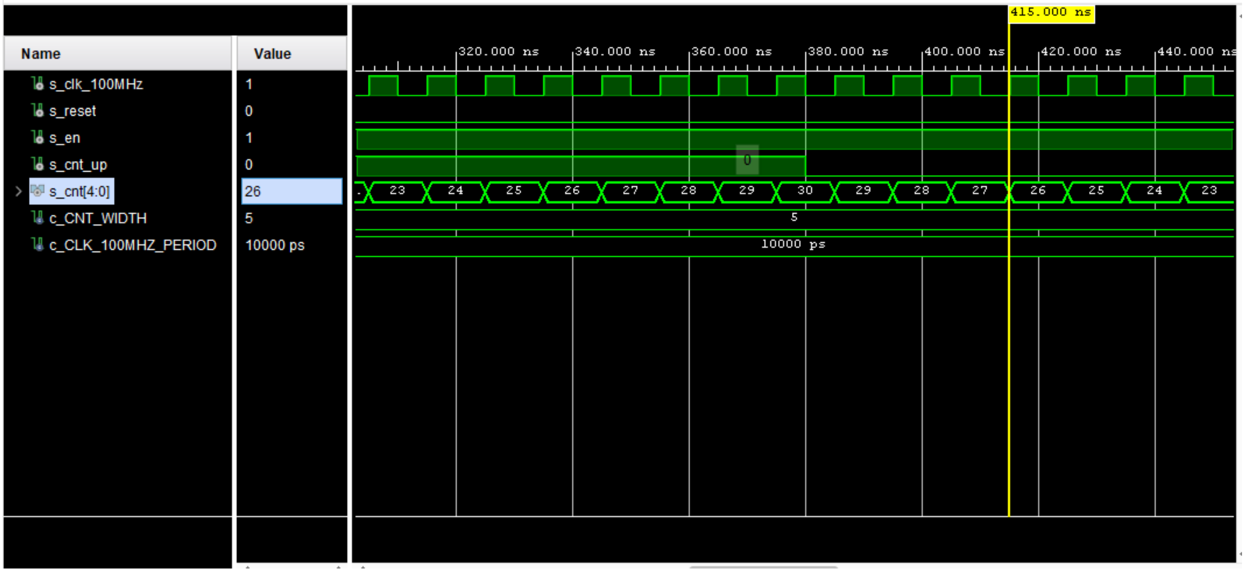
Time Interval	Number of clk periods	Number of clk periods in hex	Number of clk periods in binary

```
 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"  |
```

2. Seven-segment display decoder

```
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
```

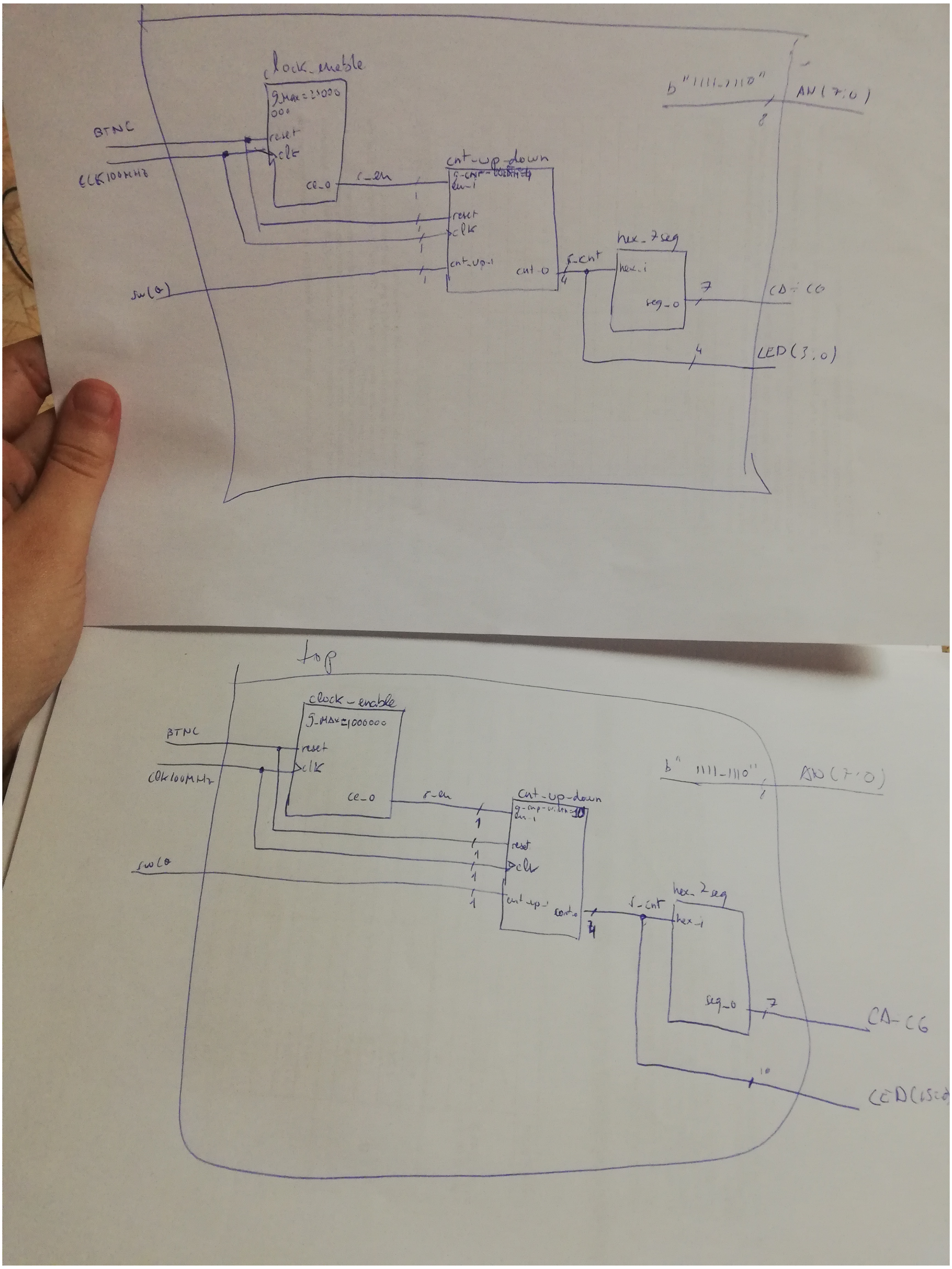
```vhdl
            -- TEST COUNTER DIRECTION HERE
            if (cnt_up_i = '1') then
                s_cnt_local <= s_cnt_local + 1;
            else
                s_cnt_local <= s_cnt_local - 1;
            end if;

        end if;
    end if;
end process p_cnt_up_down;

-- Output must be retyped from "unsigned" to "std_logic_vector"
cnt_o <= std_logic_vector(s_cnt_local);



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;

--------------------------------------------------------------------
-- 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;
```



Name	Value
s_clk_100MHz	1
s_reset	0
s_en	1
s_cnt_up	0
s_cnt[4:0]	26
c_CNT_WIDTH	5
c_CLK_100MHZ_PERIOD	10000 ps

3. Top Level

```
------------------------------------------------------------------------
-- 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 => g_MAX
       )
       port map(
           --- WRITE YOUR CODE HERE
        clk   => CLK100MHZ,
       reset => BTNC,
       ce_o  => s_en
       );

   --------------------------------------------------------------------
   -- Instance (copy) of cnt_up_down entity
   bin_cnt0 : entity work.cnt_up_down
       generic map(
           --- WRITE YOUR CODE HERE
                g_CNT_WIDTH => g_CNT_WIDTH
       )
       port map(
           --- WRITE YOUR CODE HERE
       clk  => CLK100MHZ,
       reset => BTNC,
       en_i => s_en,
       cnt_up_i => SW,
       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;
```


clock_enable
g_MAX=25000000
BTNC
reset
clk
CLK100MHZ
ce_o
s_en
cnt_up_down
g_CNT_WIDTH=4
en_i
reset
clk
cnt_up_i
cnt_o
sw(0)
s_cnt
4
hex_7seg
hex_i
seg_o
7
CA÷CG
4
LED(3:0)
b"1111_1110"
8
AN(7:0)
top
clock_enable
g_MAX=1000000
BTNC
reset
clk
CLK100MHz
ce_o
s_en
cnt_up_down
g_cnt_width 10
en_i
reset
clk
sw(0)
cnt_up_i
cnt_o
1
4
s_cnt
hex_7seg
hex_i
seg_o
7
CA-CG
10
LED(15:0)
b" 1111_1110"
AN(7:0)