

## FPGA Homework - 2

---

Parham Alvani (9231058)

April 2, 2016

### 1 PROBLEM 2

```
-----  
-- Author:      Parham Alvani (parham.alvani@gmail.com)  
--  
-- Create Date:  23-03-2016  
-- Module Name:  p2.vhd  
-----  
  
library IEEE;  
use IEEE.std_logic_1164.all;  
  
entity clk_dvdr is  
    port (clk : in std_logic;  
          clk_2, clk_5 : out std_logic);  
end entity;  
  
architecture BEHAVIORAL of clk_dvdr is  
    signal clk_2_tmp : std_logic := '0';  
    signal clk_5_tmp : std_logic := '0';  
begin  
    -- divide clock by 2:  
    -- counter values: 0 ... 1;  
    process (clk)  
        variable clk_2_var : integer := 0;
```

```

begin
    if clk'event and clk = '1' then
        clk_2_var := clk_2_var + 1;
        if clk_2_var = 1 then
            clk_2_var := 0;
            clk_2_tmp <= not clk_2_tmp;
        end if;
    end if;
end process;

-- divide clock by 5:
-- counter values: 0 ... 3; toggle: true;
-- counter values: 0 ... 2; toggle: false;
process (clk)
    variable clk_5_var : integer := 0;
    variable clk_5_toggle : boolean := false;
begin
    if clk'event then
        clk_5_var := clk_5_var + 1;
        if clk_5_var = 3 and clk_5_toggle then
            clk_5_var := 0;
            clk_5_tmp <= not clk_5_tmp;
            clk_5_toggle := not clk_5_toggle;
        elsif clk_5_var = 2 and not clk_5_toggle then
            clk_5_var := 0;
            clk_5_tmp <= not clk_5_tmp;
            clk_5_toggle := not clk_5_toggle;
        end if;
    end if;
end process;

clk_2 <= clk_2_tmp;
clk_5 <= clk_5_tmp;
end architecture BEHAVIORAL;

```

## 2 PROBLEM 3

```

-----
-- Author:          Parham Alvani (parham.alvani@gmail.com)
--
-- Create Date:     30-03-2016
-- Module Name:     p2.vhd
-----

```

```

library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_unsigned.all;

entity eight_bin_to_bcd is
    port (data_in : in std_logic_vector (7 downto 0);
          clk : in std_logic;
          R0, R1, R2 : out std_logic_vector (3 downto 0));
end entity eight_bin_to_bcd;

architecture rtl of eight_bin_to_bcd is
    signal R_t0, R_t1, R_t2 : std_logic_vector (3 downto 0);
begin
    process (clk, data_in)
        variable data_buff : std_logic_vector (7 downto 0);
    begin
        if data_in'event then
            data_buff := data_in;
            R_t0 <= "0000";
            R_t1 <= "0000";
            R_t2 <= "0000";
        elsif clk'event and clk = '1' then
            if data_buff >= "01100100" then
                data_buff := data_buff - "01100100";
                R_t2 <= R_t2 + "0001";
            elsif data_buff >= "00001010" then
                data_buff := data_buff - "00001010";
                R_t1 <= R_t1 + "0001";
            elsif data_buff >= "00000001" then
                data_buff := data_buff - "00000001";
                R_t0 <= R_t0 + "0001";
            else
                R0 <= R_t0;
                R1 <= R_t1;
                R2 <= R_t2;
            end if;
        end if;
    end process;
end architecture rtl;

```

### 3 PROBLEM 4

- 4 PROBLEM 5
- 5 PROBLEM 6
- 6 PROBLEM 7
- 7 PROBLEM 8
- 8 PROBLEM 9
- 9 PROBLEM 10
- 10 PROBLEM 11