In The Name of God

COMPUTER ENGINEERING DEPARTMENT OF AMIRKABIR UNIVERSITY OF TECHNOLOGY

FPGA Homework - 2

Parham Alvani (9231058)

April 3, 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

- Variables

  - Provide convenient mechanism for local storage
  - Scope is process in which they are declared
  - All variable assignments take place immediately

- Signals
  - Used for communication between VHDL components
  - Real, physical signals in system often mapped to VHDL signals
  - ALL VHDL signal assignments require either delta cycle or user-specified delay before new value is assumed

4 Problem 5

A VHDL entity can have different VHDL architectures. You can select the correct binding between entity and achitecture with the configuration. The entity is describing the inputs and outputs.

5 Problem 6

Singal	L	M	N
Value	Z	0	1

6 Problem 7

7 Problem 8

```
--------------------------------------------------------------------------------
-- Author:         Parham Alvani (parham.alvani@gmail.com)
--
-- Create Date:    28-03-2016
-- Module Name:    p8.vhd
--------------------------------------------------------------------------------
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_signed.all;

entity thermostat_ctrl is
        port (temp_needed, temp_sense : in integer;
                command : out boolean);
end entity thermostat_ctrl;

architecture rtl of thermostat_ctrl is
        signal temp_diff : integer;
```

```
begin
        temp_diff <= temp_sense - temp_needed;
        command <= false when temp_diff >= 2 else
                   true when temp_diff <= -2;
end architecture rtl;

architecture behavioral of thermostat_ctrl is
begin
        process (temp_needed, temp_sense)
        begin
                if temp_sense - temp_needed >= 2 then
                        command <= false;
                elsif temp_needed - temp_sense >= 2 then
                        command <= true;
                end if;
        end process;
end architecture behavioral;
```

8 Problem 9

```
--------------------------------------------------------------------------------
-- Author:         Parham Alvani (parham.alvani@gmail.com)
--
-- Create Date:    28-03-2016
-- Module Name:    p9.vhd
--------------------------------------------------------------------------------
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_unsigned.all;

entity counter is
        generic (N : integer := 4);
        port (clk, reset : in std_logic;
                count : out std_logic_vector (N - 1 downto 0));
end entity;

architecture behavioral of counter is
begin
        process (clk, reset)
                variable count_buff : std_logic_vector (N - 1 downto 0) := (others => '0')
        begin
                if clk'event and clk = '1' then
                        count_buff := count_buff + '1';
```

```
                        count <= count_buff;
                end if;
                if reset = '1' then
                        count_buff := (others => '0');
                        count <= count_buff;
                end if;
        end process;
end architecture behavioral;
```

9 Problem 10

```
----------------------------------------------------------------------------------
-- Author:         Parham Alvani (parham.alvani@gmail.com)
--
-- Create Date:    29-03-2016
-- Module Name:    p10.vhd
----------------------------------------------------------------------------------
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_unsigned.all;
use IEEE.numeric_std.all;

entity squart is
        generic (N : integer := 8);
        port (clk : in std_logic;
                data_in : in std_logic_vector (N - 1 downto 0);
                data_out : out std_logic_vector (N - 1 downto 0));
end squart;

architecture rtl of squart is
        signal result : std_logic_vector (N - 1 downto 0);
        signal mask : std_logic_vector (N - 1 downto 0);
begin
        process (clk, data_in)
                variable data_buff : std_logic_vector (N - 1 downto 0);
        begin
                if data_in'event then
                        data_buff := data_in;
                        mask <= ((N - 2) => '1', others => '0');
                        result <= (others => '0');
                elsif clk'event and clk = '1' then
                        if mask > data_in then
                                mask <= std_logic_vector(shift_right(unsigned(mask), 2));
```

```
                        elsif unsigned(mask) /= 0 then
                                if data_buff >= result + mask then
                                        data_buff := data_buff - (result + mask);
                                        result <= std_logic_vector(shift_right(unsigned(re
                                else
                                        result <= std_logic_vector(shift_right(unsigned(re
                                end if;
                                mask <= std_logic_vector(shift_right(unsigned(mask), 2));
                        else
                                data_out <= result;
                        end if;
                end if;
        end process;
end architecture rtl;
```

10 Problem 11

```
--------------------------------------------------------------------------------
-- Author:        Parham Alvani (parham.alvani@gmail.com)
--
-- Create Date:   28-03-2016
-- Module Name:   p11.vhd
--------------------------------------------------------------------------------
library IEEE;
use IEEE.std_logic_1164.all;

entity n_shift_register is
        generic (N : integer := 32);
        port (serial_in : in std_logic;
                w_s : in std_logic := '1';
                clk : in std_logic;
                serial_out : out std_logic;
                parallel_in : in std_logic_vector (N - 1 downto 0);
                parallel_out : out std_logic_vector (N - 1 downto 0));
end entity n_shift_register;

architecture rtl of n_shift_register is
        component d_register is
                port (d, clk : in std_logic;
                        q : out std_logic);
        end component;
        for all:d_register use entity work.d_register;
        signal Q : std_logic_vector (N downto 0);
```

```
        signal D : std_logic_vector (N downto 1);
begin
        Q(0) <= serial_in;
        serial_out <= Q(N);
        registers: for I in 1 to N generate
                D(I) <= Q(I - 1) when w_s = '1' else parallel_in(I - 1);
                ds : d_register port map (D(I), clk, Q(I));
                parallel_out(I - 1) <= Q(I);
        end generate registers;
end architecture rtl;

----------------------------------------------------------------------------------
-- Author:         Parham Alvani (parham.alvani@gmail.com)
--
-- Create Date:    22-02-2016
-- Module Name:    register.vhd
----------------------------------------------------------------------------------
library IEEE;
use IEEE.std_logic_1164.all;

entity d_register is
        port (d, clk : in std_logic;
                q : out std_logic);
end entity d_register;

architecture behavioral of d_register is
begin
        process (clk)
        begin
                if clk = '1' and clk'event then
                        q <= d;
                end if;
        end process;
end architecture behavioral;
```