Homework set 6

(These problems should be done individually, not with project partner)

Due – As indicated on Canvas

Instructions for submission

1) All answers are to be submitted in a **single** word/pdf file. This will be considered the final submission.

* ***Submit only one pdf/docx for the complete homework.***

2) Include Quartus Prime, Modelsim/Altera vwf and observation screenshots whenever possible with appropriate annotations for better presentation in word/pdf file.

****Do not submit extra screenshots in submission folder**

***** Screen shots outside this pdf will not be graded***

3) Naming convention and files required

a) LastName_FirstName_HWSet9.docx/pdf - for Observations and Answers

b) **<Codefilename>.vhd/.v - coded by you,** submit separately for each question **outside** the zip folder or paste in pdf/docx

**Commenting of code, references and presentation of material also carries weight*

*You can add snippets of code if you want to show critical code

1. A Handshake

a. For the data source, create an 8-bit counter in your favorite HDL language. Set the input clock to 25 MHz; use the 8 bits as a parallel source of data that is then registered with a synchronous enable. To transfer this registered data to the sink, create a handshake state machine at the source to generate a request signal and receive and synchronize an acknowledgement. For the data sink, create a receiving register that runs on a 37.5 MHz clock with a clock enable. Also make a state machine at the sink that synchronizes and reads a request signal and generates an acknowledgement signal and a transfer enable.

i. Place and route this circuit in the SmartFusion2 device. Determine the fMax. Run a simulation in Modelsim for at least 5 data transfers, be sure to capture a screen shot.

ii. How many data transfers are correct, with the intended data? If you add constraints can you improve the performance?

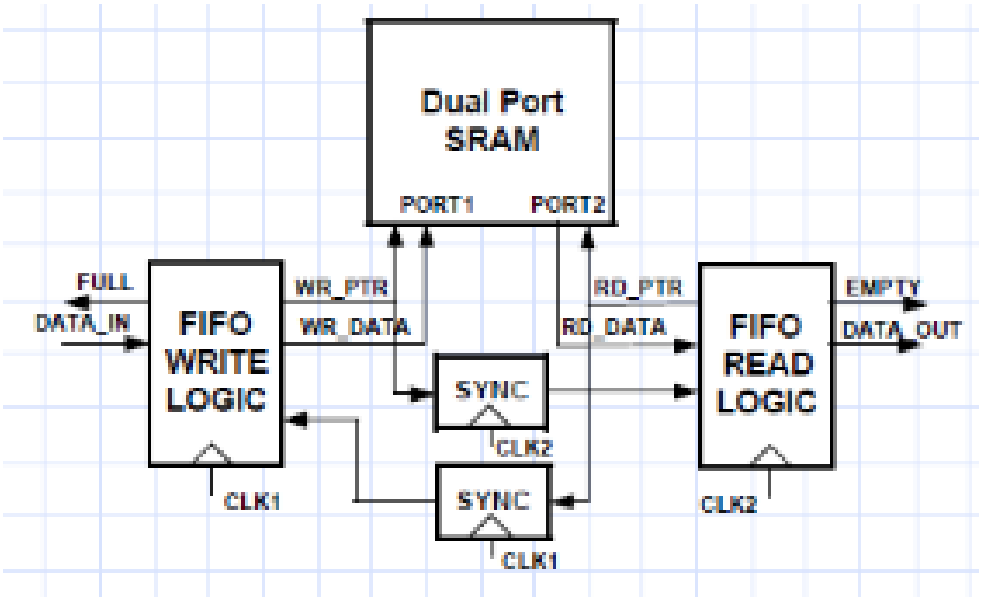
b. Repeat (a) with the Aclk at 37.5 MHz, and the Bclk at 25 MHz.

2. FIFO Design for CDC.

Complete the 2-stage FIFO design in VHDL shown by adding the full and empty signal generators (see the yellow sections below) in a Smartfusion2 device. Include statements for the synchronizers. Provide a Modelsim simulation.

Note: The rdptr and wrptr are bits with value 0 and 1 due to this being a 2-stage design.

If you are changing HDL to Verilog use the same signal names




```
Library ieee;
Use ieee.std_logic_1164.all;
Use ieee.numeric_std.all;

Entity fifo2x8 is port(
      clk1, clk2, rst:        in std_logic;
      rd, wr, rdinc, wrinc:   in std_logic;
      rdptrclr, wrptrclr:      in std_logic;
      full, empty: out std_logic;
      data_in:          in std_logic_vector(7 downto 0);
      data_out:         out std_logic_vector(7 downto 0);
end fifo2x8;

architecture archfifo2x8 of fifo2x8 is
      type fifo_array is array(1 downto 0) of std_logic_vector(7 downto 0); -
- makes use of VHDL's enumerated -- type
      signal fifo: fifo_array;
      signal wrptr, rdptr: std_logic;
      signal wrptrd1, wrptrd2: std_logic;
      signal rptrd1, rptrd2: std_logic;
      signal en: std_logic_vector(1 downto 0);
```

```
      signal dmuxout: std_logic_vector(7 downto 0);

begin

-- fifo register array (Dual port RAM);
reg_array: process(rst, clk1)
begin
      if rst = ‘1’ then
            for i in 1 downto 0 loop
                  fifo(i) <= (others => ‘0’);
            end loop;
      elsif (clk1’event and clk1 = ‘1’) then
            if wr = ‘1’ then -- wr could be the same as not FULL
                  for i in 1 downto 0 loop
                        if en(i) = ‘1’ then
                              fifo(i) <= data_in;
                        else
                              fifo(i) <= fifo(i);
                        end if;
                  end loop;
            end if;
      end if;
end process;
-- read pointer
Read_cout: process (rst, clk2)
Begin
      If rst = ‘1’ then
            rdptr <= ‘0’;
      elsif (rising_edge(clk2)) then
            if rdptrclr = ‘1’ then
                  rdptr <= ‘0’;
            elsif rdinc = ‘1’ then
                  rdptr <= nor(rdptr);
            end if;
      end if;
end process;

-- write pointer
Write_count: process (rst, clk1)
Begin
      if rst = ‘1’ then
            wprtr <= ’0’;
      elsif (clk1’event and clk1 = ‘1’) then
            if wrptrclr = ‘1’ then
                  wrptr <= ‘0’;
            elsif wrinc = ‘1’ then
```

```
                  wrptr <= not(wrptr);
            end if;
      end if;
end process;



-- generate FULL signal
detFull: process (rst, clk1)




end process;

-- generate EMPTY signal
detFull: process (rst, clk2)




end process;

-- 2:1 output data mux
With rdptr select
      Dmuxout <= fifo(0) when ‘0’,
                        fifo(1) when ‘1’,
fifo(1) when others;

-- FIFO register selector decoder
With wrptr select
      en <= “01” when ‘0’,
                  “10” when ‘1’,
                  “10” when others;

-- Clk2 control of outputs
Read_out: process (clk2, rd, dmuxout)
      Begin
            if (rising_edge(clk2)) then
                  If rd = ‘1’ then -- typically rd could be same as not empty
                        data_out <= dmuxout;
                  else
                        data_out <= data_out;
                  end if;
      end process;
end archfifo2x8;
```

3. Using the **Altera Power Play Early Power Estimator**, Estimate the power used by the smallest commercial temperature Cyclone V FPGA when it contains 32 16-bit counters running at 200 MHz Assume the last stage of each counter drives a 3.3v CMOS output. Show a screenshot of the final power calculation. Does this part require a heat sink?

Grading Rubric

1) – [12 points]
    [6 point]Completion of Implementation using Verilog/VHDL Code
    [6v point]Simulation using Modelsim
2) - [10 points]
    [5 point]Completion of Implementation using Verilog/VHDL Code
    [5 point]Simulation using Modelsim
3) - [8 points]
    [4 points]Power estimation and screenshots
    [4 point]Observation