Homework set 5

(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 as a final submission.

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

2) Include Quartus Prime, Modelsim 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_HWSetI.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

Important Instructions

*******Start Early*******

****Provide only Modelsim simulation and RTL view screenshots when mentioned in grading rubric ****

1. Synchronizer: Create an asynchronous data source and data sink using your favorite HDL.

a. For the data source, create an Aclk of 50 MHz and use a toggle D flip-flop to generate the data. For the data sink, create a Bclk of 27.7 MHz and use a **3 Flip-Flop synchronizer** to read the data source.

i. Build this circuit in a SmartFusion2 device.

ii. Run a post-fit simulation in Modelsim for at least 20 clock cycles. Capture the screenshot.

iii. Run Smart Time and report any errors listed.

iv. Add a clock domain constraint to Smart Time to improve slack time. (Consider using a false path constraint to remove timing analysis in the path in the first synchronizer FF, or other timing constraints). Take a screenshot of the report after your constraints are applied.

v. What is the initial and resulting slack time?

b. Repeat (a) with an Aclk of 27.7 MHz, and a Bclk of 50 MHz.

Use this entity/module:

```
entity CDC3FF is
port (
   Aclk : in std_logic;
   Bclk : in std_logic;
```

```
    reset : in std_logic;
    Dout : out std_logic
);
end CDC3FF;
```

2. Going fast and slow.

a. For the data source, create a 4-bit shift register in your favorite HDL language, fed by a toggle D flip-flop with a 100 MHz Aclk driving it. Use the 4-bits of the shift register as a parallel output. For the data sink, use a 4-bit input register driven by a 3.33 MHz clock. Use the SyncSignal method to synchronize the data coming in. Create the SyncSignal by using the slow clock as a data input to a FF clocked by the fast clock.

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

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 3.33 MHz, and the Bclk at 100 MHz.

*** you may need to have two different codes for Question 2, one for a and the other for b.*

Use this entity/module:

```
entity SlowFast is
port (
    Aclk, Bclk : in std_logic;
    reset : in std_logic;
    Dout : out std_logic_vector(3 downto 0)
);
end SlowFast;
```

Grading Rubric

1) - [10 points]

- [5 point] Implementation Verilog/VHDL Code
- [5 point]Simulation Screenshots (Modelsim), RTL view screen-shot, Observations

2) - [10 points]

- [5 point] Implementation Verilog/VHDL Code - Two separate code
- [3 point]Simulation Screenshots (Modelsim),
- [2 point] Observations, RTL view screenshot