Homework set 4

(These problems should be done individually, not with a 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 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_HWSet<>.docx/pdf - for Observations and Answers
b) **<Code filename>.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
***** ***Follow Grading Rubric for submission***

1. Calculate the MTBF for a Smart Fusion FPGA Flip-Flop under the following possible metastable conditions:

a. Clock frequency = 50 MHz, Tmet = 0 ns, C1 = 9.11 x 10^-12 Hz, C2 = 1.57 x 10^10 Hz,
fd = 10MHz.

Use equations from the **ProASIC3 Fabric User's Guide** found on the Microsemi website or on Canvas.

b. Clock frequency = 10 MHz, Tmet = 0 ns, C1 = 9.11 x 10^-12 Hz, C2 = 1.57 x 10^10 Hz,
fd = 1 MHz

2. **A Skewed problem.**

a. Create a 16 bit-counter in your favorite HDL language. Set the input clock to 50 MHz, and place and route this counter in the SmartFusion2 device. Determine the fMax. Run a timing simulation in ModelSim, be sure to capture a screenshot, and write down the number of glitches you see after running from a count of 0 to a count of 64.

b. Create a skewed clock. Using your 50 MHz input clock from before, feed this clock into a string of 16 inverters. Label the signal at every 4^{th} inverter sC(i), where i will be from 0 to 3. This will create a line of skewed clocks at each sC output. You may need to use synthesizer directives like syn_keep or syn_noprune to prevent your inverter string from being optimized out.

c. Replace the clocks in the last 4 bits (MSBs, bits 12 through 15) of your counter with sC(0) to sC(3). Recompile the design, and report any errors. Make changes to the circuit to resolve any errors, and then run

SmartTime and determine the fMax. Run the timing simulation as before and report the number of glitches or other errors you see after running from a count of 0 to 64.

```
library IEEE;

use IEEE.std_logic_1164.all;
use IEEE.Numeric_STD.ALL;


entity Skewed is
port (
    clk : in std_logic;
    reset : in std_logic;
    TC : out std_logic
);
end Skewed;
```

3. Please Constrain me.

Create a 24 bit-counter in your favorite HDL language. Set the input clock to 50 MHz, and place and route this counter in the fastest SmartFusion2 device. Drive a single bit TC output that is equivalent to the last FF output.

a. Constrain the clock, the input delay, and the output delay. Compute the Setup Data Arrival Time and Data Required Time for just the LSB and the MSB flip-flops in the counter. Compute the Hold Data Arrival Time and Data Required Time for the same.

b. Run SmartTime and determine the fMax. Report the Data Arrival and Data Setup times.

c. Think about ways you can improve fMax. Use different constraints or other circuit design methods, and try them. Determine the new fMax, and simulate your result. Make sure you have timing closure.

```
library IEEE;

use IEEE.std_logic_1164.all;

entity Strained is
port (
    clk : in std_logic;
    reset : in std_logic;
    TC : out std_logic  -- terminal count
);
end Strained;
```

Grading Rubric

1) - [5 points] Correct MTBF calculations

2) - [7 points] 2 point each for a, b, 3 points for c

3) - [8 points] 3 points each for a and c, 2 points for b