CSc 21100 (Fall 2019)
Project 05 (20 points)

IMPORTANT!

Please follow the **submission guidelines** below or your submission will be rejected.

1. You are expected to submit both a lab report and the source files to Blackboard in a single submission attempt.
2. The source codes must be under a single project.
3. The VHDL project needs to be exported from Xilinx ISE Design Suite. To export VHDL project file, please refer to Blackboard -> Content -> Lab -> **Exporting VHDL Project Files**.
4. Naming convention:
   Report: "FirstName_LastName_Project_**XX**_MM**Y**.pdf"
   Project: "FirstName_LastName_Project_**XX**_MM**Y**.zip"

   Replace "XX" and "Y" with the actual project number and section number, respectively.

In this project, students are expected to use the Xilinx ISE Design Suite (Webpack edition) 14.7 to complete the following tasks.

Please read the instructions carefully. Failing to follow the instructions would lead to significant point deductions.

Task 1: Counting 1s from two inputs (11 points)

In the homework assignment of Chapter 7, you are asked to design a clocked synchronous state machine to count the number of 1s from two inputs. The circuit has two binary inputs, X and Y, and one binary output, Z. It counts the total number of 1s from X or Y. If the number of 1 inputs on X and Y is a multiple of 4 (e.g., 0 or 4), the output is "1", otherwise the output is "0". When the output is "1", a new 1 from either X or Y (or both) will reset the output to "0".

Using structural design, implement the clocked synchronous state machine using VHDL. Please adopt the following entity declaration.

```
entity counter_XY is
    Port ( CLK : in  STD_LOGIC;
           X : in  STD_LOGIC;
           Y : in  STD_LOGIC;
           Z : out  STD_LOGIC);
end counter_XY;
```

Once you have the module implemented, use the test-bench program listed below and run simulations to validate your design.

```
 1  LIBRARY ieee;
 2  USE ieee.std_logic_1164.ALL;
 3
 4  ENTITY counter_XY_TB IS
 5  END counter_XY_TB;
 6
 7  ARCHITECTURE behavior OF counter_XY_TB IS
 8      COMPONENT counter_XY
 9      PORT(
10           CLK : IN  std_logic;
11           X : IN  std_logic;
12           Y : IN  std_logic;
13           Z : OUT  std_logic
14          );
15      END COMPONENT;
16     --Inputs
17     signal CLK : std_logic := '0';
18     signal X : std_logic := '0';
19     signal Y : std_logic := '0';
20     --Outputs
21     signal Z : std_logic;
22     -- Clock period definitions
23     constant CLK_period : time := 10 ns;
24  BEGIN
25     uut: counter_XY PORT MAP (
26            CLK => CLK,
27            X => X,
28            Y => Y,
29            Z => Z
30          );
31     -- Clock process definitions
32     CLK_process :process
33     begin
34        CLK <= '0';
35        wait for CLK_period/2;
36        CLK <= '1';
37        wait for CLK_period/2;
38     end process;
```

```
39    -- Stimulus process
40    stim_proc: process
41    begin
42       wait for 5 ns;
43       -- Test case #1
44       X<='0'; Y<='1'; wait for CLK_period*4; X<='0'; Y<='0';
45       -- Test case #2
46       X<='1'; Y<='0'; wait for CLK_period*4; X<='0'; Y<='0'; wait for CLK_period;
47       -- Test case #3
48       X<='1'; Y<='0'; wait for CLK_period; X<='0'; Y<='1'; wait for CLK_period;
49       X<='1'; Y<='0'; wait for CLK_period; X<='0'; Y<='1'; wait for CLK_period;
50       X<='0'; Y<='0'; wait for CLK_period*2;
51       -- Test case #4
52       X<='0'; Y<='1'; wait for CLK_period; X<='1'; Y<='0'; wait for CLK_period;
53       X<='0'; Y<='1'; wait for CLK_period; X<='1'; Y<='0'; wait for CLK_period;
54       X<='0'; Y<='0'; wait for CLK_period*2;
55       -- Test case #5
56       X<='0'; Y<='1'; wait for CLK_period; X<='0'; Y<='1'; wait for CLK_period;
57       X<='0'; Y<='0'; wait for CLK_period*2;
58       X<='0'; Y<='1'; wait for CLK_period; X<='1'; Y<='0'; wait for CLK_period;
59       X<='0'; Y<='0'; wait for CLK_period*2;
60       wait;
61    end process;
62 END;
```

Requirement(s):
In your VHDL implementation
- Please follow the structural design method;

Deliverable(s):

Your report:

1. Use your own language to describe the function of the module to be implemented in VHDL. (1 point)
2. Draw a circuit diagram of the module to show the design. (2 point)
3. Include your VHDL architecture definition(s). (1 point)
4. Show the RTL schematic of the module. (1 point)
5. Show simulation results (e.g. the waveforms). Include the outcome of **each** testcase with screenshots. Explain why your simulation result is correct. (2 points)

Your project file(s):

1. Can compile without any errors. (2 point)
2. Can run simulations without any errors. (2 point)

Note: no points will be given if requirements are not satisfied.

Task 2: Behavioral Design (9 points)

Repeat Task 1 but use a behavioral design. You can use the same test bench to run simulations and validate your implementation.

Requirement(s):
In your VHDL implementation
- Please follow the behavior design method;

Deliverable(s):

Your report:
1. Use your own language to explain how the circuit works. (1 point)
2. Include your VHDL entity declaration(s) and architecture definition(s). (1 point)
3. Show the RTL schematic of the module from Xilinx. (1 point)
4. Show simulation results (e.g. the waveforms). Explain the outcome of **each** testcase with screenshots. Show why the simulation result is correct. (2 points)

Your project file(s) should:
1. Compile without any errors. (2 point)
2. Run simulations without any errors. (2 point)

Note: no points will be given if requirements are not satisfied.