CSc 21100 (Fall 2019)
Project 03 (30 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: S′-R′ Latch (10 points)

An S′-R′ latch operates according to the following function table.

S_L	R_L	Q	QN
0	0	1	1
0	1	1	0
1	0	0	1
1	1	last Q	last QN

Write a VHDL program to implement an S′-R′ latch using *structural design*. Please adopt the following entity declaration.

```
entity slrl_latch is
   port ( R_L : in    std_logic;
          S_L : in    std_logic;
          Q   : out   std_logic;
          QN  : out   std_logic);
end slrl_latch;
```

Write a test-bench program and run simulations to validate your design. Use the given test cases in your test-bench program. Pay attention to the signal names, signal values, and the time.

```
47     BEGIN
48
49        s_l <= '0'; r_l <= '0';
50        wait for 50 ns;
51        s_l <= '0'; r_l <= '1';
52        wait for 50 ns;
53        s_l <= '1'; r_l <= '1';
54        wait for 50 ns;
55        s_l <= '1'; r_l <= '0';
56        wait for 50 ns;
57        s_l <= '1'; r_l <= '1';
58        wait for 50 ns;
59        s_l <= '0'; r_l <= '0';
60        wait for 50 ns;
61
62        WAIT; -- will wait forever
63     END PROCESS;
```

Requirements:

In your VHDL implementation please follow the structural design method.

Deliverables:

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. (1 point)
3. Include your VHDL entity declaration(s), architecture definition(s) and the testbench program. (1 point)
4. Show simulation results (e.g. the waveforms). Explain the outcome of **each** test case with screenshots. Show why the simulation result is correct. (3 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: S-R Latch with enable (10 points)

An S-R latch with enable operates according to the following function table.

S	R	C	Q	QN
0	0	1	last Q	last QN
0	1	1	0	1
1	0	1	1	0
1	1	1	1	1
x	x	0	last Q	last QN

It can be built based on a S′-R′ Latch. Write a VHDL program to implement the S-R Latch with Enable using *structural design*. Please adopt the following entity declaration.

```
entity sr_latch_en is
    Port ( S : in  STD_LOGIC;
           R : in  STD_LOGIC;
           C : in  STD_LOGIC;
           Q : out  STD_LOGIC;
           QN : out  STD_LOGIC);
end sr_latch_en;
```

Write a test-bench program and run simulations to validate your design. Use the given test cases in your test-bench program. Pay attention to the signal names, signal values, and the time.

```
51      -- c is asserted
52      s <= '0'; r <= '1'; c <= '1';
53      wait for 50 ns;
54      s <= '0'; r <= '0'; c <= '1';
55      wait for 50 ns;
56      s <= '1'; r <= '0'; c <= '1';
57      wait for 50 ns;
58      s <= '0'; r <= '0'; c <= '1';
59      wait for 50 ns;
60
61      -- c is negated
62      s <= '0'; r <= '1'; c <= '0';
63      wait for 50 ns;
64      s <= '0'; r <= '0'; c <= '0';
65      wait for 50 ns;
66      s <= '1'; r <= '0'; c <= '0';
67      wait for 50 ns;
68      s <= '0'; r <= '0'; c <= '0';
69      wait for 50 ns;
70      s <= '1'; r <= '1'; c <= '0';
71      wait for 50 ns;
72
73      -- c is again asserted
74      s <= '1'; r <= '1'; c <= '1';
75      wait for 50 ns;
76
77      WAIT; -- will wait forever
```

Requirement(s):
In your VHDL implementation
(1) Please follow the structural design method;
(2) Make use of the module(s) you implemented before.

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. (1 point)
3. Include your VHDL entity declaration(s), architecture definition(s) and the testbench program. (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. (3 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 3: D-Latch (10 points)

Build a D latch in Xilinx according to the following function table.

C	D	Q	QN
1	0	0	1
1	1	1	0
0	x	last Q	last QN

The D latch can be built based on an S-R Latch with Enable. Write a VHDL program to implement the D latch *using structural design*. Please adopt the following entity declaration.

```
entity d_latch is
    Port ( C : in  STD_LOGIC;
           D : in  STD_LOGIC;
           Q : out  STD_LOGIC;
           QN : out  STD_LOGIC);
end d_latch;
```

Write a test-bench program and run simulations to validate your design. Use the given test cases in your test-bench program. Pay attention to the signal names,

```
49    d <= '0'; c <= '1';
50    wait for 50 ns;
51    d <= '1'; c <= '1';
52    wait for 50 ns;
53    d <= '0'; c <= '0';
54    wait for 50 ns;
55    d <= '1'; c <= '0';
56    wait for 50 ns;
57    d <= '0'; c <= '1';
58    wait for 50 ns;
59    d <= '1'; c <= '1';
```

signal values, and the time.

Requirement(s):
In your VHDL implementation
(1) Please follow the structural design method;
(2) Make use of the module(s) you implemented before.

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. (1 point)
3. Include your VHDL entity declaration(s), architecture definition(s) and the testbench program. (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. (3 point)

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.