CSc 21100 (Spring 2021)
Project 04 (20 points)

IMPORTANT!

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

1. You are expected to submit two files (a PDF lab report and a ZIP file containing the source files) to Blackboard. Note that both files must be upload in the same submission attempt.
2. Each task must have its own source file(s). For VHDL assignments, all source files of this assignment must be in a single Xilinx VHDL project.
3. For VHDL assignments, you must use the export function of the Xilinx ISE Design Suite to create the ZIP file properly. For detailed steps, please refer to the document "Exporting_VHDL_project_files" on Blackboard.
4. Naming convention:

   Report: "FirstName_LastName_Project_**XX**_CC**Y**.pdf"*

   Project: "FirstName_LastName_Project_**XX**_CC**Y**.zip"*

   *Replace "XX" and "Y" with the actual project number (two digits) and section number, respectively.
5. After the due day, all submissions are final. You cannot change it for any reasons. Double check before you make the submission.

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 (5 points)

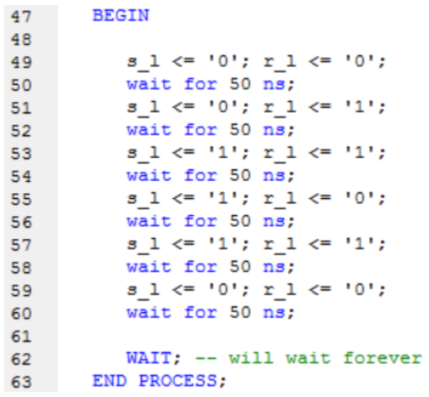
An S′-R′ latch operates according to the following function table.

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 make sure you use the entity declaration provided below. No points would be given if failed to follow it.

```
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.



```
    BEGIN

       s_l <= '0'; r_l <= '0';
       wait for 50 ns;
       s_l <= '0'; r_l <= '1';
       wait for 50 ns;
       s_l <= '1'; r_l <= '1';
       wait for 50 ns;
       s_l <= '1'; r_l <= '0';
       wait for 50 ns;
       s_l <= '1'; r_l <= '1';
       wait for 50 ns;
       s_l <= '0'; r_l <= '0';
       wait for 50 ns;

       WAIT; -- will wait forever
    END PROCESS;
```

Requirement(s):

(1) You must follow the structural design method.

(2) You must follow the submission guidelines.

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

Rubric (Report)

1.1	Draw a circuit diagram of the module to show the design. Use your own language to describe the function of the module to be implemented in VHDL. (1 point)
1.2	Include your VHDL entity declaration(s), architecture definition(s) and the testbench program. (1 point)
1.3	Show simulation results (e.g. the waveforms). Describe the outcome of each testcase with screenshots. Explain why the simulation result is correct. (1 point)

Rubric (Source Code)

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

Task 2: S-R Latch with enable (5 points)

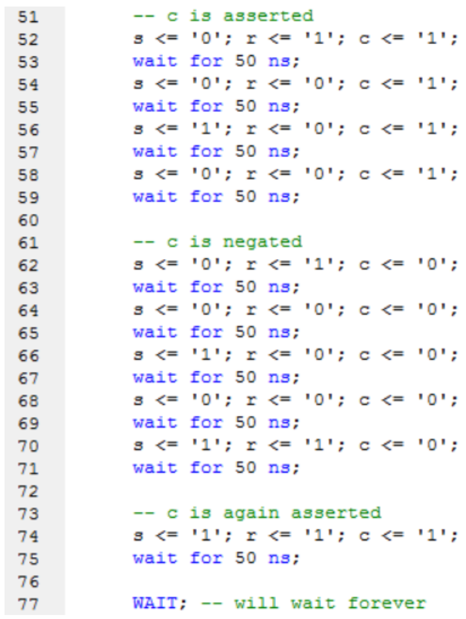
An S-R latch with enable operates according to the following function table.

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 make sure you use the entity declaration provided below. No points would be given if failed to follow it.

```vhdl
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.



```
        -- c is asserted
        s <= '0'; r <= '1'; c <= '1';
        wait for 50 ns;
        s <= '0'; r <= '0'; c <= '1';
        wait for 50 ns;
        s <= '1'; r <= '0'; c <= '1';
        wait for 50 ns;
        s <= '0'; r <= '0'; c <= '1';
        wait for 50 ns;

        -- c is negated
        s <= '0'; r <= '1'; c <= '0';
        wait for 50 ns;
        s <= '0'; r <= '0'; c <= '0';
        wait for 50 ns;
        s <= '1'; r <= '0'; c <= '0';
        wait for 50 ns;
        s <= '0'; r <= '0'; c <= '0';
        wait for 50 ns;
        s <= '1'; r <= '1'; c <= '0';
        wait for 50 ns;

        -- c is again asserted
        s <= '1'; r <= '1'; c <= '1';
        wait for 50 ns;

        WAIT; -- will wait forever
```

Requirement(s):
(1) You must follow the structural design method.
(2) You must use the module(s) implemented before.
(3) You must follow the submission guidelines.

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

Rubric (Report)

2.1	Draw a circuit diagram of the module to show the design. Use your own language to describe the function of the module to be implemented in VHDL. (1 point)
2.2	Include your VHDL entity declaration(s), architecture definition(s) and the testbench program. (1 point)
2.3	Show simulation results (e.g. the waveforms). Describe the outcome of each testcase with screenshots. Explain why the simulation result is correct. (1 point)

Rubric (Source Code)

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

Task 3: D-Latch (5 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 make sure you use the entity declaration provided below. No points would be given if failed to follow it.

```
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, signal values, and the time.

```
d <= '0'; c <= '1';
wait for 50 ns;
d <= '1'; c <= '1';
wait for 50 ns;
d <= '0'; c <= '0';
wait for 50 ns;
d <= '1'; c <= '0';
wait for 50 ns;
d <= '0'; c <= '1';
wait for 50 ns;
d <= '1'; c <= '1';
```

Requirement(s):
(1) You must follow the structural design method.
(2) You must use the module(s) implemented before.
(3) You must follow the submission guidelines.

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

Rubric (Report)

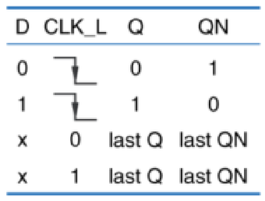
3.1	Draw a circuit diagram of the module to show the design. Use your own language to describe the function of the module to be implemented in VHDL. (1 point)
3.2	Include your VHDL entity declaration(s), architecture definition(s) and the testbench program. (1 point)
3.3	Show simulation results (e.g. the waveforms). Describe the outcome of each testcase with screenshots. Explain why the simulation result is correct. (1 point)

Rubric (Source Code)

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

Task 4: Negative Edge Triggered D Flip-Flop (5 points)

Build a negative edge triggered D flip-flop in Xilinx according to the following function table.

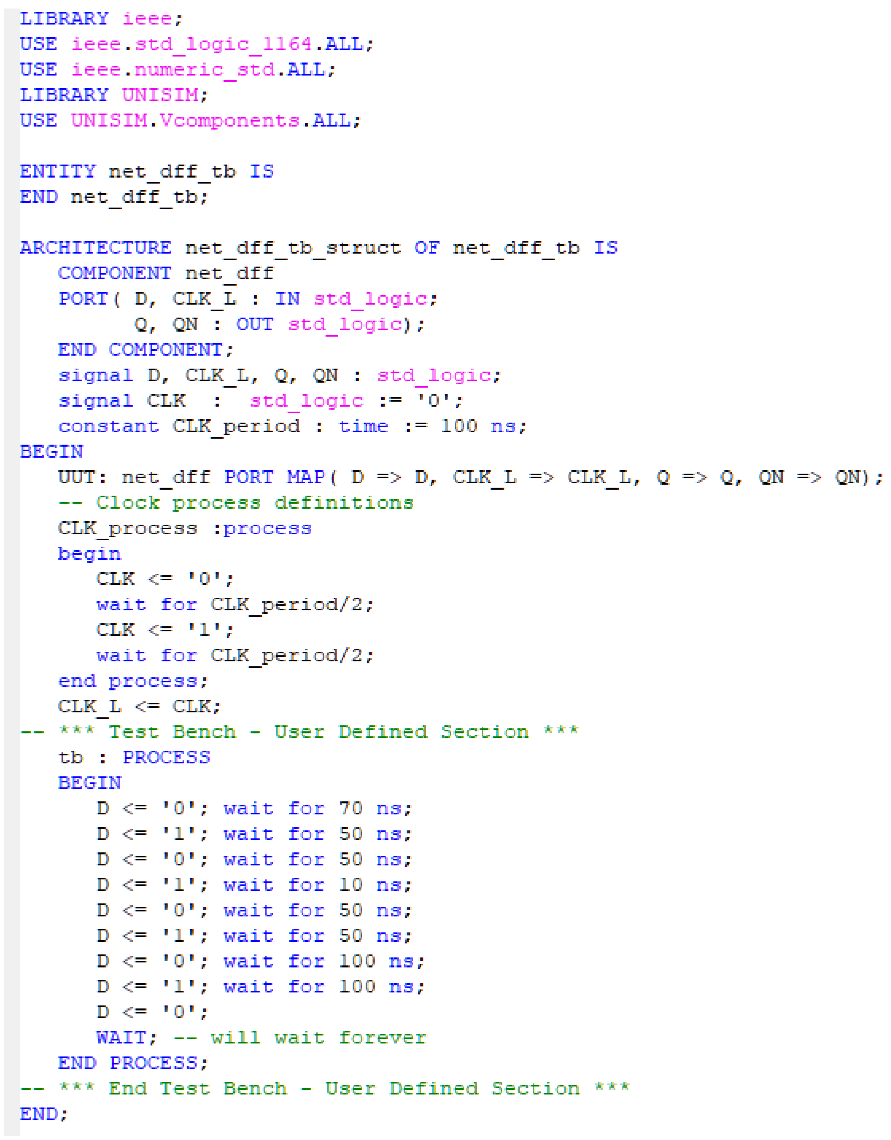


D	CLK_L	Q	QN
0	↓	0	1
1	↓	1	0
x	0	last Q	last QN
x	1	last Q	last QN

The negative edge triggered D flip-flop can be built based on the D latch in the previous task. Write a VHDL program to implement the negative edge triggered D flip-flop *using structural design*. Please make sure you use the entity declaration provided below. No points would be given if failed to follow it.

```
entity net_dff is
   port ( CLK_L : in    std_logic;
          D     : in    std_logic;
          Q     : out   std_logic;
          QN    : out   std_logic);
end net_dff;
```

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.



```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.numeric_std.ALL;
LIBRARY UNISIM;
USE UNISIM.Vcomponents.ALL;

ENTITY net_dff_tb IS
END net_dff_tb;

ARCHITECTURE net_dff_tb_struct OF net_dff_tb IS
   COMPONENT net_dff
   PORT( D, CLK_L : IN std_logic;
         Q, QN : OUT std_logic);
   END COMPONENT;
   signal D, CLK_L, Q, QN : std_logic;
   signal CLK  :  std_logic := '0';
   constant CLK_period : time := 100 ns;
BEGIN
   UUT: net_dff PORT MAP( D => D, CLK_L => CLK_L, Q => Q, QN => QN);
   -- Clock process definitions
   CLK_process :process
   begin
      CLK <= '0';
      wait for CLK_period/2;
      CLK <= '1';
      wait for CLK_period/2;
   end process;
   CLK_L <= CLK;
-- *** Test Bench - User Defined Section ***
   tb : PROCESS
   BEGIN
      D <= '0'; wait for 70 ns;
      D <= '1'; wait for 50 ns;
      D <= '0'; wait for 50 ns;
      D <= '1'; wait for 10 ns;
      D <= '0'; wait for 50 ns;
      D <= '1'; wait for 50 ns;
      D <= '0'; wait for 100 ns;
      D <= '1'; wait for 100 ns;
      D <= '0';
      WAIT; -- will wait forever
   END PROCESS;
-- *** End Test Bench - User Defined Section ***
END;
```

Requirement(s):

(1) You must follow the structural design method.
(2) You must follow the submission guidelines.

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

Rubric (Report)

4.1	Draw a circuit diagram of the module to show the design. Use your own language to describe the function of the module to be implemented in VHDL. (1 point)

4.2	Include your VHDL entity declaration(s), architecture definition(s) and the testbench program. (1 point)
4.3	Show simulation results (e.g. the waveforms). Describe the outcome of each testcase with screenshots. Explain why the simulation result is correct. (1 point)

Rubric (Source Code)

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