Group Coursework - II Group 9 (Deadline Week 24 - 16/04/2021)

Dr Purav Shah
CST2540 Digital System Design
Faculty of Science and Technology, Middlesex University

Learning Objective

In this group coursework, you will be tasked to design a few digital circuits and also design and implement using VHDL and perform timing simulations.

This coursework emphasises the mastering of VHDL/FPGA tools, VHDL syntaxes and complex digital circuit design knowledge you should have learned from your labs.

It is essential that you go through your lecture/seminar notes thoroughly, along with the essential book reading to complete this coursework.

You are advised to learn the VHDL syntaxes and coding techniques as much as you can from the book: FREE RANGE VHDL (http://www.freerangefactory.org/dl/free_range_vhdl.pdf). You should at very least skim through all the chapters to get better concept of VHDL programming.

The coursework is split into two sections:

1. Design challenges – specific for combinational circuits (aimed at problem solving)
2. Design challenges including full VHDL implementation and logic verification using Testbench simulation.

For the design and implementation problem including full VHDL programming:

I. Show how you developed your logic (present all the necessary steps).
II. Think of how your logic will be utilised in the VHDL program and decide your entity, architecture, test-bench etc.
III. Identify the possible timing simulation scenarios and use the testbench to evaluate your logic.

Key points:

i. Each student has to select one individual task from the design problem. This has to be highlighted in the report. The design and implementation tasks have to be completed no matter the group size.
ii. If you are a group of 2 students, then you complete only 2 out of 3 design tasks.
iii. Note that the marks are split as 50 marks for individual design task and 50 marks for group-based design and implementation. Thus, each student will be assessed based on individual and group activity.
iv. Highlight each member's contribution in your report.
v. Fill in the meeting form (see UniHub page where coursework is located). Add the completed meeting form in your report.
vi. Think of how your logic will be utilised in the VHDL program and decide your entity, architecture, test-bench etc.
vii. Go through your lecture notes, previous labs, seminars, etc. in order to get help in accomplishing your task.

Design Problems (Each student selects one of the tasks):

1. A sequential circuit has two inputs (X_1, X_2) and one output (Z). The output remains a constant value unless one of the following input sequences occurs:
   a. The input sequence $X_1 X_2$ = 01, 11 causes the output to become '0'.
   b. The input sequence $X_1 X_2$ = 10, 11 causes the output to become '1'.
   c. The input sequence $X_1 X_2$ = 10, 01 causes the output to change value.

   The notation $X_1 X_2$ = 10, 01 means X_1 = 0, X_2 = 1 followed by X_1 = 1, X_2 = 1.
   Derive a Moore state graph and state table for the circuit.

   (50 marks)

2. A Mealy sequential circuit has one input 'x' and one output 'z'. The output 'z' is '1' when the fourth, eighth, twelfth, etc. inputs are present. Also, z=1 if and only if the most recent input combined with the preceding three inputs was not a valid BCD encoding for a decimal digit; otherwise z=0. Assume the BCD digits are received *most* significant bit first. Derive a state table for the circuit (Hint: Eight states are sufficient)

   (50 marks)

3. Design a sequential circuit which will output Z = 1 for exactly four clock cycles each time a person pushes a button (which sets X = 1). The clock for a digital circuit is usually much faster than a person's finger! The person probably will not have released the button by the time four clock cycles have passed, so X may still be 1 when the four Z = 1 outputs have been generated. Therefore, after Z is 1 for four clock cycles, Z should go to 0, until X returns to 0 and then becomes 1 again. Design a Mealy state graph for this circuit.

   (50 marks)

Design & Implementation Problem (GROUP): (50 marks)

1. A 4-bit UP/DOWN binary counter has output Q and it works as follows: The state changes occur at the rising edge of the CLK input. When ClrN = 0 (asynchronous input), the counter is reset regardless of the values of the other inputs. If the LOAD input is 0, the data input D is loaded into the counter. The counter also has two additional ports, ENT and ENP (both active HIGH). They both need to be high for counting. If LOAD = ENT = ENP = UP = 1, the counter is incremented. If LOAD = ENT = ENP = 1 and UP = 0, the counter is decremented. If ENT = UP = 1, the carry output (CO) = 1 when the counter is in state 15. If ENT = 1 and UP = 0,the carry output (CO) = 1 when the counter is in state 0.

   a. Write a VHDL description of the counter.

   b. Draw a block diagram and write a VHDL description of an 8-bit binary up/down counter that uses the 4-bit counter (implemented in (a)) twice. (Look at cascading of counters). Test your code on the Nexys4 Board.

   (35 marks)

2. Below are some examples of VHDL code. Perform the task as mentioned.

(15 marks)

a. Following VHDL code represents a 2-to-1 MUX, but it contains mistakes. Identify those mistakes.

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity mux2 is
        port (d0, d1 : in bit; sel : in Boolean;z : out bit);
end mux2;

architecture bvhr of mux2 is
        signal muxsel : integer range 0 to 1;
        begin
                process(d0, d1, select)
                begin
                        muxsel <= 0;
                        if sel then muxsel <= muxsel + 1; end if;
                        case muxsel is
                        when 0 => z <=d0 after 2ns;
                        when1 =>  z <= d1 after 2ns;
                        end case;
                end process;
        end bvhr;
```

b. Describe the state table that is implemented in the following VHDL code.

```
entity Problem is
        port(X, CLK: in bit; Z1, Z2: out bit);
end Problem;

architecture Table of Problem is
        signal State, Nextstate: integer range 0 to 3 : = 0;
begin
process(State, X) --Combinational Circuit
begin
        case State is
        when 0 =>
                if X = '0' then Z1 <= '1'; Z2 <= '0'; Nextstate <= 0;
                else Z1 <= '0'; Z2 <= '0'; Nextstate <= 1; end if;
        when 1 =>
                if X = '0' then Z1 <= '0'; Z2 <= '1'; Nextstate <= 1;
                else Z1 <= '0'; Z2 <= '1'; Nextstate <= 2; end if;
        when 2 =>
                if X = '0' then Z1 <= '0'; Z2 <= '1'; Nextstate <= 2;
                else Z1 <= '0'; Z2 <= '1'; Nextstate <= 3; end if;
        when 3 =>
                if X = '0' then Z1 <= '0'; Z2 <= '0'; Nextstate <= 0;
                else Z1 <= '1'; Z2 <= '0'; Nextstate <= 0; end if;
        end case;
end process;

process(CLK) -- State Register
        begin
```

```
            if CLK'event and CLK = '1' then
                State <= Nextstate;
        end if;
        end process;
end Table;
```

Coursework Participation

You are required to work in a group of 2-3 students. Please note, make sure you identify any issues regarding the group work well in advance. Last minute complaints about the group member will be disregarded and the penalty of not submitting the work will be applied to all members. Please use the group meeting form to make sure you document your meetings that have taken place to complete the coursework.

Marking Scheme

This coursework will be marked out of 100% where the marks are distributed as shown in each design challenge based on its difficulty level. **Note: This coursework is worth 20% contributing towards the module's overall grade.**

Deliverable

You need to submit the lab report in .pdf format along with a zip file containing the implementation files.

Your lab report and zip files should be submitted via myUniHub (link provided on module page) by **16th April 2021**. The format of the report is of your choice, but it should contain the contents relevant to the marking criteria (see below). The marking criteria is available on MyUniHub. In the report you should use screenshots to demonstrate the successful completion of each task.

Your submission should also include a zip file containing:

- For each design, you should submit a Xilinx Project containing your solution (VHDL codes, Xilinx Project file), please remove any intermediate files generated by the tools.

Report Format:

The report must contain the following:

1) Introduction to the tasks
2) Design Steps, how you achieved the outcome. (K-Maps, Logic Equations, Block diagram, FSM, Transition Tables, etc.)
3) Programming snapshots
4) Testbench evaluation for each design
5) Group Meeting form (evidence of group work and selection of individual tasks.