*Name: _____

*Registration: _____

Department of Computer Systems Engineering
University of Engineering & Technology Peshawar

Digital System Design

CSE 308

Midterm Examination Spring 2016

1 April 2016, Duration: 120 Minutes

Exam Rules

Please read carefully before proceeding.

1- This exam is open books/notes but closed laptops/Internet.
2- No calculators/mobiles of any kind are allowed.
3- It's good to share but sharing of books, notes, and other materials during this exam is not permitted.
4- There are 4 problems in total. Some problems are harder than others. Answer the easy ones first to maximize your score.
5- This exam booklet contains 13 pages, including this cover. Count them to be sure you have them all.

Problem 1	_____	(15 pts)
Problem 2	_____	(15 pts)
Problem 3	_____	(50 pts)
Problem 4	_____	(20 pts)
Exam Total	_____	(100 pts)

Good Luck!

*Registration: _____

Problem 1: (15 pts)

The following code is intended to implement a 1:2 de-MUX. The input is x and two outputs are y and z, and the switch control is s.

```
wire x, y;
reg z;
always @(x or y or z)
   begin
      if (!s) y=x;
      else z=x;
   end
```

(a) (5 pts) Find out all things wrong in this code.

(b) (5 pts) Does the code produce any latches? If no, state why. If yes, how many latches can be produced? And by which statement(s)?

(c) (5 pts) After all problems you find in **(a)** are fixed, draw the logic diagram of the synthesized hardware from the corrected code.

*Registration: _____

Problem 2: (15 pts)

(a) (8 pts) Below is a short snippet of Verilog RTL code of a digital functional block. Draw the logic circuit diagram that is described by the Verilog code.

```
module M(Q, S, A, B, C);
      output Q;
      input S, A, B, C;
      reg Q;

      always @(posedge C)
            Q <= (S) ? A : B;
endmodule
```

(b) (7 pts) Complete the following clk_gen module, which generates a clock signal that initially goes to zero for 15 ns, then goes to one for 5 ns, and then repeats this pattern indefinitely. Your module can only use one initial statement.

```
                        // COMPLETE CODE HERE
module clk_gen;
      reg clk;

      initial
            begin
            // COMPLETE CODE HERE









            end
endmodule
```

*Registration: _____

Problem 3: (50 pts)

This problem involves creating several Verilog designs. Four lower-level modules are written and then two of these are combined to create a top-level design.

For each module, write a complete Verilog program with correct declaration, etc. Write your answers below the module description. A completely correct solution will receive the number of points indicated in parentheses. Partially correct solutions will receive partial credit.

(a) Module 1: Gray Counter. (10 pts)

In this module, design a 3-bit gray counter with positive reset. When reset, the count value becomes "000". Recall that a gray counter changes only one bit at a time. For example, a 2-bit gray counter has a count sequence 00, 01, 11, 10 corresponding to decimal count values of 0, 1, 2, and 3 respectively. In the gray count sequence, only one bit changes between adjacent count values, and the right-most bit is changed as long as it does not result in a code word that has been visited earlier.

The following are the ports of the module:

CLK	1-bit clock input, all actions must be on the positive edge
RESET	1-bit reset, causes reset on the positive edge
GRAY_OUT	3-bit result

(b) Module 2: Parallel-in, Serial-out Shift Register. (10 pts)

In this module, create a register that loads data in parallel but shifts data out serially, MSB first.

The following are the ports of the module:

CLK	1-bit clock, all operations must be on the rising edge
PI	8-bit parallel data input
Q	1-bit serial output
LD	1-bit input, when high, PI is loaded into the shift register
SHIFT	1-bit shift enable input, when high, contents of the shift register are shifted out on to the serial output Q

(c) Module 3: Up-Down, Loadable Counter. (10 pts)

In this module, create a counter that counts in both the up and down directions. The preset is to be set to decimal value 3. That is, upon asserting the reset signal RESET low, the register value should be reset to 3. Also, upon asserting the load signal LD high, the register value should be set to the input value DIN. Both the reset and the load are synchronous and the module should count on the falling edge of the clock.

The following are the ports of the module:

CLK	1-bit clock input, all actions performed on falling edge
RESET	1-bit preset (synchronous)
UP_DN	1-bit input (if '1', then count up, if '0', then count down)
LD	1-bit load enable input, loads synchronized with CLK falling edge
DIN	3-bit input data for loading counter value
Q	3-bit result

(d) Module 4: Multiplexer. (10 pts)

In this module, create a byte-wide 8-to-1 multiplexer. In this case, the value on the 3-bit select line will route 1 of 8 inputs to the output. This module is purely combinatorial.

The following are the ports of the module:

SEL	3-bit select line
D0, D1, D2, D3, D4, D5, D6, and D7	8-bit data inputs
O	8-bit output

(e) Top-Level Design. (10 pts)

For this design, combine gray counter (**Module 1**) with the multiplexer (**Module 4**) to create a circuit such that the output of the gray counter controls the select lines of the multiplexer.

The top-level design has the following port definitions:

CLK	1-bit clock
RESET	1-bit reset line
IN0, IN1, IN2, IN3, IN4, IN5, IN6, and IN7	8-bit data inputs
OUT_DATA	8-bit data output

Problem 4: (20 pts)

You are a Hardware Design Engineer working for DCSE (suppose only!). They want you to design a FSM that will automate their peon Peerzada's movement on the UET campus. The HoD wirelessly transmits the travel plans to Peer, and then Peer moves according to that information.

To design your FSM, you first select the following locations around the UET campus and assign each location with a state in 3-bit binary representation: HoD's office[**000**], Lab-1[**001**], Lab-2[**010**], Lab-3[**011**], Main office[**100**], Dean's office[**101**], Registrar's office[**110**], and the VC's office[**111**].

To simplify your design, you inform the HoD to send Peer a binary sequence for travel plans (e.g. '1-0-0-0-1' to cause Peer to move five times). In other words, Peer receives either '0' or '1' for each move and travels to the next destination as specified below. Peer starts off at HoD's office, and your FSM should output Peer's current location.

Current location	Next location	
HoD's office[**000**]:	If 0, stay at HoD's office.	If 1, go to Lab-1.
Lab-1[**001**]:	If 0, go to Lab-2.	If 1, go to Main office.
Lab-2[**010**]:	If 0, go to Lab-3.	If 1, go to Main office.
Lab-3[**011**]:	If 0, stay at Lab-3.	If 1, go to HoD's office. (Laiq sb. chai ghwardi!)
Main office[**100**]:	If 0, go to VC's office.	If 1, go to Dean's office.
Dean's office[**101**]:	If 0, go to Lab-3.	If 1, go to Registrar's office.
Registrar's office[**110**]:	If 0, go to VC's office.	If 1, stay at Registrar's office.
VC's office[**111**]:	If 0, go to Lab-1.	If 1, go to Dean's office.

(a) (5 pts) Draw the state transition diagram for this FSM. Please use a scratch page for scratch work and make a neat copy of your final diagram below.

(b) (5 pts) Is the FSM a Moore or Mealy machine? Justify your answer.

Moore/Mealy machine (1pt):
Justification (4 pts):

(c) (10 pts) Design a module in Verilog for this FSM – the following schematic shows the appropriate inputs and outputs. Please use a scratch page as scratch space and write a neat copy of the final code here.

*Registration: _____

⟨This page is intentionally left blank. This page can be used for scratch work or as extra space. If you write work here that you want me to grade, be sure to clearly indicate which problem(s) the work corresponds to!⟩

*Registration: _____

⟨This page is intentionally left blank. This page can be used for scratch work or as extra space. If you write work here that you want me to grade, be sure to clearly indicate which problem(s) the work corresponds to!⟩