*Name: Solution

*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!

Problem 1: (15 pts)

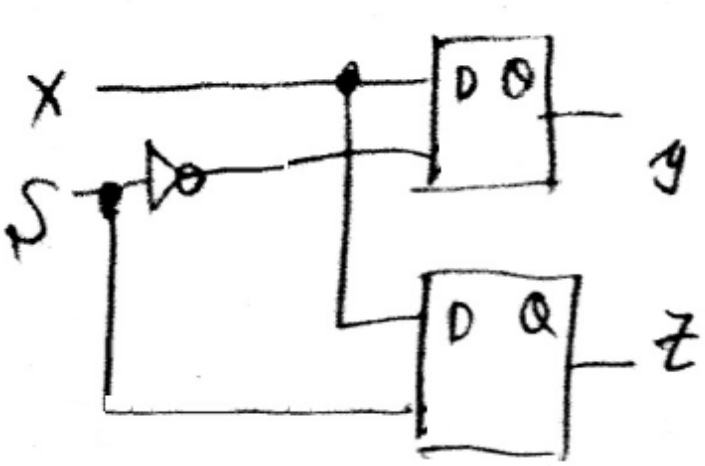
(a) (5 pts)

1. y has to be defined as reg
2. y and z must not be in the trigger list
3. s must be in the sentivity list

(b) (5 pts)

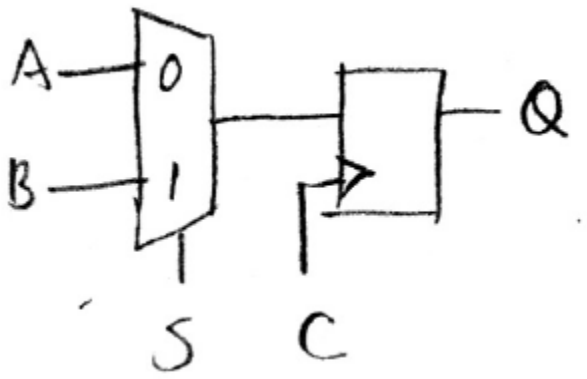
Yes. 2 latches
One to store previous value of z, when s = 0 (y = x).
One to store previous value of y, when s = 1 (z = x).

(c) (5 pts)



Problem 2: (15 pts)

(a) (8 pts)



(b) (7 pts)

```
‘timescale 1ns / 1ns
module clk_gen;
      reg clk;

      initial
      begin
      clk = 1’b0;
      forever
                begin
                      #15 clk = ~ clk;
                      #5  clk = ~ clk;
                end
      end
endmodule
```

Problem 3: (50 pts)

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

```
module GRAY_CNT (
                 CLK,
                 RESET,
                 GRAY_OUT);

     input CLK;
     input RESET;
     output [2:0] GRAY_OUT;

     reg [2:0] GRAY_OUT;

     always @ (posedge CLK)
          if (RESET)
               GRAY_OUT = 3'b000;
          else
               case (GRAY_OUT)
                    3'h0: GRAY_OUT = 3'h1;
                    3'h1: GRAY_OUT = 3'h3;
                    3'h3: GRAY_OUT = 3'h2;
                    3'h2: GRAY_OUT = 3'h6;
                    3'h6: GRAY_OUT = 3'h7;
                    3'h7: GRAY_OUT = 3'h5;
                    3'h5: GRAY_OUT = 3'h4;
                    3'h4: GRAY_OUT = 3'h0;
                    default: GRAY_OUT = 3'h0;
               endcase
endmodule
```

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

```
module PISO_SHIFT (
                   CLK,
                   LD,
                   SHIFT,
                   PI,
                   Q);
     input CLK;
     input LD;
     input SHIFT;
     input [7:0] PI;
     output Q;

     reg Q;
     reg [7:0] SHIFTER;

     always @ (posedge CLK)
          if (LD)
               SHIFTER = PI;
          else if (SHIFT)
               begin
                    Q = SHIFTER[7];
                    SHIFTER = {SHIFTER[6:0], 1'b0};
               end
endmodule
```

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

```
module UP_DN_LD_CNT (
                     CLK,
                     RESET,
                     UP_DN,
                     LD,
                     DIN,
                     Q);
     input CLK;
     input RESET;
     input UP_DN;
     input LD;
     input [2:0] DIN;
     output [2:0] Q;

     reg [2:0] Q;

     always @ (negedge CLK)
          if (~RESET)
               Q = 3'b011;
          else
               if (LD)
                    Q = DIN;
               else if (UP_DN)
                    Q = Q + 1;
               else
                    Q = Q - 1;
endmodule
```

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

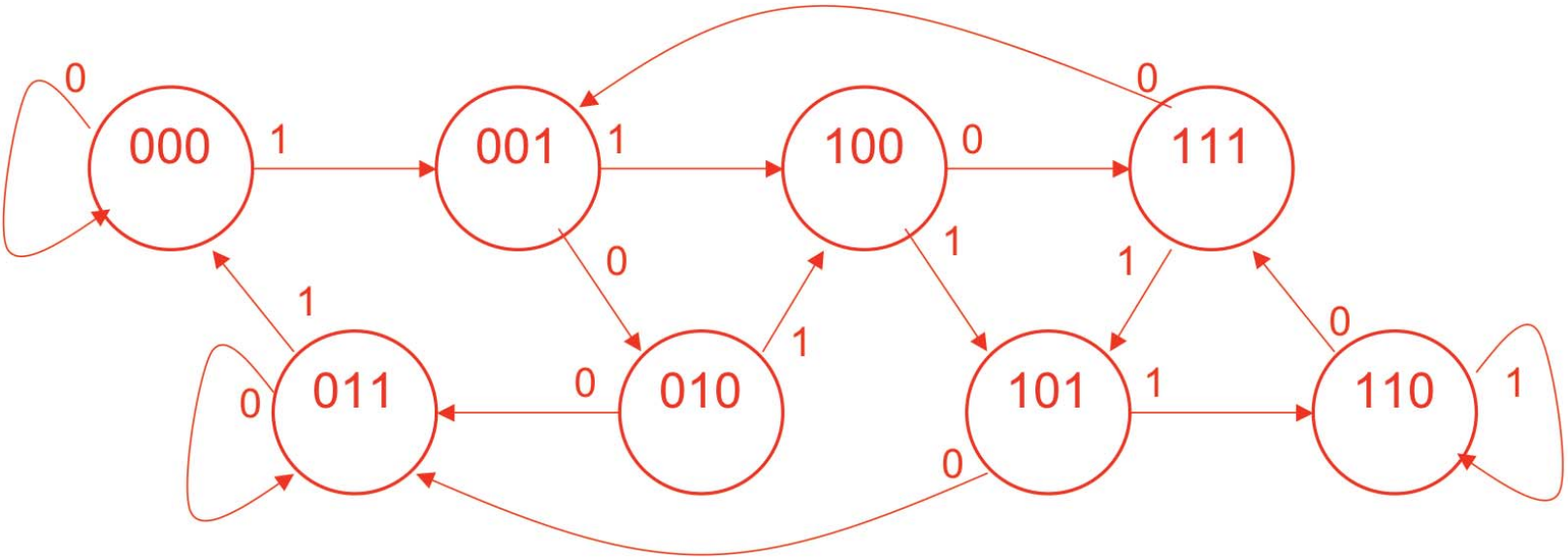
```
module MUX_8 (
              SEL,
              D0,
               D1,
               D2,
               D3,
               D4,
               D5,
              D6,
              D7,
               O);

     input [2:0] SEL;
     input [7:0] D0;
     input [7:0] D1;
     input [7:0] D2;
     input [7:0] D3;
     input [7:0] D4;
     input [7:0] D5;
     input [7:0] D6;
     input [7:0] D7;
     output [7:0] O;

     reg O;

     always @ (*)
          case (SEL)
               3'b000: O = D0;
               3'b001: O = D1;
               3'b010: O = D2;
               3'b011: O = D3;
               3'b100: O = D4;
               3'b101: O = D5;
               3'b110: O = D6;
               3'b111: O = D7;
          endcase
endmodule
```

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

```
module TOP (
            CLK,
            RESET,
            OUT_DATA,
            IN0,
            IN1,
            IN2,
            IN3,
            IN4,
            IN5,
            IN6,
            IN7);

     input CLK, RESET;
     input [7:0] IN0;
     input [7:0] IN1;
     input [7:0] IN2;
     input [7:0] IN3;
     input [7:0] IN4;
     input [7:0] IN5;
     input [7:0] IN6;
     input [7:0] IN7;
     output [7:0] OUT_DATA;

     wire [7:0] GRAYOUT;

     GRAY_CNT GRAYCNT_INST1 (CLK, RESET, GRAYOUT);
     MUX_8 MUX8_INST1 (GRAYOUT, IN0, IN1, IN2, IN3, IN4, IN5, IN6, IN7,
     OUT_DATA);

endmodule
```

Problem 4: (20 pts)

(a) (5 pts)



(b) (5 pts)

Moore/Mealy machine (1pt): Moore machine
Justification (4 pts): This is a Moore machine because the output (location) can be determined from the state.

(c) (10 pts)

```
module PEER (
              CLK,
              RESET,
              MOVE,
              STATE);

     input CLK, RESET, MOVE;
     output reg [2:0] STATE;

     always @ (posedge CLK)
          begin
               if (RESET)
                    STATE <= 3'b000;
               else
                    case (STATE)
                         3'b000: STATE <= MOVE ? 3'b001 : 3'b000;
                         3'b001: STATE <= MOVE ? 3'b100 : 3'b010;
                         3'b010: STATE <= MOVE ? 3'b100 : 3'b011;
                         3'b011: STATE <= MOVE ? 3'b000 : 3'b011;
                         3'b100: STATE <= MOVE ? 3'b101 : 3'b111;
                         3'b101: STATE <= MOVE ? 3'b110 : 3'b011;
                         3'b110: STATE <= MOVE ? 3'b110 : 3'b111;
                         3'b111: STATE <= MOVE ? 3'b101 : 3'b001;
                         default: STATE <= 3'b000;
                    endcase
          end
endmodule
```