# ECE 426/516 Implementation of VLSI Systems

## Practice Final Exam Paper

**Name: _____ SUID: _____**

**Question 1 (3 marks)**

Draw a schematic of the circuit defined in each Verilog module below.

(1)
```
module myblock1 (a, b, c, y);
input a, b, c;
output y;
        assign y = ~a & ~b | ~a & ~c | ~b & ~c;
endmodule
```

(2)
```
module myblock2 (d0, d1, s, y);
input [3:0] d0, d1;
input s;
output [3:0] y;
        assign y = s ? d1:d0;
endmodule
```

(3)
```
module myblock3 (co, s, ci, a, b);
input ci, a, b;
output co, s;
wire a_xor_b;
wire a_and_b;
wire ci_and_a_xor_b;

xor u1 (a_xor_b, a, b);
and u2 (a_and_b, a, b);
and u3 (ci_and_a_xor_b, ci, a_xor_b);
or u4 (co, a_and_b, ci_and_a_xor_b);
xor u5 (s, ci, a_xor_b);
endmoudule
```

**Question 2 (5 marks)**

An entry level engineer has implemented a Verilog code with the objective of designing a counter that continuously counts down from 7 to 0 in binary, and then back to 7 again (count 111, 110, 101, … 000, 111, 110, 101, …). Unfortunately, the designer made a number of mistakes in the design. Please fix all the bugs using minimal changes. The counter bits are: c3 c2 c1 (assume c3 is the most significant bit).

```
module top (clk, r, c1, c2, c3);
input r, clk;
output c1, c2, c3;

reg c1, c2, c3;
assign x = 1'b1;

my_unit u1 (clk, x, x, c1);
my_unit u2 (clk, r, c1, x, c2);
my_unit u3(clk, r, x, c2, c3);
endmodule

module my_unit (clk, r, a, b, c);
input clk, r, a, b;
output c;
reg c;

always @ (negedge clk)
c <= (~a*~c) | ((~b)*c);
always @ (r)
c <= 1'b0;
endmodule
```

**Question 3 (5 marks)**

Sketch the state transition diagram for the FSM described by Verilog code below.

```
module fsm (clk, reset, a, b, y);
input clk, reset, a, b;
output y;
reg [1:0] state, nextstate;

parameter S0 = 2'b00;
parameter S1 = 2'b01;
parameter S2 = 2'b10;
parameter S3 = 2'b11;

always @ (posedge clk, posedge reset)
        if (reset) state <= S0;
                else state <= nextstate;

always @ (*)
        case (state)
                S0: if(a ^ b) nextstate = S1;
                else nextstate = S0;
                S1: if(a & b) nextstate = S2;
                else nextstate = S0;
                S2: if(a | b) nextstate = S3;
                else nextstate = S0;
                S3: if(a | b) nextstate = S3;
                else nextstate = S0;
        endcase
assign y = (state==S1) | (state == S2);
endmodule
```

**Question 4 (7 marks)**

A recognizer has one input “X” and one output “Y”. At each clock cycle, the input “X” value is read. When a sequence of “101” is observed in the input sequence, the output “Y” will become 1, otherwise it will be 0.

(a) Draw Moore state machine diagram with minimum number of states (3 marks)

(b) Write the Verilog representation of your Moore state machine (4 marks)