Name: _____ CWID: _____ Date: _____

CALIFORNIA STATE UNIVERSITY, FULLERTON

Computer Engineering

EGCP 446 – Advanced Digital Design using Verilog HDL (Fall 2018)

Midterm Exam 1(Total Points = 75)

(Time allowed: 7:00 — 9:00 PM)

Academic Dishonesty Policy

In line with University policies, the Computer Engineering program supports a strict and well-defined policy against academic dishonesty. Thus, to assure a fair and equitable testing environment for all students, there will be zero tolerance during exam for any of the following:

- Cheating of any type (looking at or copying another student's answers) or helping another student with answers.
- Use of notes, phones, or other aids (other than that allowed by instructor)
- Talking or texting during exams
- Leaving the classroom during the exam (without permission)

Consequences for violating these policies will be a "zero" on the exam at a minimum, with the possibility of an F in the course.

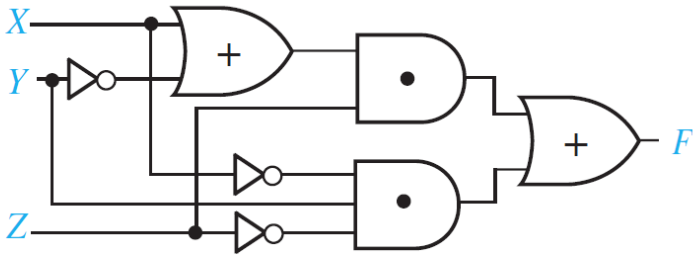
Only one page of handwritten notes (two side), pens/pencils, erasers, and a calculator (shouldn't be necessary) are allowed with the exam.

Normally, full credit is given only if work is shown when appropriate.

Name: _____ CWID: _____ Date: _____

1. **(20 Points) Behavior Modelling**

From a 3-bit combinational circuit shown below a) Write the Truth Table b) Express the Boolean functions using either Sum of Products (SOP), or Product of Sums (POS). c) Write a Verilog code to implement the design on an FPGA board.



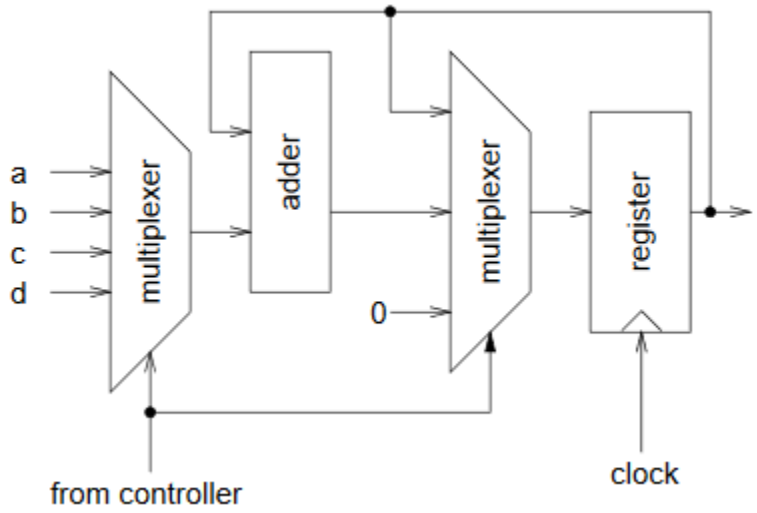
X	Y	Z	F
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

```
module Comb_Ckt(
    input X, Y, Z,
    output F
    );









endmodule
```

Name: _____ CWID: _____ Date: _____

2. (20 Points) Structural Modelling

Verilog code for adder, multiplexer and register is given. Connect all the Verilog modules to create digital circuits shown in the block diagram below.



Half Adder

```
module Half_Adder(
    input A, B,
    output Sum, Carry
    );

 assign Sum = A ^ B;
 assign Carry = A & B;
endmodule
```

Register

```
module Register(
    input D, clock,
    output reg Q
    );
 always @ (D, clock)
 if(clock)
 Q <= D ;
endmodule
```

Multiplexer

```
module Mux(
    input A, B, C, D,
    input [1:0] S,
    output Mux_Out
    );

 assign Mux_Out = (S == 0) ? A : (S == 1) ? B : (S == 2) ? C : D;

endmodule
```

Name: _____ CWID: _____ Date: _____

```
module Top(
    input a,b,c,d,
    input [1:0] Ctrl,
    input Clk,
    output Sys_out
    );









endmodule
```

Name: _____ CWID: _____ Date: _____

3. (10 Points) RTL Combinational Circuit Design

a. Draw the schematic for the Verilog code given below;

```
module abc (a, b, c, d, s1, s0);
    input s1, s0;
    output a, b, c,d;

    not  (s1_, s1), (s0_, s0);

    and (a, s1_, s0_);
    and (b, s1_, s0);
    and (c, s1, s0_);
    and (d, s1, s0);

endmodule
```

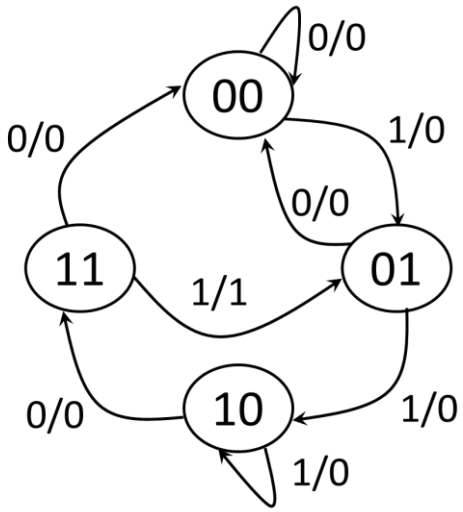
b. Draw the schematic for the Verilog code given below;

```
module Always_Code(
     input a,b,c,
     output reg F
     );

always @(a, b, c)
begin
 F = ~a | b;
 F = ~F;
 F = ~(F & b);
 F = F & c;
 F = F | b;
end

endmodule
```

Name: _____ CWID: _____ Date: _____

4. (25 Points) FSM

The following State Diagram is given.



a) The state table using D - Flip Flops is shown below. Fill it according to the state diagram.

Present State		Input	Next State		Output
A	B	X	A+	B+	Z

b) Write Verilog code to implement the counter (ignore the clock divider)

```
module FSM(
    input Clk, Reset, X,
    output Z
    );









endmodule
```