1. Use CMOS logic to implement the $F = a \text{ XNOR } b$.
   a) Draw the transistor level design with PMOS and NMOS. Assume only $a$ and $b$ signals are available.
   b) Calculate the worst case delay of the circuit. Assume PMOS on resistance is $2R_n$, NMOS is $R_n$, gate capacitance is $C_g$ and load capacitance is $2C_g$.

2. In Lab 3, we use the following circuit to generate a one cycle pulse. Assume that D-flip-flop has propagation delay of 2ns, setup time of 2ns and hold time of 1ns. INV has delay of 1ns and AND gate has delay of 2ns.

   ![Circuit Diagram](image)

   a) Draw the timing diagram for signals $B$, $Q$, $Q'$ and Out. Assume clock cycle is 10ns. Input signals are given as follows. Write the transition times of each signal explicitly in your timing diagram.

   ![Timing Diagram](image)

   b) Calculate the maximum allowable frequency for the circuit.

3. Design a FSM that counts the number of sequences of consecutive 1's in a bit sequence. For example 101010 output=3, 01101110 output=2.
   a) Draw the state diagram.
   b) Draw the state transition table.
   c) Implement the circuit using J-K flip flops and combinational logics.
4. True/False questions

a) For a given sequential circuit, considering clock skew will decrease its maximum allowable frequency.
b) In Verilog the variable with type “reg” means it is a register.
c) In Verilog, both combinational logic and sequential logic can be described behaviorally and structurally.
d) For the given function $F=a’c+c’d+ab$, assume the complement signals are available. Three 2-input AND gates are used for $a’c$, $c’d$ and $ab$ terms respectively. One 3-input OR gate is used for generating $F$ from three outputs of AND gates. This circuit is hazard free.
e) For the module of sine reader in Lab 3, it needs to access the sine ROM with a given step size. The step size has the same bit width as the address of the sine ROM.
f) In Lab 4, the instruction decoder generates all the signals that needed by regfile module.

5. Multiple choice questions

a) Given two modules defined as follows, which statement is correct after the first rising edge of the clock? Assume that initial values are $b=0$, $a=1$.

```
1 module A(clk,a,c);
2     input clk;
3     input a;
4     output c;
5
6     wire clk;
7     wire a;
8     reg c;
9     reg b;
10
11    always @ (posedge clk )
12    begin
13      b = a;
14      c = b;
15    end
16
17   endmodule
```

```
1 module B(clk,a,c);
2     input clk;
3     input a;
4     output c;
5
6     wire clk;
7     wire a;
8     reg c;
9     reg b;
10
11    always @ (posedge clk )
12    begin
13      b <= a;
14      c <= b;
15    end
16
17   endmodule
```

(A) In both module A and B, $c=b=1$.
(B) In module A, $c=b=1$. In module B, $c=0$, $b=1$.
(C) In module A, $c=0$, $b=1$. In module B, $c=b=1$. 
(D) In both module A and B, c=0, b=1.

b) Assume you want assign c to b if a is 0, and assign c’ to b if a is 1. Which of the following Verilog code implements this functionality correctly?

(A) module A (a, b, c);
    input a, c;
    output b;
    assign b=a? ~c : c;
endmodule

(B) module B (a, b, c);
    input a, c;
    output b;
    if (a==0)
        assign b=c;
    else
        assign b=~c;
endmodule

(C) module B (a, b, c);
    input a, c;
    output b;
    always@(negedge a)
        assign b=c;
    always@(posedge a)
        assign b=~c;
endmodule

(D) module B (a, b, c);
    input a, c;
    output b;
    wire c_bar, a_bar;
    not n0(c, c_bar);
    not n1(a, a_bar);
    and n2(a_bar, c, b);
    and n3(a, c_bar, b);
endmodule