

Farnam Adelkhani

915815724

## Engineering 378 HW #2

Fall 2016 - SFSU

---

---

---

---

---

---

---

---

---

---

---

---



### Homework two problems 2.3 2.4 2.6 2.8 2.10 2.11

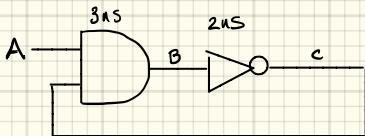
#3

2.3 Given the concurrent Verilog statements:

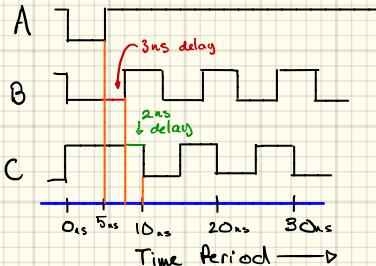
```
assign #3 B = A && C;
assign #2 C = !B;
```

- (a) Draw the circuit the statements represent.
- (b) Draw a timing diagram if initially  $A = B = 0$  and  $C = 1$ , and  $A$  changes to 1 at time 5ns.

(a)

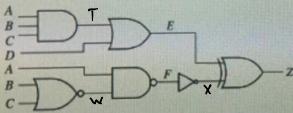


(b)



#4

2.4 Write a Verilog description of the following combinational circuit using concurrent statements. Each gate has a 5-ns delay, excluding the inverter, which has a 2-ns delay.



```
assign #5 T = A and B and C;
```

```
assign #5 w = B nor C;
```

```
assign #6 E = T or D;
```

```
assign #5 F = A nand D;
```

```
assign #2 X = not F;
```

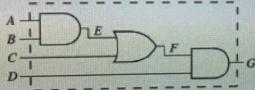
```
assign #5 Z = E xor X;
```

# Q

2.6 Write Verilog code for the following circuit. Assume that the gate delays are negligible.

(a) Using concurrent statements.

(b) Using an always block with sequential statements. No latches should be generated.



(a)

Module ThreeGates (A,B,C,D,G);

```

input A,B,C,D;
output G;
wire E,F;
assign E = A&&B;
assign F = E || C;
assign G = F&&D;
end module
    
```

(b)

Module ThreeGates (A,B,C,D,G);

```

input A,B,C,D;
output G;
wire E,F;
always @ (A,B,C,D)
begin
    E = A&&B;
    F = E || C;
    G = F&&D;
end
end module
    
```

# 8

2.8 (a) What device does the following Verilog code represent?

```

always @(CLK, Clr, Set)
begin
    if(Clr == 1'b1)
        Q <= 1'b0;
    else if(Set == 1'b1)
        Q <= 1'b1;
    else if(CLK == 1'b0)
        Q <= D;
    else begin
    end
end
    
```

(b) What happens if Clr = Set = 1 in the device in part a?

(a) A D-flip flop

output triggered when clk is 0

clr → output low

set → output high

(b)

Code executes @ A to Clr, set or clk  
... but priority is Clr → Set → CLKSince priority of clr is higher the output will be low.

2.10 An M-N flip-flop responds to the falling clock edge as follows:

If  $M = N = 0$ , the flip-flop changes state.

If  $M = 0$  and  $N = 1$ , the flip-flop output is set to 1.

If  $M = 1$  and  $N = 0$ , the flip-flop output is set to 0.

If  $M = N = 1$ , no change of flip-flop state occurs.

The flip-flop is cleared asynchronously if  $CLRn = 0$ .

Write a complete Verilog module that implements an M-N flip-flop.

\* 10

```
Module MNFlipFlop (M, N, CLRn, CLK, Q, QB);
    input M, N, CLK, CLRn;
    output Q, QB;
    always @ (CLK or M or N or CLRn)
        begin
            if(CLRn) begin // CLRn set → Q = 0 & QB = 1
                Q = 0;
                QB = 1;
            end
            else if (negedge CLK) begin // flip flop working on neg edge.
                if (M == 0 && N == 0) begin
                    Q = !Q;
                    QB = !QB; // if M=0 & N=0 → state change
                end
                if (M == 0 && N == 1) begin
                    Q = 1;
                    QB = 0; // if M=0 & N=1 → Q=1 & QB=0
                end
                if (M == 1 && N == 0) begin
                    Q = 0;
                    QB = 1; // if M=1 & N=0 → Q=0 QB=1
                end
                if (M == 1 && N == 1) begin
                    Q = Q;
                    QB = QB; // if M=1 & N=1 → Q=Q QB=QB
                end
            end
        end
    end
endmodule
```

# 11

- 2.11 A DD flip-flop is similar to a D flip-flop, except that the flip-flop can change state ( $Q^+ = D$ ) on both the rising edge and the falling edge of the clock input. The flip-flop has a direct reset input,  $R$ , and  $R = 0$  resets the flip-flop to  $Q = 0$  independent of the clock. Similarly, it has a direct set input,  $S$ , that sets the flip-flop to 1 independent of the clock. Write a Verilog description of a DD flip-flop.

Active low reset & active high set independent of clk

Module DDflipflop (R,S,clk,Q,QB,O);

input

output

reg

always @ (posedge clk or negedge clk or !R or S)

begin

if (!R) begin

$Q \leftarrow 1'b0$

$QB \leftarrow 1'b1$

end

else if (S) begin

$Q \leftarrow 1'b1$

$QB \leftarrow 1'b0$

end

else begin

$Q \leftarrow 0$

$QB \leftarrow !D$

end

end

end module

// if  $R=0 \rightarrow Q=0 \& QB=1$

// if S is set  $\rightarrow Q=1 \& QB=0$