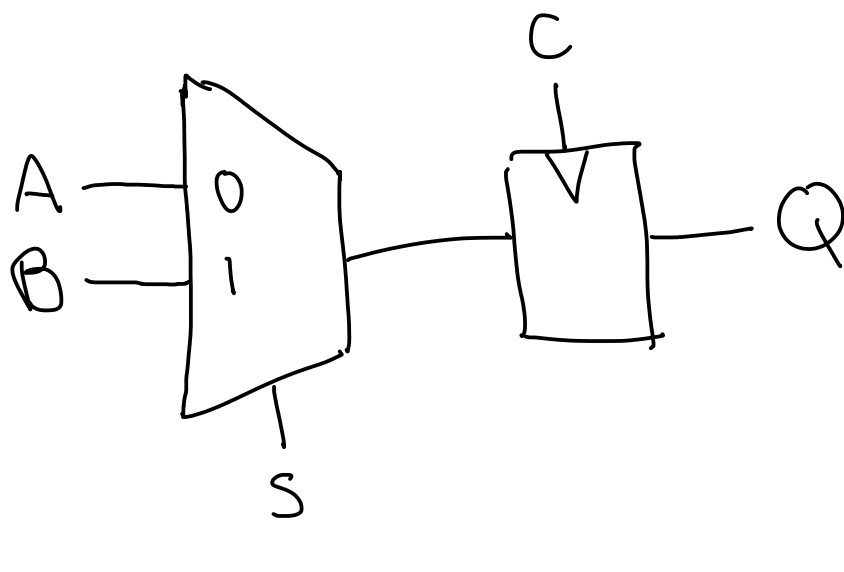


## Homework 2

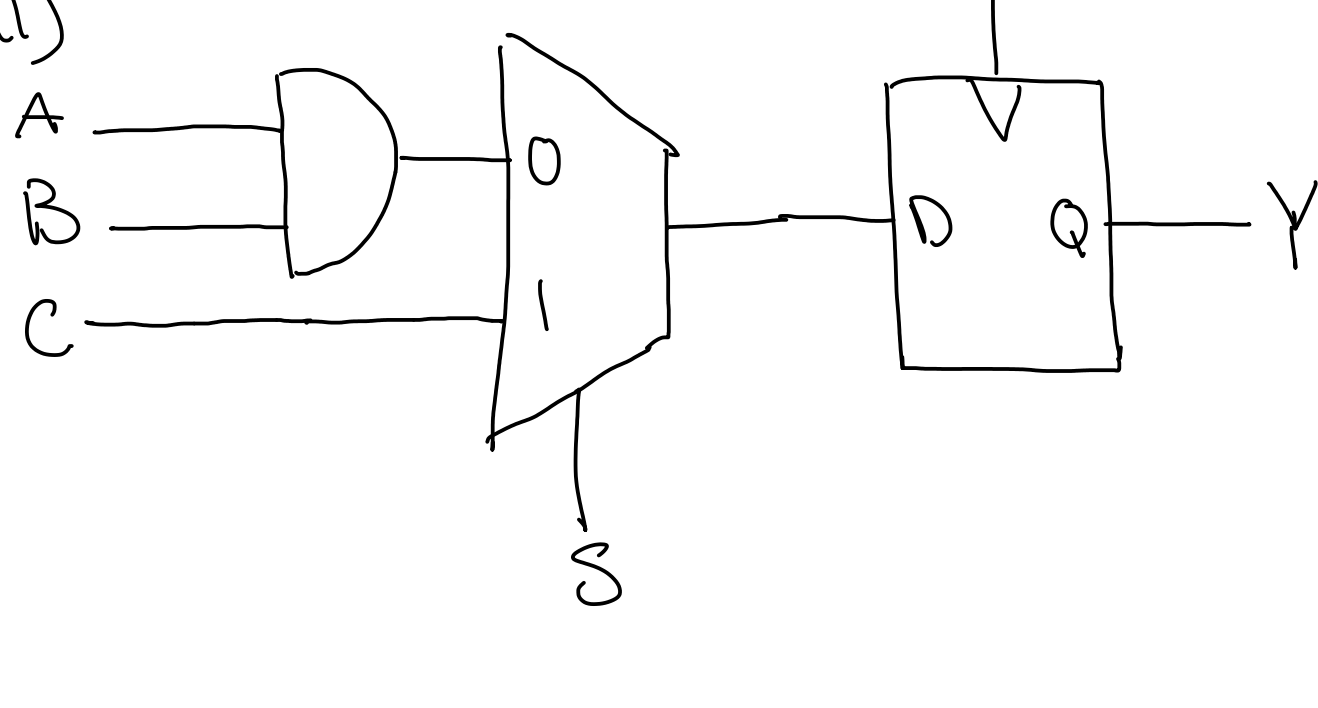
Thursday, July 14, 2022

9:51 AM

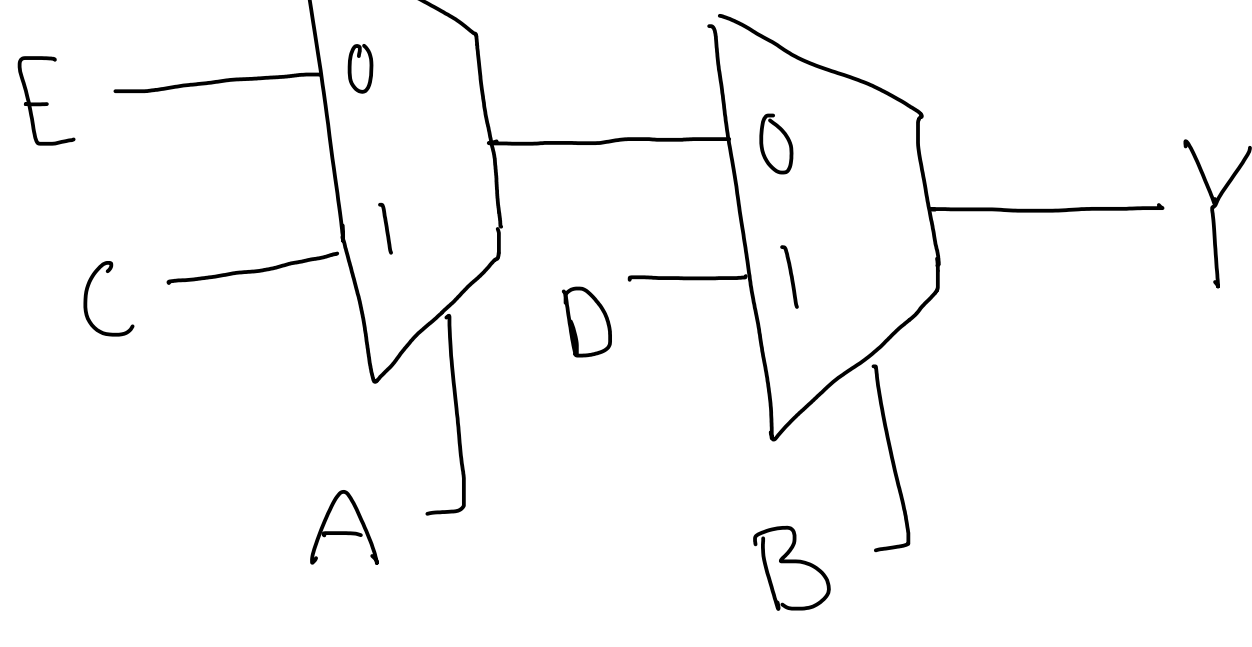
i)



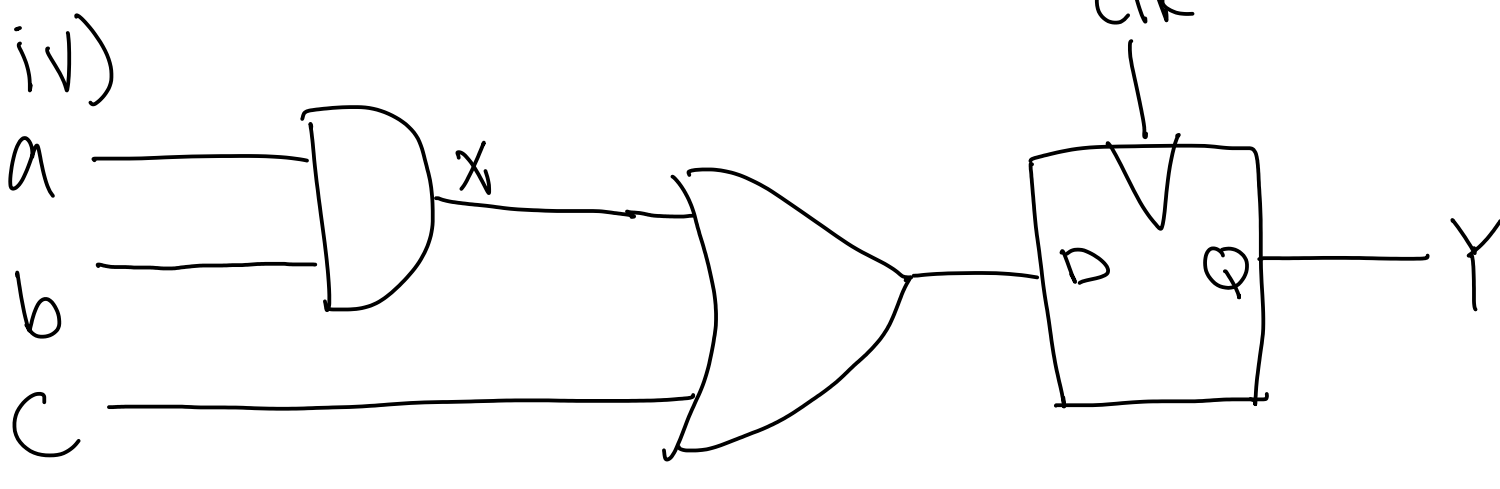
ii)



iii)



iv)



Q2:

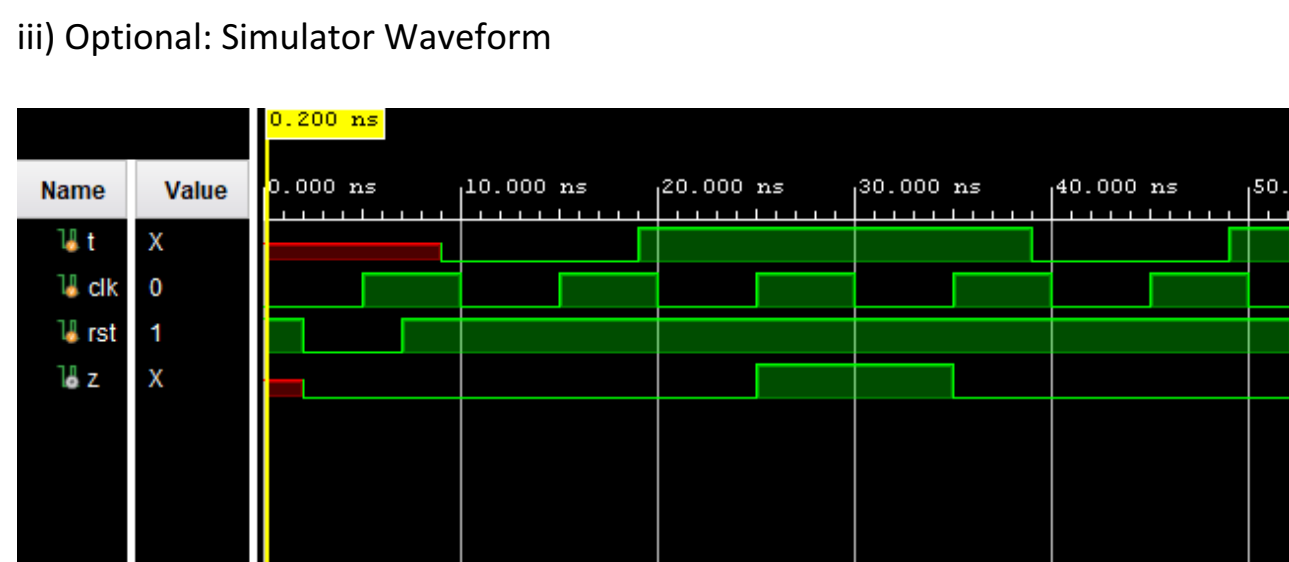
i)

```
23 module tflipflop(  
24     input t,  
25     input clk,  
26     input rst,  
27     output z  
28 );  
29  
30 reg CS, NS;  
31 parameter zero = 0;  
32 parameter one = 1;  
33  
34 always @ (posedge clk, negedge rst) begin  
35     if (rst == 1'b0) begin NS <= 0; CS <=0; end  
36     else begin  
37         CS <= NS; end  
38     end  
39  
40  
41 always@(CS, t) begin  
42     case(CS)  
43         zero: NS = t? one : zero;  
44         one: NS = t? zero : one;  
45         default: NS = zero;  
46     endcase  
47 end  
48  
49 assign z = (CS == one);  
50 endmodule  
51
```

ii)

```
23 module tflipflop_tb;  
24     reg t, clk, rst=1;  
25     wire z;  
26  
27     tflipflop UUT(  
28         .rst(rst),  
29         .z(z),  
30         .t(t),  
31         .clk(clk));  
32  
33     initial begin  
34         clk = 0;  
35         forever #5 clk = ~clk;  
36     end  
37  
38     $monitor("z = ", z);  
39     #2 rst = 0; #5 rst = 1;  
40     #2 t = 0; #10 t = 1; #20 t = 0; #10 t = 1;  
41     #10 $finish;  
42 end  
43  
44 endmodule  
45
```

iii) Optional: Simulator Waveform



Q3:

i) Table

CNT[3]	CNT[2]	CNT[1]	CNT[0]
0	0	0	0
0	0	0	1
0	0	1	1
0	1	1	1
1	1	1	0
1	1	0	0
1	0	0	0
0	0	0	0

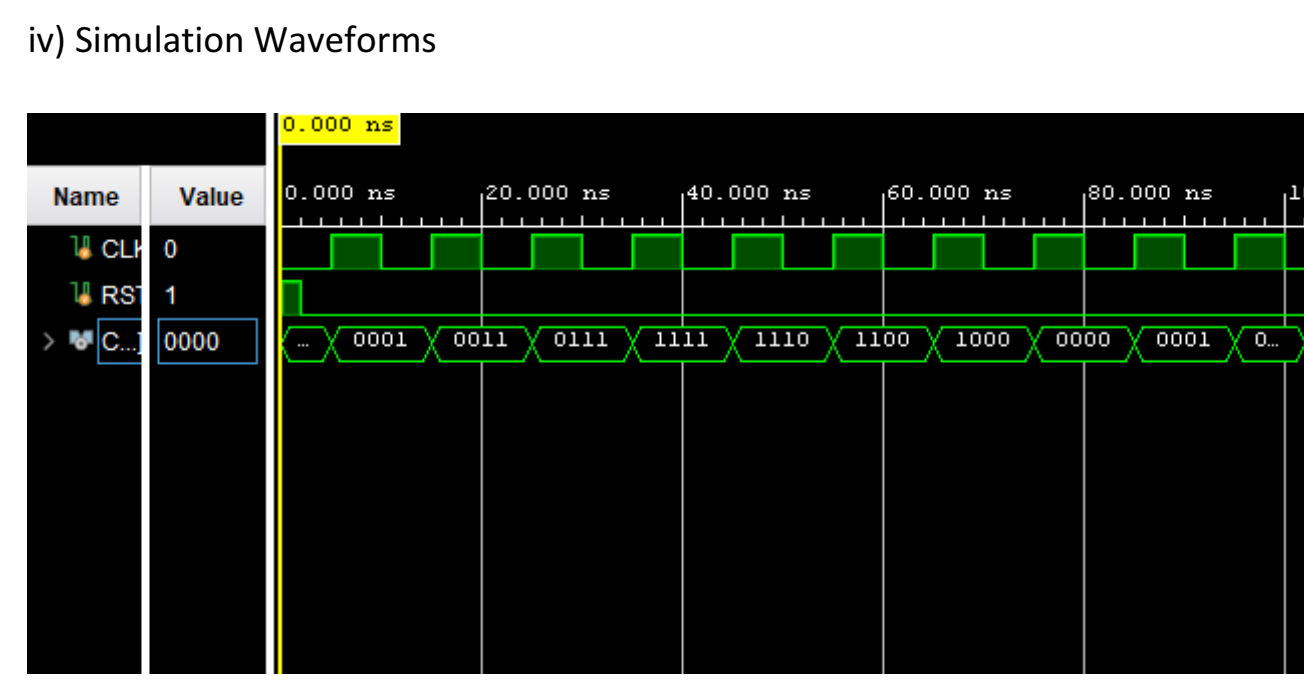
ii) Verilog

```
22 module QCounter #(parameter wordsize = 4) (  
23     input CLK,  
24     input RST,  
25     output reg [wordsizel-1:0]CNT  
26 );  
27  
28 always @ (posedge CLK, posedge RST)  
29     if (RST) CNT<= 0;  
30     else begin  
31         CNT <= {CNT[wordsizel-2:0], ~CNT[3]};  
32     end  
33 end  
34  
35 endmodule  
36
```

iii) Test Bench

```
22 module QCounter_tb;  
23     reg CLK, RST=1;  
24     wire [3:0]CNT;  
25  
26     QCounter UUT(  
27         .CLK(CLK),  
28         .RST(RST),  
29         .CNT(CNT)  
30     );  
31  
32     initial begin  
33         CLK=0;  
34         forever #5 CLK=~CLK;  
35     end  
36  
37     $monitor("CNT = ", CNT);  
38     #2 RST =0;  
39     #100; $finish;  
40 end  
41  
42 endmodule  
43
```

iv) Simulation Waveforms



Q4:

```
13 module moore_fsm (  
14     input D,  
15     input clk,  
16     output z  
17 );  
18 parameter a = 2'd0, b = 2'd1, c = 2'd2, d = 2'd3;  
19  
20 reg [1:0] CS, NS;  
21 always @(posedge clk) //edge sensitive behavior  
22     begin CS<= NS; end  
23 always @(CS, D) begin //level-sensitive behavior  
24     NS = a; //default state  
25     case (CS)  
26         a : NS = D ? c:b;  
27         b : NS = D ? d:c;  
28         c : NS = D ? d:b;  
29         d : NS = D ? a:c;  
30     endcase  
31 end  
32 assign z = (CS == b | c);  
33 endmodule
```