May 8, 2017

CSE 111

~~Final~~ Exam 3 : is on May 12, 2017 during lecture.

- designated seating

Exam3 Review.ppt on webpage ubleams

We'll review today & Wed.

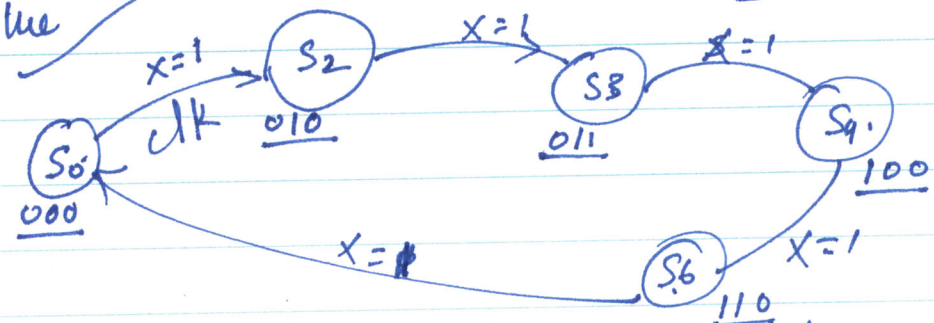
May 8, 2017
Spring 2017

CSE241

Q4: Design of a counter (0, 2, 3, 4, 6)
use (i) D-FF. (ii) T-FF

We want: circuit

Coding of the states



Counter advances ~~when the clock~~ with every clock pulse. Eliminates x,

5 states [Flag 5] no x; clock advances the counter.

Current state			next state			FF input		
A(t)	B(t)	C(t)	A(t+1)	B(t+1)	C(t+1)	DA	DB	DC
S_0 0	0	0	0	1	0	0	1	0
S_2 0	1	0	0	1	1	0	1	1
S_3 0	1	1	1	0	0	1	0	0
S_4 1	0	0	1	1	0	1	1	0
S_6 1	1	0	0	0	0	0	0	0

Verl

S1 S5 S7

$DA(A,B,C) = ?$
$DB(A,B,C) = ?$
$DC(A,B,C) = ?$ $\Sigma(S_2) = \Sigma(m_2)$ $\Sigma(2)$

3 K-maps

May 8, 2017

A \ BC	00	01	11	10
0	0	?	1	0
1	1	?	?	0

$DA(A,B,C)$: ⇓ using unused states as don't cares

A \ BC	00	01	11	10
0	0	X	1	0
1	1	X	X	0

$DA(A,B,C) = C + AB'$

$DB(A,B,C) = ?$ $\Sigma(0,1,2,4)$

A \ BC	00	01	11	10
0	1	X		1
1	1	X	X	

$DB(A,B,C) = B' + A'C'$

A \ BC	00	01	11	10
0		X		1
1		X	X	

$DC(A,B,C) = A'BC'$

group it in 8's
4's
2's
1's

don't have to cover X's
You have to cover only the 1's

X = 0 or 1

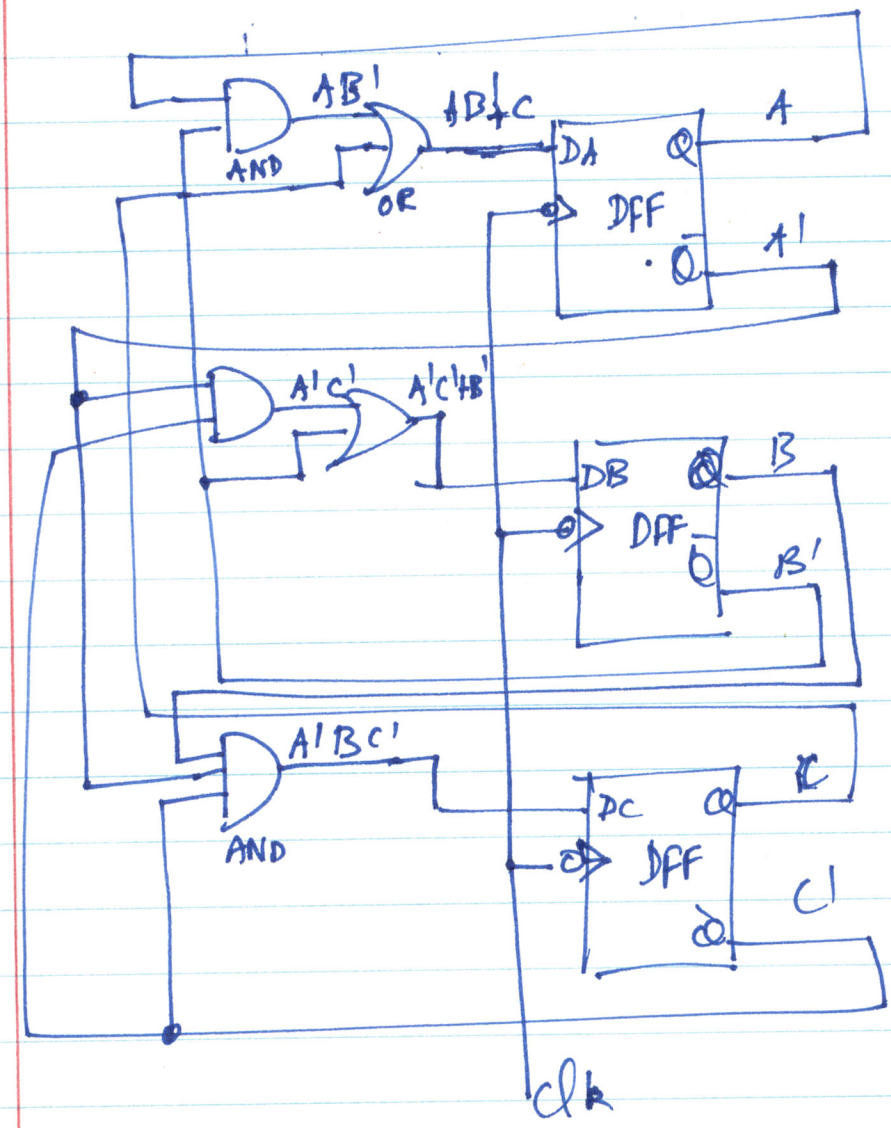
~~2~~

May ~~8~~, 201~~7~~

3

May 8, 2017

DA (A, B, C)



May 8, 2017

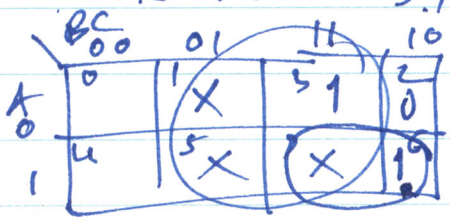
	current state			next state			FF inputs		
	A(t)	B(t)	C(t)	A(t+1)	B(t+1)	C(t+1)	TA	TB	TC
S_0	0	0	0	0	1	0	0	1	0
S_2	0	1	0	0	1	1	0	0	1
S_3	0	1	1	1	0	0	1	1	1
S_4	1	0	0	1	1	0	0	0	0
S_6	1	1	0	0	0	0	1	1	0

$TA(A,B,C) = \sum(3,6)$

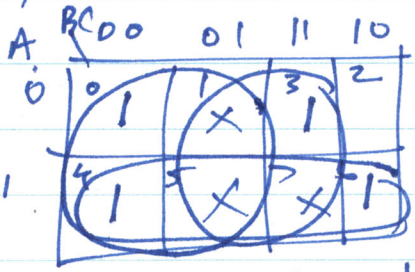
$TB(A,B,C) = \sum(0,3,4,6)$

$TC(A,B,C) = \sum(2,3)$

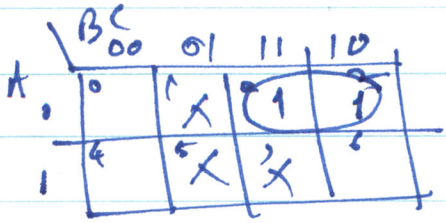
Assume S_5, S_1, S_7 don't cares.



$TA = C + AB$



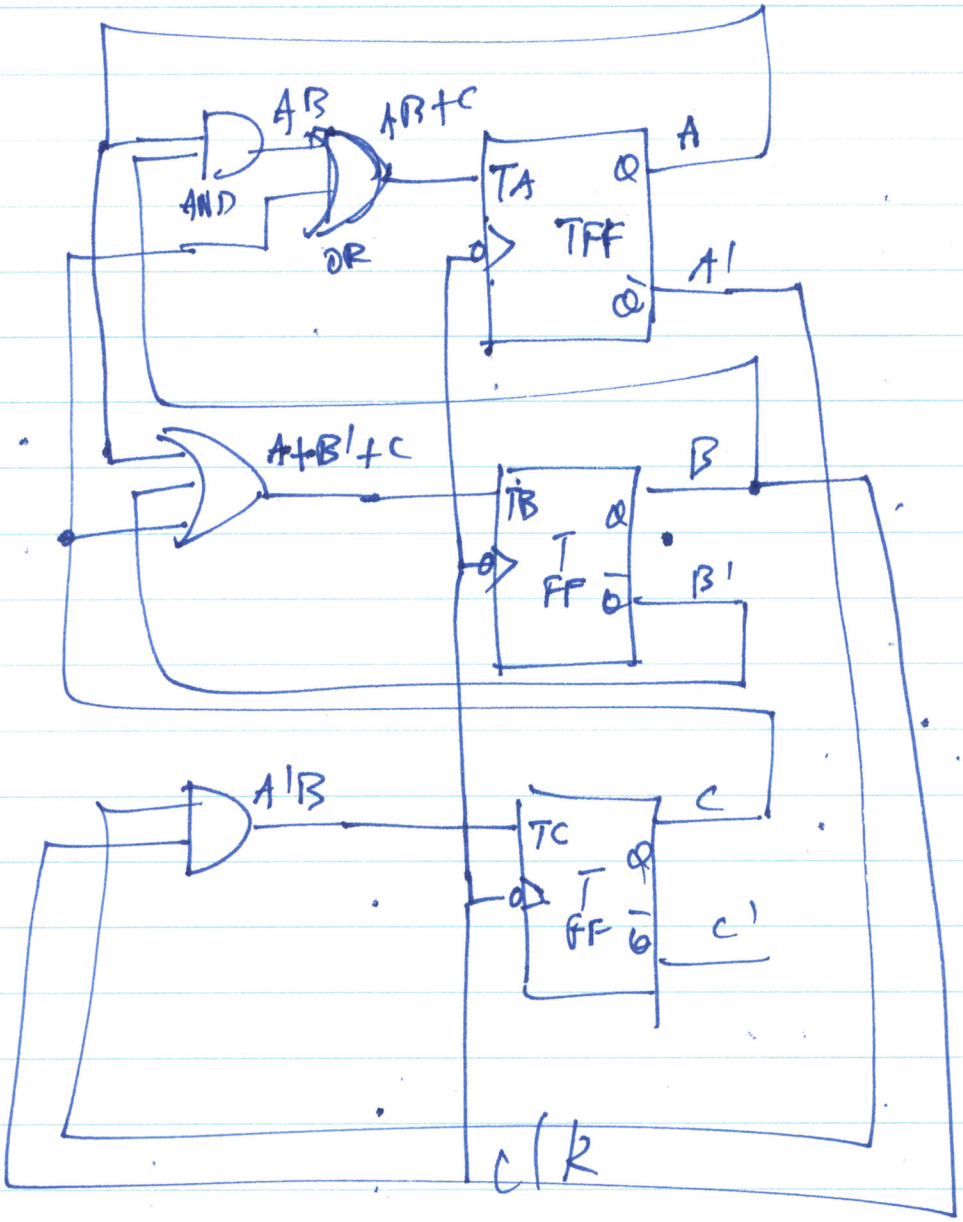
$TB = A + B' + C$



$TC = A'B$

don't cares are <u>not</u> used

May 8, 2013



CSE 2411

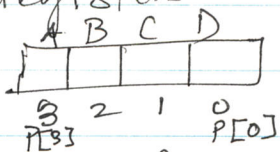
May 10, 2017

F1 F2 F3 F4 PB7 PB03

```
module review1 (W, X, Y, Z, A, B, C, D);
 input A, B, C, D;
 output W, X, Y, Z;
  wire t1, t2, t3, t6, t7, t8; // intermediates
  // gates - traverse top to bottom, left to right
  NOT g1 (t1, B);
  NOT g2 (t2, C);
  OR  g3 (t3, C, D);

  XNOR g4 (Y, C, D);
  NOT  g5 (Z, D);

  AND g6 (t6, B, t3);
  AND g7 (t7, t1, t3);
  AND g8 (t8, B, t2, Z);

  OR g9 (W, A, t6);
  OR g10 (X, t7, t8);
end module
```

May 17, 2017

```
module review1_tb;
  // test pattern registers
  reg [3:0] P;
```



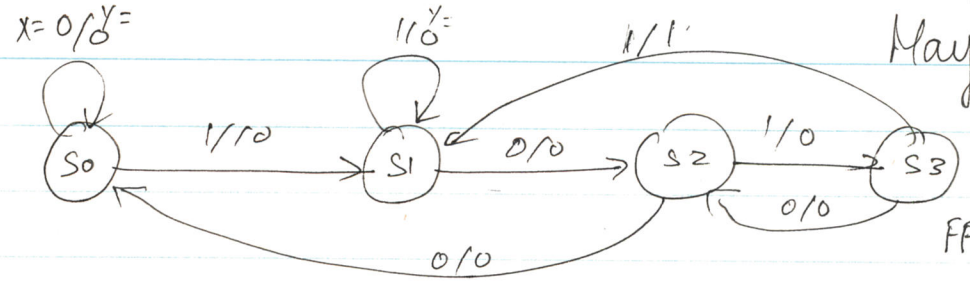
$2^4 = 16$ patterns

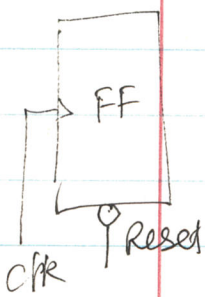
```
  wire F1, F2, F3, F4; // four functions
                       // corresponding to W, X, Y, Z
  // instantiate the circuit to be tested
  review1 r1 (P[3], P[2], P[1], P[0], F1, F2, F3, F4);
             (F1, F2, F3, F4, P[3], P[2], P[1], P[0]);
                  // order of paramter is
                  // important
  initial begin
    P = 4'b0000;
    repeat (15)
      #1000 P = P + 1'b0001;
  end
  initial begin
    $monitor ("ABCD = %b %b %b %b %b",
                F1, F2, F3, F4, P);
    // $monitor ("format string", variables);
    $monitor("ABCD = %b %b %b %b %b", F1, F2, F3, F4,
                P, F1, F2, F3, F4);
  end
endmodule
```

Single print out

ABCD = 0000 F1 = 1 F2 = 0 F3 = 0 F4 = 1

May 10, 2017

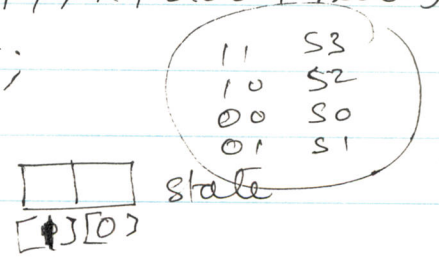




↳ module review2 (input x, input clock, input reset);

alternatively module review2 (x, clock, reset);
input x, clock, reset;

Step1: module review2 (state, y, x, clock, reset);
input x, clock, reset;
output y, state;
reg [1:0] state;



Step2 parameter S0 = 2'b00, S1 = 2'b01, S2 = 2'b10, S3 = 2'b11;

// (Symbolic constants //

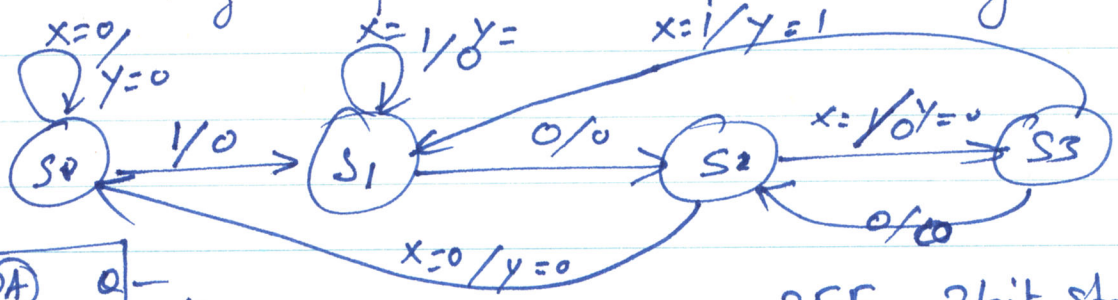
// Synchronous Reset, clock

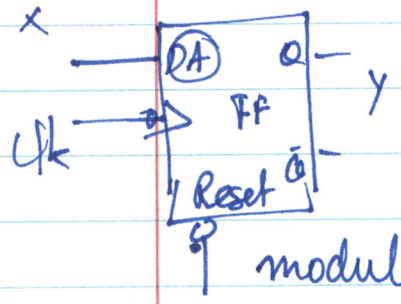
Step3 always @ (posedge clock, negedge reset)

Step4 if (reset == 0) begin state = S0; Y=0; end
else case (state)
S0 : if (x==0) state = S0; else state = S1;
S1 : if (x==0) state = S2; else state = S1;
S2 : if (x==0) state = S0; else state = S3;
S3 : if (x==0) state = S2; else begin state = S1; Y=1; end

endcase
end module

Verilog Synthesis: Seq circuit May 12, 2017





2FF = 2bit state

↑ ↑ ↓ ↓ ↓

module review2(state, Y, x, clock, reset);

Slip1
input x, clock, reset;
output Y, state;
reg [1:0] state;

Slip2
parameter S0 = 2'b00, S1 = 2'b01, S2 = 2'b10, S3 = 2'b11;
// symbolic constants for states

Slip3
// synchronous clock event,
// asynchronous reset event — attend to these
Y = 0;
always @ (posedge clock, negedge reset)
if (reset == 0) begin state = S0; end

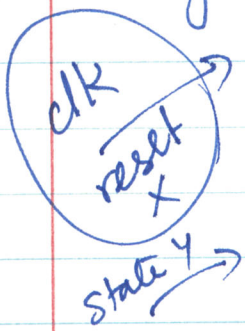
Slip4
else case (state)
S0: if (x == 0) state = S0; else state = S1;
S1: if (x == 0) state = S2; else state = S1;
S2: if (x == 0) state = S0; else state = S3;
S3: if (x == 0) state = S2; else begin
state = S1;
Y = 1;
end

end case
endmodule

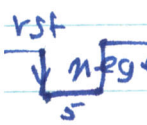
May 12, 2017

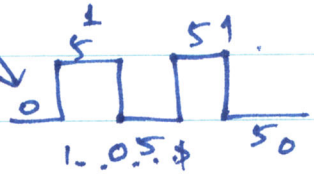
Goal of test bench:

1. Generate the input (patterns) needed for testing the circuit
2. instantiate the tested circuit
3. monitor the output




```
module review2_tb;
    reg x, clk, rst;
    wire y, s[1:0];
    review2 r2(s, y, x, clk, rst);
    // Set simulation time / experiment
    initial #200 $finish;
    // generate rst, clk, x
    initial begin
        rst = 0; clk = 0;
        #5 rst = 1;
        repeat (16)
        #5 clk = ~clk;
    end
    initial begin x = 1'b0;
        repeat (8) #10 x = ~x; end
    //monitoring
    initial begin
        $monitor("x = %d State %d y = %b",
        $dumpfile("xyz.vcd");
        $dumpvars; end endmodule
```





x, s, y);

May 12, 2017

```
$monitor(
"clk= %b  rst= %b  x= %b  state= %d
y= %b", clk, rst, x, s, y);
```

format string

variables

"reuse"

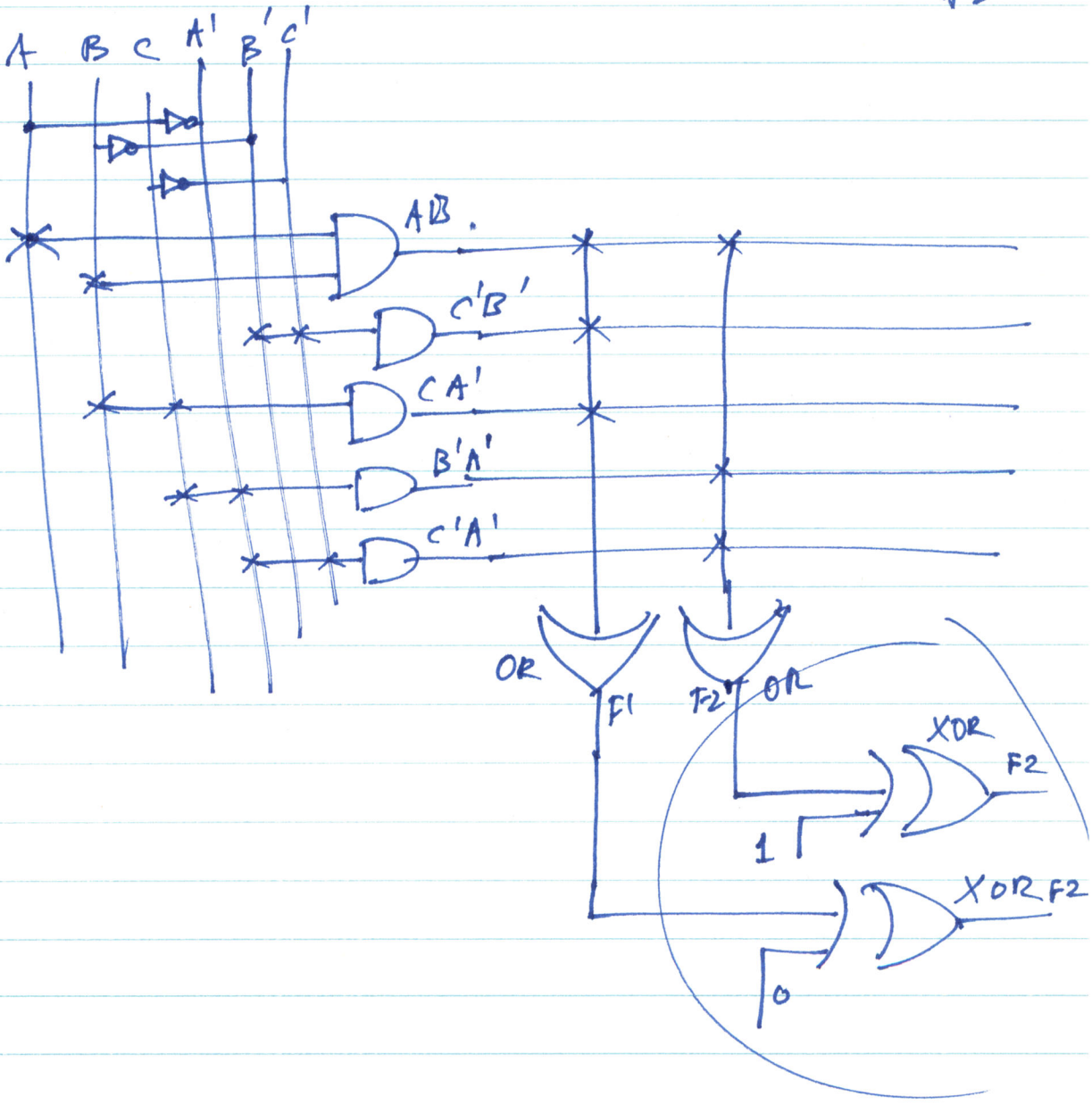
May 12, 2015

$F_1(A, B, C) = AB + C'B' + CA'$

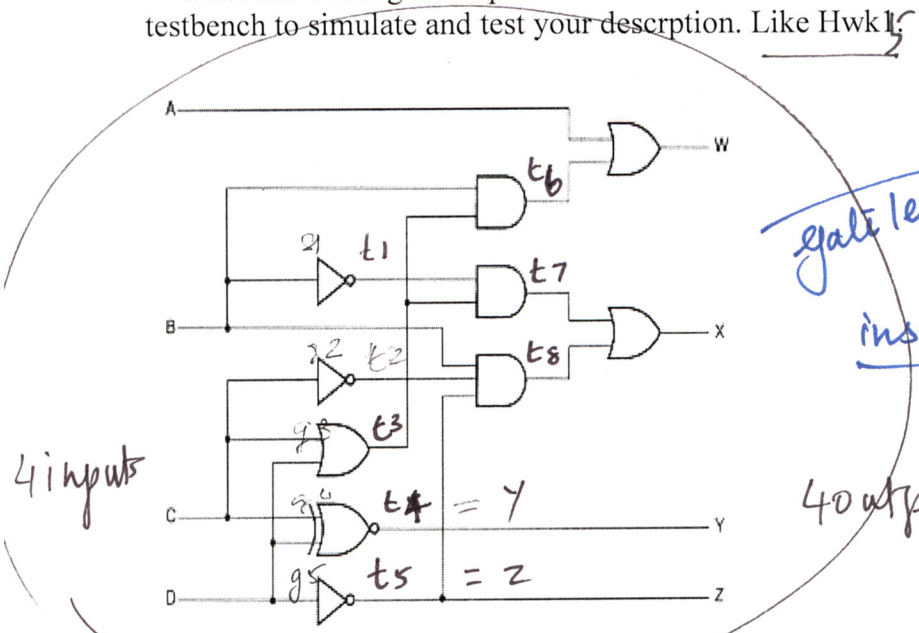
$F_2(A, B, C) = (AB + B'A' + CA')'$

NOT, AND, OR, XOR

F_2'

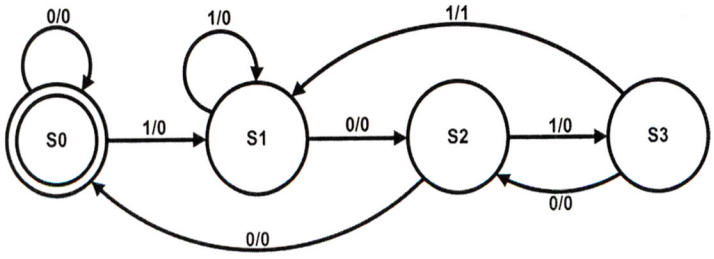


- Write the Verilog description of the combinational circuit given below and provide the testbench to simulate and test your descrption. Like Hwk1.



- Write the Verilog Description of the sequential circuit given below and provide the test bench.

state machine to detect the sequence 1011



Verilog Description

Combinational Circuit

gate level Description + test bench

Sequential Circuit

behavioral Description + test bench