19EC302: Digital System Design

Department of Electronics and Communication Engineering

NITTE
DEEMED TO BE
UNIVERSITY

NMAM INSTITUTE OF TECHNOLOGY

(An Autonomous Institution affiliated to VTU, Belagavi)

Nitte – 574110, Karkala, Udupi District, Karnataka, India

Sequential Circuit Design - II

UNIT-5

Clocked synchronous sequential circuits

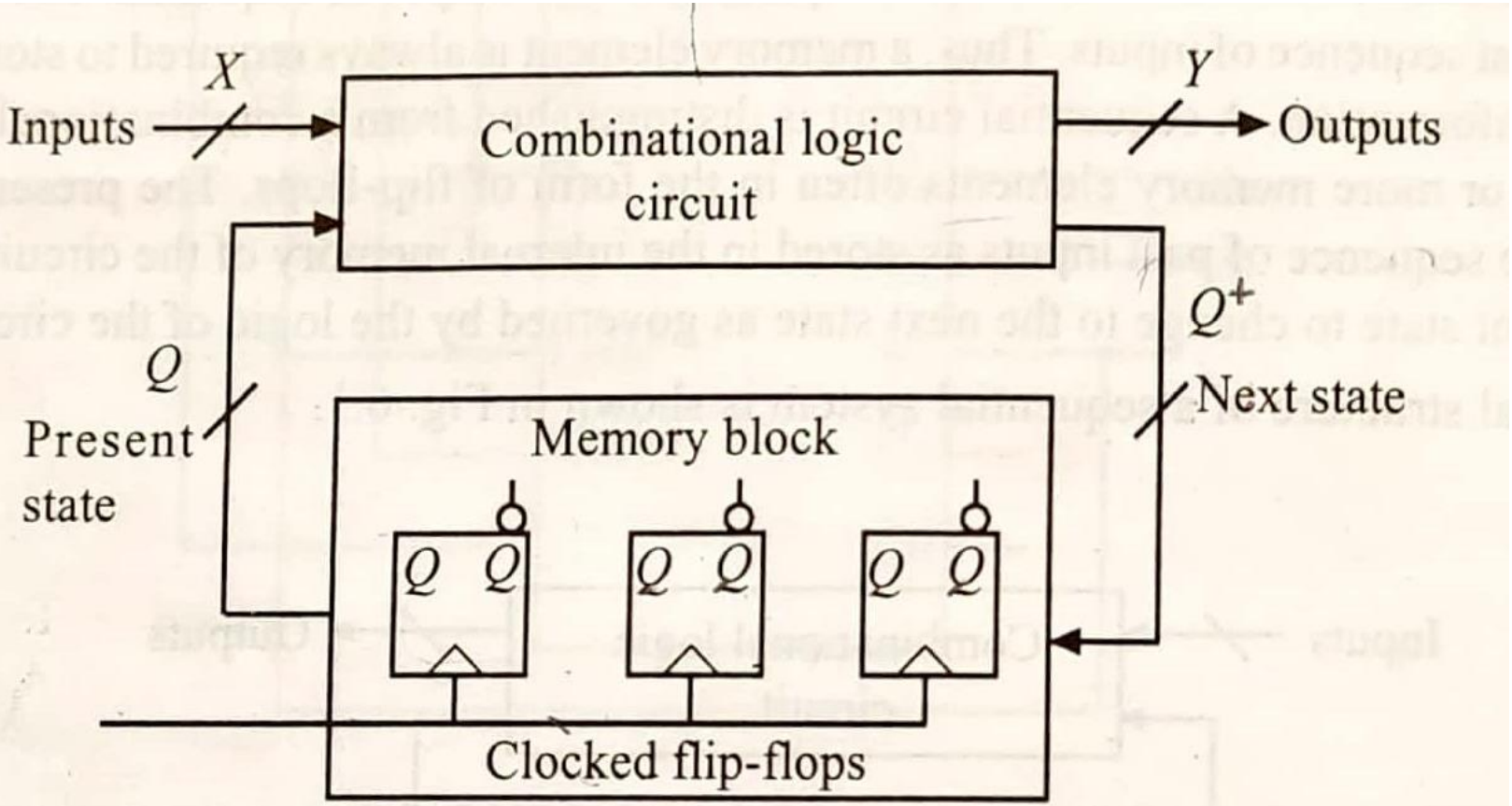


Fig. Configuration of clocked synchronous sequential circuits

Mealy model

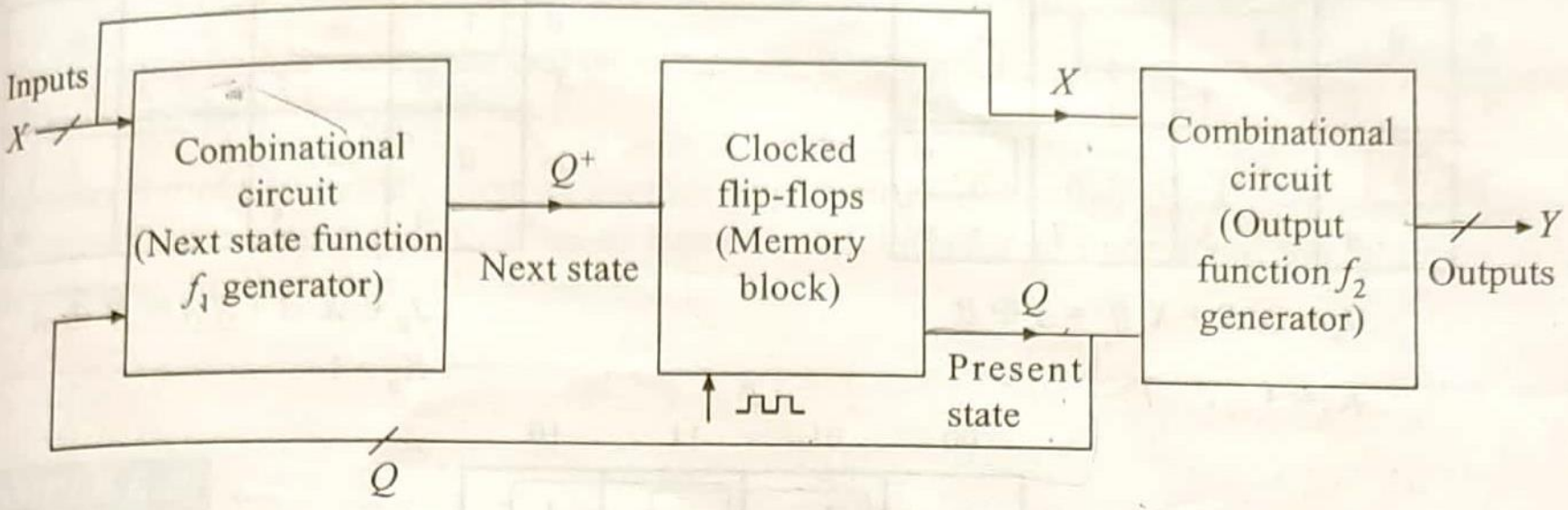


Fig. Mealy model of a clocked synchronous sequential circuits

$Q = \{Q_0, Q_1, \ldots Q_n\}$ be the present state of the circuit,

$Q^+ = \{Q_0^+, Q_1^+, \ldots Q_n^+\}$ be the next state of the circuit,

$X = \{X_1, X_2 \ldots X_n\}$ be the present input and

$Y = \{Y_1, Y_2 \ldots Y_n\}$ be the outputs.

The next state is a Boolean function of the present inputs and the present state.

$$\therefore \quad Q^+ = f_1 \ (X, Q)$$

and the outputs are also a function of present inputs and the present state

$$\therefore \quad Y = f_2 \ (X, Q)$$

sign a synchronous circuit using positive edge triggered *JK* flip-flops with minimal combinational ing to generate the following sequence.

$$0 - 1 - 2 - 0 \text{ if input } X = 0 \quad \text{and}$$

$$0 - 2 - 1 - 0 \text{ if input } X = 1$$

Provide an output which goes high to indicate the non-zero states in the 0 – 1 – 2 – 0 sequence. Is a Mealy machine ?

Let us write the excitation table as shown in Table

Excitation table for Example

Cell no.	Input X	Present state A	B	Next state A^+	B^+	Flip-flop inputs J_A	K_A	J_B	K_B	Output Y
0	0	0	0	0	1	0	–	1	–	0
1	0	0	1	1	0	1	–	–	1	1
2	0	1	0	0	0	–	1	0	–	1
3	0	1	1	–	–	–	–	–	–	–
4	1	0	0	1	0	1	–	0	–	0
5	1	0	1	0	0	0	–	–	1	0
6	1	1	0	0	1	–	1	1	–	0
7	1	1	1	–	–	–	–	–	–	–

The Karnaugh maps for simplifying flip-flop inputs and the output are shown in Fig.

X \ AB	00	01	11	10
0	0 (0)	1 (1)	– (3)	– (2)
1	1 (4)	0 (5)	– (7)	– (6)

$$J_A = \bar{X}B + X\bar{B} = X \oplus B$$

$$K_A = 1$$

X \ AB	00	01	11	10
0	1 (0)	– (1)	– (3)	0 (2)
1	0 (4)	– (5)	– (7)	1 (6)

$$J_B = \bar{X}\bar{A} + XA = \overline{X \oplus A}$$

$$K_B = 1$$

X \ AB	00	01	11	10
0	0 (0)	1 (1)	– (3)	1 (2)
1	0 (4)	0 (5)	– (7)	0 (6)

$$Y = \bar{X}B + \bar{X}A = \bar{X}(A + B)$$

Fig. Karnaugh maps related to Table 6.1

The implementation is shown in Fig.

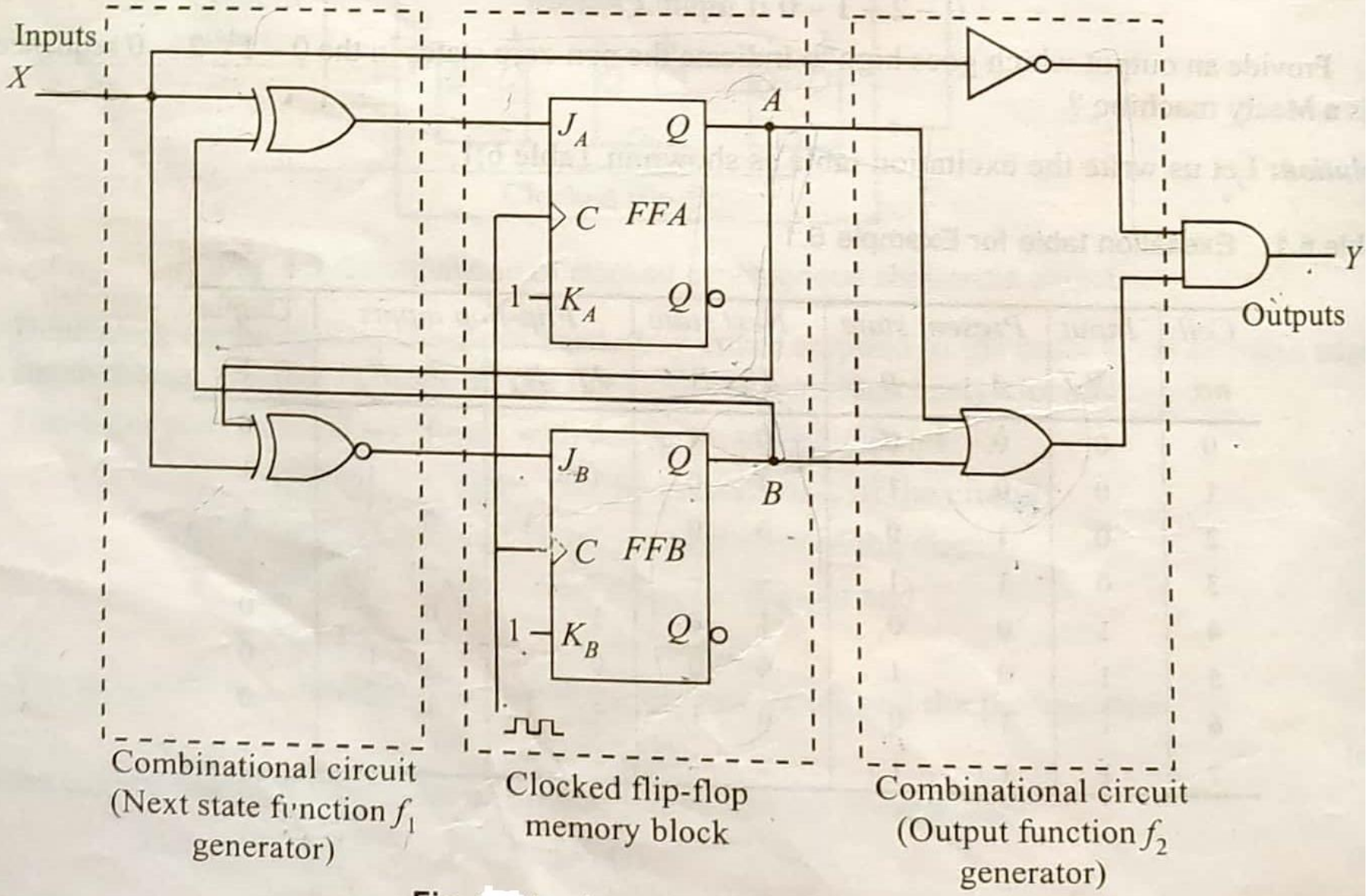


Fig. Implementation of Example

Here,

$$Q = \{A, B\}, Q^+ = \{A^+, B^+\}$$

$$Q^+ = f_1(X, Q) = f_1(X, A, B)$$

$$Y = f_2(X, Q) = f_2(X, A, B) = \overline{X}(A + B)$$

Observe that both the next state and the output are Boolean functions of the input and the present tate. This is indeed a Mealy machine.

Moore model

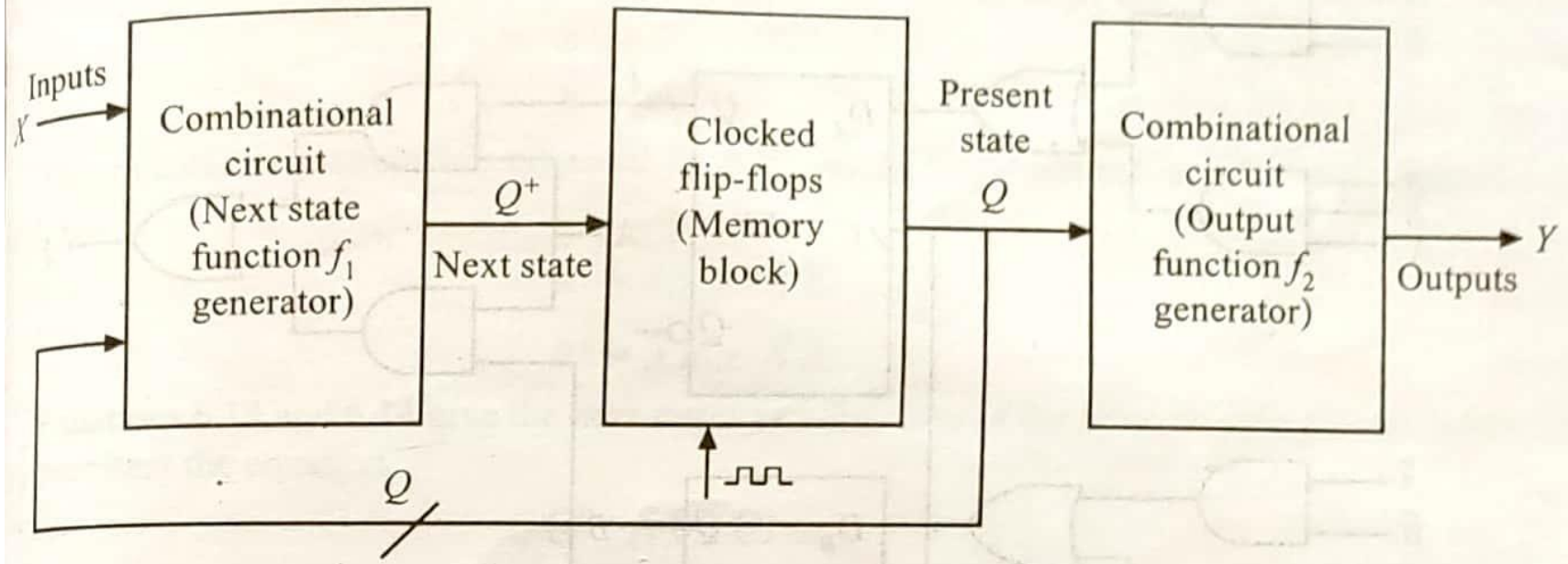


Fig. Moore model of a clocked synchronous sequential circuit

The Moore machine is the realization of the following Boolean functions.

$$Q^+ = f_1\,(X, Q)$$

$$Y = f_2\,(Q)$$

The characteristic feature of the Moore machine is that the output is a function of only the ent state.

EXAMPLE

Repeat Example with the output now to go high whenever the circuit is in non-zero states irrespective of the sequence.

Solution: The synchronous circuit design remains same with exception to the output combinational circuit. The truth table for the output is shown and the Karnaugh map are shown in Fig.

Cell no.	*Input X*	*Present state A*	*B*	*Output Y*
0	0	0	0	0
1	0	0	1	1
2	0	1	0	1
3	0	1	1	–
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	–

(a)

	AB 00	01	11	10
0 (*X*)	0 / 0	1 / 1	– / 3	1 / 2
1	0 / 4	1 / 5	– / 7	1 / 6

$Y = A + B$

Fig. Truth table and Karnaugh map for the output function of Example

The implementation is shown in Fig.

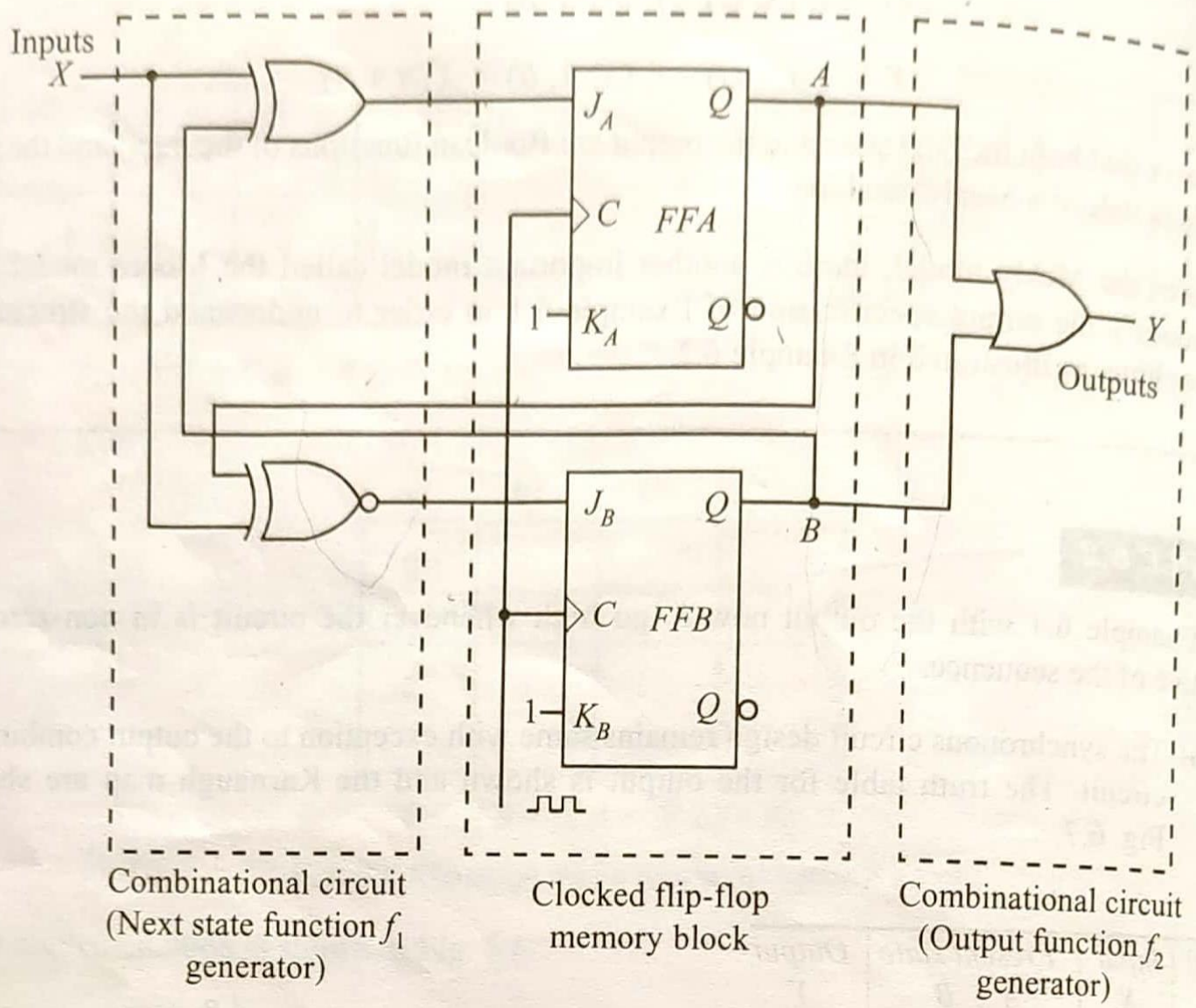


Fig. Implementation

Observe that the next state remains a function of the input and the present state whereas the output now is function only of the present state.

i.e., $Q^+ = f_1(X, A, B)$

and $Y = f_2(A, B)$

In other words

$$Q^+ = f_1(X, Q)$$

and $Y = f_2(Q)$

such a configuration is called a Moore machine.

19EC302: Digital System Design

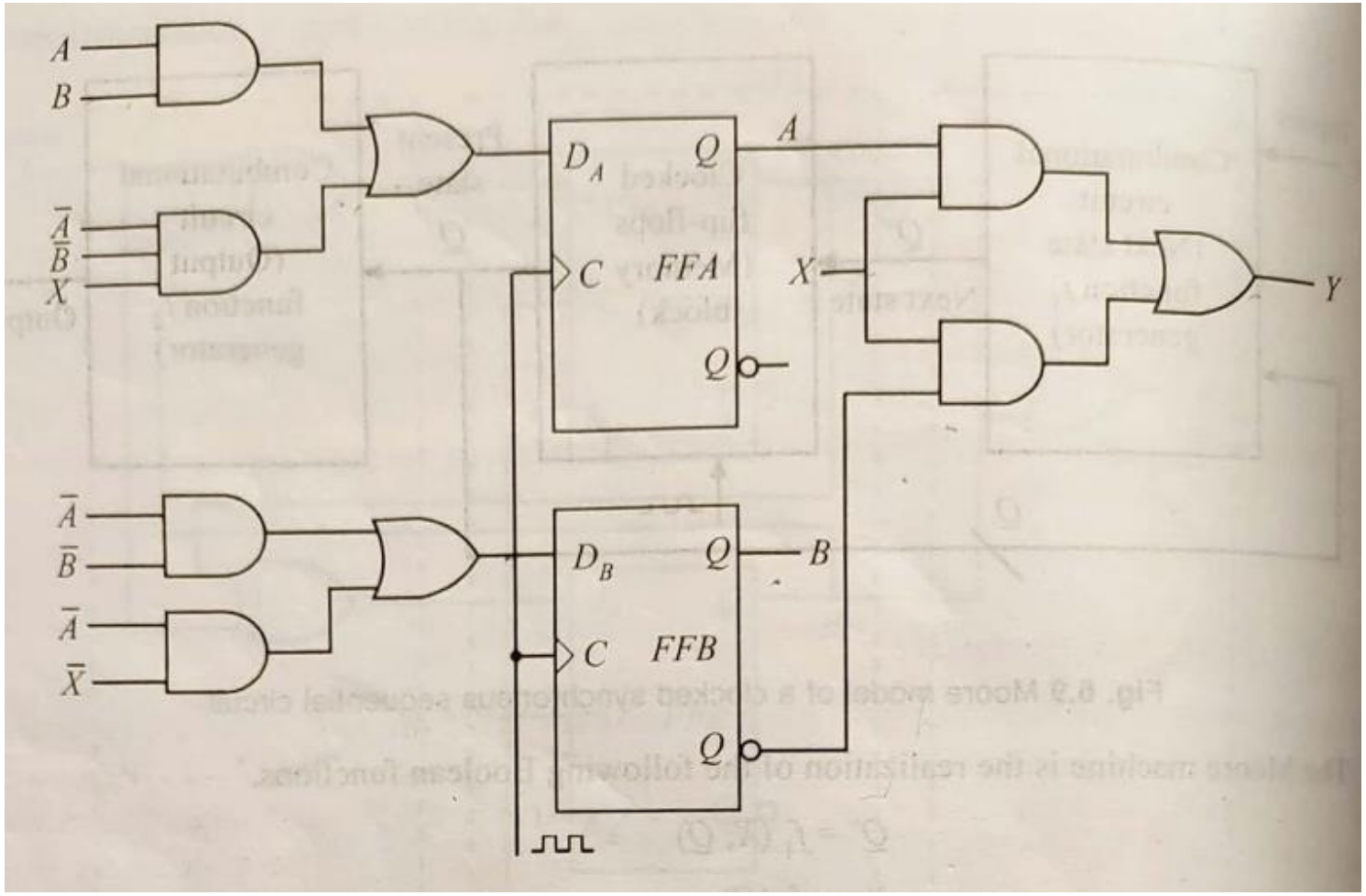
Department of Electronics and Communication Engineering

NMAM INSTITUTE OF TECHNOLOGY

(An Autonomous Institution affiliated to VTU, Belagavi)

Nitte – 574110, Karkala, Udupi District, Karnataka, India

Construct the excitation table, transition table, state table and state diagram for the sequential circuit below



the excitation expressions are

$$D_A = AB + X\,\overline{A}\,\overline{B}$$

$$D_B = \overline{A}\,\overline{B} + \overline{X}\,\overline{A}$$

The output expression is

$$Y = XA + X\overline{B}$$

Truth Table

								B^+		A^+				Y		
Row no.	A	B	X	$\overline{A}$	$\overline{B}$	$\overline{X}$	$\overline{X}\,\overline{A}$	$\overline{A}\,\overline{B}$	$X\overline{A}\,\overline{B}$	AB	A^+	B^+	XA	$X\overline{B}$	Y
0	0	0	0	1	1	1	1	1	0	0	0	1	0	0	0
1	0	0	1	1	1	0	0	1	1	0	1	1	0	1	1
2	0	1	0	1	0	1	1	0	0	0	0	1	0	0	0
3	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0
4	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0
5	1	0	1	0	1	0	0	0	0	0	0	0	1	1	1
6	1	1	0	0	0	1	0	0	0	1	1	0	0	0	0
7	1	1	1	0	0	0	0	0	0	1	1	0	1	0	1

Excitation Table

Present state A B	Excitation $D_A D_B$ Input (X)		Output Y Input (X)	
	$X = 0$	$X = 1$	$X = 0$	$X = 1$
0 0	0 1	1 1	0	1
0 1	0 1	0 0	0	0
1 0	0 0	0 0	0	1
1 1	1 0	1 0	0	1

Transition Table

Present state A B	Next state A^+ B^+ Input (X)		Output Y Input (X)	
	$X = 0$	$X = 1$	$X = 0$	$X = 1$
0 0	0 1	1 1	0	1
0 1	0 1	0 0	0	0
1 0	0 0	0 0	0	1
1 1	1 0	1 0	0	1

Let the states be labeled as follows

$$0\ 0 = a$$

$$0\ 1 = b$$

$$1\ 0 = c$$

$$1\ 1 = d$$

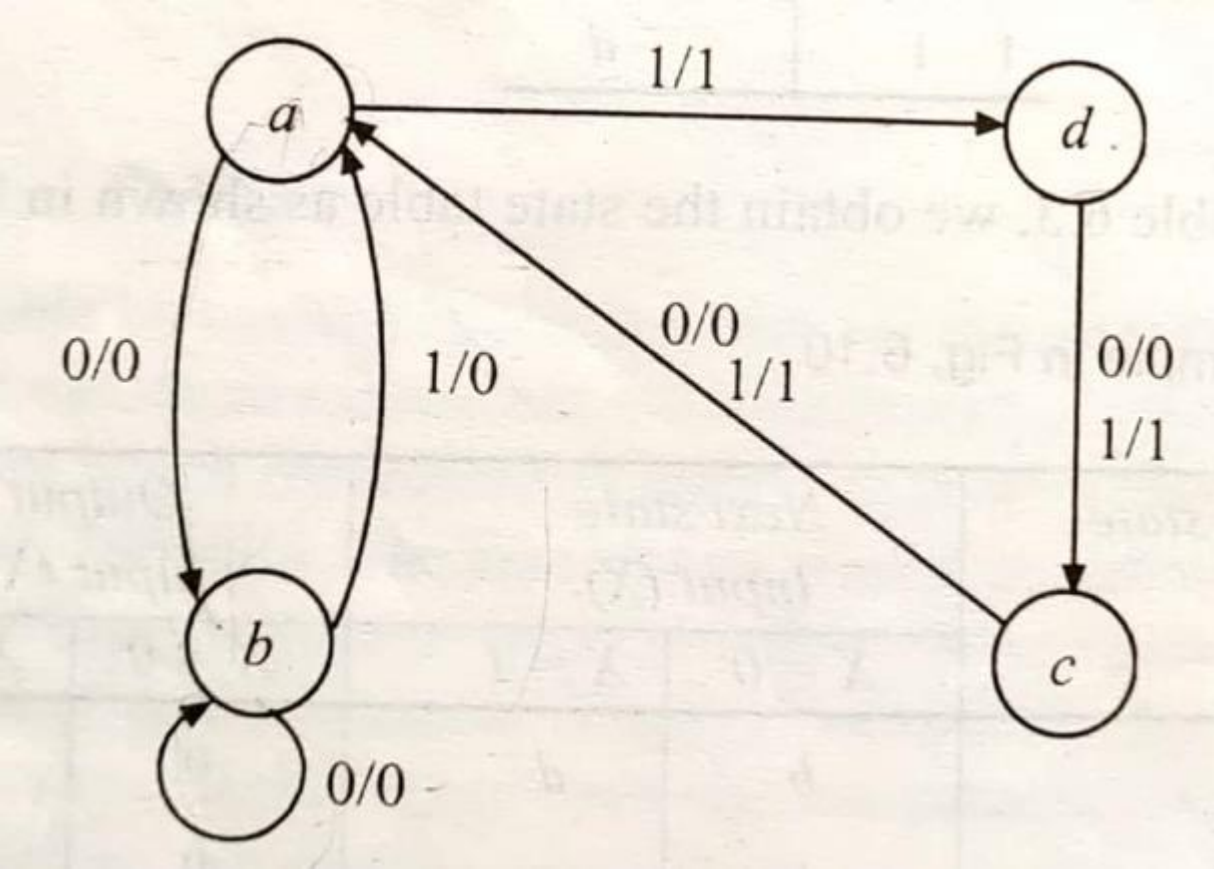
State Table

Present state	*Next state* *Input (X)*		*Output* *Input (X)*	
	X = 0	*X = 1*	*X = 0*	*X = 1*
a	*b*	*d*	0	1
b	*b*	*a*	0	0
c	*a*	*a*	0	1
d	*c*	*c*	0	1

Compact State Table

Present state	Next state, output (Y) Input (X)	
	$X = 0$	$X = 1$
a	b, 0	d, 1
b	b, 0	a, 0
c	a, 0	a, 1
d	c, 0	c, 1

State Diagram



19EC302: Digital System Design

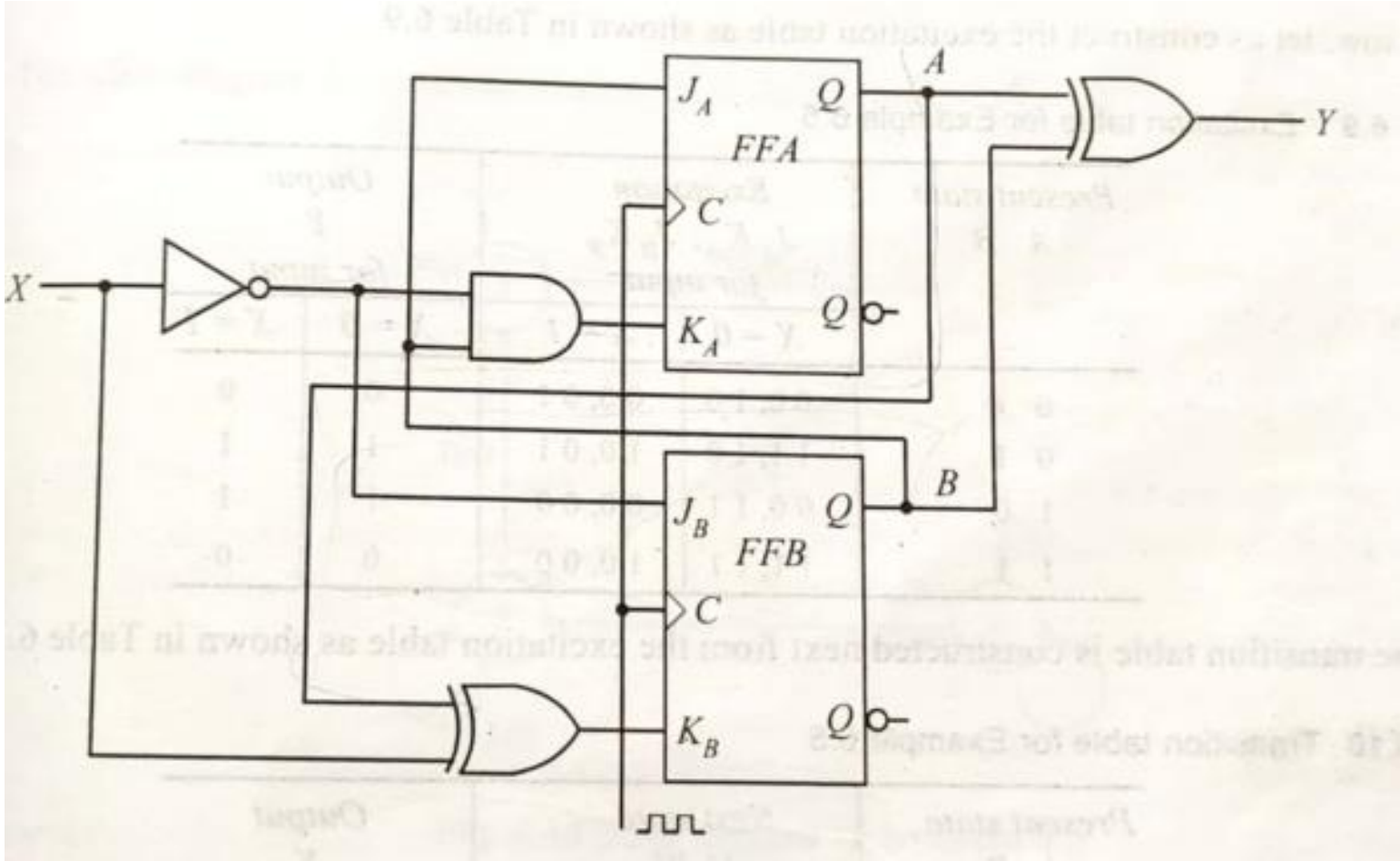
Department of Electronics and Communication Engineering

NMAM INSTITUTE OF TECHNOLOGY

(An Autonomous Institution affiliated to VTU, Belagavi)

Nitte – 574110, Karkala, Udupi District, Karnataka, India

Construct the excitation table, transition table, state table and state diagram for the sequential circuit below



The excitation expression or flip-flop input expressions are

$$J_A = B$$

$$K_A = B\,\overline{X}$$

$$J_B = \overline{X}$$

$$K_B = A \oplus X$$

The output expression is

$$Y = A \oplus B = A\overline{B} + \overline{A}\,B$$

Truth Table

A	B	X	$\overline{A}$	$\overline{B}$	$\overline{X}$	$B\overline{X}$	$A \oplus X$	J_A	K_A	J_B	K_B	$Y = A \oplus B$
0	0	0	1	1	1	0	0	0	0	1	0	0
0	0	1	1	1	0	0	1	0	0	0	1	0
0	1	0	1	0	1	1	0	1	1	1	0	1
0	1	1	1	0	0	0	1	1	0	0	1	1
1	0	0	0	1	1	0	1	0	0	1	1	1
1	0	1	0	1	0	0	0	0	0	0	0	1
1	1	0	0	0	1	1	1	1	1	1	1	0
1	1	1	0	0	0	0	0	1	0	0	0	0

Excitation Table

Present state A B	Excitation $J_A K_A$, $J_B K_B$ for input		Output Y for input
	$X = 0$	$X = 1$	
0 0	0 0, 1 0	0 0, 0 1	0
0 1	1 1, 1 0	1 0, 0 1	1
1 0	0 0, 1 1	0 0, 0 0	1
1 1	1 1, 1 1	1 0, 0 0	0

Transition Table

Present state A B	Next state A^+ B^+ for input $X = 0$	Next state A^+ B^+ for input $X = 1$	Output Y for input
0 0	0 1	0 0	0
0 1	1 1	1 0	1
1 0	1 1	1 0	1
1 1	0 0	1 1	0

Let the states be labeled as follows

$$0\ 0 = a$$

$$0\ 1 = b$$

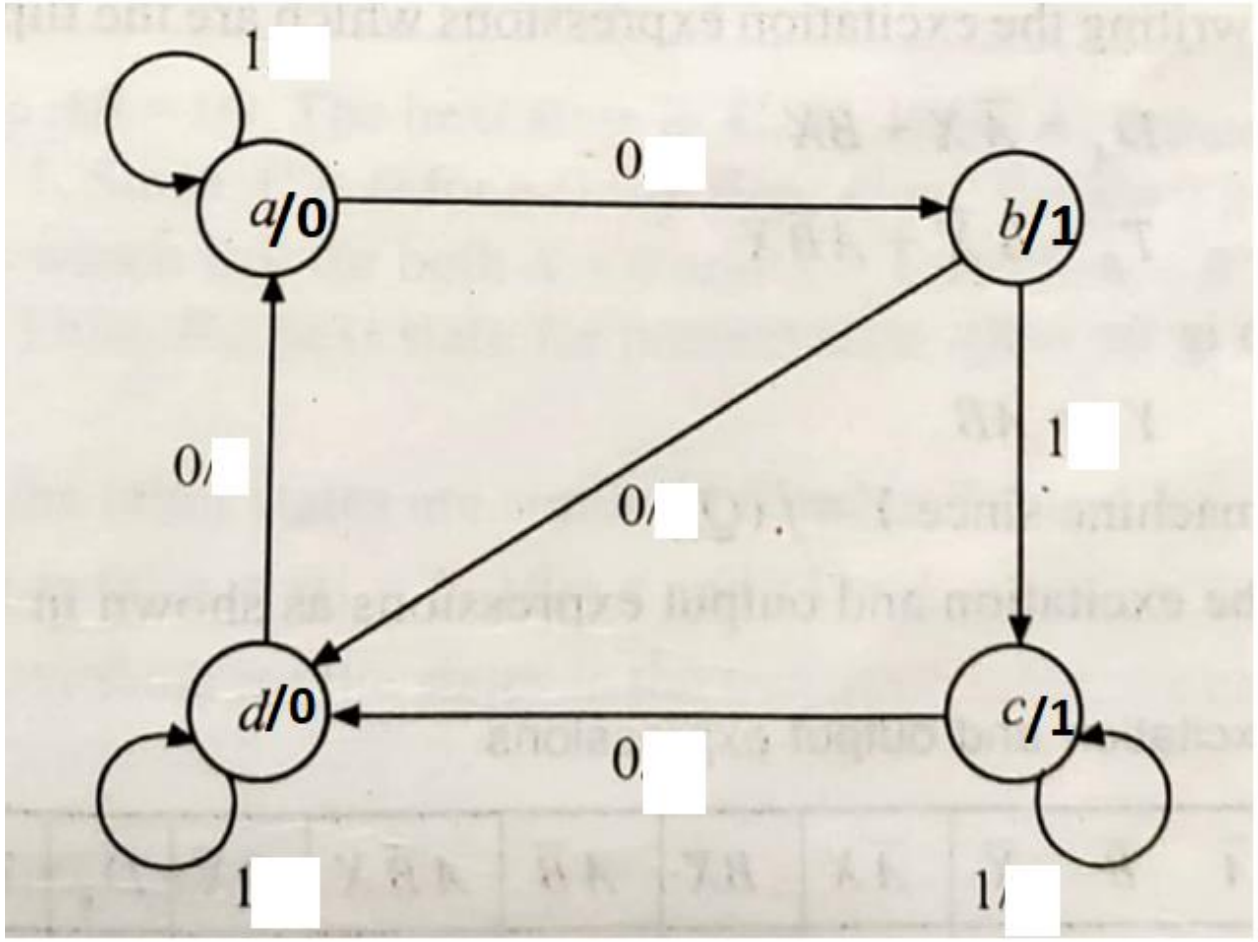
$$1\ 0 = c$$

$$1\ 1 = d$$

State Table

Present state	Next state for input		Output Y for input
	$X = 0$	$X = 1$	
a	b	a	0
b	d	c	1
c	d	c	1
d	a	d	0

State Diagram



19EC302: Digital System Design

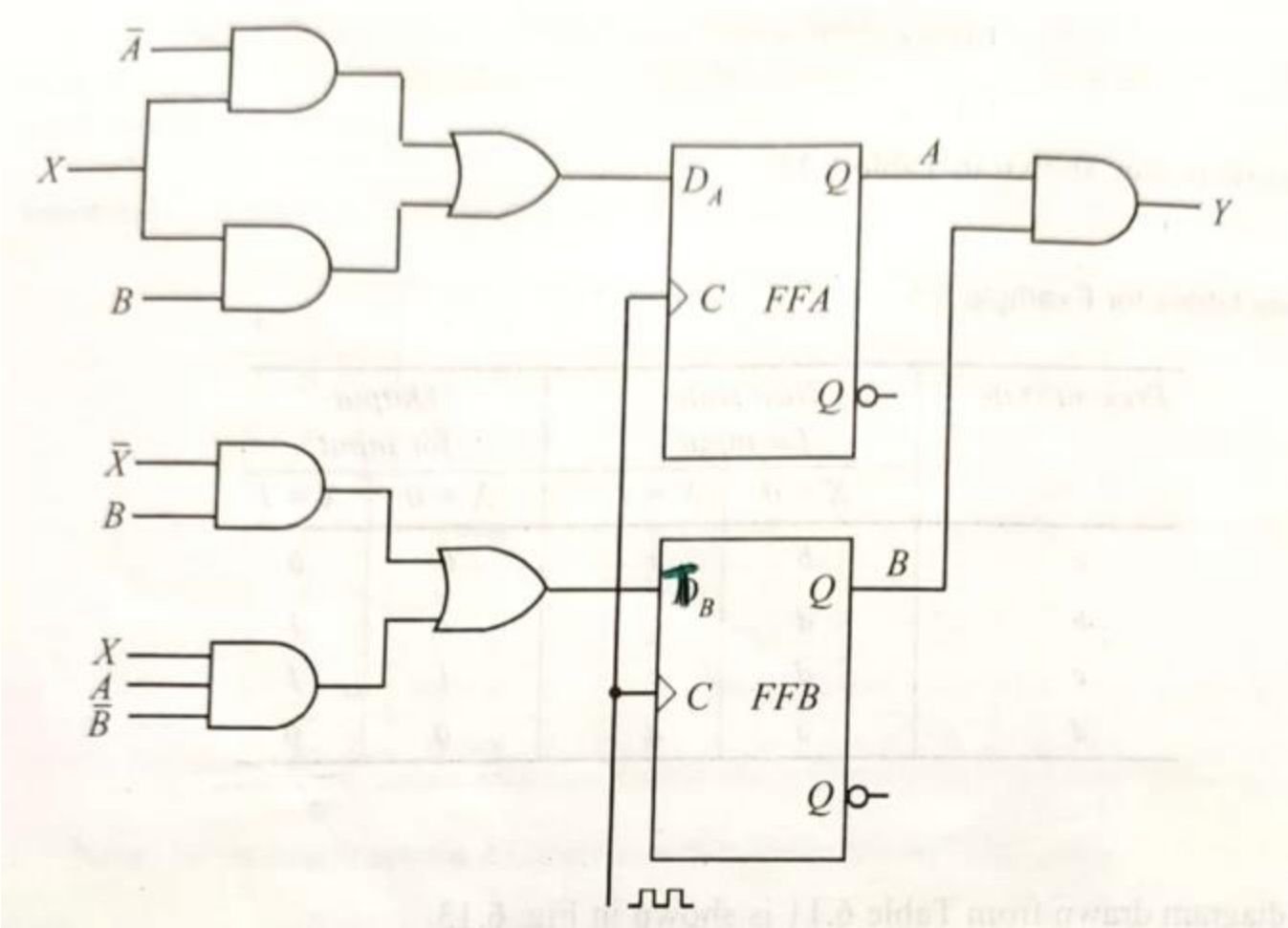
Department of Electronics and Communication Engineering

NMAM INSTITUTE OF TECHNOLOGY

(An Autonomous Institution affiliated to VTU, Belagavi)

Nitte – 574110, Karkala, Udupi District, Karnataka, India

Construct the excitation table, transition table, state table and state diagram for the sequential circuit below



$$D_A = \overline{A}X + BX$$

$$T_B = B\overline{X} + A\overline{B}X$$

The output expression is

$$Y = AB$$

This is also a Moore machine since $Y = f(Q)$

Truth Table

A	B	X	$\overline{A}$	$\overline{B}$	$\overline{X}$	$\overline{A}X$	BX	$A\overline{B}$	$A\overline{B}X$	$B\overline{X}$	D_A	T_B	$Y = AB$
0	0	0	1	1	1	0	0	0	0	0	0	0	0
0	0	1	1	1	0	1	0	0	0	0	1	0	0
0	1	0	1	0	1	0	0	0	0	1	0	1	0
0	1	1	1	0	0	1	1	0	0	0	1	0	0
1	0	0	0	1	1	0	0	1	0	0	0	0	0
1	0	1	0	1	0	0	0	1	1	0	0	1	0
1	1	0	0	0	1	0	0	0	0	1	0	1	1
1	1	1	0	0	0	0	1	0	0	0	1	0	1

Excitation Table

Present state A B	Excitation D_A, T_B for input		Output Y for input X
	$X = 0$	$X = 1$	
0 0	0, 0	1, 0	0
0 1	0, 1	1, 0	0
1 0	0, 0	0, 1	0
1 1	0, 1	1, 0	1

Transition Table

Present state A B	Next state A^+ B^+ for input $X = 0$	Next state A^+ B^+ for input $X = 1$	Output Y for input X
0 0	0 0	1 0	0
0 1	0 0	1 1	0
1 0	0 0	0 1	0
1 1	0 0	1 1	1

Let the states be labeled as follows

$$0\ 0 = a$$

$$0\ 1 = b$$

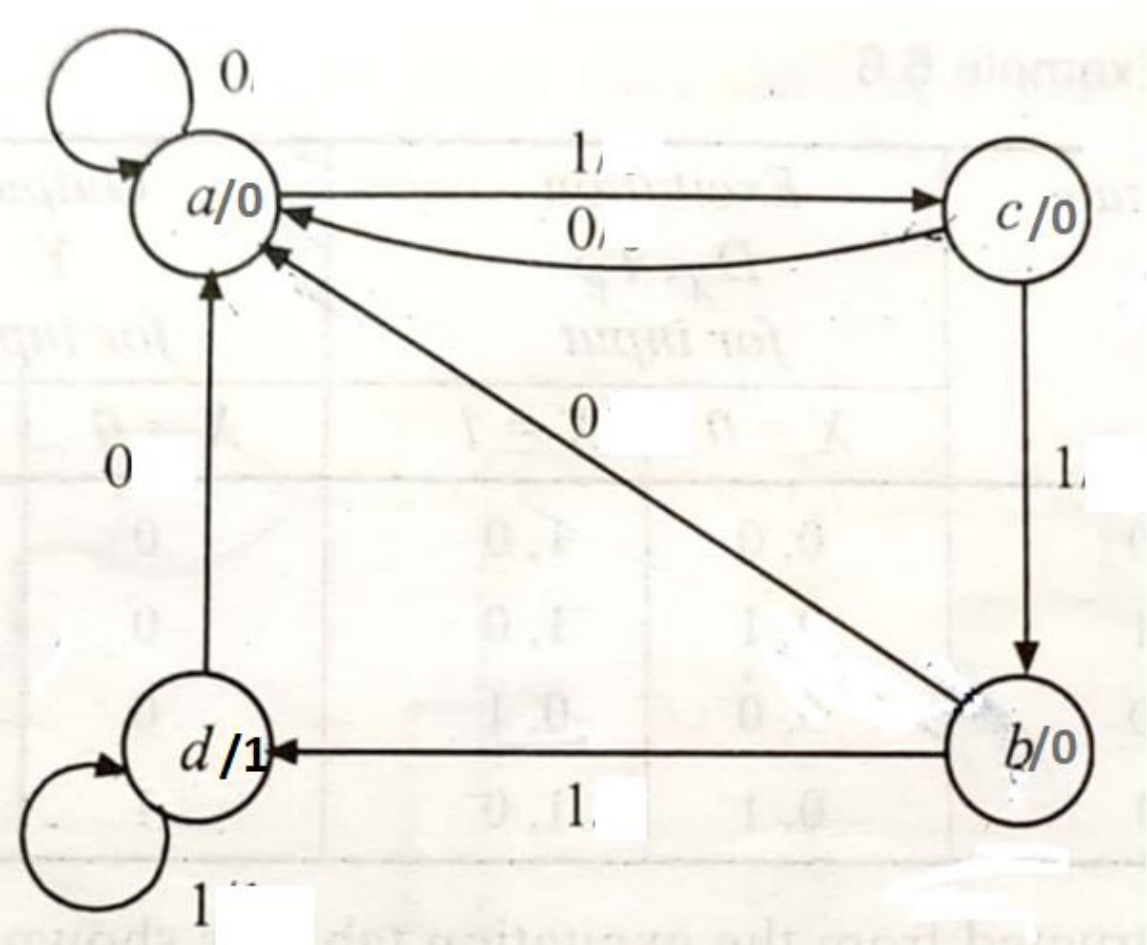
$$1\ 0 = c$$

$$1\ 1 = d$$

State Table

Present state	Next state for input		Output Y for input
	$X = 0$	$X = 1$	X
a	a	c	0
b	a	d	0
c	a	b	0
d	a	d	1

State Diagram



19EC302: Digital System Design

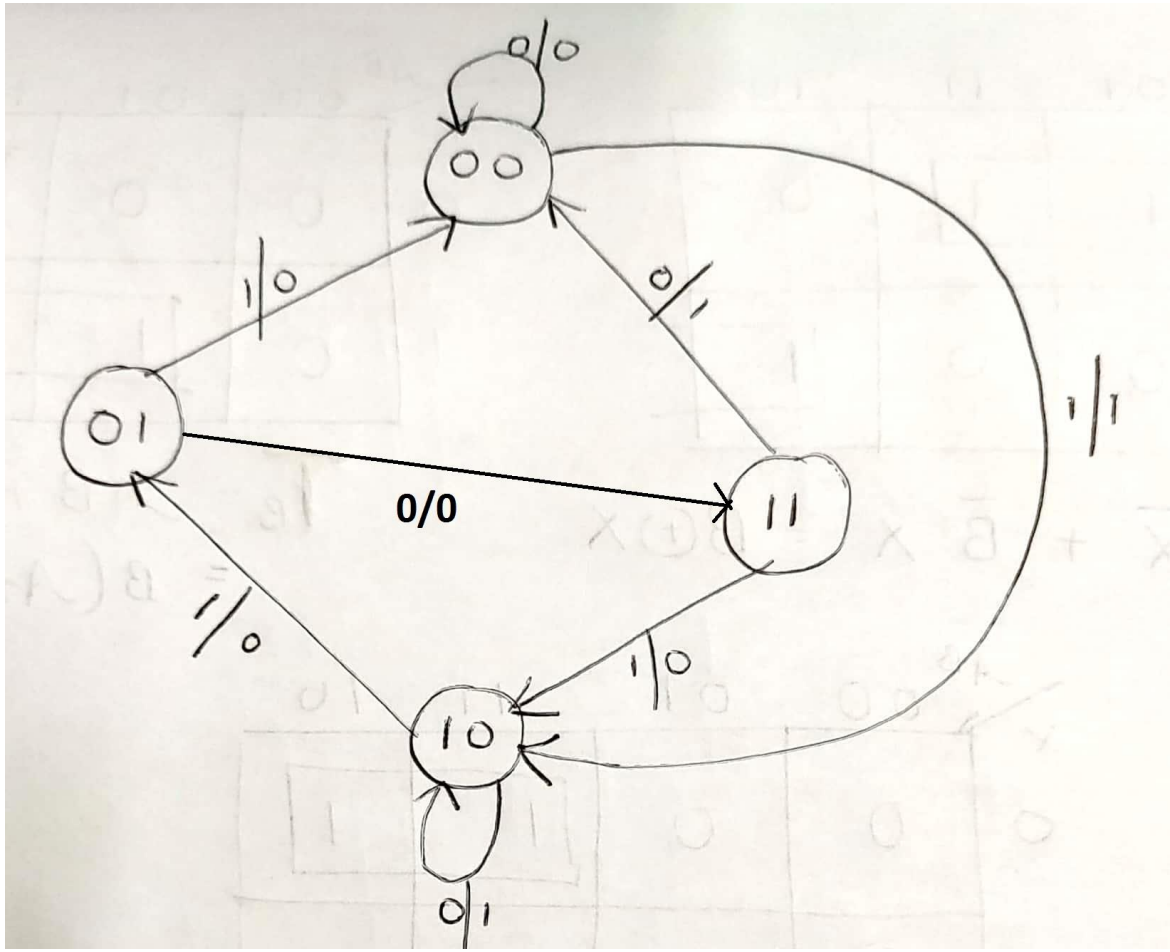
Department of Electronics and Communication Engineering

NMAM INSTITUTE OF TECHNOLOGY

(An Autonomous Institution affiliated to VTU, Belagavi)

Nitte – 574110, Karkala, Udupi District, Karnataka, India

Construct a sequential circuit from the state diagram using Clocked T Flip Flops



Transition Table

Present State		Next State $A^+ B^+$		output Y	
A	B	X = 0	X = 1	X = 0	X = 1
0	0	0 0	1 0	0	1
0	1	1 1	0 0	0	0
1	0	1 0	0 1	1	0
1	1	0 0	1 0	1	0

Excitation Table.

Present state			Next state		F/F I/Ps		output Y
X	A	B	A^+	B^+	T_A	T_B	
0	0	0	0	0	0	0	0
0	0	1	1	1	1	0	0
0	1	0	1	0	0	0	1
0	1	1	0	0	1	1	1
1	0	0	1	0	1	0	1
1	0	1	0	0	0	1	0
1	1	0	0	1	1	1	0
1	1	1	1	0	0	1	0

X \ AB	00	01	11	10
0	0	1	1	0
1	1	0	0	1

$$T_A = B\bar{X} + \bar{B}X = B \oplus X$$

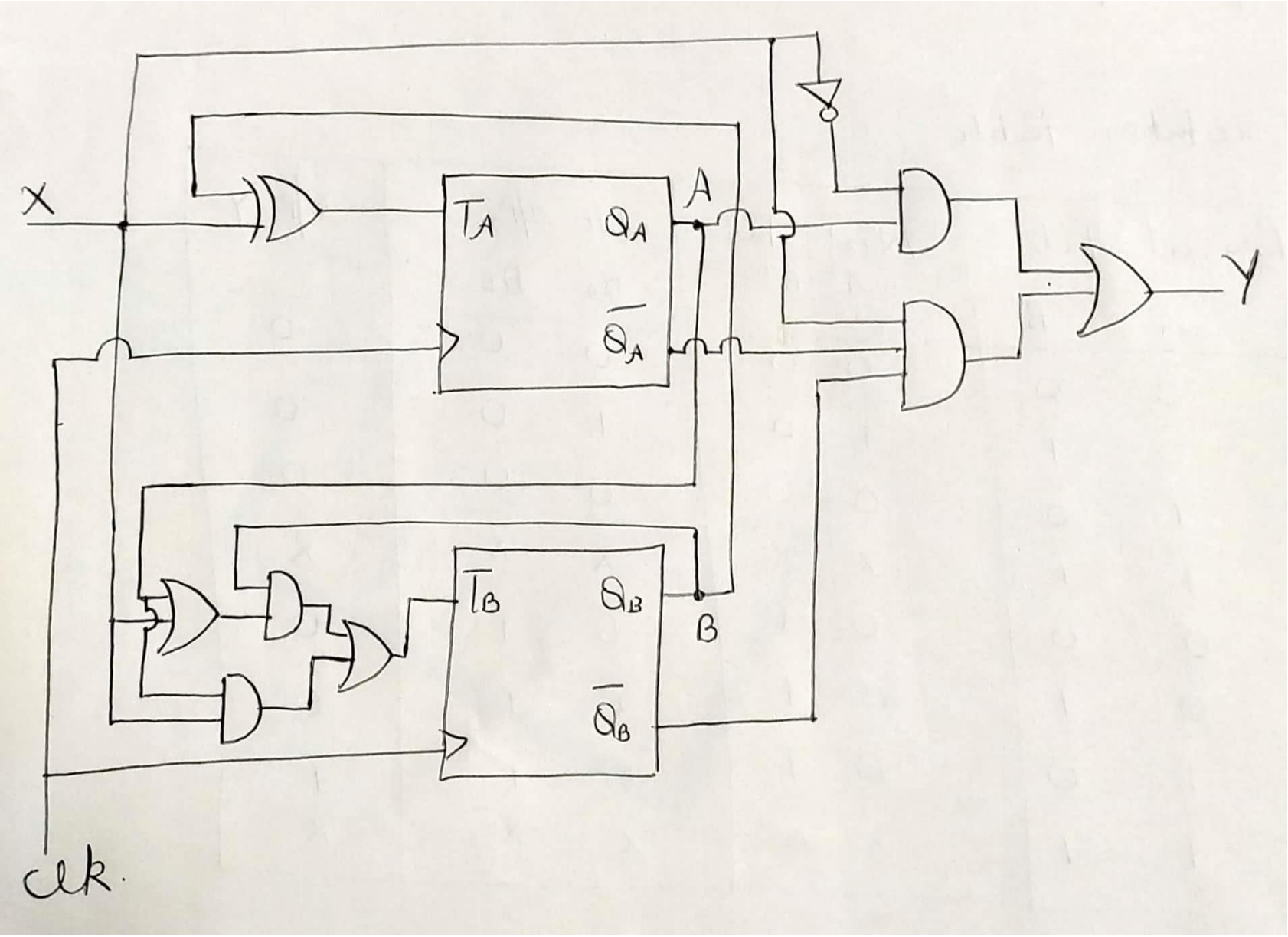
X \ AB	00	01	11	10
0	0	0	1	0
1	0	1	1	1

$$T_B = AB + BX + AX$$
$$= B(A+X) + AX$$

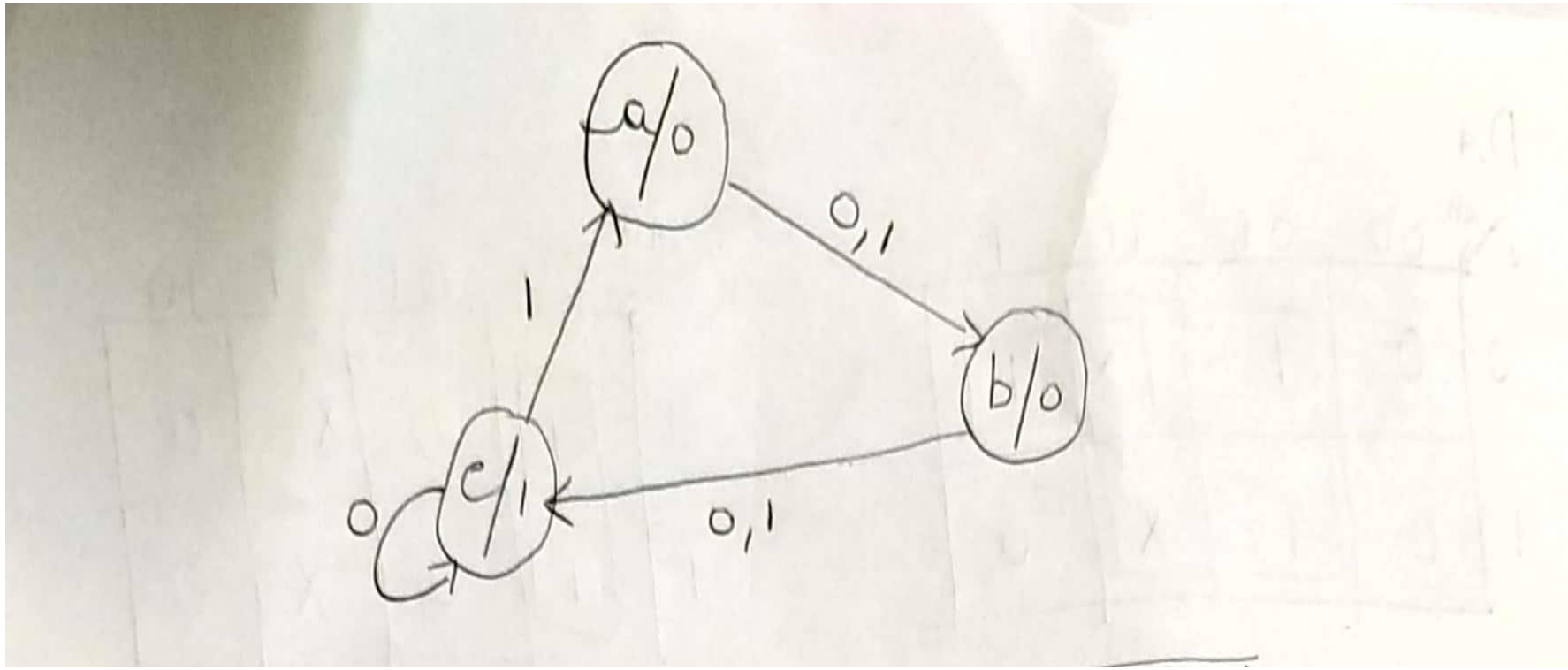
X \ AB	00	01	11	10
0	0	0	1	1
1	1	0	0	0

$$Y = A\bar{X} + \bar{A}\bar{B}X$$

Sequential circuit



Construct a sequential circuit from the state diagram using Clocked D Flip Flops



Transition Table

Present State		Next State $A^+ B^+$		output
A	B	X=0	X=1	Y
0	0 (a)	0 1 (b)	0 1 (b)	0
0	1 (b)	1 0 (c)	1 0 (c)	0
1	0 (c)	1 0 (c)	0 0 (a)	1

Excitation table.

X	A	B	Next State A^+	B^+	F/F i/ps D_A	D_B	o/p Y
0	0	0	0	1	0	1	0
0	0	1	1	0	1	0	0
0	1	0	1	0	1	0	1
0	1	1	X	X	X	X	X
1	0	0	0	1	0	1	0
1	0	1	1	0	1	0	0
1	1	0	0	0	0	0	1
1	1	1	X	X	X	X	X

D_A

X \ AB	00	01	11	10
0	0	1	X	1
1	0	1	X	0

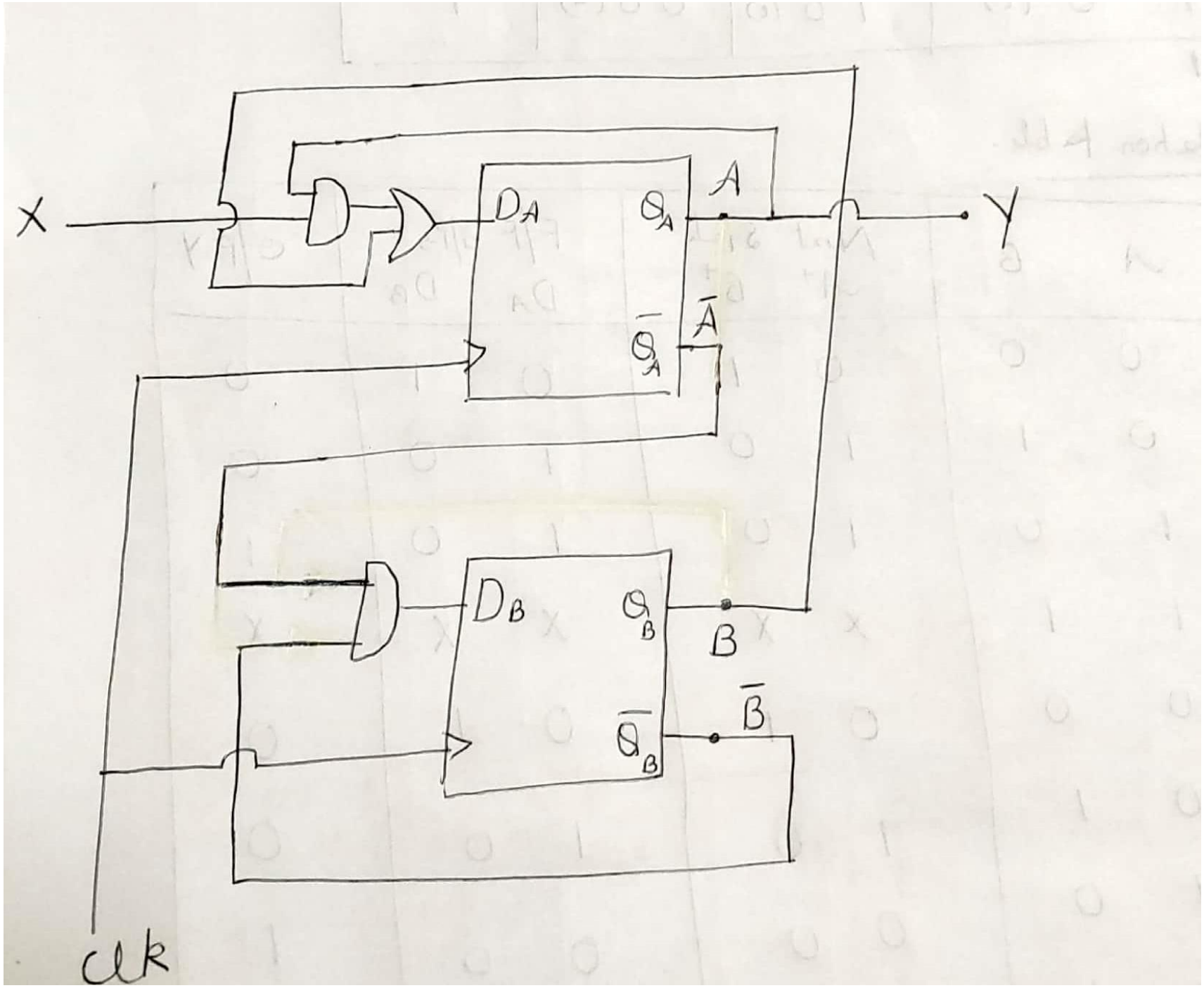
$D_A = B + AX$

X \ AB	00	01	11	10
0	1	0	X	0
1	1	0	X	0

$D_B = \bar{A}\,\bar{B}$

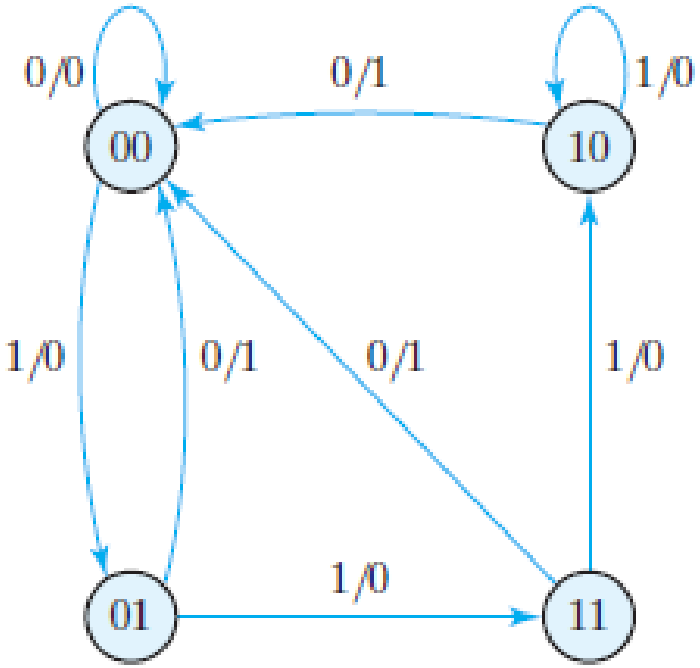
X \ AB	00	01	11	10
0	0	0	X	1
1	0	0	X	1

$Y = A$


X
Y
D_A
Q_A
A
$\overline{Q_A}$
$\bar{A}$
D_B
Q_B
B
$\overline{Q_B}$
$\bar{B}$
clk

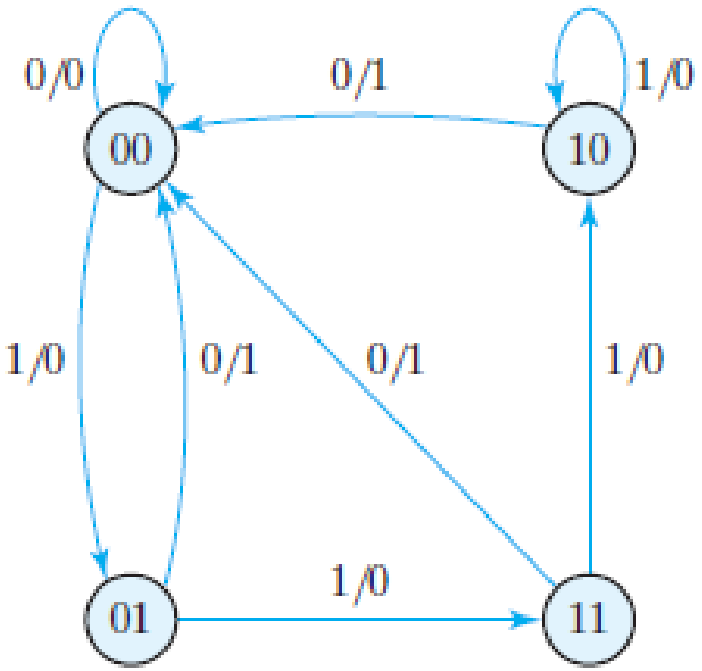
STATE DIAGRAM BASED HDL MODELS

Write a Verilog behavioral code for the Mealy Machine shown using state diagram



HDL Example

```
// Mealy FSM zero detector
module Mealy_Zero_Detector (
  output reg y_out,
  input   x_in, clock, reset
);
  reg [1: 0]          state, next_state;
  parameter           S0 = 2'b00, S1 = 2'b01, S2 = 2'b10, S3 = 2'b11;
  always @ (posedge clock, negedge reset)
    if (reset == 0) state <= S0;
    else state <= next_state;

  always @ (state, x_in)                          // Form the next state
    case (state)
      S0:        if (x_in)  next_state = S1; else next_state = S0;
      S1:        if (x_in)  next_state = S3; else next_state = S0;
      S2:        if (~x_in)  next_state = S0; else next_state = S2;
      S3:        if (x_in)  next_state = S2; else next_state = S0;
    endcase

  always @ (state, x_in)                          // Form the Mealy output
    case (state)
      S0:          y_out = 0;
      S1, S2, S3: y_out = ~x_in;
    endcase
endmodule
```

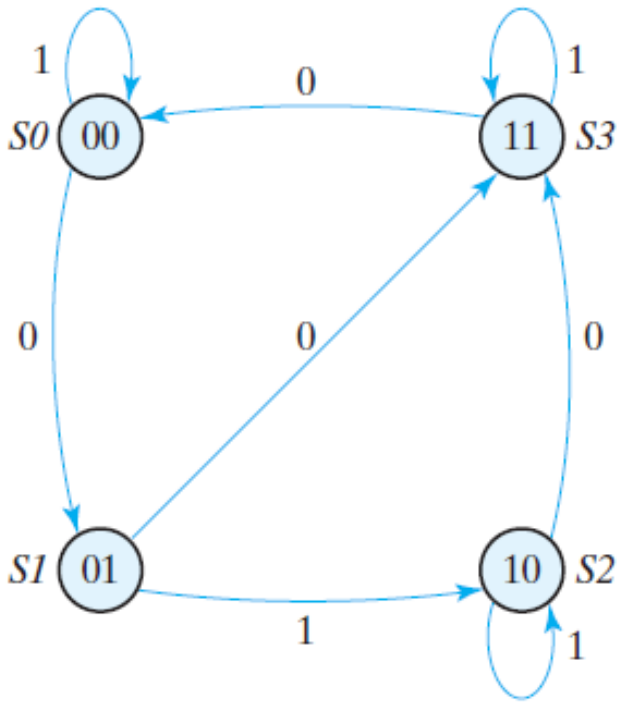


Consider the sequential circuit with two JK flip-flops A and B and one input x. The circuit has no outputs. The circuit can be specified by the flip-flop input equations. Consider the output to be the state value.

JA = B KA = Bx’

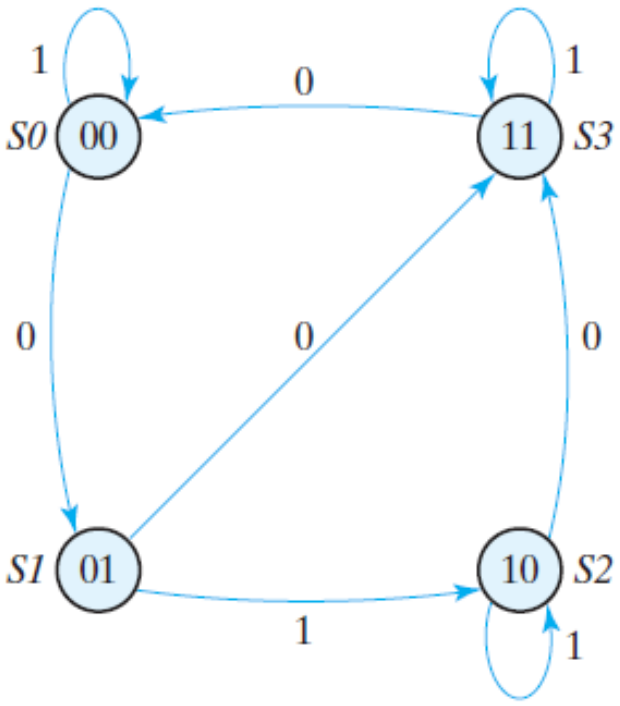
JB = x’ KB = A’x + Ax` = A $\oplus$x

Write a Verilog code for the synchronous sequential circuit represented by the state diagram shown below.

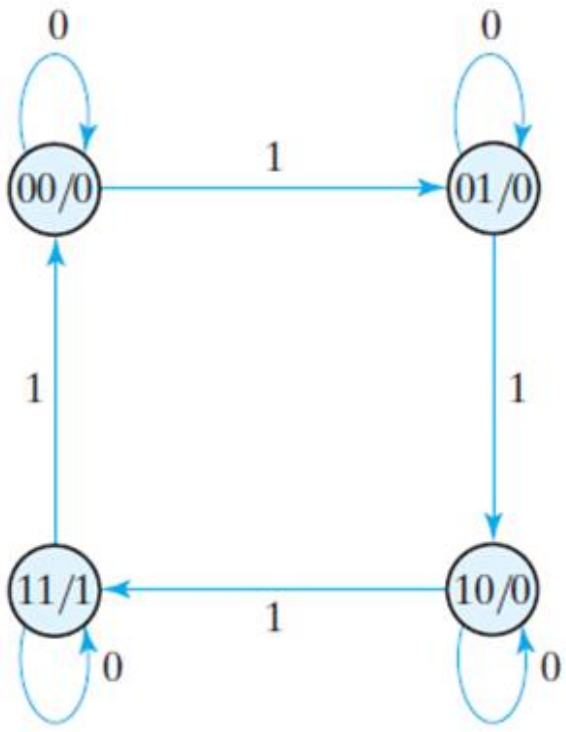


HDL Example

```
// Moore model FSM
module Moore_Model_    (
  output [1: 0]          y_out,
  input                  x_in, clock, reset
);
  reg [1: 0]             state;
  parameter              S0 = 2'b00, S1 = 2'b01, S2 = 2'b10, S3 = 2'b11;

  always @ (posedge clock, negedge reset)
    begin

  if (reset == 0) state <= S0;                    // Initialize to state S0
  else case (state)
    S0:      if (~x_in) state <= S1; else state <= S0;
    S1:      if (x_in)  state <= S2; else state <= S3;
    S2:      if (~x_in) state <= S3; else state <= S2;
    S3:      if (~x_in) state <= S0; else state <= S3;
  endcase

          y_out = state;        // Output of flip-flops
      end
 endmodule
```



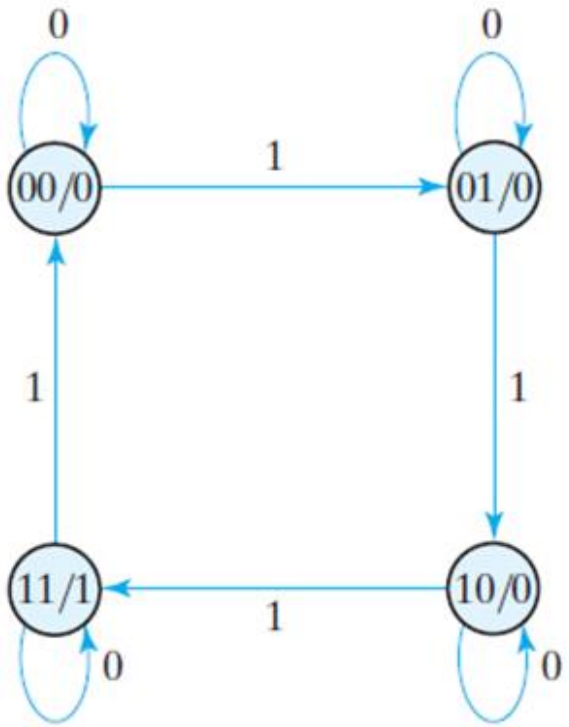
Ex: 2-bit Binary counter. The sequential circuit depicted by the state diagram is a two-bit binary counter controlled by input x_in . The output, y_out, is enabled when the count reaches binary 11. Write a Verilog behavioral code for the sequential circuit.



(b) State diagram

HDL Example (Binary Counter_Moore Model)

```
// State-diagram-based model
module Moore_Model_Fig (
  output  y_out,
  input   x_in, clock, reset
);
  reg [1: 0]          state;
  parameter           S0 = 2'b00, S1 = 2'b01, S2 = 2'b10, S3 = 2'b11;
  always @ (posedge clock, negedge reset)
    begin
   if (reset == 0) state <= S0;              // Initialize to state S0
   else case (state)
    S0:     if (x_in)  state <= S1; else state <= S0;
    S1:     if (x_in)  state <= S2; else state <= S1;
    S2:     if (x_in)  state <= S3; else state <= S2;
    S3:     if (x_in)  state <= S0; else state <= S3;
   endcase
          y_out = (state == S3);          // Output of flip-flops
    end
  endmodule
```



(b) State diagram

19EC302: Digital System Design

Department of Electronics and Communication Engineering

NMAM INSTITUTE OF TECHNOLOGY

(An Autonomous Institution affiliated to VTU, Belagavi)

Nitte – 574110, Karkala, Udupi District, Karnataka, India

Draw the state diagram of a Mealy machine whose output is 1 iff the input has been 1 for three consecutive clock cycles, but non-overlapping.

Solution: Let the input be x and the output be z. A sample sequence for the problem is shown in Fig.

x	0	0	1	1	1	1	0	0	1	1	0	0	1	1	1	1	1	1	0	0
z	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0

Step 1: Start at state A. if $x = 0$, remain at A, else go to B, implying occurence of one 1 as shown in Fig. 8.8 (a).

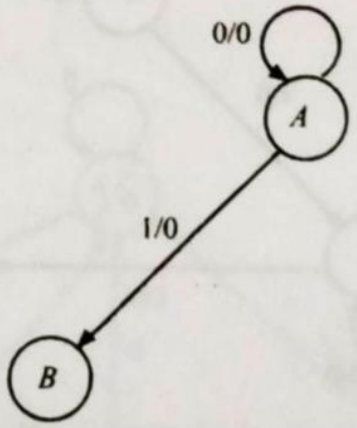


Fig. 8.8 (a) State diagram after step 1

Step 2: At state B, if $x = 0$, go back to A, else go to state C, implying the occurrence of two 1s, as shown in Fig. 8.8 (b).

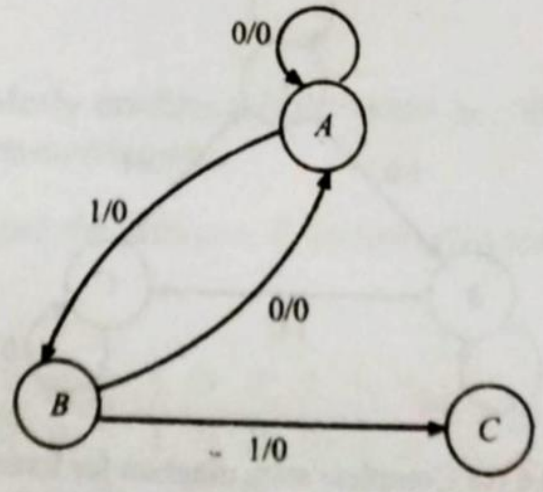
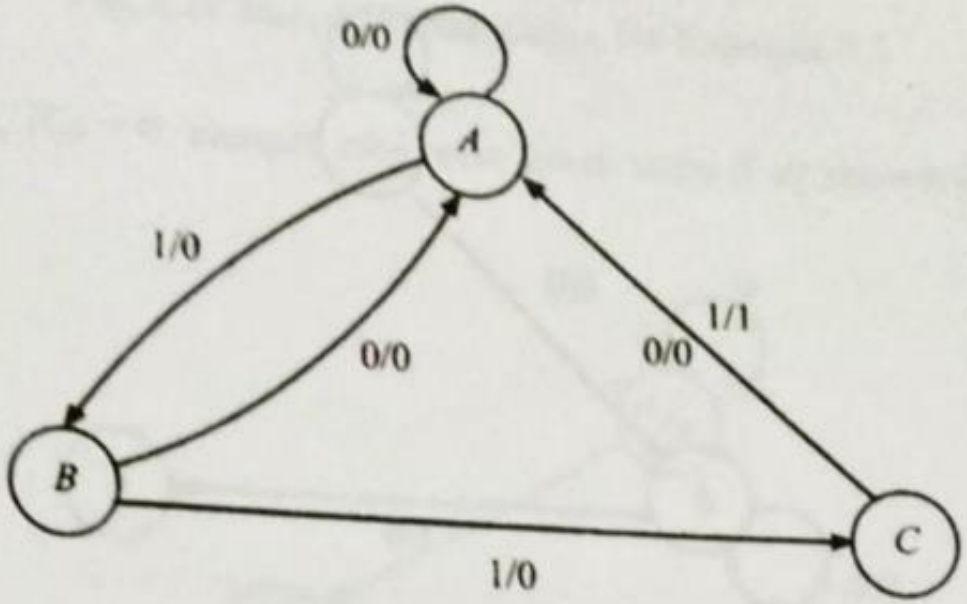


Fig. 8.8 (b) State diagram after step 2

Step 3: At state C, if $x = 0$, go back to state A with zero output else go back to state A with 1 output indicating the occurrence of three consecutive 1s as shown in Fig.



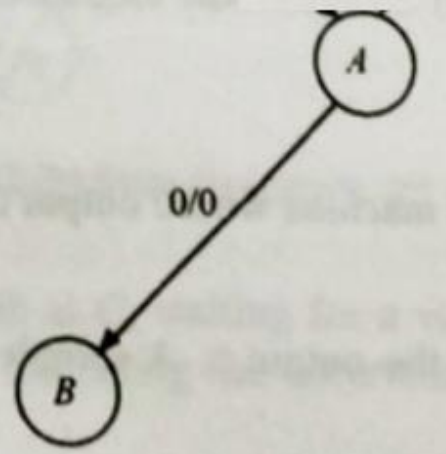
Draw the state diagram of a Mealy machine whose output is 1 iff the last three inputs were 010 assuming that the sequence could overlap.

Solution: Assuming input to be x and output to be z, a sample input/output sequence is shown in Fig.

x	1	1	0	1	0	1	0	1	0	0	1	0	0	1	0	1	0	1	0	0
z	0	0	0	0	1	0	1	0	1	0	0	1	0	0	1	0	1	0	1	0

Fig . Sample I/O sequence

Step 1: Start at state A. If x is 0, go to state B indicating the possible start of a valid sequence, els remain at A as shown in Fig. (a).



Step 2: At state B, if $x = 0$, remain at B, else go to state C indicating the occurrence of the second valid bit in the sequence to be detected as shown in Fig. (b).

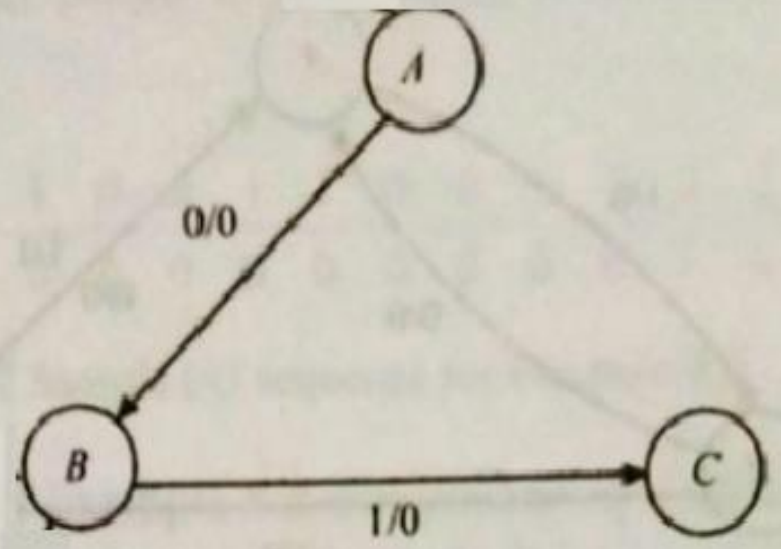


Fig. (b) State diagram after step 2

Step 3: At state C, if $x = 0$, return to state A with output 1, indicating the detection of the complete sequence 010, else return to state A with output 0 as shown in Fig (c).

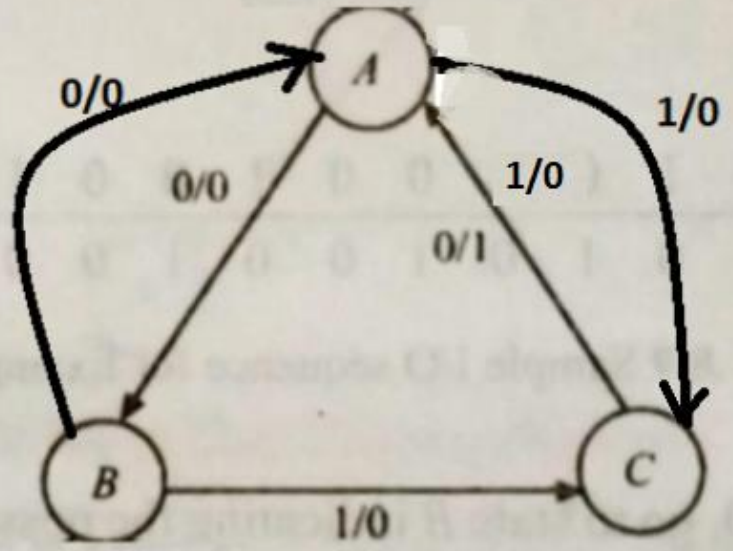


Fig. (c) Complete state diagram for

19EC302: Digital System Design

Department of Electronics and Communication Engineering

NMAM INSTITUTE OF TECHNOLOGY

(An Autonomous Institution affiliated to VTU, Belagavi)

Nitte – 574110, Karkala, Udupi District, Karnataka, India

Draw the state diagram of a Moore machine to output a 1 if the input has been 1 for three consecutive clock cycles.assuming that the sequence could overlap

Solution: Let us first write an example which is described by the problem. Let the input be x and the output be z.

A sample input/output sequence is shown in Fig.

x	0	1	1	1	1	1	1	1	0	1	1	0	1	1	1	0	0
z	0	0	0	1	1	1	1	1	0	0	0	0	0	0	1	0	0

Fig Sample 1/0 sequence for Example

Step 1: Start at state A. If $x = 0$, remain at A implying zero 1s and if $x = 1$, go to state B, implying the occurrence of one 1, as shown in Fig.

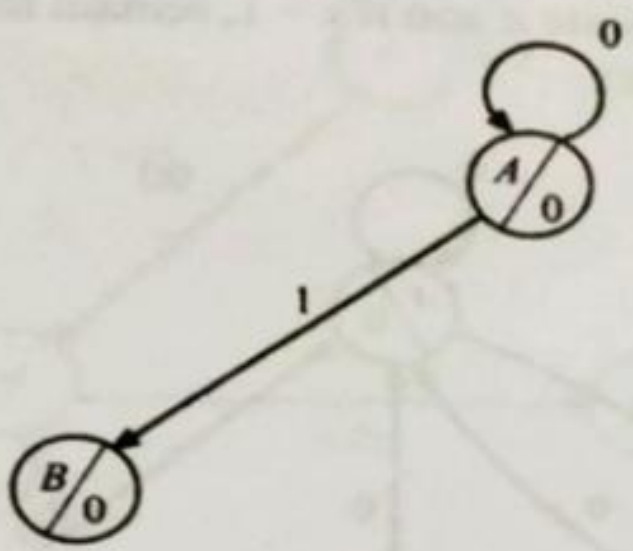


Fig. State design after step 1

Step 2: At B, if $x = 0$, go back to state A and if $x = 1$, go to state C implying occurrence of two 1s as shown in Fig. (a).

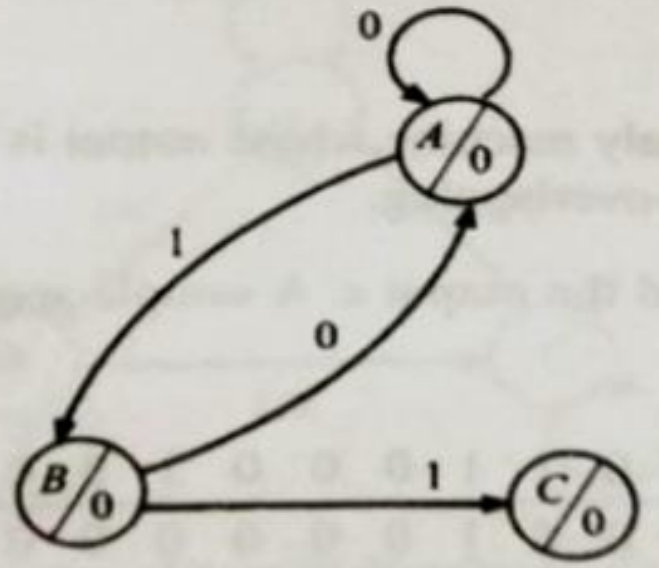


Fig. (a) State diagram after step 2

Step 3: At C, if $x = 0$, go back to A and if $x = 1$, go to state D implying the occurrence of three 1s at which point a 1 is output, since 1s have been encountered in three consecutive clocks as shown in Fig. (b).

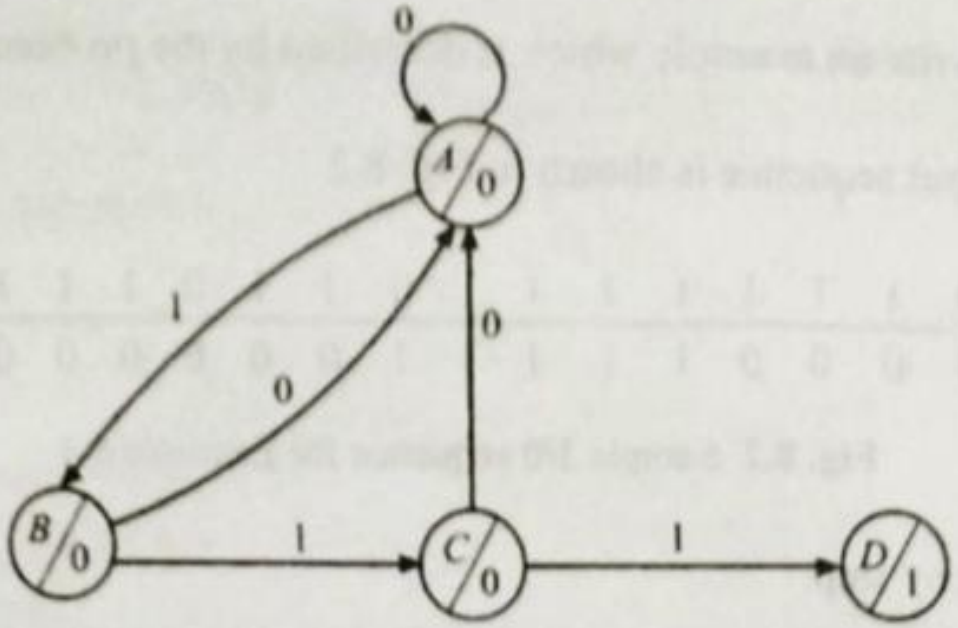


Fig. (b) State diagram after step 3

Step 4: At D, if $x = 0$, go back to state A and if $x = 1$, remain in state D implying more consecutive 1s as shown in Fig. (c).

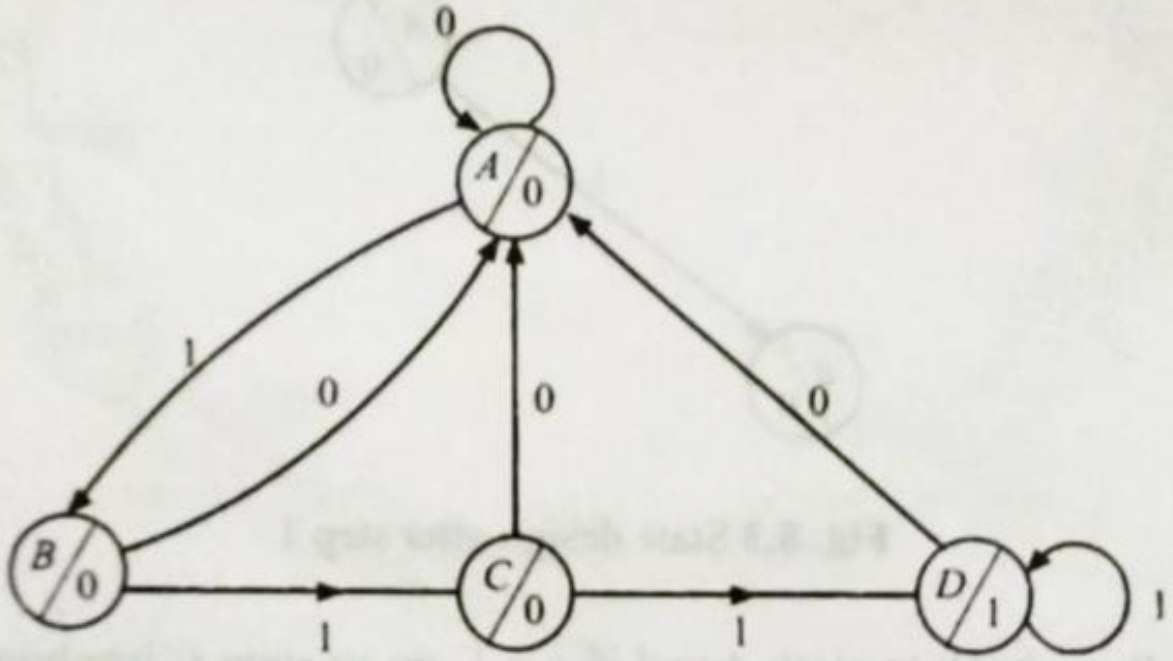


Fig. (c) Complete state diagram for

19EC302: Digital System Design

Department of Electronics and Communication Engineering

NMAM INSTITUTE OF TECHNOLOGY

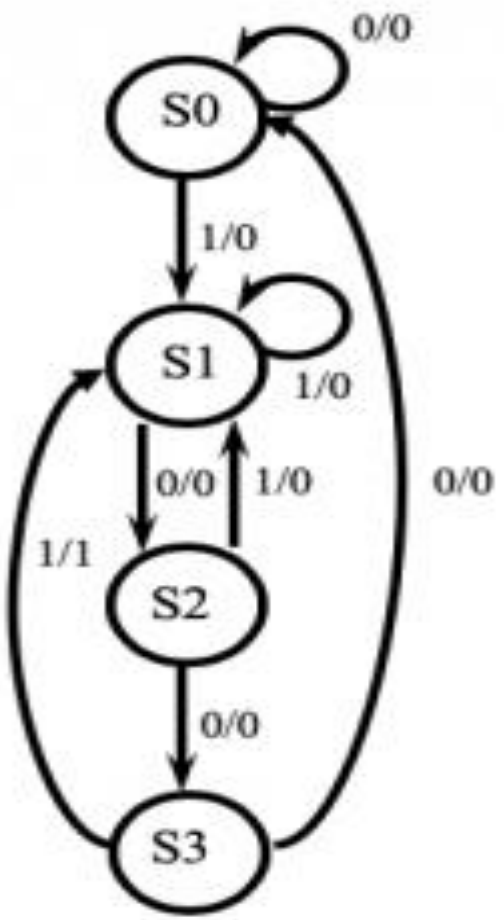
(An Autonomous Institution affiliated to VTU, Belagavi)

Nitte – 574110, Karkala, Udupi District, Karnataka, India

Design and write verilog code for sequence detector 1001 using Mealy model for overlapping

```
module sd1001_mealy_over(input clk, reset, din, output reg dout);
reg[1:0]  state;
parameter S0=2'b00,S1=2'b01,S2=2'b10,S3=2'b11;
always @(posedge clk or posedge reset) begin
    if(reset)
    begin  dout <= 1'b0; state <= S0; end
    else
    begin
      case(state)
        S0: begin
          if(din) begin
            state <= S1;
            dout <=1'b0;
          end
          else
            dout <=1'b0;
        end
      S1: begin
          if(~din) begin
            state <= S2;
            dout <=1'b0;
          end
        else begin
            dout <=1'b0;
          end
        end
        S2: begin
            if(~din) begin
              state <= S3;
              dout <=1'b0;
            end
            else begin
              state <= S1;
              dout <=1'b0;
            end
          end
          S3: begin
            if(din) begin
              state <= S1;
              dout <=1'b1;
            end
            else begin
              state <= S0;
              dout <=1'b0;
            end
          end
        endcase
      end
    end
endmodule
```

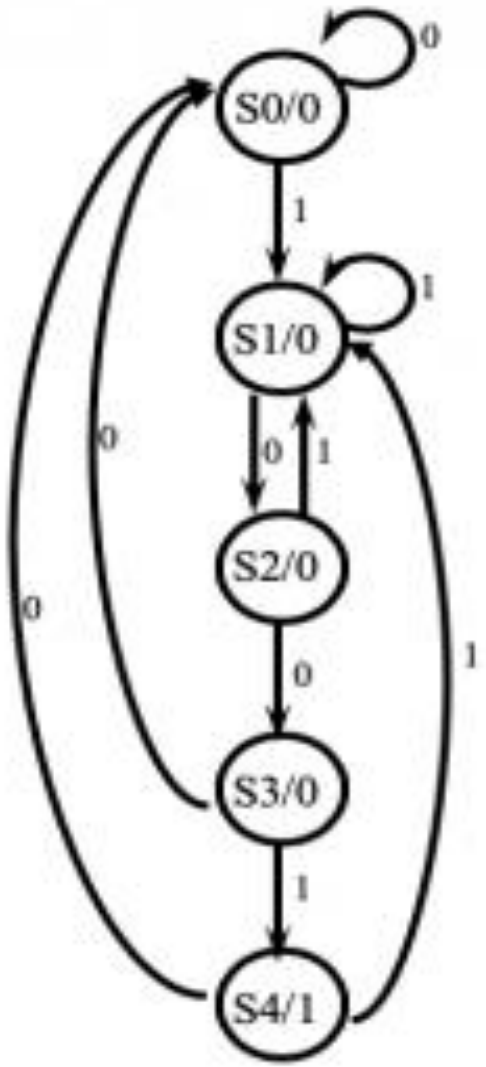
Mealy Machine (Overlapping)



Design and write verilog code for sequence detector 1001 using Moore model for non overlapping

```
module sd1001_moore(input clk, reset, din, output dout);
reg[2:0]  state;
parameter S0=3’b000,S1=3’b001,S2=3’b010,S3=3’b011,
S4=3’b100;

 always @(posedge clk or posedge reset)
 begin
    if(reset)
    begin
      dout <= 1'b0;
      state <= S0;
    end
    else
     begin
      case(state)
        S0: begin
          dout <=1'b0;
          if(din)
            state <= S1;
        end
        S1: begin
          dout <= 1'b0;
          if(~din)
            state <= S2;
        end
        S2: begin
          dout <= 1'b0;
          if(~din)
            state <= S3;
          else
            state <= S1;
        end
        S3: begin
          dout <= 1'b0;
          if(din)
            state <= S4;
          else
            state <= S0;
        end
        S4: begin
          dout <= 1'b1;
          if(din)
            state <= S1;
          else
            state <= S0;
        end
      endcase
    end
  end
endmodule
```

Moore Machine (Non-Overlapping)



Extra

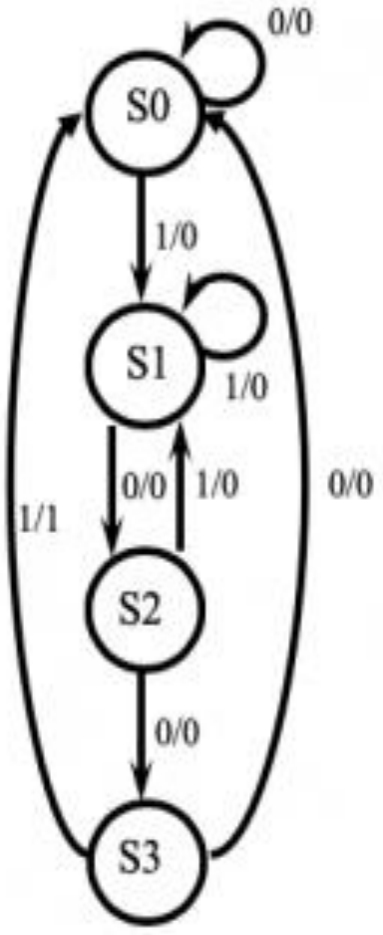
Design and write verilog code for sequence detector 1001 using Mealy model for non overlapping

```
module sd1001_mealy(input clk, reset, din, output dout);
reg[1:0]  state;
parameter S0=2'b00,S1=2'b01,S2=2'b10,S3=2'b11;
always @(posedge clk or posedge reset) begin
 if(reset) begin
     dout <= 1'b0;
     state <= S0;
   end
   else begin
     case(state)
       S0: begin
         if(din) begin
           state <= S1;
           dout <=1'b0;
         end
         else
           dout <=1'b0;
       end
```

```
S1: begin
        if(~din) begin
          state <= S2;
          dout <=1'b0;
        end
        else begin
          dout <=1'b0;
        end
      end
      S2: begin
        if(~din) begin
          state <= S3;
          dout <=1'b0;
        end
        else begin
          state <= S1;
          dout <=1'b0;
        end
      end
```

```
S3: begin
        if(din) begin
          state <= S0;
          dout <=1'b1;
        end
        else begin
          state <= S0;
          dout <=1'b0;
        end
      end
    endcase
  end
end

endmodule
```

Mealy Machine
(Non-Overlapping)



Design and write verilog code for sequence detector 1001 using Moore model for overlapping

```
module sd1001_moore_over(input clk, reset, din, output dout);
reg[1:0]  state;
parameter S0=3’b000,S1=3’b001,S2=3’b010,S3=3’b011,
S4=3’b100;
always @(posedge clk or posedge reset) begin
   if(reset) begin
     dout <= 1'b0;
     state <= S0;
   end
   else begin
     case(state)
       S0: begin
         dout <=1'b0;
         if(din)
           state <= S1;
       end
       S1: begin
         dout <= 1'b0;
         if(~din)
           state <= S2;
       end
       S2: begin
         dout <= 1'b0;
         if(~din)
           state <= S3;
         else
           state <= S1;
       end
       S3: begin
         dout <= 1'b0;
         if(din)
           state <= S4;
         else
           state <= S0;
       end
       S4: begin
         dout <= 1'b1;
         if(din)
           state <= S1;
         else
           state <= S2;
       end
     endcase
   end
 end
endmodule
```

Moore Machine (Overlapping)