YORK
UNIVERSITÉ
UNIVERSITY

EECS 3201: Digital Logic Design Lecture 10

Ihab Amer, PhD, SMIEEE, P.Eng.

Blocking Vs Non-Blocking Assignment

- Example: If initially, A = 4 and B = 13
  - Non-Blocking Assignment

```
B <= A
C <= B + 1
```

C = 14

  - Blocking Assignment

```
B = A
C = B + 1
```

C = 5

Wakerly Rules for an **always** block

- Always use blocking assignment (=) in **always** blocks intended to create **combinational** logic
- Always use non-blocking assignment (<=) in **always** blocks intended to create **sequential** logic
- Do not mix blocking and non-blocking assignments in the same **always** block
- Do not make assignments to the same variable in two different **always** blocks

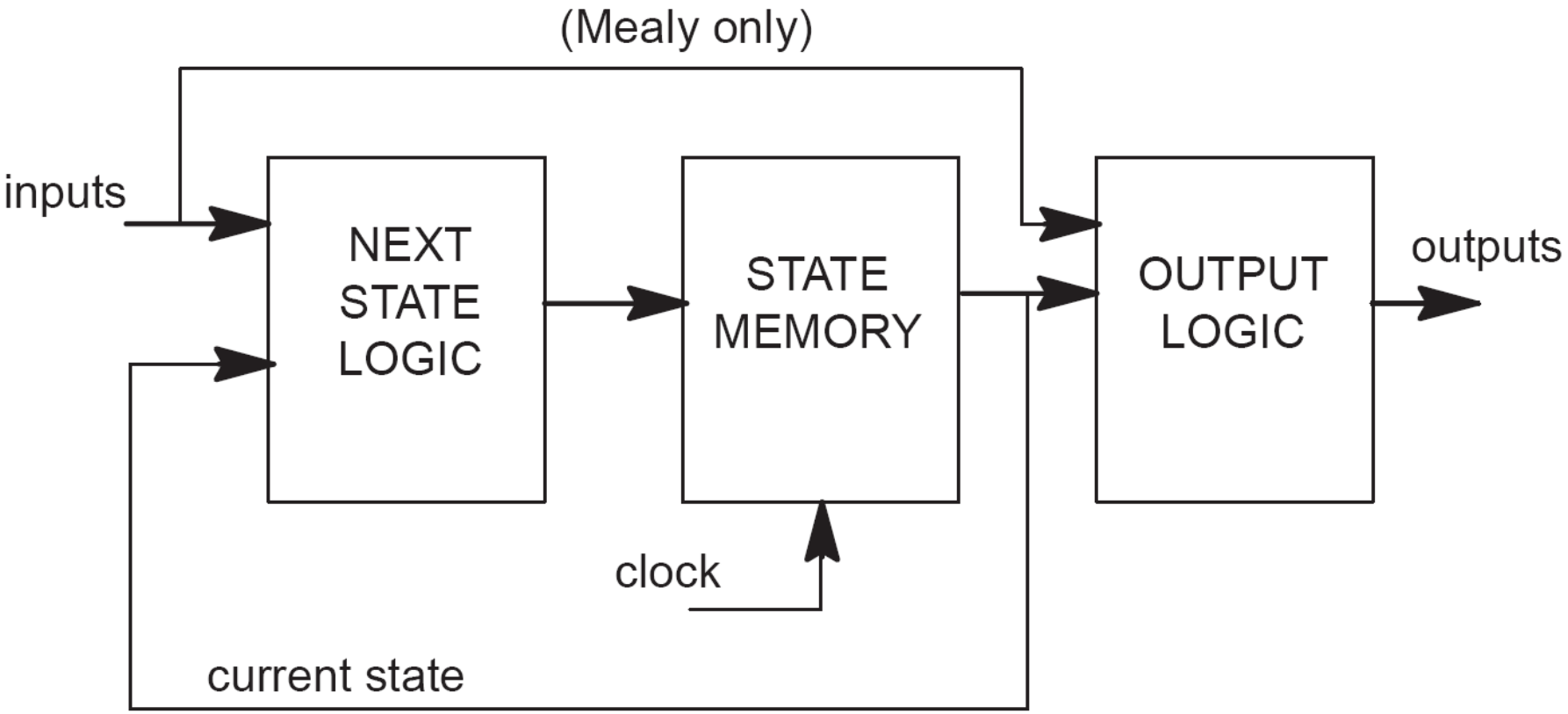
Important Book Chapters

- Related sections of chapter 5 in the textbook

Finite State Machines

- Sequential circuits are sometimes called finite state machines
- The name comes from the fact that such circuits have a finite number of possible states ($\leq 2^n$), where n is the number of storage elements (e.g. FF's) in the circuit

State Machine Structure

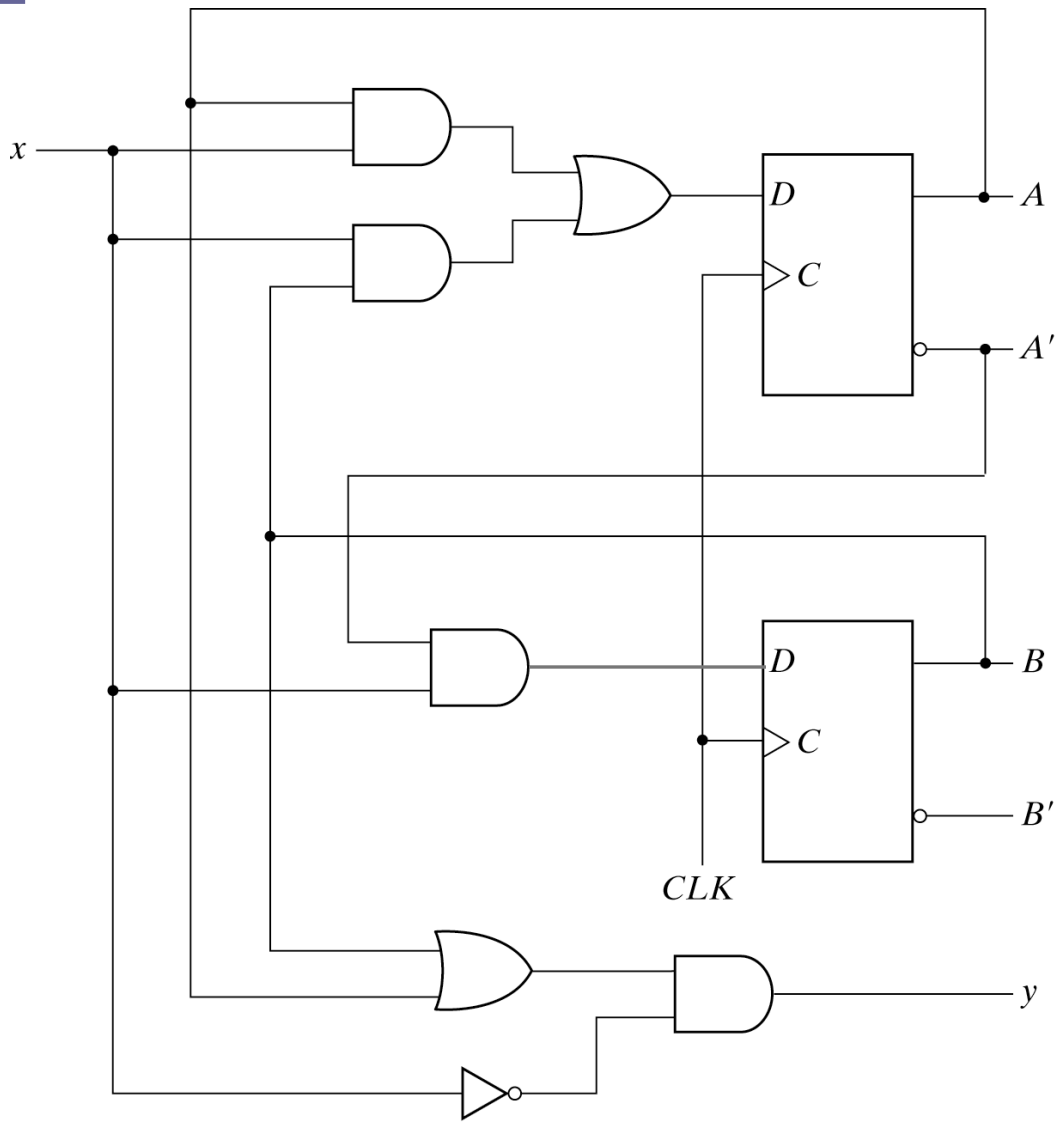


I. Analysis of Sequential Circuits

- Given a sequential circuit, obtain the state table, state diagram, and/or the timing diagram

Example 1

Analyze the following sequential circuit:



FF Input Equation(s)

$D_A = Ax + Bx$

$D_B = A'x$

Output Equation(s)

$y = (A + B)\,x'$

State Table

By using the FF ch's table

By substitution in the o/p eq.

Present State		Input	Next State		Output	FF Inputs	
A	B	x	A	B	y	D_A	D_B
0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	1
0	1	0	0	0	1	0	0
0	1	1	1	1	0	1	1
1	0	0	0	0	1	0	0
1	0	1	1	0	0	1	0
1	1	0	0	0	1	0	0
1	1	1	1	0	0	1	0

FF Input Equation(s)

$D_A = Ax + Bx$

$D_B = A'x$

Output Equation(s)

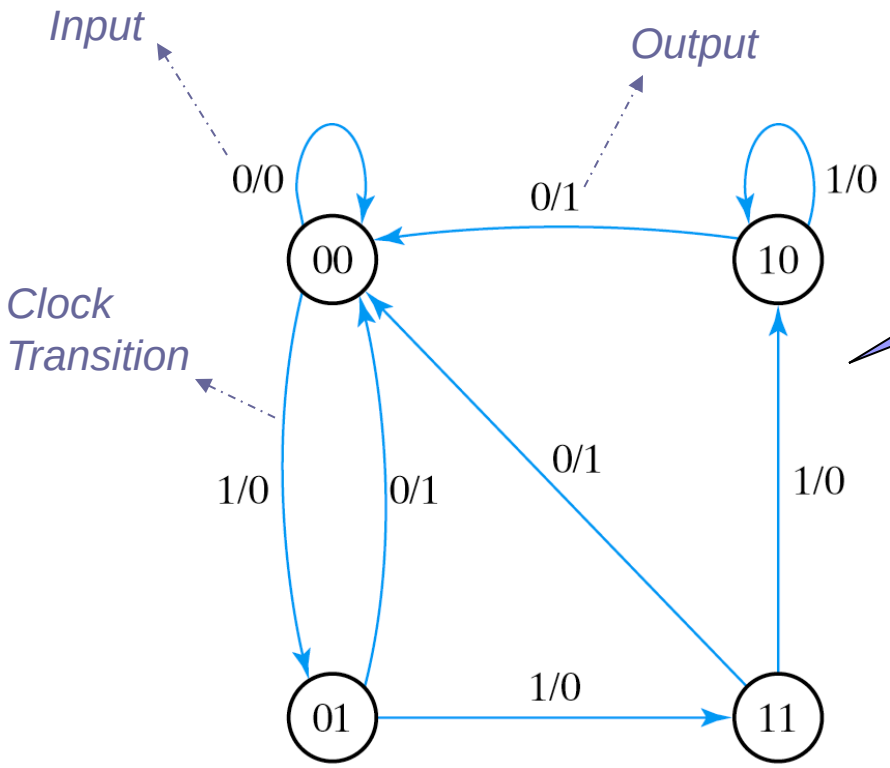
$y = (A + B)\,x'$

By substitution in the FF i/p eq.

D–FF Ch's Table

D	Q(t+1)
0	0
1	1

State Diagram



Mealy Machine

Output depends on current state and INPUT

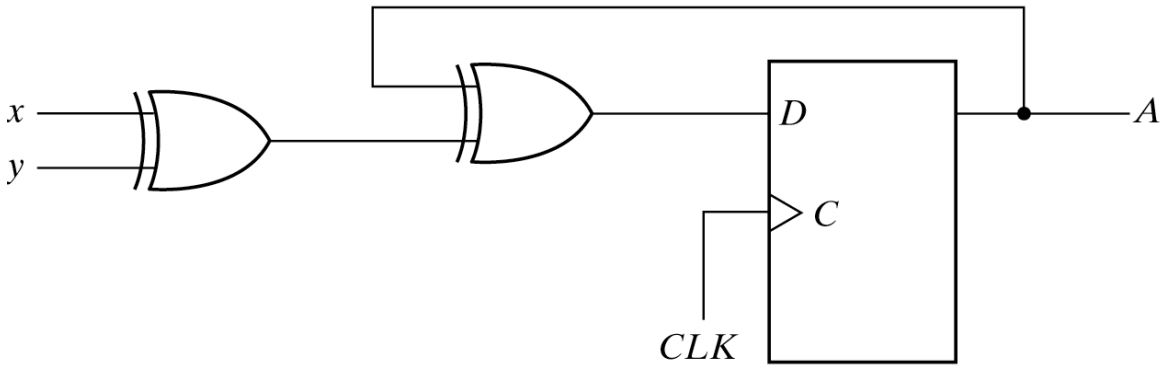
HDL for Mealy FSM

```
module Mealy_mdl (x,y,CLK,RST);
  input x,CLK,RST;
  output y;
  reg y;
  reg [1:0] Prstate, Nxtstate;
  parameter S0 = 2'b00, S1 = 2'b01, S2 = 2'b10, S3 = 2'b11;
   always @ (posedge CLK or negedge RST)
      if (~RST) Prstate = S0;  //Initialize to state S0
      else Prstate = Nxtstate; //Clock operations
   always @ (Prstate or x)     //Determine next state
         case (Prstate)
            S0: if (x) Nxtstate = S1;
            S1: if (x) Nxtstate = S3;
                else Nxtstate = S0;
            S2: if (~x)Nxtstate = S0;
            S3: if (x) Nxtstate = S2;
                else Nxtstate = S0;
         endcase
```

There are other consistent and non-consistent styles to code a Mealy FSM

```
always @ (Prstate or x)     //Evaluate output
         case (Prstate)
            S0: y = 0;
            S1: if (x) y = 1'b0; else y = 1'b1;
            S2: if (x) y = 1'b0; else y = 1'b1;
            S3: if (x) y = 1'b0; else y = 1'b1;
         endcase
endmodule
```

Example 2



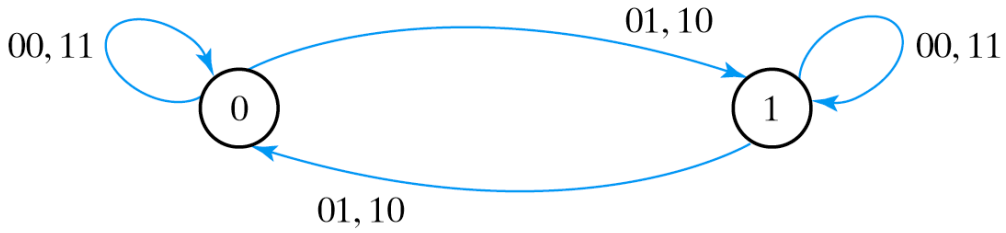
(a) Circuit diagram

Present state	Inputs		Next state
A	x	y	A
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

(b) State table

FF Input Equation

$D_A = A \oplus x \oplus y$

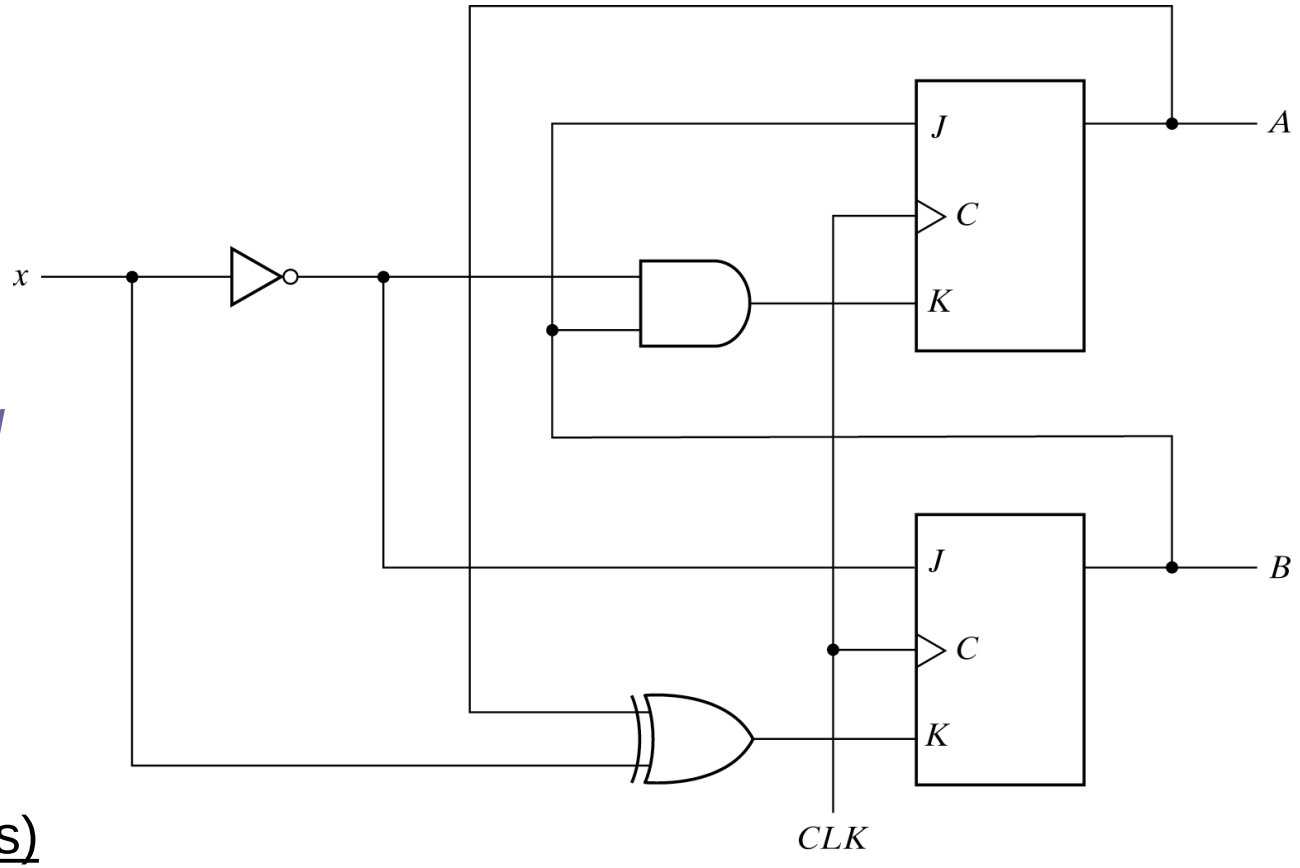


Moore Machine

(c) State diagram

Example 3

Analyze the following sequential circuit:



FF Input Equation(s)

$J_A = B$ $K_A = Bx'$

$J_B = x'$ $K_B = A \oplus x$

State Table

Present State		Input	Next State		FF Inputs			
A	B	x	A	B	J_A	K_A	J_B	K_B
0	0	0	0	1	0	0	1	0
0	0	1	0	0	0	0	0	1
0	1	0	1	1	1	1	1	0
0	1	1	1	0	1	0	0	1
1	0	0	1	1	0	0	1	1
1	0	1	1	0	0	0	0	0
1	1	0	0	0	1	1	1	1
1	1	1	1	1	1	0	0	0

FF Input Equation(s)

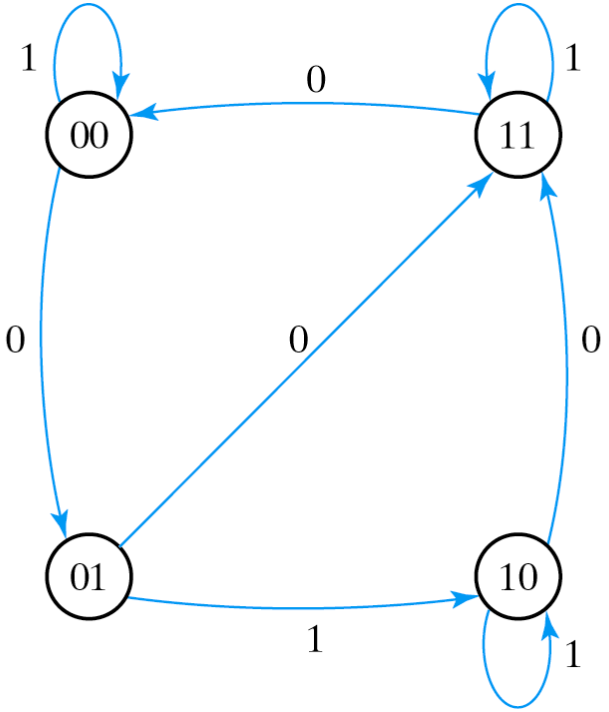
$J_A = B$ $\quad K_A = Bx'$

$J_B = x'$ $\quad K_B = A \oplus x$

JK–FF Ch's Table

J	K	Q(t+1)
0	0	Q(t)
0	1	0
1	0	1
1	1	Q′(t)

State Diagram



Moore Machine

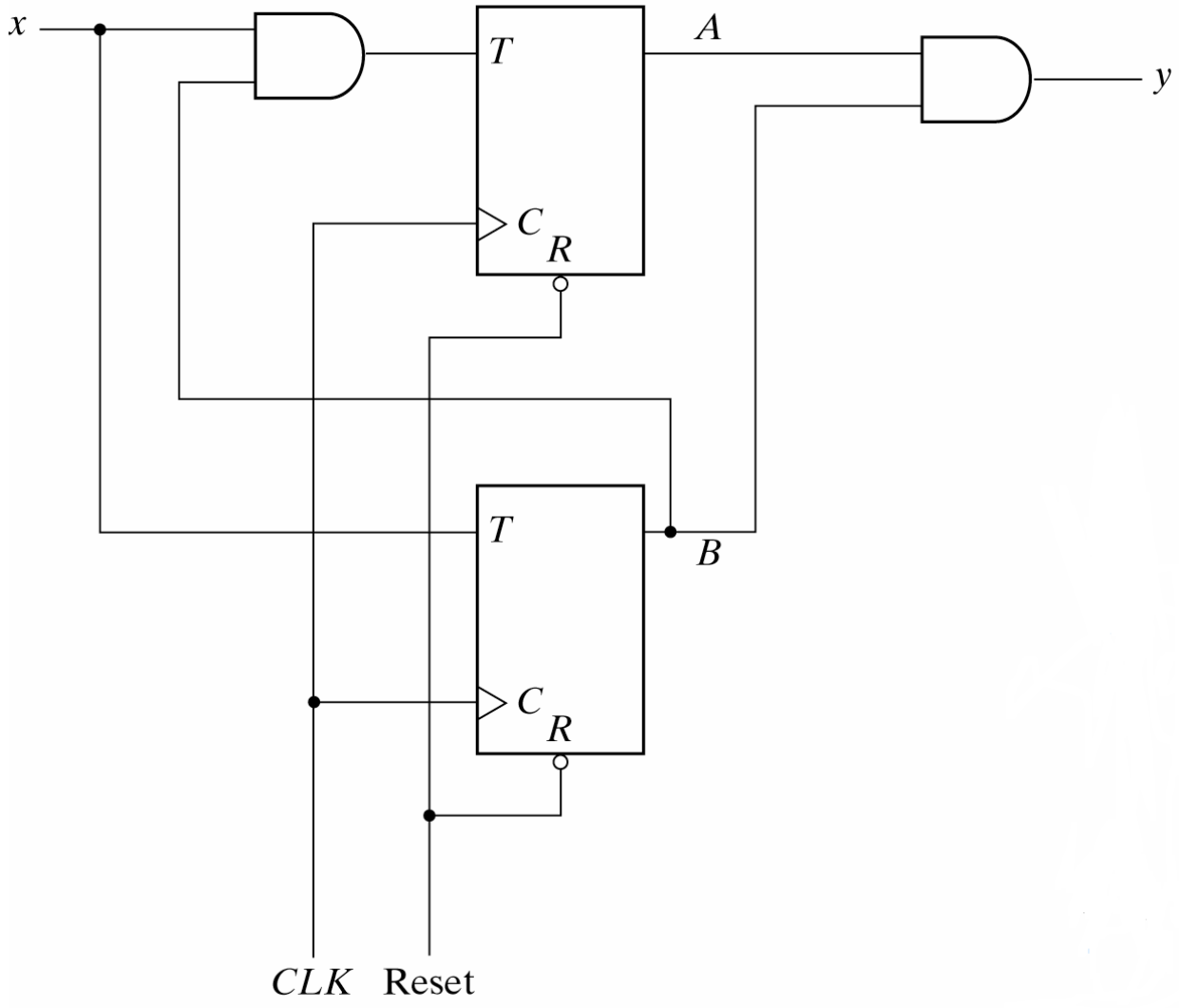
HDL for Moore FSM

```
module Moore_mdl (x,AB,CLK,RST);
   input x,CLK,RST;
   output [1:0]AB;
   reg [1:0] state;
   parameter S0 = 2'b00, S1 = 2'b01, S2 = 2'b10, S3 = 2'b11;
      always @ (posedge CLK or negedge RST)
         if (~RST) state = S0;  //Initialize to state S0
        else
        case (state)
           S0: if (~x) state = S1;          // else state = S0;
           S1: if (x)  state = S2; else state = S3;
           S2: if (~x) state = S3;          // else state = S2;
           S3: if (~x) state = S0;          // else state = S3;
         endcase
   assign AB = state;       //Output of flip-flops
endmodule
```

There are other consistent and non-consistent styles to code a Moore FSM

Example 4

Analyze the following sequential circuit:



FF Input Equation(s)

$T_A = Bx$

$T_B = x$

Output Equation(s)

$y = AB$

State Table

Present State		Input	Next State		Output	FF Inputs	
A	B	x	A	B	y	T_A	T_B
0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	1
0	1	0	0	1	0	0	0
0	1	1	1	0	0	1	1
1	0	0	1	0	0	0	0
1	0	1	1	1	0	0	1
1	1	0	1	1	1	0	0
1	1	1	0	0	1	1	1

FF Input Equation(s)

$T_A = Bx$

$T_B = x$

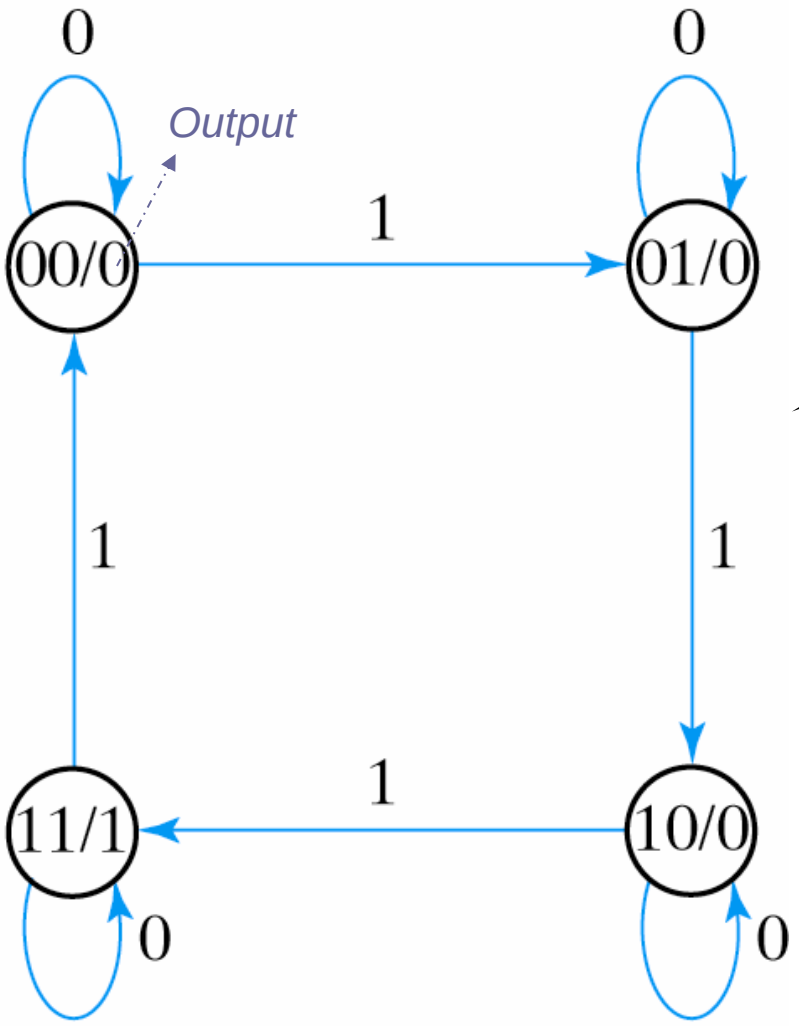
Output Equation(s)

$y = AB$

T–FF Ch's Table

T	Q(t+1)
0	Q(t)
1	Q′(t)

State Diagram



Moore Machine

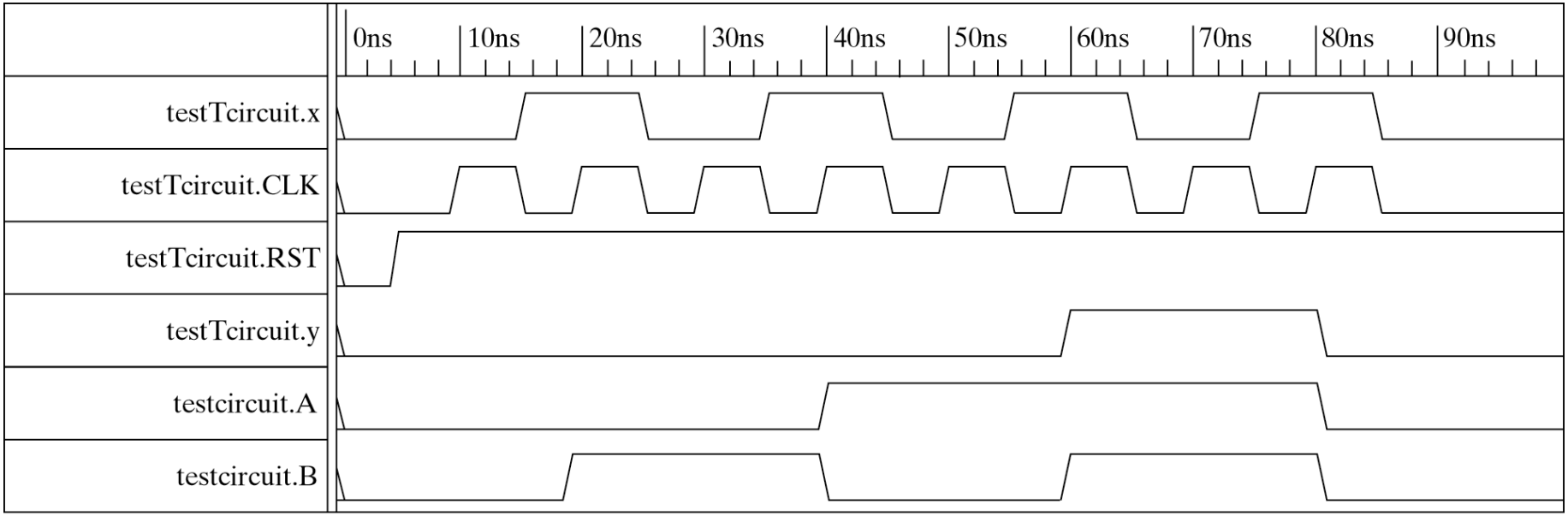
Output depends on current state ONLY

Structural Description of FSM

```
module Tcircuit (x,y,A,B,CLK,RST);
   input x,CLK,RST;
   output y,A,B;
   wire TA,TB;
//Flip-flip input equations
   assign TB = x,
          TA = x & B;
//Output equation
   assign y = A & B;
//Instantiate T flip-flops
   T_FF BF (B,TB,CLK,RST);
   T_FF AF (A,TA,CLK,RST);
endmodule
/***************************************/
//T flip-flop
module T_FF (Q,T,CLK,RST);
   output Q;
   input T,CLK,RST;
   reg Q;
     always @ (posedge CLK or negedge RST)
       if (~RST) Q = 1'b0;
       else Q = Q ^ T;
endmodule
```

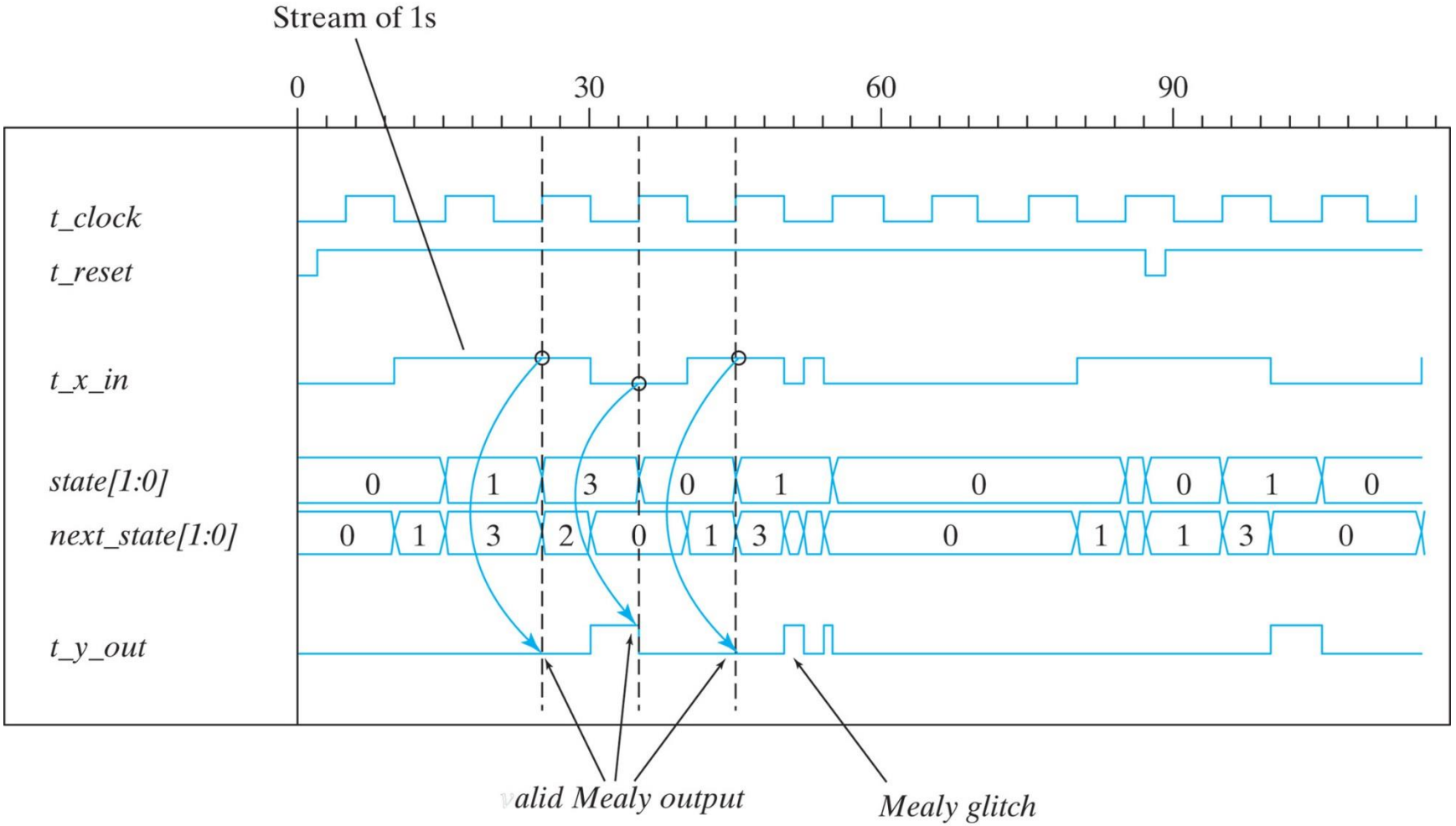
```
//Stimulus for testing sequential circuit
module testTcircuit;
  reg x,CLK,RST;  //inputs for circuit
  wire y,A,B;     //output from circuit
 // instantiate circuit
  Tcircuit TC (x,y,A,B,CLK,RST);
  initial
     begin
         RST = 0;
         CLK = 0;
      #5 RST = 1;
         repeat (16)
      #5 CLK = ~CLK;
     end
  initial
     begin
          x = 0;
      #15 x = 1;
          repeat (8)
      #10 x = ~ x;
     end
endmodule
```

Timing Diagram of FSM



What if Mealy FSM?

Example of a Mealy Glitch



Important Book Chapter

- Chapter 5 in textbook

References

- Lecture Notes of Dr. Sebastian Magierowski – Fall 2013
- “Digital Design”, Morris Mano , Prentice Hall
- “Digital Fundamentals (10th Edition)”, Thomas L. Floyd, Prentice Hall, 2010
- http://ece.gmu.edu/coursewebpages/ECE/ECE448/S10/
- cpk.auc.dk/education/SSU-2007/mm10/ssu_mm10.pdf
- www.ece.cmu.edu/~thomas/VSLIDES.pdf
- http://ece.gmu.edu/courses/ECE448/index_S06.htm
- MIT Lecture Notes on: http://www.ece.concordia.ca/~asim