COE211: Digital Logic Design

Synchronous Sequential Logic

Part 2:

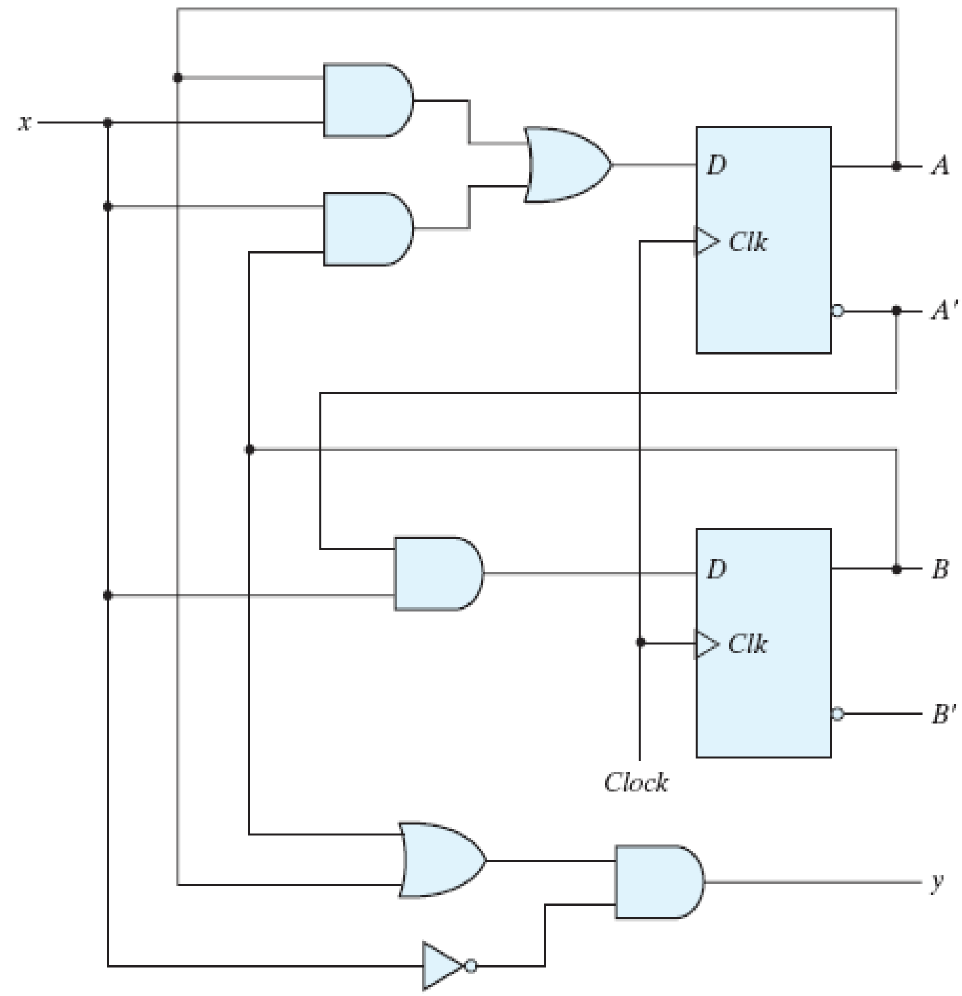
Sequential Circuits Analysis and Design

Analysis of Sequential Circuits

- Behavior of clocked sequential circuit determined by
  - Inputs
  - Outputs
  - State of flip-flops
- Analysis process
  - Consider all combinations of
    - Inputs
    - Flip-flop states
  - Determine next state and output of circuit
- Concept of a Finite State Machine (FSM)
- Methods
  - State equations
  - State table
  - State diagram

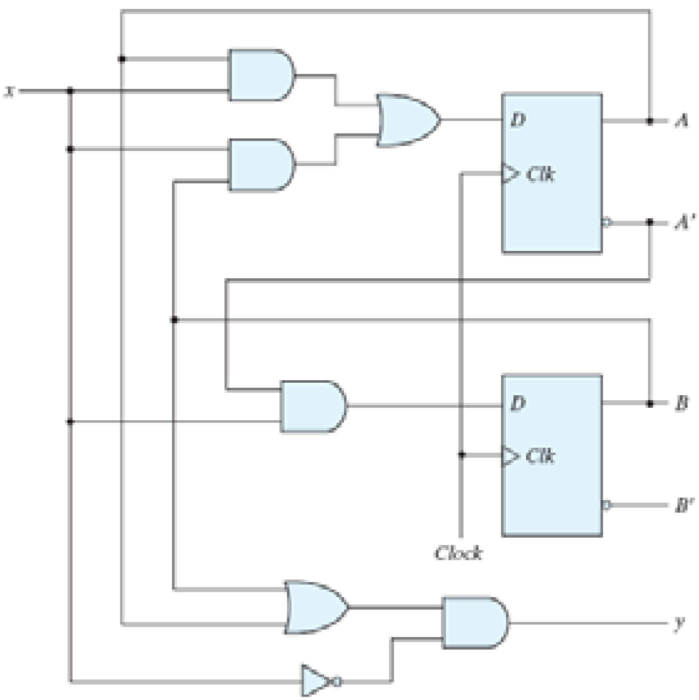
Example Circuit

- Sequential circuit
  - Input: x
  - Output: y
  - Flip-flops:
    - 2 D-type FFs A and B
- When is y=1?
  - Very difficult to answer
  - Systematic analysis necessary

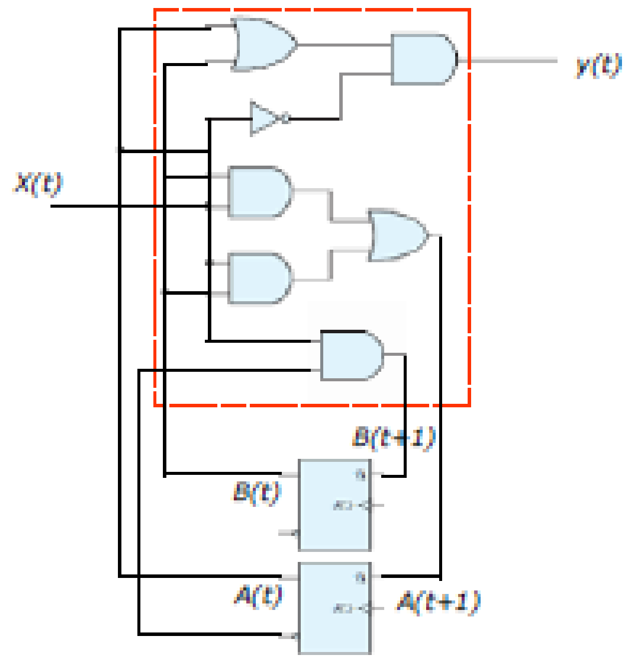


State and Output Equations

- Circuit view



FSM view



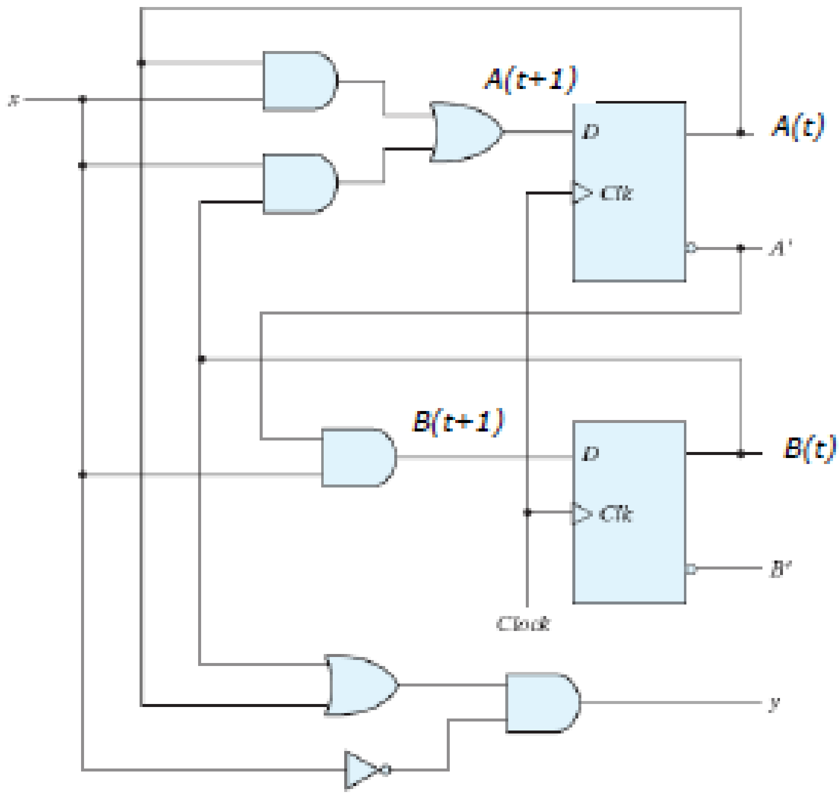
$$A(t+1) = A(t)\,x(t) + B(t)\,x(t)$$

$$B(t+1) = A'(t)\,x(t)$$

$$y(t) = (A(t) + B(t))\,x'(t)$$

State and Output Equations

- State equation specifies next state
  - Function of current state and inputs
- State equation for flip-flops:
  - $A(t+1) = A(t)\,x(t) + B(t)\,x(t)$
  - $B(t+1) = A'(t)\,x(t)$
- Output expression:
  - $y(t) = (A(t) + B(t))\,x'(t)$
- Simplified notation:
  - $A(t+1) = A\,x + B\,x$
  - $B(t+1) = A'\,x$
  - $y = (A+B)\,x'$
- How to derive these Eqs?

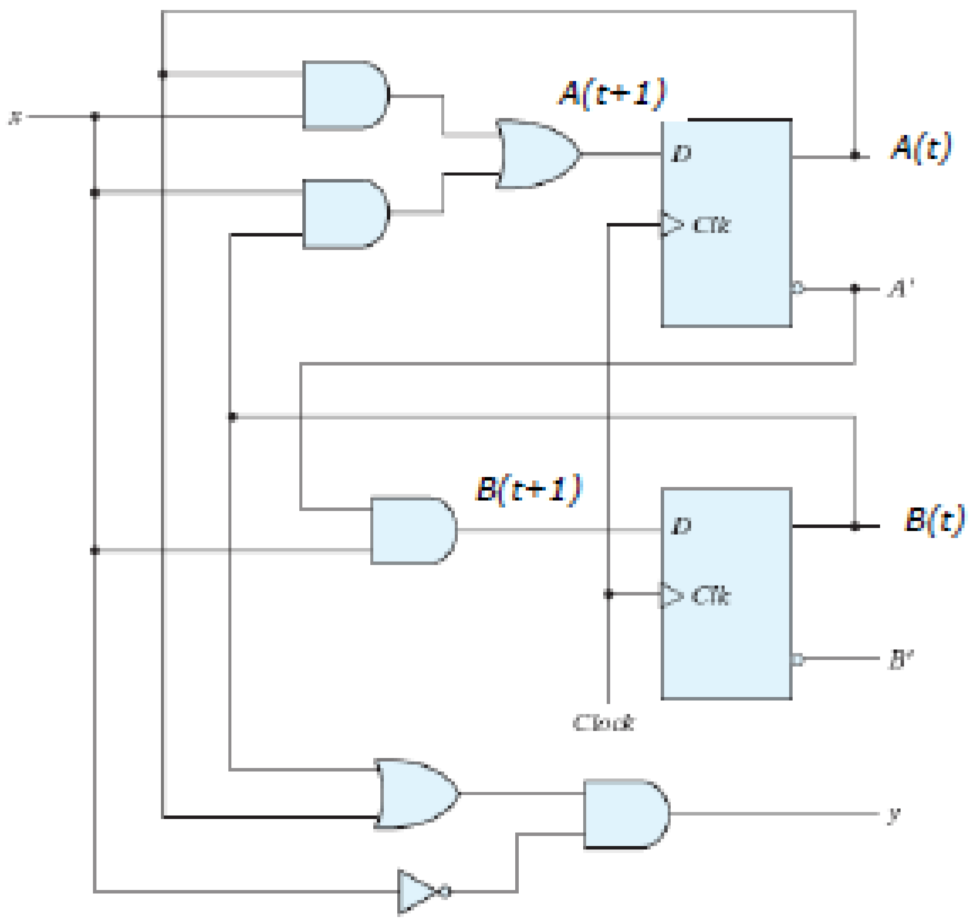


Flip-flop Input Equations

- Similar to state equations
  - Specifies type of flip-flop used
  - In case of D-FFs

    they are the

    same as state equations.
- This circuit:
  - DA = A x + B x
  - DB = A′ x
  - y = (A + B) x′



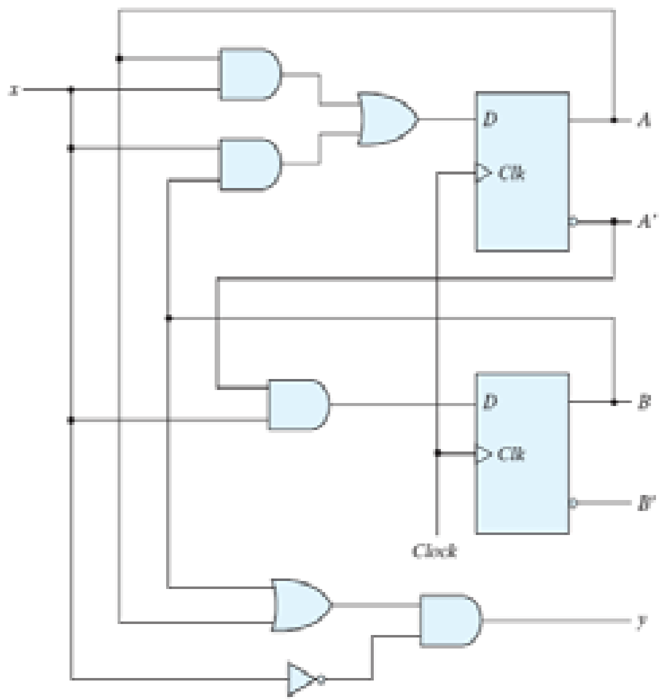
State Table

- What needs to be considered in table?
  - Inputs
  - State of flip-flops
  - Next state of flip-flops
  - Outputs
- How many entries in state table?
  - n inputs
  - m number of flip-flops
  - Total of 2^{m+n} entries
- For every entry
  - Determine flip-flop change by input and current state
    - State equation
  - Determine the output (Output equation)

State Table

Table 5.2
State Table for the Circuit of Fig. 5.15

Present State		Input	Next State		Output
A	*B*	*x*	*A*	*B*	*y*
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	1	0	0
1	1	0	0	0	1
1	1	1	1	0	0



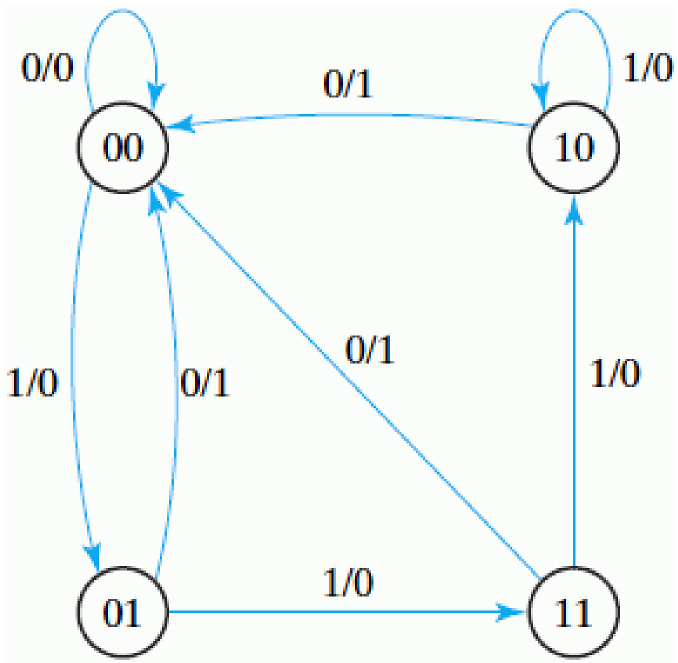
State Table

Table 5.3
Second Form of the State Table

Present State		Next State				Output	
		$x = 0$		$x = 1$		$x = 0$	$x = 1$
A	B	A	B	A	B	y	y
0	0	0	0	0	1	0	0
0	1	0	0	1	1	1	0
1	0	0	0	1	0	1	0
1	1	0	0	1	0	1	0

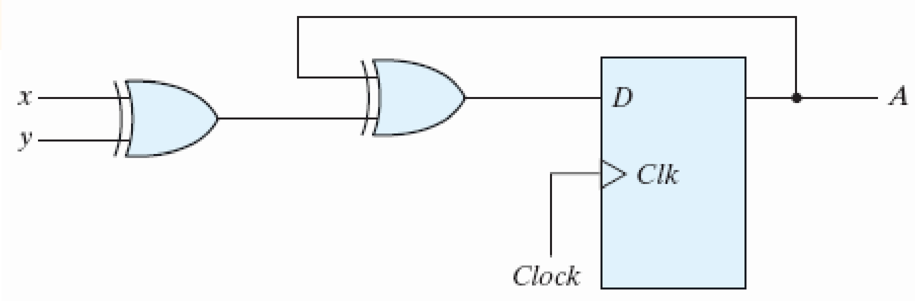
State Diagram

- State transitions represented as graph
  - Vertices indicate states
  - Edges represent transitions
    - Edge annotation: "x/y" meaning input is x and output is y



Present State		Input	Next State		Output
A	***B***	***x***	***A***	***B***	***y***
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	1	0	0
1	1	0	0	0	1
1	1	1	1	0	0

Analysis Example 2



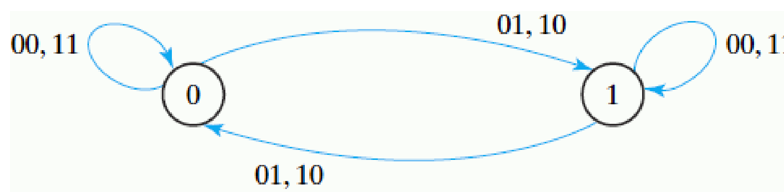
Present state	Inputs		Next state
A	x	y	A
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

- Equation:

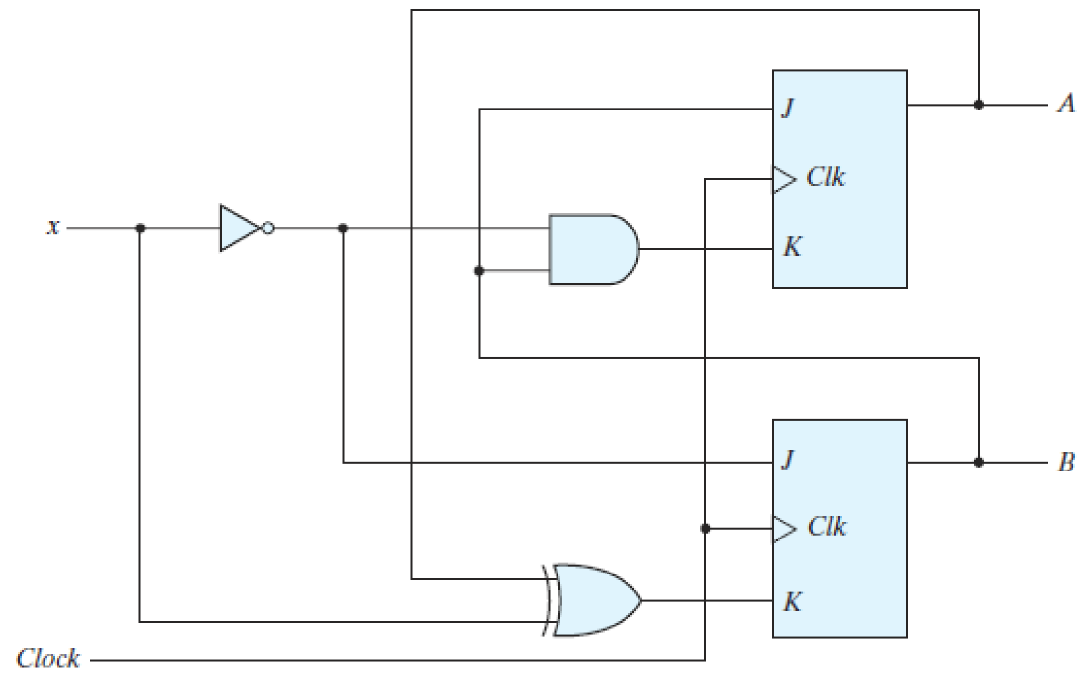
$A(t+1) = A \oplus x \oplus y$ (state equation)

$D_A(t+1) = A \oplus x \oplus y$ (flip-flop equation)

- State table:
- State diagram:
  - Note: no outputs



Analysis with JK FF



- Flip-flop Input Equations

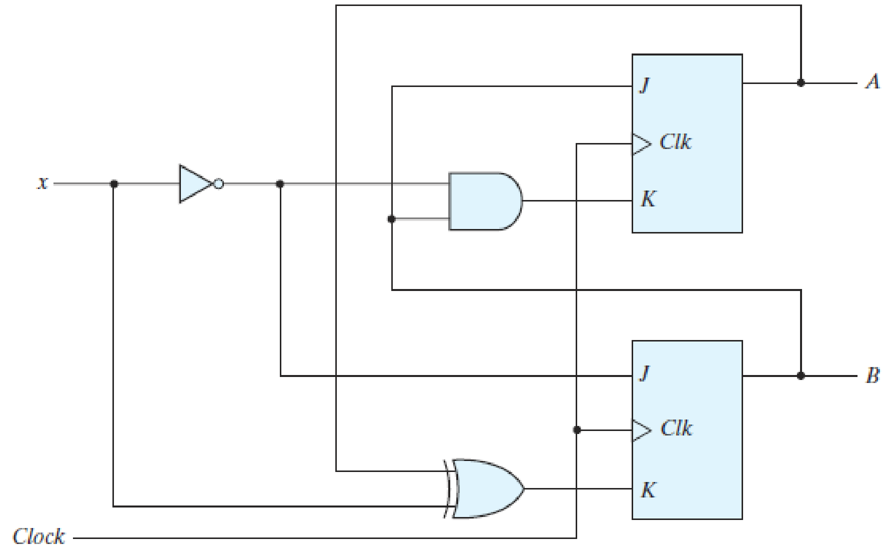
$J_A = B \quad K_A = Bx'$

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

Analysis with JK FF

- State Equations

$$A(t+1) = JA' + K'A$$
$$B(t+1) = JB' + K'B$$



$$J_A = B \quad K_A = Bx'$$
$$J_B = x' \quad K_B = A'x + Ax' = A \oplus x$$

$$A(t+1) = BA' + (Bx')'A = A'B + AB' + Ax$$

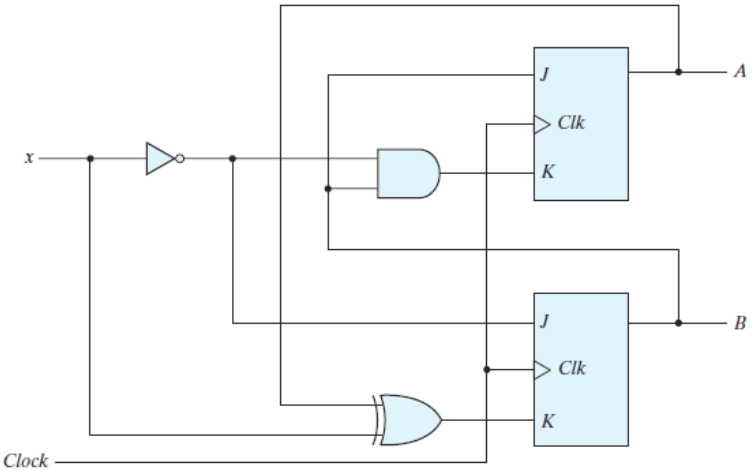
$$B(t+1) = x'B' + (A \oplus x)'B = B'x' + ABx + A'Bx'$$

Analysis with JK FF

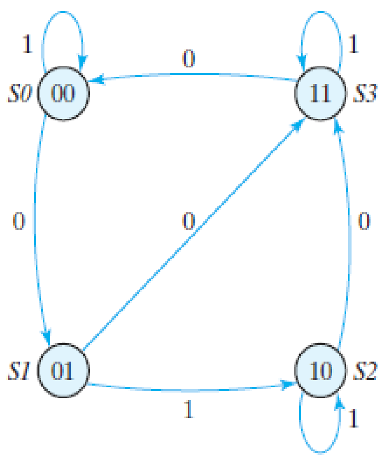
- State Equations

$$A(t+1) = BA' + (Bx')'A = A'B + AB' + Ax$$

$$B(t+1) = x'B' + (A \oplus x)'B = B'x' + ABx + A'Bx'$$



- State table:
- State Diagram:

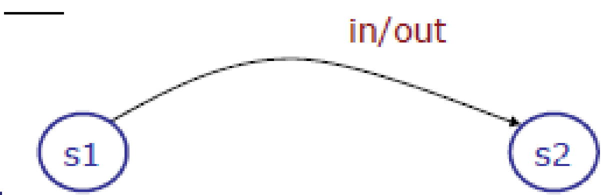


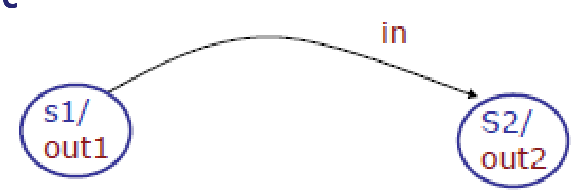
State Table for Sequential Circuit with JK Flip-Flops

Present State		Input	Next State		Flip-Flop 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

Finite State Machines

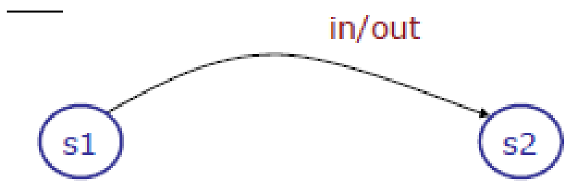
- State diagrams are representations of Finite State Machines (FSM)
- Two flavors of FSMs:
  - Mealy FSM
  - Moore FSM
- Difference:
  - How output is determined
- Mealy FSM
  - Output depends on input and state
  - Output is not synchronized with clock
  - can have temporarily unstable output
- Moore FSM
  - Output depends only on state

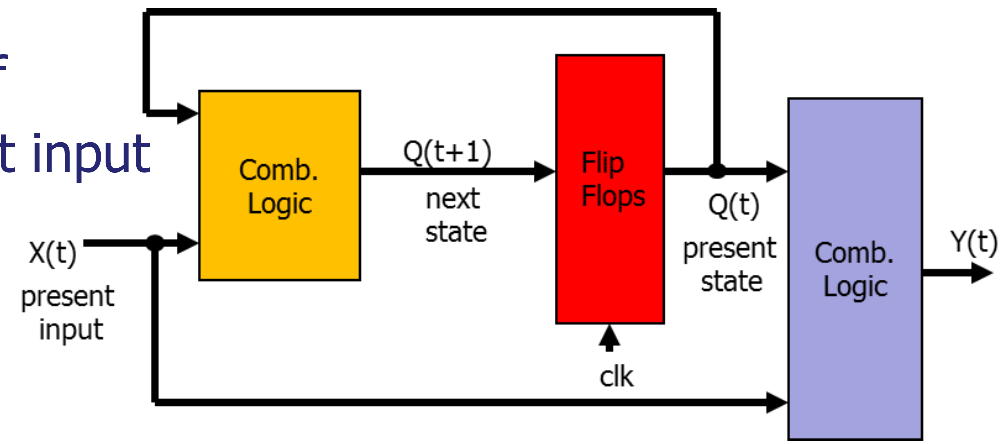




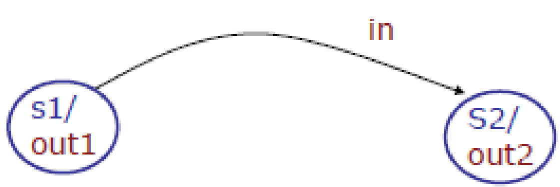
Mealy vs Moore Machine

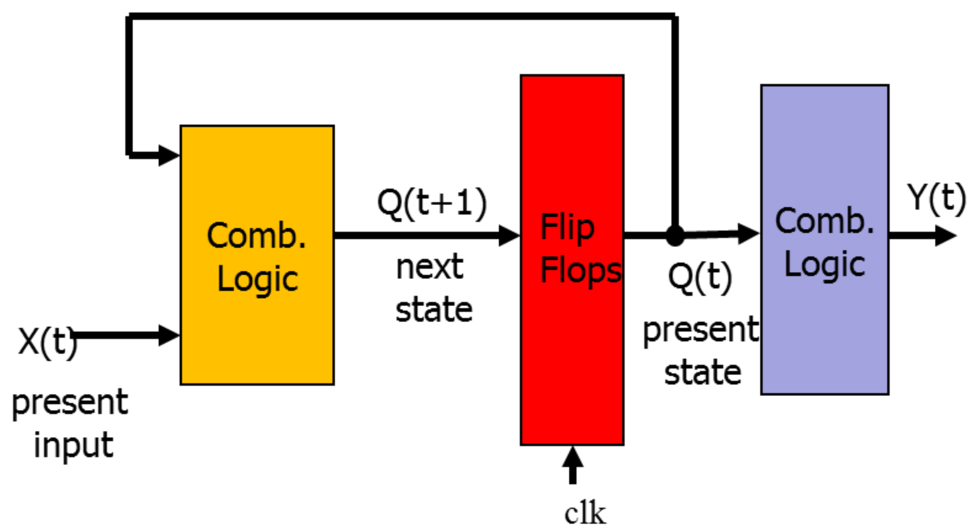
- Mealy Machine
  - Output is a function of current state and present input





- Moore Machine
  - Output is a function of current state only





Design of Sequential Circuits

Design of Sequential Circuits

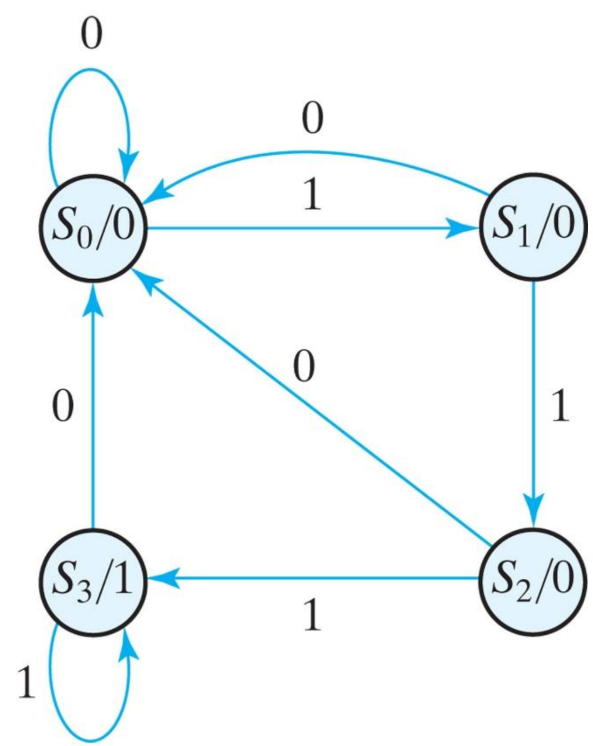
1. Derive state diagram from description

   Reduce number of states if necessary (will NOT be covered)

2. Assign binary values to states
3. Obtain binary coded state table (transition table)
4. Choose type of flip-flops
5. Derive flip-flop input equations and output equations
6. Draw logic diagram

Example 1: Sequence Detector

- Circuit specification:
  - Design a circuit that outputs a 1 when three consecutive 1s have been applied to input, and 0 otherwise."
- Step 1: derive state diagram
  - What should a state represent?
    - E.g., "number of 1's seen so far"
  - Moore or Mealy FSM?
    - Both possible
    - Chose Moore to simplify diagram
  - State diagram:
    - State S0: zero 1s detected
    - State S1: one 1 detected
    - State S2: two 1s detected
    - State S3: three 1s detected



Example 1: Sequence Detector

- Step 2: state assignment
  - Two flip-flops
  - Binary state coding
- Step 3: Binary coded state table
  - Name flip-flops A and B
- Step 4: Choose type of flip-flops
  - E.g., D flip-flop
  - Characteristic equation: Q(t+1)=DQ

Table 5.11
State Table for Sequence Detector

Present State		Input	Next State		Output
A	*B*	*x*	*A*	*B*	*y*
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	1	1	0
1	1	0	0	0	1
1	1	1	1	1	1

Example 1: Sequence Detector

- Step 6: derive flip-flop input equations and output equation
  - Use state table

$$A(t+1) = D_A(A,B,x) = \Sigma(3,5,7)$$

$$B(t+1) = D_B(A,B,x) = \Sigma(1,5,7)$$

$$y(A,B,x) = \Sigma(6,7)$$
$$\text{or } y(A,B) = \Sigma(3)$$

Table 5.11
State Table for Sequence Detector

Present State		Input	Next State		Output
A	***B***	***x***	***A***	***B***	***y***
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	1	1	0
1	1	0	0	0	1
1	1	1	1	1	1

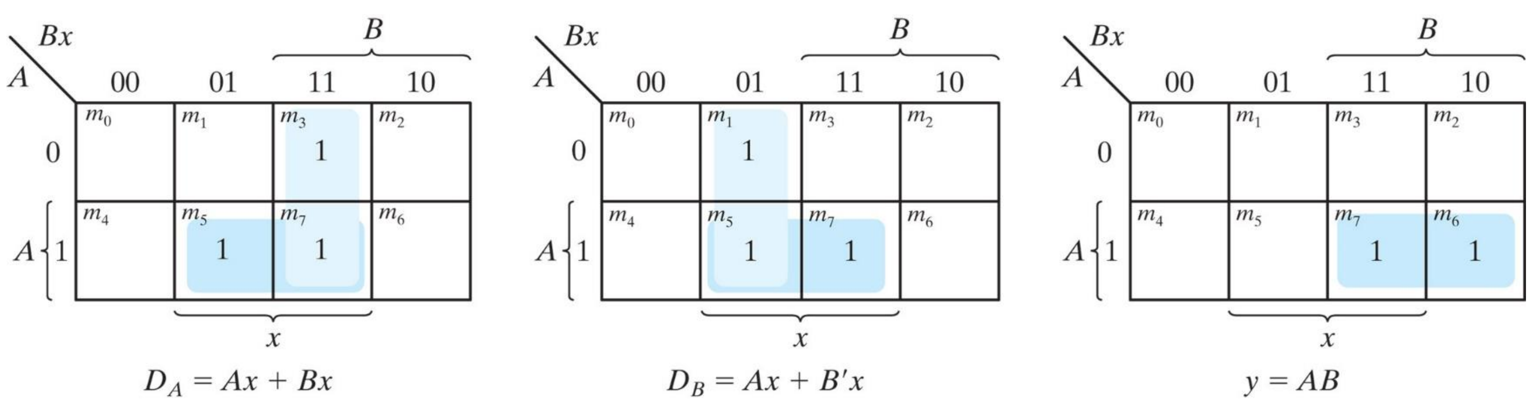
Example 1: Sequence Detector

- Step 6b: minimize equations

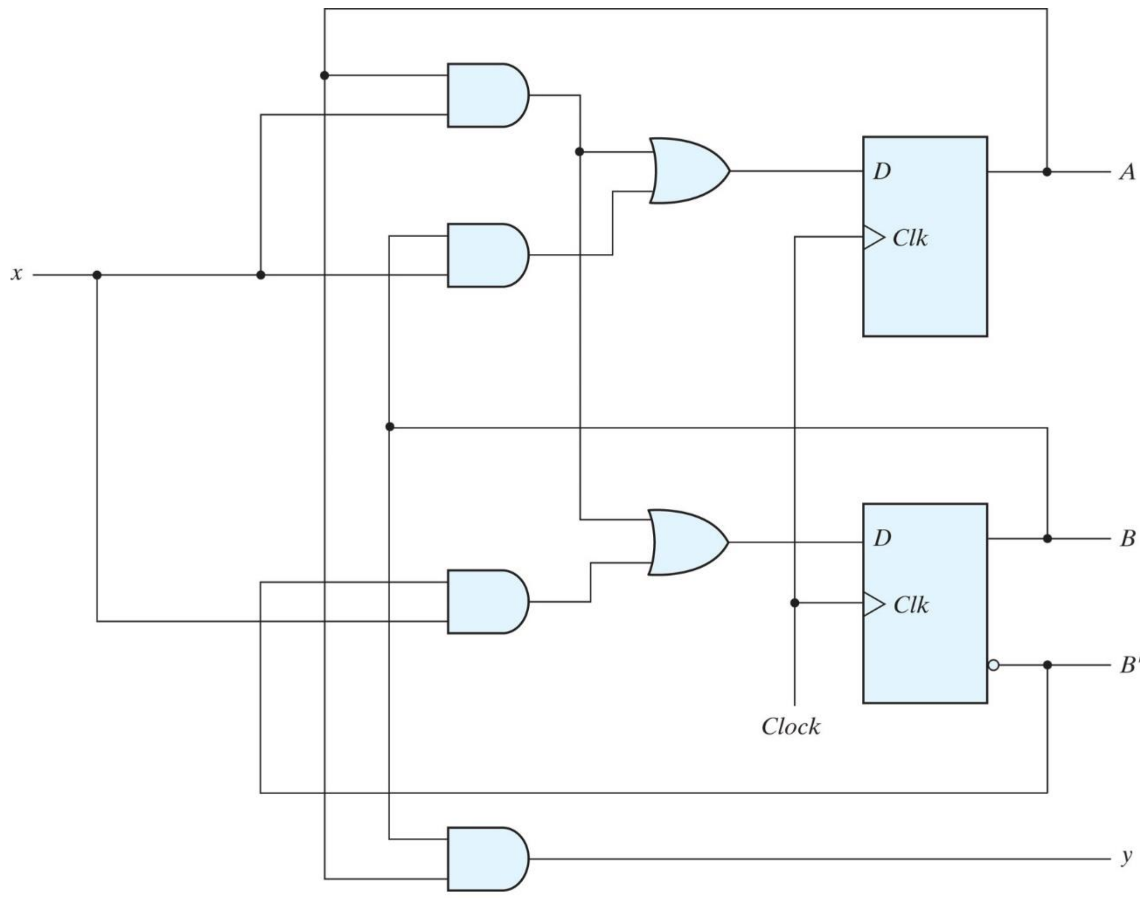
$A(t+1) = \Sigma(3,5,7)$

$B(t+1) = \Sigma(1,5,7)$

$y(A,B) = \Sigma(3)$ – easy: $y = AB$



Example 1: Sequence Detector



Summary

- Flip flops contain state information
- State can be represented in several forms:
  - State equations
  - State table
  - State diagram
- Possible to convert between these forms
- Circuits with states can take on a finite set of values
  - Finite state machine
- Two types of “machines”
  - Mealy machine
  - Moore machine