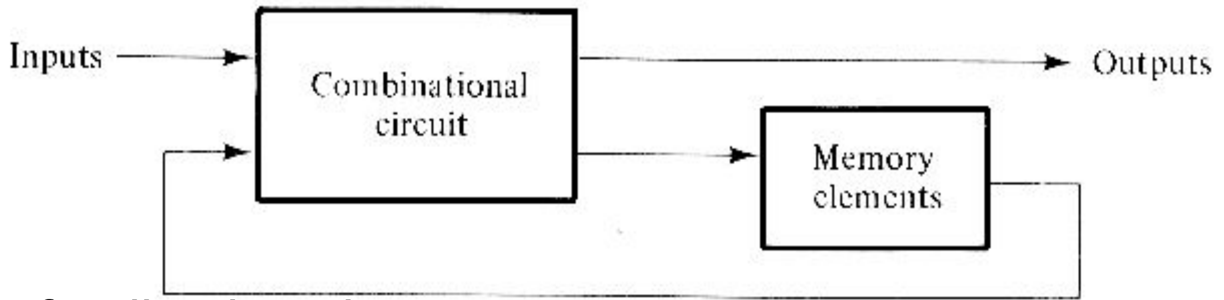
ELECTRICAL
Engineering KMITNB

Latches, Flip-flops, Synchronous Sequential Logic Analysis and Design

Chapter 7,8 (excl. registers and counters)

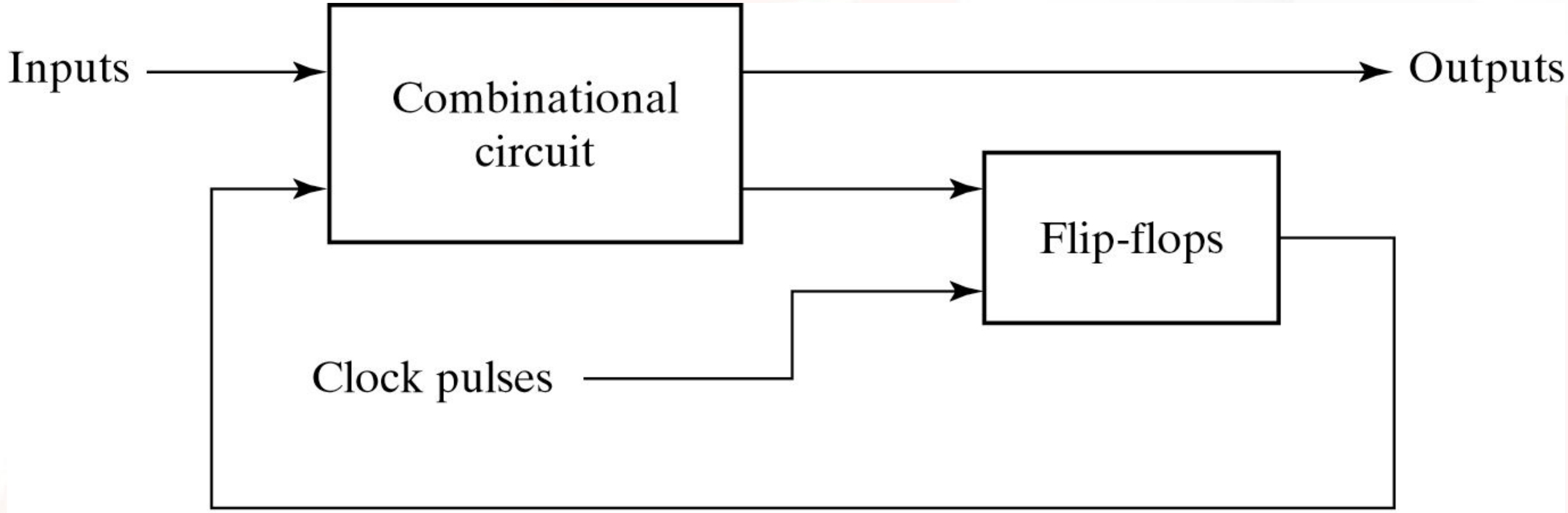
Sequential Circuits

- Combinational circuits
  - contains no memory elements
  - the outputs depends on the inputs
- Sequential circuits

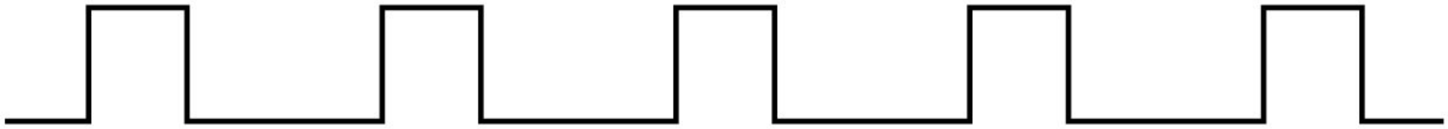


  - a feedback path
  - the state of the sequential circuit
  - (inputs, current state) $\Rightarrow$ (outputs, next state)
  - synchronous: transition happens at discrete instants of time
  - asynchronous: transition happens at any instant of time

- Synchronous sequential circuits
  - a master-clock generator to generate a periodic train of clock pulses
  - the clock pulses are distributed throughout the system
  - clocked sequential circuits
  - most commonly used
  - no instability problems
  - the memory elements: flip-flops
    - binary cells capable of storing one bit of information
    - two outputs: one for the normal value and one for the complement value
    - maintain a binary state indefinitely until directed by an input signal to switch states

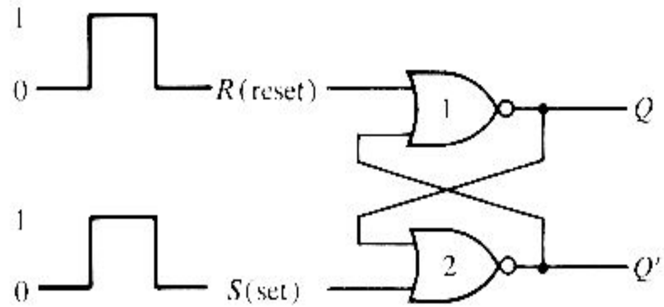


(a) Block diagram


(b) Timing diagram of clock pulses

Latches

- Basic flip-flop circuit
  - two NOR gates



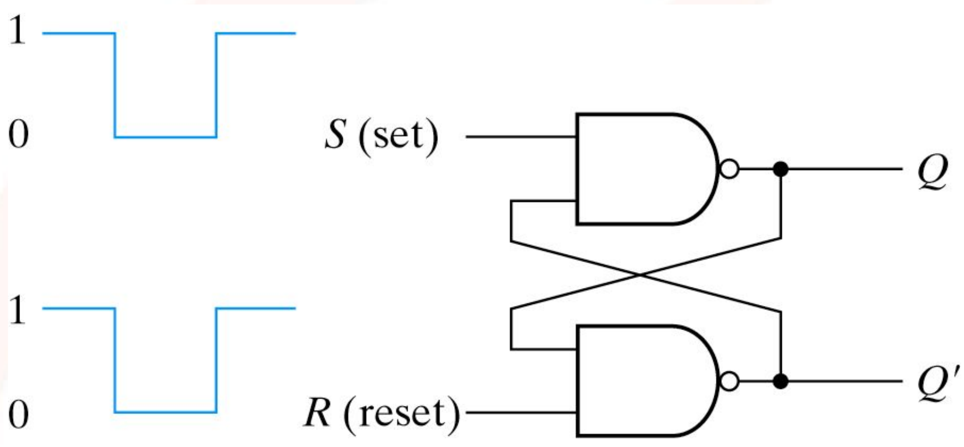
(a) Logic diagram

S	R	Q	Q'	
1	0	1	0	
0	0	1	0	(after S = 1, R = 0)
0	1	0	1	
0	0	0	1	(after S = 0, R = 1)
1	1	0	0	

(b) Truth table

  - more complicated types can be built upon it
  - directed-coupled RS flip-flop: the cross-coupled connection
  - an asynchronous sequential circuit
  - (S,R)= (0,0): no operation
    (S,R)=(0,1): reset (Q=0, the clear state)
    (S,R)=(1,0): set (Q=1, the set state)
    (S,R)=(1,1): indeterminate state (Q=Q'=0)
  - consider (S,R) = (1,1) $\Rightarrow$ (0,0)

- SR latch with NAND gates

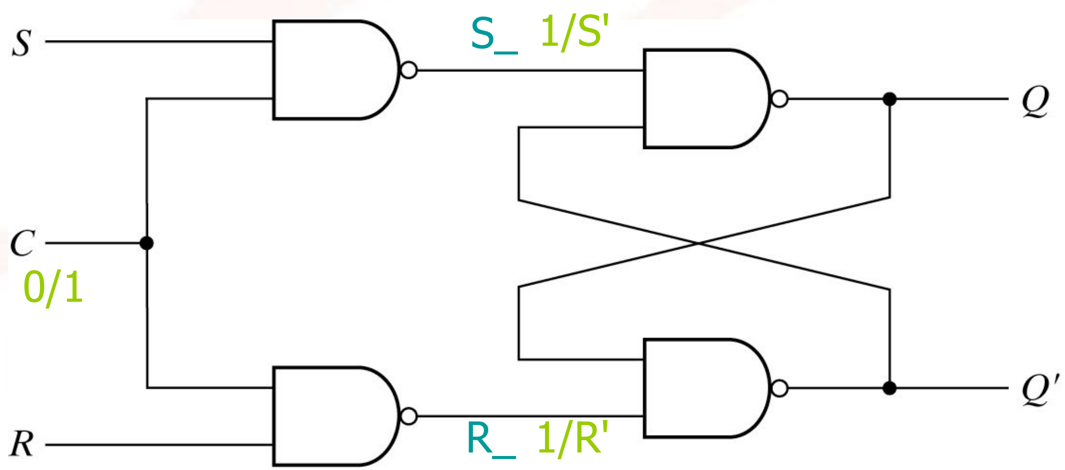


(a) Logic diagram

S	R	Q	Q'	
1	0	0	1	
1	1	0	1	(after $S = 1, R = 0$)
0	1	1	0	
1	1	1	0	(after $S = 0, R = 1$)
0	0	1	1	

(b) Function table

- SR latch with control input
  - C=0, no change
  - C=1,



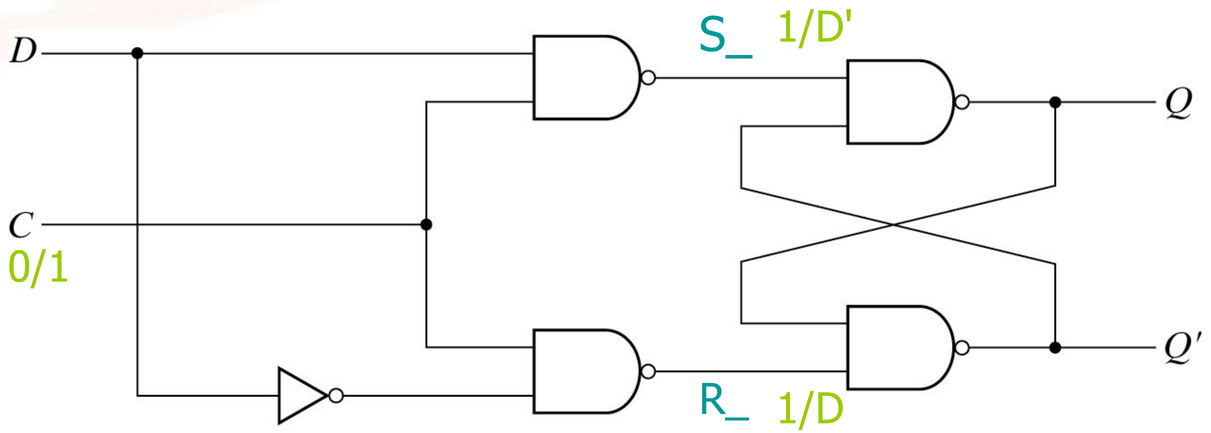
(a) Logic diagram

C	S	R	Next state of Q
0	X	X	No change
1	0	0	No change
1	0	1	$Q = 0$; Reset state
1	1	0	$Q = 1$; set state
1	1	1	Indeterminate

(b) Function table

D Latch

- eliminate the undesirable conditions of the indeterminate state in the RS flip-flop
- D: data
- gated D-latch
- D $\Rightarrow$ Q when C=1; no change when C=0

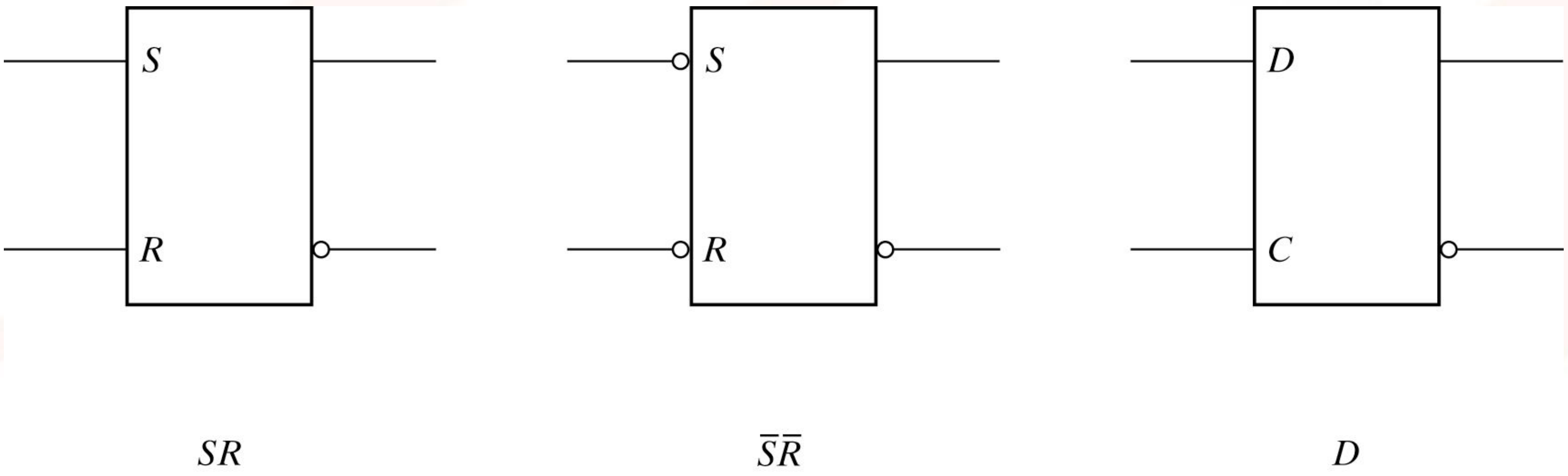


(a) Logic diagram

C	D	Next state of Q
0	X	No change
1	0	$Q = 0$; Reset state
1	1	$Q = 1$; Set state

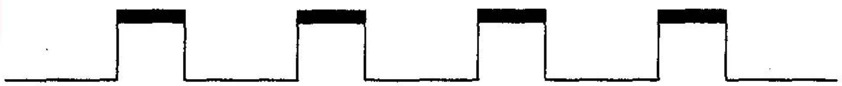
(b) Function table

Graphic Symbols for Latches

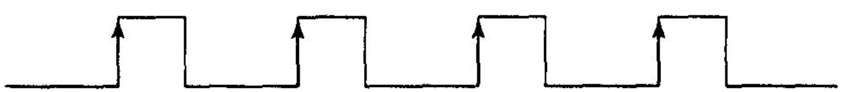


Flip-Flops

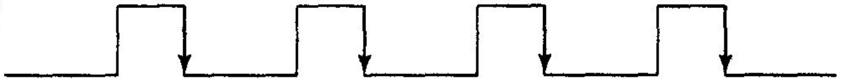
- A trigger
  - The state of a latch or flip-flop is switched by a change of the control input
- Level triggered – latches
- Edge triggered – flip-flops



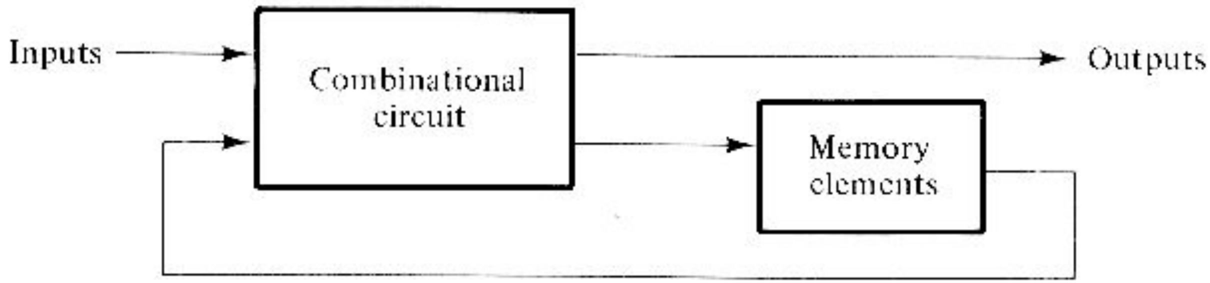
(a) Response to positive level



(b) Positive-edge response



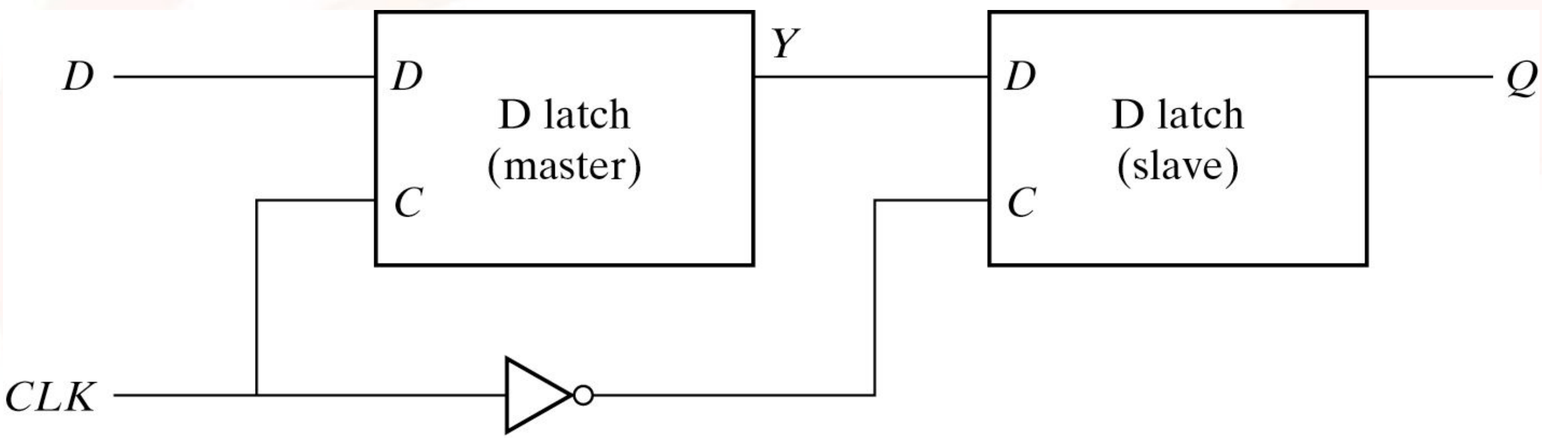
(c) Negative-edge response



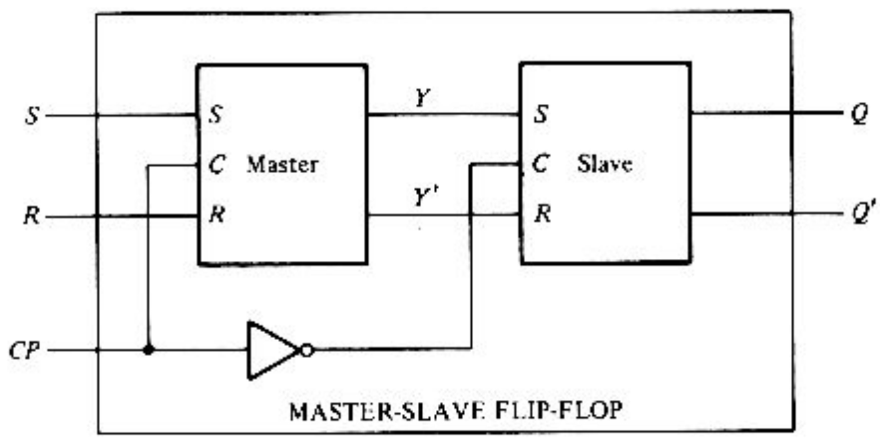
- If level-triggered flip-flops are used
  - the feedback path may cause instability problem
- Edge-triggered flip-flops
  - the state transition happens only at the edge
  - eliminate the multiple-transition problem

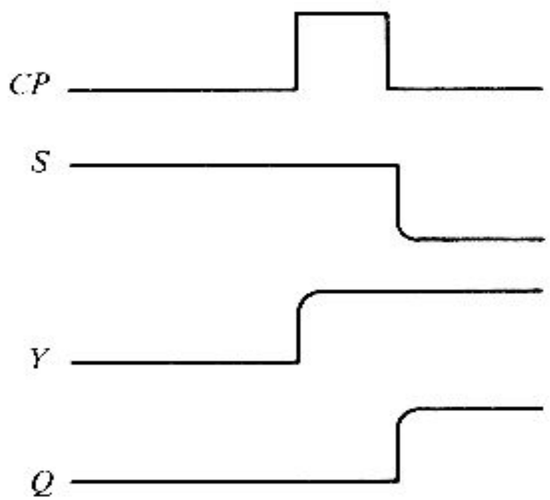
Edge-triggered D flip-flop

- Master-slave D flip-flop
  - two separate flip-flops
  - a master flip-flop (positive-level triggered)
  - a slave flip-flop (negative-level triggered)

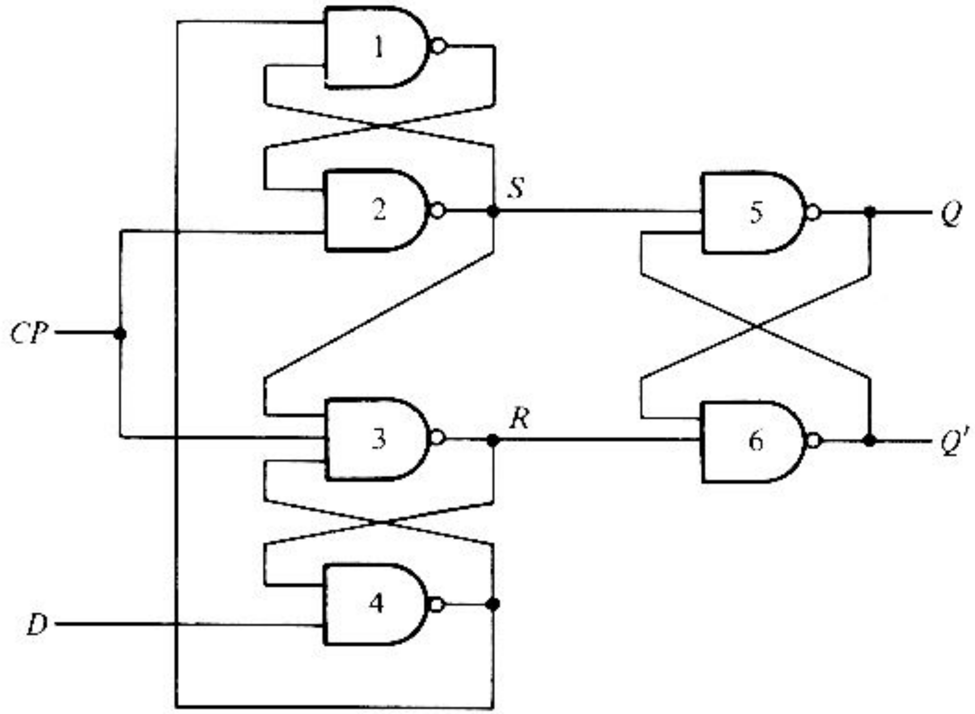


- CP = 1: (S,R) $\Rightarrow$ (Y,Y'); (Q,Q') holds
- CP = 0: (Y,Y') holds; (Y,Y') $\Rightarrow$ (Q,Q')
- (S,R) could not affect (Q,Q') directly
- the state changes coincide with the negative-edge transition of CP

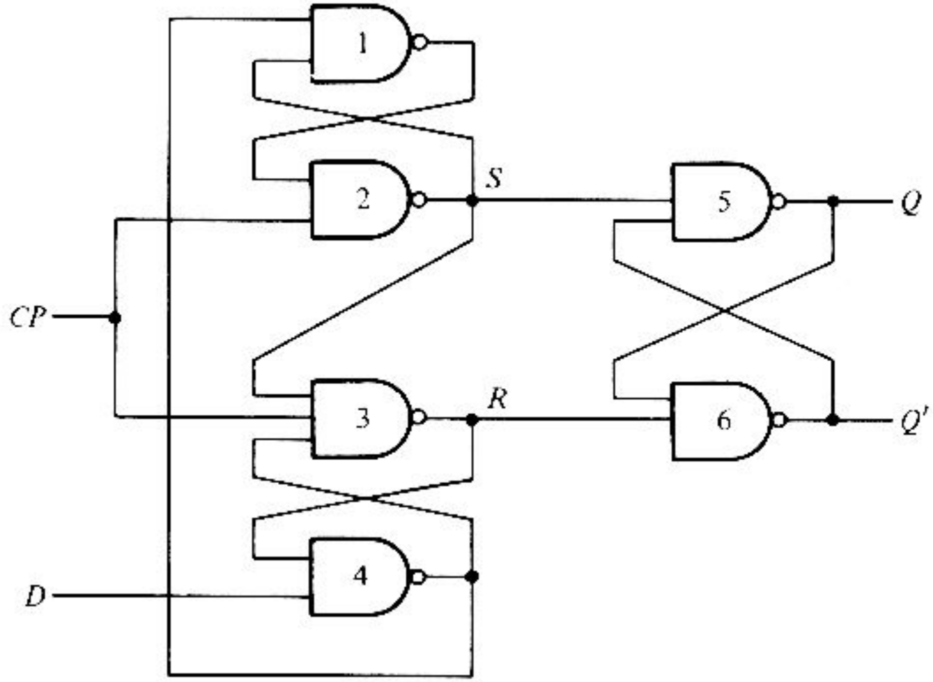


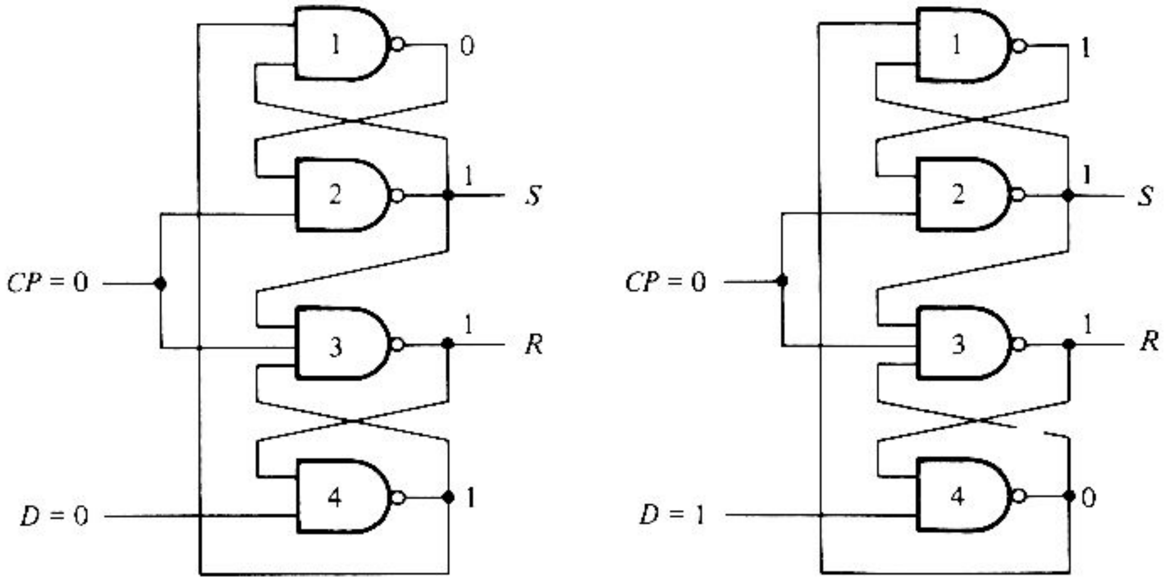


- Edge-triggered flip-flops
  - the state changes during a clock-pulse transition
- A D-type positive-edge-triggered flip-flop

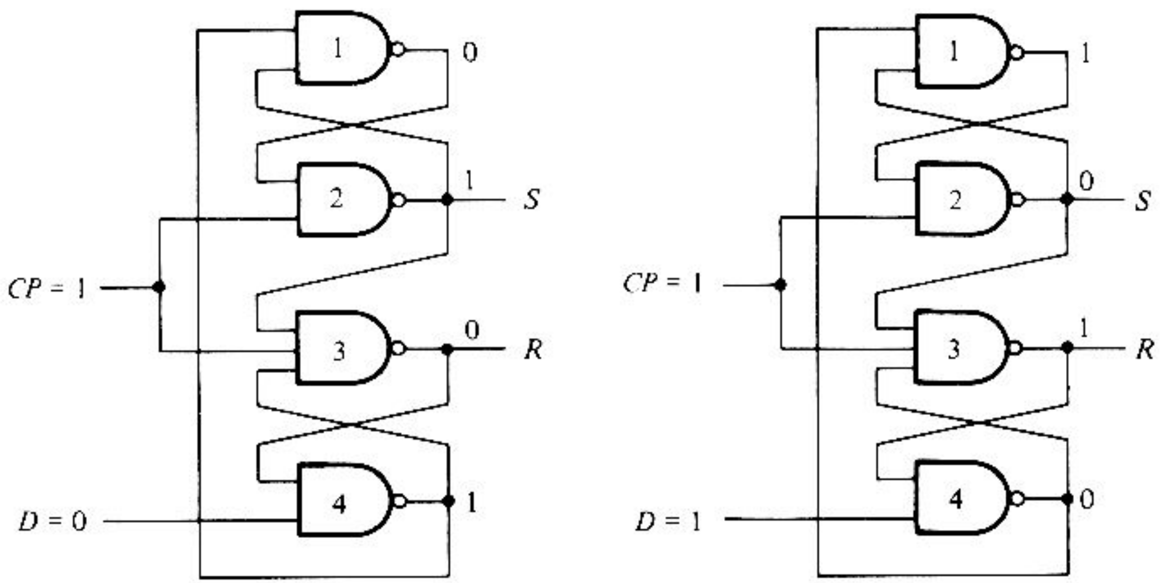


- three basic flip-flops
- (S,R) = (0,1): Q = 1
- (S,R) = (1,0): Q = 0
- (S,R) = (1,1): no operation
- (S,R) = (0,0): should be avoided





(a) With $CP = 0$



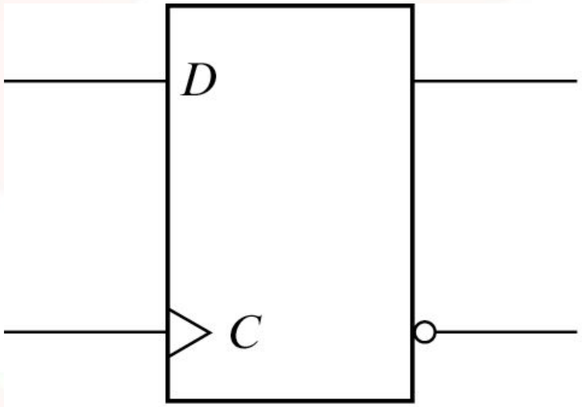
(b) With $CP = 1$

- The setup time
  - D input must be maintained at a constant value prior to the application of the positive CP pulse
  - = the propagation delay through gates 4 and 1
  - data to the internal latches
- The hold time
  - D input must not changes after the application of the positive CP pulse
  - = the propagation delay of gate 3
  - clock to the internal latch

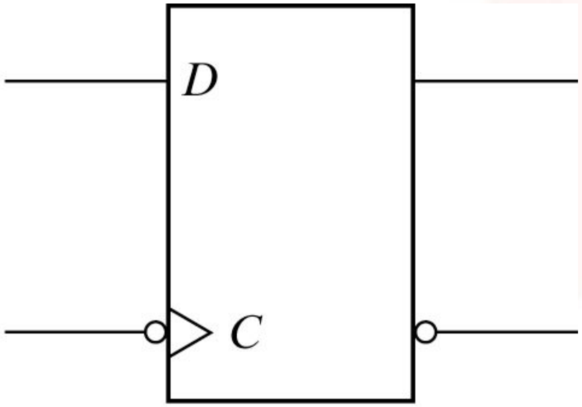
- Summary
  - CP=0: (S,R) = (1,1), no state change
  - CP=↑: state change once
  - CP=1: state holds
  - eliminate the feedback problems in sequential circuits
- All flip-flops must make their transition at the same time

Other Flip-Flops

- The edge-triggered D flip-flops
  - The most economical and efficient
  - Positive-edge and negative-edge

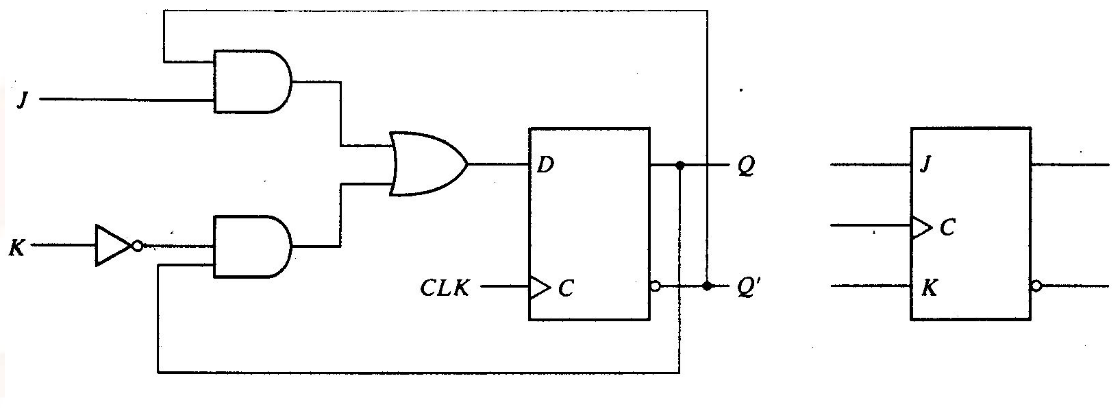


Positive-edge



Negative-edge

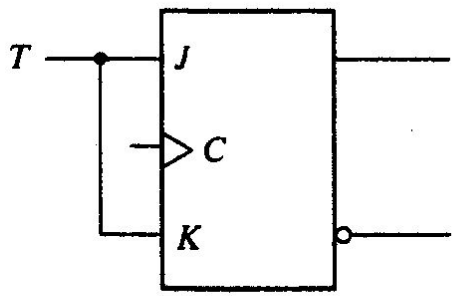
- JK flip-flop



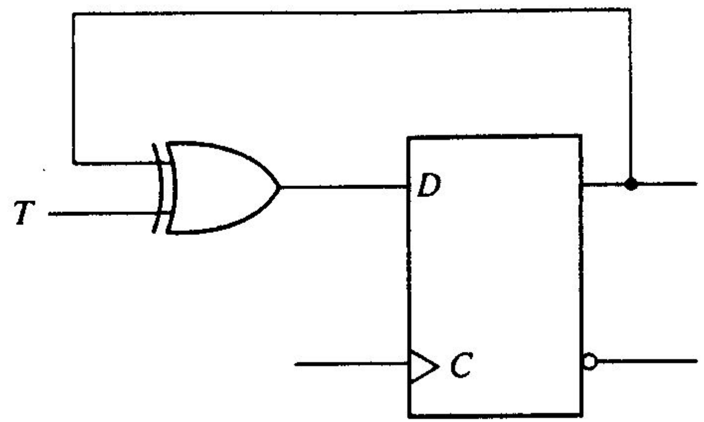
(a) Circuit diagram

(b) Graphic symbol

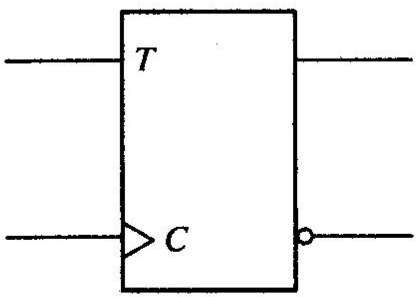
- $D = JQ' + K'Q$
  - J=0, K=0: D=Q, no change
  - J=0, K=1: D=0 $\Rightarrow$ Q =0
  - J=1, K=0: D=1 $\Rightarrow$ Q =1
  - J=1, K=1: D=Q' $\Rightarrow$ Q =Q'



(a) From *JK* flip-flop



(b) From *D* flip-flop



(c) Graphic symbol

- $D = T \oplus Q = TQ' + T'Q$
  - T=0: D=Q, no change
  - T=1: $D=Q' \Rightarrow Q=Q'$

- Characteristic tables

JK Flip-Flop

J	K	$Q(t+1)$	
0	0	$Q(t)$	No change
0	1	0	Reset
1	0	1	Set
1	1	$Q'(t)$	Complement

RS Flip-Flop

S	R	$Q(t+1)$	
0	0	$Q(t)$	No change
0	1	0	Reset
1	0	1	Set
1	1	?	Unpredictable

D Flip-Flop

D	$Q(t+1)$	
0	0	Reset
1	1	Set

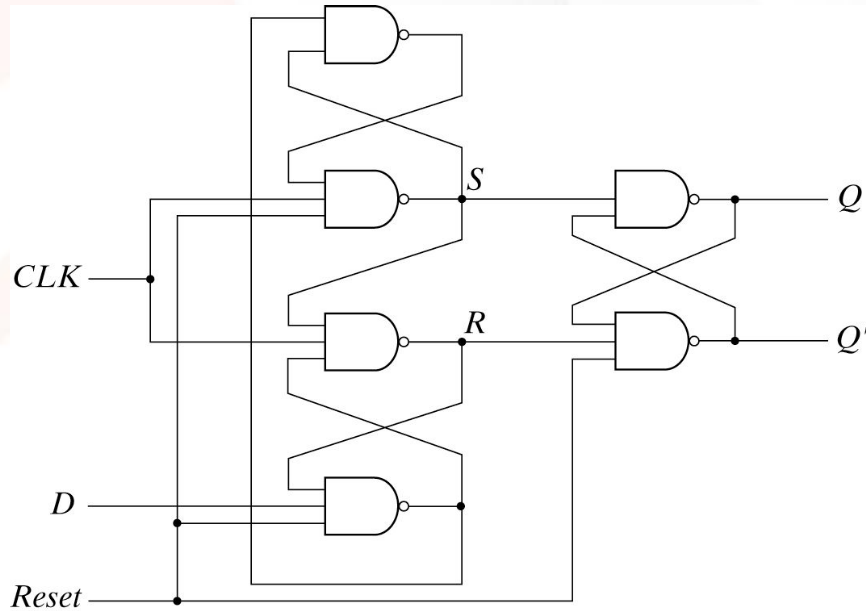
T Flip-Flop

T	$Q(t+1)$	
0	$Q(t)$	No change
1	$Q'(t)$	Complement

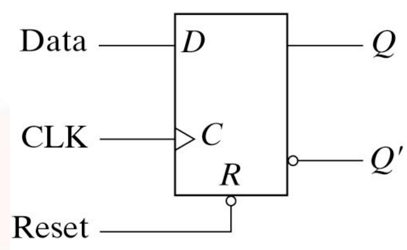
- Characteristic equations
  - D flip-flop
    - $Q(t+1) = D$
  - JK flip-flop
    - $Q(t+1) = JQ'+K'Q$
  - T flop-flop
    - $Q(t+1) = T \oplus Q$

Direct inputs

- asynchronous reset



(a) Circuit diagram



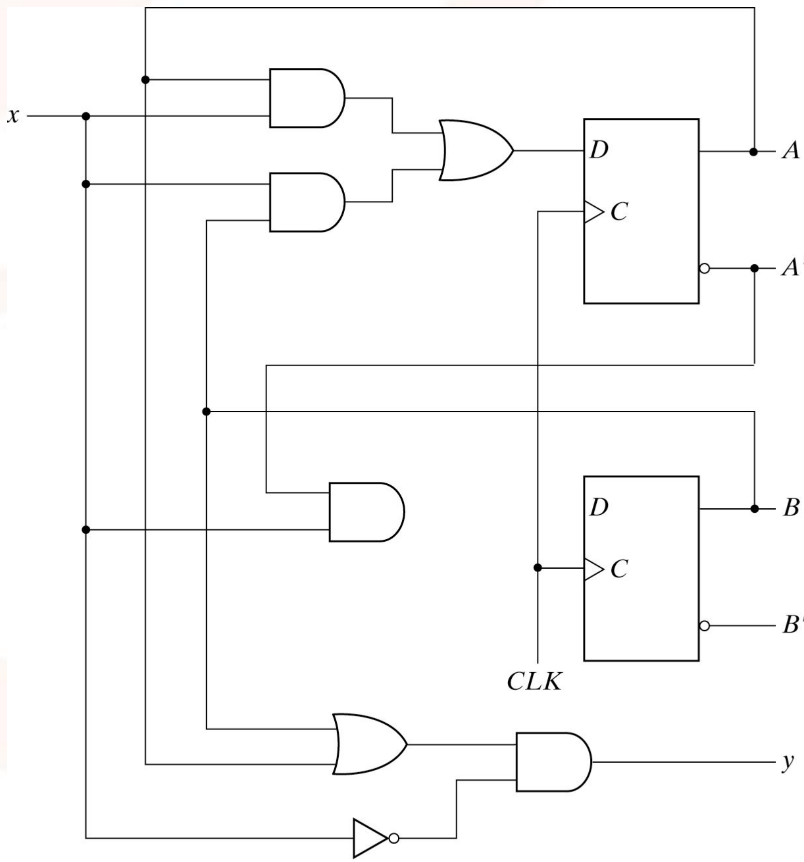
(b) Graphic symbol

R	C	D	Q	Q'
0	X	X	0	1
1	↑	0	0	1
1	↑	1	1	0

(b) Function table

Analysis of Clocked Sequential Circuits

- A sequential circuit
  - (inputs, current state) ⇒ (output, next state)
  - a state transition table or state transition diagram



State equations

- $A(t+1) = A(t)x(t) + B(t)x(t)$
- $B(t+1) = A'(t)x(t)$

- A compact form
  - $A(t+1) = Ax + Bx$
  - $B(t+1) = Ax$
- The output equation
  - $y(t) = (A(t)+B(t))x'(t)$
  - $y = (A+B)x'$

State table

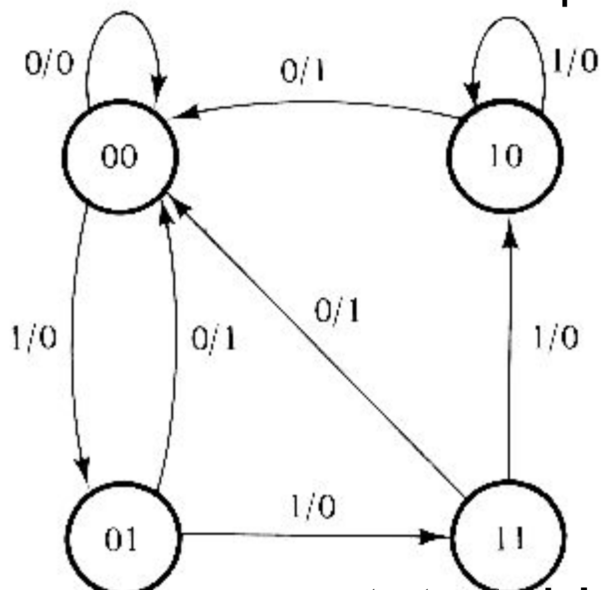
- State transition table

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

Present State	Next State		Output	
	$x = 0$	$x = 1$	$x = 0$	$x = 1$
AB	AB	AB	y	y
00	00	01	0	0
01	00	11	1	0
10	00	10	1	0
11	00	10	1	0

State diagram

- State transition diagram
  - a circle: a state
  - a directed lines connecting the circles: the transition between the states
    - Each directed line is labeled 'inputs/outputs'



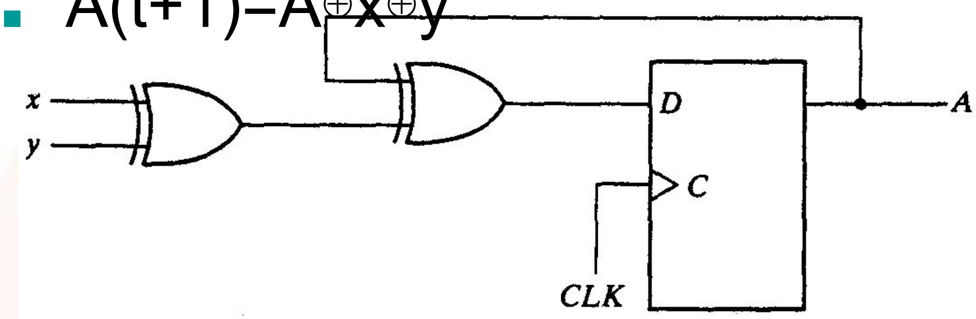
  - a logic diagram ⇔ a state table ⇔ a state diagram

Flip-flop input equations

- The part of circuit that generates the inputs to flip-flops
  - Also called excitation functions
  - DA = Ax +Bx
  - DB = A'x
- The output equations
  - to fully describe the sequential circuit
  - y = (A+B)x'

Analysis with D flip-flops

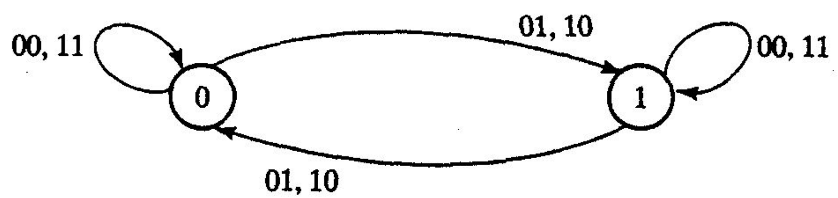
- The input equation
  - $D_A = A \oplus x \oplus y$
- The state equation
  - $A(t+1) = A \oplus x \oplus y$



(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

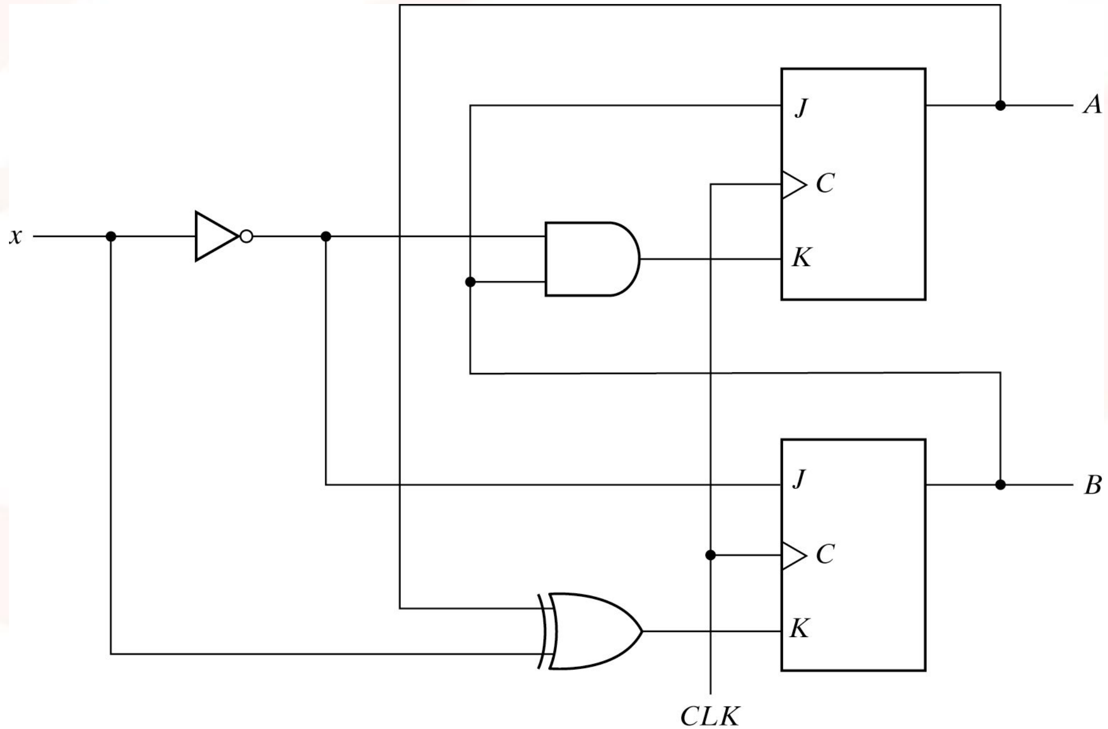
(b) State table



(c) State diagram

Analysis with JK flip-flops

- Determine the flip-flop input function in terms of the present state and input variables
- Used the corresponding flip-flop characteristic table to determine the next state

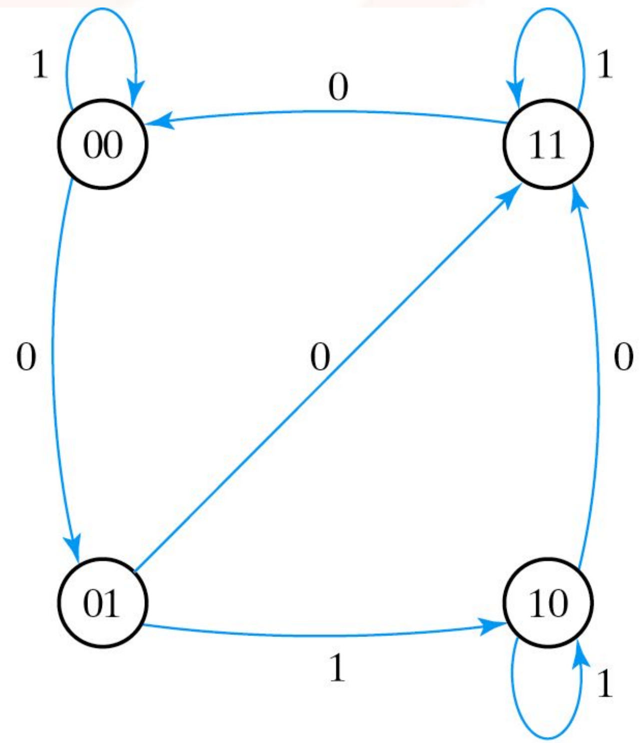


- JA = B, KA= Bx'
- JB = x', KB = A'x + Ax'
- derive the state table

Present state		Input	Next state		Flip-flop inputs			
A	*B*	*x*	*A*	*B*	*JA*	*KA*	*JB*	*KB*
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

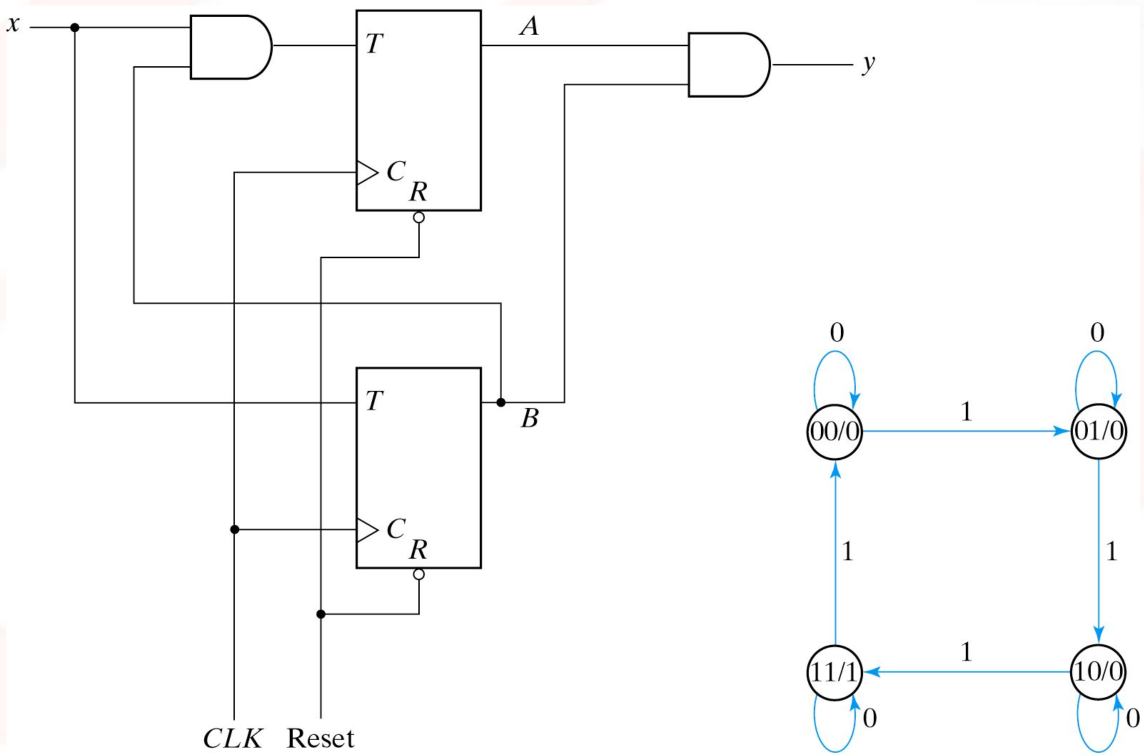
- Or, derive the state equations using characteristic eq.

- State transition diagram



Analysis with T flip-flops

- The characteristic equation
  - $Q(t+1) = T \oplus Q = TQ' + T'Q$



(a) Circuit diagram (b) State diagram

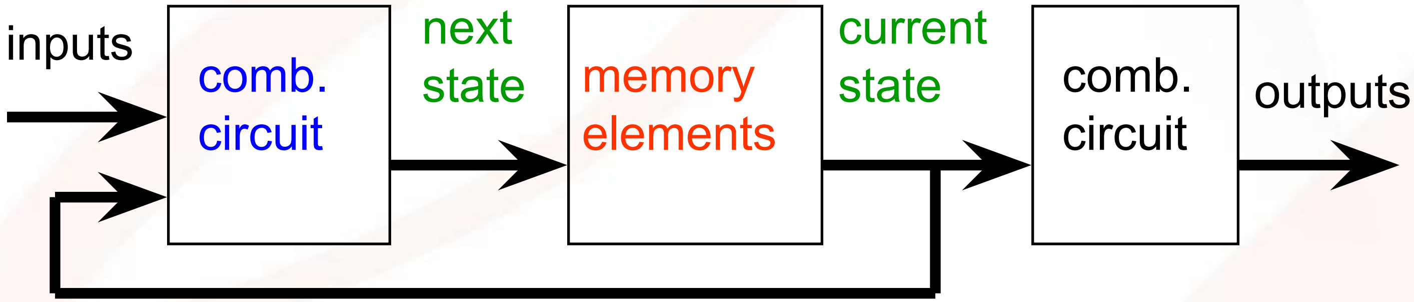
- The input and output functions
  - $T_A = Bx$
  - $T_B = x$
  - $y = AB$
- The state equations
  - $A(t+1) = (Bx)'A + (Bx)A' = AB' + Ax' + A'Bx$
  - $B(t+1) = x \oplus B$

Mealy and Moore models

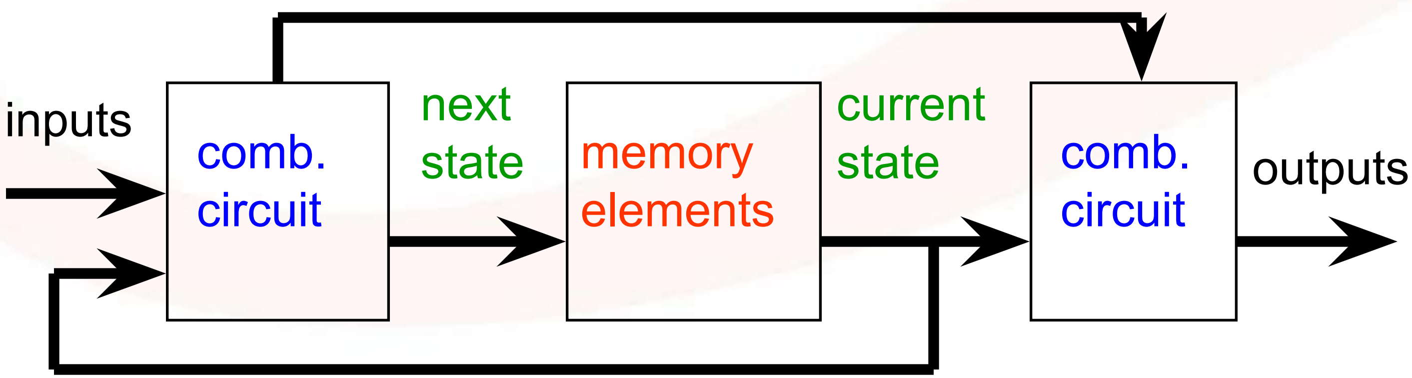
- the Mealy model: the outputs are functions of both the present state and inputs
  - the outputs may change if the inputs change during the clock pulse period
    - the outputs may have momentary false values unless the inputs are synchronized with the clocks
- The Moore model: the outputs are functions of the present state only
  - The outputs are synchronous with the clocks

Mealy and Moore Models

- Moore machine

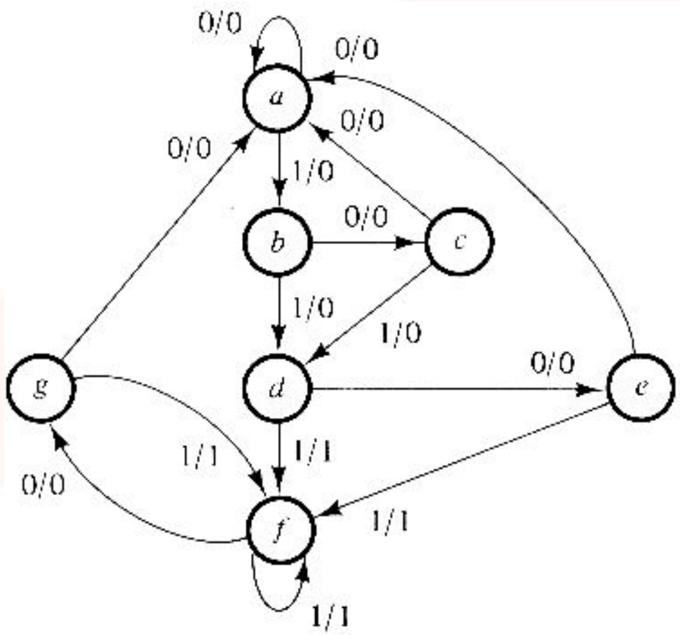


- Mealy machine

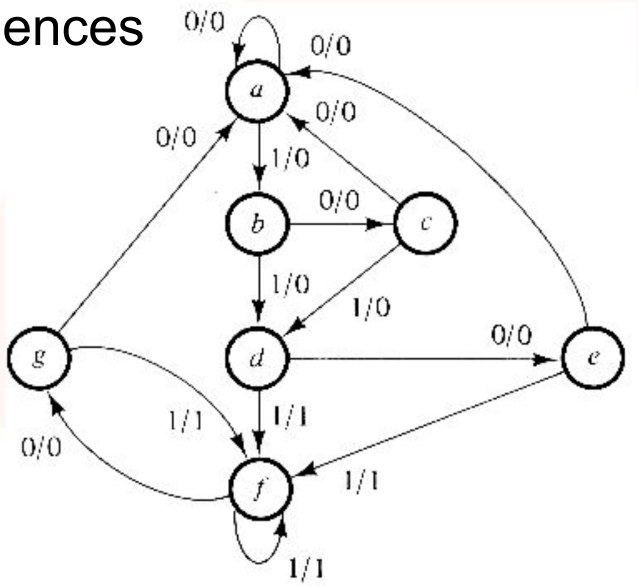


State Reduction and Assignment

- State Reduction
  - reductions on the number of flip-flops and the number of gates
  - a reduction in the number of states may result in a reduction in the number of flip-flops
  - a example state diagram



- state a a b c d e f f g f g a
  input 0 1 0 1 0 1 1 0 1 0 0
  output 0 0 0 0 0 1 1 0 1 0 0
- only the input-output sequences are important
- two circuits are equivalent
  - have identical outputs for all input sequences
  - the number of states is not important



- Equivalent states
  - two states are said to be equivalent
    - for each member of the set of inputs, they give exactly the same output and send the circuit to the same state or to an equivalent state
    - (

Present State	Next State $x = 0$	Next State $x = 1$	Output $x = 0$	Output $x = 1$
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	f	0	1
e	a	f	0	1
f	g	f	0	1
g	a	f	0	1

- Reducing the state table
  - e=f
  - d=?

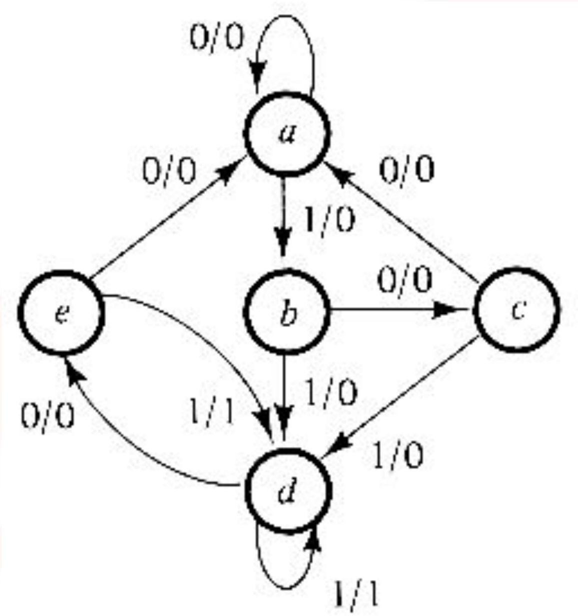
Present State	Next State		Output	
	$x = 0$	$x = 1$	$x = 0$	$x = 1$
a	*a*	*b*	0	0
b	*c*	*d*	0	0
c	*a*	*d*	0	0
d	*e*	*f*	0	1
e	*a*	*f*	0	1
f	*g*	*f*	0	1
g	*a*	*f*	0	1

Present State	Next State		Output	
	$x = 0$	$x = 1$	$x = 0$	$x = 1$
a	*a*	*b*	0	0
b	*c*	*d*	0	0
c	*a*	*d*	0	0
d	*e*	*f*	0	1
e	*a*	*f*	0	1
f	*e*	*f*	0	1

- the reduced finite state machine

Present State	Next state $x = 0$	Next state $x = 1$	Output $x = 0$	Output $x = 1$
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	d	0	1
e	a	d	0	1

- state a a b c d e d d e d e a
  input 0 1 0 1 0 1 1 0 1 0 0
  output 0 0 0 0 0 1 1 0 1 0 0



- the checking of each pair of states for possible equivalence can be done systematically (9-5)
- the unused states are treated as don't-care condition ⇒ fewer combinational gates

State assignment

- to minimize the cost of the combinational circuits
- three possible binary state assignments

State	Assignment 1 Binary	Assignment 2 Gray code	Assignment 3 One-hot
a	000	000	00001
b	001	001	00010
c	010	011	00100
d	011	010	01000
e	100	110	10000

- any binary number assignment is satisfactory as long as each state is assigned a unique number
- use binary assignment 1

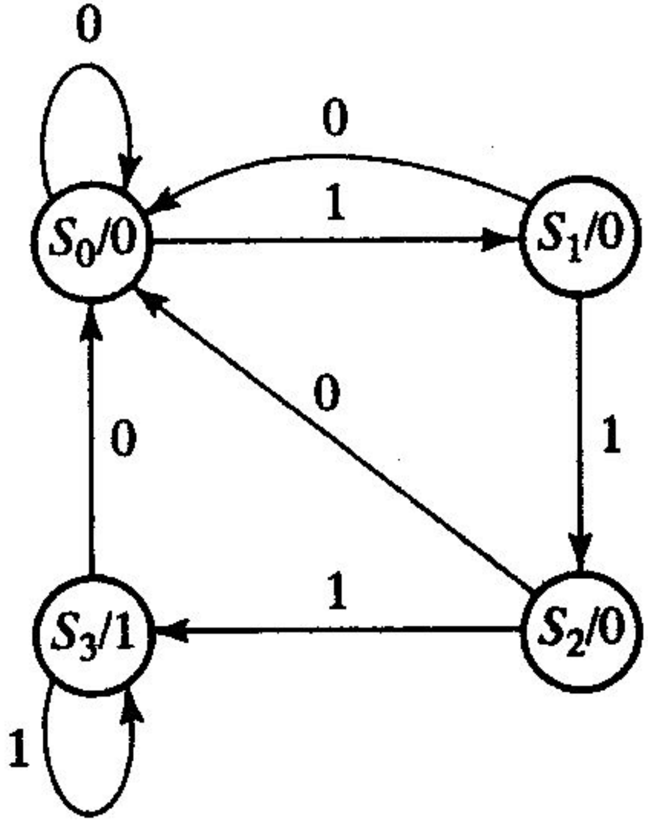
	Next State		Output	
Present state	$x = 0$	$x = 1$	$x = 0$	$x = 1$
001	001	010	0	0
010	011	100	0	0
011	001	100	0	0
100	101	100	0	1
101	001	100	0	1

Design Procedure

- the word description of the circuit behavior (a state diagram)
- state reduction if necessary
- assign binary values to the states
- obtain the binary-coded state table
- choose the type of flip-flops
- derive the simplified flip-flop input equations and output equations
- draw the logic diagram

Synthesis using D flip-flops

- An example state diagram and state table

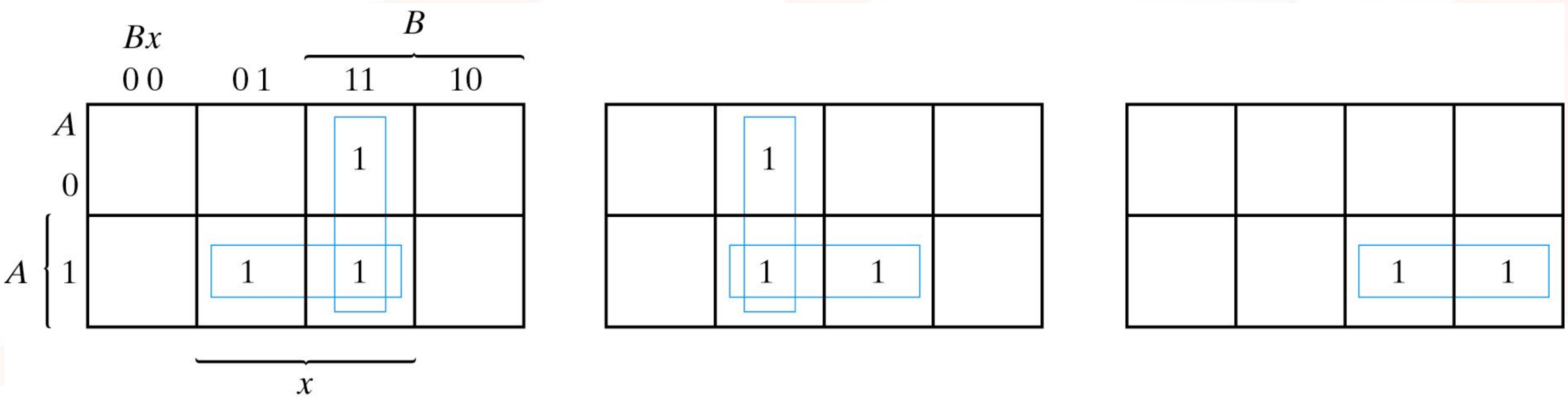


State Table for Sequence Detector

Present State A	Present State B	Input x	Next State A	Next State B	Output 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

- The flip-flop input equations
  - $A(t+1) = D_A(A,B,x) = \Sigma(3,5,7)$
  - $B(t+1) = D_B(A,B,x) = \Sigma(1,5,7)$
- The output equation
  - $y(A,B,x) = \Sigma(6,7)$
- Logic minimization using the K map
  - $D_A = Ax + Bx$
  - $D_B = Ax + B'x$
  - $y = AB$

Maps for Sequence Detector

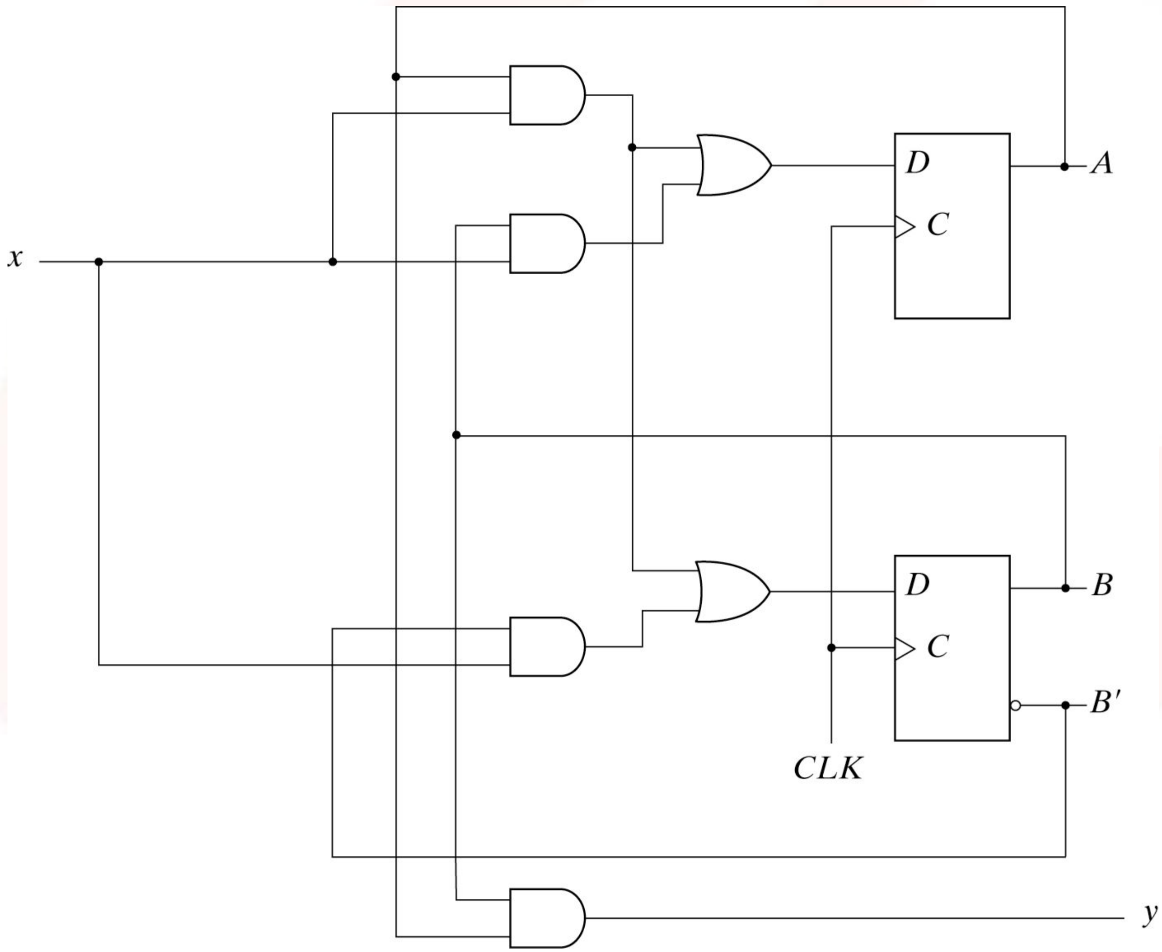


$D_A = Ax + Bx$

$D_B = Ax + B'x$

$y = AB$

- The logic diagram



Excitation tables

- A state diagram $\Rightarrow$ flip-flop input functions
  - straightforward for D flip-flops
  - we need excitation tables for JK and T flip-flops

Flip-Flop Excitation Tables

$Q(t)$	$Q(t+1)$	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

(a) JK

$Q(t)$	$Q(t+1)$	T
0	0	0
0	1	1
1	0	1
1	1	0

(b) T

Synthesis using JK flip-flops

- The same example
- The state table and JK flip-flop inputs

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	0	0	X	0	X
0	0	1	0	1	0	X	1	X
0	1	0	1	0	1	X	X	1
0	1	1	0	1	0	X	X	0
1	0	0	1	0	X	0	0	X
1	0	1	1	1	X	0	1	X
1	1	0	1	1	X	0	X	0
1	1	1	0	0	X	1	X	1

- $J_A = Bx'$; $K_A = Bx$
- $J_B = x$; $K_B = (A \oplus x)'$
- $y = ?$

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

$J_A = Bx'$

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

$K_A = Bx$

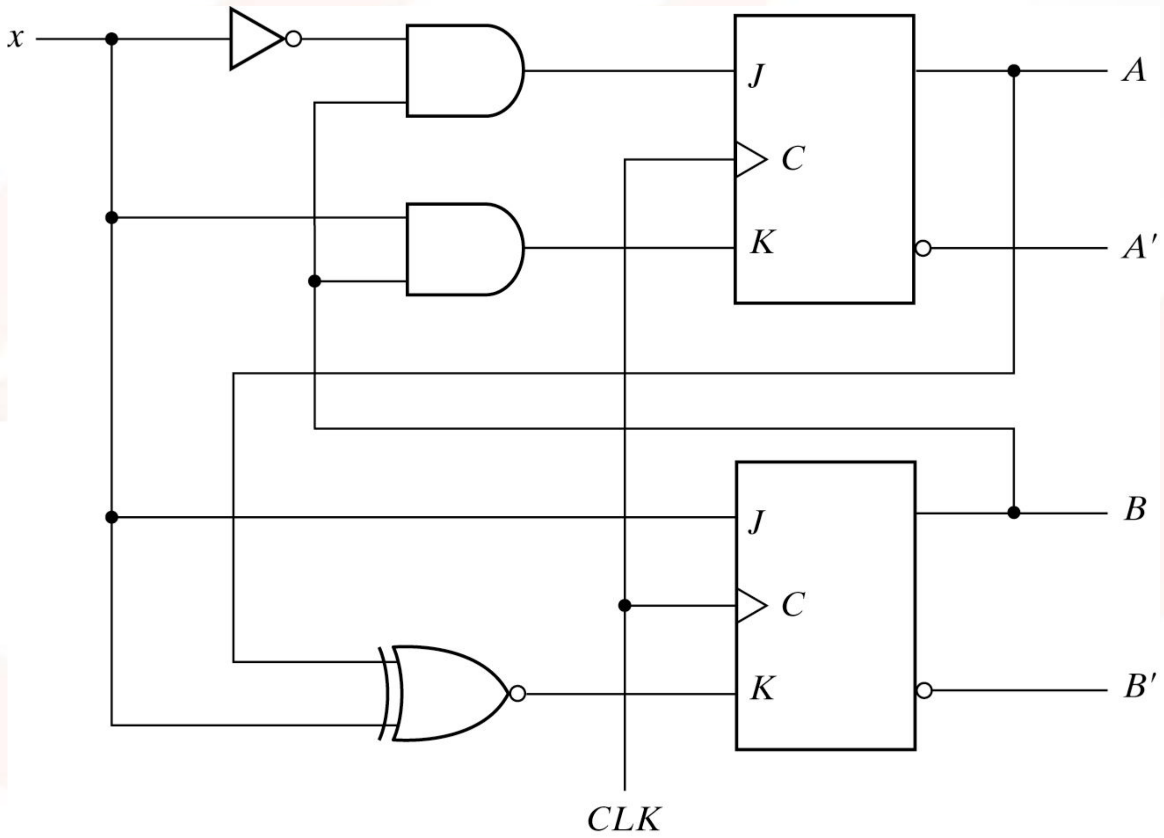
A \ Bx	00	01	11	10
0		1	X	X
1		1	X	X

$J_B = x$

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

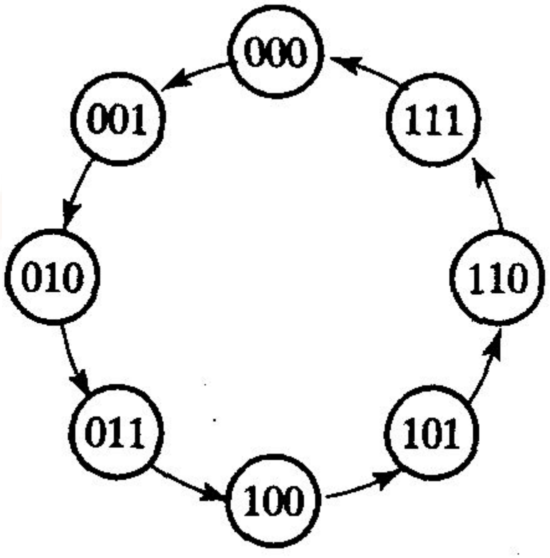
$K_B = (A \oplus x)'$

Logic diagram for sequential circuit with J-K flip-flops



Synthesis using T flip-flops

- A n-bit binary counter
  - the state diagram

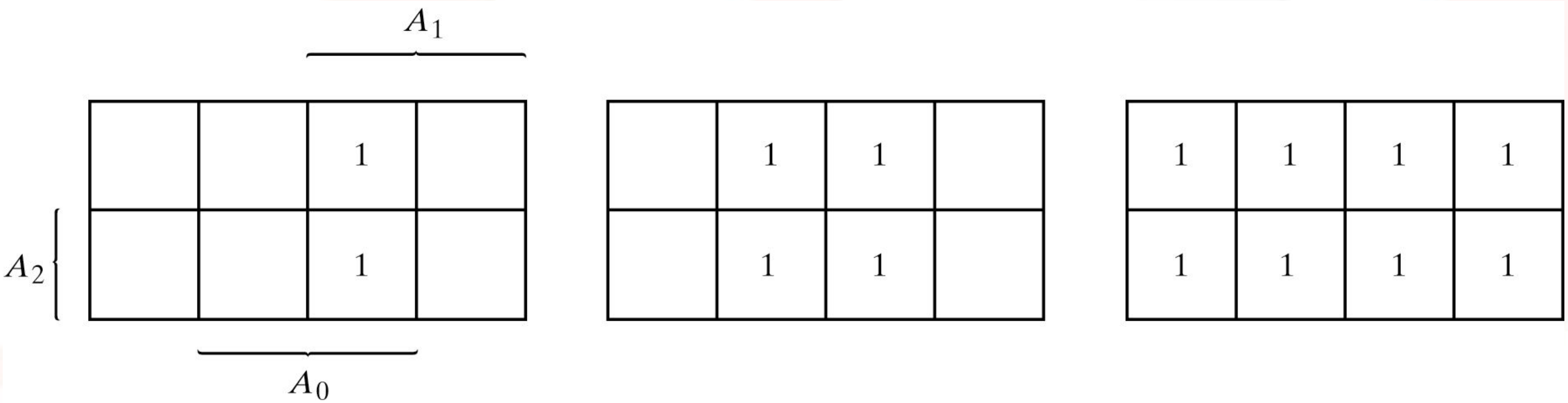


  - no inputs (except for the clock input)

- The state table and the flip-flop inputs

Present State			Next State			Flip-Flop Inputs		
A_2	A_1	A_0	A_2	A	A_0	T_{A2}	T_{A1}	T_{A0}
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	1	1	1
1	0	0	1	0	1	0	0	1
1	0	1	1	1	0	0	1	1
1	1	0	1	1	1	0	1	1
1	1	1	0	0	0	1	1	1

Maps for 3-bit Binary Counter



$T_{A2} = A_1 A_0$ $\quad$ $T_{A1} = A_0$ $\quad$ $T_{A0} = 1$

- Logic simplification using the K map
  - $T_{A2} = A_1A_2$
  - $T_{A1} = A_0$
  - $T_{A0} = 1$
- The logic diagram