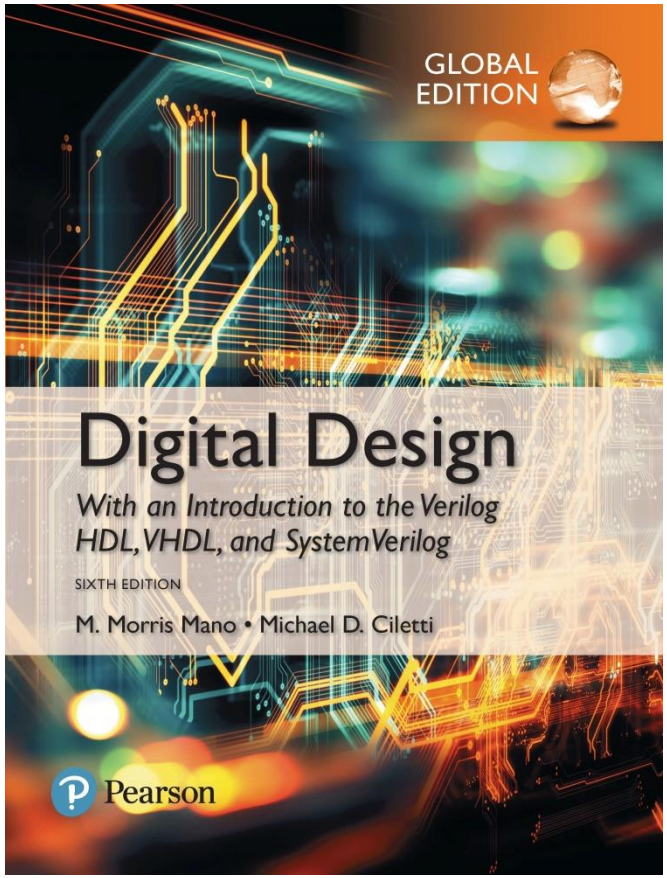
Digital Design

With an Introduction to the Verilog HDL, VHDL, and SystemVerilog

6th Edition, Global Edition

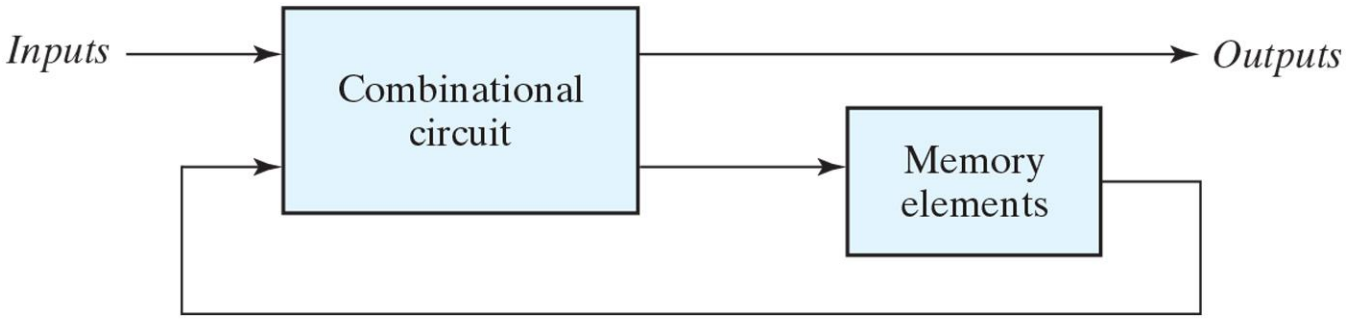


Chapter 05

Synchronous Sequential Logic

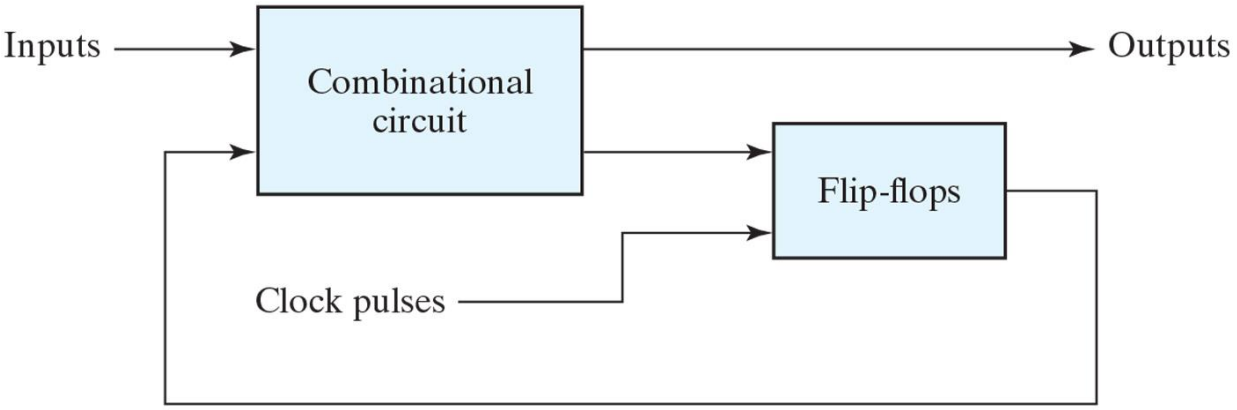
Pearson

Block diagram of sequential circuit

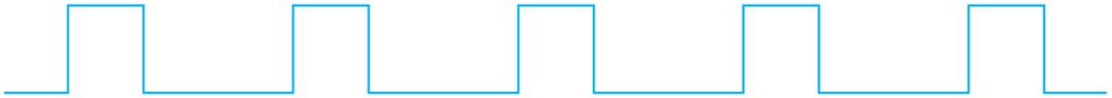


- Outputs are function of inputs and present states
- Present states are supplied by memory elements

Synchronous clocked sequential circuit



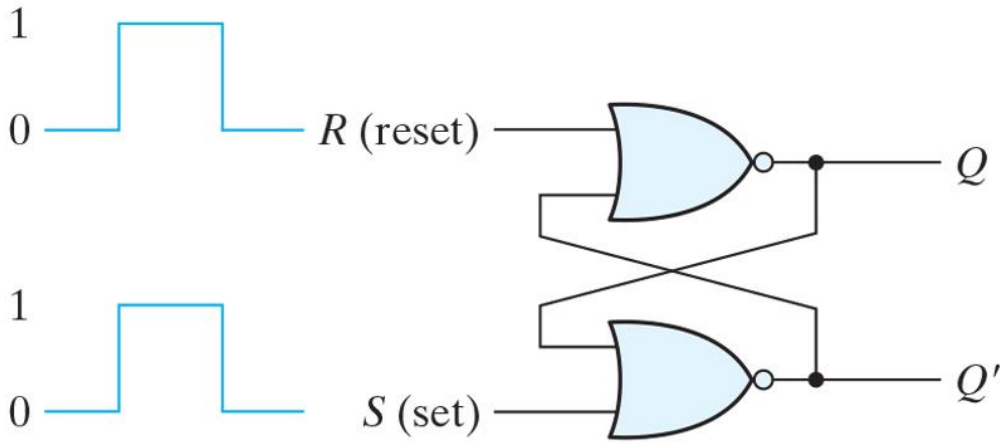
(a) Block diagram



(b) Timing diagram of clock pulses

- Two types of sequential logic
  - Synchronous : behavior depends on the signals affecting storage elements at discrete time
  - Asynchronous : behavior depends on inputs at any instance of time

SR latch with NOR gates



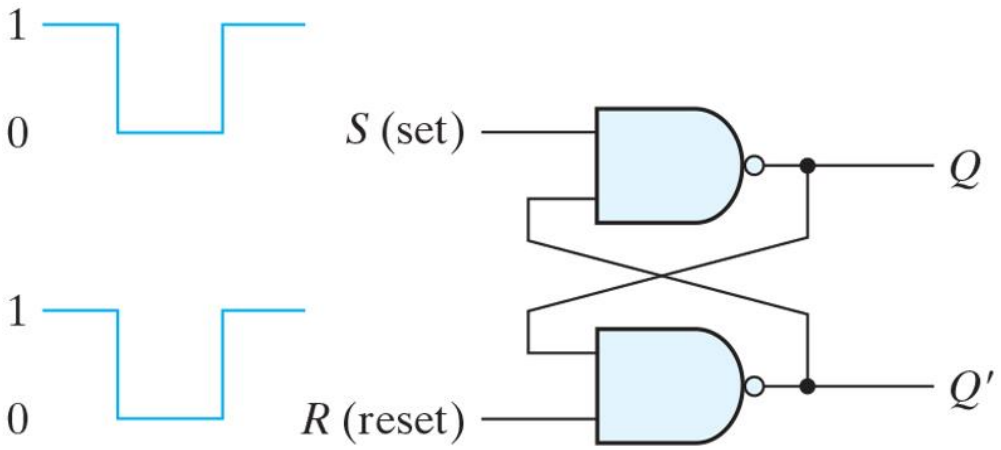
(a) Logic diagram

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	(forbidden)

(b) Function table

- S=1,R=0 then Q=1(set)
- S=0,R=1 then Q=0(reset)
- S=0,R=0 then no change(keep condition)
- S=1,R=1 Q=Q′=0 (undefined)

SR latch with NAND gates

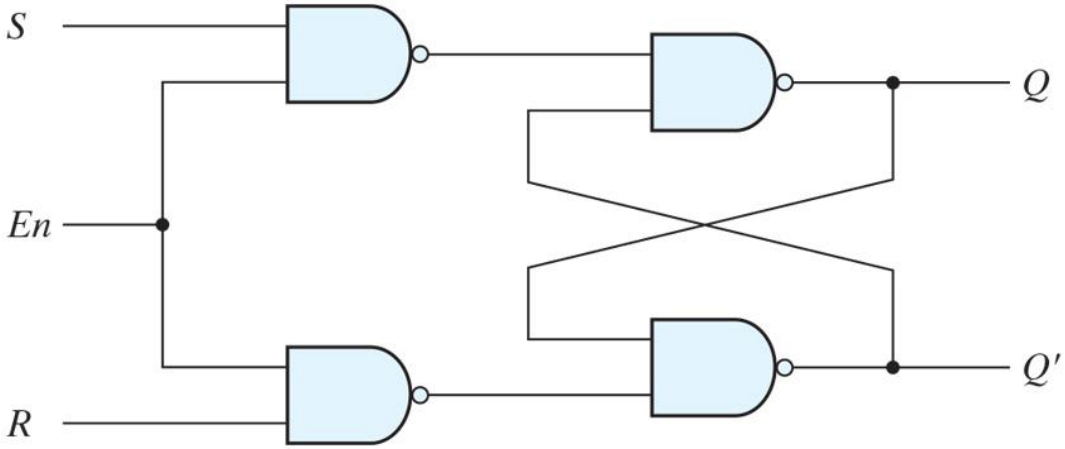


(a) Logic diagram

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	(forbidden)

(b) Function table

SR latch with control input

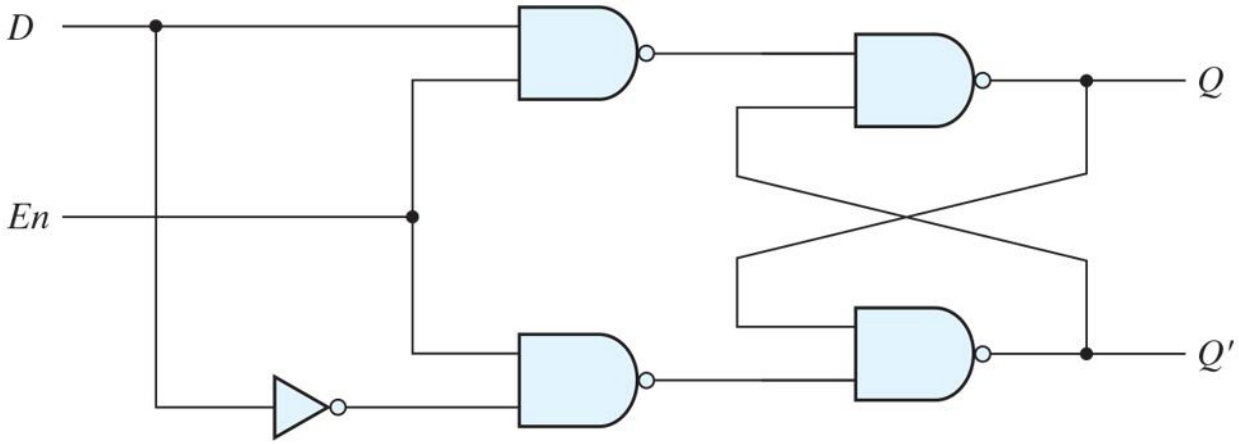


(a) Logic diagram

En	*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

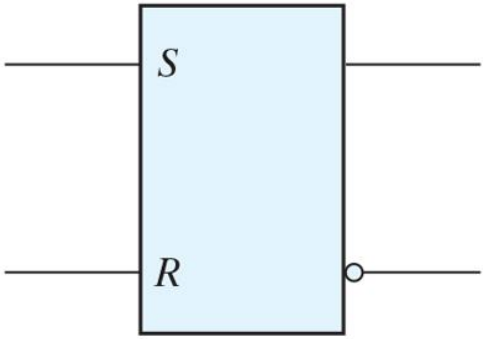


(a) Logic diagram

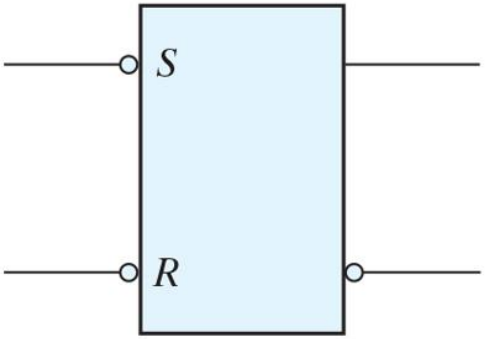
En	*D*	Next state of *Q*
0	X	No change
1	0	$Q = 0$; reset state
1	1	$Q = 1$; set state

(b) Function table

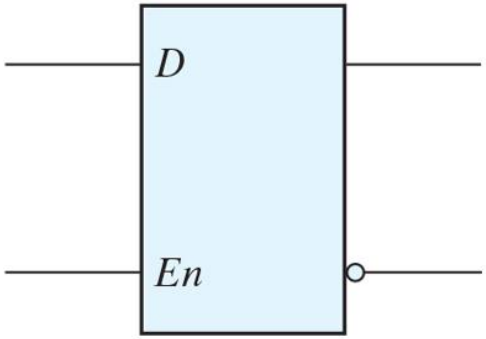
Graphic symbols for latches



SR

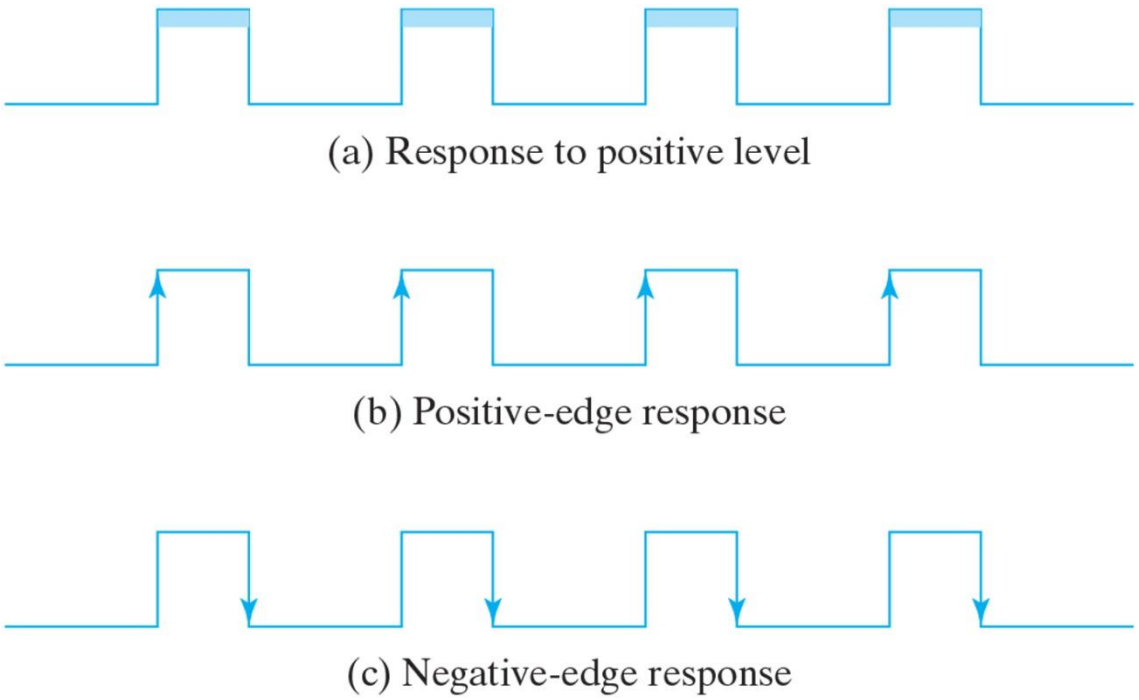


$\overline{S}\overline{R}$



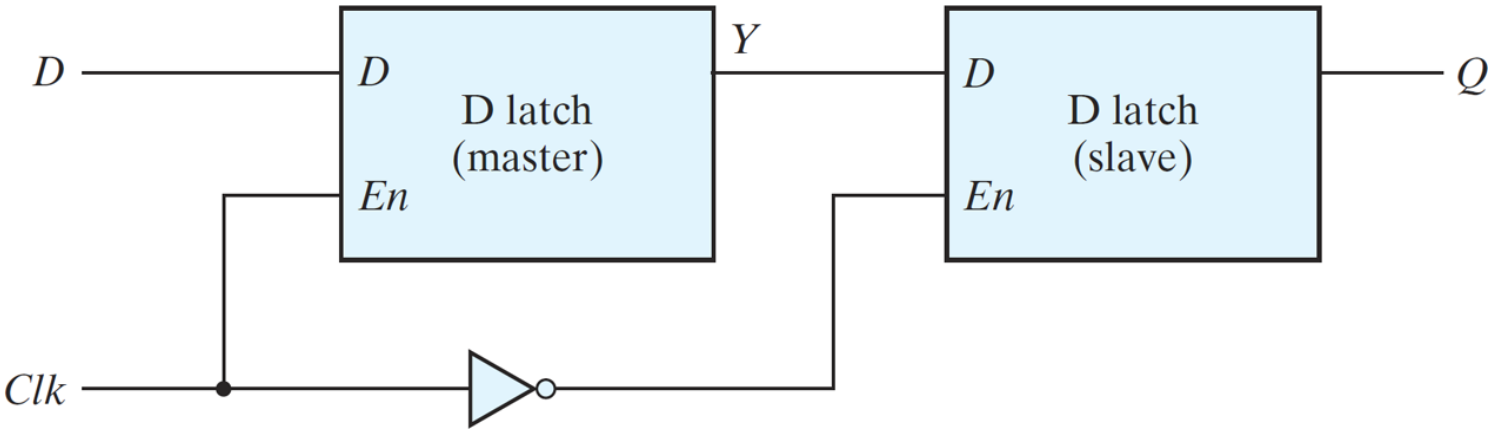
D

Clock response in latch and flip-flop



- Latch : case (a), output changes as input changes
- Flip-flop : output only changes at clock edge

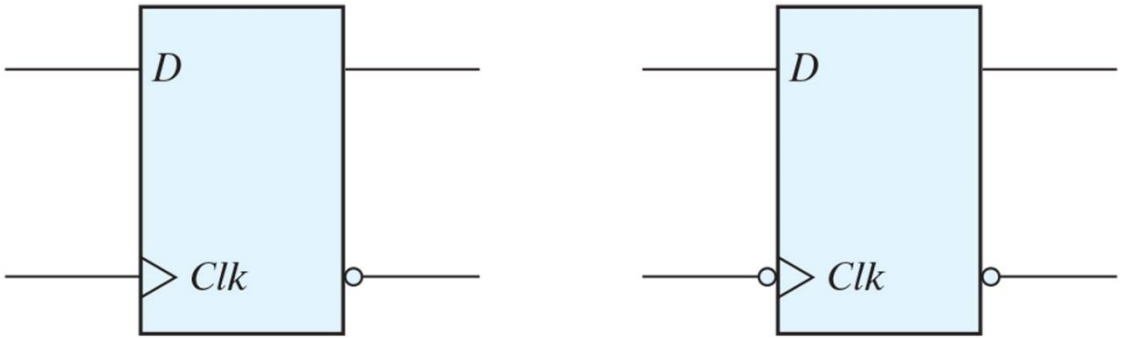
Master–slave *D* flip-flop



- Negative edge triggered D flip-flop

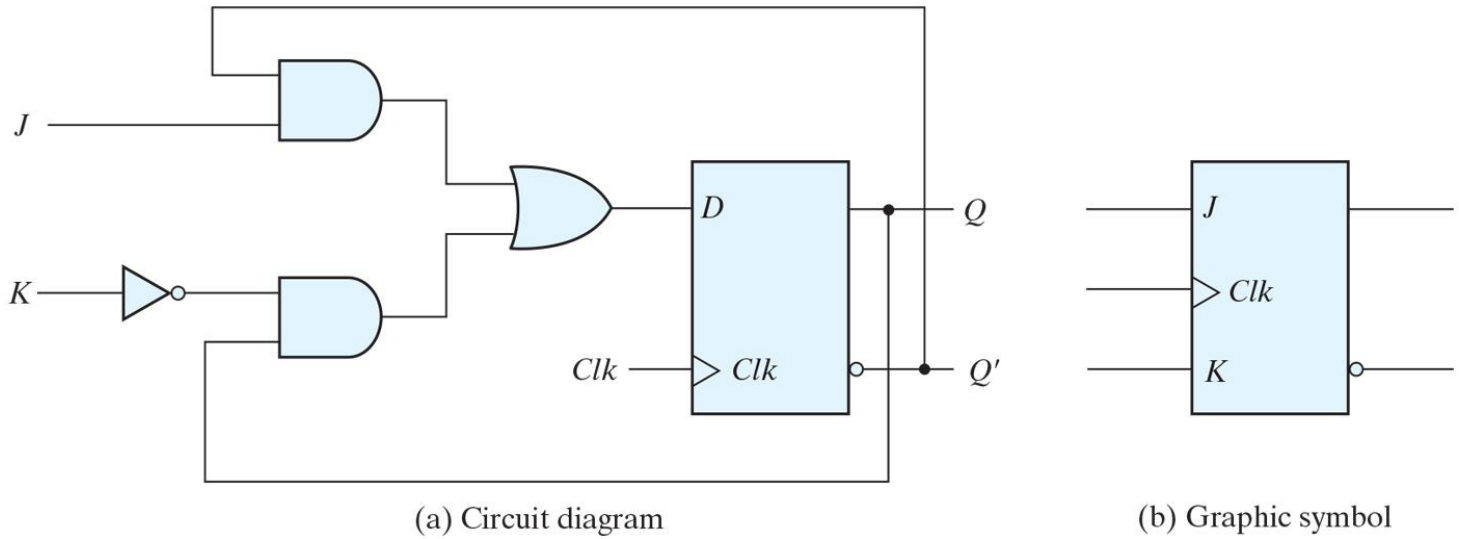
- *Clk*=0 : master disable, slave enable
  - Output has no relation with input
- *Clk*=1 : master enable, slave disable

Graphic symbol for edge-triggered *D* flip-flop



(a) Positive-edge (b) Negative-edge

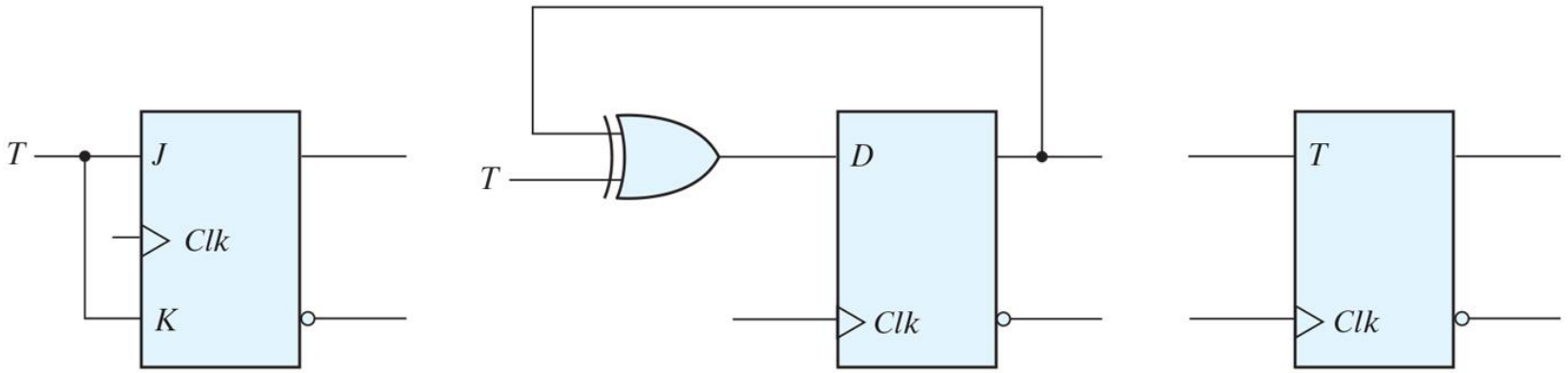
JK flip-flop



(a) Circuit diagram (b) Graphic symbol

- Set(J), Reset(K), Complement(J=K=1)
- D=JQ′+K′Q

T flip-flop



(a) From *JK* flip-flop (b) From *D* flip-flop (c) Graphic symbol

- Complementing flip-flop
- D=TQ′+T′Q

Flip-Flop 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

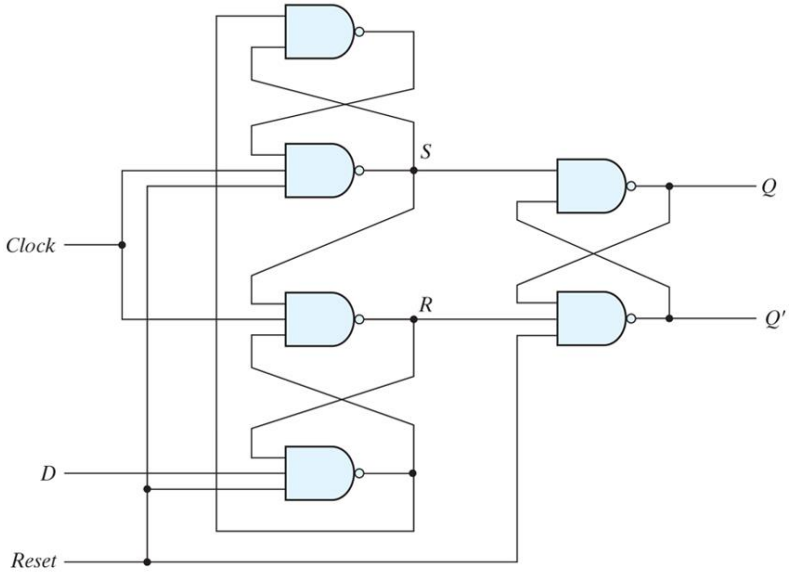
D Flip-Flop

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

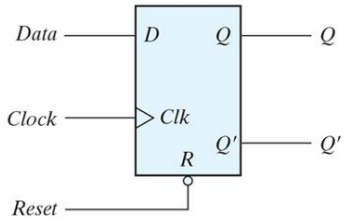
T Flip-Flop

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

D flip-flop with asynchronous reset



(a) Circuit diagram

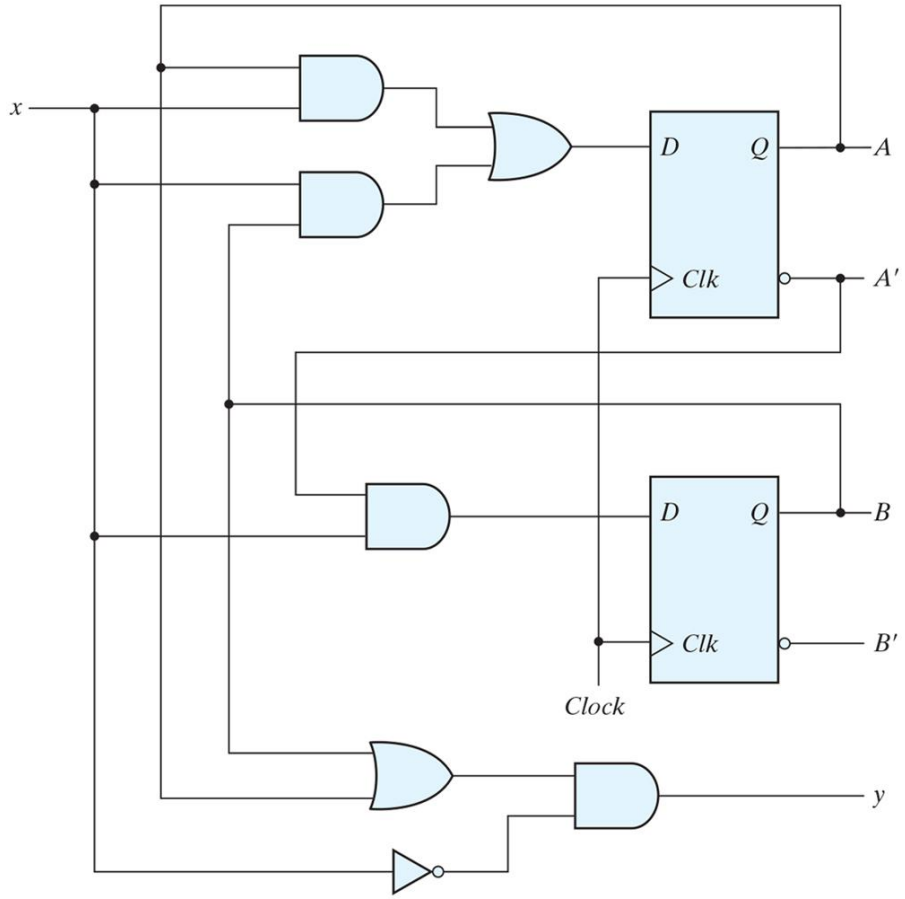


(b) Graphic symbol

R	Clk	D	Q	Q′
0	X	X	0	1
1	↑	0	0	1
1	↑	1	1	0

(c) Function table

Example of sequential circuit



Analysis of clocked sequential circuit

- Behavior of clocked sequential circuit is determined from input, output and present state
- Outputs and next states are a function of inputs and present states

State Equations

- Specifies the next state and output as a function of the present state and inputs
- A(t+1)=A*x* + B*x*
- B(t+1)=A′*x*
- *Y*=(A+B)*x*′

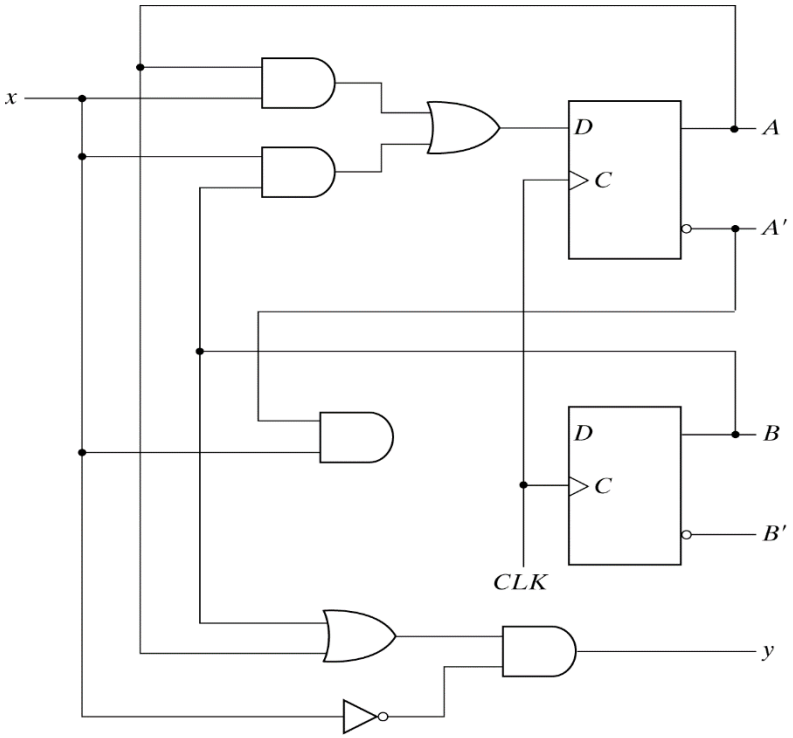


Fig. 5-15 Example of Sequential Circuit

State Table for the Circuit of Fig. 5.15

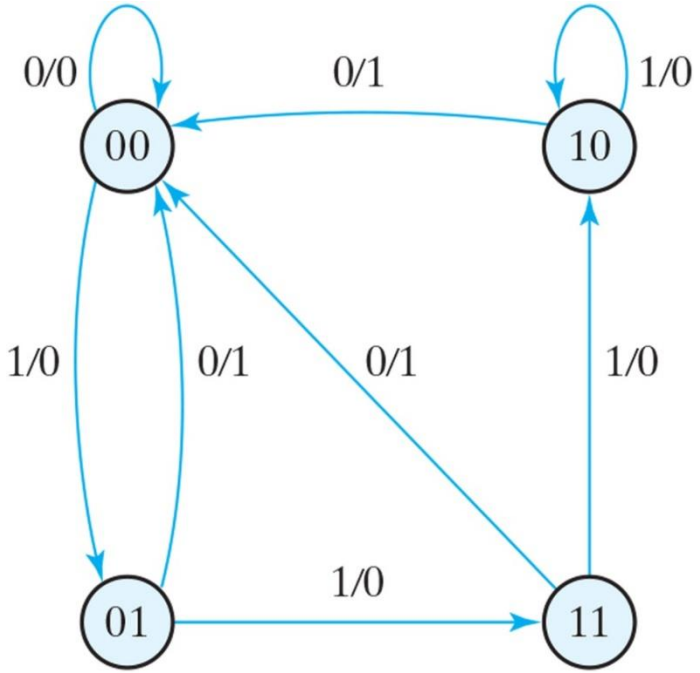
Present State		Input	Next State		Output
A	*B*	*x*	*A*	*B*	*y*
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	1	0	0
1	1	0	0	0	1
1	1	1	1	0	0

Second Form of the State Table

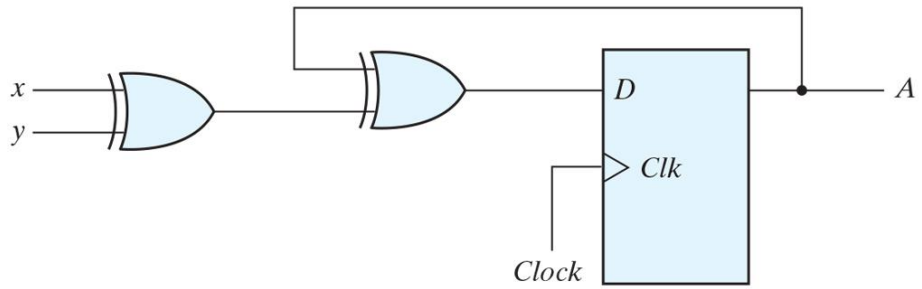
Present State		Next State				Output	
		$x = 0$		$x = 1$		$x = 0$	$x = 1$
A	B	A	B	A	B	y	y
0	0	0	0	0	1	0	0
0	1	0	0	1	1	1	0
1	0	0	0	1	0	1	0
1	1	0	0	1	0	1	0

State diagram of 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



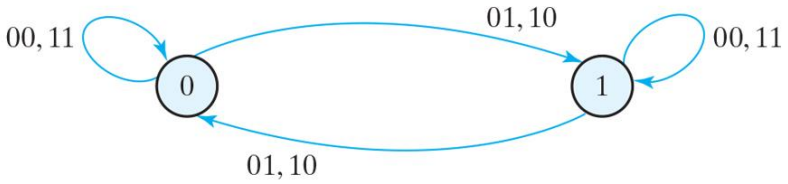
Sequential circuit with *D* flip-flop



(a) Circuit diagram

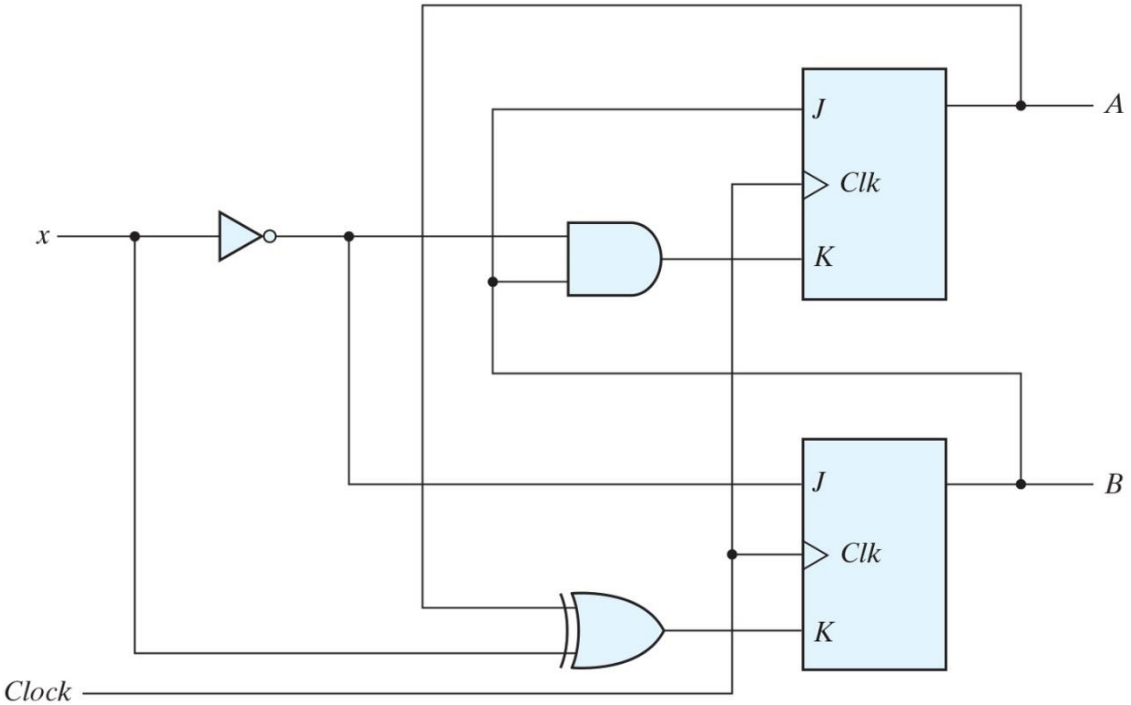
Present state	Inputs		Next state
A	*x*	*y*	*A*
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

(b) State table



(c) State diagram

Sequential circuit with *JK* flip-flop

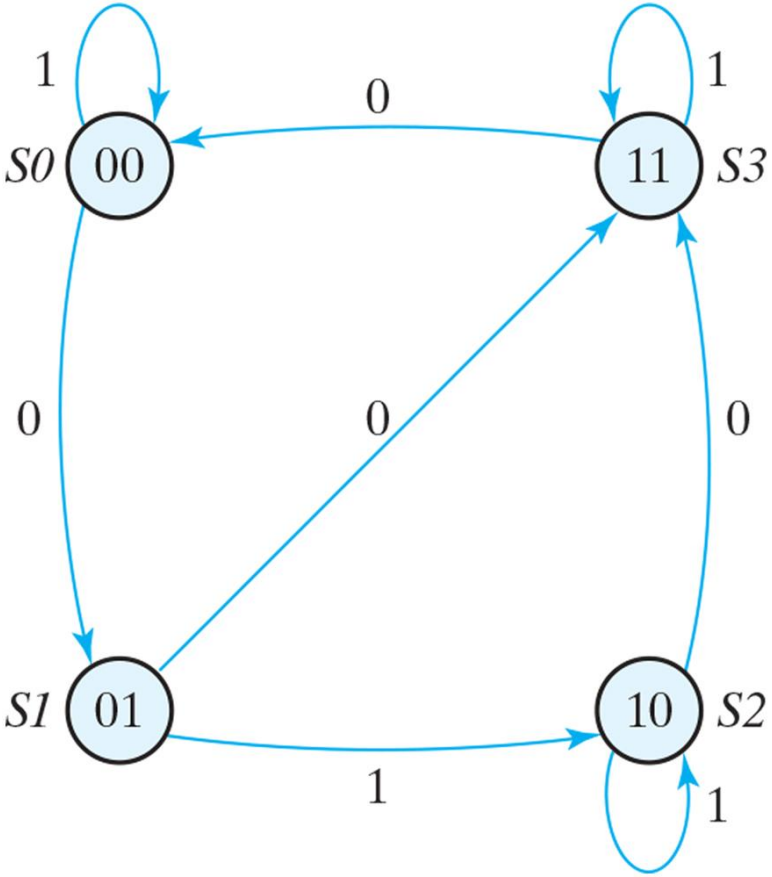


- State equation is not the same as the input equation
- Characteristic table or characteristic equation is needed
- Input equations
  - $J_A=B \quad K_A=Bx'$
  - $J_B=x' \quad K_B=A'x+Ax'$

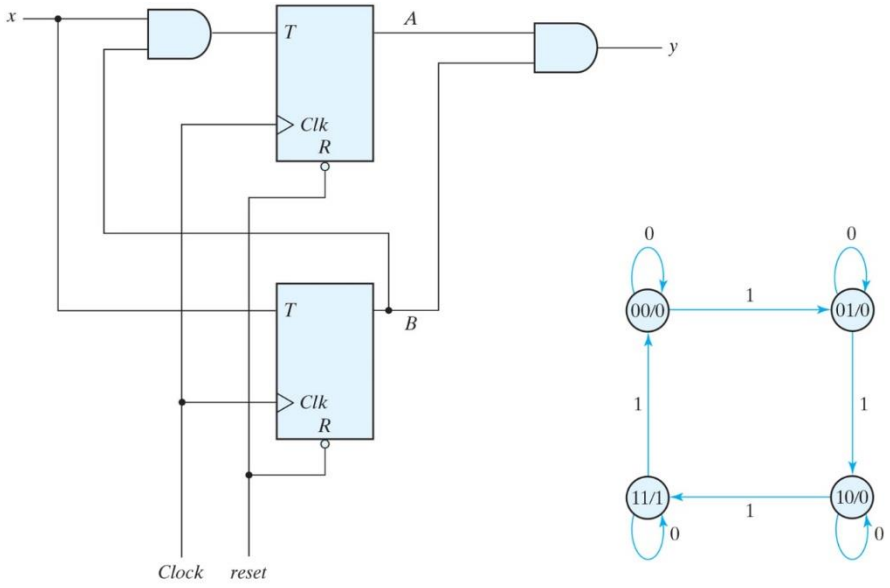
State Table for Sequential Circuit with JK Flip-Flops

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

State diagram of the circuit of Fig. 5.18



Sequential circuit with T flip-flops (Binary Counter)



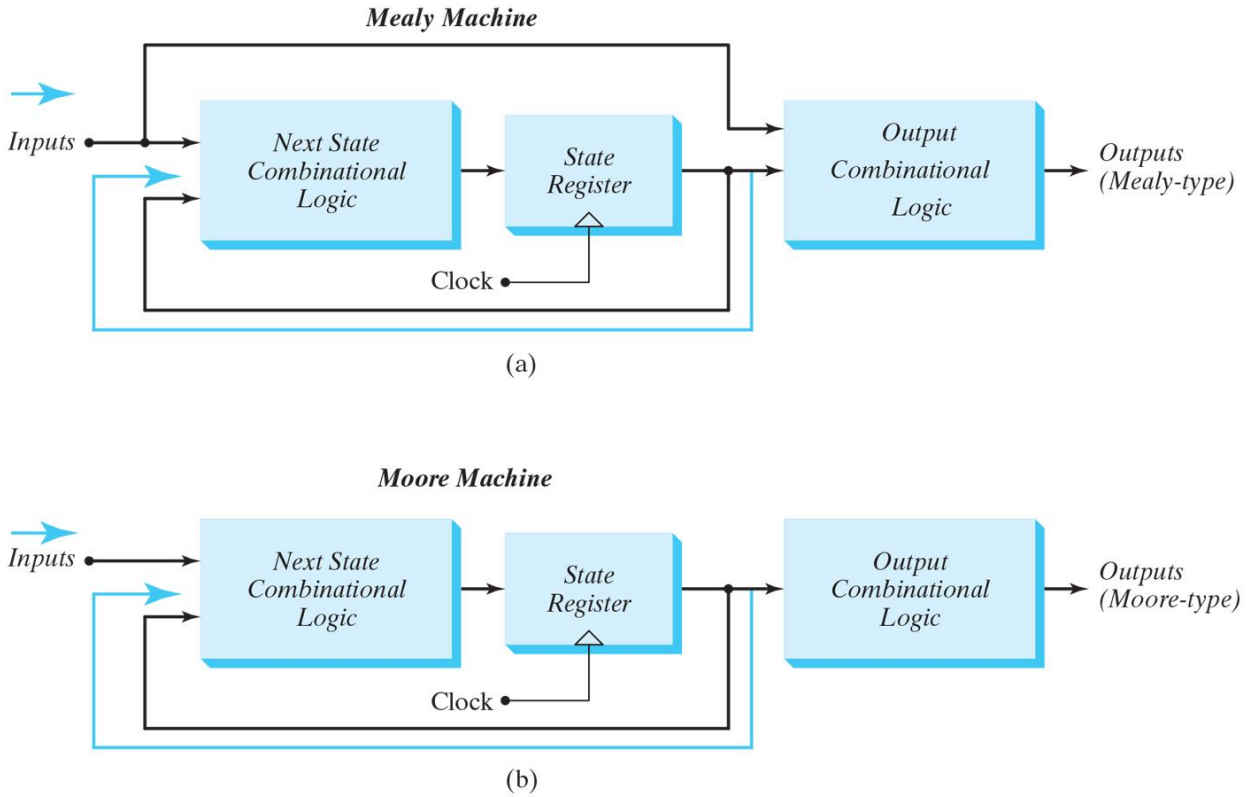
(a) Circuit diagram (b) State diagram

- Input equations and output equation
  - $T_A=Bx$, $T_B=x$
  - $y=AB$
- State equations are derived from characteristic equation
  - $A(t+1)=T_AA'+T_A'A$
  - $B(t+1)=T_BB'+T_B'B$

State Table for Sequential Circuit with T Flip-Flops

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	1	0
0	1	1	1	0	0
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	1

Block diagrams of Mealy and Moore state machines

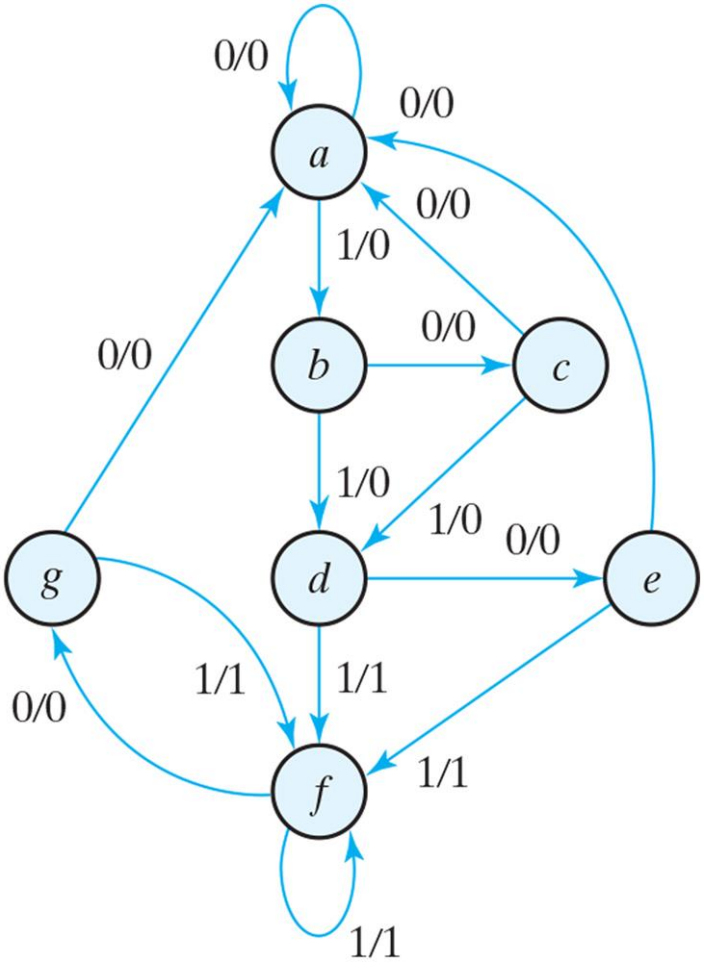


- Mealy model : output is a function of the present state and input
  - Inputs must be synchronized with the clock
  - Outputs must be sampled at the clock edge
- Moore model : output is a function of the present state only

State Reduction

- State reduction is used to reduce the number of flip-flop and gates
- Only input/output sequences are important
- Interested in present states that go to the same next state and have the same output

State diagram



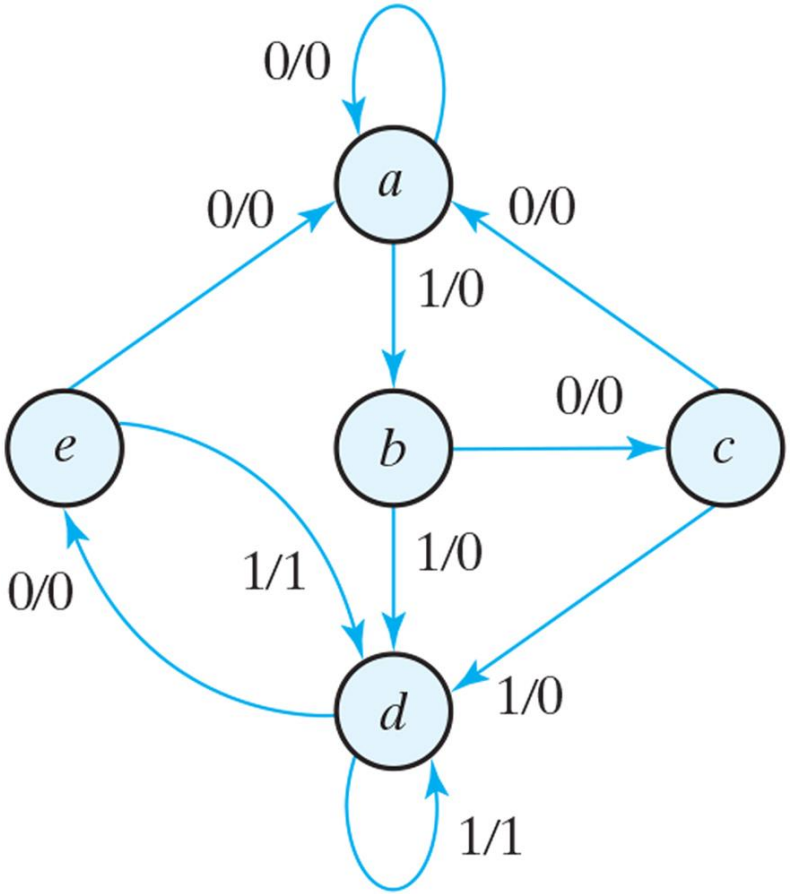
State Table

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

Reducing the State Table

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

Reduced State diagram



Reduced the State Table

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	d	0	1
e	a	d	0	1

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

- *m* states circuit, codes must contain *n* bits where $2^n \geq m$
- Three possible binary state assignments

Reduced State Table with Binary Assignment 1

Present State	Next State		Output	
	$x = 0$	$x = 1$	$x = 0$	$x = 1$
000	000	001	0	0
001	010	011	0	0
010	000	011	0	0
011	100	011	0	1
100	000	011	0	1

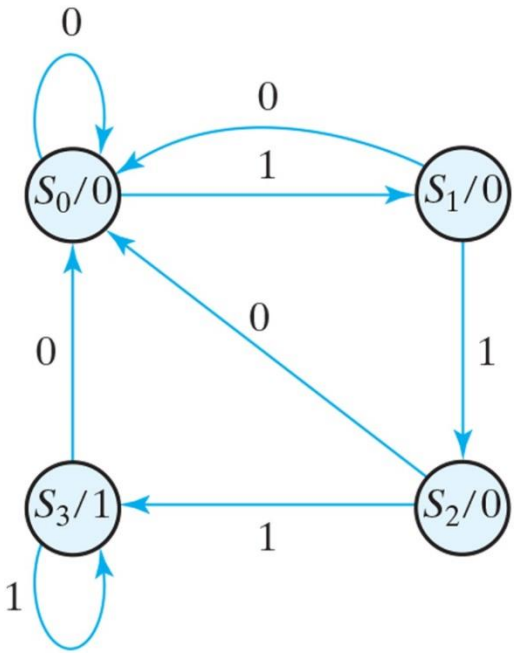
Design Procedure

- Sequential circuit design : requires state table
  ⇔ Combinational circuit : truth table
- The number of flip-flop is determined from the number of states
  - If 2^n states exist, there are *n* flip-flops

- Design steps
  1) Derive a state diagram or state table
  2) Reduce the number of states if necessary
  3) Assign binary code to the state
  4) Choose the type of flip-flops to be used
  5) Derive the flip-flop input equations and output equations
  6) Draw the logic diagram

State diagram for sequence detector

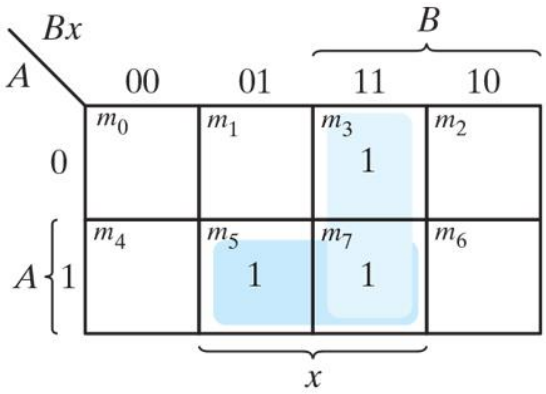
Three or more consecutive 1's in a string of bits coming through an input line



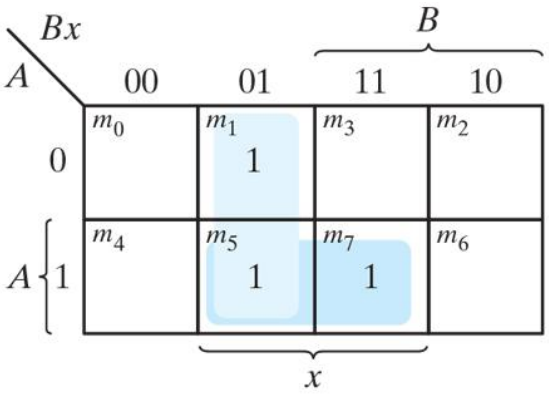
State Table for Sequence Detector

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

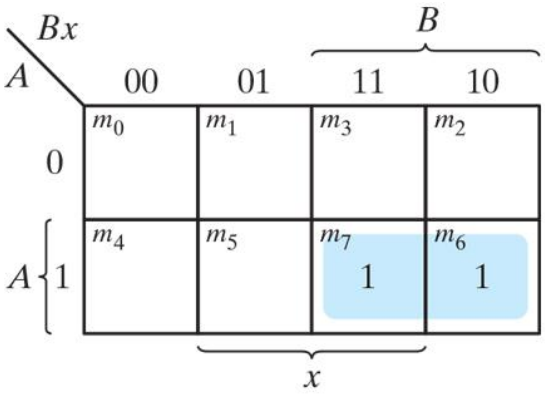
K-Maps for sequence detector



$D_A = Ax + Bx$

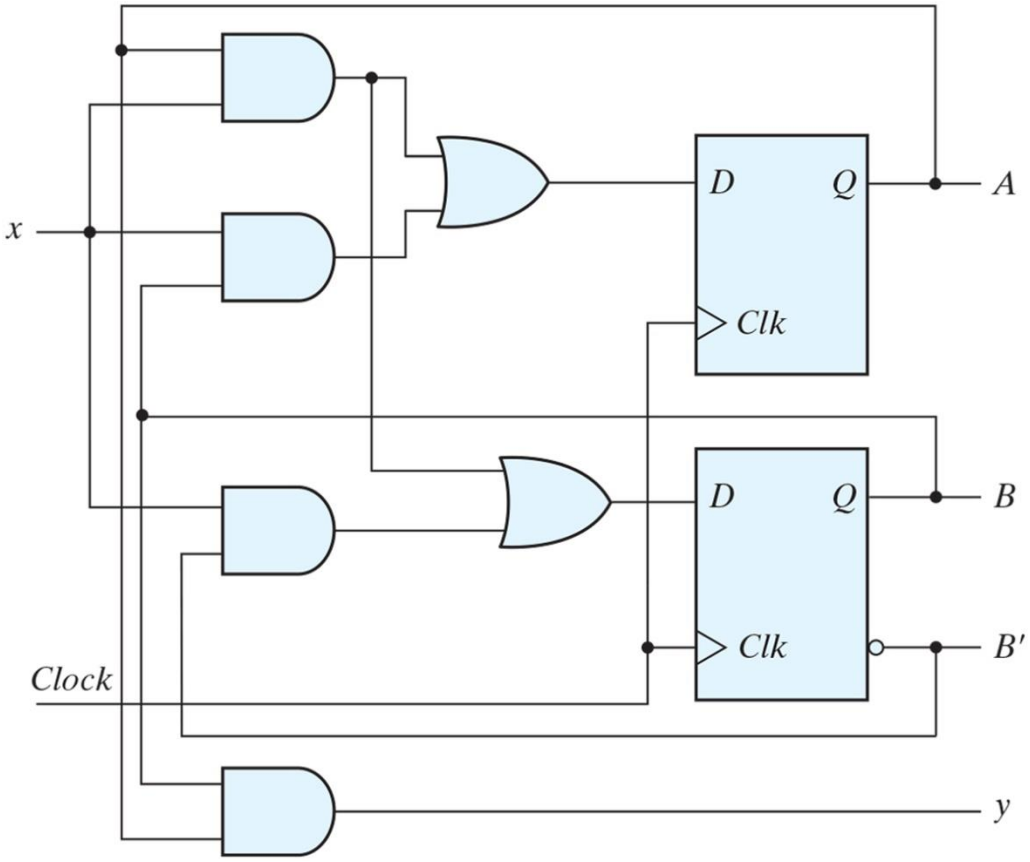


$D_B = Ax + B'x$



$y = AB$

Logic diagram of a Moore-type sequence detector



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* Flip-Flop

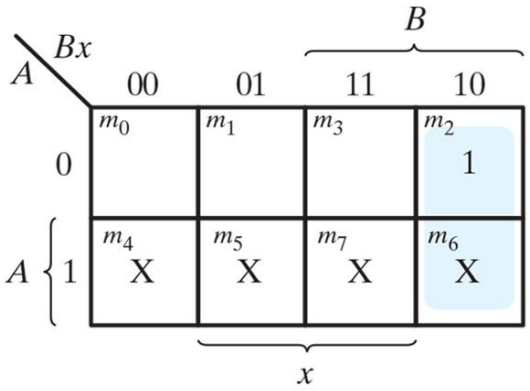
$Q(t)$	$Q(t+1)$	T
0	0	0
0	1	1
1	0	1
1	1	0

(b) *T* Flip-Flop

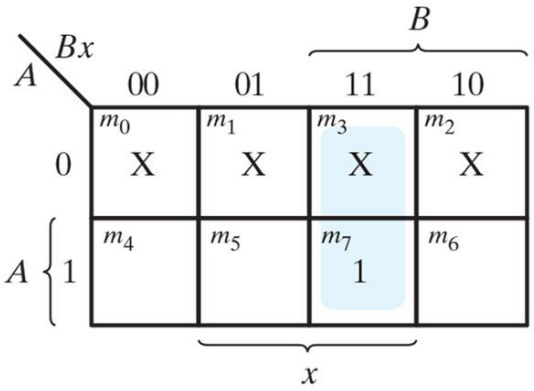
State Table and JK Flip-Flop Inputs

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

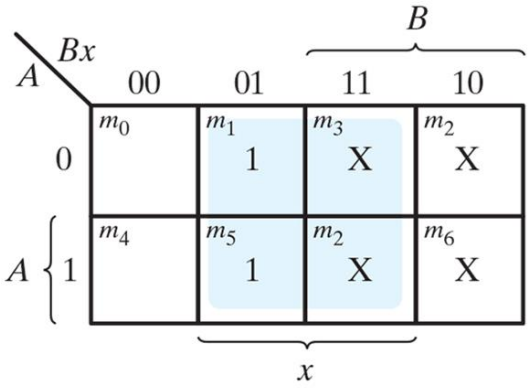
Maps for *J* and *K* input equations



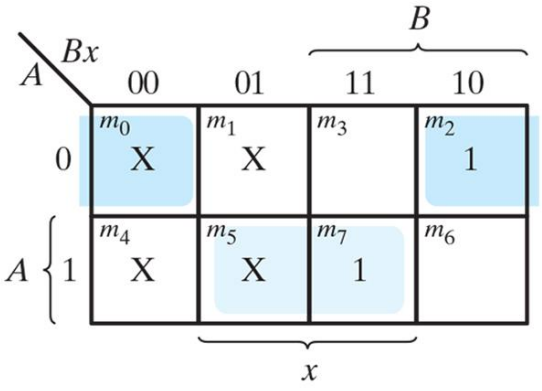
$J_A = Bx'$



$K_A = Bx$

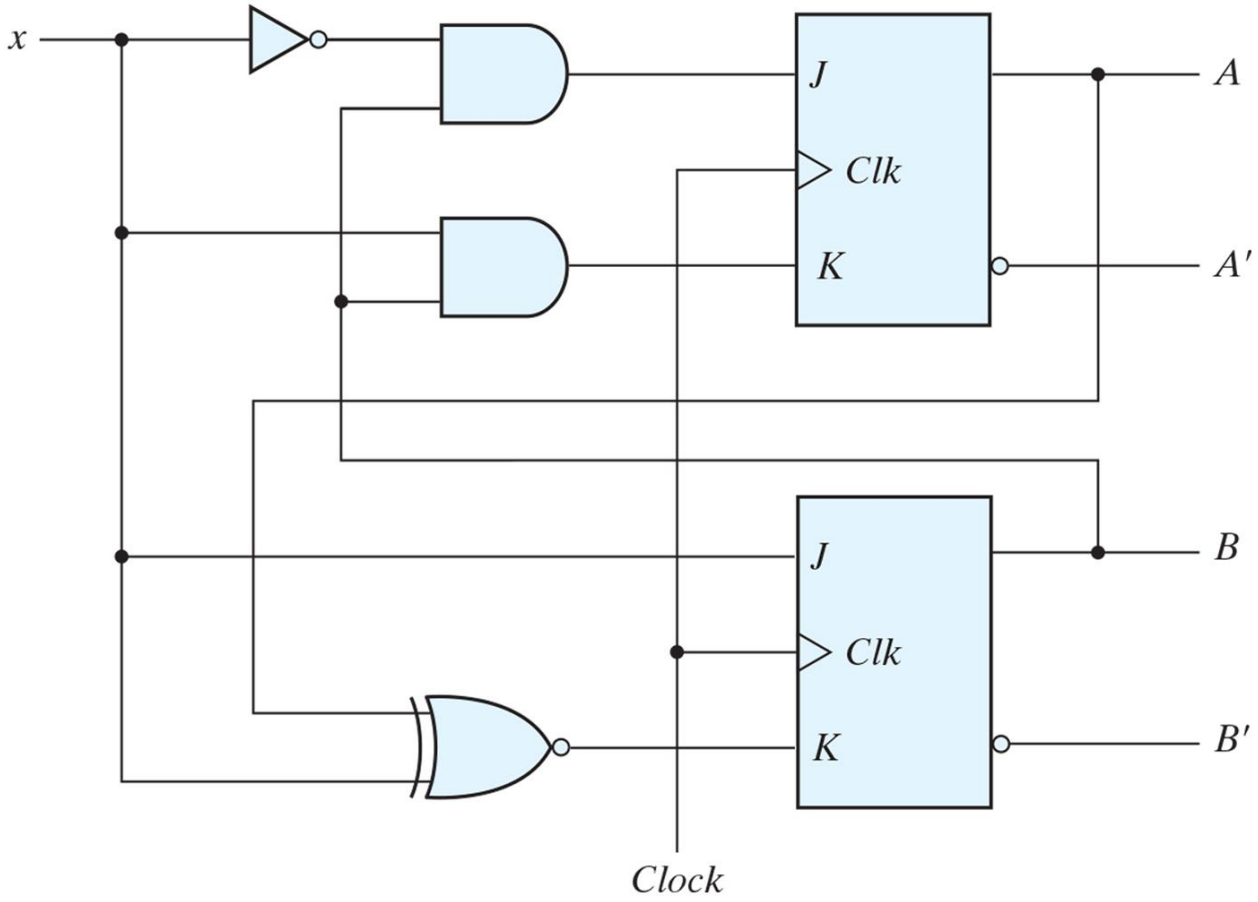


$J_B = x$



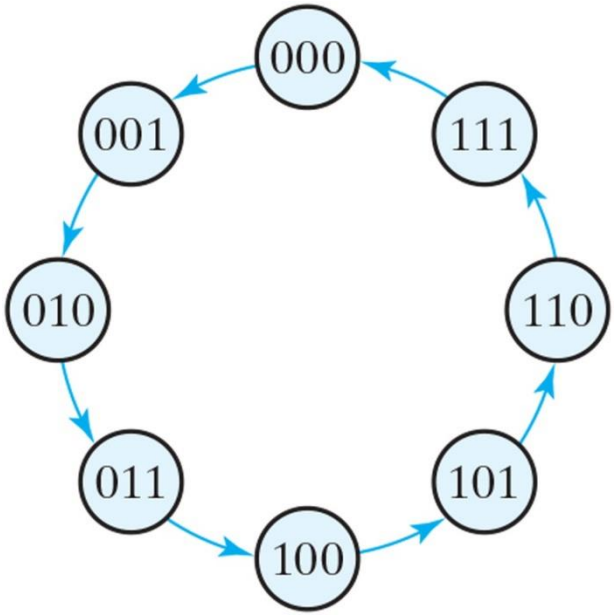
$K_B = (A \oplus x)'$

Logic diagram for sequential circuit with *JK* flip-flops



State diagram of three-bit binary counter

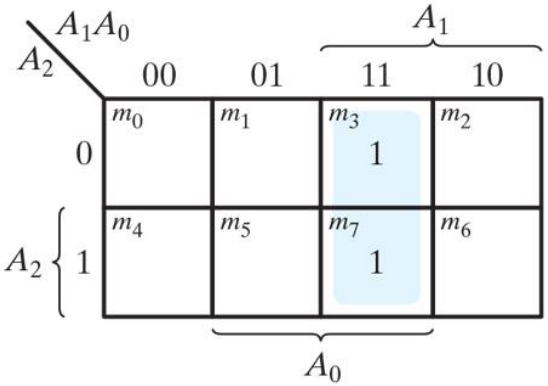
3-bit counter has 3 flip-flops and can count from 0 to 2^n-1(n=3)



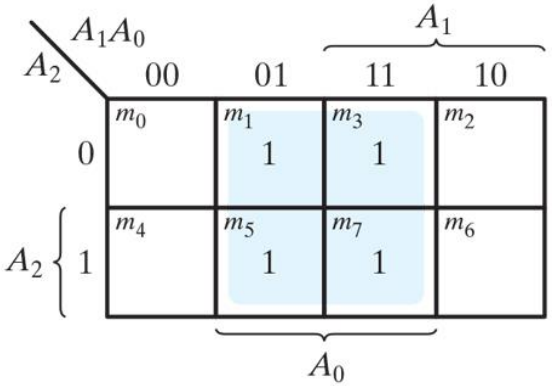
State Table for Three-Bit Counter

Present State			Next State			Flip-Flop Inputs		
A_2	A_1	A_0	A_2	A_1	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	0	1
1	1	1	0	0	0	1	1	1

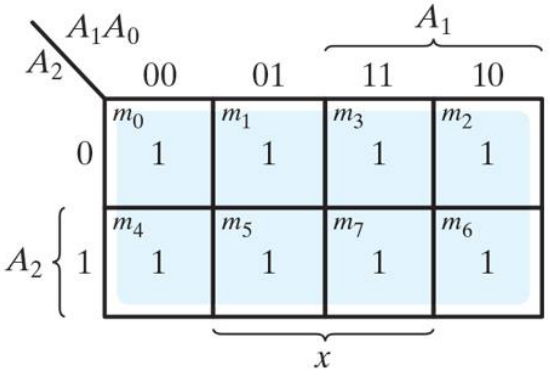
Maps for three-bit binary counter



$T_{A2} = A_1A_0$



$T_{A1} = A_0$



$T_{A0} = 1$

Logic diagram of three-bit binary counter