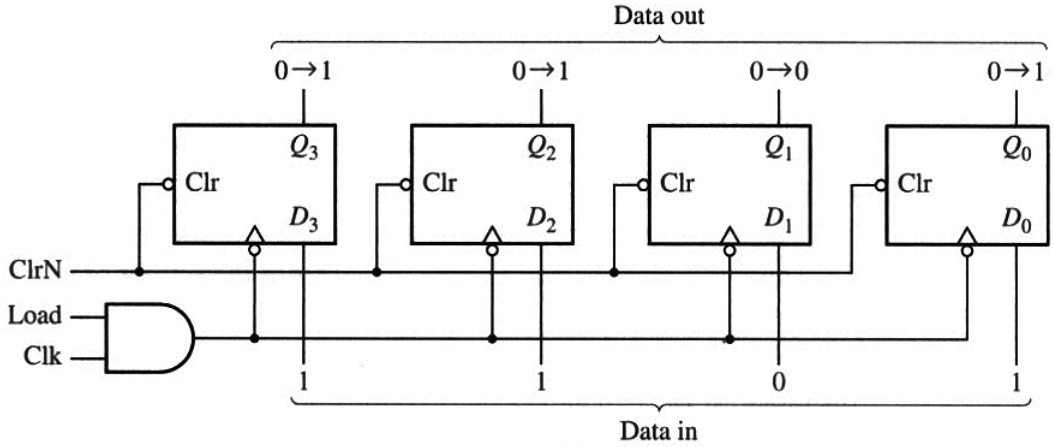
Register and Counters

Outline

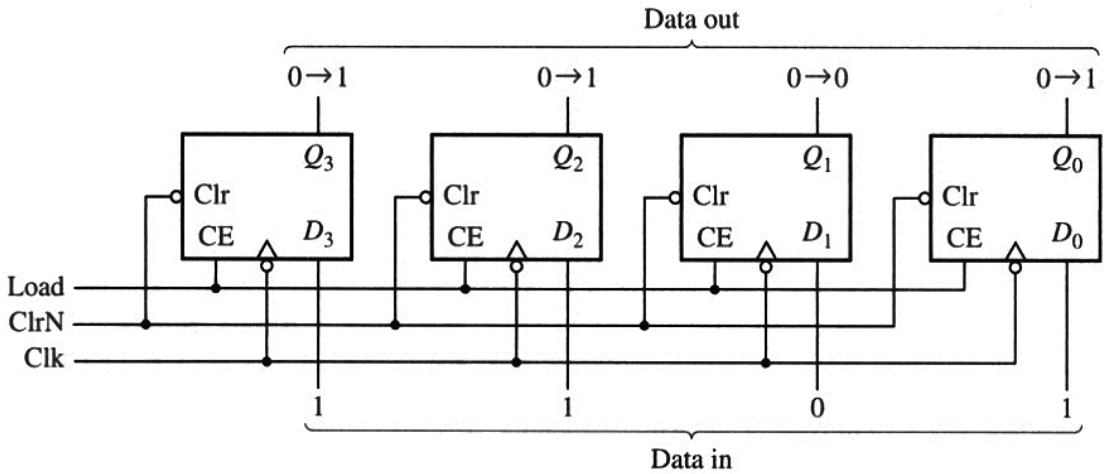
- 12.1 Register and Register Transfers
- 12.2 Shift Register
- 12.3 Design of Binary Counters
- 12.4 Counters for Other Sequences

  Counter Design Using D Flip-Flops
- 12.5 Counter Design Using S-R and J-K
- 12.6 Derivation of Flip-Flop Input Equations

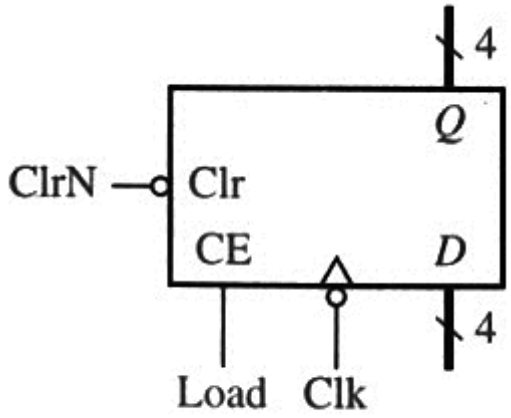
4bit D Flip-Flop Registers



(a) Using gated clock



(b) With clock enable



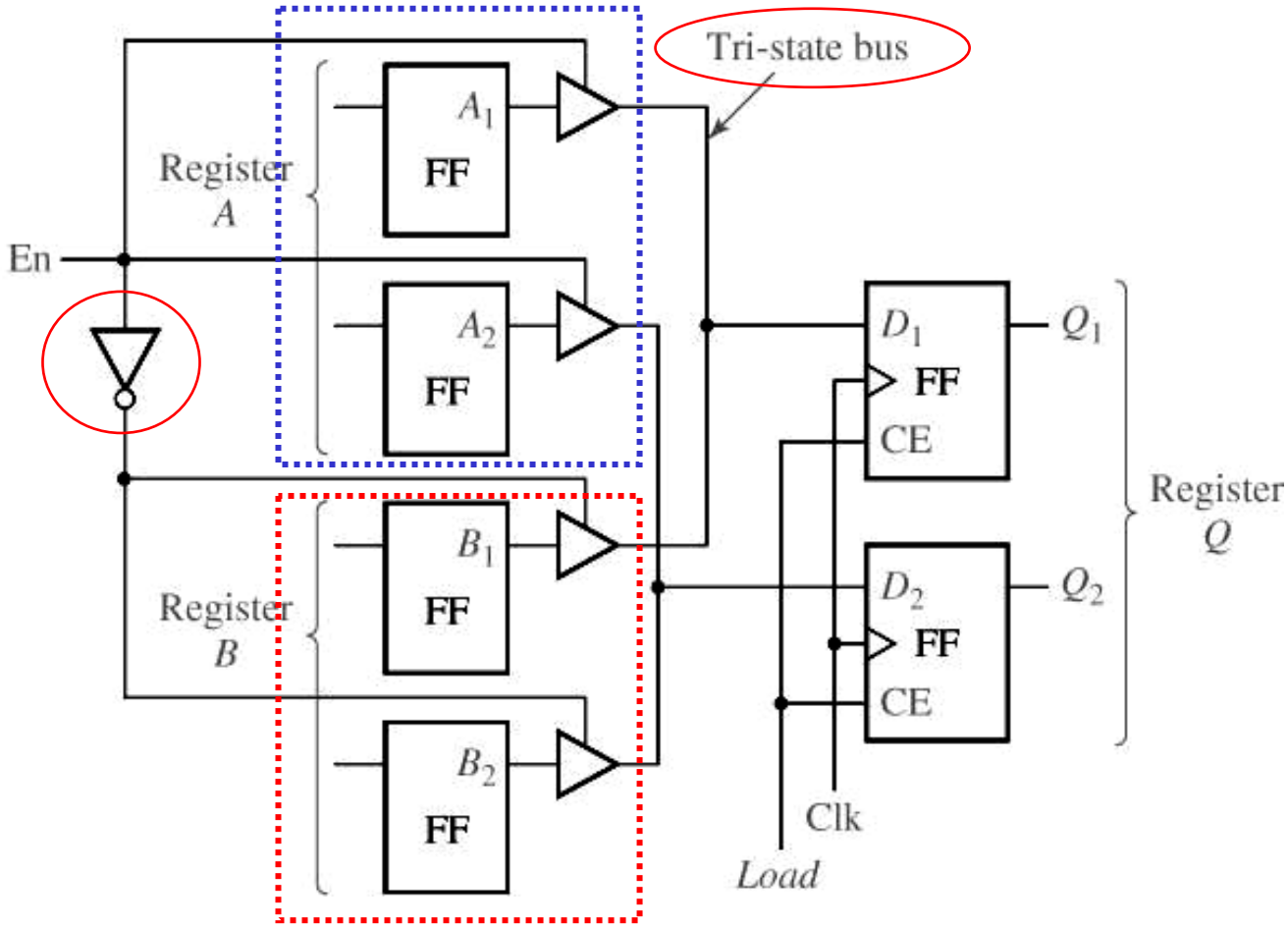
(c) Symbol

Data Transfer Between Registers

Register A = flip-flops A_1 and A_2

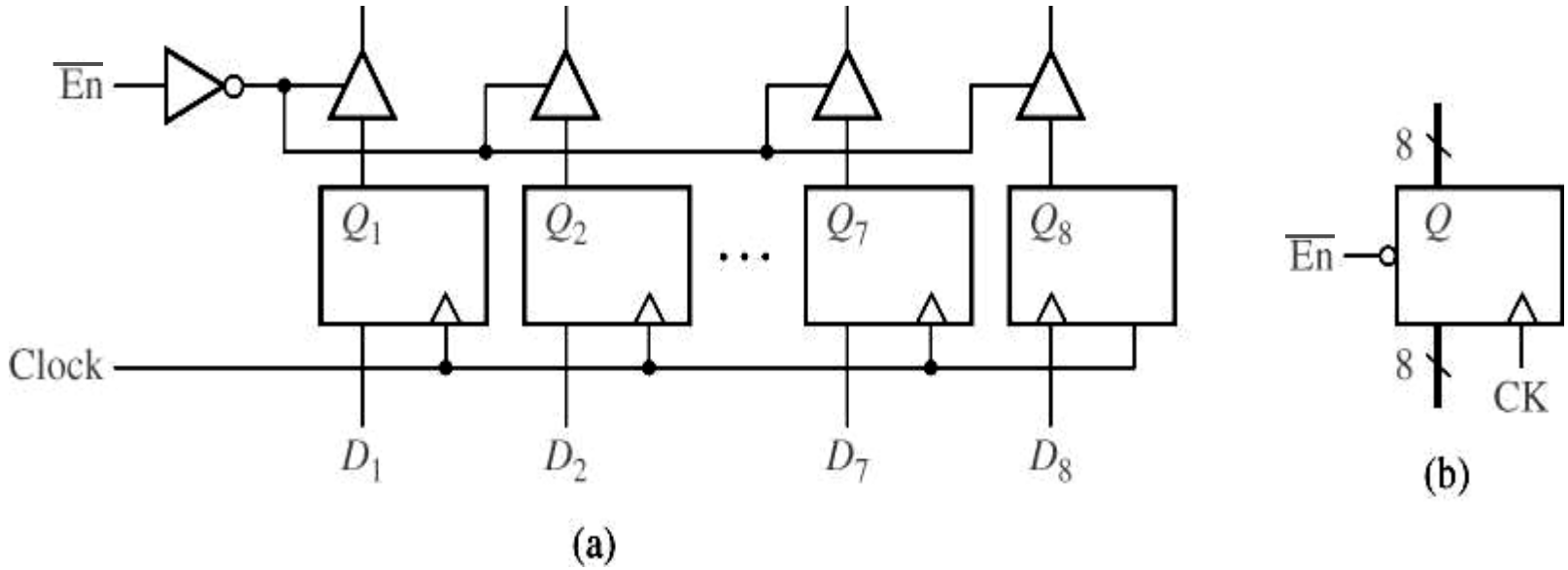
Register B = flip-flops B_1 and B_2

Register Q = flip-flops Q_1 and Q_2

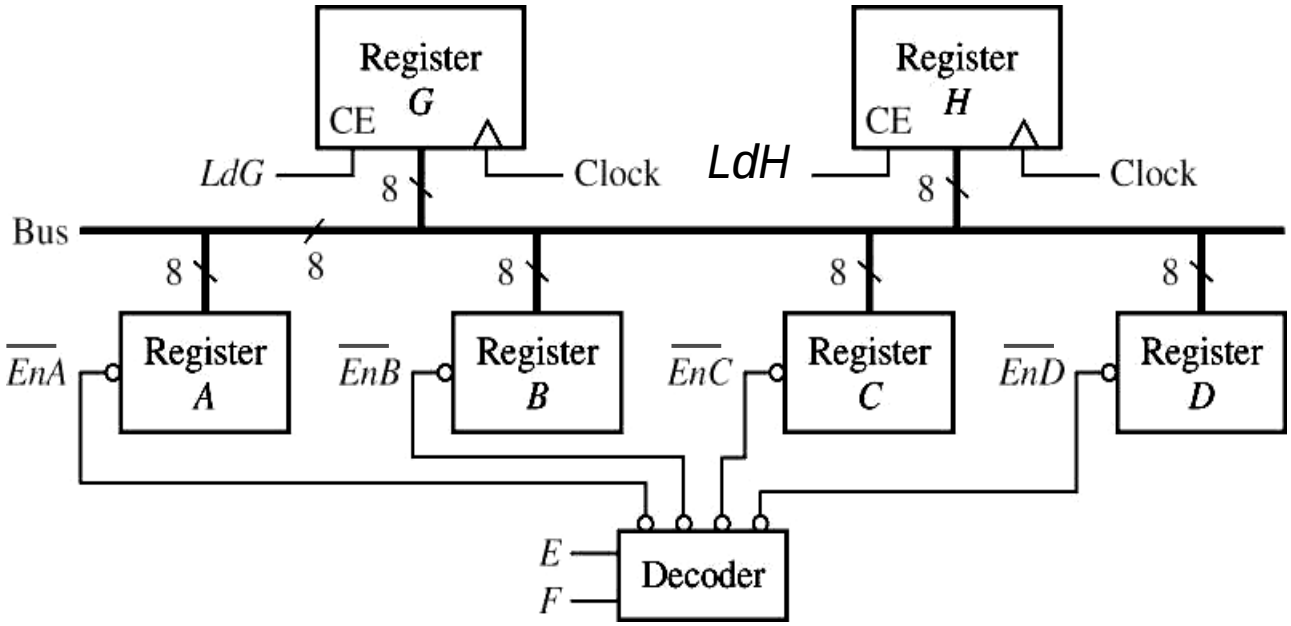


Use of Registers(1/2)

- Logic Diagram for 8-bit Register with Tri-State Output.

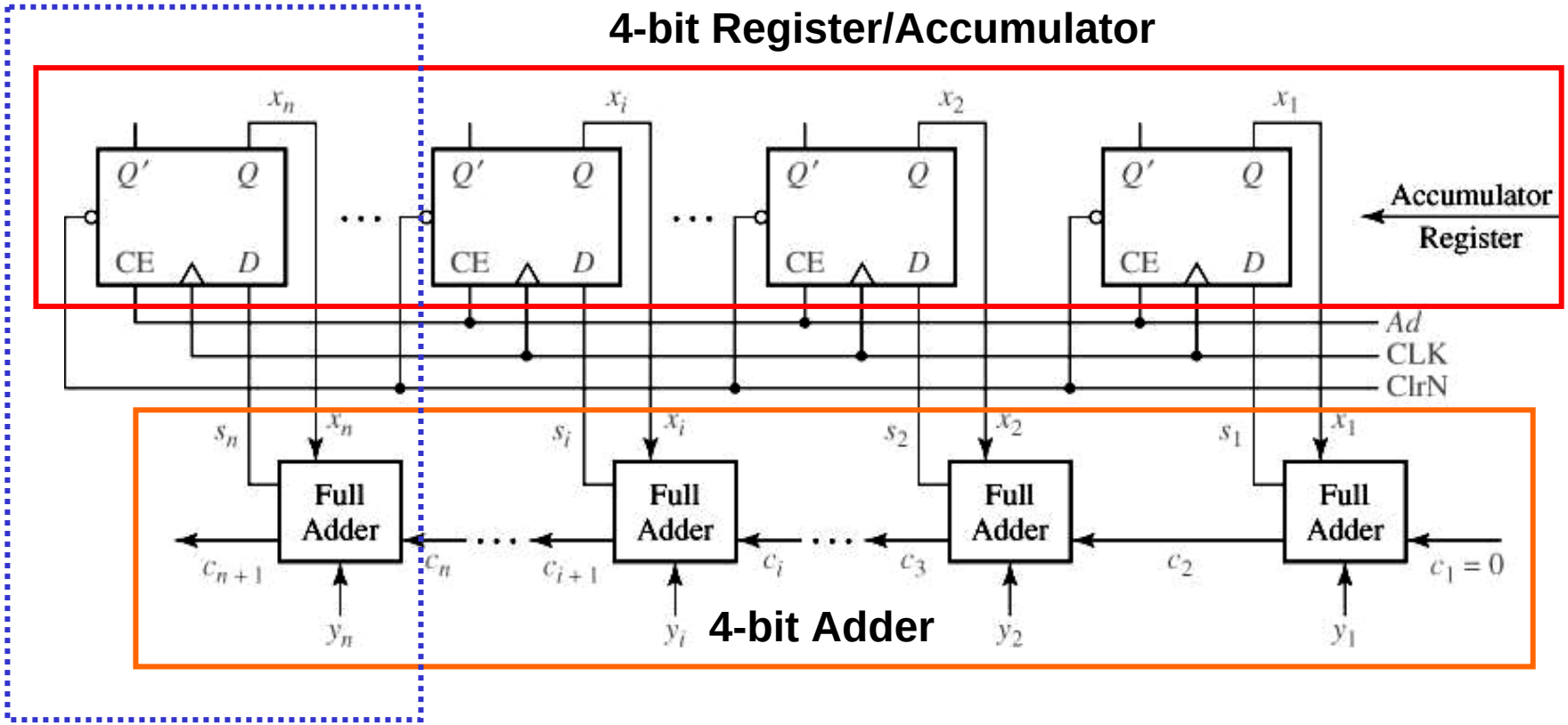


Use of Registers(2/2)

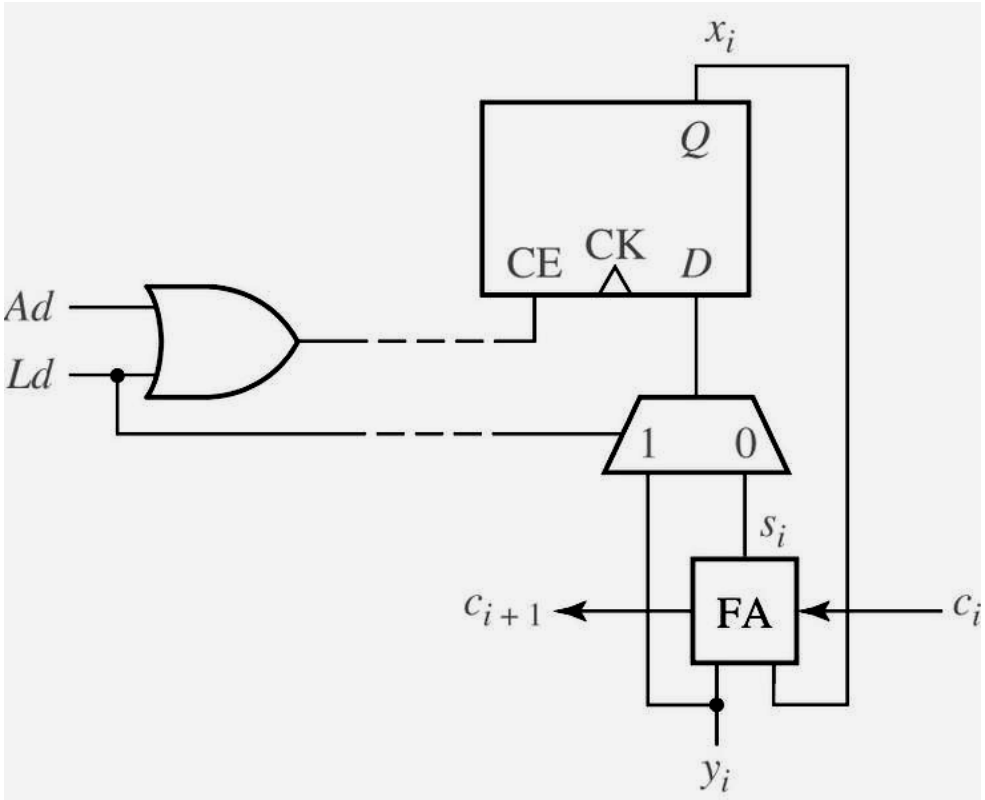


- If EF = 00, A is stored in G (or H).
- If EF = 01, B is stored in G (or H).
- If EF = 10, C is stored in G (or H).
- If EF = 11, D is stored in G (or H).

N-Bit Parallel Adder with Accumulator



Adder Cell with Mux

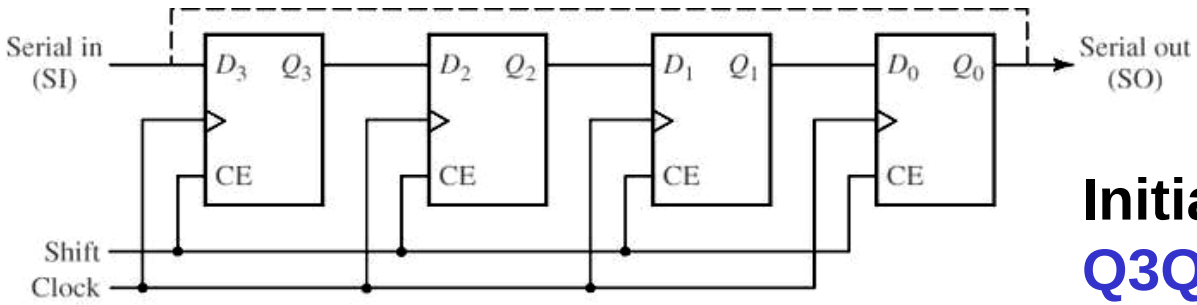


- **A design example for ‘modular’ design**
- **Suitable for Verilog “sub-circuit” design module**

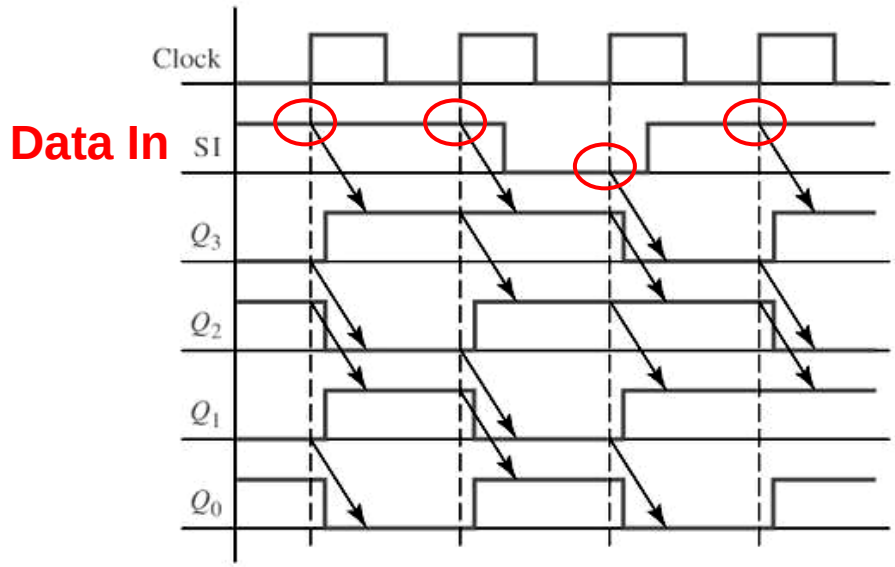
Outline

- 12.1 Register and Register Transfers
- 12.2 Shift Register
- 12.3 Design of Binary Counters
- 12.4 Counters for Other Sequences
  Counter Design Using D Flip-Flops
- 12.5 Counter Design Using S-R and J-K
- 12.6 Derivation of Flip-Flop Input Equations

Shift Registers



(a) Flip-flop connections



(b) Timing diagram

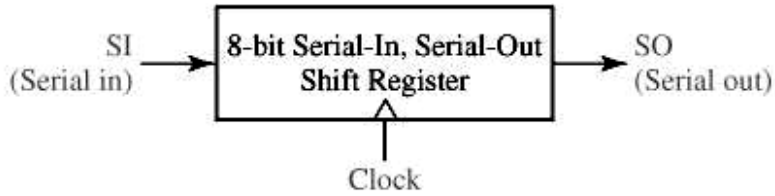
Initial:
Q3Q2Q1Q0 = 0101

SI = 1,1,0,1

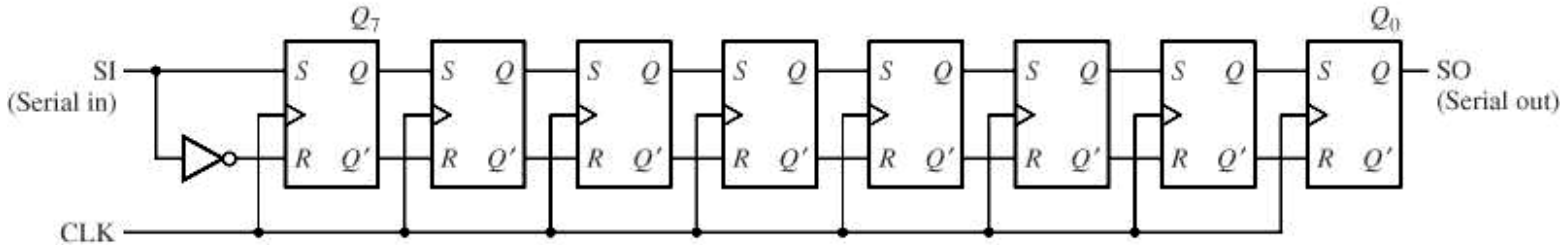
Register States:

0101
1010
1101
0110
1011

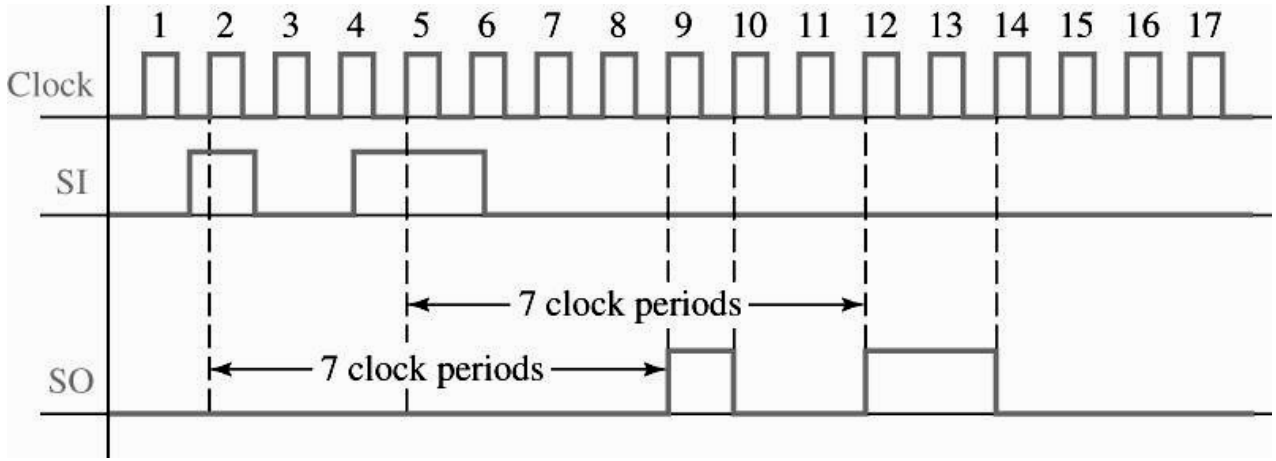
Application



(a) Block diagram



(b) Logic diagram

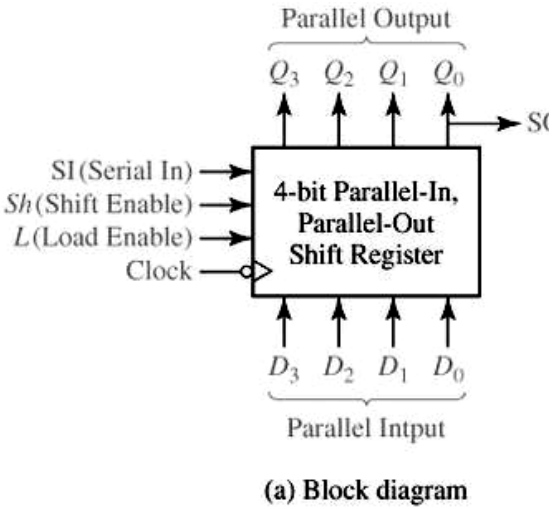


For the purpose

- **Delay of 7 clock Cycles**
- **Buffer of data**

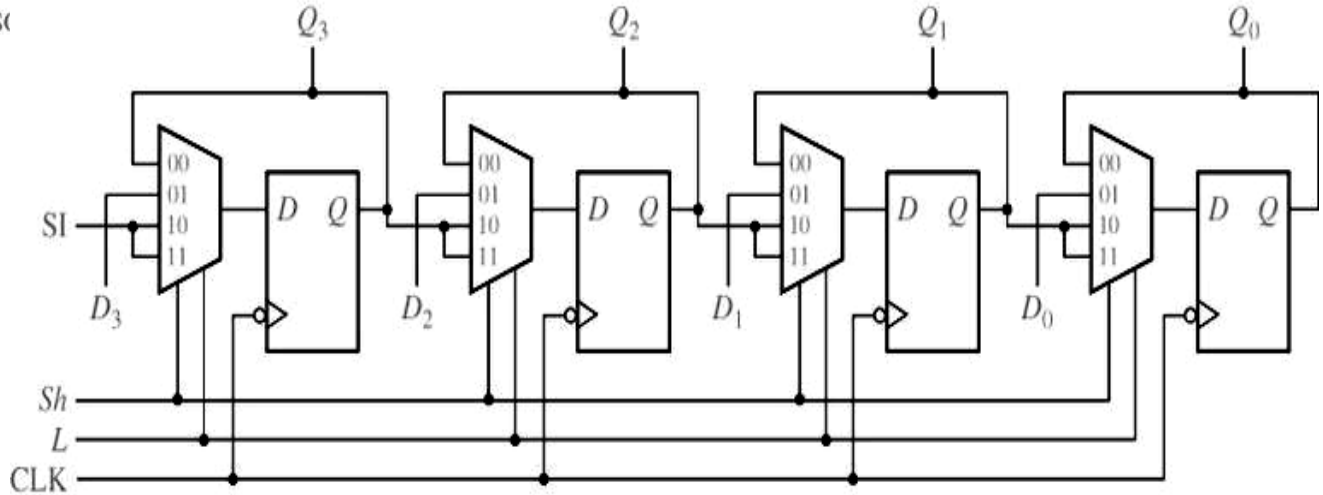
Parallel-in ,Parallel-out (PIPO) Right Shift Register

Inputs		Next State				Action
Sh (Shift)	L (Load)	Q_3^+	Q_2^+	Q_1^+	Q_0^+	
0	0	Q_3	Q_2	Q_1	Q_0	**No change**
0	1	D_3	D_2	D_1	D_0	**Load**
1	X	SI	Q_3	Q_2	Q_1	**Right shift**



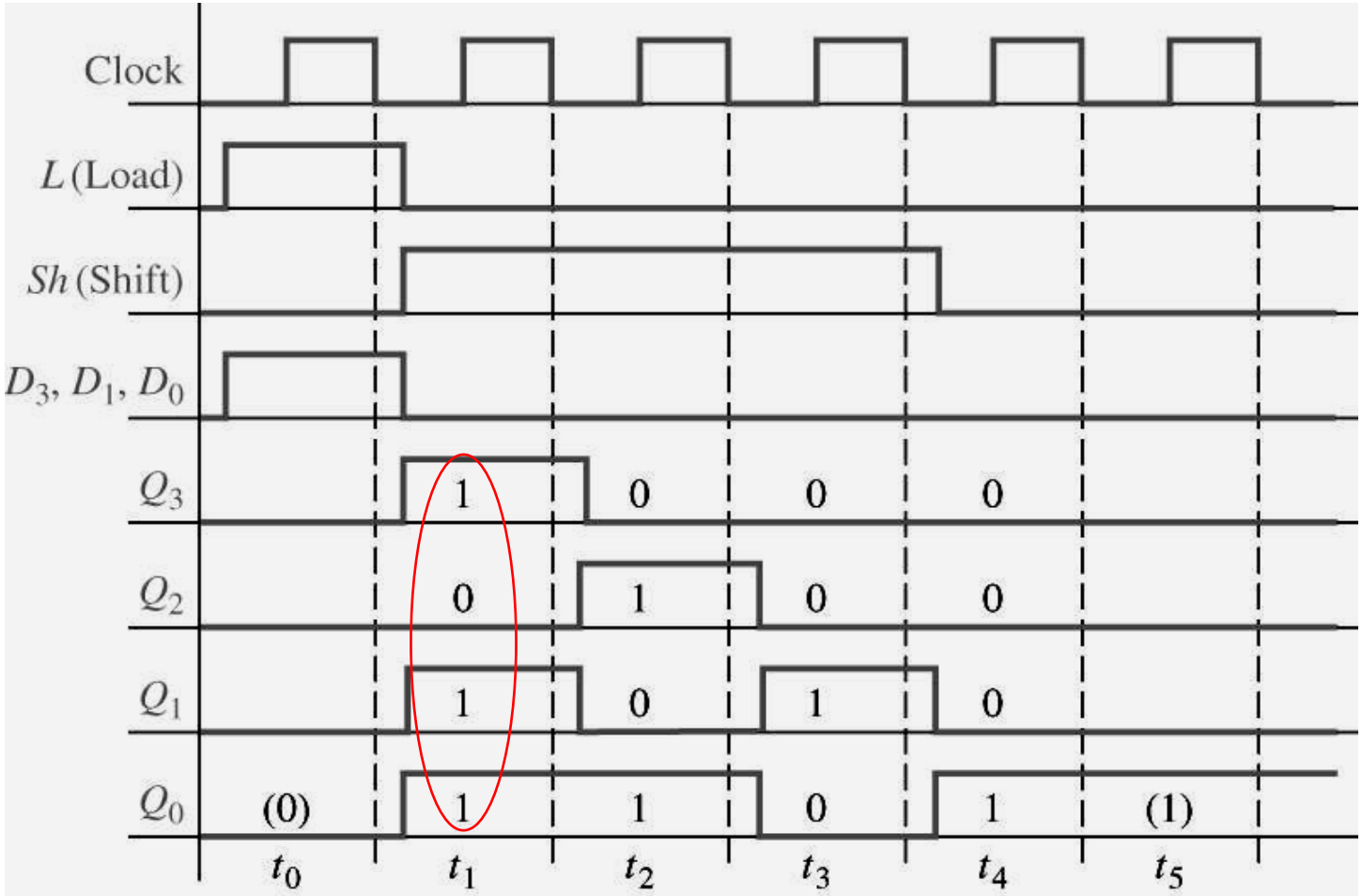
(a) Block diagram

$D_3D_2D_1D_0$= 1011



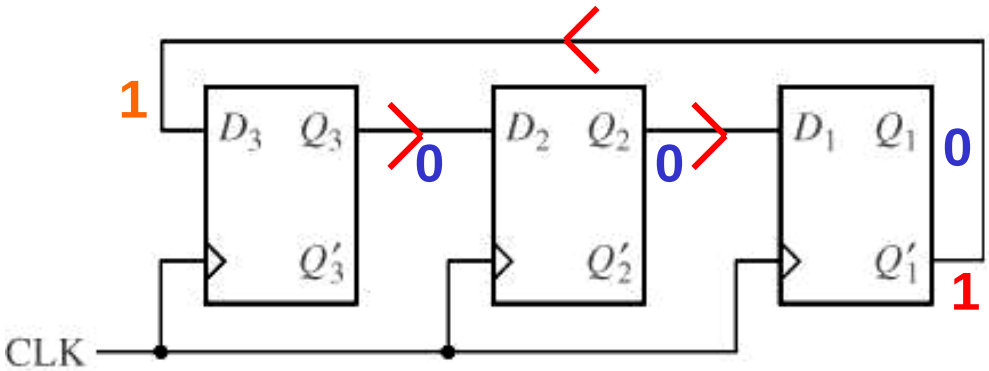
(b) Implementation using flip-flops and MUXes

Timing Diagram

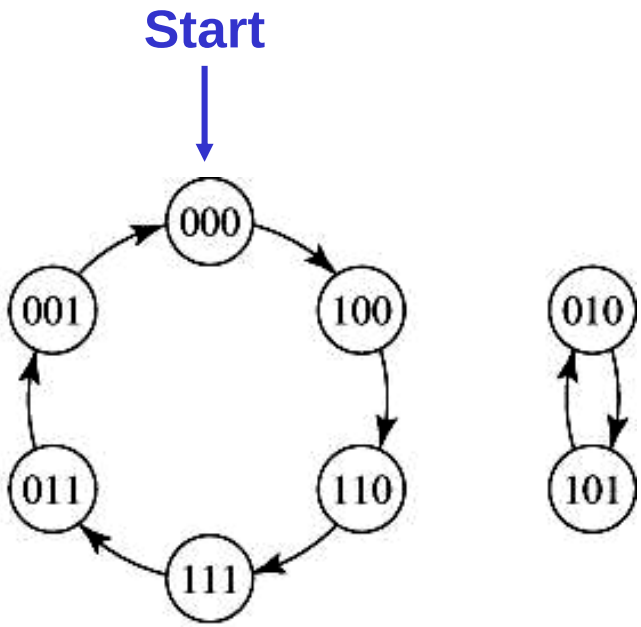


$D_3D_2D_1D_0$= 1011

Shift Register with Inverted Feedback



(a) Flip-flop connections



(b) State graph

Outline

- 12.1 Register and Register Transfers
- 12.2 Shift Register
- 12.3 Design of Binary Counters
- 12.4 Counters for Other Sequences

  Counter Design Using D Flip-Flops
- 12.5 Counter Design Using S-R and J-K
- 12.6 Derivation of Flip-Flop Input Equations

Use T F/F (1/2)

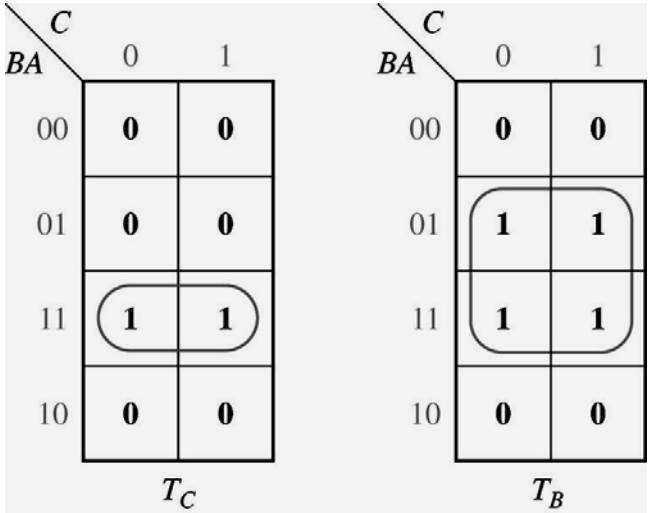
State Table

Present State			Next State			Flip-Flop Inputs		
C	B	A	C^+	B^+	A^+	T_C	T_B	T_A
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

As Function of (A,B,C)!

Using T F/F (2/2)

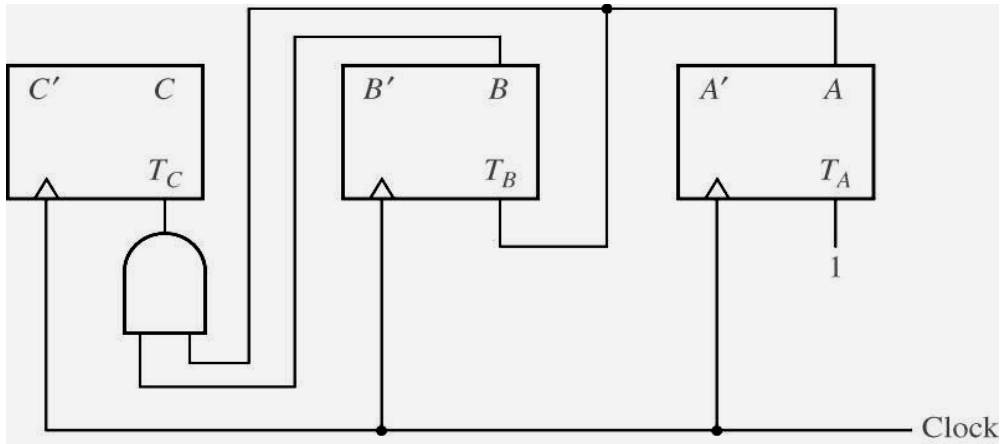
- K-map



$T_C = AB$

$T_B = A$

- Implement



Use D F/F (1/2)

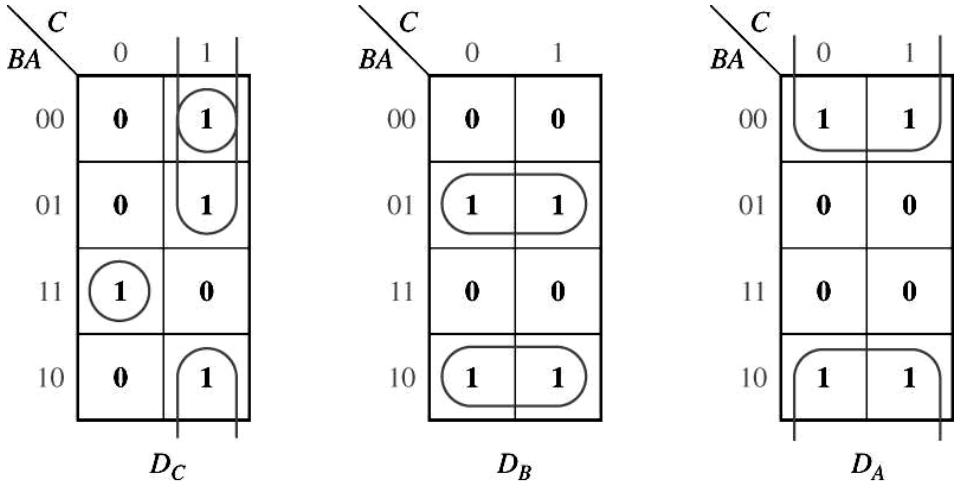
State Table

Present State			Next State			Flip-Flop Inputs		
C	B	A	C^+	B^+	A^+	D_C	D_B	D_A
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	0
0	1	0	0	1	1	0	1	1
0	1	1	1	0	0	1	0	0
1	0	0	1	0	1	1	0	1
1	0	1	1	1	0	1	1	0
1	1	0	1	1	1	1	1	1
1	1	1	0	0	0	0	0	0

As Function of (A,B,C)!

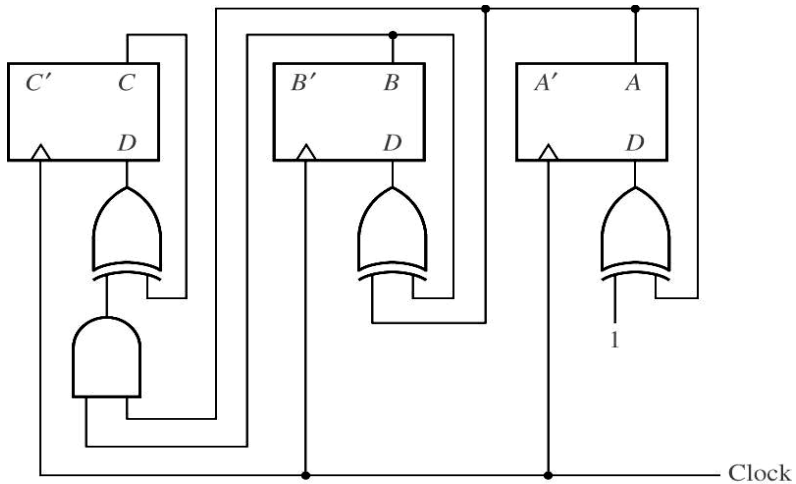
Using D F/F (2/2)

- K-map



= C ⊕ AB **= A ⊕ B** **= A'**

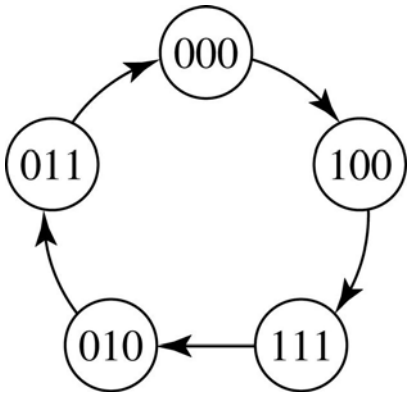
- Circuit Implement



Outline

- 12.1 Register and Register Transfers
- 12.2 Shift Register
- 12.3 Design of Binary Counters
- 12.4 Counters for Other Sequences
  - Counter Design Using D Flip-Flops
- 12.5 Counter Design Using S-R and J-K
- 12.6 Derivation of Flip-Flop Input Equations

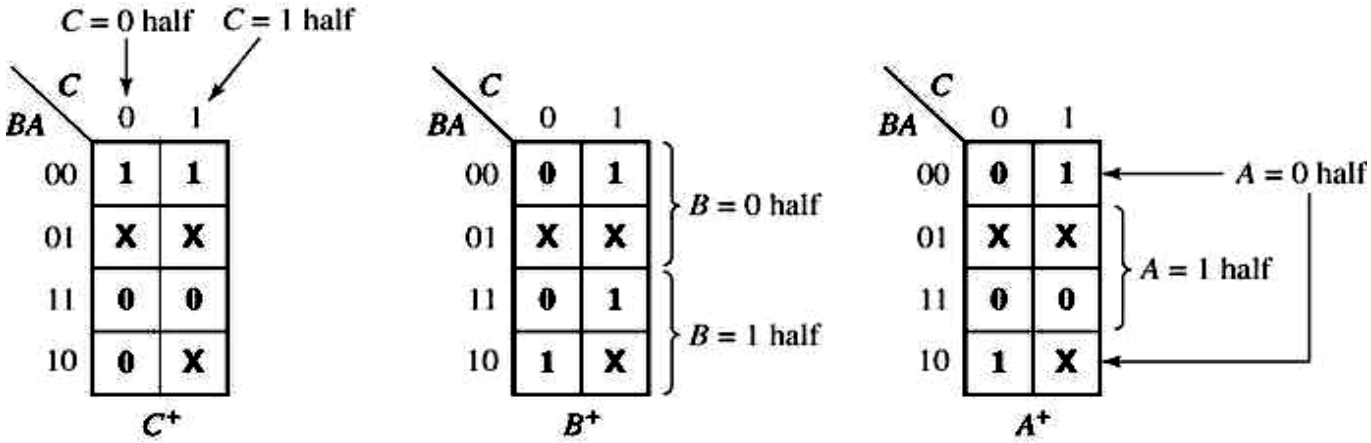
State Diagram of a Counter



C B A	$C^+B^+A^+$
0 0 0	1 0 0
0 0 1	- - -
0 1 0	0 1 1
0 1 1	0 0 0
1 0 0	1 1 1
1 0 1	- - -
1 1 0	- - -
1 1 1	0 1 0

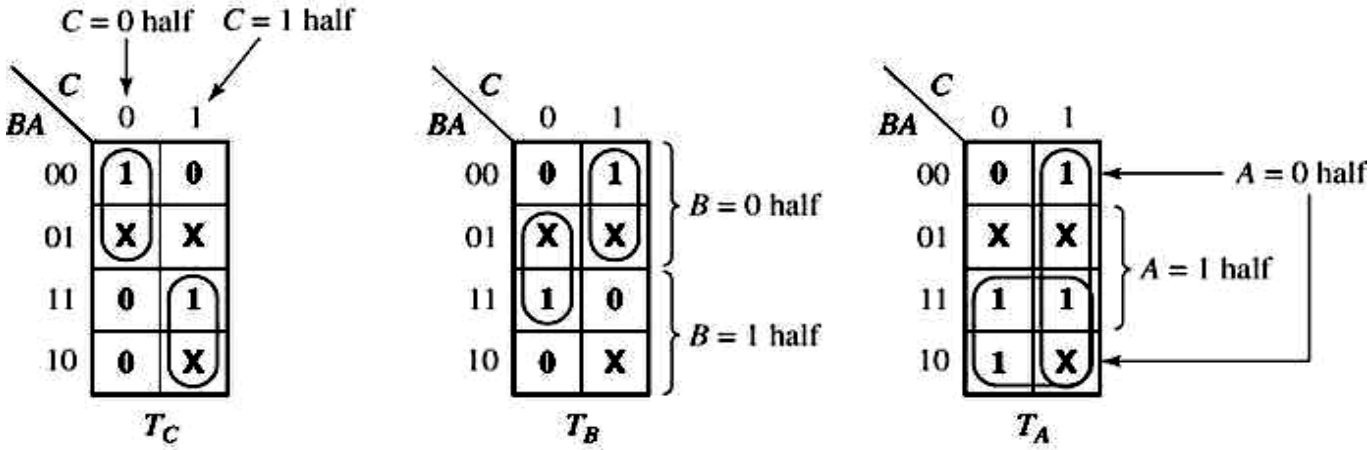
$T_C T_B T_A$
1 0 0
- - -
0 0 1
0 1 1
0 1 1
- - -
- - -
1 0 1

K-map Derivation



(a) Next-state maps for Table 12-3

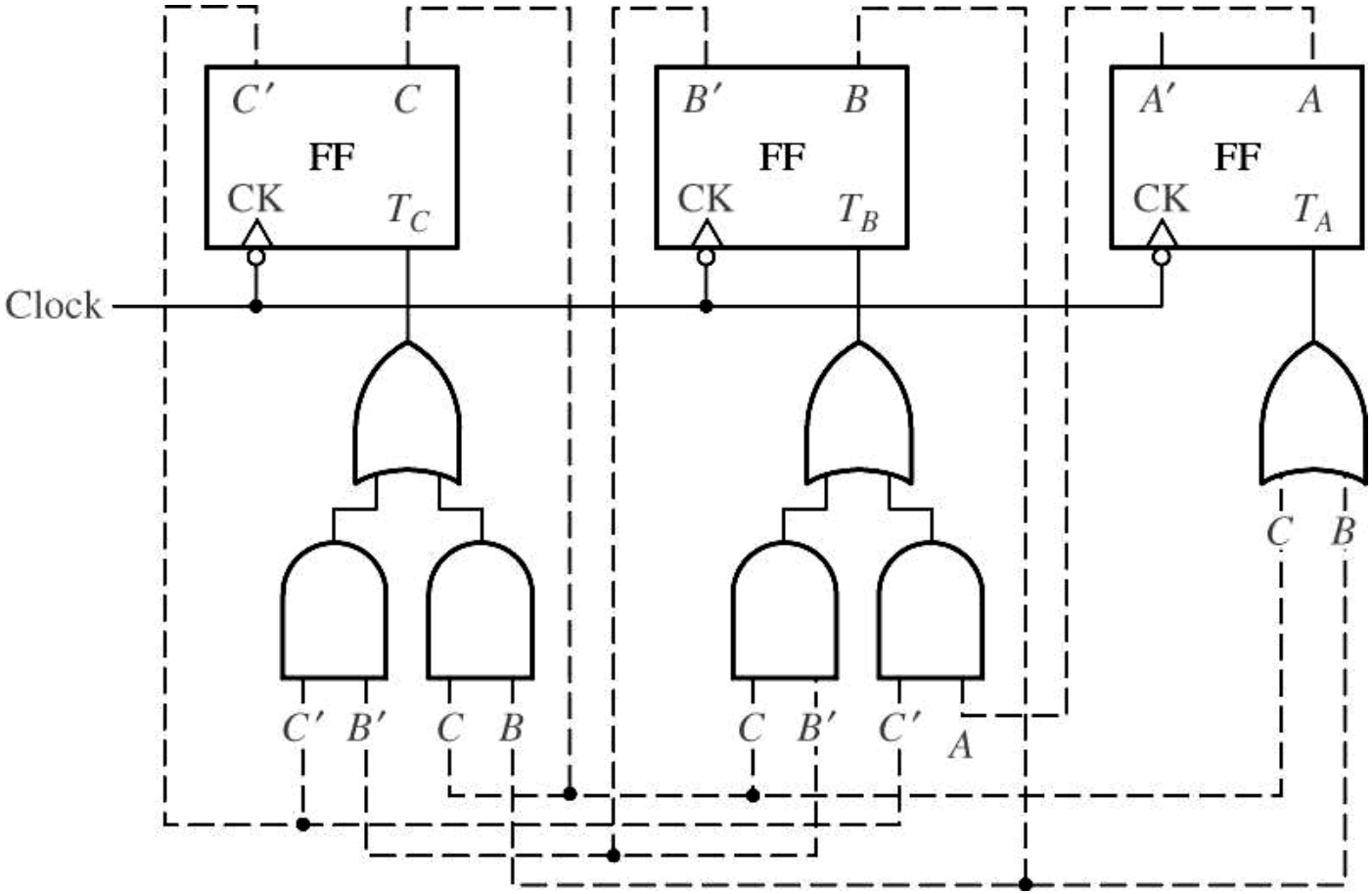
Q Q^+	T
0 0	0
0 1	1
1 0	1
1 1	0



(b) Derivation of T inputs

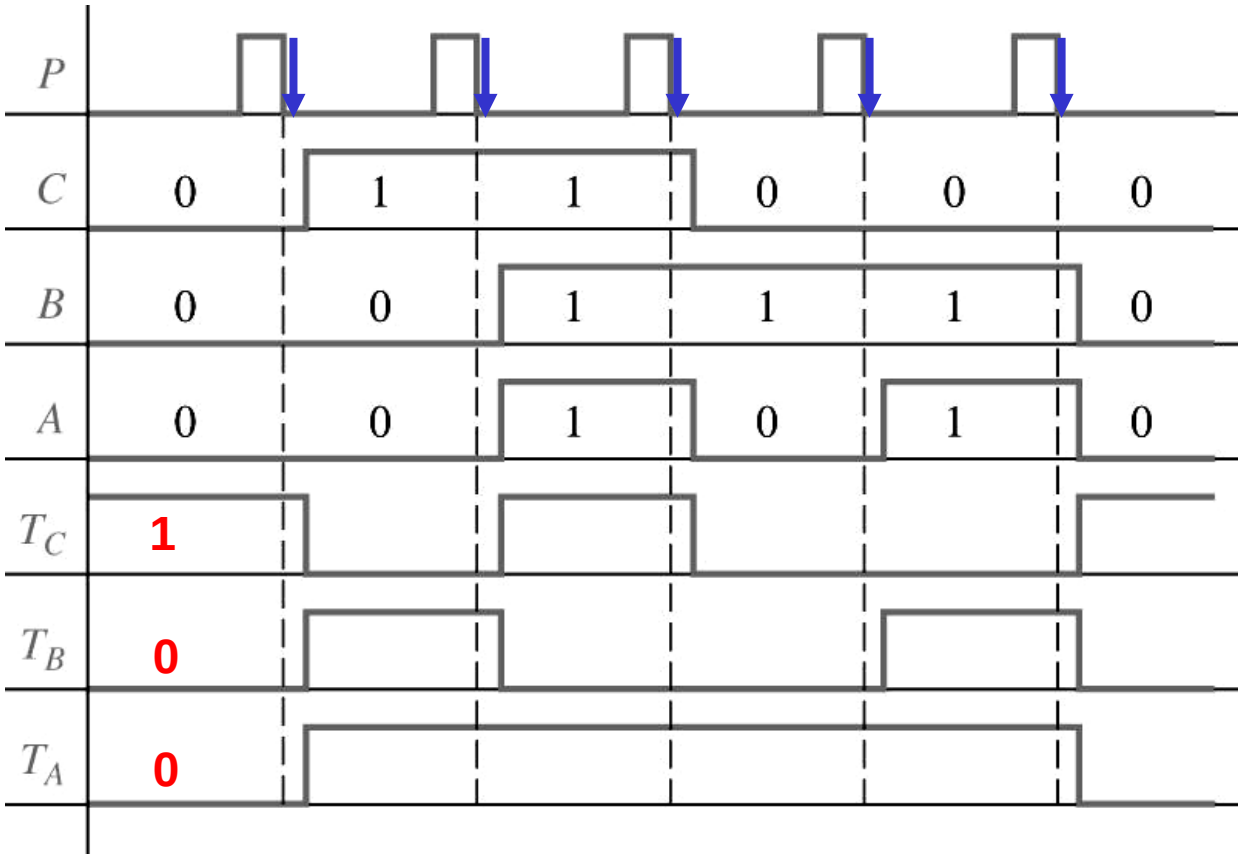
$$T = Q^+ \oplus Q$$

Logic Network



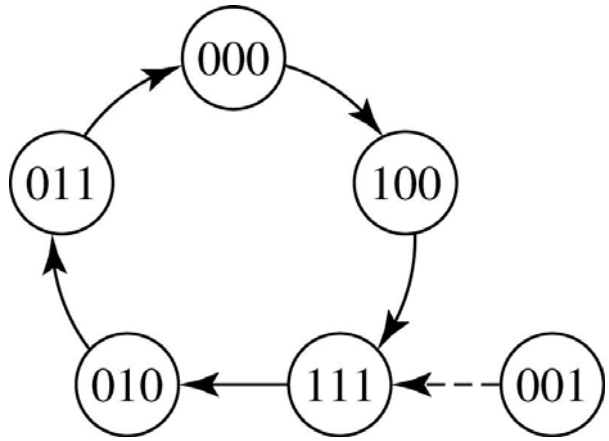
Timing Diagram of Counter

- Negative-edge triggered counter (3-bits).



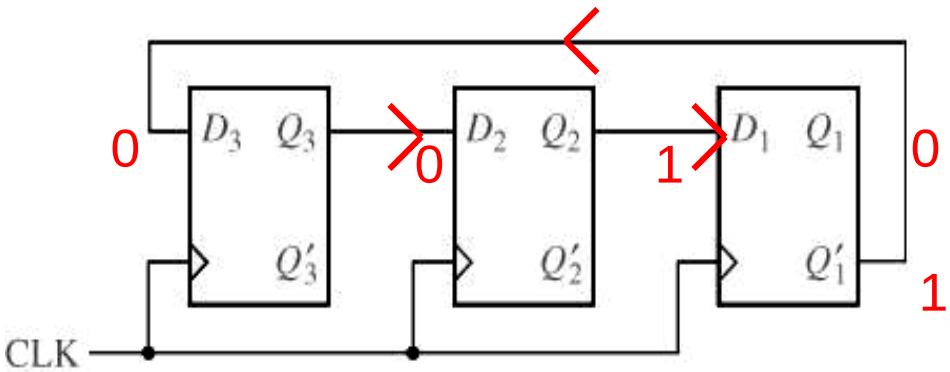
IF....

- IF F/F's are initially set to A=0, B=0, C=1. Tracking signals through the network shows that $T_A=T_B=1$, so the state changes to 111.

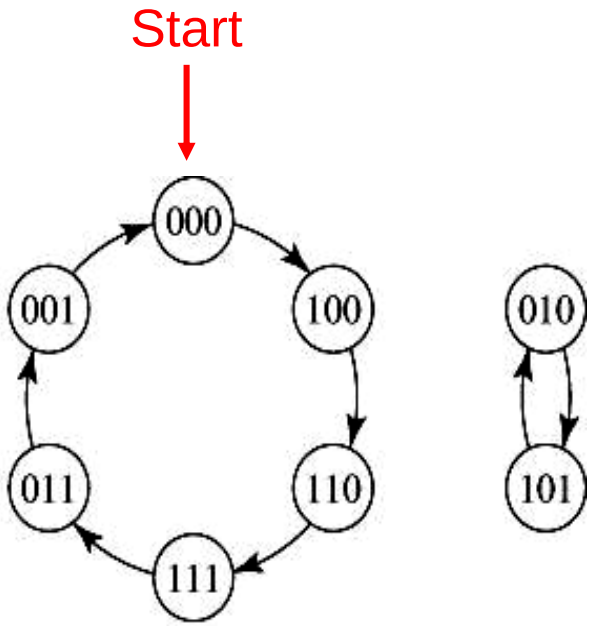


- When the power-on, the states of all F/F's are unpredictable
- ➔ Don't care states should be checked to make sure that they eventually lead into the main counting sequence → or use **"power-on" reset.**

c.f. Shift Register with Inverted Feedback



(a) Flip-flop connections

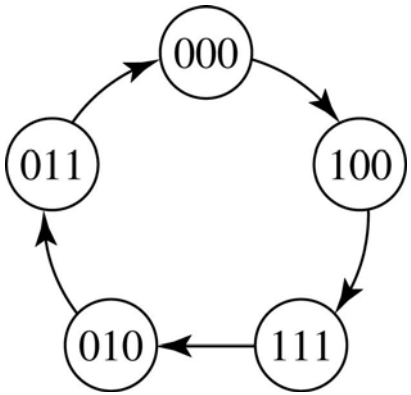


(b) State graph

Outline

- 12.1 Register and Register Transfers
- 12.2 Shift Register
- 12.3 Design of Binary Counters
- 12.4 Counters for Other Sequences
  - Counter Design Using D Flip-Flops
- 12.5 Counter Design Using S-R and J-K
- 12.6 Derivation of Flip-Flop Input Equations

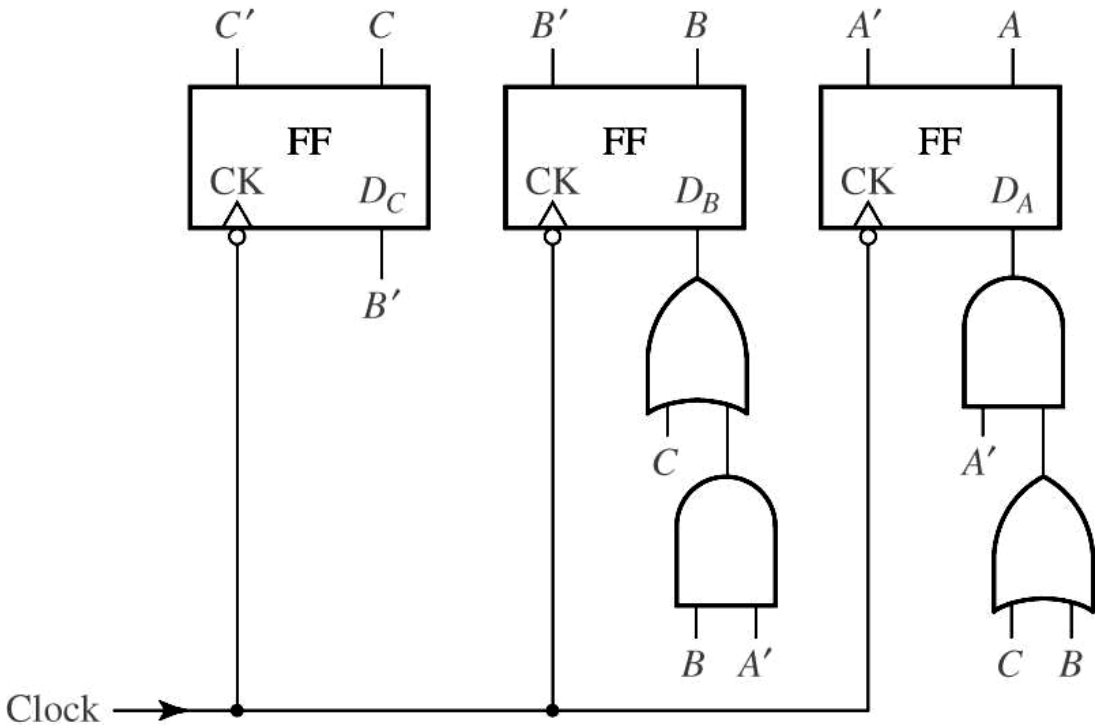
State Diagram of a Counter



C B A	$C^+B^+A^+$	$D_CD_BD_A$
0 0 0	1 0 0	1 0 0
0 0 1	- - -	- - -
0 1 0	0 1 1	0 1 1
0 1 1	0 0 0	0 0 0
1 0 0	1 1 1	1 1 1
1 0 1	- - -	- - -
1 1 0	- - -	- - -
1 1 1	0 1 0	0 1 0

Counter Design Using D Flip-Flops

- $D_C = C^+ = B'$
- $D_B = B^+ = C + BA'$
- $D_A = A^+ = CA' + BA' = A'(C + B)$

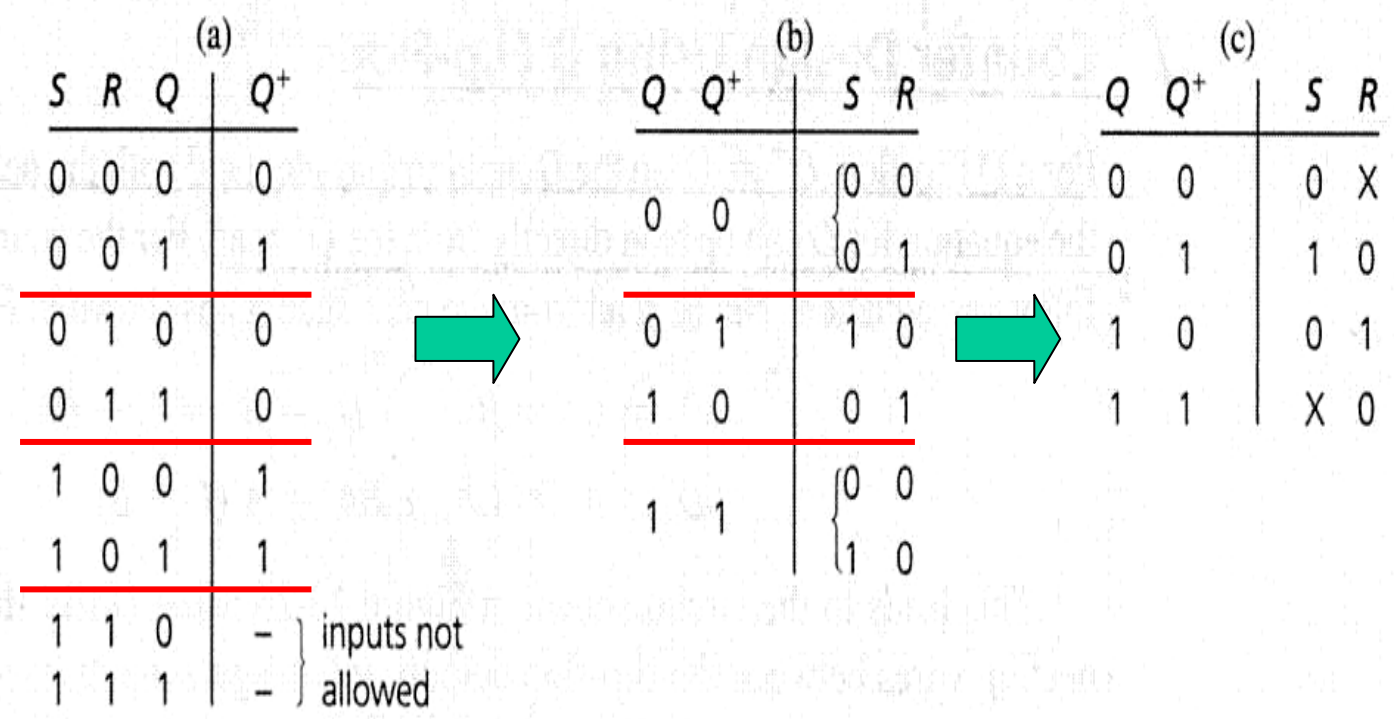


Outline

- 12.1 Register and Register Transfers
- 12.2 Shift Register
- 12.3 Design of Binary Counters
- 12.4 Counters for Other Sequences
  - Counter Design Using D Flip-Flops
- 12.5 Counter Design Using S-R and J-K
- 12.6 Derivation of Flip-Flop Input Equations

Using S-R F/F for Counter

- S-R flip-flop inputs



(a)

S	R	Q	Q^+	
0	0	0	0	
0	0	1	1	
0	1	0	0	
0	1	1	0	
1	0	0	1	
1	0	1	1	
1	1	0	–	inputs not allowed
1	1	1	–	

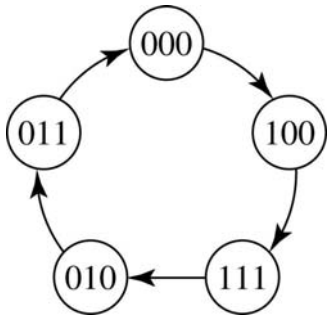
(b)

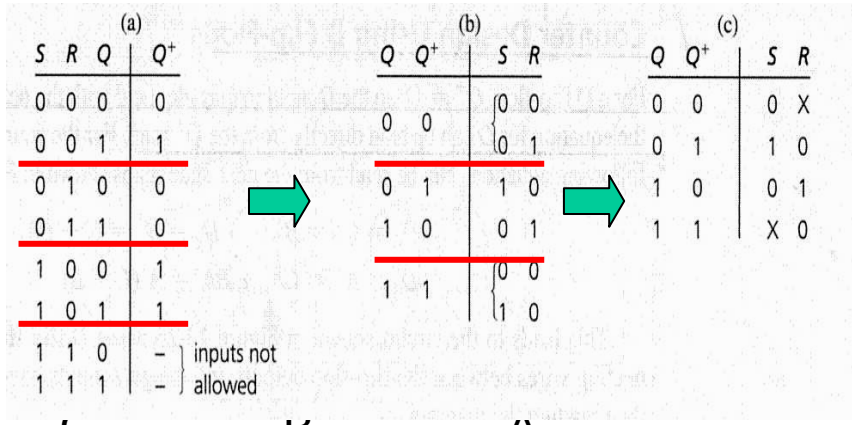
Q	Q^+	S	R
0	0	0	0
		0	1
0	1	1	0
1	0	0	1
1	1	0	0
		1	0

(c)

Q	Q^+	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

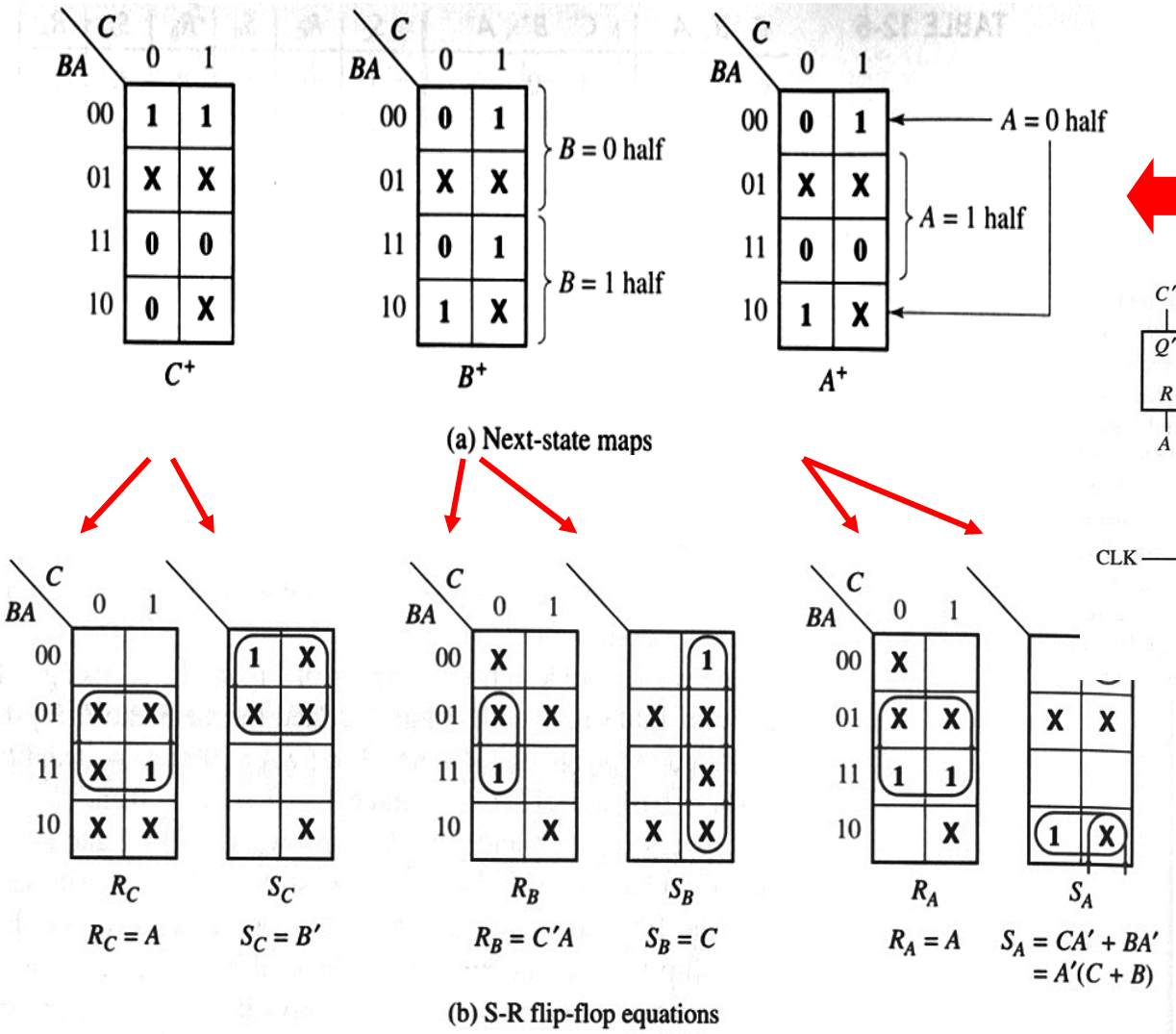
Using S-R F/F



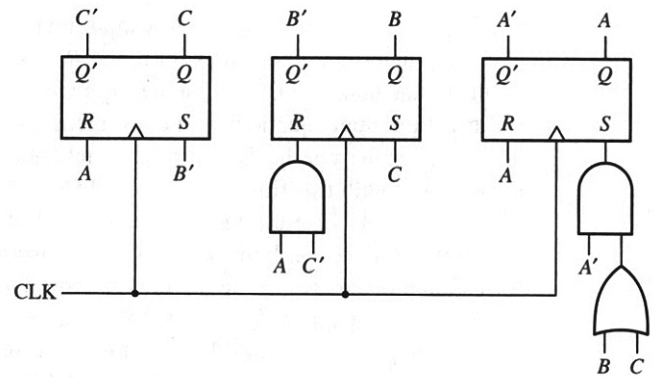


C	B	A	C^+	B^+	A^+	S_C	R_C	S_B	R_B	S_A	R_A
0	0	0	1	0	0	1	0	0	X	0	X
0	0	1	–	–	–	X	X	X	X	X	X
0	1	0	0	1	1	0	X	X	0	1	0
0	1	1	0	0	0	0	X	0	1	0	1
1	0	0	1	1	1	X	0	1	0	1	0
1	0	1	–	–	–	X	X	X	X	X	X
1	1	0	–	–	–	X	X	X	X	X	X
1	1	1	0	1	0	0	1	X	0	0	1

Using S-R

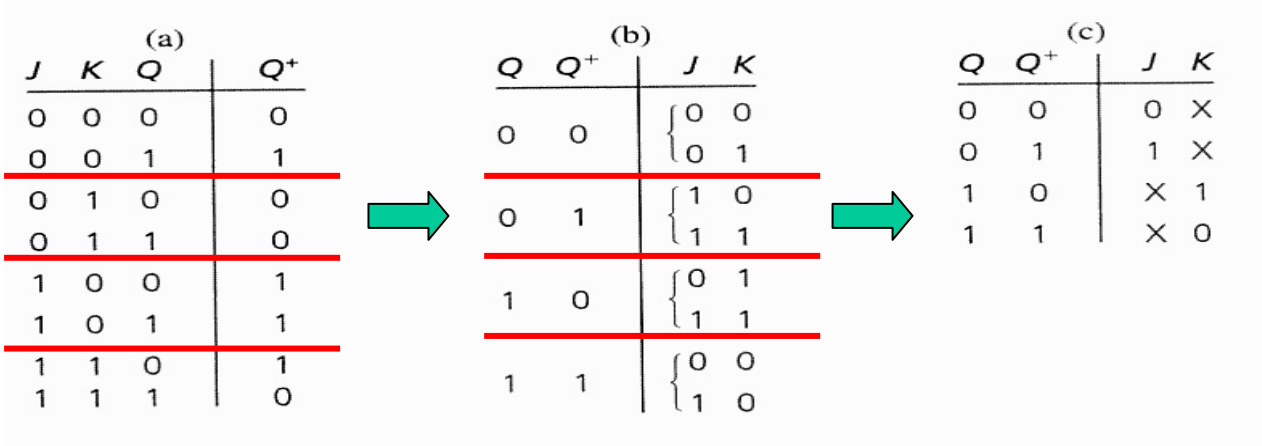


From setup table



Using J-K

- J-K flip-flop inputs



(a)

J	K	Q	Q^+
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

(b)

Q	Q^+	J	K
0	0	0	0
		0	1
0	1	1	0
		1	1
1	0	0	1
		1	1
1	1	0	0
		1	0

(c)

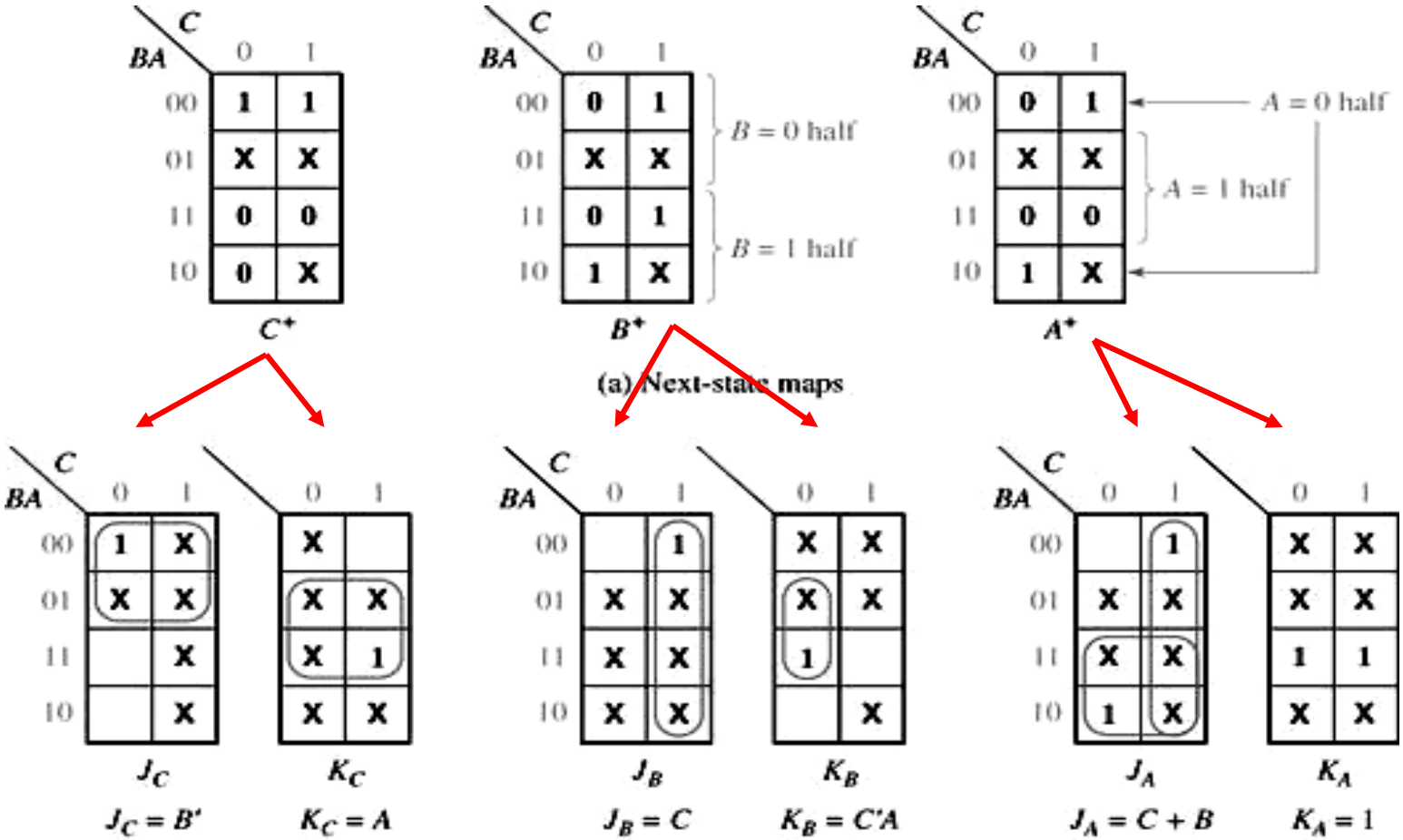
Q	Q^+	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Set J,k

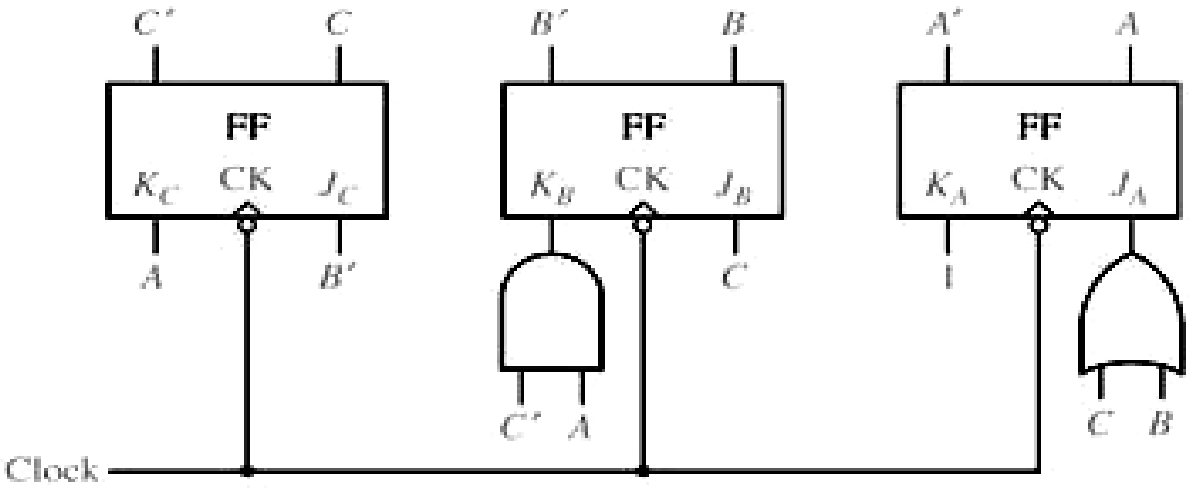
C	B	A	C^+	B^+	A^+	J_C	K_C	J_B	K_B	J_A	K_A
0	0	0	1	0	0	1	X	0	X	0	X
0	0	1	–	–	–	X	X	X	X	X	X
0	1	0	0	1	1	0	X	X	0	1	X
0	1	1	0	0	0	0	X	X	1	X	1
1	0	0	1	1	1	X	0	1	X	1	X
1	0	1	–	–	–	X	X	X	X	X	X
1	1	0	–	–	–	X	X	X	X	X	X
1	1	1	0	1	0	X	1	X	0	X	1

JK Excitation Table

Using J K


C
BA
0 1
00 1 1
01 X X
11 0 0
10 0 X
C+
B = 0 half
B = 1 half
B+
A = 0 half
A = 1 half
A+
(a) Next-state maps
JC = B'
KC = A
JB = C
KB = C'A
JA = C + B
KA = 1
(b) J-K flip-flop input equations

Implementation of JK-based Counter



(c) Logic circuit (omitting the feedback lines)

Outline

- 12.1 Register and Register Transfers
- 12.2 Shift Register
- 12.3 Design of Binary Counters
- 12.4 Counters for Other Sequences
  Counter Design Using D Flip-Flops
- 12.5 Counter Design Using S-R and J-K
- 12.6 Derivation of Flip-Flop Input Equations

Derivation of Flip-Flop Input Equations

- Determine the F/F input equations from the ***Next-State Equations***

Type of Flip-Flop	Input	$Q = 0$		$Q = 1$		Rules for Forming Input Map From Next-State Map*	
		$Q^+ = 0$	$Q^+ = 1$	$Q^+ = 0$	$Q^+ = 1$	$Q = 0$ Half of Map	$Q = 1$ Half of Map
Delay	D	0	1	0	1	no change	no change
Trigger	T	0	1	1	0	no change	complement
Set-Reset	S	0	1	0	X	no change	replace 1's with X's**
	R	X	0	1	0	replace 0's with X's**	complement
J-K	J	0	1	X	X	no change	fill in with X's
	K	X	X	1	0	fill in with X's	complement

Important Tables

Q Q^+	D
0 0	0
0 1	1
1 0	0
1 1	1

Q Q^+	T
0 0	0
0 1	1
1 0	1
1 1	0

Q Q^+	S R
0 0	0 X
0 1	1 0
1 0	0 1
1 1	X 0

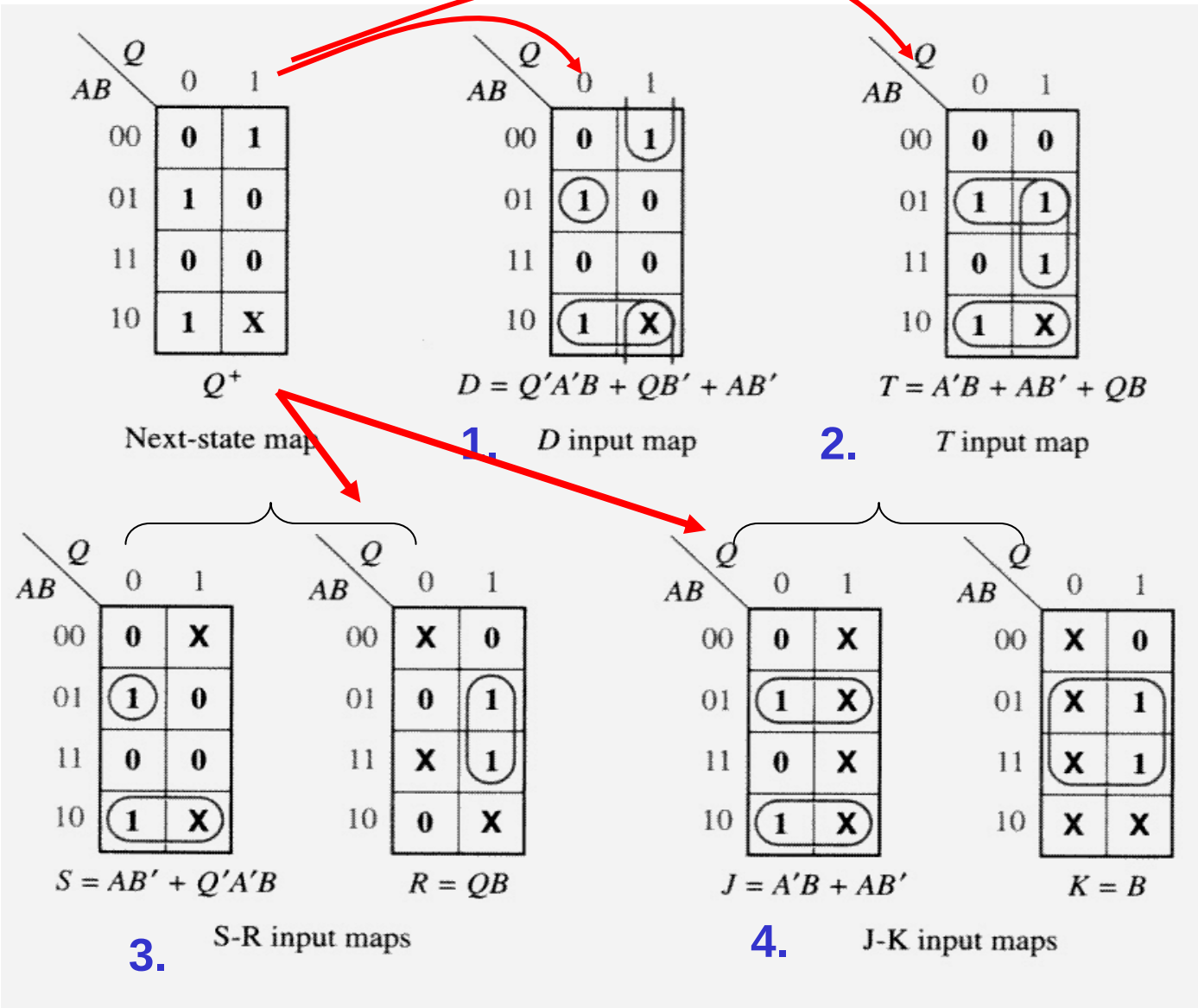
Q Q^+	J K
0 0	0 X
0 1	1 X
1 0	X 1
1 1	X 0

For Low-cost implement

Example:

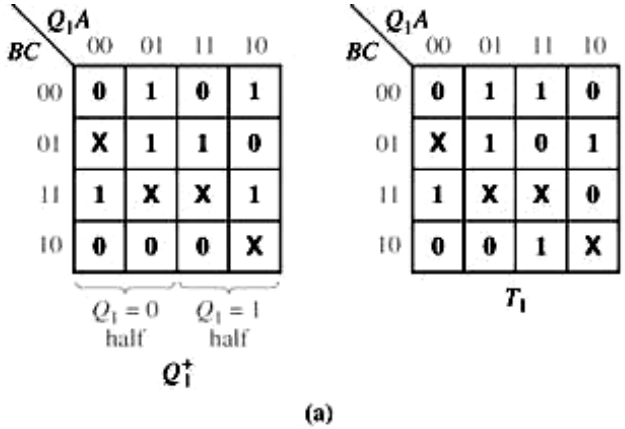
Table 12-9

Input: A,B,Q



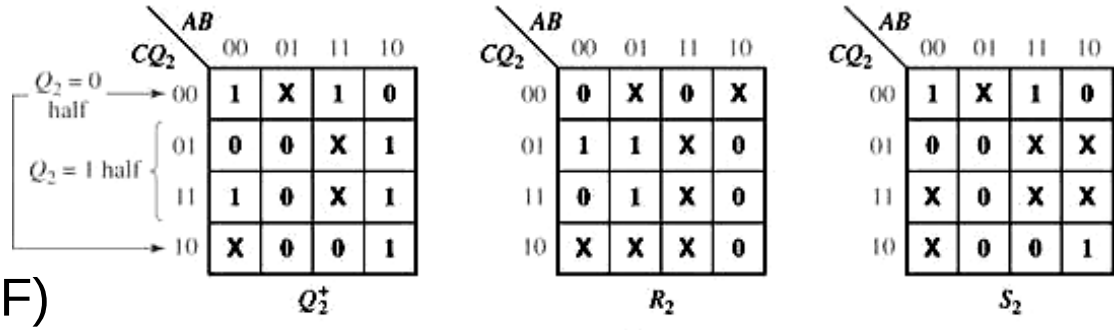
Derivation of Flip-Flop Input Equations Using 4-Variable Maps

Q1 (T F/F)



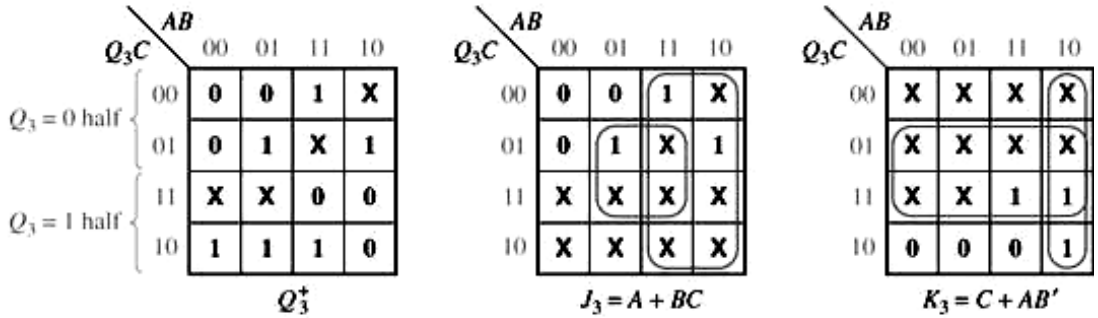
(a)

Q2 (SR F/F)



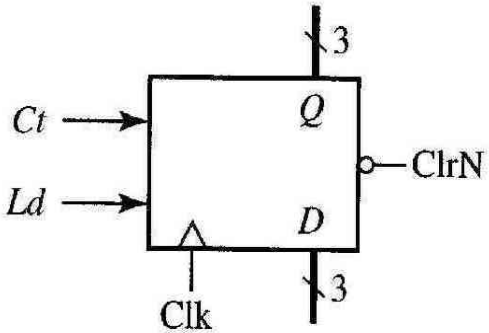
(b)

Q3 (JK F/F)



(c)

Pre settable Counter



(a)

ClrN	Ld	Ct	C^+	B^+	A^+	
0	X	X	0	0	0	
1	1	X	D_C	D_B	D_A	(load)
1	0	0	C	B	A	(no change)
1	0	1	Present state + 1			

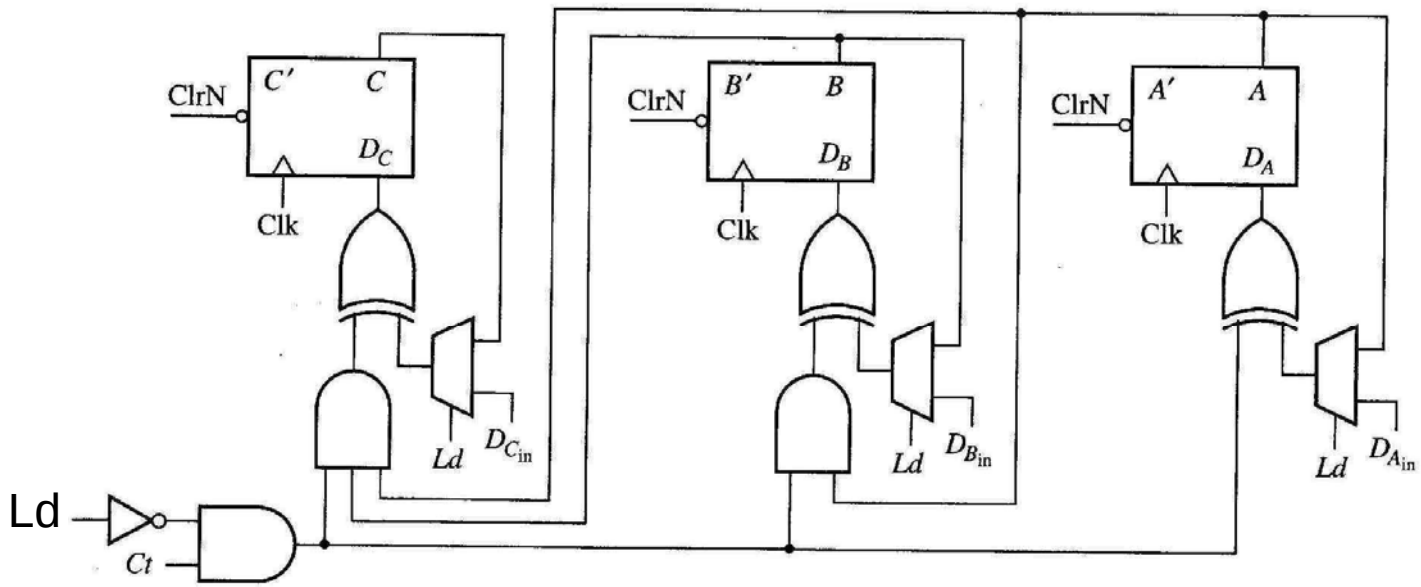
(b)

Pre-settable Counter

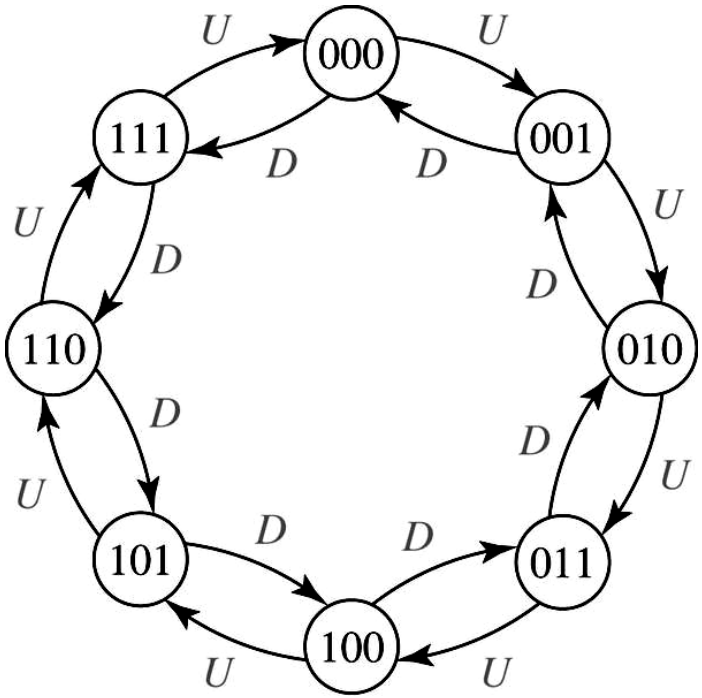
- Ld = Load
- Ct = Count
- ClrN = Clear

The next-state equations for the counter of Figure 12-20 are

$$A^+ = D_A = (Ld' \cdot A + Ld \cdot D_{Ain}) \oplus Ld' \cdot Ct$$
$$B^+ = D_B = (Ld' \cdot B + Ld \cdot D_{Bin}) \oplus Ld' \cdot Ct \cdot A$$
$$C^+ = D_C = (Ld' \cdot C + Ld \cdot D_{Cin}) \oplus Ld' \cdot Ct \cdot B \cdot A$$



Up down Counter



U = 1, D = 0 Up Counter

U = 0, D = 1 Down Counter

CBA	$C^+B^+A^+$	
	U	D
000	001	111
001	010	000
010	011	001
011	100	010
100	101	011
101	110	100
110	111	101
111	000	110

Up down Counter

- The up-down can be implemented using D flip-flops and gates, as shown in Figure 12-18. The corresponding logic equations are

$$D_A = A^+ = A \oplus (U + D)$$

$$D_B = B^+ = B \oplus (UA + DA)$$

$$D_C = C^+ = C \oplus (UBA + DB'A')$$

- When U=1 and D=0, these equations reduce to equations for a binary up counter (Equations(12-2)).
- When U=0 and D=1, these equations reduce to

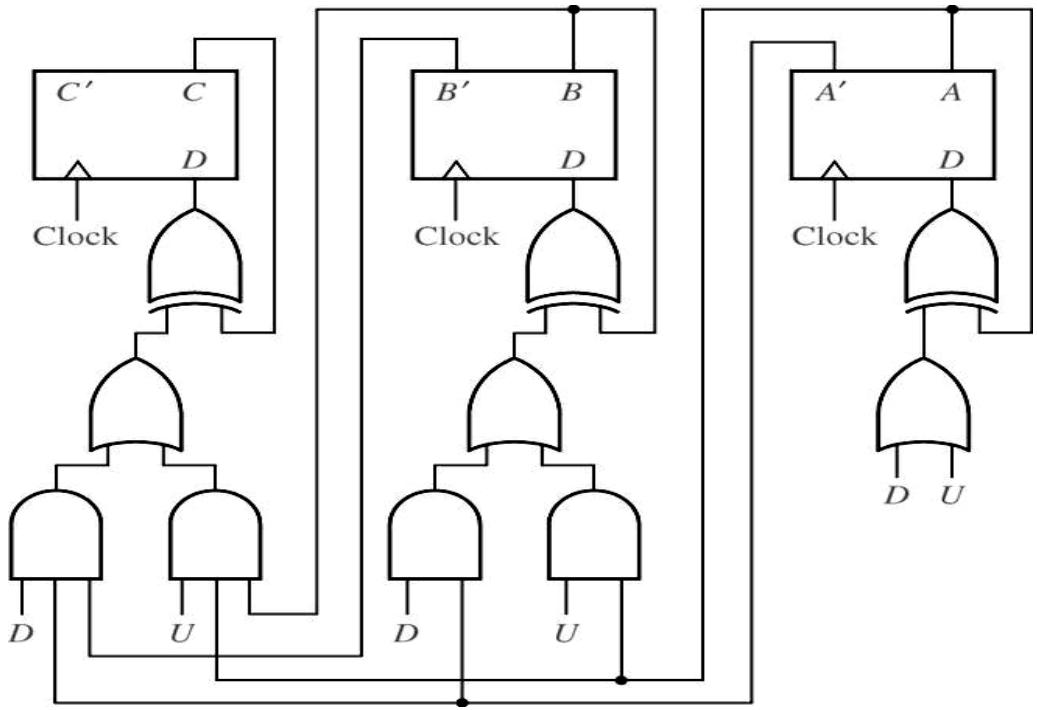
$$D_A = A^+ = A \oplus 1 = A' \quad (\text{A changes state every clock cycle})$$

$$D_B = B^+ = B \oplus A' \quad (\text{B changes state when A} = 0)$$

$$D_C = C^+ = C \oplus B'A' \quad (\text{C changes state when B} = \text{A} = 0)$$

Up-down Counter

- $D_A = A^+ = A \oplus (U + D)$
- $D_B = B^+ = B \oplus (UA + DA')$
- $D_C = C^+ = C \oplus (UBA + DB'A')$



Summary

- For a counter of ***N*** states, we need ***Log_2(N)*** flip-flops (F/F) to record the state.
- Watch for Excitation equations for T F/F, SR F/F, and JK F/F in your derivation. Usually, JF F/F leads to low-cost implementations (but with more efforts).
- Watch for unknown states in the operations of the counter. Usually, we need to preset/reset the states for normal operations.