Unit 12. Registers & Counters

Spring 2015

School of Electrical Engineering

Prof. Jong-Myon Kim

Objectives – To Learn

1. Explain the operation of registers. Understand how to transfer data between registers using tri-state bus.

2. Explain the shift register operation, how to build them and analyze operation. Construct a timing diagram for a shift register.

3. Explain the operation of binary counters, how to build them using F/F and gates and analyze operation.

4. Given the present state and desired next state of F/F, determine the required F/F inputs.

5. Given the desired counting sequence for a counter, derive F/F input equations.

6. Explain the procedures used for deriving F/F input equation.

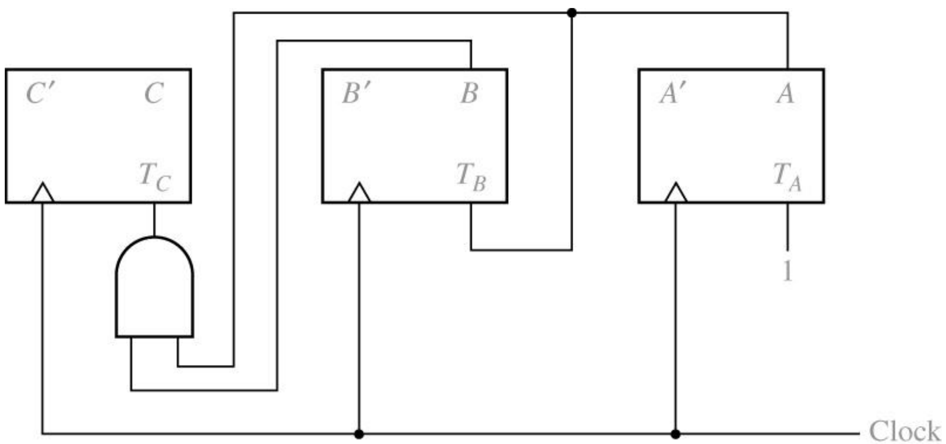
7. Construct a timing diagram for a counter by tracing signals through the circuit.

Counters

- Counters
  - Simplest sequential networks
  - usually constructed from 2 or more F/F's
- Synchronous counter
  - operation of F/F's are synchronized by common input pulse
- Asynchronous counter
  - example : ripple counter – the state change of current F/F affects the next F/F

Sync Binary Counter using T F/F

- Design a binary counter using 3 T F/F's to count clock pulses
- Counting sequence:
  - CBA : 000, 001, 010, 011, 100, 101, 110, 111, 000
- 1. Design a counter by inspection of the counting sequence
- Observe :
  - A changes state every time a pulse is received
  - B changes state every time a pulse is received only if A=1
  - C changes state every time a pulse is received only if B=A=1
- solution as in Fig 12-13
- Verify the operation by tracing signals



Redesign using a State Table

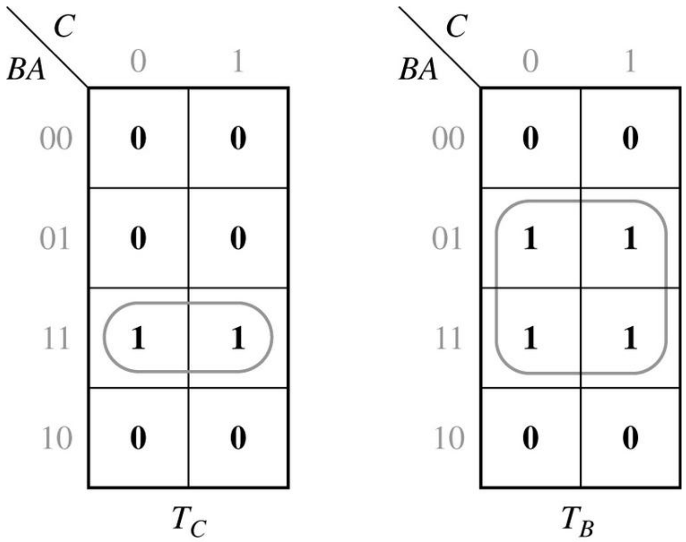
State Table for Binary Counter (Table 12-2)

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

Redesign using a State Table

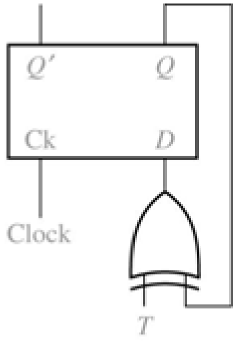
- How to create the table in the previous slide?
  - If A and A+ differs, the status of Flip-Flop A is changed. So, T_A=1.

    Variable B and C are calculated from similar way of A
- Make a K-map from a state table : Fig 12-14(below)
- Derive F/F input equations
  - T_C=AB, T_B=A, T_A=1
- same result as previous

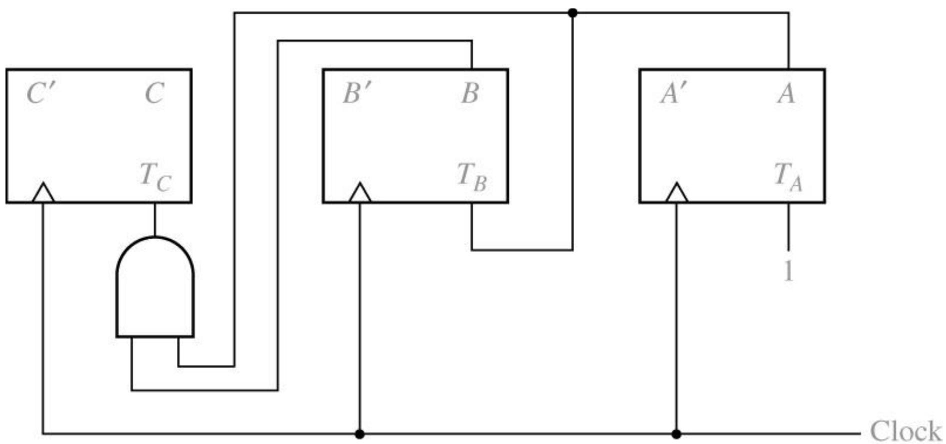


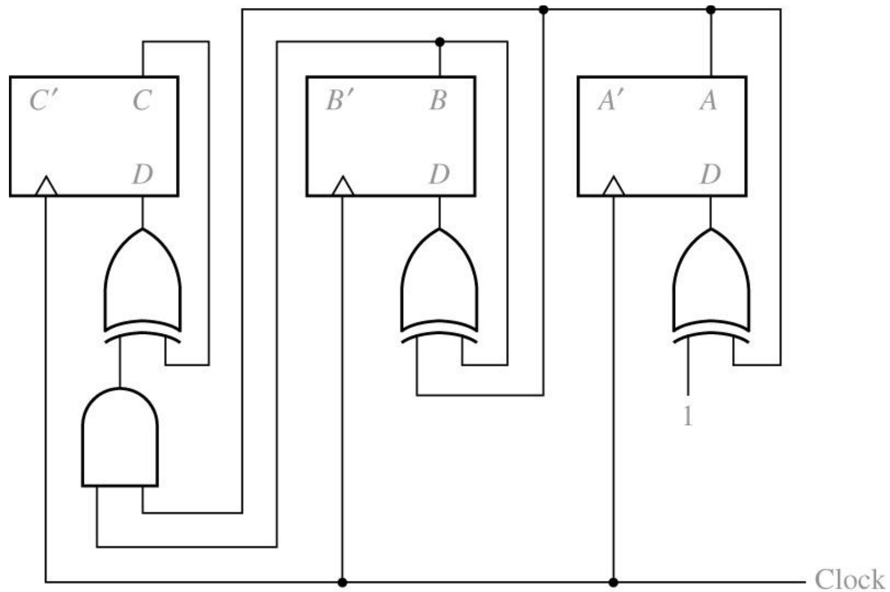
Design of Binary Counters

Binary Counter with D Flip-Flops using conversion of D to T (Fig 11-24(b)



(b) Conversion of *D* to *T*





Redesign of Binary Counter

The D input equations derived from the maps are

$$D_A = A^+ = A'$$

$$D_B = B^+ = BA' + B'A = B \oplus A$$

$$D_C = C^+ = C'BA + CB' + CA' = C'BA + C(BA)' = C \oplus BA$$

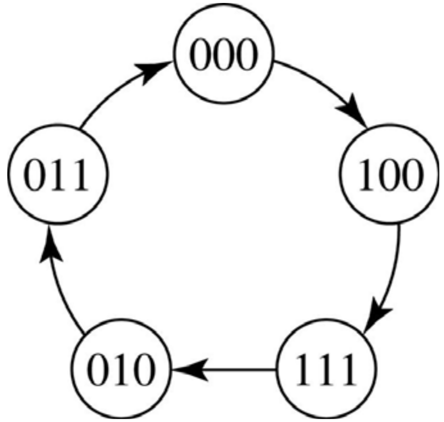
- Compare the above results with Figure 12-15
- Same Diagram?
- why?? ☞ A xor 1 = A'

Counters for Other Sequences

The sequence of states of a counter is not in straight binary order.

State Graph for Counter

(Figure 12-21)



State Table for Figure 21.21

(Table 12-3)

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

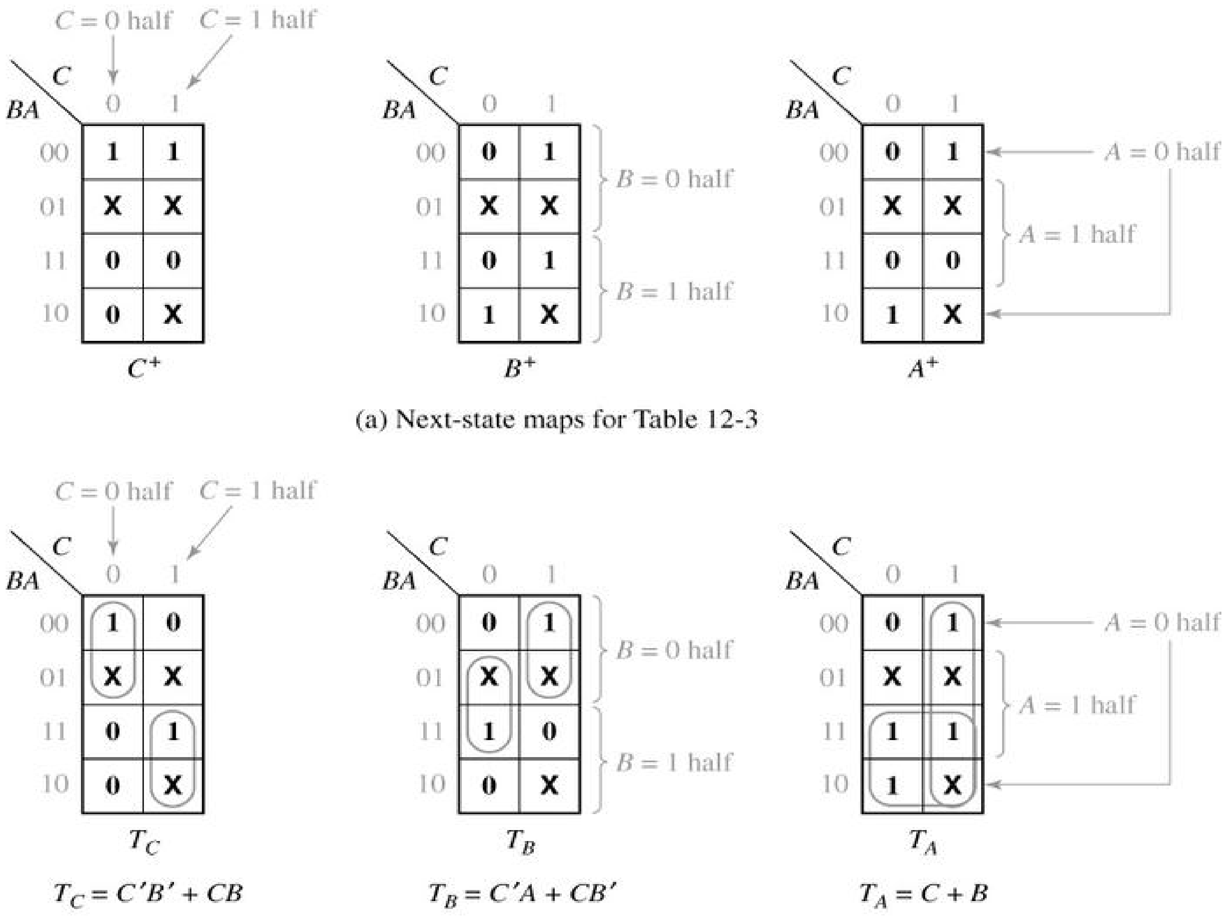
Counters for Other Sequences

- State graph
  - ❖ 임의의 순서를 나타냄
- State table
  - ❖ note : unspecified next state
  - ❖ 이전 예제처럼 state table을 완성하거나(as in Table 12-2)
  - ❖ next-state map을 먼저 plot한 후 이 map으로부터 T_A, T_B, T_C 유도
    - ❖ 편리한 방법

Counters for Other Sequences

The next-state maps in Figure 12-22(a) are easily plotted from inspection of Table 12-3 → **Use T-F/F**

Figure 12-22



$T_C = C'B' + CB$ $T_B = C'A + CB'$ $T_A = C + B$

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

(b) Derivation of T inputs

Counters for Other Sequences

- make next state maps – how?
- derive T input maps from next-state map
- T=1 whenever Q^+ is different from Q
- From the below table

  $T=Q^+$ when Q=0

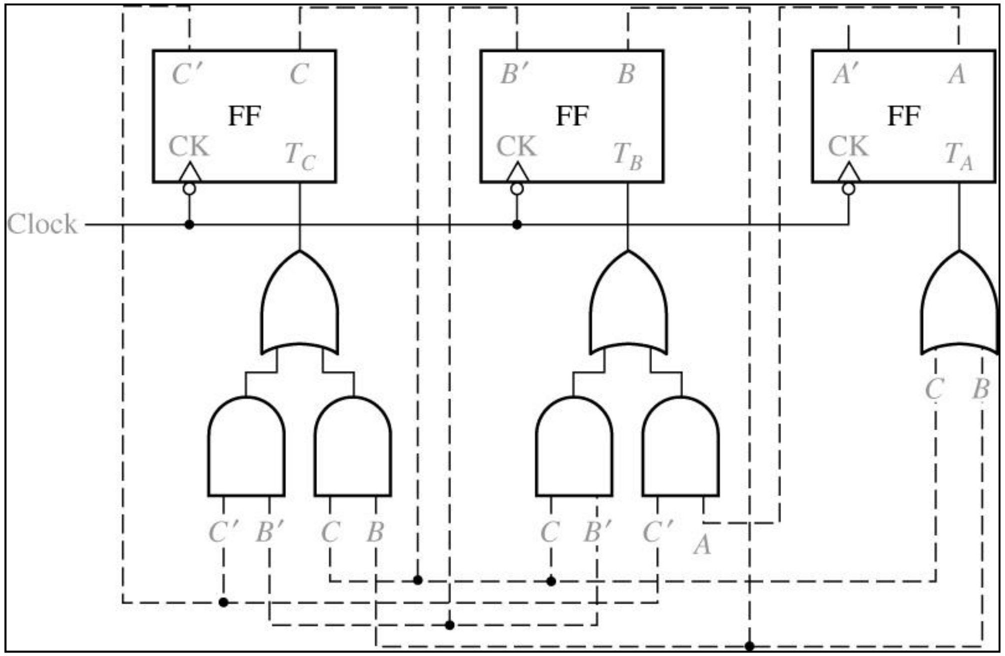
  $T=(Q^+)'$ when Q=1
- Therefore, Copy Q=0 half, complement Q=1 half

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

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

Counters for Other Sequences

Counter Using T Flip-Flops (Figure 12-23)



Counters for Other Sequences

Timing Diagram for Figure 12-23

(Figure 12-24)

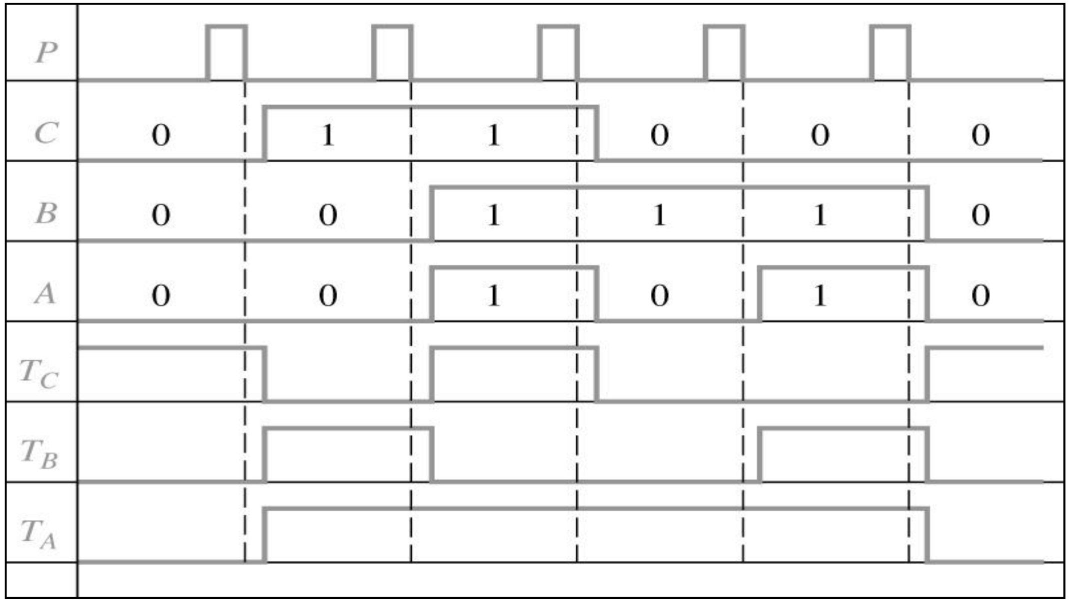
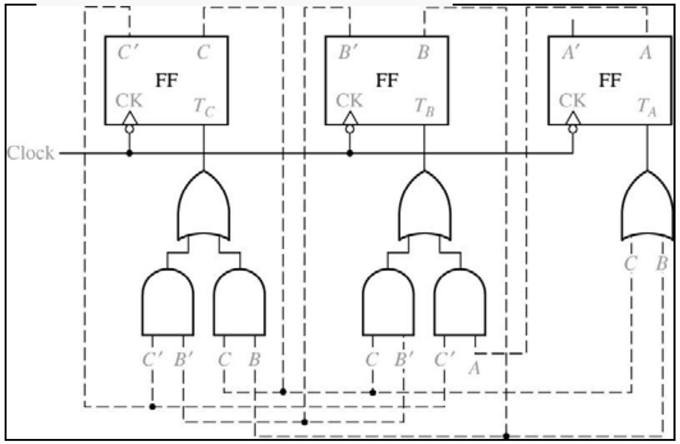


Figure 12-23



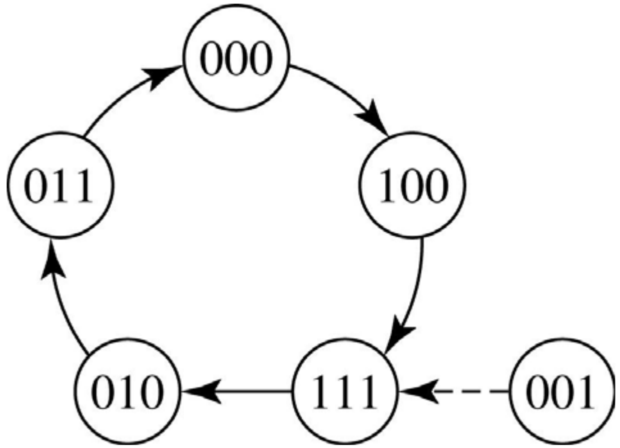
State Graph for Counter

(Figure 12-25)

C=0, B=0, A=1 인경우

Tc=Tb=1, Ta=0

=>From Fig. 12-23



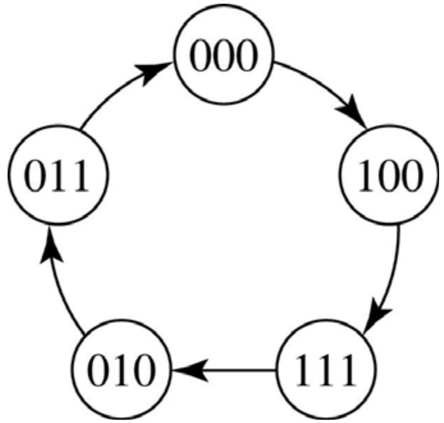
Counters for Other Sequences

Procedure to design a counter using T F/F's:

1. Form a state table which gives the next F/F states for each combination of present F/F states.
2. Plot the next-state maps from the table.
3. Plot a T input map for each F/F .
4. Find the T input equations from the maps and realize the circuit.

Counter Design using D F/F

Fig 12-21



- Counting sequence : same as in Fig 12-21
- $Q^+=D$ → D input map is identical with next-state map
- $D_C=C^+=B'$
  $D_B=B^+=C+BA'$
  $D_A=A^+=CA'+BA'=A'(C+B)$
- Result : Fig 12-26

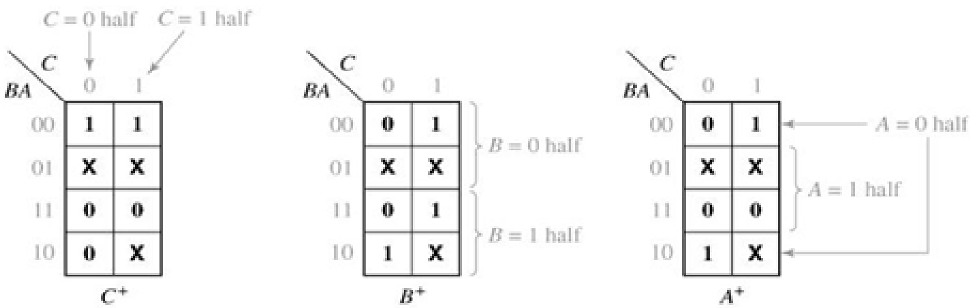
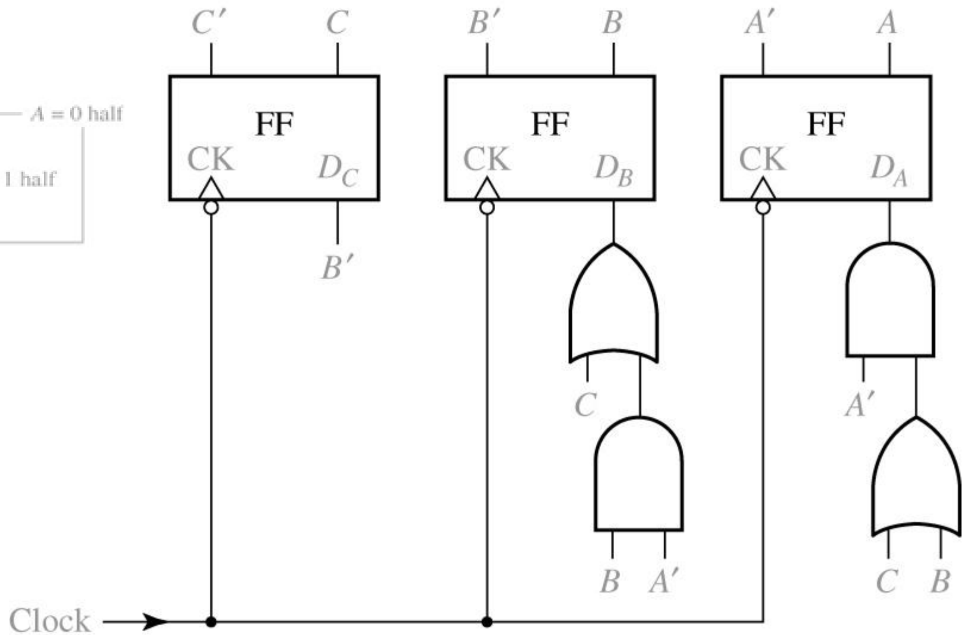


Fig 12-26

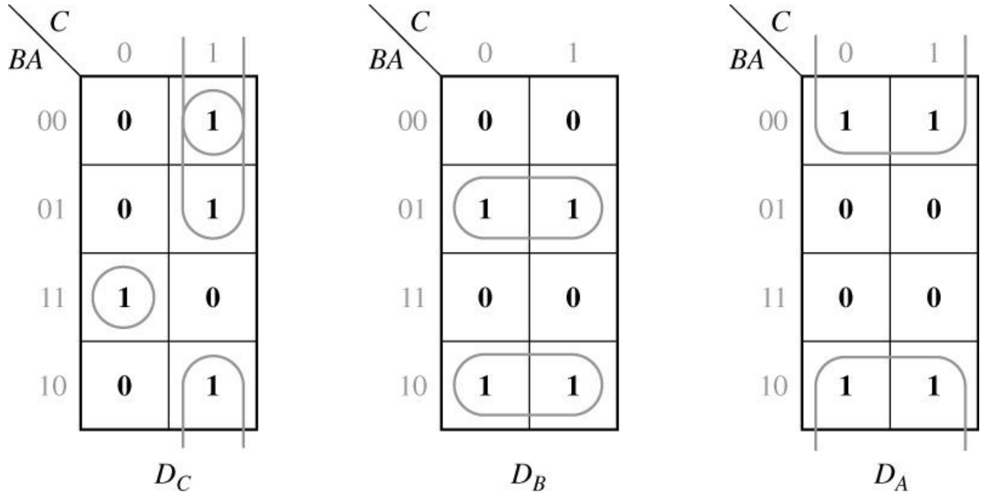


Redesign Binary Counter Using D F/F’ s

- 1st method : by converting F/F's ☞ Result Fig 12-15
- 2nd method:
  - State table (Table 12-2)
  - Next state maps
  - Flip-flop input maps
  - Flip-flop input equations
- For D flip-flops,
  next state map = flip-flop input map (since $Q^+=D$)

Karnaugh Maps for D Flip-Flops from Table 12-2 (Figure 12-16)

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



Counter Design using S-R and J-K F/Fs

S-R Flip-Flop Inputs (Table 12-5)

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	-
1	1	1	-

Inputs not allowed

(a)

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

(b)

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

(c)

Counter Design Using S-R F/Fs

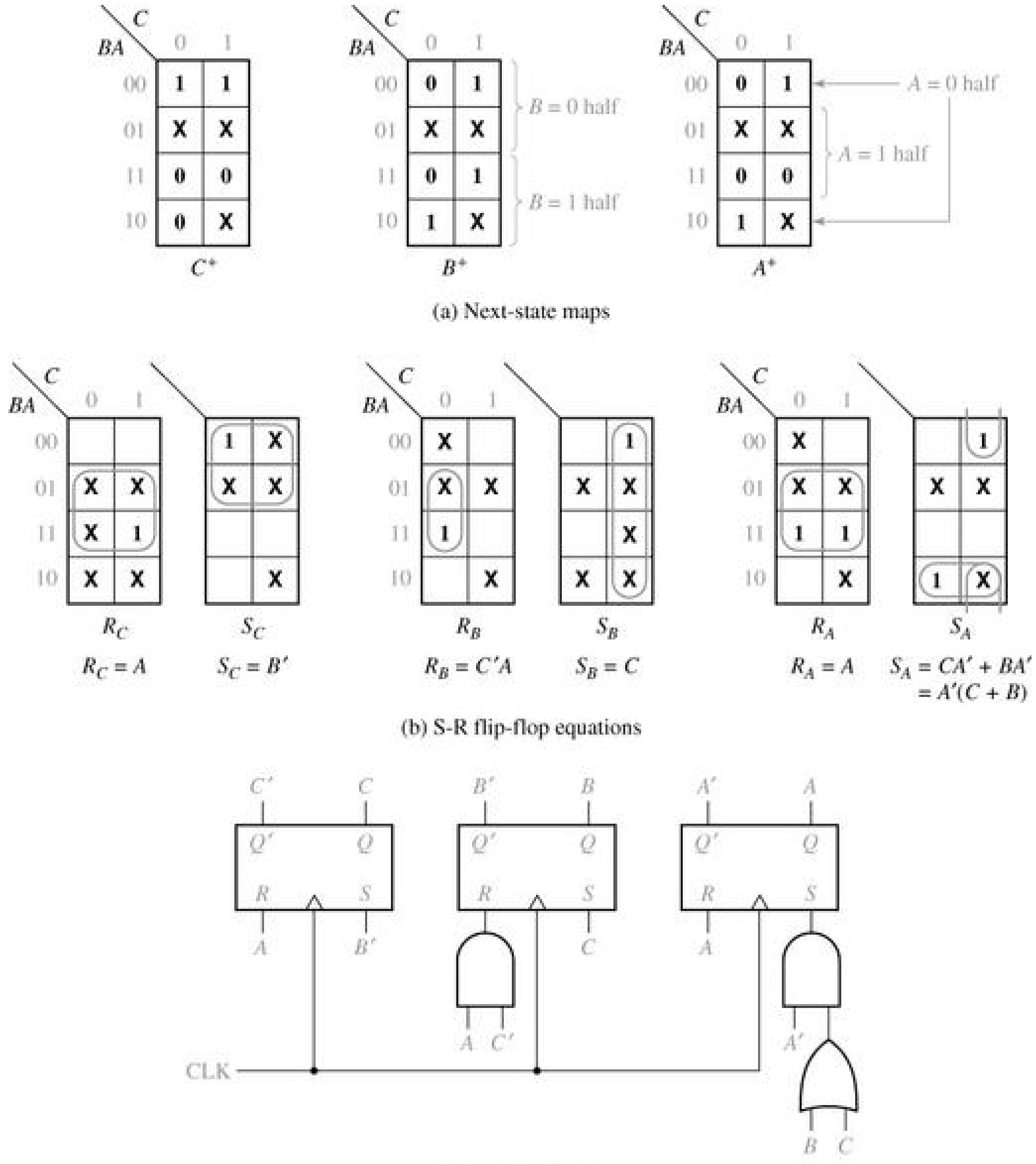
Q	Q^+	S	R
0	0	0	×
0	1	1	0
1	0	0	1
1	1	×	0

With columns added for the S and R flip-flop inputs (Table 12-6)

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	×	0	×
0	0	1	-	-	-	×	×	×	×	×	×
0	1	0	0	1	1	0	×	×	0	1	0
0	1	1	0	0	0	0	×	0	1	0	1
1	0	0	1	1	1	×	0	1	0	1	0
1	0	1	-	-	-	×	×	×	×	×	×
1	1	0	-	-	-	×	×	×	×	×	×
1	1	1	0	1	0	0	1	×	0	0	1

Counter Design using S-R Flip-Flops

- S
  - Q=0 half : copy
  - Q=1 half : replace 1 with X
- R
  - Q=0 half : replace 0 with X, fill 0
  - Q=1 half : complement



(a) Next-state maps
(b) S-R flip-flop equations
(c) Logic circuit

Counter Design Using J-K Flip-Flops

J-K Flip-Flop Inputs (Table 12-7)

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

(a)

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

(b)

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

(c)

Counter Design using J-K Flip-Flops

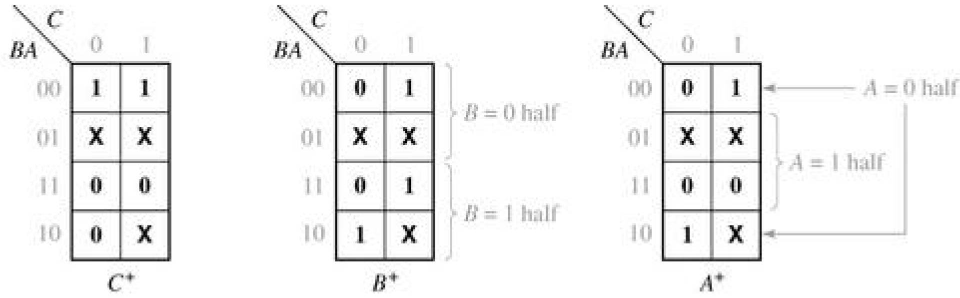
Q	Q^+	J	K
0	0	0	×
0	1	1	×
1	0	×	1
1	1	×	0

With columns added for the J and K flip-flop inputs (Table 12-8)

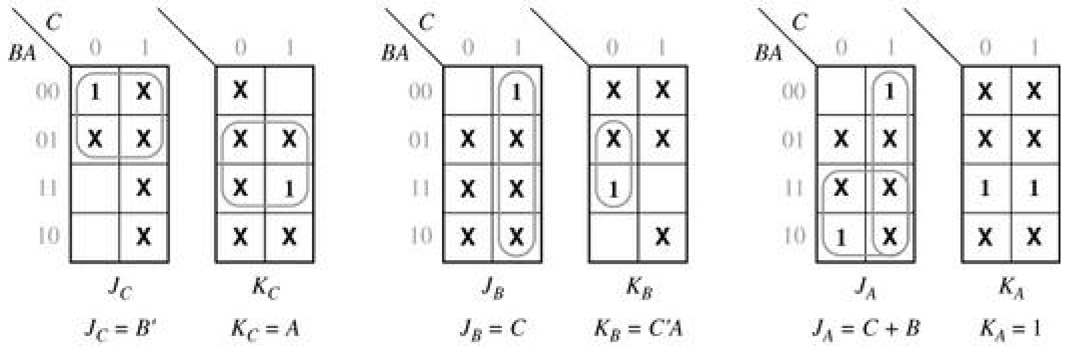
C	B	A	C^+	B^+	A^+	J_C	K_C	J_B	K_B	J_A	K_A
0	0	0	1	0	0	1	×	0	×	0	×
0	0	1	-	-	-	×	×	×	×	×	×
0	1	0	0	1	1	0	×	×	0	1	×
0	1	1	0	0	0	0	×	×	1	×	1
1	0	0	1	1	1	×	0	1	×	1	×
1	0	1	-	-	-	×	×	×	×	×	×
1	1	0	-	-	-	×	×	×	×	×	×
1	1	1	0	1	0	×	1	×	0	×	1

Counter Design using J-K Flip-Flops

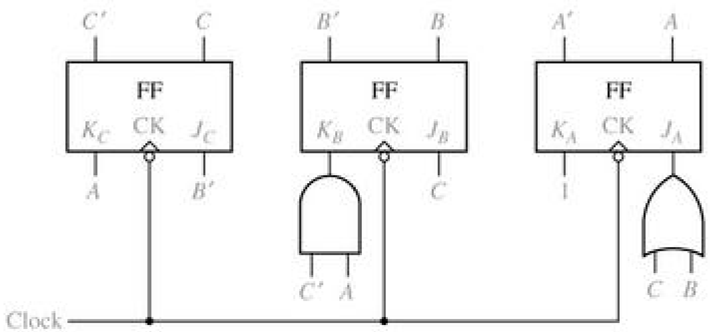
- J
  - Q=0 half : copy
  - Q=1 half : X
- K
  - Q=0 half : X
  - Q=1 half : complement



(a) Next-state maps



(b) J-K flip-flop input equations



(c) Logic circuit (omitting the feedback lines)

Counter of Figure 12-21

Using J-K Flip-Flops

(Figure 12-28)

Derivation of Flip-Flop Input Equations–Summary

Determination of Flip-Flop Input Equations from Next-State Equations Using Karnaugh Maps (Table 12-9)

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

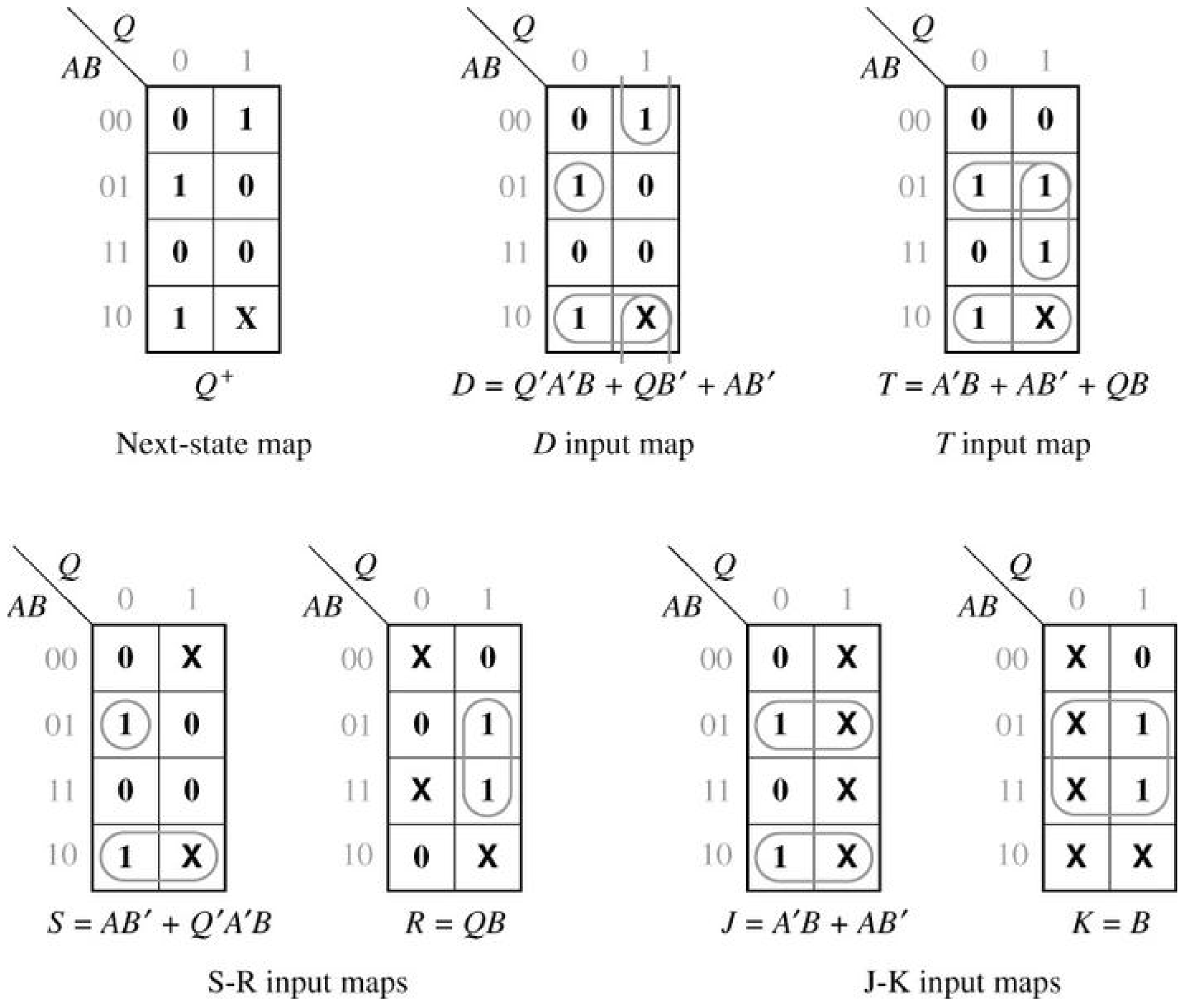
Q^+ means the next state of Q
X is a don't care
*Always copy X's from the next-state map onto the input maps first.
**Fill in the remaining squares with 0's.

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

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

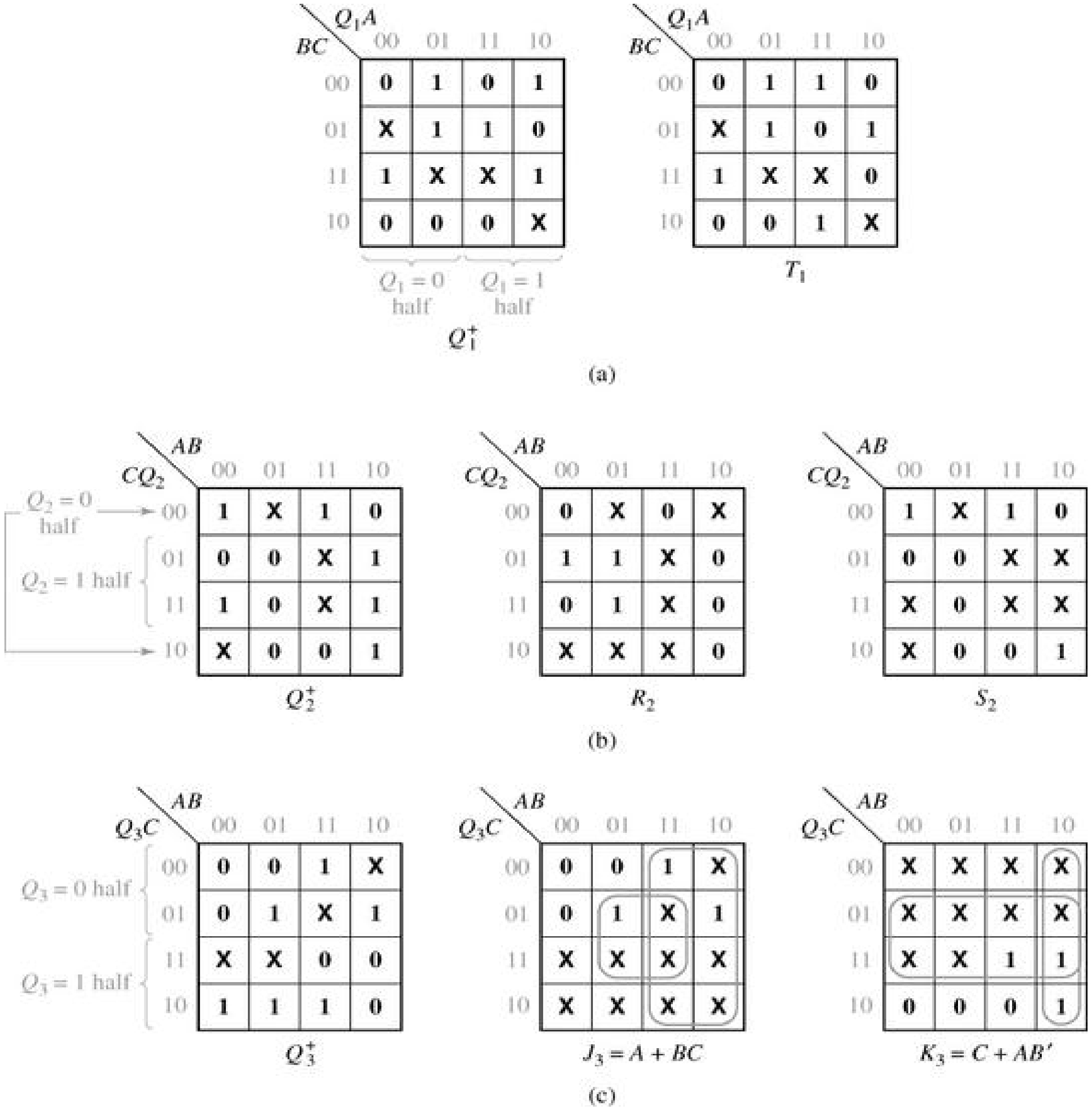
Table 12-9

- T
  - Q=0 half : copy
  - Q=1 half : complement
- S
  - Q=0 half : copy
  - Q=1 half : replace 1 with X
- R
  - Q=0 half : replace 0 with X, fill 0
  - Q=1 half : complement
- J
  - Q=0 half : copy
  - Q=1 half : X
- K
  - Q=0 half : X
  - Q=1 half : complement



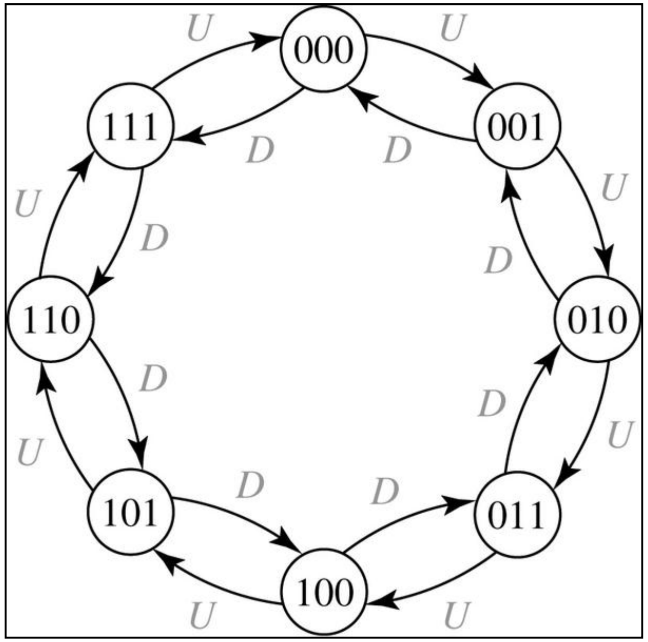
Derivation of Flip-Flop Input Equations

Derivation of Flip-Flop Input Equations Using 4-Variable Maps (Figure 12-29)



Analysis of Un-Down Counter

State Graph and Table for Up-Down counter (Figure 12-17)



When U=1, Up counting

When D=1, Down counting

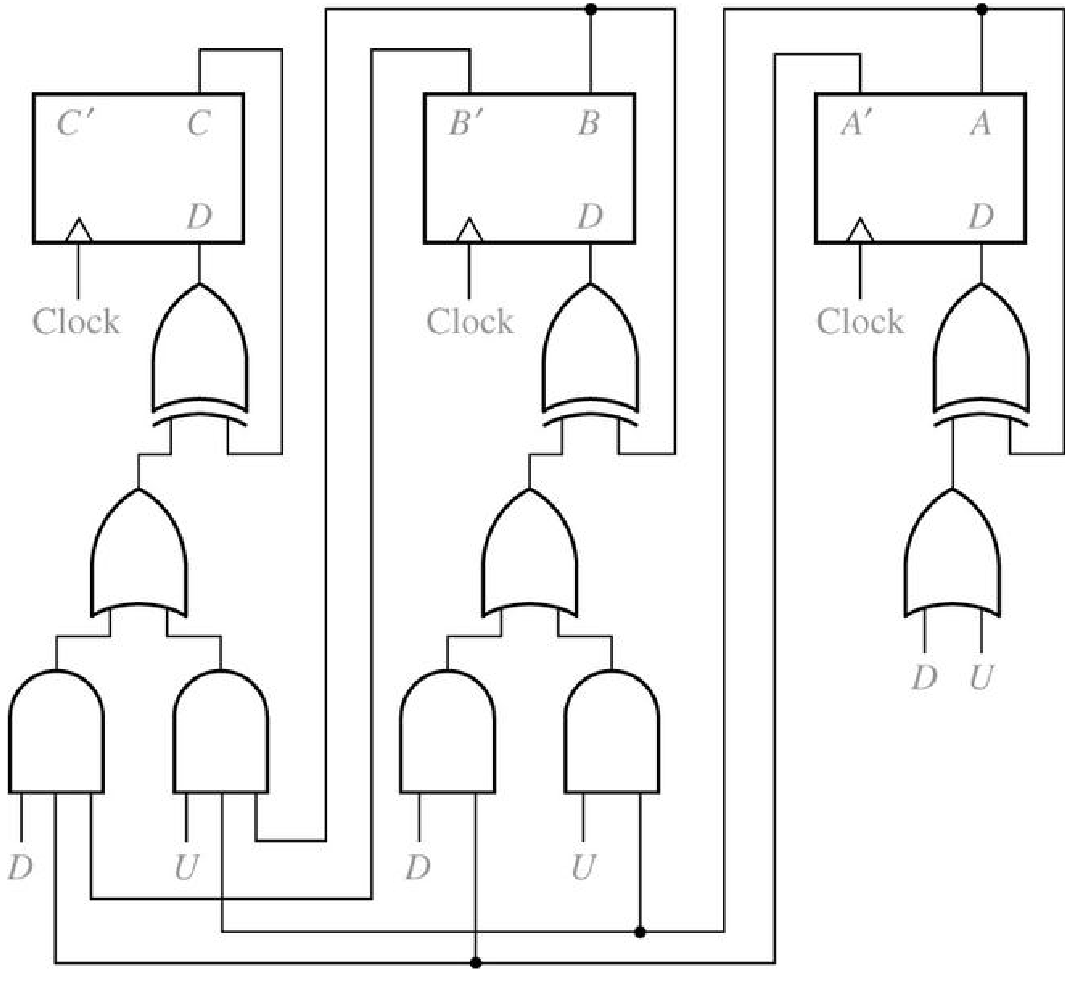
CBA	$C^+B^+A^+$ U	$C^+B^+A^+$ D
000	001	111
001	010	000
010	011	001
011	100	010
100	101	011
101	110	100
110	111	101
111	000	110

Up-Down Counter

The up-down counter can be implemented using D F/F and gate

Binary Up-Down Counter

(Figure 12-18)



Up-Down Counter

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, → Eq (12-2)

U=0 and D=1, these equations reduce to

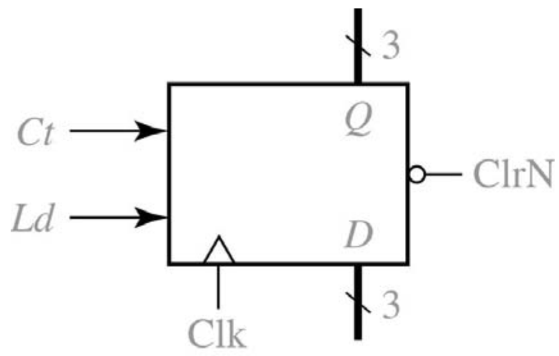
$$D_A = A^+ = A \oplus 1 = A' \quad (A \text{ change state every clock cycle})$$
$$D_B = B^+ = B \oplus A' \quad (B \text{ change state when } A = 0)$$
$$D_C = C^+ = C \oplus B'A' \quad (C \text{ change state when } B = A = 0)$$

Loadable Counter

Loadable Counter with Count Enable (Figure 12-19)

Loadable counter

(Figure 12-19(a))



(a)

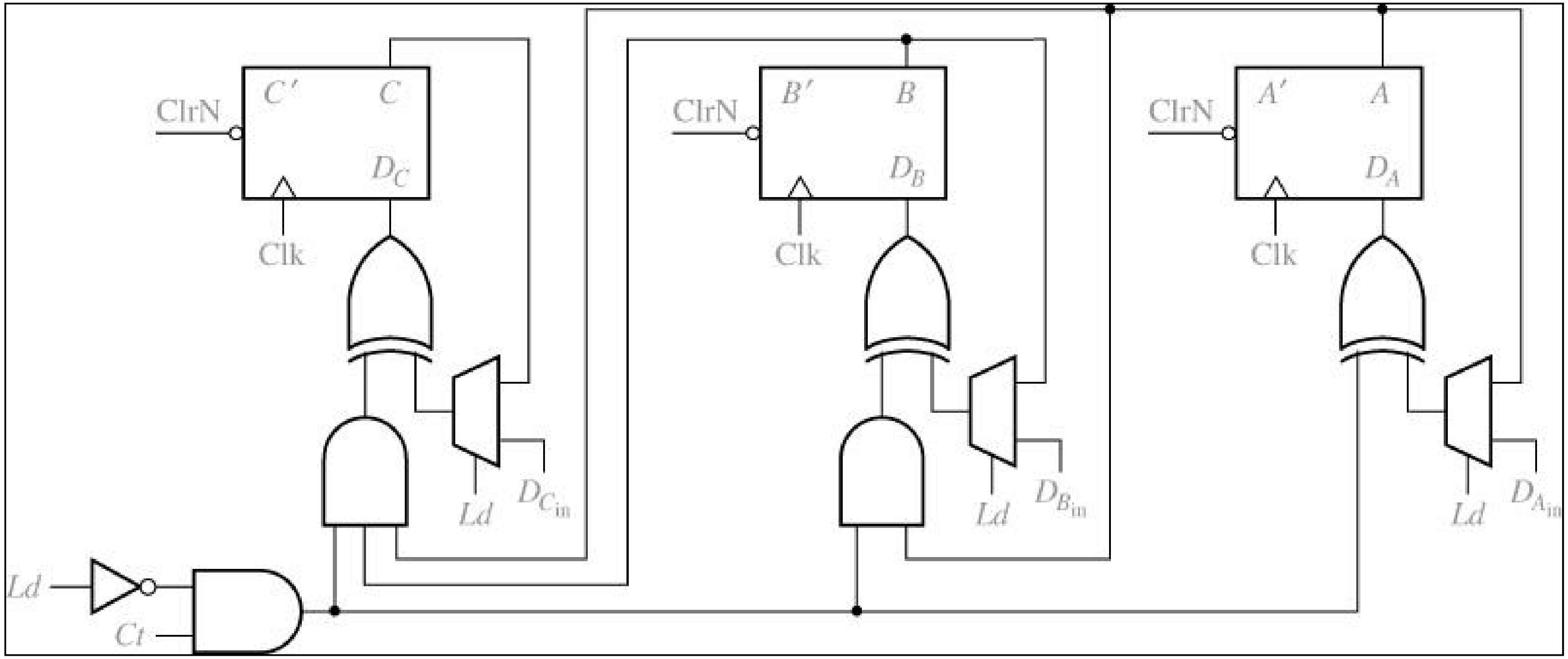
Summarizes the counter operation

(Figure 12-19(b))

$ClrN$	Ld	Ct	C^+	B^+	A^+	
0	×	×	0	0	0	
1	1	×	D_C	D_B	D_A	(*load*)
1	0	0	C	B	A	(*no change*)
1	0	1	present state + 1			

Loadable Counter

- Ld=1
  - MUX selects D input
- Ld=0 & Ct=1
  - MUX selects one of F/F output



Design of Binary Counters

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

$$A^{+} = D_{A} = (Ld' \cdot A + Ld \cdot D_{Ain}) \oplus Ld' \cdot Ct$$

$$B^{+} = D_{B} = (Ld' \cdot B + Ld \cdot D_{Bin}) \oplus Ld' \cdot Ct \cdot A$$

$$C^{+} = D_{C} = (Ld' \cdot C + Ld \cdot D_{Cin}) \oplus Ld' \cdot Ct \cdot B \cdot A$$

Registers and Register Transfers

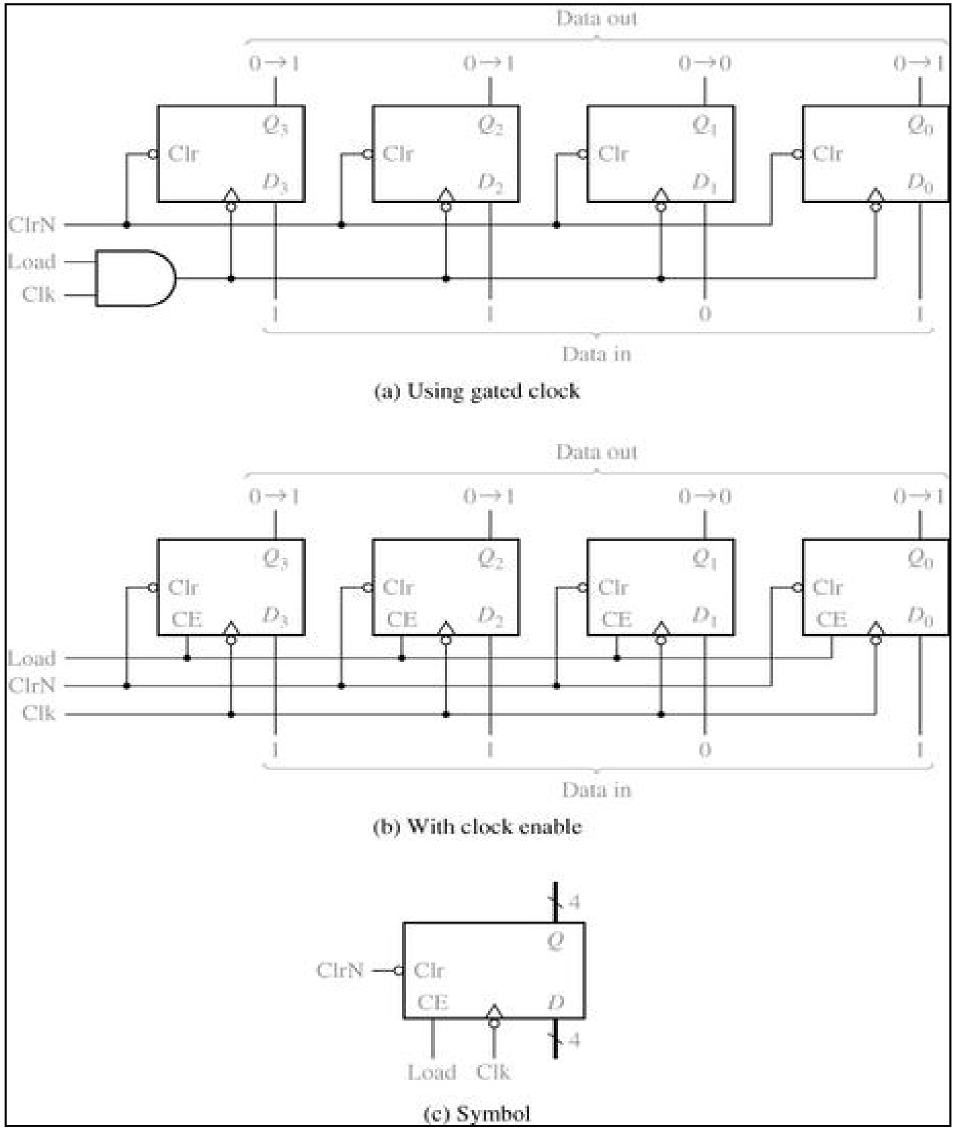
4-Bit D Flip-Flop Registers with Data, Load, Clear, and Clock inputs (Figure 12-1)

Grouped together D F/F

Using gated clock 12-1(a)

F/F with clock enable Figure 12-1(b)

Symbol for the 4-bit register using bus notation Figure 12-1(c)



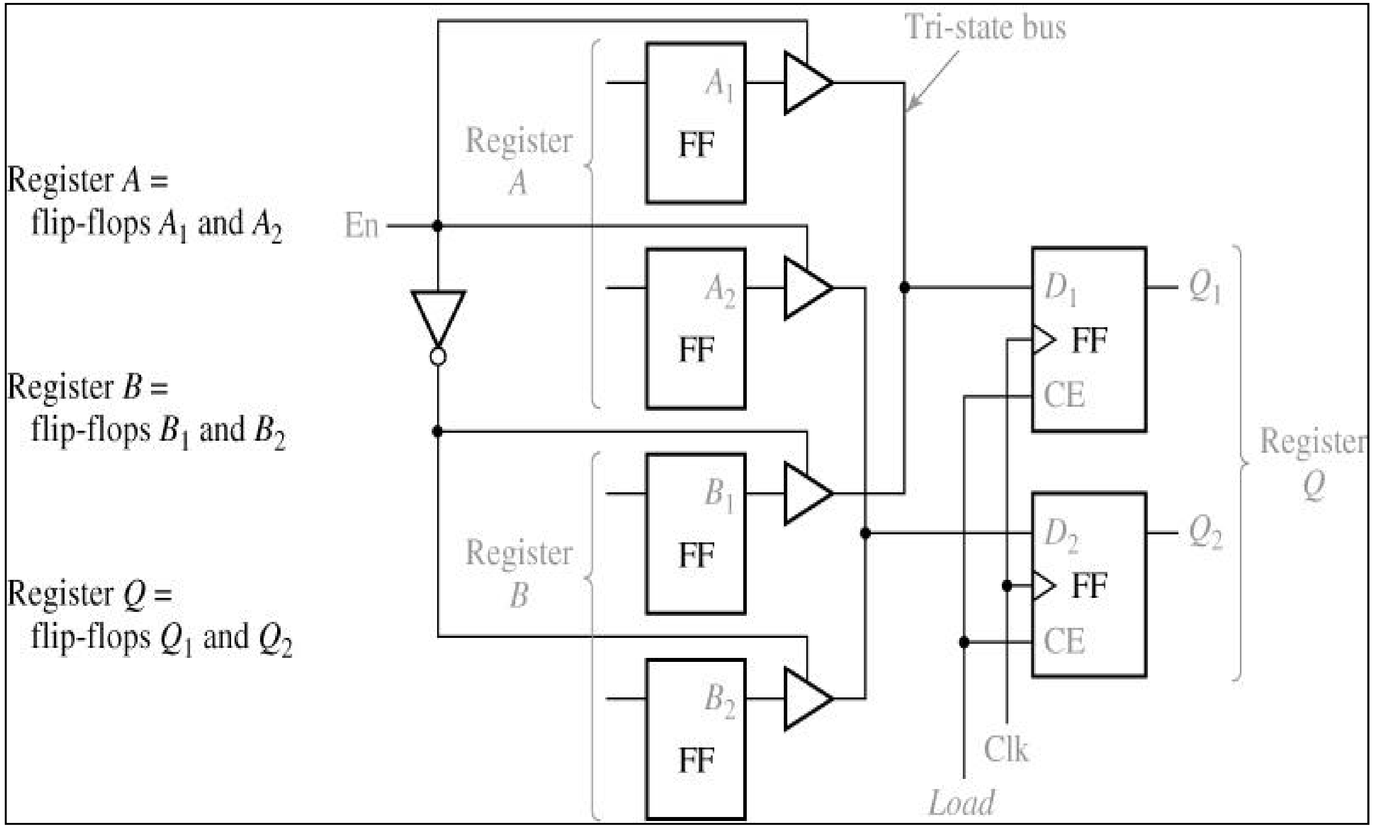
(a) Using gated clock

(b) With clock enable

(c) Symbol

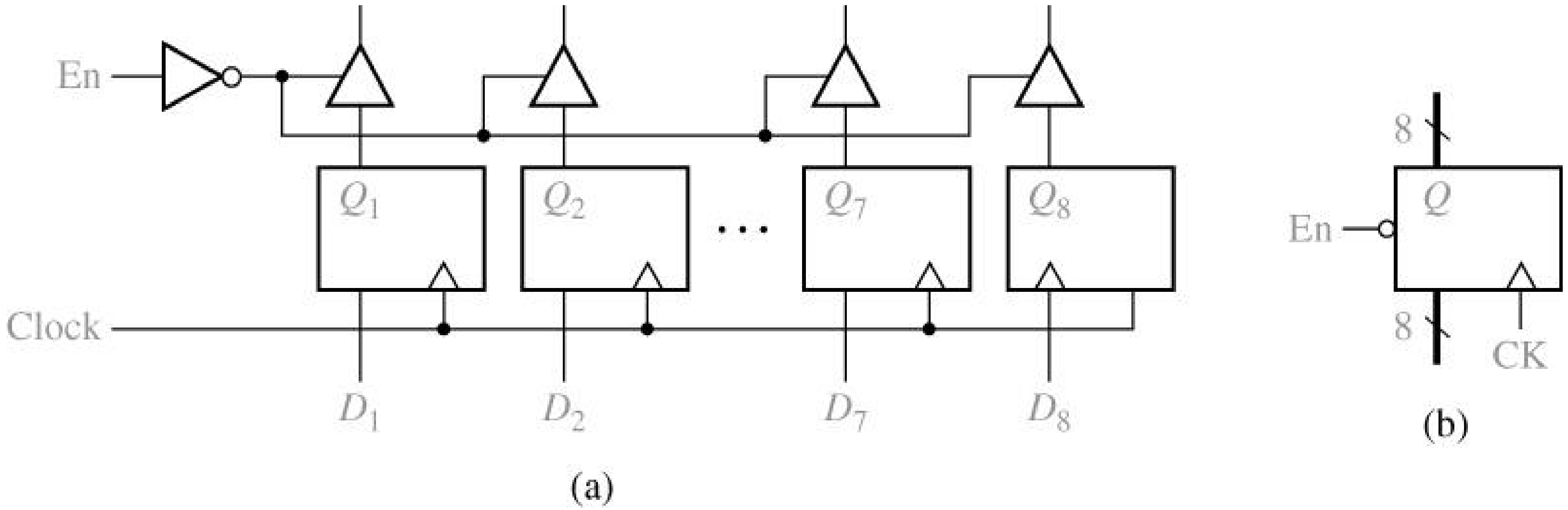
Registers and Register Transfers

Data Transfer Between Registers



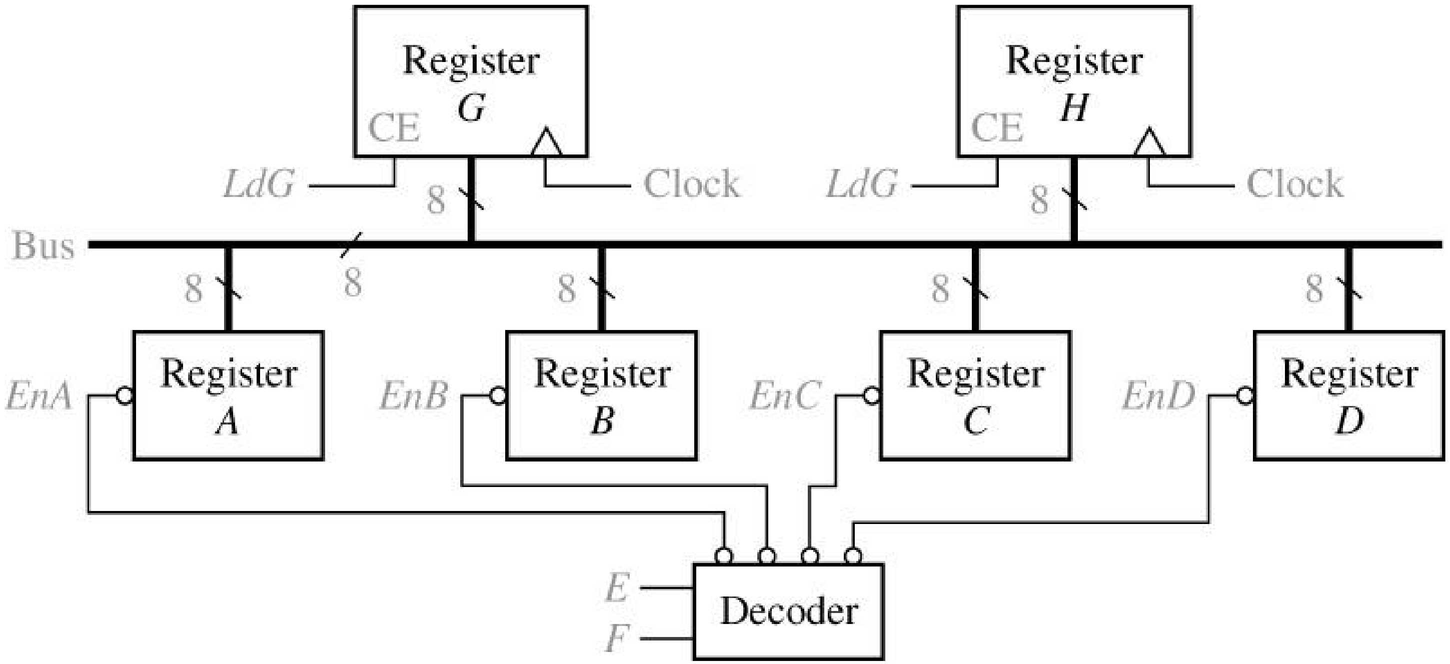
Registers and Register Transfers

Logic Diagram for 8-Bit Register with Tri-State Output



Registers and Register Transfers

Data Transfer Using a Tri-State Bus



Registers and Register Transfers

How data can be transferred?

The operation can be summarized as follows:

$$\text{If } EF = 00, A \text{ is stored in } G (\text{or } H).$$

$$\text{If } EF = 01, B \text{ is stored in } G (\text{or } H).$$

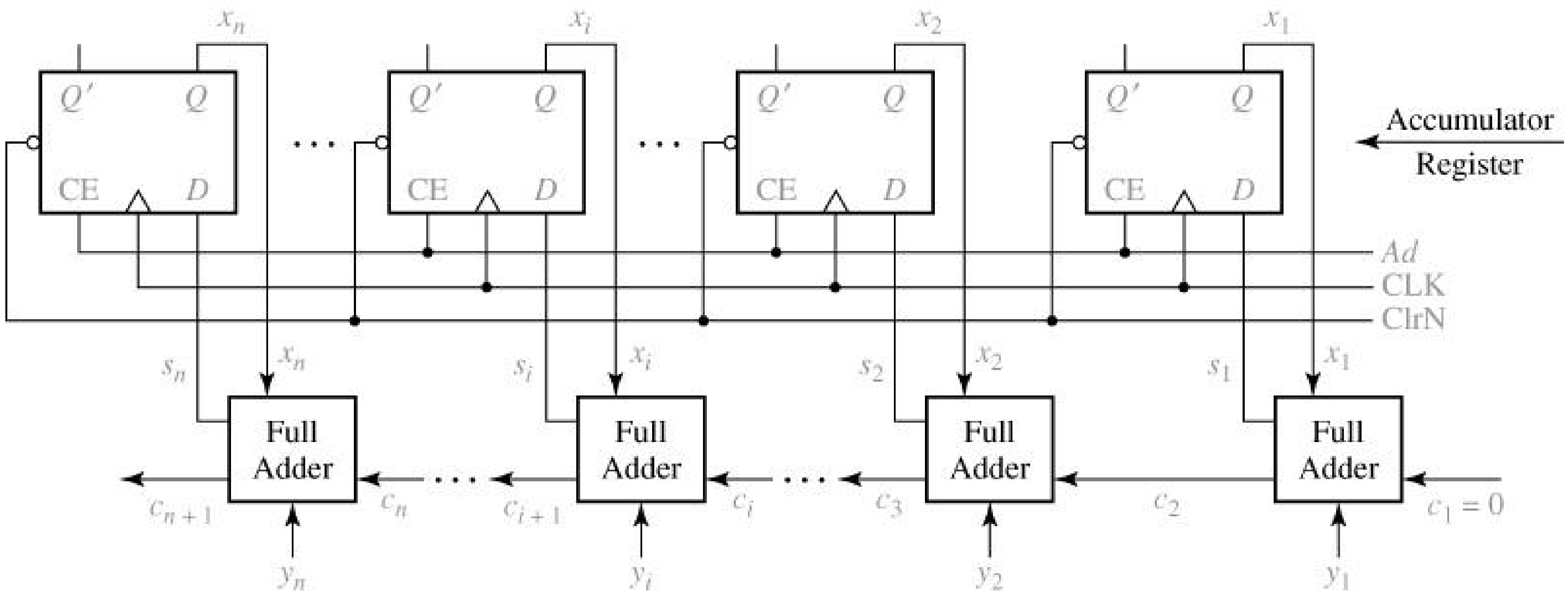
$$\text{If } EF = 10, C \text{ is stored in } G (\text{or } H).$$

$$\text{If } EF = 11, D \text{ is stored in } G (\text{or } H).$$

Registers and Register Transfers

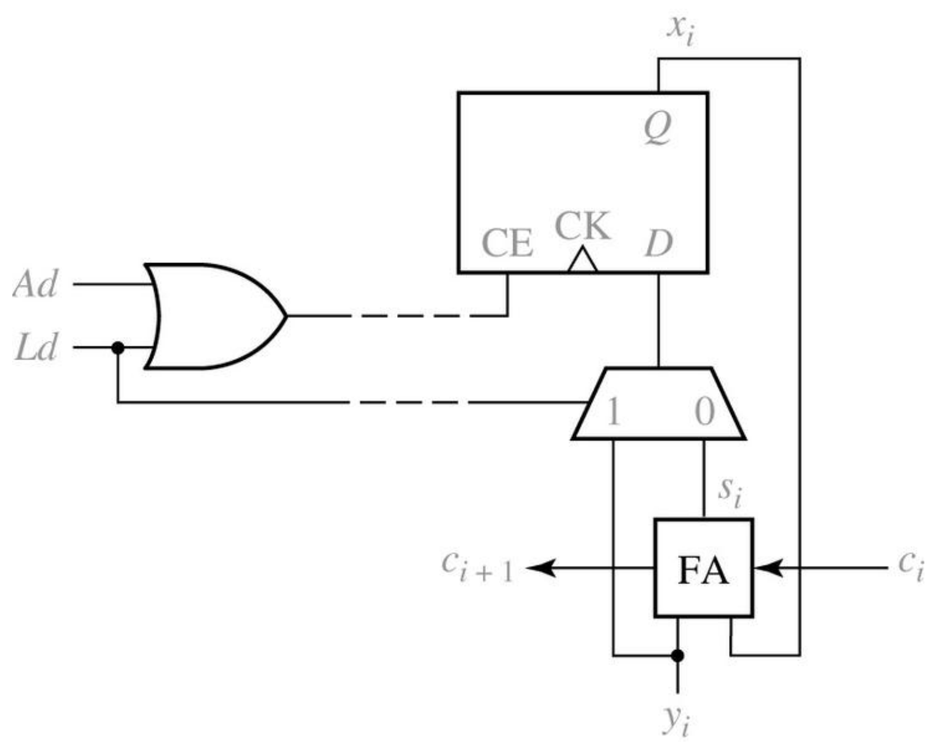
Parallel Adder with Accumulator

N-Bit Parallel Adder with Accumulator



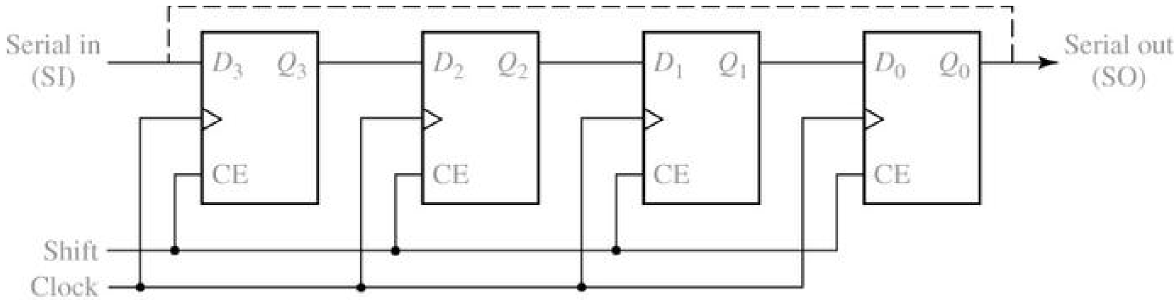
Registers and Register Transfers

Adder Cell with Multiplexer (Figure 12-6)

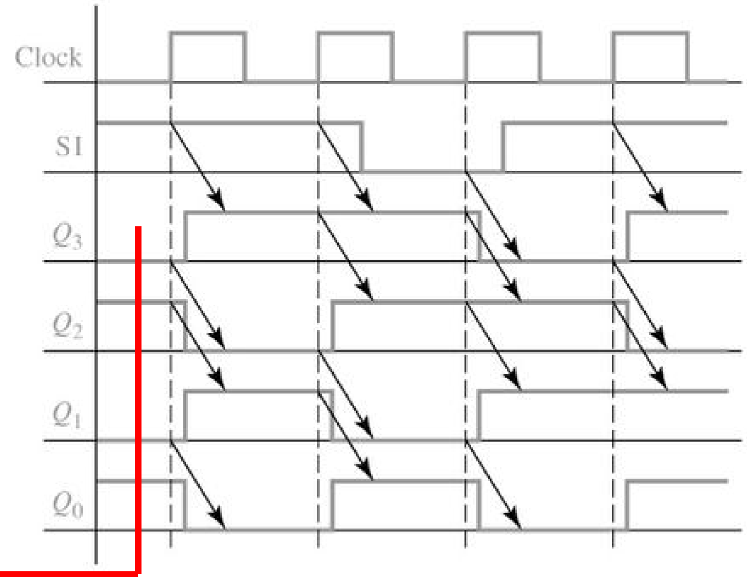


Shifter Registers

Right-Shift Register



(a) Flip-flop connections



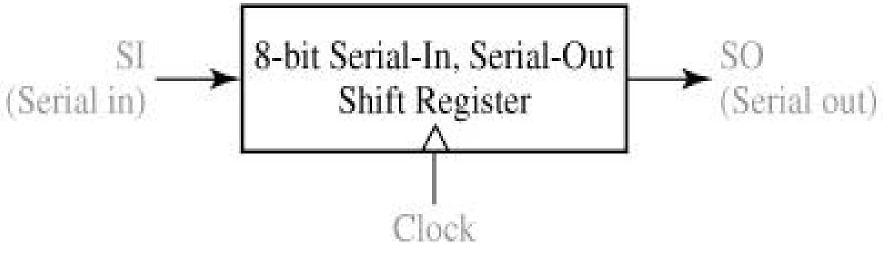
0101→1010→1101→
0110→1011

Initially 0101

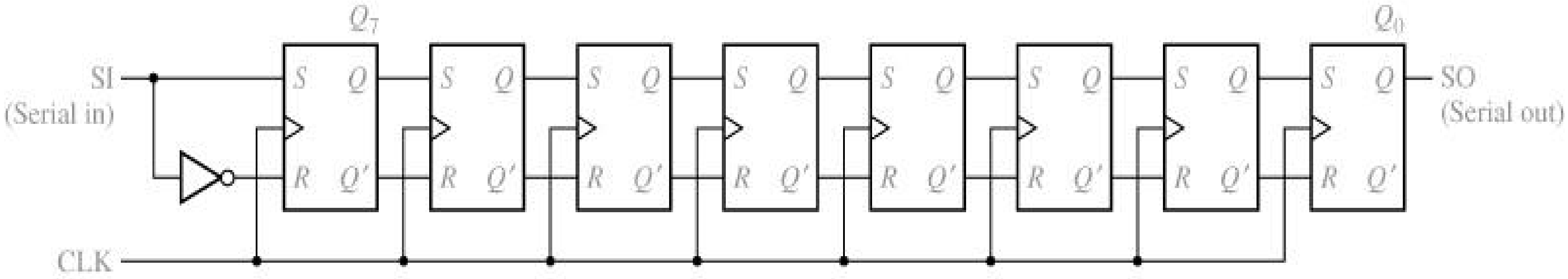
(b) Timing diagram

Shifter Registers

8-Bit Serial-in, Serial-out Shift Register



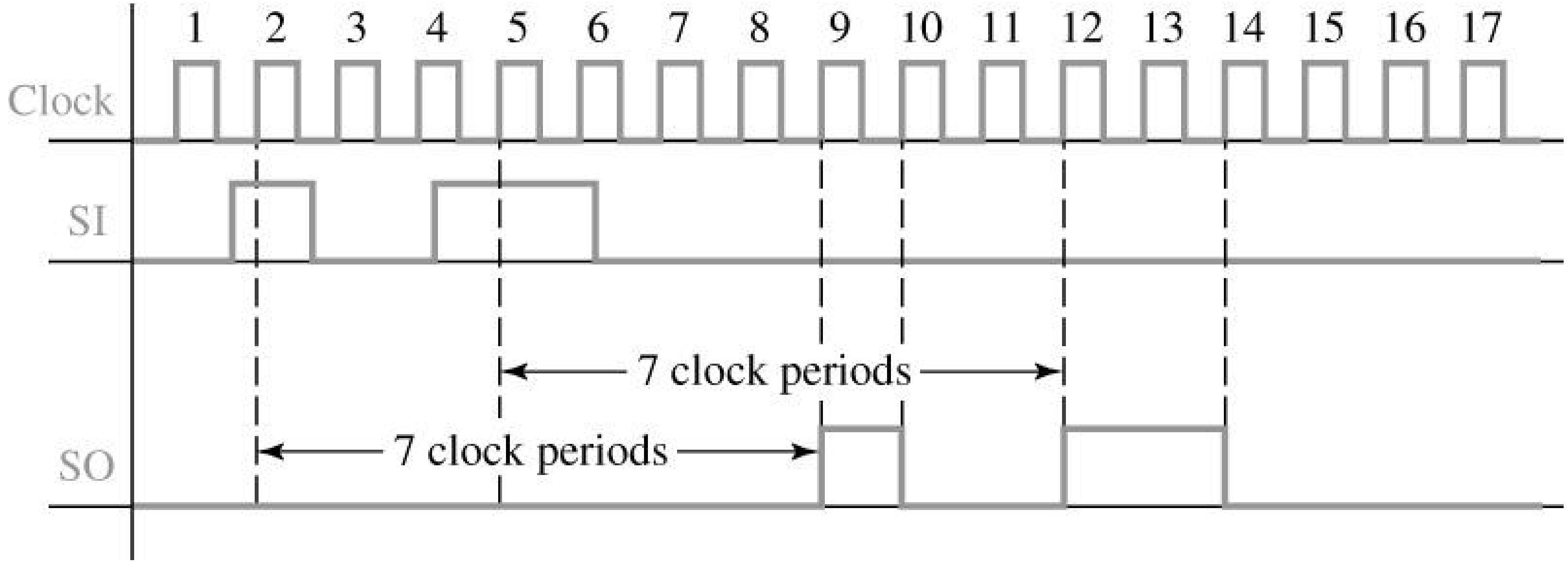
(a) Block diagram



(b) Logic diagram

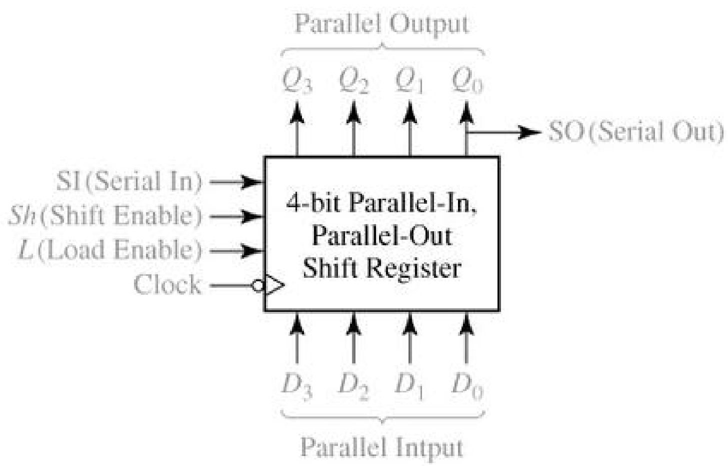
Shifter Registers

Typical Timing Diagram for Shift Register

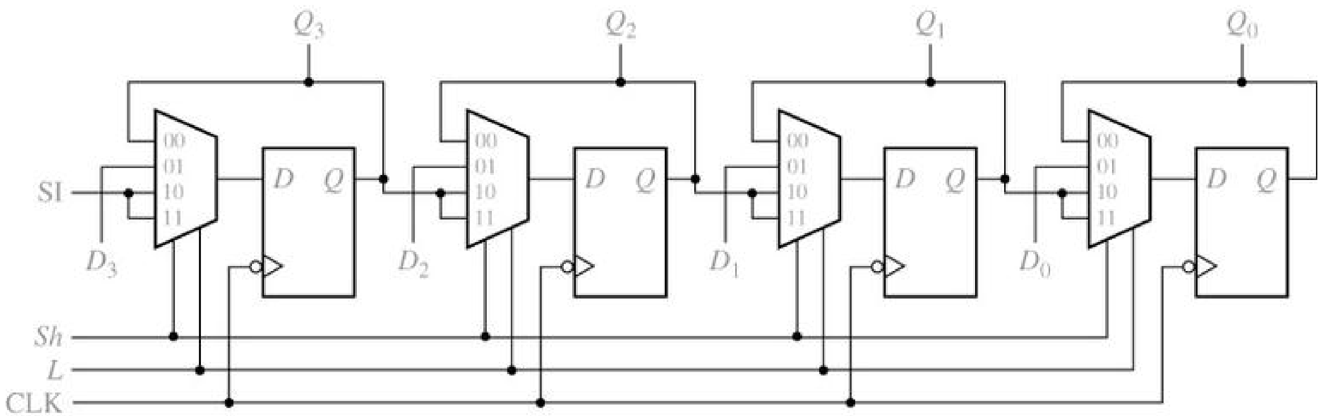


Shifter Registers

Parallel-in, Parallel-Out Right Shift Register



(a) Block diagram



(b) Implementation using flip-flops and MUXes

Shifter Registers

Shift Register Operation (Table 12-1)

Input		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	$\times$	SI	Q_3	Q_2	Q_1	right shift

Shifter Registers

Input		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	×	SI	Q_3	Q_2	Q_1	right shift

The Next-state equations for the F/F are

$$Q_3^+ = Sh' \cdot L' \cdot Q_3 + Sh' \cdot L \cdot D_3 + Sh \cdot \mathrm{SI}$$

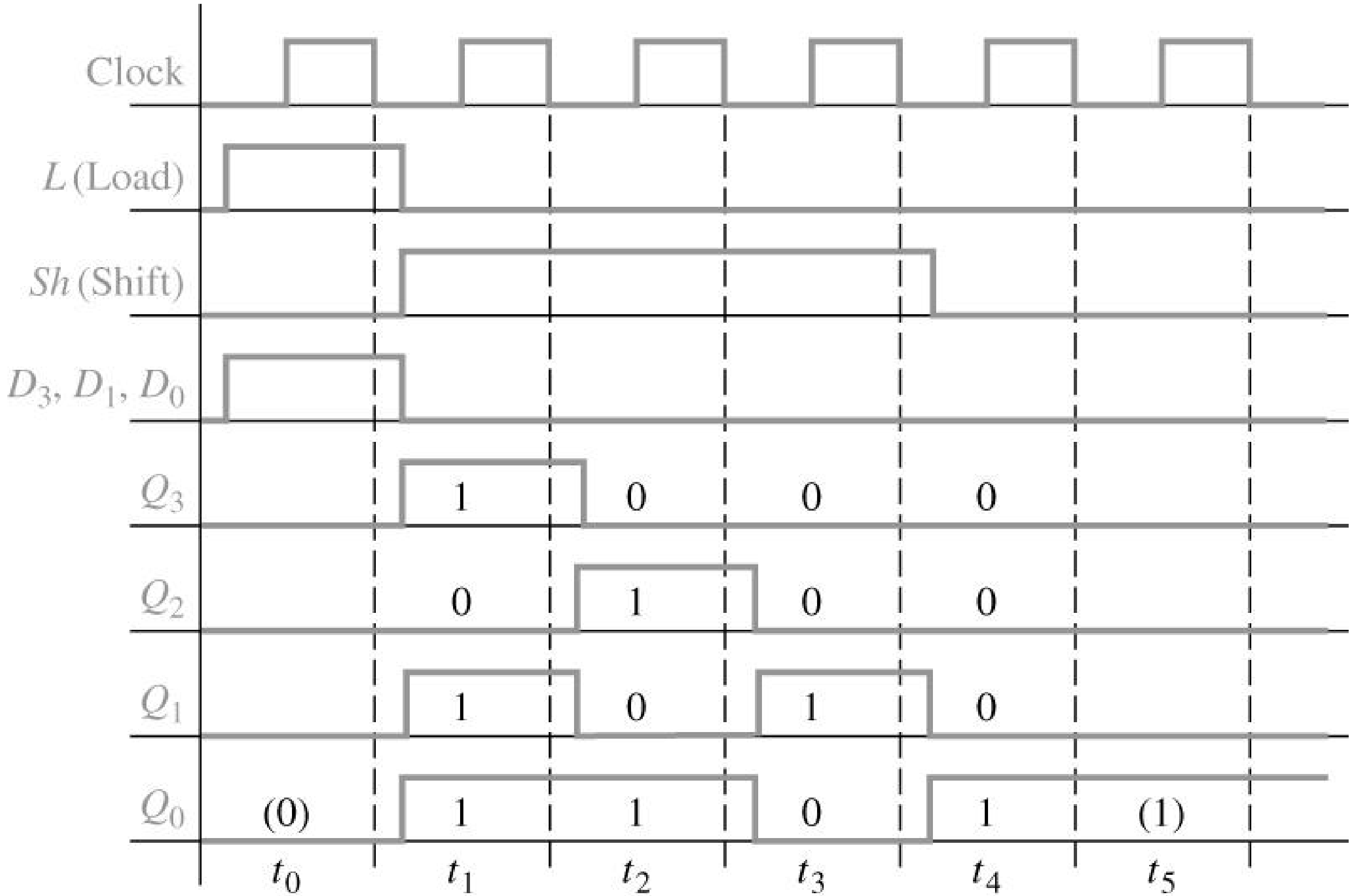
$$Q_2^+ = Sh' \cdot L' \cdot Q_2 + Sh' \cdot L \cdot D_2 + Sh \cdot Q_3$$

$$Q_1^+ = Sh' \cdot L' \cdot Q_1 + Sh' \cdot L \cdot D_1 + Sh \cdot Q_2$$

$$Q_0^+ = Sh' \cdot L' \cdot Q_0 + Sh' \cdot L \cdot D_0 + Sh \cdot Q_1$$

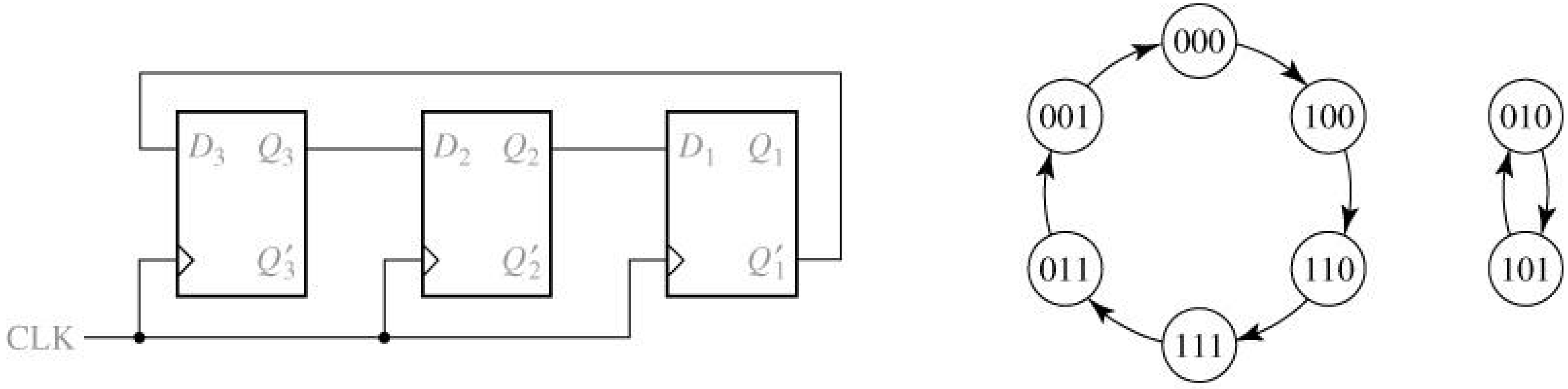
Shifter Registers

Timing Diagram for Shift Register



Shifter Registers

Shift Register with Inverted Feedback (Figure 12-12) → *Johnson Counter*



(a) Flip-flop connections

(b) State graph

A 3-bit shift register 12-12(a)

Successive states 12-12(b)

. 000 => 100 => 110 => 111 => 011 => 001 => 000 ...
. 010 => 101 => 010 => 101