

CS204: 數位系統設計

Registers and Counters

Outline of Chapter 6

- ▣ 6.1 Registers
- ▣ 6.2 Shift Registers
- ▣ 6.3 Ripple Counters
- ▣ 6.4 Synchronous Counters
- ▣ 6.5 Other Counters

6-1 Registers (p.271)

■ Clocked sequential circuits

- ◆ A group of flip-flops and combinational gates
- ◆ Connected to form a feedback path
- ◆ Flip-flops + Combinational gates
(essential) (optional)

■ Register

- ◆ A group of flip-flops and gates
 - » Flip-flops store binary data
 - » Gates determine how the information is transferred into the register

■ Counter

- ◆ A register that goes through a predetermined sequence of states

6-1 Registers (p.272)

■ A n -bit register

- ◆ n flip-flops capable of storing n bits of binary information
- ◆ 4-bit register is shown in Fig. 6.1

Clear_b = 0 (active low): $A_x = 0$

Clear_b = 1 (normal operation)

Clock = \uparrow : $A_x = I_x$

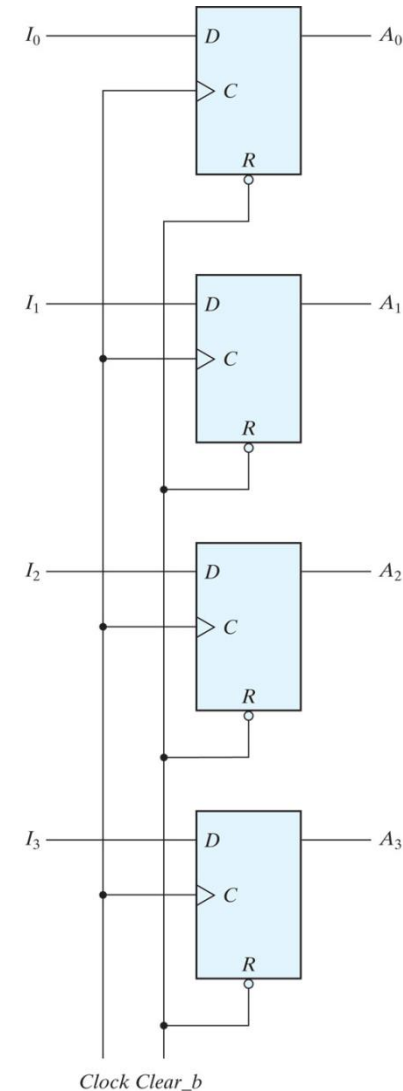


Fig. 6.1 Four-bit register

4-bit Register with Parallel Load (p.273)

1: Parallel load

0: No change

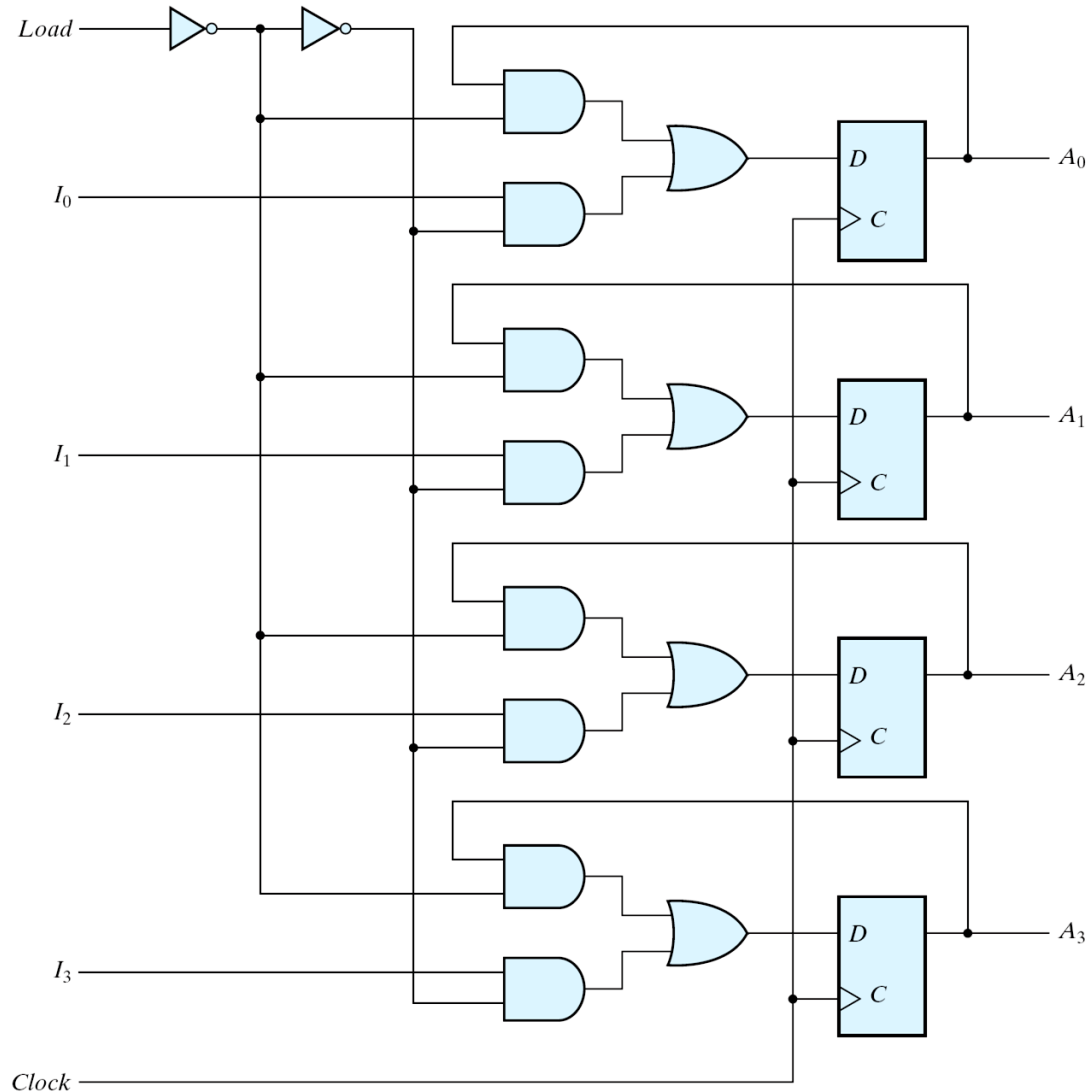


Fig. 6.2 Four-bit register with parallel load

6-2 Shift Registers (p.274, 275)

▣ Shift register

- ◆ A register capable of shifting its binary information in one or both directions

▣ Simplest shift register

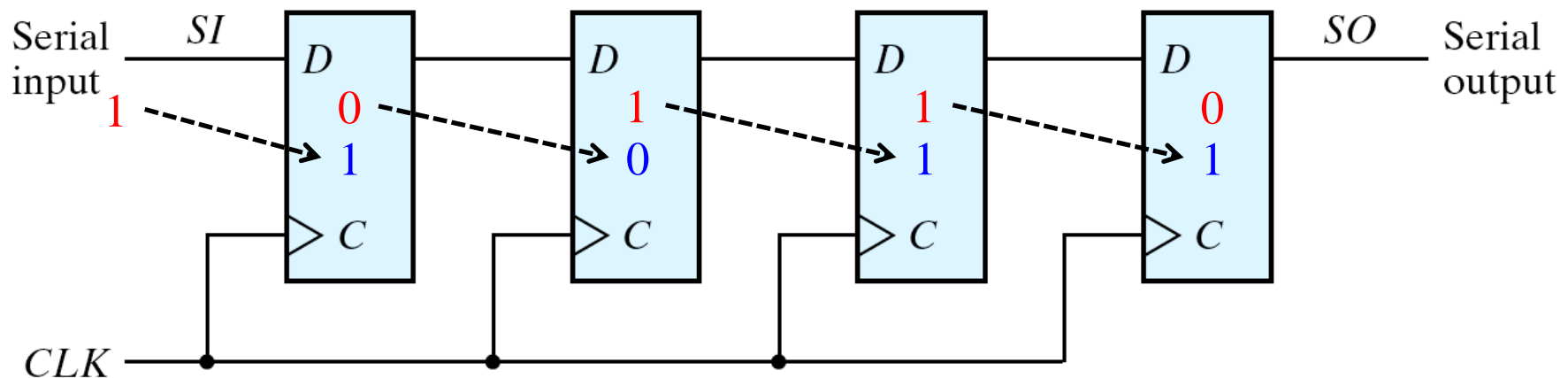


Fig. 6.3 Four-bit shift register

Data Transfer (p.275)

□ Serial transfer vs. Parallel transfer

◆ Serial transfer

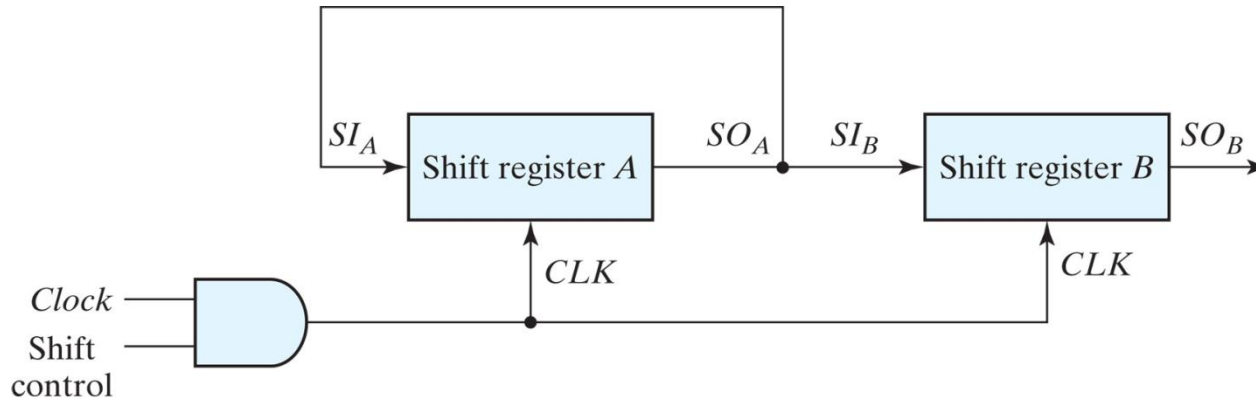
- » Information is transferred **one bit at a time**
- » Shifts the bits out of the source register into the destination register

◆ Parallel transfer

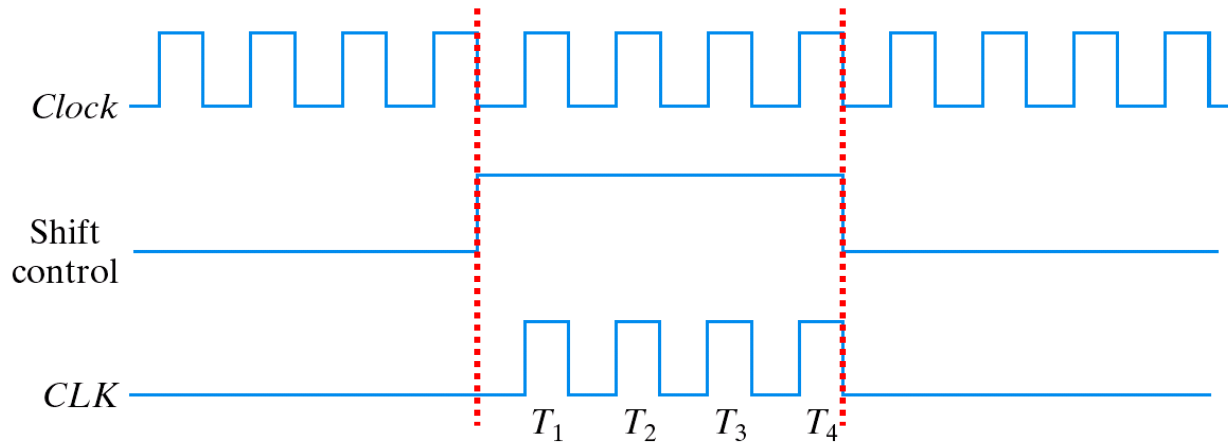
- » All the bits of the register are transferred **at the same time**

Serial Transfer (1/2) (p.276)

Example: Serial transfer from register A to register B



(a) Block diagram

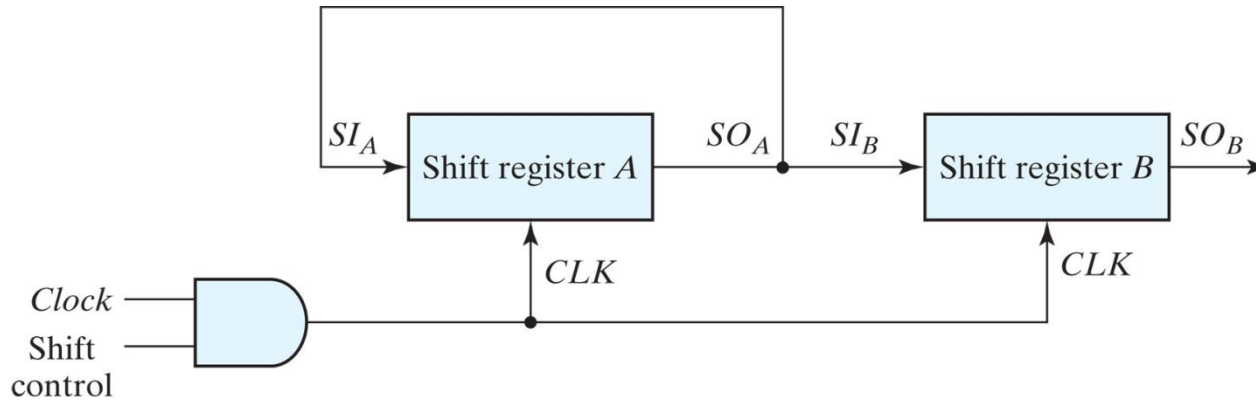


(b) Timing diagram

Fig. 6.4 Serial transfer from 4-bit register A to register B

Serial Transfer (2/2) (p.277)

Example: Serial transfer from register A to register B



(a) Block diagram

Table 6.1
Serial-Transfer Example

Timing Pulse	Shift Register A	Shift Register B
Initial value	1 0 1 1	0 0 1 0
After T_1	1 1 0 1	1 0 0 1
After T_2	1 1 1 0	1 1 0 0
After T_3	0 1 1 1	0 1 1 0
After T_4	1 0 1 1	1 0 1 1

Serial Addition Using D Flip-Flops (p.278)

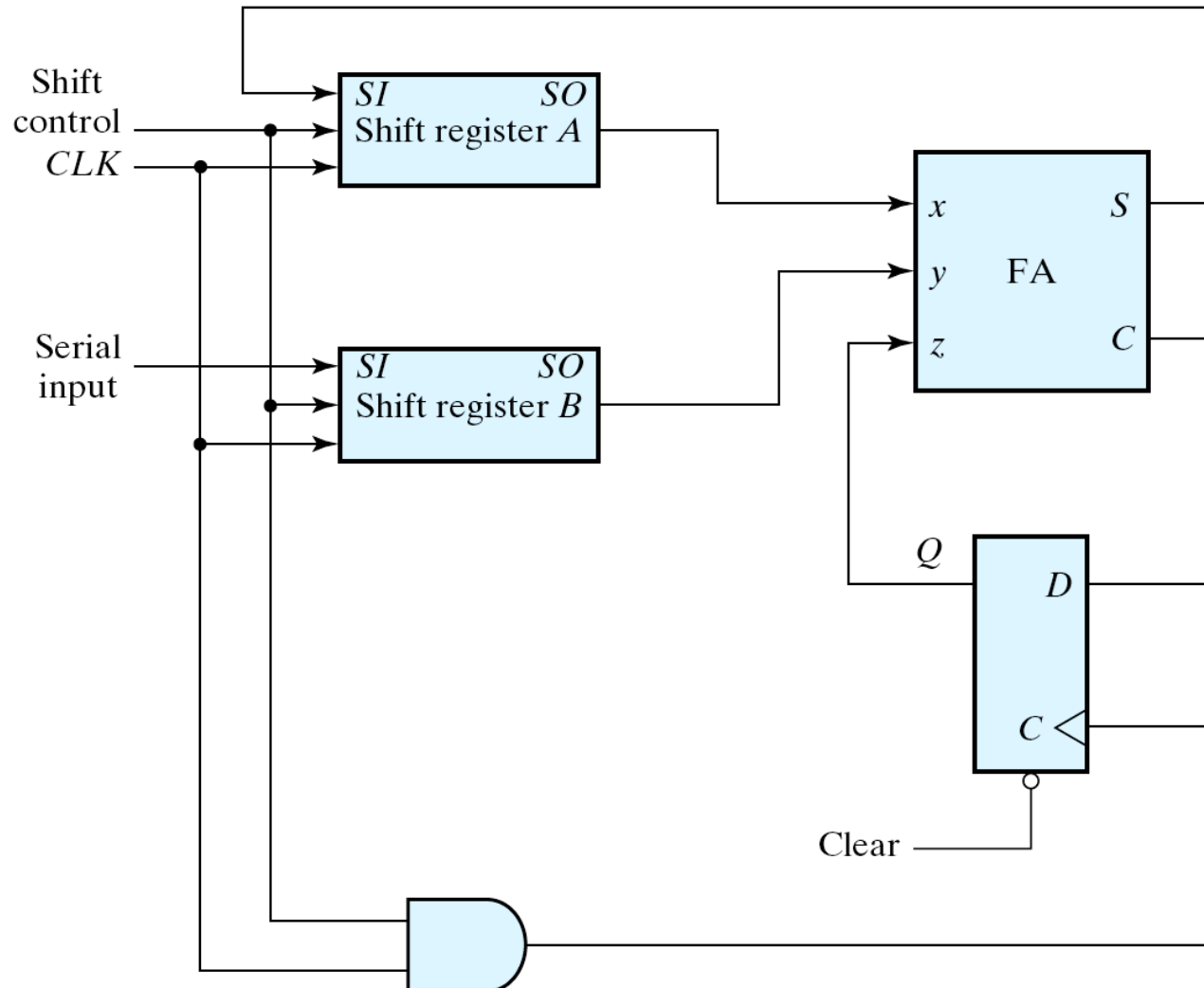


Fig. 6.5 Serial adder

Serial Adder Using JK FFs (1/2) (p.279)

Serial adder using *JK* flip-flops

Table 6.2
State Table for Serial Adder

Present State	Inputs		Next State	Output	Flip-Flop Inputs	
	$Q (C_{in})$	$x \quad y$			J_Q	K_Q
0	0	0	0	0	0	0
0	0	1	1	1	1	0
0	1	0	1	0	1	1
0	1	1	0	1	0	1
1	0	0	0	0	0	1
1	0	1	1	1	1	1
1	1	0	0	0	0	1
1	1	1	1	1	1	1

Serial Adder Using JK FFs (2/2) (p.280)

Circuit diagram

◆ $J_Q = xy$

◆ $K_Q = x'y' = (x+y)'$

◆ $S = x \oplus y \oplus Q$

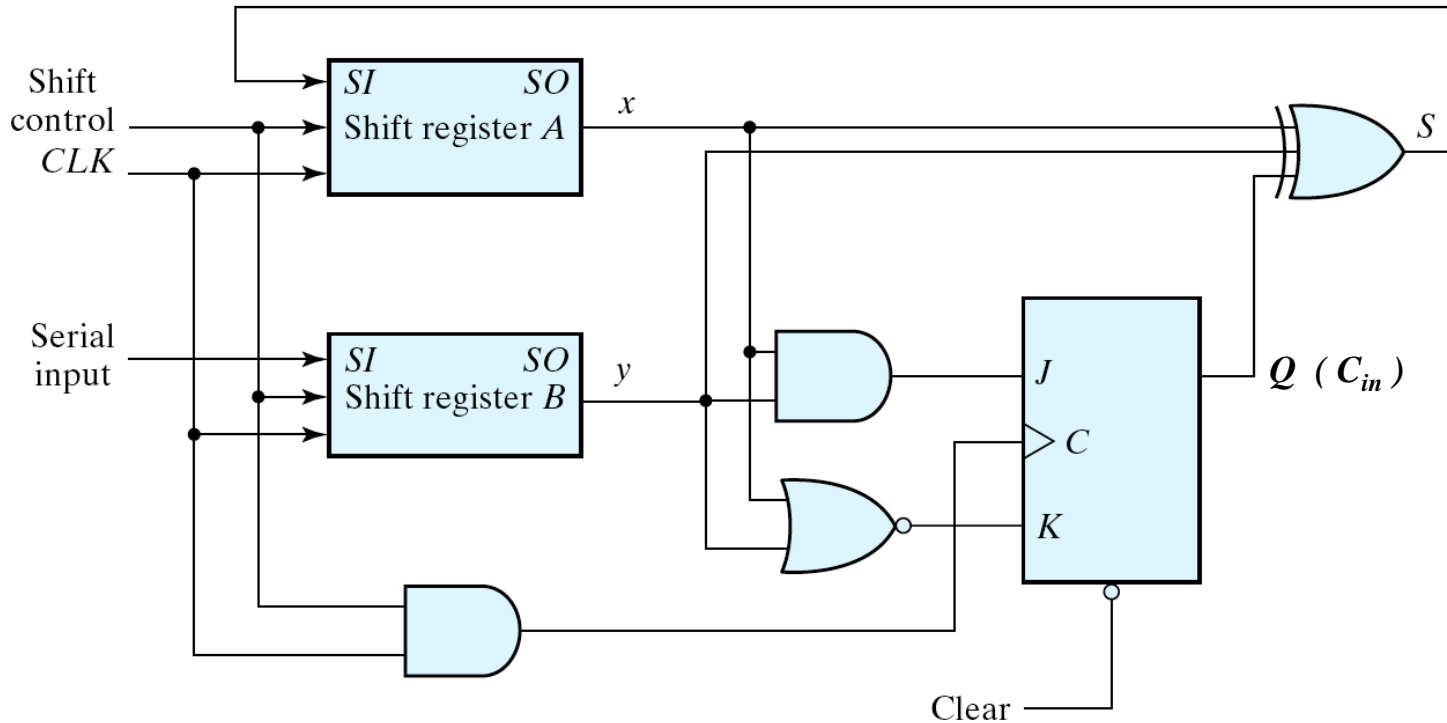


Fig. 6.6 Second form of serial adder

Registers and Counters-12

Universal Shift Register (1/5) (p.280)

■ Three types of shift register

◆ Unidirectional shift register

- » A register capable of shifting in one direction

◆ Bidirectional shift register

- » A register can shift in both directions

◆ Universal shift register

- » Support both direction shifts & parallel load/out

Universal Shift Register (2/5)

■ Capability of a universal shift register:

1. A **clear** control to clear the register to 0
2. A **clock** input to synchronize the operations
3. A **shift-right** control to enable the shift right operation and the **serial input** and **output** lines associated w/ the shift right
4. A **shift-left** control to enable the shift left operation and the **serial input** and **output** lines associated w/ the shift left
5. A **parallel-load** control to enable a parallel transfer and the n **parallel input** lines associated w/ the parallel transfer
6. n **parallel output** lines
7. A control state that leaves the information in the register unchanged in the presence of the clock

Universal Shift Register (3/5)

■ Example: 4-bit universal shift register

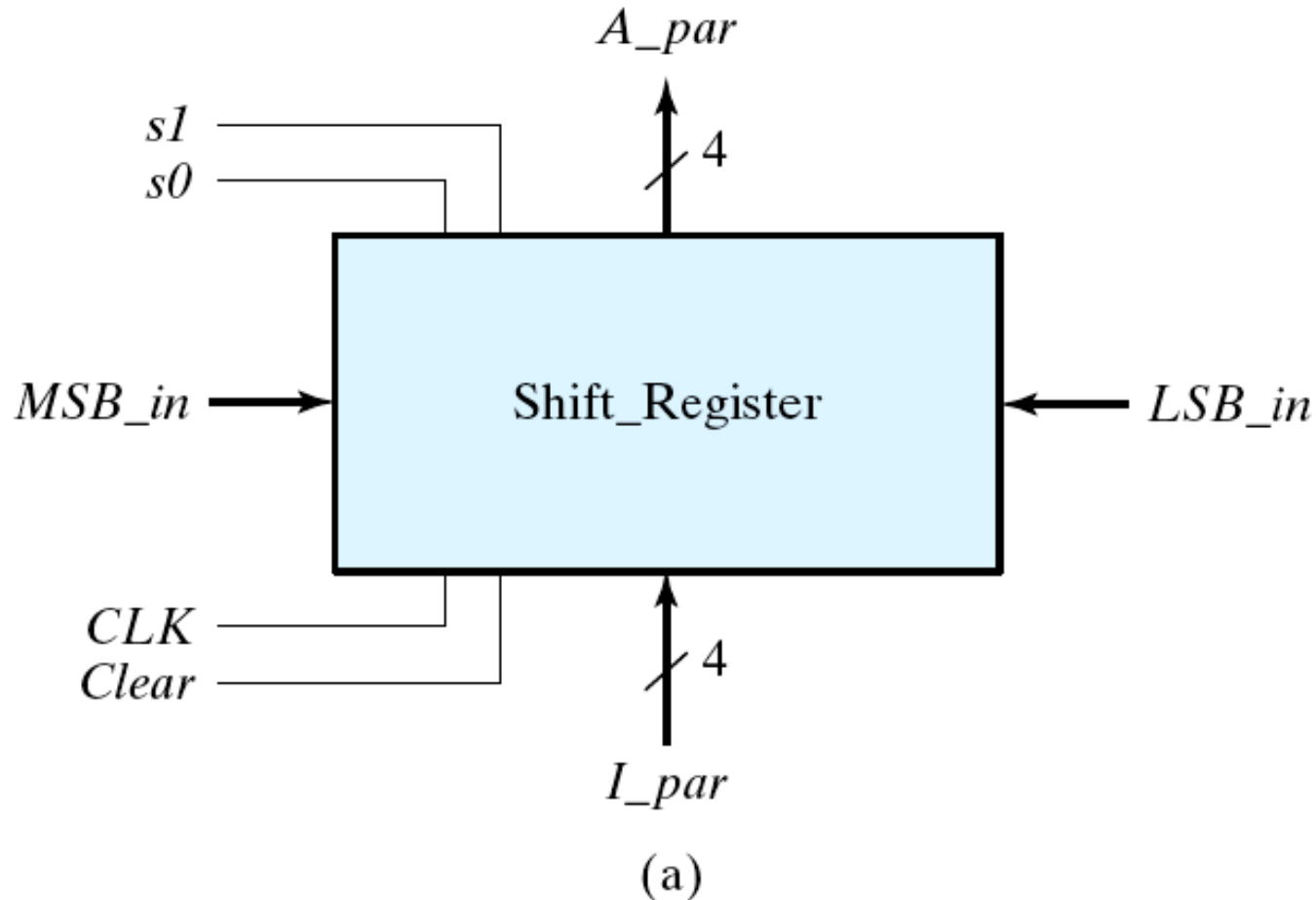


Fig. 6.7 Four-bit universal shift register

Universal Shift Register (4/5)

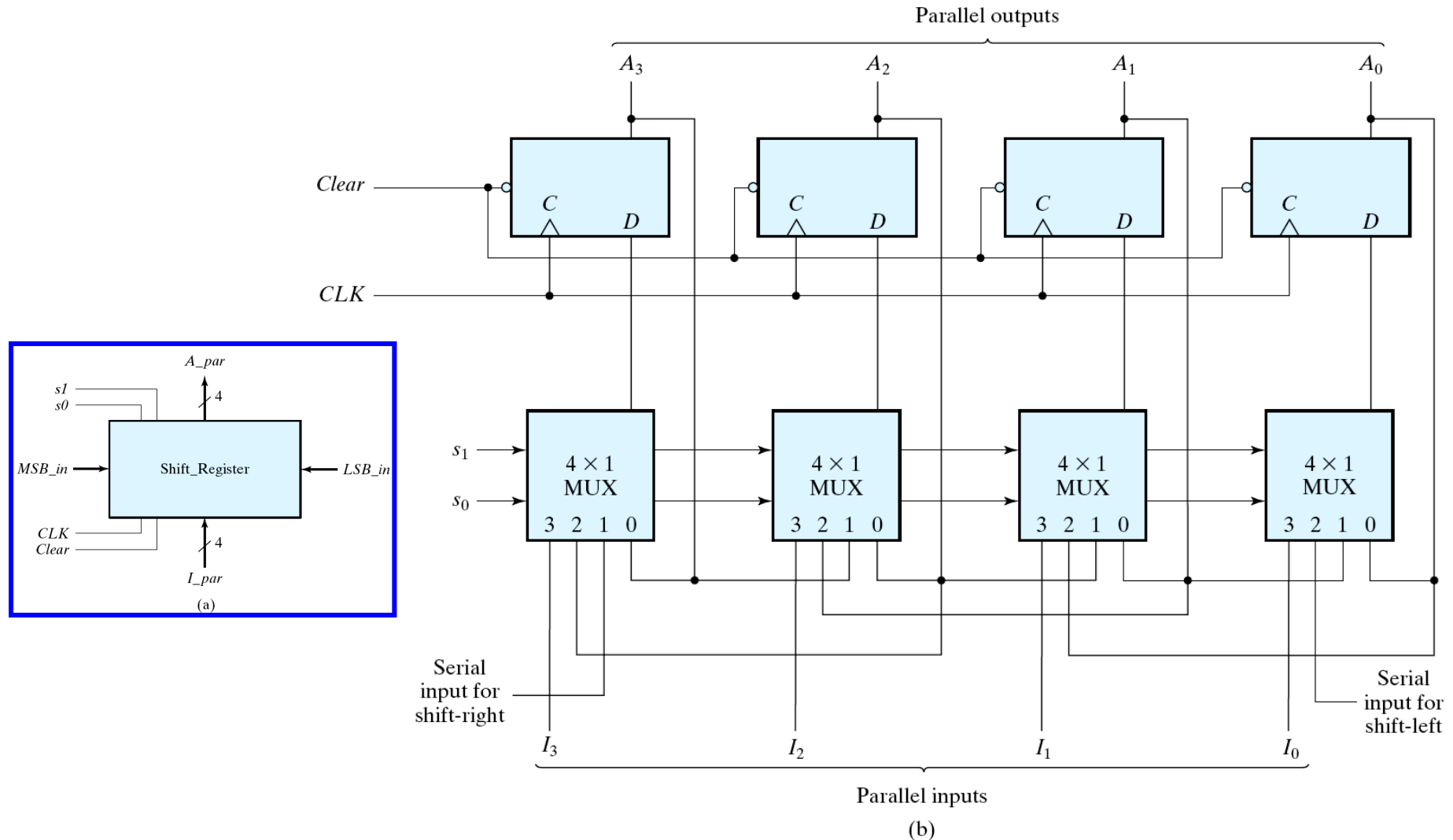


Fig. 6.7 Four-bit universal shift register

Universal Shift Register (5/5)

■ Function Table

Table 6.3
Function Table for the Register of Fig. 6.7

Mode Control		Register Operation
s_1	s_0	
0	0	No change
0	1	Shift right
1	0	Shift left
1	1	Parallel load

Clear	S1	S0	A3+	A2+	A1+	A0+	(operation)
0	x	x	0	0	0	0	Clear
1	0	0	A3	A2	A1	A0	No change
1	0	1	sri	A3	A2	A1	Shift right
1	1	0	A2	A1	A0	sli	Shift left
1	1	1	I3	I2	I1	I0	Parallel load

6-3 Ripple Counters (p.282)

▣ Counter

- ◆ A register that goes through a prescribed sequence of states upon the application of input pulses
 - » Input pulses: may be clock pulses or originate from some external source
 - » The sequence of states: may follow the binary number sequence (\Rightarrow **binary counter**) or any other sequence of states
 - » A n -bit binary counter $\rightarrow n$ FFs \rightarrow count from 0 to 2^n-1

Counters

▣ Categories of counters

1. Ripple counters

- ◆ The flip-flop output transition serves as a source to trigger other flip-flops
- ◆ **No common clock pulse** (not synchronous)

2. Synchronous counters

- ◆ The CLK inputs of all flip-flops receive a **common clock**

4-bit Binary Count Sequence (p.283)

■ Binary count sequence: 4-bit

Table 6.4
Binary Count Sequence

A_3	A_2	A_1	A_0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0

Ripple Counter

A_3	A_2	A_1	A_0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0

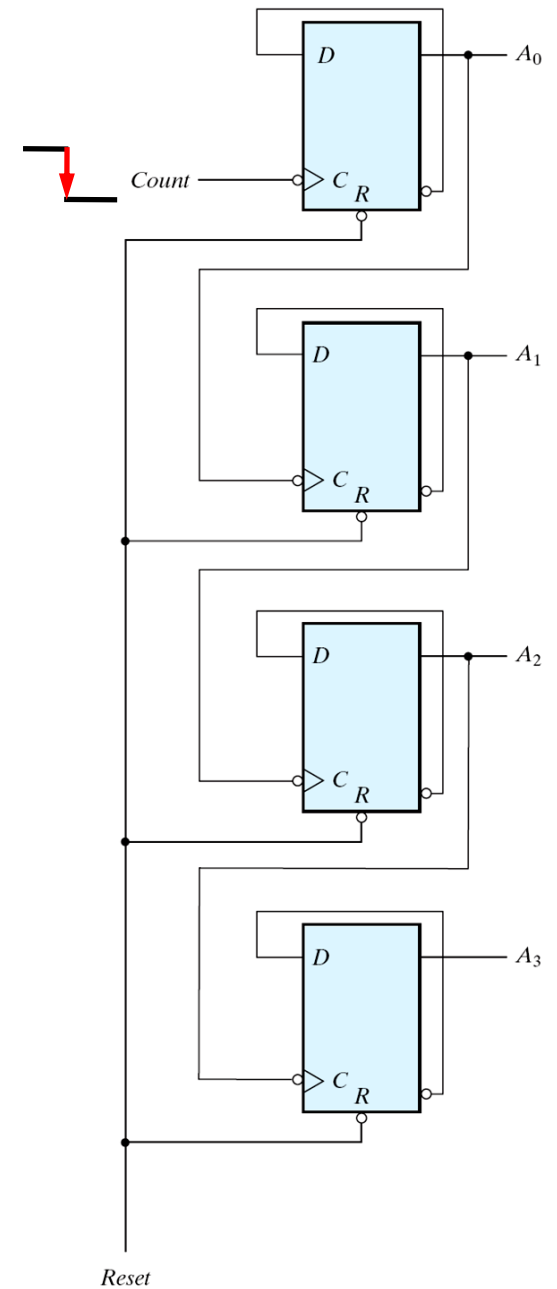
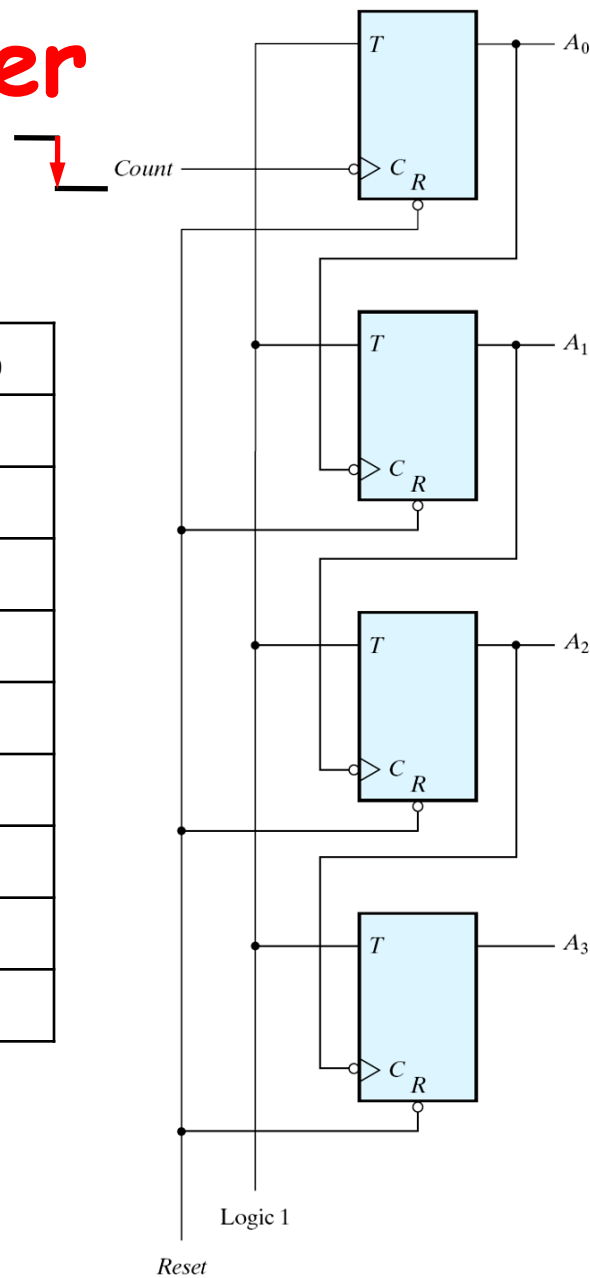


Fig. 6.8 Four-bit binary ripple counter

(a) With T flip-flops

(b) With D flip-flops

BCD Ripple Counter (p.285)

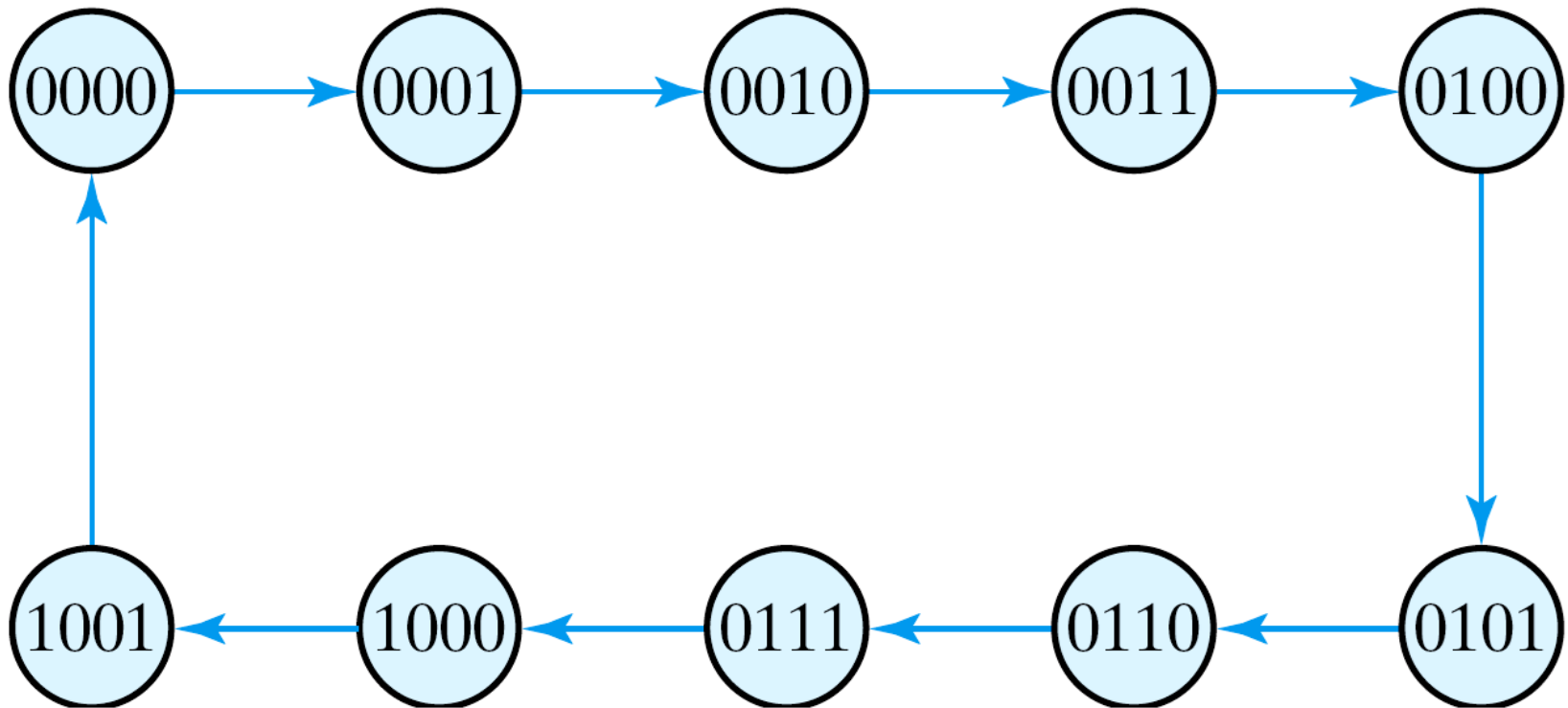
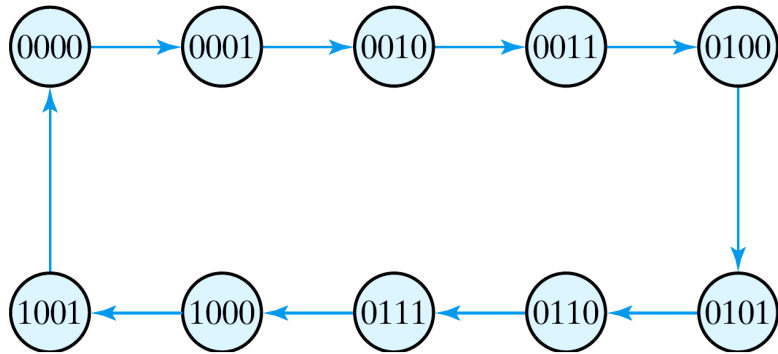


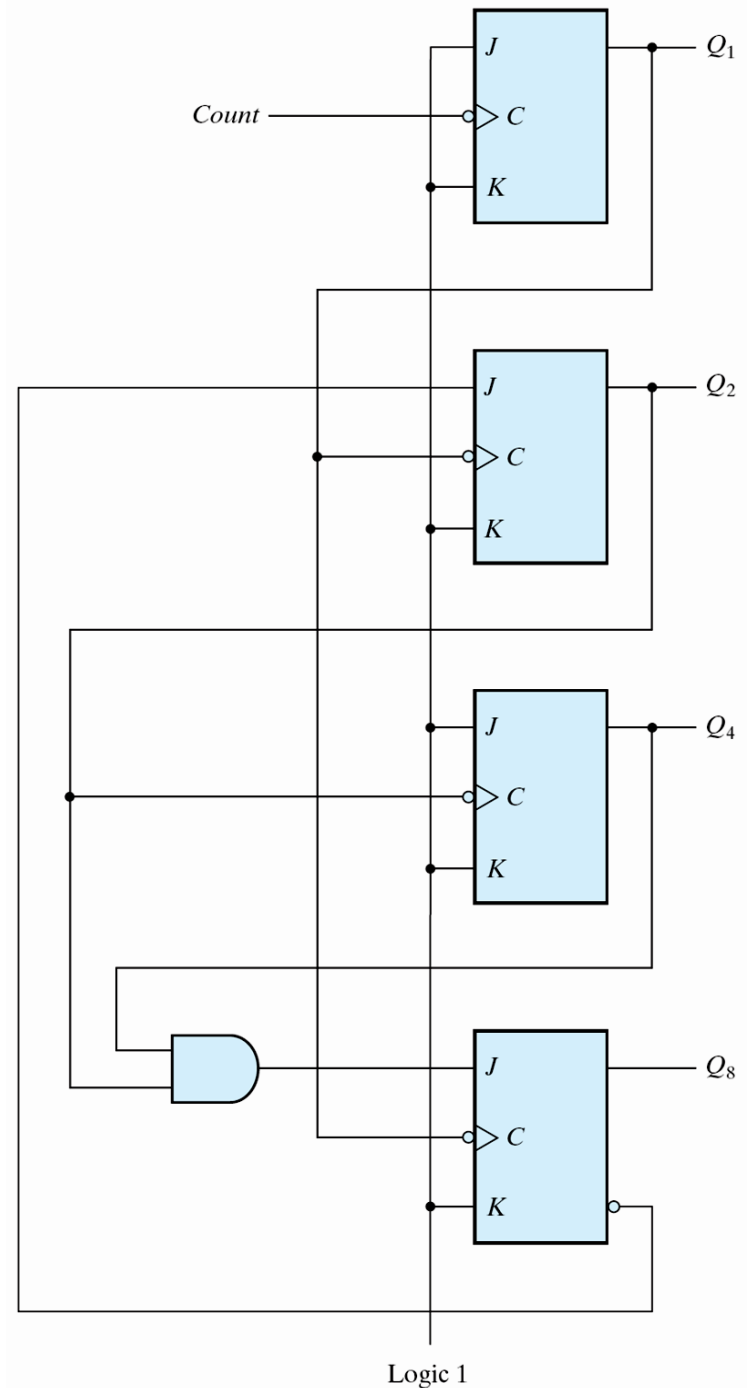
Fig. 6.9 State diagram of a decimal BCD counter

BCD Ripple Counter



Q_8	Q_4	Q_2	Q_1
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1

Fig. 6.10 BCD ripple counter



Decade Counter (p.287)

■ Three-decade BCD counter

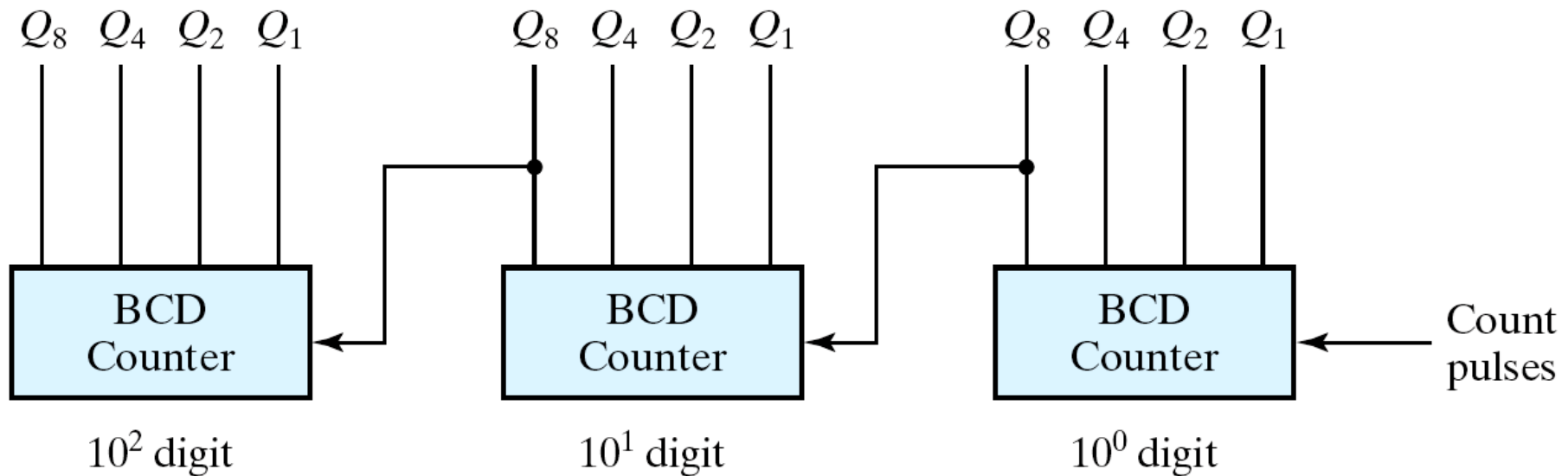


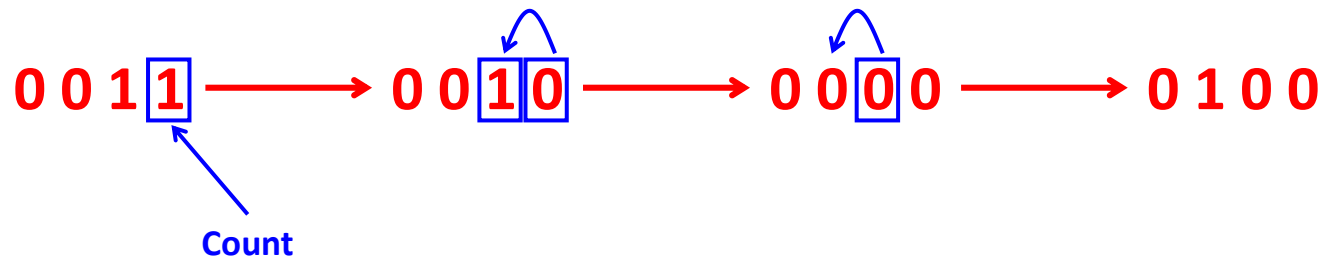
Fig. 6.11 Block diagram of a three-decade decimal BCD counter

6-4 Synchronous Counters (1/2) (p.287)

■ Review of counters

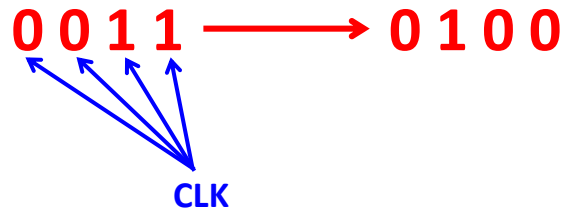
1. Ripple counters

- ◆ The flip-flop output transition serves as a source to trigger other flip-flop
- ◆ \Rightarrow No common clock pulse (not synchronous)



2. Synchronous counters

- ◆ The CLK inputs of all flip-flops receive a common clock



6-4 Synchronous Counters (2/2)

■ Synchronous counter

- ◆ A common clock triggers all flip-flops simultaneously

■ Design procedure

- ◆ Apply the same procedure of sync sequential circuits
- ◆ Sync counter is simpler than general sync sequential circuits

■ T and JK FFs

- ◆ $T=0$ or $J=K=0$: no change
- ◆ $T=1$ or $J=K=1$: complement

Sync Counters using JK FFs (p.289)

4-bit binary counter

A_3	A_2	A_1	A_0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

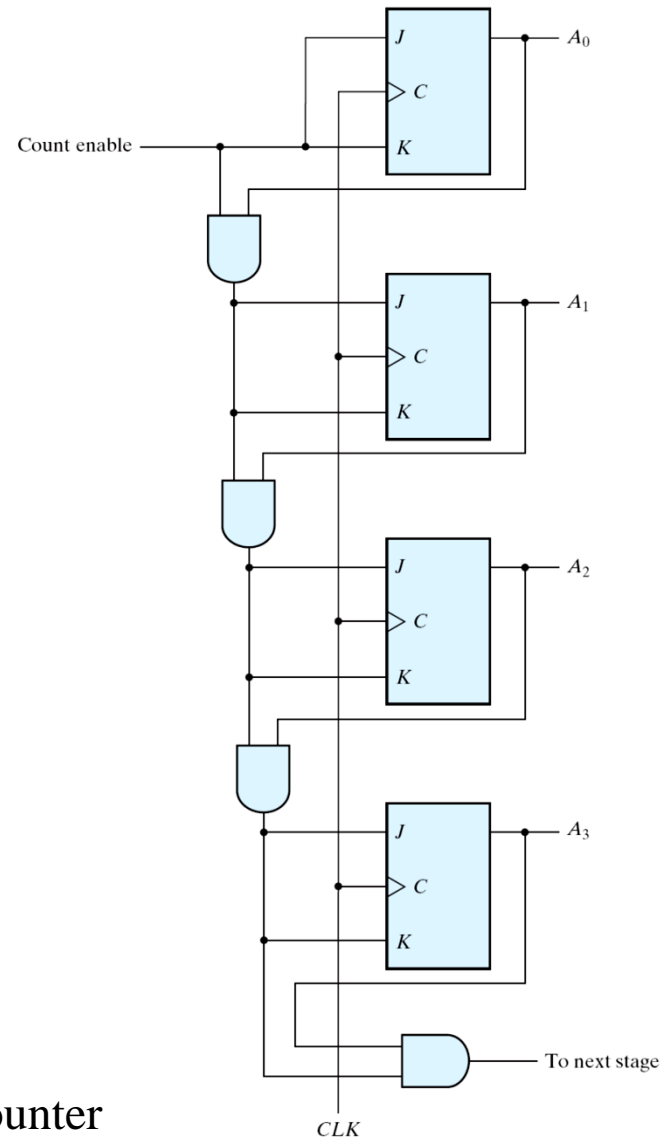


Fig. 6.12 Four-bit synchronous binary counter

4-bit Up/Down Binary Counter

Up	Down	Function
0	0	No change
0	1	Down Count
1	0	Up Count
1	1	Up Count

A ₃	A ₂	A ₁	A ₀
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

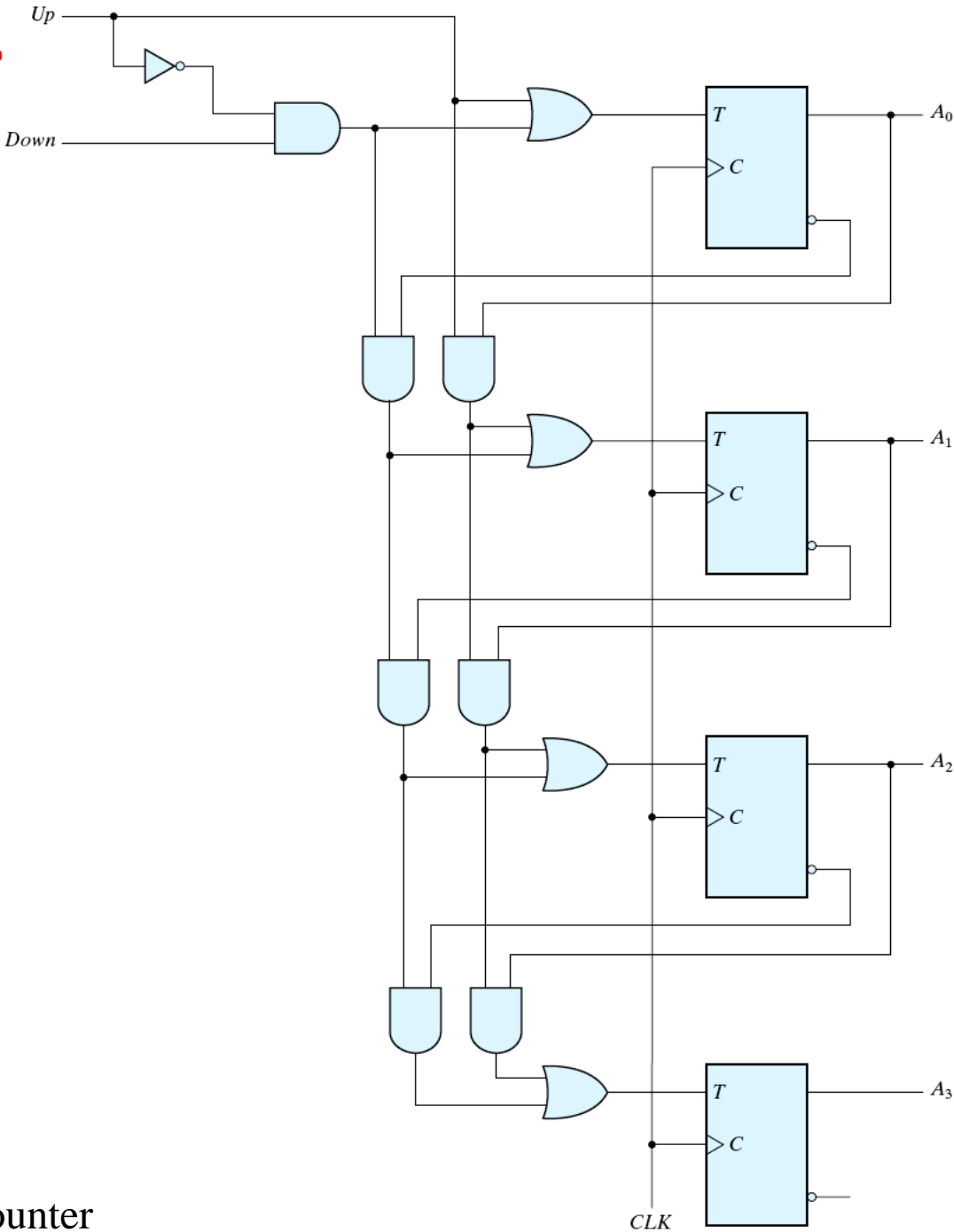


Fig. 6.13 Four-bit up-down binary counter

BCD Counters (p.291)

Table 6.5
State Table for BCD Counter

Present State				Next State				Output	Flip-Flop Inputs			
Q_8	Q_4	Q_2	Q_1	Q_8	Q_4	Q_2	Q_1	y	TQ_8	TQ_4	TQ_2	TQ_1
0	0	0	0	0	0	0	1	0				
0	0	0	1	0	0	1	0	0				
0	0	1	0	0	0	1	1	0				
0	0	1	1	0	1	0	0	0				
0	1	0	0	0	1	0	1	0				
0	1	0	1	0	1	1	0	0				
0	1	1	0	0	1	1	1	0				
0	1	1	1	1	0	0	0	0				
1	0	0	0	1	0	0	1	0				
1	0	0	1	0	0	0	0	1				

■ Simplified functions

$$\begin{aligned} T_{Q1} &= 1 \\ T_{Q2} &= Q_8'Q_1 \end{aligned}$$

$$\begin{aligned} T_{Q4} &= Q_2Q_1 \\ T_{Q8} &= Q_8Q_1 + Q_4Q_2Q_1 \\ y &= Q_8Q_1 \end{aligned}$$

Binary Counter with Parallel Load (p.292)

■ 4-bit binary counter with parallel load

Table 6.6

Function Table for the Counter of Fig. 6.14

Clear	CLK	Load	Count	Function
0	X	X	X	Clear to 0
1	↑	1	X	Load inputs
1	↑	0	1	Count next binary state
1	↑	0	0	No change

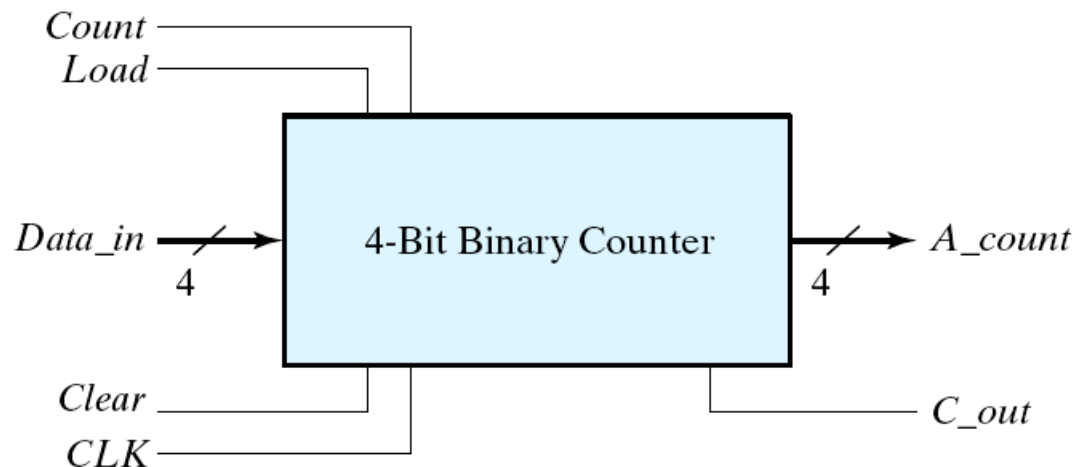


Fig. 6.14 Four-bit binary counter with parallel load

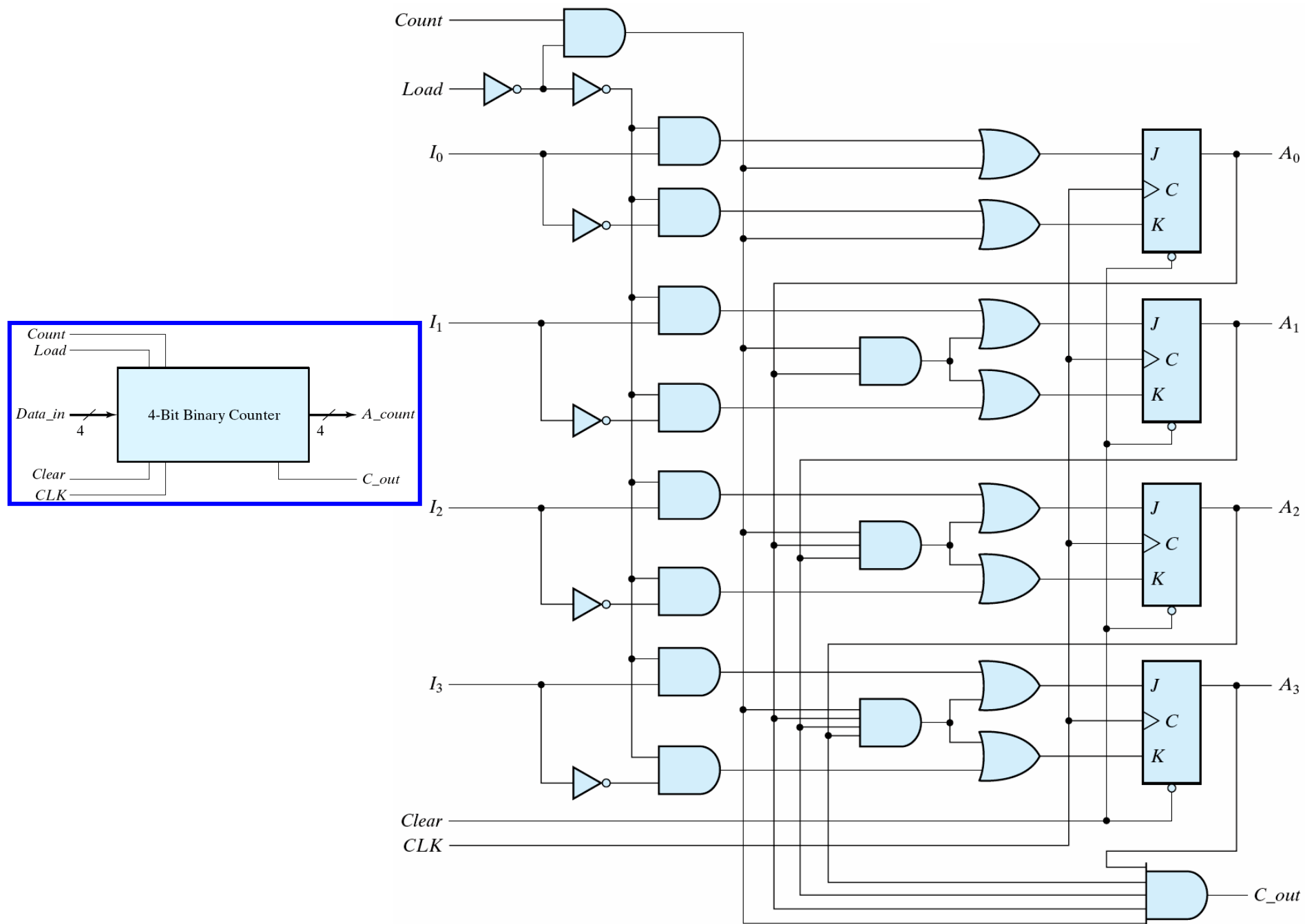
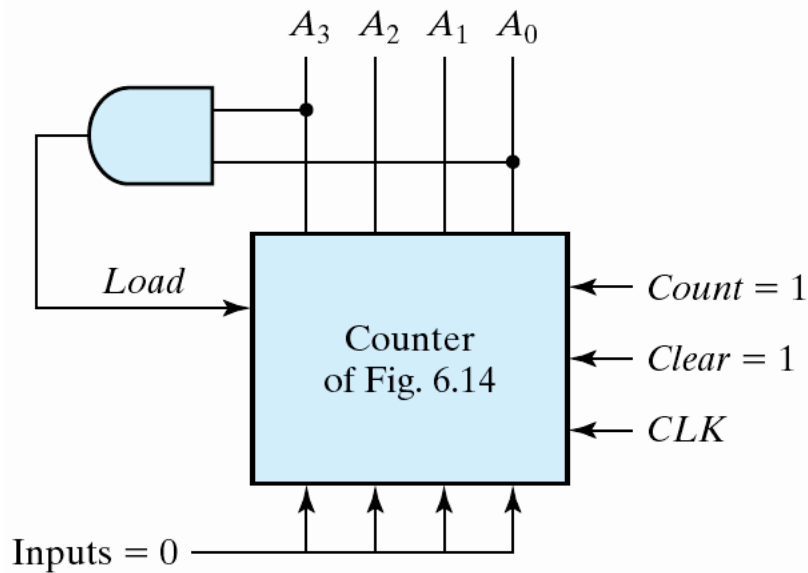


Fig. 6.14 Four-bit binary counter with parallel load

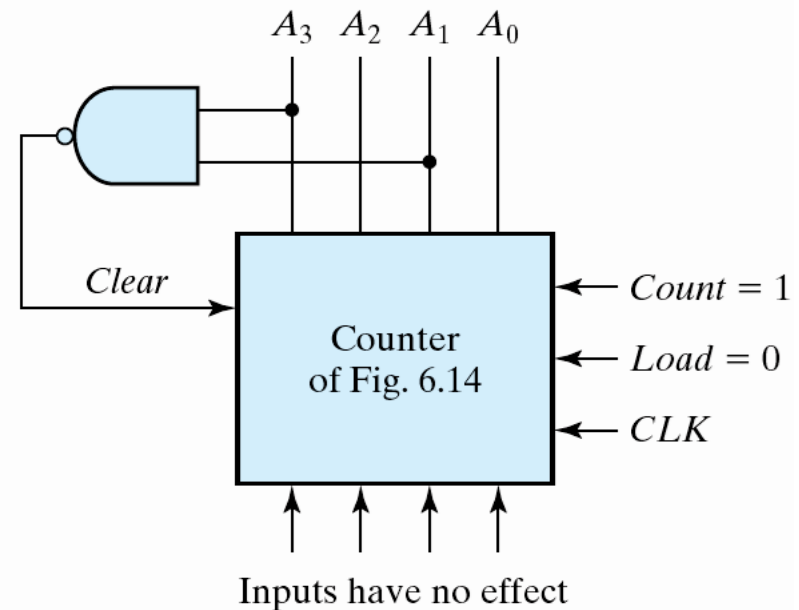
Extensions of Parallel Load Counter (p.294)

■ Generate any count sequence

◆ BCD counter



(a) Using the load input



(b) Using the clear input

Fig. 6.15 Two ways to achieve a BCD counter using a counter with parallel load

6-5 Other Counters (p.294)

▣ Counters

- ◆ Can be designed to generate any desired sequence of states

▣ Divide-by- N counter (modulo- N counter)

- ◆ A counter that goes through a repeated sequence of N states
- ◆ The sequence may follow the binary count or may be any other arbitrary sequence

Counter with Unused States (p.295)

- ▣ n flip-flops $\Rightarrow 2^n$ binary states
- ▣ Unused states
 - ◆ States that are not used in specifying the FSM
 - ◆ May be treated as don't-care conditions or may be assigned specific next states
- ▣ **Self-correcting** counter
 - ◆ Ensure that when a circuit enter one of its unused states, it eventually goes into one of the valid states after one or more clock pulses so it can resume normal operation
 - \Rightarrow Analyze the circuit to determine the next state from an unused state after it is designed

Self-Correcting Counter (1/2) (p.295, 296)

■ An example

Table 6.7
State Table for Counter

Present State			Next State			Flip-Flop Inputs					
A	B	C	A	B	C	J_A	K_A	J_B	K_B	J_C	K_C
0	0	0	0	0	1	0	X	0	X	1	X
0	0	1	0	1	0	0	X	1	X	X	1
0	1	0	1	0	0	1	X	X	1	0	X
1	0	0	1	0	1	X	0	0	X	1	X
1	0	1	1	1	0	X	0	1	X	X	1
1	1	0	0	0	0	X	1	X	1	0	X

- ◆ Two unused states: 011 & 111
- ◆ The simplified flip-flop input equations:
 - » $J_A = B, K_A = B$
 - » $J_B = C, K_B = 1$
 - » $J_C = B', K_C = 1$

Self-Correcting Counter (2/2) (p.295, 296)

■ The logic diagram & state diagram of the circuit

The simplified flip-flop input equations:

$$J_A = B, K_A = B$$

$$J_B = C, K_B = 1$$

$$J_C = B', K_C = 1$$

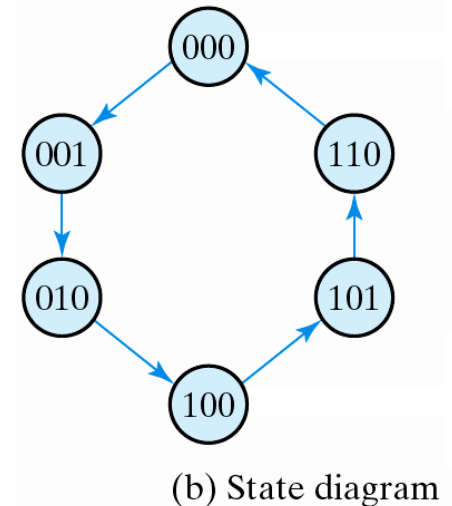
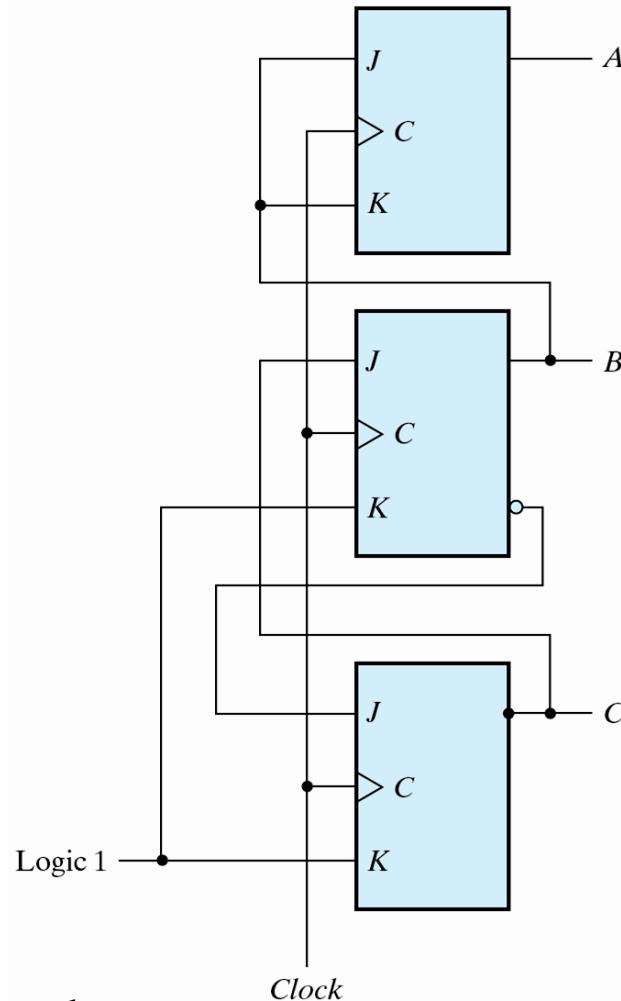


Fig. 6.16 Counter with unsigned states

(a) Logic diagram

Ring Counter (1/4) (p.296, 297)

□ Ring counter

- ◆ A circular shift register with only one flip-flop being set at any particular time, all others are cleared (initial value = 1 0 0 ... 0)
- ◆ The single bit is shifted from one flip-flop to the next to produce the sequence of timing signals

Ring Counter (2/4) (p.296, 297)

■ A 4-bit ring counter

A3	A2	A1	A0
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1
1	0	0	0

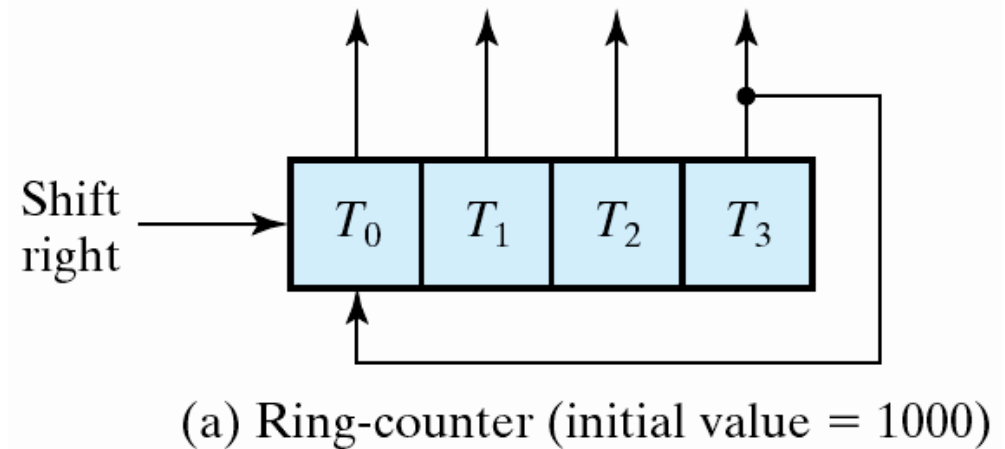


Fig. 6.17 Generation of timing signals

Ring Counter (3/4) (p.296, 297)

■ Application of counters

- ◆ Counters may be used to generate timing signals to control the sequence of operations in a digital system

■ Approaches for generation of 2^n timing signals

1. A shift register with 2^n flip-flops
2. An n -bit binary counter together with an n -to- 2^n -line decoder

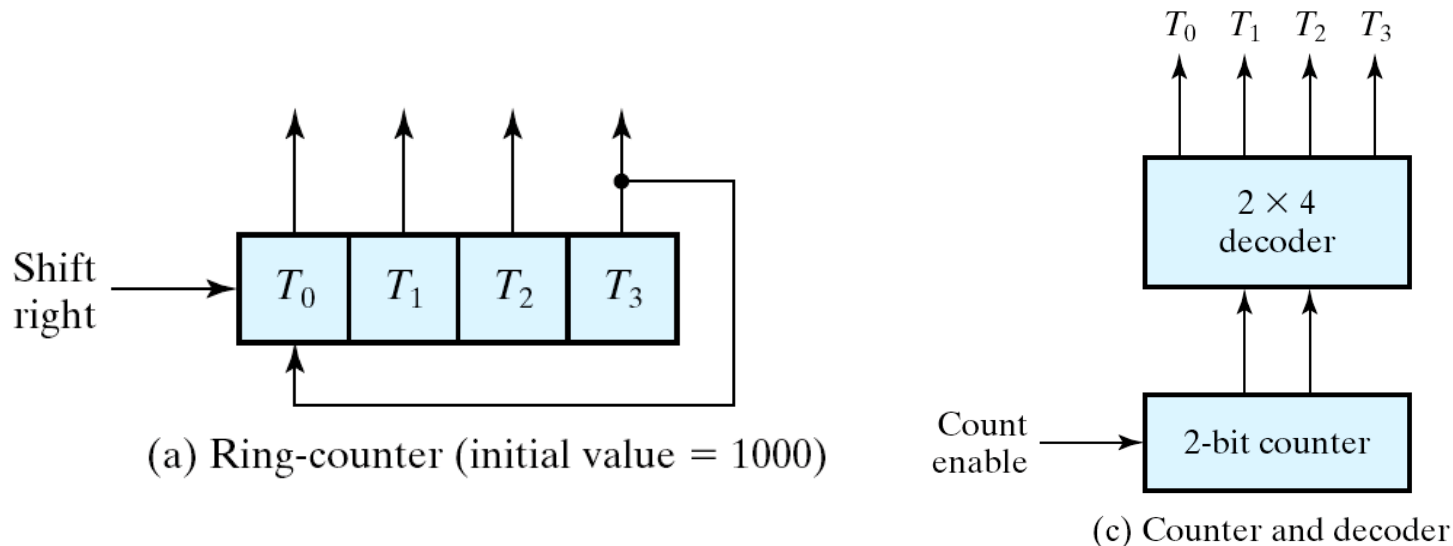


Fig. 6.17 Generation of timing signals

Ring Counter (4/4) (p.296, 297)

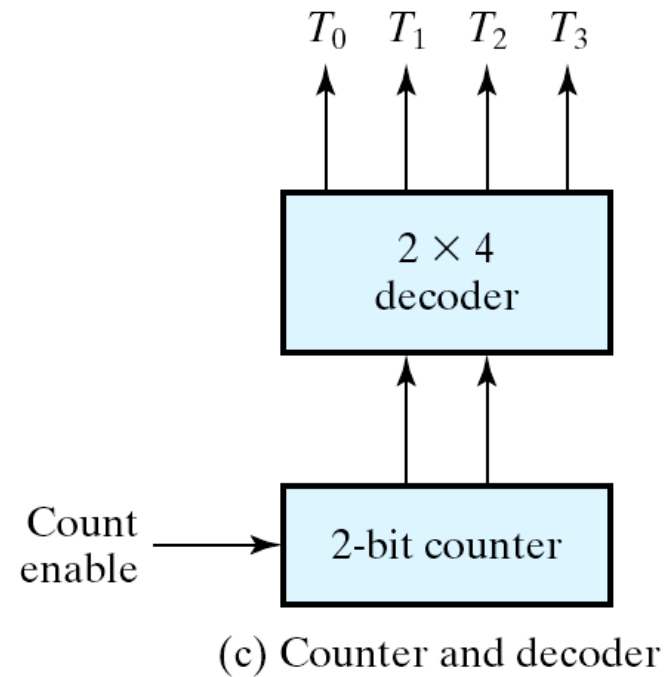
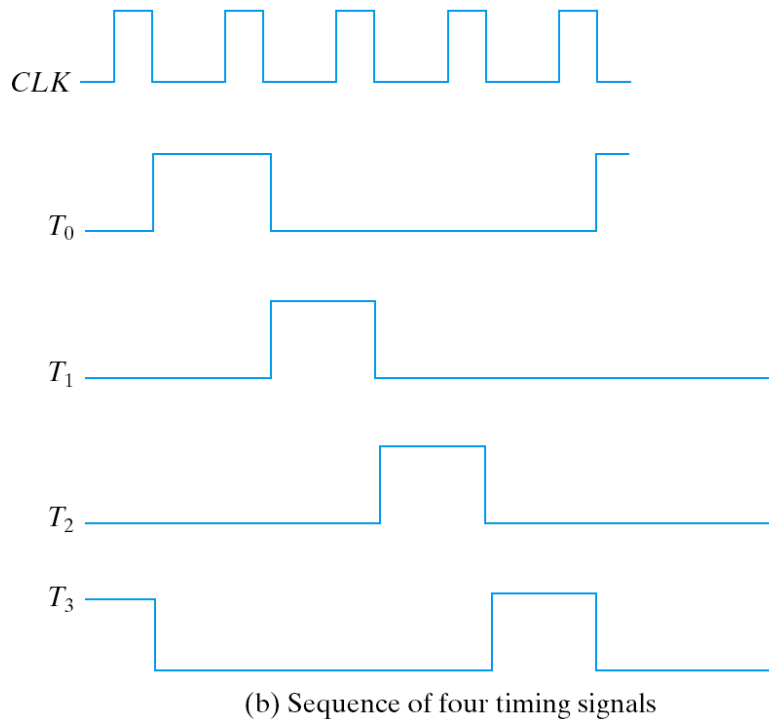
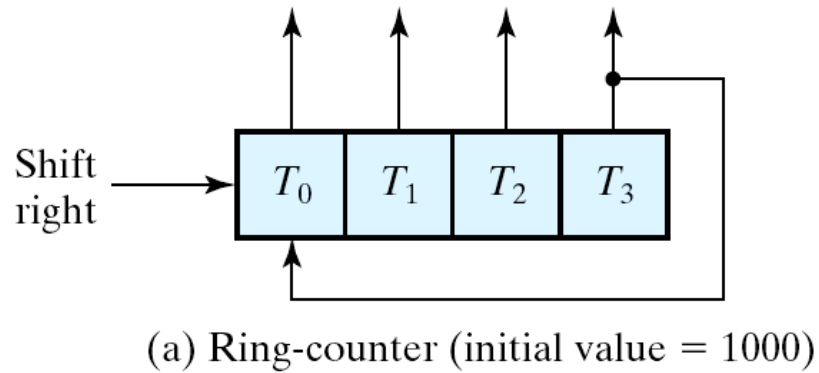


Fig. 6.17 Generation of timing signals

Johnson Counter (1/4) (p.298, 299)

□ Ring counter vs. Switch-tail ring counter

◆ Ring counter

- » A k -bit ring counter circulates a single bit among the flip-flops to provide k distinguishable states (initial value = 1 0 ... 0)

State	Q0	Q1	Q2	Q3
0	1	0	0	0
1	0	1	0	0
2	0	0	1	0
3	0	0	0	1
0	1	0	0	0
1	0	1	0	0
2	0	0	1	0
3	0	0	0	1
0	1	0	0	0

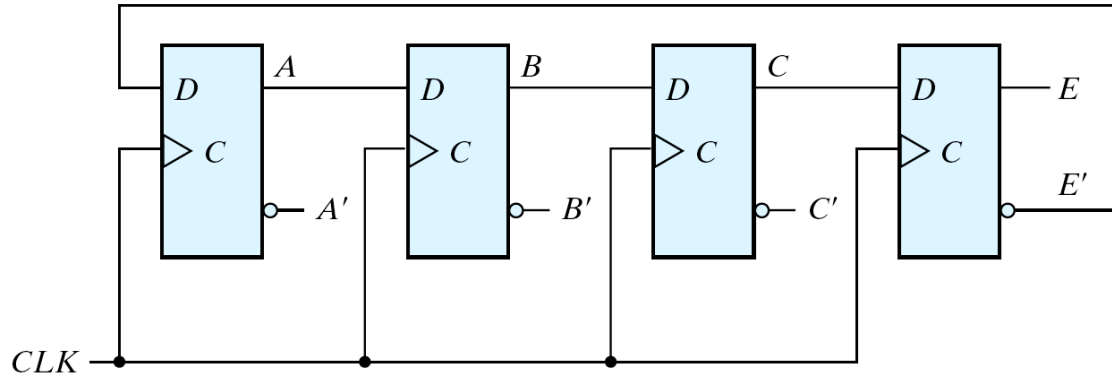
◆ Switch-tail ring counter

- » It is a circular shift register with the complement output of the last flip-flop connected to the input of the first flip-flop
- » A k -bit switch-tail ring counter will go through a sequence of $2k$ distinguishable states (initial value = 0 0 ... 0)

State	Q0	Q1	Q2	Q3
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	1	1	1	1
5	0	1	1	1
6	0	0	1	1
7	0	0	0	1
0	0	0	0	0

Johnson Counter (2/4) (p.298, 299)

■ An example: Switch-tail ring counter



(a) Four-stage switch-tail ring counter

Sequence number	Flip-flop outputs				AND gate required for output
	A	B	C	E	
1	0	0	0	0	$A'E'$
2	1	0	0	0	AB'
3	1	1	0	0	BC'
4	1	1	1	0	CE'
5	1	1	1	1	AE
6	0	1	1	1	$A'B$
7	0	0	1	1	$B'C$
8	0	0	0	1	$C'E$

(b) Count sequence and required decoding

Fig. 6.18 Construction of a Johnson counter

Johnson Counter (3/4) (p.298, 299)

■ Johnson counter

- ◆ A k -bit switch-tail ring counter + $2k$ decoding gates
- ◆ Provide outputs for $2k$ timing signals

» E.g.: 4-bit Johnson counter

Sequence number	Flip-flop outputs				AND gate required for output
	A	B	C	E	
1	0	0	0	0	$A'E'$
2	1	0	0	0	AB'
3	1	1	0	0	BC'
4	1	1	1	0	CE'
5	1	1	1	1	AE
6	0	1	1	1	$A'B$
7	0	0	1	1	$B'C$
8	0	0	0	1	$C'E$

(b) Count sequence and required decoding

- ◆ The decoding follows a regular pattern
 - » 2 inputs per decoding gate

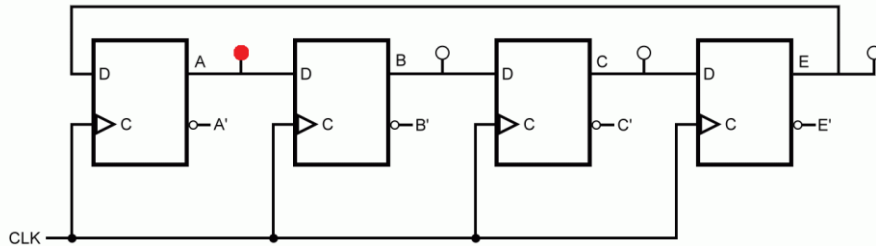
Johnson Counter (4/4) (p.298, 299)

❑ Disadvantage of the switch-tail ring counter

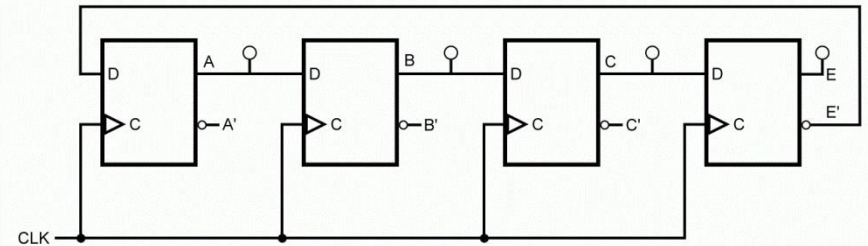
- ◆ If it finds itself in an unused state, it will persist to circulate in the invalid states and never find its way to a valid state
- ◆ One correcting procedure: $D_C = (A + C) B$

❑ Summary

- ◆ Johnson counters can be constructed for any number of timing sequences
 - » Number of flip-flops = 1/2 (the number of timing signals)
 - » Number of decoding gates = number of timing signals 2-input per gate



4-bit ring counter



4-bit Johnson Counter