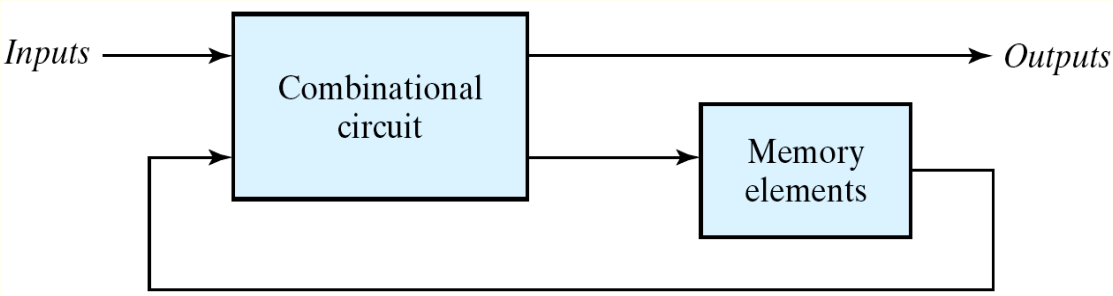
Chapter 6

Registers and Counters

Sequential Circuits

- **Clocked sequential circuits (循序,序向,順序)**
  - **a group of flip-flops and combinational gates**
  - **connected to form a feedback path**

Flip-flops + Combinational gates
(essential) (optional)



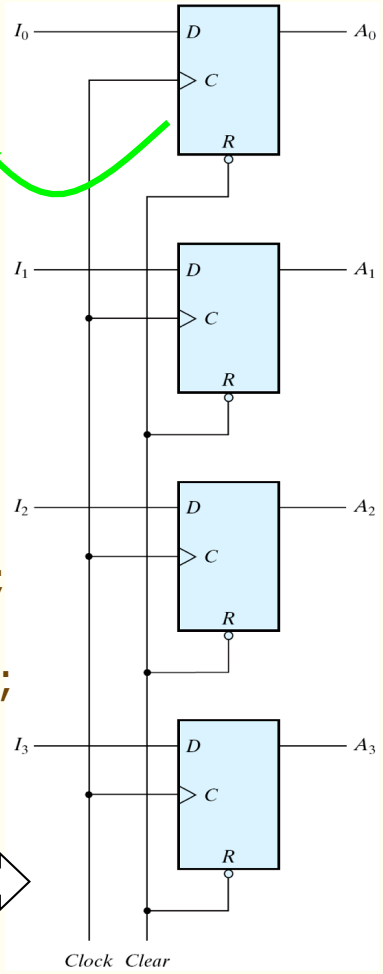
Registers

Register:

- a group of flip-flops (正反器)
- gates that determine how the information is transferred into the register

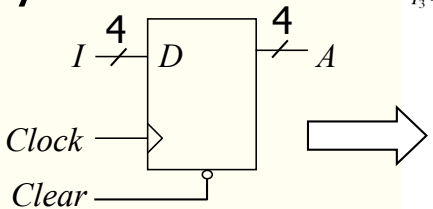
Function table

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



A n-bit register

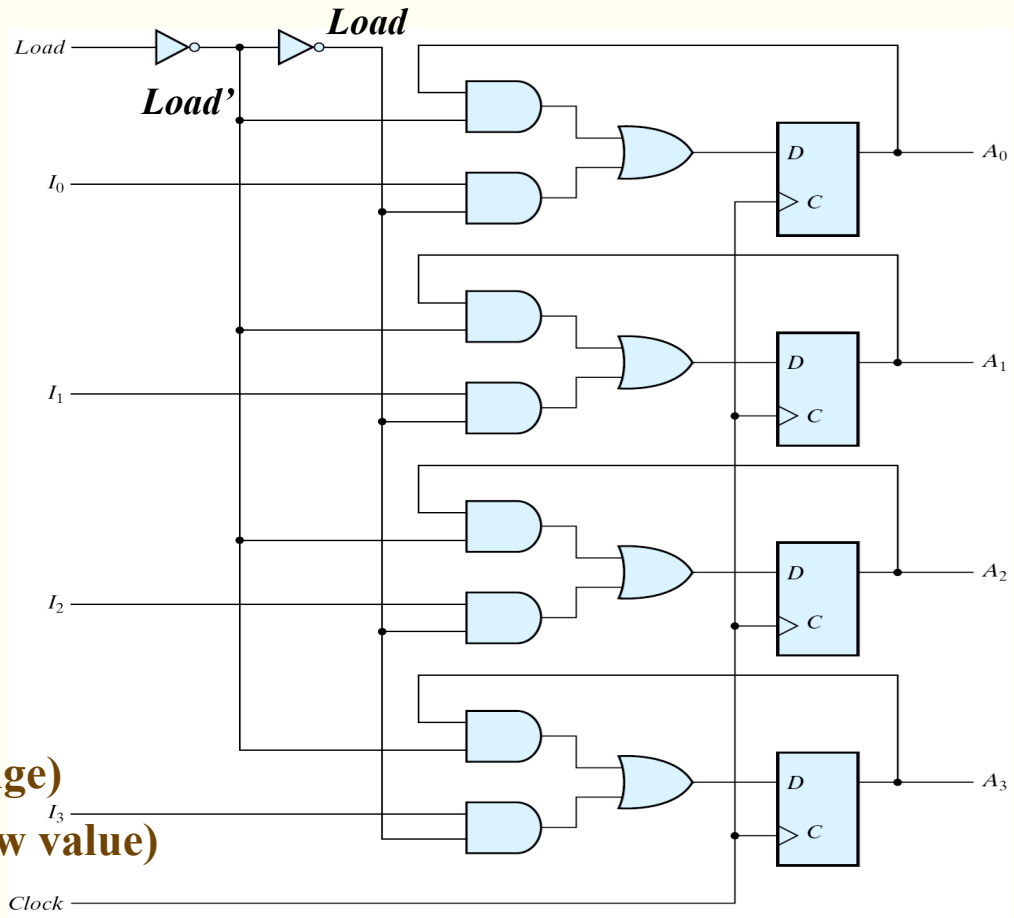
- n flip-flops capable of storing n bits of binary information
- 4-bit register



If Clear=0
clear (reset);
else
normal node;

4-bit register with parallel load

Four-bit register
with parallel load



Load	*Clock*	*A(t+1)*	
0	↑	*A(t)*	(no change)
1	↑	*I*	(load new value)

Shift Registers (1/2)

- **Shift register**
  - **a register capable of shifting its binary information in one or both directions**
- **Simplest shift register**

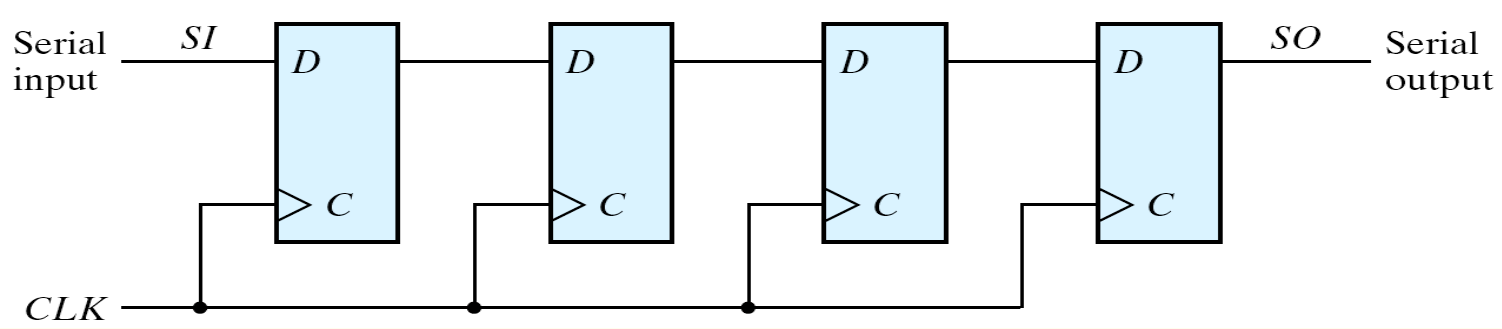
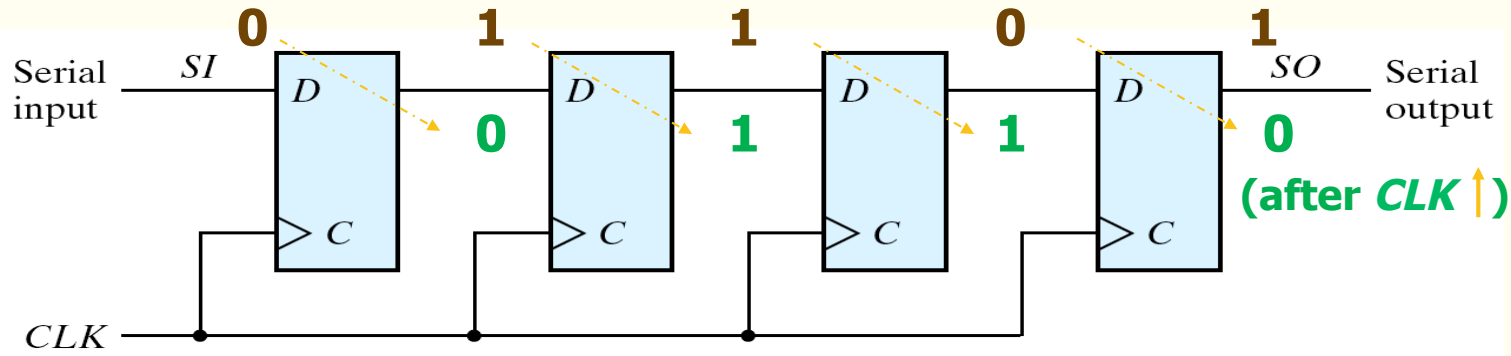


Fig. 6.3 Four-bit shift register

Shift Registers (2/2)



Note: output value is the same if no clock comes

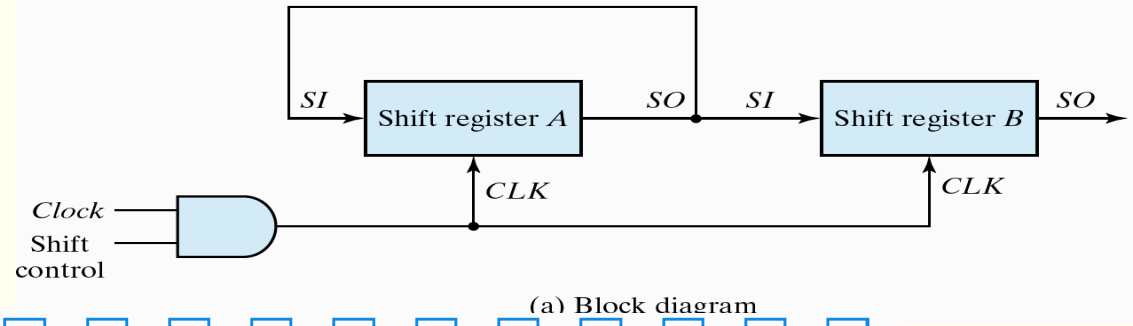
- A chain of FFs in cascade
- All FFs receive common clock pulses that activate the shift from one stage to the next
- Each clock pulse shifts the contents of the register one bit position to the right
- Serial input determines what goes into the leftmost FF during the shift
- Serial output is taken from the output of the rightmost FF.

Shifter

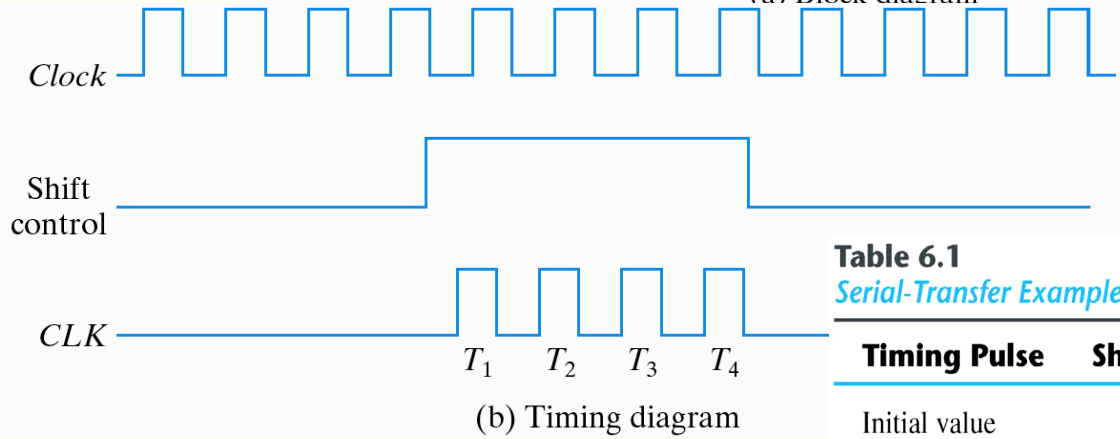
Serial transfer vs. Parallel transfer

- 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 from reg A to reg B



(a) Block diagram



(b) Timing 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

A=A+B

If both A and B are 4-bit SRs, the correct result will come out after 5 cycles. (serial input=0)

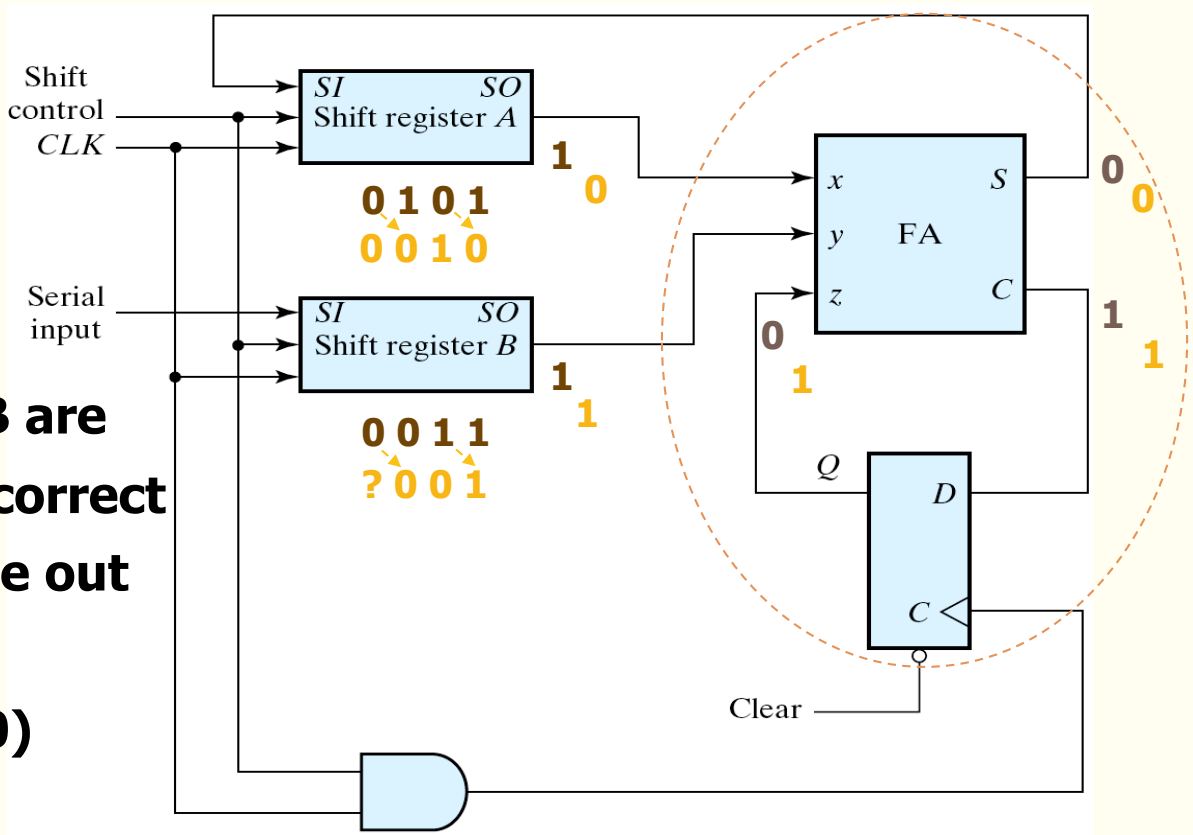


Fig. 6.5
Serial adder

Serial adder using JK flip-flops

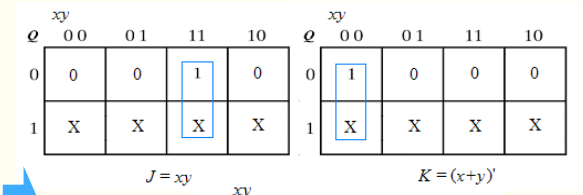
carry

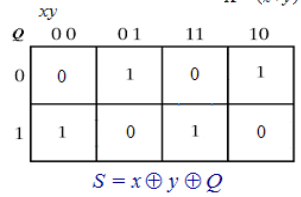
Table 6.2
State Table for Serial Adder

Present State	Inputs		Next State	Output	Flip-Flop Inputs	
Q	x	y	Q	S	J_Q	K_Q
0	0	0	0	0	0	X
0	0	1	0	1	0	X
0	1	0	0	1	0	X
0	1	1	1	0	1	X
1	0	0	0	1	X	1
1	0	1	1	0	X	0
1	1	0	1	0	X	0
1	1	1	1	1	X	0

Excitation Table

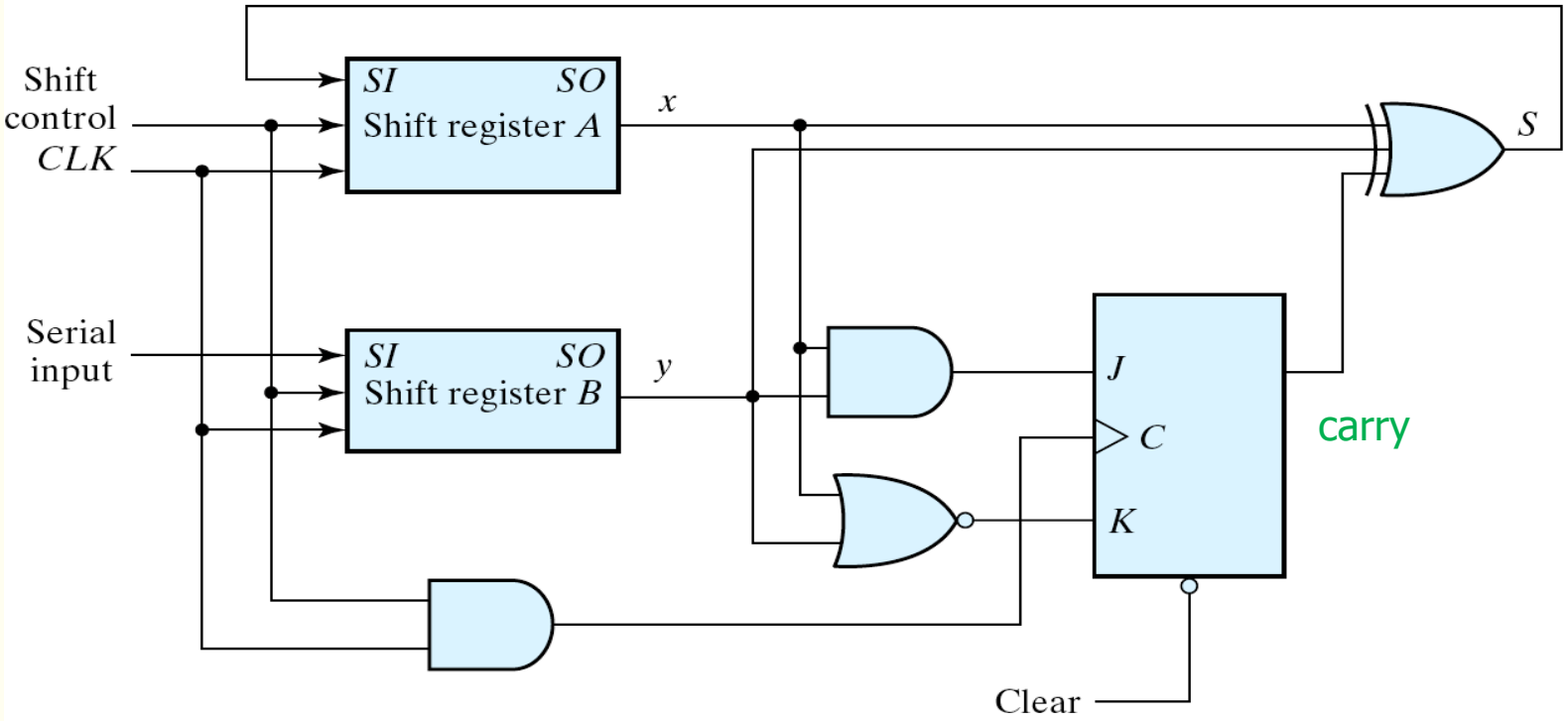
Q(t)	Q(t+1)	J	K
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0





Circuit Diagram

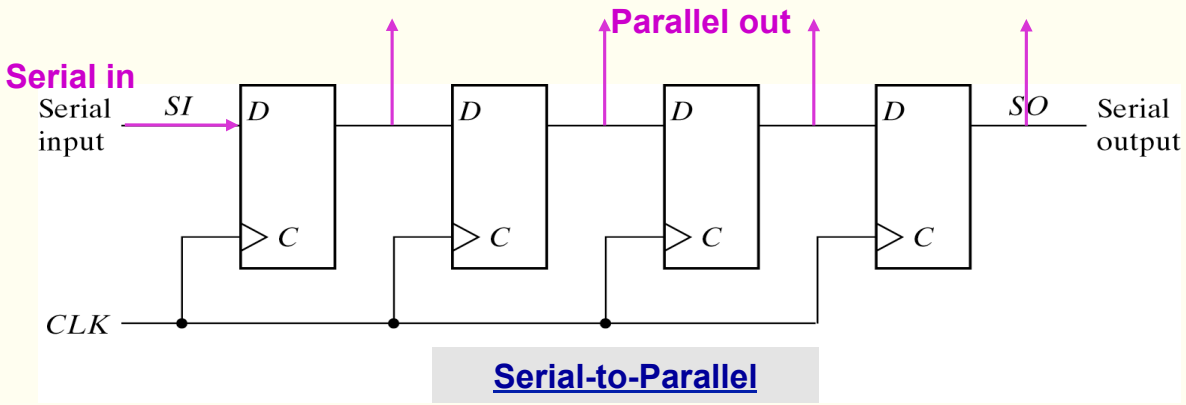
$$JQ = x\,y$$
$$KQ = x'\,y' = (x + y)' \qquad S = x \oplus y \oplus Q$$



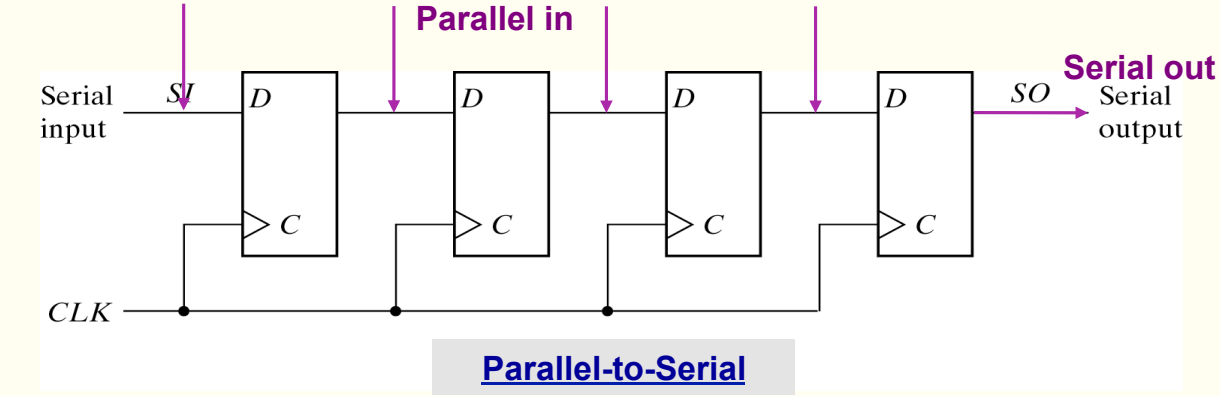
Universal Shift Register

- **Universal Shift Register**
  - **Unidirectional shift register**
  - **Bidirectional shift register**
  - **Universal shift register:**
    - **has both direction shifts & parallel load/out capabilities**

Parallel Load + Shift

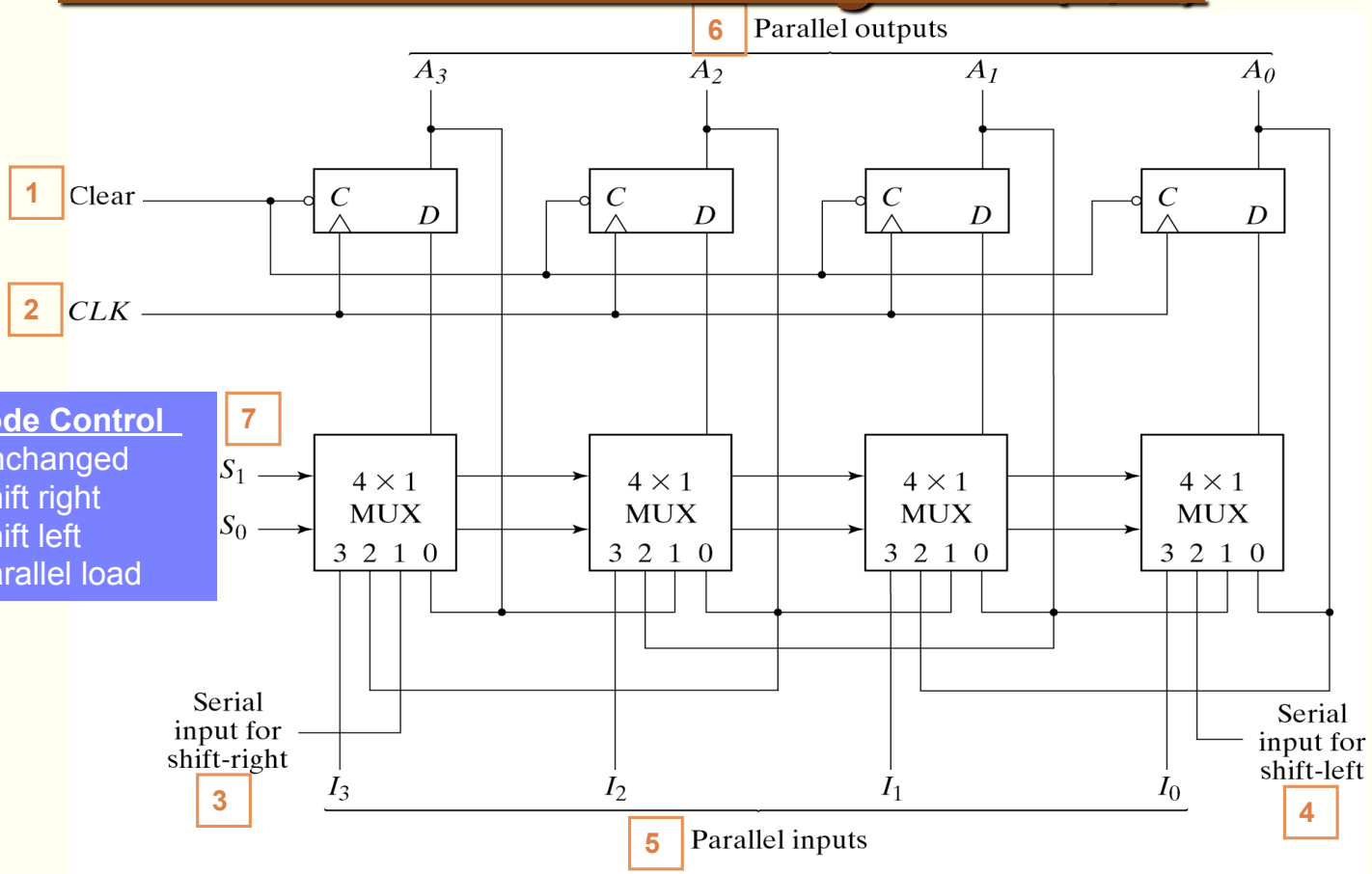


Serial-to-Parallel



Parallel-to-Serial

4-Bit Universal Shift Register (1/2)



S_1S_0	Mode Control
0 0	(0) Unchanged
0 1	(1) Shift right
1 0	(2) Shift left
1 1	(3) Parallel load

Fig. 6-7 4-Bit Universal Shift Register

4-Bit Universal Shift Register (2/2)

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

Ripple Counters

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 (⇒ Binary Counter) or any other sequence of states

Counter

Categories of counters

1. **Ripple counters**

   The flip-flop output transition serves as a source for triggering other flip-flops

   ⇒ no common clock pulse (not synchronous)

2. **Synchronous counters:**

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

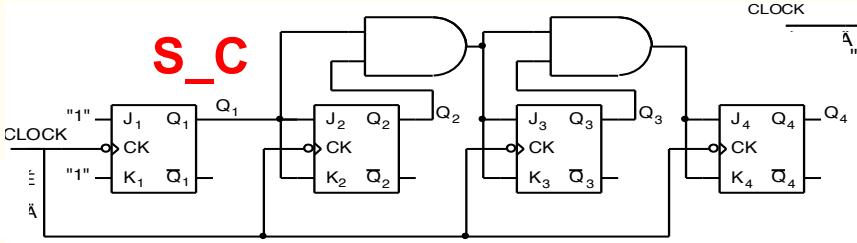
Synchronous/Asynchronous Counter

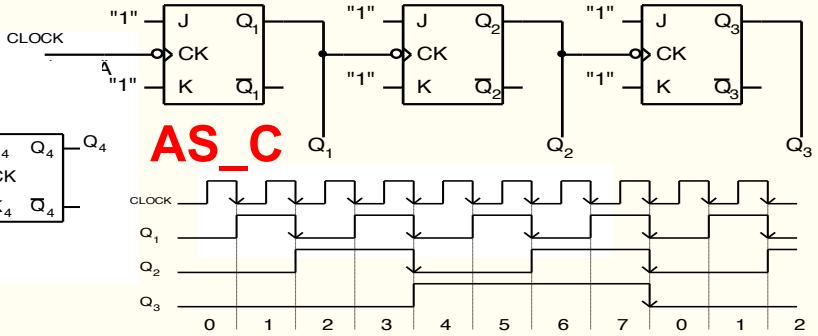
Synchronous counter:

All flip-flops in a synchronous counter receive the same clock pulse and so change state simultaneously.

Asynchronous (Ripple) counter:

Flip-flops transitions ripple through from one flip-flop to the next in sequence until all flip-flops reach a new stable value (state). Each single flip-flop stage divides the frequency of its input signal by two.

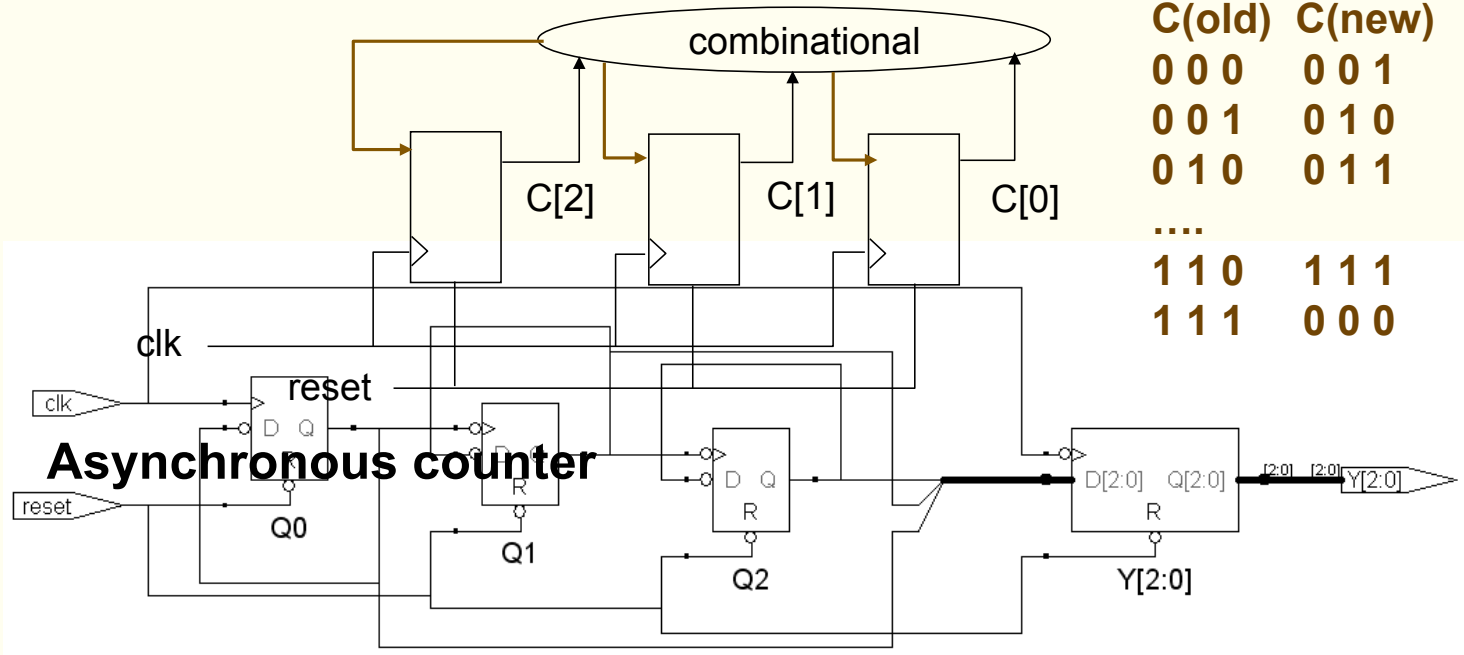




0	0	0	0
0	0	0	1
0	0	1	0

Counter Implementation

Synchronous counter



C(old)	C(new)
0 0 0	0 0 1
0 0 1	0 1 0
0 1 0	0 1 1
....	
1 1 0	1 1 1
1 1 1	0 0 0

Asynchronous counter

Synchronous Counters

- **Synchronous Counter**
  - **A common clock triggers all flip-flops simultaneously**
- **Design procedure**
  - **apply the same procedure of synchronous sequential circuits**
  - **synchronous counter is simpler than general synchronous sequential circuits**

4-bit Binary Counter

0 0 1 1
0 1 0 0

A_0=1, A_1=1, A_2=0, A_3=0

➔ JA_0=1, KA_0=1

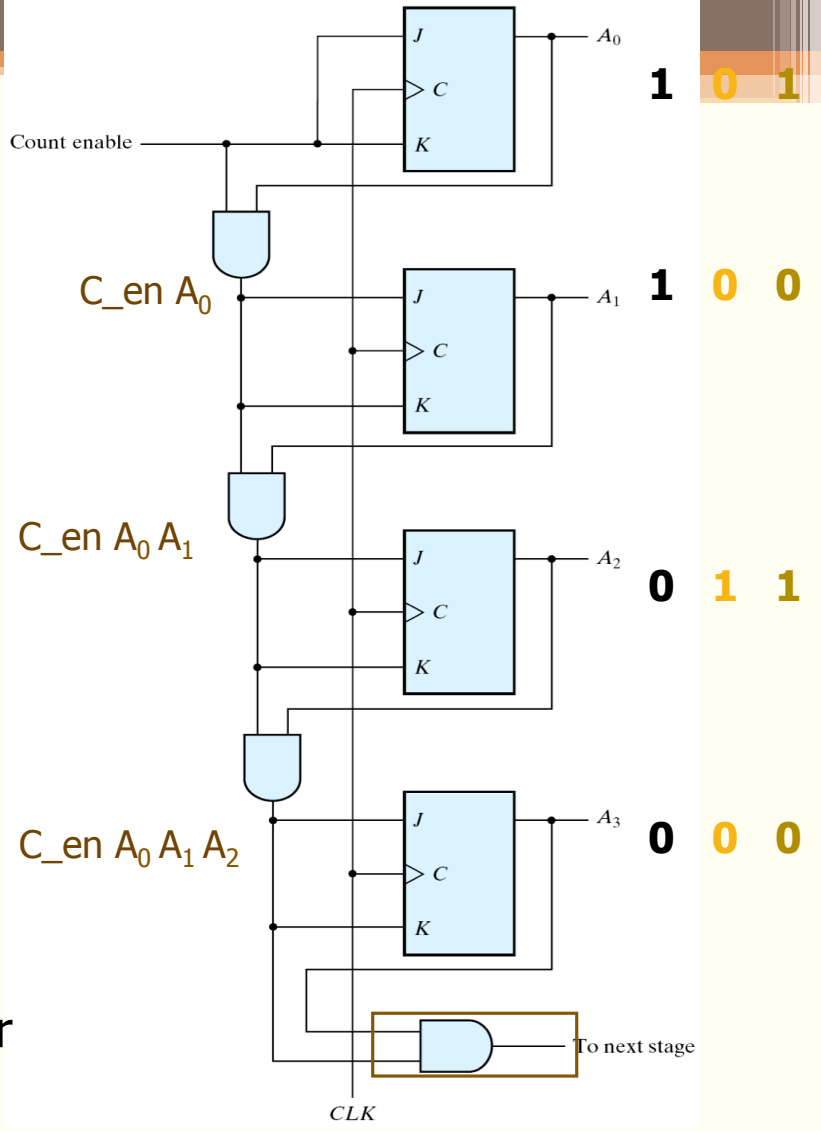
$C_{en}A_0$=1 ➔ JA_1=1, KA_1=1

$C_{en}A_0A_1$=1 ➔ JA_2=1, KA_3=1

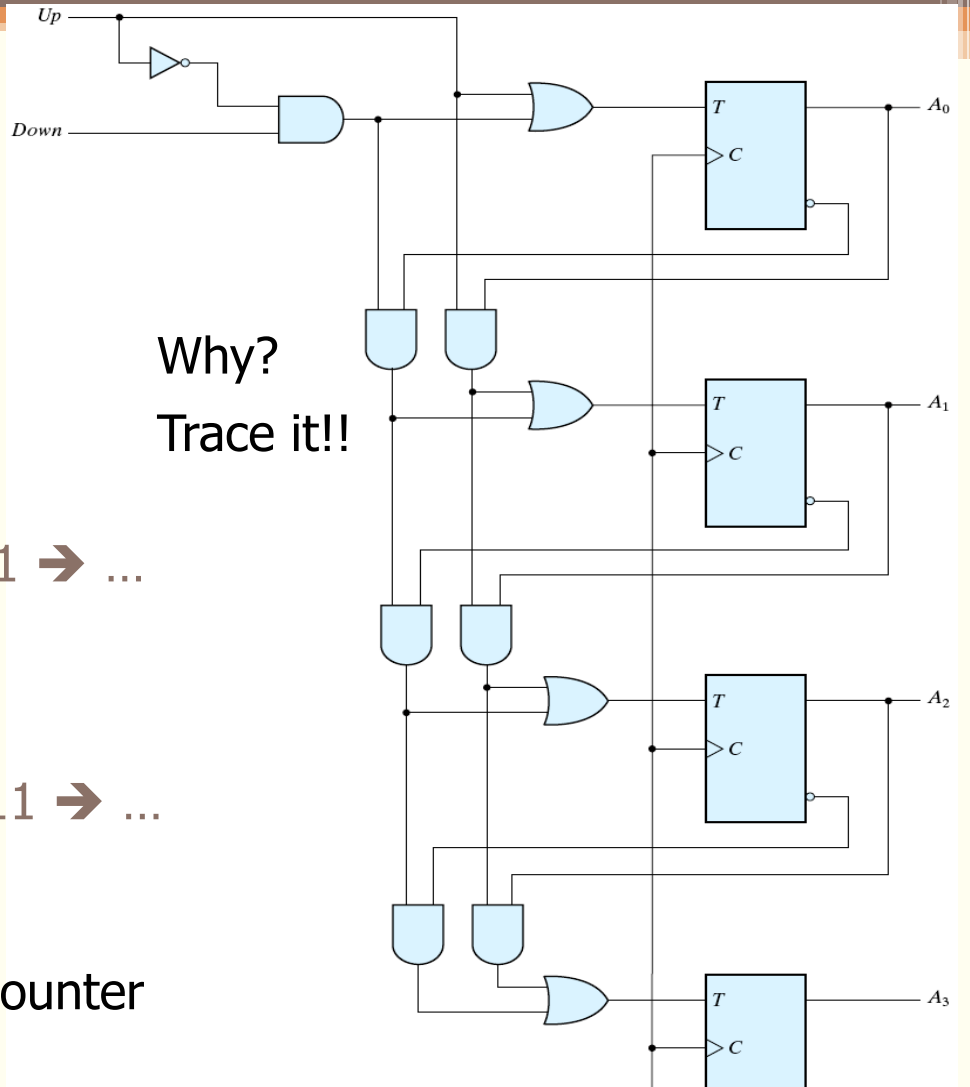
$C_{en}A_0A_1A_2$=0 ➔ JA_3=0, KA_3=0

↑ A_0=0, A_1=0, A_2=1, A_3=0

Four-bit synchronous binary counter



4-bit Up/Down Binary Counter



Why?

Trace it!!

Up Counter

010➔ 011➔ 100➔ 101 ➔ …

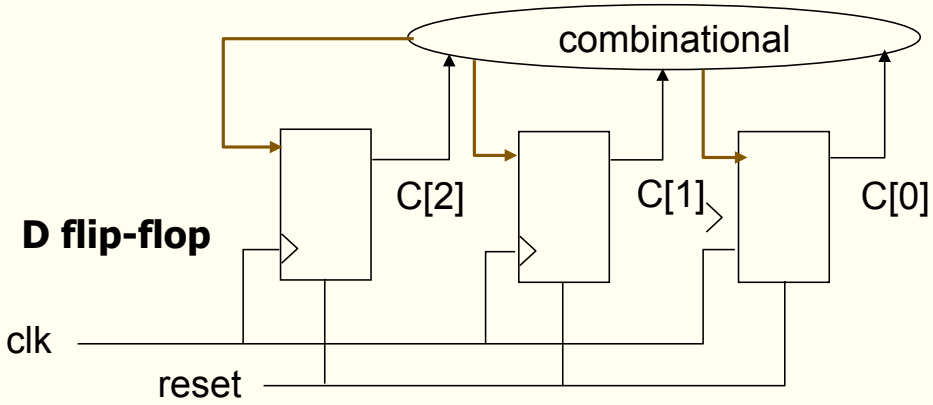
Down Counter

110➔ 101➔ 100➔ 011 ➔ …

Four-bit up-down binary counter

Synchronous Counter

Synchronous Counter



C(old)	C(new)
0 0 0	0 0 1
0 0 1	0 1 0
0 1 0	0 1 1
….	
1 1 0	1 1 1
1 1 1	0 0 0

Draw three K-maps

0 → 1 → 2 → 3 → 4 → 5 → 6 → 7 → 0 → 1 → … . .

C(old)	C(new)
0 0 0	0 0 1
0 0 1	0 1 0
0 1 0	0 1 1
. . .	
. . .	1 1 0
1 1 0	0 0 0
0 0 0	0 0 1

0 → 1 → 2 → 3 → 4 → 5 → 6 → 0 → 1 → 2 → …..

Draw three K-maps

Other Counters (1/2)

- **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**

Other Counters (2/2)

- 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**

Table 6.7
State Table for Counter

0 → 1 → 2 → 4 → 5 → 6 → 0 → 1 → 2 → .....

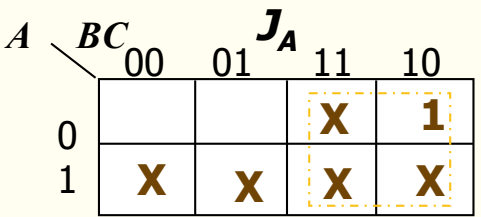
Present State			Next State			Flip-Flop Inputs					
A	B	C	A	B	C	J_A	K_A	J_B	K_B	J_C	K_C
0	0	0	0	0	1	0	X	0	X	1	X
0	0	1	0	1	0	0	X	1	X	X	1
0	1	0	1	0	0	1	X	X	1	0	X
1	0	0	1	0	1	X	0	0	X	1	X
1	0	1	1	1	0	X	0	1	X	X	1
1	1	0	0	0	0	X	1	X	1	0	X

An Example Two unused states: 011 & 111

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
0	**1**	**1**				**X**	**X**	**X**	**X**	**X**	**X**
1	**1**	**1**				**X**	**X**	**X**	**X**	**X**	**X**

$J_A = B$ . . .



. .

Logic Diagram

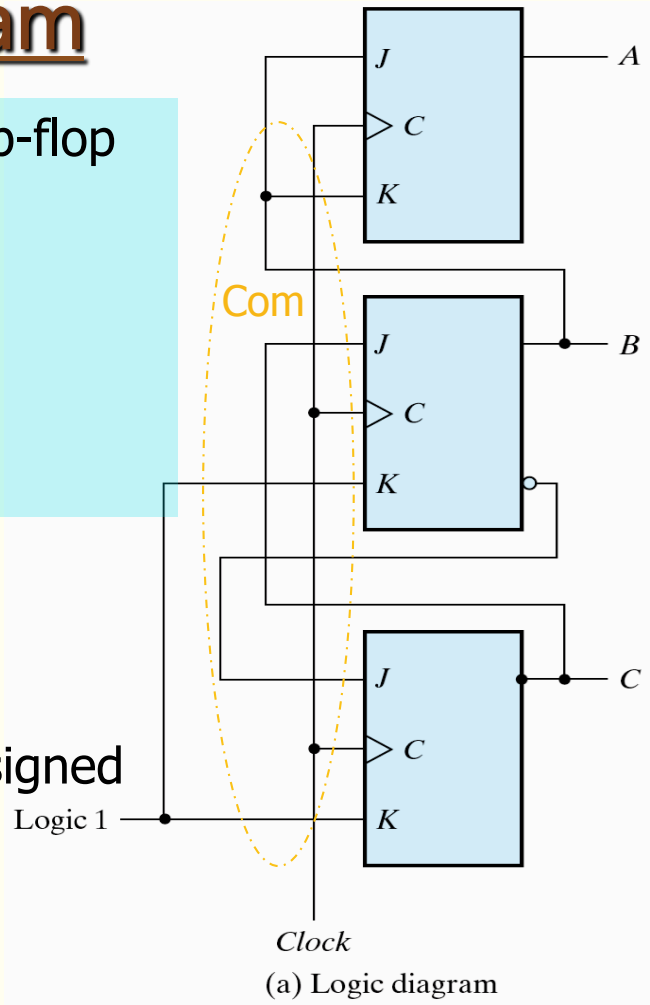
The simplified flip-flop input equations:

$J_A = B, \quad K_A = B$

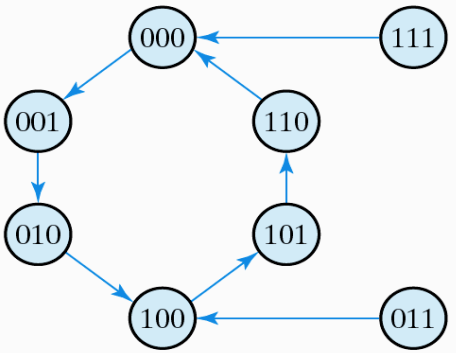
$J_B = C, \quad K_B = 1$

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

Counter with unsigned states



(a) Logic diagram



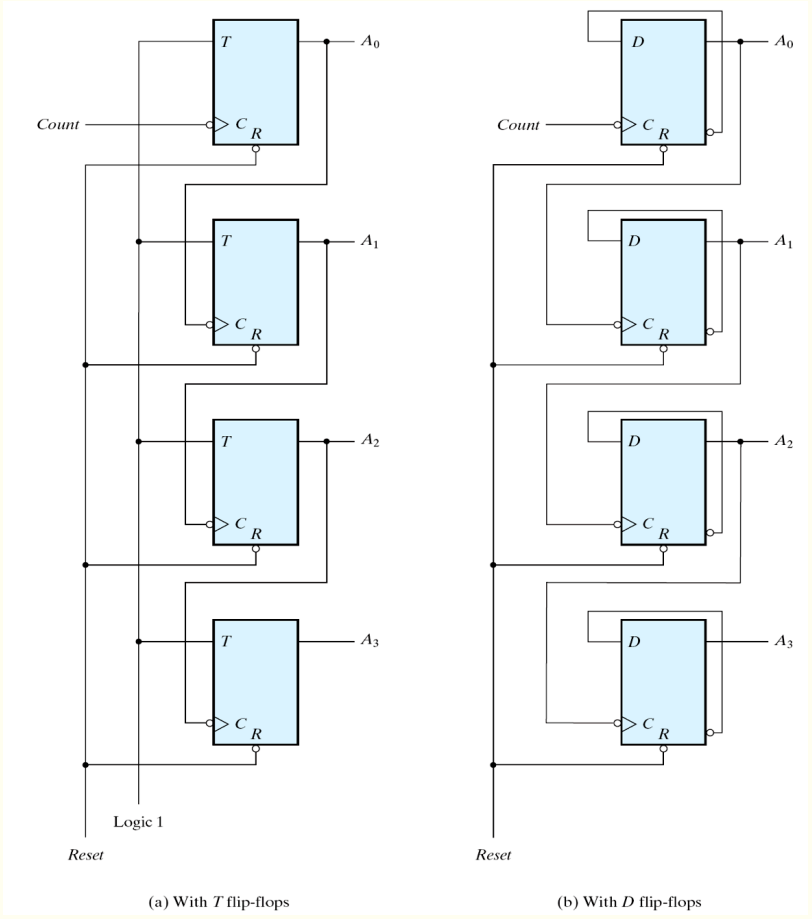
(b) State diagram

Four-bit Binary Ripple Counter

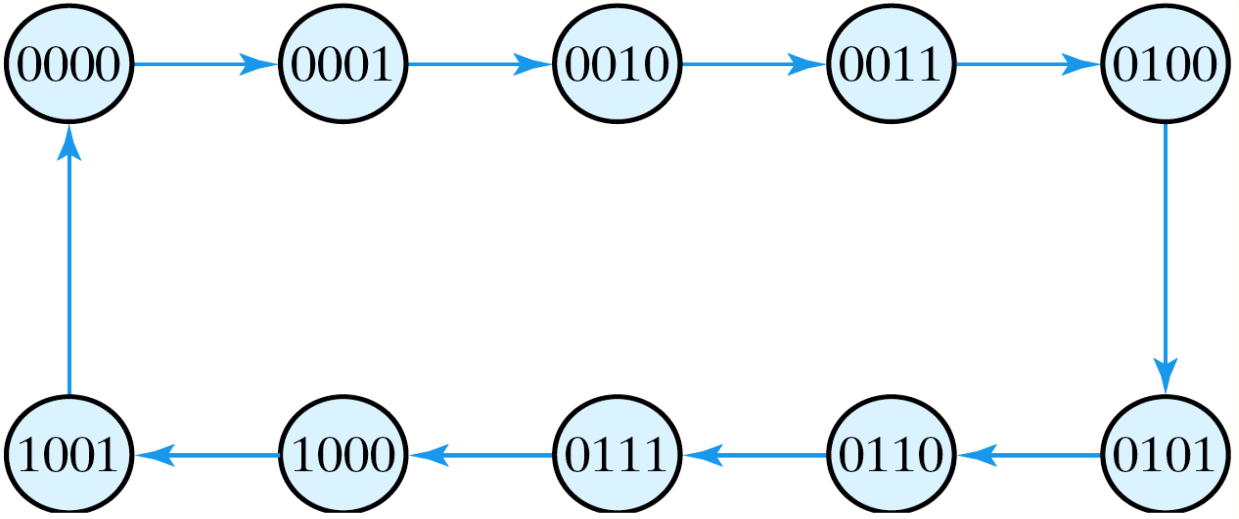
Table 6.4
Binary Count Sequence

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

Four-bit binary ripple counter

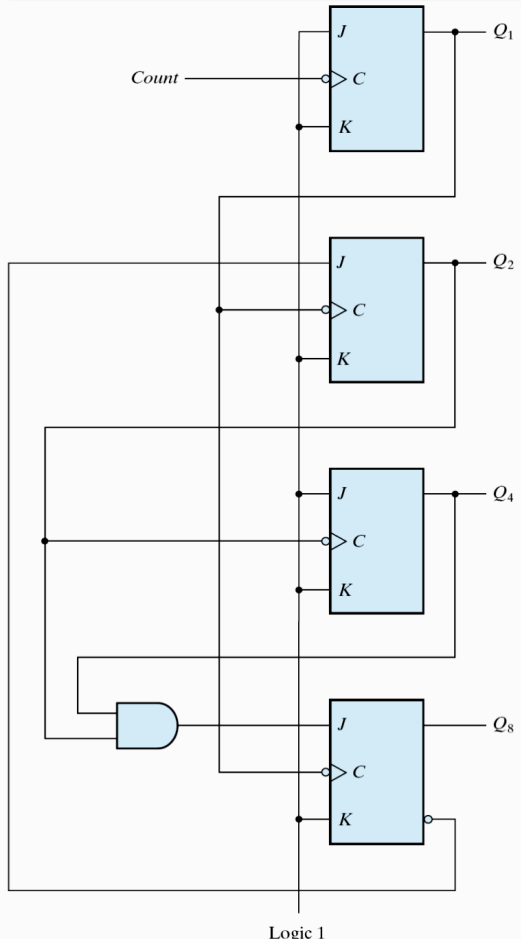


BCD ripple counter



State diagram of a decimal BCD counter

Circuit of BCD Ripple Counter



1 0 1

0 0 0

0 0 0

1 0 0

9 ➔ 10

Three-decade BCD counter

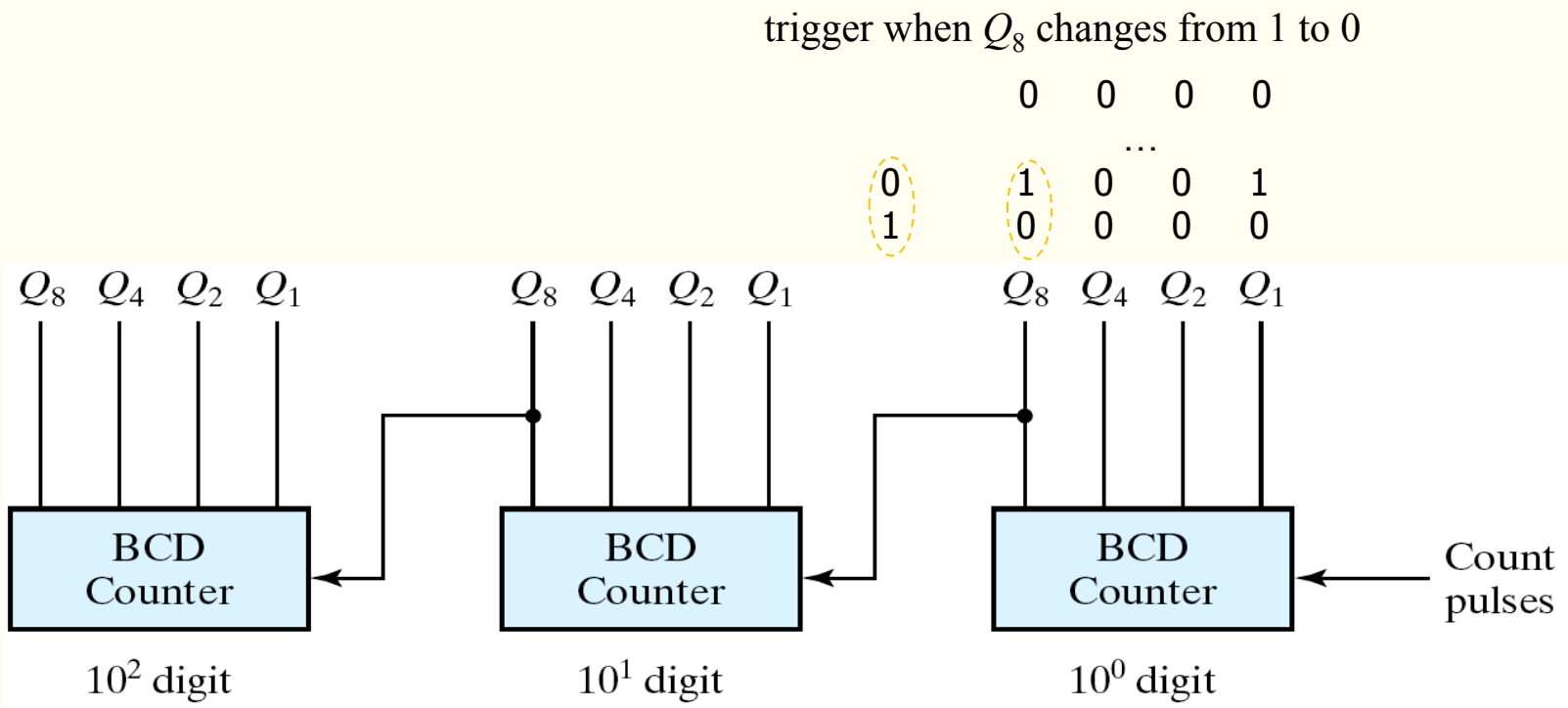


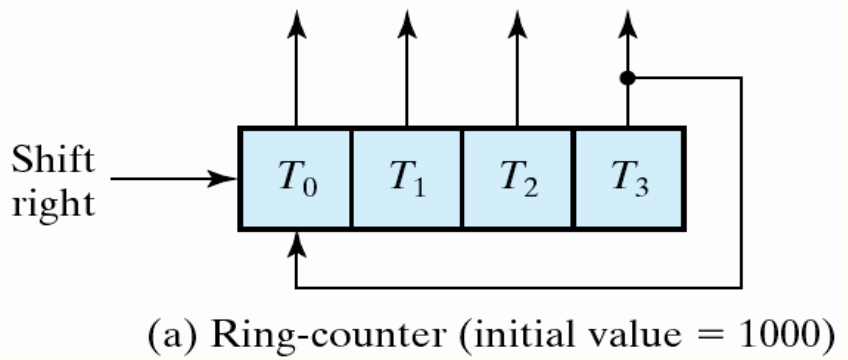
Fig. 6.11
Block diagram of a three-decade decimal BCD counter

Ring counter (1/2)

- a circular shift register w/ only one flip-flop being set at any particular time, all others are cleared

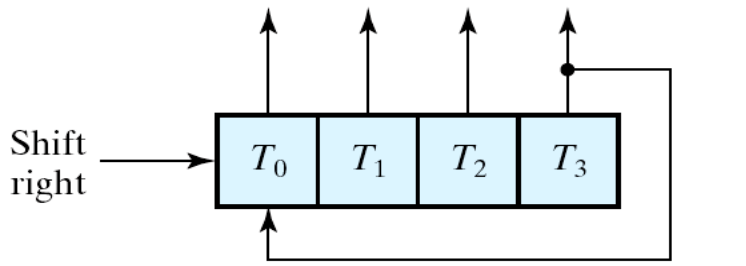
  (initial value = 1 0 0 … 0)
- The single bit is shifted from one flip-flop to the next to produce the sequence of timing signals.

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

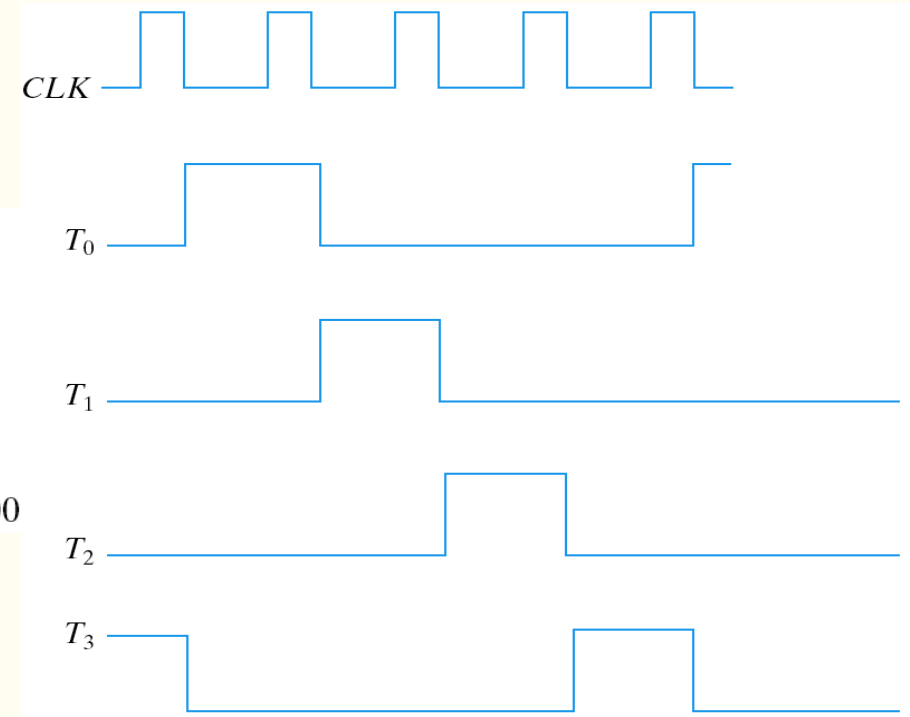


(a) Ring-counter (initial value = 1000)

Ring counter (2/2)



(a) Ring-counter (initial value = 1000



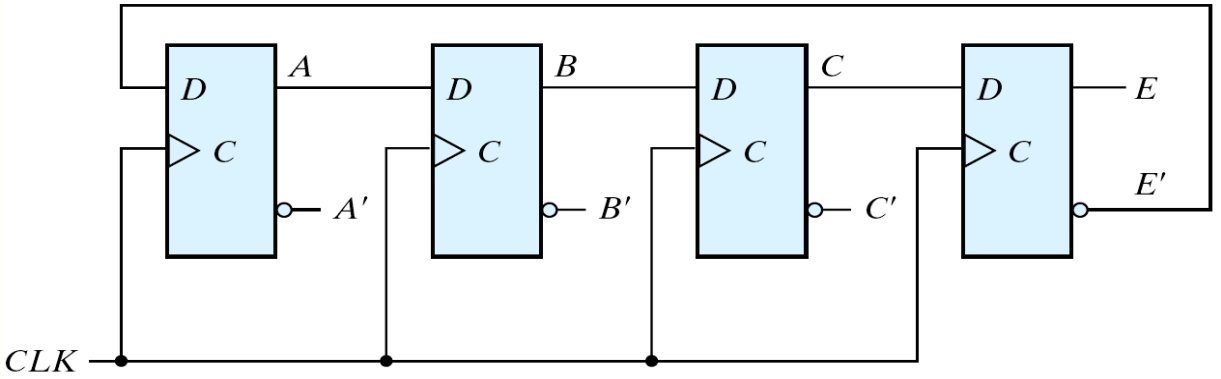
(b) Sequence of four timing signals

Johnson Counter

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.**
- **Switch-tail ring counter (Johnson Counter)**
  - **is a circular shift register w/ 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 2*k* distinguishable states. (initial value = 0 0 … 0)**

Johnson Counter



(a) Four-stage switch-tail ring counter

Sequence number	Flip-flop outputs A	B	C	E	AND gate required for output
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