ELECTRICAL
Engineering KMITNB

Registers and Counters

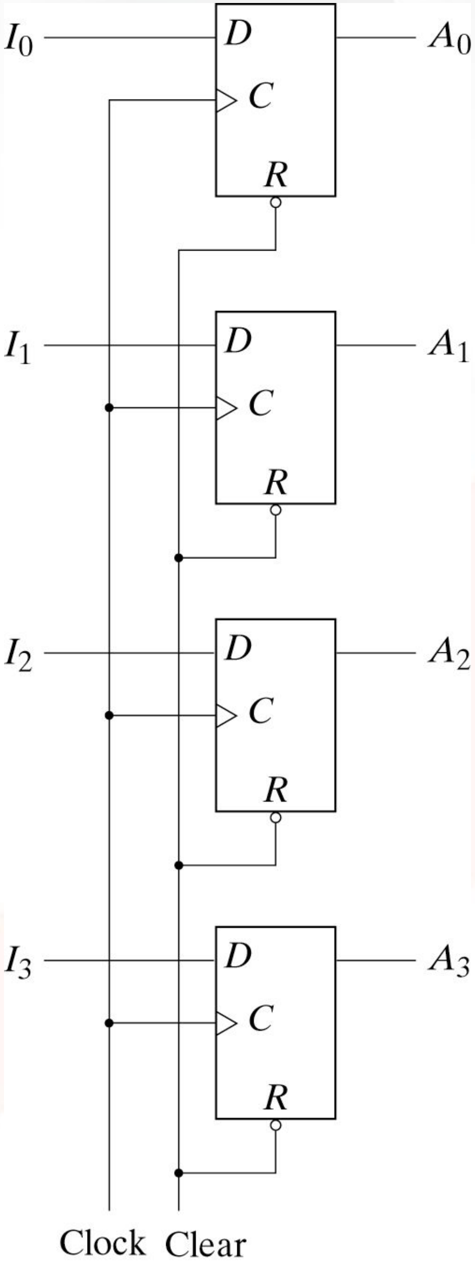
Chapter 7,8

- Clocked sequential circuits
  - a group of flip-flops and combinational gates
  - connected to form a feedback path

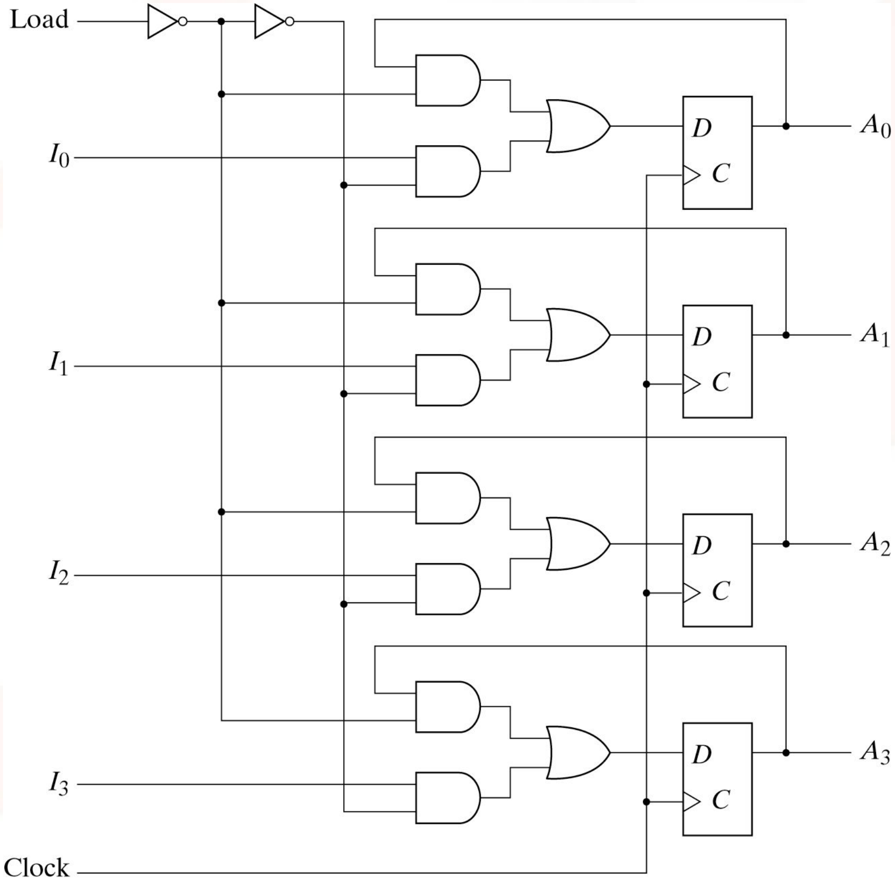
    Flip-flops + Combinational gates
- Register:
  - a group of flip-flops
  - gates that determine how the information is transferred into the register
- Counter:
  - a register that goes through a predetermined sequence of states

Registers

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

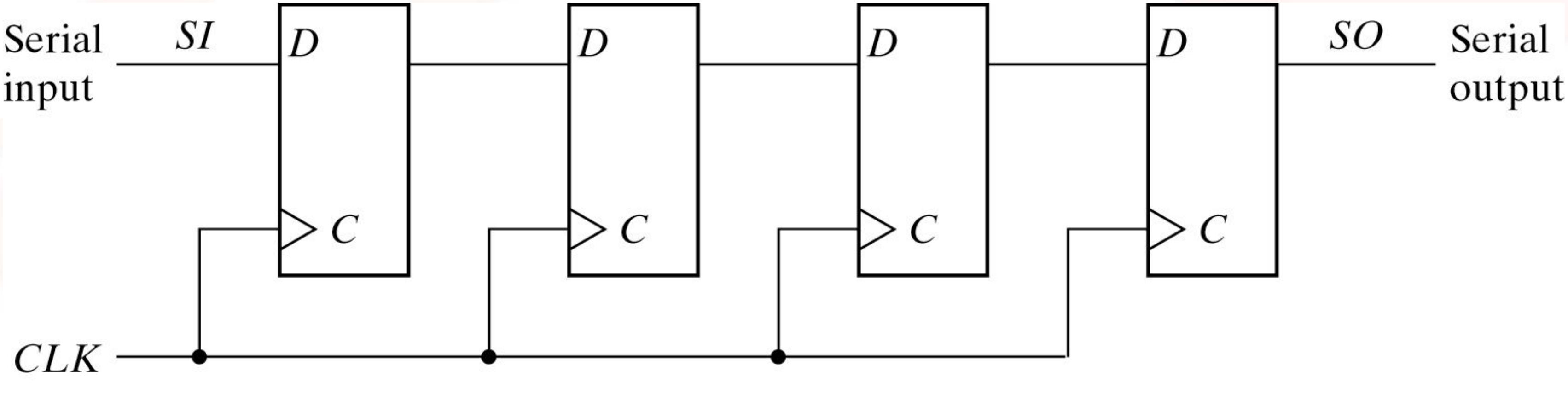


- 4-bit register with parallel load



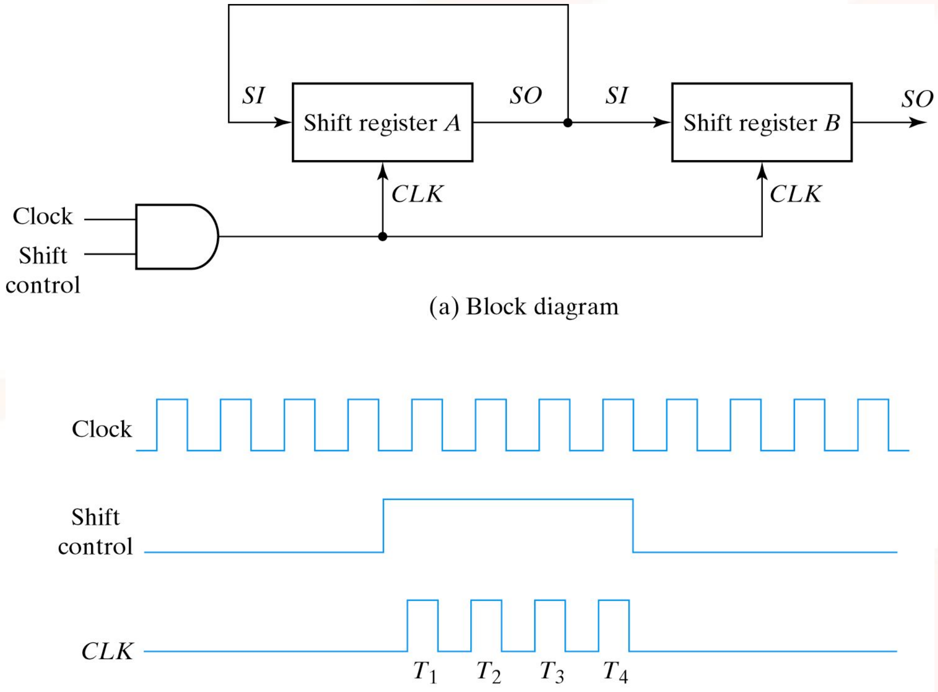
Shift Registers

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



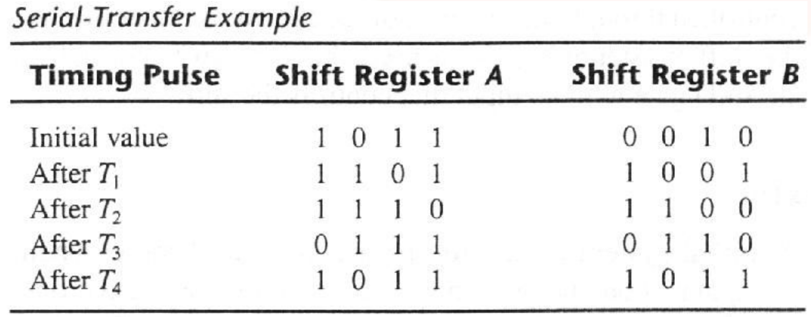
- Serial transfer vs. Parallel transfer
  - Serial transfer
    - Information is transferred one bit at a time
    - shifts the bits out of the source register into the destination register
  - Parallel transfer:
    - All the bits of the register are transferred at the same time

- Example: Serial transfer from reg A to reg B



(a) Block diagram

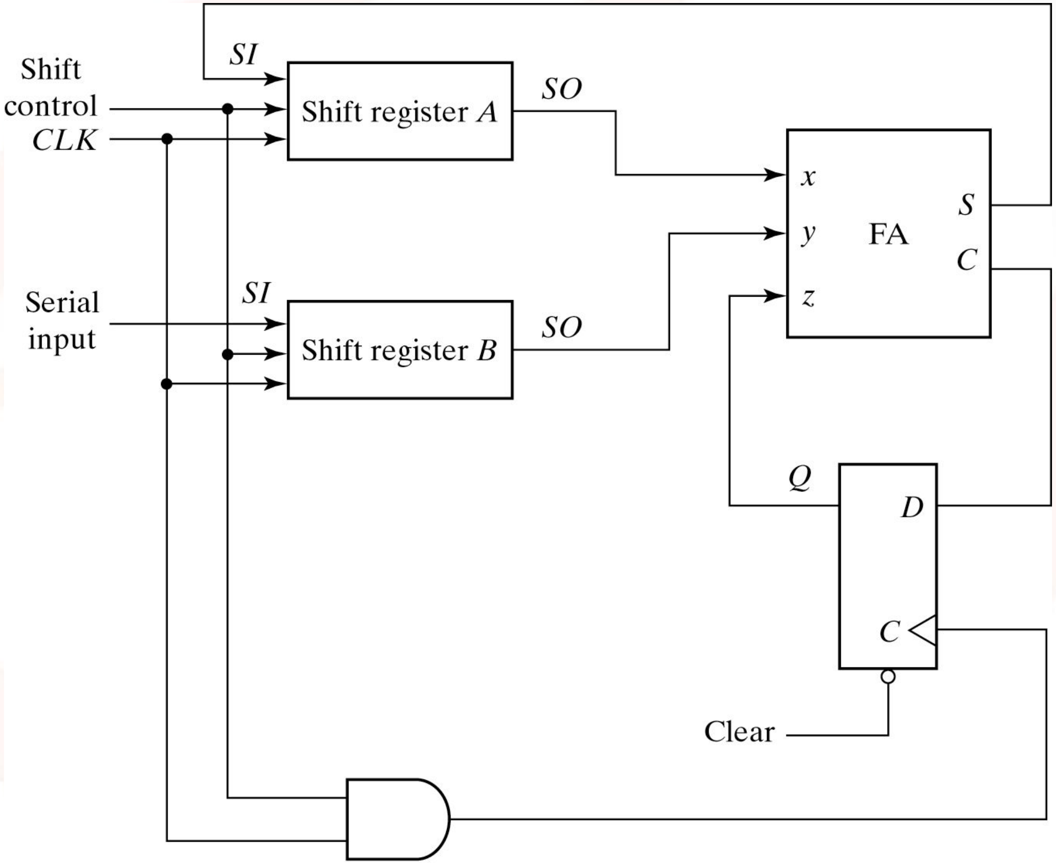
(b) Timing diagram



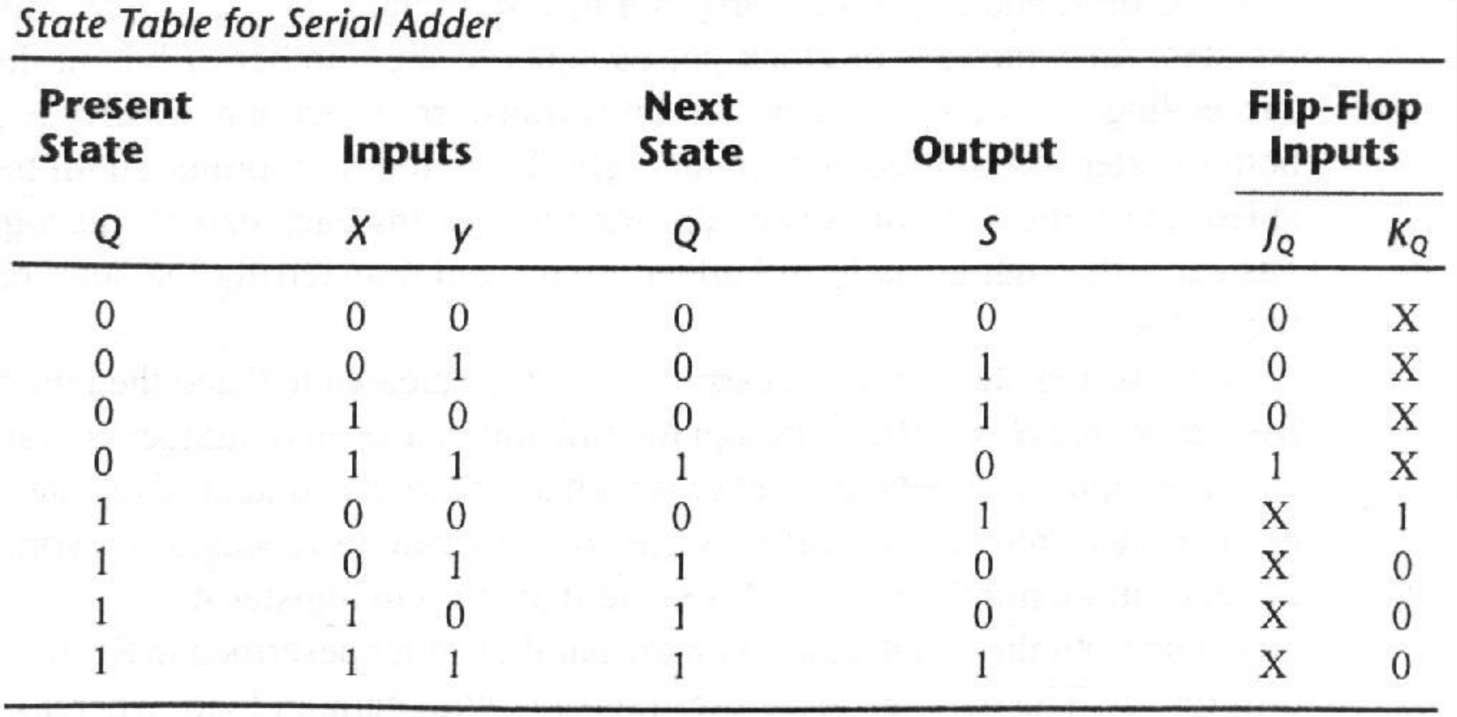
Serial-Transfer Example

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



- Serial adder using JK flip-flops



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

$$J_Q = x\,y$$

$$K_Q = x'\,y' = (x + y)'$$

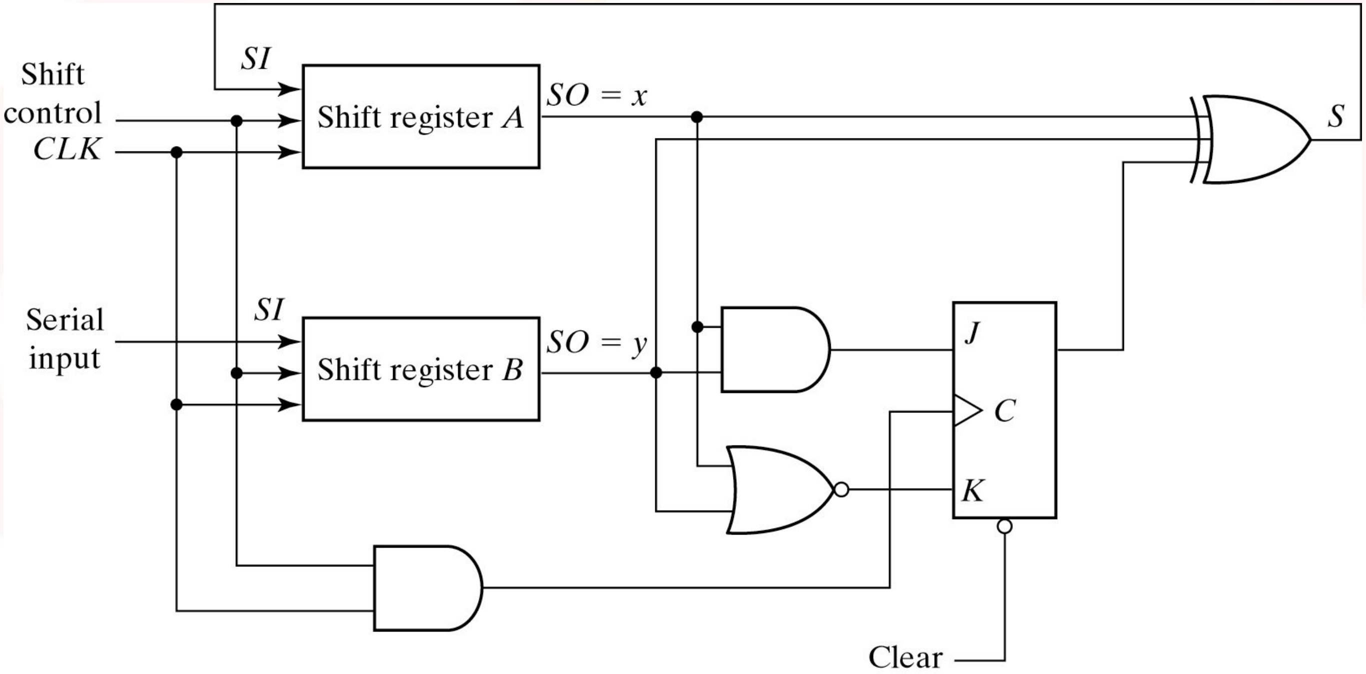
$$S = x \oplus y \oplus Q$$

- Circuit diagram

$J_Q = x\,y$

$K_Q = x'\,y' = (x + y)'$

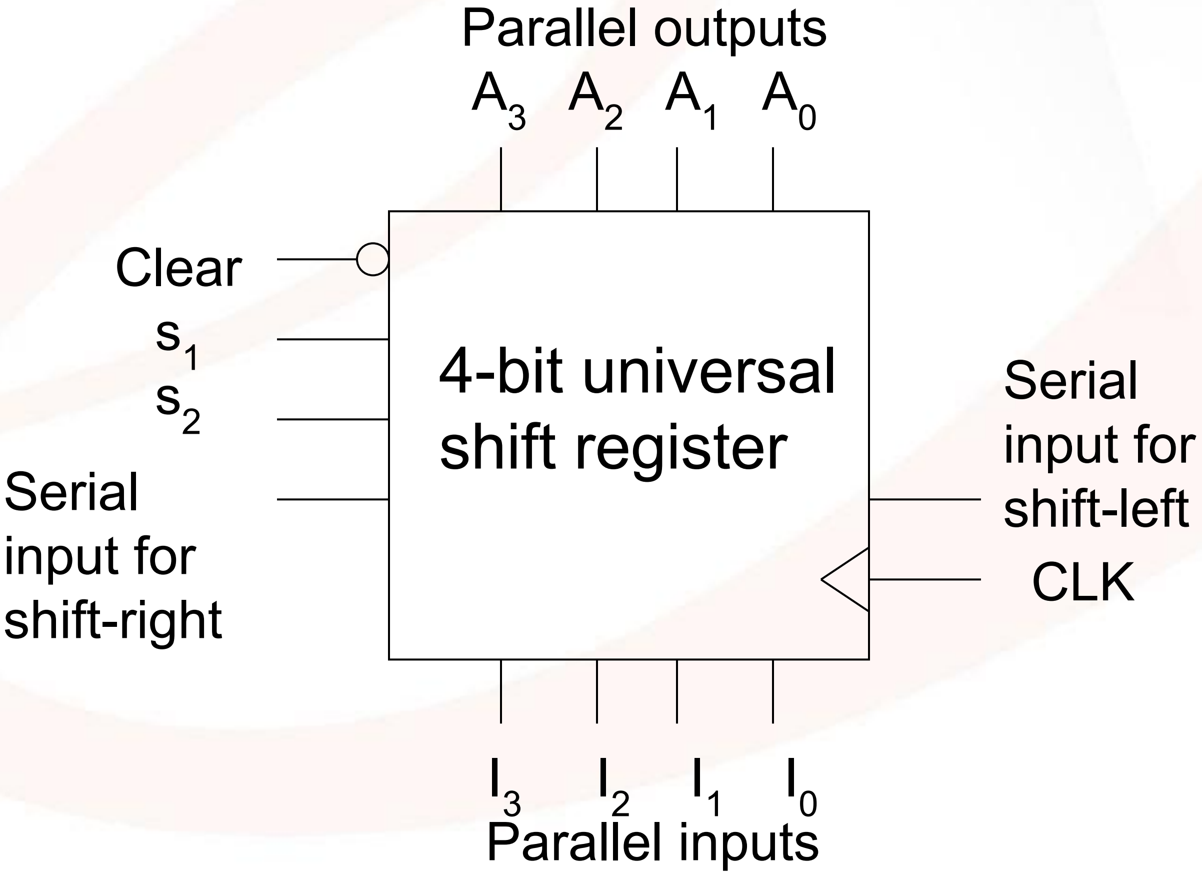
$S = x \oplus y \oplus Q$



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

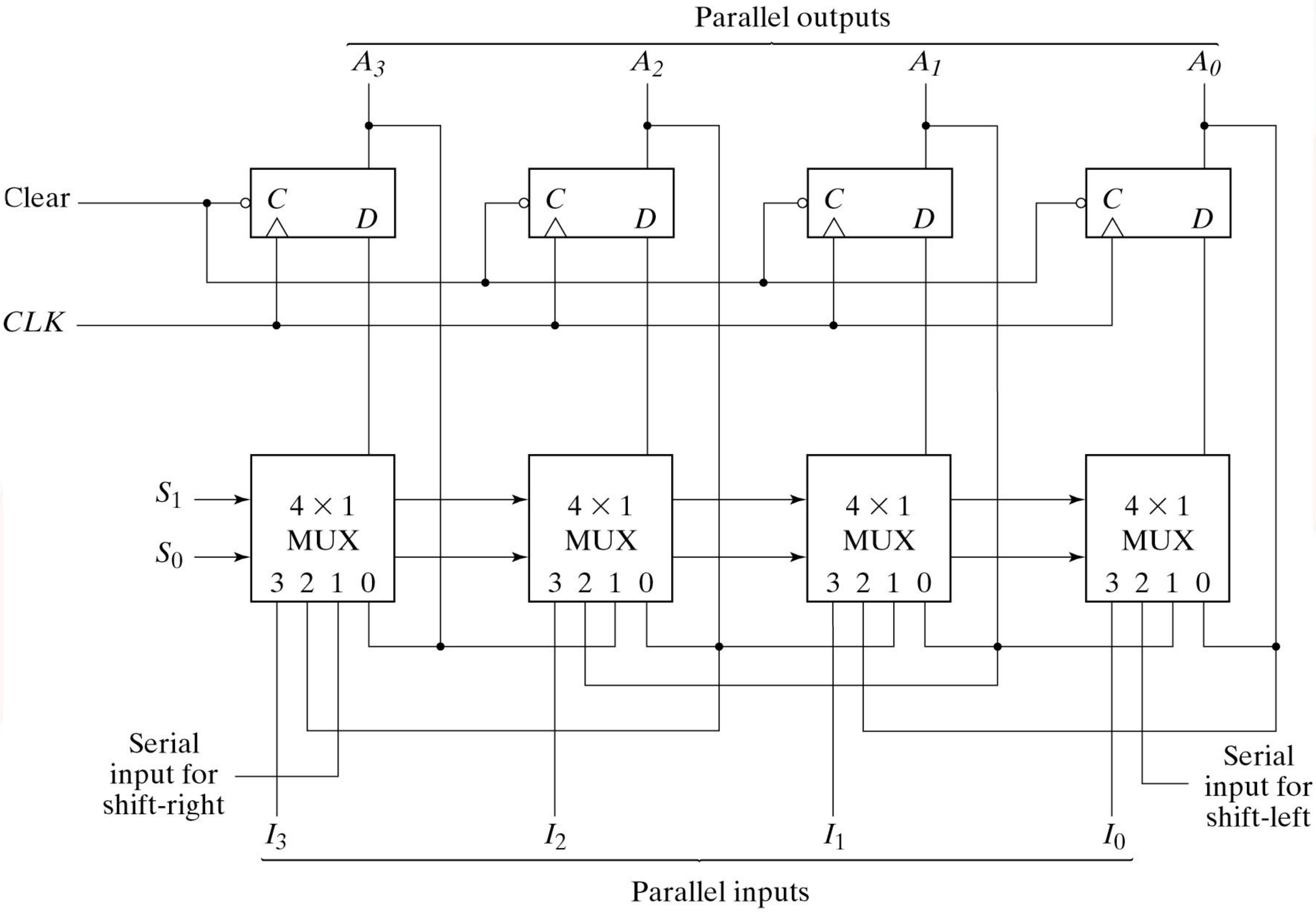
- Capability of a universal shift register:
  - A *clear* control to clear the register to 0.
  - A *clock* input to synchronize the operations.
  - A *shift-right* control to enable the shift right operation and the *serial input* and *output* lines associated w/ the shift right.
  - A *shift-left* control to enable the shift left operation and the *serial input* and *output* lines associated w/ the shift left.
  - A *parallel-load* control to enable a parallel transfer and the *n* *parallel input* lines associated w/ the parallel transfer.
  - *n* *parallel output* lines.
  - A control state that leaves the information in the register unchanged in the presence of the clock.

- Example: 4-bit universal shift register



- Function table

Clear	s_1	s_0	A_3^+	A_2^+	A_1^+	A_0^+	(operation)
0	×	×	0	0	0	0	Clear
1	0	0	A_3	A_2	A_1	A_0	No change
1	0	1	sri	A_3	A_2	A_1	Shift right
1	1	0	A_2	A_1	A_0	sli	Shift left
1	1	1	I_3	I_2	I_1	I_0	Parallel load


Parallel outputs
A_3
A_2
A_1
A_0
Clear
C
D
CLK
S_1
S_0
4 × 1
MUX
3 2 1 0
Serial input for shift-right
Serial input for shift-left
I_3
I_2
I_1
I_0
Parallel inputs

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

- Categories of counters

  a. Ripple counters

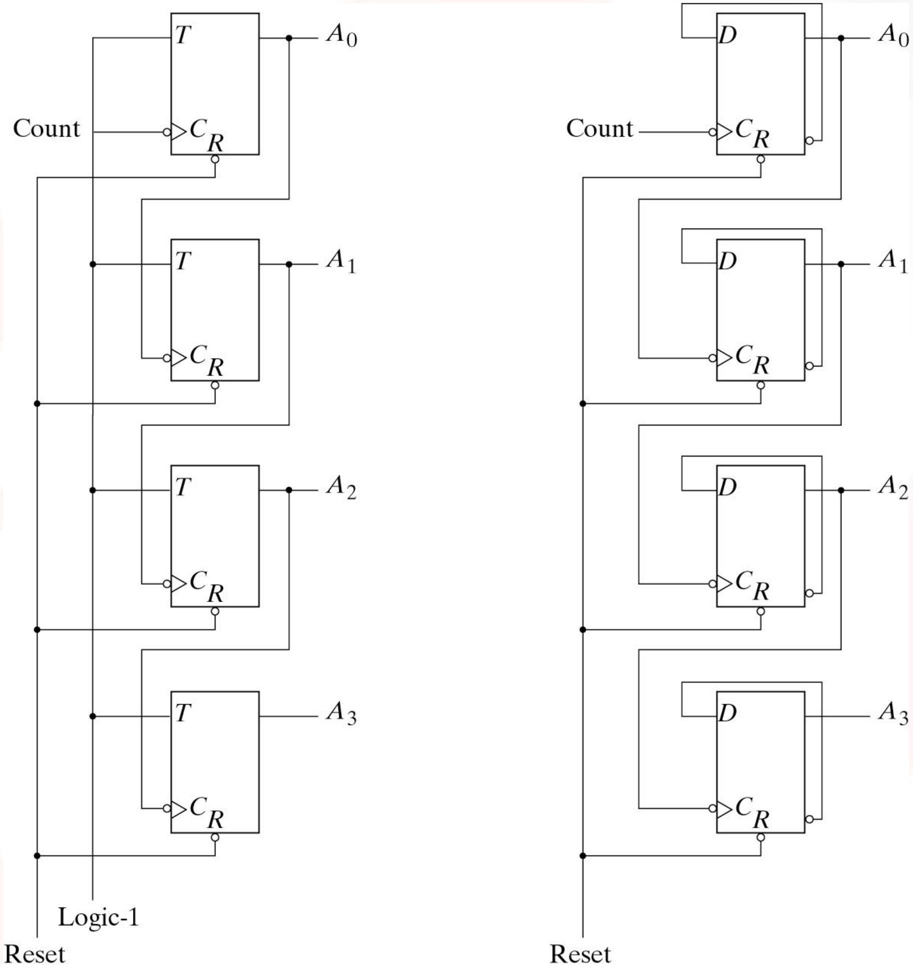
     i. The flip-flop output transition serves as a source for triggering other flip-flops

     ii. ⇒ no common clock pulse (not synchronous)

  b. Synchronous counters:

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

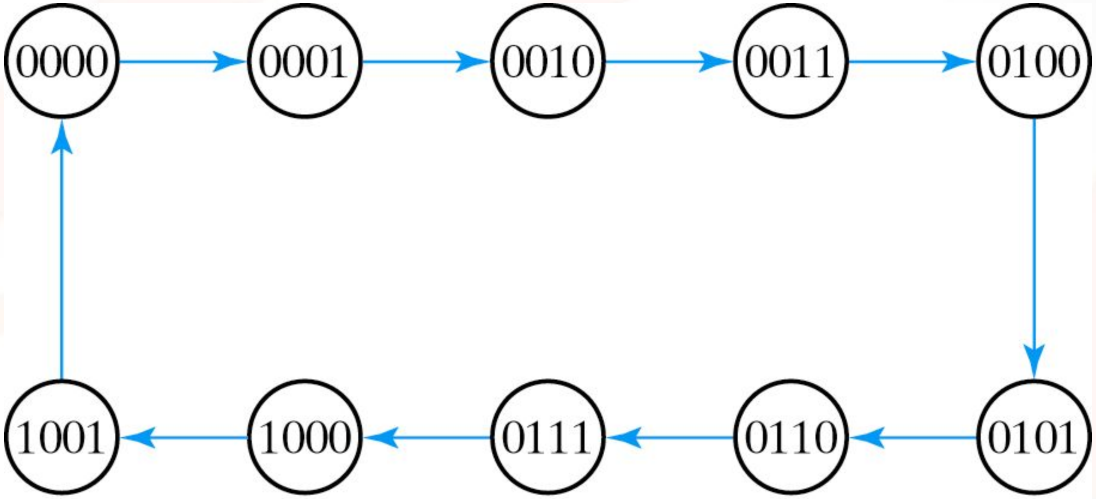
4-bit ripple counter



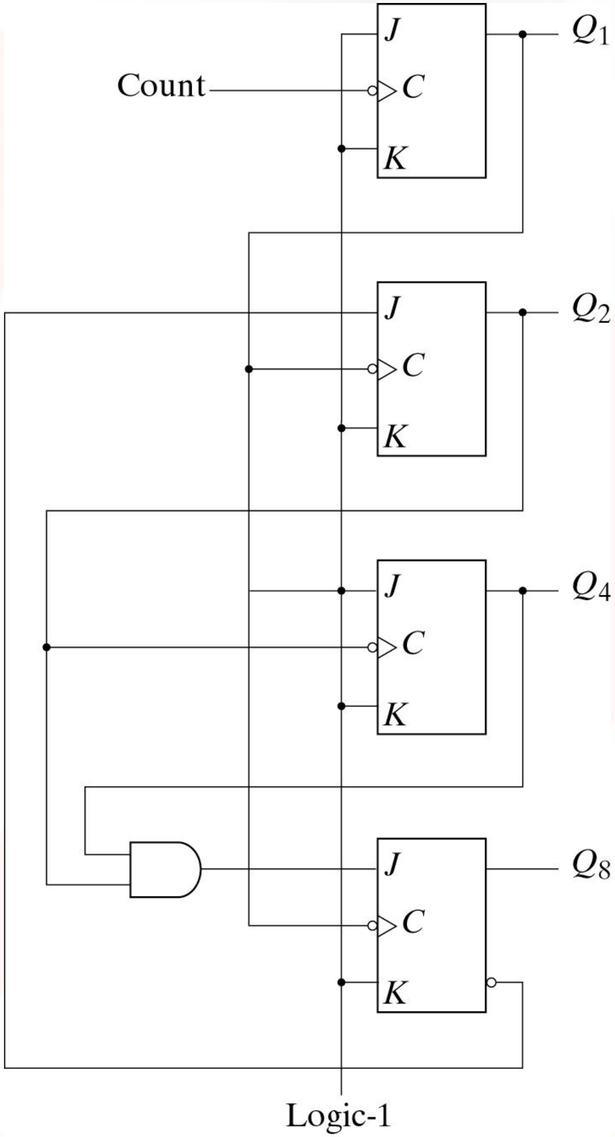
(a) With T flip-flops

(b) With D flip-flops

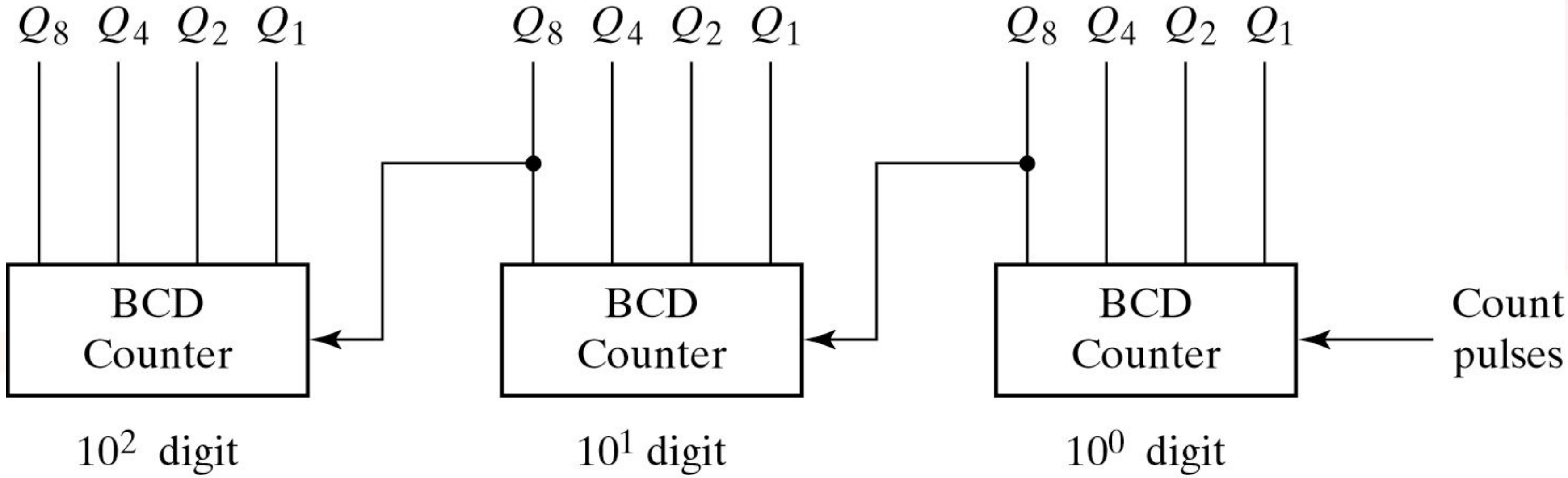
- BCD ripple counter



BCD Counter



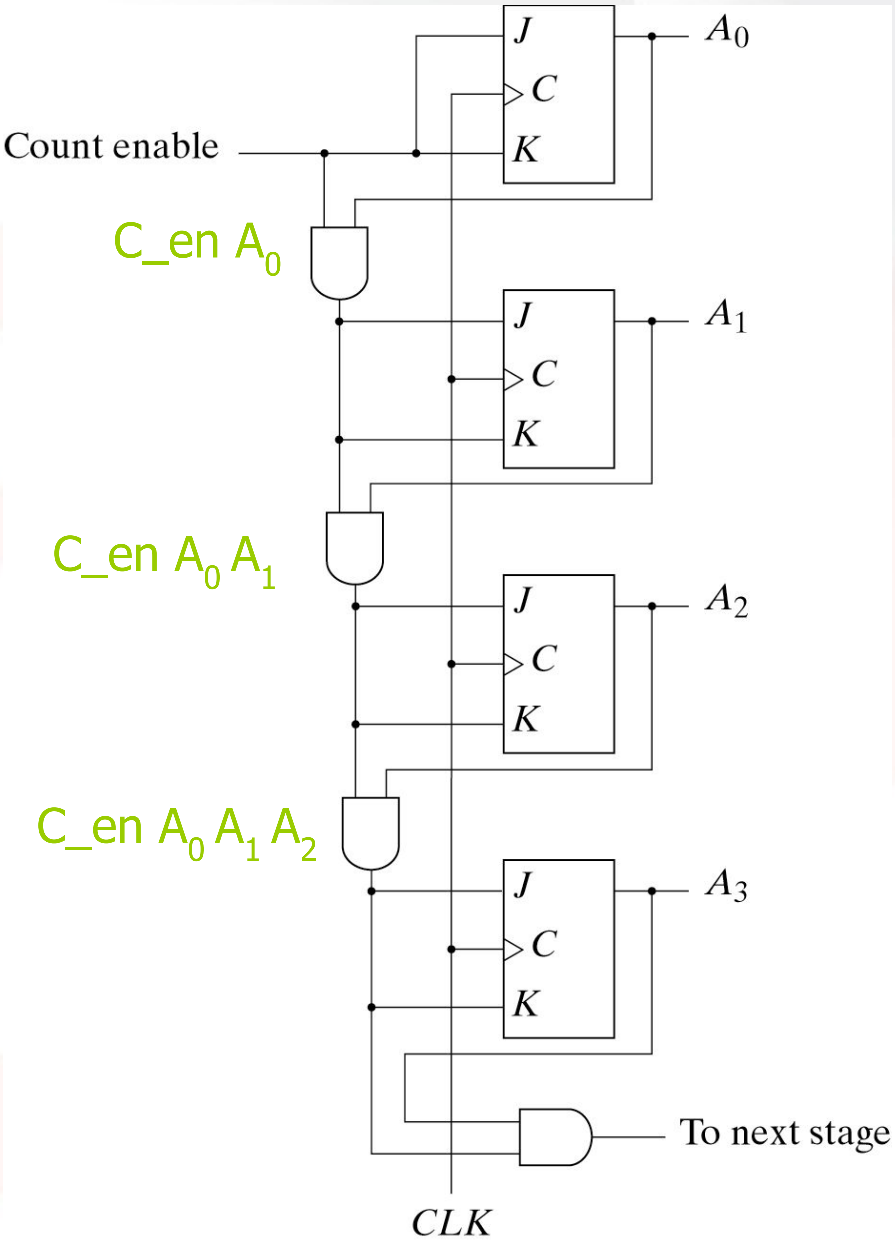
- Three-decade BCD counter



Synchronous Counters

- Synchronous counter
  - A common clock triggers all flip-flops simultaneously
- Design procedure
  - apply the same procedure of sync seq ckts
  - Sync counter is simpler than general sync seq ckts

- 4-bit binary counter

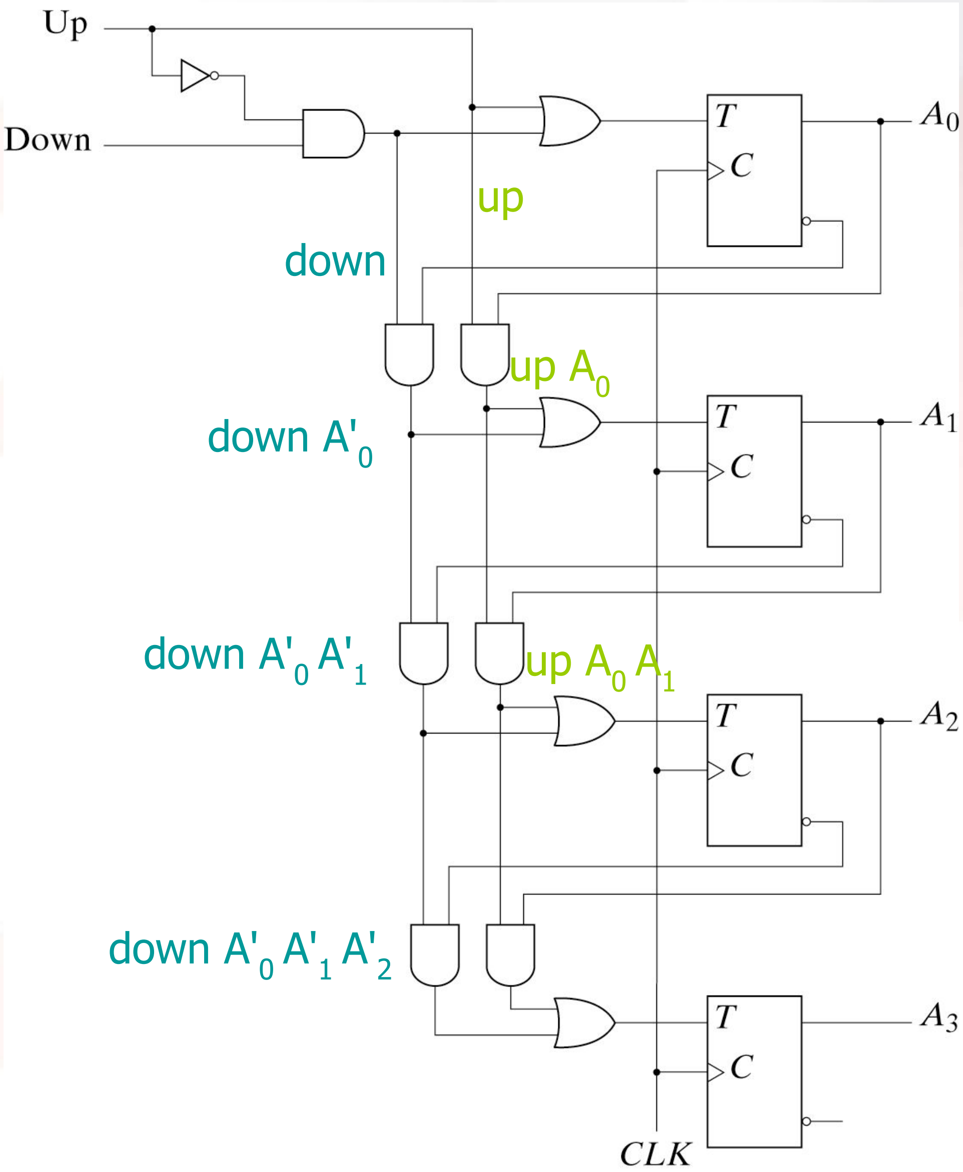


C_en A_0

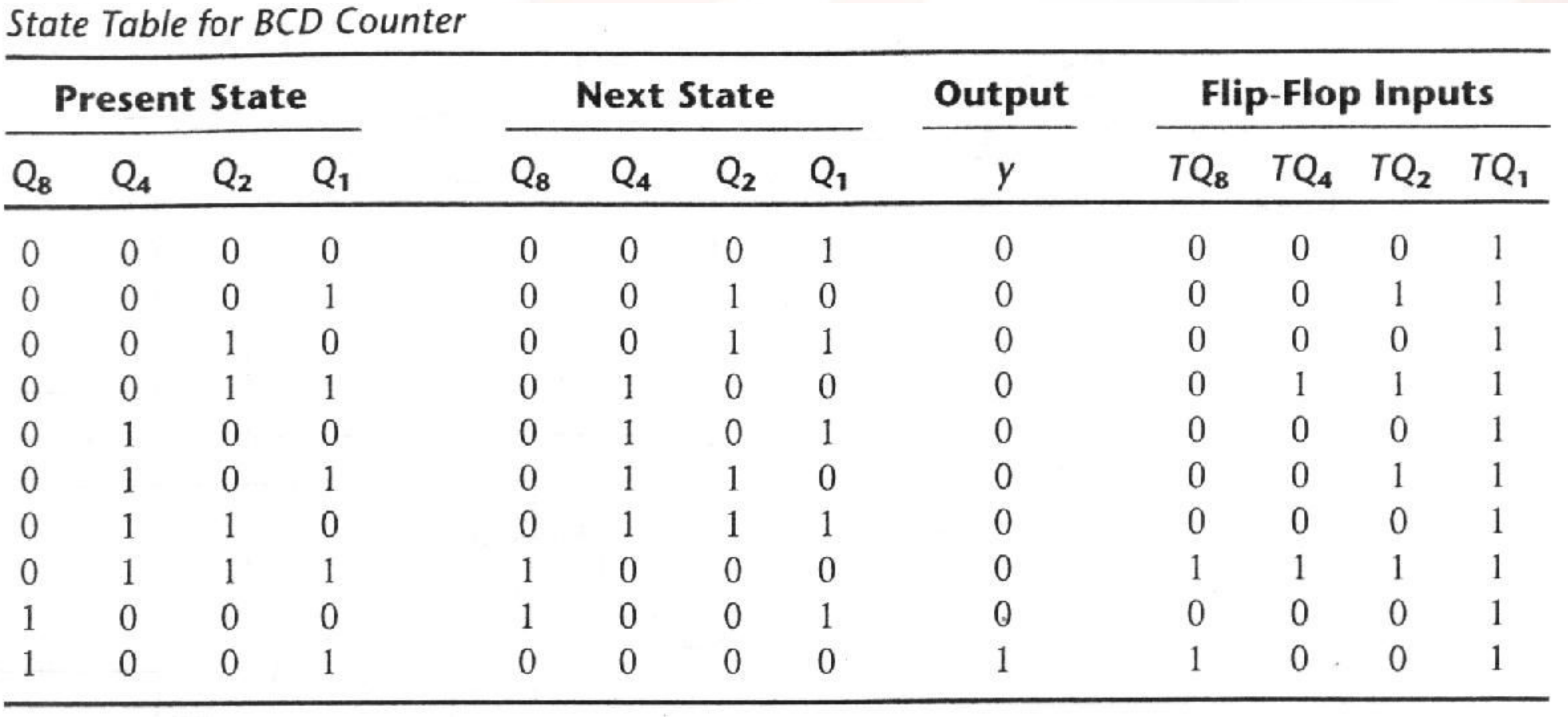
C_en A_0 A_1

C_en A_0 A_1 A_2

- 4-bit up/down binary counter



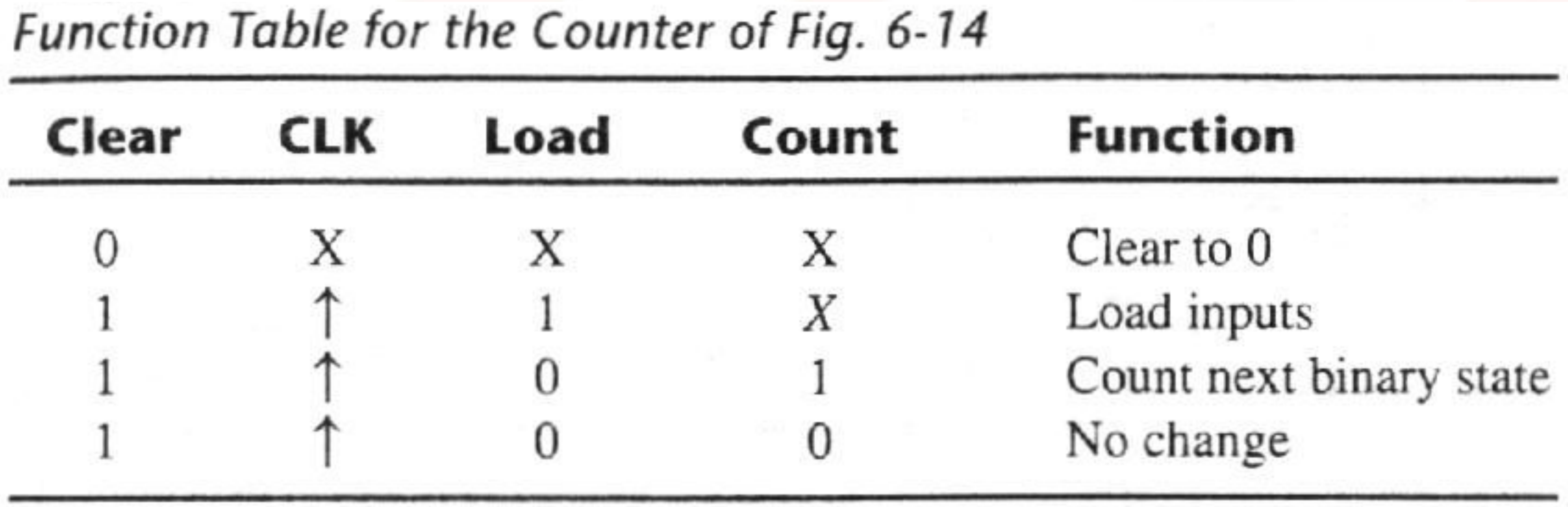
BCD counters



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

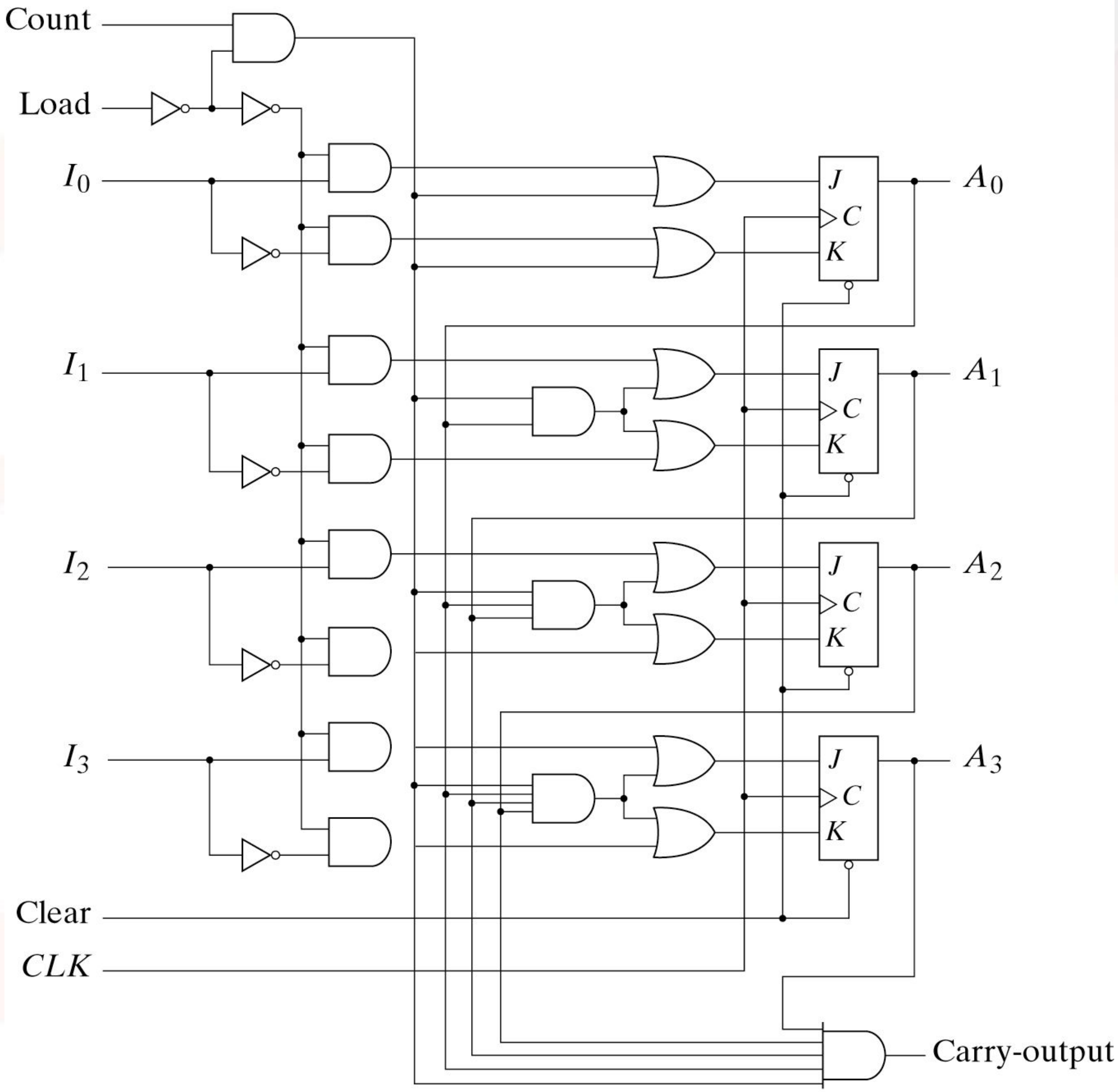
- 4-bit binary counter w/ parallel load



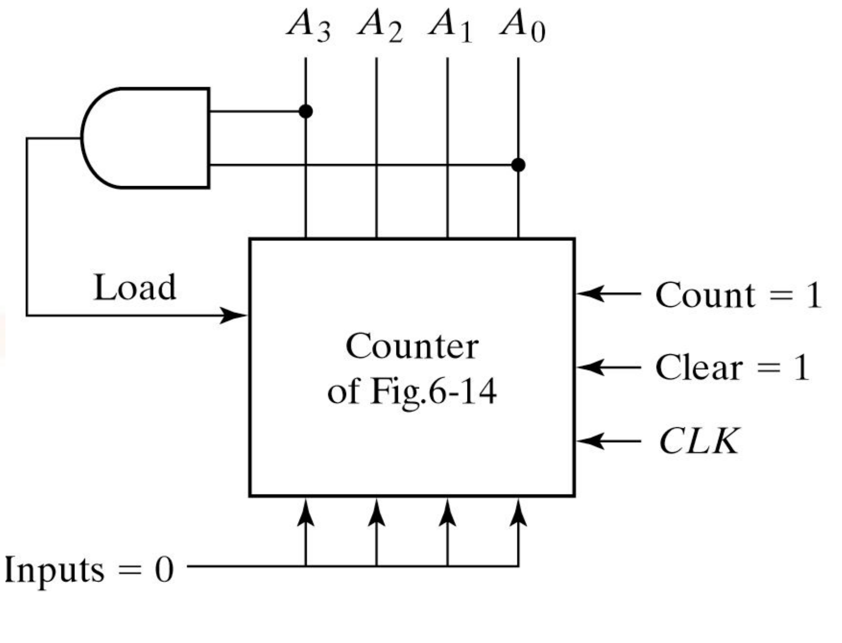
Function Table for the Counter of Fig. 6-14

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

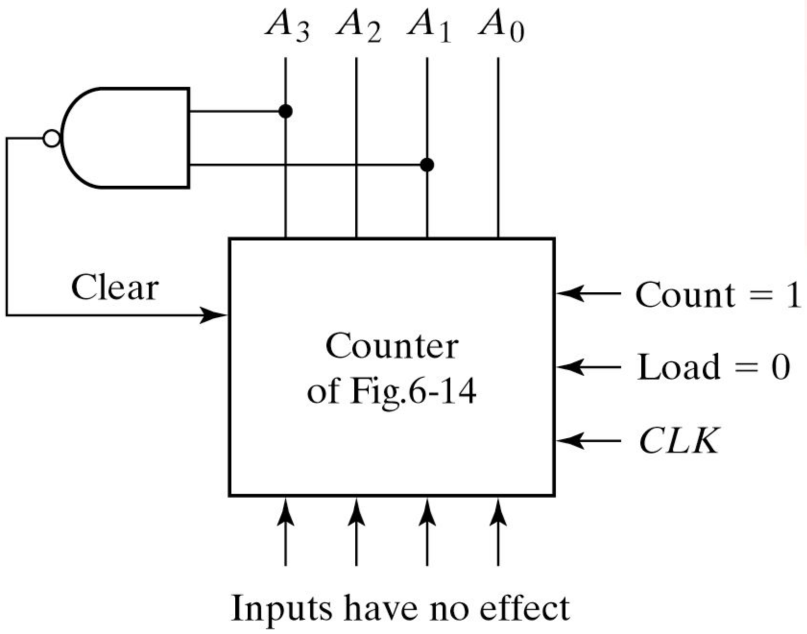
4-bit binary counter with parallel load



- Generate any count sequence:
  - e.g.: BCD counter ⇐ Counter w/ parallel load



(a) Using the load input

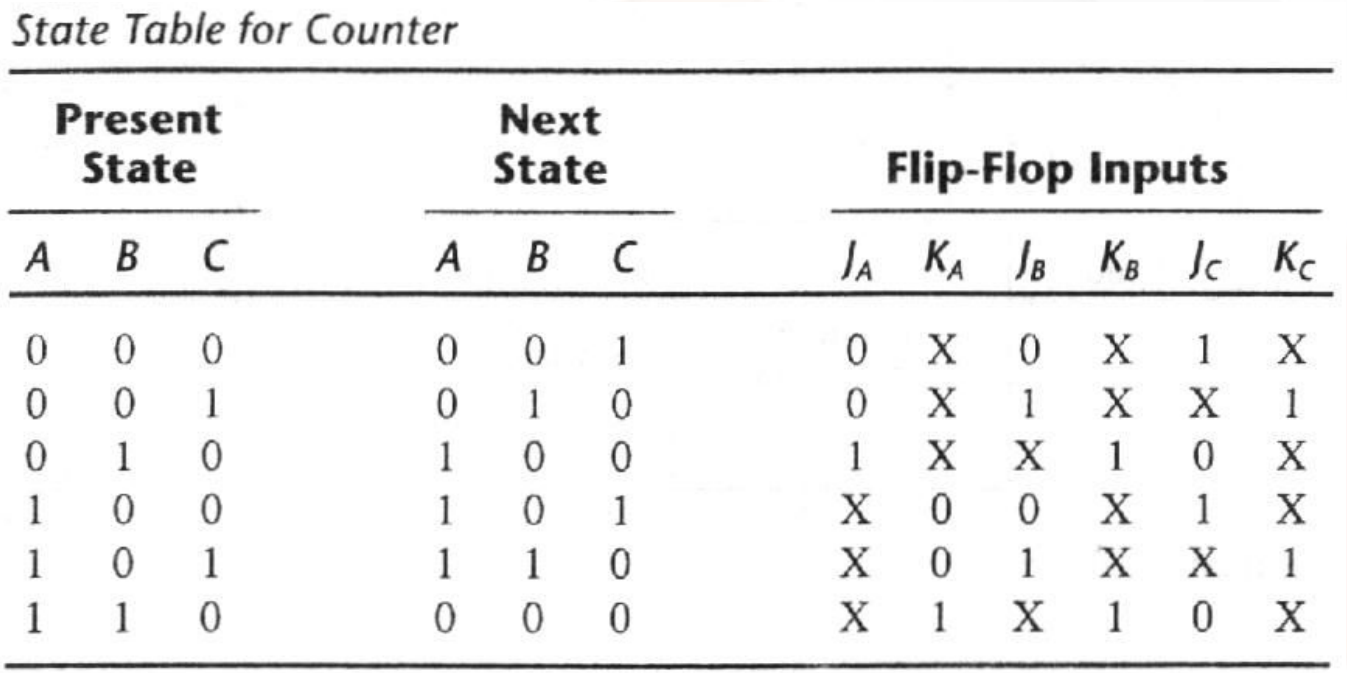


(b) Using the clear input

Other Counters

- 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

- *n* flip-flops ⇒ 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 ckt 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.
    ⇒ Analyze the ckt to determine the next state from an
    unused state after it is designed



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

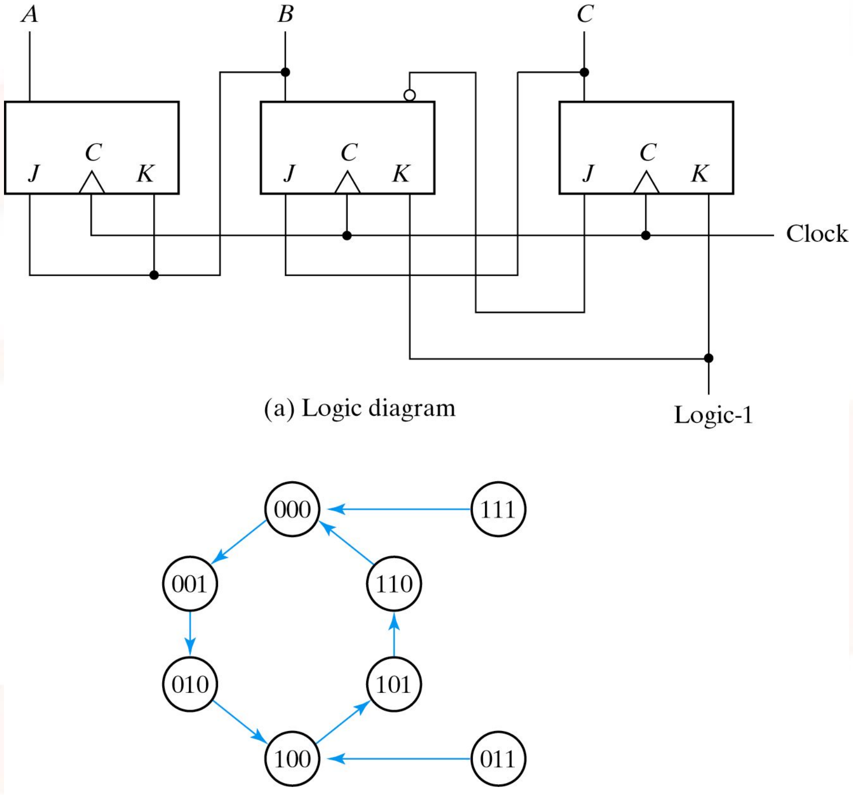
The simplified flip-flop input eqs:

$J_A = B$, $K_A = B$

$J_B = C$, $K_B = 1$

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

- The logic diagram & state diagram of the ckt



(a) Logic diagram

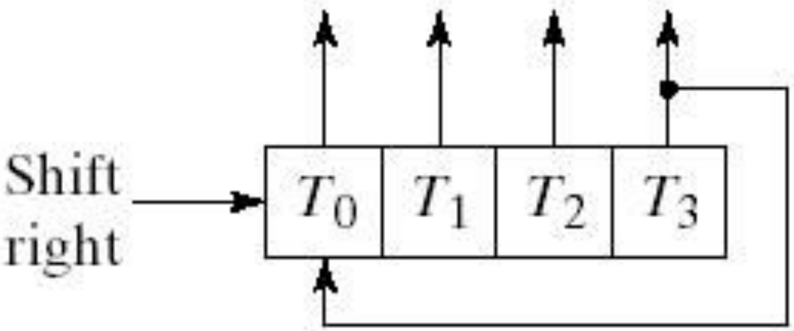
(b) State diagram

- Ring counter:
  - 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.

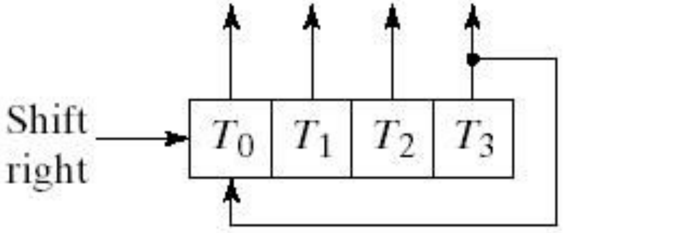
- A 4-bit ring counter

A_3	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

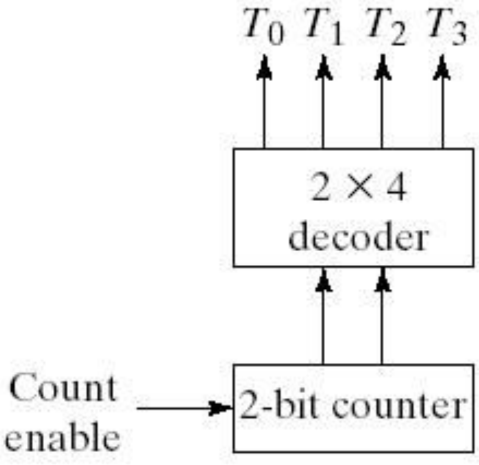


(a) Ring-counter (initial value = 1000)

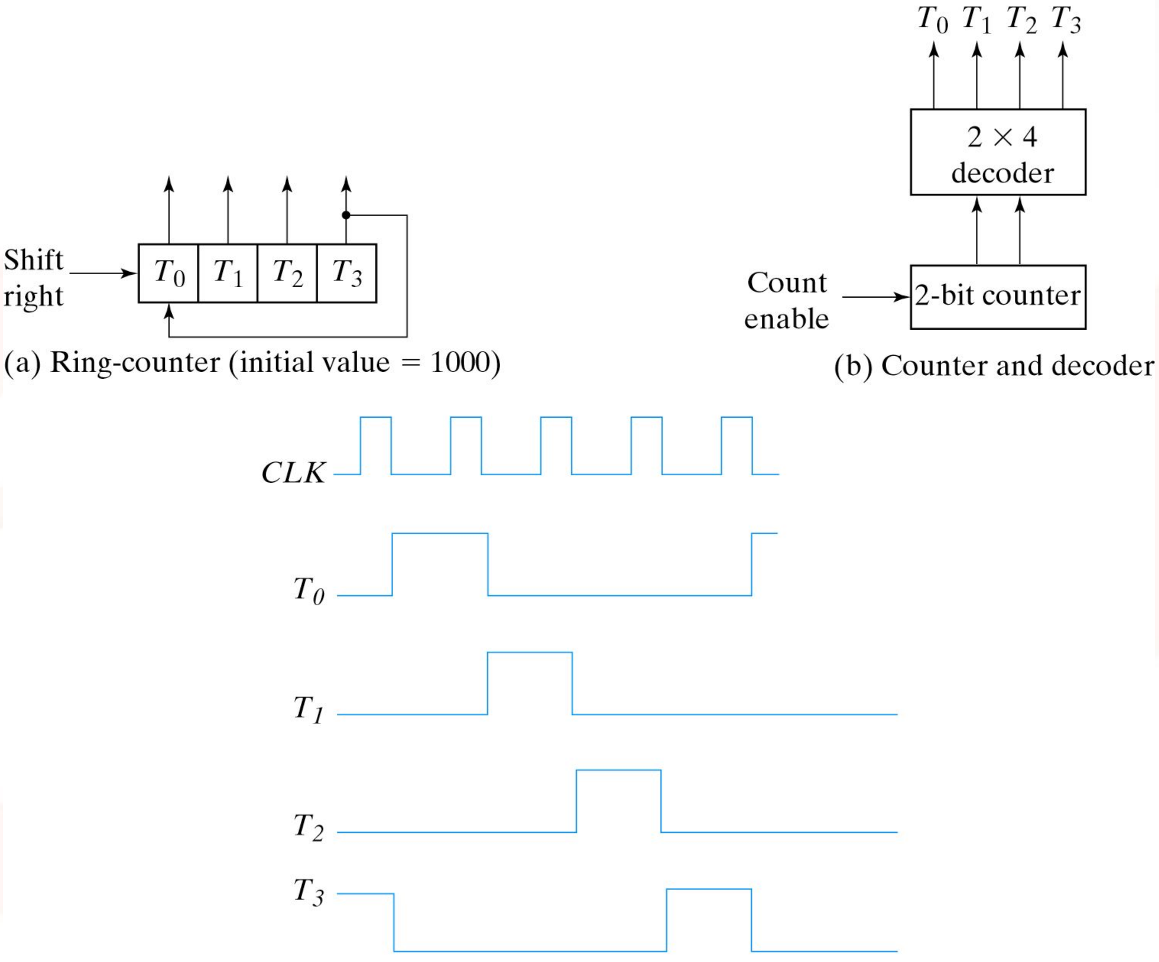
- Application of counters
  - Counters may be used to generate timing signals to control the sequence of operations in a digital system.
- Approaches for generation of 2^n timing signals
  1. a shift register w/ 2^n flip-flops
  2. an *n*-bit binary counter together w/ an *n*-to-2^n-line decoder



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



(b) Counter and decoder



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

(b) Counter and decoder

(c) Sequence of four timing signals