

CSE 205: DIGITAL LOGIC DESIGN

Dr. Tanzima Hashem
Associate Professor
CSE, BUET

REGISTERS AND COUNTERS

- 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.
 - 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 is shown in Fig. 6.1.

Clear =0 (active low); $A_x = 0$

Clock= \uparrow ; $A_x = I_x$

Normal Operation; Clear =1

- If some bits must be left unchanged, how to do it ?

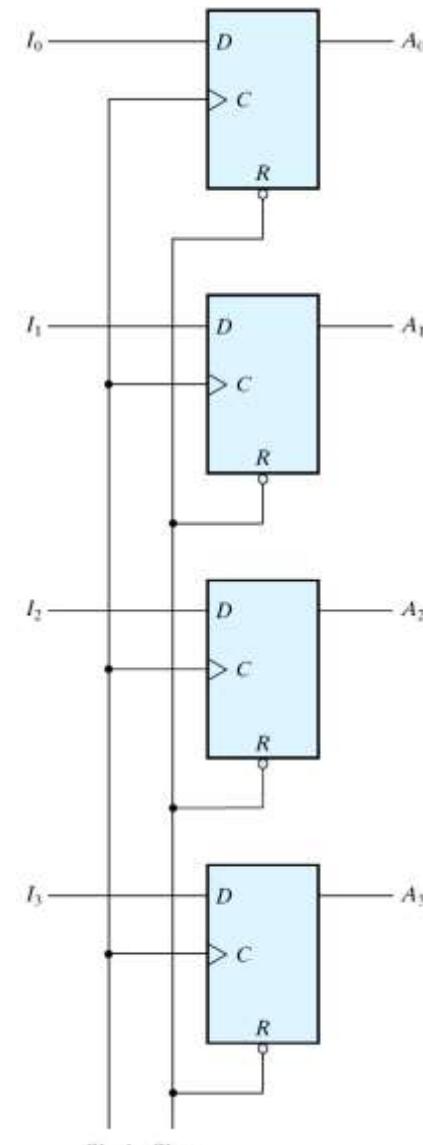
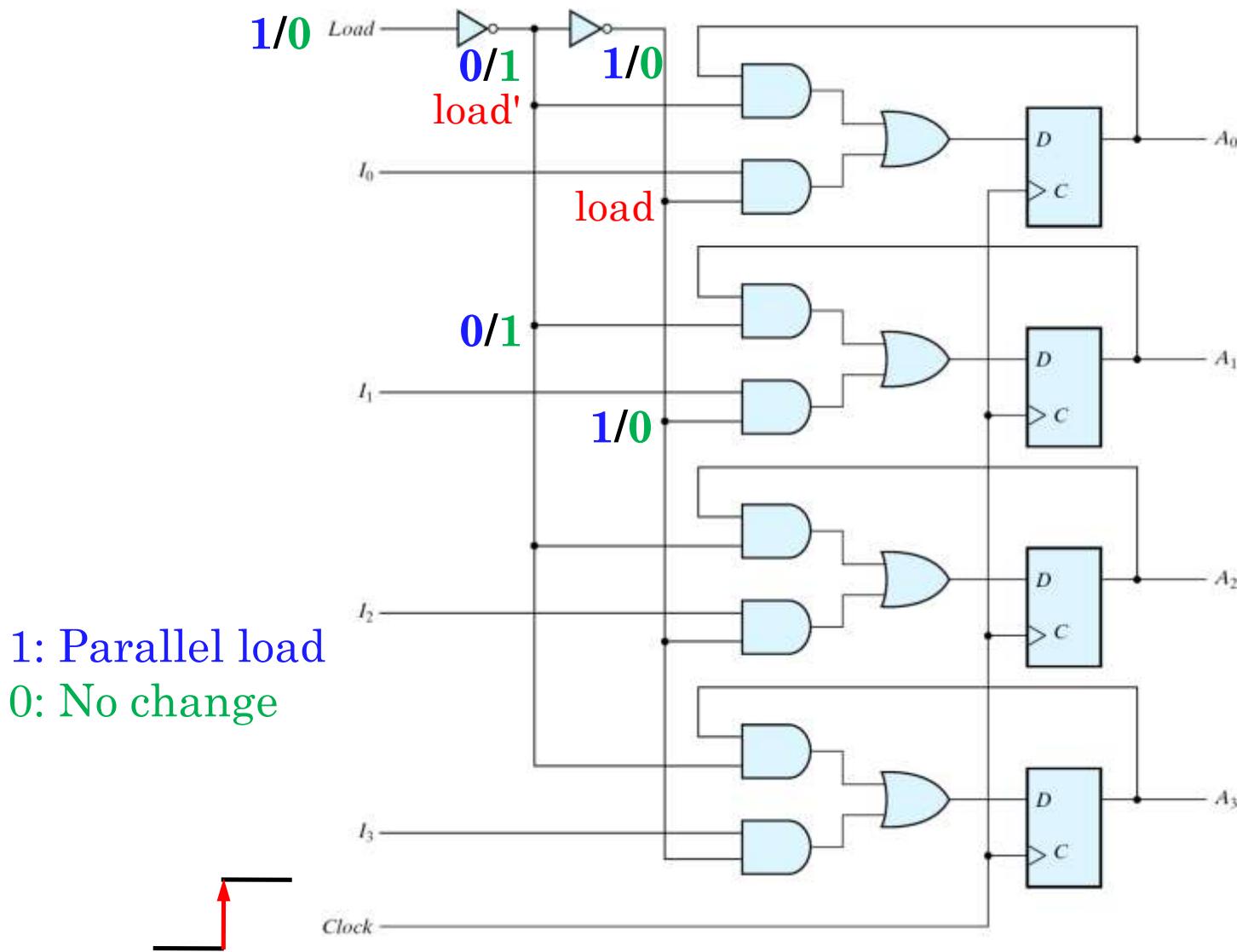


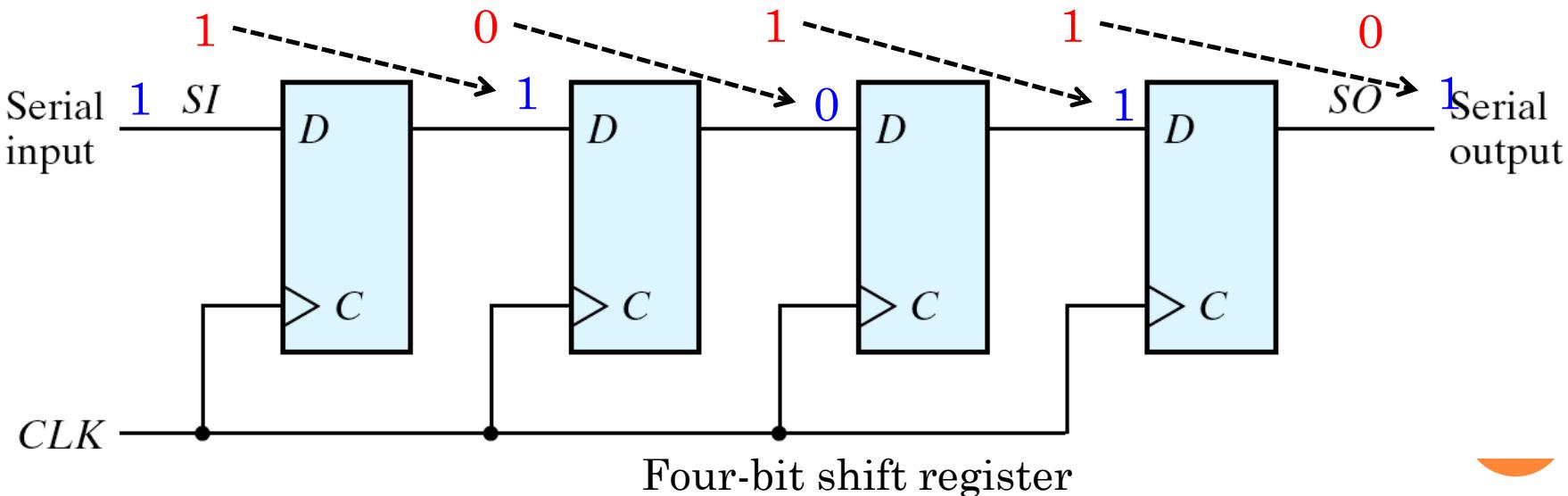
Fig. 6.1 Four-bit register

4-BIT REGISTER WITH PARALLEL LOAD



SHIFT REGISTERS

- A register capable of **shifting the binary information** held in each cell to its neighbouring cell, in a selected direction is called a **shift register**.
 - Clock controls the shift operation



DATA TRANSFER

- 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

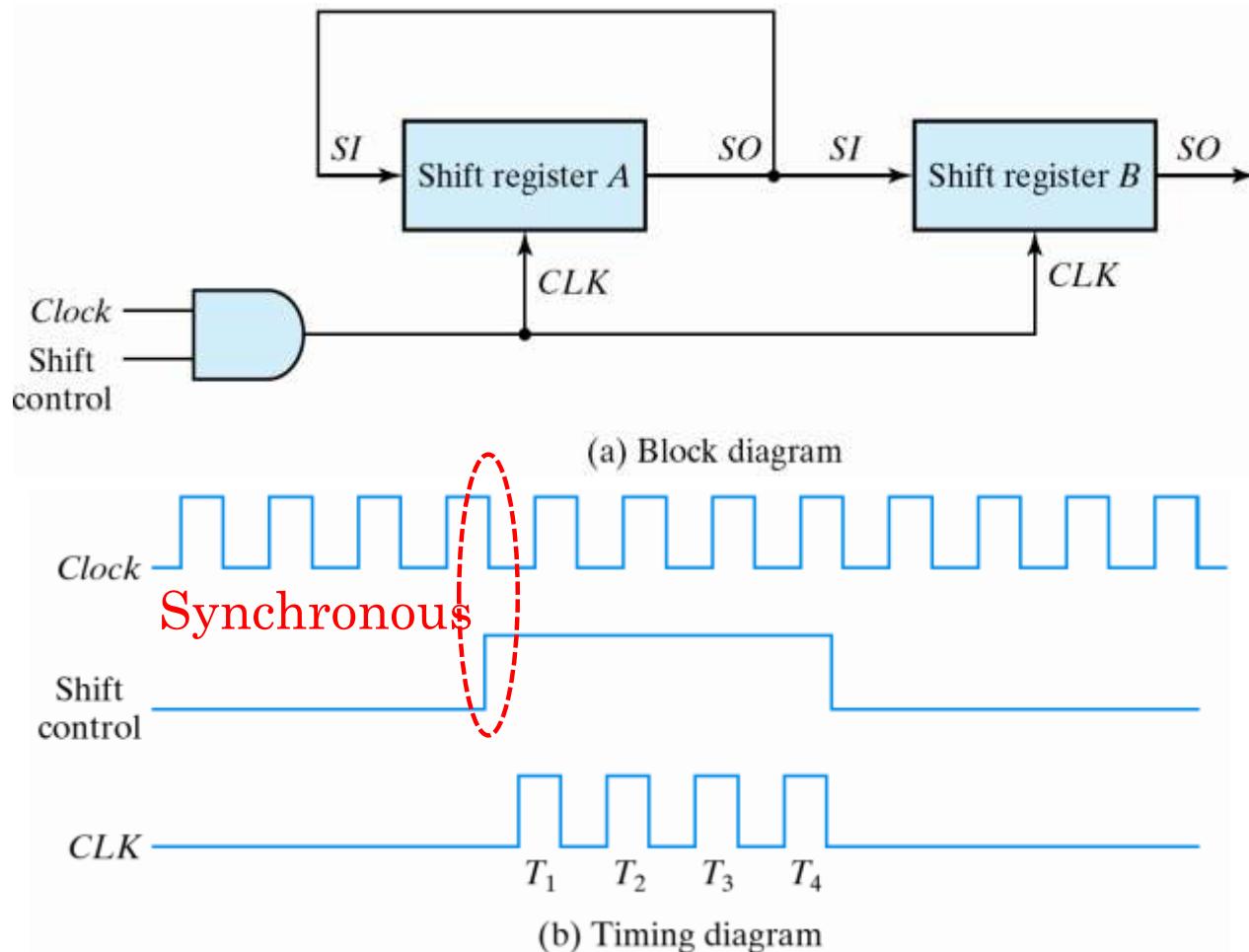
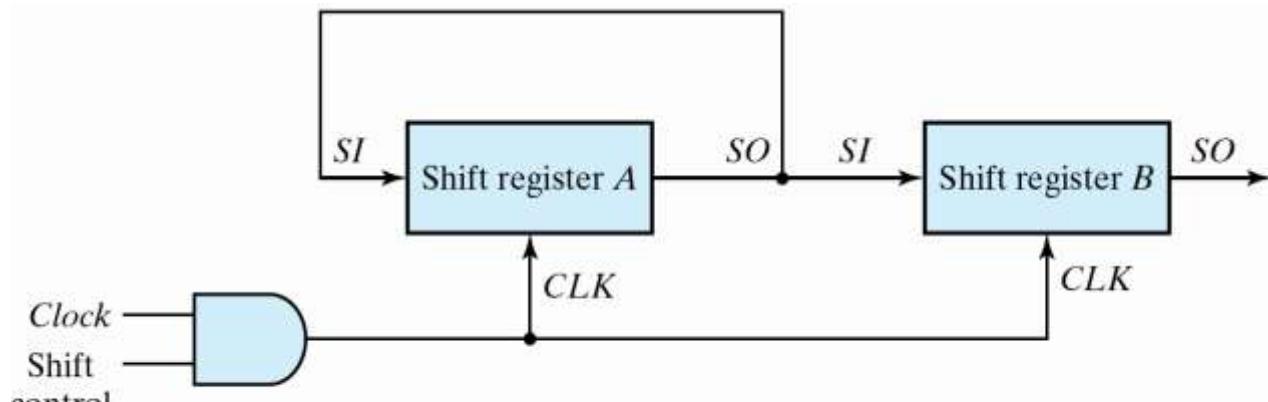


Fig. 6.4 Serial transfer from register A to register B

SERIAL TRANSFER FROM REG A TO REG 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-
transfer

SERIAL ADDITION

- Parallel adders
 - Faster,
 - cost more logic
- Serial adders
 - Slower
 - n -bit addition \longrightarrow n clock cycles
 - Less hardware



SERIAL ADDITION USING D FLIP-FLOPS

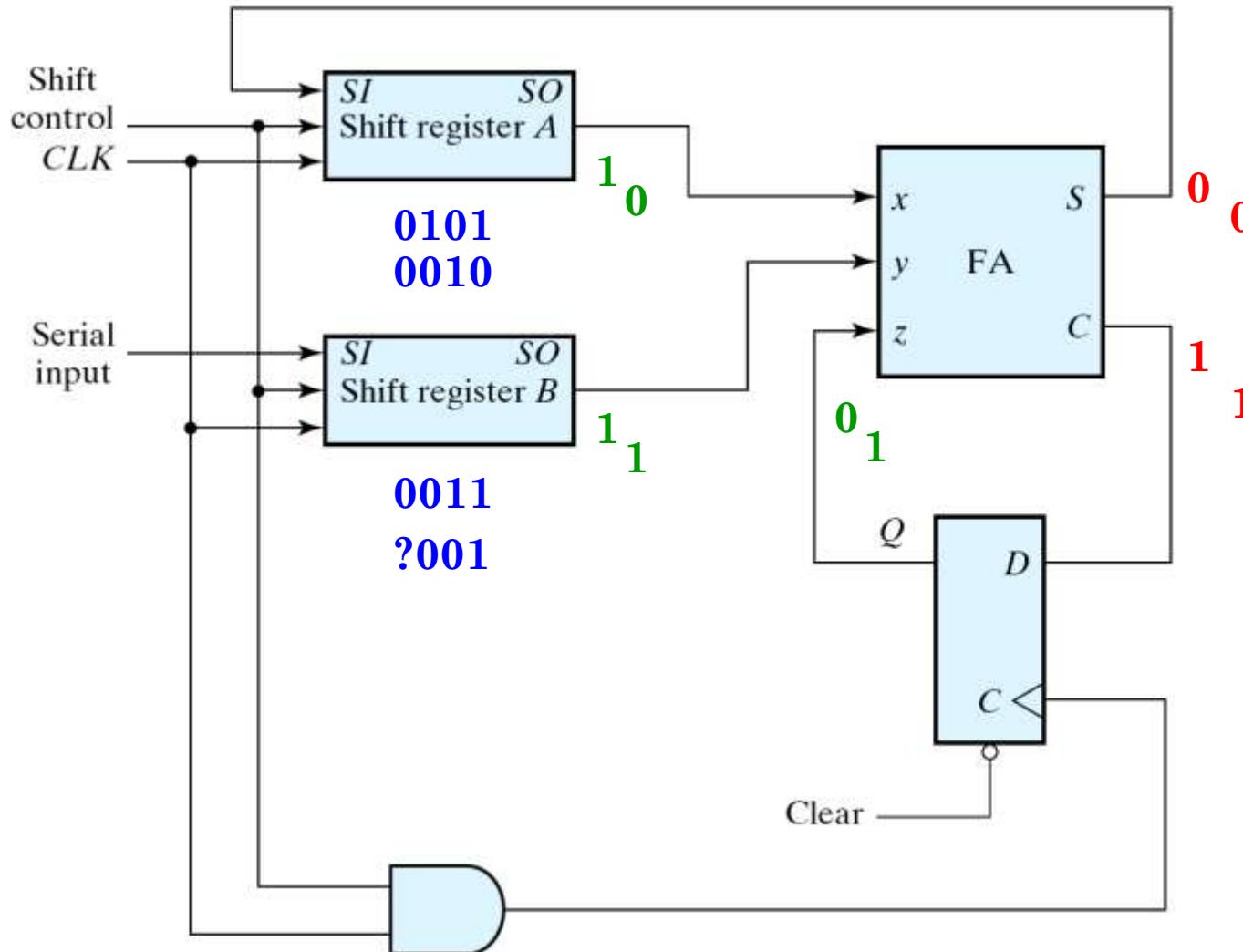


Fig. 6.5 Serial adder

SERIAL ADDER USING JK FFs (1/2)

- Serial adder using JK flip-flop

Table 5.12
Flip-Flop Excitation Tables

$Q(t)$	$Q(t = 1)$	J	K	$Q(t)$	$Q(t = 1)$	T
0	0	0	X	0	0	0
0	1	1	X	0	1	1
1	0	X	1	1	0	1
1	1	X	0	1	1	0

(a) JK

(b) T

Table 6.2
State Table for Serial Adder

Present State	Inputs		Next State	Output	Flip-Flop Inputs	
	Q	x			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

SERIAL ADDER USING JK FFs (2/2)

- Circuit diagram

- $J_Q = xy \quad K_Q = x'y' = (x+y)'$
- $S = x \oplus y \oplus Q$

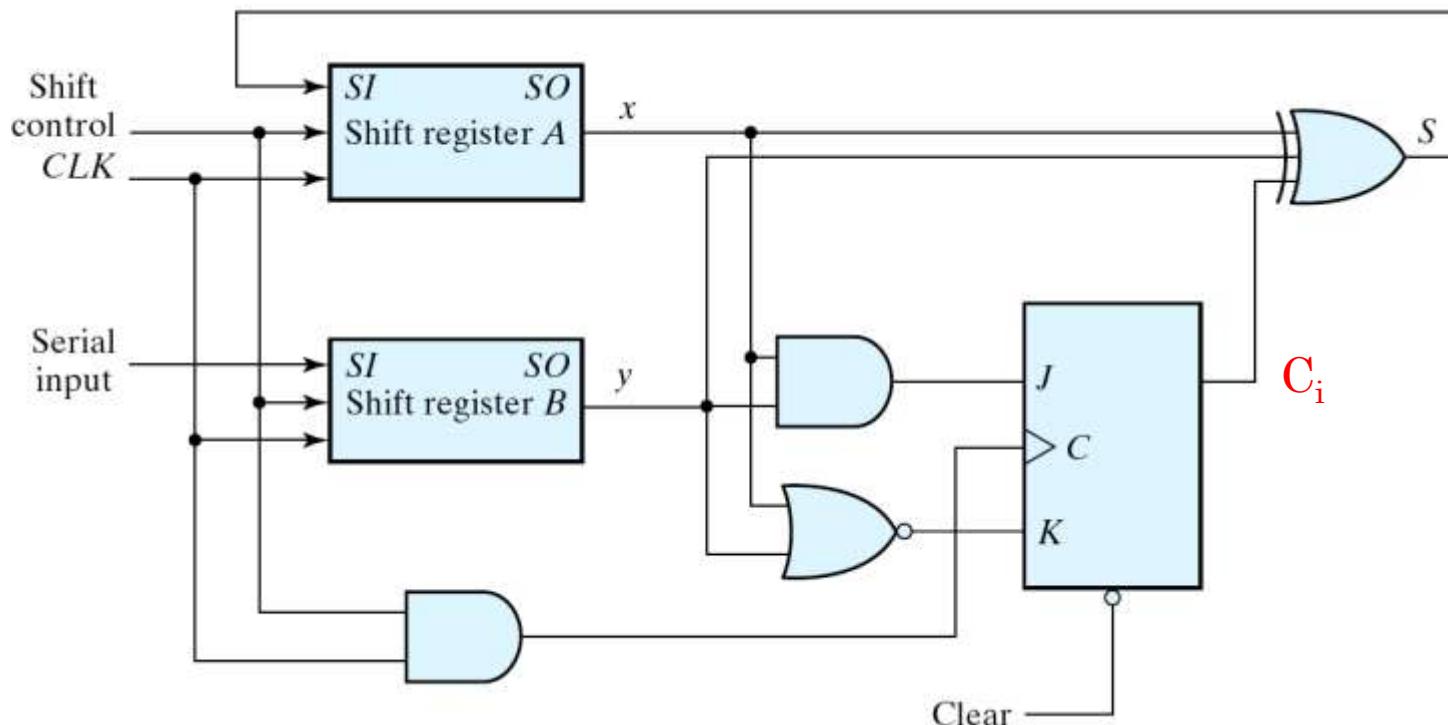


Fig. 6.6 Second form of serial adder

UNIVERSAL SHIFT REGISTER

- 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
 - Has both direction shifts & parallel load/out capabilities.



UNIVERSAL SHIFT REGISTER (1/4)

- 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 (2/4)

- Example: 4-bit universal shift register

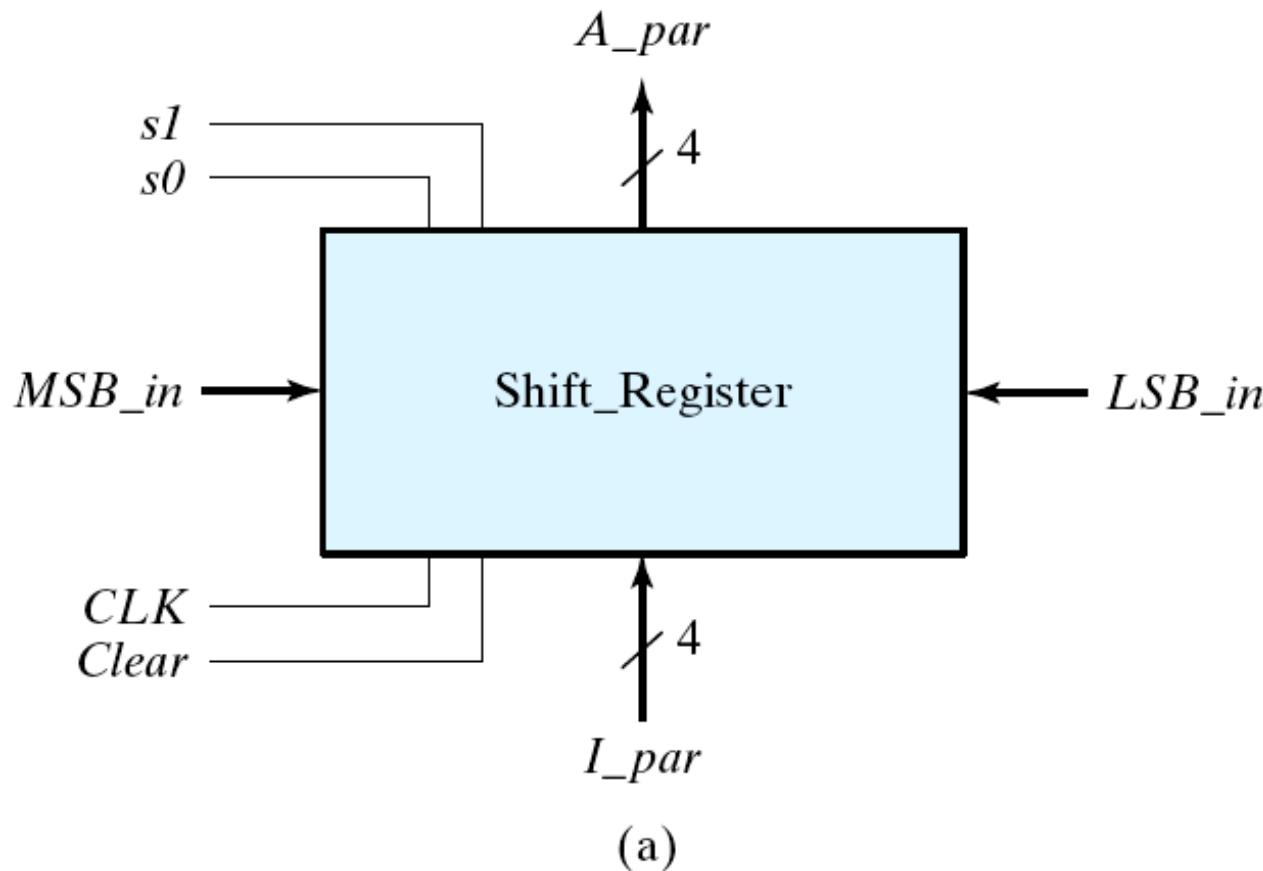


Fig. 6.7 Four-bit universal shift register

UNIVERSAL SHIFT REGISTER (3/4)

○ 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

UNIVERSAL SHIFT REGISTER (4/4)

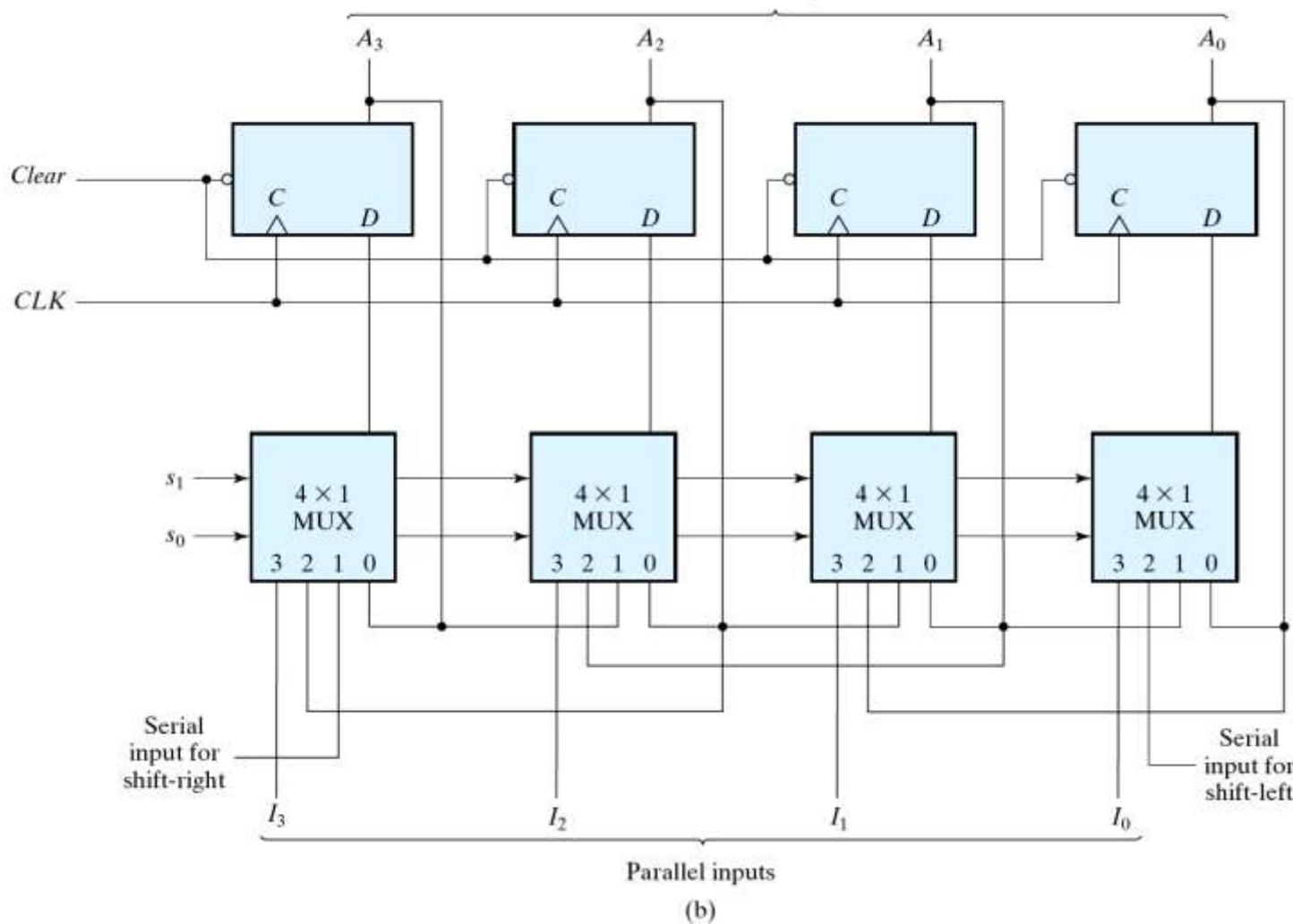


Fig. 6.7 Four-bit universal shift register

COUNTERS

- Counter : A register that goes through a prescribed sequence of states.
 - Special purpose arithmetic circuits used for the purpose of counting
 - The sequence of states: may follow the binary number sequence (\Rightarrow **Binary counter**) or any other sequence of states (e.g., Gray Code).
- Counter circuits serve many purposes
 - Count occurrences of certain events
 - Generate timing intervals for controlling various tasks in a digital system
 - Track elapsed time between events



COUNTERS

- Categories of counters
 1. Asynchronous counters (Ripple counters) The flip-flops within the counter do not change state at the same time
 - The flip-flop output transition serves as a source for triggering other flip-flop.
 - ◆ \Rightarrow no common clock pulse.
 2. Synchronous counters
 - ◆ The CLK inputs of all flip-flops receive a common clock.



BINARY RIPPLE COUNTER

- A n -bit binary counter $\rightarrow n$ FFs \rightarrow count from 0 to $2^n - 1$.
- Example: 4-bit binary ripple counter
 - Binary count sequence: 4-bit

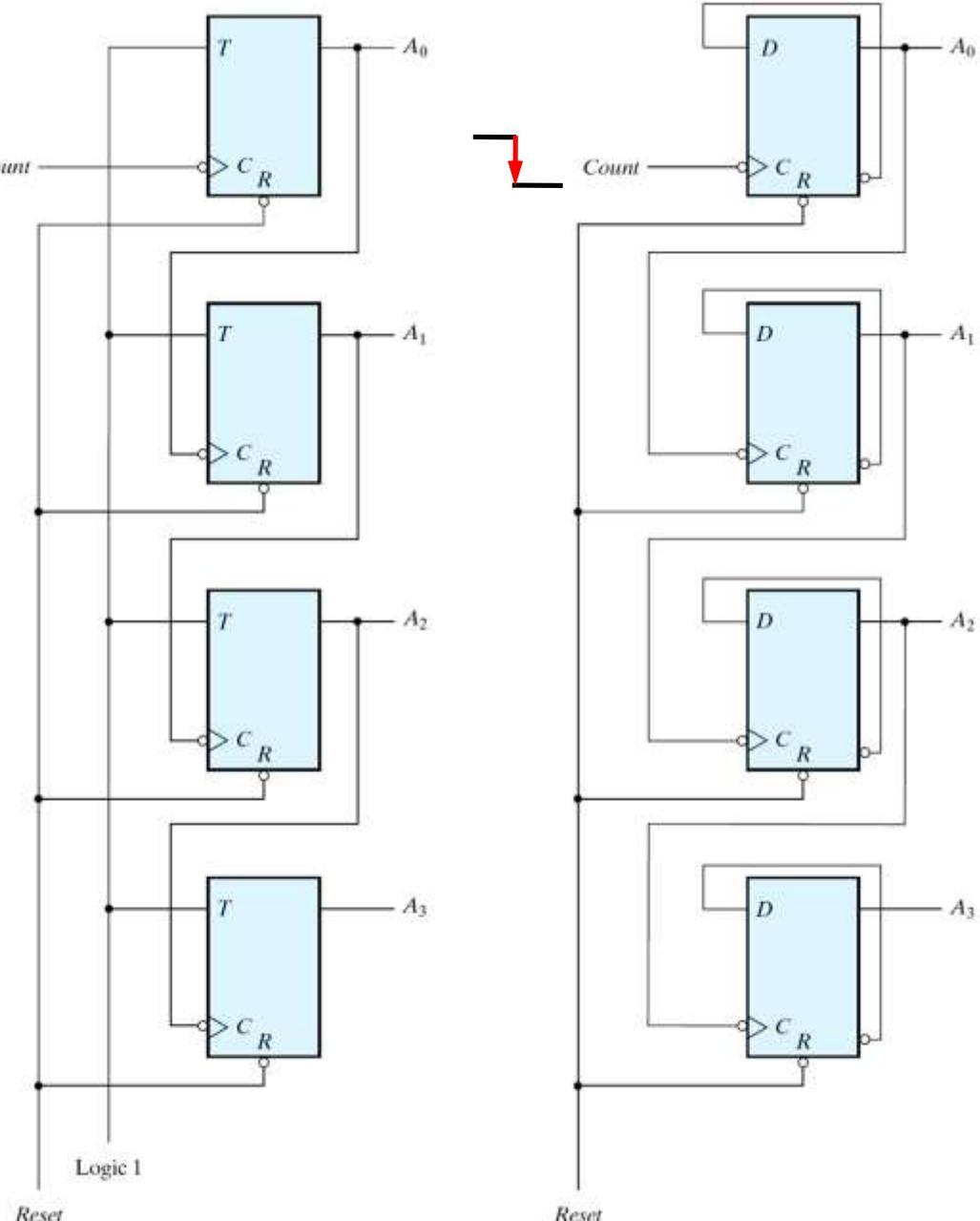
Table 6.4
Binary Count Sequence

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



BINARY RIPPLE COUNTER

- Reset signal sets all outputs to 0
- Negative edge triggered
- Count signal toggles output of low-order flip flop
- Low-order flip flop provides trigger for adjacent flip flop
- Output of one flipflop → Clock to the next
- Not all flops change value simultaneously
 - Lower-order flops change first



(a) With T flip-flops

(b) With D flip-flops

BINARY 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

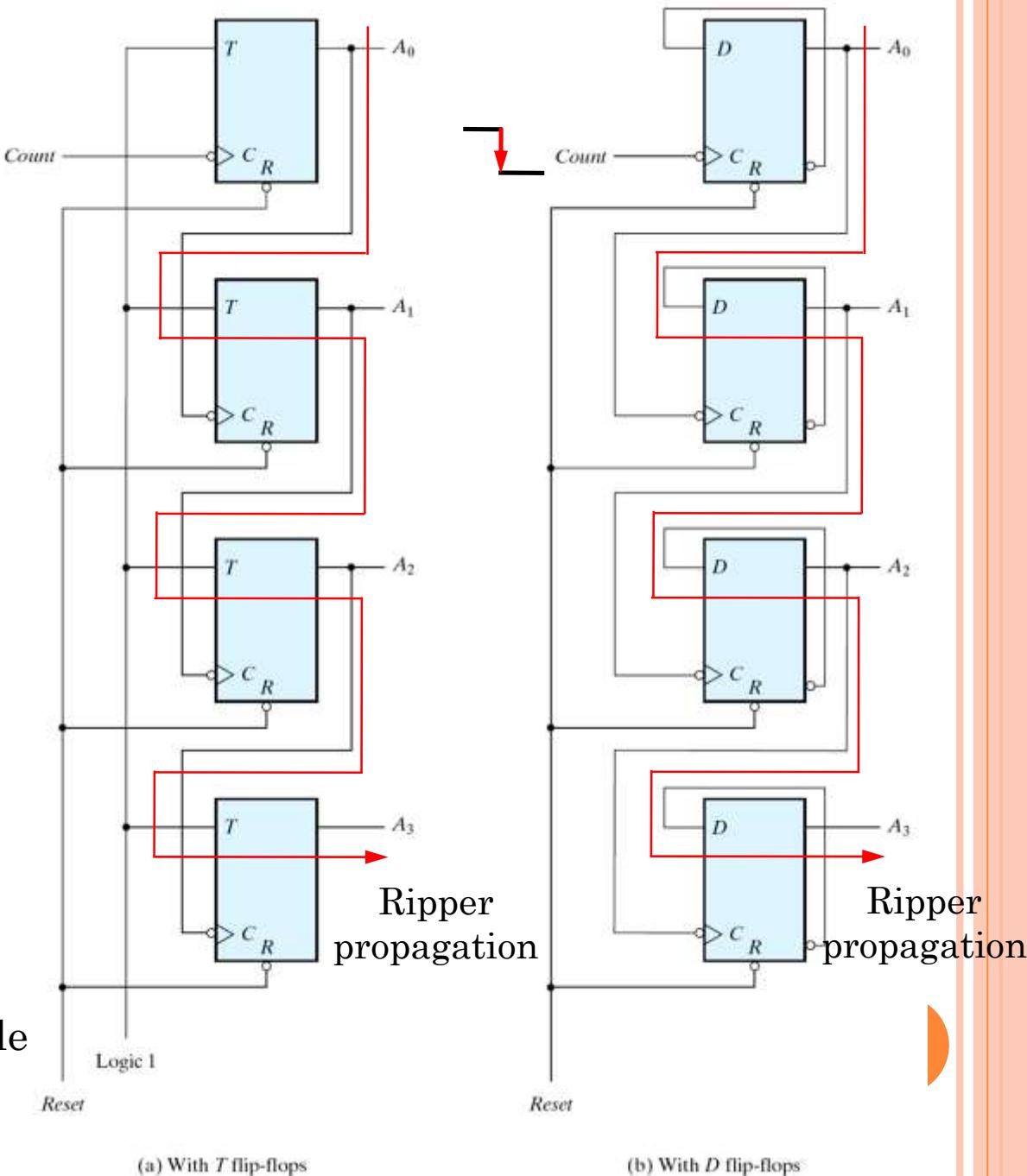


Fig. 6.8 Four-bit binary ripple counter

ASYNCHRONOUS (RIPPLE) COUNTERS

- Asynchronous counters are commonly referred as ripple counters
 - The effect of the input clock pulse is first felt by the first flip-flop FF0.
 - The effect cannot get to the next flip-flop FF1 immediately as there is a propagation delay through FF0
 - Then there is a propagation delay through FF1 before the next flip-flop FF2 is triggered
 - Thus, the effect of an input clock pulse “ripples” through the counter, taking some time, due to propagation delays, to reach the last flip-flop.

BCD RIPPLE COUNTER

- Counter must reset itself after counting the terminal count

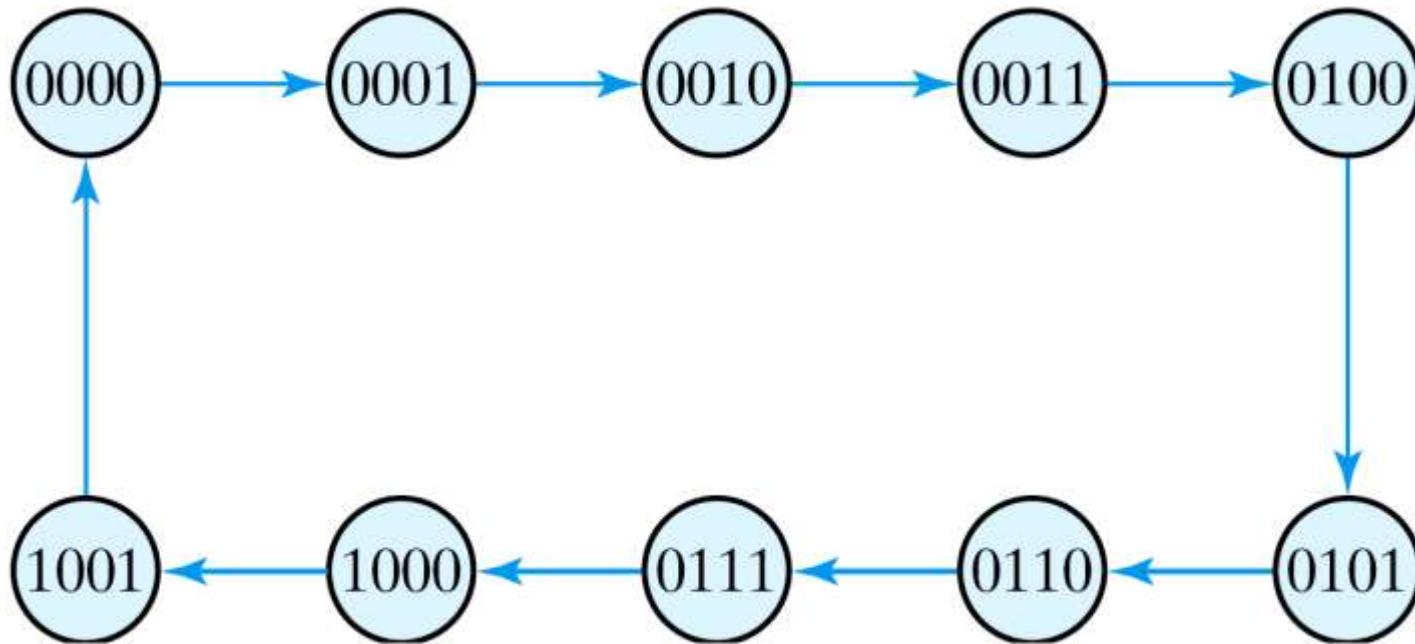
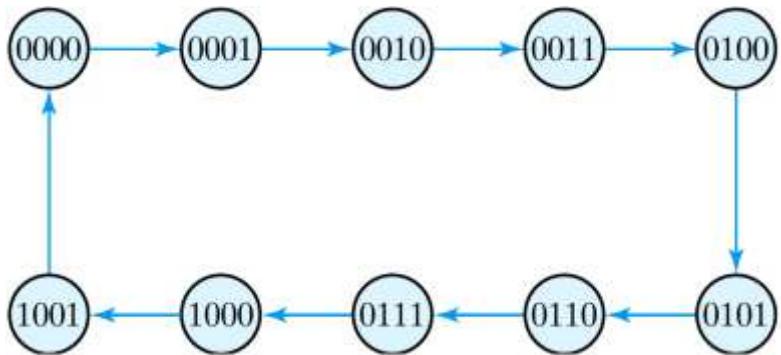


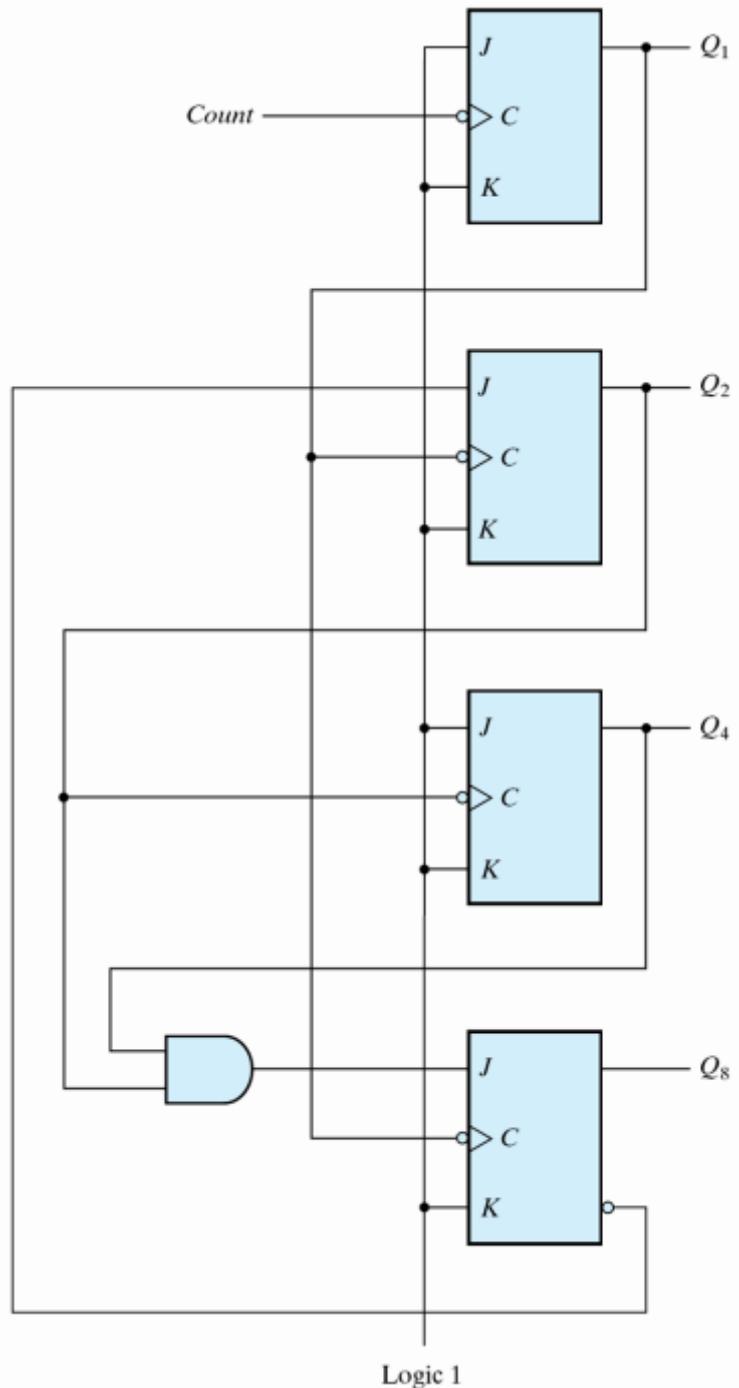
Fig. 6.9 State diagram of a decimal BCD counter

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



BCD RIPPLE COUNTER

- Three-decade BCD counter

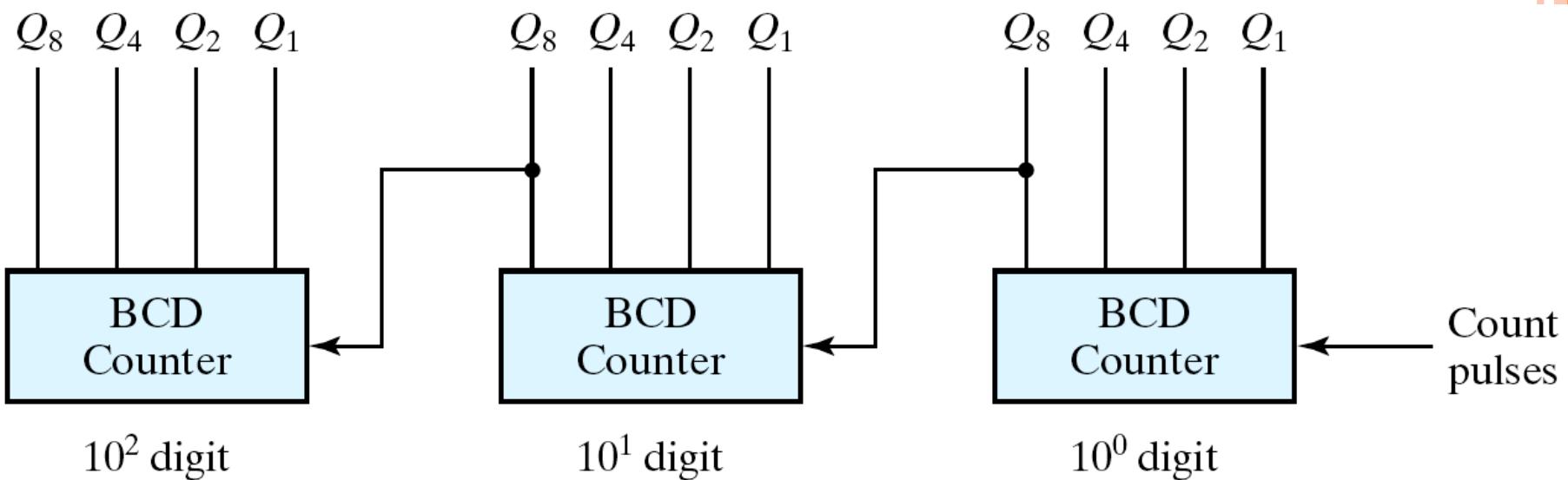


Fig. 6.11 Block diagram of a three-decade decimal BCD counter

SYNCHRONOUS COUNTERS

- All of the FFs are triggered simultaneously by the clock input pulses.
- All FFs change at same time
- Synchronous counters can be designed using **sequential circuit procedure**

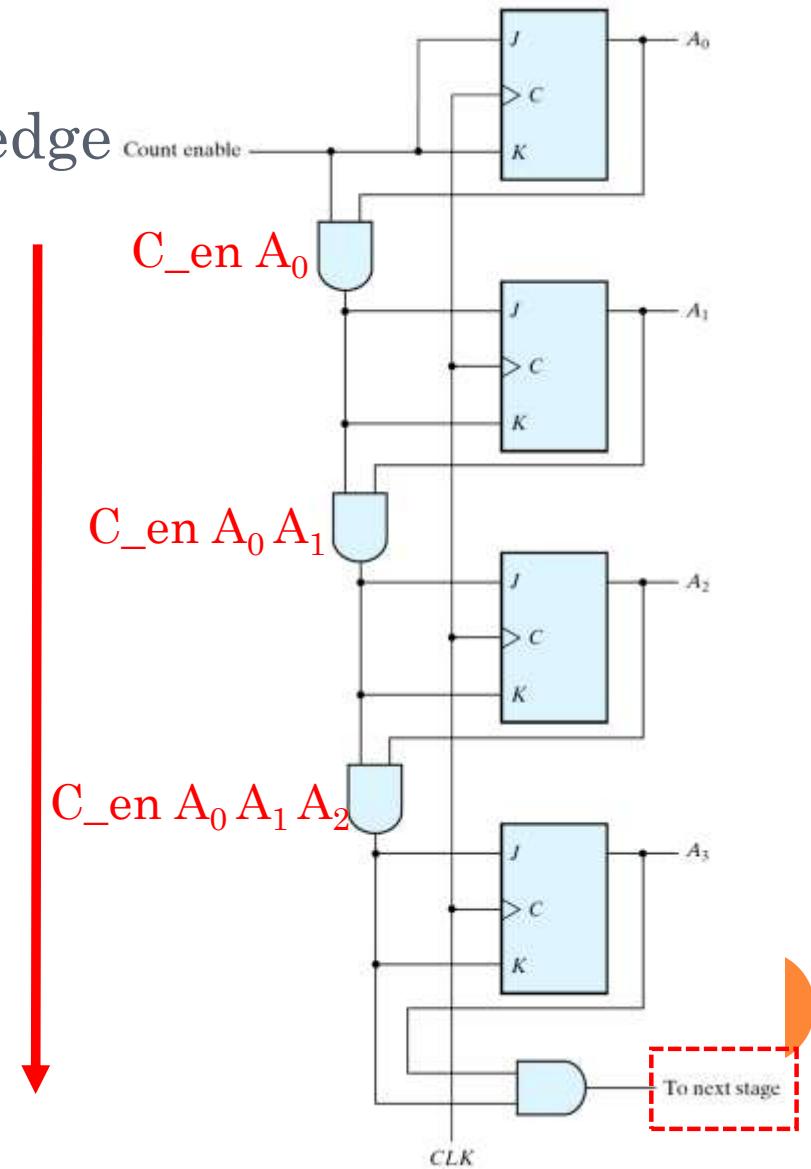


4-BIT BINARY COUNTER

Value increments on positive edge

- $J=K=0$,
→ no change;
- $J=K=1$,
→complement.

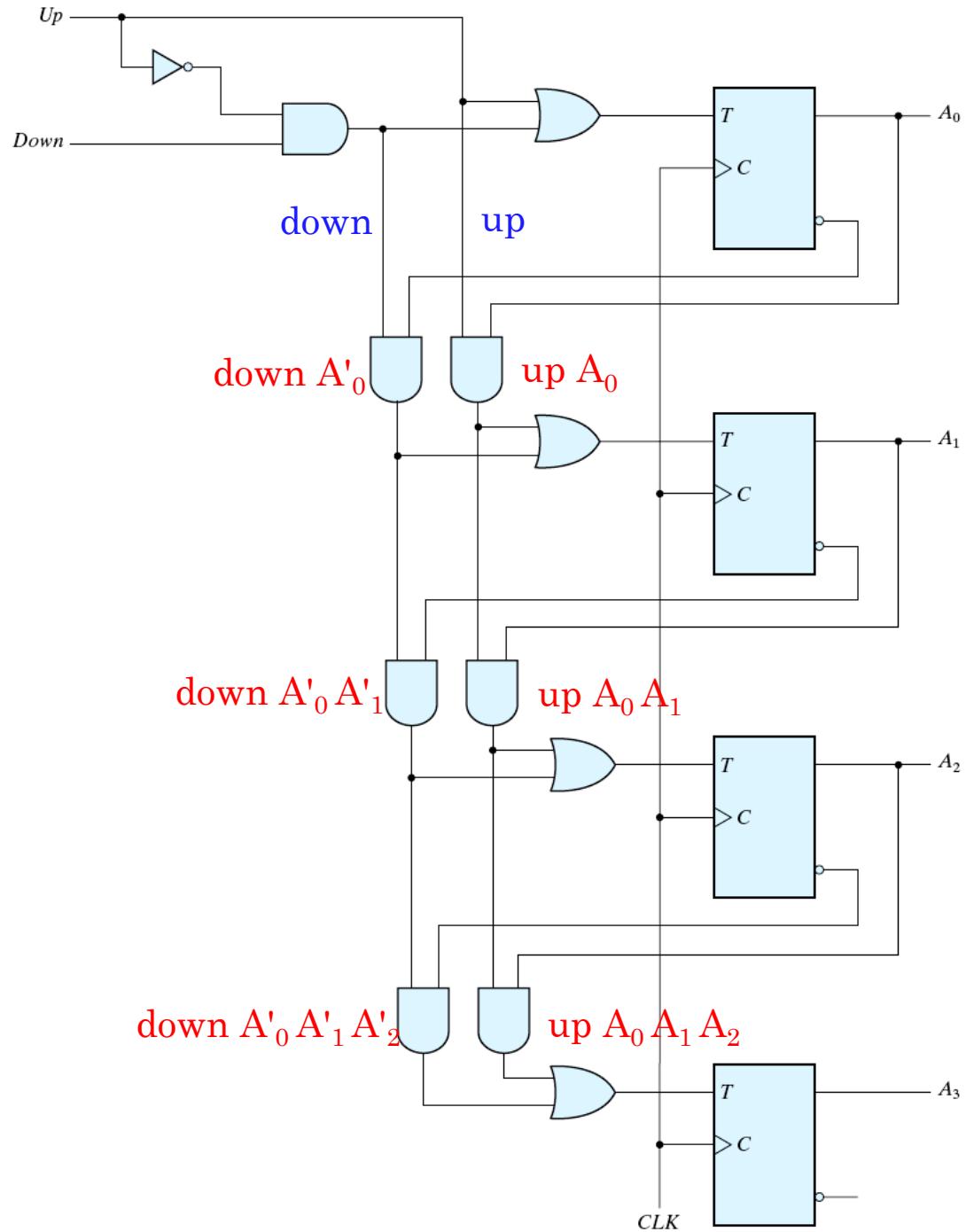
A_3	A_2	A_1	A_0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1



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



BCD COUNTERS

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	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

- Simplified functions

$$T_{Q1} = 1$$

$$T_{Q2} = Q'_8 Q_1$$

$$T_{Q4} = Q_2 Q_1$$

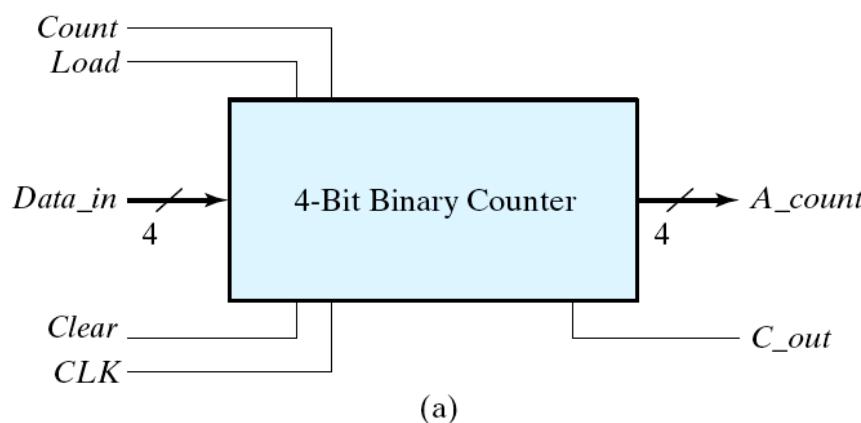
$$T_{Q8} = Q_8 Q_1 + Q_4 Q_2 Q_1$$

$$y = Q_8 Q_1$$

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



(a)

Fig. 6.14 Four-bit binary counter with parallel load

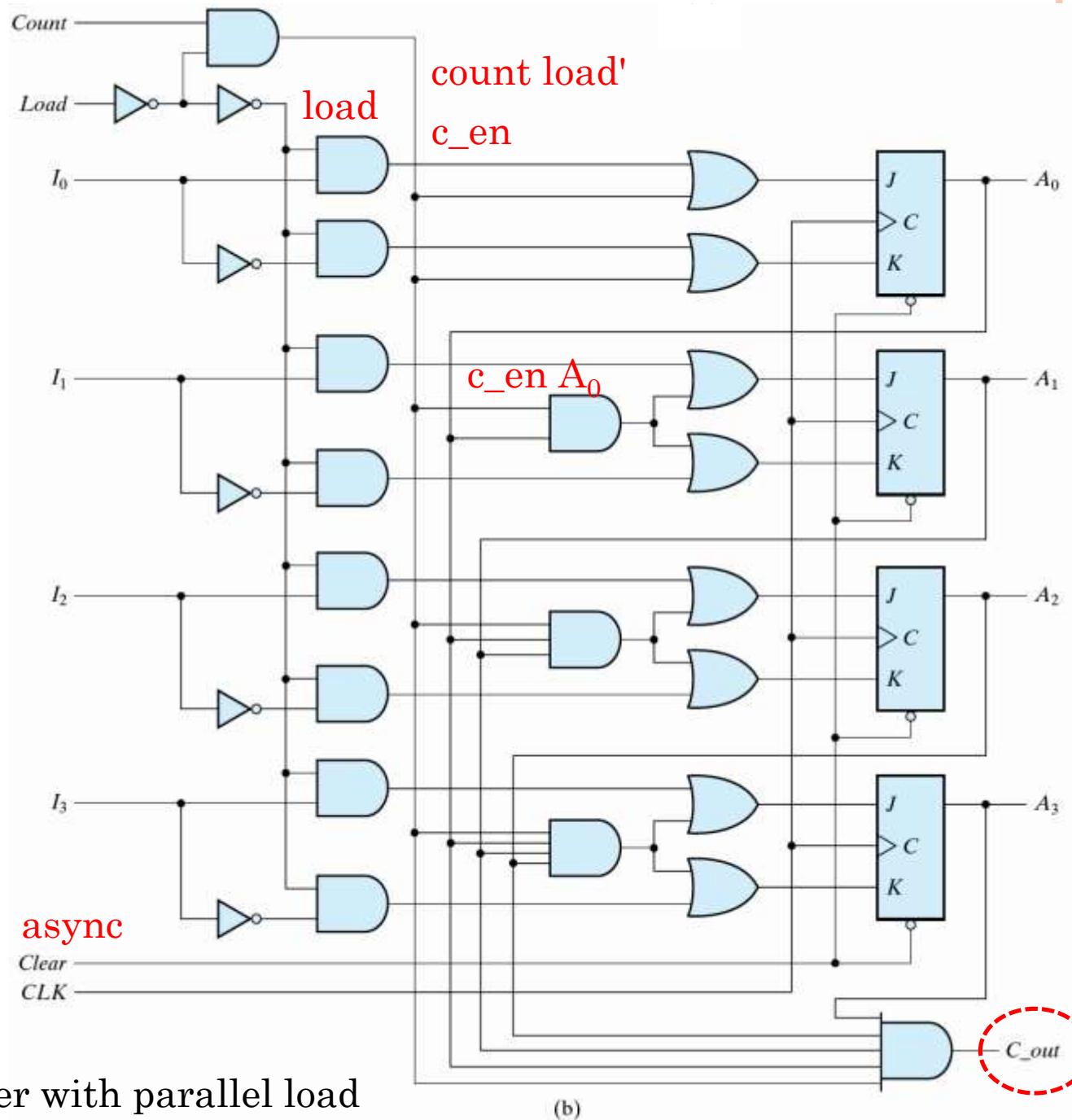
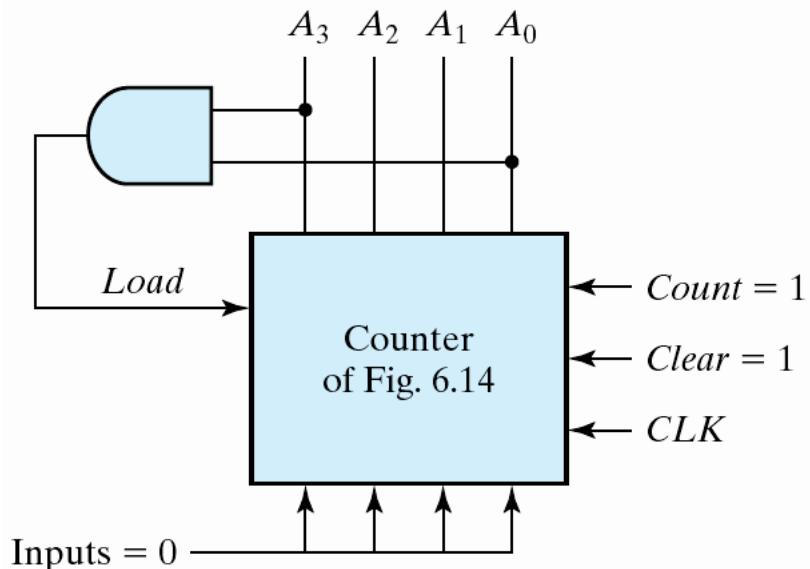


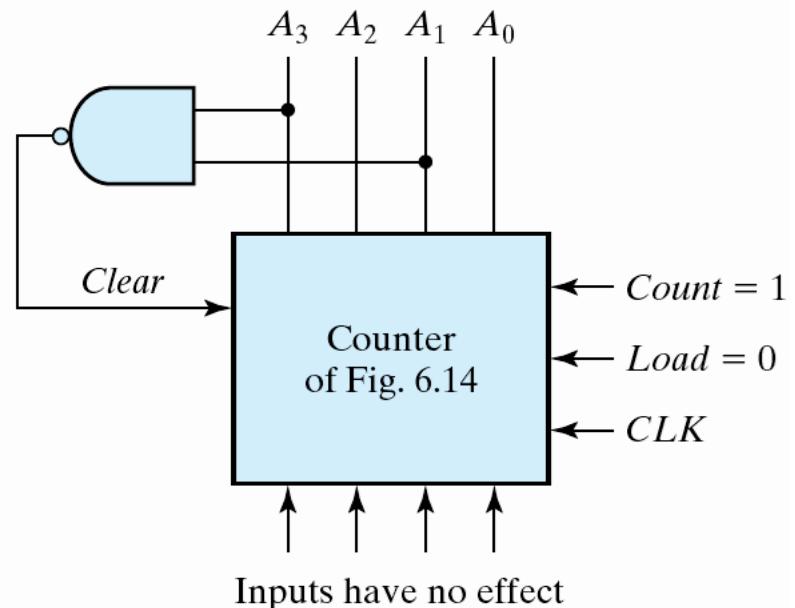
Fig. 6.14
Four-bit binary counter with parallel load

BCD COUNTER USING A COUNTER WITH PARALLEL LOAD

- Generate any count sequence
 - E.g.: BCD counter \Leftarrow Counter with parallel load



(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

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.



COUNTER WITH UNSIGNED STATES

- A circuit with n *flip-flops* has 2^n states
 - We may have to design a counter with a given sequence (unused states)
 - Unused states may be treated as don't care or assigned specific next state
 - Outside noise may cause the counter to enter unused state
 - Must ensure counter eventually goes to the valid state



COUNTER WITH UNSIGNED STATES

- 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:
 - $J_A = B, K_A = B$
 - $J_B = C, K_B = 1$
 - $J_C = B', K_C = 1$



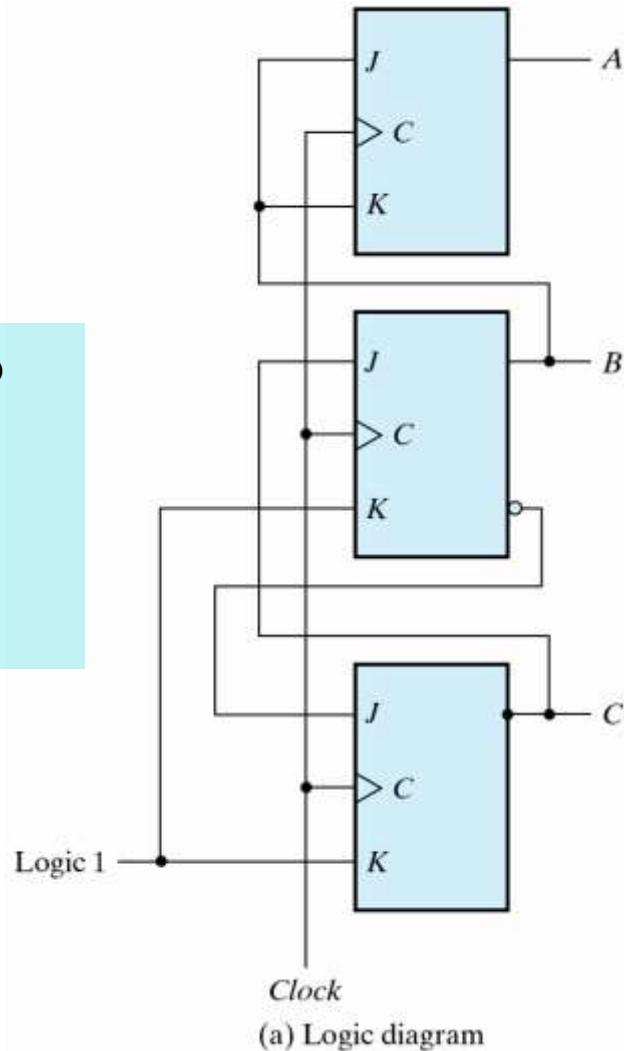
COUNTER WITH UNSIGNED STATES

The simplified flip-flop input equations:

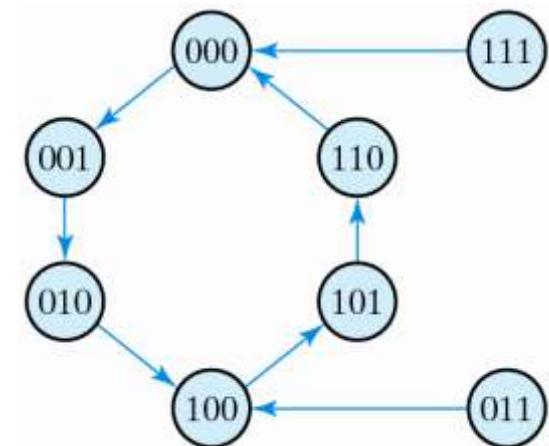
$$J_A = B, \quad K_A = B$$

$$J_B = C, \quad K_B = 1$$

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



(a) Logic diagram



(b) State diagram

RING COUNTER

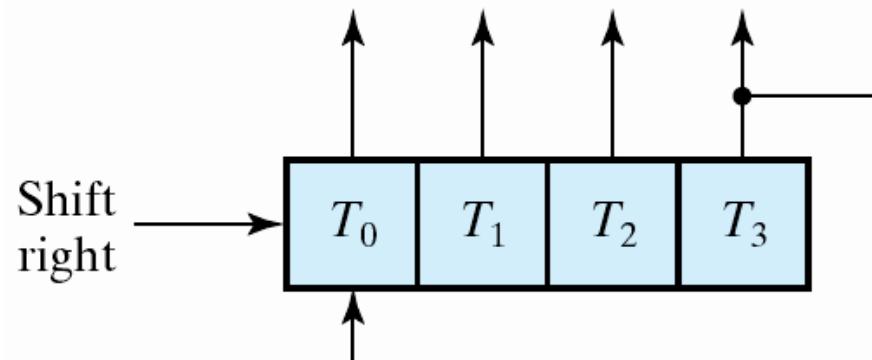
- A circular shift register with only one flip-flop being set at any particular time, all others are cleared (initial value = 1 0 0 ... 0).
- The single bit is shifted from one flip-flop to the next to produce the sequence of timing signals.



RING COUNTER

- 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



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

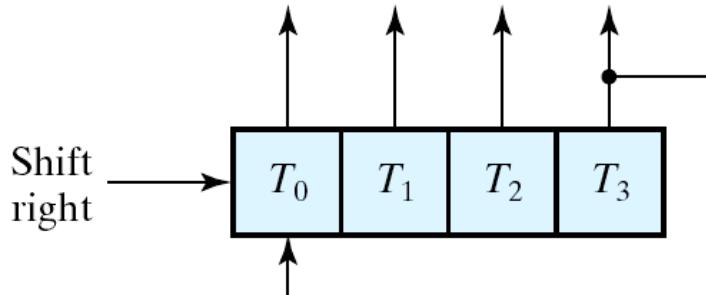
Fig. 6.17 Generation of timing signals

RING COUNTER

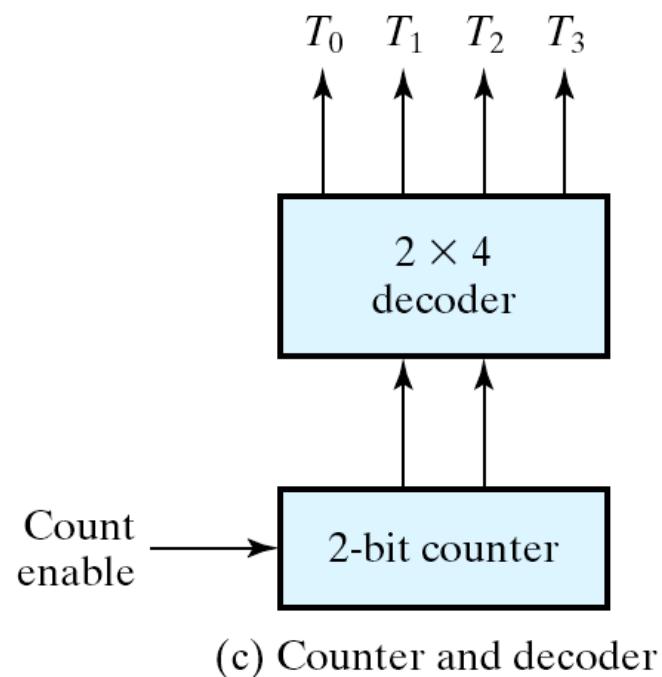
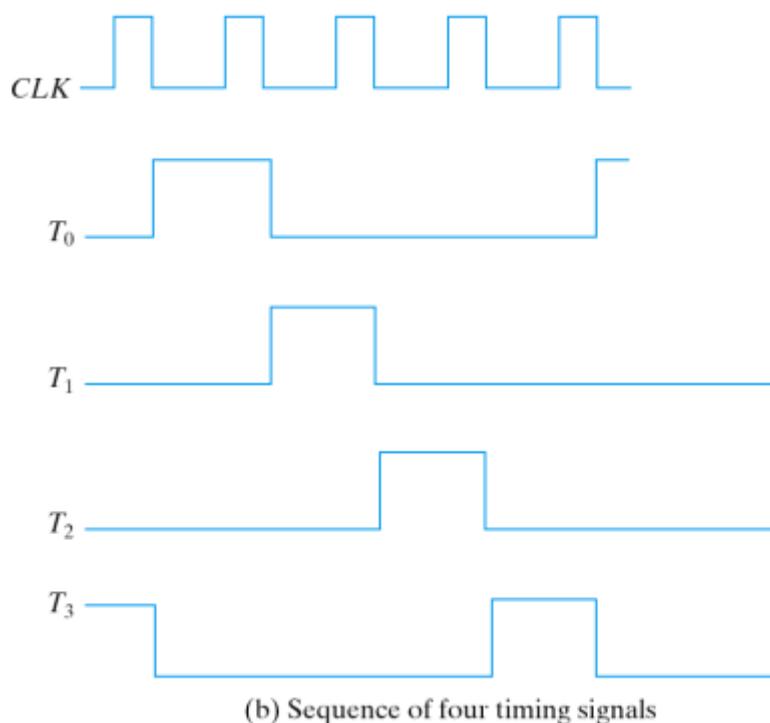
- Application of counters
 - Counters may be used to generate timing signals to control the sequence of operations in a digital system.
- Approaches for generation of 2^n timing signals
 1. A shift register with 2^n flip-flops
 2. An n -bit binary counter together with an n -to- 2^n -line decoder



GENERATION OF TIMING SIGNALS



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



(c) Counter and decoder

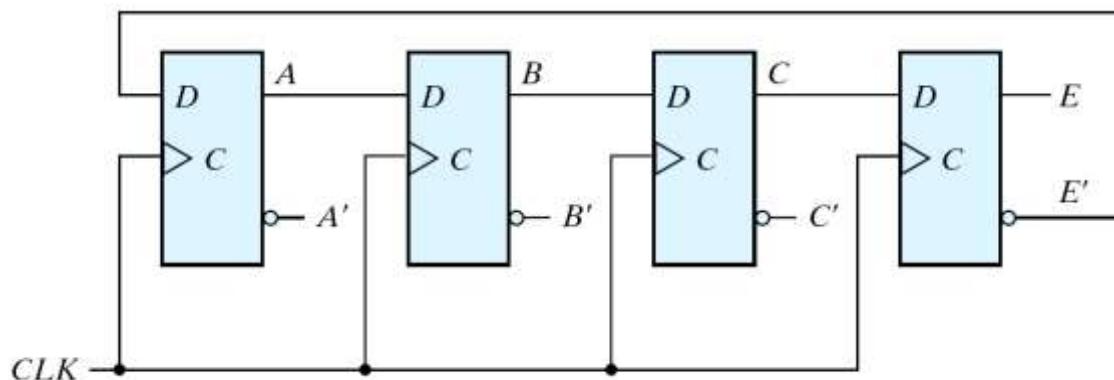
Fig. 6.17 Generation of timing signals

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
 - 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).



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

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

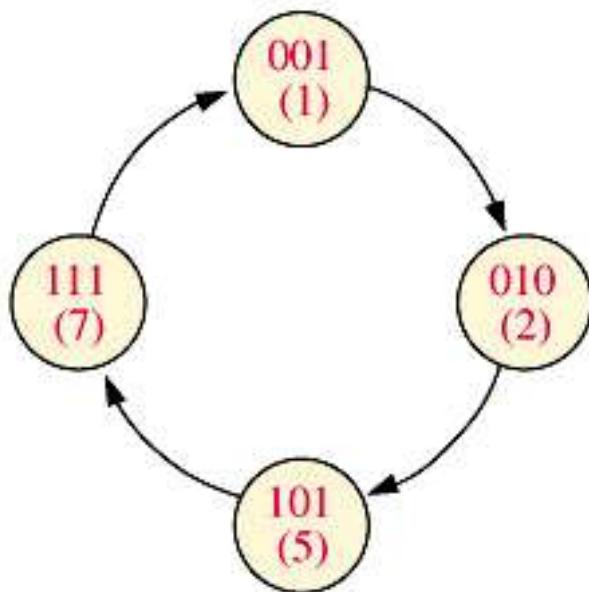
- The decoding follows a regular pattern
 - 2 inputs per decoding gate

SUMMARY

- 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.

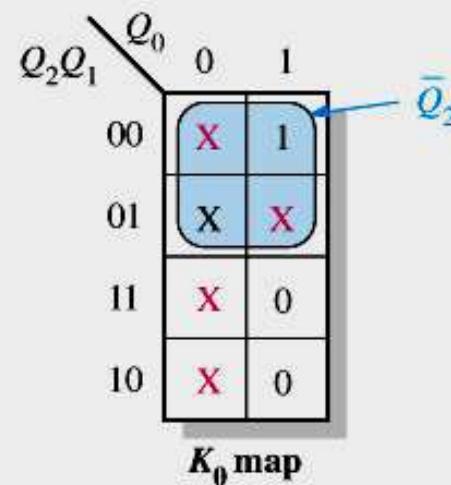
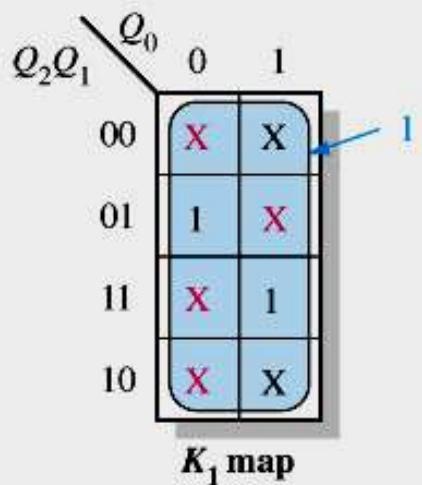
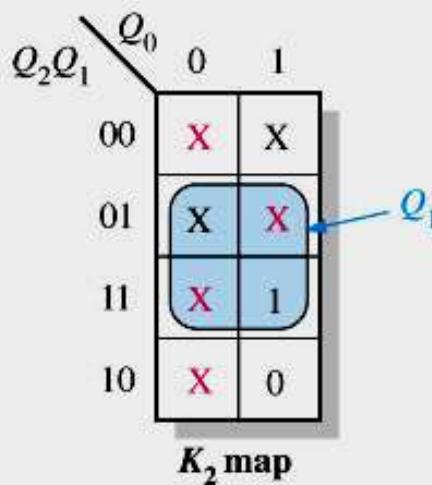
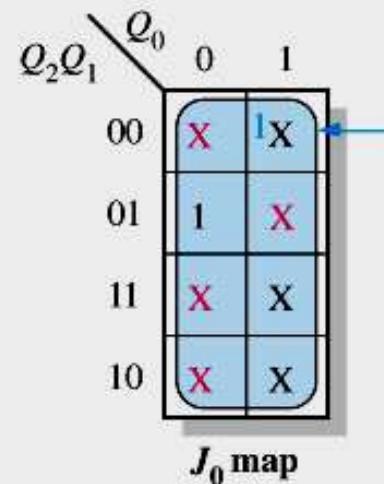
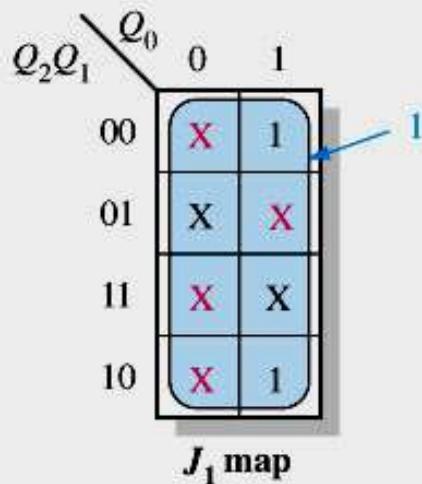
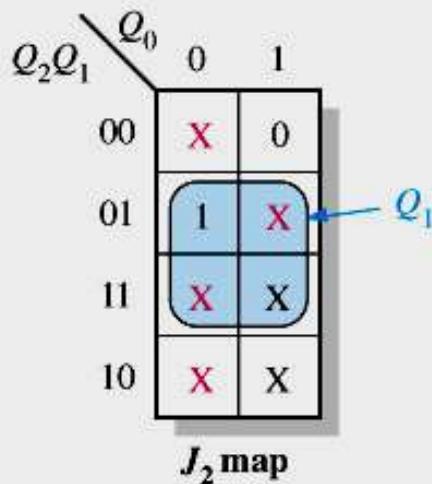


PRACTICE: COUNTER WITH IRREGULAR BINARY COUNT SEQUENCE



Present State			Next State		
Q2	Q1	Q0	Q2	Q1	Q0
0	0	1	0	1	0
0	1	0	1	0	1
1	0	1	1	1	1
1	1	1	0	0	1

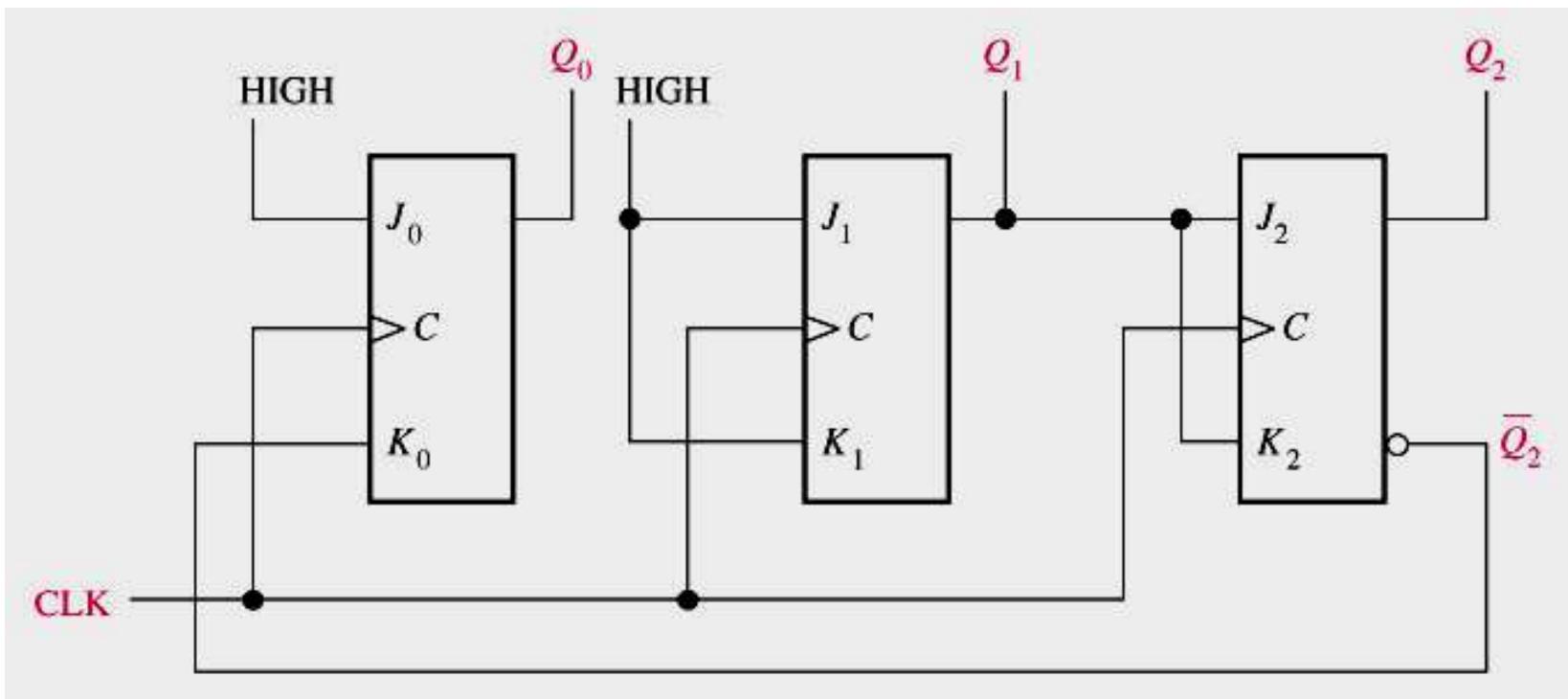
Output Transitions		Flip-Flop Inputs	
QN	QN+1	J	K
0	→ 0	0	x
0	→ 1	1	x
1	→ 0	x	1
1	→ 1	x	0



$$J_0 = 1, \quad K = \overline{Q_2}$$

$$J_1 = K_1 = 1$$

$$J_2 = K_2 = Q_1$$



- An analysis shows that if the counter, by accident, gets into one of the invalid states (0, 3, 4, 6) it will always return to a valid state according to the following sequences: $0 \rightarrow 3 \rightarrow 4 \rightarrow 7$ and $6 \rightarrow 1$.

SYLLABUS

- Chapter 6 (Excluding Section 6.6)

