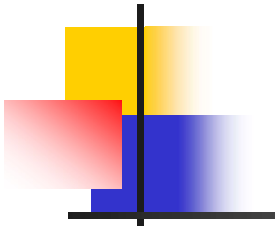
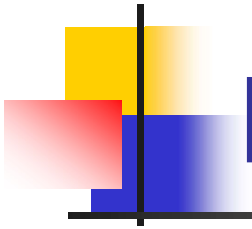
Module 5 – Part I

Registers



Registers and Counter

- The filp-flops are essential component in clocked sequential circuits.
- Circuits that include filp-flops are usually classified by the function they perform. Two such circuits are registers and counters.
- An n-bit register consists of a group of n flip-flops capable of storing n bits of binary information.



Registers

- In its broadest definition, a register consists a group of flip-flops and gates that effect their transition.
  - The flip-flops hold the binary information.
  - The gates determine how the information is transferred into the register.
- Counters are a special type of register.
- A counter goes through a predetermined sequence of states.

Registers

- Fig shows a register constructed with four D-type filpflops.
- "Clock" triggers all flip-folps on the positive edge of each pulse.
- "Clear" is useful for clearing the register to all 0's prior to its clocked operation.

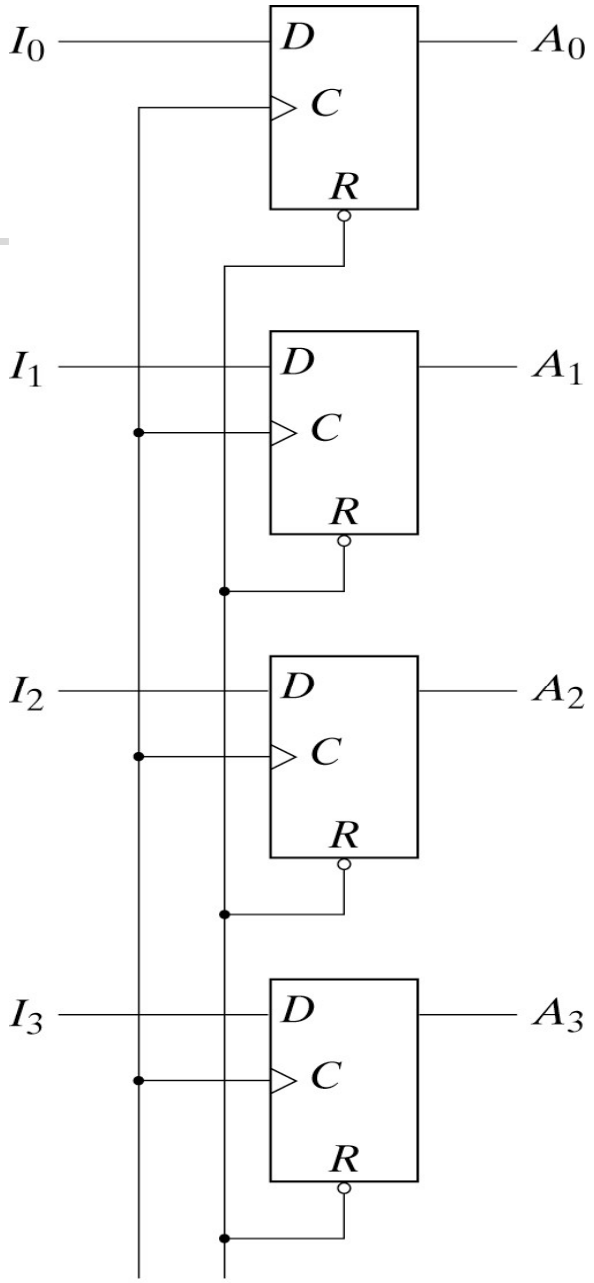
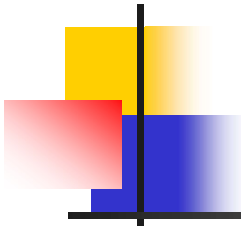


Fig. 6-1 4-Bit Register



Register with Parallel Load

- A clock edge applied to the C inputs of the register of Fig. 6-1 will load all four inputs in parallel.
- For synchronism, it is advisable to control the operation of the register with the D inputs rather than controlling the clock in the C inputs of the flip-flops.
- A 4-bit register with a load control input that is directed through gates and into the D inputs of the flip-flops si shown in Fig. 6-2.

Register with Parallel Load

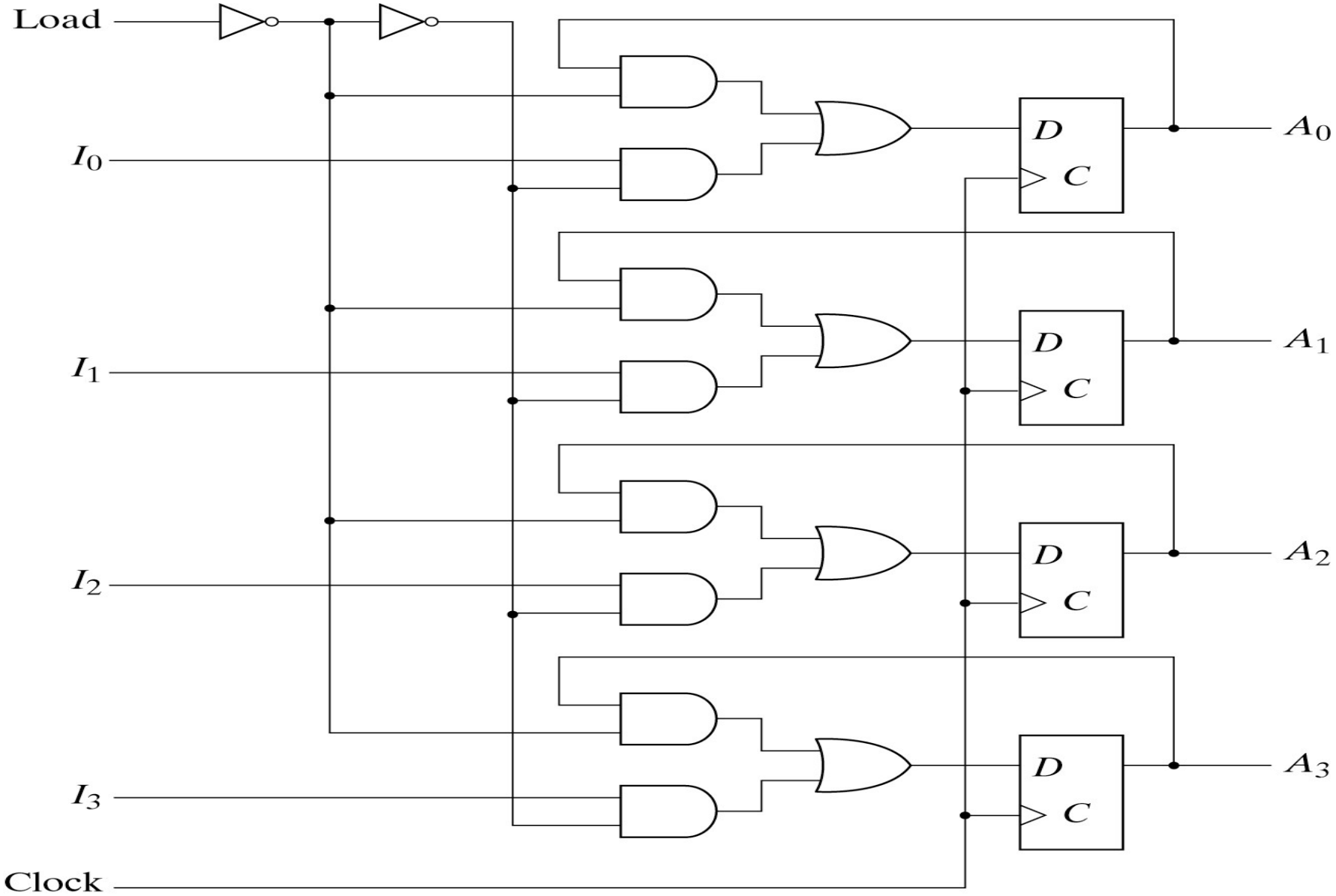
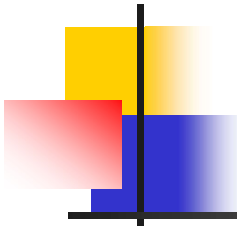


Fig. 6-2 4-Bit Register with Parallel Load



Register with Parallel Load

- When the load input is 1 , the data in the four inputs are transferred into the register with next positive edge of the clock.
- When the load input is 0 ,the outputs of the flip-flops are connected to their respective inputs.
- The feedback connection from output to input is necessary because the D flip-flops does not have a “no change” condition.

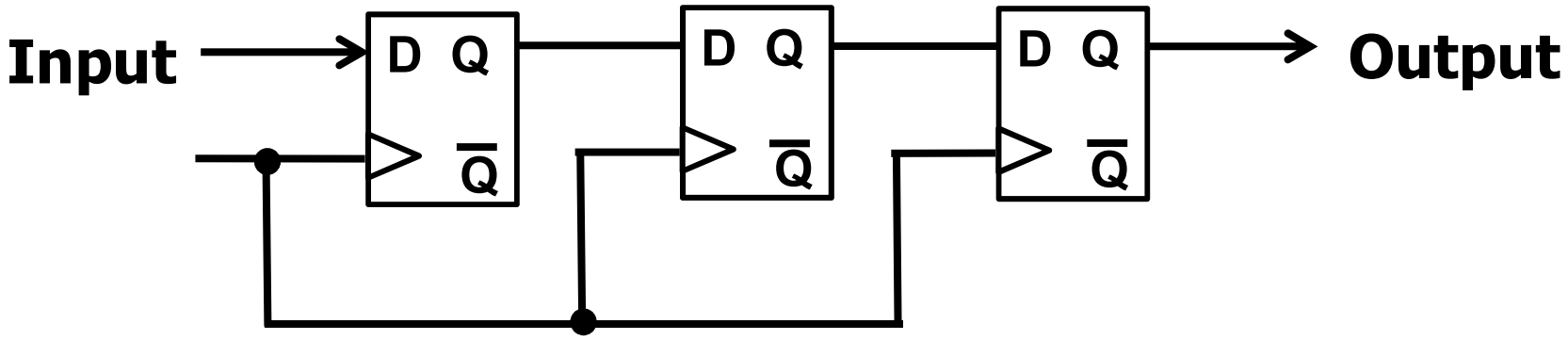
Shift Registers & its Types

Construction

- Shift registers are constructed from flip-flops due to their characteristics:
  - Edge-triggered devices
  - Output state retention
- Each Flip-Flop in a shift register can retain one binary digit.
  - For instance, if a 5-bit binary number needs to be stored and shifted, 5 flip-flops are required.
- Each binary digit transfer operation requires a clock edge.
- Asynchronous inputs are useful in resetting the whole configuration.

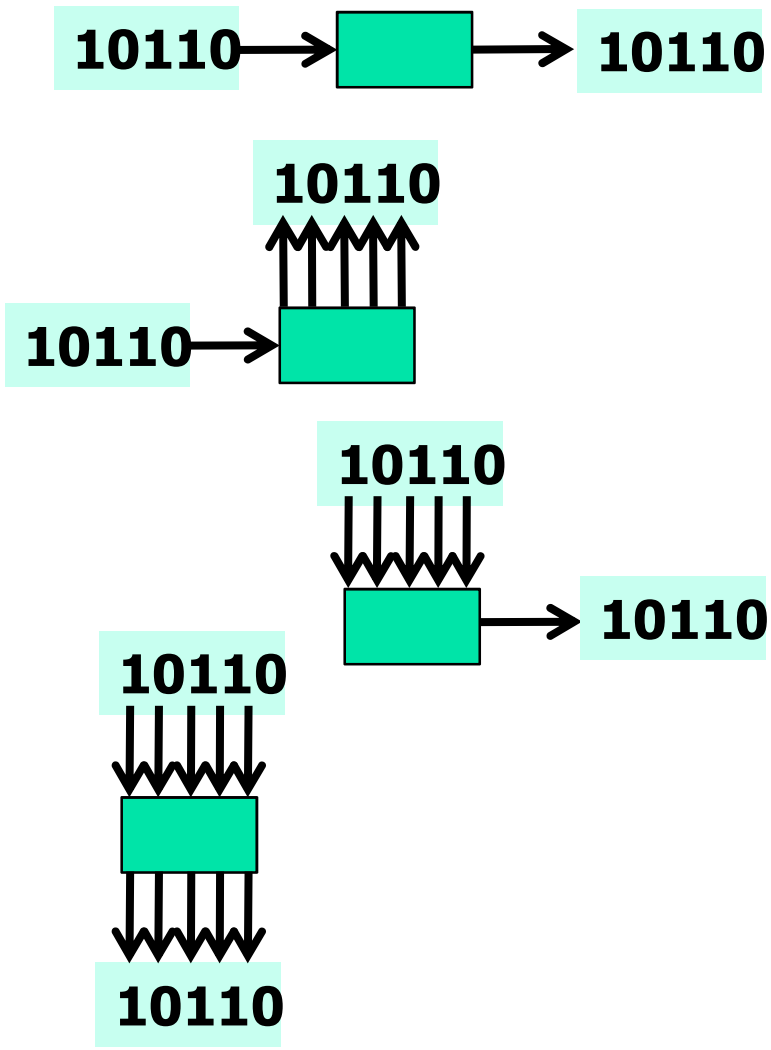
Shift Register Construction

- Shift registers are comprised of D Flip-Flops that share a common clock input.



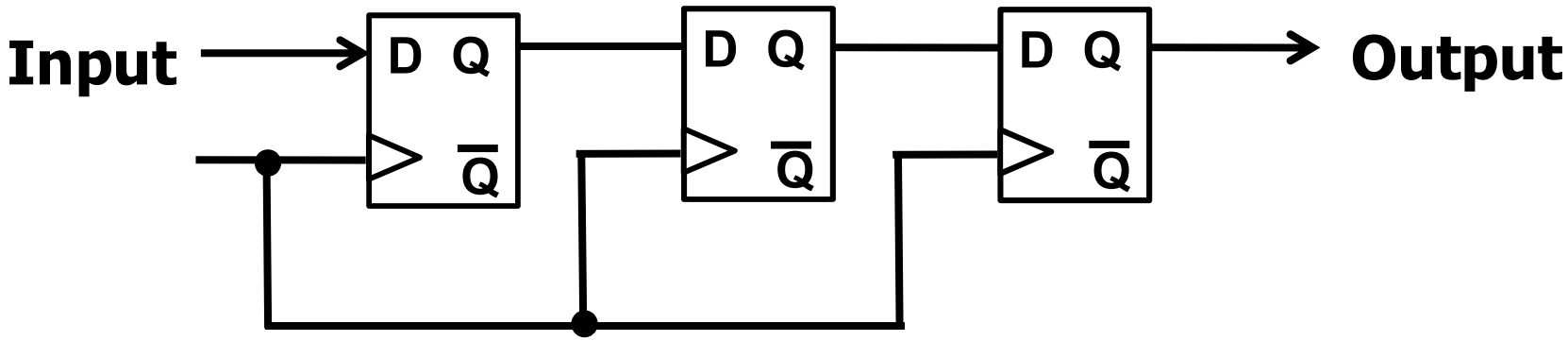
Shift Register Types

- SISO: Serial In, Serial Out
- SIPO: Serial In, Parallel Out
- PISO: Parallel In, Serial Out
- PIPO: Parallel In, Parallel Out



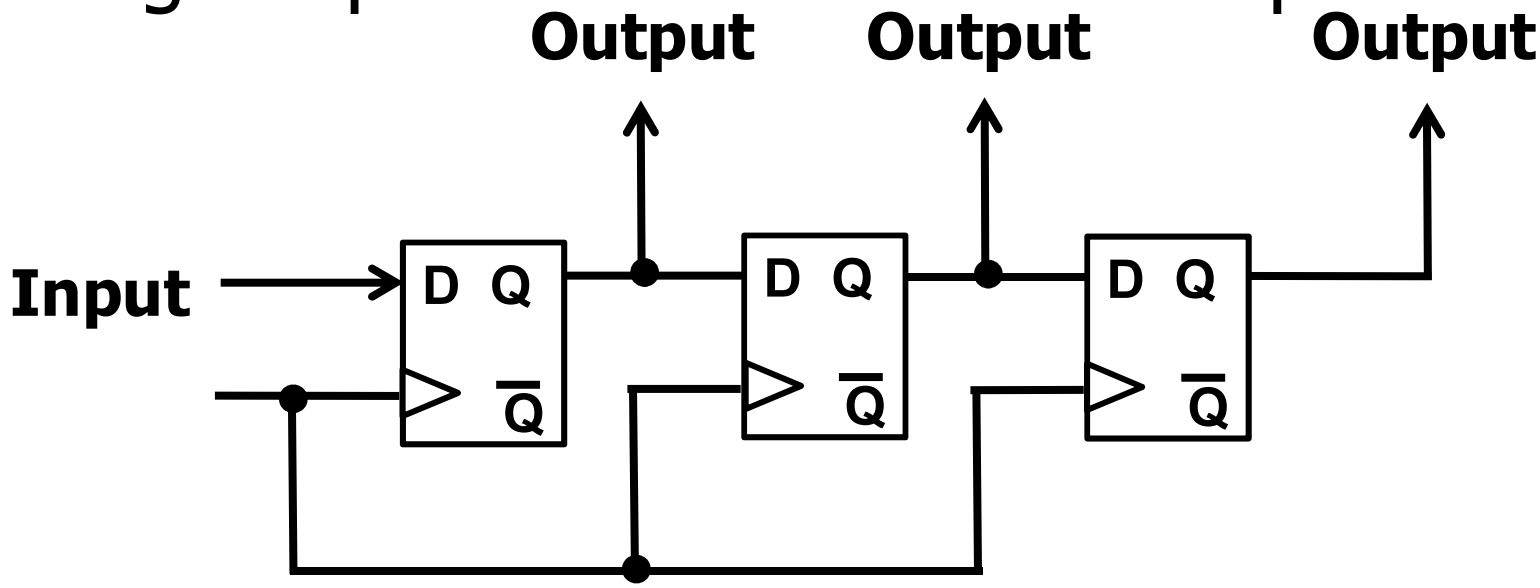
SISO Flip-Flop Shift Register

- a **Serial In Serial Out** shift register has a single input and a single output



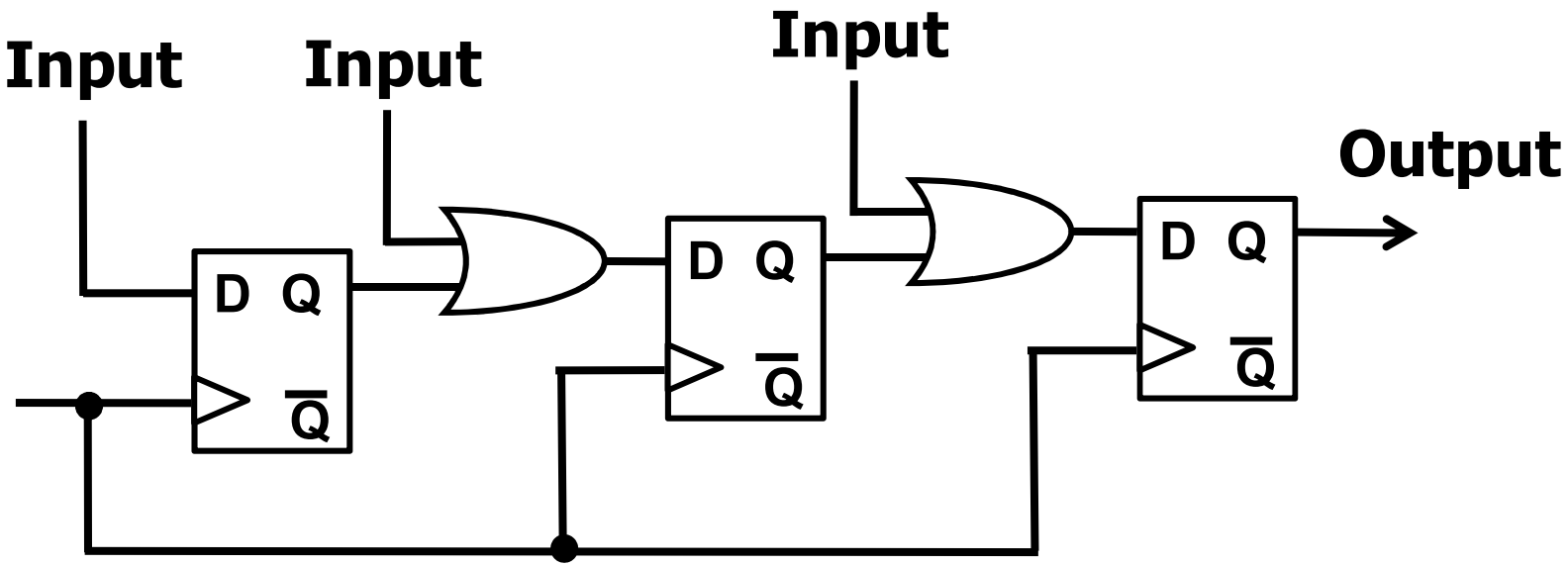
SIPO Flip-Flop Shift Register

- a **Serial In Parallel Out** shift register has a single input and access to all outputs



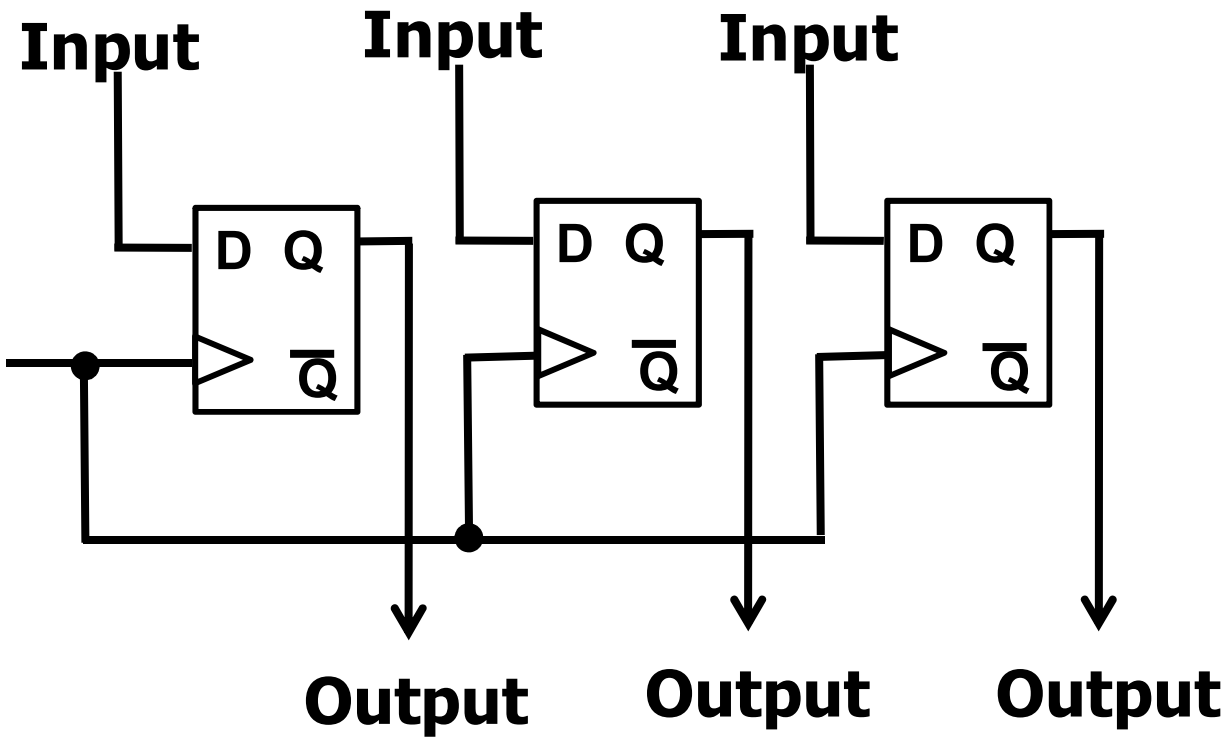
PISO Flip-Flop Shift Register

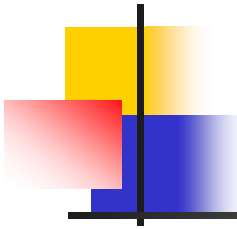
- a **Parallel In Serial Out** shift register requires additional gates. In this example the parallel input must revert to logic low; in other configurations steering gates are used to switch between loading and shifting operations.



PIPO Flip-Flop Shift Register

- a **Parallel In Parallel Out** register has the simplest configuration. It represents a memory device.

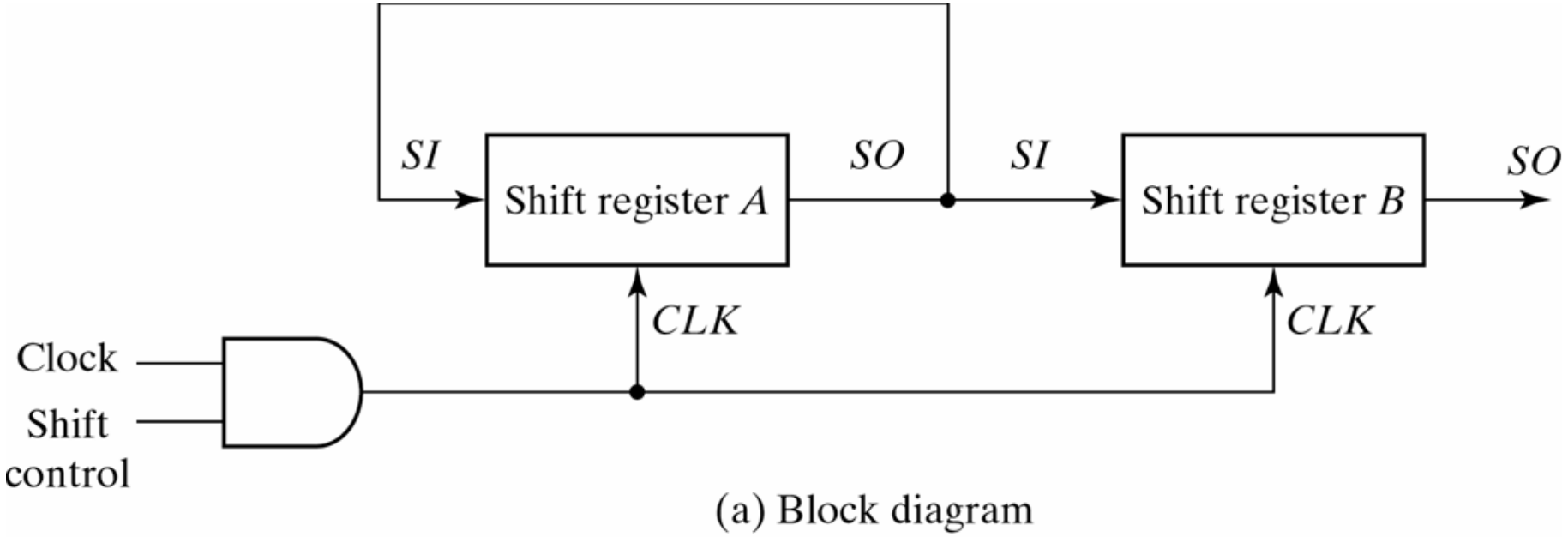




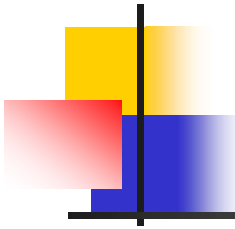
Serial Transfer

- A digital system is said tp operate in a serial mode when information is transferred and manipulated one bit at a time.
- This in contrast to parallel transfer where all the bits of the register are transferred at the same time.
- The serial transfer us done with shift registers, as shown in the block diagram of Fig. 6-4(a).

Serial Transfer



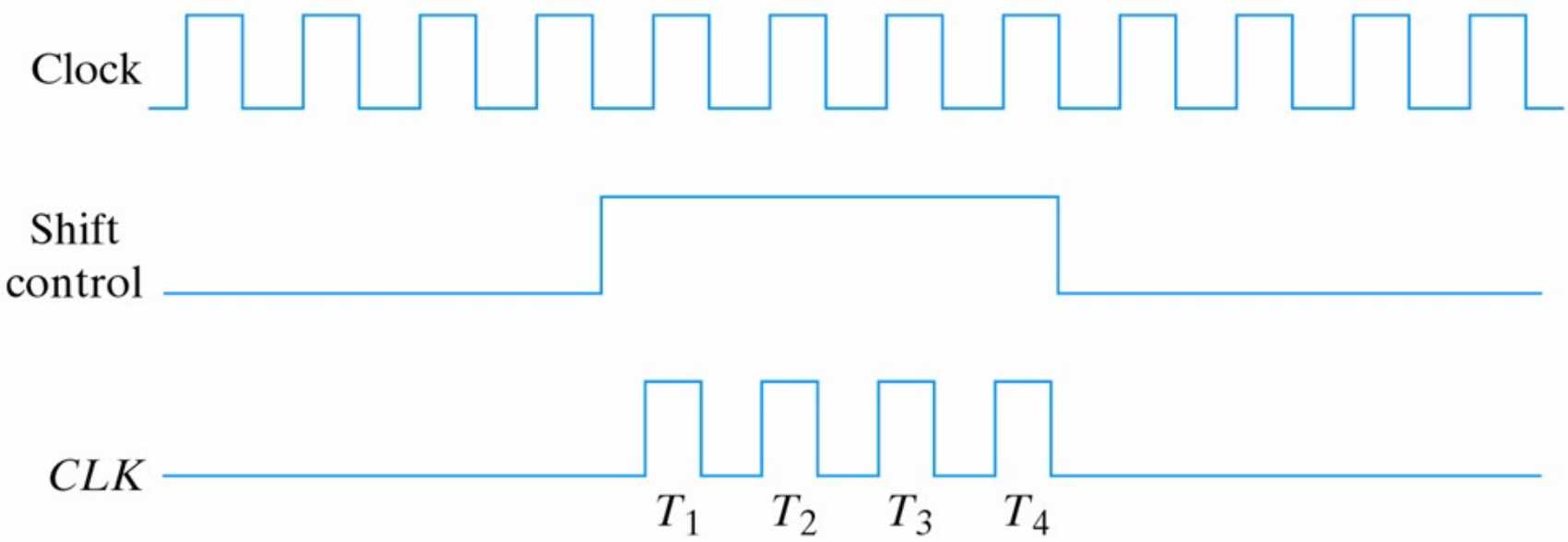
(a) Block diagram



Serial Transfer

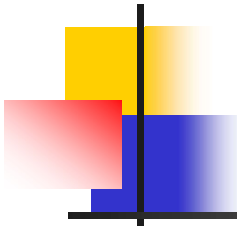
- To prevent the loss of information stored in the source register, the information in register A is made to circulate by connecting the serial output to its serial input.
- The shift control input determines when and how many times the registers are shifted. This is done with an AND gate that allows clock pulses to pass into the CLK terminals only when the shift control is active. [Fig. 6-4(a)].

Serial Transfer



(b) Timing diagram

Fig. 6-4 Serial Transfer from Register *A* to register *B*



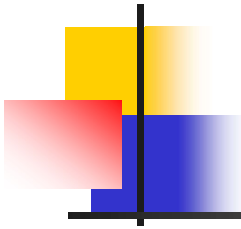
Serial Transfer

- The shift control signal is synchronized with the clock and changes value just after the negative edge of the clock.
- Each rising edge of the pulse causes a shift in both registers. The fourth pulse changes the shift control to 0 and the shift registers are disabled.

Serial Transfer

Table 6-1
Serial-Transfer Example

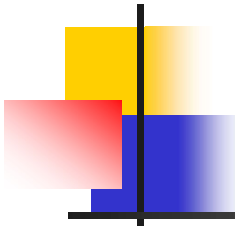
Timing Pulse	Shift Register A	Shift Register B
Initial value	1 0 1 1	0 0 1 0
After T1	1 1 0 1	1 0 0 1
After T2	1 1 1 0	1 1 0 0
After T3	0 1 1 1	0 1 1 0
After T4	1 0 1 1	1 0 1 1



Serial Transfer

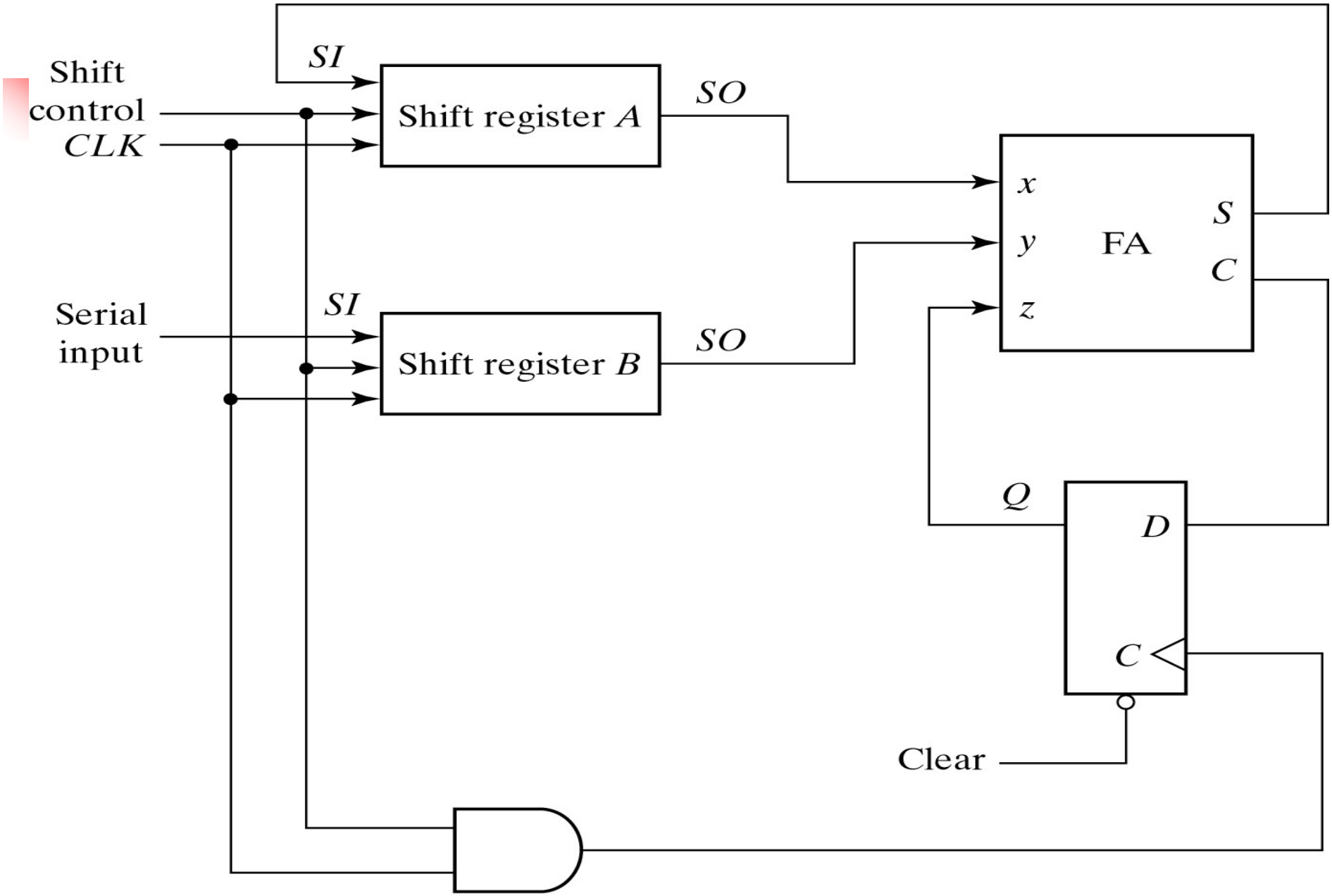
- In the parallel mode, information is available from all bits can be transferred simultaneously during one clock pulse.
- In the serial mode, the registers have a single serial input and a single serial output. The information us transferred one bit at a time while the registers are shifted in the same direction.

Serial Adder

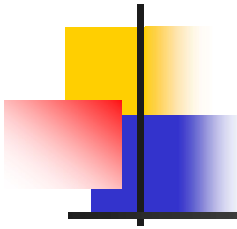


Serial Adder

- Operations in digital computers are usually done in parallel because this is a faster mode of operation.
- Serial operations are slower, but have the advantage of requiring less equipment.
- The two binary numbers to be added serially are stored in two shift registers.
- Bits are added one pair at a time through a single full adder.

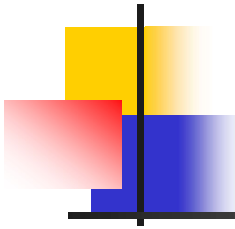


Block Diagram – Serial Adder



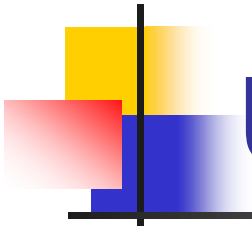
Serial Adder

- By shifting the sum into A while the bits of A are shifted out, it is possible to use one register for storing both the augend and sum bits.
- The carry out of the full adder is transferred to a D flip-flop.
- The output of the D flip-flop is then used as carry input for the next pair of significant bits.



Serial Adder

- To show that serial operations can be designed by means of sequential circuit procedure, we will redesign the serial adder using a state table.
- The serial outputs from registers are designated by x and y.
- The sequential circuit proper has two inputs, x and y, that provide a pair of significant bits, an output S that generates the sum bit, and flip-flop Q for storing the carry. [Table. 6-2]



University Question

- With the help of a neat diagram discuss how a serial adder can be designed using full-adder, shift registers and flip-flop. (10 marks) (May 2019 KTU)

Design a serial adder using JK flip flop

Table: State Table for serial Adder using JK

Present State	Inputs		Next State	Output	Flip-Flop Inputs	
Q	X	y	Q (Carry)	S (Sum)	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

- The two flip-flop input equations and the output equation can be simplified by means of map to obtain
  - $J_Q = xy$
  - $K_Q = x'y' = (x+y)'$
  - $S = x \oplus y \oplus Q$
- The circuit diagram is shown in following fig

Serial Adder using JK

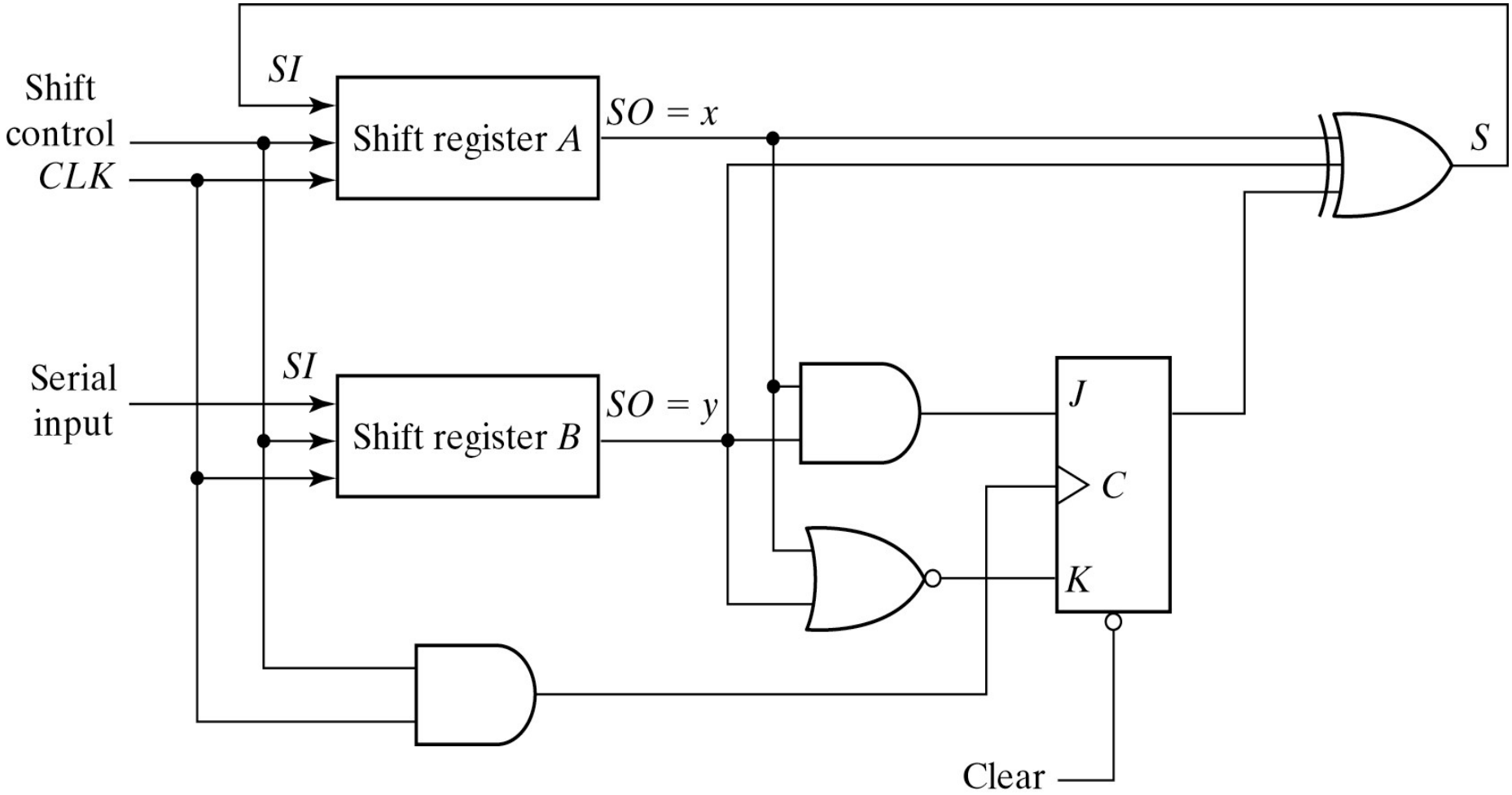
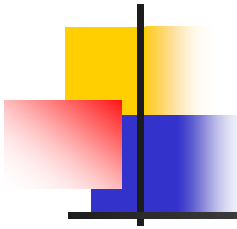


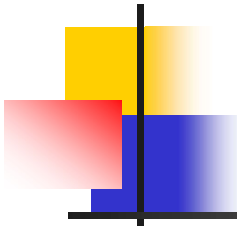
Fig. 6-6 Second form of Serial Adder

Universal Shift Register



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 operation and the *serial input* and *output* lines associated with the shift right.
- A *shift-left* control to enable the shift operation and the *serial input* and *output* lines associated with the shift left.



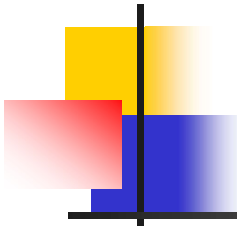
Universal Shift Register

- A *parallel-load* control to enable a parallel transfer and the n input lines associated with the parallel transfer.
- n parallel output lines.
- A control state that leaves the information in the register unchanged in the presence of the clock.
- If the register has both shifts and parallel load capabilities, it is referred to as a *universal shift register.*

Universal Shift Register

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



Universal Shift Register

- Shift registers are often used to interface digital system situated remotely from each other.
- If the distance is far, it will be expensive to use n lines to transmit the n bits in parallel.
- Transmitter performs a parallel-to-serial conversion of data and the receiver does a serial-to-parallel conversion.

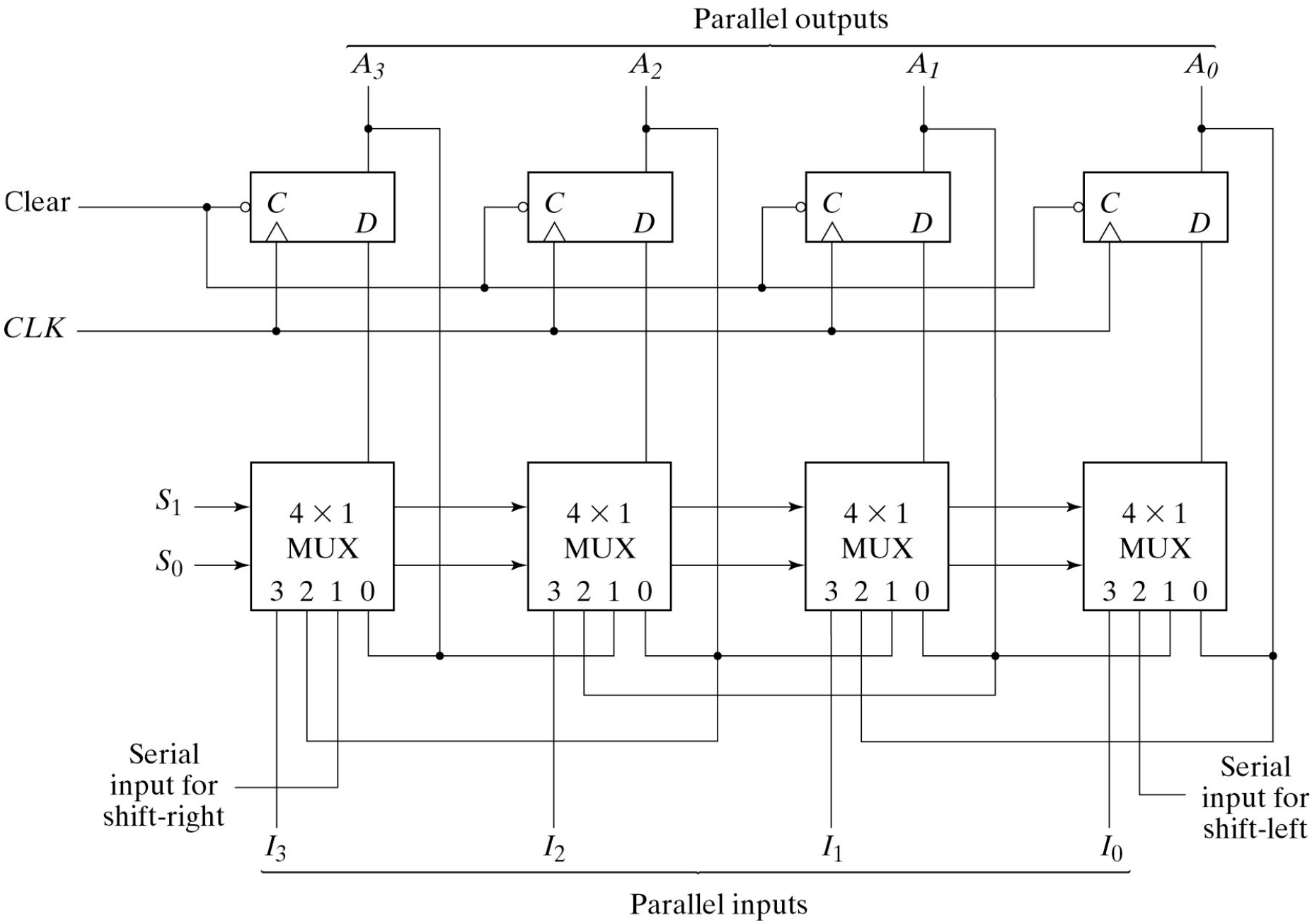
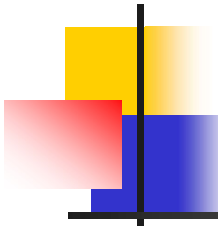


Fig. 6-7 4-Bit Universal Shift Register

Module 5 – Part II

Counters



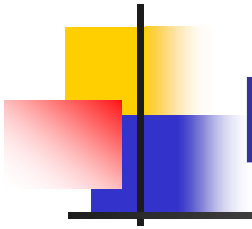
Counters Types

- Asynchronous Counters Or Ripple Counters
- Synchronous Counters

Differences between Asynchronous & Synchronous Counters

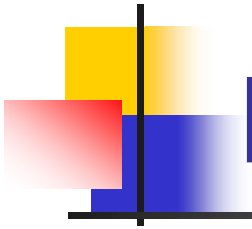
	Asynchronous	Synchronous
Circuit	The logic circuit of this type of counters is simple to design and we feed output of one FF to clock of next FF	The circuit diagram for type of counter becomes difficult as number of states increase in the counter
Propagation Time	Propagation time delay of this type of counter is : Tpd = N * (Delay of 1 FF) which is quiet high N is number of FFs	Propagation time delay of this type of counter is: Tpd = (Delay of 1 FF) + delay of 1 gate Inclusion of delay of 1 gate would be illustrated when we design higher counters:
Clock	Clock is same for all Flip flops	Clock is different for all Flip flops
Maximum operating frequency	And hence operating frequency is Low	And hence operating frequency is Higher
Speed	slow	fast
Components required	Less	more
Cost	Less	more
Disadvantage	Glitches	Lock out condition.

Ripple Counters



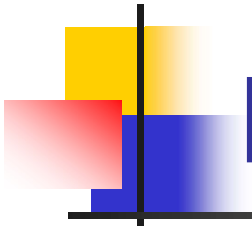
Ripple Counters

- A register that goes through a prescribed sequence of states upon the application of input pulse is called a counter.
- A counter that follows the binary number sequence is called a binary counter.



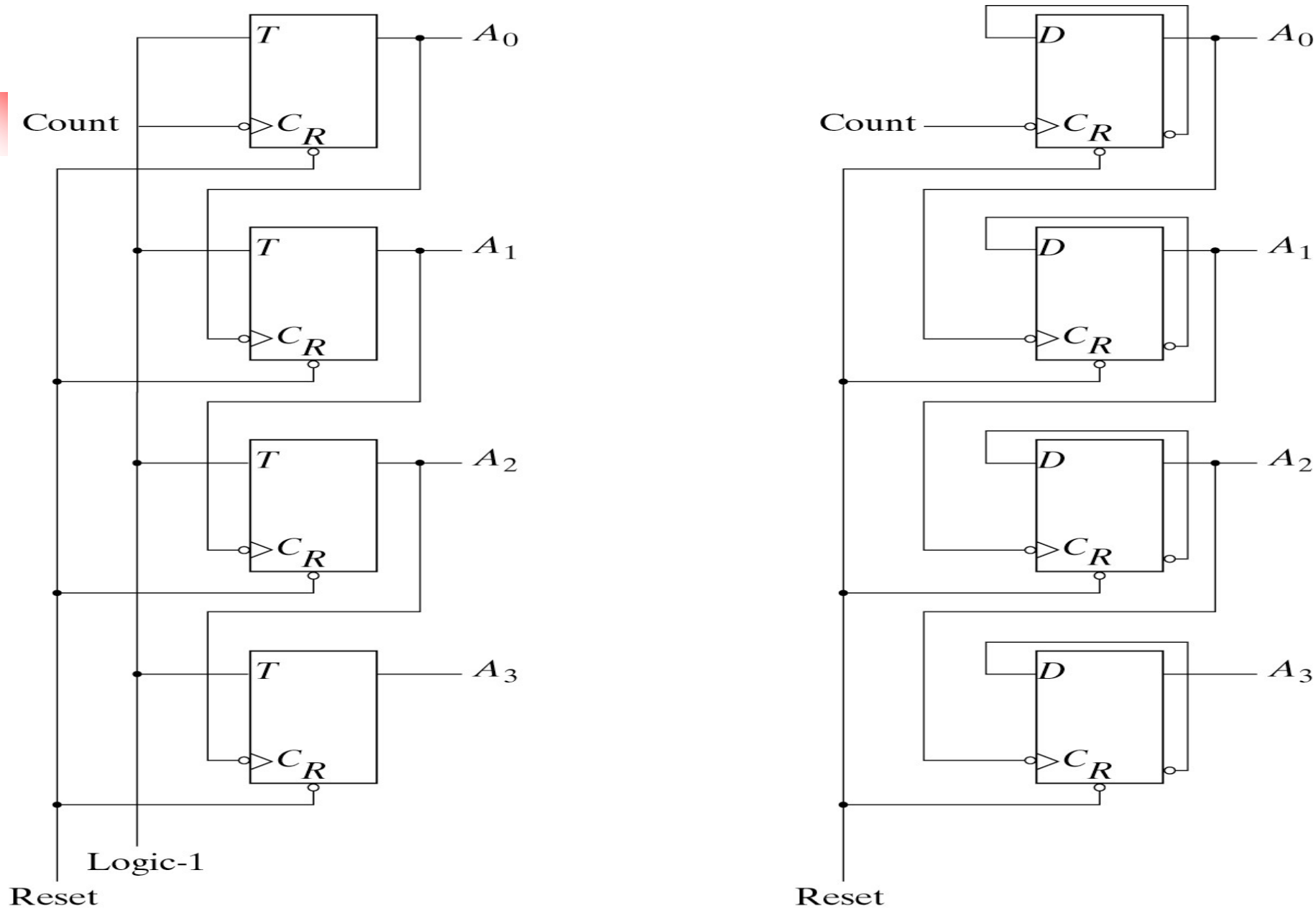
Ripple Counters Types

- Binary Ripple Counters
- BCD Ripple Counters



Binary Ripple Counters

- Counters whose counting sequence corresponds to that of the binary numbers are called binary counters.
- Modulus is 2^n, where n is the number of flip-flops in the counter.
- Binary up-counter, binary down-counter



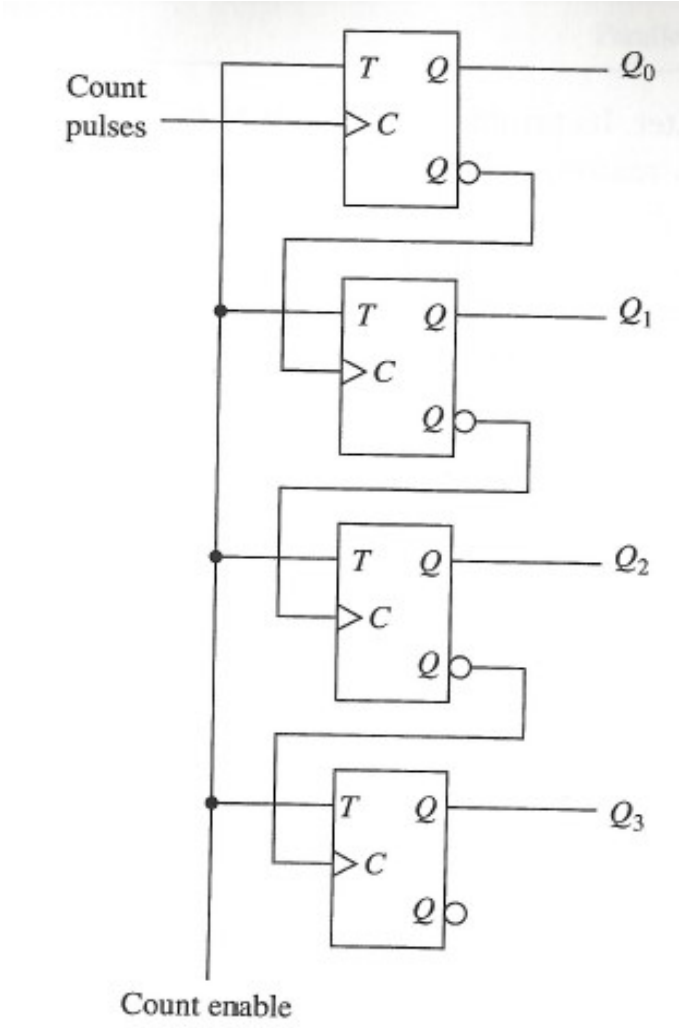
(a) With T flip-flops

(b) With D flip-flops

Fig. 6-8 4-Bit Binary Ripple Counter

4-bit Binary Ripple Counter

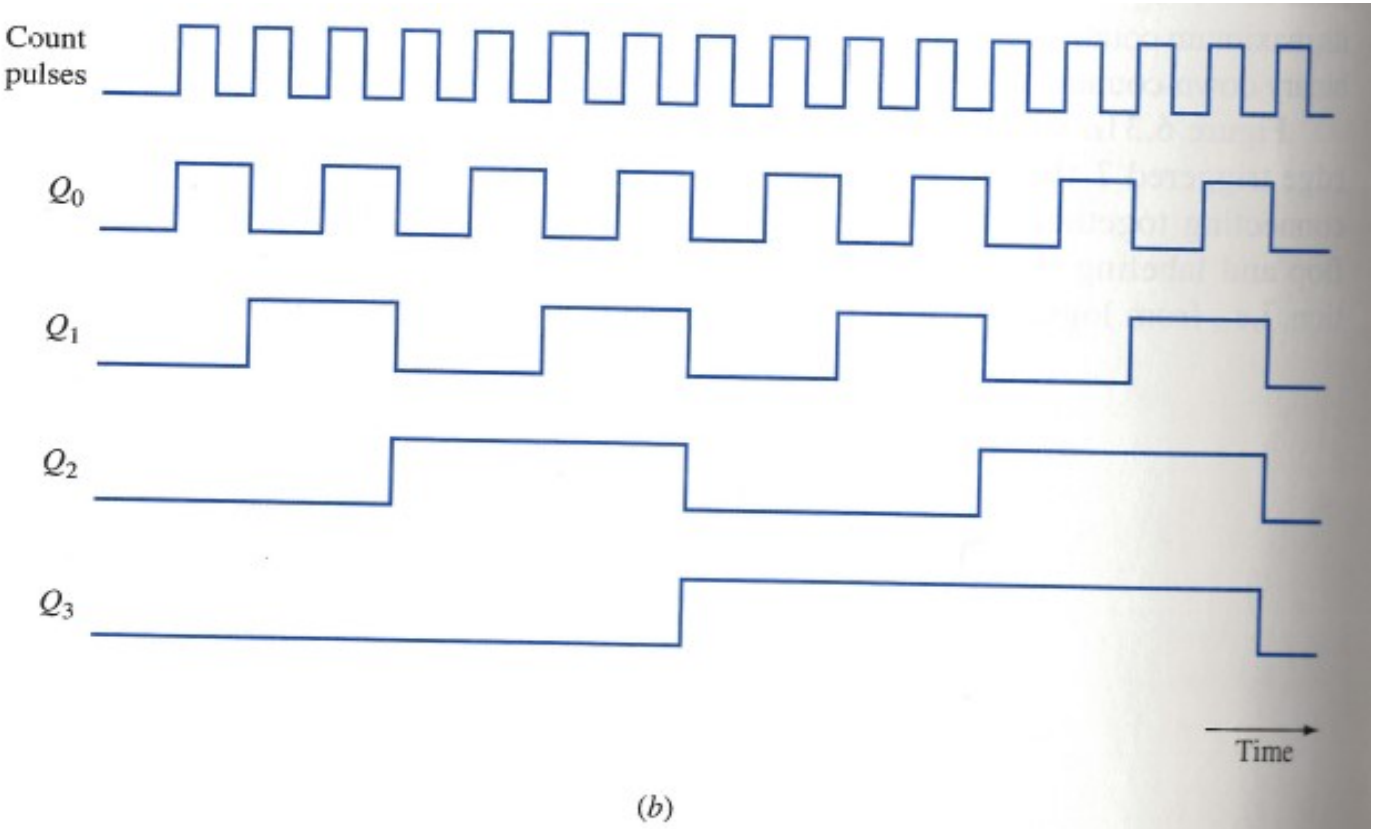
- Recall positive edge-triggered T-Flip-Flop.
  - Each positive transition from logic-0 to logic-1 causes the flip-flop to toggle.



Q_3	Q_2	Q_1	Q_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
0	0	0	0

etc.

4-bit Binary Ripple Counter

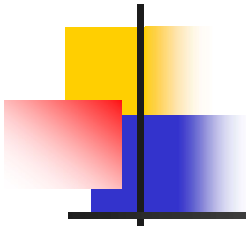


(*b*)

Figure 6.31 Four-bit binary ripple counter. (*a*) Logic diagram. (*b*) Timing diagram. (*c*) Counting sequence.

4-bit Binary Ripple Counter

- Known as a ripple counter since a change in the state of the Q_{i-1} flip-flop is used to toggle the Q_i flip-flop.
  - The effect of a count pulse must ripple through the counter.
  - Ripple counters also referred to as asynchronous counters.
- Propagation Delay
  - There is a propagation delay between the input and output of a flip-flop.
  - Rippling behavior affects the overall time delay between the occurrence of a count pulse and when the stabilized count appears at the output terminals.
  - Worst Case?
  - Going from $111 \cdots 111$ to $000 \cdots 000$ since toggle signals must propagate through the entire length of the counter.
  - For n-stage binary ripple counter, the worst case time is $n \cdot t_{pd}$, where t_{pd} is the propagation delay time associated with each flip-flop.



BCD Ripple Counter

- A decimal counter follows a sequence of ten states and returns to 0 after the count of 9.
- This is similar to a binary counter, except that the state after 1001 is 0000.
- The operation of the counter can be explained by a list of conditions for flip-flop transitions.

BCD Ripple Counter

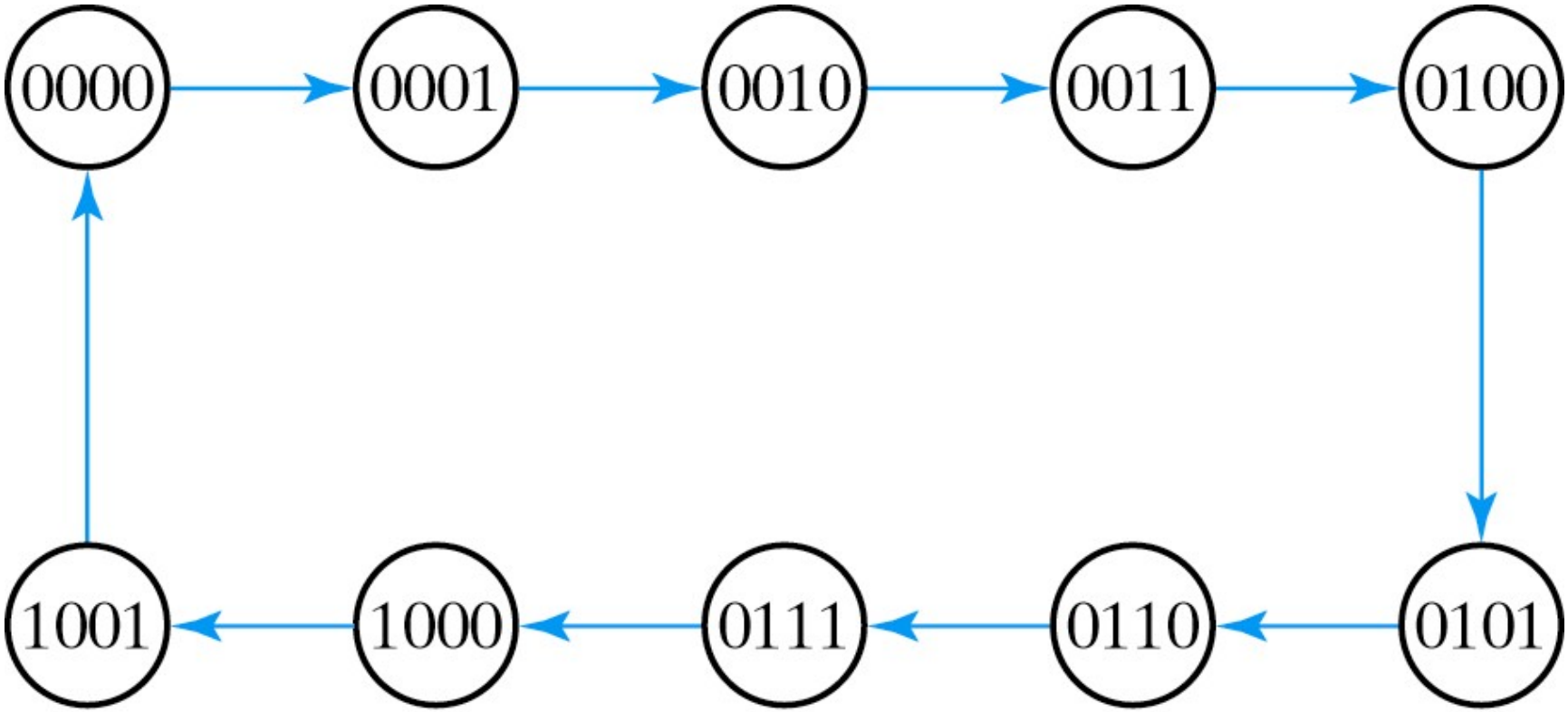


Fig. 6-9 State Diagram of a Decimal BCD-Counter

BCD Ripple Counter

- The four outputs are designated by the letter symbol Q with a numeric subscript equal to the binary weight of the corresponding bit in the BCD code.

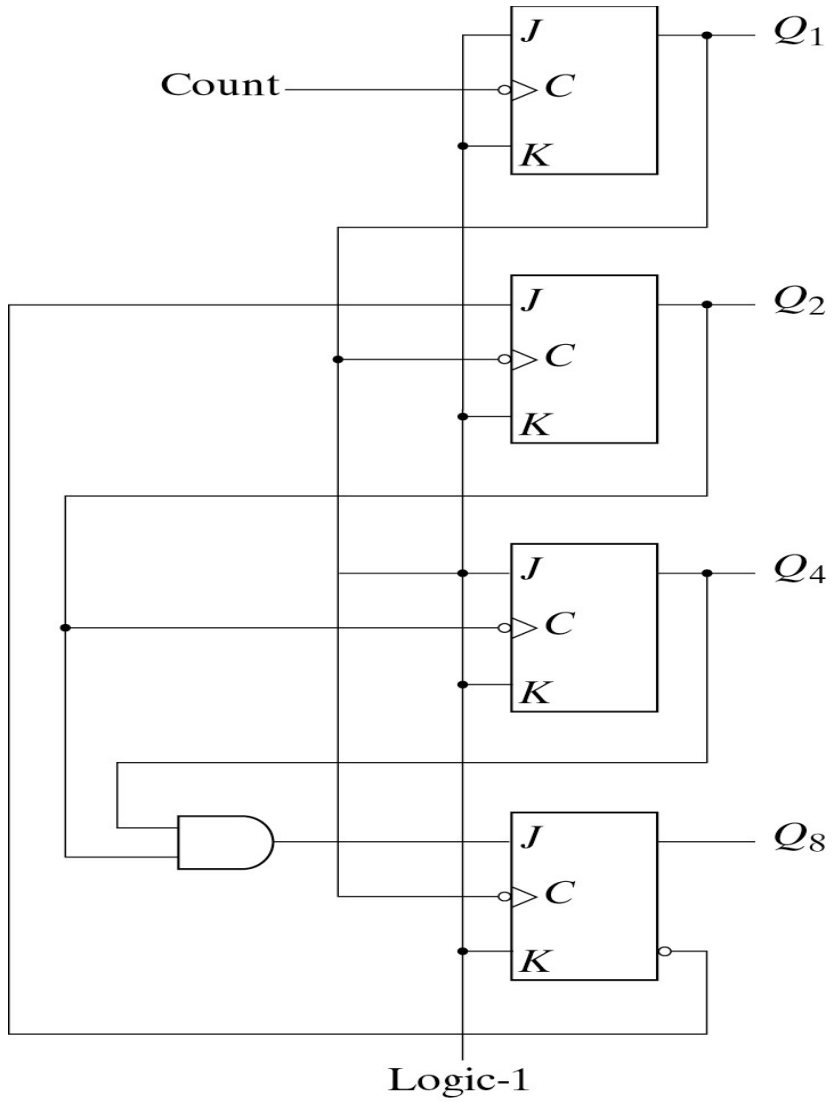
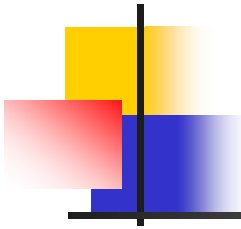


Fig. 6-10 BCD Ripple Counter



BCD Ripple Counter

- The BCD counter of [Fig. 6-9] is a decade counter.
- To count in decimal from 0 to 999, we need a three-decade counter. [Fig. 6-11]
- Multiple decade counters can be constructed by connecting BCD counters ic cascade, one for each decade.

BCD Ripple Counter

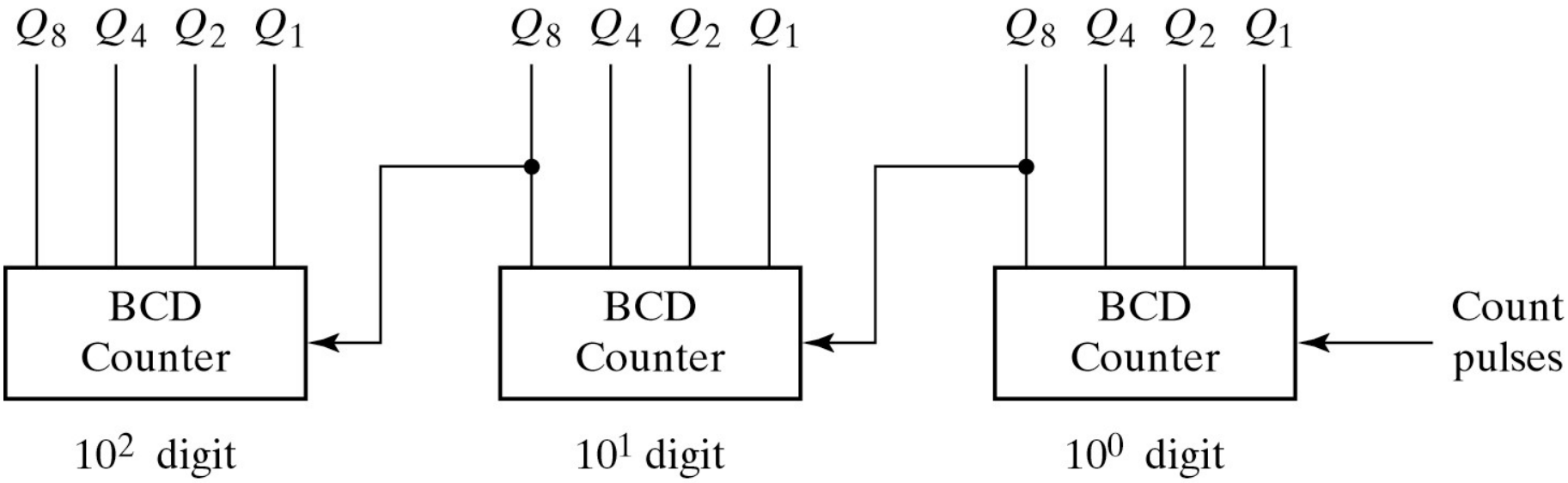
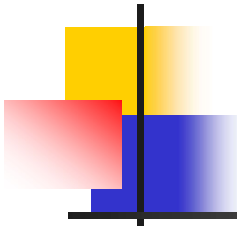


Fig. 6-11 Block Diagram of a Three-Decade Decimal BCD Counter

Synchronous Counters



Synchronous Counters

- Synchronous counters are different from ripple counters in that clock pulses are applied to the inputs of all flip-flops.
- A common clock triggers all flip-flops simultaneously rather than one at a time in succession as in a ripple counter.

Up-Down Binary Counter

- The two operations can be combined in one circuit to form a counter capable of counting up or down.
- It has an up control input and down control input.

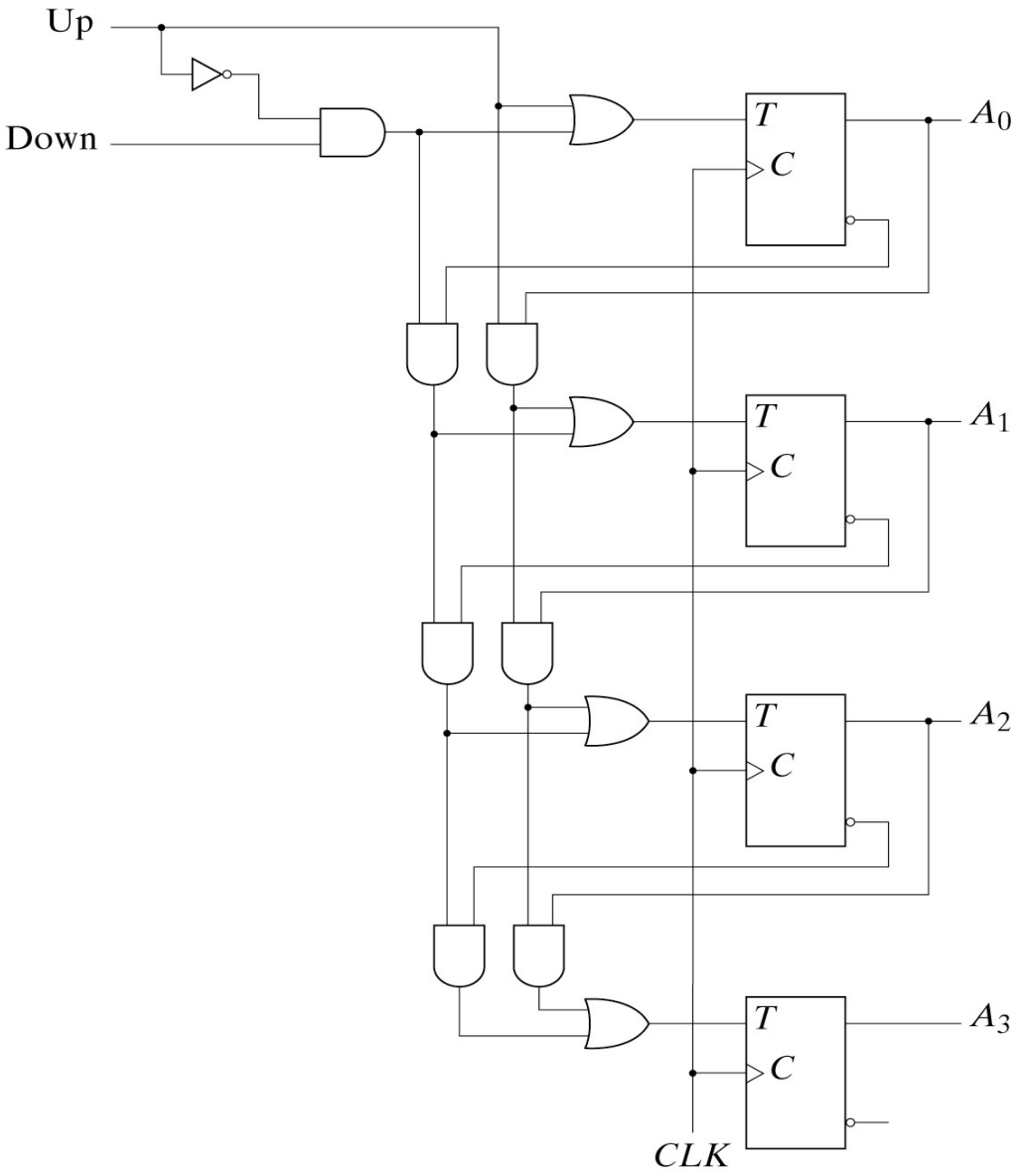
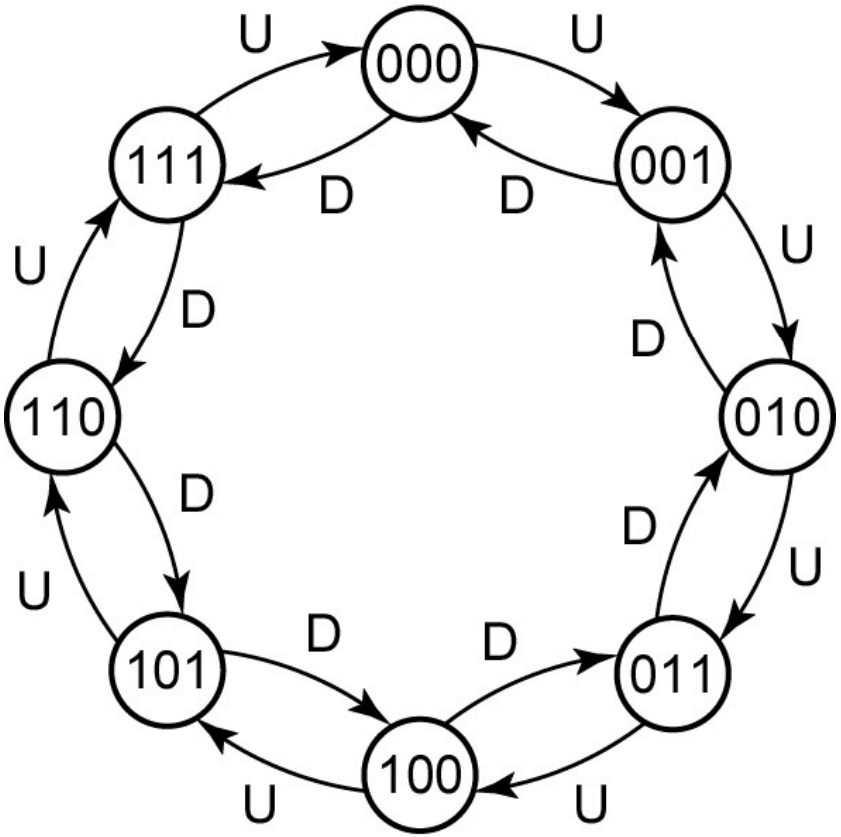


Fig. 6-13 4-Bit Up-Down Binary Counter

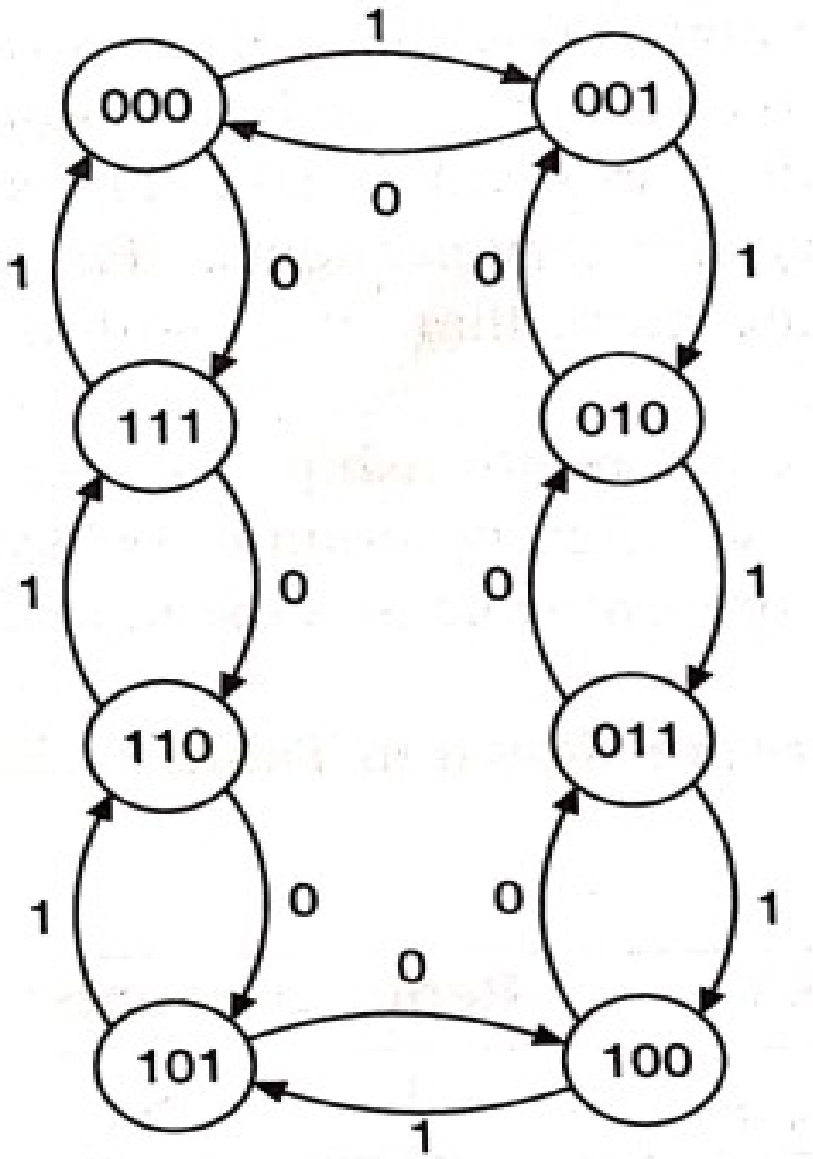
Example: Binary Up/Down Counter



State Graph and Table for Up-Down Counter

CBA	$C^+B^+A^+$	
	U	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

Design of a synchronous 3 bit Up-down Counter Using JK



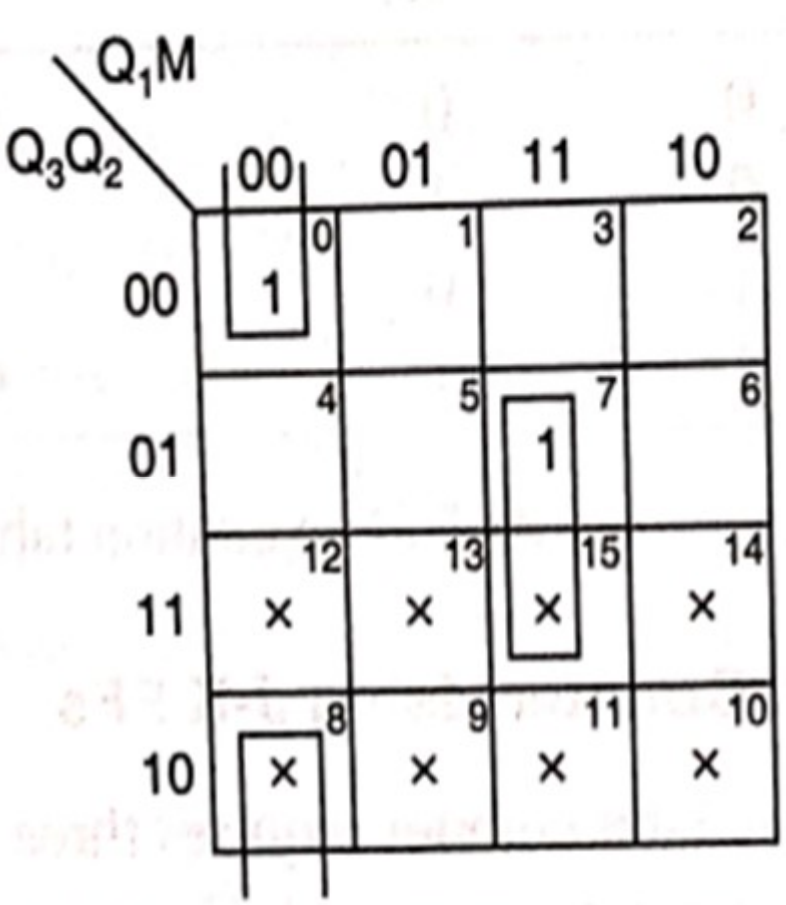
(a) State diagram

Excitation Table

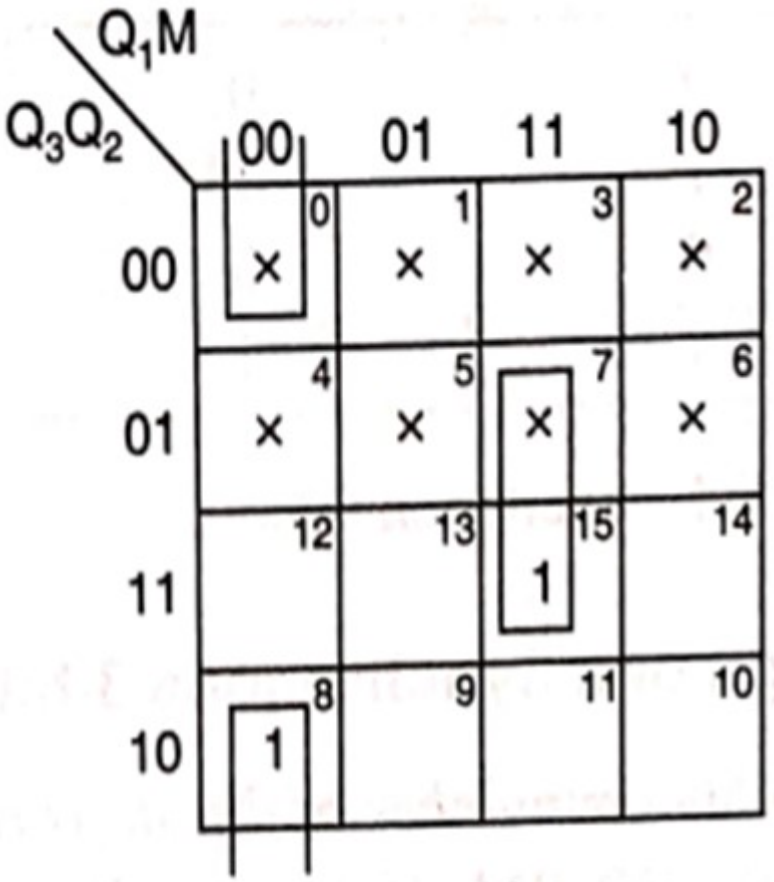
PS			Mode	NS			Required excitations					
Q_3	Q_2	Q_1	M	Q_3	Q_2	Q_1	J_3	K_3	J_2	K_2	J_1	K_1
0	0	0	0	1	1	1	1	x	1	x	1	x
0	0	0	1	0	0	1	0	x	0	x	1	x
0	0	1	0	0	0	0	0	x	0	x	x	1
0	0	1	1	0	1	0	0	x	1	x	x	1
0	1	0	0	0	0	1	0	x	x	1	1	x
0	1	0	1	0	1	1	0	x	x	0	1	x
0	1	1	0	0	1	0	0	x	x	0	x	1
0	1	1	1	1	0	0	1	x	x	1	x	1
1	0	0	0	0	1	1	x	1	1	x	1	x
1	0	0	1	1	0	1	x	0	0	x	1	x
1	0	1	0	1	0	0	x	0	0	x	x	1
1	0	1	1	1	1	0	x	0	1	x	x	1
1	1	0	0	1	0	1	x	0	x	1	1	x
1	1	0	1	1	1	1	x	0	x	0	1	x
1	1	1	0	1	1	0	x	0	x	0	x	1
1	1	1	1	0	0	0	x	1	x	1	x	1

(b) Excitation table

Minimal Expressions ($J_1=1, K_1=1$)

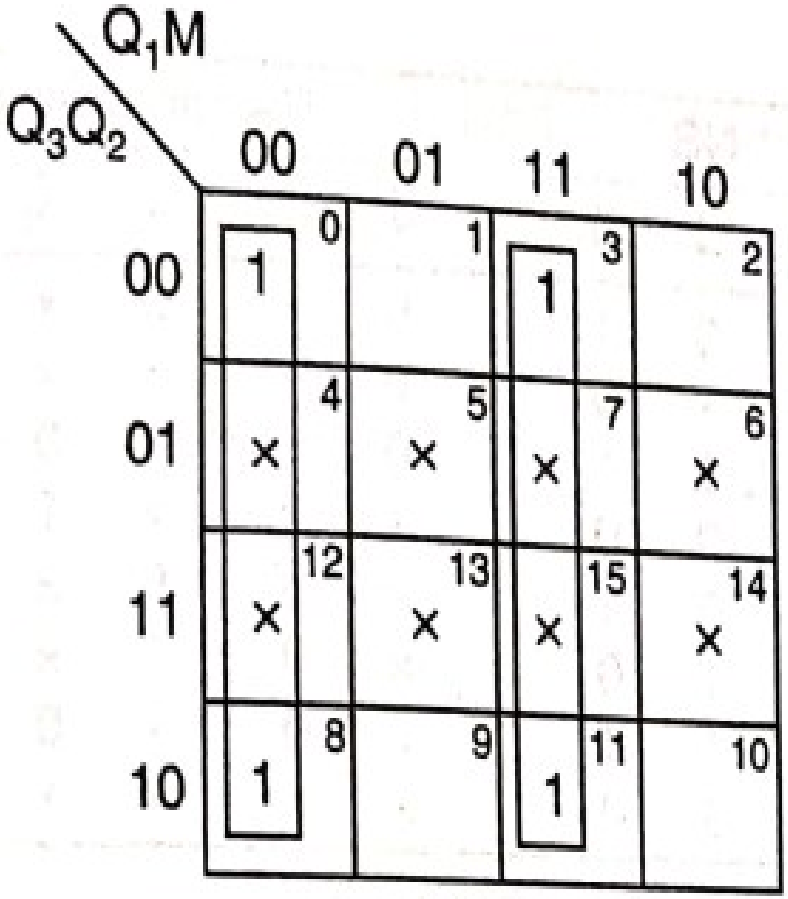


$J_3 = \bar{Q}_2\bar{Q}_1\bar{M} + Q_2Q_1M$

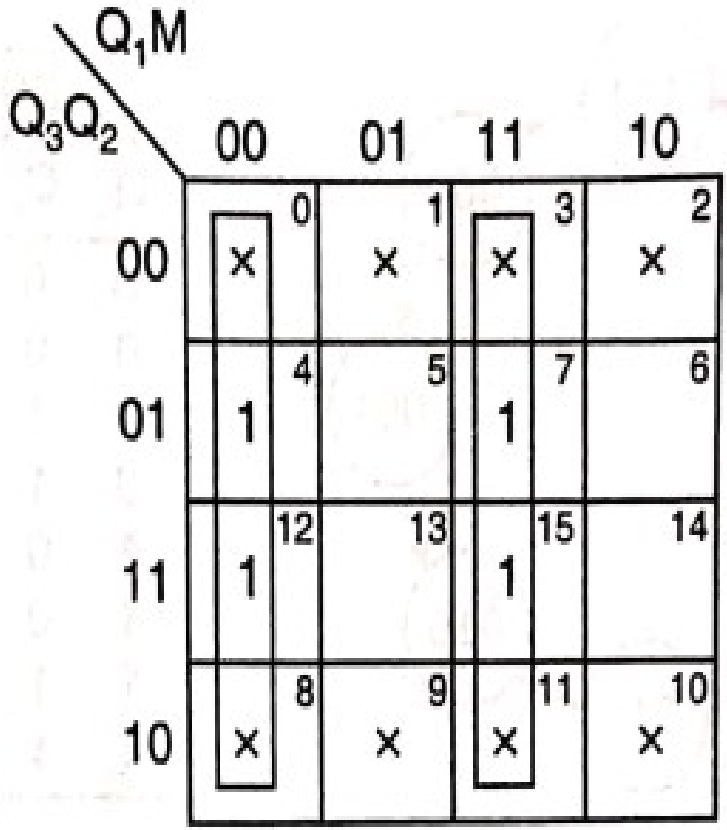


$K_3 = \bar{Q}_2\bar{Q}_1\bar{M} + Q_2Q_1M$

Minimal Expressions

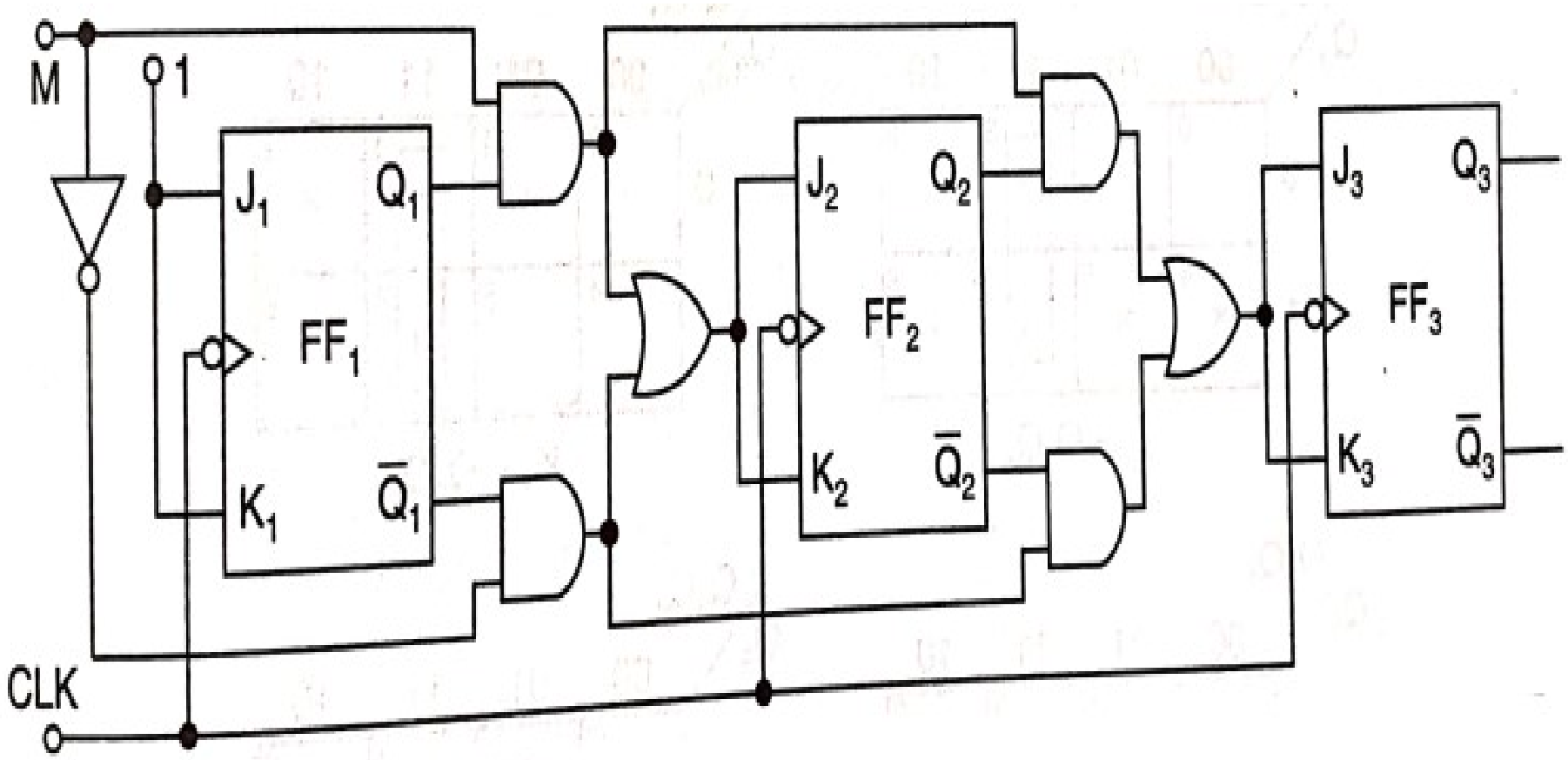


$$J_2 = \bar{Q}_1\bar{M} + Q_1M$$



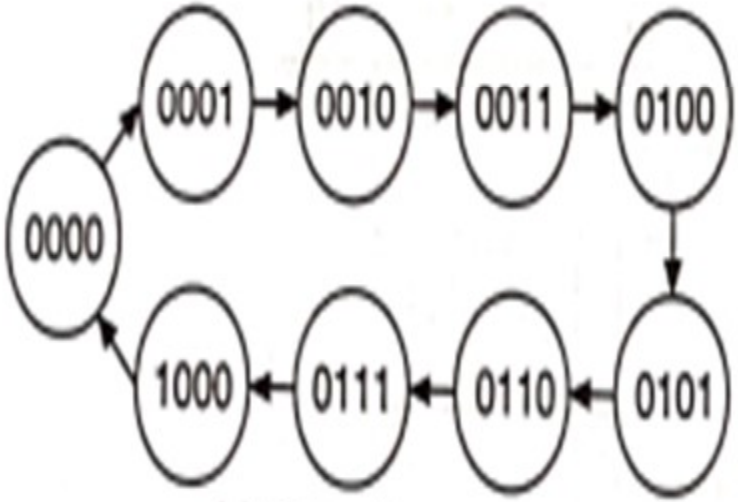
$$K_2 = \bar{Q}_1\bar{M} + Q_1M$$

Logic Diagram



Design a synchronous Modulo-9 counter using T FFs

(University Question)



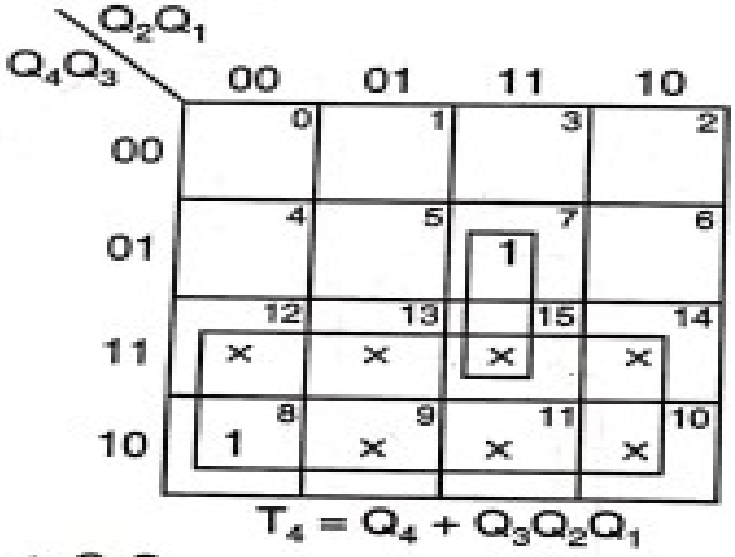
(a) State diagram

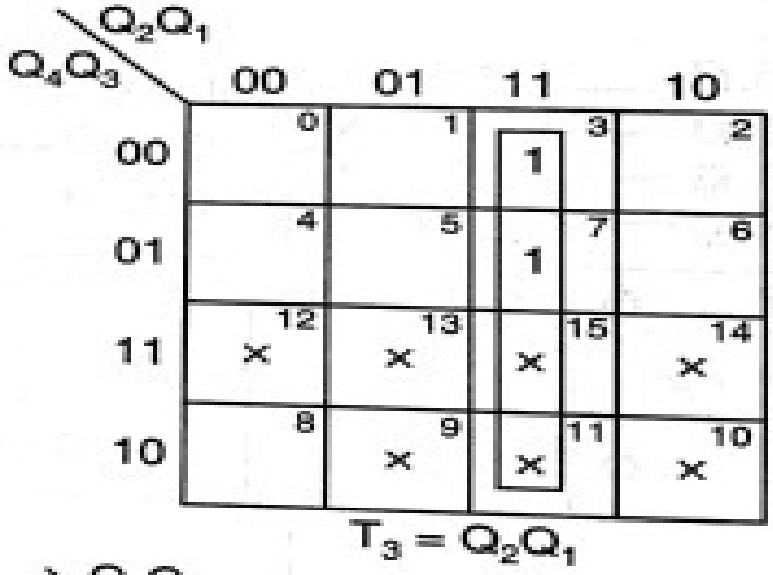
PS				NS				Required Excitations			
Q_4	Q_3	Q_2	Q_1	Q_4	Q_3	Q_2	Q_1	T_4	T_3	T_2	T_1
0	0	0	0	0	0	0	1	0	0	0	1
0	0	0	1	0	0	1	0	0	0	1	1
0	0	1	0	0	0	1	1	0	0	0	1
0	0	1	1	0	1	0	0	0	1	1	1
0	1	0	0	0	1	0	1	0	0	0	1
0	1	0	1	0	1	1	0	0	0	1	1
0	1	1	0	0	1	1	1	0	0	0	1
0	1	1	1	1	0	0	0	1	1	1	1
1	0	0	0	0	0	0	0	1	0	0	0

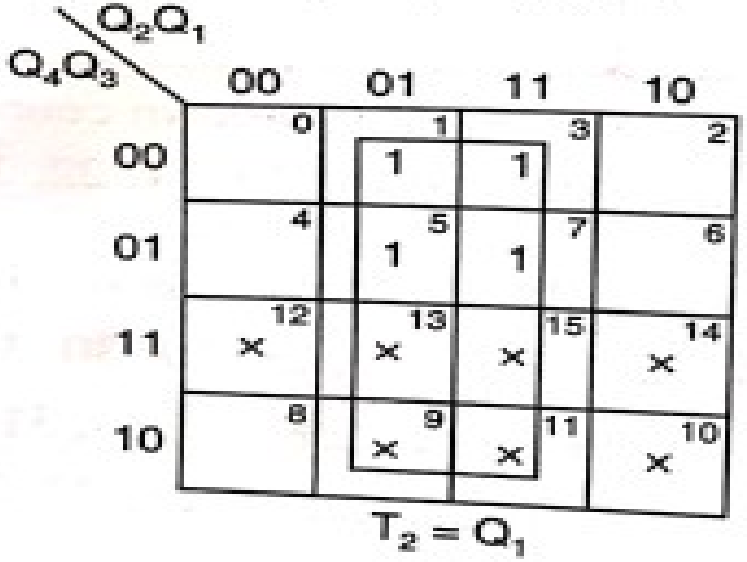
(b) Excitation table

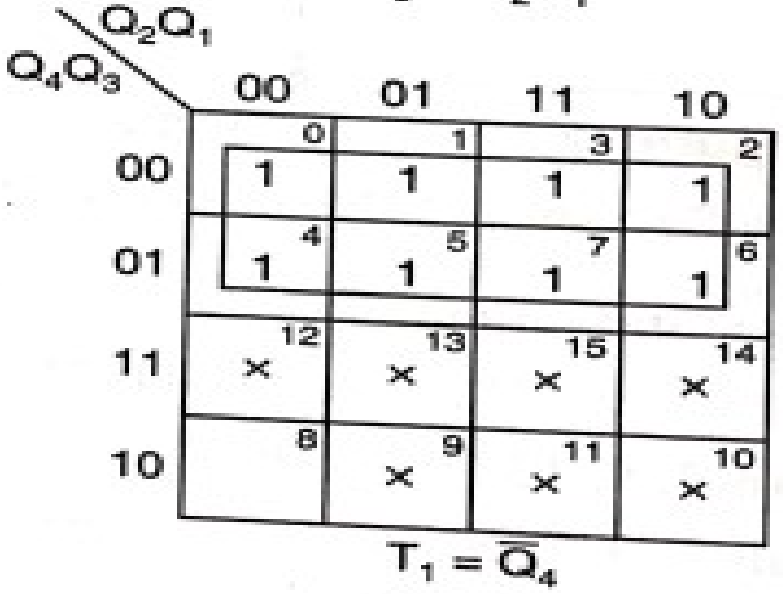
Figure 12.23 A synchronous mod-9 counter.

Minimal Expressions For T1,T2,T3,T4









Logic Diagram

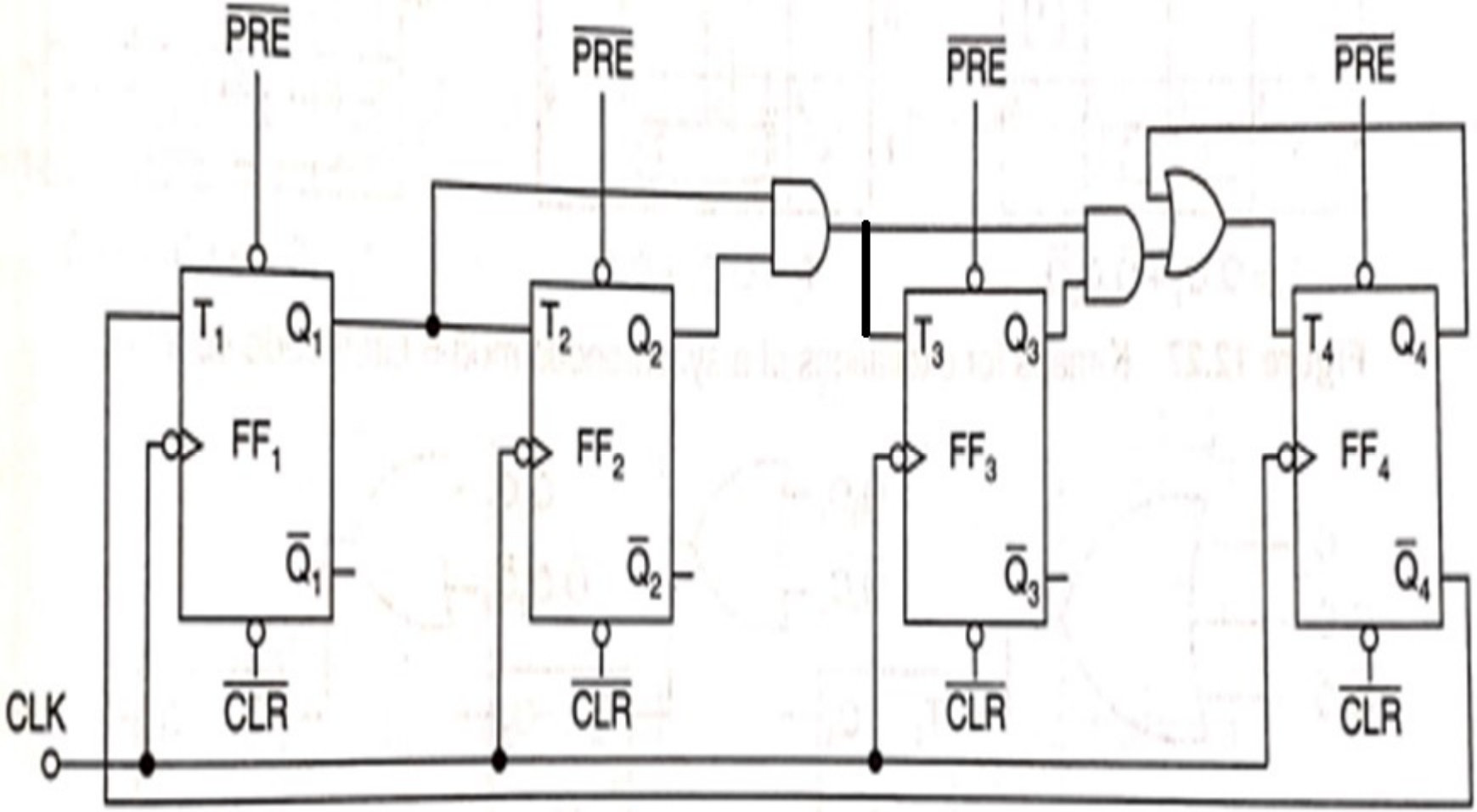
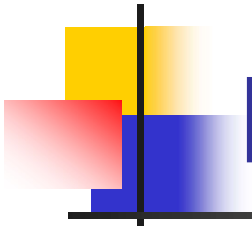


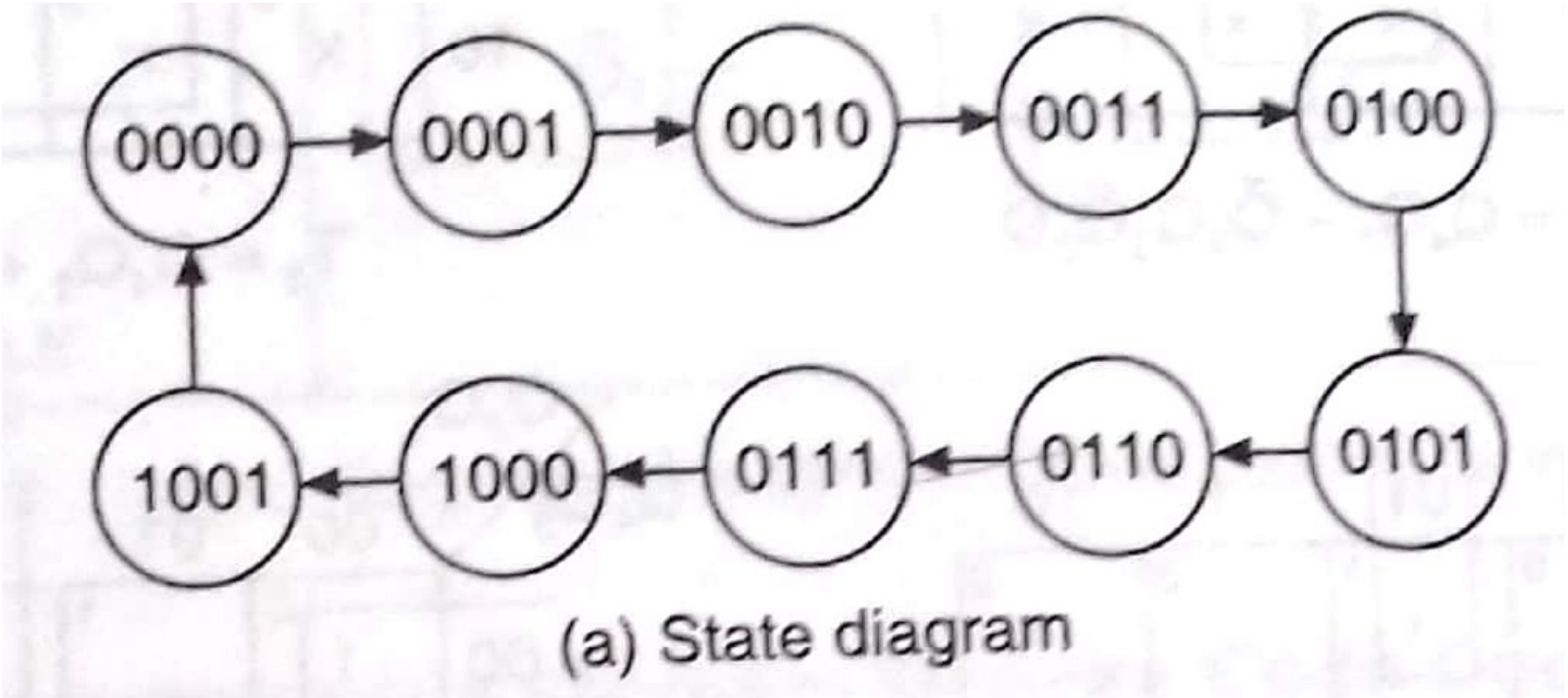
Figure 12.25 Logic diagram of the synchronous mod-9 counter using T flip-flops.



BCD Counter

- Because of the return to 0 after a count of 9, a BCD counter does not have a regular pattern as in a straight binary count.
- To derive the circuit of a BCD synchronous counter, it is necessary to go through a sequential circuit design procedure.

Design a BCD counter using JK Flip flop (University Question)



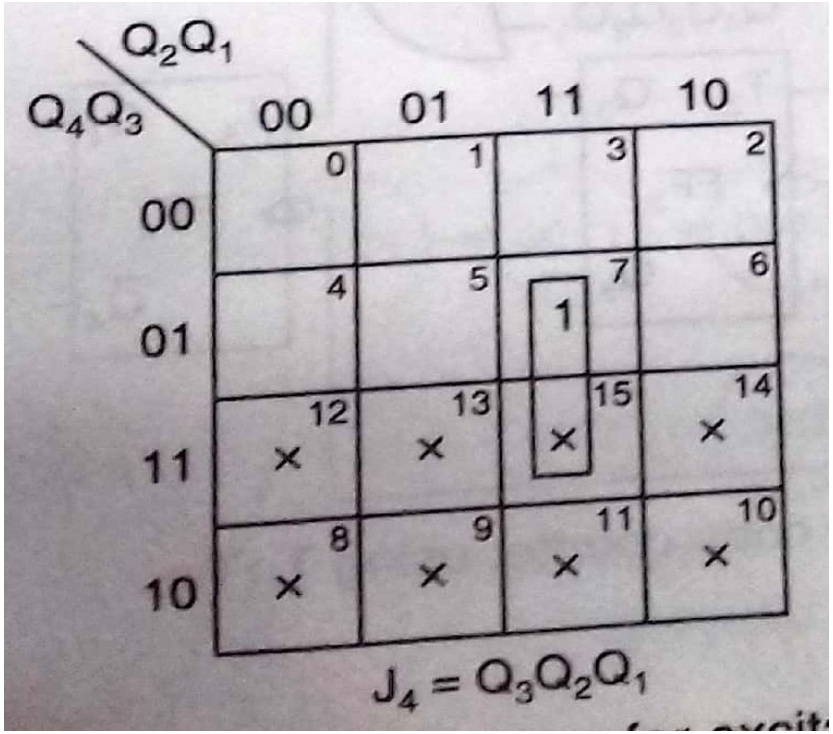
(a) State diagram

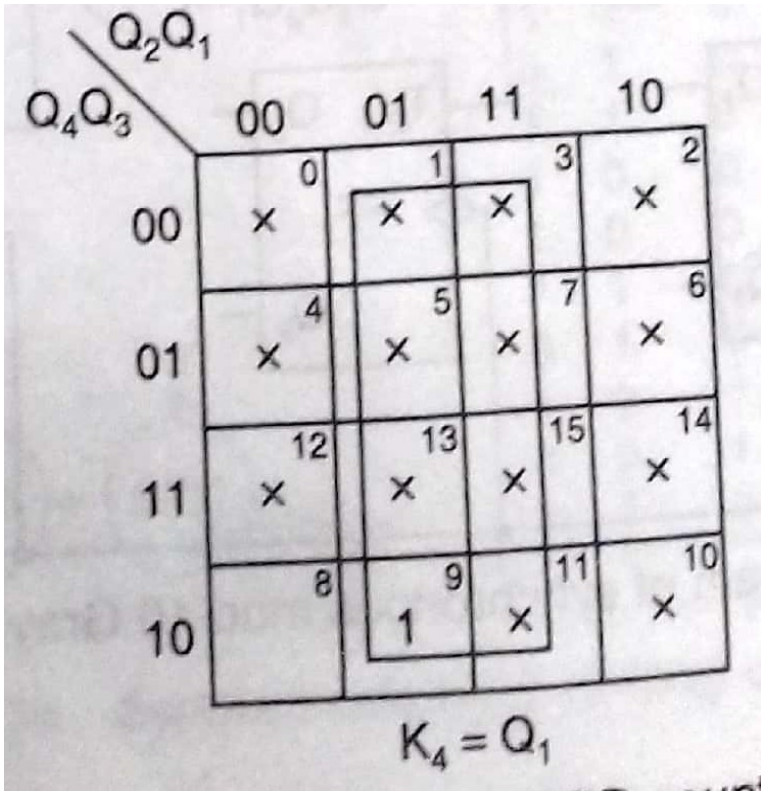
Excitation Table

PS				NS				Required excitations							
Q_4	Q_3	Q_2	Q_1	Q_4	Q_3	Q_2	Q_1	J_4	K_4	J_3	K_3	J_2	K_2	J_1	K_1
0	0	0	0	0	0	0	1	0	×	0	×	0	×	1	×
0	0	0	1	0	0	1	0	0	×	0	×	1	×	×	1
0	0	1	0	0	0	1	1	0	×	0	×	×	0	1	×
0	0	1	1	0	1	0	0	0	×	1	×	×	1	×	1
0	1	0	0	0	1	0	1	0	×	×	0	0	×	1	×
0	1	0	1	0	1	1	0	0	×	×	0	1	×	×	1
0	1	1	0	0	1	1	1	0	×	×	0	×	0	1	×
0	1	1	1	1	0	0	0	1	×	×	1	×	1	×	1
1	0	0	0	1	0	0	1	×	0	0	×	0	×	1	×
1	0	0	1	0	0	0	0	×	1	0	×	0	×	×	1

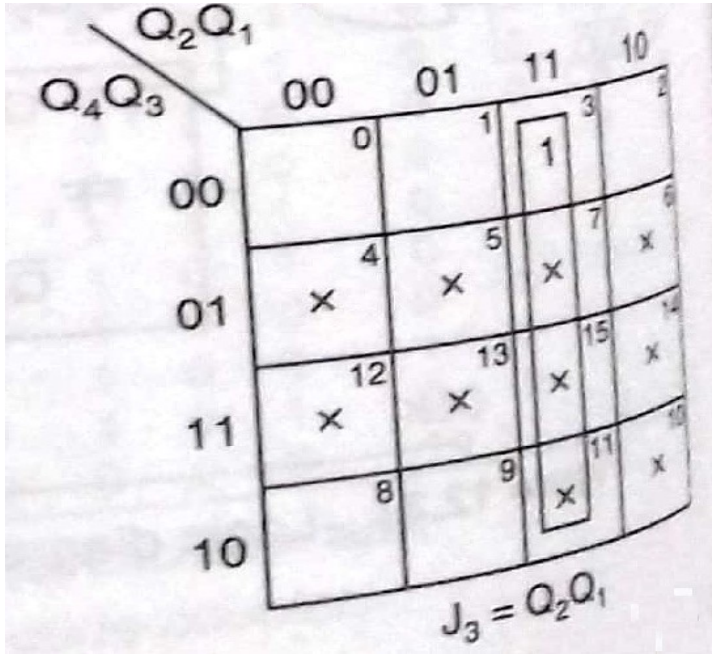
(b) Excitation table

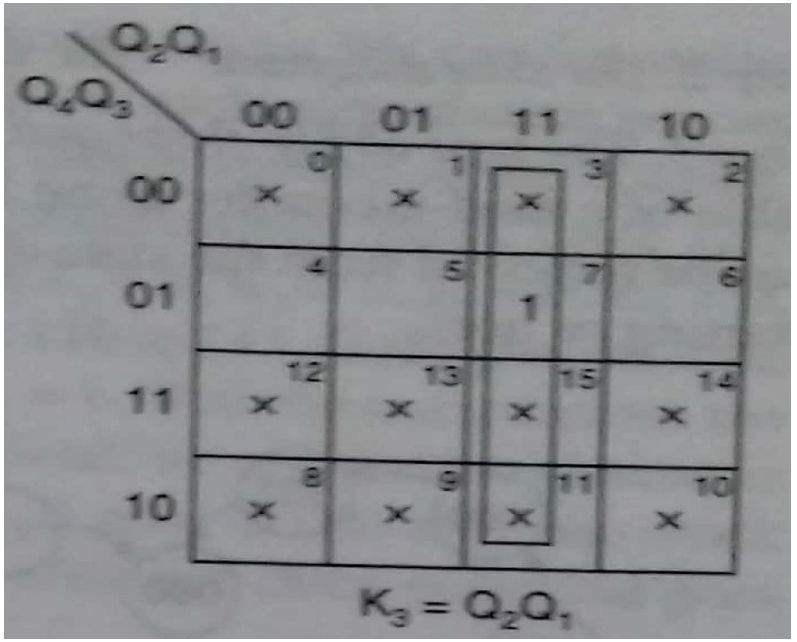
Minimal Expressions



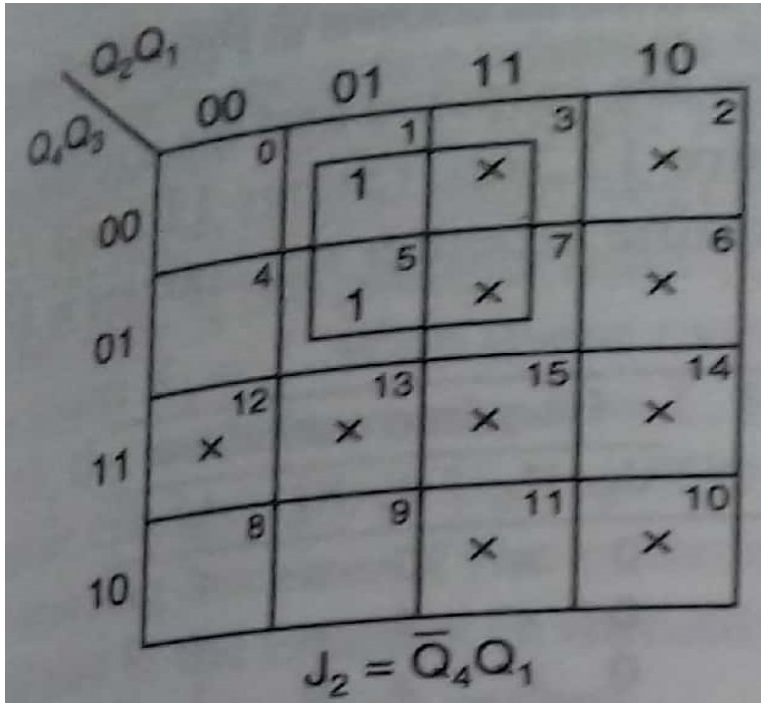


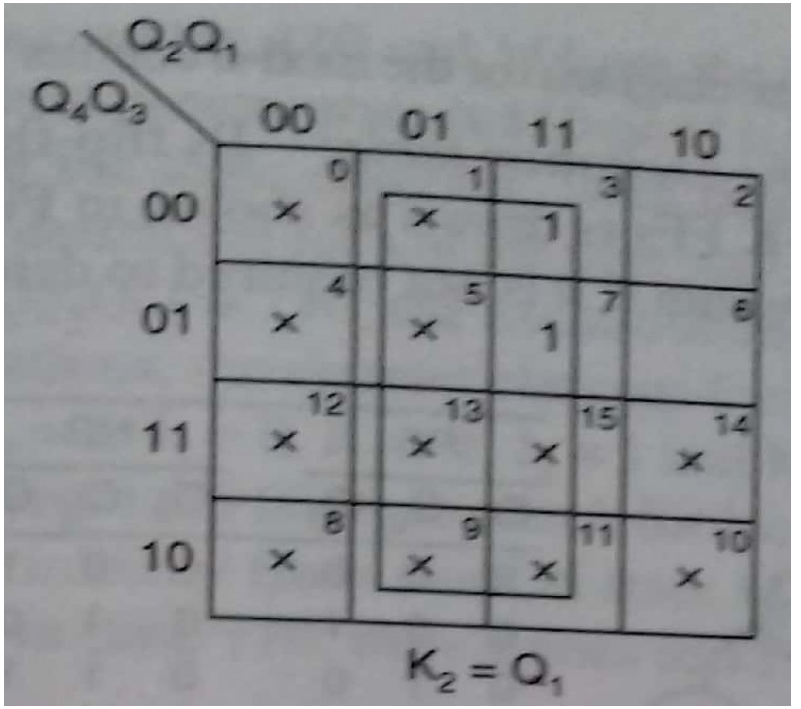
Minimal Expressions





Minimal Expressions





Logic Diagram

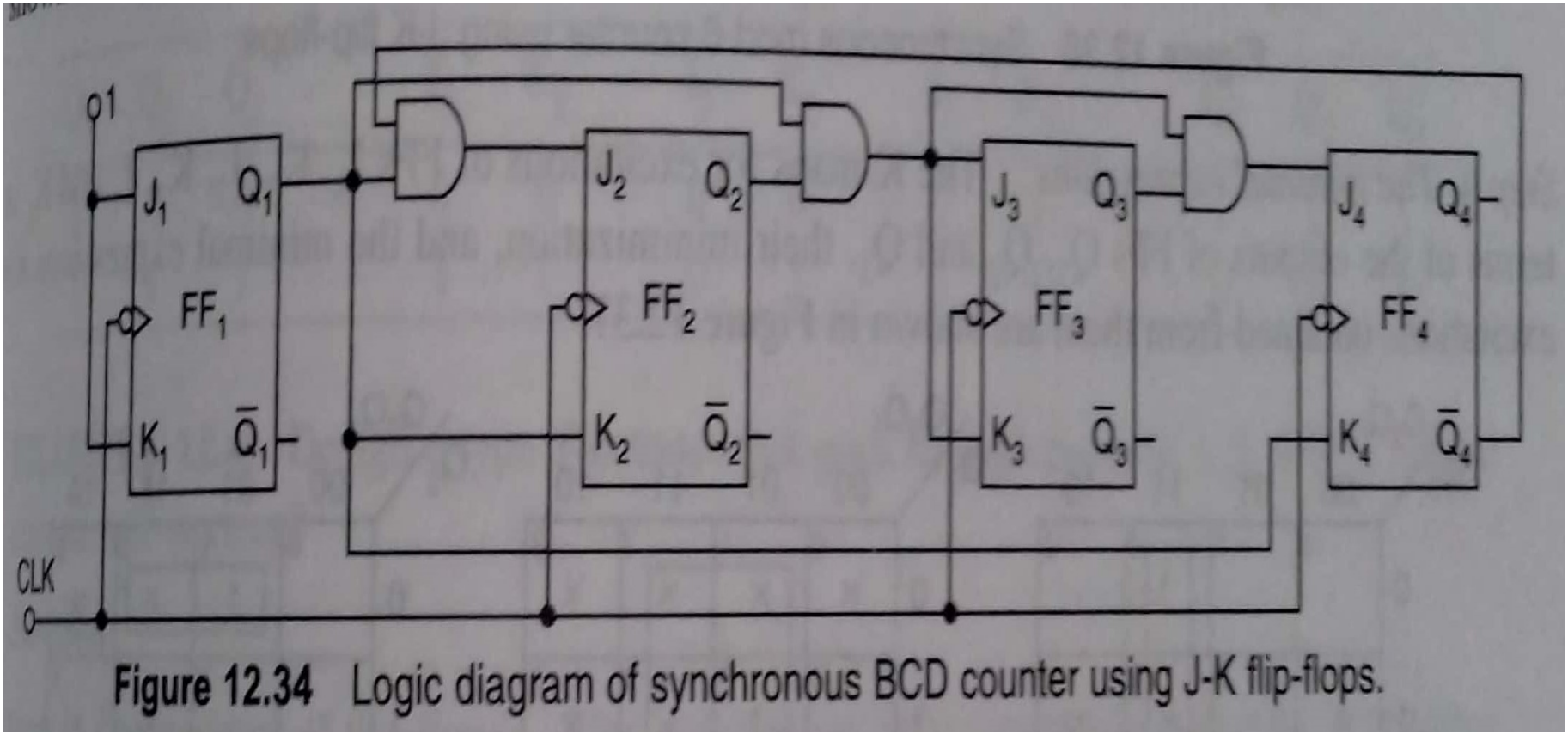
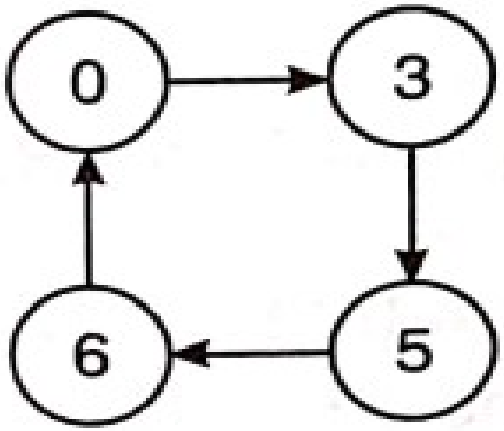


Figure 12.34 Logic diagram of synchronous BCD counter using J-K flip-flops.

Design a type T counter that goes through states 0,3,5,6,0… (University Question – 10 Marks)

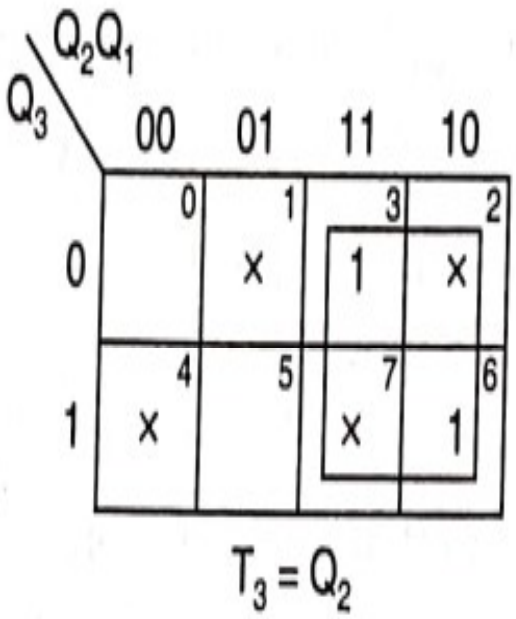


(a) State diagram

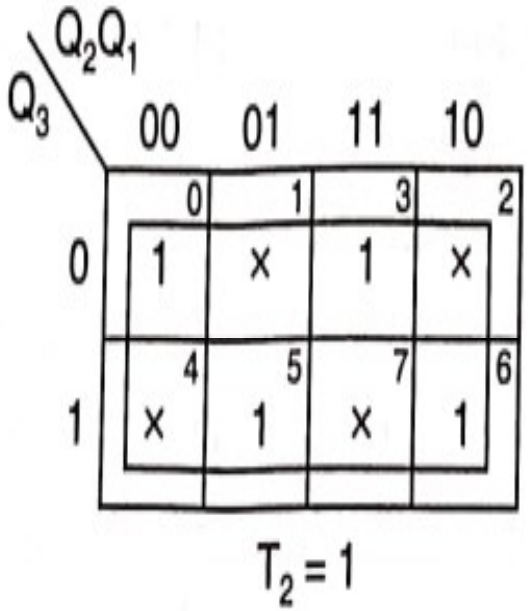
PS			NS			Required excitations		
Q_3	Q_2	Q_1	Q_3	Q_2	Q_1	T_3	T_2	T_1
0	0	0	0	1	1	0	1	1
0	1	1	1	0	1	1	1	0
1	0	1	1	1	0	0	1	1
1	1	0	0	0	0	1	1	0

(b) Excitation table

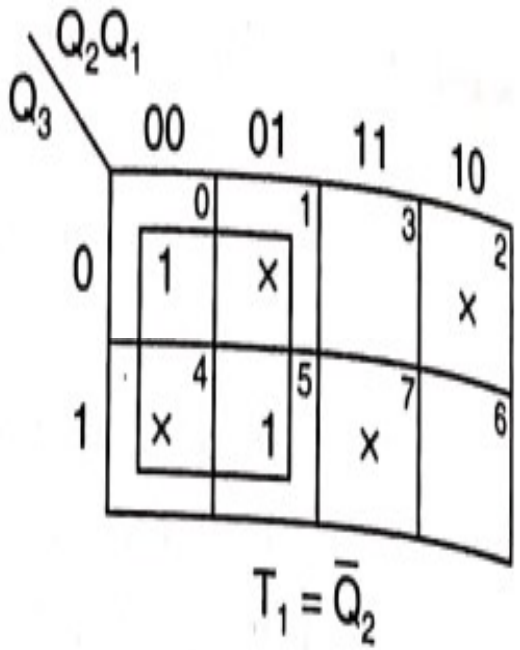
Minimized Expressions



$T_3 = Q_2$

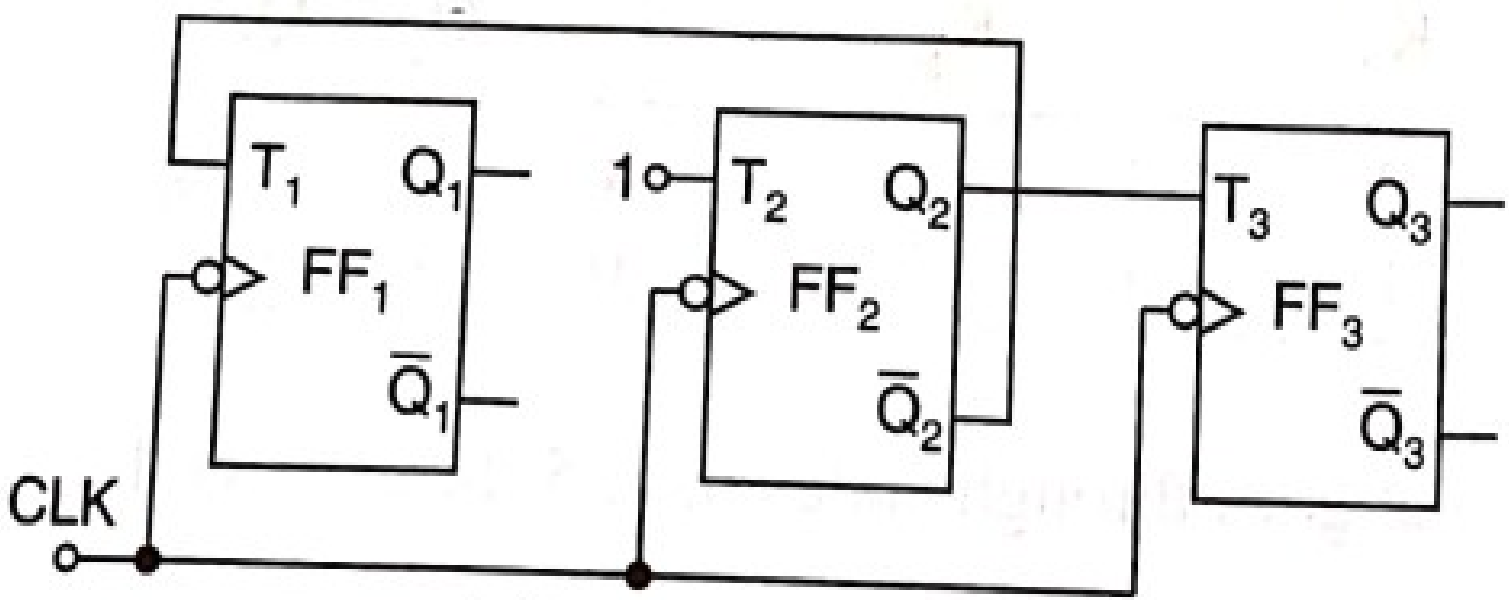


$T_2 = 1$



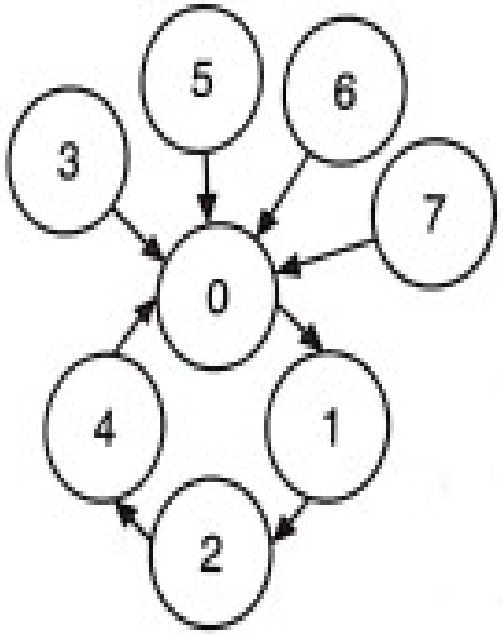
$T_1 = \bar{Q}_2$

Logic Diagram



(a) Logic diagram

Design a type D counter that goes through states 0, 1, 2, 4, 0… The undesired (unused) states must always goes to zero (000) on the next pulse (University Question – 10 Marks)



(a) State diagram

PS			NS			Required excitations		
Q_3	Q_2	Q_1	Q_3	Q_2	Q_1	D_3	D_2	D_1
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	0
0	1	0	1	0	0	1	0	0
0	1	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0

(b) Excitation table

Figure 12.45 Example 12.5: 0, 1, 2, 4, 0, ... counter.

Minimized Expressions (Use present state for deriving Kmap)

$$D_3 = \bar{Q}_3 Q_2 \bar{Q}_1; \quad D_2 = \bar{Q}_3 \bar{Q}_2 Q_1; \quad D_1 = \bar{Q}_3 \bar{Q}_2 \bar{Q}_1$$

Logic Diagram

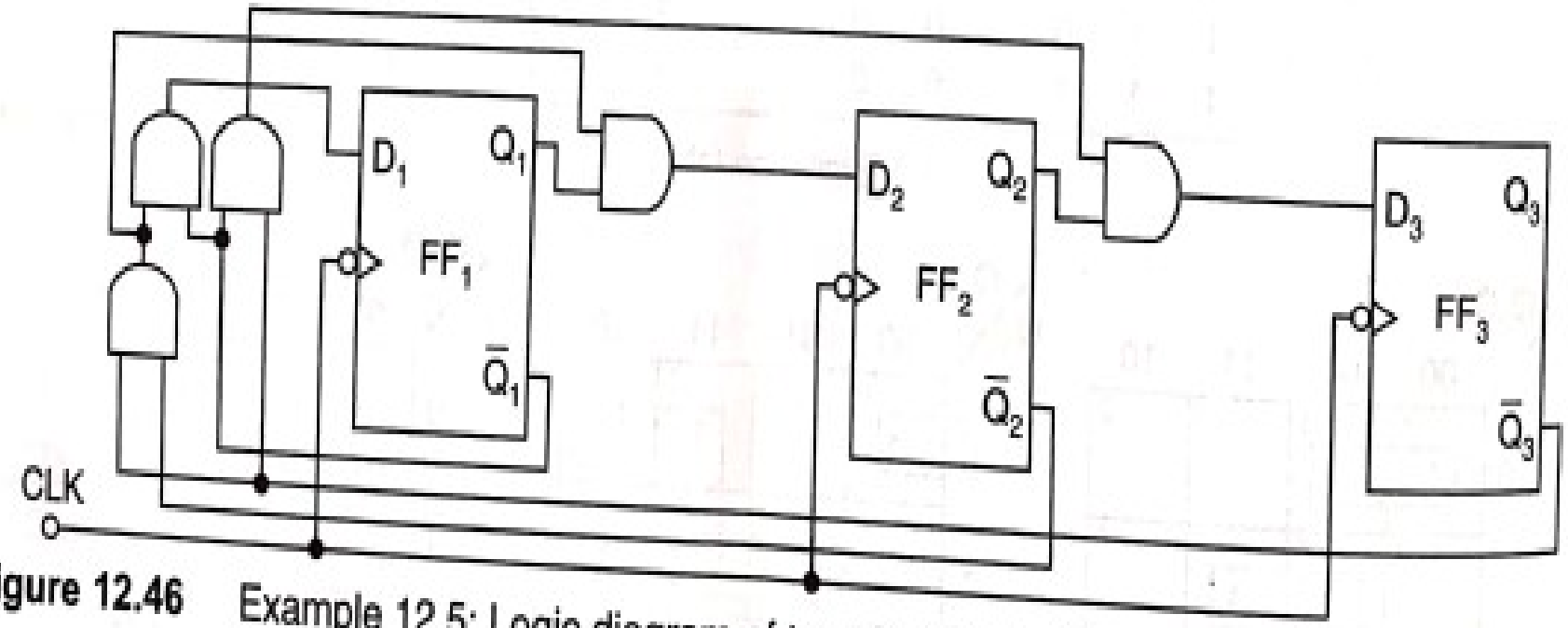


Figure 12.46 Example 12.5: Logic diagram of type D counter that goes through states 0, 1, 2, 4, 0, ...

Ring Counter

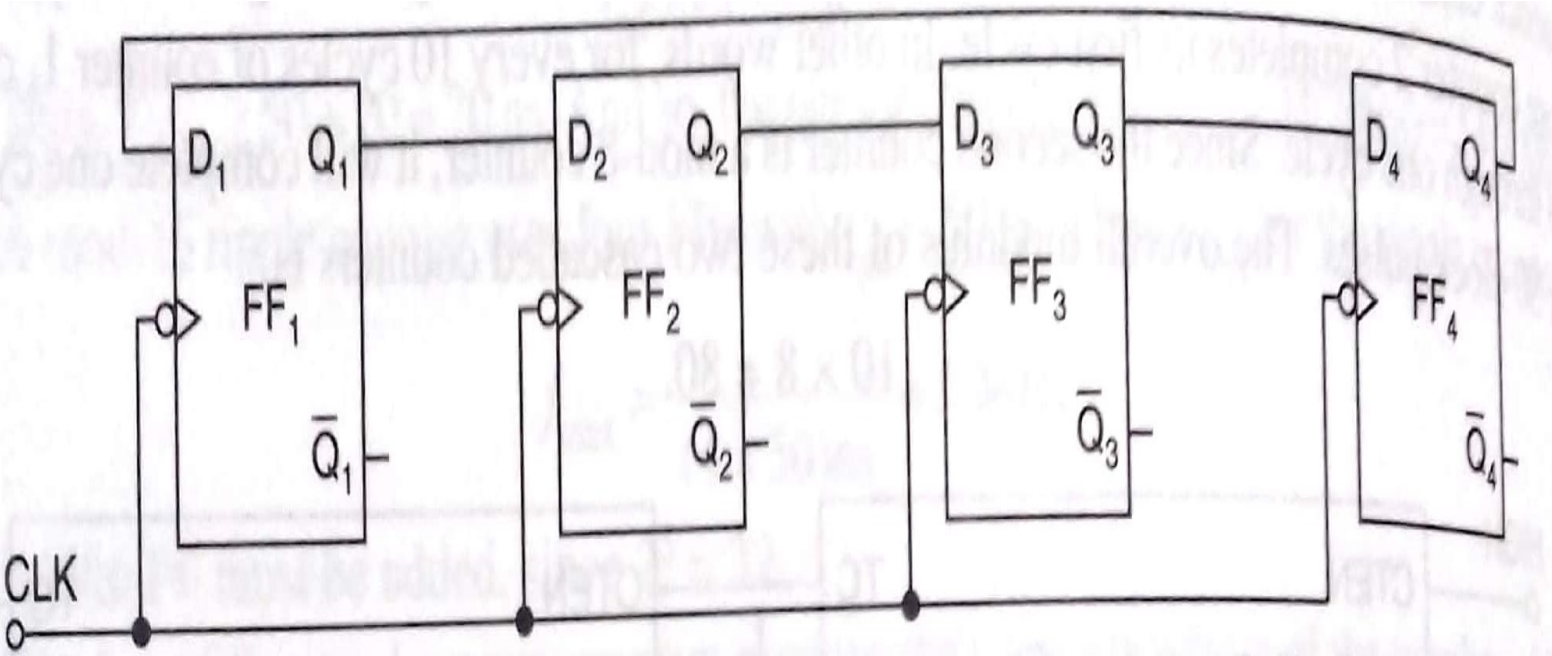


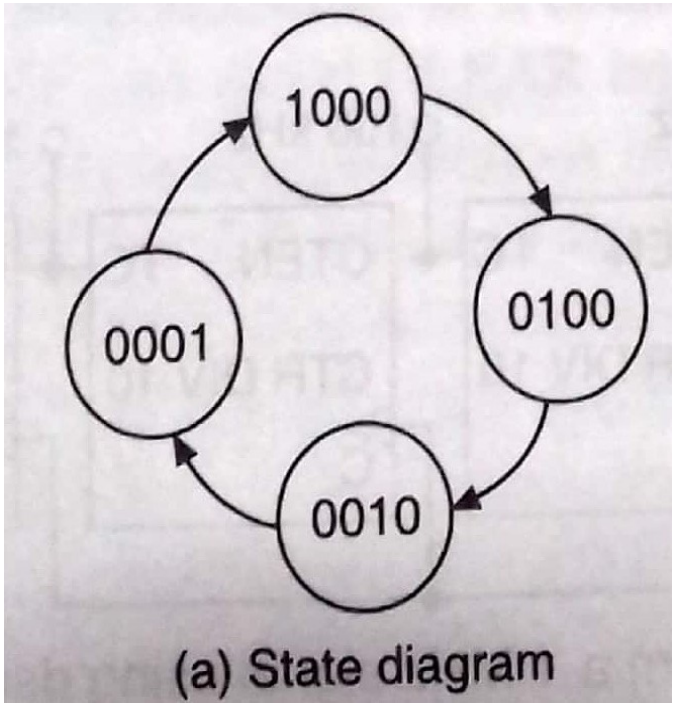
Figure 12.57 Logic diagram of a 4-bit ring counter using D flip-flops.

Logic for shifting in Ring Counter

- Q2=Q1
- Q3=Q2
- Q4=Q3

- Q1=Q4

Continued



(a) State diagram

Q_1	Q_2	Q_3	Q_4	After clock pulse
1	0	0	0	0
0	1	0	0	1
0	0	1	0	2
0	0	0	1	3
1	0	0	0	4
0	1	0	0	5
0	0	1	0	6
0	0	0	1	7

(b) Sequence table

To get initial state it requires n clock pulse. n is no. of bits
Here n=4. So after 4 clock pulse we will get initial state

Timing Diagram

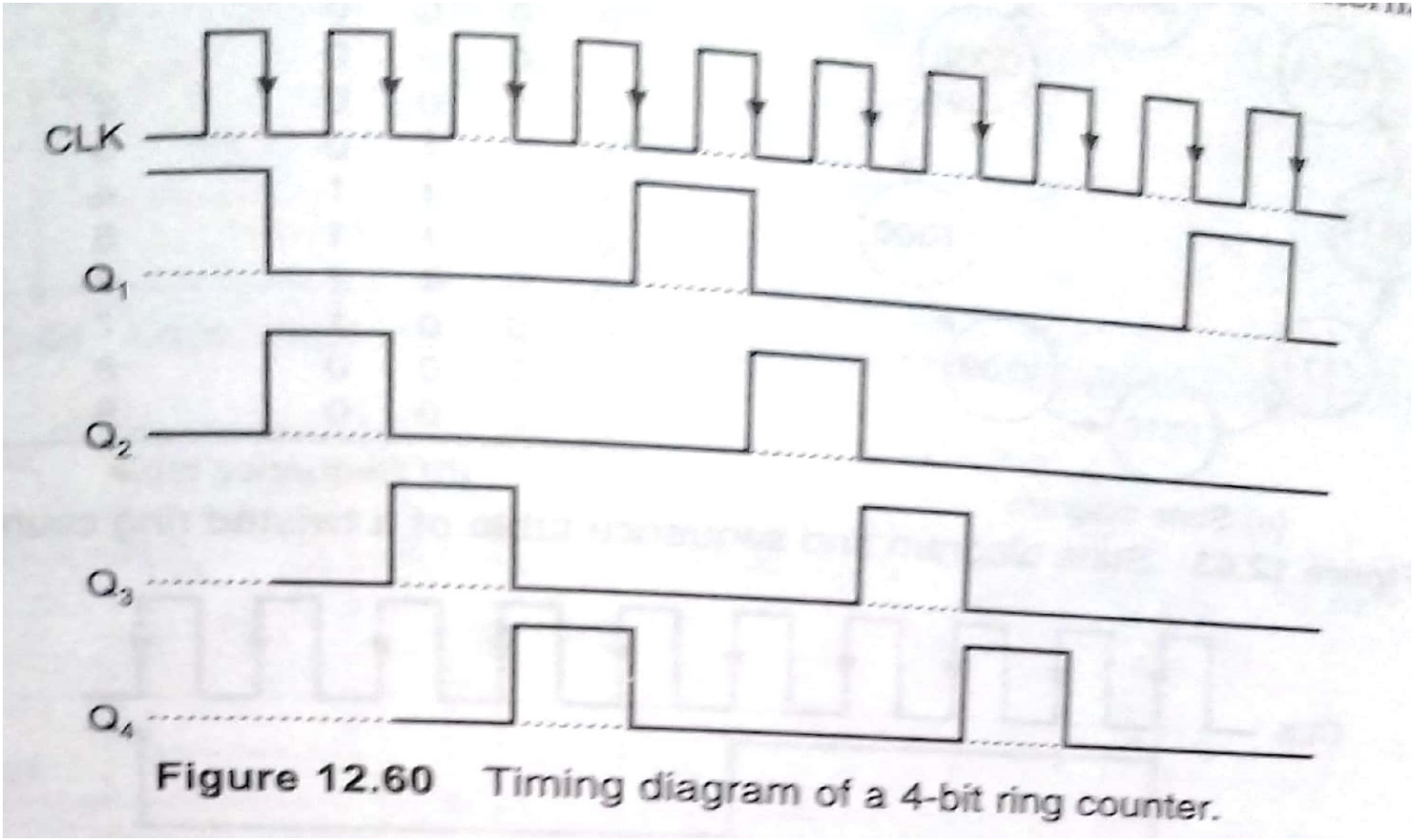


Figure 12.60 Timing diagram of a 4-bit ring counter.

Johnson Counter (Twisted Ring Counter)

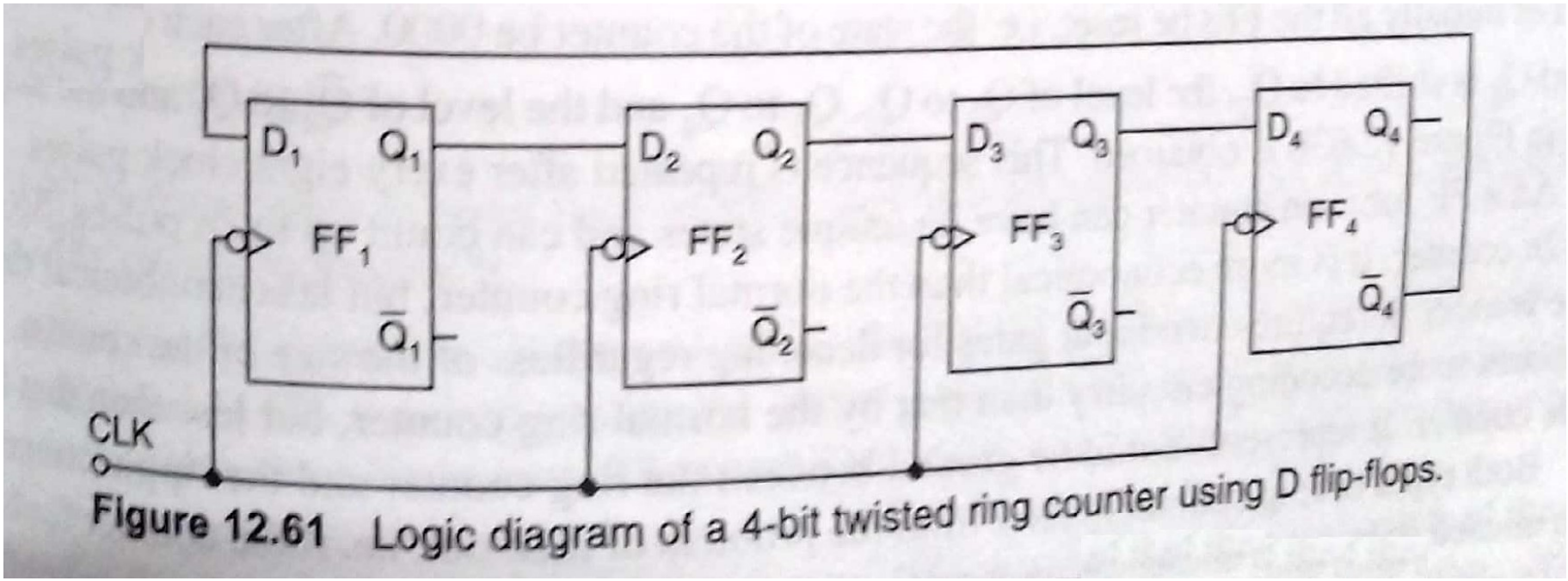


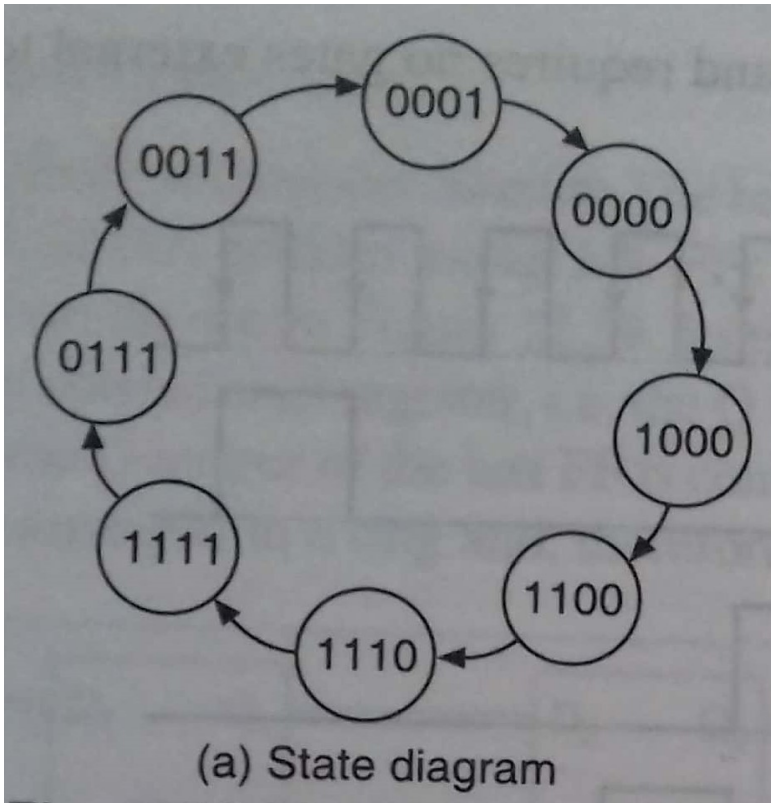
Figure 12.61 Logic diagram of a 4-bit twisted ring counter using D flip-flops.

Logic for shifting in Johnson Counter

- Q2=Q1
- Q3=Q2
- Q4=Q3

- Q1=Q4’

State Diagram and Sequence Table



(a) State diagram

Q_1	Q_2	Q_3	Q_4	After clock pulse
0	0	0	0	
1	0	0	0	0
1	1	0	0	1
1	1	1	0	2
1	1	1	1	3
0	1	1	1	4
0	0	1	1	5
0	0	0	1	6
0	0	0	0	7
1	0	0	0	8
				9

(b) Sequence table

To get initial state it requires 2n clock pulse. n is no. of bits
Here n=4. So after 8 clock pulse we will get initial state

Timing Sequence –Johnson Counter

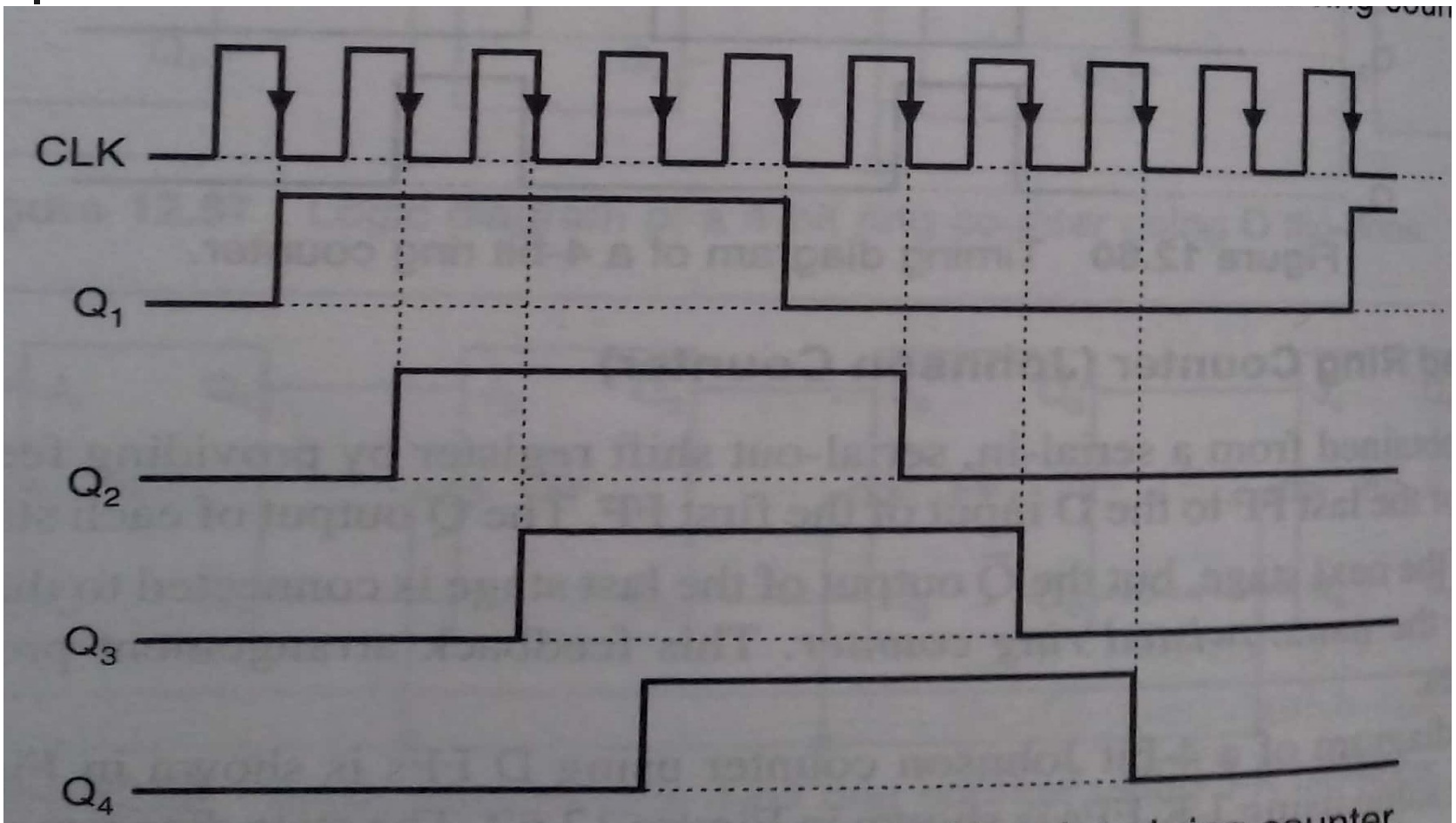


Figure 12.64 Timing diagram of a 4-bit twisted ring counter.