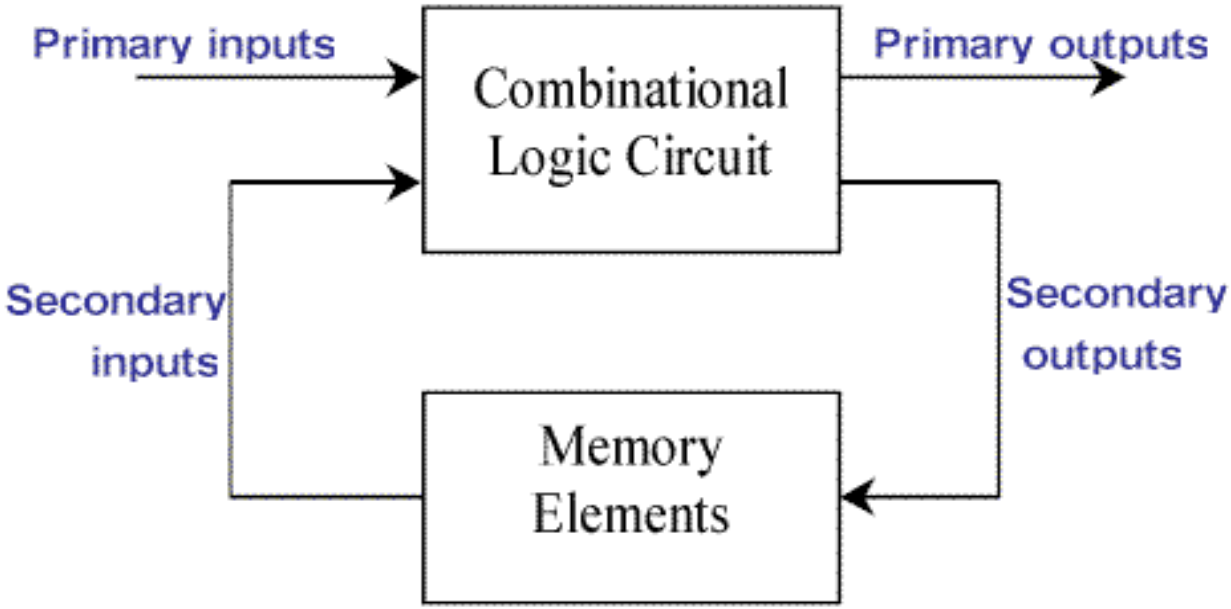
CHAPTER-5

SEQUENTIAL LOGIC CIRCUIT (LATCHES, FLIP-FLOP AND COUNTER)

INTRODUCTION

- The logic circuits whose outputs at any instant of time depend on the **present inputs** as well as on the **past outputs** are called **sequential circuits.**
- In sequential circuits, the output signals are fed back to the input side. A block diagram of a sequential circuit is shown in Figure below.



Block Diagram of Sequential Circuit

cont ……

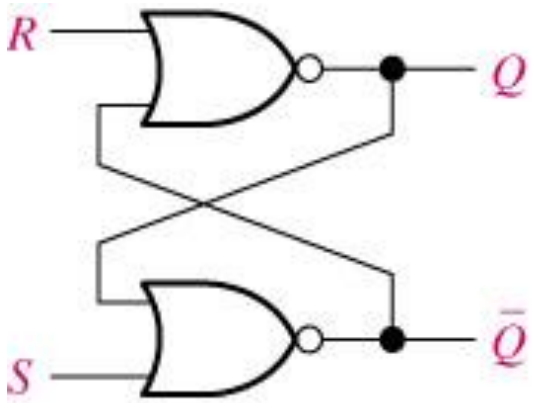
- Sequential circuits are broadly classified into two main categories, known as synchronous or clocked and asynchronous or unclocked sequential circuits, depending on the timing of their signals.
- A synchronous sequential circuit is a system whose behavior can be defined from the knowledge of its signals at discrete instants of time.
- The behavior of an asynchronous sequential circuit depends upon the order in which its input signals change and can be affected at any instant of time.
- Storage Element (memory): Latches & Flip-Flops
- Storage elements that operate with signal levels (rather than signal transitions) are referred to as Latches; those controlled by a clock transition are flip-flops.
- Latches are said to be level sensitive devices; flip-flops are edge-sensitive devices.

LATCHES

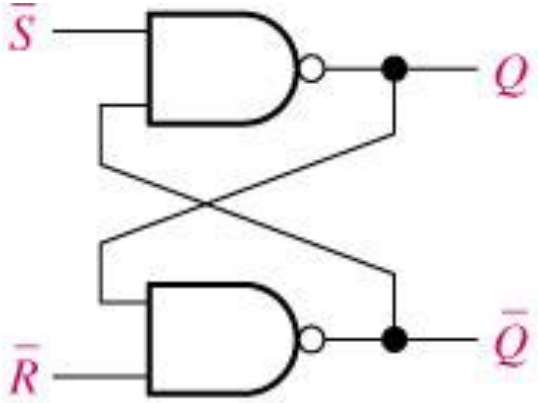
- A latch is a type of temporary storage device that has two stable states (bistable) a is normally placed in a category separate from that of flip-flops.
- Latches are similar to flip-flops because they are bistable devices that can reside in either of two states using a feedback arrangement, in which the outputs are connected back to the opposite inputs.
- The difference between latch and flip-flop is in the method used for changing their state.
- S – set; R-reset
- Type of Latch :
  - **A) S-R Latch**
  - **B) Gated S-R Latch**
  - **C) Gated D Latch**

A) S-R (SET-RESET) Latch

- A latch is a type of bistable logic device or multivibrator.
- An active-HIGH input S-R (SET- RESET) latch is formed with two cross-coupled NOR gates, as shown in Figure (a);
- An active-LOW input S-R latch is formed with two cross-coupled NAND gates, as shown in Figure (b).
- Notice that the output of each gate is connected to an input of the opposite gate.
- This produces the regenerative feedback that is characteristic of all latches and flip-flops.

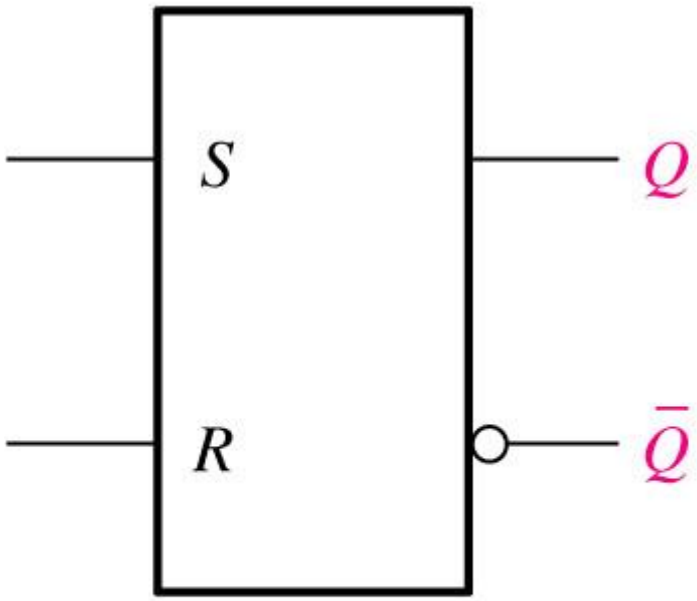


(a) Active-HIGH input S-R latch

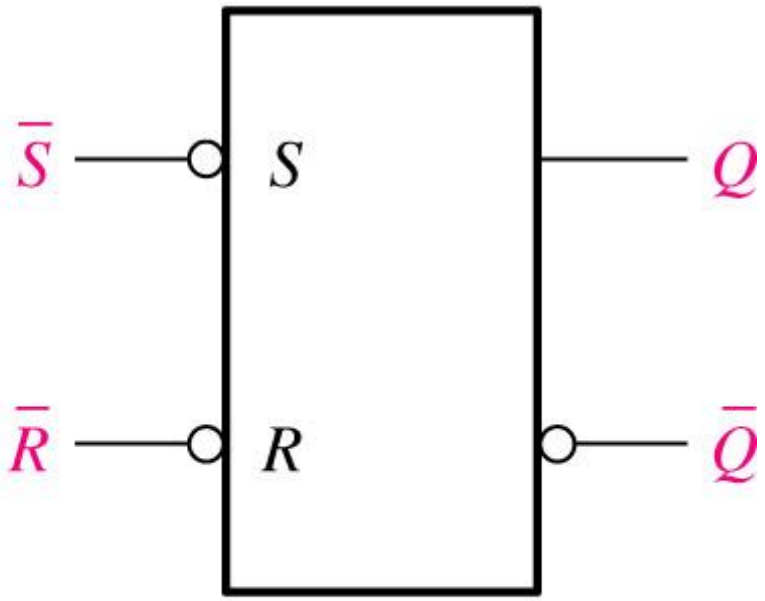


(b) Active-LOW input $\bar{S}$-$\bar{R}$ latch

Logic Symbol : S-R and $\bar{S}$-$\bar{R}$ Latch



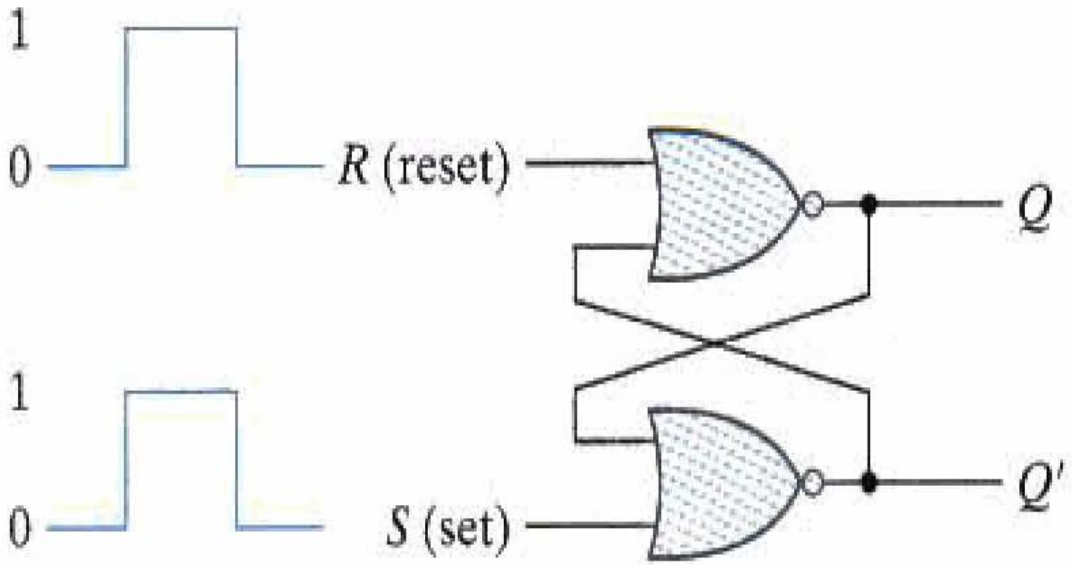
(a) Active-HIGH input S-R latch



(b) Active-LOW input $\bar{S}$-$\bar{R}$ latch

Cont-----

Truth table for an active-High (NOR gate) input S-R latch.



(a) Logic diagram

S	R	Q	Q'	
1	0	1	0	
0	0	1	0	(after $S = 1, R = 0$)
0	1	0	1	
0	0	0	1	(after $S = 0, R = 1$)
1	1	0	0	(forbidden)

(b) Function table

✓The Q and not-Q outputs are supposed to be in opposite states.

✓ **Q=1** and not-Q =0 is defined as ***set*** (by making S=1 and R=0)

✓ Q=0 and **not-Q =1** is conversely defined as ***reset*** (by making S=0 and R=1)

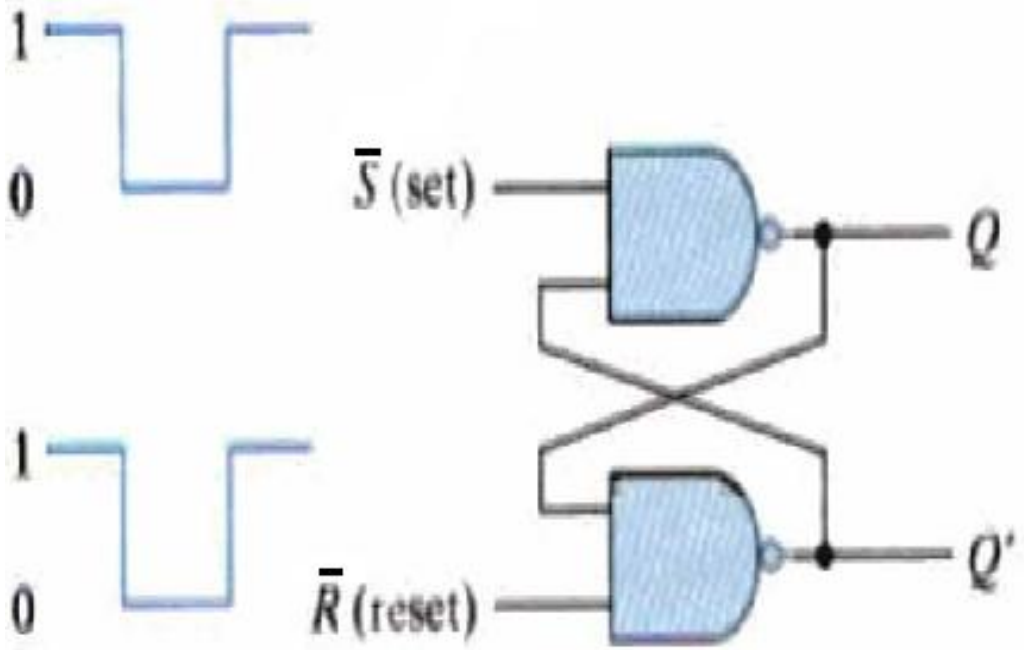
✓ When S and R are both equal to **0**, the multivibrator's outputs "not change" in their prior states.

✓ If Q and not-Q happen to be forced to the same state both 1, that state is referred to as ***invalid.***

Cont…

Truth table for an active-LOW(NAND gate) input S-R latch.

✓ The arrangement, shown in Figure below, is similar to the NOR gate latch except that the Q and Q' output save reversed positions.

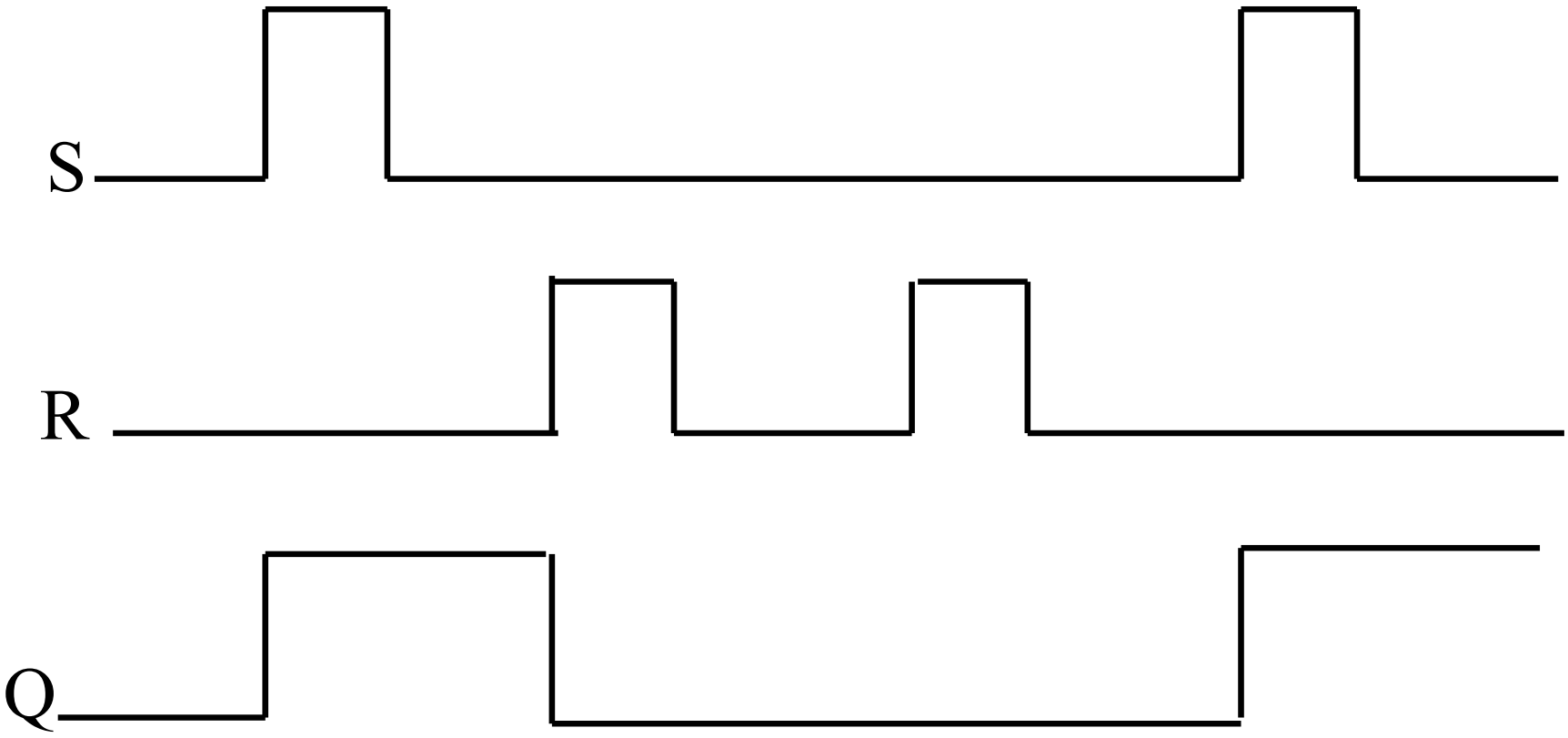


(a) Logic diagram

$\overline{S}$	$\overline{R}$	Q	$\overline{Q}$	
1	0	0	1	
1	1	0	1	(after $S = 1, R = 0$)
0	1	1	0	
1	1	1	0	(after $S = 0, R = 1$)
0	0	1	1	(forbidden)

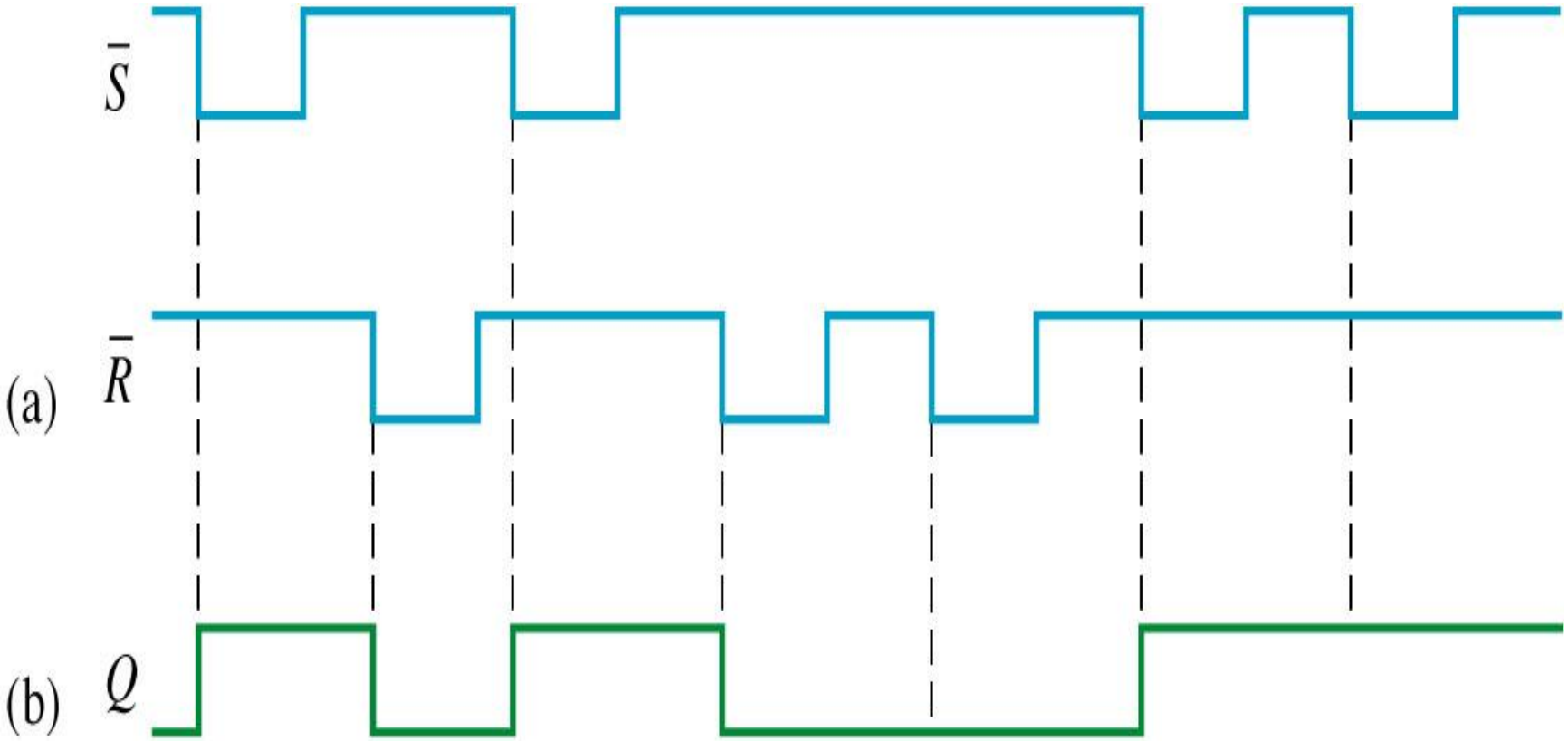
(b) Function table

Example-1:- Assume that Q = 0 initially, and determine the Q waveform for the NOR latch inputs of Figure below.



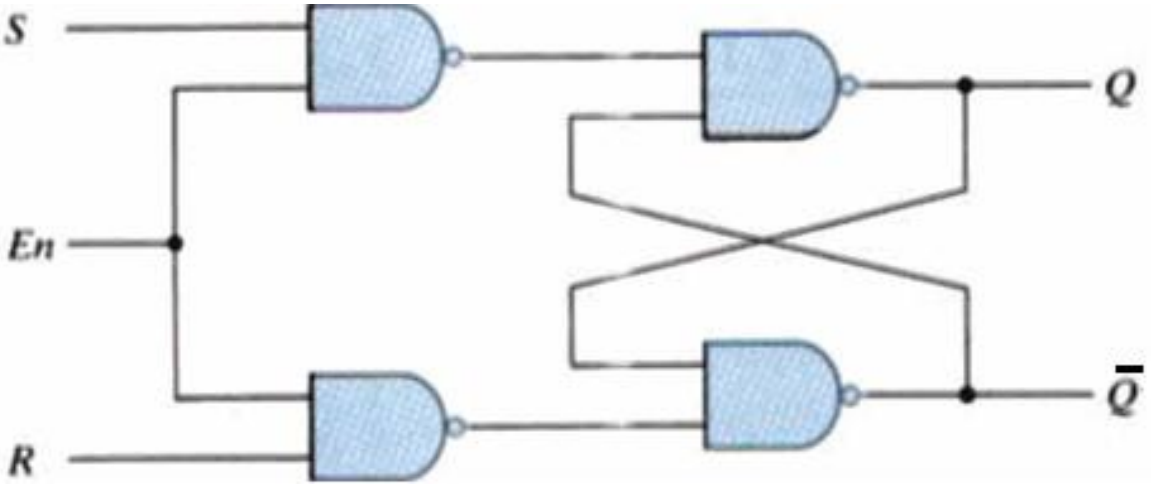
Example-2:-

The waveforms of Figure below are applied to the inputs of the NAND latch. Assume that initially Q = 0, and determine the Q waveform.

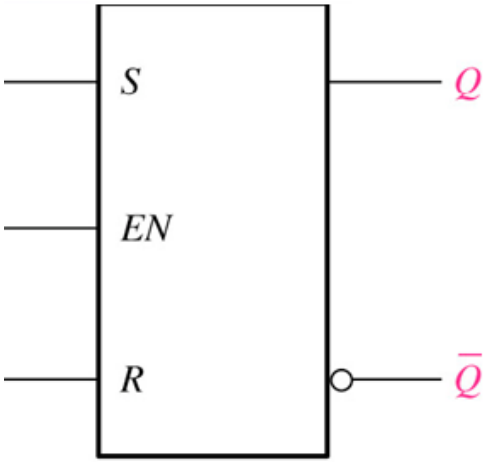


B) Gated S-R Latch

- A gated latch requires an enable input, EN .
- The S and R inputs control the state to which the latch will go when a HIGH level is applied to the EN input.
- The latch will not change until EN is HIGH; but as long as it remains HIGH, the output is controlled by the state of the S and R inputs.
- In this circuit, the invalid state occurs when both S and R are simultaneously HIGH.



(a) Logic diagram



(b) Logic symbol

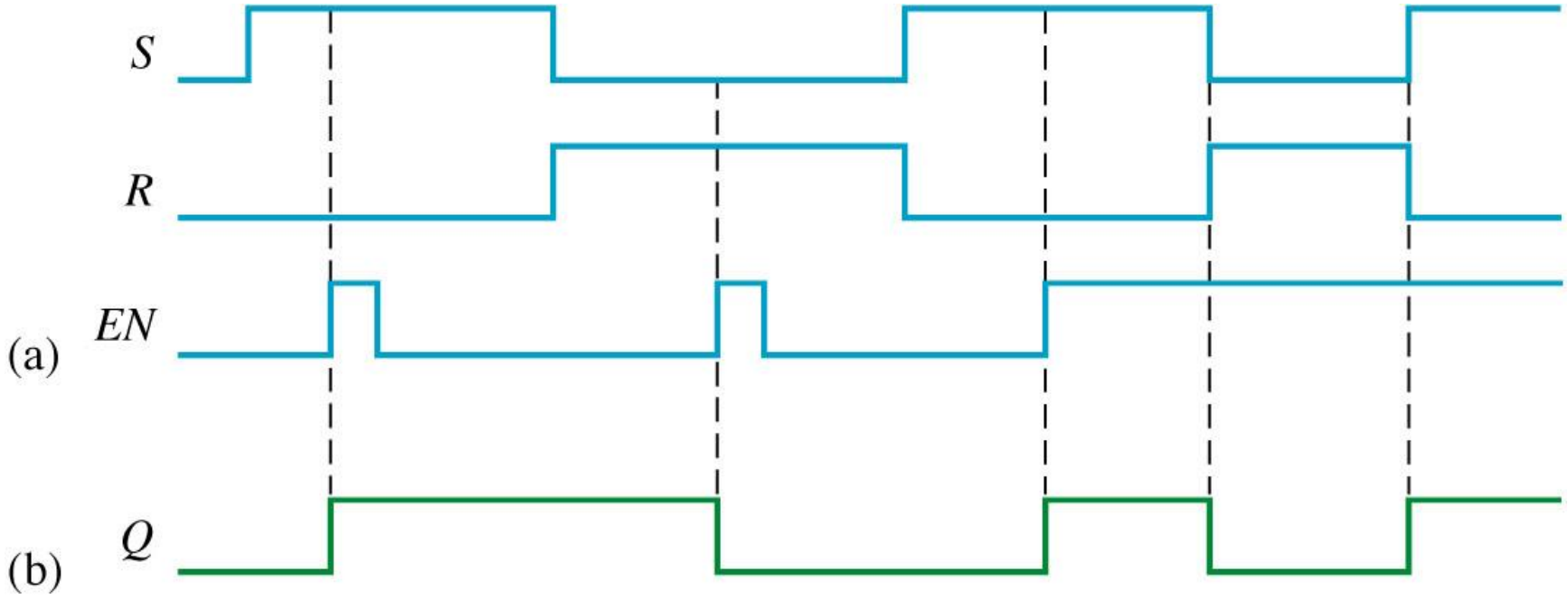
cont . . .

EN	S	R	Output
0	0	0	No Change
0	0	1	No Change
0	1	0	No Change
0	1	1	No Change
1	0	0	No Change
1	0	1	Q=0; Reset state
1	1	0	Q=1; set state
1	1	1	Invalid

(c) Truth Table

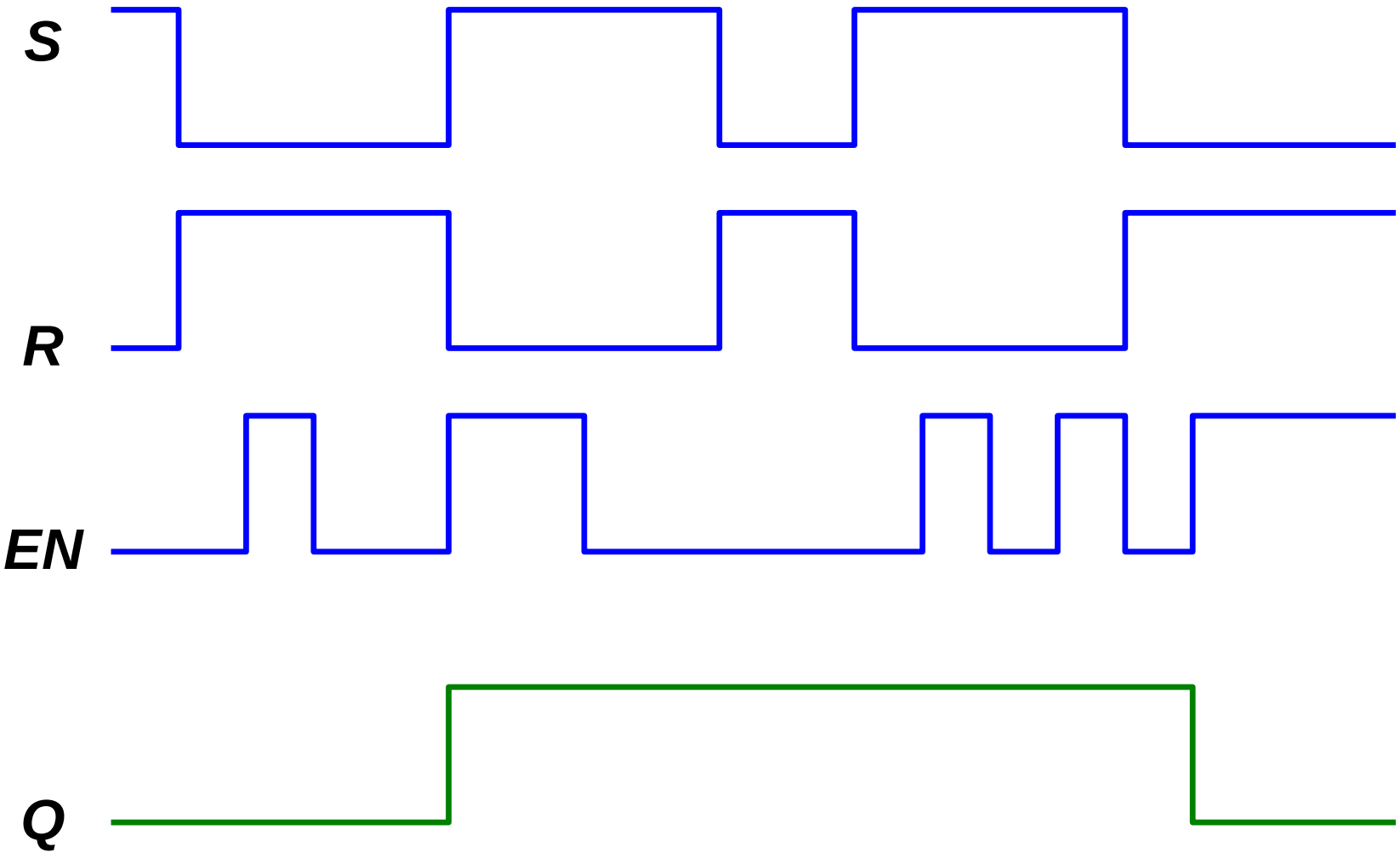
Example 1: Gated S-R Latch

Determine the Q output waveform if the inputs shown in Figure below are applied to a gated S-R latch that is initially RESET.



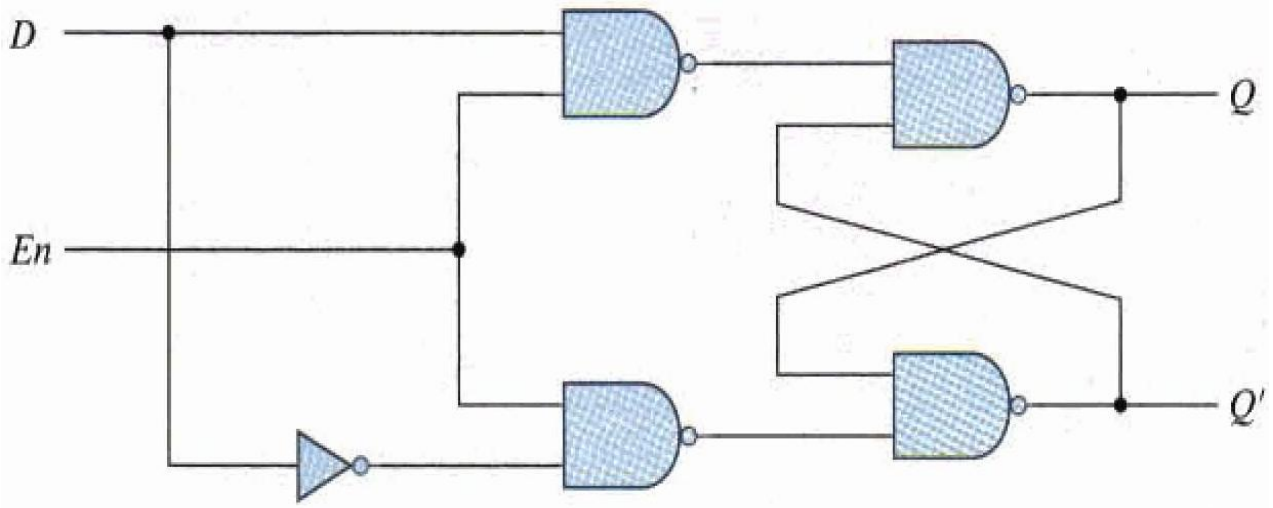
Example 2 : Gated S-R Latch.

Find waveform for Q.

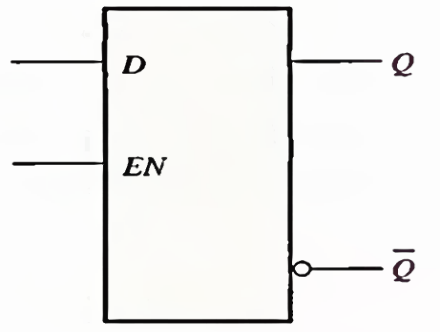


C) Gated D Latch

- Another type of gated latch is called the D latch.
- It differs from the S-R latch because it has only one input in addition to EN.
- This input is called the D (data) input.
- When the D input is HIGH and the EN input is HIGH, the latch will set.
- When the D input is LOW and EN is HIGH, the latch will reset. Stated another way, the output Q follows the input D when EN is HIGH.



(a) Logic diagram

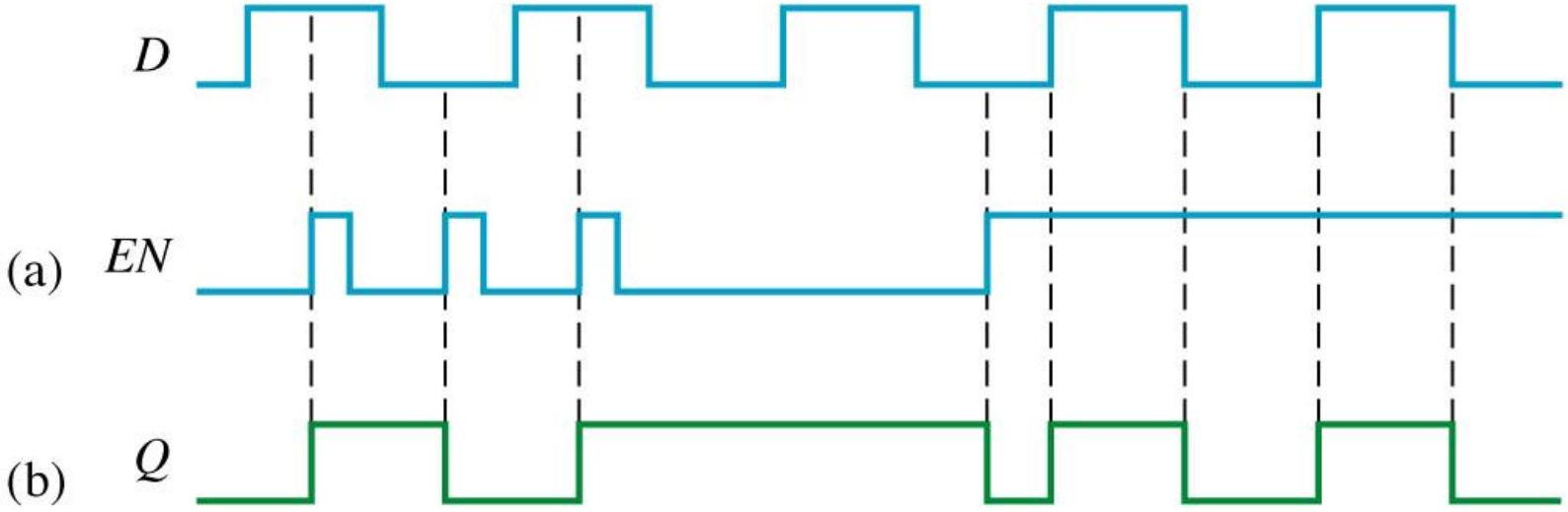


(b) Logic symbol

(c) Truth Table

EN	D	Output
0	0	No Change
0	1	No Change
1	0	Q = 0
1	1	Q = 1

Example:-Determine the Q output waveform if the inputs shown in Figure below are applied to a gated D latch, which is initially RESET.

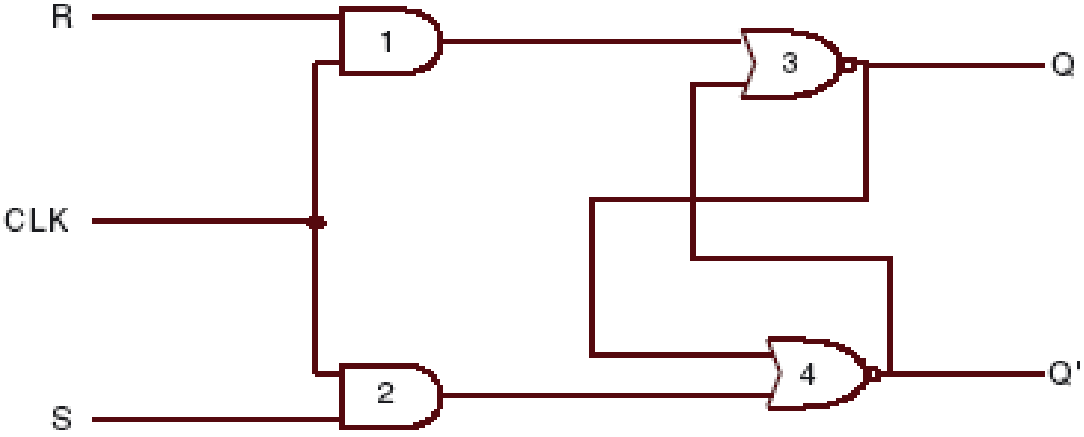


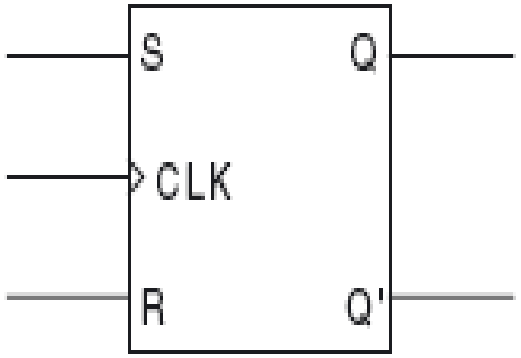
FLIP - FLOP (FF)

- The main difference between latches and flip-flops is the method used to change their states.
- **Flip-flops** are **edge-triggered**, that is that they depend on the transition of a signal.
- This may either be a LOW-to-HIGH (rising edge) or a HIGH-to-LOW (falling edge) transition.
- There are different types of flip-flops depending on how their inputs and clock pulses cause transition between two states.

1. Edge – triggered S-R (set-reset) Flip-flop
2. Edge – triggered D flip-flop (direct FF)
3. Edge – triggered J-K flip flop
4. Edge – triggered T Flip flop (Toggle FF)

1. S-R Flip flop





S	R	CLK	Output		Comments
			Q	Q’	
0	0	X	Q_0	Q'_0	No Change
0	1	↑	0	1	RESET
1	0	↑	1	0	SET
1	1	↑	?	?	Invalid

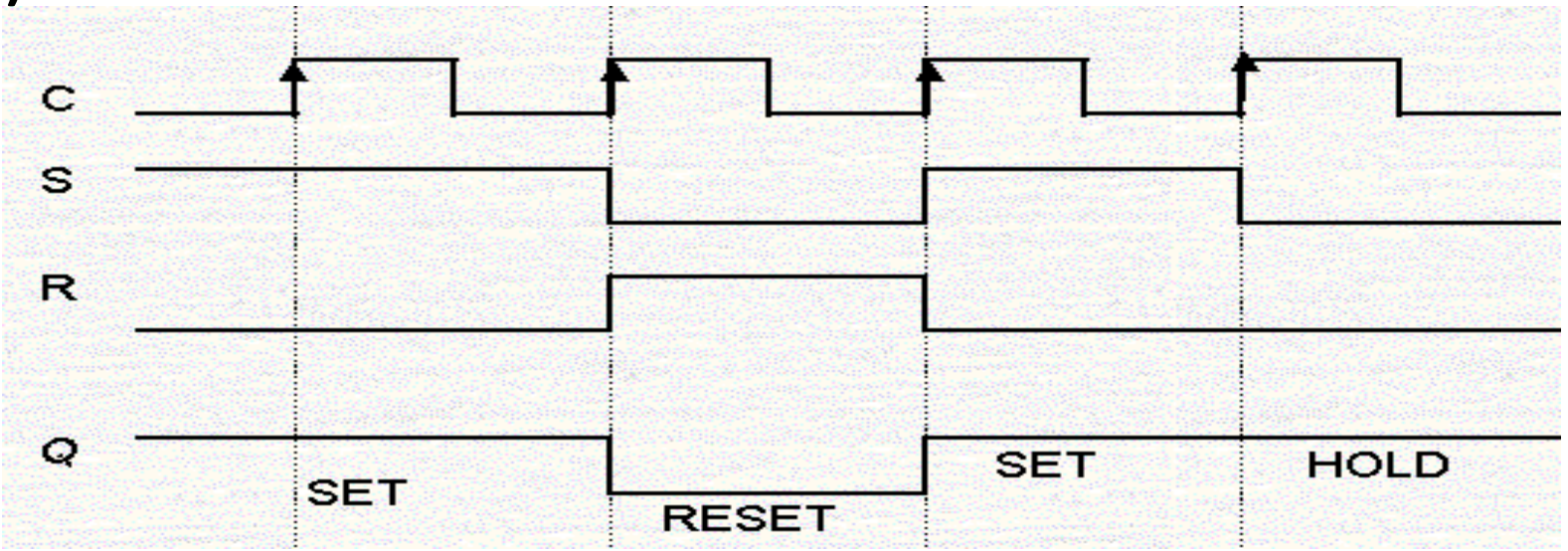
↑ = clock transition LOW to HIGH

X = irrelevant (don’t care)

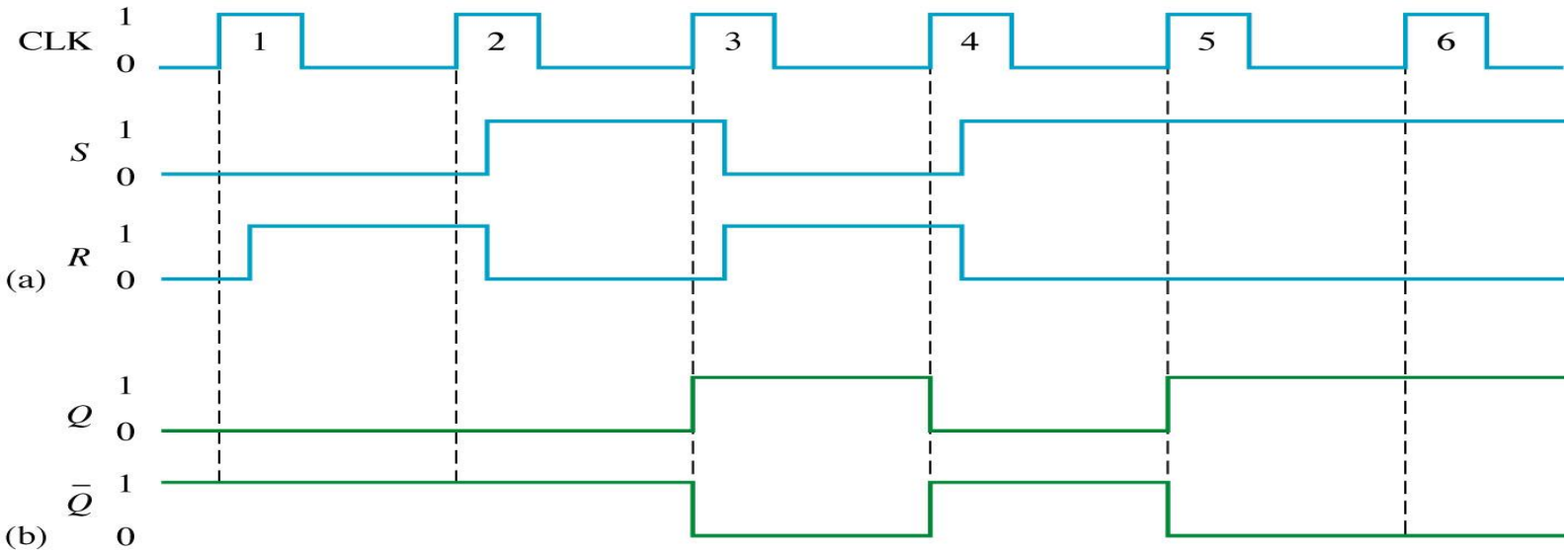
Q_0= output level prior to clock transition

Examples: Edge.. S-R Flip-Flops

a)



b)



Characteristic Table of an S-R Flip-flop

- From the name itself it is very clear that the *characteristic table of a flip-flop actually* gives us an idea about the character, *i.e., the working of the flip-flop. Now, from all our* above discussions, we know that the next state flip-flop output (Q_{n+1}) *depends on the present* inputs as well as the present output (Q_n).

Flip-flop inputs		*Present output*	*Next output*
S	R	Q_n	Q_{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	X
1	1	1	X

Cont----

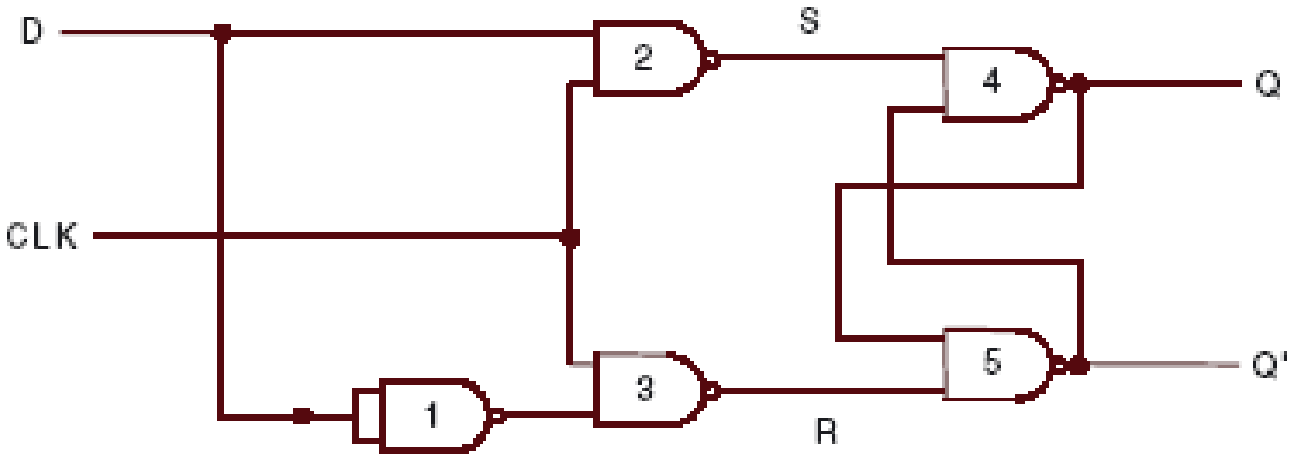
EXCITATION TABLE OF A FLIP-FLOP

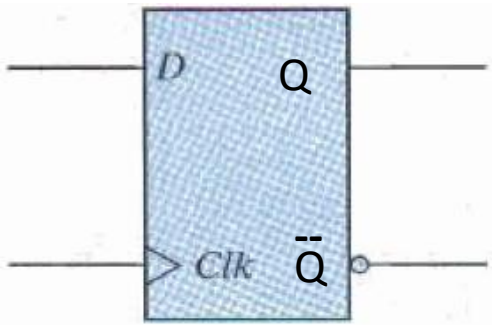
- The truth table of a flip-flop is also referred to as the characteristic table of a flip-flop, since this table refers to the operational characteristics of the flip-flop.
- If the present state and next state values are known we can find out the exciting values of the flip flop. If it drawn in a table form the called as excitation table.

Present	*Next*	*S-R FF*	
State (Q_n)	*State* (Q_{n+1})	S_n	R_n
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

2. D Flip flop

- D FF is useful when a single data bit (1 or 0) is to be stored.
- The addition of an inverter to an S-R FF creates basic D FF where a positive edge-triggered type is shown.





INPUTS		OUTPUTS		COMMENTS
D	CLK	Q	Q’	
1	↑	1	0	SET (store a 1)
0	↑	0	1	RESET (store 0)

Characteristic Table of D Flip-flop

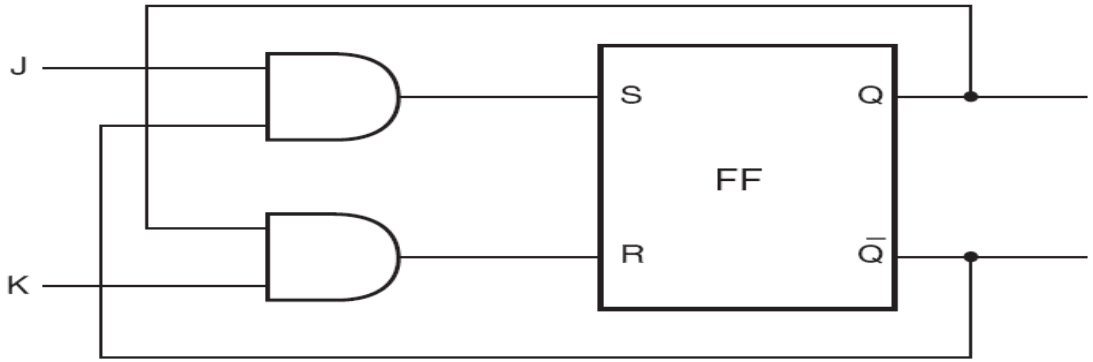
D	Q_n	Q_{n+1}
0	0	0
0	1	0
1	0	1
1	1	1

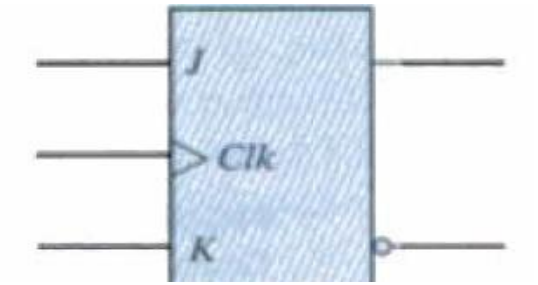
Excitation Table

Q_n	Q_{n+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

3. J-k Flip flop

- The JK FF is versatile and is a widely used type of FF.
- The difference between J-K and S-R is a J-K has no invalid state as SR.





J	K	CLK	Output		Comments
			Q	Q'	
0	0	↑	Q_0	Q'_0	No Change
0	1	↑	0	1	RESET
1	0	↑	1	0	SET
1	1	↑	Q'_0	Q_0	Toggle

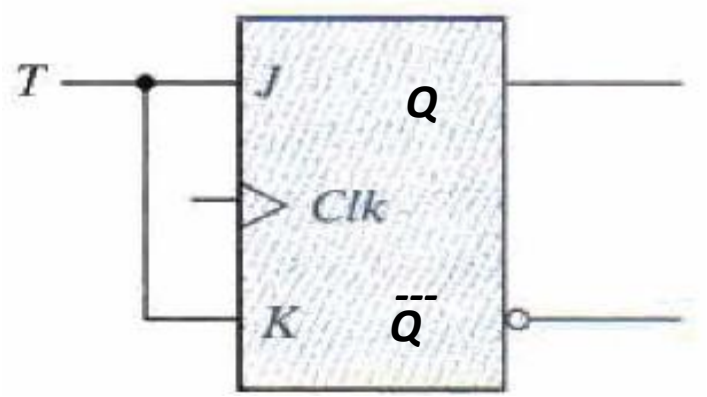
X'cs table

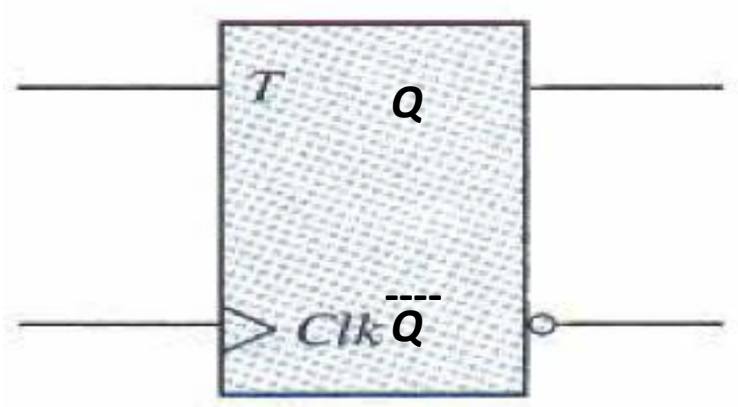
J	K	Q_n	Q_{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Excitation table

Q_n	Q_{n+1}	J	K
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

4. T- Flip flop





X'cs table

T	Q_n	Q_{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

Excitation table

Q_n	Q_{n+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

COUNTER DESIGN

- Flip-flops can be connected together to perform counting operations. Such a group of flip-flops is a counter.
- The number of flip-flops used and the way in which they are connected determine the number of states (called the modulus) and also the specific sequence of states that the counter goes through during each complete cycle.
- Counters are classified into two broad categories:

  A) Asynchronous Counter and

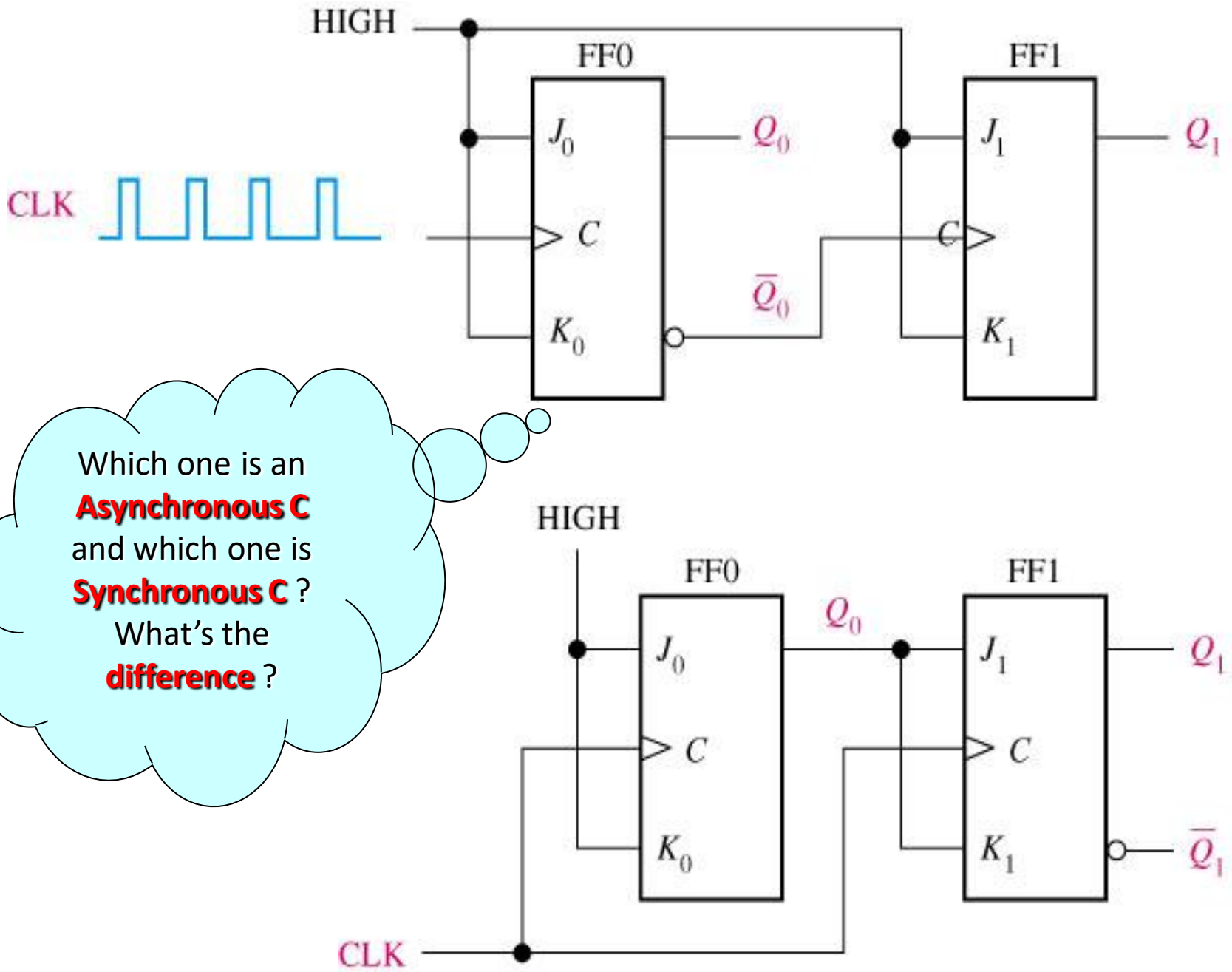
  B) Synchronous Counter
- In asynchronous counters, commonly called ripple counters, the first flip-flop is clocked by the external clock pulse and then each successive flip-flop is clocked by the output of the preceding flip-flop.
- In synchronous counters, the clock input is connected to all of the flip-flops so that they are clocked simultaneously.
- Within each of these two categories, counters are classified primarily by the type of sequence, the number of states, or the number of flip-flops in the counter.

A) ASYNCHRONOUS COUNTER (AC)

- The term asynchronous refers to events that do not have a fixed time relationship with each other and, generally, do not occur at the same time.
- An asynchronous counter is one in which the flip-flops (FF) within the counter do not change states at exactly the same time because they do not have a common clock pulse.

I 2-BIT ASYNCHRONOUS BINARY COUNTER

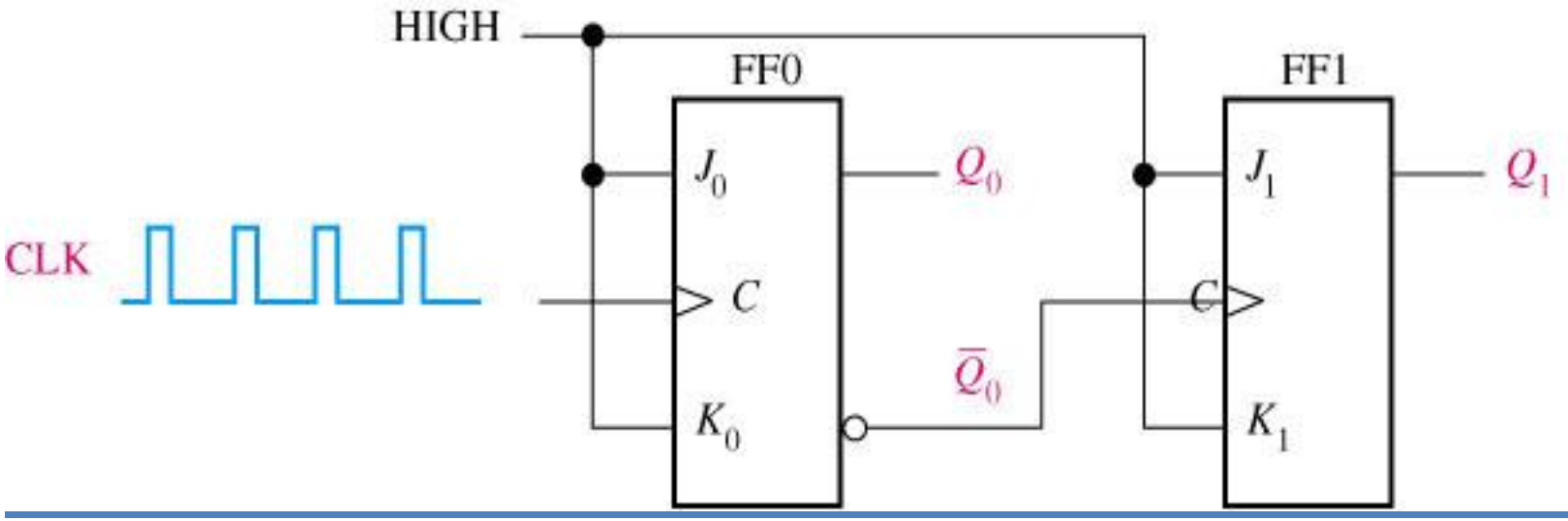
- Figure below shows a 2-bit counter connected for asynchronous operation. Notice that the clock (CLK) is applied to the clock input (C) of only the first flop-flop, FF0, which is always the least significant bit (LSB).
- The second flip-flop, FFI, is triggered by the $\overline{Q}$o output of FF0.
- FF0 changes state at the positive-going edge of each clock pulse, but FF1 changes only when triggered by a positive-going transition of the $\overline{Q}$o output of FF0.
- Therefore, the two flip-flops are never simultaneously triggered, so the counter operation is asynchronous.
- Asynchronous counters are also known as ripple counters.

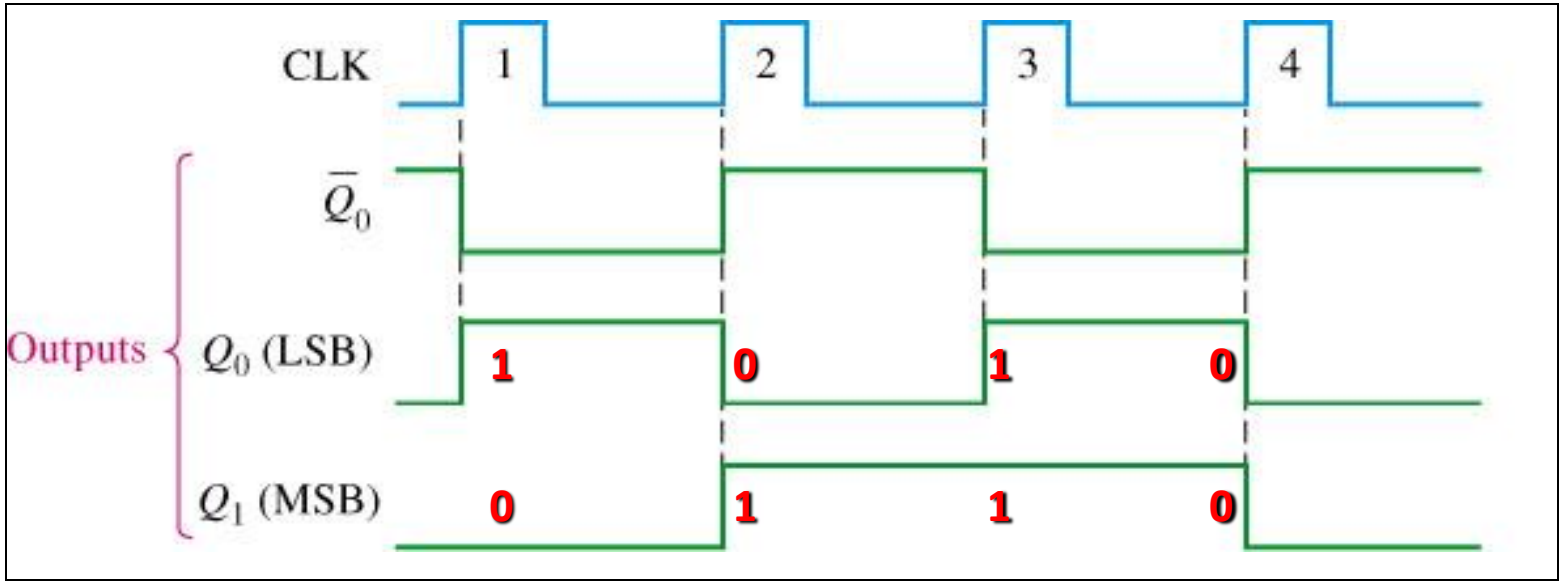

HIGH
FF0
FF1
J_0
Q_0
J_1
Q_1
CLK
C
C
$\overline{Q}_0$
K_0
K_1
Which one is an
Asynchronous C
and which one is
Synchronous C ?
What's the
difference ?
HIGH
FF0
FF1
Q_0
J_0
J_1
Q_1
C
C
K_0
K_1
$\overline{Q}_1$
CLK

….. 2-Bit Asynchronous Counter

CLOCK PULSE	$Q_{1(MSB)}$ (Bit 2)	$Q_{0\ (LSB)}$ (Bit 1)
Initially	0	0
1	0	1
2	1	0
3	1	1
4	0	0

.. 2-Bit Asynchronous Counter

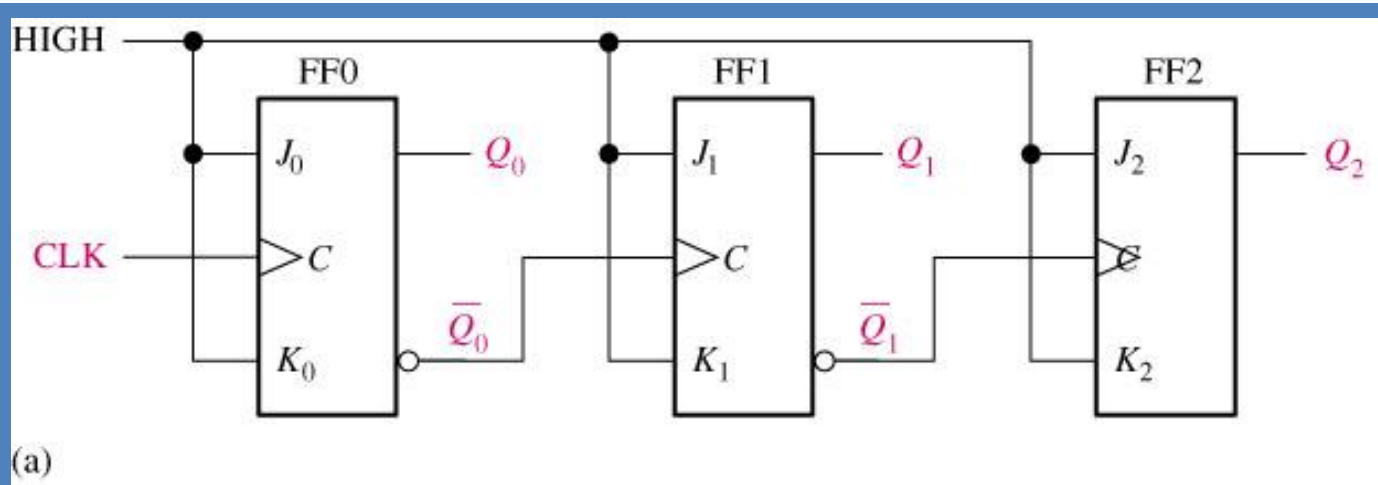




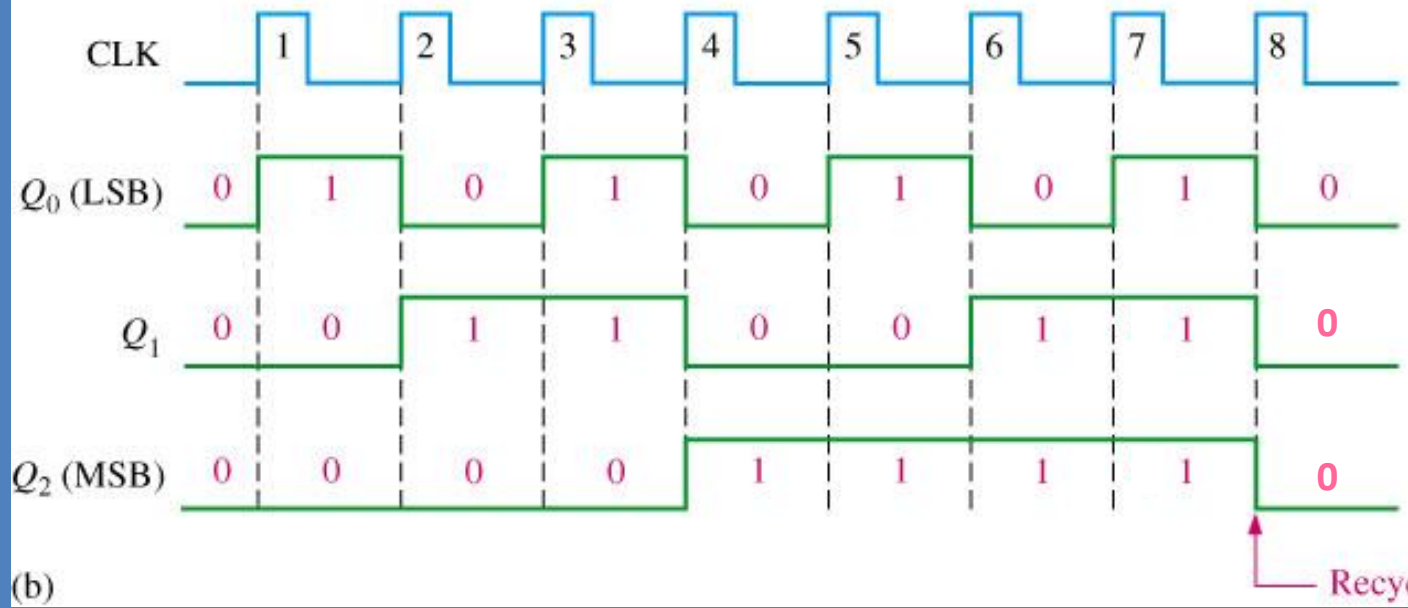
II 3-Bit Asynchronous Counter

CLOCK PULSE	$Q_{2(MSB)}$ (Bit 3)	Q_1 (Bit 2)	$Q_{0(LSB)}$ (Bit 1)
Initially	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1
8	0	0	0

.. 3-Bit Asynchronous Counter



(a)



(b)

….. 3-Bit Asynchronous Counter

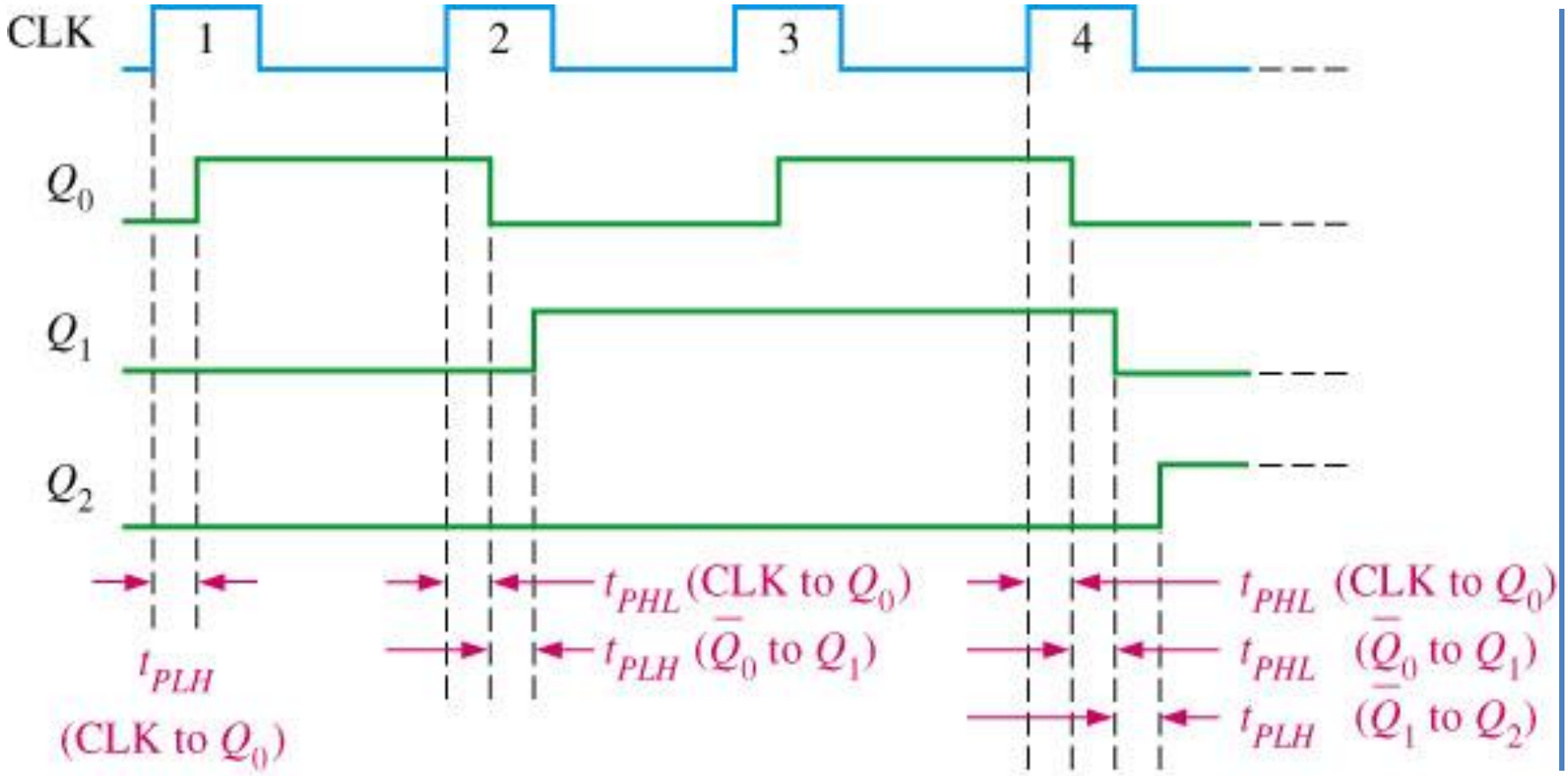
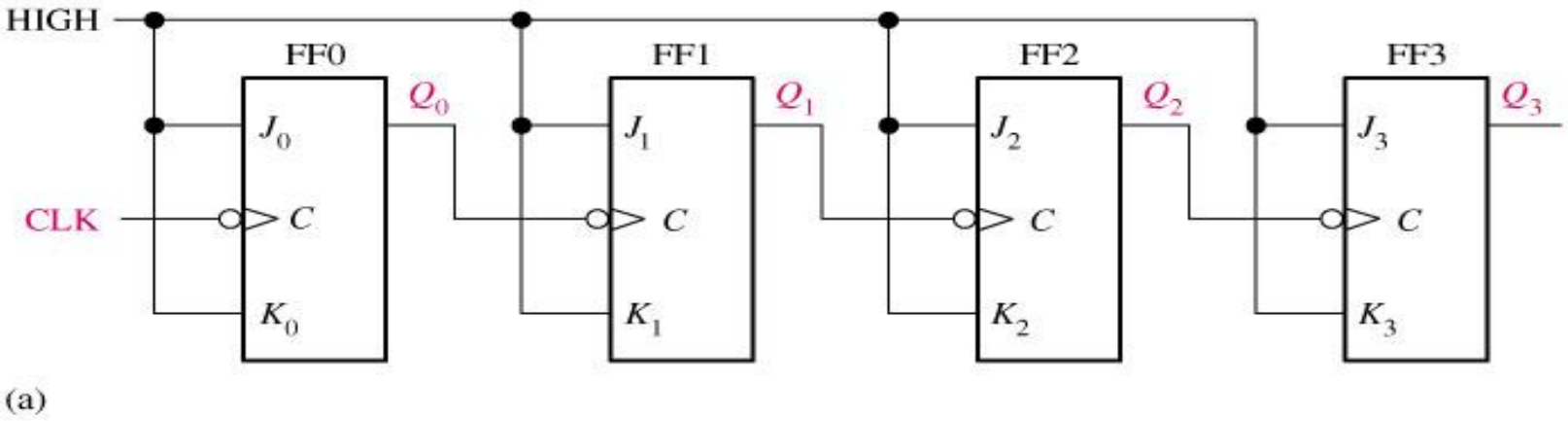
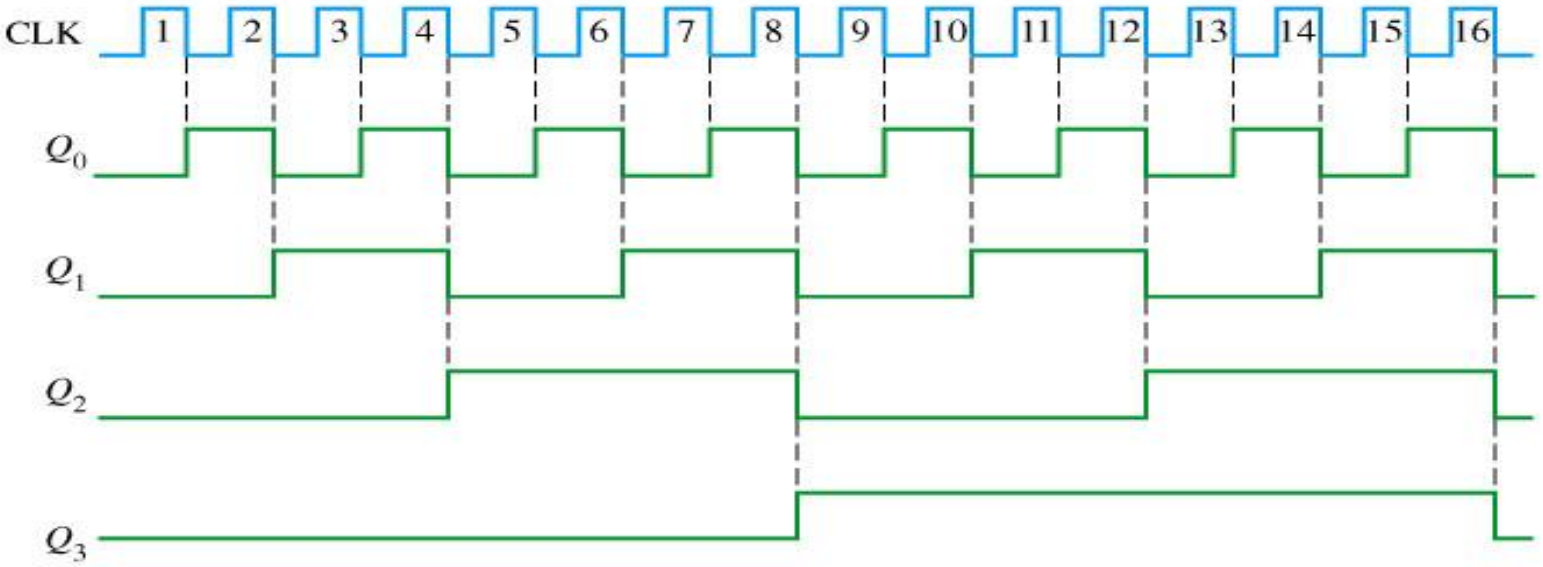


Figure : **Propagation delays in a 3-bit asynchronous (ripple-clocked) binary counter.**

III 4-Bit Asynchronous Counter



(a)



(b)

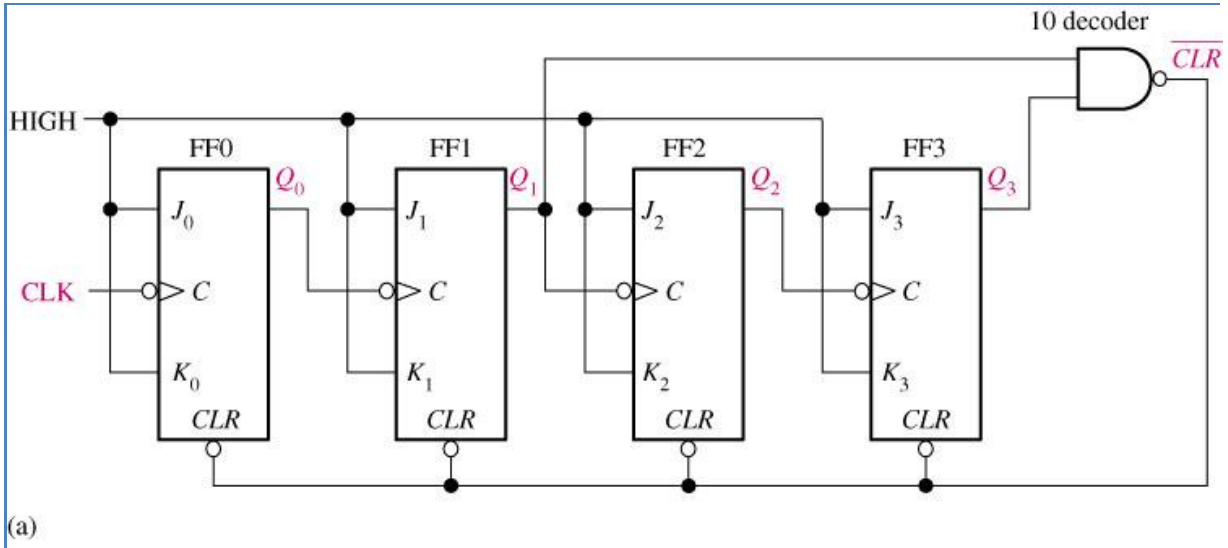
IV Asynchronous Decade Counter

- The **modulus** is the number of unique states through which the counter will sequence
- Maximum possible number of states of counter is 2^n , n is the number of flip-flops in the counter
  - Example : **Modulus 8 = 2^3** (Need 3 flip flops)
- Counter can be designed to have a number of states in their sequence that is less than maximum, 2^n**.** This called **truncated sequence**

.. Asynchronous Decade Counter

- One common modulus for counters is truncated sequences is ten (MOD10).
- It called **BCD decade counters**.
- Requires 4 flip-flops. Max 2^4=16
- count zero,(0000) through nine (1001)
  - 0,1,2,3,4,5,6,7,8,9,0,1,…..
- Done by : when the counter goes into ten (1010), the decoding gate output goes LOW and asynchronously **resets all** the flip-flops.

.. ..Asynchronous Decade Counter



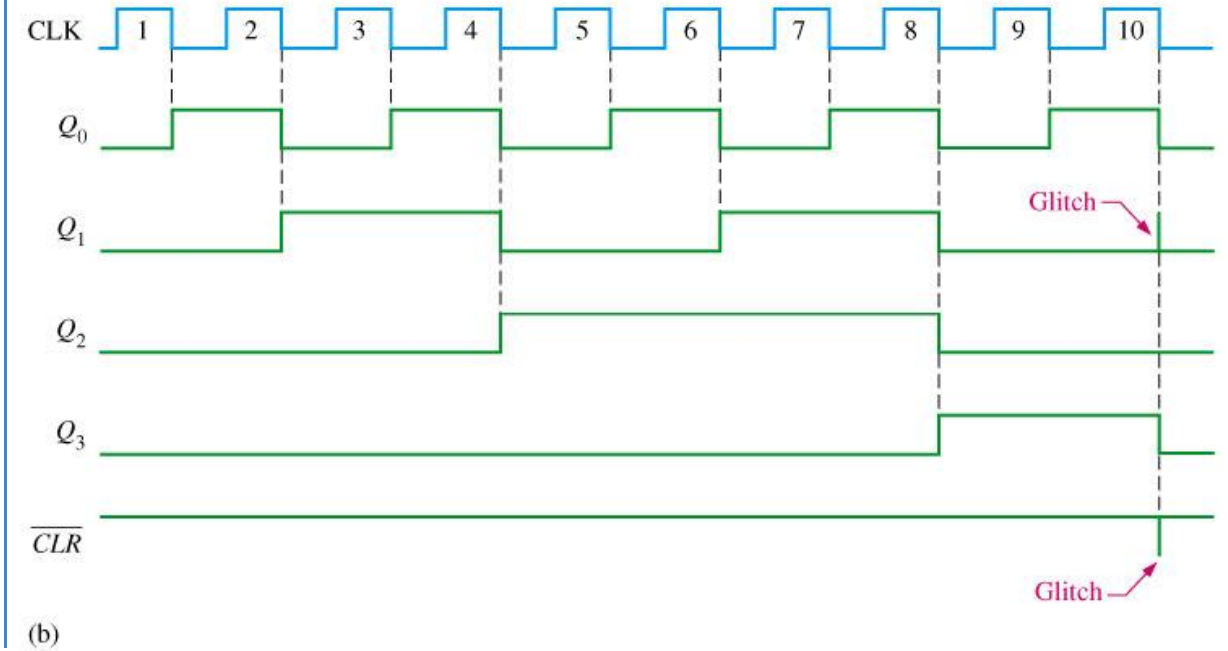
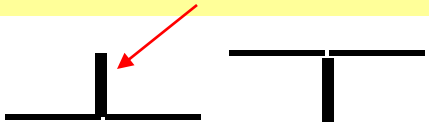


Figure **An asynchronously clocked decade counter with asynchronous recycling.**

Notice that there is a glitch in Q1. The reason of this glitch is that Q1 must first go HIGH before the count 10 can be decoded. Several nanoseconds after the decoding gate goes LOW.



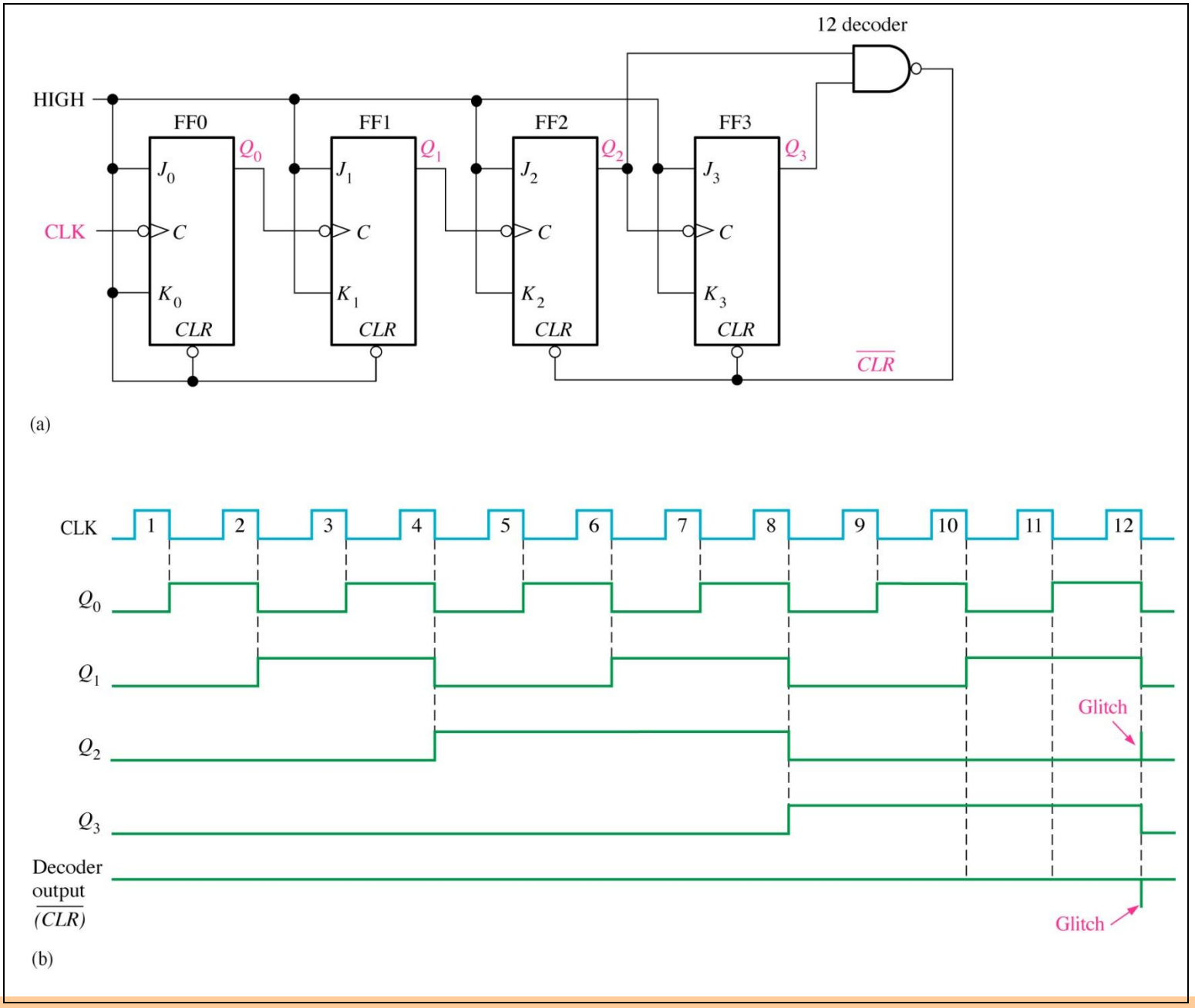
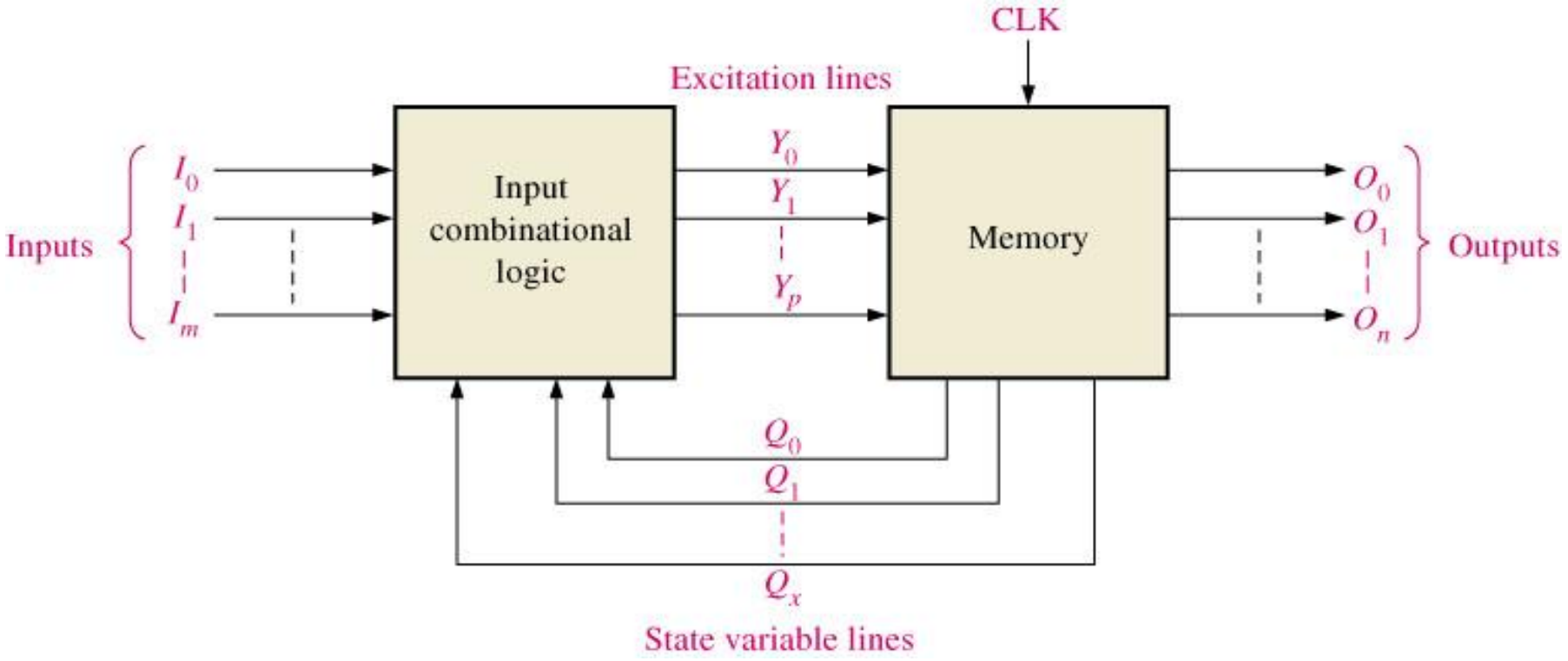


Figure **Asynchronously clocked modulus-12 counter with asynchronous recycling.**

B) SYNCHRONOUS COUNTERS

- Before proceeding with a specific counter design technique, let's begin with a general definition of a sequential circuit or state machine: A general sequential circuit consists of a combinational logic section and a memory section (flip-flops), as shown in Figure below.
- In a clocked sequential circuit, there is a clock input to the memory section as indicated.



Procedure to design synchronous *sequential circuit*

1. Construct the state diagram according to the given problem (if require)
2. Construct the state table from the problem or from the state diagram.
3. Construct the excitation table with the present and next value obtain from state table for the given flip flop.
4. Find out the expression for each and every flip flop from the excitation table
5. Transfer the transition table to Karnaugh maps. There is a Karnaugh map for each input of each flip-flop.
6. Group the Karnaugh map cells to generate and derive the logic expression for each flip-flop input.
7. Implement the expressions with combinational logic and combine with the flip-flops to create the sequential ckt

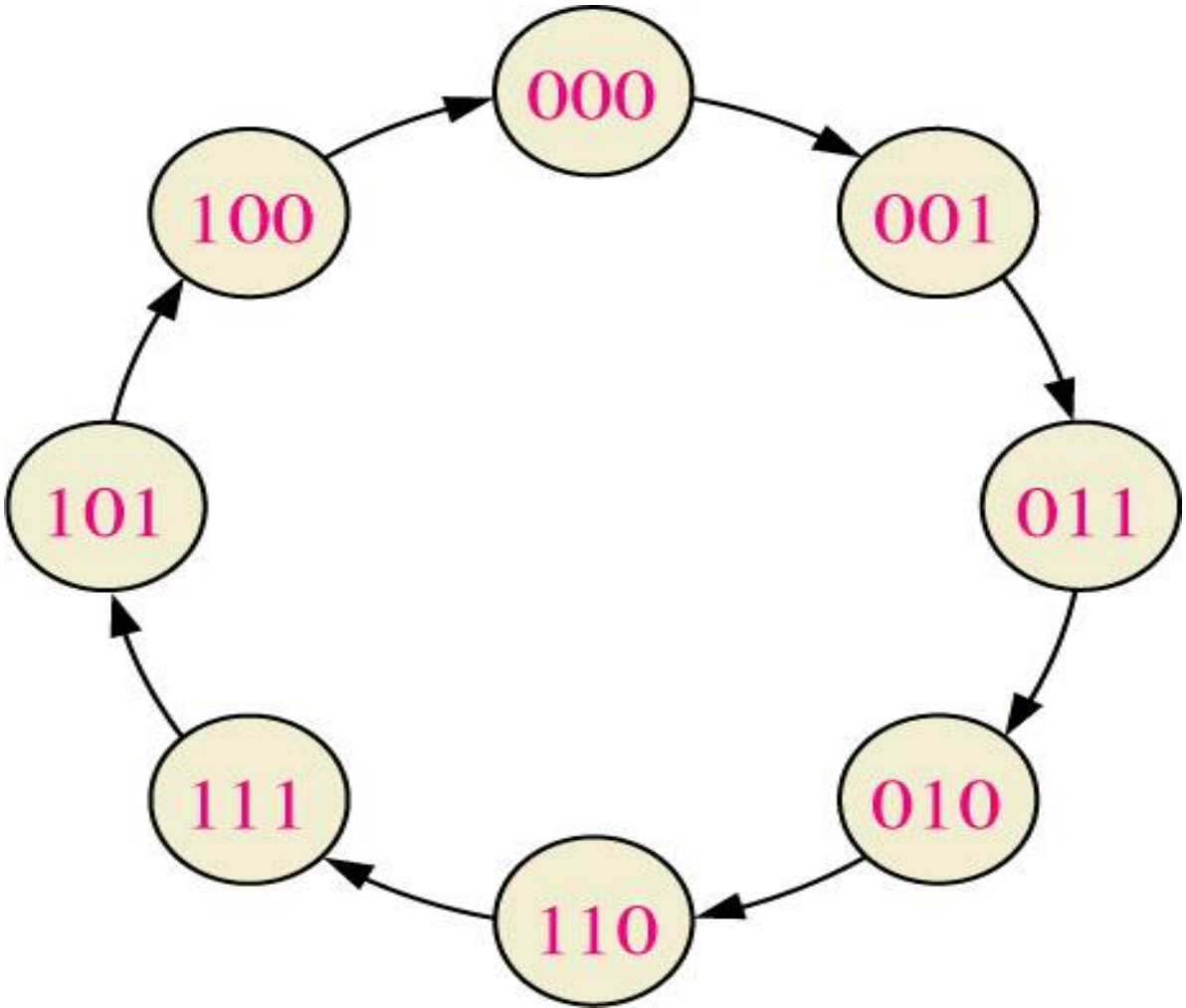
....SYNCHRONOUS COUNTERS

Design of Synchronous Counter procedure:

- **Step 1 : State Diagram**
- **Step 2 : Create Next State Table**
- **Step 3 : Flip-Flop Transition Table**
- **Step 4: Karnaugh Maps**
- **Step 5 and Step 6 : Logic Expressions and Counter Implementation**

Example 1: Count with adjacent binary number 000, 001, 011, 010, 110, 111, 101, 100, 000….

Step 1 : State Diagram



Step 2 : Create Next State Table

Present state			Next state		
Q_{2n}	Q_{1n}	Q_{0n}	$Q_{2(n+1)}$	$Q_{1(n+1)}$	$Q_{0(n+1)}$
0	0	0	0	0	1
0	0	1	0	1	1
0	1	0	1	1	0
0	1	1	0	1	0
1	0	0	0	0	0
1	0	1	1	0	0
1	1	0	1	1	1
1	1	1	1	0	1

Step 3: Flip-Flop Transition Table (J-K FF)

OUTPUT TRANSITIONS			FLIP-FLOP INPUTS	
Q_N		Q_{N+1}	J	K
0	⟶	0	0	X
0	⟶	1	1	X
1	⟶	0	X	1
1	⟶	1	X	0

Q_N: present state

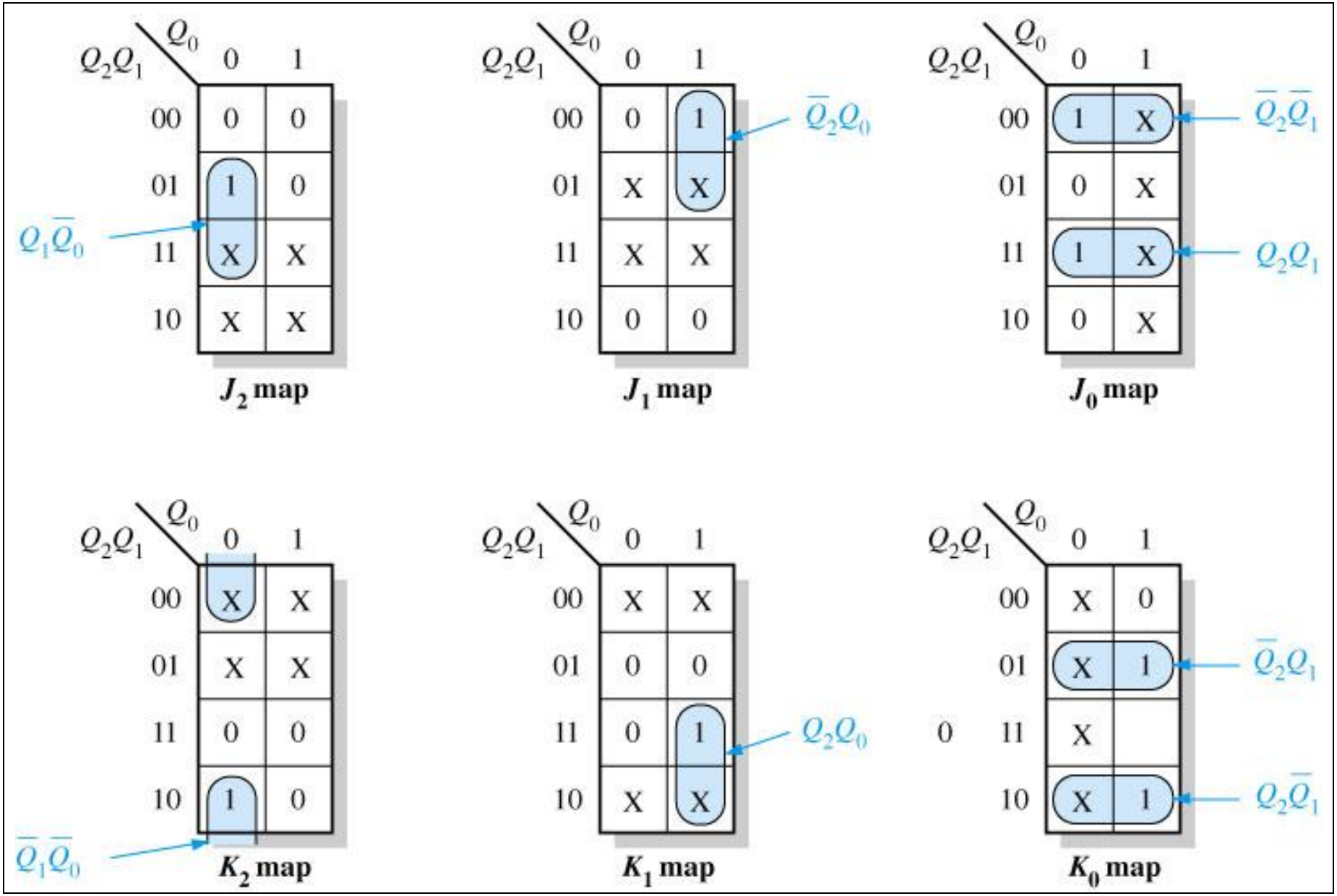
Q_{N+1}: next state

X: "don't care"

State Table

Present state			Next state			Flip-Flop Inputs					
Q_{2n}	Q_{1n}	Q_{0n}	$Q_{2(n+1)}$	$Q_{1(n+1)}$	$Q_{0(n+1)}$	J_2	K_2	J_1	K_1	J_0	K_0
0	0	0	0	0	1	0	X	0	X	1	X
0	0	1	0	1	1	0	X	1	X	X	0
0	1	0	1	1	0	1	X	X	0	0	X
0	1	1	0	1	0	0	X	X	0	X	1
1	0	0	0	0	0	X	1	0	X	0	X
1	0	1	1	0	0	X	0	0	X	X	1
1	1	0	1	1	1	X	0	X	0	1	X
1	1	1	1	0	1	X	0	X	1	X	0

Step 4: **Karnaugh Maps**



Step 5 and Step 6 :
Logic Expressions and Counter Implementation

$J2 = Q_1\overline{Q}_0$

$J1 = \overline{Q}_2 Q_0$

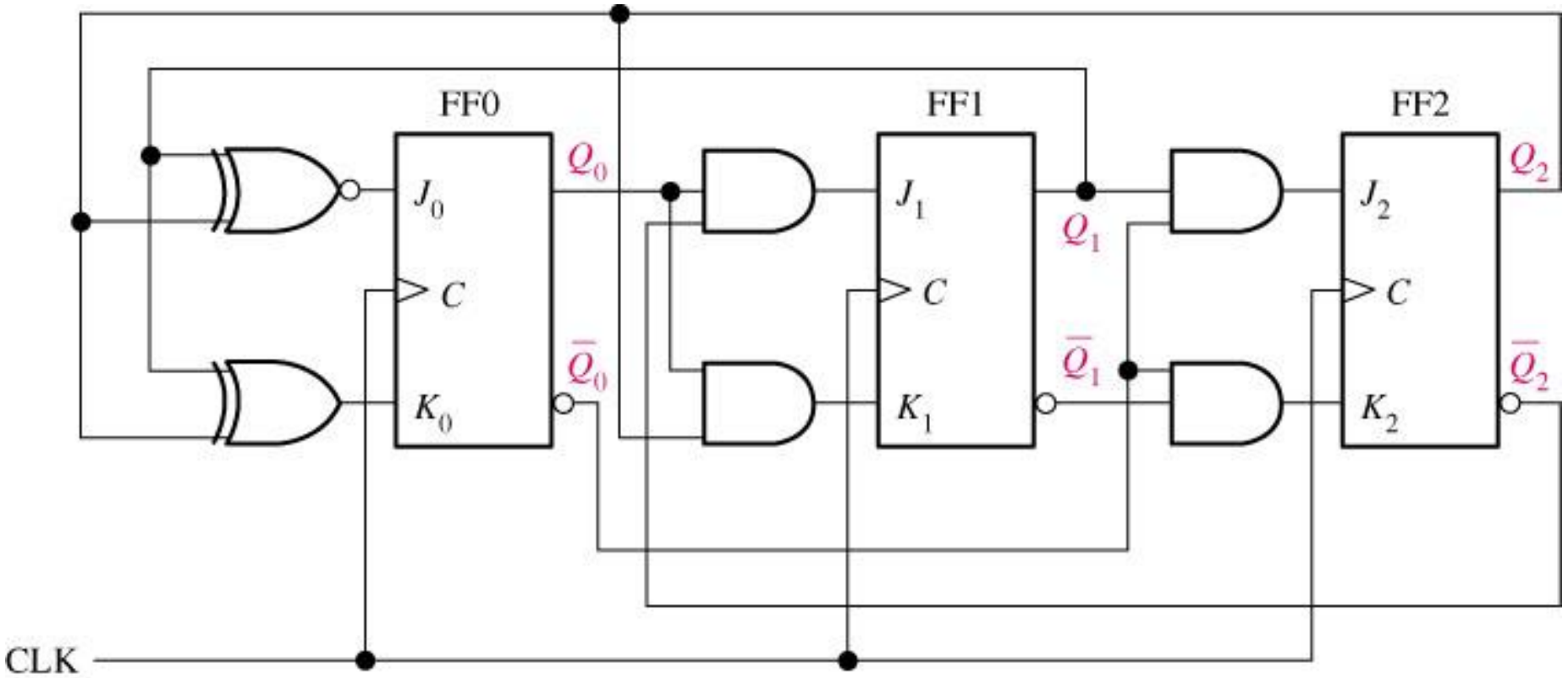
$J0 = \overline{Q}_2\overline{Q}_1 + Q_2 Q_1$

$K2 = \overline{Q}_1\overline{Q}_0$

$K1 = Q_2 Q_0$

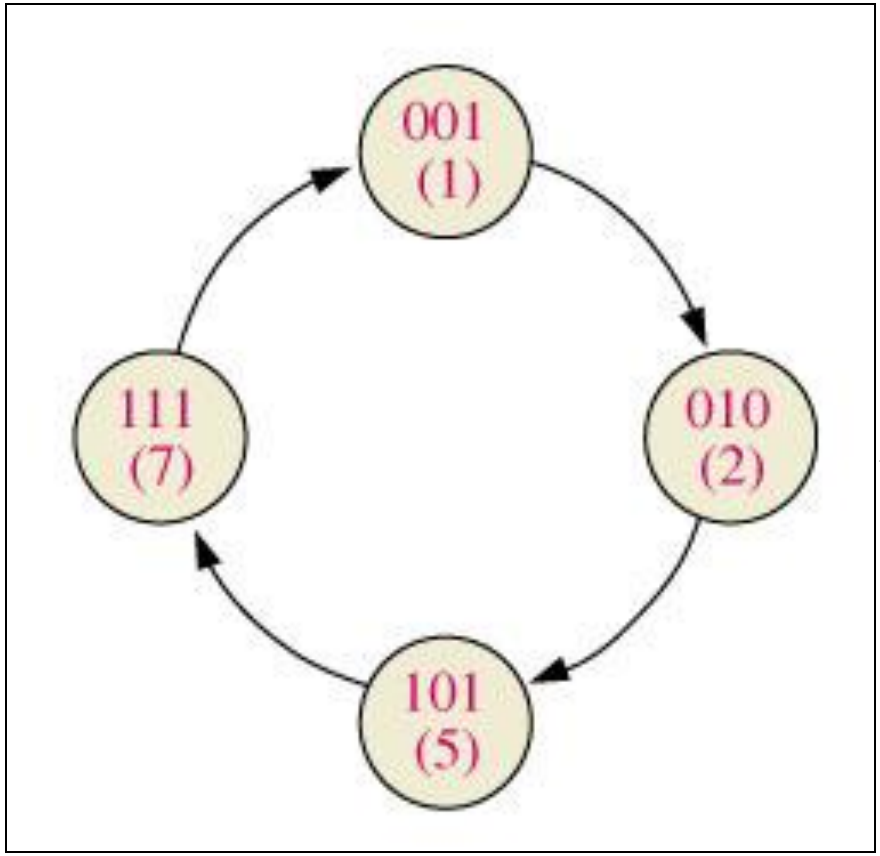
$K0 = \overline{Q}_2 Q_1 + Q_2\overline{Q}_1$

Step 5 and Step 6 :
Logic Expressions and Counter Implementation



Example 2 : 001, 010, 101, 111, 001….

Step 1 : State Diagram



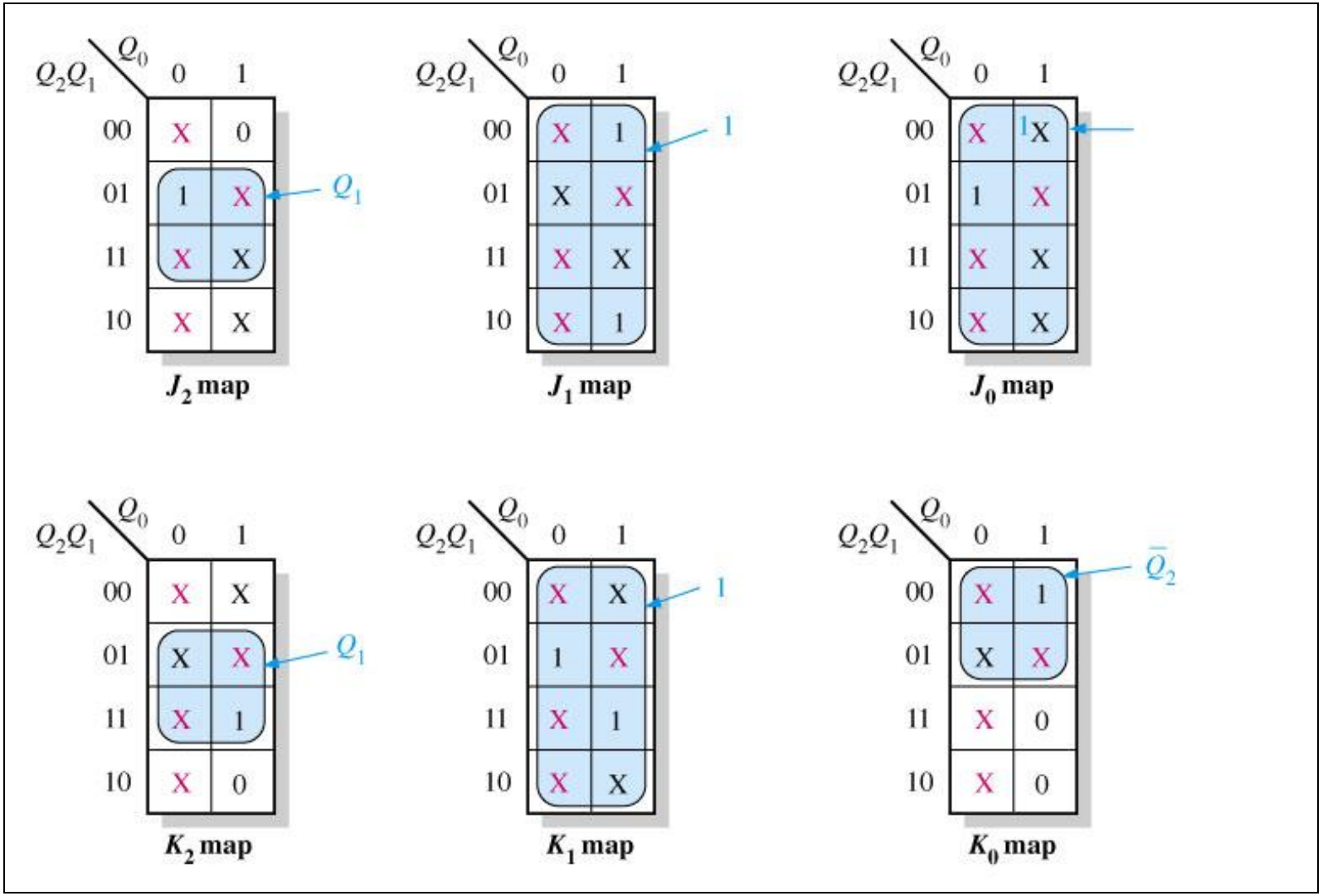
What is the **Next State Table** (Step 2) for this State Diagram ? Create it.

Step 2 & 3- State Table

Exercise

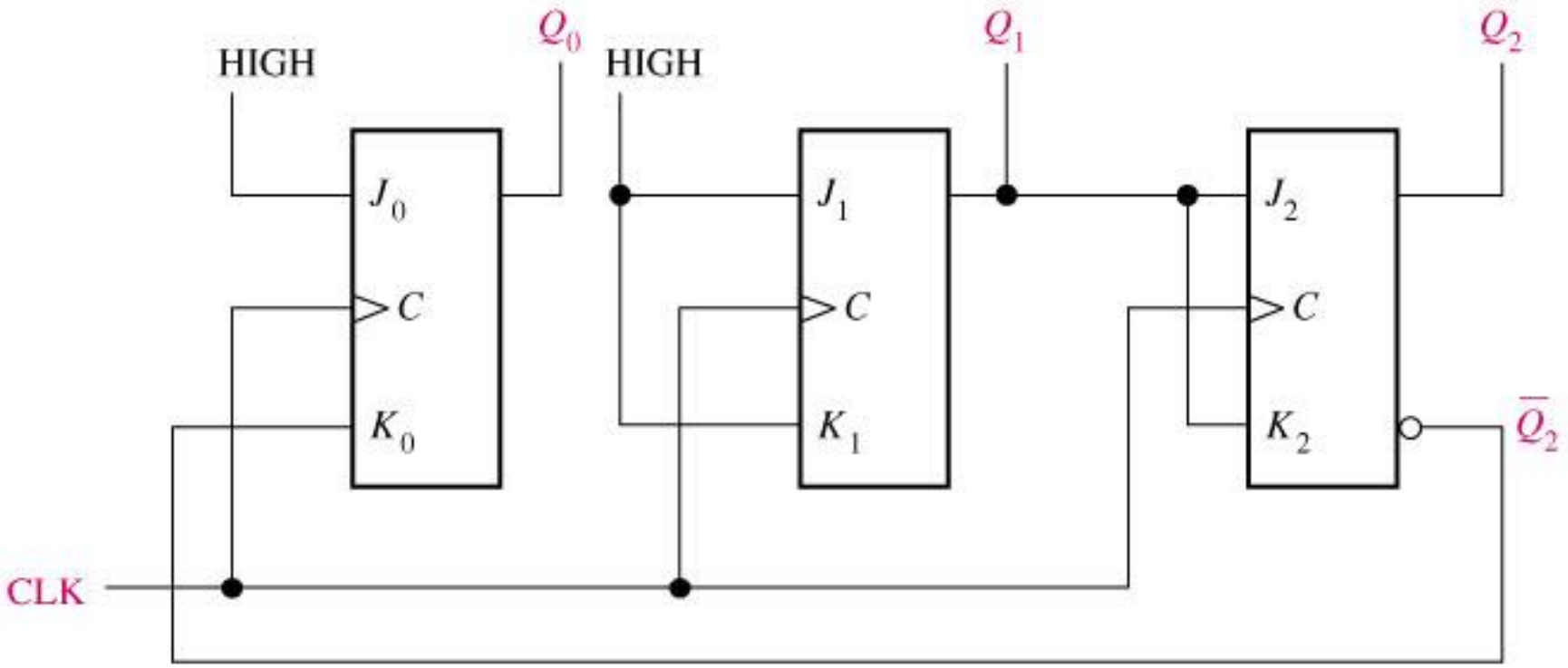
------Example 2 :

Step 3 & 4 : Karnaugh Map



Example 2 :

Step 5 & 6 : Logic Expressions and Counter Implementation



Modulo- n counter:

Modulo –n counter is a counter that count from 0 to n-1.

Ex. Mod-4 counter: 0, 1, 2, 3,0,1..

Mod-10 counter: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 1, …

Examples :-

1. **Design a counter with the following repeated binary sequence: 0, 1, 2, 3, 4, 5, 6, 7. Use JK flip flops.**
2. Design mod-5 counter using D-Flip Flop.
3. Design a 3-bit up/down counter that works according to mode bit , if the mode = 0, then it is counting down and if mode =1 it counts upwards.
4. Design a sequence generator to generate 0, 1, 5, 6, 2, if mode is 0 and 2, 7, 1, 3, 5 if mode is 1.
5. Design 4-Bit Synchronous Decade Counter to produce a BCD counting sequence of the following binary sequence. Use T- flip-flops.

   0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0,…

END OF CHAPTER-6

QUESTION???