Flip flops

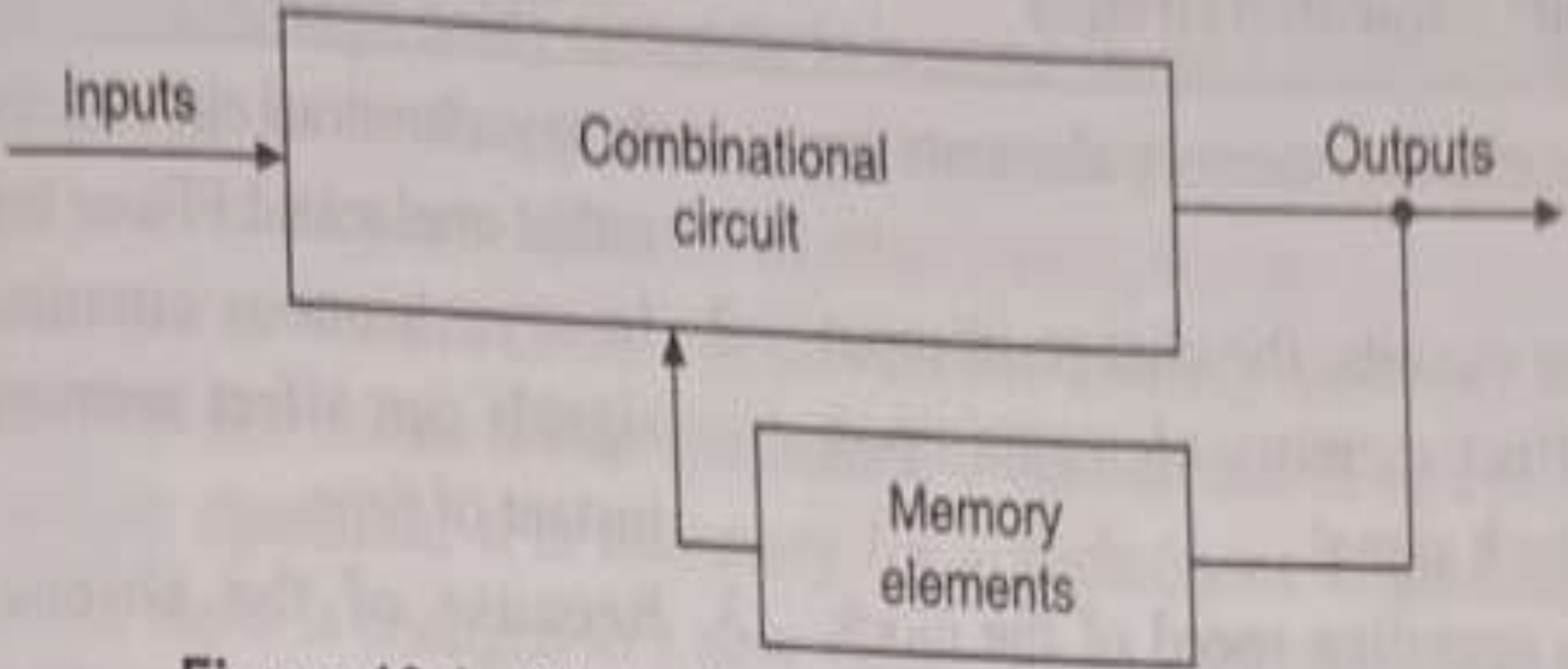


Figure 10.1 Block diagram of a sequential circuit.

- The information stored in the memory element at any given time defines the present state of the sequential circuit.
- The present state and external inputs determine the outputs of the next state of the sequential circuit.

Table 10.1 Comparison between combinational and sequential circuits

Combinational circuits	Sequential circuits
1. In combinational circuits, the output variables at any instant of time are dependent only on the present input variables.	1. In sequential circuits, the output variables at any instant of time are dependent not only on the present input variables, but also on the present state, i.e. on the past history of the system.
2. Memory unit is not required in combinational circuits.	2. Memory unit is required to store the past history of the input variables in sequential circuits.
3. Combinational circuits are faster because the delay between the input and the output is due to propagation delay of gates only.	3. Sequential circuits are slower than combinational circuits.
4. Combinational circuits are easy to design.	4. Sequential circuits are comparatively harder to design.

Classification of sequential circuits

- Synchronous sequential circuits
- Asynchronous sequential circuits
- depending upon timing of their signals

- Circuits will be active only when clock is present.
- The sequential circuits which are not controlled by a clock are called as Asynchronous sequential circuits

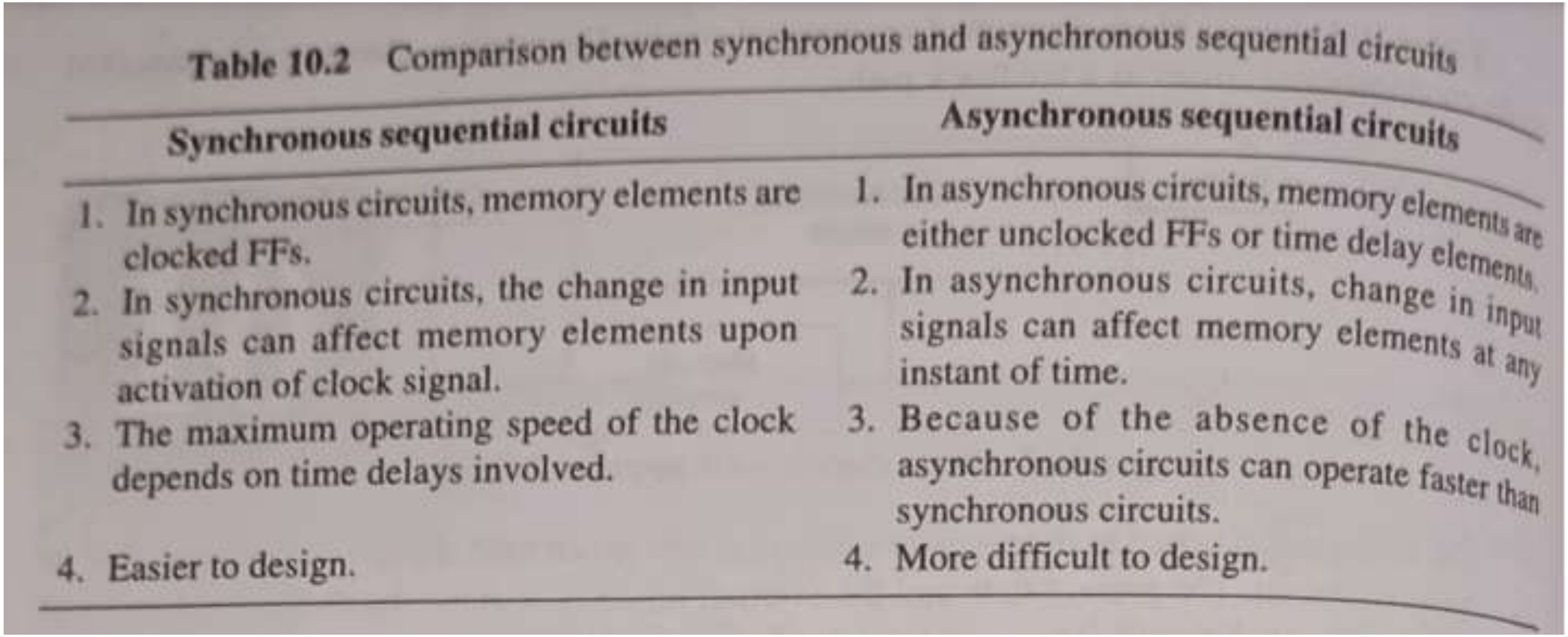


Table 10.2 Comparison between synchronous and asynchronous sequential circuits

Synchronous sequential circuits	Asynchronous sequential circuits
1. In synchronous circuits, memory elements are clocked FFs.	1. In asynchronous circuits, memory elements are either unclocked FFs or time delay elements.
2. In synchronous circuits, the change in input signals can affect memory elements upon activation of clock signal.	2. In asynchronous circuits, change in input signals can affect memory elements at any instant of time.
3. The maximum operating speed of the clock depends on time delays involved.	3. Because of the absence of the clock, asynchronous circuits can operate faster than synchronous circuits.
4. Easier to design.	4. More difficult to design.

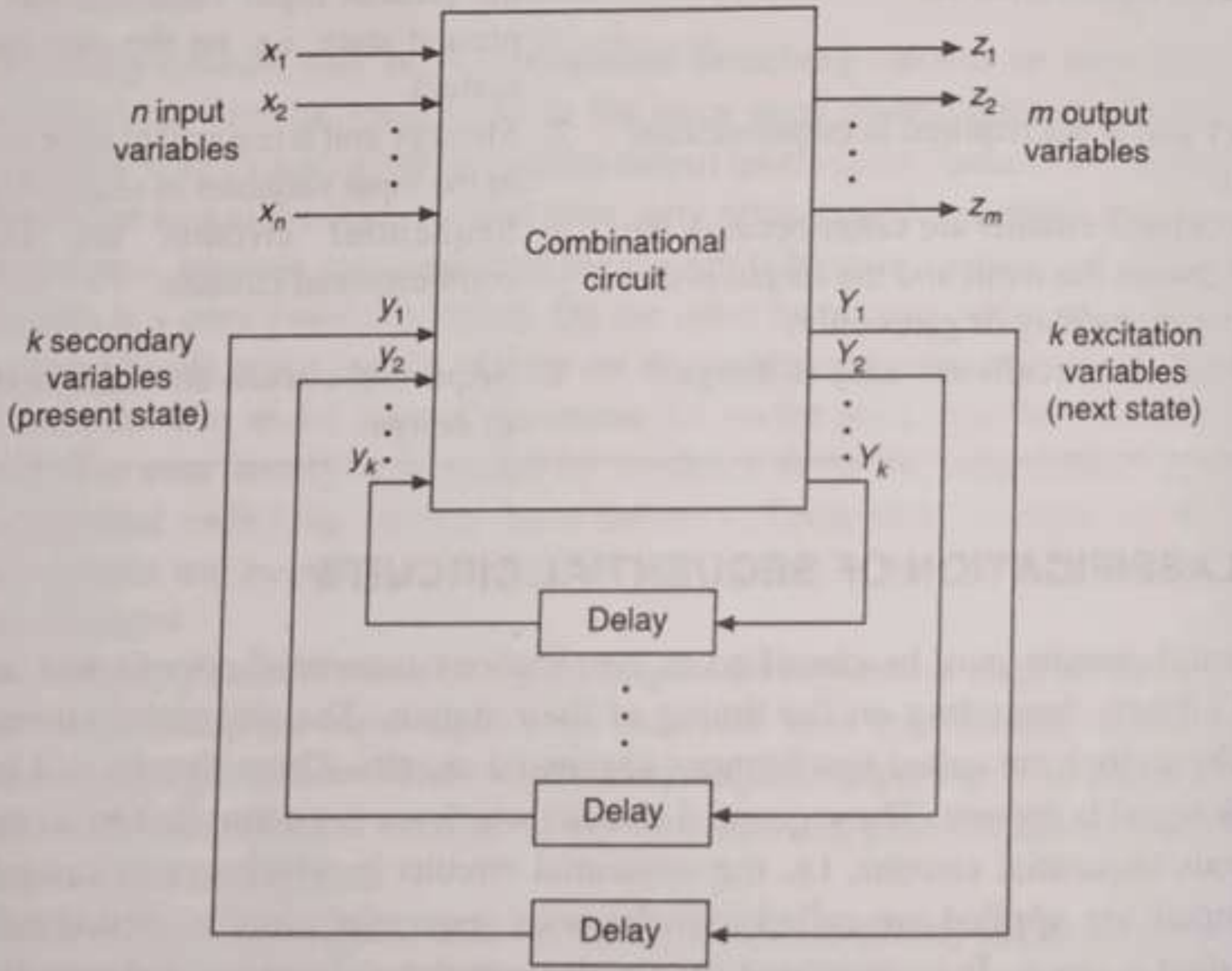


Figure 10.2 Block diagram of an asynchronous sequential circuit.

Latches and flip flops

- Most important memory element is the flip flop.
- Logic gate – no storage
- More logic gates – storage capacity
- Flip flops basic building blocks of sequential circuits
- Flip flops – bistable multivibrator
- Two stable states
- Flip flops has two outputs Q and Q’
- Q = 1 FF sets , Q’ = 1 FF resets

Flip flop symbol

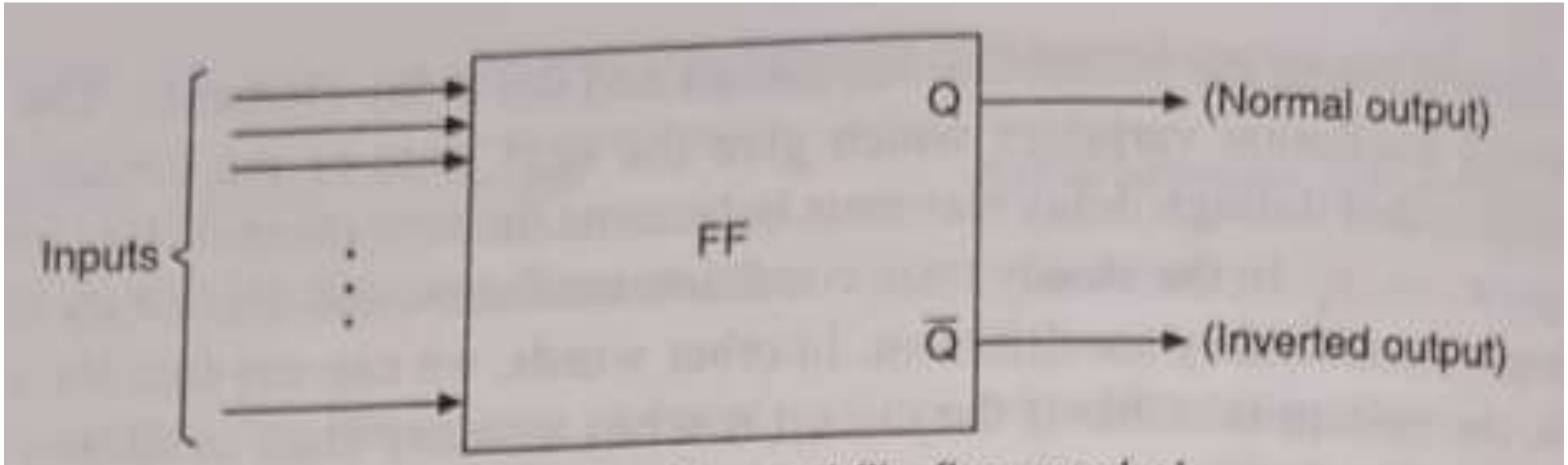


Figure 10.3 General flip-flop symbol.

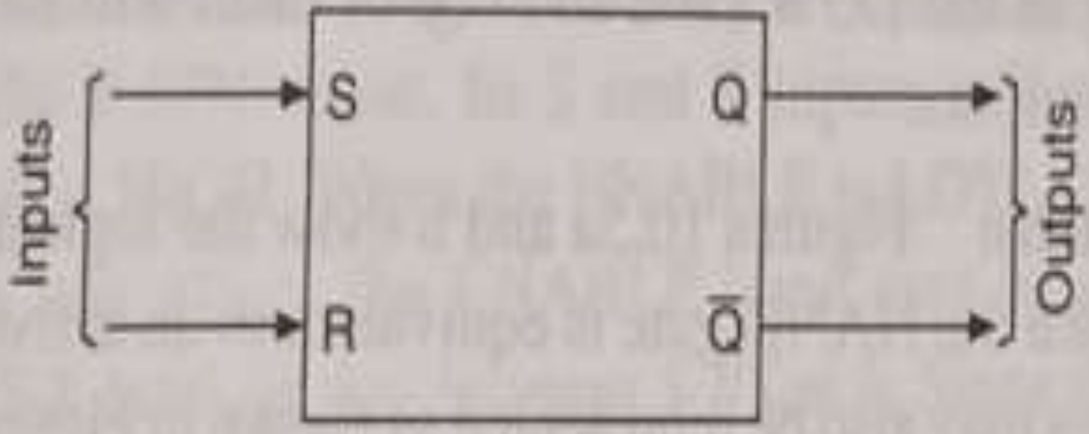
- FF serves as a one bit memory, when Q = 1 stores 1, and when Q’ = 0 stores o.
- FFs are fundamental component of counters and shift resistors

LATCH

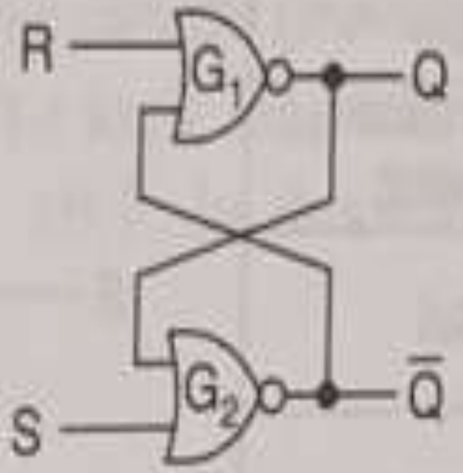
- Latch – Non clocked FF
- Because these FFs ‘latch on’ to a 1 or to a 0 immediately upon on receiving the input pulse called SET or RESET
- Not dependent on clock signal

The S R Latch

- Simplest type of FF
- Two outputs Q and Q’, two inputs S and R
- Q – HIGH
- Q’ – LOW
- Constructed using NAND gates or NOR gates
- S – SET
- R - RESET



(a) Logic symbol



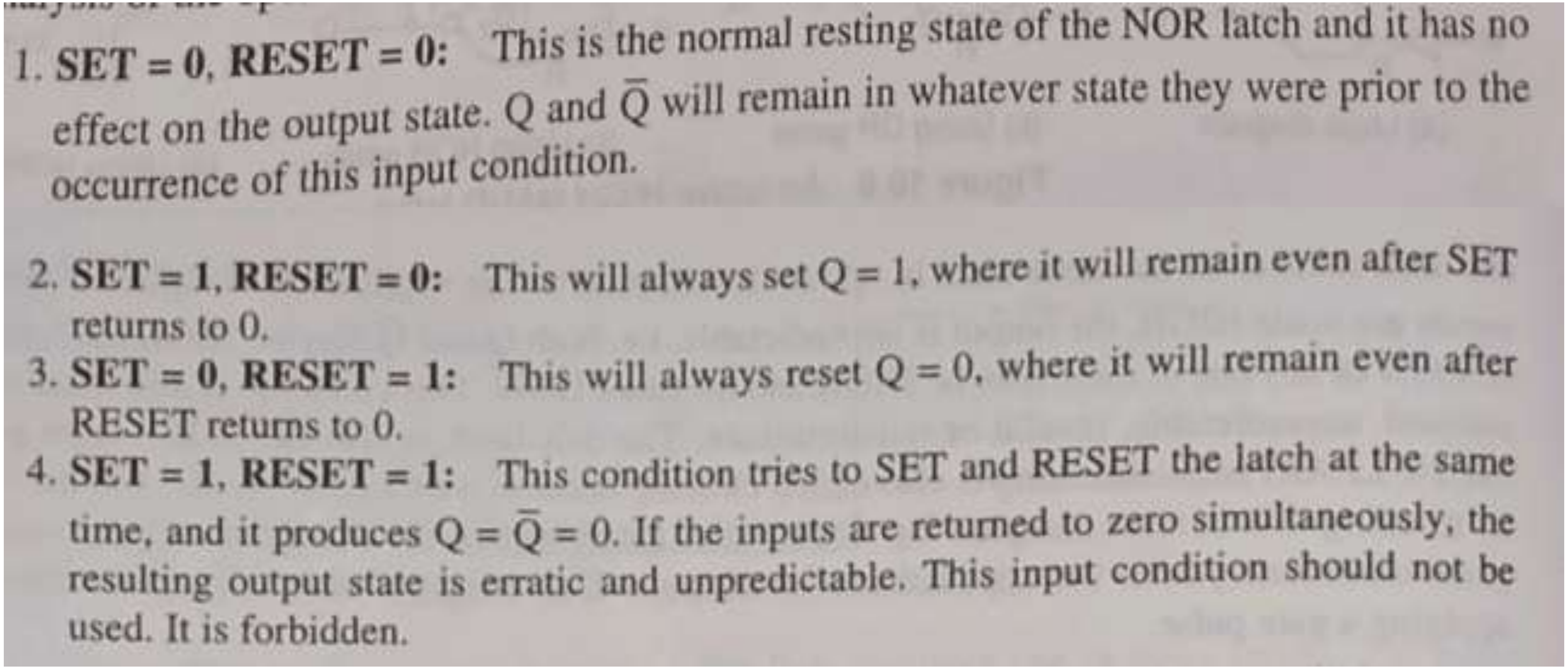
(b) Logic diagram

S	R	Q_n	Q_{n+1}	State
0	0	0	0	No Change (NC)
0	0	1	1	
0	1	0	0	Reset
0	1	1	0	
1	0	0	1	Set
1	0	1	1	
1	1	0	x	Indeterminate (invalid)
1	1	1	x	

(c) Truth table

Figure 10.4 Active-HIGH S-R latch.

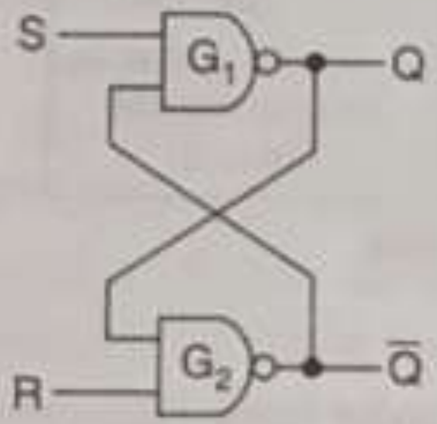
Operation



1. **SET = 0, RESET = 0:** This is the normal resting state of the NOR latch and it has no effect on the output state. Q and $\bar{Q}$ will remain in whatever state they were prior to the occurrence of this input condition.

2. **SET = 1, RESET = 0:** This will always set Q = 1, where it will remain even after SET returns to 0.
3. **SET = 0, RESET = 1:** This will always reset Q = 0, where it will remain even after RESET returns to 0.
4. **SET = 1, RESET = 1:** This condition tries to SET and RESET the latch at the same time, and it produces $Q = \bar{Q} = 0$. If the inputs are returned to zero simultaneously, the resulting output state is erratic and unpredictable. This input condition should not be used. It is forbidden.

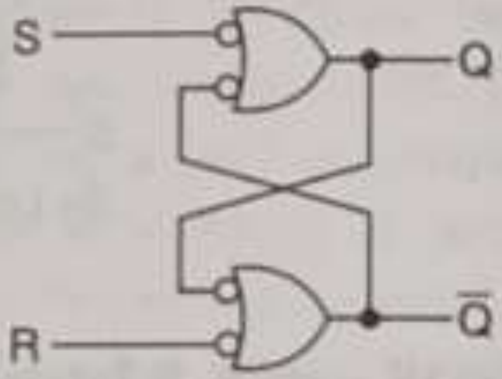
The NAND gate S-R latch (active-low S-R latch): Figures 10.5a and b show the logic diagram and truth table of an active-LOW S-R latch. Since the NAND gate is equivalent to an active-LOW OR gate, an active-LOW S-R latch using OR gates may also be represented as shown in Figure 10.5c.



(a) Using NAND gates

S	R	Q_n	Q_{n+1}	State
0	0	0	×	Indeterminate (invalid)
0	0	1	×	
0	1	0	1	Set
0	1	1	1	
1	0	0	0	Reset
1	0	1	0	
1	1	0	0	No Change (NC)
1	1	1	1	

(b) Truth table



(c) Using OR gates

Figure 10.5 An active-LOW S-R latch.

The $\bar{S}$-$\bar{R}$ latch (active-high NAND latch): An active-LOW NAND latch can be converted into an active-HIGH NAND latch by inserting the inverters at the S and R inputs. Figure 10.6 shows the proof.

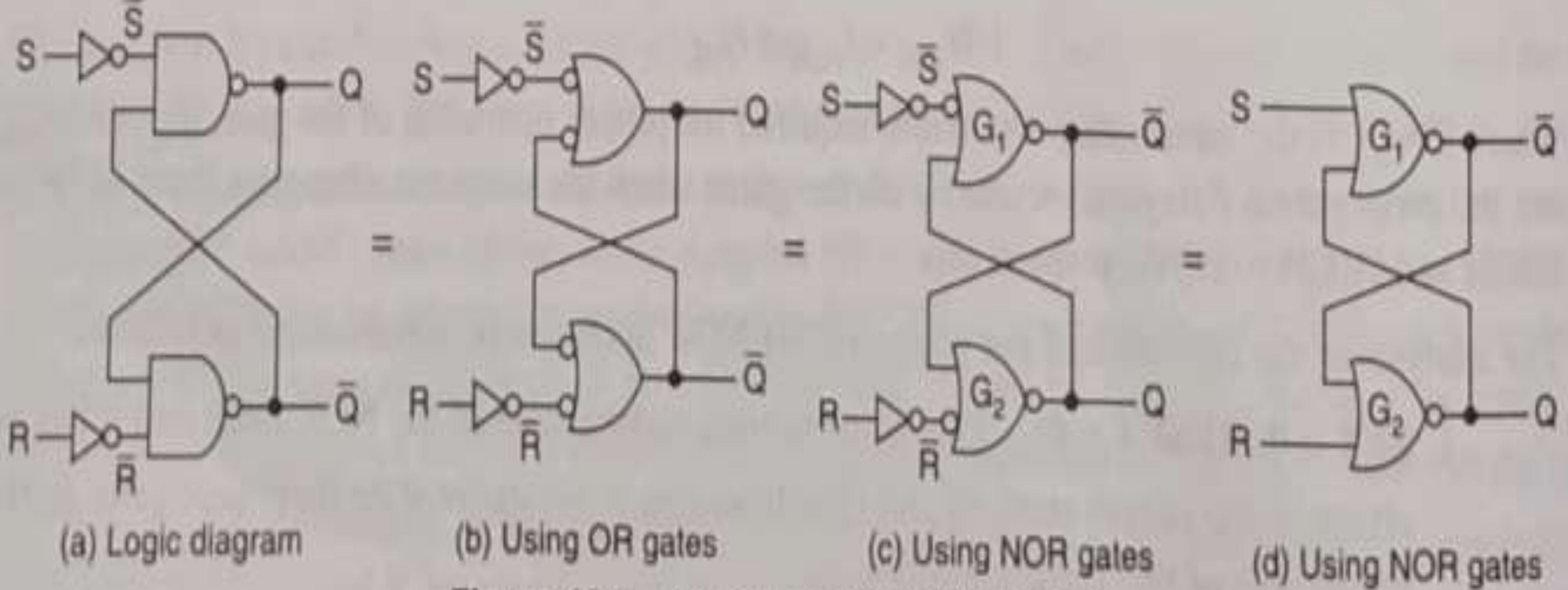
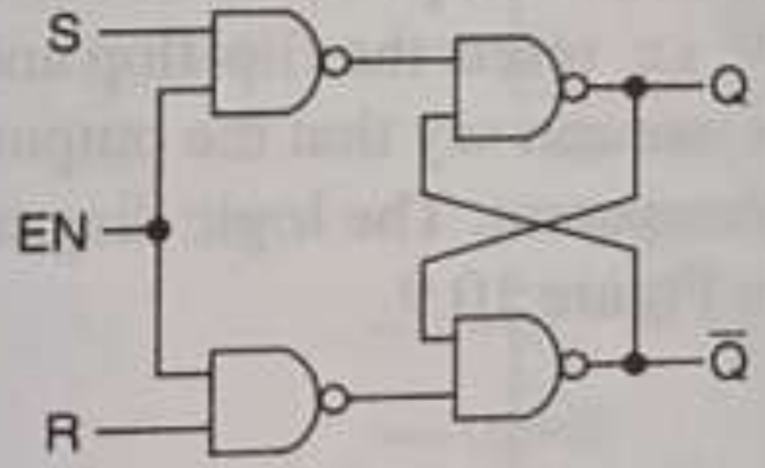
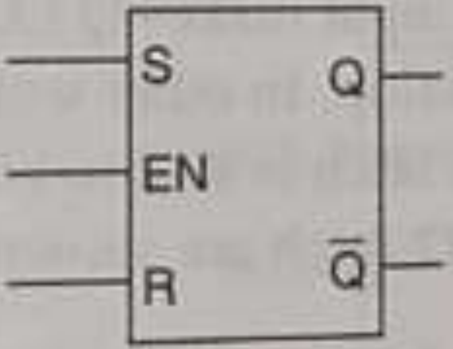


Figure 10.6 An active-HIGH NAND latch.

Gated latches (clocked Flip flops)



(a) Logic diagram



(b) Logic symbol

EN	S	R	Q_n	Q_{n+1}	State
1	0	0	0	0	No change (NC)
1	0	0	1	1	
1	0	1	0	0	Reset
1	0	1	1	0	
1	1	0	0	1	Set
1	1	0	1	1	
1	1	1	0	×	Indeterminate (invalid)
1	1	1	1	×	
0	×	×	0	0	No Change (NC)
0	×	×	1	1	

(c) Truth table

Figure 10.7 A gated S-R latch.

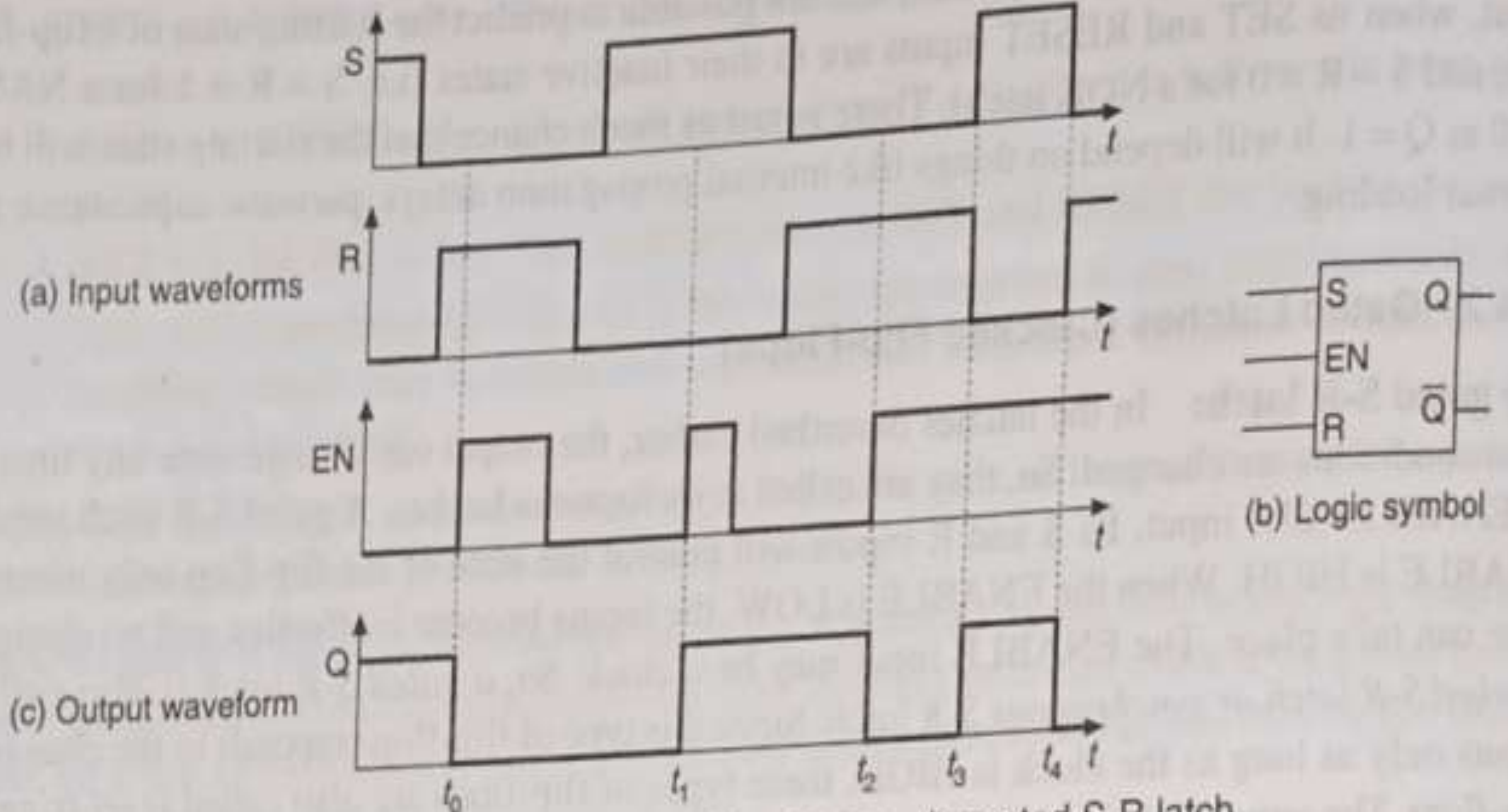
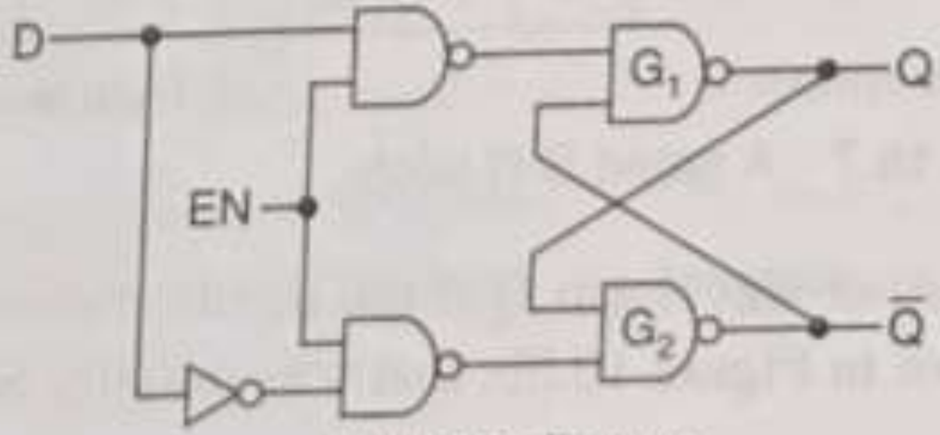
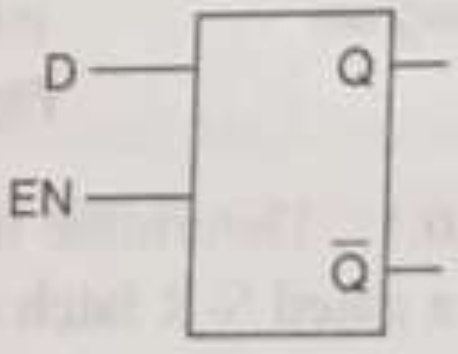


Figure 10.8 Example 10.1: Waveforms—the gated S-R latch.

The gated D latch



(a) Logic diagram



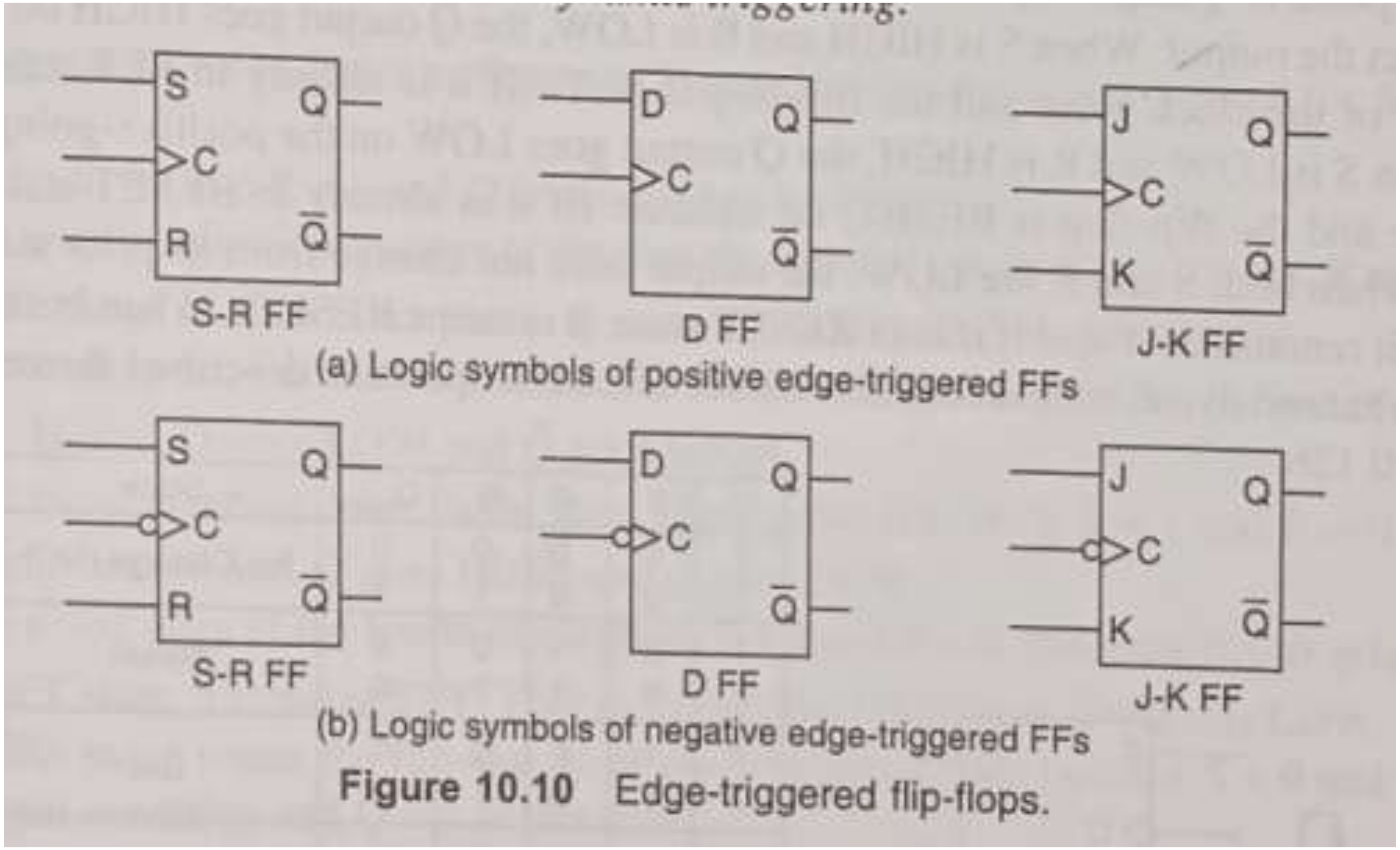
(b) Logic symbol

EN	D	Q_n	Q_{n+1}	State
1	0	0	0	Reset
1	0	1	0	
1	1	0	1	Set
1	1	1	1	
0	×	0	0	No Change (NC)
0	×	1	1	

(c) Truth table

Figure 10.9 A gated D-latch.

Edge triggered flip flops

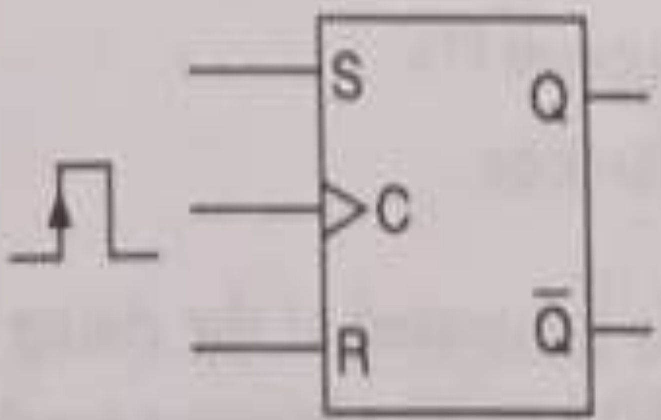


(a) Logic symbols of positive edge-triggered FFs

(b) Logic symbols of negative edge-triggered FFs

Figure 10.10 Edge-triggered flip-flops.

- Positive edge triggered
- Negative edge triggered
- Level triggered



(a) Logic symbol

C	S	R	Q_n	Q_{n+1}	State
↑	0	0	0	0	No Change (NC)
↑	0	0	1	1	
↑	0	1	0	0	Reset
↑	0	1	1	0	
↑	1	0	0	1	Set
↑	1	0	1	1	
↑	1	1	0	×	Indeterminate
↑	1	1	1	×	
0	×	×	0	0	No Change (NC)
0	×	×	1	1	

(b) Truth table

Figure 10.12 Positive edge-triggered S-R flip-flop.

shown in Figure 10.13b. Sketch the output waveforms.

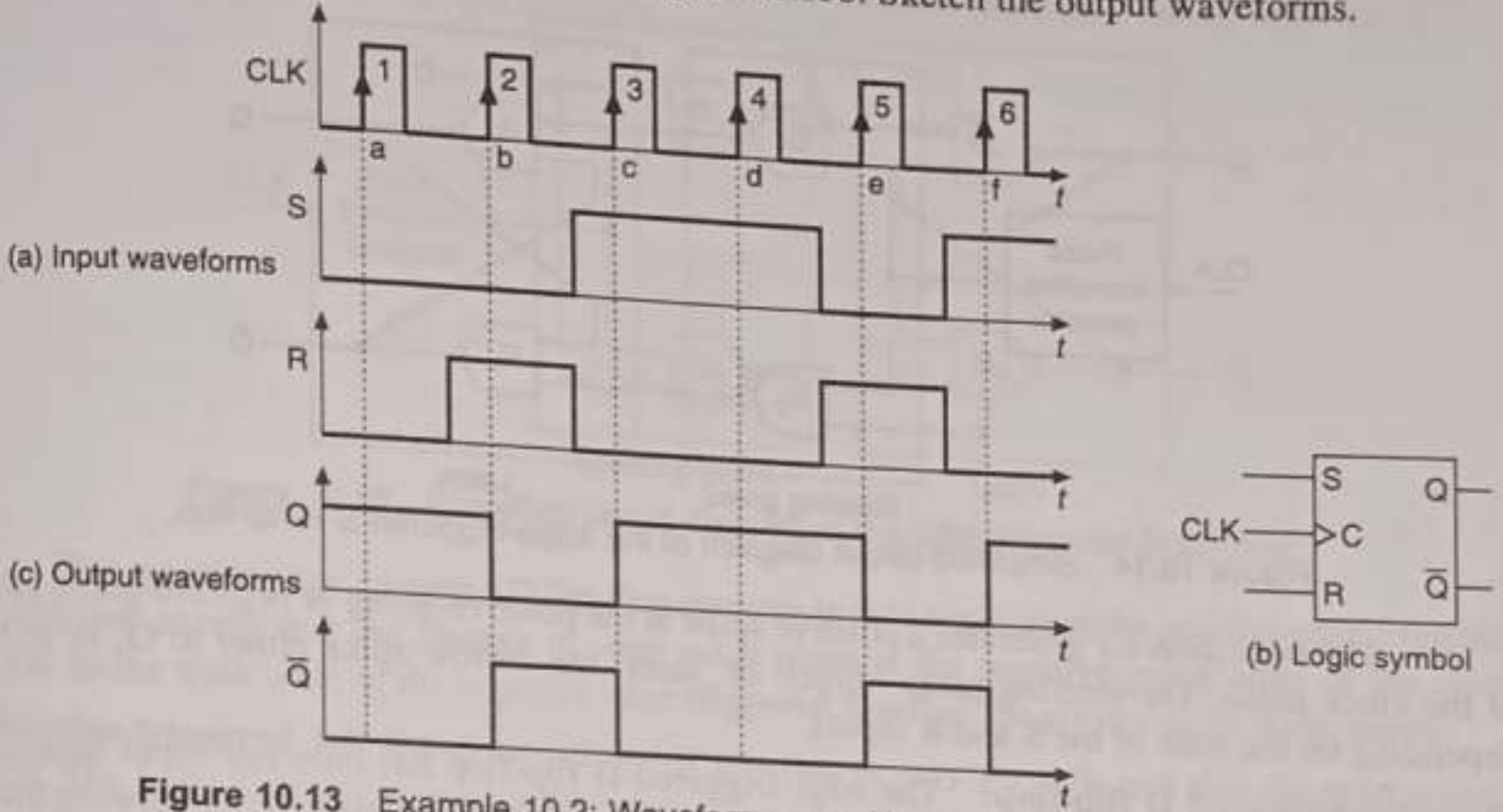


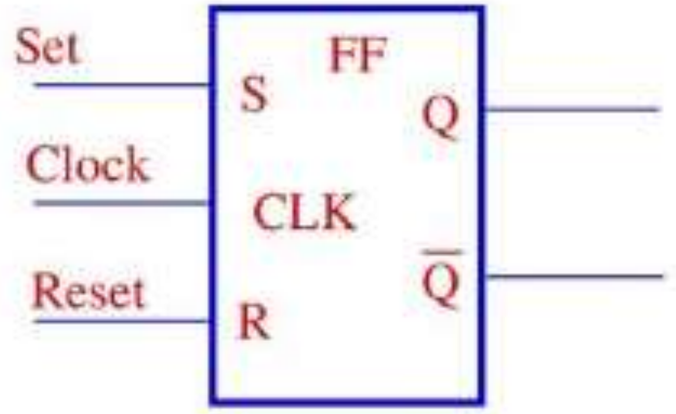
Figure 10.13 Example 10.2: Waveforms—positive edge-triggered S-R flip-flop.

Solution

Types of Flip flops

- There are 4 mainly types of flipflop
- 1. SR flip flop
- 2. JK flip flop
- 3. D flip flop
- 4. T flip flop

CLOCKED R-S FLIP-FLOP

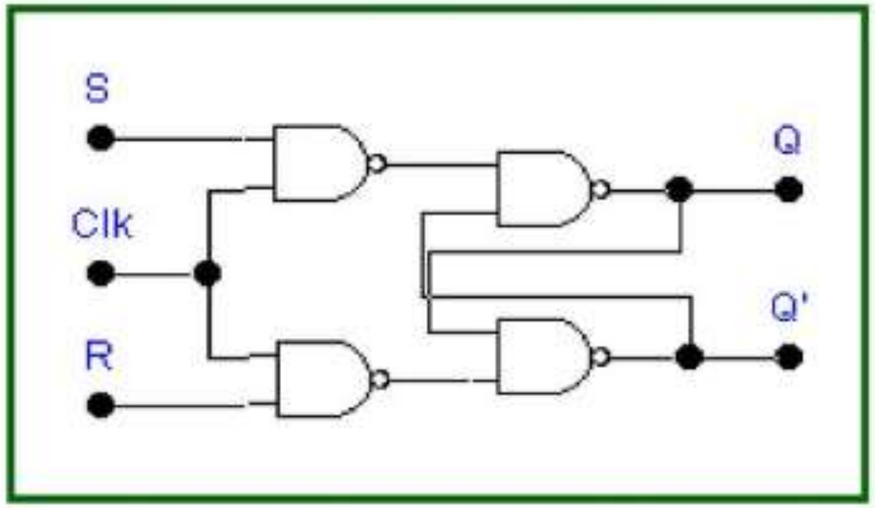


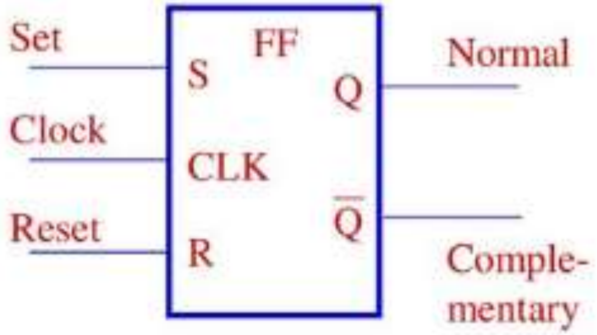
SYNCHRONOUS

Clock signal determines exact time at which any output can change state

CLOCKED R-S FLIP-FLOP

Symbols:



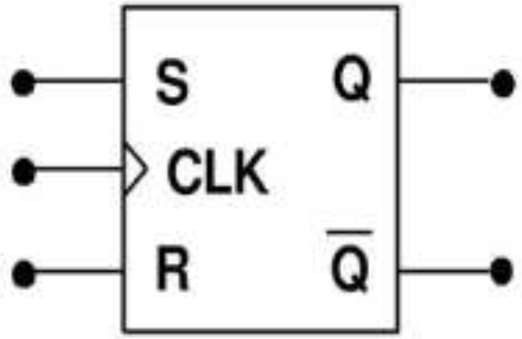


Truth Table:

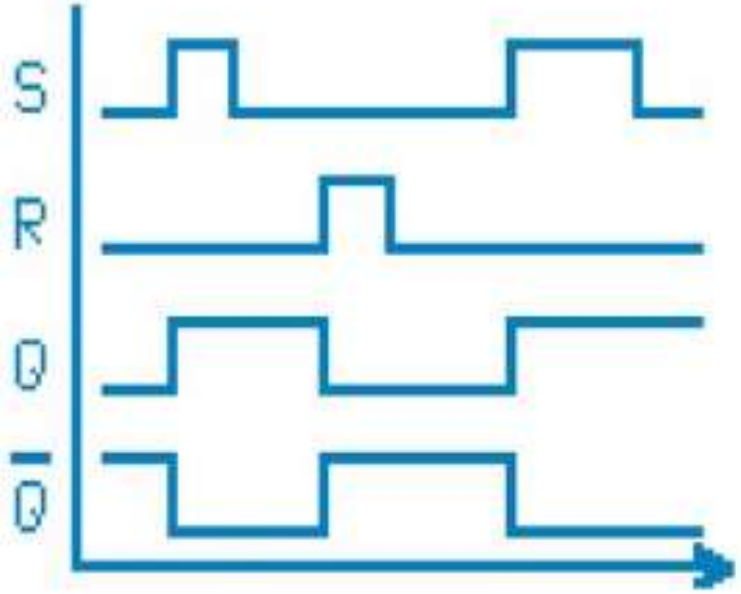
Clk	S	R	Q_n	$\overline{Q_n}$
↑ ↓	0	0	No change	
↑ ↓	0	1	0	1
↑ ↓	1	0	1	0
↑ ↓	1	1	Race	Race

POSITIVE EDGE TRIGGERED R-S FLIP-FLOP

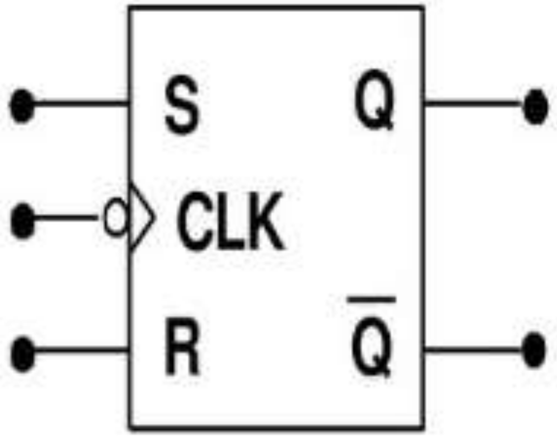
TIMING DIAGRAMS



CLK	R	S	Q
▲	0	0	NO CHG
▲	0	1	SET
▲	1	0	RESET
▲	1	1	ILLEGAL

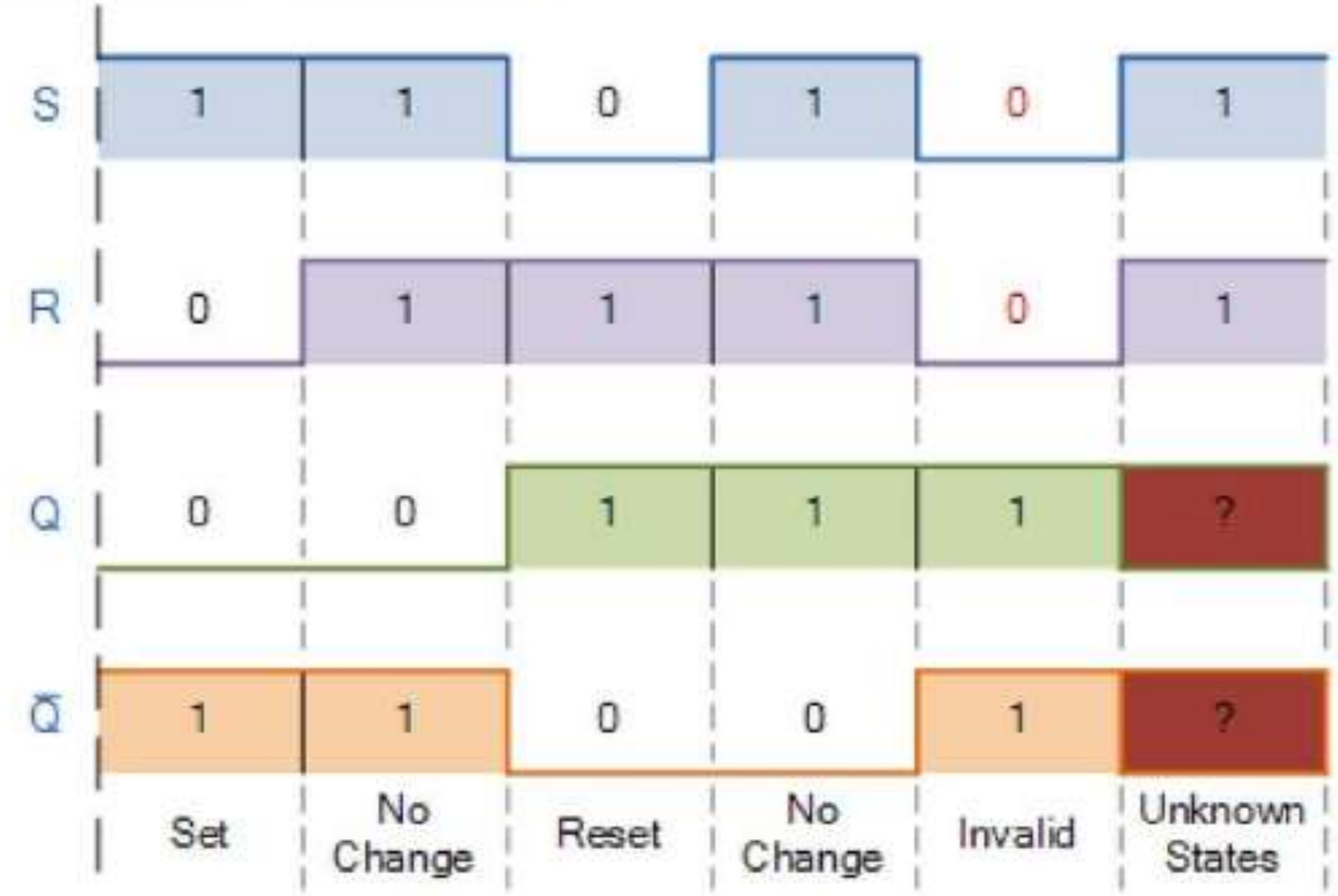


NEGATIVE EDGE TRIGGERED R-S FLIP-FLOP

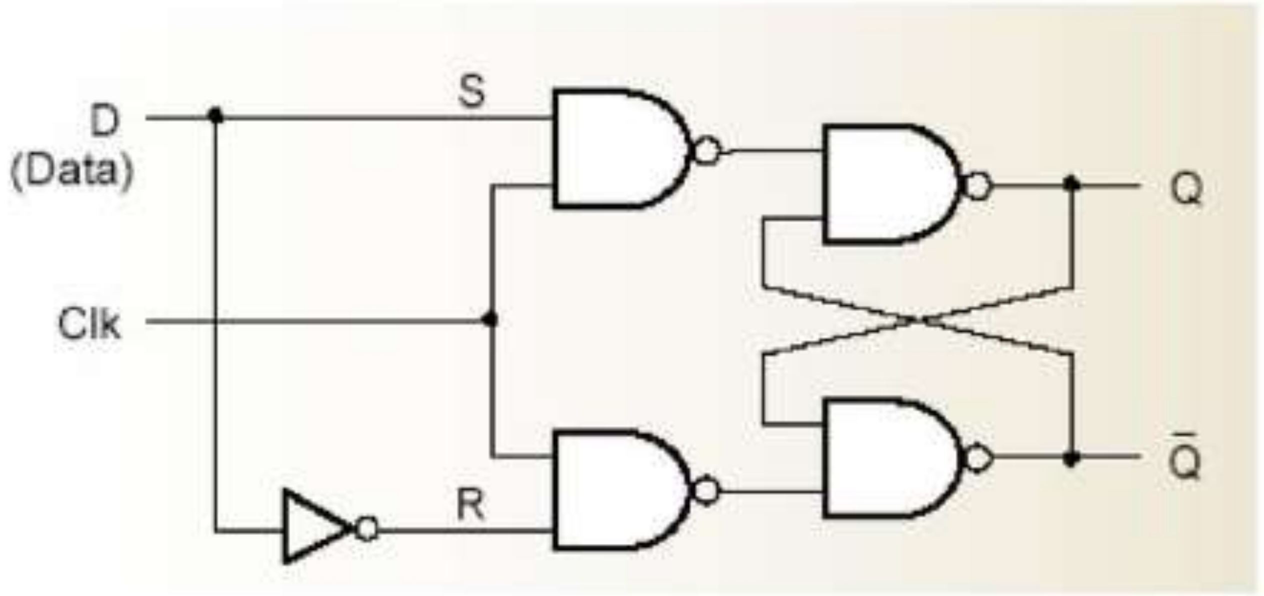


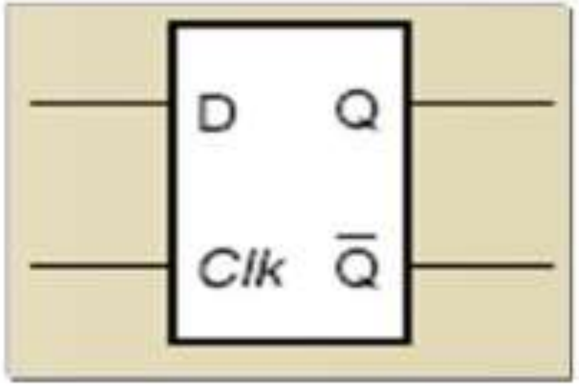
CLK	R	S	Q
▼	0	0	NO CHG
▼	0	1	SET
▼	1	0	RESET
▼	1	1	ILLEGAL

S-R Flip-flop Switching Diagram



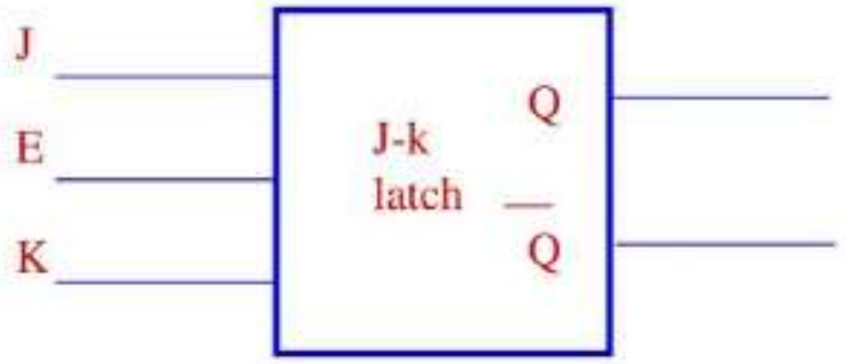
D Flip-Flop

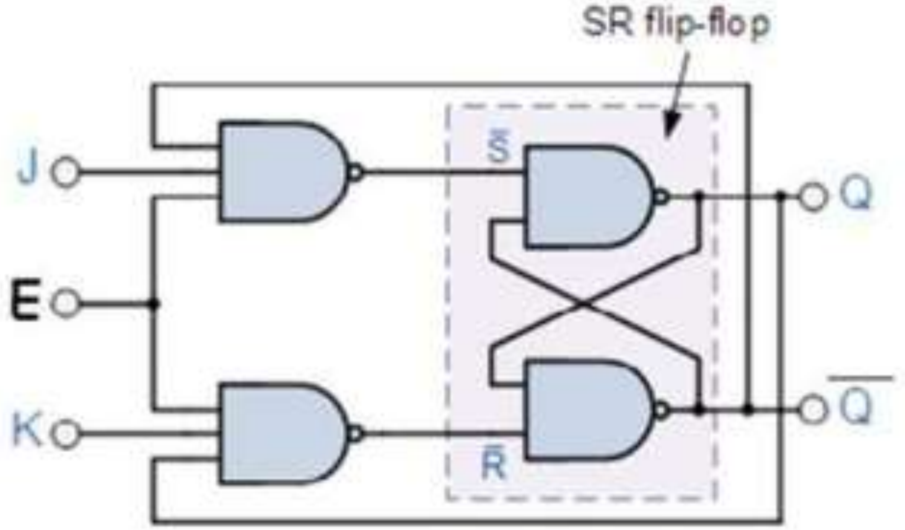




clk	D	Qn+1	$\overline{Qn+1}$
↑	0	0	1
↑	1	1	0

J-k latch





E	S	R	Q_n	$\overline{Q_n}$
1	0	0	No change	
1	0	1	0	1
1	1	0	1	0
1	1	1	$\overline{Q_n}$	Q_n

Toggle
Race around condition

Race around condition

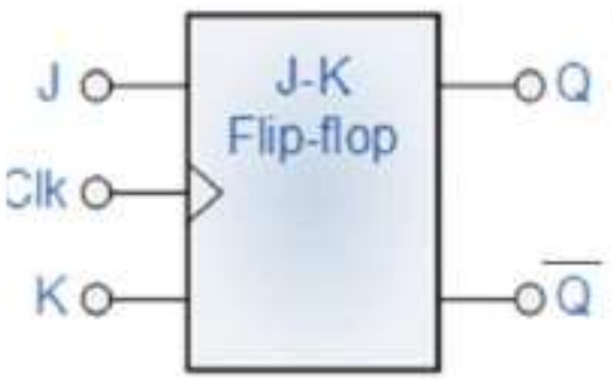
This condition occur when j=k=1 i.e when the latch is in toggle mode.

This can be avoided by

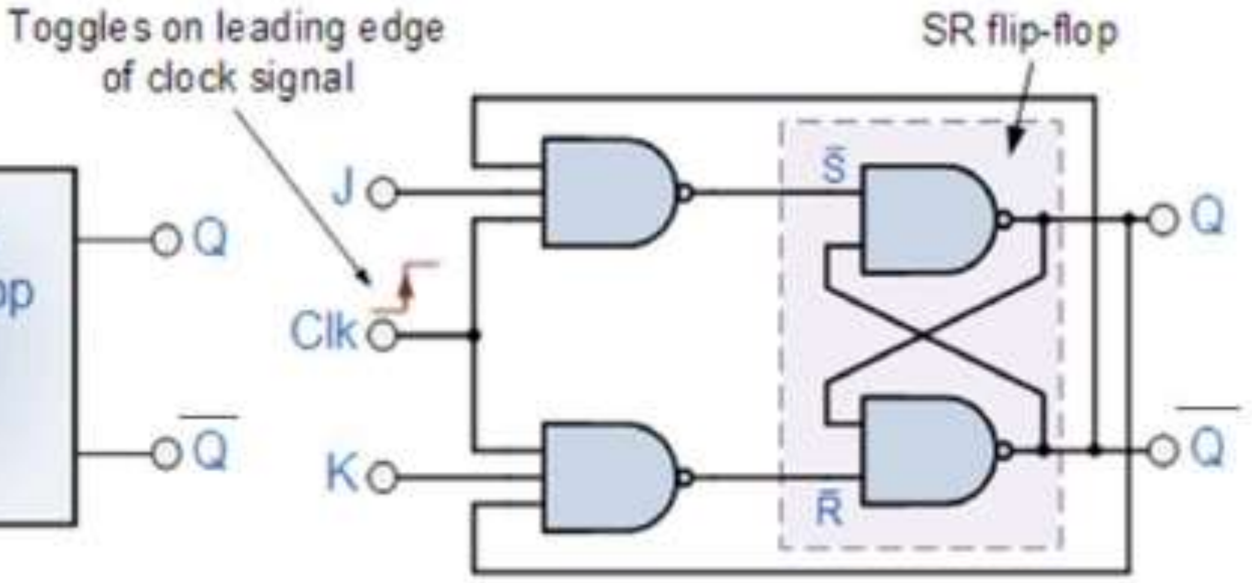
- Using edge triggering J-k flip-flop
- Using master slave flip-flop

- For the J K flip flop if J=K=1
- If the width of the clock pulse tp is too long, the state of the FF will keep on changing from 0 to 1 , 1 to 0, 0 to 1 and so on, and at the end of clock pulse , its state will be uncertain.
- This phenomenon is called the **race around condition.**

J-K flip-flop



Symbol

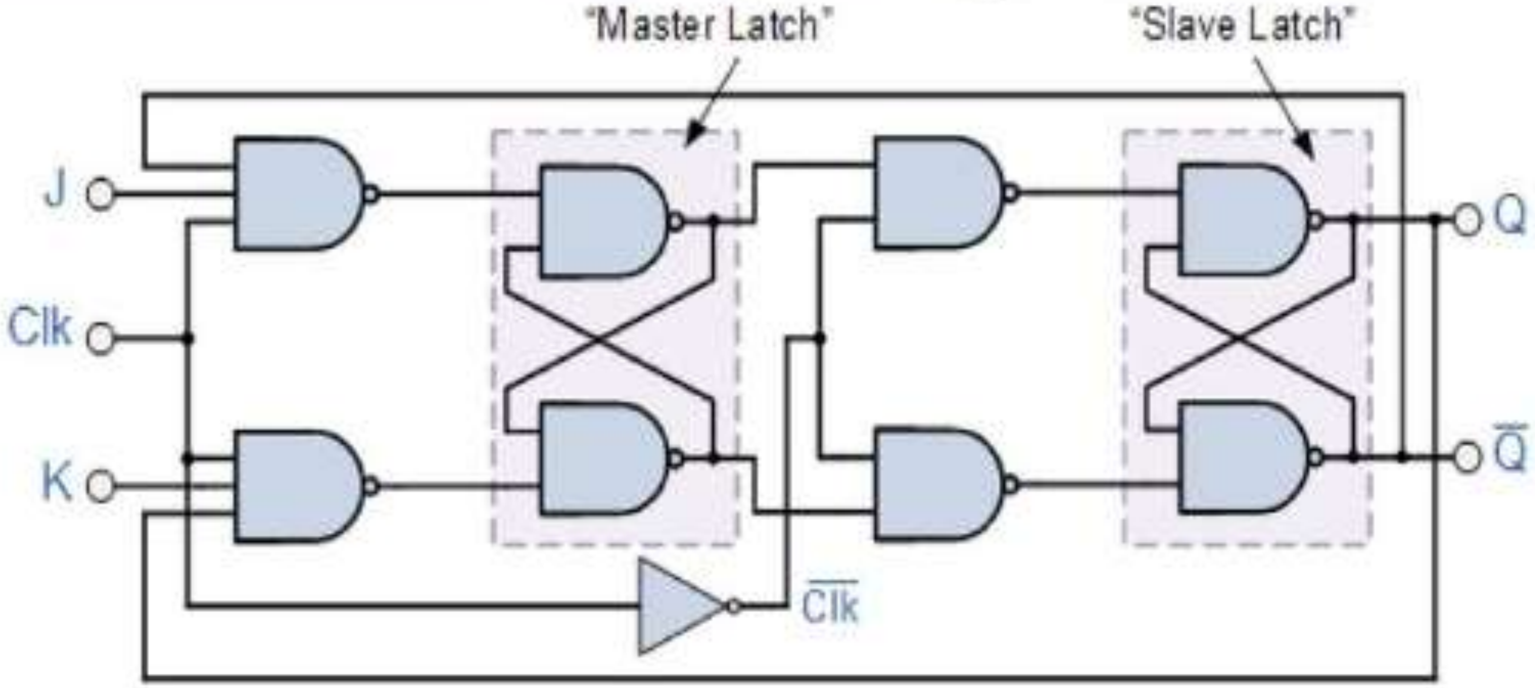


Circuit

Clk	J	k	Q_n	$\overline{Q_n}$
↑	0	0	No change	
↑	0	1	0	1
↑	1	0	1	0
↑	1	1	$\overline{Q}$	Q

Toggle

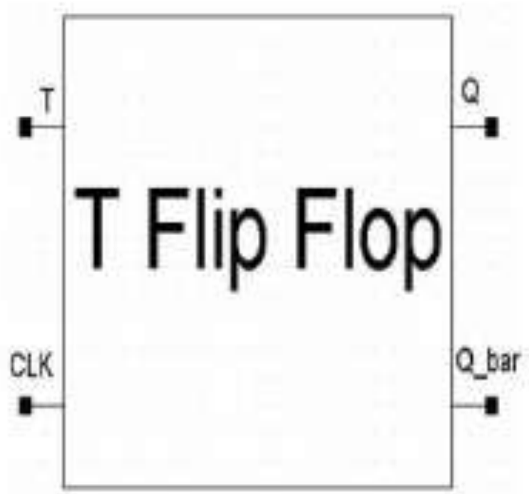
Master slave flip-flop

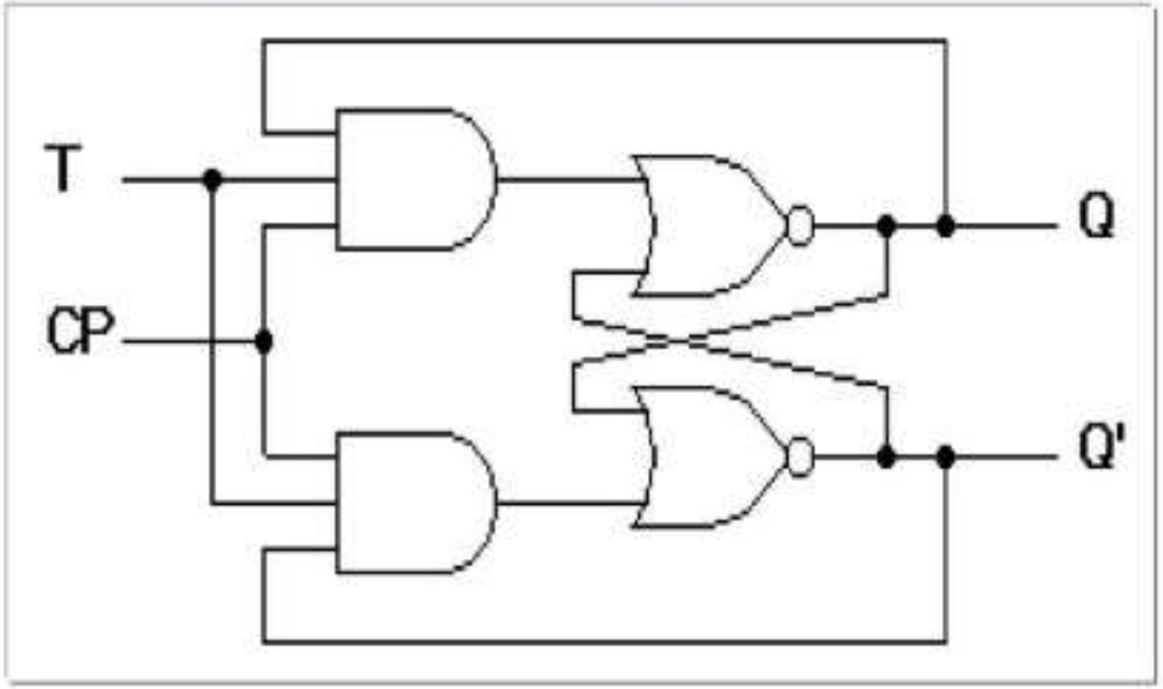


Clk	j	k	Q_n	$\overline{Q_n}$
↑	0	0	No change	
↑	0	1	0	1
↑	1	0	1	0
↑	1	1	$\overline{Q_n}$	Q_n

Toggle

T flip flop





clk	T	Q	$\overline{Q}$	comments
$\uparrow$	0	Q	$\overline{Q}$	No change
$\uparrow$	1	$\overline{Q}$	Q	toggle