Digital Electronics

(Second Year B. Tech program in Computer Engineering)

Combinational and Sequential Logic Circuit

Combinational Circuit –

In this output depends only upon present input.

Speed is fast.

It is designed easy.

There is no feedback between input and output.

This is time independent.

Elementary building blocks: Logic gates

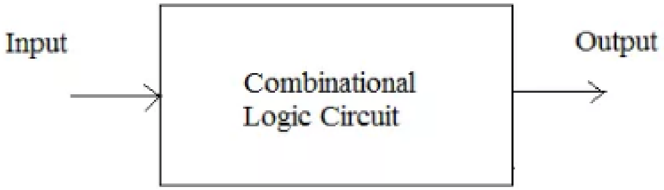
Used for arithmetic as well as boolean operations.

Combinational circuits don’t have capability to store any state.

As combinational circuits don’t have clock, they don’t require triggering.

These circuits do not have any memory element.

Examples – Encoder, Decoder, Multiplexer, Demultiplxer, Parity Checker & Generator



Combinational and Sequential Logic Circuit

Sequential Circuit –

1.In this output depends upon present as well as past input.

2.Speed is slow.

3.It is designed tough as compared to combinational circuits.

4.There exists a feedback path between input and output.

5.This is time dependent.

6.Elementary building blocks: Flip-flops

7.Mainly used for storing data.

8.Sequential circuits have capability to store any state or to retain earlier state.

9.As sequential circuits are clock dependent they need triggering.

10.These circuits have memory element.

Examples – Flip-flops, Counters

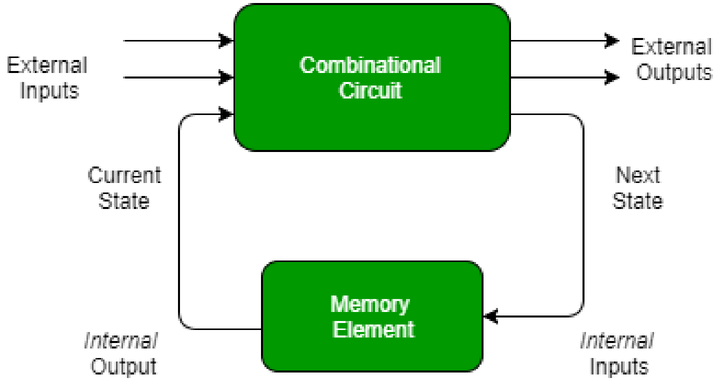
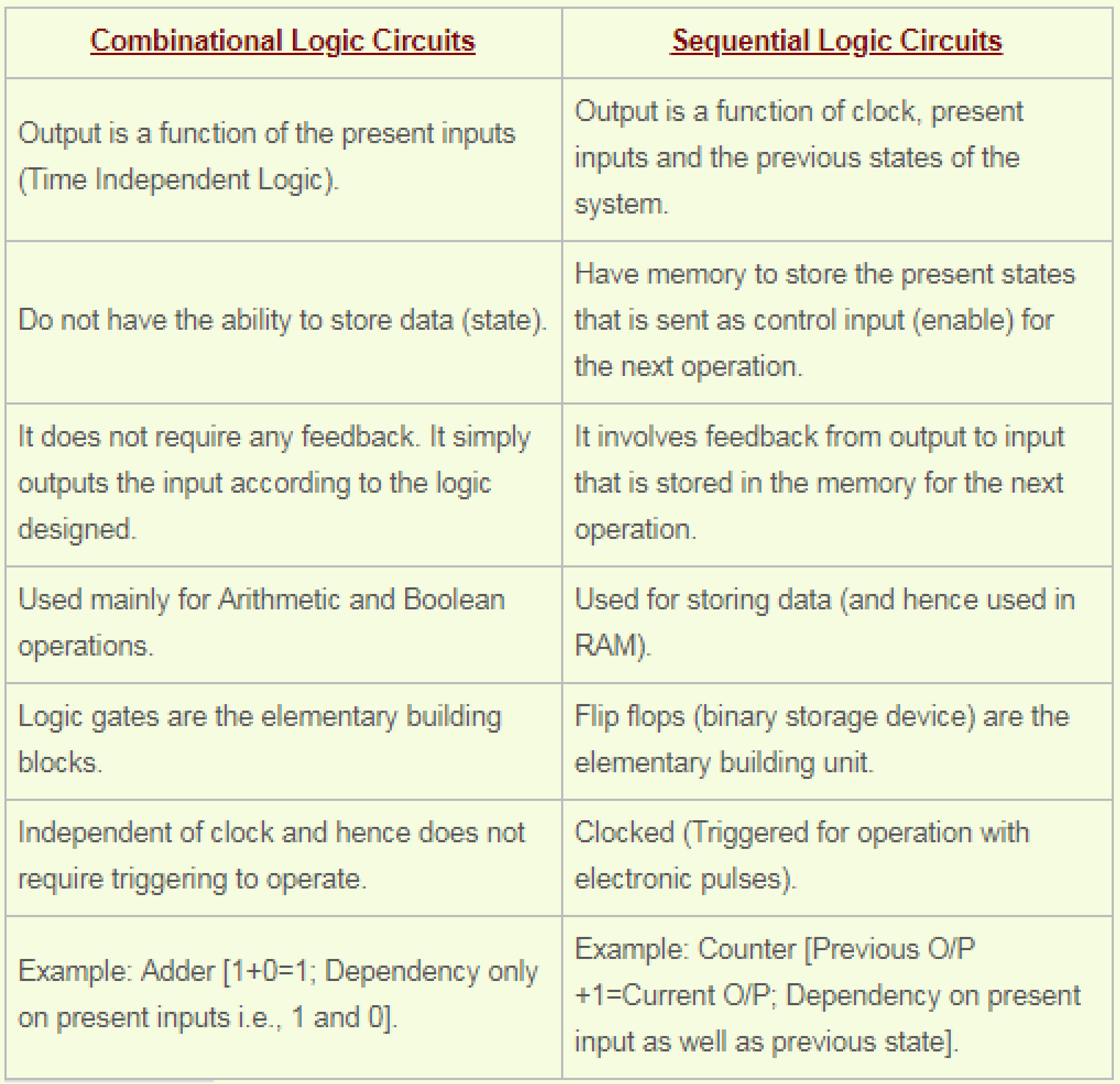


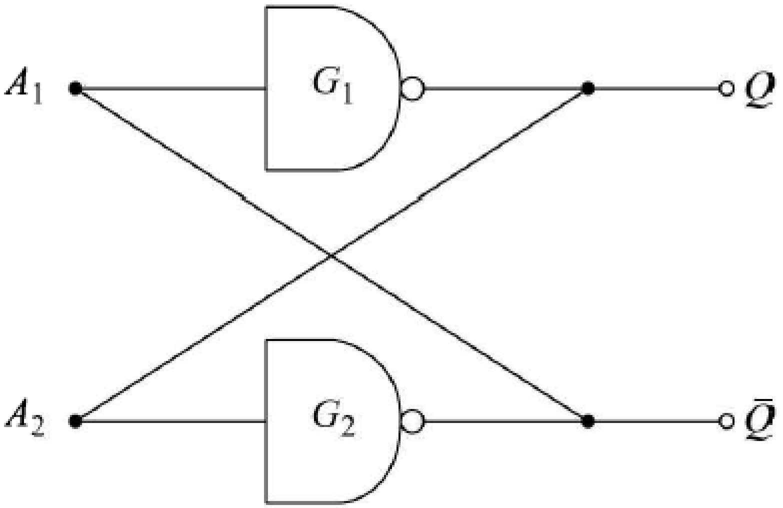
Figure: Sequential Circuit

Combinational and Sequential Logic Circuit



Combinational Logic Circuits	Sequential Logic Circuits
Output is a function of the present inputs (Time Independent Logic).	Output is a function of clock, present inputs and the previous states of the system.
Do not have the ability to store data (state).	Have memory to store the present states that is sent as control input (enable) for the next operation.
It does not require any feedback. It simply outputs the input according to the logic designed.	It involves feedback from output to input that is stored in the memory for the next operation.
Used mainly for Arithmetic and Boolean operations.	Used for storing data (and hence used in RAM).
Logic gates are the elementary building blocks.	Flip flops (binary storage device) are the elementary building unit.
Independent of clock and hence does not require triggering to operate.	Clocked (Triggered for operation with electronic pulses).
Example: Adder [1+0=1; Dependency only on present inputs i.e., 1 and 0].	Example: Counter [Previous O/P +1=Current O/P; Dependency on present input as well as previous state].

1 Bit memory cell



Cross-coupled Inverters as a Memory Element

Properties:

1. Both outputs are complement of each other.

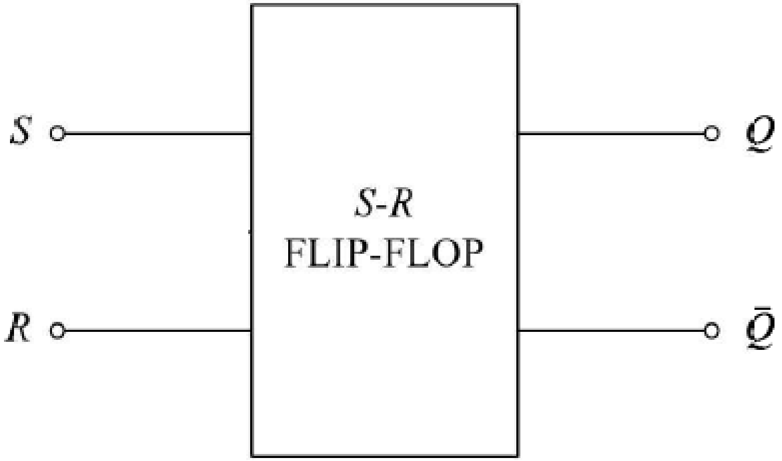
2. Circuit has two stable states i.e. Set state/1 state/ Preset State and Reset State/0 state/Clear state.

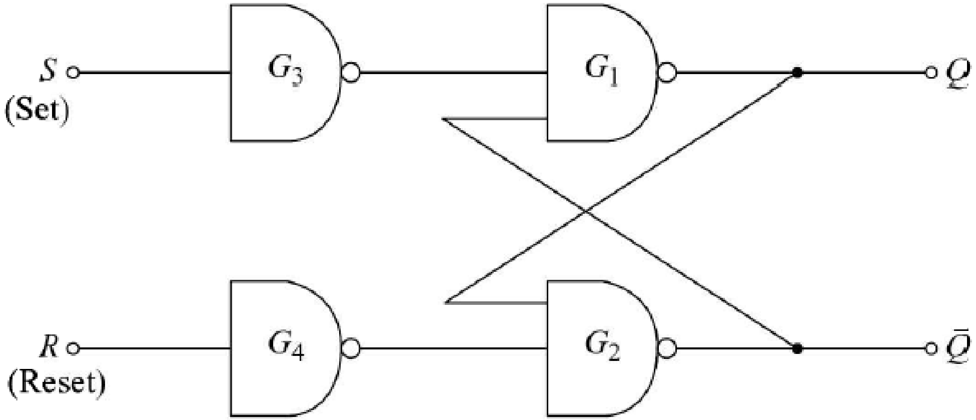
3. If the circuit is in 1 state, it remains in this state and similarly for 0 state

Types of Flip Flop

- S-R Flip Flop
- J-K Flip Flop
- T Flip Flop
- D Flip Flop

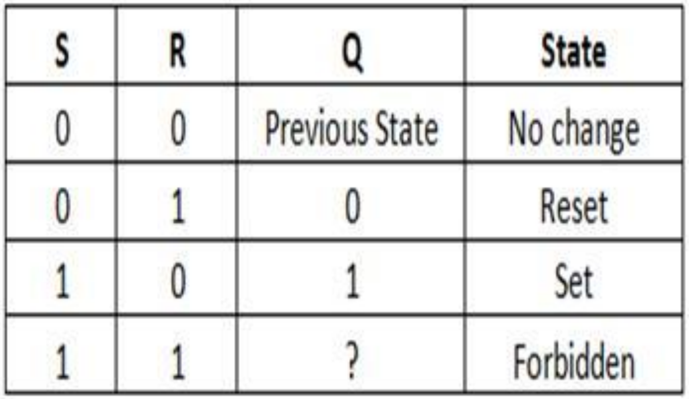
S-R Flip Flop





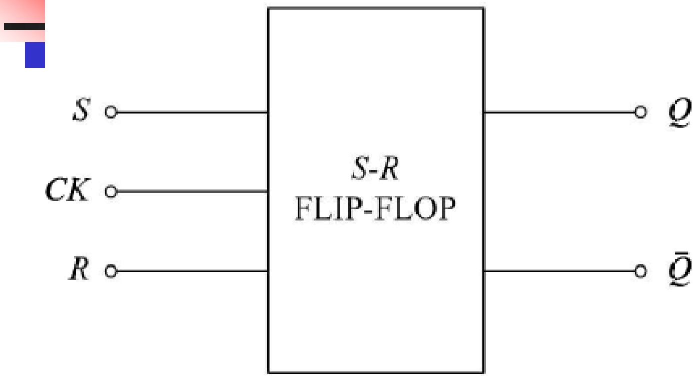
S R Flip Flop Circuit

Truth Table

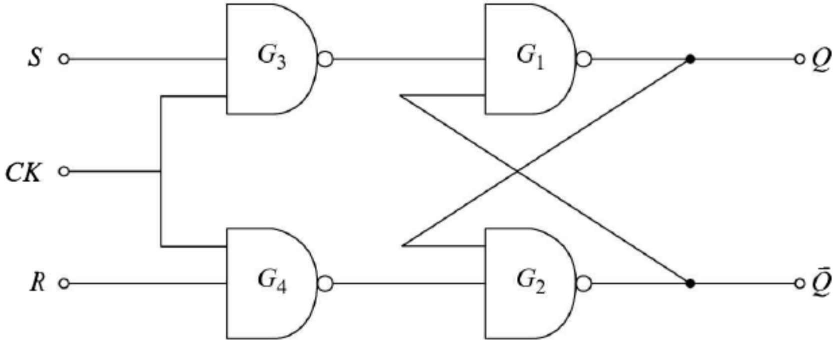


S	R	Q	State
0	0	Previous State	No change
0	1	0	Reset
1	0	1	Set
1	1	?	Forbidden

Clocked S-R Flip Flop



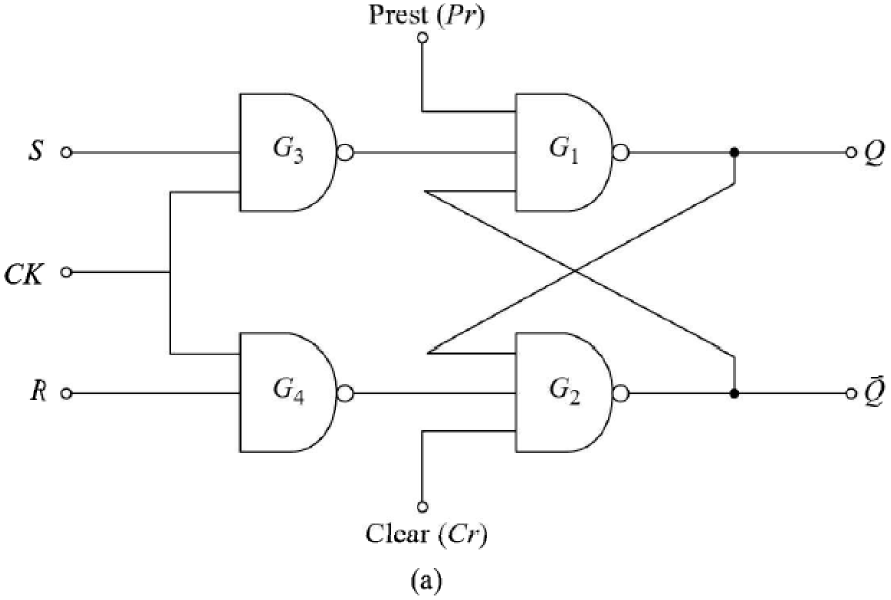
Logic Symbol of Clocked S–R FLIP-FLOP



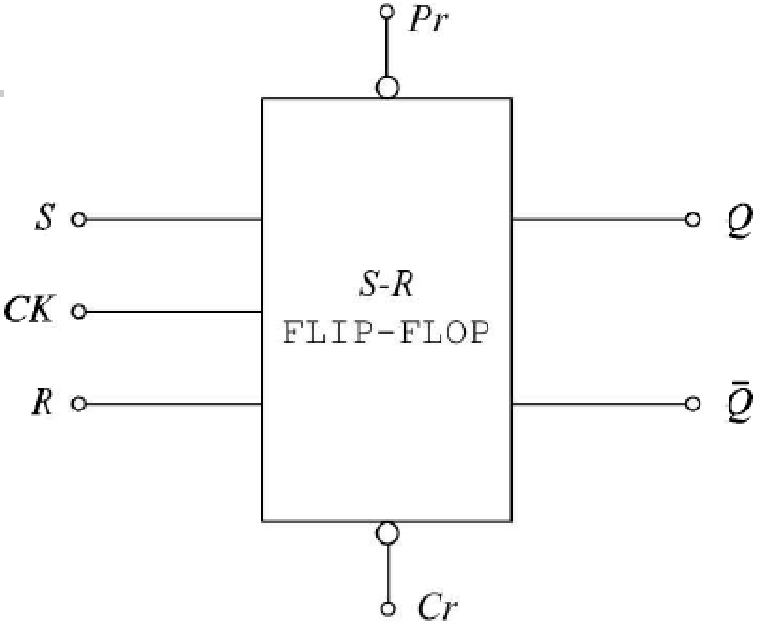
A Clocked S–R FLIP-FLOP

CLK	S	R	Q	Q	State
1	0	0	NC	NC	
1	0	1	0	1	Reset
1	1	0	1	0	Set
1	1	1	X	X	Uncertain
0	0	0	X	X	
0	0	1	X	X	
0	1	0	X	X	
0	1	1	X	X	

Circuit of S-R Flip-Flop with Preset & Clear I/P



(a) An S–R FLIP-FLOP *with Preset and Clear,*

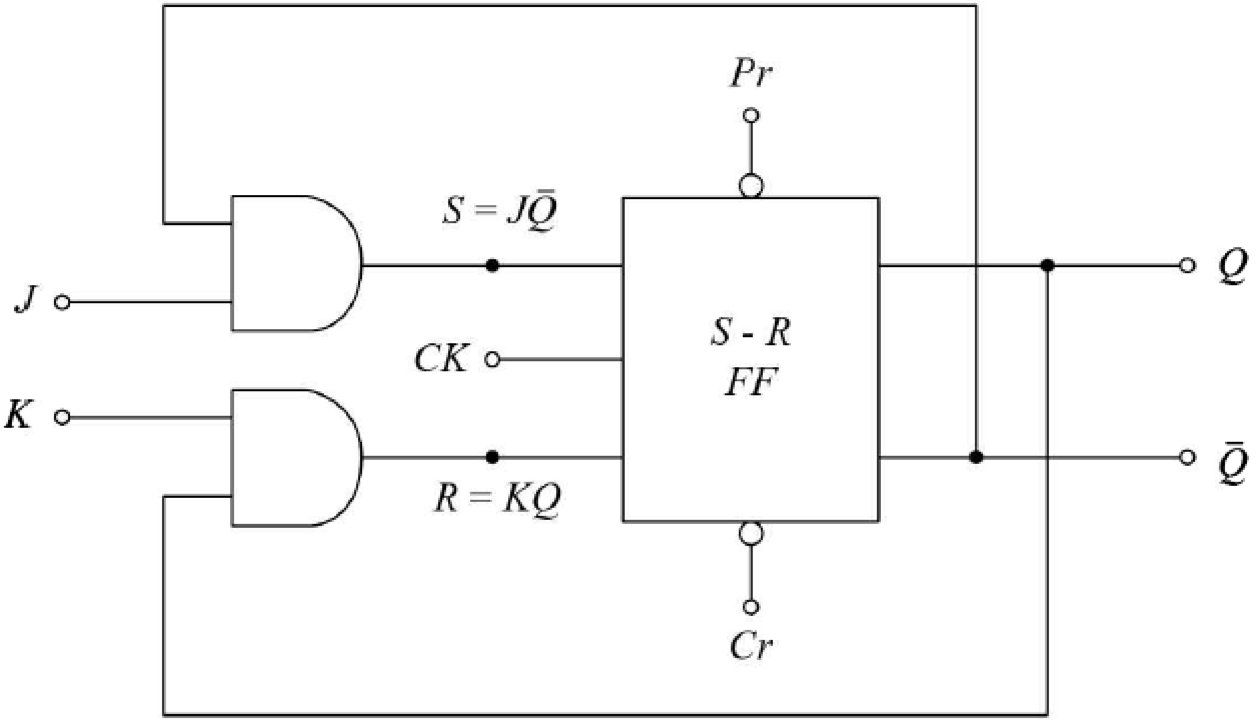


(b) Its Logic Symbol

Summary of Operation of S–R FLIP-FLOP

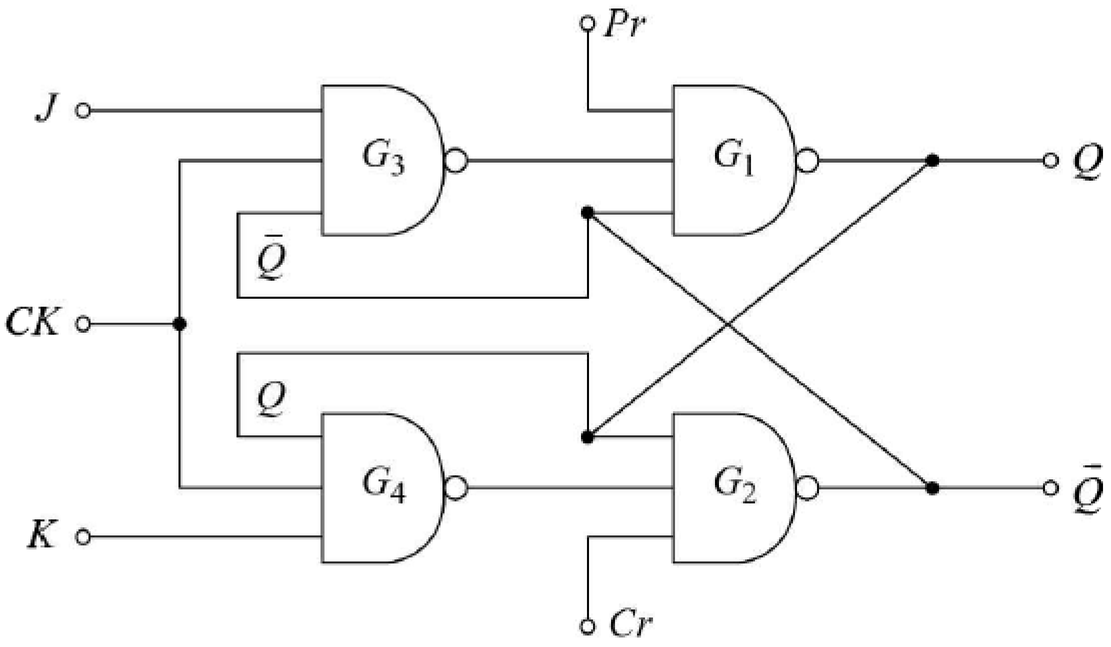
Inputs			Output	Operation performed
CK	***Cr***	***Pr***	***Q***	
1	1	1	Q_{n+1}(Table 7.1)	Normal FLIP-FLOP
0	0	1	0	Clear
0	1	0	1	Preset

Conversion of S-R to J-K Flip-Flop

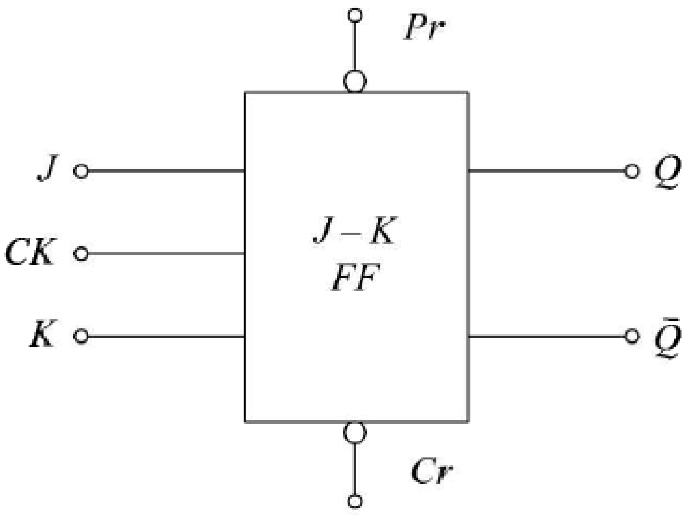


An S–R FLIP-FLOP ***Converted into J–K*** FLIP-FLOP

Circuit of J-K (Jack Kilby) Flip-Flop

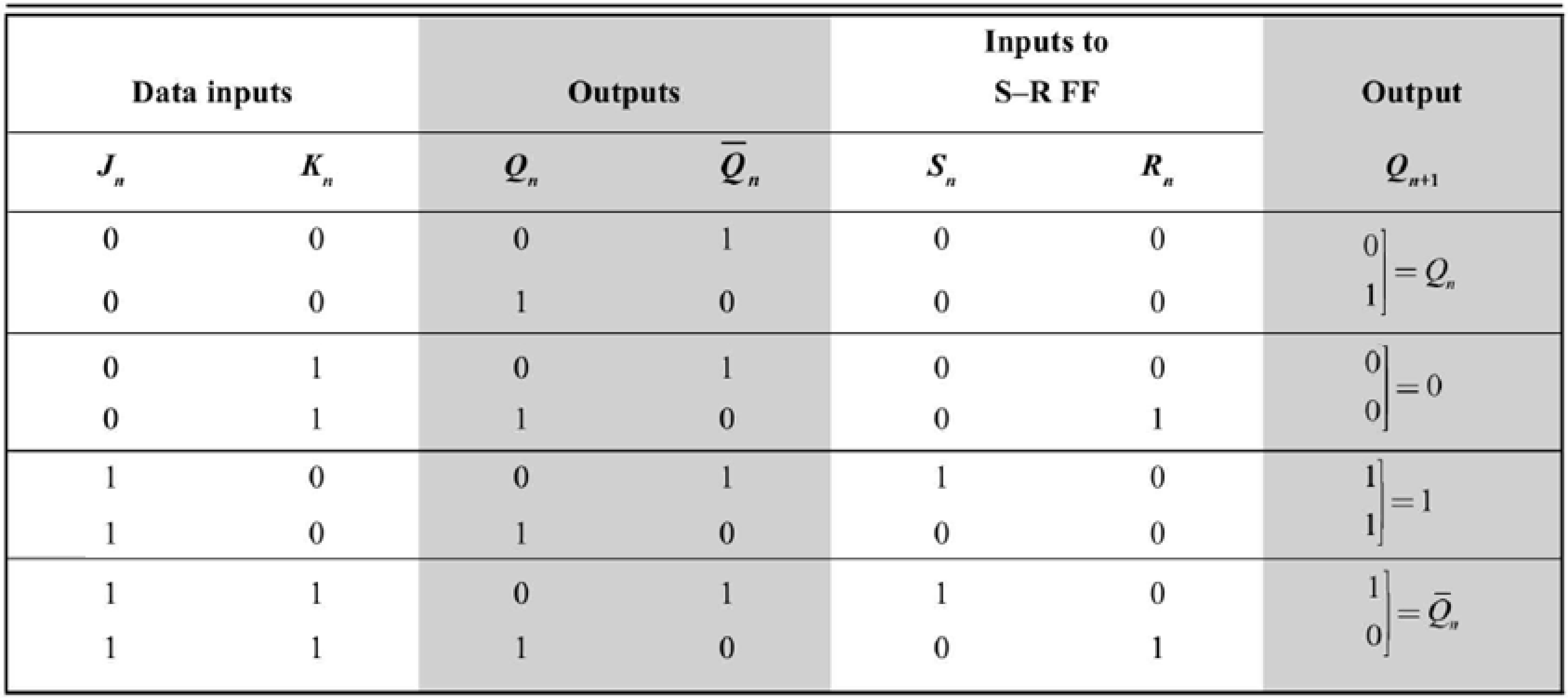


A J–K FLIP–FLOP ***Using*** NAND ***Gates***



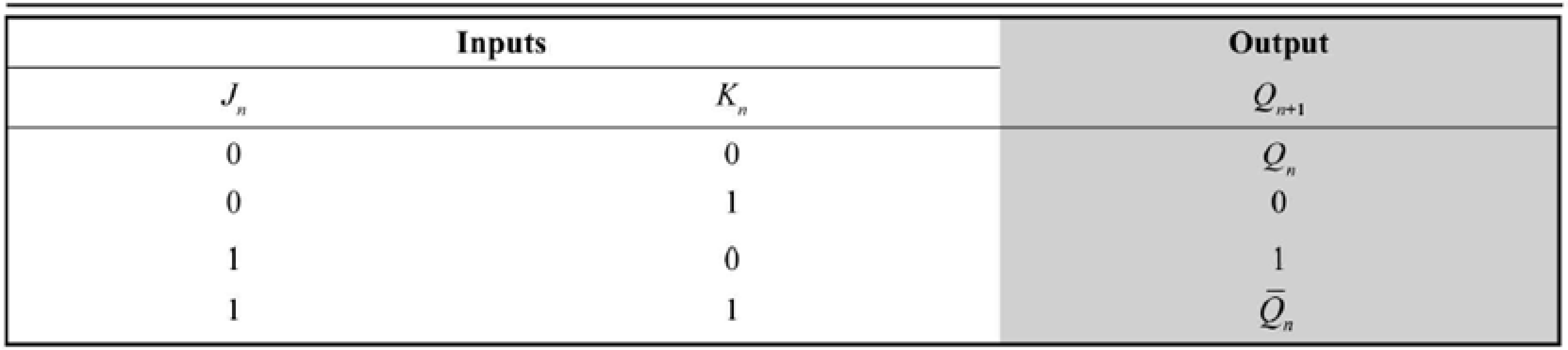
Logic Symbols of J–K FLIP–FLOP

Truth Table of J-K Flip-Flop



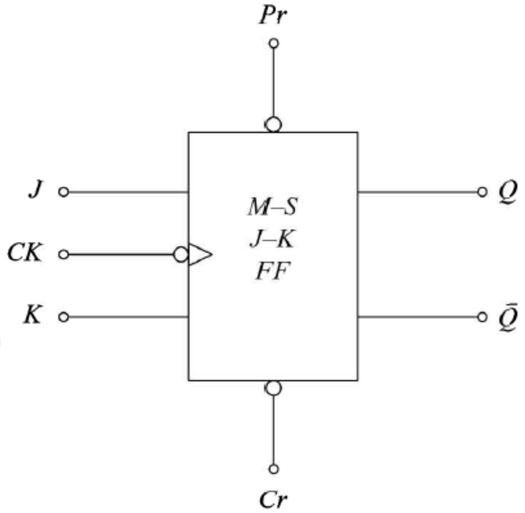
Data inputs		Outputs		Inputs to S–R FF		Output
J_n	K_n	Q_n	$\bar{Q}_n$	S_n	R_n	Q_{n+1}
0	0	0	1	0	0	$\left.\begin{matrix}0\\1\end{matrix}\right] = Q_n$
0	0	1	0	0	0	
0	1	0	1	0	0	$\left.\begin{matrix}0\\0\end{matrix}\right] = 0$
0	1	1	0	0	1	
1	0	0	1	1	0	$\left.\begin{matrix}1\\1\end{matrix}\right] = 1$
1	0	1	0	0	0	
1	1	0	1	1	0	$\left.\begin{matrix}1\\0\end{matrix}\right] = \bar{Q}_n$
1	1	1	0	0	1	

Truth Table of J–K FLIP-FLOP

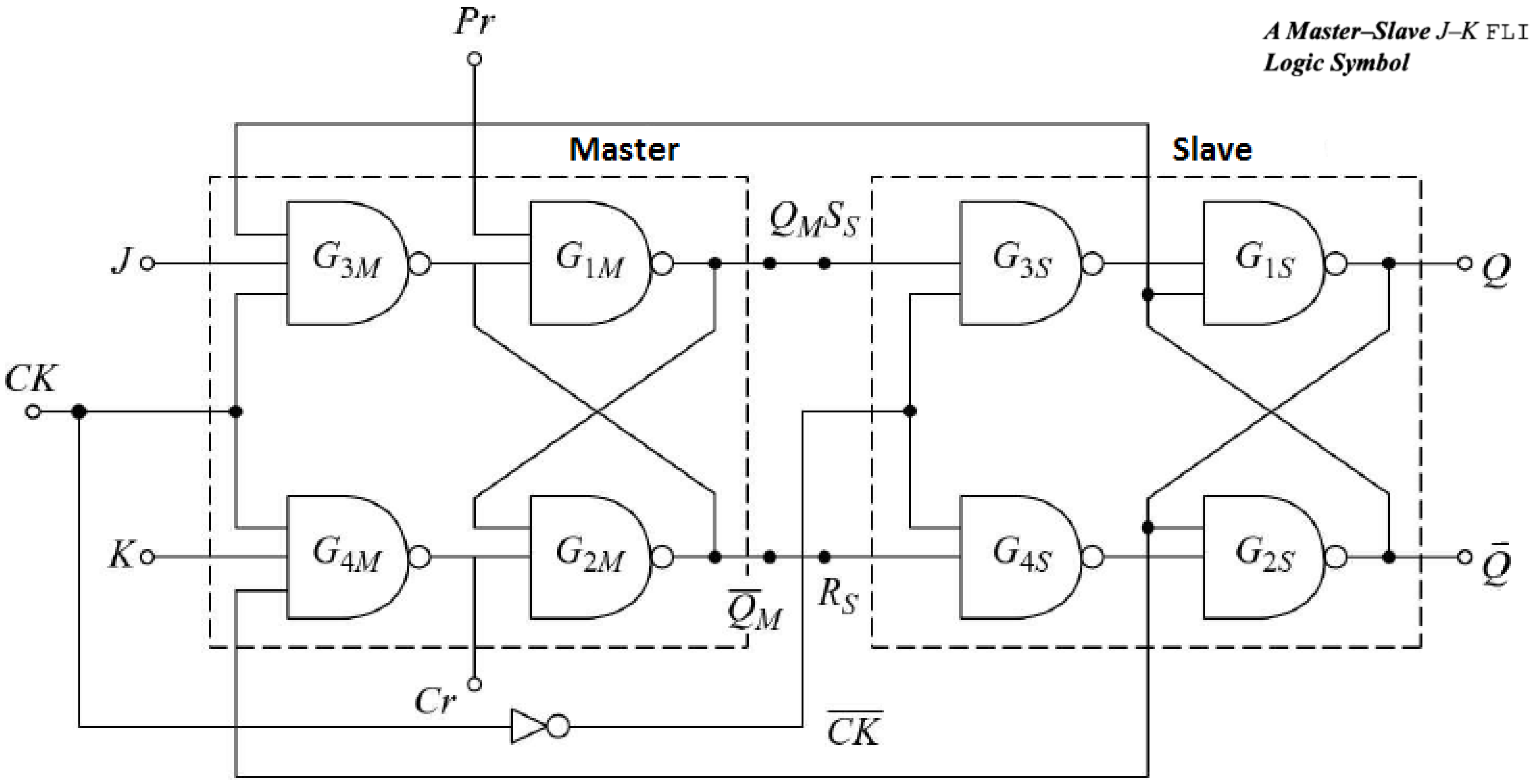


Inputs		Output
J_n	K_n	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	$\bar{Q}_n$

Master Slave J-K Flip-Flop

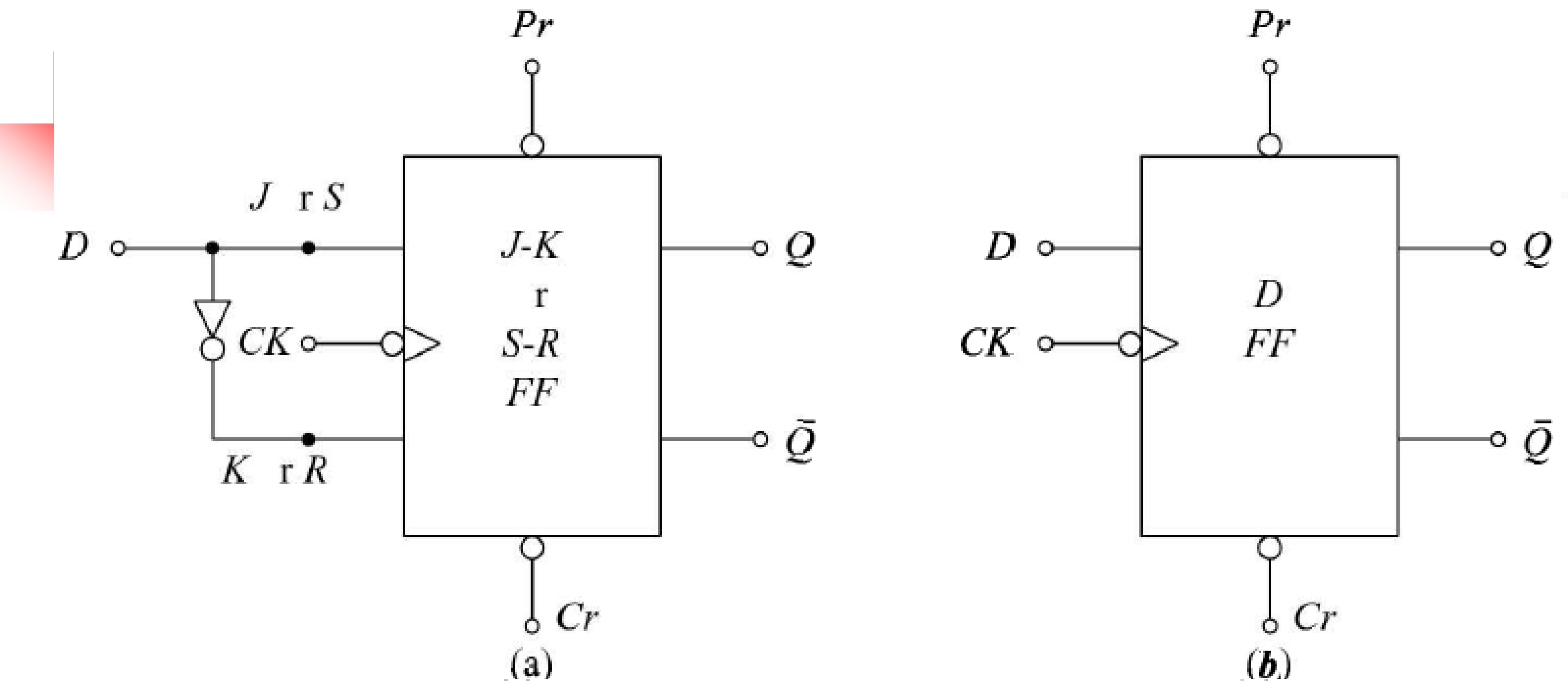


A Master–Slave ***J–K*** FLIP-FLOP
Logic Symbol



A Master–Slave ***J–K*** FLIP-FLOP

D (Delay) Flip-Flop



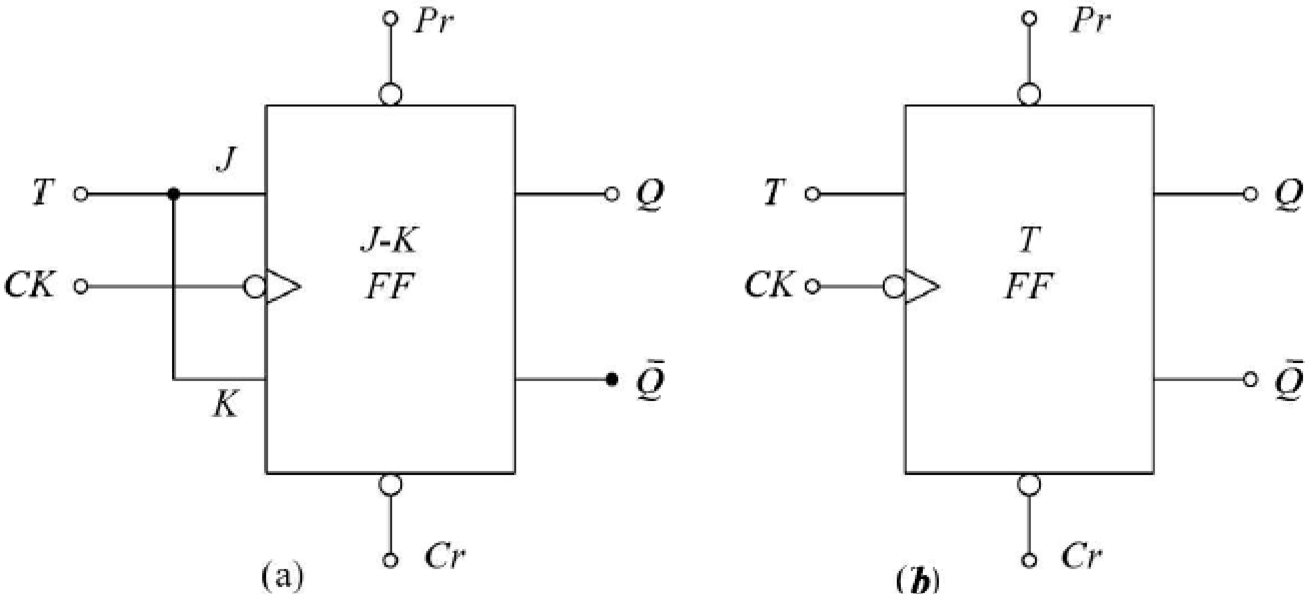
(a) A J–K or S–R* FLIP-FLOP *Converted into a D-type* FLIP-FLOP *(b) its Logic Symbol

***Truth Table of a D-type* FLIP-FLOP**

Input	**Output**
D_n	Q_{n+1}
0	0
1	1

If we use only middle 2 rows of truth table of S-R or JK flip flop , we obtain D Flip-Flop

T (Toggle) Flip-Flop



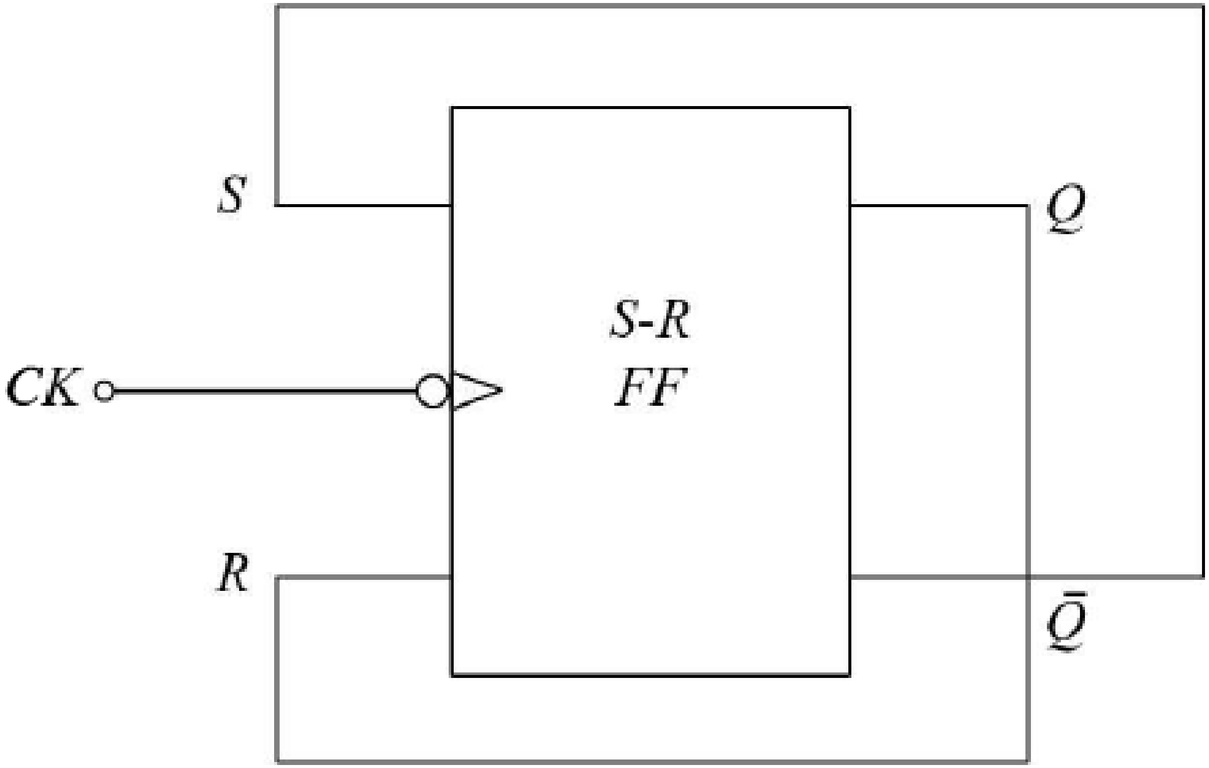
(a) A J–K* FLIP-FLOP *Converted into a T-type* FLIP-FLOP *(b) its Logic Symbol

Truth Table of T-type FLIP-FLOP

Input	Output
T_n	Q_{n+1}
0	Q_n
1	$\bar{Q}_n$

In JK flip flop , If J=K, the resulting Flip-Flop is referred to as T Flip-Flop

Conversion of S-R into T (Toggle) Flip-Flop



An S–R* FLIP-FLOP *as a Toggle Switch

Applications of Flip-Flop

1. Bounce Eliminating Switch
2. Latch
3. Registers
4. Counters
5. Memory

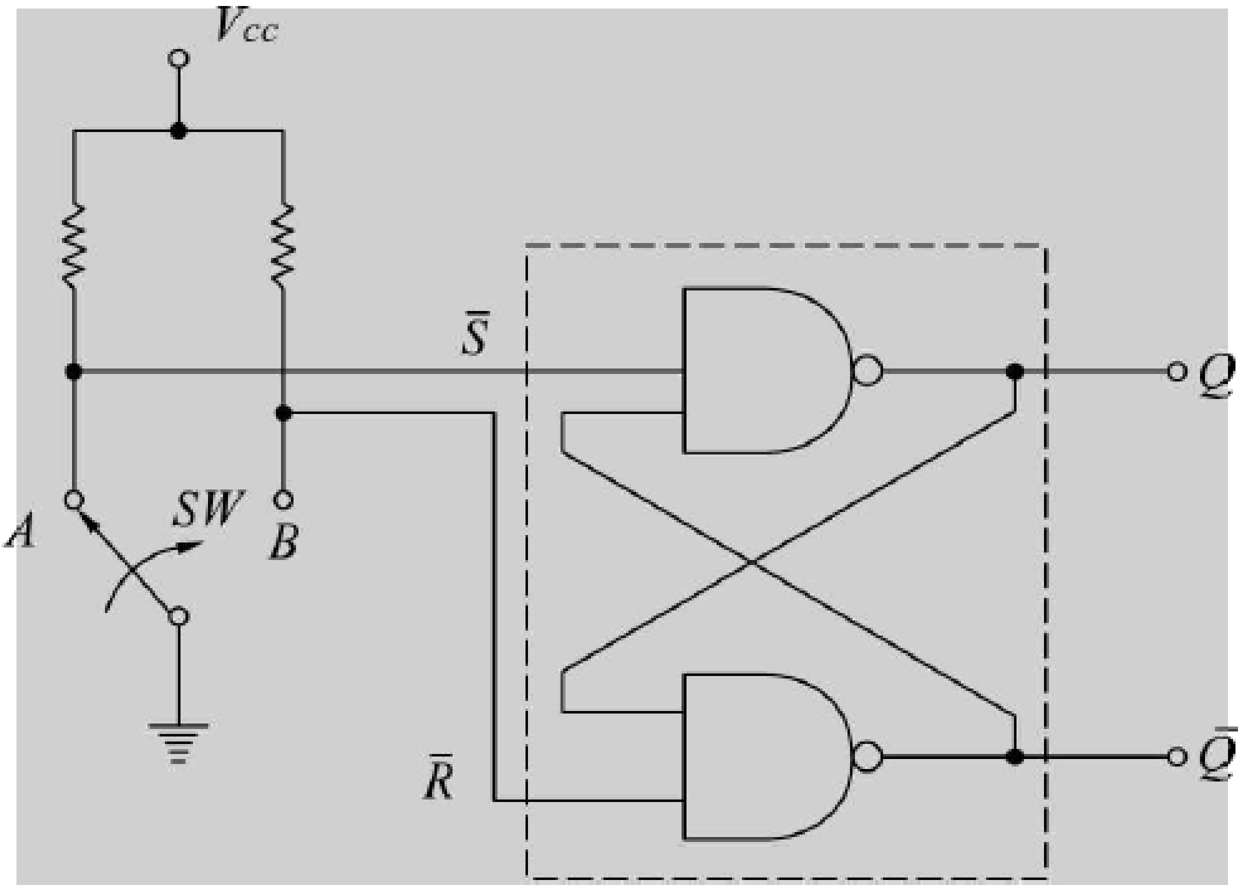
Applications of Flip-Flop

1. Bounce Eliminating Switch

- In digital system, mechanical switches are used as a input device by which binary information is entered into system
- Problem associated is switch-bouncing or chattering
- When switch changes from 1 position to other, it chatters or bounces several times due to spring loaded in it
- In sequential circuit, if 1 is to be entered into system through switch, then switch goes to position 1 and produces output 1 but before that output changes between 0 & 1 for some times due to bouncing of switch.
- This will changes the output of sequential circuit & creates difficulties in the operation of the system

Applications of Flip-Flop

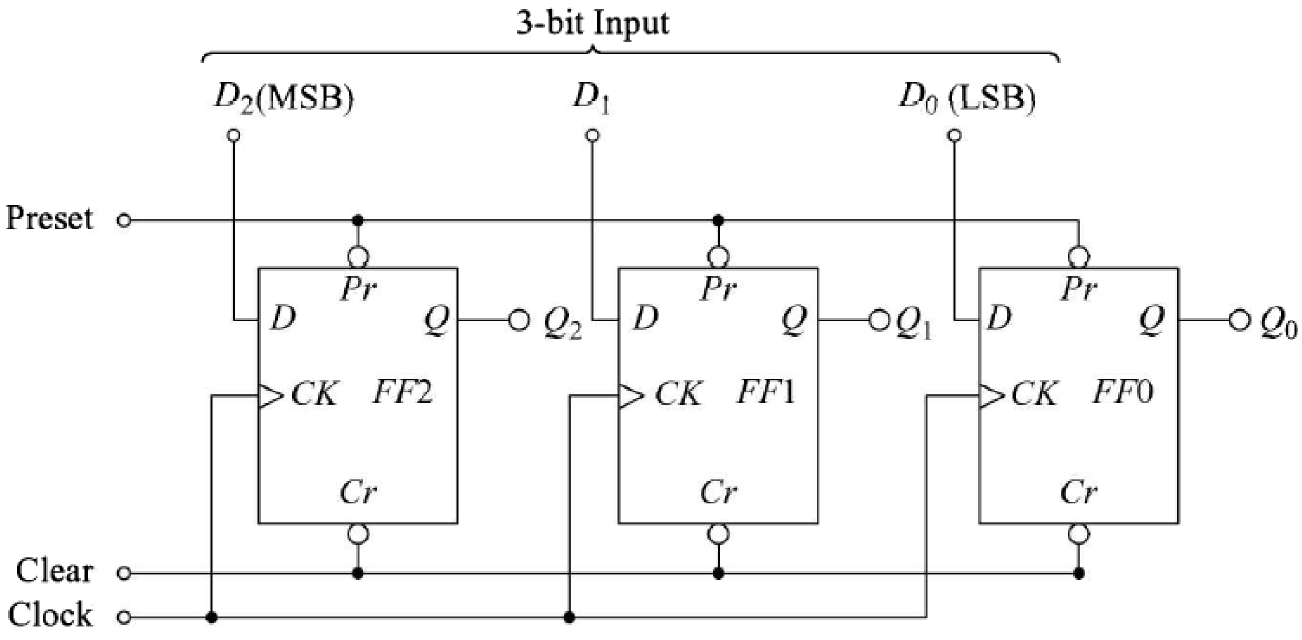
1. Bounce Eliminating Switch



Applications of Flip-Flop

2. Registers

- Register is a composed of a group of flip-flop to store a group of bits.
- For storing n bits, we require a n flip-flops
- 3 bit register using 7474 flip-flop is shown in fig. below
- The bits to be stored are applied at D inputs



A 3-bit Register Using FLIP-FLOPs

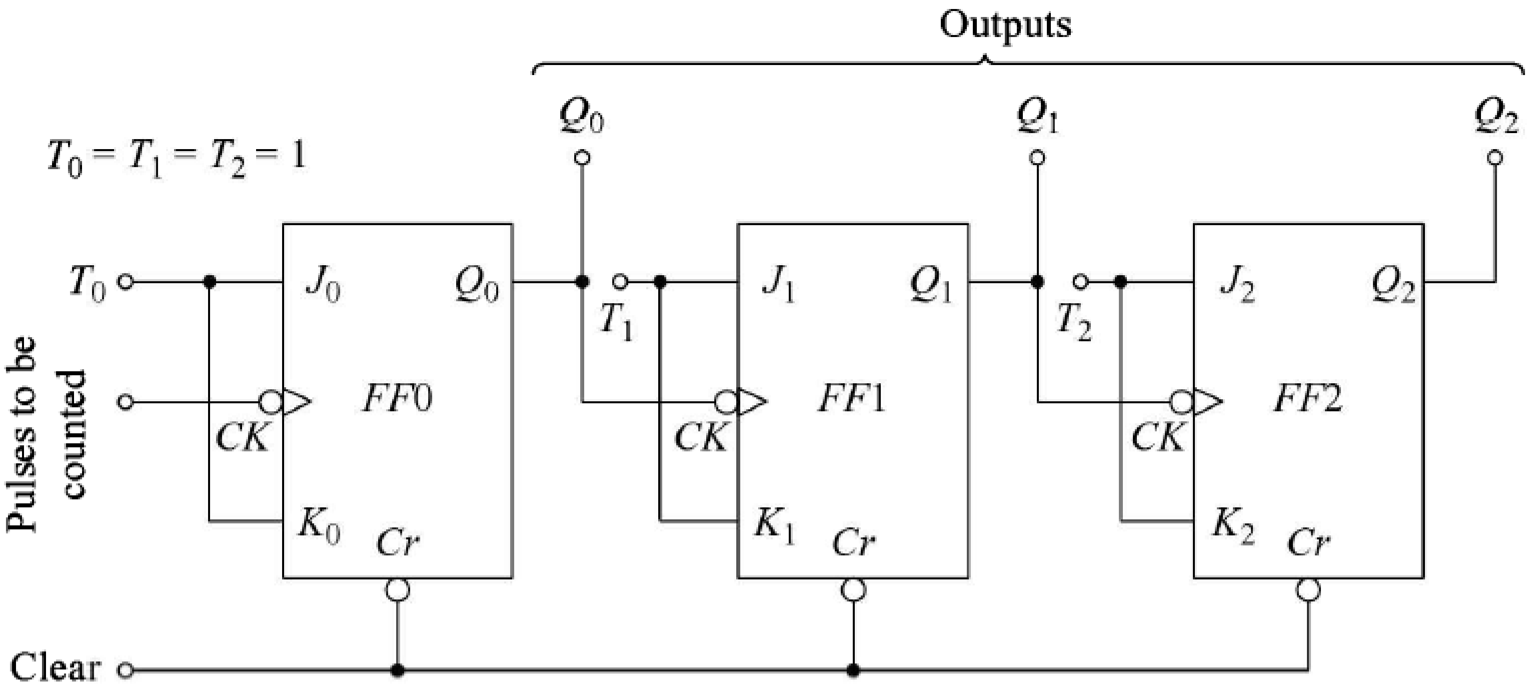
Applications of Flip-Flop

3. Counters

- Digital counters are also composed of flip-flop used to count events.
- Here electrical pulses corresponding to the event are produced using transducer & these pulses are counted using a counter.
- 3 bit counter consisting of 3 flip-flops is shown in figure below. It has 2^n possible states counts from 0 to 7.
- Here pulses to be counted are connected at the CLK input of FF0. Q_0 o/p of FF0 is connected to CLK input of FF1 & similarly Q_1 is connected to CLK input of FF2.

Applications of Flip-Flop

3. Counters



***A 3-bit Counter Using* FLIP-FLOPs**

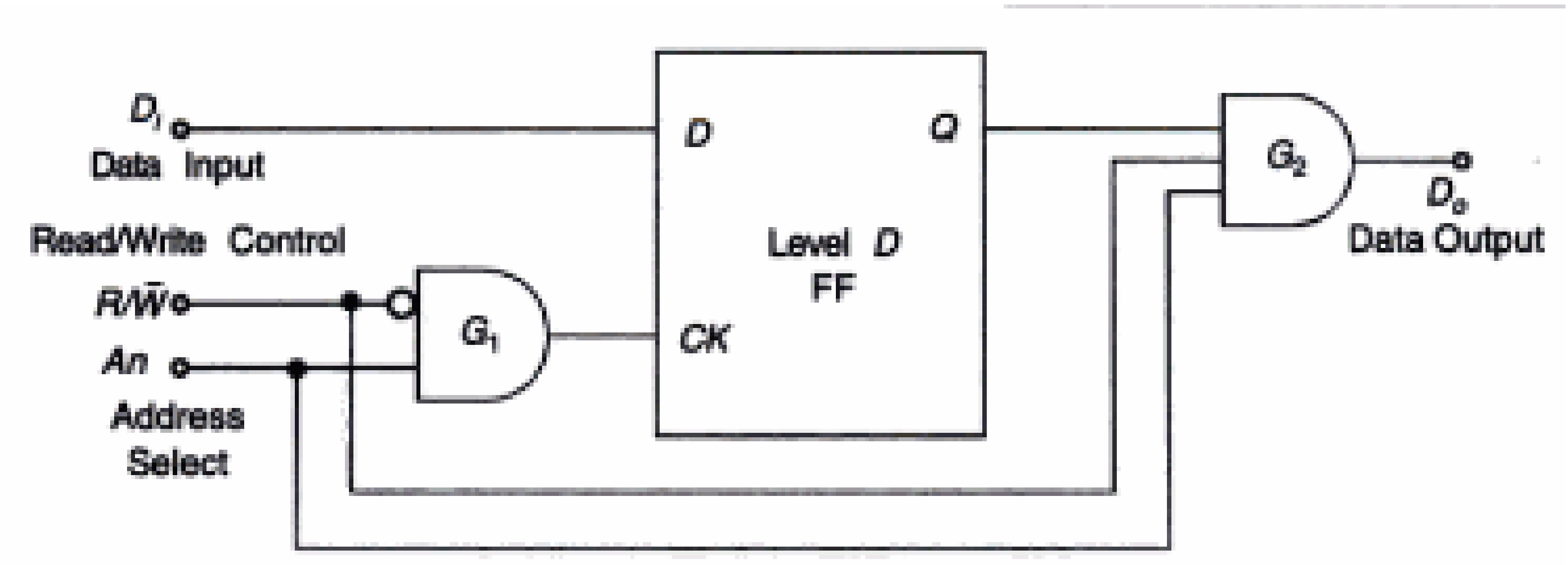
Applications of Flip-Flop

4. RAM

- Flip flops can be used for making memories in which data can be stored for any desired length of time & then read out whenever required. In such memory data can be write into the memory or retrieved from the memory in random fashion
- 1 bit Read/Write memory is shown in fig below has 3 i/p (D i, R/W and A_n) & 1 o/p signal (D_0)
- Read operation is nondestructive i.e. you may read data any number of times without disturbing it even though power is ON.

Applications of Flip-Flop

4. RAM



https://www.youtube.com/watch?v=lnQD2_M9uDI&t=188s

Excitation Table of S-R Flip Flop

Truth Table

S	R	Output (Qn+1)
0	0	Qn
0	1	0
1	0	1
1	1	X

Characteristic Table

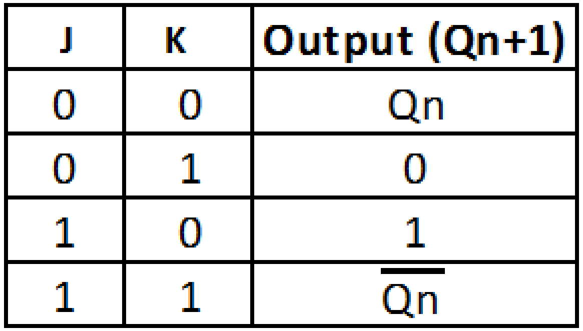
Qn	S	R	Qn+1
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	X
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	X

Excitation Table

Qn	Qn+1	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

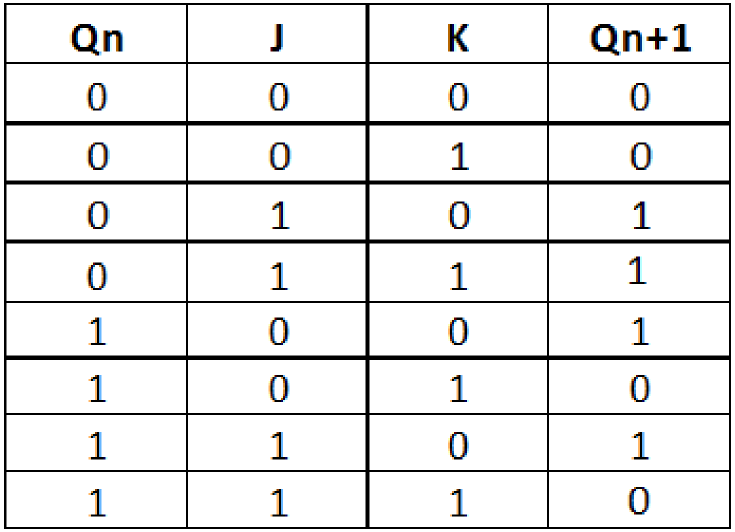
Excitation Table of J-K Flip Flop

Truth Table



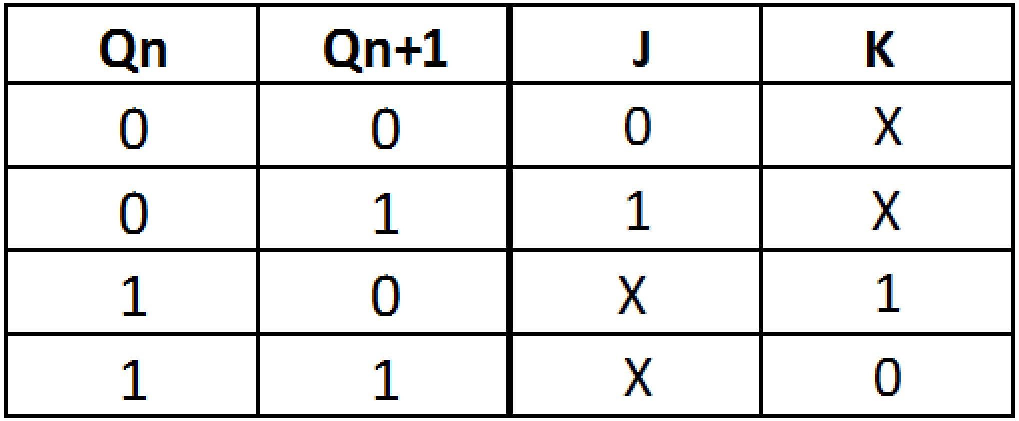
J	K	Output (Qn+1)
0	0	Qn
0	1	0
1	0	1
1	1	$\overline{Qn}$

Characteristic Table



Qn	J	K	Qn+1
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Excitation Table



Qn	Qn+1	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

https://www.youtube.com/watch?v=uiKKRPZbuXA&t=467s

Excitation Table of D Flip Flop

Truth Table

D	Q_n+1
0	0
1	1

Characteristic Table

Q_n+1	D	Q_n
0	0	0
0	1	1
1	0	0
1	1	1

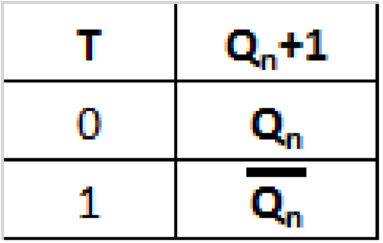
Excitation Table

Q_n	Q_n+1	D
0	0	0
0	1	1
1	0	0
1	1	1

https://www.youtube.com/watch?v=uiKKRPZbuXA&t=467s

Excitation Table of T Flip Flop

Truth Table



T	Q_n+1
0	Q_n
1	$\overline{Q_n}$

Characteristic Table

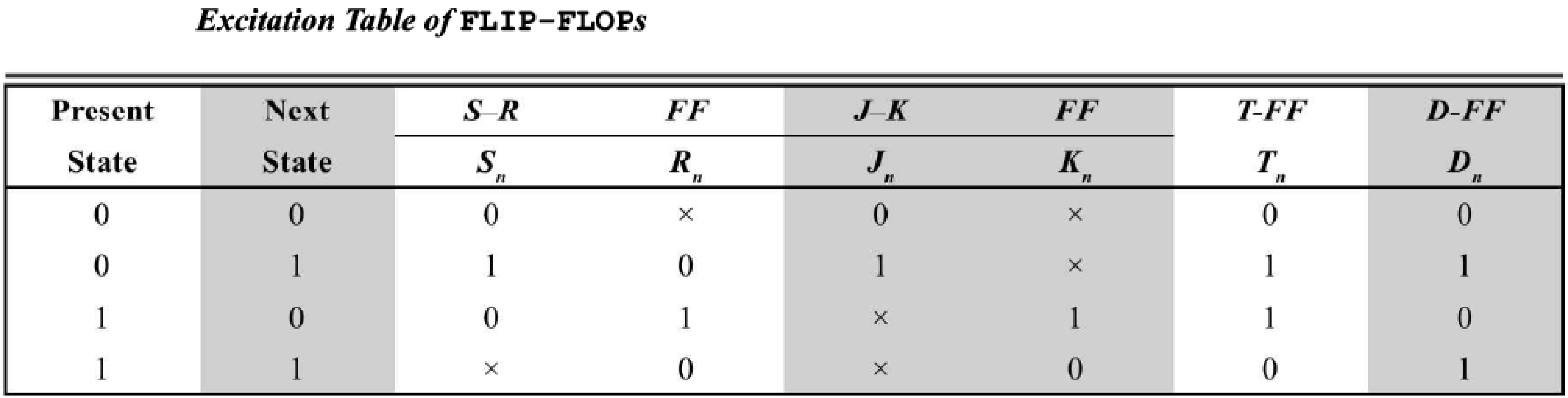
Q_n+1	T	Q_n
0	0	0
0	1	1
1	0	1
1	1	0

Excitation Table

Q_n	Q_n+1	T
0	0	0
0	1	1
1	0	1
1	1	0

https://www.youtube.com/watch?v=uiKKRPZbuXA&t=467s

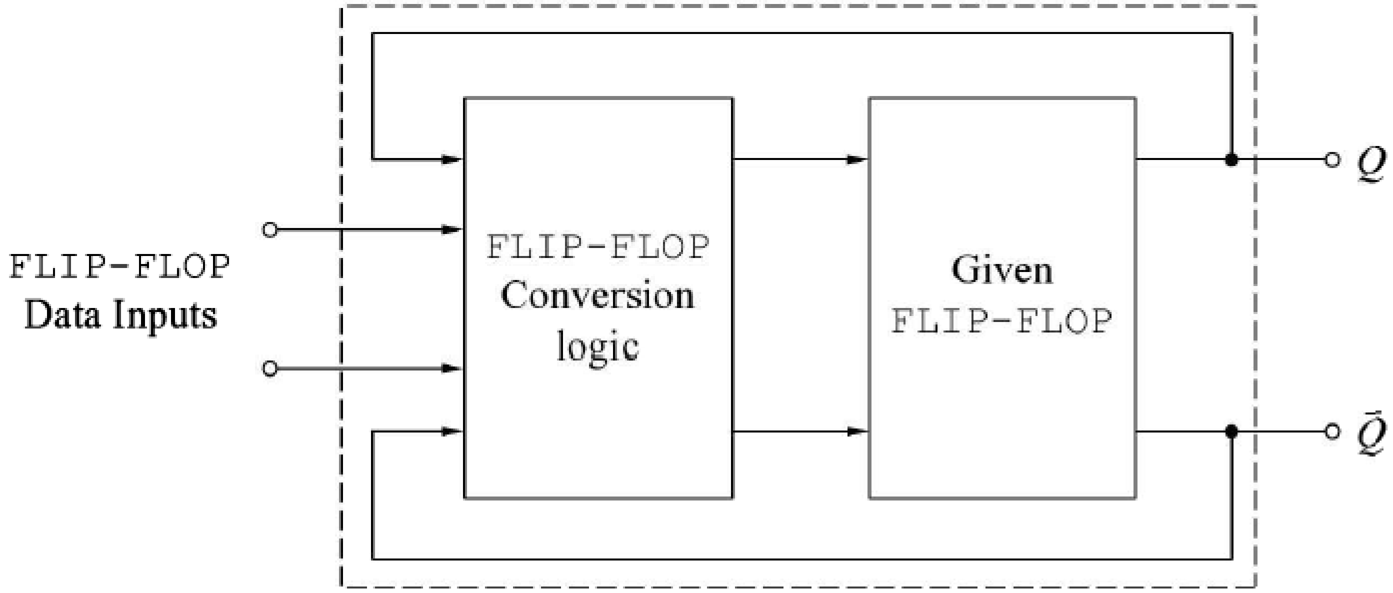
Excitation Table of All Flip Flops



Excitation Table of FLIP-FLOPs

Present	Next	*S–R*	*FF*	*J–K*	*FF*	*T-FF*	*D-FF*
State	State	S_n	R_n	J_n	K_n	T_n	D_n
0	0	0	×	0	×	0	0
0	1	1	0	1	×	1	1
1	0	0	1	×	1	1	0
1	1	×	0	×	0	0	1

Conversion from One Type of Flip Flop to Other



The General Model Used to Convert One Type of* FLIP-FLOP *to Another Type

To design the conversion logic we need to combine the excitation tables for both FLIP-FLOPs and make a truth table with data input(s) and Q as the inputs and the input(s) of the given FLIP-FLOP as the output(s).

Flip-Flop Conversion

- Identify available & required flip-flop
- Make excitation table for available flip-flop
- Make characteristic table for required flip-flop
- Write Boolean expression for the available flip-flop
- Draw the circuit diagram

https://www.youtube.com/watch?v=c4fA4qWys38

Flip-Flop Conversion (SR to JK)

- Identify available & required flip-flop

  Available FF – SR and Required FF - JK

- Make excitation table for available flip-flop

Qn	Qn+1	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

- Make characteristic table for required flip-flop
- Write Boolean expression for the available flip-flop
- Draw the circuit diagram

https://www.youtube.com/watch?v=c4fA4qWys38

Flip-Flop Conversion (SR to JK)

- Make characteristic table for required flip-flop

Q_n	J	K	Q_{n+1}	S	R
0	0	0	0	0	X
0	0	1	0	0	X
0	1	0	1	1	0
0	1	1	1	1	0
1	0	0	1	X	0
1	0	1	0	0	1
1	1	0	1	X	0
1	1	1	0	0	1

- Write Boolean expression for the available flip-flop

Prepare Kmap for output S and R

- Draw the circuit diagram

https://www.youtube.com/watch?v=c4fA4qWys38

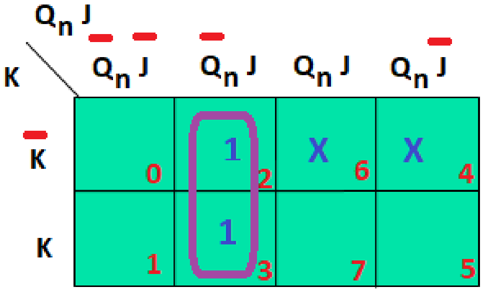
Flip-Flop Conversion (SR to JK)

- Write Boolean expression for the available flip-flop

Prepare Kmap for output

Q_n	J	K	Q_{n+1}	S	R
0	0	0	0	0	X
0	0	1	0	0	X
0	1	0	1	1	0
0	1	1	1	1	0
1	0	0	1	X	0
1	0	1	0	0	1
1	1	0	1	X	0
1	1	1	0	0	1

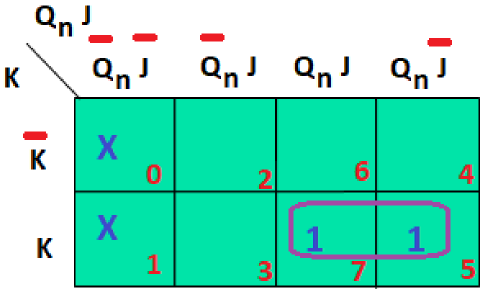
Kmap for S



K \ Q_n J	$\overline{Q_n}\,\overline{J}$	$\overline{Q_n}$ J	Q_n J	$Q_n\,\overline{J}$
$\overline{K}$	0	1 2	X 6	X 4
K	1	1 3	7	5

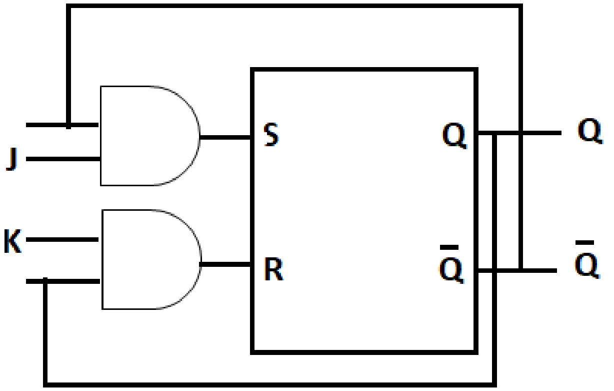
$S = \overline{Q_n} J$

Kmap for R



K \ Q_n J	$\overline{Q_n}\,\overline{J}$	$\overline{Q_n}$ J	Q_n J	$Q_n\,\overline{J}$
$\overline{K}$	X 0	2	6	4
K	X 1	3	1 7	1 5

$R = Q_n K$



- Draw the circuit diagram

https://www.youtube.com/watch?v=c4fA4qWys38

Flip-Flop Conversion (JK to T)

- Identify available & required flip-flop

  Available FF – JK and Required FF - T

- Make excitation table for available flip-flop

Qn	Qn+1	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

- Make characteristic table for required flip-flop
- Write Boolean expression for the available flip-flop
- Draw the circuit diagram

https://www.youtube.com/watch?v=c4fA4qWys38

Flip-Flop Conversion (JK to T)

- Make characteristic table for required flip-flop

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

- Write Boolean expression for the available flip-flop

Prepare Kmap for output S and R

- Draw the circuit diagram

https://www.youtube.com/watch?v=-I_Nn2FtzF8&list=TLPQMTIxMjIwMjI-iKDWk1_5Wg&index=3

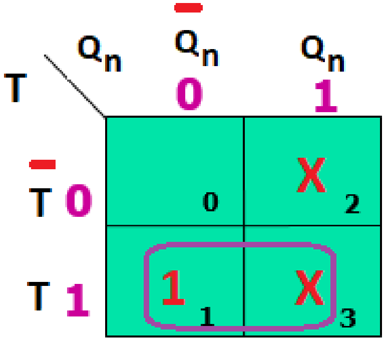
Flip-Flop Conversion (JK to T)

- Write Boolean expression for the available flip-flop

Prepare Kmap for output

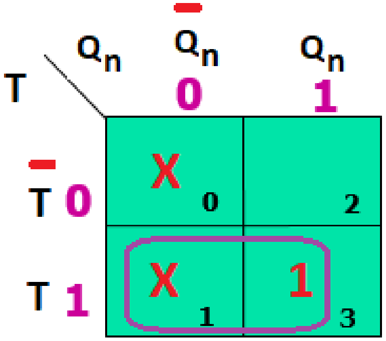
Q_n	T	Q_{n+1}	J	K
0	0	0	0	X
0	1	1	1	X
1	0	1	X	0
1	1	0	X	1

Kmap for J

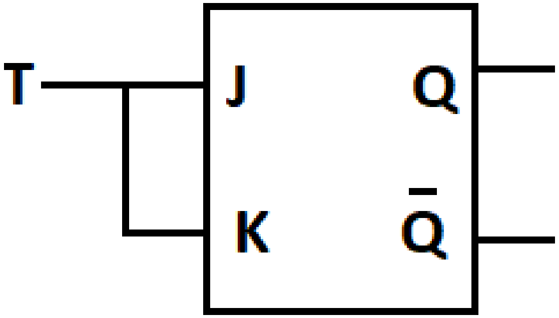


J = T

Kmap for K



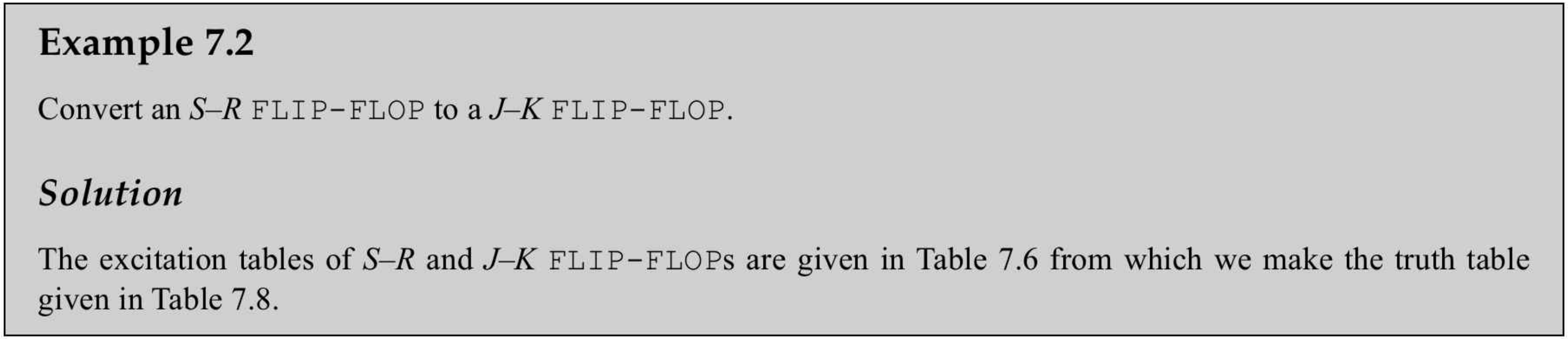
K = T



- Draw the circuit diagram

https://www.youtube.com/watch?v=c4fA4qWys38

Conversion from One Type of Flip Flop to Other



Example 7.2

Convert an *S–R* FLIP-FLOP to a *J–K* FLIP-FLOP.

Solution

The excitation tables of *S–R* and *J–K* FLIP-FLOPs are given in Table 7.6 from which we make the truth table given in Table 7.8.

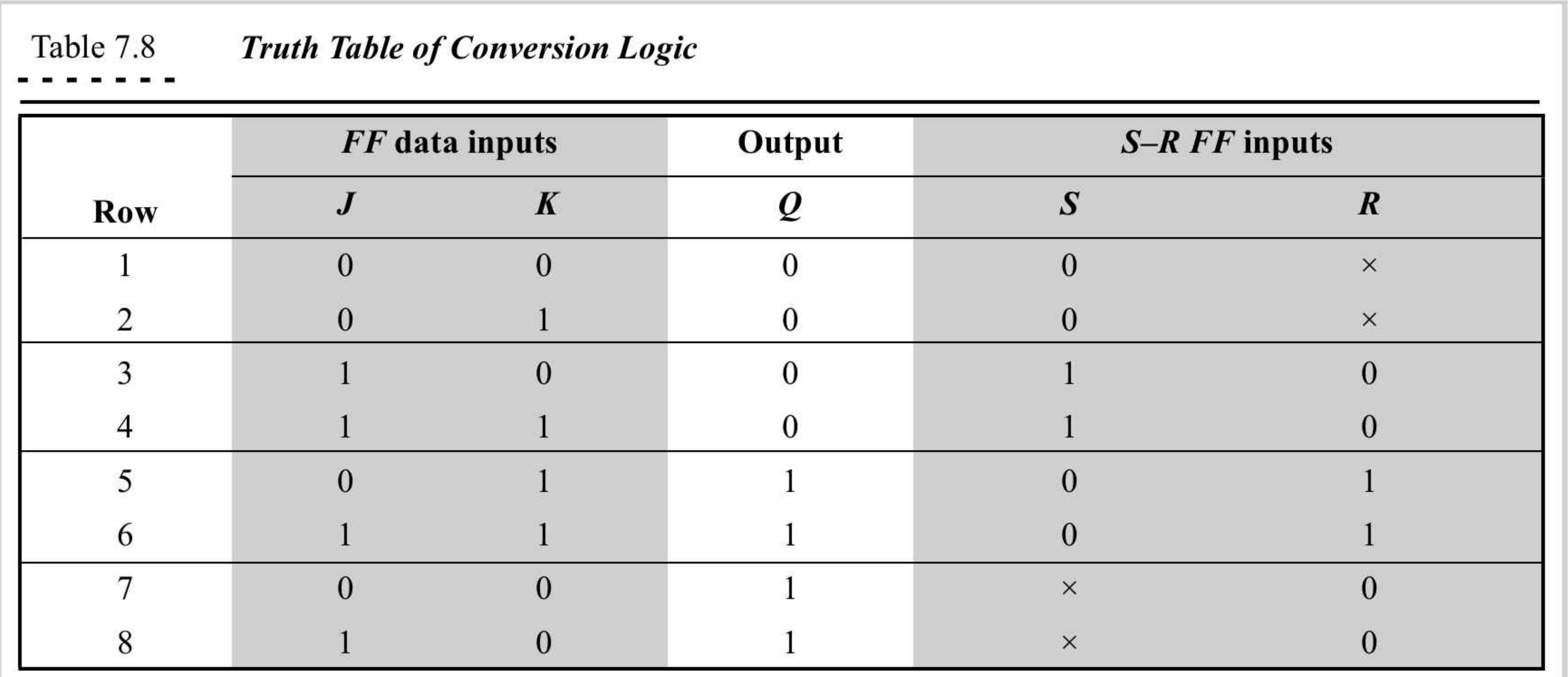
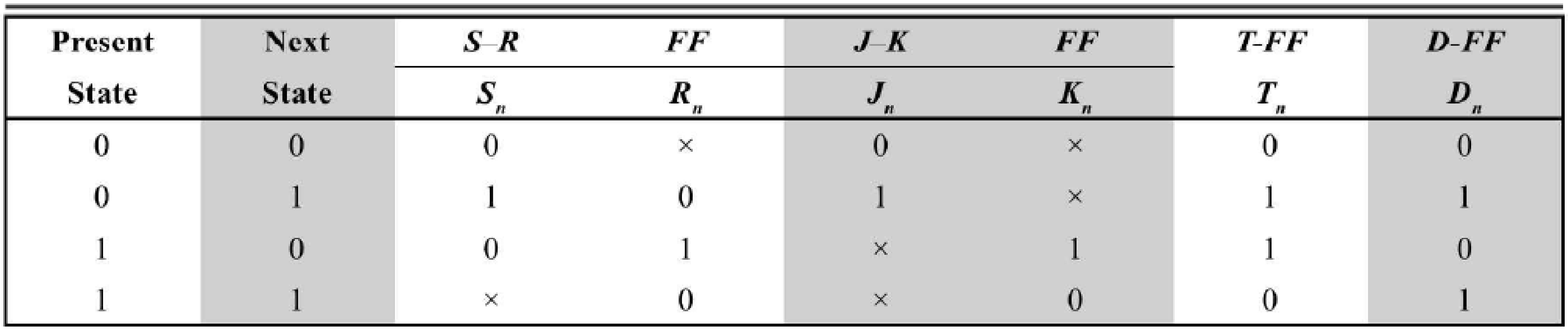


Table 7.8 ***Truth Table of Conversion Logic***

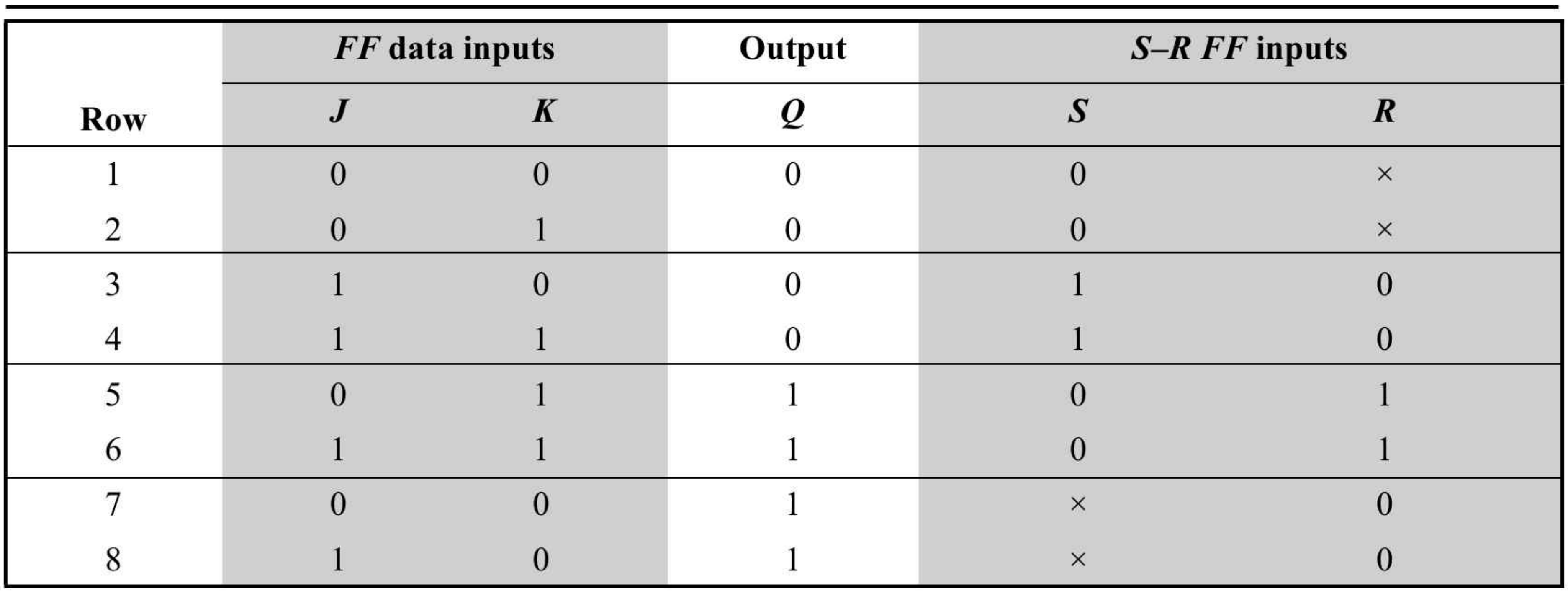
	***FF* data inputs**		**Output**	***S–R FF* inputs**	
Row	***J***	***K***	***Q***	***S***	***R***
1	0	0	0	0	×
2	0	1	0	0	×
3	1	0	0	1	0
4	1	1	0	1	0
5	0	1	1	0	1
6	1	1	1	0	1
7	0	0	1	×	0
8	1	0	1	×	0

Conversion from One Type of Flip Flop to Other

***Excitation Table of* FLIP-FLOPs**

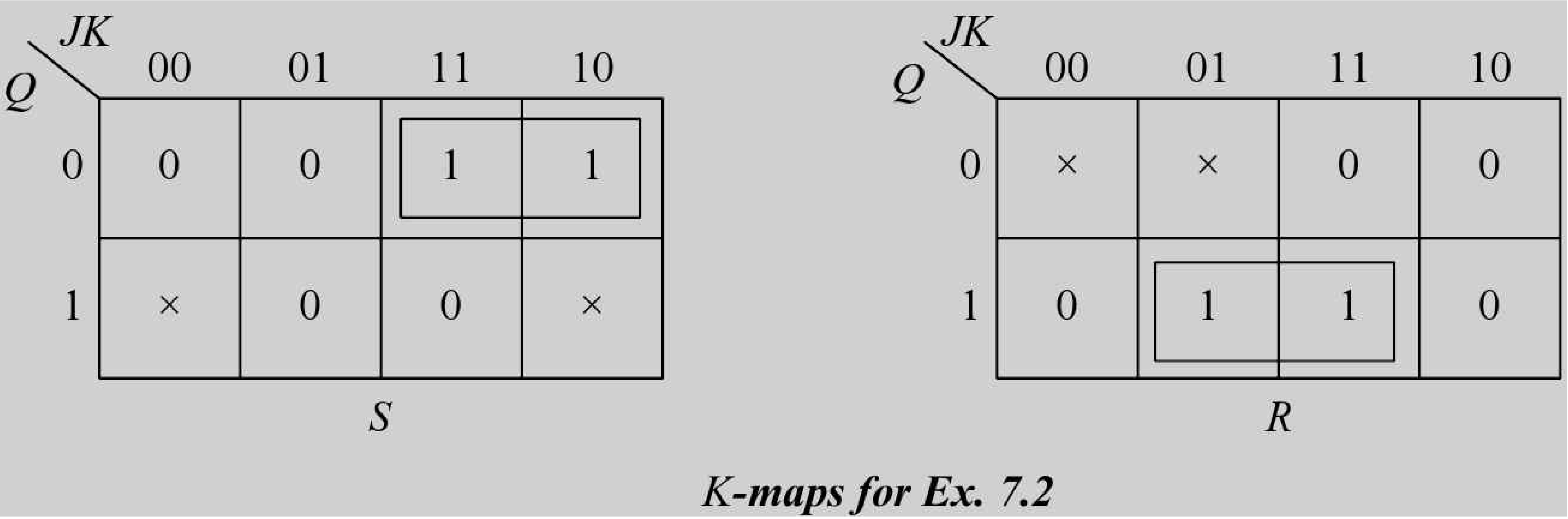


Present	Next	*S–R*	*FF*	*J–K*	*FF*	*T-FF*	*D-FF*
State	State	S_n	R_n	J_n	K_n	T_n	D_n
0	0	0	×	0	×	0	0
0	1	1	0	1	×	1	1
1	0	0	1	×	1	1	0
1	1	×	0	×	0	0	1



	FF data inputs		Output	*S–R FF* inputs	
Row	*J*	*K*	*Q*	*S*	*R*
1	0	0	0	0	×
2	0	1	0	0	×
3	1	0	0	1	0
4	1	1	0	1	0
5	0	1	1	0	1
6	1	1	1	0	1
7	0	0	1	×	0
8	1	0	1	×	0

Conversion from One Type of Flip Flop to Other



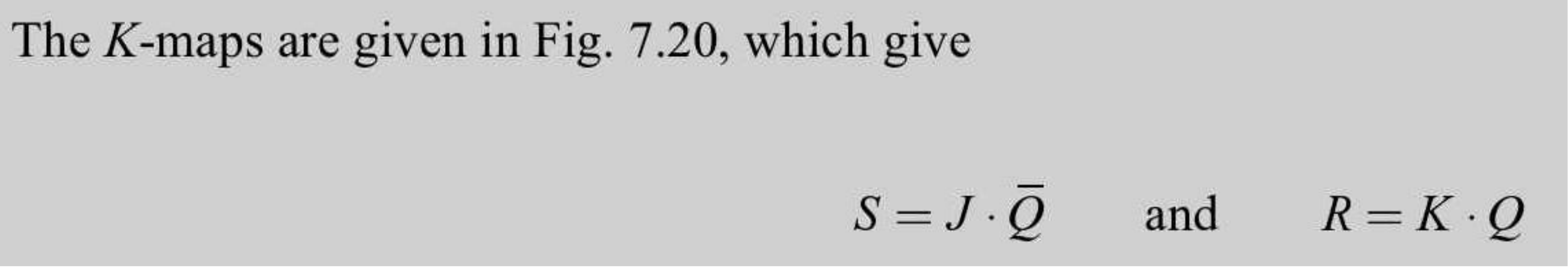
Q \ JK	00	01	11	10
0	0	0	1	1
1	×	0	0	×

S

Q \ JK	00	01	11	10
0	×	×	0	0
1	0	1	1	0

R

K-maps for Ex. 7.2



The *K*-maps are given in Fig. 7.20, which give

$$S = J \cdot \bar{Q} \qquad \text{and} \qquad R = K \cdot Q$$