The third input, C_{in} represents the carry from the previous lower significant position. The binary variable S gives the value of the LSB of the sum and the binary variable C_{out} gives the output carry.

A full adder can be formed using two half adder circuits and OR gate as shown in figure 6.17.

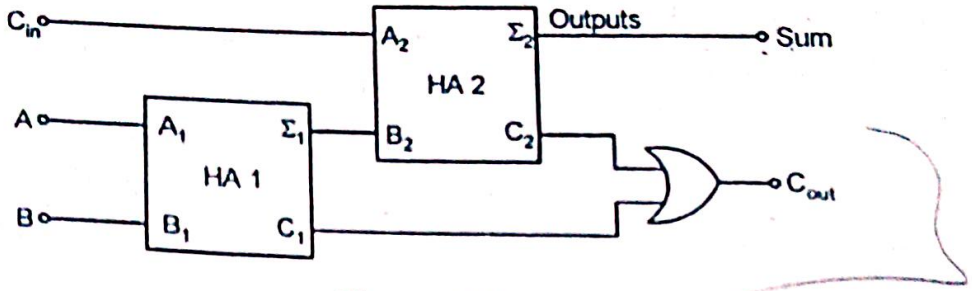


Figure 6.17

6.6 Flip Flops

The output levels of the combinational logic circuits at any instant of time are dependent on the levels present at the inputs at that time. Since combinational logic circuits have no memory, the prior input level conditiions have no effect on the present outputs. Most digital systems are made up of both combinational circuits and memory elements.

Figure 6.18 shows a block diagram of general digital system that combines combinational logic gates with memory devices.

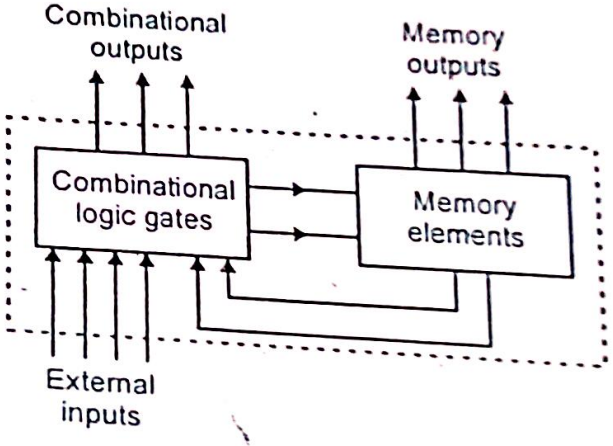


Figure 6.18

The combinational portion accepts logic signals from external inputs and from the outputs of the memory elements. The combinational circuit operates on these inputs and produce various outputs. Some of these outputs are used to determine the binary values to be stored in the memory elements. The outputs of some of the memory elements, in turn, go to the inputs of logic gates in the combinational circuits. This shows that the

external outputs of a digital system are a function of both its external inputs and the information stored in its memory elements.

The most important memory element is the "flip-flop". These flip flops are made up of an assembly of logic gates. We know that, a logic gate, by itself, has no storage capability. But, if several logic gates can be connected together in ways that permit information to be stored. There are many different gate arrangements that are used to produce these flip flops (FF).

Flip-Flops

A flip flop can have one or more inputs. These inputs make the FF to switch back and forth ("flip-flop") between its possible output states. Hence it is called so. A flip-flop input only has to be pulsed for a moment in order to make a change in the FF output state. But the output will remain in that new state itself even after the input pulse is over. This is the FF's "memory" characteristics.

The flip flop is known by other names, such as "Latch" and "Bistable Multivibrator". The term "Latch" is used for certain types of flip flops. The term "bistable multivibrator" is the more technical name for a flip-flop.

Four types of flip-flops are commonly used. They are

i) J-K flip flop

ii) S-R flip flop

iii) D flip flop

iv) T flip flop

Each FFs stores binary data but each has a unique set of input variables. Each flip-flop type can be described in terms of its input and output characteristics by writing a special truth table, called an "excitation table". A characteristics equation can be generated from the excitation tables. The names "flip-flops" and "latches" are sometimes used interchangeably. However, the term "flip-flop" is associated with the devices that change state only on a clock edge or pulse, whereas latches change state without being clocked. Flip-flops are clocked and latches are not; both are bistable (two stable states) and both are used to store binary data. Some latches are allowed to change state only when gated with an external "enable" signal.

6.6.1 SR Flip Flops

The most basic FF circuit can be constructed from either two NAND gates or two NOR gates. The NAND gate version, which is called as "NAND gate Latch" is shown in the figure 6.19.

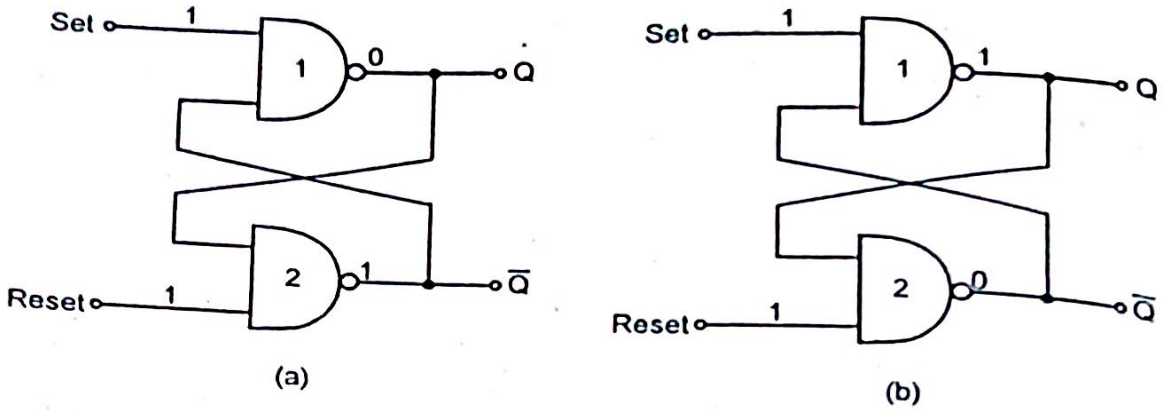


Figure 6.19

The two NAND gates are cross-coupled so that the output of NAND-1 is connected to one of the inputs of NAND-2 and vice-versa. The gate outputs, $\overline{Q}$ and q are the latch outputs. Under normal conditions, these outputs are always be inverse of each other. There are two latch inputs. The set input is the input that "sets" Q to the 1 state; the RESET input is the input that "resets" Q to the 0 state.

The operation of the S-R Flip-Flop can be summarized as follows

1. **SET = RESET = 0**

This condition is the normal resting state and it has no effect on the output state. The Q and $\overline{Q}$ outputs will remain in whatever state they were before this input condition.

2. **SET = 0; RESET = 1**

This will always cause the output to go to the Q = 1 state, where it will remain even after SET goes to HIGH. This is called "setting" the latch.

3. **SET = 1; RESET = 0**

This will always produce the Q = 0 state, where the output will remain even after RESET goes to HIGH. This is called "clearing or resetting" the latch.

4. SET = RESET = 0

This condition tries to set and clear the latch at the same time and can produce uncertain or doubtful results. It should not be used.

The truth table for the S-R latch is given below.

Inputs		*Outputs*		*Comments*
S	R	Q	$\overline{Q}$	
0	0	1	1	Invalid
0	1	1	0	Set
1	0	0	1	Reset
1	1	x	$\overline{x}$	No change. Same as Previous state; set or reset

The timing diagram for a NAND latch is given in figure 6.20.

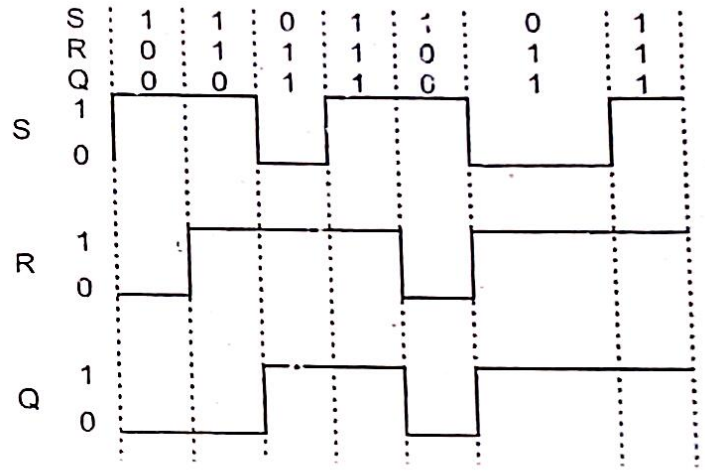


Figure 6.20

From the analysis of NAND SR latch operation, we are able to know that the SET and RESET inputs are active -low. The set input will set Q = 1 when SET goes low; the RESET input will clear Q = 0 when RESET goes low. For this reason, the NAND latch is drawn by the labelling the signals as $\overline{SET}$ and $\overline{RESET}$ to indicate the active LOW status of these inputs.

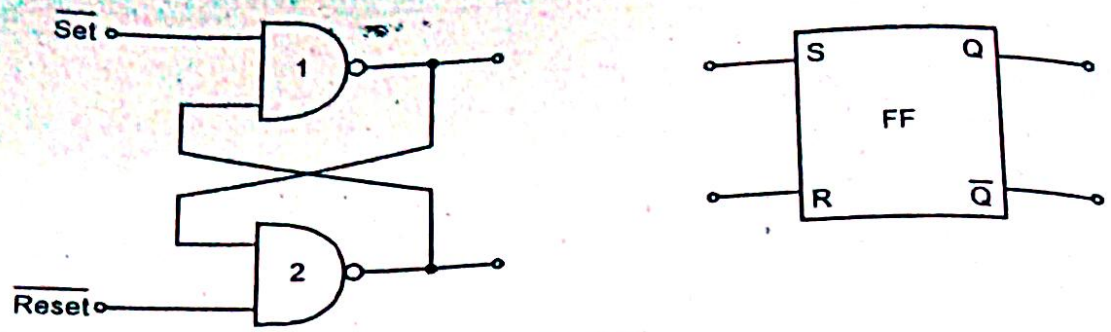
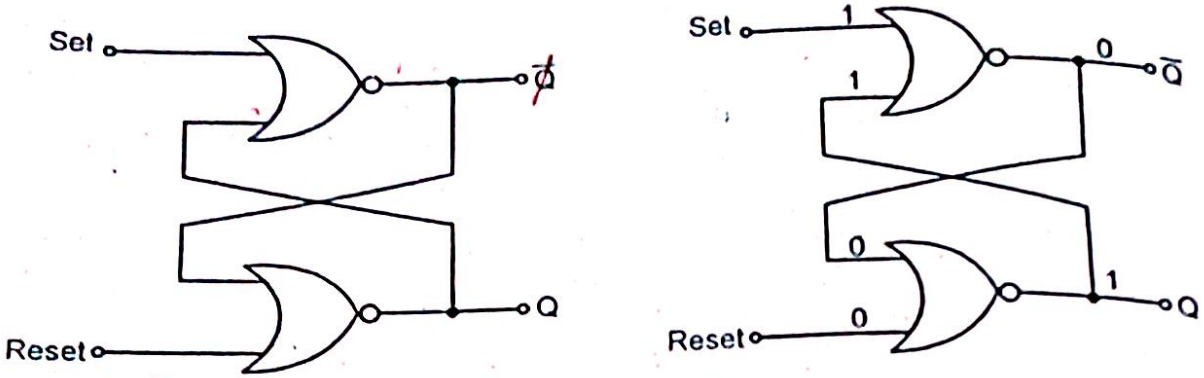


Figure 6.21

NOR - SR Flip Flops

Figure 6.22 shows a NOR SET-RESET flip-flop. Here the inputs are not complemented; therefore they are active high.



(a) NOR SR Flip flop

(b) Setting the Q output for a NOR SR Flip flop

Figure 6.22

When the SET input goes to a 1 and the RESET remains at 0, then the Q output goes to a 1 state. Any 1 into a NOR gate will produce 0 output.

When the SET input returns to 0 and RESET is also 0, the outputs Q and $\overline{Q}$ donot change. This is because the outputs of the NOR gates are tied back to the opposite gates input. This keeps the gates changing from states. To bring Q back to a 0, the RESET input must be made 1 while the SET input is held at 0. The unused state for the NOR SR flip flop is where the SET is 1 and the RESET is 1. The first input to return to the 0 state will determine the output state of the Q and $\overline{Q}$ outputs. The truth table for the NOR SR flip flop is shown below.

Inputs		Output		Comments
S	R	Q	$\overline{Q}$	
0	0	Q	$\overline{Q}$	Unchanged state
0	1	0	1	
1	0	1	0	
1	1	1	1	Unused state

The difference between the NOR SR FF and NAND SR FF are

i) The values of $\overline{Q}$ and Q in the unused state.

ii) The another difference is the SET and RESET inputs.

The NOR SR FF inputs are active HIGH and the $\overline{SET}$, $\overline{RESET}$ ip/s on the NAND gates are active low. This means that the NOR SR FF will change state when an input goes HIGH or 1 and the NAND will change state when input goes LOW or 0.

6.6.2 JK Flip Flop

The uncertain state of output in RS FF was eliminated in D FF by joining the inputs with an inverter. But D FF is a single input FF which is able to transmit data from input to output. JK FF is also similar to that of a RS FF, in such a way that it has 2 synchronous control inputs, but the uncertain state has been eliminated. When J = K = 1, the FF toggles (changes).

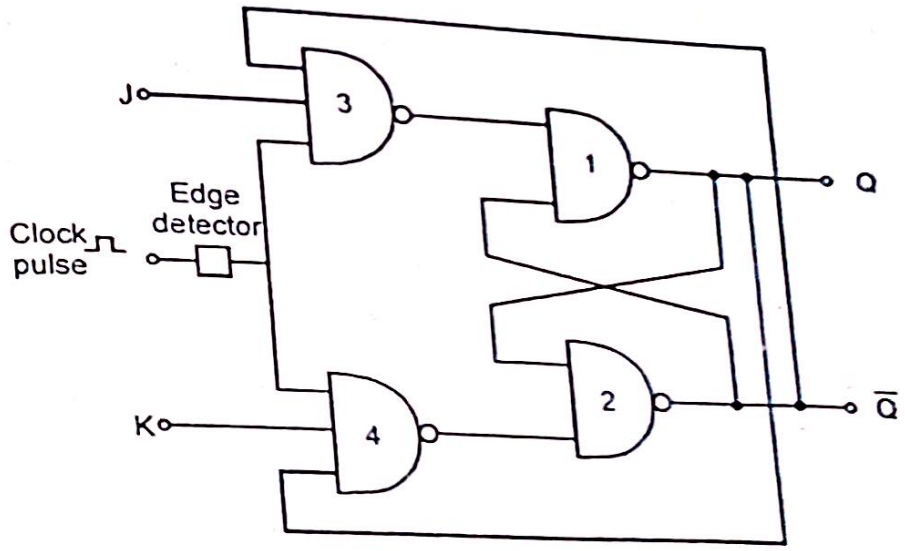
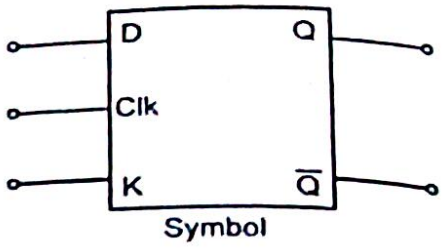


Figure 6.23

Figure 6.23 shows the JK FF. The truth table and the symbol is shown below.

Clk	*J*	*K*	*Q*
↓	x	x	No change
↑	0	0	Q_0 No change
↑	0	1	0
↑	1	0	1
↑	1	1	Toggles Q_0.



Symbol

Except from the feedback from $\overline{Q}$ and $\overline{Q}$, the circuit remains the same as a RS FF. This feed back clears the uncertainity (ambiguity) in the RS FF.

The logic diagram for JK NOR FF is shown in the figure 6.24.

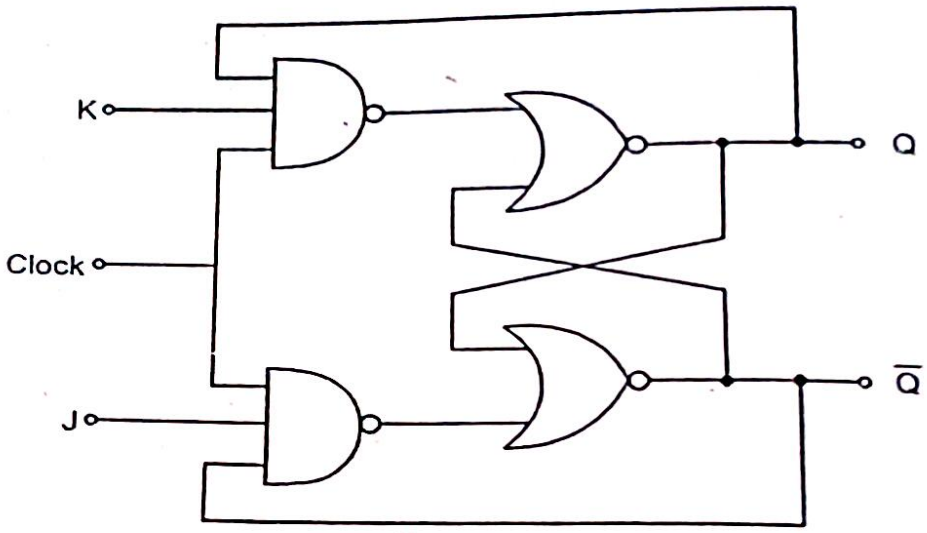


Figure 6.24

The truth table is the same as for NAND JK FF.

Operation of JK NAND FF

i) When J = K = 0

If Q was previously by 0, $\overline{Q} = 1$. When we apply clock input, the output of gate-3 is 1. This, in turn, along with $\overline{Q}$ produces 0 in Q. If Q was previously 1, $\overline{Q} = D$. Similarly, the output of gate-3 is 0, if we apply the clock input. This along with $\overline{Q}$ produces 1 in Q. Since, where both J and K inputs are zero, there is no change in the output.

ii) When J = 0, K = 1

If $\bar{Q} = 1$, then $\bar{Q} = 0$. The output of gate-4 is zero. This inturn with Q produces a 1 is $\bar{Q}$. Hence Q = 0. If Q = 0 and $\bar{Q} = 1$, then output of gate-4 is one. This along with Q produces a 1 in $\bar{Q}$. So Q = 0. Thus, with out considering of any previous state outputs, when J = 0 and K = 1, the FF is reset.

iii) When J = 1 and K = 0

When Q = 1, $\bar{Q} = 0$, the output of gate - 3 is 0, output of gate - 1 is 1. When Q = 0, $\bar{Q}$=1, the output of gate - 3 is 0. Output of gate - 1 is 1. Hence when J = 1 and K = 0, the output Q is always 1, the FF is set.

iv) When J = 1 and K = 1

If Q = 1, then $\bar{Q} = 0$. The output of JK FF is to be determined, when both the inputs are 1, by the previous state. Here Q = 1. Hence along with K produces 0 in the gate-4 output. This output and Q together again produces a 1 in $\bar{Q}$. Therefore Q = 0. If Q = 0, then $\bar{Q}$=1. $\bar{Q}$ along with J produces 0 in the gate-3 output. This output and $\bar{Q}$ then produces a 1 in Q. Hence $\bar{Q} = 0$. We can notice that the output is complemented when J = K = 1.

6.6.3 JK Master Slave Flip-Flop

The clocked flipflops shall bring some other problems. i.e., if the FF is a level triggered one, the FF changes state according to its inputs, as shown as the clock becomes 1. But, if the inputs change, before the clock switches off to 0, then another state transition occurs for the same clock pulse. The input change can occur because of the feedbacks in FF circuit. This kind of multiple transition problem is called "Racing". It is shown in figure 6.25.

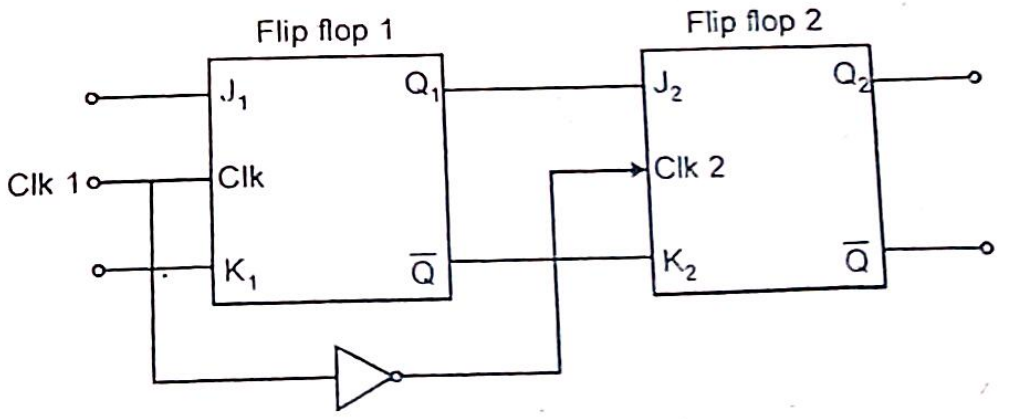


Figure 6.25

Also, when there is a continuous connection of FFS together, i.e., the outputs of one FF connected to the input of another FF, an unexpected response may occur. This is also another case of racing problem.

Hence to solve these problems, Master-Slave FFS are used. The figure 6.26 shows block diagram of a master - slave FF.

The master Flip flop is positively clocked but the slave if negatively is negatively clocked. Hence, the master FF responds to J and K inputs before the slave FF does. If J_1=1 and K_1= 0, the master sets on the positive clock edge. So, Q_1 = HIGH. Now, the HIGH output of Q_1 drives J_2. During the negative clock transmission, the 2nd FF is set by copying the action of the master. If $J_1 = 0$ and $K_1 = 1$, the master resets on the positive clock edge. $\therefore$ Q_1 = LOW. The LOW output of Q_1 is given to J_2. During the negative clock transmission, the 2nd FF is reset, again by copying the action of the master.

The figure below shows a master slave FF along with its detailed diagram.

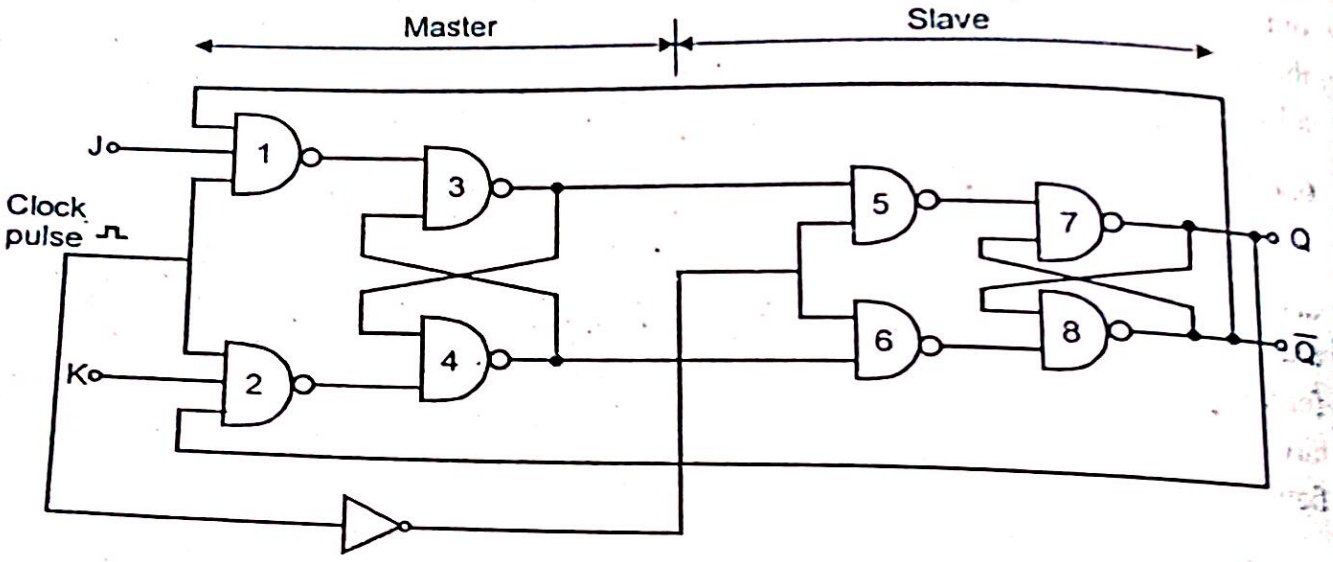


Figure 6.26

If both the inputs are 1, the master FF toggles positive edge and the slave FF toggles on negative edge of the clock. If both the inputs are low, there is no change in the output. This mode is called "Inhibit mode".

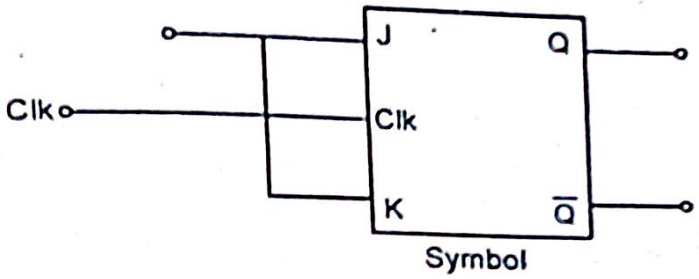
The four modes of operation for the master-slave FF is given in the table below.

Inputs		*Outputs*	
J	K	Q(t)	Q(t+1)
0	0	x	x-inhibit
0	1	x	0 - Reset
1	0	x	1 - Set
1	1	x	x Toggle

The master-slave if is best-suited for synchronous data transfer. Also when the FF is edge triggered, the race condition is avoided.

6.6.4 T - Flip Flop

When the two inputs of JK FF are shorted, the T-FF is formed. The symbol of T-FF is shown in figure 6.27.



Symbol

Figure 6.27

Truth Table

T	Q
0	No change
1	Complement

The T-FF is named so, because of its ability to "Toggle" (or change or complement) its state. When the T input is 0, the FF state does not have any change. But when T = 1, the state of FF complements.

6.6.5 The Transparent D Flip Flop

There is one problem with the NAND SR FF. That is there can be a 1 on the Q and a 1 on the $\bar{Q}$ when the SET and RESET inputs are both 1. This is the unused state which should be avoided if possible. Also, it will be much more easy if one input could SET and RESET the flip-flop. Both of these problems can be relieved by placing an inverter between the SET and RESET inputs as shown in figure 6.28. This makes a new input which we call the D input.

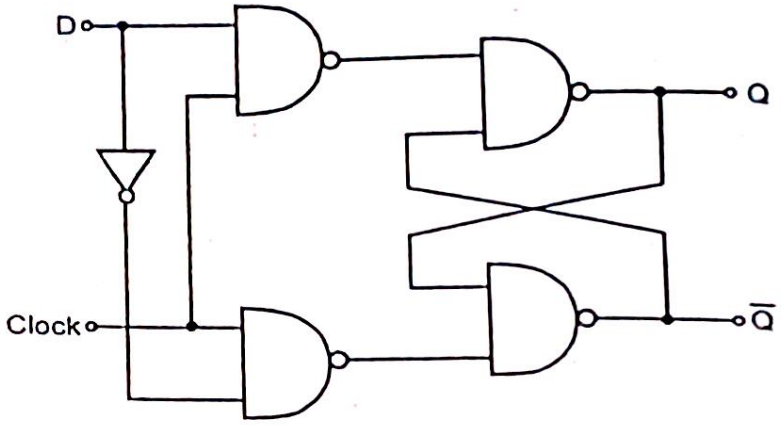


Figure 6.28

Here we can notice that, the SET and RESET inputs can never be the same value because of the inverter. This means that the unused state can never exists. Also, there is one D or data input to set or reset the FF.

When the clock is 1, it enables the gates to the $\overline{SET} - \overline{RESET}$ flip flop, the value of D (1 or 0) is transferred to the Q output. When the clock is 0, the Q and $\bar{Q}$ outputs cannot be changed by the D input.

This type of D-Flip flop is called as "transparent D flip-flop" because when the clock is 1, the Q changes when D changes. The flip flop appears transparent until the clock falls to 0, at which the flip flop becomes opaque. (not transparent). Figure shows the truth table for the transparent D-flip flop.

D	Clock	Q	$\bar{Q}$	Comments
0	0	Q	$\bar{Q}$	Unchanged
1	0	Q	$\bar{Q}$	State
0	1	0	1	
1	1	1	0	

The D flip flop is used to store bits of binary numbers. Because it can be turned on or off by the clock, it is also used to catch or latch binary number present on the D input for a short time and store it on the Q and $\bar{Q}$ outputs. A D flip flop can be used as the output port of a micro-computer.

The symbol of the D flip-flop is shown below figure 6.29.

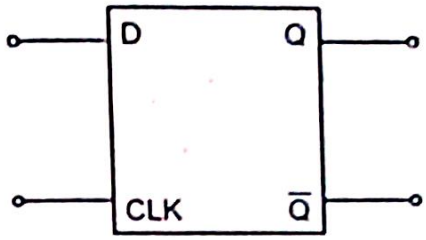


Figure 6.29

From the truth table, we can understand that when D is 0, the output is also zero. When the input is 1, the output is also 1. Hence it is called "DELAY FF" or "Buffer" or "Gated" D latch.

6.7 Basic of Counters

The counter has the ability to count and is a very important and useful subsystem of a digital system. A flip-flop can store one binary information. Hence, in order to store more binary information, we need registers. Registers are the one which is having a group of cascaded flip-flops.

A counter is a register. It is capable of counting the number of clock pulses, which have arrived at its clock input. So, it has to actually remember the number of clock pulses applied at the input.

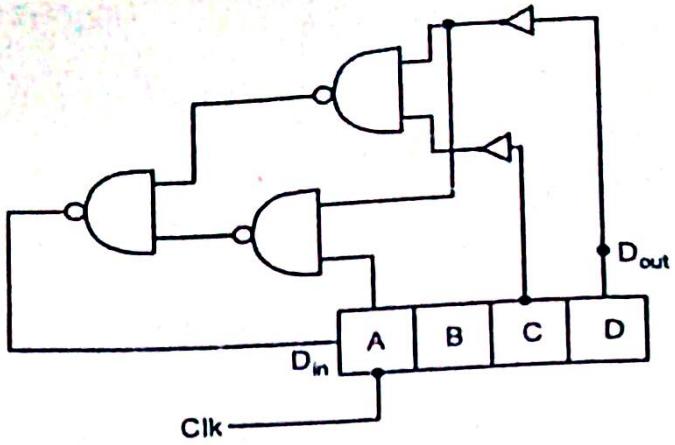


Figure 6.36

6.8 Shift Registers

A register which is capable of shifting the binary information to the left or right is called a register. The most basic form is formed using D.FFs. D.FF is the most suitable one because, D-FF transfers the input to the output. The serial input SI contains the data and it is transferred to the next FF through the Q outputs for every clock pulse till it reaches the serial output SO in the last FF. Figure 6.37 shows the timing diagram of a shift register.

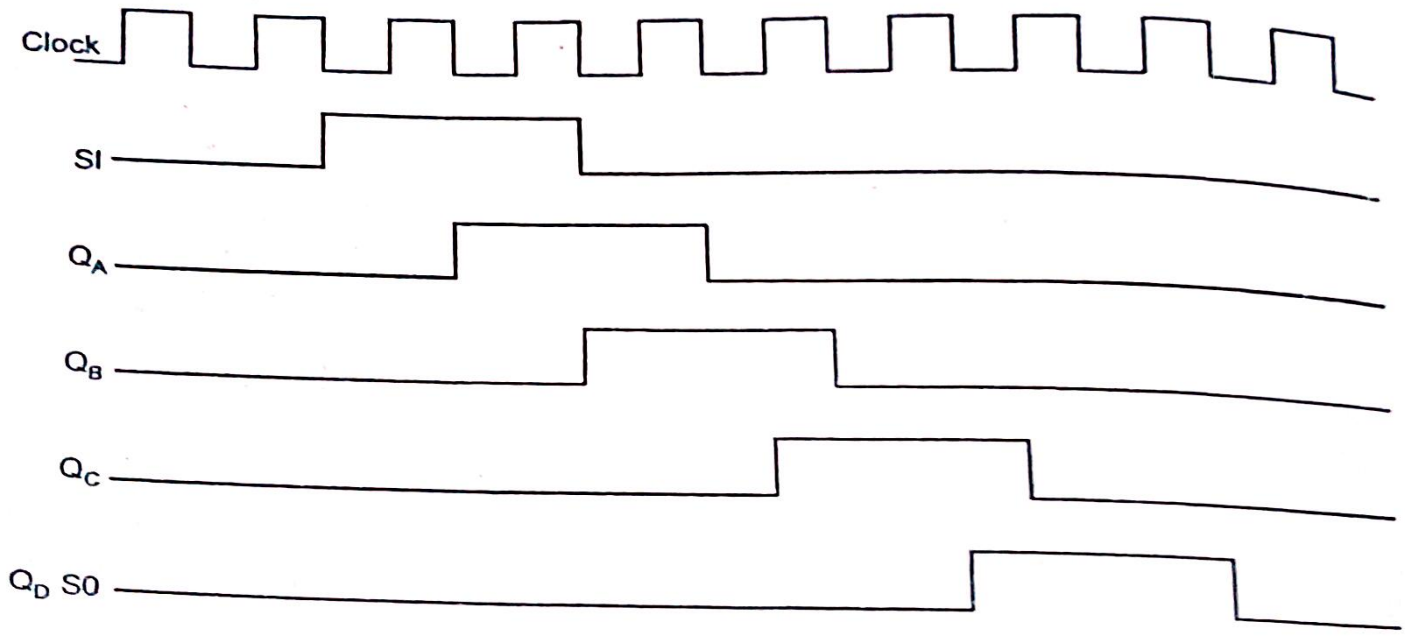


Figure 6.37

There are 4 types of shift registers, namely.

i) Serial in serial out (SISO)

ii) Serial in parallel out (SIPO)

iii) Parallel in serial out (PISO)

iv) Parallel in parallel out (PIPO).

A serial in serial out (SISO) register is shown in figure 6.38. It is a unidirectional (right) shift register.

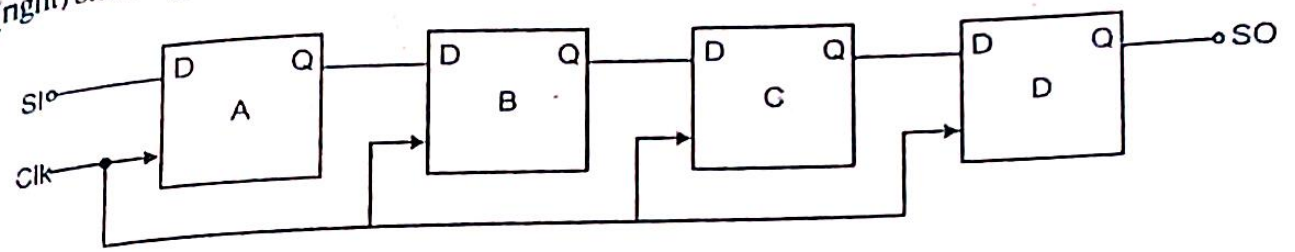


Figure 6.38

If the outputs are taken from the Q outputs of all the FFs simultaneously, then we get parallel outputs. And if the data is fed into the FFs through simultaneous inputs then they are parallel inputs.

The block diagrams of a shift register are shown in figure 6.39.

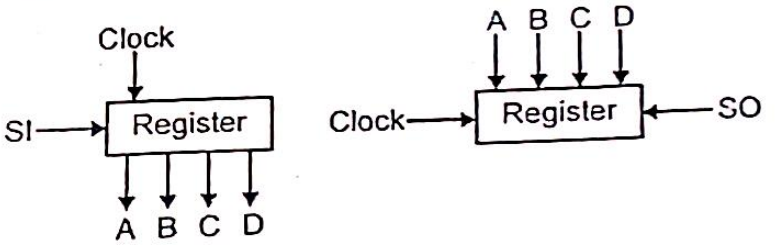


Figure 6.39

A shift register can shift the data either to the left or right. A left shift register is shown in figure 6.40.

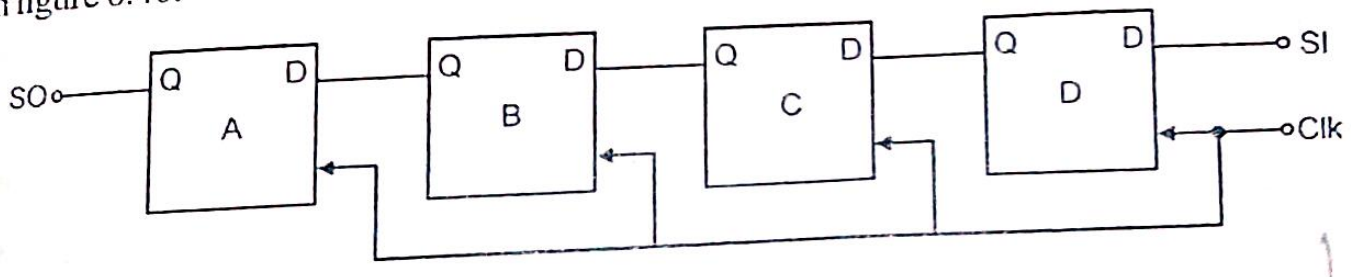


Figure 6.40

A shift registers with all the combinations [SISO, SIPO, PISO, PIPO], serial and parallel inputs and shifting left and right functions is given in the figure 6.41.

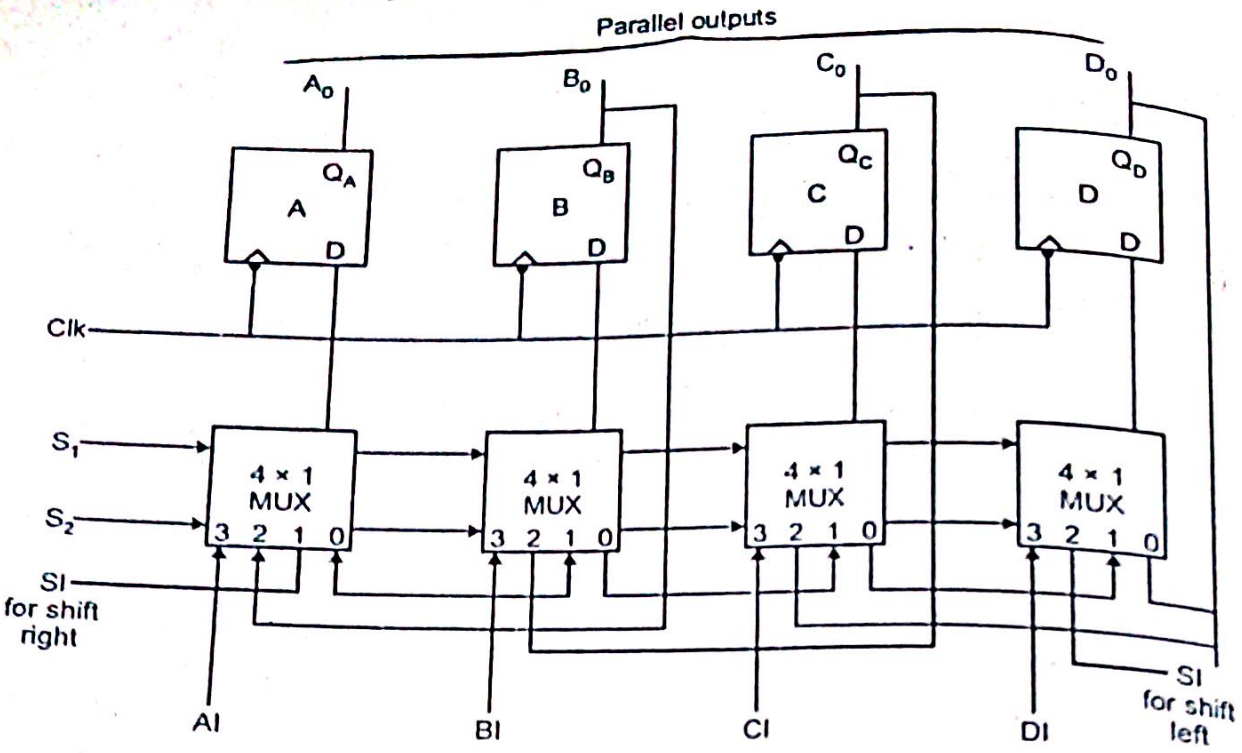


Figure 6.41

This is a bidirectional shift register. Any one combination and any one function can be selected at any instant, with the help of multiplexes. Depending on the selection lines in the multiplexer, any one of the following four functions are selected.

S_0	S_1	*MUX input*	*Function*
0	0	0	No change
0	1	1	Shift right
1	0	2	Shift left
1	1	2	Parallel load

Since this type of registers are available as Medium scale Integrated chip, they include all operations. Hence, they are also called as Universal Shift Register or Bidirectional shift register.

The timing diagram for the left and right shift register resembles the preceding ones. If it is a 4-bit shift register, it requires 4 clock pulses to shift a bit of bit of data to the serial output. After 4 clock pulses, the parallel output will contain the data. If a serial output is required, 3 more clock pulses are necessary. Hence to shift a 4-bit data to the serial output 2n-1 pulses are required.

6.9 Analog to Digital Converters

Analog to digital converters is a digital network which converts analog signal into digital signal. The basic block diagram of Analog to digital converter is shown in figure 6.42.

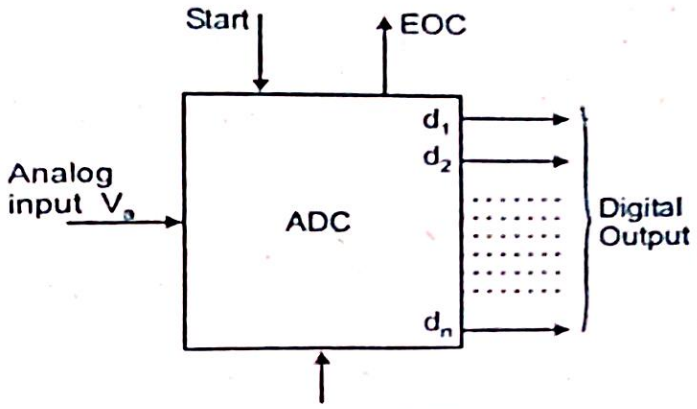


Figure 6.42

When analog input is given, an output binary data of $d_1, d_2, ... d_n$ is produced as function of D.

$$D = d_1 2^{-1} + d_2 2^{-2} + ... + d_n 2^{-n}$$

Analog to digital converter has two additional control signals. One is START and another one is EOC. START informs the converter when to start the conversion whereas EOC informs when the conversion is complete. EOC stands for 'End of conversion'.

Analog to digital converters are classified as two groups.

1. Direct type
2. Indirect type

Direct type are classified as

1. Flash (comparator) type converter.
2. Staircase type converter.
3. Tracking or servo converter.
4. Successive approximation type converter.