Lecture Notes

for

Sequential Logic Circuits

(PHYS4008: Electronics)

Dr. Pawan Kumar

(Assistant Professor)

Department of Physics

Mahatma Gandhi Central University

Motihari-845401, Bihar

Sequential Logic

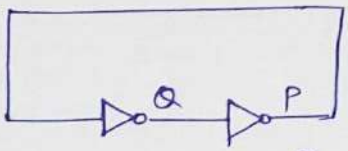
* In sequential logic, the outputs are determined not only by the current inputs but also by the sequence of inputs that led to the current state. In other words, the circuit has the characteristic of memory.
* Among the most important groups of sequential logic elements are various forms of multivibrators. These circuits can be divided into three types.
  1. Bistables: have two stable states
  2. Monostables: have one stable and one metastable state.
  3. Astables: which have no stable states
* The most widely used class of multivibrator is the bistable. These devices can be divided into
  * latches
  * edge-triggered flip-flops
  * pulse-triggered (master-slave) flip-flops.
* Each class of bistable may be divided into a range of devices with different operation characteristics. These are often described by symbolic names such as R-S, J-K, D or T-type devices
* Bistables are frequently used in groups to form registers or counters.
* Registers form the basis of computer memories and also used for serial to parallel conversion.
* Counters are used extensively for timing and sequencing functions. It is of two types
  i) Asynchronous or ripple counters
  ii) Synchronous counters.

* Asynchronous or ripple counters in which the clock for one stage is generated from the output of the previous stage. The result is a ripple effect as each stage changes in sequence.

* Synchronous counters in which all stages are clocked simultaneously so that all the outputs change at the same time.

* Both techniques can be used to produce counters that can count up or down.

* Regenerative switching circuit.

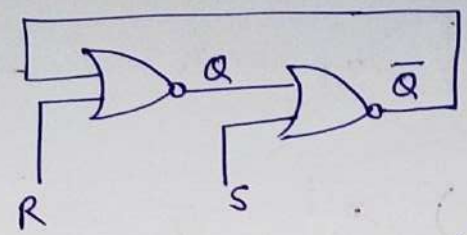


If the output of first inverter Q is equal to 1, this signal is fed to the input of the second inverter, making its output P equal to 0. This in turn forms the input to the first inverter, which makes its output 1. Thus, the circuit is stable with Q=1 and P=0. Alternatively, if Q is equal to 0, this corresponds to a stable state with Q=0 and P=1.

The circuit therefore has two stable states. It also has outputs of Q and P where $P = \bar{Q}$. It is considered to be a form of bistable multivibrator. However, it is of little practical importance because its state is stable until power is removed.

It is an example of regenerative switching.

The S-R latch



We now have a circuit with two input signals, R and S, and two outputs which are now labelled Q and $\bar{Q}$. If one input of a two input NOR gate is held at 0, the relationship bet^n the other input and output is that of an inverter. Therefore, if R and S are both held at 0, the circuit behaves in the same manner as the previous circuit of regenerative switching and will stay in whichever state it finds itself. We would call this situation the memory mode of the circuit.

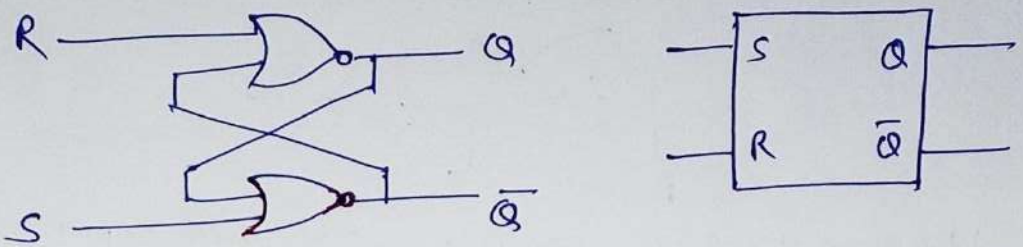
→ If the R is taken to 1 while S remain at 0, Q will be reset to zero regardless of its previous state. In turn, this will set $\bar{Q}$ to 1.

→ If now R is returned to 0, the circuit will reenter the memory mode and will stay in this state. Similarly, if S is taken to 1 while R remains at 0, $\bar{Q}$ will be cleared to 0 and Q will be set to 1. Again, if S returns to 0, the circuit will re-enter its memory mode

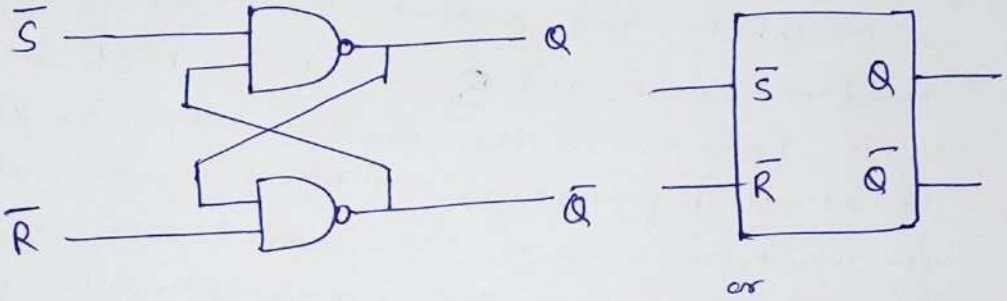
→* Thus the R input "reset" Q to 0 and S input "sets" Q to 1, while other input is at 0.

The above circuit is called a SET-RESET latch or S-R latch. It should be noted that S=R=1 results in both outputs being 0 which is prohibited.

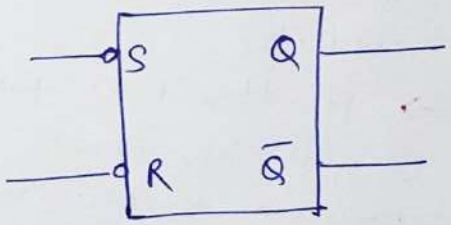
The S-R latch can be redrawn as follows



The above S-R latch can be produced using two NAND gates.



or



* Comparing the operation of NAND gate with that of a NOR gate, we note that while NOR gate resembles an inverter when one of its inputs is connected to zero, the NAND gate resembles an inverter when one of its inputs is connected to 1. Therefore, the memory mode of the circuit corresponds to both inputs being at 1.

* $\bar{S}$ (low) sets Q to 1 and $\bar{R}$ (low) resets Q to 0.

* Hence, these inputs are called active low inputs. and their names are given as $\bar{S}$ and $\bar{R}$ rather than S and R.

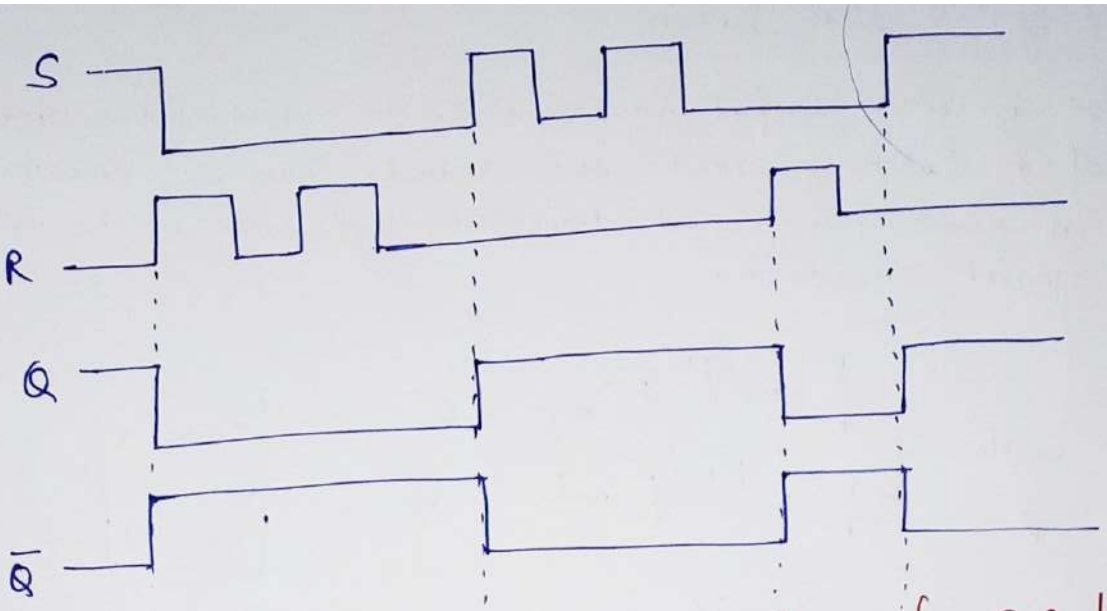
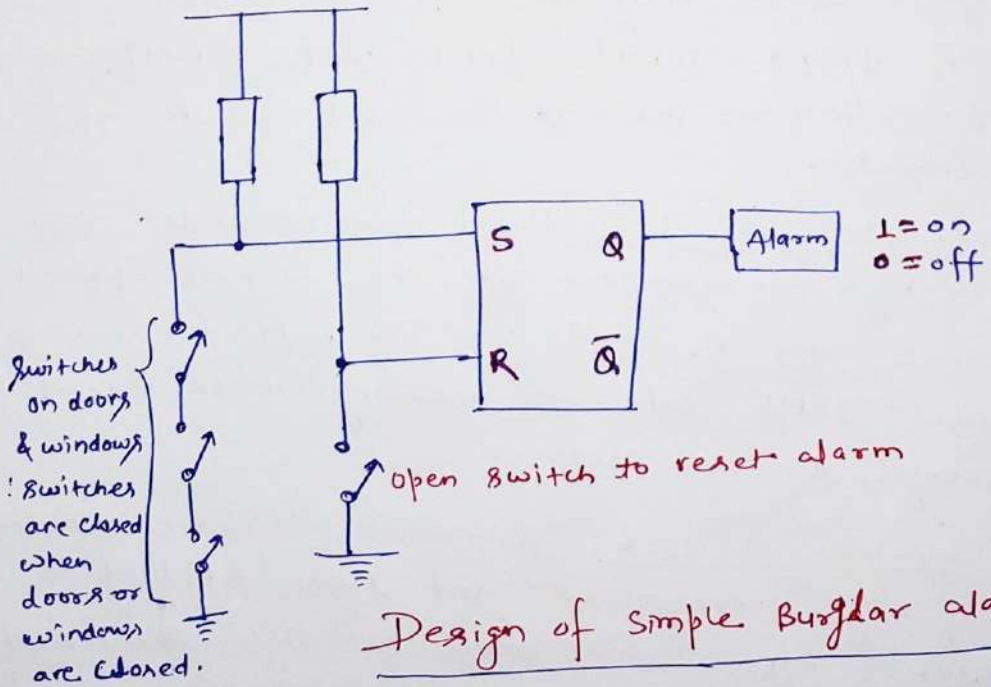


Fig: Simple input and output waveforms for S-R latch.

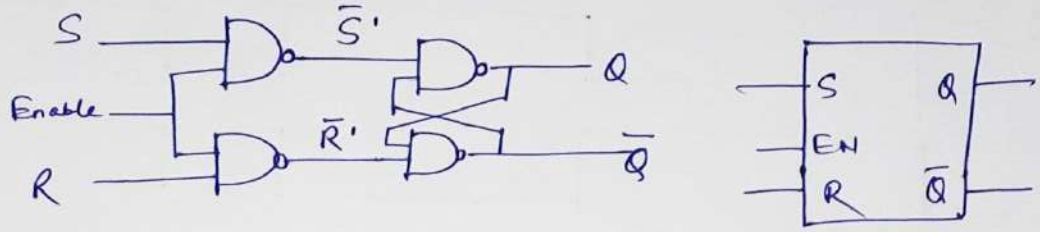


Design of simple Burglar alarm.

* we can use S-R latch for switch debouncing.
* All mechanical switches suffer from switch bounce.

The gated S-R latch

It is often useful to be able to control the operation of a latch so that the inputs can be enabled at some times and disabled at others. The following circuit can do it.

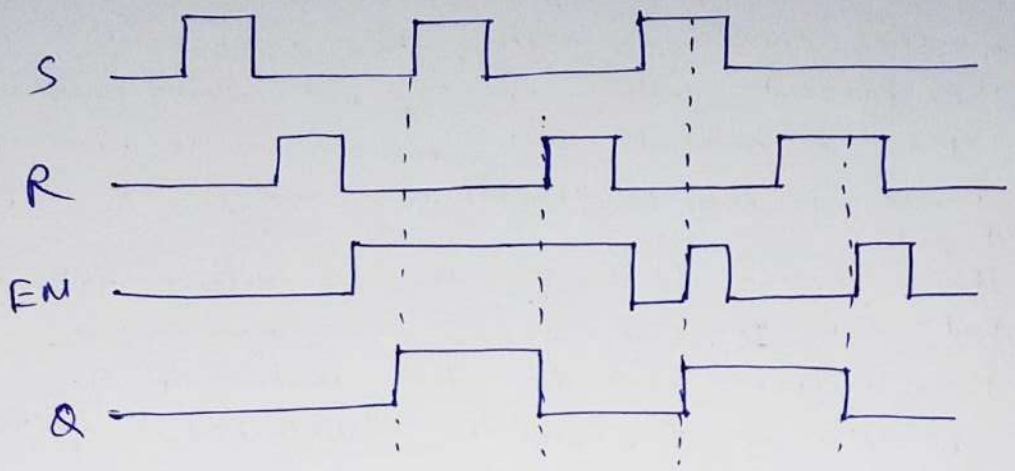


circuit — logic symbol

⟹ Two NAND gates are used to 'gate' the S and R input signals before they are applied to the latch. A third input, latch enable (EN), can be used to allow or inhibit the actions of the other inputs.

⟹ When enable signal is low, the signals $\bar{S}'$ & $\bar{R}'$ are both high, regardless of the signals applied to the S and R inputs. This puts the active low input latch into its memory mode, preventing any change to its states.

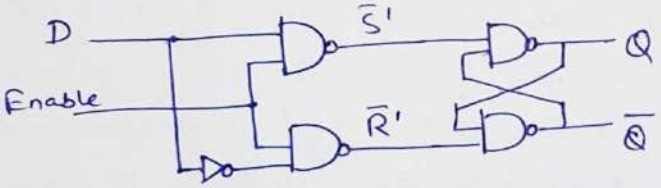
⟹ When enable is high, S & R signals are inverted by the gating arrangement and then applied to the latch. Thus, when enable is high, the circuit acts as a conventional active high input S-R latch, but when enable is low, the circuit ignores any signals applied to the S and R inputs.



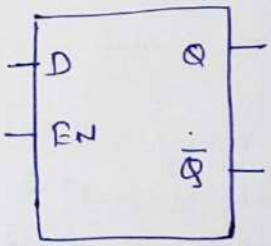
Sample input and output waveforms for a gated S-R latch.

The D latch

Another form of latch that is widely used is D latch. It is also known as the transparent D latch. This circuit has two inputs, D and EN, as shown in fig.



Circuit



logic symbol

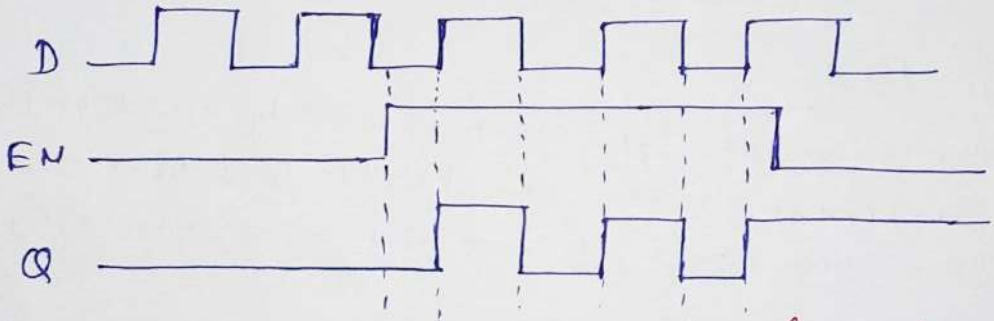
It uses a single signal D and its inverse $\bar{D}$ to acts as inputs to the gating network.

If the enable is low, the signals fed to the latch are both high and the latch is placed in its memory mode.

If the enable is high, D input determines the inputs to the latch as $\bar{S}'$ and $\bar{R}'$. If D is high, $\bar{S}'$ will be low and $\bar{R}'$ will be high and the latch will be set with Q=1. If D is low, $\bar{S}'$ will be high and $\bar{R}'$ will

be low, which will reset the latch with Q = 0. Thus, when the enable is high, the Q output takes the present value of D, but when enable is low, the Q output will remain in its present state. Hence, it may be thought to be analogue of sample and hold gate.

⇒ When the enable is high, the Q output follows the input date D, but when the enable goes low, the output remember the value of D. The operation of D-latch is illustrated in following figure.

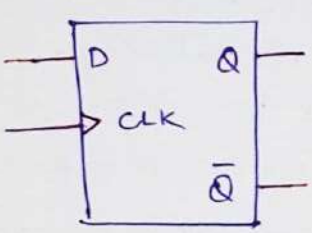


Sample input and output waveforms for a D latch

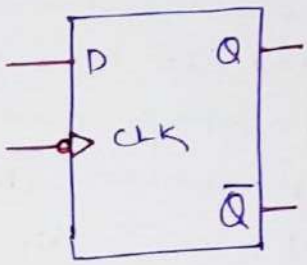
⇒ In many situations it is necessary to synchronise the operation of a number of different circuits and it is useful to be able to control precisely when a circuit will change state. Some bistables are constructed so that they only change state on the application of a trigger signal. This is defined as the rising or falling edge of an input signal called the clock. These devices are termed as edge-triggered bistables or more commonly flip-flops. These are divided into those that are triggered by the rising edge of the clock signal (so-called positive edge triggered devices) and those that are triggered on the falling edge of clock (negative edge-triggered devices)

Flip-flops are available in a number of different forms, including the S-R flip-flop and D flip-flop which are edge-triggered versions of the latches discussed earlier.

The circuit symbols used for these circuits are similar to those of corresponding latch, except that the enable input is replaced with a clock input. The clock input is conventionally indicated by a triangle, while an inverting circle is used to show negative edge-triggered device.



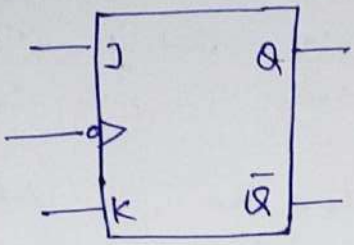
Positive edge triggered



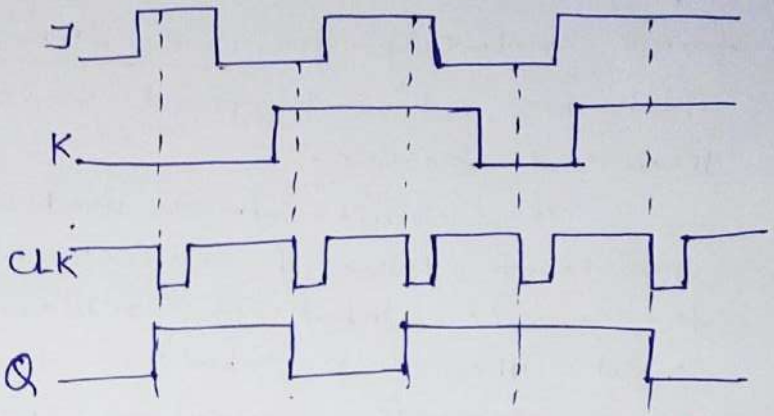
Negative edge triggered

J-K flip-flop

It is perhaps one of the most widely used form of bistable. As its name suggests, it has two inputs, J and K. Taking J to 1 while K is at 0 sets Q to 1, whereas taking K to 1 while J is at 0 reset Q to 0. As in the S-R device, when neither input is active, the circuit is in its memory state, but the operation of the arrangement is different when both inputs are active simultaneously. This is the ambiguous situation in the case of S-R bistable and so it is avoided. But in the case of the J-K device, when both inputs are active, the circuit changes state (or toggles) when a trigger event occurs.

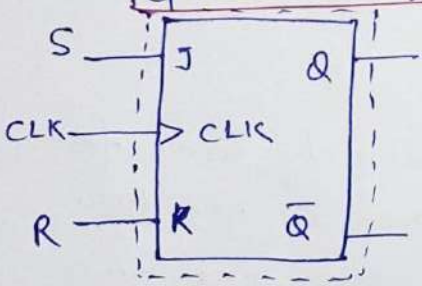


Logic Symbol

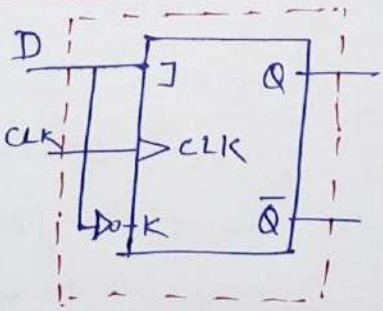


Sample input and output waveforms

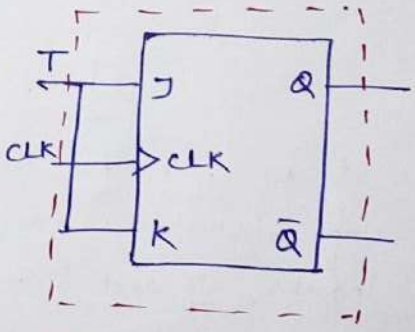
One of the reasons for the J-K flip-flop being so widely used is its great versatility. Several different operating modes are possible; including using it to reproduce the functions of other types of flip-flop. J-K flip-flop can also be used as replacement of SR device.



(a) S-R flip-flop



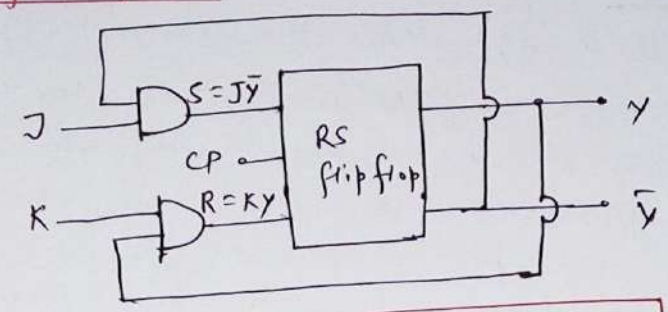
(b) D flip-flop



(c) T flip-flop

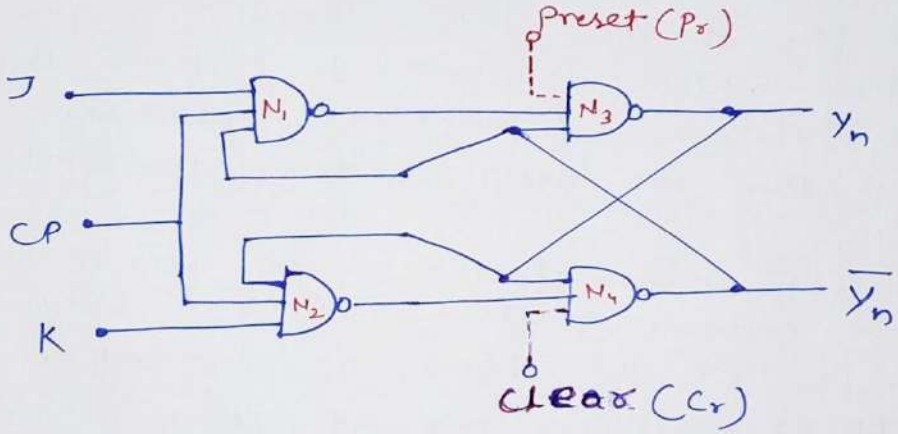
* When J and K inputs are joined to form a single input, T; it is known as T flip-flop. If T = 1, both J and K are 1, and the device will toggle on every clock pulse. If T is 0, the device is in its memory state (mode) and simply stay in the present state. This is known as Toggle flip-flop (T flip-flop)

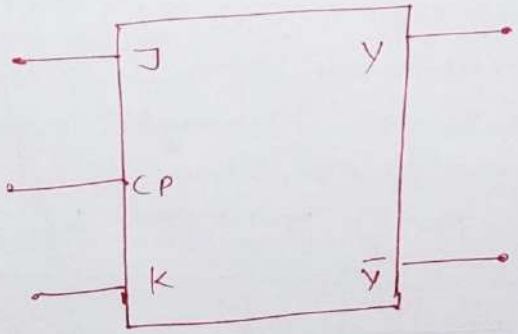
JK flip-flop



for $J = K = 1$, $Y_{n+1} = \bar{Y}_n$

or





Logic symbol of a JK flip-flop.

J_n	K_n	Y_{n+1}
0	0	Y_n
1	0	1
0	1	0
1	1	$\bar{Y}_n$

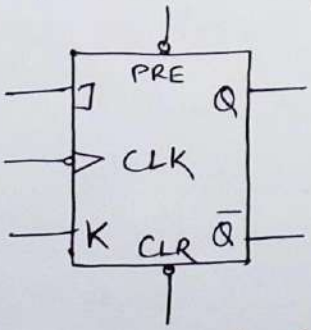
Reduced truth table of JK flip flop.

Asynchronous Inputs

⇒ We have seen that in flip-flops, the control inputs (for example, the J and K inputs in a J-K flip-flop) affect the operation of the circuit only at the moment of an appropriate transition of the clock signal (CLK). We therefore refer to these inputs as synchronous; because their operation is synchronised to the clock input.

⟹ In many applications, it is advantageous to be able to set or clear the output at other times, independently of the clock. Therefore, some devices have additional inputs to perform these functions. These are termed as asynchronous inputs as they are not bound by the state of the clock.

⟹ However, IC manufacturers are unable to agree on common names for these inputs; so they may be called PRESET and CLEAR, DC SET and DC CLEAR, SET and RESET or DIRECT SET and DIRECT RESET. Here, we use the names PRESET (PRE) and CLEAR (CLR). As with control input, these lines can be active high or active low, although more often than not they are active low.



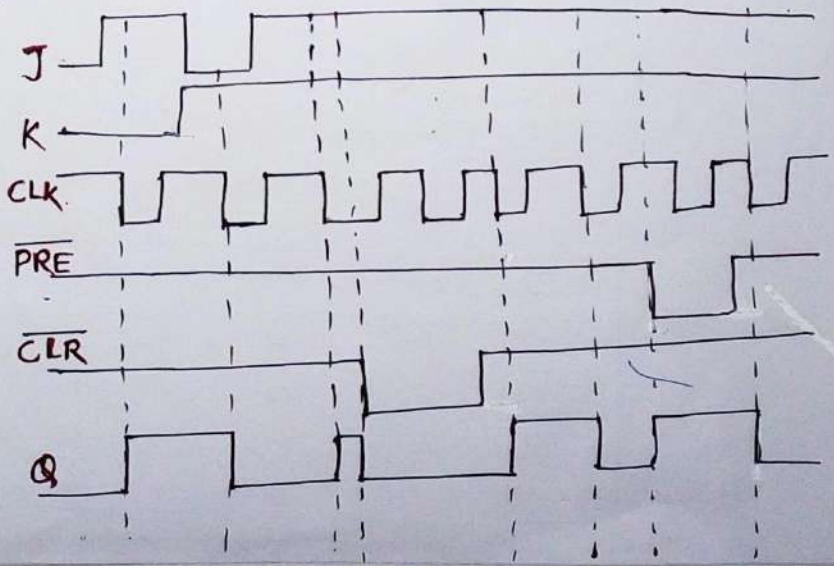
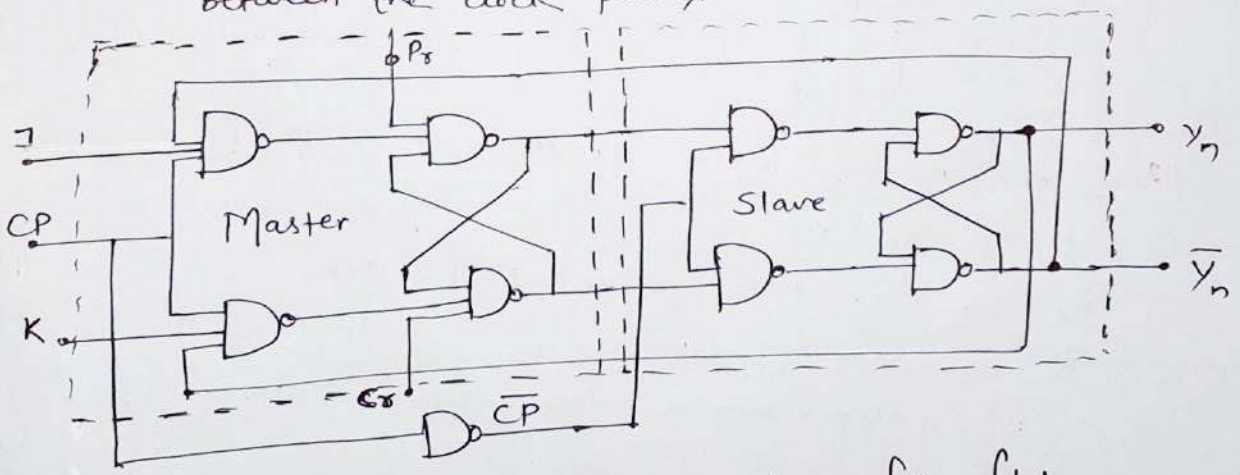


Fig: Logic symbol of J-K flip-flop with PRESET and CLEAR

Master-Slave JK Flip-flop:

It is a cascade of two RS flip flops where the output of the second, called the slave, is fedback to the input of the first, called the master. Positive clock pulses are applied directly to the master and the same after inversion are applied to the slave. The master is enabled when $P_s = 1$, $C_r = 1$ and $CP = 1$. Since $\overline{CP} = 0$, the slave is inhibited and so cannot change state. Thus, the Y_m remains invariant during the time t_p. Consequently, the race-around condition does not occur.

When $CP = 0$, the master is inhibited and the slave is enabled since $\overline{CP} = 1$. The slave is RS flip flop. When $S = Y_m = 1$ and $R = \overline{Y}_m = 0$, then $Y = 1$ and $\overline{y} = 0$. Similarly, when $S = Y_m = 0$ and $R = \overline{Y}_m = 1$ then $Y = 0$ and $\overline{y} = 1$. That is, the value of Y_m is transmitted to the output Y in the time interval between the clock pulses.



A JK Master-Slave flip flop

* It may seem that this arrangement would suffer from 'race' problem. When the clock goes high, the master's output might change just as the slave is being disabled. In fact, this is not the case. It is designed such that the circuit ensures that the delay produced by master latch is greater than that of inverter used for clock pulse.

Registers

➡ A register consists of a group of flip-flops that can store binary information. Since a flip-flop can store 1 bit of information, an n-bit word register requires n number of flip-flops connected in cascade, i.e. the output of one flip-flop is the input of the following, and so on. Besides the flip-flops, a register also contain additional gates to execute the data processing operations.

➡ The binary information of a register can be shifted either to the right or to the left. It is thus commonly known as shift register. A common clock pulse applied to all the flip-flops causes the shift from one stage to next synchronously.

➡ The shifting in or shifting out of data can be carried out through serial shifting and parallel shifting. In serial shifting data bits are allowed to move through the register from one bit to the next in succession. Here the data enter the register at one extreme end (either LSB or MSB) and leave it at the other.

* In parallel shifting, the data are made to enter or leave involving all the bits simultaneously. Parallel shifting process is faster than serial shifting.
* Parallel shifting are mostly used in digital computers.

Depending on the mode of data shifting, a shift register can be classified into four types.

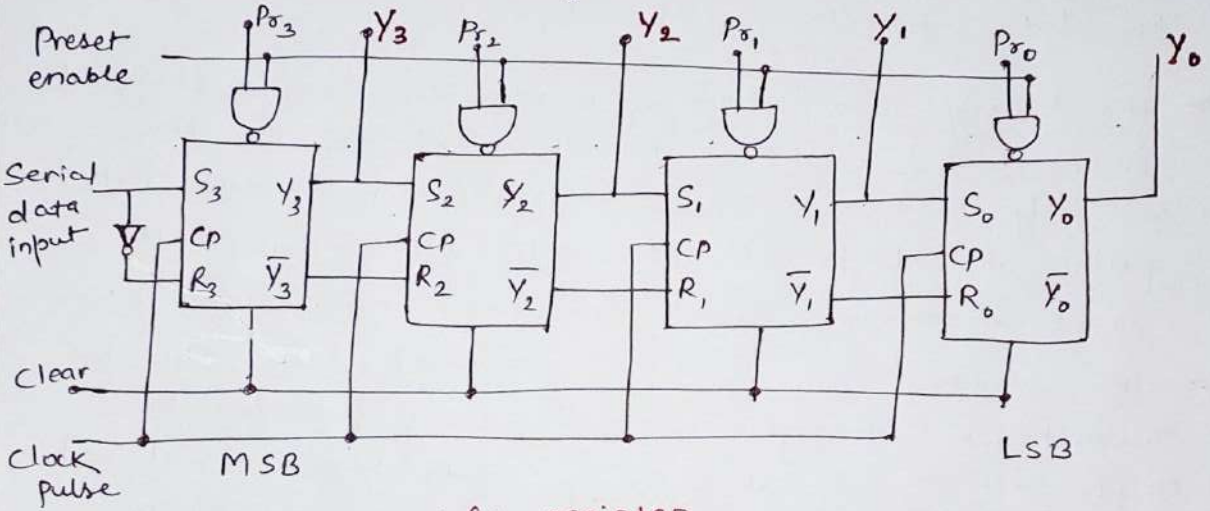
i) Serial-in serial-out
ii) Serial-in parallel-out
iii) Parallel-in Serial-out
iv) Parallel-in parallel-out

We will consider the operation of 4-bit register of serial-in parallel-out type as shown in Fig.

⇒ Each flip-flop is of the SR (or JK) master-slave type.

⇒ The stage which stores the most significant bit (MSB) is transformed into a D-type latch by connecting an inverter between S and R.

⇒ The binary serial data, say, 1011 (LSB), is to be entered into the register via the S terminal



A 4-bit shift register

To start with, the flip-flops are cleared by applying a clear (C_r) input so that every output Y_0, Y_1, Y_2 & Y_3 is 0. Then set $C_r = P_r = 1$ (by maintaining the preset enable at 0). The serial data train and the clock pulse are now switched on. The LSB which is 1 here, is entered into FF3 when clock pulse (CP) assumes a 1 from a 0 by the action of a D-type flip-flop. Thus after clock pulse, $Y_3 = 1$ and $Y_2 = Y_1 = Y_0 = 0$.

When the second clock pulse appears, the state of Y_3 is shifted to the master latch of FF2 by the operation of an SR flip flop. At the same time, the next bit 1 enters the master of FF3. At the end of 2nd clock pulse, the bit in each master shifts to

its slave and results in $Y_3 = 1$, $Y_2 = 1$ and $Y_1 = Y_0 = 0$. The state of the register after each pulse is shown in following table.

It is evident that after the third clock pulse, Y_2 has shifted to Y_1, Y_3 to Y_2, and the third input bit 0 has entered FF3, so that $Y_3 = 0$. Similarly after the 4th clock pulse, Y_3 has shifted to Y_2, Y_2 to Y_1 and Y_1 to Y_0 and 4th input bit 1 has entered FF3 resulting in $Y_3 = 1$.

Clearly, the input word 1011 has been installed in the register after 4th clock pulse. The clock pulses are stopped after the word is registered. The outputs may be read simultaneously since they are available on separate lines. As the data enter the system serially, but come out in parallel, it is called a serial-in parallel-out register. Shift registers are available in IC forms.

Clock pulse	Serial data input	Y_3	Y_2	Y_1	Y_0
		0	0	0	0
1	1	1	0	0	0
2	1	1	1	0	0
3	0	0	1	1	0
4	1	1	0	1	1

Reading of the shift register after each clock pulse.

Counters

A counter is a sequential circuit that keeps a record of clock pulses sent through it. Like a register, a counter also consists of a group of flip-flops. However, a counter has a characteristic internal sequence of states through which it passes when a series of clock pulses are fed to it. A register, on the other hand, has no such characteristic sequence. the states of a register are controlled by the data applied to it.

The counters are divided into two categories:

i) ripple (asynchronous) counters

ii) Synchronous counters

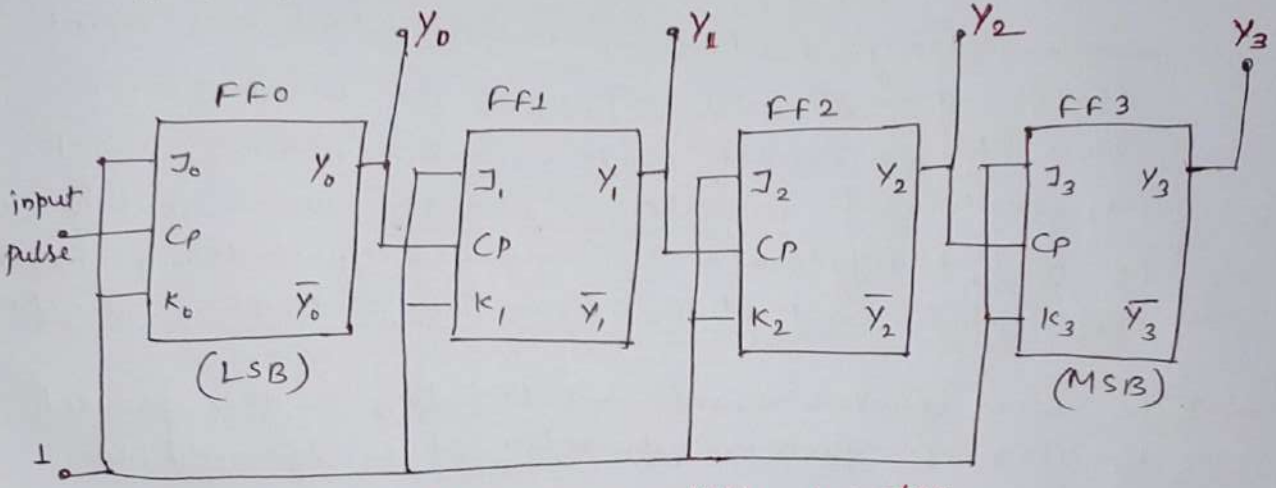
In a ripple counter, all the flip flop donot receive their triggering pulses simultaneously. Here the first flip flop only get the incoming clock pulse. The output of the first flip flop provides the clock pulse for the second flip-flop, the output of the second provides the clock pulse for the third and so on.

While in a synchronous counter, all the flip-flops of the counter receive their triggering pulse simultaneously, since the same clock pulse input triggers them all.

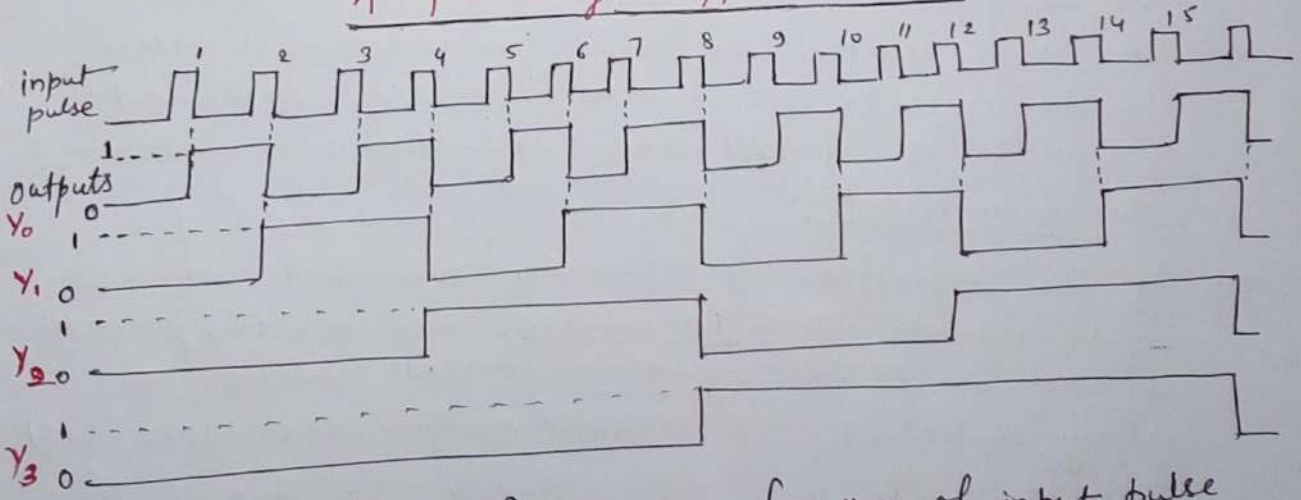
* Ripple (Asynchronous) Counters: Consider a chain of four JK master-slave flip-flops with the output y of each flip-flop providing the clock input of the following flip-flop as shown in fig.

The pulses to be counted are fed to the clock input of the first flip-flop (FF0) which displays the LSB of the binary number. J and K of all the flip flop are tied to the supply voltage giving J=K=1. Hence the master changes the state each

time the waveform at its clock input rises from 0 to 1, and this state of the master is transferred to the slave when the clock input drops from 1 to 0. Thus Y_0 changes state at the trailing edge of each pulse and any of the other Y's suffers a transition only when the output of its previous flip-flop drops from 1 to 0. This type of negative transition 'ripples' down the counter from the first flip flop FF0 (LSB) to the last flip-flop FF3 (MSB).



A four stage ripple counter



Output waveforms as a function of input pulse

The figure shows that after the passage of first input pulse the output of the counter reads 0001, after the second pulse it reads 0010 and so on.

⟹ Since there are four flip-flops and the output of each flip-flop has two states (0 and 1), the counter has 2^4 (=16) states. After 16 pulses the counter output shows 0000, i.e. the counter returns to its original state. Thus 16 pulses in succession form a cycle.

⟹ Clearly, a chain of n flip-flops will count states up to 2^n (= N, say) before the counter come back to its initial state. Such a chain is called a counter of modulo 2^n or N, where modulo (or mod) represents the number of states of counter. The counter is read by means of a decoder circuit.

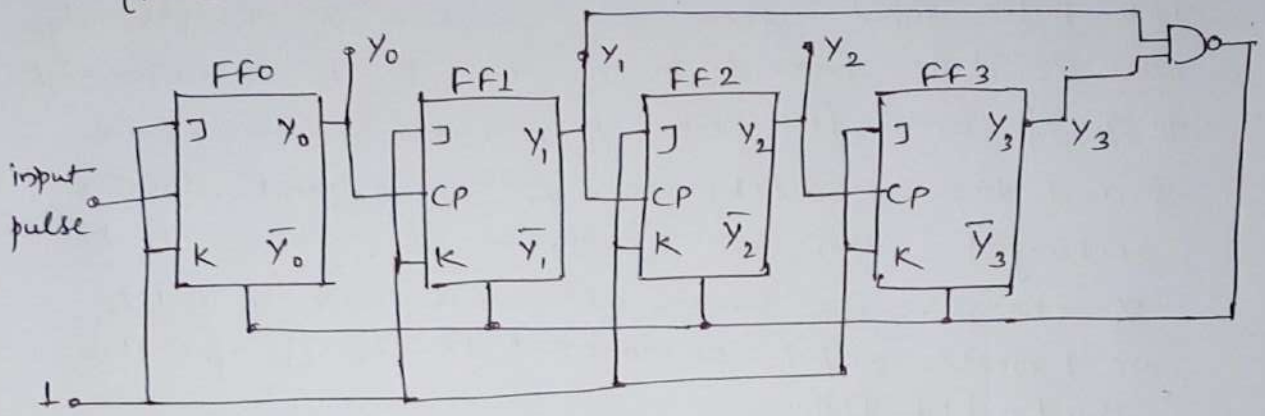
⟹ The above counter counts input pulses in the forward direction, i.e. from 0 upwards. It is termed as up counter.

⟹ The above counter can count in reverse direction if the outputs are taken from the complement terminals $\bar{y}$ of all the flip-flops. It is termed as down counter.

⟹ If the sequence of states is truncated and the cycle repeat after 10 states, the resulting circuit becomes a decade counter or mod 10 counter. The binary equivalent of decimal number 10 is 1010 (LSB) and so $Y_0 = 0$, $Y_1 = 1$, $Y_2 = 0$ and $Y_3 = 1$. Thus at the count 10, the outputs $Y_1 = 1$ and $Y_3 = 1$. These two outputs are applied to a NAND gate, the output of which feeds all the clear inputs in parallel as shown in next figure.

Consequently, whenever $Y_1 = Y_3 = 1$, the output of the NAND gate becomes 0, and all the flip-flops reset to 0 after 10 input cycles.

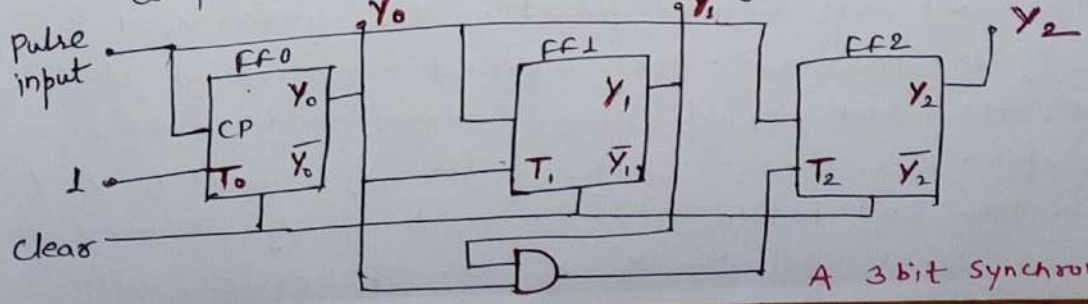
If it is required to count 1000 ($=10^3$), then three decade counter units are cascaded.



A decade counter

ii) Synchronous Counter: All the flip-flops are clocked simultaneously. Hence propagation delay time is reduced significantly. The other advantage of the synchronous counter is absence of unwanted spikes at the decoder output because all flip-flops change states simultaneously.

A 3-bit synchronous counter is shown below. Each flip-flop is T-type, formed by connecting the J and K inputs together of a JK flip-flop. When T=0, there is no change of state of the output when the clock pulse is applied. When T=1 the flip flop is complemented with each incoming clock pulse.



A 3 bit Synchronous Counter

The operation of the counter circuit shown in figure is explained with reference to the following table which displays the counting sequence of a 3bit binary counter over a cycle.

The output Y_0 (LSB) changes state for each clock pulse input. Also, the output Y_1 changes only when Y_0 drops from 1 to 0. So if Y_0 is connected to T_1 of next flip flop, Y_1 will alter its state from 1 to 0 (or 0 to 1) when $Y_0 = T_1 = 1$ and remains unchanged when $Y_0 = T_1 = 0$. Table shows that Y_2 changes its state whenever both Y_1 and Y_0 are 1, whose output is connected to the T_2 of the third flip-flop.

No of pulse input	output states of Y_2	Y_1	Y_0
0	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

← Counting sequence of a 3-bit counter

* If the output Y_0 of figure (SIPO shift register) is connected to the serial data input, the shift register is then converted to a circulating memory known as a ring counter. Suppose all the flip-flops are cleared and only FF0 is preset. This gives $Y_0 = 1$ & $Y_1 = Y_2 = Y_3 = 0$. Now with the application of first clock pulse, the state Y_0 of FF0 is transferred to FF3. Thus after the first clock pulse $Y_3 = 1$ and $Y_2 = Y_1 = Y_1 = 0$. After the 2nd pulse(clock), the bit 1 of Y_3 will move to Y_2. i.e. $Y_2 = 1$ and $Y_3 = Y_1 = Y_0 = 0$. Therefore, with

every clock pulse input, the state 1 will move progressively around the ring formed by Y_3, Y_2, Y_1, Y_0.

* Applications: To count the objects by loading them single-file on a conveyor belt betn a light source and a photoelectric cell. Here, each object, while passing across light source and photoelectric cell produces an electric signal which is fed to the counter.

References:

1. Digital Fundamentals by Thomas Floyd
2. Digital principles and Applications by A. P. Malvino and D. P. Leach
3. Modern Digital Electronics by R.P. Jain
4. Digital Design by M. Mano
5. Electronics Fundamentals and Applications by D. Chattopadhyay and P.C. Rakshit
6. Electronic Devices and Circuits by J. Millman and C.C. Halkias
7. Integrated Electronics by J. Millman and C.C. Halkias