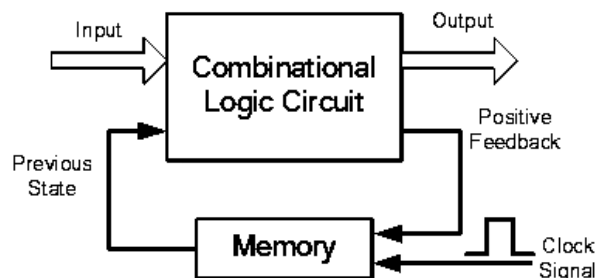


Unit 5

SEQUENTIAL LOGIC

Sequential logic circuits are circuit having memory elements and depend upon the feedback from the previous output. In other words, the output state of a “sequential logic circuit” is a function of the following three states, the “present input”, the “past input” and/or the “past output”.

Sequential logic circuits are generally termed as two state or Bistable devices which can have their output or outputs set in one of two basic states, a logic level “1” or a logic level “0” and will remain “latched” (hence the name latch) indefinitely in this current state or condition until some other input trigger pulse or signal is applied which will cause the bistable to change its state once again. The word “Sequential” means that things happen in a “sequence”, one after another and in Sequential Logic circuits, the actual clock signal determines when things will happen next. Simple sequential logic circuits can be constructed from standard Bistable circuits such as: Flip-flops, Latches and Counters and which themselves can be made by simply connecting together universal NAND Gates and/or NOR Gates in a particular combinational way to produce the required sequential circuit.



Difference between Combinational logic circuit and Sequential logic circuit.

Combinational logic circuit	Sequential logic circuit
It depends only on the present input	It depends on the present input and previous output
It does not have any memory element	It has memory elements to store bits of information
It is implemented using logic gates only. (AND,OR,NOT,NAND,NOR,XOR,XNOR)	It is implemented using flipflops (RS,D,T,JK) and logic gates
It does not use clock input	It uses clocked input. (positive or negative pulse)

It is mainly used to implement boolean expression	It is mainly used in registers, shift register, counters etc.
---	---

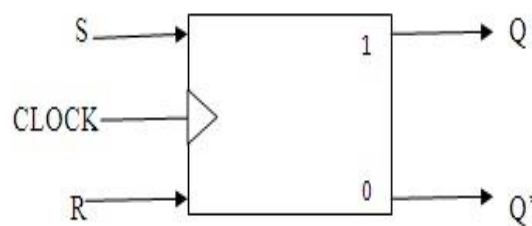
FLIP FLOPS

A Flip flop is a smallest memory element and can store 1 bit of information. Flip flop is a sequential circuit which generally samples its inputs and changes its outputs only at particular instants of time and not continuously. Flip flop is said to be edge sensitive or edge triggered rather than being level triggered like latches.

RS FLIP FLOPS

There are majorly 4 types of flip flops, with the most common one being RS or SR flip flop. This simple flip flop circuit has a set input (S) and a reset input (R). In this circuit when you Set “S” as active (1) and R as low (0), the output Q would be high (1) and Q’ will be low (0). When R is high (1) and S is low (0), then the output Q will be low (0) and Q’ being high. When both the R and S is low, there will be no change in the output but when both these lines are high, the circuit will go to indeterminate state. In other words, it will be unstable and the output cannot be determined. This is one of the disadvantages of RS flipflop. So, once the outputs are established, the wiring of the circuit is maintained until “S” or “R” go high, or power is turned off. The truth table of SR Flip Flop is highlighted below.

Block Diagram



Truth Table

S	R	Q	Q'
0	0	No Change	
0	1	0	1
1	0	1	0
1	1	Unstable state	

Logic Circuit

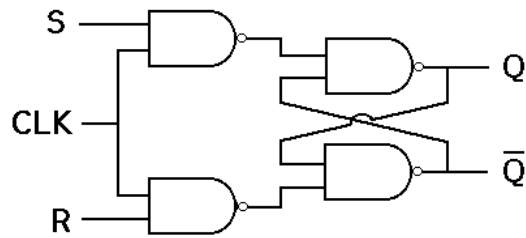


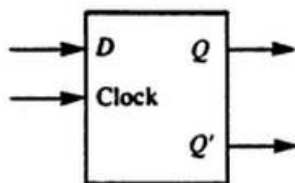
Fig. Clocked RS flipflop

In this circuit diagram, the output is changed (i.e. the stored data is changed) only when you give an active clock signal. Otherwise, even if the S or R is active the data will not change. Let's look at the types of flip-flops to understand better.

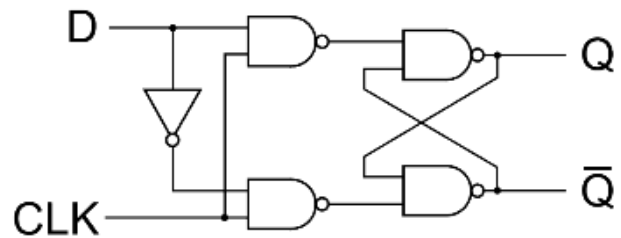
D FLIP FLOPS

D flip flop is a better alternative that is very popular with digital electronics. They are commonly used for counters and shift-registers and input synchronization. It is also called data flipflop in which the input data is saved as the output bit.

Block Diagram



Logic Circuit



D Flip-Flop

In a D flip flop, the output can be only changed at the clock edge, and if the input changes at other times, the output will be unaffected. When D=0 (low), the output will be 0. When D=1 (high), the output will be 1. So, D flipflop is delay flipflop which produce the same output but after the clock pulse change.

Truth Table

D Flip Flop		
Input	Output	
D	Q	Q [^]
0	0	1
1	1	0

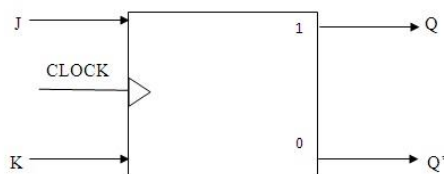
Clock	D	Q	Q'
1 » 0	0	0	1
0 » 1	0	0	1
1 » 0	1	0	1
0 » 1	1	1	0

The change of state of the output is dependent on the rising edge of the clock. The output (Q) is same as the input and can only change at the rising edge of the clock. Q_t is the previous output and Q_{t+1} is the present output.

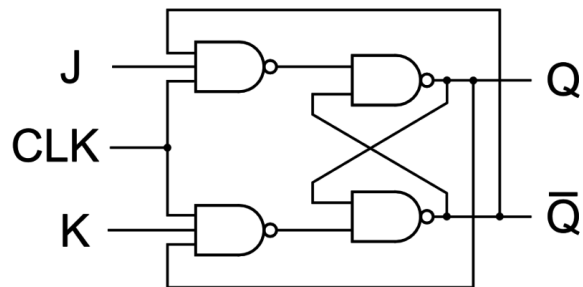
JK FLIP FLOPS

Due to the undefined state in the SR flip flop, another flip flop is required in electronics. The JK flip flop is an improvement on the SR flip flop where $S=R=1$ is not a problem.

Block Diagram



Circuit Diagram



The input condition of $J=K=1$, gives an output inverting the output state. However, the outputs are the same when one tests the circuit practically. In simple words, If J and K data input are different (i.e. high and low) then the output Q takes the value of J at the next clock edge. If J and K are both low then no change occurs. If J and K are both high at the clock edge then the output will toggle from one state to the other. JK Flip Flop can function as Set or Reset Flip flop.

Truth Table

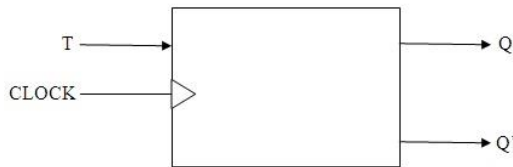
Truth Table

J	K	CLK	Q
0	0	↑	Q_0 (no change)
1	0	↑	1
0	1	↑	0
1	1	↑	\bar{Q}_0 (toggles)

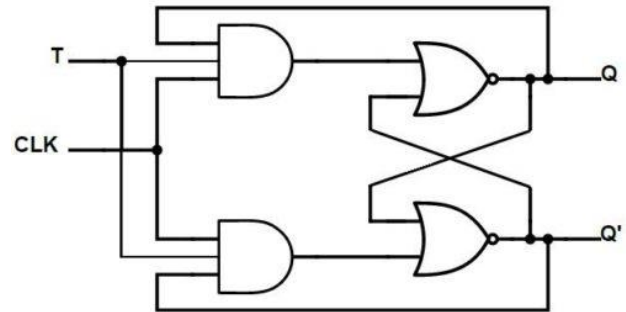
T FLIP FLOPS

A T flip flop is like JK flip-flop. These are basically a single input version of JK flip flop. This modified form of JK flip-flop is obtained by connecting both inputs J and K together. This flip-flop has only one input along with the clock input.

Block Diagram



Logic Circuit



T flipflop

These flip-flops are called T flip-flops because of their ability to complement its state (i.e.) Toggle, hence the name Toggle flip-flop. So, when T is 0, the output(Q(t+1)) will remain unchanged and when T=1, the output will be the complement of the previous output(Q).

Truth Table

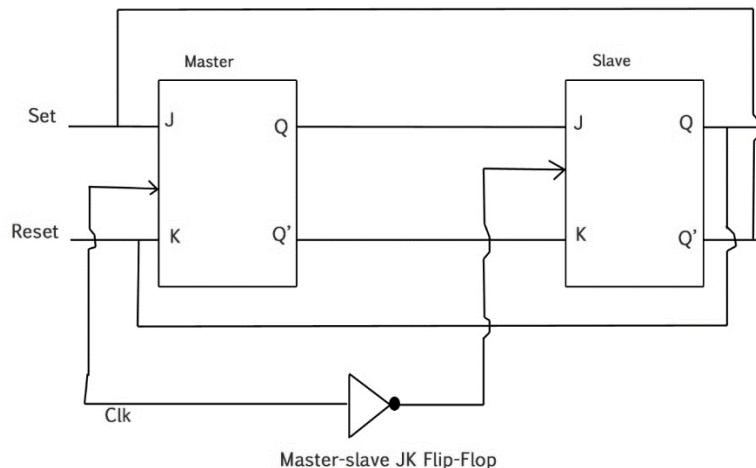
CLK	T	Q _{n+1}
↑	0	Q _n
↑	1	Q _n '

MASTER SLAVE FLIP FLOP

Master Slave flip flop are the cascaded combination of two flip-flops among which the first is designated as master flip-flop while the next is called slave flip-flop. Here the master flip-flop is triggered by the external clock pulse train while the slave is activated at its inversion i.e. if the master is positive edge-triggered, then the slave is negative-edge triggered and vice-versa. This means that the data enters into the flip-flop at leading/trailing edge of the clock pulse while it is obtained at the output pins during trailing/leading edge of the clock pulse. Hence a master-slave flip-flop completes its operation only after the appearance of one full clock pulse for which they are also known as pulse-triggered flip-flops.

The Master-Slave Flip-Flop is basically a combination of two JK flip-flops connected together in a series configuration. Out of these, one acts as the “master” and the other as a “slave”. The output from the master flip flop is connected to the two inputs of the slave flip flop whose output is fed back to inputs of the master flip flop.

In addition to these two flip-flops, the circuit also includes an inverter. The inverter is connected to clock pulse in such a way that the inverted clock pulse is given to the slave flip-flop. In other words, if $CP=0$ for a master flip-flop, then $CP=1$ for a slave flip-flop and if $CP=1$ for master flip flop then it becomes 0 for slave flip flop.



Working of a master slave flip flop –

When the clock pulse goes to 1, the slave is isolated; J and K inputs may affect the state of the system. The slave flip-flop is isolated until the CP goes to 0. When the CP goes back to 0, information is passed from the master flip-flop to the slave and output is obtained.

Firstly, the master flip flop is positive level triggered and the slave flip flop is negative level triggered, so the master responds before the slave.

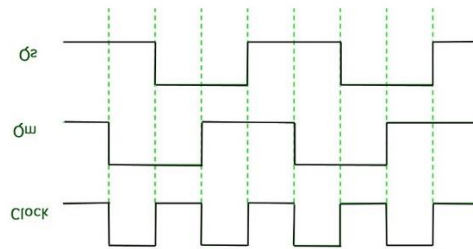
If $J=0$ and $K=1$, the high Q' output of the master goes to the K input of the slave and the clock forces the slave to reset, thus the slave copies the master.

If $J=1$ and $K=0$, the high Q output of the master goes to the J input of the slave and the Negative transition of the clock sets the slave, copying the master.

If $J=1$ and $K=1$, it toggles on the positive transition of the clock and thus the slave toggles on the negative transition of the clock.

If $J=0$ and $K=0$, the flip flop is disabled and Q remains unchanged.

Timing Diagram of a Master flip flop –



When the clock pulse is high, the master is operational but not the slave thus the output of the slave remains low till the clock remains high. When the clock is low, the slave becomes operational and remains high until the clock again becomes low. Toggling takes place during the whole process since the output is changing once in a cycle. This makes the Master-Slave J-K flip flop a Synchronous device as it only passes data with the timing of the clock signal.

EDGE TRIGGER FLIP FLOP

The output of a flip flop can be changed by bring a small change in the input signal. This small change can be brought with the help of a clock pulse or commonly known as a trigger pulse.

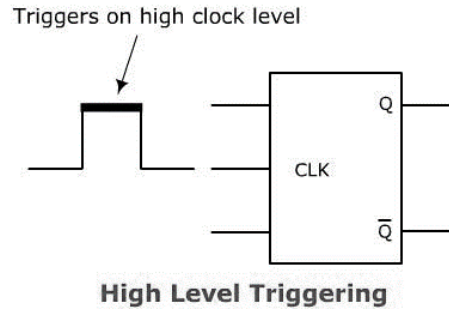
When such a trigger pulse is applied to the input, the output changes and thus the flip flop is said to be triggered. Flip flops are applicable in designing counters or registers which stores data in the form of multi-bit numbers. But such registers need a group of flip flops connected to each other as sequential circuits. And these sequential circuits require trigger pulses.

The number of trigger pulses that is applied to the input of the circuit determines the number in a counter. A single pulse makes the bit move one position, when it is applied onto a register that stores multi-bit data.

There are mainly four types of pulse-triggering methods. They differ in the manner in which the electronic circuits respond to the pulse. They are

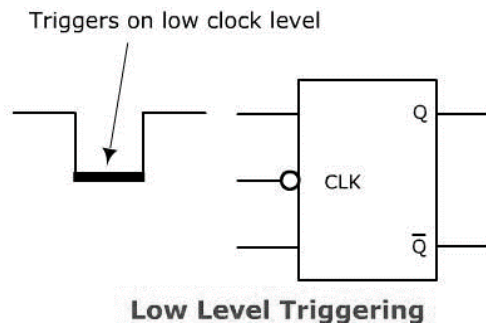
1. High Level Triggering

When a flip flop is required to respond at its HIGH state, a HIGH level triggering method is used. It is mainly identified from the straight lead from the clock input. Take a look at the symbolic representation shown below.



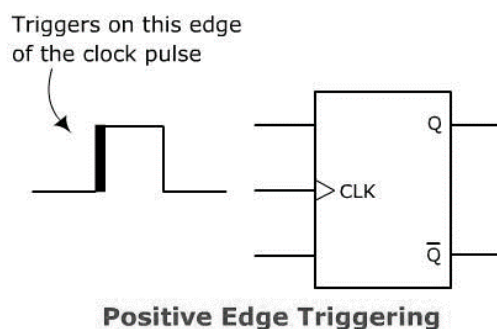
2. Low Level Triggering

When a flip flop is required to respond at its LOW state, a LOW level triggering method is used. It is mainly identified from the clock input lead along with a low state indicator bubble. Take a look at the symbolic representation shown below.



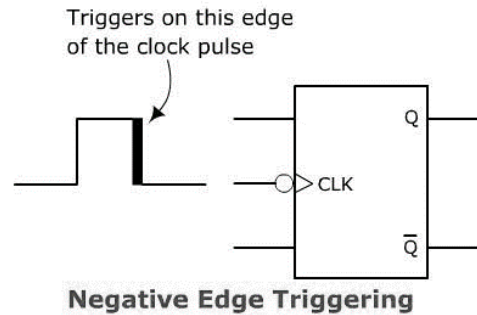
3. Positive Edge Triggering

When a flip flop is required to respond at a LOW to HIGH transition state, POSITIVE edge triggering method is used. It is mainly identified from the clock input lead along with a triangle. Take a look at the symbolic representation shown below.



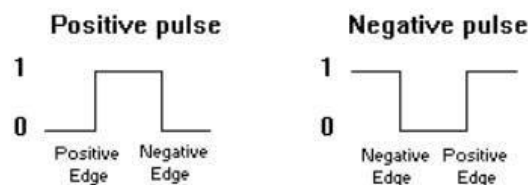
4. Negative Edge Triggering

When a flip flop is required to respond during the HIGH to LOW transition state, a NEGATIVE edge triggering method is used. It is mainly identified from the clock input lead along with a low-state indicator and a triangle. Take a look at the symbolic representation shown below.



Clock Pulse Transition

The movement of a trigger pulse is always from a 0 to 1 and then 1 to 0 of a signal. Thus it takes two transitions in a single signal. When it moves from 0 to 1 it is called a positive transition and when it moves from 1 to 0 it is called a negative transition. To understand more take a look at the images below.



Definition of clock pulse transition

Clock Pulse Transition

The clocked flip-flops already introduced are triggered during the 0 to 1 transition of the pulse, and the state transition starts as soon as the pulse reaches the HIGH level. If the other inputs change while the clock is still 1, a new output state may occur. If the flip-flop is made to then the multiple-transition problem can be eliminated. The multi-transition problem can be stopped is the flip flop is made to respond to the positive or negative edge transition only, other than responding to the entire pulse duration.

Applications of Flip-Flops

These are the various types of flip-flops being used in digital electronic circuits and the applications of Flip-flops are as specified below.

- Counters
- Frequency Dividers
- Shift Registers
- Storage Registers

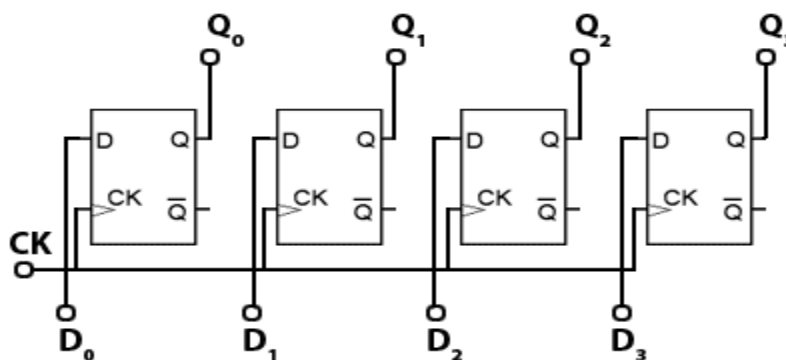
REGISTERS

Flip-flop is a 1-bit memory cell which can be used for storing the digital data. To increase the storage capacity in terms of number of bits, we have to use a group of flip-flops. Such a group of flip-flops is known as a Register. The n-bit register will consist of n number of flip-flop and it is capable of storing an n-bit word. The binary data in a register can be moved within the register from one flip-flop to another. The registers that allow such data transfers are called as shift registers.

REGISTER CIRCUIT

An electronic register is a form of memory that uses a series of flip-flops to store the individual bits of a binary word, such as a byte (8 bits) of data. A flip flop is used to store single bit digital data. For storing a large number of bits, the storage capacity is increased by grouping more than one flip flops. If we want to store an n-bit word, we have to use an n-bit register containing n number of flip flops.

The length of the stored binary word depends on the number of flip-flops that make up the register. A simple 4-bit register is illustrated in figure below and consists of four D Type flip-flops, sharing a common clock input, providing synchronous operation ensuring all bits are stored at exactly the same time.



4-bit register

The binary word to be stored is applied to the four D inputs and is remembered by the flip-flops at the rising edge of the next clock (CK) pulse. The stored data can then be read from the Q outputs at any time, as long as power is maintained, or until a change of data on the D inputs is stored by a further clock pulse, which overwrites the previous data. When $Q_3Q_2Q_1Q_0$ is 0000, then $D_3D_2D_1D_0$ is 0000, when $Q_3Q_2Q_1Q_0$ is 0001, then $D_3D_2D_1D_0$ is 0001.

SHIFT REGISTER

The information stored within these registers can be transferred with the help of shift registers. Shift Register is a group of flip flops used to store multiple bits of data. The bits stored in such registers can be made to move within the registers and in/out of the registers by applying clock pulses. An n-bit shift register can be formed by connecting n flip-flops where each flip flop stores a single bit of data.

SHIFT LEFT REGISTER

The registers which shift the bits to left are called “Shift left registers”. The shift left register is constructed using four D flipflop. It received a common clock pulse CK which make the circuit synchronous. The leftmost flipflop is used to input a data D. The output bits are collected from $Q_D Q_C Q_B Q_A$. Each time the circuit received the clock pulse, each bit is shifted one position to the left. The logic diagram and truth table are shown below:

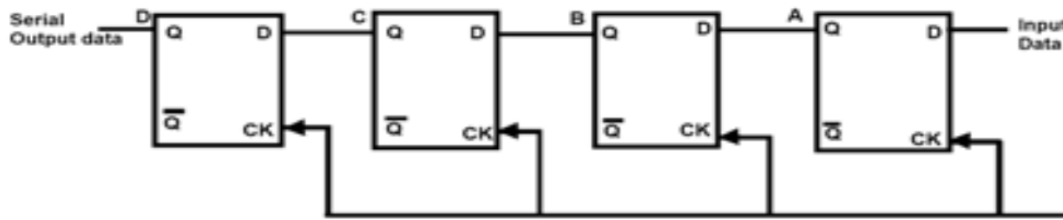


Fig. 4 bit shift left register

Truth table

CLK	Q_D	Q_C	Q_B	Q_A	D
0	0	0	0	0	
1	0	0	0	1	1
2	0	0	1	0	0
3	0	1	0	0	0
4	1	0	0	0	0

← Direction of data travel

SHIFT RIGHT REGISTER

The registers which shift the bits to right are called “Shift right registers”. The shift right register is constructed using four D flipflop. It received a common clock pulse CLK which make the circuit synchronous. The rightmost flipflop is used to input a data D. The output bits are collected from $Q_D Q_C Q_B Q_A$. Each time the circuit received the clock pulse, each bit is shifted one position to the right. The logic diagram and truth table are shown below:

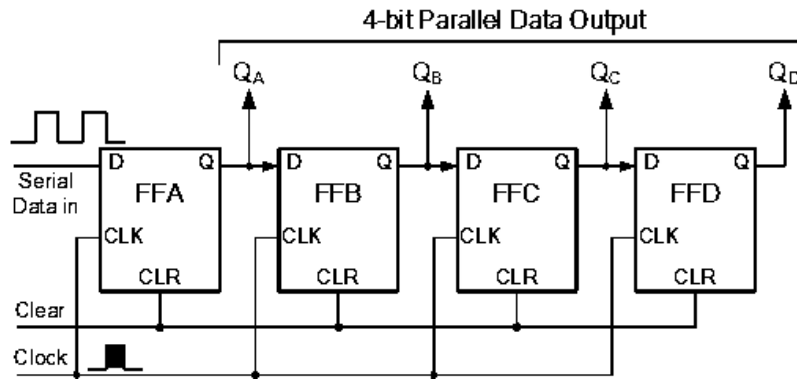


Fig. 4 bit shift right register

The operation is as follows. Lets assume that all the flip-flops (FFA to FFD) have just been RESET (CLEAR input) and that all the outputs QA to QD are at logic level “0” ie, no parallel data output. If a logic “1” is connected to the DATA input pin of FFA then on the first clock pulse the output of FFA and therefore the resulting QA will be set HIGH to logic “1” with all the other outputs still remaining LOW at logic “0”.

The second clock pulse will change the output of FFA to logic “0” and the output of FFB and QB HIGH to logic “1” as its input D has the logic “1” level on it from QA. The logic “1” has now moved or been “shifted” one place along the register to the right as it is now at QA. The operation goes on as before.

Then the data has been converted from a serial data input signal to a parallel data output. The truth table and following waveforms show the propagation of the logic “1” through the register from left to right as follows.

Truth Table:

CLK	D	Q _A	Q _B	Q _C	Q _D
0		0	0	0	0
1	1 →	1	0	0	0
2	0 →	0	1	0	0
3	0 →	0	0	1	0
4	0 →	0	0	0	1

—————→ Direction of data travel

ASYNCHRONOUS COUNTERS

A counter is a device which can count any particular event on the basis of how many times the particular event(s) is occurred. In a digital logic system or computers, this counter can count and store the number of times any particular event or process have occurred, depending on a clock signal. Most common type of counter is sequential digital logic circuit with a single clock input and multiple outputs. The outputs represent binary or binary coded decimal numbers. Each clock pulse either increase the number or decrease the number.

Asynchronous stands for the absence of synchronization. Something that is not existing or occurring at the same time. In computing or telecommunication stream, Asynchronous stands for controlling the operation timing by sending a pulse only when the previous operation is completed rather than sending it in regular intervals.

FOUR BIT RIPPLE COUNTER

Ripple counter is an Asynchronous counter. It got its name because the clock pulse ripples through the circuit. In 4-bit ripple counter, n value is 4 so, 4 JK flip flops are used and the counter can count up to 16 pulses. Below the circuit diagram and timing diagram are given along with the truth table.

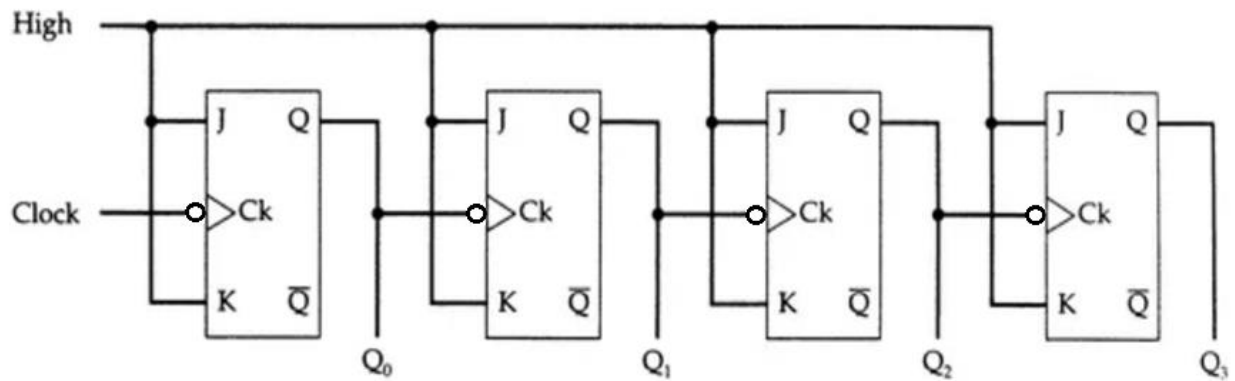


Fig. 4 bit ripple counter

All the JK flip-flops are configured to toggle their state on a downward transition of their clock input, and the output of each flip-flop is fed into the next flip-flop's clock. So, when each bit changes from 1 to 0, it "carries the one" to the next higher bit. For a 4-bit counter, the range of the count is 0000 to 1111 (2^4-1). A counter may count up or count down or count up and down depending on the input control. The count sequence usually repeats itself.

Truth Table

Count Sequence for a Binary Ripple Counter

Count Sequence					
Q_3	Q_2	Q_1	Q_0	Conditions for Complementing Flip-Flops	
0	0	0	0	Complement A_1	
0	0	0	1	Complement A_1	A_1 will go from 1 to 0 and complement A_2
0	0	1	0	Complement A_1	
0	0	1	1	Complement A_1	A_1 will go from 1 to 0 and complement A_2 ; A_2 will go from 1 to 0 and complement A_3
0	1	0	0	Complement A_1	
0	1	0	1	Complement A_1	A_1 will go from 1 to 0 and complement A_2
0	1	1	0	Complement A_1	
0	1	1	1	Complement A_1	A_1 will go from 1 to 0 and complement A_2 ; A_2 will go from 1 to 0 and complement A_3 ; A_3 will go from 1 to 0 and complement A_4
1	0	0	0		and so on . . .

BCD COUNTER

As there is a maximum output number for Asynchronous counters like MOD-16 with a resolution of 4-bit, there are also possibilities to use a basic Asynchronous counter in a configuration that the counting state will be less than their maximum output number. Modulo or MOD counters are one of those types of counters. The configuration made in such a way that the counter will reset itself to zero at a pre-configured value and has truncated sequences. So, if a counter with the specific number of resolutions (n-bit Resolution) count up to is called as full sequence counter and on the other hand, if it is count less than the maximum number, is called as a truncated counter.

To get the advantage of the asynchronous inputs in the flipflop, Asynchronous Truncated counter can be used with combinational logic.

Modulo 16 asynchronous counter can be modified using additional logic gates and can be used in a way that the output will give a decade (divided by 10) counter output, which is useful in counting standard decimal numbers or in arithmetic circuits. This type of counters called as Decade Counters.

Decade Counters requires resetting to zero when the output reaches a decimal value of 10.

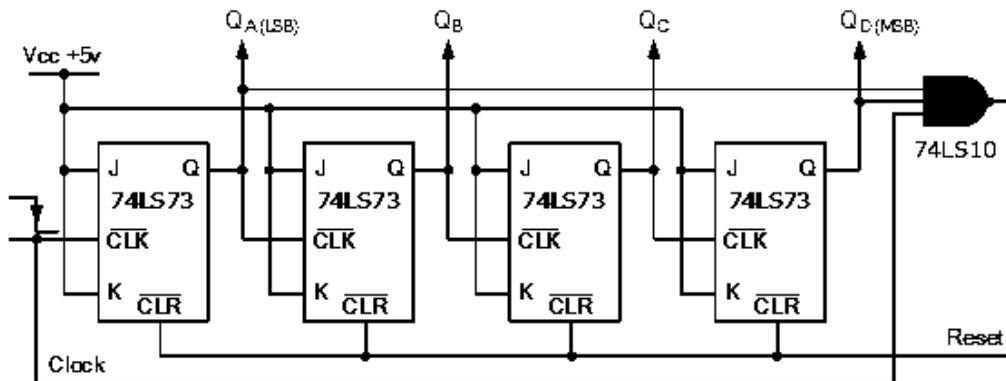


Fig. 4 bit BCD counter

This type of asynchronous counter counts upwards on each trailing edge of the input clock signal starting from 0000 until it reaches an output 1001 (decimal 9). Both outputs QA and QD are now equal to logic “1”. On the application of the next clock pulse, the output from the 74LS10 NAND gate changes state from logic “1” to a logic “0” level.

As the output of the NAND gate is connected to the CLEAR (CLR) inputs of all the 74LS73 J-K Flip-flops, this signal causes all of the Q outputs to be reset back to binary 0000 on the count of 10. As outputs QA and QD are now both equal to logic “0” as the flip-flop’s have just been

reset, the output of the NAND gate returns back to a logic level “1” and the counter restarts again from 0000. We now have a decade or Modulo-10 up-counter.

Decade Counter Truth Table

Clock Count	Output bit Pattern				Decimal Value
	QD	QC	QB	QA	
1	0	0	0	0	0
2	0	0	0	1	1
3	0	0	1	0	2
4	0	0	1	1	3
5	0	1	0	0	4
6	0	1	0	1	5
7	0	1	1	0	6
8	0	1	1	1	7
9	1	0	0	0	8
10	1	0	0	1	9
11	Counter Resets its Outputs back to Zero				

SYNCHRONOUS COUNTER

Synchronous Counter, the external clock signal is connected to the clock input of EVERY individual flip-flop within the counter so that all of the flip-flops are clocked together simultaneously (in parallel) at the same time giving a fixed time relationship. In other words, changes in the output occur in “synchronisation” with the clock signal.

The result of this synchronisation is that all the individual output bits changing state at exactly the same time in response to the common clock signal with no ripple effect and therefore, no propagation delay.

BINARY COUNTER

In digital electronics, a binary counter is a type of sequential logic circuit which is able to count in binary numbers. A binary counter can counter from 0 to $2(n-1)$, where n is the total number of bits in the counter.

Basically, a binary counter is a type of digital circuit which counts the number of clock pulses that occur over a time period.

The binary counters are built up of flip flops, where a flip flop is a most elementary memory element that can store 1-bit of information. In a binary counter, each flip flop represents one bit of the **binary number**. **The counter increases its count by one whenever a clock pulse occurs.**

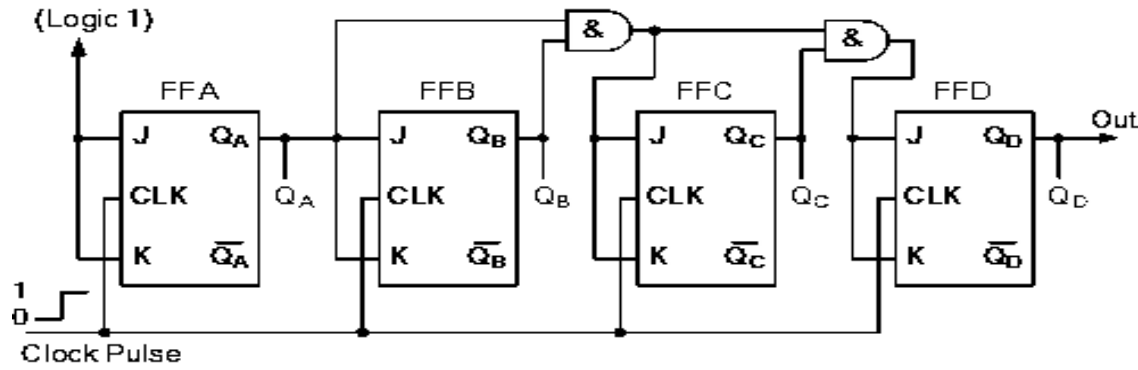


Fig. 4 bit binary counter

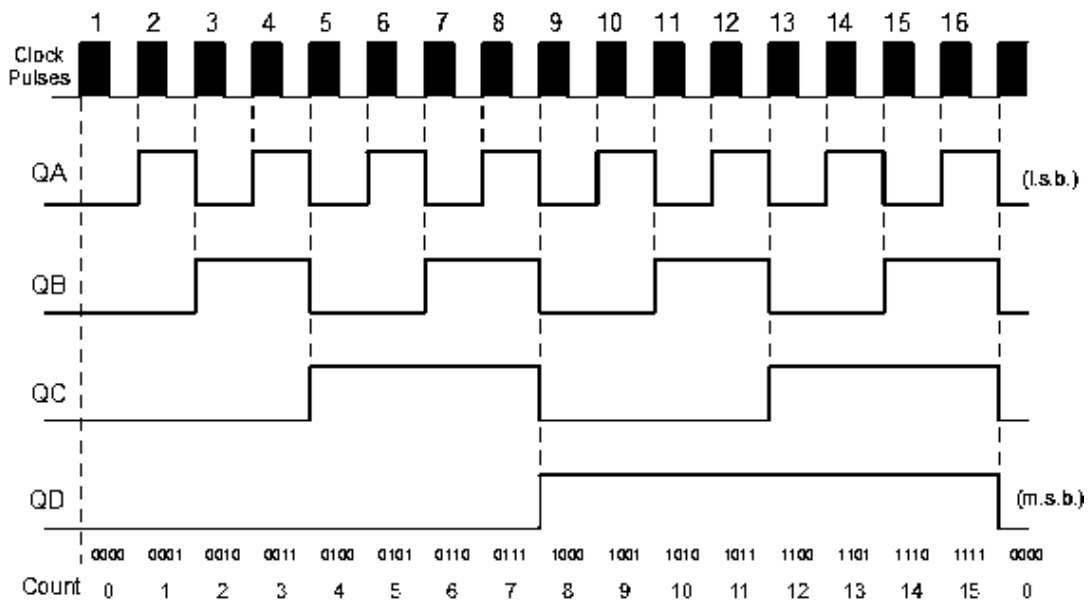
It can be seen above, that the external clock pulses (pulses to be counted) are fed directly to each of the J-K flip-flops in the counter chain and that both the J and K inputs are all tied together in toggle mode, but only in the first flip-flop, flip-flop FFA (LSB) are they connected HIGH, logic “1” allowing the flip-flop to toggle on every clock pulse. Then the synchronous counter follows a predetermined sequence of states in response to the common clock signal, advancing one state for each pulse. The J and K inputs of flip-flop FFB are connected directly to the output QA of flip-flop FFA, but the J and K inputs of flip-flops FFC and FFD are driven from separate AND gates which are also supplied with signals from the input and output of the previous stage. These additional AND gates generate the required logic for the JK inputs of the next stage.

If we enable each JK flip-flop to toggle based on whether or not all preceding flip-flop outputs (Q) are “HIGH” we can obtain the same counting sequence as with the asynchronous circuit but without the ripple effect, since each flip-flop in this circuit will be clocked at exactly the same time. Then as there is no inherent propagation delay in synchronous counters, because all the counter stages are triggered in parallel at the same time, the maximum operating frequency of this type of frequency counter is much higher than that for a similar asynchronous counter circuit.

Truth table of Binary Counter:

<i>State</i>	Q_D	Q_C	Q_B	Q_A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1
0	0	0	0	0

4-bit Synchronous Counter Waveform Timing Diagram



Because this 4-bit synchronous counter counts sequentially on every clock pulse the resulting outputs count upwards from 0 (0000) to 15 (1111). Therefore, this type of counter is also known as a 4-bit Synchronous Up Counter.

However, we can easily construct a 4-bit Synchronous Down Counter by connecting the AND gates to the Q output of the flip-flops as shown to produce a waveform timing diagram the reverse of the above. Here the counter starts with all of its outputs HIGH (1111) and it counts down on the application of each clock pulse to zero, (0000) before repeating again.

BCD COUNTER

A BCD counter is a type of sequential logic circuit which is able to count binary numbers from 0 upto 9. After it reaches 9, the circuit is reset back to 0. It is also called as decade counter as it can count upto 10 binary number at a time (0 to 9). A 4-bit decade synchronous counter can also be built using synchronous binary counters to produce a count sequence from 0 to 9. A standard binary counter can be converted to a decade (decimal 10) counter with the aid of some additional logic to implement the desired state sequence. After reaching the count of “1001”, the counter recycles back to “0000”. We now have a decade or **Modulo-10** counter.

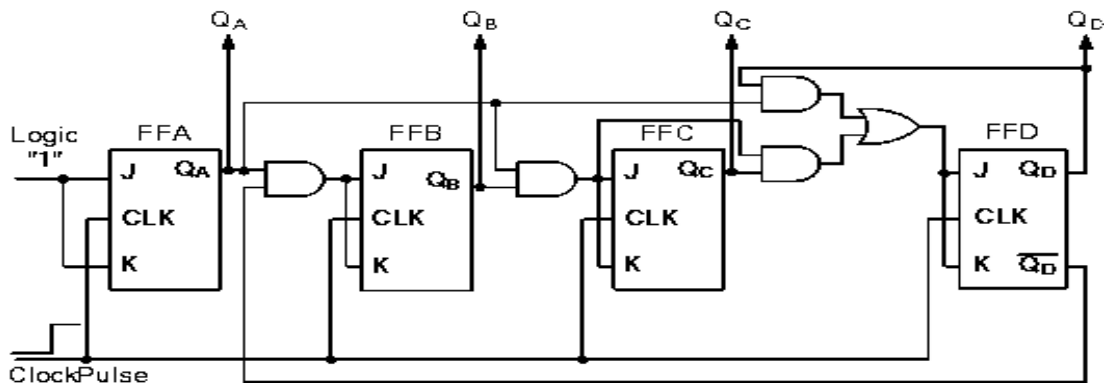


Fig. 4 bit BCD counter

The additional AND gates detect when the counting sequence reaches “1001”, (Binary 10) and causes flip-flop FF3 to toggle on the next clock pulse. Flip-flop FF0 toggles on every clock pulse. Thus, the count is reset and starts over again at “0000” producing a synchronous decade counter.

Truth table of BCD counter

Input Pulses	D	C	B	A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
0	0	0	0	0 (resets)

DIFFERENCE BETWEEN SYNCHRONOUS COUNTER AND ASYNCHRONOUS COUNTER

S.NO	SYNCHRONOUS COUNTER	ASYNCHRONOUS COUNTER
1.	All flip flops are triggered with same clock simultaneously.	Different flip flops are triggered with different clock, not simultaneously.
2.	Synchronous Counter is faster than asynchronous counter in operation.	Asynchronous Counter is slower than synchronous counter in operation.
3.	Synchronous Counter does not produce any decoding errors.	Asynchronous Counter produces decoding error.
4.	Synchronous Counter is also called Parallel Counter.	Asynchronous Counter is also called Serial Counter.
5.	Synchronous Counter designing as well implementation are complex due to increasing the number of states.	Asynchronous Counter designing as well as implementation is very easy.
6.	Synchronous Counter will operate in any desired count sequence.	Asynchronous Counter will operate only in fixed count sequence (UP/DOWN).
7.	Example: Ring counter, Johnson counter.	Example: Ripple UP counter, Ripple DOWN counter.