Counters

Counters are sequential circuits which "count" through a specific state sequence. They can count up, count down, or count through other fixed sequences.

Counters can be implemented using a number of flip flops that are connected to give required function. Each flip-flop is used to represent one bit. The number of flip-flop in the counter depends on the type of the counter. Counter that has n flip-flop can be designed to have number of states in their sequence that is less than 2^n . An n–bits binary counter consists of **n** flip – flops can count in binary from 0 to 2^n –1. The number of states that used in the counter is called modulus (mod). For example a modulus 6 would count from 0 (000)2 to 5 (101)2.

Digital counter are classified as **synchronous** or **asynchronous**, dependent on how they are clocked. The synchronous counters are a series of Flip-Flops, each clocked at the same time, causing the outputs of the states (Flip-Flops) to change together. The asynchronous counters are a series of Flip-Flops, each clocked by the previous state, one after the other. Within these two types, counters are categorizing by the type of sequence, the number of state, or the number of flip-flops in the counter. Counters play an important role in computers. They are used to hold values that are internal to the central processing unit.

1-Asynchronous (Ripple) Counters

Asynchronous counters called ripple counters, the first flip-flop is clocked by the external clock pulse and then each successive flip-flop is clocked by the output of the preceding flip flop. The term asynchronous refers to events that do not have a fixed time relationship with each other . An asynchronous counter is one in which the flip-flops within the counter do not change states at exactly the same time because they do not have a common clock pulse.

2- Synchronous Counters

Clock is directly connected to the flip-flop clock inputs. Logic is used to implement the desired state sequencing. The clock signal (CLK) is applied to all FF, which means that all FF shares the same clock signal.

Asynchronous (Ripple) binary counter

A counter that follows the binary sequence is called a binary counter. A binary counter with a reverse count is called a binary down counter There are four basic configuration used to design ripple binary counters. These are classified according to the triggering of the counter flip-flops (positive or negative edge) and the direction of counting (up and down) as shown below:

2-Bit Asynchronous Binary Counter

Fig.1 shows a 2-bit counter connected for asynchronous operation. Notice that the clock (CLK) is applied to the clock input (C) of only the first flop-flop, FF0, which is always the least significant bit (LSB). The second flip-flop, FF1, is triggered by the $\overline{Q}_0$ out-put of FF0. FF0 changes state at the positive-going edge of each clock pulse. But FF1 changes only when triggered by a positive-going transition of the $\overline{Q}_0$ output of FF0. Because of the inherent propagation delay tie through a flip-flop, a transition of the input clock pulse (CLK) and a transition of the $\overline{Q}_0$ output of FF0 can never occur at exactly the same time. Therefore, the two flip-flops are never simultaneously triggered, so the counter operation is asynchronous.

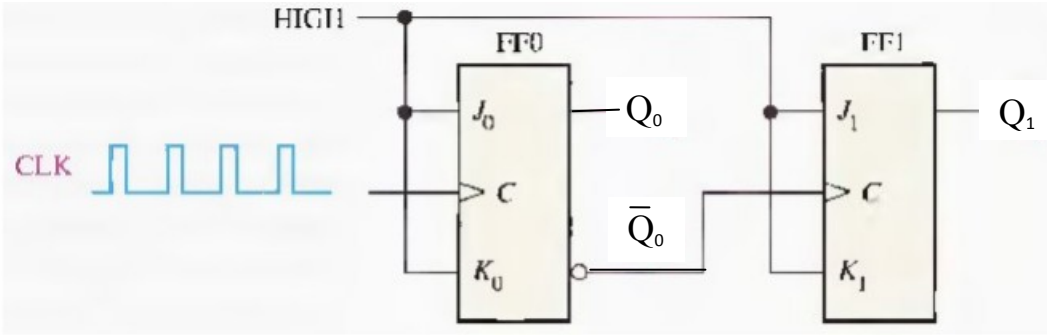
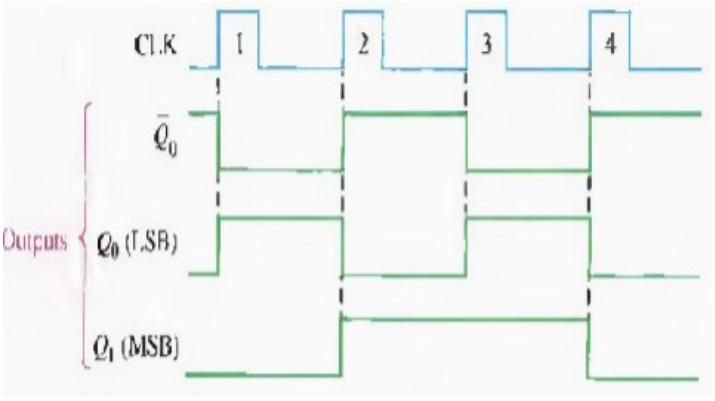


Fig.1



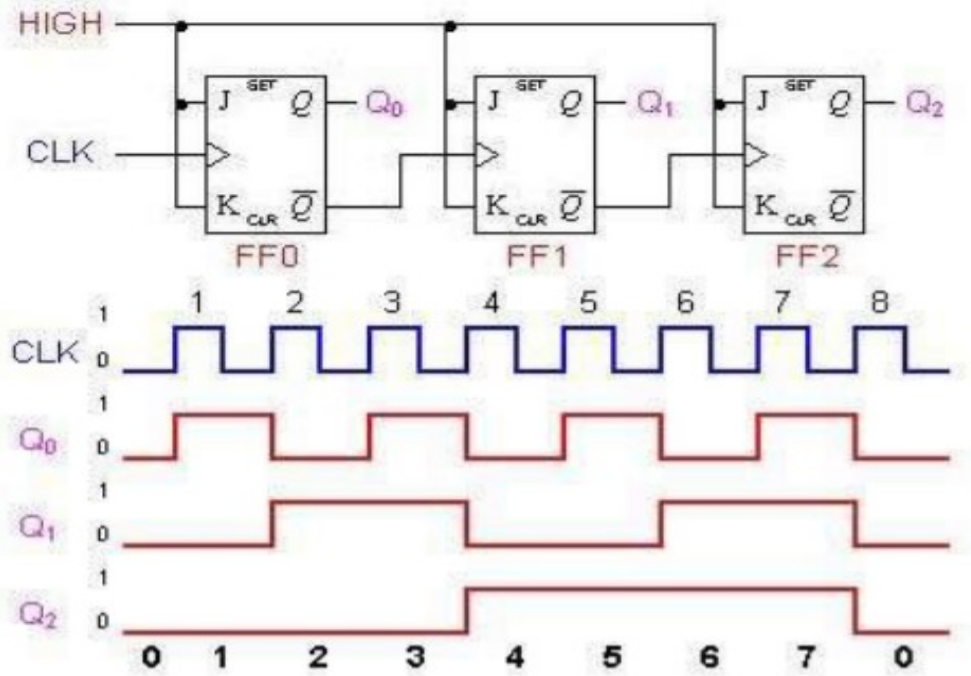
CLOCK PULSE	Q_1	Q_0
Initially	0	0
1	0	1
2	1	0
3	1	1
4 (recycles)	0	0

Timing diagram for the counter of Fig1 Binary state sequence

3 bit Binary Asynchronous MOD-8 up Counter

• A 3-bit asynchronous binary counter and its timing diagram is shown in the adjoining Fig.

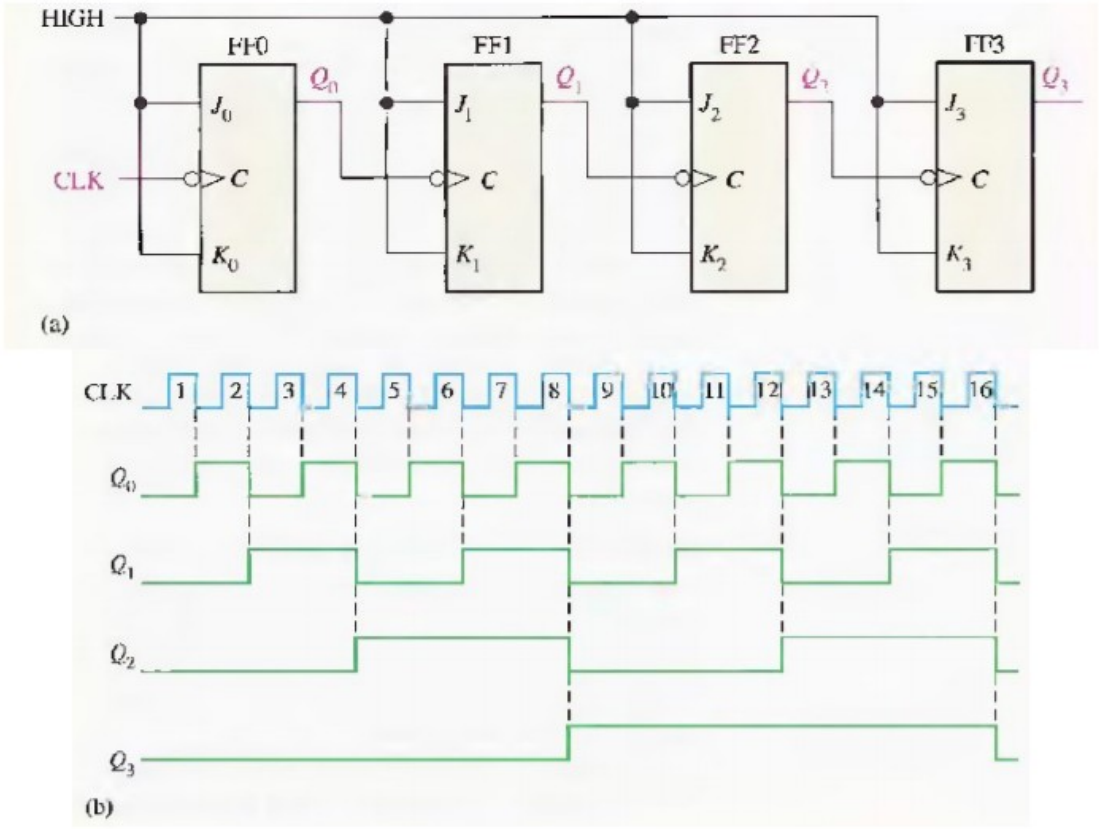
• It works exactly the same way as a 2 bit asynchronous binary counter mentioned above, except it has eight states due to the third flip-flop.



Clock Pulse	Q_2	Q_1	Q_0
Initially	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
8 (recycles)	0	0	0

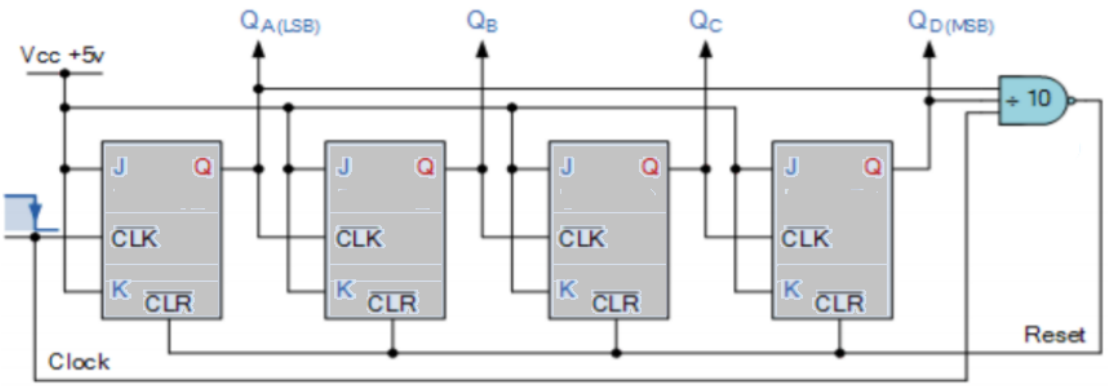
4 bit Binary Asynchronous Decade/MOD-10 up Counter

A 4-bit asynchronous binary counter is shown below:

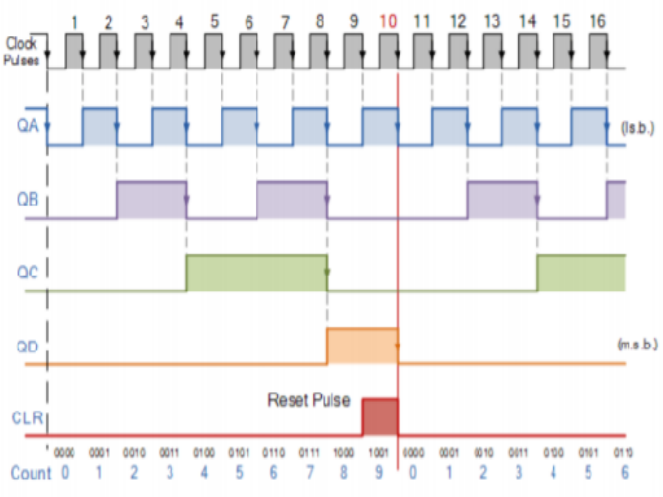


Also called BCD counter. For a counter to count from 0000 to 1001, four flip flops are required. But we need to mechanism to restrict the count to 1001 and thereafter reset the counter to 0000 again otherwise our counter (with 4 flip flops) will continue to 1111 making it MOD-16 counter instead of MOD-10.

One way to make the counter recycle after the count of 9 (1001) is by using NAND gate that connect its output to the clear (CLR) inputs of these flip-flops, as shown below.



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 Q_A and Q_D are now equal to logic "1". On the application of the next clock pulse, the output from the 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 J-K Flipflops, this signal causes all of the Q outputs to be reset back to binary 0000 on the count of 10.
As outputs Q_A and Q_D 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.



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				

• A common modulus for counters with truncated sequences is ten and is called a decade counter.
• By using the same methodology, any other MOD-N truncated up or down counter can be designed.