Chapter Eight

8.1 Introduction:

A counter is an electronic device which is able to generate an orderly sequence of binary numbers. The addresses which command digital circuits as decoder, multiplexer, demultiplexer etc. are most of the time generated by counters. There two main basic types of counters: asynchronous counter and synchronous counter. Counters are designed using flip-flops, usually J-K flip-flops.

Asynchronous counters functions in such a way that the clock signal does not affect all the flip-flops at the same instant; in fact, the clock signal triggers the first flip-flop which in his turn trigger the second flip-flop using its output signal. The second flip-flop in his turn will trigger the third one, and so on, till the last flip-flop. It is just like the triggering signal was being propagated from the first flip-flop to the last. For this reason, asynchronous counters are also called propagation counters.

The flip-flops of a synchronous counter are all triggered by the same clock signal at the same instant. In fact, all the flip-flops function in synchronism with the clock signal. The outputs of all the flip-flops change their status at the same instant.

8.2 Asynchronous counters:

Let us consider the following binary count sequence. The numbers are coded in four bits. Q_0 is the LSB and Q_3 is the MSB. It can be noticed that each bit in this four-bit sequence toggles when the bit before it (the bit having a lesser significance or place weight), toggles from 1 to 0. Small arrows are used to indicate those places in the above count sequence. So, to design an asynchronous counter which is able to generate the above sequence of numbers, we need to determine how to connect the clock inputs of each of the four J-K flip-flops (each flip-flop generating one bit) in such a way that each bit will toggle only when the bit just before it is transitioning from 1 to 0. The easier way of doing that is to use flip-flops with negative edge triggering. The clock input of each flip-flop will simply be connected to the non-complemented output Q of the flip-flop situated just before it (Flip-flop having one lesser place weight). In this condition, each flip-flop will toggle any time the output of the flip-flop one place lesser weighted than it is transitioning from 1 to 0 (negative edge). Such a connection is done as shown in the following figure.

Q_3	Q_2	Q_1	Q_0	Decimal
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2

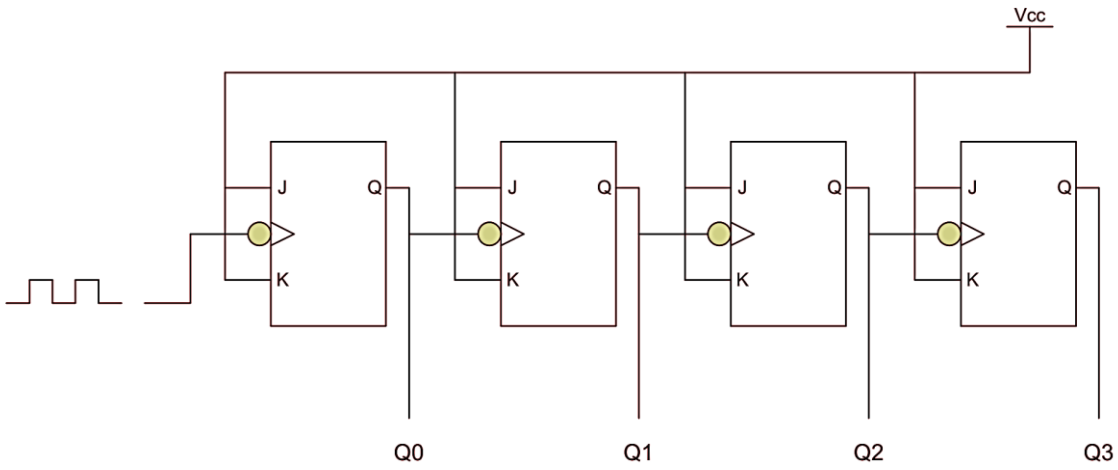
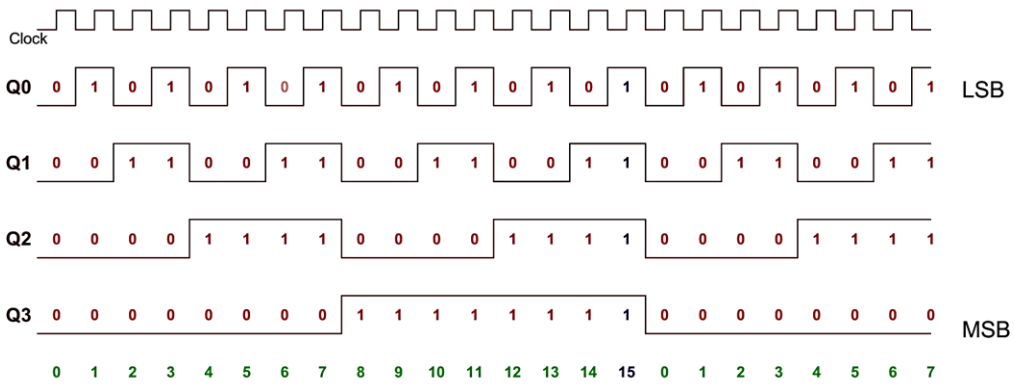


Figure 8.1: Four bits asynchronous counter.

From the functioning of this asynchronous counter, the following waveforms can be generated.



Each of the signals generated by the outputs of the flip-flops is used as the clock signal for the following flip-flop. So, the falling edge of each signal causes the output of the following flip-flop to toggle. It can also be clearly observed that those waveforms sketch the count sequence from 0_{10} (0000) to 15_{10} (1111). When the maximum number of the sequence is reached, the counter is automatically reset (brought back to 0).

8.3 Counter modulus:

The modulus of a counter can be defined as the number of different states that it occupies within a count sequence. The four bits counter above has 16 different states (from 0000 to 1111), therefore its modulus is 16. In general, N being the number of flip-flops used to design a counter, the modulus of the counter is given by the following formula:

$$\text{Modulus} = 2^N$$

Therefore, to add the modulus of a counter, we just need to add the number of counters used to design it.

Exercise 8.1:

Conceive an asynchronous counter modulus 8, modulus 32.

8.4 Counters having modulus < 2^N:

It is possible to conceive a counter which is such that its modulo is not a power of 2 as in the cases above. For this issue, asynchronous inputs presented in the previous chapter will be used. Let us realise an asynchronous counter modulo 5 (the count sequence goes from 000 to 100).

$2^2 = 4 < 5$;

$2^3 = 8 > 5$.

Therefore, we cannot use 2 flip-flops to design the counter. The suitable number of flip-flops is 3. The three flip-flops should be connected in such a way that; the count sequence will end at 4_{10} (100) instead of ending at 7_{10} (111) as expected. It means that the counter should be obliged to reset when the binary number 101 (5_{10}) occurs.

1	0	1
Q_2	Q_1	Q_0

To achieve this task, one method consists in gathering all the bits of the number which are in high logic state to the input of a NAND gate. The output of that NAND gate (which is low only when all its inputs are high) will be used to reset the counter when the number 101 occurs. The NAND gate is used in the case were the asynchronous inputs CLR of the flip-flops function in low logic level. If they were functioning at high logic level, AND gate would have been used in such a way that, when all its inputs are high, it outputs a high logic level that reset the counter. The following figure shows how the connection can be done.

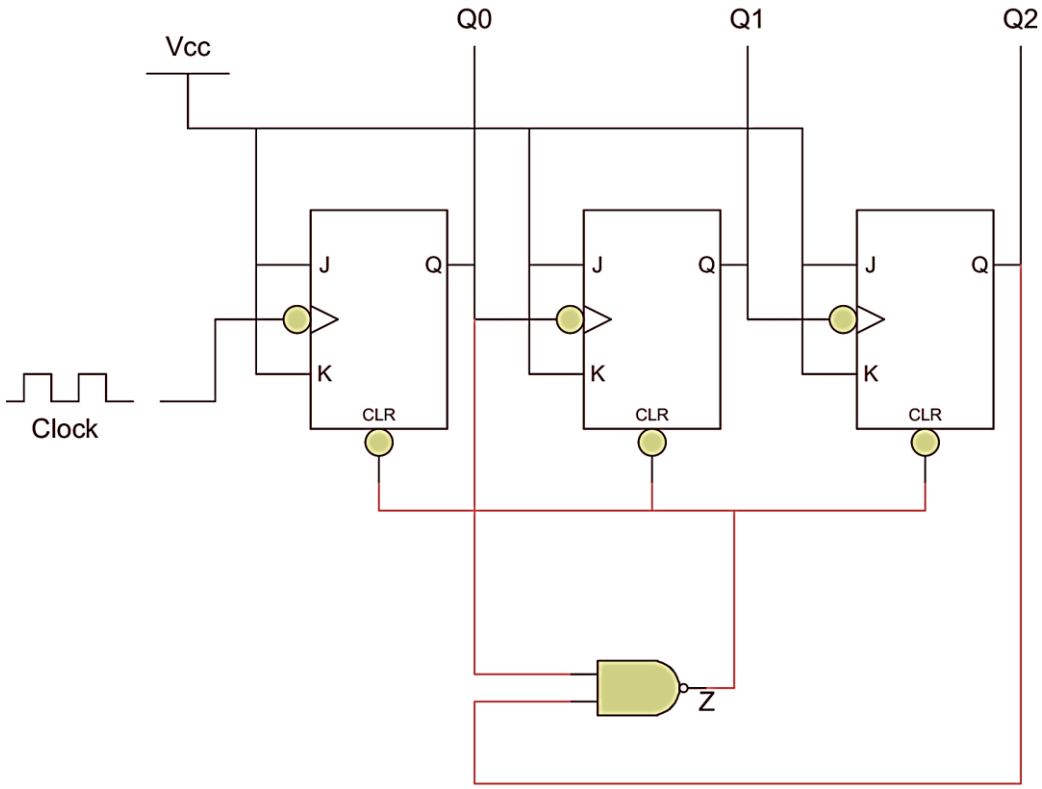
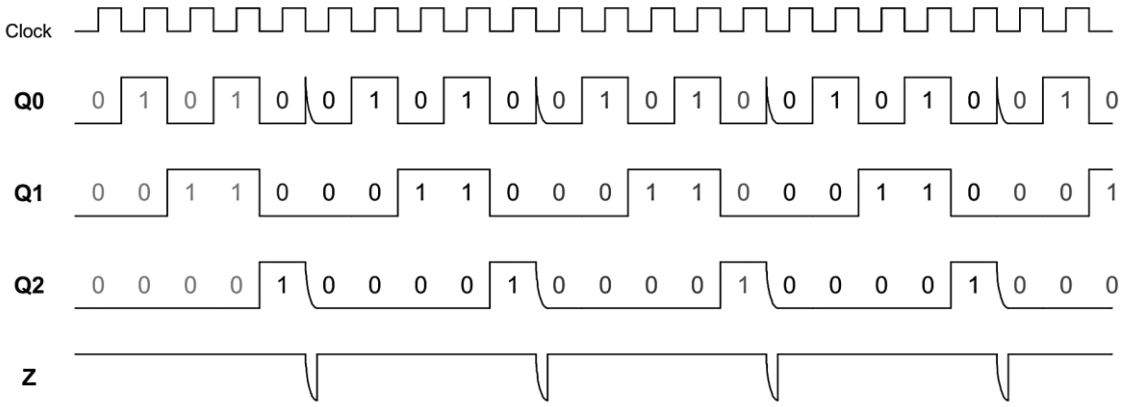


Figure 8.2: Asynchronous counter modulo 5.

From the functioning of the above asynchronous counter modulus 5, the

following waveforms can be generated.



The output Z of the NAND gate is connected to the CLR input of the three J-K flip-flops. When Z = 1, the counter is not affected. However, when Z = 0 (Q_0 = 1 and Q_2 = 1), the counter is reset. Therefore, a new count sequence can start. The count sequence is as follows:

Q2	Q1	Q0
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
0	0	0
0	0	1

The number 101 does not occur, for the counter is immediately reset after 100_2.

8.5 Advantages and disadvantages of asynchronous counters:

The main advantage of the synchronous counter is its simplicity as far as the conception is concern. However, this type of counter has a great problem: the ripple effect. This is the time delay due to the propagation of the signal throughout the flip-flops which the counter is made up of. The ripple effect can cause error in computer circuits. To avoid the ripple effect, another type of counter has been designed: The synchronous counter.

8.6 Synchronous counter:

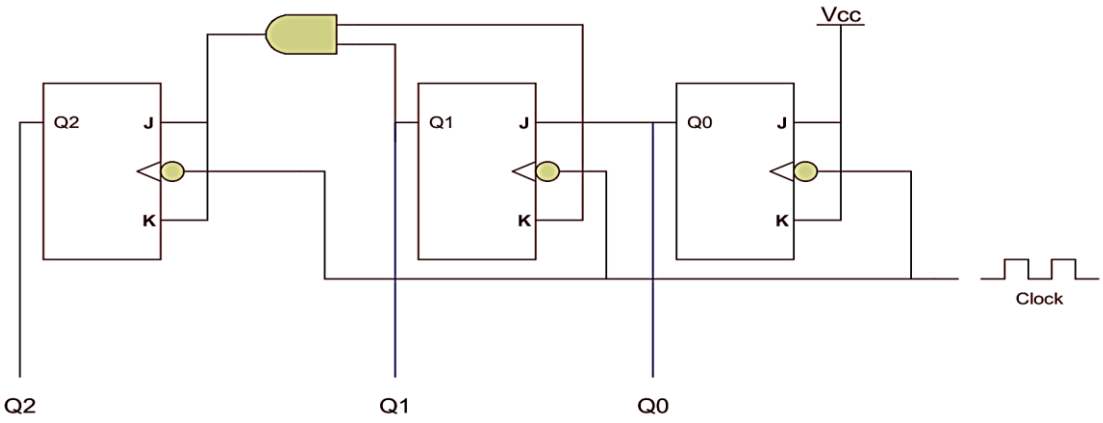
A synchronous counter, in contrary to an asynchronous counter functions in such a way that all its flip-flops toggle at the same instant, because they are triggered by the same clock signal.

8.6.1 Principle of an asynchronous counter:

Let us consider again a four-bit count sequence:

Q_3	Q_2	Q_1	Q_0	Decimal
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2

While examining this count sequence, it can be noticed that, for a bit belonging to a given rank to toggle (Q0, Q1, Q2, Q3), the bits of all the previous rank lesser weighted than it should be at high logic level. This property can be exploited to design a synchronous counter, as shown by the following diagram.



The flip-flop Q0 toggles on every rising edge of the clock pulse. The flip-flop Q1 toggles only when Q0 is high. The flip-flop Q2 toggles only when Q0 and Q1 are high. We can also notice that all the flip-flops are triggered by the same clock signal.

Each of the higher order flip-flops are made ready to toggle (both J and K inputs are high) if the Q input of all the previous flip-flops are high. Otherwise, the J and K inputs for that flip-flop will be both low, placing it into the latch mode where it will maintain its present output state. Since the first flip-flop (LSB) needs to toggle at every clock pulse, its J and K inputs are connected to Vcc. The following equations can be deduced.

$J_0 = K_0 = 1$

$J_1 = K_1 = Q_0$

$J_2 = K_2 = Q_0Q_1$

$J_3 = K_3 = Q_0Q_1Q_2$

Generally…

$J_n = K_n = Q_0Q_1Q_3\ldots Q_{n-1}$

Remark 8.1:

For the conception of synchronous counters having modulo < 2^N, N being an integer, asynchronous inputs should be used in the same manner as they were used for asynchronous counters.

8.6.2 Shift register

A register can be defined as a set of flip-flops connected together and intended to store a memory word. Each flip-flop of the register stores one bit. A shift register is a set of register connected together and intended to be used for the transfer of data. The transfer of data can be serial or parallel. In serial transfer, data are sent one bit after another over long distances. In parallel transfer, a whole

memory word is transmitted at once, over a relatively short distance.

Serial data transmission over a distance of meters to kilometres, uses shift registers to convert parallel data to serial form. Serial data transmission replaces many slow parallel wires with a single high speed transmission line.

Basic shift registers are classified by structure according to the following types:

- Serial in / Serial out;
- Parallel in / Serial out;
- Serial in / Parallel out;
- Parallel in / Parallel out;
- Ring counters.

a. Serial in/ serial out shift register:

Data are sent in the register one bit after another and are sent out from it one bit after another. The following diagram illustrates this type of transfer.

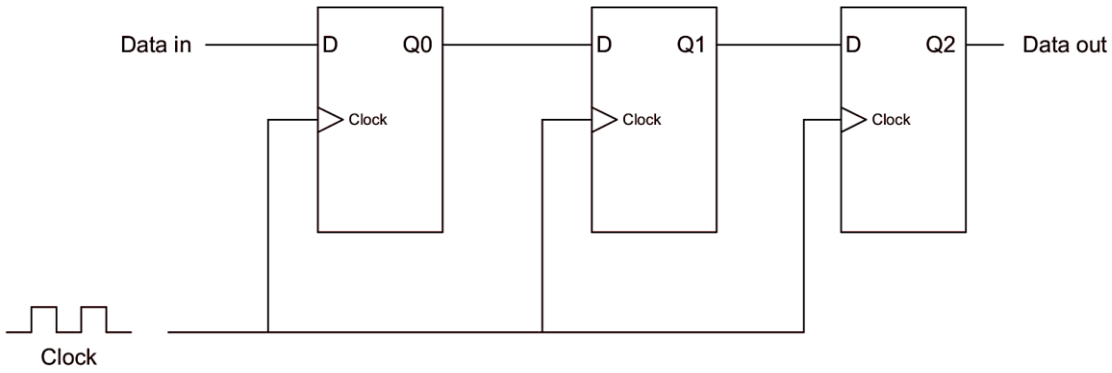


Figure 8.3: Serial in / serial out shift register using D flip-flops.

Once the pulse of the clock signal occurs, the bit available at the input of the first flip-flop is shifted to that of the next flip-flop. The same process is repeated till the last flip-flop. J-K flip-flops can also be used to achieve the same task. The diagram is done as follows.

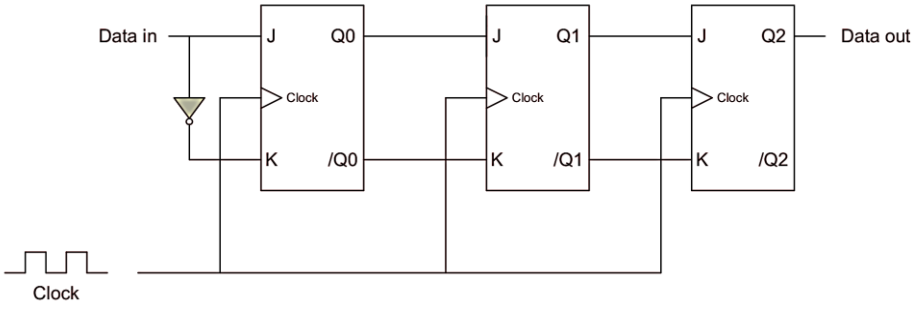


Figure 8.4: Serial in/ Serial out shift register using J-K flip-flops.

b. Ring counter:

A ring counter is obtained when the output of a shift register is fed back to the input. One of the most known ring counter is the Johnson counter. The principle of the Johnson counter is presented by the following diagram.

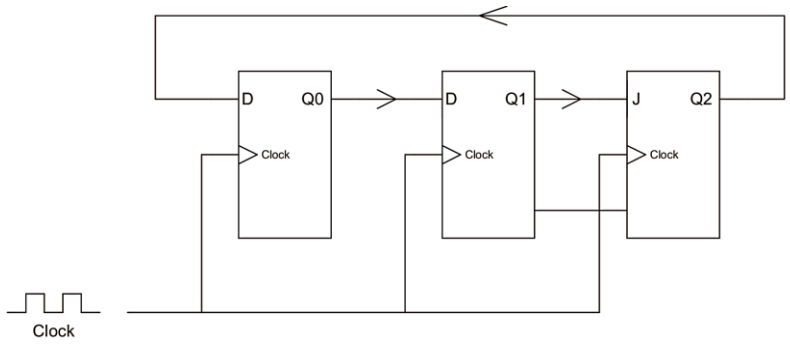


Figure 8.5: Johnson counter.

During the functioning of the Johnson counter, a single bit is being transferred from one flip-flop to another in a ring made up of the flip-flops connected together. This functioning can be described using the following table (it is assumed that initially Q0 = 1, Q1 = Q2 = Q3 = 0).

Q0	Q1	Q2	Q3	Clock
1	0	0	0	1
0	**1**	0	0	2
0	0	**1**	0	3
0	0	0	**1**	4
1	0	0	0	5
0	**1**	0	0	6
0	0	**1**	0	7
0	0	0	**1**	8

The following waveforms can be deduced from the functioning of the Johnson counter.