UNIT 12 REGISTERS, COUNTERS, MEMORY CIRCUITS AND ANALOG/DIGITAL CONVERTERS

Structure .

12.1 INTRODUCTION

In Unit 11, you learnt about the combinational logic circuits and adders. The operation of all such circuits is faithfully defined by the respective truth table and their outputs do not depend upon the previous input or output conditions. Hence they do not have memory. However, since the output of flipflops depend upon the previous input or output conditions or sequence of input or output, therefore these circuits, called sequential circuits, give us a basic memory element.

The registers and counters are combinations of several flipflops and their use in digital circuits is very important. A register is a group of memory elements which stores a binary word and it may modify the stored word in a particular fashion as is desired by the application in which it is used. It is capable of shifting the stored binary word a step or more towards left or right. A counter is basically a register which counts the number of clock (CLK) pulses arriving at the input. In this unit, you will learn about several types of registers and counters.

In a digital computer, the 'memory' is the key device. You often come across the terms RAM, ROM, floppy, and hard disk. The registers are memory devices. They are connected in different ways and are available as integrated circuits in the market. You know that the whole world is analog. To use a digital circuit or computer, you have to convert an analog quantity into a digital one so that the quantity can be acted upon or manipulated by the digital circuit or computer as per requirement. The output of the digital circuit is also in digital form which can not be perceived by you. For this purpose, you have to convert the digital output in analog form. Therefore. it is necessary to have circuits which will convert an analog quantity (such as voltage) into digital form and vice versa. Such circuits are analog-to-digital (AD) and digital-to-analog (DA) converters. In this unit, you will learn different kinds of memories, and AD and DA Converters also.

Objectives

After **studying** this unit, you should be able to

- explain the functioning of buffer and controlled buffer registers,
- describe the functioning of the shift register,
- describe the functioning of the shift left and shift right registers,
- explain the functioning of the controlled shift register,

a explain the construction and functioning of an asynchronous (ripple) counter,

- describe the functioning of ring and mod 10 (decade) **counters,**
- explain several memory terms used in digital circuits,
- explain the capacity of memory and specify how many bits can be stored in a memory device,

a describe general memory operation,

- explain and distinguish between RAM and ROM,
- . describe the functioning of a digital-to-analog and an analog-to-digital converters.

12.2 REGISTERS

A register is a group of memory elements which stores a binary word and it may modify the stored word in a particular fashion **as** is desired by the **application** in which it is used. It is capable of shifting the stored binary word a step or more towards left or right. In this section you will learn about them.

12.2.1 Buffer Register

The simplest kind of register is a buffer register which stores a binary word. It is made up of several D flipflops, the number of which depends on the number of bits present in a binary word. A buffer register for storing a 4-bit word, $X_3X_2X_1X_0$, with $Q_3Q_2Q_1Q_0$ as its output word is shown in Fig. 12.1.

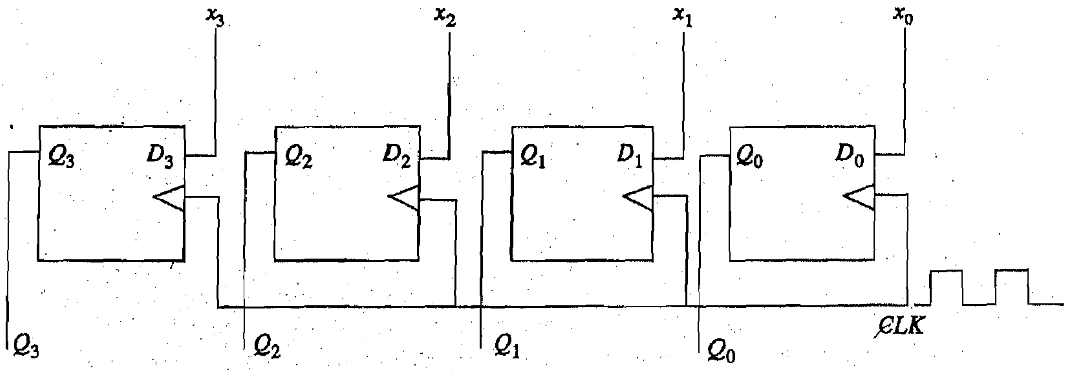


Fig. 12.1: Buffer Register.

Each flipflop is positive edge triggered. At every clock the output, Q of each flipflop is same as the input X. For this 4-bit register, we can write

$$Q_3Q_2Q_1Q_0 = X_3X_2X_1X_0$$

In chunked notation, this expression is written as

$$Q = X$$

This circuit is very basic. We should have some method to hold the input word till such time we are ready to store it. This is achieved by a controlled buffer register.

12.2.2 Controlled Buffer Register

A controlled buffer register is shown in Fig. 12.2. All flipflops are with CLEAR which resets flipflops when HIGH. The CLEAR is inactive when LOW. The control LOAD terminal when HIGH allows input X to reach the flipflop and does not allow when LOW. When CLR is HIGH, all flipflops reset and the stored word is

$$Q = 0000.$$

When CLR returns LOW, the register is ready for desired action.

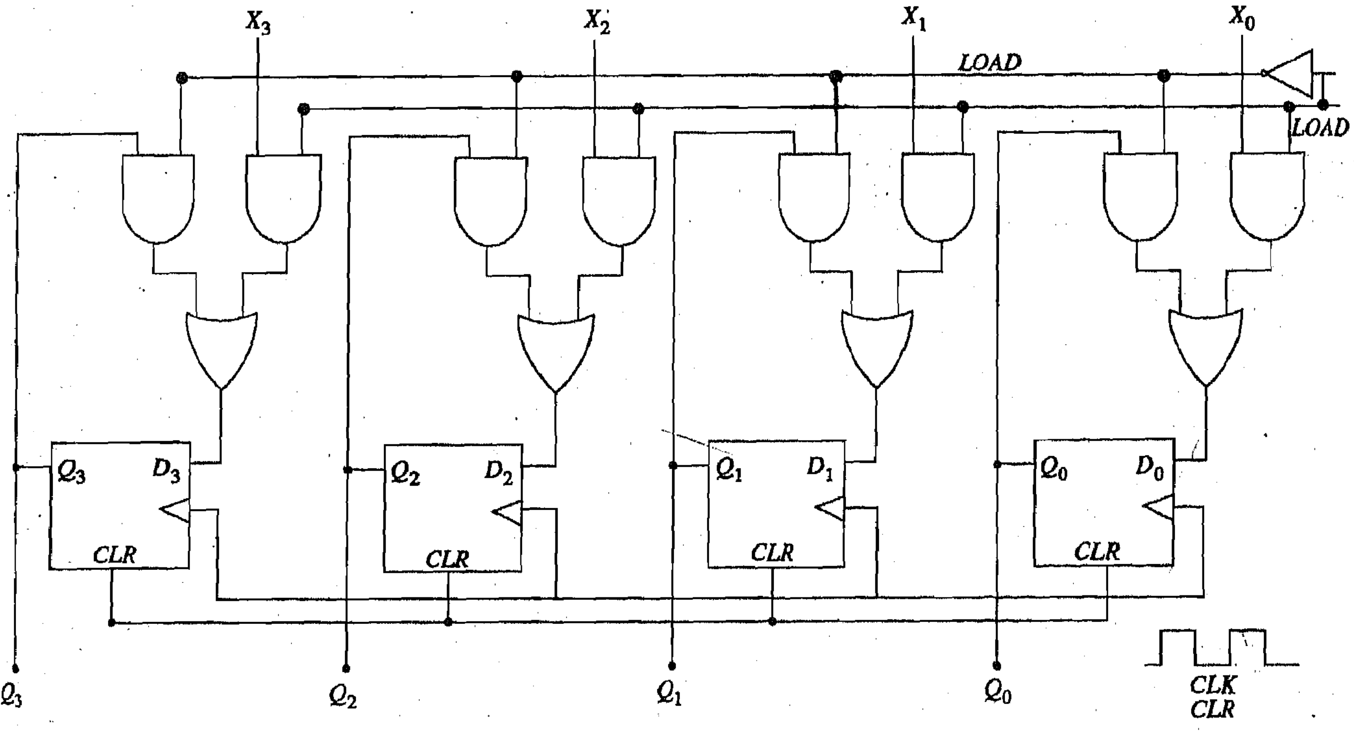


Fig. 12.2: Controlled buffer register.

The control terminal LOAD determines the circuit function. When LOAD is HIGH, the data X is allowed to reach the flipflop.

However when LOAD is LOW, $\overline{\text{LOAD}}$ is HIGH which allows the Q outputs to go to D inputs. It means that so long as the LOAD is LOW, the input data X is circulated or retained at the PGT of the each CLK. That is, the contents of the register continue to remain unchanged so long as LOAD is LOW.

When the LOAD is made HIGH, the word or data X is transmitted to the D inputs and the flipflops are ready to change. When the PGT of the CLK arrives, the X input is loaded and is available at Q outputs, and

$$Q_3Q_2Q_1Q_0 = X_3X_2X_1X_0$$

With LOAD returning to LOW, the input word is stored. That is, so long as the LOAD remains LOW, it is not affected even when X input is changed. In this kind of register, as is seen from the circuit, the input is given to all the flipflops simultaneously and the output is also obtained **from** all the flipflops simultaneously. This is quite often referred to **as parallel-in/parallel-out** register.

12.2.3 Shift Registers

The shift registers move the stored word towards left or right. Therefore there are two types of shift registers — Shift-left and shift-right registers. Shifting of bits of the stored word towards left or right is essential in arithmetical operations.

Shift Left Register

A register which shifts the bits of the stored word towards left, called shift-left **register,** is shown in Fig. 12.3. As is clear from the circuit, the data input D_{in} sets up first flipflop, and the Q_0 output of this **flipflop** sets up second flipflop, Q_1 sets up the third and Q_2 sets up the fourth. Since the data is given to the input of the first flipflop, **i.e.,** D_{in} and the output **is** obtained **simultaneously** from all the flipflops, the circuit **is** known as **serial-in/parallel-out.**

The working of shift-left register can be understood by the following example:

Consider that the input data D_{in} is 1, **i.e.,** the input to **flipflop-1,** $D_0 = 1$ and the initial output

Q = 0000.

That is, initially the inputs **to all** the other three flipflops are 0. Now with the arrival of the **PGT** of the first CLK, **the Q_0 output** is 1, and the stored word becomes

Q = 0001.

Now with D, = **1 and** $D_0 = 1$, when the **PGT** of the second CLK arrives then **first** and second flipflops are set **making** the register output to be

Q = 0011.

Now $D_2 = 1$, $D_1 = 1$, and $D_0 = 1$. **When** the PGT of the third CLK arrives then first, second and third **flipflops are** sat making the register output to be

Q = 0111.

Similarly when **the PGT** of the fourth CLK arrives, then output becomes

Q = 1111.

The stored word is thus-1111 and it remains unchanged so long as $D_{in} = 1$. However, if $D_{in} = 0$, then with successive **CLK** pulses the register output or content becomes

At 1st **CLK**	Q = **1110**
At 2nd CLK	Q = **1100**
At 3rd CLK	**Q = 1000**
AT 4th CLK	**Q = 0000**

This word **0000** remains **stored so long** as D„, = 0. The entire operation of the shift-left **register** in **terms** of its timing diagram is shown in Fig. 12.4.

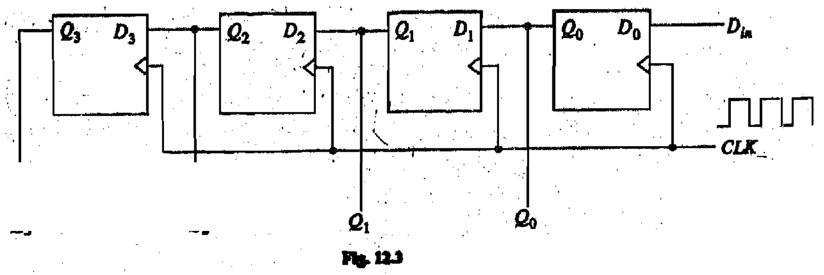


Fig. 12.3

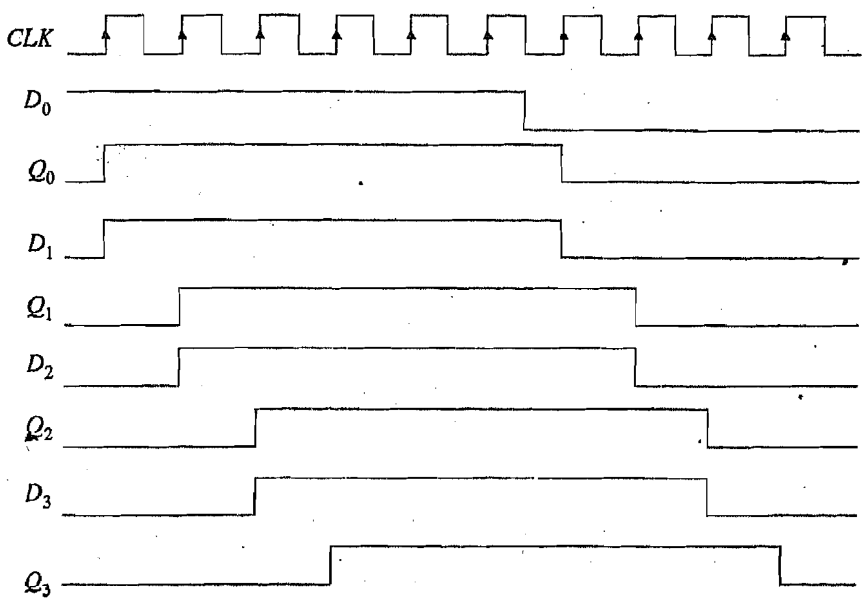


Fig. 12.4: Timing diagram of shift-left register.

Shift Right Register

The circuit for a shift-right register is shown in Fig. 12.5. The data input, D , is given to the input of the fourth flipflop as D_3. The Q output of each flipflop is fedback to the D input of the previous flipflop, i.e. Q_3 is given to D_2, Q_2 is given to D_1, and Q, is given to D_0. When the PGT of the CLK arrives, the stored word shifts one step to its right.

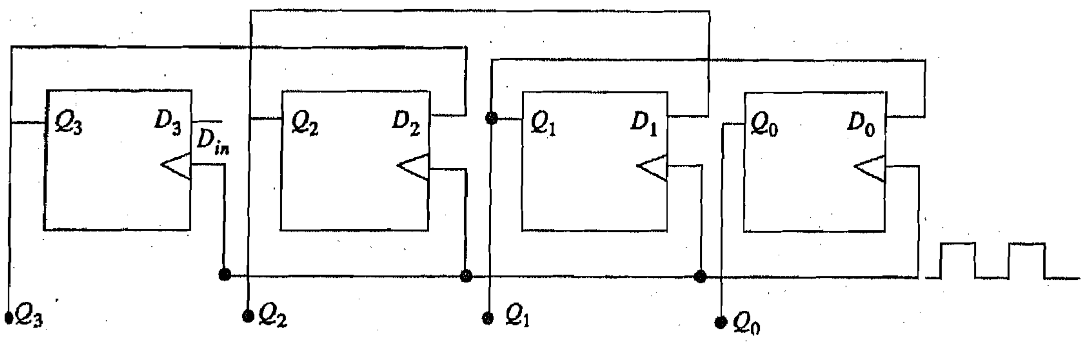


Fig. 12.5: Shift-right register.

The operation of the'shift-right register can be understood as follows. Consider that in the beginning D_{in} = 1, and

Q = 0000.

At the arrival of the PGT of the first CLK, D_3 = 1, and all other D inputs are 0. Therefore, the fourth flipflop is set and the stored word is

Q = 1000.

Now D_3 = 1 and D_2 = 1. When the PGT of the second CLK arrives, third nnd fourth flipflops are set; and the stored word becomes

Q = 1100.

Similarly, with the arrival of the PGT of the third CLK, the stored word becomes

Q = 1110.

And with the arrival of the PGT of the fourth CLK, the stored word becomes

Q = 1111.

12.2.4 Controlled Shift Register

In general the operation of a shift register is controlled by some additional arrangement so that when the PGT of the CLK arrives the stored word should or should not change as desired by the application. Such a controlled shift-left register is shown in Fig. 12.6.

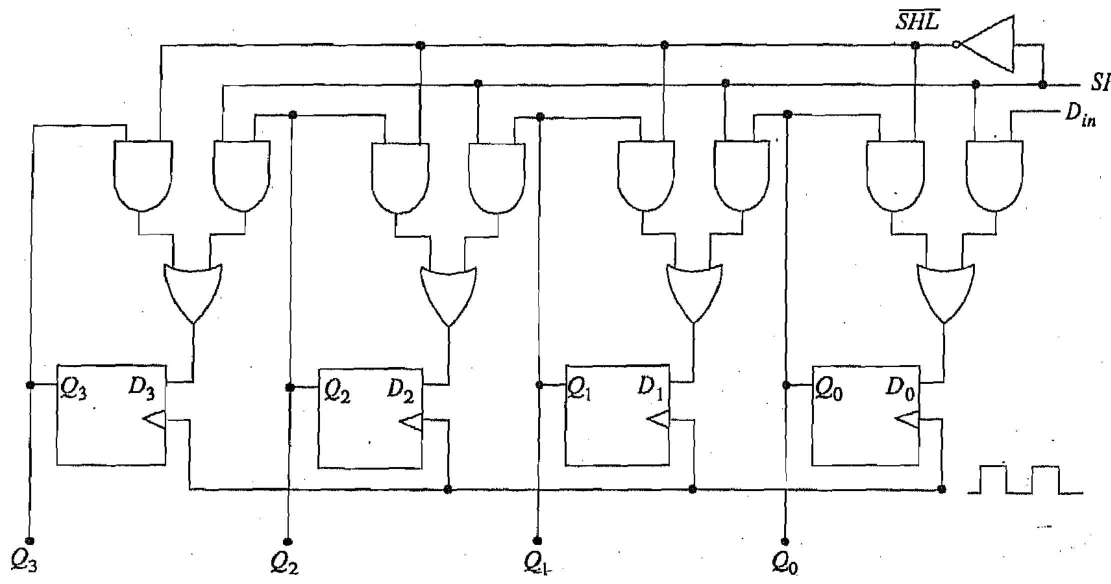


Fig. 12.6: Controlled Shift-left register.

Its operation is as follows: When the control input signal SHL is 0, the inverted signal $\overline{SHL}$ is 1. In this condition, the Q outputs of the flipflops are circulated back to their respective D inputs. **It** means that the data stored in the register remains stored even at the arrival of the PGT of the CLK. That is the stored word is stored indefinitely.

Let us now reverse the control signal. When the control input signal SHL is 1, the inverted signal $\overline{SHL}$ is 0. In this condition, D_{in} is available at the D_0 input, and at the arrival of the PGT of the first CLK the first flipflop is set by D_0. With successive CLKs; Q_0 sets second flipflop, Q_1 sets the third, and Q_2 sets the fourth flipflop. At each PGT of the CLK, the stored word shifts a step towards the left.

The loading of the word to be stored in this kind of register is done serially, that is the word is loaded by entering one bit per CLK. To store a 4-bit word we require four CLK pulses. For example, X = 1001 is loaded serially as follows:

Keep SHL = 1, and make D_{in} = 1. At the first CLK

Q = 0001.

Now keeping SHL = 1, make D_{in} = 0. At the second CLK

Q = 0010.

At the third CLK

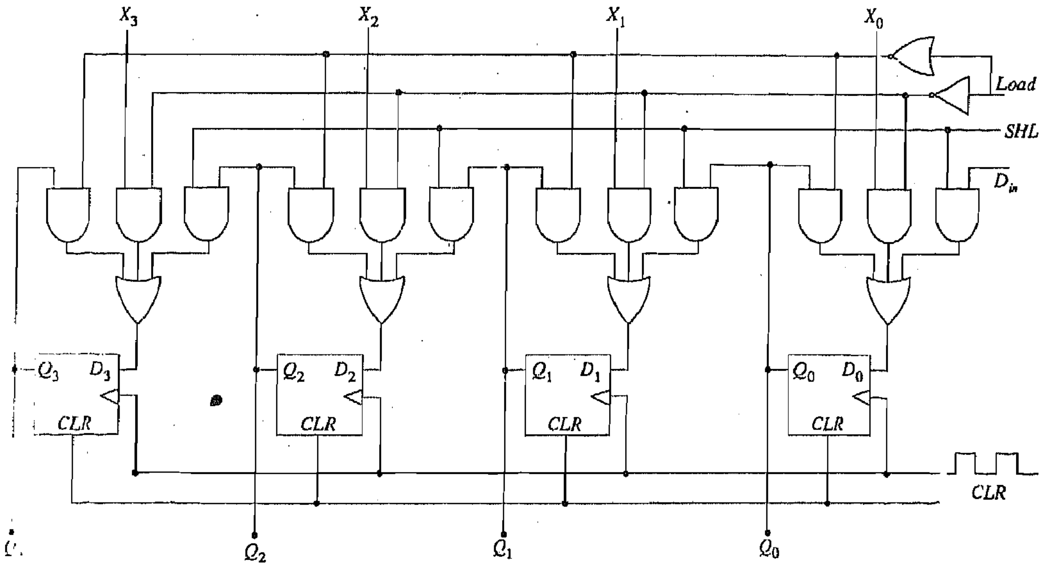
Q = 0100.

Now keeping **SHL** = 1, make D_{in} = 1. At the fourth CLK

Q = 1001.

The data is thus entered serially and stored word is available parallelly from all the Q outputs.

All the bits of a word can, however, be loaded simultaneously and it needs only one CLK pulse as is done in buffer register. The circuit for this kind of loading is given in Fig. 12.7 which can be used for serial as well as parallel loading of a word to be stored.



Pig. 12.7: Controlled shift register witlı parallel as well as serial loading arrungcrncnt.

If LOAD and SHL arc 0, the output of NOR gate is 1. With this condition the Q outputs are circulated back to their respective D inputs. The previously stored word continues to be stored. The register, in this state, is known as inactive register.

If the LOAD is 0 and SHL is 1, the register is used for serial loading as is done in case of register shown in Fig. 12.6. If Load is 1 and SHL is 0, then X bits set the D inputs simultaneously on the first CLK itself. This is the case of parallel loading.

For a word of more bits to be stored, more flipflops are required. Actually you require the same number of flipflops as is the number of bits in the word to be stored.

12.3 COUNTERS

A counter is an equivalent of binary odometer. It counts the number of CLK pulses which arrive at the CLK input. Basically, there are two types of counters, asynchronous (ripple) and synchronous, We shall learn about them in this section.

12.3.1 Asynchronous (Ripple) Counter

Fig. 12.8 shows a 4-bit binary counter circuit which is made by using JK flipflops. All the JK inputs are kept at 1. The CLK signal is given to the CLK input of the first flipflop. The Q_0 output is given to the CLK input of the second flipflop, Q_1 output is given to the CLK input of the third, and so on. The CLR inpul is activated when it is made 0. All CLR inputs have been joined together so that all the flipflops could be reset simultaneously. Such a counter where each flipflop output serves as the CLK input for the next flipflop is known as asynchronous counter. This name is given because all the flipflops do not change state in exact synchronism with the CLK pulses. Only the first flipflop responds to the CLK pulse, while all others wait for the previous flipflops to change state. Therefore, there is a delay between the responses of consecutive flipflops. This type of counter is also known as ripple counter,

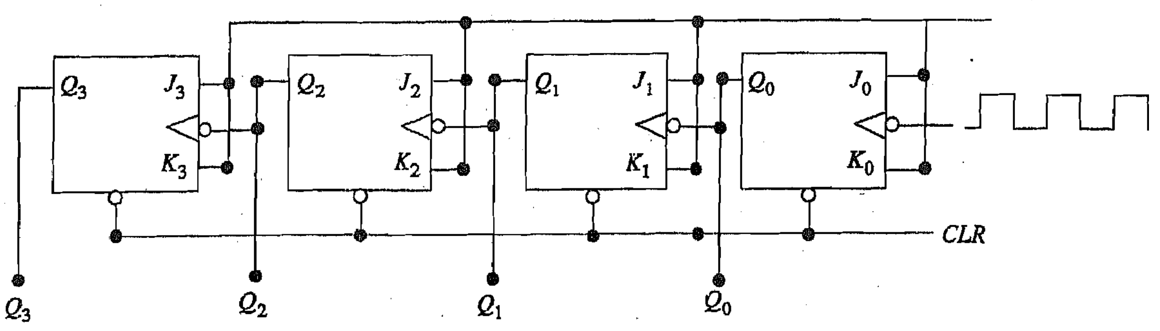


Fig. 12.8: Asynchronous **(ripple) counter.**

Let us understand the operation of the ripple counter. The clock pulses are applied to the CLK input of the first flipflop. Since the flipflops are driven by the NGT of the CLK, with J = K = I, the first flipflop toggles when the CLK pulse goes form 1 to 0. The Q_1 output of second flipflop toggles when Q_0 output of the first flipflop goes from 1 to 0, and so on. With CLR = 0, all the flipflops are reset to

Q = 0000.

After resetting keep CLR = 1. Now the counter is ready to count. The Q_0 toggles for each NGT, Therefore, when the **NGT** of the first **CLK** arrives, then the Q output is

Q = 0001.

At the second CLK, Q_0 toggles from 1 to 0 which acts as a NGT for the CLK input of the second flipflop, the Q_1 output of which toggles to 1. Therefore,

Q = 0010.

At the third CLK, Q_0 toggles from 0 to 1, and there. is no change in Q_1. Therefore,

Q = 0011.

At the fourth CLK, Q_0 toggles from 1 to 0 resulting in toggling of Q_1 from 1 to 0. The Q, going from 1 to 0 acts as a NGT for the CLK input of the third flipflop, **the** Q_2 output of which toggles from 0 to 1. Therefore,

Q = 0100.

The Q output of the counter at each CLK is summarised in Table **12.1.**

Table 12.1:

No. of CLK pulses	Q
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

Next CLK resets all the flipflops and the Q outputs on successive CLK would be

16	0000 (recycles)
17	0001
18	0010
...	....
..	....

While analysing the Q outputs, we iind that whenever a flipflop resets to 0, the output of the next flipflop is 1. That is, resetting of a flipflop send a carry to the next higher flipflop. Therefore, the counter acts like a binary odometer. The Q_0 output of the first flipflop acts as a LSB and that of the last flipflop as the MSB. This would now be clear as to why asynchronous counter is called a ripple counter. It is because the carry in the output moves like a ripple on water.

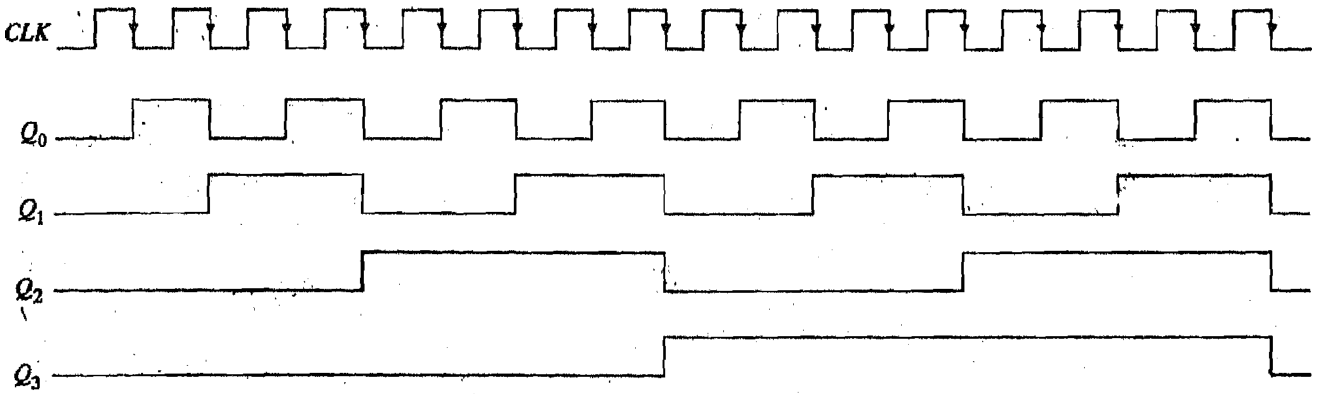
Mod of a Counter

The counter described above has 16 distinct states or outputs (0000 to 1111). It is said that the Mod number of this counter is 16. The Mod number of a counter is equal to the number of states which the counter goes through in each complete cycle before it recycles back to its starting state. The Mod number can be increased by increasing the number of flipflops, If n is the number of flipflops used in a counter, then

Mod Number = 2^n.

Frequency Division

The output of each fliptlop and the CLK are shown in Fig. 12.9, It is clear that the frequency of Q_0 output is half the frequency of the CLK, The Q_0 output acts as a CLK to the second flipflop, and the frequency of its Q_1 output is half the frequency of Q_0 or one-fourth the frequency of the CLK.



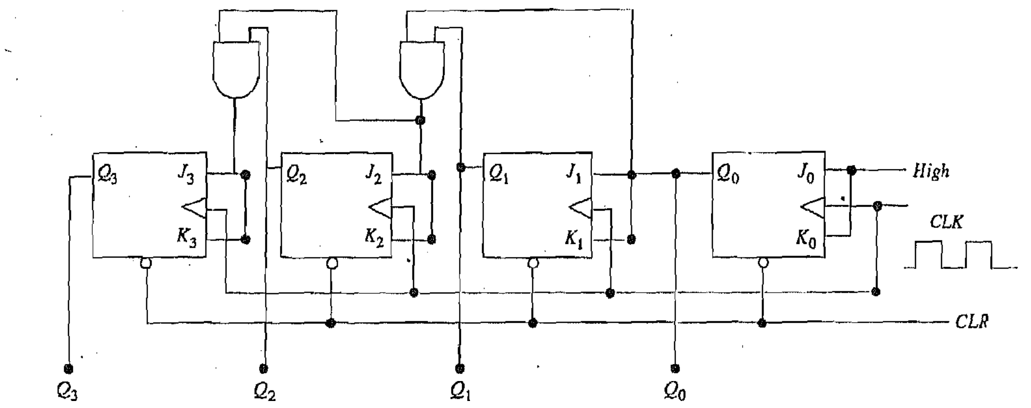
First flipflop divides by	2
Second flipflop divides by	4
Third flipflop divides by	8
Fourth flipflop divides by	16
nth flipflop divides by	2^n.

SAQ 1

'If the frequency is 100 kHz, what will be the output frequency of the third flipflop of a ripple counter?

12.3.2 Synchronous Counter

As stated in the previous section, there is a lot of time delay in ripple counter because a carry has to pass through n flipllops. Therefore, the ripple counters are very slow. IS Tpd is time delay for one flipflop, then for *n* flipflops the time delay is *n*Tpd. Therefore, there is a need for synchronous counter in which all the flipflops respond on each CLK pulse simultaneously. The circuit for a synchronous counter is given in Fig. 12.10.



Pig. 12.10: Synchronous counter.

The CLK inputs of all the flipflops are connected with each other so that the CLK signal reaches them simultaneously. Similarly, the CLR inputs of all the flipflops arp connected with each other so that they can be reset simultaneously by making CLR = 0. All J and K inputs have not been connected to each other' as is the case in ripple counter. The JK inputs of the first flipflop are always kept at 1. The flipflops toggle at the arrival of the PGT of the clock pulse at their CLK inputs provided their JK inputs are at 1. The operation of this counter can be understood as follows:

When reset in the beginning, the Q output is

Q = 0000.

At the arrival of the PGT of the first CLK, Q, toggles from 0 to 1 bringing JK inputs of the second flipflop also to 1. Now this flipflop is also ready to toggle. However, by now the PGT of the CLK pulse has disappeared. It has to wait for the PGT of the second CLK. As is clear from the circuit, the JK inputs of third and fourth flipflops continue to be at 0, hence they are in no change condition. Thus, at the arrival of the first CLK, Q = 0001.

Now at the PGT of the second CLK, Q_0 toggles from 1 to 0 and Q, toggles from 0 to 1. However, the JK inputs of the third flipflop continue to be nt 0 because the inputs to the AND gate (the output of which is connected to these JK inputs) are Q_1 = 1 and Q_0 = 0. Therefore, it is in no change condition and hence Q_2 continue to be at 0. Similarly, the JK inputs of the fourth flipflop are at 0 because Q_2 = 0 and therefore Q_3 continues to be at 0. Thus at the arrival of the PGT of the second CLK,

Q = 0010.

At tlie arrival of the PGT of the third CLK, the JK inputs of second, third and fourth flipflops are at 0, therefore they are in no change condition. Only the first flipflop is ready to toggle from 0 to 1. Thus at the third CLK,

Q = 0011.

Now since Q, and Q_0 are at 1, ... therefore JK inputs of the third flipflop are at 1. However, the JK inputs of the fourth flipflop are still at 0. Hence first three flipflops are ready to toggle at the arrival of the PGT of the fourth CLK. Thus Q_0 and Q_1 toggle from 1 to 0, and Q_2 toggles from 0 to 1. Therefore, at the fourth CLK,

Q = 0100.

Successive, Q outputs are 0101, 0110, and 0111. At the arrival of the eighth CLK, the JK inputs of all flipflops arc at 1. The Q outputs of all the flipflops toggle, and we have

Q = 1000.

The successive CLK pulses change the Q outputs in the same way as described above. The Q output at each CLK is summarised in Table. 12.2.

Table 12.2:

No, of CLK pulses	Q
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

At the arrival of the PGT of the next CLK, the counter resets to Q = 0000.

A countcr of any length can be built by adding more number of flipflops. The advantage of synchronous counter is that it requires only one propagation delay time in getting the Q output. The Mod of this counter is also 16 (= 2^4).

12.3.3 Controlled Synchronous Counter

The circuit of the controlled synchronous counter is shown in Fig. 12.11. The COUNT is the control input. When the COUNT is at 0, the JK input of all the flipflops are at 0 keeping the flipflops in no change condition. When the COUNT is at 1, the circuit is the synchronous counter which works exactly in the same way as the counter of

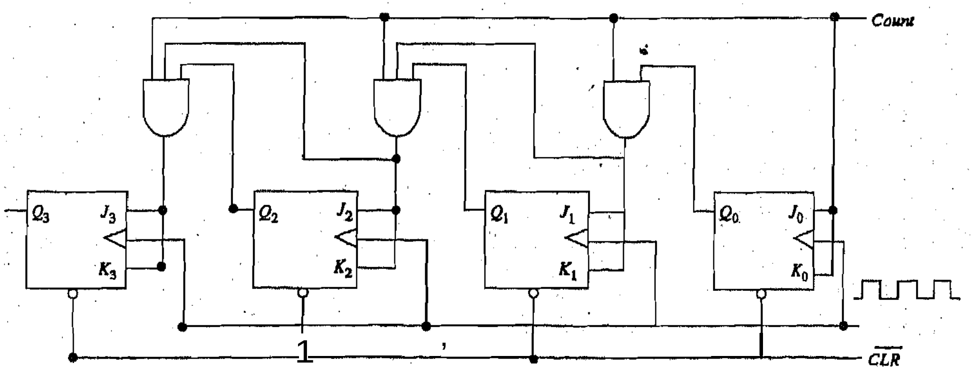
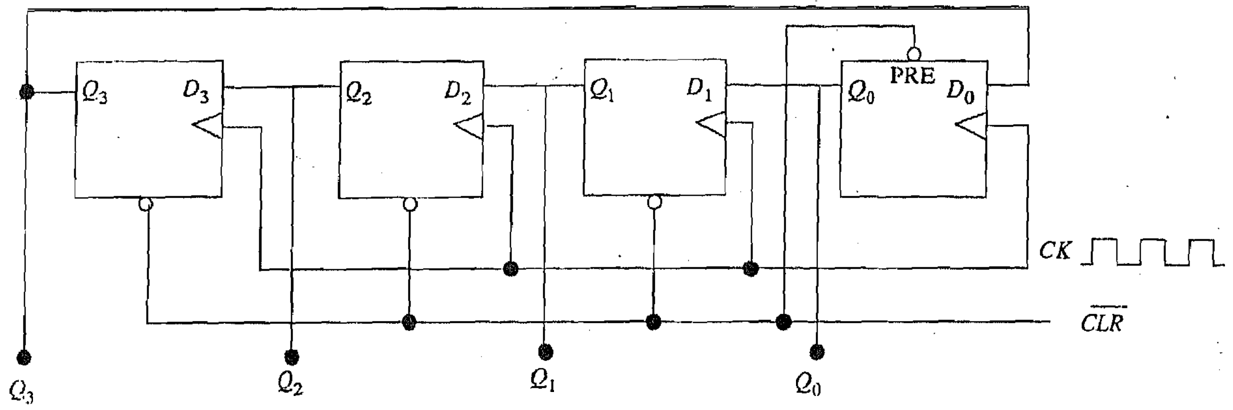


Fig. 12.11: Controlled synchronous counter.

12.3.4 Ring Counter

The ring counter does not count the binary number. The.Q output of this counter has only a single 1 bit and all other bits are 0. At each CLK the bit 1 shifts a step to its left. The digital circuit of the ring counter is shown in Fig. 12.12. It is made up of D flipflops. Note that the CLR inputs of second, third and fourth flipflops are connected with the PRESET input of the first flipflop. It means that w.hen CLR is brought at 0, it presets the Q_0 to I, and resets Q_1, Q_2 and Q_3 outputs to 0.



Pig. 12.12: Ring Counter.

The working of the ring counter can be understood as follows. When the CLR is made active, i.e. when it is made 0, the first flipflop is set and all others are reset, Therefore, Q output is

Q = 0001.

Now Q_3 = 0 is fedback to D_0 input of the first flipflop. Therefore, at the arrival of the PGT of the first CLK, Q_0 is 0 and Q_1 is 1, while Q, and Q_3 continue to be 0. Thus, at the first CLK,

Q = 0010.

At the time the PCT of the second **CLK** arrives, D_0, D,, and D_3 are at 0, and D_2 is at I. , Therefore, the Q output is

Q = 0100.

Similarly at the arrival of the PGT of the third **CLK,** the Q output becomes,

Q = 1000.

The **PGT** of the fourth CLK starts the cycle again, and

Q = 0001.

Thus we find that bit I shifts a step to its left and it rotates back to its initial position, and so on. It is because of this effect that it is known as ring counter. Instead of a ring of four bits, if you want a bigger ring tlien add more flipflops.

12.3.5 Mod 10 (Decade) Counter

The Mod number of a Mod 10 counter is 10, i.e. it counts from 0 to 9 and then resets to 0. This is an asynchronous counter and its digital circuit is given in Fig. 12.13. The circuit counts from 0000 to 1001 and then resets to 0000. As described in the section on' ripple counter, the Q outputs of the counter at the arrival of the NGTs of the first nine CLK pulses are summarised in Table 12.3.

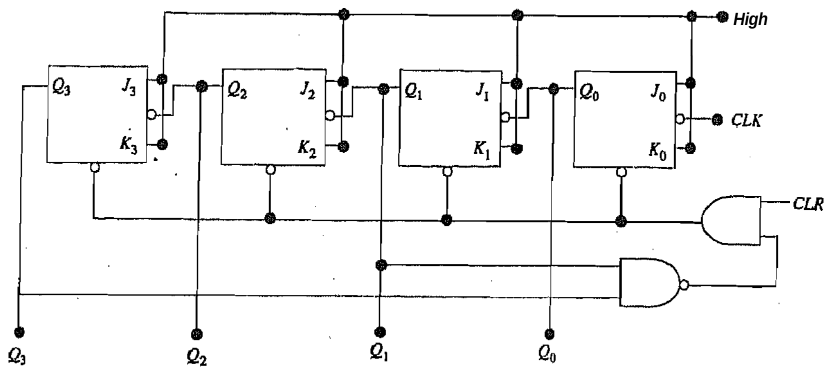


Fig. 12.13: Mod-10 (Decade) counter.

Table 12.3:

No. of CLK pulses	Q	Decimal Equivalent
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	0000	0 (resets)

The circuit skips the states from 10 lo 15, i.e. from 1010 lo 1111. The circuit is made lo skip these stales by the combination of NAND and AND gales present in the circuit. The idea is that when Q = 1010 is expected at the tenth CLK, the flipflops should be cleared to be reset to 0000 which is done by bringing CLR input to 0. This is achieved by connecting Q_1 and Q_3 to the inputs of a NAND gate which gives output 0 when its inputs are 1 (which is the case when Q = 1010 is expected at the tenth CLK). The output 0 of the NAND gate makes the AND gate output to be 0. This makes the CLR active and the flipflops reset to

Q = 0000.

When the CLR is a made inactive, i.e.. When CLR = 1, the counter becomes ready'to count once again.

Since it takes 10 CLK pulses to resct tho countcr, the frequency of the Q_3 output is nne-tenth of that of the CLK. It is therefore called a divide-by-10 circuit. It is used in BCD applications and frequency counters.

Example 12.1

Design a Mod 5 counter.

Recall that in a Mod 10 counter. the expected output of the counter on the arrival of the 10th CLK, i.e. 1010 was used to reset the counter so as to skip the states from 1010 to 1111.

In this example a mod 5 countcr which will count from 0 to 4 is to be designed. It means that at the arrival of the 5th CLK, the counter should reset. For counting upto 4, not more than three bits are required. Therefore, we require three JK flipflops. The expected output at the arrival of 5th CLK is 101 which should be used to activate CLR to reset the flipflops. The required circuit is given in Fig. 12.14.

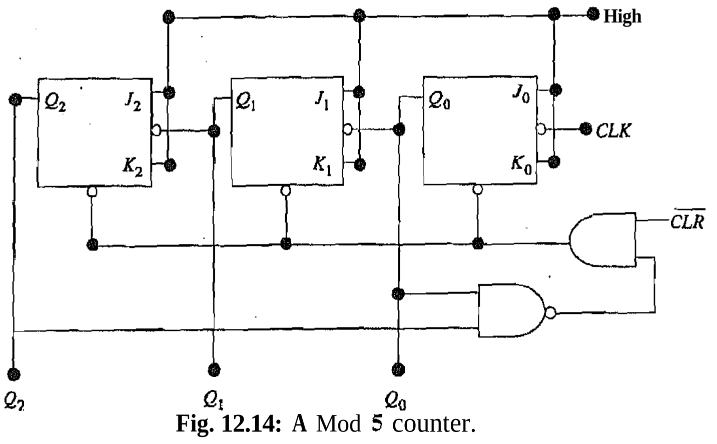


Fig. 12.14: A Mod 5 counter.

SAQ 2

What is the Mod of a counter which consists of six flipflops?

12.4 SEMICONDUCTOR MEMORIES

The advantage of digital systems over analog systems is their ability lo store information for short as well as long periods which makes them versatile. A digital computer has a minimum amount of memory with the help of which it is able to manipulate information or data as desired by us. It also has memory which makes it capable of storing this information as long as we want and make it available to us whenever we want.

We have already studied about the basic memory element which stores a single bit, that is the flipflop. We have also learnt about registers which store a word of any number of bits. The registers are very high speed memory elements and are used extensively in the internal operation of a digital computer. With the invent of integrated circuit technology and its further advancement in LSI (Large Scale Integration) and VLSI (Very Large Scale Inlegration), a large number of registers can be obtained on a single chip.

The cost of these semiconductor devices is also decreasing. However, the cost of these devices per bit of storage is very high which prohibits their use as mass storage devices. A computer has internal memory which is constantly in communication with the central processing unit of the computer as a program of instructions is being executed. The program and any other information or data used by the program are also stored in the internal memory.

The mass storage memory devices are external to the computer and are capable of storing millions of bits even without requiring any electrical power. The mass storage memory is generally very slow compared to the internal memory and the information stored is the one which is not currently required by the computer, It is supplied to the computer only when required. The mass storage memory devices are floppies, magnetic tapes and disks, etc. The cost of per bit storage of these device is much less compared to the internal memory.

12.4.1 What a Memory is!

A 'memory' is simply an array of registers, and each register storing a word. Every register has an address number which identifies the location of a word in a memory. The location of a word is nothing but the register that stores the word to be identified. The address of each location is unique and is described by a binary number. To illustrate, let us consider that we have a memory which consists of eight registers. It is clear that this memory has eight memory locations. The unique addresses of the memory locations are given in Table 12.4.

Table 12.4:

Address	Location
000	word 0
001	word 1
010	word 2
011	word 3
100	word 4
101	word 5
110	word 6
111	word 7

Each word in the memory is thus identified by an address. By a read operation, the binary word stored in a memory location is sensed and, if desired, it can be transferred to another device. For example, if we have to read word 6, then we have to do read operation on address 110. By a write operation, a new word can be placed or stored on a particular melnory location.

The memories are volatile and nonvolatile. A memory is volatile if it requires electrical power to store information and if the power is removed then the stored information is lost. Many types of semiconductor memories are volatile. The nonvolatile melnory retains the stored information even when electrical power is removed. The mass storage memory devices fall in this category. The other types of memories like Random-Access Memory (RAM) and Read Only Memory (ROM) will be described in later sections.

12.4.2 Capacity of Memory

Before understanding the meaning of capacity of memory let us know some of the melnory terms. A device, such as a flipflop,· which can store a single bit (0 or 1) is called a memory cell. In a memory a group of hits or cells which represents instructions or data is known as a melnory word. A register consisting of four flipflops is a memory which can store a 4-bit word. Similarly, a register having eight flipflops is a memory which can store a 8-bil word. The size of the word in modern computers range from 4 to 64 bits. A 4-bit word is called a nybble and 8-bit word is called a byte. A byte is the most commonly used word size.

The capacity of a memory is a term used to express how many bits can be stored in a particular memory device or in a complete memory system. For example, let us say that we have a memory which can store 2048 eight-bit worcls. This memory can storc 2048 × 8 = 16384 bits and we say that this memory can store 16384 bits. Another way to express this capacity is as 2048 × 8. This kind of expression of memory means that there are 2048 words and the size of the word is 8 bits. The nuntber of words in a memory is generally a multiple of 1024. The figure of $1024 = 2^{10}$ is commonly represented as '1K'. Thus memory capacity of 2048 X 8 is also expressed as 2K X 8. For larger memories, '1M' or '1 rneg' is used for 2^{20} = 1,048,576. Therefore, a 4M × 8 memory has a capacity of 4,194,304 × 8 or alternatively of 33,554,432 bits.

Example 12.2

A user has two memory devices. One of these stores 10M words of 8-bit size, while the other stores 2M words of 16-bit size. Which of the two stores most bits7

Solution

The two memories are of 10M × 8 and 2M × 16.

10M × 8 = 10 × 1,048,576 × 8 = 83,886, 080 bits.

2M X 16 = 2 X 1,048,576 X 16 = 33,554,432 bits.

Therefore, the memory of 10M × 8 stores more bits.

SAQ 3

A certain memory is specified as 32K × 8.

a) What is the size of the word?

b) What is the total number of bits stored by the memory?

12.4.3 Random-Access Memory (RAM)

A Random-Access Memory (RAM) is also known as read-write memory. It is a group of registers which have their unique addresses, and using an appropriate address the stored word on a memory location can he read and new contents, if desired, can be written on this location. Actual physical location of a stored word in RAM does not tnake any difference because the access time (which is the speed of a metnory device, that is the time required to perform the read operation) is same for any address in the memory. The semiconductor RAMs.are volatile, because when the electrical power is turned off the stored data is lost.

While working with a computer when a user is giving instructions or doing some calculations using a program, it is the RAM that is being continuously used to read the stored information and write the new informations. You might have heard the term being used that a particular computer has 1 or 4 MB (Mega Byte) RAM or so.

12.4.4 General Memory Operation

Despite the fact that the internal operation of each type of memory is different, the general memory operation remains same for all. Every memory system will have terminals for data input, data output, address input, selecting read or write operation and for enabling or disabling the memory operation.

In a general memory operation, first the address of a memory location is selected where the read or write operation is to be performed. Decide whether you want to perform read or write operation. If you want to write, then perform the write operation and supply the data to the memory. If you want to read, then perform the read operation and hold the output data coming from the memory. If you want that the mernory should respond to the address and read/write operation, then enable the mernory and if you do not want the memory to respond then disable the memory.

To illustrate the aforesaid operation consider a 16 × 4 memory device shown in Fig. 12.15. Since the word size is 4 bit, therefore, it has four data input lines and four output data lines. It will also have four address lines because the given memory device has 16 memory locations which can be expressed by 4-bit addresses. It has one read/write command terminal, it will read if kept at 1 and write if at 0. It has one enable/disable terminal. In the diagram shown, if this terminal is kept at 1, it enables the memory and it disables if kept at 0. A virtual arrangement of memory cells into 4-bit words is shown in Fig. 12.16 alongwith their address.

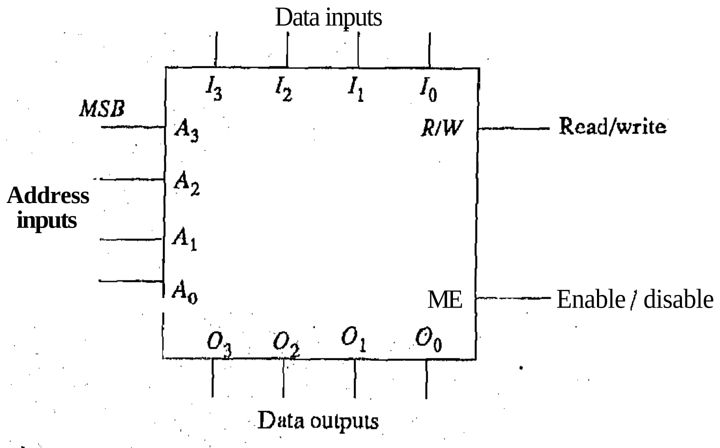


Fig. 12.15: A 16 × 4 memory.

Memory Cells				Addresses
0	0	1	1	0000
1	0	0	1	0001
0	0	0	0	0010
0	1	1	1	0011
1	0	1	0	0100
1	0	0	0	0101
1	1	1	0	0110
0	0	0	1	0111
1	0	1	1	1000
0	0	0	0	1001
1	1	0	0	1010
0	1	0	1	1011
1	0	0	1	1100
1	1	1	1	1101
0	0	0	1	1110
1	0	1	0	1111

Fig. 14.16: Virtual arrangement of memory cells into sixteen 4-bit words

As a further illustration, let us say that you want to change the word 1111 stored in the fourteenth location to 0101. To do so, choose the address 1101, keep read/write terminal at 1 so that write operation is chosen, and then Seed the desired word 0101 to the data input. Thus the new word is stored in place of the old one.

12.4.5 Read Only Memory (ROM)

The Rcad-Only Memory (ROM) is a broad class of semiconductor memories which are designed for those kinds of applications where only read operation is required. These memories hold the data permanently. In general, no new data can be written on ROM but it can bc read. The data to be stored permanently in ROM i s selected and built in by the manufacturer at the time of IC fabrication. However, there are some varieties of ROM in which data can be entered electrically once only. The process of entering the data is known as programming or burning the ROM. Suck ROMs are called PROMs (Programmable-ROM). In some other ROMs the data stored can be erased and the ROM can bc reprograinmed. Such ROMs are called EPROMs (Erasable-PROM). All ROMs are nonvolatile, that is they keep storing the data even when electrical power is removed.

A typical block diagram of a 16 × 8 memory is shown in Fig. 12.17. It has Sour address lines, eight terminals for data output and one terminal called chip select (CS) which enables or disables the memory. To read the data, say, at the location with address 1010, we have to apply $A_3A_2A_1A_0 = 1010$ to the address inputs and then sclcct the chip select so as to enable the memory. The data output terminals will show the actual word stored in that location.

Fig. 12.17: Block diagram of a 16 × 8 ROM.

12.5 A/D AND D/A CONVERTERS

As pointed out in the introduction of this unit, digital systems or computers perform all of their functions and internal operations using digital circuits which require digital inputs. *A* digital quantity will have a value either 0 or I, while an analog quantity can take auy value over a continuous range of values and its exact value is significant. Most physical variables are analog in nature, such as temperature, pressure, light intensity, audio signals, position, speed, etc. Therefore, it is essential to put an analog quantity to be analysed using a digital system first in a digital form. The analog-to-digital (A/D or ADC) converter is a digital circuit which converts an analog quantity into digital form consisting of a number of bits that represents the vblue of the analog input. This circuit is used as an interface between the digital system or computer and the analog system of the input stage. The output of a digital system is digital and, has to be converted back into analog quantity. The digital-to-analog (D/A or DAC) converter serves this purpose and its output is a proportional analog voltage or current corresponding to an analog quantity. This is used as an interface between the digital system or computer and the analog system of the output stage.

Pictorially this is summarised in Fig. 12.18. Let us say that in a physical system a quantity, such as temperature, is to be controlled using a computer. This physical quantity is first converted into a corresponding voltage or current with the use of a transducer. A transducer is a device which converts a physical variable into an electrical signal. Thermistors, bolometers, photocells, Lhermo-couples are some of the commonly available transducers. Actuator used in this illustration is a device that controls the physical quantity, temperature, in a computer controlled system. In this section, we shall learn about design and working of digital-to-analog and analog-to-digital converters.

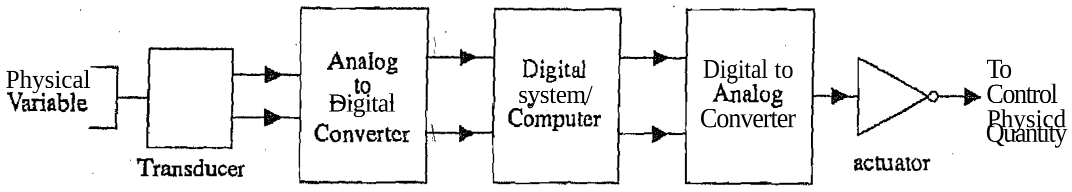

Physical Variable
Transducer
Analog to Digital Converter
Digital system/ Computer
Digital to Analog Converter
actuator
To Control Physicd Quantity

Fig. 12.18: ADC and DAC used as **interfaces.**

12.5.1 Digital-to-Analog Converter

We are first treating Digital-to-Analog Converter (DAC) because the Analog-to-Digital Converter (ADC) requires the use of DAC. The circuit for DAC takes the BCD or binary input and converts it to a voltage or current that is proportional to the digital value. The digital input is generally derived from an output register of the digital system which can theoretically be of any number of bits, In general, the registers used are 8-bit registers. For the purpose of an illustration, let us consider that the digital output from the digital system is of four bits. Therefore, we require a DAC that can convert a 4-bit digital output to a proportional analog value.

A block diagram of such a DAC is shown in Fig. 12.19. It has four binary input lines representing $A_3A_2A_1A_0$ and one output line representing corresponding proportional analog quantity. Each 4-bit input has unique proportional output voltage. There are 2^4 = 16 states that the binary input can have. Let us say that each input specifies a decimal number. Let us designate 1V output equivalent to decimal number 1, 2V as number 2, and so on.

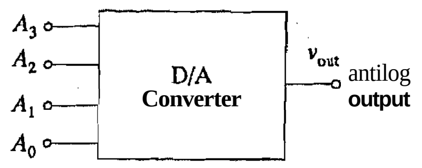


Fig. 12.19: Block diagram of DAC.

The digital input and the corresponding proportional voltage as the analog output is summarised in Table 12.5 In this example, the analog output voltage is equal in volts to the binary number. The out put voltage could be twice the binary number or any multiple. We can, therefore, write

Analog output = k × digital input

Where k is proportionality factor, a constant for a (12.1) given DAC.

The value of k in the given example is 1V, therefore V_{out} is 1V times the digital input. For 01LO, = 6_{10}, we get

V_{out} =. 1V × 6 = 6V.

Table 12.5:

A_3	A_2	A_1	A_0	V_{out}
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

Analog Output

The DAC output is technically not an analog quantity. It can have only specific values. In the above example, it can have values only from 0 to 15 in steps of 1, that is 1, 2, 3,..., 15. Therefore, strictly speaking it is digital. By increasing the number of input bits, the number of possible output values can be increased and the difference between successive values decreased. Thus the output can be made more or less analog. For the time being we can only say that the DAC output is pseudo analog.

Input weights

Let us analyse the inputs and outputs given in Table 12.5 and consider only those digit inputs where one of the four bits is 1 and other three bits are 0. Such inputs and corresponding outputs are rewritten in Table 12.6.

Table 12.6:

A_3	A_2	A_1	A_0	V_{out}
0	0	0	1	1
0	0	1	0	2
0	1	0	0	4
1	0	0	0	8

It is clear from the entries included in Table 12.6 that the contributions of 1 are weighted according to their position in binary number. The bit **A,** has weight of 8, **A,** has weight of 4, **A,** has weight of **2** and **A,** has weight of 1. Thus the weight of the LSB is the smallest change. To check,

$$1001 = 8 + 0 + 0 + 1 = 9.$$

Example 12.3

A 5-bit DAC produces 0.5V for 00001. Find V_{out} for 11010.

Solution

In the example the smallest change is 0.5V. Therefore,

$$11010 = 16 \times \mathbf{0.5} + 8 \times 0.5 + 0 + 2 \times 0.5 + 0 = 8 + 4 + 1$$

$$= 13V.$$

Example 12.4

A 5-bit DAC produces a 10mV output for a digital input of 10100.

What will V_{out} be for a digital input of 11101?

Solution

$$\begin{aligned} 10100_2 &= 20_{10} \\ V_{out} &= k \times \text{digital input} \\ 10\,mV &= k \times 20_{10} \\ k &= 0.5\,mV \end{aligned}$$

Now V_{out} for 11101 is obtained as follows:

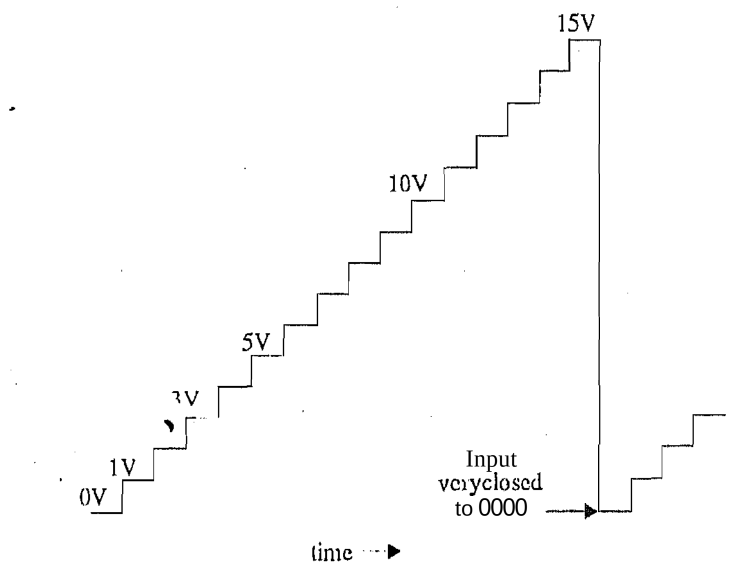
$$\begin{aligned} 11101_2 &= 29_{10} \\ V_{out} &= 0.5\,mV \times 29 \\ &= 14.5\,mV. \end{aligned}$$

SAQ 4

The smallest change in a 4-bit DAC is 0.25V. What is V_{out} for a DAC input 1110?

Resolution (Step Size)

The resolution is the smallest change that can take place in the analog output as a result of a change in the digital input. In the example of Table 12.5, the smallest change is 1V. Therefore, the resolution in that example is 1V. The resolution is also known as the step size. In the said example, the voltage rises in step of 1V and goes up from 0 to 15 in 15 steps. Pictorially, this can be represented as shown in Fig. 12.20.



Pig. 12.20: Pictorial representation of V_{out} of the example given in Table 12.5.

It can easily be seen that there are 16 levels from 0 to 15V, but there are only 15 jumps. That is the number of steps between 0 to 16 **is** 15. The number of steps in general can be calculated as

Number of steps = $2^n - 1$.

The resolution or step size is actually the constant k in equation **(12.1).** The percentage resolution is defined as

$$\% \text{ resolution} = \frac{\text{step size}}{\text{full scale } \mathbf{(FS)}} \times \mathbf{100\%} \qquad (12.2)$$

Example 14.5

What is resolution (step size) of the **DAC** of Example 12.4? Describe the staircase signal out of this **DAC.**

Solution

The LSB for this converter has n weight of **0.5** mV. This is the resolution (step size). **A** staircase waveform can be generated by connecting a 5-bit counter to the **DAC** inputs. The staircase will have 2^8 = **32** levels from **0** mV up to a full scale output (V_{out} for **input 11111** = **0.5** x **16 + 0.5** x **8 + 0.5** x 4 + **0.5** x 2 + 0.5 = 8 + 4 + 2 + 1 + 0.5 = 15.5 mV and **31** steps of **0.5** mV each.

SAQ 5

What is the percentage resolution of the DAC of Example 12.5?

DAC circuit

There are several methods and circuits for digital to analog conversion which need not be known. A basic DAC circuit is obtained using an op-amp as a summing amplifier. A 4-bit DAC circuit is shown in Fig. 12.21. The input resistors arc binary weighted, that is they are in the ratio of 1 : 2 : 4 : 8. The output voltage of this circuit is given as

$$V_{out} = -(V_{A3} + \frac{1}{2} \cdot V_{A2} + \frac{4}{4} \cdot V_{A1} + \frac{1}{8} \cdot V_{A0})$$

Negative sign indicates that it is an inverting amplificr. Note that the digital input bits can be either 0 or 1, therefore V_{A3}, V_{A2}, V_{A1}, V_{A0} will have valucs either 0 or 5V.

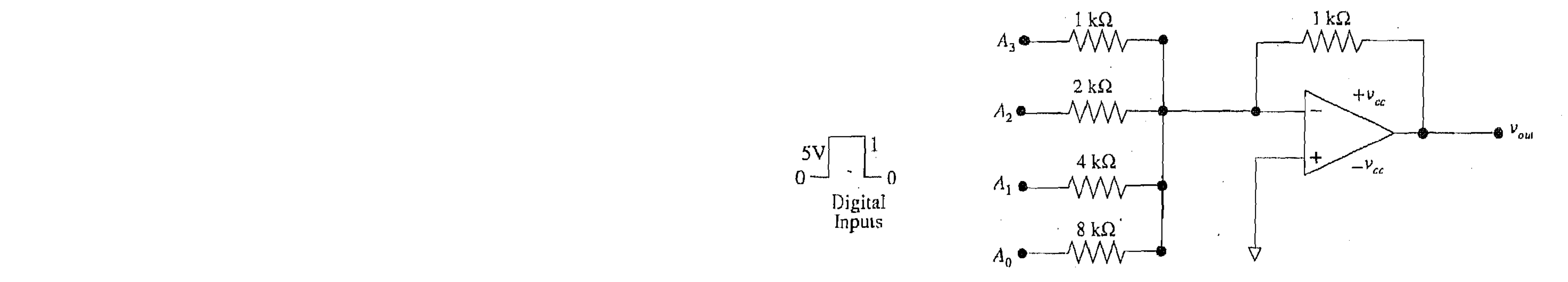


Fig. 12.21: A 4-bit DAC.

Therefore, V_{out} for 0001 or LSB would be one-eighth of 5V, i.e. 0.625V. And this is the step size of this converter. Sixteen levels of the V_{out} are shown in Tablc 12.7.

Table 12.7: Ideal valucs of V_{out} for a 4-bit DAC.

A_3	A_2	A_1	A_0	V_{out}
0	0	0	0	0
0	0	0	1	- 0.625 LSB
0	0	1	0	- 1.250
0	0	1	1	- 1.875
0	1	0	0	- 2.500
0	1	0	1	-3.125
0	1	1	0	- 3.750
0	1	1	1	- 4.375
1	0	0	0	- 5.000
1	0	0	1	- 5.625
1	0	1	0	- 6.250
1	0	1	1	- 6.875
1	1	0	0	- 7.500
1	1	0	1	- 8.125
1	1	1	0	- 8.750
1	1	1	1	- 9.375 MSB Full Scalc

These values are ideal values. However, the actual values may not be same. There may be some error due to fluctuations in the voltages or inaccurate resistors. The error in a DAC is specified by a term called full scale error which is the maximum deviation of the DAC's output from its expected ideal value expressed as the percentage of the full scale (FS). Let us say that a DAC has an error of + 0.01 % FS in the example considered above. It means that error is 0.01 % of 9.375V, i.e + 0.9375 mV.

SAQ 6

What are the weights of each input bit of Fig. 12.21.

Example 12.6

If in the DAC circuit of Fig. 12.21, R_F is reduced to half, i.e. 500Ω, then what will V_{out} be for 1001?

Solution

The MSB passes with gain 0.5. Therefore, its weight is reduced to half of the previous case. That is, it is now 2.5V. Thus each input weight is the half of the previous case, i.e. 1.25, 0,625 and 0.312V. The V_{out} for 1001 is

$$2.5 + 0 + 0 + 0.312 = 2.812V.$$

12.5.2 Analog-to-Digital Converter

The circuit of a counter type (or digital ramp) Analog-to-Digital Converter (ADC) is shown in Fig. 12.22. It consists of an op amp as a comparator, a DAC, counter and a 3-input AND gate. The functioning of this type of ADC is as follows:

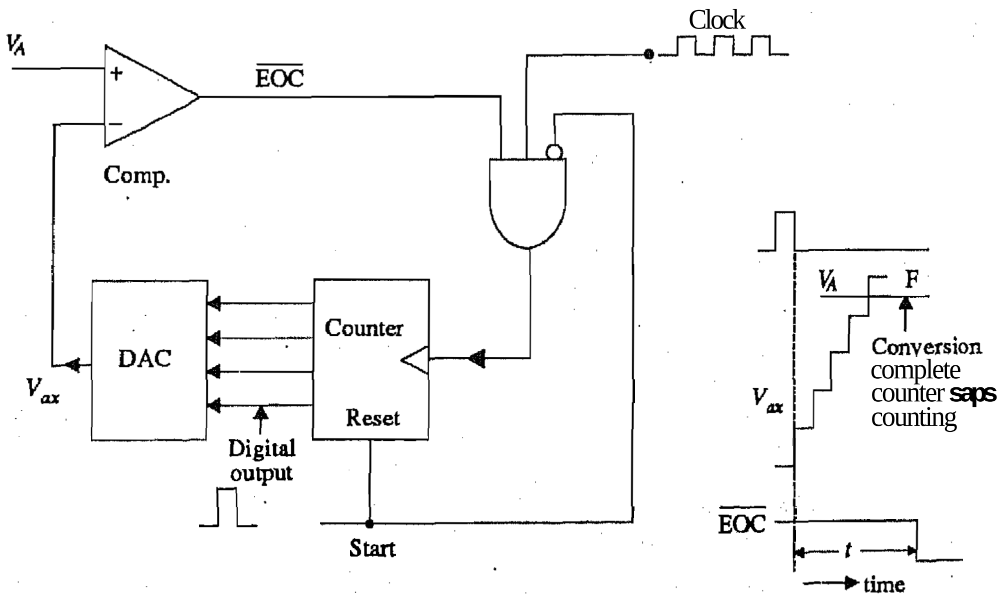


Fig. 12.22: Counter type ADC.

Apply start pulse, i.e. make START input equal to 1. This resets the counter to 0 output. With 1 at START input, the AND gate is inhibited which docs not allow the CLK from passing through the AND gate. The counter output is the input to the DAC. With counter reset, the DAC output V_{ax} = 0. V_A is the analog input to be converted into its digital equivalent. Since $V_{ax} < V_A$, the op amp comparator output EOC is HIGH, i.e. I. When the start pulse returns to 0, AND gate allows the CLK to pass through and the CLK reaches the counter which starts counting. As the counter advances, the DAC output V_{ax} advances step by step as shown in the figure. When V_{ax} reaches a step that exceeds V_A, EOC goes low. i.e. 0 disabling the AND gate. Therefore, the CLK cannot pass through and the counter stops advancing further. The conversion of analog input into its digital equivalent is complete. The contents of the counter are the digital representative of V_A. The counter holds the output until the next start pulse to initiate a **new** convcrsion is supplied.

12.6 SUMMARY

- A register is a combination of D flipflops and stores as many bits as the number of flipflops. The registcr is most commonly used memory dcvicc. The controlled register can retain the contents of the registcr as long as we want. It can shift the contents towards left (shift left register) and right (shift right registcr). The data can be supplied to and the output cnn be obtained from the registcr either serially or parallely. The registers are, sometimes, classified as serial-in serial-out, serial-in-parallel-out, parallel-in-parallel-out.

- A counter is like a register made of JK flipflops and counts the number of clock pulses arriving at CLK input of the counter. The CLK input to asynchronous counter or ripple counter is given to the first flipflop. The CLK signal to the other flipflops is the output of the preceding flipflop. Therefore, it is very slow because of high propagation delay. In synchronous counter, the CLK input is supplied to all the flipflops simultaneously. The synchronous counter is fast and there is only one propagation dclay.

- Modulus of a counter is the number of states of the output and it is 2" where n is the number of flipflops used. A 4-bit counter having four flipflops has a modulus of 16. A counter can be designed for a particular modulus. Mod 10 or decade countcr has ten states of the output, i.e. it counts from 0 to 9 (or 0000 to 1001).

- A memory is a device made up of several registers. The contents of each rcgister can be read or new contents can be stored in it. The computer or a digital system has internal memory which is used while entering data, etc. and is in constant communication with the central processing unit. Random Access Memory (RAM) is used as internal memory. It is volatile that is it requires the application of electrical power. If the electrical power is removed, then all information stored is lost. Its cost of storage per bit is very high. Read Only Memory (ROM) is used as mass storage device. Most ROMs are nonvolatile, i.e. the information stored is not lost when the electrical power is removed. The contents of this memory can only be read. However, several types of ROMs are available, The PROM can be programmed only once. The EPROM is an Erasable-PROM.

- The input to and output from a computer is digital. But the whole world is analog. Therefore, every analog signal to be processed or manipulated by a computer is first converted into a digital one using Analog-to-Digital Converter (ADC). A counter type ADC is quite often used for this purpose. The digital output of a computer is converted into an equivalent analog voltage or current using a digital-to-analog (DAC) converter. The basic circuit of an ADC is like an inverting op amp amplifier used as an adder where the input resistors are binary weighted. The resolution of a DAC can be increased by increasing the number of bits.

12.7 TERMINAL QUESTIONS.

1) Design a Mod 12 counter.

2) A computer X has memory 1M x 8 and computer Y has memory 500K × 16. What are the word sizes of the memories of the two computers? Which of the two computers can store more bits?

3) A 4-bit DAC produces an output of 7V for 1110. What is the smallest change in its output voltage? Find the output voltage for 1001.

4) What is the largest value of output voltage from an 8-bit DAC that produces 1V for a digital input of 00110010?

5) If the values of R_1s in the DAC circuit of Fig. 12.21 are reduced to half, then (a) what is the resolution, and (b) what is the output voltage for 1101?

12.8. SOLUTIONS AND ANSWERS

SAQs

1, The third flipflop divides the CLK frequency by **8.** Therefore, the output frequency will be 12.5 kHz,

2. The Mod of a counter is 2^n, where n' is the number of flipflops used. Therefore, the Mod of a counter consisting of six flipflops is $2^6 = 64$.

3 a) The size of the word is 8-bit.

b) The total number of bits that the memory stores is

$$32 \times 1024 \times 8 = 262,144 \text{ bits.}$$

4, $8 \times 0.25 + 4 \times 0.25 + 2 \times 0.25 + 0 = 2 + 1 + 0.5$

$= 3.5\text{V}.$

5. $$\% \text{ resolution } = \frac{0.5\text{mV}}{15.5\text{mV}} \times 100\% = 3.23\%.$$

6. The MSB passes with gain = 1, and so its weight is 5V. Thus,

MSB — 5V

2nd MSB — 2.5V

3rd MSB — 1.25V .

4th MSB, i.e. LSB — 0.625V.

TQs

1) A Mod 12 counter counts from 0000 to 1011. When 1100 appears the counter should reset. Therefore, a circuit is to be made which will clear the flipflops when 1100 appears. The circuit for Mod 12 counter is shown in Fig. 12.23.

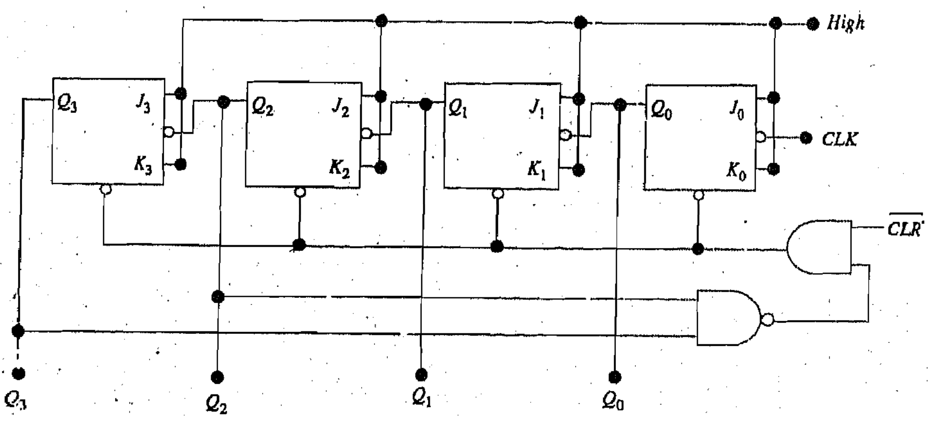


Fig. 12.23: The Mod 12 counter.

2) The word size of the computer X is 8 bits and that of computer Y is 16 bits.

For computer X

1M × 8 = 1 × 1,048,576·8 = 8,388,608 bits

For computer Y

500 × 1024 × 16 = 8, 192,000 bits.

Therefore, computer X can store more bits.

3) Follow Example 12.4. Smallest change in the output voltage is 0.5V. The output voltage for 1001 is 4.5V.

4) $00110010_2 = 50_{10}$

$1\ V = K \times 50$

Therefore. K = 20 mV.

The largest output will occur for 11111111, = 255_{10}.

$V_{out(max)} = 20\ mV \times 255$

$= 5.10V.$

5) Follow the example 12.6. The resolution is 0.312V and the output voltage for 1101 is 4.062V.