

# CHAPTER 11

# SEMICONDUCTOR MEMORIES

## 11.1 INTRODUCTION

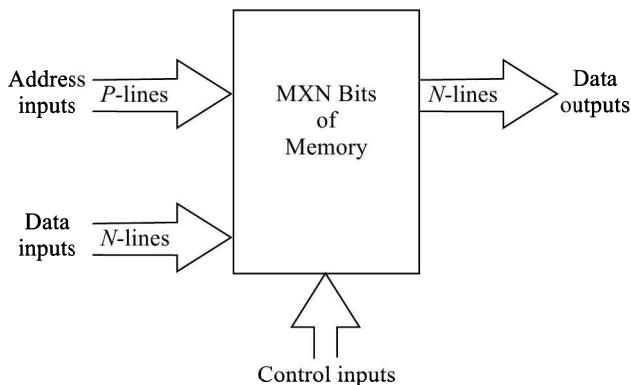
A digital processing system invariably requires a facility for storing digital information. The information usually consists of instructions (processing steps) coded in binary form, data to be processed, intermediate and final results, etc. The sub-system of a digital processing system, which provides the storage facility, is referred to as the *memory*. Till recently, the memories used were mostly of magnetic type. With unprecedented developments in semiconductor technology, it has become possible to make semiconductor memories of various types and sizes. These memories have become very popular due to their small size, low cost, high speed, high reliability, and ease of expansion of the memory size. Therefore, it is necessary for a designer of digital processors to know thoroughly the principles of operation and limitations of various semiconductor memory devices.

## 11.2 MEMORY ORGANISATION AND OPERATION

The basic element of a semiconductor memory is a which has been discussed in Chapter 7. The information is stored in binary form. There are a number of locations in a memory chip, each location being meant for one word of digital information. The number of locations and the number of bits comprising the word vary from memory to memory. The size of a memory chip is specified by two numbers  $M$  and  $N$  as  $M \times N$  bits. The number  $M$  specifies the number of locations available in the memory and  $N$  is the number of bits at each location. In other words, this means that  $M$  words of  $N$  bits each can be stored in the memory. The commonly used values of the number of words per chip are 64, 256, 512, 1024, 2048, 4096, etc. whereas the common values for the word size are 1, 4, and 8, etc. Memories requiring higher number of words and/or larger word sizes can be formed by using these chips.

The block diagram of a memory device is shown in Fig. 11.1. Each of the  $M$  locations of the memory is defined by a unique *address* and, therefore, for accessing any one of the  $M$  locations,  $P$  inputs are required, where  $2^P = M$ . This set of lines is referred to as *address inputs* or *address bus*. The address is specified in the binary form. For convenience, octal and hexadecimal representations are commonly employed.

In fact, the address input is applied to a  $P$ -to- $M$  decoder circuit, which activates one of its  $M$  outputs depending on the address and, thus, the desired memory location is selected.

Fig. 11.1 **Block Diagram of a Memory Device****Example 11.1**

Consider a memory of size 16 words. Find the binary address of each location.

**Solution**

Since  $M = 16$ , therefore,  $2^P = M$  gives  $P = 4$ , i.e. for selecting one out of 16 words, a 4-bit address is required. The address is specified as  $A_3 A_2 A_1 A_0$ , where  $A_3$  represents the most-significant bit (MSB) and  $A_0$  represents the least-significant bit (LSB) of the address. The address of each location is given in Table 11.1.

Table 11.1 **Memory Addresses for Ex. 11.1**

Word number	Binary address			
	$A_3$	$A_2$	$A_1$	$A_0$
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

The number of inputs required to store the data into or read the data from any memory location is  $N$ . One set of  $N$  lines is required for storing the data into the memory, referred to as *data inputs* and another set of  $N$  lines is required for reading the data already stored in the memory, which is referred to as *data outputs*. The concept of *bus* is used to refer to a group of conductors carrying related set of signals. Therefore, the set of lines meant for data inputs is *input data bus* and for data outputs is *output data bus*. Input and output data buses are unidirectional, i.e. the data can flow in one direction only. In most of the memory chips available, the same set of lines is used for data input as well as data output and is referred to as *bidirectional bus*. This means that the data bus is time multiplexed. It is used as input bus for some specific time and as output bus for some other time depending upon a Read/Write control input as shown in Fig. 11.2.

A number of control inputs are required to give commands to the device to perform the desired operation. For example, a command signal is required to tell the memory whether a read or a write ( $R/W$  is Fig. 11.2) operation is desired.

When  $R/\bar{W}$  is HIGH, the data bus will be used for reading the memory (output bus) whereas when  $R/\bar{W}$  is LOW, the bus will be acting in the input direction and the data on the bus will go into the memory. Other command include inputs chip enable ( $CE$ ), chip select ( $CS$ ), etc.

In addition to the above-mentioned functional pins, a minimum of two pins are required for power supply and ground. The internal organization of a  $16 \times 4$  memory chip is illustrated in Fig. 11.3. The write and read operations are discussed below.

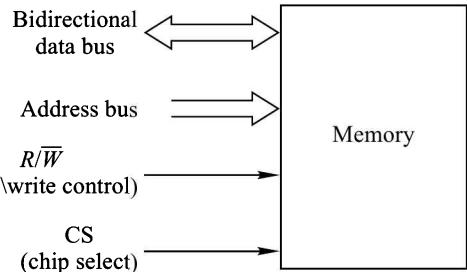


Fig. 11.2 **Block Diagram of Memory with Bidirectional Data Bus**

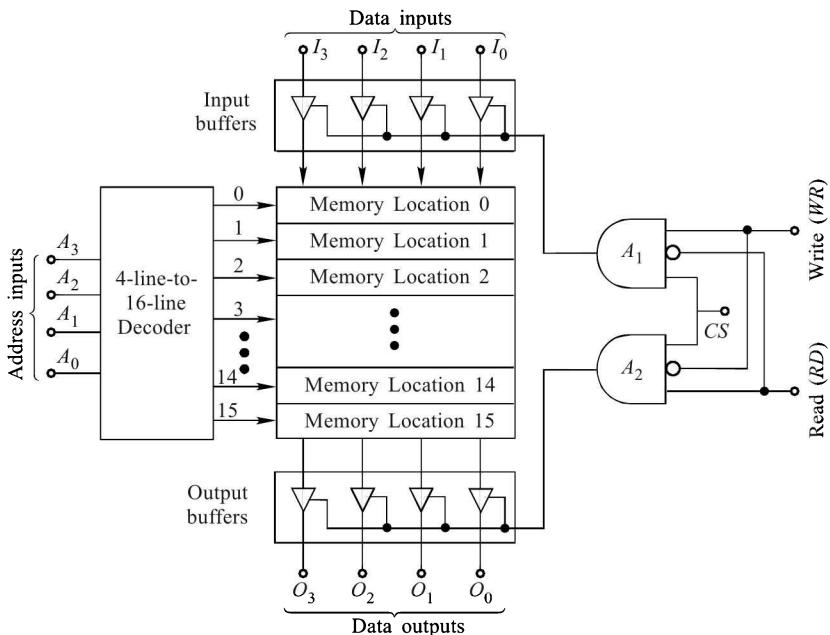


Fig. 11.3 **Internal Organization of a  $16 \times 4$  Memory Chip**

### 11.2.1 Write Operation

To write a word into the selected memory location requires logic 1 voltage to be applied to *CS* (Chip select), and Write (*WR*) inputs and logic 0 voltage to Read (*RD*) input. This combination of inputs gives outputs of AND gates  $A_1$  and  $A_2$  1 and 0 respectively. A 1 at the output of  $A_1$  enables the input buffers so that the 4-bit word applied to the data inputs will be loaded (entered) into the selected (addressed) memory location. A 0 at the output of  $A_2$  disables (tristate) the output buffers so that the data outputs are not available. The outputs are in the Hi-Z state.

For writing a word into a particular memory location, following sequence of operations is to be performed:

1. The chip select signal is applied to the *CS* terminal.
2. The word to be stored is applied to the data-input terminals.
3. The address of the desired memory location is applied to the address-input terminals.
4. A write command signal is applied to the write-control input terminal with *RD* = 0.

In response to the above operations, the addressed memory location is cleared of any word that might have been stored in it earlier, and the information presented at the data input terminal replaces it.

Figure 11.4 illustrates the various waveforms during the write operation. The important timing characteristics of the write cycle are:

#### **Write Cycle Time ( $t_{WC}$ )**

This is the minimum amount of time for which the valid address must be present for writing a word in the memory. In other words, it is the minimum time required between successive write operations.

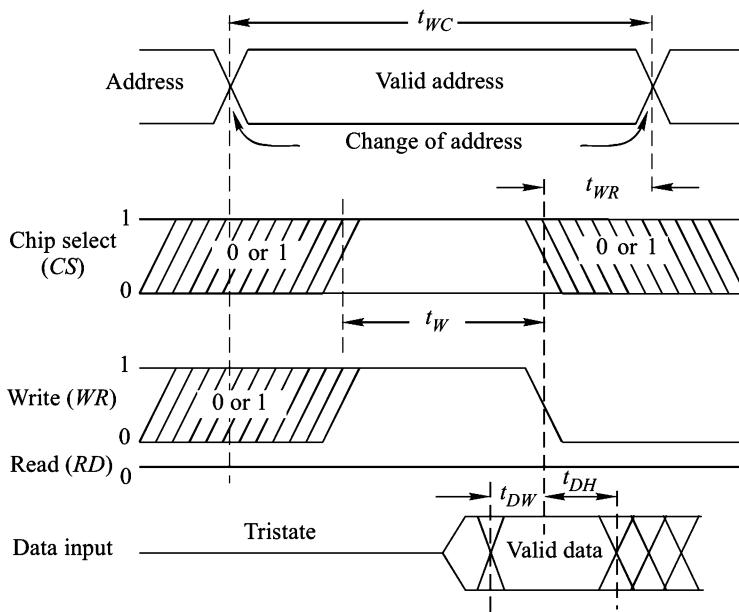


Fig. 11.4      **Write-Cycle Waveforms**

### **Write Pulse Time ( $t_w$ )**

This is the minimum length of the write pulse.

### **Write Release Time ( $t_{WR}$ )**

This is the minimum amount of time for which the address must be valid after the write pulse ends.

### **Data Set Up Time ( $t_{DW}$ )**

This is the minimum amount of time for which the data must be valid before the write pulse ends.

### **Data Hold Time ( $t_{DH}$ )**

This is the minimum amount of time for which the data must be valid after the write pulse ends.

## **11.2.2 Read Operation**

In order to read the contents of a selected memory location, the Read ( $RD$ ), and the Chip select ( $CS$ ) inputs must be at logic 1 level and  $WR$  at logic 0 level. This gives output of  $A_2 = 1$  which enables the output buffers so that the contents of the selected (addressed) memory location will appear at the data outputs.  $RD = 1$  tristates the input buffers so that the data inputs do not affect the memory during a read operation.

To read (or retrieve) a data word, known to be stored at a particular address, the following sequence of operations is required to be performed:

1. The chip-select signal is applied to the  $CS$  terminal.
2. The address of the desired memory location is applied to the address-input terminals.
3. A read-command signal is applied to the read control-input terminal.

In response to the above operations, the data word stored at the addressed location appears on the data-output terminals.

Figure 11.5 illustrates the various waveforms during the read operation. The important timing characteristics of the read cycle are:

### **Read Cycle Time ( $t_{RC}$ )**

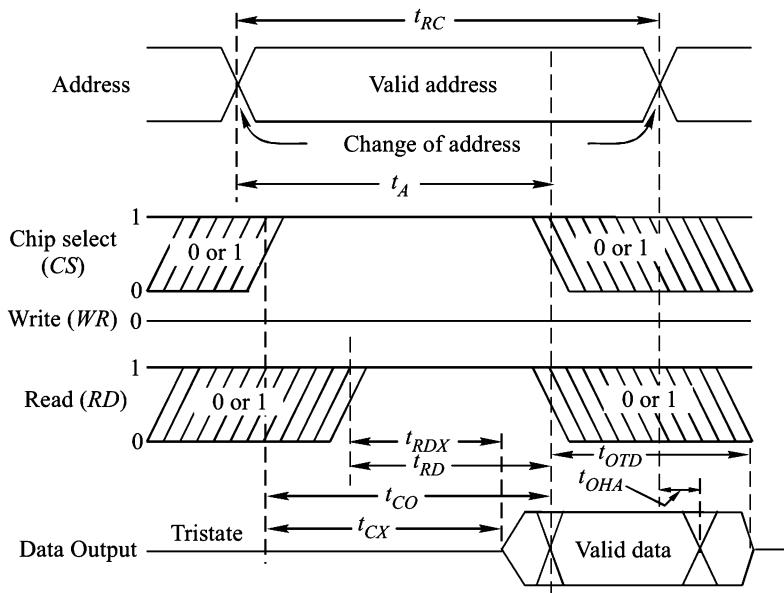
This is the minimum amount of time for which the valid address must be present for reading a word from the memory. In other words, it is the minimum time required between successive read operations.

### **Access Time ( $t_A$ )**

This is the maximum time from the start of the valid address of the read cycle to the time when the valid data is available at the data outputs. The access time is at most equal to the read-cycle time, i.e.  $t_A \leq t_{RC}$ . In other words, the data outputs might be ready before the memory is actually ready for the next read operation.

### **Read To Output Valid Time ( $t_{RD}$ )**

This is the maximum time delay between the beginning of the read pulse and the availability of valid data at the data outputs.

Fig. 11.5 *Read-Cycle Waveforms*

### **Read To Output Active Time ( $t_{RDX}$ )**

This is the minimum time delay between the beginning of the read pulse and the output buffers coming to active state (from the high-impedance state).

### **Chip-Select To Output Valid Time ( $t_{co}$ )**

This is the maximum time delay between the beginning of the chip-select pulse and availability of valid data at the data outputs.

### **Chip-Select To Output Active Time ( $t_{cx}$ )**

This is the minimum time delay between the beginning of the chip-select pulse and the output buffers coming to active state.

### **Output Tristate From Read ( $t_{OTD}$ )**

This is the maximum time delay between the end of the read pulse and the output buffers going to high-impedance state.

### **Data Hold Time ( $t_{OHA}$ )**

This is the minimum time for which the valid data is available at the data outputs after the address ends.

The write- and read-cycle timings of a typical memory chip are given in Table 11.2.

Table 11.2 *Timing Parameters of a Typical Memory Chip*

Parameter	Time (ns)
$t_{WC}$	200
$t_w$	120
$t_{WR}$	0
$t_{DW}$	120
$t_{DH}$	0
$t_{RC}$	200
$t_A$	200
$t_{RD}$	70
$t_{RDX}$	20
$t_{CO}$	70
$t_{CX}$	20
$t_{OTD}$	60
$t_{OHA}$	50

### Example 11.2

For the memory timing of Table 11.2, find the maximum rate (words/second) at which

- (a) Data can be stored, and
- (b) Data can be read.

#### Solution

- (a) The maximum rate at which data can be stored is

$$\frac{1}{t_{WC}} = \frac{1}{200 \times 10^{-9}} = 5 \times 10^6 \text{ words/s}$$

- (b) The maximum rate at which data can be read is

$$\frac{1}{t_{RC}} = \frac{1}{200 \times 10^{-9}} = 5 \times 10^6 \text{ words/s}$$

## 11.3 EXPANDING MEMORY SIZE

In many memory applications, the required memory capacity, i.e. the number of words and/or word size, cannot be satisfied by a single available memory IC chip. Therefore, several similar chips have to be combined suitably to provide the desired number of words and/or word size.

### 11.3.1 Expanding Word Size

If it is required to have a memory of word size  $n$  and the word size of the available memory ICs is  $N$  ( $n > N$ ), then a number of similar ICs can be combined together to achieve the desired word size. The number of IC

chips required is an integer, next higher to the value  $n/N$ . These chips are to be connected in the following way:

1. Connect the corresponding address lines of each chip individually, i.e.  $A_0$  of each chip is connected together and it becomes  $A_0$  of the overall memory. Similarly, connect other address lines together.
2. Connect the  $RD$  input of each IC together and it becomes the read input for the overall memory. Similarly, connect the  $WR$  and  $CS$  inputs.

Now, the number of data-input/output lines will be equal to the product of the number of chips used and the word size of each chip. The following example illustrates the above procedure clearly.

### Example 11.3

Obtain a  $16 \times 8$  memory using  $16 \times 4$  memory ICs.

#### Solution

Since the word size required is  $n = 8$  and the word size of the available IC is  $N = 4$ , therefore,  $n/N = 2$  chips are required to obtain the desired memory.

Since each chip can store 16 4-bit words and we want to store 16 8-bit words, each chip is required to store half of each word. Figure 11.6 shows the relevant connections of the two chips. Here, we have assumed bidirectional input/output ( $I/O$ ) lines which is common in available memory chips. In this  $16 \times 8$  memory, the higher order four bits ( $D_7, D_6, D_5, D_4$ ) of each 8-bit word are located in memory  $M_1$  and the lower order four bits ( $D_3, D_2, D_1, D_0$ ) are located in memory  $M_0$ .

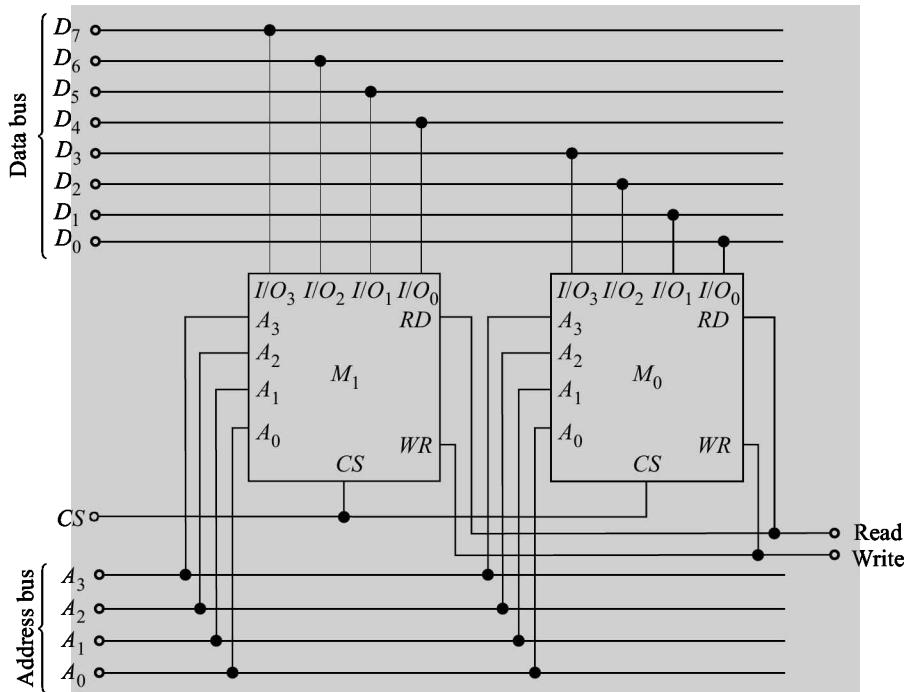


Fig. 11.6 A  $16 \times 8$  Memory Obtained by Combining Two  $16 \times 4$  Memory Chips

### 11.3.2 Expanding Word Capacity

Memory chips can be combined together to produce a memory, with the desired number of locations. To obtain a memory of capacity  $m$  words, using the memory chips with  $M$  words each, the number of chips required is an integer next higher to the value  $m/M$ . These chips are to be connected in the following way:

1. Connect the corresponding address lines of each chip individually.
2. Connect the RD input of each chip together. Similarly, connect the WR inputs.
3. Use a decoder of proper size and connect each of its outputs to one of the CS terminals of memory chips. For example, if eight chips are to be connected, a 3-line-to-8-line decoder is required to select one out of eight chips at any one time.

The following example clearly illustrates the above procedure.

#### Example 11.4

Obtain a  $2048 \times 8$  memory using  $256 \times 8$  memory chips.

#### Solution

The number of chips required is  $\frac{2048}{256} = 8$ . At any one time, only one of the 2048 locations is to be accessed, which

will be in one of the eight chips. That means only one of the eight chips must get selected at a time.

For selecting one out of 2048 locations, the number of address lines required is 11 ( $2^{11} = 2048$ ). The lower order eight bits of the address  $A_7 - A_0$  will be same for each chip, and the higher order three bits of the address  $A_{10} - A_8$  must select one out of the eight chips. For this purpose, a 3-line-to-8-line decoder is required. The memory connections are shown in Fig. 11.7. Here, we have assumed a common terminal ( $R/\bar{W}$ ) for read and write. For the read operation, logic 1 is to be applied to  $R/\bar{W}$ , whereas logic 0 is to be applied for the write operation. The chip-select input is assumed to be active-low. The addresses of the chips are given in Table 11.3.

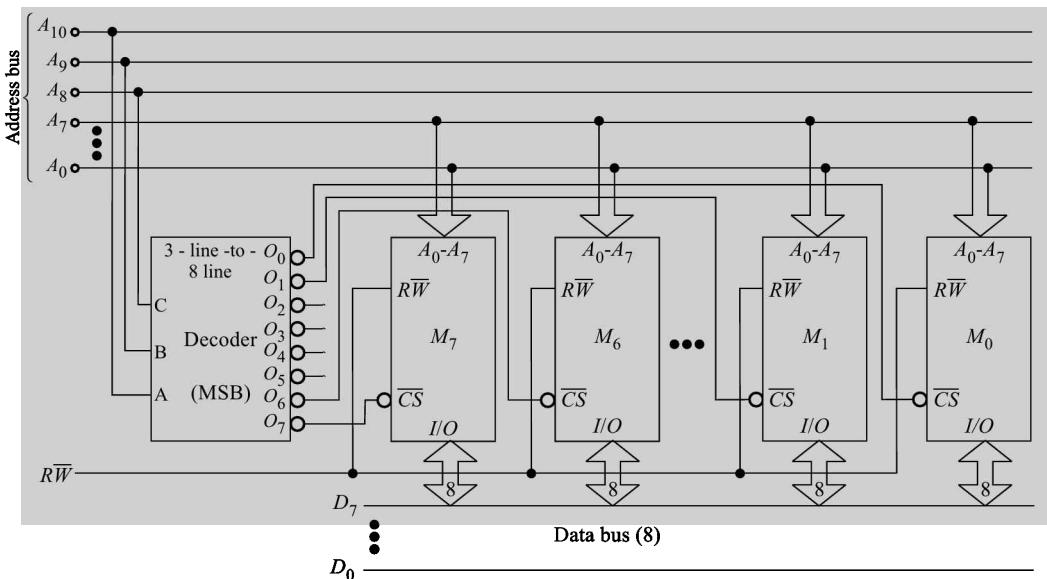


Fig. 11.7 A  $2048 \times 8$  Memory Obtained by Combining Eight  $256 \times 8$  Memory Chips

Table 11.3 *Addresses of the Memory Chips*

Memory chip	Addresses (hex.)
$M_0$	000 – 0FF
$M_1$	100 – 1FF
$M_2$	200 – 2FF
$M_3$	300 – 3FF
$M_4$	400 – 4FF
$M_5$	500 – 5FF
$M_6$	600 – 6FF
$M_7$	700 – 7FF

## 11.4 CLASSIFICATION AND CHARACTERISTICS OF MEMORIES

Various memory devices can be classified on the basis of their principle of operation, physical characteristics, mode of access, technology used for fabrication, etc.

### 11.4.1 Principle of Operation

Memories can be classified according to their principle of operation. The most commonly used memories are:

- Read-only memories (ROM)
- Read-and Write memories (RWM or RAM)
- Flash memories
- Content-addressable memories (CAM)
- First-in, First-out (FIFO) memories

#### ***Read-Only Memories (ROM)***

*Read-only-memory (ROM)*, as the name suggests, is meant only for reading the information from it. This does not mean that information is not written into it, because unless some information is stored into it, there cannot be anything to read from. Actually, the process of entering information into this type of memory is much more complicated than for RAM and is done outside the system where it is used. Therefore, it is called as read-only memory. It is used to store information which is fixed, such as tables for various functions, fixed data and instructions. The ROMs are also organised so that every memory location requires equal time for reading the data already stored in that location.

The various types of read-only memories are further sub-classified on the basis of the technique employed for storing information into the memory (referred to as *programming*) or their erasability properties. These are:

#### ***Read-Only Memory***

It is programmed at the time of manufacturing, as the last process of fabrication, according to the information specified by the user. It is referred to as *custom programmed* or mask programmed. The data stored can not

be changed after fabrication. Since, this process is quite costly, therefore, this type of ROM is suitable only for bulk requirements, of the order of millions of chips.

### *Programmable Read-Only Memory (PROM)*

This type of ROM is programmed by the user using a special circuit—a PROM programmer. A PROM can be programmed only once after which its contents are permanently fixed as in a ROM. This type of ROM is suitable for storage of data which is of permanent nature. The chip is available without any data stored from the vendor.

### *Erasable and Programmable ROM*

This type of ROMs are reprogrammable i.e. it can be programmed again and again. It is referred to as *erasable and programmable* ROM.

There are two techniques used for erasing; in one technique the chip is exposed to ultraviolet radiation, and in the other technique the contents are altered electrically. The erasable programmable ROM using ultraviolet radiation for erasing is known as EPROM, whereas the device using electrical voltage for erasing is known as *electrically alterable* ROM (EAROM) or (EEPROM).

### *Electrically Erasable and Programmable (EEPROM)*

The detailed operation of various types of ROMs is discussed in Section 11.5.

### *Random Access Memories (RWM or RAM)*

In this type of memories, the memory locations are organised in such a way so that any memory location requires equal time for writing or reading. This type of memory is also known as read-and-write memory (RWM) or RAM. RAMs can be static or dynamic and are fabricated using bipolar or unipolar technologies. The static RAM (SRAM) generally uses bistable latch as storage element, whereas the *dynamic* RAM (DRAM) uses capacitor as storage element which requires periodic refreshing. The SRAMs are faster than the DRAMs. However, DRAMs can store much more data than SRAMs for a given physical size and cost. The SRAMs can be fabricated by using bipolar devices or MOSFETs, but the DRAMs can only be made using MOSFETs.

The SRAMs can be *asynchronous* SRAM or *synchronous* SRAM. An asynchronous SRAM is one in which the operation is not synchronised with a system clock. The operation a synchronous SRAM is synchronised with the system clock. The synchronous SRAMs normally have a *burst* feature. The burst feature allows the memory to read or write at upto four locations using a single address. The two lowest order address bits  $A_1$  and  $A_0$  are applied to the burst logic circuit which contains a Mod-4 counter. This produces a sequence of four internal addresses by using the two lowest order bits of the address as 00, 01, 10, and 11 on successive clock pulses. The sequence always begins with the base address, which is the external address applied.

The types of DRAMs are:

- Fast Page Mode DRAM (FPM DRAM)
- Extended Data Out DRAM (EDO DRAM)
- Burst EDO DRAM (BEDO DRAM)
- Synchronous DRAM (SDRAM)

The first three types of DRAMs are asynchronous DRAMs.

The detailed operations of bipolar SRAM, and MOSFET SRAM and DRAM are discussed in Section 11.6.

## Flash Memories

Flash memories are non-volatile, large bit storage capacity, read and write memories. The storage cell in a flash memory consists of a single floating-gate MOS transistors. Its operation is explained in Section 11.7.

## Content Addressable Memories (CAM)

Is a special purpose random access memory which performs association operation in addition to read/write operations. The detailed operation of the CAM is discussed in Section 11.8.

## First-in, First-out Memories (FIFO)

In this type of memory, the data which is entered first is taken out first. The storage device used in this is a SRAM array with two separate ports that allows data to be written into and read from its array at independent data rates. The detailed operation of FIFO memory is discussed in Section 11.9.

### 11.4.2 Physical Characteristics

Memories can be classified according to their physical characteristics, such as:

1. Erasable or non-erasable, and
2. Volatile or non-volatile.

## Erasable or Non-Erasable Memories

A memory in which the information stored can be erased and new information stored is called *erasable* memory. On the other hand, the information stored in the *non-erasable* memory cannot be erased, for example ROM is non-erasable.

The erase operation is performed in the following ways:

- Electrically
- By exposing the chip to ultraviolet (UV) radiation. The EEPROMs are electrically erasable whereas the EPROMs are erased by exposing the chip to UV radiation.

## Volatile or Non-Volatile Memories

If the information stored in a memory is lost when electrical power is switched off, the memory is referred to as a *volatile memory*. For example, the RAM is a volatile memory. On the other hand, in a *non-volatile memory*, the information once stored remains intact until changed deliberately. All types of ROMs are non-volatile memories.

### 11.4.3 Mode of Access

Mode of access refers to the manner in which a memory location is accessed for reading or writing. In case of ROMs only reading is possible. There are two modes of access. These are:

- Sequential access, and
- Random access.

*Sequential memories* are referred to as sequentially accessed memories, whereas RAM, ROM, and CAM are random-access memories. In random-access memories any memory location requires equal time for accessing (*access time*) whereas the access time is different for different locations in the case of sequentially accessed memory.

#### 11.4.4 Fabrication Technology

Memories can be classified on the basis of the fabrication technology used. The two broad categories of memories based on fabrication technology used are:

1. Bipolar, and
2. Unipolar (MOS).

These technologies have been discussed in detail in Chapter 4. Static RAM, ROM and PROM can be fabricated using either bipolar technology (TTL, ECL, etc.) or MOS technology, whereas dynamic RAM, EPROM, EEPROM and flash memories can be fabricated using only unipolar devices (MOSFETs).

### 11.5 READ-ONLY MEMORY

A read-only memory (ROM) is a semiconductor memory device used to store information which is permanent in nature. It has become an important part of many digital systems because of its low cost, high speed, system-design flexibility and data non-volatility. The read-only memory has a variety of applications in digital systems, such as implementation of combinational logic and sequential logic, character generation, look-up table, microprocessor programme storage, etc.

ROMs are well-suited for LSI manufacturing processes and are available in many forms. Two major semiconductor technologies are used for the manufacturing of ROM integrated circuits, viz. bipolar technology and MOS technology, which differ primarily in access time. In general, bipolar devices are faster and have higher drive capability, whereas MOS devices require less silicon area and consume less power. With improvements in MOS technology, it is now possible to make MOS memories with speeds comparable to those of bipolar memories.

The process of entering information into a ROM is referred to as *programming* the ROM. Depending on the programming process employed, the ROMs are categorized as:

1. *Mask programmable read-only memories*, which are referred to as ROMs. In these memories, the data pattern must be programmed as part of the fabrication process. Once programmed, the data pattern can never be changed. These are highly suited for very high volume usage due to their low cost.
2. *Programmable read-only memories*, which are referred to as PROMs. A PROM is electrically programmable, i.e. the data pattern is defined after final packaging rather than when the device is fabricated. The programming is done with an equipment referred to as *PROM programmer*. The programming techniques used will be discussed later.
3. *Erasable programmable read-only memories*, which are referred to as EPROMs. As the name suggests, in these memories, data can be written any number of times, i.e. they are reprogrammable. Reprogrammable ROMs are possible only in MOS technology. For erasing the contents of the memory, one of the following two methods are employed:
  - (a) Exposing the chip to ultraviolet radiation for about 30 minutes.
  - (b) Erasing electrically by applying voltage of proper polarity and amplitude. Electrically erasable PROM is also referred to as E<sup>2</sup>PROM or EAROM (Electrically alterable ROM).

### 11.5.1 ROM Organisation

A read-only memory is an array of selectively open and closed unidirectional contacts. The address decoder of Fig. 11.3 is usually divided in two parts for simplifying the decoder design. One half of the address lines are decoded by one decoder used to energize one of the row lines, whereas the other half of the address lines are decoded by another decoder used to activate column lines. This method of addressing is referred to as two-dimensional,  $X-Y$ , or coincident-selection, addressing. A unidirectional switch is incorporated at the junction of every row and column.

A 16-bit ROM array is shown in Fig. 11.8. To select any one of the 16 bits, a 4-bit address ( $A_3, A_2, A_1, A_0$ ) is required. The lower order two bits ( $A_1, A_0$ ) are decoded by the decoder  $D_L$  which selects one of the four rows, whereas the higher order two bits ( $A_3, A_2$ ) are decoded by the decoder  $D_H$  which activates one of the four-column sense amplifiers.

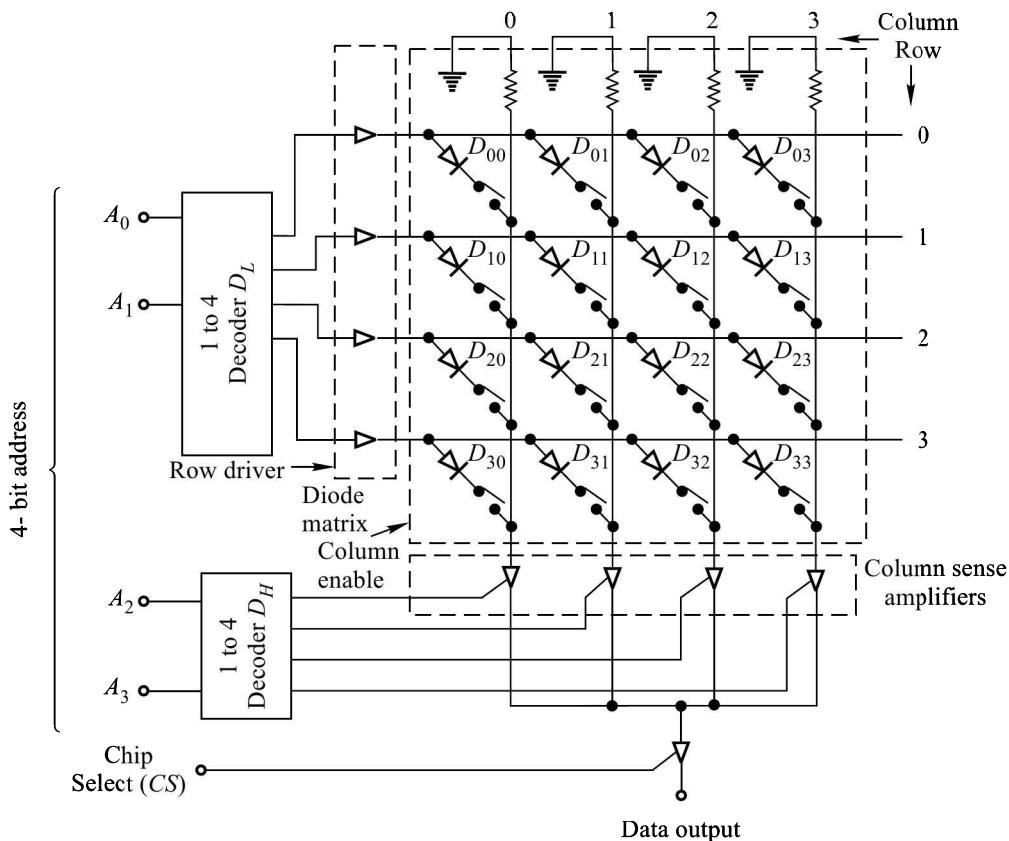


Fig. 11.8 16-bit ROM Array

The diode matrix is formed by connecting one diode, along with a switch between each row and column. For example, the diode  $D_{21}$  is connected between row 2 and column 1.

The output is enabled by applying logic 1 at the chip select ( $CS$ ) input.

Programming a ROM means to selectively open and close the switches in series with the diodes. For example, if the switch of the diode  $D_{21}$  is in closed position and if the address input is 0110, row 2 is activated connecting it to column 1. Also, the sense amplifier of column 1 is enabled which gives logic 1 output if the chip is selected ( $CS = 1$ ). This shows that a logic 1 is stored at the address 0110. On the other hand, if the switch of diode  $D_{21}$  is open, logic 0 is stored at the address 0110.

### Example 11.5

In the 16-bit ROM of Fig. 11.8, the switches of diodes  $D_{00}$ ,  $D_{03}$ ,  $D_{12}$ ,  $D_{13}$ ,  $D_{21}$ , and  $D_{33}$  are programmed as closed. Find the bit stored in each location.

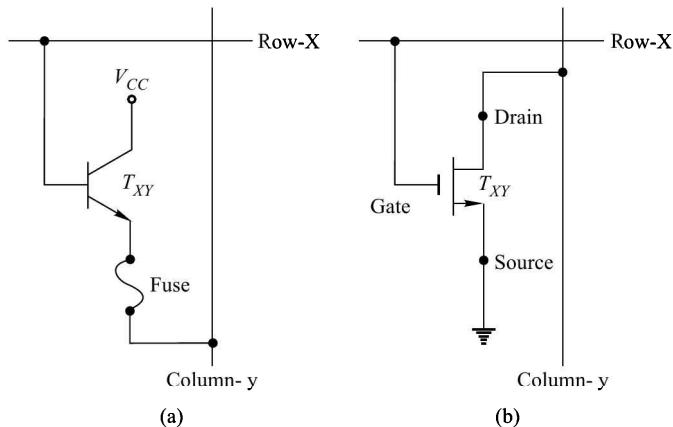
#### Solution

The bit stored in each location are given in Table 11.4.

Table 11.4 Bit Pattern of Ex. 11.5

Address				Bit stored
$A_3$	$A_2$	$A_1$	$A_0$	
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

The ROMs can also be implemented by using bipolar junction transistors or MOSFETs, instead of diodes. Each transistor is to be connected as shown in Fig. 11.9a for bipolar memories and as shown in Fig. 11.9b, for MOS memories. Each column line is to be connected to ground through a resistance for bipolar devices, whereas it is to be connected to the  $V_{DD}$  supply through a load MOSFET for MOS devices. The fuse used in the circuit of Fig. 11.9a is meant for programming, and is either retained intact or blown off depending upon whether 1 or 0 is to be stored in a selected location.

Fig. 11.9     **Unidirectional Switch (a) Bipolar (b) MOS**

### 11.5.2 Programming Mechanisms

#### *Mask Programmable ROMs*

Integrated circuits are fabricated through a number of processing steps, such as photomasking, etching, diffusion, etc. One of the final steps in the manufacturing process is to deposit a layer of aluminium on the entire surface of the silicon wafer. The desired interconnecting pattern is produced by selectively etching away portions of aluminium. The row-to-column contacts can be, retained or etched away, as desired, in the final aluminium etching process, in the manufacture of mask-programmed read-only memories.

#### *Programmable ROMs*

The most commonly used programming mechanisms for the bipolar memory devices are:

1. Fusible link process, and
2. Avalanche-induced migration (AIM) process.

In the fusible link process, a fuse is added in the emitter lead of every transistor (Fig. 11.9a). Usually, the fuse material used is either nichrome or polycrystalline silicon. At the time of the fabrication of these memory devices, all the fuse links are in place and can be selectively open circuited after packaging.

When nichrome is used as the fuse material, it is deposited as a very thin layer, which can be blown off (opening the connection between the row and column lines) by making a large current (20–50 mA) to flow through it. The fusing time varies between 5 and 200  $\mu$ s. Typically, the programming rate is 5 ms per bit. It has power dissipation of about 170  $\mu$ W per bit while operating from a 5 V supply. The nichrome fuse elements are to be left intact for storing 1s and blown off for storing 0s.

Another fuse material used is polycrystalline silicon. It is deposited as a thick layer ( $\sim 3000\text{\AA}$ ) during the manufacturing process. The fuse is blown with a pulse train of successively wider pulses. Typically, a current of 20–30 mA is needed to blow the fuse. Temperatures of the order of 1400°C are reached during the blowing process. The silicon gets oxidized and forms an insulating material.

From Fig. 11.9a, we observe the following: When a row is selected (say  $X$ ), the transistor  $T_{xy}$  is turned ON. If the fuse is intact, the column line  $Y$  is pulled towards  $V_{CC}$  (logic 1) whereas if the fuse is blown (open), the column bus is left floating (logic 0).

Another mechanism used for programming PROMs utilizes an avalanche approach. Figure 11.10 shows the arrangement used. In this, the diode  $D_1$  is reverse-biased and the heavy flow of electrons in the reverse direction causes aluminium atoms from the emitter contact to migrate through the emitter to the base. This causes an emitter-to-base short. This process requires higher currents (200–300 mA) and voltages than the fusible link, but is faster, requiring 0.02–0.05 ms time.

The above mechanisms are irreversible, i.e. a device can be programmed only once in its life time. These mechanisms do not work with MOS memory devices, where resistance and current levels required for the fusing process are incompatible with MOS impedance levels. Commonly used MOS technologies for the fabrication of programmable memories are:

1. Floating gate avalanche injection MOS (FAMOS), and
2. MAOS.

### FAMOS PROM

In this, the storage device used is silicon gate MOSFET with no electrical connection to the gate, i.e. the gate is electrically floating in an insulating layer of silicon dioxide, as shown in Fig. 11.11a. The symbol of FAMOS is shown in Fig. 11.11b.

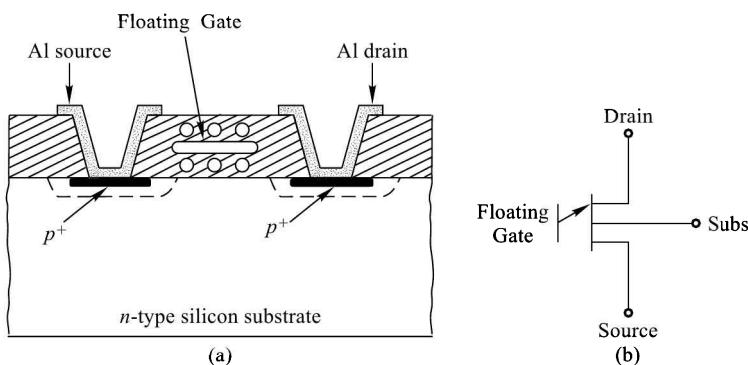


Fig. 11.11 *p-channel FAMOS Device: (a) Cross-Sectional View, and (b) Symbol*

The operation of this device depends on the charge transport to the floating gate by avalanche injection of electrons from either the source or drain, caused by the application of high voltage (25–30 V) across the transistor. The amount of charge transferred to the floating gate is a function of the amplitude and duration of the applied voltage. The presence or absence of charge can be sensed by measuring the conductance between the source and the drain.

When the applied voltage is removed, the charge remains trapped in the gate, since no discharge path is available for the accumulated electrons because the gate is surrounded by a very low conductivity dielectric. The transistor now behaves as if an external voltage were permanently connected to the gate terminal.

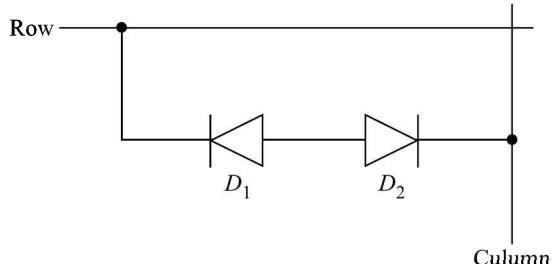


Fig. 11.10 *A Shorted Junction Cell*

The charge accumulated on the gate can be removed by illuminating the FAMOS device with ultraviolet light. This results in the flow of a photo current from the floating gate back to the silicon substrate, thereby discharging the gate to its initial condition.

These devices are packaged with a transparent quartz lid for exposing the device to ultraviolet radiation, for the purpose of erasing. Since erasing is possible in these devices, these are reprogrammable and are known as *erasable programmable read-only memory* (EPROM) devices. In case the device is packaged in inexpensive package without quartz lid, it works as a *one-time programmable* (OTP) ROM which is same as a PROM.

The use of second metal gate (*erase gate*) permits reprogramming without the use of ultraviolet radiation. Figure 11.12 shows a cross-sectional view of a *p*-channel device with an erase gate. The programming is done by applying a negative voltage to both the source ( $V_{SX}$ ) and the drain ( $V_{DX}$ ), for inducing avalanche breakdown at both junctions. Simultaneously, a positive voltage applied to the second gate  $G_2$  ( $V_{G2S}$ ) increases the rate at which electrons accumulate on the floating gate. As electrons accumulate on the floating gate, the channel appears between the source and the drain and the device turns ON.

For the purpose of erasing, again a negative voltage is applied to both the source and drain junctions, for inducing avalanche breakdown similar to programming mode, but a negative voltage is applied to the gate  $G_2$ . This results in the accumulation of holes on the floating gate, which neutralizes the existing charge. Thus, the erasing is done electrically rather than with ultraviolet radiation.

These devices are referred to as *electrically alterable ROMs* (EAROMs) or *electrically erasable and programmable ROMs* (EEPROMs or E<sup>2</sup>PROMs), and are reprogrammable.

## MAOS PROM

Another PROM uses the gate dielectric, such as alumina ( $\text{Al}_2\text{O}_3$ ) and silicon nitride itself, for charge storage and provide a reprogramming feature. A cross-sectional view of such a device using alumina dielectric is shown in Fig. 11.13. This device is referred to as MAOS memory element.

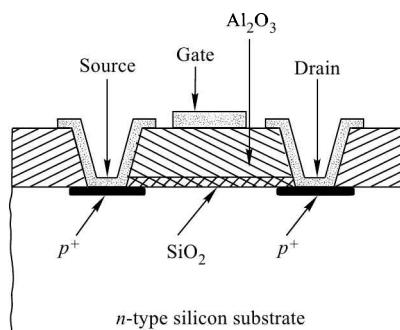


Fig. 11.13    *Cross-Sectional View of a MAOS Device*

For a *p*-channel device, a positive gate voltage of about 50 V amplitude is required for 10–20  $\mu\text{s}$ , for programming. Erasing requires a voltage of opposite polarity on the gate.

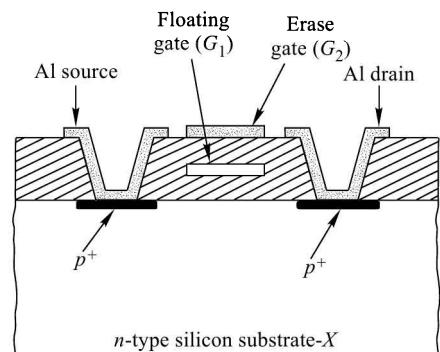


Fig. 11.12    *Cross-Sectional View of a FAMOS Device with an Erase Gate*

### 11.5.3 ROM ICs

Various types of programmable/erasable ROM ICs are commercially available. The programmable ROM devices (PROM) are available in bipolar technology in which erasing is not possible. In CMOS technology, PROM devices are available which are actually EPROM devices with the provision of only one time programming. These EPROM devices are without quartz lid and therefore, erasure is not possible in these devices. These EPROMs are known as one time programmable (OTP EPROM). The other type of EPROM with quartz lid are used for multiple programming. For reprogramming, the already stored data is erased by exposing the chip to ultraviolet light. The electrically erasable and programmable read-only memory (EEPROM) devices are available as *parallel EEPROM* and *serial EEPROM*. Some of the available ICs are given in Table 11.5. A number of low-voltage CMOS (LVC MOS) devices are also available which require lower than 5 V power supply.

Some of these devices are briefly described here. However, their complete technical and operational details can be obtained from the websites of the manufacturers/vendors.

Table 11.5 *Available ROM ICs*

IC No.	Organization No. of bits	Output*	Power supply voltage	Technology	Types of ROM
74S188	32 × 8	O.C	5 V	Schottky TTL	PROM
74S288	32 × 8	TS	5 V	Schottky TTL	PROM
74S571	512 × 4	TS	5 V	Schottky TTL	PROM
27C010	128 K × 8	TS	5 V	CMOS EPROM	OTP EPROM
27C020	256 K × 8	TS	5 V	CMOS EPROM	OTP EPROM
27C040	512 K × 8	TS	5 V	CMOS EPROM	OTP EPROM
27C080	1024 K × 8	TS	5 V	CMOS EPROM	OTP EPROM
27C1024	64 K × 16	TS	5 V	CMOS EPROM	OTP EPROM
27C64	8 K × 8	TS	5 V	CMOS EPROM	EPROM/OTP EPROM
27C128	16 K × 8	TS	5 V	CMOS EPROM	EPROM/OTP EPROM
27C256	32 K × 8	TS	5 V	CMOS EPROM	EPROM/OTP EPROM
27C512	64 K × 8	TS	5 V	CMOS EPROM	EPROM/OTP EPROM
28C64	8 K × 8	TS	5 V	NV CMOS	Parallel EEPROM
28C256	32 K × 8	TS	5 V	NV CMOS	Parallel EEPROM
28C010	128 K × 8	TS	5 V	NV CMOS	Parallel EEPROM
28C040	512 K × 8	TS	5 V	NV CMOS	Parallel EEPROM
24C01	128 × 8	O.D	5 V	NV CMOS	Serial EEPROM
24C02	256 × 8	O.D	5 V	NV CMOS	Serial EEPROM
24C04	512 × 8	O.D	5 V	NV CMOS	Serial EEPROM
24C08	1024 × 8	O.D	5 V	NV CMOS	Serial EEPROM
24C16	2048 × 8	O.D	5 V	NV CMOS	Serial EEPROM
24C32	4096 × 8	O.D	5 V	NV CMOS	Serial EEPROM
24C64	8 K × 8	O.D	5 V	NV CMOS	Serial EEPROM
24C128	16 K × 8	O.D	5 V	NV CMOS	Serial EEPROM
24C256	32 K × 8	O.D	5 V	NV CMOS	Serial EEPROM

\*O.C-open collector, O.D-open drain, TS-Tristate

## 74S288 TTL PROM

The 74S288 is a 256 bit Schottky TTL PROM organised as  $32 \times 8$  bits. Its logic diagram is shown in Fig. 11.14. It is available in 16-pin DIP and has one enable ( $\overline{G}$ ) input terminal which controls the output state. When the device is enabled ( $\overline{G}$  LOW), the outputs ( $O_0 - O_7$ ) represent the contents of the selected word by the address input. When disabled ( $\overline{G}$  HIGH), the outputs go to the OFF (high-impedance) state. This IC is available with LOWs in all locations. A HIGH may be programmed into any selected location by blowing off the titanium-tungsten fuse which requires 10.5 V for programming.

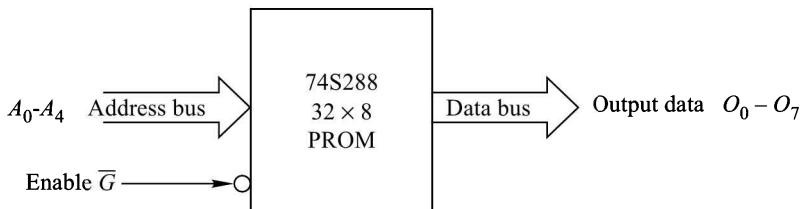


Fig. 11.14 Logic Diagram of 74S288 32 x 8 Schottky TTL PROM

## 27C010 OTP EPROM

Figure 11.15 shows the logic diagram of 27C010-1 Megabit OTP EPROM. It is a one time programmable CMOS read-only memory (OTP EPROM) organised as  $128 \text{ K} \times 8$  bits. It has three control inputs: Chip Enable ( $\overline{C}$ ), Output Enable ( $\overline{OE}$ ), and Program Store ( $\overline{G}$ ).

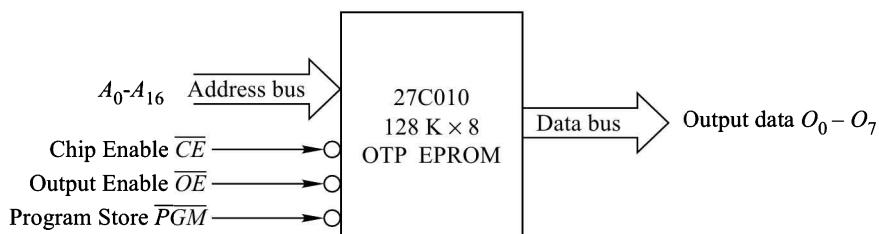


Fig. 11.15 Logic Diagram of 27C010 128 K x 8 OTP EPROM

The chip enable input ( $\overline{C}$ ) is used for enabling the chip for read operation and the output enable input ( $\overline{OE}$ ) for gating the data to the output pins. For programming, a LOW level pulse is applied at  $\overline{G}$  input with  $\overline{C}$  LOW.

## 27C64A UV EPROM/OTP EPROM

Figure 11.16 shows the logic diagram of the 27C64A 8 K  $\times$  8 EPROM. It is available in two ranges UV EPROM (reprogrammable version) and OTP EPROM (one time programmable version). The reprogrammable version is having a transparent lid for erasing the chip when exposed to ultraviolet light, whereas the device for one time programmable (OTP) version is not having the transparent lid. There are three control inputs whose

operation is similar to that explained for the 27C010 device. This device is available with all the bits in '1' state. Data is stored by selectively programming '0's into the desired bit locations.

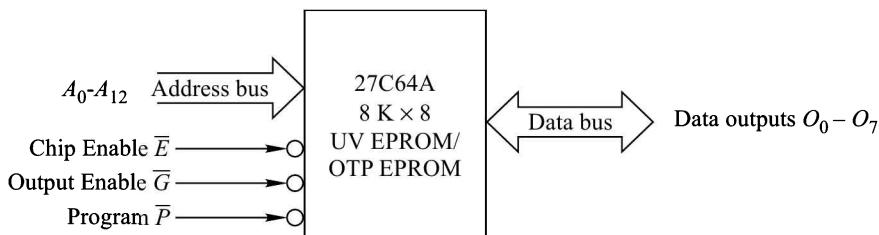


Fig. 11.16 Logic Diagram of 27C64A UV EPROM/OTP EPROM

## 28HC256 Parallel EEPROM

Figure 11.17 shows the logic diagram of 28HC256 high-speed parallel EEPROM organized as 32K  $\times$  8 bits.

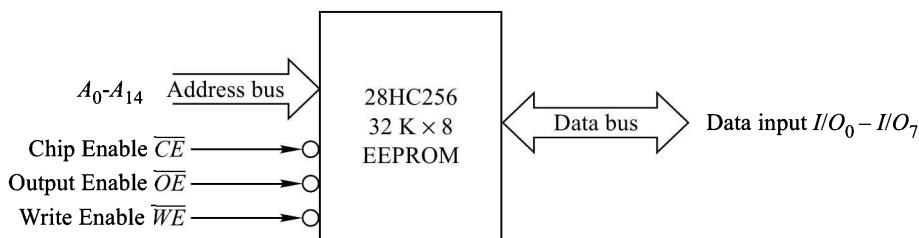


Fig. 11.17 Logic Diagram of 28HC256 32 K  $\times$  8 EEPROM

It is accessed like a static RAM for the read or write cycle. This device contains a 64-byte page register to allow writing of upto 64 bytes simultaneously. During a writing cycle, the address and 1–64 bytes of data are internally latched in the device and the data and address buses are freed for other operations. The end of a write cycle is detected by  $\overline{\text{DATA}}$  polling of  $I/O_7$ . Once the end of a write cycle has been detected a new access for a read or write can begin.

## Device Operation

**Read** When  $\overline{C}$  and  $\overline{W}$  inputs are LOW and  $\overline{W}$  input is HIGH, the data stored in the addressed memory location are available on the outputs  $I/O_0 - I/O_7$ . The outputs are in the high-impedance state when either the chip is not selected ( $\overline{C} = 1$ ) or output is disabled ( $\overline{G} = 1$ ).

**Write** A single byte or a page containing 64 bytes can be written in a single internal programming period. For write operation,  $\overline{C}$  and  $\overline{W}$  are required to be LOW and  $\overline{G}$  HIGH. The address is latched on the falling edge of  $\overline{C}$  or  $\overline{W}$  whichever occurs last. The data is latched by the first rising edge of  $\overline{C}$  or  $\overline{W}$ . In the page write operation, the first byte written can be followed by 1 to 63 additional bytes. All the bytes during a

page write operation must reside on the same page. A page is defined by  $A_6 - A_{14}$  address inputs. The  $A_0$  to  $A_5$  inputs are used to specify which bytes within the page are to be written.

During a byte or page write cycle, the complement of the last byte written will appear on the  $I/O_6$  pin. This is used for DATA polling. Once the write cycle has been completed, valid data in true form is available on all outputs. In addition to DATA polling, there is another method of determining the end of a write cycle. During the write operation, successive attempts to read data from the device will result in  $I/O_6$  toggling between '1' and '0'. Once the write cycle has completed,  $I/O_6$  will stop toggling and valid data will be read.

It is possible to erase the entire contents of the device by using a 6-byte software code by the controlling device such as a microprocessor.

## **Serial EEPROM**

Due to the unprecedented developments and applications of programmable ICs, a large number of consumer, telecommunication, medical, industrial and PC related products, etc. have become available which are small in size but may require a large non-volatile memory capable of read and write operations. The electrically erasable and programmable read-only memory (EEPROM) is the only type of memory which is non-volatile read and write memory. The parallel EEPROM discussed above requires a large number of pins for the input/output and addressing resulting in a large package size and high power consumption. Therefore, a need was felt to develop memories for large storage capacity which are non-volatile SRAM requiring smaller number of pins, smaller package size, and lower power consumption, etc. To meet all these requirements, *serial EEPROM* devices were developed. Table 11.5 gives some of the available serial EEPROMs.

### **24C01/24C02/24C04/24C08/24C16 Serial EEPROMs**

All these serial memories are electrically erasable and programmable read-only memory (EEPROM) available in 8-pin package and organised as given in Table 11.5. Figure 11.18 shows their logic diagram. In all these devices, there are three address pins  $A_2$ ,  $A_1$ , and  $A_0$  which are used for device and page addressing as discussed below. The addressing mechanism is different in different devices. The serial data (SDA) pin is bi-directional for serial input/output data transfer. It is open-drain (O.D) driven and may be wire-ORed with any number of other open-drain or open-collector devices. The serial clock input (SCL) is used to clock data into EEPROM (write operation) at the positive edge and clock data out (read operation) at the negative edge.

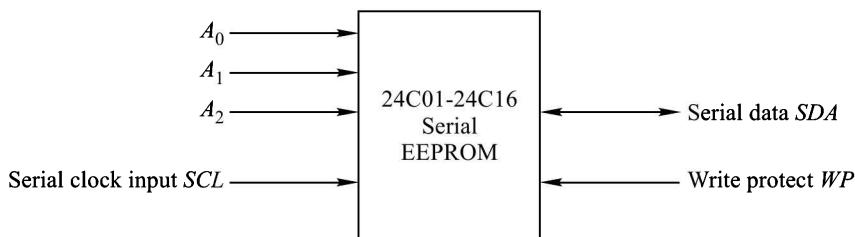


Fig. 11.18     *Logic Diagram of 24C01–24C16 Serial EEPROMs*

## **Device Addressing**

All these serial EEPROM devices require an 8-bit device address word. The four most significant bits  $D_7$ ,  $D_6$ ,  $D_5$ ,  $D_4$  are fixed as 1010 for all the devices and the next three bits  $D_3$ ,  $D_2$ ,  $D_1$  correspond to  $A_2$ ,  $A_1$  and  $A_0$ .

respectively. These are used for device/page addressing. The least-significant bit is the read/write operation select bit. It is HIGH for read and LOW for write operation. Figure 11.19 gives the arrangement of bits of the addressing word. The read and write operations are controlled by the addressing device such as a microcontroller.

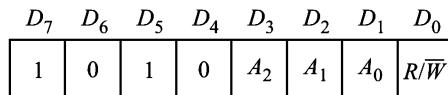


Fig. 11.19     *Serial EEPROMs Addressing Word*

The operation of  $A_2$ ,  $A_1$ ,  $A_0$  pins is given in Table 11.6.

Table 11.6     *Serial EEPROM Device Addressing Operation*

Device	Status of pins		
	$A_2$	$A_1$	$A_0$
24C01/24C02	Hard wired and upto eight 24C01/24C02 devices may be addressed		
24C04	Hard wired and upto four 24C04 devices may be addressed	Memory page address bit $P_0$ To be left unconnected	
24C08	Hard wired for upto two 24C08 devices	Memory page address bits $P_1$ To be left unconnected	$P_0$
24C16	$P_2$	$P_1$ To be left unconnected	$P_0$

## 11.6 READ AND WRITE MEMORY

Many digital systems require memories in which it should be possible to write into, or read from, any memory location with the same speed. In such memories, the data stored at any location can be changed during the operation of the system. This type of memory is known as a read/write memory and is usually referred to as RAM (random-access memory).

The read-write memories (RWM)/random-access memories (RAM) are fabricated using bipolar devices or unipolar (MOS) devices. There are two types of RAMs. These are *static* RAM (SRAM) and *dynamic* RAM (DRAM). Bipolar RAMs are static, whereas the MOS RAMs can be static or dynamic. The basic storage cell of a static RAM is a bistable circuit, i.e., a latch, which simply consists of two cross-coupled inverters as shown in Fig. 7.3. A RAM is an array of these storage cells requiring as many FLIP-FLOPs as the bit storage capacity of the RAM, which is usually a large number. The storage cell of a DRAM is simply a capacitor, therefore, only MOS devices can be used for dynamic random-access memories. Since capacitors leak charge, therefore, the voltage stored in it gets reduced with time which requires periodic refreshing. In general, bipolar RAMs are faster than the MOS RAMs. With improvements in the MOS technology, it has become possible to make MOS RAMs with speeds (access time) comparable to those of bipolar RAMs.

### 11.6.1 Bipolar RAM Cell

The basic bipolar RAM storage cell is shown in Fig. 11.20. In this FLIP-FLOP, one transistor is ON and the other is OFF. When the OFF transistor is forced into the ON state by an external trigger pulse, the transistor that was initially ON turns OFF. Thus, it has two stable states which are used to store information in the form of logic 0 and 1. Special features are incorporated in this cell for addressing the cell, writing into it, and reading from it.

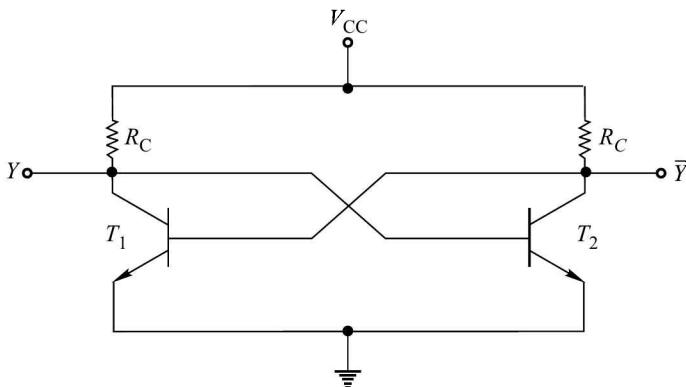


Fig. 11.20 Basic Bipolar RAM Storage Cell

Figure 11.21 illustrates a bipolar RAM cell with the required facilities incorporated. The FLIP-FLOP consists of the transistors  $T_1$  and  $T_2$ . These transistors are provided with additional emitters to incorporate the facility of addressing them. The remaining circuitry provides a mechanism for writing and reading data. Signals  $A_x$  and  $A_y$  are used to address the cell which are the outputs of the row-select and column-select address decoders, respectively. The cell is accessed for reading or writing when  $A_x = A_y = 1$ .

Let  $A_x$ ,  $A_y$ , and  $W/\bar{R}$  inputs be at logic 0. The outputs of gates  $G_1$  and  $G_2$  will be 1, which make transistors  $T_3$  and  $T_4$  ON. Therefore, the diodes  $D_1$  and  $D_2$  are non-conducting. If, say, the state of the FLIP-FLOP is such that  $T_1$  is ON and  $T_2$  is OFF, then the emitter current will flow in  $E_x$  and  $E_y$ . A bias voltage of 0.5 V is applied through the resistor  $R_3$  to emitter  $E_D$ . Therefore,  $E_D$  is more positive than  $E_x$  and  $E_y$  and hence  $E_D$  does not conduct. The transistors  $T_5$  and  $T_6$  are also OFF, and the data output terminal is at logic 1 and will remain in this state, independent of the state of the FLIP-FLOP.

Now, let the cell be addressed by raising both  $A_x$  and  $A_y$  to logic 1. Then, the currents through  $E_x$  and  $E_y$  will be diverted to  $E_D$ . A component of this current will flow into the base of  $T_5$  and the data output terminal will assume the same logic level as present at the collector of  $T_1$ . Thus, with the  $W/\bar{R}$  at logic 0, the operation of addressing the cell provides a reading of the cell.

Now, let  $A_x = A_y = 1$  and  $W/\bar{R}$  input be at logic 1. If the data input is at logic 1, the output of  $G_1$  will be 1 and the output of  $G_2$  will be 0. Therefore,  $T_3$  is ON and  $T_4$  is OFF. The voltage at the collector of  $T_4$  rises,  $D_2$  conducts and raises the voltage at  $\bar{E}_D$ . Hence, irrespective of the original state of the FLIP-FLOP,  $T_2$  cannot conduct. Hence, the logic level at the collector of  $T_2$  will become the logic level of the data input. If the cell is not addressed,  $E_D$  and  $\bar{E}_D$  will not be carrying current and the FLOP-FLOP will not respond to the writing operation. When a new data is written, the earlier stored data, if any, gets replaced with the new data.

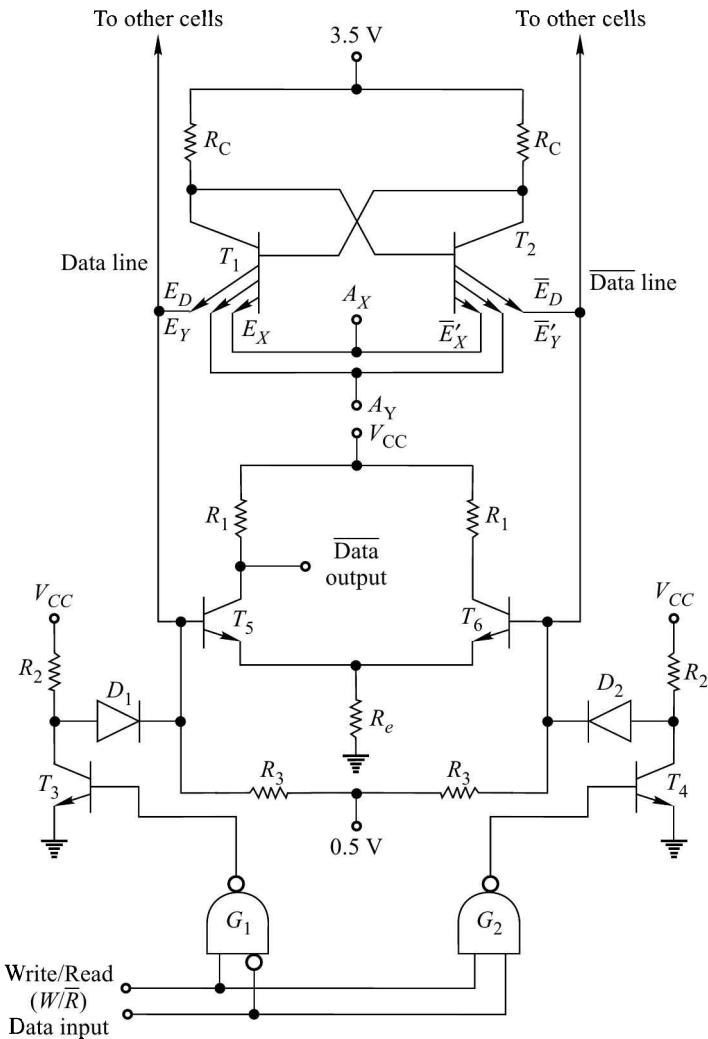


Fig. 11.21 **Bipolar RAM Cell**

### 11.6.2 MOSFET Static RAM (SRAM) Cell

A CMOS SRAM memory cell is shown in Fig. 11.22. Each bit in an SRAM is stored on four transistors, two NMOS and two PMOS, that form two cross-coupled inverters. Two additional transistors  $T_5$  and  $T_6$  serve to control the access to a storage cell for read and write operations. Access to the cell is enabled by the word line ( $WL$ ) which controls the two transistors  $T_5$  and  $T_6$  which, in turn control whether the cell should be connected to the bit lines  $BL$  and  $\overline{BL}$ . These bit lines are used to transfer data for both read and write operations.

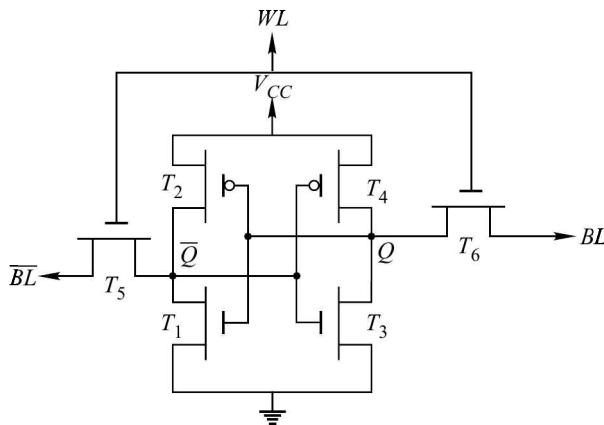


Fig. 11.22 A Six-Transistor CMOS SRAM Cell

### Read Operation

Assume that the content of the memory is  $Q = 1$ . The read cycle is started by precharging both the bit lines  $BL$  and  $\bar{BL}$  to logic 1, then asserting the word line  $WL = 1$  enables both the access transistors  $T_5$  and  $T_6$ . The values stored in  $Q$  and  $\bar{Q}$  are now transferred to the bit lines by leaving  $BL$  at its precharged value and discharging  $\bar{BL}$  through the transistors  $T_1$  and  $T_5$  to logic 0. On the  $BL$  side, the transistors  $T_4$  and  $T_6$  pull the bit line to  $V_{CC}$ , i.e., logic 1. If the content of memory is  $Q = 0$ , the opposite will happen and  $\bar{BL}$  would be pulled towards 1 and  $BL$  towards 0.

### Write Operation

For writing into the cell, the bit to be stored is applied at  $BL$  and its inverse at  $\bar{BL}$ . When the word line  $WL$  is asserted, the value to be stored is latched. The new bit replaces the earlier bit stored.

#### 11.6.3 Asynchronous SRAM

Figure 11.23 shows a functional block diagram of an  $8\text{ K} \times 8$  CMOS static RAM. It has a single decoder circuit and three control inputs: chip enable ( $\bar{CE}$ ), output enable ( $\bar{OE}$ ) and write enable ( $\bar{WE}$ ). All the three control inputs are active-low. Its truth table is given in Table 11.7.

Table 11.7 Truth Table of Asynchronous SRAM of Fig. 11.23

Mode	$\bar{WE}$	$\bar{CE}$	$\bar{OE}$	I/O operation
Not selected (Power-down)	X	H	X	High-Z
Output disabled	H	L	H	High-Z
Read	H	L	L	$D_{out}$
Write	L	L	X	$D_{in}$

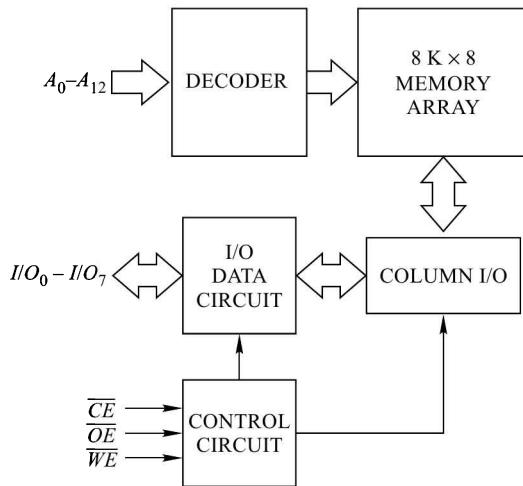


Fig. 11.23 Functional Block Diagram of an Asynchronous SRAM

Figure 11.24 shows the functional block diagram of another asynchronous SRAM which has two-dimensional decoding mechanism. It uses two decoders.

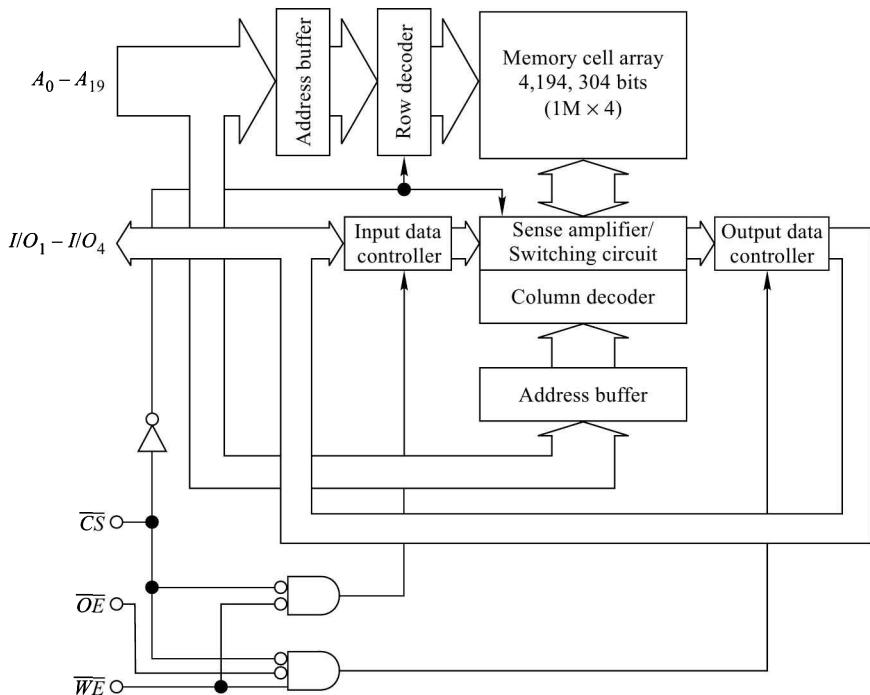


Fig. 11.24 Functional Block Diagram of an Asynchronous SRAM with Two Dimensional Decoding

The first half of the address bus  $A_0$ – $A_9$  are decoded by the row decoder and the other half  $A_{10}$ – $A_{19}$  are decoded by the column decoder. Using two decoders save hardware requirement for decoders in comparison to using one decoder. The truth table of this SRAM is same as that given in Table 11.7.

#### 11.6.4 Synchronous SRAM

A synchronous SRAM uses the system clock to synchronise its operation with the controlling device, such as a microprocessor, for faster operation. Its memory cell, memory array, address decoder, and read ( $\overline{OE}$ )/write ( $\overline{WE}$ ) enable inputs are similar to an asynchronous SRAM, but the various registers used operate in synchronism with the system clock. The address, read/write and chip enable (or select), and the input data are all latched into their respective registers on an active edge of the clock pulse. Figure 11.25 shows the block diagram of a synchronous SRAM with burst feature. Here, the address and the data bus are shown as single line with a slash and the number of lines mentioned by its side. This is a most commonly used convention in digital systems.

There are two basic types of synchronous SRAMs depending upon the way the output data is obtained from the SRAM. These are:

- Flow-through SRAM
- Pipe-lined SRAM

The only difference between these two types of SRAMs is the presence or absence of the Data output register (shown as shaded in Fig. 11.25). The *flow-through* synchronous SRAM does not have a Data output register, therefore, the output data flow asynchronously to the data I/O lines through the output buffers. On the other hand, the data output register is present in the pipelined synchronous SRAM so that the output data are synchronously placed on the data I/O lines.

The *Burst feature* is generally incorporated in a synchronous SRAM. It allows the memory read from or write at upto four locations using a single address. A 2-bit binary counter alongwith the two EX-OR gates is used to achieve this feature in the synchronous SRAM chip. When the external address, 16-bit in this case,  $A_{15}A_{14}\dots A_2A_1A_0$  is latched into the address register, the lowest significant two bits  $A_1$  and  $A_0$  are applied to the EX-OR gates, as shown in the figure. The outputs of the EX-OR gates are  $A'_1$  and  $A'_0$ . These  $A'_1A'_0$  bits are used to replace  $A_1A_0$  bits when the address (16-bit) gets applied to the address decoder.

Let the  $A_1A_0$  bits of the external 16-bit address be 00. The counter outputs  $Q_1Q_0$  will be 00, 01, 10, 11 on the successive clock pulses, which makes  $A'_1A'_0$  to be 00, 01, 10, 11 on every successive clock pulse. The address used for accessing the memory array is  $A_{15}A_{14}\dots A_2A'_1A'_0$ . Thus, four memory locations are accessed for read/write using a single external address. This increases the speed of the memory.

There are various other features incorporated in synchronous SRAM to make them suitable for different applications. Various types of such SRAMs are:

- Late Write Synchronous SRAM
- No Wait State Bus Synchronous SRAM
- DDR Synchronous SRAM
- Quad Synchronous SRAM

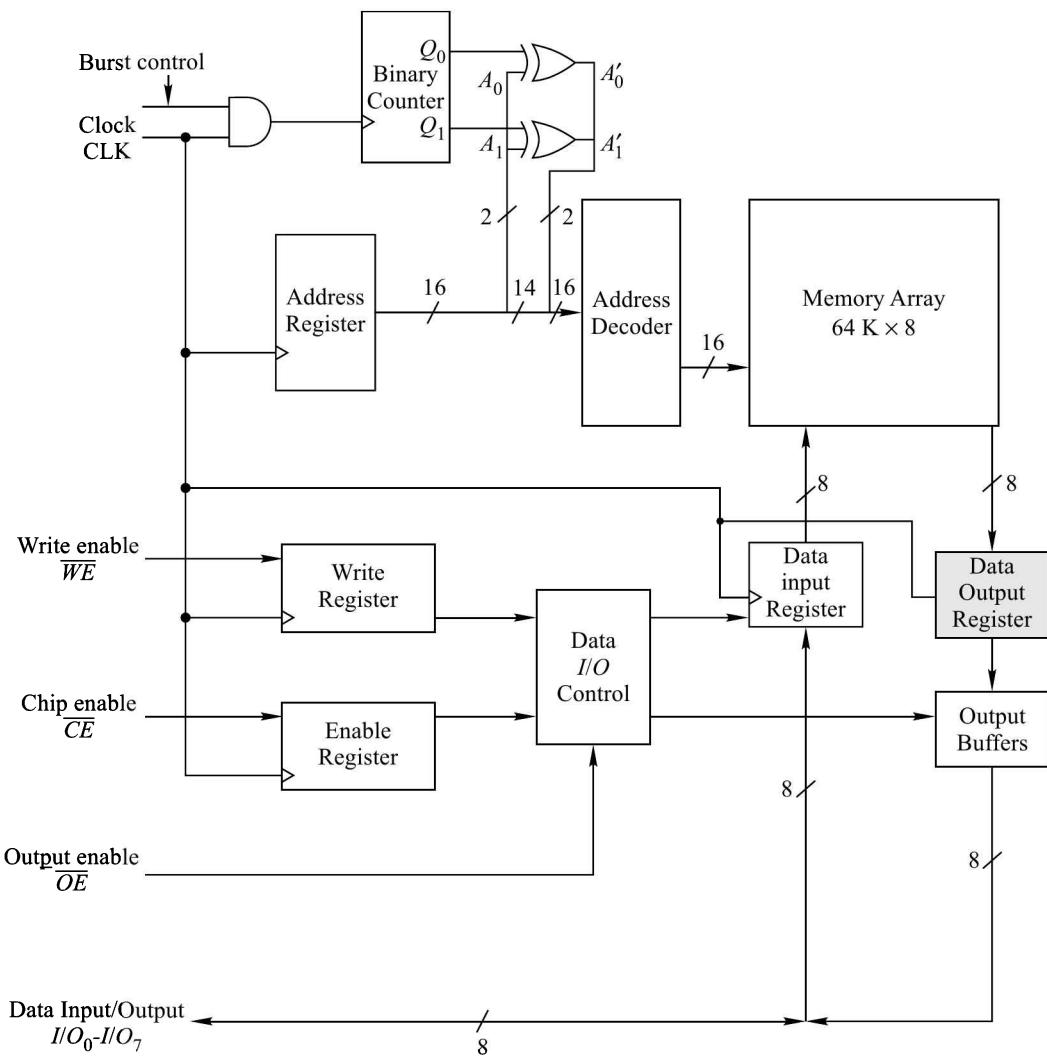


Fig. 11.25 **Block Diagram of a Synchronous SRAM with Burst Feature**

### Late Write Synchronous SRAM

In a synchronous SRAM, when the operation repeats from read to write to read to write ..., wait cycles are there which decreases its speed of operation. To reduce the number of wait cycles, Late Write synchronous SRAM has been developed which has a 'Write Address Register' which is not present in a standard synchronous SRAM. This allows an operation to write data in a cycle next to the one in which an address input to a synchronous memory is applied. The write address register is used to save an address to the write address register once even if an address for read operation is input in a cycle in which written data is to be input. The

operation can be completed by using the address of the write address register after the read operation has been completed.

### No Wait State Bus Synchronous SRAM

Similar to the Late write synchronous SRAM, zero wait state synchronous SRAM is available in, which there is no wait state. It has a ‘Write Address Register’, like the Late Write SRAM. Its flow-through version has a ‘Write Address Register’ of one address and the pipeline version has a ‘Write Address Register’ of two addresses. The operation of this ‘Write Address Register’ can completely eliminate wait cycles when an operation is performed from read → write → read, and so on. This device has ZBT as the trademark of IDT, and ZEROSB is the trademark of NEC.

### Double Data Rate (DDR) Synchronous SRAM

A DDR enables data to be read or written twice every clock. This type of SRAM realises a much higher data transfer rate than conventional synchronous SRAM. Data *I/O* at 500 MHz is obtained with an external input clock of 250 MHz. Two versions of DDR synchronous SRAM are

- Common *I/O* version
- Separate *I/O* version

In the case of common *I/O* version, there is common bus for input/output, whereas in separate *I/O* version, there are separate buses for input and output.

### Quad Synchronous SRAM

Quad synchronous SRAM enables data to be read or written four times to that of conventional synchronous SRAM. It has separate read and write ports with concurrent read and write operation, and DDR interface for read and write operation.

#### 11.6.5 MOSFET Dynamic RAM (DRAM) Cell

The earlier DRAMs were made using 3-transistor cell, which were later replaced by 1-transistor cell. Figure 11.26 shows a 1-transistor DRAM cell. In this cell, the data bit is stored in a small capacitor rather than in a latch used for SRAM cell. Also, in this cell, only one transistor and a capacitor are required per bit, compared to six transistors in SRAM (Fig. 11.22). This allows DRAM to have very high density in comparison to SRAM. The main disadvantage in a DRAM cell is that since the charge is stored in a capacitor, which can not hold it over an extended period of time. Therefore, the stored bit can not remain unless the charge is replenished or refreshed periodically. This requires additional circuitry.

Figure 11.27 shows a DRAM cell alongwith the simplified circuitry for read, write, and refresh operations.

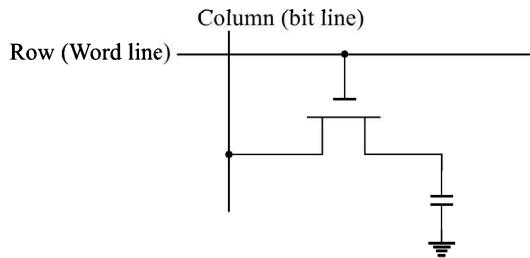


Fig. 11.26 A 1-Transistor DRAM Cell

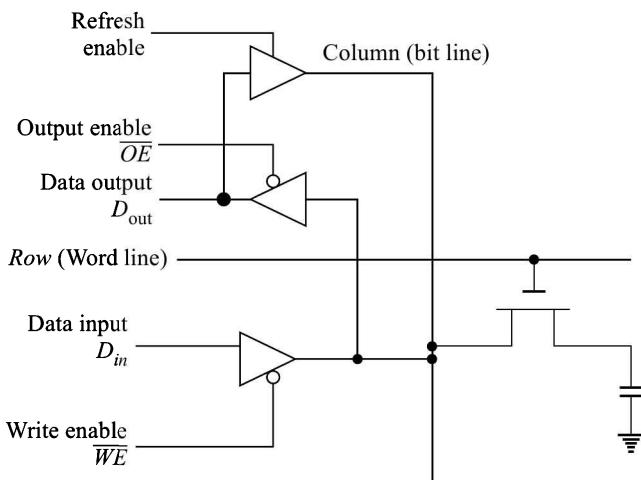


Fig. 11.27 A DRAM Cell with Read, Write, and Refresh Circuitry

## Read Operation

For reading or writing operation, the word line (*Row*) is to be selected which switches ON the transistor. The output enable  $\overline{OE}$  LOW will enable the output buffer, making its output same as the bit line which is at the same logic level as the voltage on the capacitor. Thus, the output is at logic 1 corresponding to the capacitor charged and logic 0 corresponding to discharged capacitor.

## Write Operation

With the *Row* line selected, the write enable  $\overline{WE}$  LOW allows writing into the cell. If the  $D_{in}$  bit is 1, the capacitor gets charged to logic 1 through the ON transistor, whereas, if  $D_{in}$  bit is 0, the capacitor gets discharged through the ON transistor to the logic 0. When the  $\overline{WE}$  is made HIGH, the charge on the capacitor remains trapped on the capacitor (1 or 0).

### 11.6.6 Asynchronous DRAM

#### DRAM Organisation

Figure 11.28 shows the functional block diagram of a  $512 \text{ K} \times 8$  DRAM. Its address bus width is 19. To reduce the number of pins in the chip for addressing and complexity of a decoder of 19-to-524288 size, the address is divided in two halves and two decoders are used. The lower order ten bits ( $A_0 - A_9$ ) are applied to the row decoder and the higher order nine bits are applied to the column decoder. Also, the address is multiplexed. The number of address inputs available is ten,  $A_0 - A_9$ . The address is applied to the DRAM address bus in two parts using a multiplexer arrangement. The first ten address bits ( $A_0 - A_9$ ) are entered as row address and latter nine address bits ( $A_0 - A_8$ ) are entered as column address. The row address is latched by the Row Address Strobe ( $RAS$ ) and the column addressed is latched by the Column Address Strobe ( $CAS$ ).

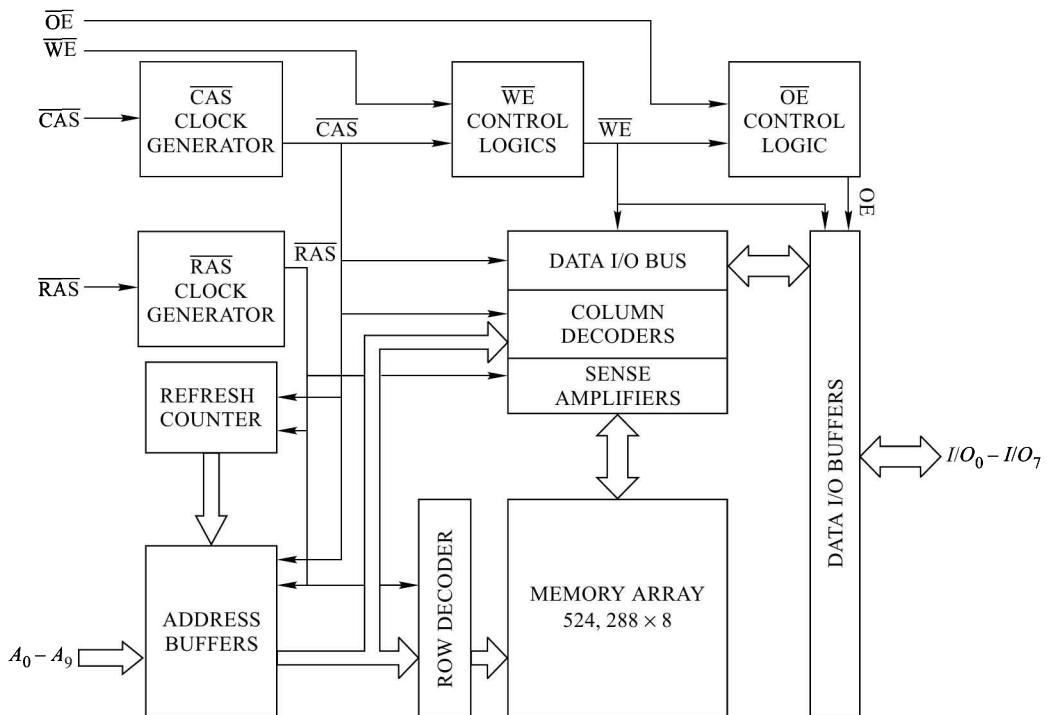


Fig. 11.28 Functional Block Diagram of a 512 K x 8 DRAM

A read cycle is initiated by falling edge of  $\overline{CAS}$  or  $\overline{OE}$ , whichever occurs last, while holding  $\overline{WE}$  HIGH. Similarly, a write cycle is initiated by falling edge of  $\overline{CAS}$  or  $\overline{WE}$ , whichever occurs last.

To retain data, all rows in the memory array are refreshed consecutively each refresh period. The refresh period is normally of 16 ms. Since there are 10 rows in 512 K x 8 DRAM, therefore, 1024 refresh cycles are required in each 16 ms period. All the 1024 row addresses ( $A_0 - A_9$ ) with  $\overline{RAS}$  LOW are clocked atleast once every 16 ms.

### Fast Page Mode DRAM (FPM DRAM)

In a DRAM, when a row address is applied, all the columns in a given row can be accessed by applying successive column addresses at random with the row address applied only once. A page in memory is referred to all of the column addresses contained within one row address. In a fast page mode operation, the  $\overline{CAS}$  must remain LOW until the valid data from a given address are accepted (latched) by the external system, such as a microprocessor. When  $\overline{CAS}$  goes HIGH, the data outputs are disabled. The next column address must not occur until the data from the current column address are transferred to the external device.

### Extended Data Out DRAM (EDO DRAM)

In an EDO DRAM, the data outputs are not disabled when the  $\overline{CAS}$  signal goes from LOW to HIGH. Therefore, the valid data are available from the current address until the next  $\overline{CAS}$  signal goes LOW.

Therefore, the next column address can be accessed before the external system accepts the valid data from the current column address. Because of this type of operation, the access time of EDO DRAM is reduced in comparison to FPM DRAM.

### **Burst EDO DRAM (BEDO DRAM)**

BEDO DRAM is a modified EDO which was developed by adding an address counter to provide for burst memory operation. Similar to synchronous burst SRAM, it can process four memory addresses in one burst. The burst operation takes place in synchronism with the controlling device, such as a microprocessor. This device is faster than EDO but it never became popular because of the development of synchronous DRAM at about the same time which is much superior to BEDO RAM.

### **11.6.7 Synchronous DRAM (SDRAM)**

The operation of the synchronous DRAM is synchronised with the system clock. The SDRAMs are extensively used in computers and other microprocessor based systems requiring large capacity memory. Similar to synchronous SRAM, synchronous DRAMs are available with double data rate (DDR) transfer. The DDR SDRAMs are widely used in various digital systems and computers.

### **11.6.8 RAM ICs**

A large variety of SRAM and DRAM ICs are available from various manufacturers. CMOS technology is the most preferred technology for all the latest SRAM and DRAM ICs. Some of the available SRAM ICs are given in Table 11.8 and DRAM ICs are given in Table 11.9. There complete technical and operational details can be obtained from the websites of the manufacturers/vendors.

Table 11.8 Available CMOS SRAM ICs

IC No.	Asynchronous		Speed ns	Characteristics
	Organisation No. of bits	Power supply voltage		
61C64AL	8 K × 8	5 V	10	High speed TTL compatible interface level
61C3216AL	32 K × 16	5 V	12	High speed TTL compatible interface level
444001	4 M × 1	5 V	10, 11, 12	High speed
444004	1 M × 4	5 V	8, 10, 12	High speed
62C256AL	32 K × 8	5 V	25, 45	LOW power TTL compatible inputs/outputs
6C1008	128 K × 8	2.7–5.5 V	55	LOW power
6C2008A	256 K × 8	2.7–5.5 V	55	LOW power

IC No.	Synchronous		Speed MHz	Type
	Organisation No. of bits	Power supply voltage		
61LP6432A	64 K × 32	3.3 V	133	Pipeline Burst
61LPS12832A	128 K × 32	3.3 V	250	Pipeline Burst
61LF6432A	64 K × 32	3.3 V	90	Flow through Burst
61LF6436A	64 K × 36	3.3 V	90	Flow through Burst
61NLP6432A	64 K × 32	3.3 V	250, 200	Pipeline ‘No Wait’ state
61NLP6436A	64 K × 36	3.3 V	250, 200	Pipeline ‘No Wait’ state
61NLP128 × 18 A	128 K × 18	3.3 V	250, 200	Pipeline ‘No Wait’ state
61DDB 41M36	1 M × 36	1.8 V	250	DDR II common I/O
61DDB42M18	2 M × 18	1.8 V	250	DDR II common I/O
61QDB41M36	1 M × 36	1.8 V	250	Quad Separate input, output
61QDB42M18	2 M × 18	1.8 V	250	Quad Separate input, output

Table 11.9 Available CMOS DRAM ICs

IC No.	Asynchronous		Read/Write cycle time (min.) ns	Type
	Organisation No. of bits	Power supply voltage		
41C85125	512 K × 8	5 V	60	FPM TTL compatible
41LV85125	512 K × 8	3.3 V	60	FPM TTL compatible
41LV16257B	256 K × 16	3.3 V	60	FPM TTL compatible
41C85120	512 K × 8	5 V	60	EDO TTL compatible
41LV85120	512 K × 8	3.3 V	60	EDO TTL compatible
41LV16256B	256 K × 16	3.3 V	60	EDO TTL compatible

IC No.	Synchronous		Speed MHz	Type
	Organisation No. of bits	Power supply voltage		
42S16100C1	512 K × 16	3.3 V	200	SDRAM
43R16800B	8 M × 16	2.5 V	200	DDR SDRAM
43DR32800A	8 M × 32	1.8 V	800	DDR II SDRAM
43DR32801A	8 M × 32	1.8 V	800	DDR II SDRAM

## 11.7 FLASH MEMORY

Flash memory is non-volatile memory that can be electrically erased and reprogrammed. It is a specific type of EEPROM that is erased and programmed, in-circuit, in large blocks in contrast to EEPROM which is erased and reprogrammed at the byte level. Since the flash memory can be written to in block size rather than byte, it is easier to update it. On the other hand, it is not useful as RAM because RAM needs to be addressable

at the byte and not the block level. However, it is also sometimes referred to as ‘non-volatile RAM’. This type of memory has been named as ‘flash memory’ because a large block of memory could be erased at one time, i.e., in a single action or ‘flash’.

### 11.7.1 Flash Memory Cell

The flash memory cell consists of one transistor with a floating gate similar to an EPROM cell. However, there is a difference in the geometry and technology between flash devices and EPROM devices. The gate oxide between the silicon and the floating gate is thinner for flash technology and the source and drain diffusion are also different. Other flash cell concepts are based upon EEPROM technology. Although the flash cells are larger than the conventional one-transistor EPROM cell, but are far smaller than the conventional two-transistor EPPOM cell. The flash memory chip size is thus considerably less than the EEPROM, due to which flash memory’s density is higher and its cost per bit is lower than EEPROM.

### 11.7.2 Flash Memory Architecture

There are two common flash memory architectures: NOR and NAND. The NOR architecture is the most popular flash memory architecture. It is commonly used in EPROM and EEPROM designs also. Although the NAND structure is considerably more compact, its main drawback is that when a cell is read, the sense amplifier sees a weaker signal, than that on a NOR configuration, since several transistors are in series in NAND structure. The transistors are in parallel in NOR structure (See Figs. 4.29 and 4.30).

### 11.7.3 Flash Memory ICs

The flash memory ICs are available in a variety of configurations and architectures to address various different requirements. The devices are available in different sector sizes ranging from 64–256 bytes, making it suitable for code and data storage. The flash memories are available from 256 K bits to 128 M bits in various sizes and organisations and operate with single voltage supply of 5V, 3 V, 2.7 V, and 1.8 V, etc. Similar to EEPROM devices, flash memories are also available in parallel as well as serial types.

Some of the available parallel flash memory ICs are given in Table 11.10.

Table 11.10 *Available CMOS Parallel Flash Memory ICs*

IC No.	Organisation No. of bits	Power supply voltage	No. of sectors	Sector size bytes
29C256	32 K × 8	5 V	512	64
29LV256	32 K × 8	3 V	512	64
29C512	64 K × 8	5 V	512	128
29LV512	64 K × 8	3 V	512	128
29C010A	128 K × 8	5 V	1024	128
29LV010A	128 K × 8	3 V	1024	128
29C1024	64 K × 16	5 V	512	128 words
29LV1024	64 K × 16	3 V	512	128 words

The devices listed in Table 11.10 are having large memory arrays broken up into small individually reprogrammable sectors. For example, 29C010A is divided into 1024 sectors of 128 bytes each. Each sector's contents may be altered independently and no previous erase is required. It takes about 10–20 ms for altering the data of a sector. In case of large-sectored devices or whole chip flash devices, the write time required may extend to several minutes. Therefore, when only a small portion of the total memory is required to be altered, the small sector approach saves considerable time.

The programming method requires only nanoamp of high voltage (15 V–20 V) programming current, allowing the use of an on chip charge pump to generate the necessary programming voltages.

A number of serial flash memory devices are also available with varying sizes. For example, AT25DF021 is a 2-M bit 2.3 or 2.7 V, 70 MHz serial flash memory. It can be byte or page programmable and block (4 KB, 32 KB, and 64 KB) and chip erasable.

The detailed technical and operational specifications of various parallel and serial flash memories, can be obtained from the manufacturers' catalogues or their websites.

## 11.8 CONTENT ADDRESSABLE MEMORY

The content addressable memory (CAM) is a special purpose random access memory device that can be accessed by searching for data content. For this purpose, it is addressed by associating the input data, referred to as *key*, simultaneously with all the stored words and produces output signals to indicate the match conditions between the key and the stored words. This operation is referred to as *association* or *interrogation* and this type of memory is also known as *associative memory*.

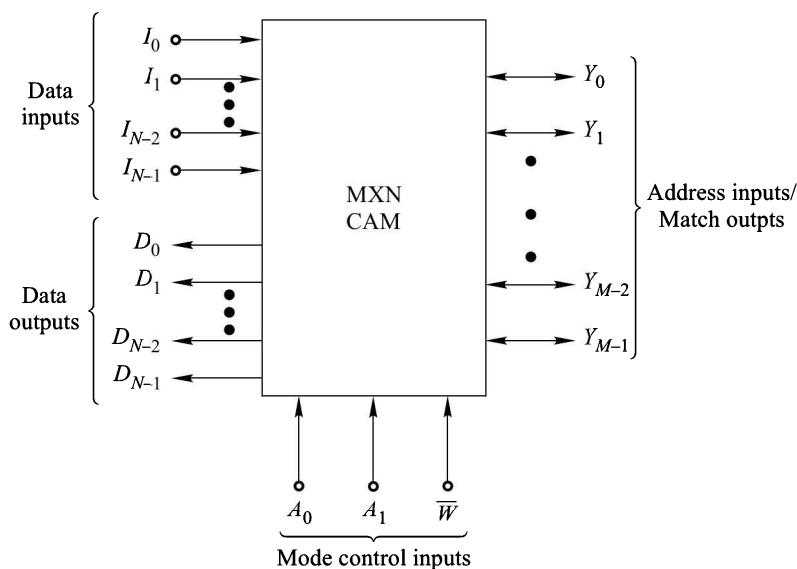
After identifying the locations whose contents match the key, read or write operations can be performed to these locations. The key to be used may either consist of the entire data word or only some specific bits of the data word, i.e. the other bits can be masked.

A CAM differs from the conventional memory organization in that the addressing of a location in the latter has no relation to the memory content. A CAM has the ability to search out or interrogate stored data on the basis of its contents and, therefore, can be a powerful asset in many applications. For example, consider a list containing the names of persons, their ages, professions, and nationalities stored in a CAM. If one is interested in finding out engineers in the list, the CAM is able to check every memory location simultaneously by using the coded form for engineer as the key. On the other hand, if it is required to find the engineers of Indian nationality, the key will consist of the combination of the codes corresponding to engineer and Indian nationality. All the memory locations with engineers of Indian nationality will be identified and the remaining data (name and age) can then be retrieved by using the read operation. To do the same search process with a conventional memory, each memory word is to be read out and compared with the key. This search is a serial process and hence time consuming. Thus, CAMs are better suited for information retrieval than the conventional memories.

CAMs are manufactured using MOS, CMOS, or bipolar technologies. The most popular CAMs use ECL circuitry because of its high speed operation.

### 11.8.1 Operation of CAM

A CAM can perform three basic operations: read, write and associate. Figure 11.29 shows a block diagram of a CAM. Its storage capacity is  $M \times N$  bits and is organized as  $M$  words of  $N$  bits each. It has  $N$  data input and  $N$  data output lines (one line for each bit of a word). The data input lines  $I_0$  through  $I_{N-1}$  are used to input

Fig. 11.29 **Block Diagram of a CAM**

data to be written into the memory and for key word in case of associate operation. Data are read out of the CAM at the data output lines  $D_0$  through  $D_{N-1}$ .

The  $Y$  lines ( $Y_0$  through  $Y_{M-1}$ ) are bidirectional. During a read or write operation, these lines are used to select the storage location. There is one address input line for each word in the CAM. For example,  $Y_0$  is the address line for memory location 0,  $Y_1$ , for memory location 1 and so on. Notice that linear selection addressing is used in CAMs rather than coincident selection addressing.

The  $Y$  lines serve as match output lines one for each memory location, when an association operation is performed. For example, if the keyword matches with the word stored in memory locations 5 and 8, lines  $Y_5$  and  $Y_8$  will become HIGH to indicate the match condition.

The mode control inputs are used to select the required operation. The read and write operations are performed in a manner similar to that used for RAM. However, during the write operation, the input data also appear at the data outputs. The reading of the data is non-destructive. The internal architecture of a typical  $8 \times 2$  ECL CAM is shown in Fig. 11.30. It has eight 2-bit storage locations  $M_0$  through  $M_7$ , each consisting of two parts, one for the higher order bit ( $M_{01}$  through  $M_{71}$ ) and the other, for the lower order bit ( $M_{00}$  through  $M_{70}$ ). The detailed circuitry of one of the locations  $M_0$  is shown in the figure and all other locations have similar circuitry.

Each memory location consists of two  $D$ -type FLIP-FLOPs, one for each bit, and some logic gates to control the function of the CAM for read, write, and associate operations. The outputs  $Y_0$  through  $Y_7$  and the data outputs  $D_1$  and  $D_0$  are produced using wired-OR connections as shown in Fig. 11.30. The operation of this CAM is summarized in Table 11.11.

The word length and/or word size can be expanded by suitably connecting the available CAM chips in a manner similar to the one used for RAMs and ROMs expansion.

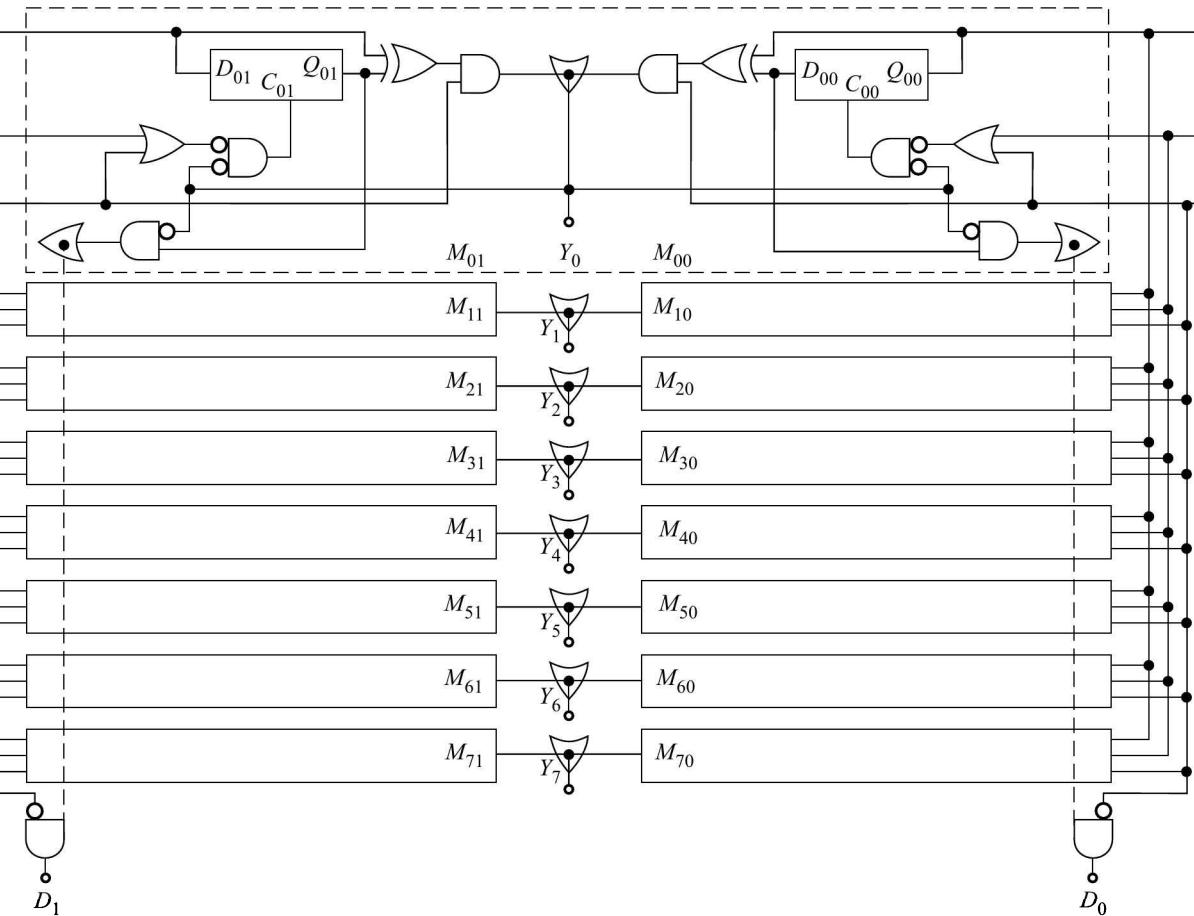


Fig. 11.30 Internal Architecture of an ECL  $8 \times 2$  CAM

Table 11.11 Operation of the  $8 \times 2$  CAM

Operation	Control inputs			Data inputs		Data outputs		$Y_n$	Value
	$A_1$	$A_0$	$\bar{W}$	$I_1$	$I_0$	$D_1$	$D_0$		
Associate	1	1	$\times$	1/0	1/0	0	0	Output	1 = Mismatch 0 = Match
Associate (higher bit masked)	0	1	1	$\times$	1/0	$D_1$	0	Output	1 = Lower bit mismatch 0 = Lower bit match
Associate (lower bit masked)	1	0	1	1/0	$\times$	0	$D_0$	Output	1 = Higher bit mismatch 0 = Higher bit match
Read	0	0	1	$\times$	$\times$	$D_1$	$D_0$	Input	0 = Selected address
Write	0	0	0	1/0	1/0	$I_1$	$I_0$	Input	0 = Selected address
	0	1	0	1/0	1/0	$I_1$	0	Output	1 = Lower bit mismatch 0 = Lower bit match
Associate and write at match addresses	1	0	0	1/0	1/0	0	$I_0$	Output	1 = Higher bit mismatch 0 = Higher bit match

### Example 11.6

Consider the CAM of Fig. 11.30. Assume that it contains data given in Table 11.12

Table 11.12 Contents of CAM

Memory location	Content
0	00
1	01
2	00
3	11
4	10
5	01
6	11
7	01

What is the word at data outputs  $D_1$  and  $D_0$  for each of the following conditions? Also find if there is any change in the memory contents.

(a)  $A_1 A_0 = 00, \overline{W} = 1$

$$Y_7 Y_6 Y_5 Y_4 Y_3 Y_2 Y_1 Y_0 = 11011111$$

(b)  $A_1 A_0 = 00, \overline{W} = 1$

$$Y_7 Y_6 Y_5 Y_4 Y_3 Y_2 Y_1 Y_0 = 11001111$$

(c)  $A_1 A_0 = 00, \overline{W} = 0$

$$Y_7 Y_6 Y_5 Y_4 Y_3 Y_2 Y_1 Y_0 = 11110111$$

$$I_1 I_0 = 00$$

(d)  $A_1 A_0 = 00, \overline{W} = 0$

$$Y_7 Y_6 Y_5 Y_4 Y_3 Y_2 Y_1 Y_0 = 11010111$$

$$I_1 I_0 = 10$$

### Solution

- (a) Since  $Y_5 = 0$ , the memory location 5 is selected for read out, i.e.

$$D_1 D_0 = 01$$

The memory contents do not change.

- (b) Since  $Y_5 = Y_4 = 0$ , the memory locations 4 and 5 are selected for read out. The output is obtained by ORing the contents of these locations, i.e.

$$D_1 D_0 = 11$$

The memory contents do not change.

- (c) Since  $Y_3 = 0$ , write operation is performed in memory location 3. The input data is stored in this location and also appears at the output.

$$D_1 D_0 = 00$$

Contents of memory location 3 = 00.

- (d) In this case, the memory locations 3 and 5 are selected for writing since  $Y_3 = Y_5 = 0$ . The contents of these locations will become 10 and also  $D_1 D_0 = 10$ .

### Example 11.7

Assume that the CAM of Fig. 11.30 contains the data given in Table 11.12.

For each of the following conditions, find the  $Y$  outputs. Also find if there is any change in the memory contents.

(a)  $A_1 A_0 = 11, I_1 I_0 = 01$

(b)  $A_1 A_0 = 01, \overline{W} = 1, I_1 I_0 = 01$

(c)  $A_1 A_0 = 10, \overline{W} = 1, I_1 I_0 = 01$

(d)  $A_1 A_0 = 01, \overline{W} = 0, I_1 I_0 = 01$

(e)  $A_1 A_0 = 10, \overline{W} = 0, I_1 I_0 = 01$

### Solution

- (a) The association operation is performed with keyword 01. The memory locations 1, 5, and 7 match the keyword giving out logic 0 at the corresponding  $Y$  outputs. Therefore,

$$Y_7 Y_6 Y_5 Y_4 Y_3 Y_2 Y_1 Y_0 = 01011101$$

- (b) In this case, the association operation is performed between the lower bit of the key with the lower bits of the stored words. Therefore,

$$Y_7 Y_6 Y_5 Y_4 Y_3 Y_2 Y_1 Y_0 = 00010101$$

- (c) In this case, the association operation is performed between the higher bit of the key with the higher bits of the stored words. Therefore,

$$Y_7 Y_6 Y_5 Y_4 Y_3 Y_2 Y_1 Y_0 = 01011000$$

- (d) The association operation is performed between the lower bit of the key and the lower bits of the stored words. The  $Y$  outputs are

$$Y_7 Y_6 Y_5 Y_4 Y_3 Y_2 Y_1 Y_0 = 00010101$$

The higher bit  $I_1$  is stored in the higher bit positions of the memory locations 7, 6, 5, 3, and 1. Therefore, the contents of the memory locations will be as given in Table 11.13.

Table 11.13

Memory location	Contents
0	00
1	01
2	00
3	01
4	10
5	01
6	01
7	01

- (e) In this case, the association operation is performed between the higher bit of the key and the higher bits of the stored words. The  $Y$  outputs are

$$Y_7 Y_6 Y_5 Y_4 Y_3 Y_2 Y_1 Y_0 = 01011000$$

The lower bit  $I_0$  is stored in the lower bit positions of the memory locations 7, 5, 2, 1, and 0. The new memory contents are given in Table 11.14.

Table 11.14

Memory location	Contents
0	01
1	01
2	01
3	11
4	10
5	01
6	11
7	01

## 11.9 FIRST-IN, FIRST-OUT MEMORY (FIFO)

A FIFO memory is a storage device in which data is read out from its memory array (SRAM) in the same order in which they were written into the memory. The first word written into the memory block is the first word that is read out of the memory block. Because of this type of operation in which the first word written in is the first word read out, it is referred to as the *first-in, first-out* memory or FIFO memory. When all the locations in the memory are filled, no more data can be written into it. Similarly, when the memory is empty, nothing can be read from it. The reading and writing operations are internally organised and the appropriate locations for writing into and reading out are pointed to by *Write Pointer* and *Read Pointer* respectively, and no address is required for accessing the memory for reading or writing.

The writing and reading operations are performed at independent data rates. This type of memory is used to interface slow input/output (*I/O*) devices to the fast operating computers. The FIFO memory is useful as a *data-rate buffer*.

A FIFO memory is basically a SRAM array with two ports and control circuitry. One port is for writing into and the other for reading from the memory array.

Each port has separate access, data, and control signal for accessing a common SRAM array. The SRAM with two ports is known as a *dual-port* SRAM. The dual-port SRAMs can be

- Asynchronous dual-port SRAM
- Synchronous dual-port SRAM
- Sequential access SRAM

In asynchronous dual-port SRAM, the operation of both the ports is asynchronous, whereas it is synchronous with two different clocks in the case of synchronous dual-port SRAM. There are two different options available in the case of synchronous dual-port SRAMs; similar to normal synchronous SRAMs. These are pipe-lined and flow-through versions.

A sequential access two-port SRAM has one of the ports random (asynchronous) access and the other port is sequential access. It is used to interface the asynchronous and synchronous components of a digital system. Normally, the controlling processor is connected to the asynchronous port and the slave processor on the synchronous side. This type of FIFO memory is used for peripheral controllers, networking equipment such as bridges, routers, etc.

The following types of FIFO memories are available:

- Asynchronous FIFO memory
- Synchronous FIFO memory
- Bi-directional FIFO memory

### 11.9.1 Asynchronous FIFO Memory

Figure 11.31 shows the functional block diagram of an asynchronous FIFO memory. It has a RAM array of size  $M \times N$ , where  $M$  is the number of words, also known as *depth*, and  $N$  is the number of bits in the word (width). The width is normally 9-bit or 18-bit. The 9-bit/18-bit wide data array allows for control and parity bits. This feature is especially useful in data communications applications where it is necessary to use a parity bit for transmission/reception error checking. There are two ports, one for writing data into the array (DATA INPUTS  $D_0-D_8$ ) and the other for reading out the data (DATA OUTPUTS  $Q_0-Q_8$ ). Here, a 9-bit wide

data array has been assumed. The READ and WRITE POINTERS point to the locations that are available at any point of time for reading or writing. There are two flags  $\overline{FF}$  (full flag) and  $\overline{EF}$  (empty flag) provided to prevent data overflow and underflow. The  $\overline{FF}$  LOW indicates that the memory is full which prevents further writing of data. Similarly, the  $\overline{EF}$  LOW indicates that the memory is empty and further reading is prevented. The expansion of data width and depth can be obtained by making use of multiple devices for which EXPANSION LOGIC BLOCK with pins  $\overline{XI}$  and  $\overline{XO}$  are available. The FIFO can be reset by activating  $\overline{RS}$  (reset) input. During reset, both the internal read and write pointers are set to the first location.

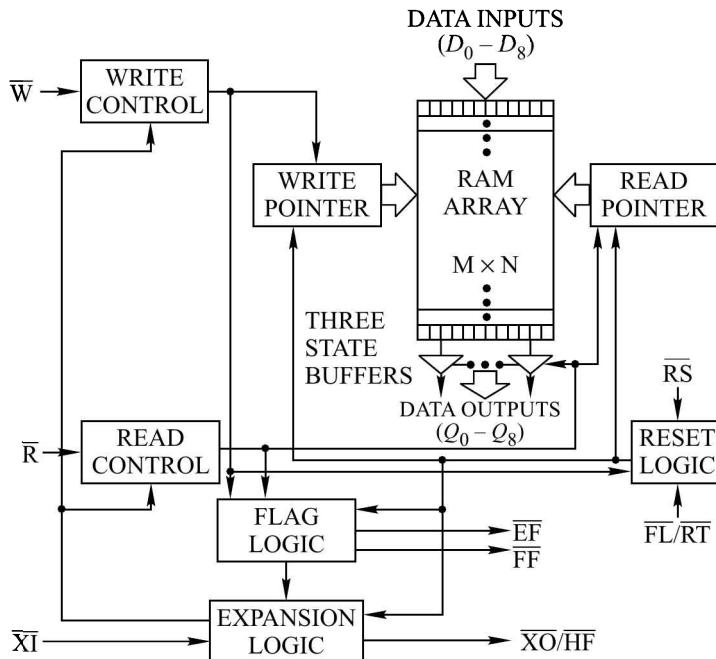


Fig. 11.31 Functional Block Diagram of an Asynchronous FIFO Memory

The  $\overline{XO/HF}$  output is a dual-purpose output. In the single device mode, when  $\overline{XI}$  input is grounded, this output acts as an indication of a half-full memory. After half of the memory is filled and at the falling edge of the next write ( $\overline{W}$  low) operation, the  $\overline{HF}$  (half full flag) will be set LOW and will remain LOW until the difference between the write pointer and read pointer is less than or equal to one half of the total memory of the device. It is then reset (HIGH) by using rising edge of the read operation. In the depth expansion mode, the input  $\overline{XI}$  is connected to  $\overline{XO}$  of the previous device.

FIFO memory has a retransmit feature ( $\overline{RT}$ ) that allows for the reset of the read pointer to its initial position when  $\overline{RT}$  is LOW. This allows retransmission from the beginning of data. This feature is available in only single device mode and not in the depth expansion mode. In the depth expansion mode, the  $\overline{FL/RT}$  is grounded to indicate that it is the first loaded device. Figure 11.32 shows the block diagram of the asynchronous FIFO memory.

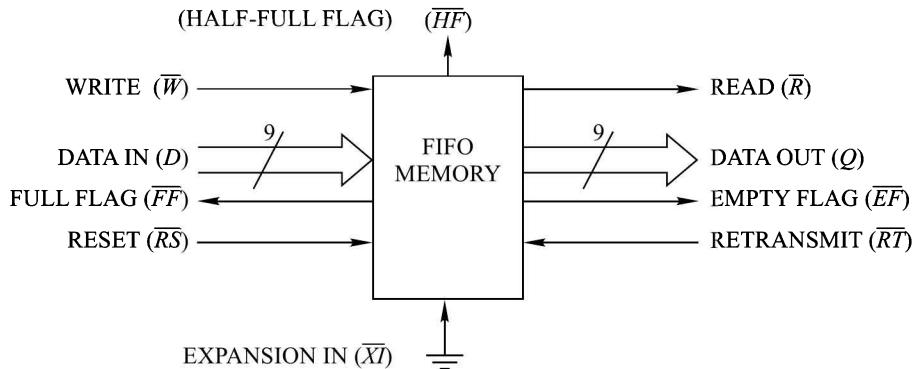


Fig. 11.32 Block Diagram of Asynchronous FIFO Memory

### Width Expansion

The width of FIFO system can be increased by simply connecting the corresponding input control signals of multiple devices. The flags can be detected from any one of the device. Figure 11.33 shows the method of width expansion of FIFO system. Here, there are two FIFO devices with 9-bit width. The FIFO-1 can be used for upper 9 bits and FIFO-2 for the lower 9 bits of 18-bit data input. Similarly, the corresponding 18-bit data output can be interfaced to the processor on the other side.

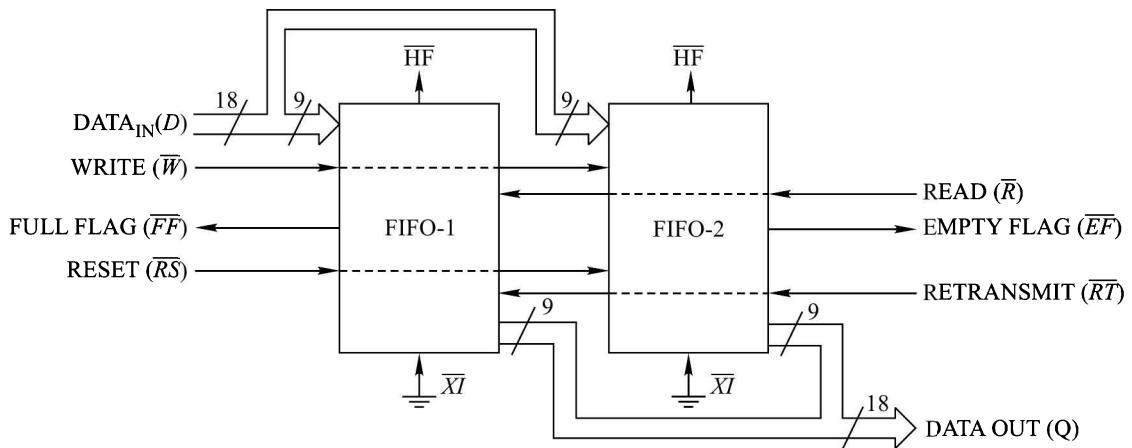


Fig. 11.33 Width Expansion of Asynchronous FIFO Memories

### Depth Expansion

For the depth expansion of FIFO memories the following conditions are required to be met:

- $\bar{FL}$  of the first device must be grounded and for all other devices, it is to be connected to logic level HIGH

- The  $\overline{XO}$  output of each device is to be connected to the  $\overline{XI}$  input of the next device.
- External logic is required to generate composite  $\overline{FF}$  and  $\overline{EF}$  signals.

Figure 11.34 shows the depth expansion method of the FIFO memories. Here, three FIFO memory devices are connected giving an overall depth of  $3 \times M$  words. The FIFO-1 is the first device which will be loaded.

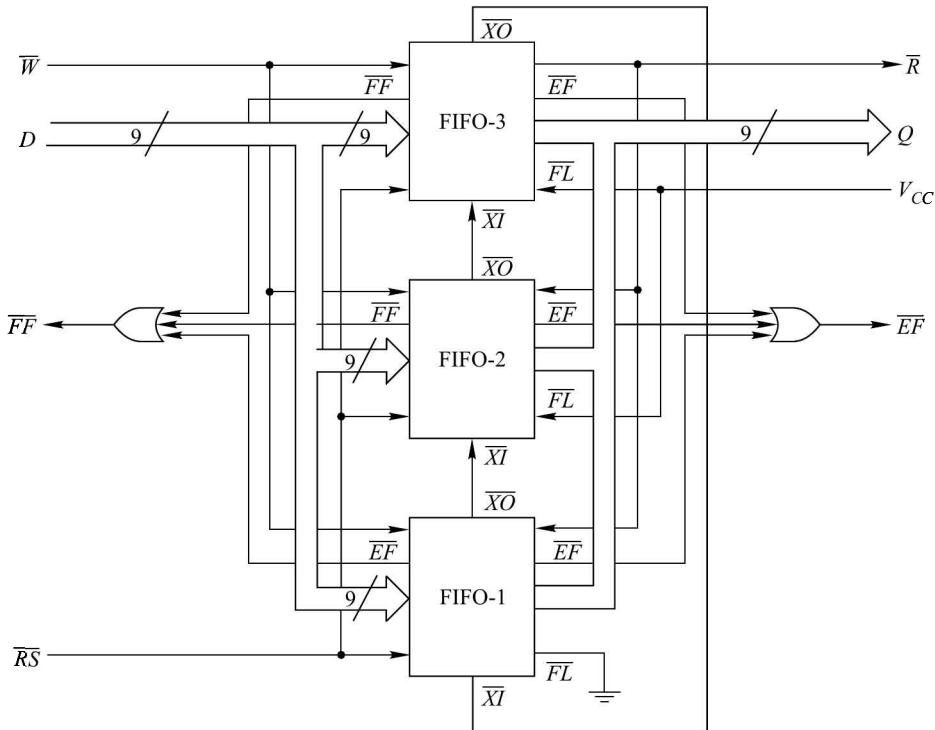


Fig. 11.34     **Depth Expansion of Asynchronous FIFO Memories**

### **Compound Expansion**

The two expansion techniques described above can be applied together in a straight forward manner to achieve large FIFO arrays.

### **Bi-directional Operation**

In a bi-directional FIFO system, data buffering between two systems, in which each system is capable of read and write operations, is allowed. It can be designed by pairing two FIFO memory devices as shown in Fig. 11.35.

### **Parallel-To-Serial FIFO**

The parallel-to-serial FIFO memories have parallel input port and serial output port. These are available in various sizes. Their width and depth can be increased by using multiple devices. For the serial output, it is

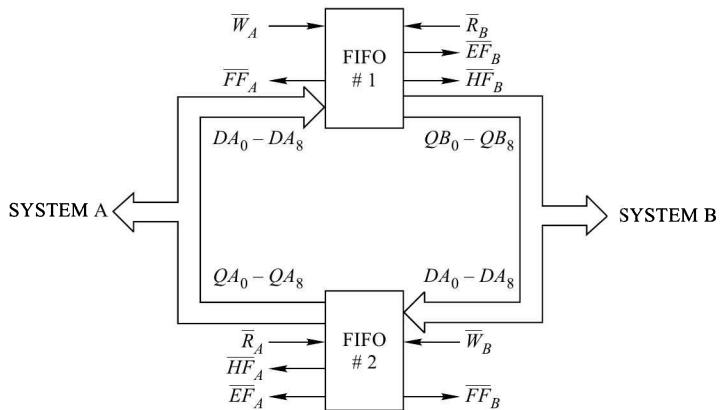


Fig. 11.35 Bi-Directional Asynchronous FIFO Memory

possible to get the least-significant bit (LSB) or the most-significant bit (MSB) first by programming. It can be used to buffer wide word widths which make these FIFOs ideal for laser printers, FAX machines, local area networks (LANs), Video storage, and disk/tape controller applications. These are asynchronous FIFO memory devices.

### 11.9.2 Synchronous FIFO Memory

Figure 11.36 shows the functional diagram of a synchronous FIFO memory. The synchronous FIFO memories are similar to the asynchronous FIFO memories as far as their SRAM array is concerned, but their read

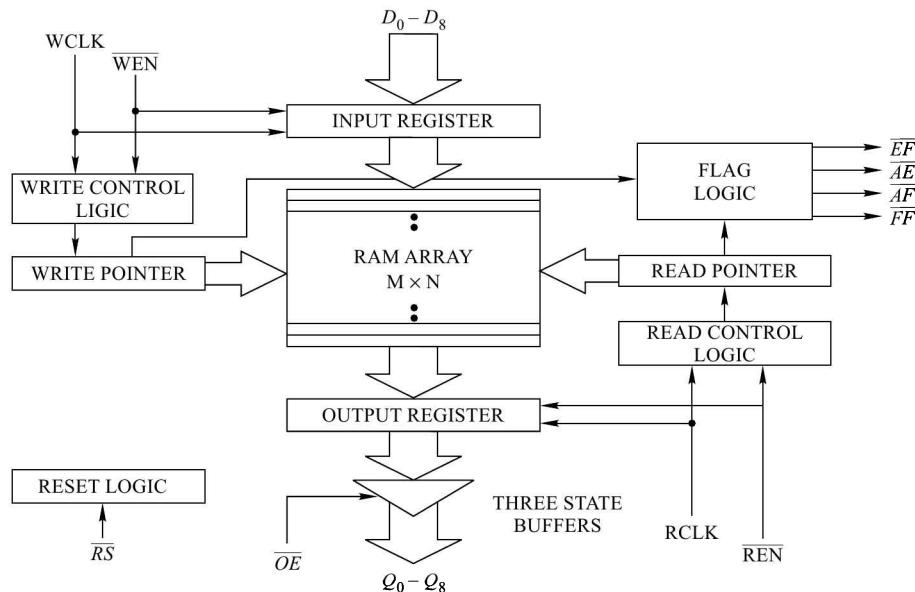


Fig. 11.36 Functional Block Diagram of a Synchronous FIFO Memory

and write controls are synchronous. There are two asynchronous (independent) clocks, one for the read operation ( $RCLK$ ) and another one for the write operation ( $WCLK$ ). Both the read and write operations can be performed using a single clock also by connecting  $WCLK$  and  $RCLK$  together. The  $\overline{WEN}$  and  $\overline{REN}$  inputs are the write enable and read enable inputs respectively. The synchronous FIFO devices have four flags. In addition to full flag ( $\overline{FF}$ ) and empty flag ( $\overline{EF}$ ), there are two programmable flags. These are almost-full flag ( $\overline{AF}$ ) and almost-empty flag ( $\overline{AE}$ ). These flags are set to Full – 7 and Empty +7 values by default, and can be programmed to any other values. For example, the default setting of  $\overline{AF}$  flag for a FIFO memory of depth 1024 words is 1017. Figure 11.37 shows the block diagram of a synchronous FIFO memory.

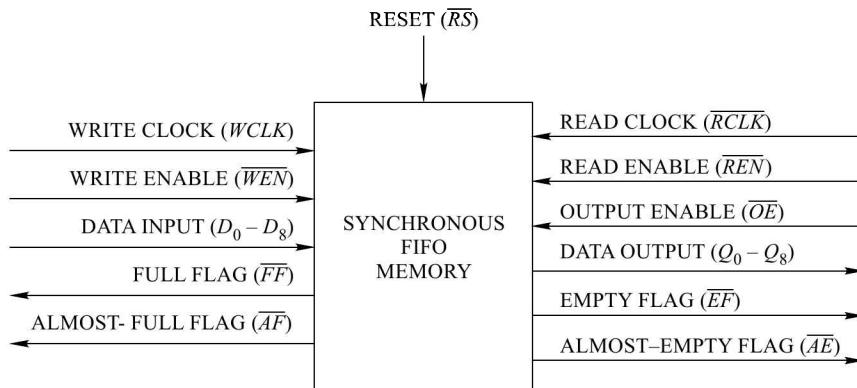


Fig. 11.37 Block Diagram of Synchronous FIFO Memory

### Width Expansion

The word width of synchronous FIFO can be increased simply by connecting the corresponding input control signals of multiple devices. Composite  $\overline{FF}$  and  $\overline{EF}$  are created by using external logic. The  $\overline{AE}$  and  $\overline{AF}$  can be detected from any one device. Figure 11.38 shows the width expansion method.

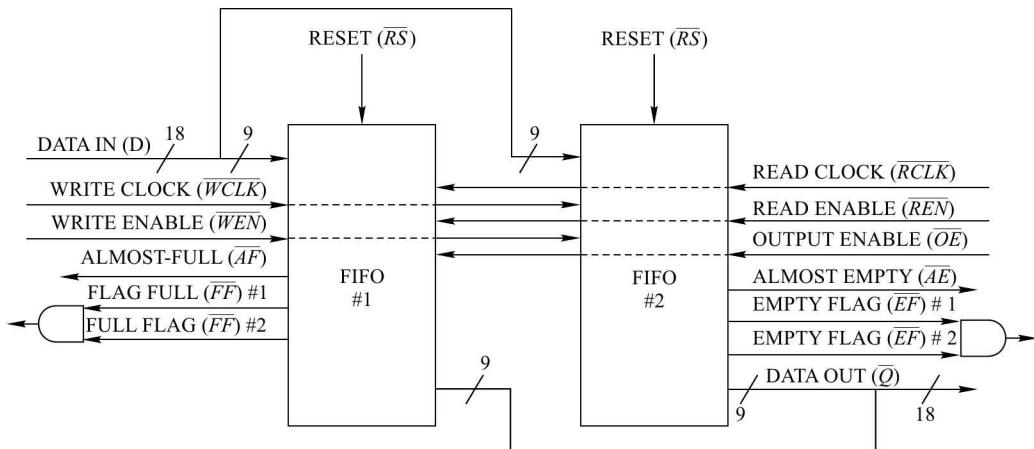


Fig. 11.38 Width Expansion of Synchronous FIFO Memories

## Depth Expansion

The depth expansion of synchronous FIFO memories is possible by using expansion logic to direct the flow of data. It can have alternate data access from one device to the next in a sequential manner.

### 11.9.3 Bi-Directional FIFO Memory (BiFIFO)

The BiFIFO memory is a synchronous FIFO device in which there are two independent clocked FIFOs buffering data in opposite directions. There are two clocks, one for each port which may be asynchronous or coincident. Figure 11.39 shows a simplified block diagram of a synchronous Bi-directional FIFO memory.

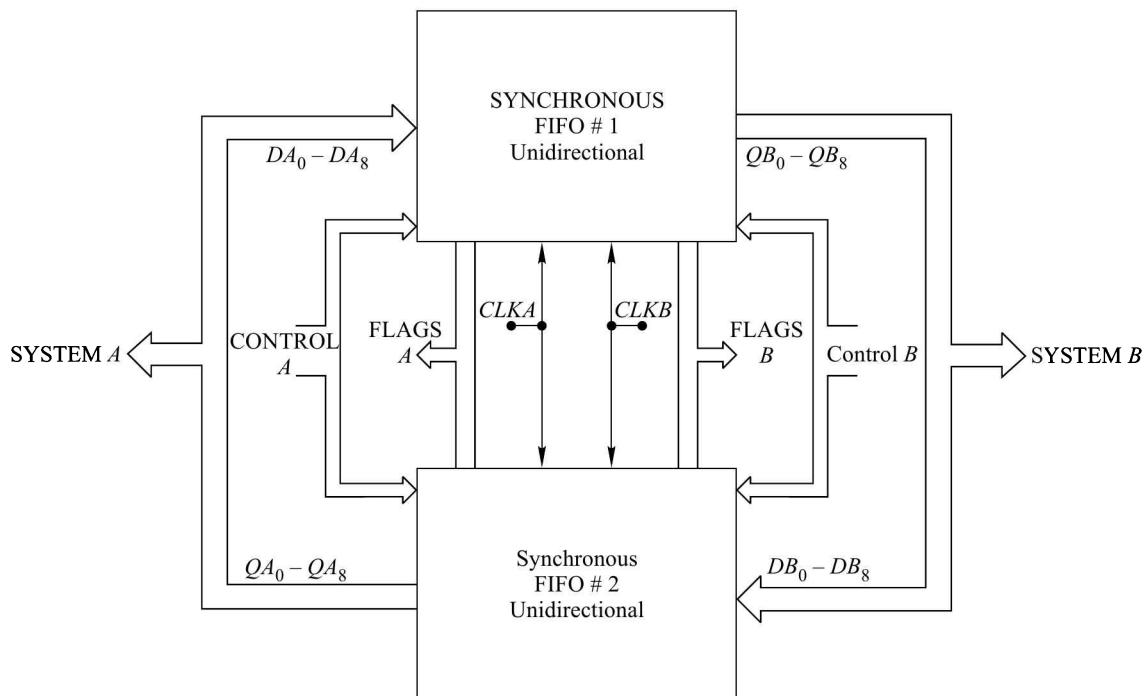


Fig. 11.39      *Block Diagram of a BiFIFO*

### 11.9.4 Dual-Port SRAM and FIFO Memory ICs

Multi-port RAM ICs are available in a variety of configurations, architectures, and options. The first-in, first-out (FIFO) memories are basically dual-port SRAMs with various types of input/output interfaces to suit different applications. Some of the available dual-port SRAM and FIFO memory ICs are given in Tables 11.15 and 11.16 respectively. Their detailed technical and operational specifications can be obtained from the manufacturers' catalogues or their websites.

Table 11.15 Available CMOS Dual-Port SRAM ICs

IC No.	Organisation No. of bits	Power supply voltage	Speed	Output compatibility	Type
7005	8 K × 8	5 V	15 ns	TTL Compatible	Asynchronous
70P244	4 K × 16	1.8 V	40 ns	LVTTL Compatible	Asynchronous
70P254	8 K × 16	1.8 V	40 ns	LVTTL Compatible	Asynchronous
70P264	16 K × 16	1.8 V	40 ns	LVTTL Compatible	Asynchronous
709079	32 K × 8	5 V	12 ns	TTL Compatible	Synchronous
709089	64 K × 8	5 V	12 ns	TTL Compatible	Synchronous
70824	4 K × 16	5 V	20 ns	TTL Compatible	Sequential access*

\*one port is sequential and the other port is a asynchronous (random).

## 11.10 CHARGE COUPLED DEVICE MEMORY

The charge coupled device (CCD), a new concept for storage of digital information, was announced in early 1970 by Bell Telephone Laboratories of U.S.A. It is an array of MOS capacitors operating as a dynamic shift register. CCDs are simple, versatile, and low cost devices and can be used wherever a serially accessed memory is required.

The operation of CCDs involve the following steps:

1. Conversion of digital input signal into charge,
2. Transfer of charge through various stages in sequential manner, and
3. Conversion of charge at the output into digital signal.

During each charge transfer step, a small amount of charge is lost. Also, due to thermal effects, undesirable charge may be generated which is known as *dark current*. To overcome these defects, the charge is recirculated around the shift register for refreshing.

### 11.10.1 Basic Concept of CCD

Consider a *p*-type silicon substrate covered with a thin oxide layer and closely spaced metallic electrodes, as shown in Fig. 11.40. Each metallic electrode (gate) and the substrate form a MOS capacitor which can store charge. If a positive voltage is applied at a gate electrode, a depletion region is formed in the substrate immediately under the metallic electrode. This happens due to the repulsion of free holes in the substrate because of the positive voltage at the gate electrode. These holes are driven downward away from the oxide layer and consequently immobile negative ions are exposed and a depletion region comes into existence. In Fig. 11.40, a positive voltage  $V_1$  is applied at the  $G_1$  gate and the other two gates are held at the same potential as the substrate. The depletion region is indicated below the gate  $G_1$ . This plot also represents the potential-energy barrier (well) for electrons, which are the minority charge carriers. Now, if a packet of negative charge is injected into the depletion region, these charges can move freely within the well, but cannot penetrate the potential-energy walls of the well. This means that as long as the voltage  $V_1$  is present, the negative charge will be held (trapped) there.

Table 11.16 Available CMOS FIFO Memory ICs

IC No.	Organisation No. of bits	Power supply voltage	Speed	Architecture	Output compatibility	Type
72V01	512 × 9	3.3 V	15 ns	Unidirectional	3.3 V LVTTL	Asynchronous
72V02	1 K × 9	3.3 V	15 ns	Unidirectional	3.3 V LVTTL	Asynchronous
72V05	8 K × 9	3.3 V	15 ns	Unidirectional	3.3 V LVTTL	Asynchronous
72V06	16 K × 9	3.3 V	15 ns	Unidirectional	3.3 V LVTTL	Asynchronous
7207	32 K × 9	5 V	30 ns	Unidirectional	TTL	Asynchronous
7208	64 K × 9	5 V	30 ns	Unidirectional	TTL	Asynchronous
72V81	512 × 9 × 2	3.3 V	15 ns	Dual FIFO	3.3 V LVTTL	Asynchronous
72V85	8 K × 9 × 2	3.3 V	15 ns	Dual FIFO	3.3 V LVTTL	Asynchronous
72105	256 × 16	5 V	25 ns	Parallel/Serial FIFO	TTL	Asynchronous
72115	512 × 16	5 V	25 ns	Parallel/Serial FIFO	TTL	Asynchronous
72125	1 K × 16	5 V	25 ns	Parallel/Serial FIFO	TTL	Asynchronous
72V201	256 × 9	3.3 V	10 ns	Unidirectional	3.3 V LVTTL	Synchronous
72V211	512 × 9	3.3 V	10 ns	Unidirectional	3.3 V LVTTL	Synchronous
72V251	8 K × 9	3.3 V	10 ns	Unidirectional	3.3 V LVTTL	Synchronous
72421	64 × 9	5 V	10 ns	Unidirectional	TTL	Synchronous
72241	4 K × 9	5 V	10 ns	Unidirectional	TTL	Synchronous
72251	8 K × 9	5 V	10 ns	Unidirectional	TTL	Synchronous
72V801	256 × 9	3.3 V	10 ns	Dual FIFO	3.3 V LVTTL	Synchronous
72V841	4 K × 9	3.3 V	10 ns	Dual FIFO	3.3 V LVTTL	Synchronous
72V851	8 K × 9	3.3 V	10 ns	Dual FIFO	3.3 V LVTTL	Synchronous
72V3682	16 K × 36 × 2	3.3 V	10 ns	Bi-directional	3.3 LVTTL	Synchronous
72V3692	32 K × 36 × 2	3.3 V	10 ns	Bi-directional	3.3 LVTTL	Synchronous
72V36102	64 K × 36 × 2	3.3 V	10 ns	Bi-directional	3.3 LVTTL	Synchronous
72605	256 × 18 × 2	5 V	20 ns	Bi-directional	TTL	Synchronous
72615	512 × 18 × 2	5 V	20 ns	Bi-directional	TTL	Synchronous

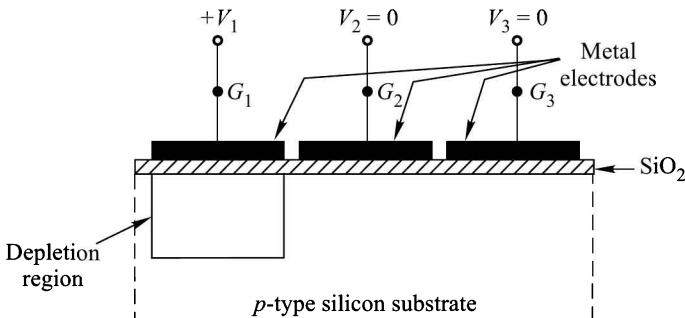


Fig. 11.40 Basic CCD Structure

In a conventional capacitor, the charges are held on conducting plates, whereas in a MOS device the charges are held on a conductor and in the depletion region under the conductor. The stored charge can be moved from left to right down the channel by applying voltages at the gates in a proper sequence. We assume that a logic 1 is stored when a negative charge is held in the depletion region and a logic 0 is stored when the depletion region is empty. This type of operation makes it possible to make long shift registers using these devices.

### 11.10.2 Operation of CCD

A portion of the structure of a 4-phase charge-coupled device (CCD) shift register, along with the charge transfer plots, is shown in Fig. 11.41a. The four clock waveforms  $\phi_1$ ,  $\phi_2$ ,  $\phi_3$ , and  $\phi_4$  used to drive the circuit are shown in Fig. 11.41b. An array of four adjacent electrodes driven by the 4-phase clock constitutes a single dynamic FLIP-FLOP. The 4-phase arrangement is required to give this dynamic FLIP-FLOP the operating features of a master-slave FLIP-FLOP, which allows shift register operation and assures that the data move in only one direction. The whole device is a long array of MOS devices in which all the  $\phi_1$  electrodes are connected together. A similar arrangement exists for the  $\phi_2$ ,  $\phi_3$ , and  $\phi_4$  electrodes.

During the time interval  $t_1$  only  $\phi_1$  is at a positive voltage, so that depletion regions are formed only under  $\phi_1$ . The charge indicated in the depletion region under  $\phi_1$  is injected either from an outside source or from the preceding  $\phi_4$  gate.

During the interval  $t_2$ , the depletion regions under  $\phi_1$  gates persist while new depletion regions are formed under  $\phi_3$  gates because the clock  $\phi_3$  becomes positive. In the interval  $t_3$ , the clock  $\phi_2$  also becomes positive while the clocks  $\phi_1$  and  $\phi_3$  are held positive. Therefore, depletion regions are generated extending from the  $\phi_1$  gates to the  $\phi_3$  gates. As a result, the charge is now able to spread throughout the extended region. During the interval  $t_4$ , the clock  $\phi_1$  becomes 0 thereby eliminating the depletion regions under the  $\phi_1$  gates. Similarly, during the interval  $t_5$ , the depletion regions under  $\phi_2$  gates vanish and the charge (or no charge) originally under the  $\phi_1$  gates is pushed laterally to regions under the  $\phi_3$  gates.

Continuing the above logic, we observe that in the succeeding intervals  $t_6$  through  $t_8$ , and finally back to  $t_1$ , the charge under the gate  $\phi_3$  will be moved to the region under the next  $\phi_1$  gate. Hence, altogether, after eight intervals, charge (or no charge) under a  $\phi_1$  gate will shift to the next  $\phi_1$  gate.

Special arrangement must be made to inject charge into the first depletion region as required, and to detect the presence (logic 1) or absence (logic 0) of charge at the last depletion region. The injection and detection of charge must be done in synchronism with the clock waveform.

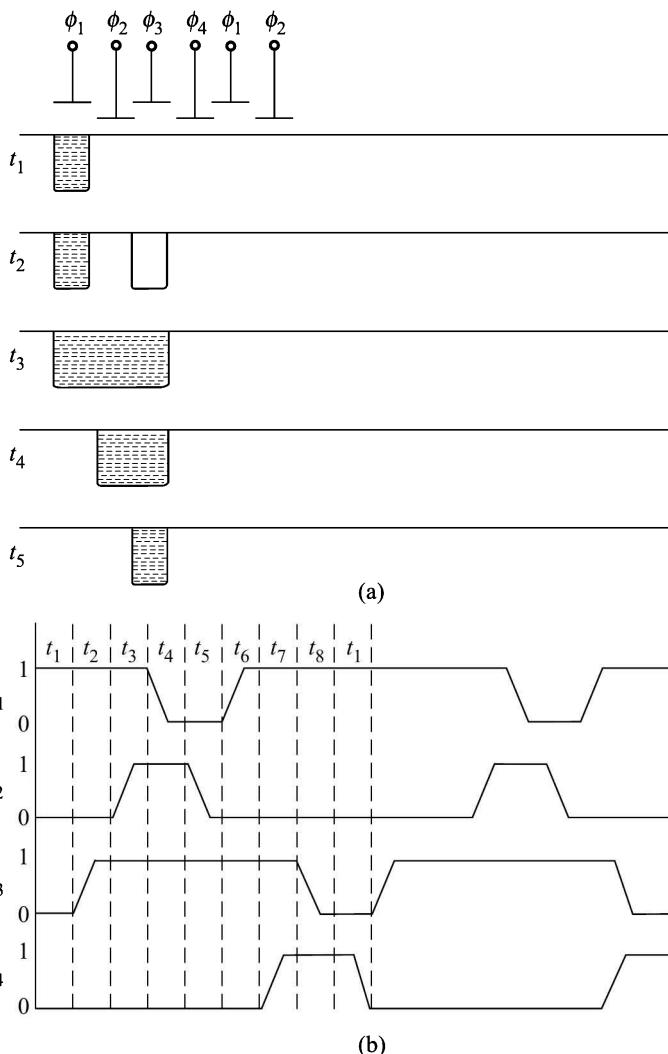


Fig. 11.41 (a) A Portion of Four-Phase CCD Structure Along with the Charge Transfer Plots and (b) Four-Phase Clock Waveform

When the charge transfer takes place down the shift register, there is some loss of charge. Therefore, it is necessary to incorporate provision for refreshing the charge at periodic intervals along the length of the CCD structure. In fact, the charge transfer efficiency is so high (~99.999%) that 100 or more FLIP-FLOPs can be cascaded before refreshing becomes necessary.

### 11.10.3 A Practical CCD Memory Device

The basic organisation of the intel 2416 CCD memory is shown in Fig. 11.42. It is a  $16\ 384 \times 1$  bit serial memory, organised as 64 independent recirculating shift registers of 256 bits each. Any one of the 64 registers

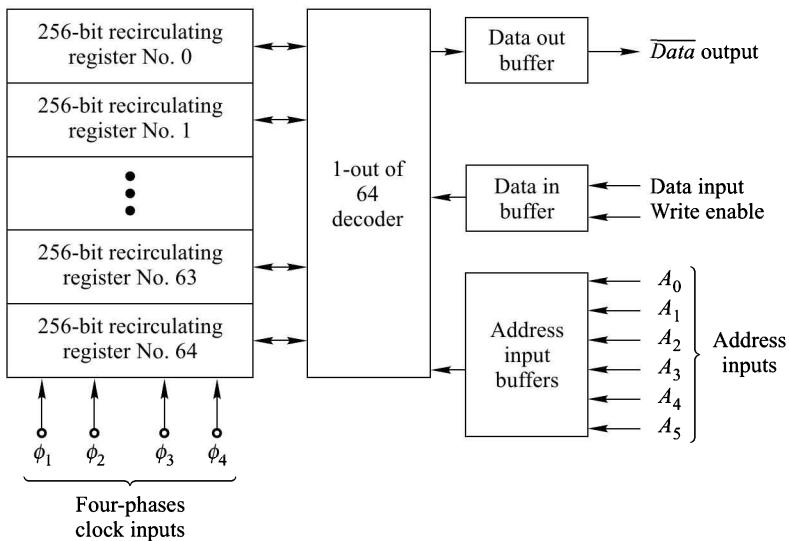


Fig. 11.42      **Basic Organisation of Intel 2416 CCD Memory**

can be accessed by applying the appropriate 6-bit code at the address inputs. The data in the shift registers is simultaneously shifted by using the 4-phase clock signals  $\phi_1$  through  $\phi_4$ . After a shift cycle, each of the 64 registers can be selected for an input/output operation by applying the appropriate 6-bit address code.

When addressed, one bit is written into or read from the memory. If the address input is fixed, then as shifting progresses, the bit positions in the addressed register will be presented serially for reading or writing. The output is open-drain which allows wired-OR connection. During the interval between shifts, we can have access to a bit from each of the registers by changing the address. The 64 bits, which can be accessed by changing address, are available on a random access basis. The 256 bits in a single register are available only in the serial mode.

In serial mode operation, access to a desired bit may require no shift or it may require upto a maximum of 256 shifts. On an average,  $256/2 = 128$  shifts are required for accessing a bit. This access time in serial operation is known as *latency* or *latency time*.

The Intel 2464 is a 65536-bit CCD memory organised in a manner similar to the 2416 chip, but with 256 independent circulating registers of 256 bits each. It has an active-low chip enable signal  $\overline{CE}$ .

The bit capacity and word lengths can be expanded by using appropriate circuitry (Probs. 11.19 and 11.20).

## SUMMARY

Semiconductor memories are invariably an essential part of any digital system. All memory devices store binary logic levels (1 and 0) in an array structure. Memory ICs are available with various sizes of array in terms of the number of words and the number of bits in the word. The number of words and the word size can be increased, if necessary, by using multiple chips. Different types of memory devices have been dealt in detail and some of the commercially available ICs have been given for each type. Various types of ROM devices, such as ROM, PROM, OTP EPROM, EEPROM, parallel and serial EEPROM have been discussed alongwith their erasing and programming techniques. All types of ROMs are non-volatile.

Both the types of RAM devices, i.e., SRAM and DRAM have been discussed starting from their basic cell. Asynchronous and synchronous operations of SRAM and DRAM have been covered in detail. One of the major applications of SRAM is in *Cache memories* because of their very high speed. In general, the SRAM devices are very much faster in comparison to DRAM devices. All types of RAMs are volatile.

Flash memory, a specific type of EEPROM, is a non-volatile memory which allows in-circuit writing. It is used in many digital systems, such as cellular phones, cameras, LAN switches, embedded controllers, memory cards, and USB flash drives etc. The flash memory ICs are available in parallel as well as serial interface.

The first-in, first-out FIFO memory devices are very useful for interfacing various types of *I/O* devices to the computers as data-rate buffers. They are available in various sizes. Their depth and width can be increased by using multiple devices. Bi-directional BiFIFO and parallel-to-serial FIFO devices have also been discussed.

## GLOSSARY

**Access time** Time required for reading or writing a memory location.

**Address** The binary code of a memory location.

**Address bus** A parallel array of conductors used for accessing a memory location.

**Address decoder** A  $n$ -line-to- $2^n$  lines decoder used to select a specific memory chip or memory location.

**Asynchronous memory** A memory device in which read and write are unclocked.

**BEDO DRAM** Burst extended data output dynamic random-access memory.

**Bidirectional bus** A bus (group of electrical lines) capable of transmitting data in both the directions.

**Bi-directional FIFO memory** A FIFO memory capable of buffering data between two systems for read and write.

**BIOS** Basic input/output system – A set of programs in ROM that interfaces the *I/O* devices in a digital system.

**Burst** A memory feature which allows read from or write at upto four locations using a single address.

**Cache memory** A high-speed memory (normally SRAM) that stores the most recently used instructions or data from the slower main memory.

**CAM (Content addressable memory)** A special purpose RAM device which can be accessed by its contents. It is also known as ‘associative memory’.

**CCD (Charge coupled device)** It is serial-access semiconductor memory device with very high bit packing density.

**Chip** A piece of silicon or other semiconductor material on which an IC is fabricated.

**Chip enable** An input control signal which when activated enables the chip.

**Chip select** An input control signal that allows the chip to be selected.

**Control bus** A bus used for handling control signals.

**Cycle time** The minimum time between successive read or write cycles in a memory.

**Data bus** A bus used for carrying data.

**Data-rate buffer** A buffer that allows two digital systems with different data rates (speeds) to communicate.

**DDR** Double data rate.

**DRAM** Dynamic random-access memory.

**Dual-port SRAM** A static random-access memory with two ports.

**Dynamic memory** A memory in which data needs to be refreshed periodically. Data is stored on MOS capacitors.

**EAROM (Electrically alterable read-only memory)** A read-only memory in which the stored words can be erased electrically. Also known as E<sup>2</sup>PROM (Electrically erasable and programmable read-only memory).

**EDO DRAM** Extended data output dynamic random-access memory.

**EEPROM** Electrically erasable and programmable read-only memory.

**EPROM (Erasable programmable read-only memory)** A read-only memory which can be erased by exposure to ultraviolet light, and then reloaded with new information.

**Erasable memory** A memory contents of which can be erased.

**First-in, First-out (FIFO) memory** A static random-access memory array with two ports in which data words are read out in the same order in which they were written in.

**Flash memory** A specific type of EEPROM, a non-volatile memory, which allows in-circuit writing.

**FPM DRAM** Fast page mode dynamic random-access memory.

**Fusible link** A link of nichrome, or some other materials, used in PROMs and other programmable devices which can be either burnt or kept intact while storing 0s and 1s in these devices.

**Non-erasable memory** A memory contents of which can not be erased, such as a ROM.

**Non-volatile memory** A memory that does not lose its contents when power is turned off, such as ROMs.

**Non-volatile RAM** A flash-memory is also referred to as a non-volatile random-access memory.

**OTP** One time programmable.

**Parallel-to-serial FIFO memory** A FIFO memory with parallel input port and serial output port.

**Programming (of Memory)** To store the desired data in a programmable memory which is of read-only type.

**PROM (Programmable read-only memory)** A read-only memory that can be programmed only once by the user by selectively opening the fusible links.

**PROM programmer** An equipment that is used to program a programmable ROM.

**RAM (Random-access memory)** A read-and-write semiconductor memory in which any memory location can be accessed for reading or writing at random.

**Refresh** The act of restoring the data in a dynamic memory.

**ROM (Read-only memory)** A semiconductor memory in which data can only be read.

**SDRAM** Synchronous DRAM

**Semiconductor memory** A memory fabricated using semiconductor material.

**Serial-access memory** A memory in which the access time of a stored bit or word depends on its location in the memory.

**Serial EEPROM** An EEPROM device with serial I/O.

**Serial flash memory** A flash memory with serial I/O.

**Static RAM (SRAM)** A random-access memory in which read out and write into are not clocked.

**Volatile memory** A memory that loses its contents when power is turned off.

## REVIEW QUESTIONS

- 11.1 Information in a memory chip is stored in \_\_\_\_\_ form.
- 11.2 The maximum number of bytes which can be stored in a memory of size  $1024 \times 8$  is \_\_\_\_\_.
- 11.3 The number of address lines required in a memory of  $128 K \times 8$  is \_\_\_\_\_.
- 11.4 While specifying the memory size, the letter *K* stands for \_\_\_\_\_.
- 11.5 An EPROM is a \_\_\_\_\_ access memory.
- 11.6 A shift register is a \_\_\_\_\_ memory.
- 11.7 The contents of location *0A00H* of RAM is 01001100, its contents after a read operation will be \_\_\_\_\_.
- 11.8 The contents of location *BAC0H* of EPROM is 11100101, its contents after a read operation will be \_\_\_\_\_.
- 11.9 The number of IC chips of memory size  $1024 \times 4$  required to have  $16 K \times 8$  memory will be \_\_\_\_\_.
- 11.10 The contents of location *FEE0H* of a RAM is *BEH*, its contents after a write operation is performed with *ECH* data at the data bus will be \_\_\_\_\_.
- 11.11 The access-time of a RAM is 10 ns, the minimum time which must elapse between two read operations will be \_\_\_\_\_.
- 11.12 An EAROM is \_\_\_\_\_ erasable.
- 11.13 An EPROM is erased by \_\_\_\_\_.
- 11.14 A dynamic RAM is fabricated using \_\_\_\_\_ technology.
- 11.15 CAM stands for \_\_\_\_\_.
- 11.16 A serial EEPROM has \_\_\_\_\_ input/output.
- 11.17 An SRAM with two ports is known as a \_\_\_\_\_ SRAM.
- 11.18 A FIFO memory has \_\_\_\_\_ ports.
- 11.19 A serial flash memory has \_\_\_\_\_ input/output
- 11.20 A FIFO memory can be used as a \_\_\_\_\_ buffer.
- 11.21 A \_\_\_\_\_ FIFO memory can be used for two way communication between two digital systems operating at different speeds.
- 11.22 A serial EEPROM IC has \_\_\_\_\_ number of pins than a parallel EEPROM IC.
- 11.23 The burst feature in a synchronous SRAM \_\_\_\_\_ its speed.
- 11.24 The data rate of a synchronous SRAM operating at 200 MHz is \_\_\_\_\_ MHz.

## PROBLEMS

- 11.1 For a memory with *M* words storage, find the number of pins required for addressing and the address range in binary format for each of the following cases:

- (a)  $M = 4$
- (b)  $M = 16$
- (c)  $M = 64$
- (d)  $M = 256$
- (e)  $M = 1024 = 1 \text{ K}$
- (f)  $M = 2048 = 2 \text{ K}$
- (g)  $M = 64 \text{ K}$
- (h)  $M = 1 \text{ M}$

**11.2** Express the address range for each of the cases of Prob. 11.1 in

- (a) Hexadecimal format.
- (b) Octal format.

**11.3** The access time and cycle time for a set of memories are given in Table 11.17. Determine the maximum rate at which data can be accessed in each case.

Table 11.17

Memory	Access time ns	Cycle time ns
A	1500	1500
B	300	580
C	450	450
D	200	200
E	60	60
F	800	800

**11.4** The block diagram of a  $1 \text{ K} \times 4$  bit static RAM is shown in Fig. 11.43. Find the number of RAM chips required, if any, to obtain.

- (a)  $4096 \times 4$  bit RAM
- (b)  $1024 \times 8$  bit RAM
- (c)  $16 \text{ K} \times 8$  bit RAM.

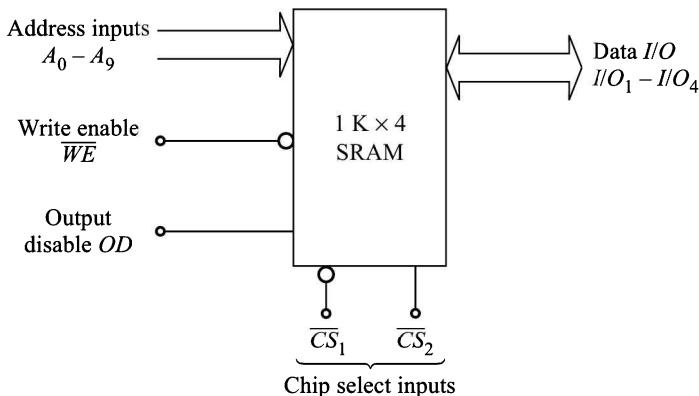


Fig. 11.43 Block Diagram of a  $1 \text{ K} \times 4$  SRAM

**11.5** Implement the RAMs of Prob. 11.4.

**11.6** The block diagram of Intel 2716 2K  $\times$  8 EPROM is shown in Fig. 11.44. Find the number of 2716 and other ICs required to obtain.

- (a) 4 K bytes of ROM
- (b) 2 K  $\times$  16 ROM
- (c) 4 K  $\times$  16 ROM

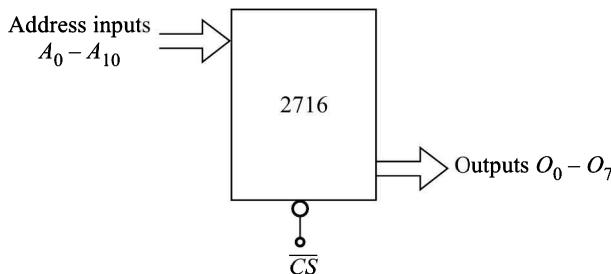


Fig. 11.44 **Block Diagram of 2716 EPROM**

**11.7** Implement the ROMs of Prob. 11.6.

**11.8** Explain the following:

- (a) Linear selection addressing.
- (b) Coincident selection addressing.

**11.9** Figure 11.45 shows the block diagram of the asynchronous SRAM 65C256. Design an SRAM of size 64 K  $\times$  16 bits.

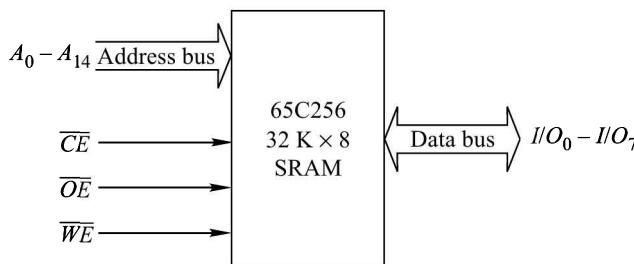
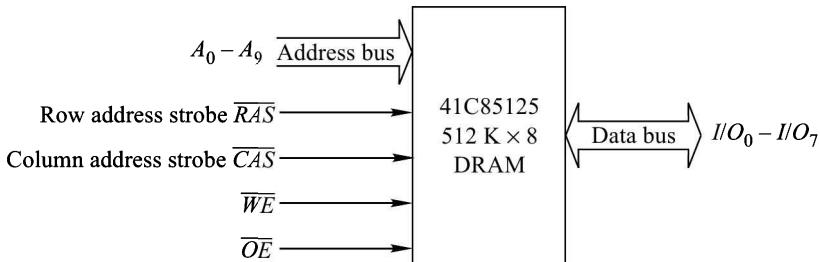


Fig. 11.45 **Block Diagram of 65C256 SRAM**

**11.10** Explain the following:

- (a) Late-write SRAM
- (b) ZBT or ZEROSB SRAM
- (c) Flow-through SRAM
- (d) Pipe-lined SRAM

**11.11** Figure 11.46 shows the block diagram of 41C85125 DRAM. It is a FPM DRAM with 1024 row addresses. Design a DRAM of size 512 K  $\times$  16.

Fig. 11.46 **Block Diagram of 41C85125 DRAM**

- 11.12** Verify the operations given in Table 11.11.
- 11.13** Design a  $16 \times 2$  CAM using two  $8 \times 2$  CAM chips.
- 11.14** Design an  $8 \times 8$  CAM using four  $8 \times 2$  CAM chips.
- 11.15** Design a  $16 \times 8$  CAM using  $8 \times 2$  CAM chips.
- 11.16** It is desired to find the maximum valued number stored in a CAM of size  $16 \times 8$ . Suggest a suitable method. Compare this method with the method used if a RAM is used instead of a CAM.
- 11.17** Repeat Prob. 11.16 to Find the minimum valued number.
- 11.18** It is desired to design a memory system for storing information which is not already stored in it. This type of memory is known as *learning memory*. Will you prefer to use RAM or CAM for this purpose, why?
- 11.19** Suggest a suitable arrangement for expanding the bit capacity of CCDs.
- 11.20** Suggest a suitable arrangement for expanding the word length of CCDs.

## CHAPTER 12

# PROGRAMMABLE LOGIC DEVICES

### 12.1 INTRODUCTION

The combinational and sequential digital circuits have been discussed in earlier chapters. Various ICs for performing basic digital operations and other functions, such as multiplexers, demultiplexers, adders, comparators, code converters, shift registers and counters, etc. have also been discussed. These ICs are referred to as *fixed-function* ICs, i.e. each one of them performs a specific, fixed function. These devices are designed by their manufacturers and are manufactured in large quantities to meet the needs of a wide variety of applications and are readily available.

To design a circuit, a designer can select from the available ICs most appropriate for the circuit, usually working from a block diagram design concept. The design may have to be modified to meet the special requirements of these devices. The advantages of this method are:

1. Low development cost,
2. Fast turn around of designs, and
3. Relatively easy to test the circuits

Some of the disadvantages of this method are:

1. Large board space requirements,
2. Large power requirements,
3. Lack of security, i.e. the circuits can be copied by others, and
4. Additional cost, space, power requirements, etc. required to modify the design or to introduce more features.

To overcome the disadvantages of designs using fixed-function ICs, *application specific integrated circuits* (ASICs) have been developed. The ASICs are designed by the users to meet the specific requirements of a circuit and are produced by an IC manufacturer (*foundry*) as per the specifications supplied by the user. Usually, the designs are too complex to be implemented using fixed-function ICs.

The advantages of this method are:

1. Reduced space requirement,
2. Reduced power requirement,

3. If produced in large volumes, the cost is considerably reduced,
4. Large reduction in size through the use of high level of integration, and
5. Designs implemented in this form are almost impossible to copy.

The disadvantages of this method are:

1. initial development cost may be enormous, and
2. testing methods may have to be developed which may also increase the cost and effort.

Another approach which has the advantages of both the above methods is the use of *programmable logic devices* (PLDs). A programmable logic device is an IC that is user configurable and is capable of implementing logic functions. It is a VLSI chip that contains a ‘regular’ structure and allows the designer to customize it for any specific application, i.e. it is programmed by the user to perform a function required for his application.

PLDs have the following advantages of fixed function ICs:

- (i) Short design cycle
- (ii) Low development cost

The advantages over fixed-function ICs are:

- (i) Reduction in board space requirements
- (ii) Reduction in power requirements
- (iii) Design security
- (iv) Compact circuitry
- (v) Higher switching speed

PLDs have many of the advantages of ASICs as given below:

- (i) Higher densities
- (ii) Lower quantity production costs
- (iii) Design security
- (iv) Reduced power requirement
- (v) Reduced space requirement

The PLDs allow designers more flexibilities to experiment with designs because these can be reprogrammed in seconds. The design deficiencies and modifications etc. can be carried out in short time thereby reducing the possibility of huge cost over-runs.

PLDs are also useful for prototyping ASIC designs since foundry produced ASICs may require months of costly development.

Because of various advantages of PLDs mentioned above, a large number of PLDs have been produced by IC manufacturers with variety of flexibilities and options available for a circuit designer and have become very popular. The architecture and various other features of PLDs such as ROMs, programmable logic arrays (PLAs), programmable array logic (PAL), simple programmable logic devices (SPLDs), complex programmable logic devices (CPLDs), and field-programmable gate arrays (FPGA) have been discussed in this chapter. The use of these devices require changes in the traditional design methods, although the basic concepts remain the same.

## 12.2 ROM AS A PLD

Read-only memories (ROMs) have been discussed in Section 11.5. A read-only memory is basically a combinational circuit and can be used to implement a logic function. A ROM of size  $M \times N$  has  $M$  number

of locations and  $N$  number of bits can be stored at each location. The number of address inputs is  $P$ , where  $2^P = M$  and the number of data output lines is  $N$ . It can also be considered as a logic device with  $P$  inputs and  $N$  outputs as shown in Fig. 12.1.

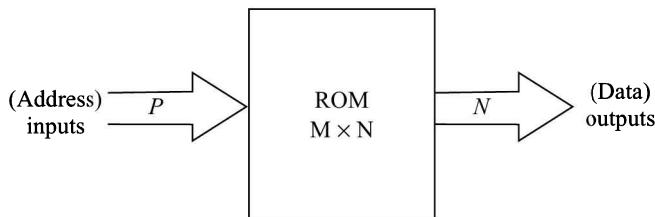


Fig. 12.1     **ROM as a Combinational Circuit**

The 16-bit ROM array shown in Fig. 11.8 has four inputs and one output, i.e.  $M = 16$ ,  $N = 1$  and  $P = 4$ . The bit pattern stored, as given in Table 11.4, can be considered as truth table with  $A_3, A_2, A_1, A_0$  as 4-bit input and  $Y$  as the output which is same as the bit stored. The logic function corresponding to this is

$$Y = \Sigma m(0, 6, 9, 12, 13, 15)$$

In general, a  $P$  variable,  $N$  outputs logic function can be implemented using a ROM of size  $2^P \times N$  since all the possible  $2^P$  minterms are effectively generated as is clear from the above discussion.

If a mask programmable ROM is used, the user can specify the bit pattern to be stored according to the requirements of the logic function, whereas the user can himself program it in case of PROM, EPROM and E<sup>2</sup>PROM. Since programmable ROMs can be used for logic design, therefore, it is also referred to as a programmable logic device (PLD).

The advantages of using ROM as a programmable logic device are:

1. Ease of design since no simplification or minimization of logic function is required.
2. Designs can be changed, modified rapidly.
3. It is usually faster than discrete SSI/MSI circuit.
4. Cost is reduced.

There are few disadvantages also of ROM-based circuits, such as non-utilization of complete circuit, increased power requirement, and enormous increase in size with increase in number of input variables making it impractical.

## 12.3 PROGRAMMABLE LOGIC ARRAY

A programmable logic device (PLD) usually consists of programmable array of logic gates and interconnections with array inputs and outputs connected to the device pins through fixed logic elements, such as inverting/non-inverting buffers and FLIP-FLOPs. The logic gates used may be two-level AND-OR, NAND-NAND or NOR-NOR configuration. In some cases, AND-OR-EX-OR configuration is also used. Basically, there are two types of PLDs, *programmable logic array* (PLA) and *programmable array logic* (PAL). These are suitable for implementing logic functions in SOP form.

A PLA consists of two-level AND-OR circuits on a single chip. The number of AND and OR gates and their inputs are fixed for a given PLA chip. The AND gates provide the product terms, and the OR gates logically sum these product terms and thereby generate a SOP expression. It has  $M$  inputs,  $n$  product terms, and  $N$  outputs with  $n < 2^M$ , and can be used to implement a logic function of  $M$  variables with  $N$  outputs. Since all

of the possible  $2^M$  minterms are not available, therefore, logic minimization is required to accommodate a given logic function.

The internal architecture of a PLA is shown in block diagram form in Fig. 12.2.

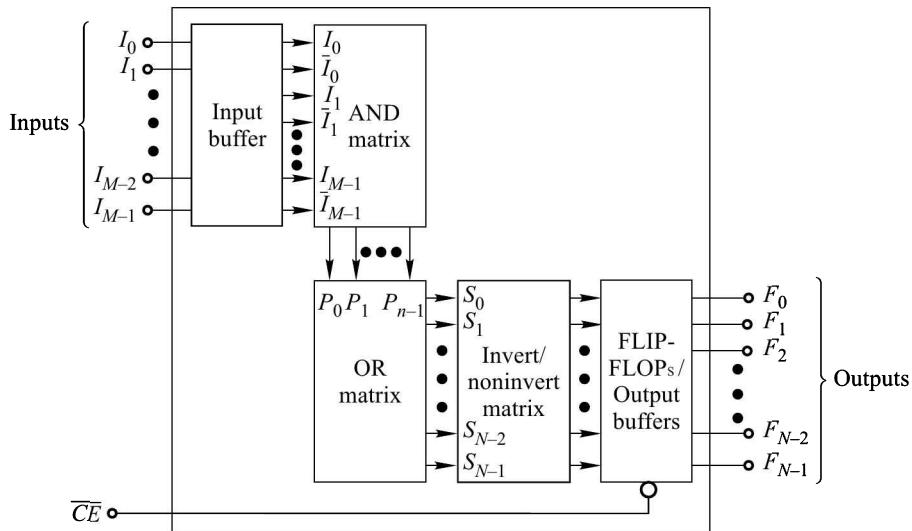


Fig. 12.2     *Block Diagram of a PLA Device*

### 12.3.1 Input Buffer

The buffer circuits at the input are required to limit loading of the sources that drive the inputs. It produces inverted as well as non-inverted inputs at the output as shown in Fig. 12.3 for one input. Similar buffers are there for each of the  $M$  inputs.

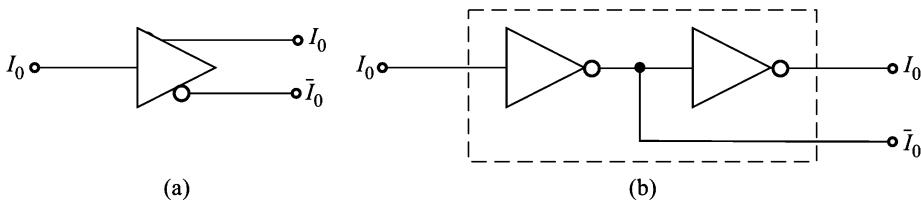


Fig. 12.3     *An Input Buffer*

### 12.3.2 AND Matrix

An AND matrix is used to form product terms. A typical AND matrix is shown in Fig. 12.4. It has  $n$  AND gates with outputs  $P_0$  through  $P_{n-1}$  and  $2M$  inputs ( $I_0$  through  $I_{M-1}$  and  $\bar{I}_0$  through  $\bar{I}_{M-1}$ ) for each AND gate. This shows that each AND gate has all the input variables in complemented and uncomplemented form. There is a nichrome fuse link in series with each diode. All the links are intact in an unprogrammed PLA device. Each AND gate generates one product term which is given by

$$P = I_0 \cdot \bar{I}_0 \cdot I_1 \cdot \bar{I}_1 \dots I_{M-1} \cdot \bar{I}_{M-1}$$

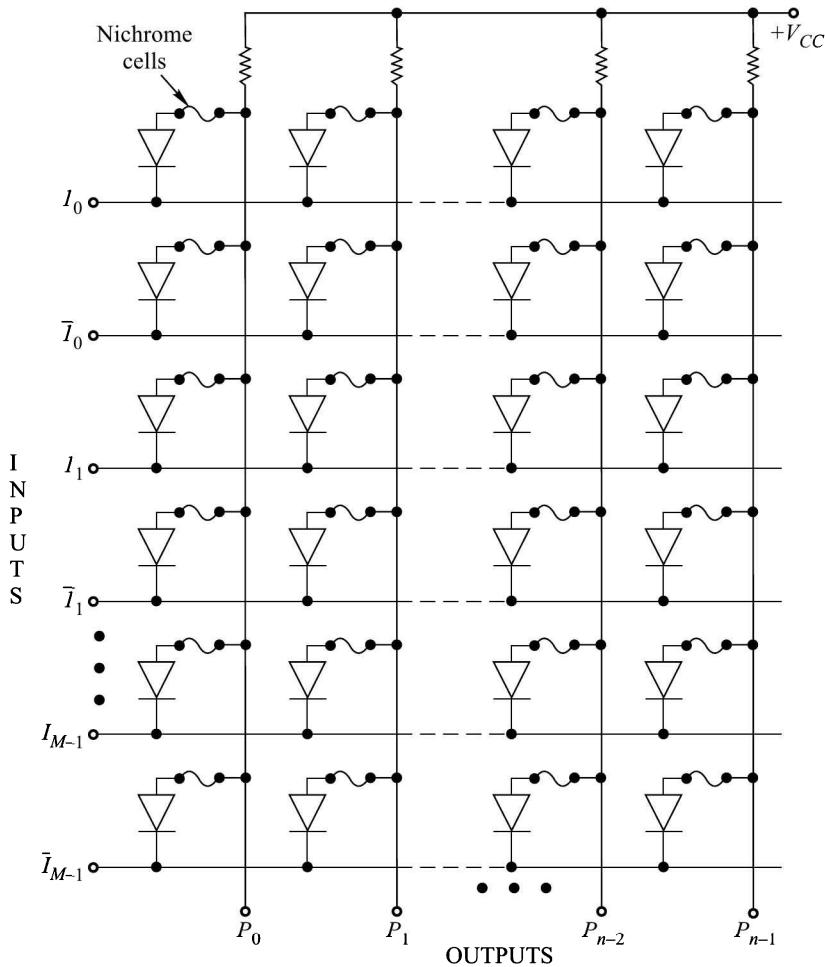


Fig. 12.4 An AND Matrix

and it is logic 0 in an unprogrammed device. For generating a required product term the unwanted links are opened through the method of programming by a programmer device in case of a field-programmable logic device (FPLA). A section of the AND matrix for  $P_0$  output is shown in Fig. 12.5a and its equivalent logic gate representation is shown in Fig. 12.5b. A similar arrangement exists for each of the other outputs.

### Example 12.1

In Fig. 12.5, if all the links except the ones present for inputs  $I_0$ ,  $\bar{I}_2$ ,  $\bar{I}_3$ , and  $I_6$  are opened, find the product term  $P_0$ .

### Solution

For an open link, the input to the AND gate is logic 1, whereas for a closed link the corresponding input to the AND gate is same as the voltage applied at that input, therefore,

$$P_0 = I_0 \cdot \bar{I}_2 \cdot \bar{I}_3 \cdot I_6$$

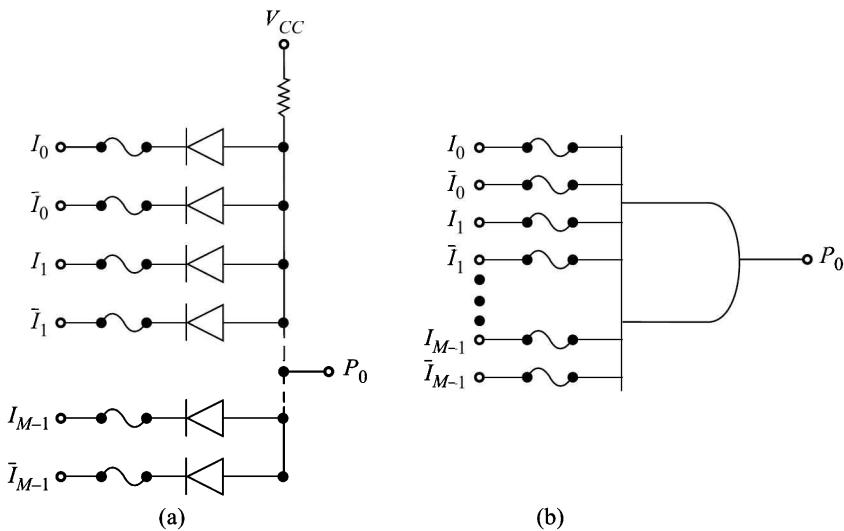
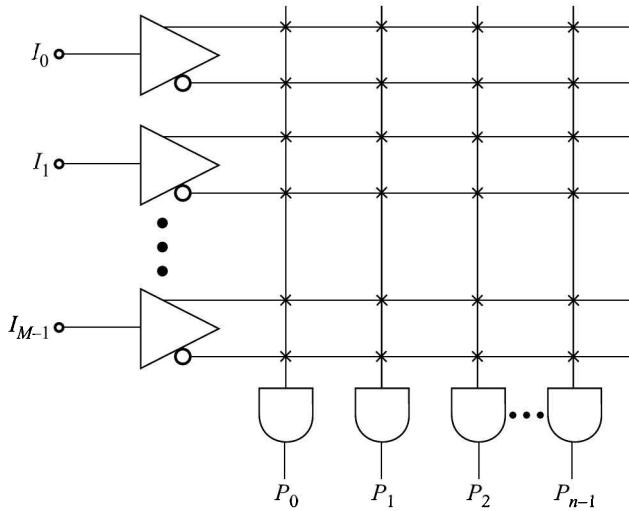
Fig. 12.5 *A Section of the AND Matrix*

Figure 12.6 illustrates a convenient method of showing the input buffers and the AND matrix with the interconnections marked as  $X$ s. Each AND gate has  $2M$  inputs which is indicated by a single line in the figure. When an array is programmed to implement a particular logic function, the desired interconnections are left with  $X$  marks and the unwanted interconnections without  $X$  marks.

Fig. 12.6 *Representation of Input Buffers and AND Matrix*

### 12.3.3 OR Matrix

The OR matrix is used to produce the logical sum of the product term outputs of the AND matrix. Figure 12.7 shows an OR matrix using RTL circuitry. An OR gate consists of parallel connected transistors with a common

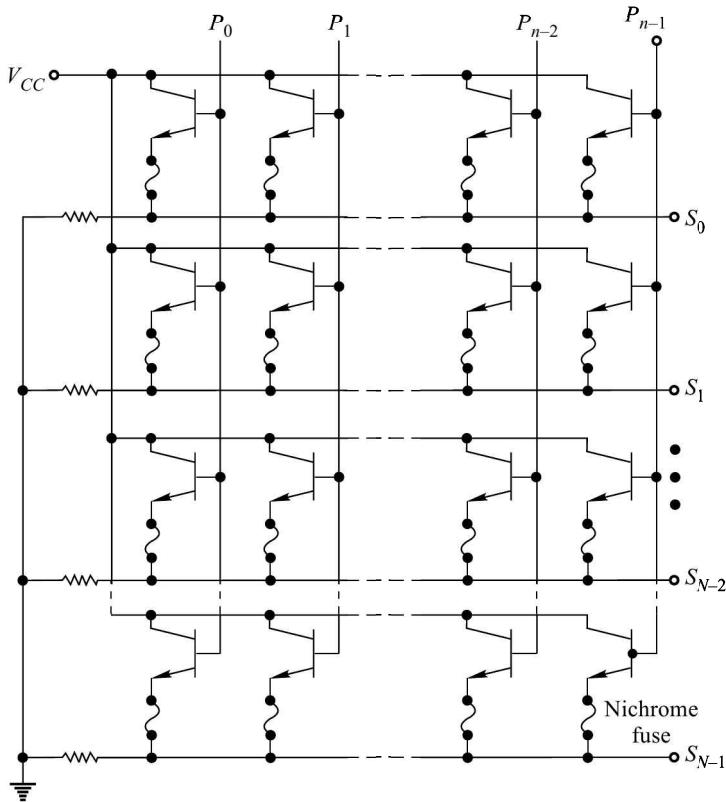


Fig. 12.7 An OR Matrix

emitter load. The outputs of the OR matrix are obtained at  $S_0$  through  $S_{N-1}$ . Consider the output  $S_0$  when all the fuse links are intact, which is given by

$$S_0 = P_0 + P_1 + \dots + P_{n-1}$$

The required sum terms can be generated by opening the unwanted fuse links. For example, if all the fuse links, except the ones for the product terms  $P_0$  and  $P_1$ , are blown off for the output  $S_0$ , then

$$S_0 = P_0 + P_1$$

Thus an OR matrix can be programmed by opening the unwanted fuse links, which effectively makes logic level 0 at the corresponding OR gate inputs.

Figure 12.8 gives logic symbol for one of the OR gates, and Fig. 12.9 illustrates the relevant portion of the PLA containing OR matrix in a way similar to Fig. 12.6.

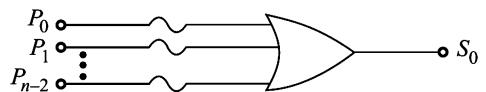
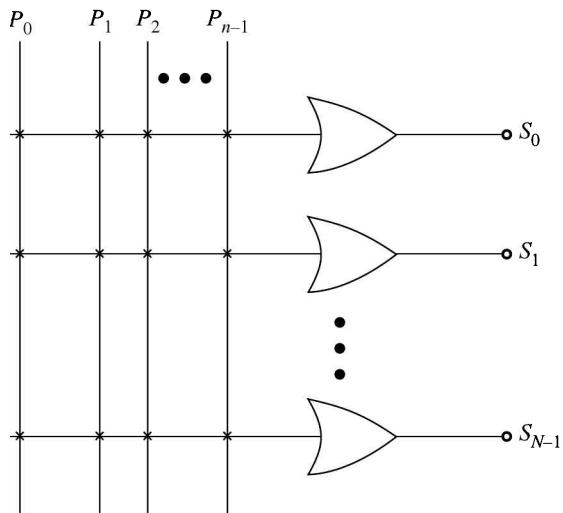


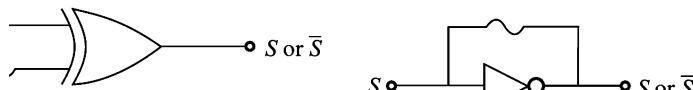
Fig. 12.8 Logic Symbol of a Section of OR Matrix

#### 12.3.4 INVERT/NON-INVERT Matrix

This is a programmable buffer that can be set for INVERTING or NON-INVERTING operation corresponding to active-low or active-high output, respectively. Typical circuits for this operation are shown in Fig. 12.10.

Fig. 12.9 *Representation of OR Matrix*

ate, if the fuse is intact, the output is  $S$ , whereas the output is  $\bar{S}$  if the fuse is cut in case of Fig. 12.10b is  $S$  or  $\bar{S}$  depending upon whether the fuse is intact or



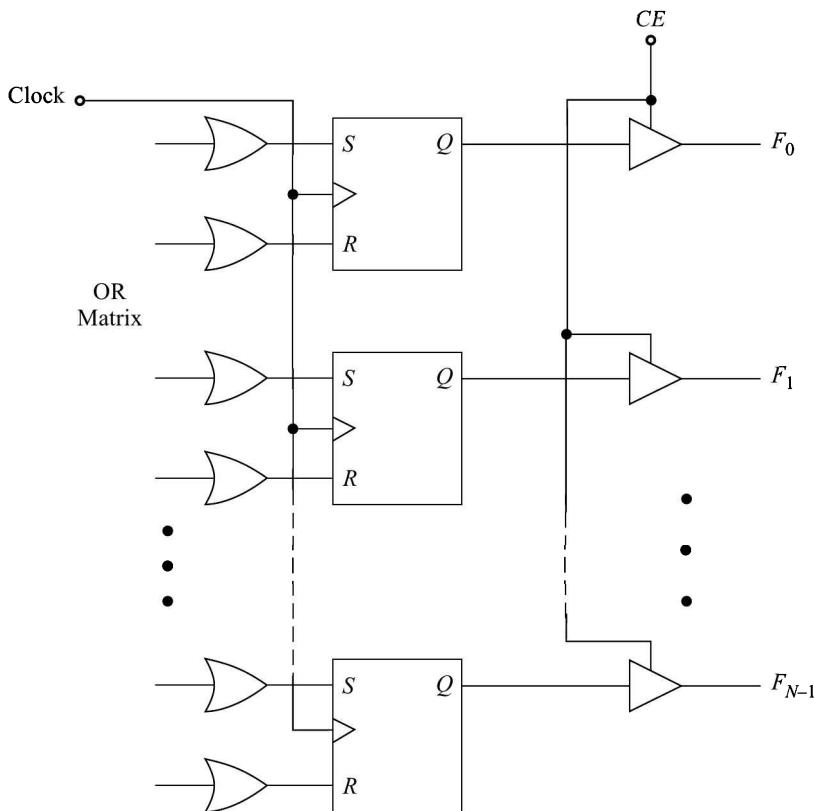


Fig. 12.12 *A Section of PLA with FLIP-FLOPs in the Output*

### 12.3.7 Programming the PLA

A PLA device is to be programmed for the desired input–output relationship similar to the programming of ROMs. For a mask-programmable PLA device, the data pattern is to be specified by the customer. The appropriate masks are designed by the manufacturers and the data pattern are built in during the manufacturing process.

An FPLA has all its nichrome fuse links intact at the time of manufacturing. The unwanted links are electrically open circuited during programming. The links to be opened are accessed by applying voltages at the inputs and outputs of the device. The FPLAs are not reprogrammable.

### 12.3.8 Expanding PLA Capacity

Some applications require a capacity that exceeds the capacity of a single PLA. The required capacity can be achieved by suitably connecting several identical devices together. For increasing the number of outputs, the inputs of two or more devices are to be connected individually in parallel. This connection does not change the number of inputs and the product terms.

For increasing the number of product terms keeping the number of inputs and outputs unchanged, the inputs and outputs of two or more devices are to be individually connected in parallel.

The number of inputs can be increased by making the connections as shown in Fig. 12.13. This circuit has  $(M+Q)$  inputs and  $N$  outputs. This connection also increases the number of product terms. The number of product terms is  $P \times (2^Q - 1)$ . The outputs are allowed to be connected together for the devices with passive pull-up only.

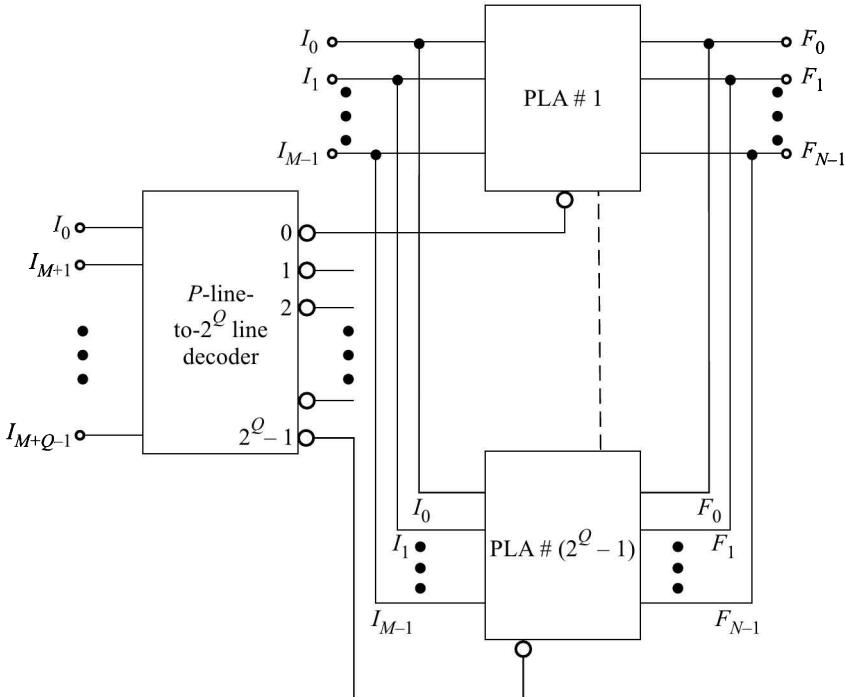


Fig. 12.13     *Expanding the Input Word Length of PLA*

### 12.3.9 Applications of PLAs

The PLAs can be used to implement combinational and sequential logic circuits. For the design of combinational circuits, PLAs with output circuit as shown in Fig. 12.10 or Fig. 12.11 are used, whereas the devices having FLIP-FLOPs in the output circuit as shown in Fig. 12.12 are required for the design of sequential circuits.

The following steps can be used for implementing combinational logic functions:

1. Prepare the truth table.
2. Write the Boolean equations in SOP form.
3. Simplify the equations to obtain minimum SOP form. The main criterion is to minimize the number of product terms.
4. Determine the input connections of AND matrix to generate the required product terms.
5. Determine the input connections of OR matrix to generate the required sum terms.
6. Determine the connections required for INVERT/NON-INVERT matrix to set the active logic levels of the outputs.
7. Program the PLA.

**Example 12.2**

Design a 4-input, 5-output combinational circuit using PLS100 PLA. The input variables are  $A$ ,  $B$ ,  $C$ , and  $D$ .

$$Y_1 = \Sigma m(0, 3, 5, 6, 9, 10, 12, 15)$$

$$Y_2 = \Sigma m(0, 1, 2, 3, 11, 12, 14, 15)$$

$$Y_3 = \Sigma m(0, 4, 8, 12)$$

$$Y_4 = \Sigma m(0, 2, 3, 5, 7, 8, 12, 13)$$

$$Y_5 = \Sigma m(0, 1, 3, 4, 5, 6, 11, 13, 14, 15)$$

**Solution**

The PLS 100 PLA device (Fig. 12.14) has 16 inputs, 8 outputs and 48 product terms. The output is through EX-OR gate, for controlling the polarity of the output, and a 3-state buffer enabled through pin 19 of the chip. A low signal

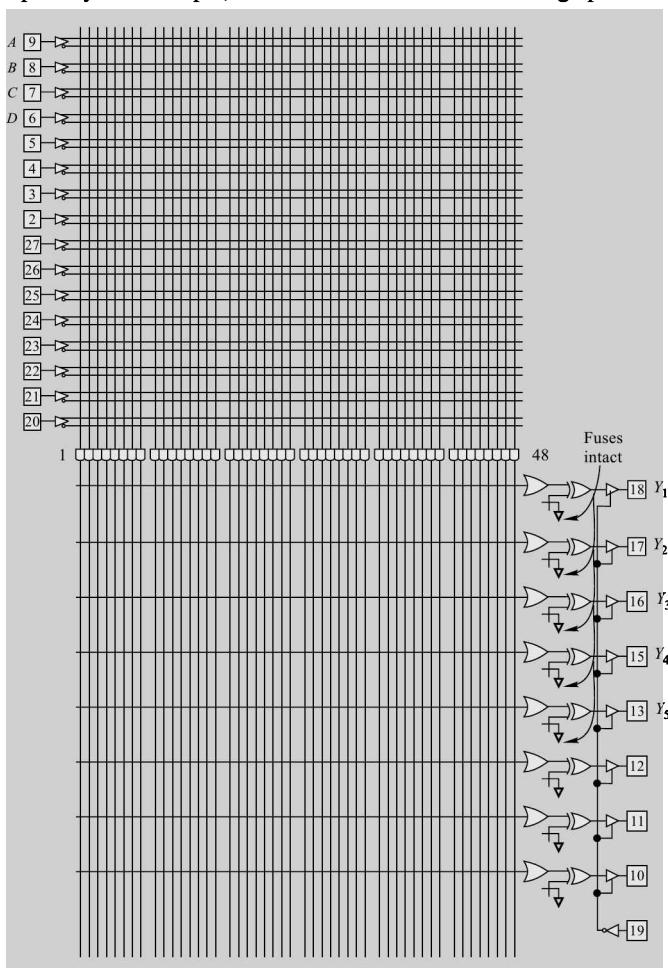


Fig. 12.14 PLS Device Programmed for Ex. 12.2

is required to be applied to this pin for enabling the output buffers. The fuses of EX-OR gates are to be kept intact to maintain these inputs at ground (0) level for active-high outputs.

The functions  $Y_1 - Y_5$  are given in minterm form which are 16 in number. This requires only 16 product terms (or 16 AND gates) which are available in the device and, therefore, no simplification or minimisation is required.

The inputs  $A, B, C$ , and  $D$  are applied at pins 9, 8, 7 and 6, respectively, and the AND gates are numbered from 1 to 48 starting from the left most gate.

Gates 1 to 16 are used for generating 0 to 15 minterms. The relevant  $X$  are indicated in Fig. 12.14 in the rows corresponding to the inputs. The output pins used for the outputs  $Y_1$  to  $Y_5$  are 18, 17, 16, 15 and 13, respectively. The relevant  $X$  are indicated in the figure. One input of each of the output EX-OR gate is connected to ground through a fuse.

The design of sequential circuits follow the same method as discussed in chapter 8 and then the PLA is configured accordingly.

### Example 12.3

Design a 4-bit UP/DOWN counter with direction control  $M$ . Assume  $M=0$  for UP counting and  $M=1$  for DOWN counting.

#### Solution

The sequential PLS 105 PLA device can be used for this. It has 14 FLIP-FLOPs, 6 of which are buried within the device and are not available for outputs. The outputs of these 6 FFs are fed back to the AND array. The remaining 8 FFs are associated with the device outputs and are not used for feedback to the AND array. Pin 1 is used for clock input and 19 for providing preset input ( $P$ ) to the FLIP-FLOPs and enable input to the output buffers.

Table 12.1 gives the count sequence and the inputs required for the FFs. The K-maps for  $S_3, R_3, S_2, R_2, S_1, R_1, S_0$ , and  $R_0$  are given in Fig. 12.15, and the expressions for these inputs in minimized form are also given. The device is programmed using these equations and is shown in Fig. 12.16.

Table 12.1

UP/DOWN Control $M$	Counter state				S-R FFs inputs							
	$Q_3$	$Q_2$	$Q_1$	$Q_0$	$S_3$	$R_3$	$S_2$	$R_2$	$S_1$	$R_1$	$S_0$	$R_0$
0	0	0	0	0	0	$\times$	0	$\times$	0	$\times$	1	0
0	0	0	0	1	0	$\times$	0	$\times$	1	0	0	1
0	0	0	1	0	0	$\times$	0	$\times$	$\times$	0	1	0
0	0	0	1	1	0	$\times$	1	0	0	1	0	1
0	0	1	0	0	0	$\times$	$\times$	0	0	$\times$	1	0
0	0	1	0	1	0	$\times$	$\times$	0	1	0	0	1
0	0	1	1	0	0	$\times$	$\times$	0	$\times$	0	1	0
0	0	1	1	1	1	0	0	1	0	1	0	1
0	1	0	0	0	$\times$	0	0	$\times$	0	$\times$	1	0
0	1	0	0	1	$\times$	0	0	$\times$	1	0	0	1
0	1	0	1	0	$\times$	0	0	$\times$	$\times$	0	1	0
0	1	0	1	1	$\times$	0	1	0	0	1	0	1
0	1	1	0	0	$\times$	0	$\times$	0	0	$\times$	1	0
0	1	1	0	1	$\times$	0	$\times$	0	1	0	0	1
0	1	1	1	0	$\times$	0	$\times$	0	$\times$	0	1	0
0	1	1	1	1	0	1	0	1	0	1	0	1

(Continued)

Table 12.1 (Continued)

UP/DOWN Control $M$	Counter state				S-R FFs inputs							
	$Q_3$	$Q_2$	$Q_1$	$Q_0$	$S_3$	$R_3$	$S_2$	$R_2$	$S_1$	$R_1$	$S_0$	$R_0$
1	0	0	0	0	1	0	1	0	1	0	1	0
1	1	1	1	1	x	0	x	0	x	0	0	1
1	1	1	1	0	x	0	x	0	0	1	1	0
1	1	1	0	1	x	0	x	0	0	x	0	1
1	1	1	0	0	x	0	0	1	1	0	1	0
1	1	0	1	1	x	0	0	x	x	0	0	1
1	1	0	1	0	x	0	0	x	0	1	1	0
1	1	0	0	1	x	0	0	x	0	x	0	1
1	1	0	0	0	0	1	1	0	1	0	1	0
1	0	1	1	1	0	x	x	0	x	0	0	1
1	0	1	1	0	0	x	x	0	0	1	1	0
1	0	1	0	1	0	x	x	0	0	x	0	1
1	0	1	0	0	0	x	0	1	1	0	1	0
1	0	0	1	1	0	x	0	x	x	0	0	1
1	0	0	1	0	0	x	0	x	0	1	1	0
1	0	0	0	1	0	x	0	x	0	x	0	1
	0	0	0	0	0	0	0	0	0	0	0	0

		$Q_3 Q_2$	$M = 0$			
		00	01	11	10	
$Q_1$	$Q_0$	00	0	0	x	x
00	00	0	0	x	x	
01	01	0	0	x	x	
11	11	0	(1)	0	x	
10	10	0	0	x	x	

$$S_3 = \overline{Q}_3 \cdot Q_2 \cdot Q_1 \cdot Q_0 \cdot \overline{M} + \overline{Q}_3 \cdot \overline{Q}_2 \cdot \overline{Q}_1 \cdot \overline{Q}_0 \cdot M$$

(a) K-map for  $S_3$ 

		$Q_3 Q_2$	$M = 1$			
		00	01	11	10	
$Q_1$	$Q_0$	00	(1)	0	x	0
00	00	(1)	0	x	0	
01	01	0	0	x	x	
11	11	0	0	x	x	
10	10	0	0	x	x	

		$Q_3 Q_2$	$M = 0$			
		00	01	11	10	
$Q_1$	$Q_0$	00	x	x	0	0
00	00	x	x	0	0	
01	01	x	x	0	0	
11	11	x	0	(1)	0	
10	10	x	x	0	0	

$$R_3 = Q_3 \cdot Q_2 \cdot Q_1 \cdot Q_0 \cdot \overline{M} + \overline{Q}_3 \cdot \overline{Q}_2 \cdot \overline{Q}_1 \cdot \overline{Q}_0 \cdot M$$

(b) K-map for  $R_3$ 

		$Q_3 Q_2$	$M = 1$			
		00	01	11	10	
$Q_1$	$Q_0$	00	0	x	0	(1)
00	00	0	x	0	(1)	
01	01	x	x	0	0	
11	11	x	x	0	0	
10	10	x	x	0	0	

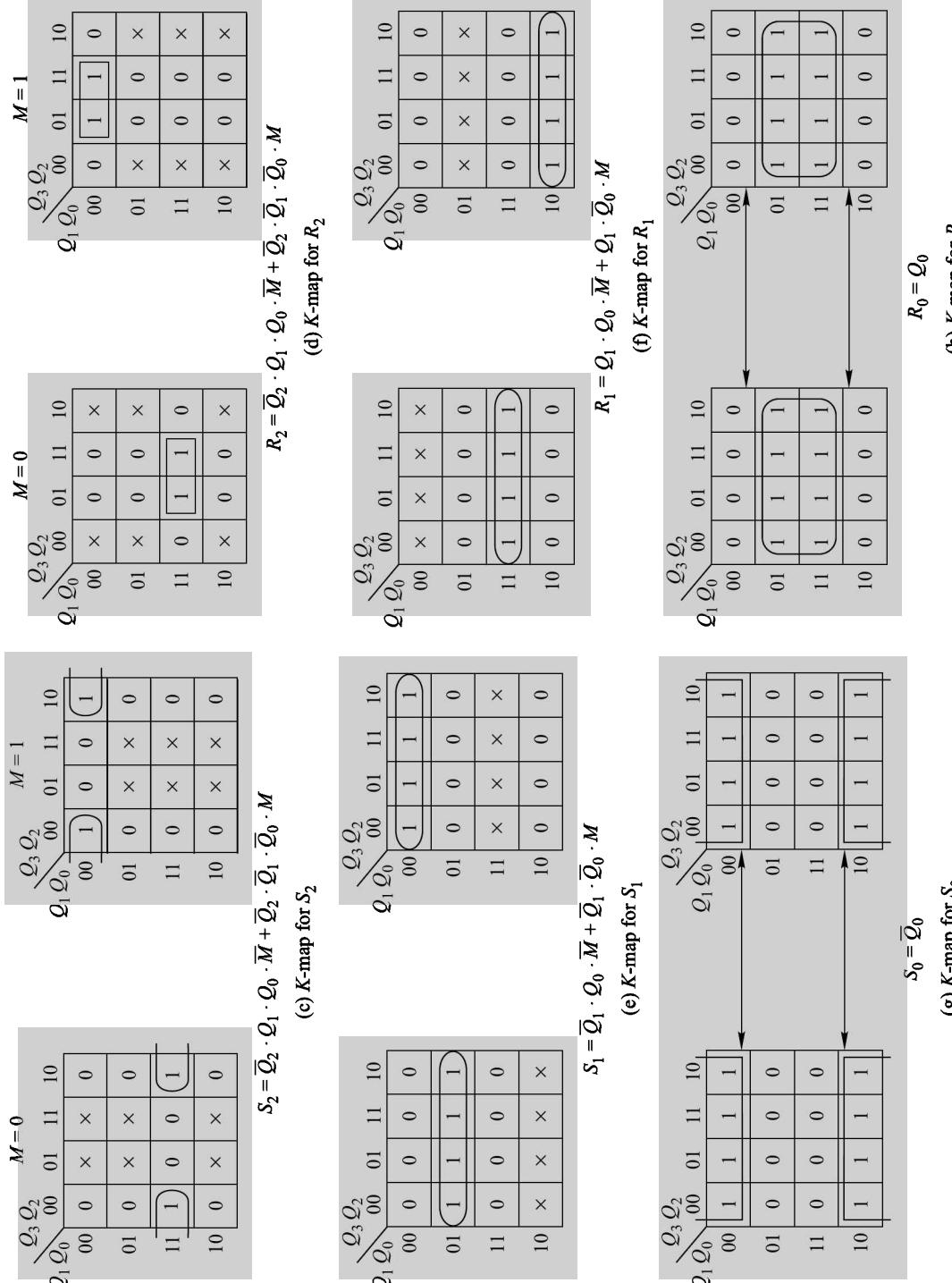
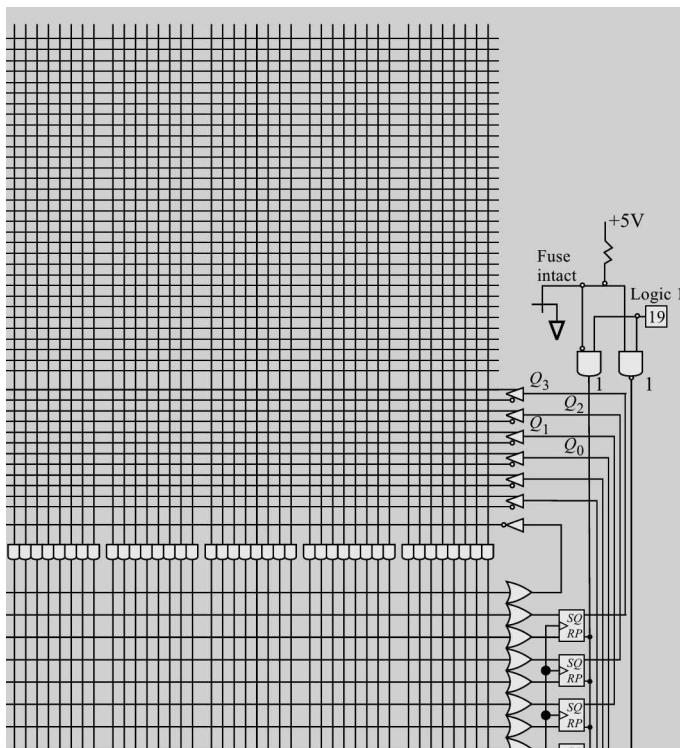


Fig. 12.15 K-Maps for Ex. 12.3

*Modern Digital Electronics*

using these equations and is shown in Fig. 12.16.



### 12.3.10 Available PLAs

Some of the commercially available PLA ICs with their features are given in Table 12.2. The 82S200 and 82S201 are pin-for-pin mask programmable replacements for the 82S100 and 82S101, respectively. The PLS 100 and PLS 105 PLAs are equivalent to 82S100 and 82S105, respectively. The DM 7575 PLA has totem-pole output, whereas DM7576 and IM5200 have passive pull-up. The devices with passive pull-up are useful for expanding functions by wire-ANDing the outputs of similar other devices.

The PLA devices are not used for new designs because of the availability of more powerful and power efficient devices, such as GALs, CPLDs and FPGAs. However, the concept of PLA is used in some CPLDs. The FPLAs are used mostly in state-machine design, where a large number of product terms are needed in each SOP expression.

## 12.4 PROGRAMMABLE ARRAY LOGIC

A most commonly used type of PLD is *programmable array logic* (PAL). It is programmable array of logic gates on a single chip in AND-OR configuration. In contrast to PLA, it has programmable AND array and a fixed OR array in which each OR gate gets inputs from some of the AND gates, i.e. all the AND gate outputs are not connected to any OR gate. Figure 12.17 illustrates the configuration of AND and OR arrays for a PAL with 5 inputs, 8 programmable AND gates and 4 fixed OR gates. Each AND gate has all the 10 inputs (in complemented and uncomplemented form) with fusible links intact which can be programmed to generate 8 product terms. Each OR gate gets inputs from the outputs of only two AND gates shown by •. The input and output circuits of PALs are similar to those of PLAs. The number of fusible links in a PAL is the product of  $2M$  and  $n$ , where  $M$  is the number of input variables and  $n$  is the number of product terms.

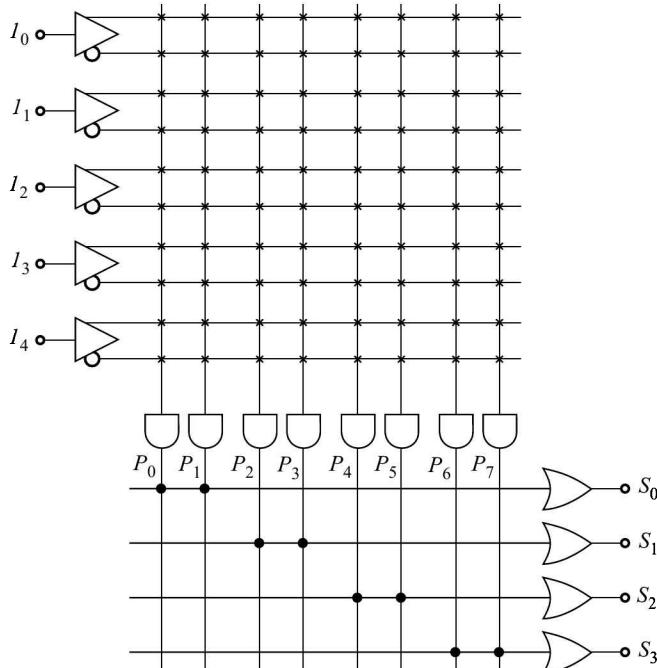


Fig. 12.17 Programmable Array Logic

Table 12.2 *Some Available PLAs with their Features*

IC No.	PLA/ FPLA	Number of Product- terms			Output	Access time ns	Supply voltage V	Power dissipation mW	Packaging	Input/ output logic levels
		Inputs	Product- terms	Output						
82S200	PLA	16	48	8	TS	80	+5	600	28-pin	TTL
82S201	PLA	16	48	8	OC	80	+5	600	DIP	compatible
82S100	FPLA	16	48	8	TS	80	+5	600	28-pin	TTL
82S101	FPLA	16	48	8	OC	80	+5	600	DIP	compatible
DM7575	PLA	14	96	8	TS	150	+5	550	24-pin	TTL
DM7576	PLA	14	96	8	OC	150	+5	550	DIP	compatible

### 12.4.1 Combinational PAL

One of the most commonly used PAL 16L8 is shown in Fig. 12.18. Its programmable AND array consists of 64 gates. It has 16 input variables and 8 outputs, therefore, each AND gate has  $2 \times 16 = 32$  inputs. Eight AND gates are associated with each output pin, seven of them provide inputs to a fixed 7-input OR gate and the eighth is connected to the enable input terminal of the output buffer. Thus, any output can perform only logic functions that can be expressed as sums of seven or fewer product terms. Each product term can be a function of up to all 16 inputs but only 7 such product terms are available. Since, in a PAL the inputs to the OR gates are fixed, therefore, no two OR gates can share a product term. Where a product term is needed by two OR gates, it must be generated twice.

There is a three-state inverter between the output of each OR gate and the output pin of the device, therefore, the output may be programmed as always enabled, always disabled, or enabled by a product term involving the inputs (Problem 12.11).

This PAL has 10 pins ( $I_1$  to  $I_{10}$ ) for 10 inputs, the other six pins are used for inputs as well as for six of the outputs. These are  $IO_2$  to  $IO_7$ ,  $O_1$  and  $O_8$  are dedicated output pins. The device can be programmed for the following options:

1. If an  $IO$  pin's output inverter is disabled, then the pin can be used only as an input.
2. If an  $IO$  pin is not required to be input, then it can be used as an output. The output inverter can be enabled either continuously or for certain input conditions by using a product term. In case it is enabled continuously, its output can also be used as an input.
3. This can be used either for the implementation of logic functions that can not be accommodated in 7 product terms by using two pass AND-OR configuration or for implementing sequential circuits by providing feedback.

Since limited number of product terms are available for each output, logic minimization techniques are required when logic circuits are implemented in PAL devices.

PAL 16H8 is same as 16L8 except that it does not have inverters in the output. Another PAL 16P8 has programmable output polarity using EX-OR gates as shown in Fig. 12.10a. The alphabets  $L$ ,  $H$ , and  $P$  used in device numbers stand for active-low, active-high, and programmable outputs, respectively, and these are all *combinational* programmable array logic devices, i.e. these do not have memory elements.

### 12.4.2 Registered PALS

Sequential digital circuits use FLIP-FLOPs in addition to combinational circuits and, therefore, for the design of sequential circuits, PALs have been developed with FLIP-FLOPs in the outputs and these devices are known as *registered* PALs. Some of the available registered PALs are 16R4, 16R6, 16R8, and their programmable output polarity versions 16RP4, 16RP6, and 16RP8.

Figure 12.19 illustrates the structure of a 16R6 registered PAL. This device has the following features:

- Eight primary (external) inputs,  $I_1$  to  $I_8$ .
- Two pins common for input/output similar to the device 16L8,  $IO_1$  and  $IO_8$ .
- Six outputs ( $O_2$  to  $O_7$ ) through positive-edge-triggered D-FFs and three state inverters.  $\bar{Q}$  outputs of these FFs are routed back to the AND array, making total number of inputs to the AND array as 16 and 8 outputs.
- The FFs are all controlled by a common clock through a pin and another dedicated input pin is available for output enable control of inverters.

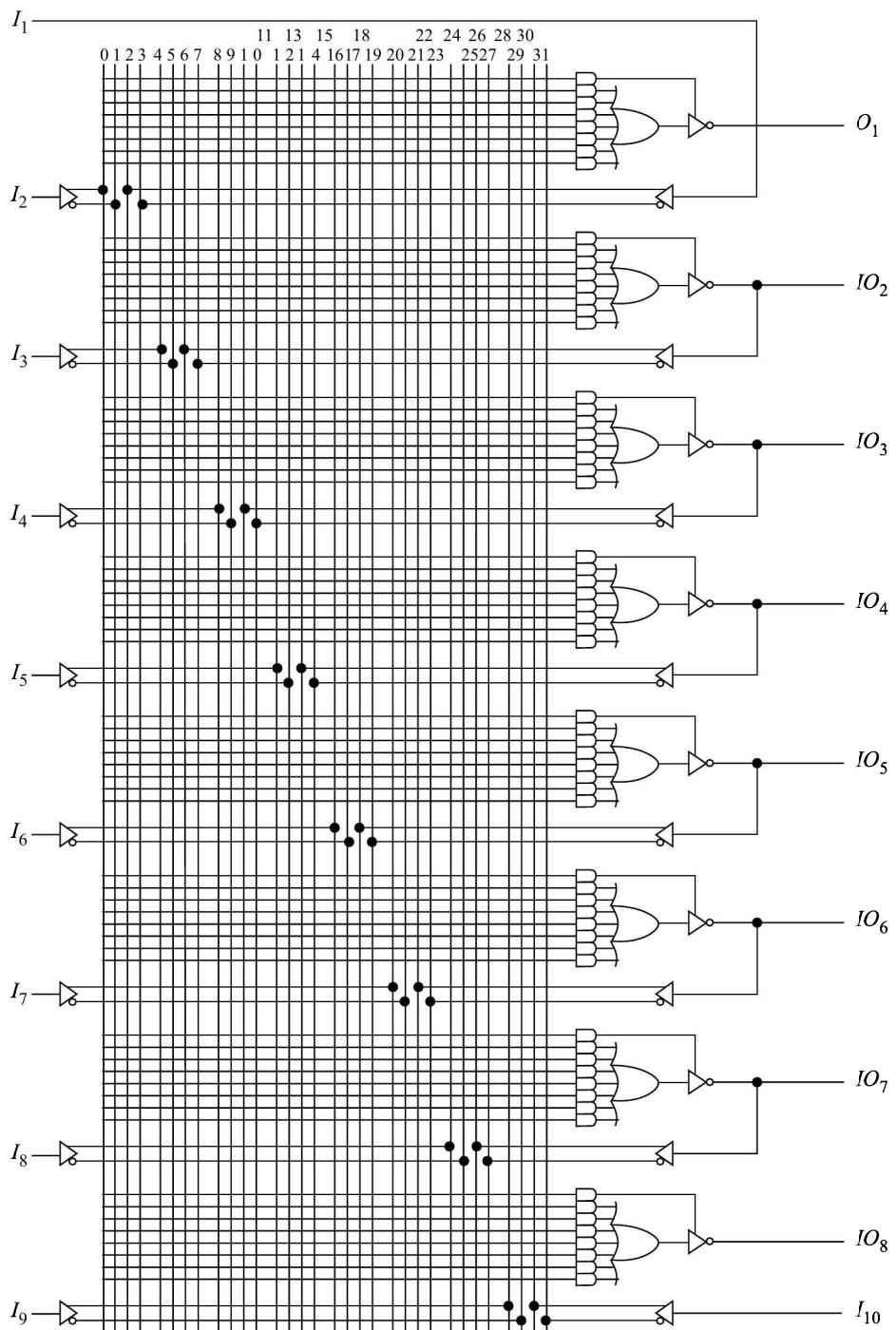


Fig. 12.18 Logic Diagram of the PAL 16L8

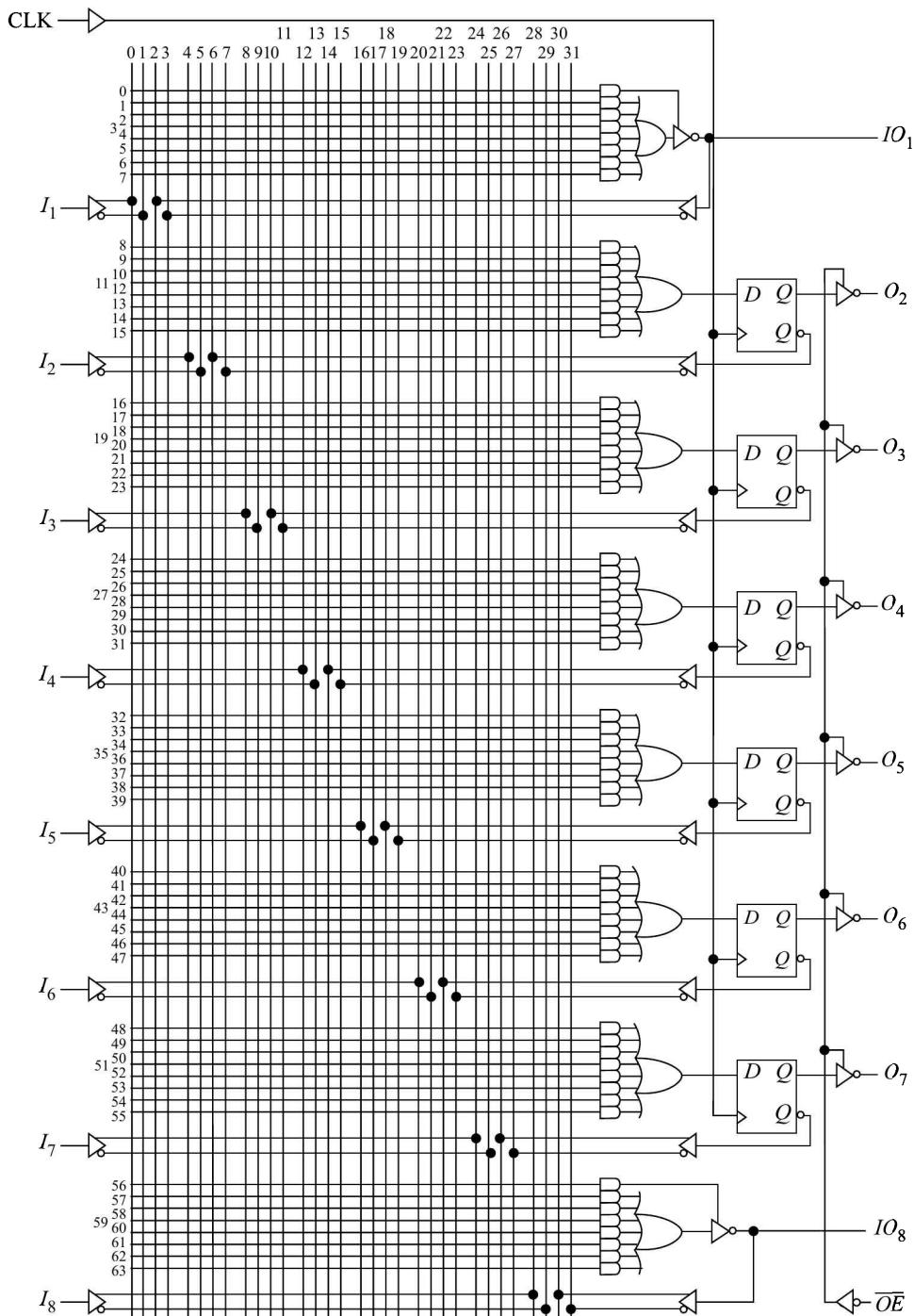


Fig. 12.19 Logic Diagram of the Registered PAL 16R6

### 12.4.3 Configurable PALs

Developments in the design of programmable array logic devices have led to the introduction of configurable outputs enhancing the output capabilities of such devices. The *configurable* (also known as *generic*) device architecture is provided by equipping the device with enhanced special circuitry, known as *output macrocells*. A macrocell has circuitry with fuses which can be configured for a variety of output options, giving flexibility to the device. A configurable PAL can replace a large number of simpler PAL type devices with a ‘one size fits all’ device and allows designs to be implemented that are challenging or simply impossible for the simpler PAL devices to handle.

The most popular industry standard PAL devices are 16V8, 20V8, and 22V10.

Figure 12.20 gives logic diagram of 22V10 configurable PAL. It has 22 inputs, 10 outputs, and 120 product terms. Out of 22 inputs, 12 are dedicated inputs ( $I_1$  to  $I_{12}$ ) and ten may be used for inputs as well as outputs ( $IO_1$ – $IO_{10}$ ). One of the dedicated inputs  $I_1$  also functions as the common clock input to the positive-edge-triggered D-type FF of each of the ten output macrocells. All the ten outputs are through configurable macrocells and three-state inverters. The output pins can also be used as inputs.

There are two extra product terms also available as seen in Fig. 12.20 which can be used for synchronously presetting (SP) and asynchronously resetting (AR) the D-type FFs in the output macrocells.

Figure 12.21 gives the logic diagram of an output macrocell. All the output macrocells are identical. Each macrocell contains a positive edge-triggered D-type FF and fuse-configurable multiplexers. The two fuses,  $S_1$  and  $S_0$  that control the multiplexers can be configured in four different ways, as shown in Fig. 12.22. From this, it is observed that this configurable PAL can function as registered PAL or combinational PAL, and in each case the output may be in inverted or non-inverted form. It may also be noted from Fig. 12.20 that all the number of product terms available to various OR gates in the device are not same. The number of product terms associated with each OR gate is indicated near the OR gate. Because of this, the logic functions of significantly more complexity can be implemented using 22V10.

### 12.4.4 Electrically Erasable Programmable Logic Devices (EEPLDs)

The industry standard programmable array logic devices are now available using CMOS electrically erasable flash technology. These EEPLDs are reprogrammable, and are available in a variety of voltage and power-saving options to meet various different requirements. Some of the features of ATF16V8B, ATF20V8B, and ATF22V10B are:

- Programming and erasing are performed using standard PLD programmers.
- A single *security fuse* is provided to prevent unauthorised fuse patterns. It should be programmed last, since once it is programmed, no further data can be programmed.
- There is a provision of *electronic signature*, for which a 64-bit of programmable memory is always available to the user for user-specific data. This feature is available even if the device is secured.
- They can retain data for 20 years and 100 erase/write cycles are possible.
- Their inputs and outputs are CMOS and TTL compatible.

These devices are discussed below.

There is another family of electrically erasable devices that is known as programmable electrically erasable logic (PEEL) device. It uses CMOS electrically erasable technology and it is reprogrammable. The 18CV8 PEEL device will also be discussed below.

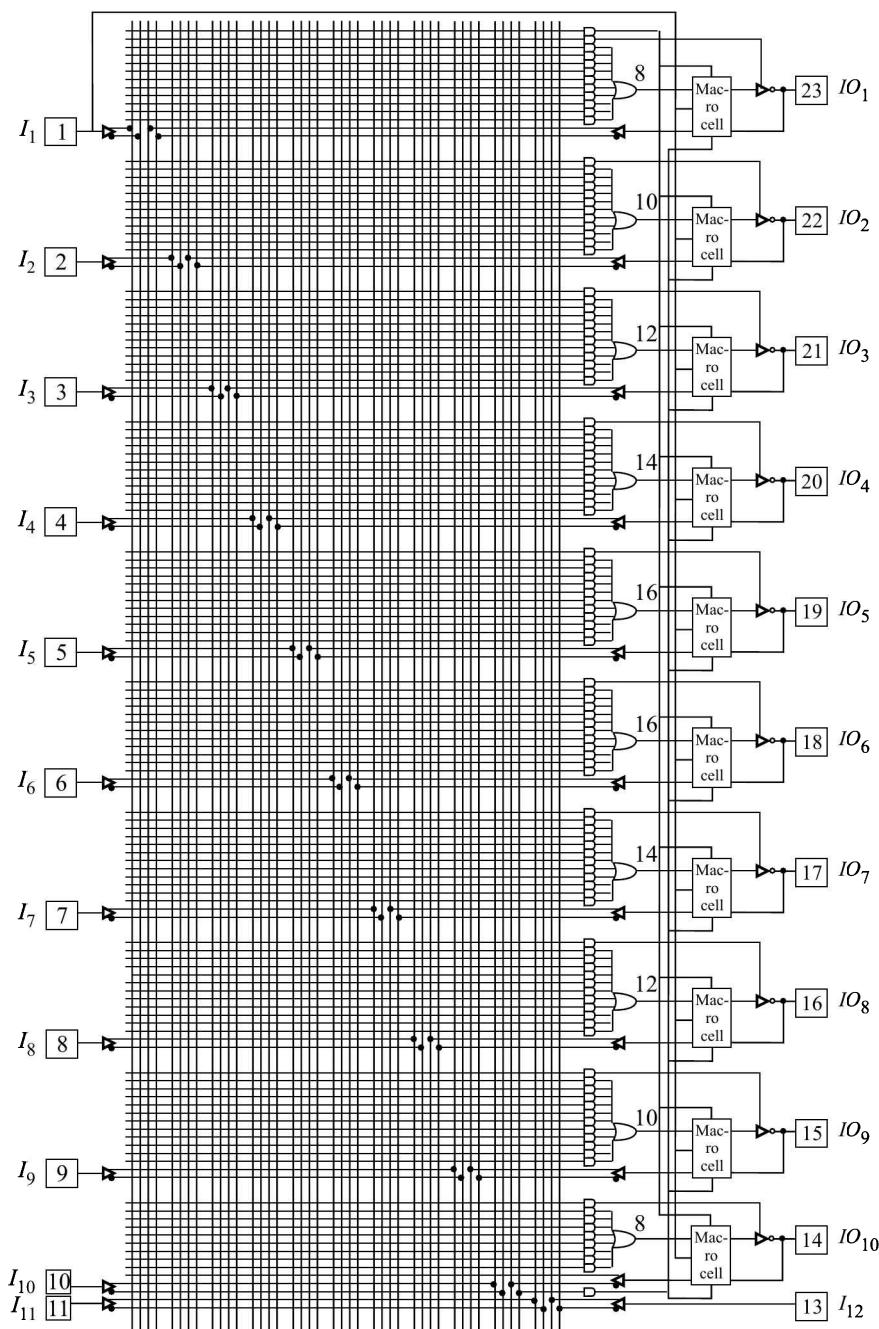


Fig. 12.20

22V10 Configurable PAL

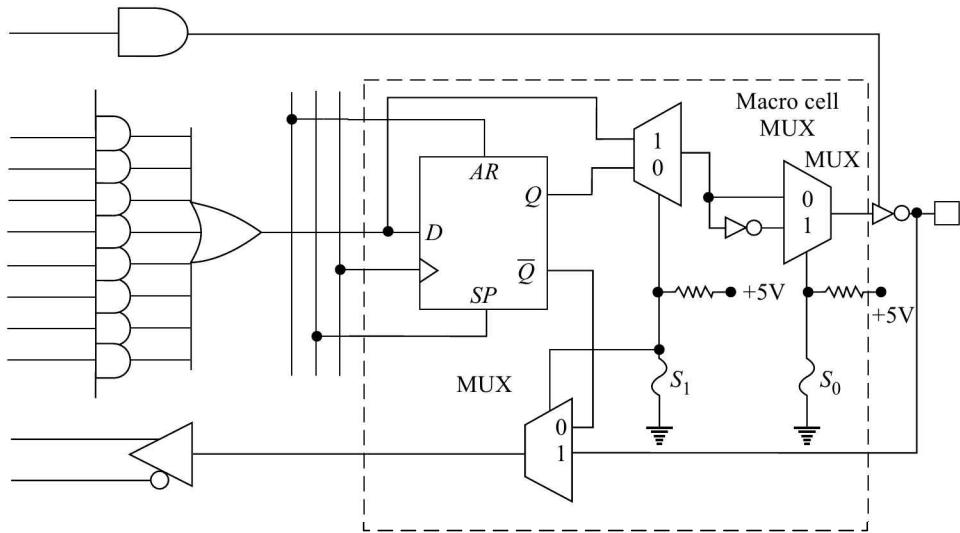
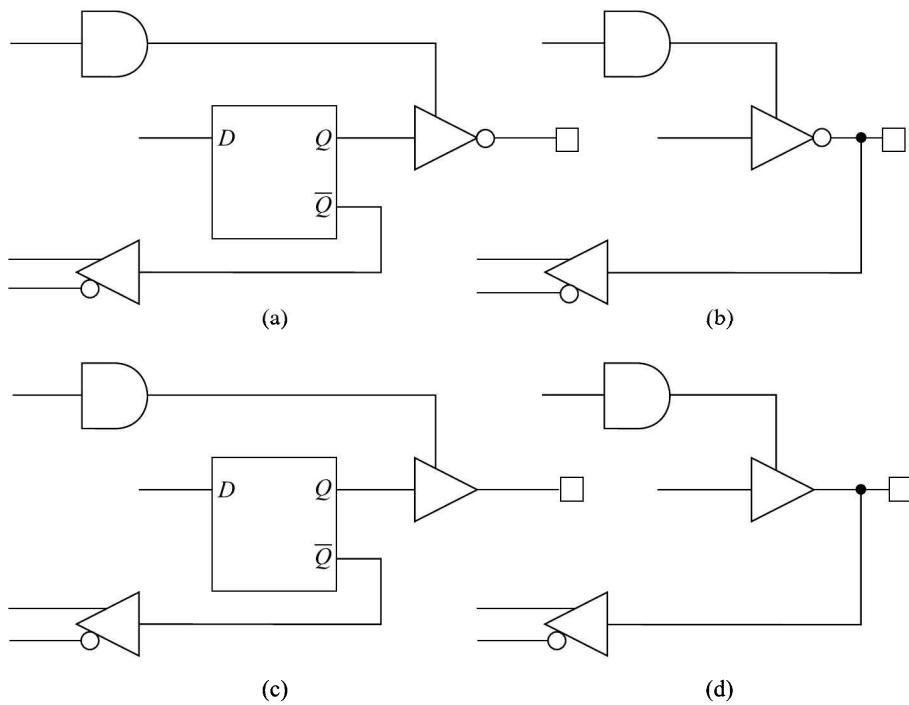


Fig. 12.21 Logic Diagram of a 22V10's Output Macrocell

Fig. 12.22 Output Configurations for Fuses (a) S<sub>1</sub> and S<sub>0</sub> Intact (b) S<sub>0</sub> Intact and S<sub>1</sub> Open (c) S<sub>0</sub> Open and S<sub>1</sub> Intact (d) S<sub>1</sub> and S<sub>0</sub> Open

## **ATF16V8B EEPROM**

This device has 16 inputs (8 dedicated inputs and 8 I/O) and is available in 20-pin package. There are eight configurable macrocells which can be configured as

- registered output
- combinatorial I/O
- combinatorial output, or
- dedicated input.

It can be configured in three different modes:

- Registered mode
- Complex mode
- Simple mode

### *Registered Mode*

The registered mode is used if one or more registers are required. In this mode, each macrocell can be configured as registered or combinatorial output, or I/O, or as an input. Figure 12.23 shows the logic diagram and Fig. 12.24 shows its configuration in the registered mode. Here, pin 1 is used for common clock for the registered outputs and pin 11 is used for common output enable ( $\overline{OE}$ ). Figure 12.25 shows the macrocell configuration for combinatorial output, or I/O or as an input.

The following registered PAL devices can be emulated using the registered mode: 16R8, 16R6, 16R4, 16RP8, 16RP6, and 16RP4.

### *Complex Mode*

In the complex mode, combinatorial output and I/O functions are possible. Here, in addition to eight dedicated inputs, the pins 1 and 11 also are used as regular inputs. Pin 13 through 18 have feedback paths back to the AND-array, which makes full I/O capability possible. Pins 12 and 19 are outputs only.

Figure 12.26 shows the logic diagram and Fig. 12.27 shows the macrocell configuration in the complex mode. The PAL 16L8, 16H8, and 16P8 devices can be emulated using complex mode.

### *Simple Mode*

In the simple mode, 8 product terms are allocated to the sum term. Pins 15 and 16 are permanently configured as combinatorial outputs. The remaining six macrocells can be configured as either inputs or combinational outputs with pin feedback to the AND-array. Pins 1 and 11 are regular inputs. Figure 12.28 shows the logic diagram and Fig. 12.29 shows the macrocell configuration in the simple mode.

The following PALs can be emulated using the simple mode:

10L8	10H8	10P8
12L6	12H6	12P6
14L4	14H4	14P4
16L2	16H2	16P2

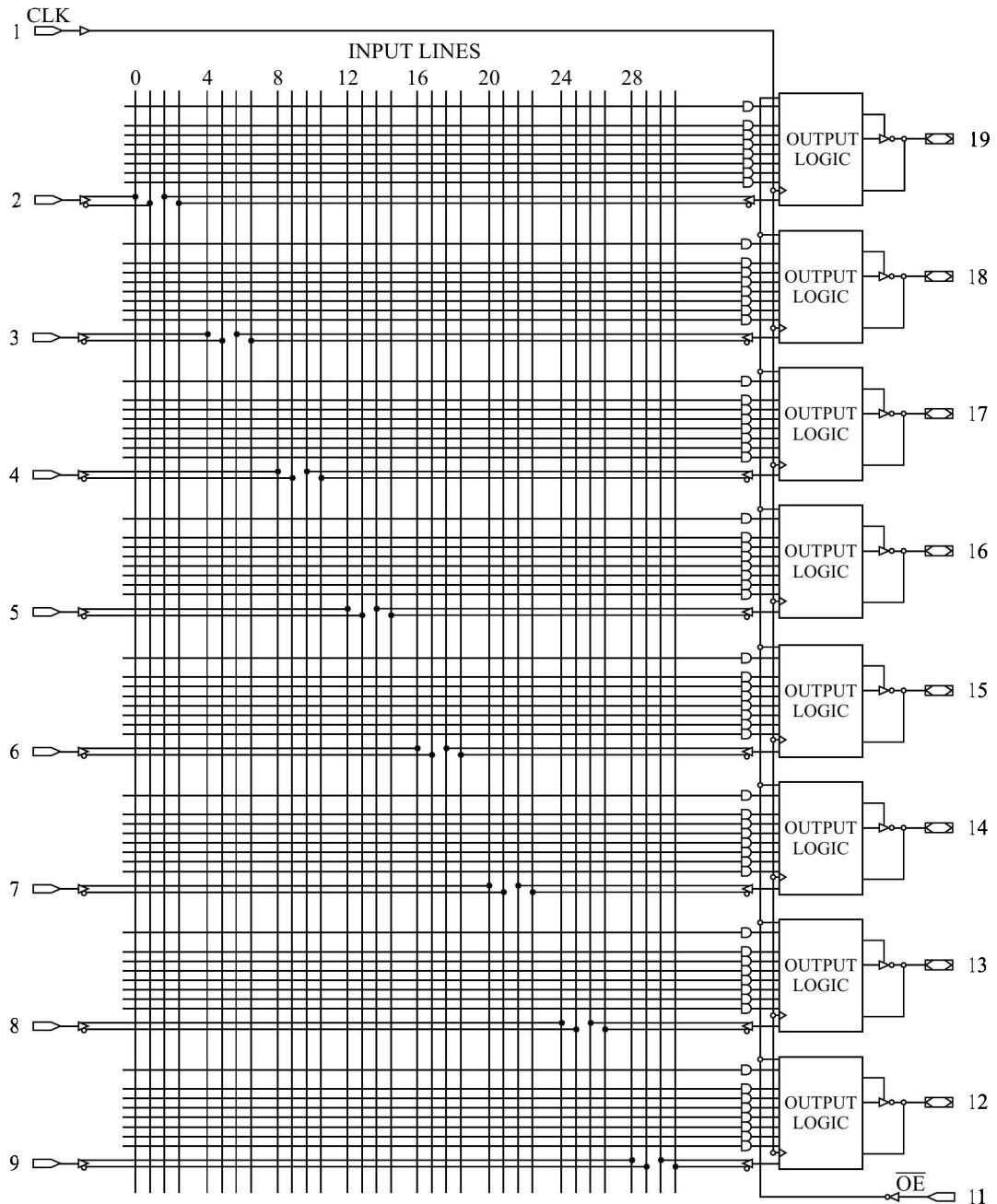


Fig. 12.23 Logic Diagram of 16V8B in Registered Mode

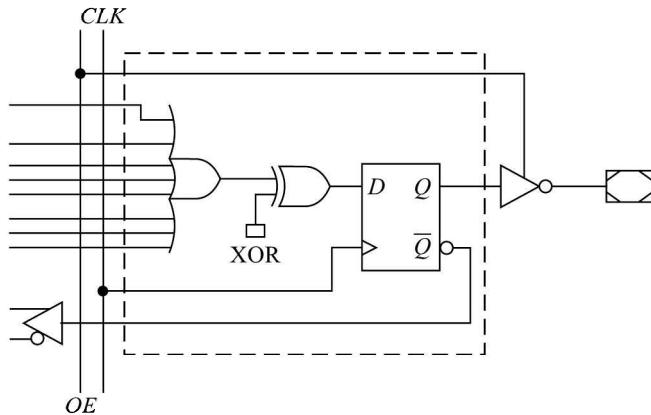


Fig. 12.24 *Registered Mode Output Cell Configuration of 16V8B*

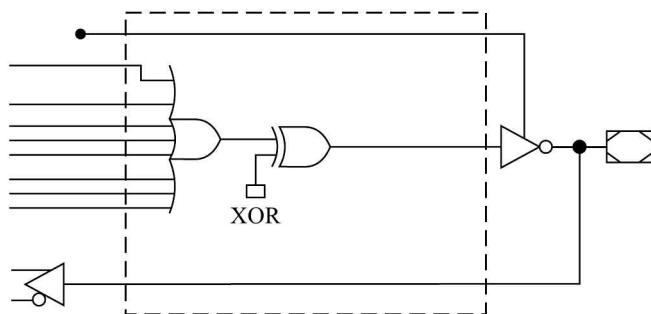


Fig. 12.25 *Combinatorial Configuration in Registered Mode Configuration of 16V8B*

## ATF20V8B EEPLD

The 20V8 device has 20 inputs (8 I/O and 12 dedicated inputs) and is available in 24 and 28 pin packages. There are eight configurable macrocells which can be configured in registered, complex, and simple modes in a way similar to the 16V8 device. It incorporates a superset of the generic architectures, which allows direct replacement of the 20R8 family and most of the 24-pin combinatorial PLDs.

## 18CV8 PEEL

The 18CV8 PEEL device (Fig. 12.30) has ten dedicated inputs and 8 I/O providing up to 18 inputs and 8 outputs. It can implement up to eight SOP logic expressions. Its output macrocell is shown in Fig. 12.31 and can be configured in twelve different configurations (Problem 12.10).

### 12.4.5 Generic Array Logic (GAL) Devices

The electrically erasable PLDs of Lattice Semiconductor Corporation have GAL as their registered trademark. The SPLD devices are GAL 16V8, GAL 20V8, and isp GAL 22V10 corresponding to 16V8, 20V8, and

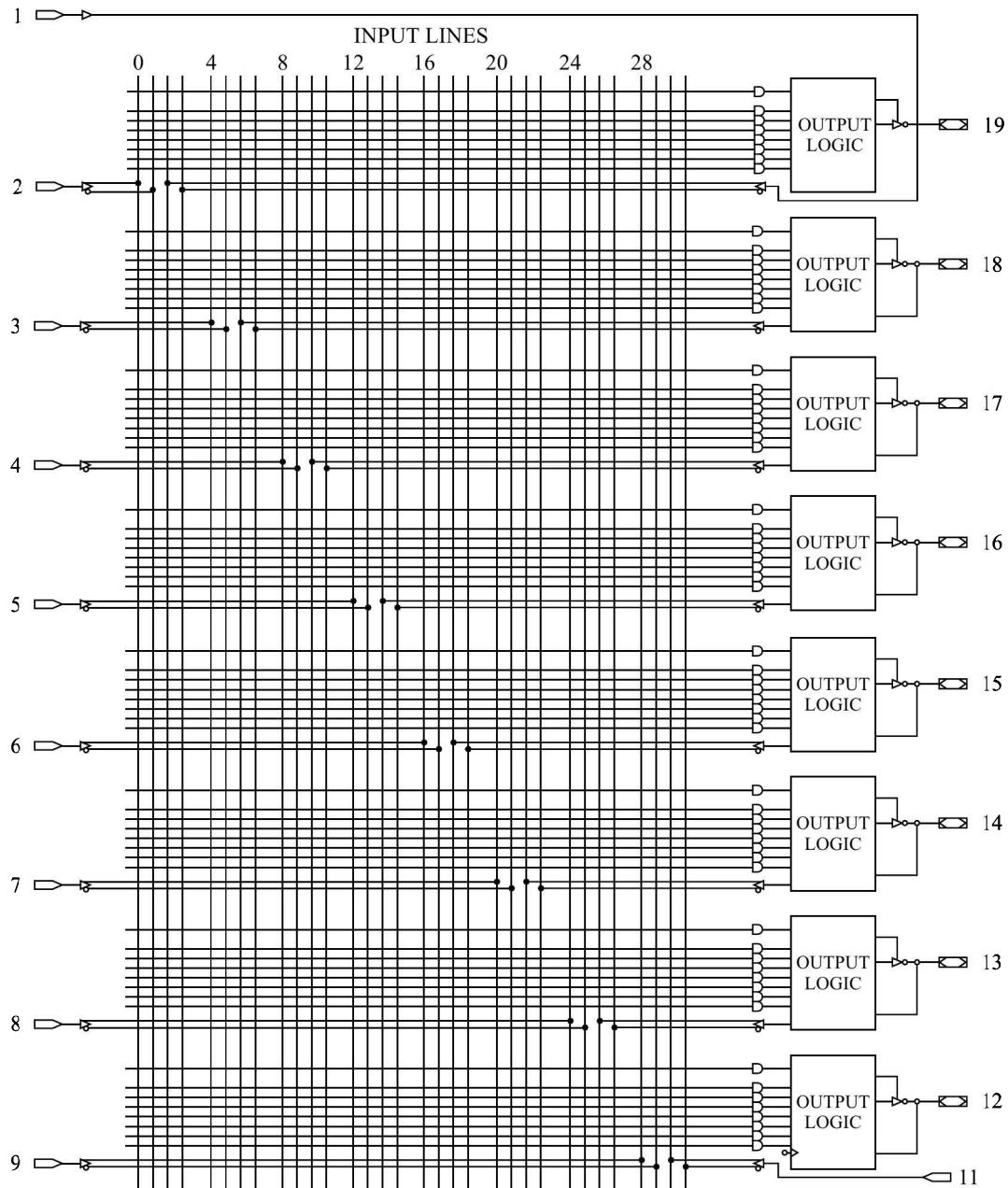


Fig. 12.26 Complex Mode Logic Diagram of 16V8B

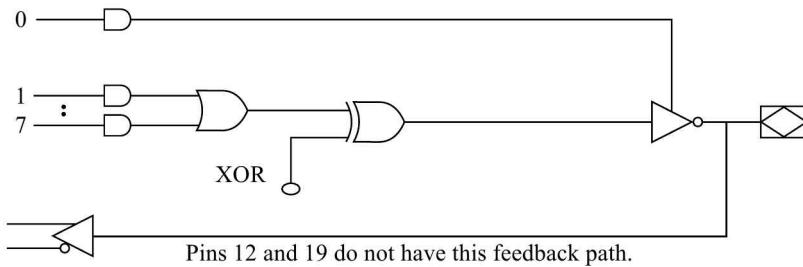


Fig. 12.27 Macrocell Configuration for Complex Mode

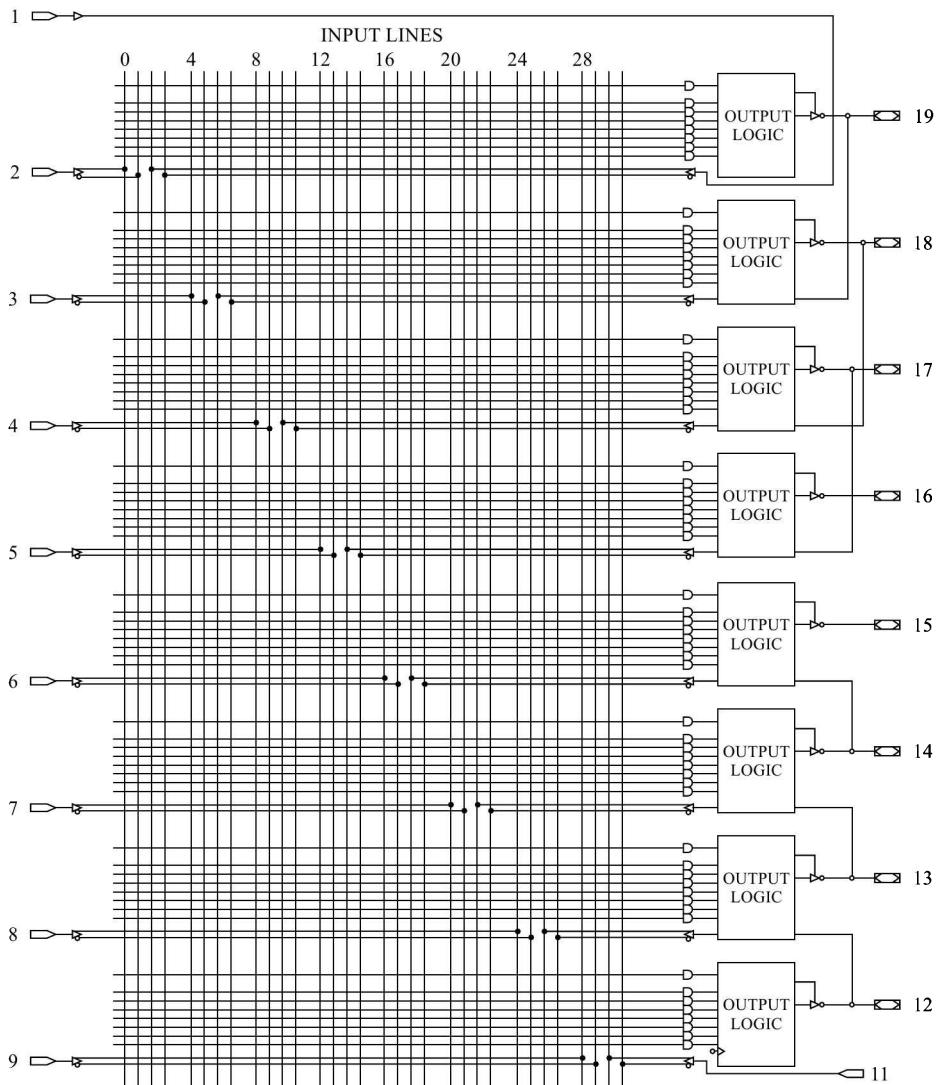
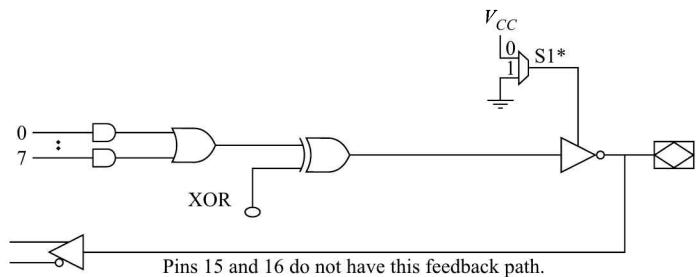
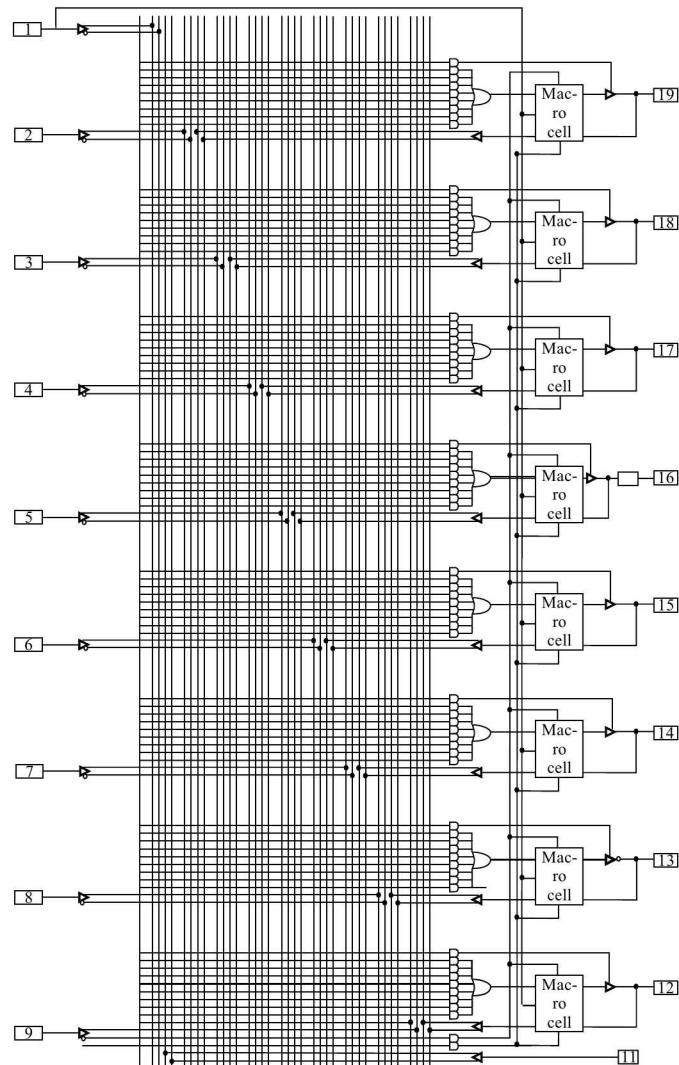
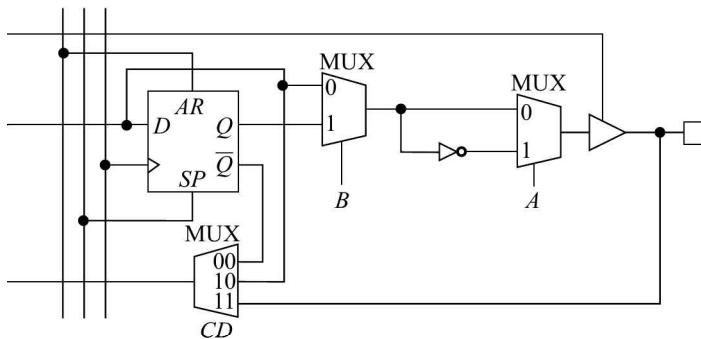


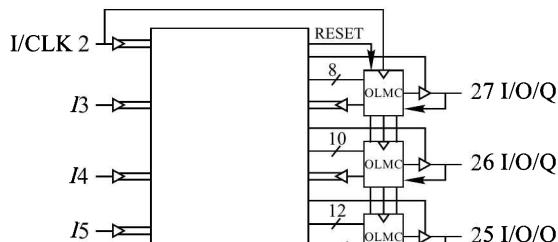
Fig. 12.28 Simple Mode Logic Diagram of 16V8B

Fig. 12.29 **Macrocell Configuration for Simple Mode**Fig. 12.30 **18CV8 PEEL Device**

Fig. 12.31    **Macrocell Details of 18CV8**

eatures and modes of GAL 16V8 and GAL 20V8 are similar to the 16V8 and discussed earlier. The isp GAL 22V10 has some additional features in comparison manufacturers. These are discussed below.

ically erasable  
fully function,  
mpatible with  
2V10 devices.  
100 ms) which  
figure quickly



Once the function is programmed, the non-volatile E<sup>2</sup>CMOS cells will not lose the pattern even when the power is turned off.

The GAL devices and other EEPLDs can be used for DMA control, state machine control, and high-speed graphics processing etc.

#### 12.4.6 EX-OR PALs

The use of EX-OR gates in PALs to implement fuse configurable output polarity has already been discussed. EX-OR gates are also used in AND-OR-EX-OR configuration as shown in Fig. 12.33 in the EX-OR PAL device. Here, the EX-OR outputs are fed to the inputs of the FLIP-FLOPs. Each of these EX-OR gates is fed in turn by two sum-of-products arrays of two product terms each. This configuration is used to reduce the amount of logic required for many applications, particularly counters which allow complex designs to be implemented using very few product terms.

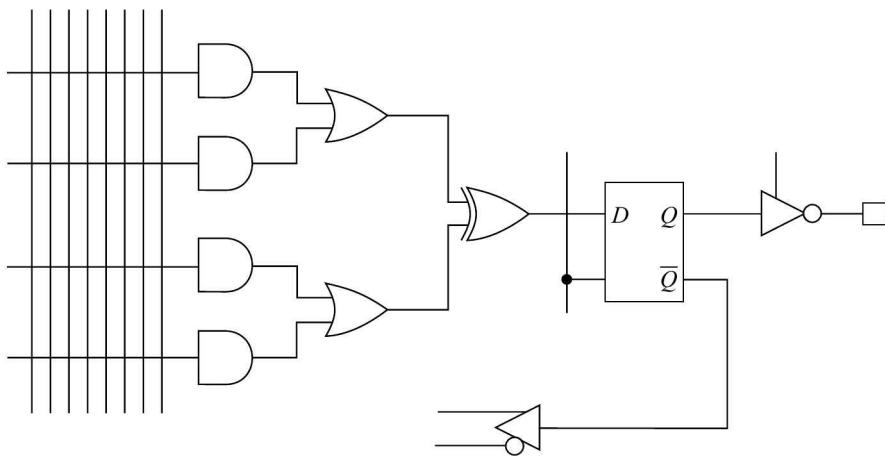


Fig. 12.33 EX-OR PAL

There are a number of EX-OR PAL devices available. Figure 12.34 shows the logic diagram of EX-OR PAL 20 × 10A. It is an EX-OR registered 24-pin programmable logic device. It has EX-OR gates preceding each FLIP-FLOP. The EX-OR gate combines two sum terms, each composed of two product terms as shown in Fig. 12.33. Similar to EEPLDs and GALS, this EX-OR PAL also has security fuse. After programming and verification, the design can be secured by programming the security fuse. When the security fuse is programmed, the array will be read as if every fuse is intact. Therefore, once programmed, the internal programmed pattern can not be read by a device programmer securing proprietary designs from competitors.

The 20 × 8A and 20 × 4A EX-OR PALs have 2 and 6 combinational outputs respectively in addition to their EX-OR registered outputs.

#### 12.4.7 Available SPLDs

Some of the commercially available SPLDs with their features are given in Table 12.3.

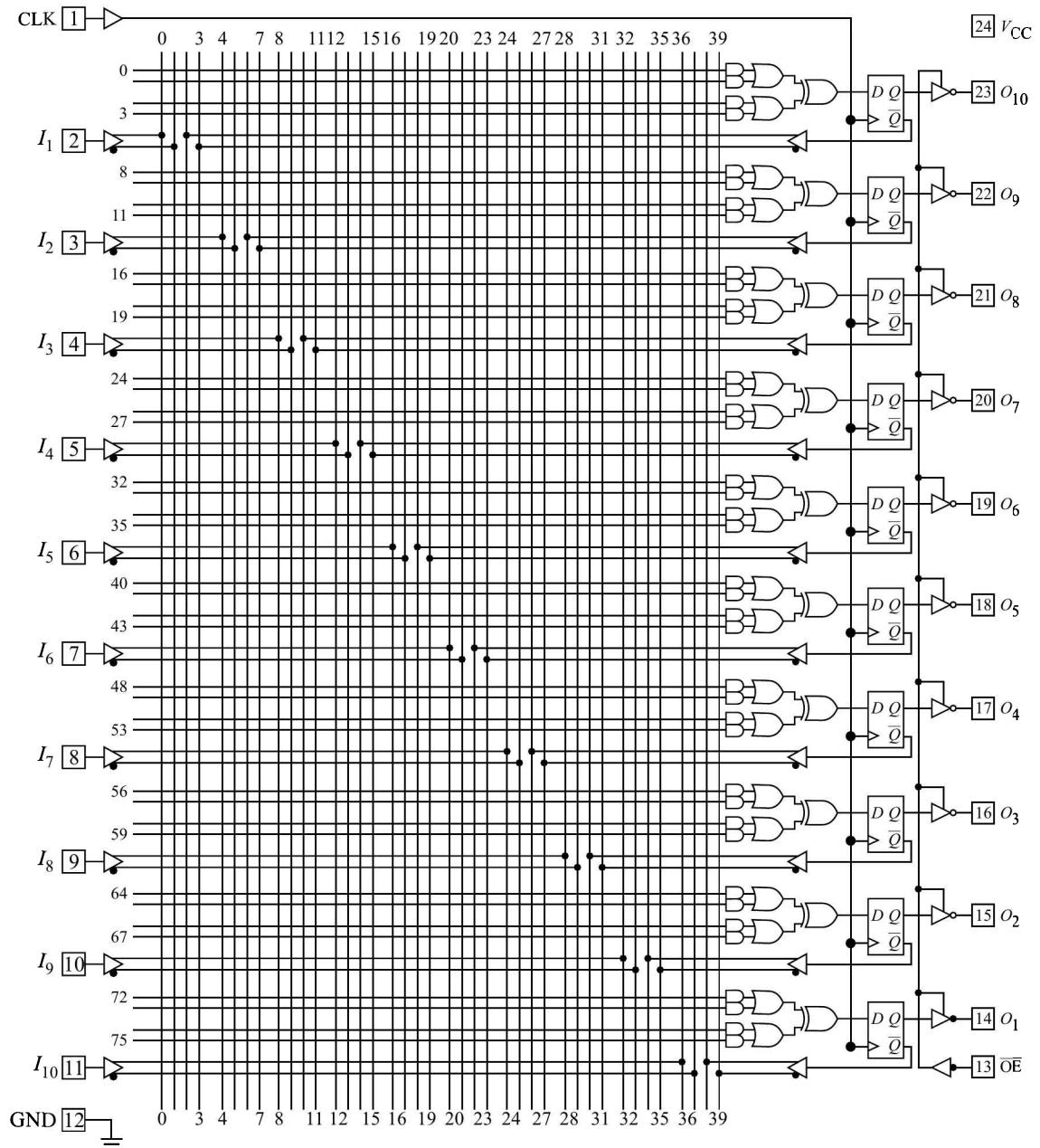


Fig. 12.34 Logic Diagram of EX-OR PAL 20x10A

Table 12.3 Some available SPLDs

Device	Inputs			Outputs				Package pins
	External	Feedback	Total	Bidirectional combinational	Regi-stered	Combi-national	Total	
PAL	10L8	10	0	10	0	0	8	8
PAL	16L8	10	6	16	6	0	2	8
PAL	20L8	14	6	20	6	0	2	8
PAL	20L10	12	8	20	8	0	2	10
PAL	16R4	8	8	16	4	4	0	8
PAL	16R6	8	8	16	2	6	0	8
PAL	16R8	8	8	16	0	8	0	8
PAL	20R4	12	8	20	4	4	0	8
PAL	20R6	12	8	20	2	6	0	8
PAL	20R8	12	8	20	0	8	0	8
PAL	20X4	10	10	20	6	4	0	10
PAL	20X8	10	10	20	2	8	0	10
PAL	20X10	10	10	20	0	10	0	10
PAL/	16V8	8	8	16	8	8	8	8
GAL/	20V8	12	8	20	8	8	8	24
EEPLD	22V10	12	10	22	10	10	0	10
PEEL	18CV8	10	8	18	8	8	8	24

### 12.4.8 Manufacturers of SPLDs

Some of the major manufacturers of SPLDs, alongwith their some of the SPLD products and their WWW locators are given in Table 12.4

Table 12.4 SPLD Manufacturers

Manufacturer	SPLD Products	WWW Locator
Altera	Classic	<a href="http://www.altera.com">http://www.altera.com</a>
Atmel	PAL	<a href="http://www.atmel.com">http://www.atmel.com</a>
Cypress	PAL	<a href="http://www.cypress.com">http://www.cypress.com</a>
Lattice	GAL	<a href="http://www.latticesemi.com">http://www.latticesemi.com</a>
Philips	PLA, PAL	<a href="http://www.philips.com">http://www.philips.com</a>
Vantis	PAL	<a href="http://www.vantis.com">http://www.vantis.com</a>

### 12.5 COMPLEX PROGRAMMABLE LOGIC DEVICES (CPLDS)

The simple programmable logic devices (SPLDs), such as PALs, EEPLDs, and GALs etc. have limited number of inputs, product terms, and outputs. These devices, therefore, can support up to about 32 total number of inputs and outputs only (see Table 12.3).

For implementation of circuits that require more inputs and outputs than that are available in a single SPLD chip, either multiple SPLD chips can be employed as discussed in Sec. 12.3.8 or more sophisticated type of chip, referred to as *complex programmable logic device* (CPLD) can be used.

The expansion of PLD using multiple SPLD chips have the following disadvantages:

- PC board area requirement increases with the number of chips.
- Connecting wires will result in adverse capacitive effects.
- Power requirement increases with the number of chips.
- The system cost increases.

Another method of increasing the I/O and product terms can be designing of PLDs using the architecture of SPLDs. This approach was discarded by the designers because of the following problems associated with this approach

- increase in capacitive effects
- increase in leakage currents
- decrease in speed
- cost effectiveness

In view of the above difficulties, complex programmable logic devices (CPLDs) were evolved. A CPLD is just a collection of individual PLDs on a single chip and programmable interconnection structure. By using programming methods, the resources available in various PLDs can be shared in different ways to design complex logic functions.

The complexity of any digital IC chip can be specified in terms of number of equivalent 2-input NAND gates. A typical PAL has 8 macrocells, if each macrocell represents about 20 equivalent gates, then the PAL can accommodate a circuit that needs up to about 160 gates. For circuits requiring a very large number of gates, CPLDs having large number of macrocells (say 512 macrocells) can implement circuits of up to about 10,000 equivalent gates. There are a number of manufacturers of CPLDs manufacturing a wide range of products with different features. Some of the major manufacturers of CPLDs are given in Table 12.5.

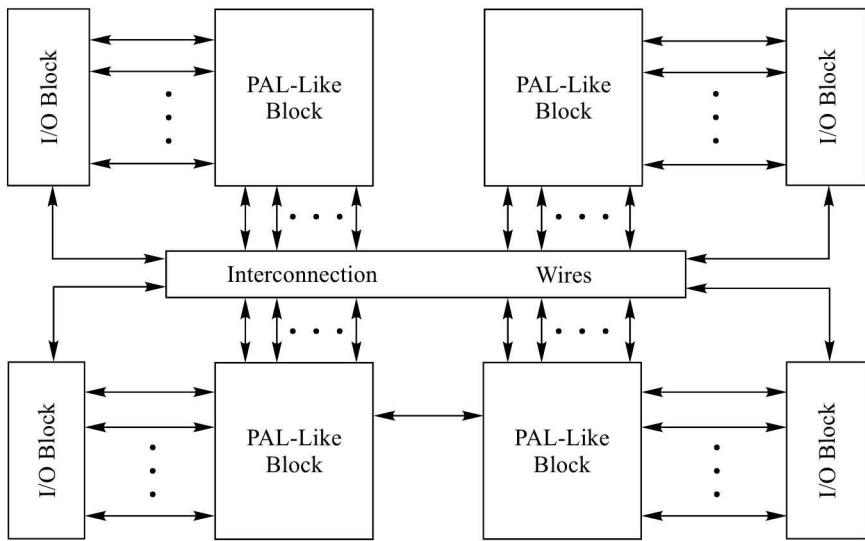
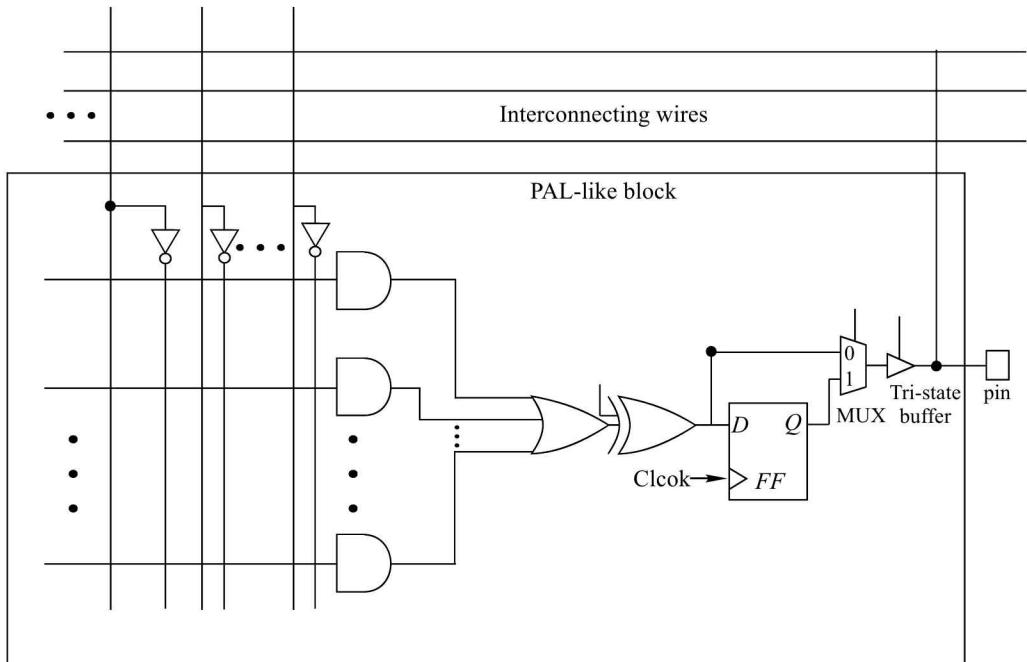
Table 12.5 **CPLD Manufacturers**

Manufacturer	CPLD Products	WWW Locator
Altera	MAX 3000, 7000 and 9000	<a href="http://www.altera.com">http://www.altera.com</a>
Atmel	ATF, ATV	<a href="http://www.atmel.com">http://www.atmel.com</a>
Cypress	FLASH 370, Ultra 37000, Delta 39 K, Quantum 38 K	<a href="http://www.cypress.com">http://www.cypress.com</a>
Lattice	isp LSI 1000 to 8000	<a href="http://www.lattice.com">http://www.lattice.com</a>
Philips	XPLA	<a href="http://www.philips.com">http://www.philips.com</a>
Vantis	MACH 1 to 5	<a href="http://www.vantis.com">http://www.vantis.com</a>
Xilinx	XC9500, CoolRunner-II	<a href="http://www.xilinx.com">http://www.xilinx.com</a>

### 12.5.1 Block Diagram

Figure 12.35 gives block diagram of a complex programmable logic device (CPLD). It consists of a number of *PAL-like blocks*, *I/O blocks*, and a set of *interconnection* wires. The PAL-like blocks are connected to a set of interconnection wires and each block is also connected to an I/O block to which a number of chip's input and output pins are attached.

A PAL-like block usually consists of about 16 macrocells. Each macrocell consists of an AND-OR configuration, an EX-OR gate, a FLIP-FLOP, a multiplexer, and a tri-state buffer. A typical macrocell is shown in Fig. 12.36. Each AND-OR configuration usually consists of 5-20 AND gates and an OR gate

Fig. 12.35 **Block Diagram of a CPLD**Fig. 12.36 **A Typical Macrocell of a CPLD**

with 5-20 inputs. An EX-OR gate is used to obtain the output of OR gate in inverted or non-inverted form depending upon its other input being 1 or 0 respectively. A D-FF stores the output of the EX-OR gate, a multiplexer selects either the output of the D-FF or the output of the EX-OR gate depending upon its select

input (1 or 0). The tri-state buffer acts as a switch which enables the chip's pin to be used either as an output (tri-state enabled) or as an input (tri-state disabled). In case the chip's pin is used as an input pin, an external source can drive a signal on to the pin which can be connected to other macrocells using the interconnection wiring. When used as an input pin, the macrocell becomes redundant and it is wasted.

### 12.5.2 Programming

Programmable logic devices, SPLDs and CPLDs, are implemented using *electrically erasable programmable read-only memory (EEPROM)* technology. These are programmed in the same way as EEPROMs. The SPLD chips have a small number of pins and can therefore be taken out of the circuit board, without much of inconvenience, and put in a programming unit. In the case of CPLDs, instead of relying on a programming unit to configure a chip, it would be very convenient and advantageous if it is possible to perform the programming with the chip remaining attached to the circuit board itself. This method of programming is known as *in-system programming (ISP)*. There are two main reasons for employing ISP.

- CPLDs have large number of pins (may even exceed 200) on the chip package, and these pins are fragile and easily bent.
- A socket is required to hold the chip in a programming unit. For large CPLDs the packages used are very expensive, sometimes more expensive than the CPLD device itself.

For the reasons mentioned above, CPLD devices usually support the ISP technique. For programming SPLDs and CPLDs a large number of programmable switches are required to be configured, hence it is not practically feasible for a user of these chips to specify manually the desired state of each switch. For this purpose computer-aided design (CAD) systems are employed. Once the user has completed the design of a circuit using CAD tools, a *programming file* or *fuse map* is generated, that specifies the state of each switch in the target PLD required to realize the designed circuit. A computer system that runs the CAD tools is connected by a cable to the programming unit. In the case of the ISP technique a small connector is included on the printed circuit board (PCB) that houses the CPLD and the computer system is connected by a cable to this connector. The programming involves transferring the programming file generated by the CAD system from the computer into the CPLD through this cable. The circuitry on the CPLD that allows in-system programming has been standardized by the IEEE and is usually called a *JTAG port*. The abbreviation JTAG stands for Joint Test Action Group. It uses four wires to transfer information between the computer and the device being programmed. Figure 12.37 illustrates the use of a JTAG port for programming two CPLDs on a circuit board.

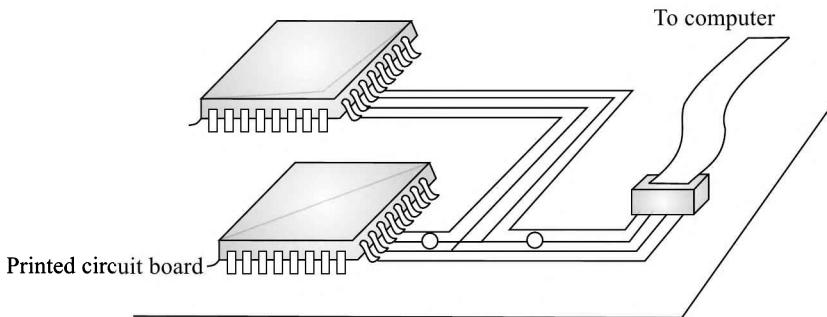


Fig. 12.37 **JTAG ISP Programming**

### 12.5.3 Packaging

CPLDs have a large number of pins, making it impractical to use dual-in-line packaging (DIP). Some of the commonly used packages for CPLDs are:

**Plastic-Leaded Chip Carrier (PLCC)** A PLCC package has pins on all the four sides that ‘wrap around’ the edges of the chip, rather than extending straight down as in the case of a DIP. The IC socket of PLCC is soldered to the PCB, and the chip is held in the socket by friction. Figure 12.38 illustrates a PLCC package with socket.

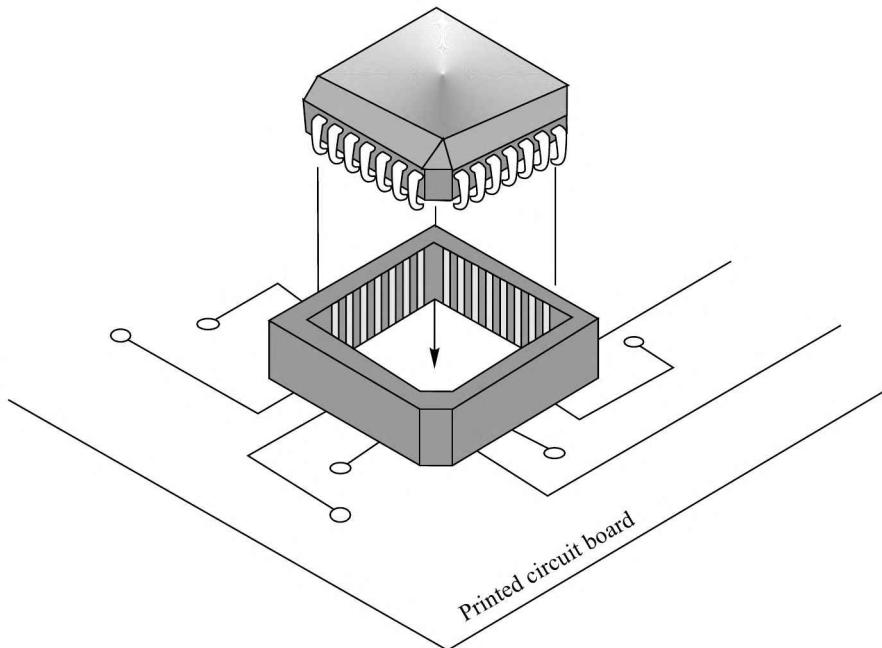


Fig. 12.38 *A PLCC Package with Socket*

**Quad flat pack (QFP)** A QFP package also has pins on all four sides like a PLCC package, but with pins extending outward from the package with a downward-curving shape as shown in Fig. 12.39. The QFP’s pins are much thinner than those on a PLCC, making it suitable for supporting a larger number of pins. QFPs are available with more than 200 pins, whereas PLCCs are limited to fewer than 100 pins. Some of the varieties of QFPs available are: Plastic quad flat pack (PQFP), power quad flat pack (RQFP), and 1.0 mm thin quad flat pack (TQFP).

**Ceramic pin grid array (PGA)** It has pins extending straight outwards from the bottom of the package in a grid pattern. It can accommodate a few hundred pins in total. Figure 12.40 illustrates bottom view of a PGA package.

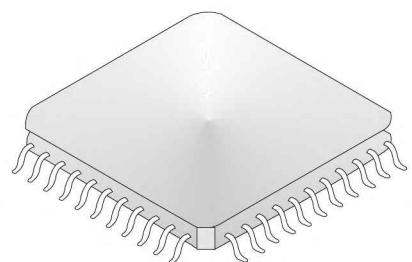


Fig. 12.39 *A QFP Package*

The ball grid array (BGA) and grid array (PGA) packaging use round balls, instead of posts. They are very small, hence more compact.

## Ds

lable from various manufacturers using CMOS EEPROM technology features of ALTERA's MAX 10, MAX 7000, and MAX

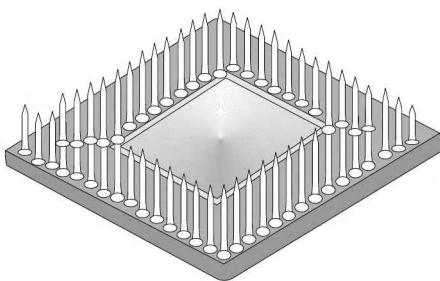


Fig. 12.40 **Bottom View of a PGA Package**

The CoolRunner-II CPLD family of Xilinx is discussed below. Data sheets for these devices manufactured by various manufacturers can be consulted for details.

## **ALTERA CPLDs**

<b>MAX 3000 A Device Family</b>	<b>MAX 7000 Device Family</b>	<b>MAX 9000 Device Family</b>
600-10,000	600-5,000	6,000-12,000
32-512	32-256	320-560
2-32	2-16	
34-208	36-164	168-216
4.5-7.5	6-12	10-15

Table 12.7 CoolRunner-II CPLD Family Packages and I/O Count

	<b>XC2C32A</b>	<b>XC2C64A</b>	<b>XC2C128</b>	<b>XC2C256</b>	<b>XC2C384</b>	<b>XC2C512</b>
QFG32*	21	—	—	—	—	—
VQ44	33	33	—	—	—	—
VQG44*	33	33	—	—	—	—
QFG48*	—	37	—	—	—	—
CP56	33	45	—	—	—	—
CPG56*	33	45	—	—	—	—
VQ100	—	64	80	80	—	—
VQG100*	—	64	80	80	—	—
CP132	—	—	100	106	—	—
CPG132*	—	—	100	106	—	—
TQ144	—	—	100	118	118	—
TQG144*	—	—	100	118	118	—
PQ208	—	—	—	173	173	173
PQG208*	—	—	—	173	173	173
FT256	—	—	—	184	212	212
FTG256*	—	—	—	184	212	212
FG324	—	—	—	—	240	270
FGG324*	—	—	—	—	240	270

\*The letter "G" as the third character indicates a Pb-free package.

macrocells is 32 and the largest number is 512. The maximum I/O count varies from 33–270. These devices are fabricated using 0.18 micron CMOS technology and are industry's very fast low power consumption CMOS CPLDs optimised for 1.8 V systems.

## Architecture

Figure 12.41 shows architecture of the CoolRunner-II CPLD family. There are a number of *function blocks* (FBs) each containing 16 macrocells (MCs). The FBs are interconnected with *advanced interconnect matrix* (AIM). The FBs use programmable logic array (PLA) configuration which allows all product terms to be routed and shared among any of the macrocells of the FB. The BSC path is the *JTAG boundary scan control path*. The BSC and ISP block has the JTAG controller and *in-system programming* (ISP) circuits.

## Function Block

Figure 12.42 shows the block diagram of a *function block* (FB) of the CoolRunner-II CPLD family. All the FBs are identical and have 40 entry sites for signals to arrive for logic creation and connection. The internal logic engine is a 56 product term (p-term) PLA. The CoolRunner-II CPLD family uses PLA which has many advantages over other CPLDs which normally use PAL structure. Most of these PAL based CPLDs rely on capturing unused p-terms from neighbouring macrocells to expand their p-terms tally, when needed. Therefore, this type of architecture gives variable timings for capturing different p-terms and also may not be using the available unused logic within the FB. The PLA based FB has the following options available:

- Any p-term can be attached to any OR gate inside the FB macrocells.
- Any logic function can have as many p-terms as needed within the FB, subject to an upper limit of 56.

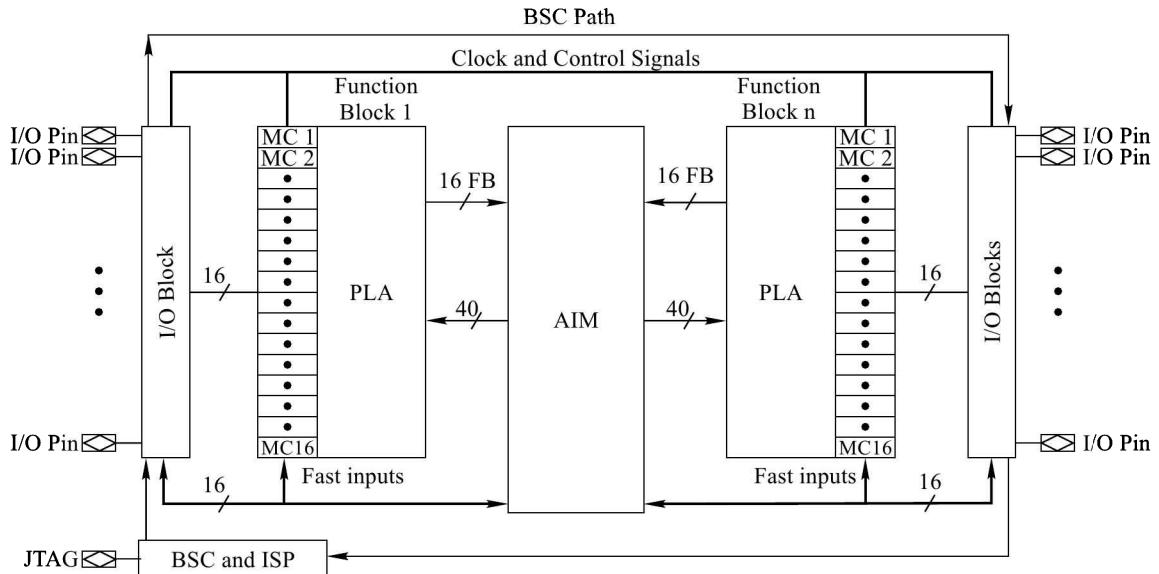


Fig. 12.41 CoolRunner-II Architecture

- p-terms can be re-used at multiple macrocells OR functions so that within a FB a particular logic product need only be created once, but can be used upto 16 times (number of MCs in a FB) within the FB.

Therefore, the advantages of using PLA based CPLDs are:

- Product terms can be shared by macrocells in a FB and as many functions as possible can be placed into FBs by the software.
- The functions need not share a common clock, common set/reset, or common output enable which allows full use of the PLA.
- Every p-term arrives with the same time delay incurred. Therefore, there are no cascade time adders for putting more p-terms in the FB.
- When all the p-terms have been used in the FB, then additional logic can be created by routing signals to another FB. This puts a small interconnecting timing penalty. The Xilinx design software handles all this automatically.

## Macrocell

The CoolRunner-II CPLDs macrocell is shown in Fig. 12.43. It consists of PLA, a FLIP-FLOP, number of multiplexers, and an EX-OR gate. The FLIP-FLOP has a clock enable and can be clocked on either edge

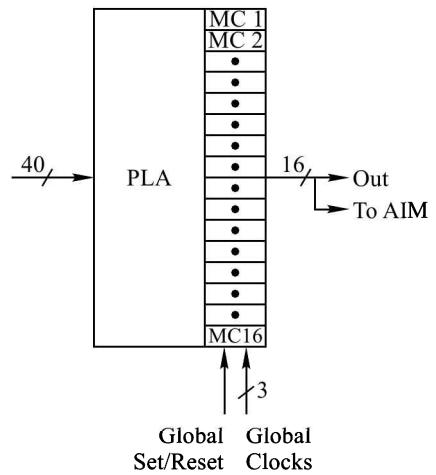
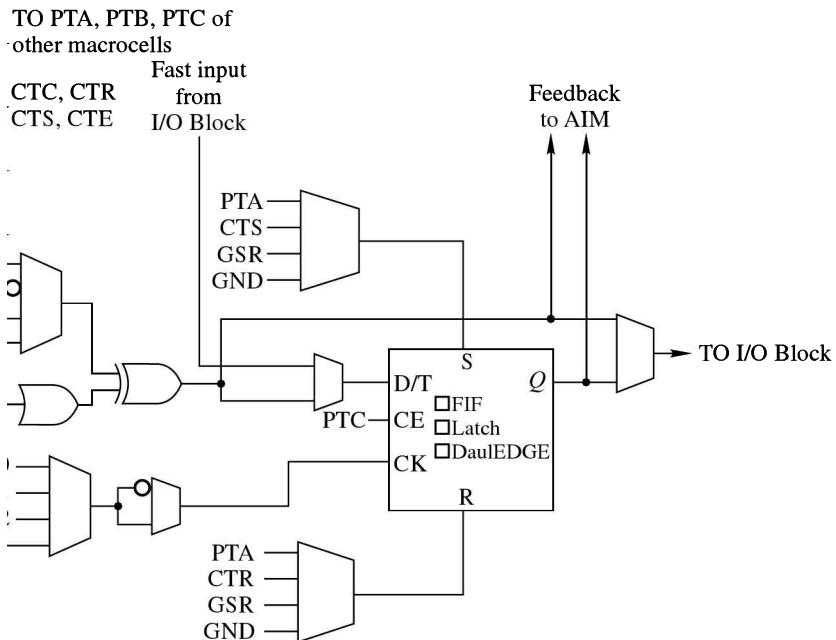


Fig. 12.42 CoolRunner-II Function Block

## *Modern Digital Electronics*



ed are marked X's. terms) are fed to the the outputs of the macrocell output A of Fig. 12.44 is 2.9 combined in one λ, the outputs of the

Output

$$I_1 \cdot I_2 + I_{40}$$

$$I_1 \cdot I_2 + I_{40}$$

Not used  
its corresponding to

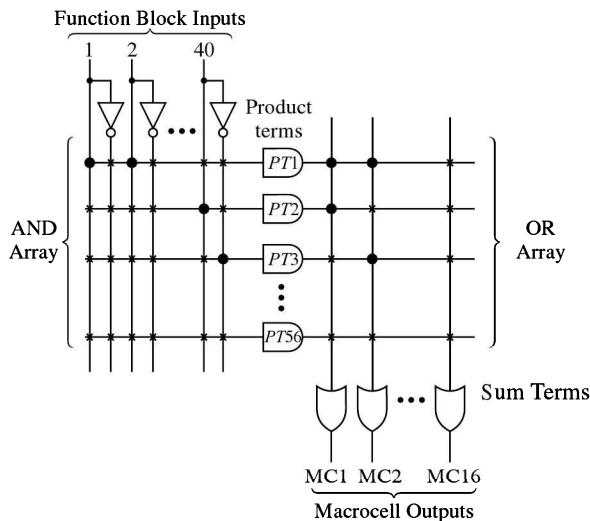


Fig. 12.44 PLA Architecture of CoolRunner-II Family

at signals and their own in consolidated form, the p-terms have

ed form. The top AND corresponds to 49 p-terms (PT1 through PT49), the next 5 (PT50 through PT53), and the remaining 3 p-terms (PT54, PT55, PT56) are C respectively. The four p-terms PT50 through PT53 correspond to four CT m, the intersections are marked as small dots (•) that are programmable.

From the conditions (iii) and (iv), we observe that the OR gate is avoided and therefore, it becomes a high speed path. In this way, p-term passes through the EX-OR gate rather than sum term which saves typically 0.3 ns delay time of the OR gate. This path is especially useful for microprocessor address decoding for fast operation. This approach can also be used to build faster shift registers, counters, and some simple state machines.

### *Design Security*

In CoolRunner-II CPLD designs can be secured during programming using four independent levels of security provided on-chip. This eliminates any electrical or visual detection of configuration patterns, which prevents either any accidental overwriting or pattern theft via readback. The security bits programmed can be reset only by erasing the entire chip.

### *In-system programming*

All CoolRunner-II CPLD parts are 1.8 V in-system programmable. They derive their programming voltage and currents from the 1.8 V  $V_{CC}$  on the part.

## 12.6 FIELD-PROGRAMMABLE GATE ARRAY (FPGA)

The programmable logic devices(SPLDs and CPLDs) are based on similar basic architecture—the programmable array logic (PAL) or the programmable logic array (PLA). Over the years, programmable arrays have increased in size and complexity, and highly configurable output macrocells have been added to enhance their flexibility and expandability. To increase the effective size and to add more functionality in a single programmable device, alternative architectures have been developed which are known as *field-programmable gate arrays* (FPGAs). The logic densities of FPGAs are much higher than those of CPLDs. They range in size from a few thousands to hundreds of thousands equivalent gates. From modern standards digital circuits with hundreds of thousands of gates is not too large. FPGA devices support implementation of relatively large complex logic circuits.

The FPGAs do not contain AND, OR planes, instead they provide logic blocks for implementation of the required digital functions.

An FPGA is composed of a number of relatively independent configurable logic blocks (CLBs), configurable I/O blocks, and programmable interconnection paths (known as routing channels). All the resources of the device are uncommitted and that these must be selected, configured and interconnected by a user to form a logic circuit for his application. The basic architecture of an FPGA is shown in Fig. 12.45.

There are a number of manufacturers of FPGA devices. The various families of FPGAs manufactured by different manufacturers differ primarily in the number of logic modules (from few hundreds to hundreds of thousands), supply voltage range, power consumption, speed, architecture, process technology, number of pins, and type of packages, etc. Some of the major manufacturers of FPGAs manufacturing a wide range of products to suit various types of requirements are given in Table 12.8.

The basic FPGA architecture consists of an array of configurable logic blocks (CLBs). The logic blocks are surrounded by configurable input/output blocks. There are rows and columns of programmable interconnection paths. The I/O blocks can be individually configured as input, output, or bidirectional.

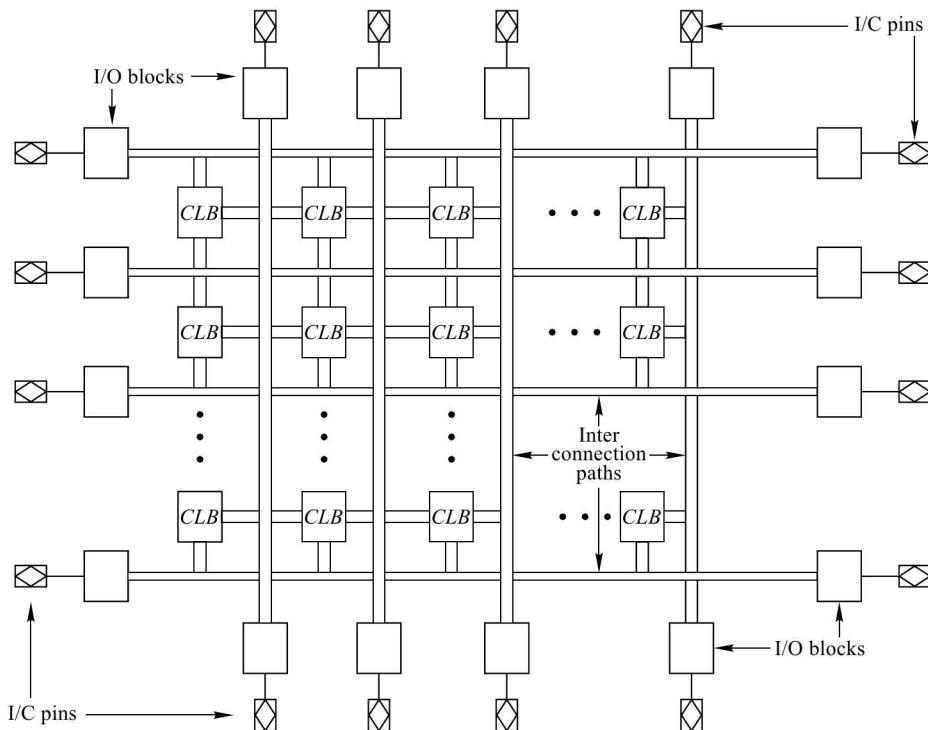


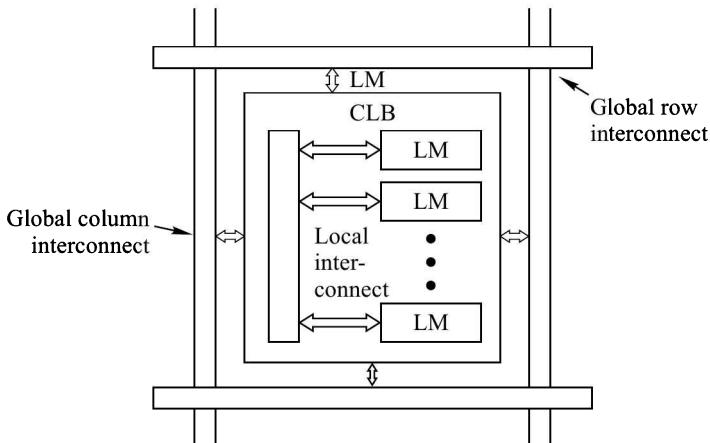
Fig. 12.45 Basic Architecture of FPGA

Table 12.8 **FPGA Manufacturers**

Manufacturer	FPGA products	www locator
Actel	Act 1, 2, 3, IGLOO	<a href="http://www.actel.com">http://www.actel.com</a>
Altera	Flex 6000, 8000, 10K, APEX 20K, Stratix II, III, IV	<a href="http://www.altera.com">http://www.altera.com</a>
Atmel	AT6000, AT40K	<a href="http://www.atmel.com">http://www.atmel.com</a>
Lucent	Lattice SC, ECP2, XP2	<a href="http://www.lucent.com">http://www.lucent.com</a>
Quick Logic	PASIC 1, 2, 3, Eclipse	<a href="http://www.quicklogic.com">http://www.quicklogic.com</a>
Vantis	VF1	<a href="http://www.vantis.com">http://www.vantis.com</a>
Xilinx	XC4000, XC5200, Virtex, Spartan	<a href="http://www.xilinx.com">http://www.xilinx.com</a>

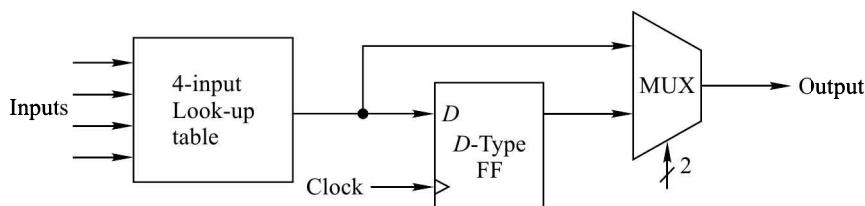
### Configurable Logic Blocks

There are a number of configurable logic blocks (CLBs) in an FPGA organized as an array of rows and columns. The logic blocks are connected to the I/O blocks through common row/column programmable interconnects. The common row/column interconnects are known as global interconnects. A logic block consists of a number of logic modules (LMs). The logic modules are the basic logic elements in an FPGA. The logic modules within a CLB are connected through local programmable interconnects. Figure 12.46 shows a CLB.

Fig. 12.46 **Basic Configurable Logic Block**

## Logic Module

A logic module consists of an LUT (look-up table), a D-type FLIP-FLOP and a multiplexer (MUX). Most of the FPGAs are based on 4-input LUT. Figure 12.47 shows a block diagram of a logic module with 4-input LUT. Output of the LUT becomes the output of the logic module either directly or through D-type FLIP-FLOP. Thus, the output can be configured for combinational or registered (i.e., through FLIP-FLOP).

Fig. 12.47 **Block Diagram of a Logic Module**

## Look-Up Table (LUT)

An LUT (look-up table) consists of a programmable memory and it can be used to generate logic function in SOP form. For example, from Table 11.4, we can see that this table is similar to a 4-input and one output logic circuit's truth table. Therefore, a memory can generate canonical product terms. Figure 12.48 shows a block diagram of an LUT. It consists of a memory and a multiplexer (MUX). Let us assume the memory contents as given in the figure. Since, it is an 8-bit memory, therefore an 8 : 1 multiplexer is required. If the 3-bit logical input is  $A_2 A_1 A_0$ , then

$$Y = \bar{A}_2 \bar{A}_1 A_0 + \bar{A}_2 A_1 \bar{A}_0 + A_2 \bar{A}_1 \bar{A}_0 + A_2 A_1 A_0$$

The look-up table in most of the commercially available FPGAs is 4-input circuit. Larger LUTs would allow for more complex logic to be performed per logic block, thus reducing the wiring delay between blocks as fewer blocks would be needed. This will require larger multiplexer and an increased chance of waste if all of the functionality of the larger LUTs were not to be used. On the other hand, smaller look-up tables

may require a design to consume a large number of logic blocks, thus increasing wiring delay between blocks while reducing per logic block delay. Therefore, 4-input LUT structure makes the best trade-off between area and delay for a wide range of circuits. However, some of the latest FPGAs have been designed with 6-input LUT structure. The Xilinx Virtex-5 family of FPGAs have used 6-input LUT technology.

### FPGA Cores

A commercially available FPGA may have all the CLBs available for a user to program them according to his requirements. However, some FPGAs are available in which a portion of CLBs is used by the manufacturer to provide a specific built-in function that can not be changed by a user. This is referred to as *hard-core* logic. The hard-core logic approach has the following advantages:

- The hard-core logic may normally be implemented using lesser number of CLBs than the same logic being programmed by a user. This saves the available chip resources, i.e., programmable area to the user.
- There is saving in the development time of user in developing a digital system.
- The built-in hard-core function can be thoroughly tested by the manufacturer, thereby increasing its reliability.

Some of the commonly used functions, such as microprocessors, standard I/O interfaces, and digital signal processors (DSPs) are available in hard-core FPGAs. Since the hard-core designs are developed by the manufacturers', therefore, these are the manufacturers' *intellectual property* (IP).

In case, the manufacturer's programmed function has some programmable features also, it is known as a *soft-core* function. Some intellectual properties may be combination of both hard-core and soft-core functions. The FPGAs containing either or both hard-core and soft-core embedded processors and other functions are known as the *platform FPGA* because they can be used to implement an entire system without the need for any external devices.

### FPGA Process Technology

There are different process technologies used by various FPGA manufacturers. These are:

- SRAM technology—It is based on static memory technology. The CMOS devices are fabricated by this technology and these are in-system programmable (ISP) and are reprogrammable.
- Antifuse technology—The antifuse technology developed by Actel Corporation of America is used for processing CMOS FPGAs which are one-time programmable (OTP).
- EPROM technology—It may be one-time programmable (OTP) or ultraviolet erasable type of CMOS device.
- EEPROM technology—It is electrically erasable which may be in-system programmable type or off-system programmable type CMOS technology.
- Flash technology—It is flash-erase CMOS technology which may be in-system programmable type.
- Fuse technology—It is a bipolar one-time programmable FPGA.

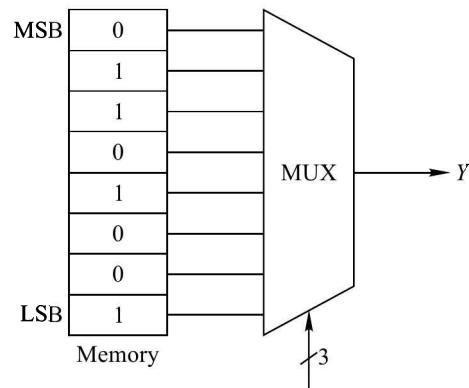


Fig. 12.48 Block Diagram of an LUT

## ***Configuration Memory***

Most of the modern FPGAs use SRAM. The SRAM bits are used to hold the user defined configuration values.

### **12.6.1 Xilinx Virtex FPGAs**

The field-programmable gate array was invented by Xilinx in 1984 and they are the leading manufacturers of FPGA devices. There are two major lines of Xilinx FPGAs, Spartan and Virtex. The Extended Spartan-3A and the Virtex-5 families are the latest available FPGAs from Xilinx.

#### ***Configurable Logic Blocks***

There are a number of CLBs organised as an array. Each CLB contains multiple basic logic units called logic cells (LCs). The logic cells are same as the logic modules (LMs) discussed earlier. The logic cells are LUT based. Each logic cell consists of an LUT, a FLIP-FLOP, and a multiplexer. In the Virtex-4 family, XC4VLX200 FPGA has CLB array of 192X116 containing 200,448 logic cells. The number of LUTs and the internal registers, i.e., FLIP-FLOPs is 178,176 each. In the Virtex series of FPGAs there is a concept of *slice*. A CLB of Virtex-4 family of FPGAs consists of 89,088 slices. There are two LUTs and two FFs in each slice. A CLB is made up of four slices.

The Virtex-5 family of FPGAs has a maximum of  $240 \times 108$  array of CLBs, 51,840 slices, each slice contain four LUTs and four FFs. A CLB of Virtex-5 family FPGAs is made up of two slices. The function generators are configurable as 6-input LUTs or dual-output 5-input LUTs.

In addition to function generators and storage elements (FFs), each slice in both of the above FPGA families, contain arithmetic logic gates, large multiplexers, and fast carry look-ahead chain.

#### ***Configuration***

The devices of Virtex family are configured by loading the bitstream into internal configuration memory in various modes.

#### ***IP Cores***

In these devices, there are IP cores for commonly used complex functions including DSP, bus interfaces, processors, and processor peripherals.

#### ***Process Technology***

The Virtex-4 family of FPGAs are produced using 90-nm copper CMOS process technology, whereas the Virtex-5 family of FPGAs are produced using 65-nm copper CMOS process technology.

### **12.6.2 Altera Stratix FPGAs**

Altera Corporation of America is producing wide range of FPGAs to meet different requirements. The Stratix series of FPGAs started in 2002 with the introduction of Stratix family. Subsequently, Stratix GX(2003), Stratix II (2004), Stratix II GX (2005), Stratix III (2006) and Stratix IV (2008) were introduced in the years mentioned in parentheses along the family. The basics of Stratix II family of FPGAs have been chosen for discussion here.

Stratix FPGAs contain a two dimensional row- and column-based architecture to implement custom logic. A series of column and row interconnects of varying length and speed provides signal interconnects between logic array blocks (LABs) and various other blocks, such as memory block structures and digital signal processing (DSP).

### **Logic Array Block (LAB)**

The configurable logic block (CLB) is called as logic array block in Altera FPGAs. Each LAB consists of eight adaptive logic modules (ALMs). An ALM is the basic building block of logic for efficient implementation of user logic functions. LABs are grouped into rows and columns across the device. In addition to eight ALMs, each LAB contains carry chains, shared arithmetic chains, LAB control signals, local interconnect, and register chain connection lines. The LAB structure is shown in Fig. 12.49. The local interconnect transfers signals between ALMs in the same LAB. The local interconnect is driven by column and row interconnects, ALM outputs in the same LAB, and neighbouring LABs from the left and right through the direct link connection. The direct link connection feature helps in minimising the use of row and column interconnects which increases the performance and flexibility. Multiple LABs are linked together via the global row and column interconnects.

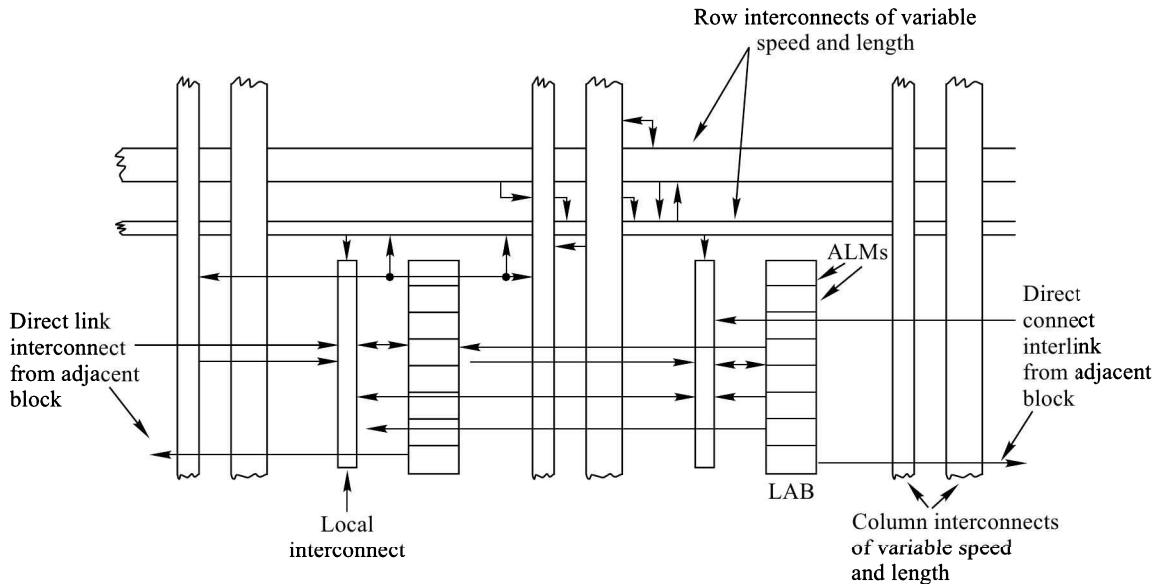


Fig. 12.49      **Stratix II LAB Structure**

### **Adaptive Logic Modules (ALMs)**

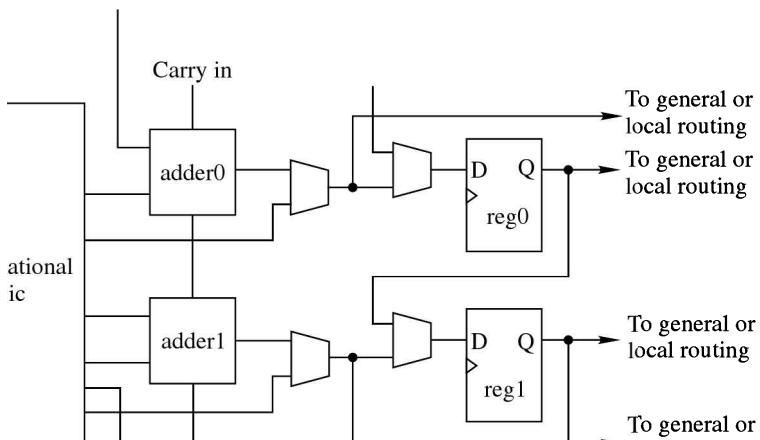
The basic building block of logic in these FPGAs is the adaptive logic module (ALM). Each ALM contains a variety of look-up table (LUT)-based resources that can be divided between two adaptive LUTs (ALUTs). There are eight inputs in an ALM and one ALM can be used to implement various combinations of two functions including any function of up to six inputs and certain seven input functions.

In addition to the two ALUTs, each ALM contains two programmable registers (D-type FFs), two dedicated full adders, a carry chain, a shared arithmetic chain, and a register chain. Using these resources, the ALM can

## *Modern Digital Electronics*

functions and shift registers. Each ALM drives all types of interconnects: local, / chain, shared arithmetic chain, register chain, and direct connect interlinks. agram of the ALM. The eight data inputs are: data a, data b, data c, data d, data f1. The first four data inputs, data a, data b, data c, and data d can be shared ta e0 and data f0 are dedicated to adder 0 and reg 0, and data e1 and data f1 eg 1.

outputs (combinational and registered) that drives the global and local routing output can be either LUTs output or adder output. For combinational output, e registered output is obtained via the register. The two sets of outputs are



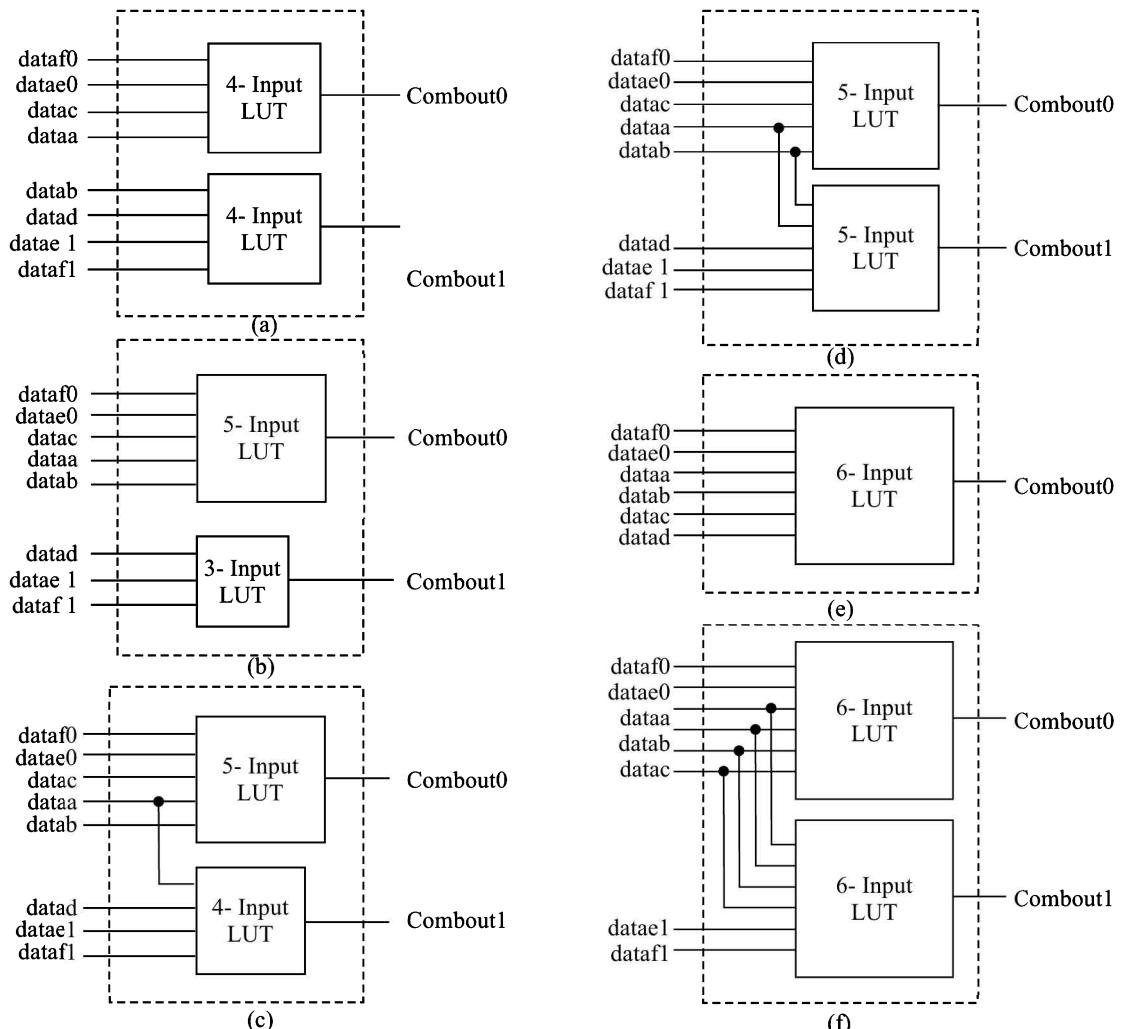


Fig. 12.51 *Various Combinations of Logic Functions Generated using an ALM (a) Two 4-Input Functions (b) 5 and 3 Input Functions (c) 5 and 4 Input Functions (d) Two 5-Input Functions (e) 6-Input Function (f) 6 and 2 Input Functions*

**Extended LUT Mode** Some specific seven-input functions can be implemented by making two five input functions, sharing four inputs, and applying these two five-input functions to a 2 : 1 multiplexer. Figure 12.52 shows its implementation.

**Arithmetic Mode** The arithmetic mode is used for implementing adders, counters, accumulators, comparators, and parity functions. In arithmetic mode, an ALM uses two sets of two four-input LUTs alongwith two dedicated full adders.

**Shared Arithmetic Mode** In shared arithmetic mode, an ALM can implement a three-input add. In this mode, the ALM is configured with four 4-input LUTs.

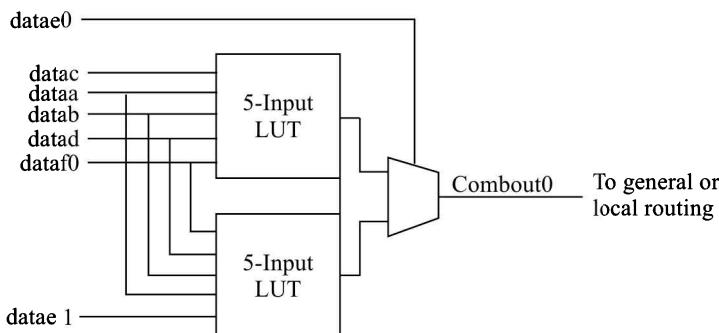


Fig. 12.52     ***Extended LUT Mode of an ALM***

## SUMMARY

The basic concepts of programmable logic devices and programmable gate arrays have been introduced. With the development of these devices, it has become possible to design complex digital systems. However, high-level design techniques and computer-aided tools are required to produce efficient PLD and FPGA implementations. Testing of PLD and FPGA implementations also require computer-assisted test tools. The design and test tools for these programmable devices are beyond the scope of this book.

The emergence of these devices has revolutionized the design of digital systems similar to the emergence of microprocessor. The programmable logic concept has emerged as a technology that has given the power to design one's own custom ICs which cannot be copied by others.

The design of embedded systems in small size has become possible by using FPGAs with intellectual properties, such as microprocessors, and DSP, etc. The embedded systems without using external devices and battery operated digital systems have proved very useful because of the availability of low-power FPGAs with intellectual properties.

## GLOSSARY

***Antifuse*** A programmable element invented by Actel Corporation named as PLICE (programmable low-impedance circuit element).

***ASIC (Application specific integrated circuit)*** An IC configured by the manufacturer as per the specifications supplied by the user for a specific application.

***BGA (Ball grid array)*** An IC package used for ICs requiring large number of pins. The pins are of small round ball shapes.

***CLB (Configurable logic block)*** A logic block in an FPGA consisting of logic modules and local programmable interconnect that is used to connect logic modules within the CLB to generate required logic functions.

***Configuration memory*** A memory in an FPGA meant to store bit pattern for configuring the FPGA.

***CPLD (Complex programmable logic device)*** A programmable logic device containing a large number of equivalent gates.

ant for general purpose specific functions.

**(gate array)** A programmable logic device containing very large number of

**(logic array)** A PLA programmable by the user.

A type of configurable PAL.

Logic provided by the manufacturer in an FPGA for some commonly required user's intellectual property (IP).

Hard-core and soft-core provided by a manufacturer in an FPGA.

Arrangement for providing connections between various logic elements, logic

) A technique for programming large programmable logic devices such as the device is programmed on the circuit board itself rather than in a programming

p) An IEEE standard for ISP.

ame as configurable logic block (CLB).

**(ule)** A basic unit of logic in an FPGA that usually contains an LUT (look-up xer.

of memory that is programmable for creating the desired logic function in SOP

unction provided in large programmable logic devices which are used to design

## *Modern Digital Electronics*

### S

of programmable \_\_\_\_\_ arrays.

tal circuit of 32 variables \_\_\_\_\_ PLAs with 16 inputs and 8 outputs are

if programmable \_\_\_\_\_ gates.

d by the \_\_\_\_\_.

le security of a digital circuit \_\_\_\_\_ design is preferred.

circuits of the complexity of a few hundreds of gates \_\_\_\_\_ is preferred.

ins, the number of inputs to each of the AND gates is \_\_\_\_\_.

As \_\_\_\_\_ programming technique is the most suitable.

umming uses \_\_\_\_\_ IEEE standard.

for ICs with more than 200 pins is \_\_\_\_\_.

.

rogramming \_\_\_\_\_ FPGAs.

be used for digital circuits requiring more than 200,000 equivalent gates.

re programmed using \_\_\_\_\_ tools.

gital circuit ASIC is \_\_\_\_\_ expensive than designing using FPGA.

and PAL devices have \_\_\_\_\_ outputs.

orated between the OR gate and output buffer in a registered PAL.

gic device, an \_\_\_\_\_ is provided for programming the polarity of output.

d using \_\_\_\_\_ technology.

\_\_\_\_\_ -system programmable.

.....

## PROBLEMS

- 12.1** Design a BCD-to-Excess-3 code converter using a (a) PROM, (b) PLA, (c) PAL.
- 12.2** Design an Excess-3-to-BCD code converter using a (a) PROM, (b) PLA, (c) PAL.
- 12.3** Design a BCD-to-seven segment decoder using a (a) PROM, (b) PLA, (c) PAL.
- 12.4** How will you obtain 16-bit output word using 82 S100 FPLAs?
- 12.5** Explain the function of the circuit of Fig. 12.13.
- 12.6** What is meant by the term ‘architecture of a PLD’? Give some suitable examples.
- 12.7** For 16L8 PAL device shown in Fig. 12.18, find out the input lines (columns) corresponding to the inputs  $I_1$  through  $I_{10}$  and  $IO_2$  through  $IO_7$ .
- 12.8** For 16R6 Registered PAL shown in Fig. 12.19, find out the input lines (columns) corresponding to the inputs  $I_1$  through  $I_8$ ,  $IO_1$ ,  $IO_8$ , and feedback connections corresponding to D-FFs in  $O_2$  through  $O_7$  lines.
- 12.9** For the output macrocell of 18CV8 PEEL device shown in Fig. 12.31, determine the twelve output configurations for the select inputs  $A = 0, 1$ ;  $B = 0, 1$  and  $CD = 00, 10, 11$  of the multiplexers.
- 12.10** In the PAL 16L8 device, program the output as  
 (a) Always enabled  
 (b) Always disabled  
 (c) Enabled by a product term  $ABCD\bar{E}\bar{F}GH$  of eight variables.
- 12.11** A 2-input look-up table (LUT) of an FPGA’s logic block is shown in Fig. 12.53. Determine its truth table.

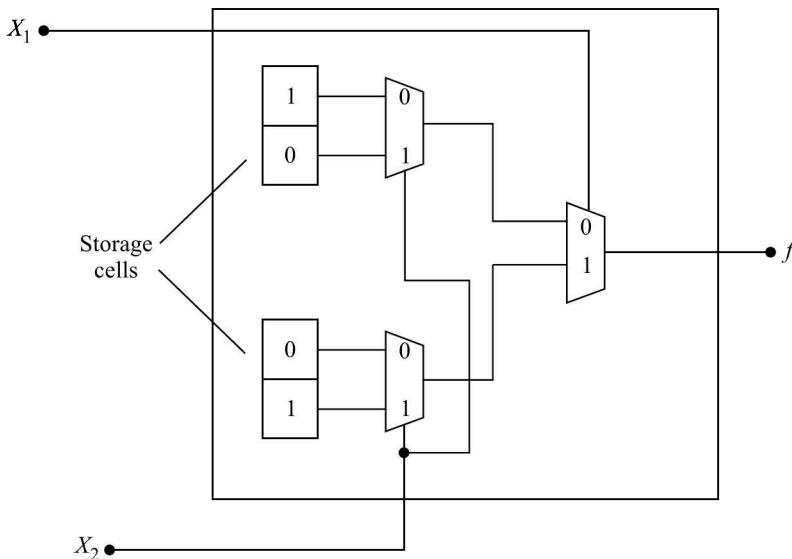


Fig. 12.53

- 12.12** A 3-input LUT is shown in Fig. 12.54. Determine the bits to be stored in the storage cells to realize the logic function.

$$f = \bar{x}_1 \bar{x}_2 x_3 + \bar{x}_1 x_2 \bar{x}_3 + x_1 \bar{x}_2 \bar{x}_3 + x_1 x_2 x_3$$

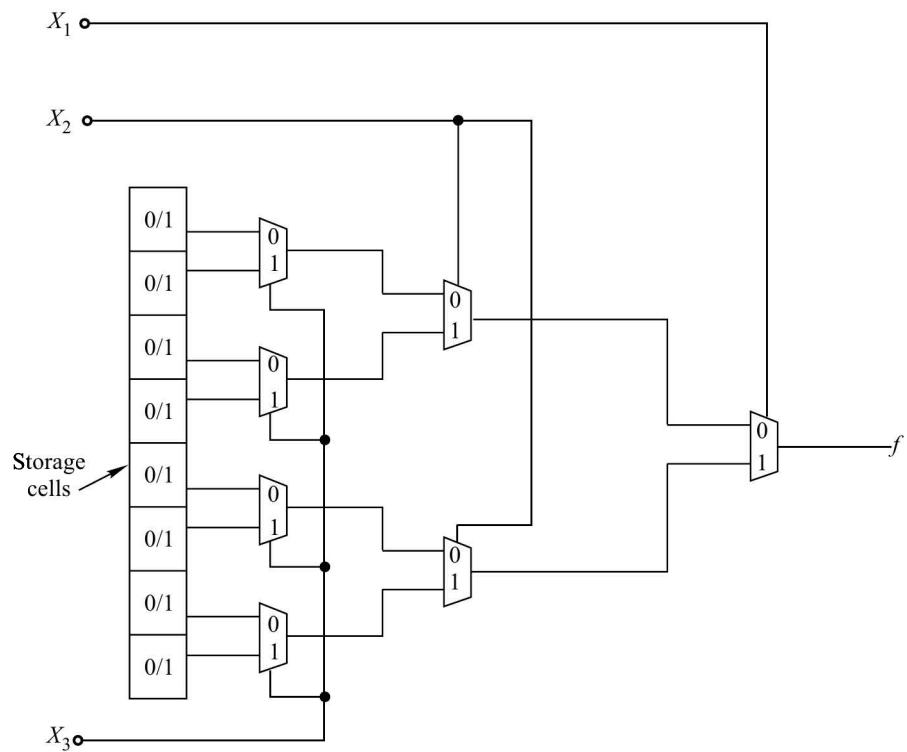


Fig. 12.54

# CHAPTER 9

# TIMING CIRCUITS

## 9.1 INTRODUCTION

Most digital systems require some kind of a timing waveform, for example, a source of trigger pulses is required for all clocked sequential systems. In digital systems, a rectangular waveform is most desirable (unlike analog systems where sinusoidal signals are often used). The generators of rectangular waveforms are referred to as *multivibrators*.

There are three types of multivibrators. These are:

1. Astable (or free-running) multivibrator,
2. Monostable multivibrator (or one-shot), and
3. Bistable multivibrator (or FLIP-FLOP).

An astable multivibrator is nothing but an oscillator which generates rectangular waveform. It has two quasi-stable states and does not require any triggering, hence it is referred to as a *free-running* multivibrator. This is used as a source of clock pulses in sequential circuits.

A monostable multivibrator has one stable state, i.e. under steady-state condition its output is fixed; it is in either the LOW or the HIGH state. When the circuit is triggered with an externally applied pulse it goes into the other state, i.e. if it was in LOW state, it goes to the HIGH state and if it was in the HIGH state, it goes to the LOW state. The circuit remains in this state for a time duration depending upon the values of the elements used in the circuit. This state of the circuit is referred to as *quasi-stable state* since it recovers back to the stable state without any external triggering pulse. The width of the trigger pulse is very small and the width of the output pulse depends upon the time duration for which the circuit remains in the quasi-stable state. This circuit is also referred to as a *one-shot* since one trigger pulse produces only one pulse but of a different pulse width. This circuit is very useful because it can generate a relatively long pulse (tens of ms) from a narrow pulse, therefore, it is also known as *pulse stretcher*.

For example, a microprocessor may signal an output device to print something by transmitting a pulse. The electromechanical output device in general is slower than the microprocessor and hence would require the signal pulse for a longer duration. This is achieved by the interface circuitry consisting of a monostable multivibrator.

A multivibrator circuit in which both the states are stable is referred to as a *bistable* multivibrator or FLIP-FLOP. This circuit makes transitions from one stable state to another only when a triggering pulse is applied. These are often used as memory elements in digital systems and have been discussed extensively in Chapters 7 and 8.

Until a few years ago, multivibrators were designed using discrete devices like vacuum triodes, BJTs, FETs, etc. which have become obsolete now because of the availability of various ICs. Therefore, we shall be dealing with multivibrator circuits using various ICs only. The ICs used are:

1. Logic gates,
2. OP-AMPS,
3. Monostable multivibrators, and
4. Timers.

## 9.2 APPLICATIONS OF LOGIC GATES IN TIMING CIRCUITS

Logic gates can be used for generating the pulses required in digital systems. These circuits are easy to design and analyse but due to the lack of precision, their applications are rather restricted.

### 9.2.1 Free-Running Multivibrator

A simple, free-running multivibrator is shown in Fig. 9.1a. This circuit works on the same principle as an *RC* phase-shift oscillator. In this circuit, the phase shift is provided by the propagation delay time of the inverters.

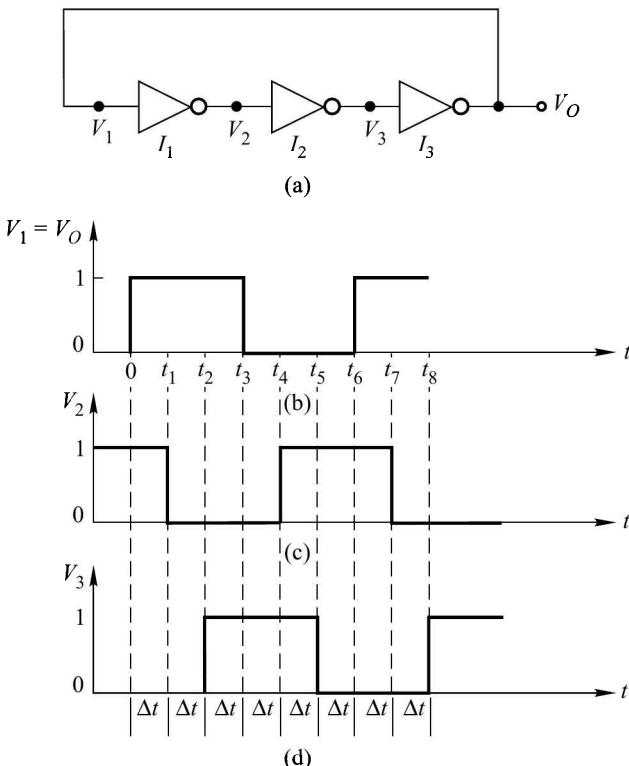


Fig. 9.1 (a) Free-Running Multivibrator Using Inverters. Waveforms of (b)  $V_1 = V_O$  (c)  $V_2$ , and (d)  $V_3$

Suppose, for example, the input  $V_1$  to inverter  $I_1$  makes a transition from logic 0 to logic 1 at  $t = 0$ . At  $t_1$  (after a propagation delay time  $\Delta t$ ) its output changes from logic 1 to logic 0. This will cause the output of inverter  $I_2$  to change from logic 0 to logic 1 at  $t_2$ . At  $t_3$ , the output  $V_o$  and the voltage  $V_1$  will change from logic 1 to logic 0. This process will go on indefinitely. The various voltage waveforms are illustrated in Fig. 9.1b, c, and d. The time period  $T$  of the output square wave is given by

$$T = 6\Delta t \quad (9.1)$$

For TTL gates,  $\Delta t$  is of the order of 10 ns, therefore, the circuit's operating frequency is of the order of 16 MHz.

In this circuit, we have no control over the frequency of the square wave and it is difficult to determine the exact propagation delay time of the logic gate and hence the frequency of the square wave. Therefore, it cannot be used in a system calling for precise and stable operating frequencies. However, because of its simplicity, it is useful whenever it is necessary to get high frequency trigger pulses at a very low cost.

It is possible to have some control over the frequency of the square wave by using timing elements—a resistance and a capacitance (Problem 9.1).

## 9.2.2 Monostable Multivibrator

A monostable multivibrator circuit using gates is shown in Fig. 9.2. Under steady-state condition, the voltage across the resistance  $R$ ,  $v_R$  is 0, i.e. the input to the inverter is at logic 0, hence its output  $v_o$  is at logic 1. Now, if the trigger input is at logic 1, then both the inputs of the NAND gate are at logic 1 and its output is at logic 0. Hence,  $v_o$  in logic 1 state is the stable state of this circuit. It can be verified that  $v_o$  in 0 state is not a stable state (Problem 9.2). When the trigger input goes from logic 1 to logic 0, the output of the NAND gate goes to logic 1 and hence,  $v_R$  corresponds to logic 1.

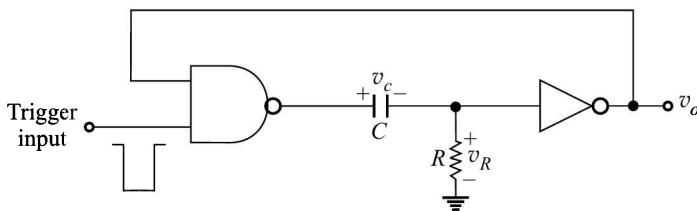
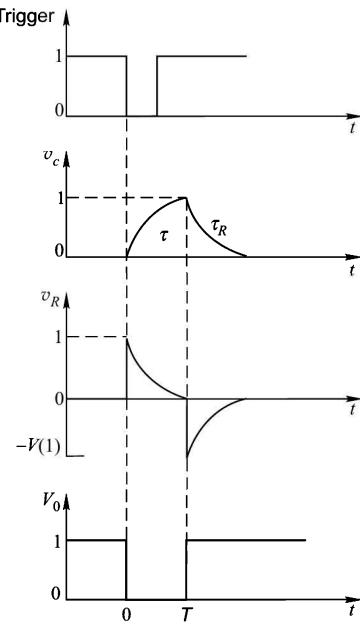


Fig. 9.2      *Monostable Multivibrator*

Therefore, the output  $v_o$  of the inverter goes to logic 0. The 0 at  $v_o$  which is returned to one input of NAND gate holds the output of the NAND gate at logic 1. As voltage  $v_c$  increases exponentially with the time constant  $\tau = RC$ , the voltage  $v_R$  decreases and goes to logic 0 driving the output  $v_o$  to logic 1. As soon as both the inputs of the NAND gate become 1, its output is 0. The capacitor now discharges through the output circuit of the NAND gate and the circuit comes to the stable state. The circuit should not be triggered during this recovery interval. The various waveforms are shown in Fig. 9.3.

It is not possible to obtain the exact value of the pulse duration  $T$  because of the somewhat uncertain values of the input/output voltages corresponding to 0/1 level and the propagation delay time of the gates. However, the circuit is useful because of its simplicity and low cost.

Fig. 9.3 *Waveforms of Monostable Multivibrator*

### 9.3 OP AMP AND ITS APPLICATIONS IN TIMING CIRCUITS

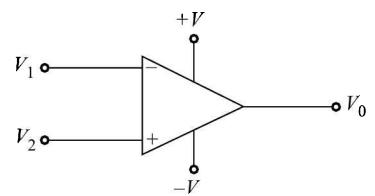
An *operational amplifier*, popularly known as OP AMP is a very high gain d.c. amplifier. Originally, it was designed using vacuum tubes to perform mathematical operations such as addition, multiplication by a constant, differentiation, integration, etc. and it was a basic building block of analog computers. Now-a-days, it is available as a linear IC and, because of its low cost, versatility, and reliability, has become very popular and finds applications in the generation of waveforms like square, triangular, pulse, sweep, staircase, etc.

An OP AMP is shown in Fig. 9.4. It has two inputs, labelled  $-$  and  $+$ , and are referred to as *inverting*, and *non-inverting* inputs, respectively. The input  $V_1$  is applied between the inverting input terminal and ground, and input  $V_2$  is applied between the non-inverting input terminal and ground. The ground terminal is not there in this circuit but is the ground terminal of the supplies  $+V$  and  $-V$ . The output,  $V_o$ , is obtained between the output terminal and ground. The input terminals are called *differential inputs* and the output is *single-ended*. The output voltage depends on the difference voltage  $V_i = V_1 - V_2$  and is given by

$$V_o = A_v \cdot V_i \quad (9.2)$$

where  $A_v$  is the voltage gain of the amplifier, which has a large negative value (ideally,  $A_v \rightarrow -\infty$ ).

From Eq. (9.2), we observe that the polarity of the output voltage is the same as the polarity of the non-inverting input voltage and is opposite (or inverted) to the polarity of the inverting input voltage.

Fig. 9.4 *Block Diagram of an OP AMP*

Since the value of  $A_v$  is extremely large, often 200,000 or more, therefore, the input voltage  $V_i$  is extremely small ( $V_i \approx 0$ ) and the output voltage  $V_o$  can never exceed positive or negative saturation voltages  $+V_{sat}$  and  $-V_{sat}$ , respectively. These voltages are normally within about 2V of  $+V$  and  $-V$ . For example, if supply voltages are  $\pm 15$  V, then  $+V_{sat} = +13$  V and  $-V_{sat} = -13$  V. Thus, the output is restricted to a peak-to-peak swing of  $\pm 13$  V.

### 9.3.1 OP AMP Comparator

The OP AMP can be used as an analog comparator to compare two analog signals. The analog signals to be compared are applied at the two inputs and the polarity of the output voltage indicates the comparison. The magnitude of the output voltage is  $V_{sat}$ . This is the basic building block required for non-sinusoidal waveform generators.

#### Example 9.1

Find the output waveform of an OP AMP under the following conditions:

- Inverted input terminal connected to ground and a sinusoidal signal of 4 V peak at the non-inverting input.
- Non-inverting input terminal connected to ground and a sinusoidal signal of 4 V peak at the inverting input.
- +3 V at the inverting input and a sinusoidal signal of 5 V peak at the non-inverting input.

#### Solution

- The circuit with inverting input connected to ground and the voltage source connected at the non-inverting input is shown in Fig. 9.5a. The input and the output waveforms are shown in Fig. 9.5b and c, respectively. When the input passes through 0 V while changing from negative to positive, the output changes from  $-V_{sat}$  to  $+V_{sat}$ . Similarly, when the input passes through 0 V while changing from positive to negative, the output changes from  $+V_{sat}$  to  $-V_{sat}$ .
- With the source connected at the inverting input and the non-inverting input connected to ground, the circuit, and the input and output waveforms are shown in Fig. 9.6.

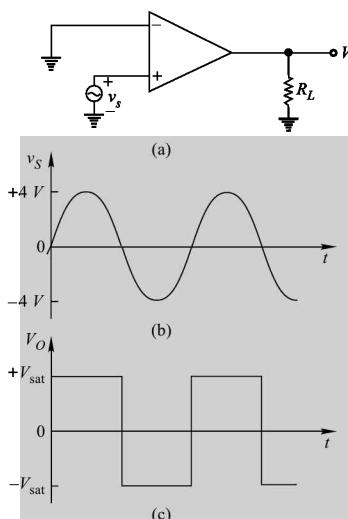


Fig. 9.5 (a) Comparator for Ex. 9.1a (b) Input Waveform (c) Output Waveform

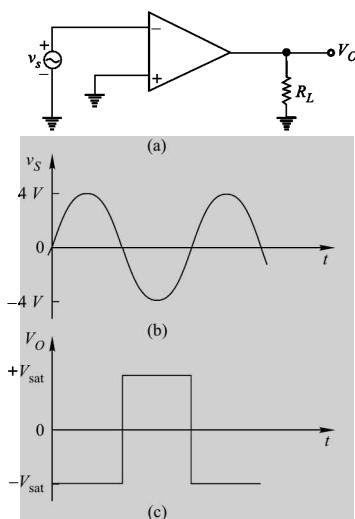


Fig. 9.6 (a) Comparator for Ex. 9.1b (b) Input Waveform (c) Output Waveform

- (c) In this case a reference voltage,  $V_{ref} = 3$  V is maintained at the inverting input terminal. Whenever the voltage  $v_s$  is greater than  $V_{ref}$ , the output is  $+V_{sat}$  and whenever it is less than  $V_{ref}$ , the output is  $-V_{sat}$ . The circuit diagram, and the input and output waveforms are shown in Fig. 9.7.

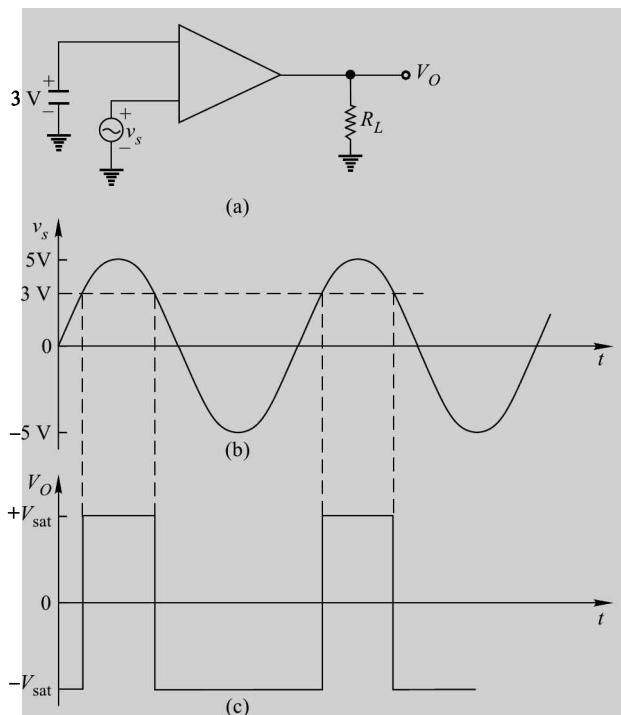


Fig. 9.7 (a) Comparator for Ex. 9.1c (b) Input Waveform (c) Output Waveform

In some practical applications, the input voltage to the comparator may approach the reference voltage very slowly and/or may oscillate around  $V_{ref}$ . In such a situation,  $V_O$  either would not switch quickly from one saturation voltage to the other or would oscillate between  $+V_{sat}$  and  $-V_{sat}$ . This oscillation may also occur due to ringing caused from the fast voltage transitions or due to the presence of noise on wires leading to the OP AMP's input terminals.

### Example 9.2

- (a) If the triangular waveform shown in Fig. 9.8a is applied to the circuit of Fig. 9.6a, what will be the output waveform? (b) If noise is present as shown in Fig. 9.9a, find the effective input and output voltage waveforms.

### Solution

- (a) In this circuit the reference voltage  $V_{ref} = 0$  and, therefore, the output makes a transition whenever the inputs passes through 0 V. The output waveform is shown in Fig. 9.8b.  
 (b) For simplicity, we assume the noise voltage to be sinusoidal. The waveform of effective input,  $v_i$  (combination of triangular waveform and noise) is shown in Fig. 9.9b and the corresponding output waveform is shown in Fig. 9.9c, which shows false transitions in the output waveform due to noise.

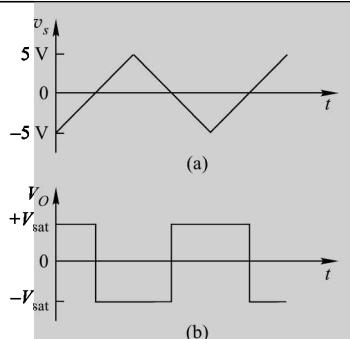


Fig. 9.8 (a) Triangular Voltage Waveform Applied to the Comparator of Fig. 9.6a (b) Its Output Waveform

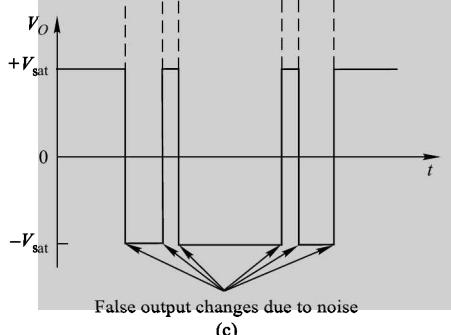
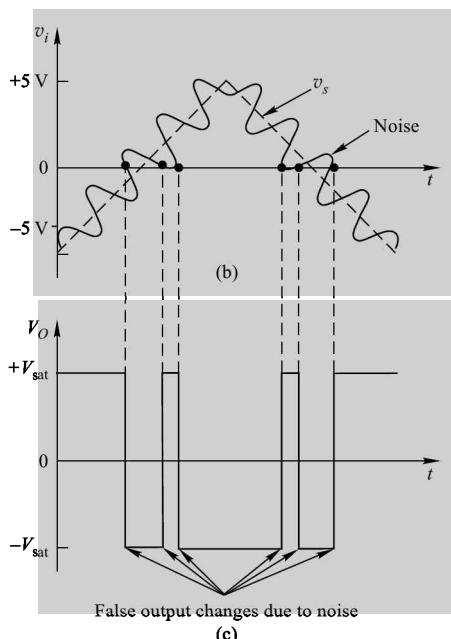
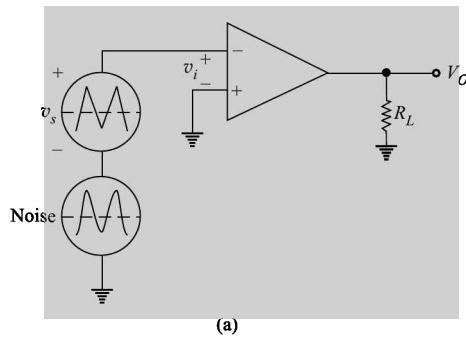
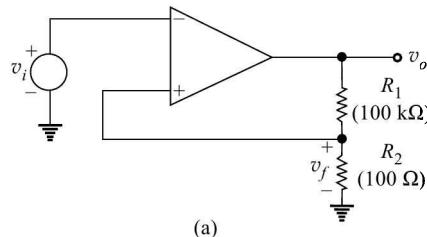


Fig. 9.9 (a) Circuit for Ex. 9.2b (b) Input Waveform (c) Output Waveform

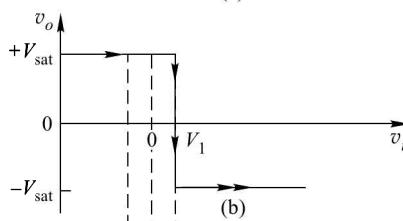
### 9.3.2 Regenerative Comparator (Schmitt Trigger)

Since it may not be possible to completely eliminate the noise voltage, therefore, we must prevent the circuit from sensing these false changes at the input. This is achieved by using positive feedback in the circuit and the resulting circuit is referred to as a *regenerative comparator* and is also known as *Schmitt trigger* (after the inventor of a vacuum-tube version of this circuit).

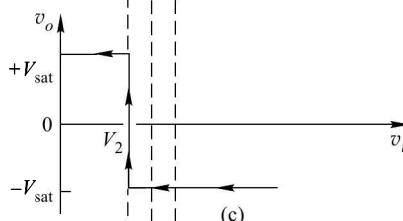
Figure 9.10a shows a Schmitt trigger circuit. Positive feedback is applied by taking a fraction of the output voltage,  $v_f = R_2 v_o / (R_1 + R_2)$  and applying it to the non-inverting input.



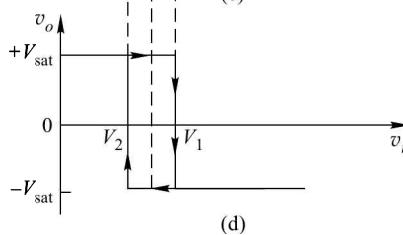
(a)



(b)



(c)



(d)

Fig. 9.10 (a) An Inverting Schmitt Trigger. The Input–Output Characteristics for (b) Increasing  $v_i$  and (c) Decreasing  $v_i$ , (d) The Complete Input–Output Characteristic

Let  $v_o = +V_{\text{sat}}$  and  $v_i < v_f$ . If  $v_i$  is now increased, then  $v_o$  remains constant at  $+V_{\text{sat}}$ , and  $v_f = R_2 / (R_1 + R_2) \cdot V_{\text{sat}}$  is constant until

$$v_i = V_1 = \frac{R_2}{R_1 + R_2} \cdot V_{\text{sat}} \quad (9.3)$$

At this voltage  $V_1$ , known as *upper-threshold voltage* or *upper-triggering voltage* ( $V_{UT}$ ), the output switches to  $-V_{sat}$  and remains at this value as long as  $v_i > V_1$ . This input–output characteristic is shown in Fig. 9.10b.

The voltage at the non-inverting input terminal for  $v_i > V_1$  is

$$v_i = -\frac{R_2}{R_1 + R_2} V_{sat}$$

If we now decrease  $v_i$ , then the output remains at  $-V_{sat}$  until  $v_i$  equals  $V_2$ , where

$$V_2 = -\frac{R_2}{R_1 + R_2} V_{sat} \quad (9.4)$$

and is known as *lower-threshold voltage* or *lower-triggering voltage* ( $V_{LT}$ ). At this input voltage, the output switches to  $+V_{sat}$  and remains at this value as long as  $v_i < V_2$ . This input–output characteristic is shown in Fig. 9.10c. The complete input–output characteristic is shown in Fig. 9.10d, where the portions without arrows may be traversed in either direction, but the segments with arrows can only be obtained if  $v_i$  varies as indicated by the arrows. This characteristic exhibits the hysteresis action in this circuit. The difference in voltage of  $V_{UT}$  and  $V_{LT}$  is called the *hysteresis voltage*,  $V_H$ .

### Example 9.3

- (a) In the Schmitt trigger circuit of Fig. 9.10a, if  $V_{sat} = 13$  V, find  $V_{UT}$  and  $V_{LT}$   
 (b) If  $v_i = 5 \sin wt$ , find the waveform of the output voltage.

#### Solution

$$(a) V_{UT} = \frac{0.1}{100.1} \times 13 \approx 13 \text{ mV}$$

$$V_{LT} = \frac{0.1}{100.1} (-13) \approx -13 \text{ mV}$$

- (b) The input and output voltage waveforms are shown in Fig. 9.11.

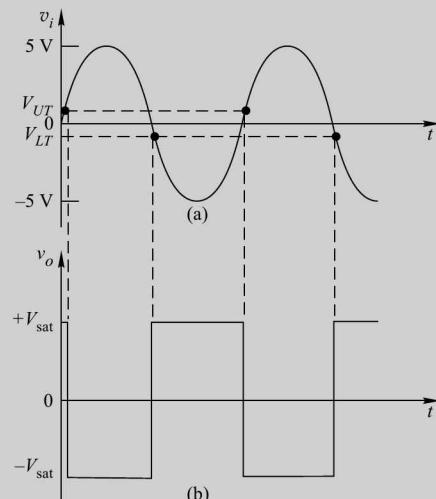


Fig. 9.11 Voltage Waveforms for Ex. 9.3b (a) Input (b) Output

## Limiting Output Voltage

For an operational amplifier, the output voltage levels are limited to  $+V_{sat}$  and  $-V_{sat}$  and are dependent on supply voltages. These output voltages may not be compatible with voltage levels required by a particular load. For example, TTL requires input voltages that are approximately +5 V and 0 V.

To limit the output voltages, which are independent of the power supply voltages, two back-to-back Zener diodes are used at the output as shown in Fig. 9.12. The resistance  $R$  is used to limit the current in the Zener diodes.

For this circuit, the upper and lower threshold voltages are given by (Problem 9.3)

$$V_{UT} = \frac{R_2}{R_1 + R_2} (V_{Z1} + V_D) + \frac{R_1}{R_1 + R_2} V_R \quad (9.5)$$

and

$$V_{LT} = -\frac{R_2}{R_1 + R_2} (V_{Z2} + V_D) + \frac{R_1}{R_1 + R_2} V_R \quad (9.6)$$

where  $V_D$  = voltage across a forward-biased diode ( $\sim 0.7$  V) and  $V_{Z1}$  and  $V_{Z2}$  are the Zener voltages.

The output voltages will be  $(V_{Z1} + V_D)$  and  $-(V_{Z2} + V_D)$ . If the input signal  $v_i$  is applied in place of the reference voltage  $V_R$ , and  $V_R$  is applied at the inverting input terminal, then a non-inverting comparator is obtained.

### 9.3.3 Astable (or Free-Running) Multivibrator

An astable or free-running multivibrator is a square-wave generator. The circuit of Fig. 9.13 is a comparator circuit discussed above in which an  $RC$  low-pass circuit (combination of  $R_f$  and  $C$ ) is used to integrate the output voltage  $v_o$  and the voltage across the capacitor,  $C$ , is applied to the inverting input terminal in place of the external signal.

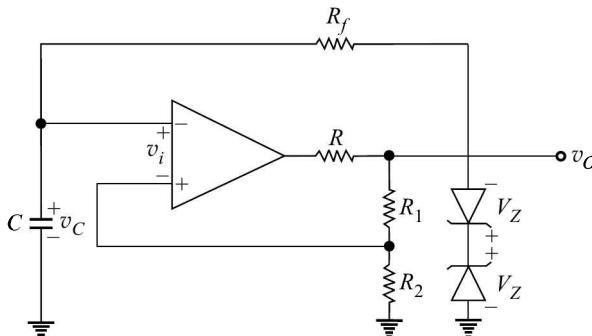


Fig. 9.13 Free-Running Multivibrator

The output voltage  $v_o = V_z + V_D = V_o$  if  $v_i < 0$  and  $v_o = -(V_z + V_D) = -V_o$  if  $v_i > 0$ . Consider an instant of time when  $v_i < 0$ , therefore  $v_o = V_o$ . The voltage at the non-inverting input is  $R_2 V_o / (R_1 + R_2) = \beta V_o$  where  $\beta = R_2 / (R_1 + R_2)$ . The capacitor  $C$  charges exponentially towards  $V_o$  with the time constant  $R_f \cdot C$ . The output voltage remains constant at  $V_o$  until  $v_c = \beta V_o (= V_{UT})$  at which time the comparator output reverses to  $-V_o$ . Now  $v_c$  changes exponentially towards  $-V_o$  with the same time constant and again the output makes a transition from  $-V_o$  to  $+V_o$  when  $v_c = -\beta V_o (= V_{LT})$ . The waveforms of the capacitor voltage,  $v_c$ , and the output voltage,  $v_o$ , are illustrated in Fig. 9.14.

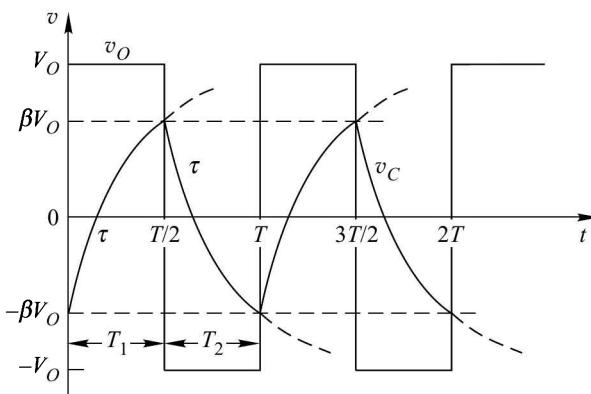


Fig. 9.14     *Waveforms of Output and Capacitor Voltage for Free-Running Multivibrator*

The time period,  $T$ , of the output square waveform is determined using the charging and discharging of capacitor. The voltage across the capacitor,  $v_c$ , when it is charging from  $-\beta V_o$  to  $+V_o$  is given by

$$v_c = V_o [1 - (1 + \beta) e^{-t/\tau}] \quad (9.7)$$

where,

$$\tau = R_f \cdot C$$

At

$$t = T/2, v_c \left( \frac{T}{2} \right) = +\beta V_o$$

Therefore,

$$T = 2\tau \ln \left( \frac{1 + \beta}{1 - \beta} \right) = 2R_f \cdot C \ln \left( 1 + \frac{2R_2}{R_1} \right) \quad (9.8)$$

The frequency,  $f = 1/T$ , of the square wave is independent of  $V_o$ . The circuit of Fig. 9.13 has two quasi-stable states. The output remains in one of these states for a time  $T_1$  and then makes an abrupt transition to the second state and remains in that state for a time  $T_2$ . The cycle of period  $T (= T_1 + T_2)$  repeats itself and hence this circuit is known as *free-running* or *astable* multivibrator.

This square-wave generator is useful in the frequency range of about 10 Hz to 10 kHz. At higher frequencies, the slew rate of the OP AMP limits the slope of the output square wave. The symmetry of the output waveform depends on the matching of the two Zener diodes (Problem 9.5). The unsymmetrical square wave ( $T_1 \neq T_2$ ) can be obtained by using different time constants for charging the capacitor to  $+V_o$  and  $-V_o$  (Problem 9.6).

## Duty Cycle

For an unsymmetrical square waveform, the percentage of time-period ( $T$ ) for which the output is HIGH is referred to as *duty cycle* and is given by

$$\text{Percent duty cycle} = \frac{T_{\text{ON}}}{T_{\text{ON}} + T_{\text{OFF}}} \times 100 \quad (9.9)$$

where  $T_{\text{ON}}$  and  $T_{\text{OFF}}$  are indicated in Fig. 9.15.

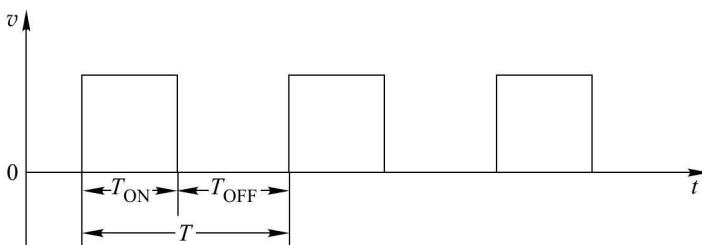


Fig. 9.15      *Definition of  $T_{\text{ON}}$  and  $T_{\text{OFF}}$  of a Repetitive Square Wave*

### 9.3.4 Monostable Multivibrator

A monostable multivibrator has one stable state and one quasi-stable state. The circuit remains in its stable state until an external triggering pulse causes a transition to the quasi-stable state. The circuit returns to its stable state after a time  $T$ . Hence it generates a single output pulse in response to an input pulse and is referred to as a *one-shot* or *single-shot*.

The astable multivibrator of Fig. 9.13 is modified to operate as a monostable multivibrator by connecting a diode ( $D_1$ ) across  $C$ , so that  $v_C$  is clamped at  $V_D$  during positive excursion.

Under steady-state condition, this circuit will be in its stable state with the output  $v_o$  at  $+V_o$ . The capacitor  $C$  is clamped at the voltage  $V_D$  (ON voltage of diode  $\approx 0.7$  V). The voltage  $V_D$  must be less than  $\beta V_o$  for  $v_i < 0$ . It can be verified that this circuit cannot remain in the other state ( $v_o = -V_o$ ) under steady-state condition (Problem 9.7). This circuit can be switched to the other state by applying a negative trigger pulse with amplitude greater than  $\beta V_o - V_D$  at the non-inverting input terminal. The triggering pulse is applied through a high-pass  $RC$  circuit ( $C_i, R_i$ ) and diode  $D_2$ . The complete circuit of the monostable multivibrator is shown in Fig. 9.16.

When a trigger pulse is applied  $v_i$  goes positive causing a transition in the state of the circuit ( $v_o = -V_o$ ). The capacitor  $C$  now charges exponentially with a time constant  $\tau = R_f \cdot C$  towards  $-V_o$  (diode  $D_1$  being reverse-biased). When  $v_C$  becomes more negative than  $-\beta V_o$ ,  $v_i$  becomes negative and consequently the output swings back to  $+V_o$  (steady-state output). From the above discussion, it is clear that the circuit has one stable state and the other state is quasi-stable. The width of the trigger pulse ( $T_p$ ) must be much smaller than the duration  $T$  of the output pulse generated. The diode  $D_2$  is used to avoid malfunctioning of the circuit due to any positive noise spikes present in the triggering line. The waveforms of  $v_C$  and  $v_o$  are shown in Fig. 9.17.

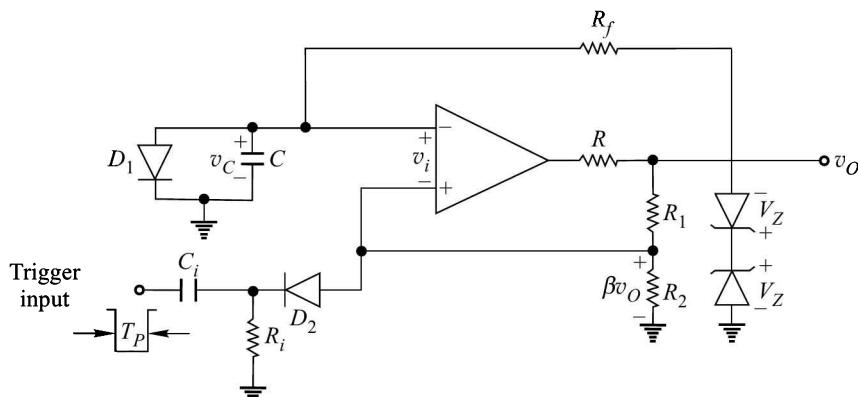
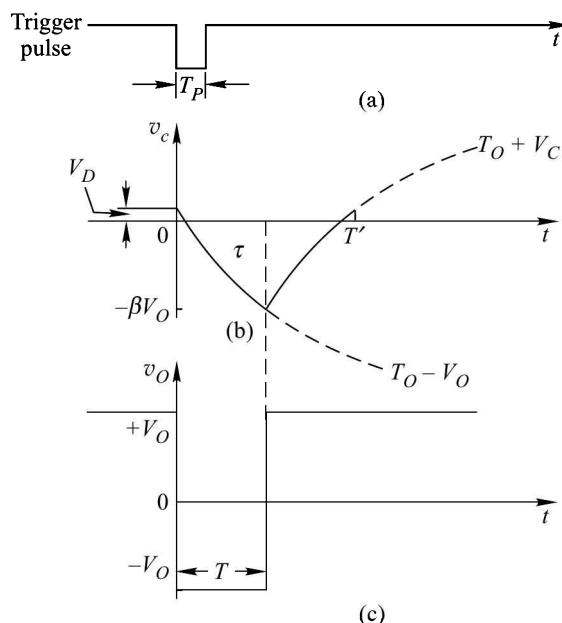


Fig. 9.16 Monostable Multivibrator

Fig. 9.17 (a) Negative Trigger Pulse (b) Waveforms of  $v_c$  (c)  $v_o$ 

During the quasi-stable state, the capacitor voltage  $v_c$  is given by

$$v_c = -V_o + (V_o + V_d)e^{-t/\tau} \quad (9.10)$$

At

Therefore,

$$t = T, v_c = -\beta V_o$$

$$T = \tau \ln \left( \frac{1 + V_d/V_o}{1 - \beta} \right) \quad (9.11)$$

Usually  $V_D \ll V_o$  and if  $R_1 = R_2$ , so that  $\beta = \frac{1}{2}$ , then

$$T \approx 0.69R_f \cdot C \quad (9.12)$$

This circuit comes back to its steady-state condition at  $T'$  and the time interval  $T' - T$  is referred to as *recovery time*. The circuit should not be triggered again before  $T'$  for reliable operation.

## 9.4 SCHMITT TRIGGER ICs

Some TTL and CMOS Schmitt trigger input inverters and gates are available in 74 series ICs. These are given in Table 9.1. The outputs of these devices are fast changing, similar to the outputs of other TTL and CMOS circuits, but they can respond to slowly varying inputs.

Table 9.1 Available Schmitt Trigger ICs in TTL and CMOS Logic Families

IC No.	Description	Logic symbol
7413	Dual 4-input NAND Schmitt triggers	
7414	Hex Schmitt trigger inverters	
74132	Quad 2-input NAND Schmitt triggers	

### 9.4.1 Schmitt Trigger Square-Wave Generator

A very simple square-wave generator can be made using a Schmitt trigger inverter as shown in Fig. 9.18. If its output is HIGH, the capacitor  $C$  charges with the time constant  $\tau = RC$ . When the capacitor voltage ( $v_C$ ) reaches  $V_{UT}$ , the output voltage ( $v_o$ ) goes LOW. Now the capacitor discharges through the output transistor of the gate which is in saturation. When  $v_C$  reaches  $V_{LT}$  the output voltage goes HIGH. This process goes on and a square waveform shown in Fig. 9.19 is obtained at  $v_o$ . The time period of the square wave is given by

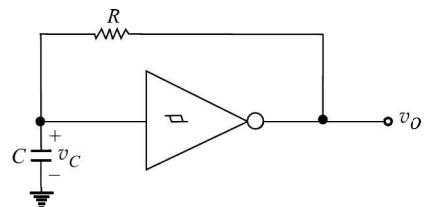


Fig. 9.18

A Schmitt Trigger Square Wave Generator

$$T = T_1 + T_2 = RC \left( \ln \frac{V(1) - V_{LT}}{V(1) - V_{UT}} + \ln \frac{V_{UT}}{V_{LT}} \right) \quad (9.13)$$

where  $V(1)$  is the logic 1 output voltage.

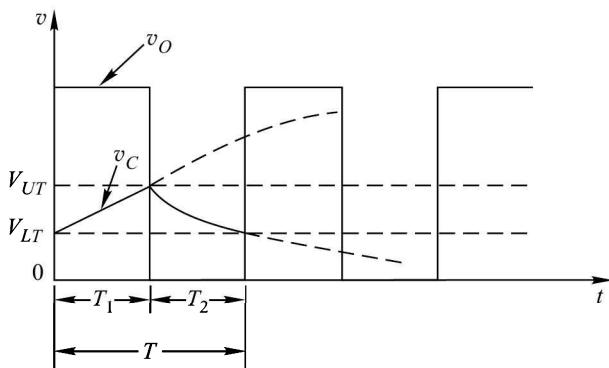


Fig. 9.19      *Voltage Waveforms of Square Wave Generator*

## 9.5 MONOSTABLE MULTIVIBRATOR ICs

Some very useful monostable multivibrators are available in IC form. These are given in Table 9.2.

Table 9.2      *Available Monostable Multivibrator ICs in TTL and CMOS Logic Families*

IC No.	Description
74121	Monostable multivibrator (one-shot)
74122	Retriggerable monostable multivibrator with clear
74123	Dual retriggerable monostable multivibrator with clear
74221	Dual monostable multivibrator with clear

A brief description of these ICs is given below.

### 9.5.1 74121 Monostable Multivibrator

The functional diagram and the function table of the most popular and commonly used one-shot TTL IC 74121 are given in Figs 9.20 and 9.21 respectively.

In order to trigger the one-shot, there must be a rising pulse edge at point Z. This is possible in one of the following two ways:

1. One or both of the A inputs are at logic 0 and the B input makes a transition from logic 0 to 1 ( $\uparrow$ ).
2. The B input is at logic 1 and either one of the A inputs makes a transition from logic 1 to 0 ( $\downarrow$ ) while the other A input remains at logic 1 or both A inputs go from logic 1 to 0 ( $\downarrow$ ) simultaneously.

The duration of the output pulse is dependent upon the values of the resistor ( $R_{\text{EXT}}$  or  $R_{\text{INT}}$ ) and capacitor ( $C_{\text{EXT}}$ ) used. A timing capacitor ( $C_{\text{EXT}}$ ) is to be externally connected between the terminals marked  $R_{\text{EXT}}$ / $C_{\text{EXT}}$ . In case of electrolytic capacitor, + terminal is to be connected to  $R_{\text{EXT}}/C_{\text{EXT}}$ . The maximum

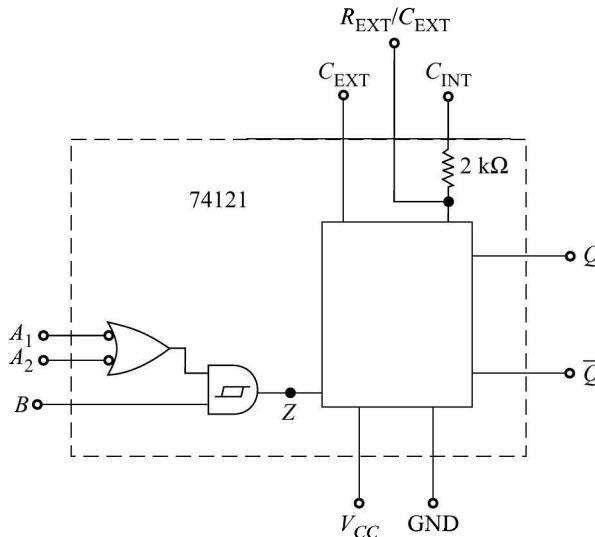


Fig. 9.20 Functional Block Diagram of Monostable Multivibrator IC 74121

Inputs			Outputs	
$A_1$	$A_2$	$B$	$Q$	$\bar{Q}$
0	$\times$	1	0	1
$\times$	0	1	0	1
$\times$	$\times$	0	0	1
1	1	$\times$	0	1
1	$\downarrow$	1	↑	↑
$\downarrow$	1	1	↑	↑
$\downarrow$	$\downarrow$	1	↑	↑
0	$\times$	$\uparrow$	↑	↑
$\times$	0	$\uparrow$	↑	↑

Fig. 9.21 Function Tables of 74121

allowable value of the external capacitor is  $1,000 \mu\text{F}$ . If the external timing capacitor is not used, the stray capacitance between the pins of the IC will result in a very low pulse width output.

For timing resistor, there are two options: (i) an internal timing resistor ( $R_{INT}$ ) of  $2 \text{ k}\Omega$  becomes effective if the  $R_{INT}$  terminal is connected to  $V_{CC}$ , (ii) external timing resistor ( $R_{EXT}$ ) is to be connected between  $R_{EXT}/C_{EXT}$  terminal and  $V_{CC}$ . The range of  $R_{EXT}$  is from  $1.4 \text{ k}\Omega$  to  $40 \text{ k}\Omega$ .

In any case, both  $R_{\text{EXT}}$  and  $R_{\text{INT}}$  must not be used simultaneously. The duration of the output pulse is given by

$$T_{\text{ON}} \approx 0.7RC \quad (9.14)$$

where  $R$  and  $C$  are the values of the timing resistor and capacitor used respectively.

The minimum possible pulse width is 30 to 35 ns (with  $R_{\text{INT}}$  and without  $C_{\text{EXT}}$ ) and the maximum possible pulse width is about 28s (with  $R_{\text{EXT}} = 40 \text{ k}\Omega$  and  $C_{\text{EXT}} = 1,000 \mu\text{F}$ ).

The maximum allowable duty cycle is 67% with  $R_{\text{INT}}$  and goes up to 90% with  $R_{\text{EXT}}$  of 40 kΩ.

The  $A$  inputs are applied to a normal TTL gate and therefore should make fast transitions, but the  $B$  input is a Schmitt trigger input that responds to very slowly changing inputs. Therefore, if slow waveform is required to trigger a one-shot it should be applied to the  $B$  input.

### 9.5.2 Retriggerable Monostable Multivibrators (74122 and 74123)

IC 74121 is a non-retriggerable one-shot, i.e. it responds to a trigger pulse only when it is in the quiescent (stable) state. A retriggerable one-shot responds to a trigger pulse even when it is in the quasi-stable (ON) state. If a trigger pulse occurs when it is in quasi-stable state, it resets the timing and does not go to stable state (OFF) until one pulse duration after the last trigger pulse. Its duty cycle is, therefore, unlimited. The functional block diagram and the function table of 74122 retriggerable monostable multivibrator are shown in Figs 9.22 and 9.23 respectively. It has an internal timing resistor ( $R_{\text{INT}}$ ) of 10 kΩ and it is to be used in the same way as it is used in 74121. The limits of external resistor ( $R_{\text{EXT}}$ ) is from 5 kΩ to 50 kΩ. The pulse width is to be determined in the following way:

1. If the timing capacitor,  $C_{\text{EXT}} < 1000 \text{ pF}$ , the pulse width is to be determined from the graph of Fig. 9.26.
2. If the timing capacitor,  $C_{\text{EXT}} \geq 1000 \text{ pF}$ , then

$$T_{\text{ON}} \approx 0.3R \cdot C_{\text{EXT}} (1 + 0.7/R) \quad (9.15)$$

where  $T_{\text{ON}}$  is in nanoseconds,  $R$  in kilo ohms and  $C_{\text{EXT}}$  in picofarads.

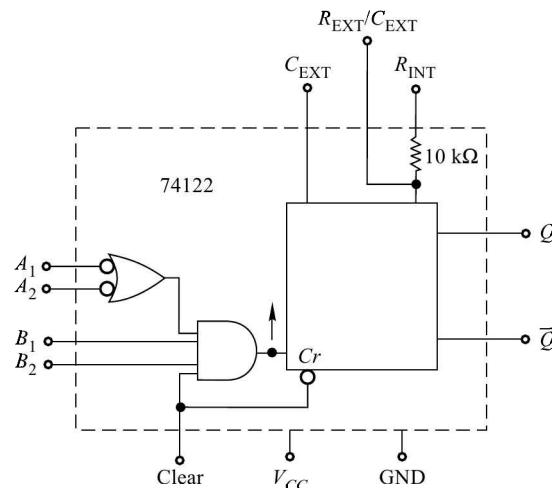


Fig. 9.22 Functional Block Diagram of 74122

$Cr$	Inputs				Outputs	
	$A_1$	$A_2$	$B_1$	$B_2$	$Q$	$\bar{Q}$
0	x	x	x	x	0	1
x	1	1	x	x	0	1
x	x	x	0	x	0	1
x	x	x	x	0	0	1
x	0	x	1	1	0	1
1	0	x	↑	1	[waveform]	[waveform]
1	0	x	1	↑	[waveform]	[waveform]
1	x	0	1	1	0	1
1	x	0	↑	1	[waveform]	[waveform]
1	x	0	1	↑	[waveform]	[waveform]
1	1	↓	1	1	[waveform]	[waveform]
1	↓	↓	1	1	[waveform]	[waveform]
1	↓	1	1	1	[waveform]	[waveform]
↑	0	x	1	1	[waveform]	[waveform]
↑	x	0	1	1	[waveform]	[waveform]

Fig. 9.23 Function Table of 74122

74123 is a dual retriggerable one-shot IC. Its operation is similar to 74122 except that it does not have an internal timing resistor. Its block diagram is shown in Fig. 9.24 and the function table of each section is shown in Fig. 9.25.

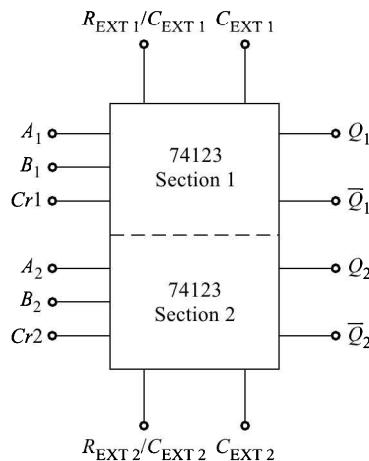


Fig. 9.24 Block Diagram of 74123

These ICs have asynchronous clear inputs which are active-low and are used to clear ( $Q = 0$ ) the one-shot.

Inputs			Outputs	
$Cr$	$A$	$B$	$Q$	$\bar{Q}$
0	$\times$	$\times$	0	1
$\times$	1	$\times$	0	1
$\times$	$\times$	0	0	1
1	0	$\uparrow$		
1	$\downarrow$	1		
$\uparrow$	0	1		

Fig. 9.25 **Function Table of 74123**

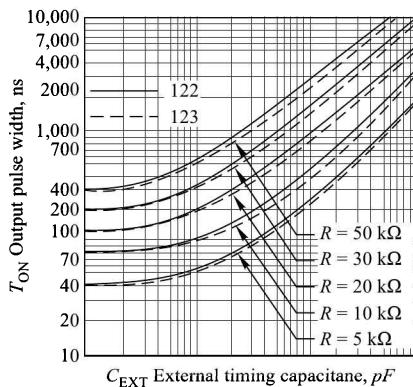


Fig. 9.26 **Graph for Determination of Pulse Width of Retriggerable One-Shots**

### 9.5.3 Non-retriggerable Monostable Multivibrator with Clear (74221)

IC 74221 is a dual monostable multivibrator with performance characteristics identical to those of 74121 and its block diagram is same as that of IC 74123 (Fig. 9.24). Its function table is given in Fig. 9.27. There is an active-low asynchronous clear input for resetting the circuit.

Inputs			Outputs	
$Cr$	$A$	$B$	$Q$	$\bar{Q}$
0	$\times$	$\times$	0	1
$\times$	1	$\times$	0	1
$\times$	$\times$	0	0	1
1	0	$\uparrow$		
1	$\downarrow$	1		

Fig. 9.27 **Function Table of 74221**

**Example 9.4**

(a) A 74121 is used to generate 8 ms pulse with duty cycle of 80%. Its inputs are  $A_1 = A_2 = 0$  and the trigger pulses shown in Fig. 9.28a are applied at the  $B$  input. Sketch the waveform of the  $Q$  output, (b) If the same trigger pulses are applied at the  $B$  input of one section of a 74123 (used to generate 8 ms pulse) with  $A = 0$  and  $Cr = 1$ , what will be the  $Q$  output?

**Solution**

- (a) The output waveform is shown in Fig. 9.28b.  
 (b) The output waveform is shown in Fig. 9.28c.

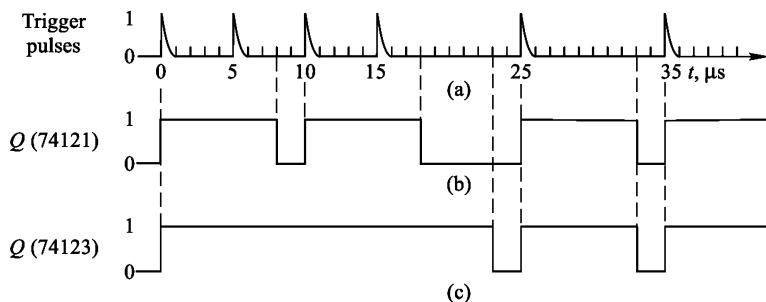


Fig. 9.28

(a) Trigger Pulses (b)  $Q$  Output of Non-Retriggerable One-Shot 74121 (c)  $Q$  Output of Retriggerable One-Shot 74123

**Example 9.5**

- (a) In the circuit of Ex. 9.4b, if clear input goes LOW every 12  $\mu$ s (Figs. 9.29a and b), obtain the output waveform.  
 (b) In the above circuit, if 74221 IC is used instead of 74123, obtain the output waveform.

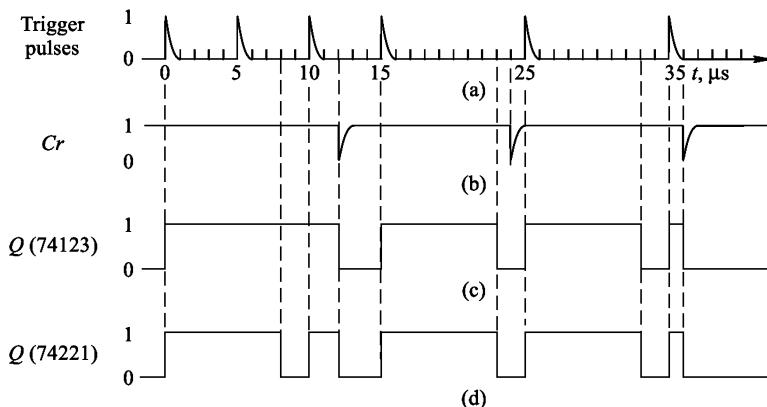


Fig. 9.29      Waveforms for Ex 9.5

**Solution**

- (a) The output waveform is shown in Fig. 9.29c.  
 (b) The output waveform is shown in Fig. 9.29d.

**Example 9.6**

Design a circuit to monitor a.c. mains voltage. If it ever misses a cycle, a buzzer should sound continuously until reset manually by depressing a push-button.

**Solution**

First the sinusoidal voltage is to be converted into a square wave which can be used as input for digital circuits. The circuit for this is shown in Fig. 9.30. The a.c. mains voltage is reduced to about 20 V peak-to-peak. The input to the gate is clamped to  $V_{CC}$  during positive cycle and 0 V during negative cycle of sinusoidal waveform with the help of the diodes  $D_1$  and  $D_2$  respectively.

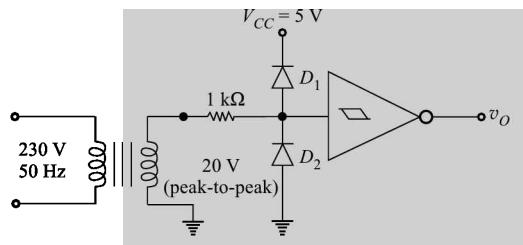


Fig. 9.30 **Circuit for Converting a.c. Mains Voltage to Square Waveform**

Since the output of the transformer is a slowly varying voltage, a Schmitt trigger circuit is used. The output of this circuit is a square wave of frequency 50 Hz.

Now, a retriggerable monostable multivibrator can be used to produce 100% duty cycle output as long as a.c. voltage is present. The circuit for this is shown in Fig. 9.31. The operation of this circuit is explained below.

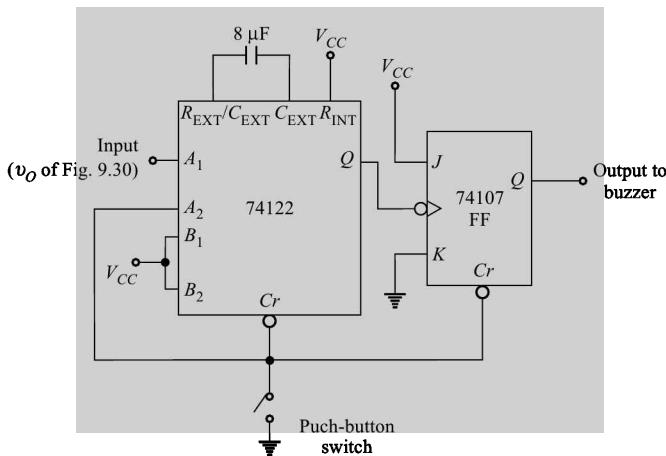


Fig. 9.31 **Circuit for Detecting Missing Pulse**

The output of the circuit of Fig. 9.30 produces a 50 Hz square wave, i.e. it produces pulse edges every 20 ms. These pulses are applied at the  $A_1$  input of 74122 set for a pulse width of little more than 20 ms, say 25 ms, so that the one-shot is never reset as long as a.c. power is present.

With internal timing resistor of  $10\text{ k}\Omega$ ,  $C_{\text{EXT}}$  required is  $\sim 8\ \mu\text{F}$  (Eq. 9.15). When a.c. power goes OFF, the output of 74122 goes LOW which sets the FLIP-FLOP. This drives the buzzer. If the voltage and current requirements of the buzzer are high, then buffer/driver will be required.

When the push-button is depressed, it clears both the FLIP-FLOPs and the 74122. When the push-button is released, it triggers the 74122 and the circuit assumes normal operation if the a.c. power is resumed, otherwise it will sound the alarm again after 25 ms.

### 9.5.4 Astable Multivibrator Using One-Shots

Two one-shots can be coupled together as shown in Fig. 9.32 to make an astable multivibrator. Initially,  $Q_2$  is LOW and when the switch SW is opened, the first multivibrator (IC1) goes to the quasi-stable state. When the output  $Q_1$  goes LOW ( $\downarrow$ ) at a time  $T_1$  (pulse width of IC1), IC2 is triggered and output  $Q_2$  goes HIGH and remains HIGH for a time  $T_2$  (pulse width of IC2) and then goes LOW ( $\downarrow$ ) which triggers IC1. This process is continuous and a square wave is produced at  $Q_1$  and  $Q_2$ . The frequency of the square wave can be controlled by the timing elements.

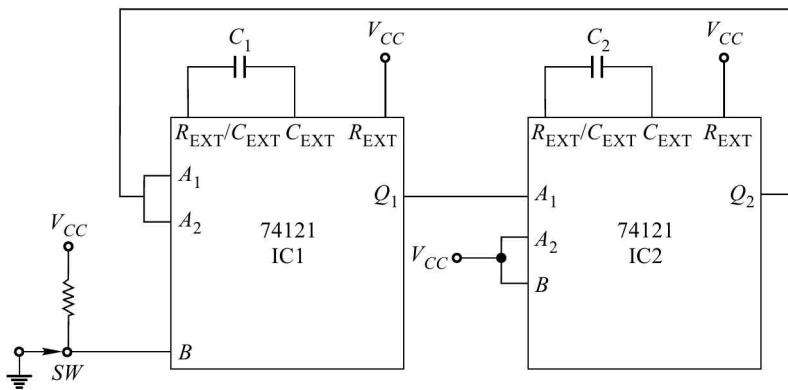


Fig. 9.32 An Astable Multivibrator Using two 74121 ICs

## 9.6 555 TIMER

IC 555 timer (available in 8 pin DIP or TO-99 package) is one of the most popular and versatile sequential logic devices which can be used in monostable and astable modes. Its inputs and output are directly compatible with both TTL and CMOS logic circuits. The functional block diagram of 555 timer is shown in Fig. 9.33.

It has two comparators which receive their reference voltages by a set of three resistances connected between the supply voltage  $V_{\text{CC}}$  and ground. The reference voltage for comparator 1 is  $2V_{\text{CC}}/3$ , and for comparator 2 is  $V_{\text{CC}}/3$ . These reference voltages have control over the timing which can be varied electronically, if required,

by applying a voltage to the control-voltage input terminal. If this is not required, then a bypass capacitor ( $0.01 \mu F$ ) should be connected between this terminal and ground to bypass noise and/or ripple voltages from the power supply.

On a negative-going excursion of the trigger input, when the trigger input passes through the reference voltage  $V_{CC}/3$ , the output of the comparator 2 goes HIGH and sets the FLIP-FLOP ( $Q = 1$ ). On a positive-going excursion of the threshold input, the output of the comparator 1 goes HIGH when the threshold voltage passes through the reference voltage  $2V_{CC}/3$ . This resets the FLIP-FLOP ( $\bar{Q} = 1$ ).

The FLIP-FLOP is cleared (irrespective of the  $S$  input) when the reset input is less than about 0.4 V. When this input is not required to be used, it is normally returned to  $V_{CC}$ .

An external timing capacitor,  $C$  is to be connected between the discharge terminal and ground. When the FLIP-FLOP is in the reset state, its  $\bar{Q} = 1$ , which drives  $T_1$  to saturation thereby discharging the timing capacitor. The timing cycle starts when the FLIP-FLOP goes to set state and therefore  $T_1$  is OFF. The timing capacitor charges with the time constant  $\tau = R_A \cdot C$ , where  $C$  is the timing capacitor and  $R_A$  is an external resistor to be connected between the discharge terminal and  $V_{CC}$ .

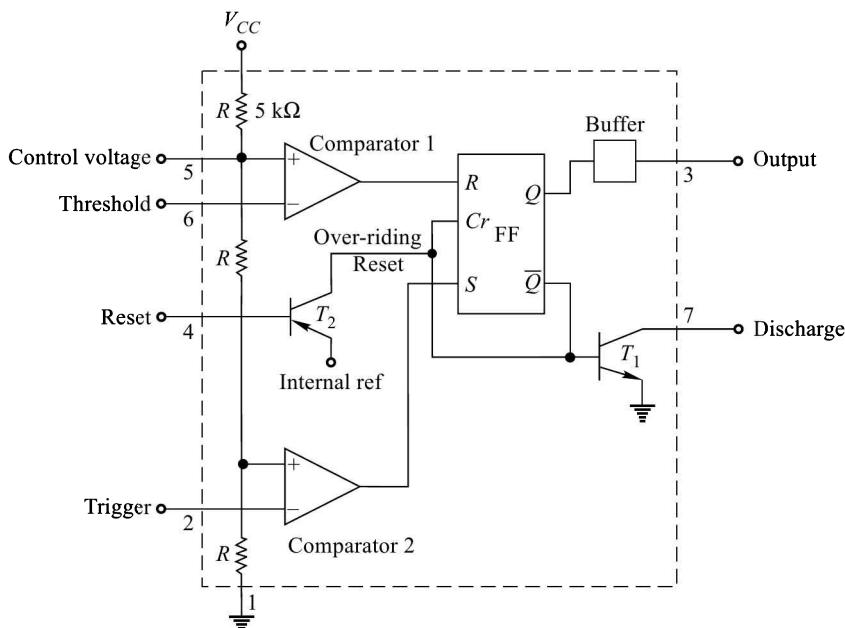


Fig. 9.33 Functional Block Diagram of 555 Timer

The output is at logic 1 whenever the transistor  $T_1$  is OFF and at logic 0 when  $T_1$  is ON. The load can be connected either between the output terminal and  $V_{CC}$  or between the output and ground terminals. The four possible output connections are shown in Fig. 9.34. The maximum sink or source current is 200 mA. The voltage corresponding to HIGH output is approximately 0.5 V below  $V_{CC}$  and for LOW it is approximately 0.1 V. IC 556 timer contains two 555 timer circuits and is available in 14 pin DIP.

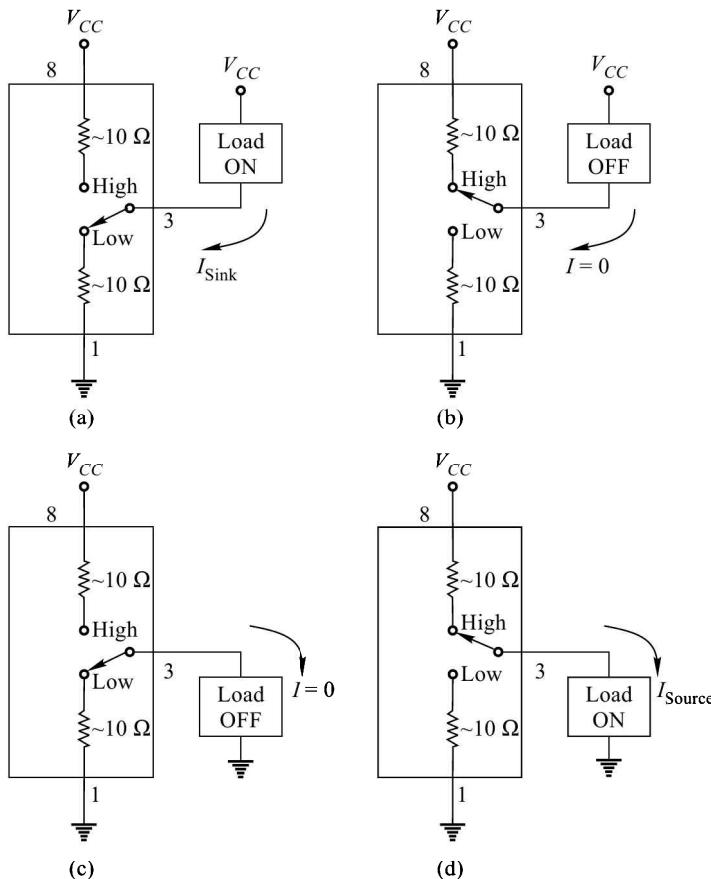


Fig. 9.34 555 Timer Output Terminal Operation

### 9.6.1 Monostable Multivibrator

A monostable multivibrator circuit using a 555 Timer is shown in Fig. 9.35. If the trigger input is held HIGH, then under steady-state condition, the transistor  $T_1$  is ON, the discharge and output terminals are at LOW level. It can be verified that  $T_1$  cannot be OFF under steady-state (Problem 9.15). When a negative pulse applied at the trigger input crosses the voltage  $V_{CC}/3$ , the output of the comparator 2 goes HIGH which sets the FLIP-FLOP and consequently  $T_1$  turns OFF and the output goes HIGH. The capacitor  $C$  starts getting charged to  $V_{CC}$  with the time constant,  $\tau = R_A \cdot C$ . The circuit remains in this condition even after the trigger has returned to logic 1. When the increasing capacitor voltage reaches  $2/3 V_{CC}$ , the output of the comparator 1 goes HIGH which resets the FLIP-FLOP. The transistor  $T_1$  goes to saturation, thereby discharging the capacitor and the output goes LOW. The various waveforms are shown in Fig. 9.36. This is a non-retriggerable monostable multivibrator. This circuit can be converted to a retriggerable type if reset is connected to trigger input instead of to  $V_{CC}$  (Problem 9.16) and it is triggered at the positive-edge of the trigger pulse.

The output timing interval,  $T$  is given by  $1.1 R_A \cdot C$ .

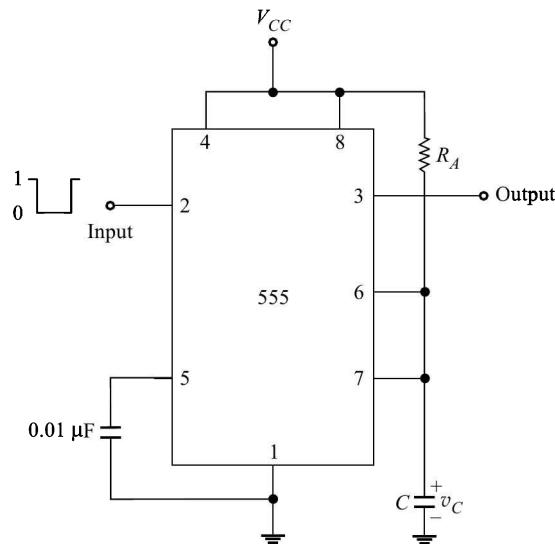


Fig. 9.35 A Monostable Multivibrator Using 555

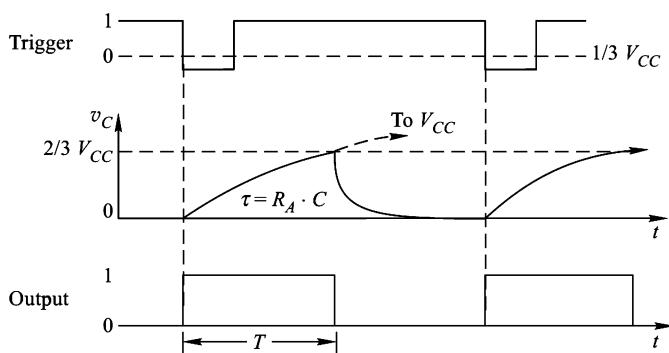


Fig. 9.36 Waveforms of Monostable Multivibrator

## 9.6.2 Astable Multivibrator

An astable multivibrator using 555 timer is shown in Fig. 9.37. Let us assume that the output is in HIGH state and the capacitor  $C$  is charging through resistors  $R_A$  and  $R_B$  [time constant  $\tau_1 = (R_A + R_B) C$ ]. When the voltage across  $C(v_C)$  reaches  $2/3 V_{CC}$ , the output goes LOW and  $C$  starts discharging through  $R_B$  with a time constant  $\tau_2 \approx R_B \cdot C$ . When  $v_C$  drops below  $V_{CC}/3$ , the timer is triggered and the output again goes HIGH. The capacitor  $C$  now again starts charging towards  $V_{CC}$  with the time constant  $\tau_1$ . The various waveforms are shown in Fig. 9.38. The charging and discharging time intervals are given by

$$T_1 \approx 0.7 C(R_A + R_B) \quad (9.16)$$

and

$$T_2 \approx 0.7CR_B \quad (9.17)$$

Therefore,

$$f = \frac{1}{T} = \frac{1}{T_1 + T_2} = \frac{1.4}{C(R_A + 2R_B)} \quad (9.18)$$

and the duty cycle

$$D = \frac{R_A + R_B}{R_A + 2R_B} \times 100\% \quad (9.19)$$

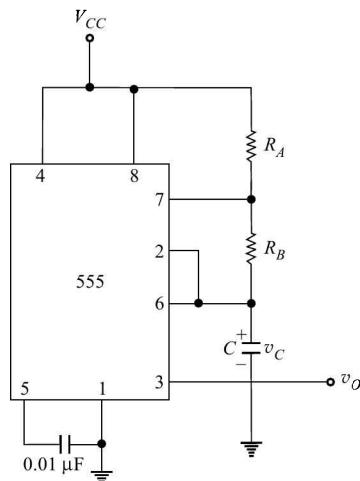


Fig. 9.37 An Astable Multivibrator Using 555

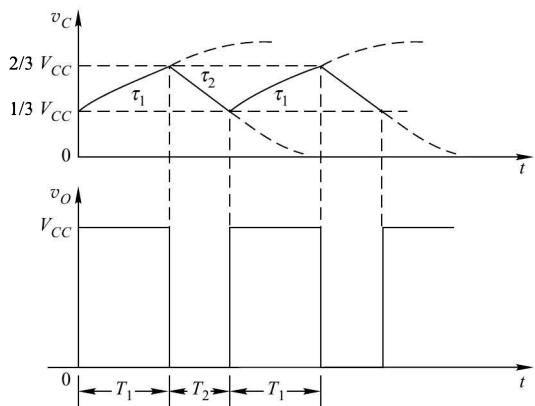


Fig. 9.38 Waveforms of 555 Astable Multivibrator

From Eq. (9.19) we note that the duty cycle is always different from 50%. It can be made 50% (symmetrical square wave) by connecting a diode across  $R_B$ , which will clamp the voltage across  $R_B$  when the capacitor is charging and therefore if  $R_A$  and  $R_B$  are equal,  $\tau_1$  and  $\tau_2$  will be same. It is also possible to generate a symmetrical square wave by using the output of 555 as clock input of a T-type FLIP-FLOP with  $T = 1$ . The output of the FLIP-FLOP will be a symmetrical square wave.

## SUMMARY

Some of the very useful circuits for generation of timing waveforms have been discussed in this chapter. These circuits can be designed using discrete devices but because of the availability of very useful ICs like gates, OP AMPS, monostable multivibrators, and 555 timer, circuits with discrete devices are no longer used. A variety of useful circuits can be designed using these ICs which are very convenient to use. A thorough understanding of the operations of these ICs is very essential for using them effectively in new applications.

## GLOSSARY

**Analog comparator** An analog circuit which compares two analog signals and produces an output whose polarity indicates the result of comparison.

**Astable multivibrator** A multivibrator circuit having both the states as quasi-stable states. It is a square wave oscillator. Also known as free-running multivibrator.

**Bistable multivibrator** A multivibrator circuit that has both the states as stable. It goes from one stable state to another when triggered. It is same as the FLIP-FLOP.

**Free-running multivibrator** Same as the astable multivibrator.

**Hysteresis** The difference between the upper and lower triggering voltages in a Schmitt trigger circuit.

**Jitter** When the successive pulse durations of a monostable multivibrator are not same it is said to jitter.

**Monostable multivibrator** An electronic circuit that produces an output pulse for a fixed time period in response to a trigger pulse and then returns to its quiscent state. The width of the output pulse produced depends upon the timing elements (R and C) used in the circuit. It is also known as one-shot circuit.

**One-shot** Same as the monostable multivibrator.

**Operational amplifier (OP AMP)** A high gain differential input amplifier.

**Pulse stretcher** Same as the monostable multivibrator.

**Quasi-stable state** The state which is not a stable state. When a circuit is forced to go to a quasi-stable state, it comes back to its stable state after a time depending upon the elements of the circuit.

**Recovery time** The time required for a timing circuit to come back to steady-state after having been triggered.

**Regenerative comparator** Same as the Schmitt trigger.

**Retriggerable monostable multivibrator** A monostable multivibrator that is capable of getting triggered even when it is in its non-quiscent (or quasi-stable) state.

**Schmitt trigger** An analog comparator circuit with different upper and lower triggering voltages. It exhibits hysteresis effect. It produces a single, sharp transition from a slowly changing input. It is same as regenerative comparator.

## REVIEW QUESTIONS

- 9.1 A free-running multivibrator has \_\_\_\_\_ quasi-stable states.
- 9.2 A \_\_\_\_\_ monostable multivibrator can be triggered even when it is not in stable state.
- 9.3 A pulse-stretcher circuit is same as a \_\_\_\_\_ multivibrator.
- 9.4 The symbol used for an inverting Schmitt trigger circuit is \_\_\_\_\_.
- 9.5  represents a \_\_\_\_\_ Schmitt trigger circuit.
- 9.6 A sinusoidal waveform can be converted into square waveform by using a \_\_\_\_\_.
- 9.7 The upper triggering voltage and the lower triggering voltage are not same for a \_\_\_\_\_ circuit.
- 9.8 A multivibrator circuit having one stable state and other quasi-stable state is known as a \_\_\_\_\_ multivibrator.
- 9.9 An astable multivibrator does not require \_\_\_\_\_ input.
- 9.10 \_\_\_\_\_ circuit is used for converting slowly varying signals suitable as inputs to logic circuits.
- 9.11 A multivibrator with two stable states is known as a \_\_\_\_\_ multivibrator.
- 9.12 False triggering in a comparator circuit due to the presence of noise can be eliminated by the use of \_\_\_\_\_ comparator.

- 9.13 The output timing interval  $T$  of a monostable multivibrator using 555 timer is \_\_\_\_\_.
- 9.14 The frequency of input sinusoidal waveform to a comparator circuit is 1 kHz. The frequency of the output square wave will be \_\_\_\_\_.
- 9.15 555 timer circuits can be used for making monostable and \_\_\_\_\_ multivibrators.

## PROBLEMS

- 9.1** Verify that the circuit of Fig. 9.39 acts as an astable multivibrator.

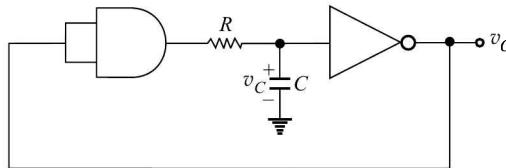


Fig. 9.39 Circuit for Problem 9.1

- 9.2** Verify that  $v_o$  in logic 0 state is not a stable state for the circuit of Fig. 9.2.
- 9.3** Verify Eqs (9.5) and (9.6).
- 9.4** In the circuit of Fig. 9.12, the two Zener diodes are identical with  $V_{z1} = V_{z2} = V_z = 4.6$  V, voltage across conducting diode ( $V_D$ ) = 0.6 V,  $V_R = 1$  V. Find  $V_{UT}$  and  $V_{LT}$ .  
Obtain the output voltage waveform if  $v_i$  is a sinusoidal voltage with 5 V peak value.
- 9.5** In the free-running multivibrator circuit of Fig. 9.13, the breakdown voltages of the Zener diodes are not same. Obtain the expression for the frequency of the square wave.
- 9.6** If the resistance  $R_f$  in Fig. 9.13 is replaced by the circuit of Fig. 9.40, obtain the output voltage waveform and its time period.

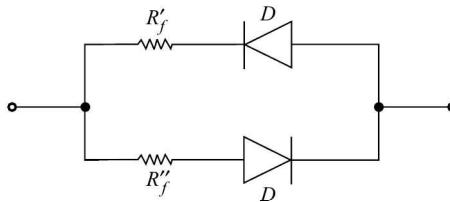


Fig. 9.40

- 9.7** Verify that the circuit of Fig. 9.16 cannot remain in the state  $v_o = -V_o$  under steady-state condition.
- 9.8** Verify Eq. (9.13).
- 9.9** A traffic light is required to be red for 30s and green for 60s with automatic switching. Design a circuit for this purpose.
- 9.10** Design a one-shot to generate pulses of 0.2  $\mu$ s, using  
(a) 74121,  
(b) 74122.

**9.11** Design a one-shot to generate a 5 ms pulse, using

- (a) The internal resistor,
- (b) A 40 k $\Omega$  external resistor.

What is the maximum frequency of trigger pulses in each case. Use 74121.

**9.12** Design a 100 kHz, 60% duty cycle square-wave generator using 555.

**9.13** (a) An astable multivibrator using 555 timer is shown in Fig. 9.41. Explain its operation and sketch the relevant waveforms.

- (b) Find the expression for the time period of the output waveform.
- (c) Is it possible to get square waveform with 50% duty cycle? If yes, find out the condition under which it is possible.
- (d) If  $R_B = 20 \text{ k}\Omega$ , find  $R_A$  for a 50% duty cycle.
- (e) Find, if any, the restriction on the maximum value of  $R_B$  (in terms of  $R_A$ ).

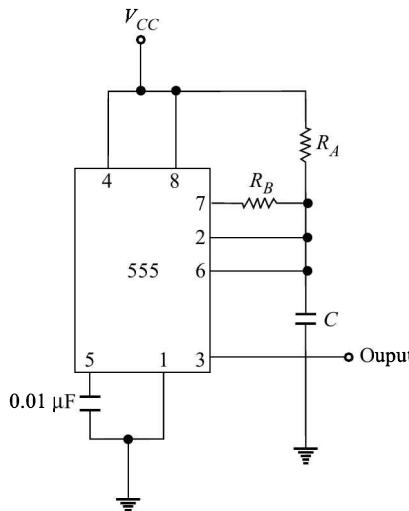


Fig. 9.41 *Circuit for Problem 9.13*

**9.14** (a) Explain how a 555 timer monostable multivibrator can be used as a frequency divider.

- (b) Design a circuit using 555 timer to divide the frequency of a train of pulses by 3.

**9.15** Verify that the circuit of Fig. 9.35 cannot have HIGH output in stable state.

**9.16** Design a retriggerable monostable multivibrator using 555 timer and explain its operation.