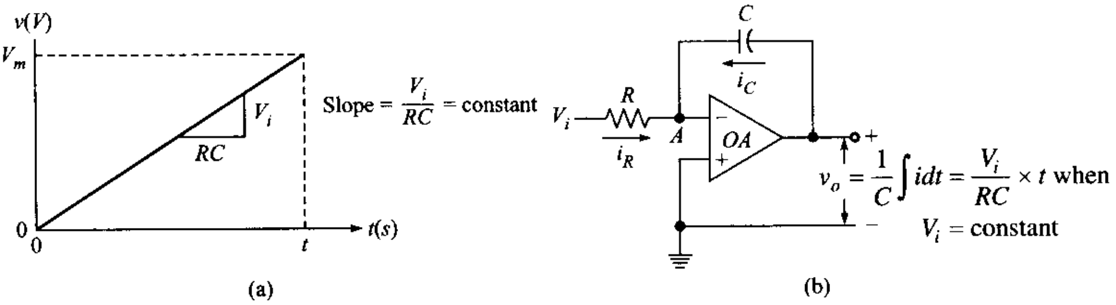

$v(V)$
$V_m$
$V_i$
$RC$
0
0
$t$
$t(s)$
Slope = $\frac{V_i}{RC}$ = constant
(a)
$C$
$i_C$
$R$
$V_i$
$A$
$i_R$
OA
−
+
+
−
$v_o = \frac{1}{C}\int idt = \frac{V_i}{RC} \times t$ when
$V_i$ = constant
(b)

Fig. 12.33 An integrating circuit

Shortly thereafter, a reset pulse is generated by the CONTROL box that resets the RAMP and clears the decade counters to 0s, and another conversion cycle begins. In the meantime, the contents of the previous conversion are contained in the latches and are displayed on the seven-segment LEDs.

As a specific example, suppose that the clock in Fig. 12.32 is set at 1.0 MHz and the ramp voltage slope is 1.0 V/ms. Note that the decade counters have the ability to store and display any decimal number from 000 up to 999. From the beginning of a conversion cycle, it will require 999 clock pulses (999 $\mu$s) for the counters to advance full scale. During this same time period, the ramp voltage will have increased from 0.0 V up to 999 mV. So, this circuit as it stands will display the value of any input voltage between 0.0 V and 999 mV.

In effect, we have a digital voltmeter! For instance, if $V_X$ = 345 mV, it will require 345 clock pulses for the counter to advance from 000 to 345, and during the same time period the ramp will have increased to 345 mV. So, at the end of the conversion cycle, the display output will read 345—we supply the units of millivolts.

One weakness of the single-slope A/D converter is its dependency on an extremely accurate ramp voltage. This in turn is strongly dependent on the values of $R$ and $C$ and variations of these values with time and temperature. The dual-slope A/D converter overcomes these problems.

## Dual-Slope A/D Converter

The logic diagram for a basic dual-slope A/D converter is given in Fig. 12.32. With the exception of the ramp generator and the comparator, the circuit is similar to the single-slope A/D converter in Fig. 12.32. In this case, the integrator forms the desired ramp—in fact, two different ramps—as the input is switched first to the unknown input voltage $V_X$ and then to a known reference voltage $V_r$. Here's how it works.

We begin with the assumptions that the clock is running, and that the input voltage $V_X$ is positive. A conversion cycle begins with the decade counters cleared to all 0s, the ramp reset to 0.0 V, and the input switched to the unknown input voltage $V_X$. Since $V_X$ is positive, the integrator output $V_c$ will be a negative ramp. The comparator output $V_g$ is thus positive and the clock is allowed to pass through the CLOCK GATE to the counters. We allow the ramp to proceed for a fixed time period $t_1$, determined by the count detector for time $t_1$. The actual voltage $V_c$ at the end of the fixed time period $t_1$ will depend on the unknown input $V_X$, since we know that $V_c = -(V_x/RC) \times t_1$ for an integrator.

When the counter reaches the fixed count at time $t_1$, the CONTROL unit generates a pulse to clear the decade counters to all 0s and switch the integrator input to the negative reference voltage $V_r$. The integrator will now begin to generate a ramp beginning at $-V_c$ and increasing steadily upward until it reaches 0.0 V. All this time, the counter is counting, and the conversion cycle ends when $V_c$ = 0.0 V since the CLOCK GATE

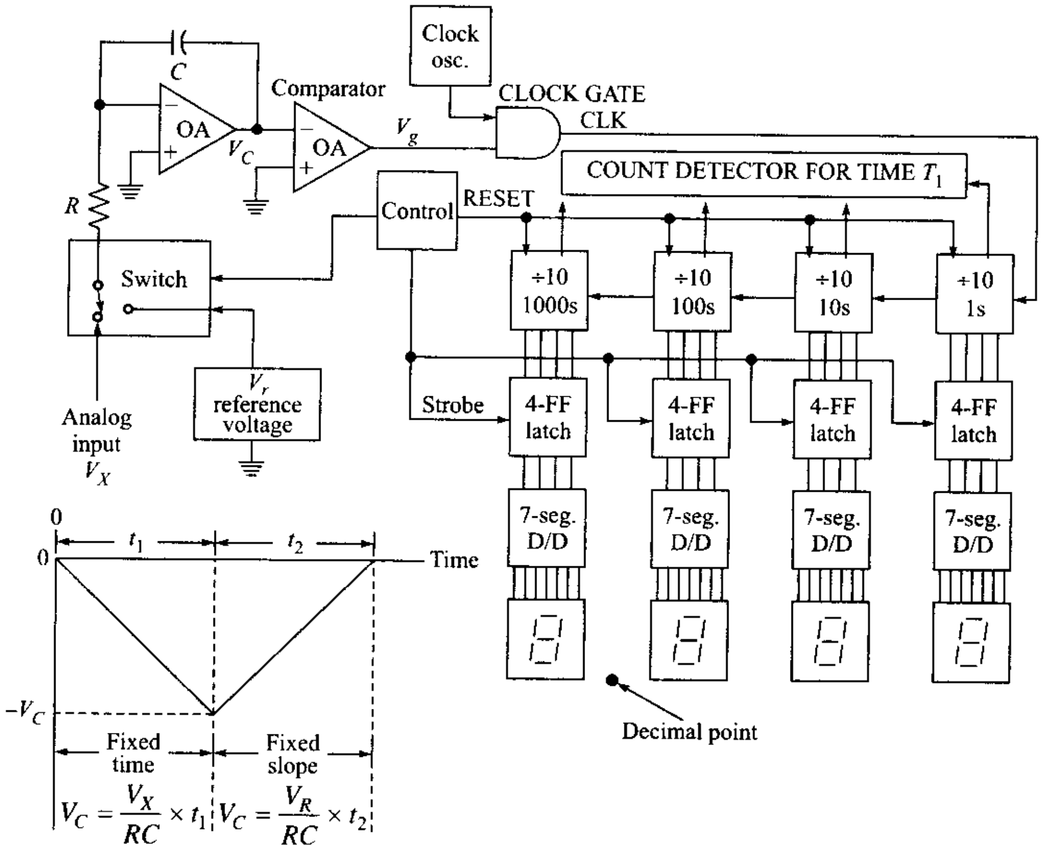


Fig. 12.34 **Dual-slope A/D converter**

is now disabled. The equation for the positive ramp is $V_c = (V_r/RC) \times t_2$. In this case, the slope of this ramp $(V_r/RC)$ is constant, but the time period $t_2$ is variable.

In fact, since the integrator output voltage begins at 0.0 V, integrates down to $-V_c$, and then integrates back up to 0.0 V, we can equate the two equations given for $V_c$. That is:

$$\frac{V_X}{RC} \times t_1 = \frac{V_r}{RC} \times t_2$$

The value $RC$ will cancel from both sides, leaving

$$V_X = V_r \times \frac{t_2}{t_1}$$

Since $V_r$ is a known reference voltage and $t_1$ is a predetermined time, clearly the unknown input voltage is directly proportional to the variable time period $t_2$. However, this time period is exactly the contents of the decade counters at the end of a conversion cycle! The obvious advantage here is that the $RC$ terms cancel from both sides of the equation above—in other words, this technique is free from the absolute values of either $R$ or $C$ and also from variations in either value.

As a concrete example, let's suppose that the clock in Fig. 12.34 is 1.0 MHz, the reference voltage is – 1.0 Vdc, the fixed time period $t_1$ is 1000 $\mu$s, and the $RC$ time constant of the integrator is set at $RC$ = 1.0 ms. During the time period $t_1$, the integrator voltage $V_c$, will ramp down to –1.0 Vdc if $V_X$ = 1.0 V. Then, during time $t_2$, $V_c$ will ramp all the way back up to 0.0 V, and this will require a time of 1000 $\mu$s, since the slope of this ramp is fixed at 1.0 V/ms. The output display will now read 1000, and with placement of a decimal as shown, this reads 1.000 V.

Another way of expressing the operation of this A/D converter is to solve the equation $V_X = V_r(t_2/t_1)$ for $t_2$, since $t_2$ is the digital readout. Thus $t_2 = (V_X/V_r)t_1$. If the same values as given above are applied, an unknown input voltage $V_X$ = 2.75 V will be digitized and the readout will be $t_2$ = (2.75/1.0)1000 = 2750, or 2.75 V, using the decimal point on the display. Notice that we have used $t_1$ = 1000, the number of clock pulses that occur during the time period $t_1$. Likewise, $t_2$ is the number of clock pulses that occur during the time period $t_2$.

SELF-TEST

16. Is a single-ramp A/D converter slower or faster than a successive-approximation A/D converter?
17. What is the greatest weakness of a single-ramp A/D converter?
18. What advantage does the dual-slope A/D converter offer over the single-ramp A/D converter?

## 12.10 A/D ACCURACY AND RESOLUTION

Since the A/D converter is a closed-loop system involving both analog and digital systems, the overall accuracy must include errors from both the analog and digital positions. In determining the overall accuracy it is easiest to separate the two sources of error.

If we assume that all components are operating properly, the source of the digital error is simply determined by the resolution of the system. In digitizing an analog voltage, we are trying to represent a continuous analog voltage by an equivalent set of digital numbers. When the digital levels are converted back into analog form by the ladder, the output is the familiar staircase waveform. This waveform is a representation of the input voltage but is certainly not a continuous signal. It is, in fact, a discontinuous signal composed of a number of discrete steps. In trying to reproduce the analog input signal, the best we can do is to get on the step which most nearly equals the input voltage in amplitude.

The simple fact that the ladder voltage has steps in it leads to the digital error in the system. The smallest digital step, or quantum, is due to the LSB and can be made smaller only by increasing the number of bits in the counter. This inherent error is often called the *quantization error* and is commonly ±1 bit. If the comparator is centered, as with the continuous converter, the quantization error can be made ±1/2 LSB.

The main source of analog error in the A/D converter is probably the comparator. Other sources of error are the resistors in the ladder, the reference-voltage supply ripple, and noise. These can, however, usually be made secondary to the sources of error in the comparator.

The sources of error in the comparator are centered around variations in the dc switching point. The dc switching point is the difference between the input voltage levels that cause the output to change state. Variations in switching are due primarily to offset, gain, and linearity of the amplifier used in the comparator. These parameters usually vary slightly with input voltage levels and quite often with temperature. It is these changes which give rise to the analog error in the system.

An important measure of converter performance is given by the differential linearity. *Differential linearity* is a measure of the variation in voltage-step size that causes the converter to change from one state to the next. It is usually expressed as a percent of the average step size. This performance characteristic is also a function of the conversion method and is best for the converters having counters that count continuously. The counter-type and continuous-type converters usually have better differential linearity than do the successive-approximation-type converters. This is true since the ladder voltage is always approaching the analog voltage from the same direction in the one case. In the other case, the ladder voltage is first on one side of the analog voltage and then on the other. The comparator is then being used in both directions, and the net analog error from the comparator is thus greater.

The next logical question that might be asked is: what should be the relative order of magnitudes of the analog and digital errors? As mentioned previously, it would be difficult to justify construction of a 15-bit converter that has an overall error of 1 percent. In general, it is considered good practice to construct converters having analog and digital errors of approximately the same magnitudes. There are many arguments for and against this, and any final argument would have to depend on the situation. As an example, an 8-bit converter would have a quantization error of $\frac{1}{256} \cong 0.4$ percent. It would then seem reasonable to construct this converter to an accuracy of 0.5 percent in an effort to achieve an overall accuracy of 1.0 percent. This might mean constructing the ladder to an accuracy of 0.1 percent, the comparator to an accuracy of 0.2 percent, and so on, since these errors are all accumulative.

**Example 12.13** What overall accuracy could one reasonably expect from the construction of a 10-bit A/D converter?

*Solution* A 10-bit converter has a quantization error of $\frac{1}{1024} \cong 0.1$ percent. If the analog portion can be constructed to an accuracy of 0.1 percent, it would seem reasonable to strive for an overall accuracy of 0.2 percent.

## SUMMARY

Digital-to-analog conversion, the process of converting digital input levels into an equivalent analog output voltage, is most easily accomplished by the use of resistance networks. The binary ladder has been found to have definite advantages over the resistance divider. The complete D/A converter consists of a binary ladder (usually) and a flip-flop register to hold the digital input information.

The simultaneous method for A/D conversion is very fast but becomes cumbersome for more than a few bits of resolution. The counter-type A/D converter is somewhat slower but represents a much more reasonable solution for digitizing high-resolution signals. The continuous-converter method, the successive-approximation method, and the section-counter method are all variations of the basic counter-type A/D converter which lead to a much faster conversion time. A dual-slope A/D converter is somewhat slower than the previously discussed methods but offers excellent accuracy in a relatively inexpensive circuit. Dual-slope A/D converters are widely used in digital voltmeters.

The D/A converter and A/D converter logic circuits given in this chapter are all drawn in logic block diagram form and can all be constructed by simply connecting these commercially available logic blocks. For instance, a D/A converter can be constructed by connecting resistors that have values of $R$ and $2R$, or an A/D converter can be constructed by connecting the various inverters, gates, flip-flops, and so on; however, you must realize that these units are now readily available as MSI circuits. The only really practical and economical way to build D/A converters or A/D converters is to make use of these commercially available circuits; this is exactly the subject pursued in the next chapter.

## GLOSSARY

- ***A/D conversion*** The process of converting an analog input voltage to a number of equivalent digital output levels.
- ***A/D converter flash type*** Effects fast and simultaneous conversion of analog data to digital through number of comparators.
- ***A/D converter tracking type*** Effects tracking of analog input through its continuous comparison with converter's digital output.
- ***binary equivalent weight*** The value assigned to each bit in a digital number, expressed as a fraction of the total. The values are assigned in binary fashion according to the sequence 1, 2, 4, 8, … , $2^n$, where $n$ is the total number of bits.
- ***D/A conversion*** The process of converting a number of digital input signals to one equivalent analog output voltage.
- ***differential linearity*** A measure of the variation in size of the input voltage to an A/D converter which causes the converter to change from one state to the next.
- ***Millman's theorem*** A theorem from network analysis which states that the voltage at any node in a resistive network is equal to the sum of the currents entering the node divided by the sum of the conductances connected to the node, all determined by assuming that the voltage at the node is zero.
- ***monotonicity*** A consistent increase in output in response to a consistent increase in input (voltage or current).
- ***quantization error*** The error inherent in any digital system due to the size of the LSB.
- ***sample and hold circuit*** Samples analog voltage signal and holds briefly to facilitate analog to digital conversion.
- ***SAR*** Sequential approximation register, used in a sequential A/D converter.

## PROBLEMS

### Section 12.1

12.1 What is the binary equivalent weight of each bit in a 6-bit resistive divider?

12.2 Draw the schematic for a 6-bit resistive divider.

12.3 Verify the voltage output levels for the network in Fig. 12.4, using Millman's theorem. Draw the equivalent circuits.

12.4 Assume that the divider in Prob. 12.2 has +10 V full-scale output, and find the following:

a. The change in output voltage due to a change in the LSB

b. The output voltage for an input of 110110

12.5 A 10-bit resistive divider is constructed such that the current through the LSB resistor is 100 $\mu$A. Determine the maximum current that will flow through the MSB resistor.

### Sections 12.2, 12.3 and 12.4

12.6 What is the full-scale output voltage of a 6-bit binary ladder if 0 = 0 V and 1 = +10 V? Of an 8-bit ladder?

12.7 Find the output voltage of a 6-bit binary ladder with the following inputs:

a. 101001 b. 111011

c. 110001

12.8 Check the results of Prob. 11-7 by adding the individual bit contributions.

12.9 What is the resolution of a 12-bit D/A converter which uses a binary ladder? If the full-scale output is +10 V, what is the resolution in volts?

12.10 How many bits are required in a binary ladder to achieve a resolution of 1 mV if full scale is +5 V?

## Section 12.5

12.11 How many comparators are required to build a 5-bit simultaneous A/D converter?

12.12 Redesign the encoding matrix and READ gates in Fig. 12.20, using NAND gates.

12.13 Assuming that the input reference voltage is $V$ = 10.0 Vdc, determine the digital output of the A/D converter in Fig. 12.21a for an input voltage of:

a. 1.25 V b. 3.33 V
c. 8.05 V

## Section 12.6

12.14 Find the following for a 12-bit counter-type A/D converter using a 1-MHz clock:

a. Maximum conversion time
b. Average conversion time
c. Maximum conversion rate

12.15 What clock frequency must be used with a 10-bit counter-type A/D converter if it must be capable of making at least 7000 conversions per second?

12.16 Design additional control circuitry for Fig. 12.24 such that the A/D converter in Fig. 12.23 will continue to make conversions after an initial START pulse is applied.

## Sections 12.7 and 12.8

12.17 What is the conversion time of a 12-bit successive-approximation-type A/D converter using a 1-MHz clock?

12.18 What is the conversion time of a 12-bit section-counter-type A/D converter using a 1-MHz clock? The counter is divided into three equal sections.

## Section 12.9

12.19 For the integrator in Fig. 12.31, show that the output voltage is given by $V_o = \{V_i/RC)t$, assuming that the input voltage $V_i$ is a constant. [*Hint*: Using Kirchhoff's current law at node $A$, the resistor current $i_R$ is equal to the capacitor current $i_C$, but $i_R = V_i/R$ and $i_C = q/t = (V_oC)/t$.]

12.20 Design the control logic for the CONTROL box in Fig. 12.32 to generate the proper control signals shown in that figure.

12.21 Calculate a value for $C$ in Fig. 12.33 to obtain a fixed slope $V_i/(RC)$ = 1000 V/s, given $V_i$ = 1.0 Vdc and $R$ = 100 kΩ.

12.22 Can you design an amplifier such that the output is always positive and is equal to the magnitude of the input voltage? In other words, the input can be either $+V_i$ or $-V_i$ but in either case, the output will be $+V_i$.

12.23 Design the CONTROL logic for the converter in Fig. 12.34.

## Section 12.10

12.24 What overall accuracy could you reasonably expect from a 12-bit A/D converter?

12.25 Discuss the overall acceptable accuracy of a 10-bit A/D converter in terms of quantization error, ladder accuracy, comparator accuracy, converter accuracy, and other factors.

## Answers to Self-tests

1. 1/63
2. 30/15 = 2 V
3. 0.15625 V
4. 9.84375 V
5. A monotonicity test checks to see that the D/A output voltage increases regularly as the input digital signals increase.
6. +5 Vdc
7. Resolution = 10/256 = 39.06 mV
8. Its conversion time is very fast.
9. Possibilities include radar signal processing, video displays, high-speed instrumentation, and television broadcasting.
10. 128 $\mu$s

11. 64 $\mu$s
12. The continuous type A/D converter uses an up-down counter.
13. The continuous type A/D converter is faster than the counter-type A/D converter.
14. SAR stands for successive-approximation register.
15. The ADC0804 is an 8-bit CMOS successive-approximation A/D converter.
16. Slower
17. One major weakness of the single-slope A/D converter is that it is extremely sensitive to variations in ramp voltage and hence to errors in ramp voltage.
18. The *RC* time constant cancels out, making the conversion much less sensitive to variations in ramp voltage accuracy.

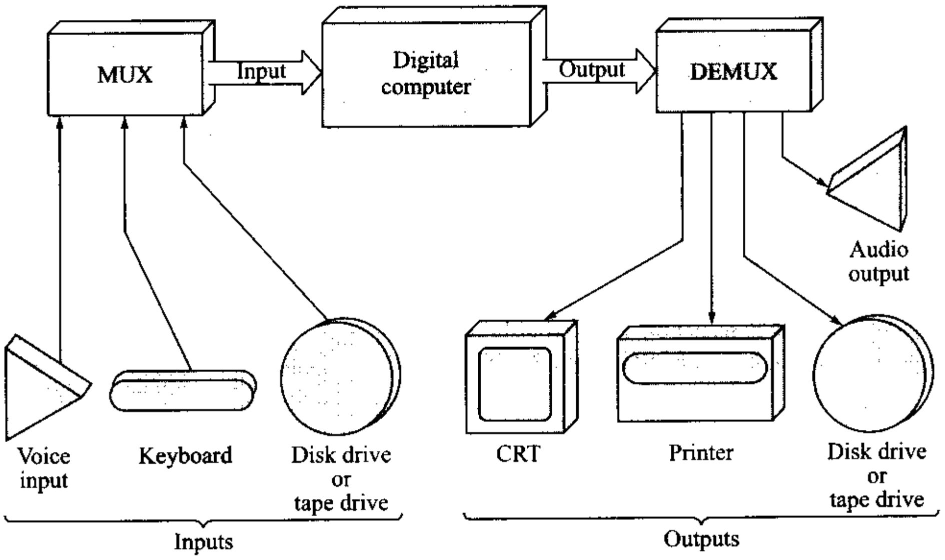
# Memory

# 13

## OBJECTIVES

- List the various forms of magnetic and optical memory and explain how each works
- Discuss memory addressing techniques
- Describe ROM, PROM, and EPROM and their characteristics and differences
- Compare the advantages and disadvantages of SRAM and DRAM and be familiar with the basic features of SRAM and DRAM chips
- Describe how content addressable memory works

The ability to store information (to remember) is an important requirement in a digital system. Circuits and/or systems designed specifically for data storage are referred to as memory. In the simplest application, the memory may be a flip-flop, or perhaps a number of flip-flops connected to form a register. In a larger system, such as a microcomputer, the memory may be composed of semiconductor memory chips. Semiconductor memories are composed of bipolar transistors or MOS transistors on an integrated circuit (IC), and are available in two general categories—read-only memory (ROM) and random-access memory (RAM). ROM and RAM memories can be constructed to store impressive amounts of data entirely within a computer system. Both programmed instructions and data are stored in a computer by means of ROM and RAM. But really large amounts of data (such as banking or insurance records) are generally stored using magnetic memory techniques. Magnetic memory includes the recording of digital information on magnetic tape, hard disks, and floppy disks. Magnetic storage systems are quite sophisticated and are usually externally accessed, as shown in Chapter 1 in Fig. 1.28 and repeated here for reference. However, in last two decades there has been tremendous growth in optical memory devices like compact disk, digital versatile disk etc. that gives low cost high capacity alternative storage solution.



## 13.1 BASIC TERMS AND IDEAS

### Semiconductor Memory

Recent advances in semiconductor technology have provided a number of reliable and economical MSI and LSI memory circuits. The typical semiconductor memory consists of a rectangular array of *memory cells*, fabricated on a silicon wafer, and housed in a convenient package, such as a DIP. The basic memory cell is typically a transistor flip-flop or a circuit capable of storing charge and is used to store 1 bit of information. Memories are usually classified as either bipolar, metal oxide semiconductor (MOS), or complementary metal oxide semiconductor (CMOS) according to the type of transistor used to construct the individual memory cells. The total number of cells in a memory determine its *capacity*. For instance, a 1024 bipolar memory chip is a semiconductor memory that has 1024 memory cells, each cell consisting of a flip-flop constructed with the use of bipolar transistors. *Chip* is a term commonly used to refer to a semiconductor memory device. In general, faster operation is obtained with a bipolar memory chip, but greater packing density and thus reduced size and cost, as well as lower power requirements, are characteristics of MOS and CMOS memory chips.

### Characteristics

The two general categories of memory, RAM and ROM, can be further divided as illustrated in Fig. 13.1. A dc power supply is required to energize any semiconductor memory chip. Once dc power is applied to a *static RAM (SRAM)*, the SRAM retains stored information indefinitely, without any further action. A *dynamic RAM (DRAM)*, on the other hand, does not retain stored data indefinitely; any stored data must be stored again (refreshed) periodically. Both SRAMs and DRAMs are used to construct the memory inside a microcomputer or minicomputer (see Fig. 1.34 in Chapter 1). DRAMs are used as the bulk of the memory, and high-speed SRAMs are used for a smaller, rapid-access type of memory known as *cache memory*. The cache is used to momentarily store selected data in order to improve computer speed of operation. SRAMs can be either bipolar or MOS, but all DRAMs are MOS.

The information (data) stored in a ROM is *fixed* and will be retained permanently even if dc power is removed. Clearly, a ROM is ideal for storing permanent instructions necessary for the startup and operation of a computer. These instructions are retained, even when the computer is off, and become immediately available each time the computer is turned on. Data stored in a *pogrammable ROM* (PROM) is permanent—a PROM can be programmed only once! However, the data stored in an *erasable PROM* (EPROM) can be "erased"; the EPROM can then be used to store new data. PROMs can be either bipolar or MOS, but all EPROMs are MOS.

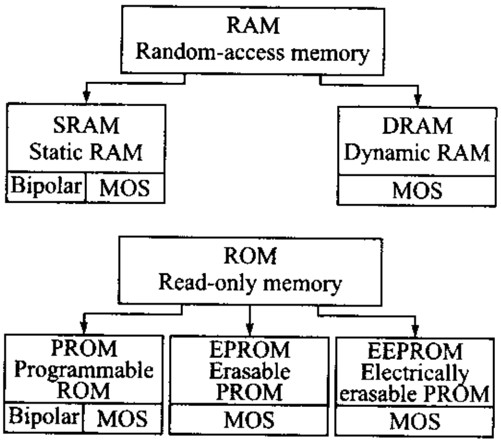


Fig. 13.1

## RAM

A block diagram of a typical RAM chip is shown in Fig. 13.2a. An application in which data changes frequently calls for the use of a RAM. The logic circuitry associated with a RAM will allow a single bit of information to be stored in any of the memory cells—this is the write operation. There is also logic circuitry that will detect whether a 0 or a 1 is stored in any particular cell—this is the read operation. The fact that a bit can be written (stored) in any cell or read (detected) from any cell suggests the description *random access*. A control signal, usually called *chip-select* or *chip-enable*, is used to enable or disable the chip. In the read mode, data from the selected memory cells is made available at the output. In the write mode, information at the data input is written into (stored in) the selected cells. The address lines determine the cells written into or read from. Since each cell is a transistor circuit, a loss of dc power means a loss of data—a RAM that has this type of memory cell is said to provide *volatile storage*.

## ROM

A typical ROM chip is shown in Fig. 13.2b. An application in which the data does not change dictates the use of a ROM. For instance, a "lookup table" that stores the values of mathematical constants such as trigonometric functions or a fixed program such as that used to find the square root of a number could be stored in a ROM. The content of a ROM is fixed during manufacturing, perhaps by metallization or by the

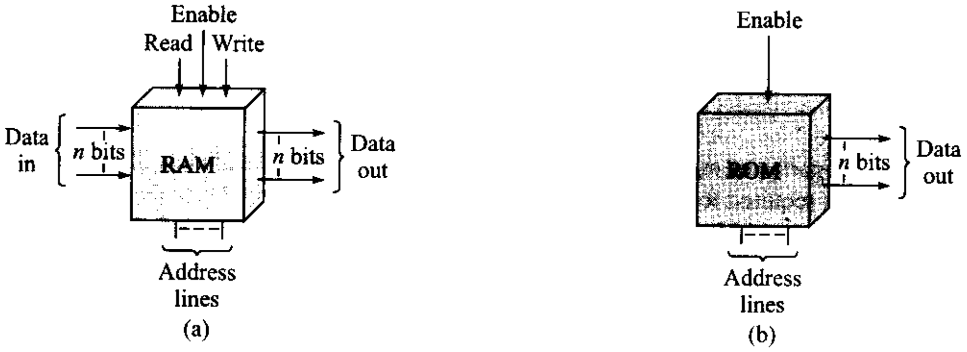


Fig. 13.2

presence or absence of a working transistor in a memory cell, by opening or shorting the gate structure, or by the oxide-layer thickness. A ROM is still random access, since there are logic circuitry and address lines to select any desired cell in the memory. When enabled, data from the selected cells is made available at the output. There is, of course, no write mode. Since data is permanently stored in each cell, a loss of power does not cause a loss of data, and thus a ROM provides *nonvolatile data storage.*

An application in which the data does not change but the required data will not be available until a later time suggests the use of a PROM, where the stored data can be set in the memory by writing into the PROM at the user's convenience. An application in which the data may change from time to time might call for the use of an EPROM.

**Example 13.1** State the most likely type of semiconductor memory for each application: (a) main memory in a hand calculator; (b) storing values of logarithms; (c) storing prices of vegetable produce; (d) emergency stop procedures for an industrial mill now in the design stage.

*Solution* (a) RAM; (b) ROM; (c) EPROM; (d) PROM.

**SELF-TEST**

1. **What is the operational difference between an SRAM and a DRAM?**
2. **What is an EPROM?**
3. **What is a cache memory?**

## 13.2 MAGNETIC MEMORY

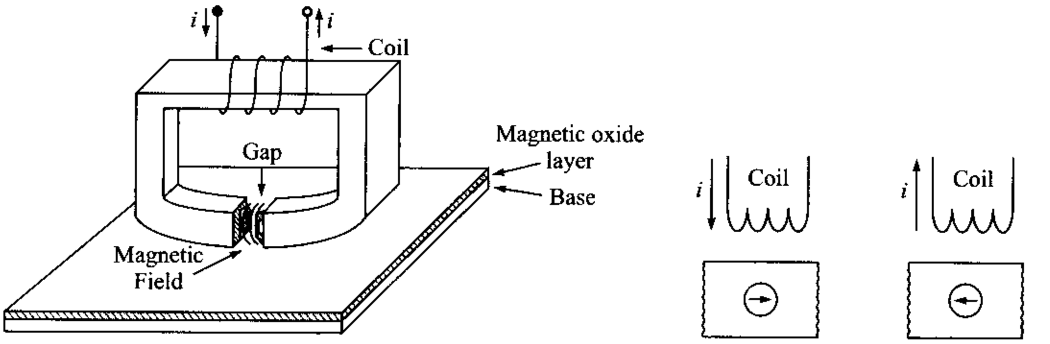
Magnetic tape, floppy disks, and hard disks are all capable of storing large quantities of digital data. A hard disk drive and a floppy disk drive are important components in nearly all microcomputer and minicomputer systems. Large reels of magnetic tape are economical and widely used mass storage components in large computer systems. The basic principle involved in each case is the magnetization of small spots in a thin film of magnetic material.

### Magnetic Recording

*Magnetic tape* is produced by the deposition of a thin film of magnetic material on a long strip of plastic, which is then wound on a reel. Magnetic material deposited on a rigid disk forms the basis of a *hard disk*; the same material on a *semirigid* disk is used to construct a *floppy disk.* Digital information is recorded on any of these surfaces in essentially the same fashion.

A current *i* in the coil shown in Fig. 13.3a or, the next page will produce a magnetic field across the gap. A portion of this field will extend into the magnetic material below the gap, and the material will be magnetized with a fixed orientation. When the current is removed, a magnetized spot remains, as shown in Fig. 13.3b. Thus, information has been stored. If the current is reversed in direction, a spot will again be magnetized, but with the opposite fixed orientation, as shown in Fig. 13.3c. Clearly this is a binary system, and it can be used to store binary information. For example, one could "define" Part b of Fig. 13.3 as 1 (high) and Part c as 0 (low). Introducing current *i* to record a 0 or a 1 is *writing* (or recording or storing) data.

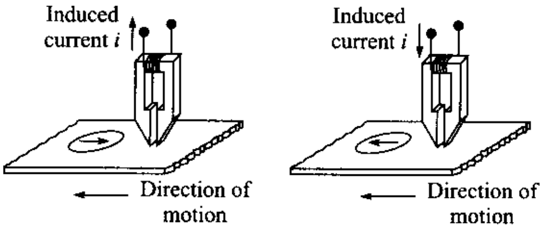
Now if a fixed, magnetized spot with a given orientation is moved past a gap as shown in Fig. 13.4a on the next page, a current with the direction shown will be induced in the coil. But if a magnetized



(a) Recording on a magnetic film (b) Recording a 1 (c) Recording a 0

Fig. 13.3

spot with the opposite orientation is moved past the gap, a current will be induced in the opposite direction, as shown in Fig. 13.4b. Detecting the orientation of the magnetized spot by measuring the induced current is *reading* information (1 or 0).



(a) Reading a 1 (b) Reading a 0

Fig. 13.4

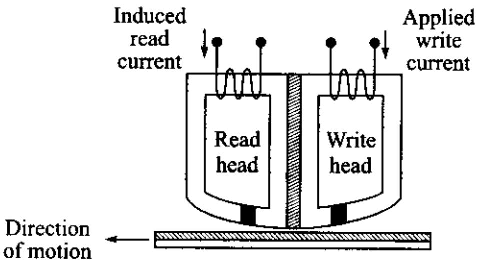
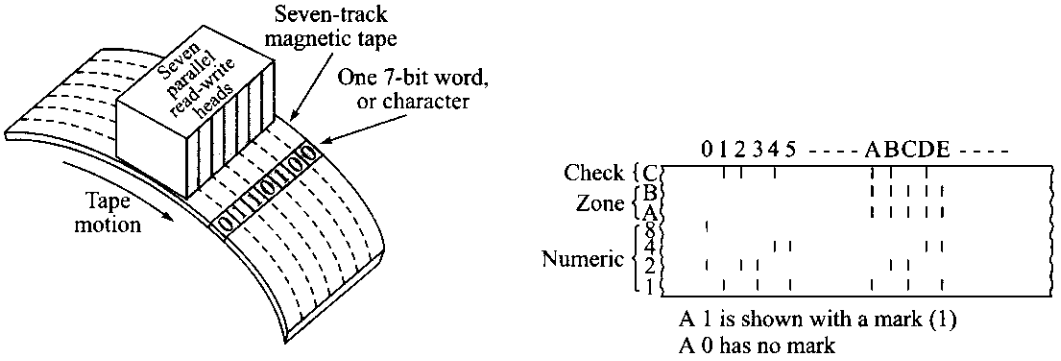


Fig. 13.5 Dual read-write head

The same magnetic read-write head in Fig. 13.3a can be used to write digital data or to read digital data. However, the dual read-write head in Fig. 13.5 is more common. Here's why. The tape or disk is moved under the heads in the direction shown. At time $t_1$, a spot is recorded under the write head. A short time later, at time $t_2$, this spot passes under the read head. It can then be read out and a check can be made to ensure that the correct data was in fact recorded.

## Magnetic Tape

Either seven or nine dual read-write heads are connected in parallel for use with magnetic tape as illustrated in Fig. 13.6a. As the tape moves past the heads, data is read or written, 7 (or 9) bits at a time. In the 7-bit system, alphanumeric information is recorded in coded form, and there is 1 parity bit (even or odd). There are numerous coding schemes, but a portion of a commonly used IBM code is shown in Fig. 13.6b. In the 9-bit system, a data word is composed of 8 bits, and the ninth bit is for parity (either even or odd). Data can be stored in coded form or in straight binary form.


Seven-track magnetic tape
One 7-bit word, or character
Seven parallel read-write heads
Tape motion
0 1 2 3 4 5 - - - - A B C D E - - - -
Check
C
Zone
B
A
Numeric
8
4
2
1
A 1 is shown with a mark (1)
A 0 has no mark

(a) Magnetic tape recording

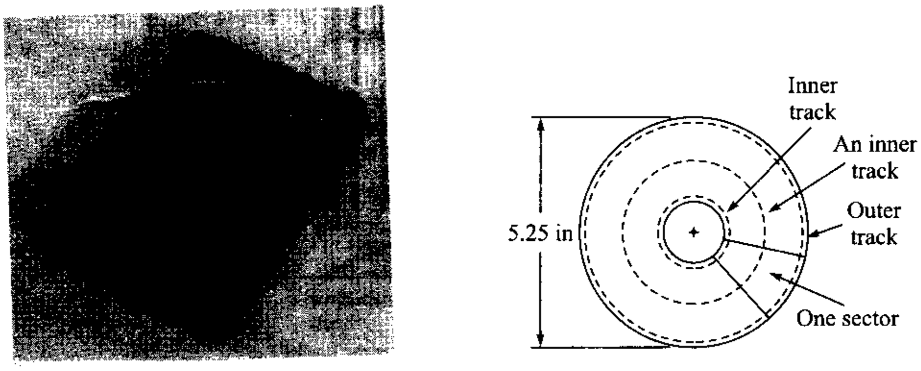
(b) A common 7-bit code

Fig. 13.6

Data storage on a magnetic tape is *sequential*. That is, data is stored one word after another, in sequence. To recover (read) data from the tape requires sequential searching. Clearly, the storage (or recovery) of data in a sequential system such as this requires considerably more time than storage (or recovery) using RAM. Tape is said to have a longer *access time* than RAM. Typical access times are measured in seconds, compared with nanosecond access times for RAMs.

## Hard Disks

Magnetic material deposited on a rigid disk (usually aluminum) is the basis for a hard disk system. One or more of these disks are mounted in an enclosure similar to that shown in Fig. 13.7a. The hard disks used in small computer systems are typically 3.5 in. or 5.25 in. in diameter. Hard disk drives with 40 to 400 gigabyte capacities are common in microcomputer systems. The disk is rotated at speeds between 3600 to 7200 rpm and in high end servers up to 15000 rpm resulting in typical access time of 16 ms to 3.6 ms. Because of the relatively short access times and the high storage density, hard disks are widely used in all computer systems.



(a)

(b) Hard disk

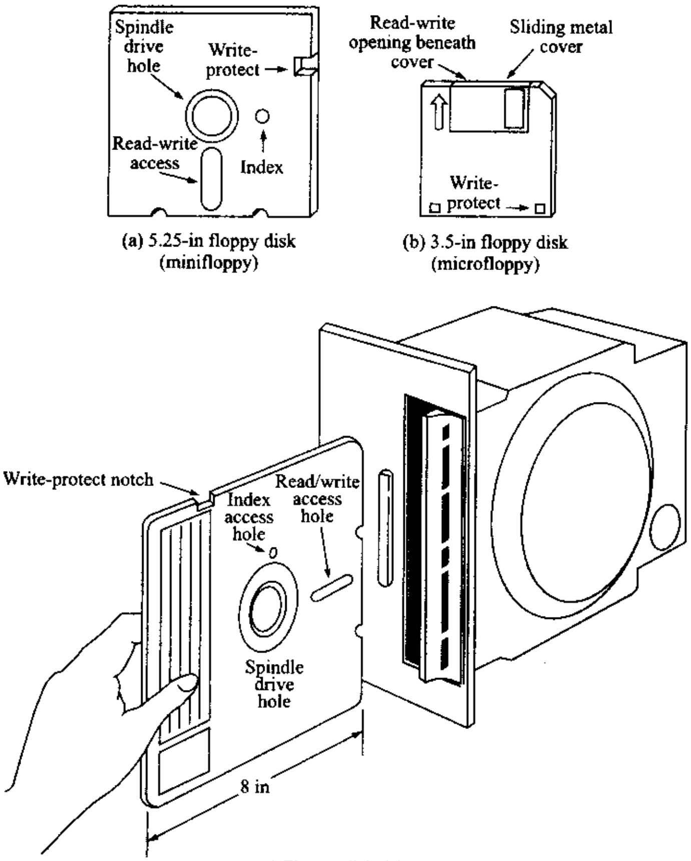
Fig. 13.7 Hard disk system

Information is stored in tracks (concentric rings) around the disk. The disk is further divided into sectors (pie-shaped sections), as shown in Fig. 13.7b. The number of tracks and sectors differ, depending on the computer system and on the individual manufacturer. The smaller hard disks used in microcomputer systems typically have 300 or so tracks.

Besides internal Hard Disks, a modern computer has the option to use external 3.5 inch hard drives having capacity of 80 GB and above, portable external 2.5 inch hard drive of capacity 40 GB to 120 GB and palm size pocket hard drive of capacity 2.5 GB or 5 GB.

## Floppy Disks

A floppy disk is formed by the deposition of magnetic material on a semirigid plastic disk housed in a protective cover as shown in Figs. 13.8a and b. The read-write opening provides access for the read-write head, and the index access hole allows the use of a photosensor to establish a reference position. When the write-protect notch is covered, data cannot be recorded on the disk, preventing accidental loss of data. Double-sided high-density 5.25-in disks have a capacity of 1.2 MB. Double-sided high-density 3.5-in disks have a capacity of 2.88 MB.



(a) 5.25-in floppy disk (minifloppy)

(b) 3.5-in floppy disk (microfloppy)

(c) Floppy disk drive

Fig. 13.8

As with hard disks, data is stored in tracks and the disk is divided into sectors. In IBM format, 5.25-in disks have 40 tracks per side and 3.5-in disks have 77 tracks per side. The IBM standard for sectors is nine.

The floppy disk is portable, and it must be inserted into a disk drive as shown in Fig. 13.8c. The drive unit consists of a single read-write head, read-write and control electronics, a drive mechanism, and a track-positioning mechanism. The spindle drive rotates the magnetic disk at a speed of 360 rpm. Access time is thus somewhat higher than the hard disk, being about 80 ms on average.

Note that, all the numbers that refer to maximum capacity, speed, given in this section or at other places are improving day by day by rapid technological advancements in this field.

**SELF-TEST**

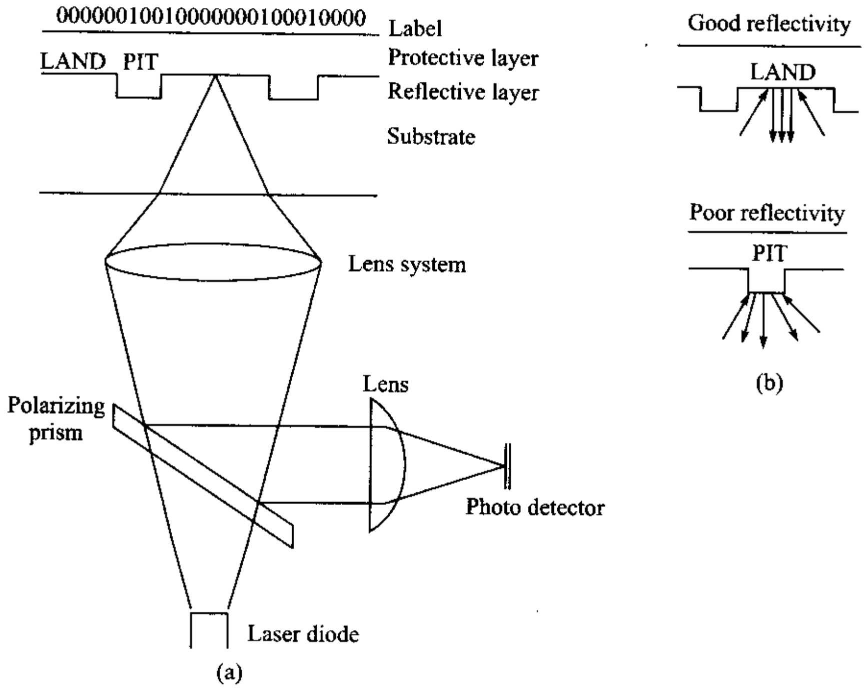
4. **Does the code in Fig. 13.6b have even or odd parity?**
5. **Magnetic tape provides inexpensive storage of large quantities of digital data. Why not use it, instead of RAM, in a microcomputer?**
6. **How can binary information be recorded on magnetic film?**

## 13.3 OPTICAL MEMORY

Introduced in 1982 jointly by Philips and Sony for storing digital audio data, Compact Disk (CD) found its way into computer storage in 1985. There was no looking back since then and today we find different types of CDs flooding the market where binary data is optically coded. The memory capacity of a CD is in the range of 650–700 MB, i.e. nearly 500 times more than 1.44 MB magnetic floppy disk. Both come in movable data storage category with almost same price tag but data integrity in optical disk is maintained over much longer period of time. Its newer variety called Digital Versatile Disk (DVD) can store data from 4.7 GB to 17.1 GB depending on configuration and make. Thus, the growth in optical storage media has been spectacular in last two decades. In this section we'll first discuss how CDs store binary data, what differentiates one type of CD from the other and then we'll look into DVD features.

### CD ROM

CD ROM or CD Read Only Memory devices are mass produced in factory using a stamp press technology. CD ROM drives uses LASER (Light Amplification by Stimulated Emission of Radiation) technology to read data from it. A semiconductor LASER generates a high intensity light wave of stable wavelength ≈ 780 nm. A lens system is used to direct the LASER towards the disk over approximately 1 micron diameter spot. Refer to simplified diagram of Fig. 13.9a. The intensity of the reflected light from metallic reflection layer, received by photo sensors gives the information of binary data is stored in CD. There are two different surfaces called *pit* and *land* from which reflection occurs. The pit is approximately 0.12 micron deep compared to land and reflected intensities are about 25% and more than 70% respectively (Fig. 13.9b). Every time laser beam travels from land to pit or pit to land there is a change in intensity of reflected light. This change is read as binary digit 1 and a constant intensity reflected light is interpreted as zero. The pit width is such that there is at least 2 and at most 10 zeroes between every 1. This is achieved by converting every 8-bit byte into a 14-bit value, a process called Eight to Fourteen Modulation (EFM). Such arrangement makes it easy for the read laser to detect bits and also helps in synchronization of internal clock as CDs are essentially self clocking. Data corresponding to a small portion of the track is shown above label in Fig. 13.9a.



(a)

(b)

**Fig. 13.9** A compact disk reading system

A compact disk normally comes in 12 cm diameter. The 1.2 mm thickness has four distinct parts. They are (a) label layer, (b) protective layer, (c) reflective layer, and (d) a transparent substrate layer on which land and pits are formed. The high reliability of CD comes from protection of data on one side by 10–20 microns thick protective lacquer layer and label and on the other side a tough approximately 1.2 mm thick polycarbonate layer. Thus, data integrity is maintained for years against most normal physical abuses and also for the fact that it is not susceptible to magnetic fields or radiations. Note that, small scratches on the surface of CD do not directly erase the data, but create additional areas of light scattering. This can make things difficult for drive's electronic, which is also much less sensitive to radial scratches than to the circumferential ones. The other reason that increases reliability of data stored in a CD is the ability to use efficient error correction codes. The data is stored in the form of a spiral of around 20000 windings totaling approximately 4.5 km of length and contains nearly 2 billion shallow pits on its surface. A macroscopic view of a part of CD is presented in Fig. 13.10.

You definitely have seen some kind of symbol like 48X, 52X etc. written on a CD. What is that? It gives the speed at which data is read from CD where 1X stands for 150 KB/Sec. Earlier versions of CD ROM drive below 12X were built on constant linear velocity (CLV) where motor had variable speed to maintain CLV. Present-day drives are based on constant angular velocity (CAV) where motor rotates at constant speed and this requires less seek time to access data from CD.

## CD-R

CD-R or CD-Recordable allows user to write data but once. The CD-R drive has laser unit which uses higher intensity light wave for write operations than read. CD-R disk does not have pits and lands but a photosensitive organic dye (between reflective layer and polycarbonate substrate) that write laser heats

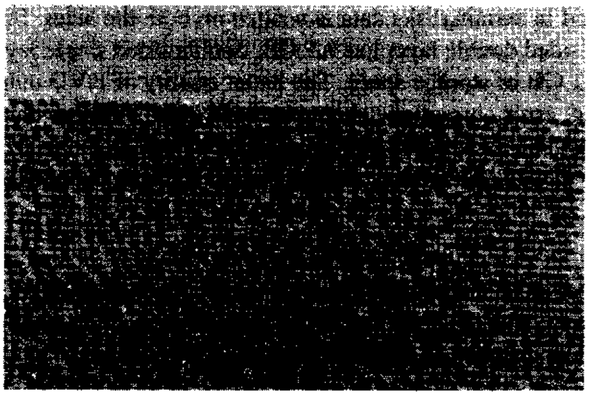


Fig. 13.10 A macroscopic view of a part of compact disk surface

to approximately 250°C. This melts or chemically decomposes the dye to form a depression mark in the recording layer in appropriate places. The places burnt have lower reflectivity of light. Thus read laser gets two different intensities on reflected light while reading the disk, similar to read operation of a CD ROM. Earlier versions of CD-R called WORM, abbreviation of write once read many times required data to be written in only one session or one go. Now it is possible to write data in CD-R in multiple sessions till it is completely filled. The writing speed of CD-R is much slower than the read speed.

## CD-RW

CD-RW or CD Read Write, previously known as CD-Erasable gives user facility to write and erase data many times, unlike CD-R. CD-RW uses an active layer of Ag-In-Sb-Te (silver-indium-antimony-tellurium) alloy that has a polycrystalline structure making it reflective (reflectivity 25%). Writing data on disk uses highest power of laser that heats up selected spots to 500°–700°C. At this temperature the chemical structure liquefies losing its polycrystalline structure and on cooling solidifies to an amorphous state that has reduced reflectivity of 15%. The read process is like CD-ROM and CD-R that notes the difference in reflectivity of the reflecting surface.

To reverse the phase or erase data, the laser operates at a lower power setting and heats the active material to nearly 200 °C. This reverses the material from its amorphous to its polycrystalline state and then becomes reflective again. According to manufacturers, in a CD-RW the rewrite operations can be done 1000 times or more. The main drawback of CD-RW is very low reflectivity of the material and the difference between two levels is also not much. This often limits readability of these devices. Note that CD-Recordable drives often come with three different speed ratings, one speed for write-once operations, one for re-write operations, and one for read-only operations. The speeds are typically listed in that order, e.g. 12X/10X/32X. This means CPU and media-permitting, CD drive can write to CD-R disks at 12X speed, write to CD-RW disks at 10X speed and read from CD disks at 32X speed.

## DVD

Digital Versatile Disk or Digital Video Disk, popular as DVD resemble compact disk in dimension and look but contains much higher storage space. DVD driver uses smaller wavelength (635 nm or 650 nm) and lower numerical aperture of lens system to read smaller dimension land and pits. Each side can have two layers from which data is read and in certain disks data is written on both the sides. Single sided, single layer has capacity of 4.7 GB, single sided double layer has 8.5 GB, double sided single layer has 9.4 GB while double sided double layer has 17.1 GB of storage space. The better quality of DVD output compared to CD comes from better channel coding, error correction scheme and of course a higher data transfer rate. Note that in DVD terminology, 1X refers to 1.32 MB/Sec unlike 150 KB/Sec of CD. Like CD, different varieties of DVD like DVD-R, DVD-RW, DVD-RAM are being developed and entering the market.

## Handling Tips

Before we complete this section let us provide you some useful tips for handling CDs, the most used movable storage device of these days. Of course, the list is not exhaustive and the requirements come mainly from maintaining a good reflecting surface and physical balance of the CD.

(i) Handle disks by the outer edge or the center hole. Don't touch the surface of the disk to avoid leaving fingerprints and oil behind. Keep the disk free of dirt.

(ii) Clean dirt, smudges, and liquids from disks by wiping with a clean cotton fabric in a straight line radially outward from the center of the disk. Don't wipe in circles. The error correction codes on the disk can handle only small interruptions like scratch that travels across the spiral. You may clean stubborn dirt and foreign substances with 99% isopropyl alcohol or 99% methyl alcohol. First apply the cleaner to a cotton, then rub the cloth across the disk, taking care not to get any fluid on the label side of the disk.

(iii) Label the disk with a non-solvent-based felt-tip permanent marker. Beware of permanent markers that contain strong solvents. The use of adhesive labels is not recommended. If you use a label, don't try to remove or reposition it.

(iv) Never ever bend the disk. Flexing the disk can cause stress patterns to form in the polycarbonate, and if you stretch it far enough the reflective and recording layers get deformed. Store disks vertically. Over a long period, gravity will warp the disk if it's left flat.

(v) Store disks in a cool, dry, dark environment in which the air is clean to avoid corrosion. Keep it away from areas that are excessively hot or damp and also from direct sunlight and other ultraviolet light sources. Do not expose the disk to rapid changes in temperature or humidity.

**SELF-TEST**

**7. What is the wavelength of laser used for reading CD-ROM?**

**8. What is the reflectivity of CD-ROM and CD-RW surfaces?**

**9. What is the capacity of a single sided double layer DVD ROM?**

## 13.4 MEMORY ADDRESSING

### Cell Selection

Addressing is the process of selecting one of the cells in a memory to be written into or to be read from. In order to facilitate selection, memories are generally arranged by placing cells in a rectangular arrangement of rows and columns as shown in Fig. 13.11a. In this particular case, there are $m$ rows and $n$ columns, for a total of $n \times m$ cells in the memory.

The control circuitry that accompanies the basic memory array is designed such that if one and only one row line is activated and one and only one column line is activated, the memory cell at the intersection of these two lines is selected. For instance, in Fig. 13.11b, if row *A* is activated and column *B* is activated, the cell at the intersection of this row and column is selected—that is, it can be read from or written into. For convenience, this cell is then called *AB*, corresponding to the row and the column selected. This designation is defined as the *address* of the cell. The activation of a line (row or column) is achieved by placing a logic 1 (or perhaps a logic 0) on it.

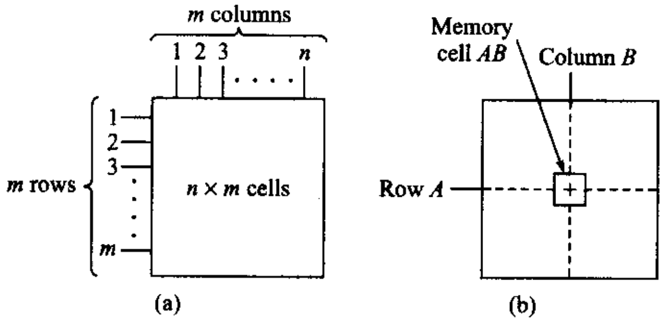


**Fig. 13.11** **(a) A rectangular array of $m \times n$ cells, (b) Selecting the cell at memory address *AB***

## Matrix Addressing

Let's take a little time to consider the various possible configurations for a rectangular array of memory cells. The different rectangular arrays of 16 cells are shown in Fig. 13.12. In each of the five cases given, there are exactly 16 cells. The 16 × 1 and the 1 × 16 arrangements in Fig. 13.12a are really equivalent; likewise, the 8 × 2 and the 2 × 8 are essentially the same. So, there are really only three different configurations, each of which contain the exact same number of cells.

For any of the three configurations, the selection of a single cell still requires a single row and a single column to define a unique address. In Fig. 13.12a, a total of 17 address lines must be used—16 rows and 1 column, or 1 row and 16 columns. The minimum requirement in either case is really only 16 lines. However, either arrangement in Fig. 13.12b requires only 10 address lines—8 rows and 2 columns, or 2 rows and 8 columns. Clearly the best arrangement is given in Fig. 13.12c, since this configuration only requires 8 address lines—4 rows and 4 columns!

In general, the arrangement that requires the fewest address lines is a square array of *n* rows and *n* columns for a total memory capacity of $n \times n = n^2$ cells. It is exactly for this reason that the square configuration is so widely used in industry. This arrangement of *n* rows and *n* columns is frequently referred to as *matrix addressing*. In contrast, a single column that has *n* rows (such as the 16 × 1 array of cells) is frequently called *linear addressing*, since selection of a cell simply means selection of the corresponding row, and the column is always used.

For instance, a 74S201 is a 256-bit bipolar RAM, arranged in a 256 × 1 array. The IEEE symbol for the 74S201 ('S201) is given in Fig. 13.13 on the next page. Eight address lines ($A_0, A_1, \ldots, A_7$) are required to select one of the 256 cells. There are three chip select lines ($\bar{S}_1$, $\bar{S}_2$ and $\bar{S}_3$), all of which must be low in order to activate (select) the chip. When the $R/\overline{W}$ line is high, the data bit at input *D* is stored at the selected address. When the $R/\overline{W}$ line is low, the *complement* of the bit at the selected address appears at the $\overline{Q}$ output.

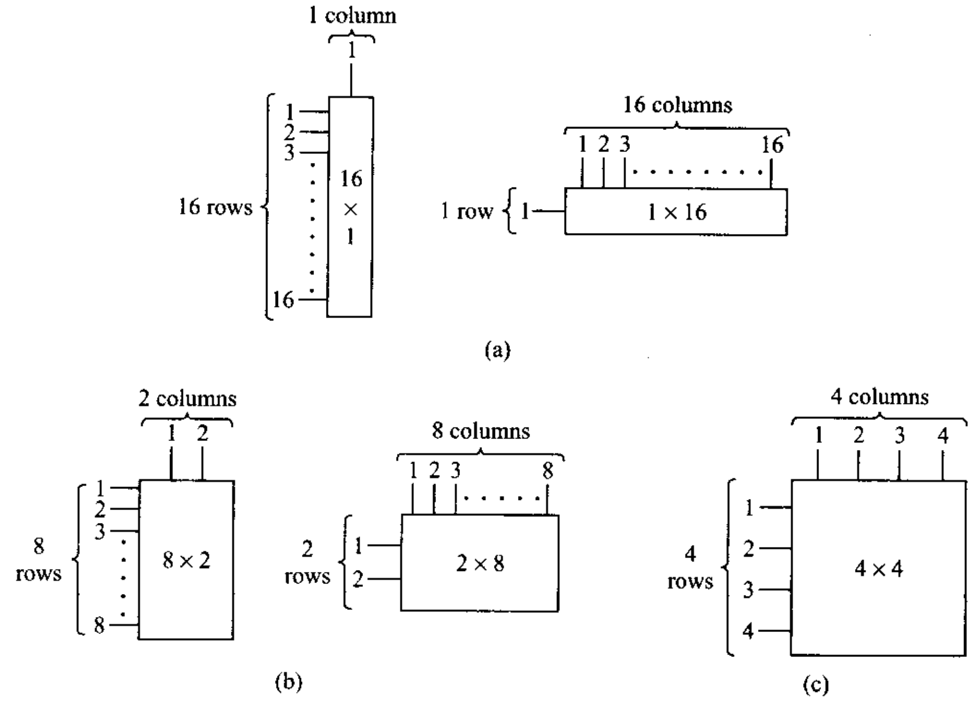


Fig. 13.12

The small triangle (∇) at the $\overline{Q}$ output means that the output is three-state (tri-state). We'll use this chip in Sec. 13.5.

## Address Decoding

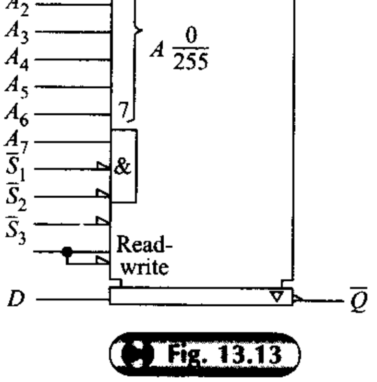


Fig. 13.13

Take another look at the 4 × 4 memory in Fig. 13.12c. To select a single cell, we must activate one and only one row, and one and only one column. This suggests the use of two 1 of 4 binary to decimal decoders as shown in Fig. 13.14. Consider the selection of the cell at address 43 (row 4 and column 3). If $A_4 = 1$ and $A_3 = 1$, the decoder will hold the row 4 line high while all other row lines will be low. Similarly, if $A_2 = 1$ and $A_1 = 0$, the decoder will hold column 3 high and all other column lines low. Thus an input $A_4A_3A_2A_1 = 1110$ will select cell 43. We can consider $A_4A_3$ as a row address of 2 bits and $A_2A_1$ as a column address of 2 bits. Taken together, any cell in the array can be uniquely specified by the 4-bit address $A_4A_3A_2A_1$. As another example, the address $A_4A_3A_2A_1 = 0110$ selects the cell at row 2 and column 3 (address 23).

The address decoders shown in Fig. 13.14 further reduce the number of address lines needed to uniquely locate a memory cell, and they are almost always included on the memory chip. Recall that a binary-to-decimal decoder having $n$ binary inputs will select one of $2^n$ output lines. For instance, a decoder that has 3 binary inputs will have $2^3 = 8$ outputs, or a decoder having 4 inputs will have 16 outputs, and so on.

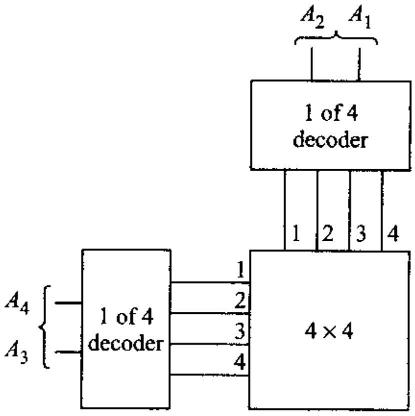


Fig. 13.14

In general, an address of $B$ bits can be used to define a square memory of $2^B$ cells, where there are $B/2$ bits for the rows and $B/2$ bits for the columns, as shown in Fig. 13.15. Notice that the total number of address bits $B$ must be an even integer (2,4,6,8, ...). Since the input to each decoder is $B/2$ bits, the output of each decoder must be $2^{B/2}$ lines. So the capacity of the memory must be $2^{B/2} \times 2^{B/2} = 2^B$. For instance, an address of 12 bits can be used in this way for a memory that has $2^{12} = 4096$ bits. There will be 6 address bits providing $2^6 = 64$ rows and likewise 6 address bits providing 64 columns. The memory will then be arranged as a square array of 64 × 64 = 4096 memory cells.

You may have noticed that most commercially available memories have capacities like 1024, 2048, 4096, 16,384, and so on. The reason for this is now clear—all of these numbers are clearly integer powers of 2! Incidentally, a memory having 1024 bits is usually referred to as a 1K memory (1000 bits) simply for convenience. Similarly, a memory advertised as 16K really has 16,384 bits. 4K is really 4096, and so on.

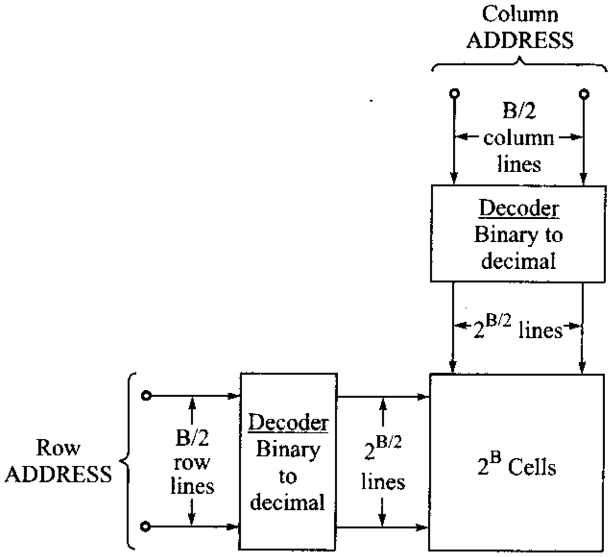


Fig. 13.15

**Example 13.2** What would be the structure of the binary address for a memory system having a capacity of 1024 bits?

*Solution* Since $2^{10} = 1024$, there would have to be 10 bits in the address word. The First 5 bits could be used to designate one of the required 32 rows, and the second 5 bits could be used to designate one of the required 32 columns. Notice that 32 × 32 = 1024.

**Example 13.3** For the memory system described in the previous example, what is the decimal address for the binary address 10110 01101? What is the address in hexadecimal?

*Solution* The first 5 bits are the row address. Thus row = 10110 = 22. The second 5 bits are the column address. So, column = 13. The decimal address is thus 22 13. In hexadecimal, this same address is 16 0D.

## Expandable Memory

So far, we have only discussed memories that provide access to a single cell or bit at a time. It is often advantageous to access groups of bits—particularly groups of 4 bits (a nibble) and groups of 8 bits (a byte). It is not difficult to extend our discussion here to accommodate such requirements. There are at least two popular methods. The first simply accesses groups of cells on the same memory chip, and we discuss this idea next. The second connects memory chips in parallel, and we consider this technique in a following section.

The logic diagram for a 64-bit (16 × 4) bipolar memory is given in Fig. 13.16. There are 16 rows of cells with four cells in each row; thus the description (16 × 4). Each cell is a bipolar junction transistor flip-flop. The address decoder has 4 address bits and thus 16 select lines—one for each row. In this case, each select line is connected to all four of the cells in a row. So, each select line will now select four cells at a time. Therefore, each select line will select a 4-bit word (a nibble), rather than a single cell.

You might think of this arrangement as a "stack" of sixteen 4-bit registers. This is really a form of linear addressing, since the 4 address bits, when decoded, select one of the sixteen 4-bit registers. In any case, when data is read from this memory it appears at the four data output lines $\overline{D}_1$, $\overline{D}_2$, $\overline{D}_3$, and $\overline{D}_4$ as a 4-bit data word. Similarly, data is presented to the memory for storage as a 4-bit data word at input lines $I_1$, $I_2$, $I_3$, and $I_4$. The 74S89 and the 74LS189 both are 64-bit (16 × 4) bipolar scratch pad memories arranged in exactly this configuration (look ahead in Fig. 13.22). The idea is easily extended to memories that access a word of 8 bits (a byte) at a time—for instance, the TBP18S030 ROM discussed in the next section.

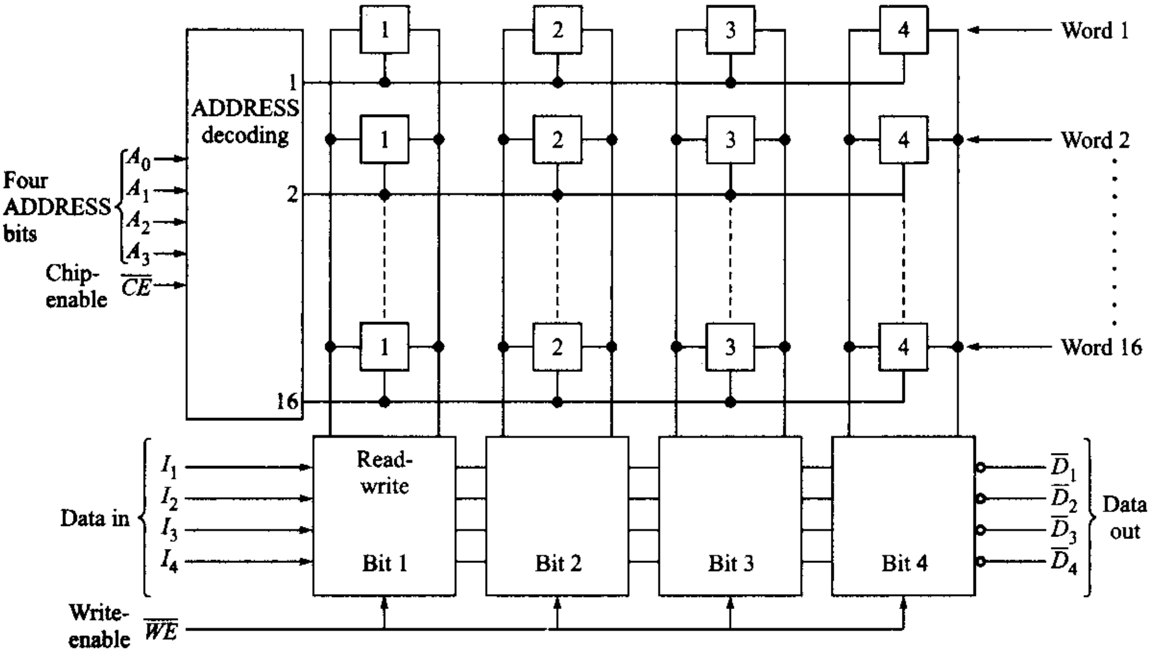


Fig. 13.16 64-bit (16 × 4) memory

SELF-TEST

10. What binary address will select cell 145 (decimal) in the 74S201 in Fig. 13.13?
11. The address applied in Fig. 13.14 is $A_4A_3A_2A_1$ = 1010. What cell is being accessed?

## 13.5 ROMs, PROMs, AND EPROMs

Having gained an understanding of memory addressing, let's turn our attention to the operation of a ROM. The term ROM is generally reserved for memory chips that are programmed by the manufacturer. Such a chip is said to be *mask-programmable*, in contrast to a PROM, which is said to be *field-programmable*—that is, it can be programmed by the user. EPROMs can be programmed, erased, and programmed again; they are clearly much more versatile chips than PROMs. Refer to Section 4.9 of Chapter 4 for a detailed discussion on its internal circuitry.

### Programming

What exactly does programming a ROM, PROM or EPROM involve? It simply involves writing, or storing, a desired pattern of 0s and 1s (data). Each cell in the memory chip can store either a 1 or a 0. As supplied from the manufacturer, most chips have a 0 stored in each cell. The chip is then programmed by entering 1s in the appropriate cells. For instance, the content of every 4-bit word in a 64 × 4 chip is initially 0000. If the desired content of a word is to be 0110, then the two inner bit positions will be altered to 1s during programming.

In the case of a ROM, you must supply the manufacturer with the exact memory contents to be stored in the ROM. The Texas Instruments TMS4732 is a ROM having 4096 eight-bit words (a 4096 × 8 ROM). The logic diagram is given in Fig. 13.17. The 8-bit word length makes this NMOS (*n*-channel MOS) chip ideal for microprocessor applications. Texas Instruments will store user-specified data during manufacturing. The user must supply data storage requirements in accordance with detailed instructions given on the TMS4732 data sheet.

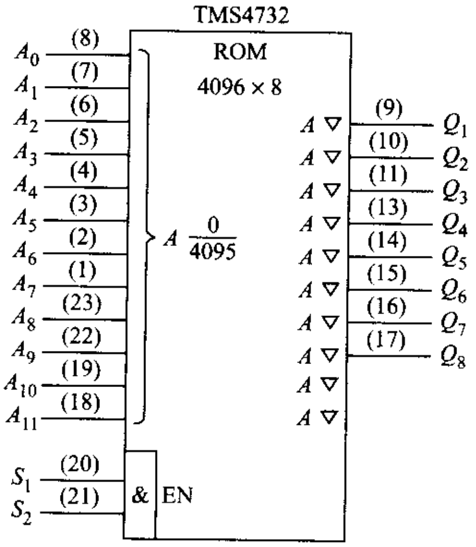


Fig. 13.17

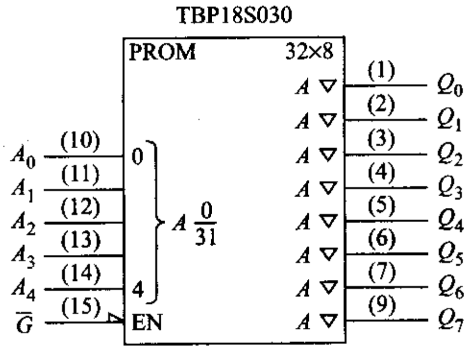
The Texas Instruments TBP18S030 is a bipolar memory chip arranged as thirty-two 8-bit words (256 bits). The logic diagram for this user-programmable PROM chip is given in Fig. 13.18. Basically, the programming is done by applying a current pulse to each output terminal where a logic 1 must appear (be stored). The current pulse will destroy an *existing fuse link*. When the fuse link is present, the transistor circuit in that cell stores a 0. After the fuse link is destroyed, the circuit stores a 1. A typical programming sequence might be:

1. Apply the proper dc power supply voltage(s) to the chip; in the case of the TBP18S030, +5 Vdc. Disable the chip (the enable input is high).

2. Apply the address of the word to be programmed ($A_0$, $A_1$, $A_2$, $A_3$, $A_4$). For instance, to programm the word at address 14H (hex 14). apply

   $$A_0A_1A_2A_3A_4 = 10100$$

3. To store the word $Q_0Q_1Q_2Q_3Q_4Q_5Q_6Q_7$ = 00101000:
   a. Ground output $Q_2$ and connect all other outputs to +5 Vdc through a 3.9-kΩ resistor. Raise the +5-Vdc supply to +9.25 Vdc and momentarily enable the chip. This will program a 1 in bit position $Q_2$.
   b. Repeat (a) for bit position $Q_4$. This will program a 1 in bit position $Q_4$.
4. Repeat steps 2 and 3 for each word to be programmed.



(a)

Fig. 13.18 (a) Logic symbol

## ROMs

The logic diagram for the Texas Instruments TMS4732, a 4096 × 8 ROM, is given in Fig. 13.17. Twelve address bits are required, $A_0$, $A_1$, … , $A_{11}$ ($2^{12}$ + 4096). There are two chip-enable inputs, $S_1$ and $S_2$. Both $S_1$ and $S_2$ must be high in order to enable the chip. Each of the eight data output lines is a three-state line (the small ∇ symbol). As mentioned previously, this chip is ideal for microprocessor applications because of the 8-bit word length. This ROM is mask-programmable, and data must be specified for the manufacturer before purchase.

Texas Instruments offers a number of other ROMs with larger memory capacity, all of which are LSI NMOS devices.

TMS4664: 8192 × 8-bit
TMS4764: 8192 × 8 bit
TMS47128: 16,384 × 8-bit
TMS47256: 32,768 × 8-bit

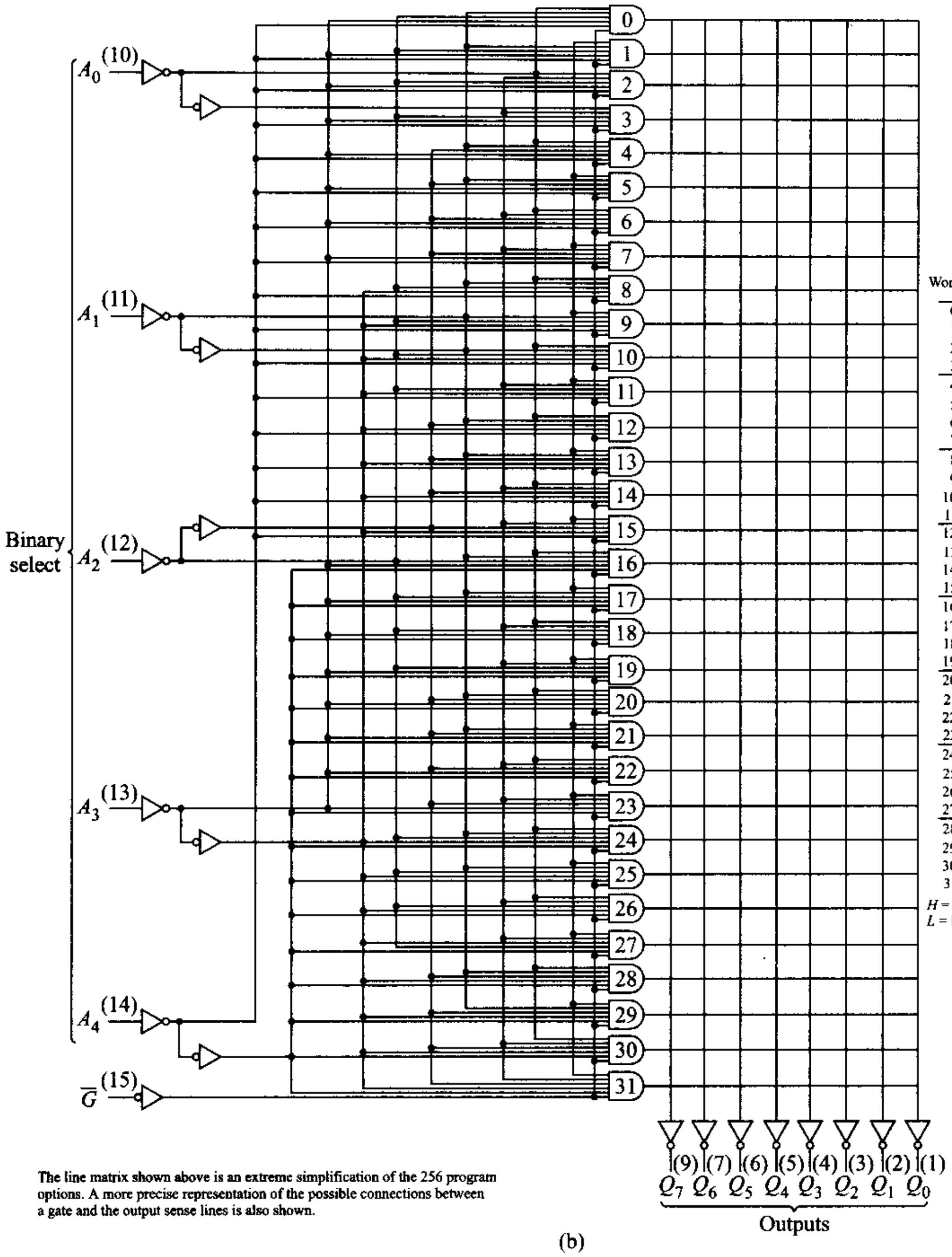
## PROMs

The TBP18S030 is a 256-bit (32 × 8) PROM arranged as a stack of thirty-two 8-bit words. The 74S288 is an equivalent designation. As shown in Fig. 13.18, the 5 row address bits are labeled $A_0$, $A_1$, $A_2$, $A_3$, $A_4$ and the 8 output bits in a word are labeled $Q_0$, $Q_1$, $Q_2$, $Q_3$, $Q_4$, $Q_5$, $Q_6$, $Q_7$.

Input $\overline{G}$ is used to enable or disable the entire set of 32 input decoding gates. When $\overline{G}$ is high, all the address decoding gates are inhibited and the memory chip is disabled, causing the eight output data bit lines to be high. When $G$ is low, the data at the outputs will correspond to the 8-bit word in memory selected by the input address. On most memory chips there is a chip-enable or chip-select input line that performs the same function as $\overline{G}$.

Using the TBP18S030 PROM is relatively simple. First, since the logic circuits are TTL, a supply voltage and ground connections must be made. The data sheet calls for a nominal supply voltage of $+V_{CC}$ = 5.0 Vdc

Functional block diagram and word selection



Word select table

| Word | Inputs | | | | |
|---|---|---|---|---|---|
| | $A_4$ | $A_3$ | $A_2$ | $A_1$ | $A_0$ |
| 0 | L | L | L | L | L |
| 1 | L | L | L | L | H |
| 2 | L | L | L | H | L |
| 3 | L | L | L | H | H |
| 4 | L | L | H | L | L |
| 5 | L | L | H | L | H |
| 6 | L | L | H | H | L |
| 7 | L | L | H | H | H |
| 8 | L | H | L | L | L |
| 9 | L | H | L | L | H |
| 10 | L | H | L | H | L |
| 11 | L | H | L | H | H |
| 12 | L | H | H | L | L |
| 13 | L | H | H | L | H |
| 14 | L | H | H | H | L |
| 15 | L | H | H | H | H |
| 16 | H | L | L | L | L |
| 17 | H | L | L | L | H |
| 18 | H | L | L | H | L |
| 19 | H | L | L | H | H |
| 20 | H | L | H | L | L |
| 21 | H | L | H | L | H |
| 22 | H | L | H | H | L |
| 23 | H | L | H | H | H |
| 24 | H | H | L | L | L |
| 25 | H | H | L | L | H |
| 26 | H | H | L | H | L |
| 27 | H | H | L | H | H |
| 28 | H | H | H | L | L |
| 29 | H | H | H | L | H |
| 30 | H | H | H | H | L |
| 31 | H | H | H | H | H |

$H$ = high level
$L$ = low level

The line matrix shown above is an extreme simplification of the 256 program options. A more precise representation of the possible connections between a gate and the output sense lines is also shown.

(b)

Fig. 13.18 (b) TBP18S030 PROM (74S288)

on pin 16, with ground connected to pin 8. The inputs and outputs are all TTL-compatible. The eight data outputs are three-state (note the symbol ∇ at each output).

Now, all that is required is to apply the correct input address to read a desired 8-bit word and then take the $\overline{G}$ input line (select line) low. The TBP18S030 data sheet states an access time $t_p$ of 12 ns (typical) and 25 ns (maximum). So, an 8-bit data word will be available at the outputs $Q_0 \dots Q_7$ within 25 ns after the falling edge of $\overline{G}$ , as shown in Fig. 13.19a. The address lines should, of course, be stable during the time data is being read out of the memory. There are two output lines in Fig. 13.19a showing that a data line may transition low to high, or high to low. To save time and space, this idea is usually conveyed in a single waveform as seen in Fig. 13.19b—this single waveform is the equivalent of the two output waveforms above it in Fig. 13.19a.

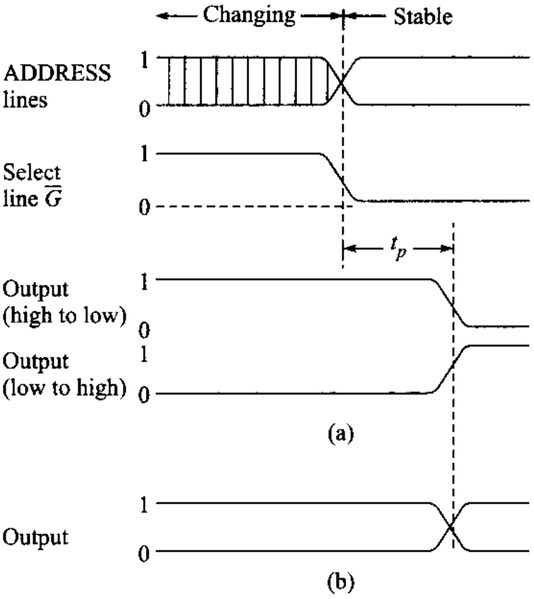


Fig. 13.19 Access time to for a TPB18S030 PROM

Example 13.4 The TTL LSI TBP24S10 is advertised as a 1024-bit PROM. Since $2^{10} = 1024$, it would seem to require 10 address bits, but the data sheet shows only 8 bits of address. Can you explain how the memory on this chip must be organized by looking at the logic diagram in Fig. 13.20?

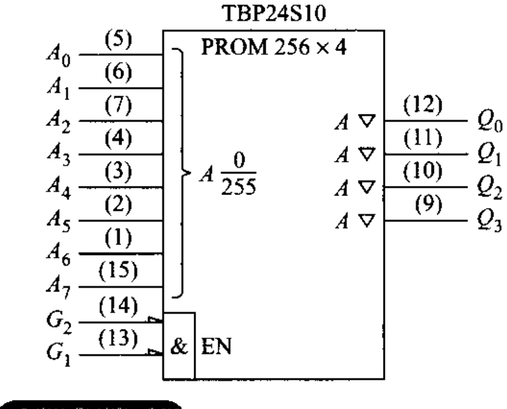


Fig. 13.20 Texas Instruments TPB24S10 PROM

*Solution* There are 4 bits appearing at the output of the chip, so it must be organized as 256 words of 4 bits each (256 × 4 = 1024). The 1024 memory cells are arranged in a square consisting of 32 rows and 32 columns. Five of the address bits (*DEFGH*) are used to select one of the 32 rows ($2^5 = 32$). The 32 columns are divided into eight groups of 4 bits each. So, it is only necessary to select one of the eight groups, and this can be done with three address lines (*ABC*), since $2^3 = 8$. As an example, the address *HGFED-CBA* = 10110 110 will select row 10110 = 22 and the 4 bits (four columns) in group 110 = 6.

## EPROMs

One disadvantage of a PROM is that once it is programmed, the contents are stored in that memory chip permanently—it can't be changed; a mistake in programming the chip can't be corrected. The EPROM overcomes this difficulty.

The EPROM has a structure and addressing scheme similar to those of the previously discussed PROM, but it is constructed using MOS devices rather than bipolar devices. Many MOS EPROMs are TTL-compatible, and even the technique used to program the chip is similar to that used with a bipolar memory. The only difference is really the mechanism for permanently storing a 1 or a 0 in an MOS memory cell.

The current pulse used to store a 1 when programming a bipolar PROM is used to destroy ("burn out") a connection on the chip. The same technique is used to program an MOS-type EPROM, but the current pulse is now applied for a period of time (usually a few milliseconds) in order to store a fixed charge in that particular memory cell. This stored charge will cause the cell to store a logic 1.

The interesting thing about this phenomenon is that the charge can be removed (or erased), and the cell will now contain a logic 0! Furthermore, the process can be repeated. The memory cells are "erased" by shining an ultraviolet light through a quartz window onto the top of the chip. The light bleeds off the charge and all cells will now contain 0s. The requirements for programming and erasing an EPROM vary widely from chip to chip, and data sheet information must be consulted in each individual case.

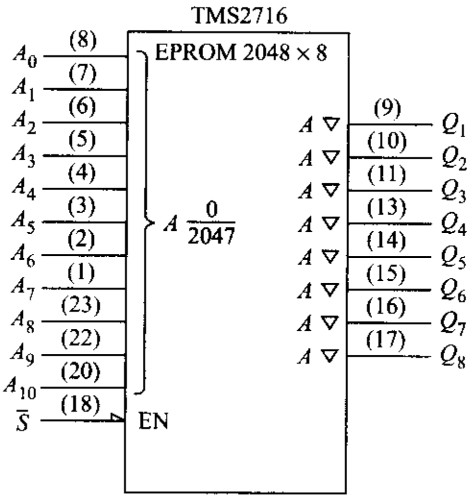


Fig. 13.21 2716 EPROM

The logic diagram for a 2716, a 16K (2K × 8) EPROM, is given in Fig. 13.21. There are actually 2048 words, 8 bits each, for a total storage capacity of 16,384 bits. The chip is completely TTL-compatible and has an access time of less than 450 ns. The 11 address bits will uniquely select one of 2048, 8-bit words ($2^{11}$ = 2048), and the selected word will appear at the data output lines if chip-select is low. The 2732 is a 32K (4K × 8) EPROM that is pin-compatible with the 2716—it simply has twice the memory storage. Likewise, the 2764 is a 64K (8K × 8) EPROM.

As a matter of fact, the logic diagram in Fig. 13.21a is virtually the same for any ROM, PROM, or EPROM. Essentially the only variation is the total number of address inputs to accommodate the number of bits in the cell matrix. As an example, a CMOS EPROM with essentially the same logic diagram is the 27C512. But this chip has 16 address input lines and a 534,288-bit cell matrix, organized as 65,536 words, each 8 bits in length. This is most impressive when you consider that there are over a half-million bits of information stored in a 28-pin package that measures less than 1.5 in. in length and about 0.5 in. in width!

## EEPROM, Flash Memory

*Electrically Erasable Programmable Read Only Memory* (EEPROM) is similar to EPROM as far as writing into memory is considered, i.e. effecting a current pulse to store charge. The erasing, however, is different and is done by removing the charge and sending a pulse of opposite polarity. There are two types of EEPROM—parallel and serial. Parallel EEPROM is faster, costlier and comes in 28xx family. Their pinout and functioning is similar to 27xx EPROM family. Serial EEPROM is slower, cheaper, uses lower number of pins and comes in 24xx family.

*Flash memory* is a further advancement on EEPROM. This, too, writes and erases data electrically—can be both parallel and serial type. The number of write/erase cycle is finite and often, there is a separate management scheme to take note of this. There is an internal voltage generation block that takes single voltage supply and generates different voltages required for writing and erasing. Different manufacturers have created different standards for flash memory chip which differ in pinout, memory organization, etc. Intel family chips are 28Fxxx while AMD chips are numbered as 29Fxxx.

**SELF-TEST**

12. **What does it mean to say that a chip is mask-programmable?**
13. **What is the meaning of the small triangle on each output line of the TMS4732 in Fig. 13.17?**
14. **The 74S288 in Fig. 13.18 is programmed with 1011 0001 in word 20. The desired content at this word address is 1011 1001. Can this be corrected?**

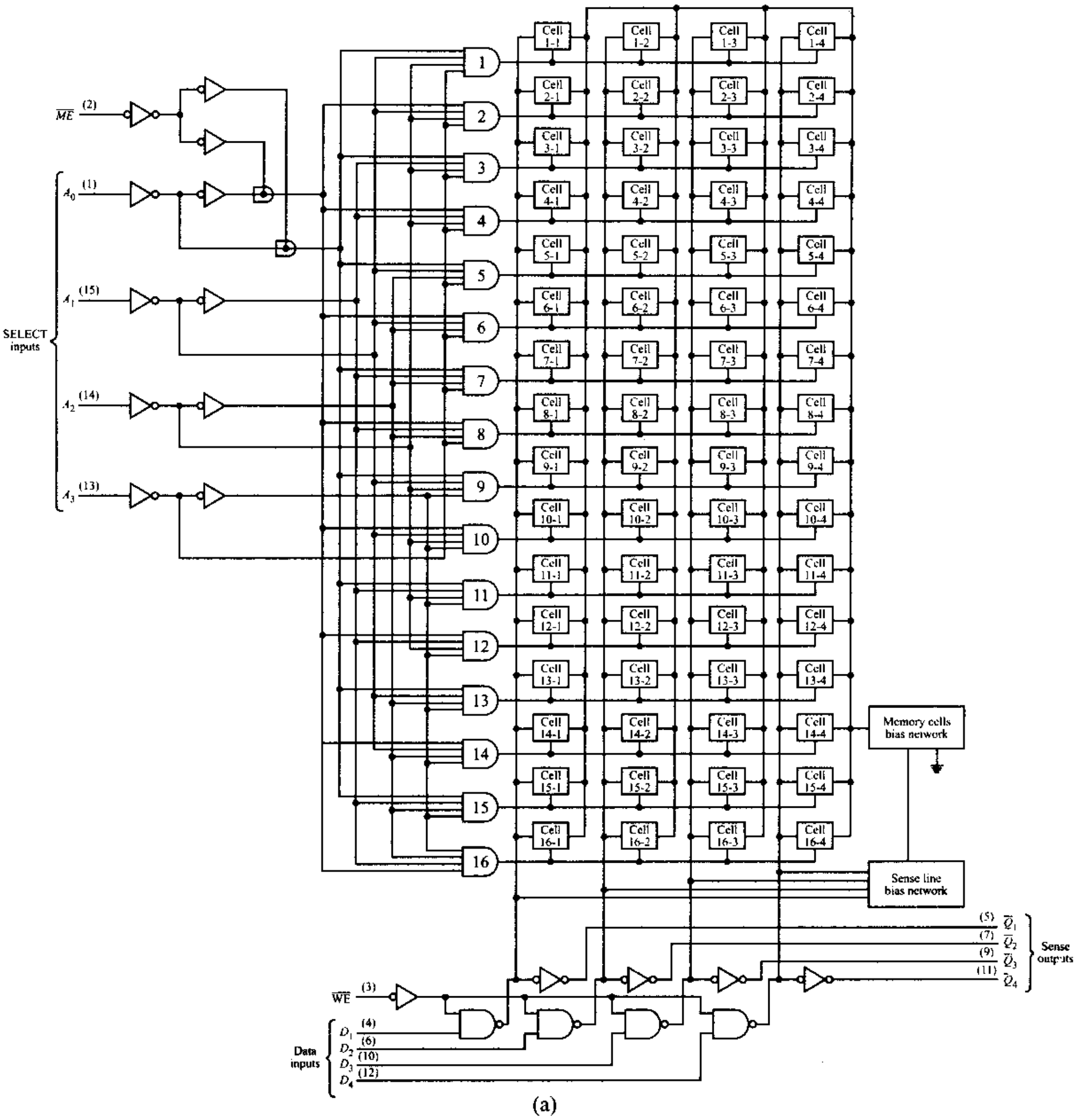
## 13.6 RAMs

The basic difference between a RAM and a ROM is that data can be written into (stored in) a RAM at any address as often as desired. Naturally data can be read from any address in either a RAM or a ROM, and the addressing and read cycles for both devices are similar. The characteristics of both bipolar and MOS "static" RAMs are discussed in this section.

A static RAM (SRAM) uses a flip-flop as the basic memory cell (either bipolar or MOS) and once a bit is stored in a flip-flop, it will remain there as long as power is available to the chip—essentially forever—thus the term "static." On the other hand, the basic memory cell in a "dynamic" RAM (DRAM) utilizes stored charge in conjunction with an MOS device to store a bit of information. Since this stored charge will not remain for long periods of time, it must periodically be recharged (refreshed), and thus the term "dynamic" RAM. Both static and dynamic RAMs are "volatile" memory storage devices, since a loss of power supply voltages means a loss of stored data. In this section we discuss SRAMs in detail that explains how a RAM unit works and how several RAM chips can be combined together to expand the memory capacity, For this purpose we'll use mostly the TTL devices however a brief discussion of MOS based SRAM and DRAM will also be presented.

### The 7489

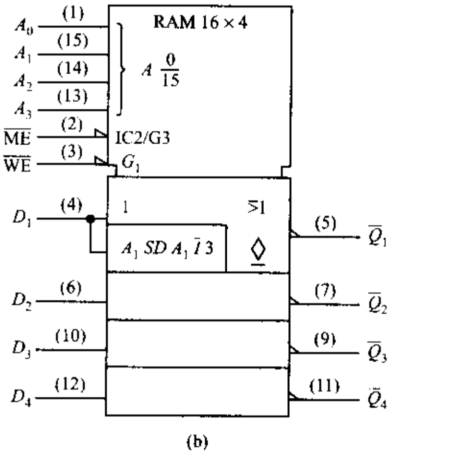
The 7489 shown in Fig. 13.22 is a TTL LSI 64-bit RAM, arranged as 16 words of 4 bits each. Holding the memory-enable ($\overline{ME}$) input low will enable the chip for either a read or a write operation, and the four data address lines will select which one of the sixteen 4-bit word positions to read from or write into. Then, if the write-enable ($\overline{WE}$) is held low, the 4 bits present at the data inputs ($D_1$, $D_2$, $D_3$, $D_4$) will be stored in the selected address. Conversely, if $\overline{WE}$ is high, the data currently stored in the memory address will be presented to the four data output lines ($\overline{Q}_1$, $\overline{Q}_2$, $\overline{Q}_3$, $\overline{Q}_4$). Incidentally, the outputs are open-collector transistors, and a pull-up resistor from each output up to $+V_{CC}$ is normally required. The operations for this chip are summarized in the truth table in Fig. 13.22.



(a)

**Fig. 13.22** **7489, 64-bit RAM**

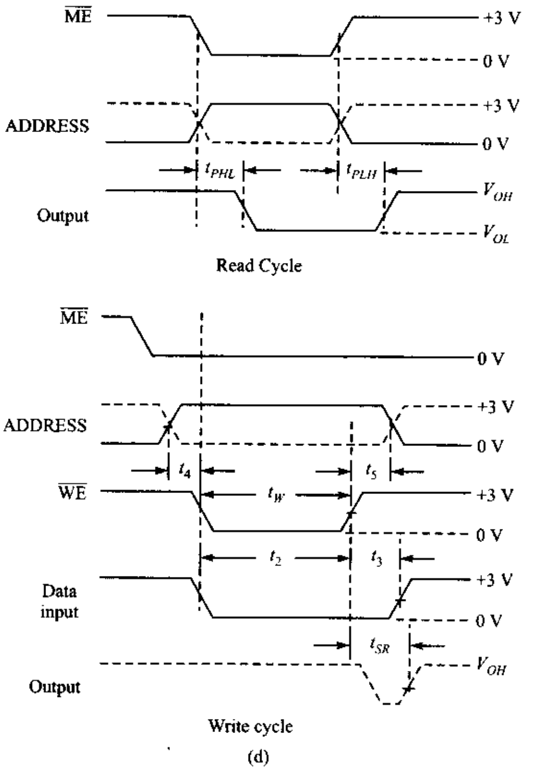
The read operation is no different from that for a ROM. For this chip, simply hold $\overline{\text{ME}}$ low and $\overline{\text{WE}}$ high, and select the desired address. The 4-bit data word then appears at the "sense" outputs. The timing for a read operation is shown by the waveforms in Fig. 13.22d. The propagation delay time $t_{\text{PHL}}$ is that period of time from the fall of $\overline{\text{ME}}$ until stable data appears at the outputs—the data sheet gives a maximum value of 50 ns, with 33 ns typical. Naturally the address input lines must be stable during the entire read operation, beginning



(b)

| $\overline{ME}$ | $\overline{WE}$ | Operation | Condition of outputs |
|---|---|---|---|
| *L* | *L* | Write | Complement of data inputs |
| *L* | *H* | Read | Complement of selected word |
| *H* | *L* | Inhibit storage | Complement of data inputs |
| *H* | *H* | Do nothing | High |

(c)



(d)

Fig. 13.22 (Continued)

with the fall of $\overline{ME}$. Notice carefully that the data appearing at the four outputs will be the *complement* of the stored data word!

You will notice from the truth table that when the chip is *deselected*, that is, when $\overline{ME}$ is high, the outputs all go to a high level, provided we are in a read mode ($\overline{WE}$ is high). So, in the read operation waveforms, the time $t_{PLH}$ is the delay time from the rise of $\overline{ME}$ until the outputs assume the high state. The data sheet gives 50 ns maximum and 26 ns typical for this delay time.

During a write operation the 4 bits present at the data inputs will be stored in the selected memory address by holding the $\overline{ME}$ input low (selecting the chip) and holding the $\overline{WE}$ low. At the same time, the complement of the data present at the four input lines will appear at the four output lines. Timing waveforms for the write operation are also shown in Fig. 13.22d.

Let's look carefully at the timing requirements for the write cycle. First, the $\overline{WE}$ must be held low for a minimum period of time in order to store information in the memory cells—this is given as time $t_W$ on the waveforms, and the data sheet calls for 40 ns minimum. Memory-enable selects the chip when low, and is allowed to go low coincident with or before a write operation is called for by $\overline{WE}$ going low.

Next, the data to be written into memory must be stable at the data inputs for a minimum period of time before $\overline{WE}$ and, also for a minimum period of time after $\overline{WE}$. The time period prior to $\overline{WE}$ is called the *data-setup time* $t_2$. This time is measured from the end of the write-enable signal back to the point where the data must be stable. The data sheet calls for 40 ns, and in this case it is the same as $t_W$. Also, the data inputs must be held stable for a period of time after $\overline{\text{WE}}$ rises—this is called the *data-hold time* $t_3$, and the data sheet calls for 5 ns minimum.

The address lines must also be stable for a period of time before as well as after the $\overline{\text{WE}}$ signal. The time period before $\overline{\text{WE}}$ is called the *address-setup* or *select-setup time* $t_4$. This time is measured from the fall of $\overline{\text{WE}}$ back to the point where the input address lines must be stable; the data sheet calls for 0.0 ns minimum. In other words, the address lines are allowed to become stable coincident with or before $\overline{\text{WE}}$ goes low. The address lines must also be stable for a period of time after the rise of $\overline{\text{WE}}$; this is called the *address-hold* or *select-hold* time $t_5$, and the data sheet calls for 5 ns minimum.

Finally, after a write operation, if the chip is deselected ($\overline{\text{ME}}$ goes high), the outputs will return to a high state. The maximum time for this to occur is the sense-recovery time $t_{SR}$, given as 70 ns maximum on the data sheet.

The operation of a 7489 ns straightforward and easy to understand; therefore it is a good chip to study in elementary discussions of RAMs. It can be used to construct memories having larger capacities by connecting chips in parallel, but it's not too practical when we wish to consider memories of 16K, 32K, ... , 256K, 512K, and so on. Nevertheless, the time spent studying this chip is well invested since the fundamentals of addressing and the read and write operations are essentially the same for all static RAMs. So, with these fundamentals in mind, let's take a look at some chips that have more memory capacity.

## The 74S201

The block diagram in Fig. 13.23 can be used to describe the operation of most SRAMs. Most of these are constructed with $n$ address lines that will uniquely select only one of the $2^n$ cells in the memory array—that is, selection is 1 bit at a time. There will be a chip-enable control (CE), a write-enable (WE), and a provision for a single input data bit ($D_i$) and a single output data bit ($D_o$).

For instance, the 74S201 in Fig. 13.24 is a 256-bit RAM, organized as 256 words, each 1 bit in length. The 256 cells are arranged in a square array of 16 rows and 16 columns. The 8 address bits ($2^8 = 256$) are divided into 4 bits that are decoded to select one of the 16 rows and 4 bits that are decoded to select one of the 16 columns. There is a single input data bit ($D$), a single output data bit ($\overline{Q}$), and a read-write line (R/$\overline{\text{W}}$). There are three memory-enable inputs ($\overline{S}_1$, $\overline{S}_2$, $\overline{S}_3$), and all three of them must be low to select or enable the chip.

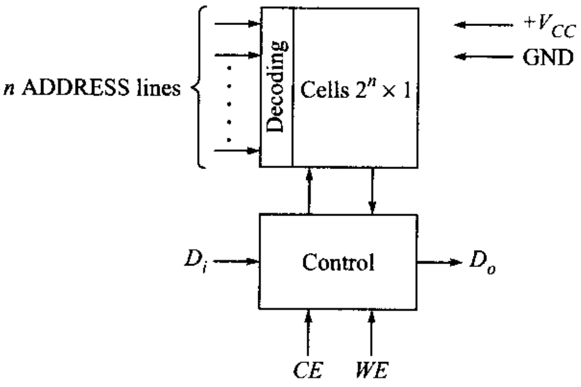
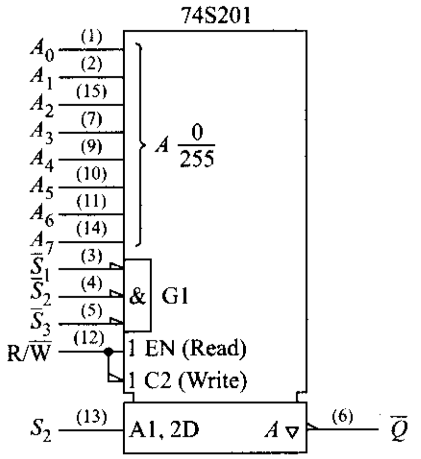


Fig. 13.23 Generalized block diagram for a static RAM

The truth table shows that if the chip is enabled, a write cycle is initiated by holding R/$\overline{\text{W}}$ low, or a read cycle can be initiated by holding R/$\overline{\text{W}}$ high. If any or all of the read-write inputs are high, the chip is inhibited and the output goes to a high-impedance state. Naturally the proper timing must be observed as defined by the timing waveforms; you will see that the timing requirements are very similar to the previously described 7489.



(a) Logic symbol

| | | |
|---|---|---|
| Write-enable pulse width (minimum) | | 65 ns |
| Setup time | ADDRESS before write | 65 ns |
| | Data before end of write | 65 ns |
| | Chip-select before end of write | 65 ns |
| Hold time | ADDRESS after write | 0 ns |
| | Data after write | 0 ns |
| | Chip-select after write | 0 ns |

(b) Recommended timing

Fig. 13.24

**Example 13.5** Using the information in Fig. 13.24, determine how long the address lines for the 74S201 must be held before R/$\overline{W}$ goes low and after R/$\overline{W}$ goes high.

*Solution* **The setup time, address to write-enable, is 0.0 ns. The hold time, address from write-enable, is 0.0 ns. Therefore, the address lines must be stable from the fall of R/$\overline{W}$ until the rise of R/$\overline{W}$.**

## Formation of Memory Banks

*Memory bank* is the concept of increasing memory's capacity by connecting more than one memory block in series, parallel or both.

Now that we understand the operation of the 74S201 (abbreviated as '201), it is a simple matter to use multiple '201 chips to construct larger memories. For instance, we can connect four '201 chips in parallel as shown in Fig. 13.25 to construct a RAM organized as 256 words, each 4 bits in length. Connecting eight '201 chips in parallel will form a memory having 8-bit words, and so on. The nice thing about connecting chips in parallel like this is that the control and timing are exactly the same as if there were only a single chip. The only difference is that there are 4 data bits in and 4 data bits out (or 8 in and 8 out), all of which are in parallel with one another.

As a matter of fact, even larger memories can be constructed by connecting basic chips such as the '201 in both series and parallel. For instance, thirty-two '201 chips are connected in a 4 × 8 matrix in Fig. 13.26 to form a memory having one thousand, twenty-four 8-bit words. This configuration requires a 10-bit address: 2 bits can be used to select one of the four rows of eight '201s, and the remaining 8 bits will be wired in parallel to all the chips; they will work exactly as for the two-hundred fifty-six 4-bit word memory in Fig. 13.25. This concept can be continued, of course, but it becomes somewhat impractical with larger memory requirements, especially since there are MOS chips readily available with greater memory capacity.

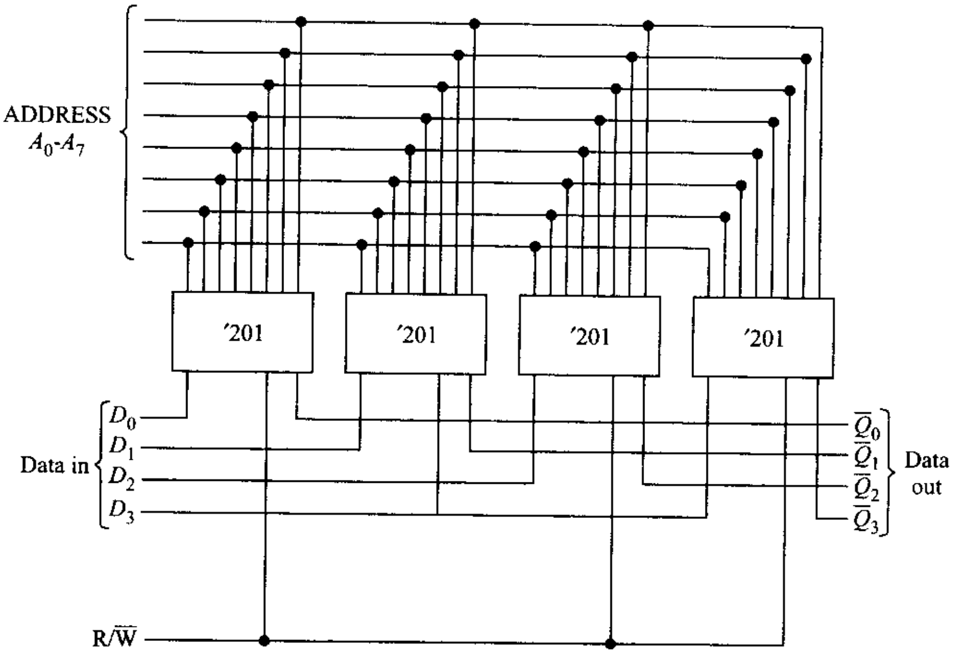


Fig. 13.25 Four 74S201's arranged as a 1024-bit memory having 256

## SRAMs

A very popular and widely used MOS memory chip is the 2114. This is an SRAM having 4096 bits arranged as 1024 words of 4-bits each. The organization of this chip is quite similar to that of the 7489 shown in Fig. 13.22, but notice that the 2114 is sixteen times larger! Nearly all SRAMs larger than 1024 bits are MOS types.

The basic memory is arranged as 64 rows and 64 columns for a total of 64 × 64 = 4096 bits. Six address bits ($A_3$ through $A_8$) are used to select one of the 64 rows ($2^6 = 64$). The 64 columns are divided into 16 groups of 4-bit words, and four address bits ($A_0$, $A_1$, $A_2$ and $A_9$) are used to select one of these 16 groups. A 10-bit address will then select a single 4-bit word from 64 rows and 16 columns, to provide a memory of 64 × 16 = 1024 four-bit words.

A basic SRAM cell or latch is shown in Fig. 13.27a. It consists of a back-to-back inverter that latches on to a particular state of logic 0 or 1. The two pass transistors enable writing and reading from two bit lines. Both the transistors are 'on' for both reading and writing when this cell is selected after decoding the address. The bit lines during writing operations write its value into the latch while during reading sense the latch state. A typical SRAM requires six transistors per bit of memory—two pass transistors and two transistors each of the inverter. However, some implementations use only a single transistor per inverter with a total requirement of four transistors per bit.

## DRAMs

A typical DRAM is essentially the same as the previously discussed SRAM chip, with the exception of the required refresh cycle. The 4116 is a widely used 16K (16,384 × 1) DRAM available from a number of

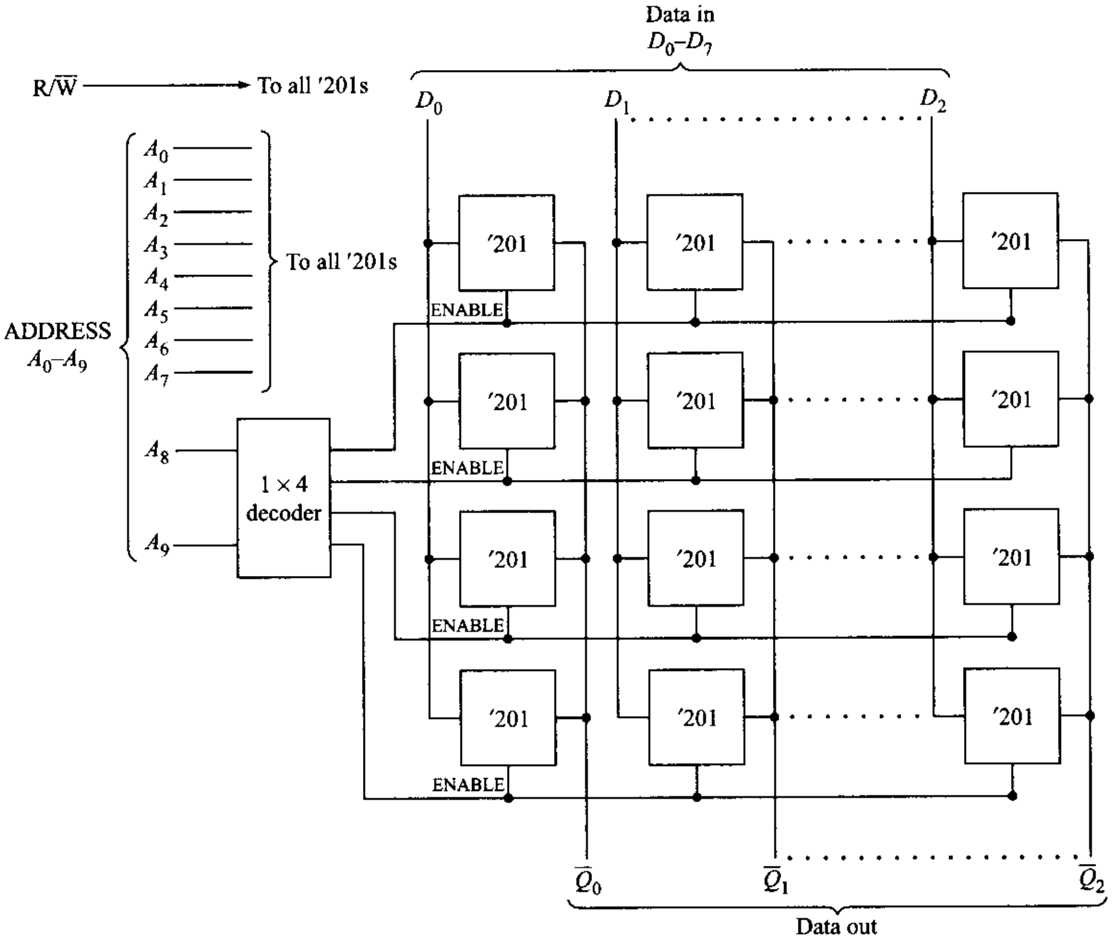


Fig. 13.26

different sources, such as Mostek (MK4116), Motorola (MCM4116), and Texas Instruments (TMS4116). Note that, 4116 must be refreshed at least once in every 2 ms. There is another widely used DRAM, the 4164, organized as a 64,536 × 1 chip. The operation of this chip is quite similar to 4116. The 64K DRAM is available under following part numbers: Texas Instruments TMS4164, Motorola MCM6665, Intel 2164, etc.

A basic DRAM cell is shown in Fig. 13.27b. A capacitor is used as a storage element, as it can store electrical charge but for a limited amount of time. This requires periodic *refresh* to replenish the charge and thus the RAM is always *dynamic*. The cell is selected by turning on the pass transistor. The bit line is used both for writing and reading (sensing). A faster writing ability requires the capacitor to be charged faster which in turn discharges the capacitor quickly requiring quicker refresh cycle. Compared to SRAM, DRAM uses less number of components that require less space, increasing the *packing density*. Also it is much less expensive. But SRAM scores over DRAM on a very important area. DRAM has a very high access time due

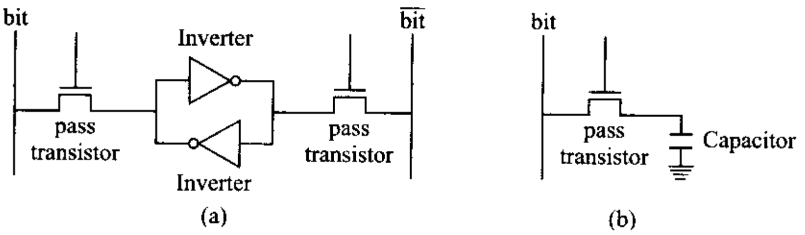


Fig. 13.27 (a) A basic SRAM cell, and (b) A basic DRAM cell

to high latency. This is why even if DRAM is used as a computer's main memory to make it cheap, SRAM is used as a *cache memory* for faster access instruction and data. Cache memory uses *locality* feature where a set of sequential instructions and data are found in contiguous locations in memory. A cache controller brings this memory block from main memory to cache (which is SRAM) for speedier operation of the computer.

## 13.7 SEQUENTIAL PROGRAMMABLE LOGIC DEVICES

We have discussed programmable devices (like PLA, PAL) for combinatorial circuits. In this section, we discuss similar devices available for sequential logic circuits. Architecture-wise, they additionally have memory elements like flip-flops. The simplest of the three widely used variety is called Simple Programmable Logic Devices (SPLD or simply PLD). Similar technology but using larger number of logic gates, suitable to address more complex sequential logic problems is called Complex Programmable Logic Devices (CPLD). The third type uses slightly different technology but of much higher capacity is known as Field Programmable Gate Array (FPGA). The term, High Capacity Programmable Logic Devices (HCPLD) is also used to refer to CPLD and FPGA together. Note that, Hardware Descriptions Languages (HDL), like Verilog, can be used to program these devices.

### PLD

Refer to discussions of Section 4.10 and 4.11 on PAL and PLA and corresponding figures (Fig. 4.44, Fig. 4.47). Note that, output is taken from OR gates following AND plane generating combinatorial logic functions in SOP form. Each OR gate with added circuitry (as shown in Fig. 13.28) that includes a flip-flop, multiplexer and tri-state output forms a *macrocell* of PLD. The flip-flop can store the OR gate output indefinitely and is triggered by a Clock. Multiplexer selects either OR gate or flip-flop output which is also fed back to AND plane for internal use. The output buffer when enabled by *Enable* makes multiplexer output available to external world through output pin, else output pin is held at high impedance state.

Each PLD typically has 8–10 macrocells. Advanced Micro Devices (AMD) manufactured SPLDs 16R8 and 22V10 are PAL based. The name "16R8" means that the PAL has a maximum of 16 inputs

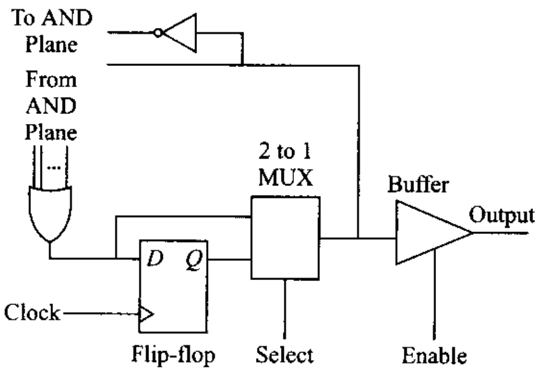


Fig. 13.28 A macrocell of PLD

(there are 8 dedicated inputs and 8 input/outputs which can be configured either as input or output), and a maximum of 8 outputs. The "R" stands for PAL outputs registered as *D* flip-flop. Similarly, the "22V10" has a maximum of 22 inputs and 10 outputs and "V" stands for versatility of the output. The other manufacturers of popular SPLDs are Altera, Lattice, Cypress and Philips-Signetics.

## CPLD

Simple PLDs can handle 10–20 logic equations. Thus, for more complex circuit design one needs to physically connect few such units. This problem is solved by the advent of CPLD, which consists of a number of PLD like blocks (Fig. 13.29). The blocks are interconnected among themselves through programmable switches present in interconnection block. This means it needs two levels of programming: one for programming PLD block the other for programming the switches. Input and output pins of a CPLD chip are routed through I/O blocks. Here, the macrocell has a two input Ex-OR gate after the OR gate. Depending on the value present in the other (Control) input, Ex-OR gate sends complemented or uncomplemented OR output to flip-flop and multiplexer. Commercial CPLDs can have up to 50 PLD blocks. Higher density is not supported by CPLD architecture and FPGAs are used for that. Transistors are used as programmable switches for CPLDs (and also for many SPLDs) by placing it between two wires in a way that facilitates implementation of wired-AND functions. EPROM used as switches does not support in-circuit programming but EEPROM does that. The advantage with them is that both are non-volatile in nature. Re-programmability feature of CPLD is a very useful advantage.

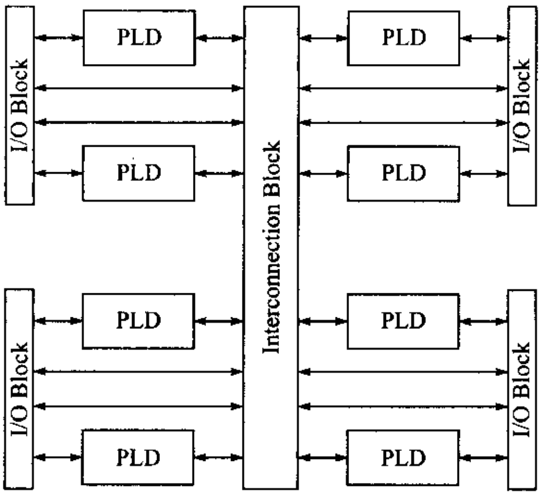


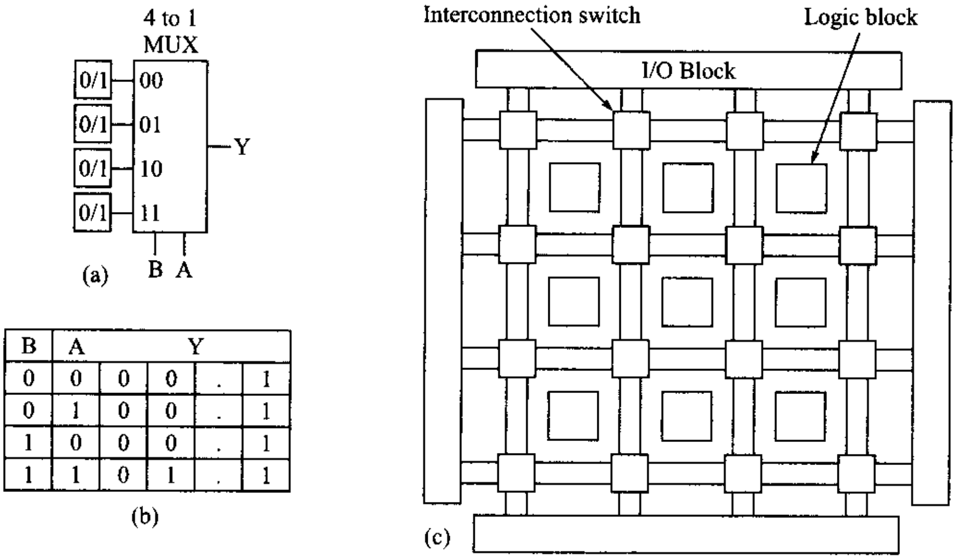
Fig. 13.29 A typical block diagram representation of CPLD

The applications of CPLDs can be found in reasonably complex designs, like graphics controller, UARTs, cache control and many others. Circuits that can exploit wide AND/OR gates, and do not need too many flip-flops are suited for CPLD implementation.

AMD offered CPLD family, Mach 1 to Mach 5 comprises multiple PAL-like blocks: Mach 1 and 2 consist of optimized 22V16 PALs, Mach 3 and 4 comprise several optimized 34V16 PALs and Mach 5 is similar but offers enhanced speed performance. Mach chips are based on EEPROM technology, Xilinx offers XC7000 and XC9500 where each chip consists of a collection of SPLD-like blocks with 9 macrocells in each. Altera has developed three families of chips that fit within the CPLD category: MAX 5000, MAX 7000, and MAX 9000. The other manufacturers of CPLD are Lattice, Cypress, etc.

## FPGA

FPGA consists of an array of circuit elements called logic blocks, which unlike AND-OR combination of CPLD has programmable look up table (LUT). The look up table can generate any logic combination for the variables involved. A multiplexer based 2-variable look up table is shown in Fig. 13.30a. Based on what value (0 or 1) is stored at input this can generate any of the $2^4 = 16$ possible functions of A, B as



| B | A | Y | | | |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | . | 1 |
| 0 | 1 | 0 | 0 | . | 1 |
| 1 | 0 | 0 | 0 | . | 1 |
| 1 | 1 | 0 | 1 | . | 1 |

**Fig. 13.30 (a) A two variable programmable look up unit, (b) Generation of any two variable logic Y = *f* (A, B) by placing appropriate combination as input to 4 to 1 Multiplexer, (c) A typical structure of FPGA**

Y = *f* (A, B). Few examples are shown in the table of Fig. 13.30b. A typical FPGA structure is shown in Fig. 13.30c which is a two dimensional array of logic blocks interconnected by horizontal and vertical wires. The interconnection switches in interconnection blocks are either SRAM or antifuse type. Antifuses are modified CMOS based, normally open circuit but provides low resistance when programmed. Antifuses are not reprogrammable and nonvolatile. SRAM switches are reprogrammable but volatile.

FPGAs have gained rapid acceptance and growth because they can be applied to a very wide range of applications like device controllers, communication encoding and filtering, small to medium sized systems with SRAM blocks and many more. The other important applications of FPGAs are prototyping of designs (later to be implemented in custom made integrated circuits) and also for emulation of large hardware systems.

In Xilinx XC4000 SRAM based FPGA, each configurable logic block (also called CLB) can generate logic functions of up to nine inputs and has two flip-flops. Each of the interconnecting horizontal or vertical channel contains some short wire segments that span a single CLB, longer segments that span two CLBs, and very long segments that span the entire length or width of the chip. In this series, XC4003E has 100 CLBs while XC40250XV has as high as 8464 CLBs. Xilinx also offers XC2000, XC3000, XC5000 and XC8100(antifuse). The other manufacturers providing commercial FPGAs are Altera: FLEX8000 and FLEX10000, Actel: Act1, Act2 and Act3, Quicklogic: pASIC, pASIC2, etc. The SRAM based FPGAs, normally comes with EPROMs that stores bit-streams. This gets loaded every time power is switched on and programs the FPGA.

## 13.8 CONTENT ADDRESSABLE MEMORY

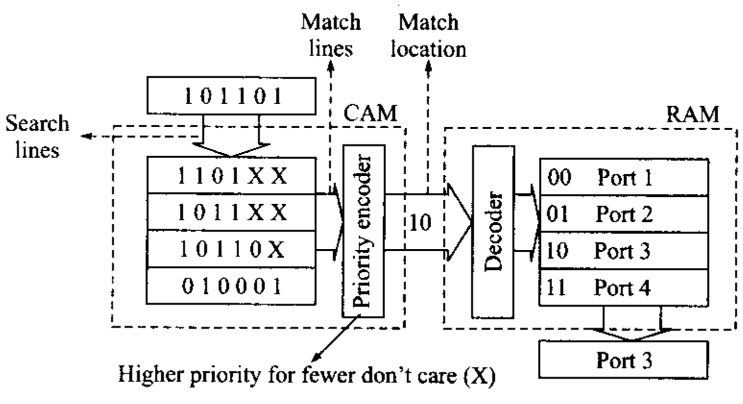
*Content addressable memory* (CAM) uses a completely different kind of addressing scheme from what has been discussed so far. It is designed to be faster to serve specific applications where speed is an issue. Take for example functioning of a network like the Internet. There, a message such as an e-mail or a web page is transferred by first breaking up the message into small data packets of a few hundred bytes, and then, sending each data packet individually through the network. These packets are routed from the source, through the intermediate nodes of the network (called routers), and reassembled at the destination to reproduce the original message. The function of a router is to compare the destination address of a packet to all possible routes and choose the appropriate one. A CAM is a good choice for implementing this lookup operation due to its fast search capability.

CAM compares input search data against a table of stored data, and returns the address of the matching data. Thus, it makes use of the content or data itself to find specific address by implementing a lookup table function using dedicated comparison circuitry. This is to reduce address decoding delays of conventional RAM by making the address meaningful and not an arbitrary one like RAM. A basic CAM cell serves two basic functions—bit storage, like RAM and in addition, bit comparison.

Let us try to understand how a CAM works from a packet forwarding example of a router. Figure 13.31a shows a simple routing table where output port assignment is shown for a range of destination addresses available from the relevant portion of input data stream. This table is a part of the CAM as shown in Fig. 13.31b. When an input data arrives with destination address 101101, the matching of the content occurs for 2$^{nd}$ and 3$^{rd}$ row, i.e. output port 2 and 3 both are eligible to transmit this data. A priority encoder following this lookup table decides which one is to be selected when a match occurs at more than one place. It follows a specific priority scheme. In this example, higher priority is given for the match that has lower number of don't care (X) states. The logic behind this is to keep a port free or available—to the extent possible—that can handle more number of destination addresses. Thus, match location 10 is the output of the priority encoder. The decoder to RAM takes this 10 as the address and selects port 3 as the selected output port.

| Destination address | Output port |
|---|---|
| 1101XX | 1 |
| 1011XX | 2 |
| 10110X | 3 |
| 010001 | 4 |

(a)



(b)

**Fig. 13.31 (a) Routing table, (b) CAM implementing address lookup**

Extending the above example, generally speaking, an input word or *tag* from the incoming data is first stored in a *Search Data Register*. Its content, i.e. the tag, is then broadcasted as *search word* over *search lines*. In a typical CAM, there could be 36 to 144 bits in search lines while the table size could be as high as 32K entries (up to 15 bits of address space). Each stored word has a match line and whenever a match with a word occurs, the corresponding match line is activated. A priority encoder selects a match location based on some priority rule when there are more than one match line in activated state. This effectively reduces a larger space of input search word to a smaller space of output match location. The matching logic corresponding to CAM tags could be *binary* or *ternary*. The binary CAM requires exact match of all the binary locations and returns corresponding match lines while ternary CAM in addition, allows matching with don't care (X) bits. As we have seen in the network routing example, ternary CAM is more useful but also more complex to manage. Often, a CAM comes with a *hit* flag to indicate if there is no matching location in CAM.

## SELF-TEST

15. **What is the organization of the 7489, 64-bit RAM?**
16. **What is the organization of the 2114 SRAM?**
17. **What is a DRAM?**
18. **What is the organization of a 4116 DRAM?**
19. **How is combinatorial logic generated in FPGA?**

## SUMMARY

This entire chapter has been devoted to the study of memories. The use of magnetic and optical memory is discussed first. Next we considered the various rectangular arrays of memory cells on a chip and found that a square array containing the same number of rows and columns requires the fewest number of address lines.

Programmable, erasable-programmable, and plain read-only memories (PROMs, EPROMs, and ROMs) are used to store data in appiications where the data changes not at all, or only infrequently. These memory chips are available as either bipolar or MOS, but the MOS devices offer much greater capacity per chip.

Random-access memories (RAMs) are also available as either bipolar or MOS devices and are used to store data that must be readily available and may be changed frequently. The dynamic random-access memory (DRAM) offers the greater advantage of more storage capacity on a chip but has the disadvantage of requiring refreshing. Careful attention to timing requirements is absolutely essential with the use of any memory chip.

The basic memory cell on a bipolar chip is a simple latch using cross-coupled bipolar junction transistors. The same is true for a static MOS or CMOS memory chip, except that the transistors used are MOS or CMOS, respectively. A dynamic memory, on the other hand, uses a capacitor and one or more MOS transistors to store charge and, therefore, a single bit.

We have not undertaken an exhaustive study of all the memory chips available, but the chips discussed in detail are representative of the most popular ones in present use.

## GLOSSARY

- ***access time*** In general, the delay time measured fron chip-enable (or address) until valid data appears at the output.
- ***address*** Selection of a cell in a memory array for a read or a write operation.
- ***cache*** Small, fast SRAM, a faster memory as an adjunct to slower main memory.
- ***CAM*** Content Addressable Memory.
- ***capacity*** The total number of bits that can be stored in a memory.
- ***chip*** A semiconductor circuit on a single silicon die.
- ***CD-ROM*** Compact Disk Read Only Memory, a kind of movable optical storage media that has higher capacity compared to magnetic counterpart.
- ***CD-R*** Compact Disk Recordable, a kind of optical memory on which data can be written but once.
- ***CD-RW*** Compact Disk Rewritable, a kind of optical memory on which data can be written and erased many times.
- ***DRAM*** Dynamic RAM.
- ***DVD*** Digital Versatile Disk or Digital Video Disk, a very high density optical memory.
- ***dynamic memory*** A memory whose contents must be restored periodically.
- ***EPROM*** An erasable-programmable read-only memory.
- ***EEPROM*** Electrically Erasable Programmable Read Only Memory.
- ***field-programmable*** Referring to a PROM that can be programmed by the user.
- ***flash memory*** A kind of nonvolatile memory which can be written and erased electrically.
- ***Floppy disk*** A movable low capacity magnetic storage media.
- ***Hard disk*** A high capacity magnetic storage media, integral part of modern computer.
- ***mask-programmable*** Referring to a PROM that can be programmed only by the manufacturer.
- ***matrix addressing*** Selection of a single cell in a rectangular array of cells by choosing the intersection of a single row and a single column.
- ***memory bank*** Connects more than one memory block to increase the capacity of the memory.
- ***memory cell*** The circuit used to store a single bit of information in a semiconductor memory chip.
- ***nonvolatile storage*** A method whereby a loss of power will not result in a loss of stored data.
- ***packing density*** Number of memory bits packed in per unit space.
- ***pass transistor*** A MOS transistor that passes information in either direction when it is turned on.
- ***PROM*** Programmable read-only memory.
- ***RAM*** Random-access memory.
- ***read operation*** The act of detecting the contents of a memory.
- ***refresh cycle*** Periodic refresh of DRAM.
- ***ROM*** read-only memory.
- ***static memory*** A memory capable of storing data indefinitely, provided there is no loss of power.
- ***SRAM*** Static RAM.
- ***volatile storage*** A method of storing information whereby a loss of power will result in a loss of the data stored.
- ***write operation*** The act of storing information in a memory.

## PROBLEMS

### Section 13.1

13.1 State the most appropriate memory type to use for each of the following:

a. The working memory in a small computer
b. The memory used to store permanent programs in a small computer
c. A memory used to store development programs in a small computer

13.2 Explain the difference between an EPROM and a PROM.

13.3 Explain the term volatile memory.

13.4 Explain why an EPROM is or is not a volatile memory.

13.5 A memory chip has a read and a write input. Is the chip ROM or RAM?

13.6 What is the difference between a memory cell and a memory word?

13.7 Why is a ROM considered nonvolatile memory?

13.8 What is the difference between an SRAM and a DRAM?

### Section 13.2

13.9 What is the advantage of using a read-write head in magnetic recording systems?

13.10 Look at the code in Fig. 13.6b and determine the proper recording for each of the following:

a. The decimal number 4
b. The letter D
c. The decimal number 8
d. The decimal number 7

13.11 Why is recording on magnetic tape not considered random access?

13.12 A 2400-ft reel of 1/2-in magnetic tape has a data storage density of 6250, 7-bit characters per inch. Assuming no gaps and no lost space, what is the maximum storage capacity of the tape?

13.13 A 2400-ft reel of 1/2-in magnetic tape has a rewind speed of 300 in/s. How much time is required to rewind the tape from mid-position to its beginning? Neglect start and stop times.

### Section 13.3

13.14 Why data integrity of optical memory is better than magnetic memory?

13.15 What is the data transfer rate of a 52X CD-ROM drive?

13.16 Briefly explain Read, Write, Erase process of CD-RW media.

13.17 What is the data transfer rate of 8X DVD-ROM drive?

13.18 Show the different possible rectangular arrangements for a memory that contains 32 memory cells. How many rows and columns for each case?

13.19 How many address lines are required for each case in Prob. 13.14?

13.20 What is the required address $A_4A_3A_2A_1$ to select cell 21 in Fig. 13.10c?

13.21 Determine how many address bits are required for a memory that has the following number of bits:

a. 1024
b. 4098
c. 256
d. 16,384

13.22 A memory chip available from Advanced Micro Devices is the Am9016, advertised as a 16K memory. How many bits of storage are there? How many address lines are required to access one bit at a time?

13.23 What address must be applied to the 74S89 in Fig. 13.20 to select the 4-bit word stored in row 14? Give the address in both binary and hexadecimal.

### Section 13.4

13.24 What is the required address in both binary and hexadecimal to select the 8-bit word in row 27 of the TBP18S030 in Fig. 13.16?

13.25 Show a method for scanning the contents of a TBP18S030 beginning with word 1, then word 2, and so on up to word 32, and then repeating. (***Hint***: Try using a five-flip-flop binary counter for the address *ABCDE*, or use a mod-5 counter with decoding gates, or maybe a shift counter, or …)

13.26 Write a Boolean expression for address row 15 in the TBP18S030 in Fig. 13.16.

13.27 Define the term mask-programmable.

13.28 Draw a set of timing waveforms for a TBP18S030 similar to Fig. 13.17, assuming an access time of 35 ns.

13.29 Redraw Fig. 13.30 and show exactly how to set the switches to program the 8-bit word at address 110 101. Explain exactly what must be done to program the word 1010 0011 at this address. Connecting switch *P* to an output will program a 1 at that cell.

13.30 In a manufacturing process, the pressure (*P*) in a pipe is related to the fluid in the pipe (*F*) according to the relation $P = 3F + 2$. Rather than compute values in real time, it is decided to store precomputed data in a PROM. In this case, *F* has only integer values between 0 and 4, so the computed values are found as shown in the accompanying table. Here's how the data is stored: Each integer value of *F* (0, 1, 2, 3, and 4) represents an address in the PROM. The value of *P* is stored in binary form at the proper address. For instance, when $F = 2$, $P = 1000$ is stored at row address 2.

a. What is the value of *P* when $F = 4$?
b. Draw a PROM having 4-bit words, and show how all data are stored.

| *F* | *P* | *P(binary)* |
|---|---|---|
| 0 | 2 | 0010 |
| 1 | 5 | 0101 |
| 2 | 8 | 1000 |
| 3 | 11 | 1011 |
| 4 | 14 | 1110 |

13.31 Repeat part (b) of Prob. 13.30 if the relation is changed to $P = 2F + 1$.

13.32 Design a ROM to be used as a look-up table for the relation $L = S^2 - 2S + 3$, where $0 \le S \le 6$, and *S* has only integer values.

## Section 13.5

13.33 Show how to connect 7489s in series to construct a memory that has thirty-two 4-bit words.

13.34 Show how to connect 7489s in parallel to construct a memory that contains sixteen 8-bit words.

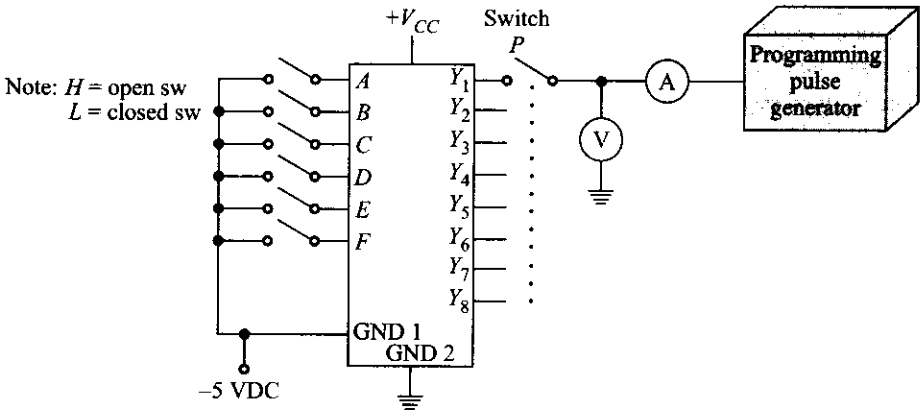


Fig. 13.32

13.35 Design the logic circuits, to provide a read and a write cycle for a 7489.

13.36 Refer to the 74S201 information in Fig. 13.26 and determine the following:

a. Minimum write-enable pulse width

b. Setup time, address to write-enable

c. Hold time, data from write-enable

13.37 Draw the logic diagram for a 256-word 8-bit memory using '201s.

## Answers to Self-tests

1. A DRAM must be refreshed periodically.
2. EPROM stands for erasable-programmable read-only memory.
3. Cache memory is a small high-speed SRAM used inside a computer to speed up operation.
4. Even
5. Tape access time is too long!
6. Binary information is recorded on as magnetic film by magnetizing spots with two different orientations.
7. 780 nm.
8. CD-ROM 25% and more than 70%. CD-RW 15% and 25%.
9. 8.5 GB.
10. 145 (decimal) = 1001 0001 = $A_7A_6A_5A_4A_3A_2A_1A_0$
11. The cell at address 22—row 2 and column 2.
12. It refers to a ROM whose contents are established during the manufacturing process.
13. The triangle is the symbol for a three-state output.
14. It can be corrected by simply programming (adding) a 1 at word position $Q_3$. Note that you can add a 1 by programming (this is destroying a fuse link), but you cannot remove a programmed 1, since this would require replacing a fuse link.
15. Sixteen 4-bit words.
16. 1024, 4-bit words.
17. DRAM stands for dynamic random-access memory.
18. 16,384 × 1 bits
19. Through multiplexer-based look up table.

# Digital Integrated Circuits

# 14

## OBJECTIVES

- **Explain how diodes and transistors can be used as electronic switches**
- **Demonstrate an understanding of TTL devices, their parameters, how to drive them, and how to use them to drive external loads**
- **Be familiar with CMOS-devices and characteristics**
- **Understand TTL-to-CMOS and CMOS-to-TTL interfacing**

In 1964 Texas Instruments introduced transistor-transistor logic (TTL), a widely used family of digital devices. TTL is fast, inexpensive, and easy to use. In this chapter we discuss several types of TTL: standard, high-speed, low-power, Schottky, and low-power Schottky. You will learn about open-collector and tristate devices because these are used to build buses, the backbone of modern computers and digital systems. Since TTL uses active-low as well as active-high signals, negative logic may be used as well as positive logic. Complementary metal-oxide semiconductor (CMOS) devices are chips that combine *p*-channel and *n*-channel MOSFETs in a push-pull arrangement. Because the input current of a MOSFET is much smaller than that of a bipolar transistor, cascaded CMOS devices have very low power dissipation compared with TTL devices. This low dissipation explains why CMOS circuits are used in battery-powered equipment such as pocket calculators, digital wristwatches, and portable computers.

Since a knowledge of the subjects covered here is not prerequisite to any other chapter in this text, the material can be studied in part or in whole, at any time. An understanding of Ohm's law and familiarity with basic dc circuits are the only background needed.