

21. Write a program sequence that will cause an 8237 DMA controller to perform a memory-to-memory transfer of the 1000 bytes beginning at SOURCE to an array beginning at DST. The port addresses of the 8237 are to be 0600 through 060F.
22. Explain exactly how the 8286s in Fig. 9-53 are controlled during an:
  - (a) Output to the control register.
  - (b) Input from the status register.
  - (c) I/O interface to memory transfer.
  - (d) Memory-to-memory transfer.given that the interface is an 8237.
23. Write a program sequence that will cause an 8237 to perform a demand mode block transfer of 500 bytes to an array beginning at BUFFER. The bytes are to be stored starting with the highest address (i.e., BUFFER + 499). The port addresses of the 8237 are to be 1020 through 102F.
24. Give a set of instructions that will:
  - (a) Mask channels 0 and 3 of an 8237.
  - (b) Cause a program-initiated DMA transfer on channel 2 of an 8237 according to the current contents of its control, address, and byte count registers.
25. Draw a timing diagram for an 8237 block mode transfer of 4 bytes from an I/O device to memory.
26. Given that the head load time is 32 ms, seek time is 8 ms per track and there are 40 tracks, and the rotational speed is 360 rpm, determine the average access time of a soft-sectorized diskette. Next determine the average time needed to first recalibrate the head (i.e., move the head to track 0) and then access an arbitrary sector.
27. Write a program sequence that will use the sense drive status command of an 8272 to put a 1 in location FLAG2 if drive 2 is write protected (bit 6 of result is 1).
28. Give a set of instructions that will loop until one of the drives in an 8272-based diskette subsystem is not busy and will then branch to ROUTn, where n is the number of the drive that is first found to be idle.
29. Draw a flowchart of the complete sequence needed to output a block of data to an 8272-based diskette subsystem that includes an 8237 DMA controller. The flowchart should take into account the initialization steps for both the diskette and DMA controllers as well as the steps to execute the phases of write data command.
30. Describe in detail the bus activity that takes place while two consecutive bytes are transferred from memory to an I/O device over a 16-bit 8086 bus by an 8237 DMA controller.
31. Consider the 16-bit parallel I/O device shown in Fig. 9-55. Give a program sequence that will put both groups in both 8255As in mode 0 with ports A and C as inputs and port B as an output. Then write a pair of instructions that will output 1720 through port B.
32. Modify Fig. 9-47 so that the data buffer register and control/status registers are assigned to consecutive even addresses.

# 10

## Semiconductor Memory

In its most general sense the word *memory* refers to any device that stores information for later use. Under this definition, the memory of a computer can be divided into two categories. One category pertains to that part of the computer that holds the instructions and data that are presently being operated on, i.e., the part that the processor can access directly. The other category consists of the facilities that can store information, but the information must be transferred to the memory in the first category before it can be used by the processor. This book has referred to components in the former category as *main memory*, or simply *memory*, and components in the latter category as *mass storage*. This chapter will be concerned with main memory and how it is interfaced to the system bus.

As we have seen, the main memory is comprised of groups of bits called bytes and words that are addressed as units. Until now the processor-oriented meaning of "word" has been used which refers to the numbers of bits that can be simultaneously transferred over the data bus and manipulated as a unit by the processor. Most memory designers, however, use the term "word" to mean the smallest group of bits that is associated with an address. In an 8086/8088 system these groups are the 8-bit bytes. Therefore, when considering memory design in general, "word" will often be used in contexts where "byte" appeared in previous discussions. When a chance of confusion exists, the term "word" will be modified by the word "memory" or the word "processor." The number of bits in a memory word is called its *length* (e.g., the length of an 8086/8088 memory word is 8).

One of the principal attributes of a memory is whether or not it can retain its contents while its power is turned off. A memory that can do so is said to be *nonvolatile*; otherwise, it is said to be *volatile*. Whether or not a memory is volatile is particularly important if it contains instructions, because if it is volatile, these

instructions will be lost during even momentary power failures and will have to be reloaded each time the computer is brought up. For most microcomputer applications, the system must contain at least some nonvolatile memory, but most of the memory can be volatile.

Another important attribute depends on the ways in which the memory can be accessed. If a memory unit can only be read from, it is called *read-only memory (ROM)*, but if it can be both read from and written into, it is called *read/write memory*. *Random access* refers to storage for which all locations can be accessed in the same amount of time, as opposed to *sequential (or serial) access*, in which the locations must be referenced in order (e.g., as on a tape). The main memory of all modern computers is random access; however, when referring to main memory the phrase "random access" has historically been used as a synonym for read/write and read/write memory has been referred to as *random access memory (RAM)*. Thus, the two principal types of main memory are ROM and RAM.

The most prominent kind of nonvolatile read/write memory is *magnetic core*, or simply *core, memory*, which is constructed of small, doughnut-shaped magnets. Volatile read/write memory is made from semiconductor ICs. Although core memory is nonvolatile, it requires a considerable amount of support electronics, which causes it to be relatively expensive. As a result, most microcomputer systems use semiconductor memory and rely on mass storage devices or memory with backup power to store information during shutdown. ROM memory, which is obviously nonvolatile, is also made from semiconductor ICs. Because it does not need to include write circuitry, ROM has the least complicated design. ROM is commonly used to store bootstrap loaders so that the system can be brought up without manually loading initialization instructions. It can, however, be used only to store data and programs that never change. Because core memory is seldom used in microcomputer systems it will not be considered in this book.

This chapter begins with a discussion of the overall design of memory. Sections 10-2 and 10-3 consider the two types of semiconductor read/write memory: static RAM and dynamic RAM. Section 10-4 considers the design of backup power for volatile memory so that it becomes, in effect, nonvolatile memory. The last section examines the various types of ROM.

## 10-1 GENERAL MEMORY ORGANIZATION

The memory of a computer system normally consists of one or more PC boards that are connected to the system bus. On each board is a module that is addressed by the high-order bits on the address bus. As shown in Fig. 10-1, most systems include both ROM and RAM modules. It should be pointed out, however, that some small systems which are used as simple controllers may contain only ROM and/or may include the memory on the same board, or even the same IC chip, as the processor.

The general design of a memory module is shown in Fig. 10-2. It consists primarily of an interface and an array of memory IC devices, each of which contains

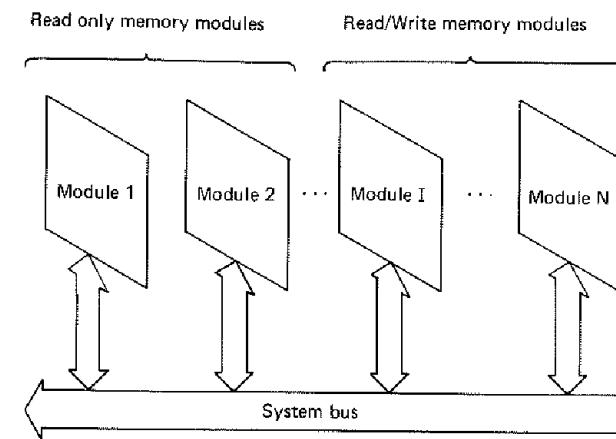


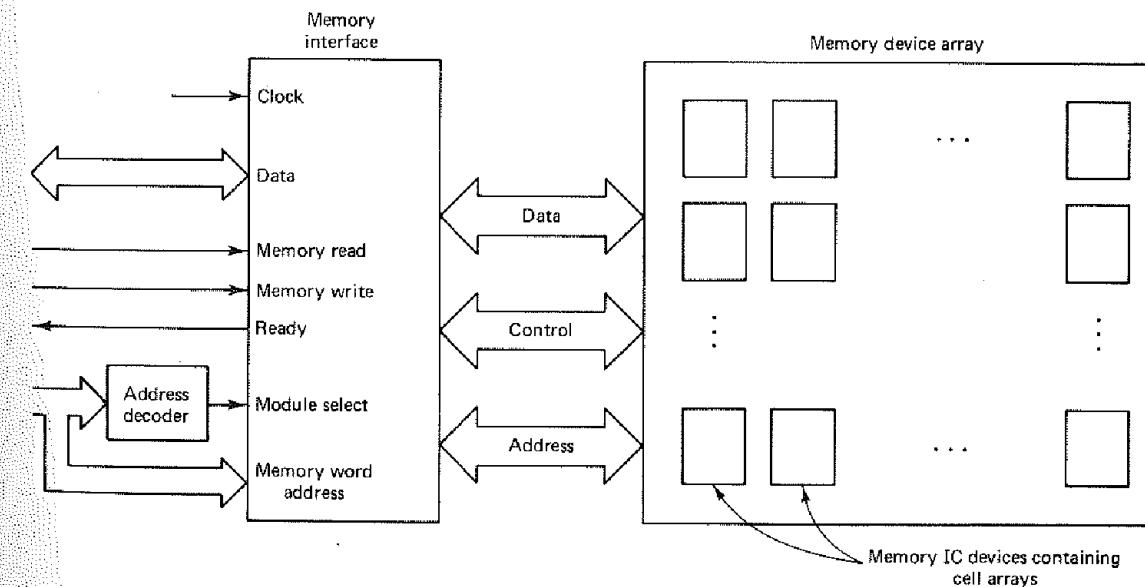
Figure 10-1 Overall organization of memory. Glenn A. Gibson, Yu-Cheng Liu, MICROCOMPUTERS FOR ENGINEERS AND SCIENTISTS © 1980, Prentice-Hall, Inc.

an array of memory cells with each cell being the circuitry needed to store 1 bit. The cells in a device may be accessed separately or in groups, but either way the following relations must hold:

$$\frac{\text{No. of devices}}{\text{in a row}} = \frac{\text{No. of bits in a word}}{\text{No. of cells in a group}}$$

$$\frac{\text{No. of devices}}{\text{in a column}} = \frac{\text{No. of words in module}}{\text{No. of groups in a device}}$$

Figure 10-2 Typical memory module design.



A memory device is said to be an  $M \times N$  device if it contains  $M$  groups with each group consisting of  $N$  cells. Similarly, a  $K \times L$  module is one that consists of  $K$  words, each having  $L$  bits. To demonstrate the application of the above relations, several memory module designs which utilize typical memory devices are summarized in Fig. 10-3.

The principal criteria involved in designing a memory are:

1. Cost.
2. Capacity.
3. Speed.
4. Power consumption.
5. Reliability.
6. Volatility and access capability.

The cost of a memory module normally consists of two components, one which is independent of the size of the module and is called *overhead*, and one which is proportional to the size and is called the *incremental cost*. The overhead cost is due primarily to the support and interface electronics in the module, and the incremental cost is most closely related to the cost of the memory devices. Both overhead and incremental costs are dependent on the number of pin connections and the complexity of the board design. Therefore, memory devices with large bit capacities and that require few supporting chips ordinarily have a cost advantage. Because the overhead tends to be the same regardless of the capacity of the module but must be included for each module, it is generally better to fulfill the overall

**Figure 10-3** Representative memory device arrays. Glenn A. Gibson, Yu-Cheng Liu, MICROCOMPUTERS FOR ENGINEERS AND SCIENTISTS © 1980, Prentice-Hall, Inc.

Memory Size	Type of RAM	No. of chips in each column	No. of chips in each row	No. of chips in the module
4K x 8	1K x 1	4	8	32
	4K x 1	1	8	8
	256 x 4	16	2	32
	1K x 4	4	2	8
4K x 16	1K x 1	4	16	64
	4K x 1	1	16	16
	256 x 4	16	4	64
	1K x 4	4	4	16
16K x 8	1K x 4	16	2	32
	4K x 1	4	8	32
	8K x 1	2	8	16
	16K x 1	1	8	8
64K x 8	16K x 1	4	8	32
	64K x 1	1	8	8

memory capacity requirement using as few modules as possible. Another factor which enters into the overhead is the cost of the power supply. The fewer the number of supply voltages that are required, the less complicated the power supply and board design.

The speed of a memory is gauged by its *access time*, which is defined as the time delay between the receipt of a stable address input and the data output. This time depends on many factors and is even related to the bit capacity of the memory devices. High-speed transistors tend to require more chip space so that fewer cells can be included on each chip. Also, high-speed devices, which are normally made using a bipolar technology, are more expensive and consume more power.

Power consumption can be very important in systems that must sometimes be powered by batteries or solar cells (e.g., space vehicles). The technology used in constructing the IC determines the power required for each cell. The technology most often used in low-power applications is the complementary metal-oxide semiconductor (CMOS) technology. The main disadvantage associated with CMOS is that a fair amount of chip space is needed for each cell; thus the capacity of each device is reduced. Unfortunately, power consumption and speed tend to be proportional and it is difficult and expensive to achieve both low power and high speed. The technology that has been found to be a good compromise of speed, power consumption, capacity per device, and cost is the high-density MOS (HMOS) technology.

Because the reliability of an IC is high once it has been thoroughly tested, the reliability of a module is strongly related to the number of solder connections and board complexity. Therefore, reliability increases as the total pin count decreases, thus supplying an additional incentive to minimize the number of memory and supporting devices in a module.

Nonvolatility and access capability are pretty much dictated by the application. Unless an application requires nonvolatility there is no reason to provide it. If nonvolatile RAM is needed, then either core memory must be used or backup power must be available so that the volatile semiconductor memory can be maintained in a powered-up state. ROM is generally used whenever possible because it is less expensive, nonvolatile, more reliable, and impervious to noise, and its simple construction lends itself to high cell density.

## 10-2 STATIC RAM DEVICES

For static memory devices, a cell is commonly implemented using six MOS transistors, as shown in Fig. 10-4. Information is stored according to the states of transistors  $Q_1$  and  $Q_2$ . This cross-coupled transistor pair is such that when one transistor is on, the other is off, and vice versa. A 1 is assigned to the state that exists when  $Q_2$  is on and  $Q_1$  is off, and a 0 is assigned to the opposite state. The transistors  $Q_3$  and  $Q_4$  serve as resistors and  $Q_5$  and  $Q_6$  act as enable gates. During a write operation, first the cell is selected by raising the voltage level on the select line. When this is done, transistors  $Q_5$  and  $Q_6$  act as short circuits, so that the

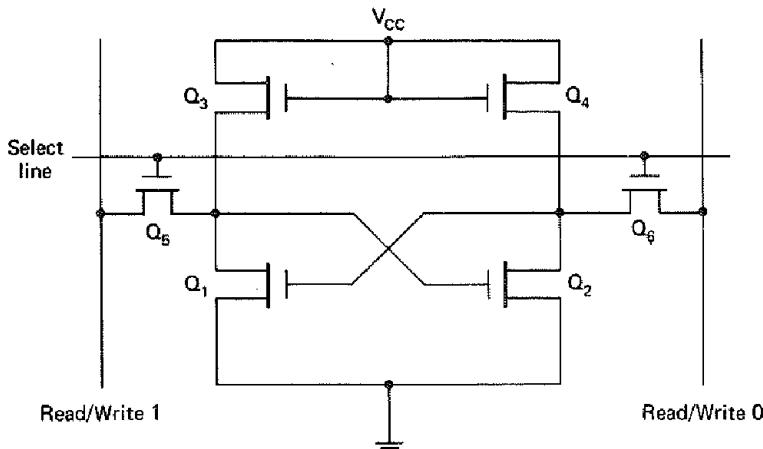


Figure 10-4 Schematic diagram of a six-transistor static RAM cell. Glenn A. Gibson, Yu-Cheng Liu, MICROCOMPUTERS FOR ENGINEERS AND SCIENTISTS © 1980, Prentice-Hall, Inc.

Read/Write 1 line is applied to the gate of  $Q_2$  and the Read/Write 0 line is applied to the gate of  $Q_1$ . To write a 1 into the cell, a 1 is placed on Read/Write 1 and a 0 is placed on Read/Write 0; this causes  $Q_2$  to be turned on and  $Q_1$  to be turned off. On the other hand, if a 0 is to be written into the cell, a 1 is placed on Read/Write 0 and a 0 is placed on Read/Write 1. In either case, once they are set the states of  $Q_1$  and  $Q_2$  will remain unchanged until the next write operation. The cell can be read simply by applying a voltage to the select line. When this is done, the state of  $Q_1$  is applied to Read/Write 0 and the state of  $Q_2$  is applied to Read/Write 1.

The number of memory cells and their arrangement in a static memory device varies considerably. Common sizes range from  $256 \times 4$  to  $16K \times 1$ . A  $256 \times 4$  RAM contains 256 locations, each having 4 bits, while a  $16K \times 1$  configuration provides 16K locations, each containing only 1 bit. The general organization of a static RAM is described by the block diagram of Fig. 10-5, which illustrates a  $1K \times 1$  RAM. The memory cells are organized in a matrix of 32 rows and 32 columns. Address bits  $A_9$  through  $A_0$  are divided into row and column addresses and together specify one of the 1024 cells. Row address inputs ( $A_4$ – $A_0$ ) are decoded to select one of the 32 rows of storage cells. Column address inputs ( $A_9$ – $A_5$ ) not only select the column, but also enable the corresponding I/O circuits consisting of drivers and sense amplifiers. These circuits permit a stored bit to be output during a read operation and to be changed during a write operation. The read/write (R/W) control input specifies the type of operation, high for read and low for write. The chip enable input (CE) is used to select the appropriate row of memory devices in the memory module.

A  $4K \times 8$  memory device array which is constructed from  $1K \times 1$  devices is shown in Fig. 10-6. If a chip is enabled, a read or write operation will proceed as specified by the R/W control input. Otherwise, the read/write signal will not be recognized and the output is forced into a high-impedance state. This allows the

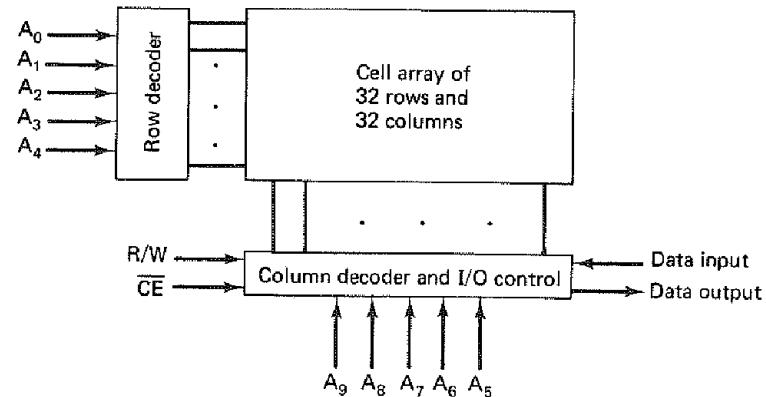


Figure 10-5 Block diagram of a  $1K \times 1$  memory device. Glenn A. Gibson, Yu-Cheng Liu, MICROCOMPUTERS FOR ENGINEERS AND SCIENTISTS © 1980, Prentice-Hall, Inc.

data outputs of several memory chips to be directly tied together so that  $2K \times 8$ ,  $4K \times 8$ , and so on, memories can be constructed from columns of  $1K \times 1$  chips with each column contributing one bit of the data byte. When this is done, the bit being output not only depends on the signals on the address lines, but also depends on which chips receive the chip enable signal. Each row in the array is connected to a row enable line and the row enable lines are activated by higher-order address bits, which for this example are bits  $A_{11}$  and  $A_{10}$ . When a row is selected each device in the row will input or output a bit according to the signals on the  $A_9$ – $A_0$  lines. In summary, if the address contains 16 bits,  $A_{15}$ – $A_{12}$  would select the module,  $A_{11}$  and  $A_{10}$  would select the row, and  $A_9$ – $A_0$  would select the bits in the devices which constitute the addressed byte.

Because each cell in a static RAM device must contain several transistors, each device generally contains fewer cells than a dynamic RAM chip of comparable design. Also, they tend to use more power since one of the two cross-coupled transistors is always on, thus consuming power continuously. The major advantage of static RAM is that it does not need to be refreshed.

As with most other digital devices, it is easy to connect semiconductor memory devices together because they have built-in supporting electronics. However, the timing constraints of the input signals are critical and the timing requirements vary from one device to another. To ensure proper operation, the control logic on a memory board must provide address inputs and control signals that satisfy the timing parameters as specified by the manufacturer of the memory devices being used. The input timing during a memory read operation is different from the input timing during a memory write operation.

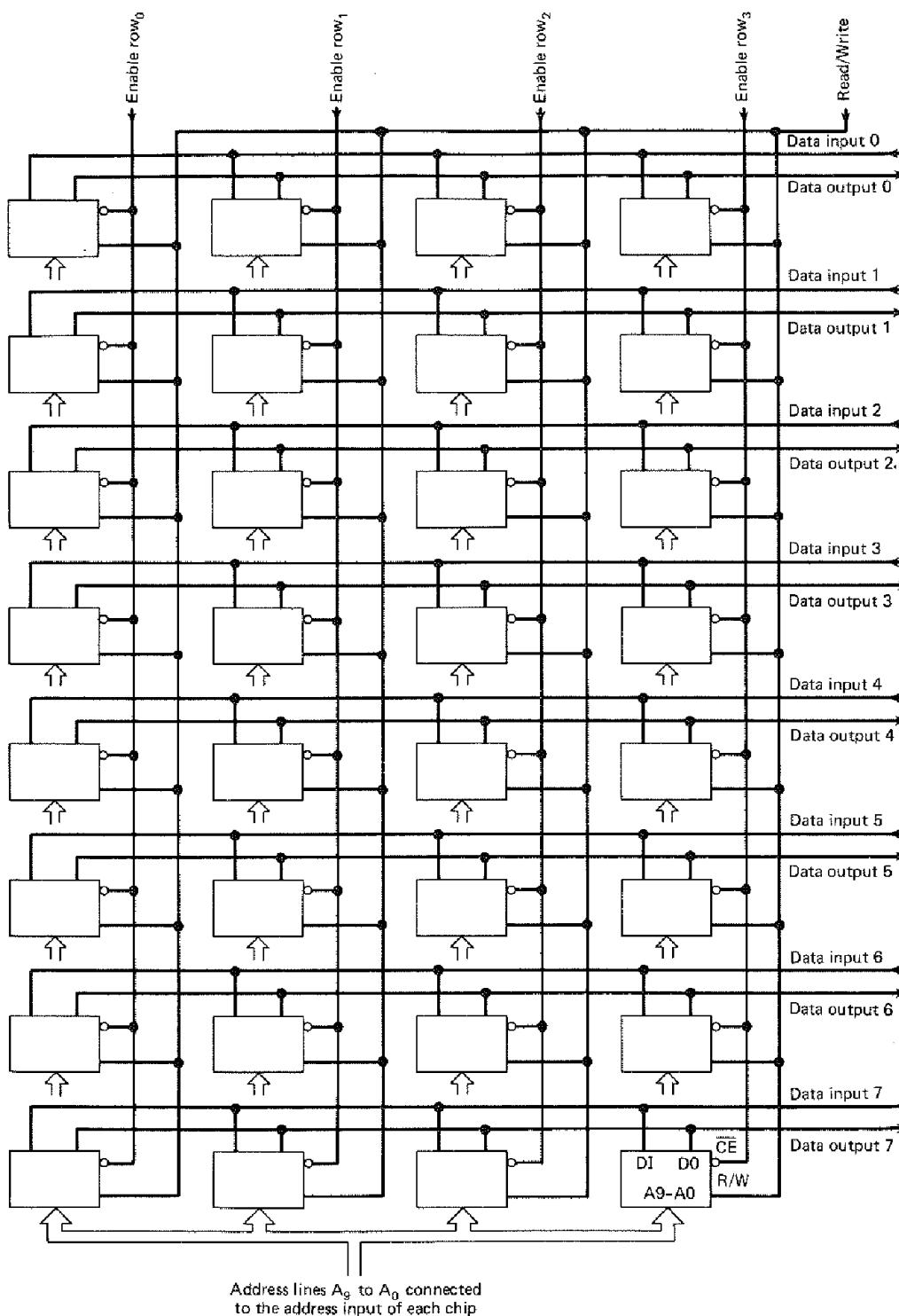
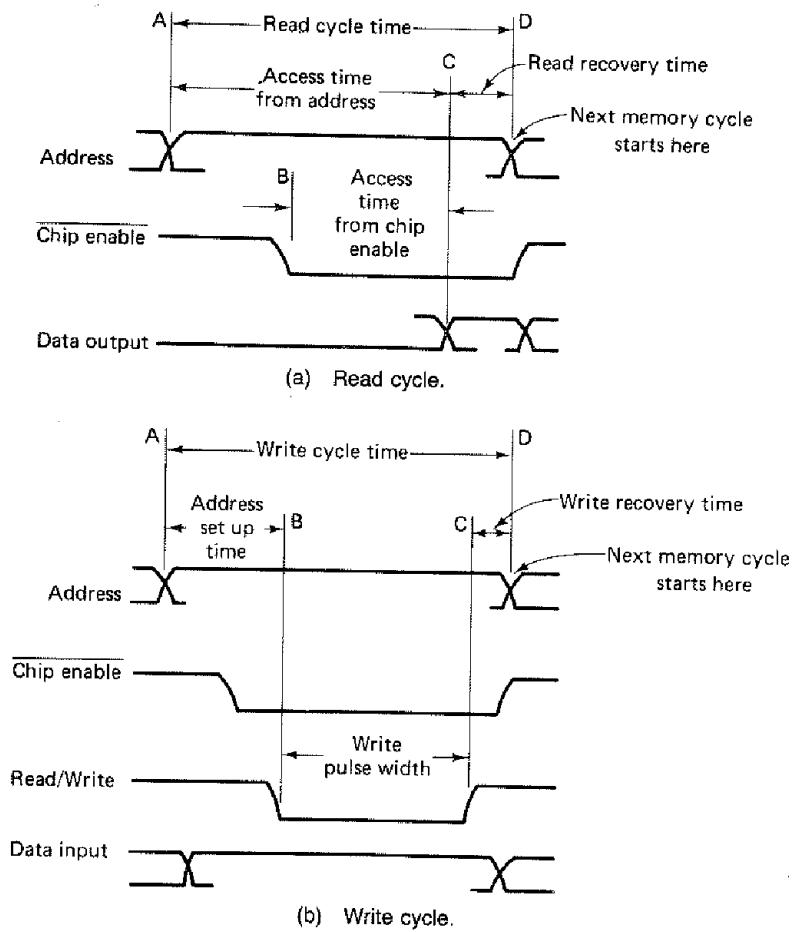
The most important timing parameter to be considered in choosing a memory device is the *access time*. The maximum time delay from an address input to a data output is longer than the delay between a chip enable and a data output, and consequently the former timing figure is normally considered to be the access time. The access time for commonly used MOS RAMs varies from 50 to 500 ns.

For a read operation, once the output data are valid, the address input cannot

## Sec. 10-2 Static RAM Devices

be changed immediately to start another read operation. This is because the device needs a certain amount of time, called *read recovery time*, to complete its internal operations before the next memory operation. The sum of the access time and read recovery time is the *memory read cycle time*. This is the time needed between the start of a read operation and the start of the next memory cycle. The *memory write cycle time* can be similarly defined and may be different from the read cycle time. Figure 10-7(a) illustrates the timing of a memory read cycle. The address is applied at point A, which is the beginning of the read cycle, and must be held stable during the entire cycle. In order to reduce the access time, the chip enable input should be applied before point B. The data output becomes valid after point C and remains valid as long as the address and chip enable inputs hold. The R/W control input is not shown in the timing diagram for the read cycle, but should remain high throughout the entire cycle.

**Figure 10-7** Timing for memory reference cycles. Glenn A. Gibson, Yu-Cheng Liu, MICROCOMPUTERS FOR ENGINEERS AND SCIENTISTS © 1980, Prentice-Hall, Inc.



**Figure 10-6** 4K × 8 memory device array constructed from 1K × 1 devices. Glenn A. Gibson, Yu-Cheng Liu, MICROCOMPUTERS FOR ENGINEERS AND SCIENTISTS © 1980, Prentice-Hall, Inc.

A typical write cycle is shown in Fig. 10-7(b). In addition to the address and chip enable inputs, an active low write pulse on the R/W line and the data to be stored must be applied during the write cycle. The timing of data input is less restrictive and can be satisfied simply by holding the data input stable during the entire cycle. However, the application of the write pulse has two critical timing parameters: the address setup time and the write pulse width. The *address setup time* is the time required for the address to stabilize and is the time that must elapse before the write pulse can be applied. In Fig. 10-7(b) the address setup time is the time interval between points A and B. The *write pulse width* defines the amount of time that the write input must remain active low. The *write cycle time* is the time interval between points A and D and is the sum of address setup time, write pulse width, and write recovery time. Both the read and write recovery times may be zero for some memory devices.

It is important to note that the access time and cycle time discussed in this section are the minimum timing requirements for the memory devices themselves. The access time and cycle time for the memory system as a whole are considerably longer because of the delays resulting from the I/O control logic, system bus logic, and memory interface logic.

An example of the design of a  $16K \times 8$  static RAM memory module for a maximum mode 8088 is shown in Fig. 10-8. It is assumed that the  $\overline{CE}$  and  $\overline{WE}$  inputs, and the D7-D0 pins of the  $4K \times 8$  static RAM, have the following relationships:

$\overline{CE}$	$\overline{WE}$	Function	D7-D0
1	X	Not selected	High-impedance state
0	1	Read	Data put on lines
0	0	Write	High impedance and data are taken from lines

where 1 is high and 0 is low. When a memory device is not selected (i.e.,  $\overline{CE}$  is high) it will go into a standby state which permits it to operate at reduced power.

The incoming address bus splits into two parts, with lines A19-A14 being used to select the module. A13 and A12 are input to the chip enable logic, which is detailed in Fig. 10-9(a). The chip enable logic has four outputs,  $\overline{CE}_0$  through  $\overline{CE}_3$ , only one of which may be active at any one time.  $\overline{CE}_0$  is connected to the Chip Enable ( $\overline{CE}$ ) input of the memory device associated with the lowest 4K addresses and is active when  $A_{13} = A_{12} = 0$ . Similarly,  $\overline{CE}_1$  is active when  $A_{13} = 0$  and  $A_{12} = 1$ ,  $\overline{CE}_2$  is active when  $A_{13} = 1$  and  $A_{12} = 0$ , and  $\overline{CE}_3$  is active when  $A_{13} = A_{12} = 1$ . In all cases the module select line must be 1 before a  $\overline{CE}$  output can be active. The module select line is active only when the module is selected and a memory read or write is recognized. The address lines A11-A0 are applied to the A11-A0 pins of all of the memory devices.

$\overline{MWTC}$  and the module select line are input to a write pulse generator which is constructed from two monostable multivibrators and is detailed in Fig. 10-9(b). The output of this circuit is connected to the Write Enable ( $\overline{WE}$ ) pins of all the memory devices and causes the data on D7-D0 to be put into the addressed byte.

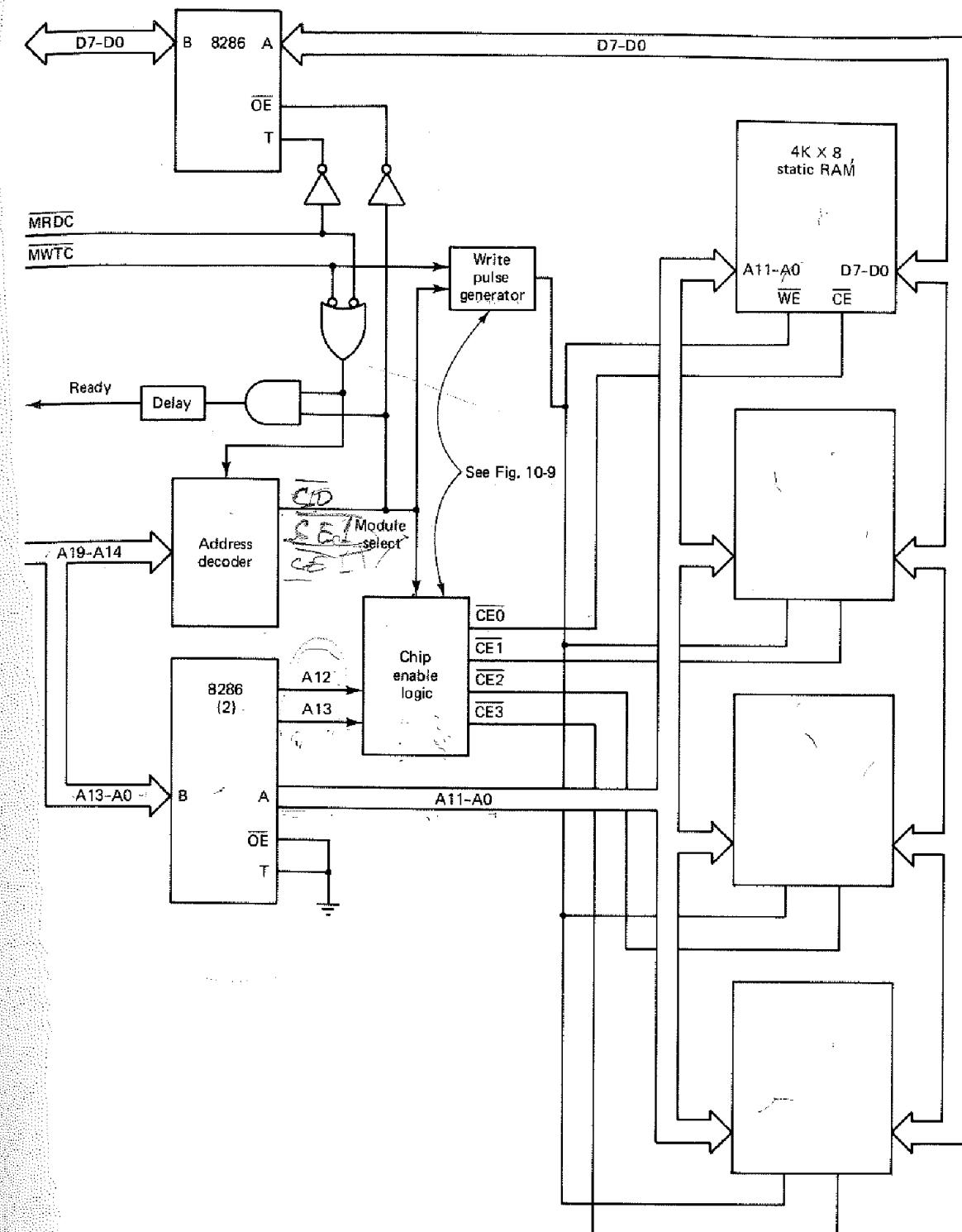


Figure 10-8 16K  $\times$  8 memory module for a maximum mode 8088.

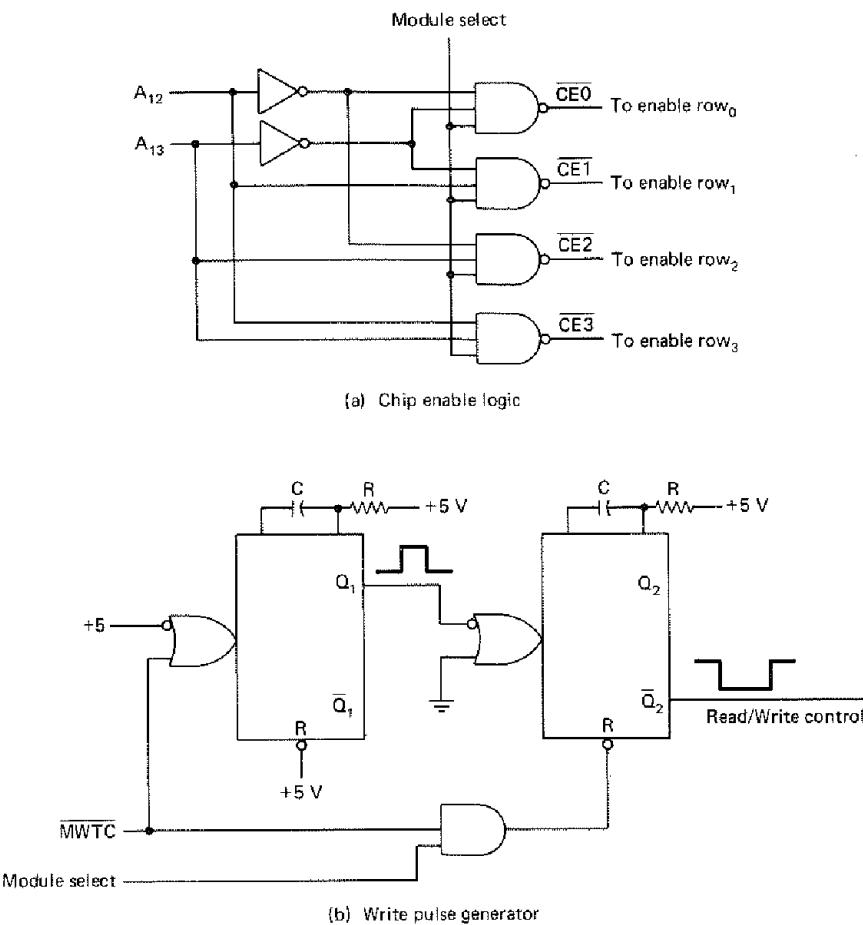


Figure 10-9 Support logic for the memory module design in Fig. 10-8. Glenn A. Gibson, Yu-Cheng Liu, MICROCOMPUTERS FOR ENGINEERS AND SCIENTISTS © 1980, Prentice-Hall, Inc.

Note that in the device array shown in Fig. 10-6 there are separate data-in and data-out lines emerging from each device, but in Fig. 10-8 the data lines are bidirectional. Internally, there must be separate lines for reading from and writing to the cells, but it is possible to separate the bidirectional signals into read and write signals inside the device.

### 10-3 DYNAMIC RAM DEVICES

Just like static RAMs, the memory on a dynamic memory chip is organized in a matrix formed by rows and columns of memory cells. The simplest type of dynamic RAM cell contains only one transistor and one capacitor, as shown in Fig. 10-10. Whether a 1 or 0 is contained in a cell is determined by whether or not there is a

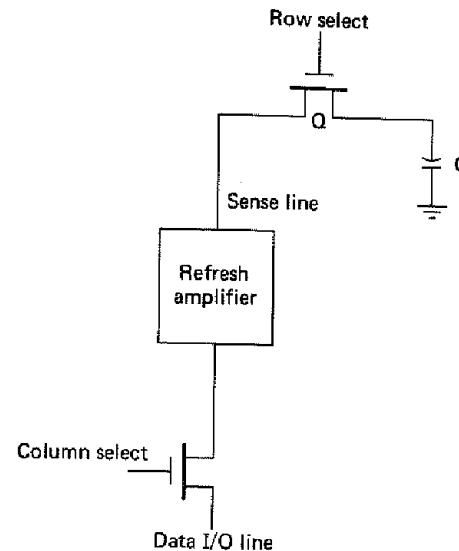


Figure 10-10 Typical one-transistor dynamic RAM cell. Glenn A. Gibson, Yu-Cheng Liu, MICROCOMPUTERS FOR ENGINEERS AND SCIENTISTS © 1980, Prentice-Hall, Inc.

charge on the capacitor. During a read operation, one of the row select lines is brought high by decoding the row address (low-order address bits). The activated row select line turns on the switch transistor Q for all cells in the selected row. This causes the refresh amplifier associated with each column to sense the voltage level on the corresponding capacitor and interpret it as a 0 or a 1. The column address (high-order address bits) enables one cell in the selected row for the output. During this process, the capacitors in an entire row are disturbed. In order to retain the stored information, the same row of cells is rewritten by the refresh amplifiers. A write operation is done similarly except that the data input is stored in the selected cell while the other cells in the same row are simply refreshed.

As a result of the storage discharge through pn-junction leakage current, dynamic memory cells must be repeatedly read and restored; this process is called *memory refresh*. The storage discharge rate increases as the operating temperature rises, and the necessary time interval between refreshes ranges from 1 to 100 ms. When operating at 70°C, the typical refresh time interval is 2 ms. Although a row of cells is refreshed during a read or write, the randomness of memory references cannot guarantee that every word in a memory module is refreshed within the 2-ms time limit. A systematic way of accomplishing a memory refresh is through memory refresh cycles.

In a *memory refresh cycle*, a row address is sent to the memory chips, and a read operation is performed to refresh the selected row of cells. However, a refresh cycle differs from a regular memory read cycle in the following respects:

1. The address input to the memory chips does not come from the address bus. Instead, the row address is supplied by a binary counter called the *refresh address counter*. This counter is incremented by one for each memory refresh

cycle so that it sequences through all the row addresses. The column address is not involved because all elements in a row are refreshed simultaneously.

2. During a memory refresh cycle, all memory chips are enabled so that memory refresh is performed on every chip in the memory module simultaneously. This reduces the number of refresh cycles. In a regular read cycle, at most one row of memory chips is enabled.
3. In addition to the chip enable control input, normally a dynamic RAM has a data output enable control. These two control inputs are combined internally so that the data output is forced to its high-impedance mode unless both control inputs are activated. During a memory refresh cycle, the data output enable control is deactivated. This is necessary because all the chips in the same column are selected and their data outputs are tied together. On the other hand, during a regular memory read cycle, only one row of chips is selected; consequently, the data output enable signal to each row is activated.

Consider a memory module of 16K bytes implemented by  $4K \times 1$  dynamic RAMs. The memory device array consists of four rows and eight columns. Each chip has 64 rows and 64 columns of memory cells and has separate row address (6 bits) and column address (6 bits) pins. It is assumed that the chip enable and output enable pins are CE and CS, respectively. The block diagram in Fig. 10-11 shows the logic needed to generate the chip enable and the refresh address signals during a memory refresh. The memory refresh cycle signal, which indicates that a refresh is in progress, can be generated by a refresh cycle timing generator in the memory module. If the current cycle is a refresh cycle, the 2-to-1 multiplexer selects the row address from the refresh address counter; otherwise, the row address comes from the address bus. Assuming that every cell in the chip array must be refreshed within a 2-ms time interval, a refresh cycle is required for every  $2 \times 10^{-3}/64 = 31.25 \mu s$ . At the end of each refresh cycle, the binary counter is incremented by 1 so that it points to the next row to be refreshed. During a refresh cycle, all chips on the module are enabled for performing a read operation by activating the chip enable signals. Their data outputs are forced into the high-impedance state by deactivating the output enable signals.

In addition to requiring refresh logic, a chief disadvantage in using dynamic RAMs is that during a refresh cycle the memory module cannot initiate a read or write cycle until the refresh cycle has been completed. As a result, a read or write request will take up to twice as long to complete if a refresh is in progress. Assuming a cycle time of 400 ns (for all memory cycles—refresh, read, write),

$$\frac{64 \times 400 \times 10^{-9}}{2 \times 10^{-3}} \times 100\% = 1.28\%$$

of the memory time is spent on refresh.

There are several reasons why dynamic RAMs are attractive to memory designers, especially when the memory is large. Three of the main reasons are:

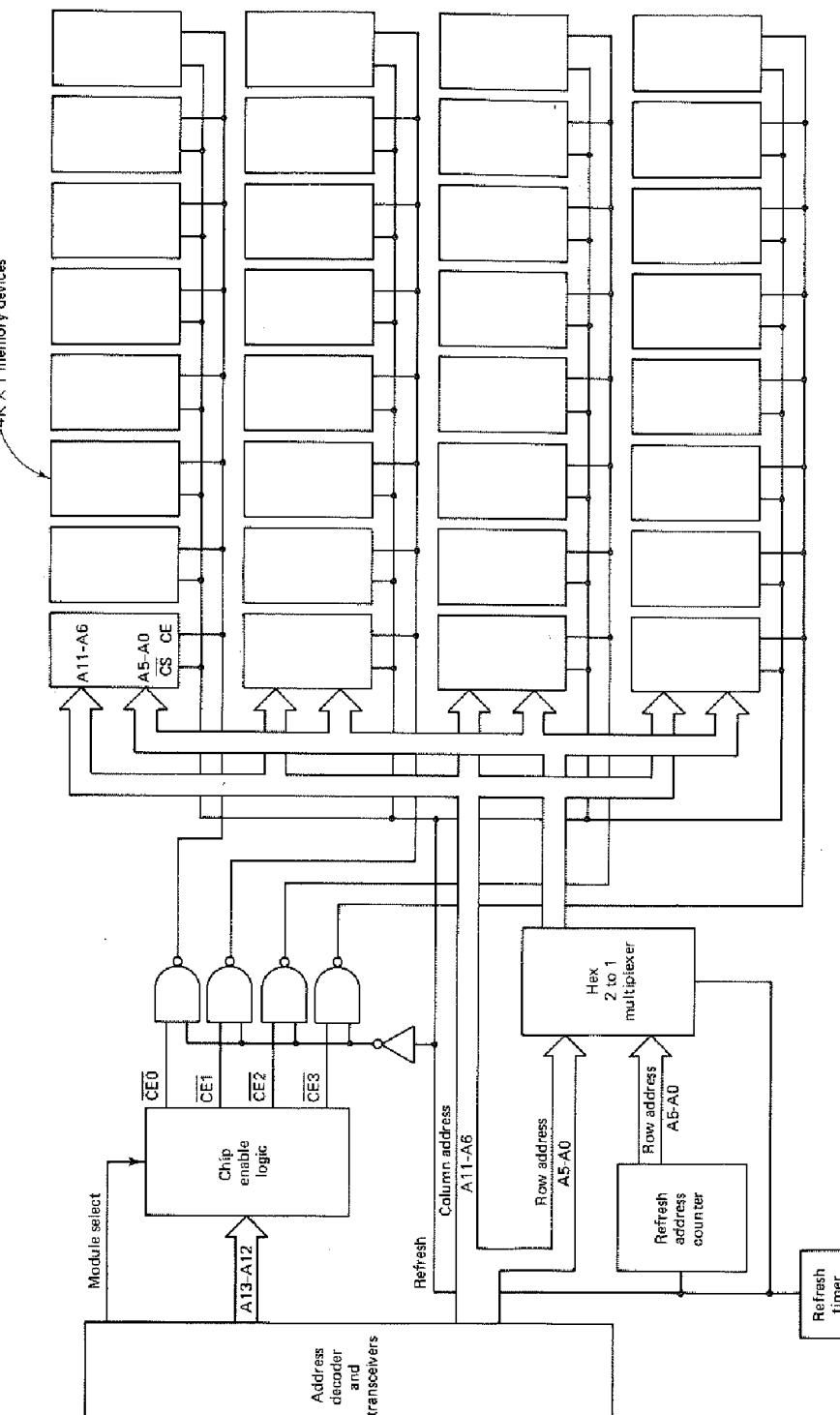


Figure 10-11 Refresh address logic for a 16K-byte RAM implemented using  $4K \times 1$  dynamic RAM devices.

- High Density**—For static RAM, a typical cell requires six MOS transistors. The structure of a dynamic cell is much simpler and can be implemented with three, or even one MOS transistor. As a result, more memory cells can be put into a single chip and the number of memory chips needed to implement a memory module is reduced. A common size for a dynamic RAM chip is  $16K \times 1$ , and  $64K \times 1$  devices are also available.
- Low Power Consumption**—The power consumption per bit of dynamic RAM is considerably lower than that of static RAM. The power dissipation is less than  $0.05\text{ mW}$  per bit for dynamic RAM and typically  $0.2\text{ mW}$  per bit for static RAM. This feature reduces the system power requirements and lowers the cost. In addition, the power consumption of dynamic RAM is extremely low in standby mode; this makes it very attractive in the design of memory that is made nonvolatile through the use of a backup power source.
- Economy**—Dynamic RAM is less expensive per bit than static RAM. However, dynamic RAM requires more supporting circuitry, and, therefore, there is little or no economic advantage when building a small memory system.

For higher-density RAM, the row address and column address normally share the same set of pins, thus reducing the device's pin count. For these RAMs some manufacturers have produced single ICs that include the refresh support logic and other logic needed in controlling the row/column address pins. Toward this end, Intel has made available its 8203 dynamic RAM controller, which is specifically designed to support its 2117, 2118, and 2164 dynamic RAM memory devices. Here we will concentrate on the 8203's use with the 2164, which is a  $64K \times 1$  device. The block diagrams for the 2164 and 8203 are shown in Fig. 10-12. (The 8203 can be put in one of two modes according to its  $16K/64K$  pin; only the assignments for the  $64K$  mode are shown.) The 2164 has four  $128 \times 128$  cell arrays but only eight address pins, A7–A0. This means that the row address and the column address must share the same pins and be received one after the other. The row address is strobed by a negative-going pulse on the RAS pin and, with RAS still low, the column address is strobed by a negative-going pulse on the CAS pin. The most significant row and column address bits specify one of the four cell arrays. During a memory refresh cycle, the address input A7 is not used and all four cell arrays perform their refreshes simultaneously. This allows the entire device to be refreshed in 128 cycles.

The timing diagrams for the read, write, and refresh-only cycles are shown in Fig. 10-13. For a read cycle, WE must be inactive before the CAS pulse is applied and remain inactive until the CAS pulse is over. After the column address is strobed, RAS is raised and with RAS high and CAS low the data bit is made available on DOUT. For a write cycle the DIN signal should be applied by the time CAS goes low, but after the WE pin goes low. The write is performed through the DIN pin while RAS, CAS, and WE are all low. The DOUT pin is held at its high-impedance state throughout the write cycle. For the refresh-only cycle only the row address is strobed and the CAS pin is held inactive. The DOUT pin is kept in its high-impedance state.

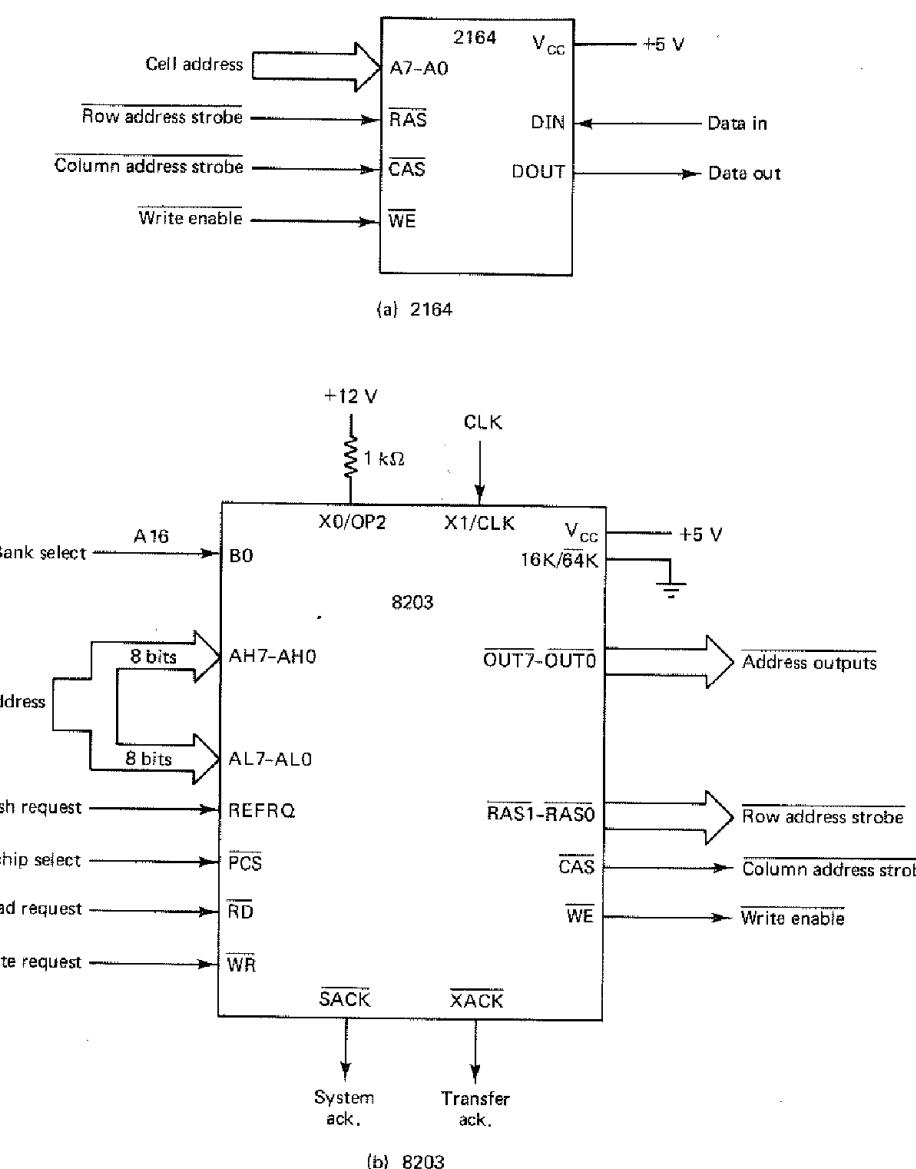


Figure 10-12 Pin assignments for the 2164 dynamic RAM and its associated 8203 controller.

The 8203 is designed to output signals whose timing meets the 2164 requirements. The OUT7–OUT0 lines provide the properly sequenced row and column addresses, RAS1–RAS0 provide the row address strobes for up to two banks of 2164s, and CAS and WE supply the column address strobe and write enable signals for all of the 2164s in the module. (Note that the addresses output by the 8203 are

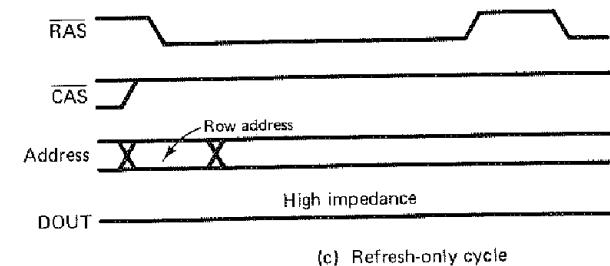
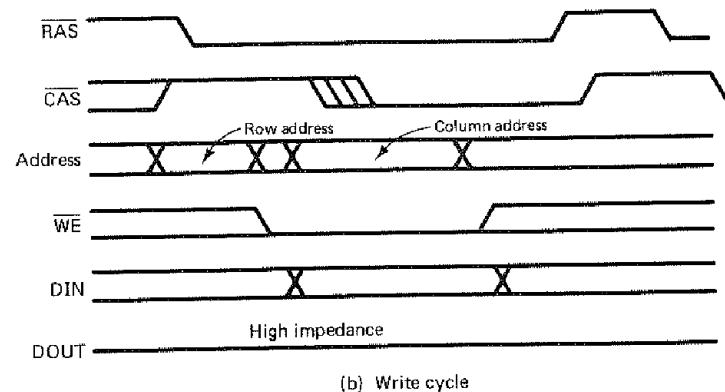
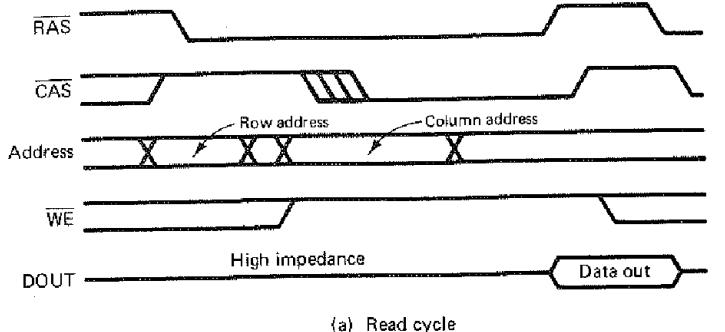


Figure 10-13 Timing diagrams for the 2164.

inverted. This does not cause a problem, but it does mean that all 0s on the address lines will access the cells having row and column addresses that are all 1s.)

The bank select input B0 determines the RAS pin to be activated. AL7–AL0 are used to generate the row address and AH7–AH0 the column address. Normally, the refresh cycle timing is generated inside the 8203, but the REFRO pin allows the refresh cycles to be initiated by an external source. The module selection is made through the PCS pin. It is called the protected chip select pin because once

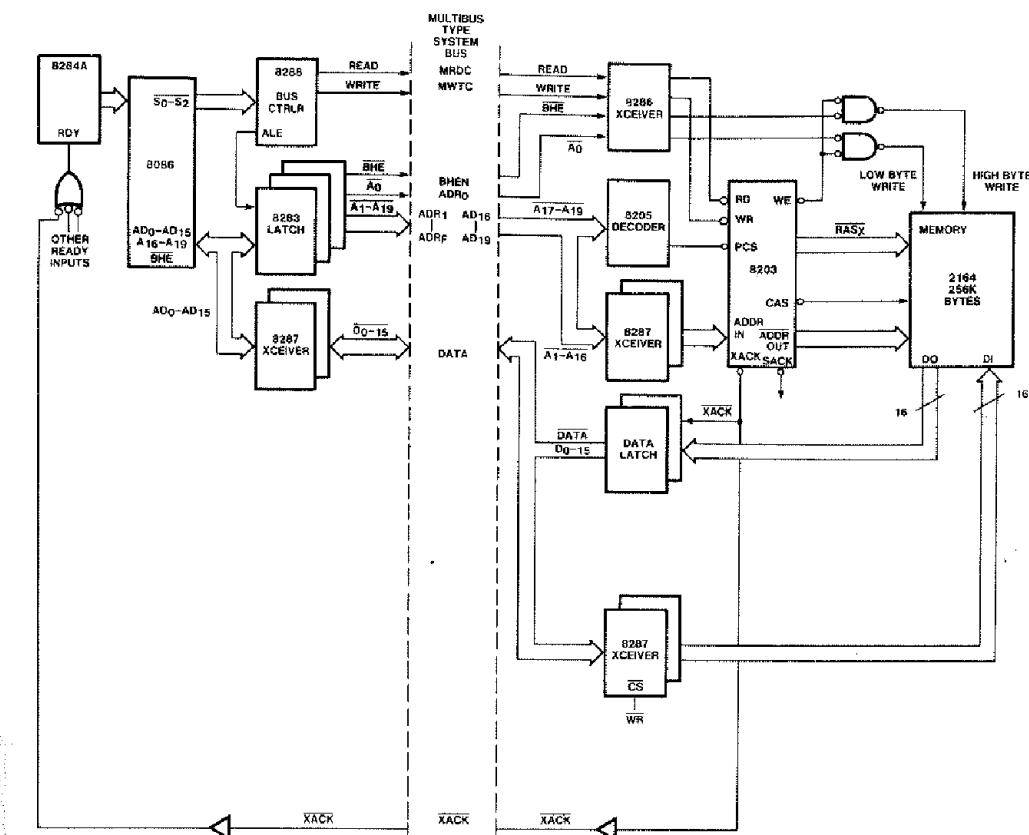


Figure 10-14 Maximum mode 8086, MULTIBUS system with a 256K-byte memory module. (Reprinted by permission of Intel Corporation. Copyright 1982.)

it becomes active the memory cycle cannot be aborted, even if it immediately returns to its inactive state. The RD and WR inputs specify whether a memory read or write is to be conducted.

The XACK output is a strobe indicating that data are available during a read cycle or that the data have been written during a write cycle. It can be used to strobe data into data output latches and to send the ready signal to the processor. The SACK output signals the beginning of a memory access cycle and if a refresh cycle is taking place when a memory request is made, the SACK signal is delayed until the read or write cycle begins. If the memory device access time is known to be sufficiently low to guarantee that a read will be complete by the end of the T<sub>3</sub> clock cycle or a write will be complete by the end of the T<sub>4</sub> cycle, the SACK output

may be used as the ready signal instead of the  $\overline{XACK}$  output, thus saving wait states that might occur if  $\overline{XACK}$  is used.

Either an oscillator must be connected across X0 and X1 or, if OP2 is connected to +12 V, an external clock signal must be applied to CLK. This signal may come from a bus clock line or a clock included in the memory module. Only +5 V is needed for the main power supply, but if the OP2 input is used, a +12-V voltage is required. (REFRQ is actually a dual-purpose pin that can be used for advanced reads, but this feature as well as the read-modify-write cycle will not be considered here.)

Figure 10-14 illustrates how an 8203 and thirty-two 2164s could be used to construct a 256K-byte memory module for a maximum mode 8086 MULTIBUS system. This is an Intel design which assumes that the data and address buses are inverted and therefore uses 8283s and 8287s to interface to these buses instead of 8282s and 8286s. The memory device array has 16 columns so that words can be accessed. This requires that the BHE and A0 bus lines be used in conjunction with the WE signal to determine whether only the low-order byte, only the high-order byte, or an entire word is being written. (Figure 10-14 was taken from Reference 1, which also includes, on page 9-88, the design of a minimum mode 8088 system with an 8203/2118-based 64K-byte memory module.)

Another way of reducing the number of chips needed in the support circuitry for dynamic RAM is to put a set of refresh logic on each memory device, thus permitting the device to refresh itself. Such a device is called an integrated RAM and, except for memory accesses sometimes being held up by refresh cycles, the device appears to the user to be a static RAM. An example of this approach is the Intel 2186/7, which is an 8K  $\times$  8 integrated RAM. The 2186/7 has pin assignments that are essentially the same as for Intel's static RAM devices. In particular, it has  $\overline{OE}$ , WE, and  $\overline{CE}$  pins which serve the same purposes.

#### 10-4 BACKUP POWER FOR SEMICONDUCTOR MEMORIES

One major disadvantage of using MOS RAMs to construct main memory is that the stored information may be lost as a result of even very short power failures. The solution is to provide a backup power supply which will support the system if the main supply fails. Because of the cost, normally only part of the memory system is protected from power failure. During a power failure, the status and vital data within the program being executed can be stored in the nonvolatile memory modules; then by restoring this information after the power has returned, the program is able to continue.

Some MOS RAMs consume much less power while they are just maintaining information than when they are executing read/write operations. To reduce the drain on the backup power supply during a power failure, the memory module can be forced into a standby mode in which all memory chips are disabled and the

stored data are simply retained. Because of this feature, batteries become a practical way to provide backup power for MOS memory over reasonable lengths of time.

Figure 10-15 shows a power supply with a battery backup. During normal operation, the power is provided by a power supply which converts an ac line voltage into a regulated dc voltage that is maintained at  $V_{CC}$ . The output of the battery is lower than the normal  $V_{CC}$ , and consequently diode D1 is forward-biased and diode D2 is cut off. If the main power fails, the capacitor discharges until its voltage is lower than the output of the battery. At this time, D2 is forward-biased and the battery supplies the power to the memory. Once the main supply is restored, D2 is cut off and the battery is recharged. Another circuit that is commonly used to switch power between the main supply and the battery is one that includes a relay. The relay is such that power comes from the main supply during normal operation and from the battery during power failure. The relay is controlled by a power-loss-detection circuit.

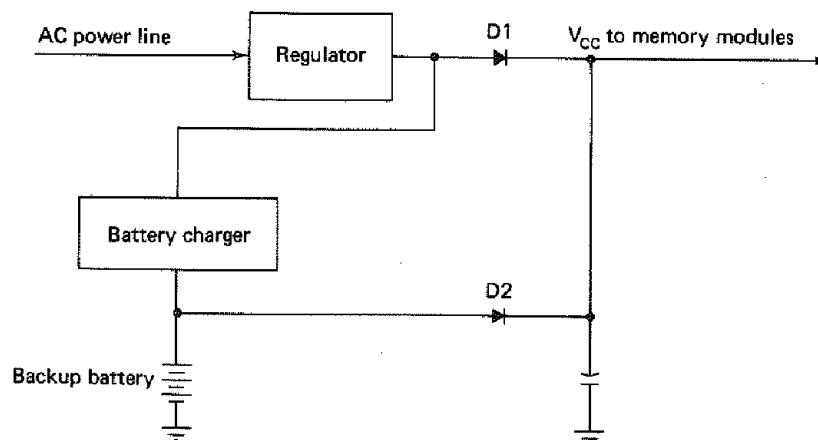
The type and numbers of batteries required in the backup supply are determined by the following factors:

1. The supply current required by the memory modules.
2. The battery discharge characteristics.
3. The size, weight, and cost of the batteries.
4. The maximum length of time the memory must be supplied by backup batteries.

Because a memory module consists of a memory chip array and supporting logic, the total discharge current requirement can be calculated by

$$\text{Discharge current} = \frac{N_m \times P_m + P_s}{V}$$

Figure 10-15 Battery backup power system. Glenn A. Gibson, Yu-Cheng Liu, MICROCOMPUTERS FOR ENGINEERS AND SCIENTISTS © 1980, Prentice-Hall, Inc.



where

$N_m$  = number of memory chips

$P_m$  = power dissipated by each memory chip

$P_s$  = total power dissipated by the supporting logic

$V$  = supply voltage

The required supply current is significantly less if the memory is forced into standby mode during a power failure. Another way to reduce standby power is by using low-power-dissipation CMOS devices for the memory interface and control logic.

The capacity of a battery is rated in terms of ampere-hours at a specific discharge current. However, the ampere-hour rating tends to decrease as the discharge current rating increases. The protection time that can be provided by a backup power source is the ratio of the ampere-hour rating to the discharge current, provided that the discharge current is less than or equal to its rating. For example, assume that a battery has a capacity of 3.2 A·h at a 1-A discharge current and that a memory module draws 0.8 A. Then the battery can supply the power to the memory module for at least 4 h. Three batteries in parallel will be able to provide protection for at least 12 h.

A desirable feature for a backup battery is a nearly constant output voltage during discharge. Batteries that satisfy this criterion are commercially available in both the rechargeable and nonrechargeable types. Examples of the nonrechargeable type are mercury and silver oxide batteries, which offer large capacity, yet small size. Nickel-cadmium and lead-calcium batteries are widely used rechargeable batteries. Although they are considerably larger and heavier than most batteries, the fact that they can be recharged whenever main power is restored may be required in some applications.

## 10-5 ROM DEVICES

ROM is such that once its contents are set they cannot be changed, at least not without the aid of special equipment. Its cell density can be higher than that of read/write memory because its construction is simpler (there is no need for write circuitry). It is also nonvolatile and reliable. However, it is restricted to uses where writing is not necessary. As noted in the introduction to this chapter, most memory systems contain both ROM and RAM modules, with the ROM modules being used to store such things as bootstrap loaders and permanent data tables. ROM is also sometimes used to hold the resident monitor and other suitable parts of the operating system (thus obviating the need for a bootstrap loader). It may even contain permanently installed language interpreters.

There are basically four types of ROMs, the four types being distinguished by the way in which their contents are set. Setting the contents of a ROM is sometimes called programming, but should not be confused with the generation of

instruction sequences discussed in the previous chapters. In one type of ROM the contents are determined by a masking operation that is performed while the chip is being manufactured. Such chips cannot be altered by the user and are referred to simply as ROMs. The contents of the second type can, if the proper equipment is available, be set by the user. They are called *programmable-read-only memories* (PROMs). As with the masked ROMs, once a memory of this type is programmed, its contents can never be changed. The third and fourth types can not only be programmed by the user, but also by using special equipment can be erased and reprogrammed many times. They are called *erasable-programmable-read-only memories* (EPROMs) and *electrically-erasable-read-only memories* ( $E^2ROMs$ ).

Because a masked ROM requires the manufacturer to first produce a mask according to a given set of contents, or *program*, it is expensive to produce the first ROM, but additional copies are inexpensive. Therefore, ROMs are used only after the development stage of a computer system is complete, when it is not likely that the contents of the ROMs will need to be changed. The only pins required on a ROM device are a set of address input pins, a set of data output pins, and a chip select pin. Figure 10-16 shows how two  $2K \times 8$  ROMs could be used to build a  $4K \times 8$  byte storage module. In addition to the logic shown, the usual module select address decoder, transceivers, and latches would be required.

Normally, PROMs are constructed of diode matrices. They are programmed by using external pins to select the diode links that are to be "burned" or "blown," thus causing the diode matrix to be permanently programmed. The 1s are represented by the diode links left intact and the 0s by the diode links that are blown. Because the PROMs for all customers are identical when they are shipped, their price is not as dependent on quantity, but their circuitry is more complex and expensive due to their programmability. Therefore, for a few devices PROMs are cheaper than masked ROMs, but for large-quantity orders masked ROMs are cheaper. As a result, PROMs (or EPROMs) are used during development, but masked ROMs are used in the final mass-produced products.

PROMs are programmed bit by bit by placing the address of the location containing the bit on the address pins and forcing a current *into* the bit's data output pin while the voltage supply and appropriate control pins are pulsed. The magnitude of the current, the magnitudes and width of the pulses, and the manner in which the pulses are applied varies from one PROM to the next. Generally, the programming cycles are interspersed with verification cycles and the cycles are alternated for approximately twice the time needed to program the bit.

As opposed to a PROM, in which the contents are permanently programmed by "burning" out diode links, the contents of an EPROM are determined by charge distribution. EPROMs are programmed by charge injection, and once programmed the charge distribution is maintained until it is disturbed by some external energy source such as an ultraviolet light. Instead of completely encasing an EPROM chip in a protective package as is done with other ICs, an EPROM is constructed by putting a quartz window over its circuitry, which will allow the external energy to pass. By exposing the memory to the external energy source for several minutes (10 to 50 minutes, depending on the specific device), the charge is redistributed

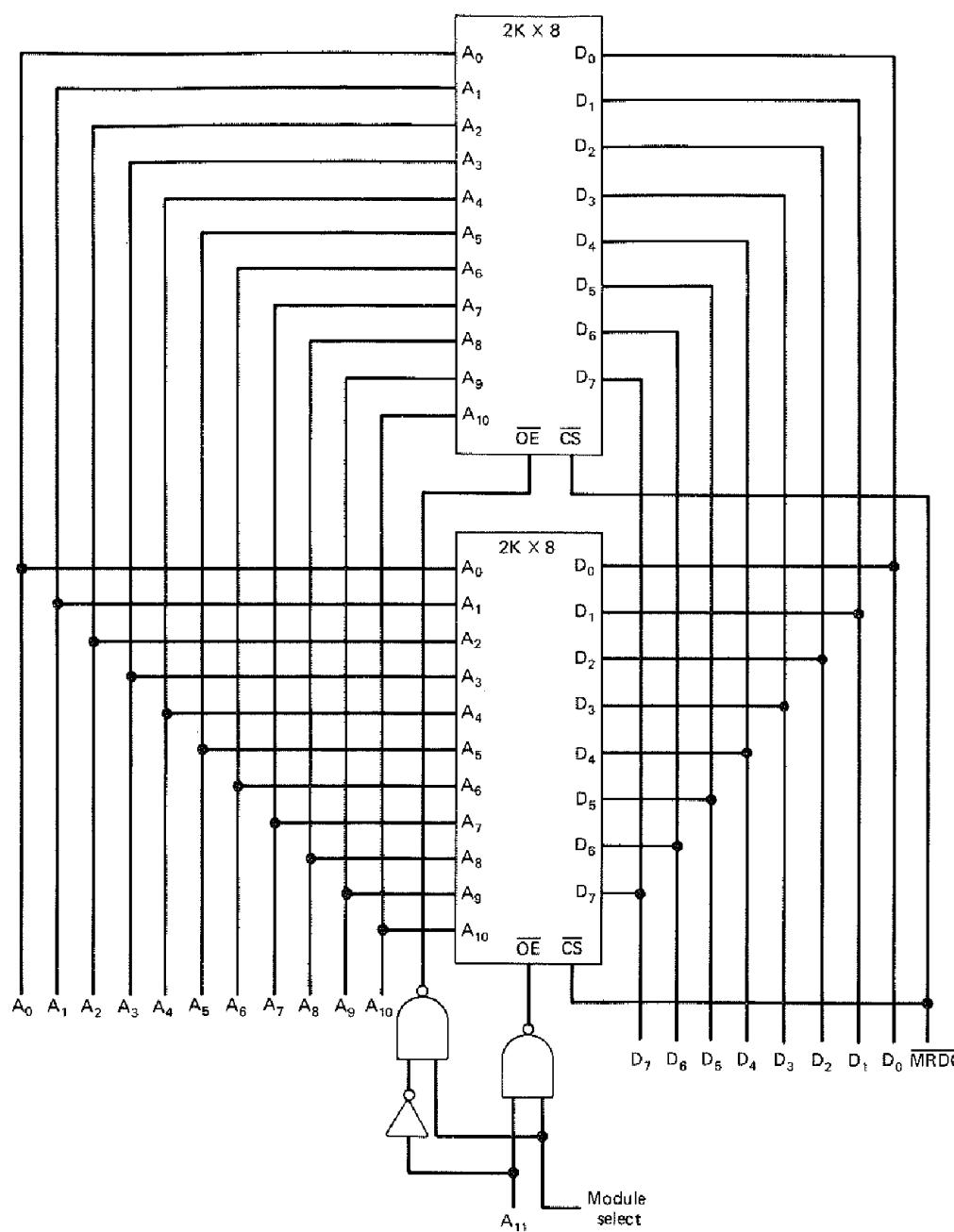


Figure 10-16 4K × 8 ROM memory. Glenn A. Gibson, Yu-Cheng Liu, MICROCOMPUTERS FOR ENGINEERS AND SCIENTISTS © 1980, Prentice-Hall, Inc.

to its natural state, thus destroying the old memory contents. The EPROM can then be reprogrammed.

As with PROMs, EPROMs are normally used during the development of a product and are replaced by masked ROMs once the design is complete. Because they can be erased, they have the advantage of being reusable. However, unlike PROMs, EPROMs may lose their contents over extended periods of time and should not be put in final products that are expected to last for years. How long one can depend on an EPROM retaining its contents depends on the environmental conditions it is subjected to, and may vary from a few months to several years.

An EPROM is programmed by applying an address to the address pins and high or low voltages to *all* of the data output pins, and then applying the proper voltages and pulses to the supply and control pins. As an example, to program an Intel 2764 PROM (8K × 8), 21 V is applied to the  $V_{PP}$  pin while the  $\overline{CE}$  pin is held low. The address of the byte to be programmed is put on pins A12-A0 and the data byte is simultaneously applied to O7-O0. The data byte is then written into the addressed byte by pulsing PGM with +5 V. The contents of each byte should be verified after it has been written.

Even PROM and EPROM memory devices made by one manufacturer may have programming signal specifications that vary considerably. Particularly in the case of PROM programming, the programming specifications may be complex and must be followed precisely. Therefore, many manufacturers have made available units, called *PROM programmers*, especially designed to program PROMs and EPROMs. To accommodate the varying electrical requirements, some programmers are built to accept circuit modules which provide customized electrical outputs. Each module is called a *personality card* and provides all the electrical signals needed for programming a given type of memory chip. In addition to personality cards for the various PROMs and EPROMs, a PROM programmer may have a control module which includes a microprocessor. The control module permits a block of data from a microcomputer development system (MDS) to be written into a PROM (or EPROM) with a single command. A programmer may also include a display for verifying the contents of a PROM. In conjunction with the Intel Universal PROM programmer, commands are available under the ISIS-II operating system for performing the following operations:

1. Loading the data to be programmed from a selected input device (disk file, paper tape, or system console) into the MDS memory.
2. Displaying or changing data in the MDS memory.
3. Programming a segment of a PROM with the data which are stored beginning at a specified address in the MDS memory.
4. Transferring a block of data in a PROM into memory so that the contents of the PROM may be examined through the system console or used to produce a duplicate PROM.
5. Transferring a block of data from a PROM into a disk file.

6. Comparing a block of data in a PROM with the contents of a segment of memory (program verification).

An E<sup>2</sup>ROM is also reprogrammable, but offers the added advantage that each individual byte can be easily erased and reprogrammed. The principal disadvantage of an E<sup>2</sup>ROM is that it tends to be more expensive.

## BIBLIOGRAPHY

1. 1982 Data Components Catalog (Santa Clara, Calif.: Intel Corporation, 1982).
2. Gibson, Glenn A., and Yu-cheng Liu, *Microcomputers for Engineers and Scientists* (Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1980).
3. Hall, Douglas V., *Microprocessors and Digital Systems* (New York: McGraw-Hill Book Company, 1980).
4. Givone, Donald D., and Robert P. Roesser, *Microprocessors/Microcomputers: An Introduction* (New York: McGraw-Hill Book Company, 1980).
5. Intel Memory Design Handbook (Santa Clara, Calif.: Intel Corporation, 1977).
6. Luecke, Gerald, Jack P. Mize, and William N. Carr, *Semiconductor Memory Design and Application* (New York: McGraw-Hill Book Company, 1973).

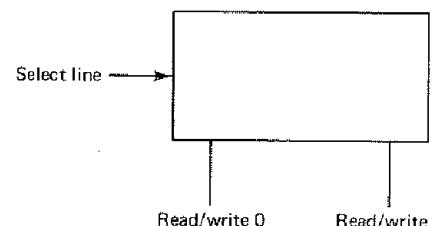
## EXERCISES

1. Complete the following table:

Memory size	Type of device	Device array	
		Number of rows	Number of columns
4K × 8	2K × 4		
64K × 8	32K × 1		
8K × 16	4K × 4		
1M × 16	64K × 1		

2. For a 32K-byte memory with a single error detection (parity) bit for each byte, determine the configuration of the chip array in terms of the numbers of memory chips in each row, in each column, and in the entire module using 8K × 1 RAMs. Repeat for 16K × 1 RAMs.
3. Construct a table that summarizes the discussion of the major design considerations given in Sec. 10-1.
4. Draw a logic diagram that shows the details of how the block diagram in Fig. 10-5 could

be implemented for a two-row by two-column cell array. The following symbol is to be used to represent a cell:



5. Show how the design in Fig. 10-8 could be simplified if the total memory capacity were only 8K × 8. Show how the design could be further simplified if a single-board minimum mode 8088 were assumed.
6. Consider a 256K × 8 memory module constructed of 32K × 1 dynamic RAM devices with 128 rows of cells. If the entire memory is to be refreshed once each millisecond, what would be the period of the row refresh cycles? Plot the percent of time required for memory refresh versus the memory cycle time as the memory cycle time increases from 100 ns to 800 ns.
7. Redesign the logic shown in Fig. 10-11 assuming that the module is to be a 64K-byte module constructed of 32K × 1 devices having 128 rows of cells.
8. Summarize the advantages and disadvantages of dynamic RAMs as opposed to the static RAMs.
9. Suppose that 64K × 1 dynamic RAM devices supplied with a +5-V supply are to be used to construct a 256K-byte memory module. If each device dissipates 1 W and the support logic requires 2 W, what is the total discharge current needed by the module? How many batteries would be needed to provide backup power for 8 h if the batteries are rated at 4 A·h at a discharge current of 2 A?
10. Repeat Exercise 9 assuming that backup power is needed for only 15 minutes but the batteries' rated discharge current is only 0.7 A.
11. Draw the necessary interfacing logic for the 4K-byte ROM module in Fig. 10-16.
12. One of the major applications of ROM is to replace SSI packages in implementing complex logic functions. Consider the following Boolean function:

$$\begin{aligned} f = & X_0 \bar{X}_1 \bar{X}_2 X_3 X_4 X_5 X_6 X_7 \bar{X}_8 \\ & + \bar{X}_0 \bar{X}_1 X_2 \bar{X}_3 X_4 X_5 X_6 X_7 \\ & + X_2 \bar{X}_3 \bar{X}_4 \bar{X}_5 X_6 X_7 X_8 \\ & + X_0 \bar{X}_2 X_3 X_4 X_5 X_6 \bar{X}_7 \bar{X}_8 \end{aligned}$$

How many basic logic gates (i.e., inverters, two-input AND gates, and two-input OR gates) would be required to implement this function? If a 512 × 8 ROM is used to implement the same function and the least significant data output bit is used as the output  $f$ , what should be the contents of the ROM? (Hint: Only the least significant data bit of each word is important.) Also, discuss the propagation delay associated with each approach.

13. How many Boolean functions can be implemented by a single 2K × 8 ROM, and what are the restrictions on the input variables of these functions?