Memory Organization

Every computer system contains a variety of devices to store the instructions and data required for its operation. These storage devices plus the algorithms (either implemented by hardware or software) needed to control or manage the stored information constitute the memory system of the computer.

The memory components of a computer system can be divided into three main groups:

1. *Internal processor memory.* This comprises a small set of high-speed registers used as a working memory for temporary storage of instructions and data
2. *Main memory* (also called primary memory). This is a relatively large fast memory used for program and data storage during computer operation. It is characterized by the fact that locations in main memory can be accessed directly and rapidly by the CPU instruction set. The principal technology used for main memory is based on semiconductor integrated circuits (ICs).
3. *Secondary memory* (also called auxiliary or backing memory). This is generally much larger in capacity but also much slower than main memory. It is used for storing system programs, large data files, and the like which are not continually required by the CPU; it also serves as an overflow memory when the capacity of the main memory is exceeded. Information in secondary storage is accessed indirectly via input-output programs that first transfer the required information to main memory. Representative technologies used for secondary memory are magnetic disks and tapes.

An increasing number of machines employ another type of memory called a *cache*, which serves as an intermediate temporary storage unit logically positioned between the processor registers and main memory. Unlike the other memory levels mentioned above, caches are transparent to the programmer.

Characteristics of Memory device :

Cost. The cost of a memory unit is most meaningfully measured by the purchase or lease price to the user of the complete unit. The price should include not only the cost of the information storage cells themselves but also the cost of the peripheral equipment or access circuitry essential to the operation of the memory. Let C be the price in dollars of a complete memory system with S bits of storage capacity. We define the *cost* c of the memory as follows:

$$c = \frac{C}{S} \text{ dollars/bit}$$

Access time. The performance of a memory device is primarily determined by the rate at which information can be read from or written into the memory. A convenient performance measure is the average time required to read a fixed amount of information, e.g., one word, from the memory. This is termed as read *access time* or, more commonly, the *access time* of the memory and is denoted by t_A. (The write access time is defined similarly; it is typically, but not always, equal to the read access time.) Access time depends on the physical characteristics of the storage medium, and also on the type of access mechanism used; a precise general definition of t_A is difficult. It is usually calculated from the time a read request is received by the memory unit to the time at which all the requested information has been made available at the memory output terminals. The *access rate* b_A of that memory defined as $1/t_A$ and measured in words per second is another widely used performance measure for memory devices.

Access modes. An important property of a memory device is the order or sequence in which information can be accessed. If locations may be accessed in any order and access time is independent of the location being accessed, the memory is termed a *random-access memory* (RAM). Ferrite-core and semiconductor memories are usually of this type. Memories where storage locations can be accessed only in certain predetermined sequences are called *serial-access memories*. Magnetic-tape, magnetic-bubble, and optical memories employ serial access methods.

Alterability; ROMs. The method used to write information into a memory may be irreversible, in that once information has been written, it cannot be altered while the memory is in use, i.e., on-line. Punching holes in cards and printing on paper are examples of essentially permanent storage techniques. Memories whose contents cannot be altered on-line (if they can be altered at all) are called *read-only memories* (ROMs). A ROM is therefore a nonerasable storage device. ROMs are widely used for storing control programs such as microprograms. ROMs whose contents can be changed (usually off-line and with some difficulty) are called *programmable read-only memories* (PROMs).

Memories in which reading or writing can be done with impunity on-line are sometimes called *read-write* memories to contrast them with ROMs. All memories used for temporary storage purposes are read-write memories. [redacted]

Permanence of storage. The physical processes involved in storage are sometimes inherently unstable, so that the stored information may be lost over a period of time unless appropriate action is taken. There are three important memory characteristics that can destroy information: destructive readout, dynamic storage, and volatility. Some memories have the property that the method of reading the memory destroys the stored information; this phenomenon is called *destructive readout* (DRO). Memories in which reading does not affect the stored data are said to have *nondestructive readout* (NDRO). In DRO memories, each read operation must be followed by a write operation that restores the original state of the memory. This restoration is usually carried out automatically using a buffer register as shown in Fig. 5.4. The word at the addressed location is transferred to the buffer register where it is available to external devices. The contents of the buffer are automatically written back into the location originally addressed.

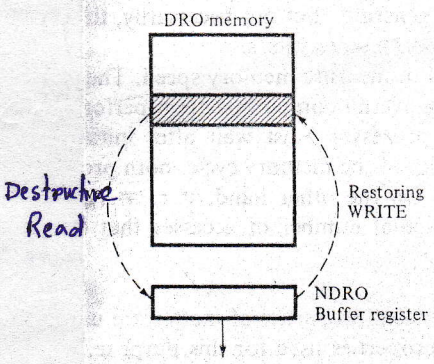


FIGURE 5.4
Memory restoration in a destructive readout (DRO) memory.

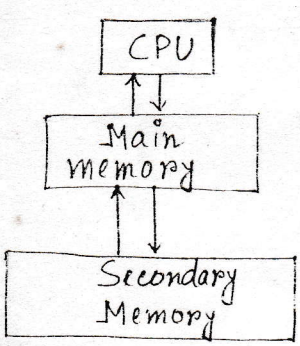
Cycle time and data-transfer rate. We defined the access time t_A of a memory as the time between the receipt of a read request by the memory and the delivery of the requested information to its external output terminals. In DRO and dynamic memories, it may not be possible to initiate another memory access until a restore or refresh operation has been carried out. This means that the minimum time that must elapse between the initiation of two different accesses by the memory can be greater than t_A; this rather loosely defined time is called the *cycle time* t_M of the memory.

It is generally convenient to assume that t_M is the time needed to complete any read or write operation in the memory. Hence the maximum amount of information that can be transferred to or from the memory every second is $1/t_M$; this quantity is called the *data-transfer rate* or *bandwidth* b_M. The data-transfer rate is measured in bits or words per second. A factor limiting memory bandwidth is the *memory bus width* w, which is the number of bits that can be transferred simultaneously over the memory bus. w is generally, but not necessarily, the same as the internal memory word size. Clearly $b_M = w/t_M$ bits/s.

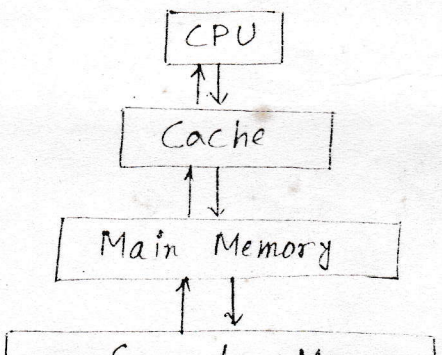
Memory hierarchy : The major units in a typical memory system can be viewed as forming a hierarchy of memories ($M_1, M_2, \ldots M_n$) in which each member M_i is in some sense sub-ordinates to the next highest member M_{i+1} of the hierarchy. In general, all the information stored in M_{i-1} at any time is also stored in M_i, but not vice-versa. The CPU communicate directly with the first member of the hierarchy. M_1, M_1 can communicate with M_2, and so on. Let C_i, t_{Ai}, and S_i denote the cost per bit, access time, and storage capacity, respectively, of M_i. The following relations normally hold between the memory levels M_i and M_{i+1} :

$$C_i > C_{i+1}$$
$$t_{Ai} < t_{Ai+1}$$
$$S_i < S_{i+1}.$$

Figure bellow shows the two most common memory hierarchy. Typical technologies used in these hierarchies are bipolar semiconductor RAMs for cache memory, MOS semiconductor RAMs for main memory, and magnetic disk for secondary memory.



(a)



(b)

Figure : Common memory hierarchy (a) Two-level (b) Three level

RAM (Random Access Memory)

Random-access memories (RAMs) are characterized by the fact that every location can be accessed independently. The access and cycle times for every location are constant and independent of its position.

Figure 5.6 shows the main components of a random-access memory unit. The storage cell unit comprises N cells each of which can store 1 bit of information. The memory operates as follows The address of the required location (a set of $w \geq 1$ cells) is transferred via the address bus to the memory address register. The address is then processed by the address decoder which selects the required location in the storage cell unit. A read-write select control line specifies the type of access to be performed. If read is requested, the contents of the selected location is transferred to the output data register. If write is requested, the word to be written is first placed in the memory input data register and then transferred to the selected cell. Since it is not usually desirable to permit simultaneous reading and writing, the input and output data registers are frequently combined to form a single data register (also called the memory buffer register). The input and output parts of the data bus may then be merged to form a single bidirectional data bus.

Figure 5.7 shows an idealized model of a RAM cell and its external connections. The address lines are used to select the cell for either reading or writing, as determined by the read-write control lines. A set of data lines is used for transferring data to and from the memory.

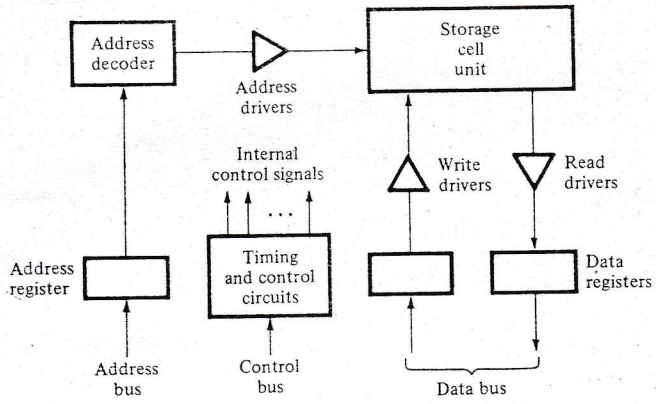


FIGURE 5.6
A random-access memory unit.

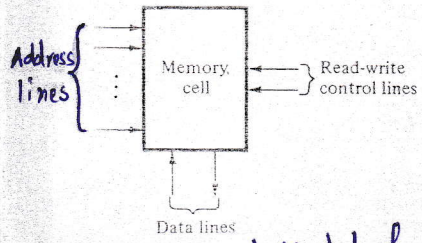


FIG 5.7 : General Model of RAM cell.

RAM organization. The access circuitry needed has a very significant effect on the total cost of any memory unit. A general approach to reducing the access circuitry cost in random-access memories is called *matrix*, or *array*, *organization*. It has two essential features:

1. The storage cells are physically arranged as rectangular arrays of cells. This is primarily to facilitate layout of the connections between the cells and the access circuitry.
2. The memory address is partitioned into d components so that the address A_i of cell C_i becomes a d-dimensional vector $(A_{i,1}, A_{i,2}, \ldots, A_{i,d}) = A_i$. Each of the d parts of an address word goes to a different address decoder and a different set of address drivers. A particular cell is selected by simultaneously activating all d of its address lines. A memory unit with this kind of addressing is said to be a *d-dimensional memory*.

The simplest array organizations have $d = 1$ and are called *one-dimensional*, or 1-D, memories. Each cell is connected to one address line, as shown in Fig. 5.8. If the storage capacity of the unit is N bits, then the access circuitry typically contains a one-out-of-N address decoder and N address drivers. In the *two-dimensional* (2-D) organization shown in Fig. 5.9, the address field is divided into two components, called X and Y, which consist of a_x and a_y bits, respectively. The cells

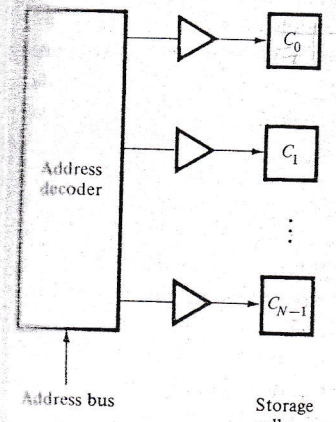


FIGURE 5.8
One-dimensional addressing scheme.

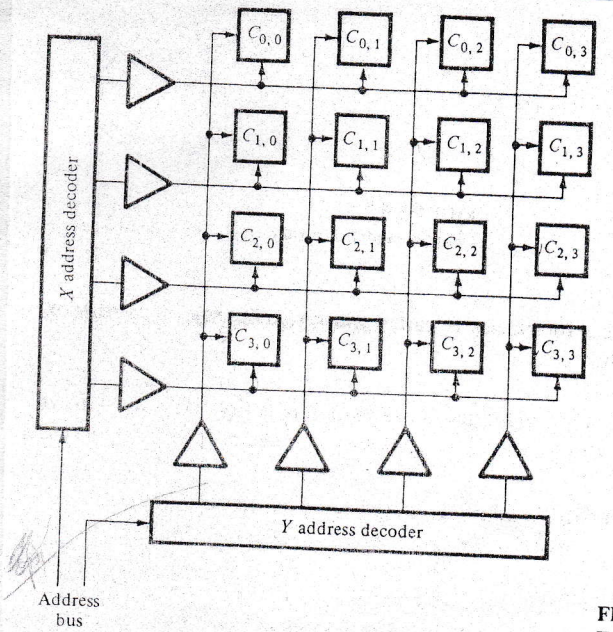


FIGURE 5.9
Two-dimensional addressing scheme

are arranged in a rectangular array of $N_x \leq 2^{a_x}$ rows and $N_y \leq 2^{a_y}$ columns so that the total number of cells is $N = N_x N_y$. A cell is selected by the coincidence of signals on its X and Y address lines. The 2-D organization requires substantially less access circuitry than the 1-D for a fixed amount of storage. For example, if $N_x = N_y = \sqrt{N}$, the number of address drivers needed is $2\sqrt{N}$. Instead of one one-out-of-N address decoder, two one-out-of-$\sqrt{N}$ address decoders are required. In addition, the 2-D memory organization matches well the inherently two-dimensional circuit structures allowed by IC technology. These advantages diminish or disappear with larger numbers of dimensions; hence values of d greater than 2 are rarely used. If the 1-bit storage cells of Figs. 5.8 and 5.9 are replaced by w-bit registers, then an entire word can be accessed in each read or write cycle but the bits within a word are not individually addressable. A RAM of this sort is referred to as an $N \times w$-bit *word-organized* memory.

Table 5.1 lists the major types of semiconductor memory. The most common is referred to as *random-access memory* (RAM). This is, of course, a misuse of the term, because all of the types listed in the table are random access. One distinguishing characteristic of RAM is that it is possible both to read data from the memory and to write new data into the memory easily and rapidly. Both the reading and writing are accomplished through the use of electrical signals.

The other distinguishing characteristic of RAM is that it is volatile. A RAM must be provided with a constant power supply. If the power is interrupted, then the

Table 5.1 Semiconductor Memory Types

Memory Type	Category	Erasure	Write Mechanism	Volatility
Random-access memory (RAM)	Read-write memory	Electrically, byte-level	Electrically	Volatile
Read-only memory (ROM)	Read-only memory	Not possible	Masks	Nonvolatile
Programmable ROM (PROM)			Electrically	
Erasable PROM (EPROM)	Read-mostly memory	UV light, chip-level		
Electrically Erasable PROM (EEPROM)		Electrically, byte-level		
Flash memory		Electrically, block-level		

data are lost. Thus, RAM can be used only as temporary storage. The two traditional forms of RAM used in computers are DRAM and SRAM.

Dynamic RAM RAM technology is divided into two technologies: dynamic and static. A dynamic RAM (DRAM) is made with cells that store data as charge on capacitors. The presence or absence of charge in a capacitor is interpreted as a binary 1 or 0. Because capacitors have a natural tendency to discharge, dynamic RAMs require periodic charge refreshing to maintain data storage. The term *dynamic* refers to this tendency of the stored charge to leak away, even with power continuously applied.

Figure 5.2a is a typical DRAM structure for an individual cell that stores one bit. The address line is activated when the bit value from this cell is to be read or written. The transistor acts as a switch that is closed (allowing current to flow) if a voltage is applied to the address line and open (no current flows) if no voltage is present on the address line.

For the write operation, a voltage signal is applied to the bit line; a high voltage represents 1, and a low voltage represents 0. A signal is then applied to the address line, allowing a charge to be transferred to the capacitor.

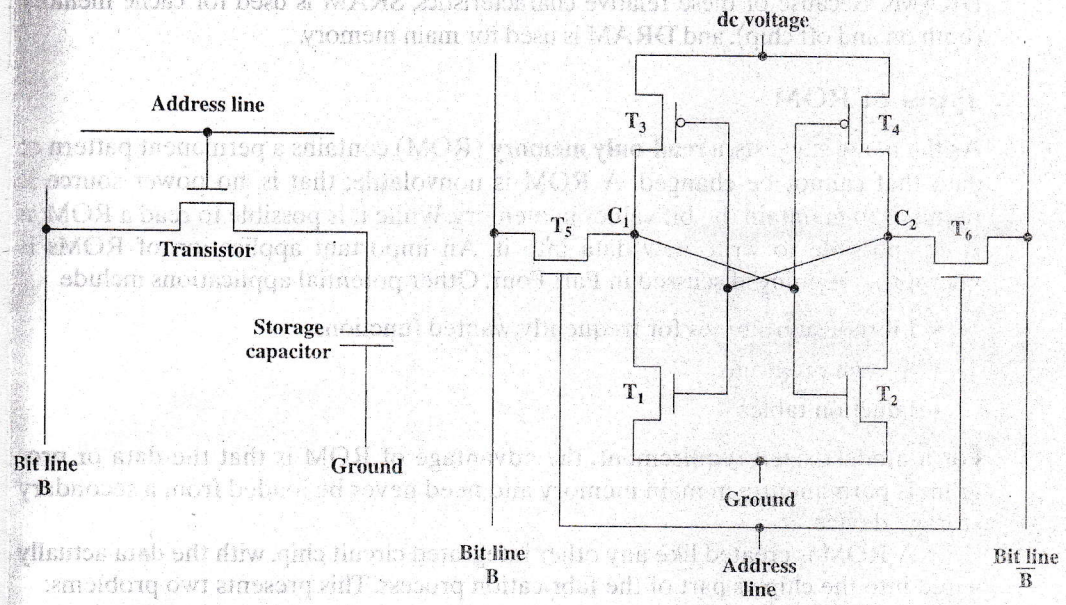
For the read operation, when the address line is selected, the transistor turns on and the charge stored on the capacitor is fed out onto a bit line and to a sense amplifier. The sense amplifier compares the capacitor voltage to a reference value and determines if the cell contains a logic 1 or a logic 0. The readout from the cell discharges the capacitor, which must be restored to complete the operation.

Although the DRAM cell is used to store a single bit (0 or 1), it is essentially an analog device. The capacitor can store any charge value within a range; a threshold value determines whether the charge is interpreted as 1 or 0.

Static RAM In contrast, a static RAM (SRAM) is a digital device, using the same logic elements used in the processor. In a SRAM, binary values are stored using traditional flip-flop logic-gate configurations (see Appendix B for a description of flip-flops). A static RAM will hold its data as long as power is supplied to it.

Figure 5.2b is a typical SRAM structure for an individual cell. Four transistors (T_1, T_2, T_3, T_4) are cross connected in an arrangement that produces a stable logic state. In logic state 1, point C_1 is high and point C_2 is low; in this state, T_1 and T_4 are off and T_2 and T_3 are on.[1] In logic state 0, point C_1 is low and point C_2 is high; in this state, T_1 and T_4 are on and T_2 and T_3 are off. Both states are stable as long as the direct current (dc) voltage is applied. Unlike the DRAM, no refresh is needed to retain data.

As in the DRAM, the SRAM address line is used to open or close a switch. The address line controls two transistors (T_5 and T_6). When a signal is applied to this line, the two transistors are switch on, allowing a read or write operation. For a write operation, the desired bit value is applied to line B, while its complement is applied to line $\overline{B}$. This forces the four transistors (T_1, T_2, T_3, T_4) into the proper state. For a read operation, the bit value is read from line B.



(a) Dynamic RAM (DRAM) cell

(b) Static RAM (SRAM) cell

Figure 5.2 Typical Memory Cell Structures

SRAM versus DRAM Both static and dynamic RAMs are volatile; that is, power must be continuously supplied to the memory to preserve the bit values. A dynamic memory cell is simpler and smaller than a static memory cell. Thus, a DRAM is more dense (smaller cells = more cells per unit area) and less expensive than a corresponding SRAM. On the other hand, a DRAM requires the supporting refresh circuitry. For larger memories, the fixed cost of the refresh circuitry is more than compensated for by the smaller variable cost of DRAM cells. Thus, DRAMs tend to be favored for large memory requirements. A final point is that SRAMs are generally somewhat faster than DRAMs. Because of these relative characteristics, SRAM is used for cache memory (both on and off chip), and DRAM is used for main memory.

A semiconductor RAM IC typically has a word-organized array structure and contains all required access circuitry, including address decoders, drivers, and control circuits. Figure 5.13 shows a simple 4 × 2-bit RAM that incorporates eight bipolar cells of the type shown in Fig. 5.2. The more important access circuitry is also shown. WE is the *write enable* line; a write (read) operation can take place only if WE = 1 (0). A second control line, the *chip enable* CE or *chip select* line CS, is also needed. A word can be accessed for either reading or writing only if CE = 1. The behavior of the bidirectional data lines connected to each cell is determined by the underlying device physics.

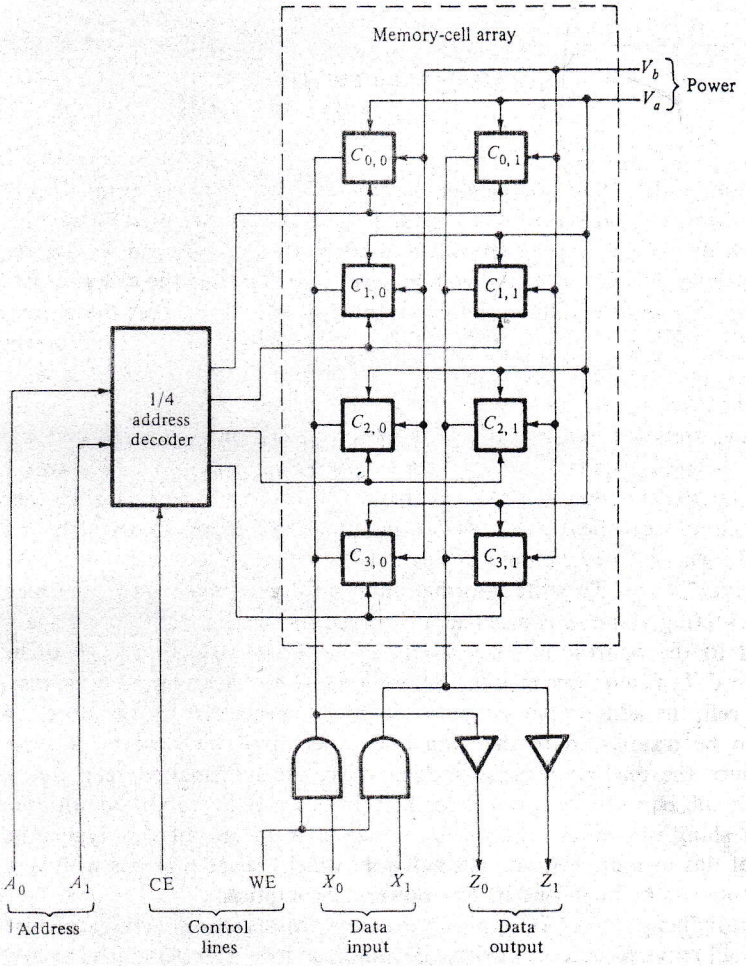
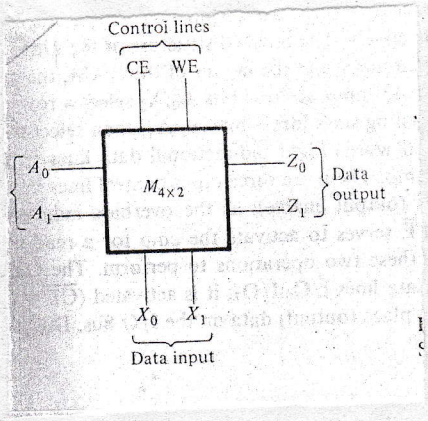


FIGURE 5.13
Structure of a 4 × 2-bit RAM.

Consider the design of a 16 × 4-bit memory using 4 × 2-bit ICs of the type shown in Fig. 5.13. It is convenient to represent each IC by a single block with its external connections labeled as in Fig. 5.14. Clearly eight of these ICs are needed. They can be arranged as a 4 × 2 array as shown in Fig. 5.15. The left column of ICs stores the two low-order data bits, while the right column stores the two high-order data bits. Since there are four address lines, some additional decoding circuitry is needed. We therefore introduce a one-out-of-four decoder with an address enable input similar to the decoder shown in Fig. 5.13. Two of the incoming address lines are connected to every IC; the remaining two address lines are inputs to the external decoder. Each of the output lines of this decoder is connected to the address enable inputs of the ICs in the same row. Thus each row of cells in the resulting array has a unique address. The output data lines of all cells in the same column are connected together under the assumption (which is valid for many semiconductor technologies) that this connection forms a wired-OR.



Address lines

Fig 5.14: Symbol for 4x2 RAM

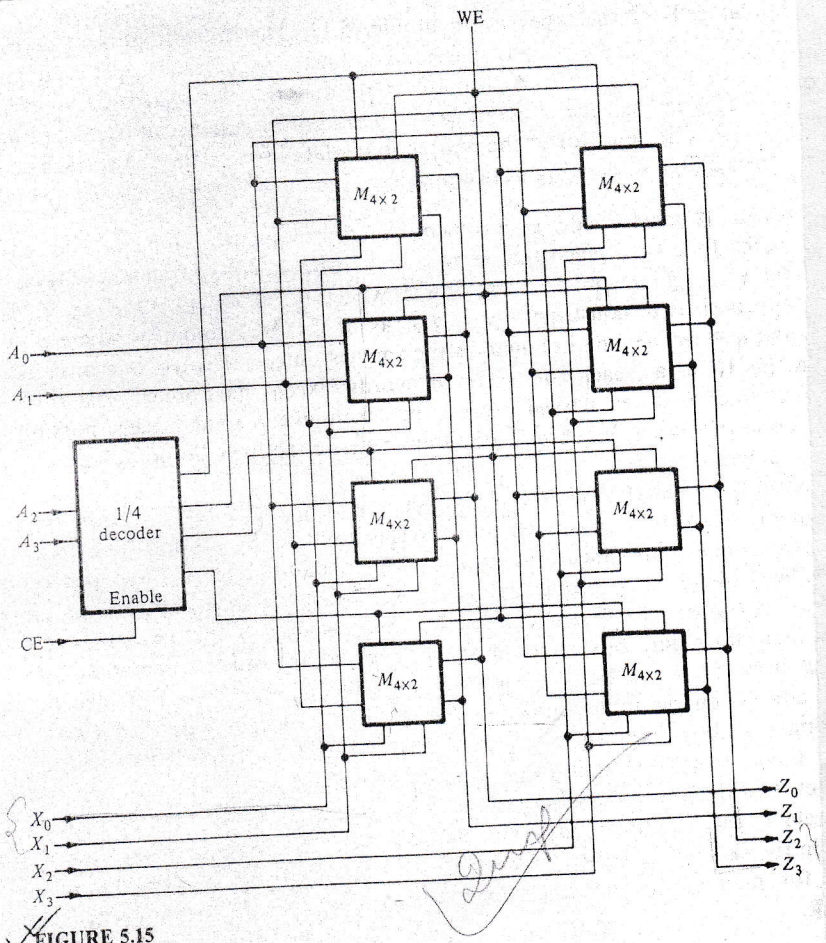


FIGURE 5.15
A 16 × 4-bit RAM.

In recent years, a number of enhancements to the basic DRAM architecture have been explored, and some of these are now on the market. The schemes that currently dominate the market are SDRAM, DDR-DRAM, and RDRAM. [redacted] CDRAM has also received considerable attention. We examine each of these approaches in this section.

Synchronous DRAM

One of the most widely used forms of DRAM is the synchronous DRAM (SDRAM) [redacted]. Unlike the traditional DRAM, which is asynchronous, the SDRAM exchanges data with the processor synchronized to an external clock signal and running at the full speed of the processor/memory bus without imposing wait states.

In a typical DRAM, the processor presents addresses and control levels to the memory, indicating that a set of data at a particular location in memory should be either read from or written into the DRAM. After a delay, the access time, the DRAM either writes or reads the data. During the access-time delay, the DRAM performs various internal functions, such as activating the high capacitance of the row and column lines, sensing the data, and routing the data out through the output buffers. The processor must simply wait through this delay, slowing system performance.

With synchronous access, the DRAM moves data in and out under control of the system clock. The processor or other master issues the instruction and address information, which is latched by the DRAM. The DRAM then responds after a set number of clock cycles. Meanwhile, the master can safely do other tasks while the SDRAM is processing the request.

Rambus DRAM

RDRAM, developed by Rambus [redacted], has been adopted by Intel for its Pentium and Itanium processors. It has become the main competitor to SDRAM. RDRAM chips are vertical packages, with all pins on one side. The chip exchanges data with the processor over 28 wires no more than 12 centimeters long. The bus can address up to 320 RDRAM chips and is rated at 1.6 GBps.

The special RDRAM bus delivers address and control information using an asynchronous block-oriented protocol. After an initial 480 ns access time, this produces the 1.6 GBps data rate. What makes this speed possible is the bus itself, which defines impedances, clocking, and signals very precisely. Rather than being controlled by the explicit RAS, CAS, R/W, and CE signals used in conventional DRAMs, an RDRAM gets a memory request over the high-speed bus. This request contains the desired address, the type of operation, and the number of bytes in the operation.

DDR SDRAM

SDRAM is limited by the fact that it can only send data to the processor once per bus clock cycle. A new version of SDRAM, referred to as double-data-rate SDRAM can send data twice per clock cycle, once on the rising edge of the clock pulse and once on the falling edge.

Cache DRAM

Cache DRAM (CDRAM), developed by Mitsubishi [redacted], integrates a small SRAM cache (16 Kb) onto a generic DRAM chip.

The SRAM on the CDRAM can be used in two ways. First, it can be used as a true cache, consisting of a number of 64-bit lines. The cache mode of the CDRAM is effective for ordinary random access to memory.

The SRAM on the CDRAM can also be used as a buffer to support the serial access of a block of data. For example, to refresh a bit-mapped screen, the CDRAM can prefetch the data from the DRAM into the SRAM buffer. Subsequent accesses to the chip result in accesses solely to the SRAM.

Types of ROM

As the name suggests, a **read-only memory** (ROM) contains a permanent pattern of data that cannot be changed. A ROM is nonvolatile; that is, no power source is required to maintain the bit values in memory. While it is possible to read a ROM, it is not possible to write new data into it. An important application of ROMs is microprogramming, [redacted]. Other potential applications include

- Library subroutines for frequently wanted functions
- System programs
- Function tables

For a modest-sized requirement, the advantage of ROM is that the data or program is permanently in main memory and need never be loaded from a secondary storage device.

When only a small number of ROMs with a particular memory content is needed, a less expensive alternative is the **programmable ROM** (PROM). Like the ROM, the PROM is nonvolatile and may be written into only once. For the PROM, the writing process is performed electrically and may be performed by a supplier or customer at a time later than the original chip fabrication. Special equipment is required for the writing or "programming" process. PROMs provide flexibility and convenience. The ROM remains attractive for high-volume production runs.

Another variation on read-only memory is the read-mostly memory, which is useful for applications in which read operations are far more frequent than write operations but for which nonvolatile storage is required. There are three common forms of read-mostly memory: EPROM, EEPROM, and flash memory.

The optically **erasable programmable read-only memory** (EPROM) is read and written electrically, as with PROM. However, before a write operation, all the storage cells must be erased to the same initial state by exposure of the packaged chip to ultraviolet radiation. Erasure is performed by shining an intense ultraviolet light through a window that is designed into the memory chip. This erasure process can be performed repeatedly; each erasure can take as much as 20 minutes to perform. Thus, the EPROM can be altered multiple times and, like the ROM and PROM, holds its data virtually indefinitely. For comparable amounts of storage, the EPROM is more expensive than PROM, but it has the advantage of the multiple update capability.

A more attractive form of read-mostly memory is **electrically erasable programmable read-only memory** (EEPROM). This is a read-mostly memory that can be written into at any time without erasing prior contents; only the byte or bytes addressed are updated. The write operation takes considerably longer than the read operation, on the order of several hundred microseconds per byte. The EEPROM combines the advantage of nonvolatility with the flexibility of being updatable in place, using ordinary bus control, address, and data lines. EEPROM is more expensive than EPROM and also is less dense, supporting fewer bits per chip.

Another form of semiconductor memory is **flash memory** (so named because of the speed with which it can be reprogrammed). First introduced in the mid-1980s, flash memory is intermediate between EPROM and EEPROM in both cost and functionality. Like EEPROM, flash memory uses an electrical erasing technology. An entire flash memory can be erased in one or a few seconds, which is much faster than EPROM. In addition, it is possible to erase just blocks of memory rather than an entire chip. Flash memory gets its name because the microchip is organized so that a section of memory cells are erased in a single action or "flash." However, flash memory does not provide byte-level erasure. Like EPROM, flash memory uses only one transistor per bit, and so achieves the high density (compared with EEPROM) of EPROM.

Associative Memory:

The time required to find an item stored in memory can be reduced considerably if stored data can be identified for access by the content of the data itself rather than by an address. A memory unit accessed by content is called an *associative memory* or *content addressable memory* (CAM). This type of memory is accessed simultaneously and in parallel on the basis of data content rather than by specific address or location. When a word is written in an associative memory, no address is given. The memory is capable of finding an empty unused location to store the word. When a word is to be read from an associative memory, the content of the word, or part of the word, is specified. The memory locates all words which match the specified content and marks them for reading.

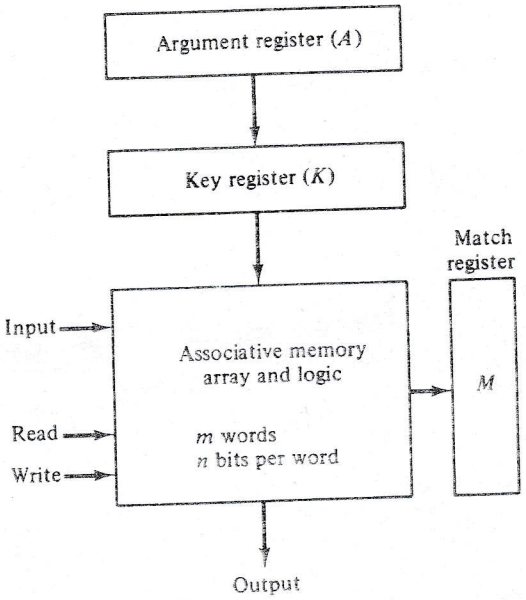
Because of its organization, the associative memory is uniquely suited to do parallel searches by data association. Moreover, searches can be done on an entire word or on a specific field within a word. An associative memory is more expensive than a random access memory because each cell must have storage capability as well as logic circuits for matching its content with an external argument. For this reason, associative memories are used in applications where the search time is very critical and must be very short.

Hardware Organization

The block diagram of an associative memory is shown in Fig. 12-6. It consists of a memory array and logic for m words with n bits per word. The argument register A and key register K each have n bits, one for each bit of a word. The match register M has m bits, one for each memory word. Each word in memory is compared in parallel with the content of the argument register. The words that match the bits of the argument register set a corresponding bit in the match register. After the matching process, those bits in the match register that have been set indicate the fact that their corresponding words have been matched. Reading is accomplished by a sequential access to memory for those words whose corresponding bits in the match register have been set.

The key register provides a mask for choosing a particular field or key in the argument word. The entire argument is compared with each memory word if the key register contains all 1's. Otherwise, only those bits in the argument that have 1's in their corresponding position of the key register are compared. Thus the key provides a mask or identifying piece of information which

Figure 12-6 Block diagram of associative memory.



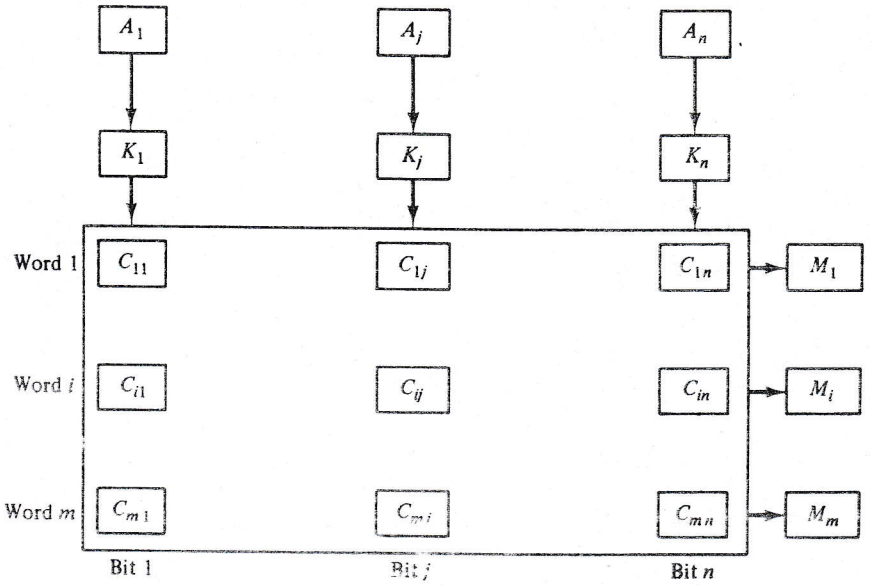
specifies how the reference to memory is made. To illustrate with a numerical example, suppose that the argument register A and the key register K have the bit configuration shown below. Only the three leftmost bits of A are compared with memory words because K has 1's in these positions.

A	101 111100	
K	111 000000	
Word 1	100 111100	no match
Word 2	101 000001	match

Word 2 matches the unmasked argument field because the three leftmost bits of the argument and the word are equal.

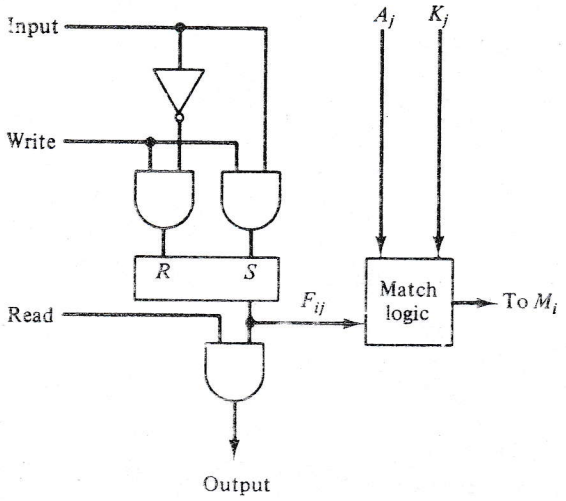
The relation between the memory array and external registers in an associative memory is shown in Fig. 12-7. The cells in the array are marked by the letter C with two subcripts. The first subscript gives the word number and the second specifies the bit position in the word. Thus cell C_{ij} is the cell for bit j in word i. A bit A_j in the argument register is compared with all the bits in column j of the array provided that $K_j = 1$. This is done for all columns $j = 1, 2, \ldots, n$. If a match occurs between all the unmasked bits of the argument and the bits in word i, the corresponding bit M_i in the match register is set to 1. If one or more unmasked bits of the argument and the word do not match, M_i is cleared to 0.

Figure 12-7 Associative memory of m word, n cells per word.



The internal organization of a typical cell C_{ij} is shown in Fig. 12-8. It consists of a flip-flop storage element F_{ij} and the circuits for reading, writing, and matching the cell. The input bit is transferred into the storage cell during a write operation. The bit stored is read out during a read operation. The match logic compares the content of the storage cell with the corresponding unmasked bit of the argument and provides an output for the decision logic that sets the bit in M_i.

Figure 12-8 One cell of associative memory.



Match Logic

The match logic for each word can be derived from the comparison algorithm for two binary numbers. First, we *neglect* the key bits and compare the argument in A with the bits stored in the cells of the words. Word i is equal to the argument in A if $A_j = F_{ij}$ for $j = 1, 2, \ldots, n$. Two bits are equal if they are both 1 or both 0. The equality of two bits can be expressed logically by the Boolean function

$$x_j = A_j F_{ij} + A_j' F_{ij}'$$

where $x_j = 1$ if the pair of bits in position j are equal; otherwise, $x_j = 0$.

For a word i to be equal to the argument in A we must have all x_j variables equal to 1. This is the condition for setting the corresponding match bit M_i to 1. The Boolean function for this condition is

$$M_i = x_1 x_2 x_3 \cdots x_n$$

and constitutes the AND operation of all pairs of matched bits in a word.

We now include the key bit K_j in the comparison logic. The requirement is that if $K_j = 0$, the corresponding bits of A_j and F_{ij} need no comparison. Only when $K_j = 1$ must they be compared. This requirement is achieved by ORing each term with K_j', thus:

$$x_j + K_j' = \begin{cases} x_j & \text{if } K_j = 1 \\ 1 & \text{if } K_j = 0 \end{cases}$$

When $K_j = 1$, we have $K_j' = 0$ and $x_j + 0 = x_j$. When $K_j = 0$, then $K_j' = 1$ and $x_j + 1 = 1$. A term $(x_j + K_j')$ will be in the 1 state if its pair of bits is not compared. This is necessary because each term is ANDed with all other terms so that an output of 1 will have no effect. The comparison of the bits has an effect only when $K_j = 1$.

The match logic for word i in an associative memory can now be expressed by the following Boolean function:

$$M_i = (x_1 + K_1')(x_2 + K_2')(x_3 + K_3') \cdots (x_n + K_n')$$

Each term in the expression will be equal to 1 if its corresponding $K_j = 0$. If $K_j = 1$, the term will be either 0 or 1 depending on the value of x_j. A match will occur and M_i will be equal to 1 if all terms are equal to 1.

If we substitute the original definition of x_j, the Boolean function above can be expressed as follows:

$$M_i = \prod_{j=1}^{n} (A_j F_{ij} + A_j' F_{ij}' + K_j')$$

where Π is a product symbol designating the AND operation of all n terms. We need m such functions, one for each word $i = 1, 2, 3, \ldots, m$.

The circuit for matching one word is shown in Fig. 12-9. Each cell requires two AND gates and one OR gate. The inverters for A_j and K_j are needed once for each column and are used for all bits in the column. The output of all OR gates in the cells of the same word go to the input of a common AND gate to generate the match signal for M_i. M_i will be logic 1 if a match occurs and 0 if no match occurs. Note that if the key register contains all 0's, output M_i will be a 1 irrespective of the value of A or the word. This occurrence must be avoided during normal operation.

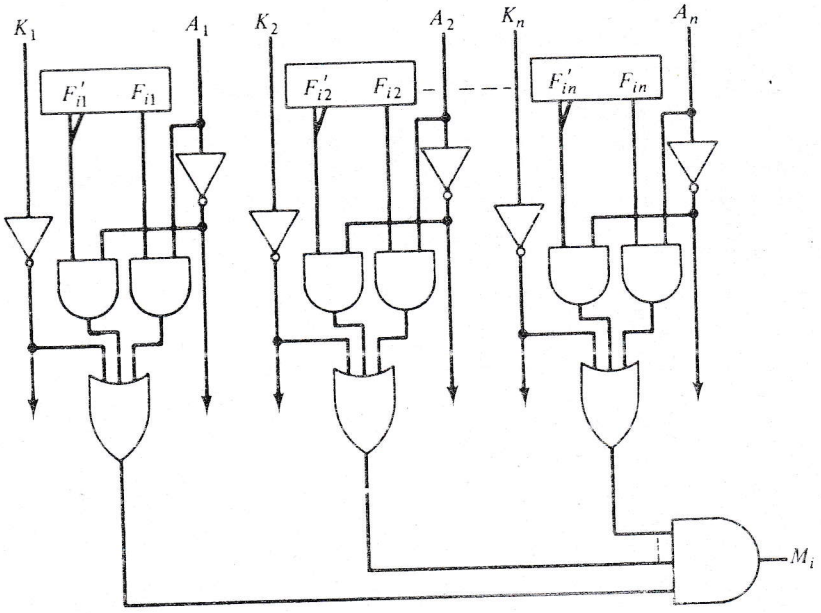


Figure 12-9 Match logic for one word of associative memory.