$\phi_2 = \phi_3 = \mathbf{0}$. Then the bit in storage in the flip-flop will be preserved on capacitors C_1 and C_2. If we return $\phi_2 = \phi_3 = \mathbf{1}$ before the charges on C_1 and C_2 have changed appreciably, the flip-flop state will not have changed.

Now let ϕ_2 and ϕ_3 go to logic **0**, as in Fig. 12.8-1*b*. (In the figure it is indicated that ϕ_2 and ϕ_3 go to logic **0** simultaneously. This feature is not essential; nor is it important which clock signal goes to **0** first.) Suppose that shortly after ϕ_2 and ϕ_3 both become **0**, ϕ_1 becomes $\phi_1 = \mathbf{1}$. Then because of the voltage held on C_2 the state of the flip-flop, i.e., the logic level held at the output of *I*2, will be transferred to the capacitor C_1' of the next flip-flop. Similarly, the state of the preceding flip-flop will be transferred to C_1. Now let ϕ_1 return to **0** and bit ϕ_2 be *first* to go back to logic level **1**, thereby closing switch $T2$. Then whatever may have been the previous state of C_2 it will now go to the state complementary to C_1. Finally, when ϕ_3 goes back to $\phi_3 = \mathbf{1}$, the flip-flop circuit loop is again complete and the flip-flop will hold its new state indefinitely. The order of the reclosing of switches $T2$ and $T3$ is important. For, as can be verified, if ϕ_3 recloses first, the flip-flop will revert to its original state.

12.9 THE READ-ONLY MEMORY

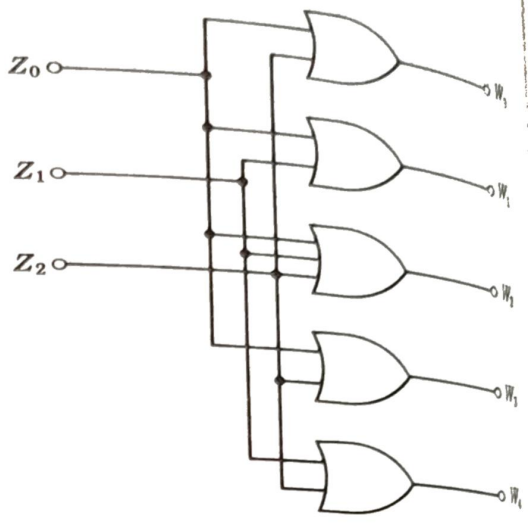
A read-only memory (ROM) is a memory device intended to store information which is fixed. That is, there is an initial operation during which information is written into the memory and thereafter the memory is *read only* and is not again written into. Generally the information in the memory is placed there by the manufacturer of the device. There are, however, memories which allow the user to establish the store of information in the memory. Such memories are referred to as programmable memories (PROM). There are also ROMs in which the stored information can be changed. However, in such cases the writing operation is accomplished in a time which is many orders of magnitude larger than the time required to read. Such memory devices (described as *erasable*) are read-only memories in the sense that to change the stored information it is necessary to interrupt the digital processing in which the memory is involved.

A most important attribute of the ROM is that the information it stores will not be lost if the electric power that it uses to operate should be interrupted. Such memories are referred to as *nonvolatile*. In contrast, the sequential memories already discussed and the random-access memories to be discussed (beginning in Sec. 12.13) are both *volatile* memories.

The ROM has many applications in a digital system. It can be used to provide the realization of an arbitrary truth table. Whenever a truth table involves enough input and output logical variables for its physical implementation to require a large number of gates, a ROM may well be more economical in size, weight, and cost. ROMs are widely used in code conversion and in connection with alphanumeric displays. The ROM is also used to yield results

Z_0	Z_1	Z_2	W_0	W_1	W_2	W_3	W_4
0	0	1	1	0	1	1	1
0	1	0	0	1	1	0	1
1	0	0	1	1	1	1	0

(*a*)



(*b*)

FIGURE 12.9-1
(*a*) A truth-table specification of an encoder. (*b*) The encoder realized with OR gates.

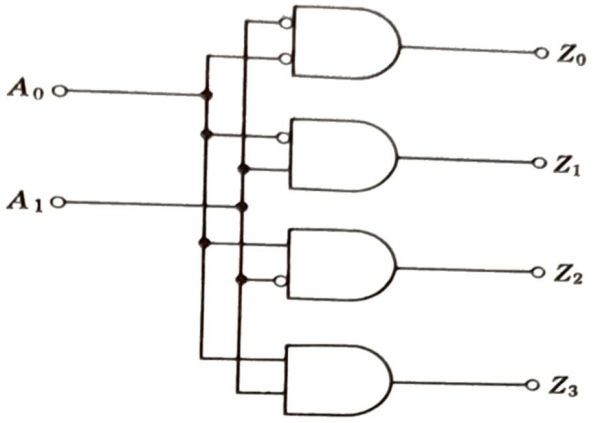
that would otherwise be achieved by a computation involving a sequence of arithmetic operations, e.g., multiplication, division, or evaluation of a trigonometric function (as sin x or ln x), or it may be used as a function generator.

A ROM is an *encoder*. An encoder is a logical-gate structure which has M inputs $Z_0, Z_1, \ldots, Z_{M-1}$ and K outputs $W_0, W_1, \ldots, W_{K-1}$. It is intended that at any time an individual input, say Z_i, is to be singled out, we set $Z_i =$ while all other inputs are at logic 0. (Alternatively we may have $Z_i = 0$ whil all other outputs are at logic 1.) Corresponding to each input Z_i, which ma be set to $Z_i = 1$, the K outputs will take on the logic levels $W_0(i)$, $W_1(i)$, ... $W_{K-1}(i)$. That is, the encoder accepts the input Z_i (only) $= 1$ and identifies th situation through the *code* word $W_0(i), W_1(i), \ldots, W_{K-1}(i)$; or if the encoder viewed in its application as a memory, the ith storage location in the memory *addressed* by setting Z_i (only) $= 1$ and the ROM responds by presenting at output the word stored in that location. A logical structure of an encoder illustrated in Fig. 12.9-1. The truth-table specification is given in Fig. 12.9- and its realization in logic gates is shown in Fig. 12.9-1*b*. The ROM illustrat stores three words, each of 5 bits.

Rather generally the address of a stored word is itself available in a digi system as a binary coded word. It is therefore necessary to interpose betwe the binary-coded address word and the ROM a device which takes note of address and singles out a corresponding individual line. Such a device perfor a function which is inverse to the function performed by the encoder and is he called a *decoder*. A truth table for a decoder is given in Fig. 12.9-2*a*. address word has 2 bits; hence four address words are available. A realizat of this decoder with logical gates is shown in Fig. 12.9-2*b*. Note that Fig. 12.9-1 only three of the four possible Z_i are being used.

A_0	A_1	Z_0	Z_1	Z_2	Z_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

(*a*)



(*b*)

FIGURE 12.9-2
(*a*) A truth table for a four-word decoder. (*b*) The decoder realized with AND gates.

As a convenience to the user, manufacturers generally provide decoders as an integral adjunct of the ROM encoder. Such a memory device with decoder included, as indicated in Fig. 12.9-3, are generally referred to as a *decoded ROM*. In providing the decoder, the manufacturer does himself a service as well. For on an integrated-circuit chip there is generally a limitation to the number of pins which can conveniently be provided for external connection. And on an M-word memory without decoder, M pins would have to be provided for addressing purposes; while with the decoder N pins, with $2^N = M$, are adequate.

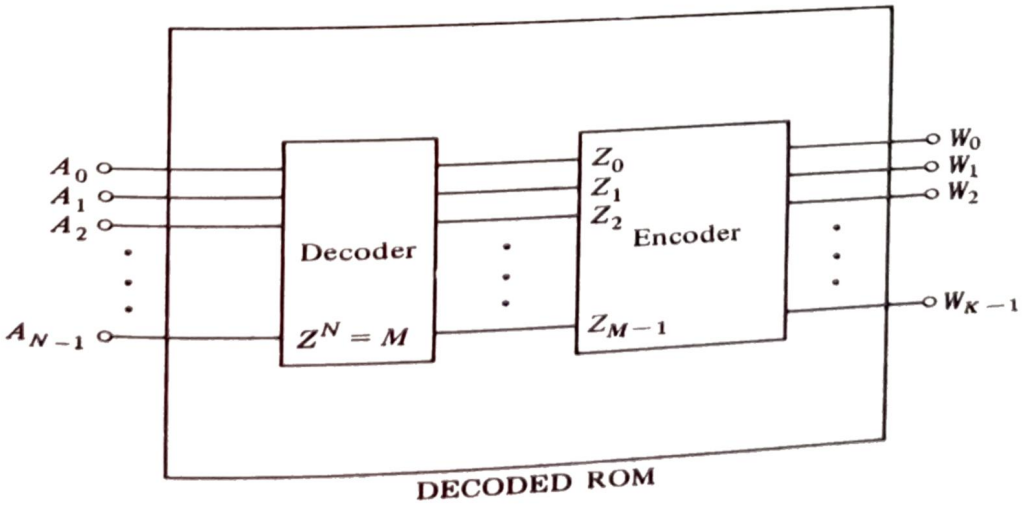


FIGURE 12.9-3
A ROM with decoder included.

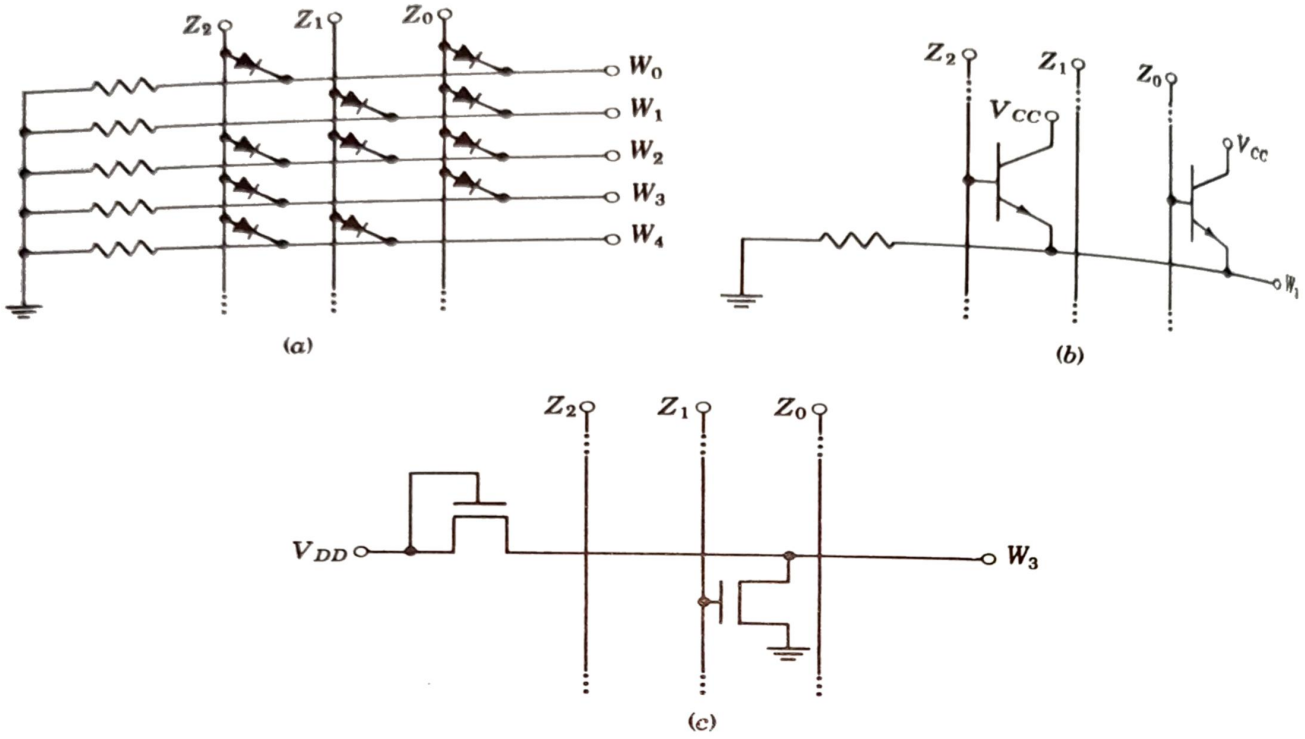


FIGURE 12.10-1
(*a*) A three-word, 5-bit ROM using diode connections between address and bit lines. (*b*) Single-bit line of the ROM in (*a*) illustrating the use of bipolar transistors in place of diodes. (*c*) The use of MOS transistors illustrated.

12.10 IMPLEMENTATION OF ROMs

A possible implementation of the ROM OR-gate encoder shown in Fig. 12.9-1*b* is shown in Fig. 12.10-1*a*. It can be verified (Prob. 12.9-2) that the circuit shown does indeed constitute precisely the OR-gate logic structure. The use of diodes, however, has a drawback in that the inputs to the encoder must directly supply the current which must be delivered at the outputs. This difficulty is relieved by using bipolar transistors in place of diodes in the manner shown in Fig. 12.10-1*b*. Here, for simplicity, we have indicated only the connections to bit output line W_3. Now the output current may be supplied principally by the V_{CC} source. Since, as we have noted, in bipolar technology, integrated-circuit diodes are generally transistors with collector and base terminal joined, fabrication in the manner of Fig. 12.10-1*b* is not substantially more complicated than the pattern in Fig. 12.10-1*a*. It is to be observed further that all the transistors connected to an input line Z_i will have their collector terminals connected in common to V_{CC} and their bases connected in common to the Z_i line. Hence such a vertical array of transistors is not infrequently replaced by a single transistor with multiple emitters. ROMs employing bipolar technology may have access times as low as 15 ns. The *access time* of a memory is the length of time that must elapse

after addressing before the addressed word is made available at the output of the memory.

The number of bits in a memory is defined to be equal to the product of the number of words and the number of bits per word. (In a ROM like that in Fig. 12.10-1*a* the number of diodes will generally be about one-half the number of bits, since we may expect **1**s and **0**s to occur with about equal frequency.) When the number of bits is large, say in excess of 1,000, MOS technology is preferred to bipolar technology since MOS devices are more economical of silicon-die real estate. The implementation of a single-bit line (again, for case of comparison, the $W3$ output bit line) is shown in Fig. 12.10-1*c*. Assuming positive logic as usual and assuming NMOS, we see that $W3$ is at logic **0** only when $Z_1 = 1$, as required by the truth table of Fig. 12.9-1*a*. Access times using MOS devices are typically about 400 ns.

12.11 PROGRAMMABLE AND ERASABLE ROMs

In a ROM, the address lines and the output word bit lines form a crossed array of lines, i.e., a grid structure. At each grid intersection is placed a device (diode, bipolar, or MOS transistor) or not, depending on whether the corresponding word bit is to be **1** or **0**. (In cases where there is no special interest in the type of device, the coupling between address line and bit line is often shown simply by a dot at the grid intersection.) In a programmable ROM (PROM) the manufacturer locates a connecting device at every grid intersection. However, in series with each such device there is provided a fusible link. Any particular fusible link is located at the intersection of some line Z_i and some line W_j. By making connection to Z_i and W_j and passing an adequately large current through the link, the link can be burned out. Thus, the user of such a PROM may burn out links as necessary, leaving transistors only on locations required to establish the memory storage desired.

One type of *erasable* or *alterable* ROM uses *floating-gate* PMOS transistors. These are transistors in which at normal operating voltage the gate is entirely insulated and isolated from electrical connection to any other part of the integrated-circuit chip. It turns out to be possible to establish a negative charge on these gates by the application of a high voltage between source and drain. The negative charge left on the gate by such treatment leaves the corresponding transistor with a conducting channel. The ROM can be erased by exposure to ultraviolet light, which serves to discharge any charged gate.

12.12 APPLICATIONS OF ROMs

A ROM multiplier Consider that we want to perform the arithmetic operation of multiplication. As we have seen in Sec. 11.16, multiplication can be performed by a sequence of shifting operations, i.e., multiplying by powers of 2, and a sequence of additions. On the other hand, we may view a multiplication table

as a truth table. Thus, the entry in the multiplication table to effect the multiplication $11 \times 10 = 0110$, that is, $3 \times 2 = 6$, may be read to mean that a memory addressed by the input code $A_1A_0B_1B_0 = 1110$ reads out the word $P_3P_2P_1P_0 = 0110$.

When the multiplicands have many bits, straightforward multiplication by a ROM may get out of hand. For example, when the multiplicands each consist of 8 bits, the product contains 16 bits and the number of bits which need to be stored in the memory is 16 bits/word $\times 2^{16}$ words $\approx 10^6$. A saving in bit storage can generally be accomplished by using a number of ROMs in a manner now to be illustrated.

Let us consider that we want to multiply two 4-bit numbers $A_3A_2A_1A_0$ and $B_3B_2B_1B_0$. We write

$$A_3A_2A_1A_0 = A_3A_200 + 00A_1A_0 \tag{12.12-1a}$$

and

$$B_3B_2B_1B_0 = B_3B_200 + 00B_1B_0 \tag{12.12-1b}$$

the product is then

$$\begin{aligned} P = {} & (A_3A_200)(B_3B_200) + (00A_1A_0)(B_3B_200) \\ & + (A_3A_200)(00B_1B_0) + (00A_1A_0)(00B_1B_0) \end{aligned} \tag{12.12-2}$$

Note that the product P appears as the sum of four products. Four ROM multipliers are required, as indicated in Fig. 12.12-1. In these multiplications, however, the 0s in Eq. (12.12-1) are ignored. Thus, to effect the last product in Eq. (12.12-2) we use a ROM which accepts as input the address $A_1A_0B_1B_0$ and yields one of 16 possible 4-bit output product words. We must now sum the outputs of the four multipliers, but in so doing we must take account of the differing numerical significance of the output bits of the different multipliers. As can be verified, the numerical significances are as given in the figure. Only one ROM yields bits of significance 2^1 and 2^0. These are brought out and become bits of the final product. Three ROMs provide bits of significance 2^2 and 2^3, and these are combined in two 2-bit adders. Similarly three ROMs provide bits of significance 2^4 and 2^5, and these are combined in two other 2-bit adders. It is left as a student exercise (Prob. 12.12-2) to show that the disposition of carry outputs from the various adders is proper to assure a correct product result.

Each of the ROMs in Fig. 12.12-1 stores 16×4 bits. The four ROMs store $16 \times 4 \times 4$ bits $= 2^8 = 256$ bits. If the multiplication had been effected by a single ROM, that ROM would need storage for 2^8 words $\times$ 8 bits/word $= 2^8 \times 8$ bits. Hence the present scheme reduces the required storage capacity by a factor of 8. The saving is more impressive as the number of bits increases. When the factors to be multiplied each have 8 bits, there is a saving by a factor of $2^7 = 128$. Of course, some of this saving is dissipated on account of the need to provide five adders. In addition, the use of adders, even with a look-ahead carry feature, results in a significant increase in the time required to perform the multiplication.

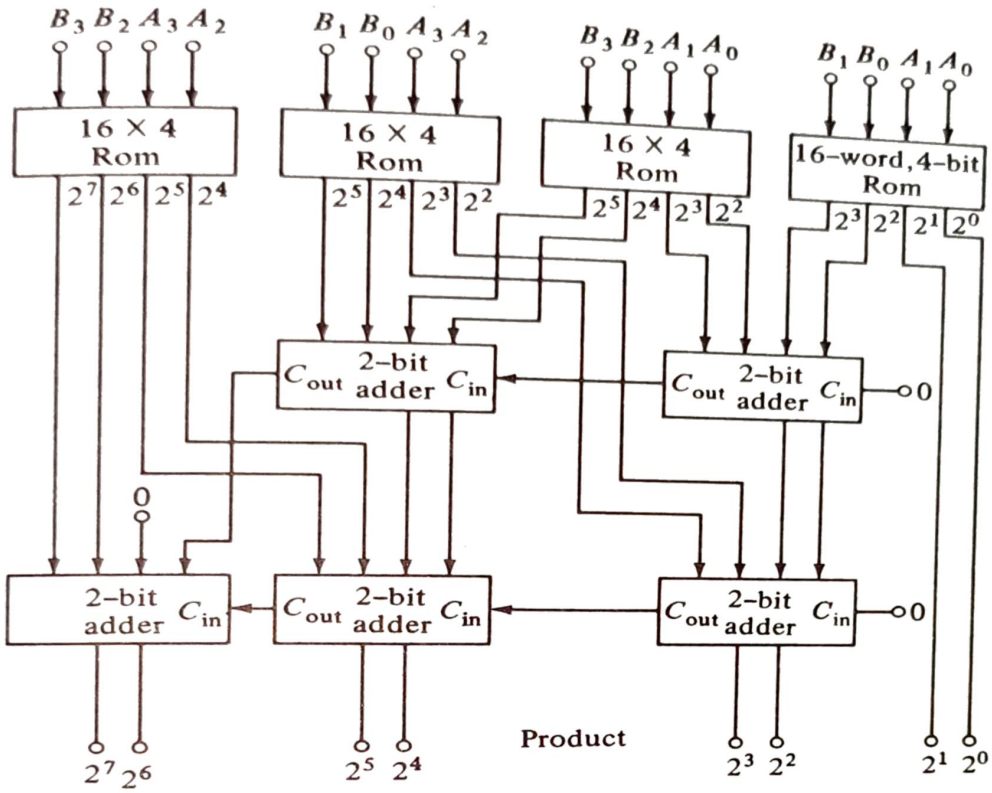


FIGURE 12.12-1
A ROM multiplier for two 4-bit numbers, illustrating a technique for reducing required storage capacity.

It is often required that a ROM multiplier produce an output having the same number of bits as each of the multiplicands. Thus, in the above example, where the inputs were each 4-bit words, the output would also be a 4-bit word. In this case, referring to Fig. 12.12-1, the output bits would consist of the four most significant bits. In this case it is possible to keep the maximum error between ± 8 by using additional circuitry (Prob. 12.12-1). If an error of 15 is permitted, the complexity of Fig. 12.12-1 can be reduced significantly to include only three ROMs and three adders.

The ROM as a *look-up* table ROMs are also widely used as look-up tables for mathematical functions such as logarithms, trigonometric functions, square roots, exponentials, etc. Techniques similar to that employed with the multiplier can be used effectively to reduce the storage capacity required of the memory. For example, consider a ROM for $\sin \theta$. If we wanted to provide for a resolution of 0.01°, then for the range $0 \leq \theta \leq 90°$, a 9,000-word memory would be required. We may, however, write $\theta = I + F$, where I is the integral part of θ and F is the fractional part. Then

$$\sin \theta = \sin (I + F) = \sin I \cos F + \cos I \sin F \qquad (12.12\text{-}3)$$

side of the transistors so that the common emitters can be returned to a constant-current source (possibly a large resistor). In this way the cell can be operated in the manner of emitter-coupled logic, where transistors $T1$ and $T2$ are not allowed to saturate.

12.15 MOS RAMs

A six-transistor static cell A six-transistor static MOS memory cell is shown in Fig. 12.15-1*a*. A bit is stored in the flip-flop, which consists of two cross-coupled inverters. Transistors $T1$ and $T2$ are the driver and load of one inverter, and $T3$ and $T4$ serve correspondingly for the other inverter.

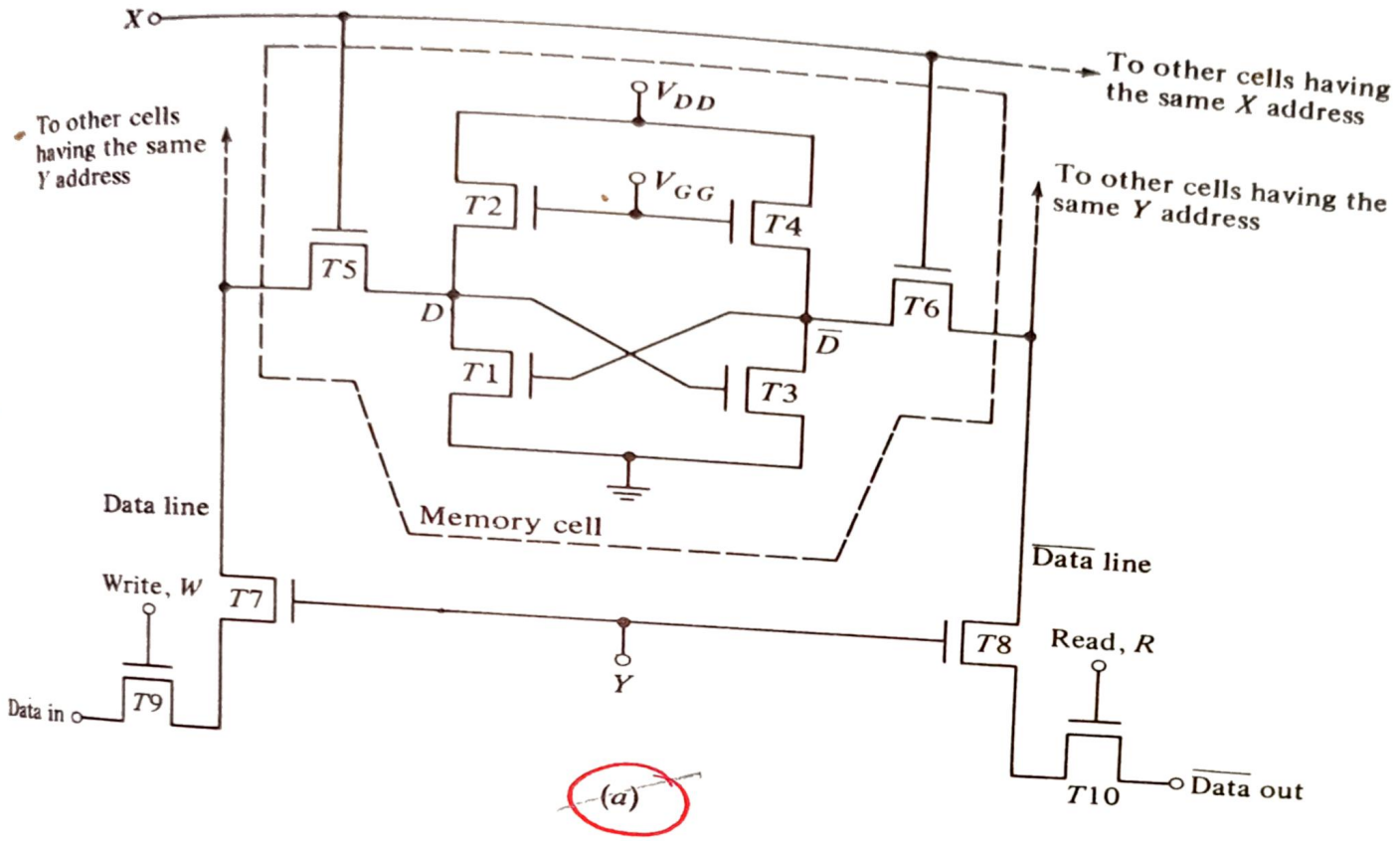
The memory cell shown is addressed by setting X and Y to logic **1**. Setting $X = 1$ connects the cell to the data line and the $\overline{\text{data}}$ line. To write into the cell we set $W = 1$. This connects the data input terminal to node D as $T5$, $T7$, and $T9$ are ON. If the data input is at logic **1**, this raises the gate of $T3$, turning it ON and making node $\bar{D} = 0$. If the data input is at logic **0**, then $T3$ would be turned OFF and $\bar{D}$ would be at **1**. To read the state of the flip-flop we set $R = 1$. This connects the $\overline{\text{data}}$ output terminal to $\bar{D}$ since now $T6$, $T8$, and $T10$ are ON. Thus, the complement of the data level written into the cell is read.

In general, in a RAM, there are many memory cells connected to the same input and output lines. Such a configuration is shown in Fig. 12.15-1*b*. In this figure there are $\alpha\beta$ memory cells, each containing six transistors. Note that there are β columns of cells, each with a different Y address, and α rows of cells, each with a different X address. There is however, a single input transistor $T9$ to connect the data input terminal to the selected memory cell when the *write* instruction is given and a single output transistor $T10$ to connect the selected memory cell to the $\overline{\text{data}}$ output terminal during the *read* operation.

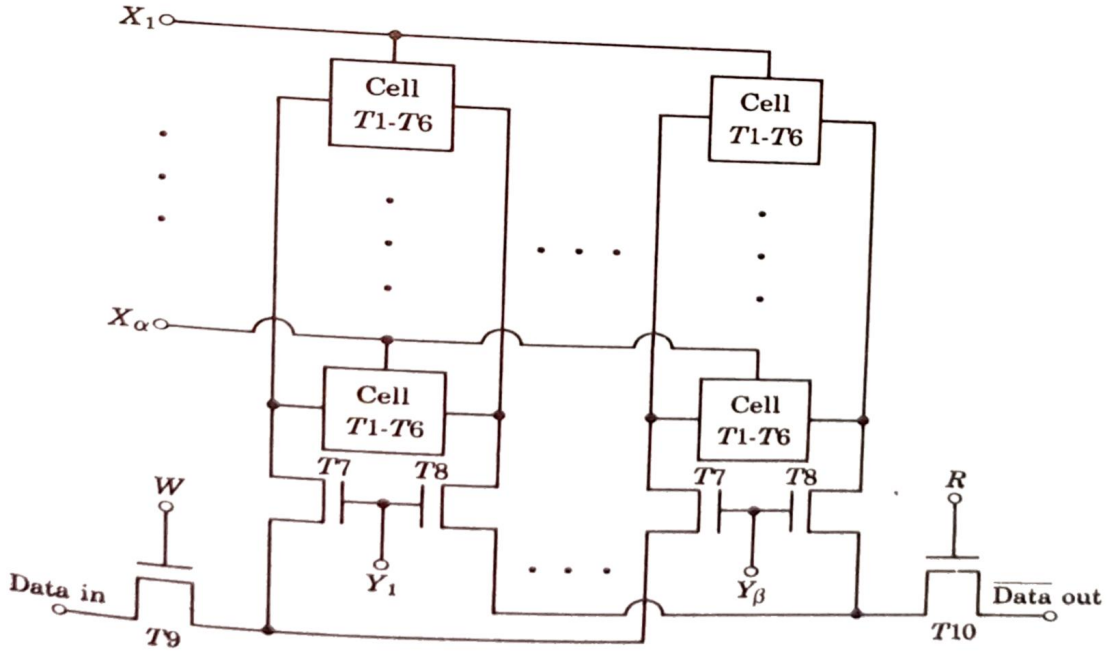
A four-transistor dynamic cell In the dynamic memory cell of Fig. 12.15-2 the transistor count is reduced from six to four with a corresponding saving of real estate on the silicon chip and a saving of power. The cell is composed of transistors $T1$, $T2$, $T5$, and $T6$. Transistors $T7$ and $T8$ as well as transistors $T11$ and $T12$ serve all cells having the same column Y address. Transistors $T9$ and $T10$ are common to all of the cells in the memory, as in Fig. 12.15-1*b*.

The state of the cell is stored on the stray capacitances C_1 and C_2, whose presence is essential. These capacitors become accessible to the data terminals when the transmission gates $T7$ and $T8$ as well as $T5$ and $T6$ are all made to conduct by simultaneously raising the X and Y addresses to logic **1**.

In one state of the cell, the voltage across C_1 is larger than the threshold voltage of $T1$, and $T1$ is ON. Correspondingly C_2 has zero voltage, and $T2$ is OFF. In the other state the voltages of C_1 and C_2 and the conducting states of $T1$ and $T2$ are reversed. The cell having been accessed, the state of the cell can be *read* by setting $R = 1$. We can *write* into the cell by setting $W = 1$.



(a)



(b)

FIGURE 12.15-1
(a) A static MOS memory cell. (b) The interconnection of cells to form a RAM.

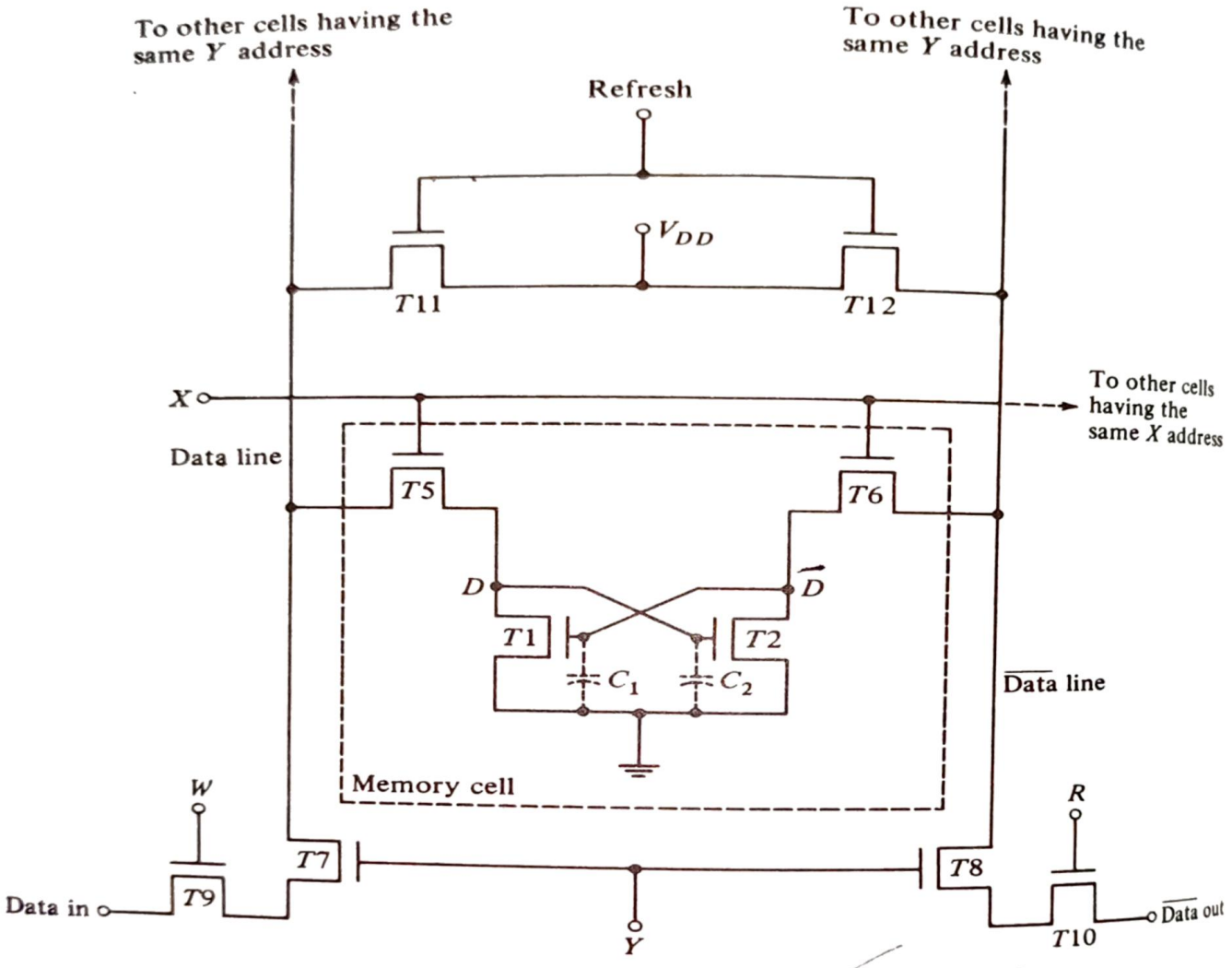


FIGURE 12.15-2
A four-transistor MOS dynamic cell.

If we have not performed a write operation for an extended time, then because of leakage of capacitor charge the information in the cell may be lost. It is therefore necessary to *refresh* the cell periodically. This refreshing operation is accomplished by allowing brief access from the supply voltage V_{DD} to the cell. This access becomes available when the X address and the refresh terminal are simultaneously at logic 1 so that $T5$ and $T6$ as well as $T11$ and $T12$ are thereby turned ON. Suppose now that initially $T1$ is ON, $T2$ is OFF, the voltage across C_1 is $V_{C_1} > V_T$, the threshold voltage, and $V_{C_2} = 0$ V. During the refresh interval V_{DD} is applied through $T12$ and $T6$ to C_1 paralleled by $T2$. However, $T2$ is OFF, and hence all the current from V_{DD} will be directed into C_1, allowing C_1 to replenish any charge lost due to leakage. Similarly V_{DD} is applied to C_2 which is in parallel to $T1$. But $T1$ is ON, and hence C_2 will not charge as rapidly as C_1. Observe that during the refresh interval $T6$ and $T12$ become a load for the driver transistor $T2$ and that $T5$ and $T11$ become a load for $T1$.

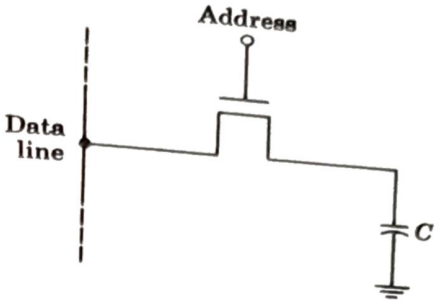


FIGURE 12.15-3
A single-capacitor memory.

Hence, during this refresh interval the cell becomes a conventional flip-flop consisting of two cross-coupled inverters. In any event it is clear that whatever the initial state of the flip-flop, during the refresh interval this initial state is reinforced.

A three-transistor memory cell It is apparent that if a bit is to be preserved through the charge stored on capacitors (as in Fig. 12.15-2), then a single capacitor is enough. Such a rudimentary memory is indicated in Fig. 12.15-3. Of course, additional circuitry will be required to allow reading, writing, and refreshing, but each component of this extra circuitry will be able to service many memory cells. We are concerned principally with the components which must be duplicated in every cell.

An important difficulty associated with the simple one-transistor cell of Fig. 12.15-3 is that during a read operation the storage capacitor C will discharge into the data line. Unless the storage capacitance is large in comparison with the data-line capacitance, the read operation may well be destructive. That is, the process of reading a bit may well erase the bit from the memory. Suppose, on the other hand, that C is made quite large. Then during a write operation a large capacitor needs to be charged, and the interval which needs to be devoted to a write operation will thereby be extended.

One resolution of the difficulty is presented in Fig. 12.15-4, where we have provided separate access paths to the capacitor C for writing and reading. However, now a three-transistor memory cell is required. In this cell, transistor $T1$ is used during the write instruction, while transistors $T2$ and $T3$ are used when the logic state of capacitor C is to be read. Since the capacitor is the gate capacitor of $T2$, it is isolated from the output data line. A refresh circuit is also needed, of course, inasmuch as the charge on capacitor C can leak off through $T1$. Refreshing is accomplished using an inverter consisting of transmission gate $T9$ and the inverter amplifier formed by capacitor C_r and transistors $T10$ and $T11$. Note the similarity between this refresh amplifier and the inverter shown in Fig. 12.3-4, consisting of $T1$, $T2$, $T3$, and C_1.

To access the cell we set X and Y to **1**. As before, there are many cells having the same Y address. To write into the cell we disconnect the refresh circuit by setting $P = \mathbf{0}$. We then set $W = \mathbf{1}$. This connects the data-input

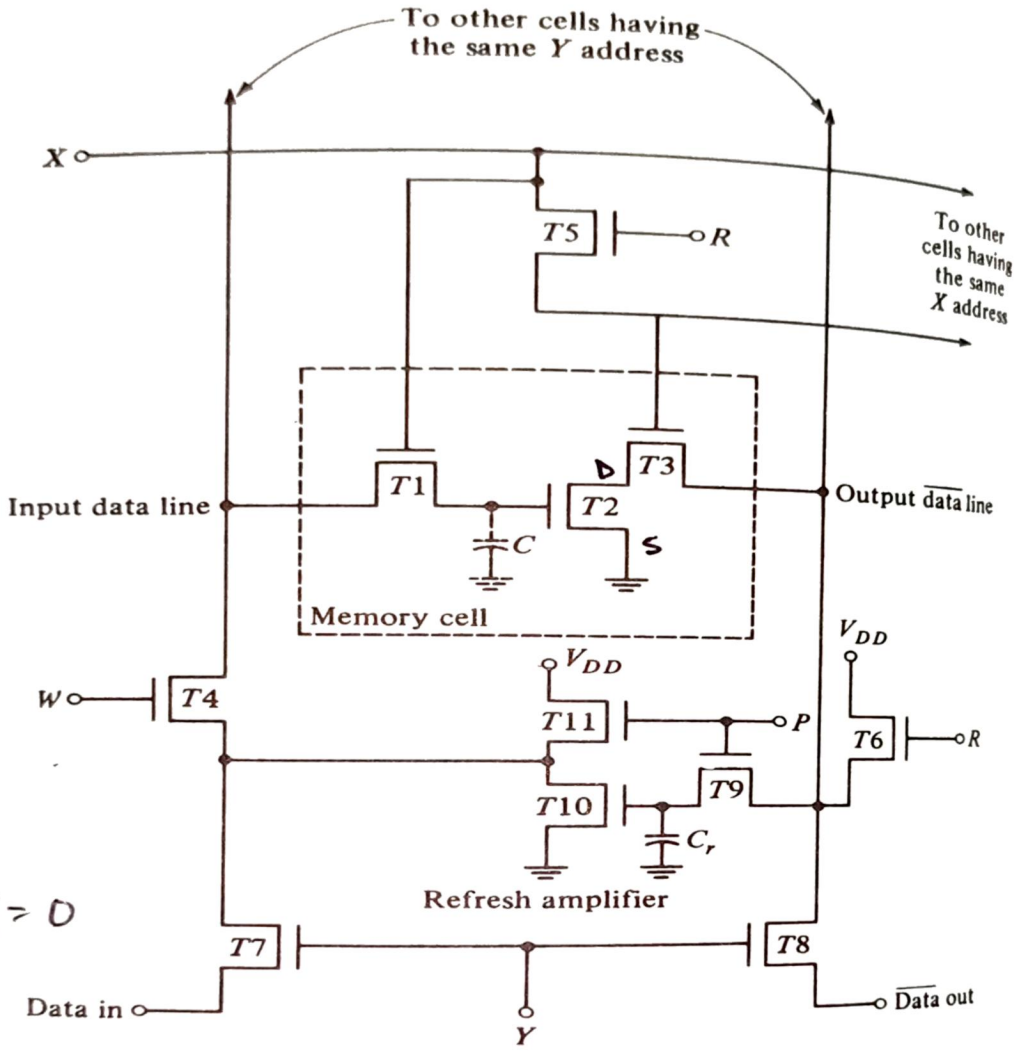


FIGURE 12.15-4
A three-transistor dynamic memory cell.

terminal across capacitor C since $T7$, $T4$, and $T1$ are ON. Capacitor C then charges to the state of the data input. To read the cell we set $R = 1$ ($W = 0$), which turns $T6$ and $T3$ of the addressed cell ON. The data-output line is then connected to the drain of $T2$ since $T3$ and $T8$ are ON. The complement of the level stored on C is read at the data-output terminal. Note that transistor $T6$ acts as a load for $T2$ during the read operation so that $T2$ and $T6$ form an inverting amplifier.

To refresh the cell we set $Y = 0$, $X = 1$, $P = 1$, and $R = 1$. The data-input and the data-output terminals are now disconnected from all the cells in the memory. The complement of the logic level of capacitor C is now transferred through $T9$, and this level is stored on capacitor C_r. The input terminal P, which permits the connection of C_r to the output data line, is called the *precharge input*. This term is used because when $P = 1$, C_r precharges to the complement

of the level on C. After this precharging is accomplished, R is set to **0** and W is then set to **1**. The output of the inverter, $T10$ and $T11$, then refreshes the charge on capacitor C. Note that initially the refresh-amplifier output is disconnected from $T1$ and capacitor C. This is to ensure that $T11$ does not initially load capacitor C until after C_r has been precharged to the correct level. If this precaution were not taken, capacitor C might be discharged erroneously.

The memory cells having the same Y address are often refreshed sequentially, the cell having address X_1 being refreshed first, then the cell with address X_2, etc. For each column of cells there is a single refresh circuit; thus if there are β columns in the RAM (see Fig. 12.15-1*b*), there are β refresh circuits. The β cells having the same X address are refreshed simultaneously. Thus, if there are α rows, and if it takes a time T_r to refresh a single cell, the entire memory is refreshed in the time αT_r.

Note that although there are only three transistors in the memory cell shown in Fig. 12.15-4, many auxiliary transistors are needed to implement the reading, writing, and refreshing operations. However, these transistors are shared by other memory cells.

12.16 ORGANIZATION OF A RAM

The organization of a memory with storage facility for M words each of 3 bits is shown in Fig. 12.16-1. Cell (1, 1) stores the first bit of the first word, cell (1, 2) the second bit and so on. These first word-storage cells are addressed by raising the level of line Z_1. If the memory stores M words, M address lines Z_1, $Z_2, \ldots, Z_M$ are required. The memory-storage location is presented here in coded form through the λ address bits $A_0 A_1 \cdots A_{\lambda-1}$ (where $M = 2^\lambda$). As required, then, a decoder has been interposed between the memory cells and the coded input address. As discussed in Sec. 12.9, such a decoder singles out one and only one of the address lines, the line so selected depending on the input address. Not all commercially available memories provide such decoding, and in such cases the decoders must be provided by the user. When a decoder is already part of a memory unit, the manufacturer will characterize his product as being "decoded" or "fully decoded."

All the cells ($C_{1,1}, C_{2,1}, \ldots, C_{M,1}$) intended to store first bits of words are connected to a common pair of data lines. All second-bit cells and all third-bit cells are similarly connected. The block marked I/O (input-output) represents all the circuitry shown in Fig. 12.13-1 or 12.15-4 with the exception of the memory-cell flip-flop itself. The bits b_{i1}, b_{i2}, and b_{i3} are input bits which will be written into the memory when the read/write (RW) line goes to logic **1**. The bits b_{o1}, b_{o2}, and b_{o3} are the output bits which will be read out of the memory when $RW = \mathbf{0}$.

If the memory cell has two input address terminals, as in Figs, 12.13-1 and 12.15-4, we can arrange to make each cell separately and individually accessible. In such a case a memory with M cells can be used to provide

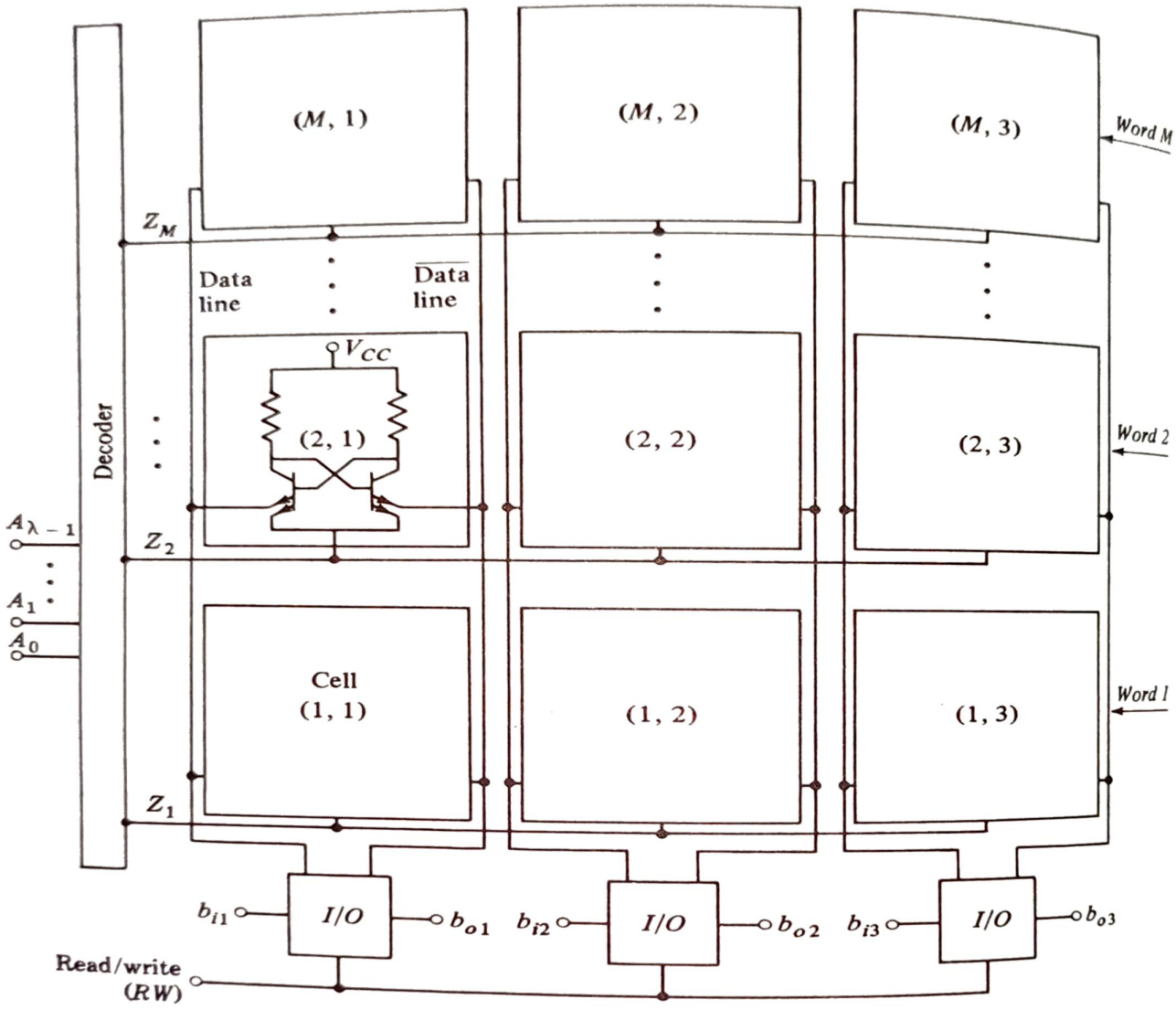


FIGURE 12.16-1
The organization of a M-word 3-bit/word memory.

storage for M words, each word having just 1 bit. The organization of such a memory is shown in Fig. 12.16-2. Note particularly that two decoders are required. At any time only one output line of the X decoder will be selected and only one line of the Y decoder will be selected. Correspondingly only that cell will be selected (addressed) which is at the intersection of these two selected lines. All the data lines of all the cells are paralleled, and only a single input-output stage is required to handle the 1-bit word.

This present arrangement allows the use of simpler decoders since some of the decoding is done in the memory cell itself. For example, suppose we need a memory with 256 (2^8) words each of 1 bit. If the cells had single address lines, we would need a decoder with 256 output lines. Suppose, however, the

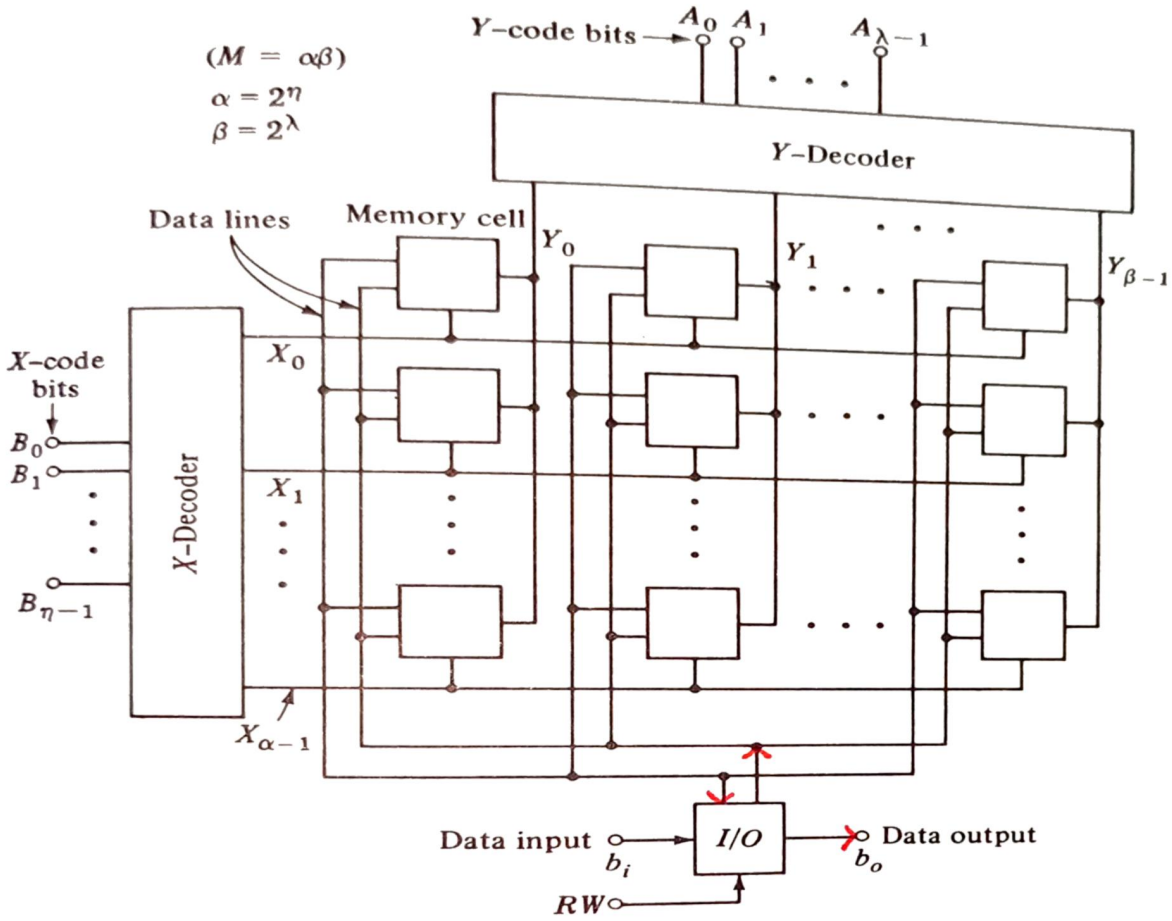


FIGURE 12.16-2
Organization of a memory with N 1-bit words.

cells were arranged, as in Fig. 12.16-2, in a square array ($\alpha = \beta = 16$). Then we should require two decoders, but each decoder would have only 4 input lines and 16 output lines.

If the memory cells employed in the RAM shown in Fig. 12.16-2 are dynamic, each column of cells has its own refresh circuit, as in Fig. 12.15-4. Thus, in a 1,024-bit memory containing 32 columns, 32 refresh circuits are used.

12.17 PARALLELING OF SEMICONDUCTOR MEMORY INTEGRATED-CIRCUIT CHIPS

On an integrated-circuit memory chip, the number of bits in a word is equal to the number of memory cells that have a common address line, and the number of words is equal to the number of separately addressable groups of bits. Manufacturers generally specify *word* and *bits per word* storage capacity through