Unit IV

Memory and Programmable Logic: RAM, Types of Memories, Memory decoding, ROM, Types of ROM, Programmable Logic Devices (PLDs): Basic concepts, PROM as PLD, Programmable Array Logic (PAL) and Programmable Logic Array (PLA), Design of combinational and sequential circuits using PLDs.

Introduction

- A **memory unit** is a device to which binary information is transferred for storage and from which information is retrieved when needed for processing. When data processing takes place, information from memory is transferred to selected registers in the processing unit. Intermediate and final results obtained in the processing unit are transferred back to be stored in memory. Binary information received from an input device is stored in memory, and information transferred to an output device is taken from memory.
- A memory unit is a collection of cells capable of storing a large quantity of binary information. There are two types of memories that are used in digital systems: ***random-access memory*** **(RAM) and** ***read-only memory*** **(ROM)**.
- **RAM** stores new information for later use. The process of storing new information into memory is referred to as a memory ***write*** **operation**. The process of transferring the stored information out of memory is referred to as a memory ***read*** **operation**. RAM can perform both write and read operations.
- **ROM** can perform only the read operation. This means that suitable binary information is already stored inside memory and can be retrieved or read at any time. However, that information cannot be altered by writing.

Introduction

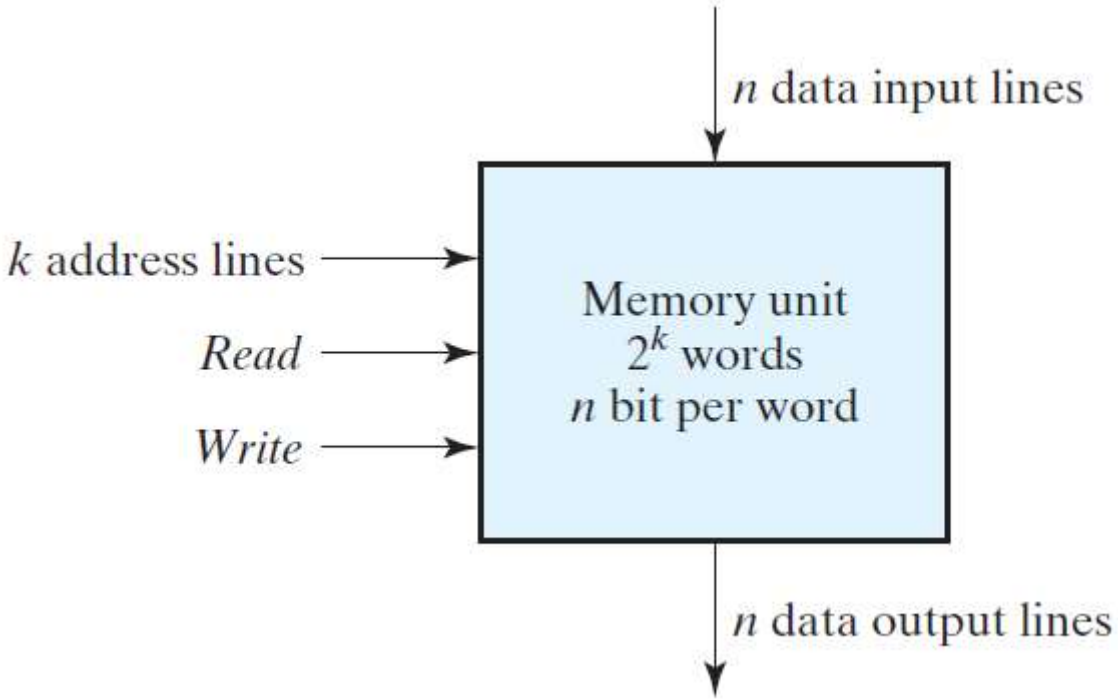
- ROM is a *programmable logic device* (PLD). The binary information that is stored within such a device is specified in some fashion and then embedded within the hardware in a process is referred to as *programming* the device.
- Programming refers to a hardware procedure which specifies the bits that are inserted into the hardware configuration of the device.
- ROM is one example of a PLD. Other such units are the programmable logic array (PLA), programmable array logic (PAL), and the field-programmable gate array (FPGA). A PLD is an integrated circuit with internal logic gates connected through electronic paths that behave similarly to fuses. In the original state of the device, all the fuses are intact. Programming the device involves blowing those fuses along the paths that must be removed in order to obtain the particular configuration of the desired logic function.

RANDOM-ACCESS MEMORY

- A memory unit is a collection of storage cells, together with associated circuits needed to transfer information into and out of a device. The architecture of memory is such that information can be selectively retrieved from any of its internal locations. The time it takes to transfer information to or from any desired random location is always the same—hence the name *random-access memory,* abbreviated RAM.
- In contrast, the time required to retrieve information that is stored on magnetic tape depends on the location of the data.
- A memory unit stores binary information in groups of bits called *words* . A word in memory is an entity of bits that move in and out of storage as a unit. A memory word is a group of 1's and 0's and may represent a number, an instruction, one or more alphanumeric characters, or any other binary-coded information.
- A group of 8 bits is called a *byte* . Most computer memories use words that are multiples of 8 bits in length. Thus, a 16-bit word contains two bytes, and a 32-bit word is made up of four bytes. The capacity of a memory unit is usually stated as the total number of bytes that the unit can store.

RANDOM-ACCESS MEMORY

★ The *n* data input lines provide the information to be stored in memory, and the *n* data output lines supply the information coming out of memory. The *k* address lines specify the particular word chosen among the many available. The two control inputs specify the direction of transfer desired: The *Write* input causes binary data to be transferred into the memory, and the *Read* input causes binary data to be transferred out of memory.



RANDOM-ACCESS MEMORY

- The memory unit is specified by the number of words it contains and the number of bits in each word. The address lines select one particular word. Each word in memory is assigned an identification number, called an *address,* starting from 0 up to 2*k* - 1, where *k* is the number of address lines. The selection of a specific word inside memory is done by applying the *k* -bit address to the address lines. An internal decoder accepts this address and opens the paths needed to select the word specified. Memories vary greatly in size and may range from 1,024 words, requiring an address of 10 bits, to 232 words, requiring 32 address bits. It is customary to refer to the number of words (or bytes) in memory with one of the letters K (kilo), M (mega), and G (giga). K is equal to 210, M is equal to 220, and G is equal to 230. Thus, 64K = 216, 2M = 221, and 4G = 232.

Write and Read Operations

- The two operations that RAM can perform are the write and read operations. As alluded to earlier, the write signal specifies a transfer-in operation and the read signal specifies a transfer-out operation. On accepting one of these control signals, the internal circuits inside the memory provide the desired operation.
- The steps that must be taken for the purpose of transferring a new word to be stored into memory are as follows:

  1. Apply the binary address of the desired word to the address lines.

  2. Apply the data bits that must be stored in memory to the data input lines.

  3. Activate the *write* input.

- The memory unit will then take the bits from the input data lines and store them in the word specified by the address lines.
- The steps that must be taken for the purpose of transferring a stored word out of memory are as follows:

  1. Apply the binary address of the desired word to the address lines.

  2. Activate the *read* input.

Types of Memory

- The mode of access of a memory system is determined by the type of components used. In a random-access memory, the word locations may be thought of as being separated in space, each word occupying one particular location. In a sequential-access memory, the information stored in some medium is not immediately accessible, but is available only at certain intervals of time. A magnetic disk or tape unit is of this type.
- Each memory location passes the read and write heads in turn, but information is read out only when the requested word has been reached. In a random-access memory, the access time is always the same regardless of the particular location of the word.
- In a sequential access memory, the time it takes to access a word depends on the position of the word with respect to the position of the read head; therefore, the access time is variable.
- Integrated circuit RAM units are available in two operating modes: *static* and *dynamic*.
- **Static RAM (SRAM)** consists essentially of internal latches that store the binary information. The stored information remains **valid as long as power is applied to the unit**.
- **Dynamic RAM (DRAM)** stores the binary information in the form of electric charges on capacitors provided inside the chip by MOS transistors. The stored

- DRAM offers reduced power consumption and larger storage capacity in a single memory chip. SRAM is easier to use and has shorter read and write cycles.
- Memory units that lose stored information when power is turned off are said to be ***volatile*****.** CMOS integrated circuit RAMs, both static and dynamic, are of this category, since the binary cells need external power to maintain the stored information.
- In contrast, **a nonvolatile memory**, such as magnetic disk, retains its stored information after the removal of power. This type of memory is able to retain information because the data stored on magnetic components are represented by the direction of magnetization, which is retained after power is turned off.
- **ROM is another nonvolatile memory**. A nonvolatile memory enables digital computers to store programs that will be needed again after the computer is turned on. Programs and data that cannot be altered are stored in ROM, while other large programs are maintained on magnetic disks. The latter programs are transferred into the computer RAM as needed. Before the power is turned off, the binary information from the computer RAM is transferred to the disk so that the information will be retained.

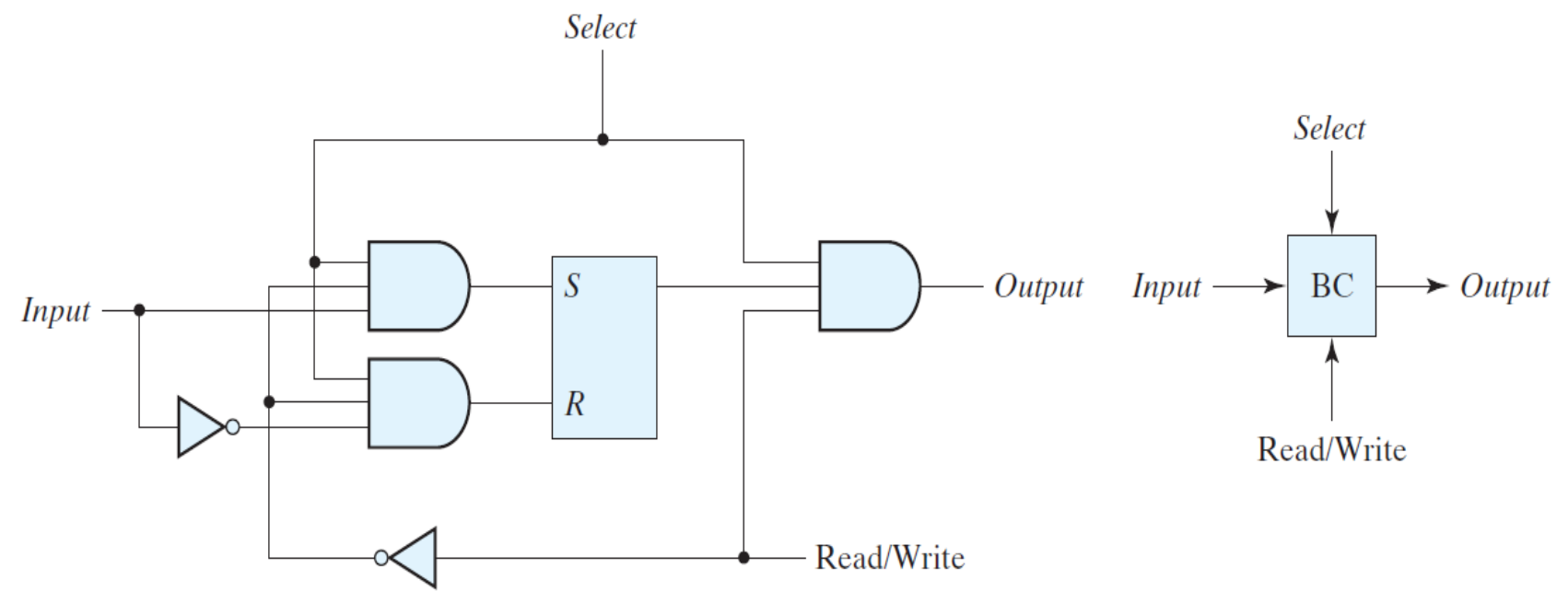
MEMORY DECODING

- To be able to include the entire memory in one diagram, the memory unit presented here has a small capacity of 16 bits, arranged in four words of 4 bits each. An example of a two-dimensional coincident decoding arrangement is presented to show a more efficient decoding scheme that is used in large memories.

Internal Construction

- The internal construction of a RAM of *m* words and *n* bits per word consists of *m* * *n* binary storage cells and associated decoding circuits for selecting individual words. The binary storage cell is the basic building block of a memory unit.
- The storage part of the cell is modeled by an *SR* latch with associated gates to form a *D* latch. Actually, the cell is an electronic circuit with four to six transistors.
- The binary cell stores one bit in its internal latch. **The select input** enables the cell for reading or writing, and **the read/write input** determines the operation of the cell when it is selected.
- **A 1 in the read/write input** provides the read operation by forming a path from the latch to the output terminal. **A 0 in the read/write input** provides the write operation by forming a path from the input terminal to the latch.

Memory cell


Select
Input
S
R
Output
Read/Write
Select
Input
BC
Output
Read/Write

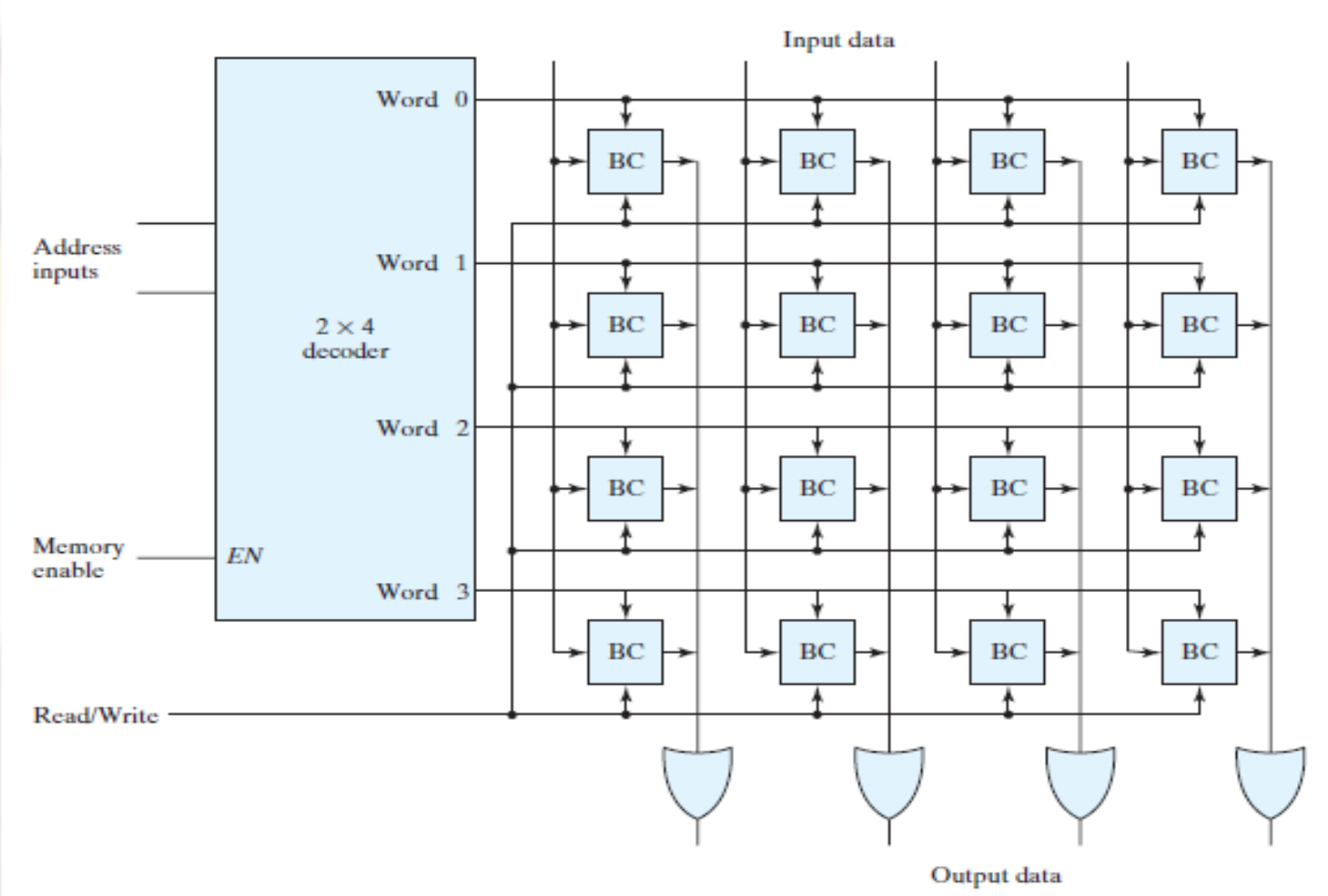


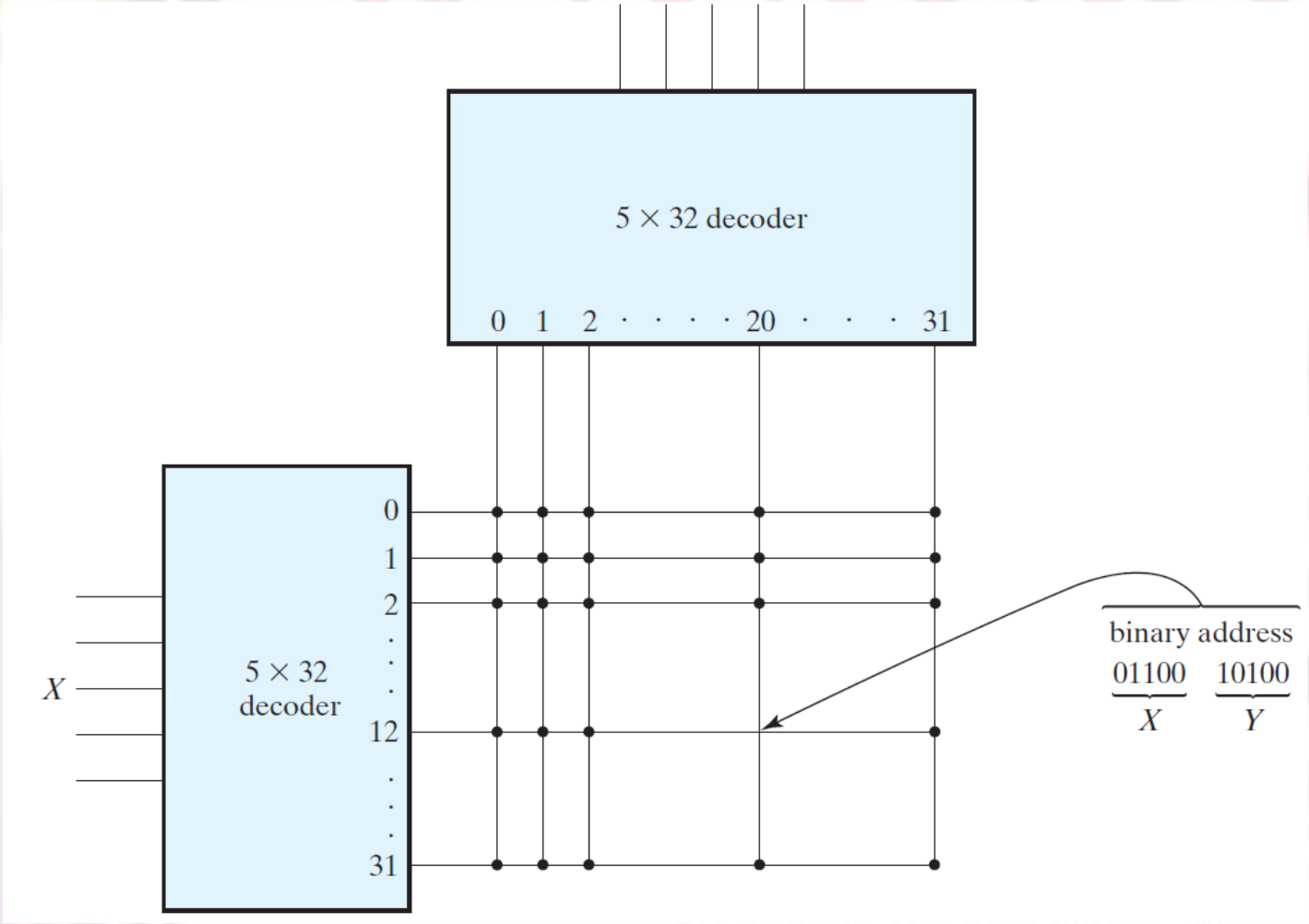
Diagram of a 4 * 4 RAM

- A memory with four words needs two address lines. The two address inputs go through a 2 * 4 decoder to select one of the four words. The decoder is enabled with the memory-enable input. When the memory **enable is 0**, all outputs of the decoder are 0 and none of the memory words are selected.
- With the **memory select at 1**, one of the four words is selected, dictated by the value in the two address lines. Once a word has been selected, the read/write input determines the operation.
- During the read operation, the four bits of the selected word go through OR gates to the output terminals. During the write operation, the data available in the input lines are transferred into the four binary cells of the selected word. The binary cells that are not selected are disabled, and their previous binary values remain unchanged.
- When the memory select input that goes into the decoder is equal to 0, none of the words are selected and the contents of all cells remain unchanged regardless of the value of the read/write input. A memory with $2k$ words of n bits per word requires k address lines that go into a $k * 2k$ decoder. Each one of the decoder outputs selects one word of n bits for reading or writing.

Coincident Decoding

- A decoder with *k* inputs and 2*k* outputs requires 2*k* AND gates with *k* inputs per gate. The total number of gates and the number of inputs per gate can be reduced by employing two decoders in a two-dimensional selection scheme.
- The basic idea in two-dimensional decoding is to arrange the memory cells in an array that is close as possible to square. In this configuration, two *k* /2-input decoders are used instead of one *k* -input decoder. One decoder performs the row selection and the other the column selection in a two-dimensional matrix configuration.
- The two-dimensional selection pattern is demonstrated in Fig. for a 1K-word memory. Instead of using a single 10 * 1,024 decoder, we use two 5 * 32 decoders. With the single decoder, we would need 1,024 AND gates with 10 inputs in each.
- In the two-decoder case, we need 64 AND gates with 5 inputs in each. The five most significant bits of the address go to input *X* and the five least significant bits go to input *Y.* Each word within the memory array is selected by the coincidence of one *X* line and one *Y* line. Thus, each word in memory is selected by the coincidence between 1 of 32 rows and 1 of 32 columns, for a total of 1,024 words. Note that each intersection represents a word that may have any number of bits.
- **As an example**, consider the word whose address is 404. The 10-bit binary equivalent of 404 is 01100 10100. This makes *X* = 01100 (binary 12) and *Y* =

Two-dimensional decoding structure for a 1K-word memory



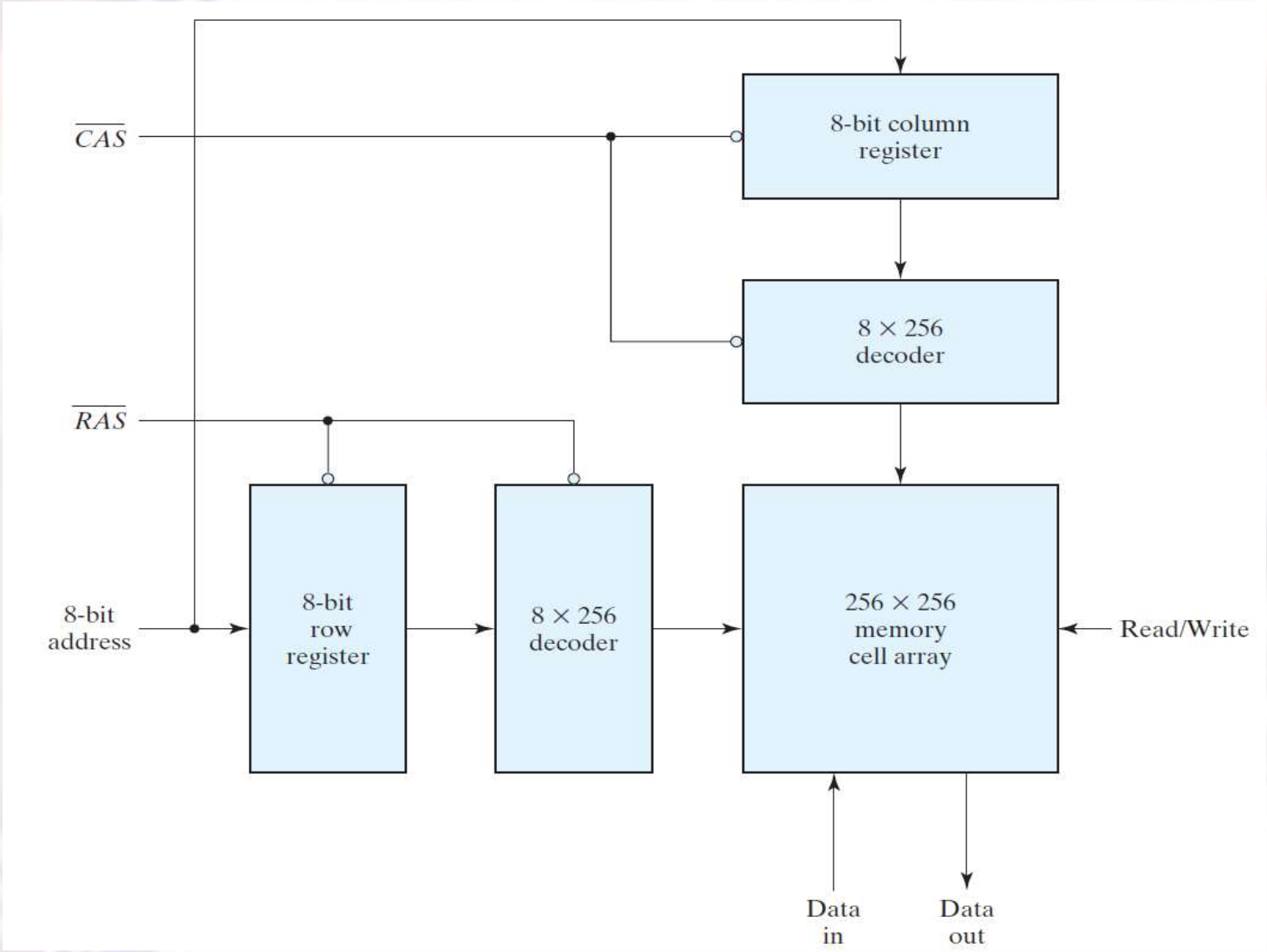
Address Multiplexing

- The SRAM memory cell modeled typically contains six transistors. In order to build memories with higher density, it is necessary to reduce the number of transistors in a cell.
- The DRAM cell contains a single MOS transistor and a capacitor. The charge stored on the capacitor discharges with time, and the memory cells must be periodically recharged by refreshing the memory. Because of their simple cell structure, DRAMs typically have four times the density of SRAMs. This allows four times as much memory capacity to be placed on a given size of chip. The cost per bit of DRAM storage is three to four times less than that of SRAM storage. A further cost savings is realized because of the lower power requirement of DRAM cells.
- Because of their large capacity, the address decoding of DRAMs is arranged in a two-dimensional array, and larger memories often have multiple arrays. To reduce the number of pins in the IC package, designers utilize address multiplexing whereby one set of address input pins accommodates the address components. In a two-dimensional array, the address is applied in two parts at different times, with the row address first and the column address second. Since the same set of pins is used for both parts of the address, the size of the package is decreased significantly.

Address Multiplexing

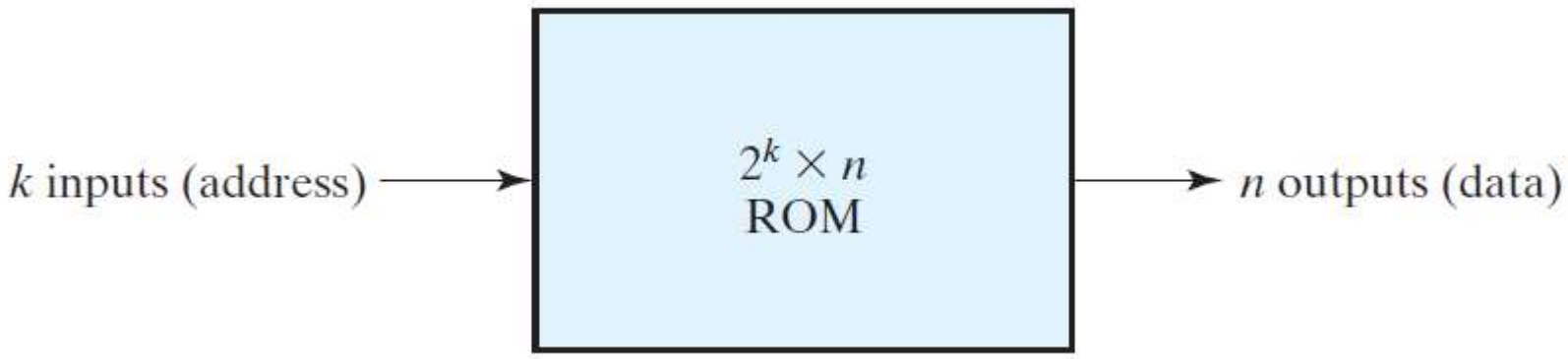
- The memory consists of a two-dimensional array of cells arranged into 256 rows by 256 columns, for a total of $2^8 * 2^8 = 2^{16}$ = 64K words. There is a single data input line, a single data output line, and a read/write control, as well as an eight-bit address input and two address *strobes,* the latter included for enabling the row and column address into their respective registers.
- The row address strobe (RAS) enables the eight-bit row register, and the column address strobe (CAS) enables the eight-bit column register. The bar on top of the name of the strobe symbol indicates that the registers are enabled on the zero level of the signal.
- The 16-bit address is applied to the DRAM in two steps using RAS and CAS. Initially, both strobes are in the 1 state. The 8-bit row address is applied to the address inputs and RAS is changed to 0. This loads the row address into the row address register. RAS also enables the row decoder so that it can decode the row address and select one row of the array. After a time equivalent to the settling time of the row selection, RAS goes back to the 1 level.
- The 8-bit column address is then applied to the address inputs, and CAS is driven to the 0 state. This transfers the column address into the column register and enables the column decoder. Now the two parts of the address are in their respective registers, the decoders have decoded them to select the one cell corresponding to the row and column address, and a read or write operation can be performed on that cell. CAS must go back to the 1 level before initiating

Address multiplexing for a 64K DRAM



READ-ONLY MEMORY

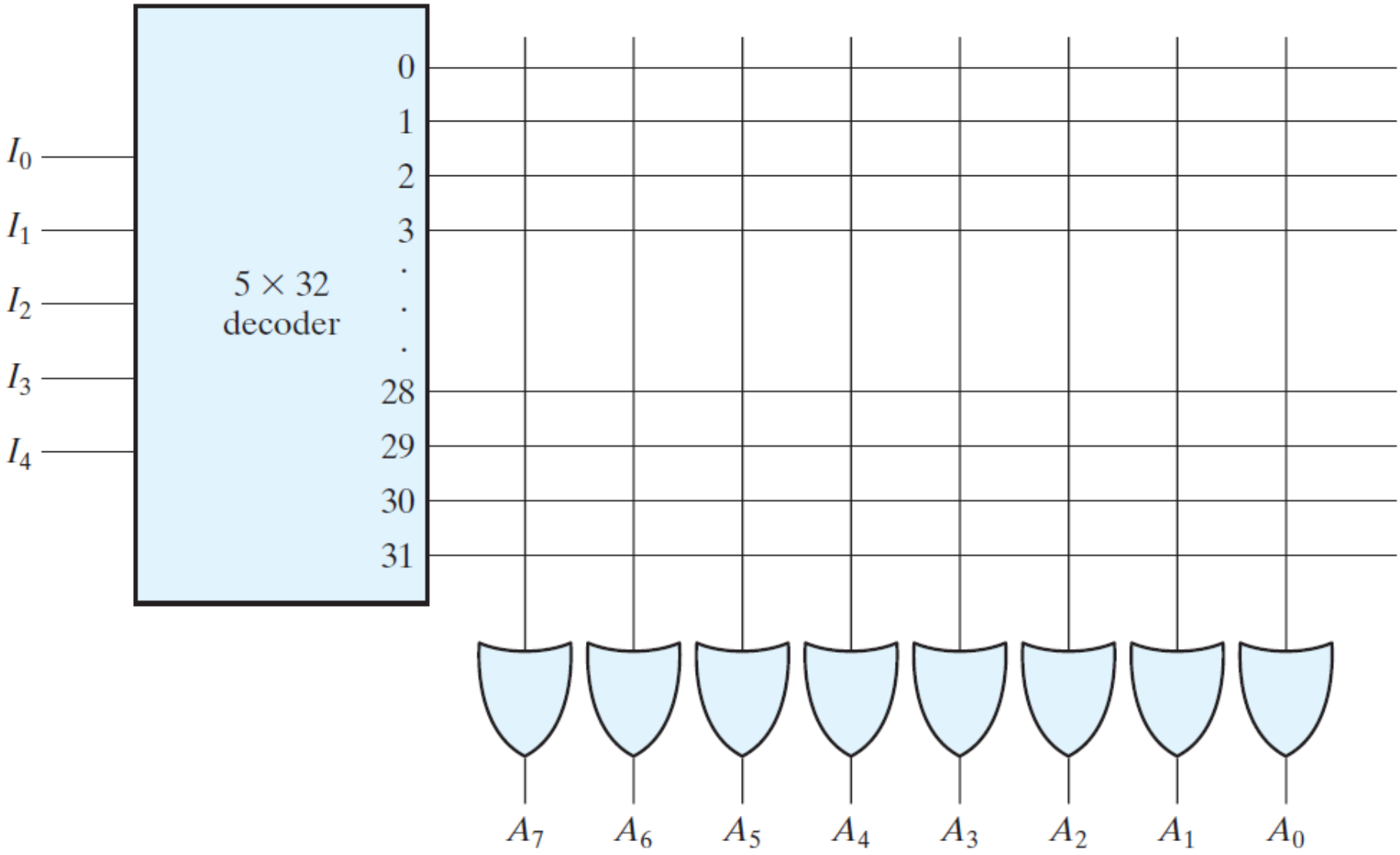
- A read-only memory (ROM) is essentially a memory device in which permanent binary information is stored. The binary information must be specified by the designer and is then embedded in the unit to form the required interconnection pattern. Once the pattern is established, it stays within the unit even when power is turned off and on again.
- A block diagram of a ROM consisting of k inputs and n outputs is shown in Fig. The inputs provide the address for memory, and the outputs give the data bits of the stored word that is selected by the address. The number of words in a ROM is determined from the fact that k address input lines are needed to specify 2k words.
- Note that ROM does not have data inputs, because it does not have a write operation. Integrated



- The hardware procedure that programs the ROM blows fuse links in accordance with a given truth table. For example, programming the ROM according to the truth table given by Table results in the configuration shown in Fig.
- Every 0 listed in the truth table specifies the absence of a connection, and every 1 listed specifies a path that is obtained by a connection. For example, the table specifies the eight-bit word 10110010 for permanent storage at address 3. The four 0's in the word are programmed by blowing the fuse links between output 3 of the decoder and the inputs of the OR gates associated with outputs *A*6, *A*3, *A*2, and *A*0. The four 1's in the word are marked with a X to denote a temporary connection, in place of a dot used for a permanent connection in logic diagrams. When the input of the ROM is 00011, all the outputs of the decoder are 0 except for output 3, which is at logic 1. The signal equivalent to logic 1 at decoder output 3 propagates through the connections to the OR gate outputs of *A*7, *A*5, *A*4, and *A*1. The other four outputs remain at 0. The result is that the stored word 10110010 is applied to the eight data outputs.

Inputs					Outputs							
I_4	I_3	I_2	I_1	I_0	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
0	0	0	0	0	1	0	1	1	0	1	1	0
0	0	0	0	1	0	0	0	1	1	1	0	1
0	0	0	1	0	1	1	0	0	0	1	0	1
0	0	0	1	1	1	0	1	1	0	0	1	0
		⋮						⋮				
1	1	1	0	0	0	0	0	0	1	0	0	1
1	1	1	0	1	1	1	1	0	0	0	1	0
1	1	1	1	0	0	1	0	0	1	0	1	0
1	1	1	1	1	0	0	1	1	0	0	1	1

Internal logic of a 32 X 8 ROM



Types of ROMs

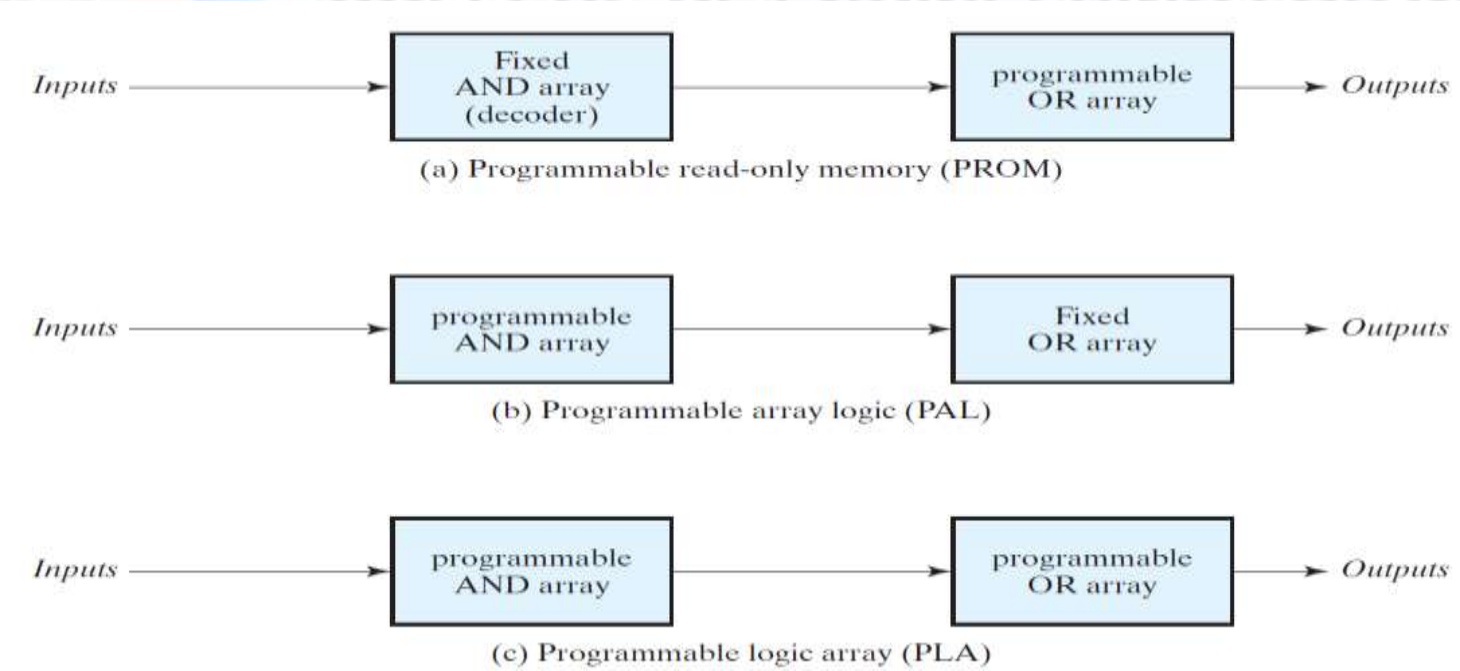
- The required paths in a ROM may be programmed in four different ways.
- The first is called *mask programming* and is done by the semiconductor company during the last fabrication process of the unit. The procedure for fabricating a ROM requires that the customer fill out the truth table he or she wishes the ROM to satisfy. The truth table may be submitted in a special form provided by the manufacturer or in a specified format on a computer output medium. The manufacturer makes the corresponding mask for the paths to produce the 1's and 0's according to the customer's truth table. This procedure is costly because the vendor charges the customer a special fee for custom masking the particular ROM. For this reason, mask programming is economical only if a large quantity of the same ROM configuration is to be ordered.
- For small quantities, it is more economical to use a second type of ROM called ***programmable read-only memory,*** **or PROM**. When ordered, PROM units contain all the fuses intact, giving all 1's in the bits of the stored words. The fuses in the PROM are blown by the application of a high-voltage pulse to the device through a special pin. A blown fuse defines a binary 0 state and an intact fuse gives a binary 1 state. This procedure allows the user to program the PROM in the laboratory to achieve the desired relationship between input addresses and stored words. Special instruments called PROM programmers are available commercially to facilitate the procedure. In any case, all procedures for programming ROMs are hardware procedures, even though the word *programming* is used.

Types of ROMs

- The hardware procedure for programming ROMs or PROMs is irreversible, and once programmed, the fixed pattern is permanent and cannot be altered. Once a bit pattern has been established, the unit must be discarded if the bit pattern is to be changed.
- A third type of ROM is the ***erasable PROM*, or EPROM**, which can be restructured to the initial state even though it has been programmed previously. When the EPROM is placed under a special ultraviolet light for a given length of time, the shortwave radiation discharges the internal floating gates that serve as the programmed connections. After erasure, the EPROM returns to its initial state and can be reprogrammed to a new set of values.
- The fourth type of ROM is the **electrically erasable PROM** (EEPROM or E2PROM). This device is like the EPROM, except that the previously programmed connections can be erased with an electrical signal instead of ultraviolet light. The advantage is that the device can be erased without removing it from its socket.
- Flash memory devices are similar to EEPROMs, but have additional built-in circuitry to selectively program and erase the device in-circuit, without the need for a special programmer. They have widespread application in modern technology in cell phones, digital cameras, set-top boxes, digital TV, telecommunications, nonvolatile data storage, and microcontrollers. Their low consumption of power makes them an attractive storage medium for laptop and notebook computers.
- Flash memories incorporate additional circuitry, too, allowing simultaneous erasing of blocks of memory, for example, of size 16 to 64 K bytes. Like EEPROMs, flash memories are subject to fatigue, typically having about 105 block erase cycles.

Basic configuration of three PLDs

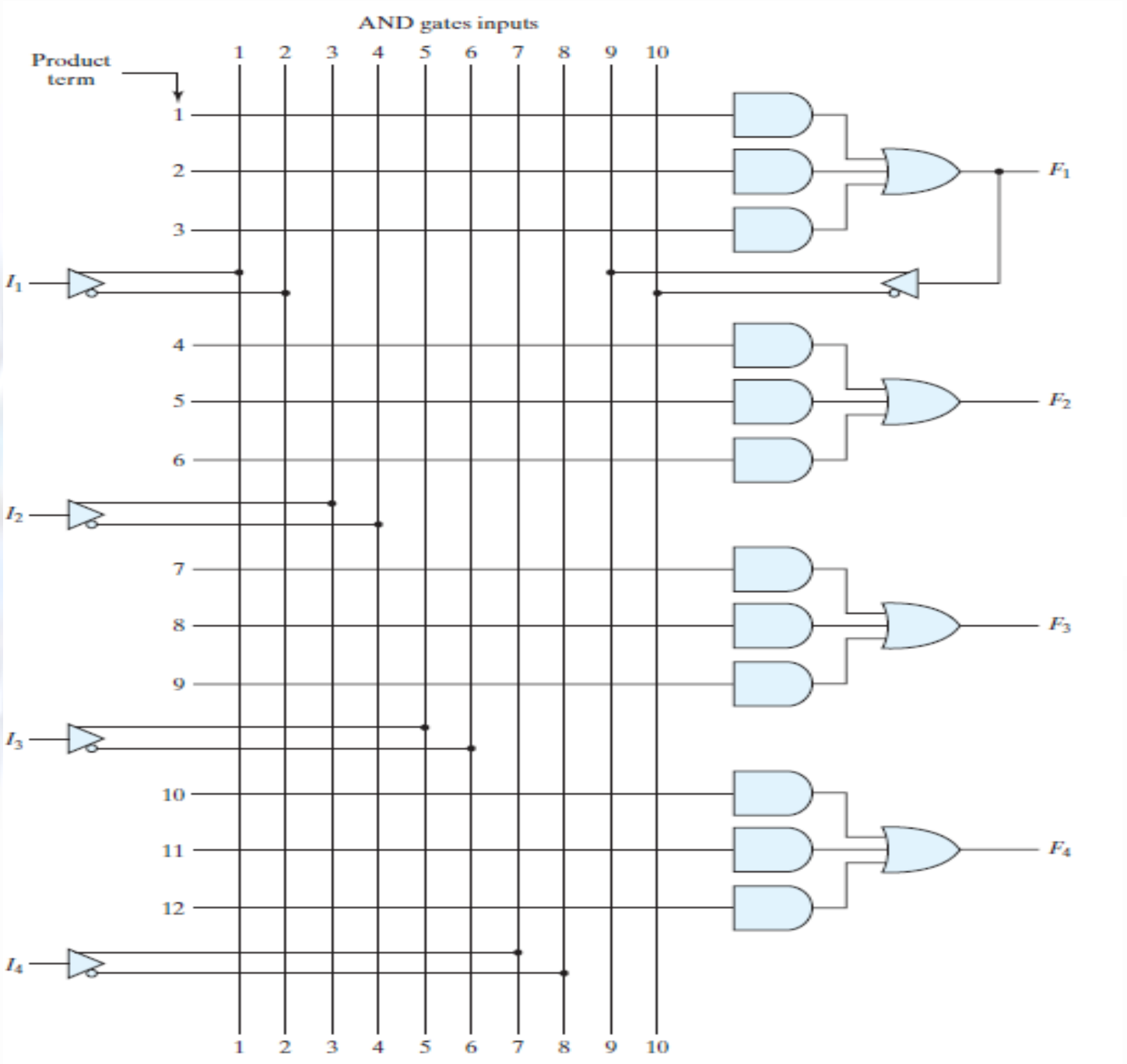
- The PROM has a fixed AND array constructed as a decoder and a programmable OR array. The programmable OR gates implement the Boolean functions in sum-of-minterms form.
- The PAL has a programmable AND array and a fixed OR array. The AND gates are programmed to provide the product terms for the Boolean functions, which are logically summed in each OR gate.
- The most flexible PLD is the PLA, in which both the AND and OR arrays can be programmed. The product terms in the AND array may be shared by any OR gate to provide the required sum-of-products implementation.



PROGRAMMABLE ARRAY LOGIC

- The PAL is a programmable logic device with a fixed OR array and a programmable AND array. Because only the AND gates are programmable, the PAL is easier to program than, but is not as flexible as, the PLA.
- Figure shows the logic configuration of a typical PAL with four inputs and four outputs. Each input has a buffer–inverter gate, and each output is generated by a fixed OR gate. There are four sections in the unit, each composed of an AND–OR array that is *three wide*, the term used to indicate that there are three programmable AND gates in each section and one fixed OR gate. Each AND gate has 10 programmable input connections, shown in the diagram by 10 vertical lines intersecting each horizontal line. The horizontal line symbolizes the multiple-input configuration of the AND gate.
- One of the outputs is connected to a buffer–inverter gate and then fed back into two inputs of the AND gates. A typical PAL integrated circuit may have eight inputs, eight outputs, and eight sections, each consisting of an eight-wide AND–OR array. The output terminals are sometimes driven by three-state buffers or inverters.
- In designing with a PAL, the Boolean functions must be simplified to fit into each section. Unlike the situation with a PLA, a product term cannot be shared among two or more OR gates. Therefore, each function can be simplified by itself, without regard to common product terms.
- The number of product terms in each section is fixed, and if the number of

PROGRAMMABLE ARRAY LOGIC



PROGRAMMABLE ARRAY LOGIC

- As an example of using a PAL in the design of a combinational circuit, consider the following Boolean functions, given in sum-of-minterms form:

  $w(A, B, C, D) = \sum(2, 12, 13)$

  $x(A, B, C, D) = \sum(7, 8, 9, 10, 11, 12, 13, 14, 15)$

  $y(A, B, C, D) = \sum(0, 2, 3, 4, 5, 6, 7, 8, 10, 11, 15)$

  $z(A, B, C, D) = \sum(1, 2, 8, 12, 13)$

- Simplifying the four functions to a minimum number of terms results in the following Boolean functions:

$w = ABC + ABCD$; $x = A + BCD$; $y = AB + CD + BD$; $z = ABC + ABCD + ACD + ABCD = w + ACD + ABCD$

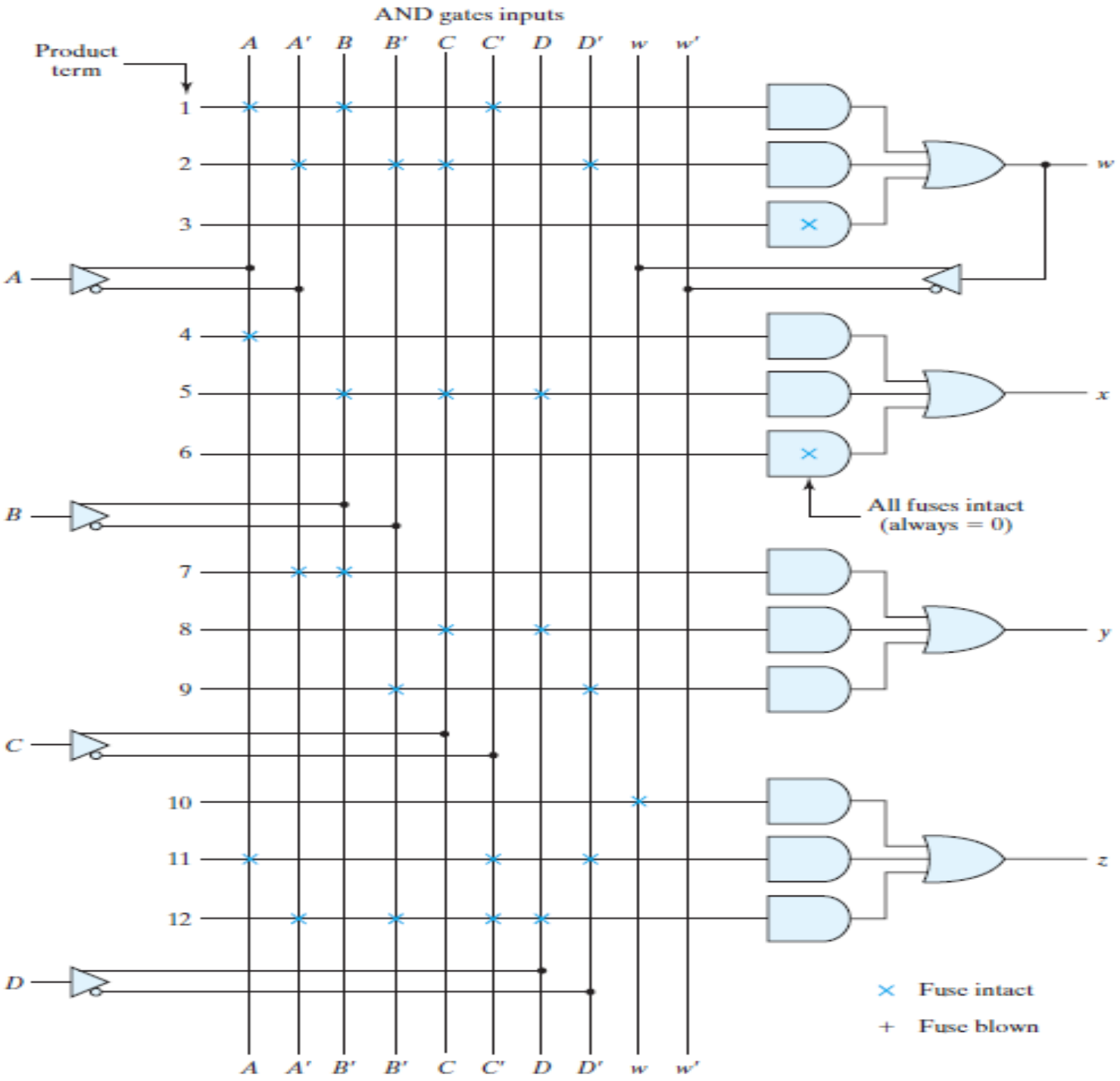
- Note that the function for *z* has four product terms. The logical sum of two of these terms is equal to *w* . By using *w,* it is possible to reduce the number of terms for *z* from four to three.

- The PAL programming table is similar to the one used for the PLA, except that only the inputs of the AND gates need to be programmed. Table lists the PAL programming table for the four Boolean functions. The table is divided into four sections with three product terms in each, to conform with the PAL of Fig. The first two sections need only two product terms to implement the Boolean function. The last section, for output *z,* needs four product terms. Using the output from *w,* we can reduce the function to three terms.

PAL Programming Table

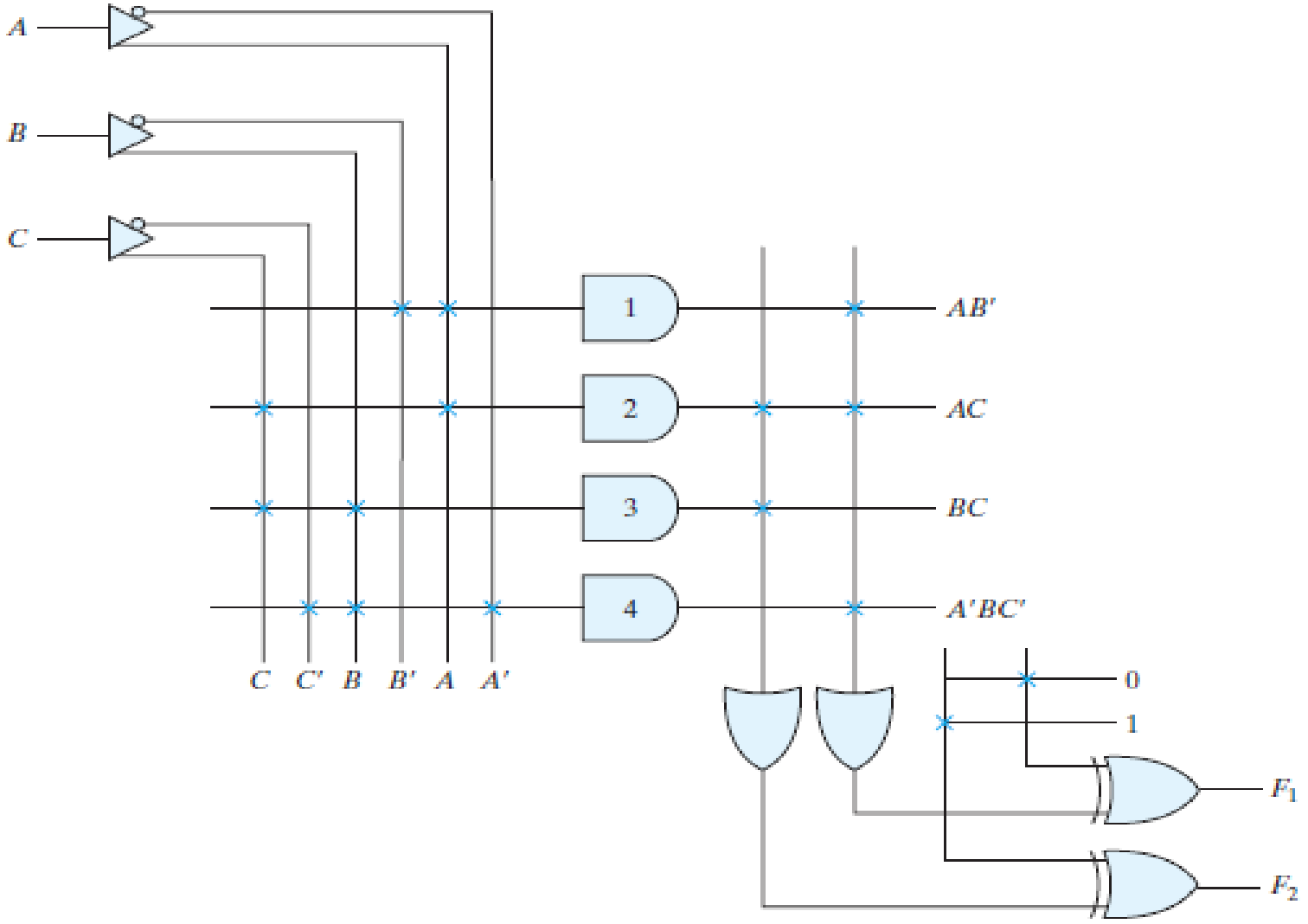
PAL Programming Table

Product Term	AND Inputs A	B	C	D	w	Outputs
1	1	1	0	—	—	$w = ABC' + A'B'CD'$
2	0	0	1	0	—	
3	—	—	—	—	—	
4	1	—	—	—	—	$x = A + BCD$
5	—	1	1	1	—	
6	—	—	—	—	—	
7	0	1	—	—	—	$y = A'B + CD + B'D'$
8	—	—	1	1	—	
9	—	0	—	0	—	
10	—	—	—	—	1	$z = w + AC'D' + A'B'C'D$
11	1	—	0	0	—	
12	0	0	0	1	—	


AND gates inputs
Product term
A A' B B' C C' D D' w w'
1
2
3
4
5
6
7
8
9
10
11
12
w
x
y
z
All fuses intact (always = 0)
Fuse intact
Fuse blown

PROGRAMMABLE LOGIC ARRAY

$$F_1 = AB' + AC + A'BC'$$
$$F_2 = (AC + BC)'$$



PROGRAMMABLE LOGIC ARRAY

- Each input goes through a buffer–inverter combination, shown in the diagram with a composite graphic symbol, that has both the true and complement outputs.
- Each input and its complement are connected to the inputs of each AND gate, as indicated by the intersections between the vertical and horizontal lines. The outputs of the AND gates are connected to the inputs of each OR gate. The output of the OR gate goes to an XOR gate, where the other input can be programmed to receive a signal equal to either logic 1 or logic 0. The output is inverted when the XOR input is connected to 1 (since $x \oplus 1 = x'$). The output does not change when the XOR input is connected to 0 (since $x \oplus 0 = x$).

PLA Programming Table

		Inputs			Outputs (T)	(C)
	Product Term	A	B	C	F_1	F_2
AB'	1	1	0	—	1	—
AC	2	1	—	1	1	1
BC	3	—	1	1	—	1
$A'BC'$	4	0	1	0	1	—