Once programmed, the data pattern can never be changed. This type of read-only memory is referred to as ROM. ROMs are highly suited for very high volume usage due to their low cost.

***Programmable read-only memory (PROM)***

This type of memory comes from the manufacturer without any data stored in it, i.e. empty. The data pattern is programmed electrically by the user using a special circuit known as *PROM programmer*. It can be programmed only once during its life time. Once programmed, the data cannot be altered. This type of memory is known as PROM. These are highly suited for high volume usage due to their low cost of production.

***Erasable programmable read-only memory (EPROM)***

In this type of memory, data can be written any number of times, i.e. they are reprogrammable. Before it is reprogrammed, the contents already stored are erased by exposing the chip to ultraviolet radiation for about 30 minutes. This type of memory is referred to as EPROM. EPROMs are possible only in MOS technology. Programming is done using a PROM programmer.

***Electrically erasable and programmable read-only memory (EEPROM or $E^2$PROM)***

This is another type of reprogrammable memory in which erasing is done electrically rather than exposing the chip to the ultraviolet radiation. It is referred to as EEPROM or electrically alterable ROM (EAROM).

## 8.6 COMBINATIONAL PROGRAMMABLE LOGIC DEVICES

A combinational PLD is an integrated circuit with programmable gates divided into an AND array and an OR array to provide an AND-OR sum of products implementation. There are three major types of combinational PLDs and they differ in the placement of the programmable connection in the AND-OR array. The various PLDs used are PALs (programmable array logics), PLAs (programmable logic arrays) and PROMs (programmable read only memories).

Figure 8.7 shows the configuration of the 3 PLDs. The programmable read-only memory (PROM) has a fixed AND array constructed as a decoder and a programmable 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 programmable array logic (PAL) has programmable AND array and

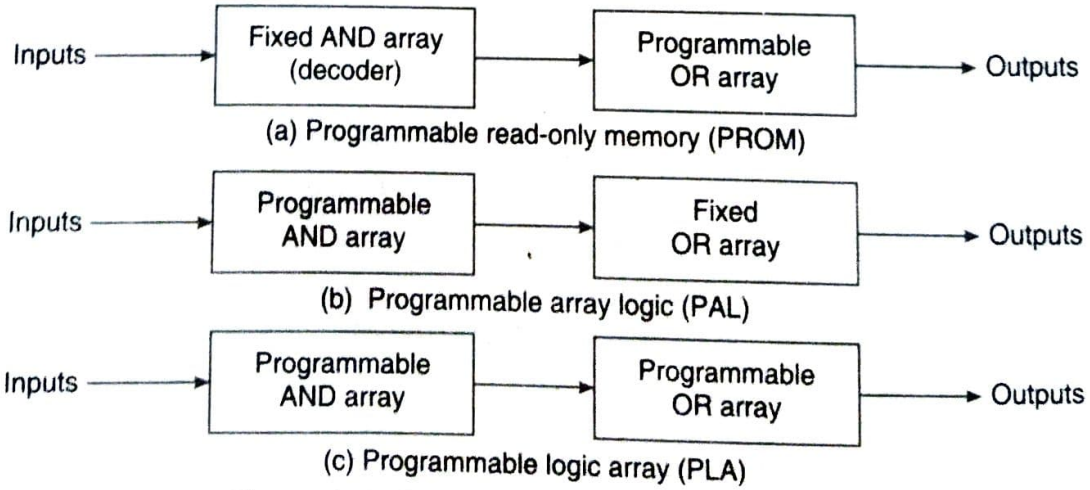



**Figure 8.7** Basic configuration of three PLDs.

a fixed OR array. The most flexible PLD is the programmable logic array (PLA) where 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.

## 8.7 PROGRAMMABLE ARRAY LOGIC (PAL)

Programmable array logic (a registered trade mark of Monolithic Memories) is a particular family of programmable logic devices (PLDs) that is widely used and available from a number of manufacturers. The PAL circuits consist of a set of AND gates whose inputs can be programmed and whose outputs are connected to an OR gate, i.e. the inputs to the OR gate are hard-wired, i.e. PAL is a PLD with a fixed OR array and a programmable AND array. Because only the AND gates are programmable, the PAL is easier to program but is not as flexible as the PLA. Some manufacturers also allow output inversion to be programmed. Thus, like AND-OR and AND-OR-INVERT logic, they implement a sum of products logic function. Figure 8.8a shows a small example of the basic structure. The fuse symbols represent fusible links that can be burned open using equipment similar to a PROM programmer. Note that every input variable and its complement can be left either connected or disconnected from every AND gate. We then say that the AND gates are programmed. Figure 8.8b shows how the circuit is programmed to implement $F = \bar{A}BC + A\bar{B}\bar{C}$. Note this important point. All input variables and their complements are left connected to the unused AND gate, whose output is, therefore, $A\bar{A}B\bar{B}C\bar{C} = 0$. The 0 has no affect on the output of the OR gate. On the other hand, if all inputs to the unused AND gate were burned open, the output of the AND gate would 'float' HIGH (logic 1), and the output of the OR gate in that case would remain permanently 1. The actual PAL circuits have several groups of AND gates, each group providing inputs to separate OR gates.

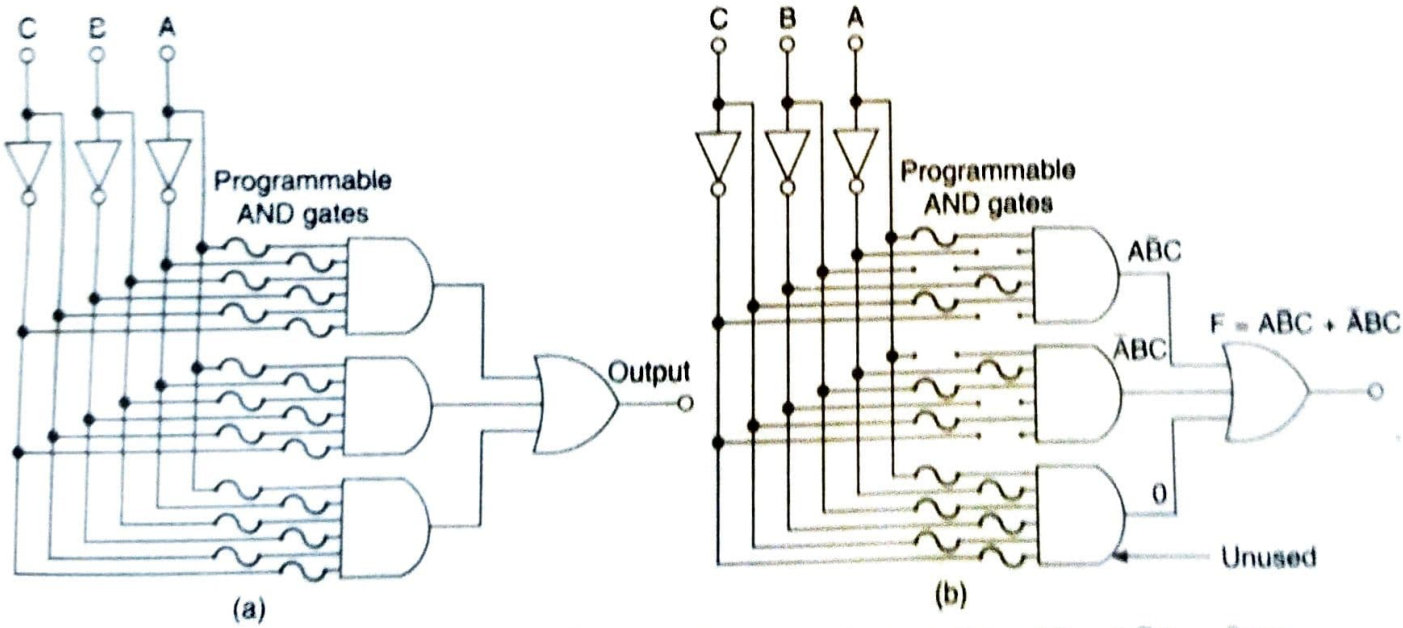



**Figure 8.8** Basic structure of a PAL circuit, and implementation of $F = A\bar{B}C + \bar{A}BC$.

Figure 8.9a shows a conventional means for abbreviating PAL connection diagrams. Note that the AND gate is drawn with a single input line, whereas in reality, it has three inputs. The vertical lines denote the inputs and the horizontal lines feed the AND gates. An × sign denotes a connection through an intact fusible link and a dot sign represents a permanent connection. The

absence of any symbol represents an open or no connection by virtue of a burned-open link. In the example shown, input A is connected to the gate through a fusible link, input C is permanently connected, and input B is disconnected. Therefore, the output of the gate is AC.

Figure 8.9b shows an example of how the PAL structure is represented using the abbreviated connections. It is a 3-input 3-wide AND-OR structure. In this example, each function can have three minterms or product terms. Notice that there are nine AND gates, which implies only nine chosen products of not more than three variables ABC. Inputs to the OR gates at the outputs are fixed as shown by ×s marked on the vertical lines. The inputs to the AND gates are marked on the corresponding line by the ×s. Removing the × implies blowing off the corresponding fuse which in turn implies that the corresponding input variable is not applied to the particular AND gate. In this example, the circuit is unprogrammed because all the fusible links are intact. Note that, the 3-input OR gates in Figure 8.9c are also drawn with a single input line.

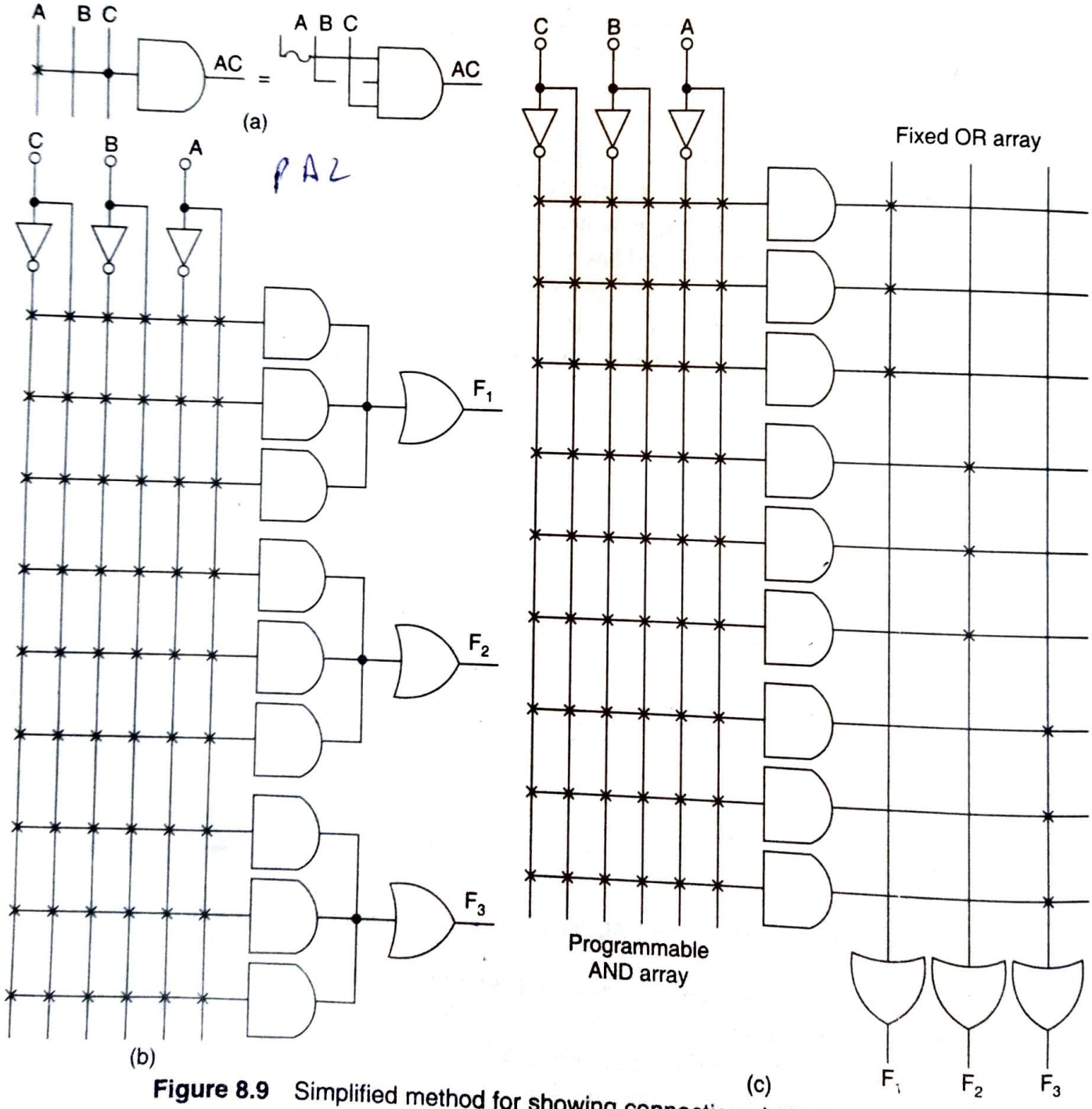



**Figure 8.9** Simplified method for showing connections in PAL circuits.

**EXAMPLE 8.4** Using the connection abbreviations, redraw the circuit in Figure 8.9c to show how it can be programmed to implement $F_1 = \bar{A}BC + A\bar{C} + A\bar{B}C$ and $F_2 = \bar{A}\bar{B}\bar{C} + BC$.

***Solution***

The redrawn circuit to implement the given functions is shown in Figure 8.10a. Note that one unused AND gate has all its links intact. All links intact can be represented by a × in the AND gate as shown in Figure 8.10b. Such a diagram is sometimes called the *fuse map*.

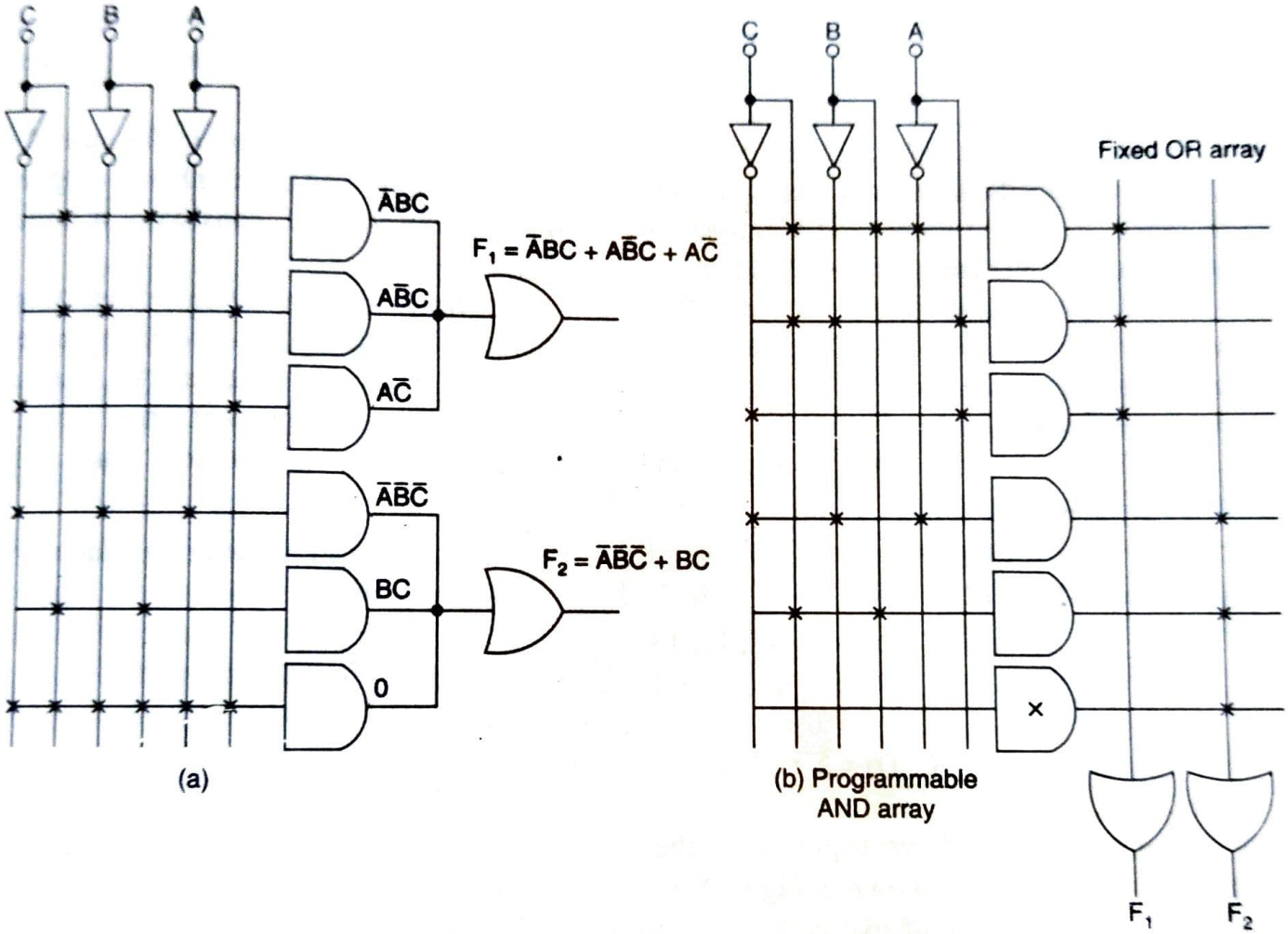
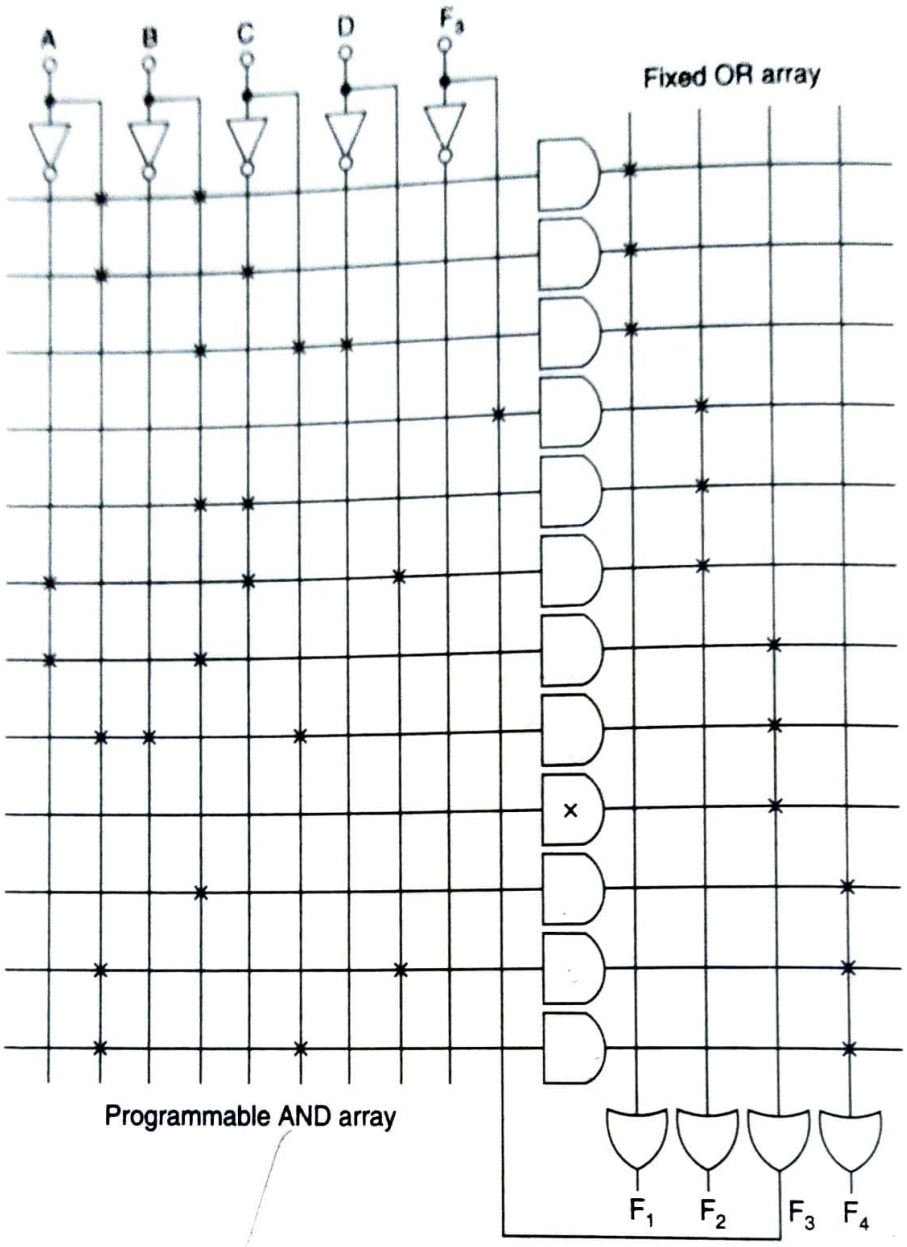



**Figure 8.10** Example 8.4: PAL programmed to implement $F_1 = \bar{A}BC + A\bar{B}C + A\bar{C}$ and $F_2 = \bar{A}\bar{B}\bar{C} + BC$.

An example of an actual PAL IC is the PAL 18L8A from Texas Instruments. It is manufactured using low power Schottky technology and has ten logic inputs and eight output functions. Each output OR gate is hard-wired to seven AND gate outputs and therefore it can generate functions that include up to seven terms. An added feature of this particular PAL is that six of the eight outputs are fed back into AND array, where they can be connected as inputs to any AND gate. This makes the device very useful in generating all sorts of combinational logic.

### 8.7.1 PAL Programming Table

The fuse map of a PAL can be specified in a tabular form. The PAL programming table consists of three columns. The first column lists the product terms numerically. The second column specifies




(b) Realization of the example functions using PAL

**Figure 8.14** Example 8.6.

## 8.8 PROGRAMMABLE LOGIC ARRAY (PLA)

The PLA represents another type of programmable logic but with a slightly different architecture. The PLA combines the characteristics of the PROM and the PAL by providing both a programmable OR array and a programmable AND array, i.e. in a PLA both AND gates and OR gates have fuses at the inputs. A third set of fuses in the output inverters allows the output function to be inverted if required. Usually X-OR gates are used for controlled inversion. This feature makes it the most versatile of the three PLDs. However, it has some disadvantages. Because it has two sets of fuses, it is more difficult to manufacture, program and test it than a PROM or a PAL. Figure 8.15 demonstrates the structure of a three-input, four-output PLA with every fusible link intact.

Like ROM, PLA can be mask programmable or field programmable. With a mask programmable PLA, the user must submit a PLA programming table to the manufacturer. This

table is used by the vender to produce a user made PLA that has the required internal paths between inputs and outputs. A second type of PLA available is called a field programmable logic array or FPLA. The FPLA can be programmed by the user by means of certain recommended procedures. FPLAs can be programmed with commercially available programmer units.

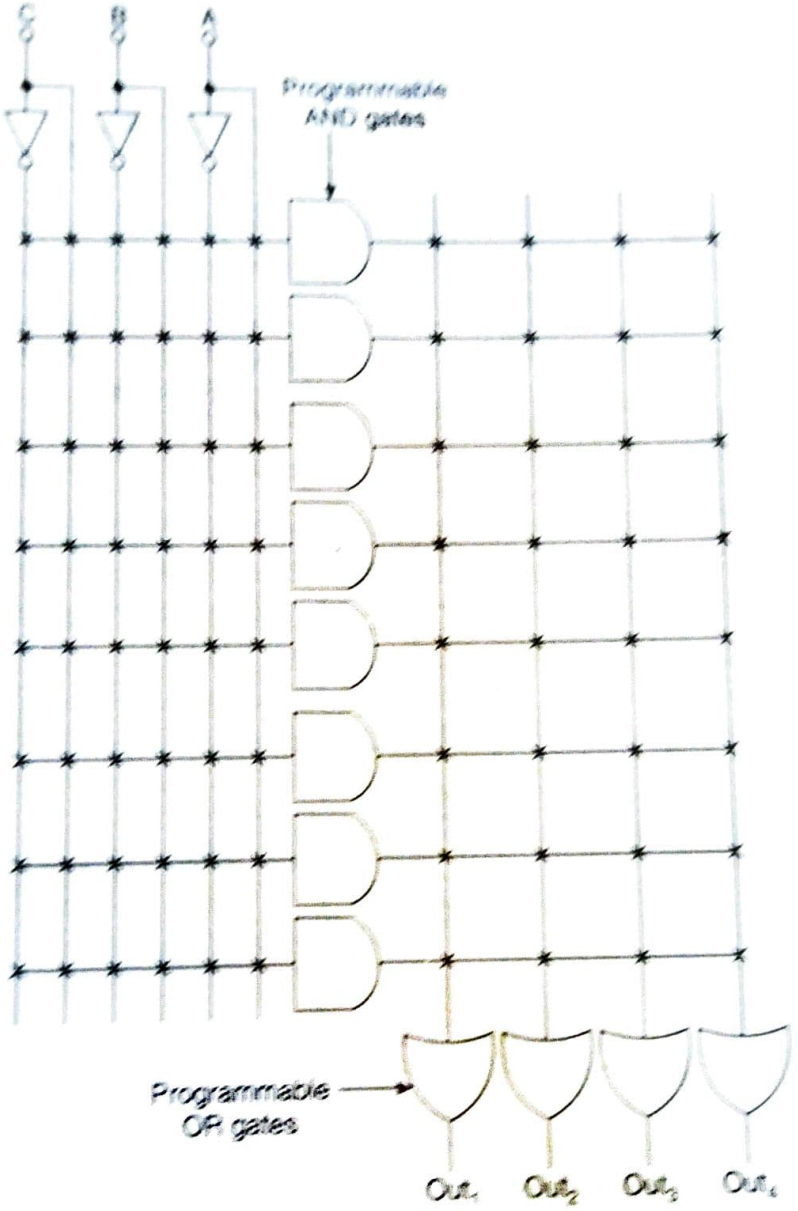



**Figure 8.15** Structure of (an unprogrammed) PLA circuit.

**EXAMPLE 8.7** *Show how the PLA circuit in Figure 8.15 would be programmed to implement the sum and carry outputs of a full adder.*

***Solution***

The truth table of a full-adder is shown in Figure 8.16a . Drawing the K-maps for the sum and carry-out terms and minimizing them, the minimal expressions for the sum and carry-out terms are:

The sum is

$$S = \bar{A}\bar{B}C_{in} + \bar{A}B\bar{C}_{in} + A\bar{B}\bar{C}_{in} + ABC_{in}$$

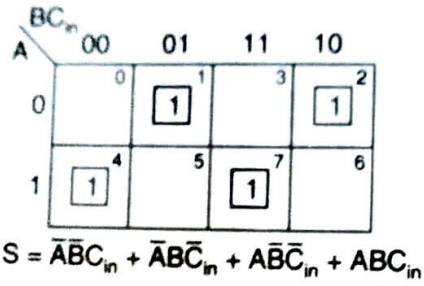
and the carry-out is

$$C_{out} = AB + AC_{in} + BC_{in}$$

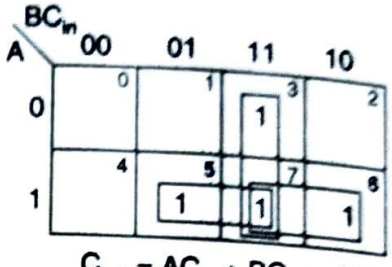
To implement these expressions, we need a 4-input OR gate and a 3-input OR gate. Since the inputs to the OR gates of the PLA can be programmed, we can implement the given expressions as shown in Figure 8.16c.

| Inputs | | | Outputs | |
|---|---|---|---|---|
| A | B | $C_{in}$ | S | $C_o$ |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

(a) Truth table

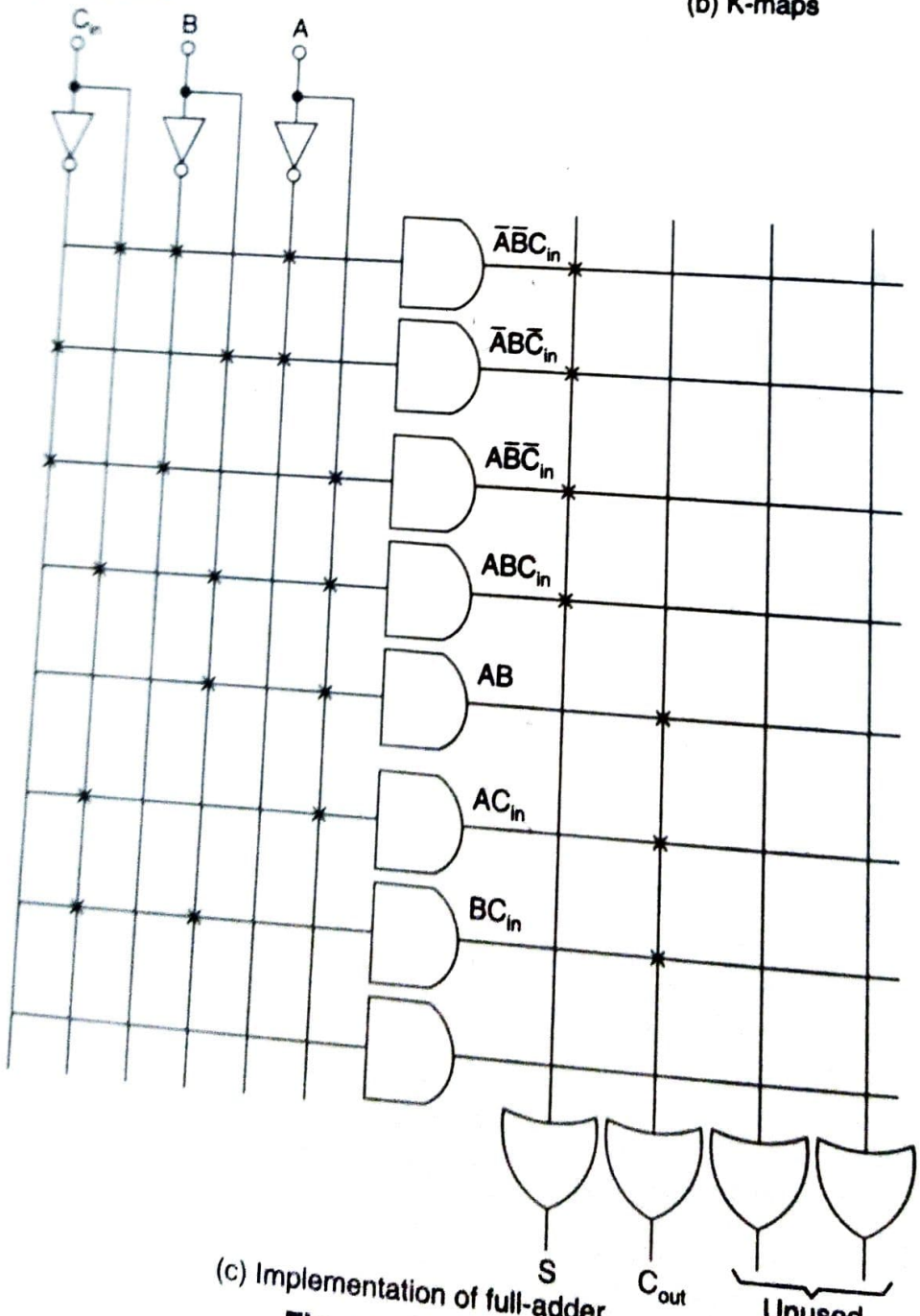



$S = \bar{A}\bar{B}C_{in} + \bar{A}B\bar{C}_{in} + A\bar{B}\bar{C}_{in} + ABC_{in}$




$C_{out} = AC_{in} + BC_{in} + AB$

(b) K-maps




(c) Implementation of full-adder

**Figure 8.16** Example 8.7.