

4.7 Programmable Logic Devices

There are a wide variety of ICs that can have their logic function “programmed” into them after they are manufactured. Most of these devices use technology that also allows the function to be reprogrammed, which means that if you find a bug in your design.

Programmable logic arrays (PLAs) were the first programmable logic devices. PLAs contained a two-level structure of AND and OR gates with user-programmable connections. Using this structure, a designer could accommodate any logic function up to a certain level of complexity using the well-known theory of logic synthesis and minimization

PLA structure was enhanced and PLA costs were reduced with the introduction of programmable array logic (PAL) devices. Today, such devices are generically called programmable logic devices (PLDs), and are the “MSI” of programmable logic industry.

The PLA is similar in concept to the PROM, except that the PLA does not provide full decoding of the variables and does not generate all the minterms. The decoder is replaced by an array of AND gates that can be programmed to generate any product term of the input variables. The product terms are then connected to OR gates to provide the sum of products for the required Boolean functions. The output is inverted when the XOR input is connected to

0 (since $x \oplus 0 = x$).

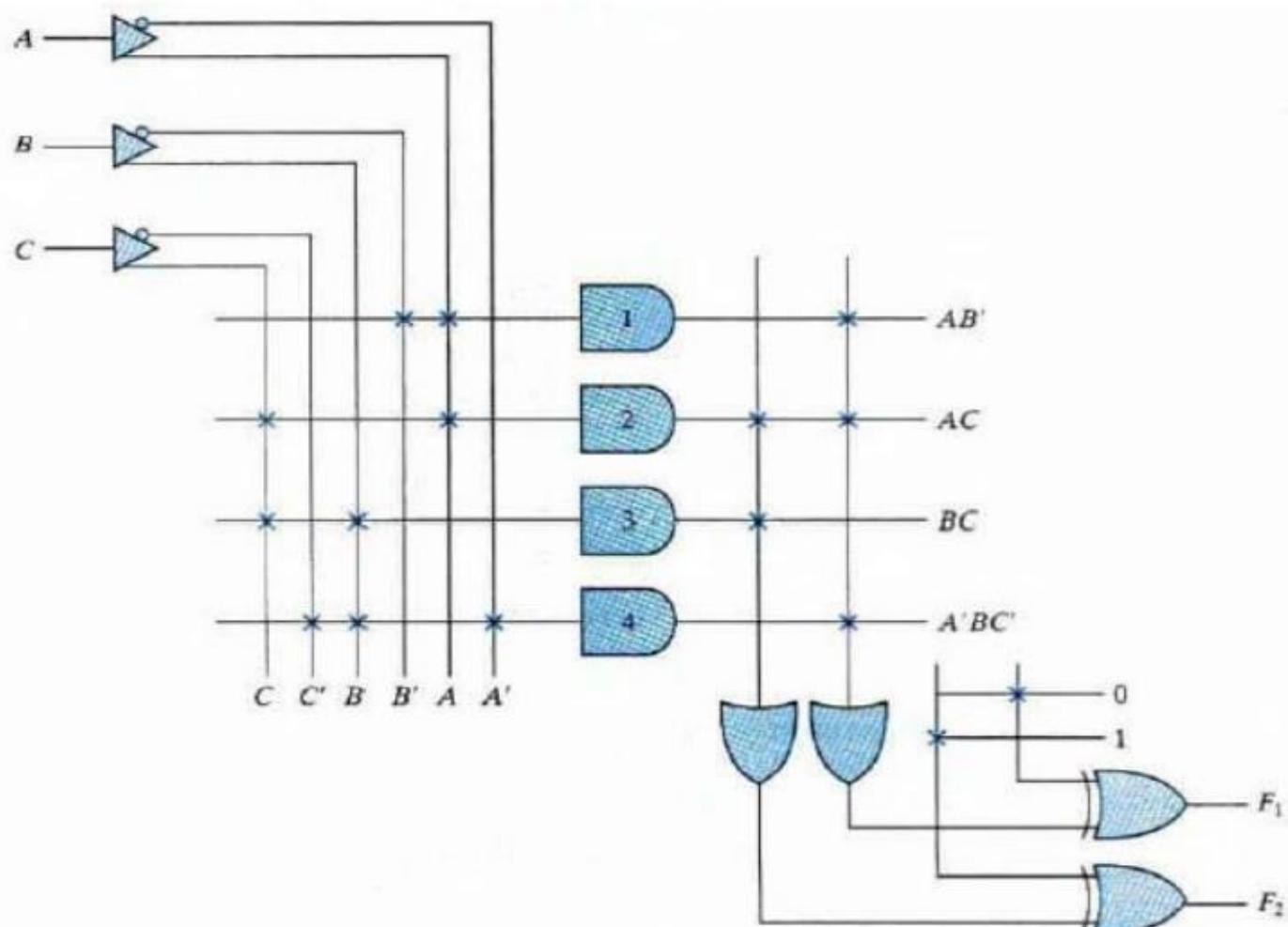
$$F_1 = A\bar{B} + AC + \bar{A}B\bar{C}$$

$$F_2 = \overline{(AC + BC)}$$



The fuse map of a PLA can be specified in a tabular form. The first section lists the product terms numerically. The second section specifies the required paths between inputs and AND gates. The third section specifies the paths between the AND and OR gates. For each output variable, we may have a T'(for true) or C (for complement) for programming the XOR gate.

For each product term, the inputs are marked with 1, 0, or - (dash). If a variable in the product term appears in the form in which it is true, the corresponding input variable is marked with a 1. If it appears complemented, the corresponding input variable is marked with a 0. If the variable is absent from the product term, it is marked with a dash.



OBSERVE OPTIMIZE OUTSPREAD

Fig 4.7.1 PLA

PLA Programming Table

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

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.

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.

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

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

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

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

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

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

$$w = ABC' + A'B'CD'$$

$$x = A + BCD$$

$$y = A'B + CD + B'D'$$

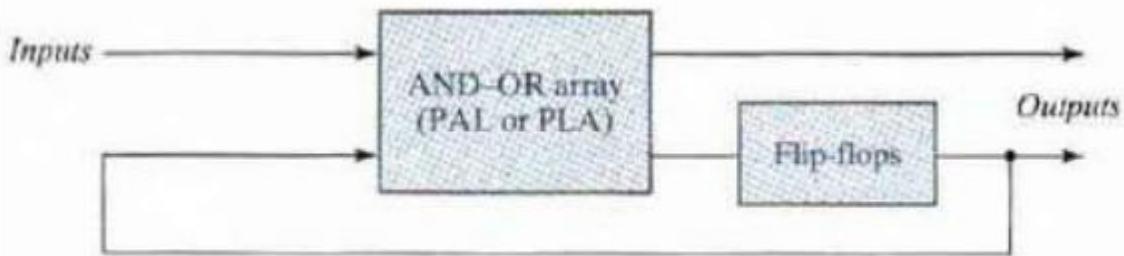
$$\begin{aligned} z &= ABC' + A'B'CD' + AC'D' + A'B'C'D \\ &= w + AC'D' + A'B'C'D \end{aligned}$$

PAL Programming Table

Product Term	AND Inputs					Outputs
	A	B	C	D	w	
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	—	

Sequential Programmable Devices

Digital systems are designed with flip-flops and gates. Since the combinational PLD consists of only gates, it is necessary to include external flip-flops when they are used in the design. Sequential programmable devices include both gates and flip-flops.



1. Sequential (or simple) programmable logic device (SPLD)
2. Complex programmable logic device (CPLD)
3. Field-programmable gate array (FPGA)

The ever-increasing capacity of integrated circuits created an opportunity for IC manufacturers to design larger PLDs for larger digital-design applications. Instead, IC manufacturers devised complex PLD (CPLD) architectures to achieve the required scale. A typical CPLD is merely a collection of multiple PLDs and an interconnection structure, all on the same chip. In addition to the individual PLDs, the on-chip interconnection structure is also programmable, providing a rich variety of design possibilities. CPLDs can be scaled to larger sizes by increasing the number of individual PLDs and the richness of the interconnection structure on the CPLD chip.

At about the same time that CPLDs were being invented, other IC manufacturers took a different approach to scaling the size of programmable logic chips. Compared to a CPLD, a field-programmable gate arrays (FPGA) contains a much

larger number of smaller individual logic blocks, and provides a large, distributed interconnection structure that dominates the entire chip.

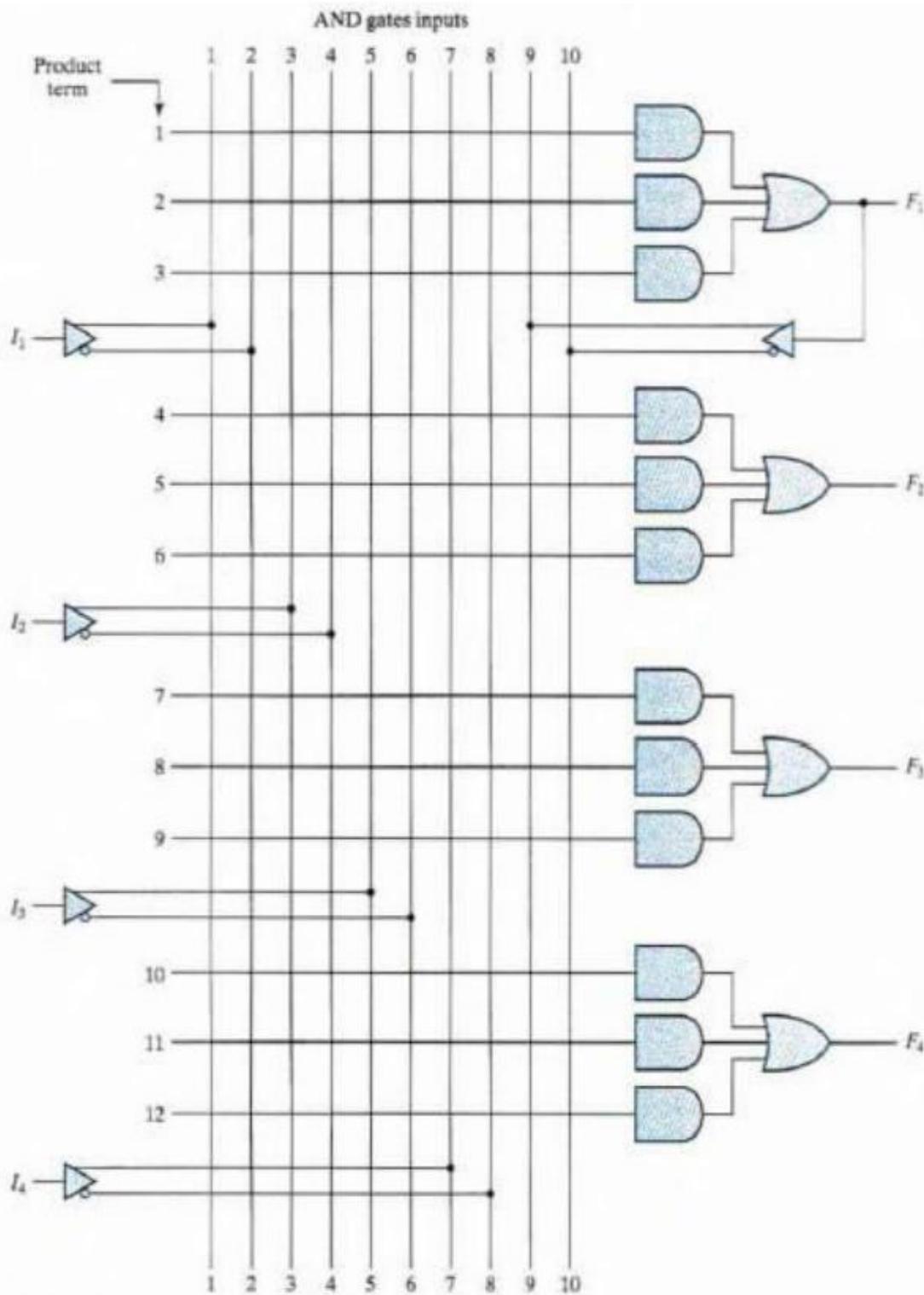


Fig: PAL with four inputs, four outputs and three AND-OR structure

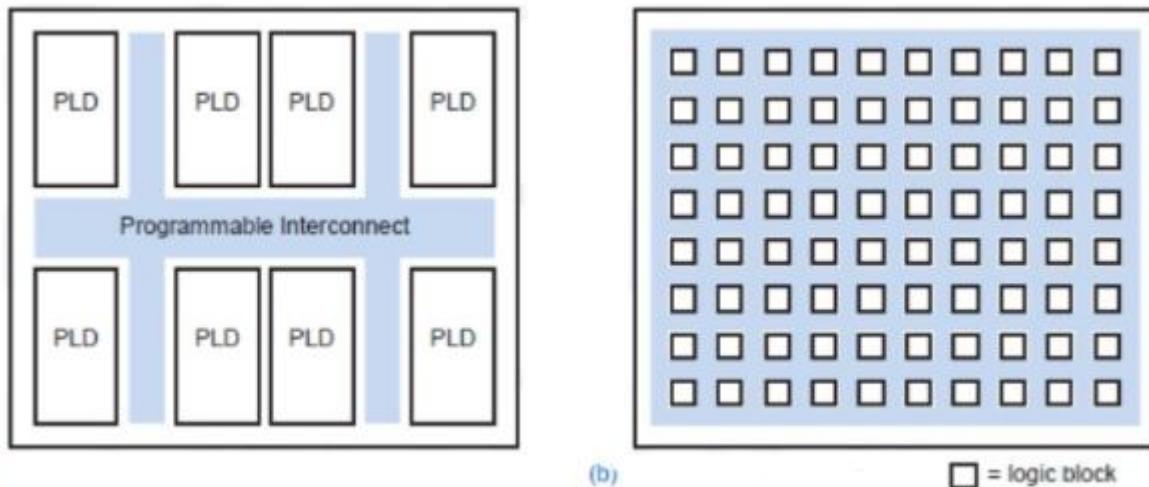


Fig 4.7.1 a) CPLD b) FPGA

Field-Programmable Gate Arrays

Even larger devices, often called *field-programmable gate arrays (FPGAs)*, use read/write memory cells to control the state of each connection. The read/write memory cells are volatile— they do not retain their state when power is removed. Therefore, when power is first applied to the FPGA, all of its read/write memory must be initialized to a state specified by a separate, external nonvolatile memory. This memory is typically either a programmable read-only memory (PROM) chip attached directly to the FPGA or it's part of a microprocessor subsystem that initializes the FPGA as part of overall system initialization.