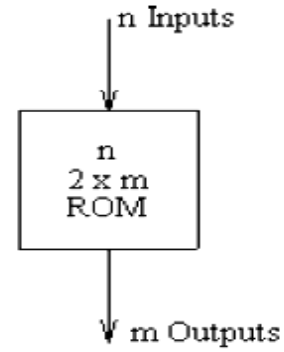
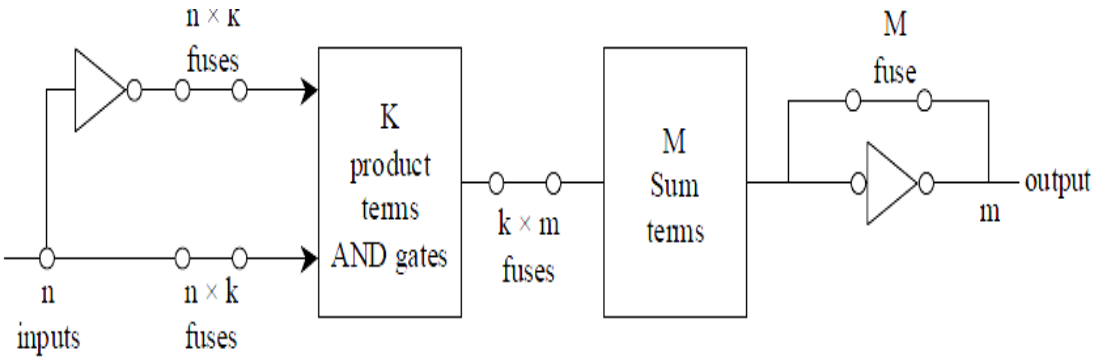
PROGRAMMABLE LOGIC ARRAY (PLA)

The use of programmable logic carry is more economical for cases where the number of don't care conditions is excessive. Conceptually the PLA is similar to ROM but it doesn't provide full decoding of the variable and does not generate all the minterms as like in the ROM. The block diagram of PLA is shown below. It consists of n inputs, m outputs, k product terms, and M sum terms. The product term consists of a group of k AND gates and the sum term constitute a group of m OR gates. Fuses are inserted between all n inputs and their complemented values to each of



Block diagram

the AND gates. Fuses are also provided between the outputs of the AND gates, and the inputs of the OR gates. Another set of fuses in the output inverters allows the output function to be generated either in the AND–OR form or in the AND–OR–INVERT form. With the Fuse blown gives AND – OR – INVERT implementation. With the fuse not blown, gives AND–OR implementation.



Each input and its complement are connected through fuse to the inputs of all AND gates, the outputs of the AND gates are connected through fuses to each input of the OR gates. Two more fuses are provided with the output inverters. By blowing selected fuses and leaving others intact, it is possible to implement Boolean functions in their sum of products form.

The PLA's available are of two types (1) Mask – programmable logic array and (2) Field-programmable logic array. In Mask - programmable logic array, the consumer must submit a PLA program table to the manufacturer. According to the PLA table given by the costumer vendor produce a custom - made PLA that has the required internal paths between the inputs and outputs. The field programmable array can be programmed by the user by means of certain recommended procedures. Commercial hardware programmer units are available for use in conjuction with certain FPLAs.

PLA Program Table :

The PLA consists of three columns. The first column lists the product terms numerically. The second column specifies the path between the AND gates and the OR gates. Under each output variable, we write T (for true) if the output inverter is to be by passed, and (for complement) if the function is to be complemented with the output inverter. The Boolean terms listed at the left are not part of the table; they are included for reference only. The products terms are the number of distinct products terms in the simplified Boolean expressions of the combinational logic circuit.

Example:

Implement the Boolean functions F1 and F2 of a Combinational logic circuit using PLA.

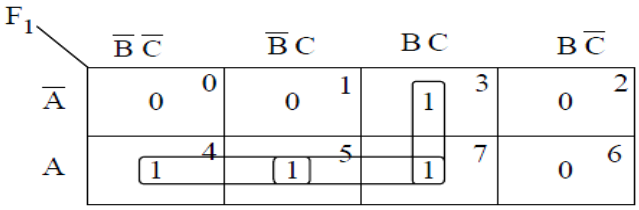
Where F1 (A, B, C) = Σ(3, 4, 5, 7)

F2 (A, B, C) = Σ (1, 4, 6)

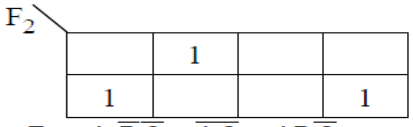
The truth table of this combinational logic circuit is shown below.

A	B	C	F_1	F_2
0	0	0	0	0
0	0	1	0	1
0	1	0	0	0
0	1	1	1	0
1	0	0	1	1
1	0	1	1	0
1	1	0	0	1
1	1	1	1	0

Truth Table



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



$F_2 = A\bar{B}\bar{C} + \bar{A}C + AB\bar{C}$

Map Simplification

	Product terms	Inputs			Outputs	
		A	B	C	F_1	F_2
$A\bar{B}$	1	1	0	–	1	–
AC	2	1	–	1	1	–
BC	3	–	1	1	1	–
$A\bar{B}\bar{C}$	4	1	0	0	–	1
$\bar{A}\bar{B}C$	5	0	0	1	–	1
$AB\bar{C}$	6	1	1	0	–	1
					T	T	T/C