DE- Unit V

Prof. Gauri Rao

UNIT CONTENTS

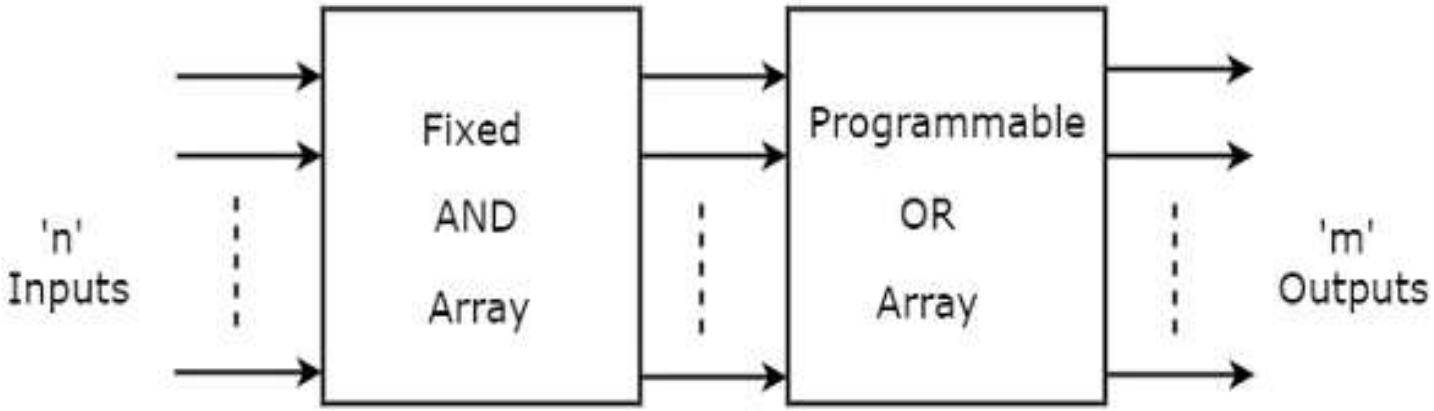
- Programmable logic devices: Architecture of PLA, Designing combinational circuits using PLDs.
- Semiconductor memories: memory organization, memory expansion, Classification and characteristics of memories, RAM, ROM, EPROM, EEPROM, NVRAM, SRAM, DRAM.

PLD is a re-configurable IC built with large numbers of gates connected through electronic fuses to implement arbitrary circuits.

- Programmable ROM (Read-Only Memory)
- Programmable Logic Array (PLA)
- Programmable Array Logic (PAL)

- Programmable Read Only Memory PROM
- Read Only Memory ROM is a memory device, which stores the binary information permanently. That means, we can't change that stored information by any means later. If the ROM has programmable feature, then it is called as **Programmable ROM** PROMPROM. The user has the flexibility to program the binary information electrically once by using PROM programmer.
- PROM is a programmable logic device that has fixed AND array & Programmable OR array.

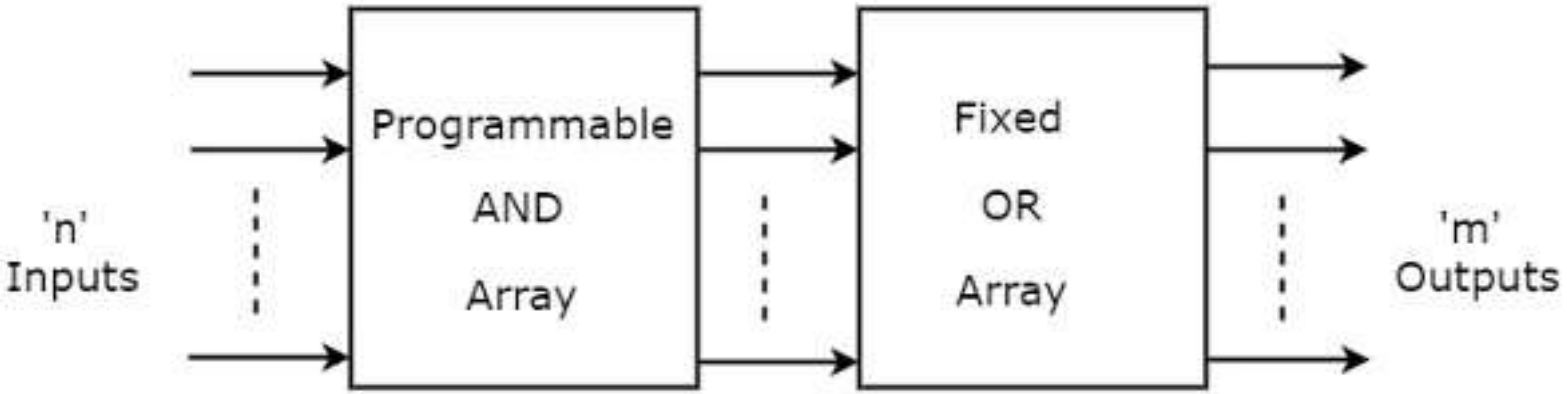
PROM



PAL

- PAL is a programmable logic device that has Programmable AND array & fixed OR array. The advantage of PAL is that we can generate only the required product terms of Boolean function instead of generating all the min terms by using programmable AND gates. Here, the inputs of AND gates are programmable. That means each AND gate has both normal and complemented inputs of variables. So, based on the requirement, we can program any of those inputs. So, we can generate only the required **product terms** by using these AND gates.
- Here, the inputs of OR gates are not of programmable type. So, the number of inputs to each OR gate will be of fixed type. Hence, apply those required product terms to each OR gate as inputs. Therefore, the outputs of PAL will be in the form of **sum of products form**.

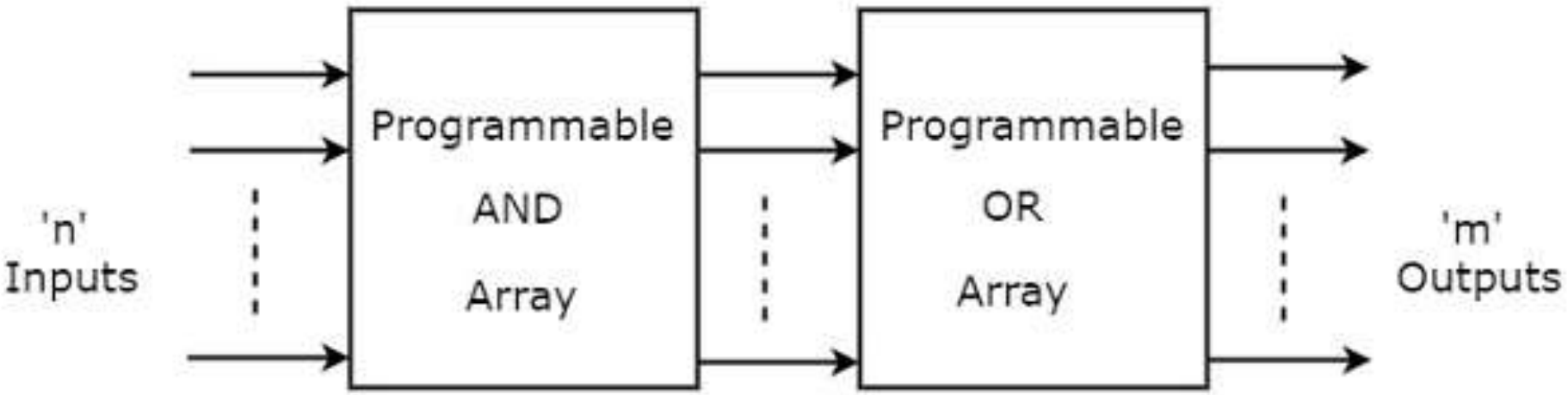
PAL



PLA

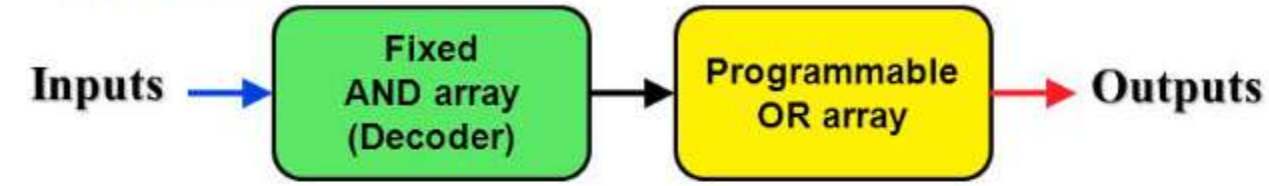
- PLA is a programmable logic device that has both Programmable AND array & Programmable OR array. Here, the inputs of AND gates are programmable. That means each AND gate has both normal and complemented inputs of variables. So, based on the requirement, we can program any of those inputs. So, we can generate only the required **product terms** by using these AND gates.
- Here, the inputs of OR gates are also programmable. So, we can program any number of required product terms, since all the outputs of AND gates are applied as inputs to each OR gate. Therefore, the outputs of PAL will be in the form of **sum of products form**.

PLA

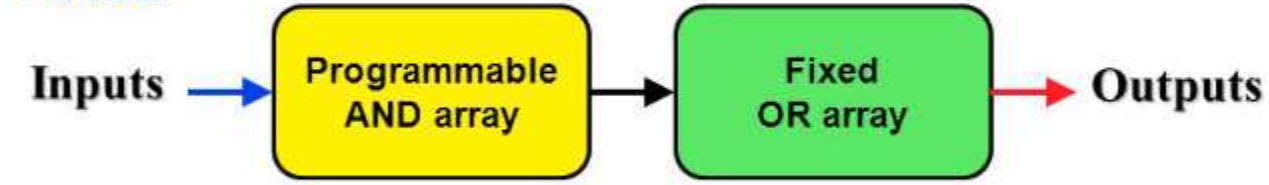

'n'
Inputs
Programmable
AND
Array
Programmable
OR
Array
'm'
Outputs

Summary

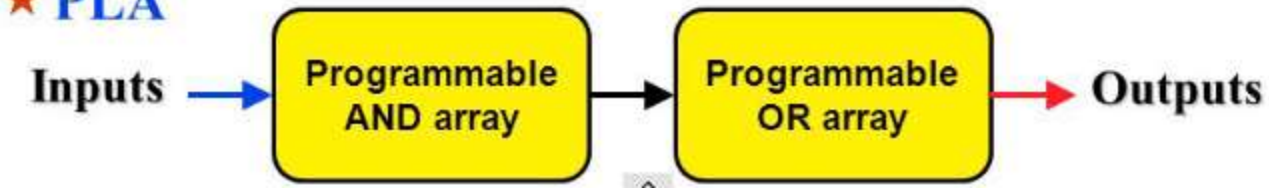
★ PROM

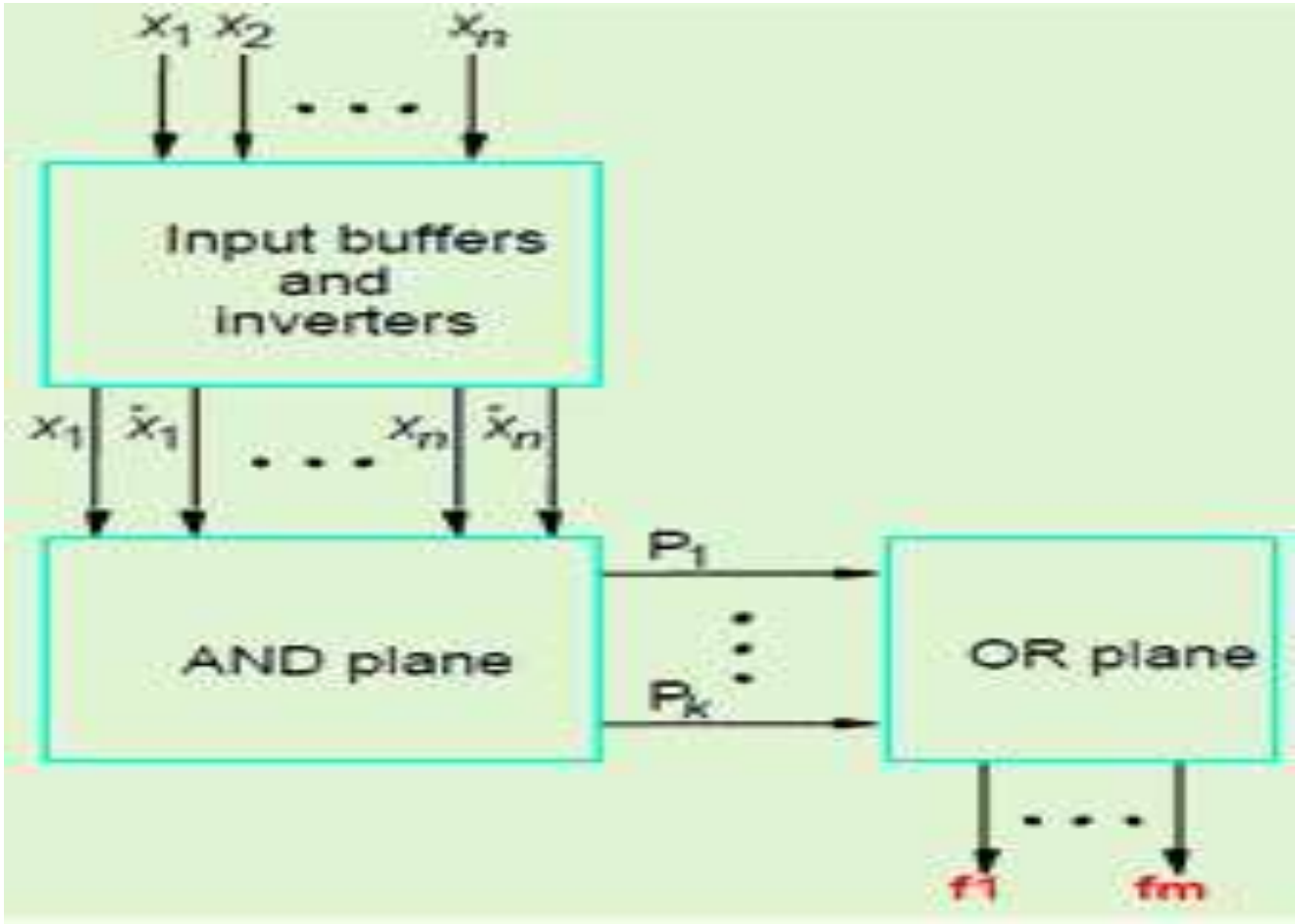


★ PAL



★ PLA

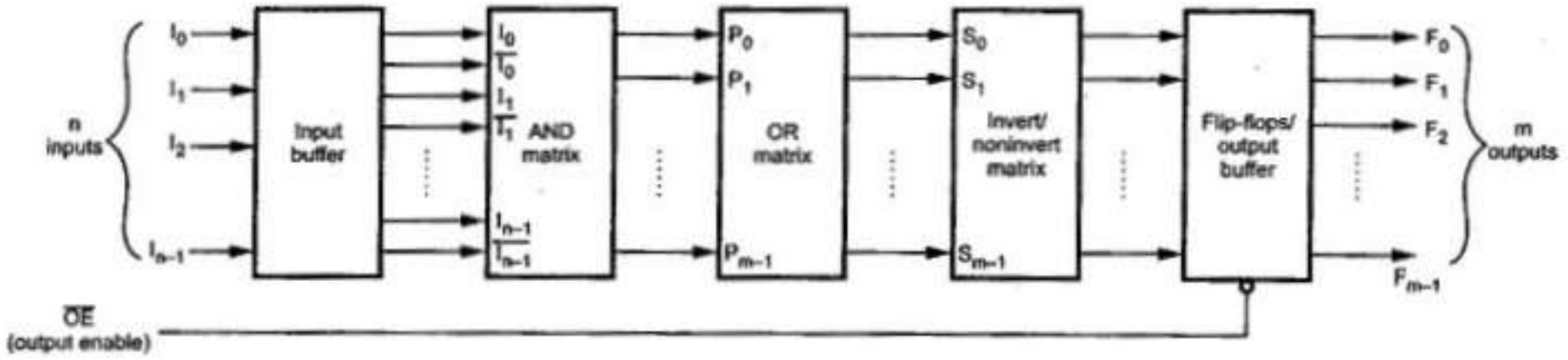



x_1 x_2 x_n
Input buffers and inverters
x_1 $\bar{x}_1$ x_n $\bar{x}_n$
AND plane
P_1
P_k
OR plane
f1
fm

PLA

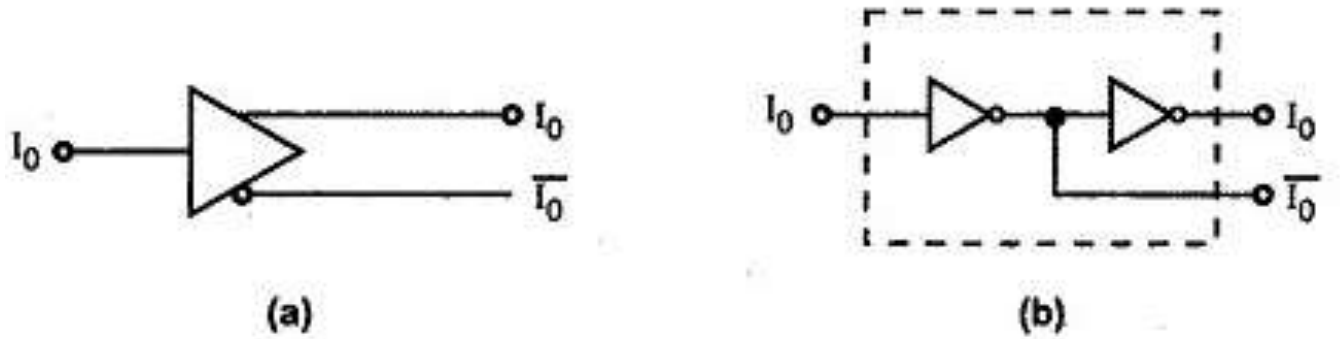
- PLA- It consists of AND –OR circuit. AND matrix can be used to implement product terms and OR matrix gives Sum of the products.
- PLA consists of
- Input buffers-They are of 2 types inverted and non-inverted. It is used for avoiding loading of sources.
- AND matrix- It is programmed to give desired product terms
- OR matrix- - It is programmed to give Sum of the products
- Invert –Non Invert matrix- EXOR gate or inverter is used to invert the output of OR matrix if required.
- Flip-flops /Output buffers- Used to increase current sourcing capability of PLA. Tristate buffers are used.

Block diagram

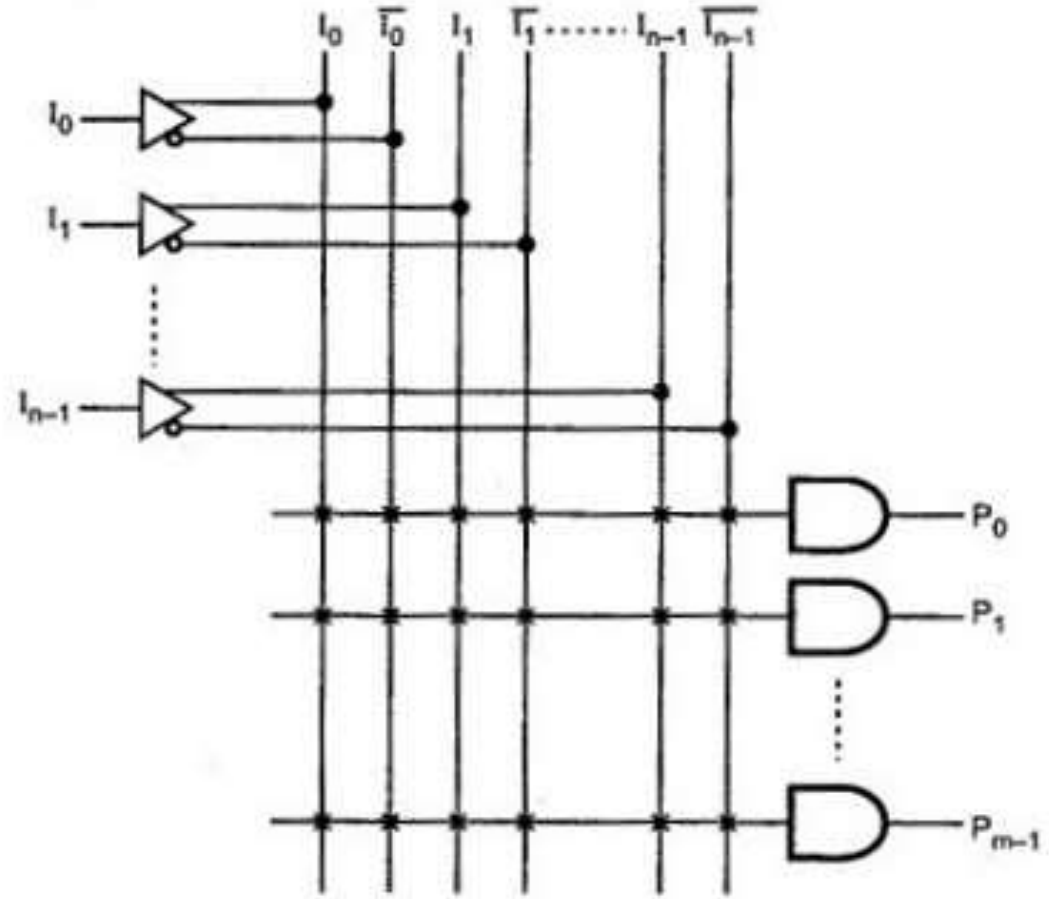


Input Buffers

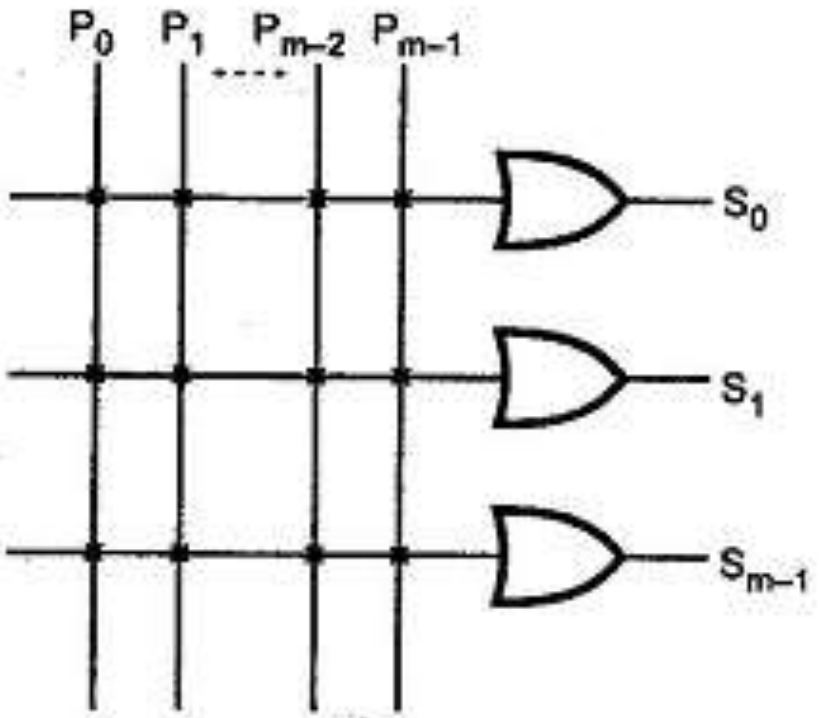
Input Buffers: Input buffers are provided in the Programmable Logic Array to limit loading of the sources that drive the inputs. They also provide inverted and non-inverted form of inputs at its output.



AND Matrix

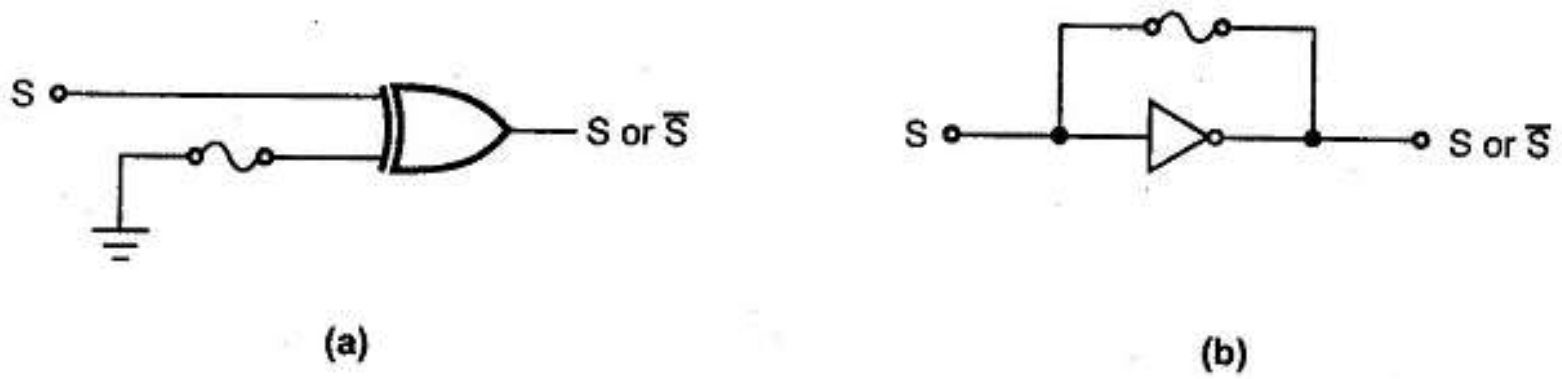

I_0
$\overline{I_0}$
I_1
$\overline{I_1}$
I_{n-1}
$\overline{I_{n-1}}$
I_0
I_1
I_{n-1}
P_0
P_1
P_{m-1}

OR Matrix



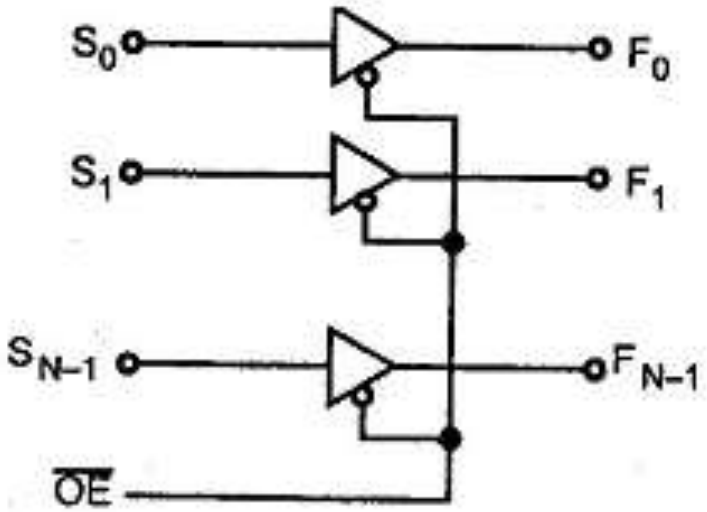
Inverting /Non inverting Matrix

Inverting /Non inverting Matrix: Invertible and Non Invertible Matrix provides output in the complement **or** un complemented form. The user can program the output in either complement or un-complement form as per design requirements. In both the cases if fuse is intact the output is in its un complemented form ; otherwise output is in the complemented form

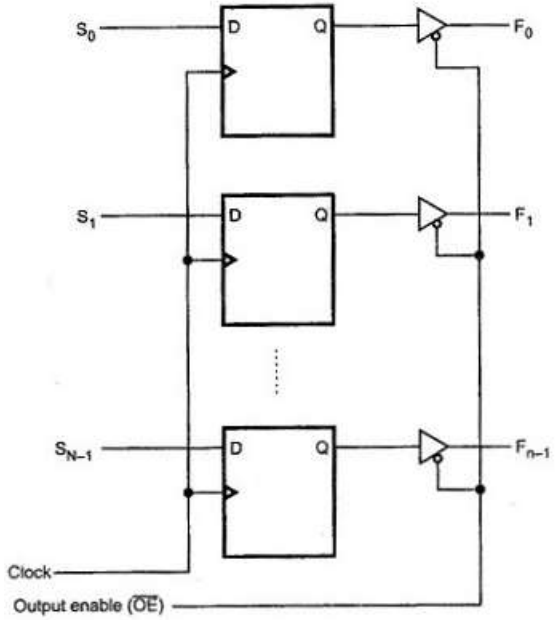


A =FUSE	B= S	Y
0	0	0 =S
0	1	1=S
1	0	1=S’
1	1	0=S’

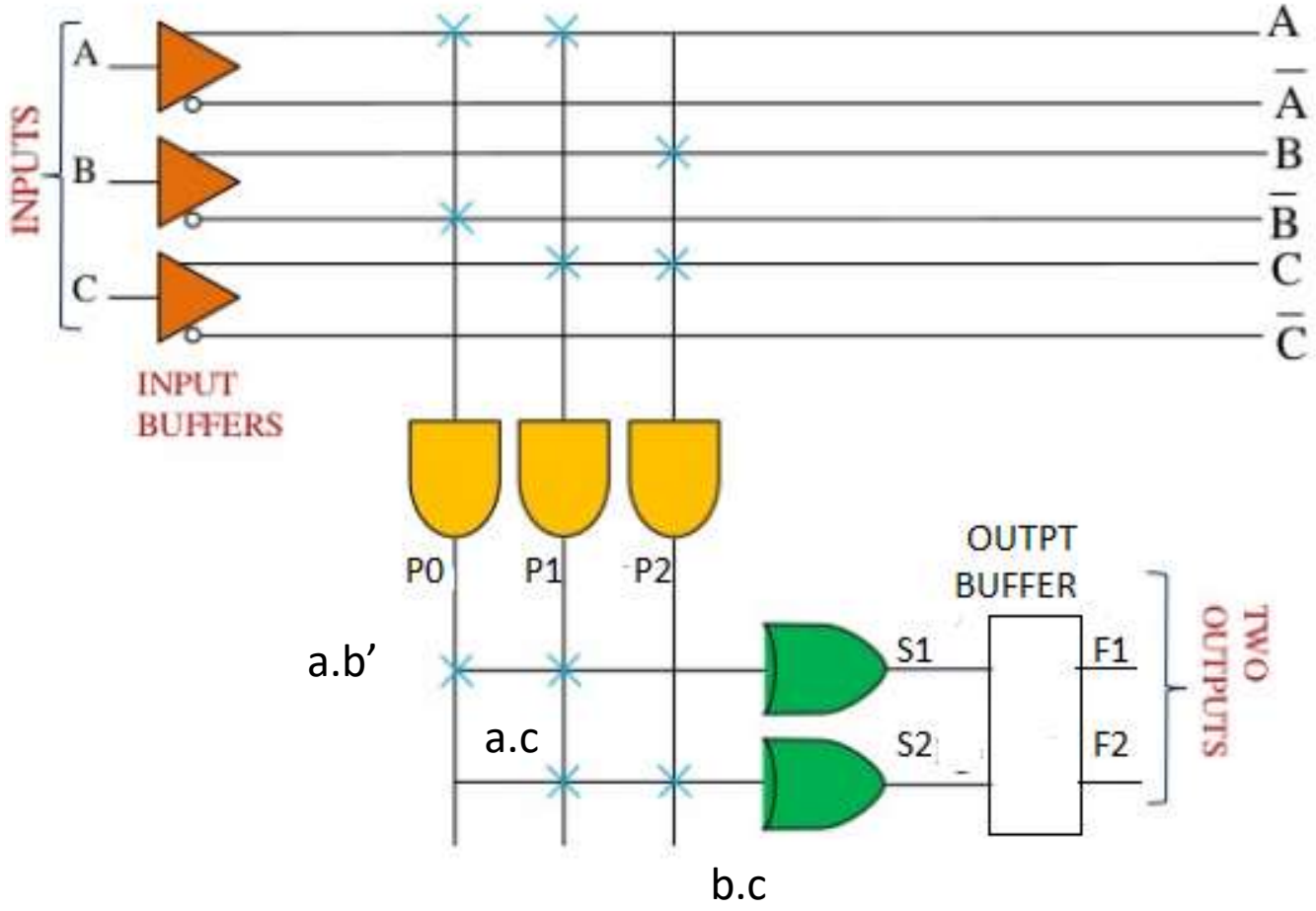
Output buffer:- The driving capacity of PLA is increased by providing buffers at the output. They are usually TTL compatible.



Flip flop:- For the implementation of sequential circuits we need memory elements, flip-flops and combinational circuitry for deriving the flip-flop inputs. To satisfy both the needs some PLAs are provided with flip-flop at each output



Draw PLA with 3 inputs, 3 product terms and 2 output terms



EXAMPLE -1: A combinational circuit is defined by the following function

$F_1(A,B,C) = \sum m(4,5,7)$

$F_2(A,B,C) = \sum m(3,5,7)$

Implement this circuit with a PLA having 3 inputs, 3 product terms and 2 outputs. Also write the PLA programming table.

SOLUTION:

STEP 1: Write the Boolean Expression in minimum SOP form.

(Note: use minimization technique. Here we use K-map)

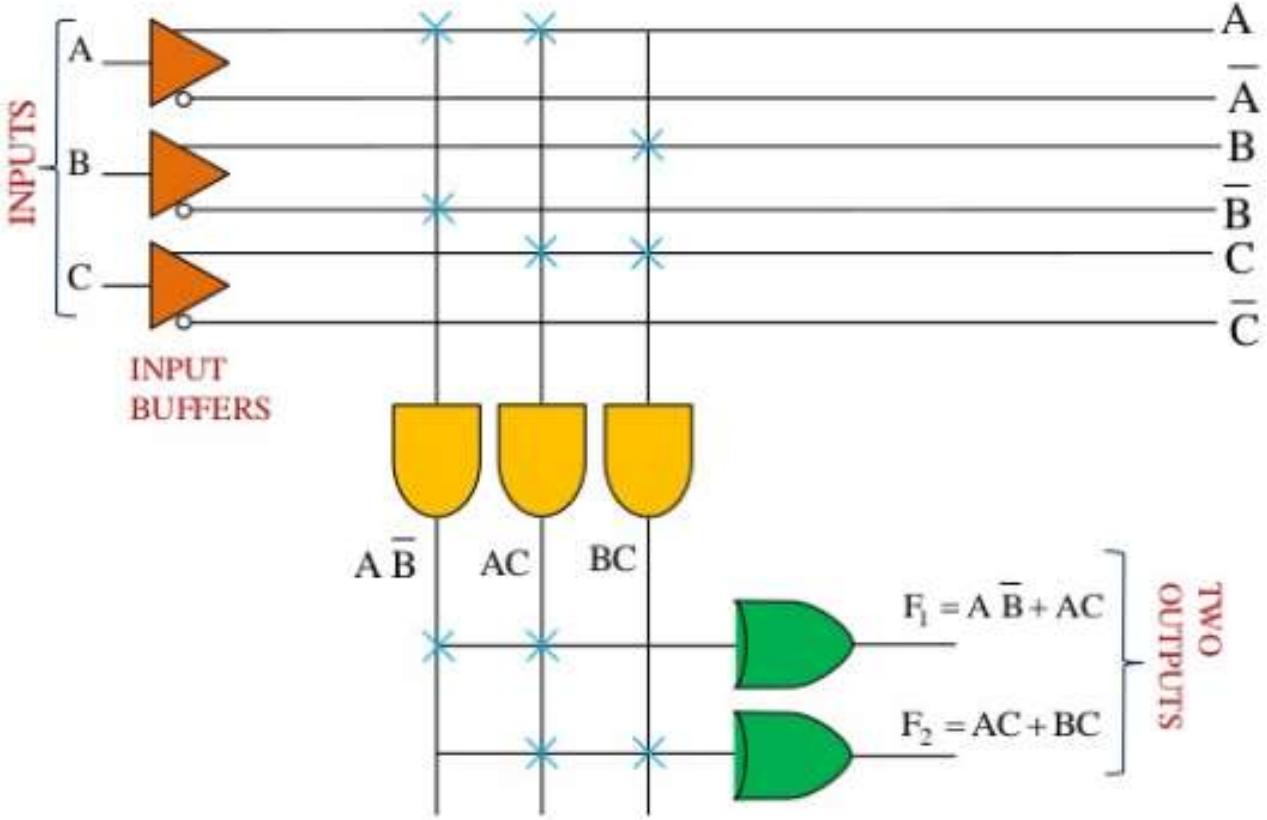
A \ BC	00	01	11	10
0	0	0	0	0
1	1	1	1	0

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

A \ BC	00	01	11	10
0	0	0	1	0
1	0	1	1	0

$F_2 = AC + BC$

- Number of inputs = 3
- Number of product terms = 4, in which 'AC' is common in both function F_1 and F_2 , so number of product terms = 3
- Number of AND gates = Number of product terms = 3
- Number of OR gates = Number of outputs = 2


INPUTS
A
B
C
INPUT
BUFFERS
A
$\overline{A}$
B
$\overline{B}$
C
$\overline{C}$
A $\overline{B}$
AC
BC
$F_1 = A\overline{B} + AC$
$F_2 = AC + BC$
TWO
OUTPUTS

EXAMPLE 2: A combinational logic is defined by function

$F_1(A,B,C)=\sum m\ (3,5,6,7)$

$F_2\ (A,B,C)=\sum m\ (0,2,4,7)$

Implement the circuit with PLA having 3 inputs, 4 product terms and 2 outputs.

SOLUTION: STEP 1: Write the function in minimize SOP form.

(Note:Here we are using K-map)

For F_1

A \ BC	00	01	11	10
0	0	0	1	0
1	0	1	1	1

$F_1 = AC + AB + BC$

FOR F_2

A \ BC	00	01	11	10
0	1	0	0	1
1	1	0	1	0

$F_2 = \overline{B}\,\overline{C} + ABC + \overline{A}\,\overline{C}$

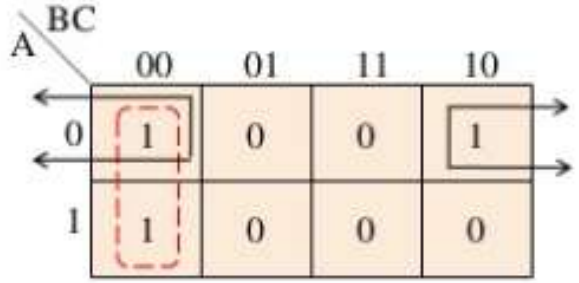
- There are six product terms in F_1 and F_2, but only 4 product terms are allowed to use.
- Now implement $\overline{F_1}(A,B,C)$

$$F_1(A,B,C) = \sum m(3,5,6,7)$$

$$\overline{F_1}(A,B,C) = \sum m(0,1,2,4)$$

$$\text{But } F_2(A,B,C) = \sum m(0,2,4,7)$$

- From the last two equation it is clear that the minterms 0, 2 and 4 are common.
- Obtain the minimized expression by using them.



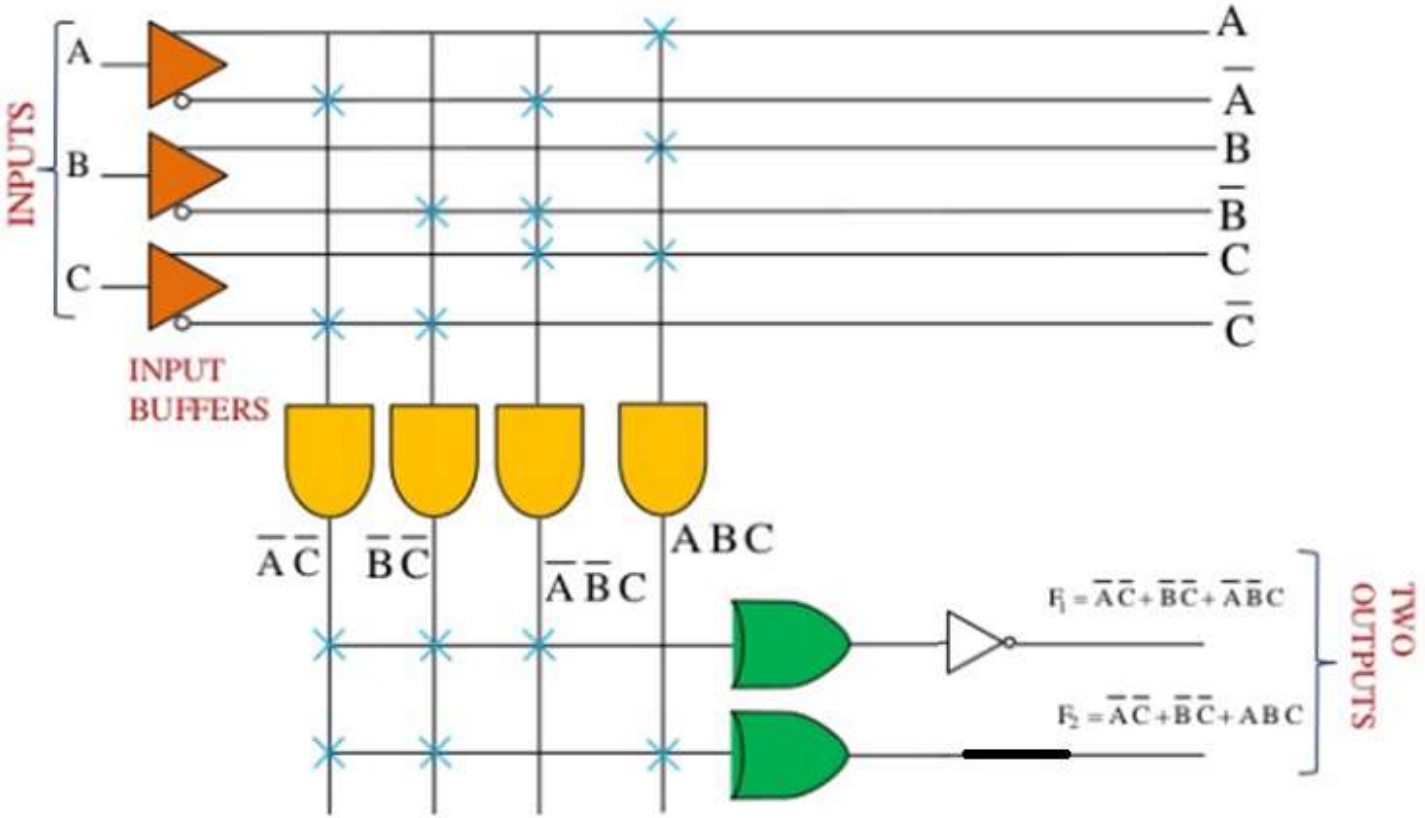
$$\sum m(0,2,4) = \overline{A}\,\overline{C} + \overline{B}\,\overline{C}$$

- Remaining minterms are $m(1) = \overline{A}\,\overline{B}\,C$ and $m(7) = A\,B\,C$
- Now four product terms are $\overline{A}\,\overline{C}, \overline{B}\,\overline{C}, \overline{A}\,\overline{B}\,C, A\,B\,C$
- $\overline{A}\,\overline{C}, \overline{B}\,\overline{C}$ are common in both functions.

$$\therefore\ F_1 = \overline{A}\,\overline{C} + \overline{B}\,\overline{C} + \overline{A}\,\overline{B}\,C$$

$$F_2 = \overline{A}\,\overline{C} + \overline{B}\,\overline{C} + A\,B\,C$$

Note: After OR gate for F_1 , $\overline{F_1}$ connect to inverter to get F_1


INPUTS
A
B
C
A
$\overline{A}$
B
$\overline{B}$
C
$\overline{C}$
INPUT BUFFERS
$\overline{A}\,\overline{C}$
$\overline{B}\,\overline{C}$
$\overline{A}\,\overline{B}\,C$
ABC
$F_1 = \overline{A}\,\overline{C} + \overline{B}\,\overline{C} + \overline{A}\,\overline{B}\,C$
$F_2 = \overline{A}\,\overline{C} + \overline{B}\,\overline{C} + ABC$
TWO OUTPUTS