Unit IV
Programmable Logic Devices

by
Prof. Sujata Wakchaure

Introduction

- To design a circuit, designer can select most appropriate IC from the available ICs for the circuit

- The design may have to be modified to meet the special requirements of these devices

- **Advantages of design method (which uses Fixed Function IC):**
  - Low development
  - Fast turn around of design
  - Relatively easy to test the circuit

- **Disadvantages of design method (which uses Fixed Function IC)**
  - Large board space requirement
  - Large power requirement
  - Lack of security
  - Additional cost, space, power requirement, etc.

Introduction (Contd..)

- **ASIC (Application Specific Integrated Circuit)**

- **Advantages of ASIC :**
  - Reduced space requirement
  - Reduced power requirement
  - Design implemented in this form are almost impossible to copy
  - Low Cost

- **Disadvantages of ASIC :**
  - Initial development cost may be enormous
  - Testing methods may have to be developed

Programmable Logic Devices (PLD)

- Special purpose ICs
- It is user configurable and is capable of implementing logic functions
- PLDs can be programmed as per requirement
- It is a VLSI chip

PLDs (Contd..)

- The purpose of a PLD device is to permit elaborate digital logic designs to be implemented by the user in a single device

- PLDs Can be erased electrically and reprogrammed with a new design

- **Advantages of fixed function ICs :**
  - Short design cycle
  - Low development cost

PLDs (Contd..)

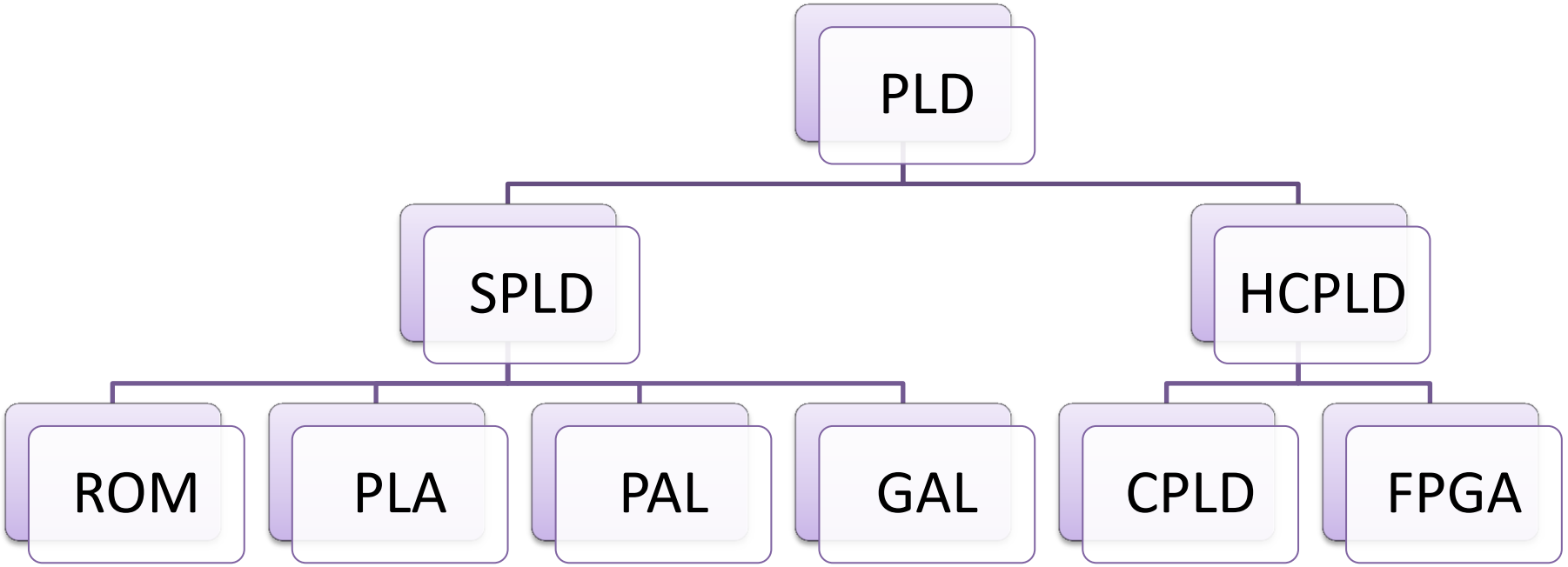
- **Advantages over Fixed function ICs :**
  - Reduction in board space requirement
  - Reduction in power requirement
  - Design security
  - Compact circuitry
  - Higher switching speed

- **Advantages of ASIC :**
  - High densities
  - Low quantity production cost
  - Reduced space requirement

PLDs (Contd..)

- **The Architecture and various other features of PLDs are:**
  - ROM
  - Programmable Logic Arrays (PLA)
  - Programmable Array Logic (PAL)
  - Sample Programmable Logic Devices (SPLDs)
  - Complex Programmable Logic Devices (CPLDs)
  - Field Programmable Gate Arrays (FPGA)

Types of PLDs (Cont.)



PLDs (Contd..)

- The differences between the first three categories are these:
  1. **ROM:** In a ROM, the input connection matrix is hardwired. The user can modify the output connection matrix.
  2. **PAL:** In a PAL / GAL (Generic Array Logic) the output connection matrix is hardwired. The user can modify the input connection matrix.
  3. **PLA:** In a PLA the user can modify both the input connection matrix and the output connection matrix.

Device	AND-array	OR-array
PROM	Fixed	Programmable
PLA	Programmable	Programmable
PAL	Programmable	Fixed
GAL	Programmable	Fixed

ROM as PLD

- ROM is basically a combinational circuit which has n I/Ps & m O/Ps

- A ROM size M X N has M number of locations and N no. of bits can be stored at each location

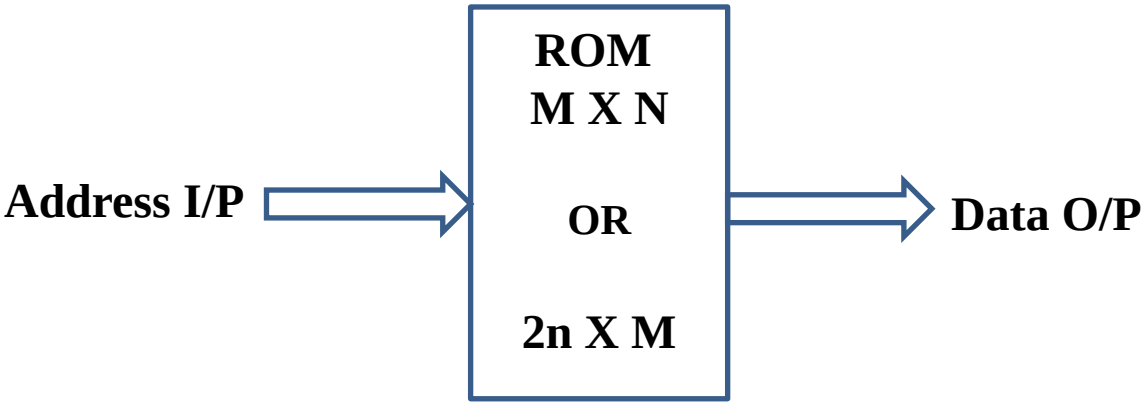
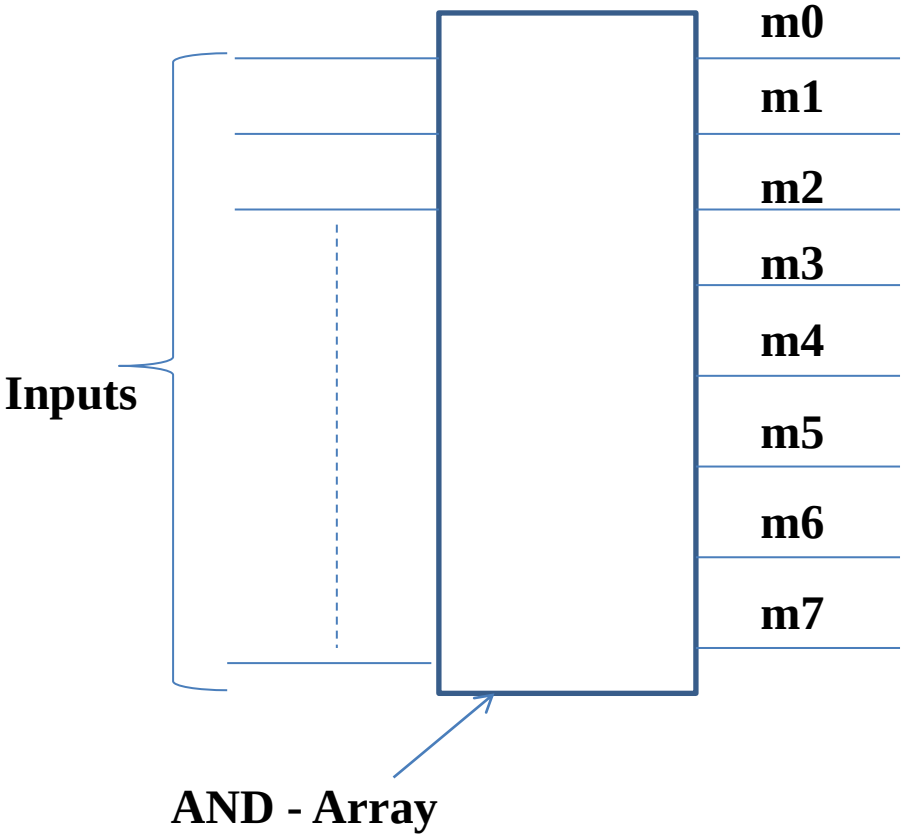


Fig 1: ROM as a Combinational Circuit

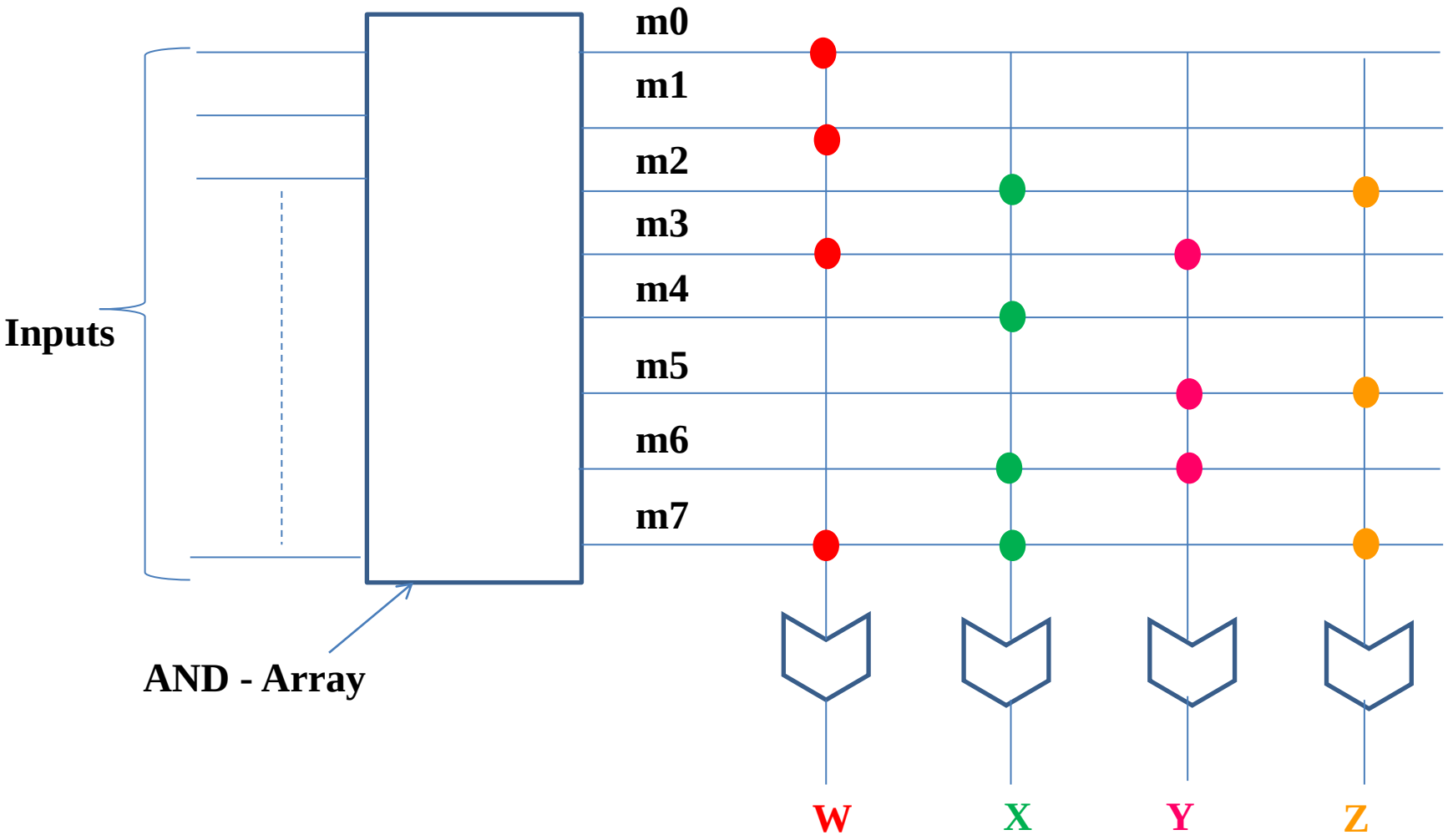
ROM as PLD (Contd..)



E.g.

1. **W = m0+m1+m3+m7**
2. **X = m2+m4+m6+m7**
3. **Y = m3+m5+m6**
4. **Z = m2+m5+m7**

ROM as PLD (Contd..)



ROM as PLD (Contd..)

- **Advantages of using ROM as a PLD:**
  - Ease of design
  - Design can be changed or modified rapidly
  - It is usually a faster circuit
  - Cost is reduced

- **Disadvantages:**
  - Increased power requirement
  - Enormous increase in size

Programmable Logic Array (PLA)

- It is a type of fixed architecture logic device with programmable AND gates followed by programmable OR gates

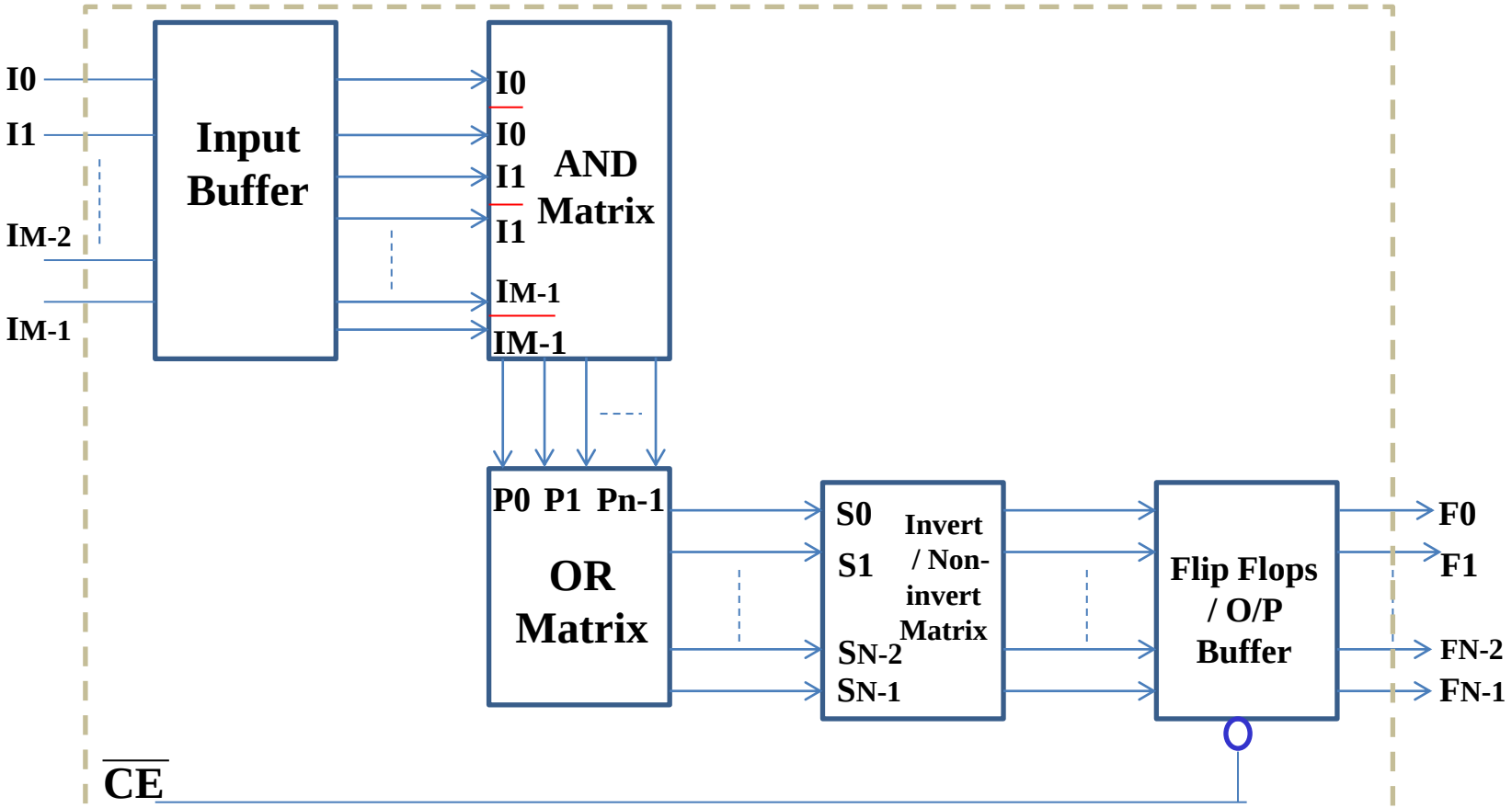
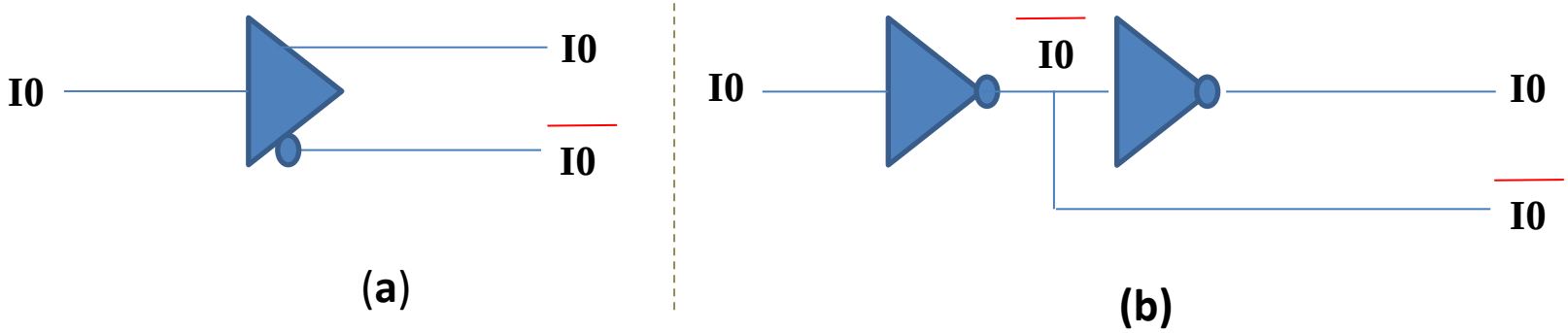


Fig 2: Block Diagram of PLA Device

PLA (Contd..)

1. **Input Buffer :**
   – Inputs are required to limit loading of sources



PLA (Contd..)

2. AND Matrix :

- It IS USED TO FORM Product terms
- **$P = I0.\overline{I0} + I1.\overline{I1} + \ldots\ldots + I_{M-1}.\overline{I_{M-1}}$**

3. OR Matrix :

- It is used to produce the logical sum of product term O/P of AND Matrix
- **$S0 = P0 + P1 + \ldots + Pn-1$**

PLA (Contd..)

4. Invert / Non-Invert Matrix:

- This is programmable buffer that can be set for inverting & non inverting operations corresponding to active-low or active high O/P respectively
- E.g. In case of EX-OR gate, if the fuse is not damage (intact), the O/P is S and if fuse is damaged O/P is $\overline{S}$
- In Fig b S or $\overline{S}$ depending upon whether the fuse is intact or open

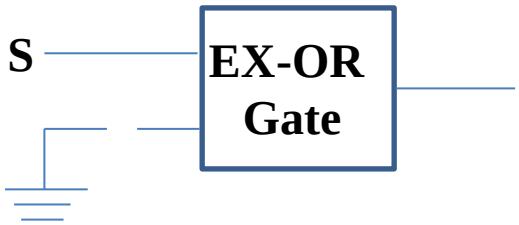


Fig (a)

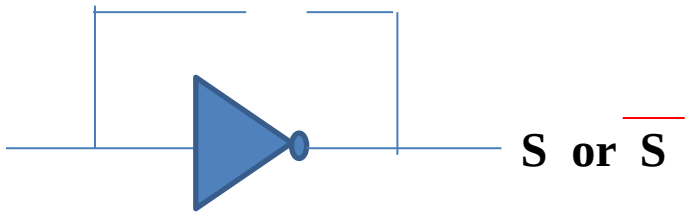


Fig (b)

PLA (Contd..)

- Example:

A	B	C	Y1	Y2
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	0	1
1	0	0	1	0
1	0	1	1	1
1	1	0	0	0
1	1	1	1	1

PLA (Contd..)

- **Step1:** To make truth table or observe truth table
- **Step 2:** Find out the minimal SOP form for given function (i.e. for Y1 & Y2)

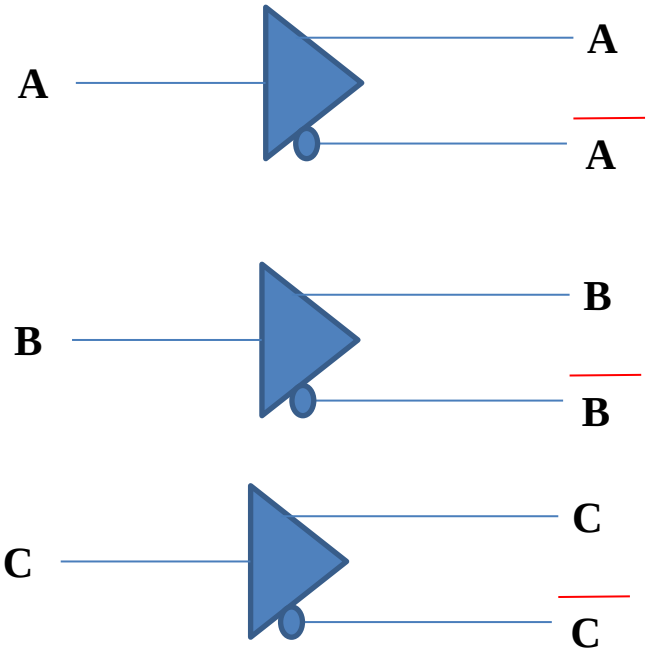
$$Y1 = A\bar{B}\bar{C} + A\bar{B}C + ABC$$

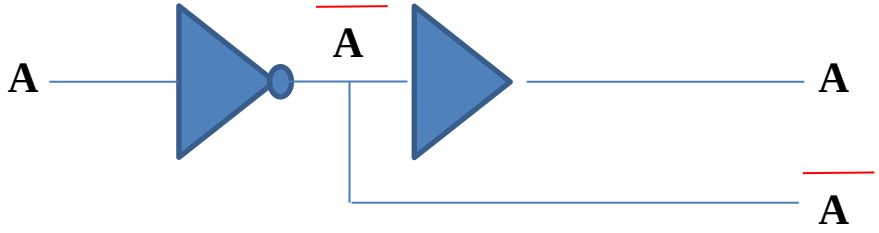
$Y1 = A\bar{B} + AC$

$Y2 = BC + AC$

PLA (Contd..)

- **Step 3:** Find out no. of I/P Buffer
  - What is I/P Buffer ?





No. of I/P Buffer = No. of Variables

No. of I/P Buffer = 3

PLA (Contd..)

- **Step 4:** Find out the no. of programmable AND gate

No. of programmable AND Gate = No. of Minterms

**** No. of Minterms should not be repeated**

1 2

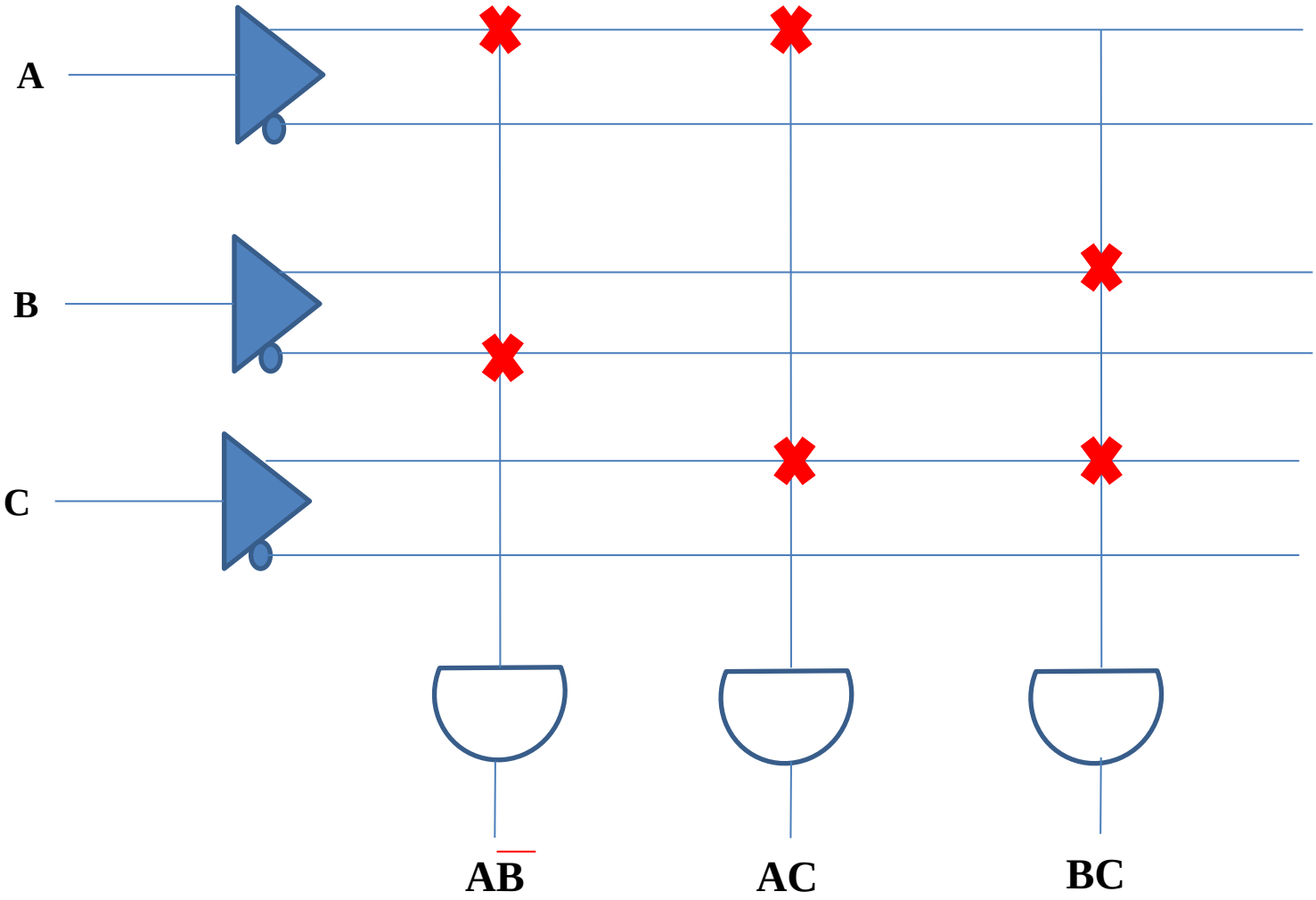
Y1 = AB + AC

3 4

Y2 = BC + AC

No. of programmable AND Gate = 3 (Because AC is repeated)

PLA (Contd..)



PLA (Contd..)

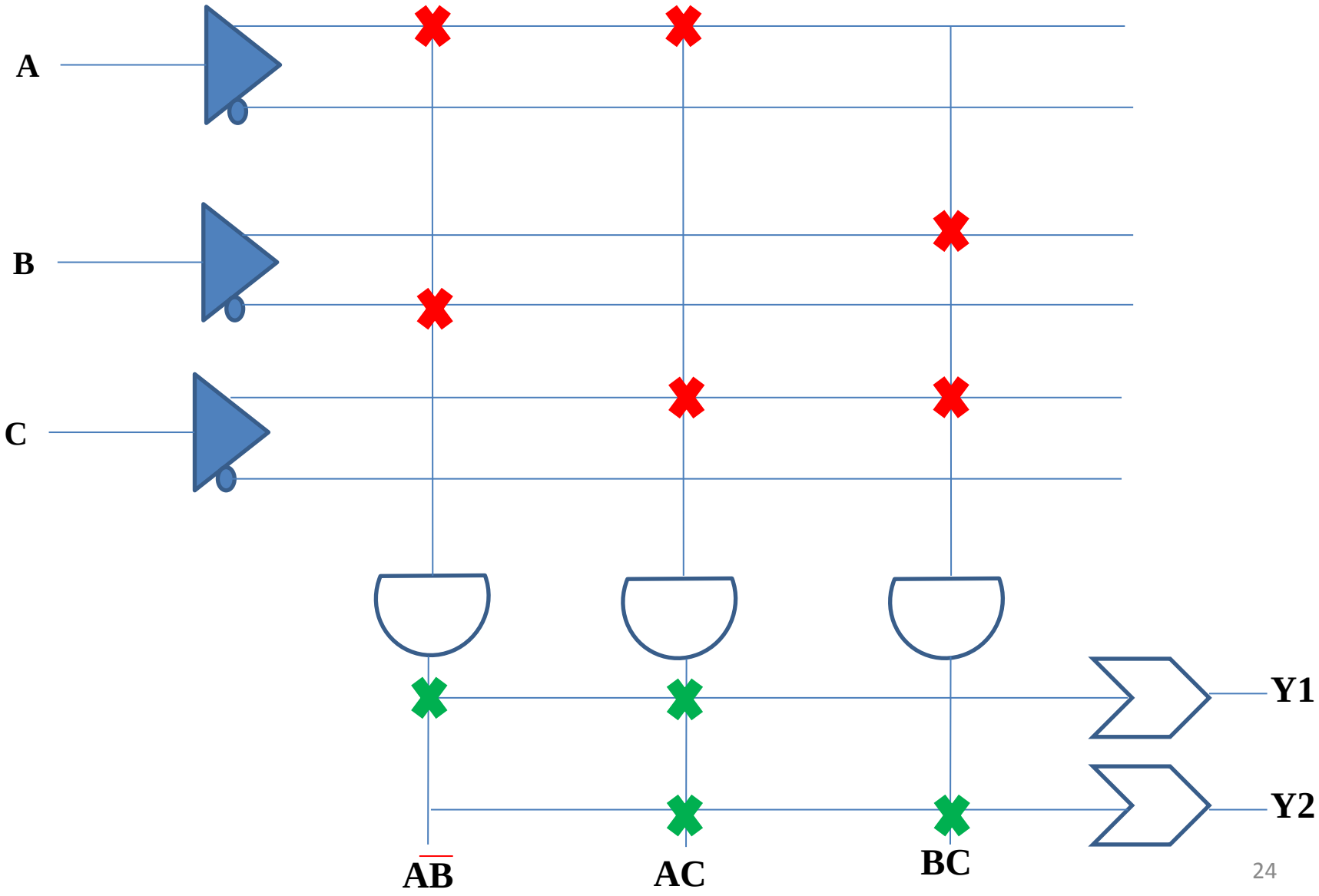
- **Step 5:**

  Find out the no. of programmable OR gate

  No. of programmable OR Gate = No. of Functions

  = 2 (Y1 & Y2)

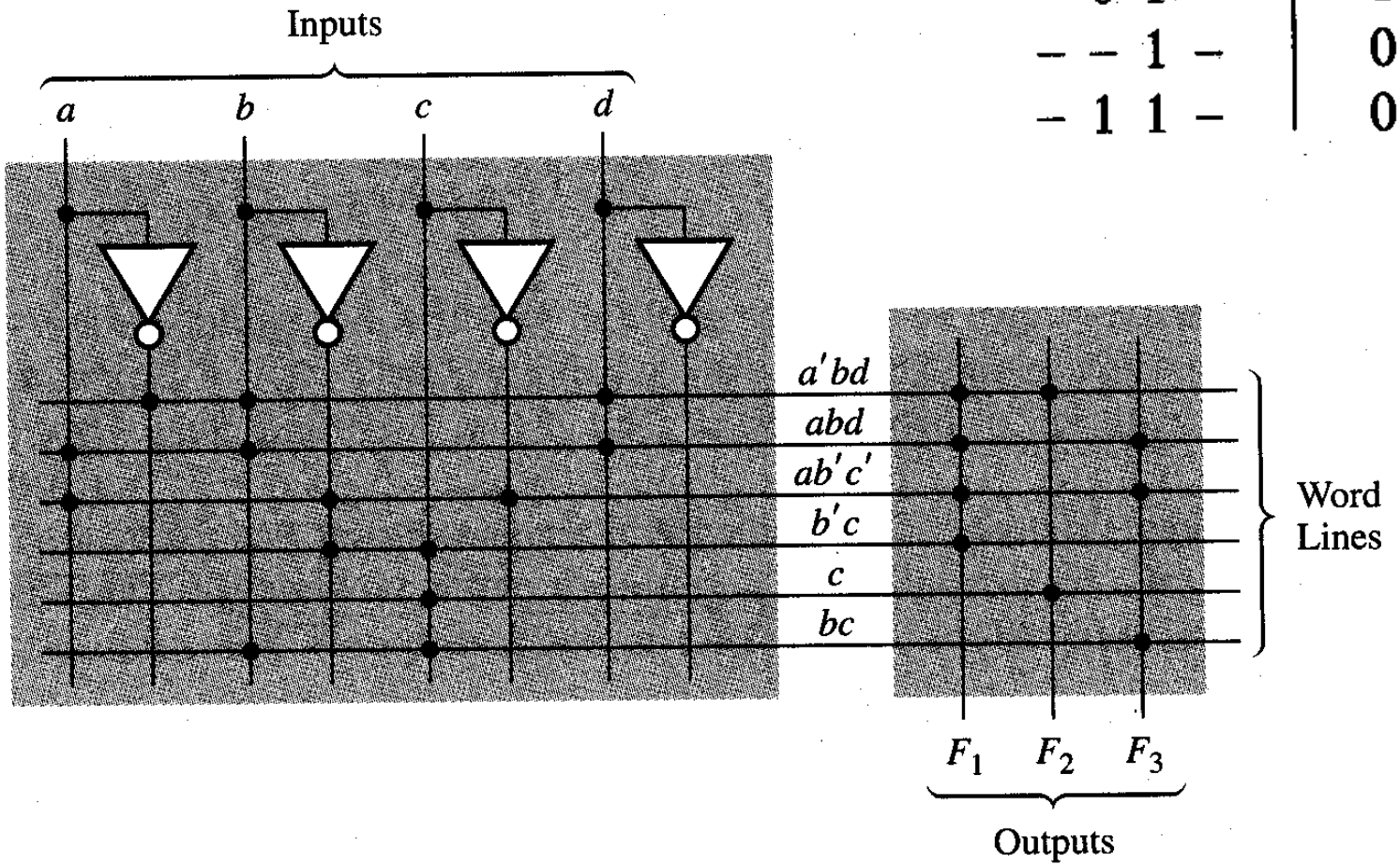
PLA (Contd..)



PLA Realization of Functions

$F_1 = a'bd + abd + ab'c' + b'c$
$F_2 = c + a'bd$
$F_3 = bc + ab'c' + abd$

a	b	c	d	f_1	f_2	f_3
0	1	–	1	1	1	0
1	1	–	1	1	0	1
1	0	0	–	1	0	1
–	0	1	–	1	0	0
–	–	1	–	0	1	0
–	1	1	–	0	0	1



PLA (Contd..)

- **H.W.**

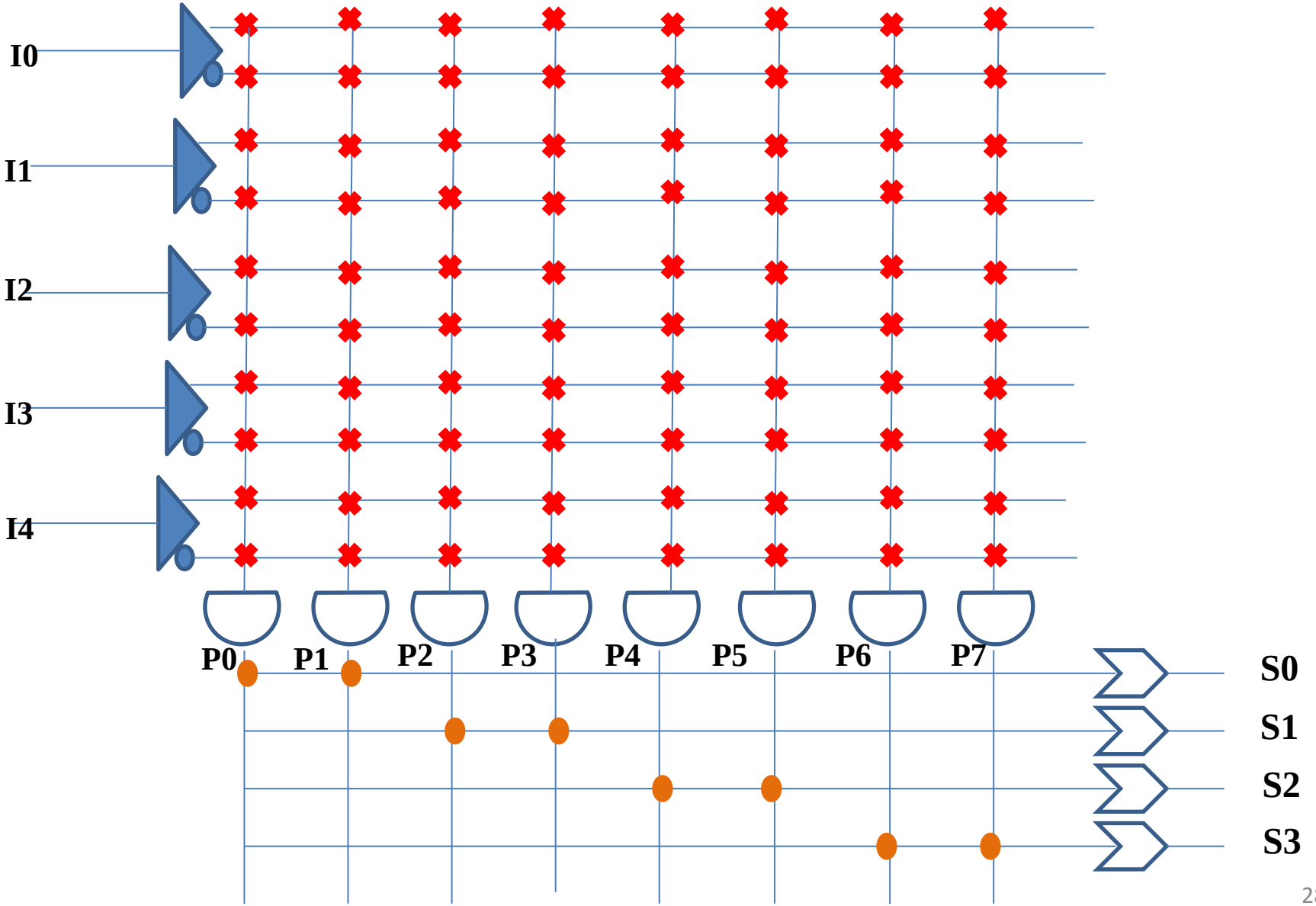
1. **Binary to Gray**

2. **F1(A,B,C)=∑m(4,5,7)**
   F2(A,B,C)=∑m(3,5,7)

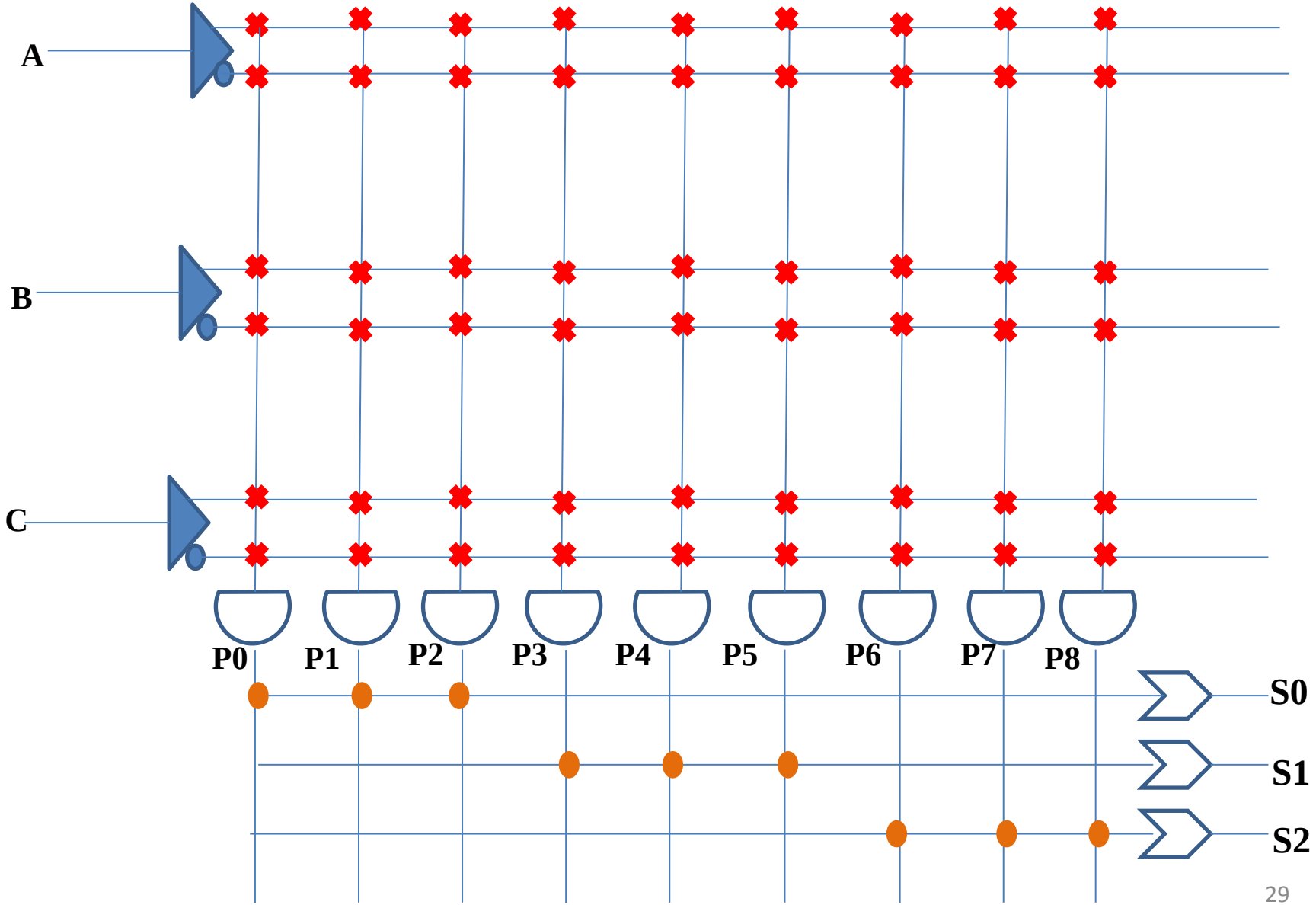
Programmable Array Logic (PAL)

- A most commonly used type of PLD
- It have fixed OR array
- Advantage of Fixed OR array is Simplicity
- Disadvantage is Flexibility

PAL (Contd..)



PAL (Contd..)

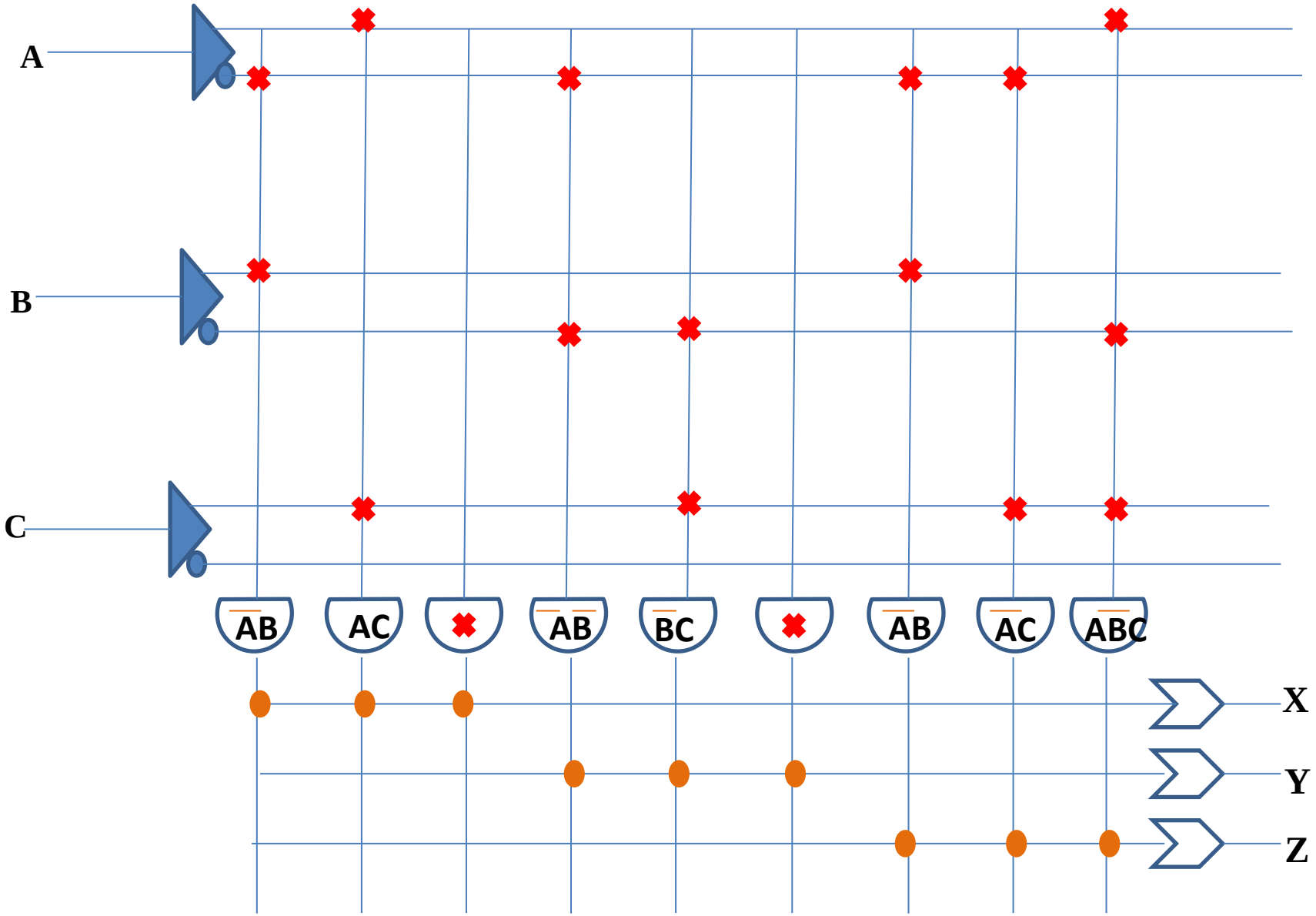


PAL (Contd..)

- Example
  1. X(A,B,C) = ∑ m(2,3,5,7) = $\bar{A}B + A\bar{C}$
  2. Y(A,B,C) = ∑ m(0,1,5) = $\bar{A}B + B\bar{C}$
  3. Z(A,B,C) = ∑ m(0,2,3,5) = $\bar{A}B + A\bar{C} + AB\bar{C}$

- Step 1: See the Minterms
- Step 2: Minimize expression using K-Map
- Step 3: Look at the variable i.e. 3. So 3 I/P buffers required
- Step 4: No. of AND arrays
- Step 5: No. of OR gates

PAL (Contd..)

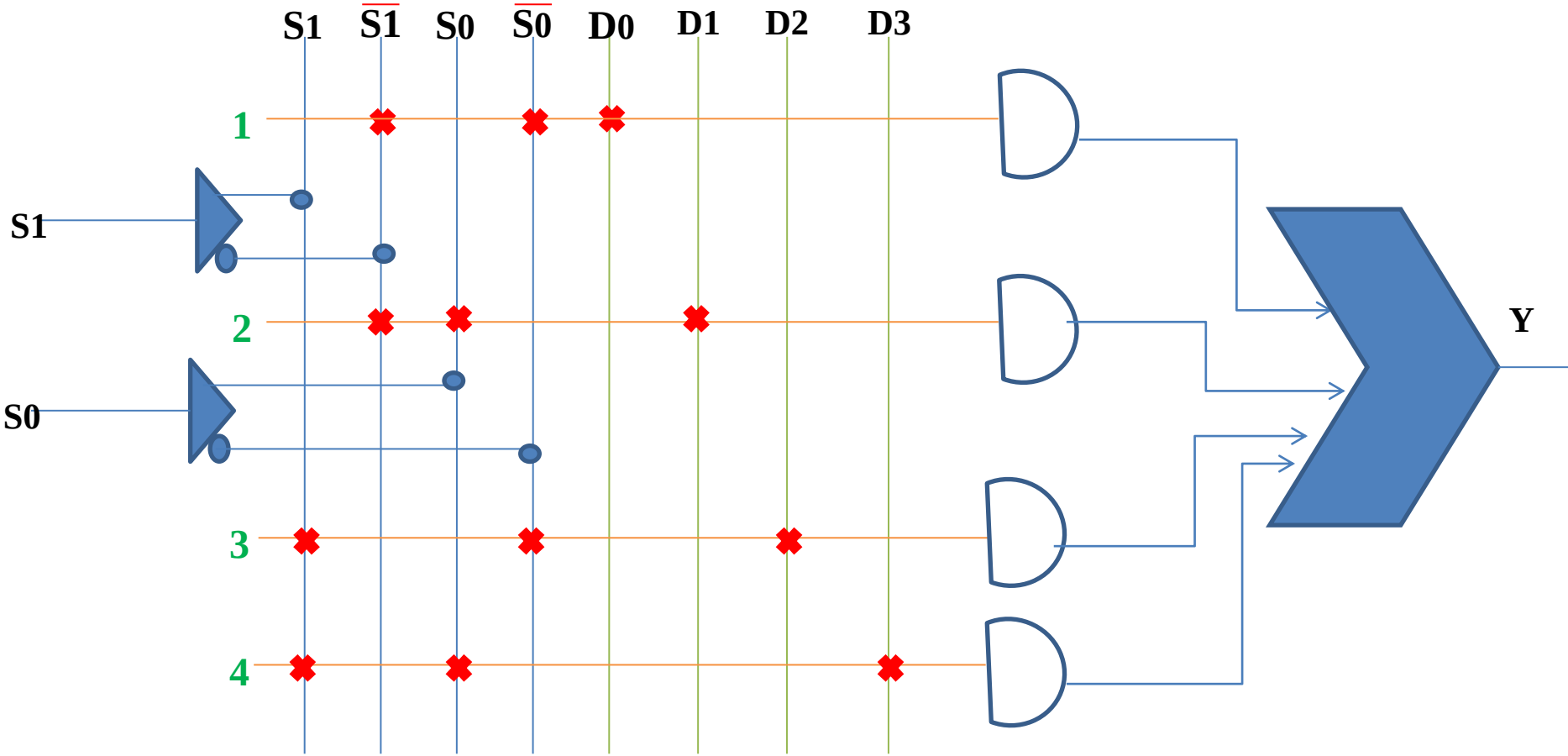


Implementation of 4:1 MUX using PAL

Select inputs		Outputs
S1	**S0**	**Y**
0	0	D0
0	1	D1
1	0	D2
1	1	D3

$$Y = \bar{S}_1\bar{S}_0D_0 + \bar{S}_1S_0D_1 + S_1\bar{S}_0D_2 + S_1S_0D_3$$

4:1 MUX (Contd..)



PAL (Contd..)

- **Examples**

1. **BCD to Excess-3**

2. **F1(A,B,C) = ∑m(0,1,5,6,7)**

Difference between PROM, PLA & PAL

PROM	PLA	PAL
It is cheap and easy to use	It is expensive than PAL and PROM and complicated to use	Moderately expensive and moderately complicated
There are fixed AND arrays and programmable OR arrays	AND arrays and OR arrays Are programmable	Only the AND array is programmable. Or array is fixed
SOP function in standard form only can be implemented	Any SOP function can be implemented	Any SOP function can be implemented
It is possible to decode any minterm	We can get any desired minterm by programming the AND matrix	We can get any desired minterm by programming the AND matrix

FPGA

- FPGA : Field Programmable Gate Arrays
- FPGA is a IC which is used to increase the effective size and to add more functionality in a single programmable device
- FPGAs are configurable and programmable
- Programmable components of FPGA
  1. Logic Blocks
  2. Interconnects

FPGA (Contd..)

- What is the term "**Field**" programmable?
  - Field stands for programming in the field.

- **FPGA generally contains**
  - PLDs
  - Logic gates
  - RAM
  - Other H/W components

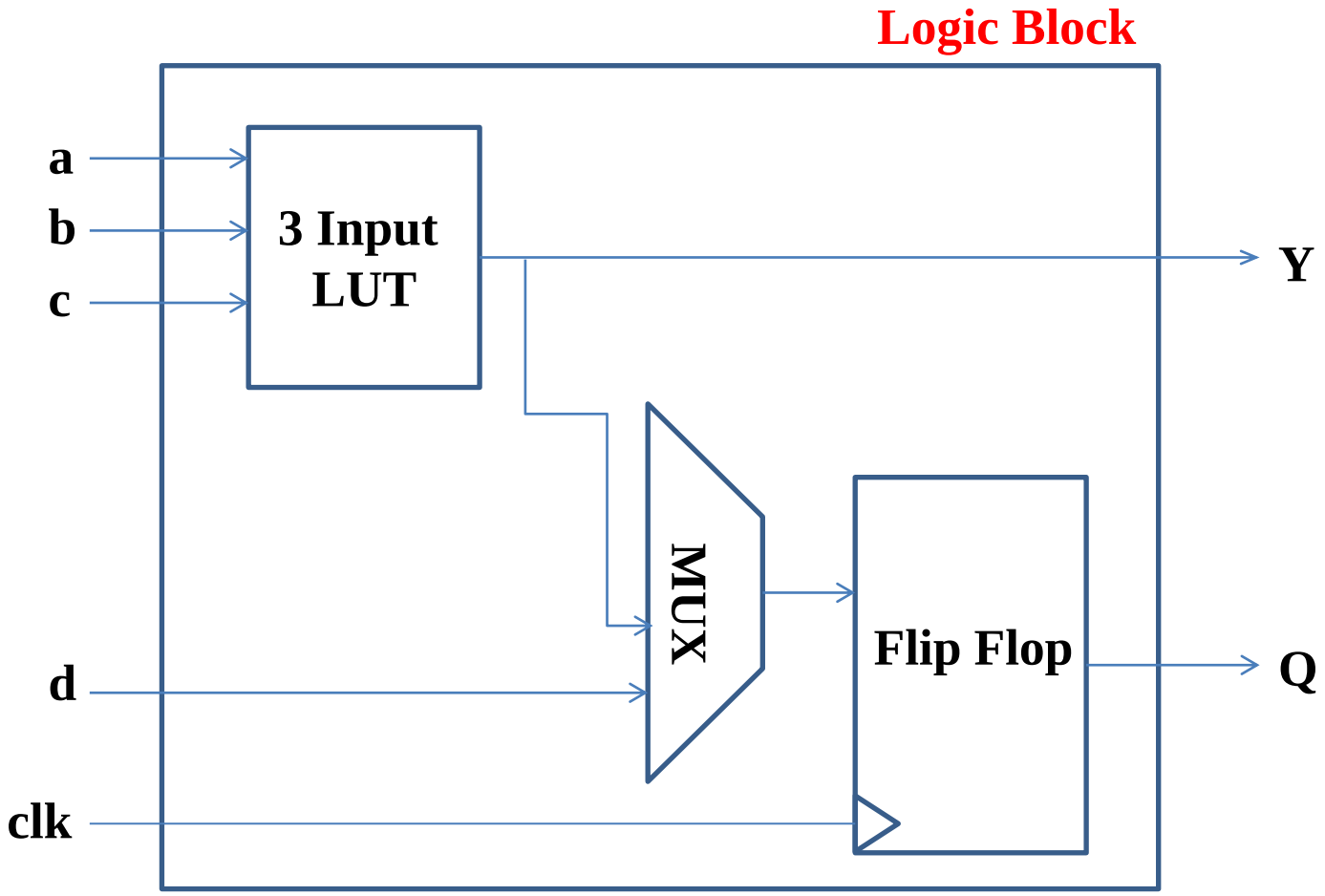
- **Family of FPGA**
  - Xilinx
  - Actel
  - Altera

FPGA (Contd..)

- **Advantages of FPGA**
  - Inexpensive
  - Easy to realize
  - Less Design and Testing time

FPGA (Contd..)

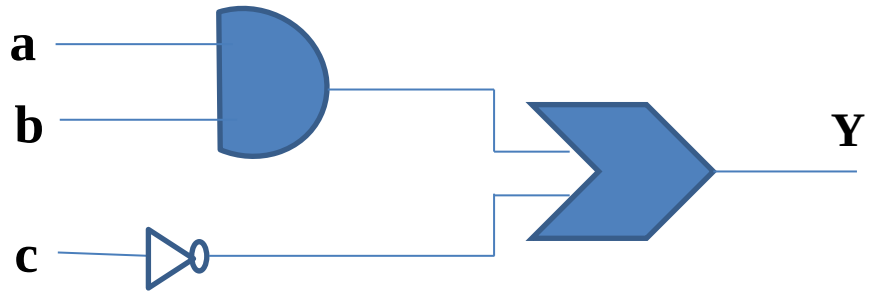
- **Basic Architecture of FPGA**



FPGA (Contd..)

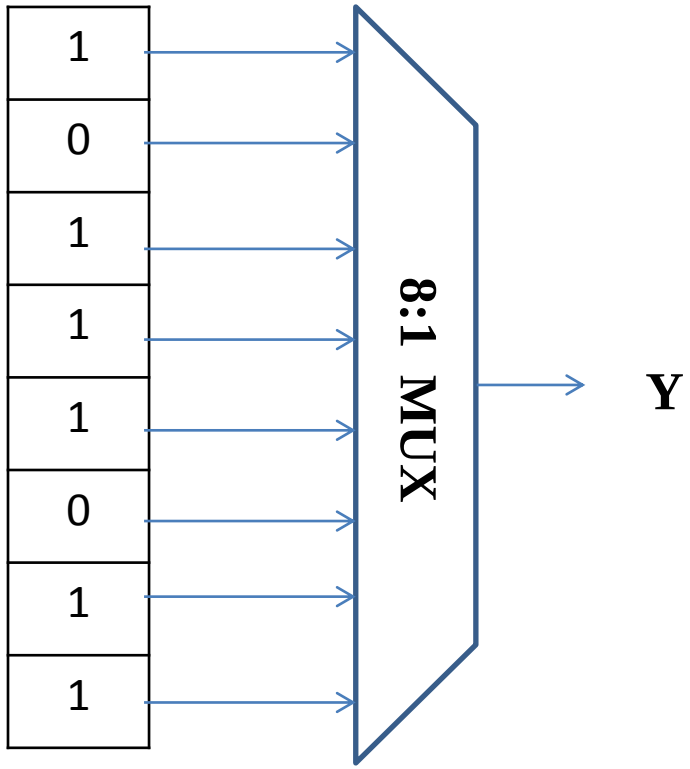
- **What does the Logic Block contains in FPGA ?**
  - Logic block of FPGA contains LUT (Look Up Table)

- **What is LUT?**
  - **E.g. Y=(a & b) | !c**

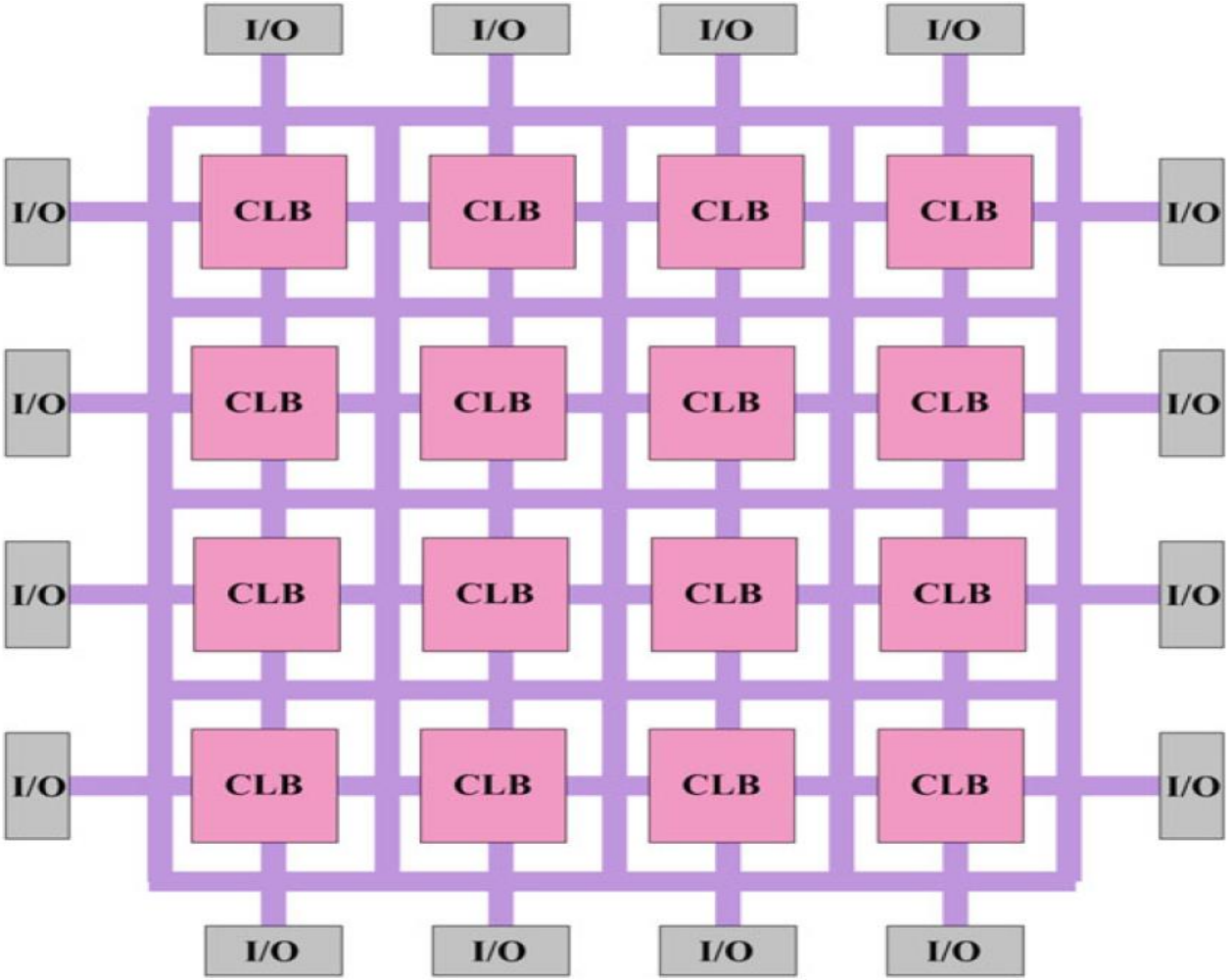


FPGA (Contd..)

a	b	c	Y
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1



Architecture of FPGA



Architecture of FPGA (Contd..)

- The basic architecture consist of array of configurable logic blocks

- **Logic blocks:**
  - Connected to the I/O blocks through common row/column programmable interconnects
  - The common row/column interconnects are known as Global interconnects