KONERU LAKSHMAIAH EDUCATION FOUNDATION

(Deemed to be University estd, u/s, 3 of the UGC Act, 1956)

(NAAC Accredited "A++" Grade University)

Green Fields, Guntur District, A.P., India – 522502

Department of Basic Engineering Science - II

I B.Tech. II Semester – CSE / AI & DS / ECE / EEE / CS & IT / IOT

A.Y.2024-25 - EVEN SEMESTER

Digital Design and Computer Architecture (23EC1202)

CO – 1: Combinational Digital Logic Circuits

Session 8: PAL and PLA design

1. Course Description (Description of the subject):

The course on "Digital Design and Computer Architecture" provides a comprehensive exploration of the foundational principles in digital design process and computer organization. Students explore the concepts of combinational and sequential circuits, memory circuits. The curriculum extends to the Basic computer architecture concepts, memory hierarchies, and input/output fundamentals, fostering a deep understanding of computer organization. Through practical projects and simulations, students develop the skills to design and implement digital circuits. Graduates emerge with a robust skill set, ready to embark on careers in hardware design, computer architecture, and related fields, equipped to contribute to the ever-evolving landscape of digital technology.

2. Aim of the Course:

The course aims to equip students with the knowledge and skills related to:

i. Proficiency in designing and optimizing Combinational and Sequential Circuits using Boolean algebra and programmable logic devices with a solid foundation in digital design.

ii. Skill development using hands-on experience in designing digital circuits which includes latches, flip-flops, and counters in combination with memory, registers, and timing and sequence control modules using hardware & modeling tools.

iii. Explore the architecture of modern computers, including the organization and structure of central processing units, memory systems, and input/output interfaces.

iv. Bridge theoretical concepts with real-world applications by examining case studies and examples of digital design and computer architecture in modern computing systems.

Overall, the aim of the course is to prepare the student well-equipped to apply their knowledge to the design and analysis of digital systems and computer architectures, preparing them for careers in areas such as hardware design, computer engineering, and embedded systems development.

3. Instructional Objectives (Course Objectives):

The course objectives for "Digital Design and Computer Architecture" typically include:

i. To Understand and apply foundational concepts in digital design which results in proficiency over designing and analyzing combinational and sequential logic circuits.

ii. To Gain hands-on experience with industry-standard simulation and modeling tools, for verifying and testing digital designs.

iii. To analyze the architecture of a computer system, including the organization and operation of the CPU, memory hierarchy, and input/output subsystems.

iv. To apply digital design and computer architecture principles to solve real-world engineering problems and challenges by reinforcing theoretical knowledge with hands-on experience.

4. Learning Outcomes (Course Outcomes):

i. Able to build the combinational and programmable digital logic circuits using logic gates and optimization methods.

ii. Able to construct the sequential and memory circuits using flip-flops, demonstrating a comprehensive understanding of the principles governing clocked sequential logic.

iii. Able to organize computer architecture and instructions sequence through a grasp of the foundational principles that govern the organization and functioning of a computer system.

iv. Capable of modeling Memory Architecture and I/O Organization modules proficiently.

v. Able to develop and analyze the computer architecture modules using basic combinational, sequential and memory logics.

5. Module Description (CO - 1 Description):

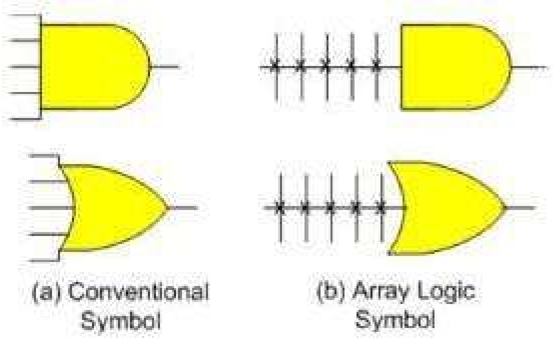
The module covers essential topics in digital electronics, starting with Boolean algebra and progressing to the representation and optimization techniques of digital logic using SOP/POS forms. Students will delve into the design of key components such as adders, subtractors, multiplexers, de-multiplexers, decoders, and encoders. The module introduces the concept of reversible gates, exploring their unique properties. Additionally, students will gain insights into Programmable Logic Devices (PLDs) like PROM, PAL, and PLA, understanding their design principles. The implementation of Complex Programmable Logic Devices (CPLDs) with macrocells and Field-Programmable Gate Arrays (FPGAs) featuring Configurable Logic Blocks (CLBs) and Look-Up Tables (LUTs) will be covered. Practical applications of these digital logic modules in various scenarios will be emphasized, providing students with a comprehensive understanding of digital electronics and its real-world applications.

6. Session Introduction:

An IC that contains large numbers of gates, flip-flops, etc. that can be configured by the user to perform different functions is called a Programmable Logic Device (PLD). Programmable Logic Devices (PLDs) are digital electronics devices that can be programmed to perform a wide variety of functions in digital circuits. Every Boolean logic can be decomposed into product-of-sum (POS) or sum-of-product.

The PLA is similar to PROM in concept except that PLA does not provide full decoding of the variable 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.

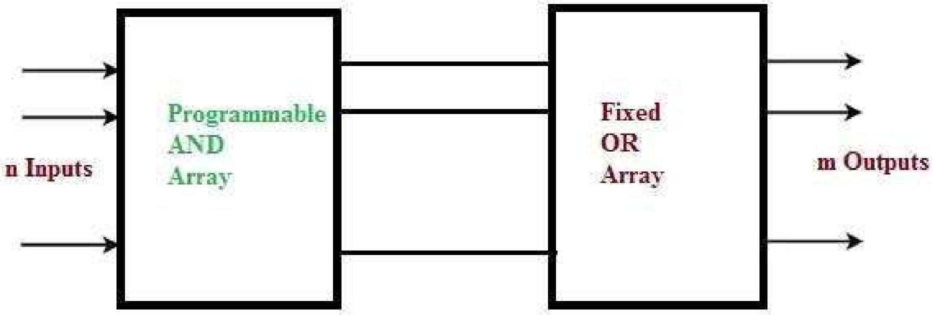
Figure shows the conventional and array logic symbols for a multiple input AND and a multiple input OR gate.



6.1 Programmable Array Logic (PAL)

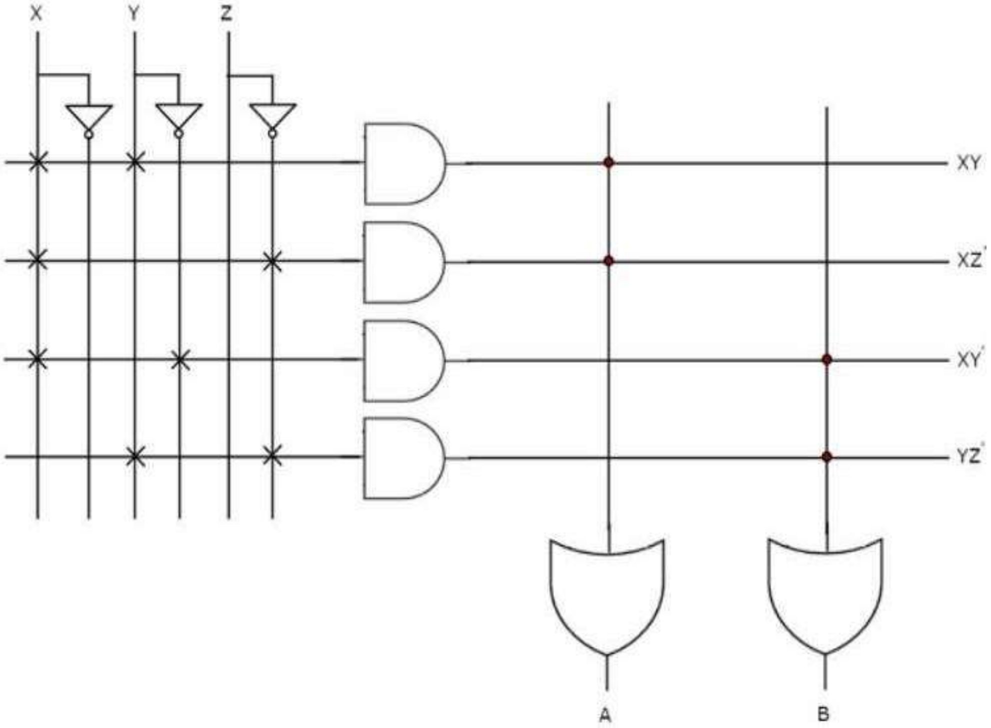
The programmable array logic (PAL) is a programmable logic device with a fixed OR array and a programmable AND array. Because only the AND gates are programmable the PAL is easier to program, but is not as flexible as the PLA .

The programmable array logic (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, but is not as flexible as the PLA.



Example 1: Implement the following Boolean functions using PAL.

$A = X Y + X Z'$ $\qquad$ $B = X Y' + Y Z'$



Here, we require four programmable AND gates & two fixed OR gates for producing those two functions.

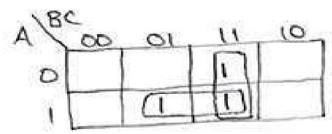
Example 2: Implement the following Boolean expression using PAL, F1= ∑ m (3,5,7) and F2 = ∑ m (4,5,7).

Since, F1= ∑ m (3,5,7) and F2 = ∑ m (4,5,7). Truth table for Boolean functions F1 and F2 can be drawn as:

Inputs			Outputs	
A	**B**	**C**	**F1**	**F2**
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	1	0
1	0	0	0	1
1	0	1	1	1
1	1	0	0	0
1	1	1	1	1

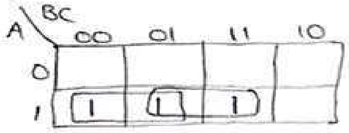
Now, for these Boolean functions, using the K-Map we can find the simplified Boolean expressions as:

for F_1



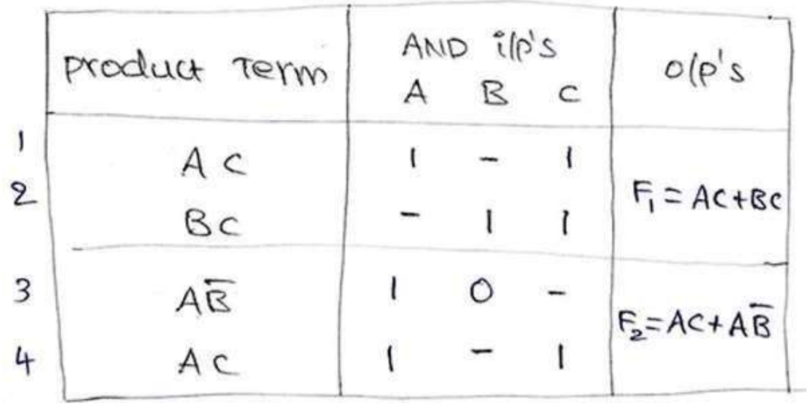
$\therefore \; F_1 = AC + BC$

for F_2



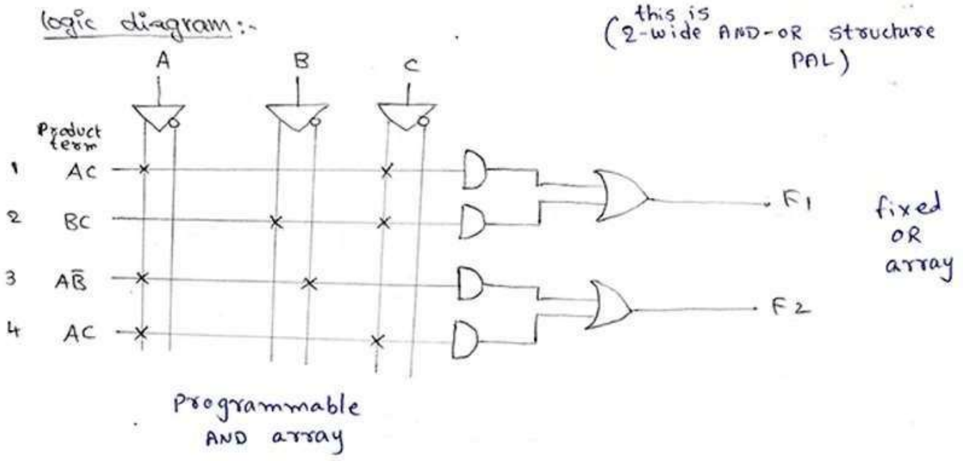
$\therefore \; F_2 = A\bar{B} + AC$

A PAL program table can be also drawn representing the terms in the Boolean expression as:



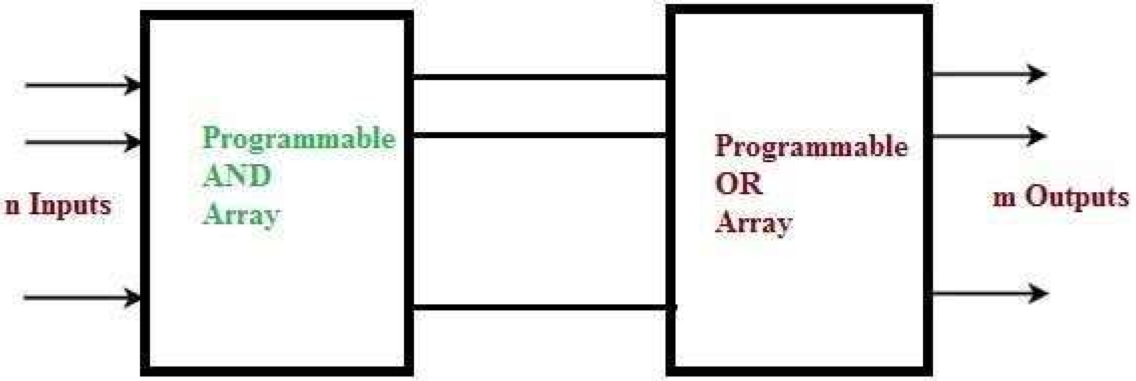
	Product term	AND i/p's			o/p's
		A	B	C	
1	AC	1	–	1	$F_1 = AC+BC$
2	BC	–	1	1	
3	$A\bar{B}$	1	0	–	$F_2 = AC+A\bar{B}$
4	AC	1	–	1	

The logic diagram of the combinational circuit implemented using PAL can be drawn as:



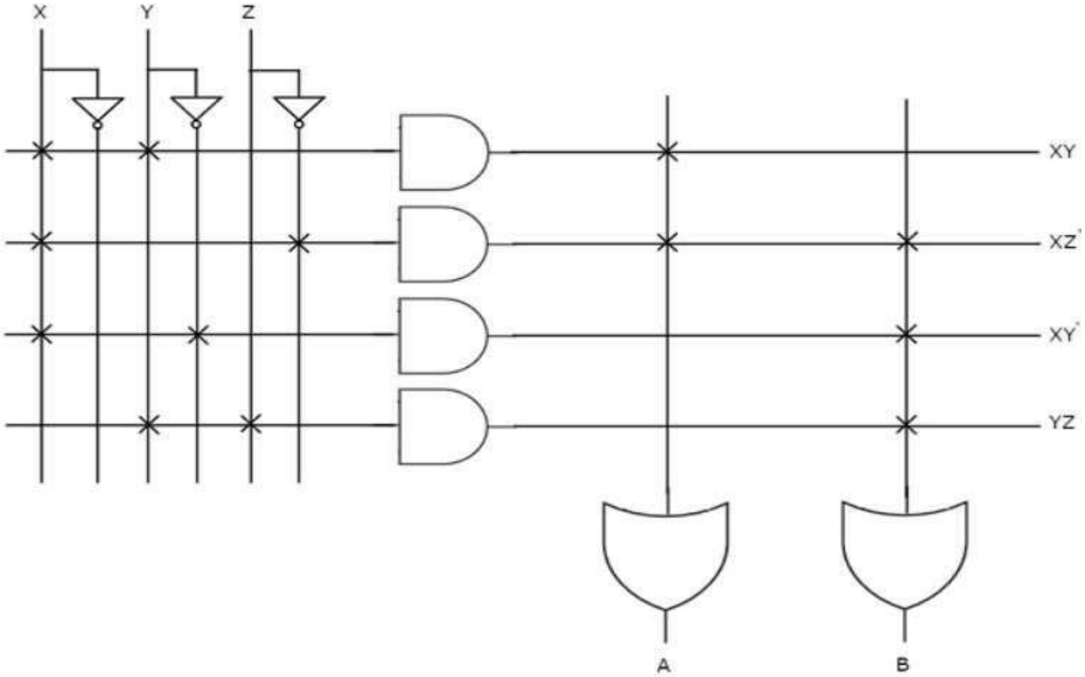
6.2 Programmable Logic Array (PLA):

The PLA is similar to PROM in concept except that PLA does not provide full decoding of the variable 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. A schematic diagram of the basic configuration of PLAs can be drawn as:



Example 1: Implement the following Boolean functions using PLA.

A = X Y + X Z′ B=XY′+YZ+XZ′

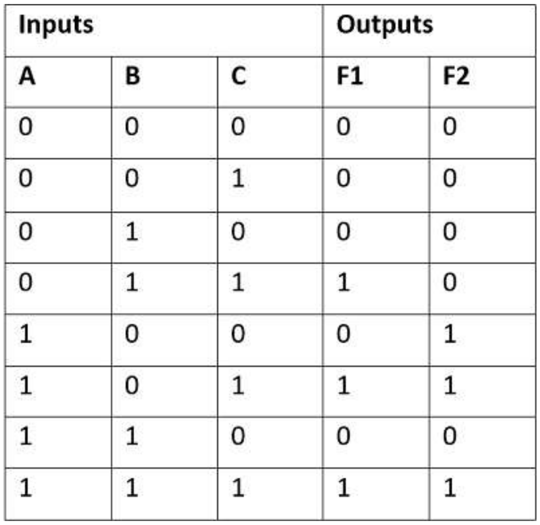


One product term, Z′X is common in each function. So, we require four programmable AND gates & two programmable OR gates for producing those two functions.

Example 2: A combinational circuit is defined by the function F1= ∑ m (3,5,7), F2 = ∑ m (4,5,7). Implement the circuit using a PLA which consists of 3 inputs (A, B and C), 3 product terms and two outputs.

Solution:

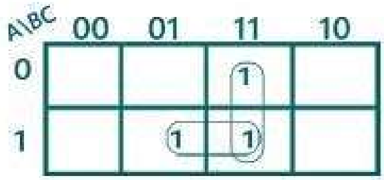
Since, F1= ∑ m (3,5,7) and F2 = ∑ m (4,5,7). Truth table for Boolean functions F1 and F2 can be drawn as:



Inputs			Outputs	
A	B	C	F1	F2
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	1	0
1	0	0	0	1
1	0	1	1	1
1	1	0	0	0
1	1	1	1	1

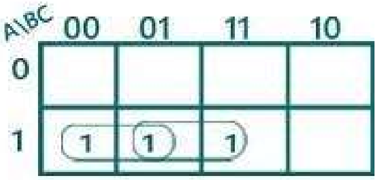
Now, for these Boolean functions, using the K-Map we can find the simplified Boolean expressions as:

for F_1



∴ $F_1 = AC + BC$

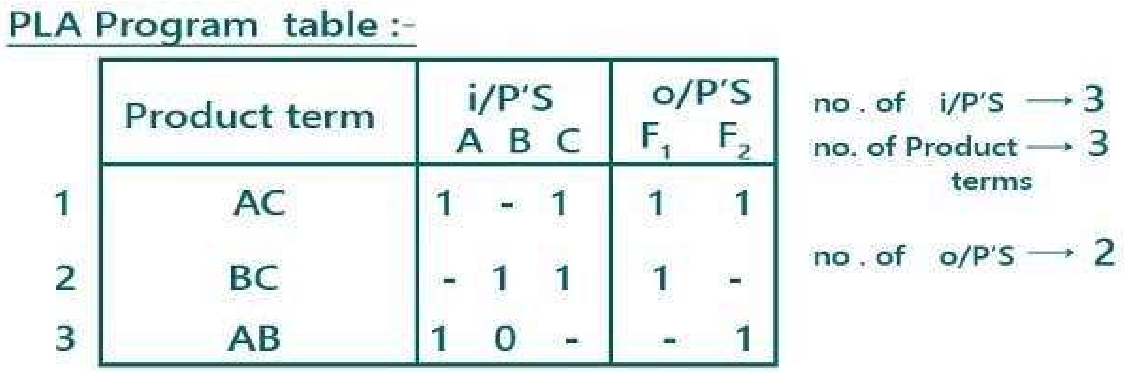
for F_2



∴ $F_2 = A\bar{B} + AC$

A PLA program table can be also drawn representing the terms in the Boolean expression as:

PLA Program table :-

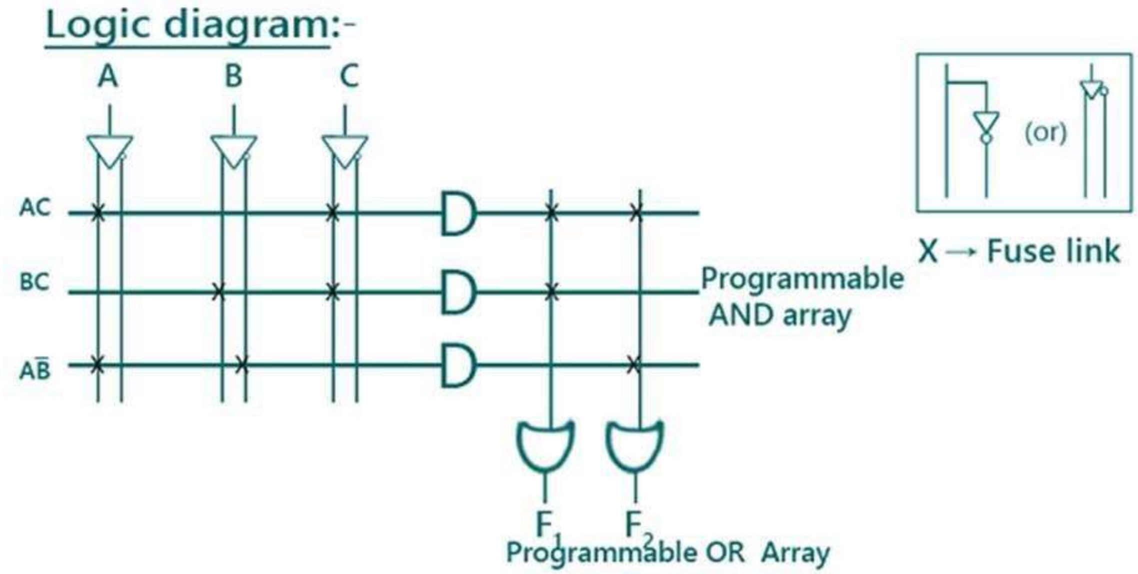


	Product term	i/P'S A B C	o/P'S F_1 F_2
1	AC	1 - 1	1 1
2	BC	- 1 1	1 -
3	AB	1 0 -	- 1

no . of i/P'S ⟶ 3
no. of Product terms ⟶ 3

no . of o/P'S ⟶ 2

The logic diagram of the combinational circuit implemented using PLA can be drawn as:



Applications of PLA and PAL:

PLA Applications:

- Versatile implementation of complex combinational logic functions.
- Commonly used in microprocessor design and arithmetic operations.
- Suitable for the implementation of control units in digital systems.

PAL Applications:

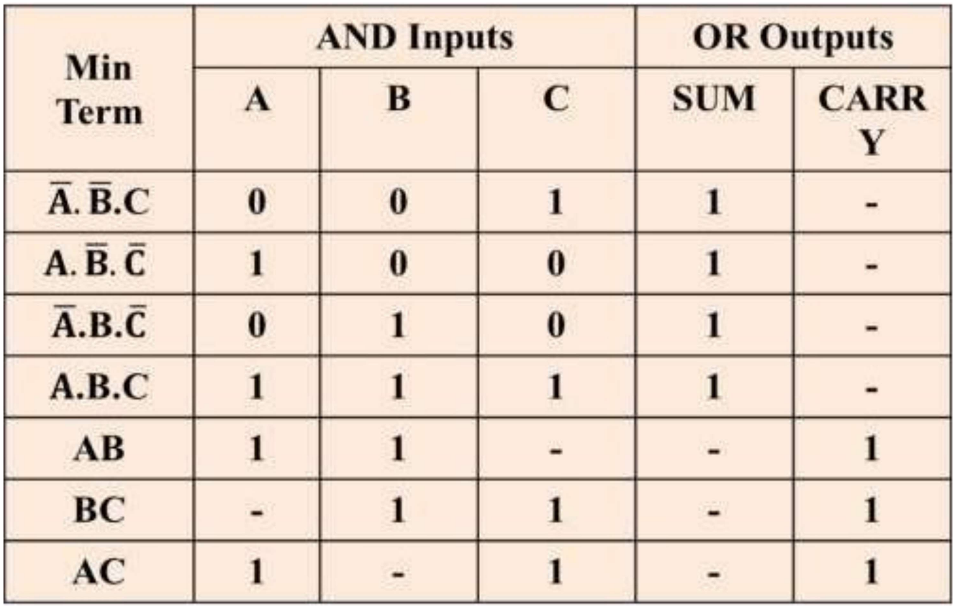
- Applied in scenarios requiring simpler combinational logic functions.
- Commonly used in memory address decoding.
- Cost-effective solution for applications with less complex logic requirements.
- Suitable for the implementation of straightforward control circuits.

7. **Activities/ Case studies/ Important facts:**

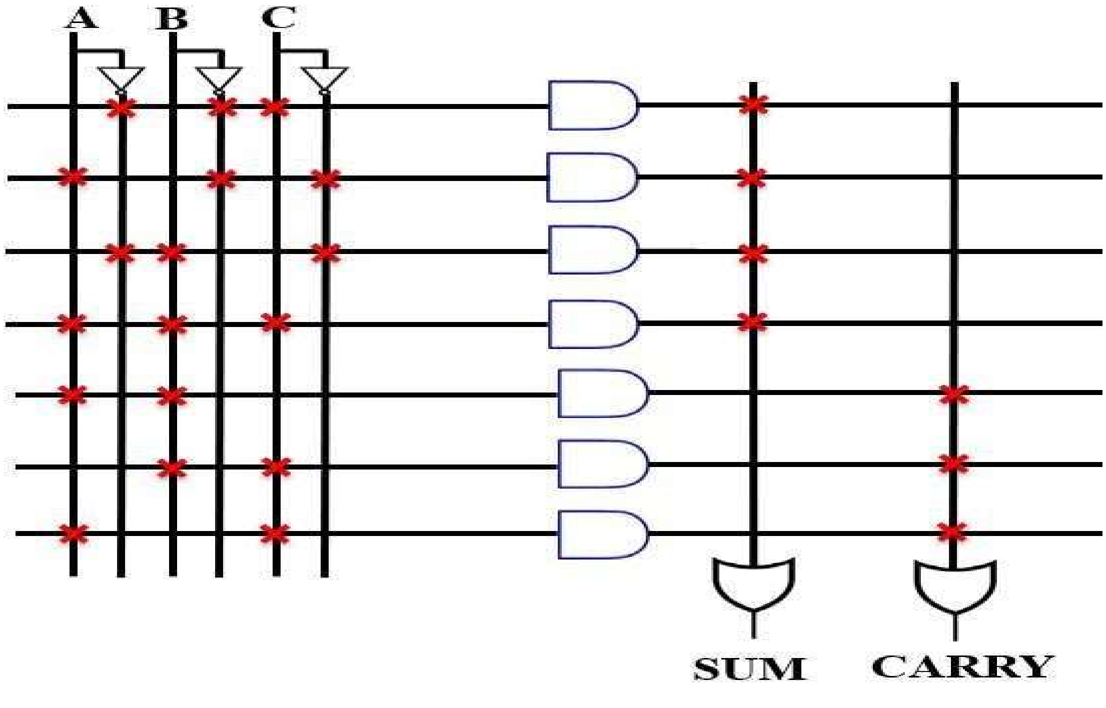
Implement Full Adder by using PLA:

SUM = $\bar{A}.\bar{B}.C + A.\bar{B}.\bar{C} + \bar{A}.B.\bar{C} + A.B.C$

Carry = AB + BC + AC



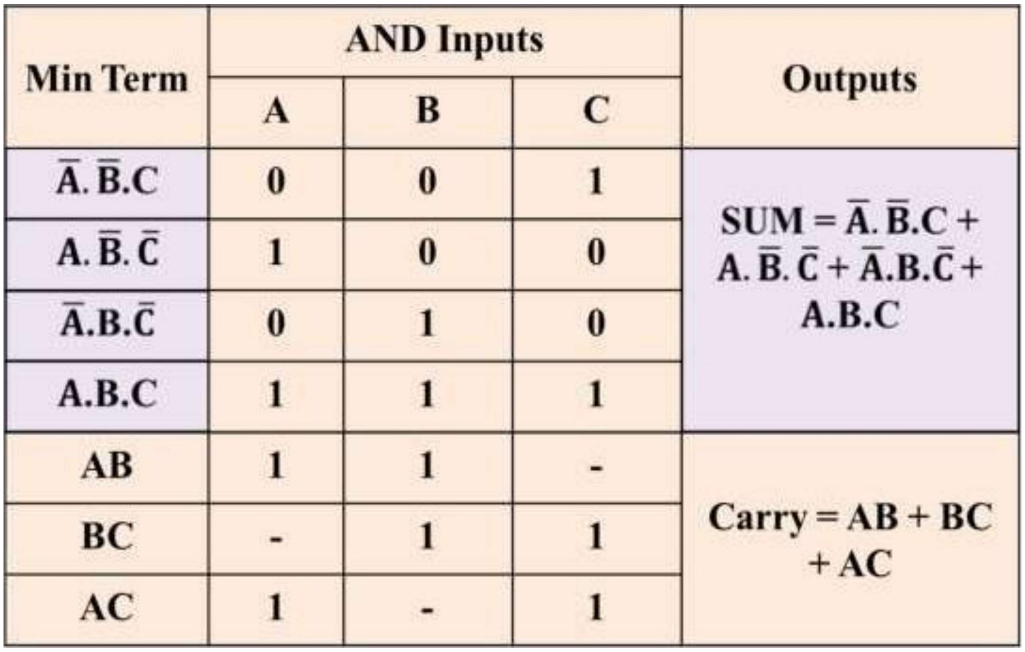
Min Term	AND Inputs			OR Outputs	
	A	B	C	SUM	CARRY
$\overline{A}.\overline{B}.C$	0	0	1	1	-
$A.\overline{B}.\overline{C}$	1	0	0	1	-
$\overline{A}.B.\overline{C}$	0	1	0	1	-
A.B.C	1	1	1	1	-
AB	1	1	-	-	1
BC	-	1	1	-	1
AC	1	-	1	-	1



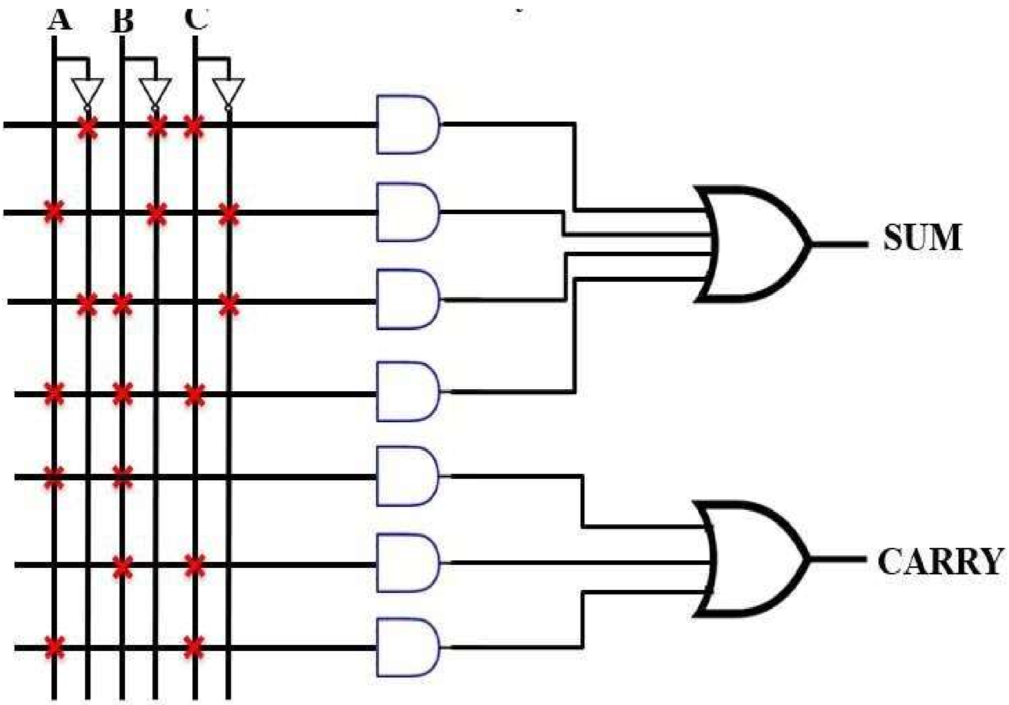
Implement Full Adder by using PAL:

SUM = $\overline{A}.\overline{B}.C + A.\overline{B}.\overline{C} + \overline{A}.B.\overline{C} + A.B.C$

Carry = AB + BC + AC



Min Term	AND Inputs			Outputs
	A	B	C	
$\bar{A}.\bar{B}.C$	0	0	1	SUM = $\bar{A}.\bar{B}.C + A.\bar{B}.\bar{C} + \bar{A}.B.\bar{C} + A.B.C$
$A.\bar{B}.\bar{C}$	1	0	0	
$\bar{A}.B.\bar{C}$	0	1	0	
A.B.C	1	1	1	
AB	1	1	-	Carry = AB + BC + AC
BC	-	1	1	
AC	1	-	1	



8. SAQ's – Self Assessment Questions:

1. PLA contains ____________

a) AND and OR arrays

b) NAND and OR arrays

c) NOT and AND arrays

d) NOR and OR arrays

2. A PLA is similar to a ROM in concept except that ____________

a) It hasn't capability to read only

b) It hasn't capability to read or write operation

c) It doesn't provide full decoding to the variables

d) It hasn't capability to write only

3. The difference between a PAL & a PLA is ____________

a) PALs and PLAs are the same thing

b) The PLA has a programmable OR plane and a programmable AND plane, while the PAL only has a programmable AND plane

c) The PAL has a programmable OR plane and a programmable AND plane, while the PLA only has a programmable AND plane

d) The PAL has more possible product terms than the PLA

4. If a PAL has been programmed once ____________

a) Its logic capacity is lost

b) Its outputs are only active HIGH

c) Its outputs are only active LOW

d) It cannot be reprogrammed

9. Summary:

Gates perform fixed logic operations, MUXes select data and can perform basic logic with proper configuration, while PALs and PLAs offer programmable arrays enabling versatile logic implementation, making them suitable for more complex logic functions with greater flexibility

PAL (Programmable Array Logic) and PLA (Programmable Logic Array) are integral components in digital design, offering programmable logic functions that allow for versatile circuit design and implementation. PALs are simpler, efficient for smaller designs, and use resources effectively.

PLAs offer more flexibility for complex logic but may require more resources.

Both PAL (Programmable Array Logic) and PLA (Programmable Logic Array) can be programmed to implement various types of logic circuits. Their configurability allows designers to create custom logic functions by programming the array of gates to perform specific operations.

10. Terminal Questions:

- Sketch a basic block diagram for a Programmable Array Logic (PAL) device.
- What is the schematic representation of a Programmable Logic Array (PLA)?
- Design the following Boolean functions using PAL.
  A (X,Y,Z) = Sum of Even Numbers (include Zero also) and B (X,Y,Z) = Sum of Odd Numbers
- Design the circuit with a PLA having three inputs and two outputs.
  i) F1 (A, B, C) = ∑m (3, 5, 6, 7)　　ii) F2 (A, B, C) = ∑(0, 2, 4, 7)
- Implement a full adder output using Programmable Logic Array.
- Implement the following Boolean functions using PAL.
  A = X Y’ + X’ Z　　B = X Y′ + Y Z′

11. Glossary:

PLAs (Programmable Logic Arrays) and PALs (Programmable Array Logic) devices are foundational to the development of customizable digital circuits, preceding the complexity and versatility of CPLDs and FPGAs. A PLA is characterized by its two programmable arrays—AND and OR arrays—allowing for the implementation of a wide range of combinational logic functions by configuring the connections between these arrays to realize any logical formula. On the other hand, PALs simplify this approach by having a programmable AND array followed by a fixed OR array, streamlining the design process but offering less flexibility compared to PLAs. Both types of devices utilize fuse-based technology or other forms of programmable interconnects to set their logic functions, and they are instrumental in realizing specific logic designs before the advent of more complex programmable logic devices. While they might not offer the same level of reconfigurability or density as their more advanced successors, PLAs and PALs continue to serve educational and niche roles in understanding and applying fundamental digital logic design principles.

12. References books:

- Computer System Architecture by M. Moris Mano
- Fundamentals of Digital Logic with Verilog HDL by Stephen Brown and ZvonkoVranesic

13. Sites and Web links:

1. https://www.tutorialspoint.com/digital_circuits/digital_circuits_programmable_logic_devices.htm
2. https://www.geeksforgeeks.org/programming-array-logic/
3. https://www.electrically4u.com/programmable-array-logic/