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 3: SOP/POS representation and optimization techniques

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):

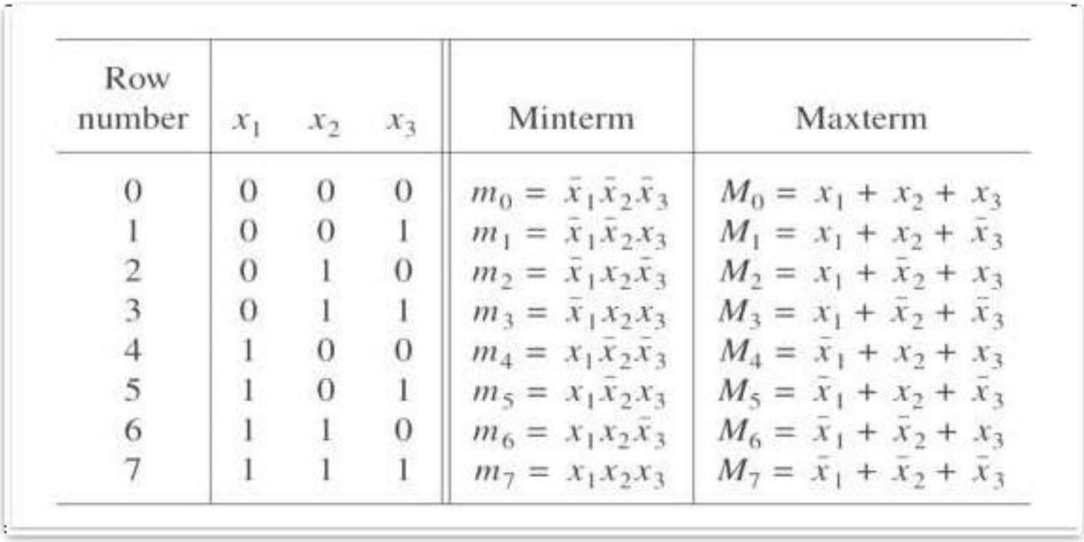
The module covers essential topics in digital electronics, starting with Boolean algebra and progressing to the representation and optimization techniques of digital logic using SOP/POS forms. Students will delve into the design of key components such as adders, subtractors, multiplexers, de-multiplexers, 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:

A Boolean expression or a function is an expression which consists of binary variables joined by the Boolean connectives AND and OR along with NOT operation.

Minterm: A minterm is a standard product which consists of all variables in either complemented or un-complemented form for which the output is 1.

Maxterm: A maxterm is a standard sum which consists of all variables in either complemented or un-complemented form for which the output is 0.



Row number	x_1	x_2	x_3	Minterm	Maxterm
0	0	0	0	$m_0 = \bar{x}_1\bar{x}_2\bar{x}_3$	$M_0 = x_1 + x_2 + x_3$
1	0	0	1	$m_1 = \bar{x}_1\bar{x}_2x_3$	$M_1 = x_1 + x_2 + \bar{x}_3$
2	0	1	0	$m_2 = \bar{x}_1x_2\bar{x}_3$	$M_2 = x_1 + \bar{x}_2 + x_3$
3	0	1	1	$m_3 = \bar{x}_1x_2x_3$	$M_3 = x_1 + \bar{x}_2 + \bar{x}_3$
4	1	0	0	$m_4 = x_1\bar{x}_2\bar{x}_3$	$M_4 = \bar{x}_1 + x_2 + x_3$
5	1	0	1	$m_5 = x_1\bar{x}_2x_3$	$M_5 = \bar{x}_1 + x_2 + \bar{x}_3$
6	1	1	0	$m_6 = x_1x_2\bar{x}_3$	$M_6 = \bar{x}_1 + \bar{x}_2 + x_3$
7	1	1	1	$m_7 = x_1x_2x_3$	$M_7 = \bar{x}_1 + \bar{x}_2 + \bar{x}_3$

Sum of Products (SOP): The sum of products is a Boolean expression containing AND terms, called Product terms, of one or more literals each; the sum denotes the ORing of these terms.

$$F = A'B + BC + A'C$$

Product of Sums (POS): It is a Boolean expression containing OR terms called Sum terms and the product denotes the ANDing of these terms.

$$F = (A + B)(B + C')(A + C')$$

Canonical form –

All the variables must be available in each term either in true or complemented form

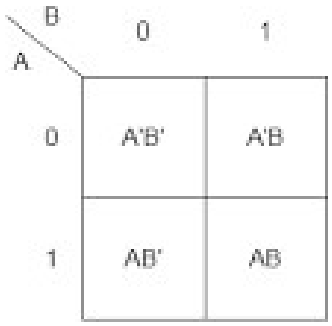
$$F = A'B'C + ABC' + ABC \quad F = (A + B + C)(A + B' + C')$$

Simplification using K-maps:

A Karnaugh map is a graphical representation of the logic system. It can be drawn directly from either rminterm (sum-of-products) or maxterm (product-of-sums) Boolean expressions. Drawing a Karnaugh map from the truth table involves an additional step of writing the minterm or maxterm expression depending upon whether it is desired to have a minimized sum-of-products or a minimized product of sums expression.

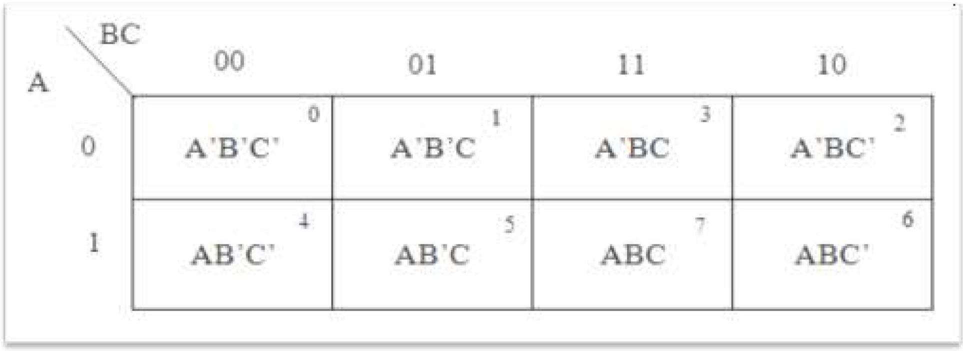
An n-variable Karnaugh map has 2^nsquares, and each possible input is allotted a square. In the case of a minterm Karnaugh map, ‘1’ is placed in all those squares for which the output is ‘1’, and ‘0’ is placed in all those squares for which the output is ‘0’. 0s are omitted for simplicity. An ‘X’ is placed in squares corresponding to ‘don’t care’ conditions.

a) **Two Variable K-map for SOP expressions:**



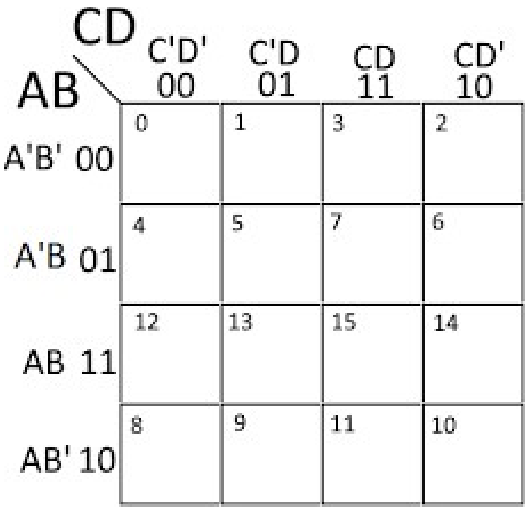
A \ B	0	1
0	A'B'	A'B
1	AB'	AB

b) **Three Variable K-map for SOP expressions:**



A \ BC	00	01	11	10
0	A'B'C' 0	A'B'C 1	A'BC 3	A'BC' 2
1	AB'C' 4	AB'C 5	ABC 7	ABC' 6

c) **Four Variable K-map for SOP expressions:**



AB \ CD	C'D' 00	C'D 01	CD 11	CD' 10
A'B' 00	0	1	3	2
A'B 01	4	5	7	6
AB 11	12	13	15	14
AB' 10	8	9	11	10

Simplification Algorithm:

Simplification of logical functions using K-maps is based on the principle of combining terms in adjacent cells. Two cells are said to be adjacent if they differ in only one variable.

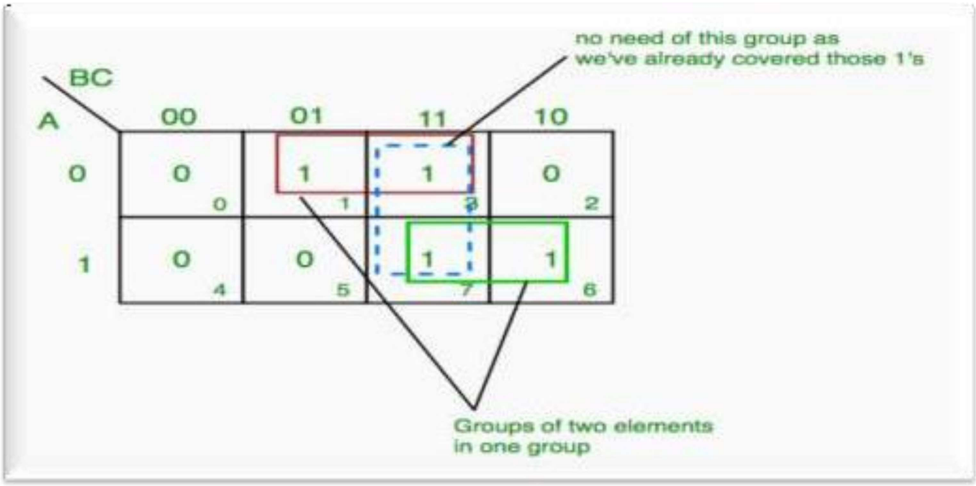
1. Identify the ones which cannot be combined with any other ones and encircle them. These are called essential prime implicants.
2. Identify the ones that can be combined in groups of two in only one way. Encircle them.
3. Identify the ones that can be combined with three other ones, to make a group of four adjacent ones, in only one way. Encircle such group of ones.
4. Identify the ones that can be combined with seven other ones, to make a group of eight adjacent ones, in only one way. Encircle them.

5. After identifying the essential groups of 2, 4, and 8 ones, if there still remain some ones which have not been encircled, then these are to be combined with each other or with other already encircled ones.

7. Examples:

1. Given Z = ∑ m (1,3,6,7) Minimize using 3 variable K-Maps

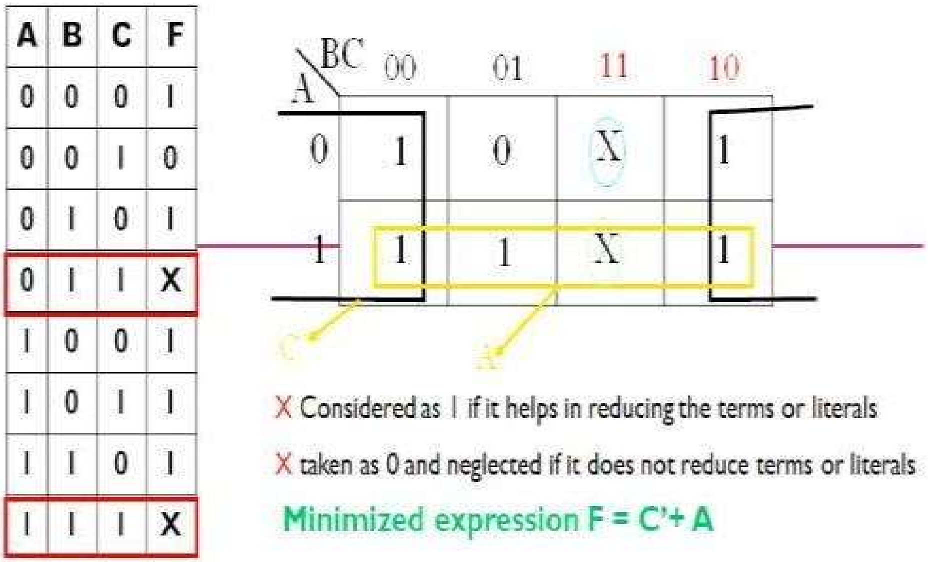
   Sol:



Minimized expression Z = A'C+AB

2. Given F (A,B,C) = ∑ (0,2,4,5,6) + d (3,7) Minimize using 3 variable K-Maps

   Sol:



A	B	C	F
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	X
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	X

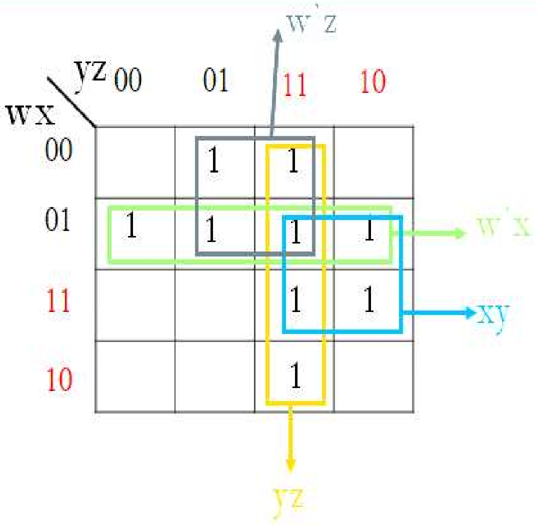
X Considered as 1 if it helps in reducing the terms or literals

X taken as 0 and neglected if it does not reduce terms or literals

Minimized expression F = C'+ A

3. Given F(w,x,y,z) = ∑(1,3,4,5,6,7,11,14,15) Minimize using 4 variable K-Maps

Sol:

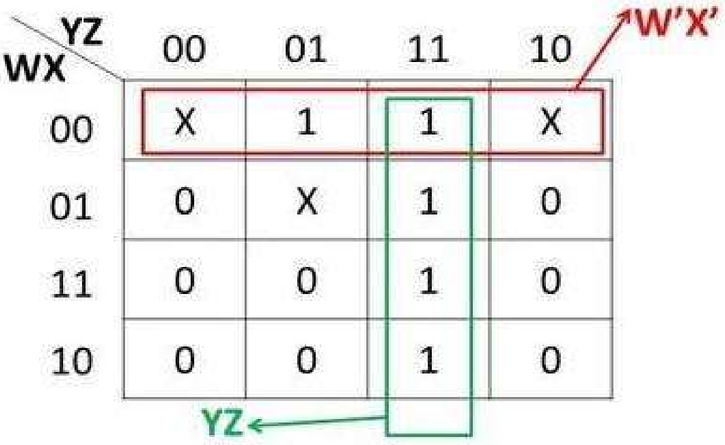


wx \ yz	00	01	11	10
00		1	1	
01	1	1	1	1
11			1	1
10			1	

Simplified expression is
F = w'x + yz + xy + w'z

4. Given F (w,x,y,z) = ∑ (1,3, 7,11,15) + d (0,2,5) Minimize using 4 variable K-Maps

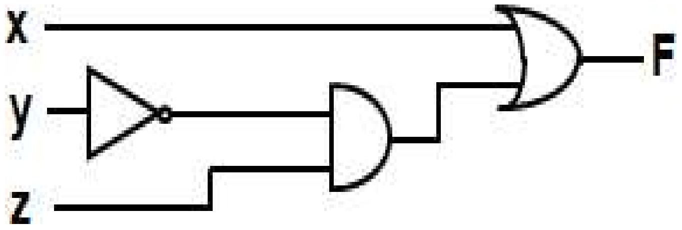
Sol:



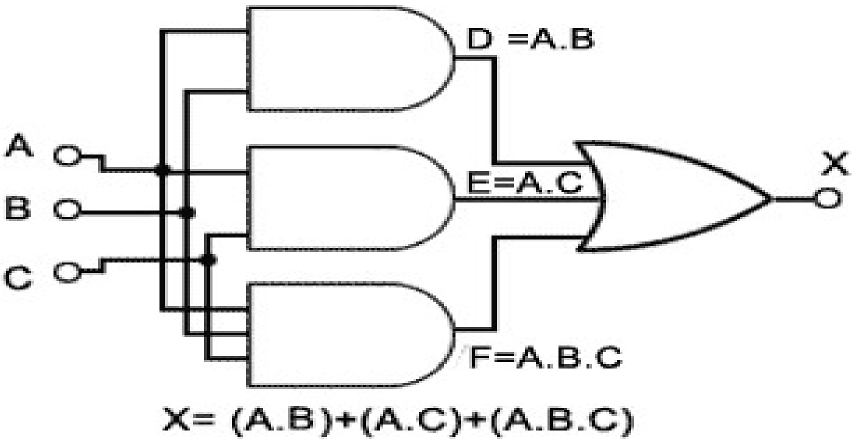
WX \ YZ	00	01	11	10
00	X	1	1	X
01	0	X	1	0
11	0	0	1	0
10	0	0	1	0

F= W'X' + YZ

5. Given F = x + y ’ . z , Realize the expression using logic gates.



6. Given X = (A.B) + (A.C) + (A.B.C) , Realize the expression using logic gates.



8. SAQ’s – Self Assessment Questions:

1. **What does SOP stand for in Digital Logic?**

   a) Sum-of-Products

   b) Systematic Output Procedure

   c) Simplified Output Protocol

   d) Sum-of-Processes

2. **Which notation is used to represent minterms in SOP expressions?**

   a) A + B

   b) A * B

   c) Σ(A, B)

   d) ∏(A, B)

3. What is the primary goal of grouping in Karnaugh Maps?

a) To make the map look organized

b) To create larger groups

c) To identify adjacent cells with '1’

d) To separate '1' and '0' values

4. How many cells are in a 2-variable Karnaugh Map?

a) 4

b) 8

c) 2

d) 16

9. Terminal Questions:

- Develop a truth table that represents the Boolean equation. F = A’B’C + AB’C’ + ABC’ + ABC = ∑ m (1,4,6,7).
- Optimize the 4 variable function F 〔A,B,C,D〕 = ∑m 〔0,1,4,5,6,10,13〕 + d 〔2,3〕 using K-Maps.
- Represent the given expression in canonical POS form Y = (A + B)(B + C)(A + C)
- Optimize the equation F (A, B, C) = AB’C + A’B’C + A’BC + A’B’C’ + AB’C’ using K-Maps and realize the resultant expression using logic gates.
- Represent the given expression in canonical SOP form Y = AC + AB + BC.
- Optimize the 4 variable function F 〔A,B,C,D〕 = ∑m 〔0,1,4,5,6,10,13〕 + d 〔2,3〕 using K-Maps and realize the minimized expression using logic gates.
- Optimize the given function using K-map F (W, X, Y, Z) = ∑m (1, 3, 4, 5, 6, 7, 11, 14, 15) and implement using logic gates.
- Optimize the 4 variable function F 〔W,X,Y,Z〕 = ∑m 〔1,3,7,11,15〕 + d 〔0,2,5〕 using K-Maps and realize the minimized expression using logic gates.
- Optimize the equation F (A, B, C) = AB’C + A’B’C + A’BC + A’B’C’ + AB’C’ using K-Maps and realize the same using logic gates.
- Create separate tables listing the Min and Max terms for 2 variables (A, B) and 3 variables (A, B, C).

10. References books:

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

11. Sites and Web links:

- https://www.geeksforgeeks.org/introduction-of-k-map-karnaugh-map/
- https://www.gatevidyalay.com/tag/k-map-sop-and-pos/#google_vignette