UNIT - I

Minimization Technique

* Number System :-

A system that is used for representing numbers is called the number system.

- There are different types of number system
  1) Binary Number system
  2) octal Number system
  3) Decimal Number system
  4) Hexadecimal Number system.

1) Binary Number system: (Base 2)

A Binary number system is used in the digital computers., it carries only two digits, either 0 or 1.

There are two types of electronic pulses present in a binary number system.

- The first one is the absence of an electronic pulse representing '0' &
- second one is the presence of an electronic pulse representing '1'.

* - Each digit is known as a bit.
* - 4 bit collection (1101) is nibble,
* - 8 bits collection (11001010) is known as a byte

- It uses the base 2

4 bit → nibble
8 bits → 1 byte
1 digit - bit

* octal Number system :- (Base 8)

- A Number system which has base 8 is called an octal Number system.
- The range of value from 0, 1, 2, 3, --- 7.
(0, 1, 2, 3, 4, 5, 6, 7)

eg: $(124)_8$
$= (1 \times 8^2) + (2 \times 8^1) + (4 \times 8^0)$
$= 64 + 16 + 4$
$= 84$
$(124)_8 = (84)_{10}$

- An octal number can be converted into an equivalent in decimal.
- each digits represent the power of 8.

* Decimal Number system :- (Base 10)

- The system of numbers which has base or radix 10.
(i.e it uses that 10 symbols to represent numbers of the system is called Decimal number system.
symbols (0, 1, 2, 3, 4, 5, 6, 7, 8, 9)

* Hexadecimal Number system :- (Base 16)

The number system with base or radix 16 is called as Hexadecimal Number system.

- Thus, hexadecimal number system uses 16 symbol to represent numbers.
(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F)
where... A = 10, B = 11, C = 12, D = 13, E = 14, F = 15 [why not 10, 11, 12, ... because on keyboard only 0...9 digits are present.]
- The hexadecimal number system is used in microprocessor & microcontrollers

* Conversion Table :-

Decimal Number	Binary Number	Octal Number	Hexadecimal Number
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Date
Page

* LOGIC GATE :-

- A Logic gate is a simple switching circuit that determines whether an i/p pulse can pass through to the o/p in digital circuits.

* Types of Logic Gates :-

1) AND
2) OR
3) NOT
4) NOR
5) NAND
6) XOR
7) XNOR

* Basic Logic Gates :

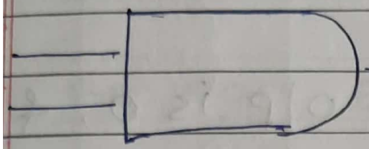
1) AND Gate :

- An AND Gate has a single o/p & two or more i/p.

1) when all the i/p are 1, then the o/p of this gate is 1

2) Boolean Logic is $Y = A \cdot B$

Symbol



$Y = AB$

Truth Table

i/p		O/P
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

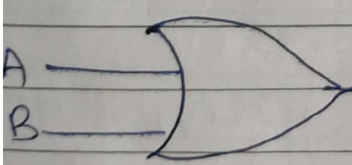
$Y = A \cdot B$

OR Gate :

- Two or more i/p & one o/p can be used in OR Gate.

i) The logic of this gate is that if at least one of the inputs is 1, the o/p will be 1.

Symbol



$Y = A+B$

Truth Table

i/p		O/P
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

$Y = A+B$

i.e in the OR Gate, the o/p is high when any of the i/p is high.

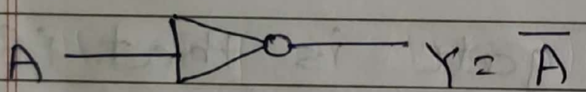
3) NOT Gate :

- The NOT gate is a basic one-input, one-output gate.

1) when the i/p is 1, the o/p is 0, & vice versa.
- A NOT gate is sometimes called an inverter because of its feature.

2) If there is only one i/p A, the o/p may be calculated using the Boolean equation $Y = A'$

symbol



Truth Table

i/p	O/P
A	NOT A
0	1
1	0

(i.e - Reverse the i/p signal)

* Universal Logic Gates :

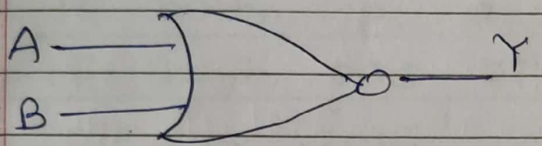
1) NOR Gate :

- A NOR Gate also known as a "NOT-OR" gate.
- consist of an OR gate followed by NOT gate

1) This gate o/p is 1 only when all of its i/p are 0. , Alternatively, when all of the i/p are low, the o/p is high.

2) Boolean Expression of NOR gate
$Y = \overline{(A+B)}$

Symbol



$Y = \overline{A+B}$

Truth Table

i/p		o/p
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

2) NAND Gate:

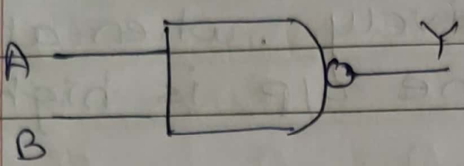
- A NAND Gate also known as 'NOT-AND' Gate.
this gate (NOT) gate followed by AND gate.

1) This gate's o/p is 0 only if none of the i/p is 0, Alternatively, when all of the i/p are not high and at least one is low, o/p is high.

2) If there are two i/p A and B, the Boolean expression
$Y = \overline{(A \cdot B)}$

Classmate
Date
Page

symbol



Truth Table

i/p		o/p
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

$Y = \overline{A \cdot B}$

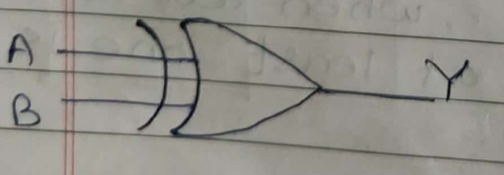
3) XOR Gate :-

The Exclusive-OR or 'Ex-OR' gate is a digital logic gate that accepts more than two i/p's but only o/p's one value.

1) If any of the i/p is 'High', the o/p of the XOR Gate is High.
if both i/p's are 'High' the o/p is 'Low'
if both i/p's are 'Low' the o/p is 'Low'

2) Boolean Expression: $Y = A \oplus B \; (\bar{A}B + A\bar{B})$

symbol



Truth Table

i/p		o/p
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

$Y = A \oplus B \; (\bar{A}B + A\bar{B})$

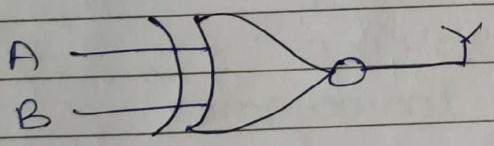
4) XNOR Gate :

- The Exclusive -NOR or 'EX-NOR' gate is a digital logic gate that accepts more than two i/p's but only one outputs.

1) if both i/p's are 'High', the o/p's of the XNOR gate is 'High'.
if both i/p's are 'Low', the o/p's is 'High'
if ~~both~~ one of the i/p's is 'Low', the o/p is 'Low'

2) Boolean Expression: $Y = \overline{A \oplus B} \; (AB + \bar{A}\bar{B})$

Symbol



Truth Table

i/p		o/p
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

classmate
Date
Page

* 1's and 2's complement of a Binary Number

* 1's complement :

1's complement of a binary number is another binary number obtained by toggling all bits in it.
i.e. transforming the 0 bit to 1 & the 1 bit to 0.

- the positive numbers remain unchanged. & the negative numbers are obtained by taking the 1's complement of positive counter parts.

For eg: +9 will be represented as
00001001 in 8-bit notation.
-9 will be represented as
11110110 → which is the 1's complement of 00001001

eg: 1's complement of "0111" is "1000"
"1100" is "0011"

* 2's complement :

2's complement of binary number is 1, added to the 1's complement of the binary number, the resulting number is known as the 2's complement of binary Number.

eg:

① 0101

1st complement : 1 0 1 0

adding 1 → + 1

$$\underbrace{1\ 0\ 1\ 1}_{\text{2's complement}}$$

② 2's complement of binary number 5 (0101) is binary Number 11 (1011)

0 1 0 1

1's + 1 0 1 0

~~1 0 +~~

adding 1 + 1

$$\underbrace{1\ 0\ 1\ 1}_{\text{2's complement}}$$

* System or circuit :-

- A system or circuit is defined as the physical device or group of devices or algorithm which performs the required operations on the signal applied as its input which can be either Analog or Digital.

- system or circuits can be classified into two types.
  1) Analog systems
  2) Digital systems

Digital systems :-

The system which process on the digital signal or, works upon the digital signal to produce another digital signal.

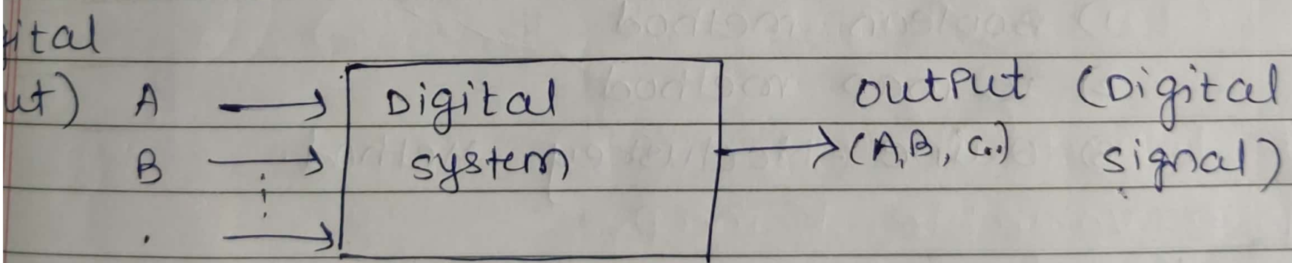


Fig: Digital system.

eg: 1) Registers
2) Flip- flop
3) counters
4) computer
5) Microprocessor
6) calculator etc.

Two voltage Levels:

HIGH logic - logic 1 level - (i.e True)
Low logic - logic 0 level (i.e False)

Note :- Digital circuit is also known as logic circuit, because they works on the logic.

- Digital system supports only binary data,
  High is represented logic 1 - Positive
  low is represented logic 0 - Negative

- Digital system is defined by Boolean expression or truth table or state diagram.

- To simplifying the Boolean expression three simplification Methods.
  1) Boolean method
  2) K-map method
  3) Quine-Mccluskey Method.

* There are six types of Boolean laws:

1) commutative law :-

commutative law states that changing the sequence of the variables does not have any effect on the output of a logic circuit.

i) $A \cdot B = B \cdot A$

ii) $A + B = B + A$

2) Associative Law :-

This law states that the order in which the logic operations are performed is irrelevant as their effect is the same.

i) $(A \cdot B) \cdot C = A \cdot (B \cdot C)$

ii) $(A+B)+C = A+(B+C)$

3) Distributive Law :-

Distributive law states the following condition.

$$A \cdot (B+C) = A \cdot B + A \cdot C$$

4) AND law :-

These law is used the AND operation. i.e they are called as AND laws.

i) $A \cdot 0 = 0$ iii) $A \cdot A = A$

ii) $A \cdot 1 = A$ iv) $A \cdot \overline{A} = 0$

5) OR Law :-

These law is used the OR operation. i.e they are called as OR Laws.

i) $A+0 = A$ iii) $A+A = A$

ii) $A+1 = 1$ iv) $A+\bar{A} = 1$

6) Inversion Law :-

These law uses NOT operation.

The inversion law stats that double inversion of a variable results in the original variable itself.

$$\overline{\overline{A}} = A$$

* Minimization of Boolean function using K-maps (upto 4 variables)

K-Map (Karnaugh - Map)

- K-map is a graphical technique to simplify Boolean expressions.

- K-map can take two forms S.O.P (Sum of Product terms) & P.O.S (Product of sum term.

- K-map can be used from two variable to six variable

- If there are 'n' variables, then K-map will contain 2^n cells.
  (2 variable $2^n = 2^2 = 4$ cells)
  (3 variable $2^n = 2^3 = 8$ cells)
  (4 variable $2^n = 2^4 = 16$ cells)

* K-map structure :-

2-variable K-map -

A \ B		$\bar{B}$ 0	B 1
$\bar{A}$	0	0	1
A	1	2	3

A \ B		$\bar{B}$ 0	B 1
$\bar{A}$	0	$\bar{A}\bar{B}$ 0	$\bar{A}B$ 1
A	1	$A\bar{B}$ 2	AB 3

* 3-variable K-map :-

stage change.
we can change only 1 bit
2 bit not possible so
we write 10 11
(only one variable can changed)

A \ BC	$\bar{B}\bar{C}$ 00	$\bar{B}C$ 01	BC 11	$B\bar{C}$ 10
$\bar{A}$ 0	0	1	3	2
A 1	4	5	7	6

A \ BC	$\bar{B}\bar{C}$ 00	$\bar{B}C$ 01	BC 11	$B\bar{C}$ 10
$\bar{A}$ 0	$\bar{A}\bar{B}\bar{C}$ 0	$\bar{A}\bar{B}C$ 1	$\bar{A}BC$ 3	$\bar{A}B\bar{C}$ 2
A 1	$A\bar{B}\bar{C}$ 4	$A\bar{B}C$ 5	ABC 7	$AB\bar{C}$ 6

Fig: K-map & its product team (SOP)

* 4-variable K-map :-

AB \ CD	$\bar{C}\bar{D}$ 00	$\bar{C}D$ 01	CD 11	$C\bar{D}$ 10
$\bar{A}\bar{B}$ 00	$\bar{A}\bar{B}\bar{C}\bar{D}$ 0	$\bar{A}\bar{B}\bar{C}D$ 1	$\bar{A}\bar{B}CD$ 3	$\bar{A}\bar{B}C\bar{D}$ 2
$\bar{A}B$ 01	$\bar{A}B\bar{C}\bar{D}$ 4	$\bar{A}B\bar{C}D$ 5	$\bar{A}BCD$ 7	$\bar{A}BC\bar{D}$ 6
AB 11	$AB\bar{C}\bar{D}$ 12	$AB\bar{C}D$ 13	$ABCD$ 15	$ABC\bar{D}$ 14
$A\bar{B}$ 10	$A\bar{B}\bar{C}\bar{D}$ 8	$A\bar{B}\bar{C}D$ 9	$A\bar{B}CD$ 11	$A\bar{B}C\bar{D}$ 10

Truth Table of 4-variable K-map:

Input variables				Output
A	B	C	D	Y
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

Rules for K-map simplification:-

1. Groups may not contain zero
2. either group 0's with 0's or 1's with 1's, but we cannot group 0's & 1's together.

3. We can group 2^n (i.e 1, 2, 4, 8, 16) or number of cells.

4. Group can be either horizontal or vertical. (we can not create groups of diagonal or any other shape).

5. Each group should be as large as possible. (opposite grouping is allowed)

6. opposite grouping & corner grouping are allowed.

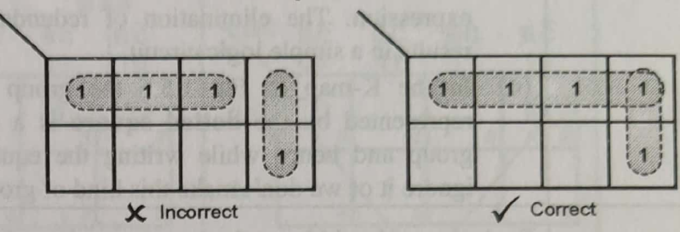
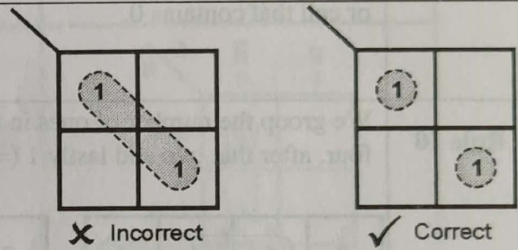
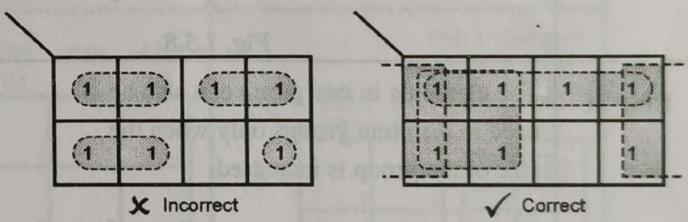
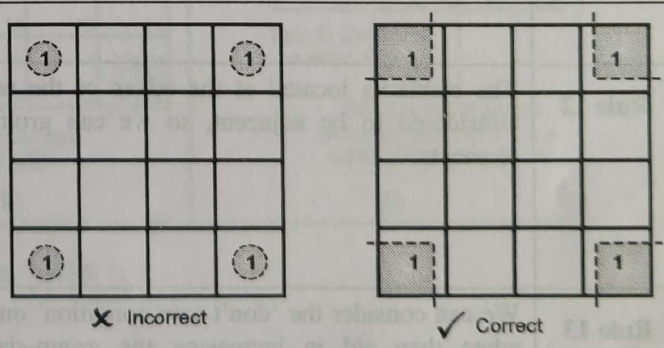
7. There should be as few groups as possible

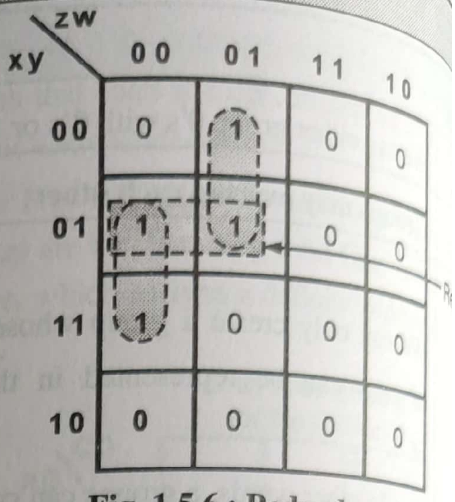
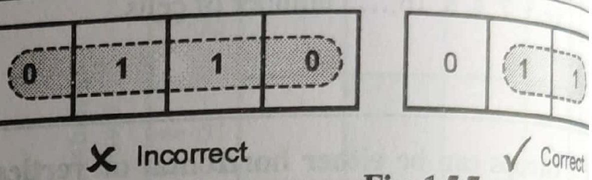
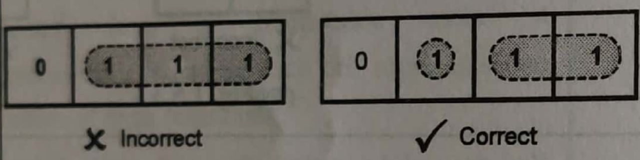
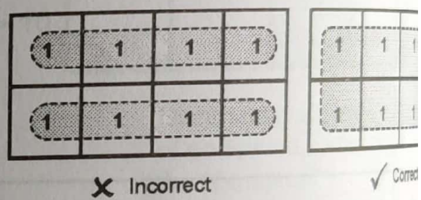
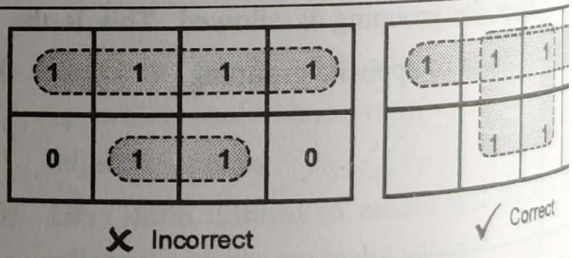
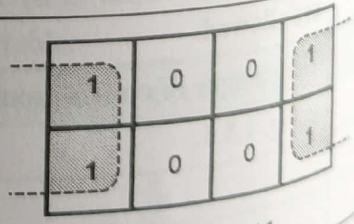
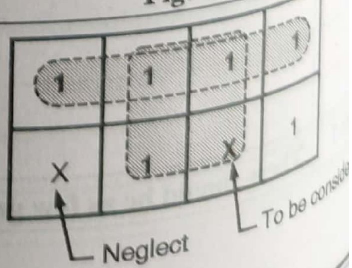
8. Redundant group.

9. Each group should have the largest number of 'ones'.
   A group can not contain an empty cell or cell that contains 0.

10. We group the number of ones in the decreasing order.

1.5.2 K-Map Simplification Rules (Most Important)

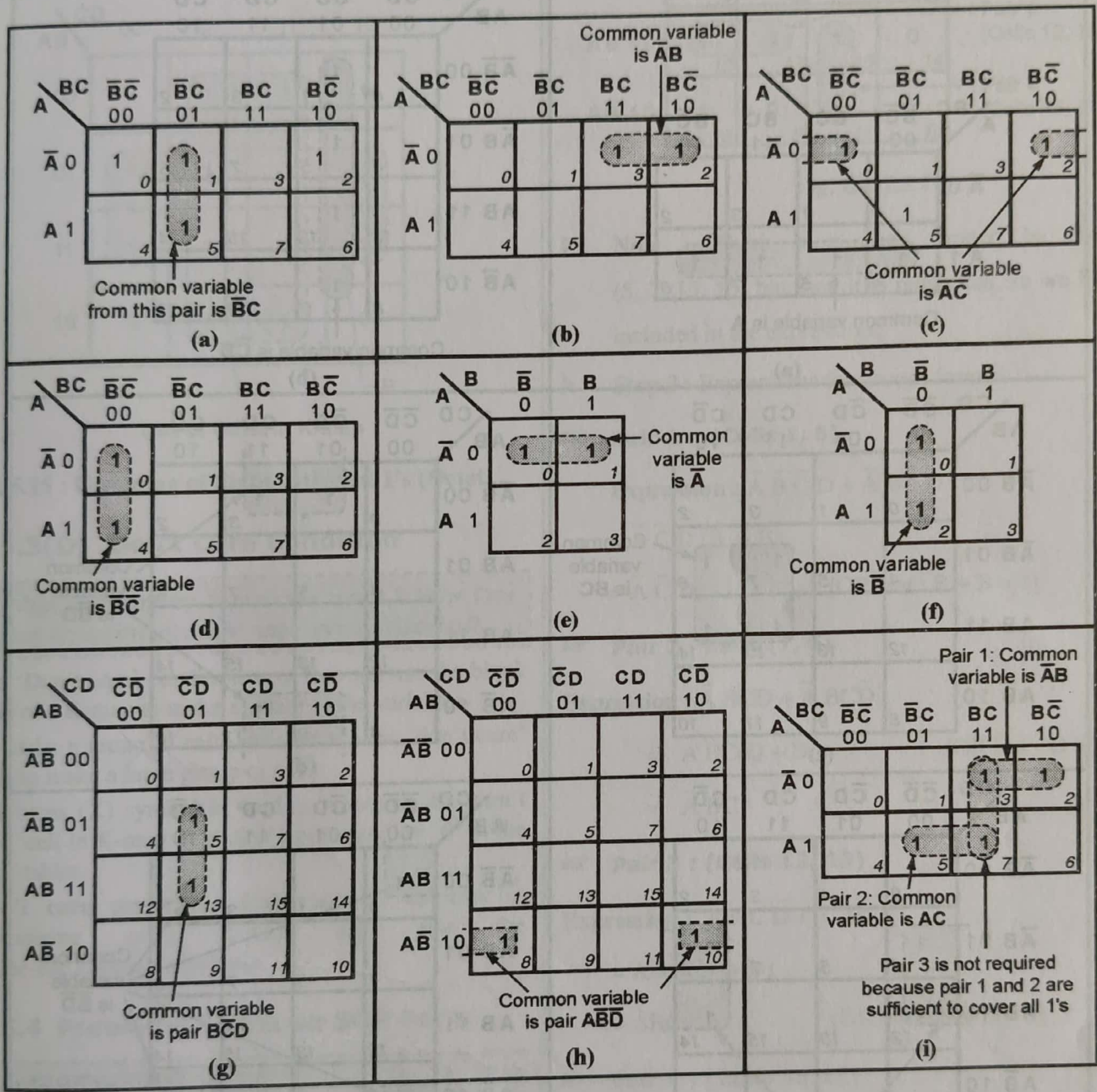
Rule No.	Explanation
Rule 1	We can either group 0's with 0's or 1's with 1's but we cannot group 0's and 1's together
Rule 2	**Groups may overlap each other.**
Rule 3	We can only create a group whose number of cells can be represented in the power of 2. In other words, a groups can contain 2^n; i.e. 1, 2 ,4, 8, 16,….. number of cells. **Example**  (1E8)Fig. 1.5.2
Rule 4	**Groups can be either horizontal or vertical.** We cannot create groups of diagonal or any other shape  (1E9)Fig. 1.5.3
Rule 5	**Each group should be as large as possible.** Opposite grouping is allowed. This is the example of opposite grouping.  (1E10)Fig. 1.5.4
Rule 6	**Opposite grouping and corner grouping are allowed.** The example of corner grouping is as shown in Fig. 1.5.5.  (1E11)Fig. 1.5.5
Rule 7	**There should be as few groups as possible.**

Rule No.	Explanation
Rule 8	**Redundant Group** UQ. Define a redundant group. (Q. 1(g), May 18, 2 Marks) (i) **A redundant group** is a group whose all 1's are overlapped by other groups. We eliminate these redundant groups to get the simplified Boolean expression. The elimination of redundant groups results in a simple logic circuit. (ii) In the K-map of Fig 1.5.6 the group which is represented by the **dotted square** is a redundant group and hence while writing the equations we ignore it or we don't make this kind of group.  **Fig. 1.5.6 : Redundant Group**
Rule 9	Each group should have the largest number of **'ones'**. A group cannot contain an empty cell or cell that contains 0.  **Fig. 1.5.7**
Rule 10	We group the number of ones in the decreasing order. First, we have to try to make the group of eight, four, after that two and lastly 1 (= 2^0).  **Fig. 1.5.8**  **Fig. 1.5.9**
Rule 11	The elements in one group can also be used in different groups only when the size of the group is increased.  **Fig. 1.5.10**
Rule 12	The elements located at the edges of the table are considered to be adjacent, so we can group these elements.  **Fig. 1.5.11**
Rule 13	We can consider the 'don't care condition' only when they aid in increasing the group-size. Otherwise 'don't care' elements are discarded.  **Fig. 1.5.12**

1.5.3 Method of Grouping

1.5.3 (A) Grouping of Two adjacent 1's (Pair)

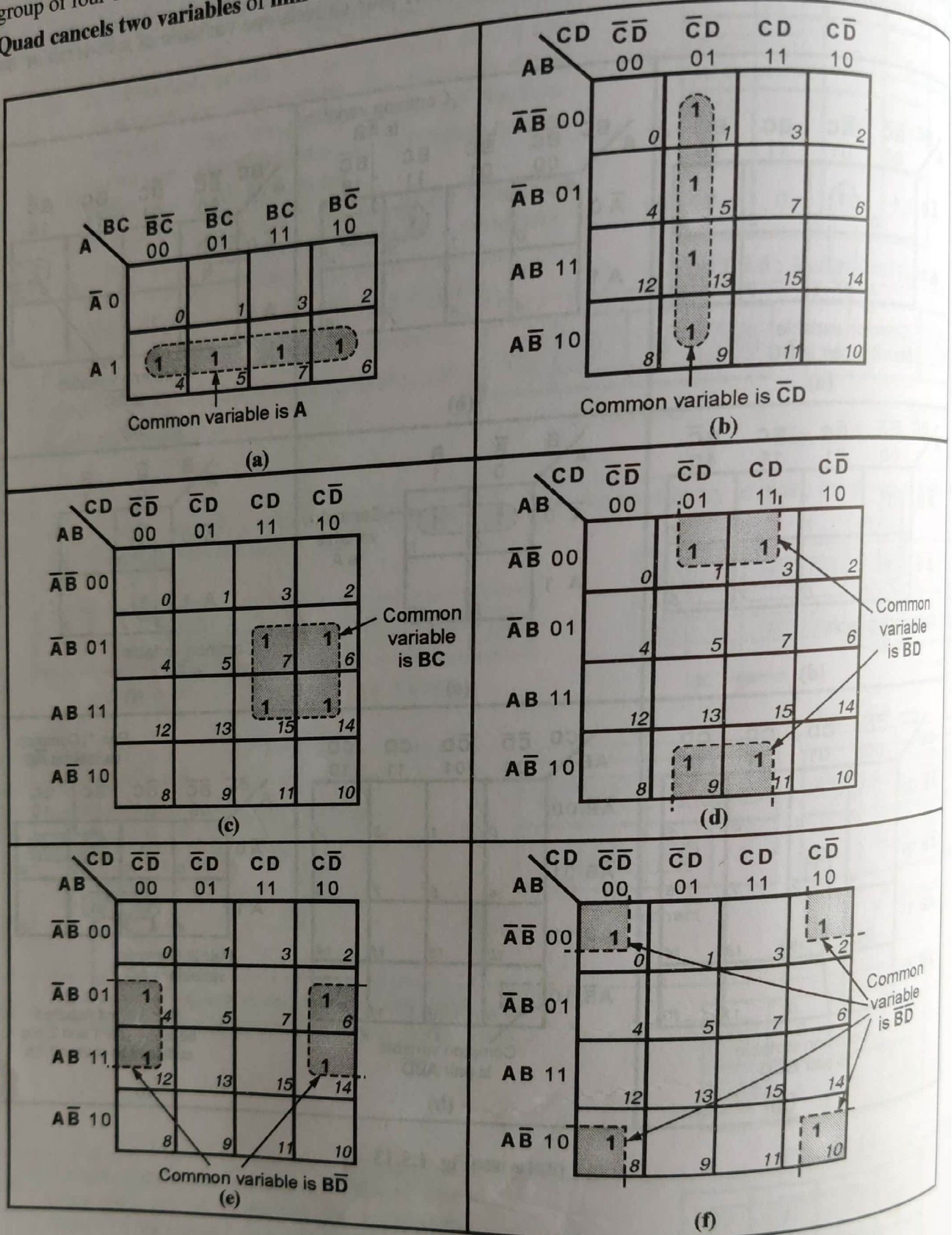
A group of two adjacent cell in a Karnaugh map is called pair. **A pair cancels one variable of min-term or max-term** in a K-map simplification.



(1E26 to 1E30)**Fig. 1.5.13**

1.5.3(B) Grouping of Four Adjacent 1's (Quad)

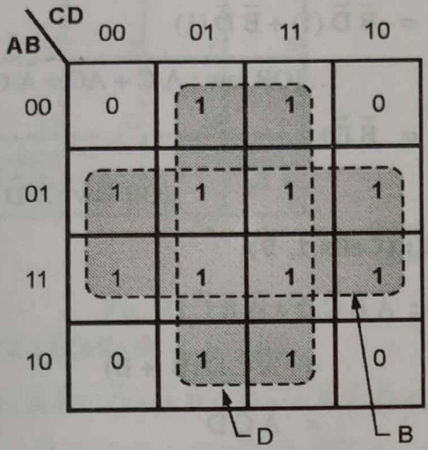
- A group of four 1's that are adjacent to one another is called **quad.**
- A **Quad cancels two variables** of **min-term** or **max-term** in a K-map simplification



(1E31 to (1E33)**Fig. 1.5.14 : Grouping of Four Adjacent 1's (Quad)**

1.5.3(C) Grouping of Eight Adjacent 1's (Octet)

A group of eight adjacent cells in a Karnaugh Map. An **octet cancels three** variables of min-term or max-term in a K-map simplification



Use of Octet in K-map

Fig 1.5.15 : Grouping of Eight Adjacent 1's (Octet)

1.5.3(D) Don't Care Condition

GQ. What is meant by 'don't care terms.? How they play an important role in express minimization ?

(1) The 'Don't care' condition says that can use the blank cells of a K-map to make a group of the variables.

(2) To make a group of cells, we can use the 'don't care' cells to make a large group of cells.

(3) The cross (X) symbol is used to represent the 'don't care' cell in K-map. It is also represented by 'd' in the truth-tables.

(4) **Don't care terms are important** to consider in minimising using K-maps and also the Quine-Mc Cluskey algorithm

1.5.4 Problems based on SOP Form

UEx. 1.5.1 **Q. 1(b), May 18, 4 Marks**

Simplify and implement following expression using K-map.

$Y = \sum m\,(1, 5, 6, 7, 11, 12, 13, 15)$

☑ **Soln. :**

- **Step I :** The given expression has 4 variables, so we draw a 4 × 4 k-map.
- **Step II :** Formation of Groups.

Refer Fig Ex. 1.5.1.

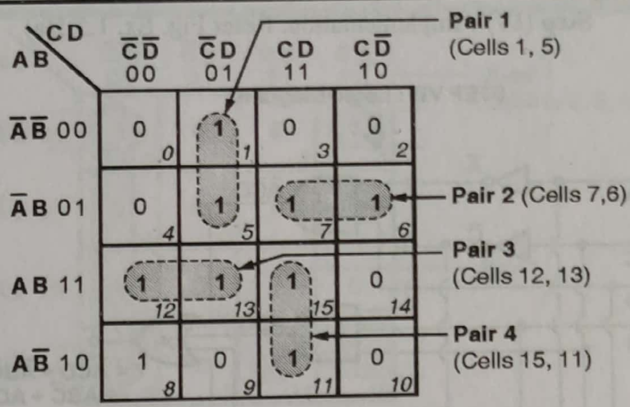


Fig. Ex. 1.5.1

- **Note :** There is a quad formed by the cells (5, 7, 13, 15) but then it is redundant, so we have not included in the expression :
- **Step 3 :** Expressions for groups formed

☞ **Pair 1 : (Cells 1, 5)**

Expression : $\bar A \bar B \bar C D + \bar A B \bar C D$

$= \bar A \bar C D(\bar B + B)$

$= \bar A \bar C D$ [OR law : $\bar B + B = 1$] ...(i)

☞ **Pair 2 : Cells (7, 6)**

Expression : $\bar A BCD + \bar A BC\bar D$

$= \bar A BC(D + \bar D)$ [OR law : $D + \bar D = 1$]

$= \bar A BC$ ...(ii)

☞ **Pair 3 : (Cells 12, 13)**

Expression : $AB(\bar C \bar D + \bar C D)$

$= AB\bar C(\bar D + D)$

$= AB\bar C$ [OR law : $\bar D + D = 1$] ...(iii)

☞ **Pair 4 : (Cells 11,15)**

Expression $= (AB + A\bar B)CD$

$= A(B + \bar B)CD$ [OR law : $B + \bar B = 1$]

$= ACD$ ...(iv)

Using (i), (ii), (iii), (iv) expression Y is :

$Y = \bar A \bar C D + \bar A BC + AB\bar C + ACD$

Method of Grouping:

1) Method of Grouping Two adjacent 1's (Pair)

A \ BC	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
$\bar{A}$	0	0	1	1
A	0	0	0	0

$\bar{A}BC$ $\bar{A}B\bar{C}$ — group of adjacent 1's i.e. a pair.

simplification →

$$Y = \bar{A}BC + \bar{A}B\bar{C}$$
$$= \bar{A}B(C+\bar{C})$$
$$= \bar{A}B \quad \text{.. where } C+\bar{C}=1$$

(Thus C is eliminated)

2) Grouping Four Adjacent ones (Quad)

A group of four 1's that are adjacent to one another is called Quad.

A Quad cancels two variables of min-term or max-term in k-map.

AB \ CD	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	0	0	0
$\bar{A}B$	0	0	0	0
AB	0	0	0	0
$A\bar{B}$	1	1	1	1

↳ $Y = A\bar{B}$

simplification:

$$Y = A\bar{B}\bar{C}\bar{D} + A\bar{B}\bar{C}D + A\bar{B}CD + A\bar{B}C\bar{D}$$

$$= A\bar{B}[\bar{C}\bar{D} + \bar{C}D + CD + C\bar{D}]$$

$$= A\bar{B}\,[\bar{C}(\bar{D}+D) + C(D+\bar{D})]$$

$$= A\bar{B}\,[\bar{C}+C] = A\bar{B}$$

* Grouping Eight Adjacent ones (octet):

- It is possible to form a group of 8 adjacent ~~cells~~ one's.

- Thus octet eleminates 3 variables.

AB \ CD	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	1	1	1	1
$\bar{A}B$	1	1	1	1
AB	0	0	0	0
$A\bar{B}$	0	0	0	0

$Y = \bar{A}$

simplification:-

$$Y = \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}CD + \bar{A}\bar{B}C\bar{D} + \bar{A}B\bar{C}\bar{D} +$$
$$\bar{A}B\bar{C}D + \bar{A}BCD + \bar{A}BC\bar{D}$$

$\therefore \; = \bar{A}\bar{B}\bar{C}\,(\bar{D}+D) + \bar{A}\bar{B}C\,(D+\bar{D}) + \bar{A}B\bar{C}\,(\bar{D}+D)$
$+ \bar{A}BC\,(D+\bar{D})$

$= \bar{A}\bar{B}\,(\bar{C}+C) + \bar{A}B\,(\bar{C}+C)$

$= \bar{A}\,(\bar{B}+B)$

$Y = \bar{A}$ (The only variable that remains same is $\bar{A}$)

* Don't care condition :

1) the don't care condition says that can use the blank cells of a K-map to make a group of the variables.

2) To make a group of cells, we can use the 'dont care' cells to make a large group of cells.

3) The cross (x) symbol is used to represent the 'don't care' cell in K-map. It also represented by 'd' in the truth table

4) Don't care terms are important to consider in minimising using K-maps & also the Quine-Mc cluskey algorithm.

classmate
Date
Page

* standard Forms.

Min-term & Max term.

Min Term :-

The Product of all literals, either with complement or without complement, is known as minterm.

For ex: Boolean function in 2 variables, the minterms are

$m_0 = \bar{x} \cdot \bar{y}$, $m_1 = \bar{x} \cdot y$,

$m_2 = x \cdot \bar{y}$, $m_3 = x \cdot y$

Minterm form values:

if the variable value is 1, we will take the variable without it's complement.

If the variable value is 0, takes it complement

$x = 1$, $y = 0$

Maxterm :-

The sum of all literals, either with complement or without complement is known as maxterm.

eg:

$M_0 = x + y$, $M_1 = x + \bar{y}$,

$M_2 = \bar{x} + y$, $M_3 = \bar{x} + \bar{y}$

Minimization using SOP & POS using K-Map.

Standard forms

- Sum of Products (SOP)
- Product of sum (POS)

* Sum of Product : (SOP)

- A Boolean function (expression) consisting purely of Minterms (Product terms) is said to be sum of product form.

- A Boolean expression involving **AND** terms with one or more literals each, **OR**ed together

- It is logical sum of the two or more logical product terms.

- It is basically an OR operation of AND operated variable.

eg: $Z = AB + BC + ABD$

$Z = AB + BC + BD + ABC$

Truth Table -

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

Product of sums (POS)

A Boolean expression consisting purely of Maxterms (sum terms) is said to be Product of sum form.

- A Boolean expression where several sum terms are multiplied together (AND operation).

- A product of sums expression is a logical product of two or more logical sum terms.

- It is basically an AND operation of OR operated variables.

eg: $Z = (A+B) \cdot (B+C) \cdot (A+B+D)$

$Z = (A+B) \cdot (B+C) \cdot (B+D) \cdot (A+B+C)$

Truth Table

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

* Two variables expressions Min terms & Max terms.

X	Y	Min-term	Max term
0	0	$\bar{X} \cdot \bar{Y} = m_0$	$X+Y = M_0$
0	1	$\bar{X} \cdot Y = m_1$	$X+\bar{Y} = M_1$
1	0	$X \cdot \bar{Y} = m_2$	$\bar{X}+Y = M_2$
1	1	$X \cdot Y = m_3$	$\bar{X}+\bar{Y} = M_3$

* Three variables Min-term & Max-term

X	Y	Z	Min-term	Max-term
0	0	0	$\bar{X} \cdot \bar{Y} \cdot \bar{Z} = m_0$	$X+Y+Z = M_0$
0	0	1	$\bar{X} \cdot \bar{Y} \cdot Z = m_1$	$X+Y+\bar{Z} = M_1$
0	1	0	$\bar{X} \cdot Y \cdot \bar{Z} = m_2$	$X+\bar{Y}+Z = M_2$
0	1	1	$\bar{X} \cdot Y \cdot Z = m_3$	$X+\bar{Y}+\bar{Z} = M_3$
1	0	0	$X \cdot \bar{Y} \cdot \bar{Z} = m_4$	$\bar{X}+Y+Z = M_4$
1	0	1	$X \cdot \bar{Y} \cdot Z = m_5$	$\bar{X}+Y+\bar{Z} = M_5$
1	1	0	$X \cdot Y \cdot \bar{Z} = m_6$	$\bar{X}+\bar{Y}+Z = M_6$
1	1	1	$X \cdot Y \cdot Z = m_7$	$\bar{X}+\bar{Y}+\bar{Z} = M_7$

Note – complement of a min-term is a max term & vice-versa.

insem oct 22

Examples.

Q.1] Simplify the expression F (A, B, C, D) = Σm (3, 4, 5, 7, 9, 13, 14, 15) using the K-map method.

→ Step 1 – The given expression has 4 variables, so we draw a 4×4 K-map.

Step II – Formation of Groups.

AB \ CD	$\bar{C}\bar{D}$ 00	$\bar{C}D$ 01	CD 11	$C\bar{D}$ 10
$\bar{A}\bar{B}$ 00	0 (0)	0 (1)	1 (3)	0 (2)
$\bar{A}B$ 01	1 (4)	1 (5)	1 (7)	0 (6)
AB 11	0 (12)	1 (13)	1 (15)	1 (14)
$A\bar{B}$ 10	0 (8)	1 (9)	0 (11)	0 (10)

Pair 1 (cells 3, 7)

Pair 2 (cells 4, 5)

Pair 4 (cells 15, 14)

Pair 3 (cells 13, 9)

Note – There is a quad formed by the cells (5, 7, 13, 15) but then it is redundant, so we have not included in the expression.

Step 3 :- Expressions

Pair 1 : cells (3, 7)

Expression $= \bar{A}\bar{B}CD + \bar{A}BCD$
$= \bar{A}CD(\bar{B}+B)$
$= \bar{A}CD$ [OR Law: $\bar{B}+B=1$]
(i)

Pair 2 : cells (4, 5)

Expression $= \bar{A}B\bar{C}\bar{D} + \bar{A}B\bar{C}D$
$= \bar{A}B\bar{C}(\bar{D}+D)$
$= \bar{A}B\bar{C}$ [OR law: $\bar{D}+D=1$]
(ii)

Pair 3 : cells (13, 9)

Expression $= AB\bar{C}D + A\bar{B}\bar{C}D$
$= A\bar{C}D(B+\bar{B})$
$= A\bar{C}D$ [OR Law: $B+\bar{B}=1$]
(iii)

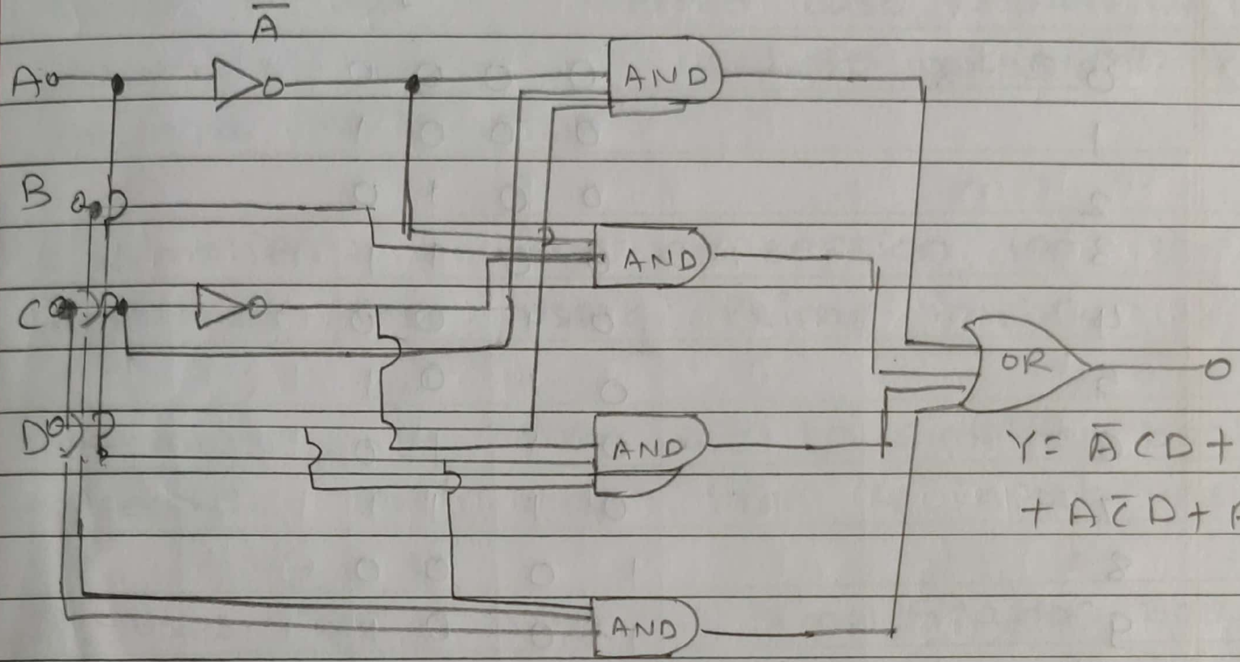
Pair 4 : cells (15, 14)

Expression $= ABCD + ABC\bar{D}$
$= ABC(D+\bar{D})$
$= ABC$ [OR Law $D+\bar{D}=1$]
(iv)

using (i), (ii), (iii), (iv) expression Y is

$Y = \bar{A}CD + \bar{A}B\bar{C} + A\bar{C}D + ABC$

Step 4: Implementation.



$Y = \bar{A}CD + \bar{A}B\bar{C} + A\bar{C}D + ABC$

Fig: Logic Diagram.

* Practice Example :-

① Minimize the following boolean function.

$F(A, B, C, D) = \sum m(0, 1, 2, 5, 7, 8, 9, 10, 13, 15)$

↳ Ans — $BD + \bar{C}D + \bar{B}\bar{D}$

② $F(ABCD) = \sum m(1, 5, 6, 7, 11, 12, 13, 15)$

Ans — $Y = \bar{A}\bar{C}D + \bar{A}BC + AB\bar{C} + ACD$

Binary Representation.

Number	Binary Representation
0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	0 1 0 1
6	0 1 1 0
7	0 1 1 1
8	1 0 0 0
9	1 0 0 1
10	1 0 1 0
11	1 0 1 1
12	1 1 0 0
13	1 1 0 1
14	1 1 1 0
15	1 1 1 1

SOP. (Sum of Product)

step 1 → Prepare the K-map & place 1's according to the given truth Table. or logical expression. Fill the remaining cells by 0's

step 2 → Formation of groups.

step 3 → Expressions for groups formed.

A	B	C	D	min-term	Max term
0	0	0	0	$\bar{A}\bar{B}\bar{C}\bar{D}$	$A+B+C+D$
0	0	0	1	$\bar{A}\bar{B}\bar{C}D$	$A+B+C+\bar{D}$
0	0	1	0	$\bar{A}\bar{B}C\bar{D}$	$A+B+\bar{C}+D$
0	0	1	1	$\bar{A}\bar{B}CD$	$A+B+\bar{C}+\bar{D}$
0	1	0	0	$\bar{A}B\bar{C}\bar{D}$	$A+\bar{B}+C+D$
0	1	0	1	$\bar{A}B\bar{C}D$	$A+\bar{B}+C+\bar{D}$
0	1	1	0	$\bar{A}BC\bar{D}$	$A+B+C+\bar{D}$
0	1	1	1	$\bar{A}BCD$	$A+\bar{B}+\bar{C}+\bar{D}$
1	0	0	0	$A\bar{B}\bar{C}\bar{D}$	$\bar{A}+B+C+D$
1	0	0	1	$A\bar{B}\bar{C}D$	$\bar{A}+B+C+\bar{D}$
1	0	1	0	$A\bar{B}C\bar{D}$	$\bar{A}+B+\bar{C}+D$
1	0	1	1	$A\bar{B}CD$	$\bar{A}+B+\bar{C}+\bar{D}$
1	1	0	0	$AB\bar{C}\bar{D}$	$\bar{A}+\bar{B}+C+D$
1	1	0	1	$AB\bar{C}D$	$\bar{A}+\bar{B}+C+\bar{D}$
1	1	1	0	$ABC\bar{D}$	$\bar{A}+\bar{B}+\bar{C}+D$
1	1	1	1	$ABCD$	$\bar{A}+\bar{B}+\bar{C}+\bar{D}$

Example.

① simplify the expression:
$F(A,B,C,D) = \Sigma m(1,3,7,11,15) + d(0,2,5)$ using the K-map method.

→ step 1: given expression has 4 variables. so, we draw 4×4 variable K-map.

step 2: Formation of groups.

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

Quad 1 (cells 1,3,5,7)

Quad 2 (cells, 3,7, 15, 11)

step 3: Expressions.

I) Quad 1 – cells (1,3,5,7)

$= \bar{A}D$

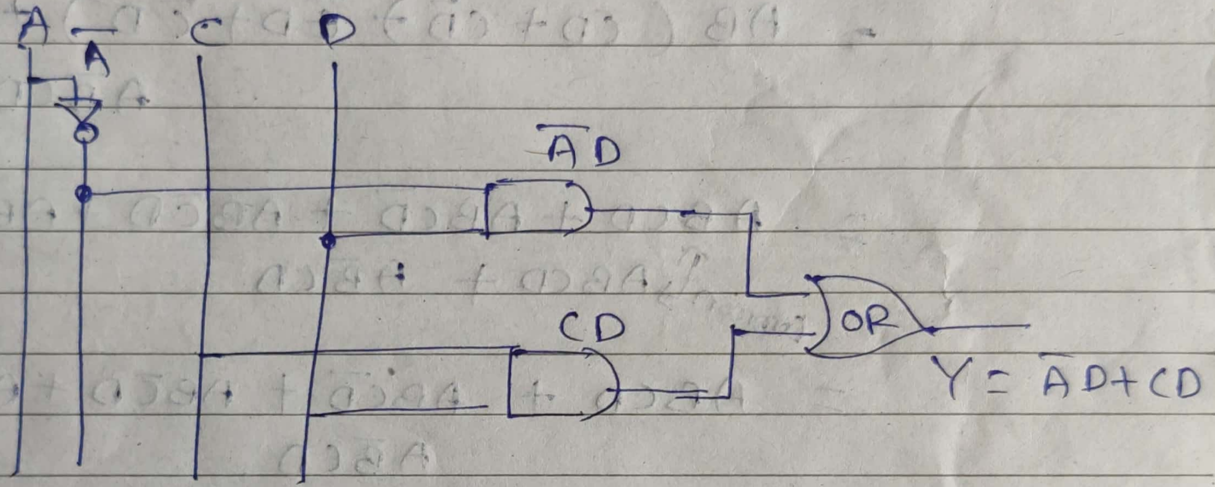
II) Quad 2 – Cells (3,7,15,11)

$= CD$

The minimised expression from I & II

$F(A,B,C,D) = \bar{A}D + CD$

Step IV = Implementation.



(2) simplify the expression
$F(A,B,C,D) = \Pi M\ (0,1,4,5,6,8,9,12,13,14)$
using K-map method.

→ step 1 – There are 4 variables – so, we draw 4×4 K-map

step 2 – Formation of group.

K-map in POS Form.

AB \ CD	$C+D$ ~~$A+B$~~ 00	$C+\bar{D}$ ~~$A+B$~~ 01	$\bar{C}+\bar{D}$ 11	$\bar{C}+D$ 10
$A+B$ 00	0 (0)	0 (1)	1 (3)	1 (2)
$A+\bar{B}$ 01	0 (4)	0 (5)	1 (7)	0 (6)
$\bar{A}+\bar{B}$ 11	0 (12)	0 (13)	1 (15)	0 (14)
$\bar{A}+B$ 10	0 (8)	0 (9)	1 (11)	1 (10)

← pair 1 (cells 6, 14)

octet (1) [cells 0, 1, 4, 5, 8, 9, 12, 13)

step 3 :- Expression for group.

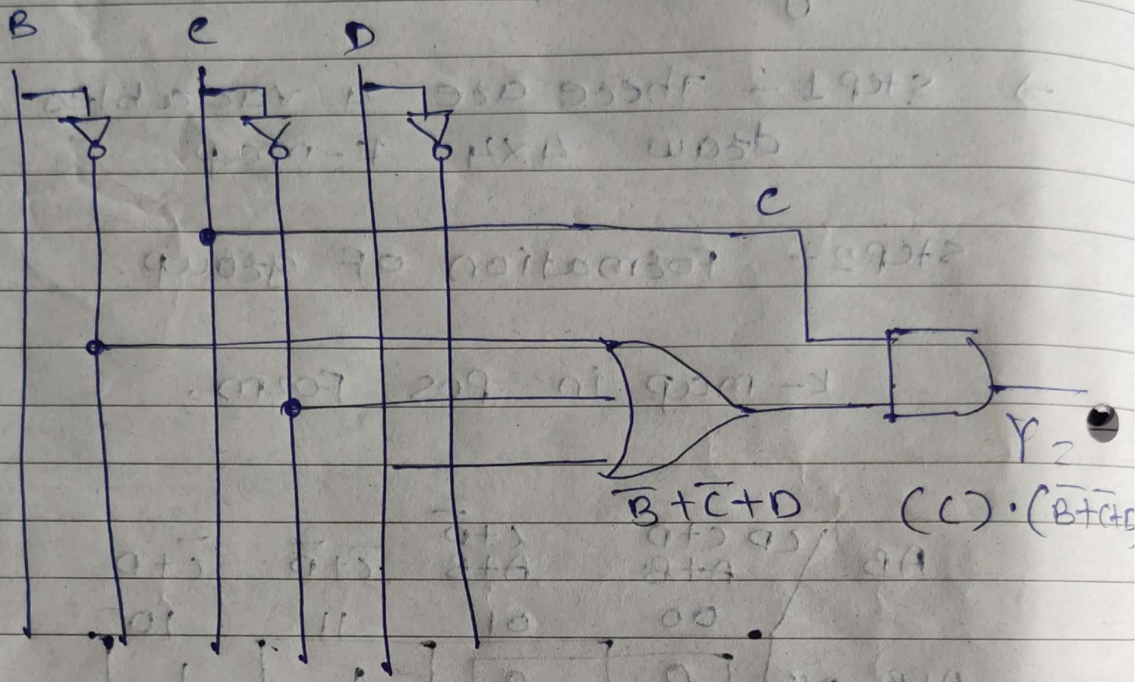
I) Octet (1) – (cells 0, 1, 4, 5, 12, 13, 8, 9)

$y = C$

II) pair (1) – (cells 6, 14)
$= \bar{B} + \bar{C} + D$

$= (C)\cdot(\overline{B}+\overline{C}+D)$

Logic Diagram:



* POS - Product of sum. (∵ symbol ΠM)

Steps —

1) Select K-map according to number of variables.
2) Identify max-terms as given in the Problem.
3) For POS, Put 0's in blocks of K-map respective to the max-terms & 1's elsewhere -, (group of 0's instead of 1's)

* Example 1:

Q) Reduce the Following using K-map Technique.

$$F(A,B,C,D) = \Pi M(0,2,3,8,9,12,13,15)$$

→ step 1 — There are 4 variables, we draw 4x4 K-map in P.O.S. form.

step 2 — Formation of groups.

• K-map in POS form:

AB \ CD	$C+D$ 00	$C+\bar{D}$ 01	$\bar{C}+\bar{D}$ 11	$\bar{C}+D$ 10
$A+B$ 00	0 [0]	1 [1]	0 [3]	0 [2]
$A+\bar{B}$ 01	1 [4]	1 [5]	1 [7]	1 [6]
$\bar{A}+\bar{B}$ 11	0 [12]	0 [13]	0 [15]	1 [14]
$\bar{A}+B$ 10	0 [8]	0 [9]	1 [11]	1 [10]

(cells 0,2) ← Pair 1

Pair 2 (cells 3,2)

Pair 3 (cells 13,15)

Quad 4 (cells 12, 13, 8, 9)

All zeros have been grouped

* step 3: Expression for groups.

I) pair 1 (cells 0, 2)

$= A + B + D$

II) Pair 2 (cells 3, 2)

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

III) Pair 3 (cells 13, 15)

$= \overline{A} + \overline{B} + \overline{D}$

IV) Quad 1 (cells 8, 9, 12, 13)

$= \overline{A} + C$

* step 4 → expression from I, II, III, IV

$$F(A, B, C, D) = (A + B + D) \cdot (A + B + \overline{C}) \cdot (\overline{A} + \overline{B} + \overline{D}) \cdot (\overline{A} + C)$$

* Quine MCKLUSKEY Method:-

- Quine McCluskey method also known as the tabulation method is used to minimize the Boolean functions.

- It simplifies boolean expression into the simplified form using prime implicants.

- This method is convenient to simplify boolean expressions with more than 4 input variables.

- It uses an automatic simplification routine

* Terminologies :

Implicant :
Implicant is defined as a group of 1's (For minterm)

Prime implicant :-
It is the largest possible group of 1's (For minterm)

Essential Prime implicant :-
Essential Prime implicants are groups that cover at least one minterm which can be covered by other applicants.

(∵ this method uses decimal to binary representation)

Insem-22

Example -

$F(A,B,C,D) = \sum m(0,1,3,7,8,9,11,15)$

⇒ Table 1

Minterm	A	B	C	D
0	0	0	0	0
1	0	0	0	1
3	0	0	1	1
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
11	1	0	1	1
15	1	1	1	1

Table 2 - consist of Group, minterms & variables.

Group	Minterms	A	B	C	D
G-0	0	0	0	0	0
G-1	1	0	0	0	1
	8	1	0	0	0
G-2	3	0	0	1	1
	9	1	0	0	1
G-3	7	0	1	1	1
	11	1	0	1	1
G-4	15	1	1	1	1

Table 3 :- Consist of Group, minterm & variables.

Group	Minterm	Variables A	B	C	D
0	0, 1	0	0	0	–
	0, 8	–	0	0	0
1	1, 3	0	0	–	1
	1, 9	–	0	0	1
	8, 9	1	0	0	–
2	3, 7	0	–	1	1
	3, 11	–	0	1	1
	9, 11	1	0	–	1
3	7, 15	–	1	1	1
	11, 15	1	–	1	1

Table 4 – In this case there are 4 groups can each group contain combinations of two min terms.
we carry now merging of min term pairs from the group.

Group	Minterms	Variables A	B	C	D
0	0, 1, 8, 9	–	0	0	–
	0, 8, 1, 9	–	0	0	–
1	1, 3, 9, 11	–	0	–	1
	1 9, 3, 11	–	0	–	1
2	3, 7, 11, 15	–	–	1	1
	3, 11, 7, 15	–	–	1	1

PI (Prime Implicants Table)

P.I	Decimal	0	1	3	7	8	9	11	15
✓ $\bar{B}\ \bar{C}$	0, 1, 8, 9	X	X			X	X		
$\bar{B}\ D$	1, 3, 9, 11		X	X			X	X	
✓ $C\ D$	3, 7, 11, 15			X	X			X	X

(Minterms: 0 1 3 7 8 9 11 15)

$$\boxed{Y = \bar{B}\bar{C} + CD}$$

$F(A, B, C, D) = \bar{B}\bar{C} + CD$

Example - (For practice)

(1) $Y(A,B,C,D) = \Sigma m(0,1,2,3,5,7,8,9,11,14)$

Ans - $F(A,B,C,D) = \bar{A}\bar{B} + \bar{B}\bar{C} + \bar{A}D + \bar{B}D$

(2) $F(A,B,C,D) = \Sigma m(0,1,3,7,10,14)$

(3) $F(A,B,C,D) = \Sigma m(1,5,6,12,13,4) + d(2,4)$

Ans - $F(A,B,C,D) = B\bar{C} + B\bar{D} + \bar{A}\bar{C}D$

(5) $F(A,B,C,D) = \Sigma m(0,1,2,5,6,7)$

Ans $= \bar{A}\bar{B}\bar{C} + \bar{A}C + \bar{A}BD$