Boolean Algebra

Boolean algebra is an algebra that may be defined with a set of elements, a set of operators and a number of unproved axioms or postulates. A set of elements is any collection of objects having common property. The set of operators AND (Boolean product) operation ⊙, the OR (Boolean Sum) operation +, and the NOT (complement operation) ' are defined in the set.

Boolean Postulates

The postulates are the basic assumptions from which it is possible to deduce the rules, theorems and properties of the system.

① Closure

A set S is closed with respect to a binary operator if, for every pair of elements of S, the binary operator specifies a rule for obtaining a unique element of S. $N = \{1,2,3,4 \ldots\}$ (natural nos) is closed under binary operator (+) since any $a, b \in N$ we obtain a unique $c \in N$ by the operation $a+b=c$.
It is not closed with respect to binary operator minus (−) because $2-3=-1$ where $-1 \notin N$

② Associative law

A binary operator $*$ on a set S is said to be associative whenever:

$$(x * y) * z = x * (y * z) \quad \text{for all } x, y, z \in S$$

③ Commutative law

A binary operator $*$ on a set S is said to be commutative whenever:

$$x * y = y * x \quad \text{for all } x, y \in S$$

④ Identity element

A set S is said to have an identity element with respect to a binary operation $*$ on S if there exists an element $e \in S$ with the property:

$$e * x = x * e = x \quad \text{for every } x \in S$$

⑤ Inverse

A set S having the identity element e with respect to a binary operator $*$ is said to have an inverse whenever, for every $x \in S$, there exists an element $y \in S$ such that

$$x * y = e.$$

$$a + (-a) = 0.$$

⑥ Distributive law

If $*$ and $\cdot$ are two binary operators on a set S, $*$ is said to be distributive over $\cdot$ whenever:

$$x * (y \cdot z) = (x * y) \cdot (x * z)$$

Q1. Use Algebraic manipulation to convert

Canonical and Standard forms

Minterms and Maxterms

A binary variable may appear either in its normal form (x) or in its complement form (x'). Consider two binary variables x and y combined with an AND operation. There are four possible combinations. $x'y'$, $x'y$, xy' and xy. Each of these four AND terms is called a minterm or a standard product. n variables can be combined to form 2^n minterms.

In similar, n variables forming an OR term with each variable provide 2^n possible combinations. These are called maxterms or Standard Sums.

			Minterms		Maxterms	
x	y	z	Term	Designation	Term	Designation
0	0	0	$x'y'z'$	m_0	$x+y+z$	M_0
0	0	1	$x'y'z$	m_1	$x+y+z'$	M_1
0	1	0	$x'yz'$	m_2	$x+y'+z$	M_2
0	1	1	$x'yz$	m_3	$x+y'+z'$	M_3
1	0	0	$xy'z'$	m_4	$x'+y+z$	M_4
1	0	1	$xy'z$	m_5	$x'+y+z'$	M_5
1	1	0	xyz'	m_6	$x'+y'+z$	M_6
1	1	1	xyz	m_7	$x'+y'+z'$	M_7

Table: Minterms and maxterms for three binary variables.

Canonical form - Boolean functions expressed as a sum of minterms or product of maxterms are said to be in canonical form. ~~Std form~~ - contain all variables either in true form or complemented form.

Qn. Use Algebraic manipulation to convert:
(i) $F(x,y,z) = xy+y+z$ into canonical PoS
(ii) $F(x,y,z) = (x+y+z)(x'+y+z)(x+y'+z)(x+z)$ into Std PoS
(S3 (S) May 2019)

Sum of minterms (Sum of Products)

- two or more AND functions ORed together.

eg. $AB + CD$.

$\bar{A}B + BCD$

$AB\bar{C} + \bar{D}E\bar{F} + FGH + A\bar{F}G$

Sum of products form can also contain a term with a single variable. $A + BCD + EFG$.

It is a useful form of Boolean expression because of the straight forward manner in which it can be implemented with logic gates. Canonical SOP contains all the term with all the variables either in complemented or uncomplemented form. eg. $A\bar{B}C + A\bar{B}\bar{C} + \bar{A}B$

Product of Sums (Product of Maxterms)

- It is the dual of Sum of products form.
- It is the AND of two or more OR functions.

eg. $(A+B)(B+C+D)$

$(A+B+\bar{C})(\bar{D}+\bar{E}+F)(F+G+H)(A+\bar{F}+G)$

Product of Sums expression can also contain a single variable term, such as $A(B+C+D)(E+F+G)$.

In canonical POS, each sumterm contains all the variables either in complemented or uncomplemented form. eg. $(A+B).(\bar{A}+B).(A+\bar{B})$

Representation of SOP

$$F = A'B'C + AB'C' + AB'C + ABC' + ABC$$

$$= m_1 + m_4 + m_5 + m_6 + m_7$$

Or the short notation.

$$F(A,B,C) = \Sigma(1,4,5,6,7)$$

Representation of POS

$$F = (x+y+z)(x+y'+z)(x'+y+z)(x'+y+z')$$

$$= M_0 M_2 M_4 M_5$$

Or, $F(x, y, z) = \Pi(0, 2, 4, 5)$.

Conversion between Canonical Forms

The complement of a function expressed as the sum of minterms equals the sum of minterms missing from the original function. Consider the function:

$$F(A, B, C) = \Sigma(1, 4, 5, 6, 7)$$

Complement is

$$F'(A, B, C) = \Sigma(0, 2, 3) = m_0 + m_2 + m_3$$

Take the complement of F' by DeMorgan's theorem,

$$F = (m_0 + m_2 + m_3)' = m_0' \cdot m_2' \cdot m_3' = M_0 M_2 M_3$$

$$= \Pi(0, 2, 3)$$

$F(x, y, z) = \Pi(0, 2, 4, 5)$ is expressed in the product of maxterm form. Its conversion to sum of minterms is

$$F(x, y, z) = \Sigma(1, 3, 6, 7)$$

Standard Forms

The function may contain one, two or any number of literals. 2 types of standard forms: Sum of products and product of sums.

Sum of products is a Boolean expression containing AND terms called product terms of one or more literals each. eg. $F_1 = y' + xy + x'yz'$.

Product of sums is a Boolean expression containing OR terms, called sum terms. Each term may have any number of literals.

eg. $F_2 = x(y' + z)(x' + y + z' + w)$.

Boolean Functions

A Boolean function is an expression formed with binary variables, the two binary operators OR and AND, the unary operator NOT, parenthesis and equal sign. For a given value of the variables, the function can be either 0 or 1. For eg.

$F_1 = xyz'$

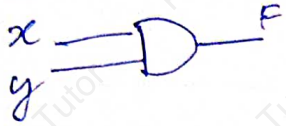
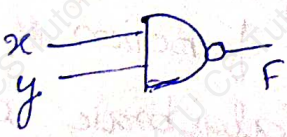
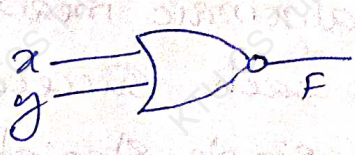
The function F_1 is equal to 1 if $x = 1$ and $y = 1$ and $z' = 1$. Otherwise $F_1 = 0$. A boolean function may also be represented in a truth table. To represent a function in a truth table, 2^n combinations of 1's and 0's of the n binary variables and a column showing the combinations for which the function is equal to 1 or 0 is used.

For eg. truth table for the functions $F_1 = xyz'$ $F_2 = x + y'z$, $F_3 = x'y'z + x'yz + xy'$ and $F_4 = xy' + xz$ is shown below.

x	y	z	F_1	F_2	F_3	F_4
0	0	0	0	0	0	0
0	0	1	0	1	1	1
0	1	0	0	0	0	0
0	1	1	0	0	1	1
1	0	0	0	1	1	1
1	0	1	0	1	1	1
1	1	0	1	1	0	0
1	1	1	0	1	0	0

Logic Gates

Logic gates are the basic elements that make up a digital system.

Name	Graphic symbol	Algebraic function	Truth table
AND		$F = xy$	
OR		$F = x + y$	
Inverter (NOT)		$F = x'$	
Buffer		$F = x$	
NAND		$F = (xy)'$	
NOR		$F = (x+y)'$	

x	y	F
0	0	0
0	1	0
1	0	0
1	1	1

x	y	F
0	0	0
0	1	1
1	0	1
1	1	1

x	F
0	1
1	0

x	F
0	0
1	1

x	y	F
0	0	1
0	1	1
1	0	1
1	1	0

x	y	F
0	0	1
0	1	0
1	0	0
1	1	0

Exclusive-OR (XOR)

$F = xy' + x'y$
$= x \oplus y$

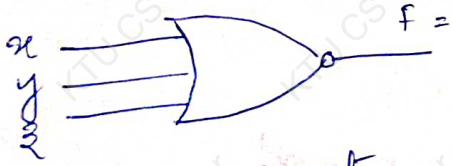
x	y	F
0	0	0
0	1	1
1	0	1
1	1	0

Exclusive NOR or Equivalence

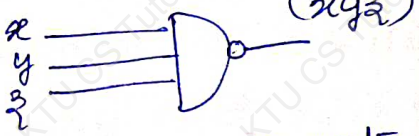
$F = xy + x'y'$
$= x \odot y$

x	y	F
0	0	1
0	1	0
1	0	0
1	1	1

The gates except for the inverter and buffer, can be extended to have more than two inputs. A gate can be extended to have multiple inputs if the binary operation it represents is commutative and associative.



3-input NOR gate



3-input NAND gate

Simplification / Minimization of Boolean Functions

① The Map Method / ~~Veitch~~ Veitch Diagram / Karnaugh Map

The complexity of the digital logic gates that implements a boolean function is directly related to the complexity of the algebraic expression from which the function is implemented. Boolean functions may be simplified by algebraic means but it lacks specific rules to predict each succeeding step. The map method provides a simple straight forward proced

for minimizing boolean functions. The map method first proposed by Veitch and Slightly modified by Karnaugh.

The map is a diagram made up of squares and each square represents one minterm.

m_0	m_1
m_2	m_3

x \ y	0	1
0	$x'y'$	$x'y$
1	xy'	xy

Two variable map.

m_0	m_1	m_3	m_2
m_4	m_5	m_7	m_6

x \ yz	00	01	11	10
0	$x'y'z'$	$x'y'z$	$x'yz$	$x'yz'$
1	$xy'z'$	$xy'z$	xyz	xyz'

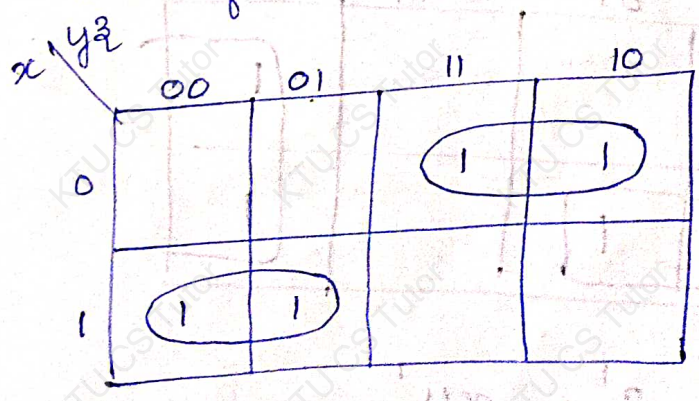
Three variable map.

Simplify the boolean function $F = x'yz + x'yz' + xy'z' + xy'z$ using K-map

① first, mark 1 in each square to represent the function.

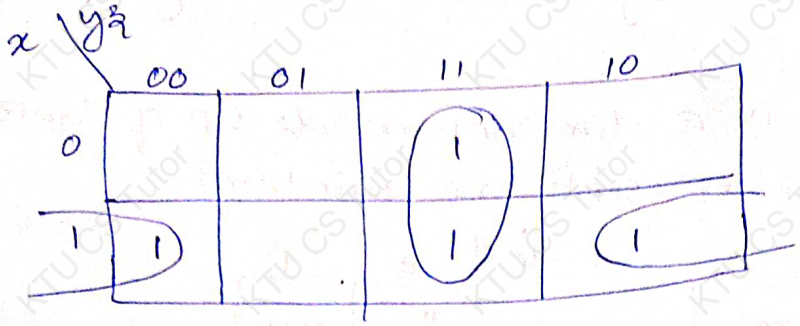
② Group the adjacent 1's.

③ Take the area of the group and represent as the sum of two terms.



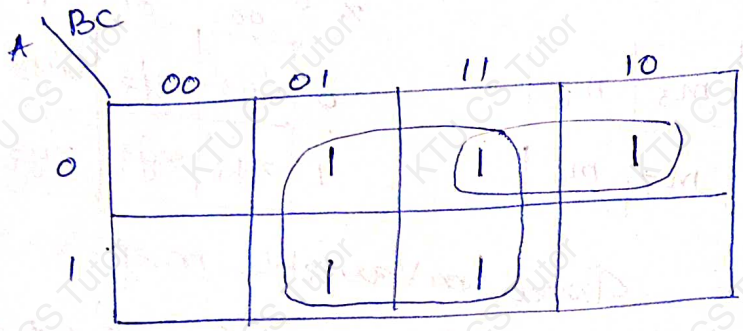
$F = x'y + xy'$.

Qn. $F = x'yz + xy'z' + xyz + xyz'$



$F = yz + xz'$

Qn. $F = A'C + A'B + AB'C + BC$.



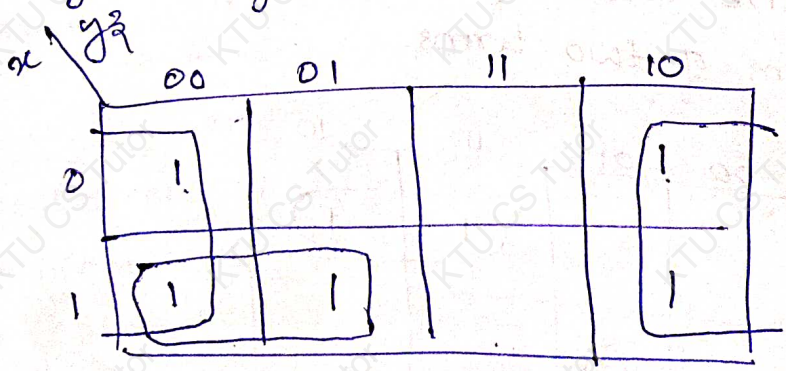
$F = C + A'B$.

Qn. $F(x, y, z) = \Sigma(0, 2, 4, 5, 6)$

$m_0 + m_2 + m_4 + m_5 + m_6$

$000 + 010 + 100 + 101 + 110$

$F = x'y'z' + x'yz' + xy'z' + xy'z + xyz'$.



$F = z' + xy'$

Four Variable Map

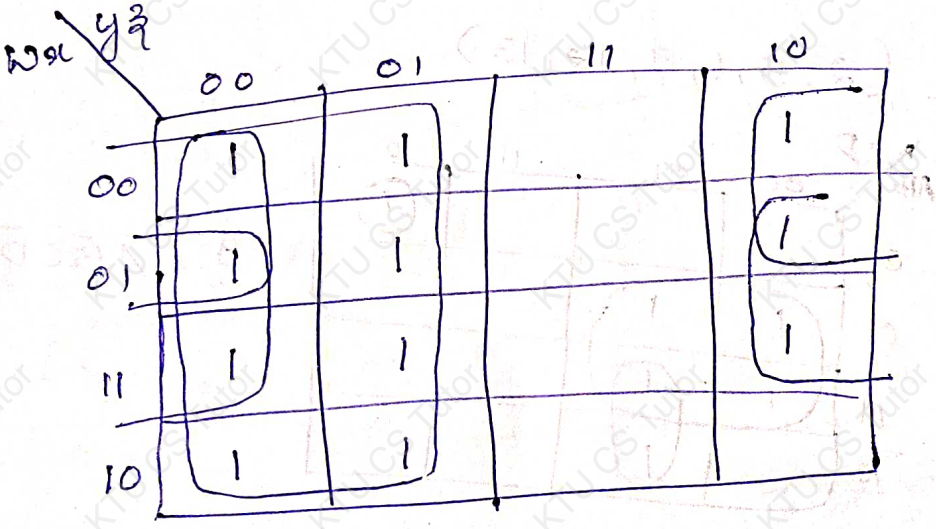
m_0	m_1	m_3	m_2
m_4	m_5	m_7	m_6
m_{12}	m_{13}	m_{15}	m_{14}
m_8	m_9	m_{11}	m_{10}

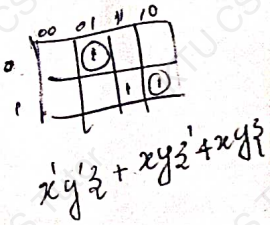
wx \ yz	00	01	11	10
00	$w'x'y'z'$	$w'x'y'z$	$w'x'yz$	$w'x'yz'$
01	$w'xy'z'$	$w'xy'z$	$w'xyz$	$w'xyz'$
11	$wxy'z'$	$wxy'z$	$wxyz$	$wxyz'$
10	$wxy'z'$	$wxy'z$	$wxy'z$	$wxyz'$

Qn. $F(w,x,y,z) = \Sigma(0,1,2,4,5,6,8,9,12,13,14)$

$0000 + 0001 + 0010 + 0100 + 0101 + 0110 +$

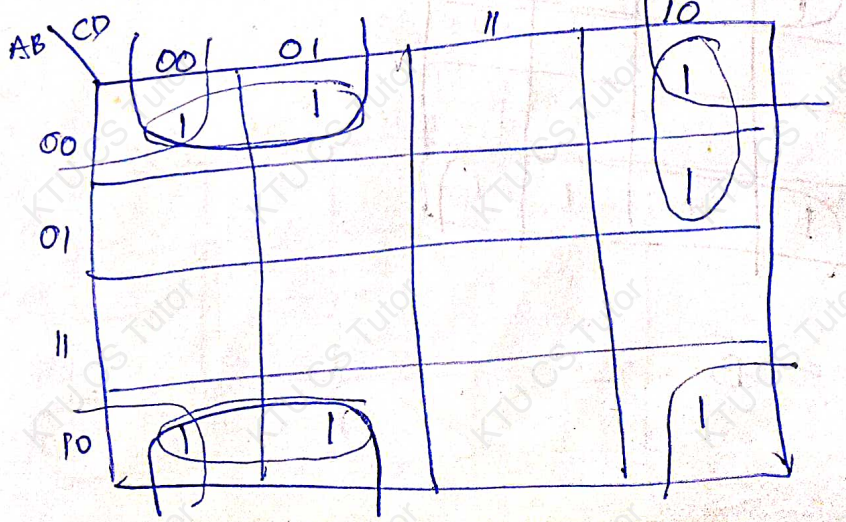
$1000 + 1001 + 1100 + 1101 + 1110$





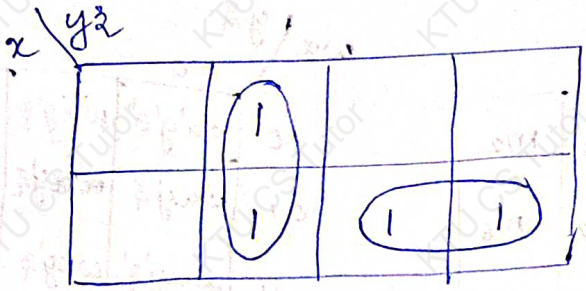
$F = y' + w'z' + xz'$

Qn. $F = A'B'C' + B'CD' + A'BCD' + AB'C'$



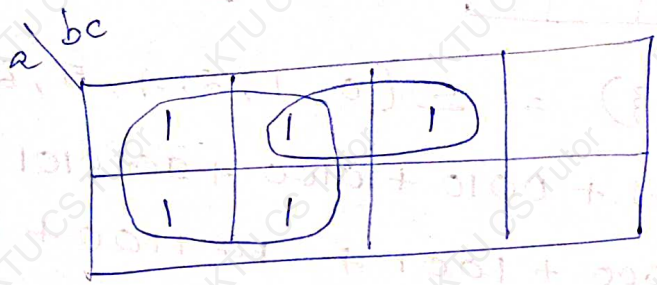
$F = B'D' + B'C' + A'CD'$

Qn. $F = xyz + xy'z + xyz' + x'y'z$



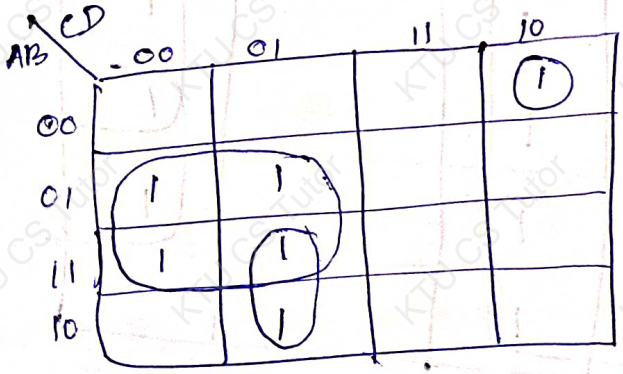
$f = y'z + xy$

Qn. $f = ab'c + a'bc + a'b'c + a'b'c' + ab'c'$



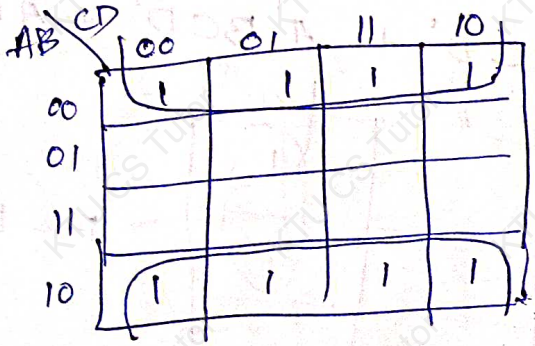
$f = b' + a'c$

Qn. $f = \Sigma(2, 5, 6, 9, 12, 13)$



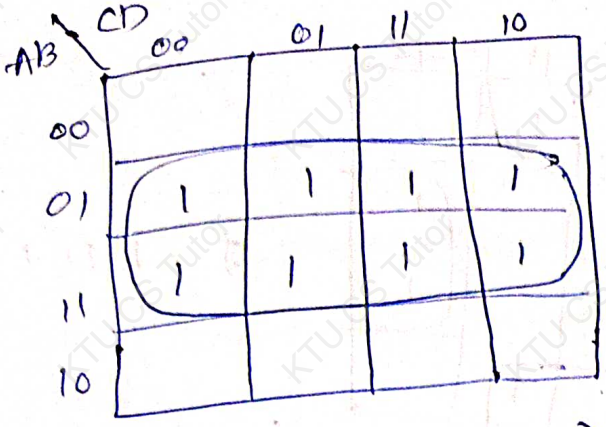
$f = \bar{A}\bar{B}C\bar{D} + B\bar{C} + A\bar{C}D$

Qn. $f = \Sigma(0, 1, 2, 3, 8, 9, 10, 11)$



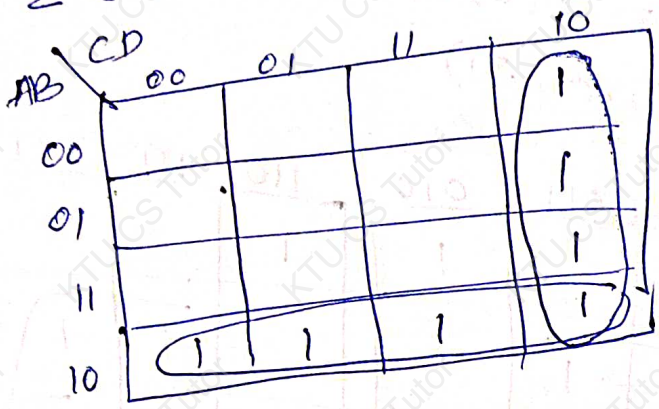
$f = \bar{B}$

Qn. $f = \Sigma(4,5,6,7,12,13,14,15)$



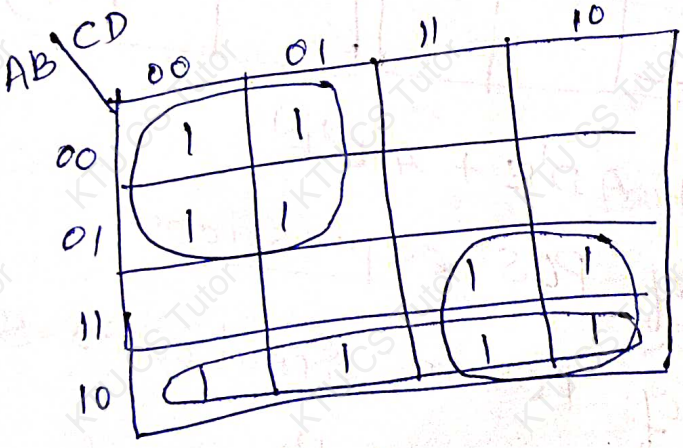
$f = B$

Qn. $f = \Sigma(2,6,8,9,10,11,14)$



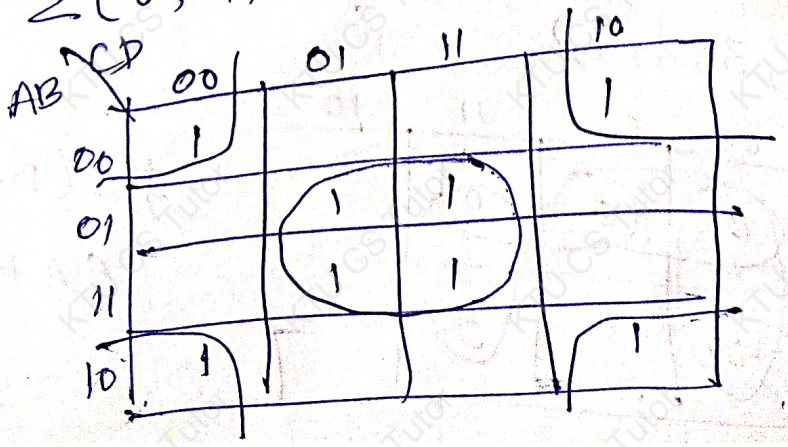
$f = C\bar{D} + A\bar{B}$

Qn. $f = \Sigma(0,1,4,5,8,9,10,11,14,15)$



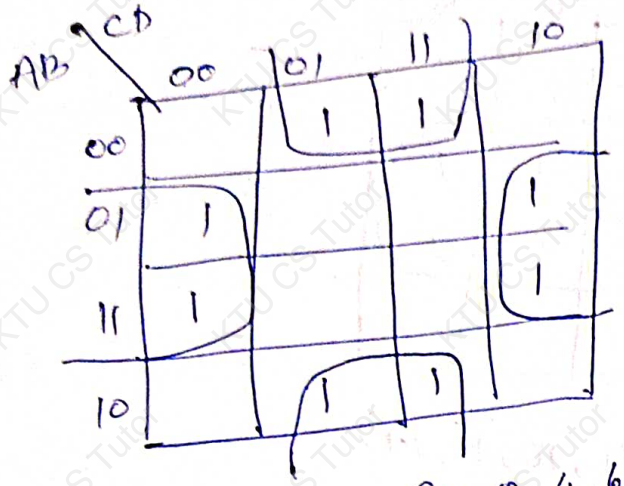
$f = \bar{A}\bar{C} + A\bar{B} + AC$

Qn. $f = \Sigma(0,2,5,7,8,10,13,15)$



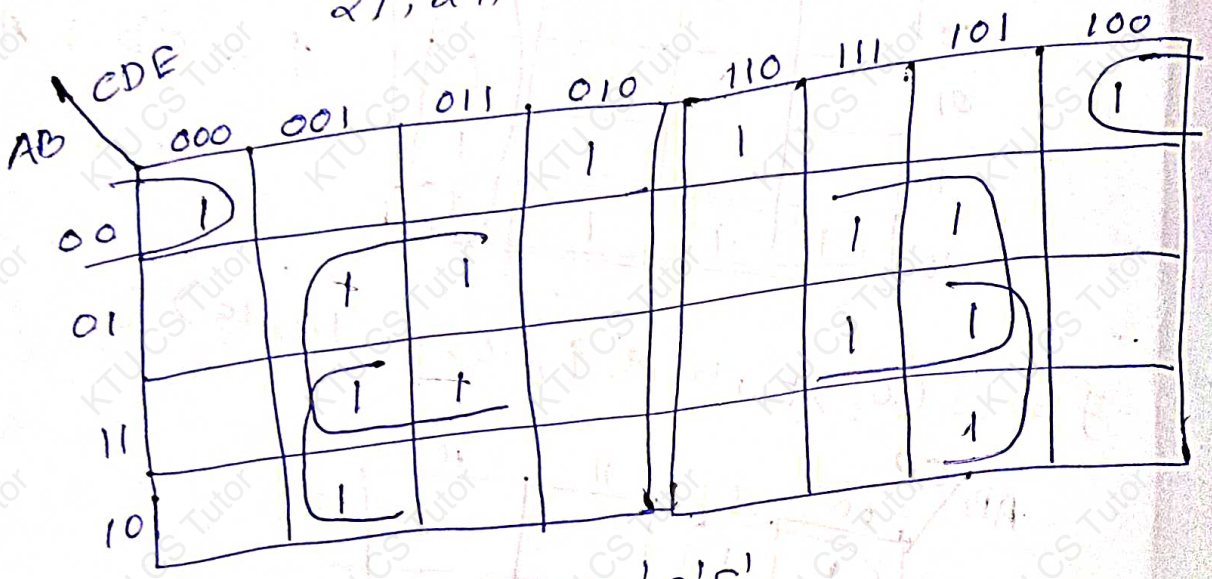
$f = BD + \bar{B}\bar{D}$

Qn. $f = \Sigma(1,3,4,6,9,11,12,14)$



$f = \bar{B}D + B\bar{D}$

Qn. $F(A,B,C,D,E) = \Sigma(0,2,4,6,9,11,13,15,17,21,25,27,29,31)$

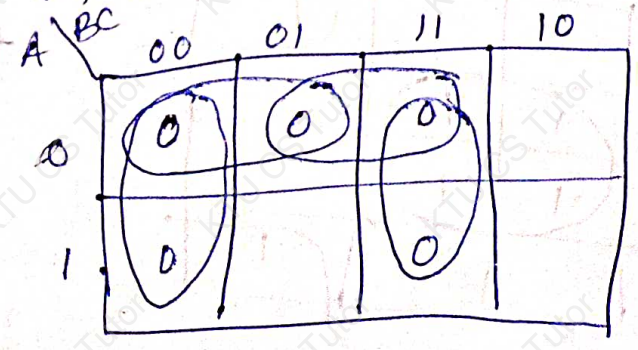


$f = BE + AD'E + A'B'E'$

Simplification of POS expressions

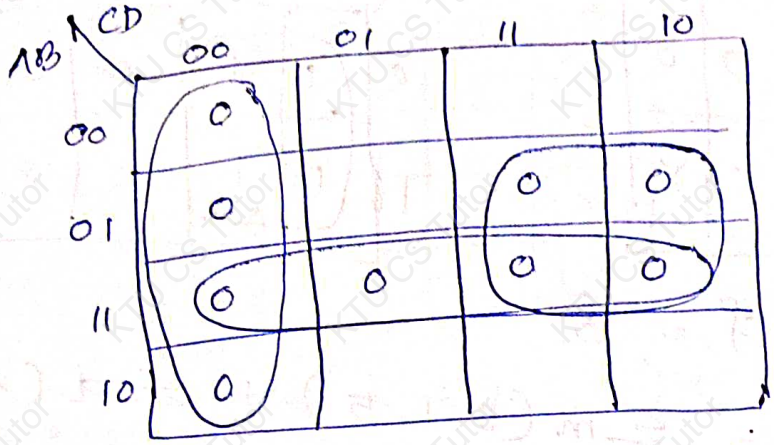
Minimization of POS is similar to SOP except plotting of 0's instead of 1's.

Qn. $f = \Pi(0,1,3,4,7)$



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

Qn $f = \Pi(0, 4, 6, 7, 8, 12, 13, 14, 15)$

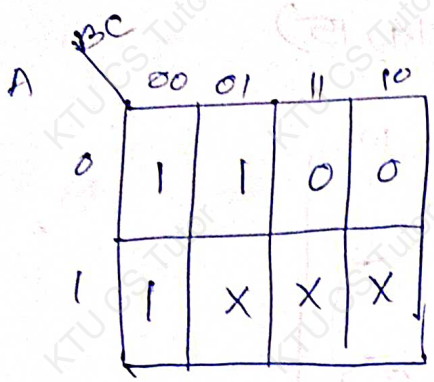


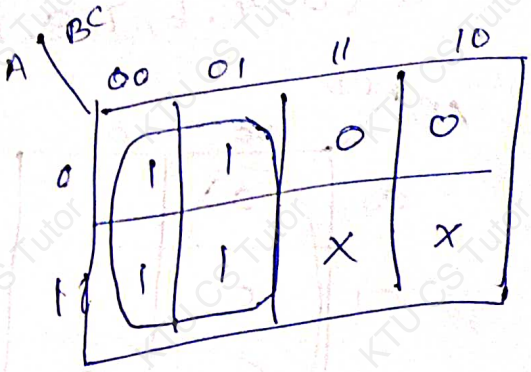
$f = (C + D)(\bar{A} + \bar{B})(\bar{B} + \bar{C})$.

Don't care conditions / Cannot Happen States

Some logic circuits can be designed so that there are certain input conditions for which there are no specified output levels. Or we "don't care". In such cases, the output level is not defined, it can be either HIGH or LOW. These output levels are indicated by 'X' or 'd' or 'ϕ' in the truth tables and are called "don't care outputs" and such combinations are called don't care conditions.

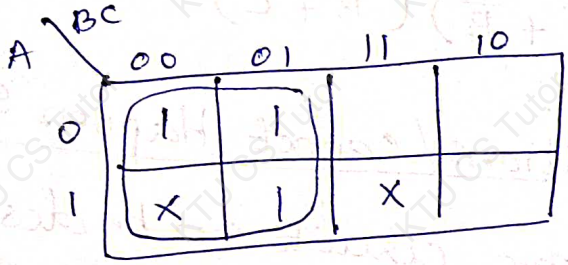
A	B	C	f
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	X
1	1	0	X
1	1	1	X





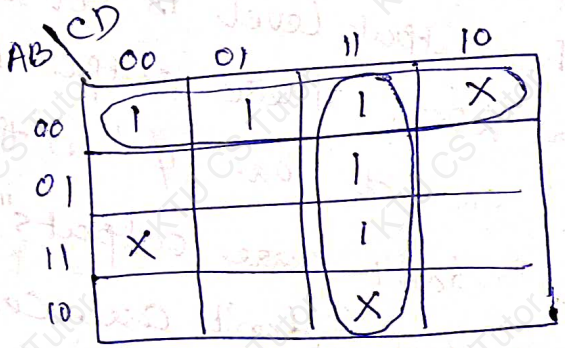
$f = \bar{B}$.

Qn. $f(A, B, C) = \Sigma m(0, 1, 5) + \Sigma d(4, 7)$



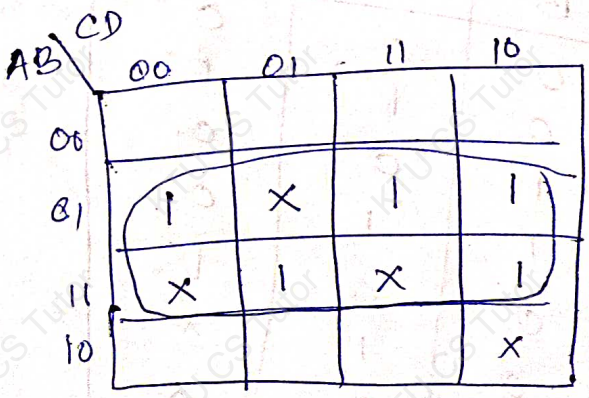
$f = \bar{B}$

Qn. $f(A, B, C, D) = \Sigma m(0, 1, 3, 7, 15) + \Sigma d(2, 11, 12)$



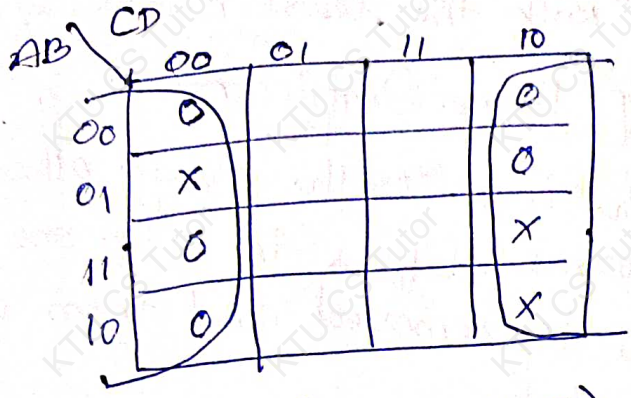
$f = \bar{A}\bar{B} + CD$.

Qn. $f(A, B, C, D) = \Sigma m(4, 6, 7, 13, 14) + \Sigma d(5, 10, 12, 15)$



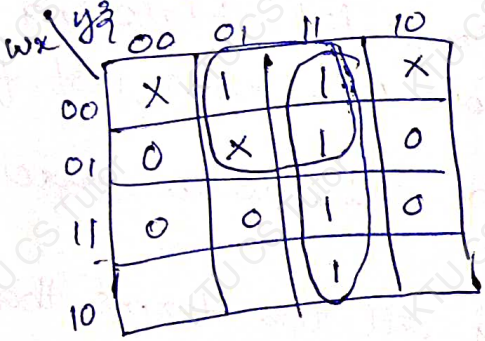
$f = B$.

Qn. $f(A,B,C,D) = \pi M(0,2,6,8,12) + \pi d(3,4,7,10,14)$



$f = \bar{D}$

Qn. $F(w,x,y,z) = \Sigma(1,3,7,11,15) + d(0,2,5)$



$F = w'z + yz$.

Tabulation Method (Quine - McCluskey Method)

The map method of simplification is convenient as long as the number of variables does not exceed five or six. Disadvantage of map is that it is essentially a trial and error procedure which relies on the ability of the human user to recognize certain patterns. For functions of six or more variables, it is difficult to be sure that the best selection has been made.

Tabulation method is a specific step by step procedure that is guaranteed to produce a simplified standard form expression for a function with any number of variables.

It consists of two parts.

① Find by an exhaustive search all the terms that are candidates for inclusion in the simplified function called prime implicants.

(2) Choose among the prime implicants those that give an expression with the least number of literals.

Determination of Prime Implicants

- Compare each minterm with every other minterm.
- If two minterms differs in only one variable, that variable is removed and term with one less literal is found.
- Repeat the process for every minterm until the exhaustive search is completed. Matching process cycle is repeated for those new terms just found.
- Process continued until the exhaustive search is completed.
- The remaining terms and all the terms that did not match during the process comprise the prime implicants.

Selection of Essential Prime Implicants

- Must select minimum number of PIs which cover all the minterms
- Prepare a prime implicant chart with the given minterms as columns and the PIs as rows.
- Put 'x' in each column corresponding to the minterms which covers the PIs.
- If any column contains just a single 'x', the PI corresponding to the row is essential PI and is included in the simplified expression.
- Once the EPI are obtained, check whether they are covering all the minterms or not. If they are covering, the resultant expression is unique

Qn. $f(A,B,C,D) = \Sigma m(0,2,3,6,7,8,10,12,13)$

Minterm	Binary representation A B C D	No. of 1's
m_0	0 0 0 0	0
m_2	0 0 1 0	1
m_3	0 0 1 1	2
m_6	0 1 1 0	2
m_7	0 1 1 1	3
m_8	1 0 0 0	1
m_{10}	1 0 1 0	2
m_{12}	1 1 0 0	2
m_{13}	1 1 0 1	3

Minterms	Index	Binary rep. A B C D
m_0	0	0 0 0 0 ✓
m_2	1	0 0 1 0 ✓
m_8	1	1 0 0 0 ✓
m_3	2	0 0 1 1 ✓
m_6	2	0 1 1 0 ✓
m_{10}	2	1 0 1 0 ✓
m_{12}	2	1 1 0 0 ✓
m_7	3	0 1 1 1 ✓
m_{13}	3	1 1 0 1 ✓

Minterms Group	Binary representation
0, 2	0 0 – 0 ✓
0, 8	– 0 0 0 ✓
2, 3	0 0 1 – ✓
2, 6	0 – 1 0 ✓
2, 10	– 0 1 0 ✓
8, 10	1 0 – 0 ✓
8, 12	1 – 0 0
3, 7	0 – 1 1 ✓
6, 7	0 1 1 – ✓
12, 13	1 1 0 –

0,2,8,10

2,3,6,7

Minterms	Binary representation
0, 2, 8, 10	$-0-0$
2, 3, 6, 7	$0-1-$

PI Table

Prime Implicants	Binary representation	Literal representation
8, 12	$1-00$	$A\bar{C}\bar{D}$
12, 13	$11O-$	$AB\bar{C}$
0, 2, 8, 10	$-0-0$	$\bar{B}\bar{D}$
2, 3, 6, 7	$0-1-$	$\bar{A}C$

Prime Implicants	m_0	m_2	m_3	m_6	m_7	m_8	m_{10}	m_{12}
$A\bar{C}\bar{D}$								X
$AB\bar{C}$								X
$\bar{B}\bar{D}$	X	X				X	X	
$\bar{A}C$		X	X	X	X			
	✓		✓	✓	✓		✓	✓

$$f = AB\bar{C} + \bar{B}\bar{D} + \bar{A}C$$

Qn. $f = \Sigma(0, 1, 2, 8, 10, 11, 14, 15)$

Minterm	Binary	No. of 1's
m_0	0000	0
m_1	0001	1
m_2	0010	1
m_8	1000	1
m_{10}	1010	2
m_{11}	1011	3
m_{14}	1110	3
m_{15}	1111	4

Minterm	Index	Binary
m_0	0	0000 ✓
m_1	1	0001 ✓
m_2	1	0010 ✓
m_8	1	1000 ✓
m_{10}	2	1010 ✓
m_{11}	3	1011 ✓
m_{14}	3	1110 ✓
m_{15}	4	1111 ✓

Minterms Group	Binary representation
0, 1	000 —
0, 2	00 — 0 ✓
0 8	— 0 0 0 ✓
2, 10	— 0 1 0 ✓
8, 10	1 0 — 0 ✓
10, 11	1 0 1 — ✓
10, 14	1 — 1 0 ✓
11, 15	1 — 1 1 ✓
14, 15	1 1 1 — ✓

Minterms	Binary representation
0, 2, 8, 10	− 0 − 0
0, 8, 2, 10	− 0 − 0
10, 11, 14, 15	1 − 1 −
10, 14, 11, 15	1 − 1 −

PI Table

Prime Implicants	Binary representation	Literal representation
0, 1	000 −	$w'x'y'$
0, 2, 8, 10	−0−0	$x'z'$
10, 14, 14, 15	1 − 1 −	wy

Prime Implicants	m_0	m_1	m_2	m_8	m_{10}	m_{11}	m_{14}	m_{15}
$w'x'y'$ (0,1)	x	x						
$x'z'$ (0,2,8,10)	x		x	x	x			
wy (10,11,14,15)					x	x	x	x
		✓	✓	✓		✓	✓	✓

Reduced function is

$f = w'x'y' + x'z' + wy.$

Qn. $f(A,B,C,D) = \Sigma m(0,1,2,5,7,8,9,10,13,15)$

Minterm	Binary	No. of 1's
m_0	0000	0
m_1	0001	1
m_2	0010	1
m_3	0101	2
m_7	0111	3
m_8	1000	1
m_9	1001	2
m_{10}	1010	2
m_{13}	1101	3
m_{15}	1111	4

Minterm	Index	Binary
m_0	0	0000 ✓
m_1	1	0001 ✓
m_2	1	0010 ✓
m_8	1	1000 ✓
m_5	2	0101 ✓
m_9	2	1001 ✓
m_{10}	2	1010 ✓
m_7	3	0111 ✓
m_{13}	3	1101 ✓
m_{15}	4	1111 ✓

Minterms	Binary representation
0, 1	000– ✓
0, 2	00–0 ✓
0, 8	–000
1, 5	0–01 ✓
1, 9	–001 ✓
2, 10	–010 ✓
8, 9	100– ✓
8, 10	10–0 ✓
5, 7	01–0
5, 13	–101 ✓
9, 13	1–01 ✓
7, 15	–111 ✓
13, 15	1–1–1 ✓

Minterms	Binary Representation
0,1,8,9	$-00-$
0,2,8,10	$-0-0$
0,8,2,10	$-0-0$
1,5,9,13	$--01$
1,9,5,13	$--01$
5,7,13,15	$-1-1$
5,13,7,15	$-1-1$

PI Table

Prime Implicants	Binary representation	Literal representation
0,1,8,9	$-00-$	$\bar{B}\bar{C}$
0,2,8,10	$-0-0$	$\bar{B}\bar{D}$
1,5,9,13	$--01$	$\bar{C}D$
5,7,13,15	$-1-1$	BD

PI Chart

PI	Literal	m_0	m_1	m_2	m_5	m_7	m_8	m_9	m_{10}	m_{13}	m_{15}
0,1,8,9	$\bar{B}\bar{C}$	x	x				x	x			
0,2,8,10	$\bar{B}\bar{D}$	x	x				x		x		
1,5,9,13	$\bar{C}D$		x		x			x		x	
5,7,13,15	BD				x	x				x	x
						✓		✓			✓