Logic functions

Basic properties of algebra.

→ Switching and boolean algebra both are same $\{0, 1\}$

Operators → $+$ (OR) $\cdot$ (AND) $\overline{\ }$ (NOT) [Also called as prime]

Idempotency.

$x \cdot x = x$

$x + x = x$

$x + 1 = 1$ $\quad$ $x + 0 = x$

$x \cdot 0 = 0$ $\quad$ $x \cdot 1 = x$,

Commutative

$x + y = y + x$

$xy = yx$

Associative

$x(yz) = (xy)z$

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

Complement

$\boxed{x + \bar{x} = 1}$ → Tautology

$\boxed{x \cdot \bar{x} = 0}$ → Always false

Distributivity

① $x(y + z) = xy + xz$

→ OR (+) and AND (·) are left and right both associative.

Imp ② $x + y \cdot z = (x + y) \cdot (x + z)$

↳ '·' is distributed over '+'.

Imp → If we have need to verify any equation then to do fast make truth table.

Dual

$\cdot \to +$, $\quad + \to \cdot$, $\quad 0 \to 1$, $\quad 1 \to 0$ → Each symbol have its dual correspondingly.

Imp → Expression and its dual both have same meaning.

→ We can find dual by exchanging the symbols

Ex $\quad x \cdot y = 1$

Dual → $x + y = 0$

Variable remains same in dual.

Switching expressions and simplification

Switching expression is a finite number of combinations of switching variables and constants $\{0, 1\}$ by means of switching operations $\{+, \cdot, \text{NOT}\}$.

Ex → ① $x + \bar{x}yz + x\bar{z}$ $\quad$ Literals = 6

② $a + bc + \bar{b}d$ $\quad$ Literals = 5

Properties for simplifying expression

Absorption: ① $x + xy = x$

$x(1 + y) = x$

or dual

$x \cdot (x + y) = x$

② $x + x'y = x + y$

$(x + x') \cdot (x + y) = 1 \cdot (x + y)$

$= x + y$

or dual

$x \cdot (x' + y) = x \cdot x' + x \cdot y$

$= xy$

Imp

Consensus Theorem

$xy + \bar{x}z + yz = xy + \bar{x}z$

(x prime)

$xy + \bar{x}z + yz(1)$

$\Rightarrow xy + \bar{x}z + yz(x+\bar{x})$

$\Rightarrow xy + \bar{x}z + xyz + yz\bar{x}$

$\Rightarrow xy(1+x) + \bar{x}z(1+y)$

$\Rightarrow \underline{xy + \bar{x}z}$.

<u>Ex</u> $x'y'z + yz + xz$

$\Rightarrow z(x+y+x'y')$

$\Rightarrow z((x'+y)(y'+y)+x)$

$\Rightarrow z(x'+y+x)$

$\Rightarrow z(1+y)$

$= \underline{z}$

→ In DLD (·) operation is commutative but in TOC it is not.

<u>Ex</u>: <u>DLD</u> → $a\cdot b = b\cdot a$

<u>TOC</u> → $a\cdot b \neq b\cdot a$

→ We can minimize using any method but answer should be same.

→ Cancellation is not allowed as,

$\cancel{a}+b = \cancel{a}+c$ ← Both are wrong

$\cancel{a}\cdot b = \cancel{a}\cdot c$

<u>DeMorgan's law & simplification</u>

It states that,

$\overline{xy} = (x'+y')$

$\overline{(x+y)} = (\bar{x}\cdot\bar{y})$

0 → 1 1 → 0 + → · · → +

Literals converts to (a,b) → Complement of literals (a',b')

<u>Ex</u>: $(a'+b') = a\cdot c$

Imp → <u>In duality we left the literal as it is. So, principle of duality and DeMorgan's law both are not same, or different.</u>

<u>Ex</u>: $\bar{x}+\bar{y} = (x\cdot y)$ DeMorgans

$\bar{x}+\bar{y} = (\bar{x}\cdot\bar{y})$ Duality

<u>Q</u> DeMorgan's law simplification

$(x+y)[x'(y'+z')]' + x'y' + x'z'$

$(x+y)[x+(y\cdot z)] + x'y' + x'z'$

$(x+y)[(x+y)(x+z)] + x'(y'+z')$

$(x+y)[(x+y)(x+z)] + x'(y'+z')$

$\Rightarrow x + xyz + xy + yz + x'y' + x'z'$

$\Rightarrow x(1+yz) + y(x+z) + x'(y'+z')$

$\Rightarrow x + xy + yz + x'y' + x'z'$

$\Rightarrow x + yz + x'y' + x'z'$

$\Rightarrow (x+x')\cdot(x+y') + yz + x'z'$

$\Rightarrow x + y' + yz + x'z'$

$\Rightarrow x + z' + y' + yz$

$\Rightarrow x + z' + y' + z$

$\Rightarrow x + y + 1 = \underline{1}$.

Switching functions

$f(a,b,c) = a + bc$

a, b, c are boolean variables then inputs can be

a	b	c	f(a,b,c)	f'(a,b,c)
0	0	0	0	1
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	1	0
1	0	1	1	0
1	1	0	1	0
1	1	1	1	0

a	b	f	g	f·g	f+g
0	0	0	1	0	1
0	1	1	0	0	1
1	0	0	0	0	0
1	1	1	1	1	1

f and g are assume to be any boolean function so that they give the corresponding outputs

Canonical form

$f = (011) + (100) + (101) + (110) + (111)$ can represent as canonical form.

~~f(3,4,5,6,7)~~, to give o/b 1s

$= \bar{a}bc + a\bar{b}\bar{c} + a\bar{b}c + ab\bar{c} + abc$

Canonical Sum of products

A product term which contains each of 'n' variables as factors either in complemented or uncomplemented form is called a min. term.

→ A min term gives the value '1' for exactly one combination of the variables.

→ The sum of all minterms of 'f' for which 'f' assumes '1' is canonical sum of products or 'disjunctive normal form.' Union

Ex→ Assume to give certain result

a	b	c	f
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

CSOP $= f(1,2,4,6)$

$= \bar{a}\bar{b}c + \bar{a}b\bar{c} + a\bar{b}\bar{c} + ab\bar{c}$

Every min. term contain all the variables in Canonical SOP.

SOP $= a + bc + ac$ → sum of product need not contain all the variables.

f (1,3,4,6) means if you assign any value from 1,3,4,6 (binary) to f^n then any particular min. term will show '1' and other show '0' but overall f^n will show 1.

So,

→ $\boxed{\Sigma(1,2,4,6) \text{ represent sum of all min. terms.}}$

Canonical Products of Sum

→ A sum term which contains each of 'n variables' as factors either in complemented or uncomplemented form is called max term.

→ A max term gives the value '0' for exactly one combination of the variable

→ The product of all max terms of 'f' for which 'f' assumes '0' is called canonical product of sums or 'conjunctive normal form.'

Ex

a	b	c	f
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Max terms → $(a+\bar{b}+c)$ ✓

$(a+c)$ ✗

$f = (a+b+c)\cdot(a+\bar{b}+\bar{c})\cdot(\bar{a}+b+c)\cdot(\bar{a}+\bar{b}+\bar{c})$

Each max terms is '0' for particular input of 0 and 1's.

→ Max terms have

imp

0 → true (No complement)

1 → false (Complement)

Min. terms have

0 → false (Complement)

1 → true (Not complement)

$\boxed{\pi(0,3,5,7)}$ is a representation.

Example on Canonial form

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

$f_1 = \Sigma(2,3,5,6,7)$

$= \pi(0,1,4)$

$f_2 = \Sigma(0,2,4,6)$

$= \pi(1,3,5,7)$

$f_3 = \Sigma(0,1,6,7)$

$= \pi(2,4,4,5)$

$f(x,y,z) = x'y + z' + xyz$

Convert the expression into either CSOP or CPOS

① Here we are converting into CSOP

② IInd way is convert or form a table from given function and find Σ and π

So,

$\Rightarrow x'y\cdot 1 + z'\cdot 1\cdot 1 + xyz$

$\Rightarrow x'y(z+z') + z'(x+x')(y+y') + xyz$

$\Rightarrow x'yz + x'yz' + z'(xy + xy' + x'y + y'x') + xyz$

$\Rightarrow \underset{(3)}{x'yz} + \underset{(2)}{x'yz'} + \underset{(6)}{xyz'} + \underset{(4)}{xy'z'} + \cancel{x'yz'} + \underset{(0)}{x'y'z'} + \underset{(7)}{xyz}$

repeat ← (2)

$= \Sigma(0,2,3,4,6,7)$

$= \pi(1,5)$

→ Delete the repeated value of term from expansion.

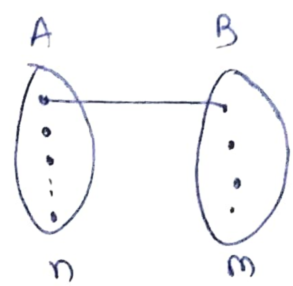
Functional Properties

The canonical sum of products or POS form of a boolean function is unique.

→ Two boolean functions $f_1(x_1 \dots x_n)$ and $f_2(x_1 \dots x_n)$ are said to be logically equivalent if and only if both function have same value for each and every combination of $(x_1, x_2 \dots x_n)$.

→ Two boolean functions are equivalent if their canonical pos or sop are identical.

Number of functions possible

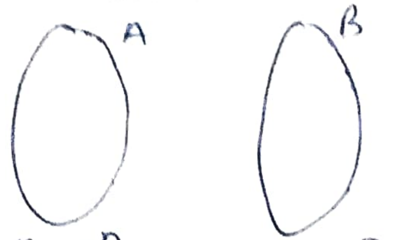


1 2 3 ... n
m m m ... m } can have

Total m^n functions are possible bcoz each value in A has m possibilities in B.

If n → boolean variable then no. of boolean fn possible —

1 2 3 ... n	f^n
2×2×2 ... 2	0 or 1
Total 2^n possibilities	



$n = 2^n$ $\quad$ $m = 2$

Total no. of boolean functions possible $= \boxed{2^{2^n}}$

If $n \to$ ternary variables then
No. of boolean function possibles $= 2^{3^n}$
bcoz each variable have 3 values $\{0, 1, 2\}$

So, if we have n k-ary variables the no. of boolean functions possible $= \boxed{2^{k^n}}$

If function is m-ary then
imp $\boxed{m^{k^n}}$

with 2 variables which are boolean, no. of binary function possible $= 2^{2^2} = 16$

a	b	f_0	f_1	f_2	f_3	-------	f_{15}
0	0	0	0	0	0	- -	1
0	1	0	0	0	0	- -	1
1	0	0	0	1	1	- -	1
1	1	0	1	0	1	- -	1

So, f can have 16 combinations

$f_0 \to 0000 \to 0$
$f_1 \to 0001 \to 1$
$f_2 \to 0010 \to 2$
|
$f_{15} \to 1111 \to 15$

0 – 16

Counting the number of functions and neutral functions

Q How many boolean functions are possible with 3 variables such that there are exactly 3 min terms.

Soln Assume variables are boolean

a	b	c	
0	0	0	
|			
1	1	1	

total $2^3 = 8$ combinations
Since it has 3 min terms means out of 8 only 3 holds the value '1'. So we can choose 3 from 8 in $^8C_3$ ways.

(ii) No. of function possible with atmost 3 min. terms & 3 boolean variables are given.

$^8C_0 + {^8C_1} + {^8C_2} + {^8C_3}$

→ If k-variables are given then 2^k rows possibles, if 'm' min terms are given then no. of function possible $= \boxed{^{2^k}C_m}$

Cont –

In neutral functions
(no. of min. term = no. of maxterm)

A	B	A	A'	B'	B	X-NOR	X-OR
0	0	0	1	1	0	1	0
0	1	0	1	0	1	0	1
1	0	1	0	1	0	0	1
1	1	1	0	0	1	1	0

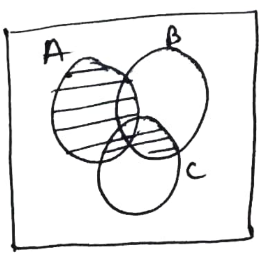
imp
for n-variables (boolean)
then no. of neutral f^n

$= \boxed{{}^{2^n}C_{2^{n-1}}}$

or $\boxed{{}^{2^n}C_{2^n/2}}$

Venn-Diagram Representation

Find the boolean function that the following Venn Diagram represents.



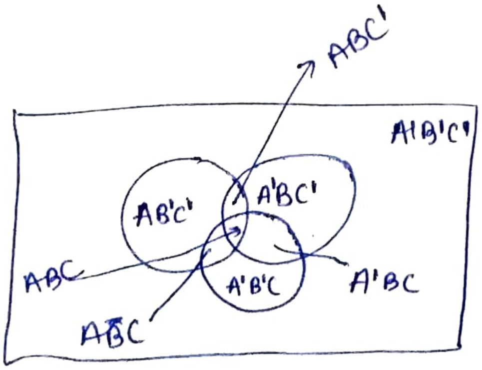
a) $A + A'B'C$

b) $A + BC$

c) $A + A'BC$

d) $AB + C$

Soln



Total 28 combinations are there

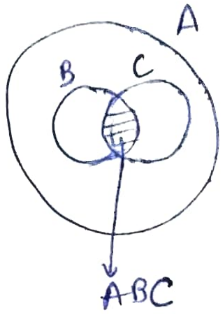
Ans
↳ $(A + A'BC)$ or $(A+A')\cdot(A+BC)$
$= (A+BC)$

→ So, (b) and (c) is the answer.

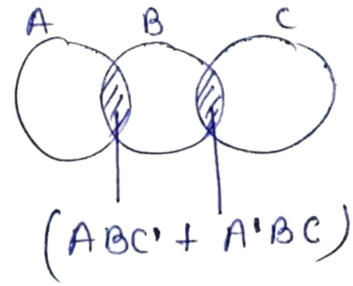
No. of f^n possible $= 2^{2^n}$

→ In 2^{2^n} ways we can shade this venn diagram.

→

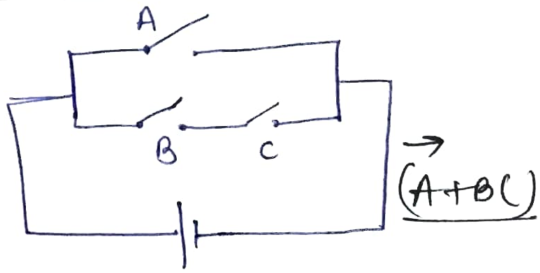


→



→ As above more diagram can ask as questions.

Contact Representation



→ Every boolean function can be represented with the help of serial and parallel contacts.

→ Serial contacts are performing "AND" operation.

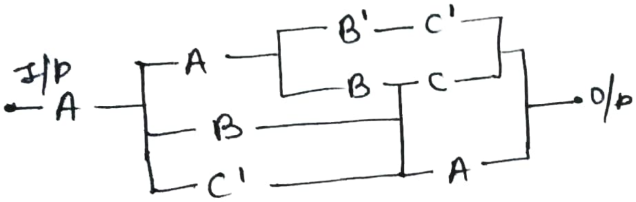
→ Parallel contacts are performing "OR" operation.

Q Which function does the following circuit represents?

$AB + A'B'$

Complement use for OFF side. But here either the switch touch 'ON' wire or 'OFF' wire the contact will occur and ckt get completed.

Q Identify the boolean expression



① Find the valid forward path.

forward path → Any path starting from i/p and ending at o/p without forming a cycle.

Imp Validity → No path should contain a variable in both true and complemented form.

$AAB'C' + AABC + ABC + AABA + ABA + AC'A + \underline{ACC'}$

Invalid path.

$\Rightarrow AB'C' + ABC + AB + ABC + AB + AC'$

$\Rightarrow AB'C' + ABC + AB + AC'$

$\Rightarrow AB'C' + AB + AC'$

$\Rightarrow \underline{AC' + AB}$

Nested function

In the following simultaneous boolean expressions, what are the values of x, w, y, z.

$x + y + z = 1$ — ①

$xy + w'z' = 0$ — ②

$xw' + yz' = 1$ — ③

(w) (x) (y) (z)

a) 0 0 0 1 × (Not satisfying 3)

b) 1 1 0 1 × (——— 3)

c) 0 1 0 1 ✓ (satisfying all 3)

d) 1 0 0 0

Soln: take each option one by one and put the value of w, x, y, z in each and every eqn.

→ find that option satisfying all the 3 equations.

Q $f(A,B) = A' + B$ then find $f(f(x+y, y), z)$?

$f(x+y, y) = (x+y)' + y$

$= x'y' + y$

$= (y + y')\cdot(y + x')$

$= \underline{x' + y}$

$f(x'+y, z) = (x'+y)' + z$

$= x \cdot y' + z$ Ans//

NAND Gate and Properties

A	B	A ↑ B
0	0	1
0	1	1
1	0	1
1	1	0

→ ↑ Symbol of NAND

Properties

① Identity: $A \uparrow A = A'$ not A

So, not identity.

② Commutative (Yes, commutative)

$A \uparrow B = B \uparrow A$

③ Associative (Not, associative)

$A \uparrow (B \uparrow C) \neq (A \uparrow B) \uparrow C$

NOR Gate and properties

A	B	A ↓ B
0	0	1
0	1	0
1	0	0
1	1	0

Properties

① Identity ? Not, identity

② Commutative ? Yes, commutative

③ Associative ? Not associative.

④ Idempotent ? No need here.

Ex-OR Gate and Properties

→ Also called modul 2 sum means (sum modulo 2).

A	B	A ⊕ B
0	0	0
0	1	1
1	0	1
1	1	0

$A \oplus B = \underbrace{\bar{A}B + A\bar{B}}_{\text{(min terms)}}$

Ex $\underbrace{1 \oplus 1 \oplus 1}_{3\%2} = 1$

$= 1$

Idempotency

$A \oplus A = 0$, not idempotent.

It might valid in

$A \oplus A \oplus A = A$ but

if one sequence is not verified means not idempodent.

Commutative

$A \oplus B = B \oplus A$

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

So, it is commutative.

Associativity

$A \oplus (B \oplus C) = (A \oplus B) \oplus C$

at $A=0$ we get

$B \oplus C = B \oplus C$

at $A=1$ we get

$(B \oplus C)' = B' \oplus C$

Both two seems different but they are same.

So, it is associative.

B	C	$(B\oplus C)'$	$B'\oplus C$
0	0	1	1
0	1	0	0
1	0	0	0
1	1	1	1

imp

So, $\boxed{\overline{(B\oplus C)} = (\overline{B}\oplus C) = (B\oplus\overline{C})}$

Ex-NOR Gate and Properties

A	B	$A\odot B$
0	0	1
0	1	0
1	0	0
1	1	1

$$A\odot B = \overline{A}\,\overline{B} + AB$$

Properties

- → Not idempotent.
- → It is commutative.
- → It is associative

and also,

imp

$\boxed{\overline{(B\odot C)} = (\overline{B}\odot C) = (B\odot\overline{C})}$

So, remember.

imp

$\overline{A\odot B} = A\oplus B = A'\odot B = A\odot B'$

$\overline{A\oplus B} = A\odot B = A'\oplus B + A\oplus B'$

→ There are two approaches to verify that given equation is correct or not.

① Create the truth table

② By putting the values

Properties of EX-OR and EX-NOR

XOR $\oplus$

A	B	$A\oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

X-NOR $\odot$

A	B	$A\odot B$
0	0	1
0	1	0
1	0	0
1	1	1

Tricks

$\oplus \rightarrow 1$, for odd no. of 1's

$\odot \rightarrow 1$ for even no. of 0's

A	B	C	$A\oplus B\oplus C$	$A\odot B\odot C$
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	0
1	0	0	1	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Here both $\odot$ and $\oplus$ have same value in o/p. Let's analyse say; n → no. of inputs, then

If n is even:

1's in i/p odd ⇒ 0's odd

o/p $\oplus = 1$ $\odot = 0$

1's are even ⇒ 0's are even

o/p $\oplus = 0$ $\odot = 1$

if n is odd:

1's is odd ⇒ 0's are even

odd+even → odd

o/p $\oplus = 1$ $\odot = 1$

1's are even ⇒ 0's are odd

$\oplus = 0$ $\odot = 0$

$\Rightarrow A \oplus B = \overline{A \odot B}$

$\Rightarrow A \oplus B \oplus C = A \odot B \odot C$

$\Rightarrow A \oplus B \oplus C \oplus D = \overline{A \odot B \odot C \odot D}$

Imp: In case of odd both are same and incase of even both are complements

K-Map for EX-OR or EX-NOR

$A \oplus B = A\bar{B} + B\bar{A}$

B \ A	0	1
0		1
1	1	

Complement

$A \odot B = AB + \bar{A}\bar{B}$

B \ A	0	1
0	1	
1		1

$A \oplus B \oplus C$

C \ AB	00	01	11	10
0		1		1
1	1		1	

Same

$A \odot B \odot C$

C \ AB	00	01	11	10
0		1		1
1	1		1	

→ And similarly we can draw for max no. of input variables.

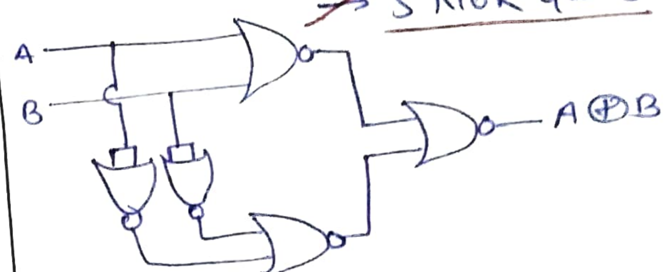
Note → Don't mug up anything. Try to relate it with concepts

Min. no. of gates required for EX-OR or EX-NOR

EX-OR

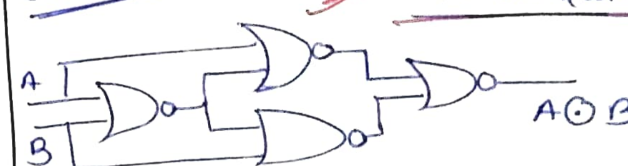
Using NAND → 4 NAND Gates

Using NOR → 5 NOR Gates

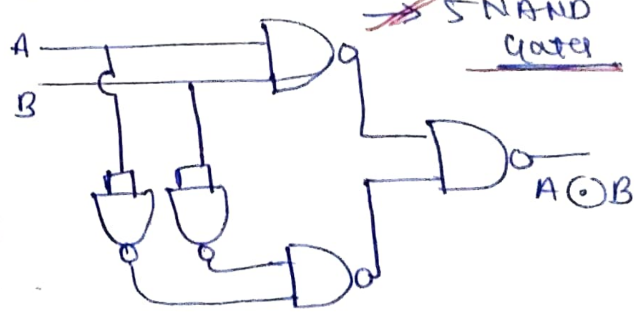


EX-NOR

Using NOR → 4 NOR Gates



Using NAND → 5 NAND Gates



→ NOR and NAND are Universal gates.

Remember

	NAND	NOR
X-OR	4	5
X-NOR	5	4

functionally Completeness

A set of operations is said to be functionally complete or universal if and only if every boolean function can be expressed by means of operations in it.

→ The set {+, ·, −} is clearly functionally complete. [Standard set]

→ The set {+, −} is said to be functionally complete.

→ The set {·, −} is also functionally complete.

Note → A set is said to be functionally complete if we can derive a set which are already functionally complete.

→ Using {+, −} we can derive (·) and using {·, −} we can derive (+).

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

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

→ So we have no need to carry all the 3 operations at a time.

Examples on functionally completeness

① $f(A, B, C) = A' + BC'$

To verify that the function is functionally complete or not we have need to find any one pair from (−, ·) or (−, +).

If we compute $f(A, A, A)$

$\Rightarrow f(A, A, A) = \bar{A} + A\bar{A}$

$= \bar{A}$

So we can find (−) of any variable.

and now have to find (·) or (+) any one from them,

So, we compute,

$f(f(A, A, A), B, f(B, B, B))$

$= [f(A, A, A)]' + B[f(B, B, B)]'$

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

$= \underline{A + B}$

So we can derive (+) too and using (−, +) we can (·).

So, given function is functionally complete.

② $f(A, B) = \bar{A} + B$

$f(A, A) = 1 = f(B, B)$.

Note → Since complement (−) is common in both (−, ·) and (−, +) so, we definitely first try to find complement.

→ To find complement replace all the variables with any of the input variables.

Here, we are not able to acheive complement using single variable.

→ So, we know that if in $f(A,B) = \bar{A}+B$ we can remove B then we get $\bar{A}$

So, we compute

$f(A,0) = \bar{A}+0 = \bar{A}$.

Here, we can say that 'f' is not uniquely sufficient to find or acheive complement we need (f, 0) together.

then

$f(f(A,0),B) = (A')'+B$
$= A+B$.

So, the function is functionally complete by taking support of 0.

So, if any function is functionally complete by taking the help of {0, 1} called as 'partially functionally complete.'

Note → Function

- Functionally Complete
- Partially functionally Complete
- Not functionally complete.

→ (f, 0) is functionally complete but f is partially functionally complete.

③ $f(A,B) = \bar{A}B$

$f(A,A) = f(B,B) = \bar{A}A$ or $\bar{B}B$
$= 0$.

$f(A,1) = \bar{A}$

$f(f(A,1),B) = (A')'B = AB$

So, this function is partially functionally complete with support of 1 and generation (—, ·).

④ $f(A,B,C) = AB+BC+CA$

$f(A,A,A) = A$
$f(B,B,B) = B$

→ In the expression there is no complement term

→ Complement can't derive it just can be extracted by making other as null.

→ In an expression if no complement term or variable is given we can't derive it.

→ Hence, given expression is not functionally complete.

⑤ $f(x,y) = \bar{x}y + x\bar{y}$ ⊕

$f(x,x) = 0$
$f(y,y) = 0$
$f(x,1) = \bar{x}$
$f(1,y) = \bar{y}$] complement (⊕, 1)

$(f, 1)$ is given as the complement.

$f(x', y) = xy + x'y'$

$f(x, y') = x'y' + xy$

$f(x', y') = xy' + x'y$

$f(x, x')$ or $f(y, y')$ are not allowed bcoz they will lead to f^n in one variable.

So no. $(+)$ and $(\cdot)$ is found.

→ So, given function is not functionally complete.

⑥ Any boolean function can be defined with which of the following operation.

a) $\oplus$, NOT ✗

b) $\oplus$, 1, OR ✓

c) $\oplus$, 1, NOT ✗

d) $\odot$, 1, NOT ✗

Soln

With $(+)$ and 1 we can get NOT$(^-)$

So, $(-, +)$ is sufficient

Ans → b)

⑦ $f(x, y) = xy + \bar{x}\bar{y}$ $[\odot]$

$f(x, x) = x + \bar{x} = 1$

$f(y, y) = y + \bar{y} = 1$

$f(x, 0) = x \cdot 0 + \bar{x} \cdot 1 = \bar{x}$

So, $(\odot, 0)$ is sufficient to generate $\bar{x}$ (complement).

$f(x', y) = \bar{x}y + x\bar{y}$

$f(x, y') = \bar{x}y + x\bar{y}$

$f(x', y') = x'y' + xy$

So, we are failed to get either $(+)$ or $(\cdot)$.

$$\boxed{\begin{aligned} x \oplus y &= x' \odot y = x \odot y' = x' \oplus y' \\ x \odot y &= x' \oplus y = x \oplus y' = x' \odot y' \end{aligned}}$$

↳ Imp as one mark.

Note → So, $\oplus$ and $\odot$ are not functionally complete, but $(\oplus, \cdot)$ $(\oplus, +)$ is functionally complete

Self Dual functions

$f(A, B, C) = AB + BC + CA$

Dual of f is fd,

$fd(A, B, C) = (A+B)(B+C)(C+A)$

$= AB + BC + CA$

$$\boxed{f = fd}$$

On converting fd into minterms

$= ABC(C + C') + BC(A + A')$

$+ AC(B + B')$

$= ABC + ABC' + A'BC + AB'C$

It is canonical sum of products

Mutually exclusive term for a term can be find as

$ABC \longrightarrow A'B'C'$] Mut. excl. terms

$AB'C \longrightarrow A'BC'$

A function (boolean) is self dual if:

i) It is neutral (no. of min. terms = no. of max. terms)

ii) The function doesn't contain two mutually exclusive terms.

Number of self Dual functions

Ex

	A	B	C
0)	0	0	0
1)	0	0	1
2)	0	1	0
3)	0	1	1
4)	1	0	0
5)	1	0	1
6)	1	1	0
7)	1	1	1

(0,7) (1,6) (2,5) (3,4) are pairs of mutually exclusive terms.

for a self dual function,
no. minterm = no. of max term

Here total 2^3 number → 4 minterm, 4 maxterm

and also mutually exclusive terms should not be present.

So, from a pair (a,b) we can choose any one as min terms or max terms or we can say that any terms from a or b can be chosen as min. or max terms.

So, for self dual,

$\Sigma(0,1,2,3)$ or $\Sigma(7,6,5,4)$

→ any combination can be chosen as min terms, other will become max. terms.

$f = 2 \times 2 \times 2 \times 2$ ... m

m → no. of order pairs

If no. of variables = n

no. of terms = 2^n

no. of pairs = $m = 2^{n-1}$

So,

$f = 2 \times 2 \times 2 \times \cdots \times 2^{n-1}$

So, no. of functions possible $= 2^{2^{n-1}}$

No. of neutral functions

$\boxed{{}^{2^n}C_{2^{n-1}}}$

→ Every self dual function should be a neutral but every neutral fn need not be self dual.

Note → If we devide all the terms in two equal parts of min. terms and max. terms then it will become neutral as

$\Sigma(0,7,5,4)$ or $\Sigma(1,2,3,6)$

→ It is neutral but there is also two mutually exclusive terms in a common pair.

Q Which of the following are self dual.

① $f(A,B,C) = \Sigma(0,2,3)$

② $f(A,B,C) = \Sigma(0,1,6,7)$

③ $f(A,B,C) = \Sigma(0,1,2,4)$

④ $f(A,B,C) = \Sigma(3,5,6,7)$

Sol^n Consider as pair as
(0,7) (1,6) (2,5) (3,4)
So, c) option is self dual function.
and option d) is also correct
→ It is observe that both c) and d) are complement to each other.
So, we can say that,
• "Self duality is closed under complementation".

Electronic Gates

① Electronic gates generally receive voltages as inputs and produces voltages as outputs

② The precise values of these voltages are not significant towards determination of logical operation of gates.

③ The significant point is that voltages are restricted to two ranges of values, high & low.

④ Thus two valued variables may be used to represent these voltages.

⑤ If we associate constant '1' with high voltage and '0' with low boltage, it is called as positive logic term.

⑥ If we associate constant '1' with low voltage and '0' with high voltage, called as (−ve) logic term.

Q Consider a gate which o/p high voltage only if all inputs are high and o/p low voltage otherwise. How does this gate behave in positive logic and negative logic.

→

A	B	A ⊙ B
H	H	H
H	L	L
L	H	L
L	L	L

Positive logic

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

AND

Negative logic

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

→ (OR)

→ (+ve) logic system and (−ve) logic system both are dual of each other.

Gate 2016

$x_1 \oplus x_2 \oplus x_3 \oplus x_4$

Sol^n find the combination for which equation is true and put the value in options.

Gate 2006 [Must do it]

Do it by self first and if require watch the video.

→ Hint for question:

Minimization

→ A switching f^n can usually be represent by using a number of expressions, but min. expression can be only one.

Ex→ $f(x,y,z) = xy + yz + zx$

$= xy'z + x'yz + xyz' + xyz$

→ After minimization functions meaning doesn't change.

Criteria to determine minimal cost:

① Minimum no. of appearance of literal.

② Minimum no. of literals in SOP or POS expression.

③ Minimum no. of terms in SOP expressions, provided there is no other such expression with the same no. of terms and fewer literals.

Irredundant and Irreducible expression

$f(x,y,z) = x'yz' + x'y'z' + xy'z' + x'yz + xyz + xy'z$

$\Rightarrow x'z'(y+y') + xy'[z+z'] + yz(x+x')$

$\Rightarrow \boxed{x'z' + xy' + yz}$

→ This expression contains no redundant term

→ This expression can't be reduce further.

→ An SOP expression from which no term or literal can be deleted without altering its logical value is called an irredundant and irreducible expression.

Note: An irredundant expression is not necessarily minimal, nor the minimal expression always unique.

K-Map Introduction [Do all the K-map Ques from book]

→ The algebraic procedure of combining various terms and applying to them the rules become very tedious as the no. of variables increases.

→ The map method provided a systematic method for combining terms and deriving minimal expression.

→ A K-map is a modified form of a truth table in which the arrangement of combinations is particularly convinient for minimization.

→ Every 'n' variable map consist of 2^n cells, representing all possible combinations of variables.

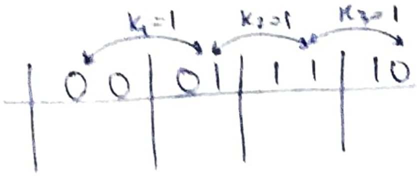
3-variable map

z \ xy	00	01	11	10
0	0	2	6	4
1	1	3	7	5

Why 11 comes before 10?

→ Each cell represents corresponding min. term.

Ex → $\bar{x}\bar{y}\bar{z}$ → 000 [0]
$\bar{x}yz$ → 011 [3]



•) $k_1 = 1-0 = 1$, $k_2 = 2-1 = 1$
$k_3 = 2-1 = 1$

•) k → difference in no. of 1's present in binary no. should be equals to 1.

→ The function value associated with a particular combination is entered in corresponding cells.

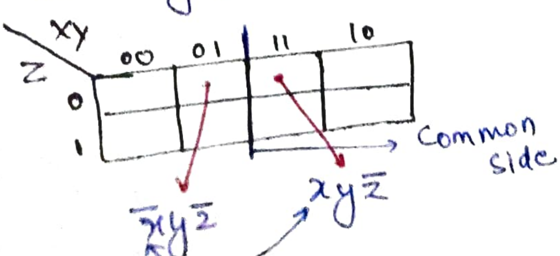
→ If $f = \Sigma(3,7)$ means only 3,7 cell have value 1 and other will have 0.

M.Imp.
→ Cycle code is used in the combination as column and row heading.

00 → 01 → 11 → 10

one bitchange in successive no.

→ Because of cycle codes two cells which have a common side correspond to combinations that differs by value of just a single variable.



→ These two cells play a major role in simplification process, bcoz they can be combined by means of rule,

$$Aa + A\bar{a} = A$$

K-map simplification

→ A collection of 2^m cells, each adjacent to m cells of collection is called a 'sub cube', and the sub cube is said to cover these cells.

→ Each subcube can be expressed by a product containing "n-m" literals, where n is no. of variables on which fn depends.

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

here, n = 4
but if m = 1
⇒ $2^m = 2$ so we choose 2 cells as $\bar{w}x\bar{y}z$ and $wx\bar{y}z$

if both will combine as

$$\underbrace{\bar{w}x\bar{y}z}_{4} + \underbrace{wx\bar{y}z}_{4} = \underbrace{x\bar{y}z}_{3}$$

So, here '$\underset{n-m}{(4-1)}$' terms is in final min. term.

Case 2 → $2^2 = 4$ cells then n-m have value = 2 then it will have 2 literals in final terms.

$$\Rightarrow \underbrace{\bar{w}x\bar{y}z + wx\bar{y}z}_{x\bar{y}z} + \left.\begin{matrix}\bar{w}xyz \\ + wxyz\end{matrix}\right] xyz$$

$$\Rightarrow xz(y+\bar{y}) = xz$$

⇒ xz have two literals

→ A function 'f' can be expressed as sum of those product terms which corresponds to subcubes necessary to cover all its '1' cells.

→ The number of product terms in the expression for 'f' is equal to the no. of subcubes, while the no. of literals in each term is determined by size of corresponding subcubes.

More the subcube size → lesser no. of literals in product term

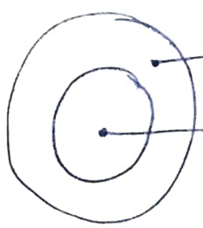
↓

lesser no. of product terms.

→ Therefore to obtain a minimal expression, we must cover all '1' cells with smallest no. of subcubes such that each subcube is as long as possible.

gap

$\boxed{M \subseteq I}$



Ex → $\Sigma(2,6,7,5) \rightarrow f$

z \ xy	00	01	11	10
0		1	1	
1			1	1

value of x is changing

subcubes of size 2

value of y is changing

$\boxed{f = xz + y\bar{z}}$

• $m = 1$
• $n = 3$ ⇒ $\underline{n - m = 2}$ size of product term

Examples (K-Map)

① Minimize the function.

$f(w,x,y,z) = \Sigma(0,4,6,7,8,9,14,15)$

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

$f = \bar{w}\bar{y}\bar{z} + w\bar{x}\bar{y} + xy$

→ write the literals that doesn't change

Q $f(w,x,y,z) = \Sigma(1,5,6,7,11,12,13,15)$

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

$f = \bar{w}\bar{y}z + wx\bar{y} + wyz + \bar{w}xy + \cancel{xz}$ → not required

→ In each subcell write the literals that is not changing its values between (0 and 1).

→ xz is not required bcoz cells required for xz are also covered in other four cells. So, xz is covered in all other product terms.

Covering functions.

Implicants:

A switching function $f(x_1, x_2, \ldots x_n)$ is said to cover $g(x_1, x_2, \ldots x_n)$ denoted by "f" is superset of 'g' if 'f' assumes true values where 'g' does.

Ex

A	B	f	g
0	0	1	0
0	1	1	1
1	0	0	0
1	1	0	0

$f \{00, 01\} \supseteq g \{01\}$

→ we can say that 'f' is covering 'g' bcoz min. terms involve in 'g' are also involve in 'f'.

Note: If 'f' and 'g' both covers each other at the same time. means both are equivalent.

A	B	g	f
0	0	0	0, 1
0	1	1	1 → यहाँ पर 1 रखना जरूरी है
1	0	0	0, 1
1	1	0	0, 1

No. of functions 'f' possible that can cover g is $(2^{2^n - x})$.

→ To cover 'g' a function 'f' must have min. terms involve in 'g' and all the other terms are optional.

→ If 'g' has 'x' min term and 'g' is a function of 'n' variables then no. of possible covering functions for g is equals to

Imp $\boxed{(2^{(2^n - x)})}$.

Ex: $f(w, x, y, z) = wx + yz$

w	x	y	z
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

These are the possible min term generate from wx.

In SOP each term contain each literal. but in wx given only 2 and since wx both are not complemented $(\bar{w}, \bar{x})$ so they both have value 1 in k-map So, with (11--) we can choose to fill the next 4 combinations to vacant places for yz.
So total 4 min terms can be generated from wx term.

Similar we can get 4 more terms from yz term.

So, total no. of min. term possible $= (8-1) = 7$

With four variable we have total term = 16.

→ we subtract 1 bcoz wxyz min. term is common in both.

So out of 16, 7 min. terms should be necessarily present in function.

and remaining 9 terms can be take any value from {0, 1}. So total no. of covering function = 2^9

Or. $2^{(2^4 - 7)} = 2^9$

→ If 'f' covers 'g' then 'g' is said to be implies 'f'. This is denoted by $g \to f$ means whenever 'g' is true 'f' is true, but we can't derive 'f' from 'g'.

→ So, here 'g' is implicant of 'f'.

Ex: $f(a,b,c) = ab + c$
ab: f_1 (say), c: f_2 (say)

a	b	c	f_1	f_2	$f = f_1 + f_2$
0	0	0	0	0	0
0	0	1	0	1	1
0	1	0	0	0	0
0	1	1	0	1	1
1	0	0	0	0	0
1	0	1	0	1	1
1	1	0	1	0	1
1	1	1	1	1	1

→ Here we can say that f_1 and f_2 are implicants of f.

→ So if a function of form SOP then each product term is implicant of the function.

If represent f_1 and f_2 in K-map we get;

c \ ab	00	01	11	10
0			1	
1	1	1	1	1

Every subcube is an implicant.

Prime Implicant:

Every subcube that can't be part of any other implicant called as prime implicant.

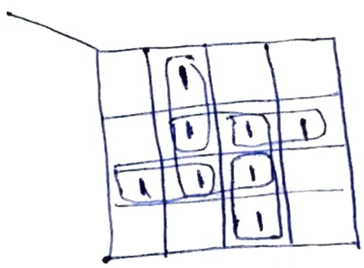
→ An implicant 'p' of a function 'f' is said to be the prime implicant if:

- P is a product term (i.e subcube).
- Deletion of any literal from 'p' results in a new product which is not covered by 'f' (i.e A subcube can't be part of any other subcube)

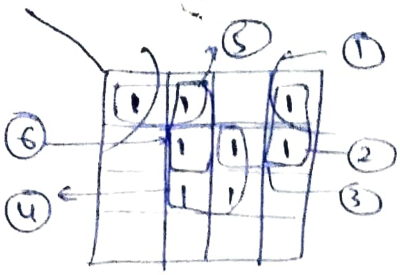
→ Subcube should be the longest subcube possible.

Essential Prime Implicant

A prime implicant 'p' of a function 'f' is said to be an essential prime implicant, if it covers at least one min term of 'f' which is not covered by any other prime implicants.



There are four Prime Implicant 4 of size 2 and one of size 4 but all four of size 2 are also essential prime implicant bcoz cube of size 4 contains min terms that can be derive other 4 too.



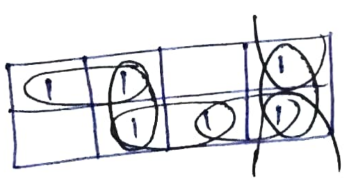
6 prime implicants only one (4) is essential prime implicant.

→ We can also combine two column element.

→ firstly find out all the prime implicants possible and then find or select the essential one that contain atleast one '1' that doesn't involve in any other prime implicant.

→ Sometimes it might also be possible that there is no essential prime implicant.

Ex:



Procedure for obtaining minimal SOP.

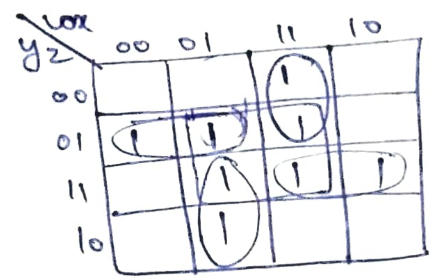
1) Determine all the essential prime implicants and include them in the minimal SOP.

2) Remove from list of prime implicants all those which are covered by essential prime implicants.

4) It is the set determined in step 1 covers all the minterms of 'f', then it is unique minimal expression. Otherwise select the additional prime implicants so that the function 'f' is covered completely and the total number and size of prime implicants added are minimal.

Ex:

$f(w,x,y,z) = \Sigma(1,5,6,7,11,12,13,15)$



Minimal expression = $w'y'z + wxy' + wyz + w'xy$

→ It is unique minimal expression.

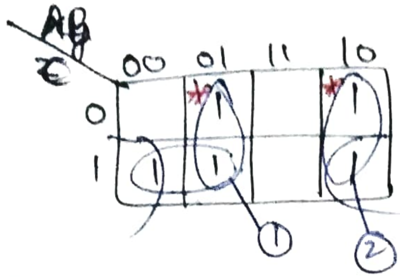
Prime Implicant chart:

Min terms	1	5	6	7	11	12	13	15
✓ $w'y'z$	1	1						
✓ wxy'						1	1	
✓ wyz					1			1
✓ $w'xy$			1	1				
✗ xz → rejected		1		1			1	1

→ Here we filled all the entries according to k-map. If product term is generating any min. terms then we fill the entry.

→ find the essential product terms and involve it in answer.

Ex: Q Which of the following prime Inplicant are essential



$\rightarrow A'B + AB' + A'C$

a) $B'C, A'B$

b) $A'C, A'B$

c) $A'B, AB'$ Ans

d) $A'B, AB', B'C$

→ ① and ② containing two 1's that doesn't involve in any other prime implicant. marked as *.

→ No. of min. prime implicants require = 3 (here)

as,

$AB' + A'B + A'C$ and $AB' + A'B + B'C$ — Both are possible

essential prime Implecants

→ So, here two minimal exp. are possible.

Prime Implecant chart

	1	2	3	4	5
AB'				1	1
$A'B$		1	1		
$A'C$	1		X1		
$B'C$	1				X1

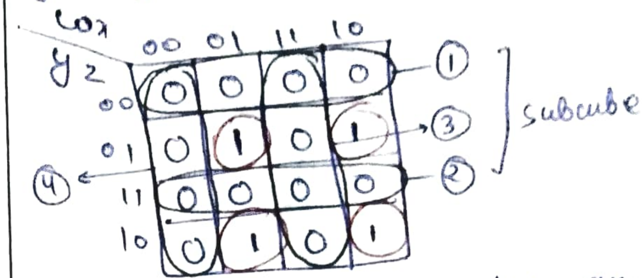
→ choose any one from 9to

Answer will contain AB' and $A'B$ both definitely.

→ To cover 1 in min. term we can choose anyone.

Minimal POS [0 → No complent, 1 → complement]

$f(x, y, z, w) = \Sigma(5, 6, 9, 10)$



In POS, we consider the max. terms (i.e '0').

$POS = (y+z)\cdot(\bar{y}+\bar{z})\cdot(w+x)\cdot(\bar{w}+\bar{x})$

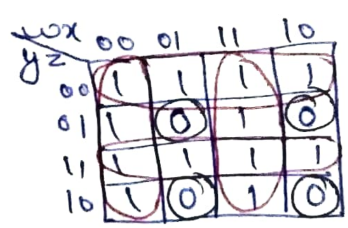
All the prime implicants are essential here, So it's the minimal expression.

In SOP, we get four 1's and all the four are essential prime implecent

So,

$SOP = \bar{w}x\bar{y}z + w\bar{x}\bar{y}z + \bar{w}xy\bar{z} + w\bar{x}y\bar{z}$

$f'(w, x, y, z) = \Pi(5, 6, 9, 10)$



$SOP = \bar{w}\bar{x} + \bar{y}\bar{z} + yz + wx$

$POS = (\bar{w}+x+\bar{y}+z)\cdot(w+\bar{x}+\bar{y}+z)\cdot(\bar{w}+x+y+\bar{z})\cdot(w+\bar{x}+y+\bar{z})$

→ Both POS and SOP are good conditionally.

Ex: Find the number of literals in minimum POS and SOP for

$$f(w,x,y,z) = \pi(1,5,6,7,11,12,13,15)$$

Sol^n

yz \ wx	00	01	11	10
00	1	1	0	1
01	0	0	0	1
11	1	0	0	0
10	1	0	1	1

→ $\pi()$ means max. terms are given

$$POS = (\bar{w}+\bar{x}+y)\cdot(w+y+\bar{z})\cdot(\bar{w}+\bar{y}+\bar{z})\cdot(w+\bar{x}+\bar{y})$$

$$SOP = \bar{w}\bar{y}\bar{z} + w\bar{x}\bar{y} + \bar{w}\bar{x}y + wy\bar{z}$$

So, here,

$$POS = (SOP)^c$$

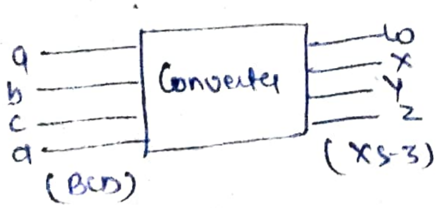
(or)

$$SOP = (POS)^c$$

No. of literals in both POS & SOP are 12.

Introduction to Don't Cares

Consider a scenario that we want to convert a BCD (binary coded decimal) to XS-3 number



Since, BCD have range (0–9) means 10 number so we need to represent it using bits as,

	a	b	c	d	w	x	y	z
0	0	0	0	0	0	0	1	1
1	0	0	0	1	0	1	0	0
2	0	0	1	0	0	1	0	1
3	0	0	1	1	0	1	1	0
4	0	1	0	0	0	1	1	1
5	0	1	0	1	1	0	0	0
6	0	1	1	0	1	0	0	1
7	0	1	1	1	1	0	1	0
8	1	0	0	0	1	0	1	1
9	1	0	0	1	1	1	0	0

→ Not a completely specified table due to absence of (10–15)

$\boxed{XS3 = BCD + 3}$

'w' have min term as (5,6,7,8,9)

$$\Rightarrow w = \Sigma(5,6,7,8,9) + \phi(10,11,12,13,14,15)$$

Since, BCD have range upto 9 but using 4 bits we can generate upto 15. So all the terms from (10–15) are not needed to take care bcoz it doesn't matter.

So, $\phi(x)$ is called as don't care function.

Don't Care Combinations

→ A function is said to be completely specified if it is given 0 or 1 for every combination of input variables.

There exists some functions which are not completely specified.

→ Combinations for which the value of a function is not specified are called don't care combinations.

→ Since each don't care combination represents two values {0, 1} an incompletely specified functions containing k-don't care

Combinations corresponds to a class of 2^k distinct functions.

Ex :

ab	f	1	2	3	4
00	1	1	1	1	1
01	1	1	1	1	1
10	ϕ	0	0	1	1
11	ϕ	0	1	0	1

So, we are free to choose any combination of $\{0,1\}$ for ϕ, bcoz it doesn't matter what we choose. So we can choose according to our convinience.

But which one will you choose from (1,2,3,4). Even all four can considered but we need to choose the best combination, with minimal representation.

We will go for 4th one bcoz it contains 1 1 and if 1 1 include in k-map it will get included in any subcube and subcube size get increases

→ More the size of subcube lesser or minimal the min. term.

→ There should be no subcube in k-map that contains only don't care, if it exist just ignore it.

Examples

$w = \Sigma(5,6,7,8,9) + \phi(10,11,12,13,14,15)$

→ Don't care can be used as either $\{0,1\}$

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

without using don't care we get

$w = w\bar{x}\bar{y} + \bar{w}xy + \bar{w}xz$

If assume $\phi = 1$ then we can form subcubes of larger size. After including don't care we get

$w = xy + xz + w$

So, after including don't care we get more reduced expression and we know that including don't cares doesn't effect anything.

No. of function possible $= 2^k$

$k \rightarrow$ no. of min. terms present in $\phi(x)$

here, $k = 6$ (10−15)

So, $2^6 = 64$ fn possible.

And we can choose the minimal of 64 using k-map.

Ex → 2.1

AB \ CD	00	01	11	10
00	0	0	1	0
01	X	X	1	X
11	0	1	1	0
10	0	1	1	0