Lecture 8 :

Minimization using K-Map

Don't Care Combinations

Prime Implicants


GO CLASSES
GATE OVERFLOW
GO Test Series
is now
GATE Overflow + GO Classes
2-IN-1 TEST SERIES
Most Awaited
GO Test Series
is Here
REGISTER NOW
http://tests.gatecse.in/
100+ More than 100 Quality Tests.
15 Mock Tests.
FROM
14th April
+91 - 6302536274
+91 9499453136


GO CLASSES
GATE OVERFLOW
LIVE
GATE 2023
Live + Recorded Lectures
Daily Home Work + Solution
Watch Any Time + Any Number of Times
Summary Lectures For Every Topic
Practice Sets From Standard Resources
Enroll Now
+91 - 6302536274
www.goclasses.in
linkedin.com/company/go-classes
instagram.com/goclasses_cs

Join GO+ GO Classes Combined Test Series for BEST quality tests, matching GATE CSE Level:

Visit www.gateoverflow.in website to join Test Series.

1. **Quality Questions:** No Ambiguity in Questions, All Well-framed questions.
2. Correct, **Detailed Explanation**, Covering Variations of questions.
3. **Video Solutions.**

NOTE :

Complete Discrete Mathematics & C-Programming Courses,

by GO Classes, are FREE for ALL learners.

Visit here to watch : https://www.goclasses.in/s/store/

SignUp/Login on Goclasses website for free and start learning.

Join GO Classes Resources, Notes, Content, information Telegram Channel:

Public Username: goclasses_cse

Join GO Classes **Doubt Discussion** Telegram Group :

Username: GATECSE_Goclasses

(Any doubt related to Goclasses Courses can also be asked here.)

Join GATEOverflow Doubt Discussion Telegram Group :

Username: gateoverflow_cse

Recap :

Minimization using K-Map

Minimization by Karnaugh Maps

- What is a Karnaugh map?
  - 3 Variable Example:

A\BC	00	01	11	10
0				
1				

  - A grid of squares
  - Each square represents one minterm
    - eg: top-left represents $\overline{A}.\overline{B}.\overline{C}$, bottom-right represents $A.B.\overline{C}$
  - The minterms are ordered according to Gray code
    - only one variable changes between adjacent squares
  - Squares on edges are considered adjacent to squares on opposite edges
  - Karnaugh maps become clumsier to use with more than 4 variables

– 4 Variable example

AB\CD	00	01	11	10
00				
01		?	??	
11				
10				

– The square marked ? represents [redacted]

– The square marked ?? represents [redacted]

– Note that they differ in only the C variable.

- 4 Variable example

C

AB\CD	00	01	11	10
00				
01		?	??	
11				
10				

A

B

D

- The square marked ? represents $\overline{A}.B.\overline{C}.D$

- The square marked ?? represents $\overline{A}.B.C.D$

- Note that they differ in only the C variable.

Filling out a Karnaugh Map

- Write the Boolean expression in SOP form
- For each product term, write a 1 in all the squares which are included in the term, 0 elsewhere
  - canonical form: one square
  - one term missing: two adjacent squares
  - two terms missing: 4 adjacent squares
- Eg:

$$X = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$$

A\BC	00	01	11	10
0	0	0	1	0
1	0	1	1	1

Karnaugh Maps

- Decimal Representation

$f(x, y, z)$

$x \backslash yz$	00	01	11	10
0	0	1	3	2
1	4	5	7	6

Karnaugh Maps

f(wxyz)

- Decimal Representation

wx \ yz	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

Examples of Subcubes

Subcubes for elimination of one variable

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

Variables in the product term are variables whose value is constant inside the subcube.

Product term: $\overline{w}xz$

Subcubes for elimination of one variable

yz

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

Product term: $wx\overline{z}$

Subcubes for elimination of one variable

yz

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

Product term: xyz

Subcubes for elimination of one variable

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

Product term: $\overline{x}\,\overline{y}\,\overline{z}$

Subcubes for elimination of two variables

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

Product term: $x\overline{y}$

Subcubes for elimination of two variables

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

Product term: yz

Subcubes for elimination of two variables

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

Product term: $\overline{w}\,\overline{x}$

Subcubes for elimination of two variables

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

Product term: $w\overline{z}$

Subcubes for elimination of two variables

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

Product term: $\overline{x}\,\overline{z}$

GO Classes

Subcubes for elimination of three variables

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

Product term: $\overline{w}$

GO Classes

Subcubes for elimination of three variables

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

Product term: $\overline{w}$

Subcubes for elimination of three variables

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

Product term: $\overline{x}$

Subcubes for elimination of three variables

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

Product term: $\overline{z}$

Subcubes for sum terms

$wx \backslash yz$	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11			0	0
10	0	0	0	0

Sum terms: $w + \overline{x} + y$

$x + y$

$\overline{y}$

More Examples of grouping

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

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

$$AB + \overline{C}D$$

$$B\overline{D} + ABC$$

More examples

	$\overline{C}$	C
$\overline{A}\overline{B}$	0	0
$\overline{A}B$	1	0
AB	1	0
$A\overline{B}$	0	0

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

(a)

	$\overline{C}$	C
$\overline{A}\overline{B}$	0	0
$\overline{A}B$	1	1
AB	0	0
$A\overline{B}$	0	0

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

(b)

	$\overline{C}$	C
$\overline{A}\overline{B}$	1	0
$\overline{A}B$	0	0
AB	0	0
$A\overline{B}$	1	0

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

(c)

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

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

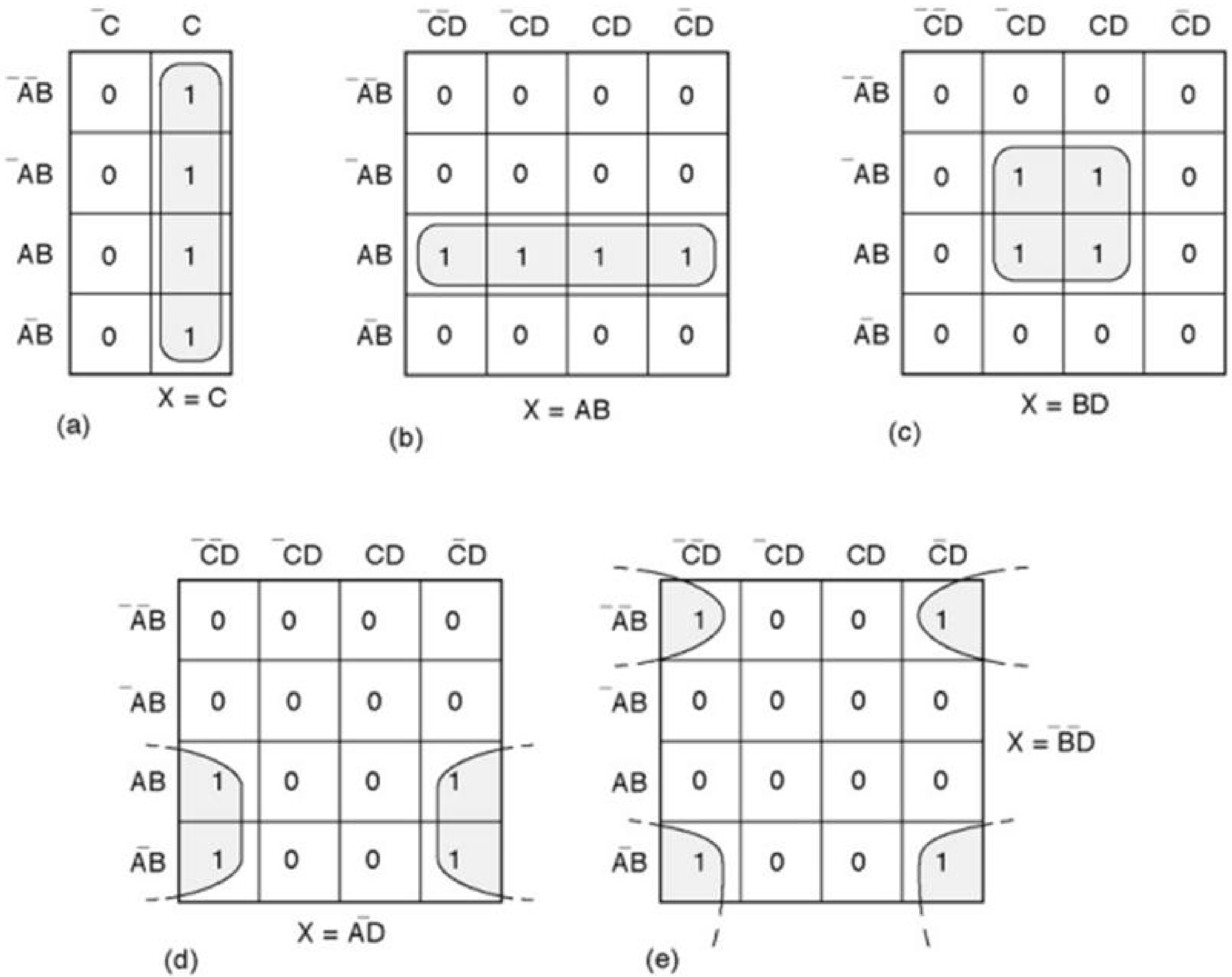
$$X = \overline{A}\overline{B}CD + \overline{A}\overline{B}C\overline{D} + A\overline{B}\overline{C}\overline{D} + A\overline{B}C\overline{D} = \overline{A}\overline{B}C + A\overline{B}\overline{D}$$

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

(d)

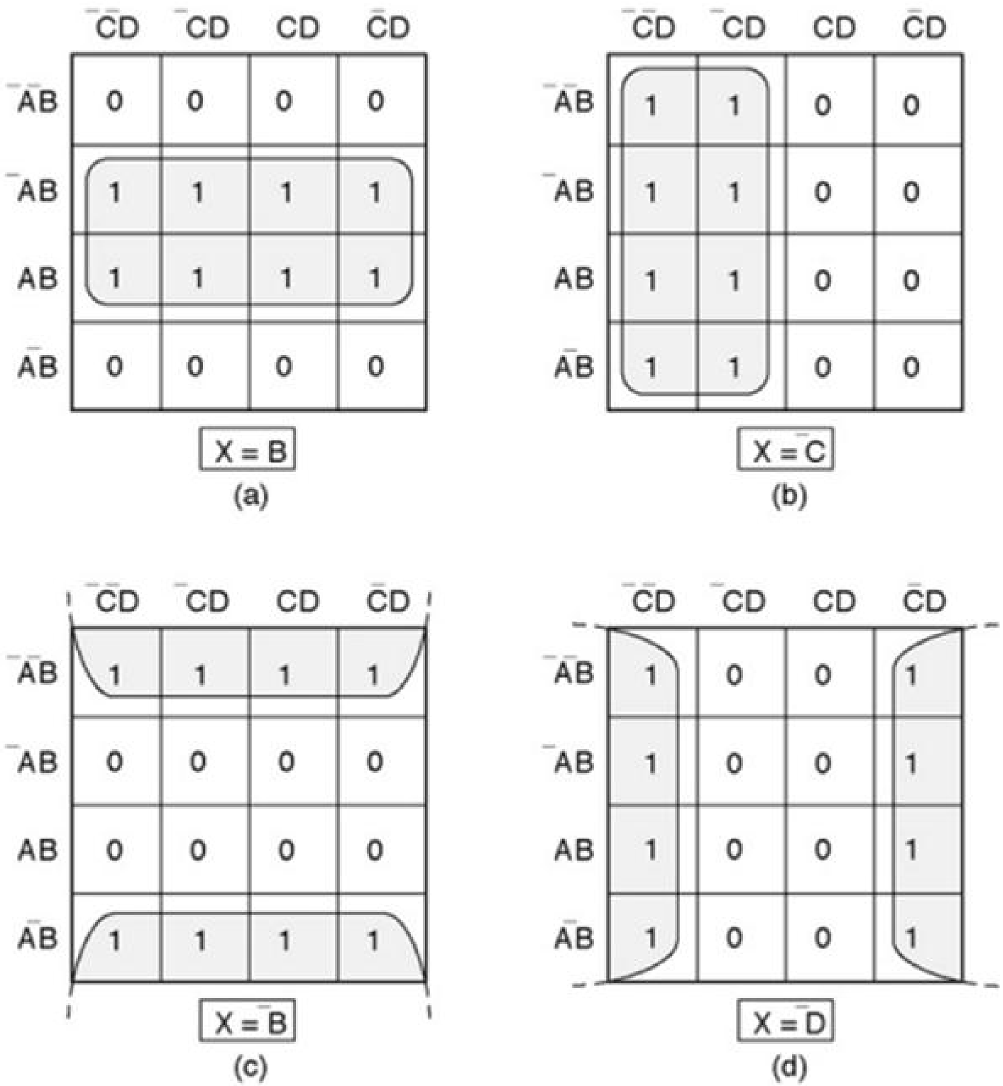
GO CLASSES

Imperial College London



O Classes

Imperial College
London



Complete Simplification Process

1. Construct the K map and place 1s and 0s in the squares according to the truth table.
2. Group the isolated 1s which are *not* adjacent to any other 1s. (single loops)
3. Group any pair which contains a 1 adjacent to only one other 1. (double loops)
4. Group any octet even if it contains one or more 1s that have already been grouped.
5. Group any quad that contains one or more 1s that have not already been grouped, *making sure to use the minimum number of groups.*
6. Group any pairs necessary to include any 1s that have not yet been grouped, *making sure to use the minimum number of groups.*
7. Form the OR sum of all the terms generated by each group.

	$\bar C\bar D$	$\bar C D$	CD	$C\bar D$
$\bar A\bar B$	0 (1)	0 (2)	0 (3)	1 (4)
$\bar A B$	0 (5)	1 (6)	1 (7)	0 (8)
AB	0 (9)	1 (10)	1 (11)	0 (12)
$A\bar B$	0 (13)	0 (14)	1 (15)	0 (16)

$$X = \underbrace{\bar A\bar B C\bar D}_{\text{loop 4}} + \underbrace{ACD}_{\text{loop 11, 15}} + \underbrace{BD}_{\text{loop 6, 7, 10, 11}}$$

(a)

	$\bar C\bar D$	$\bar C D$	CD	$C\bar D$
$\bar A\bar B$	0 (1)	0 (2)	1 (3)	0 (4)
$\bar A B$	1 (5)	1 (6)	1 (7)	1 (8)
AB	1 (9)	1 (10)	0 (11)	0 (12)
$A\bar B$	0 (13)	0 (14)	0 (15)	0 (16)

$$X = \underbrace{\bar A B}_{\text{loop 5, 6, 7, 8}} + \underbrace{B\bar C}_{\text{loop 5, 6, 9, 10}} + \underbrace{\bar A C D}_{\text{loop 3,7}}$$

(b)

	$\bar C\bar D$	$\bar C D$	CD	$C\bar D$
$\bar A\bar B$	0 (1)	1 (2)	0 (3)	0 (4)
$\bar A B$	0 (5)	1 (6)	1 (7)	1 (8)
AB	1 (9)	1 (10)	1 (11)	0 (12)
$A\bar B$	0 (13)	0 (14)	1 (15)	0 (16)

$$X = \underbrace{AB\bar C}_{9,\ 10} + \underbrace{\bar A\bar C D}_{2,\ 6} + \underbrace{\bar A B C}_{7,\ 8} + \underbrace{ACD}_{11,\ 15}$$

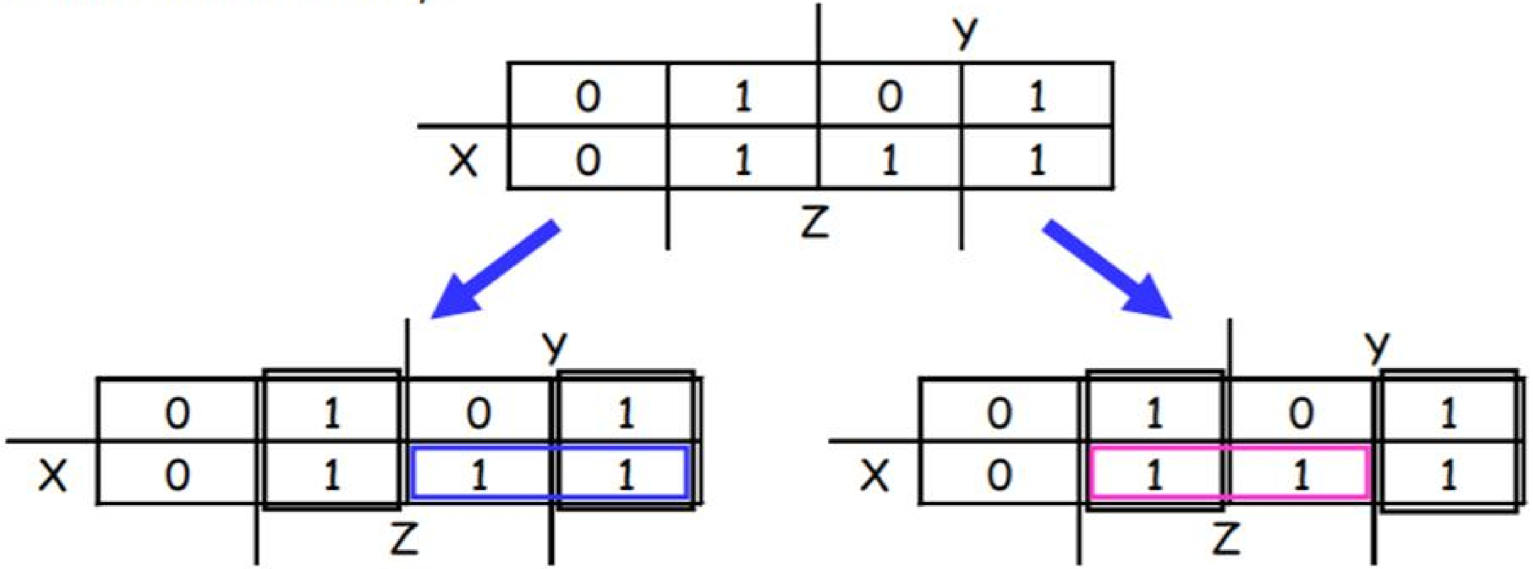
(c)

NOTE :

Minimum SOP or Minimum POS Expression **may NOT be Unique.**

K-maps can be tricky!

- There may not necessarily be a *unique* MSP. The K-map below yields two valid and equivalent MSPs, because there are two possible ways to include minterm m_7



$y'z + yz' + xy$

$y'z + yz' + xz$

- Remember that overlapping groups is possible, as shown above

Q: For a function F, if mSOP is Not unique then minimum POS is also Not unique?

Q: For a function F, if mSOP is Not unique then minimum POS is also Not unique? No

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

mSOP:

$\underbrace{2 \text{ mSOPs}}$

mSOP$_1$: $\bar{x}y + zy + x\bar{y}$

#literals = 6

x

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

mSOP:

2 mSOPs

mSOP_2 : $\bar{x}y + zx + x\bar{y}$

literals = 6

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

$\bar{x}$ $\bar{y}$ $\bar{z}$

mPOS :

unique mPOS

$$mPOS = (x+y)(\bar{x}+\bar{y}+z)$$

#literals = 5

Q: For a function F, if we have K mSOPs then we have K minimum POSs?

Q: For a function F, if we have K mSOPs then we have K minimum POSs? No

NoT Necessarily

z \ xy				
	0	1	0	1
	0	1	1	1

f: mSOPs = 2, mPOS = 1

Q: For a function F, if mSOP has K literals then minimum POS also have K literals?

Q: For a function F, if mSOP has K literals then minimum POS also have K literals? No

z \ xy				
	0	1	0	1
	0	1	1	1

Not Necessarily

function $f(a, b, c)$

m Sop form

mpos form

NOTE :

K-map can be used to find any SOP(or POS) expression, Minimized or Canonical or any other.

Using k-map:

minimized SoP

Every cube as large as possible.

Using k-map:

Canonical SOP

Every cube is Size 1.

Every cell is a cube.

Karnaugh Maps and Canonical Formulas

- Minterm Canonical Formula $\equiv$ Canonical SOP

$\bar{x}\bar{y}\bar{z}$

$y\bar{z}\bar{x}$

yz

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

$x\bar{y}\bar{z}$

$x\bar{y}z$

$$f(x) = \bar{x}\,\bar{y}\,\bar{z} + \bar{x}y\bar{z} + x\bar{y}\,\bar{z} + x\bar{y}z$$
$$= \sum m(0,2,4,5)$$

Karnaugh Maps and Canonical Formulas

- Maxterm Canonical Formula = Canonical POS

$(y+\overline{z}+x)$

$(\overline{y}+z+\overline{x})$

$x \backslash yz$	00	01	11	10
0	1	0	0	1
1	1	1	0	0

$$f(x) = (x + y + \overline{z})(x + \overline{y} + \overline{z})(\overline{x}\ \overline{y}\ \overline{z})(\overline{x}\ \overline{y} z)$$
$$= \Pi M(1,3,6,7)$$

Next Topic:

Functions with

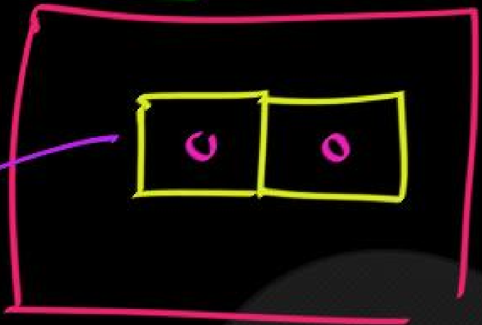
Don't Care Combinations

(Incompletely specified functions)

1. For a Boolean function/digital circuit;

Sometimes Some Input Combinations can NEVER occur: for these Input Combinations, we don't care about the function value.

Eg: Bank:



Joint locker L } $\underbrace{A, B}_{\text{Two people}}$; $A \rightarrow k_A$; $B \rightarrow k_B$

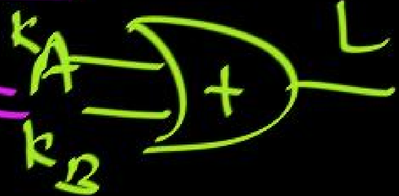
Digital circuit

L opens iff k_A or k_B

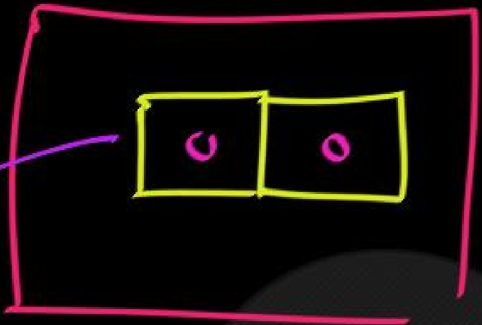
k_A	k_B	L
0	0	0
0	1	1
1	0	1
1	1	1

$$\boxed{L = k_A + k_B}$$

Digital circuit $L =$



Eg:



Digital circuit

Joint locker L } $\underbrace{A, B}_{\text{Two people}}$ — $A \to k_A$, $B \to k_B$

L opens iff k_A or k_B

Assume A lost his key.

k_A	k_B	
1	0	
1	1	

Input Combination Can Never Occur.

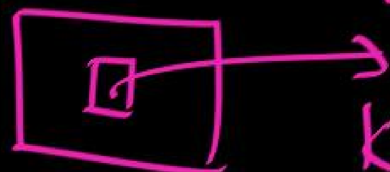
K_A	K_B	L
0	0	0
0	1	1
1	0	d
1	1	d

Can NEVER occur.

Replace Locker L with K_B

(No need of OR GATE)

New locker:



opens using K_B only.

2. For a Boolean function/digital circuit;

Sometimes Some Input Combinations can occur BUT we do not care about what happens for those combinations:

Eg: $f(A,B)$; $\boxed{f=1 \text{ if } A=1}$ ✓

→ we don't care what happens when $A=0$.

A	B	f
0	0	d
0	1	d
1	0	1
1	1	1

A	B	$f_1(A,B)$	f_2	f_3	f_4
0	0	0	0	1	1
0	1	0	1	0	1
1	0	1	1	1	1
1	1	1	1	1	1

Simplest Circuit

$\boxed{f = 1 \text{ when } A = 1}$ ✓

Don’t-care Combinations

Don’t-care combination ϕ: combination for which the value of the function is not specified. Either

- input combinations may be invalid
- precise output value is of no importance

Since each don’t-care can be specified as either 0 or 1, a function with k don’t-cares corresponds to a class of 2^k distinct functions. Our aim is to choose the function with the minimal representation

- Assign 1 to some don’t-cares and 0 to others in order to increase the size of the selected cubes whenever possible
- No cube containing only don’t-care cells may be formed, since it is not required that the function equal 1 for these combinations

functions with Don't care Combinations

$\equiv$ Incompletely Specified functions

$\equiv$ Partial functions

Every Partial function is a Class of functions

By Default,

function $\equiv$ Total function $\equiv$ Completely Specified function

$$f(a,b) = \sum(0,1) + d(2,3)$$

a	b	f	f_1	f_2	f_3	f_4
0	0	1	1	1	1	1
0	1	1	1	1	1	1
1	0	X	0	0	1	1
1	1	X	0	1	0	1

A partial fun with k Don't Care Combinations (X's) corresponds to a class of 2^k functions.

$f(a,b) = \sum(0,2) + d(3) \Rightarrow$ mSop ?

a	b	f	f_1	f_2
0	0	1	1	1
0	1	0	0	0
1	0	1	1	1
1	1	X	0	1

mSop $= \bar{b}$ $\quad \bar{b} + a$

$f(a,b) = \sum(0,2) + d(3)$

mSop $= \bar{b}$

Taking $X=0$

f_1 gives mSop

$f(a,b) = \sum(0,2) + d(3) \Rightarrow mpos$

a	b	f	f_1	f_2
0	0	1	1	1
0	1	0	0	0
1	0	1	1	1
1	1	X	0	1

$mpos = \qquad \bar{b} \qquad (\bar{b} + a)$

$f(a,b) \Rightarrow mpos = \bar{b}$

Taking $x = 0$

Cards → "Joker" → use it as any Card of your Choice/ Desire.

use as you wish,

→ Different Jokers can be taken as Different/ Same Cards.

Don't Cares X's like Jokes

0 or 1

As per your Desire.

Eg: $f(a,b) = \sum(0) + d(2,3)$

minimized SOP ?

Eg: $f(a,b) = \Sigma(0) + d(2,3)$

minimized SOP

$b \backslash a$	0	1
0	1	x
1	0	x

$\bar{a}\,\bar{b} \Longrightarrow$ minimized? No

Eg: $f(a,b) = \Sigma(0) + d(2,3)$

minimized SOP

b \ a	0	1
0	1	X=1
1	0	X=0

$\bar{b}$

minimized SOP.

Ep: $f(a,b) = \Sigma(0) + d(2,3)$

minized Sop

$b \backslash a$	0	1
0	1	X=1
1	0	X=1

$\rightarrow \bar{b}$

$\rightarrow a$

$a + \bar{b} \Rightarrow$ NoT minimized.

Eg: $f(a,b) = \sum(0) + d(2,3)$

minimized POS:

Eg: $f(a,b) = \Sigma(0) + d(2,3)$

minimized POS:

	$a = 0$	$a = 1$ ($\bar{a}$)
$b = 0$	1	X=1
$b = 1$ ($\bar{b}$)	0	X=0

$\bar{b}$

I don't care!

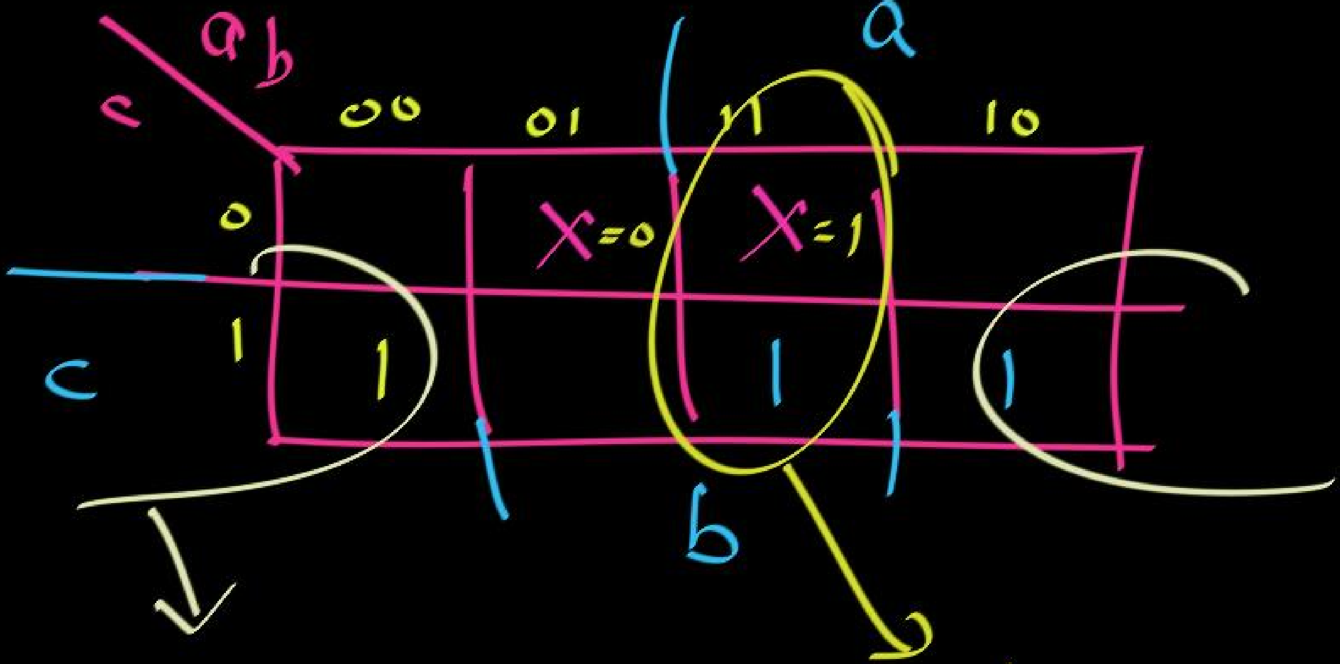
- You don't always need all 2^n input combinations in an n-variable function
  - If you can guarantee that certain input combinations never occur
  - If some outputs aren't used in the rest of the circuit
- We mark don't-care outputs in truth tables and K-maps with Xs.

x	y	z	f(x,y,z)
0	0	0	0
0	0	1	1
0	1	0	X
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	X
1	1	1	1

0/1

0/1

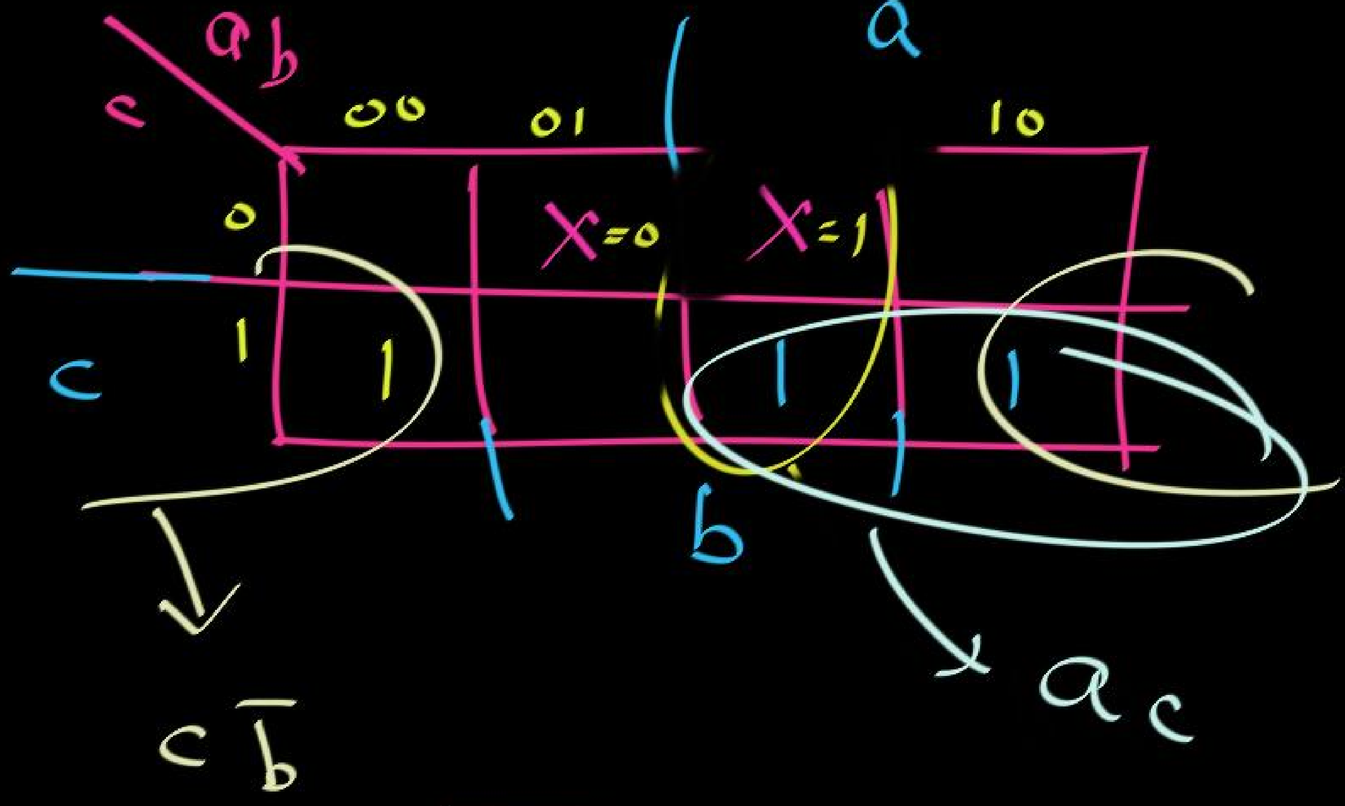
- Within a K-map, each X can be considered as either 0 or 1. You should pick the interpretation that allows for the most simplification.



$c\bar{b}$ ba

mSOP

$$\underbrace{c\bar{b} + ba}$$



mSOP

$\underbrace{c\bar{b} + ac}$

$\boxed{mSOP = 2}$

$\bar{a}$

$c \backslash ab$	00	01	11	10
0	0	X=0	X=0	0
$\bar{c}$ 1	1	0	1	1

$\bar{b}$

$\underbrace{\text{mpos}}$ ✓

$\rightarrow c$

$\rightarrow (\bar{b}+a)$

$$\text{mpos} = (c)(a+\bar{b})$$

I don't care!

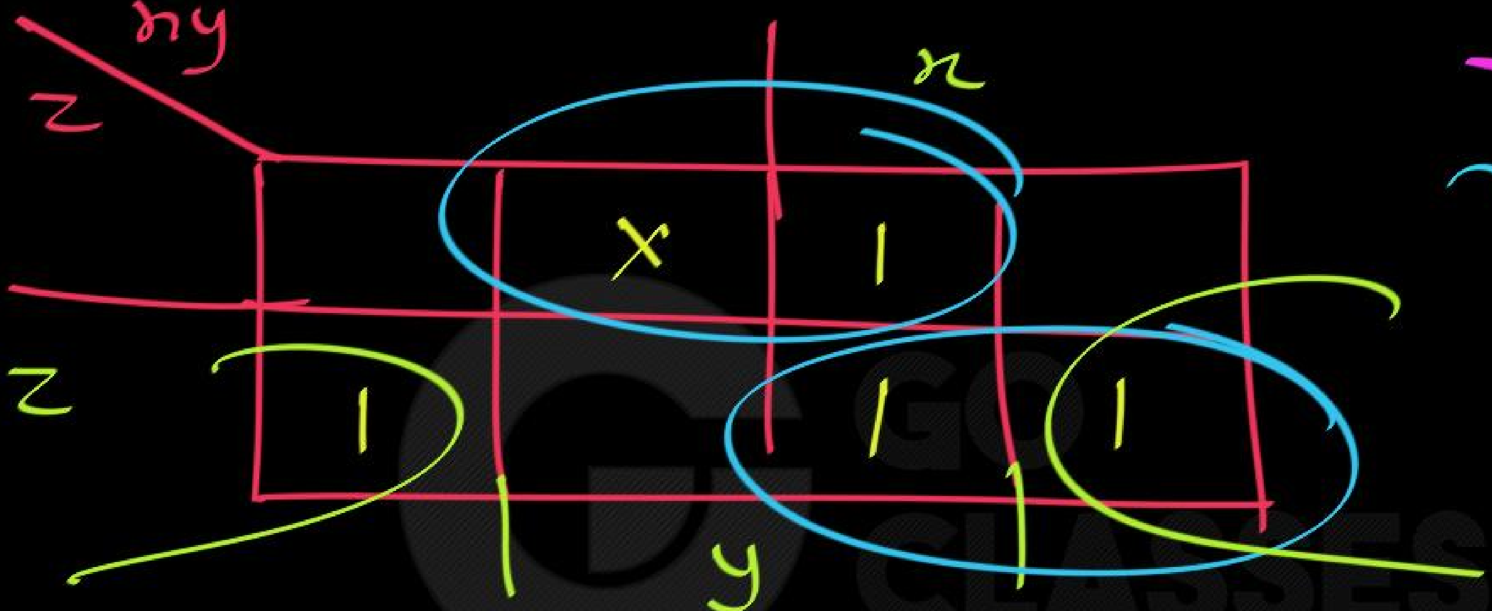
- You don't always need all 2^n input combinations in an n-variable function
  - If you can guarantee that certain input combinations never occur
  - If some outputs aren't used in the rest of the circuit
- We mark don't-care outputs in truth tables and K-maps with Xs.

x	y	z	f(x,y,z)
0	0	0	0
0	0	1	1
0	1	0	X
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

0/1

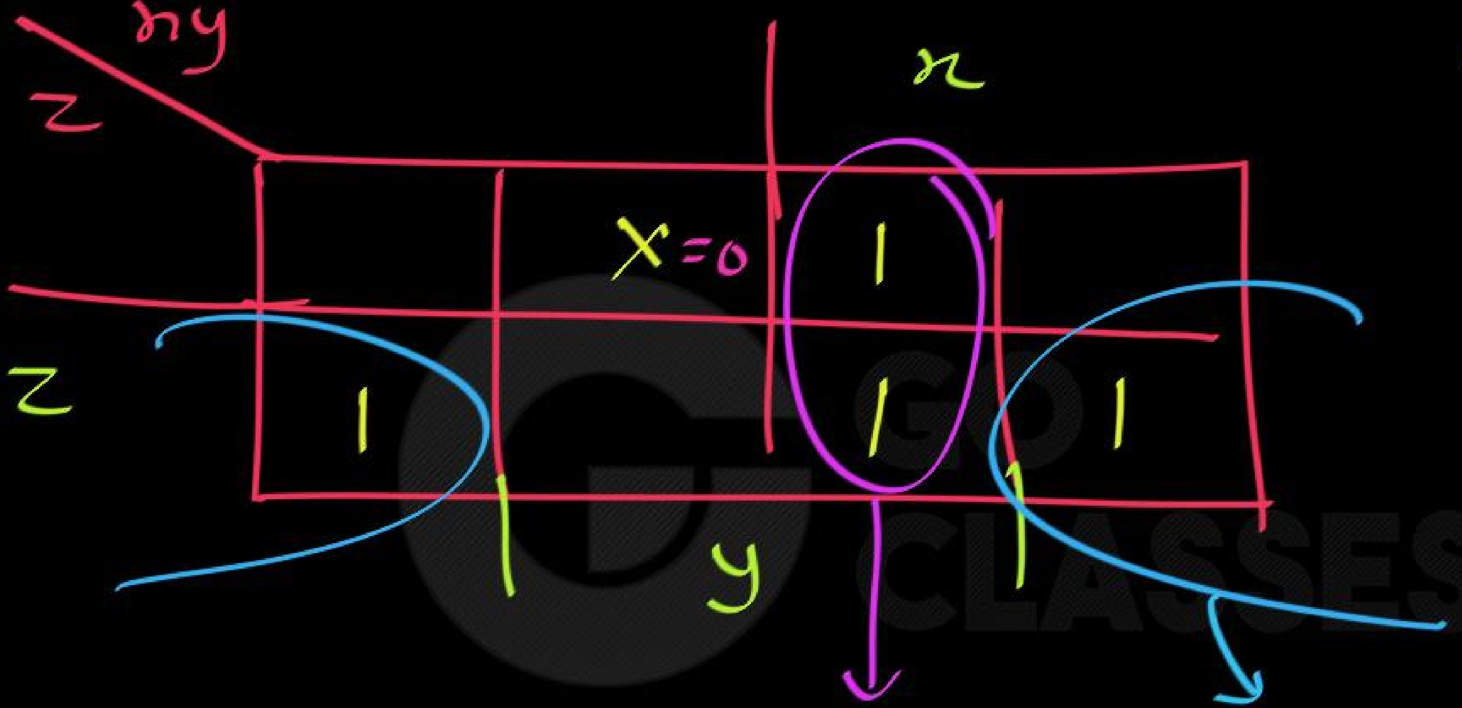
- Within a K-map, each X can be considered as either 0 or 1. You should pick the interpretation that allows for the most simplification.

msop



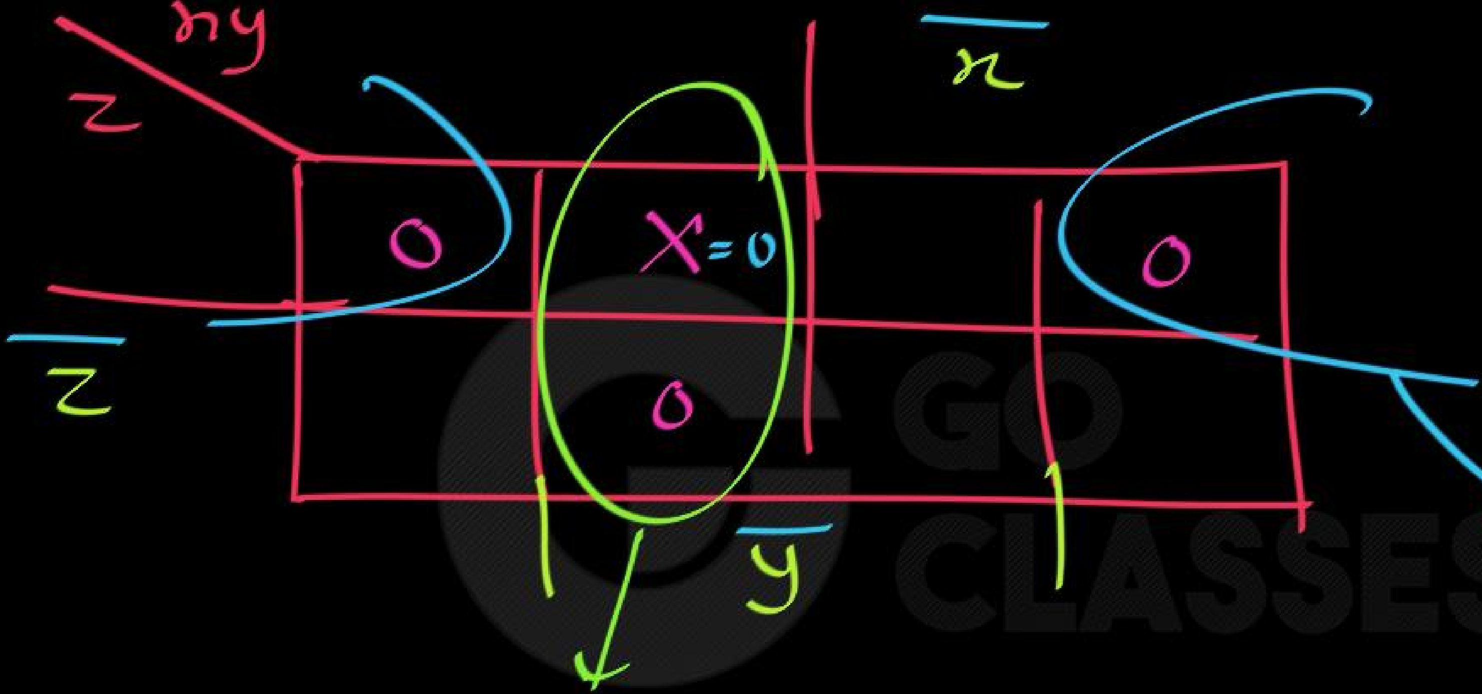
NOT msop

msop



$$msop = xy + z\bar{y}$$

mpos



$(y + z)$

$(x + \bar{y})$

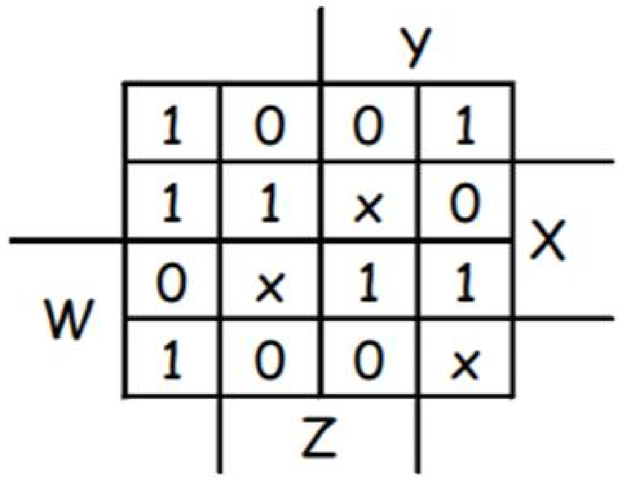
$$mpos = (y+z)(x+\bar{y})$$

Practice K-map 3

- Find a MSP for

$$f(w,x,y,z) = \Sigma m(0,2,4,5,8,14,15),\ d(w,x,y,z) = \Sigma m(7,10,13)$$

This notation means that input combinations wxyz = 0111, 1010 and 1101 (corresponding to minterms m_7, m_{10} and m_{13}) are unused.



Practice K-map 3

- Find a MSP for

$$f(w,x,y,z) = \Sigma m(0,2,4,5,8,14,15),\ d(w,x,y,z) = \Sigma m(7,10,13)$$

This notation means that input combinations wxyz = 0111, 1010 and 1101 (corresponding to minterms m_7, m_{10} and m_{13}) are unused.

			y	y	
	1	0	0	1	
	1	1	x	0	X
W	0	x	1	1	X
W	1	0	0	x	
		z	z		

4 terms

Practice K-map 3

minimized sum of product

- Find a MSP for

$$f(w,x,y,z) = \Sigma m(0,2,4,5,8,14,15),\ d(w,x,y,z) = \Sigma m(7,10,13)$$

This notation means that input combinations wxyz = 0111, 1010 and 1101 (corresponding to minterms m_7, m_{10} and m_{13}) are unused.

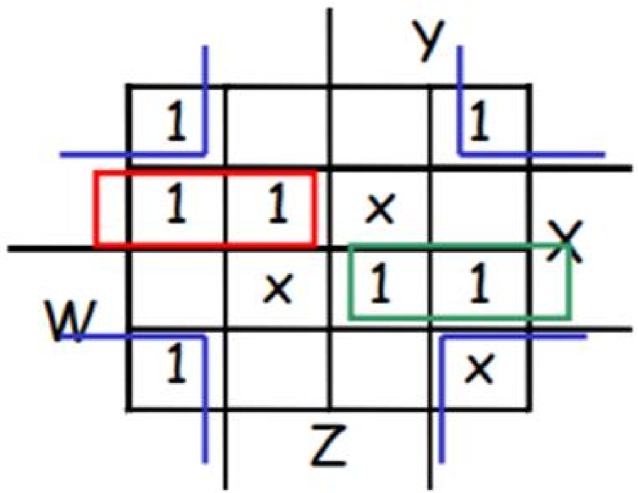
			y		
	1	0	0	1	
	1	1	x	0	X
W	0	x	1	1	
	1	0	0	x	
		Z			

3 terms ✓

Solutions for Practice K-map 3

- Find a MSP for:

$$f(w,x,y,z) = \Sigma m(0,2,4,5,8,14,15),\ d(w,x,y,z) = \Sigma m(7,10,13)$$



$$f(w,x,y,z) = x'z' + w'xy' + wxy$$

GO CLASSES

Imperial College London

Classes

Don’t Care Conditions

- In certain cases some of the minterms may never occur or it may not matter what happens if they do
  - In such cases we fill in the Karnaugh map with and X
    - meaning <u>don't care</u>
  - When minimizing an X is like a "joker"
    - X can be 0 or 1 - whatever helps best with the minimization
  - Eg:

A\BC	00	01	11	10
0	0	0	1	X
1	0	0	1	1

  - simplifies to *B* if X is assumed 1

$$X = B$$

More “Don’t Care” examples

“Don’t care” conditions should be changed to either 0 or 1 to produce K-map looping that yields the simplest expression.

A	B	C	z
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	x
1	0	0	x
1	0	1	1
1	1	0	1
1	1	1	1

x } "don't care"

(a)

	$\bar{C}$	C
$\bar{A}\bar{B}$	0	0
$\bar{A}B$	0	x
AB	1	1
$A\bar{B}$	x	1

(b)

$\Rightarrow$

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

$\rightarrow \boxed{z = A}$

(c)

K-map Summary

- K-maps are an alternative to algebra for simplifying expressions
  - The result is a *minimal sum of products*, which leads to a minimal two-level circuit
  - It's easy to handle don't-care conditions
  - K-maps are really only good for manual simplification of small expressions...
- Things to keep in mind:
  - Remember the correct order of minterms on the K-map
  - When grouping, you can wrap around all sides of the K-map, and your groups can overlap
  - Make as few rectangles as possible, but make each of them as large as possible. This leads to fewer, but simpler, product terms
  - There may be more than one valid solution

Basic Rules of Karnaugh maps

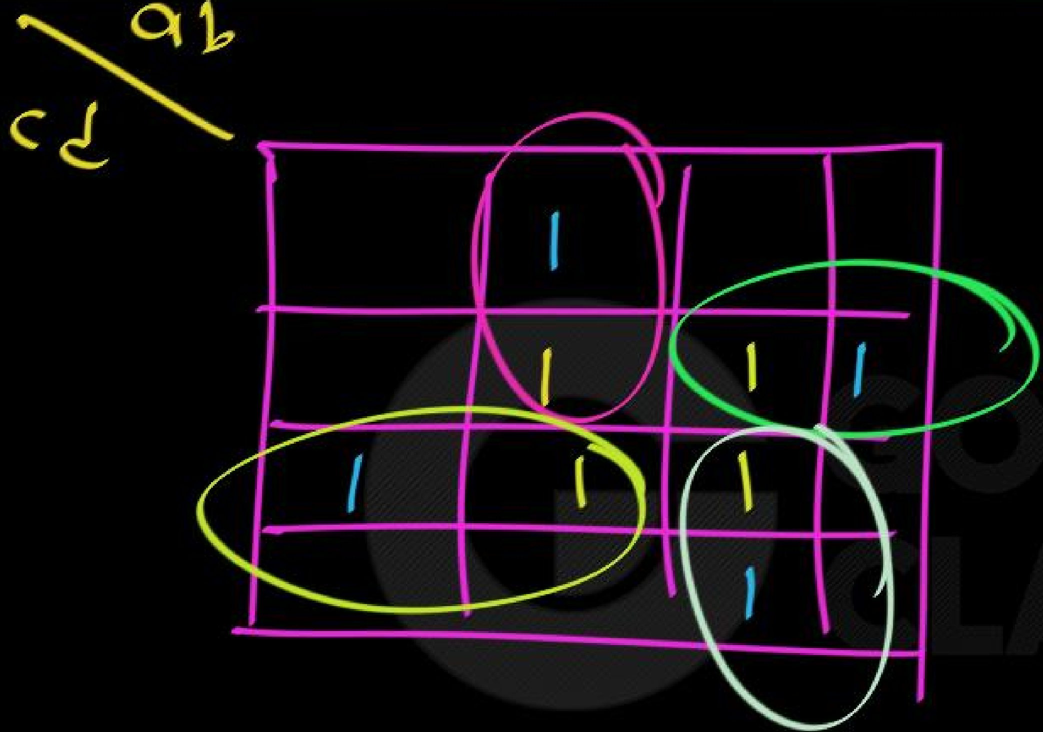
- Anytime you have N variables, you will have 2^N possible combinations, and 2^N places in a truth table or K-Map.
- In a Karnaugh Map of any size, crossing a vertical or horizontal cell boundary is a change of only one variable -- no matter how many variables there are.
- Each single cell that contains a 1 represents a minterm in the function, and each minterm can be thought of as a "product" term with N variables.
- To combine variables, use groups of **1, 2, 4, 8,** etc. A group of 2 in an N-variable Karnaugh map will give you a "product" term with N-1 variables. A group of 4 will have N-2 variables, etc.
- You will never have a group of 3, a group of 5, etc.

Q:

Largest Cube is ALWAYS present in the minimum SOP(or minimum POS) expression?

Q:

Largest Cube is ALWAYS present in the minimum SOP(or minimum POS) expression? No


ab
cd

Next Topic:

Covering *Functions*

(F -------> G)

$f = a + ab = a$

$\left\{ \begin{array}{l} a\bar{b} \\ ab \end{array} \right\}$

$\boxed{a \text{ Covers } ab.}$

a	b	f = a	g = ab
0	0	0	0
0	1	0	0
1	0	1	0
1	1	1	1

f Covers g iff whenever

g=1 then f=1

OR whenever f=0 then g=0

f covers g

g	f
1 →	1
1 →	1
0 →	0/1

f Covers g :

g	f
1	1
1	1
0	0/1
0	0
0/1	1

Q: $f = \underbrace{a+b}$ $\quad g = \underbrace{ab}$

f covers g

Whenever $g=1$ then $f=1$.

ab'

$a'b$

ab $\quad g$

Q: $f = a \oplus b \quad g = a + b$

g Covers f.

$f = 1 \Longrightarrow g = 1$

Q: $f = a \oplus b \qquad g = ab$

$\left.\begin{array}{l} f \text{ Doesn't Cover } g. \\ g \quad \text{"} \quad \text{"} \quad f. \end{array}\right\}$

Note: If f Covers g, g Covers f then $\boxed{f = g}$

f	g
1	1
1	1
0	0
0	0

Analogy

f Covers g $\equiv$ $g \rightarrow f$ $\equiv$ $g \leq f$

minterm (g) $\subseteq$ minterms (f)

f Covers g

$\equiv$ g implies f

$\equiv$ $\underbrace{g=1} \Longrightarrow \underbrace{f=1}$

$\equiv f=0 \Longrightarrow g=0$

Study in propositional logic (in Discrete maths)

- f_1 implies f_2 $(f_1 \rightarrow f_2)$
  - There is no assignment of values to the n variables that makes f_1 equal to 1 and f_2 equal to 0.
  - Whenever f_1 equals 1, then f_2 must also equal 1.
  - Whenever f_2 equals 0, then f_1 must also equal 0.
- Concept can be applied to terms and formulas.

Note: Covering of functions is Defined from 1's point of view (minterm point of view)

f covers g : $\boxed{g=1 \Rightarrow f=1}$

- Example: $x\bar{y}\,\bar{z}, x\bar{z}$

$x\bar{y}\bar{z}$ implies $x\bar{z}$.

$x\bar{z}$ Covers $x\bar{y}\bar{z}$.

$$x\bar{z} \left\{ \begin{array}{l} x\bar{z}\bar{y} \\ x\bar{z}y \end{array} \right.$$

whenever $\underbrace{x\bar{y}\bar{z}=1} \Longrightarrow \underbrace{x\bar{z}=1}$

$x=1, y=0, z=0$

- Example: [redacted]

$$x + \overline{y} + \overline{z}, x + \overline{z}$$

$x + \overline{z}$ Covers $x + \overline{y} + \overline{z}$

$x + \overline{y} + \overline{z}$ Covers $x + \overline{z}$

$x + \overline{z} + \overline{y}$, $x + \overline{z} + y$

$\underbrace{x + \overline{z} + \overline{y} = 0}$ $\Longrightarrow$ $\underbrace{x + \overline{z} = 0}$

$x = 0$

$z = 1, y = 1$

$$\boxed{x + \overline{z} = 1 \Longrightarrow x + \overline{z} + \overline{y} = 1}$$

f Covers g : (g implies f)

OR

$$\left.\begin{aligned} g=1 &\Longrightarrow f=1 \\ f=0 &\Longrightarrow g=0 \end{aligned}\right\}$$

$x+\bar{y}+\bar{z}=0 \Longrightarrow x+\bar{z}=0$

$x=0, y=1, z=1$

$\boxed{f=0 \Longrightarrow g=0}$ means

$|||$

$\boxed{g=1 \Longrightarrow f=1}$

f Covers g

$f_1 \longrightarrow$ • CNF: $(x + y)(x + y + z)$

$f_2 \longrightarrow$ • DNF: $xy + xyz$

Who covers whom?

$f_1 \rightarrow$ • CNF: $(x + y)(x + y + z) = x + y$

$f_2 \rightarrow$ • DNF: $xy + xyz = xy$

$\underbrace{xy + xyz}$

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

$x=1, y=1$

CNF Covers DNF.

$x+y$ Covers xy

Note: $f_{(a,b,c)} = \Sigma(0,5,7)$

Covering function of f =

$$g_{(a,b,c)} = \Sigma(0,5,7) + d(1,2,3,4,6)$$

Note: $f_{(a,b,c)} = \Pi(0,5,7)$

Covering function of f =

WRONG

$g_{(a,b,c)} = \Pi(0,5,7) + d(1,2,3,4,6)$

Wrong

Note: $f_{(a,b,c)} = \prod(0,5,7)$

Covering function of f =

$$g_{(a,b,c)} = \sum(1,2,3,4,6) + d(0,5,7)$$

Q:

For any F, the minimum SOP expression for "Covering function of F" ?

Q:

For any F, the minimum SOP expression for "Covering function of F" ?

= 1

f	Covering function	In SOP of Covering function
0	X	1
1	1	1
0	X	1
0	X	1
1	1	1
1	1	1
0	X	1
0	X	1

$\Rightarrow 1$

a	b	f_1	f_2	f_3	f_4	f_5
0	0	0	0	1	1	1
0	1	0	1	0	0	1
1	0	0	0	1	1	1
1	1	0	1	0	1	1

f_5 Covers all.

f_1 is Covered by everyone.

$f_2 \not\rightarrow f_4$

$f_4 \not\rightarrow f_2$

Q:

For any F, the minimum POS expression for "Covering function of F" ?

Q:

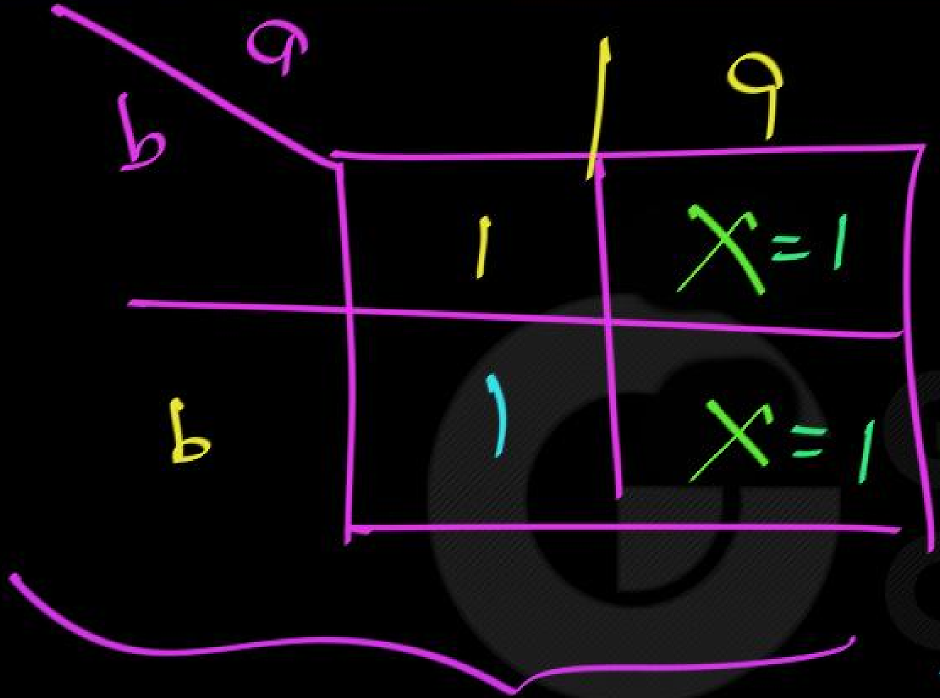
For any F, the minimum POS expression for "Covering function of F" ? = 1

POS expression

$f =$

b \ a		a
	1	0
b	1	0

find mPOS of Covering function.



$\Rightarrow$ Covering function of f.

mPOS : 1

Q:

When does one product term implies another product term?

$xz \longrightarrow x$ ✓

$xz \longrightarrow z$ ✓

$x\bar{y}z \longrightarrow x\bar{y} \longrightarrow x$

$x\bar{y} \longrightarrow \bar{y}$

$x\bar{y}z \longrightarrow xz \longrightarrow z$

$x\bar{y}z \longrightarrow \bar{y}z \longrightarrow \bar{y}$

$x\bar{y}z$

$\downarrow$

$\bar{y}z$

$\downarrow$

$\bar{y}$

Bigger product term implies its subset product term.

Product terms P_1, P_2 :

P_1 implies P_2

iff literals(P_2) $\subseteq$ literals(P_1)

Smallest product terms covers its superset product term.

Q:

When does one Sum term implies another Sum term?

$$\underbrace{x+y}_{x+y=1} \longrightarrow \underbrace{x+y+\bar{z}}_{x+y+\bar{z}=1}$$

$x+y=1 \longrightarrow x+y+\bar{z}=1$

Smaller Sum term $\longrightarrow$ its superset Sum term

Sum term : S_1, S_2

$S_1 \Longrightarrow S_2$ iff

$\text{literals}(S_1) \subseteq \text{literal}(S_2)$

bigger sum term Cover its subset sum term.

Next Topic:

Implicants(I),

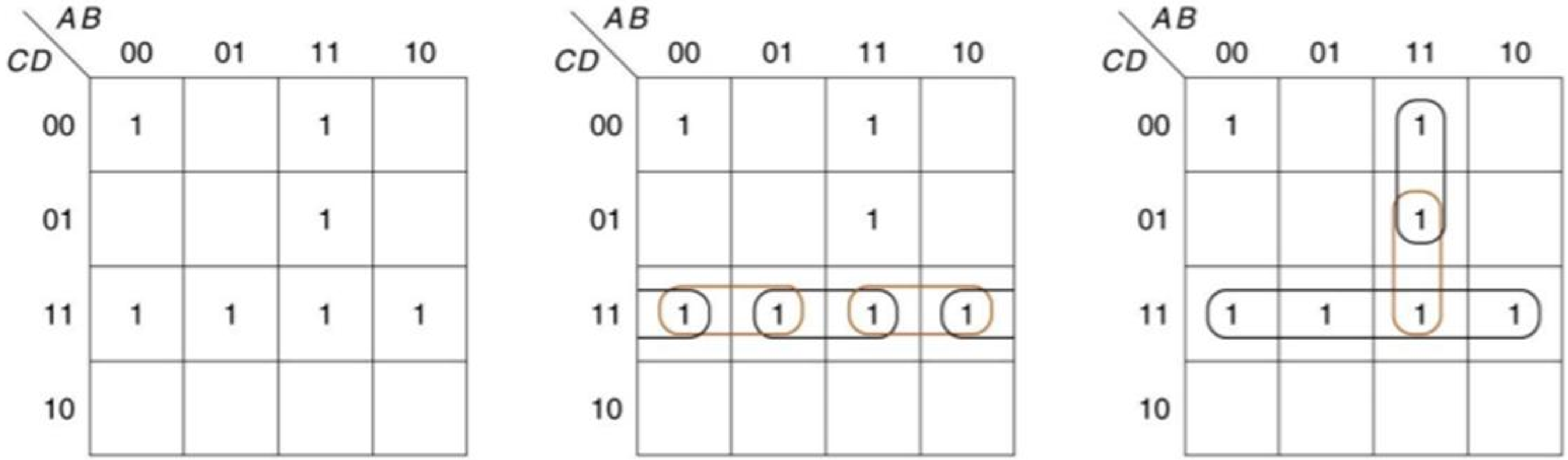
Prime Implicants(PI),

Essential Prime Implicants(EPI)

Implicant

An implicant is a rectangle of 1, 2, 4, 8, . . . (any power of 2) 1’s. That rectangle may not include any 0’s.

Map 3.12 A function to illustrate definitions.



Implicant = Every cube of 1's is Implicant.

An implicant is a rectangle of 1, 2, 4, 8, . . . (any power of 2) 1's. That rectangle may not include any 0's.

2^m cells

Map 3.12 A function to illustrate definitions.

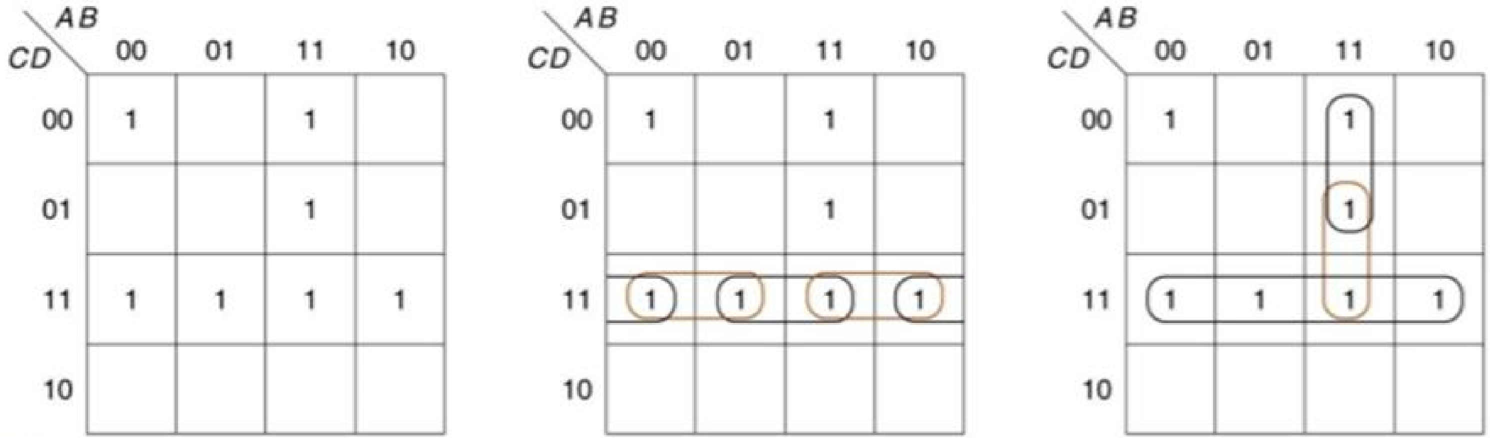
CD \ AB	00	01	11	10
00	1		1	
01			1	
11	1	1	1	1
10				

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

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

Implicant -- 2

Map 3.12 A function to illustrate definitions.



The implicants of *F* are

Minterms	*Groups of 2*	*Groups of 4*
A′B′C′D′	*A′CD*	*CD*
A′B′CD	*BCD*	
A′BCD	*ACD*	
ABC′D′	*B′CD*	
ABC′D	*ABC′*	
ABCD	*ABD*	
AB′CD		

7

6

1

#Implicants = 14

o Classes

Implicant: —— Similar to Implication

Implicant is a Product term for which Implies f.

Implicant I of f :

a product term such that

$$\boxed{I \implies f}$$

$$f_{(a,b,c,d)} = a + cd$$

How many Implicants of f ?

$f_{(a,b,c,d)} = a + cd$

product term

imply f

How many Implicants of f ?

a ✓

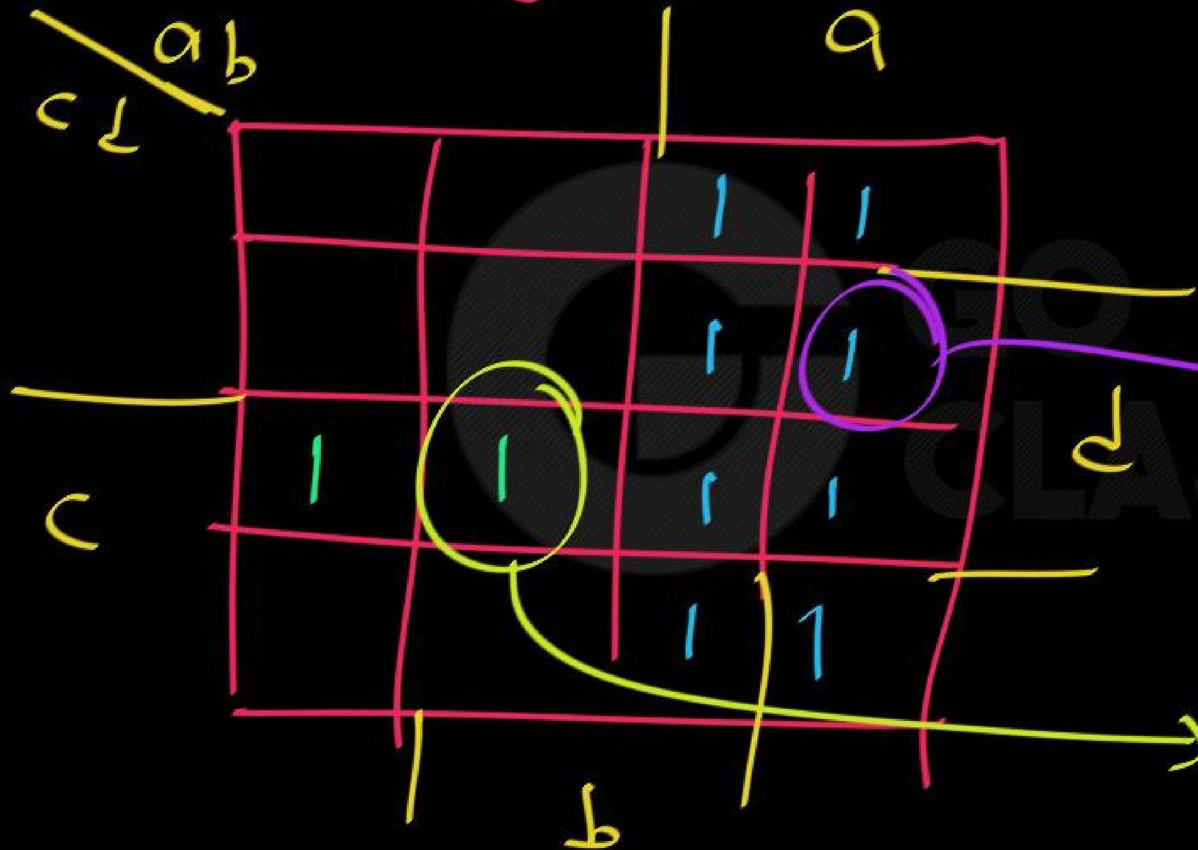
ab' ✓

cd' ✗

cd ✓

$a+cd$ ✗ → Not a product term

$f = a + cd$



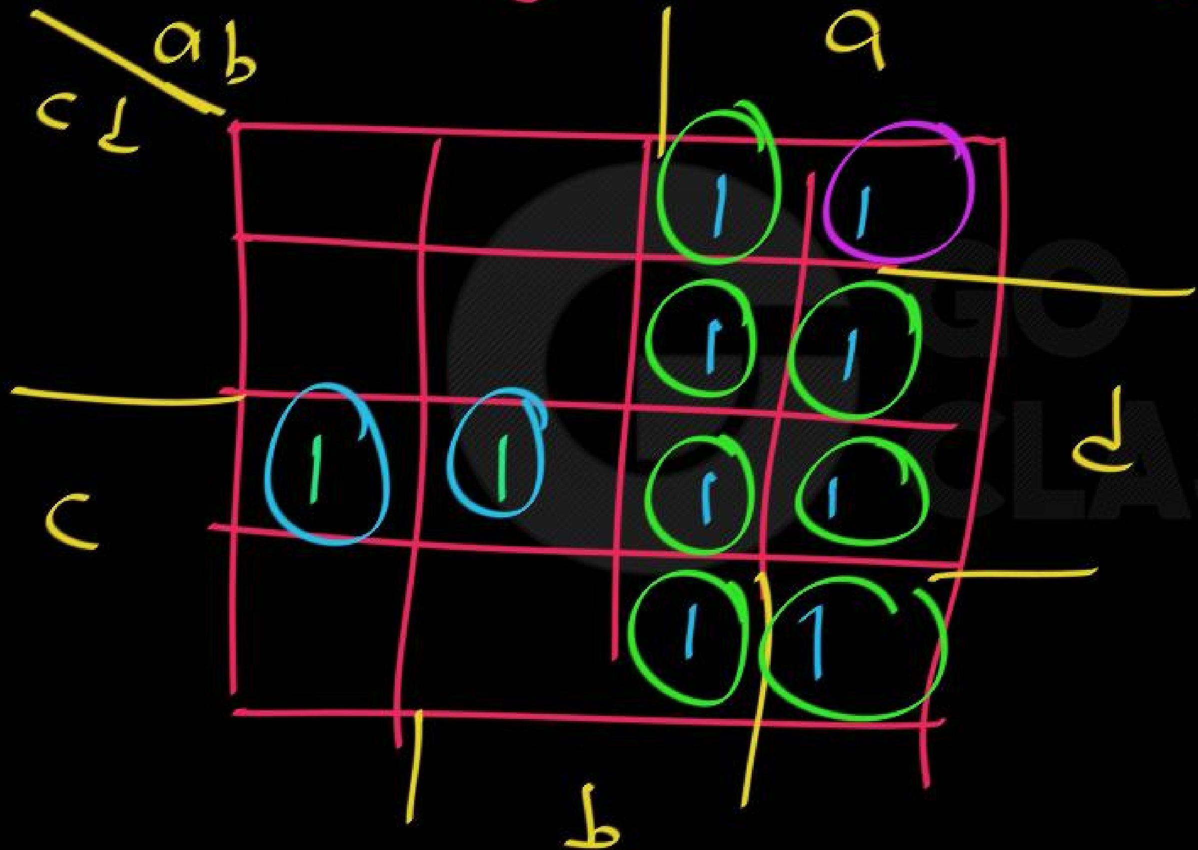
Implicants of size 1 :

10

$\underbrace{a\bar{b}\bar{c}d}$

$\bar{a}bcd$

$f = a + cd$

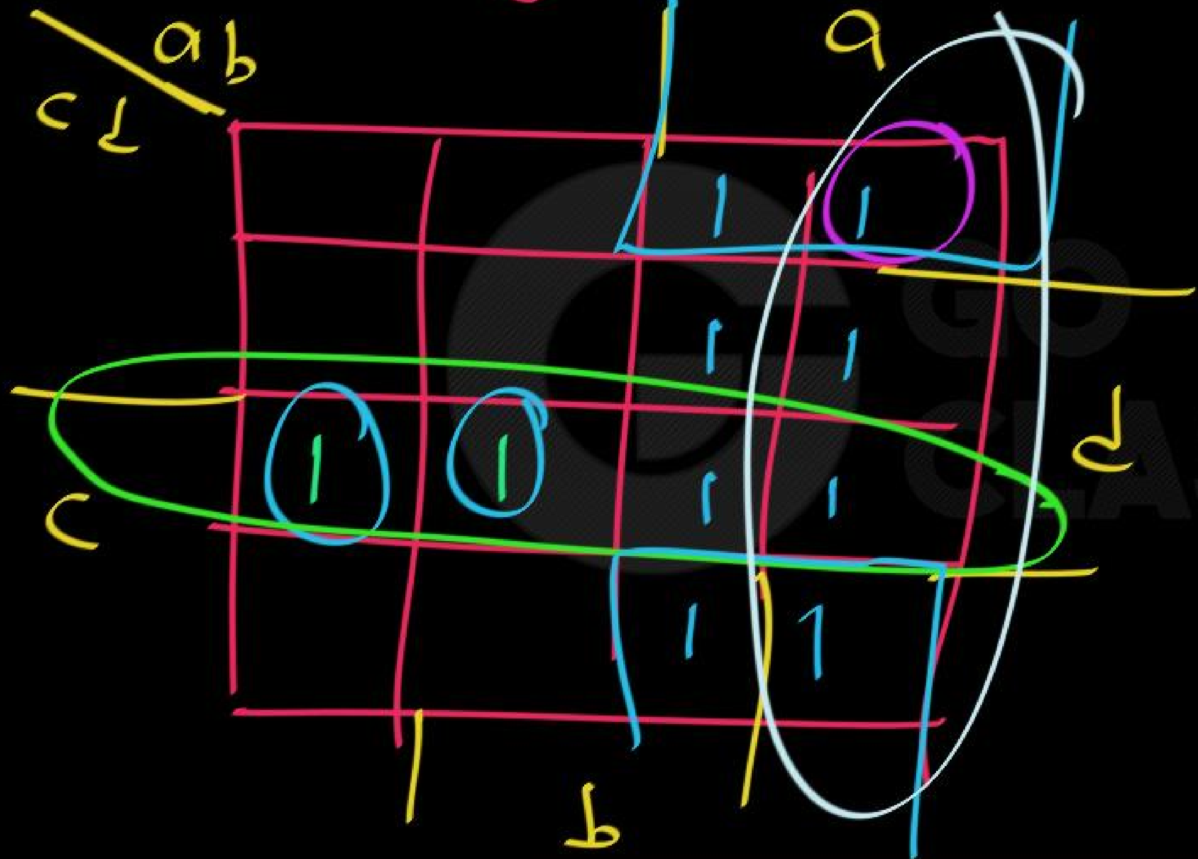


Implicants of size 2:

$\underbrace{2 + 1 + 3 + 2 + 2 + 1 + 2 + 1 + 1.}$

15 ✓

$f = a + cd$

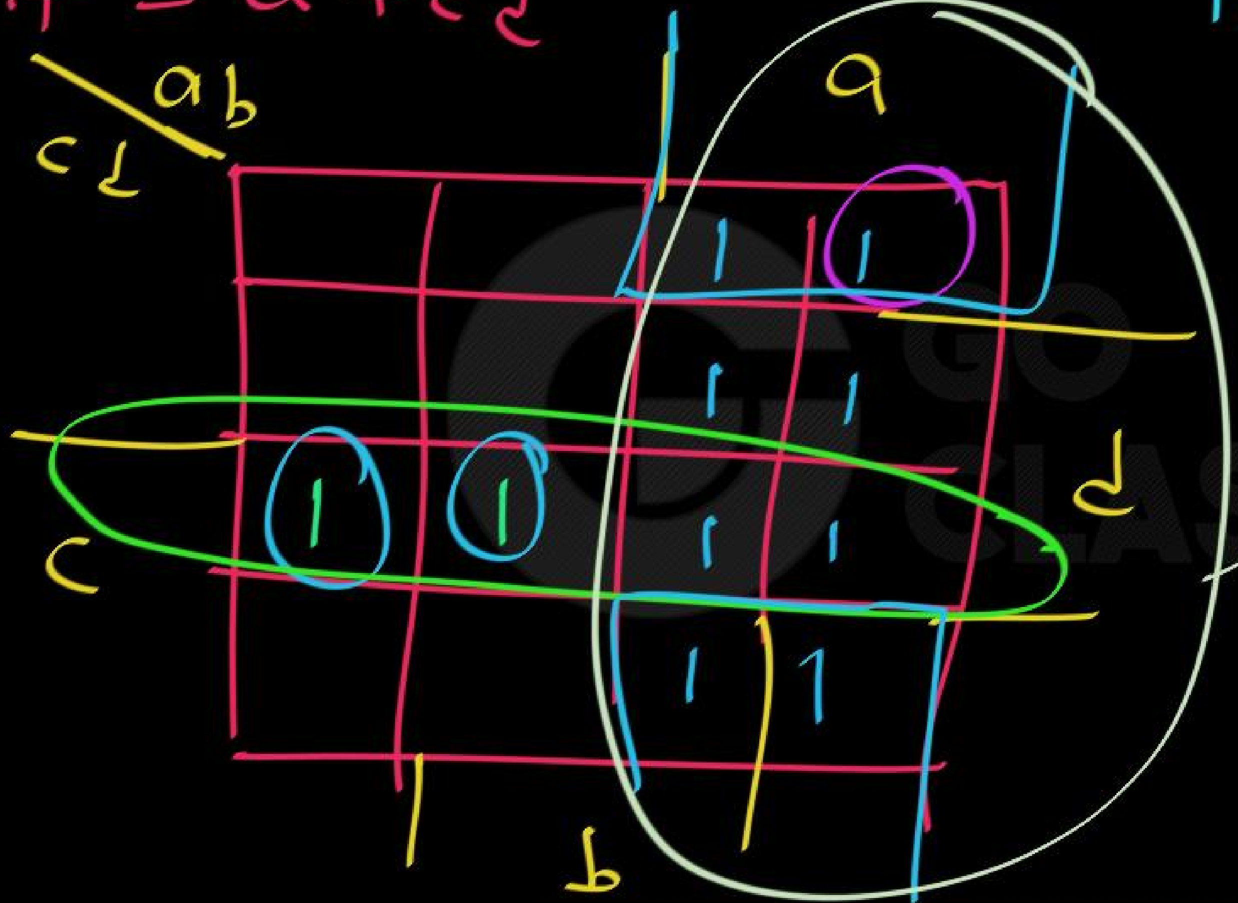


Implicants of size 4:

$\underbrace{1+1+3}+2$

7

$f = a + cd$



Implicants of size 8 = 1

$f(a,b,c,d) = a + cd$

$\#\text{Implicants} = 10 + 15 + 5 + 1$

$= 31$

		1	1
		1	1
1	1	1	1
		1	1

Implicants of Size 2:

$$2 + 1 + 3 + 2 +$$
$$2 + 1 + 2 + 1 + 1$$

$$= 15$$

Implicants: I

1. In kmap; Every cube of 2^m cells covering 1's.

2. Implicant I is a product term which implies f.

$$\boxed{I=1 \Rightarrow f=1}$$

find Implicant ?

$f = a + cd$

$\rightarrow cd$

$f = a + \boxed{cd + cd\bar{a}} + cda + a\bar{c} + a\bar{b} + ab + ab\bar{c}\bar{d} + \text{----}$

Implicant

- An implicant of a function is a product term that can be used in an SOP expression for that function
- From the point of view of the map, an implicant is a rectangle of 1,2,4,8...2^n 1's. No 0's may be included

k-map is Efficient only for ≤ 4 variables.

for ≥ 5 variables, k map Extremely Complicates.

A product term is said to be an implicant of a function if the product term implies the function.

Size 1 cube covering 1.

Each of the "minterms" in canonical SOP "1" form is an implicant of the function.

Implicant : Any cube of 1's.

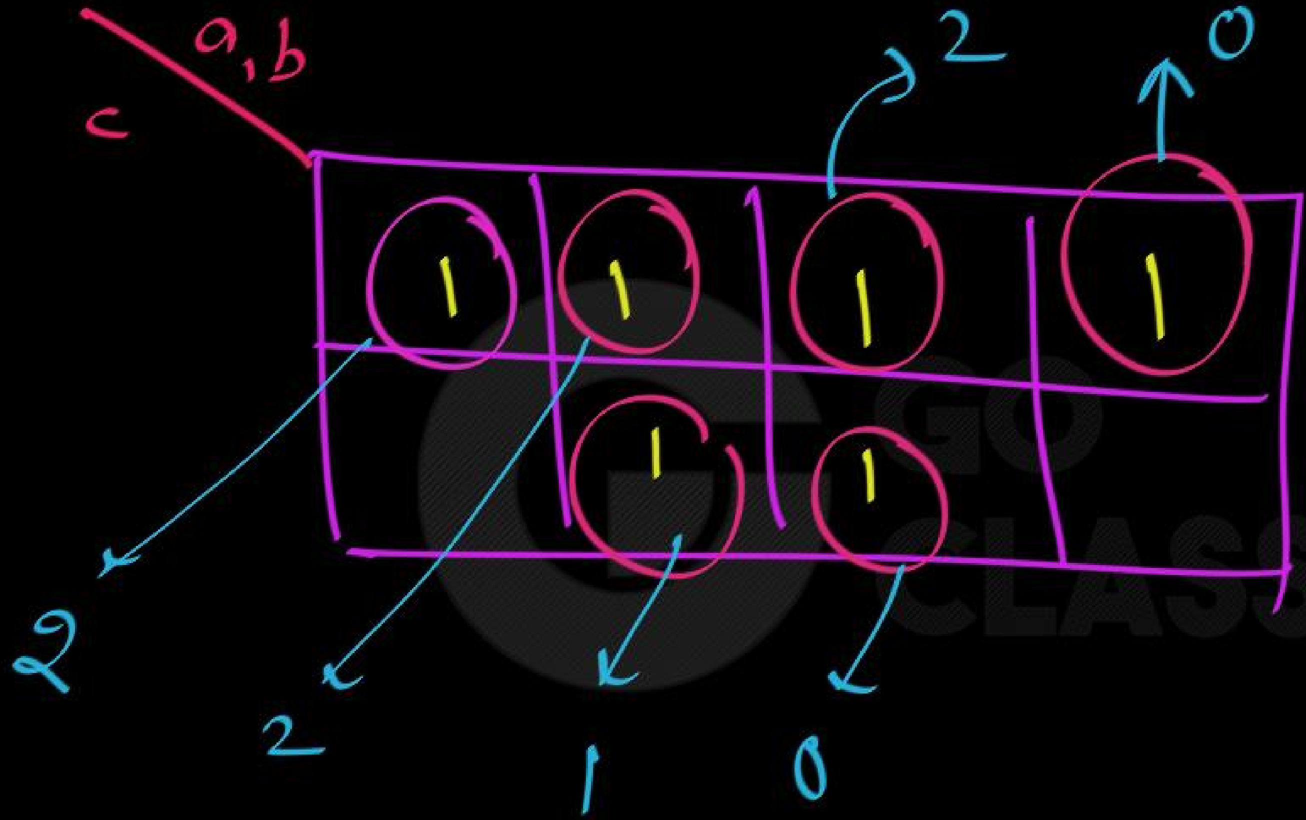
Prime Implicant : Cube of 1's which is as big as possible.

c \ a,b				
	1	1	1	1
		1	1	

Implicants:

of size 1

= 6



Implicants:

Size 2

$\underbrace{2 + 2 + 2 + 1}$

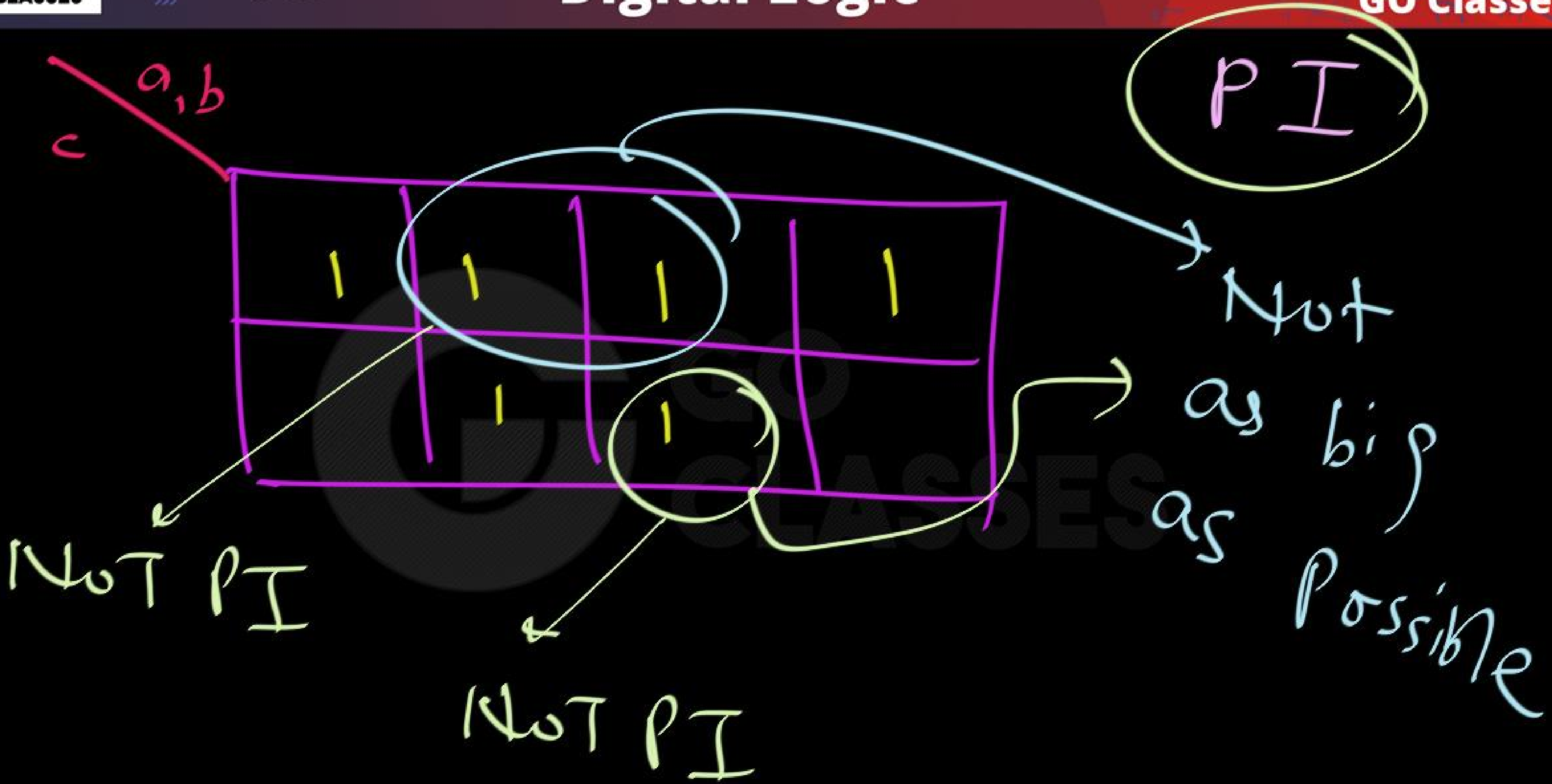
7 ✓

c \ a,b				
	1	1	1	1
		1	1	

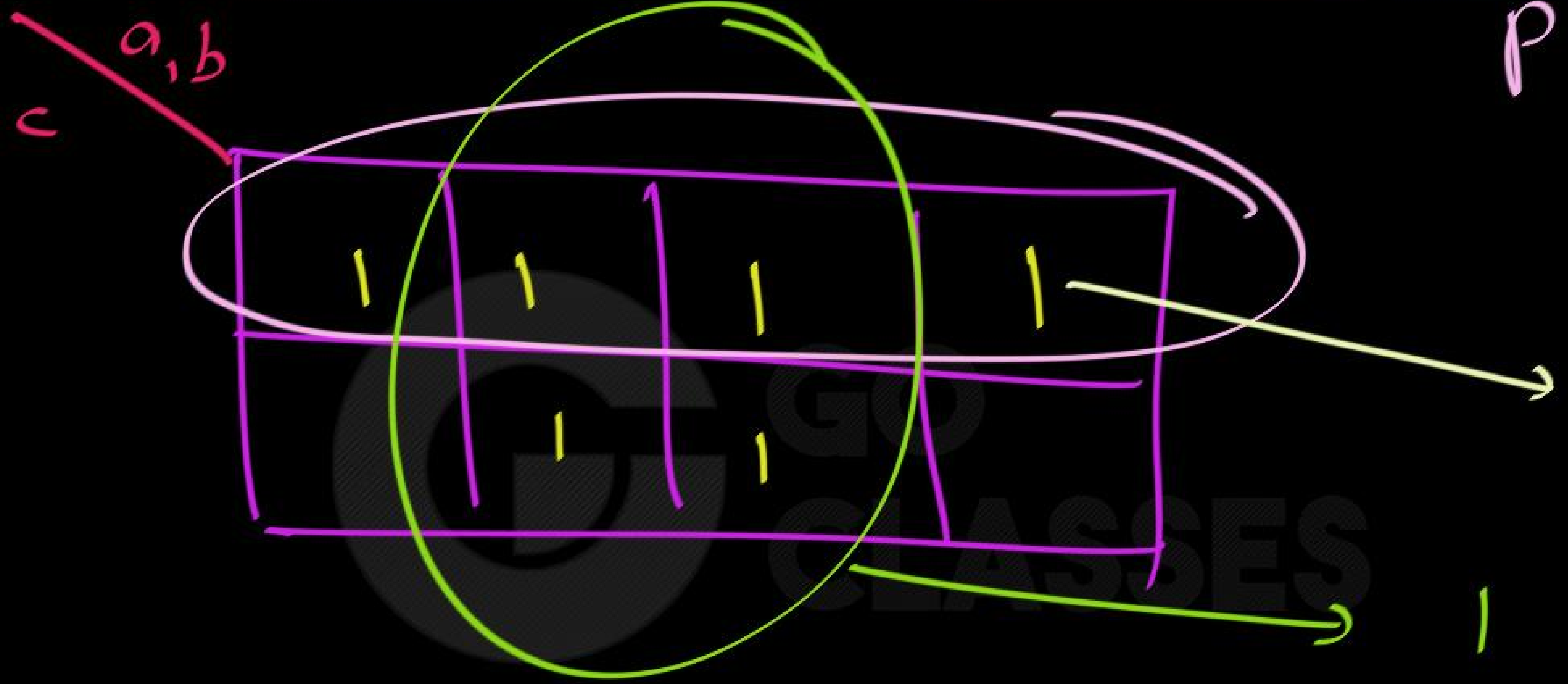
Implicants:

Size 4 = 2

Implicants = 6 + 7 + 2 = 15


a,b
c
1 1 1 1
1 1
PI
Not as big as possible
NOT PI
NOT PI

PI

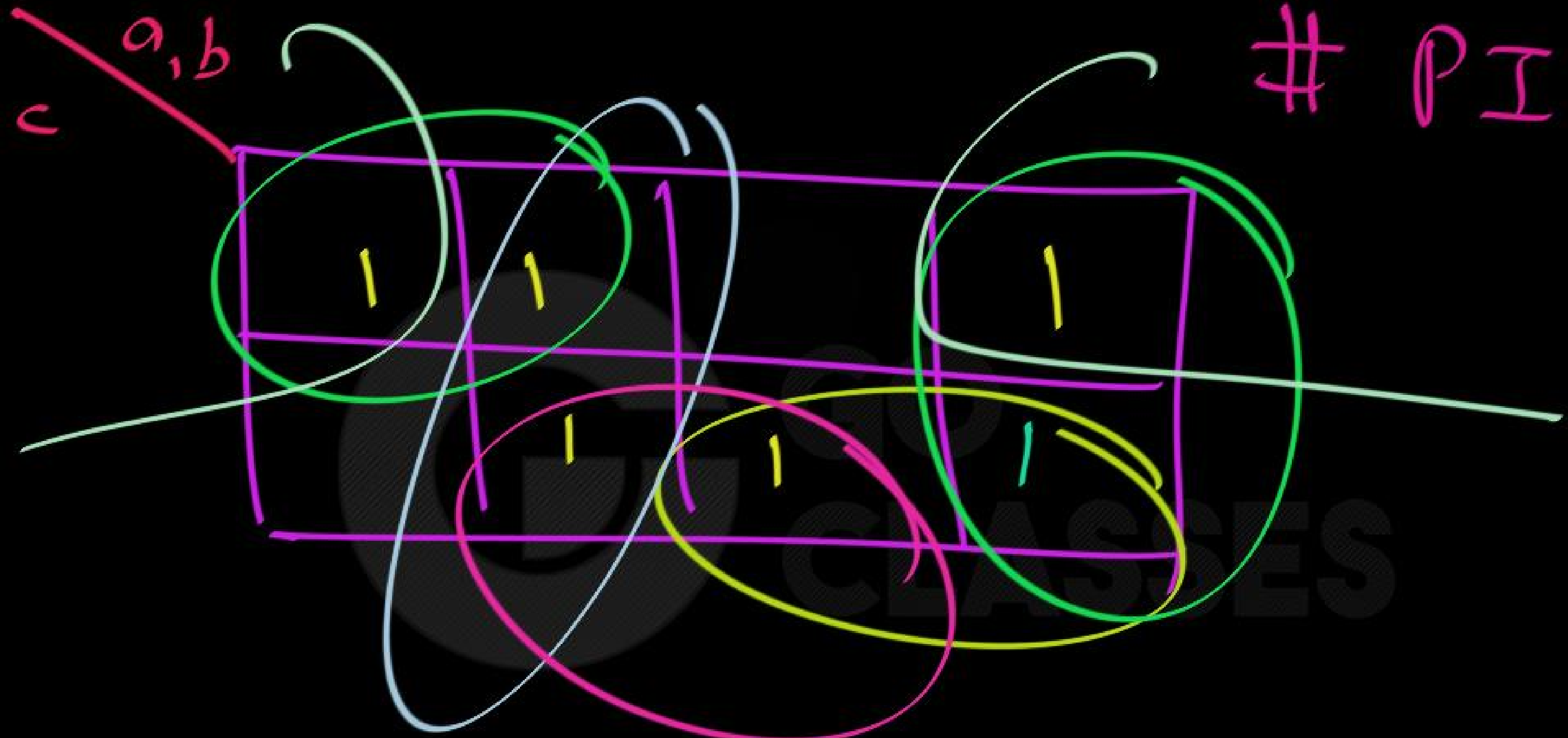


\# PI = 2

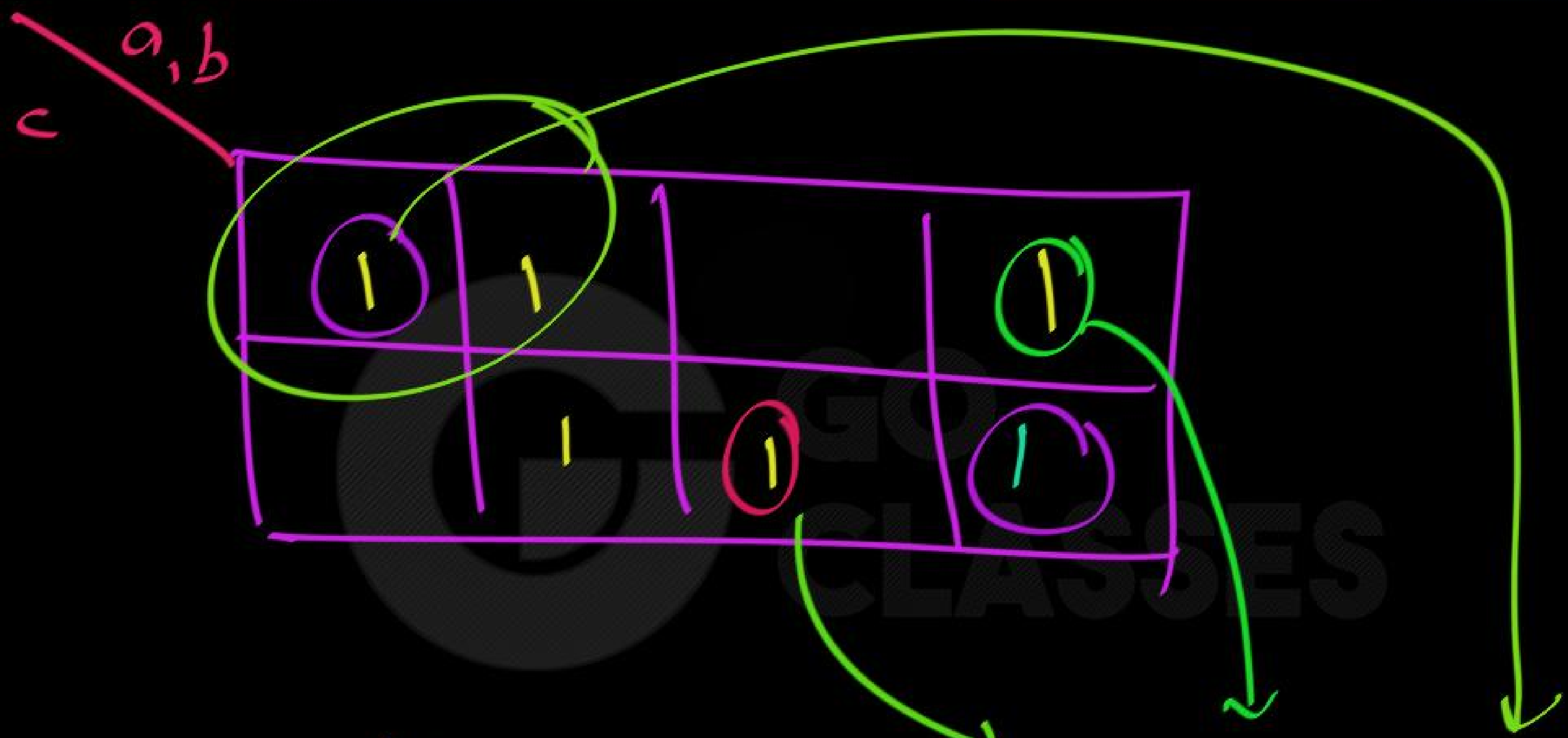
c \ a,b

1	1		1
	1	1	1

PI ?



PI ?



$\# PI = 2 + 2 + 1 + 1 = 6$

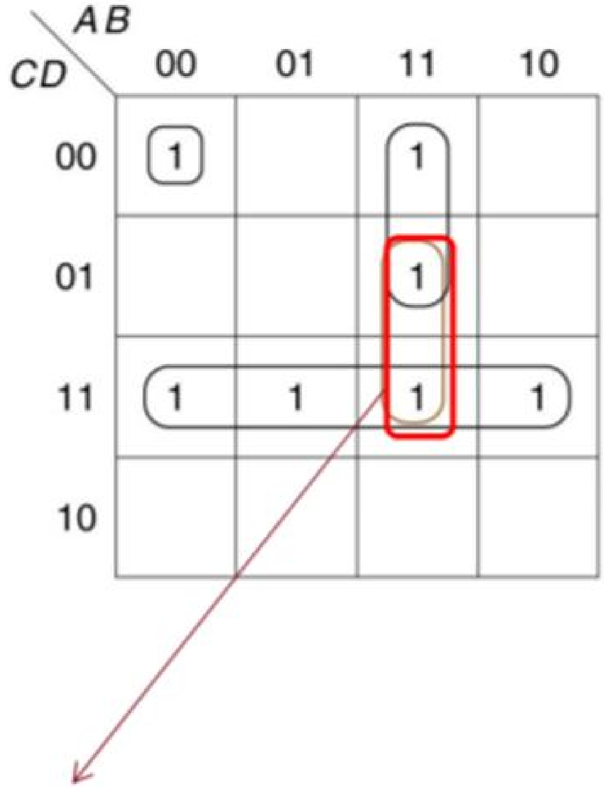
Prime Implicant ≡ a cube of 1's as big as possible.

A prime implicant is an implicant that (from the point of view of the map) is not fully contained within any one other implicant

Prime Implicant

A *prime implicant* is an implicant that (from the point of view of the map) is not fully contained in any one other implicant.

An *essential prime implicant* is a prime implicant that includes at least one 1 that is not included in any other prime implicant.



prime implicant, but not essential prime implicant