Next Topic (VERY Important):

Minimization of Expression

Using *K-Map*

Switching functions can generally be simplified by using the algebraic techniques described previously.

However, two problems arise when algebraic procedures are used:

1. The procedures are difficult to apply in a systematic way.
2. It is difficult to tell when you have arrived at a minimum solution

Unfortunately, the result of this procedure may depend on the order in which terms are combined or eliminated so that the final expression obtained is not necessarily minimum.

$$\underline{\underline{a+b}}+c \qquad a+\underline{\underline{b+c}}$$

$$= \underbrace{a+b}+\underbrace{b+c}$$

$$\alpha+\beta+\gamma+K$$

Find a minimum sum-of-products expression for

$$F(a, b, c) = \Sigma\, m\,(0, 1, 2, 5, 6, 7)$$

$$\begin{aligned} F &= a'b'c' + a'b'c + a'bc' + ab'c + abc' + abc \\ &= a'b' + b'c + bc' + ab \end{aligned} \qquad (5\text{-}1)$$

None of the terms in the above expression can be eliminated by consensus. However, combining terms in a different way leads directly to a minimum sum of products:

$$\begin{aligned} F &= a'b'c' + a'b'c + a'bc' + ab'c + abc' + abc \\ &= a'b' + bc' + ac \end{aligned} \qquad (5\text{-}2)$$

$F = a'b'c' + a'b'c + a'bc' + ab'c + abc' + abc$

Person 1:

$= a'b' + b'c + bc' + ab$

Irredundant Expression

Can Not be further minimize

Problems with Algebraic method:

1. Non-systematic
2. Error Prone
3. Result may depend on order in which we apply this method.
4. We cannot tell whether we got minimum exp or not.

Find a minimum sum-of-products expression for

$$F(a, b, c) = \Sigma m\,(0, 1, 2, 5, 6, 7)$$

$$F = a'b'c' + a'b'c + a'bc' + ab'c + abc' + abc$$

$$= (a'b') + (b'c) + (bc') + (ab) \qquad (5\text{-}1)$$

None of the terms in the above expression can be eliminated by consensus. However, combining terms in a different way leads directly to a minimum sum of products:

$$F = a'b'c' + a'b'c + a'bc' + ab'c + abc' + abc$$

$$= (a'b') + (bc') + (ac) \qquad (5\text{-}2)$$

Last two expressions are Irredundant Sum-of-Products Expressions (We will come back to it later)
So, we can't minimize both the expressions further using Boolean algebraic identities.

So, How do we know in which order to combine terms to get minimum size expression ?
Answer is using K-map.

So, How do we know in which order to combine terms to get minimum size expression ?

Answer is using K-map.

The Karnaugh map method overcomes these difficulties by providing systematic methods for simplifying switching functions.

The Karnaugh maps introduced next arranges the minterms of a function in such a way that it is easy to recognize visually how to combine terms to get minimum expression.

$f = \boxed{a + b} \longrightarrow$ SOP ✓ $(a) + (b)$

minimum SOP

$\| $

$f = (a\bar{b}) + (\bar{a}b) + (ab) \longrightarrow$ SOP ✓

Canonical SOP

$\|$

$f = a + b + ab \longrightarrow$ SOP ✓

K-map is used to get :

minimum sum-of-products expression for a function.

as well as

minimum product-of-sums expression for a function.

A minimum sum-of-products expression for a function is defined as a sum of product terms which

(a) has a minimum number of terms and

(b) of all those expressions which have the same minimum number of terms, has a minimum number of literals.

Unlike the minterm expansion(Canonical SOP) for a function, the minimum sum of products is not necessarily unique; that is, a given function may have two different minimum sum-of-products forms, each with the same number of terms and the same number of literals.

mSOP (minimum SOP)

$$\boxed{P_1 + P_2 + P_3 + P_4}$$

4 terms

mPOS (minimum POS)

$(S_1)(S_2)(S_3) \longrightarrow$ 3 terms

A minimum product-of-sums expression for a function is defined as a product of sum terms which

(a) has a minimum number of terms, and

(b) of all those expressions which have the same number of terms, has a minimum number of literals.

Unlike the maxterm expansion, the minimum product-of-sums form of a function is not necessarily unique. A given function may have two different minimum product-of-sum forms, each with the same number of terms and the same number of literals.

Canonical SOP

CPOS

unique

f

$$CSOP = \sum m(f=1) = \text{unique}$$

$$CPOS = \prod M(F=0) = \text{unique}$$

Minimization of Boolean Functions using Karnaugh Maps

Maurice Karnaugh 1953

Minimization

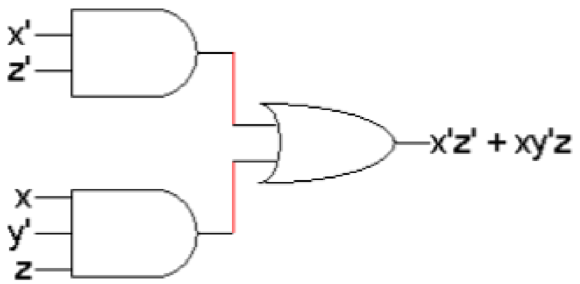
- Minimization can be done using
  - Boolean algebra
    - To combine terms

$$B\,\overline{C} + B\,C = B(\overline{C} + C) = B$$

- Or equivalently
  - Karnaugh maps
    - Visual identification of terms that can be combined

Karnaugh Maps

- Applications of Boolean logic to circuit design
  - The basic Boolean operations are AND, OR and NOT
  - These operations can be combined to form complex expressions, which can also be directly translated into a hardware circuit
  - Boolean algebra helps us simplify expressions and circuits
- Karnaugh Map: A graphical technique for simplifying an expression into a minimal sum of products (MSP) form:
  - There are a minimal number of product terms in the expression
  - Each term has a minimal number of literals
- Circuit-wise, this leads to a *minimal* two-level implementation



A Karnaugh map (K-map) is a pictorial method used to minimize Boolean expressions without having to use Boolean algebra theorems and equation manipulations.

A K-map can be thought of as a special version of a truth table, designed to ease the minimization.

Using a K-map, expressions with upto four variables are easily minimized.

a	b	f
0	0	
0	1	
1	0	
1	1	

=

b \ a	0	1
0	0	2
1	1	3

K-map

	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

=

c \ ab	00	01	11	10
0	0	2	6	4
1	1	3	7	5

k-map

The Map Method

Algebraic procedure to combine terms using the *Aa* + *Aa'* = *A* rule

Karnaugh map: modified form of truth table

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

(*a*) Location of minterms in a three-variable map.

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

(*b*) Map for function $f(x,y,z) = \sum(2,6,7) = yz' + xy$.

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

(*c*) Location of minterms in a four-variable map.

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

(*d*) Map for function $f(w,x,y,z) = \sum(4,5,8,12,13,14,15) = wx + xy' + wy'z'$.

Next Topic:

Idea Behind K-map

function f(a,b,c) → unique CSOP

Sum of minterms

Product term containing all variables

Karnaugh: starts with CSOP form.

$CSOP = m_i + m_j + m_k + \cdots$

Two minterms at a time

which two minterms you can Combine (Reduce) ??

$\overset{100}{m_4}, \overset{110}{m_6}$

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

$ac'(b+\bar{b})$

$= ac'$

$\overset{100}{m_4}, m_7 \rightarrow 111$

$a\bar{b}\bar{c} + abc$

Cannot be reduced.

$\boxed{abc + \bar{a}\bar{b}c}$

$\underbrace{m_i} + \underbrace{m_j}$

Combine/Reduce only if one variable changes.

$\bar{a}bc + abc$

bc

Variable which changes is Removed. ✓

$$abcd + a\bar{b}cd = acd$$

K-map: $f(a,b,c)$

$c \backslash ab$	00	01	11	10
0	000 m_0	010 m_2	110 m_6	100 m_4
1	001 m_1	011 m_3	111 m_7	101 m_5

Idea behind k-map:

Truth Table of any f is represented in such a way (in matrix format/ in a map format) that Adjacent cells/ minterms differ by only one variable.

$f(a,b)$

b \ a	0	1
0	00 m_0	10 m_2
1	01 m_1	11 m_3

$f(a,b,c)$

c \ ab	00	01	11	10
0	0	2	6	4
1	1	3	7	5

$110 = m_6$

$abc = \underline{011} = m_3$

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

order

cd \ ab	00	01	11	10
00	0000 m_0	0100 m_4	12	1000 m_8
01	0001 m_1	0101 m_5	13	m_9
11	0011 m_3	m_7	15	m_{11}
10	0010 m_2	m_6	14	m_{10}

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

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

$f(a,b,c)$ order

a \ bc	00	01	11	10
0	0	1	3	2
1	4	5	7	6

c \ ab	00	01	11	10
0	0	2	6	4
1	1	3	7	5

abc = 001

b \ ac	00	01	11	10
0	0	1	5	4
1	2	3	7	6

000

010 011 111

$f(a,b,c)$

c \ ab				
	0	2	6	4
	1	3	7	5

$f(a,b,c,d)$

cd \ ab				
	0	4	12	8
	1	5	13	9
	3	7	15	11
	2	6	14	10

Eg: Truth Table Representation of f

a	b	f
0	0	
0	1	
1	0	
1	1	

Adjacent minterms
Do not differ
by only one
variable

K-map Rep of f:

a \ b	0	1
0	00	01
1	10	11

The Map Method

Algebraic procedure to combine terms using the *Aa* + *Aa'* = *A* rule

Karnaugh map: modified form of truth table

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

(*a*) Location of minterms in a three-variable map.

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

(*b*) Map for function $f(x,y,z) = \sum(2,6,7) = yz' + xy$.

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

(*c*) Location of minterms in a four-variable map.

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

(*d*) Map for function $f(w,x,y,z) = \sum(4,5,8,12,13,14,15) = wx + xy' + wy'z'$.

Karnaugh Maps (K-map)

- Alternate representation of a truth table

To make minimization easy.

Adjacent cells differ by one variable only.

Karnaugh maps

- All functions can be expressed with a map

- There is one square in the map for each minterm in a function's truth table

X	Y	F
0	0	m0
0	1	m1
1	0	m2
1	1	m3

X \ Y	0	1
0	m0 $\overline{X}\,\overline{Y}$	m1 $\overline{X}Y$
1	m1 $X\overline{Y}$	m3 XY

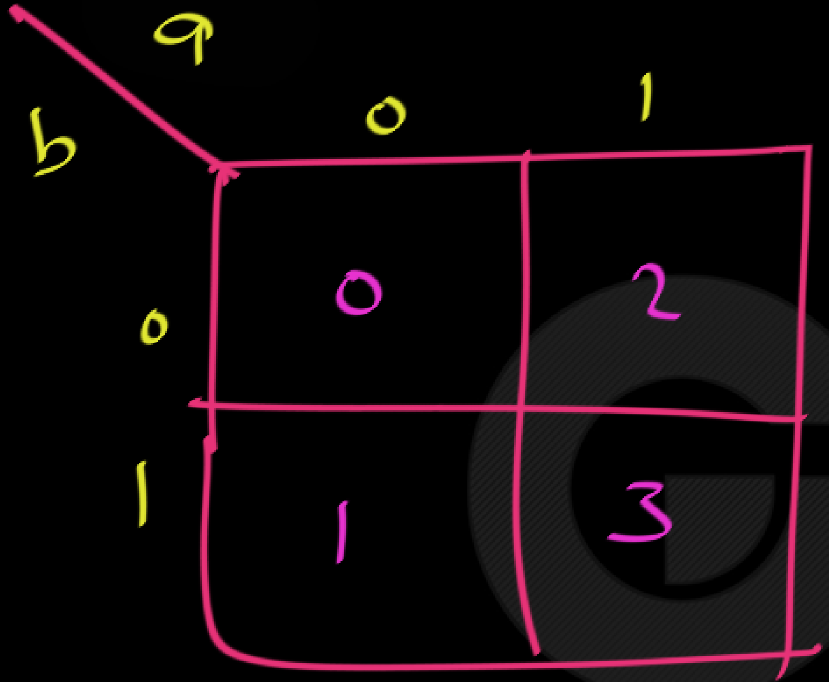
Cell in K-map $\equiv$ a minterm

Next Topic:

Adjacent Cells & Cubes in K-map

Adjacent minterms/cells in k-map:

Two cells/minterms (m_i, m_j) are Called Adjacent/Neighbours iff they differ by Exactly one variable.



(0,3) X

00 11

Adjacent cells:

(0,2) ✓

00 10

(1,2) X

01 10

(0,1) ✓

00 01

(2,3) ✓

10 11

$f(a,b,c) =$

c \ ab	00	01	11	10
0	0	2	6	4
1	1	3	7	5

Adjacent cells:

(0,2) ✓

000 010

(6,7) ✓

110 111

(0,6) ✗

000 110

(3,6) ✗

$f(a,b,c) =$

c \ ab	00	01	11	10
0	0	2	6	4
1	1	3	7	5

Adjacent cells:

(0,4) ✓

000 100

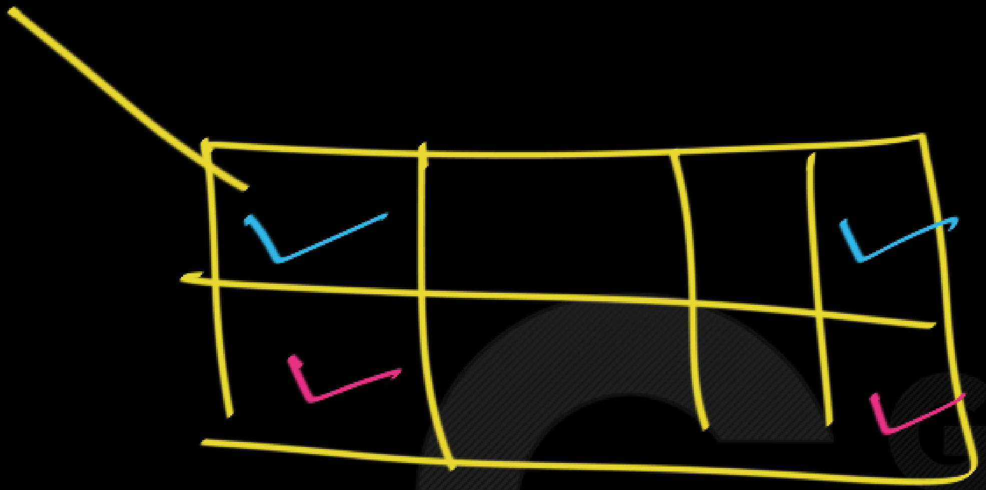
(1,5) ✓

001 101

(0,5) ✗

000 101

(1,4) ✗



Snake game in Nokia;

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

Adjacent Cells

(5,13) ✓

(5,7) ✓

(0,8) ✓

(1,9) ✓

(3,11) ✓

(2,10) ✓

(0,2) ✓

(4,6) ✓

(12,14) ✓

(8,10) ✓

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

Adjacent Cells

(8, 2) ✗

1000   0010

(0, 2) ✓   (12, 14) ✓

(4, 6) ✓   (8, 10) ✓

(0, 10) ✗

(0, 3) ✗

Cube in k-map:

Group of $\underbrace{2^m \text{ cells}}$ $(m \geq 0)$

such that Every cell is Adjacent to Exactly m cells. in the Group

b \ a	0	1
0	0	2
1	1	3

Cube :

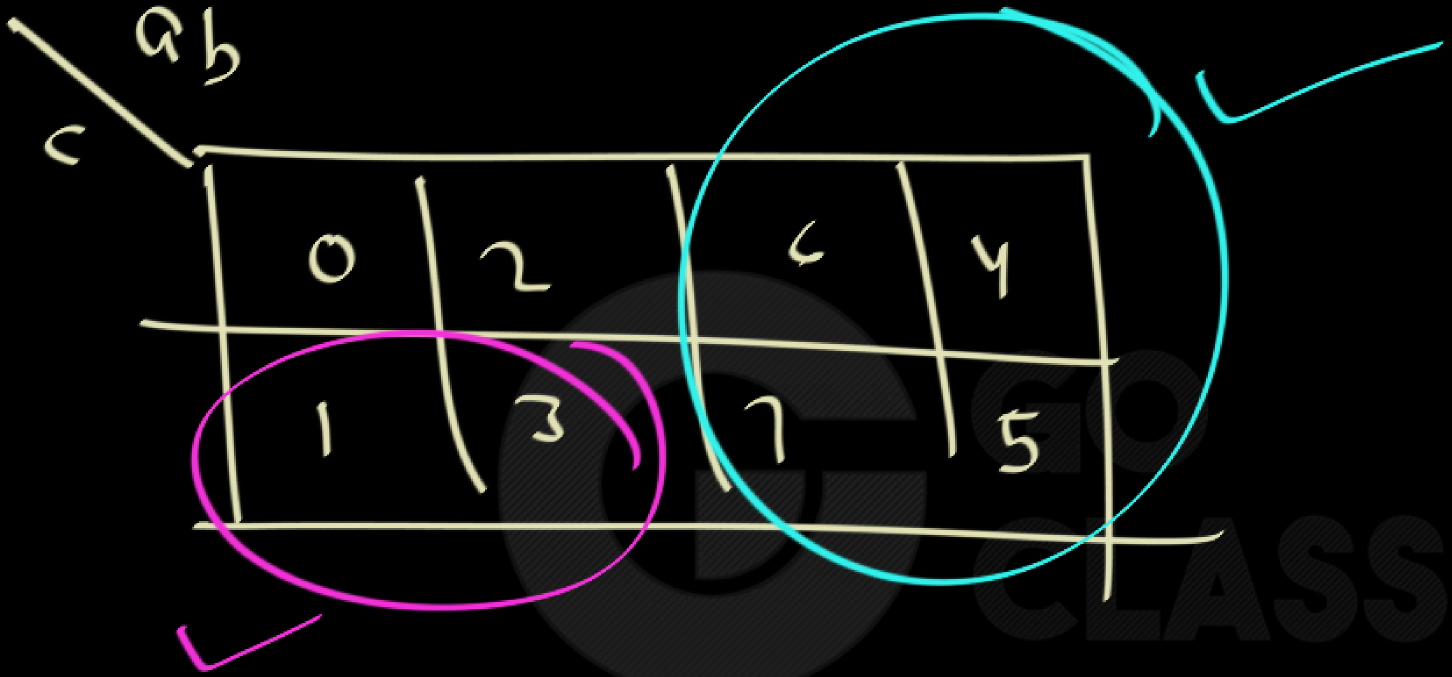
(0,2) ✓ → 2^1 cells

(0,3) → 2^1 cells ✗

(0,1) ✓

(0,1,2,3) ✓ → 2^2 cells

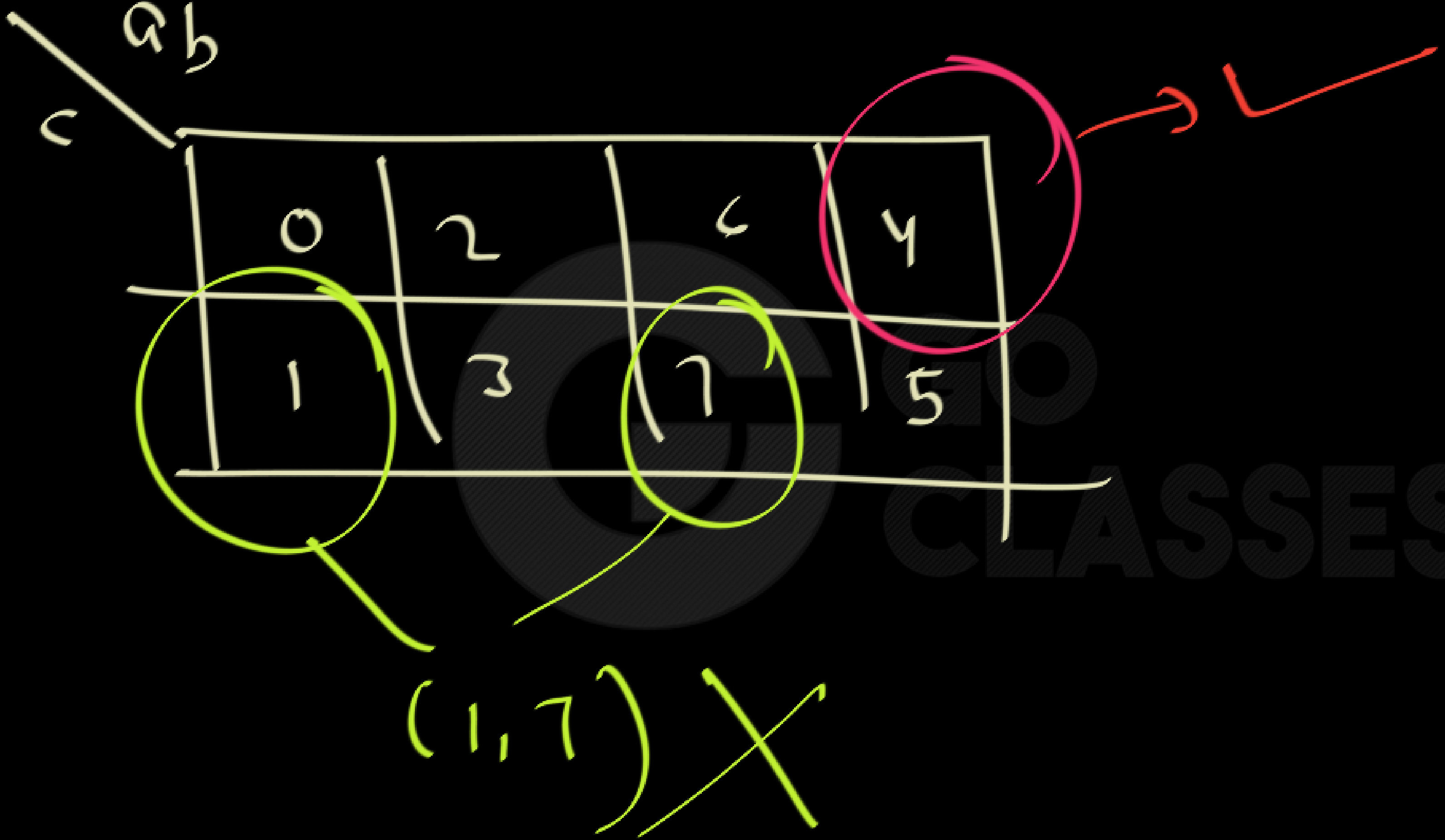
(0,2,1) ✗ → 2^m cells ✗

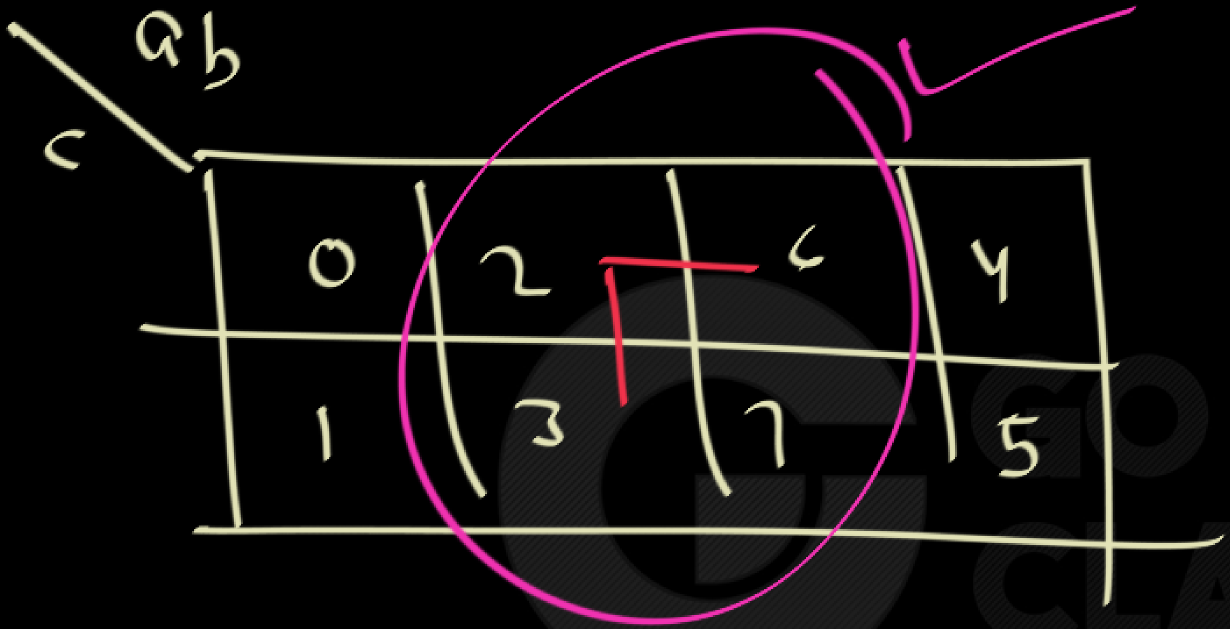


Cubes

(4, 5, 6, 7)

(1, 3)


ab
c
0
2
6
4
1
3
7
5
(1, 7)


ab
c
0
2
6
4
1
3
7
5

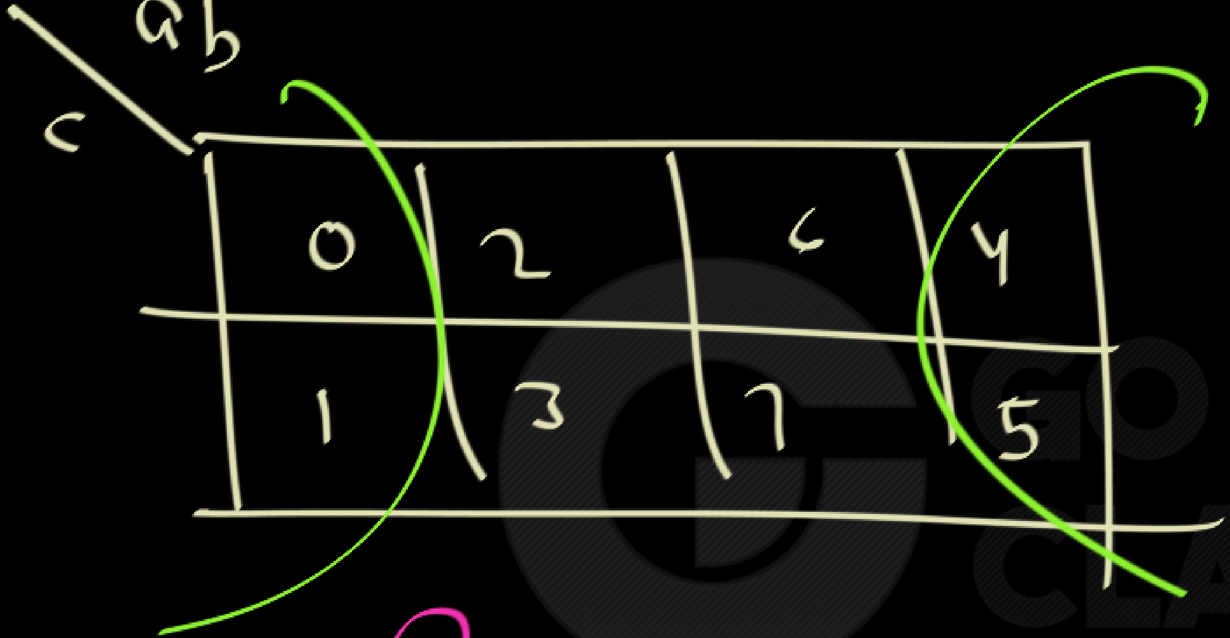
c \ ab				
	0	2	6	4
	1	3	7	5

(1, 5) ✓

(1,4) ✗

(0,1) ✓

(0, 2, 6) ✗



Cube

(0, 1, 4, 5)

2^2

cells

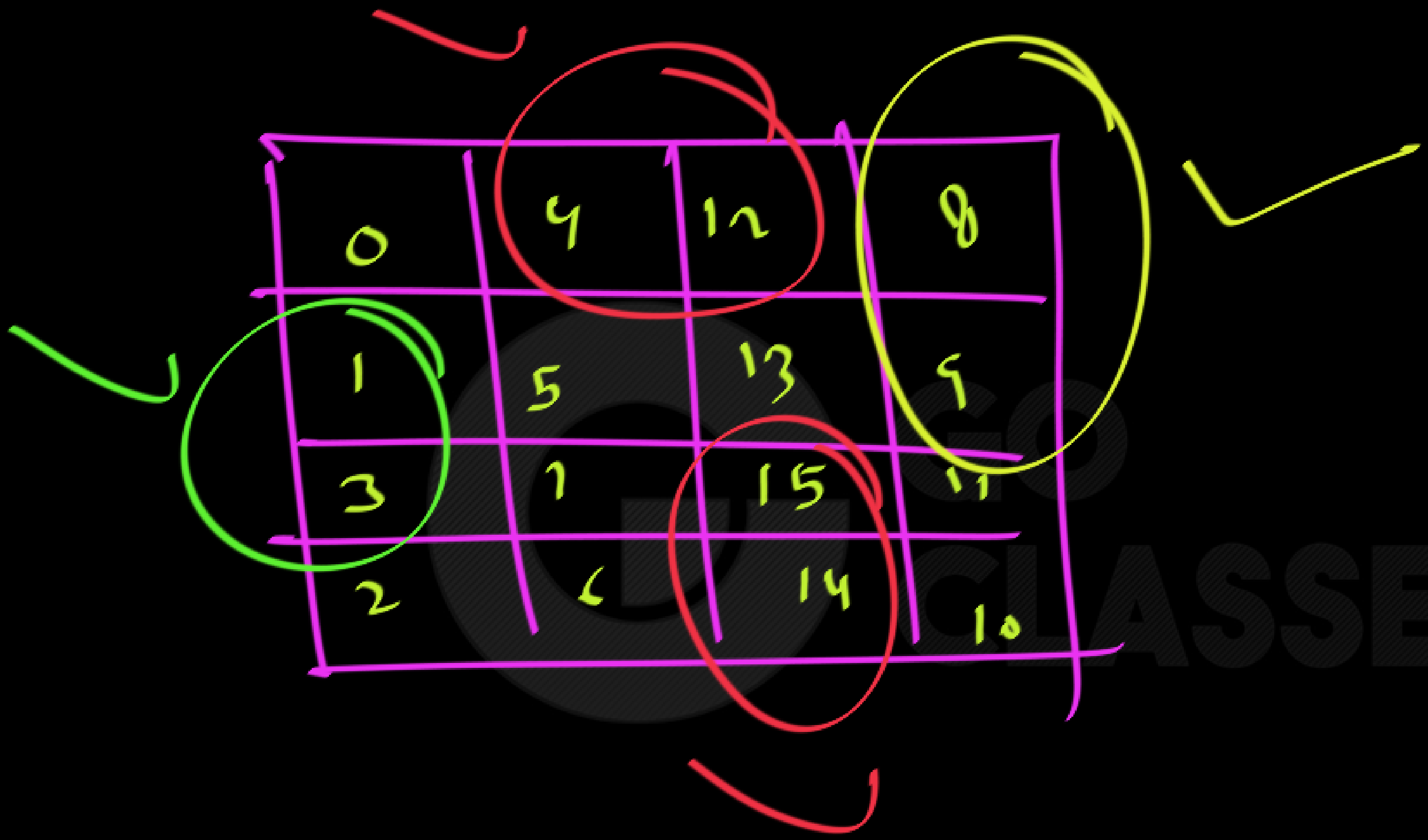
(0, 1, 6, 7) X

c \ ab				
	0	2	6	4
	1	3	7	5

$\rightarrow 2^3$ cells

	Neighbours
0	1, 2, 4
1	0, 3, 5
3	3, 1, 7
6	2, 4, 7

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


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

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

(12, 14) ✓

(12, 2) ✗

(8, 10) ✓

(0, 10) ✗

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

(0, 1, 8, 9)

(4, 12, 7, 6, 14, 15) ✗

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

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

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

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

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

(0, 2, 8, 10) ✓

2^2

cells

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

2^3 cells

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

(0, 1, 2, 3, 8, 9, 11, 10)

2^3

cells

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

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

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

0, 4, 1, 5

Next Topic:

WHY 2^m Cells in a Cube in K-map ??