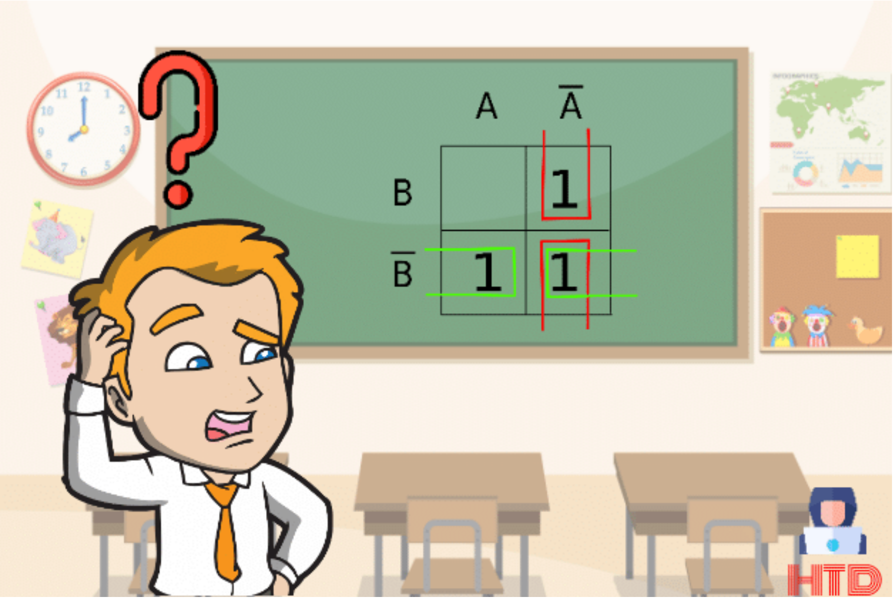
EE2000 Logic Circuit Design

Lecture 2 – Karnaugh Map and Quine-McCluskey (QM) Method



hackthedeveloper.com

What have we learnt so far?

1.1 Basic Logic Gates

1.2 Logic Circuit, Truth Table and Boolean Expression

1.3 Sum of Products vs Product of Sums and Canonical Form

1.4 Simplification using Boolean Algebra

- Based on experience

- Trial and error

Any systematic approach?

What will you learn?

2.1 What is Karnaugh map

2.2 Simplify a Boolean Function using Karnaugh map

2.3 Simplify a Boolean function with Don’t Care cases

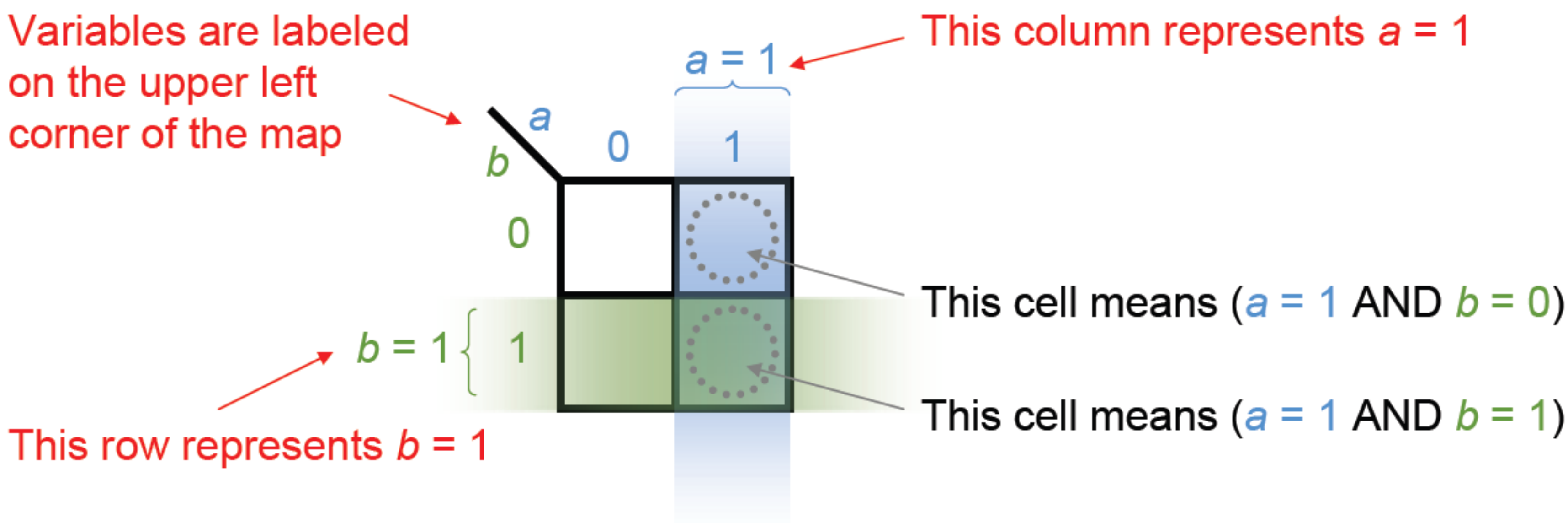
2.4 Simplify a Boolean Function using Quine-McCluskey method

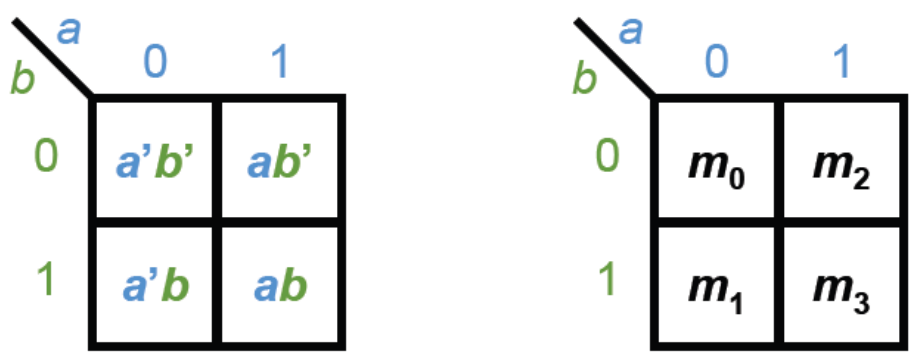
2.5 Simplify Boolean Functions (Multiple Outputs) by identifying common terms

2.1 Karnaugh Map

- In 1953, Maurice Karnaugh introduced a map method known as **Karnaugh map (K-map)**
- A straightforward procedure for minimizing Boolean functions in a tabular form
- Graphical representation of a truth table
- Minterm is used in the cell of the K-map
- *n*-variable function has 2^n cells:
    - Two-variable K-map has 4 cells
    - Three-variable K-map has 8 cells
    - Four-variable K-map has 16 cells

Two-variable K-map





Minterm representations

Plotting Functions in K-map

$f(a, b) = \Sigma m(0, 3)$ Canonical form (contains Minterm)

b \ a	0	1
0	1	0
1	0	1

or

b \ a	0	1
0	1	
1		1

Put a 0 or leave blank for those minterms not included in the function

Put a 1 in the corresponding cells

$f(a, b) = a'b' + ab'$ Function must be formed by Minterm

b \ a	0	1
0	1	1
1	0	0

or

b \ a	0	1
0	1	1
1		

Functions represented graphically with corresponding minterm cells labeled to value 1

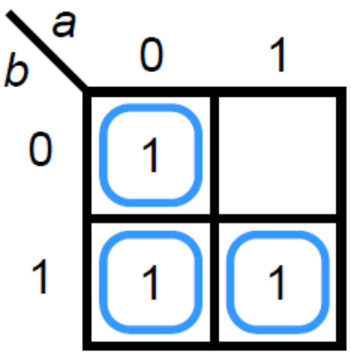
Example

Simplify $f(a, b) = \Sigma m(0, 1, 3)$

Adjacency: $a'b + ab = b$

Many ways to group them. Which is the best solution?

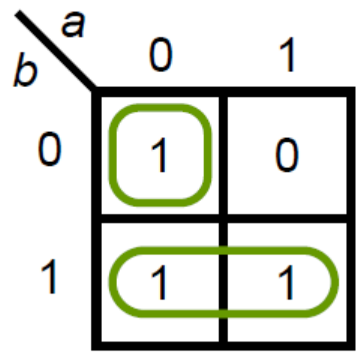
(i)



3 groups

$f = a'b' + a'b + ab$

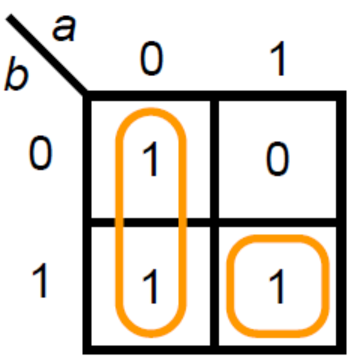
(ii)



2 groups

$f = a'b' + b$

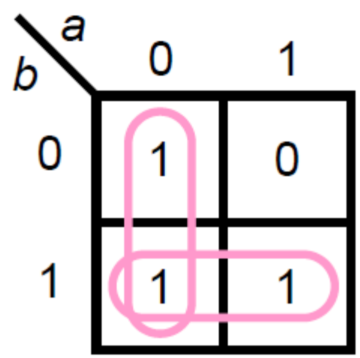
(iii)



2 groups

$f = a' + ab$

(iv)



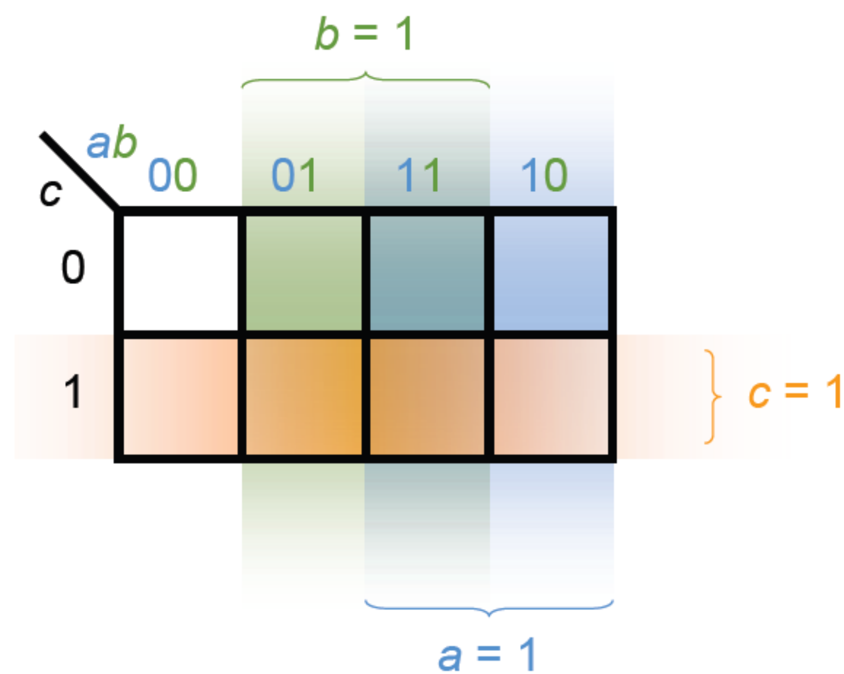
2 groups (the groups can be overlapped)

$f = a' + b$

Three-variable K-map

a=0 b=1

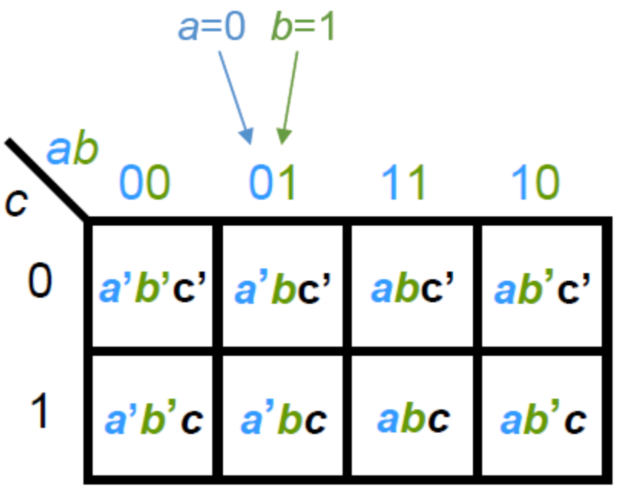
c \ ab	00	01	11	10
0	a'b'c'	a'bc'	abc'	ab'c'
1	a'b'c	a'bc	abc	ab'c



c \ ab	00	01	11	10
0	m_0	m_2	m_6	m_4
1	m_1	m_3	m_7	m_5

- Cells are not organized in numerical order.
- Only 1-bit difference between two adjacent cells.
- Why?

Gray code in K-map



- Adjacent cells have 1-bit (1-variable) difference only

  $a'bc'$ and abc'

- Forming a pair of adjacent binary combinations with neighboring cells.

  $a'bc' + abc' = bc'(a' + a) = bc'$

Rule of thumb: Group adjacent 1's on the map to form the simplified product terms.

Simplification of Product Terms

c \ ab	00	01	11	10
0	m_0	m_2	m_6	m_4
1	m_1	m_3	m_7	m_5

Example: Simplify $f(a, b, c) = \Sigma m(6, 7)$

c \ ab	00	01	11	10
0			1	
1			1	

This group contains both 0 and 1 for *c* (i.e. no longer depends on *c*, depends on *a* and *b* only)

Using Boolean Albegra:

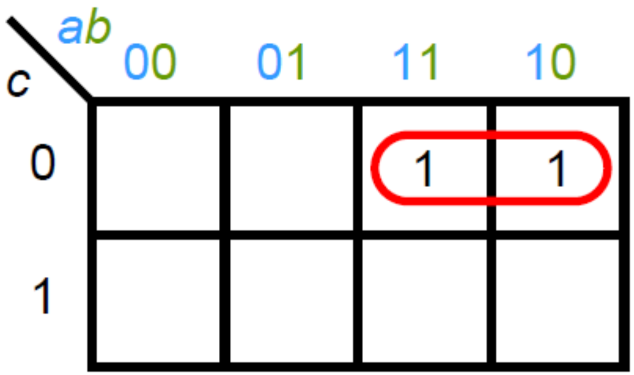
Only one-variable difference

$f(a, b, c) = \overbrace{abc' + abc}$

$= ab$ (adjacency)

Whenever we group two adjacent cells on the map, they can form a product term with one less variables!

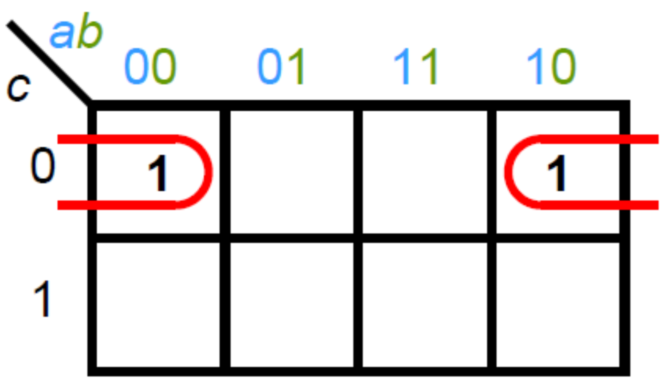
More Examples



$f(a, b, c) = abc' + ab'c'$

$= ac'$ *(adjacency)*

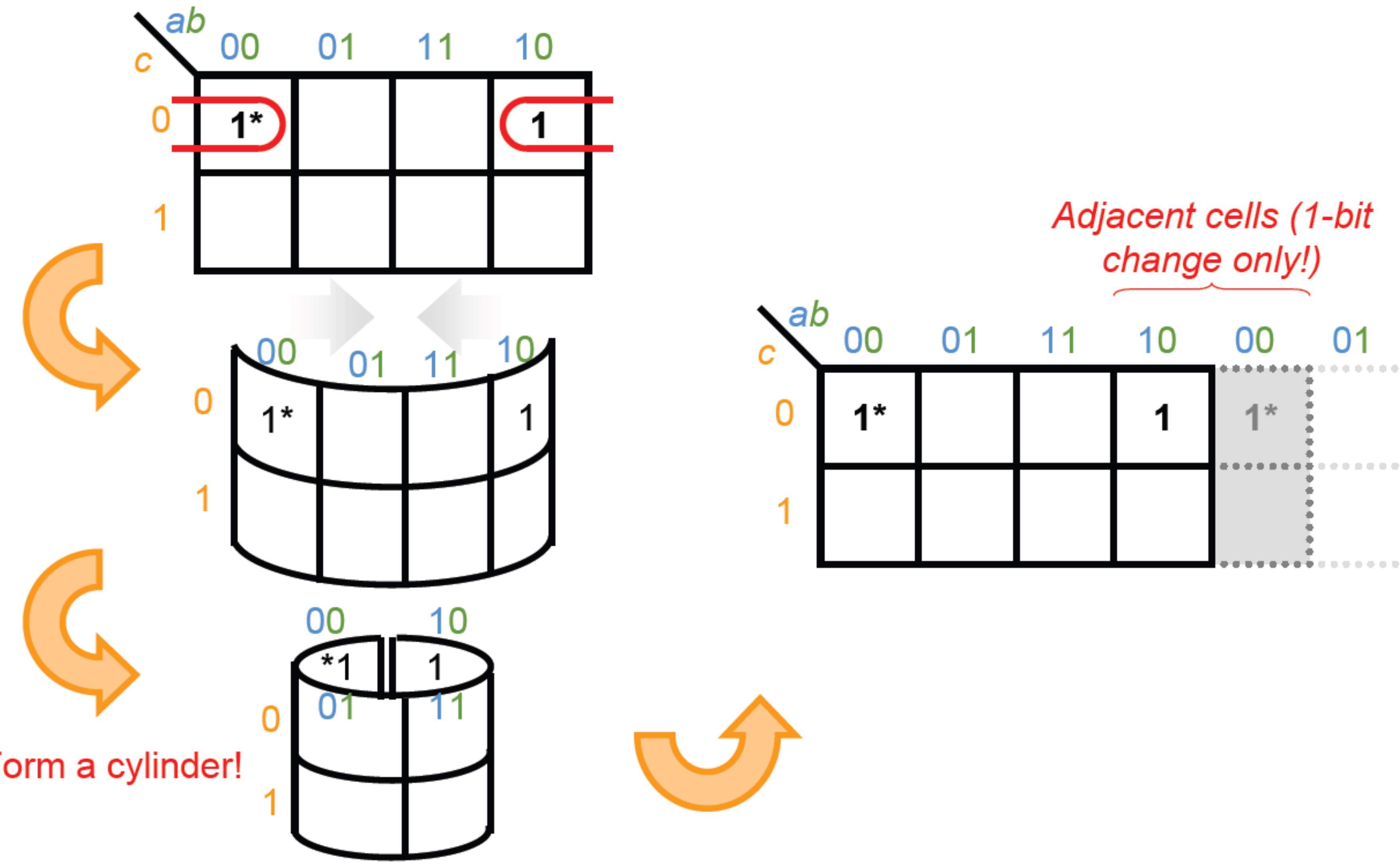
We can even group adjacent 1's across the edges:



$f(a, b, c) = \overbrace{a'b'c' + ab'c'}^{\text{Also one-variable difference!}}$

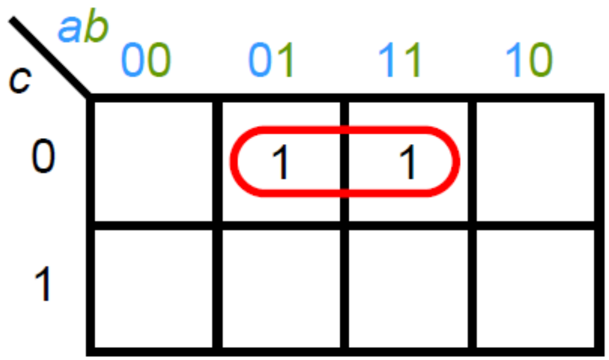
$= b'c'$ *(adjacency)*

Wrap Around Adjacency



What If We do not use Gray Code?

We can group these two adjacent 1's:

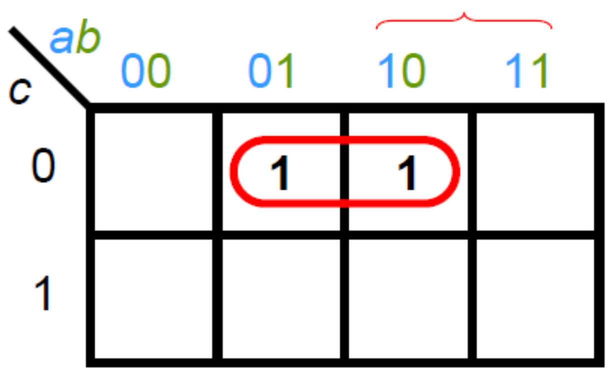


$f(a, b, c) = a'bc' + abc'$

$= bc'$

But not these two:

Incorrect arrangement



two-variable difference!

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

$= c'\,(a'b + ab')$

Cannot be simplified into a single term!

Horizontal vs Vertical K-map

Label rows with first variable, columns with the others

$a \backslash bc$	00	01	11	10
0	m_0	m_1	m_3	m_2
1	m_4	m_5	m_7	m_6

Vertical orientation of three-variable K-map

$bc \backslash a$	0	1
00	m_0	m_4
01	m_1	m_5
11	m_3	m_7
10	m_2	m_6

Although there are different ways drawing the K-Maps, we use the same method to group the adjacent 1's!

Four-variable K-map

$b = 1$

cd \ ab	00	01	11	10
00	a'b'c'd'	a'bc'd'	abc'd'	ab'c'd'
01	a'b'c'd	a'bc'd	abc'd	ab'c'd
11	a'b'cd	a'bcd	abcd	ab'cd
10	a'b'cd'	a'bcd'	abcd'	ab'cd'

$d = 1$ (rows 01, 11)

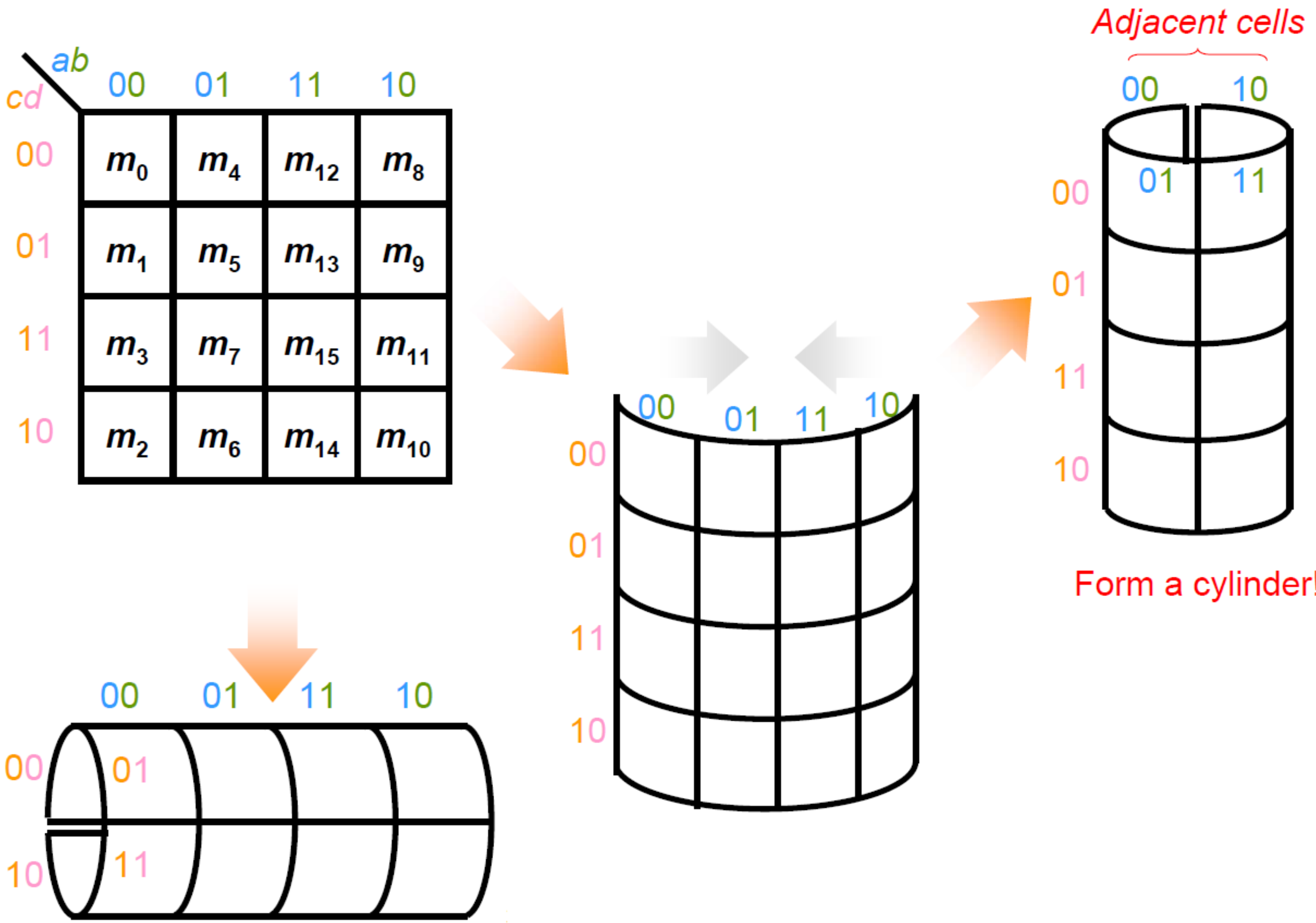
$c = 1$ (rows 11, 10)

$a = 1$

cd \ ab	00	01	11	10
00	m_0	m_4	m_{12}	m_8
01	m_1	m_5	m_{13}	m_9
11	m_3	m_7	m_{15}	m_{11}
10	m_2	m_6	m_{14}	m_{10}

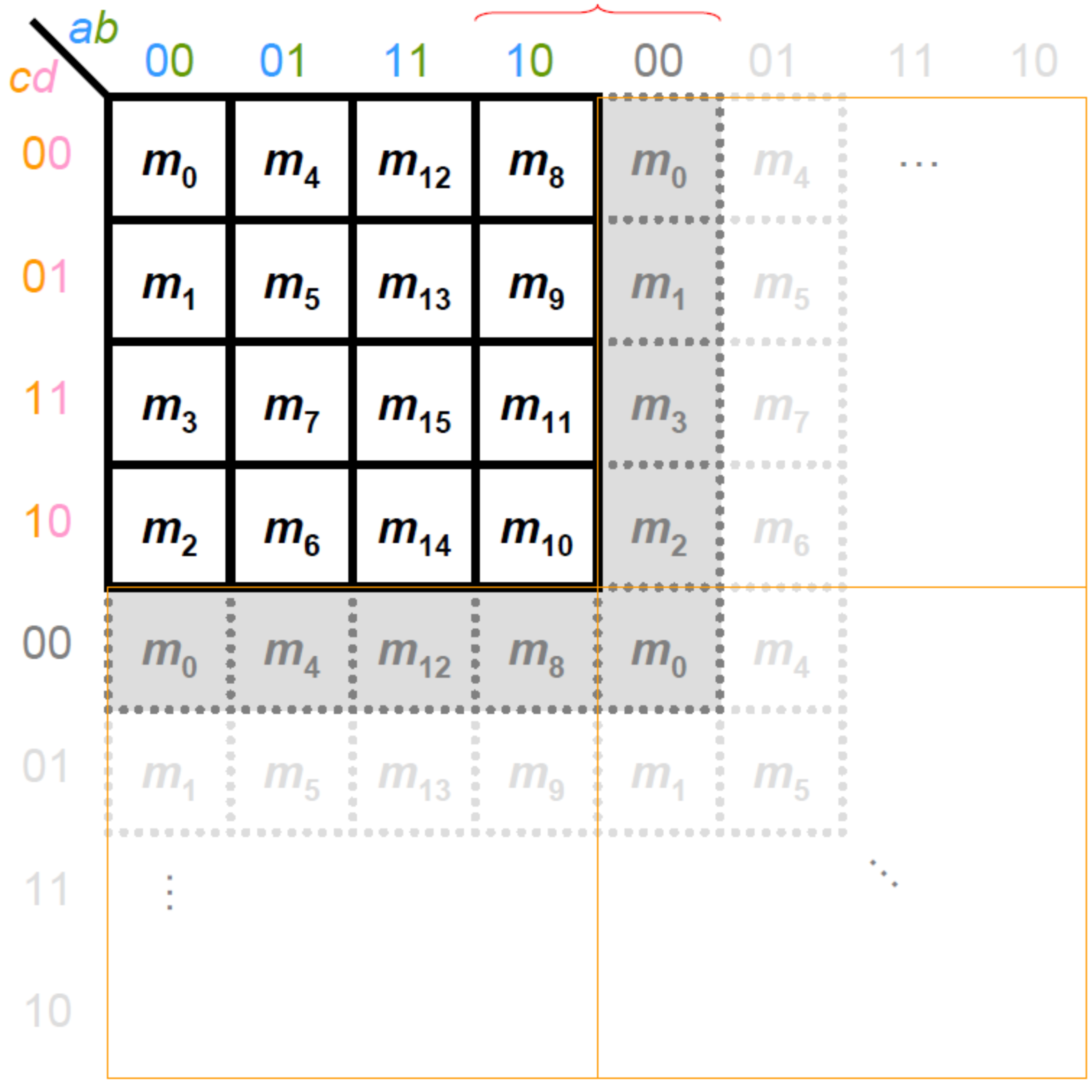
Note the Gray code order of the rows and columns

Wrap-around Adjacency



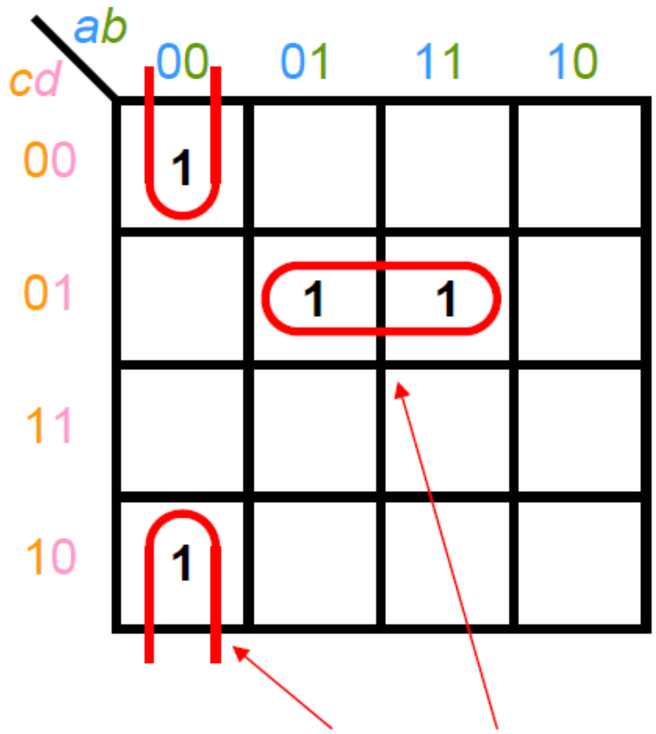
Imagine the Map as…

1-bit (1-variable) change only for every adjacent cells!

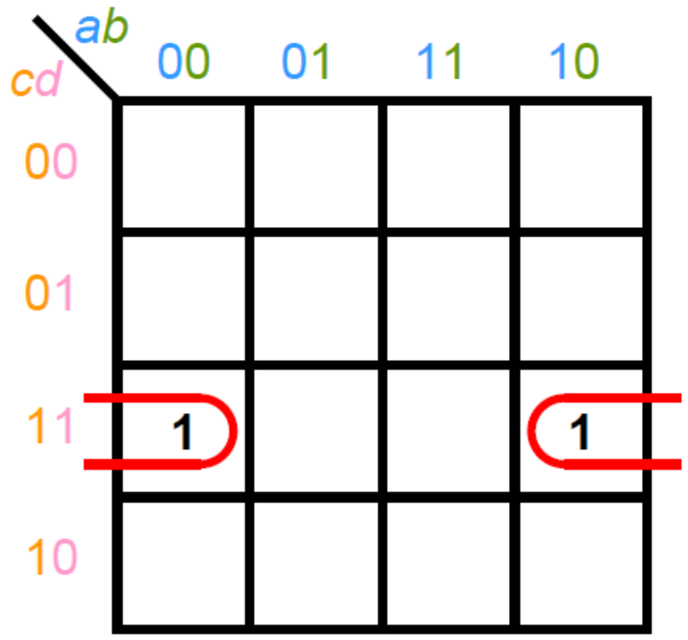


Types of Group Size

Group of 2 adjacent cells



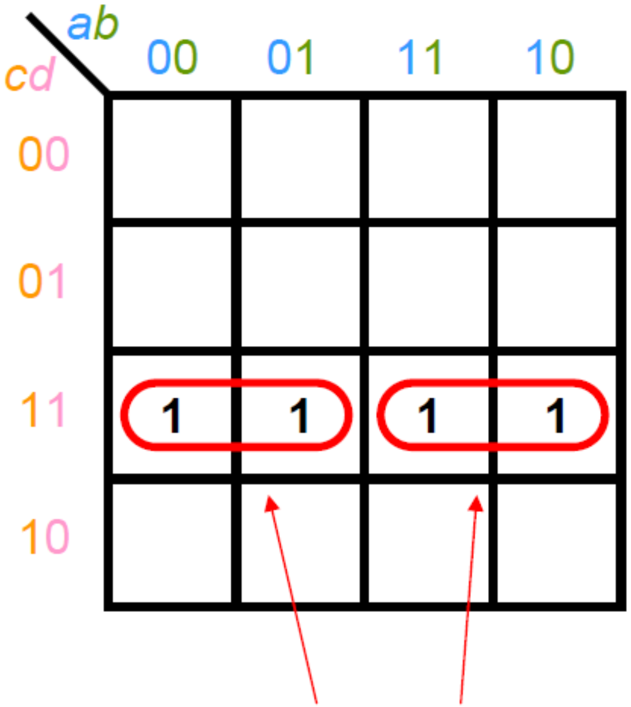
f(a, b, c, d) = a'b'd' + bc'd



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

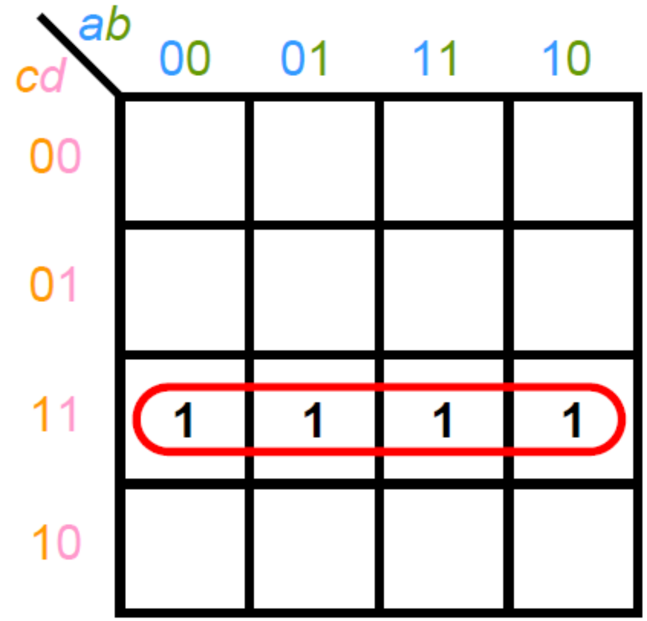
Types of Group Size

Group of 4 adjacent cells



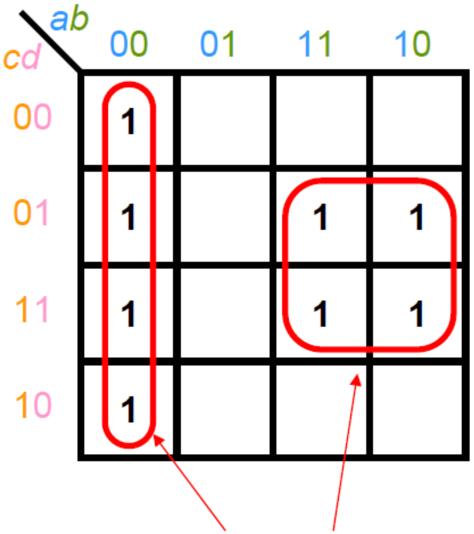
f(*a*, *b*, *c*, *d*) = *a'cd* + *acd*

= *cd*

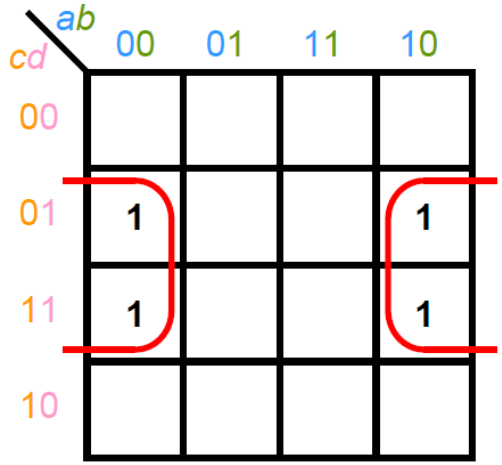


f(*a*, *b*, *c*, *d*) = *cd*

Types of Group Size

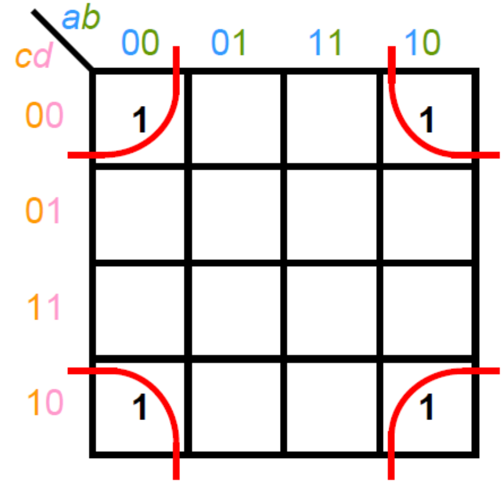


$f(a, b, c, d) = a'b' + ad$



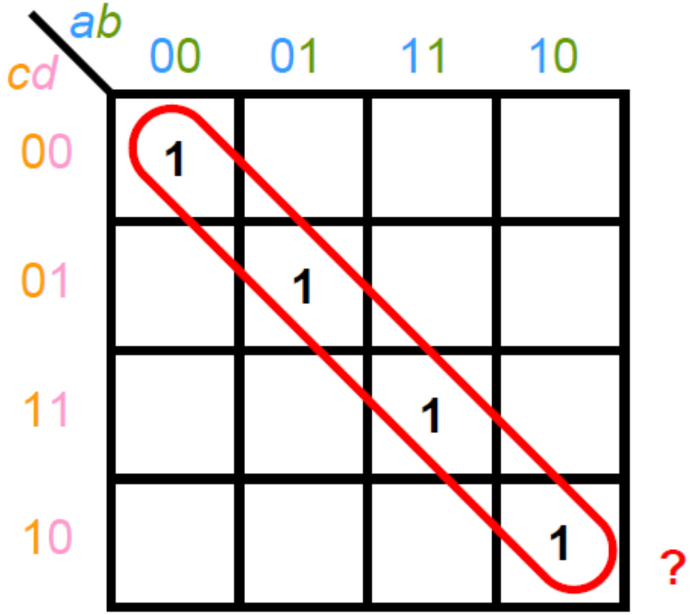
$f(a, b, c, d) = b'd$

Across 4 corners:

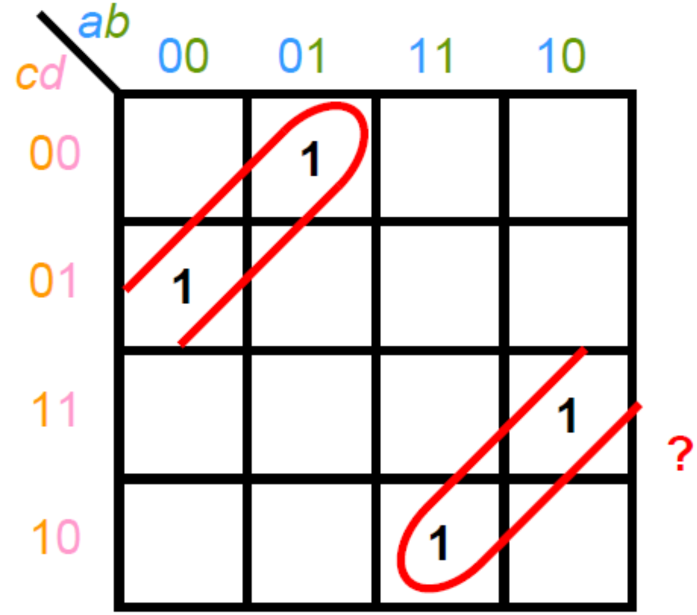


$f(a, b, c, d) = b'd'$

Are These Adjacent Cells?



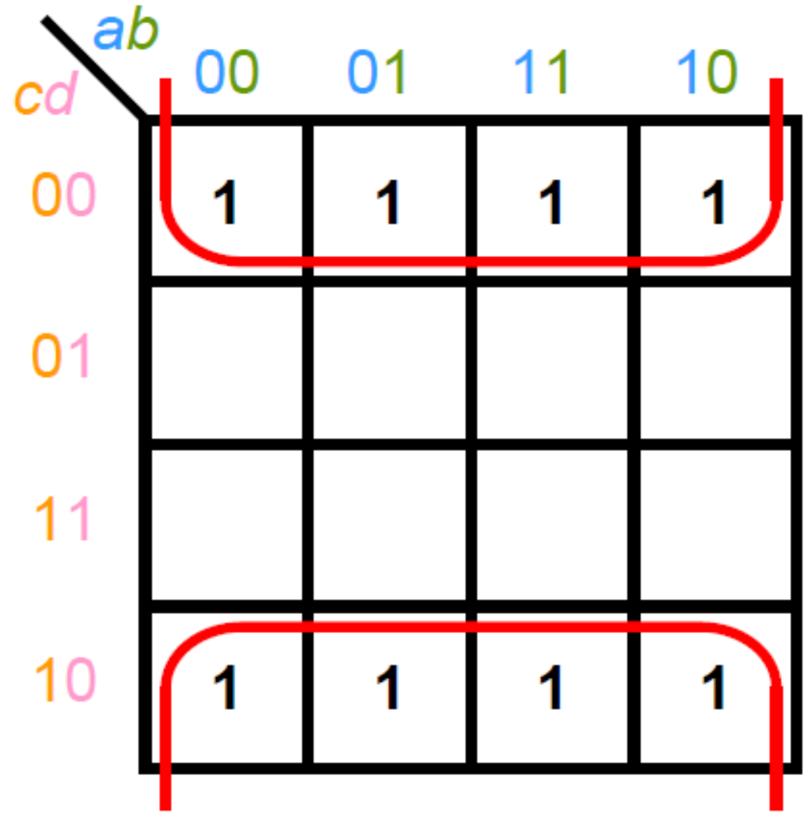
Diagonal?



Magic square?

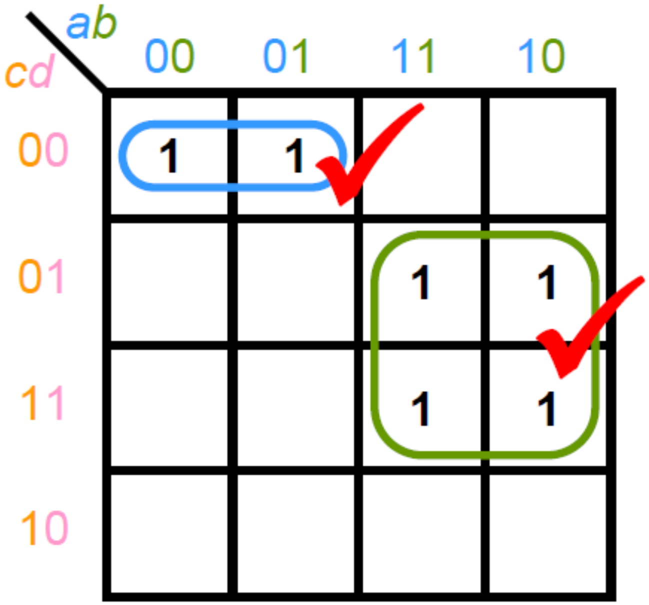
Types of Group Size

Group of 8 adjacent cells

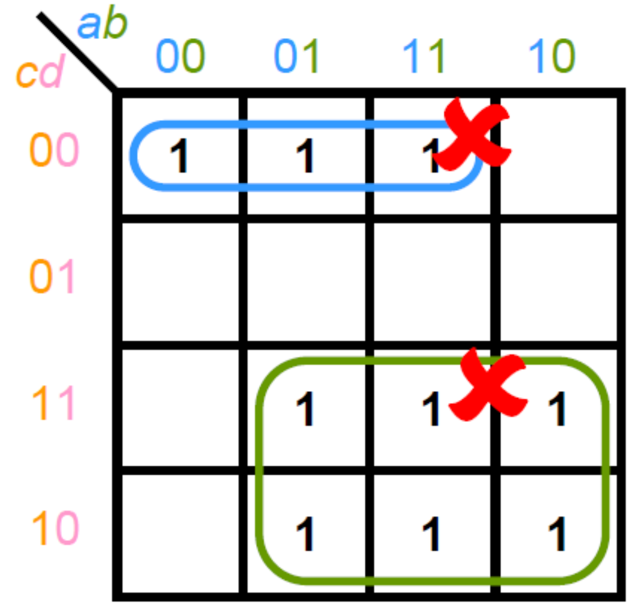


$f(a, b, c, d) = \ d'$

Summary



Group size is power of 2 (e.g. 2, 4, 8)



Other group size is illegal

- Booleans function to be minimized by K-map are always in Canonical SOP or POS (will discuss later) form
- Arrange cells in 1-bit difference
- Group adjacent cells in group size of 2^n, e.g. 2, 4, 8
- Apply adjacency law

2.2 Minimization using Karnaugh Map

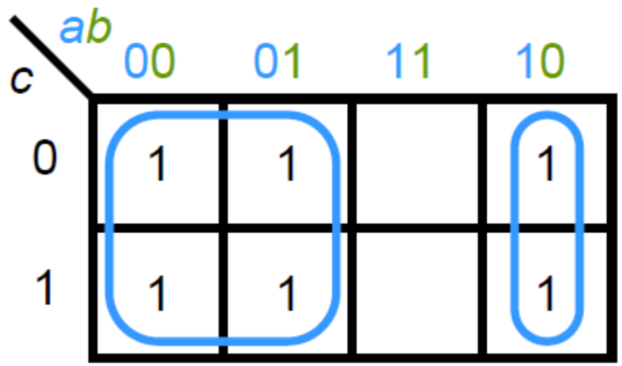
- Group adjacent cells in group size of 2^n, e.g. 2, 4, 8
- Rules:
  1. Find the fewest groups that can cover all cells marked with 1s.
  2. The groups should be as large as possible.
- Goal:
  1. Reduce the number of product terms to minimum
  2. Save the cost

Example

Simplify $f(a, b, c) = \Sigma m(0, 1, 2, 3, 4, 5)$

Which solution is better?

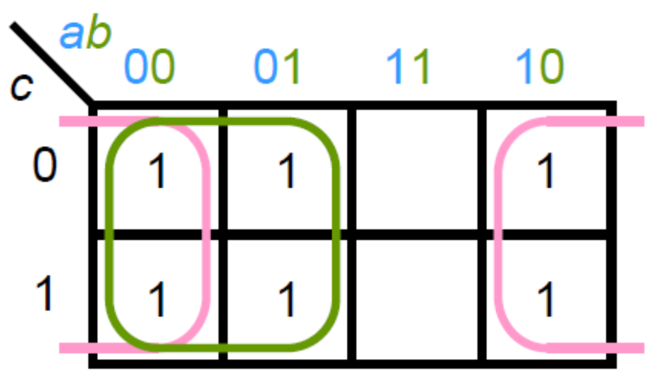
(i)



2 groups

$f(a, b, c) = a' + ab'$

(ii)



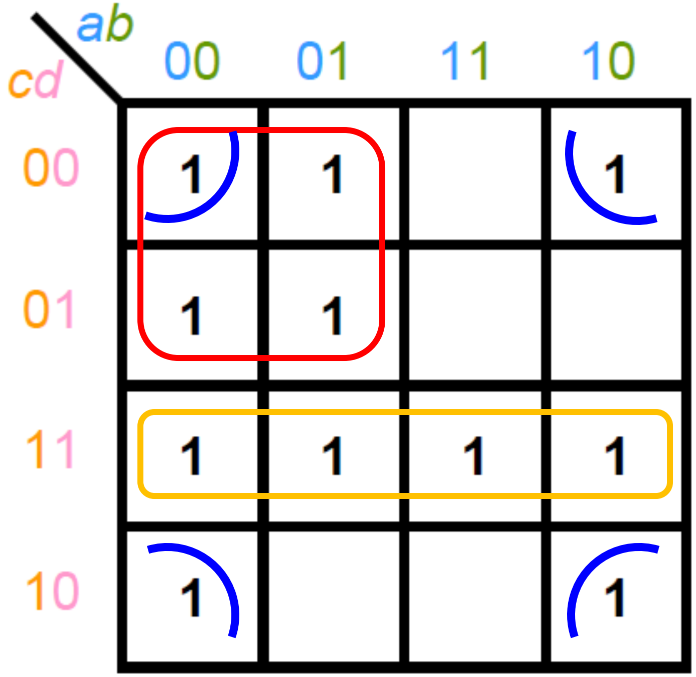
2 groups

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

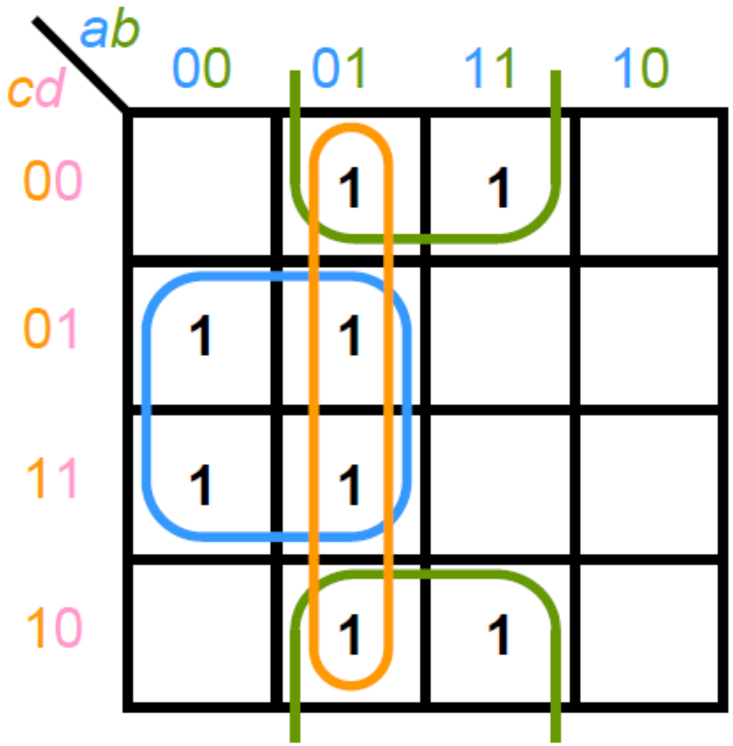
Groups should be as large as possible!

Example

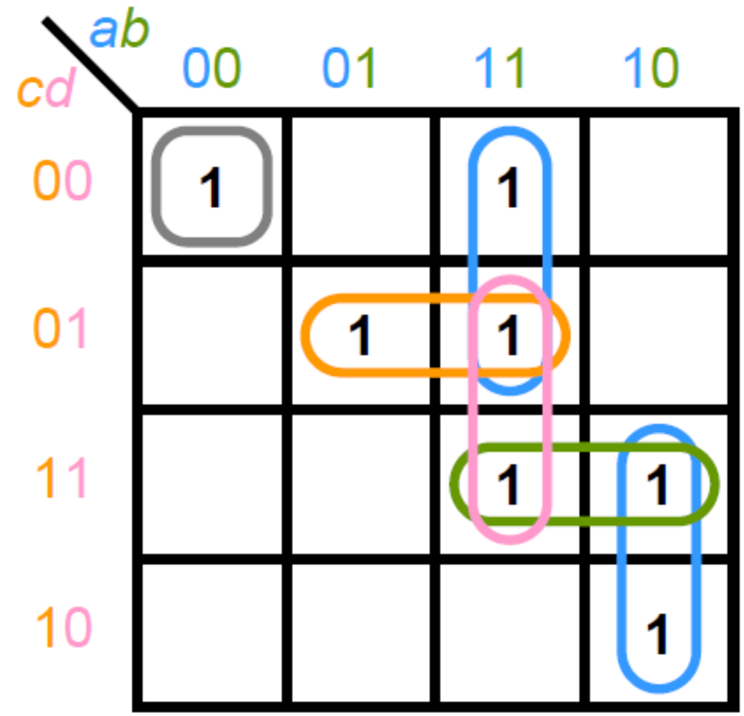
Simplify $f(a, b, c, d) = \Sigma m(0, 1, 2, 3, 4, 5, 7, 8, 10, 11, 15)$



Redundant Grouping



Three groups overlapped!

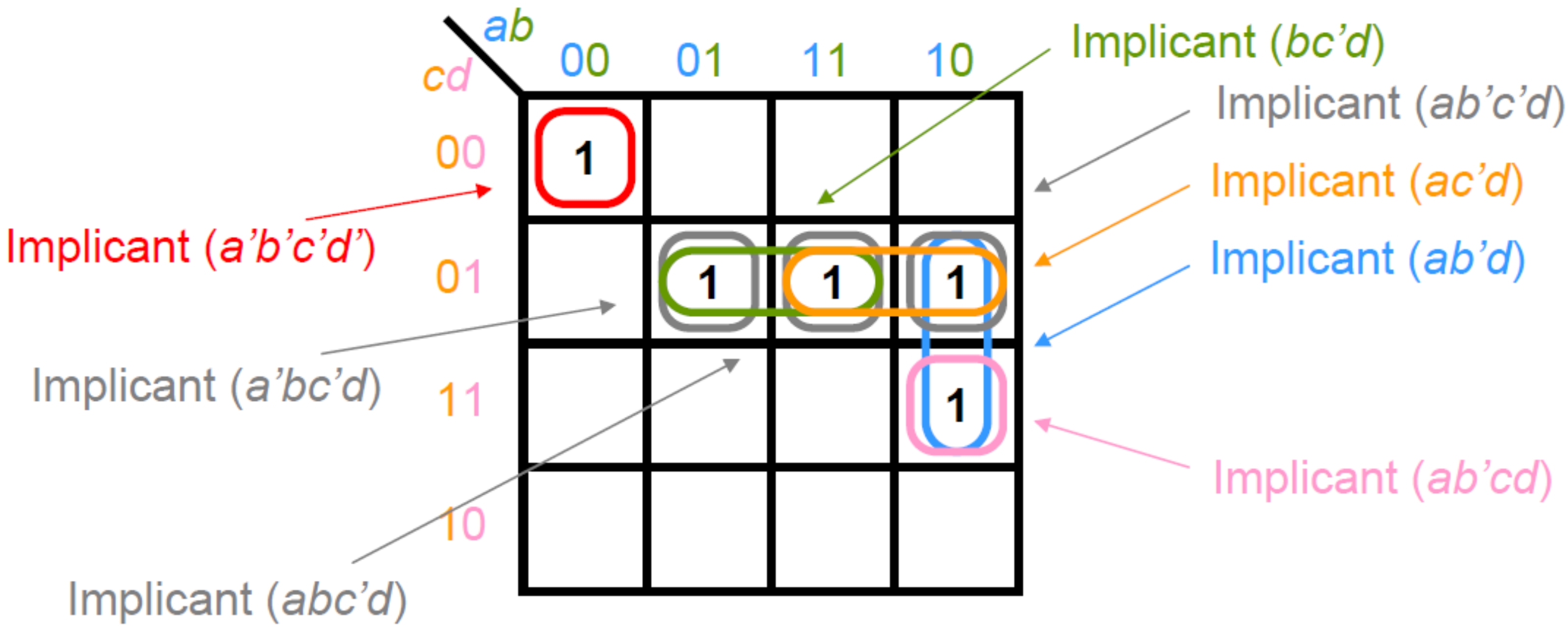


Too many overlaps!

Terminology

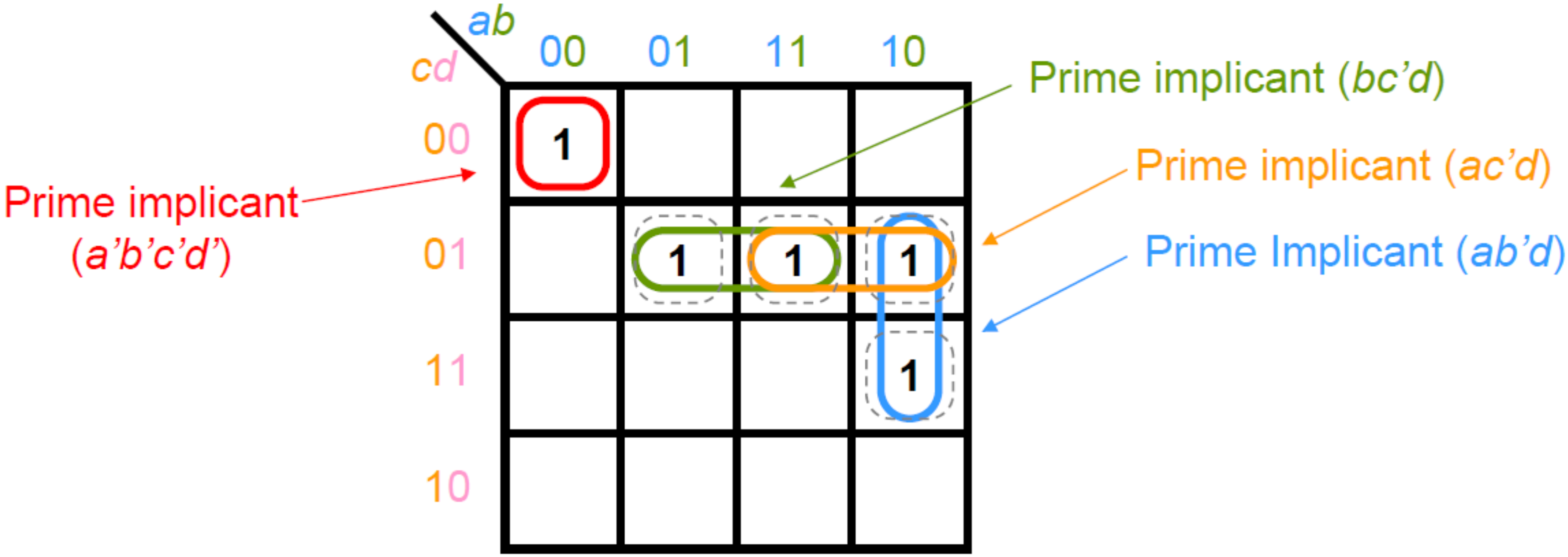
Implicant: A product term is an implicant such that the function is 1 whenever the product term is 1.

- In K-map, an implicant is a rectangle of 1, 2, 4, 8,… of 1's.



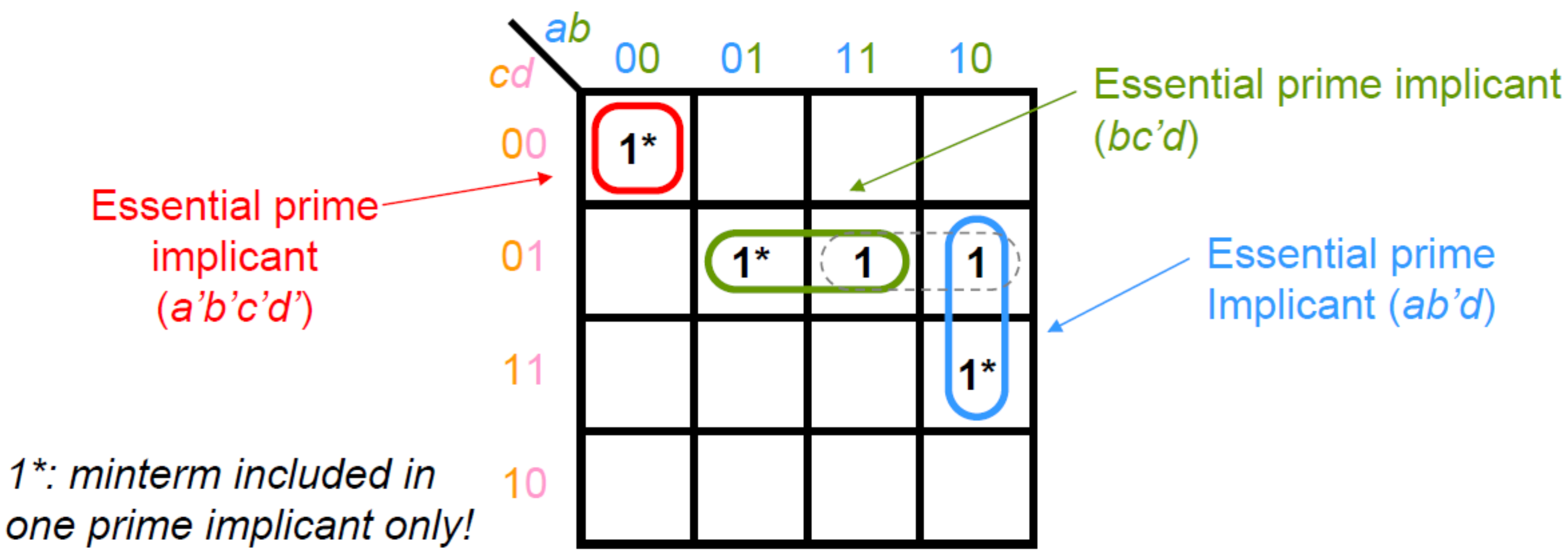
Terminology

Prime Implicant: An implicant that is not fully contained in any one other implicant.



Terminology

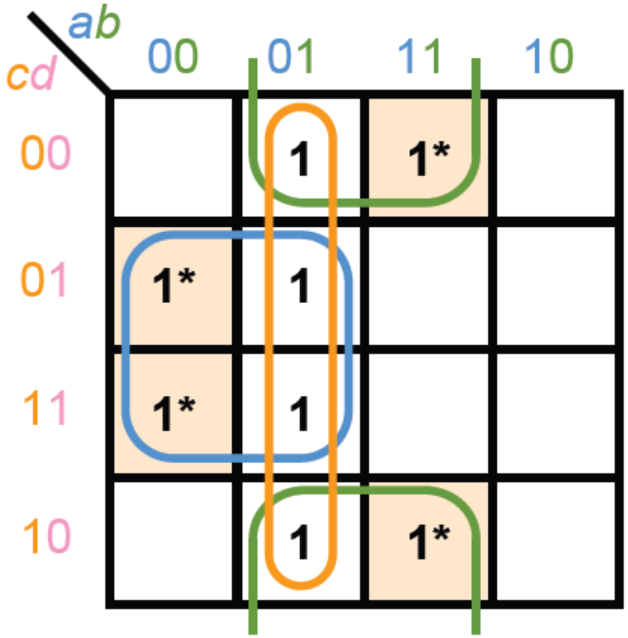
Essential Prime Implicant: If a minterm is included in only one prime implicant, that prime implicant is essential prime implicant.

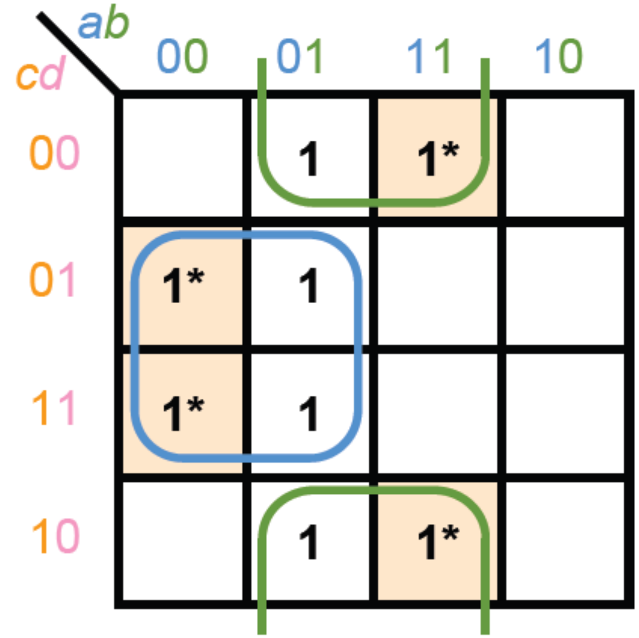


Systematic Approach

- Rules:
  1. Find the fewest groups that can cover all cells marked with 1s.
  2. The groups should be as large as possible.
- Approach:
  1. Determine all PIs.
  2. Select EPIs.
  3. Add PI to include the remaining minterm.

Previous Example

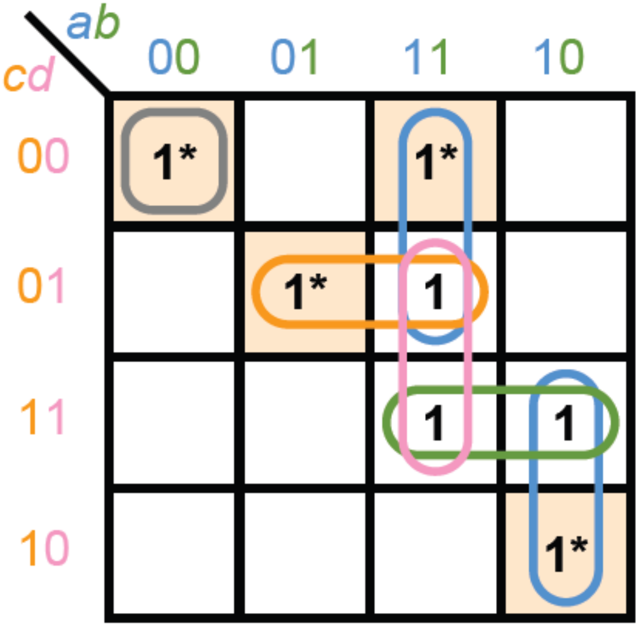


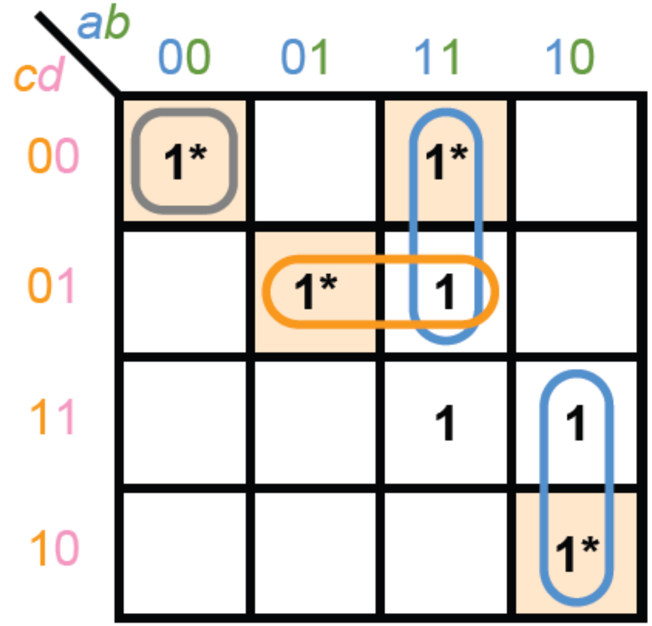


Select the essential prime implicants and no remaining minterms left!

$f(a, b, c, d) = a'd + bd'$

Previous Example





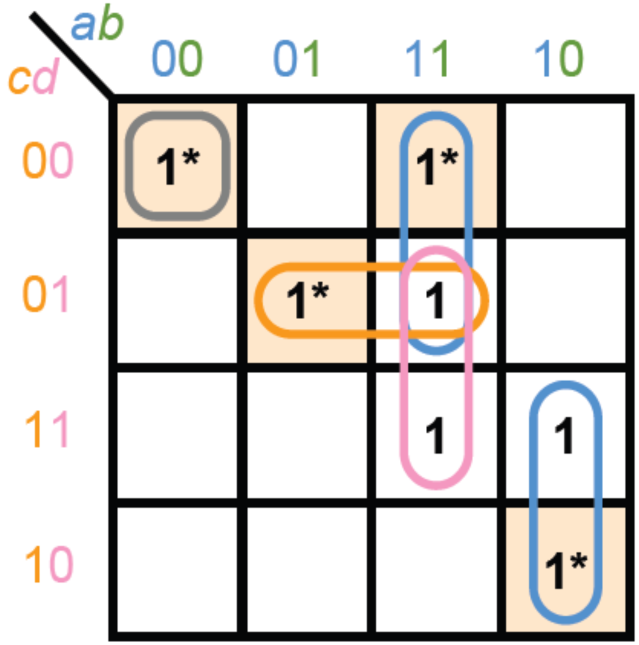
PIs: 

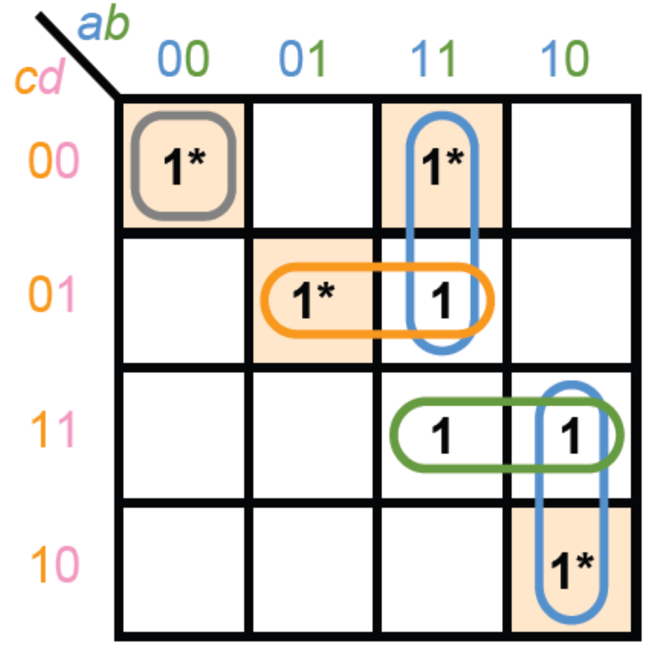
EPIs: 

Select the essential prime implicants first

Still have a remaining minterm!

Previous Example





f(*a*, *b*, *c*, *d*) = *a'b'c'd'* + *abc'* + *ab'c* + *bc'd* + *abd*

or

f(*a*, *b*, *c*, *d*) = *a'b'c'd'* + *abc'* + *ab'c* + *bc'd* + *acd*

We can choose either or

Exercise

1. Identify all PIs.
2. Select all EPIs.
3. Add PIs of remaining minterms.

cd \ ab	00	01	11	10
00	1	1	1	1
01				1
11		1		1
10	1	1	1	

Exercise

1. Identify all PIs.
2. Select all EPIs.
3. Add PIs of remaining minterms.

cd \ ab	00	01	11	10
00	1	1	1	1
01				1
11		1		1
10	1	1	1	

cd \ ab	00	01	11	10
00	1	1	1	1
01				1
11		1		1
10	1	1	1	

Exercise

Find all minimum sum of products expressions for the following K-map.

cd \ ab	00	01	11	10
00	1	1		1
01		1	1	1
11		1	1	
10	1	1		1

Exercise

cd \ ab	00	01	11	10
00	1	1		1
01				
11		1		1
10	1			

cd \ ab	00	01	11	10
00	1	1		1
01		1	1	1
11		1	1	
10	1	1		1

How about POS?

[Step 1] Group the 0s to obtain the complement of the f in SOP form

[Step2] Apply DeMorgan's Theorem to find f in POS form OR express them in Maxterm

Example

Simplify $f(a, b, c, d) = \Sigma m(0, 1, 2, 5, 8, 9, 10)$ in POS form

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

Fill the 1s and 0s into the map

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

Group the 0s using the same procedure as grouping the 1s

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

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

2.3 Boolean Functions with Don’t Care Cases

The output of Boolean functions are **incompletely specified functions**,

- For some input conditions, the outputs are unspecified
- Input condition has no effects to the function
- Output values are defined as **don’t Care**
- Don’t Care term can be minterm / maxterms
- Don’t Care term indicates by an ×, *d*, ϕ or φ

Truth Table with Don't Care

a	b	f
0	0	0
0	1	1
1	0	1
1	1	X

What the table says is:

f is 0 if	(a = 0 AND b = 0)
f is 1 if	(a = 0 AND b = 1), or
	(a = 1 AND b = 0)
f can be 0 or 1 if	(a = 1 AND b = 1)

$f(a, b) = \Sigma m(1, 2) + \Sigma d(3)$

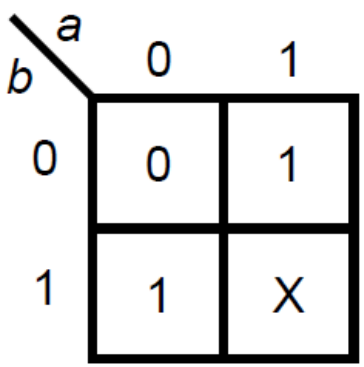
a	b	f_1	f_2
0	0	0	0
0	1	1	1
1	0	1	1
1	1	0	1

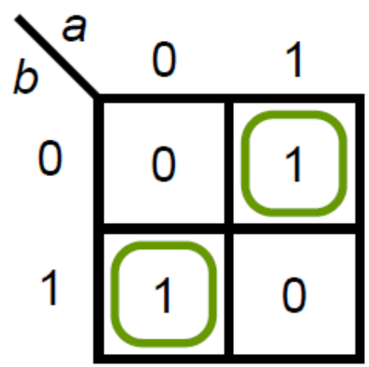
Both f_1 or f_2 of table on the left are acceptable

$$\boxed{f(a,b) = \Pi M(0)\Pi d(3)}$$

Don't Care term in K-map

Which solution is better?

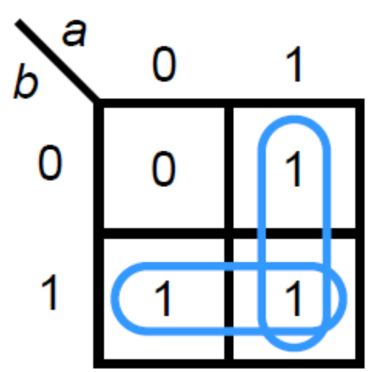




f_1 implementation

2 groups

$f = a'b + ab'$



f_2 implementation

2 groups

$f = a + b$

Is it a Good Solution?

Simplify $f(a, b, c, d) = \Sigma m(1, 3, 7, 11, 15) + \Sigma d(0, 2, 5)$

cd \ ab	00	01	11	10
00	X	0	0	0
01	1	X	0	0
11	1	1	1	1
10	X	0	0	0

cd \ ab	00	01	11	10
00	X	0	0	0
01	1	X	0	0
11	1	1	1	1
10	X	0	0	0

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

Other Solutions

1. Identify PIs that must include all 1s but don't care term × is optional.
2. Use × when possible to create larger group size.
3. Select the EPIs first , then remaining PIs

cd \ ab	00	01	11	10
00	X	0	0	0
01	1	X	0	0
11	1	1	1	1
10	X	0	0	0

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

cd \ ab	00	01	11	10
00	X	0	0	0
01	1	X	0	0
11	1	1	1	1
10	X	0	0	0

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

Choose to include those Xs that give largest PIs

Exercise

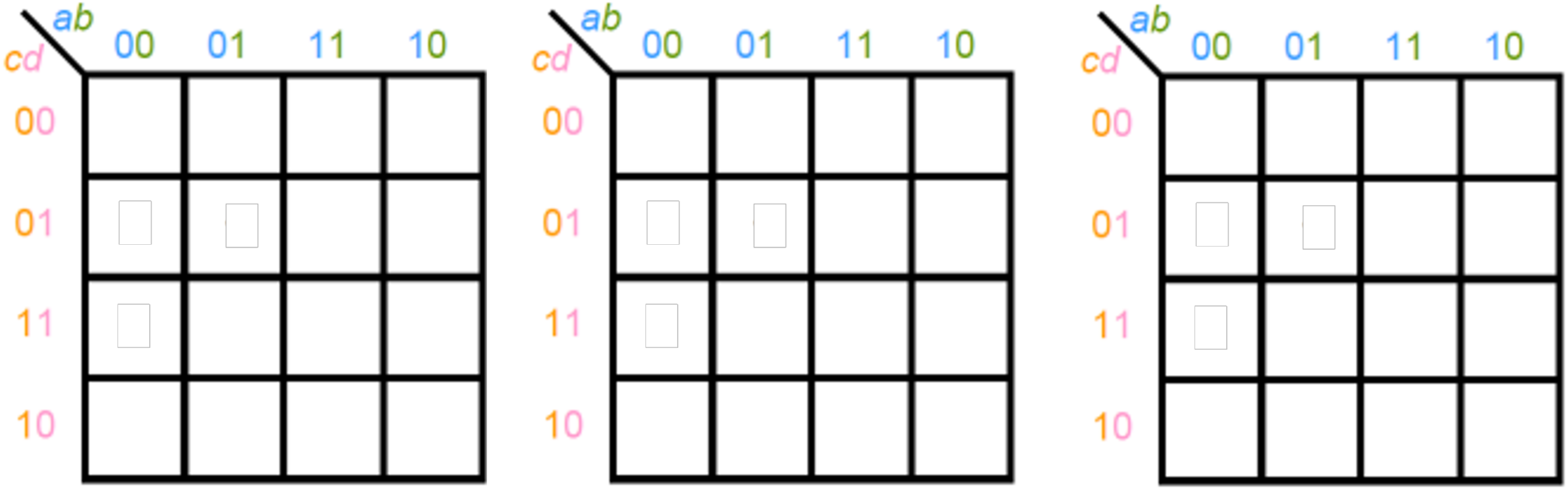
Find all minimum sum of products and all minimum product of sums expressions for the following Boolean Function.

$$f(a,b,c,d) = \sum m(1,3,4,6,11) + \sum d(0,8,10,12,13)$$

cd \ ab	00	01	11	10
00	m_0	m_4	m_{12}	m_8
01	m_1	m_5	m_{13}	m_9
11	m_3	m_7	m_{15}	m_{11}
10	m_2	m_6	m_{14}	m_{10}

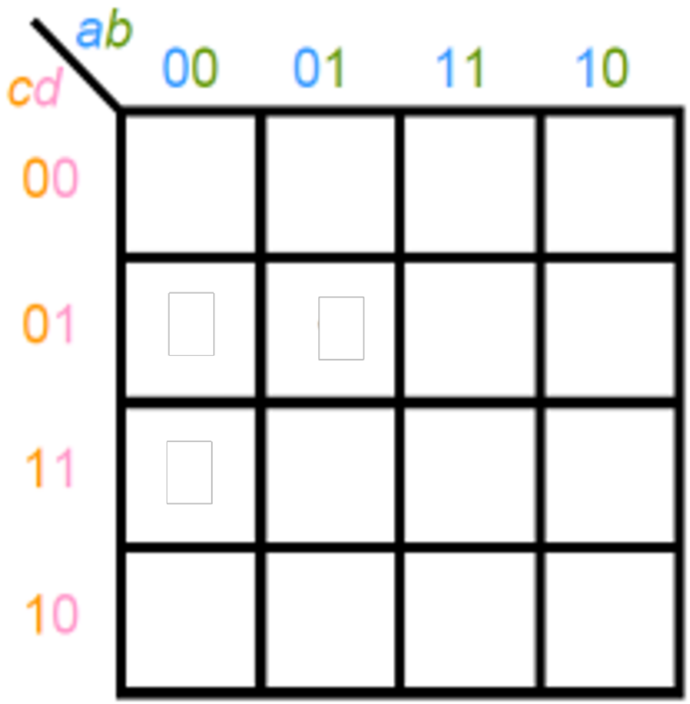
Exercise

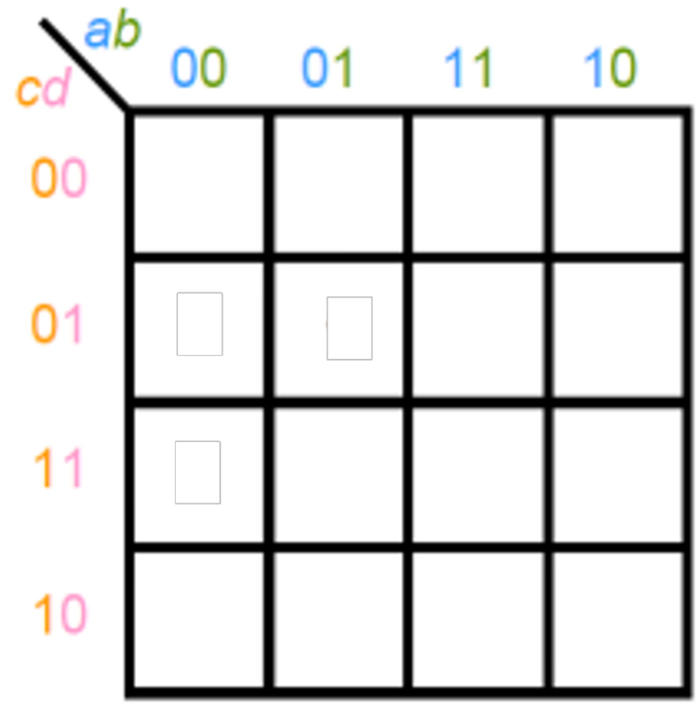
$$f(a,b,c,d) = \sum m(1,3,4,6,11) + \sum d(0,8,10,12,13)$$



Exercise

$$f(a,b,c,d) = \sum m(1,3,4,6,11) + \sum d(0,8,10,12,13)$$





5-variable K-map

V=0

WX \ YZ	00	01	11	10
00	m_0	m_1	m_3	m_2
01	m_4	m_5	m_7	m_6
11	m_{12}	m_{13}	m_{15}	m_{14}
10	m_8	m_9	m_{11}	m_{10}

V=1

WX \ YZ	00	01	11	10
00	m_{16}	m_{17}	m_{19}	m_{18}
01	m_{20}	m_{21}	m_{23}	m_{22}
11	m_{28}	m_{29}	m_{31}	m_{30}
10	m_{24}	m_{25}	m_{27}	m_{26}

6-variable K-map

AB=00

EF \ CD	00	01	11	10
00	m0	m4	m12	m8
01	m1	m5	m13	m9
11	m3	m7	m15	m11
10	m2	m6	m14	m10

AB=10

EF \ CD	00	01	11	10
00	m32	m36	m44	m40
01	m33	m37	m45	m41
11	m35	m39	m47	m43
10	m34	m38	m46	m42

AB=01

EF \ CD	00	01	11	10
00	m16	m20	m28	m24
01	m17	m21	m29	m25
11	m19	m23	m31	m27
10	m18	m22	m30	m26

AB=11

EF \ CD	00	01	11	10
00	m48	m52	m60	m56
01	m49	m53	m61	m57
11	m51	m55	m63	m59
10	m50	m54	m62	m58

2.4 Quine-McCluskey (QM) Method

- Developed by W. V. Quine and E. J. McCluskey in 1956
- Functionally identical to Karnaugh map
- More efficient in computer algorithms
- Ease to handle large number of variables

For number of variables that is less than or equal to 4, we use K-map; otherwise, QM method will be more efficient.

Procedure of QM-method (4-variable)

Step 1: Partition (Group minterms by the number of 1's)

w	x	y	z	Minterms
0	0	0	0	m_0
0	0	0	1	m_1
0	0	1	0	m_2
0	0	1	1	m_3
0	1	0	0	m_4
0	1	0	1	m_5
0	1	1	0	m_6
0	1	1	1	m_7
1	0	0	0	m_8
1	0	0	1	m_9
1	0	1	0	m_{10}
1	0	1	1	m_{11}
1	1	0	0	m_{12}
1	1	0	1	m_{13}
1	1	1	0	m_{14}
1	1	1	1	m_{15}

Minterms	$wxyz$
m_0	0000
m_1	0001
m_2	0010
m_4	0100
m_8	1000
m_3	0011
m_5	0101
m_6	0110
m_9	1001
m_{10}	1010
m_{12}	1100
m_7	0111
m_{11}	1011
m_{13}	1101
m_{14}	1110
m_{15}	1111

Given in test and exam

Procedure of QM-method (4-variable)

Step 1: Partition (Group minterms by the number of 1's)

Simplify $f(a, b, c, d) = \Sigma m(1, 4, 5, 6, 8, 9, 10, 12, 14)$

Minterms	$abcd$
m_1	0001
m_4	0100
m_8	1000
m_5	0101
m_6	0110
m_9	1001
m_{10}	1010
m_{12}	1100
m_{14}	1110

Procedure of QM-method (4-variable)

Step 2: Combine (Apply adjacency property to each pair of terms in consecutive groups)

- Combine adjacent group implicants into (n-1) variable implicants
- Mark the changed bit with "-" and tick the combined impicants

Minterms	$abcd$
m_1	0001 ✓
m_4	0100
m_8	1000
m_5	0101 ✓
m_6	0110
m_9	1001 ✓
m_{10}	1010
m_{12}	1100
m_{14}	1110

Minterms	$abcd$
m_1, m_5	0-01
m_1, m_9	-001

Procedure of QM-method (4-variable)

Step 2: Combine (Apply adjacency property to each pair of terms in consecutive groups)

- Combine adjacent group implicants into (n-1) variable implicants
- Mark the changed bit with "-" and tick the combined impicants

Minterms	$abcd$
m_1	0001 ✓
m_4	0100 ✓
m_8	1000
m_5	0101 ✓
m_6	0110 ✓
m_9	1001 ✓
m_{10}	1010
m_{12}	1100 ✓
m_{14}	1110

Minterms	$abcd$
m_1, m_5	0-01
m_1, m_9	-001
m_4, m_5	010-
m_4, m_6	01-0
m_4, m_{12}	-100

Procedure of QM-method (4-variable)

Step 2: Combine (Apply adjacency property to each pair of terms in consecutive groups)

- Combine adjacent group implicants into (n-1) variable implicants
- Mark the changed bit with "-" and tick the combined impicants

Minterms	*abcd*
m_1	0001 ✓
m_4	0100 ✓
m_8	1000 ✓
m_5	0101 ✓
m_6	0110 ✓
m_9	1001 ✓
m_{10}	1010 ✓
m_{12}	1100 ✓
m_{14}	1110

Minterms	*abcd*
m_1, m_5	0-01
m_1, m_9	-001
m_4, m_5	010-
m_4, m_6	01-0
m_4, m_{12}	-100
m_8, m_9	100-
m_8, m_{10}	10-0
m_8, m_{12}	1-00

Procedure of QM-method (4-variable)

Step 2: Combine (Apply adjacency property to each pair of terms in consecutive groups)

- Combine adjacent group implicants into (n-1) variable implicants
- Mark the changed bit with "-" and tick the combined impicants

Minterms	$abcd$
m_1	0001 ✓
m_4	0100 ✓
m_8	1000 ✓
m_5	0101 ✓
m_6	0110 ✓
m_9	1001 ✓
m_{10}	1010 ✓
m_{12}	1100 ✓
m_{14}	1110 ✓

Minterms	$abcd$
m_1, m_5	0-01
m_1, m_9	-001
m_4, m_5	010-
m_4, m_6	01-0
m_4, m_{12}	-100
m_8, m_9	100-
m_8, m_{10}	10-0
m_8, m_{12}	1-00
m_6, m_{14}	-110
m_{10}, m_{14}	1-10
m_{12}, m_{14}	11-0

Procedure of QM-method (4-variable)

Step 2: Combine (Apply adjacency property to each pair of terms in consecutive groups)

- Combine adjacent group implicants into (n-1) variable implicants
- Mark the changed bit with "-" and tick the combined impicants

Minterms	$abcd$
m_1	0001 ✔
m_4	0100 ✔
m_8	1000 ✔
m_5	0101 ✔
m_6	0110 ✔
m_9	1001 ✔
m_{10}	1010 ✔
m_{12}	1100 ✔
m_{14}	1110 ✔

Minterms	$abcd$
m_1, m_5	0-01
m_1, m_9	-001
m_4, m_5	010-
m_4, m_6	01-0 ✔
m_4, m_{12}	-100 ✔
m_8, m_9	100-
m_8, m_{10}	10-0
m_8, m_{12}	1-00
m_6, m_{14}	-110 ✔
m_{10}, m_{14}	1-10
m_{12}, m_{14}	11-0 ✔

Minterms	$abcd$
m_4, m_6, m_{12}, m_{14}	-1-0

Procedure of QM-method (4-variable)

Step 2: Combine (Apply adjacency property to each pair of terms in consecutive groups)

- Combine adjacent group implicants into (n-1) variable implicants
- Mark the changed bit with "-" and tick the combined impicants

Minterms	*abcd*
m_1	0001 ✓
m_4	0100 ✓
m_8	1000 ✓
m_5	0101 ✓
m_6	0110 ✓
m_9	1001 ✓
m_{10}	1010 ✓
m_{12}	1100 ✓
m_{14}	1110 ✓

Minterms	*abcd*
m_1, m_5	0-01
m_1, m_9	-001
m_4, m_5	010-
m_4, m_6	01-0 ✓
m_4, m_{12}	-100 ✓
m_8, m_9	100-
m_8, m_{10}	10-0 ✓
m_8, m_{12}	1-00 ✓
m_6, m_{14}	-110 ✓
m_{10}, m_{14}	1-10 ✓
m_{12}, m_{14}	11-0 ✓

Minterms	*abcd*
m_4, m_6, m_{12}, m_{14}	-1-0
$m_8, m_{10}, m_{12}, m_{14}$	1--0

Procedure of QM-method (4-variable)

Step 3: Identify Prime Implicants (PIs)

-All unmarked terms

Minterms	$abcd$
m_1	0001 ✓
m_4	0100 ✓
m_8	1000 ✓
m_5	0101 ✓
m_6	0110 ✓
m_9	1001 ✓
m_{10}	1010 ✓
m_{12}	1100 ✓
m_{14}	1110 ✓

Minterms	$abcd$
m_1, m_5	0-01 PI_3
m_1, m_9	-001 PI_4
m_4, m_5	010- PI_5
m_4, m_6	01-0 ✓
m_4, m_{12}	-100 ✓
m_8, m_9	100- PI_6
m_8, m_{10}	10-0 ✓
m_8, m_{12}	1-00 ✓
m_6, m_{14}	-110 ✓
m_{10}, m_{14}	1-10 ✓
m_{12}, m_{14}	11-0 ✓

Minterms	$abcd$
m_4, m_6, m_{12}, m_{14}	-1-0 PI_1
$m_8, m_{10}, m_{12}, m_{14}$	1--0 PI_2

Procedure of QM-method (4-variable)

Step 4: Generate PI chart

- Identify minterms that are covered by only 1 PI

- Identify essential PIs

$f(a, b, c, d) = \Sigma m(1, 4, 5, 6, 8, 9, 10, 12, 14)$

PI	Minterms	$abcd$	1	4	5	6	8	9	10	12	14
PI_1	m_4, m_6, m_{12}, m_{14}	-1-0									
PI_2	$m_8, m_{10}, m_{12}, m_{14}$	1--0									
PI_3	m_1, m_5	0-01									
PI_4	m_1, m_9	-001									
PI_5	m_4, m_5	010-									
PI_6	m_8, m_9	100-									

Procedure of QM-method (4-variable)

Step 4: Generate PI chart

- Identify minterms that are covered by only 1 PI
- Identify essential PIs

$f(a, b, c, d) = \Sigma m(1, 4, 5, 6, 8, 9, 10, 12, 14)$

PI	Minterms	*abcd*	1	4	5	6	8	9	10	12	14
PI_1	m_4, m_6, m_{12}, m_{14}	-1-0		x		x				x	x
PI_2	$m_8, m_{10}, m_{12}, m_{14}$	1--0					x		x	x	x
PI_3	m_1, m_5	0-01	x		x						
PI_4	m_1, m_9	-001	x					x			
PI_5	m_4, m_5	010-		x	x						
PI_6	m_8, m_9	100-					x	x			

Procedure of QM-method (4-variable)

Step 4: Generate PI chart

- Identify minterms that are covered by only 1 PI

- Identify essential PIs

PI	Minterms	$abcd$	1	4	5	6	8	9	10	12	14
PI_1	m_4, m_6, m_{12}, m_{14}	-1-0		x		x				x	x
PI_2	$m_8, m_{10}, m_{12}, m_{14}$	1--0					x		x	x	x
PI_3	m_1, m_5	0-01	x		x						
PI_4	m_1, m_9	-001	x					x			
PI_5	m_4, m_5	010-		x	x						
PI_6	m_8, m_9	100-					x	x			

∴ PI_1 and PI_2 are essential PIs.

Procedure of QM-method (4-variable)

Step 5: Reduce PI chart

- Remove the rows of EPIs and the columns that covered by them

PI	Minterms	$abcd$	1	4	5	6	8	9	10	12	14
PI_1	m_4, m_6, m_{12}, m_{14}	-1-0		x		x				x	x
PI_2	$m_8, m_{10}, m_{12}, m_{14}$	1--0					x		x	x	x
PI_3	m_1, m_5	0-01	x		x						
PI_4	m_1, m_9	-001	x					x			
PI_5	m_4, m_5	010-		x	x						
PI_6	m_8, m_9	100-					x	x			

Procedure of QM-method (4-variable)

Step 5: Reduce PI chart

- Remove the rows of EPIs and the columns that covered by them

PI	Minterms	$abcd$	1	5	9
PI_3	m_1, m_5	0-01	x	x	
PI_4	m_1, m_9	-001	x		x
PI_5	m_4, m_5	010-		x	
PI_6	m_8, m_9	100-			x

Solution 1

PI	Minterms	$abcd$	1	5	9
PI_3	m_1, m_5	0-01	x	x	
PI_4	m_1, m_9	-001	x		x

Solution 2

PI	Minterms	$abcd$	1	5	9
PI_3	m_1, m_5	0-01	x	x	
PI_6	m_8, m_9	100-			x

Solution 3

PI	Minterms	$abcd$	1	5	9
PI_4	m_1, m_9	-001	x		x
PI_5	m_4, m_5	010-		x	

Procedure of QM-method (4-variable)

Step 6: Express the Boolean Function

PI	Minterms	$abcd$
PI_1	m_4, m_6, m_{12}, m_{14}	-1-0
PI_2	$m_8, m_{10}, m_{12}, m_{14}$	1--0
PI_3	m_1, m_5	0-01
PI_4	m_1, m_9	-001

$$f(a,b,c,d) = \text{PI}_1 + \text{PI}_2 + \text{PI}_3 + \text{PI}_4$$

$$= bd' + ad' + a'c'd + b'c'd$$

PI	Minterms	$abcd$
PI_1	m_4, m_6, m_{12}, m_{14}	-1-0
PI_2	$m_8, m_{10}, m_{12}, m_{14}$	1--0
PI_3	m_1, m_5	0-01
PI_6	m_8, m_9	100-

$$f(a,b,c,d) = \text{PI}_1 + \text{PI}_2 + \text{PI}_3 + \text{PI}_6$$

$$= bd' + ad' + a'c'd + ab'c'$$

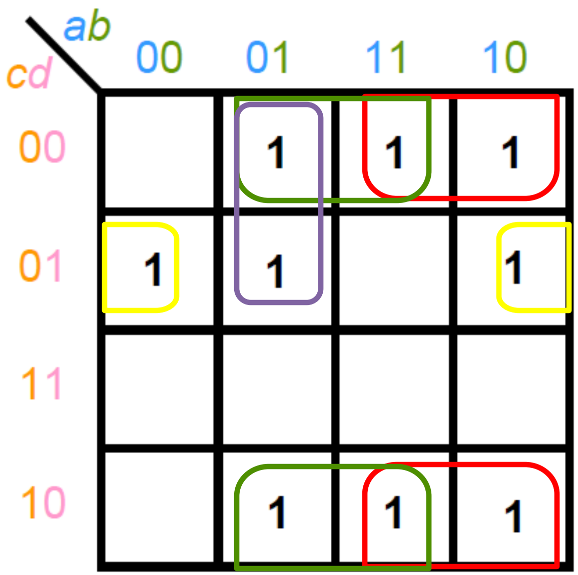
PI	Minterms	$abcd$
PI_1	m_4, m_6, m_{12}, m_{14}	-1-0
PI_2	$m_8, m_{10}, m_{12}, m_{14}$	1--0
PI_4	m_1, m_9	-001
PI_5	m_4, m_5	010-

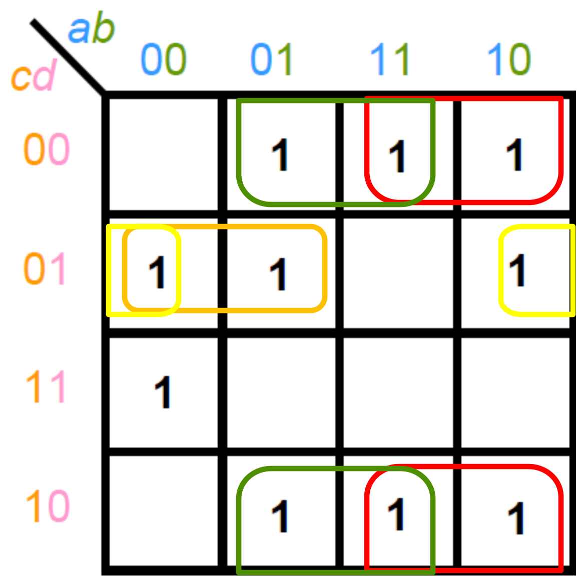
$$f(a,b,c,d) = \text{PI}_1 + \text{PI}_2 + \text{PI}_4 + \text{PI}_5$$

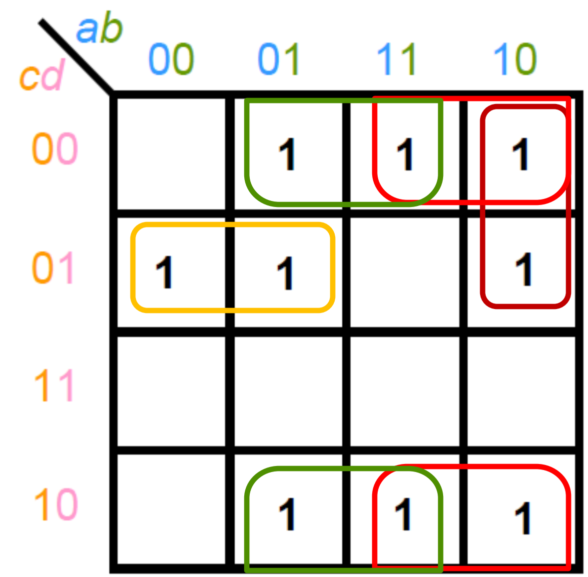
$$= bd' + ad' + b'c'd + a'bc'$$

Verification by K-map

Simplify $f(a, b, c, d) = \Sigma m(1, 4, 5, 6, 8, 9, 10, 12, 14)$







$$f(a,b,c,d) = bd' + ad' + a'bc' + b'c'd$$

$$f(a,b,c,d) = bd' + ad' + a'c'd + b'c'd$$

$$f(a,b,c,d) = bd' + ad' + a'c'd + ab'c'$$

Don’t Care Case

Step 1-3 (Partition, Combine, List PIs): Include Don’t Care minterms

Simplify $f(a, b, c, d) = \Sigma m(4, 8, 9, 10, 12, 15) + \Sigma d(2, 6, 13)$

Minterms	$abcd$
m_2	0010 ✓
m_4	0100 ✓
m_8	1000 ✓
m_6	0110 ✓
m_9	1001 ✓
m_{10}	1010 ✓
m_{12}	1100 ✓
m_{13}	1101 ✓
m_{15}	1111 ✓

Minterms	$abcd$
m_2, m_6	0-10 PI_2
m_2, m_{10}	-010 PI_3
m_4, m_6	01-0 PI_4
m_4, m_{12}	-100 PI_5
m_8, m_9	100- ✓
m_8, m_{10}	10-0 PI_6
m_8, m_{12}	1-00 ✓
m_9, m_{13}	1-01 ✓
m_{12}, m_{13}	110- ✓
m_{13}, m_{15}	11-1 PI_7

Minterms	$abcd$
m_8, m_9, m_{12}, m_{13}	1-0- PI_1

Don't Care Case

Step 4: Generate PI chart

- Exclude Don't Care Minterms

Simplify $f(a, b, c, d) = \Sigma m(4, 8, 9, 10, 12, 15) + \Sigma d(2, 6, 13)$

PI	Minterms	*abcd*	4	8	9	10	12	15
PI_1	m_8, m_9, m_{12}, m_{13}	1-0-						
PI_2	m_2, m_6	0-10						
PI_3	m_2, m_{10}	-010						
PI_4	m_4, m_6	01-0						
PI_5	m_4, m_{12}	-100						
PI_6	m_8, m_{10}	10-0						
PI_7	m_{13}, m_{15}	11-1						

Step 5-6: Reduce PI chart & express the Boolean Function

Exercise (Don't Care Case)

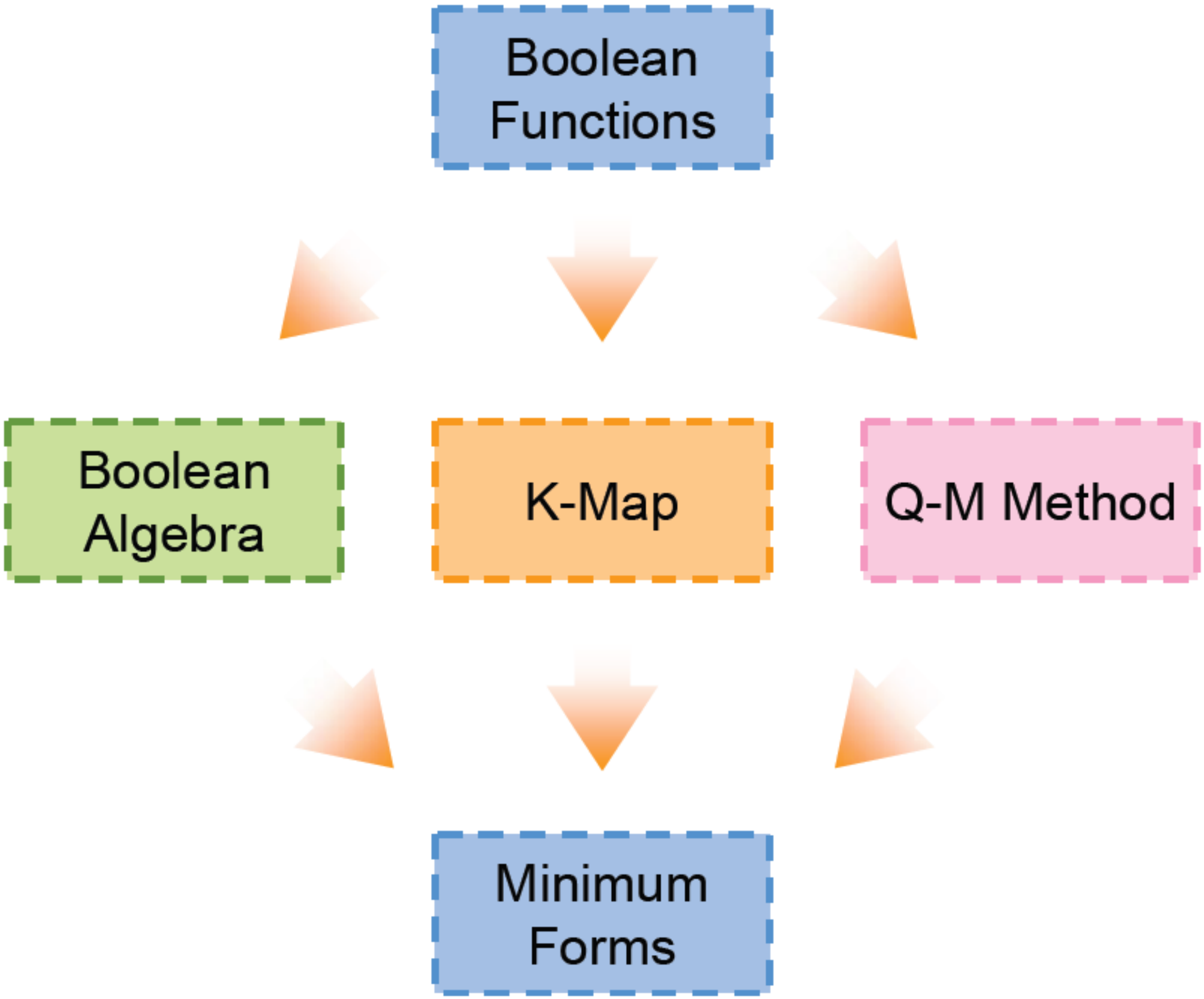
Step 5-6: Reduce PI chart & express the Boolean Function

PI	Minterms	$abcd$	4	8	9	10	12	15
PI_1	m_8, m_9, m_{12}, m_{13}	1-0-		x	x		x	
PI_2	m_2, m_6	0-10						
PI_3	m_2, m_{10}	-010				x		
PI_4	m_4, m_6	01-0	x					
PI_5	m_4, m_{12}	-100	x				x	
PI_6	m_8, m_{10}	10-0		x		x		
PI_7	m_{13}, m_{15}	11-1						x

PI	Minterms	$abcd$		

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

Summary


Boolean Functions
Boolean Algebra
K-Map
Q-M Method
Minimum Forms

Multiple Output Problems

c \ ab	00	01	11	10
0	m_0	m_2	m_6	m_4
1	m_1	m_3	m_7	m_5

$f(a,b,c) = \sum m(2,3,7)$

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

$g(a,b,c) = \sum m(4,5,7)$

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

Multiple Output Problems

$f(a,b,c) = \sum m(2,3,7)$

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

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

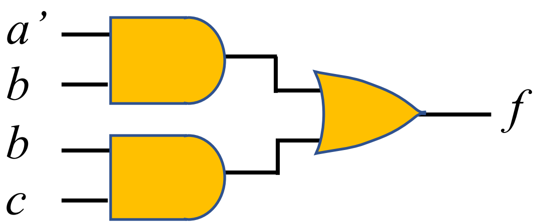
$g(a,b,c) = \sum m(4,5,7)$

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

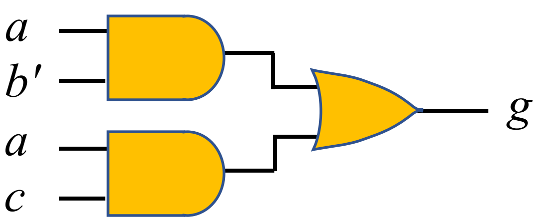
$g(a,b,c) = ab' + ac$

Multiple Output Problems

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



$g(a, b, c) = ab' + ac$



12 gate inputs and 6 gates (how to reduce the cost?)

Multiple Output Problems

c \ ab	00	01	11	10
0	m_0	m_2	m_6	m_4
1	m_1	m_3	m_7	m_5

$$f(a,b,c) = \sum m(2,3,7)$$

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

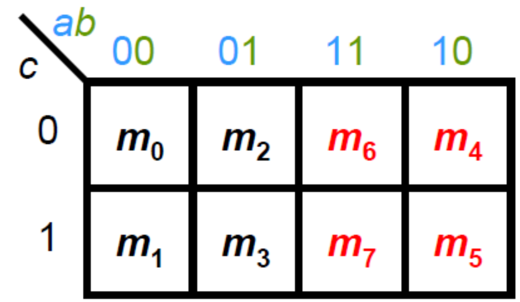
$$f(a,b,c) = a'b + bc$$

$$g(a,b,c) = \sum m(4,5,7)$$

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

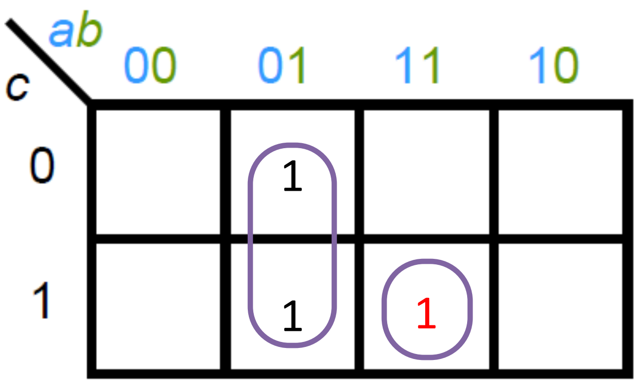
$$g(a,b,c) = ab' + ac$$

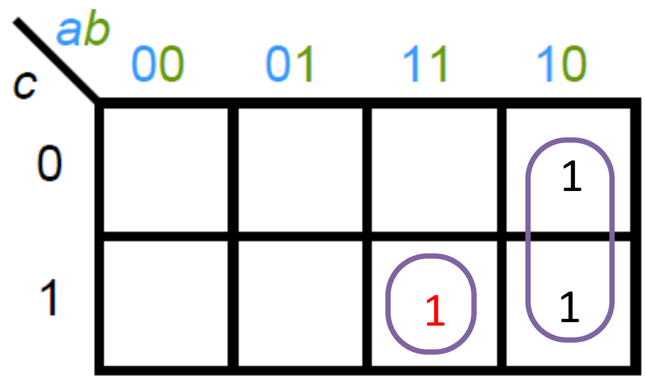
Multiple Output Problems



$$f(a,b,c) = \sum m(2,3,7)$$

$$g(a,b,c) = \sum m(4,5,7)$$





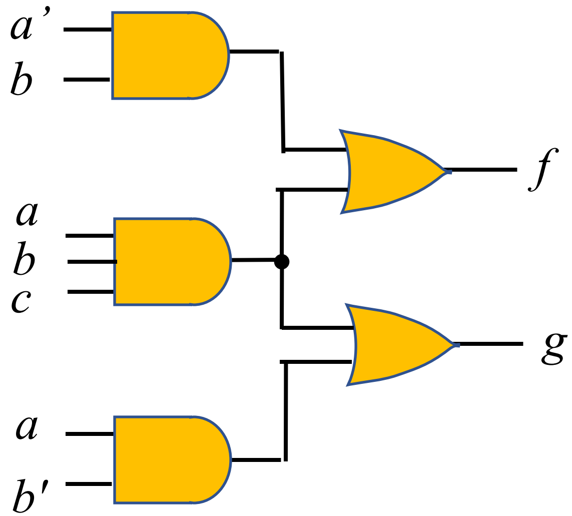
$$f(a,b,c) = a'b + abc$$

$$g(a,b,c) = ab' + abc$$

Multiple Output Problems

$f(a, b, c) = a'b + abc$

$g(a, b, c) = ab' + abc$



11 gate inputs and 5 gates!!!

Exercise

Reduce the following functions that will use the least number of gates and gate inputs.

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

cd \ ab	00	01	11	10
00	1	1		1
01	1	1		1
11				
10	1	1		

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

cd \ ab	00	01	11	10
00				1
01				1
11			1	1
10	1	1	1	1

Exercise

Reduce the following functions that will use the least number of gates and gate inputs.

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

cd \ ab	00	01	11	10
00	1	1		1
01	1	1		1
11				
10	1	1		

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

cd \ ab	00	01	11	10
00				1
01				1
11			1	1
10	1	1	1	1