Switching Circuits & Logic Design

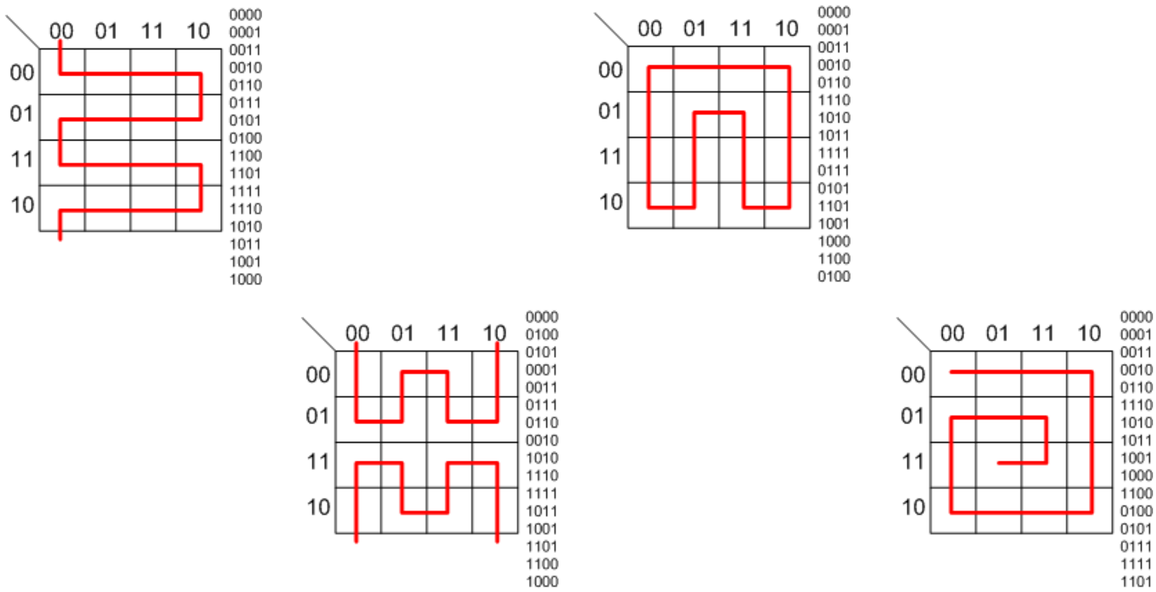
Jie-Hong Roland Jiang
江介宏

Department of Electrical Engineering
National Taiwan University

Fall 2013

§5 Karnaugh Maps

K-map Walks and Gray Codes



http://asicdigitaldesign.wordpress.com/2008/09/26/k-maps-walks-and-gray-codes/

Outline

- Minimum forms of switching functions
- Two- and three-variable Karnaugh maps
- Four-variable Karnaugh maps
- Determination of minimum expressions using essential prime implicants
- Five-variable Karnaugh maps
- Other uses of Karnaugh maps
- Other forms of Karnaugh maps

Limitations of Algebraic Simplification

- Two problems of algebraic simplification
  1. Not systematic
  2. Difficult to check if a minimum solution is achieved

- The *Karnaugh map* method overcomes these limitations
  - Typically for Boolean functions with ≤ 5 variables
  - The Quine-McCluskey method can deal with even larger functions
    - (Subject of Unit 6, skipped)

Minimum Forms of Switching Functions

- Correspondence between Boolean expressions and logic circuits
  - SOP (POS) can be implemented with two-level AND-OR (OR-AND) gate circuits
  - Reducing the number of terms and literals of an SOP expression corresponds to reducing the number of gates and gate inputs
    - Combine terms by XY'+XY=X
    - Eliminate redundant terms by consensus theorem
  - Minimum SOP is not necessarily unique
  - An SOP may be minimal (locally) but not minimum (globally)
    - E.g.,

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

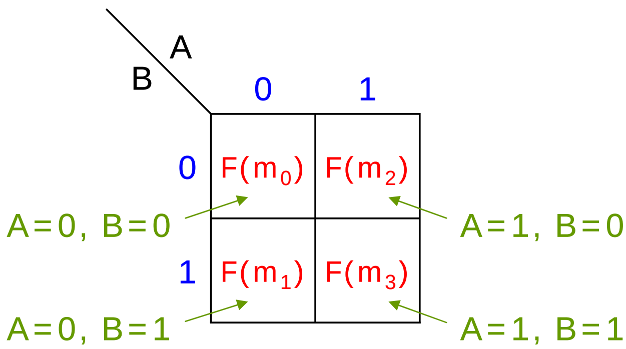
      = a'b'+b'c+bc'+ab (minimal but not minimum)

      = a'b'+bc'+ac (minimum)

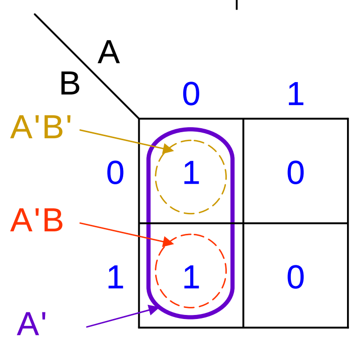
Two-Variable Karnaugh Maps

- 2-variable K-map

AB	F
00	1
01	1
10	0
11	0



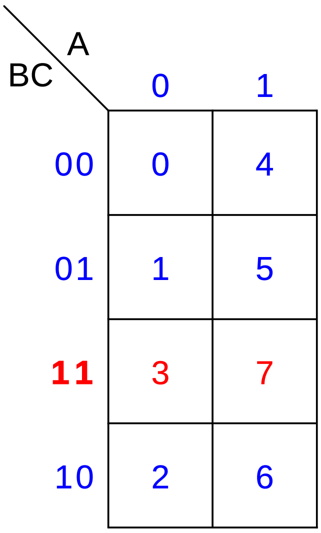
minterm locations



F = A'B'+ A'B = A'(B'+B) = A'

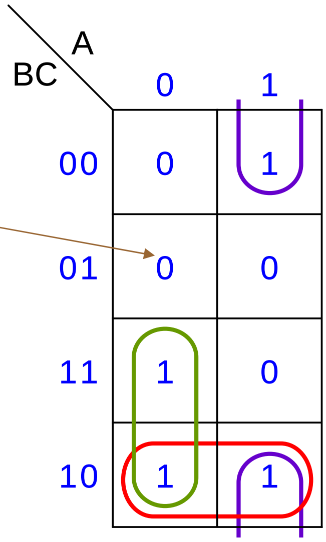
Three-Variable Karnaugh Maps

- 3-variable K-map



minterm locations

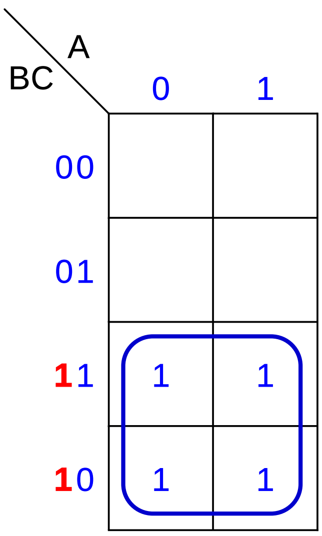
ABC	F
000	0
001	0
010	1
011	1
100	1
101	0
110	1
111	0



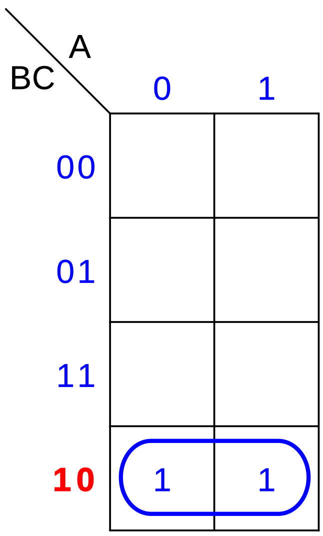
F = A'BC' + A'BC + AB'C' + ABC'
= A'B + AC' + BC'
= A'B + AC'

Three-Variable Karnaugh Maps

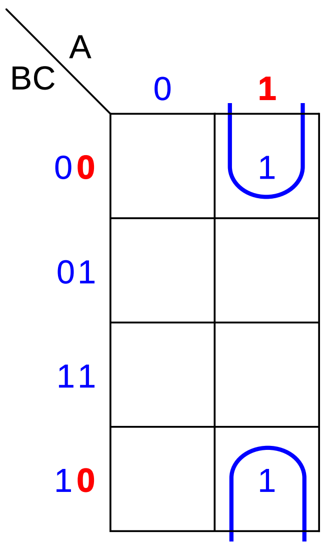
- 3-variable K-map (zeros omitted)



B



BC'

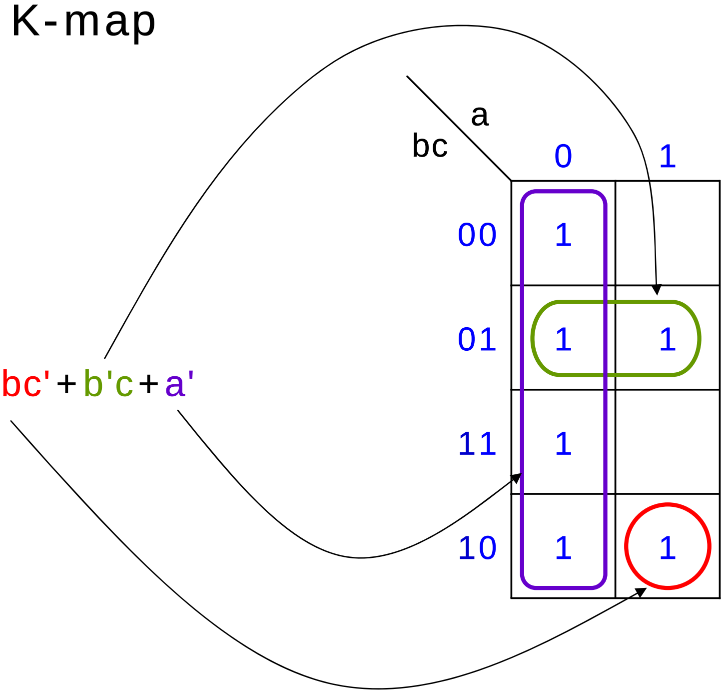


AC'

Three-Variable Karnaugh Maps

- 3-variable K-map

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

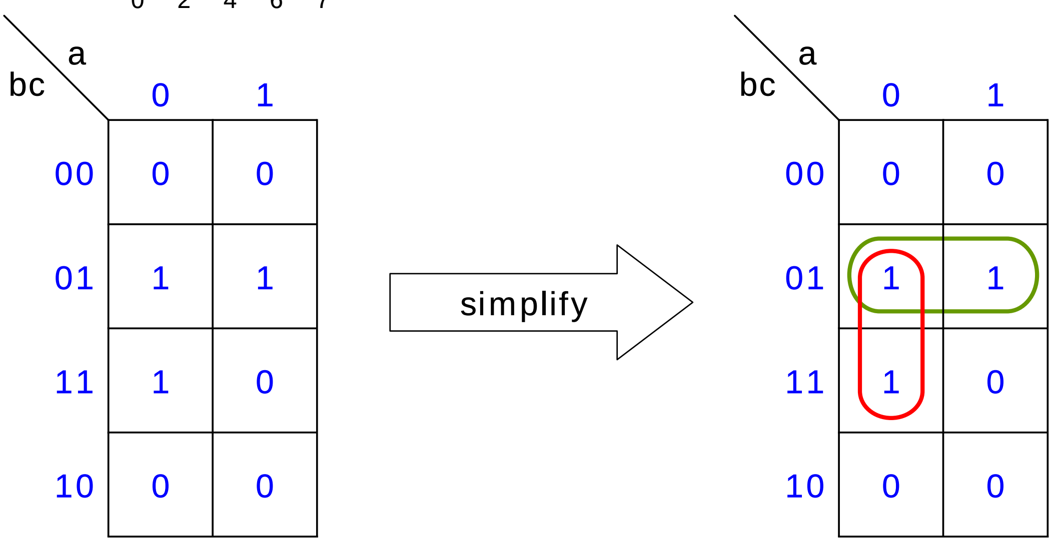


Three-Variable Karnaugh Maps

- 3-variable K-map

$F = m_1 + m_3 + m_5$
$= M_0 M_2 M_4 M_6 M_7$

F = a'c + b'c



Three-Variable Karnaugh Maps

- 3-variable K-map

$G = (m_1+m_3+m_5)'$
$= (M_0M_2M_4M_6M_7)'$

$G = ab + c'$

bc \ a	0	1
00	1	1
01	0	0
11	0	1
10	1	1

simplify

bc \ a	0	1
00	1	1
01	0	0
11	0	1
10	1	1

Three-Variable Karnaugh Maps

- 3-variable K-map

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

simplify

$xy+x'z+yz = xy+x'z$
(consensus theorem)

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

Three-Variable Karnaugh Maps

- 3-variable K-map

F = a'b'+bc'+ac = a'c'+b'c+ab

bc \ a	0	1
00	1	
01	1	1
11		1
10	1	1

bc \ a	0	1
00	1	
01	1	1
11		1
10	1	1

Four-Variable Karnaugh Maps

- 4-variable K-map

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

minterm locations

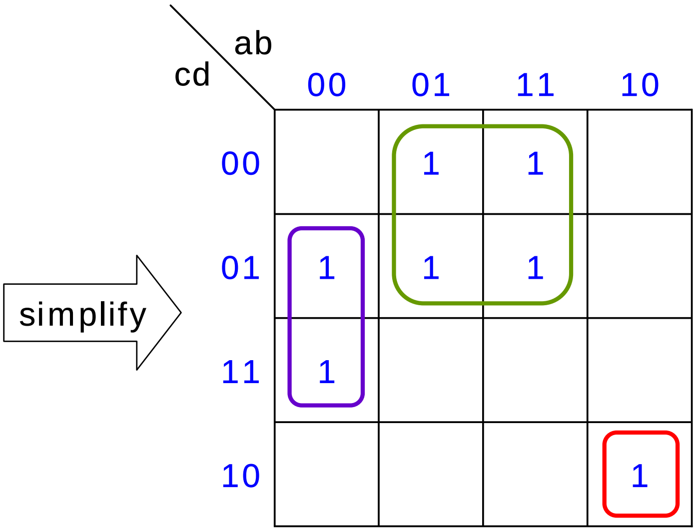
F = acd+a'b+d'

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

Four-Variable Karnaugh Maps

- 4-variable K-map

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



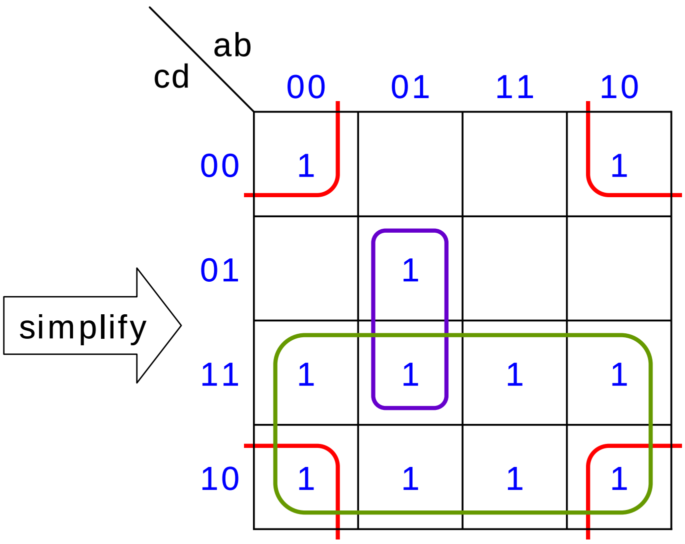
$f_1 = \sum m(1,3,4,5,10,12,13)$

$f_1 = ab'cd' + a'b'd + bc'$

Four-Variable Karnaugh Maps

- 4-variable K-map

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



$f_2 = \sum m(0,2,3,5,6,7,8,10,11,14,15)$

$f_2 = c + b'd' + a'bd$

Four-Variable Karnaugh Maps

- Simplify incompletely specified function
  - All the 1's must be covered, but X's are optional and are set to 1's only if they will simplify the expression

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

simplify

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

$f = \sum m(1,3,5,7,9) + \sum d(6,12,13)$

$f = a'd + c'd$

Four-Variable Karnaugh Maps

- Simplify product-of-sums
  - Circle 0's instead of 1's
  - Apply De Morgan's law converting SOP to POS

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

simplify

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

$f = x'z' + wyz + w'y'z' + x'y$

$f' = y'z + wxz' + w'xy$

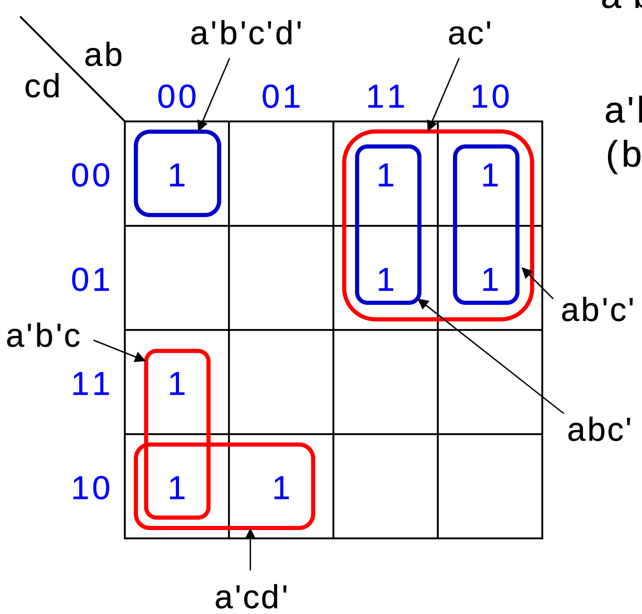
$f = (y + z')(w' + x' + z)(w + x' + y')$

Determination of Minimum Expressions Using Essential Prime Implicants

- **Implicant**
  - A product term of a function
    - Any single 1 or any group of 1’s on a K-map combined together forms a product term
- **Prime implicant**
  - A maximal implicant
    - An implicant that cannot be combined with another term to eliminate a variable
- All of the prime implicants of a function can be obtained from a K-map by expanding the 1’s as much as possible in every possible way

Determination of Minimum Expressions Using Essential Prime Implicants

Example

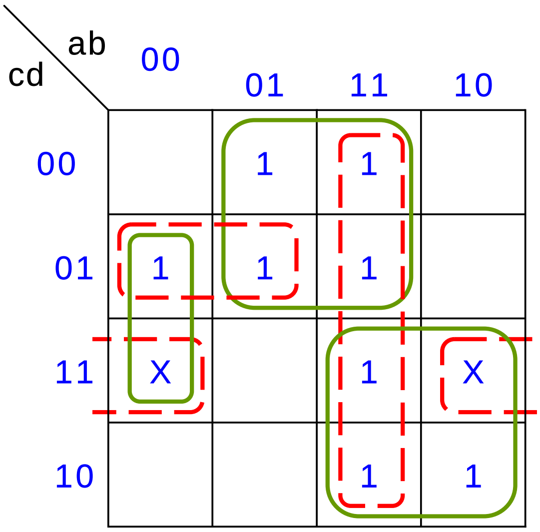


a'b'c, a'cd', ac' are prime implicants

a'b'c'd', abc', ab'c' are implicants (but not prime implicants)

Determination of Minimum Expressions Using Essential Prime Implicants

- Determine all prime implicants
  - In finding prime implicants, don't cares are treated as 1's. However, a prime implicant composed entirely of don't cares can never be part of the minimum solution
  - Not all prime implicants are needed in forming the minimum SOP

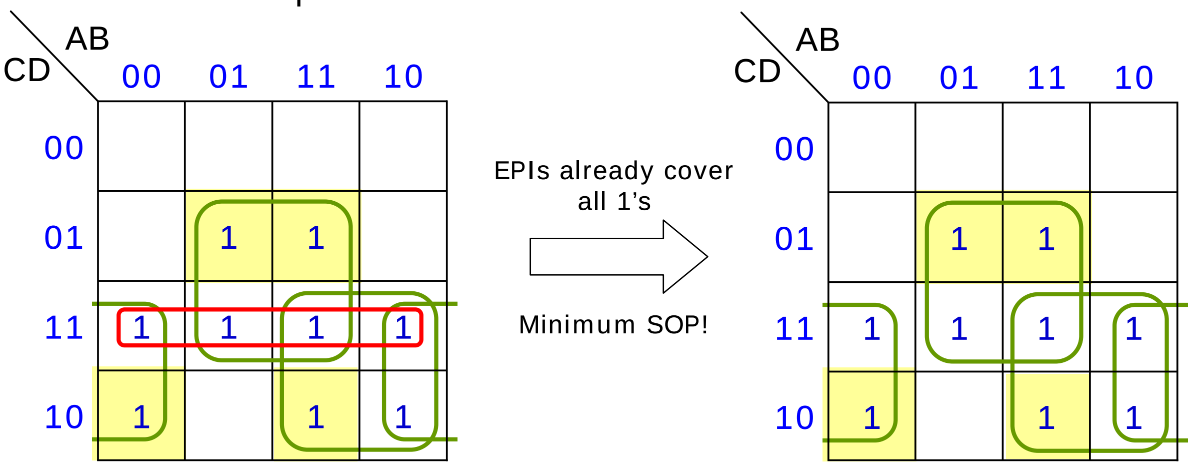


Example

- All prime implicants:
  a'b'd, bc', ac, a'c'd, ab, b'cd
  (composed entirely of don't cares)
- Minimum solution:
  F = a'b'd+bc'+ac

Determination of Minimum Expressions Using Essential Prime Implicants

- **Essential prime implicant (EPI)**
  - A prime implicant that covers some minterm not covered by any other prime implicant
    - If a single term covers some minterm and **all** of its adjacent 1's and X's, then the term is an EPI
  - Must be present in the minimum SOP

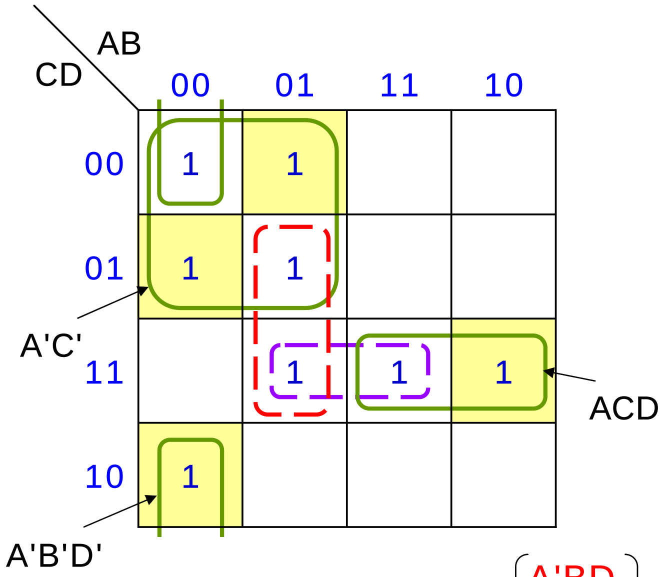


f = CD+BD+B'C+AC

f = BD+B'C+AC

Determination of Minimum Expressions Using Essential Prime Implicants

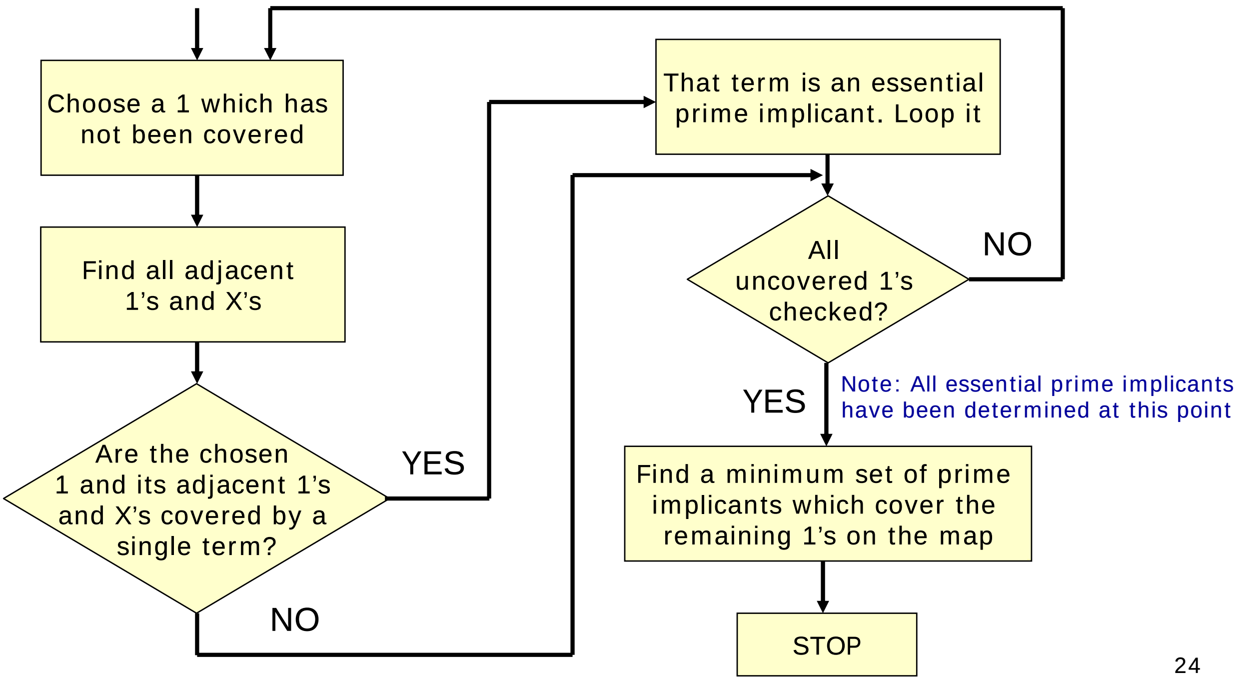
- SOP minimization
  1. Select all essential prime implicants
  2. Find a minimum set of prime implicants which cover the minterms not covered by the essential prime implicants
     - There may be freedom left after all essential prime implicants are selected (it affects optimality especially for functions with more variables)



F = A'C' + A'B'D' + ACD + { A'BD or BCD }

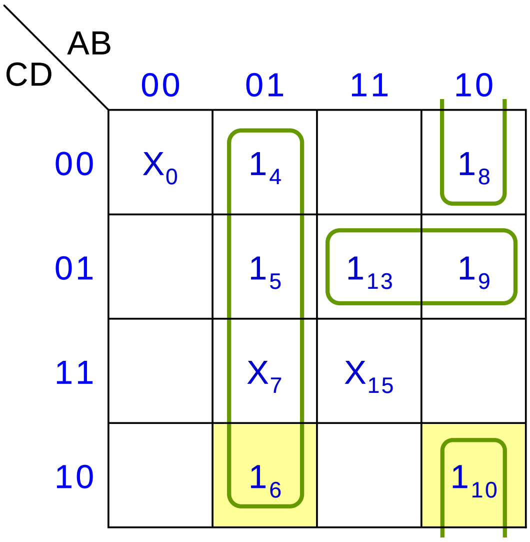
Determination of Minimum Expressions Using Essential Prime Implicants

- Flowchart for determining a minimum SOP using K-map



Determination of Minimum Expressions Using Essential Prime Implicants

Example



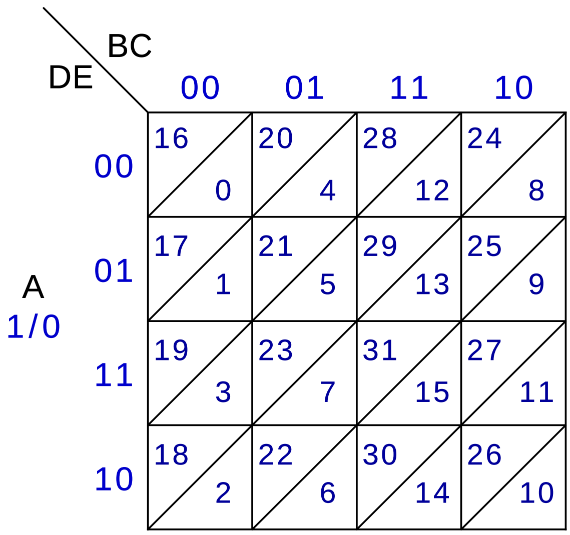
Step 1: 1_4 checked
Step 2: 1_5 checked
Step 3: 1_6 checked
EPI → A'B selected
Step 4: 1_8 checked
Step 5: 1_9 checked
Step 6: 1_{10} checked
EPI → AB'D' selected
Step 7: 1_{13} checked

(up to this point all EPIs determined)

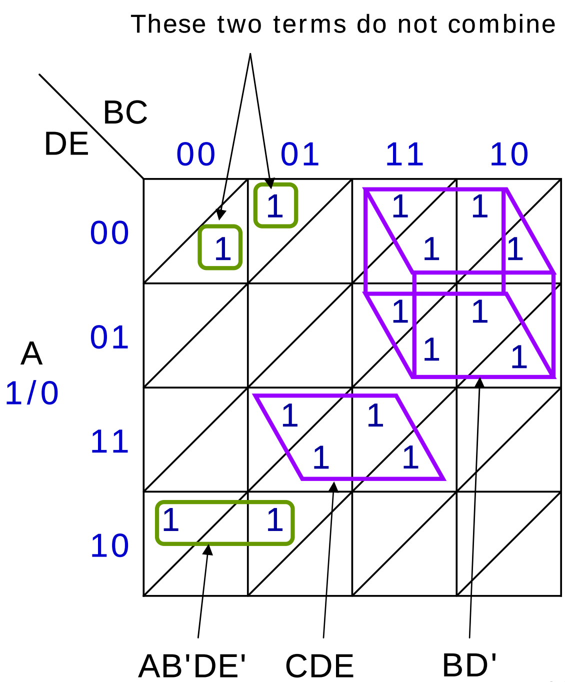
Step 8: AC'D selected to cover remaining 1's

Five-Variable Karnaugh Maps

5-var K-map

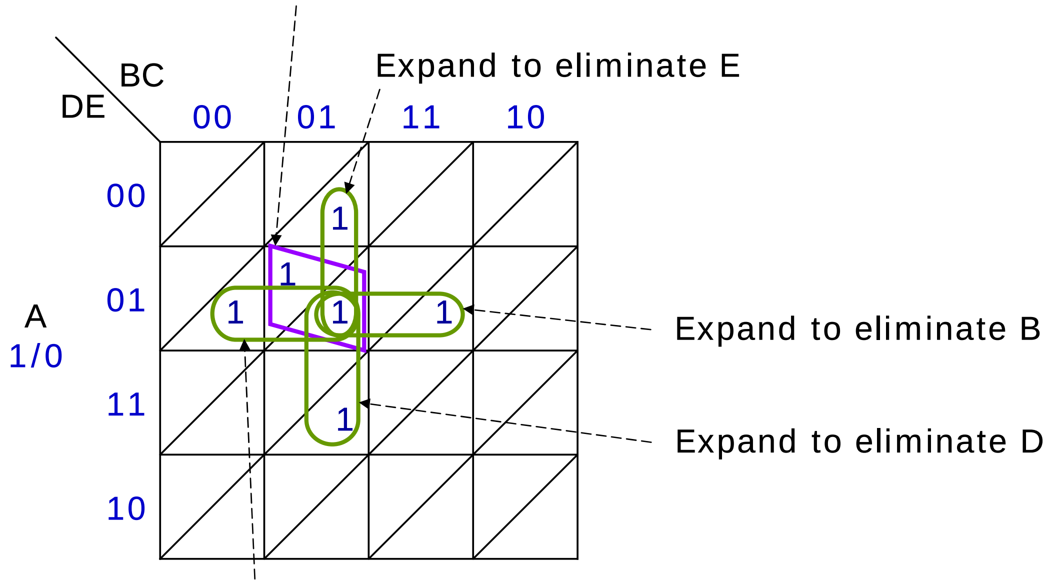


minterm locations



Five-Variable Karnaugh Maps

- 5-var K-map

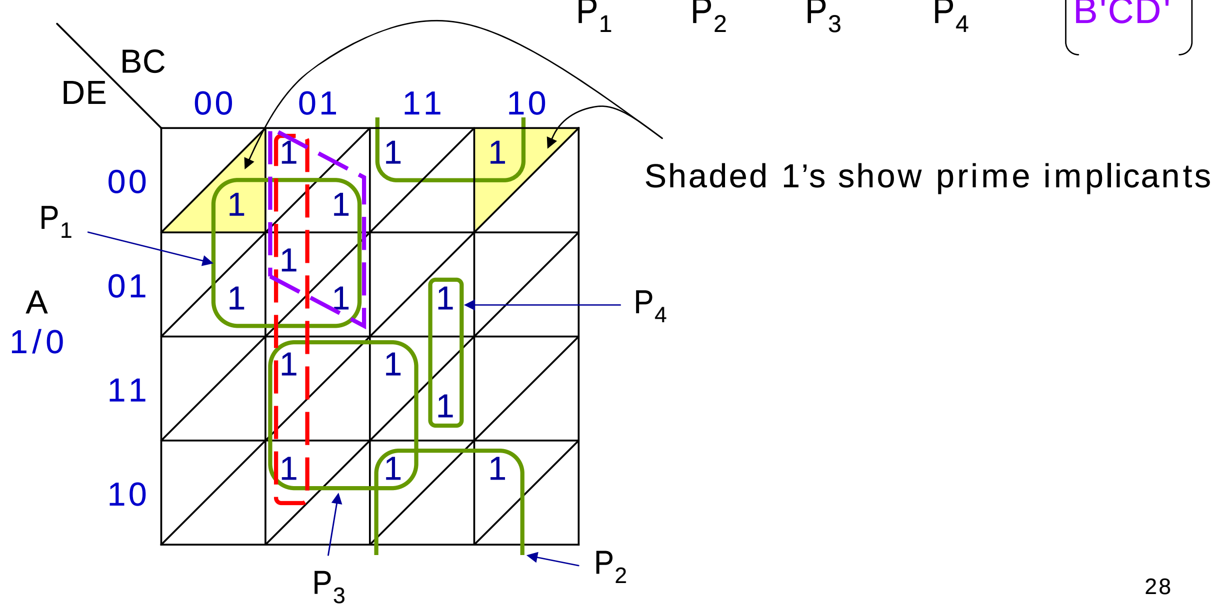


Five-Variable Karnaugh Maps

- 5-var K-map

F = A'B'D' + ABE' + ACD + A'BCE + { AB'C or B'CD' }

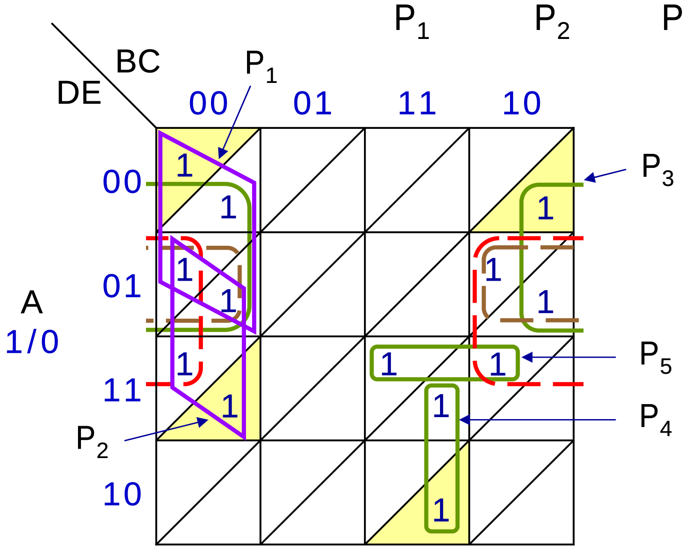
P1 = A'B'D', P2 = ABE', P3 = ACD, P4 = A'BCE



Five-Variable Karnaugh Maps

- 5-var K-map

$$F = \underset{P_1}{B'C'D'} + \underset{P_2}{B'C'E} + \underset{P_3}{A'C'D'} + \underset{P_4}{A'BCD} + \underset{P_5}{ABDE} + \left\{ \begin{matrix} AC'E \\ \text{or} \\ C'D'E \end{matrix} \right\}$$



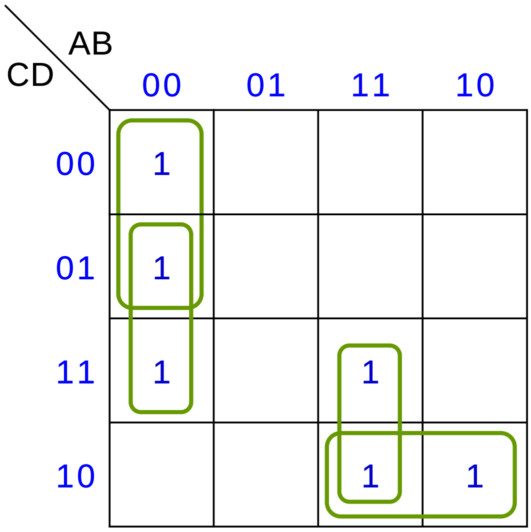
Other Uses of Karnaugh Maps

- Use K-map to prove the equivalence of two Boolean expressions
  - K-maps are canonical representations of Boolean functions, similar to truth tables

- Use K-map to perform Boolean operations
  - AND, OR, NOT operations can be done over K-maps (truth tables)

Other Uses of Karnaugh Maps

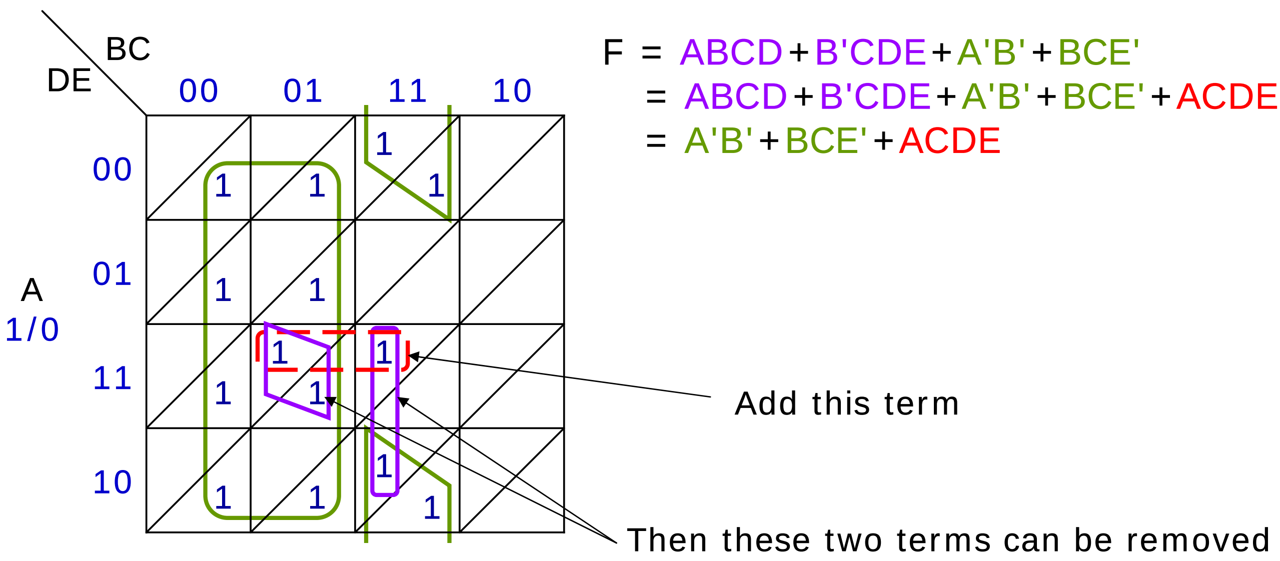
- Use K-map to facilitate factoring
  - Identify common literals among product terms



F = A'B'C'+A'B'D+ACB+ACD'
= A'B'(C'+D)+AC(B+D')

Other Uses of Karnaugh Maps

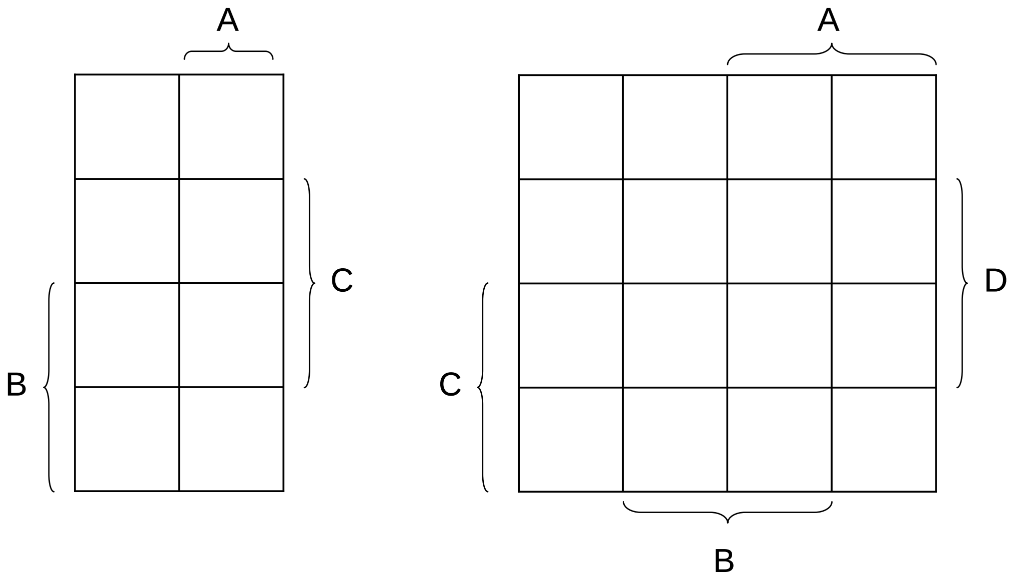
- Use K-map to guide simplification



F = ABCD+B'CDE+A'B'+BCE'
= ABCD+B'CDE+A'B'+BCE'+ACDE
= A'B'+BCE'+ACDE

Other Forms of Karnaugh Maps

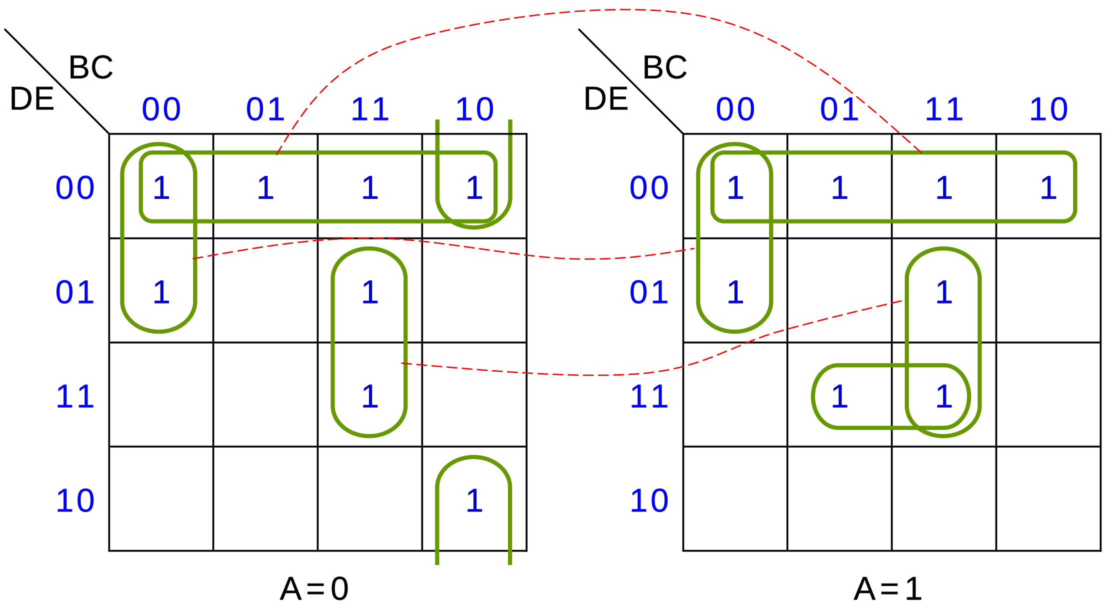
- Other conventions (Veitch diagrams)



Other Forms of Karnaugh Maps

- Other conventions (5-var K-map)

F = D'E'+B'C'D'+BCE+A'BC'E'+ACDE



Other Forms of Karnaugh Maps

- Other conventions (5-var Veitch diagram)

F = D'E'+B'C'D'+BCE+A'BC'E'+ACDE