Digital Logic Design

Hajar Falahati

Department of Computer Engineering
IRAN University of Science and Technology

hfalahati@iust.ac.ir

Digital Circuit Analysis

- Digital Circuit **Design**:
  - Word description of a function

    $\Rightarrow$ A set of switching equations

    $\Rightarrow$ Hardware realization (gates, programmable logic devices, etc.)

- Digital Circuit **Analysis**:
  - Hardware realization

    $\Rightarrow$ Switching expressions, truth tables, timing diagrams, etc.

- Analysis is used
  - To determine the behavior of the circuit
  - To verify the correctness of the circuit
  - To assist in converting the circuit to a different form

Outline

- Optimization
- Karnaugh maps

Simplification

Simplification

- Goal
  - Minimize the cost of realizing a switching function
- Cost measures and other considerations
  - Number of gates
  - Number of levels
  - Gate fan in and/or fan out
  - Interconnection complexity
  - Preventing hazards
- Two-level realizations
  - Minimize the number of gates (terms in switching function)
  - Minimize the fan in (literals in switching function)

Cost

- Literal cost (L)
  - Number of literal appearances in a boolean expression
  - All literal appearances

- Gate input cost (G)
  - # total number of inputs to the gates
    - All literal appearances
    - # terms excluding single literal terms

- Gate input cost including inverters (GN)
  - # total number of inputs to the gates + # inverters
    - All literal appearances
    - # terms excluding single literal terms
    - # distinct complemented single literals

Cost: Sample1

- Determine the form and the number of terms and literals.

$g(A,B,C) = AB' + A'B + AC$

- Two-level form
- Three products
- Two sums
- L = 6
- G = 6 + 3 =9
- GN = 9 + 2 = 11

$F = A + BC + B'C'$

- Four-level form
- Two products
- Two sums
- L = 5
- G = 5 + 2 =7
- GN = 7 + 2 = 9

Cost: Sample 2

- Determine the form and the number of terms and literals.

$f(X,Y,Z) = X'Y(Z + Y'X) + Y'Z$

- Four-level form
- Four products
- Two sums
- L = 7
- G = 7 + 4 =11
- GN = 11 + 2 = 13

$f(X,Y,Z) = B\,D + A\,B'\,C + A\,C'\,D'$

- Four-level form
- Four products
- Two sums
- L = 8
- G = 8+ 3 =11
- GN = 11 + 3 = 14

Cost: Sample 3

- Compare these two functions.

$$F = A\,B\,C + A'\,B'\,C'$$

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

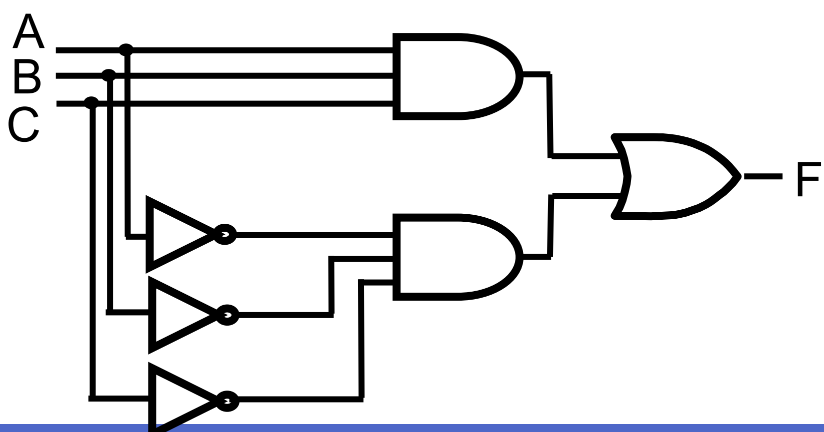
Cost: Sample 3 (cont'd)

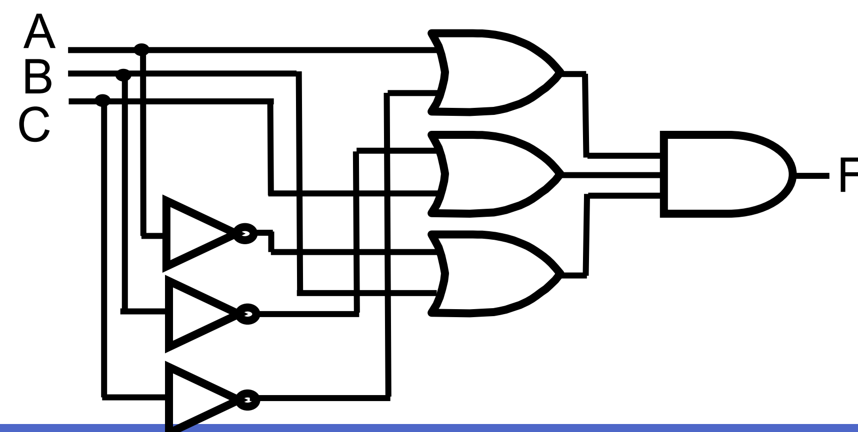
F = A B C + A' B' C'

- Two-level form
- Two products
- One sums
- L = 6
- G = 6 + 2 =8
- GN = 8 + 3 = 11

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

- Two-level form
- Three products
- Three sums
- L = 6
- G = 6 + 3 =9
- GN = 9 + 3 = 12





Simplification: Methods

- Commonly used techniques
  - Boolean algebra postulates and theorems
  - Karnaugh maps
  - Quine-McCluskey method
  - Petrick's method
  - Generalized concensus algorithm

- Characteristics
  - Heuristics (suboptimal)
  - Algorithms (optimal)

Boolean Algebra

- Minimum SOP and POS representation
- Minimum sum of products (MSOP) of *f*
  - A SOP representation of *f*
  - Contains the **fewest number** of product terms and
  - Contains the **fewest number** of literals of any SOP representation of *f*

$$f(a,b,c,d) = \sum m(3,7,11,12,13,14,15) = ab + a'cd + acd = ab + cd$$

- Minimum product of sums (MPOS) of *f*
  - A POS representation of *f*
  - Contains the **fewest number** of sum terms and
  - Contains the **fewest number** of literals of any POS representation of *f*

$$f(a,b,c,d) = \prod M(0,1,2,4,5,6,8,9,10)$$
$$= (a + c)(a + d)(a' + b + d)(b + c' + d)$$
$$= (a + c)(a + d)(b + c)(b + d)$$

Boolean Algebra: Yes or No?

- Trial and error process
- Almost never be sure that we have reached a minimal representation.

Karnaugh Maps

Karnaugh Map (K-map)

- Graphical technique
- A collection of cells (squares)
- Each cell represents a minterm (i.e., a row of an n-variable truth table)
- Collection of squares is a graphical representation of a function

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

$$f = \sum m(5,7,13,15)$$

$$= \bar{a}b\bar{c}d + \bar{a}bcd + ab\bar{c}d + abcd$$

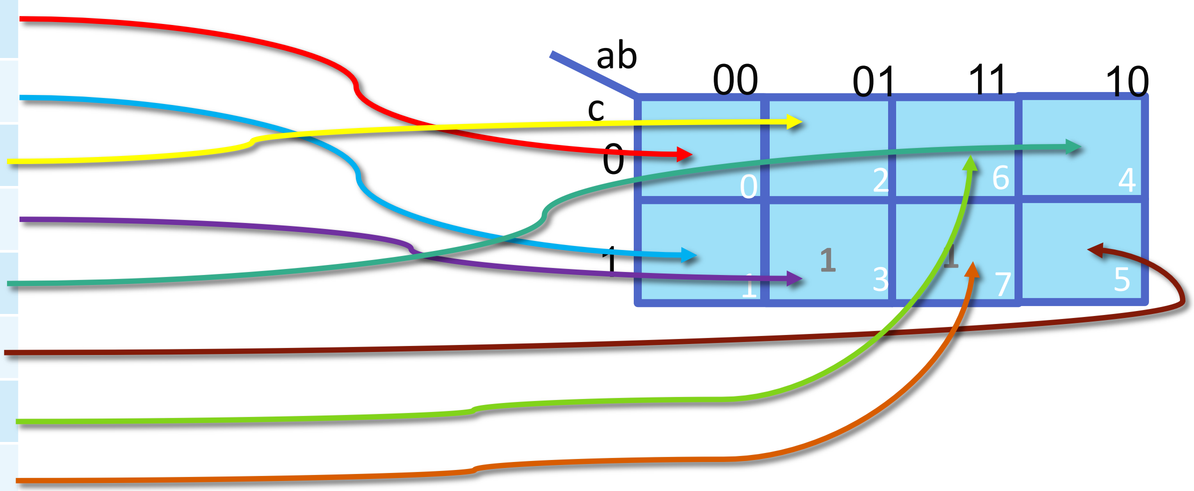
$$= b\bar{c}d + bcd$$

$$= bd$$

Karnaugh Map (K-map)

- An ***n*-variable** K-map has ***2^n^* cells**
- K-map cells are labeled with the corresponding truth-table row

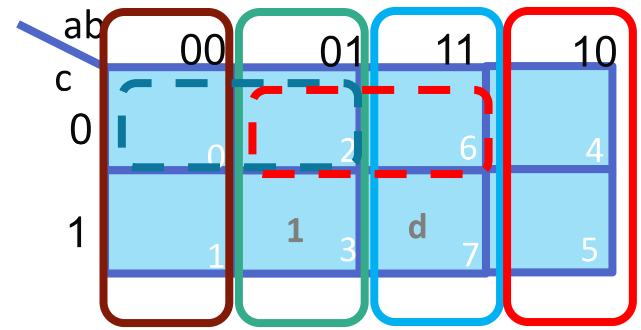
a	b	c	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1



Karnaugh Map (K-map)

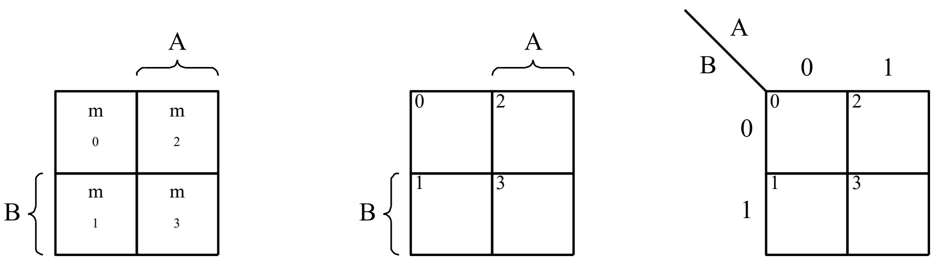
- Adjacent cells correspond to truth rows that differ in only one bit position (*logical adjacency*)
- Switching functions are mapped (or plotted) by placing the function's value (*0,1,d*) in each cell of the maps
- **A reorganized version of the truth table**

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

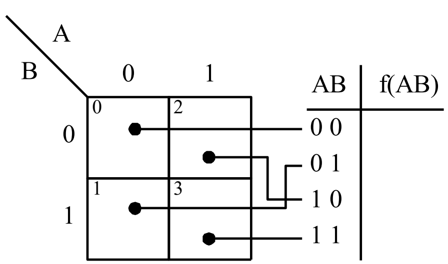


Venn diagram and equivalent K-Map

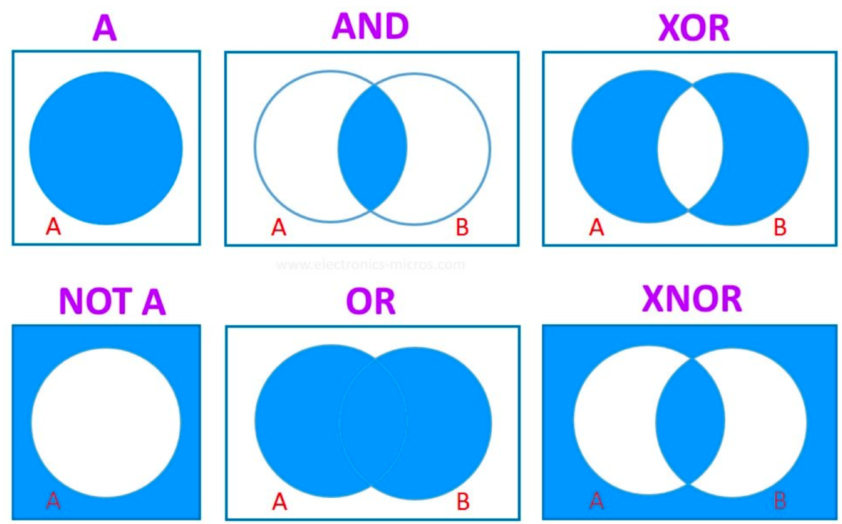
- Two variables
- m_0 and m_1 differ in variable B
- m_0 and m_2 differ in variable A



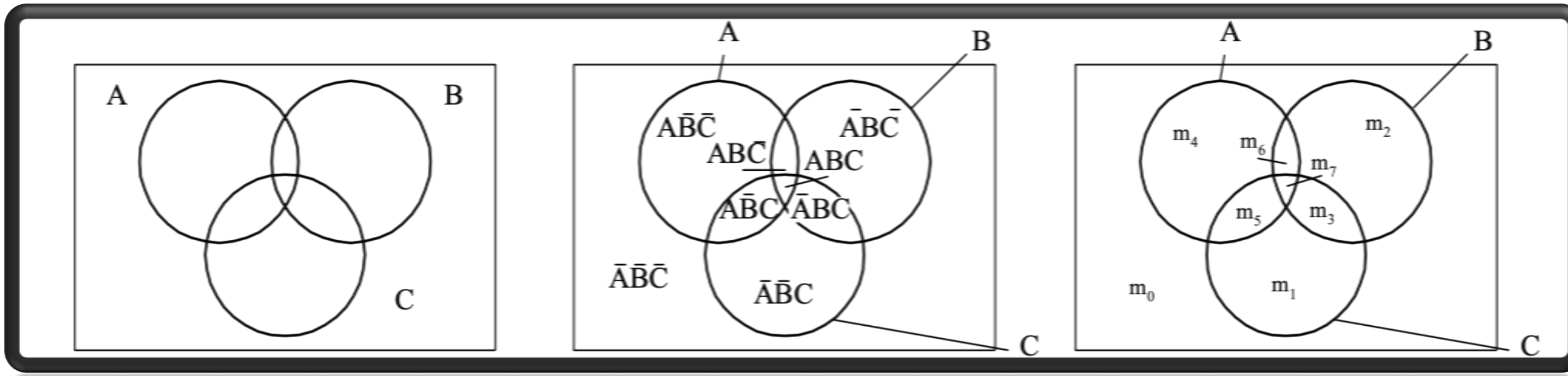
(d) (e) (f)

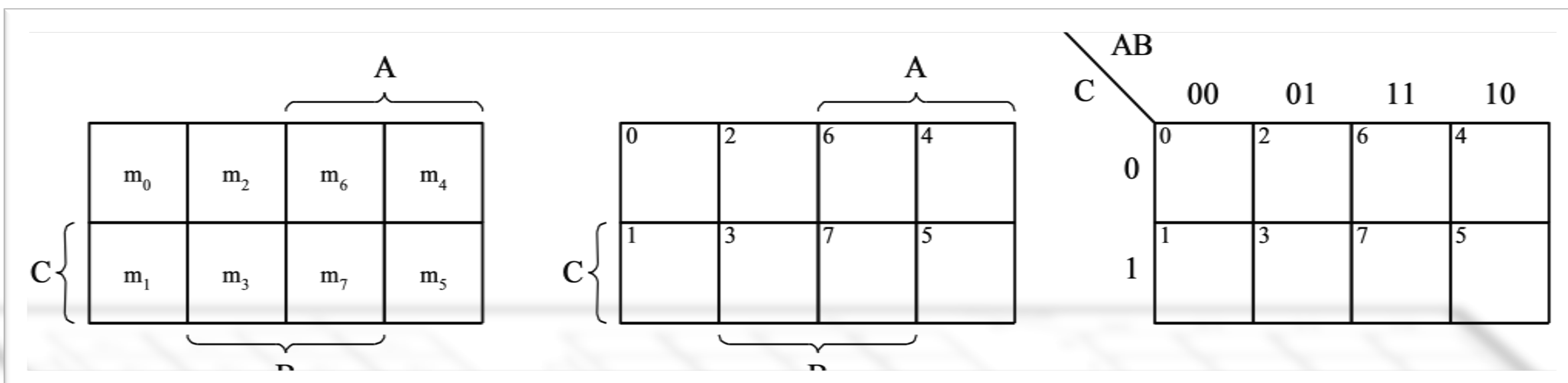


(g)

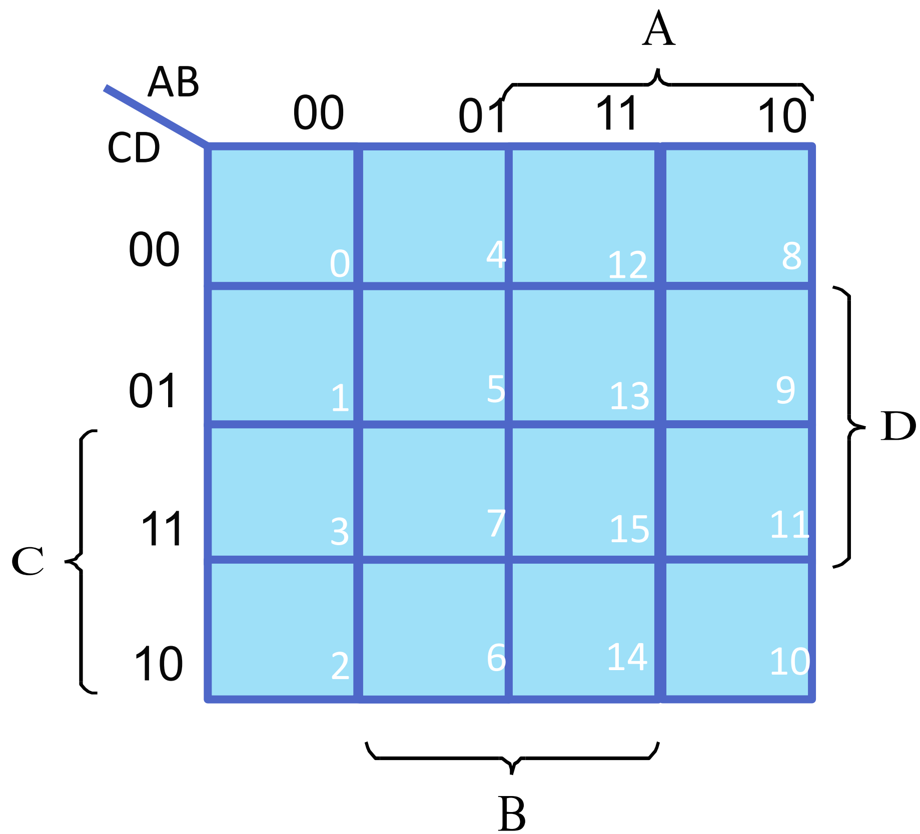


Venn diagram and equivalent K-Map (cont'd)



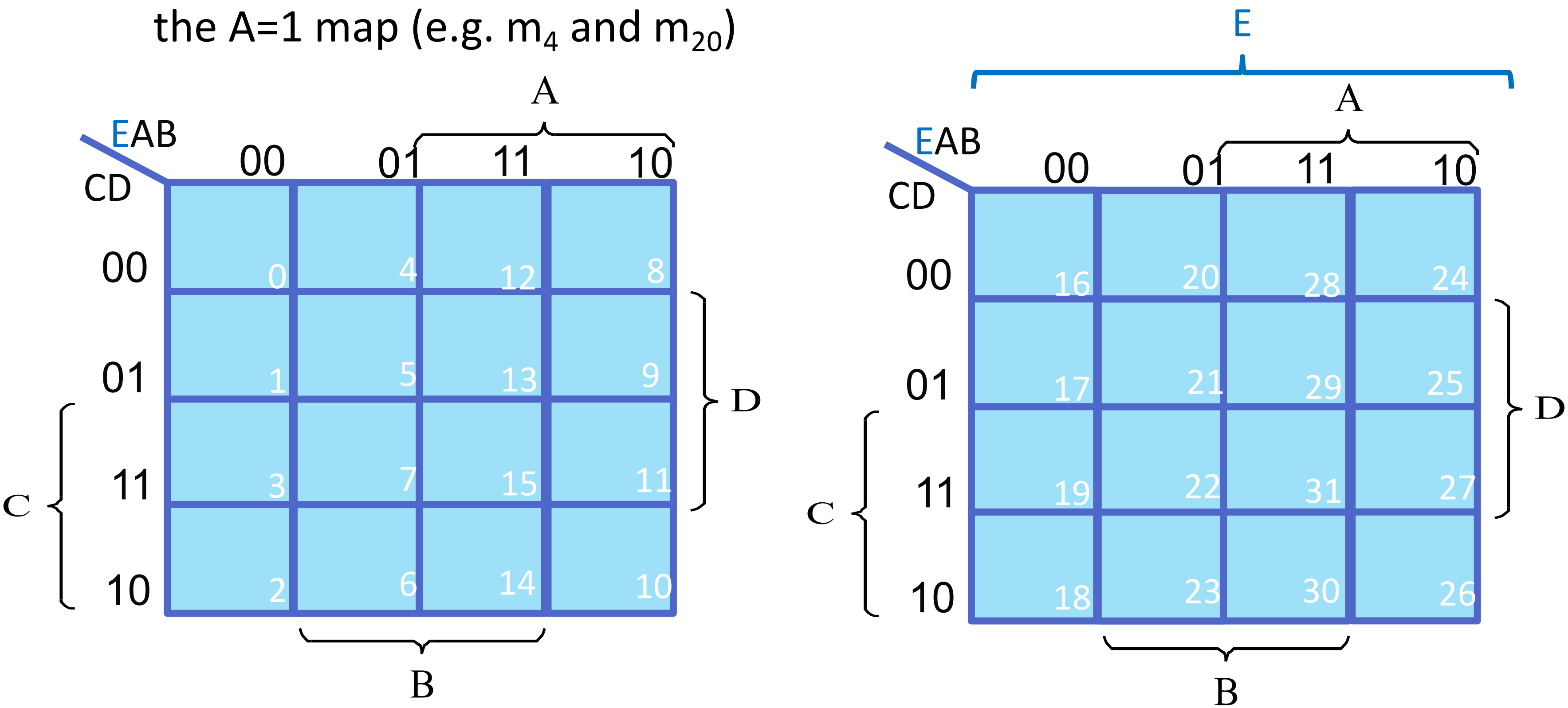


K-Map: Four Variables

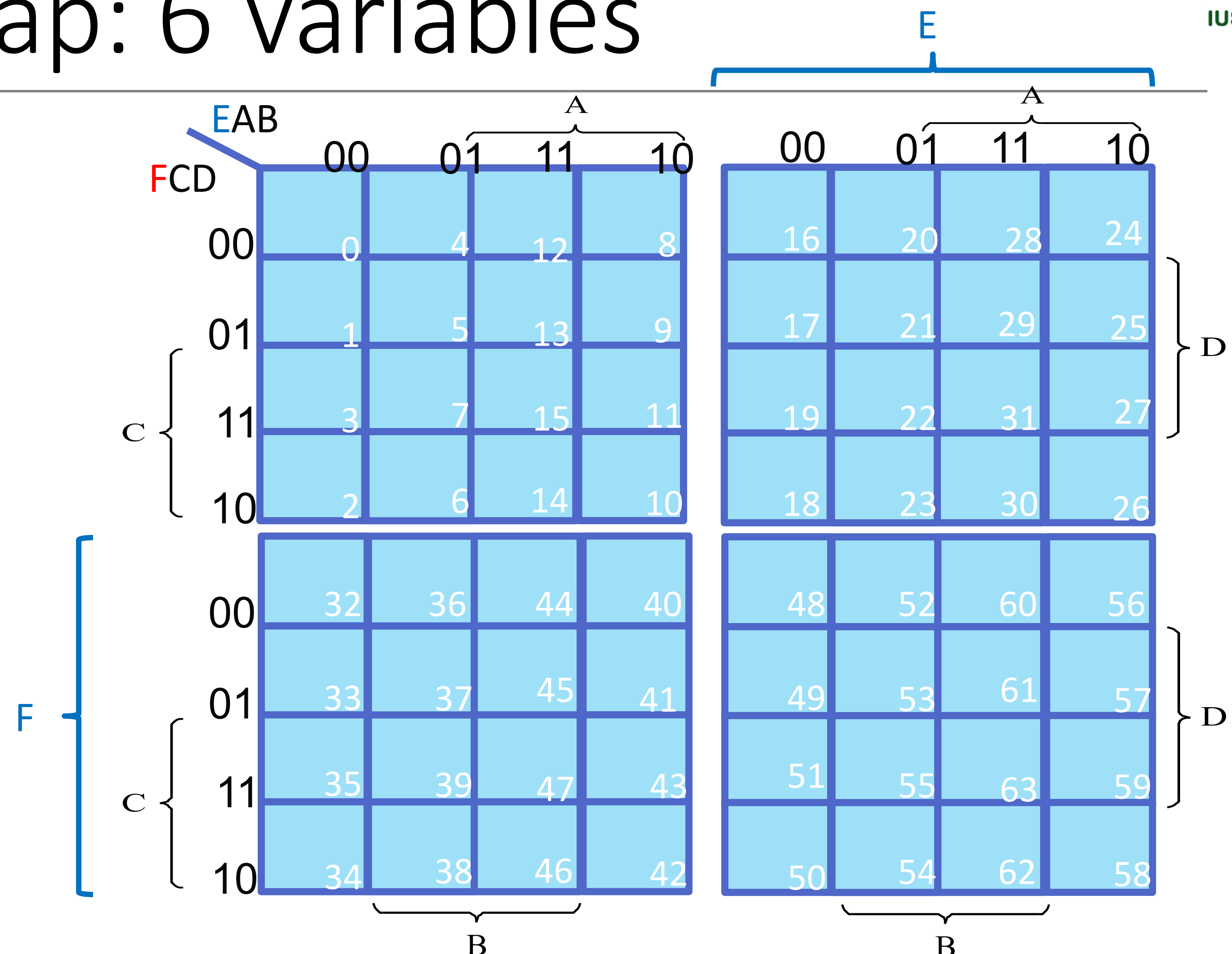


K-Map: Five Variables

- Two adjacent 4-variable K-maps
- Each square in the A=0 map is adjacent to the corresponding square in the A=1 map (e.g. m_4 and m_{20})



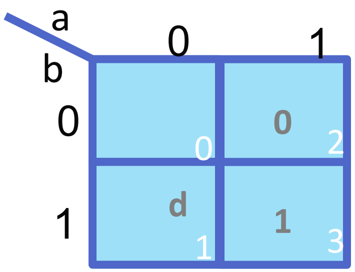
K-Map: 6 Variables



K-Map: Plotting function

Plotting function in canonical form on the k-map

- Let ***f*** be a switching function of ***n*** variables where ***n*** **≤** ***6***
- Assumption:
  - Cells p are numbered from 0 to 2^n
  - numbers correspond to the rows of the truth table of *f*
- If m_i is a minterm of *f*, then place a 1 in cell *i* of the K-map
  - *Example:* $f(A,B,C) = \sum m(0,3,5)$
- If M_i is a maxterm of *f*, then place a 0 in cell *i*
  - *Example:* $f(A,B,C) = \prod M(1,2,4,6,7)$
- If d_i is a don't care of *f*, then place a *d* in cell *i*



K-Map: Sample 4

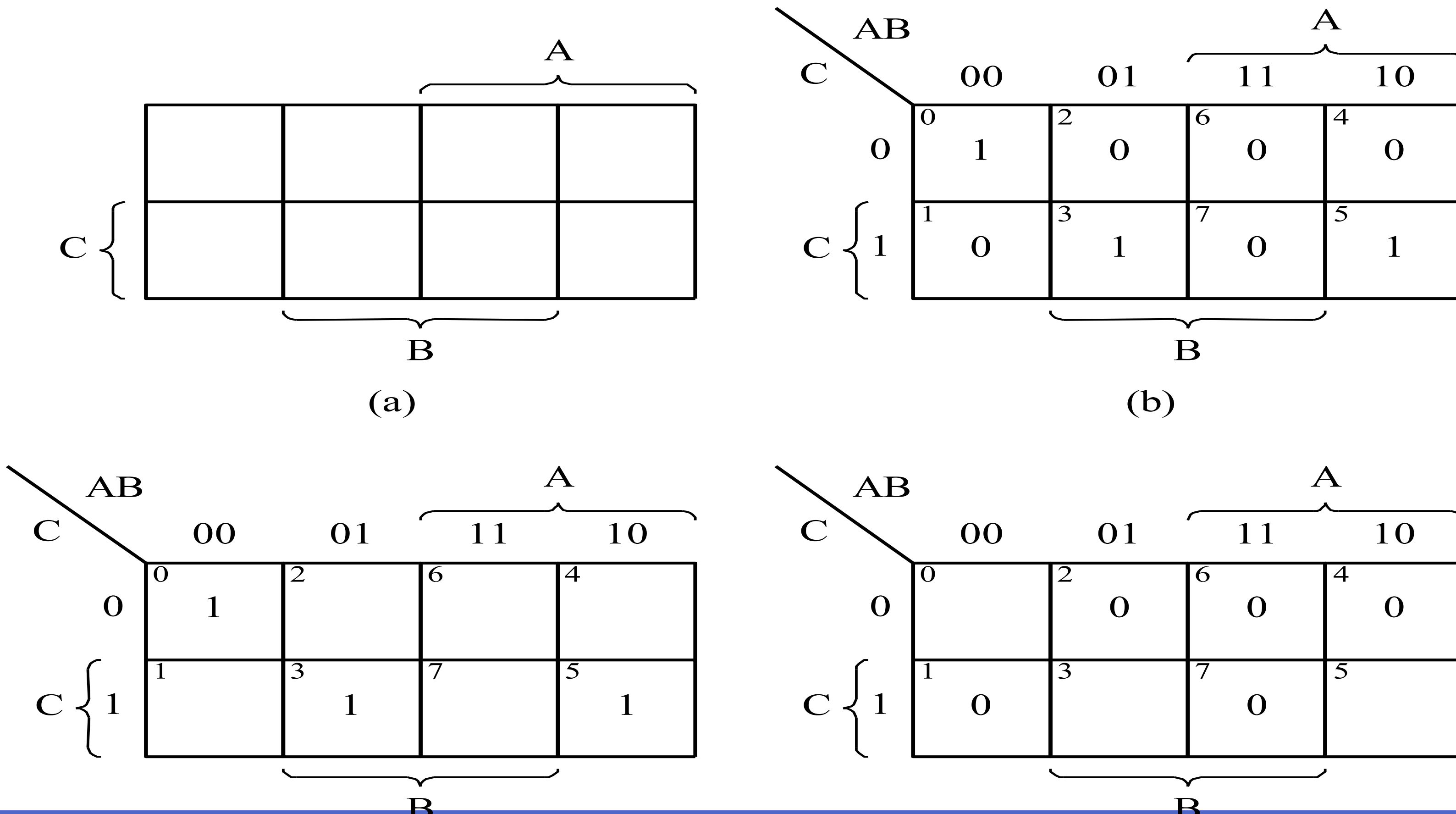
- Plotting the function f on the k-map

$f(A,B,C) = \sum m(0,3,5) = \prod M(1,2,4,6,7)$

$f(a,b,Q,G) = \sum m(0,3,5,7,10,11,12,13,14,15) = \prod M(1,2,4,6,8,9)$

K-Map: Sample 4 (cont'd)

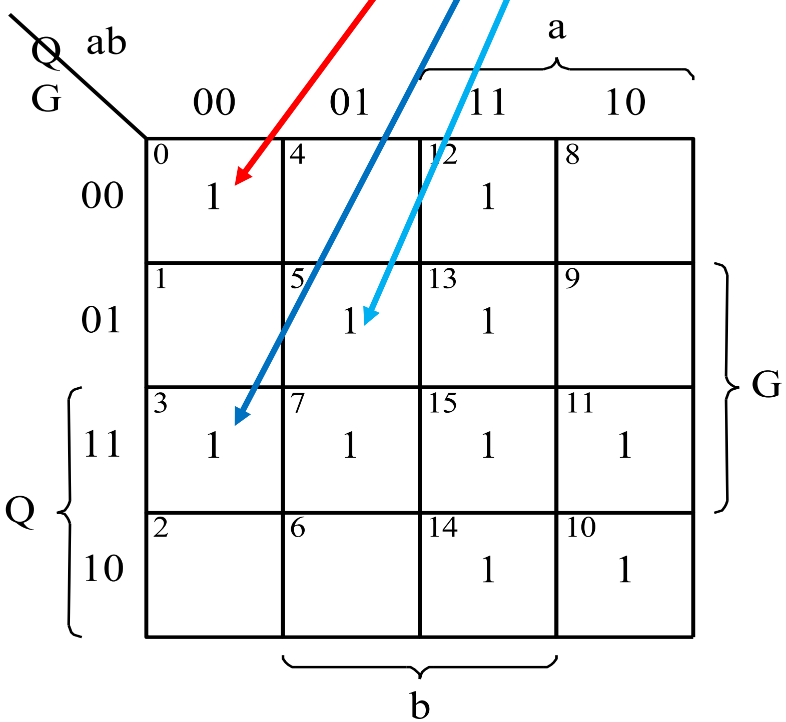
f(A,B,C) = $\sum m(0,3,5) = \prod M(1,2,4,6,7)$



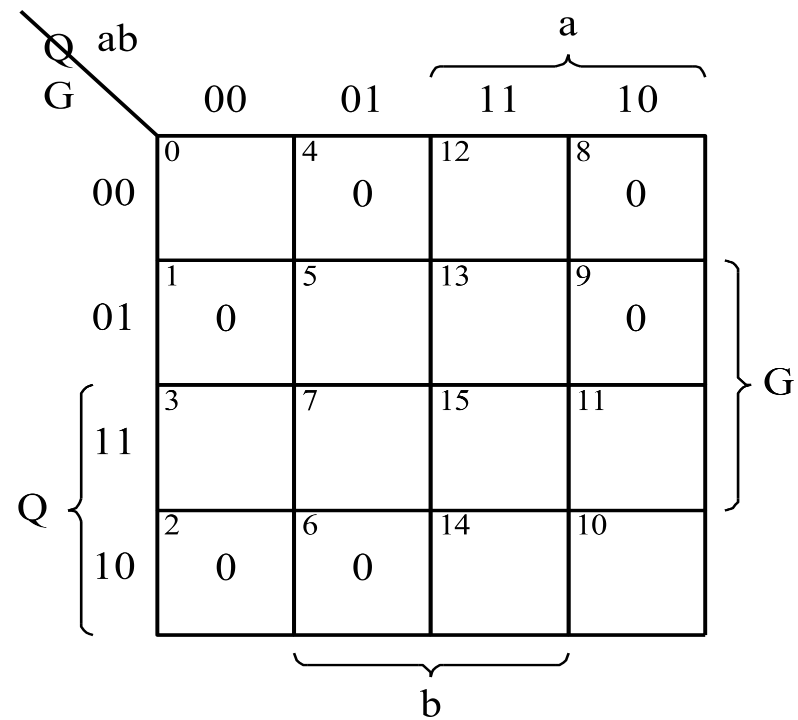
(a) (b) (c) (d)

K-Map: Sample 4 (cont’d)

$$f(a,b,Q,G) = \sum m(0,3,5,7,10,11,12,13,14,15) = \prod M(1,2,4,6,8,9)$$



(a)



(b)

K-Map: Sample 5

- Plotting the function f on the k-map

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

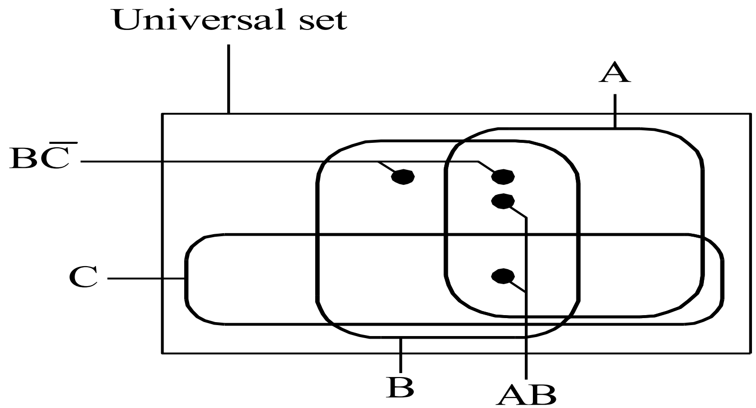
$$f(A,B,C,D) = (A + C)(B + C)(B' + C' + D)$$

$$f(A,B,C,D) = (A'+B')(A'+C+D')(B'+C'+D')$$

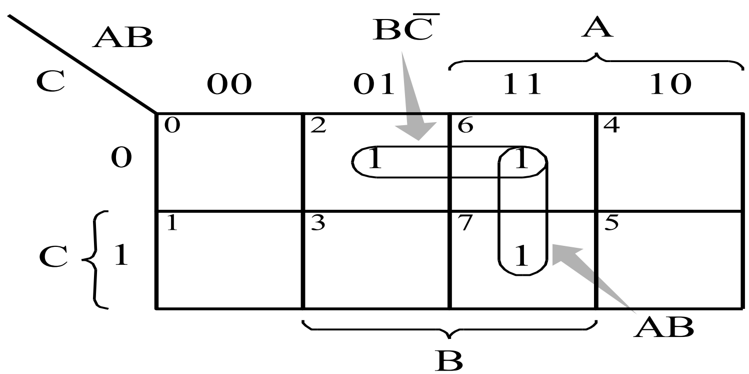
K-Map: Sample 6 (cont’d)

f(A,B,C) = AB + BC′

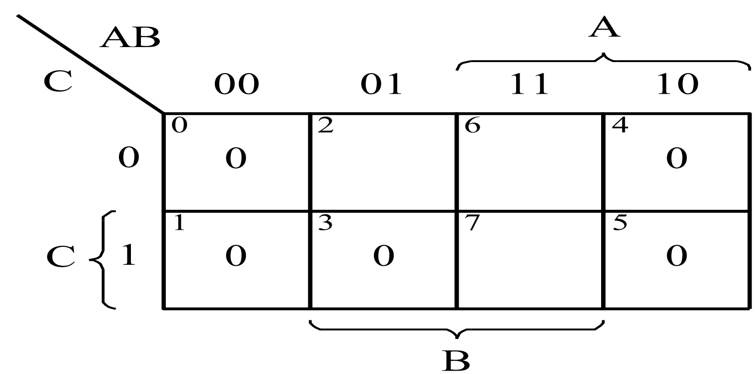
(a) Venn diagram form
(b) Sum of minterms
(c) Maxterms



(a)



(b)

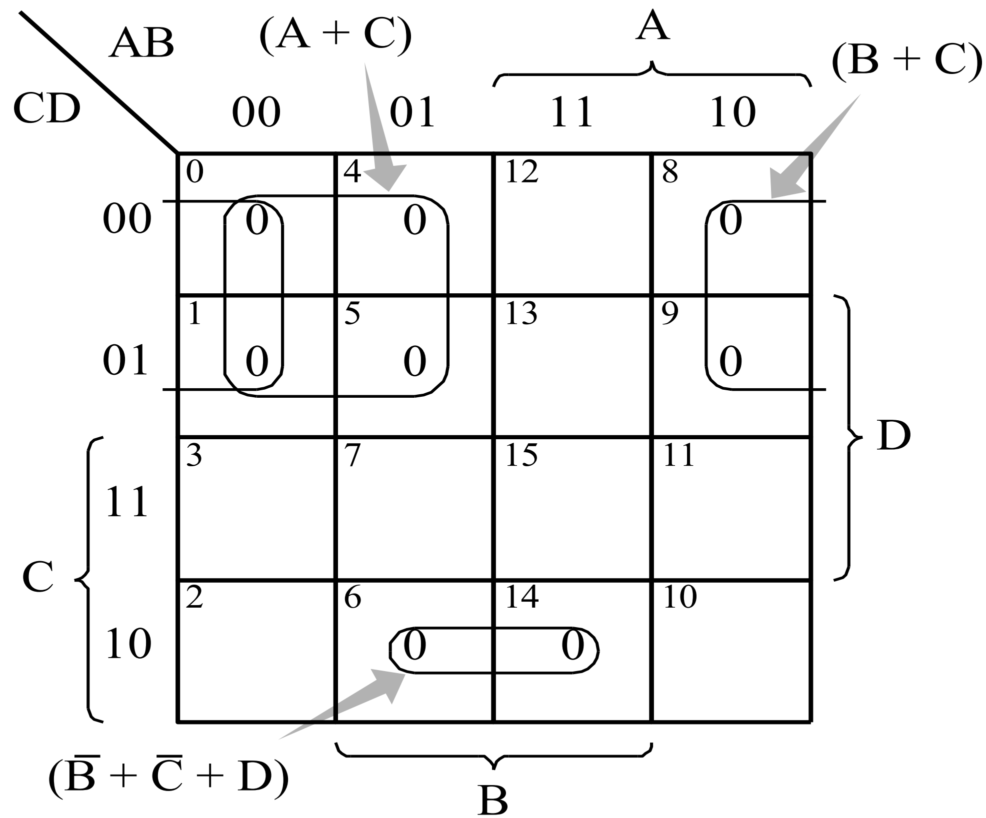


(c)

K-Map: Sample 6 (cont’d)

$f(A,B,C,D) = (A + C)(B + C)(B' + C' + D)$

(a) Maxterms

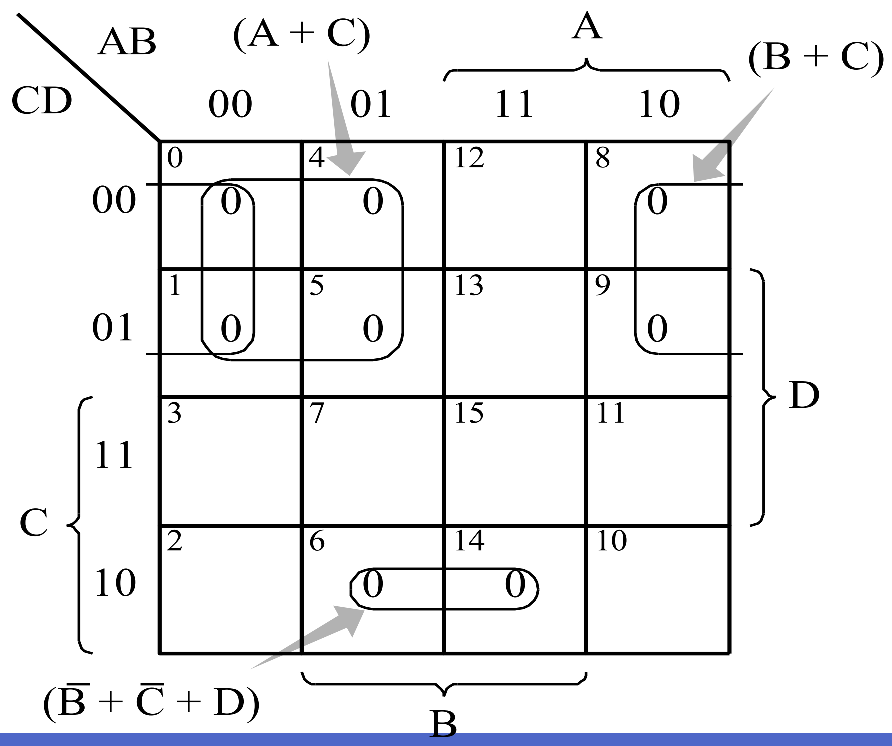


(a)

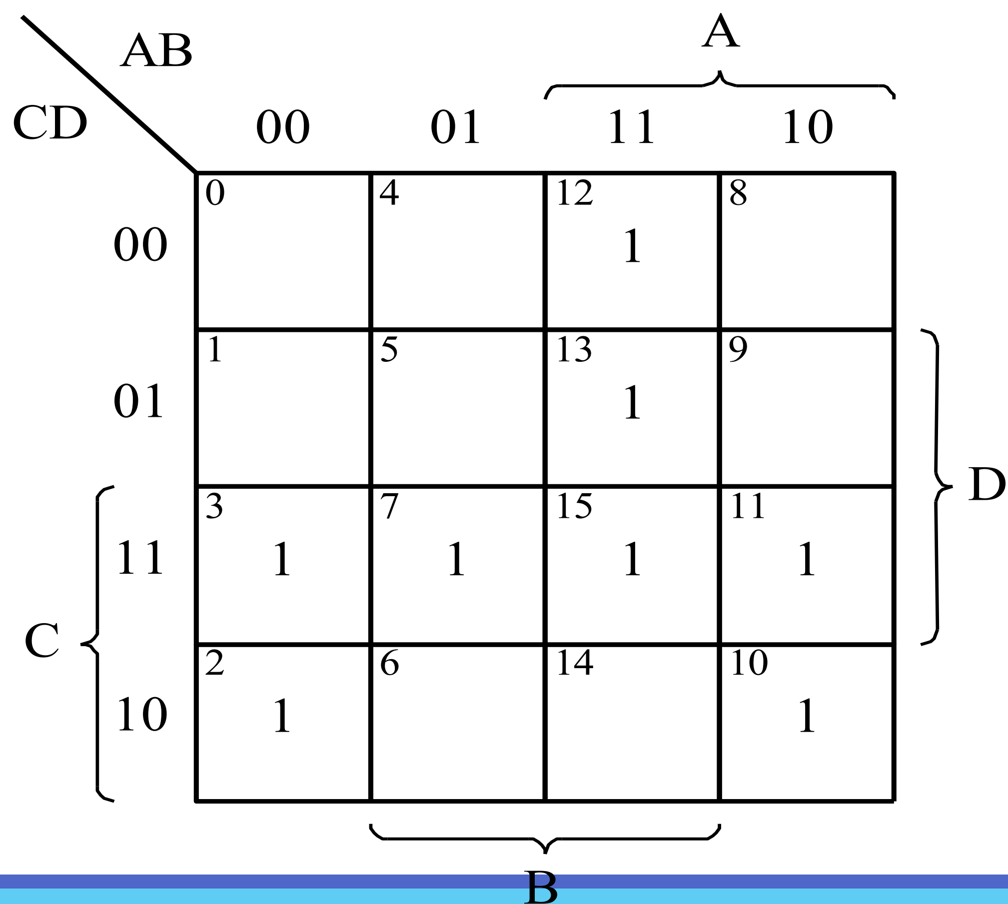
K-Map: Sample 6 (cont'd)

f(A,B,C,D) = (A + C)(B + C)(B′ + C′ + D)

(a)Maxterms
(b) minterms



(a)



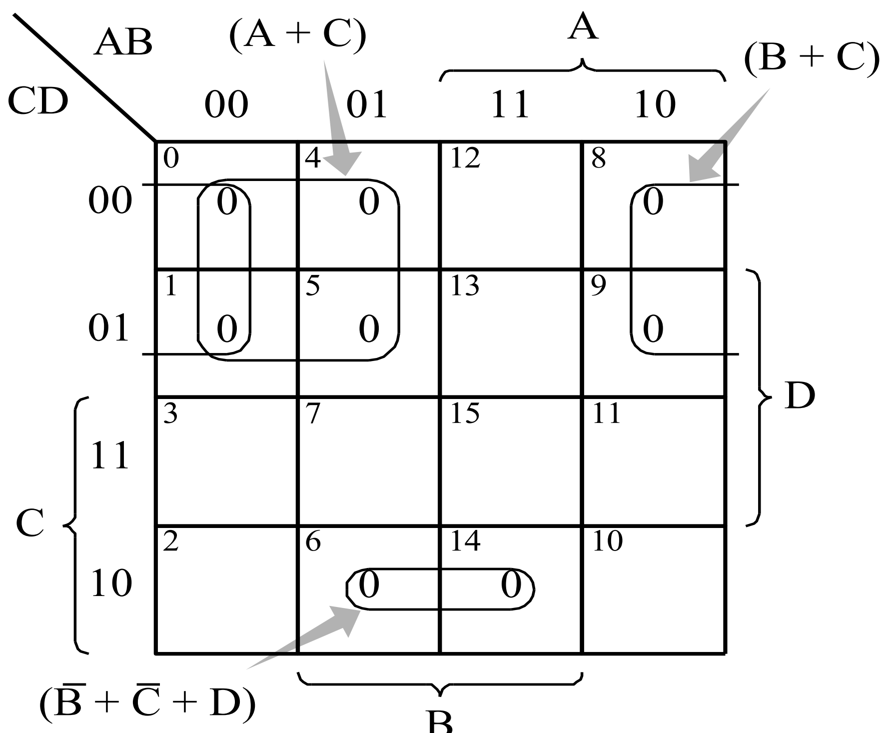
(b)

K-Map: Sample 6 (cont'd)

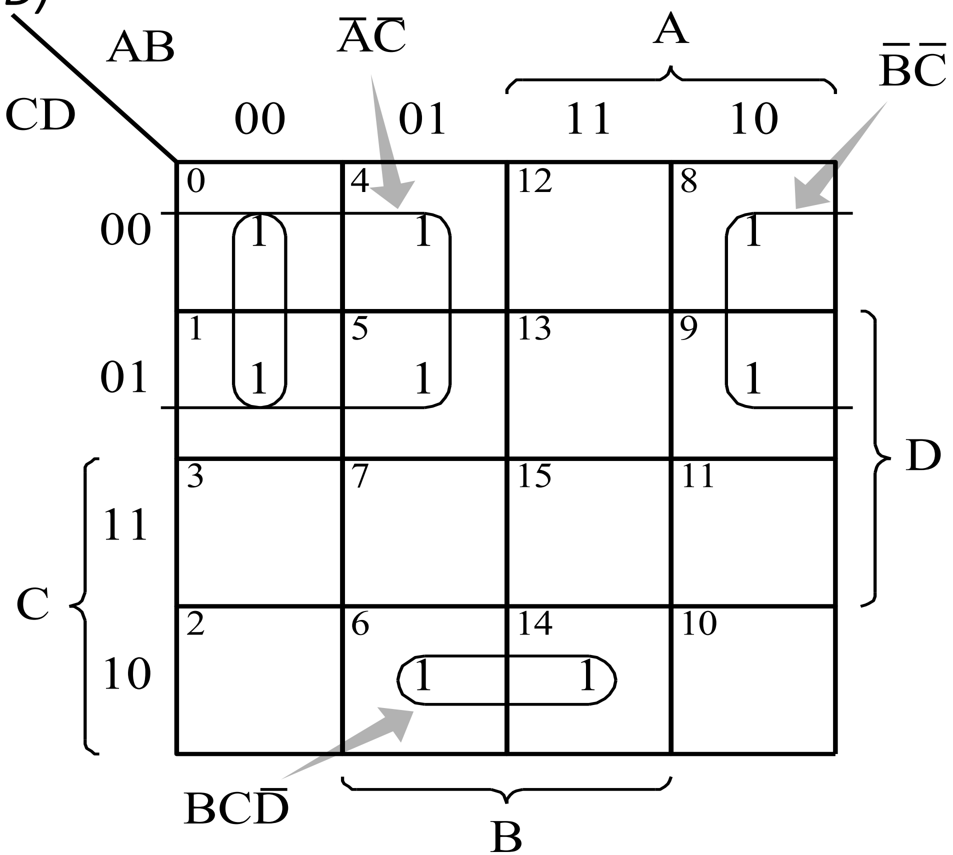
$f(A,B,C,D) = (A + C)(B + C)(B' + C' + D)$

(a) Maxterms of f

(c) Minterms of $f\,'$



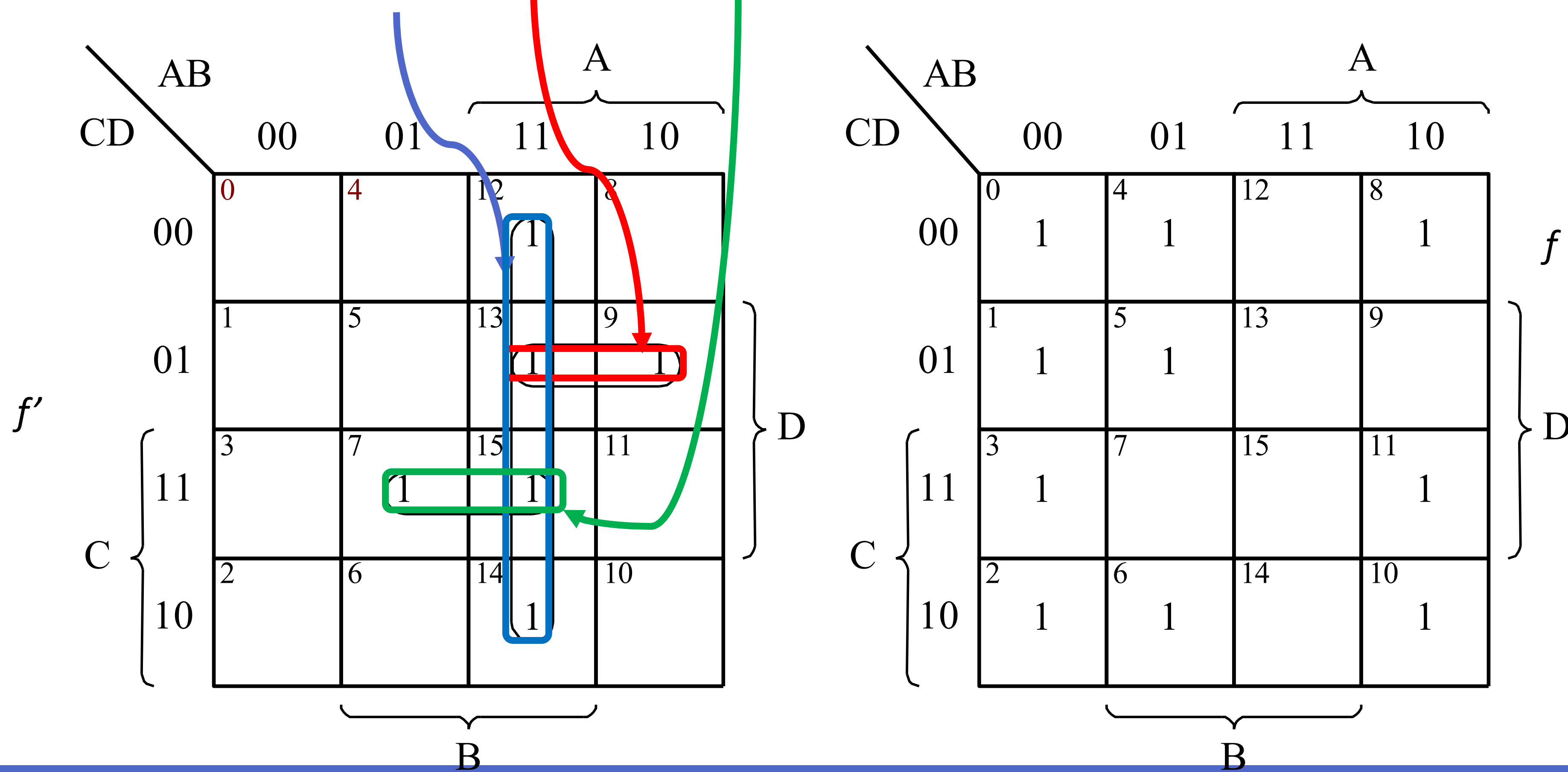
(a)



(c)

K-Map: Sample 6 (cont'd)

f(A,B,C,D)= (A′+B′)(A′+C+D′)(B′+C′+D′)



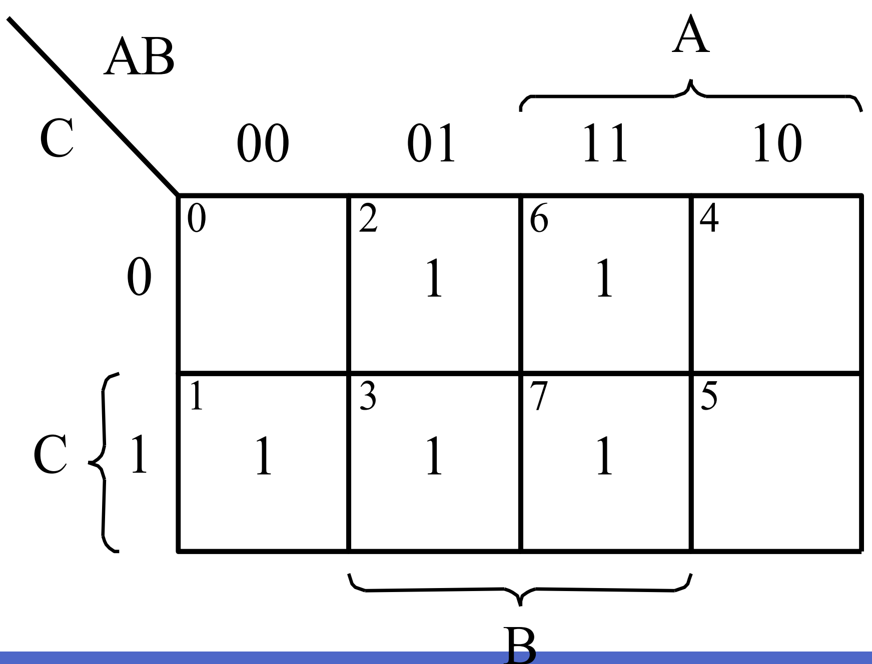
(a) (b)

K-Map: Simplification of Switching Functions

- Adjacent cells
  - K-map cells that are **physically** adjacent are also **logically adjacent**
  - Cells on an edge are **logically** adjacent to cells on the opposite edge
- Combination
  - If two logically adjacent cells both contain logical 1s
  - Eliminate the variable that has value 1 in one cell's label and value 0 in the other
  - $= aP + a' P = P$
  - => Reduces **number of literals** in a product term
  - => Reducing the **gate input** cost

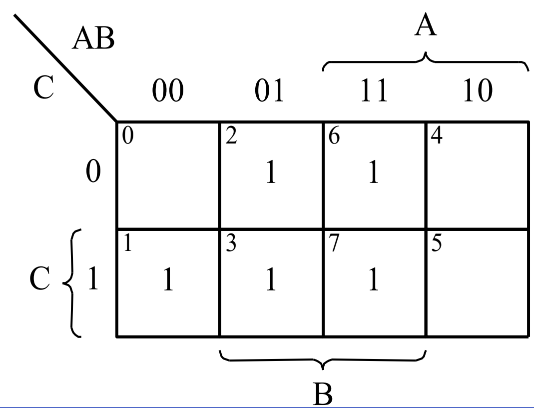
K-Map: Simplification: Guidelines

- Each cell of an *n*-variable K-map has *n* logically adjacent cells
- Combine cells in power of 2; e.g., 2, 4, 8, …, 2^k



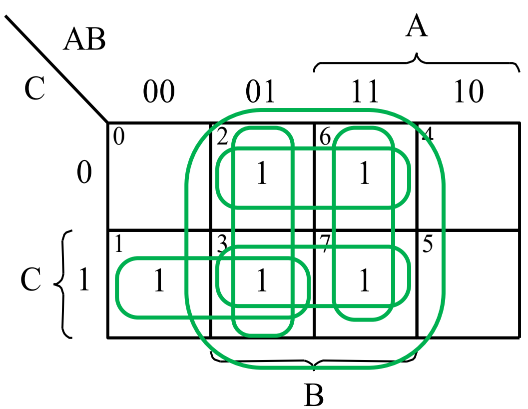
K-Map: Combination

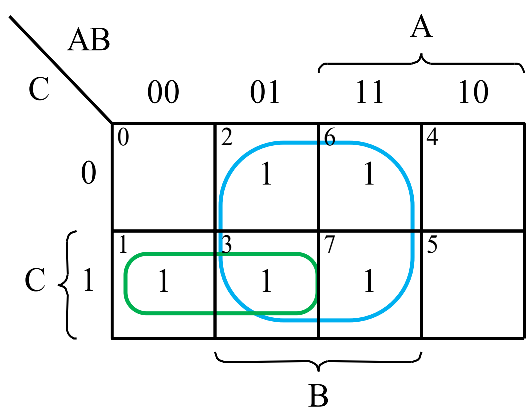
- A group of cells can be combined
  - If all cells in the group have the same value for some set of variables
  - Combine **as many cells in a group as** possible
    - => **Fewest** number of literals in the term that represents the group
  - Make **as few groupings** as possible to **cover all minterms**
    - => **Fewest** product terms
  - Cover All minterms
    - A minterm is covered if it is included in **at least one group**
    - Each minterm may be used **as many times as** it needs
    - As soon as all minterms are used once, stop.

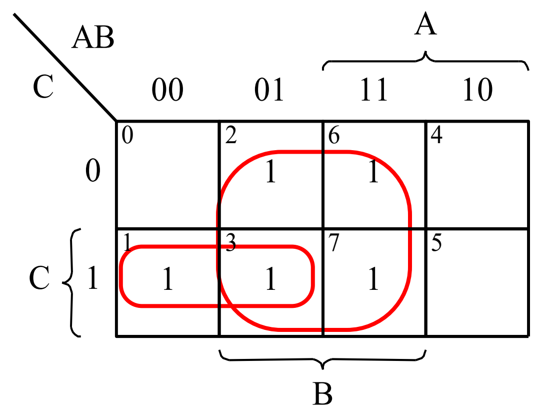


K-Map: Simplification: Terms

- Implicant
  - A product term that can cover minterms of a function.
- Prime implicant
  - A product term that is not covered by another implicant of the function.
  - Combining the **maximum possible number of adjacent squares**
- Essential prime implicant
  - A prime implicant that covers at least one minterm that is not covered by any other prime implicant.

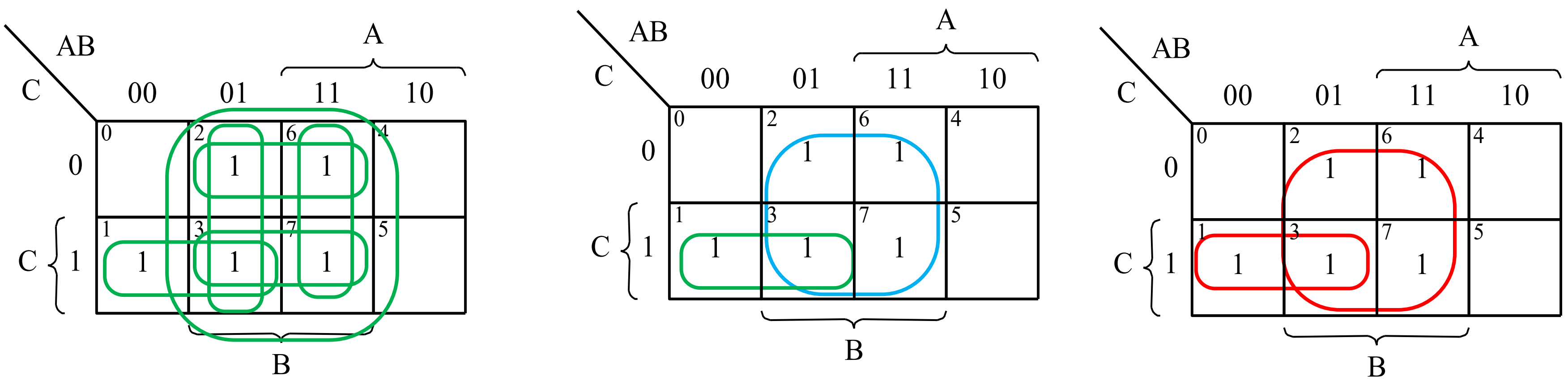




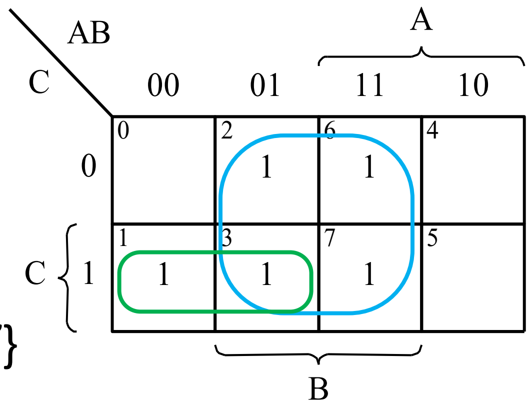


K-Map: Simplification: Terms

- Cover of a function
  - A set of implicants
  - If each minterm of the function is covered by at least one implicant in the set.
- Minimal Cover of a function
  - A cover
  - Contains the smallest number of prime implicants and the smallest number of literals



K-Map: Simplification: Terms (cont’d)



Minterms: $\{A'B'C,\ A'BC',\ A'BC,\ ABC',\ ABC\}$

Groups of two minterms: $\{A'B,\ AB,\ A'C,\ BC',\ BC'\}$

Groups of four minterms: $\{B\}$

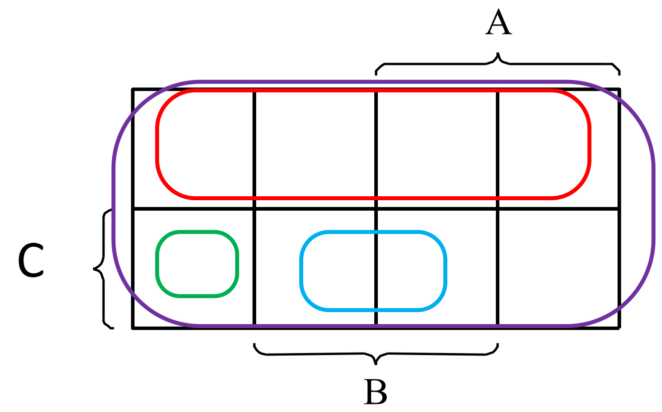
Prime implicants: $\{A'C,\ B\}$

Cover = $\{A'C,\ B\}$

MSOP = $A'C + B$

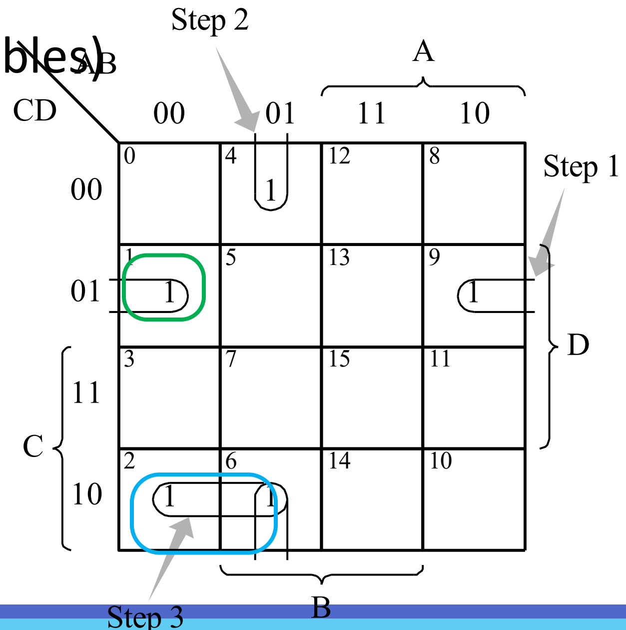
Simplification of 3-variable K-Map

- One square represents a Minterm with 3 variables; e.g., A'B'C
- Two adjacent squares represent a term with 2 variables; e.g., BC'
- Four adjacent squares represent a term with 1 variable; e.g., C
- Eight adjacent square is the function 1 (no variables)



Simplification of 4-variable K-Map

- One square represents a Minterm with 4 variables
- Two adjacent squares represent a term with 3 variables
- Four adjacent squares represent a term with 2 variable
- Eight adjacent squares represent a term with 1 variable
- Sixteen adjacent square is the function 1 (no variables)



K-Map: Algorithm 3.1

- Generating and selecting prime implicants

1. Count the number of adjacencies for each minterm on the K-map
2. Select an uncovered minterm with the **fewest** number of adjacencies.
   - Make an arbitrary choice if more than one choice is possible
3. Generate a prime implicant for this minterm and put it in the cover
   - If this minterm is covered by more than one prime implicant, select the one that covers the **most uncovered minterms**
4. Repeat steps 2 and 3 until all minterms have been covered

K-Map: Algorithm 3.1 (cont'd)

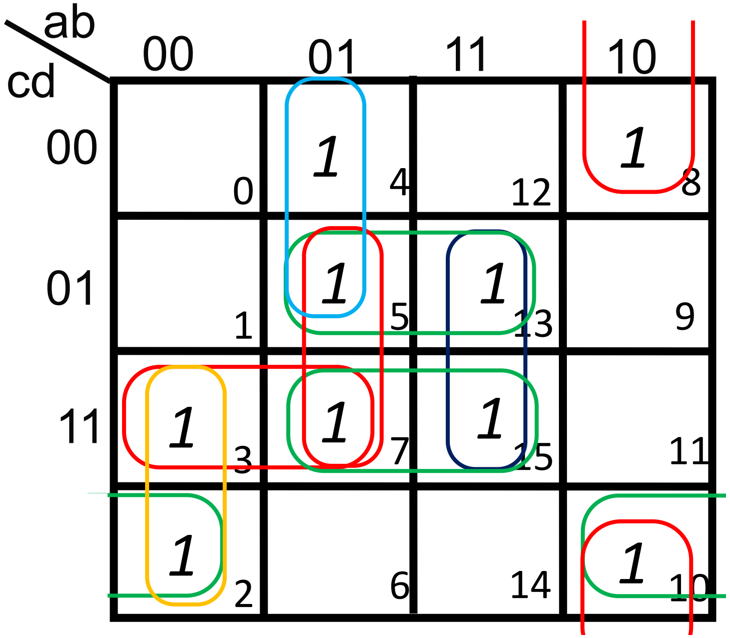
- Example: $F(a,b,c,d) = \sum m(2,3,4,5,7,8,10,13,15)$

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

K-Map: Algorithm 3.1 (cont’d)

- Example: $F(a,b,c,d) = \sum m(2,3,4,5,7,8,10,13,15)$

1. Count the number of adjacencies for each minterm on the K-map.
   - m_4: one adjacency
   - m_8: one adjacency
   - m_2: two adjacencies
   - m_3: two adjacencies
   - m_{10}: two adjacencies
   - m_{13} : two adjacencies
   - m_{15} : two adjacencies
   - m_5 : two adjacencies
   - m_7 : two adjacencies



K-Map: Algorithm 3.1 (cont'd)

- Example: $F(a,b,c,d) = \sum m(2,3,4,5,7,8,10,13,15)$

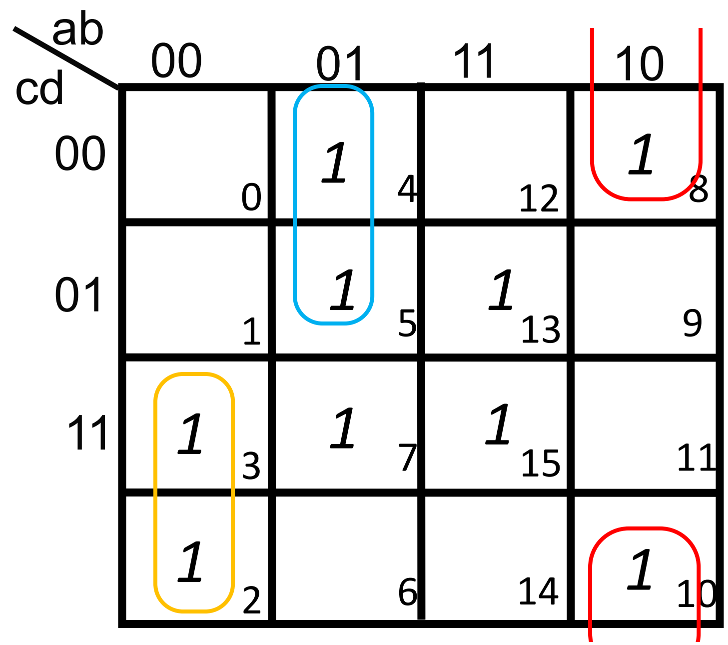
2. Select an uncovered minterm with the fewest number of adjacencies.
3. Generate a prime implicant (a product term that is not covered by another implicant of the function)
   - Begin with m_4 and m_8
   - Prime implicant 4-5
   - Prime implicant 8-10

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

K-Map: Algorithm 3.1 (cont'd)

- Example: $F(a,b,c,d) = \sum m(2,3,4,5,7,8,10,13,15)$

2. Select an uncovered minterm with the fewest number of adjacencies.
3. Generate a prime implicant (a product term that is not covered by another implicant of the function)
   - Consider m_2 ,m_3,
   - Prime implicant 2-3 or 2-10?



K-Map: Algorithm 3.1 (cont'd)

- Example: $F(a,b,c,d) = \sum m(2,3,4,5,7,8,10,13,15)$

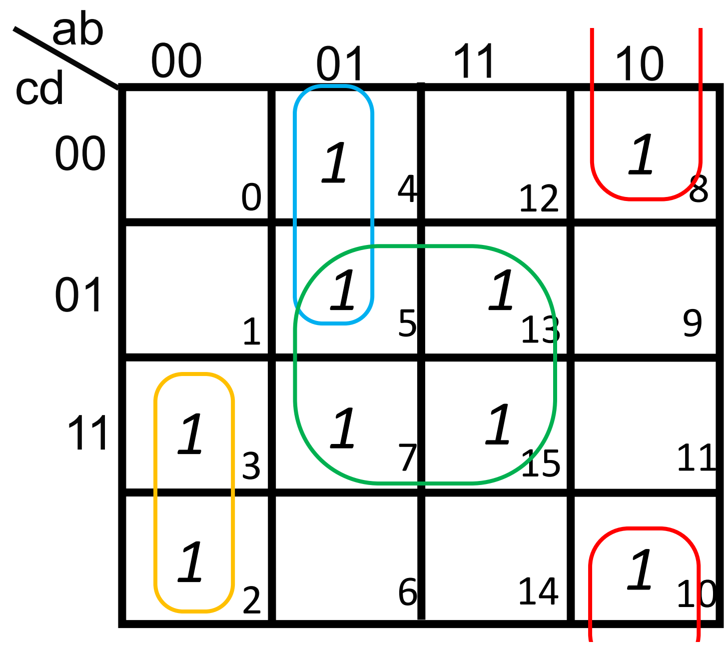
2. Select an uncovered minterm with the fewest number of adjacencies.
3. Generate a prime implicant (a product term that is not covered by another implicant of the function)
   - Consider m_2, m_3,
   - Prime implicant 2-3 or 2-10?
   - Prime implicant 2-3
   - Covers more uncovered minterms

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

K-Map: Algorithm 3.1 (cont'd)

- Example: $F(a,b,c,d) = \sum m(2,3,4,5,7,8,10,13,15)$

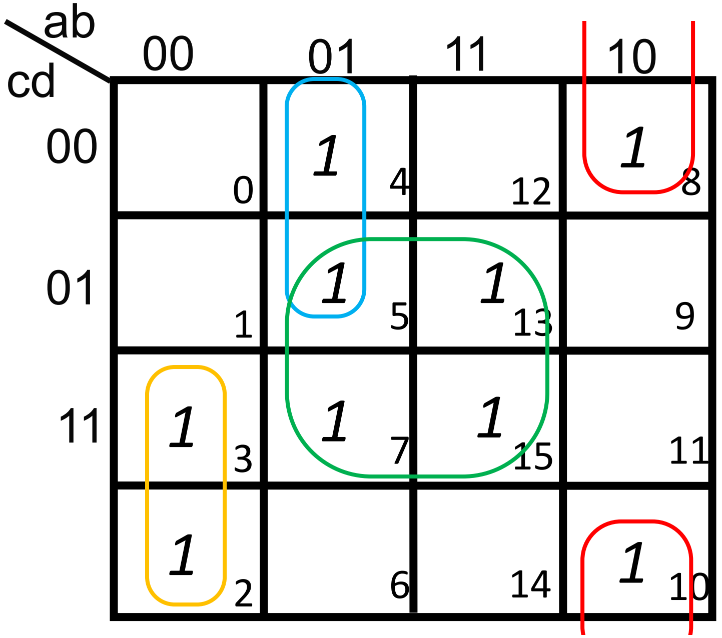
2. Select an uncovered minterm with the fewest number of adjacencies.
3. Generate a prime implicant (a product term that is not covered by another implicant of the function)
   - Consider m_{13}, m_{15},
   - Prime implicant 5-7-13-15



K-Map: Algorithm 3.1 (cont'd)

- Example: $F(a,b,c,d) = \sum m(2,3,4,5,7,8,10,13,15)$

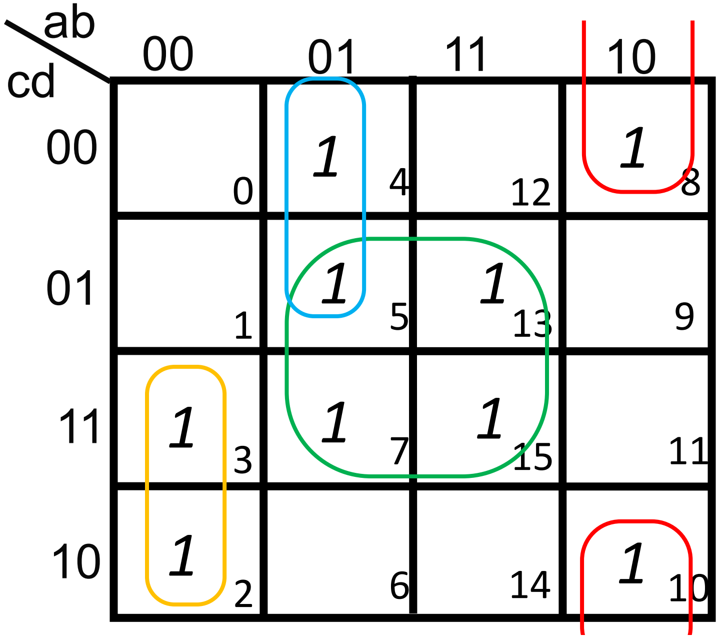
4. Repeat steps 2 and 3 until all minterms have been covered.
   - STOP



K-Map: Algorithm 3.1: Sample

$F(a,b,c,d) = \sum m(2,3,4,5,7,8,10,13,15)$

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



K-Map: Algorithm 3.2

- Generating and selecting prime implicants (revisited)

1. Circle all prime implicants on the K-map
2. Identify and select all essential prime implicants for the cover
3. Select a minimum subset of the remaining prime implicants to complete the cover
    - To cover those minterms not covered by the essential prime implicants

K-Map: Algorithm 3.2 (cont'd)

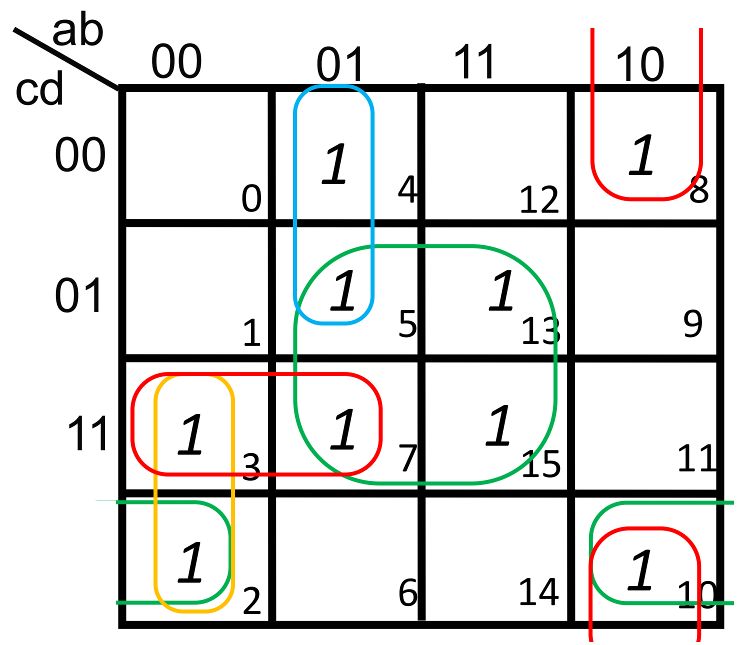
- Example: $F(a,b,c,d) = \sum m(2,3,4,5,7,8,10,13,15)$

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

K-Map: Algorithm 3.2 (cont'd)

- Example: $F(a,b,c,d) = \sum m(2,3,4,5,7,8,10,13,15)$

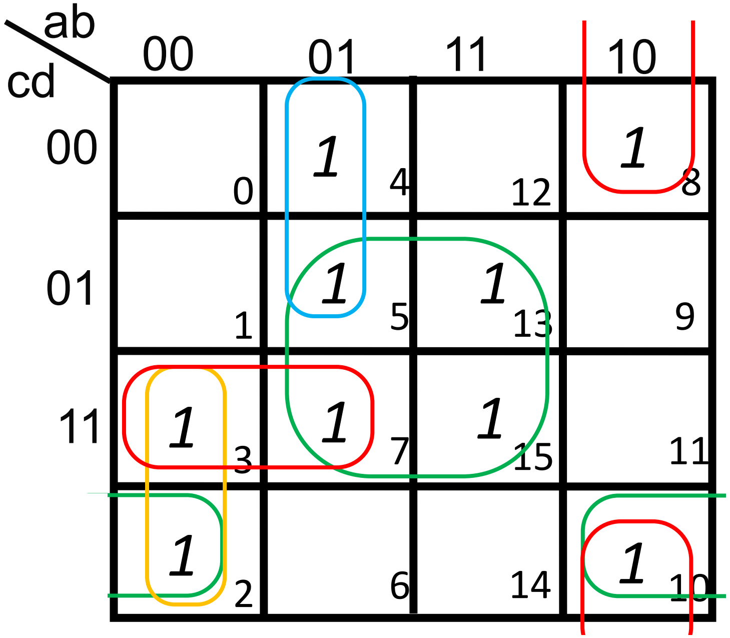
1. Circle all prime implicants (a product term that is not covered by another implicant of the function) on the K-map
   - 2-3
   - 3-7
   - 4-5
   - 5-7-13-15
   - 8-10
   - 2-10



K-Map: Algorithm 3.2 (cont'd)

- Example: $F(a,b,c,d) = \sum m(2,3,4,5,7,8,10,13,15)$

2. Identify and select all essential prime implicants (a prime implicant that covers at least one minterm that is not covered by any other prime implicant) for the cover



K-Map: Algorithm 3.2 (cont'd)

- Example: $F(a,b,c,d) = \sum m(2,3,4,5,7,8,10,13,15)$

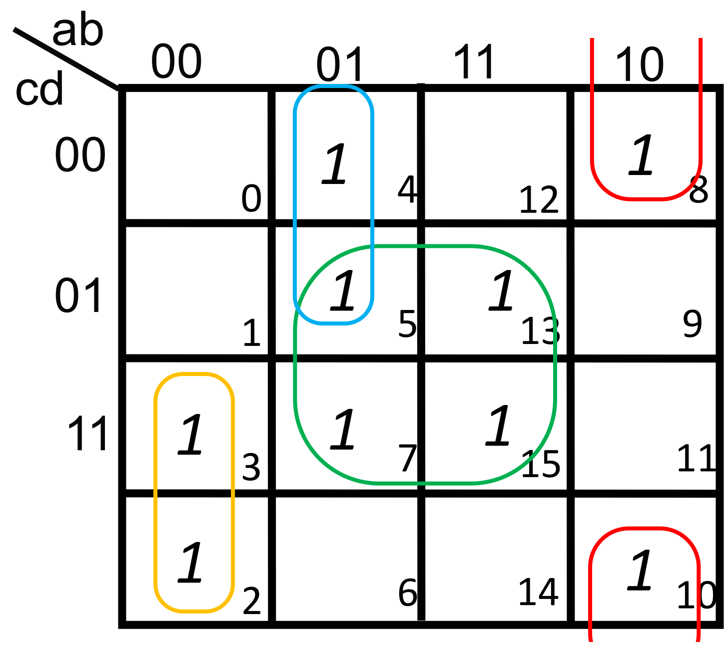
2. Identify and select all essential prime implicants (a prime implicant that covers at least one minterm that is not covered by any other prime implicant) for the cover.
   - 4-5
   - 5-7-13-15
   - 8-10

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

K-Map: Algorithm 3.2 (cont'd)

- Example: $F(a,b,c,d) = \sum m(2,3,4,5,7,8,10,13,15)$

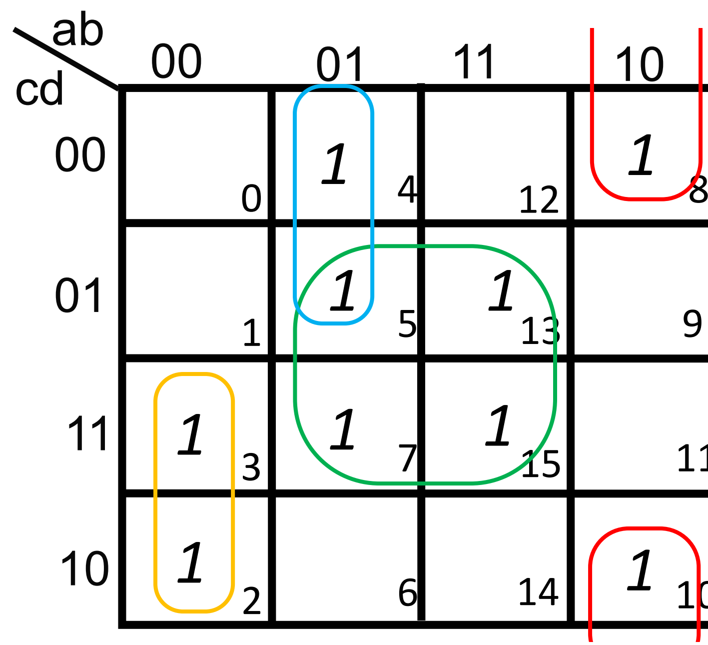
3. Select a minimum subset of the remaining prime implicants to complete the cover, that is, to cover those minterms not covered by the essential prime implicants
   - 2-3



K-Map: Algorithm 3.2: Sample

$F(a,b,c,d) = \sum m(2,3,4,5,7,8,10,13,15)$

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



K-Map: Simplification: Sample 7

$f(x, y) = \sum m(0,2,3)$

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

K-Map

y \ x	x= 0	x= 1
y= 0	1 $_{0}$	1 $_{2}$
y= 1	$_{1}$	1 $_{3}$

K-Map: Simplification: Sample 8

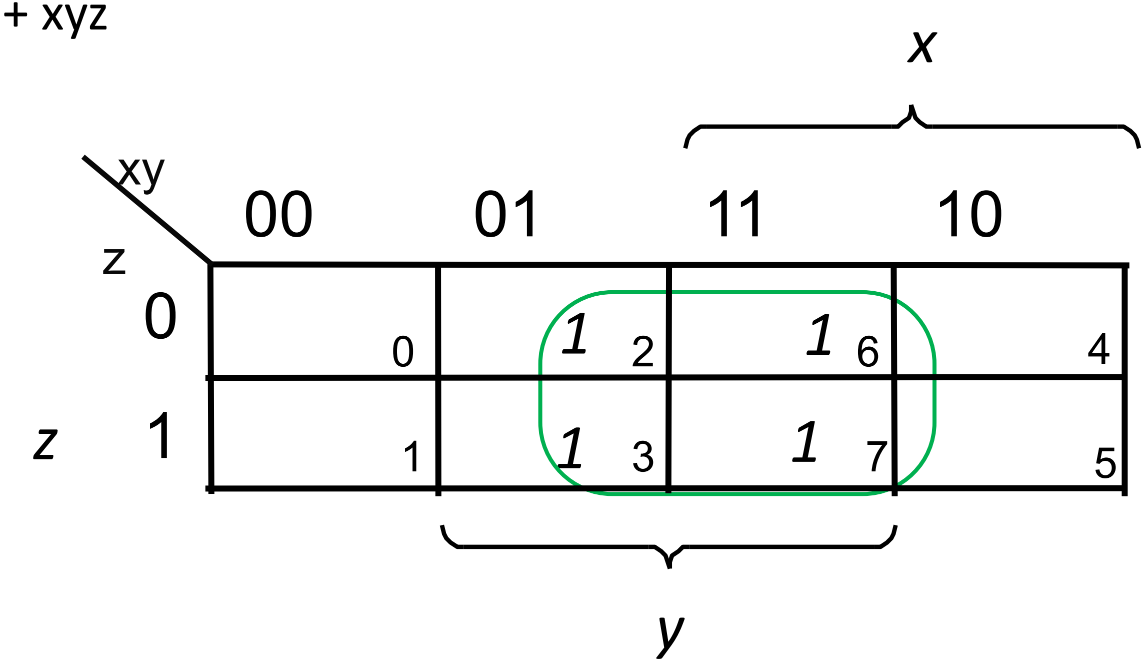
$f(x,y,z) = \sum m(2,3,6,7)$

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

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

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

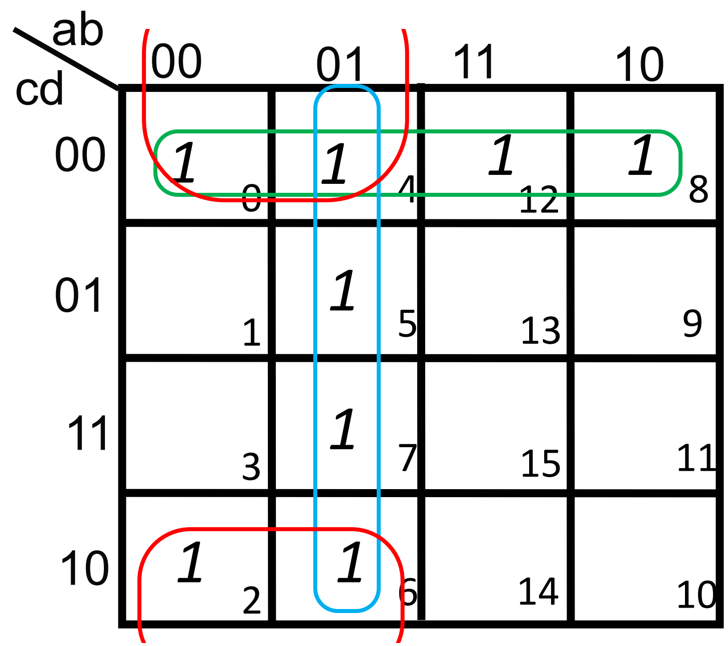
$f(x,y,z) = y$



K-Map: Simplification: Sample 9

$f(a,b,c,d) = \sum m(0,2,4,5,6,7,8,12)$

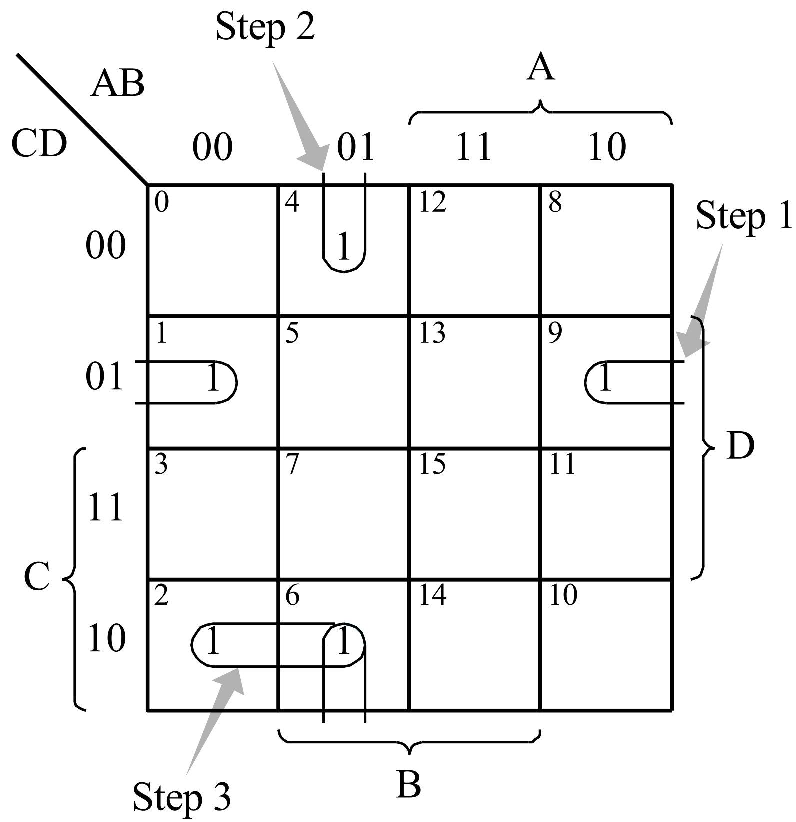
$f(a,b,c,d) = c'd' + a'b + a'd'$



K-Map: Simplification: Sample 10

f(A,B,C,D) = ∑*m*(1,2,4,6,9)

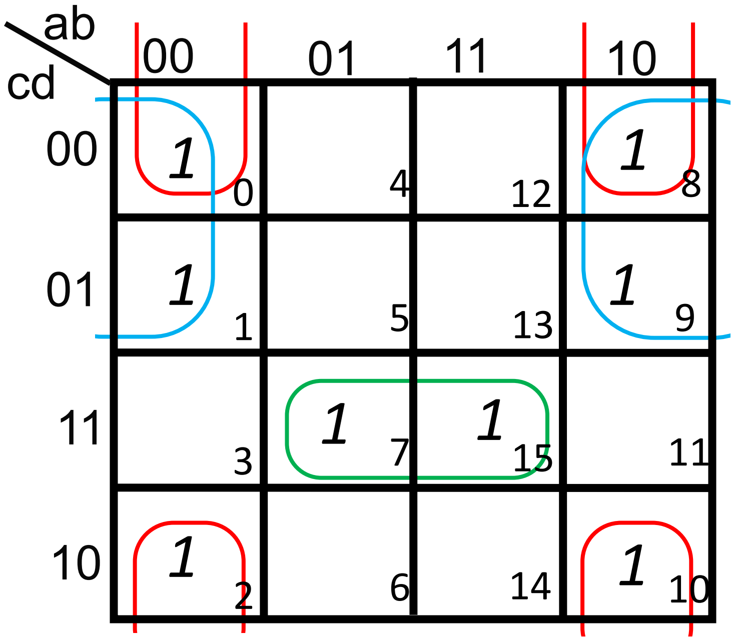
f(A,B,C,D) = B'C'D +A'BD' + + A'CD'



K-Map: Simplification: Samples 11

$F(a,b,c,d) = \sum m(0,1,2,7,8,9,10,15)$

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



K-Map: Simplification: Samples 12

$F(A,B,C,D,E) = \sum m(0,2,4,7,10,12,13,18,23,26,28,29)$

$F(A,B,C,D,E) = B'CDE + BCD' + C'DE' + A'B'D'E'$

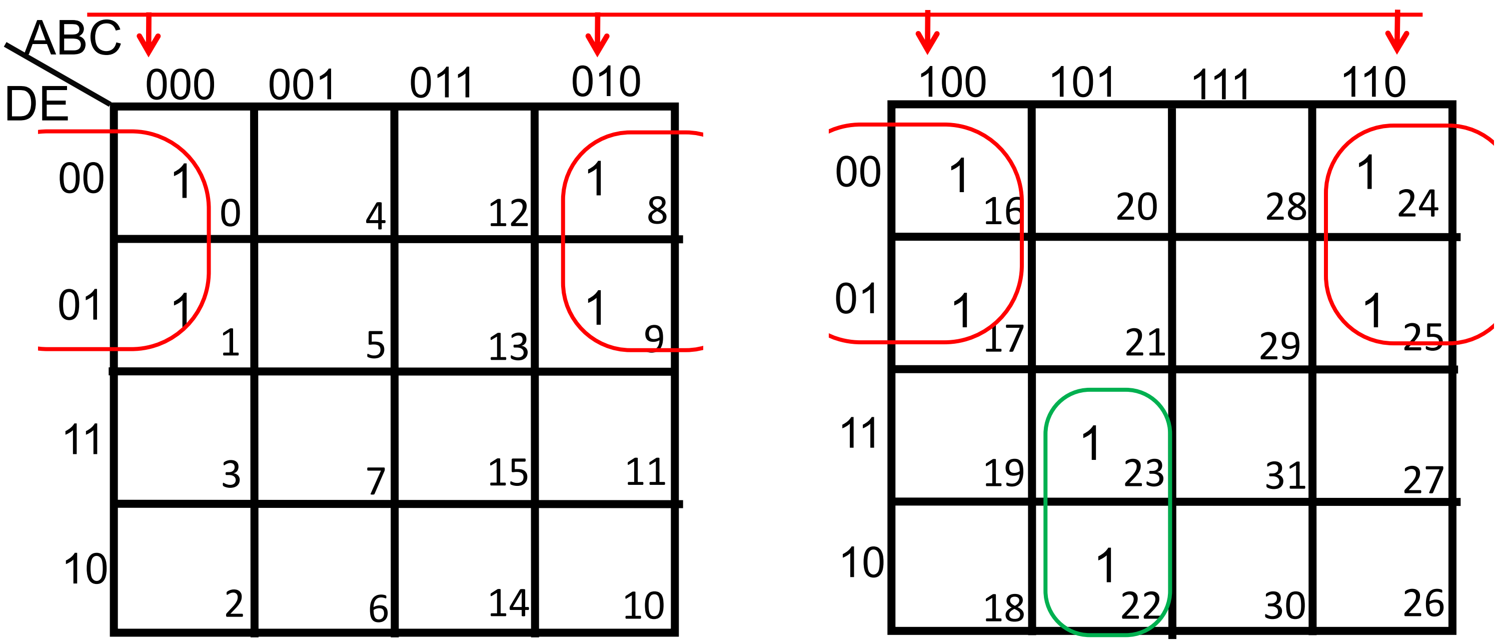
DE \ ABC	000	001	011	010
00	1 (0)	1 (4)	1 (12)	8
01	1	5	1 (13)	9
11	3	1 (7)	15	11
10	1 (2)	6	14	1 (10)

DE \ ABC	100	101	111	110
00	16	20	1 (28)	24
01	17	21	1 (29)	25
11	19	1 (23)	31	27
10	1 (18)	22	30	1 (26)

K-Map: Simplification: Samples 13

$F(A,B,C,D,E) = \sum m(0, 1, 8, 9, 16, 17, 22, 23, 24, 25)$

$F(A,B,C,D,E) =$ $AB'CD$ $+$ $C'D'$



K-Map: Simplification: Samples 14

$F(W,X,Y,Z) = \sum m(1, 2, 3, 9, 10, 11, 13, 14, 15)$

$F(A,B,C,D,E) =$

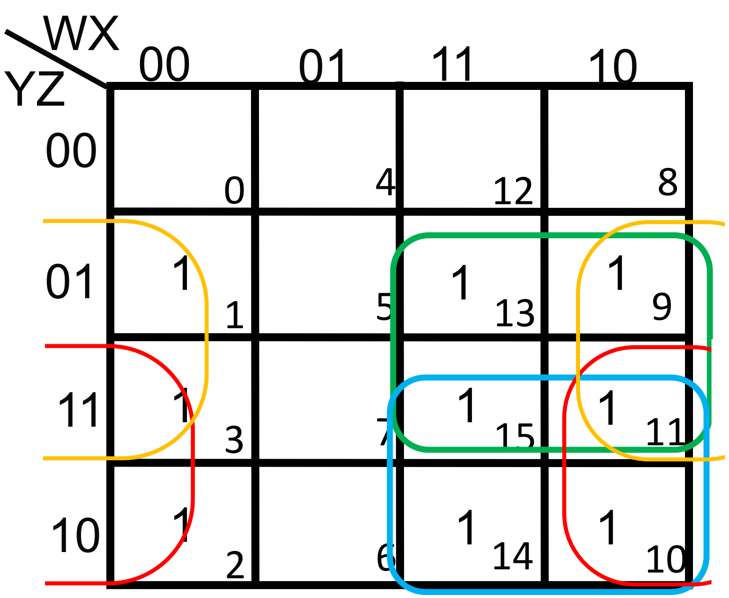
$F'(A,B,C,D,E) =$

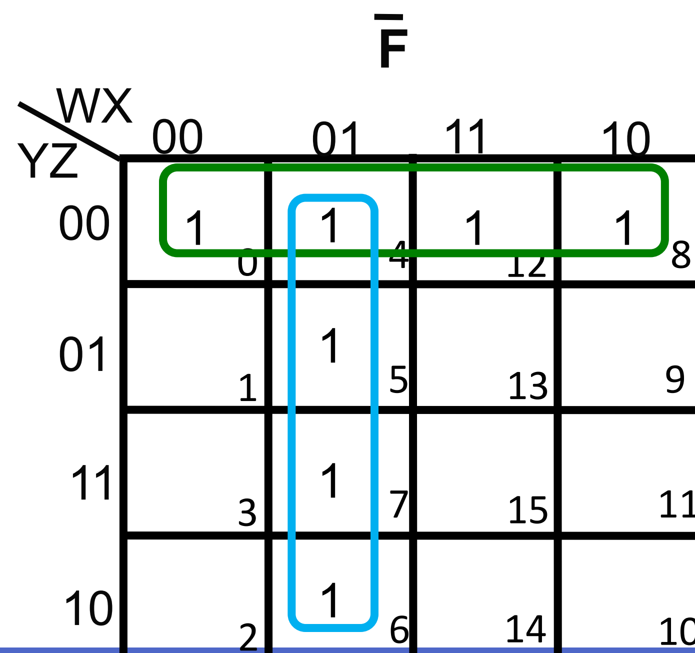
YZ \ WX	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

K-Map: Simplification: Samples 14

$F(W,X,Y,Z) = \sum m(1, 2, 3, 9, 10, 11, 13, 14, 15)$

$F(A,B,C,D,E) = WZ + WY + X'Y + X'Z$ $F'(A,B,C,D,E) = Y'Z' + W'X$

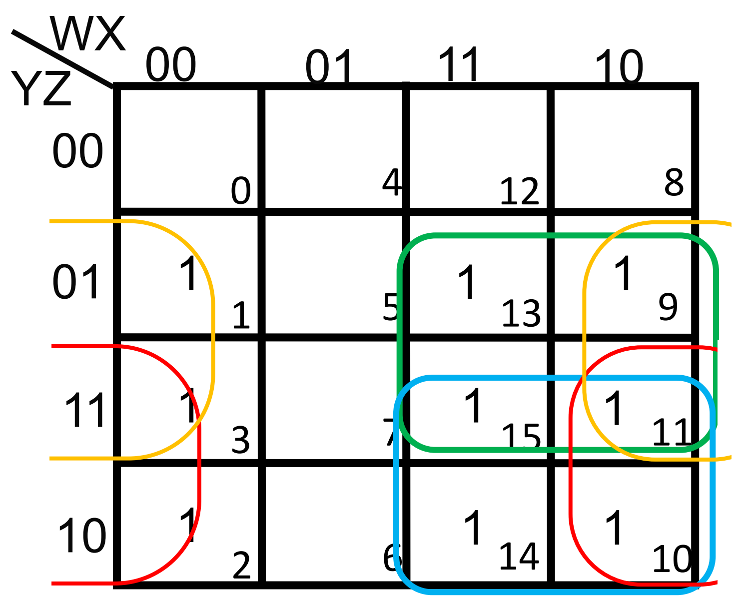




K-Map: Simplification: Samples 15

$F(W,X,Y,Z) = \sum m(1, 2, 3, 9, 10, 11, 13, 14, 15)$

$F(A,B,C,D,E) = WZ + WY + X'Y + X'Z$



K-Map: Simplification: POS

- How about POS?

K-Map: POS: Terms

- Implicant
  - A product term that can cover maxterm of a function.
- Prime implicant
  - A sum term that is not covered by another implicant of the function.
- Essential prime implicant
  - A prime implicant that covers at least one maxterm that is not covered by any other prime implicant.

K-Map: POS: Terms

- Cover of a function
  - A set of implicants
  - If each maxterm of the function is covered by at least one implicant in the set.
- Minimal Cover of a function
  - A cover
  - Contains the smallest number of prime implicants and the smallest number of literals.

K-Map: Algorithm 3.1

- Generating and selecting prime implicants

1. Count the number of adjacencies for each maxterm on the K-map.
2. Select an uncovered maxterm with the fewest number of adjacencies. Make an arbitrary choice if more than one choice is possible.
3. Generate a prime implicant for this maxterm and put it in the cover.
    - If this maxterm is covered by more than one prime implicant, select the one that covers the most uncovered maxterms.
4. Repeat steps 2 and 3 until all maxterms have been covered.

K-Map: Algorithm 3.2

- Generating and selecting prime implicants (Revisited)

1. Circle **all prime implicants** on the K-map
2. Identify and select **all essential prime implicants** for the cover.
3. Select a **minimum subset** of the **remaining prime implicants** to complete the cover, that is, to cover those maxterms not covered by the essential prime implicants

K-Map: Simplification: POS

- Draw the K-map for F', replacing the 0's of F with 1's in F' and vice versa
- Obtain a minimal Sum-of-Product (SOP) expression for F'
- Use DeMorgan's Theorem to obtain F'' = F
  - Result is a minimal Product-of-Sum (POS) expression for F
- Cost Analysis
  - Compare the cost of the minimal SOP and POS expressions
  - Select the best one

K-Map: Simplification: Samples 16

$F(W,X,Y,Z) = \prod M(0,1,2,3,6,9,14)$

$F(A,B,C,D,E) = (W+X)\ (X'+Y'+Z)\ (X+Y+Z')$

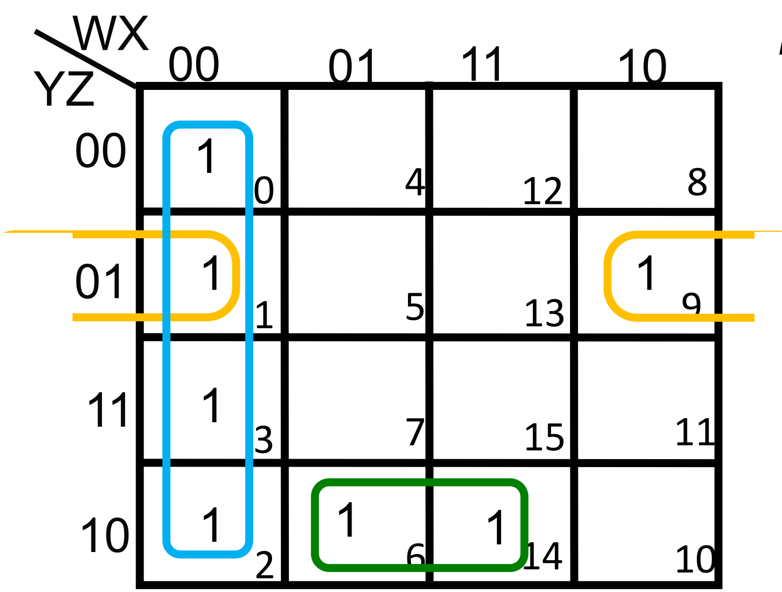
YZ \ WX	00	01	11	10
00	0 $_{0}$	$_{4}$	$_{12}$	$_{8}$
01	0 $_{1}$	$_{5}$	$_{13}$	0 $_{9}$
11	0 $_{3}$	$_{7}$	$_{15}$	$_{11}$
10	0 $_{2}$	0 $_{6}$	0 $_{14}$	$_{10}$

K-Map: Simplification: Samples 17

$F(W,X,Y,Z) = \prod M(0,1,2,3,6,9,14)$

$F'(A,B,C,D,E) = (W'X') + (X\,Y\,Z') + (X'\,Y'\,Z)$

$\bar{F}$



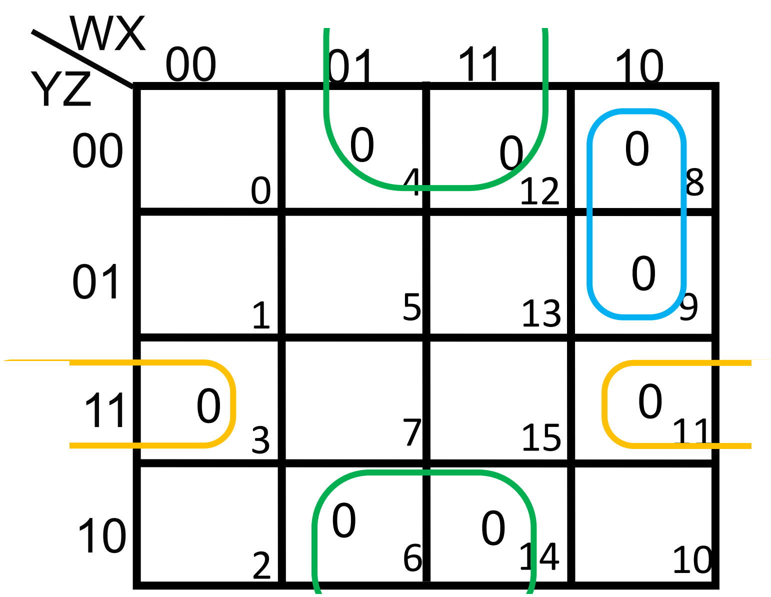
$F(A,B,C,D,E) = (W+X)\quad(X'+Y'+Z)\quad(X+Y+Z')$

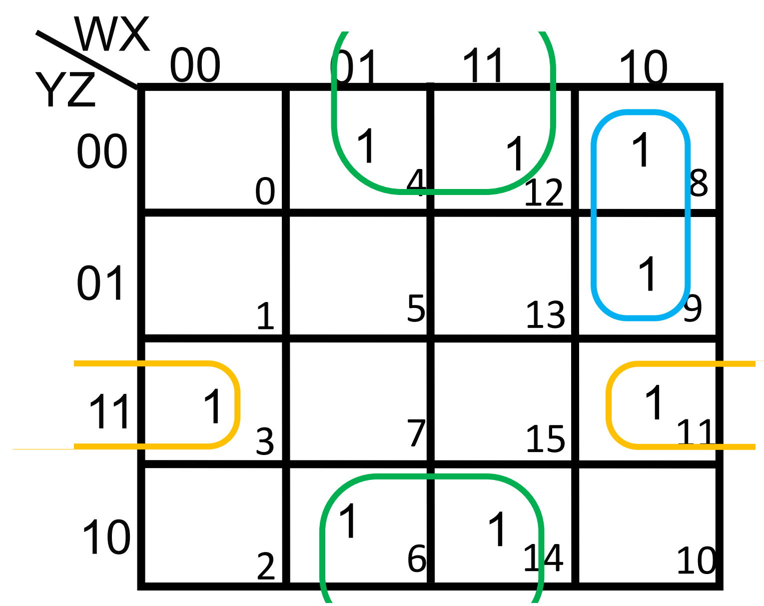
K-Map: Simplification: Samples 17

$F(W,X,Y,Z) = \prod M(3,4,6,8,9,11,12,14)$ and its complement

$F(A,B,C,D,E) = (W'+X+Y)\ (X' + Z)\ (X + Y' + Z')$

$F'(A,B,C,D,E) = (WX'Y') + (X Z') + (X' Y Z)$





K-Map: Simplification: Samples 18

$F(a,b,c,d,e) = \prod M(0,2,4,11,14,15,16,20,24,30,31)$

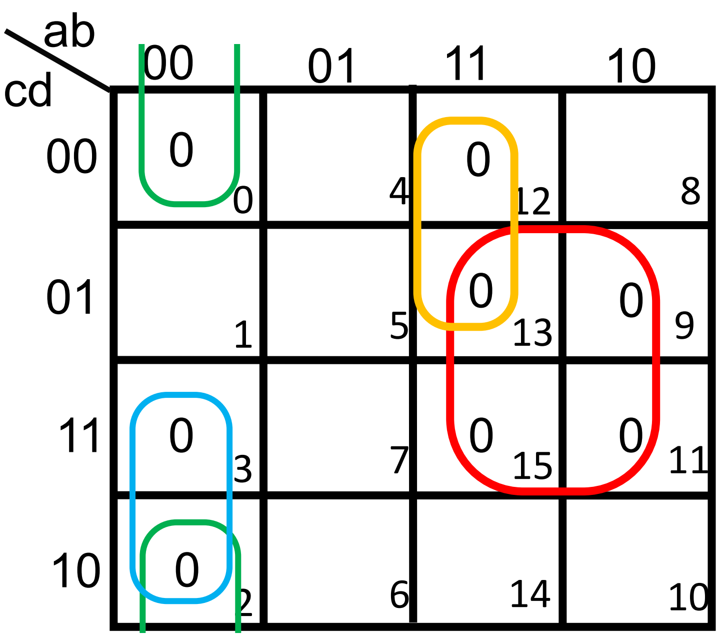
de \ abc	000	001	011	010
00	0 $_{0}$	0 $_{4}$	$_{12}$	$_{8}$
01	$_{1}$	$_{5}$	$_{13}$	$_{9}$
11	$_{3}$	$_{7}$	0 $_{15}$	0 $_{11}$
10	0 $_{2}$	$_{6}$	0 $_{14}$	$_{10}$

	100	101	111	110
00	0 $_{16}$	0 $_{20}$	$_{28}$	0 $_{24}$
01	$_{17}$	$_{21}$	$_{29}$	$_{25}$
11	$_{19}$	$_{23}$	0 $_{31}$	$_{27}$
10	$_{18}$	$_{22}$	0 $_{30}$	$_{26}$

F(a,b,c,d,e) = (b' + c' + d')(b + d + e) (a+ b + c + e) (a'+ c + d + e) (a+ b' + d' + e')

K-Map: Simplification: Samples 19

$F(a,b,c,d,e) = \prod M(0,2,3,9,11,12,13,15)$

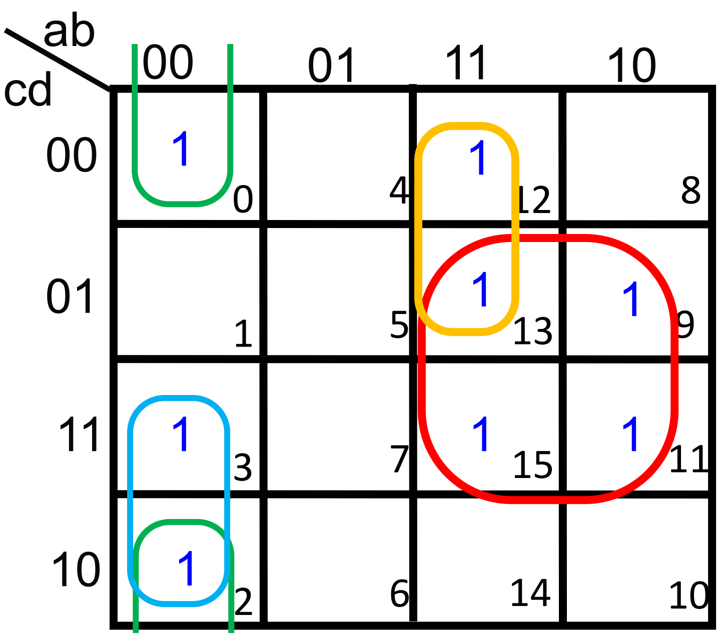


F(a,b,c,d,e) = (a' + d') (a+ b + d) (a+ b + c')

(a'+ b' + c)

K-Map: Simplification: Samples 19 (cont'd)

$F(a,b,c,d,e) = \prod M(0,2,3,9,11,12,13,15)$



$F'(a,b,c,d) = \sum m(0,2,3,9,11,12,13,15)$

$F'(a,b,c,d) = a\,d + a'b'd' + a'b'c + abc'$

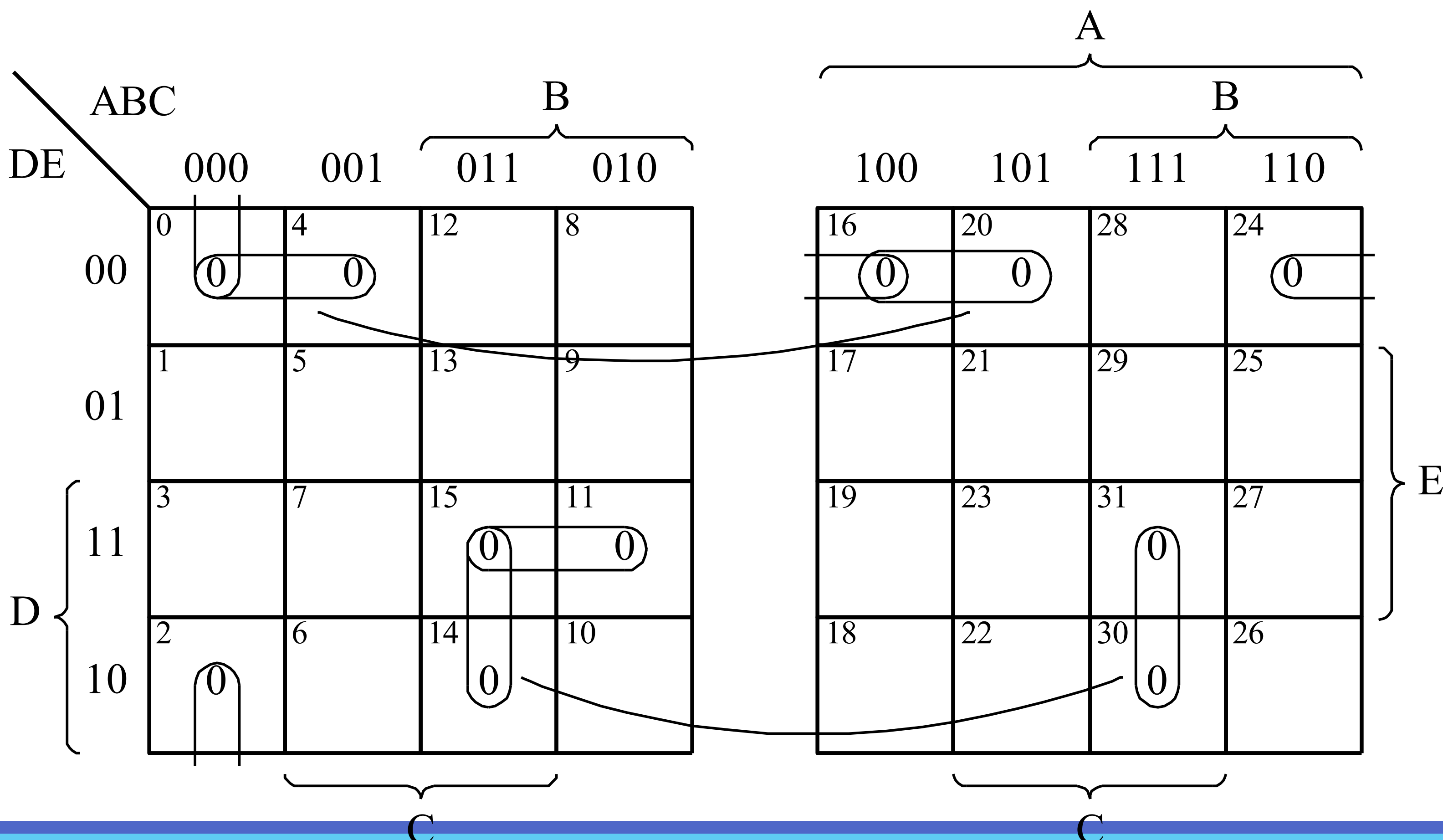
$F = (F')' = (a\,d + a'b'd' + a'b'c + abc')'$

$= (\bar{a} + \bar{d}).(a + b + d).(a + b + c')$

$(a' + b' + c)$

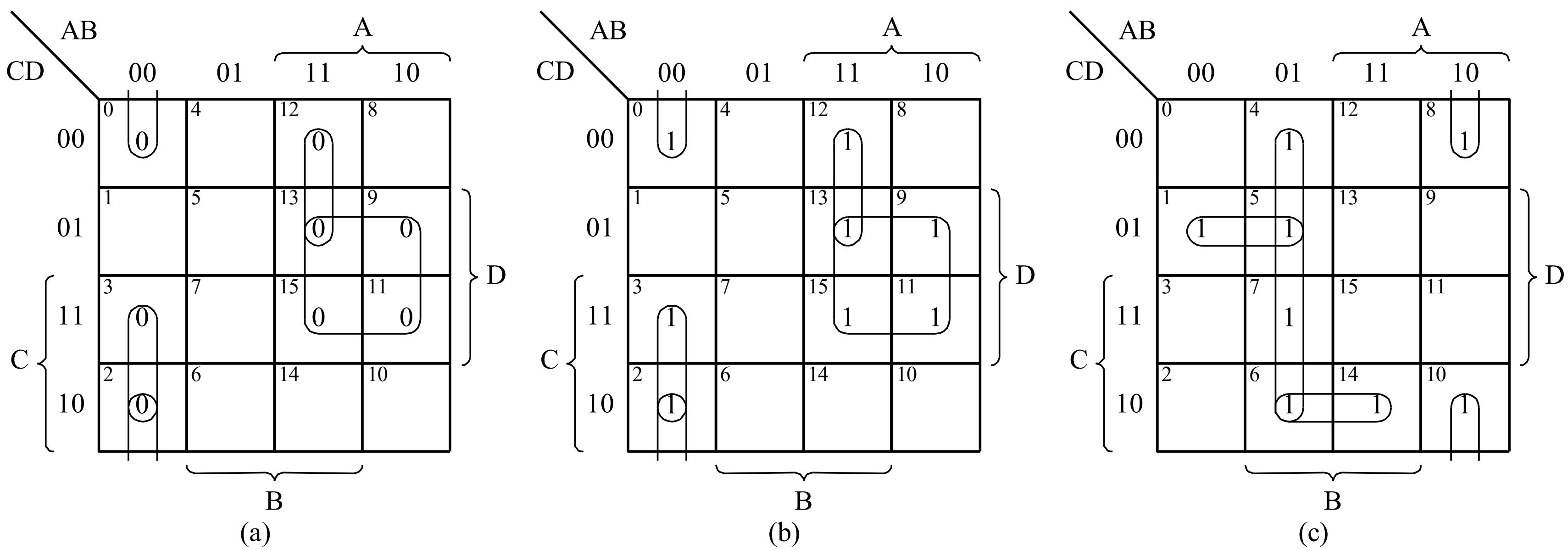
K-Map: Simplification: Samples 20

- Find a minimal POS expression for a 5-variable function



K-Map: Simplification: Samples 21

- Deriving POS and SOP forms of a function.



K-Map: Simplification: Don’t Cares

- Entries that a function table or K-map is known for
  - Input values for the minterm will never occur, or
  - Output value for the minterm is not used
- Output value need not be defined
- Output value is defined as a “don't care”

K-Map: Simplification: Don’t Cares (cont’d)

- Place “don't cares” (“d”) in the function table or map, the cost of the logic circuit may be lowered.

- Sample:
  - A logic function having the binary codes for the BCD digits as its inputs.
  - Used codes are 0-9.
  - Six codes, 1010 through 1111 never occur
  - => Output values for these codes are “d = don’t cares.”

K-Map: Simplification: Samples 22

$f(a, b, c, d) = \sum m(1, 3, 4, 7, 11) + d(5,12,13,14,15)$

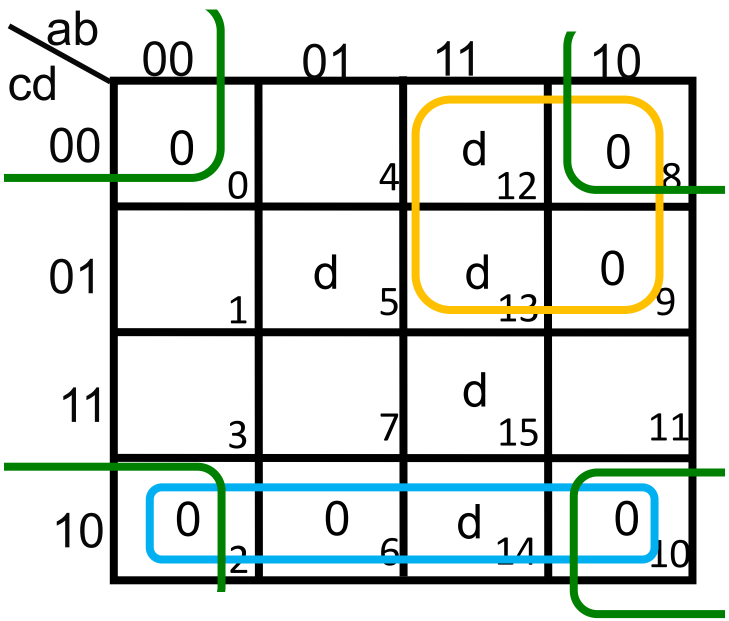
$f(a, b, c, d) = (cd) + (a' + d) + (b\,c')$

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

K-Map: Simplification: Samples 23

f(a, b, c, d) =$\prod M$(0, 2, 6, 8, 9, 10)+ d(5,12,13,14,15)

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

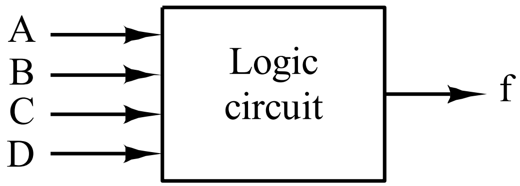


K-Map: Simplification: Samples 24

- Design a circuit to distinguish BCD digits ≥ 5 from those < 5

K-Map: Simplification: Samples 24 (cont'd)

- Design a circuit to distinguish BCD digits ≥ 5 from those < 5

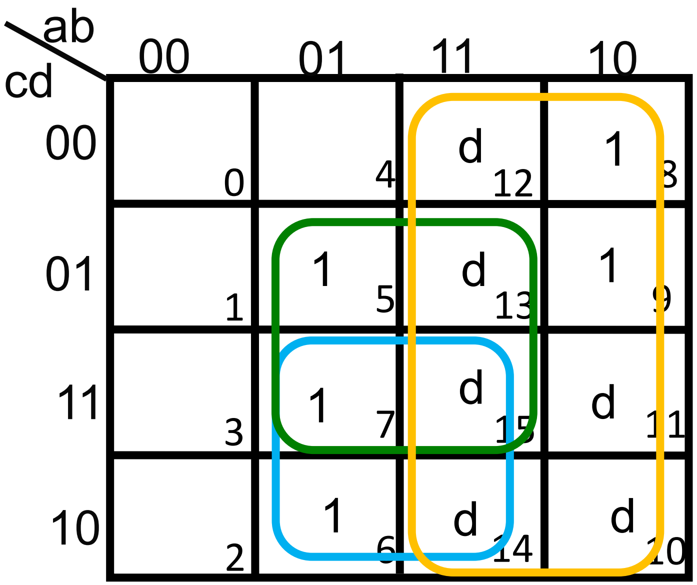


(a)

ABCD	Minterm	f(A, B, C, D)
0000	0	0
0001	1	0
0010	2	0
0011	3	0
0100	4	0
0101	5	1
0110	6	1
0111	7	1
1000	8	1
1001	9	1
1010	10	d
1011	11	d
1100	12	d
1101	13	d
1110	14	d
1111	15	d

(b)

K-Map: Simplification: Samples 24 (cont'd)



$f(A, B, c, D) = (B\ C) + (B\ D) + (A\ )$

ABCD	Minterm	f(A, B, C, D)
0000	0	0
0001	1	0
0010	2	0
0011	3	0
0100	4	0
0101	5	1
0110	6	1
0111	7	1
1000	8	1
1001	9	1
1010	10	d
1011	11	d
1100	12	d
1101	13	d
1110	14	d
1111	15	d

K-Map: Simplification: Samples 24 (cont’d)

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

$f(A, B, C, D) = (A + B)\ (A + B + C)$

ABCD	Minterm	f(A, B, C, D)
0000	0	0
0001	1	0
0010	2	0
0011	3	0
0100	4	0
0101	5	1
0110	6	1
0111	7	1
1000	8	1
1001	9	1
1010	10	d
1011	11	d
1100	12	d
1101	13	d
1110	14	d
1111	15	d

Thank You