In The Name Of Allah

Chapter 3
Simplification of Switching Functions

Department of Electrical Engineering
Islamic Azad University
Qazvin Branch

3.1 Simplification Goals

- 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)

Example 3.1

- Determine the form and the number of terms and literals in each of the following.

- $g(A,B,C) = AB' + A'B + AC$
  - Two-level form, three products , two sums, six literals

- $f(X,Y,Z) = X'Y(Z + Y'X) + Y'Z$
  - Four-level form, four products, two sums, seven literals

3.2 Minimization 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)

Minimum SOP and POS Representations

- The *minimum sum of products (MSOP)* of a function, *f*, is a SOP representation of *f* that contains the fewest number of product terms and fewest number of literals of any SOP representation of *f*.
- *Example -- f(a,b,c,d)* $= \sum m(3,7,11,12,13,14,15) = ab + a'cd + acd$
  $= ab + cd$

- The *minimum product of sums (MPOS)* of a function, *f*, is a POS representation of *f* that contains the fewest number of sum terms and the fewest number of literals of any POS representation of *f*.
- *Example -- f(a,b,c,d)* $= \prod M(0,1,2,4,5,6,8,9,10)$
  $= (a + c)(a + d)(a' + b + d)(b + c' + d)$
  $= (a + c)(a + d)(b + c)(b + d)$

3.3 Karnaugh Maps

- Karnaugh maps (K-maps) -- convenient tool for representing switching functions of up to six variables.
- K-maps form the basis of useful heuristics for finding MSOP and MPOS representations.
- An *n*-variable K-map has 2^n cells with each cell corresponding to a row of an *n*-variable truth table.
- K-map cells are labeled with the corresponding truth-table row.
- K-map cells are arranged such that 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 map.

Figure 3.1 Venn diagram and equivalent K-map for two variables

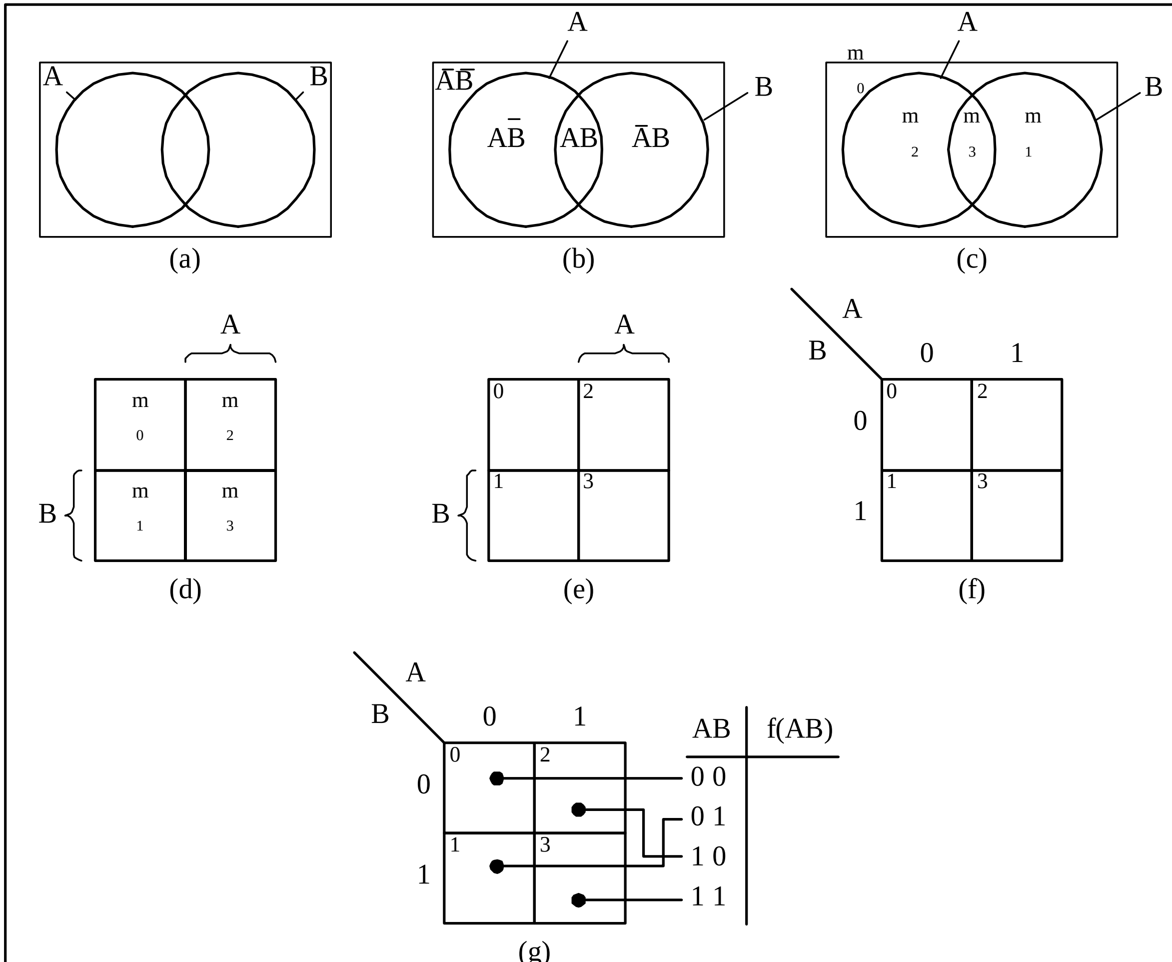


Figure 3.2 Venn diagram and equivalent K-map for three variables

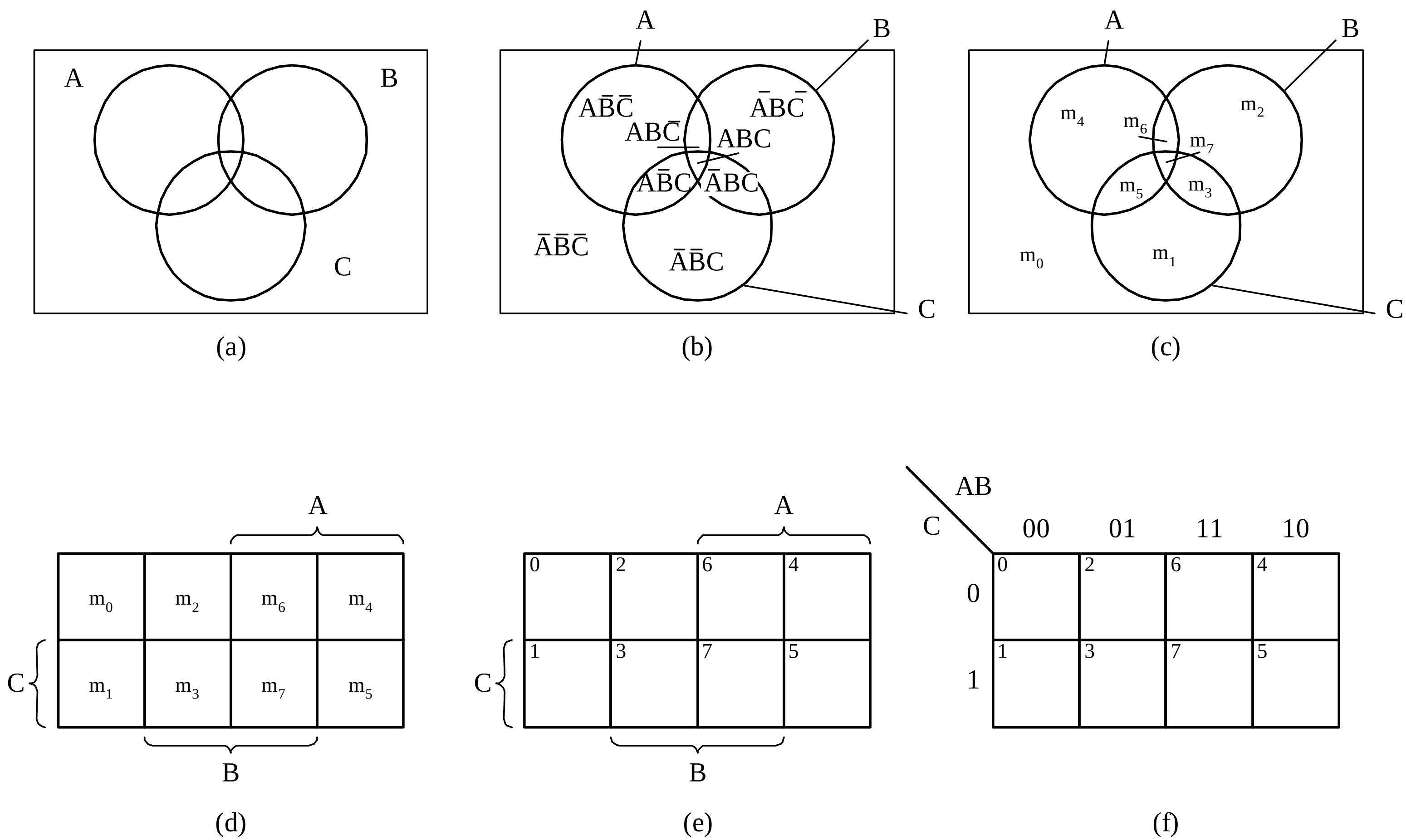


Figure 3.3 (a) -- (d) K-maps for four and five variables

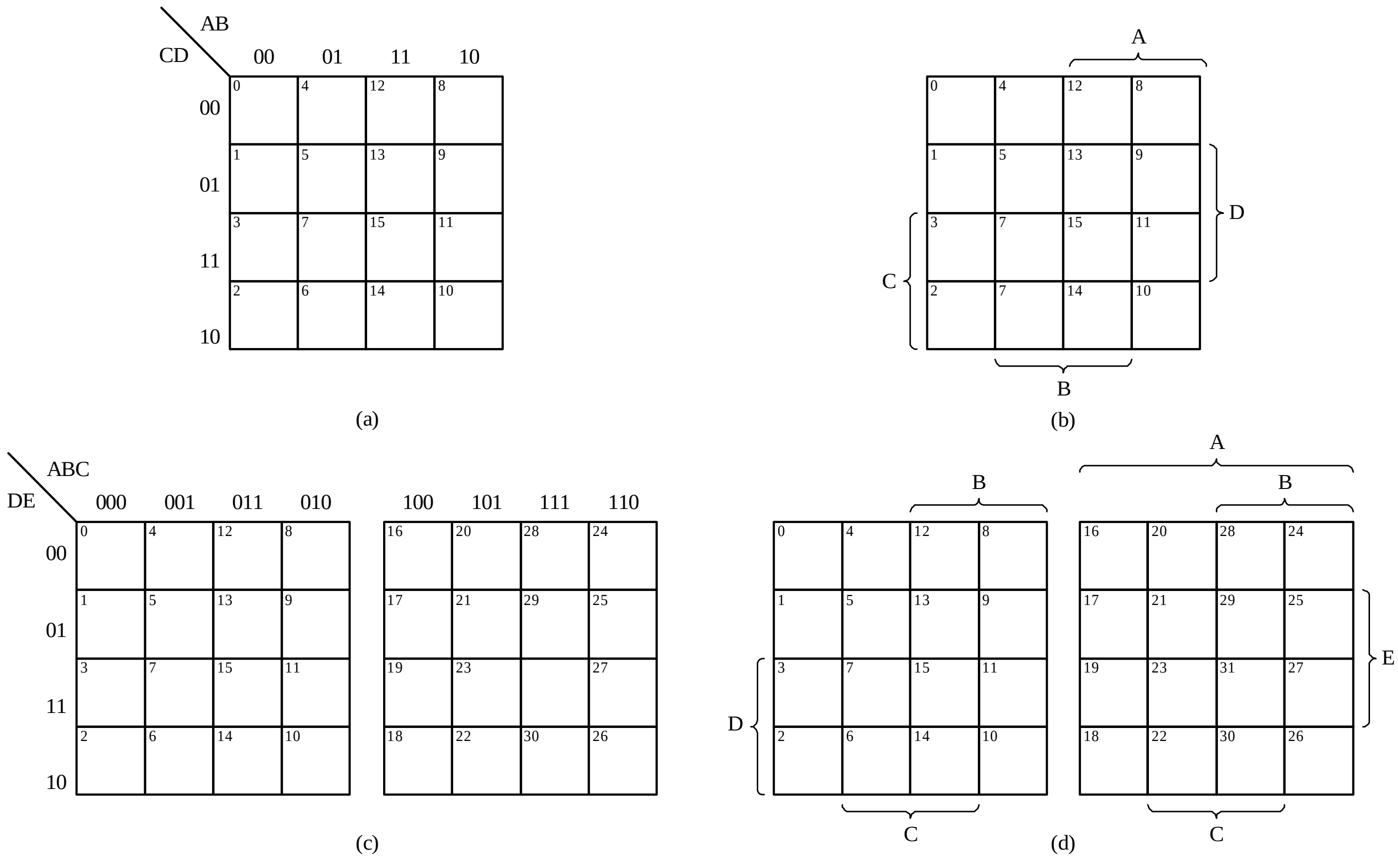
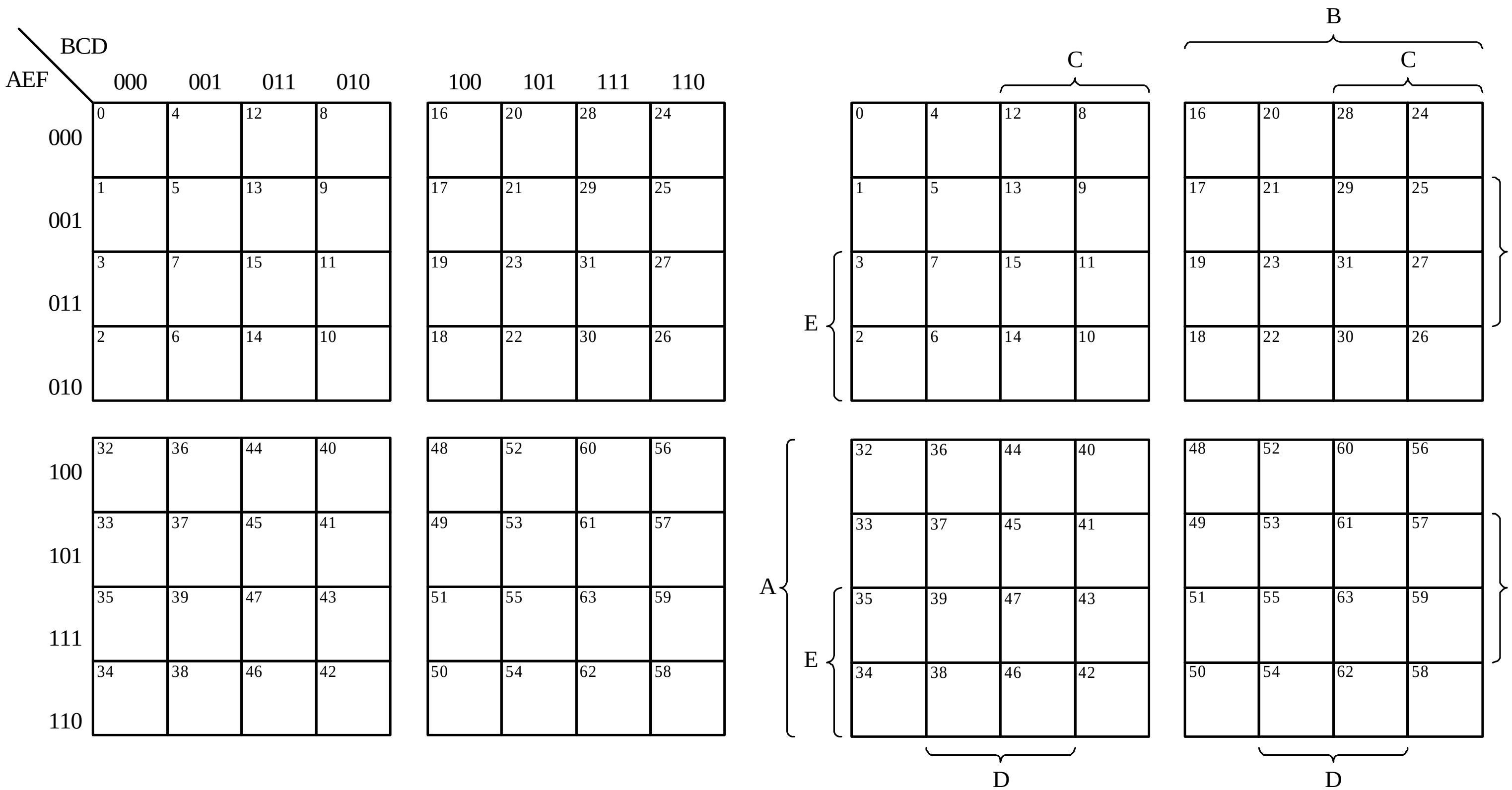


Figure 3.3 (e) -- (f) K-maps for six variables



(e) (f)

3.4 Plotting (Mapping) Functions in Canonical Form on a K-map

- Let f be a switching function of n variables where $n \leq 6$.
- Assume that the cells of the K-map are numbered from 0 to $2^n - 1$ where the 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.

Figure 3.4 Plotting functions on K-maps

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

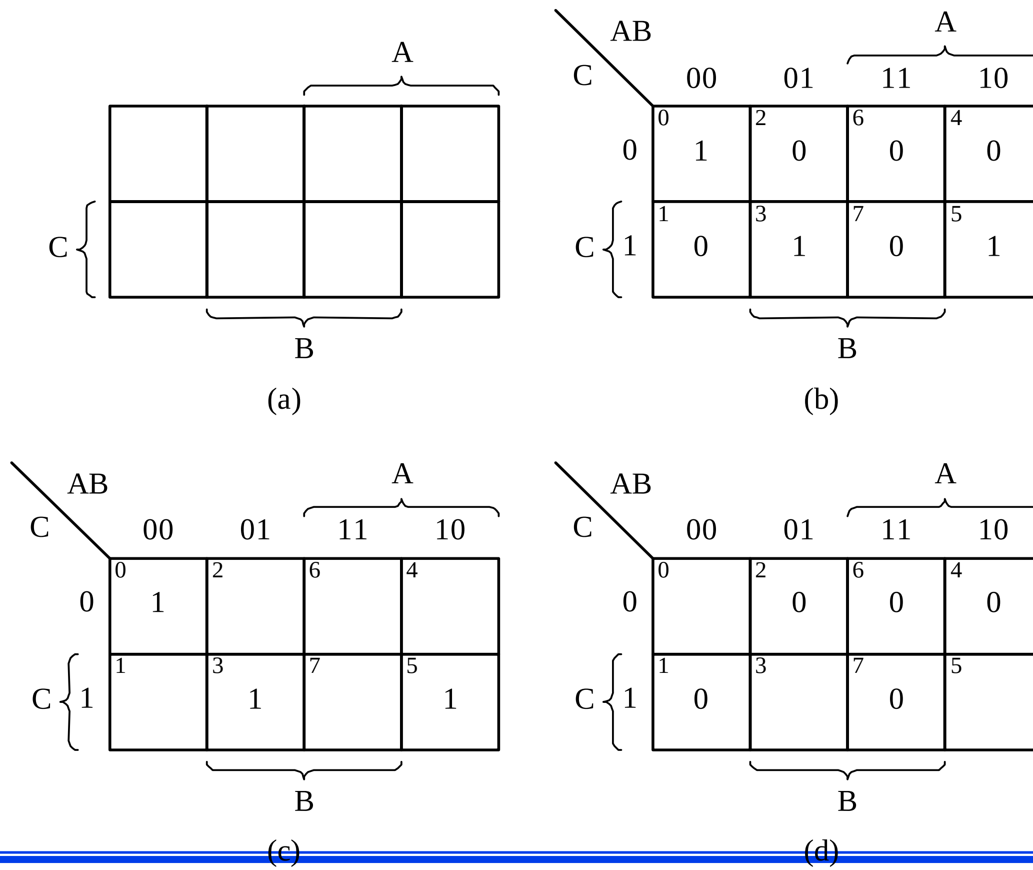


Figure 3.5 K-maps for *f(a,b,Q,G)* in Example 3.4
(a) Minterm form. (b) Maxterm form.

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

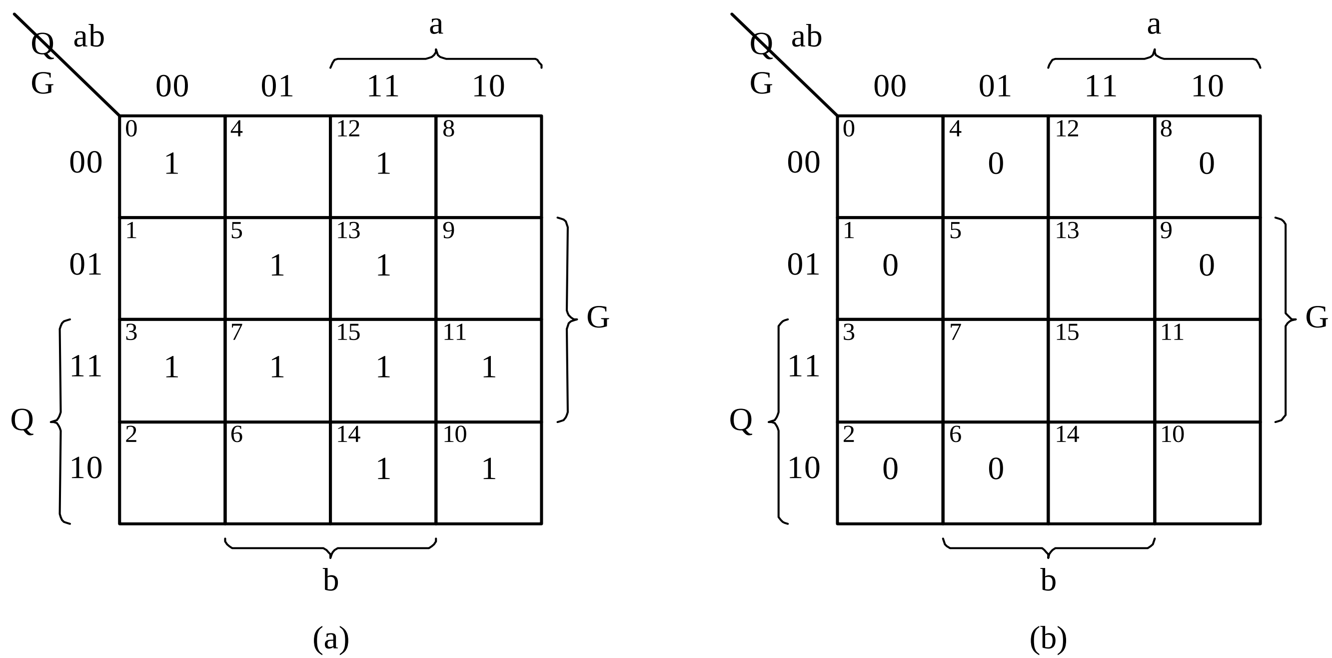


Figure 3.6 K-map of Figure 3.5(a) with variables reordered: *f(Q,G,b,a).*

$$f(Q,G,b,a) = \sum m(0,12,6,14,9,13,3,7,11,15) = \sum m(0,3,6,7,9,11,12,13,14,15)$$

ba \ QG	00	01	11	10
00	0: 1	4:	12: 1	8:
01	1:	5:	13: 1	9: 1
11	3: 1	7: 1	15: 1	11: 1
10	2:	6: 1	14: 1	10:

Plotting Functions in Algebraic Form

- Example 3.6 -- $f(A,B,C) = AB + BC'$

- Example 3.7 -- $f(A,B,C,D) = (A + C)(B + C)(B' + C' + D)$

- Example 3.8 -- $f(A,B,C,D)= (A'+B')(A'+C+D')(B'+C'+D')$

Figure 3.7 -- Example 3.6.
(a) Venn diagram form. (b) Sum of minterms. (c) Maxterms.

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

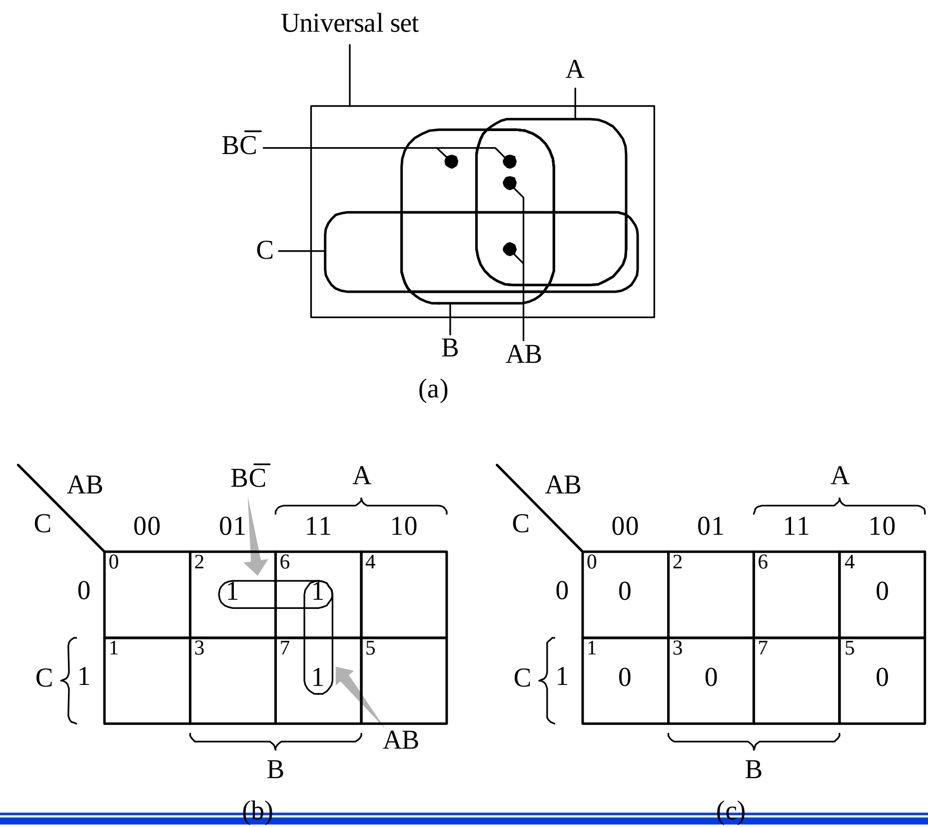
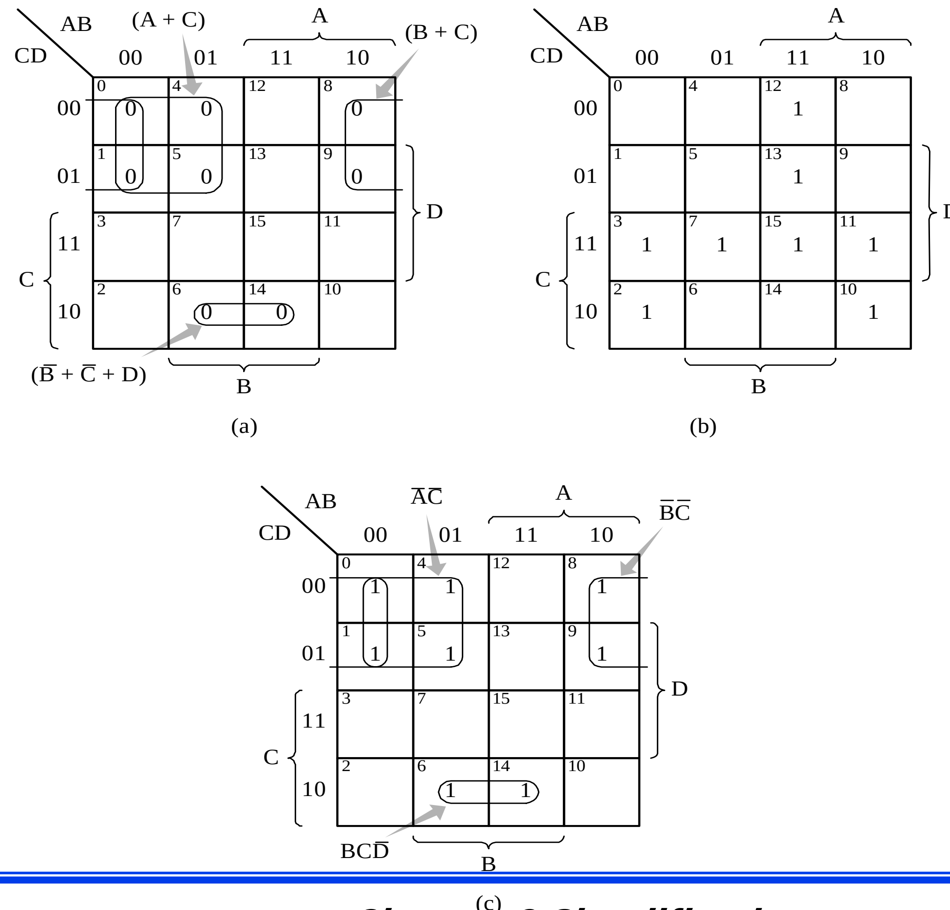


Figure 3.8 -- Example 3.7.
(a) Maxterms, (b) Minterms, (c) Minterms of *f ´*.

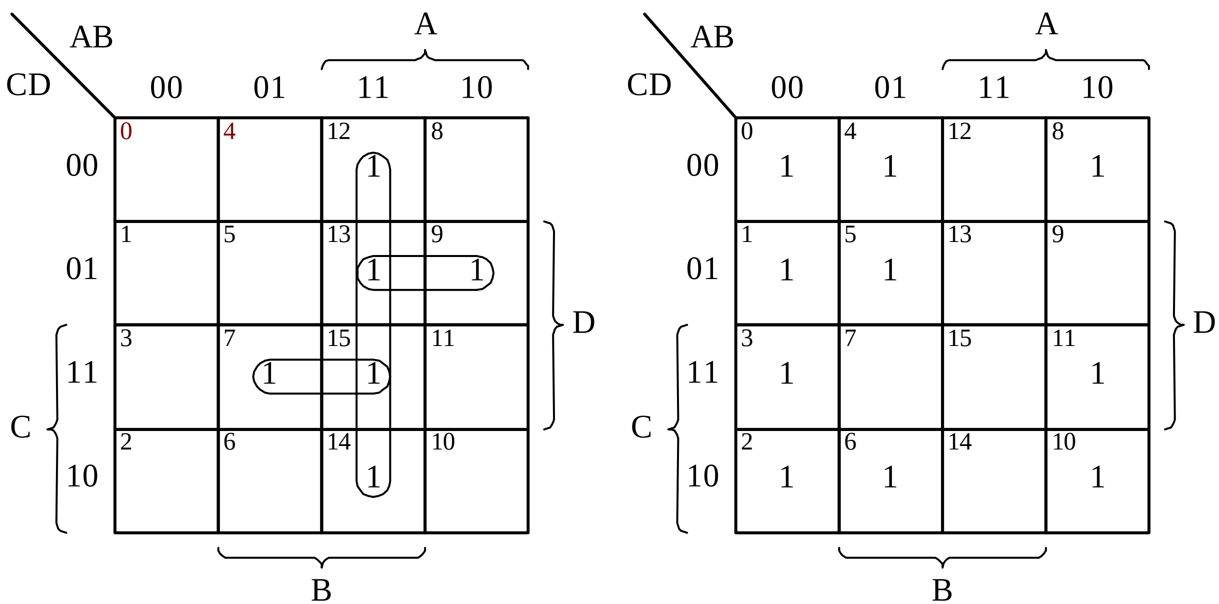
$$f(A,B,C,D) = (A + C)(B + C)(B' + C' + D)$$



QIAU

Figure 3.9 -- Example 3.8.
(a) K-map of f', (b) K-map of f.

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



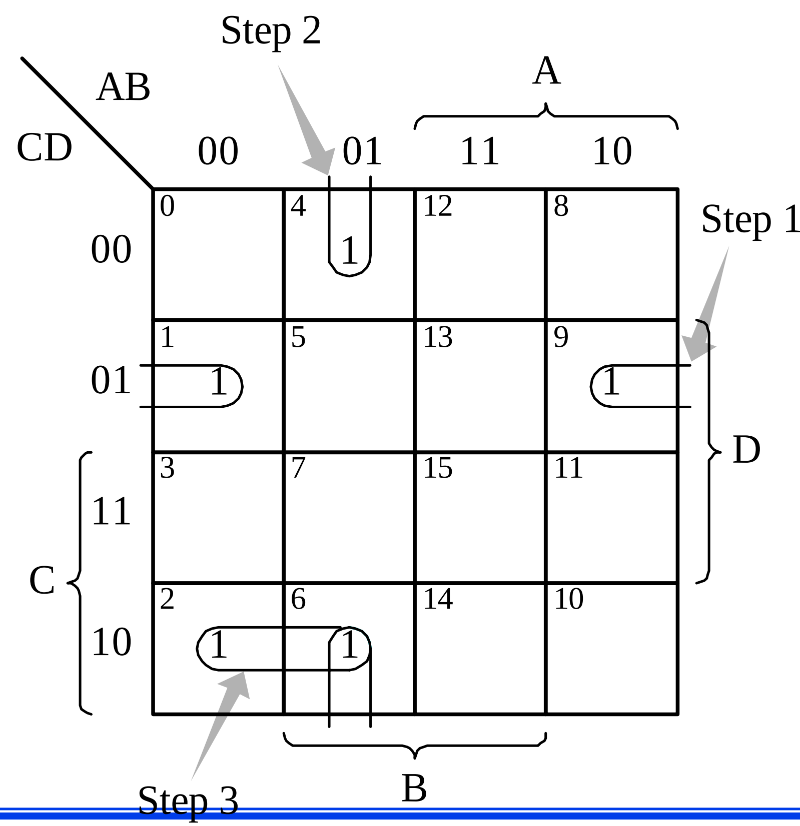
(a) (b)

3.5 Simplification of Switching Functions using K-maps

- K-map cells that are physically adjacent are also logically adjacent. Also, cells on an edge of a K-map are logically adjacent to cells on the opposite edge of the map.
- If two logically adjacent cells both contain logical 1s, the two cells can be combined to eliminate the variable that has value 1 in one cell's label and value 0 in the other.
- This is equivalent to the algebraic operation, $aP + a'P = P$ where P is a product term not containing *a* or *a*′.
- *Example -- f(A,B,C,D) =* $\sum m(1,2,4,6,9)$

Figure 3.10 K-map for Example 3.9

$$f(A,B,C,D) = \sum m(1,2,4,6,9)$$



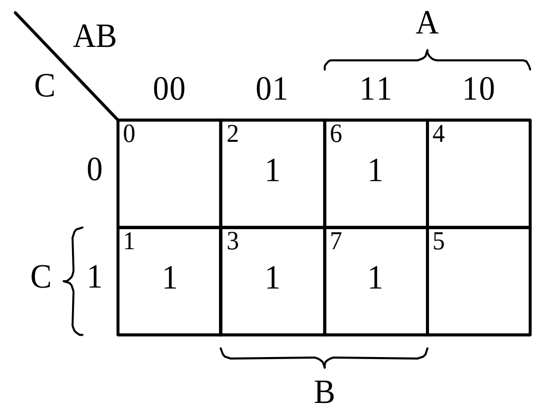
Simplification Guidelines for K-maps

- Each cell of an *n*-variable K-map has *n* logically adjacent cells.
- Cells may be combined in groups of 2,4,8,...,2^k.
- A group of cells can be combined only if all cells in the group have the same value for some set of variables.
- Always combine as many cells in a group as possible. This will result in the fewest number of literals in the term that represents the group.
- Make as few groupings as possible to cover all minterms. This will result in the fewest product terms.

Prime Implicants and Covers

- An *implicant* is a product term that can cover minterms of a function.
- A *prime implicant* is a product term that is not covered by another implicant of the function.
- An *essential prime implicant* is a prime implicant that covers at least one minterm that is not covered by any other prime implicant.
- A set of implicants is said to be a *cover of a function* if each minterm of the function is covered by at least one implicant in the set.
- A *minimal cover* is a cover that contains the smallest number of prime implicants and the smallest number of literals..

Figure 3.11 K-map illustrating implicants



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 = $A'C + B$

Algorithm 3.1 -- Generating and Selecting Prime Implicants

1. Count the number of adjacencies for each minterm on the K-map.

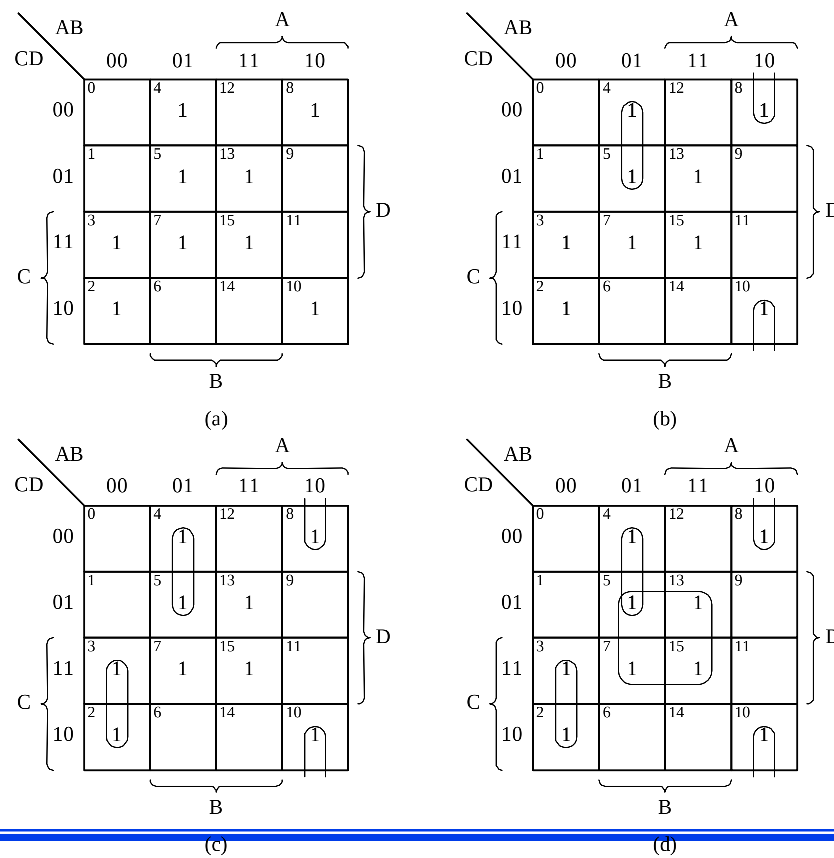
2. Select an uncovered minterm with the fewest number of adjacencies. Make an arbitrary choice if more than one choice is possible.

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.

Figure 3.12 -- Example 3.10
(Illustrating Algorithm 3.1)

$$f(A,B,C,D) = \sum m(2,3,4,5,7,8,10,13,15)$$



Algorithm 3.2 -- Generating and Selecting Prime Implicants (Revisited)

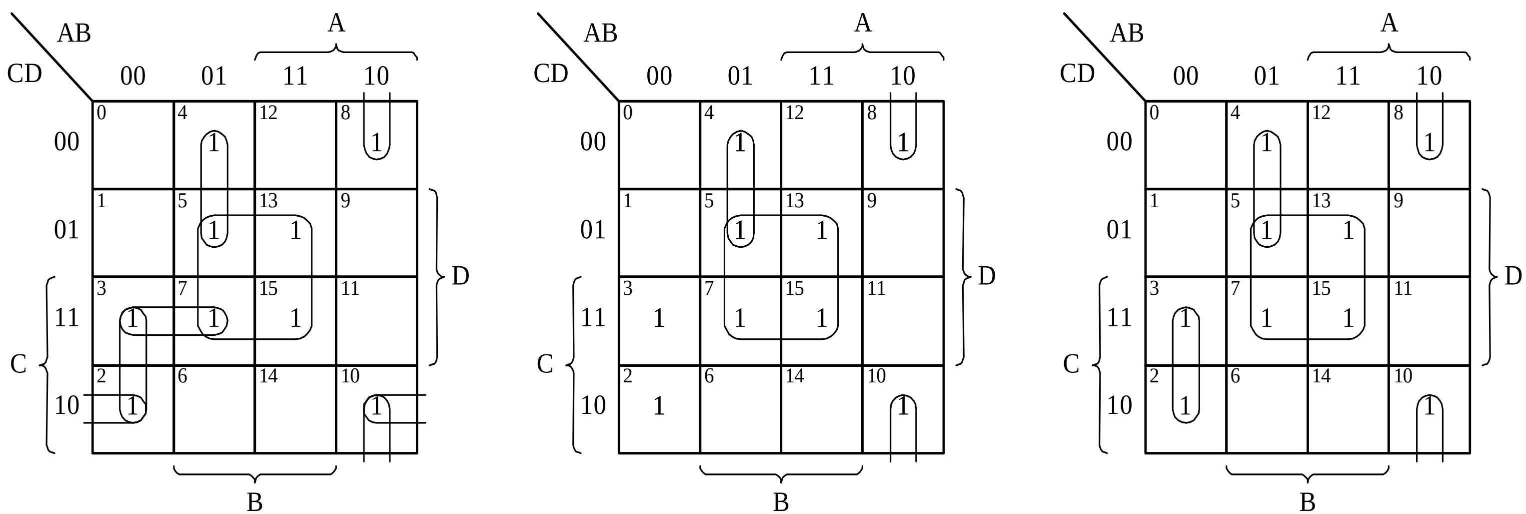
1. Circle all prime implicants on the K-map.

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 minterms not covered by the essential prime implicants.

Figure 3.13 -- Example 3.11
(Illustrates Algorithm 3.2)

$$f(A,B,C,D) = \sum m(2,3,4,5,7,8,10,13,15)$$



(a) (b) (c)

Figure 3.14 -- Example 3.12

f(A,B,C,D) = $\sum m(0,5,7,8,10,12,14,15)$

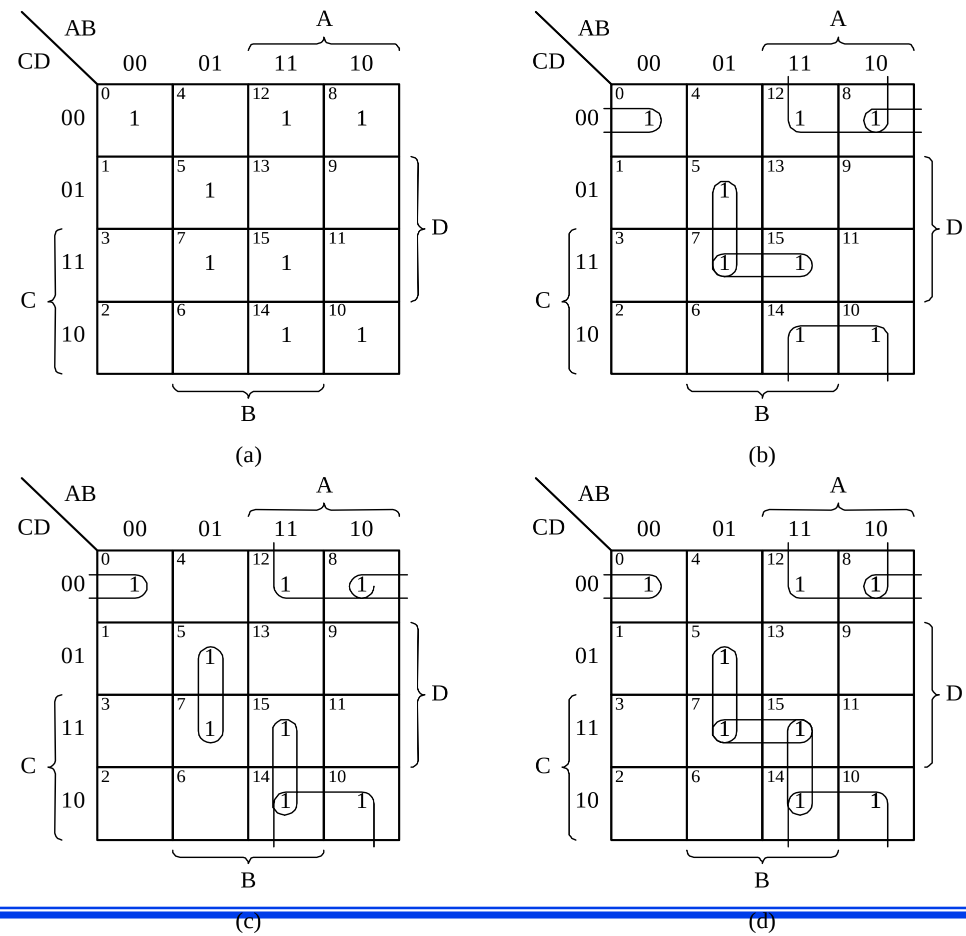


Figure 3.15 -- Example 3.13

$f(A,B,C,D) = \sum m(1,2,3,6) = A'C + BC'$

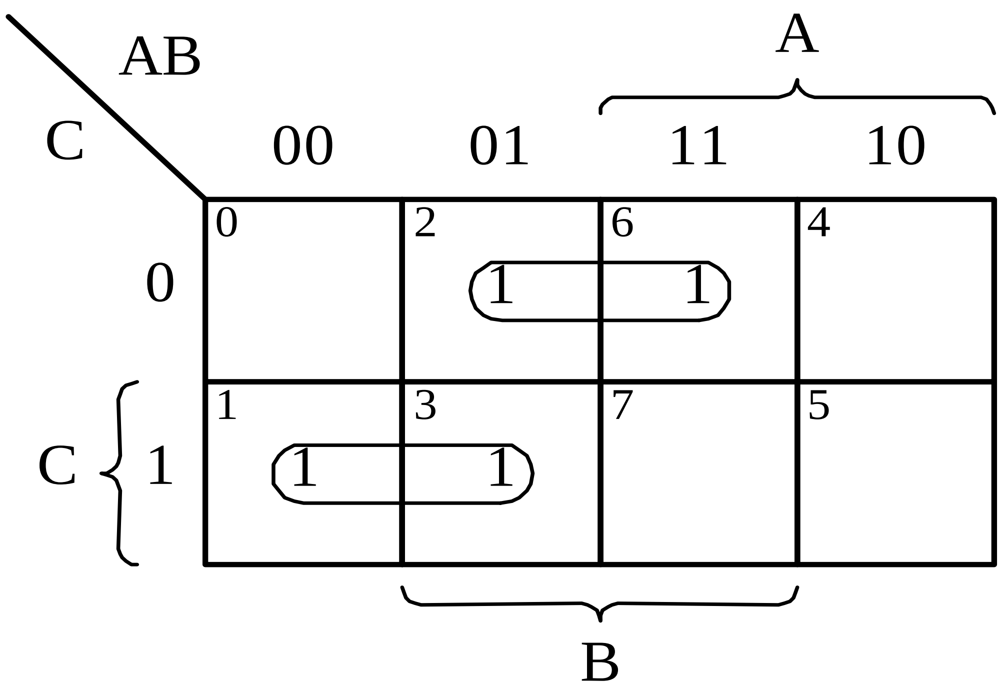


Figure 3.16 -- Example 3.14

f(A,B,C,D) = B′D′ + B′C′ + BCD

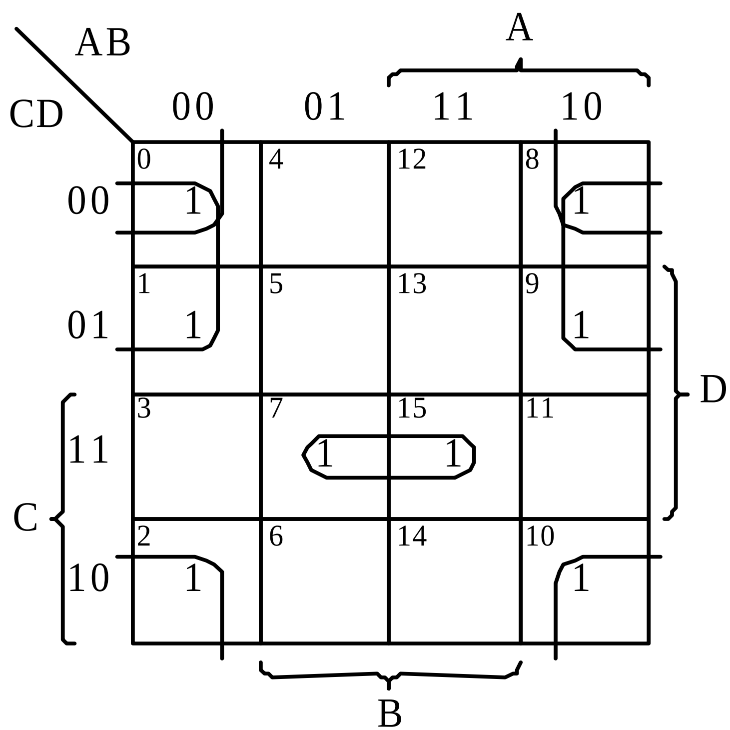
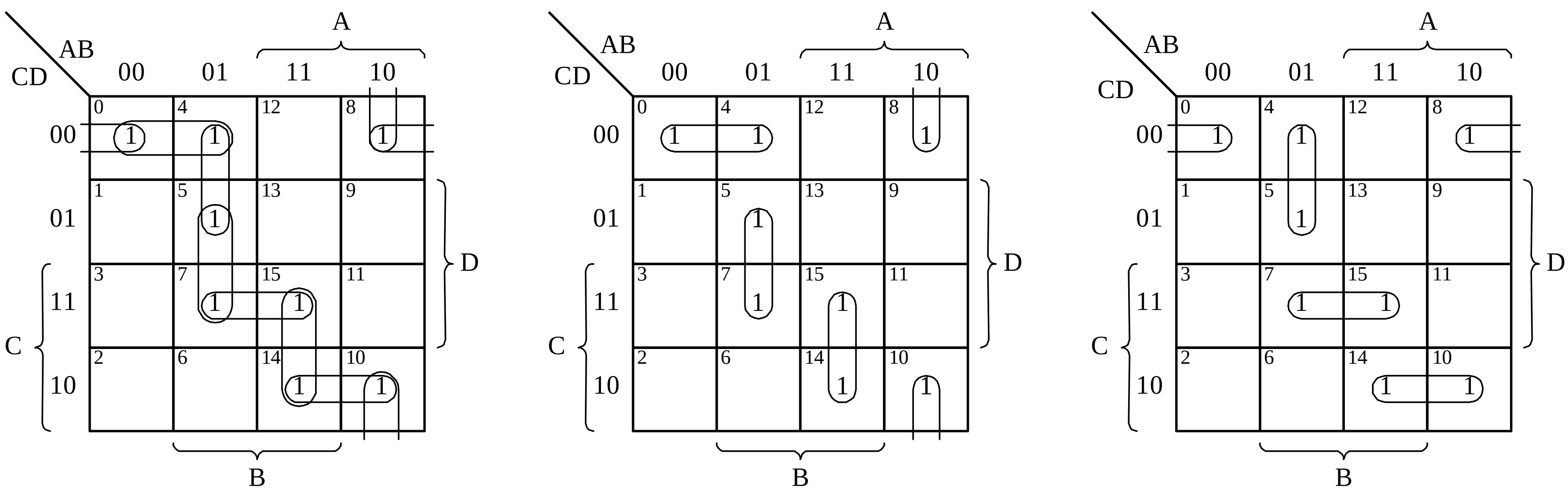


Figure 3.17 -- Example 3.15
Function with no essential prime implicants.

$$f(A,B,C,D) = \sum m(0,4,5,7,8,10,14,15)$$

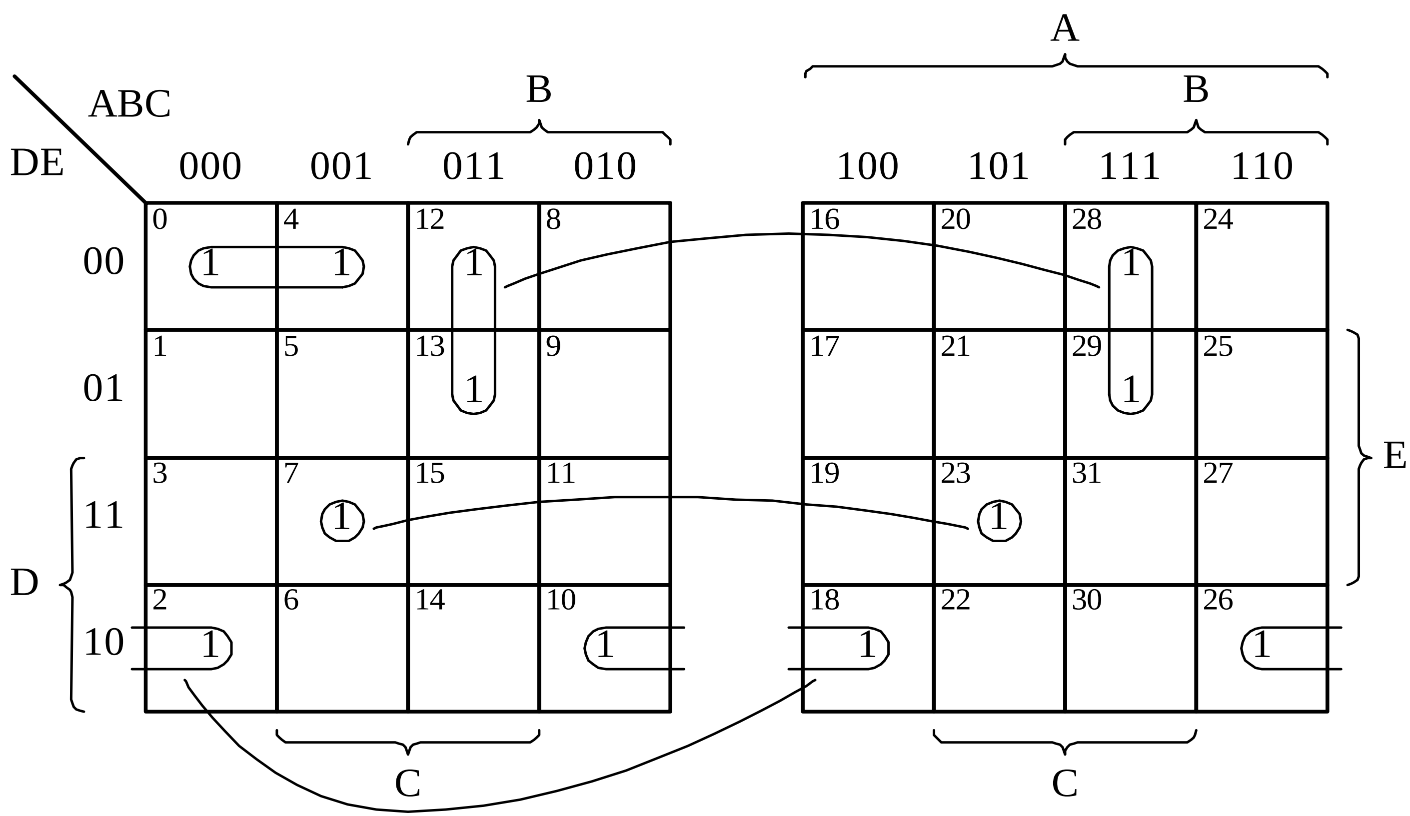


(a) (b) (c)

Figure 3.18 -- Example 3.16

Minimizing a five-variable function.

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



(a) (b)

Prime Implicates and Covers

- A *implicate* is a sum term that can cover maxterms of a function.
- A *prime implicate* is a sum term that is not covered by another *implicate* of the function.
- An *essential prime implicate* is a prime *implicate* that covers at least one maxterm that is not covered by any other prime *implicate*.
- A set of *implicate* is said to be a *cover of a function* if each maxterm of the function is covered by at least one *implicate* in the set.
- A *minimal cover* is a cover that contains the smallest number of prime *implicate* and the smallest number of literals..

Algorithm 3.3 -- Generating and Selecting Prime Implicates

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 implicate for this maxterm and put it in the cover. If this maxterm is covered by more than one prime implicate, select the one that covers the most uncovered maxterms.

4. Repeat steps 2 and 3 until all maxterms have been covered.

Algorithm 3.4 -- Generating and Selecting Prime Implicates (Revisited)

1. Circle all prime implicates on the K-map.

2. Identify and select all essential prime implicates for the cover.

3. Select a minimum subset of the remaining prime implicates to complete the cover, that is, to cover those maxterms not covered by the essential prime implicates.

Example 3.17 -- Find the minimum POS form of the function
$f(A,B,C,D) = \prod M(0,1,2,3,6,9,14)$

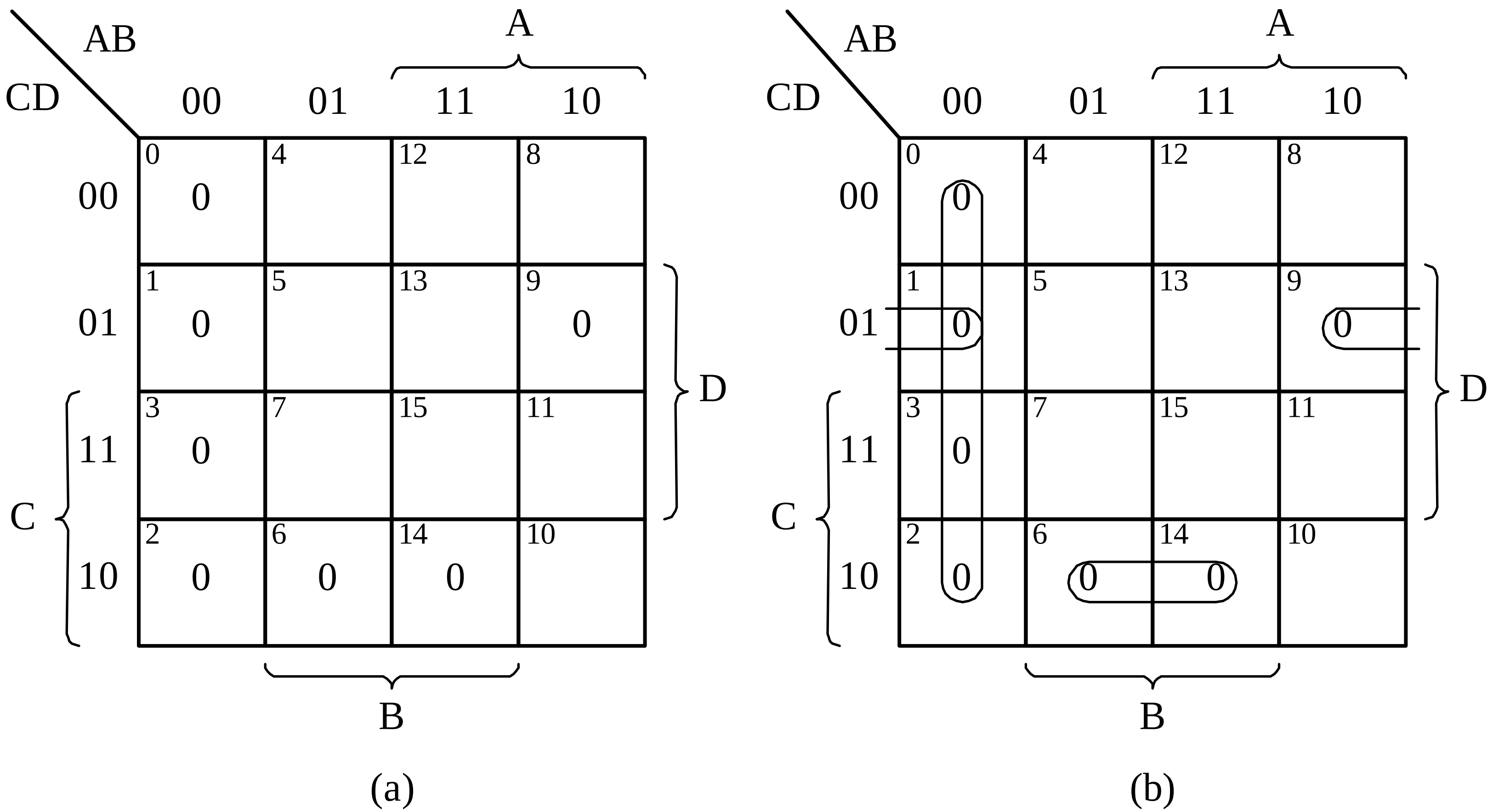


Figure 3.19 K-maps for Example 3.17.

Algorithm 3.5 -- Finding MPOS of *f* from *f′*

1. Plot the complement function *f′* on the K-map.

2. Use algorithm 3.1 or 3.2 to produce a MSOP of *f′*.

3. Complement *f′* and use DeMorgan's theorem to produce a MSOP of *f*.

Example 3.18 -- Find the MPOS of the following function using Algorithm 3.5

$f(A,B,C,D) = \prod M(0,1,2,3,6,9,14)$

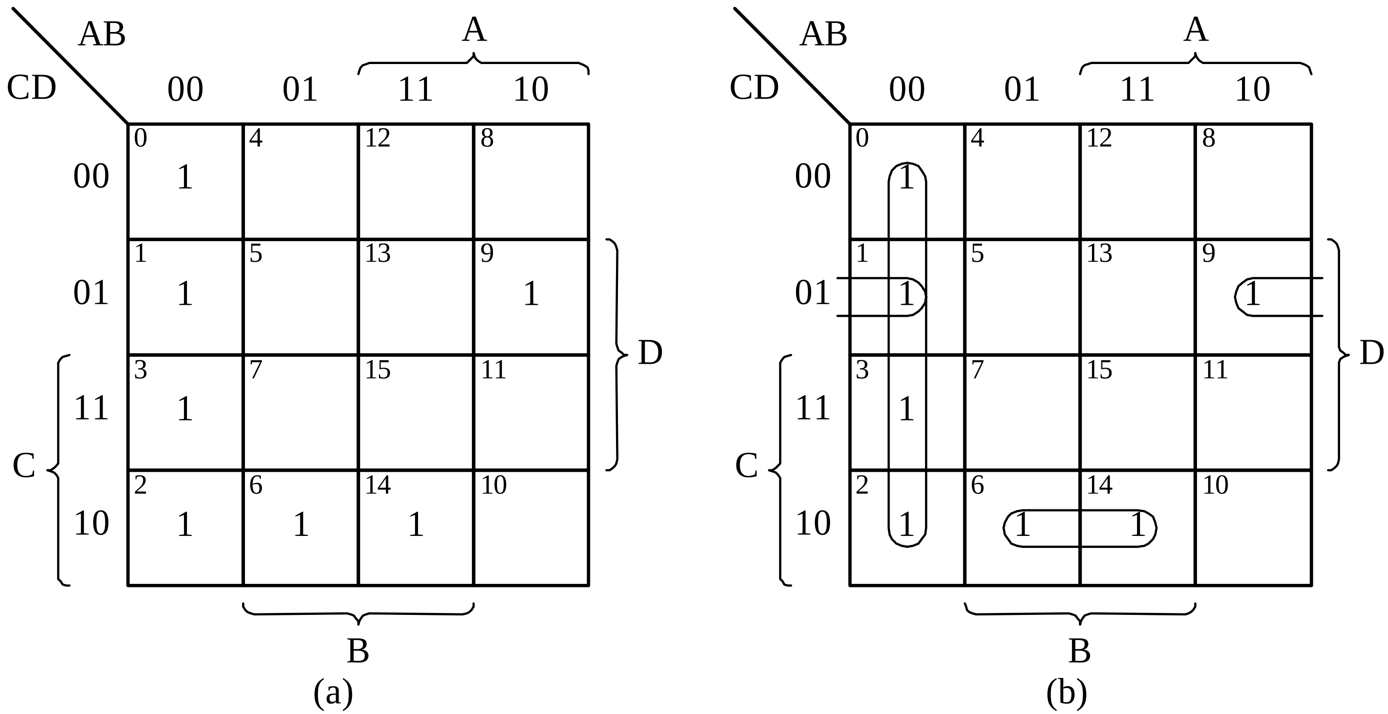


Figure 3.20 K-map of f'

$$f' = A'B' + B'C'D + BCD'$$

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

Example 3.19 -- Minimum covers of $f(A,B,C,D) = \prod M(3,4,6,8,9,11,12,14)$ and its complement.

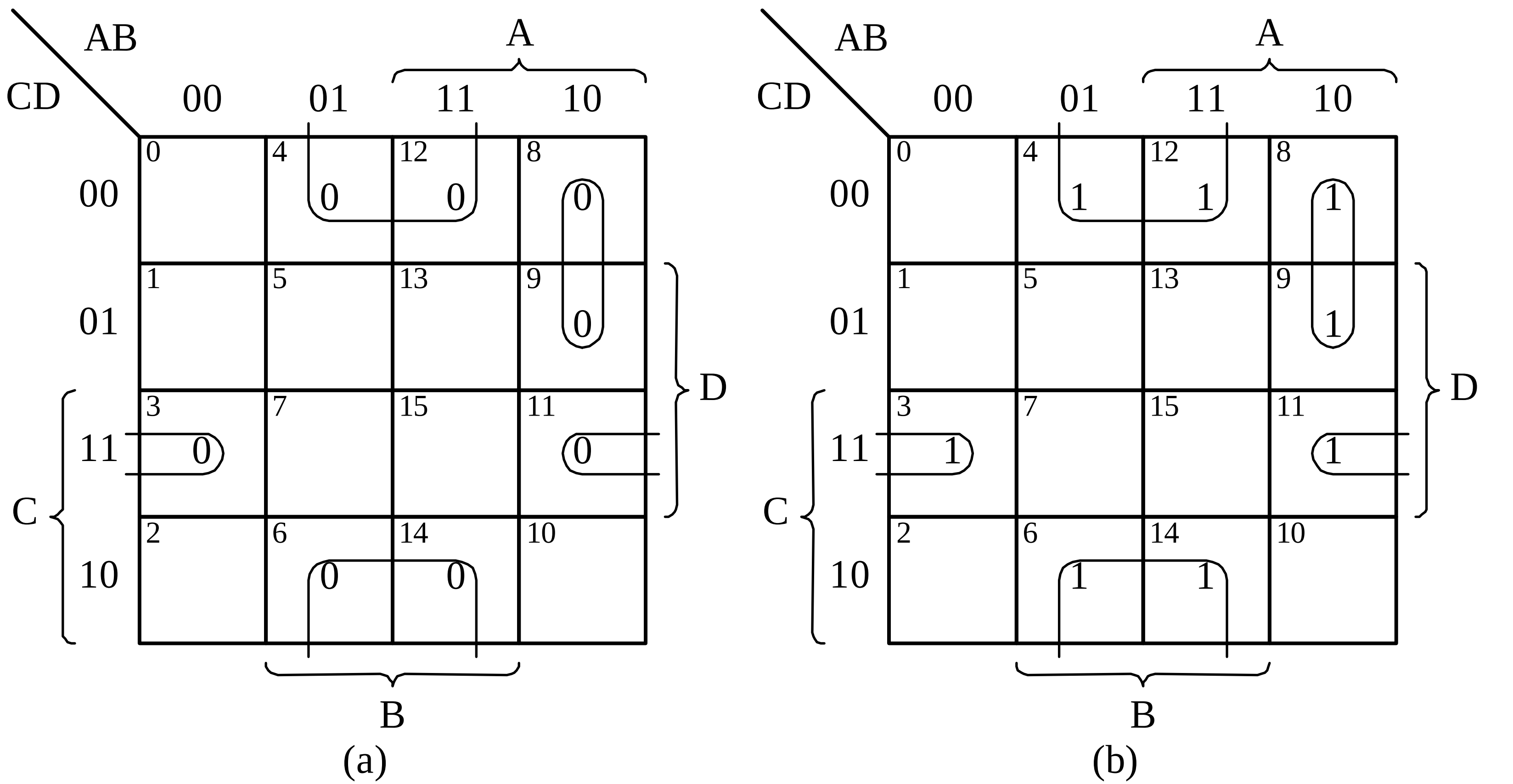


Figure 3.21

Figure 3.22 Finding a minimal POS expression for a 5-variable function.

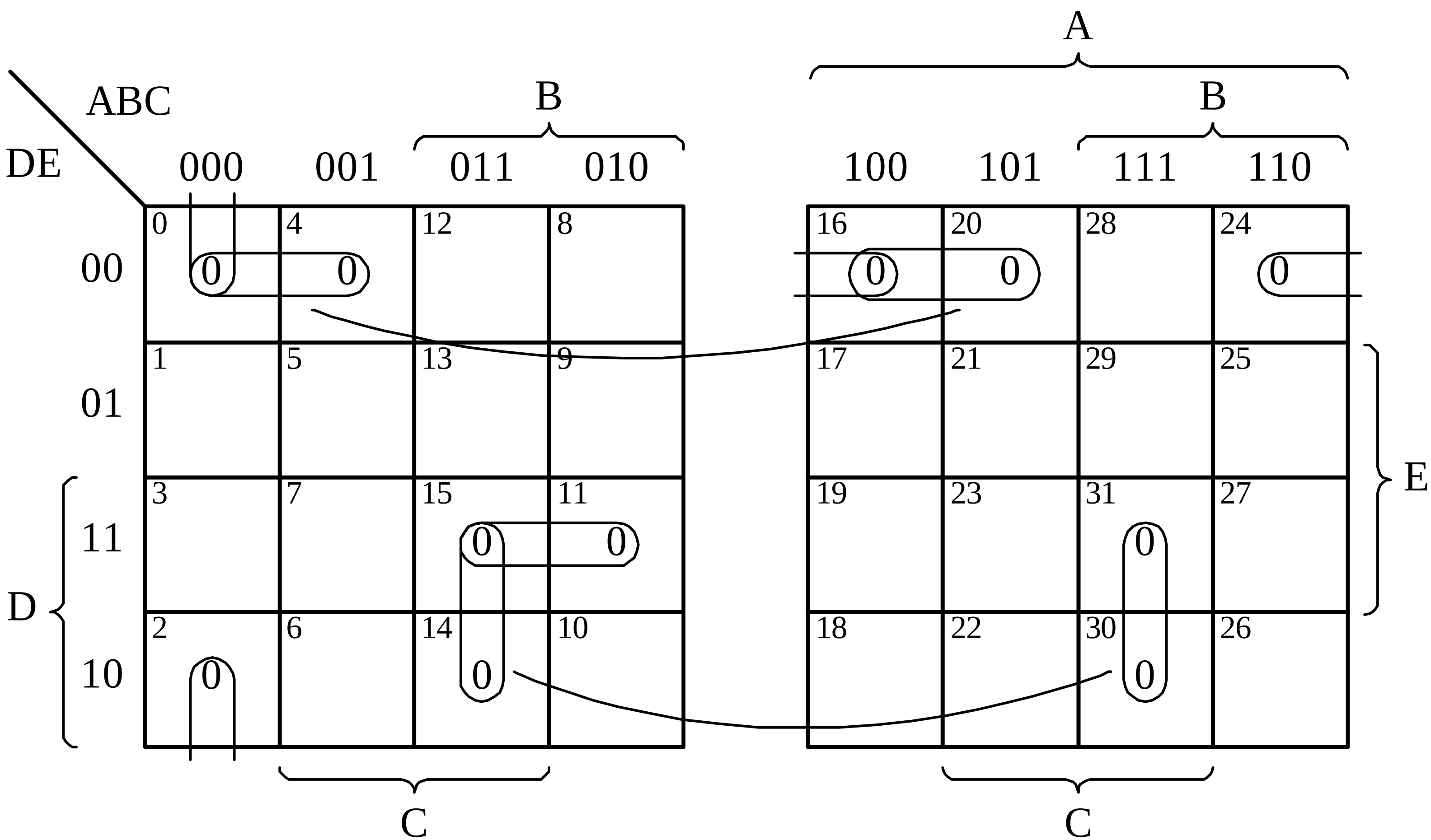
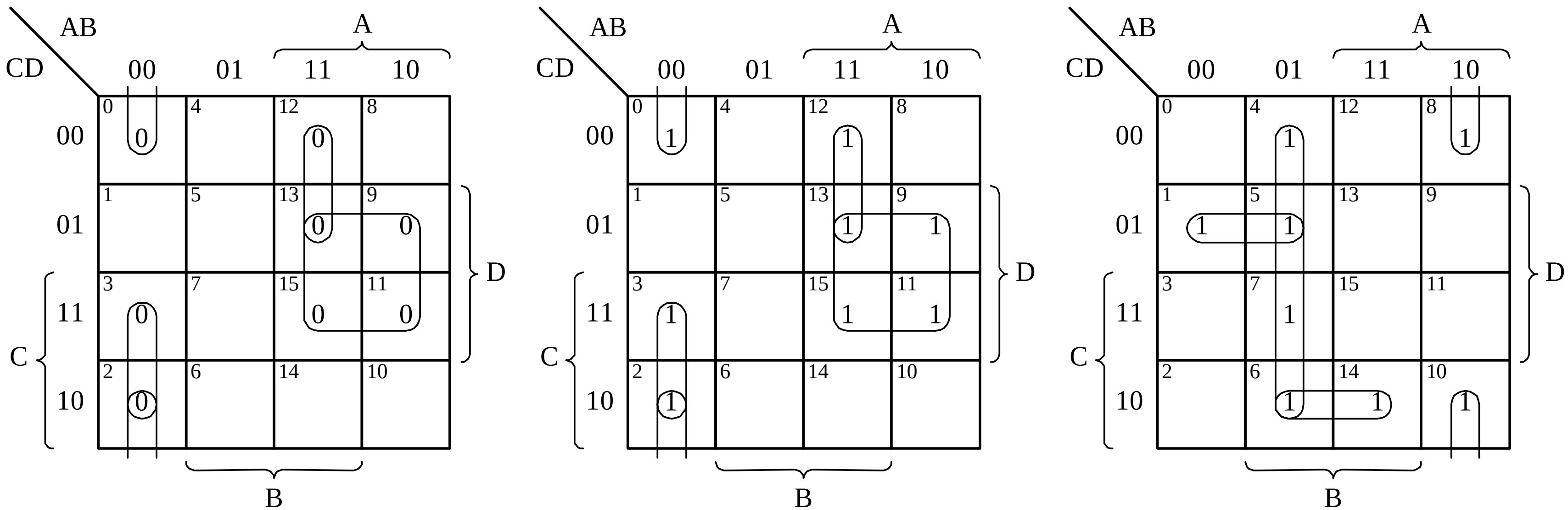


Figure 3.23 Deriving POS and SOP forms of a function.



(a) (b) (c)

Example 3.22 -- Minimizing a Function with Don't Cares.

$f(A,B,C,D) = \sum m(1,3,4,7,11) + d(5,12,13,14,15)$

$= \prod M(0,2,6,8,9,10) \bullet D(5,12,13,14,15)$

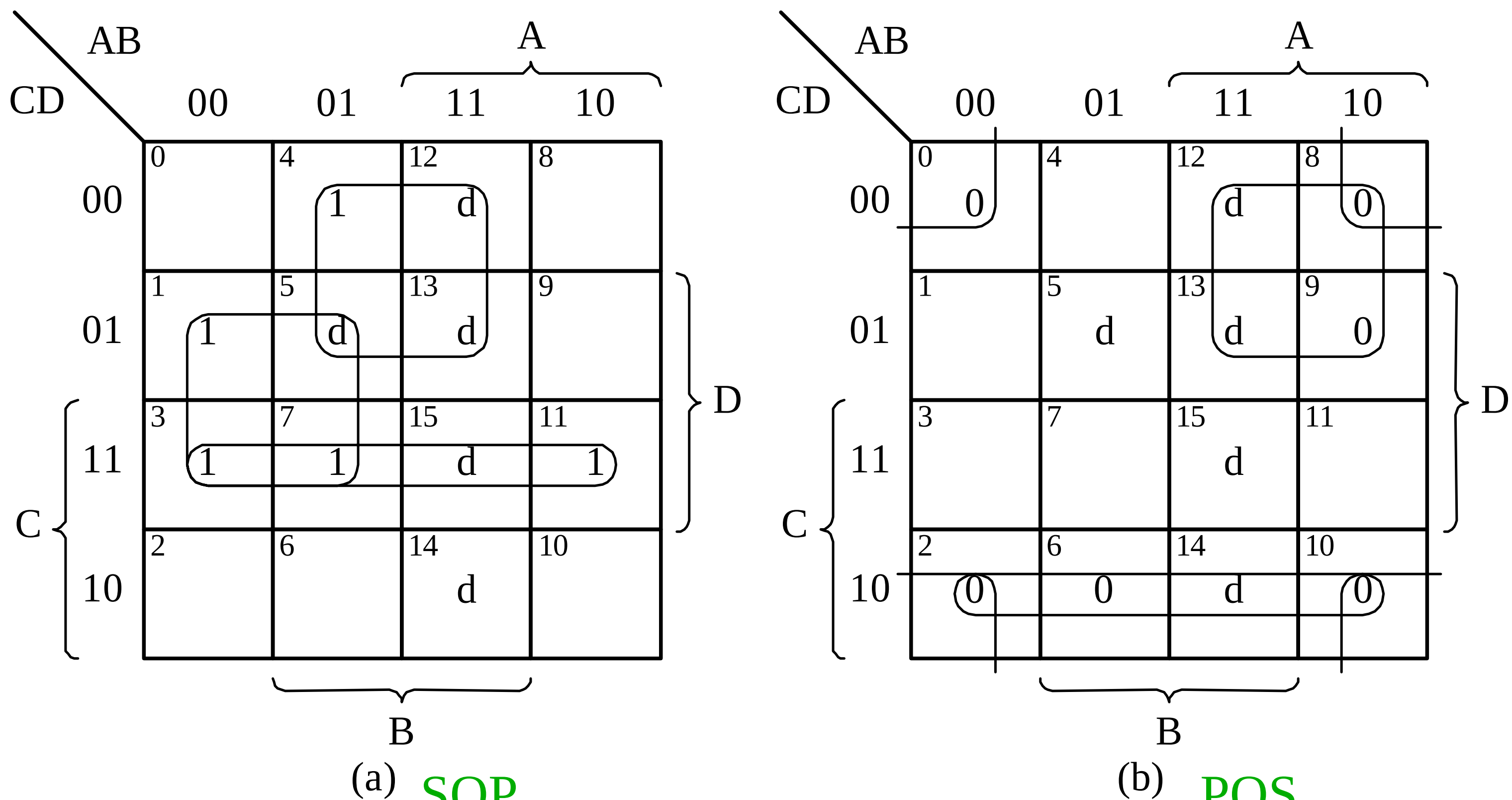
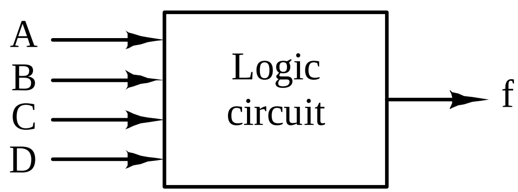


Figure 3.24 K-maps for Example 3.22.

Example 3.23 -- 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)

Figure 3.25 -- block diagram and truth table.

Example 3.23 (concluded)

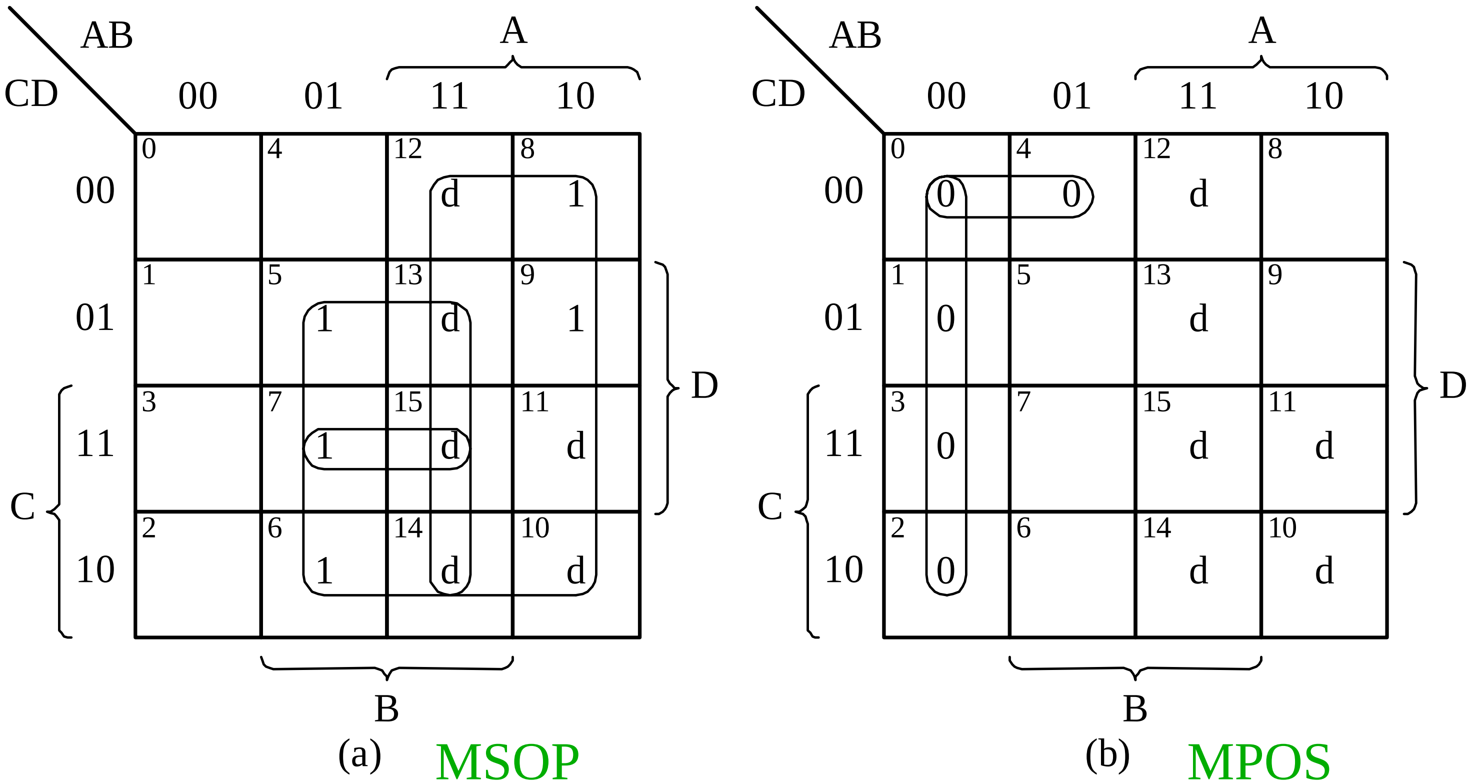


Figure 3.26 Use of don't cares for SOP and POS forms.

$$f(A,B,C,D) = A + BD + BC; \quad f(A,B,C,D) = (A + B)(A + C + D)$$

Timing Hazards in Combinational Logic Circuits

- *Hazards* are undesirable changes in the output of a combinational logic circuit caused by unequal gate propagation delays.
- *Static hazard* (glitch) -- the output momentarily changes from the correct or static state
  - *Static 1 hazard* -- the output changes from 1 to 0 and back to 1
  - *Static 0 hazard* -- the output changes from 0 to 1 and back to 0
- *Dynamic hazard* (bounce) -- the output changes multiple times during a change of state
  - *Dynamic 0 to 1 hazard* -- the output changes from 0 to 1 to 0 to 1
  - *Dynamic 1 to 0 hazard* -- the output changes from 1 to 0 to 1 to 0

Figure 3.27 (a)--(b) Illustration of a static hazard.

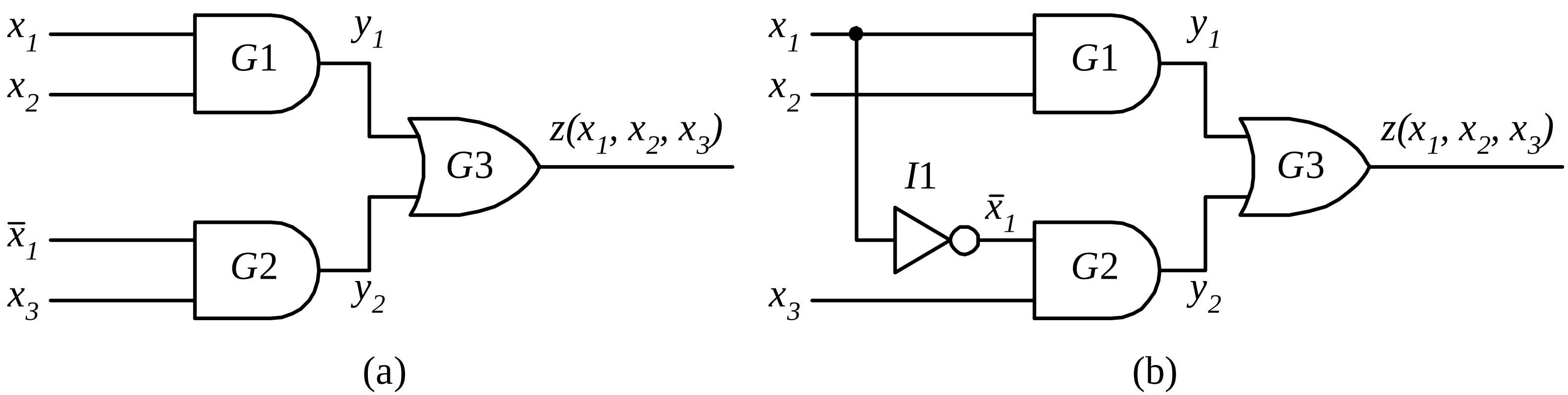
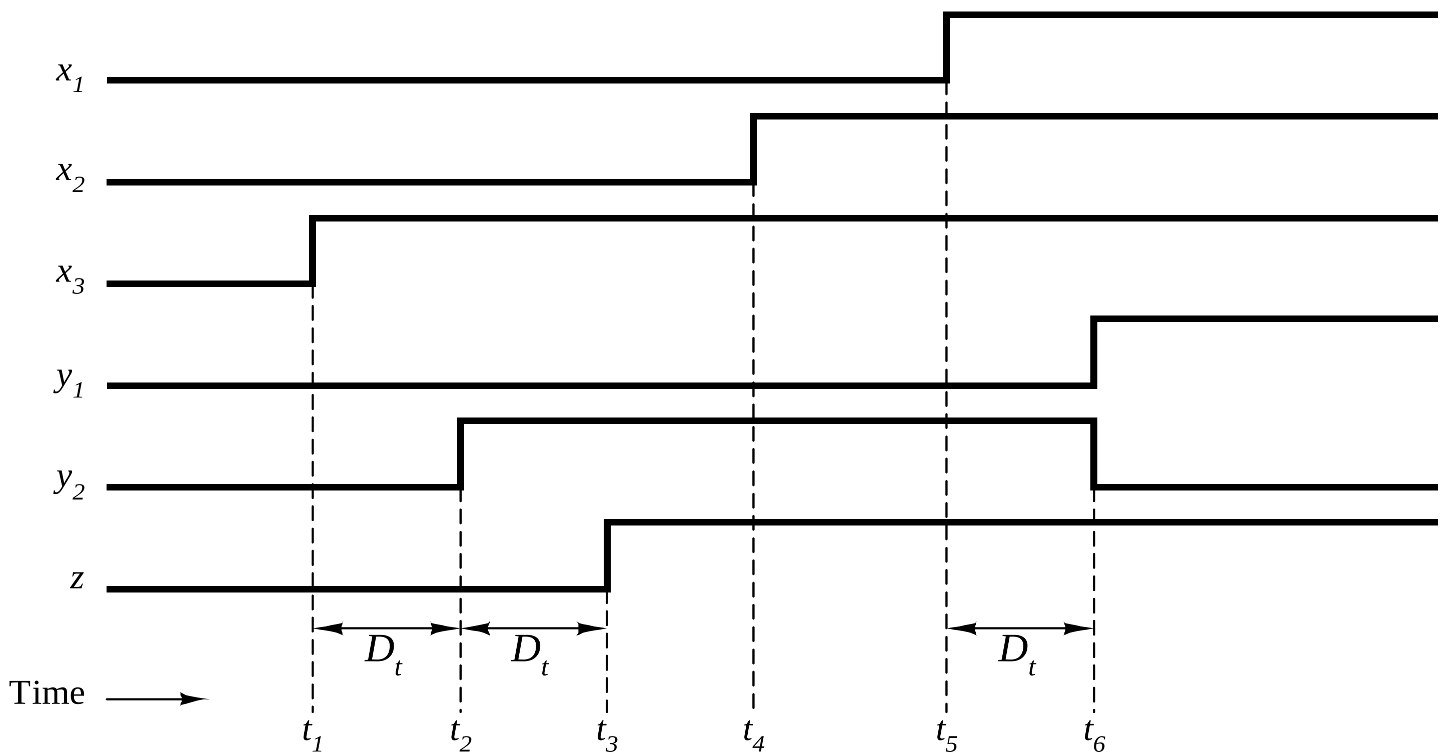
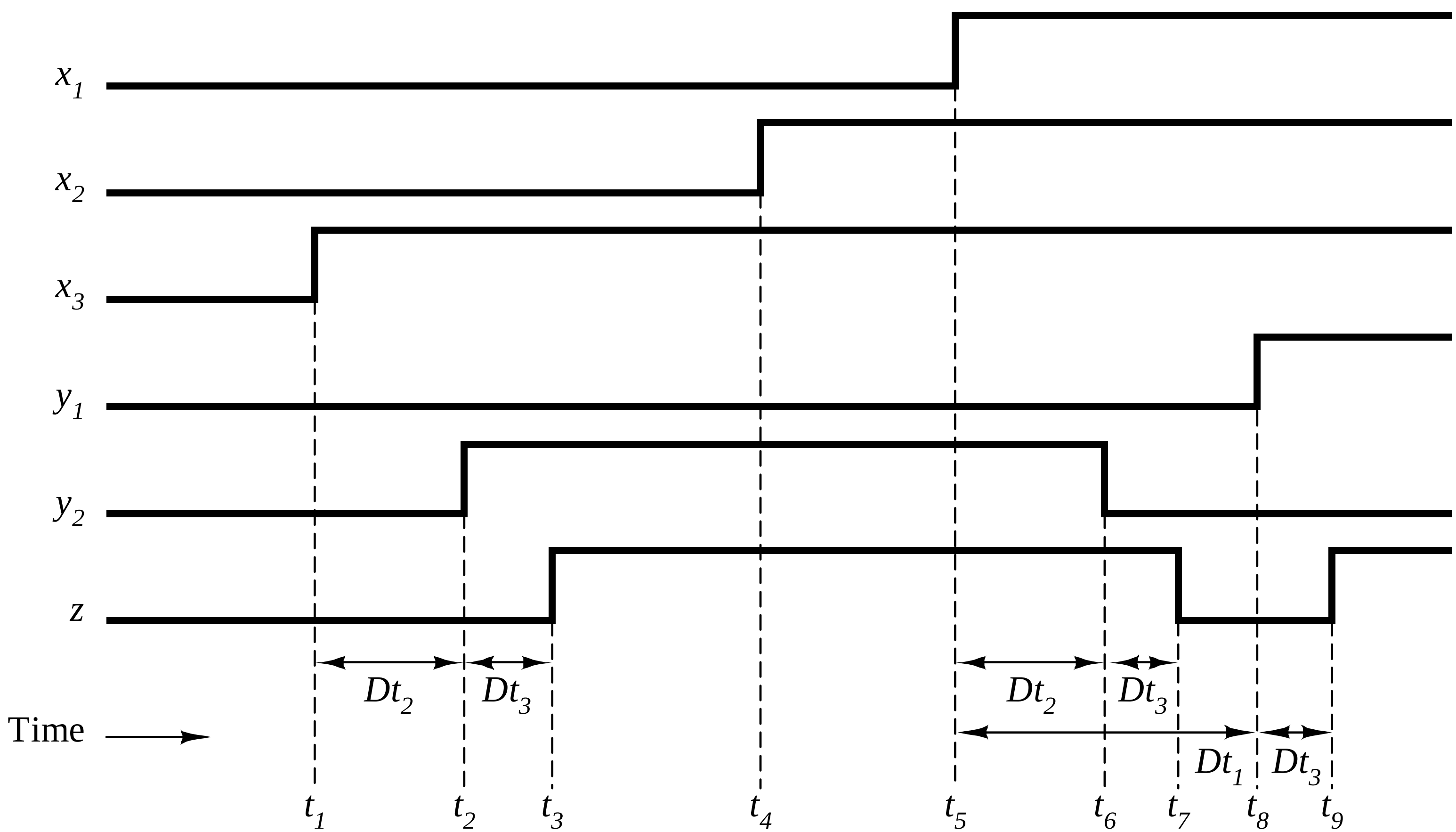


Figure 3.27 (c) Illustration of a static hazard (con't)



(c)

Figure 3.27 (d) Illustration of a static hazard (con't).



(d)

Figure 3.28 Identifying hazards on a K-map.

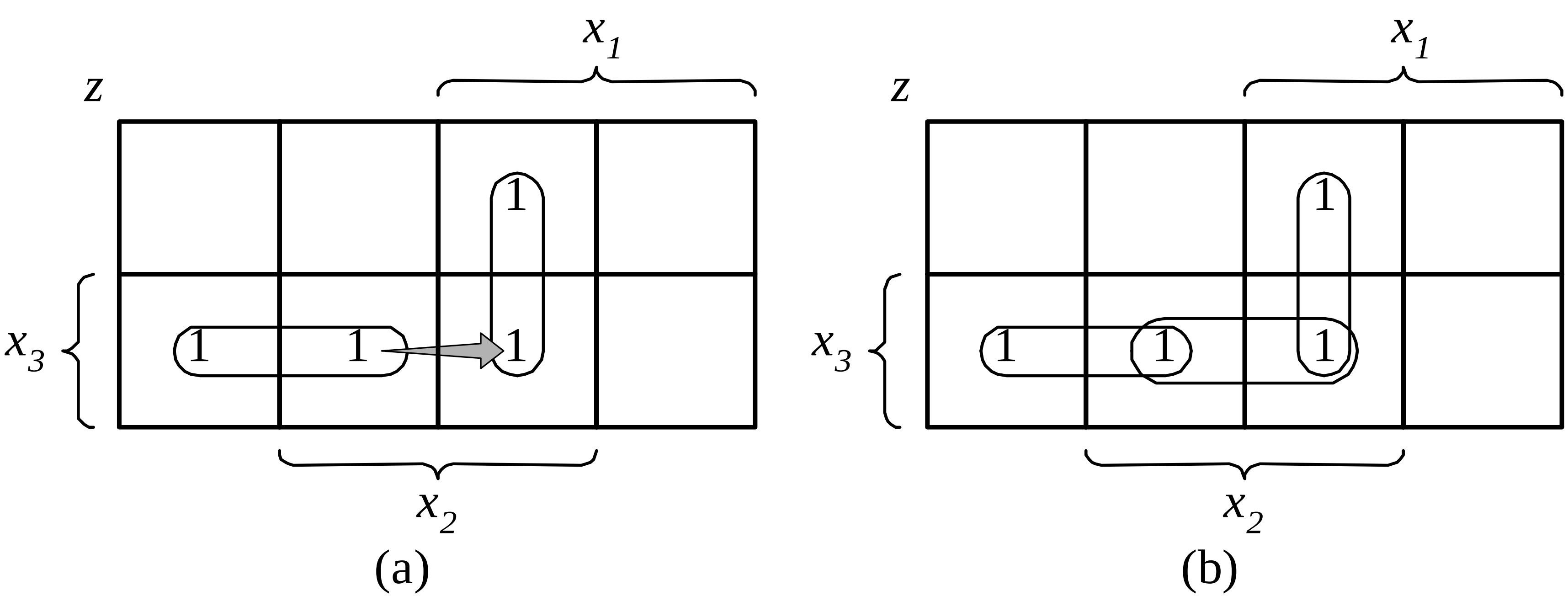


Figure 3.29 Hazard-free network.

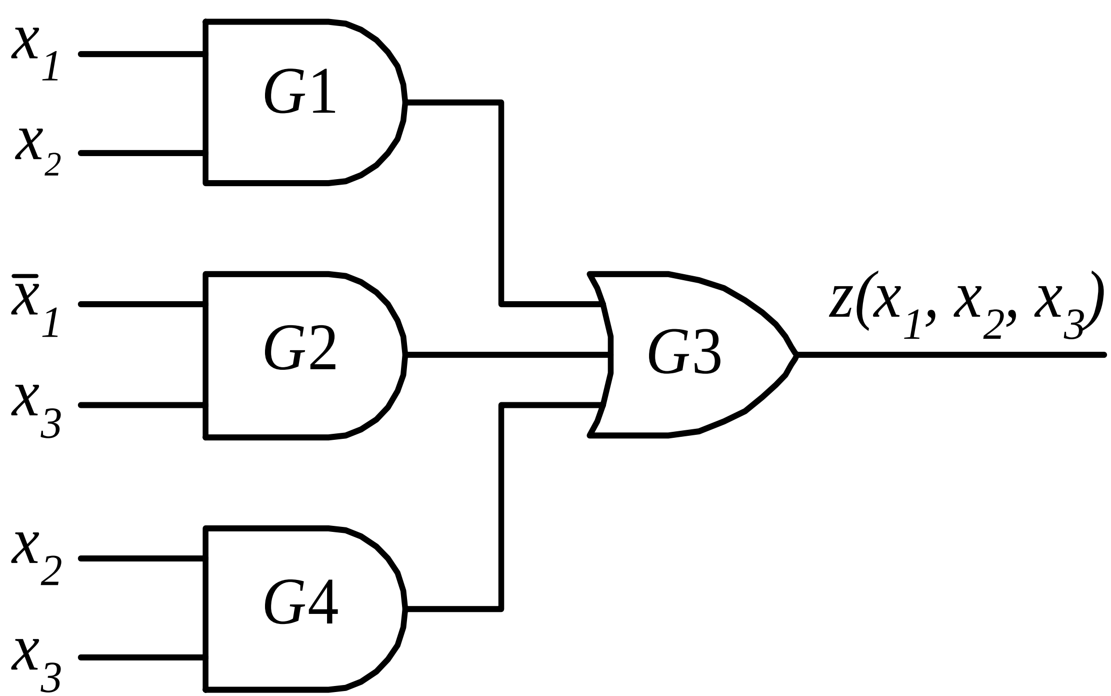
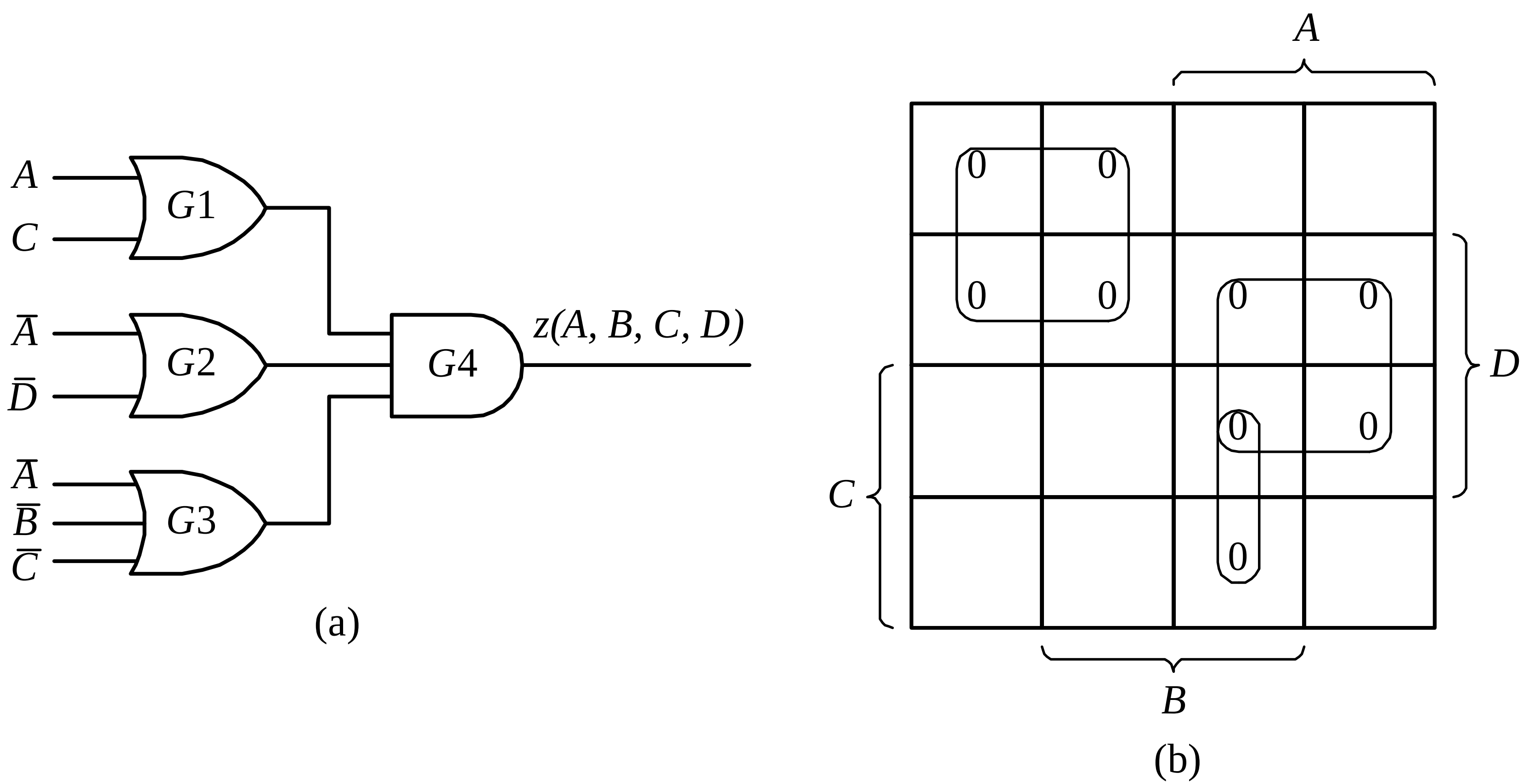
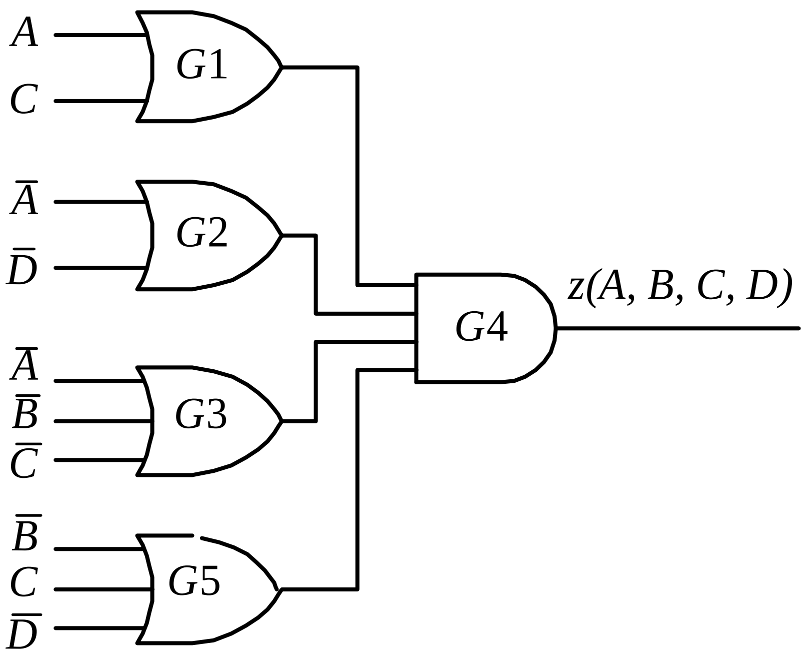


Figure 3.30 (a)--(b) Example of a static-0 hazard.

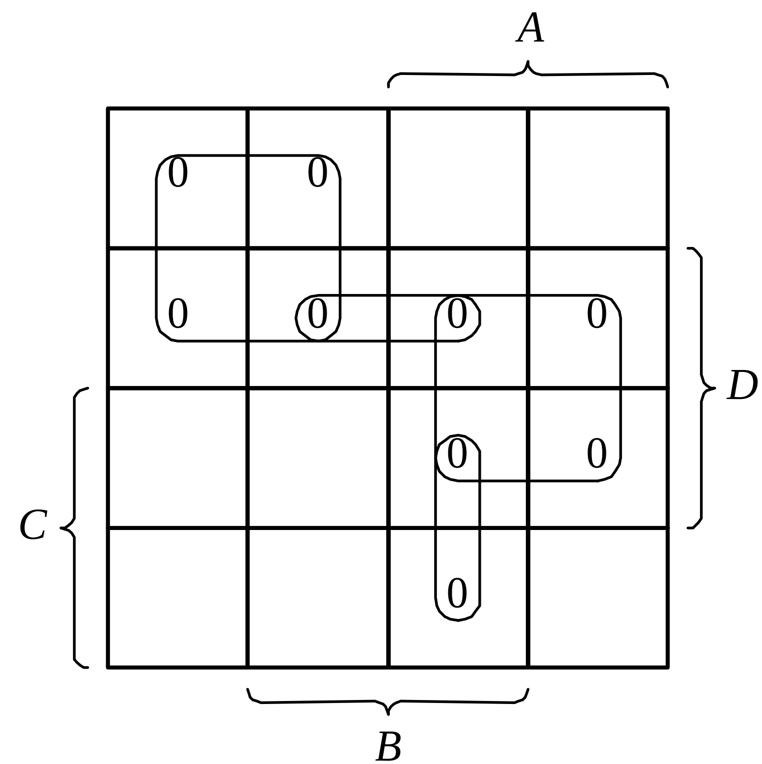


(a) (b)

Figure 3.30 (c)--(d) Example of a static-0 hazard (con't).

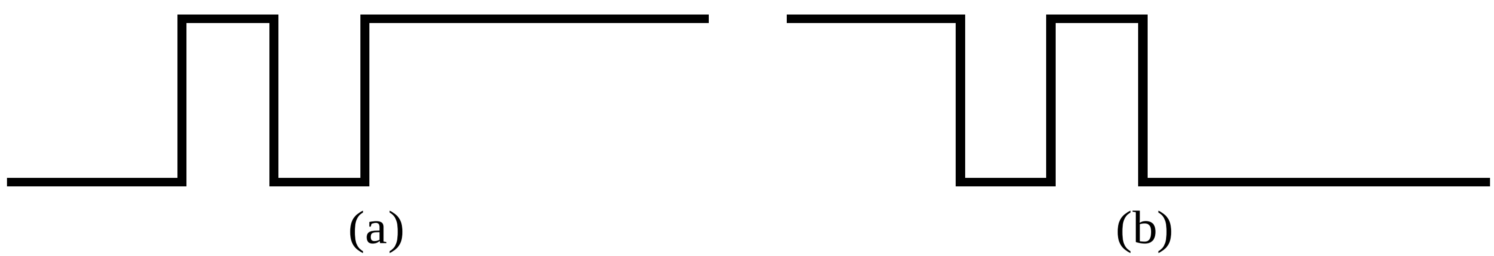


(c)



(d)

Figure 3.31 Dynamic hazards.



3.8 Quine-McCluskey Minimization Method

- Advantages over K-maps
  - Can be computerized
  - Can handle functions of more than six variables
- Overview of the method
  - Given the minterms of a function
  - Find all prime implicants (steps 1 and 2)
    - Partition minterms into groups according to the number of 1's
    - Exhaustively search for prime implicants
  - Find a minimum prime implicant cover (steps 3 and 4)
    - Construct a prime implicant chart
    - Select the minimum number of prime implicants

Example 3.24 -- Use the Q-M method to find the MSOP of the function

$f(A,B,C,D) = \prod m(2,4,6,8,9,10,12,13,15)$

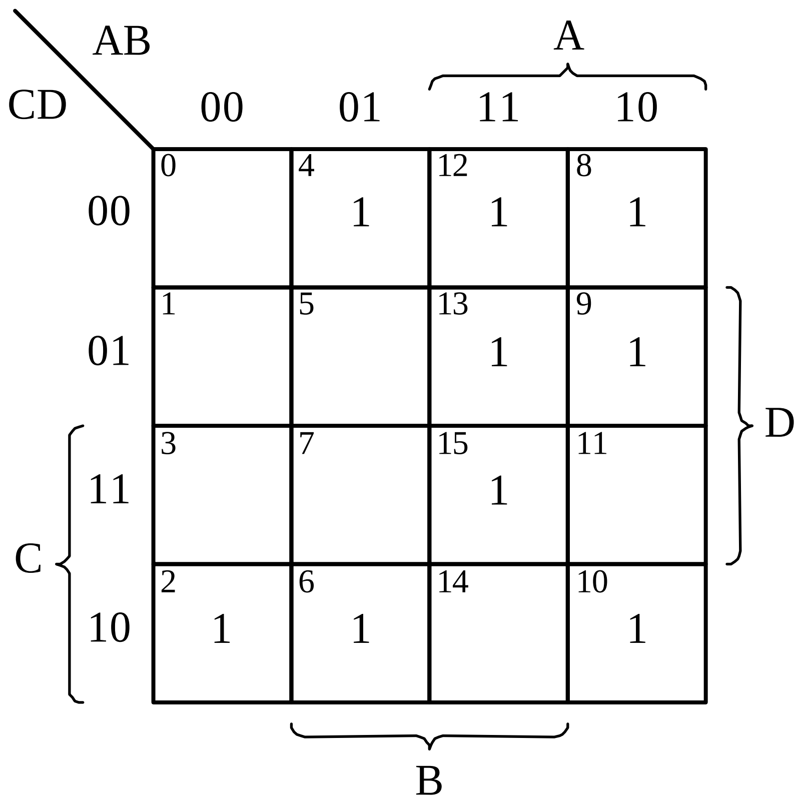


Figure 3.32 K-map for example 3.30.

QIAU

Step 1 -- List Prime Implicants in Groups (Example 3.24)

Minterms	ABCD	
2	0010	
4	0100	Group 1 (a single 1)
8	1000	
6	0110	
9	1001	Group 2 (two 1's)
10	1010	
12	1100	
13	1101	Group 3 (three 1's)
15	1111	Group 4 (four 1's)

Step 2 -- Generate Prime Implicants (Example 3.24)

List 1

Minterm	ABCD	
2	0010	✓
4	0100	✓
8	1000	✓
6	0110	✓
9	1001	✓
10	1010	✓
12	1100	✓
13	1101	✓
15	1111	✓

List 2

Minterms	ABCD	
2, 6	0-10	PI_2
2, 10	-010	PI_3
4, 6	01-0	PI_4
4, 12	-100	PI_5
8, 9	100-	✓
8, 10	10-0	PI_6
8, 12	1-00	✓
9, 13	1-01	✓
12, 13	110-	✓
13, 15	11-1	PI_7

List 3

Minterms	ABCD	
8, 9, 12, 13	1-0-	PI_1

Step 3 -- Prime Implicant Chart (Example 3.24)

	2	4	6	√ 8	√ 9	10	√ 12	√ 13	√ 15
* * PI_1				×	⊗		×	×	
PI_2	×		×						
PI_3	×					×			
PI_4		×	×						
PI_5		×					×		
PI_6				×		×			
* * PI_7								×	⊗

Step 4 -- Reduced Prime Implicant Chart (Example 3.24)

	√ 2	√ 4	√ 6	√ 10
PI_2	×		×	
$*PI_3$	×			×
$*PI_4$		×	×	
PI_5		×		
PI_6				×

The Resulting Minimal Realization of *f*

$$f(A,B,C,D) = PI_1 + PI_3 + PI_4 + PI_7$$

$$= 1\text{-}0\text{-} + \text{-}010 + 01\text{-}0 + 11\text{-}1$$

$$= AC' + B'CD' + A'BD' + ABD$$

How the Q-M Results Look on a K-map

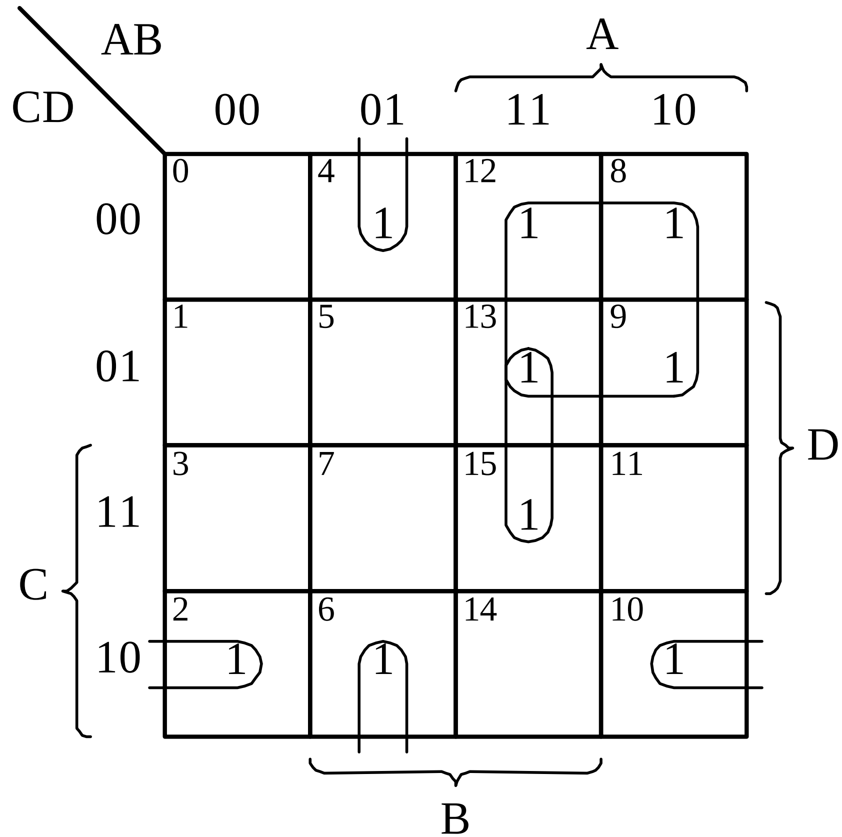


Figure 3.33 Grouping of terms.

Covering Procedure

Step 1 -- Identify any minterms covered by only one PI. Select these PIs for the cover.

Step 2 -- Remove rows covered by the PIs identified in step 1. Remove minterms covered by the removed rows.

Step 3 -- If a cyclic chart results from step 2, go to step 5. Otherwise, apply the reduction procedure of steps 1 and 2.

Step 4 -- If a cyclic chart results from step 3, go to step 5. Otherwise return to step 1.

Step 5 -- Apply the cyclic chart procedure. Repeat step 5 until a void chart or noncyclic chart chart is produced. In the latter case, return to step 1.

Coverage Example

$f(A,B,C,D) = \sum m(0,1,5,6,7,8,9,10,11,13,14,15)$

	√	√		√	√	√	√				√	√
	0	1	5	6	7	8	9	10	11	13	14	15
* * PI_1	⊗	×				×	×					
PI_2		×	×				×			×		
PI_3			×		×					×		×
PI_4						×	×	×	×			
PI_5							×		×	×		×
PI_6								×	×		×	×
* * PI_7				⊗	×						×	×

Reduced PI Charts

	5	10	11	13
PI_2	×			×
PI_3	×			×
PI_4		×	×	
PI_5			×	×
PI_6		×	×	

	√ 5	√ 10
* PI_2	×	
* PI_4		×

Cyclic PI Charts

1. No essential PIs.
2. No row or column coverage.

	√		√			
	1	2	3	4	5	6
$*PI_1$	×		×			
PI_2		×	×			
PI_3		×				×
PI_4				×		×
PI_5				×	×	
PI_6	×				×	

	2	4	5	6
PI_2	×			
PI_3	×			×
PI_4		×		×
PI_5		×	×	
PI_6			×	

	√	√	√	√
	2	4	5	6
$*PI_3$	×			×
PI_4		×		×
$*PI_5$		×	×	

Using the Q-M Procedure with Incompletely Specified Functions

1. Use minterms and don't cares when generating prime implicants

2. Use only minterms when finding a minimal cover

Example 3.25 -- Find a minimal sum of products of the following function using the Quine-McCluskey procedure.

Minimizing Table for Example 3.25

List 1

Minterm	ABCDE	
2	00010	√
3	00011	√
5	00101	√
10	01010	√
12	01100	PI_7
18	10010	√
7	00111	√
19	10011	√
21	10101	√
15	01111	√
23	10111	√
27	11011	√

List 2

Minterms	ABCDE	
2, 3	0001-	√
2, 10	0-010	PI_4
2, 18	-0010	√
3, 7	00-11	√
3, 19	-0011	√
5, 7	001-1	√
5, 21	-0101	√
18, 19	1001-	√
7, 15	0-111	PI_5
7, 23	-0111	√
19, 23	10-11	√
19, 27	1-011	PI_6
21, 23	101-1	√

List 3

Minterms	ABCDE	
2, 3, 18, 19	-001-	PI_1
3, 7, 19, 23	-0-11	PI_2
5, 7, 21, 23	-01-1	PI_3

PI Chart for Example 3.25

	2	3	7	10	12	15	27
	√		√	√	√	√	√
PI_1	×	×					
PI_2		×	×				
PI_3			×				
* * PI_4	×			⊗			
* * PI_5			×			⊗	
* * PI_6							⊗
* * PI_7					⊗		

Results of Minimization for Example 3.25

$$f(A,B,C,D,E) = PI_1 + PI_4 + PI_5 + PI_6 + PI_7 \textbf{ OR}$$

$$= PI_2 + PI_4 + PI_5 + PI_6 + PI_7$$