Chapter 3 The Karnaugh Map

- A graphic approach to simplify any function
- No guarantee of finding a minimum solution; sum of product and product of sum

- Definitions

Boolean function $f(x) : B^n \rightarrow B$

$$f(x) : \{0,1\}^n \rightarrow \{0,1\}$$

letter : a constant or a variable

literal : a letter or its complement

EX) B = {0,1}, variable : x_1, x_2

letter : x_1, x_2, 0, 1

literal : x_1, x_2, 0, 1, x_1', x_2'

Review on product and sum term

- Product term (product, term)

  1

  a non-constant literal

  a conjunction of non-constant literals where no letter appears more than once

  $1, \quad x_1, \quad x_1'x_2, \quad x_1x_2x_1'$ (X), $\quad x_1+x_2'$ (X)

- Sum term

  0

  a non-constant literal

  a disjunction of non-constant literals where no letter appears more than once

  $0, \quad x_1, \quad x_1+x_2', \quad x_1'x_2$ (X), $\quad x_1+x_2+x_1'$ (X)

Minterm and Maxterm

- Minterm

  : 모든 변수가 항상 한번씩 사용된 product term(곱항)

  Ex) 변수 : X, Y, Z

  X′YZ, XYZ′, XYZ, X′Y′Z′, XY (X), X′Z (X)

- Maxterm

  : 모든 변수가 항상 한번씩 사용된 sum term(합항)

  Ex) 변수 : X, Y, Z

  X+Y+Z, X′+Y+Z′, X′+Y′+Z′, X′+YZ (X), X+Z′ (X)

Ex) 3개의 변수에 대한 Minterm 과 Maxterm

변수	Minterm		Maxterm	
A B C	Minterm	표시	Maxterm	표시
0 0 0	A'B'C'	m0	A + B + C	M0
0 0 1	A'B'C	m1	A + B + C'	M1
0 1 0	A'BC'	m2	A + B' + C	M2
0 1 1	A'BC	m3	A + B' + C'	M3
1 0 0	AB'C'	m4	A' + B + C	M4
1 0 1	AB'C	m5	A' + B + C'	M5
1 1 0	ABC'	m6	A' + B' + C	M6
1 1 1	ABC	m7	A' + B' + C'	M7

Boolean function

- Sum of product (SOP), disjunctive normal form (DNF), ΣΠ

  $$f = x_1x_2' + x_2'x_3 + x_1x_3'$$

- Product of sum (POS), conjunctive normal form (CNF), ΠΣ

  $$f = (x_1 + x_2')(x_2 + x_3)(x_3 + x_1)$$

- Canonical sum of product : sum of minterms

  $$\begin{aligned} f(A,B,C) &= A'B'C + A'BC + AB'C + ABC \\ &= m_1 + m_3 + m_5 + m_7 \\ &= \Sigma\,(1, 3, 5, 7) \end{aligned}$$

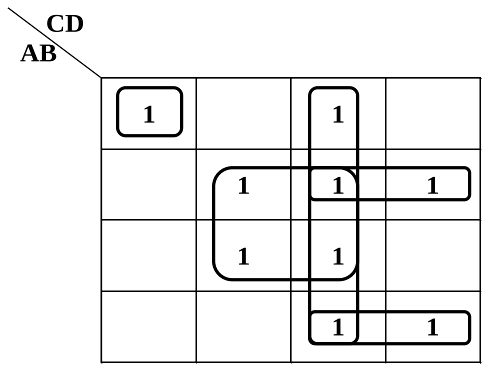
- Canonical product of sum : product of maxterms

  $$\begin{aligned} f(A,B,C) &= (A+B+C)(A+B'+C)(A'+B+C)(A'+B'+C) \\ &= M_0 M_2 M_4 M_6 \\ &= \Pi\,(0, 2, 4, 6) \end{aligned}$$

Karnaugh Map (1953. NOV)

- F(A,B,C,D) = A′B′C′D′+A′BC+AB′C+BD+CD (SOP)

  = Σ (0,3,5,6,7,10,11,13,15) canonical SOP



- Implicant : 2^k 개의 1의 묶음
- Prime Implicant : 더 큰 묶음(Implicant)에 포함되지 않는 묶음
- Essential Prime Implicant

  하나의 prime Implicant를 형성하고 있는 1 들 중에서 적어도 하나는 다른 Implicant에 속하지 않고 자신의 Prime Implicant 묶음에만 속하는 Prime Implicant
- 간략화된 함수에는

  E.P.I 전부와 non-essential P.I 일부 포함

Karnaugh Map

BC \ A	0	1
00		1
01		
11	1	1
10		1

$F = \underline{BC} + \underline{AC'} + (\ + \underline{AB})$

PI PI PI

EPI

- Implicant

  : a product term *p* that is included in the function *f* (p $\leq$ f)

  Ex) $f = xy' + yz$, xy'(PI), xyz

- PI (Prime Implicant)

  : an implicant that is not included in any other implicant of *f* (canot be combined with another term to eliminate a variable)

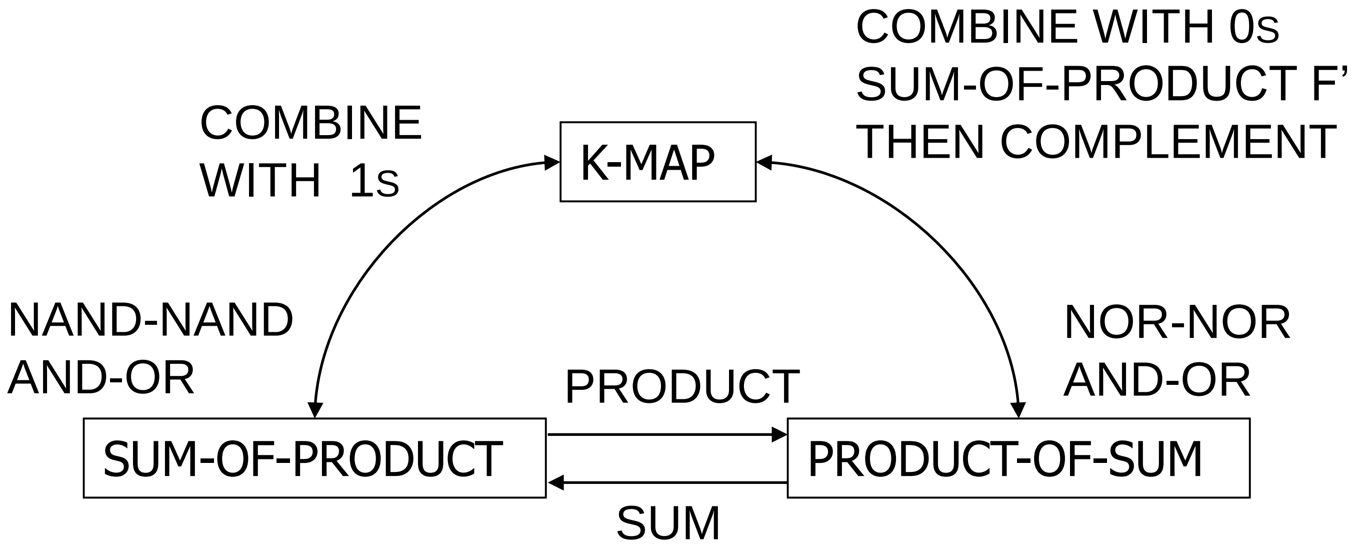
- EPI (Essential Prime Implicant)

  : a PI which includes a minterm that is not included in any other P.I.

Optimization Algorithm

- Find <u>all</u> prime implicants.
- Include <u>all</u> essential prime implicants in the solution
- Select a minimum cost set of non-essential prime implicants to cover all minterms not yet covered:
  - Obtaining an optimum solution: See Reading Supplement - More on Optimization
  - Obtaining a good simplified solution: Use the Selection Rule

Karnaugh Map



2 variable map

A′B′	**A B′**
A′B	**A B**

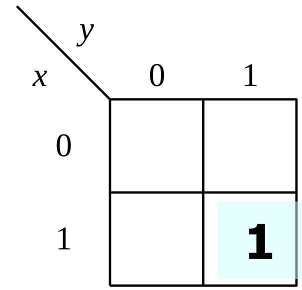
		A
	m_0	m_2
B	m_1	m_{03}

$B \backslash A$	0	1
0	*0*	*2*
1	*1*	*3*

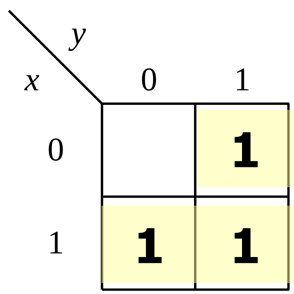
2 variable map example

x	y	AND	OR	XOR
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

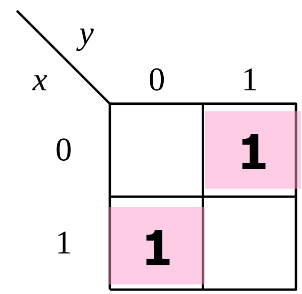
(a) Truth table



(b) AND: *xy*



(c) OR: *x + y*



(d) XOR: *x′y + xy′*

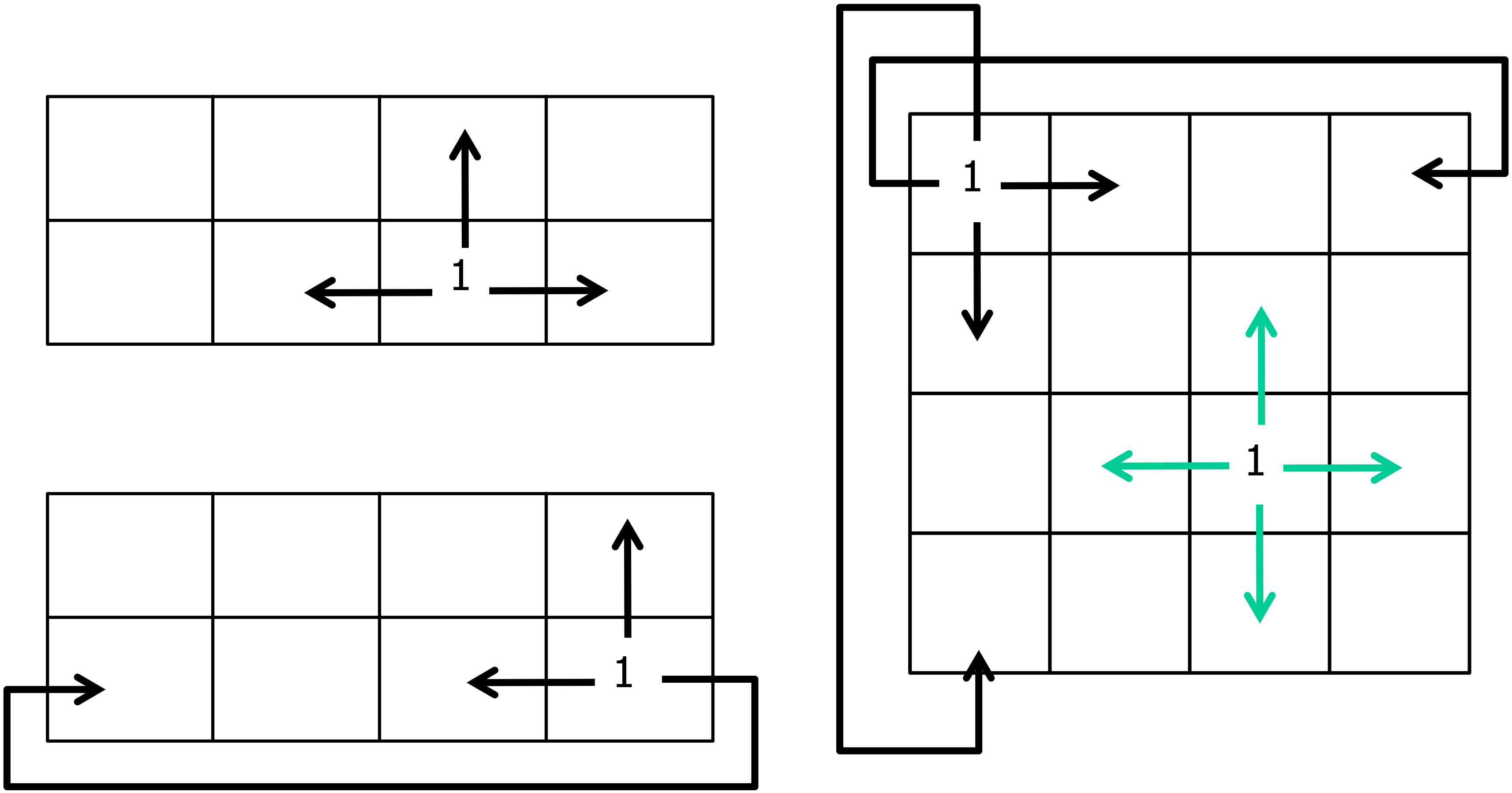
Three-variable Maps

C \ AB		A'B' 00	A'B 01	A B 11	A B' 10
C'	0	A'B'C'	A'B C'	A B C'	A B'C'
C	1	A'B'C	A'B C	A B C	A B'C

C \ AB	00	01	11	10
0	0	2	6	4
1	1	3	7	5

A \ B C	00	01	11	10
0	0	1	3	2
1	4	5	7	6

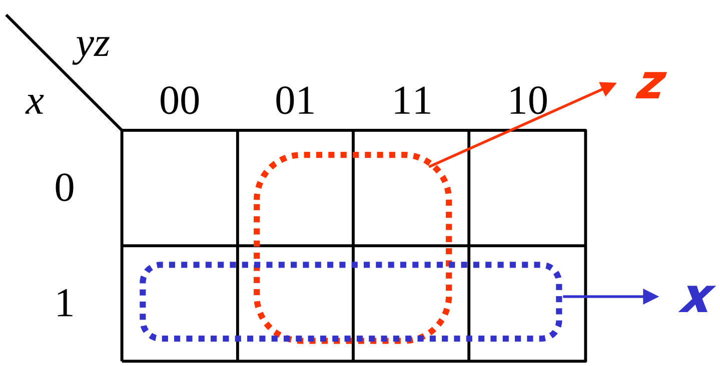
Adjacencies on three- and four-variable maps



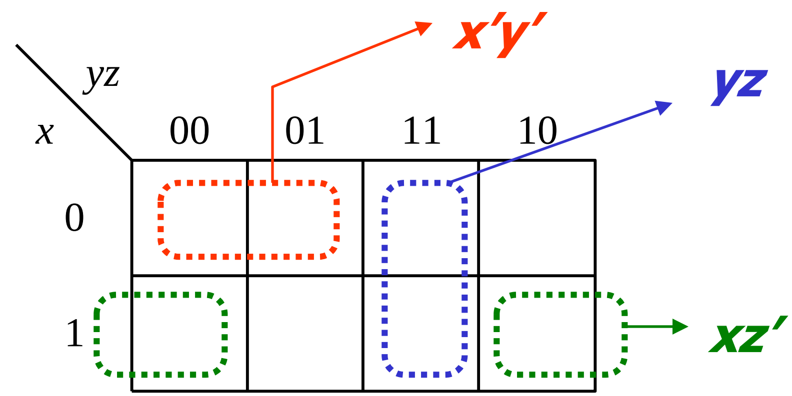
Three-variable map

x \ yz	00	01	11	10
0	$x'y'z'$	$x'y'z$	$x'yz$	$x'yz'$
1	$xy'z'$	$xy'z$	xyz	xyz'

(a) Map organization



(b) Example subcubes of size 4



(c) Example subcubes of size 2

Map representation of carry and sum functions

c_i	x_i	y_i	c_{i+1}	s_i
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

(a) Truth table

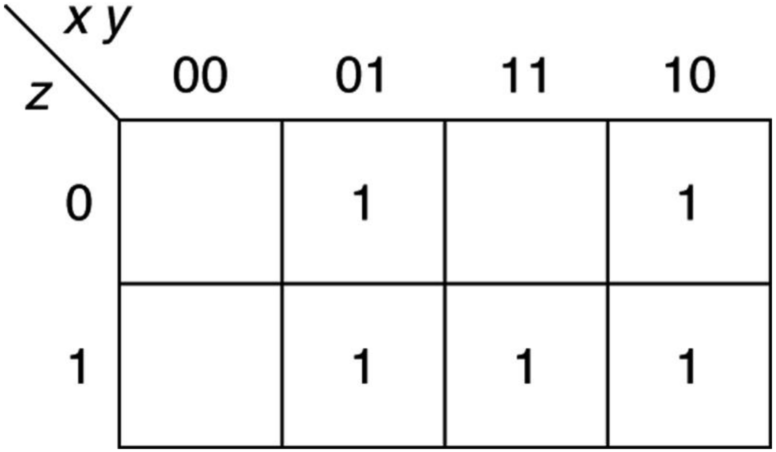
c_i \ $x_i y_i$	00	01	11	10
0			1	
1		1	1	1

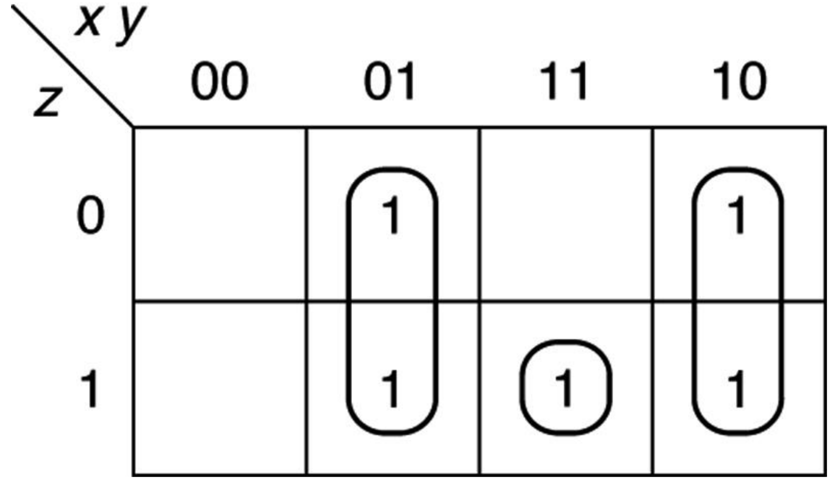
(b) Carry function: $c_{i+1} = x_i y_i + c_i x_i + c_i y_i$

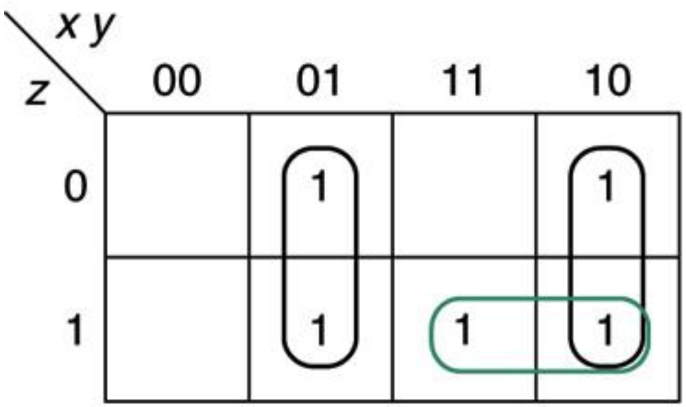
c_i \ $x_i y_i$	00	01	11	10
0		1		1
1	1		1	

(c) Sum function: $s_i = x'_i y'_i c_i + x'_i y_i c'_i + x_i y'_i c'_i + x_i y_i c_i$

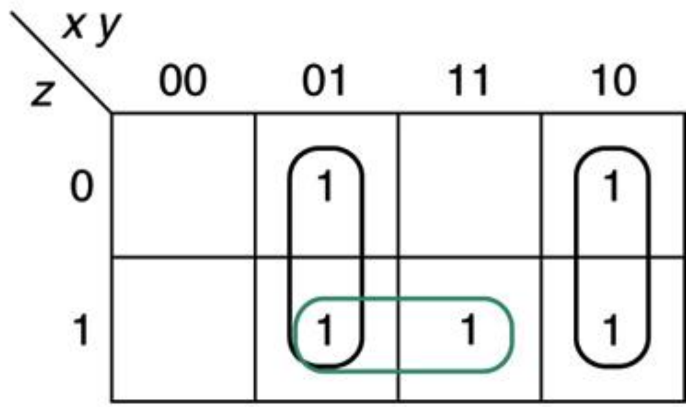
Example $x'yz' + x'yz + xy'z' + xy'z + xyz.$







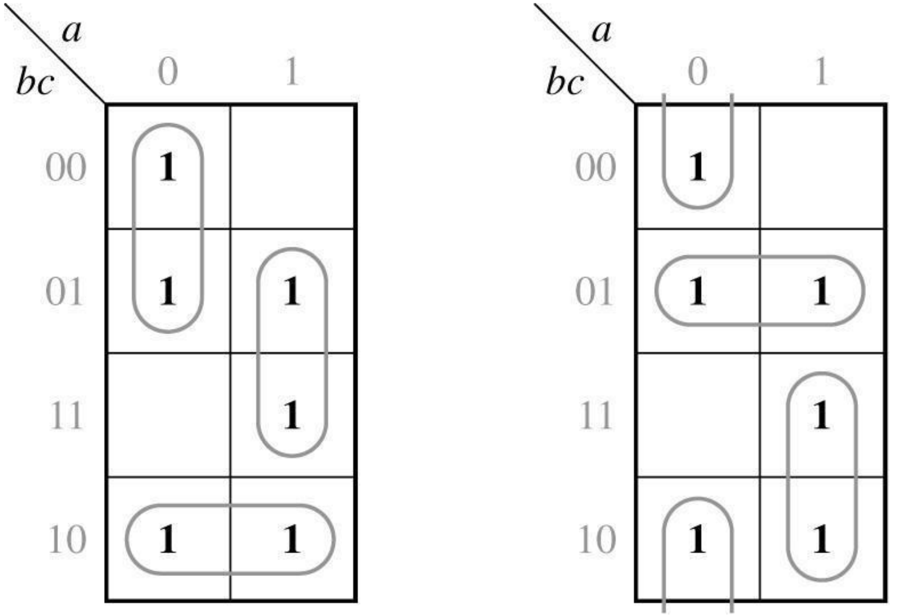
$x'y + xy' + xz$



$x'y + xy' + yz$

Example Function with Two Minimal Forms

$$F = \sum m(0,1,2,5,6,7)$$



$F = a'b' + bc' + ac$

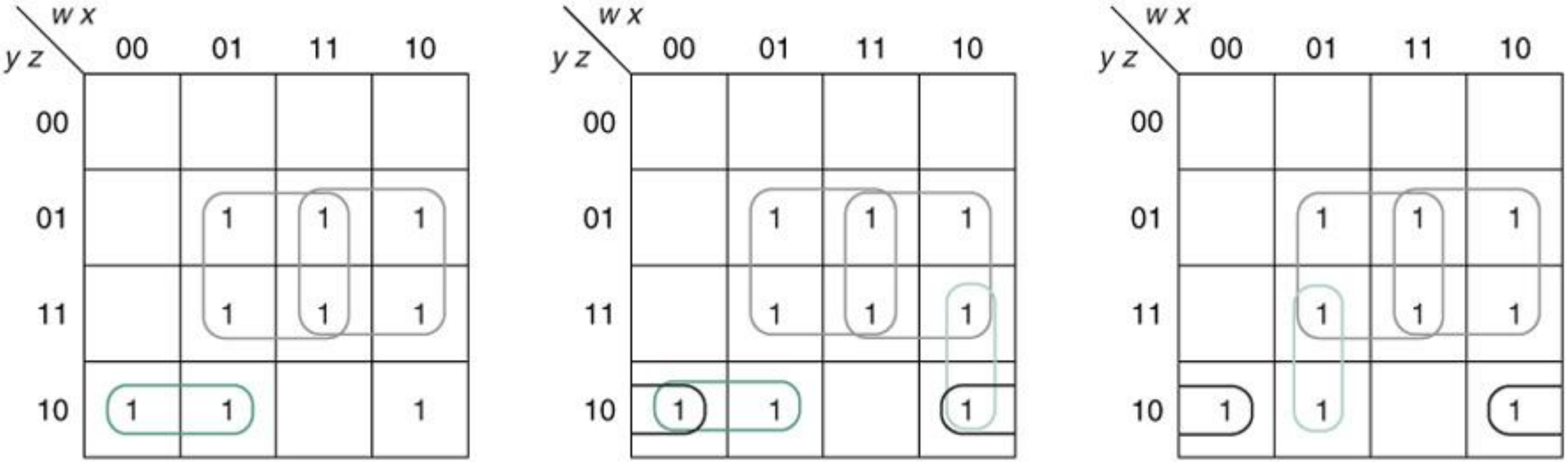
$F = a'c' + b'c + ab$

Four-variable Maps

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

CD \ AB	00	01	11	10
00	A′B′C′D′	A′B C′D′	A B C′D′	A B′C′D′
01	A′B′C′D	A′B C′D	A B C′D	A B′C′D
11	A′B′C D	A′B C D	A B C D	A B′C D
10	A′B′C D′	A′B C D′	A B C D′	A B′C D′

$g(w, x, y, z) = \Sigma m(2, 5, 6, 7, 9, 10, 11, 13, 15)$

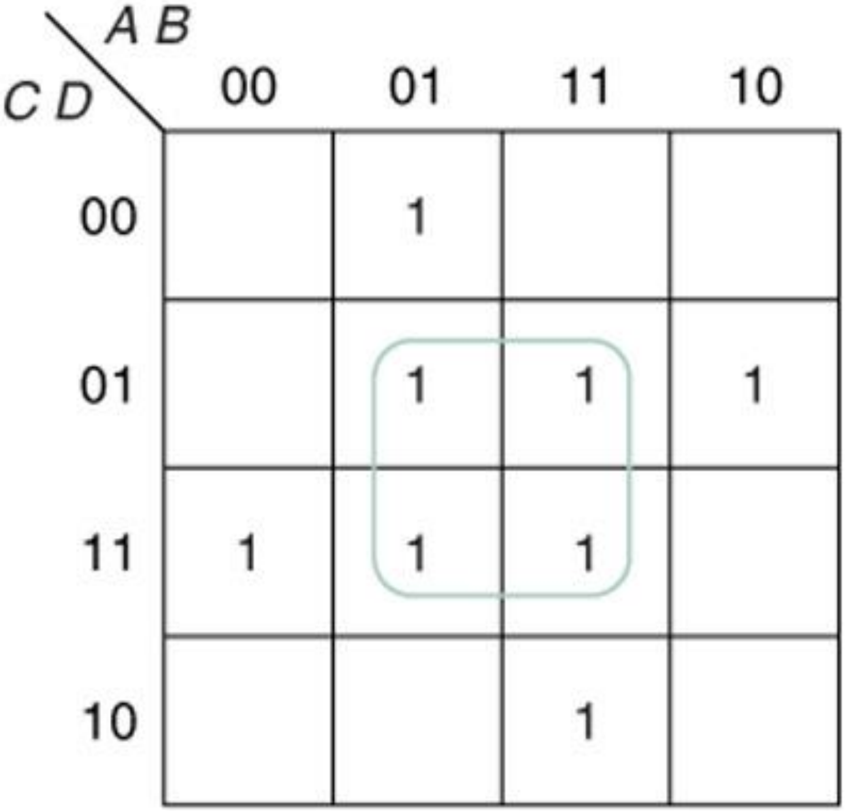


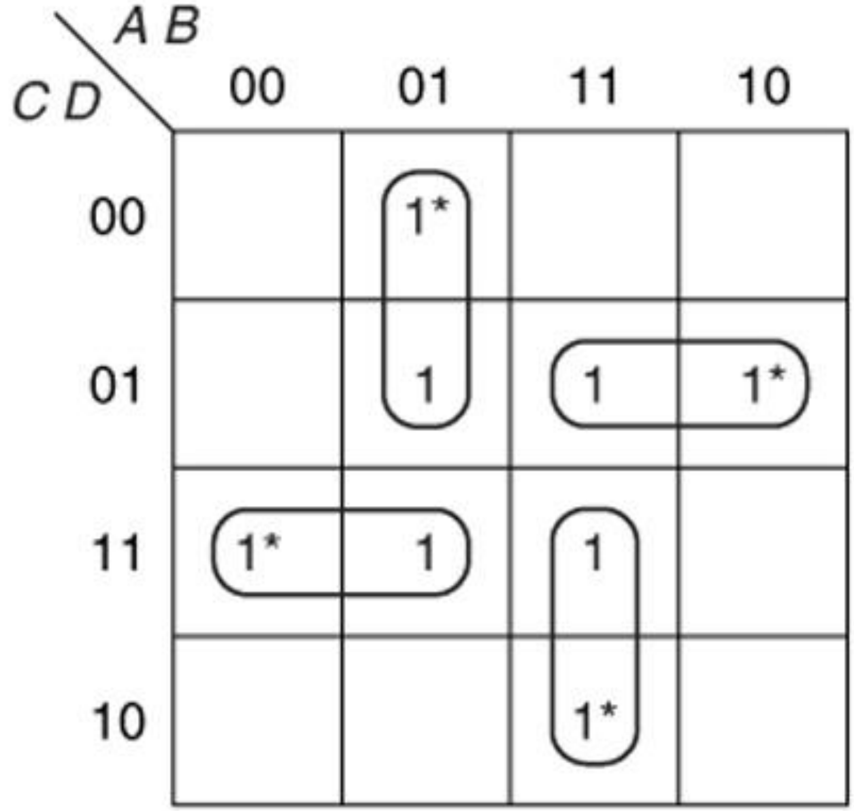
$$g = xz + wz + w'yz' + wx'y$$

$$g = xz + wz + w'yz' + x'yz'$$

$$g = xz + wz + x'yz' + w'xy$$

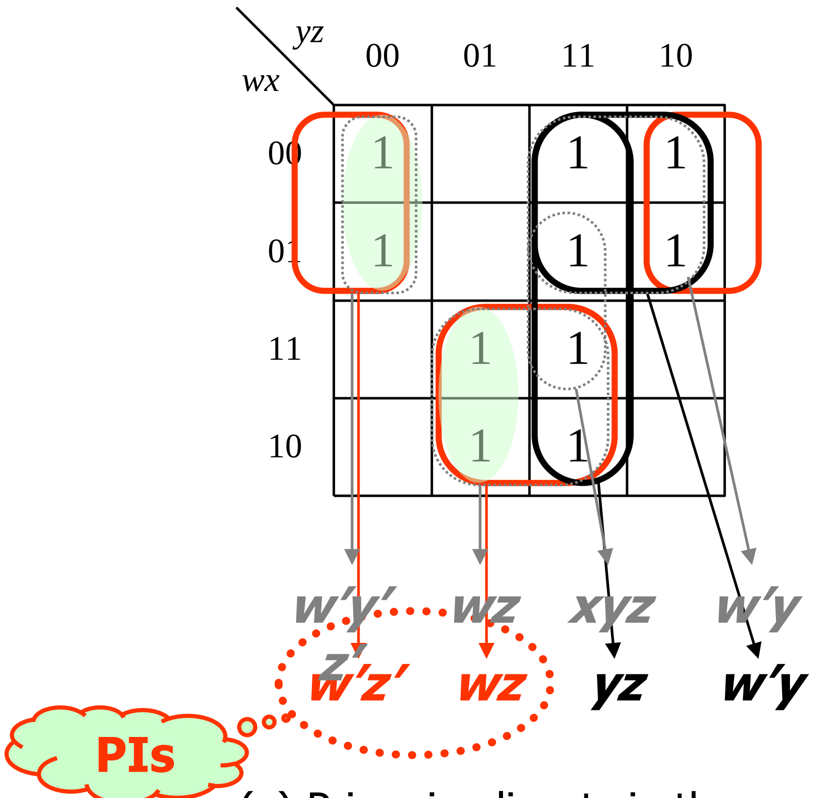
$G = A'BC' + A'CD + ABC + AC'D$





K-map example

$$F = w'y'z' + wz + xyz + w'y.$$



(a) Prime implicants in the map

PI list: $w'z'$, wz, yz, $w'y$

EPI list: $w'z'$, wz

Cover lists: (1) $w'z'$, wz, yz

(2) $w'z'$, wz, $w'y$

(b) PI, EPI, and cover lists

(1) $F = w'z' + wz + yz$

(2) $F = w'z' + wz + w'y$

(c) Two functional expressions

A Five Variable Map

- A five-variable map consists of $2^5 = 32$ squares

A = 0

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

A = 1

DE \ *BC*	00	01	11	10
00	16	20	28	24
01	17	21	29	25
11	19	23	31	27
10	18	22	30	26

K-map with Don’t Cares

- **Sometimes a function table or map contains entries for which it is known:**
  - **the input values for the minterm will never occur, or**
  - **The output value for the minterm is not used**
- **In these cases, the output value need not be defined**
- **Instead, the output value is defined as a “don't care”**
- **By placing “don't cares” (an “x” entry) in the function table or map, the cost of the logic circuit may be lowered.**

- **Example 1: A logic function having the binary codes for the BCD digits as its inputs. Only the codes for 0 through 9 are used. The six codes, 1010 through 1111 <u>never occur</u>, so the output values for these codes are “x” to represent “don’t cares.”**

K-map with Don't Cares

- Minterm expansion for incompletely specified function

$$F = \sum m(0,3,7) + \sum d(1,6)$$

Don't Cares

- Maxterm expansion for incompletely specified function

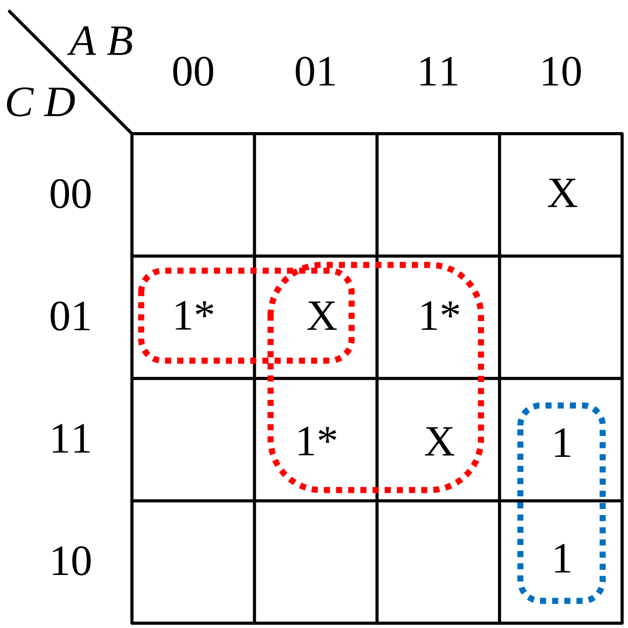
$$F = \prod M(2,4,5) \prod D(1,6)$$

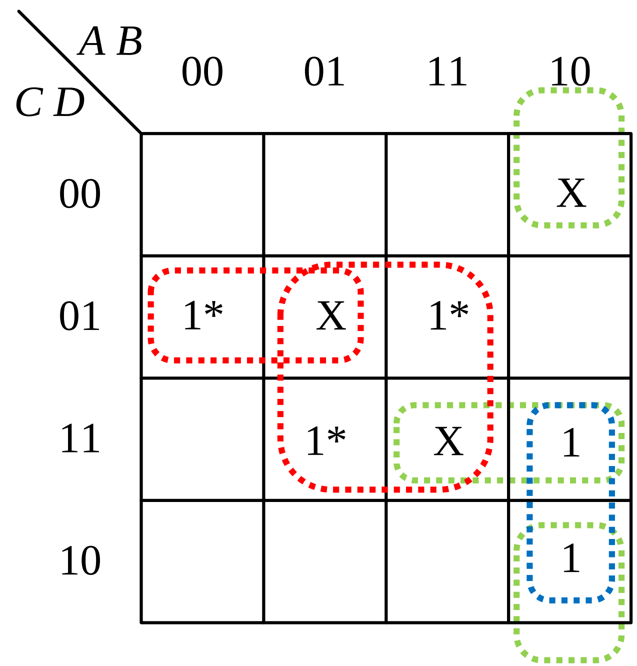
F(A,B,C,D) = ∑m(1,7,10,11,13) + ∑d(5,8,15)

Minimum Solution

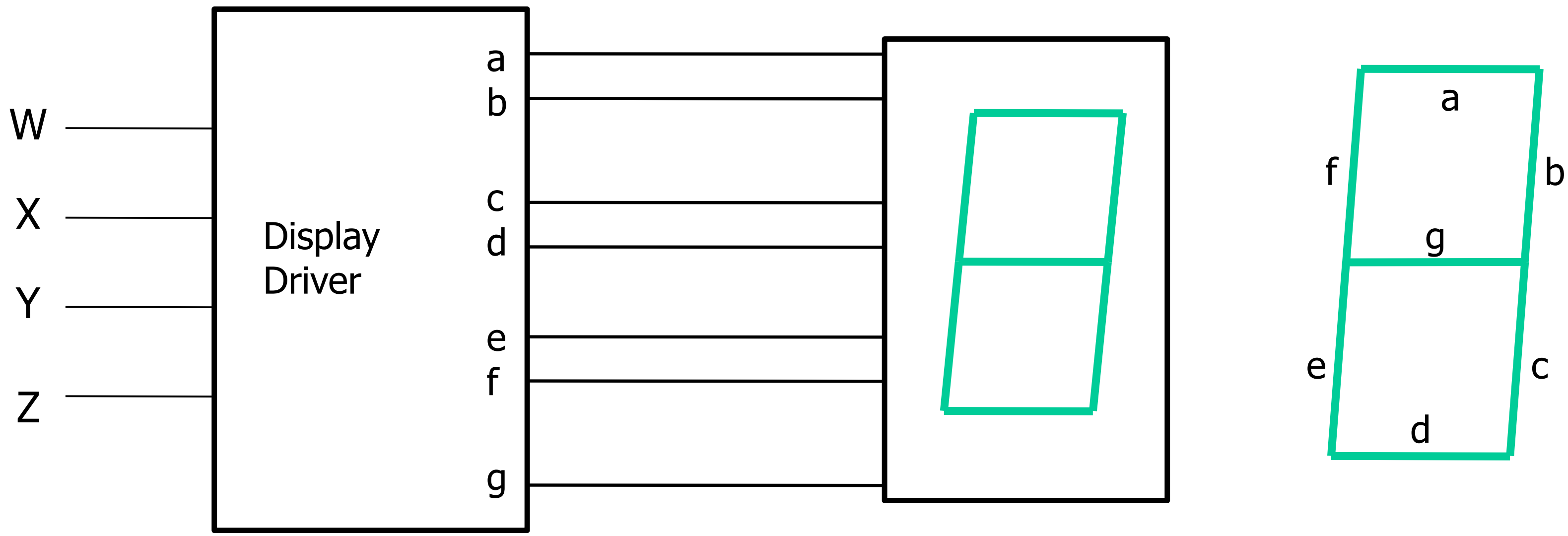
F = BD + A′C′D + AB′C

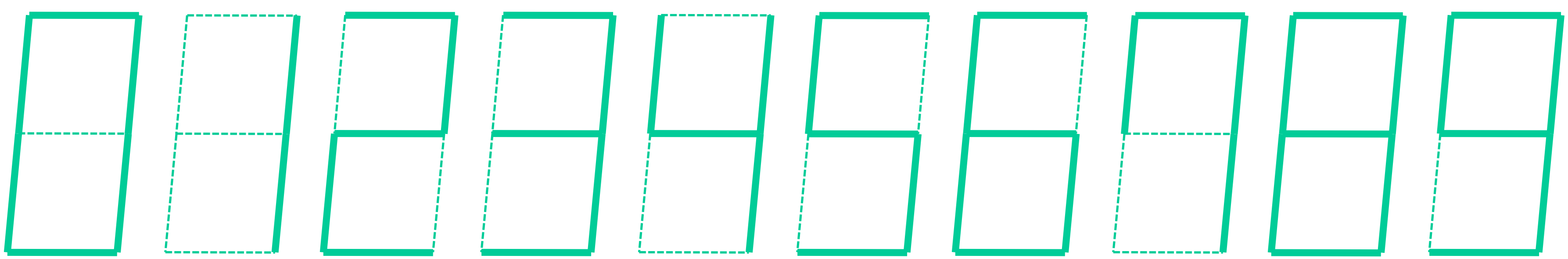
F = BD + A′C′D + (AB′D′ or ACD) are not used in the minimum solution





A seven-segment display





Truth table of seven segment display

Digit	W	X	Y	Z	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	X	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	X	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	X	0	1	1
-	1	0	1	0	X	X	X	X	X	X	X
-	1	0	1	1	X	X	X	X	X	X	X
-	1	1	0	0	X	X	X	X	X	X	X
-	1	1	0	1	X	X	X	X	X	X	X
-	1	1	1	0	X	X	X	X	X	X	X
-	1	1	1	1	X	X	X	X	X	X	X