COE 202: Digital Logic Design
Combinational Logic
Part 3

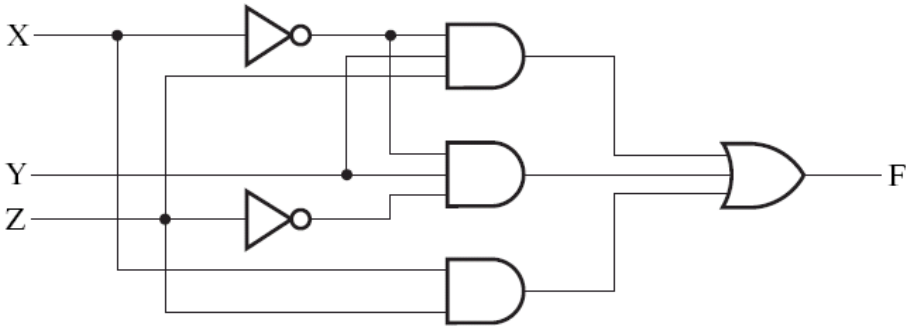
Dr. Ahmad Almulhem
Email: ahmadsm AT kfupm
Phone: 860-7554
Office: 22-324

Objectives

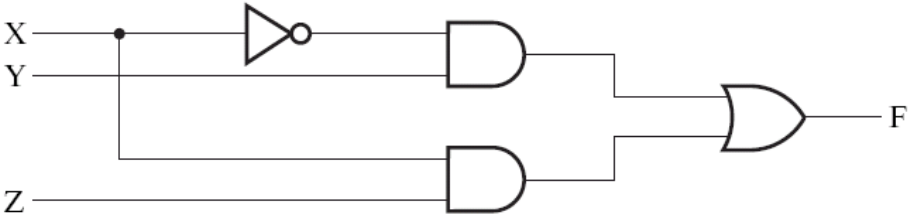
- Karnaugh Maps (K-Maps)
- Learn to minimize a function using K-Maps
  - 2-Variables
  - 3-Variables
  - 4-Variables
- Don’t care conditions
- Important Definitions
- 5-Variables K-Maps

Simplification using Algebra

F = X'YZ + X'YZ' + XZ
= X'Y(Z+Z') + XZ (id 14)
= X'Y.1 + XZ (id 7)
= X'Y + XZ (id 2)



(a) $F = \overline{X}YZ + \overline{X}Y\overline{Z} + XZ$



(b) $F = \overline{X}Y + XZ$

- Simplification may mean different things
- here it means less number of literals

Simplification Revisited

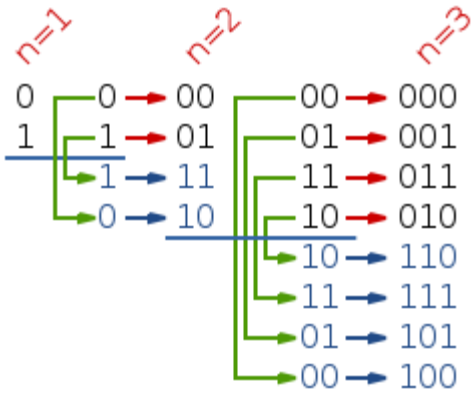
- Algebraic methods for minimization is limited:
  - No formal steps (id 10 first, then id 4, etc?), need experience.
  - No guarantee that a minimum is reached
  - Easy to make mistakes
- Karnaugh maps (k-maps) is an alternative convenient way for minimization:
  - A graphical technique
  - Introduced by Maurice Karnaugh in 1953
- K-maps for up to 4 variables are straightforward to build
- Building higher order K-maps (5 or 6 variable) are a bit more cumbersome
- Simplified expression produced by K-maps are in SOP or POS forms

Gray Codes (review)

•Only one bit changes with each number increment

•Build using recursive reflection

•To translate a binary value into the corresponding Gray code, each bit is inverted if the next higher bit of the input value is set to one.



src: wikipedia.org

Truth Table Adjacencies

A	B	F
0	0	1
0	1	1
1	0	0
1	1	0

These minterms are adjacent in a gray code sense – they differ by only one bit.

We can apply XY+XY'=X

F = A'B' + A'B = A'(B'+B) = A' (1)
= A'

A	B	F
0	0	0
0	1	1
1	0	0
1	1	1

Same idea:

F = A'B + AB = B

Keep common literal only!

K-Map

A	B	F
0	0	0
0	1	1
1	0	0
1	1	1

A different way to draw a truth table !

Take advantage of adjacency

A \ B	0	1
0	0 A' B'	1 A' B
1	0 A B'	1 A B

F = A'B + AB = B

Keep common literal only!

Minimization with K-maps

1. Draw a K-map
2. Combine maximum number of 1's following rules:
   1. Only adjacent squares can be combined
   2. All 1's must be covered
   3. Covering rectangles must be of size 1,2,4,8, … 2^n
3. Check if all covering are really needed
4. Read off the SOP expression

2-variable K-map

Given a function with 2 variables: F(X,Y), the total number of minterms are equal to 4:

m_0, m_1, m_2, m_3

The size of the k-map is always equal to the total number of minterms.

X \ Y	0	1
0	m_0	m_1
1	m_2	m_3

Each entry of the k-map corresponds to one minterm for the function:

X \ Y	0	1
0	0	1
1	2	3

Row 0 represents: X'Y', X'Y

Row 1 represents: XY', XY'

Example 1

For a given function F(X,Y) with the following truth table, minimize it using k-maps

X	Y	F
0	0	0
0	1	0
1	0	1
1	1	1

X \ Y	0	1
0	0	0
1	1	1

Combining all the 1's in only the adjacent squares

The final reduced expression is given by the common literals from the combination:

Therefore, since for the combination, Y has different values (0, 1), and X has a fixed value of 1,

The reduced function is: F(X,Y) = X

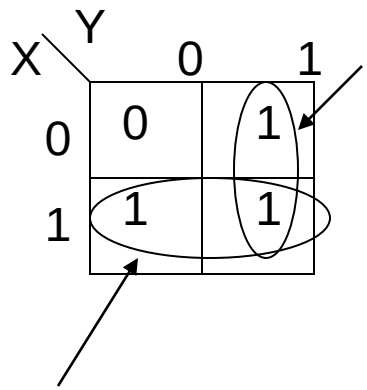
Example 2

Q. Simplify the function $F(X,Y) = \sum m(1,2,3)$

Sol. This function has 2 variables, and three 1-squares (three minterms where function is 1)

$F = m_1 + m_2 + m_3$

Note: The 1-squares can be combined more than once



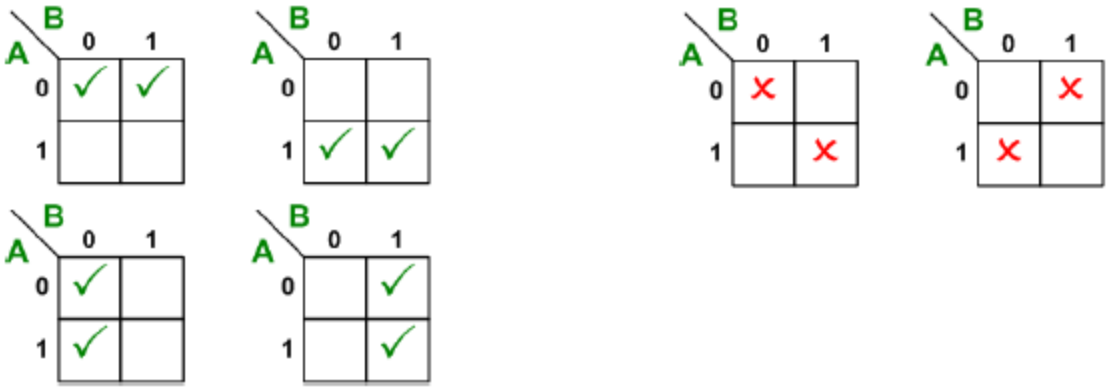
Y is the common literal in the adjacent 1-squares

X is the common literal

Minimized expression: F = X + Y

2 variable K-Maps (Adjacency)

In an n-variable k-map, each square is adjacent to exactly *n* other squares



Q: What if you have 1 in all squares?

3-variable K-maps

For 3-variable functions, the k-maps are larger and look different.

Total number of minterms that need to be accommodated in the k-map = 8

To maintain adjacency neighbors don't have more than 1 different bit

A \ BC	00	01	11	10
0	m_0	m_1	m_3	m_2
1	m_4	m_5	m_7	m_6

B

A

C

3-variable K-maps

A \ BC	00	01	11	10
0	m_0	m_1	m_3	m_2
1	m_4	m_5	m_7	m_6

Note: You can only combine a power of 2 adjacent 1-squares. For e.g. 2, 4, 8, 16 squares. You cannot combine 3, 7 or 5 squares

A \ BC	00	01	11	10
0	m_0	m_1	m_3	m_2
1	m_4	m_5	m_7	m_6

Minterms m_o, m_2, m_4, m_6 can be combined as m_0 and m_2 are adjacent to each other, m_4 and m_6 are adjacent to each other

m_o and m_4 are also adjacent to each other, m_2 and m_6 are also adjacent to each other

Example 1

Simplify F = ∑m(1, 3, 4, 6) using K-map

A \ BC	00	01	11	10
0	0	1 **1**	3 **1**	2
1	4 **1**	5	7	6 **1**

B

A

C

Example 1

Simplify F = ∑m(1, 3, 4, 6) using K-map

F = A'C + AC'

A \ BC	00	01	11	10
0	0	1 **1**	3 **1**	2
1	4 **1**	5	7	6 **1**

B

A

C

Example 2

Simplify F = ∑m(0,1, 2, 4, 6) using K-map

A \ BC	00	01	11	10
0	0: **1**	1: **1**	3:	2: **1**
1	4: **1**	5:	7:	6: **1**

B

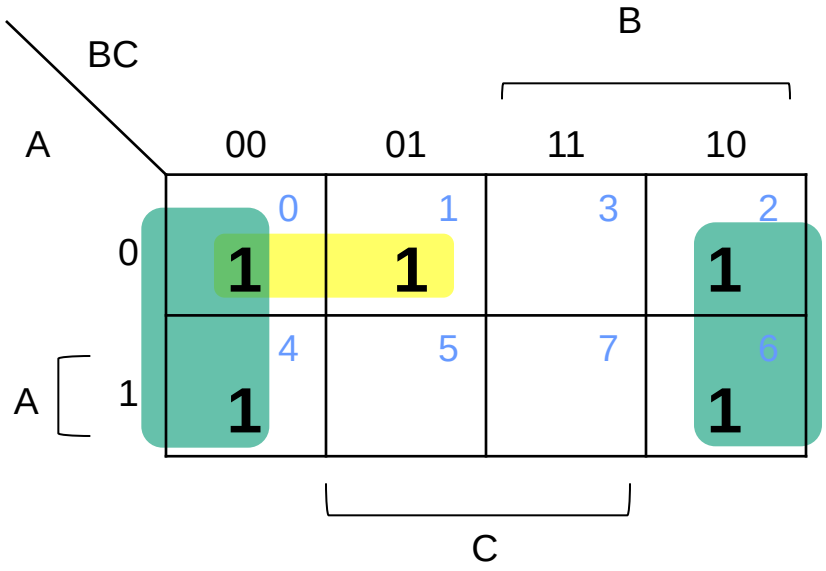
A

C

Example 2

Simplify F = ∑m(0,1, 2, 4, 6) using K-map

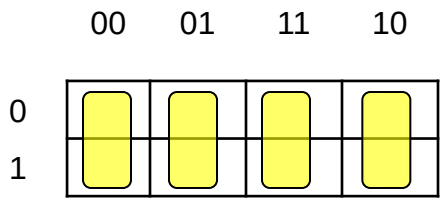
F = A’ B’ + C’

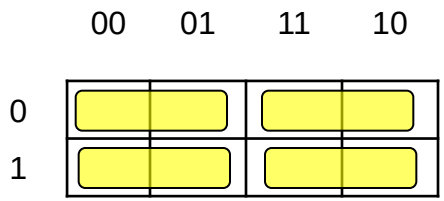


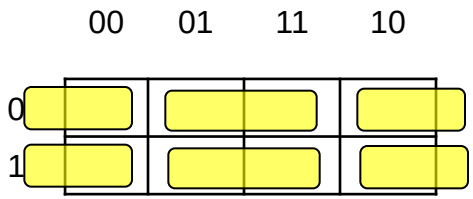
3 variable K-Maps (Adjacency)

A 3-variable map has 12 possible groups of 2 minterms

They become product terms with 2 literals



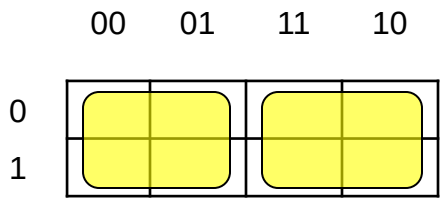


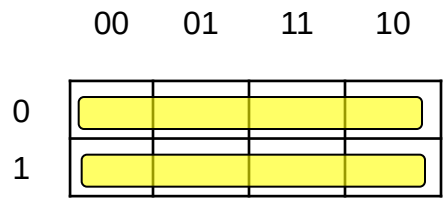


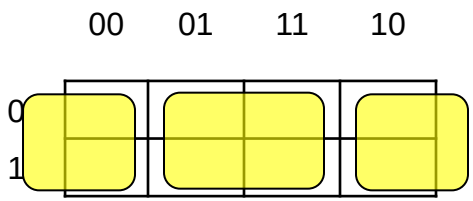
3 variable K-Maps (Adjacency)

A 3-variable map has 6 possible groups of 4 minterms

They become product terms with 1 literals







4-variable K-maps

A 4-variable function will consist of 16 minterms and therefore a size 16 k-map is needed

Each square is adjacent to 4 other squares

A square by itself will represent a minterm with 4 literals

Combining 2 squares will generate a 3-literal output

Combining 4 squares will generate a 2-literal output

Combining 8 squares will generate a 1-literal output

AB \ CD	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}

C

B

A

D

4-variable K-maps (Adjacency)

AB \ CD	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}

Note: You can only combine a power of 2 adjacent 1-squares. For e.g. 2, 4, 8, 16 squares. You cannot combine 3, 7 or 5 squares

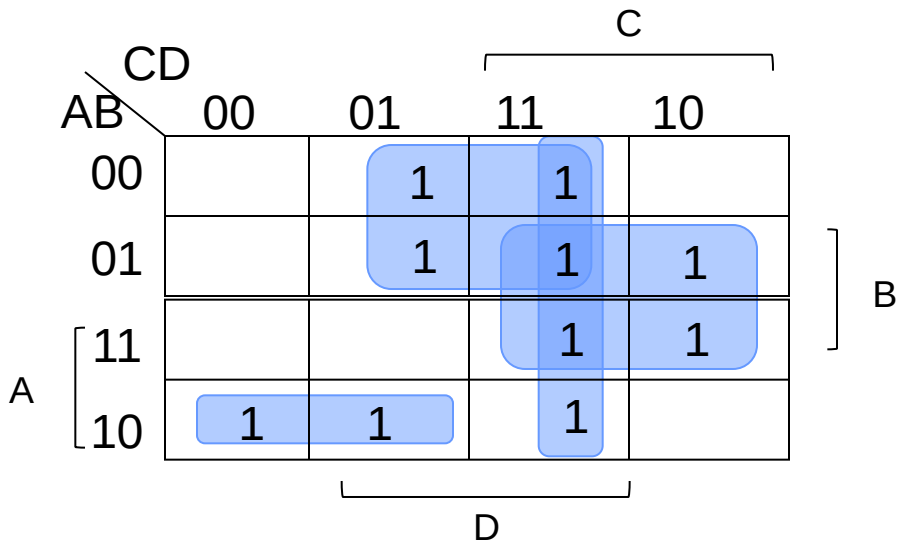
Right column and left column are adjacent; can be combined

Top row and bottom column are adjacent; can be combined

Many possible 2, 4, 8 groupings

Example

Minimize the function F(A,B,C,D)=∑m(1,3,5,6,7,8,9,11,14,15)



F = CD + A'D + BC + AB'C'

Example

F(A,B,C,D) = Σm(0,1,2,5,8,9,10)

AB \ CD	00	01	11	10
00	1	1		1
01		1		
11				
10	1	1		1

C=1

B=1

A=1

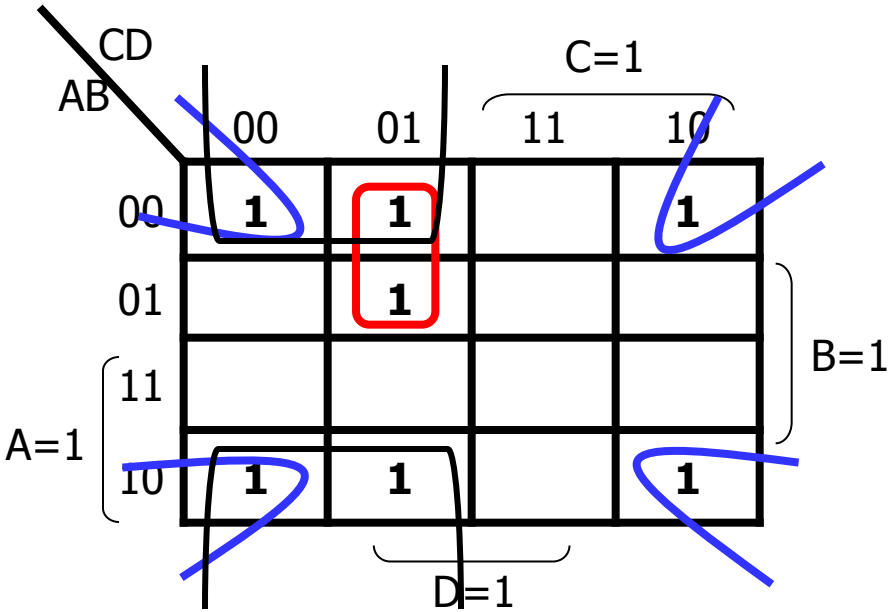
D=1

Example

F(A,B,C,D) = Σm(0,1,2,5,8,9,10)

Solution:

F = B’ D’ + B’ C’ + A’ C’ D



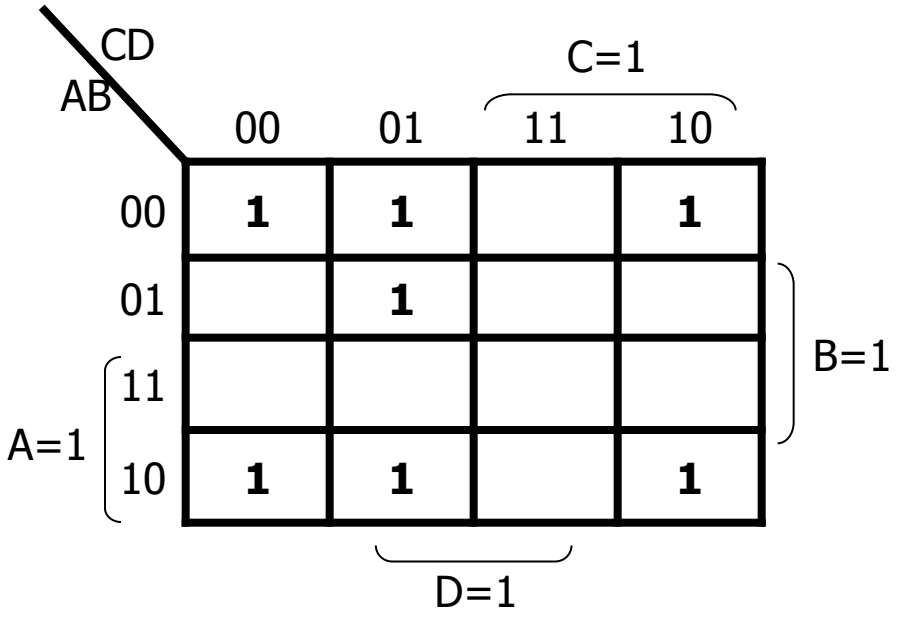
Example (POS)

F(A,B,C,D) = Σm(0,1,2,5,8,9,10)

Write F in the simplified product of sums (POS)

Two methods?

You already know one!



Example (POS)

F(A,B,C,D) = Σm(0,1,2,5,8,9,10)

Write F in the simplified product of sums (POS)

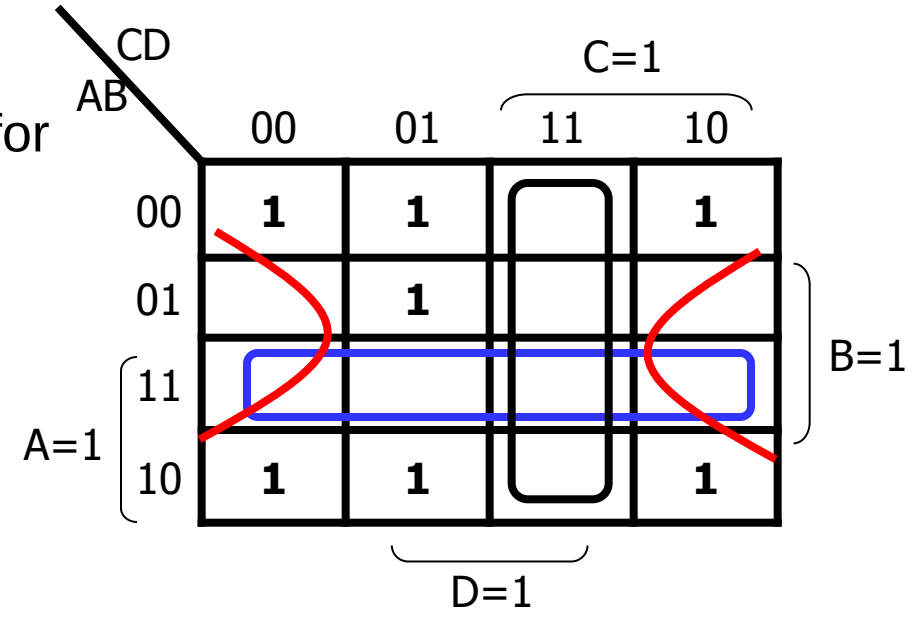
Method 2:

Follow same rule as before but for the ZEROs

F' = AB + CD + BD'

Therefore,

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



Don’t Cares

- In some cases, the output of the function (1 or 0) is not specified for certain input combinations either because
  - The input combination never occurs (Example BCD codes), or
  - We don’t care about the output of this particular combination
- Such functions are called incompletely specified functions
- Unspecified minterms for these functions are called don’t cares
- While minimizing a k-map with don’t care minterms, their values can be selected to be either 1 or 0 depending on what is needed for achieving a minimized output.

Example

$$F = \sum m(1, 3, 7) + \sum d(0, 5)$$

Circle the x's that help get bigger groups of 1's (or 0's if POS).

Don't circle the x's that don't help.

A \ BC	00	01	11	10
0	0: X	1: 1	3: 1	2
1	4	5: X	7: 1	6

B

A

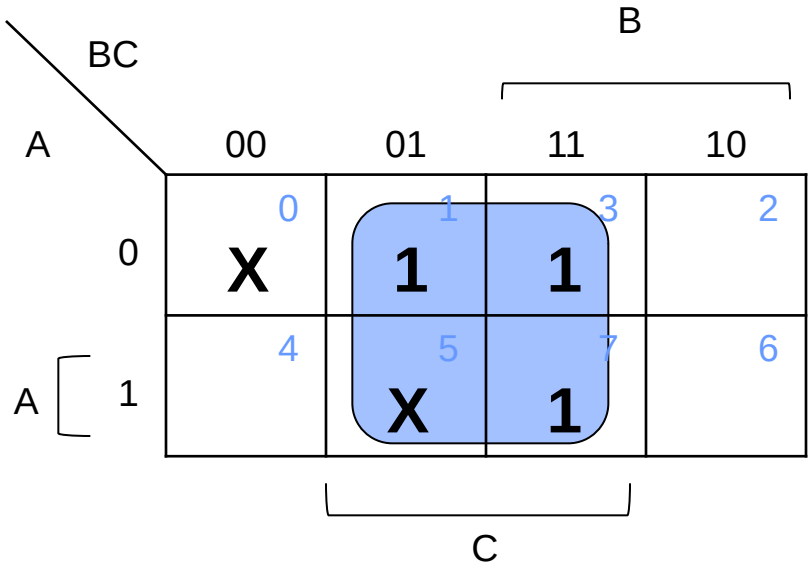
C

Example

$$F = \sum m(1, 3, 7) + \sum d(0, 5)$$

Circle the x's that help get bigger groups of 1's (or 0's if POS).

Don't circle the x's that don't help.



F = C

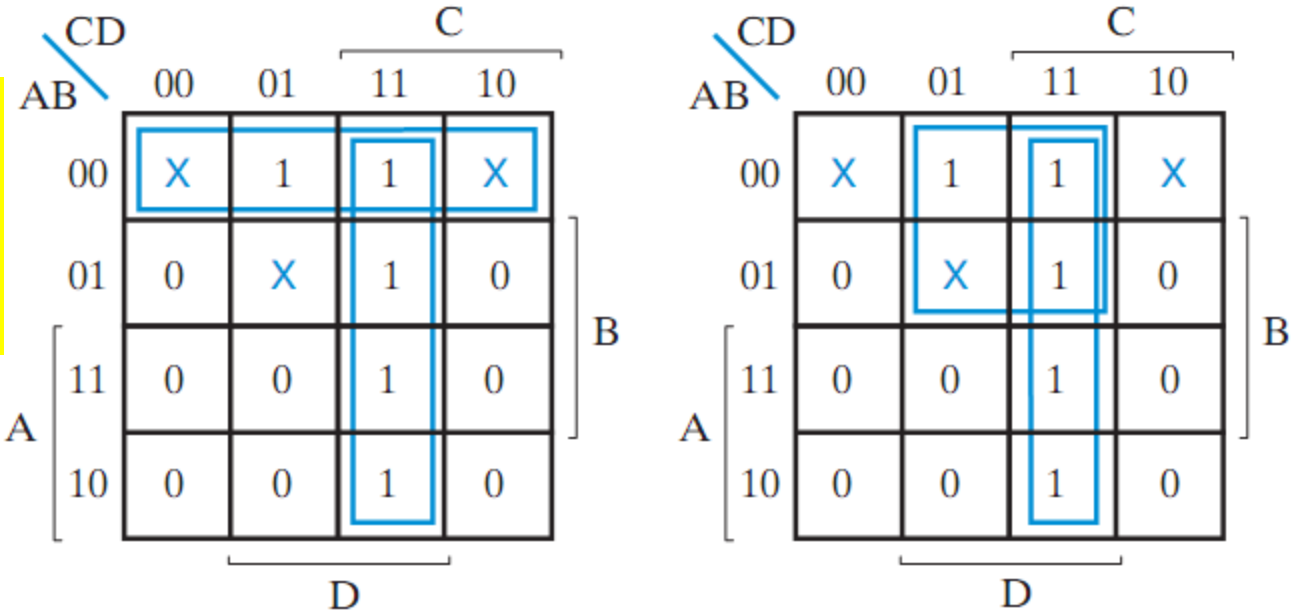
Example 2

$$F(A, B, C, D) = \sum m(1, 3, 7, 11, 15) + \sum d(0, 2, 5)$$

-Two possible solutions!

-Both acceptable.

-All 1's covered



(a) $F = CD + \overline{A}\,\overline{B}$ (b) $F = CD + \overline{A}D$

Src: Mano's Textbook

Definitions

- An **implicant** is a product term of a function
  - Any group of 1's in a K-Map
- A **prime implicant** is a product term obtained by combining the maximum possible number of adjacent 1's in a k-map
  - Biggest groups of 1's
  - Not all prime implicants are needed!
- If a minterm is covered by exactly one prime implicant then this prime implicant is called an **essential prime implicant**

Example

Consider F(X,Y,Z) = Σm(1,3,4,5,6)

List all implicants, prime implicants and essential prime implicants

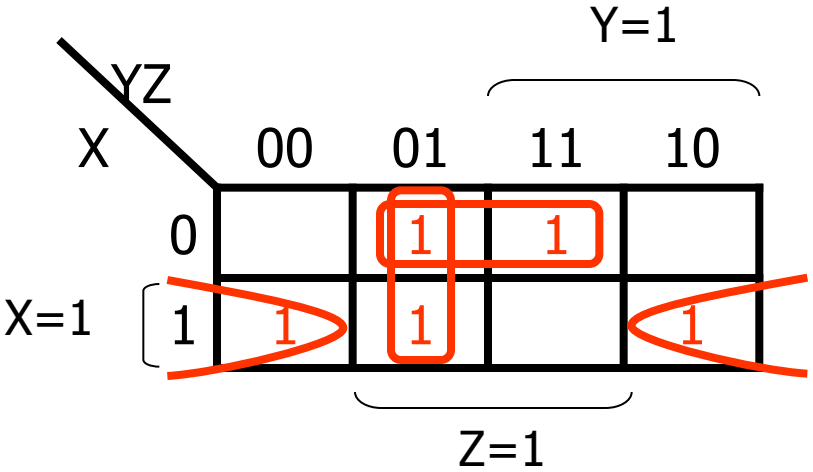
Solution:

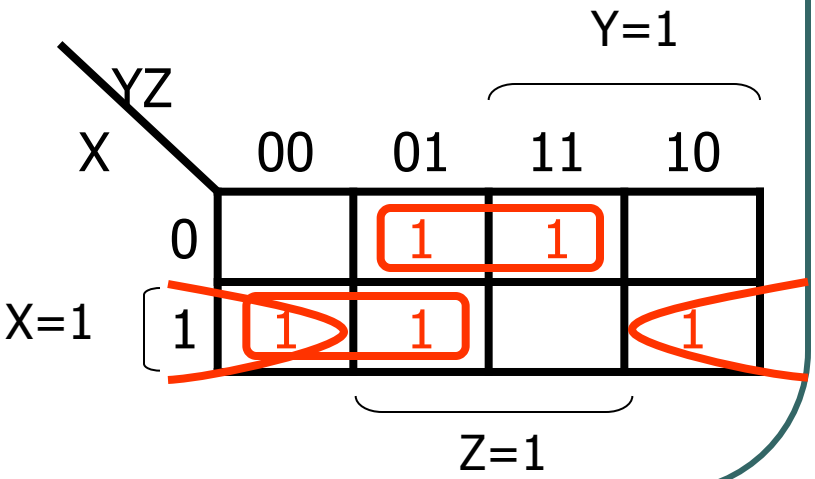
Implicants: XY'Z', XZ', XY', XY'Z, X'Y'Z, Y'Z, …

P.Is: XY', XZ', Y'Z, X'Z

EPIs: X'Z, XZ'

The simplest expression is NOT unique!





Finding minimum SOP

1. Find each essential prime implicant and include it in the solution
2. If any minterms are not yet covered, find minimum number of prime implicants to cover them (minimize overlap).

Example 2

Simplify F(A, B, C, D) = ∑ m(0, 1, 2, 4, 5, 10,11,13, 15)

Note:

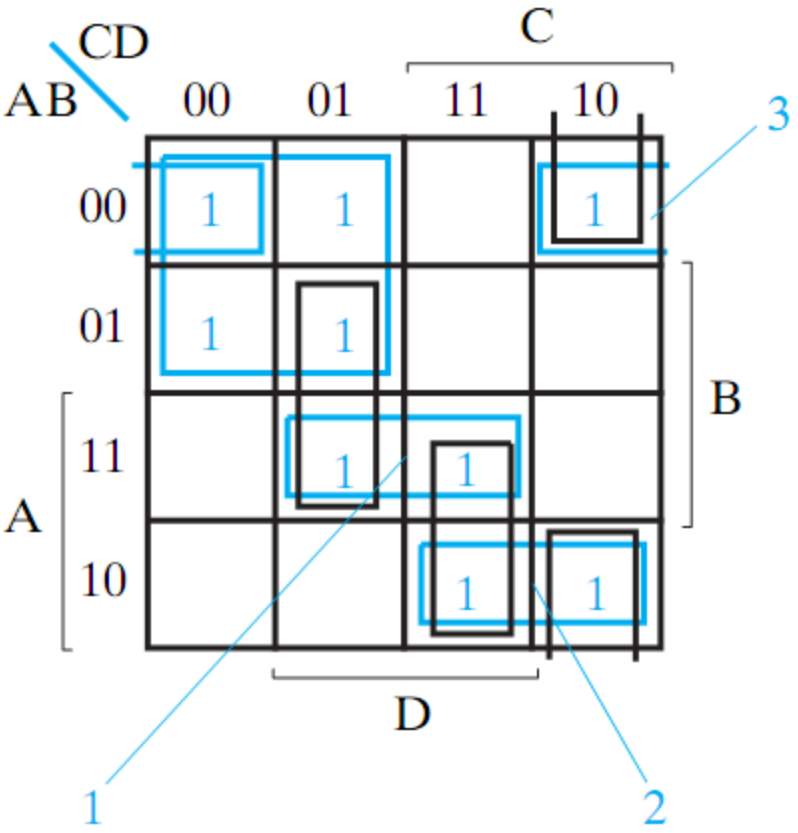
-Only A'C' is E.P.I

-For the remaining minterms:

-Choose 1 and 2 (minimize overlap)

-For m_2, choose either A'B'D' or B'CD'

F = A'C' + ABD + AB'C + A'B'D'



Src: Mano's Textbook

5-variable K-maps

A=0

BC \ DE	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}

A=1

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

- 32 minterms require 32 squares in the k-map
- Minterms 0-15 belong to the squares with variable A=0, and minterms 16-32 belong to the squares with variable A=1
- Each square in A' is also adjacent to a square in A (one is above the other)
- Minterm 4 is adjacent to 20, and minterm 15 is to 31

Conclusion

- A K-Map is simply a folded truth table, where physical adjacency implies logical adjacency
- K-Maps are most commonly used hand method for logic minimization.