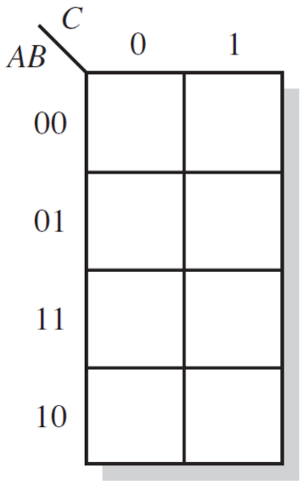
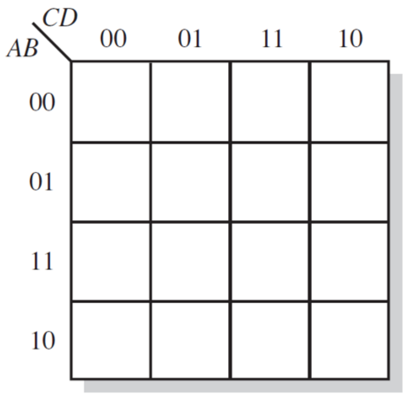
The Karnaugh map

- The Karnaugh map provides a cookbook method for simplification.
- The idea of Karnaugh map is to use adjacency to simplify the expression
- The Karnaugh map method is limited to 4 variables or below



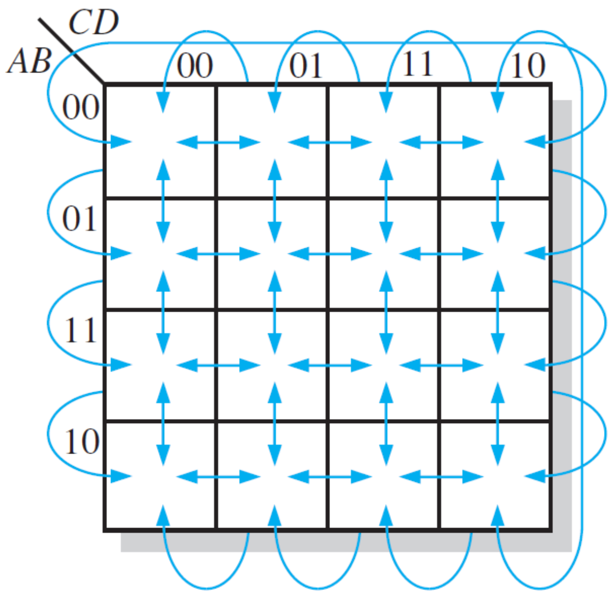
3-variable Karnaugh map



4-variable Karnaugh map

Question: can we use the order 00,10,11,01?

Cell adjacency



- Physically, each cell is adjacent to the cells that are immediately next to it on any of its four sides.
- Cells in top row are adjacent to cells in bottom row
- Cells in left column are adjacent to cells in right column.

Standard SOP

- All Boolean expressions can be converted into either sum-of-product (SOP) form or product-of-sum (POS) form.
- In SOP, a single overbar cannot extend over more than one variable.
- Standard SOP is one in which all the variables appear in each term

Question: Are they standard SOP expression?

AB+ABC

ABC+CDE+B’CD’

A’B+A’BC’+AC

AB’CD+A’B’CD’+ABC’D’

AB+A’B+AB’+A’B’

Convert to standard SOP

1. Multiply each nonstandard product term by the sum of a missing variable and its complement since *A* + *A'* = 1.
2. Repeat step 1 until all terms are in standard form

$$A\overline{B}C + \overline{A}\,\overline{B} + AB\overline{C}D$$

Question: Convert *A*+*B'C* to standard SOP

1 $A\overline{B}C = A\overline{B}C(D + \overline{D}) = A\overline{B}CD + A\overline{B}C\overline{D}$

2 $\overline{A}\,\overline{B} = \overline{A}\,\overline{B}(C + \overline{C}) = \overline{A}\,\overline{B}C + \overline{A}\,\overline{B}\,\overline{C}$

3 $$\begin{aligned}\overline{A}\,\overline{B} &= \overline{A}\,\overline{B}C + \overline{A}\,\overline{B}\,\overline{C} = \overline{A}\,\overline{B}C(D + \overline{D}) + \overline{A}\,\overline{B}\,\overline{C}(D + \overline{D}) \\ &= \overline{A}\,\overline{B}CD + \overline{A}\,\overline{B}C\overline{D} + \overline{A}\,\overline{B}\,\overline{C}D + \overline{A}\,\overline{B}\,\overline{C}\,\overline{D}\end{aligned}$$

$$A\overline{B}C + \overline{A}\,\overline{B} + AB\overline{C}D = A\overline{B}CD + A\overline{B}C\overline{D} + \overline{A}\,\overline{B}CD + \overline{A}\,\overline{B}C\overline{D} + \overline{A}\,\overline{B}\,\overline{C}D + \overline{A}\,\overline{B}\,\overline{C}\,\overline{D} + AB\overline{C}D$$

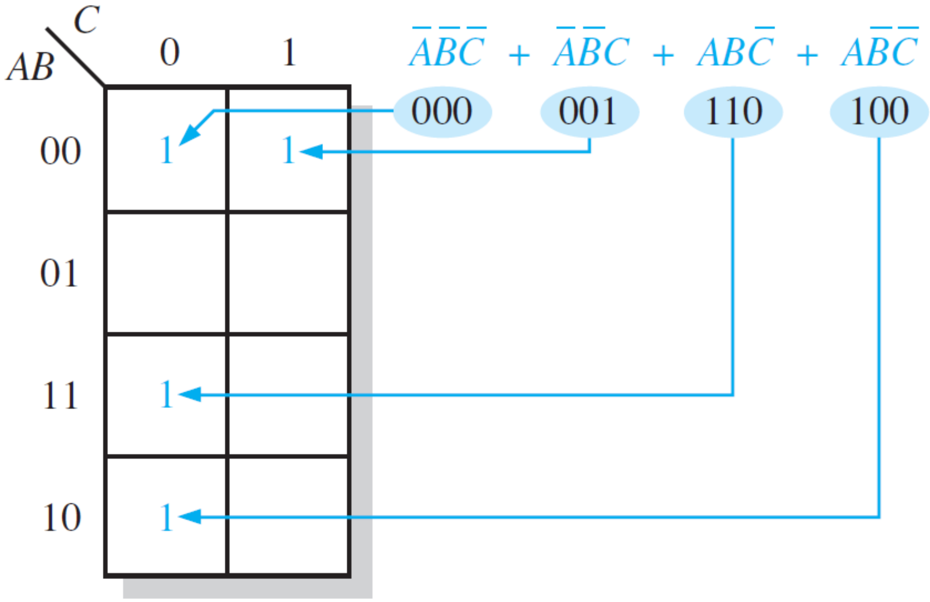
- Binary representation of a standard product term

$$A\overline{B}C\overline{D} = 1 \cdot \overline{0} \cdot 1 \cdot \overline{0} = 1 \cdot 1 \cdot 1 \cdot 1 = 1$$

The product term has a binary value of 1010.

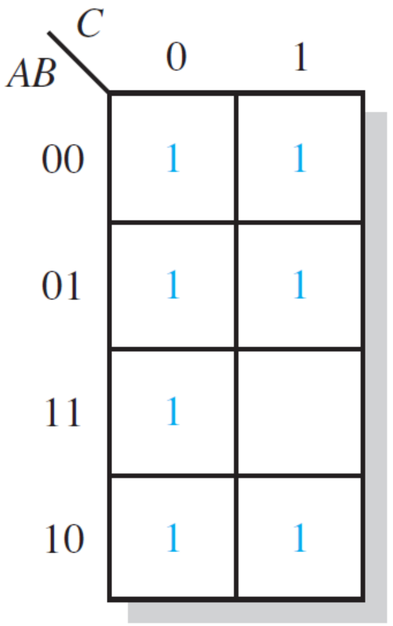
Mapping SOP expression to Karnaugh Map

- A 1 is placed for each product term



Standard SOP

$\overline{A} + A\overline{B} + AB\overline{C}$



Nonstandard SOP

Karnaugh Map SOP Minimization

Grouping the 1s

- A group must contain powers of two cells.
- Each 1 on the map must be included in at least one group.
- The 1s already in a group can be included in another group.
- Maximize the size of the groups.

AB \ C	1	0
00	1	
01		1
11	1	1
10		

AB \ C	1	0
00	1	1
01	1	
11		1
10	1	1

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

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

Karnaugh Map SOP Minimization

- Each group creates one product term. Variables that occur both uncomplemented and complemented within the group are eliminated.
- Determine the product term, e.g., for a 4-variable map:
    - (1) A 1-cell group yields a 4-variable product term; (2) A 2-cell group yields a 3-variable product term; (3) A 4-cell group yields a 2-variable term; (4) An 8-cell group yields a 1-variable term.
- Sum all the minimum product terms to form the minimum SOP expression.

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

Karnaugh Map SOP Minimization

- $Y(A,B,C) = AC' + A'C + B'C + BC'$

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

- The final results are not unique. However, the number of terms and the number of variables in each term are identical.

Think about it:

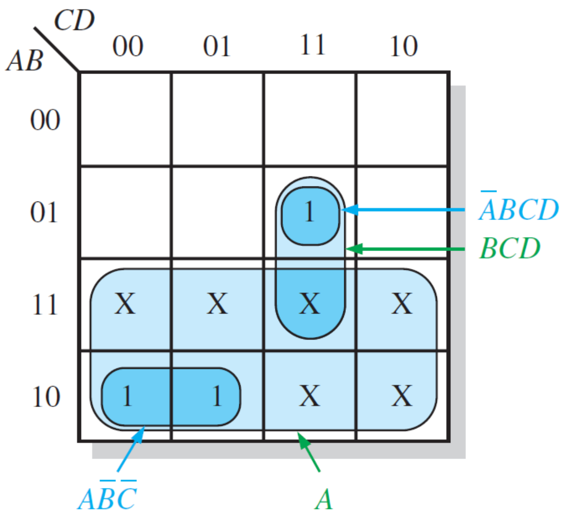
- If we have a 2 variables Karnaugh map, how many 1s in the map indicate that simplification is possible?
- How about 3 variables, 4 variables?

Don’t Care Conditions

- When some input variable combinations are not allowed, e.g., BCD code, they can be treated as “don’t care” terms (Xs) to assist the simplification.
- When grouping the 1s, the Xs can be treated as 1s to make a larger grouping or as 0s if they cannot be used to advantage.

Inputs				Output
A	*B*	*C*	*D*	*Y*
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	X
1	0	1	1	X
1	1	0	0	X
1	1	0	1	X
1	1	1	0	X
1	1	1	1	X

Don’t cares



Without “don’t cares” $Y = A\bar{B}\bar{C} + \bar{A}BCD$

With “don’t cares” $Y = A + BCD$

Alternative Way

- One can also group the 0s

AB \ CD	00	01	11	10
00	1	0	0	1
01	1	0	0	1
11	1	1	1	1
10	1	1	1	1

Standard POS

Question: Are they standard POS expression?

$$(\overline{A} + B)(A + \overline{B} + C)$$

$$(\overline{A} + \overline{B} + \overline{C})(C + \overline{D} + E)(\overline{B} + C + D)$$

$$(A + B)(A + \overline{B} + C)(\overline{A} + C)$$

$$(\overline{A} + \overline{B} + \overline{C} + \overline{D})(A + \overline{B} + C + D)(A + B + \overline{C} + D)$$

Convert to standard POS

1. Add to each nonstandard term the product of the missing variable and its complement since $A \cdot A' = 0$.
2. Repeat Step 1 until all terms are in standard form

$$(A+\overline{B}+C)(\overline{B}+C+\overline{D})(A+\overline{B}+\overline{C}+D)$$

Question: Why they are equivalent?

1 $$A + \overline{B} + C = A + \overline{B} + C + D\overline{D} = (A + \overline{B} + C + D)(A + \overline{B} + C + \overline{D})$$

2 $$\overline{B} + C + \overline{D} = \overline{B} + C + \overline{D} + A\overline{A} = (A + \overline{B} + C + \overline{D})(\overline{A} + \overline{B} + C + \overline{D})$$

$$(A + \overline{B} + C)(\overline{B} + C + \overline{D})(A + \overline{B} + \overline{C} + D) = (A + \overline{B} + C + D)(A + \overline{B} + C + \overline{D})(A + \overline{B} + C + \overline{D})(\overline{A} + \overline{B} + C + \overline{D})(A + \overline{B} + \overline{C} + D)$$

- Binary representation of a standard sum term

Question: Convert $A(B'+C)$ to standard SOP

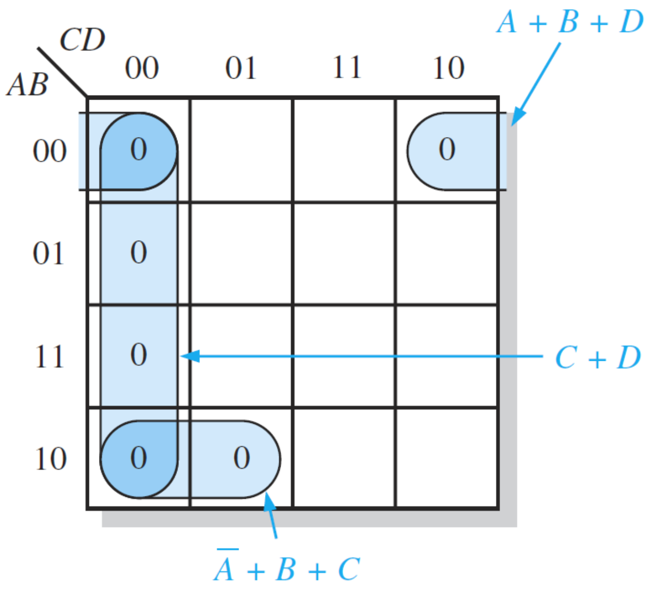
$$A + \overline{B} + C + \overline{D} = 0 + \overline{1} + 0 + \overline{1} = 0 + 0 + 0 + 0 = 0$$

The sum term has a binary value of 0101.

Karnaugh Map POS Minimization

$$(B + C + D)(A + B + \overline{C} + D)(\overline{A} + B + C + \overline{D})(A + \overline{B} + C + D)(\overline{A} + \overline{B} + C + D)$$

- A 0 is placed for each sum term.
- Group the 0s to get the minimum expression



Convert SOP to standard POS

- Step 1: determine the binary numbers that represent the product terms.
- Step 2: determine all of the binary numbers not included in the previous step.
- Step 3: write the equivalent sum term for each binary number from Step 2 and express in POS form.

Question: Convert *A*+*BC* into standard POS

$$\overline{A}\overline{B}\overline{C} + \overline{A}B\overline{C} + \overline{A}BC + A\overline{B}C + ABC$$

1 $000 + 010 + 011 + 101 + 111$

2 The remaining terms are 001, 100 and 110, these are the binary values that make the sum term 0.

3 The equivalent POS form is

$$(A + B + \overline{C})(\overline{A} + B + C)(\overline{A} + \overline{B} + C)$$

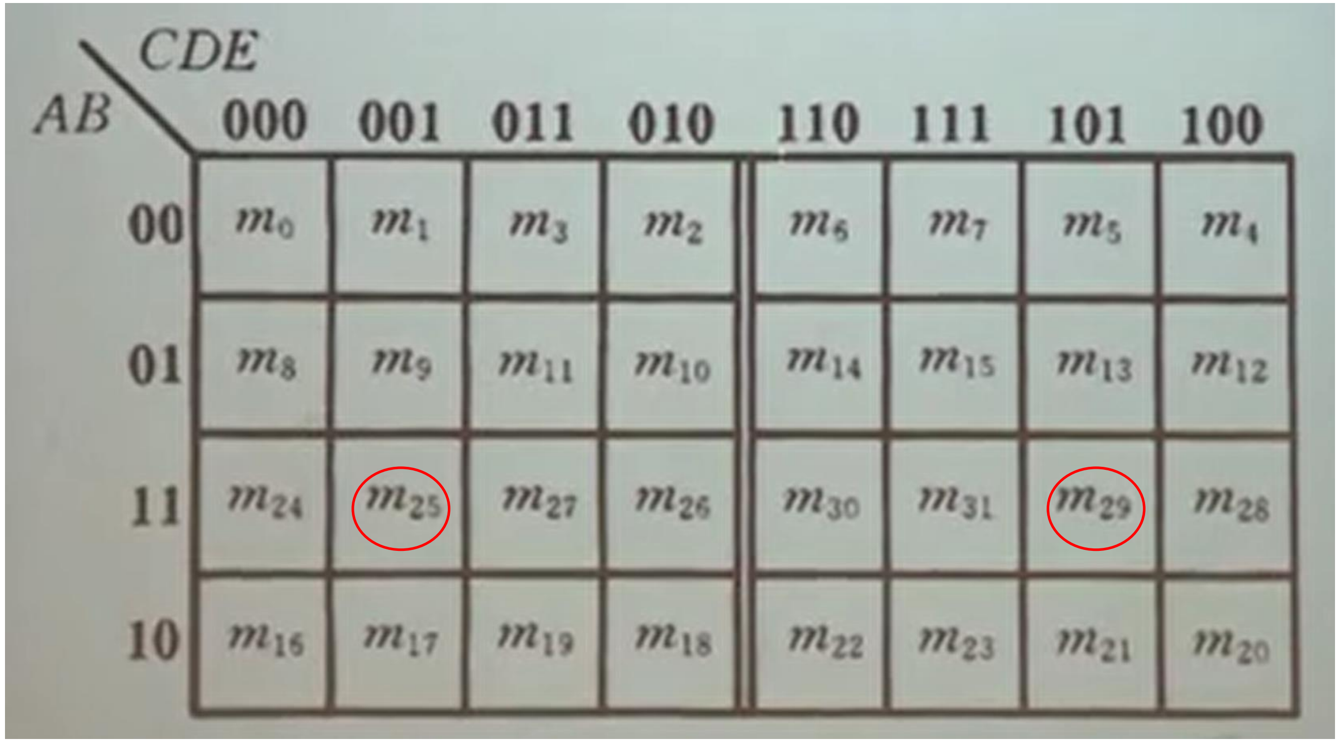
- Using a similar procedure, you can go from POS to SOP.

Compare POS and SOP

$$(\overline{A} + \overline{B} + C + D)(A + \overline{B} + C + D)(A + B + C + \overline{D})(A + B + \overline{C} + \overline{D})(\overline{A} + B + C + \overline{D})(A + B + \overline{C} + D)$$

- Get the minimum POS expression
- Get the minimum SOP expression
- Compare their gates usage

5 variables Karnaugh map



AB \ CDE	000	001	011	010	110	111	101	100
00	m_0	m_1	m_3	m_2	m_6	m_7	m_5	m_4
01	m_8	m_9	m_{11}	m_{10}	m_{14}	m_{15}	m_{13}	m_{12}
11	m_{24}	m_{25}	m_{27}	m_{26}	m_{30}	m_{31}	m_{29}	m_{28}
10	m_{16}	m_{17}	m_{19}	m_{18}	m_{22}	m_{23}	m_{21}	m_{20}

- It's difficult to visualize the adjacency

Appendix

Reading materials

- Chapter 4 of Floyd book
- Chapter 2 of 阎石 book

The Quine-McCluskey Method

- Karnaugh maps is practical only for up to four variables. It is not suitable to be automated in a computer program.
- The Quine-McCluskey method (or tabulation method) is more practical for more than four or five variables. Its tabular form makes it suitable to be used in computer algorithms. It also gives a way to check if a minimal form is reached.
- It applies the Boolean distributive law to various terms to find the minimum sum of products by eliminating literals that appear in two terms as complements. (e.g., ABCD + ABC$\overline{D}$ = ABC).

The Quine-McCluskey Method

$$X = \overline{A}\,\overline{B}\,\overline{C}D + \overline{A}\,\overline{B}CD + \overline{A}B\overline{C}\,\overline{D} + \overline{A}B\overline{C}D + A\overline{B}C\overline{D} + AB\overline{C}\,\overline{D} + AB\overline{C}D + ABCD$$

1. Write the SOP form and represent it as binary numbers on the truth table
2. Arrange the minterms in groups according to the number of 1s

$ABCD$	X	Minterm
0000	0	
0001	1	m_1
0010	0	
0011	1	m_3
0100	1	m_4
0101	1	m_5
0110	0	
0111	0	
1000	0	
1001	0	
1010	1	m_{10}
1011	0	
1100	1	m_{12}
1101	1	m_{13}
1110	0	
1111	1	m_{15}

Number of 1s	Minterm	$ABCD$
1	m_1	0001
	m_4	0100
2	m_3	0011
	m_5	0101
	m_{10}	1010
	m_{12}	1100
3	m_{13}	1101
4	m_{15}	1111

The Quine-McCluskey Method

3. Compare adjacent groups, looking for minterms that are the same in every position except one. Place a check mark. Check each minterm against all others in the following group, but it is not necessary to check any groups that are not adjacent. In the column labeled First level, one have the minterm names and the binary with an x for the literal that differs
4. The one does not have a check mark (essential prime implicant minterm, e.g., m10) must be included in the final expression

Number of 1s in Minterm	Minterm	ABCD	First Level
1	m_1	0001 ✓	(m_1, m_3) 00x1
	m_4	0100 ✓	(m_1, m_5) 0x01
2	m_3	0011 ✓	(m_4, m_5) 010x
	m_5	0101 ✓	(m_4, m_{12}) x100
	m_{10}	1010	(m_5, m_{13}) x101
	m_{12}	1100 ✓	(m_{12}, m_{13}) 110x
3	m_{13}	1101 ✓	(m_{13}, m_{15}) 11x1
4	m_{15}	1111 ✓	

The Quine-McCluskey Method

5. The First level are used to form a table with one less group. Terms in the new groups are compared against terms in the adjacent group down. Compare these terms only if the x is in the same relative position in adjacent groups. If the two expressions differ by exactly one position, a check mark is placed and listed in the Second level.
6. Read the Second level expression as $B\bar{C}$. The unchecked terms are $\bar{A}\bar{B}D$, $\bar{A}\bar{C}D$, ABD.

$$X = B\overline{C} + \overline{A}\,\overline{B}D + \overline{A}\,\overline{C}D + ABD + A\overline{B}C\overline{D}$$

First Level	Number of 1s in First Level	Second Level
(m_1, m_3) 00x1	1	$(m_4, m_5, m_{12}, m_{13})$ x10x
(m_1, m_5) 0x01		$(m_4, m_5, m_{12}, m_{13})$ x10x
(m_4, m_5) 010x ✓		
(m_4, m_{12}) x100 ✓		
(m_5, m_{13}) x101 ✓	2	
(m_{12}, m_{13}) 110x ✓		
(m_{13}, m_{15}) 11x1	3	

The Quine-McCluskey Method

5. The terms for the expression are written into a prime implicant table, with minterms for each prime implicant checked.
6. The minterm has a single check must be included in the expression. Since the two in $\bar{A}\bar{C}D$ are covered in the first two rows, so it is unnecessary.

$$X = B\overline{C} + \overline{A}\,\overline{B}D + ABD + A\overline{B}C\overline{D}$$

	Minterms							
Prime Implicants	m_1	m_3	m_4	m_5	m_{10}	m_{12}	m_{13}	m_{15}
$B\overline{C}$ (m_4, m_5, m_{12}, m_{13})			✓	✓		✓	✓	
$\overline{A}\,\overline{B}D$ (m_1, m_3)	✓	✓						
$\overline{A}\,\overline{C}D$ (m_1, m_5)	✓			✓				
ABD (m_{13}, m_{15})							✓	✓
$A\overline{B}C\overline{D}$ (m_{10})					✓			

Espresso Algorithm

- Compared to the other methods, Espresso is essentially more efficient in terms of reducing memory usage and computation time by several orders of magnitude. There is essentially no restrictions to the number of variables.
- The Espresso algorithm has been incorporated in most logic synthesis tools.