布尔代数

The Boolean algebra is used to describe certain logic design made up with several logic gates and wires connecting them.

Laws and Theorems in Boolean Algebra

Basic Theorems

- $A+B=B+A$, and $AB=BA$.
- $(A+B)+C=A+(B+C)$, and $(AB)C=A(BC)$.
- $A(B+C)=AB+AC$.
- $A+AB=A$, and $A(A+B)=A$.
  This theorem shows that B will be ‘absorbed’ in this form of expression.
- $AB+A\bar{B}=A$, and $(A+B)(A+\bar{B})=A$.
- $A+\bar{A}B=A+B$.
  $A+\bar{A}B=A+AB+\bar{A}B=A+(A+\bar{A})B=A+B$.
- $A+BC=(A+B)(A+C)$
  Why: $(A+B)(A+C)=A+AC+BA+BC$, and $A+AC+BA=A$.
- $AB+\bar{A}C+BC=AB+\bar{A}C$, and $AB+\bar{A}C+BCD=AB+\bar{A}C$.
  Use BC to absorb BCD.
- $\overline{A\bar{B}+\bar{A}B}=\bar{A}\bar{B}+AB$.
  XNOR = XOR + NOT.

DeMorgan’s Law

$$\overline{X_1X_2\cdots X_n}=\bar{X}_1+\bar{X}_2+\cdots+\bar{X}_n$$
$$\overline{X_1+X_2+\cdots+X_n}=\bar{X}_1\bar{X}_2\cdots\bar{X}_n$$

Dual Rule

By changing operators in an expression in this way you get its dual expression:

- $+\leftrightarrow\cdot$
- $\oplus\leftrightarrow\odot$

For a equation it’s dual expression is still valid. For example:

$$A+BCD=(A+B)(A+C)(A+D)\Leftrightarrow A(B+C+D)=AB+AC+AD$$

Maxterm and Minterm

It’s easy for us to realise that for a boolean expression containing some variables—say, A, B, C and D—it can be written in this form: only several AND items added together, i.e. addiction of multiplications. An example:

$$f(A,B,C,D) = AB + ACD + D$$

And if we add invisible items (which is always 1) such as $(A + A')$ and $(D + D')$ to make each AND item has all four variables appear, then the formula can be expanded to:

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

Simplify this expression using the law $A(B + C) = AB + AC$, we will get

$$f(A,B,C,D) = ABCD + ABCD' + ABC'D + ABC'D' + AB'CD + A'BCD + A'B'CD + A'BC'D + A'B'C'D$$

If we use 1011 to represent $AB'CD$, use 0110 to represent $A'BCD'$, then the items in the formula above are:

$$
1111, 1110, 1101, 1100, 1011, 0111, 0011, 0101, 0001

$$

All of them are 4-bit binary numbers so we just transform them into decimal numbers. They are

$$15, 14, 13, 12, 11, 7, 3, 5, 1$$

Intuitively, the original boolean expression can be expressed with a series of numbers. This is called the *minterm* expression. More detailed, we use minterm expression in this algebraic way:

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

Dually, an expression can also be written in this way:

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

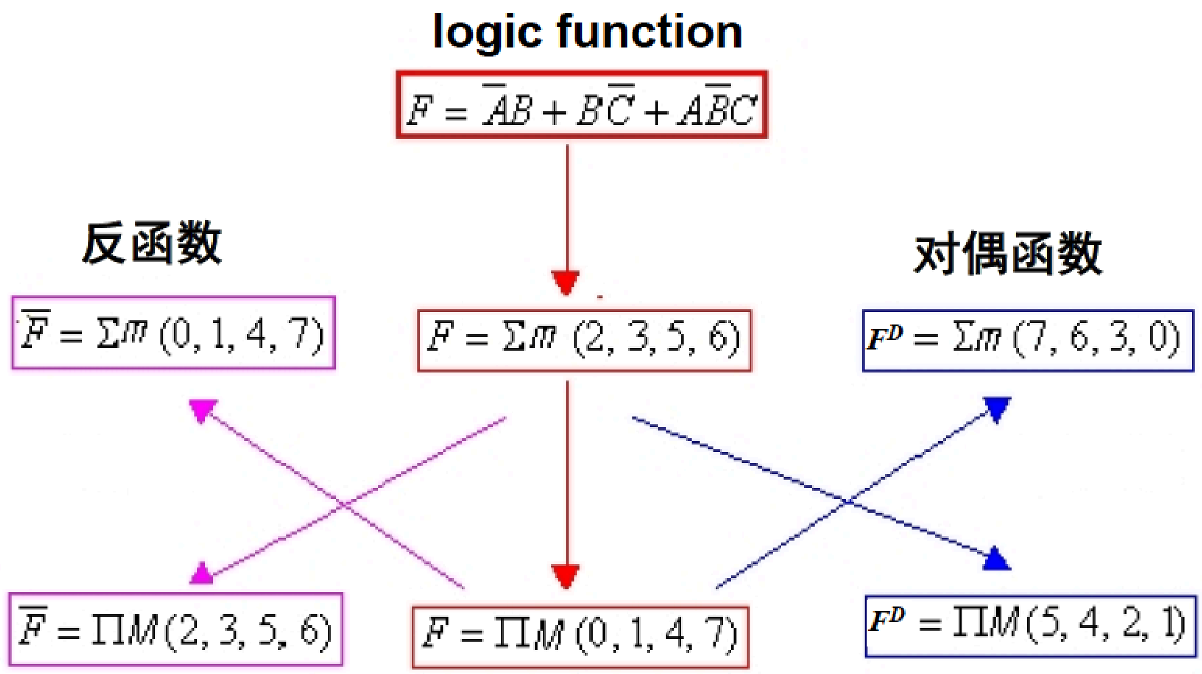
Treat $(A' + B + C)$ as binary number 100, then we have the *maxterm* expression. It is expressed in this way:

$$g(A,B,C) = \prod M(1, 5, 6)$$

According to the dual rule it’s easy for us to change an expression into the other one. For instance:

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

This figure shows some interesting relationships.



Karnaugh Map

Karnaugh Map (K. Map) is a powerful tool used in expression simplification and other scenarios.

K. Map is a table containing 2^n element cells. Each cell stands for a minterm. The cells are arranged in a Gray code way, which means the two cells next to each other have only one different bit. For 4 variables the K. Map usually likes this:

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

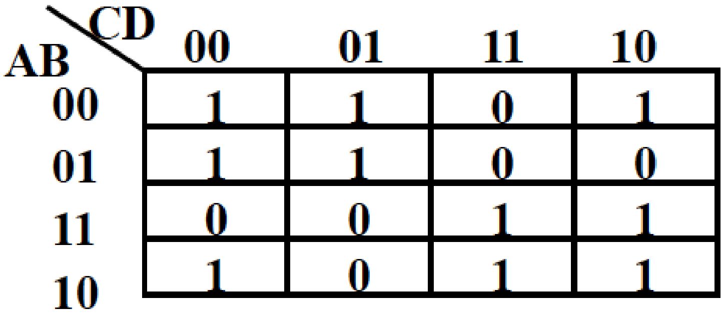
Steps when using K. Map to simplify logic expression:

Fulfil the K. Map

To fulfil the K. Map you firstly transform your expression into the form ‘addiction of multiplications’. For example,

$$F = \overline{A \oplus C \cdot \overline{\bar{B}(A\bar{C}\bar{D} + \bar{A}C\bar{D})}} = AC + \bar{A}\bar{C} + A\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}C\bar{D}$$

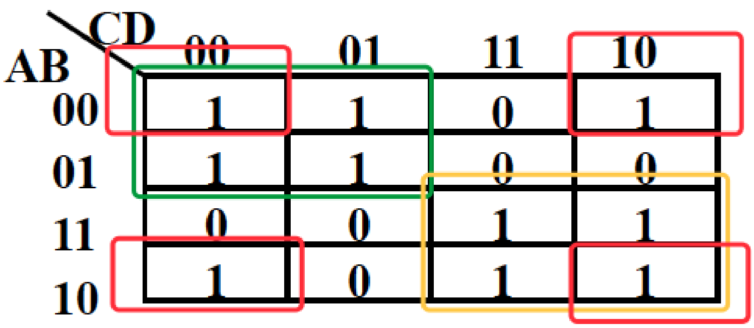
Then for each multiplication fill its corresponding K. Map cells with 1. For example, for $\bar{A}\bar{C}$, we find the row where A is 0, and mark 1 in those rows whose C is 0. Finally we will have a K. Map like this:



An interesting fact is that you can now read the minterm expression directly on this K. Map.

Circle the Prime Implicants in K. Map

Now, we make rectangular groups containing total terms in power of 2 and try to cover as many elements as you can in one group.



Note that the four red groups are the same group for the parallel edges and corners of such K. Map are connected together.

This kind of rectangular groups, or circles, are called ‘prime implicants’. ‘Implicant’ means the rectangular group only consists 1 and ‘prime’ means it has been expanded as large as possible.

Get Simplified Expression from the Implicants

Each group in the figure above stands for a multiplication item. Just write them and add them together and you get a simplified addiction of multiplications.

Look at the figure above.

- For four elements in the red circle, they have both B and D as 0, while their A and C values differ. So for this group it’s implicant is $B'D'$.
- For four elements in the green circle, they have both A and C as 0, so this group represents $A'C'$.
- For four elements in the yellow circle, they have both A and C as 1, so this group represents AC.

So finally we can read out this expression:

$$F = A'C' + AC + B'D'$$

Distinguished 1-Cell and Essential Prime Implicant

- Distinguished 1-cell means the cell in a K. map who is only covered by one prime implicant.

- Essential prime implicant means the prime implicant containing distinguished 1-cell.