Boolean Algebra-1

0.1 Karnaugh map method

A Karnaugh map is a graphocal representation of the logic system. It can be drawn directly from the truth table, minterm or maxterm Boolean expressions and it gives a minimized sum-of-products or a minimized product-of-sums.

Definition 1. *Two boolean terms are logically adjacent if and only if they contain the same variables and differ in the form of only one variable.*

Example 1. The terms $a.b.c$ and $a.\bar{b}.c$ are logically adjacent, we have $a.b.c + a.\bar{b}.c = a.c$

0.1.1 Karnaugh map construction

A Karnaugh map is composed of a certain number of cells, each of which is reserved for a term (minterm or maxterm) of a logic function. On each map, the combination of the variables are placed in accordance with the order of Gray's encoding such that adjacent terms are in the neighboring cells or in the cells at map ends. Simplification by Karnaugh map becomes difficult when the number of variables exceeds six.

b \ a	0	1
0	0	2
1	1	3

c \ ab	00	01	11	10
0	0	2	6	4
1	1	3	7	5

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

The simplification of a logic function by Karnaugh map is carried out by grouping the adjacent cells that contains $1s$. The number of cells in a group must be a power of 2, $(2^k; k = 0, 1, 2, 3, \cdots)$.

To minimize a boolean expression using Karnaugh map, we proceed as.

1. We form groups of adjacents cells which contain 1. (We must maximize the number of in each group)
2. Each group of 1 of 2^k adjacent cases give a product term of $n-k$; logical variables (n is number of variables) such that the k^{th} variable which changes state will be eliminated.
3. The minimized expression will be the sum of all minimized terms.

Example 2. 1. We shall simplify the expression $Y = \overline{a}.\overline{b}.c + \overline{a}.b.c$

c \ ab	00	01	11	10
0	0	1	1	0
1	0	0	0	0

duad of adjacent $1s$: the variable which changes state can be eliminated : $Y = b.\overline{c}$

2. We simplify $Y = a.b.c + a.b.\overline{c} + a.\overline{b}.c + a.\overline{b}.\overline{c}$.

c \ ab	00	01	11	10
0	0	0	1	1
1	0	0	1	1

quad of adjacent $1s$: the variables which change states can be eliminated : $Y = a$

3. We simplify $Y = \overline{a}.\overline{b}.\overline{c} + a.\overline{b}.\overline{c}$.

c \ ab	00	01	11	10
0	1	0	0	1
1	0	0	0	0

duad of adjacent $1s$: the variable which changes state can be eliminated : $Y = \overline{b}.\overline{c}$

4. We simplify $Y = \overline{a}.\overline{b}.\overline{c} + \overline{a}.\overline{b}.c + a.\overline{b}.\overline{c} + a.\overline{b}.c$

c \ ab	00	01	11	10
0	1	0	0	1
1	1	0	0	1

quad of adjacent $1s$: two variables which change states can be eliminated : $Y = \overline{b}$

5. We simplify $\overline{a}.\overline{b}.c + \overline{a}.b.c + a.b.c + a.\overline{b}.c$

c \ ab	00	01	11	10
0	0	0	0	0
1	1	1	1	1

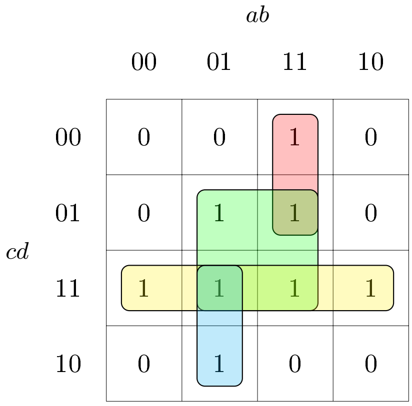
quad of adjacent $1s$: two variables which change states can be eliminated : $Y = c$

6. We simplify $Y = \overline{a}.b.\overline{c}.\overline{d}+a.b.\overline{c}.\overline{d}+\overline{a}.b.\overline{c}.d+a.b.\overline{c}.d+a.b.\overline{c}.d+\overline{a}.b.c.d+a.b.c.d+\overline{a}.b.c.\overline{d}+a.b.c.\overline{d}$.

cd \ ab	00	01	11	10
00	0	1	1	0
01	0	1	1	0
11	0	1	1	0
10	0	1	1	0

duad of adjacent $1s$: three variables which change states can be eliminated : $Y = b$

7. We simpliy $Y = \overline{a}.\overline{b}.c.d + \overline{a}.b.\overline{c}.d + \overline{a}.b.c.d + \overline{a}.b.c.\overline{d} + a.b.\overline{c}.\overline{d} + a.b.\overline{c}.d + a.b.c.d + a.\overline{b}.c.d$.

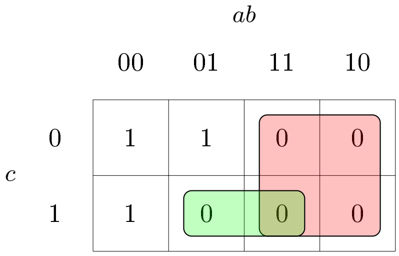


In each group only those variables that hold the same logic state, appear in the simplified expression, $Y = a.b.\overline{c} + b.d + c.d + \overline{a}.b.c$.

0.1.2 Simplification of product of sum form using Karnaugh-map

The simplification process of a POS (Product of Sums) form is similar to that used for a SOP (Sum of Products), except that $0s$ need to be grouped to produce minimized sum terms.

Example 3. We define a logical function by its product of sum. $f(a,b,c) = (\overline{a}+\overline{b}+c).(\overline{a}+b+c).(a+\overline{b}+\overline{c}).(\overline{a}+\overline{b}+\overline{c}).(\overline{a}+b+\overline{c})$. Corresponding Karnaugh-map is



$$f(a,b,c) = \overline{a}.(\overline{b}+\overline{c}).$$

0.1.3 Incompletely defined functions

Some combinations are physically impossible, and the value of the corresponding function for these combinations is indifferent or unknown. The function's value will be represented by φ or x. For simplification of incompletely defined functions, the indifferent states are included to increase the grouping's size.

Example 4. Using the Karnaugh map, simplify the following function

$$f(a,b,c,d) = \sum(1,2,5,6,9) + \sum\varphi(10,11,12,13,14,15).$$

cd \ ab	00	01	11	10
00	0	0	x	0
01	1	1	x	1
11	0	0	x	x
10	1	1	x	x

$$f(a,b,c,d) = \overline{c}.d + c.\overline{d}.$$

0.1.4 Redundant terms

We treate an example. Let us define a boolean function f defined by the following truth table.

a	b	c	$f(a,b,c,d$
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

the algebric expression is $f(a,b,c) = \overline{a}.b.\overline{c} + a.\overline{b}.c + a.b.\overline{c} + a.b.c.$

1. Algebric simplification : $f(a,b,c) = \overline{a}.b.\overline{c} + a.\overline{b}.c + a.b.\overline{c} + a.b.c = (\overline{a} + a).b.\overline{c} + a.c(b + \overline{b}) = b.\overline{c} + a.c.$
2. Karnaugh map's simplification :

c \ ab	00	01	11	10
0	0	1	1	0
1	0	0	1	1

c \ ab	00	01	11	10
0	0	1	1	0
1	0	0	1	1

$f(a,b,c) = b.\overline{c} + a.c$ $\quad\quad$ $f(a,b,c) = a.b + b.\overline{c} + a.c$

The first expression is the simplest, we can obtain it after applying Theorem **??** (Consensus theorem) to the second expression.

Definition 2. *A logical term is considered to be redundant if it covers all of the cells in a Karnaugh map that another term already covers. Without modifying the truth table, this term can be deleted from the equation.*