Simplification des fonctions logiques à l'aide des tableaux de Karnaugh

I) Définition

Le tableau de Karnaugh est une représentation de la fonction logique. Elle est plus parlante que la table de vérité et permet la simplification des fonctions.

⇨ La table de vérité : le nombre de variables n donne 2^n lignes

⇨ Le tableau de Karnaugh : le nombre de variable donne 2^n cases

Exemple : pour une équation du type $y = a.\bar{b}$

b	a	y
0	0	0
0	1	1
1	0	0
1	1	0

La table de vérité

Le tableau de Karnaugh

	$\bar{a}$	a
$\bar{b}$	0	1
b	0	0

Règle :
Les cases d'un diagramme de Karnaugh ne peuvent pas être placées dans un ordre quelconque. Il est nécessaire que le passage d'une case à une case adjacente (case ayant un côté commun) se traduise par le changement d'état d'une seule variable.

Exemple : F(a,b,c) donc une fonction à 3 variables

c \ ba	00	01	11	10
0	(0)	(1)	(3)	(2)
1	(4)	(5)	(7)	(6)

Case (3) : $a.b.\bar{c}$

Case (7) : $a.b.c$

Case (6) : $\bar{a}.b.c$

Le passage de la case (3) à la case (7) est réalisé avec le changement d'une seule variable.

II) Marquage d'une fonction dans le tableau

$$\boxed{F = \sum \text{produit}}$$

A chaque terme de la somme correspond une case du tableau.

F= $\bar{a}.\bar{b}.\bar{c} + \bar{a}.\bar{b}.c + \bar{a}.b.\bar{c}$

0 0 0 0 1 0 0 0 1

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

III) Simplification d'une fonction à deux variables

Exemple : Y= $\overline{a}.\overline{b} + \overline{a}.b + a.b$

Si au lieu de représenter la fonction Y, on cherche son complément, on obtient $\overline{Y} = a.\overline{b}$

b	a	Y	$\overline{Y}$
0	0	1	**0**
0	1	0	**1**
1	0	1	**0**
1	1	1	**0**

$$\boxed{\overline{Y} = a.\overline{b}}$$

D 'ou en appliquant la complémentation on obtient que $\overline{\overline{Y}} = Y = \overline{a.\overline{b}} = \overline{a} + b$

Cette simplification peut être obtenu rapidement à l'aide du tableau de karnaugh en appliquant la rêgle suivante : lorsque 2 cases adjacentes contiennent chacune un « 1 »dans leur représentation, une simplification peut se faire de la façon suivante :

b \ a	0	1
0	1	0
1	1	1

b

$\overline{a}$

Y

> <u>Remarque :</u> le regroupement de cases dans le but de simplifier une fonction ne peut se faire que pour un nombre de case adjacente égal à {1 ;2 ;4 ;8 ;16} soit (2^n). Chaque regroupement correspond à un produit logique dans lequel on ne prend en compte que les variables communes aux cases regroupées.

IV) Simplification d'une fonction à trois variables

Exemple 1 :

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

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

Y = $\overline{a}$

Exemple 2 :

$X = \overline{a}.\overline{b}.c + \overline{a}.b.\overline{c} + a.\overline{b}.\overline{c}$

c \ ab	00	01	11	10
0				1
1				

X =

V) Méthode de simplification d'une fonction logique

Pour n variable ⇨ Tableau de karnaugh à 2^n cases.

- Ecriture de l'équation sous une somme de produit Y = a.b.c + ….+ ….
- Construction du tableau de karnaugh avec le marquage des « 1 ».
- Regroupement des cases adjacentes marquées d'un « 1 » (groupe de 1, 2, 4, 8 ,16).
- Une case marquée d'un « 1 » peut être utilisée plusieurs fois dans les regroupements.
- Toute case marquée d'un « 1 » doit participer au moins à un regroupement (ce dernier pouvant être constitué d'une seule case).
- Rechercher les variables qui ne changent pas pour les regroupements et en déduire le produit.
- Réaliser la somme des produits pour obtenir l'équation simplifiée.

VI) Exercices

Sortir les équations simplifiées en utilisant les tableaux de KARNAUGH.

dc \ ba	00	01	11	10
00	0	1	1	0
01	1	1	1	1
11	0	1	1	0
10	0	1	1	0

A =

dc \ ba	00	01	11	10
00	1	0	0	1
01	0	0	0	0
11	0	1	1	0
10	1	0	0	1

B =

dc \ ba	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	0	0	0	0
10	1	1	1	1

C =

dc \ ba	00	01	11	10
00	0	1	1	0
01	1	1	1	1
11	1	1	1	1
10	0	1	1	0

D =

dc \ ba	00	01	11	10
00	0	0	0	0
01	1	0	0	1
11	1	0	0	1
10	0	0	0	0

E =

dc \ ba	00	01	11	10
00	0	1	0	1
01	1	1	1	1
11	1	1	1	1
10	0	1	0	1

F =

dc \ ba	00	01	11	10
00	1	1	1	1
01	0	1	1	0
11	0	1	1	0
10	1	1	1	1

G =

dc \ ba	00	01	11	10
00	1	0	0	1
01	0	1	1	0
11	0	1	1	0
10	1	0	0	1

H =

dc \ ba	00	01	11	10
00	1	0	0	1
01	1	1	1	1
11	1	1	0	0
10	0	0	0	0

I =

dc \ ba	00	01	11	10
00	0	0	1	0
01	1	0	1	1
11	1	1	1	1
10	0	0	1	0

J =

dc \ ba	00	01	11	10
00	0	1	1	0
01	1	0	0	1
11	1	0	0	1
10	0	1	1	0

K =

dc \ ba	00	01	11	10
00	1	0	0	1
01	1	0	0	1
11	1	0	0	1
10	1	1	1	1

L =

VII) Rappels sur les systèmes de codage.

7,1) Systèmes binaires.

En binaire, on distingue trois principaux systèmes de codage :

- binaire pur : (1-2-4-8) poids binaire / voir test début d'année;
- binaire réfléchi : (code GRAY ou code réfléchi / voir test début d'année);
- binaire D C B ou B C D (binaire codé décimal de 0 à 9 soit de 0000 à 1001).

a) Code binaire naturel.

Dans ce codage, on utilise le poids binaire de chaque chiffre en fonction de son rang. Nous pouvons faire l'analogie entre le système binaire et le système décimal.

1 9 9 9 ⟶ 1 1 1 1 1 0 0 1 1 1 1

10^3 10^2 10^1 10^0 2^{10} 2^9 2^8 2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0

b) Code binaire réfléchi.

Dans ce codage, un seul bit change d'état lorsque l'on passe d'un terme au suivant.

A l'apparition d'une variable supplémentaire on fait la symétrie du code déjà obtenu plus le nouveau bit à 1.

Le code peut se refermer sur lui-même sans perdre ses propriétés dans la mesure ou le dernier terme se situe juste avant un axe de symétrie.

Intérêt: Ce codage évite les états indéterminés lors du passage d'un terme à un autre terme adjacent. Risque d'aléas de fonctionnement.

c	b	a
0	0	0
0	0	1
0	1	1
0	1	0
1	1	0
1	1	1
1	0	1
1	0	0

c) Code binaire D C B (Décimal Codé Binaire).

Dans ce codage, chaque chiffre décimal est converti en binaire, indépendamment des autres chiffres.

2	3	0	1
0 0 1 0	0 0 1 1	0 0 0 0	0 0 0 1

Ce code est utilisé dans les systèmes traitant des nombres décimaux uniquement :

- En comptage (instruments de mesure et compteur)
- Dans les calculettes de poche qui travail sur 16 bits.

<u>Inconvénient</u>: Il nécessite plus de bits que le binaire naturel pour coder le même nombre décimal.

7,2) Système hexadécimal.

Le codage hexadécimal est très utilisé dans les systèmes à microprocesseur car il simplifie l'écriture des nombres binaires.

Ce codage utilise 16 symboles [0 . . 9 et A . . F]

L'analogie avec le système décimal peur être faite .

1 9 9 9 ⟶ 7 C F

10^3 10^2 10^1 10^0 16^2 16^1 16^0

Chaque chiffre héxadécimal est défini par quatre bits. (Voir calculatrice scientifique de windows)

Quartet (4 bits) ⟶ $2^4 = 16$ combinaisons;

Octet (8 bits) ⟶ $2^8 = 256$ combinaisons.

7,3) Exercices.

Convertir en binaire, puis en hexadécimal les nombres suivants :

17														
35														
62														
77														
80														
100														
77					0	1	0	0	1	1	0	1	4	D
128														
256														
1339														
759														
4096														
4095														
255	0	0	0	0	1	1	1	1	1	1	1	1	F	F

7 4) Exercices sur les tableaux de Karnaugh :

Il est demandé de traiter l'ensemble des exercices de la page 6.

T. D. d'Automatisme
Logique combinatoire

1°/ Simplifier les fonctions suivantes :

$$S1 = a\,(a + b)$$

$$S2 = (a + b)(\bar{a} + b)$$

$$S3 = (a\bar{b} + c)(a + \bar{b})\,c$$

$$S4 = (a + b)\,c + \bar{a}(\bar{b} + c) + \bar{b}$$

$$S5 = (a + b + c)(\bar{a} + b + c) + ab + bc$$

$$S6 = a + \bar{a}b + \bar{a}\bar{b}c + \bar{a}\bar{b}\bar{c}d + \bar{a}\bar{b}\bar{c}\bar{d}$$

$$S7 = a + abc + \bar{a}bc + \bar{a}b + ad + a\bar{d}$$

$$S8 = ab\bar{c} + b\,(a + \bar{c}) + \overline{\bar{a} + b + \bar{a}c}$$

2°/ Simplifier en passant par le complément des fonctions :

S9 = cd + ab + cd + ab =

S10 = ac + bc + ad + bd =

3°/ Trouver les expressions minimales de chacun des diagrammes de Karnaugh ci dessous :

M

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

N

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

P

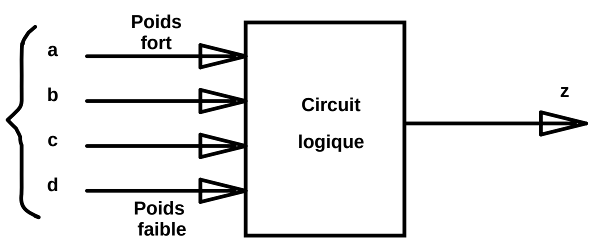
cd \ ab	00	01	11	10
00	1	0	0	1
01	1	1	1	1
11	1	1	1	1
10	1	0	0	1

S

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

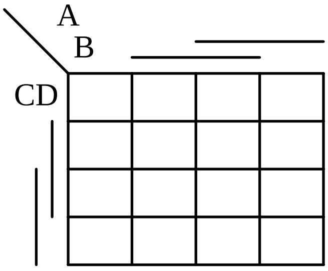
4°/ Le circuit logique ci-dessous possède 4 entrées a, b, c et d formant un nombre binaire dont le bit de poids le plus fort est : "a", et le bit de poids le plus faible est : "d". Le circuit logique donne un niveau HAUT quand le chiffre présent est supérieur à $0110_2 = 6_{10}$.

Trouver l'expression logique de ce circuit ; il est conseillé de construire une table de vérité utilisant la numération binaire pure puis de simplifier à l'aide du diagramme de Karnaugh. Tracer le logigramme correspondant.



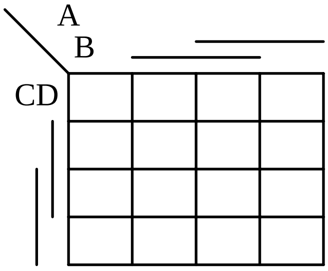
5°/ Simplifier les équations suivantes à l'aide des tableaux de Karnaugh.

S1= $abc+bcd+abd+abcd$



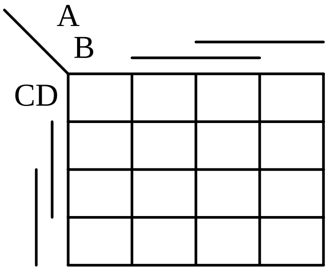
Résultat : S1=

S2 = $abc+ab\bar{c}+abd+\bar{a}bcd$



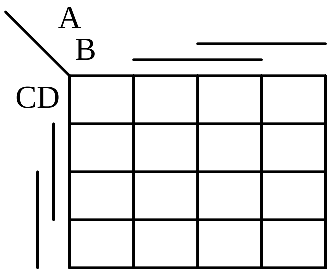
Résultat : S2=

S3 = $a\overline{bc}+ab\overline{cd}+\overline{ab}c+\bar{a}bc\bar{d}$



Résultat : S3=

S4 = $\overline{ab}c+a\overline{bcd}+\bar{a}bc+ab\overline{cd}$



Résultat : S4=

6° / Etablir les équations à partir de la table de vérité et les simplifier par la méthode de Karnaugh.

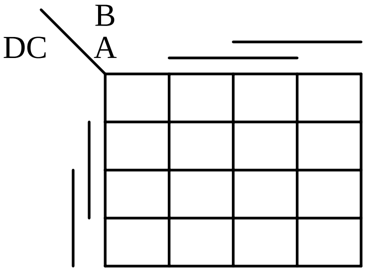
D	C	B	A	S5	S6
0	0	0	0	0	1
0	0	0	1	1	1
0	0	1	0	1	1
0	0	1	1	0	1
0	1	0	0	1	0
0	1	0	1	1	1
0	1	1	0	0	0
0	1	1	1	0	1
1	0	0	0	1	0
1	0	0	1	0	1
1	0	1	0	1	0
1	0	1	1	1	1
1	1	0	0	0	1
1	1	0	1	0	1
1	1	1	0	0	0
1	1	1	1	1	0

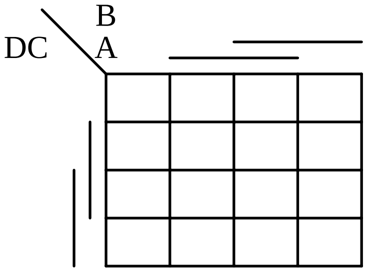
Ecrire les équations non simplifiées

S5 =

S6 =

Simplifier les équations à l'aide de Karnaugh (voir page suivante)



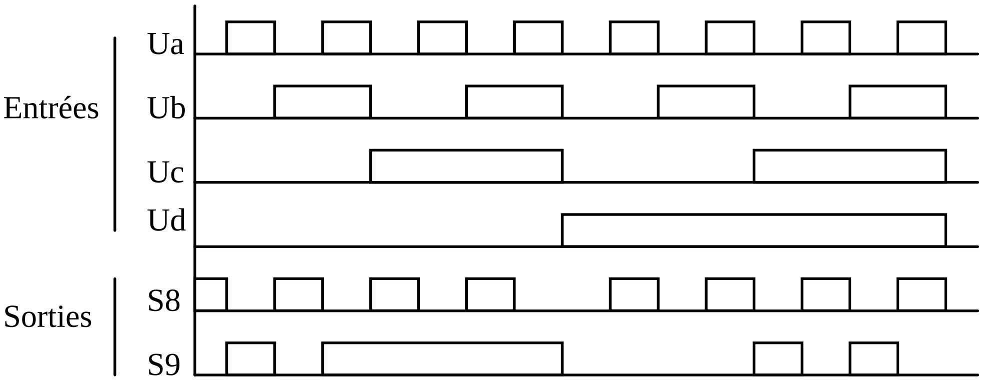


Résultat : S5= Résultat : S6=

Réaliser les logigrammes de S5 et de S6

7°/ Etablir les équations des sorties à partir des chronogrammes.et les simplifier par la méthode de Karnaugh

1° chronogrammes



2° chronogrammes