République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université M'hamed Bougara - Boumerdès

Faculté des Sciences
Département d'Informatique

Domaine : Mathématiques et Informatique

Filière : Informatique

Spécialité : Ingénieur en informatique

Projet de methode de Quin-McCluskey

Présenté par:

Bouabdallah Ayoub Ibrahim

Houmel Amar

Lamri Walid

Tous du grp 07 section 02

SOMMAIRE

INTRODUCTION :

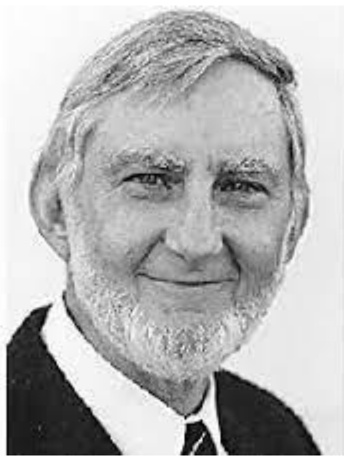
Alors comme on a étudie dans le cours « Utilisation impossible de la méthode de karnaugh si le nombre de variable est supérieur à 6. « on peut la faire facilement avec la méthode de Quine-McCluskey.

La méthode de Quine-McCluskey est un algorithme utlisé en logique booléenne pour simplifier les expressions booléennes complexes. Inventée dans les années 1950 par Willard Van Orman Quine et Edward J.McCluskey. Où Willard Van Orman a d'abbord proposé une méthode de simplification, puis Edward J.McCluskey a développé une approche algrothmique puis systématique pour résoudre ce même problème.

Alors leur méthode a eu un impact significatif dans le domaine de la conception des circuits logique et de l'agrébre booléenne, car elle permettait de simplifier les expressions complexes de manière plus efficace, réduisant ansi le nombre de composants nécessaires dans les circuits logiques.

EDWARD J.MCCLUSKEY

(1929 - 2016)



WILLARD VAN ORMAN QUINE

(1908-2000)

DÉFINITION GÉNÉRAL SUR LA MÉTHODES DE QM

Définition :

La méthode de Quine McCleskey est une approche formelle tabulaire qui applique la loi distributive booléenne à différents termes.Elle vise à trouver la somme minimale de produits en éliminant les littéraux présents simultanément dans deux termes les considérant comme des compléments.

Cette méthode est basée sur la loi sur le théorème de complémentation (propriété d'adjacence), qui dit :

$$A . B + A . B = B$$

Donc elle comporte trois étapes :

1/ Déviser les mintermes en groupes selon le nombre de répétition de nombre 1 de ces mintermes en binaires, et regroupez-les ensuite sous forme duo dans d'autres groupes ordonnés d'après le nombre de répétition du nombre 1

2/ Trouver les impliquants principaux IP (il ne peux pas être raccourci). On les trouve en comparant les mintermes de chaque groupe avec les mintermes suivant (selon certaines conditions), nous rassemblons toutes les valeurs minimales jusqu'à ce que reste à la fin soit ce qui ne peut pas être additionné

3/ Á la fin, identifier le nombre minimal des mintermes qui représentent toutes les mintermes de l'expression simplifier

Les étapes en détail

	La méthode Quine-McCluskey
étape 01	D'après la table de vérité de la fonction donner , on prend les cas dans lequel la fonction est 1, puis convertir les nombres du système binaire au système décimal pour obtenir la somme de mintermes sous la forme suivante : << F (variable) = Σ (nombres au système décimal)>> Dans le cas où on n'a que la 1er forme canonique, on utilise cette technique exemple : A = 1 et A'=0 Pour déterminer les nombres en binaire ……….
étape 02	Créer un tableau affichant les mintermes de la fonction par leurs représentations binaires.
étape 03	Créer 2éme tableau, Trier les mintermes dedans d'après la répétition de nombre de bits "1" dans leur représentation binaire. Par exemple -mintermes 0 en binaire 0000 alors en va le classer dans le grp 0 -mintermes 2 en binaire 00$\boxed{1}$0 alors en va le classer dans le grp 1 -mintermes 9 en binaire $\boxed{1}$00$\boxed{1}$ alors en va le classer dans le grp 2
étape 04	On compare les mintermes de chaque groupe avec tout les mintermes de groupe suivant si la condition suivante se réalise « Si la paire comparée est adjacente (c'est-à-dire, s'ils différent d'une seule variable) ils seront combinés pour former un nouveau termes, on élimine le nombre qu'ils partagent après on les trier dans le 3éme tableau selon la répétition le nombre de bits 1 dans la représentation binaire. Exemple : 0 000 2 (0.2) 00-0 on la metre dans le grp 0
étape 05	Pour chaque groupe de mintermes, répéter le processus précédent en générant une nouvelle liste de termes où les variables éliminées sont remplacées par des tirets
étape 06	Avec le même principle de l'étape 04. Comparez les termes du 3ème Tableau pour rechercher d'autres combinaisons si la condition suivante se réalise « si la paire comparée est adjcente (ils sont différent d'un seul variable éliminées »). Tout les termes qui restent non combinée sont des impliquants premiers. On les mettre tous (les nouveaux combinaisons et les impliquants premiers "On supprime les termes répétés ") dans le 4éme tableau ?
étape 07	Séléctionner un sous-ensemble minimal des impliquants premiers qui couvrent tous les termes de la fonction boocléenne d'origine.

Remarque 1

-Les deux termes de combinaison sont crochés (✓) dans la liste originale indiquant qu'ils ne sont pas des impliquants premiers

- Les termes de la première table qui ne trouvent pas de correspondance sont appelés les impliquants premiers et ils sont marqués d'un astérisque.(*)

- Dans le dernier étape traduit le 0 par variable bar et 1 par variable et le tirer on le traduit pas.

Exemple pour mieux comprendre :

Exemple

Soit F fonction avec la table de vérité, simplifier la fonction F :

A	B	C	D	F
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	1	1	1
1	0	0	0	1
1	0	1	1	0
1	1	0	0	0
1	1	1	1	0
1	1	0	0	1
1	1	1	1	0

Solution de l'exemple :

étape 01 & étape 02

Donc F(A,B,C,D) = Σ (0,1,2,5,6,7,8,9,10,14)

0000	0
0001	1
0010	2
0101	5
0110	6
0111	7
1000	8
1001	9
1010	10
1110	14

TABLE 01

Etape 03

		ABCD	
Grp 0	0	0000	✓
Grp 1	1 2 8	0001 0010 1000	✓ ✓ ✓
Grp 2	5 6 9 10	0101 0110 1001 1010	✓ ✓ ✓ ✓
Grp 3	7 14	0111 1110	✓ ✓

TABLE02

Etape 04 & 05 :

		ABCD	
Grp 0'	0.1 0.2 0.8	000- 00-0 -000	✓ ✓ ✓
Grp 1'	1.5 1.9 2.6 2.10 8.9 8.10	0-01 -001 0-10 -010 100- 10-0	* ✓ ✓ ✓ ✓ ✓
Grp 2'	5.7 6.7 6.14 10.14	01-1 011- -110 1-10	* * ✓ ✓

0000
0001 → On voit qu'ils partagent 3 bits et ils changent un seul bits alors la condition se réalise

-Ensuite on remplace le bits différent par un tiret '-'

-automatiquement les nouveaux groupes sont classé selon le nombre de bits '1'

TABLE 03

Etape 06 :

		ABCD
Grp 0''	0.1,8,9 0.2,8,10 ~~0.8,1,9~~ ~~0,8,2,10~~	-00- -0-0 ~~-00-~~ ~~-0-0~~
Grp 1''	2.6,10,14 ~~2.10,6,14~~	--10 ~~--10~~

(0,1) →000-
(8,9) →100- → On voit que la condition se réalise.

-Ensuite on remplace le bits différant par un tirer '-' et on éliminé la répétition.

TABLE 04

Etape 07 :

(0,1,8,9) → -00- → B'C'

(0,2,8,10) → -0-0 → B'D'

(2,6,10 ;14) → --10 → CD'

(1,5) → 0-01 → A'C'D

(5,7) → 01-1 → A'BD

(6,7) → 011- → A'BC

Impliquants Premiers	Les mintermes									
	0	1	2	5	6	7	8	9	10	14
B'C'	X	X					X	X		
B'D'	X		X				X	X		
CD'			X		X				X	X
A'C'D		X		X						
A'BD				X		X				
A'BC					X	X				

Donc on prend le sous-ensemble minimal qui couvrent tout les termes de la fonction .

F(A,B,C,D) = CD' + B'D' + A'BD

NOTRE PROPRE EXEMPLE:

Comme c'est le cas dans notre deuxième pays Palestine.

On considère l'exemple suivant :

Les forces Palestine du Hamas ont 7 lances pour l'utiliser contre les pouvoirs Israël et ces lances ont 4 (A,B,C,D) boutons différents pour les activer donc l'activation est le protocole suivant :

- Si au moins trois boutons sont enfoncés.
- Si le bouton C est enfoncé et D n'est pas enfoncé.
- Si D est le seul bouton enfoncé.

Solution de l'exemple :

Etape 01 & 02 : d'après les régles nous aurons le tableau suivant :

a	b	c	d	F
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Donc $F(A, B,C,D) = \Sigma\ (1,2,6,7,10,11,13,14,15)$

1	0001
2	0010
6	0110
7	0111
10	1010
11	1011
13	1101
14	1110
15	1111

TABLE 01

Etape 03 :

		ABCD	
Grp 1	1	0001	*
	2	0010	√
Grp 2	6	0110	√
	10	1010	√
Grp 3	7	0111	√
	11	1011	√
	13	1101	√
	14	1110	√
Grp 4	15	1111	√

TABLE02

Etape 04 & 05 :

		ABCD	
Grp 1'	2.6	0-10	√
	2.10	-010	√
Grp 1'	6.7	011-	√
	6.14	-110	√
	10.11	101-	√
	10.14	1-10	√
Grp 2'	7.15	-111	√
	11.15	1-11	√
	13.15	11-1	*
	14.15	111-	√

0010
0110 → On voit qu'ils partagent 3 bits et ils changent un seul bits alors la condition se réalise

-Ensuite on remplace le bits différent par un tiret '-'

-automatiquement les nouveaux groupes sont classé selon le nombre de bits '1'

TABLE 03

Etape 06 :

		ABCD
Grp 0"	2.6.10.14	--10
	~~2.10.6.14~~	~~--10~~
Grp 1"	6.7.14.15	-11-
	~~6.14.7.15~~	~~-11-~~
	10.11.14.15	1-1-
	~~10.14.11.15~~	~~1-1-~~

TABLE 04

(2.6) →0-10
(10.14) →1-10 → On voie que la condit se réalise

-Ensuite on remplace le bits différant par un tirer '-' et on élimine la répétition.

Etape 07 :

(2.6.14.15) → --10 → CD’

(6.7.14.15) → -11-→ BC

(10.11.14.15) → 1-1-→ AC

(1) → 0001→A’B’C’D

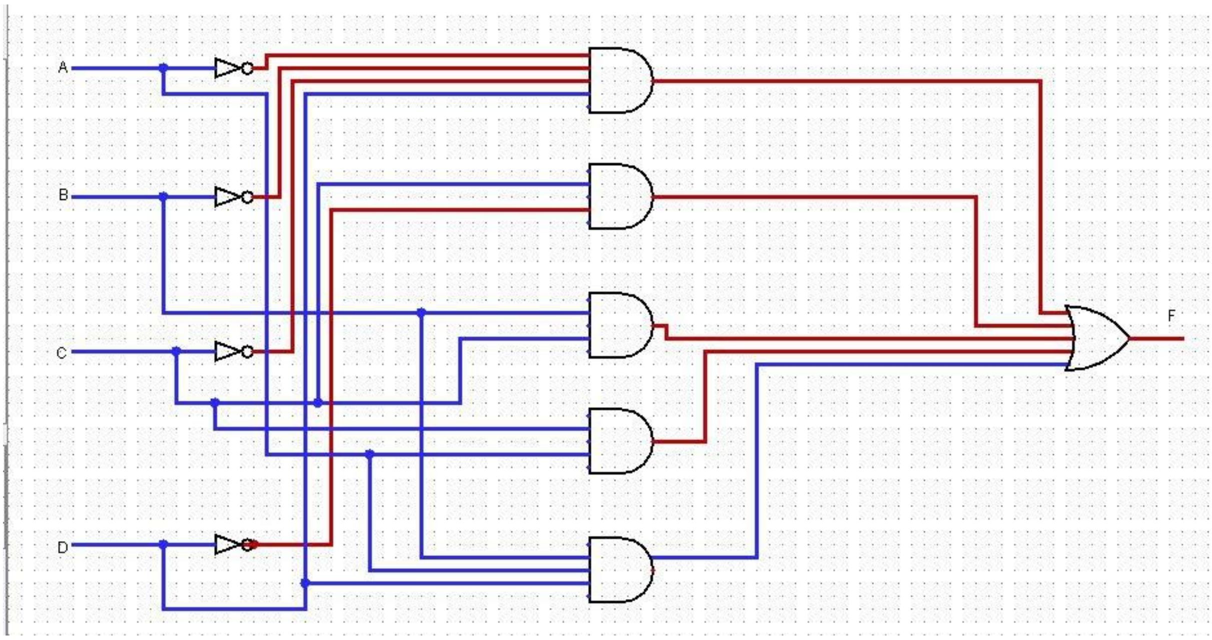
(13.15) →11-1→ABD

Impliquants Premiers	Les mintermes								
	1	2	6	7	10	11	13	14	15
CD’		X	X		X			X	
BC’			X	X				X	X
AC					X	X		X	X
A’B’C’D	X								
ABD							X		X

Donc on prend le sous-ensemble minimal qui couvrent tout les termes de la fonction .

F(A,B,C,D)=A’B’C’D+CD’+BC+AC+ABD

Le circuit combinatoire de la fonction :



CONCLUSION :

La méthode de Quine-Mccluskey est une approche puissante pour simplifier les expression booléennes , offrant une solution systématique et exhaustive pour minimiser les fonctions logiques . Son efficacité réside dans sa capacité à automatiser et à rationaliser le processus de simplification, permettant ainsi une mise réduction optimale des termes et une mise en évidence claire des relations logiques.