Algoritmo Quine-McCluskey

Es un método de simplificación de funciones booleanas desarrollado por Willard Van Omar Quine y Edward J. McCluskey. Es funcionalmente idéntico al mapa de Karnaugh, pero su forma tabular lo hace más eficiente para su implementación en lenguajes computacionales, y provee un método determinista de conseguir la mínima expresión de una función booleana.

Pasos

1) Encontrar todos los implicantes primos de la expresión

2) Usar esos implicantes en una tabla de implicantes primos para encontrar los implicantes primos esenciales, los cuales son necesarios y suficientes para generar la expresión "la función"

Complejidad

Mejor que Karnaugh para más de cuatro variables, pero el tiempo de resolución crece exponencialmente con el aumento de variables, el número de implicantes primos de "n" variables es:

3^n / n ESM $n=32 \rightarrow 6{,}5 \times 10^{15}$ implicados primos

Ejemplo

Simplificar usando método Q.Mc. $F(A,B,C,D) = \sum m(1,5,6,7,10,11,12,13,15)$

Paso 1: Hacer una tabla, organizada según su índice (número de unos del binario)

Minter	A	B	C	D	Indice
1	0	0	0	1	1
5	0	1	0	1	2
6	0	1	1	0	
10	1	0	1	0	
12	1	1	0	0	
11	1	0	1	1	3
7	0	1	1	1	
13	1	1	0	1	
15	1	1	1	1	4

Índice nos dice cuantos 1s tiene el binario

organizo la tabla anterior por su índice:

Minter	A	B	C	D	Indice
1	0	0	0	1	1
5	0	1	0	1	2
6	0	1	1	0	
10	1	0	1	0	
12	1	1	0	0	
11	1	0	1	1	3
7	0	1	1	1	
13	1	1	0	1	
15	1	1	1	1	4

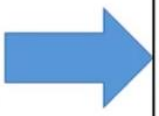
Paso 2. Identificar cuales cambian en 1 solo bit y ese bit cambiarlo por un "-"

Ejercicio

$F(A,B,C,D) = \sum m(1,5,6,7,10,11,12,13,15)$

Paso 2: Identificar cuales cambian en solo 1 bit y ese bit cambiarlo por un "-"

Minter	A	B	C	D	Indice
1	0	0	0	1	1
5	0	1	0	1	2
6	0	1	1	0	
10	1	0	1	0	
12	1	1	0	0	
11	1	0	1	1	3
7	0	1	1	1	
13	1	1	0	1	
15	1	1	1	1	4



Minter		A	B	C	D	Binario	Binario
1	1	0	0	0	1	0-01	1
5	5	0	1	0	1		
5	5	0	1	0	1	01-1	2
7	7	0	1	1	1		
5	5	0	1	0	1	-101	
13	13	1	1	0	1		
6	6	0	1	1	0	011-	
7	7	0	1	1	1		
10	10	1	0	1	0	101-	
11	11	1	0	1	1		
12	12	1	1	0	0	110-	
13	13	1	1	0	1		
7	7	0	1	1	1	-111	3
15	15	1	1	1	1		
11	11	1	0	1	1	1-11	
15	15	1	1	1	1		
13	13	1	1	0	1	11-1	
15	15	1	1	1	1		

Paso 3. Identificar adyacencia de segundo orden

→ En la columna "Binario" observar cual tiene el guion en el mismo lugar y solo cambia en 1 bit
→ Donde cambia el bit poner un "-"
→ Marcar "✓" los que no se pudieron combinar
→ Los que tienen "✓" marcarlo con "x" en la figura 3 viendo los miniterminos
→ Marcar con "★" las simplificaciones que tengan un miniterminio marcado con "x" pin repetir con uno es suficiente
→ Las simplificaciones marcadas "★" llevarlas a la simplificación final.

Ejercicio

$$F(A,B,C,D) = \sum m(1,5,6,7,10,11,12,13,15)$$

Paso 3: Identificar adyacencias de 2do orden

Minter		A	B	C	D	Binario	Binario
1	1	0	0	0	1	0-01 ✓	1
5	5	0	1	0	1		
5	5	0	1	0	1	01-1	
7	7	0	1	1	1		
5	5	0	1	0	1	-101	
13	13	1	1	0	1		
6	6	0	1	1	0	011- ✓	2
7	7	0	1	1	1		
10	10	1	0	1	0	101- ✓	
11	11	1	0	1	1		
12	12	1	1	0	0	110- ✓	
13	13	1	1	0	1		
7	7	0	1	1	1	-111	
15	15	1	1	1	1		
11	11	1	0	1	1	1-11 ✓	3
15	15	1	1	1	1		
13	13	1	1	0	1	11-1	
15	15	1	1	1	1		

Minter	A	B	C	D	Binario
5 7	0	1	-	1	-1-1 ✓
13 15	1	1	-	1	
5 13	-	1	0	1	-1-1 ✓
7 15	-	1	1	1	

Simplificación	1	5	6	7	10	11	12	13	15
★ S1	✖	✖							
★ S2			✖	✖					
★ S3					✖	✖			
★ S4							✖	✖	
S5						✖			✖
★ S6		✖		✖				✖	✖

$$F(A,B,C,D) = \overline{A}\,\overline{C}\,D + \overline{A}\,BC + A\,\overline{B}\,C + A\,B\,\overline{C} + BD$$