Álgebra de Conmutación

Contenidos

- Álgebra de boole. Teoremas básicos
- Expresiones de conmutación. Formas canónicas
- Representación y simplificación de funciones.
  - Mapas de Karnaugh
  - Método Quine-McCluskey
- Conclusiones

Álgebra de boole. Teoremas básicos

Álgebra de Boole

- Introducida en 1854 por el matemático George Boole
- El álgebra de boole son las matemáticas de los sistemas digitales.
- Definición:
  - Es una estructura algebraica (B, +, *) formada por un conjunto B y dos operaciones definidas sobre B (suma y producto) que cumple unos determinados axiomas.

Axiomas del álgebra de boole

- El conjunto **B es cerrado** respecto a + y *

$$x+y \in B \qquad \forall\, x,y \in B$$
$$x*y \in B$$

- Existe un **elemento identidad** para ambas operaciones

$$x+0=x \qquad \forall\, x \in B$$
$$x*1=x$$

- Ambas operaciones son **conmutativas**

$$x+y=y+x \qquad \forall\, x,y \in B$$
$$x*y=y*x$$

Axiomas del álgebra de boole

- Cada operación es **distributiva** respecto a la otra

$$x*(y+z)=(x*y)+(x*z) \qquad \forall\, x,y \in B$$
$$x+(y*z)=(x+y)*(x+z)$$

- Existe un **elemento complementario**

$$x+x'=1 \qquad \forall\, x \in B$$
$$x*x'=0$$

- Principio de **dualidad**:
  - Sea una igualdad entre dos expresiones booleanas, si se intercambian los operadores la igualdad se mantiene
- x + 1 = 1 => x * 0 = 0 (dualidad)
- **Idempotencia**:
  - x + x = x => x * x = x
- **Involución**:
  - (x')' = x

- **Absorción**:
  - x + x * y = x
  - x * (x + y) = x
- + y * son **asociativas**:
  - x + (y + z) = (x + y) + z
  - x * (y * z) = (x * y) * z
- **Leyes de Morgan**:
  - (x + y)' = x' * y'
  - (x * y)' = x' + y'

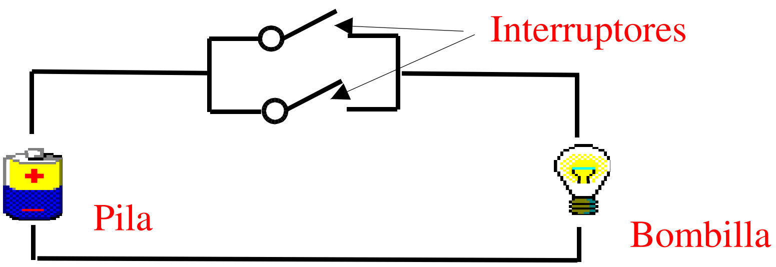
Álgebra de Conmutación

- Es un álgebra de boole en la que
  - B = {0, 1}   (bivalente)
  - + => suma lógica (OR)
  - * => producto lógico (AND)

X Y	X + Y	X * Y
0 0	0	0
0 1	1	0
1 0	1	0
1 1	1	1

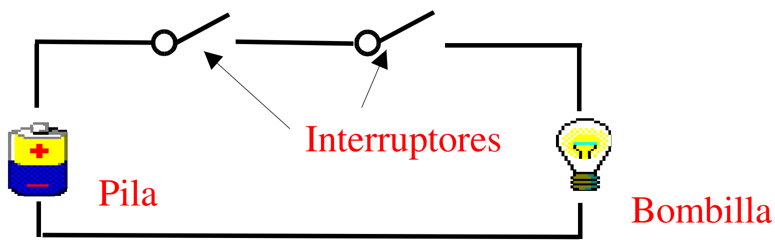
X	X'
0	1
1	0

Álgebra de Conmutación



Circuito eléctrico equivalente a la operación OR

Circuito eléctrico equivalente a la operación AND



Expresiones de Conmutación. Formas Canónicas

Variables y Funciones Booleanas

- Variable booleana:
  - la que representa cualquier elemento del conjunto B sobre el que se ha definido el álgebra de boole.
- Función booleana:
  - Expresión algebraica de variables booleanas con las operaciones +, * y complemento
  - F: B^n ------------> B , donde $(x_1, x_2, ..., x_n) \in B$

    $(x_1, x_2, ..., x_n)$ -----> X $X \in B$
  - Ejemplo: F(x, y, z) = xy +y'z + x'yz

Representación de Funciones Booleanas

- 2 formas de representación:
  - expresión algebráica
    - ejemplo: F(x,y) = x + y
  - tabla de verdad
    - tabla con los valores resultantes de todas las posibles combinaciones de valores de las variables.
    - Ejemplo:
      - F(0,0) = 0 + 0 = 0
      - F(0,1) = 0 + 1 = 1
      - F(1,0) = 1 + 0 = 1
      - F(1,1) = 1 + 1 = 1

X Y	X + Y
0 0	0
0 1	1
1 0	1
1 1	1

Representación de Funciones Booleanas

- Funciones lógicas equivalentes => Tienen la misma tabla de verdad.
- Ejemplo:
  - F(x,y) = x + y
  - G(x,y) = (x'*y')'

X Y	F	G
0 0	0	0
0 1	1	1
1 0	1	1
1 1	1	1

Las funciones son equivalentes

Representación de Funciones Booleanas

- Término canónico: término en el que aparecen todas las variables (complementadas o no)
  - producto canónico $\rightarrow$ minterm (ej: x*y'*z)
  - suma canónica $\rightarrow$ maxterm (ej: x'+y+z)

X Y Z	Minterms	Maxterms
0 0 0	x' y' z' m_0	x'+y'+z' M_0
0 0 1	x' y' z m_1	x'+y'+z M_1
0 1 0	x' y z' m_2	x'+y+z' M_2
0 1 1	x' y z m_3	x'+y+z M_3
1 0 0	x y' z' m_4	x+y'+z' M_4
1 0 1	x y' z m_5	x+y'+z M_5
1 1 0	x y z' m_6	x+y+z' M_6
1 1 1	x y z m_7	x+y+z M_7

Teorema de Shannon

- Cualquier función booleana puede expresarse como:
  - $F(x_1, \ldots, x_n) = x_1\, F(1, x_2, \ldots, x_n) + x_1'\, F(0, x_2, \ldots, x_n)$
  - $F(x_1, \ldots, x_n) = [x_1 + F(1, x_2, \ldots, x_n)]\, [x_1' + F(0, x_2, \ldots, x_n)]$
- Repitiendo el proceso => forma canónica de una función
  - ejemplo: función F(x,y) de 2 variables

$$
\begin{aligned}
F(x,y) &= x\,F(1,y) + x'\,F(0,y) = \\
&= xyF(1,1) + xy'F(1,0) + x'yF(0,1) + x'y'F(0,0) = \\
&= m_3F(1,1) + m_2F(1,0) + m_1F(0,1) + m_0F(0,0)
\end{aligned}
$$

Teorema de Shannon

- En general
  - $F(x1, x2, ..., xn) = a_0 m_0 + a_1 m_1 + ... + a_n m_n = \sum^{n} a_i m_i$
    - Expresión en forma de Suma de Productos
  - $F(x1, x2, ..., xn) = a_0 M_0 x a_1 M_1 x ... x a_n M_n = \prod^{n} a_i M_i$
    - Expresión en forma de producto de sumas

Teorema de Shannon

- Ejemplo

X Y	Minterms	Maxterms	X + Y
0 0	m_0	M_0	0
0 1	m_1	M_1	1
1 0	m_2	M_2	1
1 1	m_3	M_3	1

Suma de productos => $F(X,Y) = \sum (m_1, m_2, m_3) = x'y + xy' + xy$

Producto de sumas => $F(X,Y) = \prod (M_0) = x + y$

Representación y simplificación de funciones

Simplificación de funciones

- 2 métodos
  - reducción algebraica
    - aplicando los teoremas del álgebra de boole
  - mapas de Karnaugh
    - método gráfico. Aplicable a funciones de hasta 6 variables
  - método Quine-McCluskey
    - tabular y sistemático. Fácilmente implementable en un computador.

Reducción Algebraica

- $F(x_1, ..., x_n)$ => $G(x_1, ..., x_n)$ equivalente pero
  - mínimo número de términos
  - mínimo número de variables por término
- Aplicar propiedades del álgebra de boole.
- Ejemplo: $F(x,y,z) = \Sigma(0,2,3,4,5,7)$

$$
\begin{aligned}
F(x,y,z) &= x'yz' + x'y'z' + xy'z' + x'yz + xyz + xy'z \\
&= x'yz' + x'y'z' + x'y'z' + xy'z' + x'yz + xyz + xyz + xy'z \\
&= x'z'(y+y') \quad + \; y'z'(x+x') \quad + yz(x+x') \quad + xz(y+y') \\
&= x'z' + y'z' + yz + xz
\end{aligned}
$$

Idempotencia
Factor común
Complementario

Reducción Algebraica

- Problema:
  - Pueden obtenerse varias expresiones irreductibles
  - Irreductible no implica mínima
  - Ejemplo: $F(x,y,z) = \Sigma(0,2,3,4,5,7)$

$F(x,y,z) = x'z' + y'z' + yz + xz$ Solución anterior

x'z' Yz Xy'

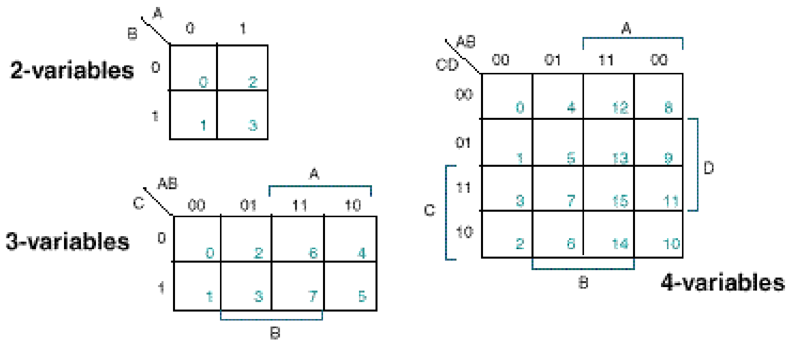
$F(x,y,z) = x'yz' + x'y'z' + xy'z' + x'yz + xyz + xy'z$

x'y y'z' Xz

$F(x,y,z) = x'y + y'z' + xz$ Soluciónes mínimas

$F(x,y,z) = x'z' + yz + xy'$

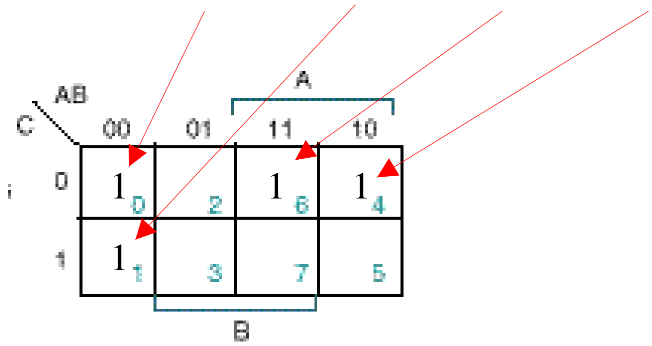
- Método de simplificación gráfico
- Número de celdas: equivale al número de términos de la tabla de verdad
- Codificación: 2 celdas adyacentes no pueden variar en más de un bit



Mapas de Karnaugh

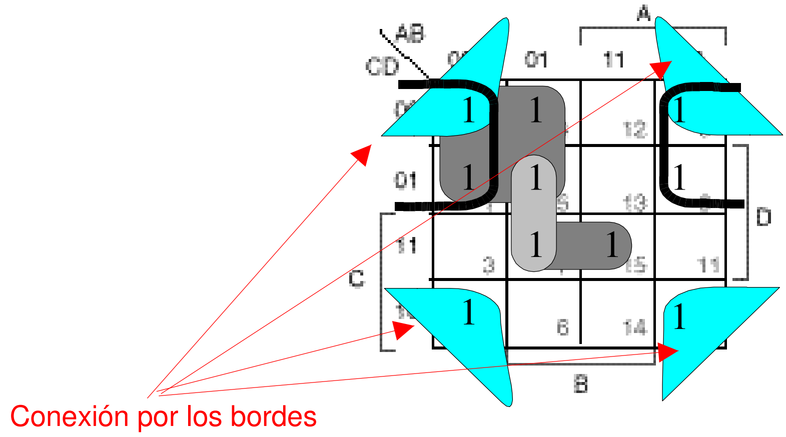
- Representación de funciones en el mapa
  - Asignar 1 a las celdas correspondientes al código de cada minterm. Asignar 0 al resto
  - Ejemplo: F(A,B,C) = A'B'C' + A'B'C + ABC' + AB'C'

000 001 110 100

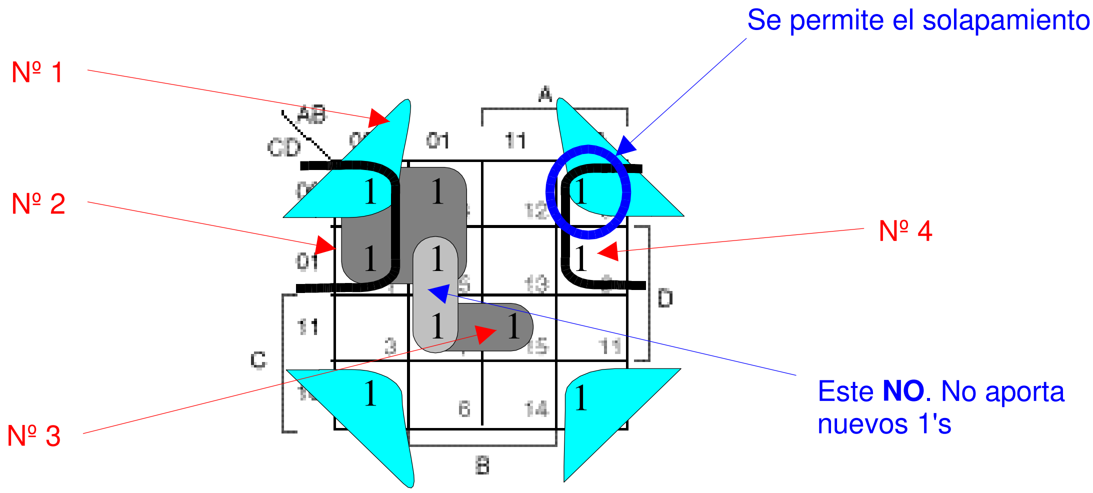


- Simplificación de funciones
  - Suma de productos:
    - Buscar grupos de 1's contiguos (no incluir 0's)
      - El número de 1’s en una agrupación es 2^n
      - Se permiten filas, columnas o rectángulos.
      - Se pueden conectar por los bordes.
    - Coger primero el grupo con más 1's.
    - Los demás grupos deben tomar al menos un 1 nuevo. Siempre agrupar el mayor nº de 1's, aunque haya solapes.
    - En cada grupo, las variables que cambian son las que desaparecen al simplificar
  - Producto de sumas: idem, pero con 0's

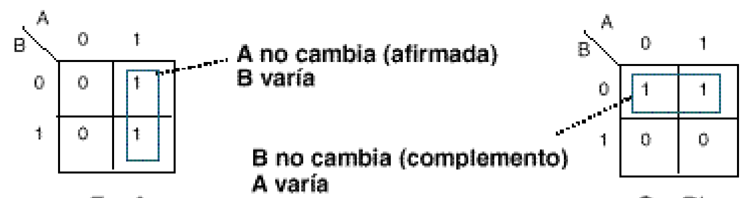
Grupos de 1's contiguos



Elección de los grupos



Expresión mínima



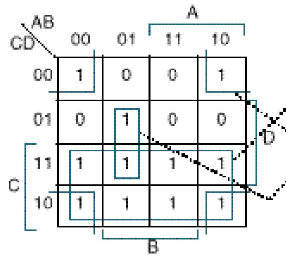
F = A

G = B'

Cout = A B + B Cin + A Cin

F(A,B,C) = A

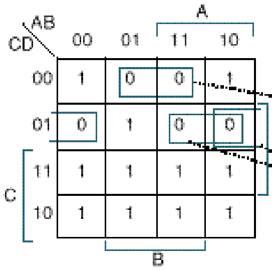
Expresión como suma de productos



$F(A,B,C,D) = \Sigma m(0,2,3,5,6,7,8,10,11,14,15)$

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

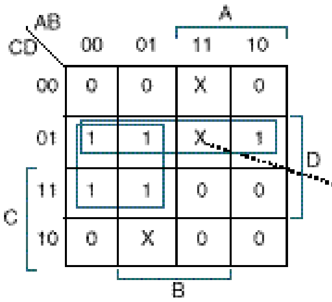
Expresión como producto de sumas



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

Don't Cares

Los Don't Cares pueden ser tratados 1's o 0's



$F(A,B,C,D) = \Sigma m(1,3,5,7,9) + \Sigma d(6,12,13)$

$F = A'D + B'C'D$ sin don't cares

$F = C'D + A'D$ con don't cares

Asignando $AB\overline{C}D$ como "1"

Quine-McCluskey

- Metodo sistemático, apropiado para su programación
- Propuesto por Quine en 1952 y mejorado por McCluskey
- Método en dos etapas
  - Identificar los implicantes primos
  - Determinar cuáles son esenciales

Quine-McCluskey: Obtención de Implicantes Primos

- Dada una tabla de verdad agrupar sus términos en función del número de 1's que contienen

$$F(d,c,b,a) = \sum(0, 1, 2, 5, 7, 8, 9, 10, 13, 15)$$

Tabla de verdad

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

Agrupando los minterms en función del número de unos →

	d	c	b	a	
0	0	0	0	0	Cero 1's
1	0	0	0	1	Un 1
2	0	0	1	0	
8	1	0	0	0	
5	0	1	0	1	Dos 1's
9	1	0	0	1	
10	1	0	1	0	
7	0	1	1	1	Tres 1's
13	1	1	0	1	
15	1	1	1	1	Cuatro 1's

Quine-McCluskey: Obtención de Implicantes Primos

- Reducción:
  - comparar uno a uno los términos de grupos adyacentes
    - los bits diferentes se sustituyen por '_'
    - los bits iguales se mantienen tal cual
    - sólo deben compararse los términos en los que haya 1 diferencia
    - los que no hayan sido combinados son primos
  - Los términos repetidos sólo se incluyen 1 vez
  - repetir el proceso hasta que no puedan reducirse más términos

Quine-McCluskey: Obtención de Implicantes Primos

	d	c	b	a	
0	0	0	0	0	✓
1	0	0	0	1	✓
2	0	0	1	0	✓
8	1	0	0	0	✓
5	0	1	0	1	✓
9	1	0	0	1	✓
10	1	0	1	0	✓
7	0	1	1	1	✓
13	1	1	0	1	✓
15	1	1	1	1	✓

Primera Reducción

	d	c	b	a	
0, 1	0	0	0	–	✓
0, 2	0	0	–	0	✓
0, 8	–	0	0	0	✓
1, 5	0	–	0	1	✓
1, 9	–	0	0	1	✓
2, 10	–	0	1	0	✓
8, 9	1	0	0	–	✓
8, 10	1	0	–	0	✓
5, 7	0	1	–	1	✓
5, 13	–	1	0	1	✓
9, 13	1	–	0	1	✓
7, 15	–	1	1	1	✓
13, 15	1	1	–	1	✓

Segunda Reducción

	d	c	b	a	
0, 1, 8, 9	–	0	0	–	P
0, 2, 8, 10	–	0	–	0	**Q**
0, 8, 2, 10	–	0	–	0	
1, 5, 9, 13	–	–	0	1	**R**
1, 9, 5, 13	–	–	0	1	
5, 7, 13, 15	–	1	–	1	**S**
5, 13, 7 15	–	1	–	1	

✓ : término combinado
A : implicante primo

Quine-McCluskey: Implicantes Primos Esenciales

- Seleccionar los implicantes primos que cubren todos los minitérminos de la función
  - implicante primo esencial: el que aparece en solitario en una columna cualquiera

I. Primos	Minterms de la función									
	0	1	2	5	7	8	9	10	13	15
P	X	X				X	X			
Q	X		X			X		X		
R		X		X			X		X	
S				X	X				X	X

No cubiertos por Q y S. Debe incluirse por lo tanto P, o bien R. Lo que dará lugar a dos soluciones distintas, pero mínimas

I. Primos Esenciales: {Q, S}

Quine-McCluskey: Función Resultante

```
                  | d c b a
----------------------------
 0, 1, 8, 9       | - 0 0 -   P
 0, 2, 8, 10      | - 0 - 0   Q
 0, 8, 2, 10      | - 0 - 0
----------------------------
 1, 5, 9, 13      | - - 0 1   R
 1, 9, 5, 13      | - - 0 1
----------------------------
 5, 7, 13, 15     | - 1 - 1   S
 5, 13, 7 15      | - 1 - 1
```

Posibles soluciones:

F(d,c,b,a) = P + Q + S = C'B' + C'A' + CA
F(d,c,b,a) = Q + R + S = C'A' + B'A + CA

Conclusiones

- Álgebra de boole: proporciona la base teórica para la manipulación de funciones booleanas.
- Simplificación de funciones
  - reducción algebraica: la calidad de la solución depende de la experiencia del diseñador
  - mapas de Karnaugh: método visual, apropiado para funciones de entre 2 y 4 variables
  - Quine-McCluskey: método sistemático. El único adecuado para trabajar con muchas variables. Es el utilizado en los computadores