



UNIDAD 2

Factorización-Shannon-Consenso-Reed Müller.

86:44 Técnica Digital Avanzada
Profesor: Ing. Miguel Antonio Martínez.

Síntesis de circuitos.

En este capítulo veremos el problema de diseñar circuitos de dos niveles y cuando la cantidad de variables de entrada es grande.

La idea final de cualquier diseño de un circuito digital es poder plasmarlo en un circuito físico real conformado por integrados comerciales o en un circuito integrado fabricado especialmente para el fin.

En cualquier problema de síntesis debemos considerar varios puntos críticos, los podemos resumir en tres:

- a) **Especificaciones.** Tienen que ser completas y poder ver todas las posibilidades para proporcionar a los algoritmos de diseño el mayor espacio de búsqueda posible.
- b) **Restricciones.** Estas pueden referirse a la velocidad, la capacidad de prueba, los tipos disponibles de encapsulado, disipación de energía, confiabilidad, etc.
- c) **Costo.** Debe tener en cuenta tantos factores como se posible: costo de fabricación del chip, costo de probarlo, costo del encapsulado, etc. Cada componente del costo depende de muchos factores, que a menudo son difíciles de estimar a partir de las especificaciones.

Por ejemplo, a menudo la disipación de potencia está fuertemente relacionada con el retardo del circuito, por lo que al buscar un circuito más rápido estoy dispuesto a incurrir en una mayor disipación de potencia. A veces esto es una restricción pues en equipos portátiles, en sensores inalámbricos o en sistemas de telecomunicación necesitamos un bajo consumo. En estos momentos la solución es de compromiso, buscando un punto óptimo donde se establezcan variables aceptables de consumo y velocidad.

Hasta ahora implementamos todo en dos niveles, que son los dos mínimos necesarios para generar una función booleana. O sea que nuestras primitivas son compuertas AND y OR. Los inversores no los contamos como un nivel más. También podríamos usar todas compuertas NAND o todas NOR aplicando adecuadamente las leyes de De Morgan. También hay factibles realizaciones usando compuertas XOR.

Hay dos razones principales por las que podemos querer implementar un circuito en dos niveles en lugar de múltiples niveles.

- a) **Velocidad.**
- b) **Sencillez.**

El retraso de un circuito depende de varios factores. Los números de etapas lógicas que una señal debe atravesar es una de las más importantes. Entonces, las implementaciones de dos niveles tienden a ser rápidas. Tenga en cuenta, sin embargo, que reducir el número de niveles puede aumentar el fan-in y el fan-out. Esto puede afectar negativamente la velocidad.

Las redes de dos niveles son más fáciles de diseñar y analizar, porque el espacio de la solución está muy restringido y son más fáciles de implementar. Estos circuitos han sido muy populares porque se conocen procedimientos de diseño sistemáticos y efectivos. Con el advenimiento de lógicas programables como las FPGA, la síntesis se concentró en celdas estándar, donde es común usar más de dos niveles.

Al minimizar una lógica para una implementación posterior de varios niveles, la función de costo intenta guiar el proceso de optimización hacia una función que pueda ser **factorizada** fácilmente. Minimizando el número de compuertas y el número de entradas de cada compuerta normalmente proporciona un buen punto de partida.

La **Factorización** es la descomposición de un problema complejo en un número de subproblemas más simples de resolver. Esta es una actividad usual en ingeniería.

En síntesis, lógica es una actividad fundamental, ya que separa un sistema lógico de un número elevado de entradas en un conjunto de subsistemas interconectados con un número menor de variables de entrada.

La descomposición funcional de los circuitos combinacionales influye poderosamente en la disminución de costos de implementación de los sistemas digitales. Y tiene su fundamento en que las tablas de verdad de funciones combinacionales, de un número elevado de variables, puede contener redundancias, las que pueden ser eliminadas por la descomposición de la función en varias funciones independientes con menos variables de entrada.

Desde el punto de vista matemático la descomposición es el proceso de expresar una función de n variables como una función de funciones con variables menores a n.

Hay muchas formas de descomponer o factorizar una función lógica. Una de ellas es la que nos ofrece el **Teorema de Shannon**.

Este **teorema** dice que cualquier función de conmutación de n variables puede descomponerse de la siguiente forma:

$$F(x_{n-1}, \dots, x_1, x_0) = \bar{x}_0 \cdot F(x_{n-1}, \dots, x_1, \mathbf{0}) + x_0 \cdot F(x_{n-1}, \dots, x_1, \mathbf{1})$$

Es la descomposición de Shannon con respecto a la variable x_0 . Las funciones que acompañan a x_0 y \bar{x}_0 se llaman **Cofactores**.

Si se van descomponiendo los cofactores de la misma manera para x_1 , se obtiene una descomposición de Shannon con respecto a x_0 y x_1 . Queda de la siguiente manera:

$$F = \bar{x}_1 \cdot \bar{x}_0 \cdot F(\dots, 0, 0) + \bar{x}_1 \cdot x_0 \cdot F(\dots, 0, 1) + x_1 \cdot \bar{x}_0 \cdot F(\dots, 1, 0) + x_1 \cdot x_0 \cdot F(\dots, 1, 1)$$

Iterando esta descomposición para todas las variables se obtiene finalmente una expresión F como suma de términos **canónicos o minitérminos**.

Como ejemplo de esto desarrollaremos la descomposición de Shannon para una función de tres variables llevándola a la forma canónica. La función es la siguiente:

$$F(A,B,C) = A + B$$

La descomposición de Shannon para tres variables queda:

$$F(A,B,C) = \bar{A}.\bar{B}.\bar{C}.F(0,0,0) + \bar{A}.\bar{B}.C.F(0,0,1) + \bar{A}.B.\bar{C}.F(0,1,0) + \bar{A}.B.C.F(0,1,1) + A.\bar{B}.\bar{C}.F(1,0,0) + A.\bar{B}.C.F(1,0,1) + A.B.\bar{C}.F(1,1,0) + A.B.C.F(1,1,1)$$

Reemplazando la función dada A + B en esta última fórmula general vemos que los dos primeros términos son iguales a cero, los demás nos quedan:

$$F = \bar{A}.B.\bar{C} + \bar{A}.B.C + A.\bar{B}.\bar{C} + A.\bar{B}.C + A.B.\bar{C} + A.B.C$$

Esta función se puede escribir como:

$$F(A,B,C) = \sum m(2, 3, 4, 5, 6, 7)$$

Se puede ver fácilmente que esta función canónica simplificada da la función original.

Lo mismo sucede si aplicamos este teorema a funciones que estén expresadas como PdS. En ese caso nos queda:

$$F(x_{n-1}, \dots, x_1, x_0) = (\bar{x}_0 + F(x_{n-1}, \dots, x_1, \mathbf{1})) \cdot (x_0 + F(x_{n-1}, \dots, x_1, \mathbf{0}))$$

Iterando esta descomposición para todas las variables se obtiene finalmente una expresión F como suma de términos **canónicos o Maxitérminos**.

Como ejemplo de esto desarrollaremos la descomposición de Shannon para una función de tres variables llevándola a la forma canónica. La función es la siguiente:

$$F(A,B,C) = A + B$$

La descomposición de Shannon para tres variables queda:

$$F(A,B,C) = (\bar{A} + \bar{B} + \bar{C} + F(1,1,1)) \cdot (\bar{A} + \bar{B} + C + F(1,1,0)) \cdot (\bar{A} + B + \bar{C} + F(1,0,1)) \cdot (\bar{A} + B + C + F(1,0,0)) \cdot (A + \bar{B} + \bar{C} + F(0,1,1)) \cdot (A + \bar{B} + C + F(0,1,0)) \cdot (A + B + \bar{C} + F(0,0,1)) \cdot (A + B + C + F(0,0,0))$$

Reemplazando la función dada A + B en esta última fórmula general vemos que los seis primeros términos son iguales a uno, los demás nos quedan:

$$F = (A + B + \bar{C}) \cdot (A + B + C)$$

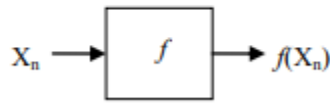
Esta función se puede escribir como:

$$F(A,B,C) = \prod M(0, 1)$$

Se puede ver fácilmente que esta función canónica simplificada da la función original.

Veremos un ejemplo teórico, tenemos una función de n variables $f(x_1, x_2, \dots, x_n)$, podemos Descomponerla de la siguiente manera.

$$f(x_1, x_2, \dots, x_{n-1}, x_n) = x_n' f(x_1, x_2, \dots, x_{n-1}, 0) + x_n f(x_1, x_2, \dots, x_{n-1}, 1)$$



Si definimos los cofactores como:

$$g_0(x_1, x_2, \dots, x_{n-1}) = f(x_1, x_2, \dots, x_{n-1}, 0)$$

$$g_1(x_1, x_2, \dots, x_{n-1}) = f(x_1, x_2, \dots, x_{n-1}, 1)$$

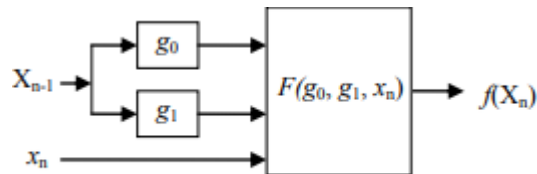
Podemos expresar:

$$f(x_1, x_2, \dots, x_{n-1}, x_n) = F(g_0, g_1, x_n)$$

Con:

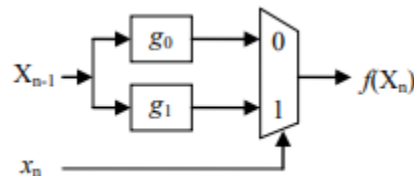
$$F(g_0, g_1, x_n) = x_n' g_0 + x_n g_1$$

Con lo cual nos queda un esquema como el siguiente:



W

Podemos usar un multiplexor de la siguiente manera:



Ahora haremos un ejemplo práctico, supongamos que queremos implementar un circuito empleando un multiplexor de tres entradas de selección y todos los multiplexores que hagan falta de dos entradas de selección, que realice la función lógica $F(x_1, x_2, \dots, x_6)$ que se caracteriza por tomar el valor "1" solo si se cumple que:

$$x_1 + x_2 + x_3 + 2x_4 + 2x_5 + 3x_6 \geq 4$$

Las operaciones de suma y producto son aritméticas. Para resolverlo vemos que tenemos 6 variables independientes, si quisiéramos plantear la tabla de verdad, la misma constaría de 64 filas. Esta función la podemos desarrollar según el teorema de expansión de Shannon de la siguiente manera:

$$\begin{aligned}
f(x_1, x_2, x_3, x_4, x_5, x_6) = & \bar{x}_4 \bar{x}_5 \bar{x}_6 f(x_1, x_2, x_3, 0, 0, 0) + \bar{x}_4 \bar{x}_5 x_6 f(x_1, x_2, x_3, 0, 0, 1) + \\
& + \bar{x}_4 x_5 \bar{x}_6 f(x_1, x_2, x_3, 0, 1, 0) + \bar{x}_4 x_5 x_6 f(x_1, x_2, x_3, 0, 1, 1) + \\
& + x_4 \bar{x}_5 \bar{x}_6 f(x_1, x_2, x_3, 1, 0, 0) + x_4 \bar{x}_5 x_6 f(x_1, x_2, x_3, 1, 0, 1) + \\
& + x_4 x_5 \bar{x}_6 f(x_1, x_2, x_3, 1, 1, 0) + x_4 x_5 x_6 f(x_1, x_2, x_3, 1, 1, 1)
\end{aligned}$$

Hemos expandido respecto de las variables x_4 , x_5 y x_6 por ser las más relevantes en la ecuación, pero se podría haber tomado cualquier otra. Ya podemos implementar un multiplexor de 3 entradas de control conectando a las mismas las tres variables mencionadas. Lo que queda es encontrar las expresiones de cada uno de los cofactores de la ecuación anterior. Vemos que, si en la ecuación reemplazamos x_4 , x_5 y x_6 por 0, 0, 0, independientemente de los valores de x_1 , x_2 y x_3 , la desigualdad no se cumple. Por lo tanto:

$$f(x_1, x_2, x_3, 0, 0, 0) = 0$$

Las siguientes funciones evalúan siempre 1, ya que la desigualdad se cumple siempre independientemente de los valores de x_1 , x_2 y x_3 .

$$f(x_1, x_2, x_3, 1, 1, 1) = 1$$

$$f(x_1, x_2, x_3, 0, 1, 1) = 1$$

$$f(x_1, x_2, x_3, 1, 0, 1) = 1$$

$$f(x_1, x_2, x_3, 1, 1, 0) = 1$$

De las tres funciones restantes podemos deducir que:

$$f(x_1, x_2, x_3, 0, 1, 0) = f(x_1, x_2, x_3, 1, 0, 0)$$

Por lo que solo debemos encontrar $f(x_1, x_2, x_3, 0, 0, 1)$ y $f(x_1, x_2, x_3, 0, 1, 0)$. Los mapas de Karnaugh resultantes son:

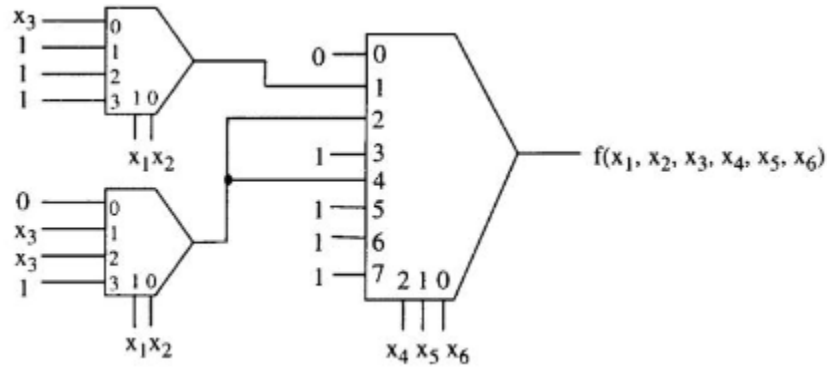
	x_1	x_2	00	01	11	10
x_3	0		0	1	1	1
	1		1	1	1	1

$f(x_1, x_2, x_3, 0, 0, 1)$

	x_1	x_2	00	01	11	10
x_3	0		0	0	1	0
	1		0	1	1	1

$f(x_1, x_2, x_3, 0, 1, 0)$

Eliendo como variables de control x_1 y x_2 resulta el circuito final:



Hemos visto que cuando tenemos funciones booleanas de muchas variables necesitamos **estrategias de descomposición** para poder separar funciones complejas en piezas más simples.

Más adelante veremos también **estrategias computacionales**, o sea formas de pensar las funciones booleanas que les permitan ser manipuladas por programas.

Suponiendo que tenemos una función $F(x_1, x_2, \dots, x_n)$. Se definen nuevas funciones si elegimos cualquier x_i y lo hacemos constante.

Por ejemplo:

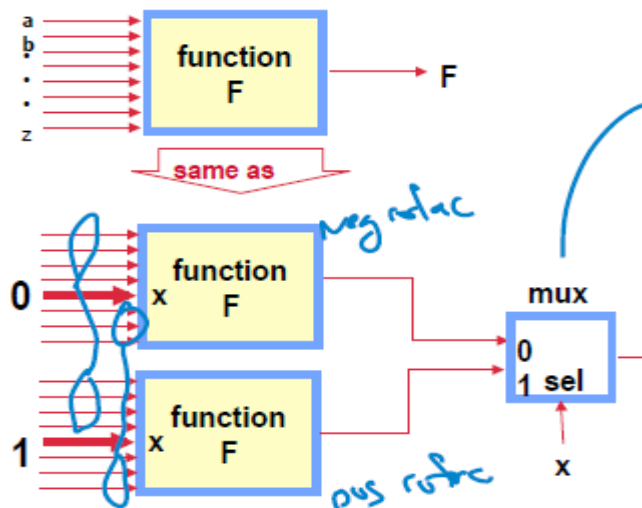
$$F(x_1, x_2, \dots, x_i = 0, \dots, x_n)$$

$$F(x_1, x_2, \dots, x_i = 1, \dots, x_n)$$

Normalmente al primero (cuando $x_i = 0$) lo llamamos **cofactor negativo de F**, y al segundo (cuando $x_i = 1$) lo llamamos **cofactor positivo de F**.

Podemos reescribir el teorema de expansión de Shannon diciendo que cualquier función y tomando cualquiera de las entradas x_i se puede representar como:

$$F(x_1, x_2, \dots, x_i, \dots, x_n) = x_i \cdot F(x_i = 1) + \bar{x}_i \cdot F(x_i = 0)$$



Ejemplo: Si tenemos una función de cuatro variables independientes

$$F(x,y,z,w) = x \cdot F(x=1) + x' \cdot F(x=0)$$

Si ahora expandimos cada cofactor respecto de la variable "y":

$$F(x=1) = y \cdot F(x=1,y=1) + \bar{y} \cdot F(x=1,y=0)$$

$$F(x=0) = y \cdot F(x=0,y=1) + \bar{y} \cdot F(x=0,y=0)$$

Finalmente nos queda:

$$F(x,y,z,w) = xy \cdot F(x=1,y=1) + x\bar{y} \cdot F(x=1,y=0) + \bar{x}y \cdot F(x=0,y=1) + \bar{x}\bar{y} \cdot F(x=0,y=0)$$

Que no es otra cosa que la expansión de la función F respecto de las variables "x" e "y".

Donde hemos escrito: $F(x_1, x_2, \dots, x_i = 1, \dots, x_j = 0, \dots, x_n)$ como $F_{x_i x_j'}$.

Obsérvese que también el orden no importa $(F_x)_y = (F_y)_x = F_{xy}$

En nuestro ejemplo nos queda que:

$$F(x,y,z,w) = xy \cdot F_{xy} + x'y \cdot F_{x'y} + xy' \cdot F_{xy'} + x'y' \cdot F_{x'y'}$$

Y recuerde que los cofactores son funciones, no números, en nuestro ejemplo:

$F_{xy} = F(x=1, y=1, z, w)$ es una **función booleana de "z" y de "w"**.

Ahora enunciaremos algunas propiedades de los cofactores:

Supongamos que tenemos dos funciones F(x) y G(x) donde $X = (x_1, x_2, \dots, x_n)$.

Ahora supongamos que creamos una nueva función H, de F y G, si decimos que:

- $H = F$
- $H = (F \cdot G)$ es decir, $H(x) = F(x) \cdot G(x)$
- $H = (F + G)$ es decir $H(x) = F(x) + G(x)$
- $H = (F \oplus G)$ es decir $H(x) = F(x) \oplus G(x)$

El planteo es que puedo decir de los cofactores de H, de F y de G:

El cofactor del complemento es el complemento del cofactor, o sea:

$$(F')_x = (F_x)'$$

El cofactor de la operación AND es igual a la AND de los cofactores:

$$(F \cdot G)_x = F_x \cdot G_x$$

El cofactor de la operación OR es igual a la OR de los cofactores:

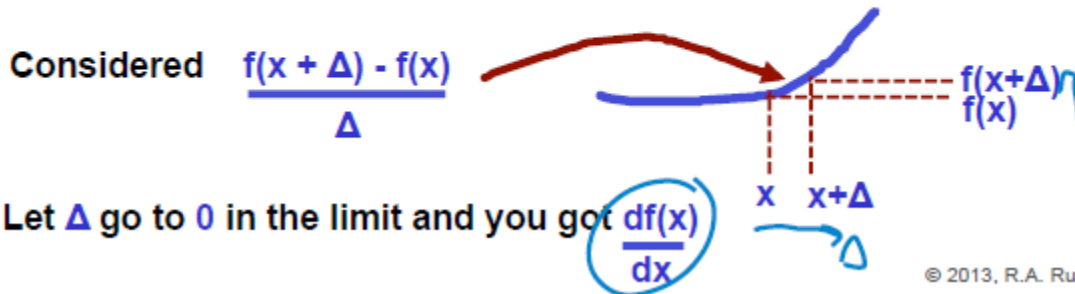
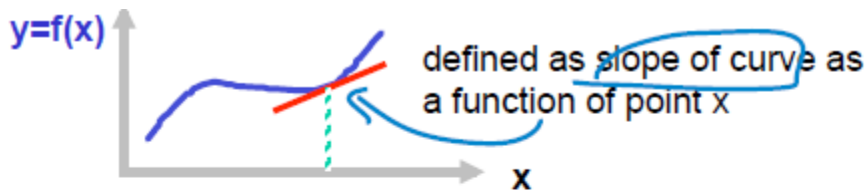
$$(F + G)_x = F_x + G_x$$

El cofactor de la operación EXOR es igual a la EXOR de los cofactores:

$$(F \oplus G)_x = F_x \oplus G_x$$

Por ejemplo, queremos ver qué pasa si hacemos operaciones con los mismos cofactores, por ejemplo si hago $F_x \oplus F_x$

Primero recordemos la definición de derivada para una función continua:



Considerando el cociente incremental, al hacer el límite cuando Δ tiende a cero decimos que tenemos la derivada de f respecto de x . Desde el punto de vista geométrico es la pendiente de la recta tangente en el punto considerado.

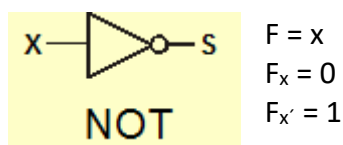
En el campo real, en una función $f(x)$, la derivada df/dx indica cómo cambia f para un cambio de x .

Para una función booleana valorada en 0, 1 no podemos cambiar x por un Δ pequeño. Solo puedo cambia 0 por 1 y 1 por 0, pero aún puedo preguntar cómo cambia f con x . Comparando el valor de $f(x)$ cuando $x = 0$ y cuando $x = 1$, vemos que siempre es igual a 1 si son diferentes. Por lo tanto, definimos la **derivada booleana como:**

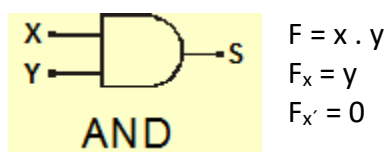
$$\frac{\partial f}{\partial x} = f_x \oplus f_x'$$

O sea, la derivada de una función booleana es la operación Exor de los cofactores de Shannon con respecto a la variable x .

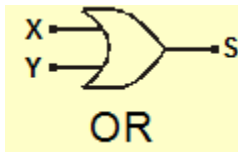
Algunos ejemplos prácticos de la función derivada



$$\frac{\partial F}{\partial x} = 0 \oplus 1 = 1$$



$$\frac{\partial F}{\partial x} = 0 \oplus y = y$$

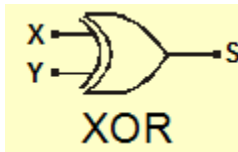


$$F = x + y$$

$$F_x = 1$$

$$F_{x'} = y$$

$$\partial F / \partial x = 1 \oplus y = \bar{y}$$



$$F = x \oplus y = \bar{x} \cdot y + x \cdot \bar{y}$$

$$F_x = \bar{y}$$

$$F_{x'} = y$$

$$\partial F / \partial x = \bar{y} \oplus y = 1$$

Ahora que hemos visto cómo se deriva una función booleana podemos definir la **Descomposición canónica de Reed-Muller**. Esta descomposición dice:
 “Cualquier función de conmutación F de n variables x se puede descomponer en la siguiente forma:

$$F(x_{n-1}, \dots, x_1, x_0) = F(x_{n-1}, \dots, x_1, 0) \oplus x_0 \cdot \partial F / \partial x_0$$

Es la descomposición de Reed-Muller con respecto de la variable x_0 . Iterando esta descomposición para todas las variables se obtiene finalmente una expresión de F como suma de cubos en los cuales todas las variables presentes están en forma **no negada**.

También se puede obtener esta expansión aplicando las identidades

$$\bar{X} = 1 \oplus X$$

$$X + Y = X \oplus Y \oplus X \cdot Y$$

Luego se puede simplificar la expresión recordando que:

$$X \oplus X = 0$$

$$0 \oplus X = X$$

Ejemplo: Dada la función booleana representada por la siguiente tabla de verdad:

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

Si simplificamos esta función aplicando Mapa de Karnaugh o el método de Quine-McCluskey nos queda:

$$F = A \cdot \bar{C} + B \cdot \bar{C} + \bar{A} \cdot \bar{B} \cdot C$$

Aplicando las identidades mencionadas más arriba:

$$\begin{aligned}
 F &= A.(1 \oplus C) + B.(1 \oplus C) + (1 \oplus A).(1 \oplus B).C \\
 F &= (1 \oplus C).(A + B) + (1 \oplus A).(1 \oplus B).C \\
 F &= \bar{C} . (A + B) + \bar{A} . \bar{B} . C \\
 F &= \bar{C} . (A + B) + \overline{(\bar{A} + \bar{B})} . C \\
 F &= (A \oplus B) \oplus C \\
 \mathbf{F} &= \mathbf{A \oplus B \oplus C \oplus A.B}
 \end{aligned}$$

Esta descomposición (ideada por los matemáticos Irving Reed y David Muller) permite expresar cualquier función booleana como una **suma exclusiva de productos**, donde no aparece ninguna variable negada. Cuando la cantidad de variables independientes es muy grande, generalmente, se usan métodos matriciales para poder lograr esta descomposición.

Este método es moderno y muy usado en detección y corrección de errores en transmisión de la información. En funciones booleanas se usa cuando la implementación con exor es más eficiente que la simplificación tradicional.

Teorema del consenso.

Este teorema nos permite otra forma de encontrar implicantes primos de una función booleana. Antes de continuar con este razonamiento veremos algo de lógica ya que el consenso entre términos de una función está relacionado con los silogismos, palabra definida por Aristóteles.

En lógica, la expresión $x \Rightarrow y$ se lee **x implica y**, es una expresión que es verdadera si **y** es verdadera siempre que **x** sea verdadera. Si **x** es falsa, la proposición es verdadera independientemente del valor de **y**.

Redondeando lo dicho en una tabla queda:

x	y	$x \Rightarrow y$
V	V	V
V	F	F
F	V	V
F	F	V

También de la tabla se puede ver que:

$$x \Rightarrow y = \bar{x} + y$$

El método tabular para encontrar Implicantes Primos.

Se nos da una función inicial como suma de productos y queremos encontrar una función que sea la suma de todos los implicantes primos de la función representada por la función inicial.

Una forma de lograr nuestro objetivo es expresar primero la función en **forma canónica**. (Recordar que una función canónica es aquella que en todos los términos figuran todas las variables independientes, negadas o nó).

Una vez expresada la función en forma canónica, se buscan todos los pares de términos que sean **adyacentes**, es decir los pares de términos donde se pueda aplicar el consenso. Recordar que dos términos son adyacentes si entre ellos cambia una sola variable. Los términos de consenso son claramente **implicantes de la función, aunque no necesariamente primos**. Todos los términos que se usaron para formar estos nuevos términos son incluidos en los nuevos términos, y por lo tanto **no son primos**. Los marcarnos como tales.

Ahora tomamos los nuevos términos y repetimos el proceso. Solo consideramos pares de términos que difieran exactamente en una variable, que debe aparecer complementada en un término y no complementada en el otro.

El proceso se repite hasta que no se puedan encontrar más términos de consenso. Todos los términos que son absorbidos (o contenidos) por los nuevos términos están marcados. Finalmente, los términos que no están marcados constituyen todos los **implicantes primos de la función**.

Para calcularlos a mano es mejor usar una tabla como la mostrada en la figura:

$w'x'y'z'$ ✓	$w'x'y'$ ✓	$x'y'$
	$w'x'z'$ ✓	$x'z'$
	$x'y'z'$ ✓	
$w'x'y'z$ ✓	$x'y'z$ ✓	
$w'x'yz'$ ✓	$x'yz'$ ✓	
$wx'y'z'$ ✓	$wx'y'$ ✓	
	$wx'z'$ ✓	
$wx'y'z$ ✓	$wy'z$	
$wx'yz'$ ✓	wyz'	
$wxyz'$ ✓	wxy	
$wxy'z$ ✓	wxz	
$wxyz$ ✓		

Esta tabla corresponde a la función:

$$f = x'y' + wxy + x'yz' + wy'z$$

Esta función llevada a su forma canónica queda:

$$f = w'x'y'z' + w'x'y'z + w'x'yz' + wx'y'z' + wx'y'z + wx'yz' + wxyz' + wxy'z + wxyz.$$

Los minitérminos que aparecen en esta última ecuación se ingresan a la tabla en la columna de la izquierda. Se agrupan como en el método de Quine-McCluskey, o sea

contando la cantidad de unos. Hacemos las comparaciones entre términos adyacentes. En nuestro ejemplo si comparamos $wxy\bar{z}$, del cuarto grupo con $wxyz$. Vemos que el término **de consenso es wxy . Ambos ($wxy\bar{z}$ y $wxyz$) son marcados. Estos dos implicantes no son primos** porque existe otro implicante que los contiene. Los resultados de la combinación de pares de términos mínimos adyacentes son implicantes con un literal menos que los minitérminos, estos se ingresan a la tabla en la columna del medio. Se repite el proceso hasta que no se puedan formar nuevos términos. En nuestro caso quedan 6 términos que no están marcados al final del proceso. Estos términos finales son los **implicantes primos de la función dada**.

$$wy'z, wyz', wxy, wxz, x'y', x'z'.$$

Haremos otro ejemplo de este método, para ello tomamos la función booleana de la página 4 de la unidad 1, la misma es:

$$F(A,B,C,D) = \sum m (0, 2, 3, 6, 7, 8, 9, 10, 13)$$

Haciendo el gráfico explicado anteriormente, nos queda:

$\overline{A}\overline{B}\overline{C}\overline{D}$	✓	$\overline{A}.\overline{B}.\overline{D}$ $\overline{B}.\overline{C}.\overline{D}$	✓	$\overline{B}.\overline{D}$
$\overline{A}\overline{B}C\overline{D}$ $A\overline{B}C\overline{D}$	✓ ✓	$\overline{A}.\overline{B}.C$ $\overline{A}.C.\overline{D}$ $\overline{B}.C.\overline{D}$ $A.\overline{B}.\overline{C}$	✓ ✓ ✓	$\overline{A}.C$
$\overline{A}.\overline{B}.C.D$ $\overline{A}.B.C.\overline{D}$ $A.\overline{B}.\overline{C}.D$ $A.\overline{B}.C.\overline{D}$	✓ ✓ ✓ ✓	$\overline{A}.B.D$ $A.\overline{C}.D$	✓	
$\overline{A}.B.C.D$ $A.B.\overline{C}.D$	✓ ✓			

El término $\overline{A}.\overline{B}.\overline{D}$ de la segunda columna no lo tenemos en cuenta ya que está absorbido por el término $\overline{B}.\overline{D}$ de la tercera columna. Vemos que los implicantes primos hallados coinciden con los encontrados en el apunte anterior mencionado. Recordar que los implicantes primos son los términos no marcados en el cuadro.

El método tabular que vimos antes se basa en el **Teorema del consenso**:

$$X . y + X . \bar{y} = X$$

Acá surge claramente que X es el término de consenso entre Xy y $X\bar{y}$ y contiene a ambos. El uso del método tabular visto anteriormente es simple y puede evitar muchas comparaciones. El inconveniente es que requiere la función expresada en su forma canónica. Queremos evitar tener que expandir la función en minitérminos para mayor

eficiencia. Por eso buscamos un enfoque diferente, basando en la forma general del teorema del consenso. Definimos **suma completa** a aquella suma de productos que contenga todos los implicantes primos de la función.

Podemos decir que una función expresada como suma de productos es una **suma completa** si:

- Ningún término incluye ningún otro término.
- El consenso de dos términos cualesquiera no existe o está contenido en algún otro término.

Ejemplo:

Dada la función:

$$f = A.B + \bar{B}.C + B.C.D$$

Si comparamos el segundo término con el primero, aparece un término de consenso (A.C) en la ecuación:

$$f = A.B + \bar{B}.C + B.C.D + A.C$$

Si comparamos el tercer término (BCD) con el primero y segundo término. El último da un término de consenso (CD) que agregamos en la fórmula:

$$f = A.B + \bar{B}.C + B.C.D + A.C + C.D$$

Finalmente viendo el tercero y el quinto término, nos queda una suma completa de la siguiente forma:

$$f = A.B + \bar{B}.C + A.C + C.D$$

Recapitulando, si tenemos una función f , definida como una suma de productos:

$$f = p_1 + p_2 + \dots + p_k$$

Si se tiene: $p_i = a. p_j$ con $a \neq 1$, siendo a una de las variables de f , se **puede descartar p_i de f**

Esto debido a que $p_i + p_j = a. p_j + p_j = (a + 1). p_j = p_j$ lo cual puede anotarse como $p_i \Rightarrow p_j$

El método consiste en introducir términos implicados, de tal manera de eliminar los términos que los implican. Es decir, se introduce p_j en la suma de productos y se elimina p_i .

Si por ejemplo se tiene:

$$f = x.T + \bar{x}.y.T$$

Debido a que yT es el término de consenso entre los dos términos, puedo escribir:

$$f = x.T + \bar{x}.y.T + yT$$

Como se tiene que:

$$\bar{x}.y.T \Rightarrow y.T$$

Entonces, se puede cambiar el producto $\bar{x}.y.T$ por el implicado $y.T$, resultando:

$$f = x.T + y.T$$

Ejemplo:

Aplicando el método el consenso simplificar la siguiente expresión:

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

$$\text{Consenso entre } (a.\bar{b}.c \text{ y } a.b.c) = a.c$$

$$\text{Consenso entre } (b.\bar{c} \text{ y } a.b.c) = a.b$$

$$\text{Consenso entre } (b.\bar{c} \text{ y } \bar{a}.b.c) = \bar{a}.b$$

$$\text{Consenso entre } (a.b.c \text{ y } \bar{a}.b.c) = b.c$$

Si se suman los cubos hallados y se eliminan los cubos incluidos en otros, se obtiene una nueva expresión:

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

$$\text{Consenso entre } (b.\bar{c} \text{ y } a.c) = a.b$$

$$\text{Consenso entre } (b.\bar{c} \text{ y } b.c) = b$$

$$\text{Consenso entre } (a.c \text{ y } \bar{a}.b) = b.c$$

$$\text{Consenso entre } (a.b \text{ y } \bar{a}.b) = b$$

Si se suman los cubos hallados y se eliminan los cubos incluidos en otros, se obtiene una nueva expresión:

$$f = a.c + b$$

No hay consensos entre los cubos de esta última expresión con lo cual son los implicantes primos de f .

Si a esta misma función la simplifico por ceros, primero la expreso como producto de Maxitérminos:

$$g = (a+b+c).(a+b+\bar{c}).(\bar{a}+b+c)$$

Viendo el consenso $(b+c)$ entre el primer y tercer término

$$g = (a+b+c).(a+b+\bar{c}).(\bar{a}+b+c).(b+c)$$

Eliminando los términos implicantes me queda:

$$g = (a+b+\bar{c}).(b+c)$$

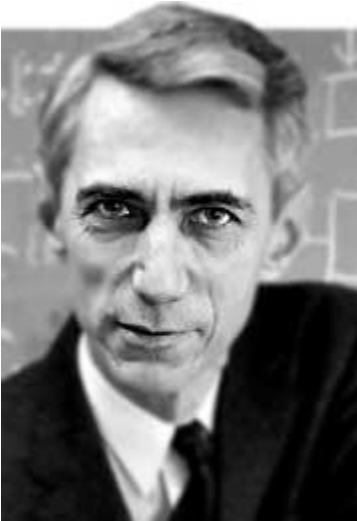
Viendo el consenso entre ambos términos $(a+b)$, nos queda:

$$g = (a+b+\bar{c}).(b+c).(a+b)$$

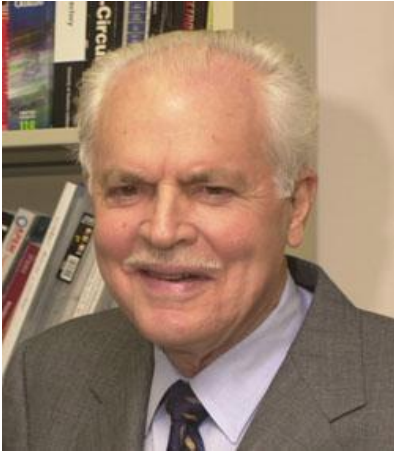
Eliminando el término implicante, nos queda la expresión final:

$$g = (a+b).(b+c)$$

Este método de consensos también es llamado método de Tison, en honor a Pierre Tison.



Claude Elwood Shannon. Nace el 30 de abril de 1916 en Potekey (EEUU) y muere el 24 de febrero de 2001 en Medford (EEUU). Fue un matemático, ingeniero eléctrico y criptógrafo estadounidense recordado como el padre de la teoría de la información. Con 21 años, mientras realizaba su maestría en el MIT demostró en su tesis que las aplicaciones electrónicas del algebra de Boole podrían construir cualquier relación lógica-numérica. En esta tesis demostró como el álgebra booleana se podía utilizar en el análisis y la síntesis de la conmutación y de los circuitos digitales.



Irving Stoy Reed. Nace el 12 de noviembre de 1923 en Seattle (EEUU) y muere el 11 de septiembre de 2012 en Washington (EEUU). Fue un matemático, ingeniero y profesor, conocido por ser el coinventor del código Reed-Muller. También colaboró en la invención del código Reed-Solomon. Hizo numerosas contribuciones al área de la ingeniería incluyendo radares, procesamiento digital de señales y procesamiento digital de imágenes. Era miembro de la United States National Academy of Engineering y Fellow del IEEE como ganador del Premio Claude Shannon.



David Eugene Muller. Nació el 2 de noviembre de 1924 en Texas (EEUU), muere el 27 de abril de 2008 en Nuevo México (EEUU). Fue un matemático e informático estadounidense. Fue profesor de matemáticas y ciencias de la computación en la Universidad de Illinois. Fue el inventor del elemento C Muller, un dispositivo utilizado para implementar circuitos asíncronos en computadoras.