

## Sumas de productos y productos de sumas

Las funciones lógicas se pueden expresar de dos formas diferentes, como una suma de productos o como un producto de sumas. La forma de obtener dicha expresión se basa en identificar que valores de salida tomarán el valor de 1 y el valor de 0.

Supongamos que se tiene una función lógica de tres variables  $x$ ,  $y$  y  $z$ , con la siguiente tabla de verdad:

$x$	$y$	$z$	$S$
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	0
1	1	1	1

1. Para expresar como **suma de productos**, se toman los minitérminos que hacen que la función devuelva 1, luego se suman los minitérminos.
2. Para expresar como **producto de sumas**, se toman los maxitérminos que hacen que la función devuelva 0, luego se multiplican los maxitérminos.

$x$	$y$	$z$	$S$	Minterminos	Maxiterminos
0	0	0	0	$m_0 = x'y'z'$	$M_0 = x + y + z$
0	0	1	1	$m_1 = x'y'z$	$M_1 = x + y + z'$
0	1	0	1	$m_2 = x'yz'$	$M_2 = x + y' + z$
0	1	1	0	$m_3 = x'yz$	$M_3 = x + y' + z'$
1	0	0	1	$m_4 = xy'z'$	$M_4 = x' + y + z$
1	0	1	0	$m_5 = xy'z$	$M_5 = x' + y + z'$
1	1	0	0	$m_6 = xyz'$	$M_6 = x' + y' + z$
1	1	1	1	$m_7 = xyz$	$M_7 = x' + y' + z'$

Con esto se tienen las formas canónicas de la función lógica, las cuales se pueden simplificar utilizando mapas de Karnaugh.

- **Suma de productos:**  $S = m_1 + m_2 + m_4 + m_7 = x'y'z + x'yz' + xy'z' + xyz$ ,
- **Producto de sumas:**  $S = M_0 \cdot M_3 \cdot M_5 \cdot M_6 = (x + y + z)(x + y' + z')(x' + y + z')(x' + y' + z)$ .

## Mapas de Karnaugh

Los mapas de Karnaugh son una herramienta útil que nos permite aplicar un método de simplificación de funciones lógicas.

El Mapa de Karnaugh tiene la característica de que puede ser visto como una representación bidimensional de una tabla de verdad. En la tabla de verdad, se colocan las variables por columnas y las combinaciones de tales variables determinan un valor de salida, 0 o 1, sin embargo, en el mapa las variables se colocan como si de un plano cartesiano se tratara, respetando cada una de las combinaciones que de ellas se generan, y colocando en la intersección de las combinaciones de las variables, el valor de salida.

		$zw$			
		00	01	11	10
$xy$	00	1	1	1	1
	01	1	0	0	0
	11	1	0	0	0
	10	1	1	1	0

Figure 1: Ejemplo de un mapa de Karnaugh

Los mapas muestran la relación que existe entre las entradas y las salidas de un circuito lógico, si se aplica adecuadamente el resultado será el más simplificado posible. Pueden ser utilizados para cualquier número de variables de entrada sin embargo se recomienda un máximo de seis variables.

### Representación de tabla de verdad en un mapa de Karnaugh

Las tablas de la verdad se pueden representar en un mapa de Karnaugh, para ello se deben seguir los siguientes pasos:

1. Se deben identificar los términos de la tabla de verdad que son 1.
2. Se deben identificar los términos de la tabla de verdad que son 0.
3. Se deben identificar los términos de la tabla de verdad que son 1 y que se pueden agrupar.
4. Se deben identificar los términos de la tabla de verdad que son 0 y que se pueden agrupar.

#### Observaciones:

- Los términos de la tabla de verdad que son 1 se representan en el mapa de Karnaugh con un 1, y sirven para expresar a la función como una suma de productos.
- Los términos de la tabla de verdad que son 0 se representan en el mapa de Karnaugh con un 0 y sirven para expresar a la función como un producto de sumas.

## Ejemplo de representación de tabla de verdad en un mapa de Karnaugh

Supongamos que se tiene la siguiente tabla de la verdad:

<b>x</b>	<b>y</b>	<b>z</b>	<b>w</b>	<b>S</b>
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

Para hacer un mapa de Karnaugh, primero se separan las variables  $xy$  y  $zw$  y se colocan en el mapa de la siguiente forma:

		$zw$			
		00	01	11	10
$xy$	00	-	-	-	-
	01	-	-	-	-
	11	-	-	-	-
	10	-	-	-	-

Figure 2: Mapa de Karnaugh para las variables  $xy$

Luego identificamos los términos de la tabla de verdad que son 1 y los colocamos en el mapa de Karnaugh. Por ejemplo esta línea de la tabla

<b>x</b>	<b>y</b>	<b>z</b>	<b>w</b>	<b>S</b>
0	0	0	1	1

Se coloca en el mapa de Karnaugh en la posición donde  $xy$  toma 00 y  $zw$  toma 01.

		$zw$			
		00	01	11	10
$xy$	00	0	1	0	0
	01	0	0	0	0
	11	0	0	0	0
	10	0	0	0	0

Figure 3: Ejemplo de representación

Pasar por cada línea de la tabla de verdad y colocar los términos en el mapa de Karnaugh. Los términos que no son 1 se colocan como 0 en el mapa de Karnaugh.

		$yx$			
		00	01	11	10
$wz$	00	0	1	0	1
	01	1	0	1	0
	11	0	1	0	1
	10	1	0	1	0

Figure 4: Mapa de Karnaugh completo del ejemplo.

## Agrupación de términos

Para simplificar la función lógica, se deben agrupar los términos que son 1 en el mapa de Karnaugh. Los términos se pueden agrupar siguiendo las siguientes reglas:

1. Los grupos deben contener una cantidad de elementos igual a potencias de 2. Es decir, los grupos solo podrán realizarse con cantidades de celdas iguales a: 1, 2, 4, 8, 16, 32, etc.
2. Se deben generar grupos con combinaciones de variables adyacentes, es decir, sólo debe de haber un cambio entre cada una de las combinaciones, por ejemplo la combinación  $ABC'$  es adyacente con  $AB'C'$ , ya que sólo cambia la variable  $B$ .
3. Grupos en las posiciones en los extremos y esquinas del mapa. Las posiciones de los extremos y las esquinas son adyacentes, por lo que se pueden agrupar. Esta adyacencia se debe a que el mapa de Karnaugh puede ser visto como un toroide.
4. Grupos de unos darán lugar a una suma de productos o mini términos.
5. Grupos de ceros darán lugar a un producto de sumas o maxi términos.
6. Los grupos deben ser lo más grande posible. Se buscará realizar grupos con la mayor cantidad de elementos posibles, entre más grande el grupo, se obtiene una función más simplificada.
7. No se pueden generar grupos en diagonal. Solo se permitirán grupos en vertical y horizontal dentro del mapa.
8. Puede existir solapamiento de grupos, siempre y cuando exista al menos un elemento que no haya sido agrupado anteriormente.
9. No deben existir grupos redundantes, es decir, no se puede realizar un grupo dentro de otro grupo, y no se puede realizar un grupo con elementos que ya hayan sido completamente agrupados en otros conjuntos.

Por ejemplo, si se tiene este mapa de Karnaugh:

		$yx$			
		00	01	11	10
$wz$	00	1	0	0	1
	01	1	0	0	1
	11	1	0	0	1
	10	0	0	0	0

Figure 5: Mapa de Karnaugh para agrupar.

Se debería agrupar de la siguiente forma:

		<i>zw</i>			
		00	01	11	10
<i>xy</i>	00	1	0	0	1
	01	1	0	0	1
	11	1	0	0	1
	10	0	0	0	0

Figure 6: Mapa de Karnaugh agrupado.

## Simplificación de la función lógica

Para la simplificación de la función lógica, se usan los grupos que se han formado en el mapa de Karnaugh. Cada grupo se convierte en un término de la función lógica, y se deben sumar los términos que se obtienen de los grupos, teniendo en cuenta lo siguiente:

1. Si un grupo es de una sola celda, se incluye igualmente en la función lógica.
2. En el grupo, las variables que cambian de valor "se me van" del término, y las que no cambian su valor permanecen.
3. Si un grupo se solapa con otro, no cambia la forma de trabajarlos, cada grupo es un término distinto.

Por ejemplo, tomando el mapa de Karnaugh anterior, se obtiene la siguiente función lógica:

$$F = x'w' + yw'$$

## Resumen

### Simplificación de términos

1. Se agrupan ya sean 1's o 0's, en potencias de 2.
2. Se pueden agrupar en vertical, horizontal o en todas las esquinas. Una forma de verlo es como un pacman, siempre las celdas por donde podría moverse son agrupables.
3. Para armar los términos, se mantienen las variables que no cambian de valor y se eliminan las que cambian.
4. Luego se suman o multiplican los términos obtenidos, dependiendo si se quiere expresar como suma de productos o producto de sumas.

### Método para expresar como suma de productos simplificada

1. Dibujar la tabla y luego el mapa de Karnaugh de la dimensión correcta, asignar nombre y valor a las filas y columnas.
2. Llenar los 1's y 0's donde correspondan (Véase la Figura 6 de ejemplo.)
3. Agrupar los 1's adyacentes en cuadrados o rectángulos lo mas grande posible.
4. Todos los 1's quedan agrupados si o sí, si no se pueden juntar con otros queda el grupo de una sola celda.
5. Representar cada término como un producto (dejando las variables que no cambian de valor) y sumarlos.
6. El resultado es la función expresada como **suma de productos**.

### Método para expresar como producto de sumas simplificada

1. Dibujar la tabla y luego el mapa de Karnaugh de la dimensión correcta, asignar nombre y valor a las filas y columnas.
2. Llenar los 1's y 0's donde correspondan.
3. Agrupar los 0's adyacentes en cuadrados o rectángulos lo mas grande posible, considerando que la cantidad de celdas debe ser potencia de 2.
4. Cada grupo de 0's se convierte en un término de la función lógica simplificada.
5. Representar cada término como una suma (dejando las variables que no cambian de valor) y multiplicarlos.

## PLA: Programmable Logic Array

Programmable Logic Array (PLA) es un dispositivo lógico de arquitectura fija con puertas AND programables seguidas de puertas OR programables. PLA es básicamente un tipo de dispositivo lógico programable que se utiliza para construir un circuito digital reconfigurable. La estructura de un PLA es la siguiente:

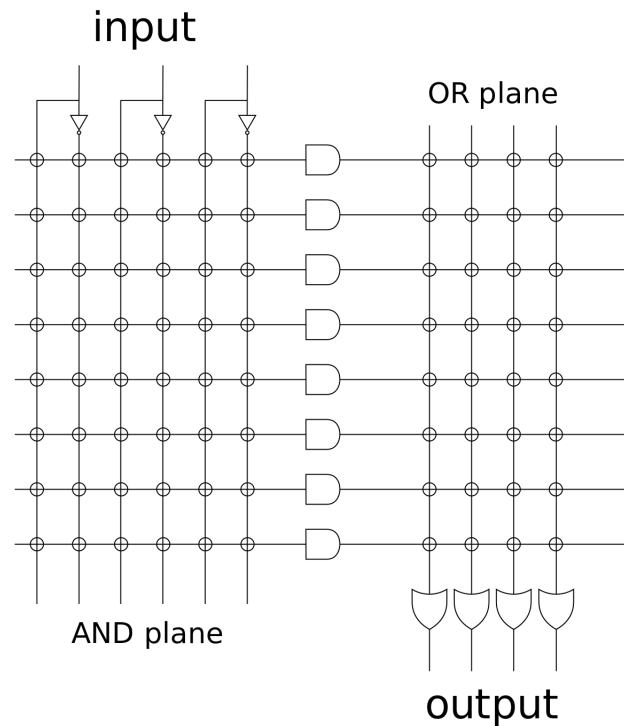


Figure 7: Estructura de un PLA

El funcionamiento se podría resumir en tres pasos:

1. **Programación:** el usuario define la función lógica que se desea implementar.
2. **Generación de términos del producto:** las entradas se aplican a la matriz de puertas AND para producir un conjunto de términos de producto.
3. **Generación de suma de términos:** los términos de producto se aplican a la matriz de puertas OR para producir la salida.

### Método para implementar una función lógica en un PLA

1. Una vez se tenga la función lógica simplificada, se deben identificar los términos de la función.
2. Cada término de la función se convierte en una fila de la matriz de puertas AND.
3. Luego se suman los términos de la función y se convierten en una fila de la matriz de puertas OR.
4. La salida de la matriz de puertas OR es la función lógica implementada en el PLA.



## Ejemplo completo de Lógica Combinacional

Un detector de paridad impar de 4 entradas y una salida funciona de la siguiente manera: si la cantidad de entradas con valor '1' es impar la salida se pone en '1', en el resto de los casos la salida toma valor '0'

- Construir la tabla de verdad para dicho sistema.
- Obtener la ecuación lógica como suma de minitérminos y producto de maxitérminos (funciones canónicas).
- Implementar el sistema con compuertas NAND de la cantidad de entradas requeridas.
- Implementar el sistema con una PLA.

### Punto a

Para construir la tabla de verdad, se deben considerar las 4 entradas posibles y la salida que se obtiene para cada combinación de entradas.

A	B	C	D	S	Minitérminos	Maxitérminos
0	0	0	0	0	$m_0 = A'B'C'D'$	$M_0 = A+B+C+D$
0	0	0	1	1	$m_1 = A'B'C'D$	$M_1 = A+B+C+D'$
0	0	1	0	1	$m_2 = A'B'CD'$	$M_2 = A+B+C'+D$
0	0	1	1	0	$m_3 = A'B'CD$	$M_3 = A+B+C'+D'$
0	1	0	0	1	$m_4 = A'BC'D'$	$M_4 = A+B'+C+D$
0	1	0	1	0	$m_5 = A'BC'D$	$M_5 = A+B'+C+D'$
0	1	1	0	0	$m_6 = A'BCD'$	$M_6 = A+B'+C'+D$
0	1	1	1	1	$m_7 = A'BCD$	$M_7 = A+B'+C'+D'$
1	0	0	0	1	$m_8 = AB'C'D'$	$M_8 = A'+B+C+D$
1	0	0	1	0	$m_9 = AB'C'D$	$M_9 = A'+B+C+D'$
1	0	1	0	0	$m_{10} = AB'CD'$	$M_{10} = A'+B+C'+D$
1	0	1	1	1	$m_{11} = AB'CD$	$M_{11} = A'+B+C'+D'$
1	1	0	0	0	$m_{12} = ABC'D'$	$M_{12} = A'+B'+C+D$
1	1	0	1	1	$m_{13} = ABC'D$	$M_{13} = A'+B'+C+D'$
1	1	1	0	1	$m_{14} = ABCD'$	$M_{14} = A'+B'+C'+D$
1	1	1	1	0	$m_{15} = ABCD$	$M_{15} = A'+B'+C'+D'$

### Punto b

Para obtener la ecuación lógica como suma de minitérminos y producto de maxitérminos, se deben identificar los términos que hacen que la función devuelva 1 y los términos que hacen que la función devuelva 0.

- Suma de minitérminos:** tomo los términos que hacen 1 la función y los sumo.

$$F = m_1 + m_2 + m_4 + m_7 + m_8 + m_{11} + m_{13} + m_{14}$$

Desarrollando la ecuación, se obtiene:

$$F = A'B'C'D + A'B'CD' + A'BC'D' + A'BCD + AB'C'D' + AB'CD + ABC'D + ABCD'$$

- Producto de maxitérminos:** tomo los términos que hacen 0 la función y los multiplico.

$$F = M_0 \cdot M_3 \cdot M_5 \cdot M_6 \cdot M_9 \cdot M_{10} \cdot M_{12} \cdot M_{15}$$

Desarrollando la ecuación, se obtiene:

$$F = (A+B+C+D)(A+B+C'+D')(A+B'+C+D')(A+B'+C'+D)(A'+B+C+D')(A'+B+C'+D')(A'+B'+C+D')(A'+B'+C'+D')$$

**Punto c**

Para implementar el sistema con compuertas NAND, se deben seguir los siguientes pasos:

1. Se toma la forma canónica de la función lógica, en este caso se tomará la forma de suma de minitérminos.
2. Luego se niega dos veces la función lógica, para obtener la forma de producto de maxitérminos.
3. Se aplica una vez más la ley de De Morgan para obtener la forma buscada.

$$\begin{aligned}
 F &= A'B'C'D + A'B'CD' + A'BC'D' + A'BCD + AB'C'D' + AB'CD + ABC'D + ABCD' \\
 &= (A'B'C'D + A'B'CD' + A'BC'D' + A'BCD + AB'C'D' + AB'CD + ABC'D + ABCD')'' \\
 &= (A'B'C'D)' \cdot (A'B'CD')' \cdot (A'BC'D')' \cdot (A'BCD)' \cdot (AB'C'D')' \cdot (AB'CD)' \cdot (ABC'D)' \cdot (ABCD')'
 \end{aligned}$$

**Punto d**

Para implementar el sistema con una PLA, se utilizará la forma canónica de la función lógica, en este caso se tomará la forma de suma de minitérminos.

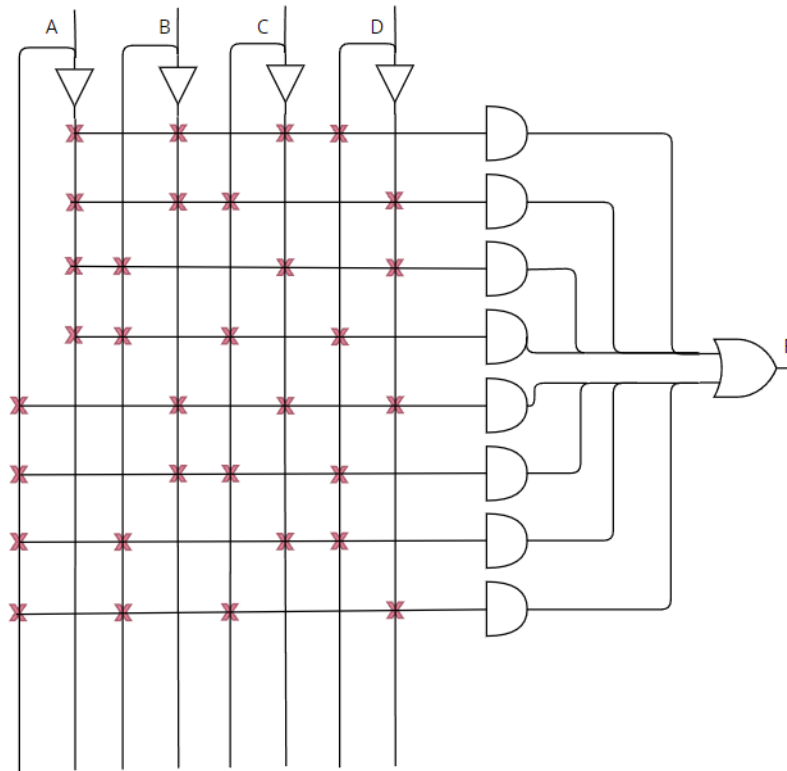


Figure 8: Estructura de la función lógica en una PLA.