Mapas de Karnaugh

Los Mapas de Karnaugh (también llamados K-Maps) son una técnica gráfica utilizada en electrónica digital para minimizar expresiones booleanas sin necesidad de aplicar algebraicamente los teoremas del álgebra de Boole.

Fueron desarrollados por Maurice Karnaugh en 1953 como una mejora del diagrama de Veitch. Son especialmente útiles para funciones con hasta 4 o 5 variables, ya que permiten visualizar patrones y simplificar funciones de forma rápida y sistemática.

Ventaja principal: Reducir el número de puertas lógicas necesarias en un circuito, lo que disminuye el costo, el consumo de energía y las posibilidades de error.

Primero revisemos que el el Código Gray

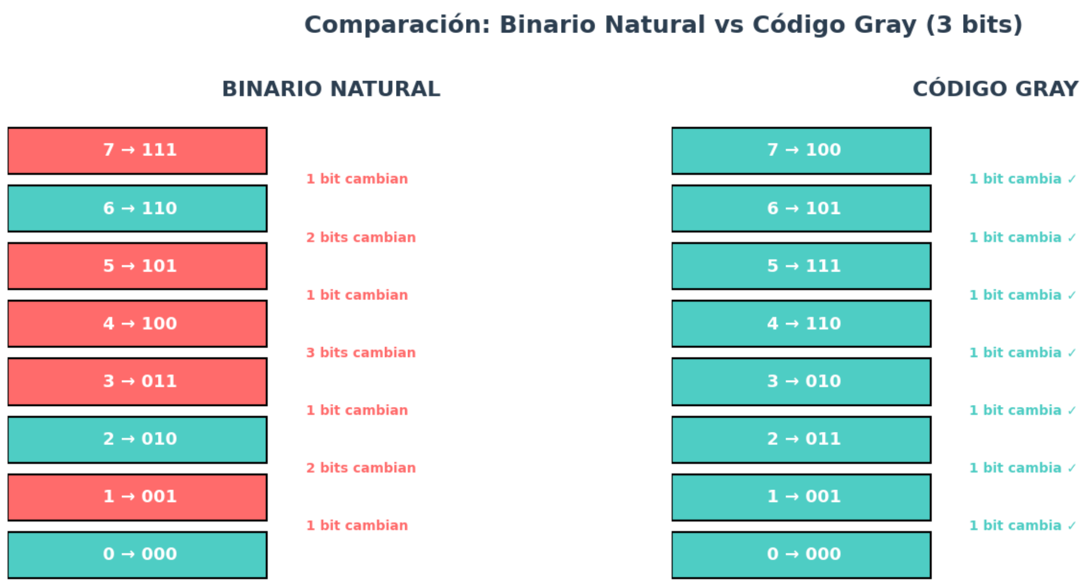
1. Código Gray

1.1. Definición y Propiedad Fundamental

El Código Gray (también llamado código binario reflejado) es un sistema de numeración binario donde dos valores consecutivos difieren en exactamente un solo bit. Esta propiedad, conocida como adyacencia unitaria, es la razón por la que Maurice Karnaugh lo adoptó para sus mapas en 1953.

> Propiedad clave: Entre dos números consecutivos en Gray, SOLO cambia UN bit. Nunca dos, nunca tres. Siempre uno.

1.2. Comparación: Binario Natural vs Código Gray

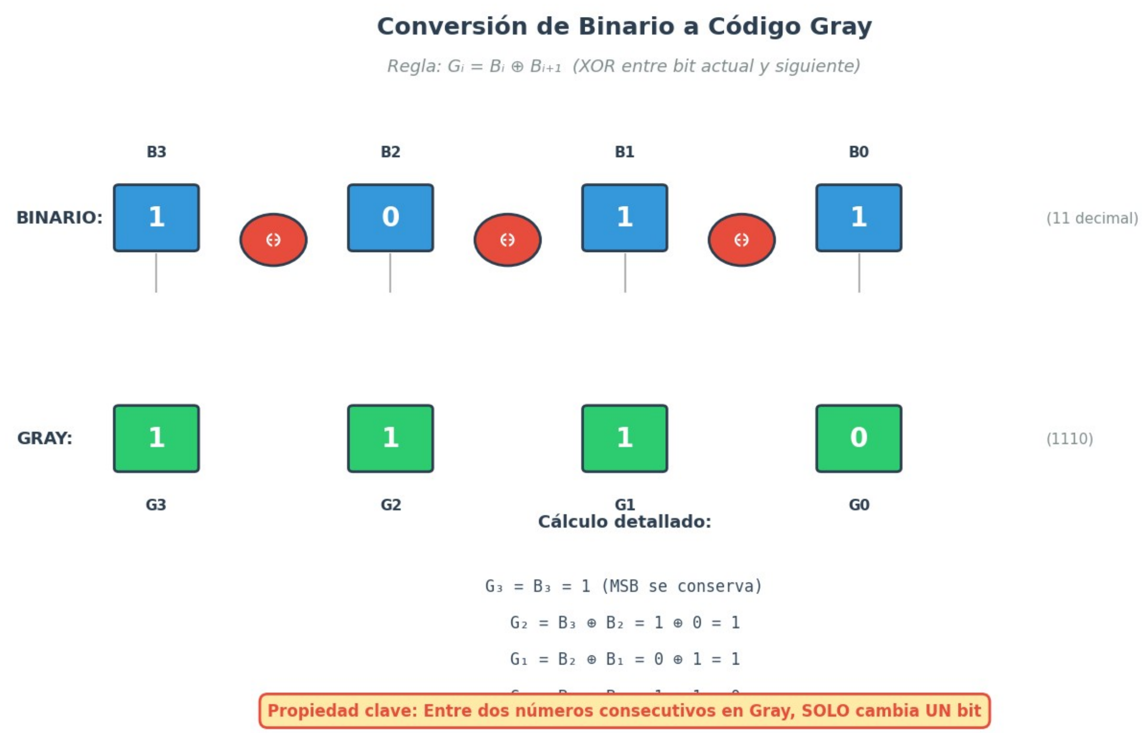


En la figura se observa claramente:

Transición	Binario Natural	Bits que cambian	Código Gray	Bits que cambian
3 → 4	011 → 100	**3 bits ✗**	010 → 110	**1 bit ✓**
1 → 2	001 → 010	**2 bits ✗**	001 → 011	**1 bit ✓**
7 → 8	111 → 1000	**4 bits ✗**	100 → 000	**1 bit ✓**

Conclusión: El binario natural falla en la transición 3→4 (todos los bits cambian), mientras que Gray mantiene siempre 1 bit de diferencia.

1.3. Conversión de Binario a Gray



Regla matemática:
$Gi=Bi\oplus Bi+1$

Donde:

- Gi = bit i del código Gray
- Bi = bit i del binario natural
- ⊕ = operación XOR (suma módulo 2)
- El bit más significativo (MSB) se conserva: $Gn-1=Bn-1$

Ejemplo paso a paso: Convertir binario **1011** (decimal 11) a Gray:

Paso	Operación	Resultado
MSB	$G3=B3$	**1**
Bit 2	$G2=B3\oplus B2=1\oplus 0$	**1**
Bit 1	$G1=B2\oplus B1=0\oplus 1$	**1**
Bit 0	$G0=B1\oplus B0=1\oplus 1$	**0**

Resultado: Gray = **1110**

1.4. Tabla Completa de Conversión (4 bits)

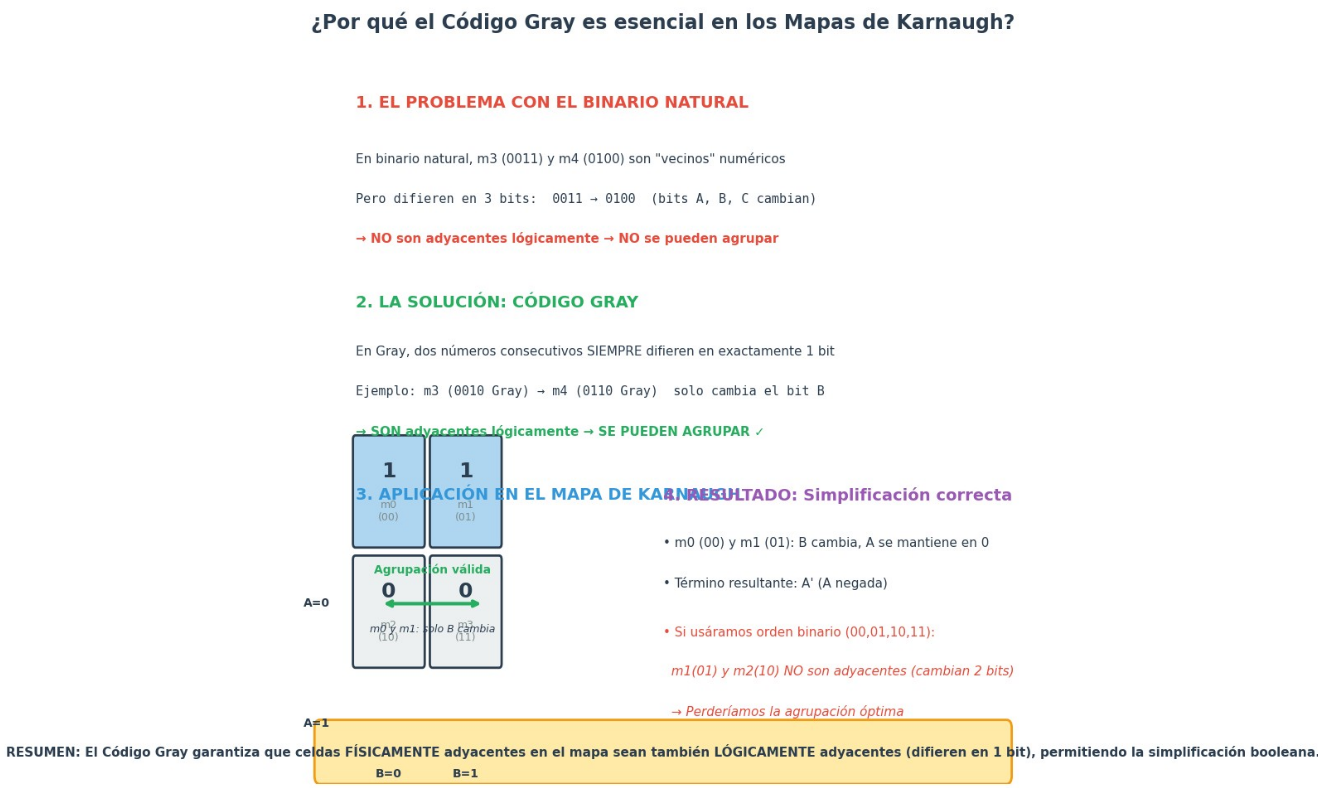
Decimal	Binario	Gray	Bits que cambian	Aplicación en K-Map
0	0000	0000	-	Fila superior
1	0001	0001	1	Fila superior
2	0010	0011	1	Fila superior
3	0011	0010	1	Fila superior
4	0100	0110	1	Fila superior
5	0101	0111	1	Fila superior
6	0110	0101	1	Fila superior
7	0111	0100	1	Fila superior
8	1000	1100	1	Columnas K-Map
9	1001	1101	1	Columnas K-Map
10	1010	1111	1	Columnas K-Map
11	1011	1110	1	Columnas K-Map
12	1100	1010	1	Columnas K-Map
13	1101	1011	1	Columnas K-Map
14	1110	1001	1	Columnas K-Map
15	1111	1000	1	Columnas K-Map

✓ Verde = Solo 1 bit cambia (propiedad fundamental del Código Gray)
✗ Rojo = Múltiples bits cambian (NO ocurre en secuencia Gray consecutiva)

Esta tabla muestra todos los valores de 0 a 15 en binario y Gray. Observa que:

- La columna "Bits que cambian" siempre marca **1** para transiciones consecutivas (comparándolo con el valor anterior)
- Los valores se organizan en secuencias para filas y columnas del mapa K

1.5. ¿Por qué Gray es esencial en Karnaugh?



El problema con el binario natural:

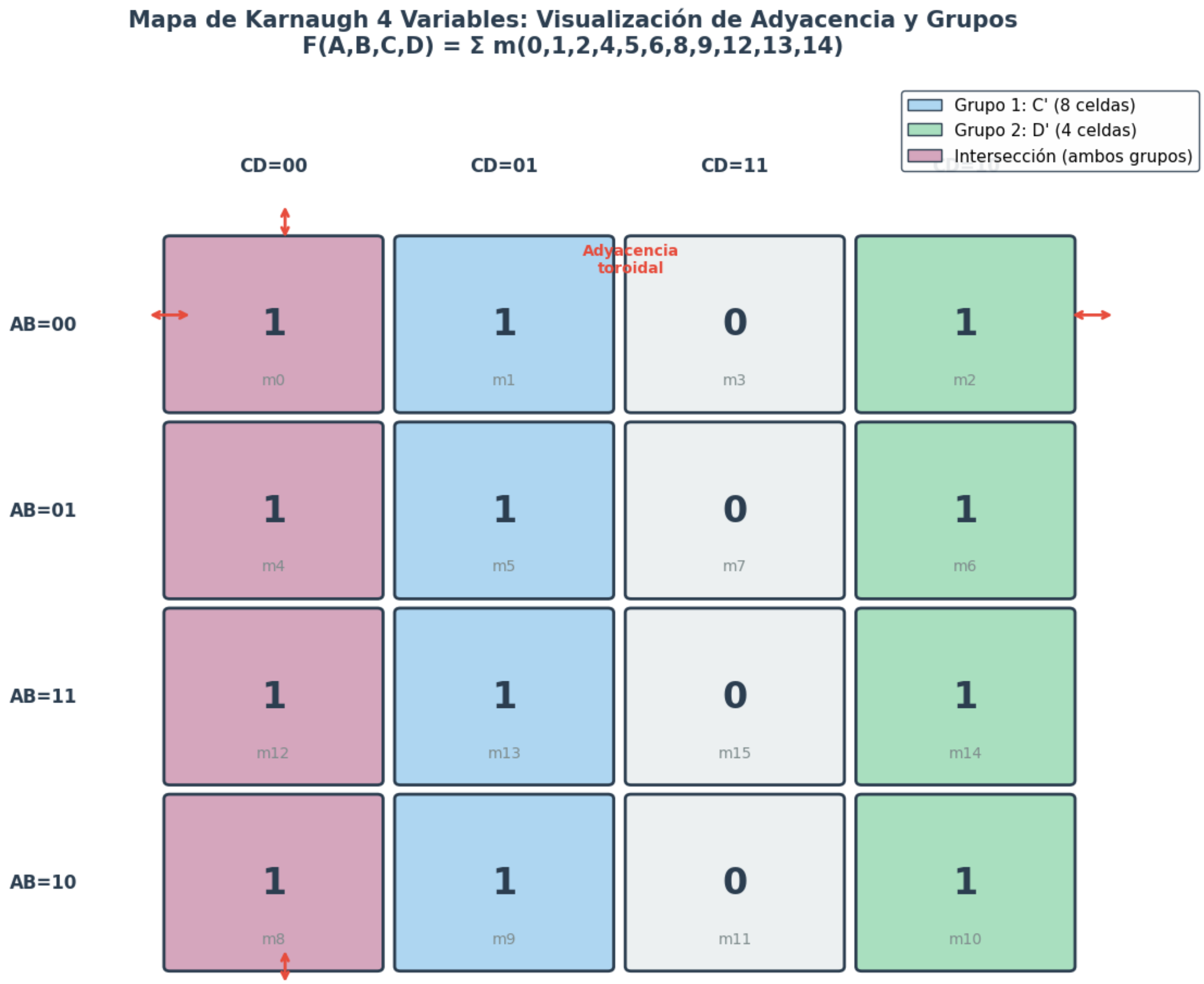
- m3 (0011) y m4 (0100) son consecutivos en decimal
- Pero difieren en **3 bits** (A, B, C cambian simultáneamente)
- **NO son adyacentes lógicamente** → No se pueden agrupar en Karnaugh

La solución Gray:

- m3 (0010 Gray) → m4 (0110 Gray): solo cambia el bit B
- **SON adyacentes lógicamente** → Se pueden agrupar perfectamente

Resultado: El Código Gray garantiza que celdas **físicamente adyacentes** en el mapa sean también **lógicamente adyacentes** (difieren en 1 bit), permitiendo la simplificación booleana correcta.

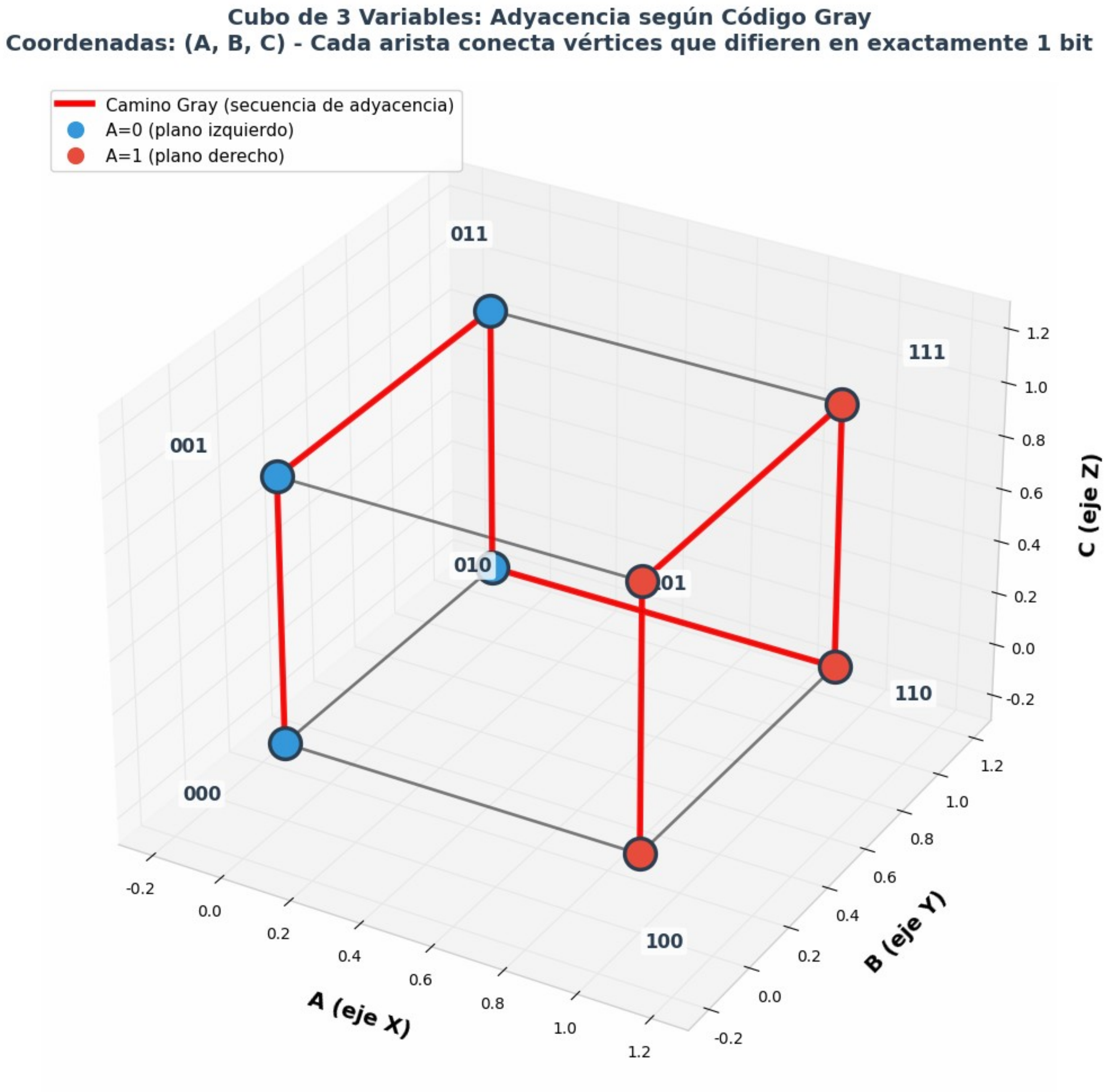
1.6. Visualización en el Mapa de Karnaugh



En el mapa de 4 variables:

- **Columnas:** Orden Gray 00, 01, 11, 10 (no 00, 01, 10, 11)
- **Filas:** Orden Gray 00, 01, 11, 10
- **Adyacencia toroidal:** Los bordes opuestos también son adyacentes (el mapa se "dobla" como un cilindro)

1.7. Interpretación Geométrica: El Cubo de 3 Variables



El mapa de Karnaugh de 3 variables es una proyección 2D de un **cubo 3D** (hipercubo para 4 variables):

- Cada **vértice** del cubo representa un minitérmino
- Cada **arista** conecta dos minitérminos que difieren en exactamente 1 bit
- El **camino Gray** (línea roja) recorre el cubo visitando cada vértice una sola vez, siempre moviéndose por aristas

Esta estructura geométrica explica por qué:

- Las agrupaciones de 2 celdas = una arista del cubo

- Las agrupaciones de 4 celdas = una cara del cubo
- Las agrupaciones de 8 celdas = todo el cubo

1.8. Resumen de Importancia para Ingeniería de Sistemas

Aspecto	**Sin Código Gray**	**Con Código Gray**
Adyacencia física	No garantiza adyacencia lógica	Garantiza adyacencia lógica
Agrupación de términos	Incorrecta o incompleta	Óptima y sistemática
Simplificación booleana	Puede fallar	Siempre encuentra el mínimo
Implementación en FPGA	Ineficiente	Mínimo número de LUTs
Diseño de circuitos	Más puertas lógicas	Menor costo y consumo

2. Estructura de los Mapas de Karnaugh

Un mapa de Karnaugh es una tabla donde cada celda representa un mintermino (combinación única de entradas). Las celdas están dispuestas de forma que entre dos celdas adyacentes solo cambia un bit (código Gray), lo que facilita la agrupación de términos.

Número de celdas
Depende del número de variables de entrada:

Número de variables	**Celdas en el mapa**
2	4 (2x2)
3	8 (4x2)
4	16 (4x4)

2.1 Reglas para la minimización con Mapas de Karnaugh

1. Solo agrupar celdas adyacentes (incluyendo bordes opuestos: el mapa es "toroidal").
2. Las agrupaciones deben tener tamaño potencia de 2: 1, 2, 4, 8, 16 celdas.
3. Cada grupo debe ser lo más grande posible.
4. Todas las celdas con "1" deben estar cubiertas.
5. Una misma celda puede pertenecer a varios grupos (para maximizar el tamaño).
6. No agrupar ceros.
7. Cada grupo genera un término producto (AND) simplificado.

2.1.1 Cómo obtener la función simplificada

Para cada grupo:
- Elimina las variables que cambian dentro del grupo.
- Conserva las variables que permanecen constantes.
- Si la variable es 1 → aparece directa; si es 0 → aparece negada.

Ejemplo (3 variables):
Grupo horizontal en A=1, C=1, B cambia → término: A . C

2.2 Mapas por número de variables

Mapa de Karnaugh para 2 variables (A, B)

	B=0	**B=1**
A=0	m0	m1
A=1	m2	m3

Ejemplo:

A	**B**	**F**
0	0	0
0	1	1
1	0	0
1	1	1

Función: $F(A,B) = \Sigma\ m(1,3)$ → minterminos 1 y 3 valen 1.

	B=0	**B=1**
A=0	0	1
A=1	0	1

Agrupación: Se agrupan las dos celdas con "1" en la columna B=1 → F = B

2.2.1 Mapa de Karnaugh para 3 variables (A, B, C)

	BC=00	**BC=01**	**BC=11**	**BC=10**
A=0	m0	m1	m3	m2
A=1	m4	m5	m7	m6

Nota: El orden de las columnas sigue el código Gray: 00, 01, 11, 10 (no binario natural).

Ejemplo:

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

F(A,B,C) = Σ m(1,3,5,7)

	BC=00	**BC=01**	**BC=11**	**BC=10**
A=0	0	1	1	0
A=1	0	1	1	0

Agrupación: Todos los "1" están en la columna BC=01 y BC=11 → corresponden a C = 1
→ F = C

2.2.2 Mapa de Karnaugh para 4 variables (A, B, C, D)

	CD=00	**CD=01**	**CD=11**	**CD=10**
AB=00	m0	m1	m3	m2
AB=01	m4	m5	m7	m6
AB=11	m12	m13	m15	m14
AB=10	m8	m9	m11	m10

Ejemplo:
F(A,B,C,D) = Σ m(0,1,2,4,5,6,8,9,12,13,14)

Se llenan las celdas correspondientes con "1" y se procede a agrupar.

Ejemplo completo: Simplificación de una función de 4 variables

Función:
F(A,B,C,D) = Σ m(0,1,2,4,5,6,8,9,12,13,14)

Paso 1:,Llenar el mapa

	CD=00	CD=01	CD=11	CD=10
AB=00	1	1	0	1
AB=01	1	1	0	1
AB=11	1	1	0	1
AB=10	1	1	0	1

Paso 2: Agrupar

	CD=00	CD=01	CD=11	CD=10
AB=00	1	1	0	1
AB=01	1	1	0	1
AB=11	1	1	0	1
AB=10	1	1	0	1

- Grupo 1: Los 8 unos de la columna izquierda (CD=00 y CD=01) → C' (C=0)
  - AB no los tomamos porque cambian en la selección.

	CD=00	CD=01	CD=11	CD=10
AB=00	1	1	0	1
AB=01	1	1	0	1
AB=11	1	1	0	1
AB=10	1	1	0	1

- Grupo 3: Los 4 unos de la columna (CD=10) y (CD=00)→ D’ (D=0)

Resultado simplificado:
Ya que hemos abarcado todos los 1, tendremos la siguiente función:

F = D' + C'

(Este es un ejemplo típico de función que se simplifica a dos términos.)

2.3 Casos especiales

Condiciones irrelevantes (Don't Care - X)
- En algunas aplicaciones, ciertas combinaciones de entrada nunca ocurren.
- Se marcan con X y pueden usarse como 0 o 1, según convenga para hacer grupos más grandes.

Ejemplo:
En un display de 7 segmentos, las combinaciones 10–15 no se usan en BCD → se marcan como X.

Aplicaciones prácticas

- Diseño de circuitos combinacionales: sumadores, decodificadores, multiplexores.
- Optimización de controladores lógicos (PLCs).
- Implementación eficiente de funciones en FPGA o microcontroladores.
- Reducción de hardware en sistemas embebidos.

Ejemplo Don’t Care X

El Escenario: Sensor de Tanque de Agua : Imagina un sistema que mide el nivel de agua de un tanque con dos sensores:

- A (Sensor de nivel bajo): Se activa (1) cuando hay al menos un poco de agua.
- B (Sensor de nivel alto): Se activa (1) solo cuando el tanque está lleno.

La Regla Lógica: Queremos que se encienda una bomba (Z) solo cuando el tanque esté completamente vacío para evitar que trabaje en seco innecesariamente.

Tabla de la verdad

A (Nivel bajo)	**B (Nivel Alto)**	**Bomba (Z)**	**Explicación**
0	0	1	Tanque vacío: Encender bomba.
0	1	X	**Imposible fisicamente:** No puede estar lleno sin estar bajo.
1	0	0	Tanque a la mitad: No encender.
1	1	0	Tanque lleno: No encender.

Físicamente es **imposible** que el sensor de nivel alto (B) esté activado si el sensor de nivel bajo (A) está apagado (no puedes tener el tanque lleno y vacío al mismo tiempo). Esa combinación es nuestra **X**.

Mapa de Karnaugh

	B = 0	**B = 1**
A = 0	1	X
A = 1	0	0

- **Si tratamos la X como 0**: El único grupo posible es la celda (0,0), lo que nos da la función: Z = A' B'. (Necesitas dos sensores y una compuerta NOR).
- **Si tratamos la X como 1**: Podemos agrupar la fila superior completa (A=0). Dentro de ese grupo, la variable B cambia (es 0 y luego 1), por lo que se elimina.

R//

Z = A'

Ejemplo:

Diseñe el circuito de control para una alarma contra incendios con las siguientes características:

- Tiene la opción de ser activada de forma manual desde un interruptor el cual siempre dispara la alarma, adicionalmente tiene un sensor de humo y uno de temperatura, los cuales disparan la alarma si los dos están accionados.

El circuito debe estar optimizado con compuertas lógicas.

A = Manual
B = Humo
C = Termperatura

A	**B**	**C**	**Z**
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

	AB 00	**AB 01**	**AB 11**	**AB 10**
C 0	0	0	1	1
C 1	0	1	1	1

Se selecciona en potencia de 2 : 1, 2, 4, 8, 16…. (primer opcion cuatro 1 y dos 1)

	AB 00	**AB 01**	**AB 11**	**AB 10**
C 0	0	0	1	1
C 1	0	1	1	1

	AB 00	AB 01	AB 11	AB 10
C 0	0	0	1	1
C 1	0	1	1	1

Z = BC + A

2.4 Ejercicios:

Ejercicio 1 :

	CD=00	CD=01	CD=11	CD=10
AB=00	1	1	0	1
AB=01	0	1	0	0
AB=11	0	1	0	0
AB=10	0	1	0	0

R//

Z = C′D + A′B′D′

Ejercicio 2 :

	CD=00	CD=01	CD=11	CD=10
AB=00	0	1	1	0
AB=01	1	0	0	1
AB=11	1	0	0	1
AB=10	0	1	1	0

R//
Z = BD’ + B’D

Ejercicio 3 :

	CD=00	CD=01	CD=11	CD=10
AB=00	1	1	0	0
AB=01	1	1	0	0
AB=11	1	1	1	1
AB=10	1	1	1	1

Ejercicio 4 :

	CD=00	CD=01	CD=11	CD=10
AB=00	1	0	0	1
AB=01	1	0	1	1
AB=11	1	0	1	1
AB=10	1	0	0	1

Ejercicio 5 :

	CD=00	CD=01	CD=11	CD=10
AB=00	1	1	1	1
AB=01	0	1	1	0
AB=11	0	1	1	0
AB=10	1	1	1	1

Ejercicio 6:

Un tribunal de oposiciones está formado por un presidente (P) y tres vocales (A, B y C). La decisión de que un alumno apruebe (1) o suspenda (0), se toma por mayoría y, en caso de empate, prevalece el voto de calidad del presidente.
Se pide:
a) La tabla de verdad del proceso de votación.
b) La función lógica simplificada por Karnaugh.
c) Implementar la función simplificada en arduino.

R// Arduino:

```
digitalWrite(ledPin, (P && (A || B || C)) || (A && B && C));
```

2.5 Ventajas y desventajas

Ventajas	Desventajas
Método gráfico y visual	Solo útil hasta 5 variables
Rápido y sistemático	Difícil de automatizar
Reduce errores algebraicos	Requiere práctica para dominarlo
Ideal para diseño manual	Para más de 5 variables se usan algoritmos (Quine-McCluskey)

2.6 Resumen de pasos para usar un Mapa de Karnaugh

1. Obtener la tabla de verdad o la suma de minterminos.
2. Dibujar el mapa según el número de variables.

3. Colocar un "1" en cada mintermino de la función.
4. Agrupar los "1" siguiendo las reglas (potencias de 2, adyacencia, máximo tamaño).
5. Escribir el término producto para cada grupo.
6. Sumar todos los términos (forma SOP: Suma de Productos).
7. Dibujar el circuito con puertas lógicas.

3. Glosario

Término	**Definición**
Mintermino	Combinación única de variables que produce un 1 en la salida
Mapa de Karnaugh	Herramienta gráfica para minimizar funciones booleanas
Agrupación	Conjunto de celdas adyacentes con valor 1
SOP	Suma de productos (forma canónica simplificada)
Don't Care (X)	Condición irrelevante que puede tomarse como 0 o 1

Recursos:

https://www.youtube.com/watch?v=nIgIREYHbx4
https://www.youtube.com/watch?v=vacBsx_ZljY
https://www.youtube.com/watch?v=9dd6eW6-p1M&t=2s
https://www.youtube.com/watch?v=AqUp-RZVzt8