# Álgebra de Boole

Prof. Rodrigo Araya E.
raraya@inf.utfsm.cl

Universidad Técnica Federico Santa María
Departamento de Informática

Valparaíso, 1er Semestre 2006

# Introducción

- En 1815 *George Boole* propuso una herramienta matemática llamada Álgebra de Boole.
- Luego en 1938 Claude Shannon propuso que con esta álgebra es posible modelar los llamados Sistemas Digitales.

# Álgebra de Boole

- El Álgebra de Boole es un sistema matemático que utiliza variables y operadores lógicos. Las variables pueden valer 0 ó 1. Y las operaciones básicas son OR**(+)** y AND**(·)**.
- Luego se definen las expresiones de conmutación como un número finito de variables y constantes, relacionadas mediante los operadores (AND y OR).
- En la ausencia de paréntesis, se utilizan las mismas reglas de precedencia, que tienen los operadores suma **(OR)** y multiplicación **(AND)** en el álgebra normal.

# Álgebra de Boole

## Leyes

**En el álgebra de Boole se cumplen las siguientes Leyes:**

- **1) Conmutatividad**:

$$X + Y = Y + X$$

$$X \cdot Y = Y \cdot X$$

# Álgebra de Boole

## Leyes

- **2) Asociatividad**:

$$X + (Y + Z) = (X + Y) + Z$$

$$X \cdot (Y \cdot Z) = (X \cdot Y) \cdot Z$$

- **3) Distributividad**:

$$X + (Y \cdot Z) = (X + Y) \cdot (X + Z)$$

$$X \cdot (Y + Z) = (X \cdot Y) + (X \cdot Z)$$

# Álgebra de Boole

## Identidades

- **4) Elementos Neutros (Identidad)**:

$$X + 0 = X$$

$$X \cdot 1 = X$$

- **5) Complemento**:

$$X + \overline{X} = 1$$

$$X \cdot \overline{X} = 0$$

# Álgebra de Boole

## Leyes

- **6) Dominación:**

$$X + 1 = 1 \quad X \cdot 0 = 0$$

Demostración:

$$X + 1 = (X + 1) \cdot 1 = (X + 1) \cdot (X + \overline{X})$$

$$(X + 1) \cdot (X + \overline{X}) = X + (1 \cdot \overline{X}) = 1$$

- **7) Idempotencia:**

$$X + X = X$$

$$X \cdot X = X$$

# Álgebra de Boole

## Leyes

- **8) Doble complemento:**

$$\overline{\overline{X}} = X$$

.

- **9) Absorción:**

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

$$X \cdot (Y + X) = X$$

Demostración:

$$X + X \cdot Y = (X \cdot 1) + (X \cdot Y) = X \cdot (1 + Y) = X$$

# Álgebra de Boole

## Leyes

- **10) DeMorgan:**

$$\overline{A \cdot B} = \overline{A} + \overline{B}$$

$$\overline{A + B} = \overline{A} \cdot \overline{B}$$

# Álgebra de Boole

## Teoremas

Luego se establecen los siguientes Teoremas:

- **Teorema de la Simplificación**

$$A + \overline{A} \cdot B = A + B$$

$$A \cdot (\overline{A} + B) = A \cdot B$$

Demostración: $\rightarrow$

$$A \cdot \overline{A} = 0$$
$$A \cdot \overline{A} + B = B$$
$$(A + B) \cdot (\overline{A} + B) = B$$
$$A \cdot (A + B) \cdot (\overline{A} + B) = A \cdot B$$
$$A \cdot (\overline{A} + B) = A \cdot B$$

# Álgebra de Boole

## Teoremas

- **Teorema del complemento único**

  Suponemos 2 complementos para $A$ ($A_1$ y $A_2$)

$$A + A_1 = 1 \quad A + A_2 = 1$$
$$A \cdot A_1 = 0 \quad A \cdot A_2 = 0$$

  Luego,

$$A_1 = A_1 \cdot 1 = A_1 \cdot (A + A_2) = A_1 \cdot A + A_1 \cdot A_2$$
$$A_1 = 0 + A_2 \cdot A_1$$
$$A_1 = A \cdot A_2 + A_1 \cdot A_2 = (A + A_1) \cdot A_2$$
$$A_1 = 1 \cdot A_2 = A_2$$

# Expresiones de Conmutación

## Algunas definiciones:

- **Literal:** Es toda ocurrencia de una variable, ya sea complementada o sin complementar, en una expresión de conmutación.
  Por ejemplo, en la expresión de conmutación:

$$\overline{A} \cdot B + C \cdot A + D + \overline{B} \cdot 1$$

$A$, $B$, $C$ y $D$ son Variables.
$\overline{A}$, $B$, $C$, $A$, $D$ y $\overline{B}$ son Literales.
1 es una Constante.

# Expresiones de Conmutación

## Algunas definiciones:

- **Expresión Dual:** Esta expresión se obtiene, intercambiando las operaciones AND por OR (y vice versa), e intercambiando las constantes 0 por 1 y 1 por 0 en la expresión de conmutación.
  Por ejemplo, para la expresión de conmutación:

$$(A \cdot B) + (C \cdot D) + 0$$

  La Expresión Dual es:

$$(A + B) \cdot (C + D) \cdot 1$$

# Funciones de conmutación

- Las funciones de conmutación se pueden expresar: de **Forma Algebraica**, mediante una **Tabla de Verdad** o en **Forma Canónica**.
- La manera más didáctica de representar una función de conmutación es mediante una Tabla de Verdad, ya que en ella se muestran los valores de salida para cada combinación de valor de entrada.
- Las Tablas de Verdad permiten modelar los Sistemas Combinacionales.

# Tablas de Verdad

## Ejemplo de una tabla de Verdad

Dada la función de conmutación: $f(X_1, X_2, X_3) = X_1 + (X_2 \cdot \overline{X_3})$
La Tabla de Verdad es:

| $X_1$ | $X_2$ | $X_3$ | $f(X_1, X_2, X_3)$ |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

# Formas Normales

- Dada una tabla de verdad también es posible obtener la forma algebraica.
- Existen 2 métodos para identificar la forma algebraica: la forma normal disyuntiva y la forma normal conjuntiva.
- En el caso de la *forma normal disyuntiva*, es necesario identificar los 1's que resultan de la tabla de verdad y formar los términos (*conjunciones fundamentales*) que los representan.
- Para formar las conjunciones fundamentales, se usa la variable complementada si para esa combinación tiene un cero, o se deja sin complementar, si en la combinación hay un 1.

# Formas Normales

## Forma normal disyuntiva

Dada la *Tabla de Verdad*:

| $X_1$ | $X_2$ | $X_3$ | $f(X_1, X_2, X_3)$ | | |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | | |
| 0 | 0 | 1 | 0 | | |
| 0 | 1 | 0 | 1 | $\rightarrow$ | $\overline{X_1} \cdot X_2 \cdot \overline{X_3}$ |
| 0 | 1 | 1 | 0 | | |
| 1 | 0 | 0 | 1 | $\rightarrow$ | $X_1 \cdot \overline{X_2} \cdot \overline{X_3}$ |
| 1 | 0 | 1 | 1 | $\rightarrow$ | $X_1 \cdot \overline{X_2} \cdot X_3$ |
| 1 | 1 | 0 | 1 | $\rightarrow$ | $X_1 \cdot X_2 \cdot \overline{X_3}$ |
| 1 | 1 | 1 | 1 | $\rightarrow$ | $X_1 \cdot X_2 \cdot X_3$ |

# Formas Normales

- Del ejemplo anterior, se suman las conjunciones fundamentales, resultando la *forma normal disyuntiva*:

$$f(X_1, X_2, X_3) = \overline{X_1} \cdot X_2 \cdot \overline{X_3} + X_1 \cdot \overline{X_2} \cdot \overline{X_3} + X_1 \cdot \overline{X_2} \cdot X_3 + X_1 \cdot X_2 \cdot \overline{X_3} + X_1 \cdot X_2 \cdot X_3$$

- Estos términos formados por todas las variables conectadas mediante operadores *AND* se denominan mintérminos (conjunciones fundamentales).
- Como la **función de conmutación** corresponde a un *OR* de todos los *mintérminos*, se puede expresar también de la forma canónica (OR canónico de AND).

$$F(X_1, X_2, X_3) = \sum\nolimits_m (m_0, m_1, \ldots, m_n)$$

# Formas Canónicas

- Para la representación de la forma canónica, se utilizan las posiciones de los mintérminos en la Tabla de Verdad.

Para el ejemplo anterior resulta:

$$f(X_1, X_2, X_3) = \sum_m(2, 4, 5, 6, 7)$$

# Formas Canónicas

## Mintérminos en una Tabla de Verdad

Dada una Tabla de Verdad:

| $X_1$ | $X_2$ | $X_3$ | **Mintérmino** | **Etiqueta** |
|---|---|---|---|---|
| 0 | 0 | 0 | $\overline{X_1} \cdot \overline{X_2} \cdot \overline{X_3}$ | 0 |
| 0 | 0 | 1 | $\overline{X_1} \cdot \overline{X_2} \cdot X_3$ | 1 |
| 0 | 1 | 0 | $\overline{X_1} \cdot X_2 \cdot \overline{X_3}$ | 2 |
| 0 | 1 | 1 | $\overline{X_1} \cdot X_2 \cdot X_3$ | 3 |
| 1 | 0 | 0 | $X_1 \cdot \overline{X_2} \cdot \overline{X_3}$ | 4 |
| 1 | 0 | 1 | $X_1 \cdot \overline{X_2} \cdot X_3$ | 5 |
| 1 | 1 | 0 | $X_1 \cdot X_2 \cdot \overline{X_3}$ | 6 |
| 1 | 1 | 1 | $X_1 \cdot X_2 \cdot X_3$ | 7 |

# Formas Normales

- En el caso de la forma normal conjuntiva, se opera de manera contraria a la vista anteriormente.
- En este caso es necesario identificar los 0's que resultan de la tabla de verdad y formar los términos (*disyunciones fundamentales* o maxtérminos) que los representan.
- Para ello se utiliza la variable complementada si para esa combinación tiene un 1, o se deja sin complementar si en la combinación hay un 0.

# Formas Normales

## Forma normal conjuntiva

Dada la *Tabla de Verdad*:

| $X_1$ | $X_2$ | $X_3$ | $f(X_1, X_2, X_3)$ | | |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | $\rightarrow$ | $X_1 + X_2 + X_3$ |
| 0 | 0 | 1 | 0 | $\rightarrow$ | $X_1 + X_2 + \overline{X_3}$ |
| 0 | 1 | 0 | 1 | | |
| 0 | 1 | 1 | 0 | $\rightarrow$ | $X_1 + \overline{X_2} + \overline{X_3}$ |
| 1 | 0 | 0 | 1 | | |
| 1 | 0 | 1 | 1 | | |
| 1 | 1 | 0 | 1 | | |
| 1 | 1 | 1 | 1 | | |

# Formas Normales

- Del ejemplo anterior, se opera con un AND sobre las disyunciones fundamentales, resultando la *forma normal conjuntiva*:

$$f(X_1, X_2, X_3) = (X_1 + X_2 + X_3) \cdot (X_1 + X_2 + \overline{X_3}) \cdot (X_1 + \overline{X_2} + \overline{X_3})$$

- De igual manera es posible expresar esta **función de conmutación**, compuesta por maxtérminos, de la forma canónica (AND canónico de OR).

$$F(X_1, X_2, X_3) = \prod_M (M_0, M_1, \ldots, M_n)$$

# Formas Canónicas

- Para la representación de la forma canónica, se utilizan las posiciones de los mintérminos en la Tabla de Verdad.

Para el ejemplo anterior resulta:

$$f(X_1, X_2, X_3) = \prod_M(0, 1, 3)$$

# Formas Canónicas

- ¿Como pasar de una forma algebraica, directamente a una forma canónica?

$$
\begin{aligned}
F(X_1, X_2, X_3) &= X_1 + (X_2 \cdot \overline{X_3}) \\
&= X_1 \cdot (X_2 + \overline{X_2}) \cdot (X_3 + \overline{X_3}) \\
&\quad + (X_1 + \overline{X_1})(X_2 \cdot \overline{X_3}) \\
&= X_1 \cdot X_2 \cdot (X_3 + \overline{X_3}) + X_1 \cdot \overline{X_2} \cdot (X_3 + \overline{X_3}) \\
&\quad + X_1 \cdot X_2 \cdot \overline{X_3} + \overline{X_1} \cdot X_2 \cdot \overline{X_3} \\
&= X_1 \cdot X_2 \cdot X_3 + X_1 \cdot X_2 \cdot \overline{X_3} + X_1 \cdot \overline{X_2} \cdot X_3 \\
&\quad + X_1 \cdot \overline{X_2} \cdot \overline{X_3} + X_1 \cdot X_2 \cdot \overline{X_3} + \overline{X_1} \cdot X_2 \cdot \overline{X_3}
\end{aligned}
$$

# Formas Canónicas

- ¿Como convertir de una forma OR canónico de AND a una forma AND canónico de OR?

$$
\begin{aligned}
F(X_1, X_2, X_3) &= \sum_m(2, 4, 5, 6, 7) \\
\overline{F(X_1, X_2, X_3)} &= \sum_m(0, 1, 3) \\
&= (\overline{X_1} \cdot \overline{X_2} \cdot \overline{X_3}) + (\overline{X_1} \cdot \overline{X_2} \cdot X_3) + (\overline{X_1} \cdot X_2 \cdot X_3) \\
F(X_1, X_2, X_3) &= (X_1 + X_2 + X_3) \cdot (X_1 + X_2 + \overline{X_3}) \cdot (X_1 + \overline{X_2} \\
&\quad + \overline{X_3}) \\
F(X_1, X_2, X_3) &= \prod_M(0, 1, 3)
\end{aligned}
$$

# Funciones equivalentes

- Se dice que dos funciones de conmutación son equivalentes si tienen expansiones en forma canónica idénticas. Es decir, que tienen valores de salida idénticos para las mismas combinaciones de entrada.
- Dicho de otra manera, dos funciones de conmutación son equivalentes si tienen la misma tabla de verdad.

# Funciones equivalentes

- ¿Cuántas funciones distintas (No equivalentes) existen para un número $n$ de variables?

$$2^{2^n}$$

- Esto se puede demostrar fácilmente, construyendo tablas de verdad y basándose en que las funciones *no equivalentes* tienen tablas de verdad distintas.

# Algunos Operadores

## Algunos operadores...

| | |
|---|---|
| NOT | $F(X_1) = \overline{X_1}$ |
| AND | $F(X_1, X_2) = X_1 \cdot X_2$ |
| OR | $F(X_1, X_2) = X_1 + X_2$ |
| NAND | $F(X_1, X_2) = \overline{X_1 \cdot X_2} = \overline{X_1} + \overline{X_2}$ |
| NOR | $F(X_1, X_2) = \overline{X_1 + X_2} = \overline{X_1} \cdot \overline{X_2}$ |
| XAND | $F(X_1, X_2) = X_1 \cdot X_2 + \overline{X_1} \cdot \overline{X_2}$ |
| XOR | $F(X_1, X_2) = X_1 \cdot \overline{X_2} + \overline{X_1} \cdot X_2$ |

Tarea: **Analizar las tablas de verdad de cada uno de estos operadores.**

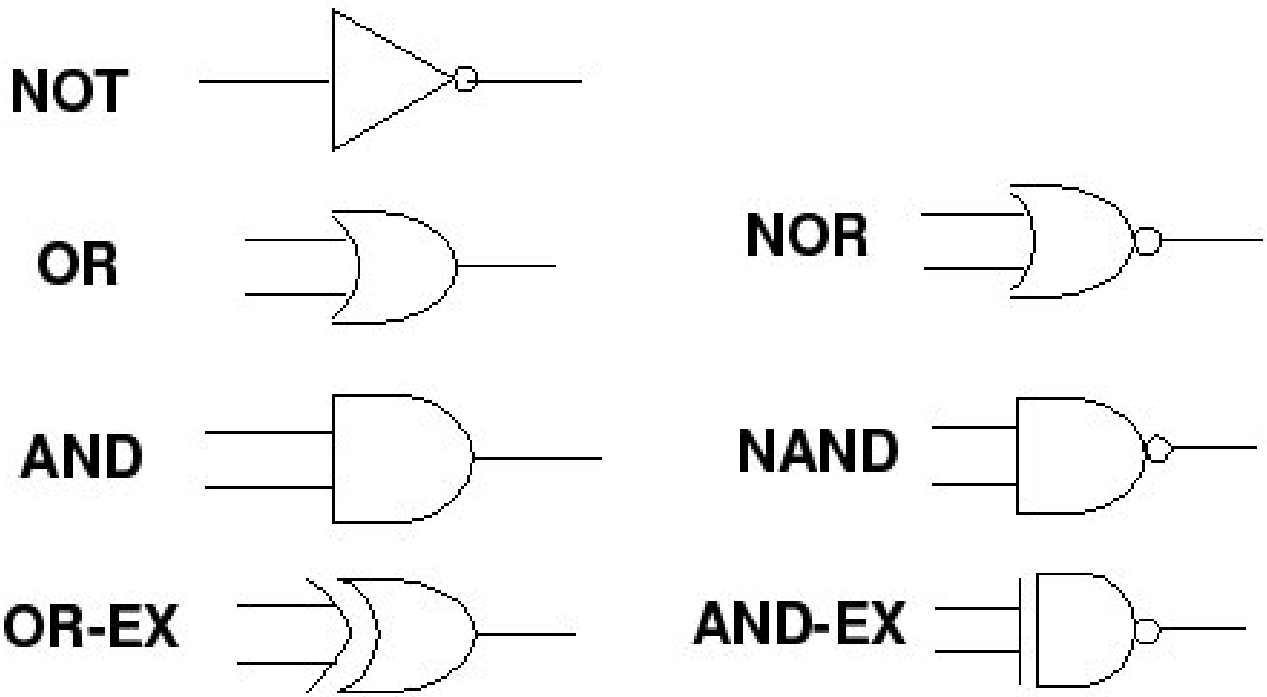
# Operadores funcionalmente completos

- Se dice que un conjunto de operadores es funcionalmente completo si se puede expresar cualquier función de conmutación, utilizando sólo los operadores del conjunto.
- Por ejemplo el conjunto {**AND, OR, NOT**} es funcionalmente completo por definición del álgebra. Sin embargo el conjunto {**AND, NOT**} también lo es.
- Otros conjuntos funcionalmente completos son: {**NOR**} y {**NAND**}.

# Compuertas Lógicas

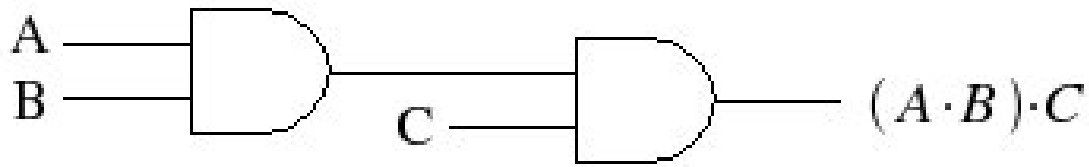
- Existen dispositivos electrónicos que son capaces de representar funciones de conmutación. Estos dispositivos denominan **Compuertas Lógicas** y están construidos a base de silicio.
- Las compuertas lógicas son altamente usadas en el campo de la electrónica digital, debido al bajo costo que se logra con la alta densidad de integración.
- Las compuertas corresponden a bloques fundamentales para la construcción de circuitos lógicos y sistemas digitales.
- Una red de compuertas lógicas constituye un circuito combinacional.

# Compuertas Lógicas

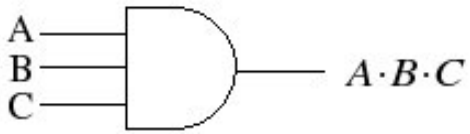


# Compuertas Lógicas

- Las compuertas pueden tener más de una o dos entradas. Por ejemplo la ecuación de conmutación $F(A, B, C) = A \cdot B \cdot C$ puede ser representada por:




O bien por:

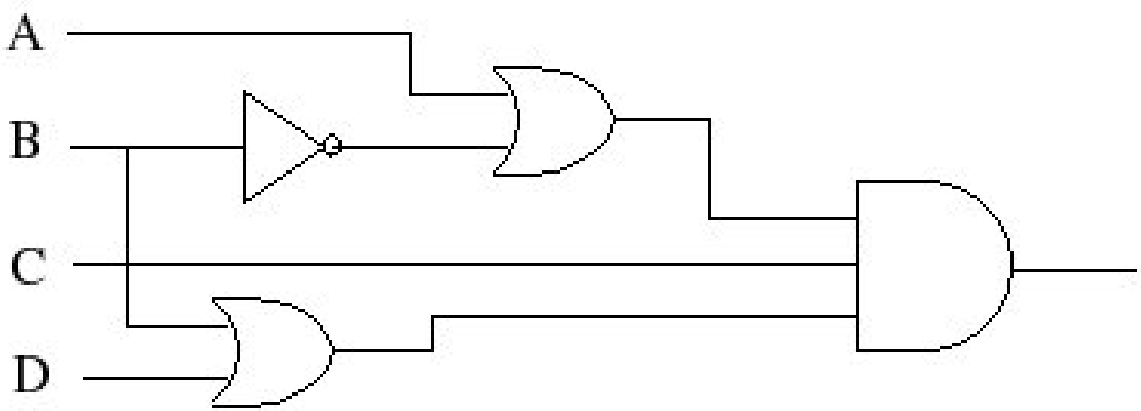


# Compuertas Lógicas

## Ejemplo de compuertas

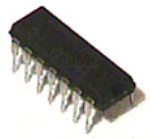
Representar la siguiente ecuación mediante compuertas lógicas.

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




# Compuertas Lógicas

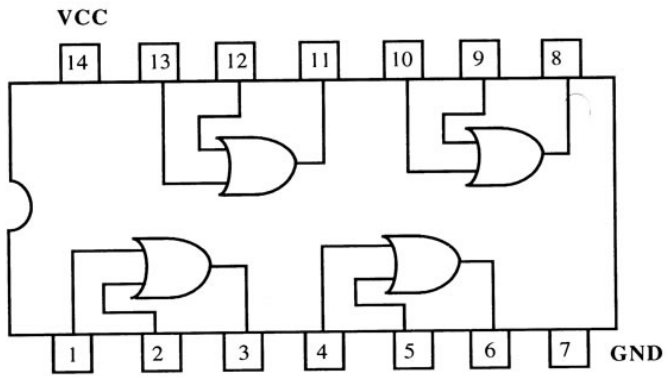
- Las compuertas lógicas se pueden encontrar en dispositivos pequeños de uso general, llamadas pastillas lógicas TTL. Su numeración corresponde a 74LSXXX.



- También existen dispositivos con alta densidad de integración como PLA, CPLD y FPGA.


# Compuertas Lógicas

- Las pastillas lógicas internamente están diseñadas con varias compuertas, dependiendo de la pastilla. Por ejemplo un 74LS32 internamente es de la siguiente forma:




*74LS32*

# Minimización de Funciones

- Minimizar una función $F(X_1, X_2, X_3, \ldots X_n)$ es encontrar una función equivalente $G(X_1, X_2, X_3, \ldots X_n)$ que tenga el mínimo número de términos y literales.

# Minimización de Funciones

- Por ejemplo, si tenemos la siguiente tabla de verdad:

| A | B | C | D | Z |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 |

| A | B | C | D | Z |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

# Minimización de Funciones

- Luego extraemos los mintérminos

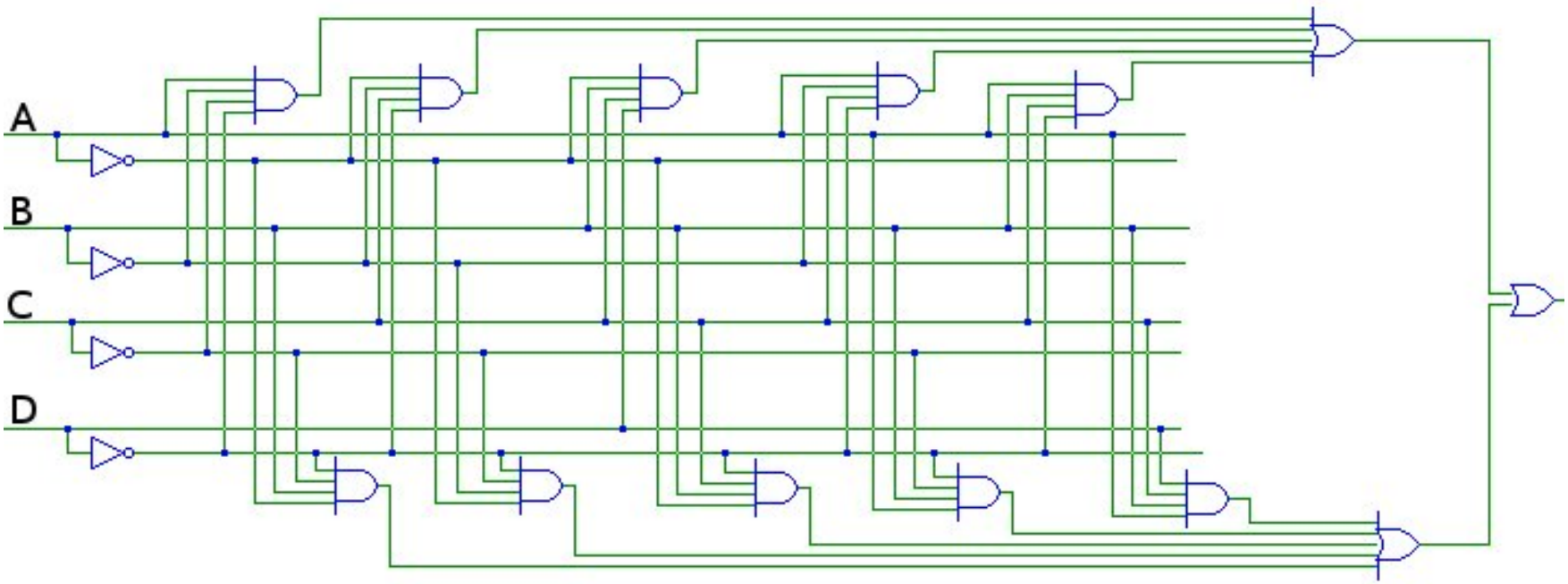
| A B C D | Z | Mintérmino | A B C D | Z | Mintérmino |
|---|---|---|---|---|---|
| 0 0 0 0 | 1 | $\rightarrow \overline{A} \cdot \overline{B} \cdot \overline{C} \cdot \overline{D}$ | 1 0 0 0 | 1 | $\rightarrow A \cdot \overline{B} \cdot \overline{C} \cdot \overline{D}$ |
| 0 0 0 1 | 0 | | 1 0 0 1 | 0 | |
| 0 0 1 0 | 1 | $\rightarrow \overline{A} \cdot \overline{B} \cdot C \cdot \overline{D}$ | 1 0 1 0 | 1 | $\rightarrow A \cdot \overline{B} \cdot C \cdot \overline{D}$ |
| 0 0 1 1 | 0 | | 1 0 1 1 | 0 | |
| 0 1 0 0 | 1 | $\rightarrow \overline{A} \cdot B \cdot \overline{C} \cdot \overline{D}$ | 1 1 0 0 | 1 | $\rightarrow A \cdot B \cdot \overline{C} \cdot \overline{D}$ |
| 0 1 0 1 | 0 | | 1 1 0 1 | 0 | |
| 0 1 1 0 | 1 | $\rightarrow \overline{A} \cdot B \cdot C \cdot \overline{D}$ | 1 1 1 0 | 1 | $\rightarrow A \cdot B \cdot C \cdot \overline{D}$ |
| 0 1 1 1 | 1 | $\rightarrow \overline{A} \cdot B \cdot C \cdot D$ | 1 1 1 1 | 1 | $\rightarrow A \cdot B \cdot C \cdot D$ |

# Minimización de Funciones

- La forma normal disyuntiva de la ecuación queda de la siguiente manera:

$$\begin{aligned} F(A,B,C,D) = & (\overline{A}\cdot\overline{B}\cdot\overline{C}\cdot\overline{D}) + (\overline{A}\cdot\overline{B}\cdot C\cdot\overline{D}) + (\overline{A}\cdot B\cdot\overline{C}\cdot\overline{D}) \\ & +(\overline{A}\cdot B\cdot C\cdot\overline{D}) + (\overline{A}\cdot B\cdot C\cdot D) + (A\cdot\overline{B}\cdot\overline{C}\cdot\overline{D}) \\ & +(A\cdot\overline{B}\cdot C\cdot\overline{D}) + (A\cdot B\cdot\overline{C}\cdot\overline{D}) + (A\cdot B\cdot C\cdot\overline{D}) \\ & +(A\cdot B\cdot C\cdot D) \end{aligned}$$

# Minimización de Funciones

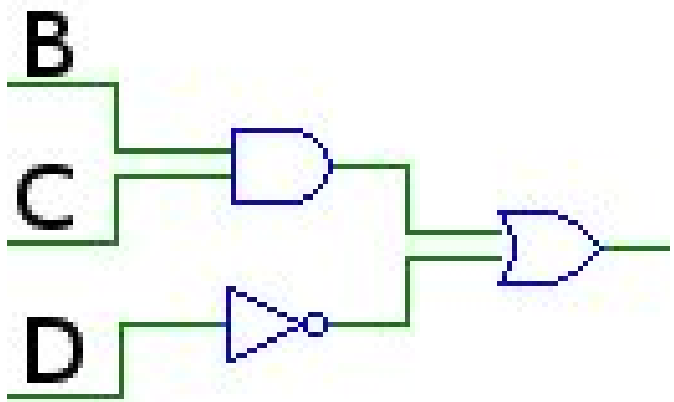


# Minimización de Funciones

- Si intentamos minimizar la ecuación, resulta la siguiente expresión:

$$
\begin{aligned}
F(A,B,C,D) &= (\overline{A}\cdot\overline{B}\cdot\overline{C}\cdot\overline{D})+(\overline{A}\cdot\overline{B}\cdot C\cdot\overline{D})+(\overline{A}\cdot B\cdot\overline{C}\cdot\overline{D}) \\
&\quad +(\overline{A}\cdot B\cdot C\cdot\overline{D})+(\overline{A}\cdot B\cdot C\cdot D)+(A\cdot\overline{B}\cdot\overline{C}\cdot\overline{D}) \\
&\quad +(A\cdot\overline{B}\cdot C\cdot\overline{D})+(A\cdot B\cdot\overline{C}\cdot\overline{D})+(A\cdot B\cdot C\cdot\overline{D}) \\
&\quad +(A\cdot B\cdot C\cdot D) \\
&= (\overline{A}\cdot\overline{B}+\overline{A}\cdot B+A\cdot B+A\cdot\overline{B})\cdot(\overline{C}\cdot\overline{D}) \\
&\quad +(\overline{A}\cdot\overline{B}+\overline{A}\cdot B+A\cdot B+A\cdot\overline{B})\cdot(C\cdot\overline{D}) \\
&\quad +(A+\overline{A})\cdot(B\cdot C\cdot D) \\
&= (\overline{A}+A)\cdot(B+\overline{B})\cdot(\overline{C}\cdot\overline{D}+C\cdot\overline{D})+(B\cdot C\cdot D) \\
&= \overline{D}+(B\cdot C\cdot D) \\
&= \overline{D}+(B\cdot C)
\end{aligned}
$$

# Minimización de Funciones

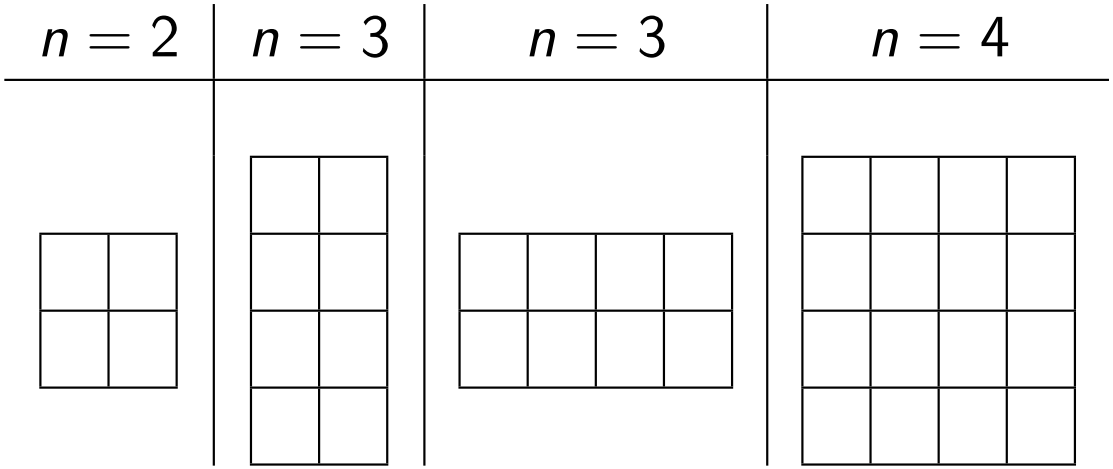



# Mapas de Karnaugh

- Los mapas de Karnaugh son una herramienta gráfica utilizada para simplificar las ecuaciones lógicas o bién, minimizar funciones de conmutación.
- Estos mapas son una versión modificada de la tablas de verdad, permitiendo mostrar la relación entre las entradas lógicas y la salida deseada.
- Los mapas de Karnaugh permiten el diseño de circuitos con el mínimo compuertas, por lo que tiene un alto impacto en la reducción de costos.

# Pasos para la construcción de un Mapa de Karnaugh

- **1)** Al igual que en las *tablas de verdad*, una función de $n$ variables tiene $2^n$ combinaciones de posibles valores de entrada. En el caso de los mapas de Karnaugh, estas combinaciones se representan mediante celdas.




# Pasos para la construcción de un Mapa de Karnaugh

- **2)** Luego, las coordenadas de las celdas se enumeran, según el código Grey, quedando de la siguiente manera:

| | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | | | | |
| 01 | | | | |
| 11 | | | | |
| 10 | | | | |

# Pasos para la construcción de un Mapa de Karnaugh

- **3)** Si se tiene una tabla de verdad, basta con escribir en cada celda la salida correspondiente de la tabla de verdad para cada combinación. Por ejemplo:

| A | B | C | Z |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

$\longrightarrow$

| C \ AB | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 |

# Pasos para la construcción de un Mapa de Karnaugh

- Equivalentemente se puede representar una función de la forma canónica, como mapa de Karnaugh. Para ello se debe asignar un 0 a una variable complementada y un 1 a una variable sin complementar.

| A B / C | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | $\overline{A}\,\overline{B}\,\overline{C}$ | $\overline{A}\,B\,\overline{C}$ | $A\,B\,\overline{C}$ | $A\,\overline{B}\,\overline{C}$ |
| 1 | $\overline{A}\,\overline{B}\,C$ | $\overline{A}\,B\,C$ | $A\,B\,C$ | $A\,\overline{B}\,C$ |

→

| A B / C | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 000 | 010 | 110 | 100 |
| 1 | 001 | 011 | 111 | 101 |

# Pasos para la construcción de un Mapa de Karnaugh

- Con esto se forma la siguiente numeración para las celdas.

| A B / C | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 0 | 2 | 6 | 4 |
| 1 | 1 | 3 | 7 | 5 |

- Luego si se quiere representar la función $F(A, B, C) = \sum_m(0, 2, 3, 7)$, resulta:

| A B / C | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 |

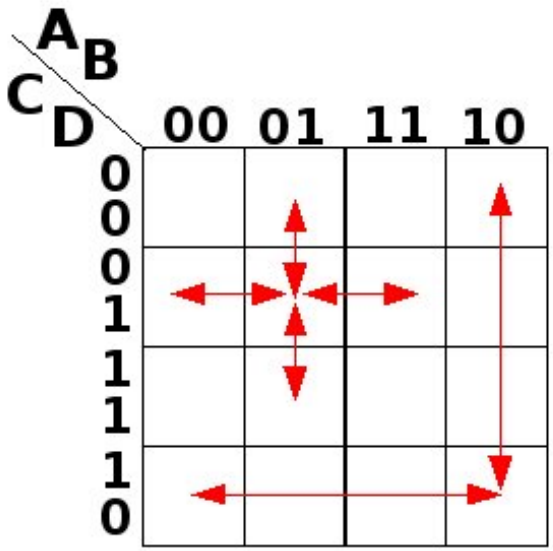
# Pasos para la construcción de un Mapa de Karnaugh

- Para 4 variables, la numeración de las celdas corresponde a:

| CD \ AB | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | 4 | 12 | 8 |
| 01 | 1 | 5 | 13 | 9 |
| 11 | 3 | 7 | 15 | 11 |
| 10 | 2 | 6 | 14 | 10 |

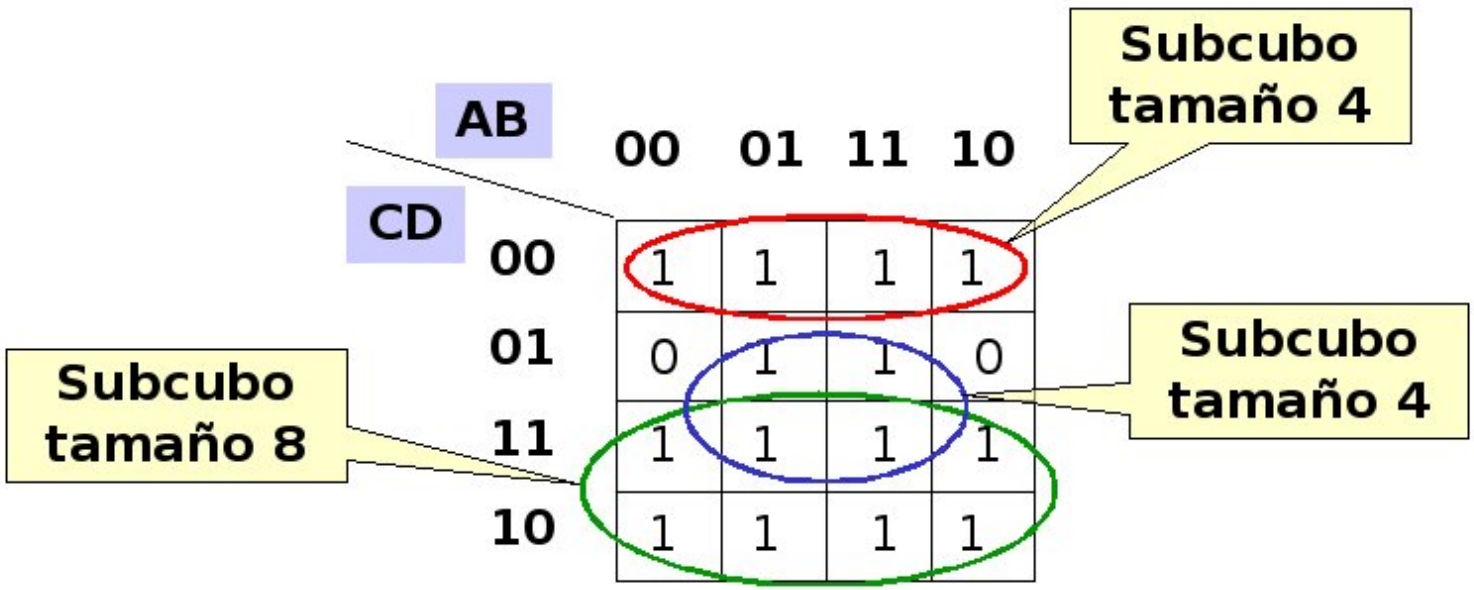
# Pasos para la construcción de un Mapa de Karnaugh

- **4)** Dos celdas son adyacentes sólo si difieren en una de las variables.



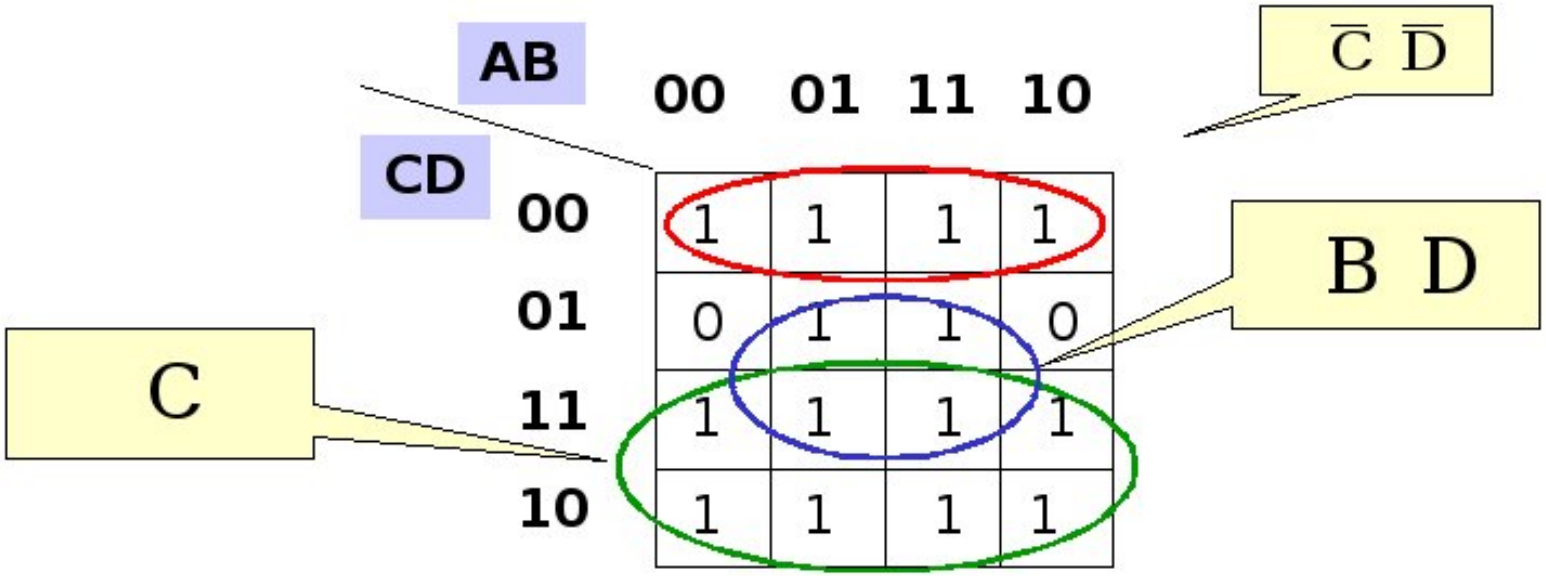

# Pasos para la construcción de un Mapa de Karnaugh

- **5)** Un subcubo es un conjunto de $2^m$ celdas con valor 1, las cuales tienen la propiedad que cada celda del subcubo es adyacente a exactamente m celdas del conjunto.



# Pasos para la construcción de un Mapa de Karnaugh

- **6)** Los **subcubos** se pueden representar mediante términos algebraicos. Estos términos están compuestos por $n - m$ literales, donde $n$ es el número de variables y $2^m$ es el tamaño del subcubo.

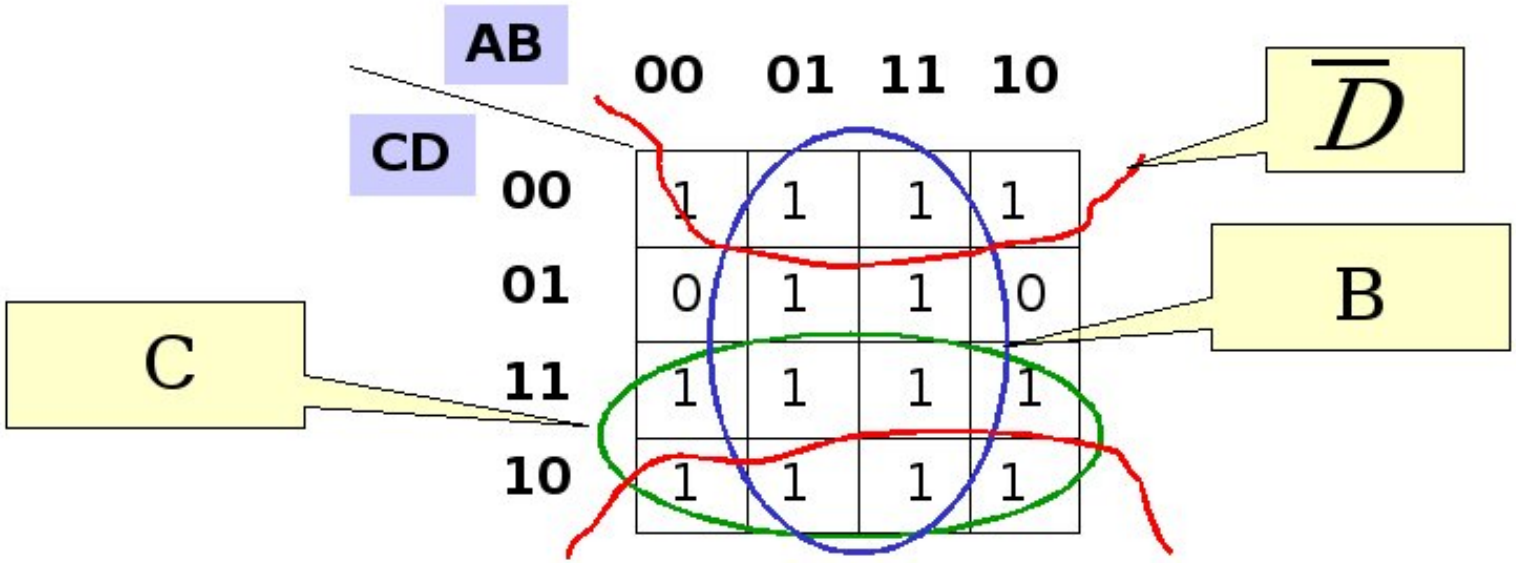


# Pasos para la construcción de un Mapa de Karnaugh

- **7)** Si se suman los términos dados por los subcubos que abarcan todos los unos del mapa, se obtiene la función algebraica.
- Para que la función sea mínima, se debe buscar el mínimo número de subcubos que cubren todos los unos. Esto se logra, buscando los subcubos de mayor tamaño posible, sin importar que se traslapen.

# Pasos para la construcción de un Mapa de Karnaugh

- El siguiente mapa de Karnaugh:




Representa la función

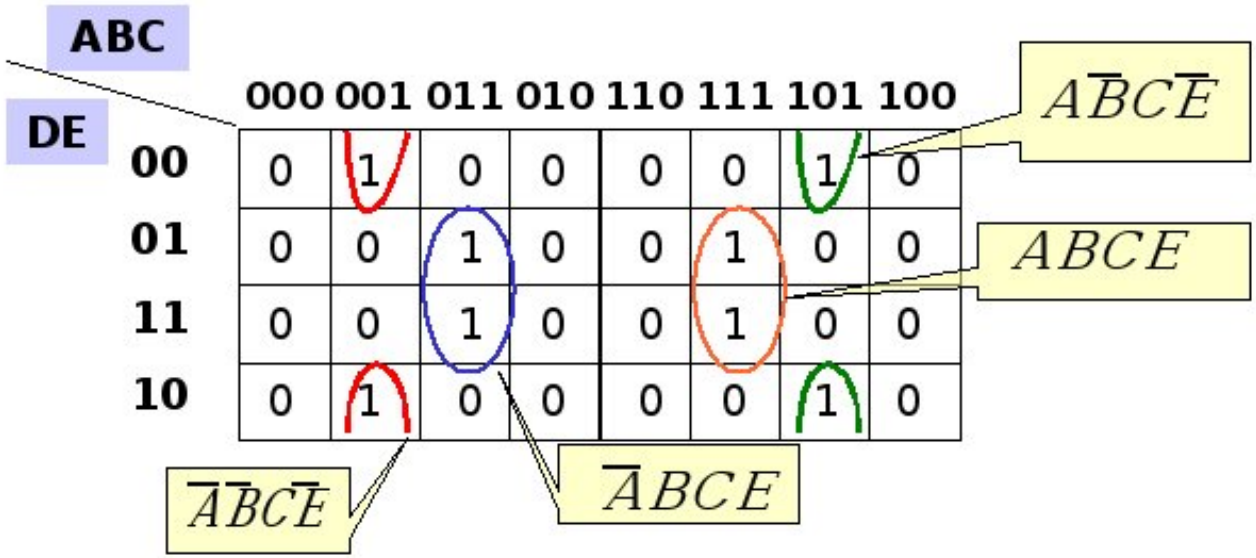
$$F(A, B, C, D) = \overline{D} + B + C$$

# Minimización mediante Mapas de Karnaugh

- En la práctica, al utilizar el método de los mapas de Karnaugh manualmente, resulta útil para un máximo de 5 o 6 variables.

# Minimización mediante Mapas de Karnaugh

- En el siguiente mapa de Karnaugh de 5 variables se identifican 4 subcubos:

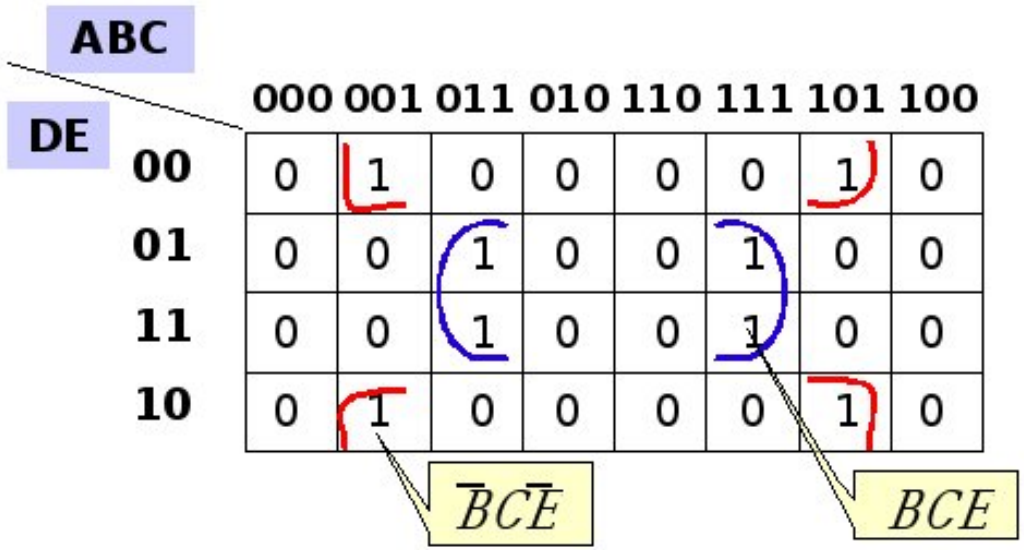



Resultando la ecuación

$$F(A,B,C) = \overline{A}\cdot\overline{B}\cdot C\cdot\overline{E} + A\cdot\overline{B}\cdot C\cdot\overline{E} + \overline{A}\cdot B\cdot C\cdot E + A\cdot B\cdot C\cdot E$$

# Minimización mediante Mapas de Karnaugh

- Sin embargo en el mapa anterior no están marcados los subcubos más grandes. Por lo que la función no es mínima.
- En el siguiente MK están marcados los subcubos más grandes.




Resultando la ecuación

$$F(A,B,C) = \overline{B} \cdot C \cdot \overline{E} + B \cdot C \cdot E$$

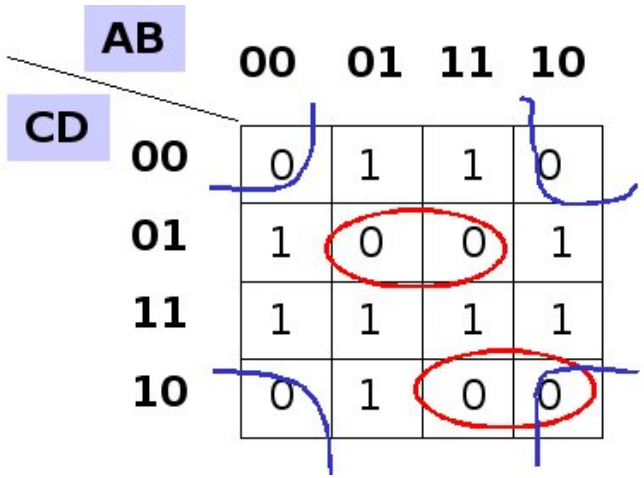
# Mapas de Karnaugh (AND de OR)

- También es posible expresar funciones de la *forma canónica AND de OR* en los mapas de Karnaugh.
- Para ello es necesario identificar los subcubos que cubren todos los ceros del MK.
- Por ejemplo minimizar

$$F(A, B, C, D) = \prod_{M}(0, 2, 5, 8, 10, 13, 14)$$

# Mapas de Karnaugh (AND de OR)

- El siguiente MK representa a la función $F(A, B, C, D) = \prod_M(0, 2, 5, 8, 10, 13, 14)$. En el se deben cubrir los ceros de mapa.




Resultando la ecuación

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

# Minimización de Funciones

- La minimización de funciones es fundamental tanto para el diseño de procesadores, como de otros componentes digitales que utilizan tecnología de alta densidad de integración (como VLSI).
- La minimización no solo tiene un alto impacto en el costo de los dispositivos, sino que también en el rendimiento.
- Sin embargo el método de MK no es viable en diseños complejos, como por ejemplo el diseño de un procesador, debido a la cantidad de variables que involucra.

# Método Quine - McKluskey

- El método de Quine y McKluskey es una técnica tabular.
- Esta técnica resulta fácil de programar, con lo que se logra una herramienta automática para la obtención de expresiones de conmutación mínimas.

# Método Quine - McKluskey

- Una expresión de conmutación se puede escribir como una suma de términos donde cada término esta compuesto de factores.

**Por ejemplo:**

$$F(A, B, C) = A \cdot B \cdot C + \overline{B} \cdot C + \ldots$$

- Se define como implicante primo a un término que está contenido en la función y que la eliminación de cualquiera de sus literales genera un nuevo término que no esta contenido en a función.

# Método Quine - McKluskey

## Implicantes Primos

- Por ejemplo la función $F(A, B, C) = AB + C$ tiene 2 términos ($AB$ y $C$), y ambos son implicantes primos.
- En cambio la función $F(A, B, C) = ABC + A + BC$ tiene 3 términos, pero sólo 2 de ellos son implicantes primos. El término $ABC$ no es implicante primo, ya que si se elimina la literal $A$, queda el término $BC$ que ya existe en la función.

# Método Quine - McKluskey

- Se puede observar que los implicantes primos corresponden a los subcubos en un mapa de Karnaugh. Por lo tanto, la ecuación minimizada tendrá tantos términos, como **implicantes primos** tenga la función.
- Los algoritmos computacionales para la minimización de funciones, se basan en la búsqueda automatizada de implicantes primos.

# Método Quine - McKluskey

- El método **Quine-McKluskey** genera el conjunto de implicantes primos de una función dada.

## Pasos para el desarrollo del método Quine - McKluskey

1) Para desarrollar el método, primero se debe contar con la función de la forma canónica OR de AND.
2) Luego se representa cada término, de la forma binaria.
3) Se agrupan los términos en función de la cantidad de 1's que tengan.

# Método Quine - McKluskey

## Pasos para el desarrollo del método Quine - McKluskey

4) Cada grupo (que representa la cantidad de 1's del término), se vuelve a agrupar con algún grupo adyacente buscando diferencias en un solo bit. El bit en que difieren es reemplazado por "-".
5) Se vuelve a aplicar el paso anterior. Para la adyacencia se debe considerar que el símbolo "-" se encuentra en la misma posición.
6) Finalmente Se deben cubrir todos los términos de la función original, utilizando el mínimo número de implicantes primos.

# Método Quine - McKluskey

## Ejemplo del método Quine - McKluskey

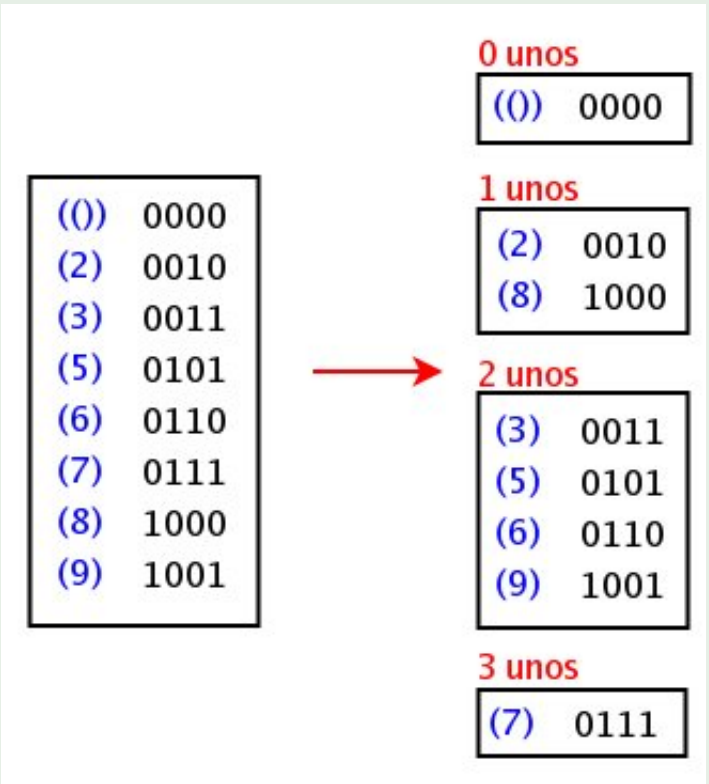
1) Como ejemplo se considera la siguiente función de la forma canónica OR de AND. $\sum_m(0,2,3,5,6,7,8,9)$
2) Se escribe cada término, de la forma binaria:

| | |
|---|---|
| (0) | 0000 |
| (2) | 0010 |
| (3) | 0011 |
| (5) | 0101 |
| (6) | 0110 |
| (7) | 0111 |
| (8) | 1000 |
| (9) | 1001 |

# Método Quine - McKluskey

## Ejemplo del método Quine - McKluskey

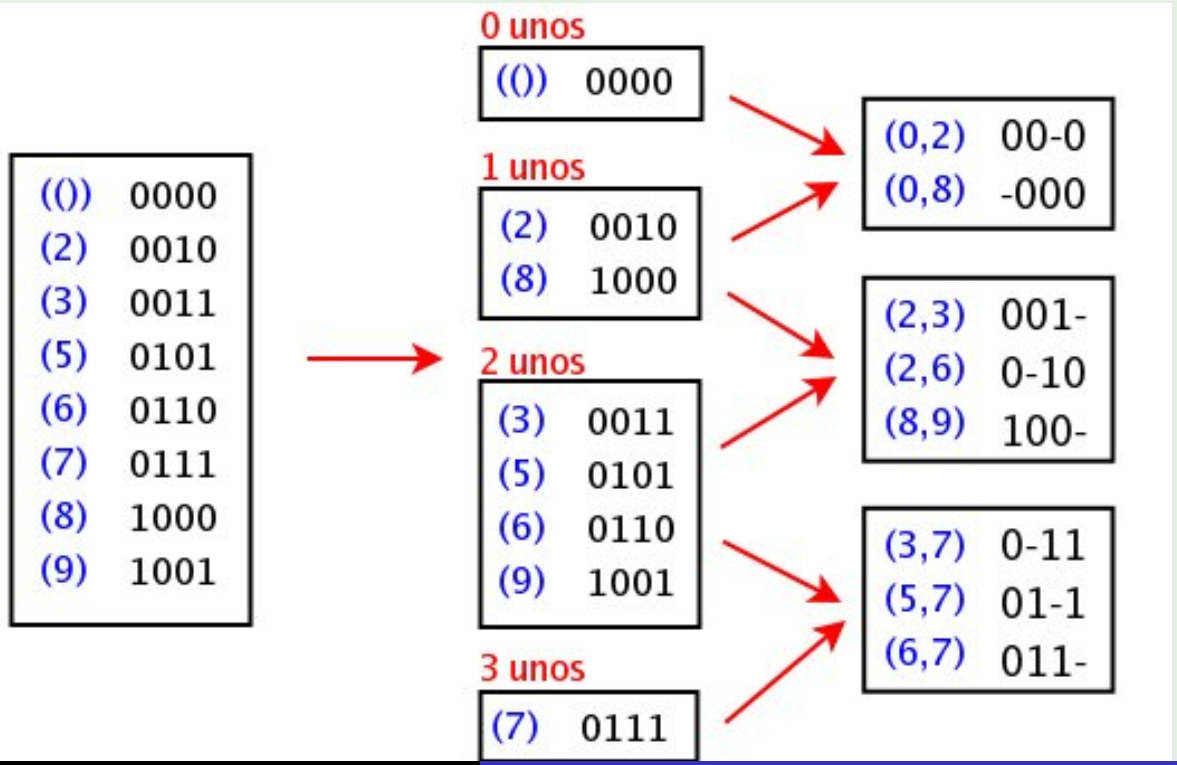
3) Luego se agrupan los términos, en función a la cantidad de 1's que tienen.



# Método Quine - McKluskey

## Ejemplo del método Quine - McKluskey

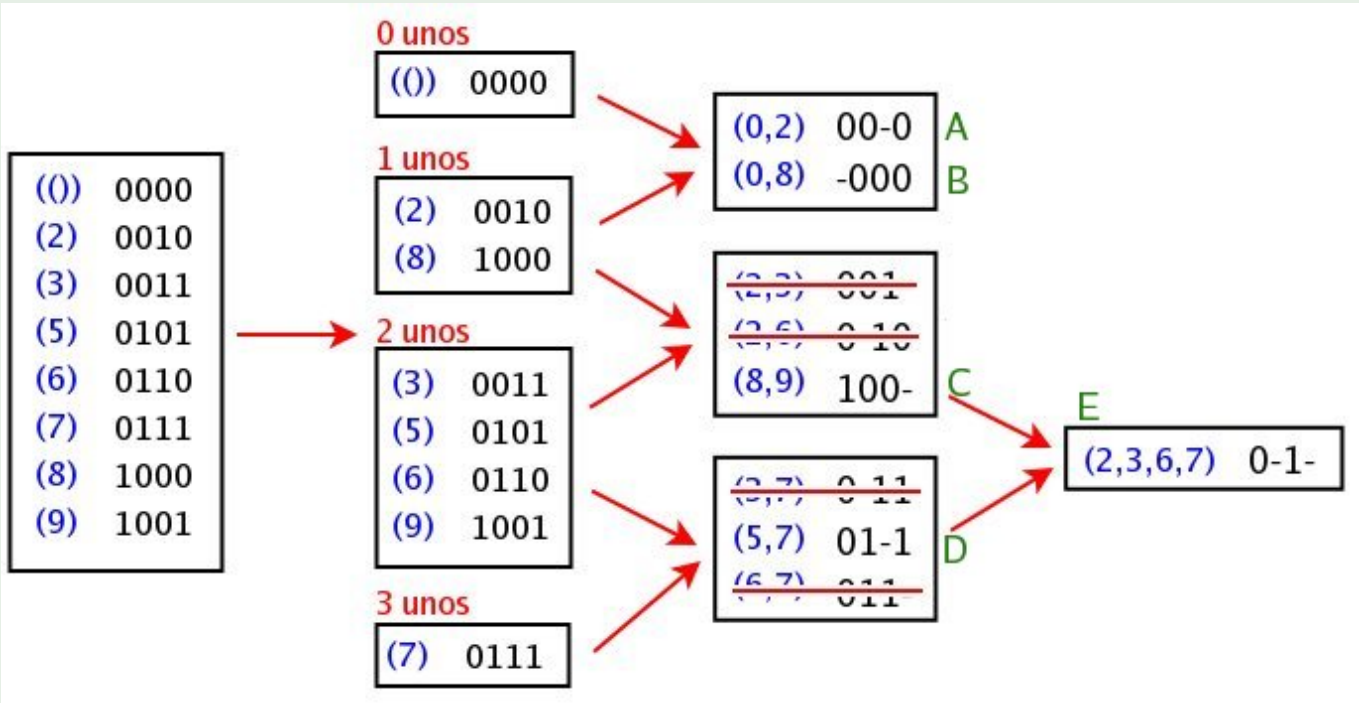
4) Se reagrupan los grupos adyacentes, buscando diferencias en un solo bit y reemplazando el bit en que difieren con un "-".



# Método Quine - McKluskey

## Ejemplo del método Quine - McKluskey

5) Se vuelve a reagrupar considerando que el símbolo "-" se encuentre en la misma posición.



# Método Quine - McKluskey

## Ejemplo del método Quine - McKluskey

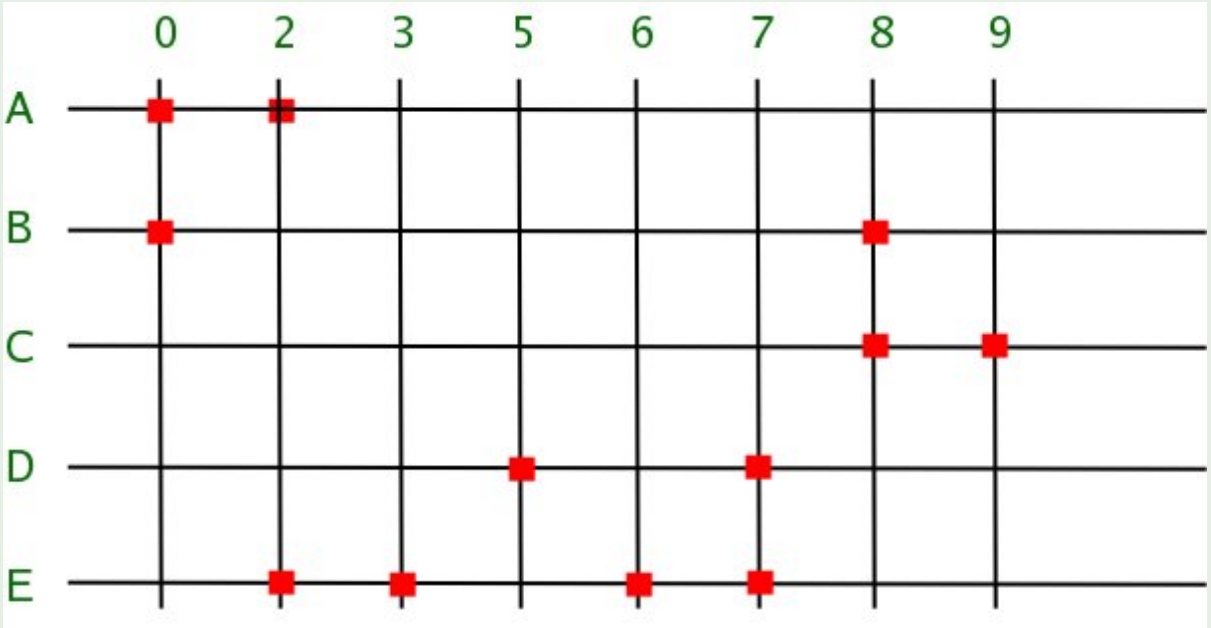
Los siguientes términos corresponden a los implicantes primos:

| | | |
|---|---|---|
| (0,2) | 00-0 | $\longrightarrow$ A |
| (0,8) | -000 | $\longrightarrow$ B |
| (8,9) | 100- | $\longrightarrow$ C |
| (5,7) | 01-1 | $\longrightarrow$ D |
| (2,3,6,7) | 0-1- | $\longrightarrow$ E |

# Método Quine - McKluskey

## Ejemplo del método Quine - McKluskey

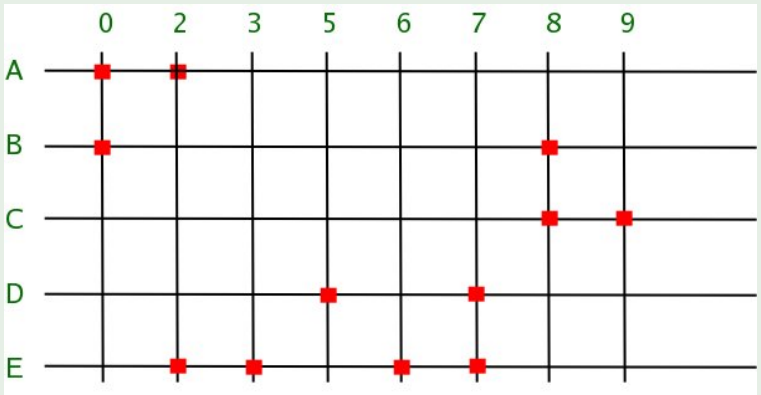
6) Finalmente se deben cubrir todos los términos de la función original, utilizando el mínimo número de implicantes primos.



# Método Quine - McKluskey

## Ejemplo del método Quine - McKluskey

Los implicantes primos que representan a todos los términos pueden ser: $C + D + E + B$ o $C + D + E + A$




Si se consideran términos con las variables $W, X, Y, Z$ se traduce a: $W \cdot \overline{X} \cdot \overline{Y} + \overline{W} \cdot X \cdot Z + \overline{W} \cdot Y + \overline{X} \cdot \overline{Y} \cdot \overline{Z}$
ó $W \cdot \overline{X} \cdot \overline{Y} + \overline{W} \cdot X \cdot Z + \overline{W} \cdot Y + \overline{W} \cdot \overline{X} \cdot \overline{Z}$

# Fin...

Fin...