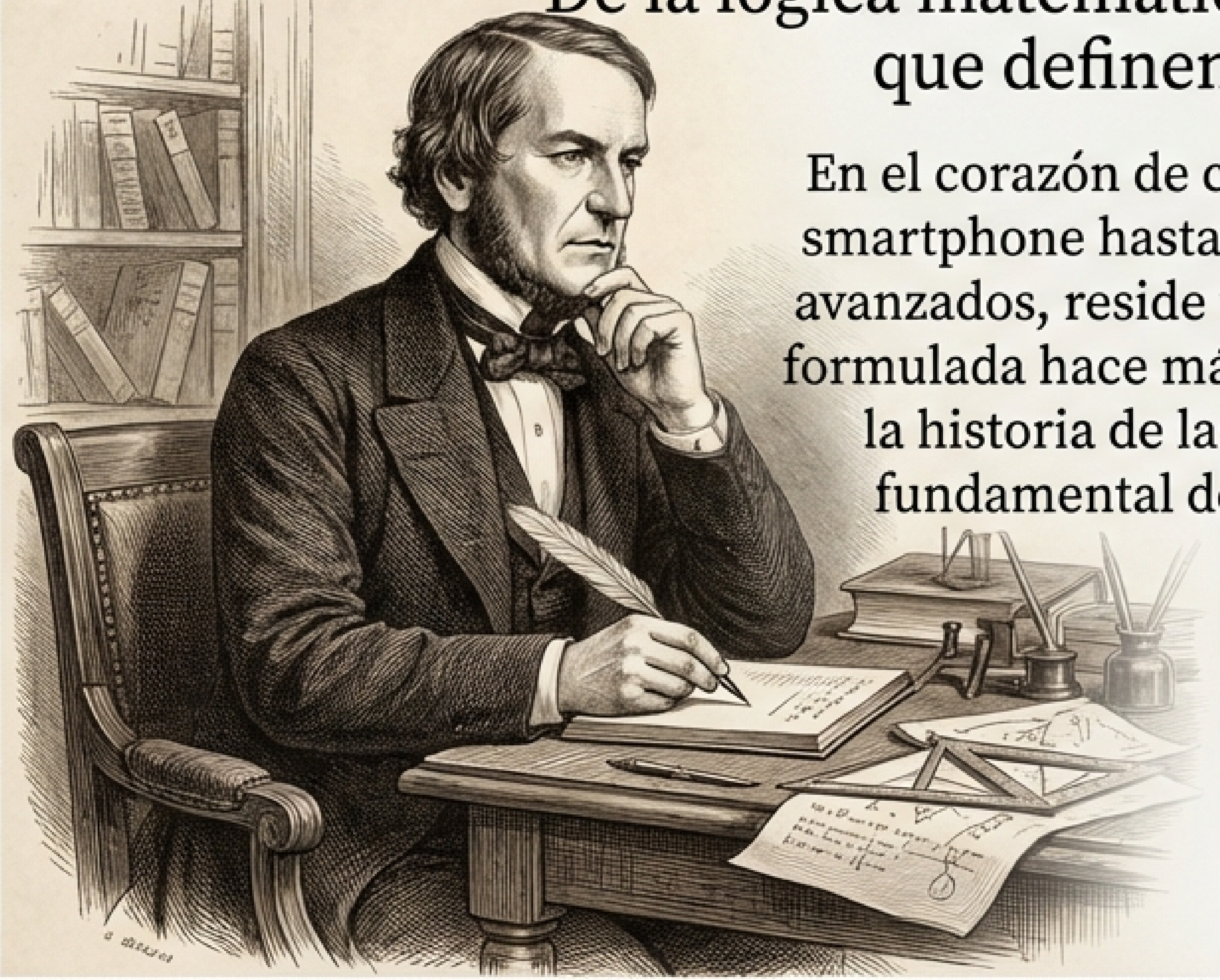
Álgebra de Boole: El Lenguaje que Construyó el Mundo Digital



De la lógica matemática del siglo XIX a los circuitos que definen nuestro presente.

En el corazón de cada sistema digital, desde tu smartphone hasta los supercomputadores más avanzados, reside una idea elegante y poderosa formulada hace más de 150 años. Esta no es solo la historia de las matemáticas, es el plano fundamental de nuestra era tecnológica.

Universidad Andina del Cusco

Título: Todo Comienza con Dos Opciones: 0 y 1

Toda la complejidad de la computación se reduce a la representación de información con solo dos estados. Esta es la base de la lógica digital.

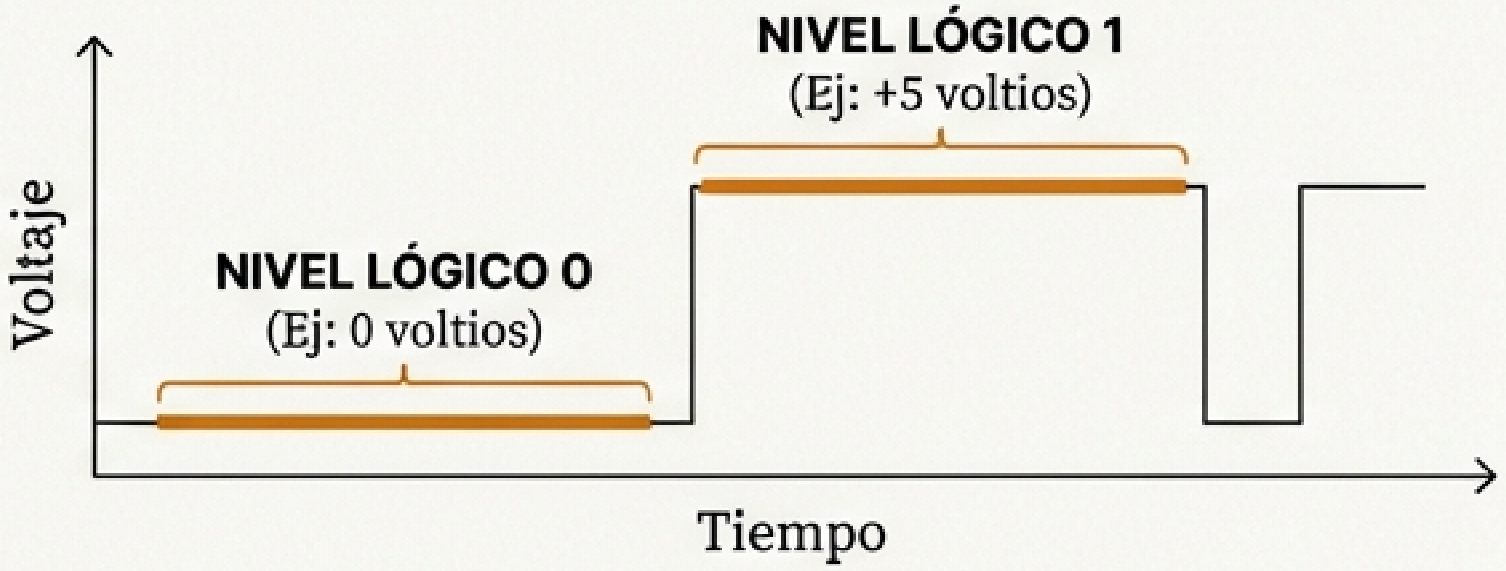
La Lógica Abstracta

0 Representa Falso, Apagado, No.

1 Representa Verdadero, Encendido, Sí.

Una **variable lógica** (A, B, C...) es un símbolo que solo puede tomar uno de estos dos valores.

La Realidad Física



Estos valores lógicos corresponden directamente a niveles de una magnitud eléctrica en un circuito, comúnmente el voltaje.

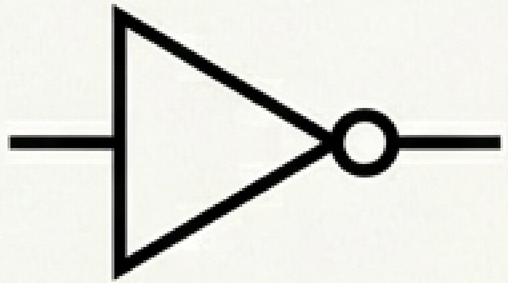
El componente electrónico clave que produce estos estados es el **transistor**, que actúa como un interruptor microscópico controlado eléctricamente.

Universidad Andina del Cusco

Título: Los Tres Pilares de la Lógica Digital

En el siglo XX, Claude Shannon demostró que cualquier operación lógica, sin importar su complejidad, puede construirse a partir de solo tres operaciones básicas. Estas se materializan en circuitos físicos llamados **puertas lógicas**.

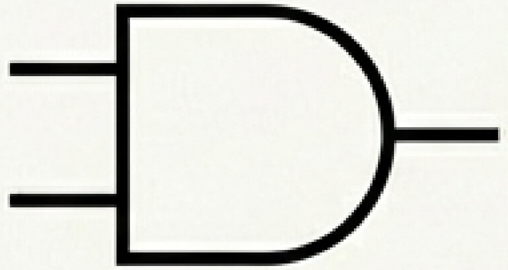
NEGACIÓN (NOT)



Función: Invierte un valor lógico.

Lógica: Si la entrada es **1**, la salida es **0**. Si es **0**, es **1**.

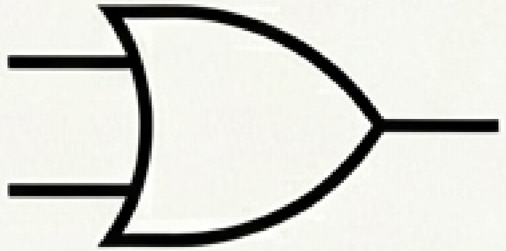
PRODUCTO LÓGICO (AND)



Función: El resultado es **1** solo si *todas* las entradas son **1**.

Lógica: Actúa como una condición de 'todo o nada'.

SUMA LÓGICA (OR)



Función: El resultado es **1** si *alguna* de las entradas es **1**.

Lógica: Actúa como una condición imlusiva de 'cualquiera es suficiente'.
Lógica: Actúa como una condi inclusiva de 'cualquiera es suficiente'.

Nota del Productor: *Estas puertas son los bloques de construcción elementales. En las siguientes diapositivas, analizaremos cada una en detalle.*

Título: El Inversor (Puerta NOT): La Simple Inversión

La puerta NOT o inversor es el elemento más simple. Su única función es invertir el nivel lógico de su entrada.

La Lógica (Abstracto)

Expresión Booleana

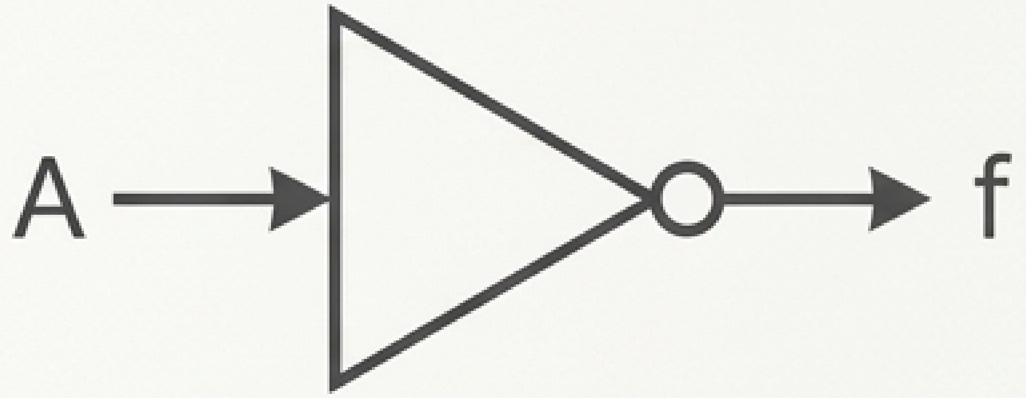
$$f = A'$$

Tabla de Verdad

A	f = A'
0	1
1	0

El Circuito (Concreto)

Símbolo Estándar



Funcionamiento

Un transistor en el circuito actúa como un interruptor. Cuando la entrada A es '1' (nivel alto), el interruptor se cierra y conecta la salida a tierra, resultando en '0' (nivel bajo). Cuando A es '0', el interruptor se abre y la salida se conecta a a la fuente de voltaje, resultando en '1'.

Nota del Productor:: ¡Construye esto en Logisim! Usa un 'Input Pin' para A y un 'Output Pin' para f. Observa cómo la salida cambia instantáneamente al conmutar la entrada.

Título: La Puerta AND: Solo si Todo es Verdadero

La puerta AND implementa el producto lógico. Su salida es '1' únicamente cuando todas y cada una de sus entradas son '1'.

La Lógica (Abstracto)

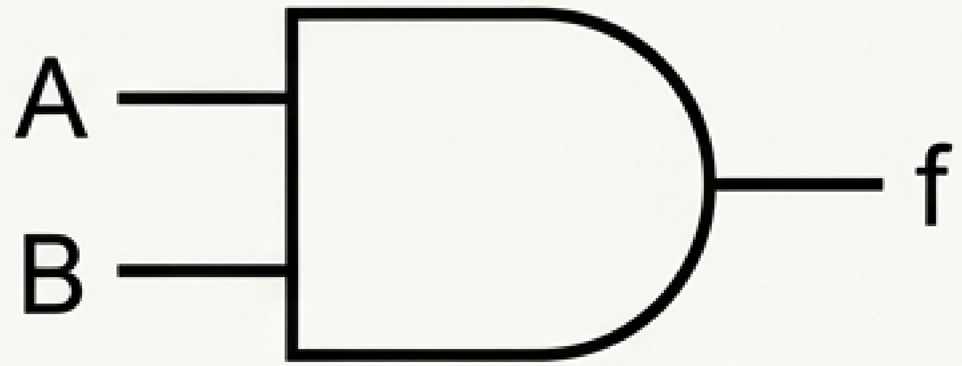
Expresión Booleana

$$f = A \cdot B$$

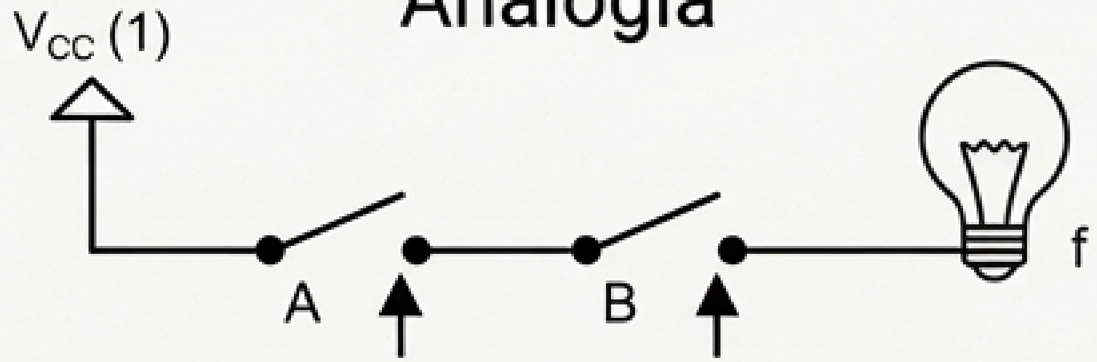
$$f = AB$$

A	B	f = AB
0	0	0
0	1	0
1	0	0
1	1	1

El Circuito (Concreto)



Analogía



Imagina un circuito con dos interruptores en serie. La bombilla solo se encenderá si el interruptor A **Y** el interruptor B están cerrados.

Nota del Productor: Simula una puerta AND de dos entradas en Logisim y verifica cada fila de la tabla de verdad. Luego, cambia el atributo de la puerta a **3 entradas** y observa cómo se expande la tabla de verdad.

Título: La Puerta OR: Basta con que Uno sea Verdadero

La puerta OR implementa la suma lógica. Su salida es '1' si al menos una de sus entradas es '1'.

La Lógica (Abstracto)

Expresión Booleana

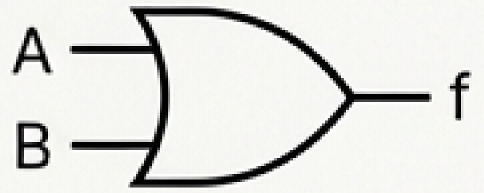
$$f = A + B$$

Tabla de Verdad

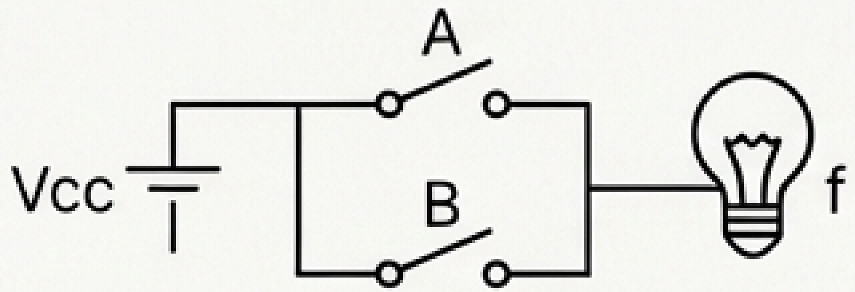
A	B	f = A+B
0	0	0
0	1	1
1	0	1
1	1	1

El Circuito (Concreto)

Símbolo Estándar



Analogía

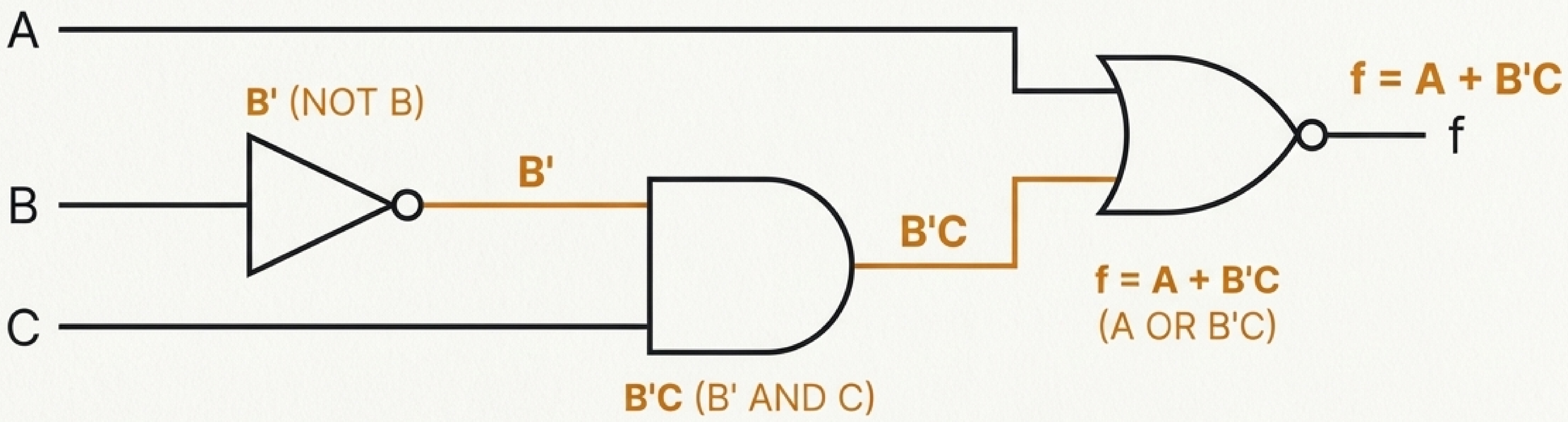


Imagina un circuito con dos interruptores en paralelo. La bombilla se encenderá si el interruptor A **O** el interruptor B (o ambos) están cerrados.

Nota del Productor: Compara el comportamiento de este circuito en Logisim con el de la puerta AND. La diferencia en la lógica se vuelve tangible.

Título: Combinando los Bloques: El Lenguaje de los Circuitos

Cuerpo: El genio de Claude Shannon fue conectar el Álgebra de Boole con los circuitos de conmutación. Cada expresión booleana tiene un circuito lógico equivalente, y viceversa. Esto nos permite diseñar formalmente cualquier sistema digital.



Cita Clave
"Cualquier resultado obtenido por aplicación del álgebra de Boole se puede reproducir mediante el empleo de los circuitos lógicos equivalentes a las expresiones booleanas implicadas."

Título: Las Reglas que Rigen la Lógica Digital

¿Por qué necesitamos reglas? Para simplificar expresiones lógicas. Una expresión más simple resulta directamente en un circuito con menos puertas lógicas.

El Resultado: Circuitos más **rápidos** (menos retardo), más **baratos** (menos componentes) y que consumen **menos energía**.

Caja de Herramientas del Álgebra de Boole		
Identidad	$A + 0 = A$	$A \cdot 1 = A$
Elemento Nulo	$A + 1 = 1$	$A \cdot 0 = 0$
Idempotencia	$A + A = A$	$A \cdot A = A$
Complemento	$A + A' = 1$	$A \cdot A' = 0$
Involución	$(A')' = A$	
Distributiva	$A(B+C) = AB+AC$	$A(B+C) = AB+AC$
Absorción	$A + AB = A$	$A(A+B) = A$
Leyes de De Morgan	$(A + B)' = A' \cdot B'$	$(A \cdot B)' = A' + B'$

Universidad Andina del Cusco

Título: El Principio de Dualidad: La Simetría Oculta de la Lógica

Cuerpo: El Álgebra de Boole tiene una propiedad elegante: el Principio de Dualidad. Si una igualdad booleana es verdadera, su 'dual' también lo es. Se obtiene aplicando dos simples reglas.

Reglas de Transformación Dual:
1. Intercambia las operaciones `·` (AND) por `+` (OR).
2. Intercambia los elementos identidad `0` por `1`.

Expresión Original (Teorema T.2'): `A + 1 = 1`

`+` se convierte en `·`

`1` se convierte en `0`

Expresión Dual Resultante (Teorema T.2). `A · 0 = 0`

Este principio duplica nuestro conocimiento. Si se demuestra la validez de un teorema, automáticamente se asume como cierto su dual, ahorrando esfuerzo de demostración.

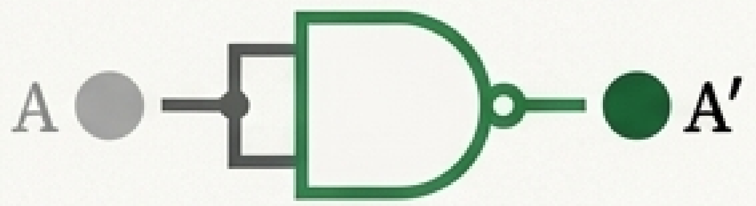
Universidad Andina del Cusco

Título: Menos es Más: La Universalidad de NAND y NOR

Cuerpo: Sorprendentemente, no es necesario fabricar puertas AND, OR y NOT. Las puertas NAND (NOT-AND) y NOR (NOT-OR) son "universales": cada una, por sí sola, puede ser utilizada para construir cualquier otra función lógica.

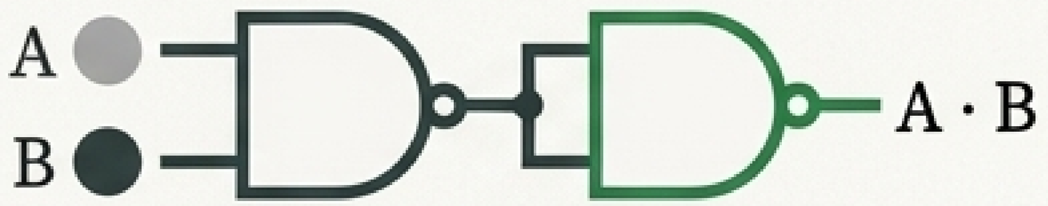
Construyendo Todo con Puertas NAND:

Implementación de NOT



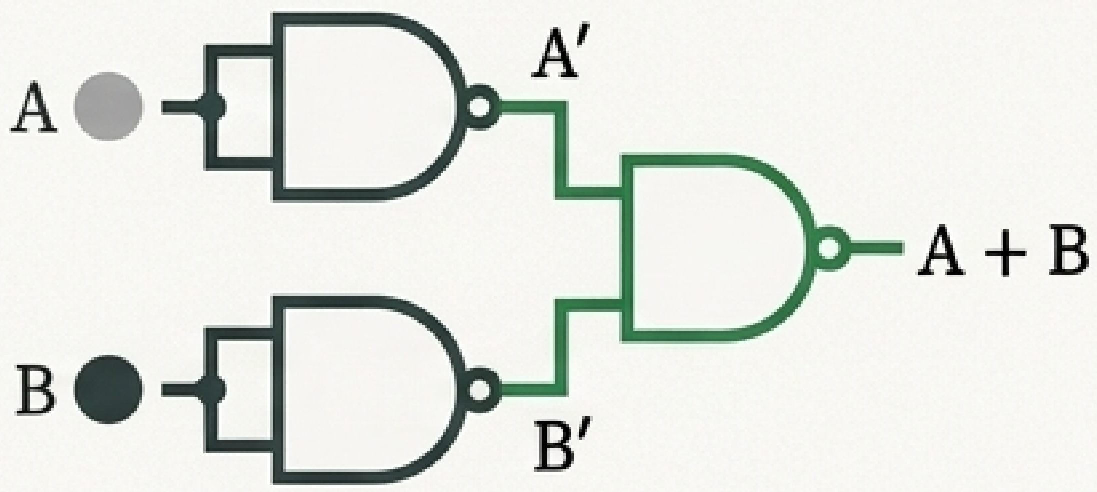
$A' = (A \cdot A)'$

Implementación de AND



$A \cdot B = ((A \cdot B)')'$

Implementación de OR



$A + B = (A' \cdot B')'$

¿Por qué es importante? En la fabricación de circuitos integrados, es mucho más eficiente y económico producir en masa un solo tipo de puerta universal.

Universidad Andina del Cusco

Título: Formas Canónicas: Los Planos Maestros de una Función

Cuerpo: Para analizar y diseñar sistemas de forma sistemática, necesitamos una manera estándar de expresar cualquier función lógica. Las dos formas canónicas se derivan directamente de la tabla de verdad.

1. Primera Forma Canónica (Suma de Productos - SOP)

- Se construye sumando (+) Minitérminos.
- Un Minitérmino es un producto de todas las variables que corresponde a una combinación de entrada donde la función es 1.
- Notación Abreviada: $f = \sum m(\ldots)$, donde los números son los índices de las filas con salida 1.

2. Segunda Forma Canónica (Producto de Sumas - POS)

- Se construye multiplicando (·) Maxitérminos.
- Un Maxitérmino es una suma de todas las variables que corresponde a una combinación de entrada donde la función es 0.
- Notación Abreviada: $f = \prod M(\ldots)$, donde los números son los índices de las filas con salida 0.

Id.	A	B	C	Minitérmino	Maxitérmino
0	0	0	0	$m_0 = A'B'C'$	$M_0 = A+B+C$
1	0	0	1	$m_1 = A'B'C$	$M_1 = A+B+C'$
2	0	1	0	$m_2 = A'BC'$	$M_2 = A+B'+C$
3	0	1	1	$m_3 = A'BC$	$M_3 = A+B'+C'$
4	1	0	0	$m_4 = AB'C'$	$M_4 = A'+B+C$
5	1	0	1	$m_5 = AB'C$	$M_5 = A'+B+C'$
6	1	1	0	$m_6 = ABC'$	$M_6 = A'+B'+C$
7	1	1	1	$m_7 = ABC$	$M_7 = A'+B'+C'$

Universidad Andina del Cusco

Título: La Tabla de Verdad: El Origen de Todo Diseño

Cuerpo: La tabla de verdad (TV) es la descripción más fundamental de una función lógica. Define el valor de la salida para *cada* posible combinación de las entradas. Es el punto de partida y la fuente de todo conocimiento sobre un circuito.

Tabla de Verdad (TV)

Id.	A	B	C	f
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	0	1
4	1	0	1	0
5	1	0	0	1
6	1	1	1	0
7	1	1	1	1

El Flujo Directo de la TV a la Expresión

Forma Canónica SOP

$$f = A'B'C + A'BC + AB'C + ABC$$

Forma Canónica POS

$$f = (A+B+C) \cdot (A+B'+C) \cdot (A'+B+C) \cdot (A'+B'+C)$$

El Flujo Directo de la TV a la Expresión:

1. **Define el Comportamiento:** Primero, llena la columna de salida (f) con los '1's y '0's que resuelven tu problema.
2. **Extrae la Forma Canónica SOP:** Identifica todas las filas donde `f = 1`. La expresión SOP es la suma lógica (OR) de los minitérminos de esas filas.
3. **Extrae la Forma Canónica POS:** Identifica todas las filas donde `f = 0`. La expresión POS es el producto lógico (AND) de los maxitérminos de esas filas.

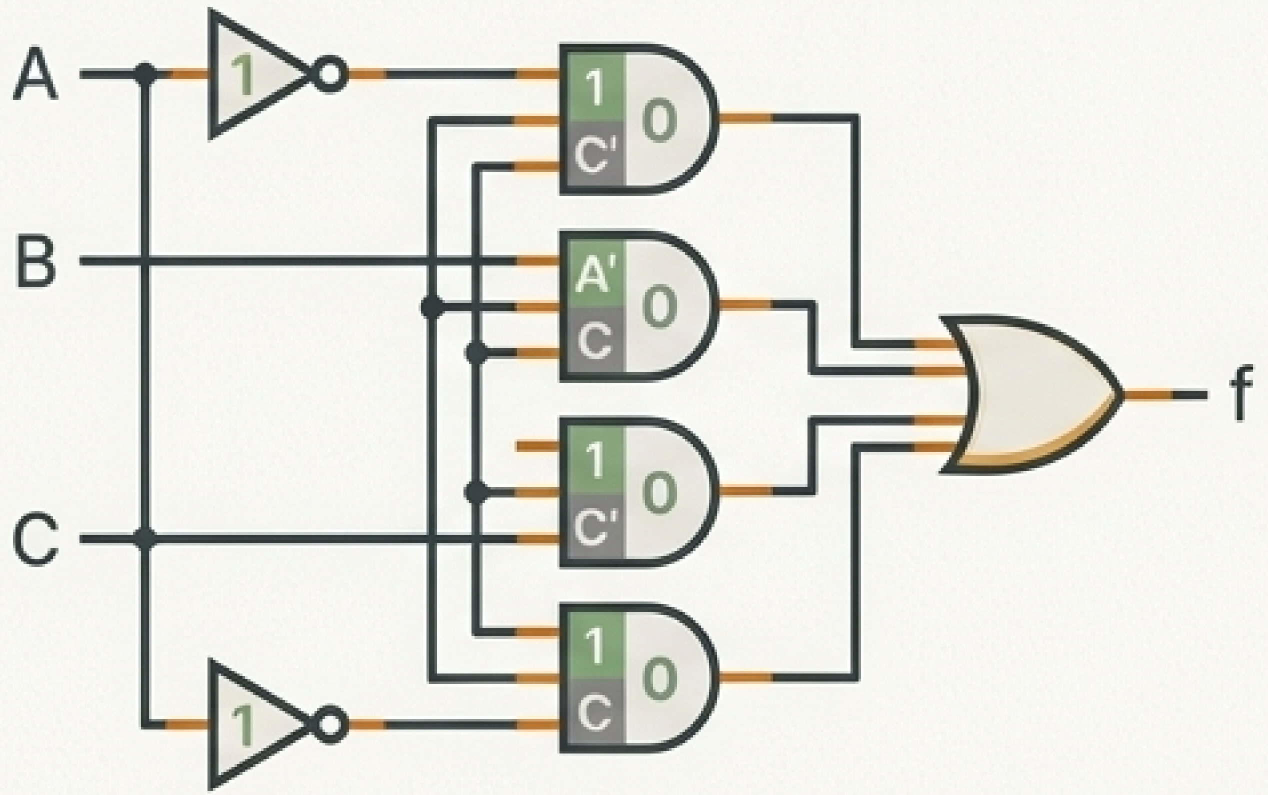
Universidad Andina del Cusco

Título: La Meta de la Ingeniería: Simplificar para Optimizar

Las formas canónicas son matemáticamente completas, pero rara vez son la implementación más eficiente. La simplificación es un paso crucial en la ingeniería de hardware.

Expresión Canónica (Completa pero Ineficiente)

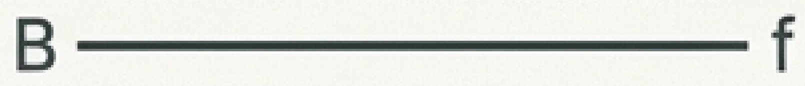
$$f(A,B,C) = A'BC' + A'BC + ABC' + ABC$$



Requiere 4 puertas AND de 3 entradas y 1 puerta OR de 4 entradas. Es más grande, más lento y consume más energía.

Expresión Simplificada (Óptima)

$$f(A,B,C) = B$$



Un simple cable desde la entrada B a la salida f. Máxima eficiencia.

La simplificación algebraica reduce directamente el costo y aumenta el rendimiento del hardware físico.

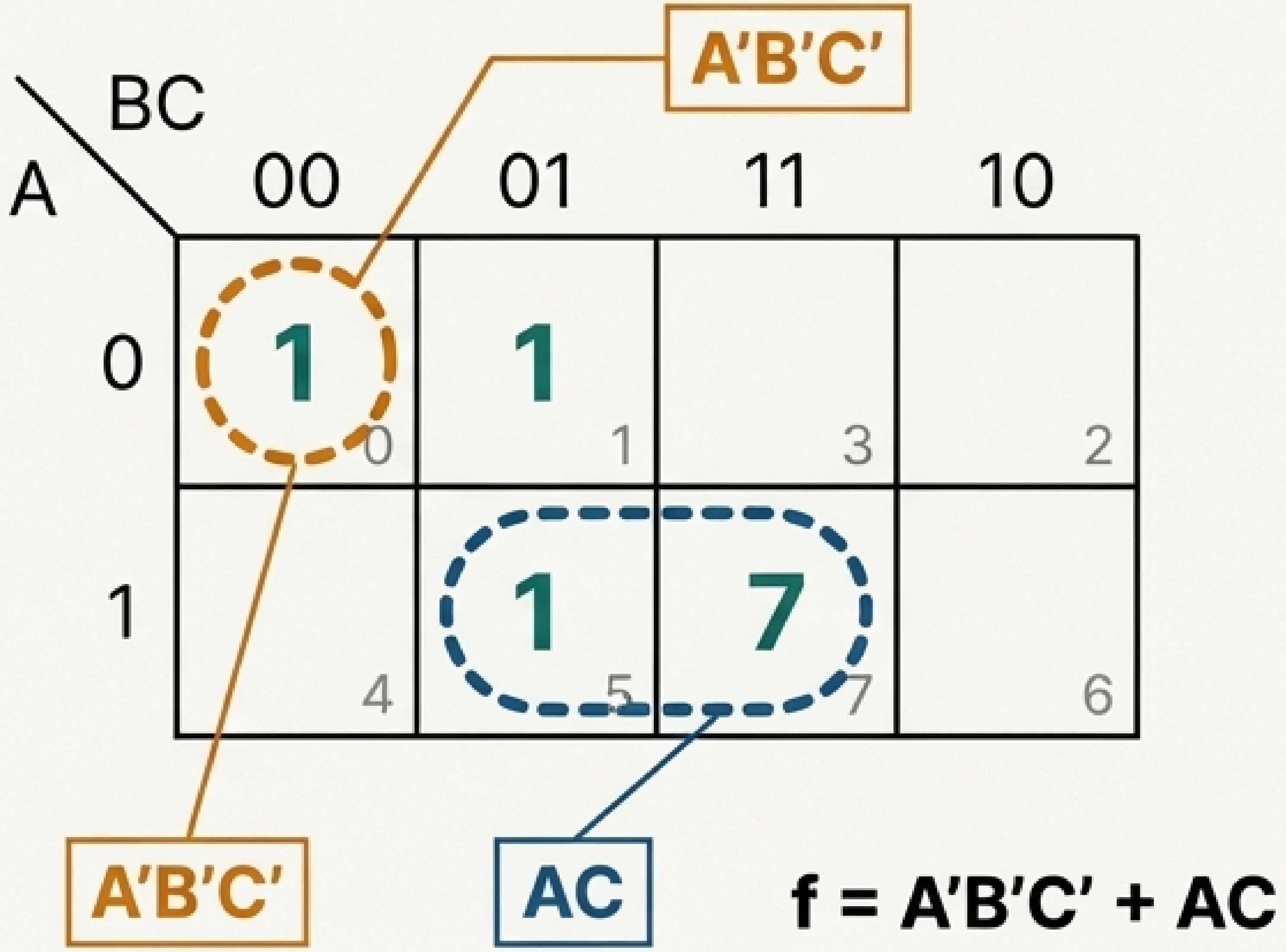
Título: Mapas de Karnaugh: Simplificación Visual e Intuitiva

Cuerpo: Los Mapas de Karnaugh (K-maps) son una reorganización gráfica de la tabla de verdad que permite encontrar la expresión más simple de forma visual. La clave es agrupar los ‘1’s adyacentes para eliminar variables redundantes.

El Método (para SOP):

1. **Llenar el Mapa:** Coloca un ‘1’ en cada celda que corresponde a un minitérmino de la función.
2. **Agrupar:** Crea los grupos rectangulares más grandes posibles de ‘1’s adyacentes. Los grupos deben tener un tamaño de 2, 4, 8, etc. Los bordes del mapa se consideran adyacentes (“se envuelven”).
3. **Extraer Términos:** Cada grupo genera un término producto. Las variables que cambian de valor (de 0 a 1 o viceversa) dentro del grupo se eliminan.
4. **Sumar:** La función simplificada es la suma (OR) de los términos producto obtenidos de cada grupo.

Ejemplo Visual



Título: El Proceso Completo: De la Idea al Circuito Lógico

Hemos ensamblado todas las piezas. El Álgebra de Boole nos proporciona un método riguroso y completo para diseñar cualquier sistema digital combinacional, transformando un requisito en un circuito funcional.

El Flujo de Trabajo del Ingeniero Digital:

Especificación
Entender y definir el problema a resolver.

Tabla de Verdad
Formalizar el comportamiento deseado para todas las entradas posibles.

SOP
POS

Expresión Canónica
Derivar una primera expresión matemática completa (SOP o POS).

Simplificación (K-Map)
Optimizar la expresión para obtener la máxima eficiencia en costo y velocidad.

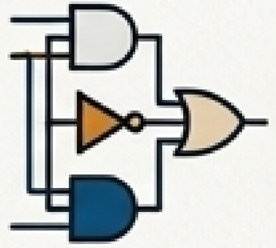
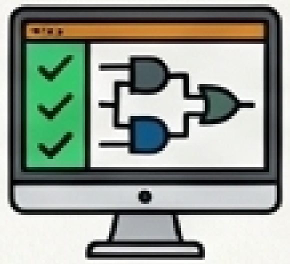


Diagrama Lógico
Dibujar el circuito final utilizando los símbolos de las puertas lógicas.



Simulación y Verificación
Usar herramientas como **Logisim** para probar virtualmente que el circuito se comporta como se esperaba antes de su implementación física.

Con solo dos valores (0, 1) y un puñado de reglas lógicas, hemos construido un sistema capaz de describir, analizar y crear el complejo universo digital que nos rodea.

Universidad Andina del Cusco