Curso
Matemáticas Discretas I

UNIVERSIDAD
DE ANTIOQUIA

Clase 12 – Algebra de Boole

Agenda

UNIVERSIDAD
DE ANTIOQUIA

- Contextualización
- Circuitos lógicos
- Introducción al Algebra Booleana
- Funciones Booleanas
- Representación de funciones Booleanas

Agenda

- **Contextualización**
- Circuitos lógicos
- Introducción al Algebra Booleana
- Funciones Booleanas
- Representación de funciones Booleanas

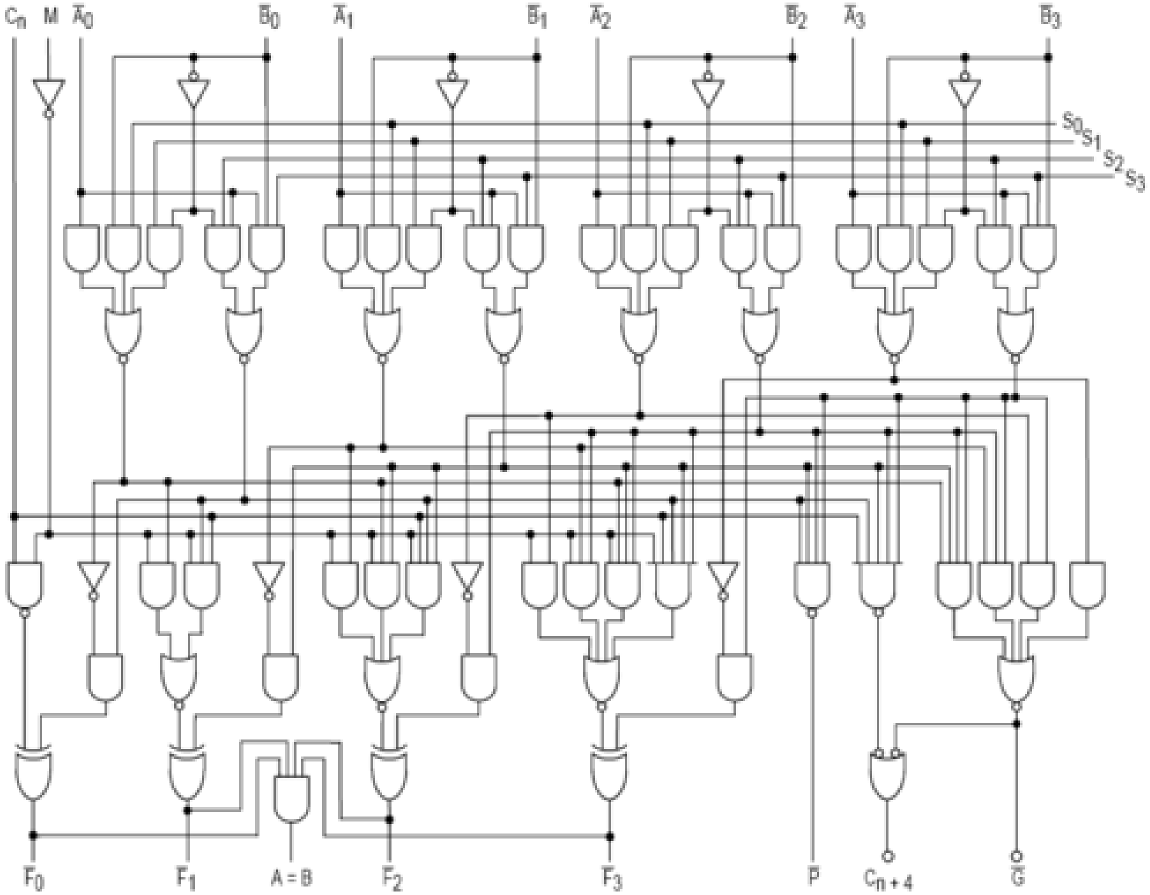
Contextualización

UNIVERSIDAD DE ANTIOQUIA

Relación con las próximas materias

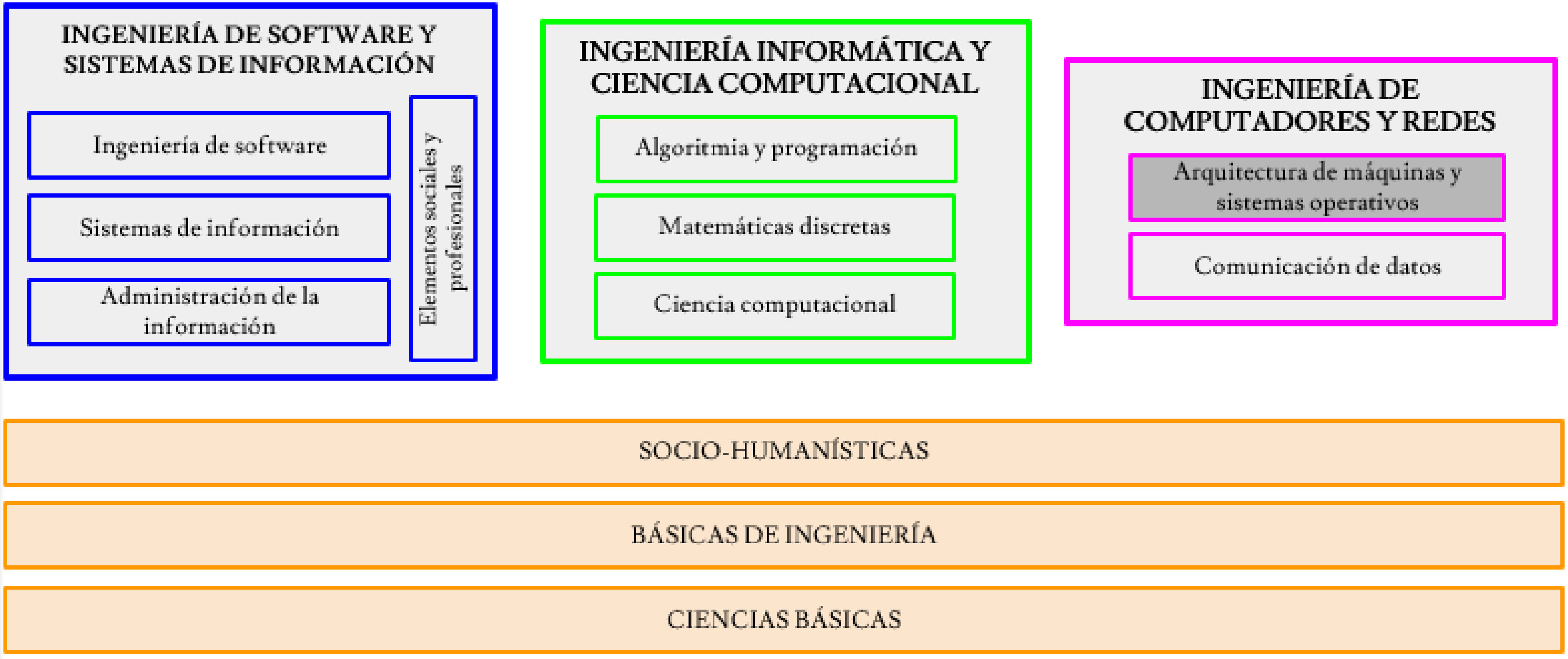
- Aplicación
- Lenguaje de alto nivel
- Sistema operativo / Compilador
- Arquitectura del conjunto de instrucciones
- Organización HW del sistema
- **Circuito digital**
- Nivel físico

Matemáticas



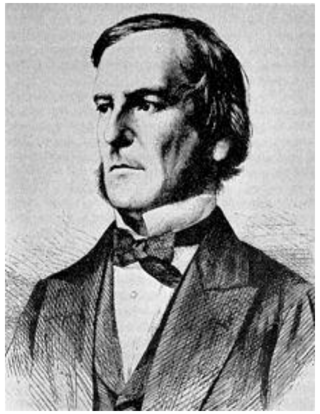
Contextualización

Relación con las próximas materias

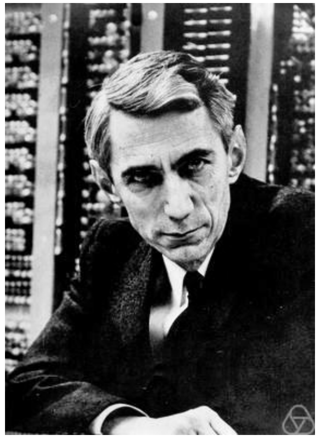


Contextualización

Protagonistas



George Boole [link]



Claude Shannon [link]

	George Boole	Claude Shannon
Época	Siglo XIX (1815–1864)	Siglo XX (1916–2001)
Disciplina principal	Matemáticas, lógica	Ingeniería eléctrica, matemáticas aplicadas
Aporte clave	Creación del álgebra booleana	Aplicación del álgebra booleana a circuitos eléctricos
Obra destacada	*An Investigation of the Laws of Thought* (1854) [link]	*A Symbolic Analysis of Relay and Switching Circuits* (1938) [link]
Enfoque	Formalización de la lógica mediante símbolos algebraicos	Traducción de la lógica simbólica a sistemas físicos
Impacto inmediato	Base teórica para la lógica y las matemáticas modernas	Base práctica para la computación digital
Legado	Fundador de la lógica matemática moderna	Fundador de la teoría de la información y la computación digital

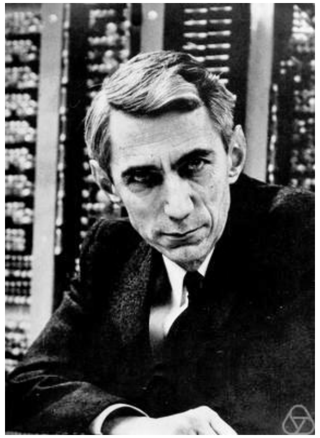
Contextualización

UNIVERSIDAD DE ANTIOQUIA

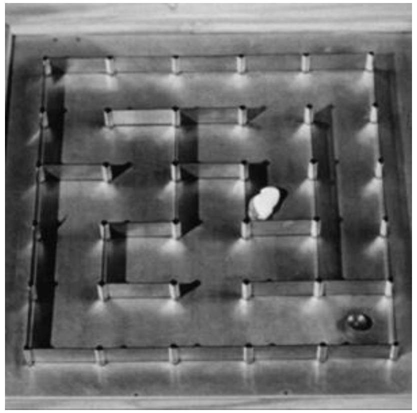
Claude Shannon

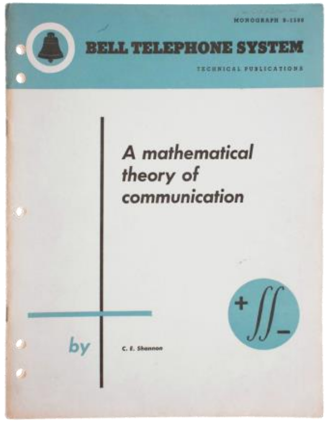
Ver:

- **Claude Shannon, el genial matemático que inventó la era digital (y luego se retiró a su "cuarto de juguetes")** (link)
- **Claude Shannon Explains Information Theory** (link)
- https://thebitplayer.com/



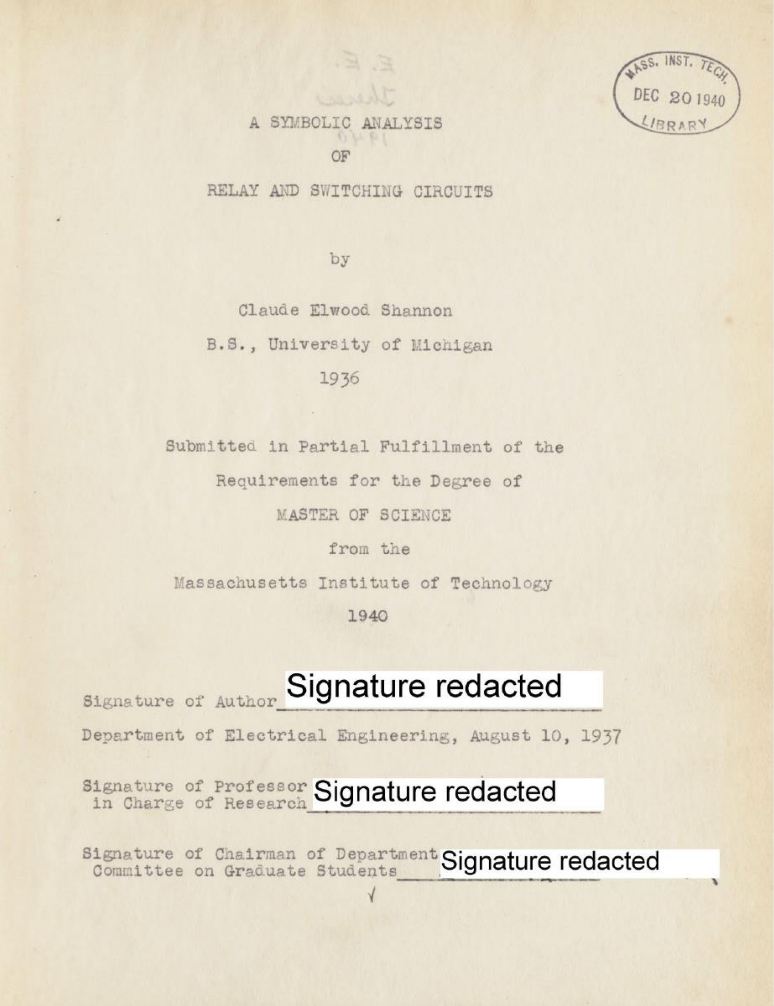
Claude Shannon [link]





Contextualización

Idea de Shannon


MASS. INST. TECH. DEC 20 1940 LIBRARY

A SYMBOLIC ANALYSIS

OF

RELAY AND SWITCHING CIRCUITS

by

Claude Elwood Shannon

B.S., University of Michigan

1936

Submitted in Partial Fulfillment of the
Requirements for the Degree of
MASTER OF SCIENCE
from the
Massachusetts Institute of Technology
1940

Signature of Author Signature redacted
Department of Electrical Engineering, August 10, 1937

Signature of Professor in Charge of Research Signature redacted

Signature of Chairman of Department Committee on Graduate Students Signature redacted

A Symbolic Analysis of Relay and Switching Circuits [link]

Idea Revolucionaria:
Shannon se dio cuenta de que el comportamiento binario (abierto/cerrado) de los relés en un circuito eléctrico era análogo al comportamiento de las proposiciones (verdadero/falso) en el álgebra de George Boole, desarrollada casi un siglo antes.

Table I. Analogue Between the Calculus of Propositions and the Symbolic Relay Analysis

Symbol	Interpretation in Relay Circuits	Interpretation in the Calculus of Propositions
X	The circuit X	The proposition X
0	The circuit is closed	The proposition is false
1	The circuit is open	The proposition is true
$X + Y$	The series connection of circuits X and Y	The proposition which is true if either X or Y is true
$X\,Y$	The parallel connection of circuits X and Y	The proposition which is true if both X and Y are true
X'	The circuit which is open when X is closed and closed when X is open	The contradictory of proposition X
=	The circuits open and close simultaneously	Each proposition implies the other

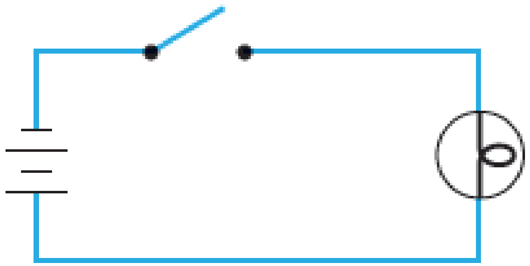
Contextualización

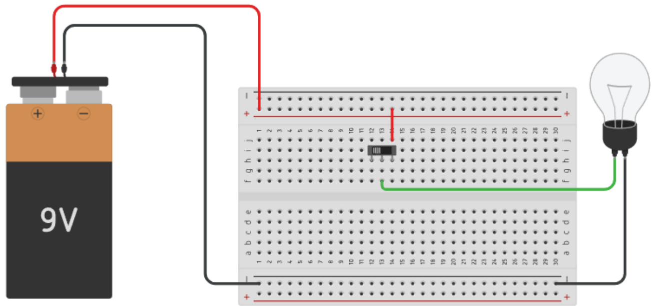
UNIVERSIDAD DE ANTIOQUIA

La Analogía Fundamental: Lógica Proposicional y Circuitos

Shannon demostró que se podía usar el álgebra de Boole para modelar y analizar cualquier circuito de conmutación.

Algebra de Boole	Circuitos Eléctricos
Una proposición **Verdadera (True / 1)**	Un interruptor **cerrado** (la corriente pasa)
Una proposición **Falsa (False / 0)**	Un interruptor **abierto** (la corriente no pasa)
El operador lógico **Y (AND / ∧)**	Dos interruptores conectados **en serie**
El operador lógico **O (OR / ∨)**	Dos interruptores conectados **en paralelo**
El operador lógico **NO (NOT / ¬)**	Un relé "normalmente cerrado" (se abre cuando se activa)



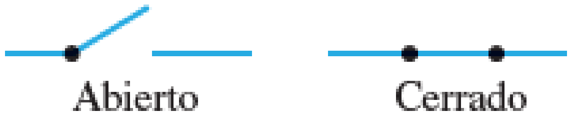


Contextualización

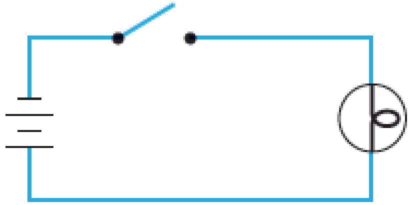
UNIVERSIDAD DE ANTIOQUIA

La Analogía Fundamental: Lógica Proposicional y Circuitos

Interruptor simple (estados)



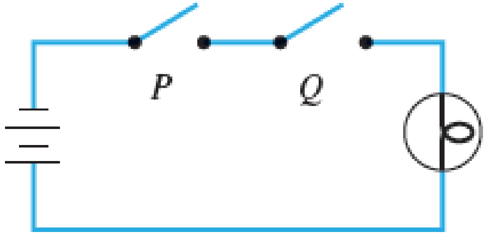
Circuito lógico



Interruptores en serie (AND)

Interruptores en serie

Interruptores		Foco
P	*Q*	Estado
cerrado	cerrado	encendido
cerrado	abierto	apagado
abierto	cerrado	apagado
abierto	abierto	apagado

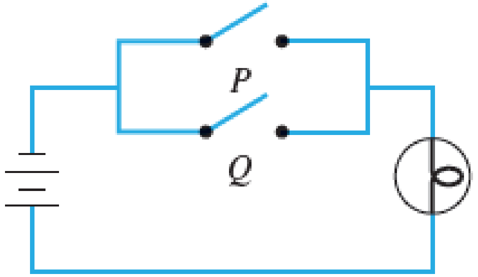


Interruptores “en serie”

Interruptores en seria (OR)

Interruptores en paralelo

Interruptores		Foco
P	*Q*	Estado
cerrado	cerrado	encendido
cerrado	abierto	encendido
abierto	cerrado	encendido
abierto	abierto	apagado

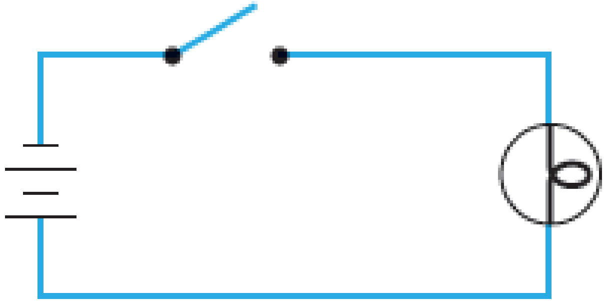


Interruptores “en paralelo”

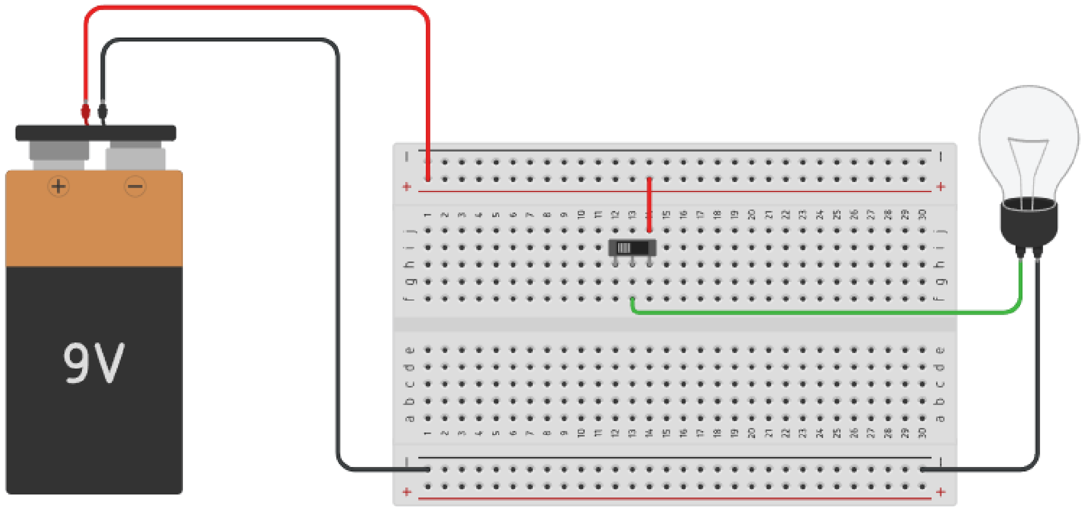
Contextualización

La Analogía Fundamental: Lógica Proposicional y Circuitos

Circuito digital



Variable lógica	Interruptor	Foco
1 (V)	Cerrado	Encendido
0 (F)	Abierto	Apagado



Circuito digital [link]

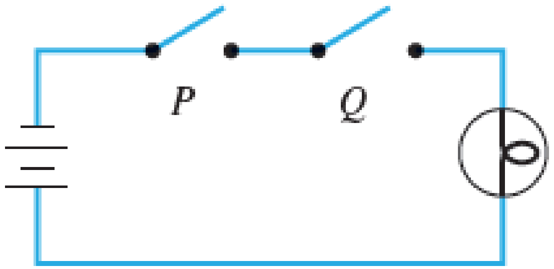
Contextualización

UNIVERSIDAD DE ANTIOQUIA

La Analogía Fundamental: Lógica Proposicional y Circuitos

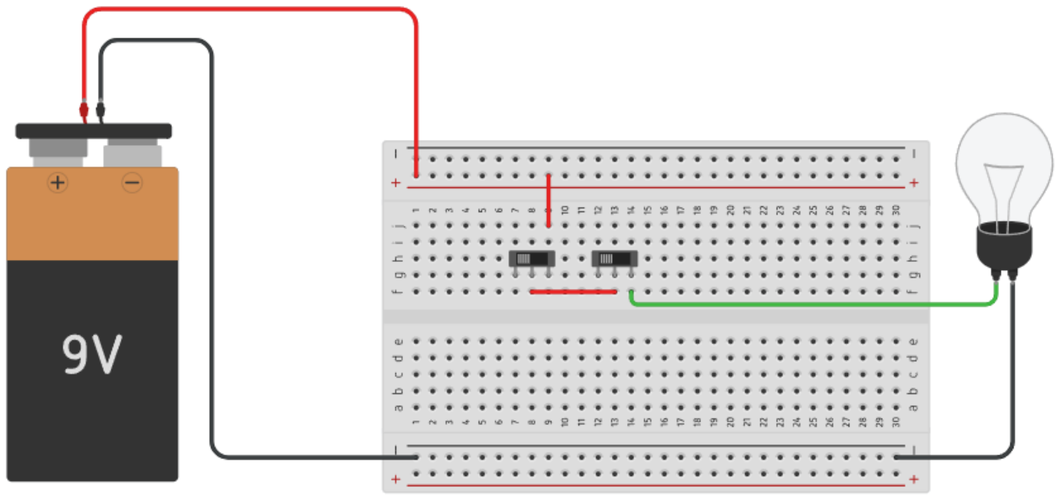
Circuito digital – Función AND

Variable lógica	Interruptor	Foco
1 (V)	Cerrado	Encendido
0 (F)	Abierto	Apagado



Interruptores “en serie”

Interruptor (P)	Interruptor (Q)	Foco (F)
0	0	0
0	1	0
1	0	1
1	1	1

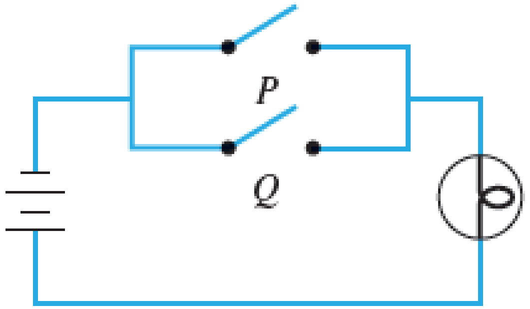


Función AND [link]

La Analogía Fundamental: Lógica Proposicional y Circuitos

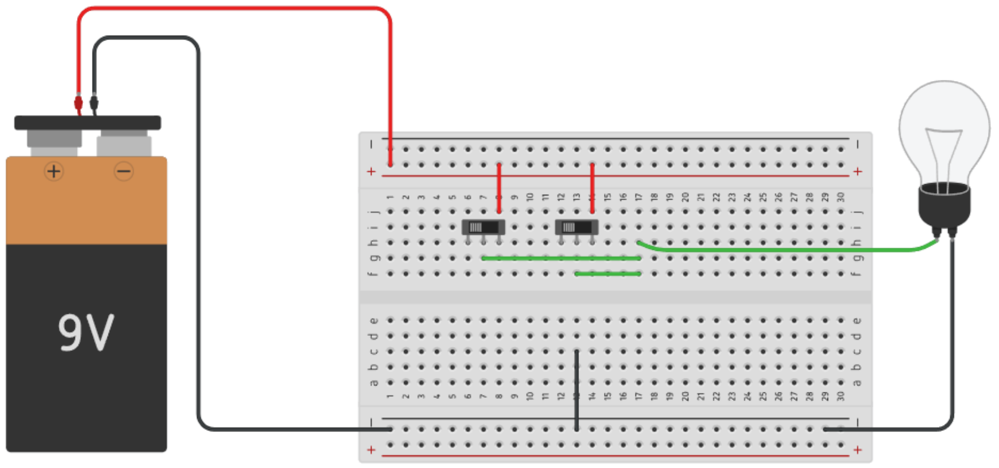
Circuito digital – Función OR

Variable lógica	Interruptor	Foco
1 (V)	Cerrado	Encendido
0 (F)	Abierto	Apagado



Interruptores "en paralelo"

Interruptor (P)	Interruptor (Q)	Foco (F)
0	0	0
0	1	1
1	0	1
1	1	1



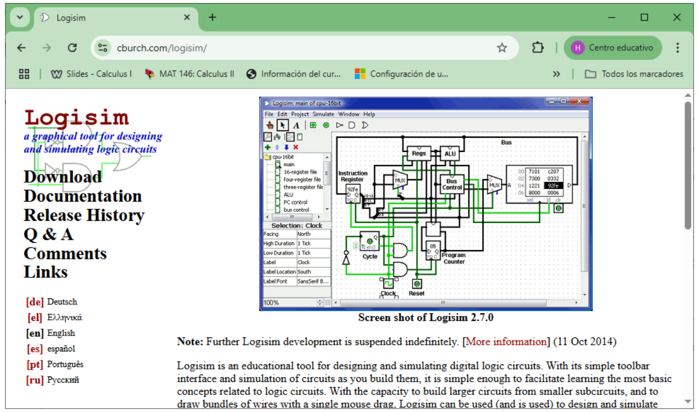
Función OR [link]

Contextualización

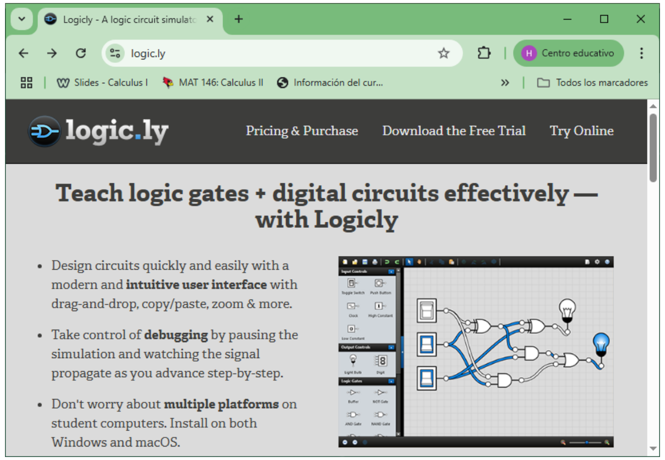
UNIVERSIDAD DE ANTIOQUIA

Simuladores

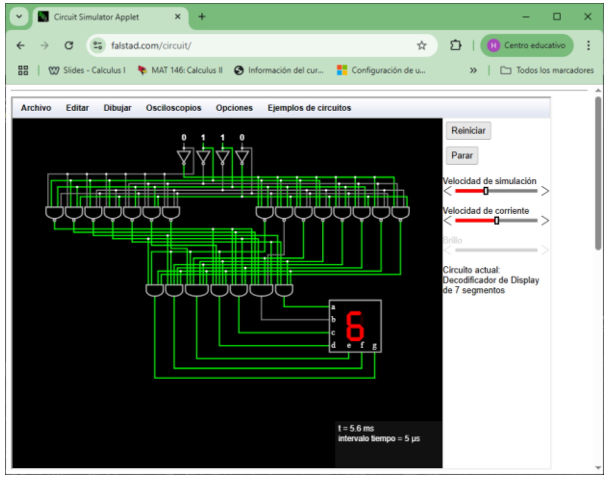
Logisim



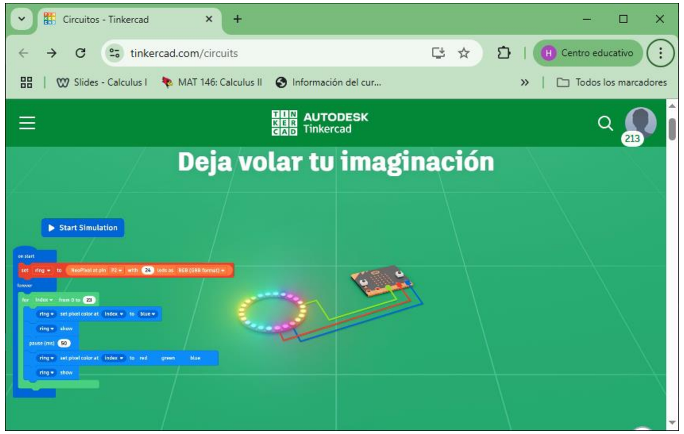
https://www.cburch.com/logisim/



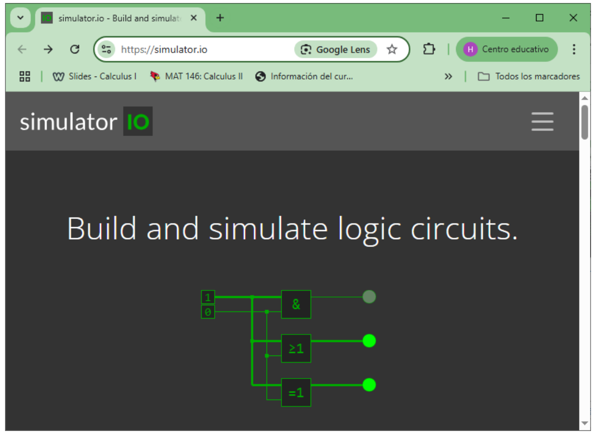
https://logic.ly/



https://www.falstad.com/circuit/



https://www.tinkercad.com/



https://simulator.io/

Agenda

- Contextualización
- **Circuitos lógicos**
- Introducción al Algebra Booleana
- Funciones Booleanas
- Representación de funciones Booleanas

Circuitos Lógicos

UNIVERSIDAD DE ANTIOQUIA

Introducción

- Un **circuito lógico** (o **circuito digital**) recibe señales de entrada $p_1, p_2, \dots, p_n$ y produce señales de salida $s_1, s_2, \dots, s_n$. Un circuito digital puede ser visto puede ser representado como una caja negra:

Una **caja negra** es un modelo que se usa para representar un sistema cuyo funcionamiento interno no se conoce, pero del cual sí se observan sus entradas y salidas.

- El interior de una caja negra contiene la implementación detallada del circuito y a menudo se ignora mientras la atención se centra en la relación entre las señales de entrada y salida.
- En los circuitos lógicos, cada entrada/salida puede ser vista como un **0** o **1**.
  - **0**: Representa Falso
  - **1**: Representa Verdadero
- Los circuitos lógicos complicados se construyen a partir de circuitos básicos llamados **compuertas lógicas**.

Circuitos Lógicos

Caja negra

El funcionamiento de una caja negra se especifica completamente mediante la construcción de una **tabla de entrada/salida** (tabla de verdad) que enumera todas sus posibles señales de entrada junto con sus correspondientes señales de salida.

Caja negra

Input			Output
P	*Q*	*R*	*S*
1	1	1	1
1	1	0	0
1	0	1	0
1	0	0	1
0	1	1	0
0	1	0	1
0	0	1	1
0	0	0	0

Tabla de entrada/salida

Tercera fila: Indica que para las entradas

- P = 1
- Q = 0
- R = 1

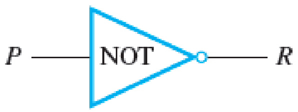
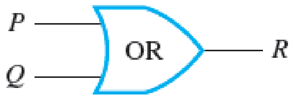
La salida S es 0.

Circuitos Lógicos

Compuertas lógicas

Los circuitos lógicos complicados se construyen a partir de **circuitos básicos** llamados **compuertas lógicas**. Veamos algunas:

- **Compuerta NOT (inversor)**: es un circuito con una señal de entrada y una de salida. Si la señal de entrada es 1, la señal de salida es 0. Por el contrario, si la señal de entrada es 0, la señal de salida es 1.
- **Compuerta AND**: es un circuito con dos señales de entrada y una de salida. Si ambas señales de entrada son 1, la señal de salida es 1. De lo contrario, la señal de salida es 0.
- **Compuerta OR**: también tiene dos señales de entrada y una de salida. Si ambas señales de entrada son 0, la señal de salida es 0. De lo contrario, la señal de salida es 1.

<table>
<tr><th>Type of Gate</th><th>Symbolic Representation</th><th>Action</th></tr>
<tr><td>NOT</td><td>
</td><td>
<table>
<tr><th>Input</th><th>Output</th></tr>
<tr><td>P</td><td>R</td></tr>
<tr><td>1</td><td>0</td></tr>
<tr><td>0</td><td>1</td></tr>
</table></td></tr>
<tr><td>AND</td><td>
</td><td>
<table>
<tr><th colspan="2">Input</th><th>Output</th></tr>
<tr><td>P</td><td>Q</td><td>R</td></tr>
<tr><td>1</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>0</td></tr>
</table></td></tr>
<tr><td>OR</td><td>
</td><td>
<table>
<tr><th colspan="2">Input</th><th>Output</th></tr>
<tr><td>P</td><td>Q</td><td>R</td></tr>
<tr><td>1</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>0</td><td>0</td></tr>
</table></td></tr>
</table>

Circuitos Lógicos

Compuertas lógicas

- En lógica, **variables** como p, q y r representan enunciados, y un enunciado puede tener uno de solo dos valores de verdad: **V** (verdadero) o **F** (falso).
- Un **enunciado** es una expresión, como $p \wedge (\neg q \vee r)$ compuesta de variables del enunciado y conectores lógicos.
- Cualquier variable, como una variable de instrucción o una señal de entrada, que solo puede tomar uno de dos valores se denomina **variable booleana**.
- Una expresión compuesta por variables booleanas y los conectores $\neg$, $\wedge$ y $\vee$ se llama **expresión booleana**.

Circuito Lógico

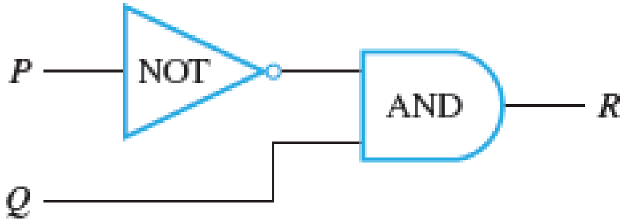


Tabla Entrada/Salida

P	*Q*	*R*
0	0	0
1	0	0
0	1	1
1	1	0

Expresión Booleana

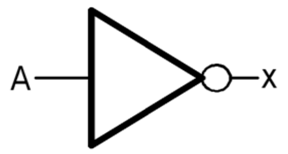
$$R = \neg P \cdot Q$$

Circuitos Lógicos

Compuertas lógicas

A continuación se muestran diferentes tipos de compuertas junto con su tabla de entrada/salida y expresión lógica asociada:

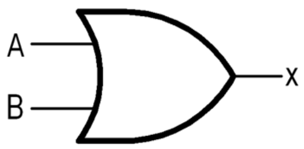
Compuerta NOT



A	$X = A'$
0	1
1	0

$X = A'$

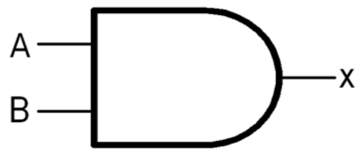
Compuerta OR



A	B	$X = A + B$
0	0	0
1	0	1
0	1	1
1	1	1

$X = A + B$

Compuerta AND



A	B	$X = A \cdot B$
0	0	0
1	0	0
0	1	0
1	1	1

$X = A \cdot B = AB$

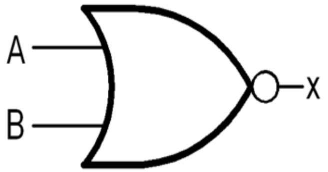
Circuitos Lógicos

UNIVERSIDAD DE ANTIOQUIA

Compuertas lógicas

Además de las compuertas NOT, AND y XOR, existen otras compuertas adicionales a partir de las cuales también se pueden construir circuitos lógicos. Esta se muestran a continuación:

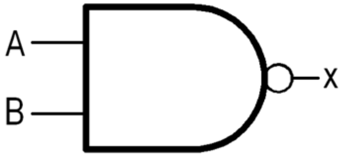
Compuerta NOR



A	B	$X=(A+B)'$
0	0	1
1	0	0
0	1	0
1	1	0

$$X=(A+B)'=A\downarrow B$$

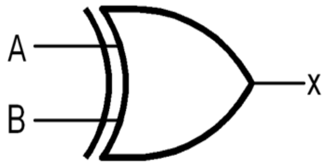
Compuerta NAND



A	B	$X=(A\cdot B)'$
0	0	1
1	0	1
0	1	1
1	1	0

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

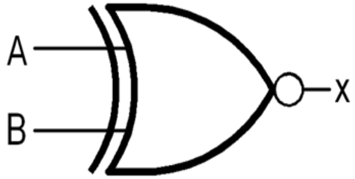
Compuerta XOR



A	B	$X=A\oplus B$
0	0	0
1	0	1
0	1	0
1	1	1

$$X=A\oplus B$$

Compuerta XNOR



A	B	$X=A\odot B$
0	0	1
1	0	0
0	1	0
1	1	1

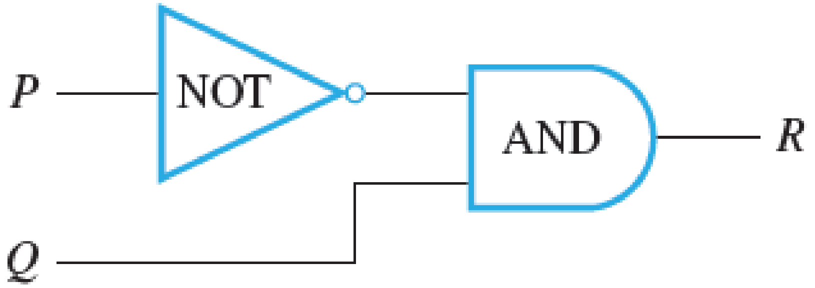
$$X=A\odot B$$

Circuitos Lógicos

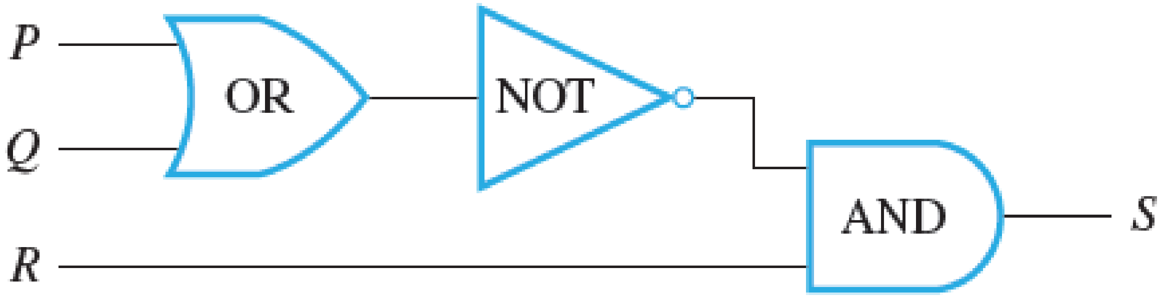
UNIVERSIDAD
DE ANTIOQUIA

Ejemplos

1. Para cada uno de los siguientes circuitos indique la salida la salida para las señales de entrada dadas.



Señales de entrada: $P = 0$ y $Q = 1$

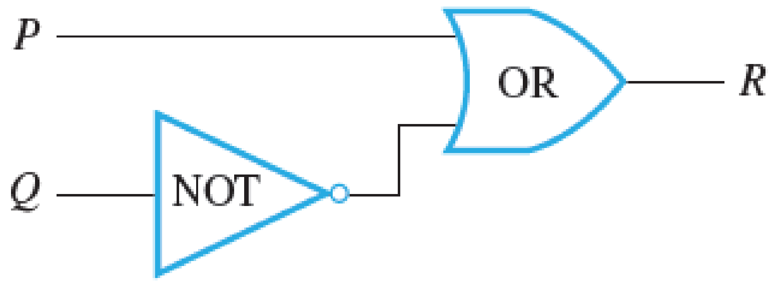


Señales de entrada: $P = 1$, $Q = 0$ y $Q = 1$

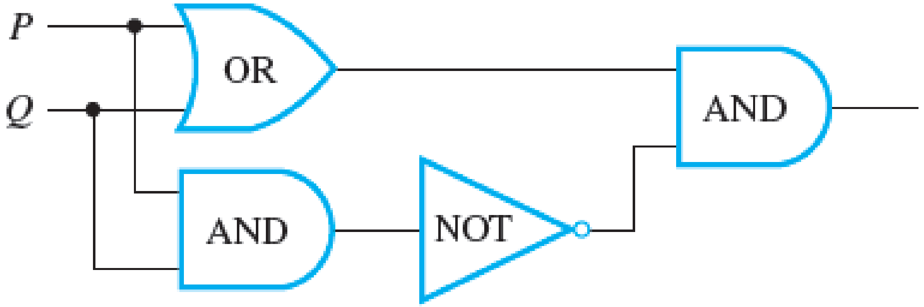
Circuitos Lógicos

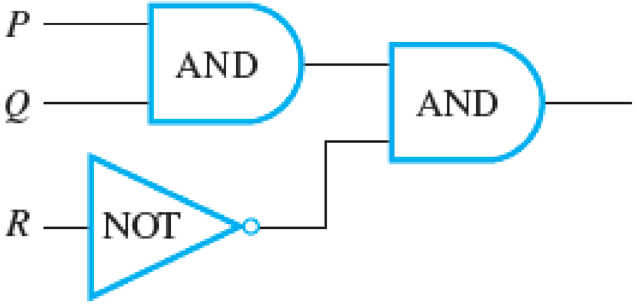
Ejemplos

2. Construya la **tabla de entrada/salida** para el siguiente circuito.



3. Encuentre las expresiones booleanas que corresponden a los circuitos que se muestran a continuación. Un punto indica una soldadura de dos alambres, cables que se cruzan sin un punto se supone que no se tocan.





Circuitos Lógicos

Ejemplos

4. Construya los circuitos correspondientes a cada una de las siguientes expresiones booleanas
   - $(\neg P \wedge Q) \vee \neg Q$
   - $\big((P \wedge Q) \wedge (R \wedge S)\big) \wedge T$
5. Diseñe un circuito para la siguiente **tabla de entrada/salida**:

Input			Output
P	Q	R	S
1	1	1	1
1	1	0	0
1	0	1	1
1	0	0	1
0	1	1	0
0	1	0	0
0	0	1	0
0	0	0	0

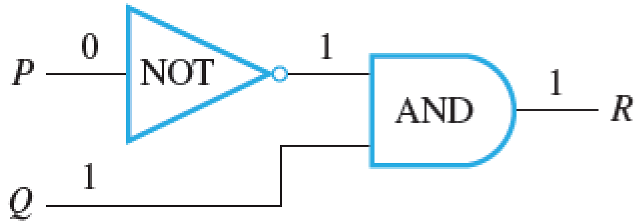
Circuitos Lógicos

Ejemplos - Solución

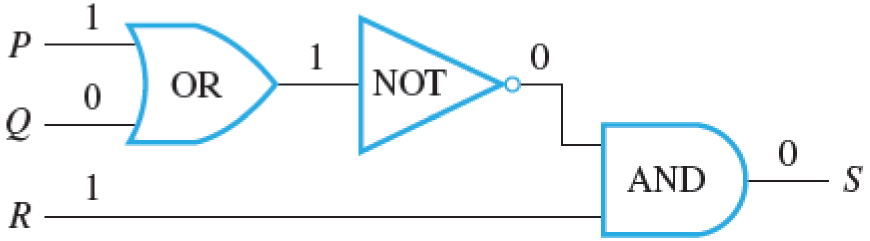
UNIVERSIDAD DE ANTIOQUIA

Ejemplo 1:

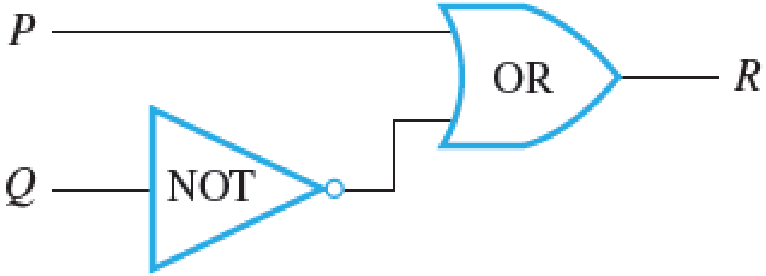
Señales de entrada: $P = 0$ y $Q = 1$



Señales de entrada: $P = 1$, $Q =$0 y $Q = 1$



Ejemplo 2:



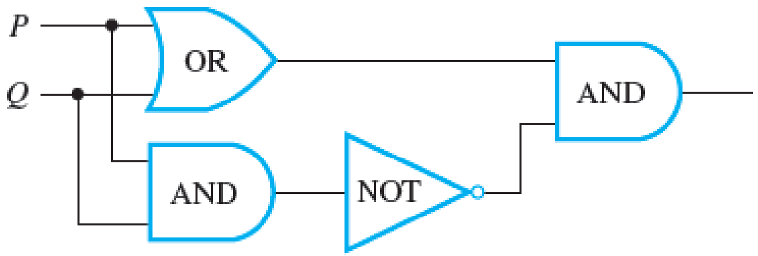
Liste las cuatro combinaciones posibles de las señales de entrada y encuentre la salida para cada una siguiendo el circuito

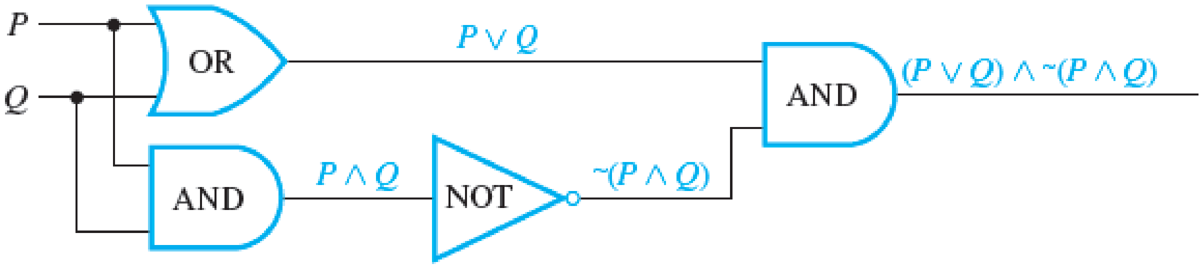
Entrada		Salida
P	**Q**	**R**
1	1	1
1	0	1
0	1	0
0	0	1

Circuitos Lógicos

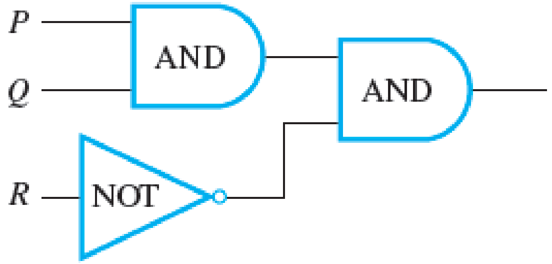
Ejemplos - Solución

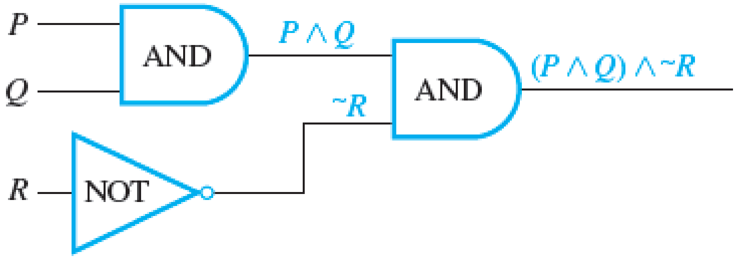
Ejemplo 3:





$$S = (P + Q) \cdot (P \cdot Q)'$$





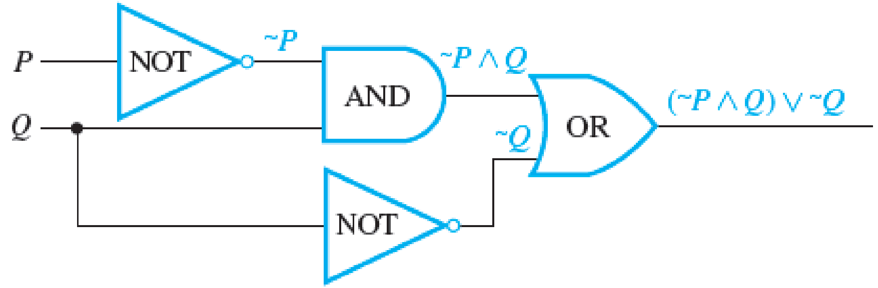
$$S = P \cdot Q \cdot R'$$

Circuitos Lógicos

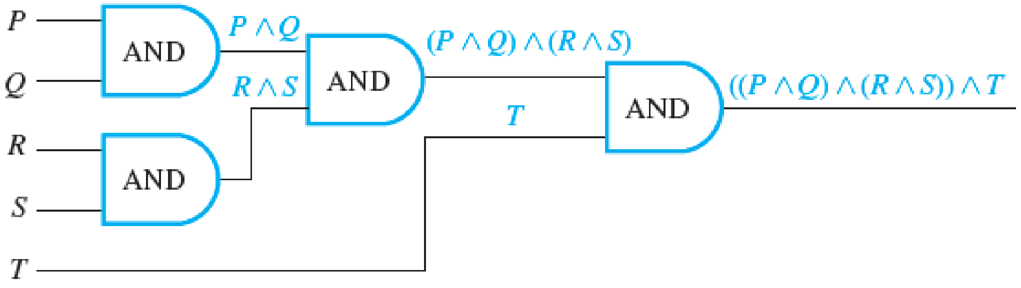
Ejemplos - Solución

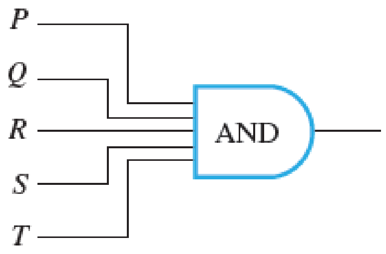
Ejemplo 4:

$(\neg P \wedge Q) \vee \neg Q$



$((P \wedge Q) \wedge (R \wedge S)) \wedge T$





Circuitos Lógicos

Ejemplos - Solución

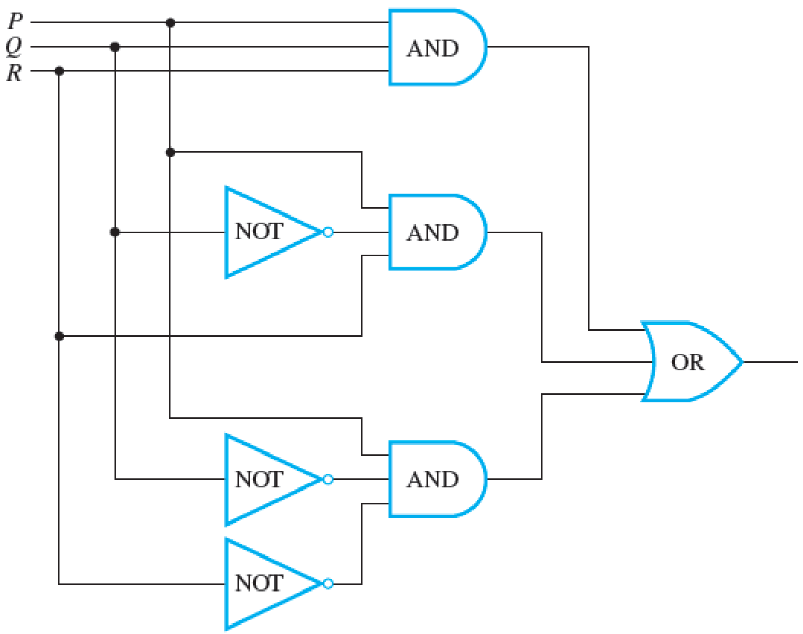
UNIVERSIDAD DE ANTIOQUIA

Ejemplo 5:

Tabla de Entrada/Salida

Input			Output
P	*Q*	*R*	*S*
1	1	1	1
1	1	0	0
1	0	1	1
1	0	0	1
0	1	1	0
0	1	0	0
0	0	1	0
0	0	0	0

Circuito lógico



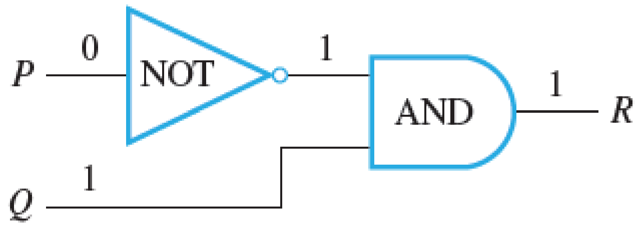
Expresión Booleana

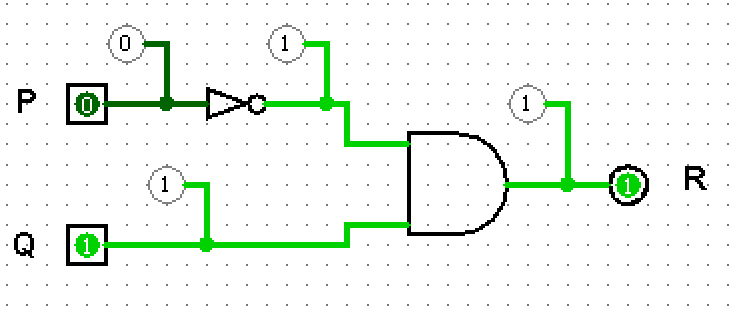
$$S = PQR + PQ'R + PQ'R'$$

Circuitos Lógicos

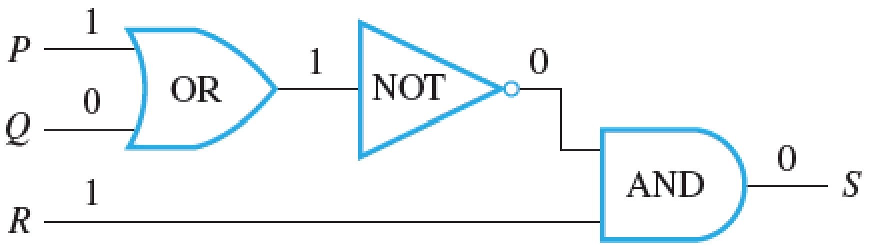
Ejemplos - Simulaciones

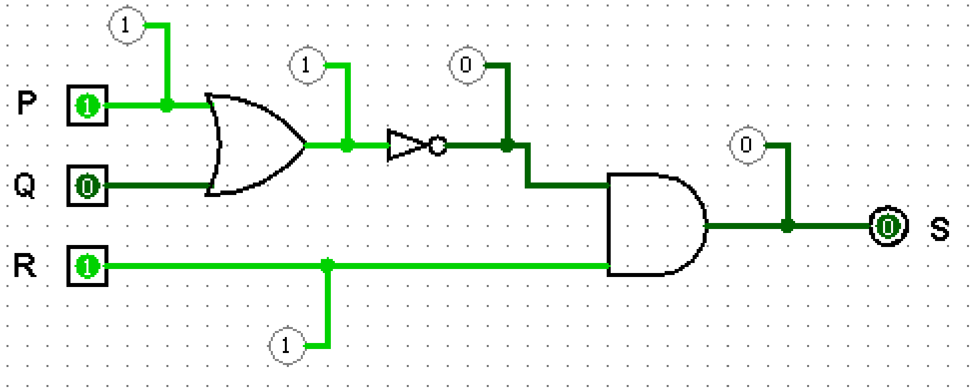
Ejemplo 1:





circuito_ejemplo1a.cir





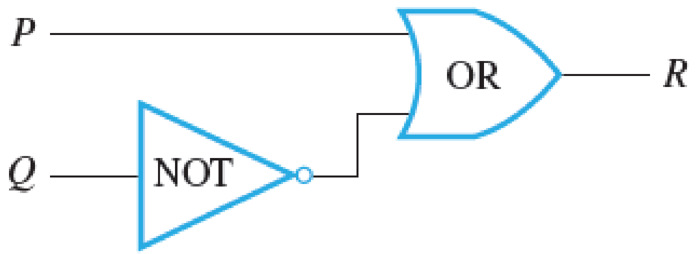
circuito_ejemplo1b.cir

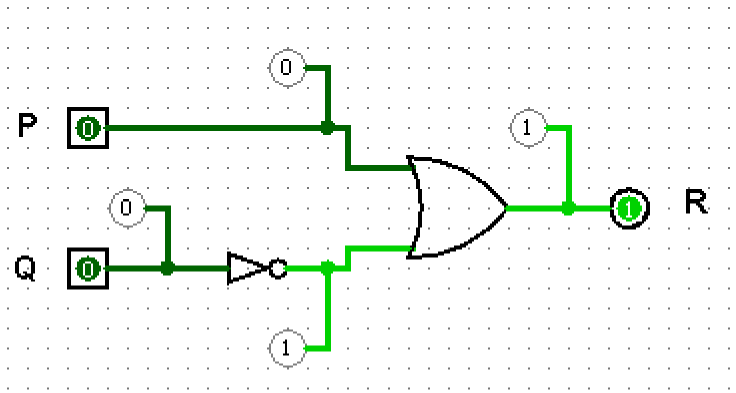
Circuitos Lógicos

Ejemplos - Simulaciones

Ejemplo 2:

Entrada		Salida
P	*Q*	*R*
1	1	1
1	0	1
0	1	0
0	0	1



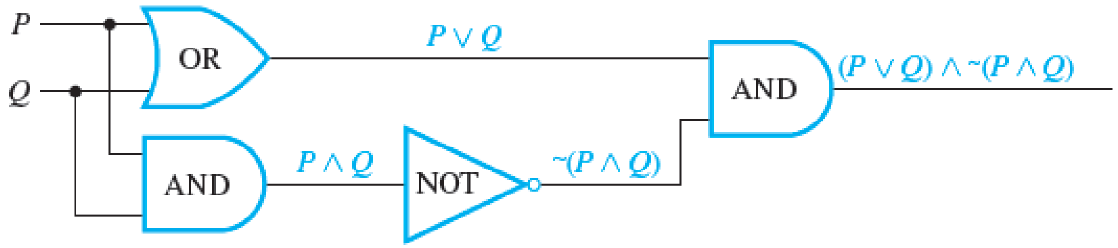


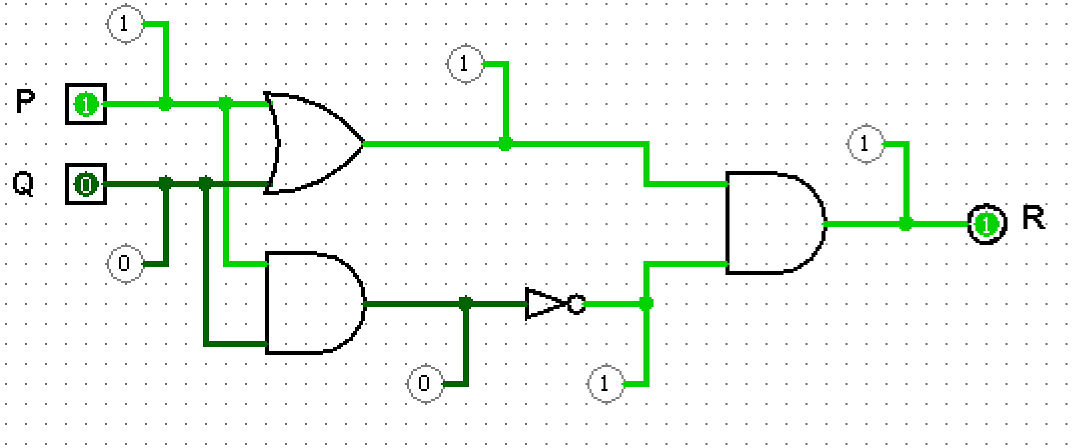
circuito_ejemplo2.cir

Circuitos Lógicos

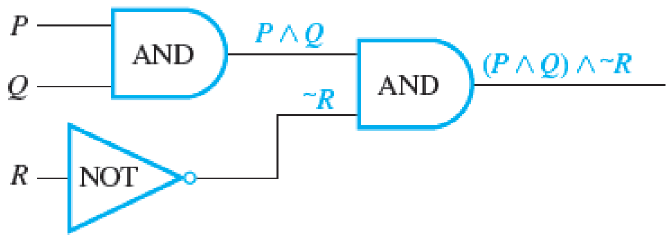
Ejemplos - Simulaciones

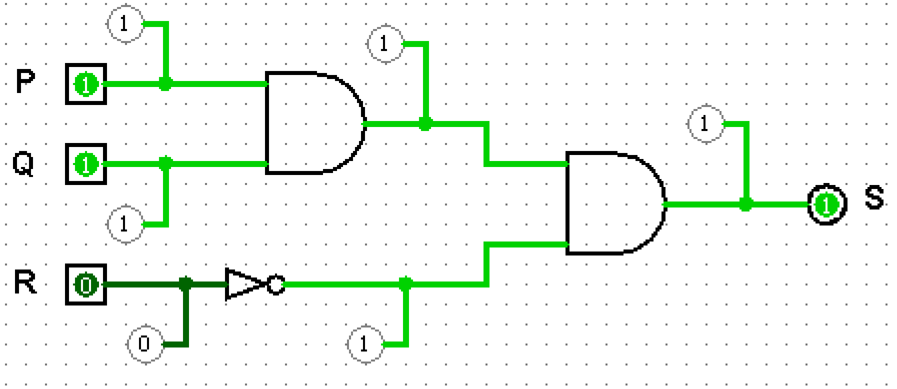
Ejemplo 3:





circuito_ejemplo3a.cir



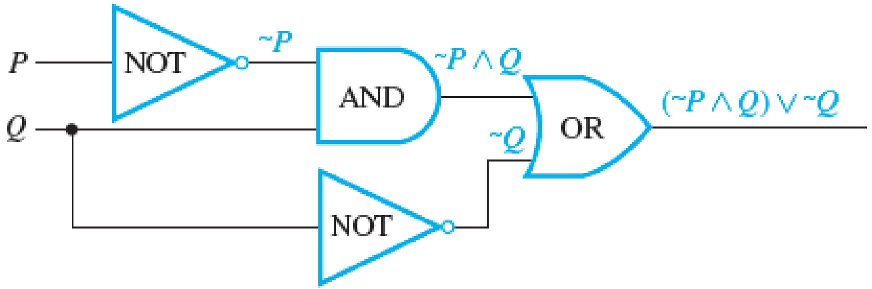


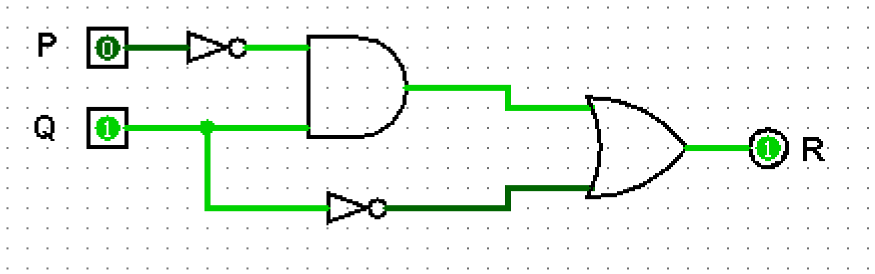
circuito_ejemplo3b.cir

Circuitos Lógicos

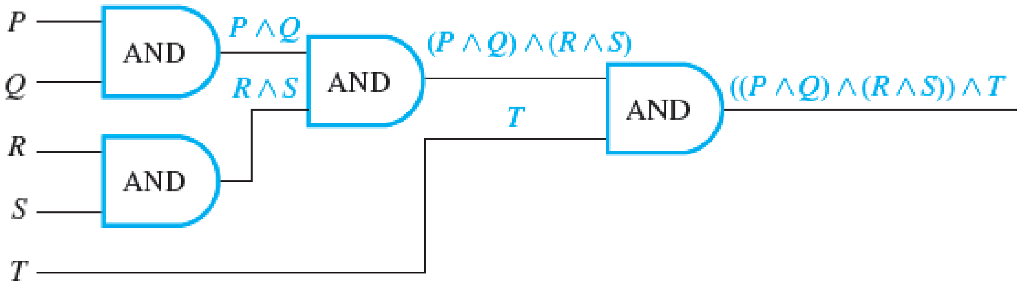
Ejemplos - Simulaciones

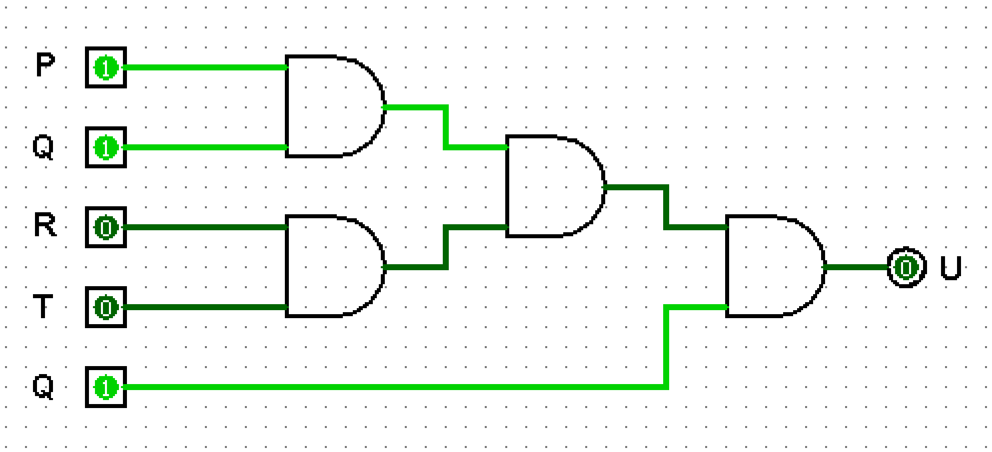
Ejemplo 4:





circuito_ejemplo4a.cir



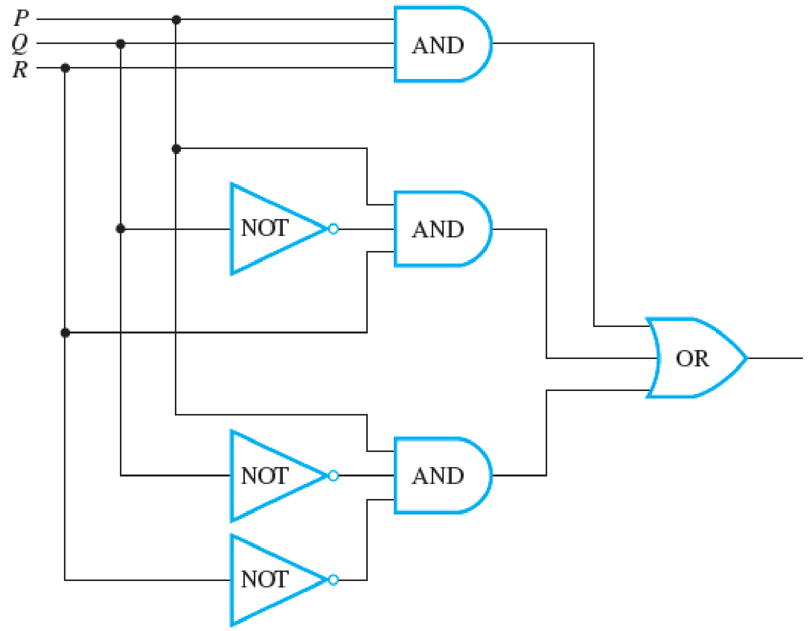


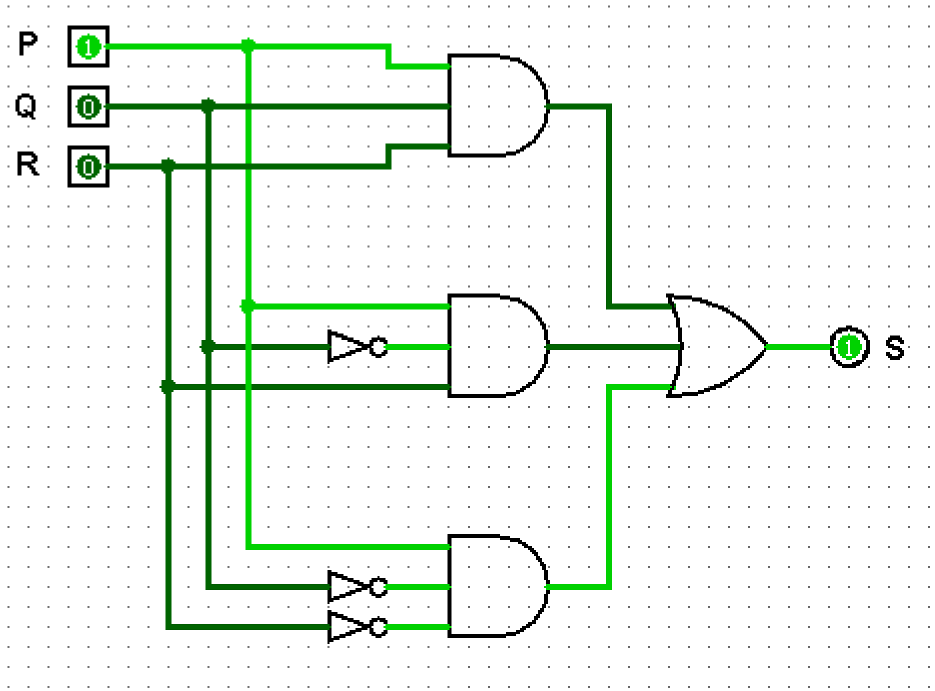
circuito_ejemplo4b.cir

Circuitos Lógicos

Ejemplos - Simulaciones

Ejemplo 5:





circuito_ejemplo5.cir

Agenda

UNIVERSIDAD DE ANTIOQUIA

- Contextualización
- Circuitos lógicos
- **Introducción al Algebra Booleana**
- Funciones Booleanas
- Representación de funciones Booleanas

Introducción al Algebra Booleana

Concepto clave

- El álgebra booleana utiliza el conjunto de valores $\{0,1\}$ junto con los operadores **suma booleana** $(+)$, **producto booleano** $(\cdot)$ y **complemento** $'$.
- **Operadores**:
  - **Suma Booleana (OR)**:

$$1+1=1; 1+0=1; 0+1=1; 0+0=0$$

  - **Producto Booleano (AND)**:

$$1\cdot 1=1; 1\cdot 0=0; 0\cdot 1=0; 0\cdot 0=0$$

  - **Complemento (NOT)**:

$$0'=1; 1'=0$$

Introducción al Algebra Booleana

UNIVERSIDAD DE ANTIOQUIA

Ejemplos

1. Encuentre el valor de $1 \cdot 0 + \overline{(0+1)}$
2. Traduzca la igualdad del ejemplo anterior: $1 \cdot 0 + \overline{(0+1)}$, a una equivalencia lógica.
3. Traduzca la equivalencia lógica $(V \wedge V) \vee \neg F \equiv V$ a una identidad en Algebra Booleana.

Introducción al Algebra Booleana

UNIVERSIDAD DE ANTIOQUIA

Ejemplo 1

Encuentre el valor de $1 \cdot 0 + \overline{(0 + 1)}$

Solución:

$$\begin{aligned} 1 \cdot 0 + (0 + 1)' &= 0 + 1' \\ &= 0 + 0 \\ &= 0 \end{aligned}$$

Introducción al Algebra Booleana

UNIVERSIDAD DE ANTIOQUIA

Ejemplo 2

Traduzca la igualdad del ejemplo anterior: $1 \cdot 0 + \overline{(0+1)}$, a una equivalencia lógica.

Solución:

Teniendo en cuenta que $0 = F, 1 = V$ y que el producto Booleano $(\cdot)$ es AND $(\wedge)$, la suma Booleana $(+)$ es OR $(\vee)$ y el complemento $(')$ es NOT $(\neg)$ tenemos que $1 \cdot 0 + \overline{(0+1)}$ queda como:

$$(V \wedge F) \vee \neg(F \vee V) \equiv F \vee \neg(V) \equiv F \vee F \equiv F$$

Introducción al Algebra Booleana

UNIVERSIDAD DE ANTIOQUIA

Ejemplo 3

Traduzca la equivalencia lógica $(V \wedge V) \vee \neg F \equiv V$ a una identidad en Algebra Booleana.

Solución:

Teniendo en cuenta que $F = 0, V = 1$ y que el producto Booleano $(\cdot)$ es AND $(\wedge)$, la suma Booleana $(+)$ es OR $(\vee)$ y el complemento $(')$ es NOT $(\neg)$ tenemos que $(V \wedge V) \vee \neg F \equiv V$ queda como:

$$(1 \cdot 1) + 0' = 1$$

Agenda

- Contextualización
- Circuitos lógicos
- Introducción al Algebra Booleana
- **Funciones Booleanas**
- Representación de funciones Booleanas

Funciones Booleanas

Definiciones fundamentales

1. **Conjunto booleano (B)**
   - Es el conjunto formado únicamente por los elementos 0 y 1.
   - $B = \{0,1\}$
2. **Conjunto de n-tuplas (B^n)**
   - Es el conjunto de todas las posibles secuencias (o "tuplas") de tamaño n que se pueden formar con 0 y 1.
   - Formalmente: $B^n = \{(x_1, x_2, \dots, x_n) | x_i \in B \ para \ 1 \leq i \leq n\}$
3. **Variable Booleana**
   - Variable que solo puede tomar dos posibles valores $0s$ y $1s$
4. **Función Booleana**
   - Una función que va desde el conjunto B^n al conjunto B se denomina **función booleana de grado n**.
   - En otras palabras, es una regla que toma n valores de entrada ($0s$ y $1s$) y produce una única salida (un 0 o un 1).

Funciones Booleanas

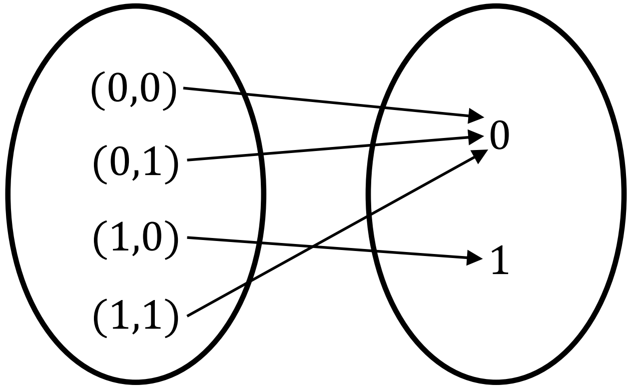
Ejemplo 1

Consideremos la función $F(x, y) = xy'$. ¿Es una función booleana?

Análisis:

- **Entradas**: La función recibe un par ordenado de variables booleanas, (x, y). Esto significa que las entradas provienen del conjunto $B^2 = \{(0,0), (0,1), (1,0), (1,1)\}$
- **Salida**: La función $F(x, y) = xy'$ se lee como: "$\boldsymbol{F}$ es Verdadera (1) si y solo si x es Verdadera $(\mathbf{1})$ **Y** y es Falsa $(\mathbf{0})$".
- **Mapeo**: La función mapea de B^2 a B lo cual se puede visualizar a continuación:

$\boldsymbol{x}$	$\boldsymbol{y}$	$\boldsymbol{F(x, y) = xy'}$
0	0	0
0	1	0
1	0	1
1	1	0



Funciones Booleanas

UNIVERSIDAD DE ANTIOQUIA

Definiciones fundamentales

Podemos definir formalmente las expresiones booleanas en las variables $x_1, x_2, \ldots, x_n$ de manera recursiva. Esto significa que definimos los bloques más básicos primero y luego damos las reglas para combinarlos.

1. **Caso Base (Los bloques fundamentales)**: Los siguientes elementos son, por sí mismos, expresiones booleanas válidas:
   - **Falso**: 0
   - **Verdadero**: 1
   - **Variables booleanas**: $x_1, x_2, \ldots, x_n$
2. **Paso Recursivo (Las reglas de construcción)**: Si E_1 y E_2 son dos expresiones booleanas ya existentes, entonces las siguientes combinaciones también son expresiones booleanas válidas:
   - **La negación de una expresión**: E_1'
   - **El producto o AND de dos expresiones**: $E_1 \cdot E_2$
   - **La suma u OR de dos expresiones**: $E_1 + E_2$

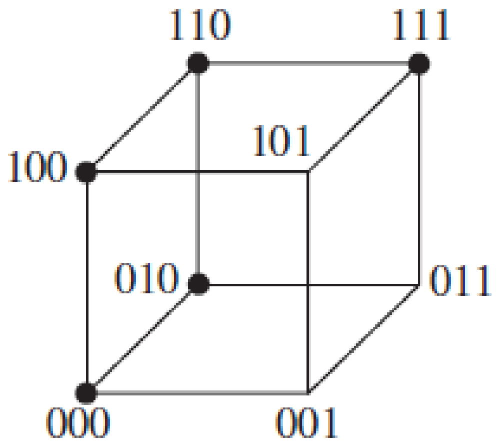
Funciones Booleanas

Ejemplo 2

Encuentre los valores de la función booleana representada por $F(x, y, z) = xy + z'$.

Solución: Los valores de esta función se muestran en la siguiente tabla:

x	y	z	xy	z'	$F(x, y, z) = xy + z'$
0	0	0	0	1	1
0	0	1	0	0	0
0	1	0	0	1	1
0	1	1	0	0	0
1	0	0	0	1	1
1	0	1	0	0	0
1	1	0	1	1	1
1	1	1	1	0	1



La función booleana pude ser representada gráficamente una como un **n-cubo** distinguiendo los vértices del **n-cubo** que corresponden a las n-tuplas de bits donde la función tiene valor **1**.

Funciones Booleanas

Igualdad y operaciones con funciones booleanas

Igualdad de funciones booleanas

Dos funciones booleanas F y G (de n variables) son iguales si y solo si producen el mismo resultado para cada combinación de entrada.

$$F(x_1, x_2, \dots, x_n) = G(x_1, x_2, \dots, x_n)$$

para toda combinación de valores x_i en $B = \{0,1\}$

Expresiones booleanas equivalentes

Dos expresiones booleanas distintas que representan la **misma función** se llaman **equivalentes**.

- **Ejemplo**: Las siguientes tres expresiones son equivalentes porque siempre producen el mismo resultado (tabla de verdad):
  - xy
  - $xy + 0$
  - $xy \cdot 1$

Funciones Booleanas

UNIVERSIDAD DE ANTIOQUIA

Igualdad y operaciones con funciones booleanas

Complemento de una Función

El complemento de una función F es una nueva función, denotada como F', que devuelve el valor opuesto a F para las mismas entradas.

$$F(x_1, x_2, \dots, x_n)' = \neg\big(F(x_1, x_2, \dots, x_n)\big)$$

Suma booleana

La suma de dos funciones F y G se define como la operación **OR** de sus salidas individuales.

$$(F + G)(x_1, x_2, \dots, x_n) = F(x_1, x_2, \dots, x_n) + G(x_1, x_2, \dots, x_n)$$

Producto booleano

La suma de dos funciones F y G se define como la operación **AND** de sus salidas individuales.

$$(F \cdot G)(x_1, x_2, \dots, x_n) = F(x_1, x_2, \dots, x_n) \cdot G(x_1, x_2, \dots, x_n)$$

Funciones Booleanas

UNIVERSIDAD DE ANTIOQUIA

Ejemplo 3

¿Qué tantas funciones booleanas de grado n se pueden formar?

Solución:

- Para una n-tupla $(x_1, x_2, \dots, x_n)$ booleana, cada variable x_i solo puede tomar dos posibles valores (0 y 1) de modo que el numero total de tuplas diferentes esta dado por.

$$\underbrace{2 \times 2 \times \cdots \times 2}_{n\ veces} = 2^n$$

- De estas dos posibles combinaciones, al tener una función booleana, cada una de estas puede también tomar dos posibles valores a la salida, de modo tenemos el siguiente numero de posibles funciones:

$$\underbrace{2 \times 2 \times \cdots \times 2}_{2^n\ veces} = 2^{2^n}$$

The Number of Boolean Functions of Degree n.

Degree	Number
1	4
2	16
3	256
4	65,536
5	4,294,967,296
6	18,446,744,073,709,551,616

Funciones Booleanas

Ejemplo 5

Una función booleana de grado dos es una función de un conjunto de cuatro elementos, es decir, pares de elementos desde $B = \{0,1\}$ hacia B, un conjunto de dos elementos. Por lo tanto, existen 16 funciones booleanas de grado dos diferentes. En la siguiente tabla se muestran los valores de las 16 funciones booleanas de grado dos diferentes, denominadas $F_1, F_2, \dots, F_{16}$.

The 16 Boolean Functions of Degree Two.

x	y	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}	F_{16}
1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
1	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
0	1	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0

Funciones Booleanas

Identidades del Algebra Booleana

La siguiente tabla muestra las identidades de algebra booleana mas importantes:

Identidades Booleanas		
Nombre	**Identidad**	
1. Ley del doble negación	$x'' = x$	
2. Ley de idempotencia	$x \cdot x = x$	$x + x = x$
3. Ley de identidad	$x \cdot 1 = x$	$x + 0 = x$
4. Ley de dominación	$x \cdot 0 = 0$	$x + 1 = 1$
5. Leyes conmutativa	$x \cdot y = y \cdot x$	$x + y = y + x$
6. Ley asociativa	$x \cdot (y \cdot z) = (x \cdot y) \cdot z$	$x + (y + z) = (x + y) + z$
7. Ley distributiva	$x \cdot (y + z) = x \cdot y + x \cdot z$	$x + y \cdot z = (x + y) \cdot (x + z)$
8. Leyes de De Morgan	$(x \cdot y)' = x' + y'$	$(x + y)' = x' \cdot y'$
9. Ley de absorción	$x \cdot (x + y) = x$	$x + x \cdot y = x$
10. Ley del complemento	$x \cdot x' = 0$	$x + x' = 1$

Funciones Booleanas

UNIVERSIDAD DE ANTIOQUIA

Principio de dualidad

- El **principio de dualidad** es una propiedad fundamental del álgebra booleana que establece que sus identidades siempre vienen en pares de modo que si se tiene una identidad booleana verdadera, es posible derivar su "identidad gemela" (su dual) de forma automática, y esta también será verdadera.
- Para obtener el dual de una expresión booleana se realizan los intercambios dados a continuación:
  - La operación suma $(+)$ por la multiplicación $(\cdot)$
  - El elemento neutro $\mathbf{0}$ por el $\mathbf{1}$
- **Ejemplo**:

Expresión original

Expresión dual

$$\begin{matrix} + & \mathbf{0} \\ \downarrow & \downarrow \\ \cdot & \mathbf{1} \end{matrix}$$

$$x + 0 = x \xrightarrow{\text{Transformación}} x \cdot 1 = x$$

Expresión original

Expresión dual

$$\begin{matrix} + & \cdot \\ \downarrow & \downarrow \\ \cdot & + \end{matrix}$$

$$x \cdot (y + z) = x \cdot y + x \cdot z \xrightarrow{\text{Transformación}} x + (y \cdot z) = (x + y) \cdot (x + z)$$

Funciones Booleanas

UNIVERSIDAD DE ANTIOQUIA

Ejemplo 6

Demuestre la ley de Idempotencia para el **producto**: $x \cdot x = x$ y para la **suma**: $x + x = x$

$$x + x = x$$

		Pasos	Razón
1	$x =$	$x + 0$	**Premisa** e Identidad para la suma $(+)$
2	$=$	$x + x \cdot x'$	Complemento para el producto $(\cdot)$ en **1**
3	$=$	$(x + x) \cdot (x + x')$	Distributividad para la suma $(+)$ en **2**
4	$=$	$(x + x) \cdot \mathbf{1}$	Complemento para la suma $(+)$ en **3**
5	$=$	$x + x$	Identidad para el producto $(\cdot)$ en **4**

$$x \cdot x = x$$

		Pasos	Razón
1	$x =$	$x \cdot 1$	**Premisa** e Identidad el producto $(\cdot)$
2	$=$	$x \cdot (x + x')$	Complemento para la suma $(+)$ en **1**
3	$=$	$x \cdot x + x \cdot x'$	Distributividad para el producto $(\cdot)$ en **2**
4	$=$	$x \cdot x + \mathbf{0}$	Complemento para el producto $(\cdot)$ en **3**
5	$=$	$x \cdot x$	Identidad para la suma $(+)$ en **4**

Funciones Booleanas

UNIVERSIDAD DE ANTIOQUIA

Ejemplo 7

Demuestre la ley la absorción $x \cdot (x+y) = x$ empleando las identidades del Algebra Booleana

$$x \cdot (x+y) = x$$

		Pasos		Razón
1	$x \cdot (x+y)$	$=$	$(x+\mathbf{0}) \cdot (x+y)$	**Premisa** e Identidad para la suma $(+)$
2		$=$	$(x+\mathbf{0}) \cdot x + (x+\mathbf{0}) \cdot y$	Distributividad para el producto $(\cdot)$ en **1**
3		$=$	$x \cdot x + (x+\mathbf{0}) \cdot y$	Identidad para la suma $(+)$ en **2**
4		$=$	$x + (x+\mathbf{0}) \cdot y$	Idempotencia para el producto $(\cdot)$ en **3**
5		$=$	$x + x \cdot y$	Identidad para la suma $(+)$ en **4**
6		$=$	$x \cdot (\mathbf{1}+y)$	Distributividad para el producto $(\cdot)$ en **5** (Factor común)
7		$=$	$x \cdot \mathbf{1}$	Dominación para la suma $(+)$ en **6**
8		$=$	x	Identidad para el producto $(\cdot)$ en **7**

Funciones Booleanas

UNIVERSIDAD DE ANTIOQUIA

Ejemplo 8

Demuestre que la identidad $x + x'y = x + y$ empleando las identidades del Algebra Booleana y la tabla de verificación (tabla de verdad)

Solución: Inicialmente se realiza la demostración empleando las identidades de Algebra Booleana.

	Pasos			Razón
1	$x + x' \cdot y$	$=$	$(x + x') \cdot (x + y)$	**Premisa** y distributividad para la suma $(+)$
2		$=$	$\mathbf{1} \cdot (x + y)$	Complemento para la suma $(+)$ en **1**
3		$=$	$x + y$	Identidad para el producto $(\cdot)$ en **2**

Ahora se procede a emplear la tabla de verificación para demostrar que la identidad es valida.

x	y	x'	$x'y$	$x + x'y$	$x + y$
0	0	1	0	0	0
0	1	1	1	1	1
1	0	0	0	1	1
1	1	0	0	1	1

Funciones Booleanas

UNIVERSIDAD DE ANTIOQUIA

Ejemplo 9

Simplifique la expresión $G(x, y, z) = x'yz + xy'z' + xy'z + xyz' + xyz$ empleando las identidades del Algebra Booleana

Solución:

	Pasos			Razón
$\mathbf{1}$	$G(x, y, z)$	$=$	$x'yz + xy'z' + xy'z + xyz' + xyz$	**Premisa**
$\mathbf{2}$		$=$	$x'yz + (xy'z' + xy'z) + (xyz' + xyz)$	Asociatividad para la suma $(+)$ en **1**
$\mathbf{3}$		$=$	$x'yz + xy'(z' + z) + xy(z' + z)$	Distributividad para el producto $(\cdot)$ en **2** (Factor común)
$\mathbf{4}$		$=$	$x'yz + xy'\mathbf{1} + xy\mathbf{1}$	Complemento para la suma $(+)$ en **3**
$\mathbf{5}$		$=$	$x'yz + xy' + xy$	Identidad para el producto $(\cdot)$ en **1**
$\mathbf{6}$		$=$	$x'yz + x(y' + y)$	Distributividad para el producto $(\cdot)$ en **5** (Factor común)
$\mathbf{7}$		$=$	$x'yz + x\mathbf{1}$	Complemento para la suma $(+)$ en **6**
$\mathbf{8}$		$=$	$x'yz + x$	Identidad para el producto $(\cdot)$ en **7**
$\mathbf{9}$		$=$	$(x' + x)(yz + x)$	Distributividad para la suma $(+)$ en **8**
$\mathbf{10}$		$=$	$\mathbf{1}(yz + x)$	Complemento para la suma $(+)$ en **9**
$\mathbf{11}$		$=$	$yz + x$	Identidad para el producto $(\cdot)$ en **10**
$\mathbf{12}$		$=$	$x + yz$	Conmutatividad para la suma $(+)$ en **11**

Funciones Booleanas

Ejemplo 9

Empleando tabla de verificación, demuestre que $G(x,y,z) = x'yz + xy'z' + xy'z + xyz' + xyz$ es igual a $F(x,y,z) = x + yz$

Solución:

x	y	z	x'	y'	z'	yz	$x'yz$	$xy'z'$	$xy'z$	xyz'	xyz	$G(x,y,z)$	$F(x,y,z)$
0	0	0	1	1	1	0	0	0	0	0	0	0	0
0	0	1	1	1	0	0	0	0	0	0	0	0	0
0	1	0	1	0	1	0	0	0	0	0	0	0	0
0	1	1	1	0	0	1	1	0	0	0	0	1	1
1	0	0	0	1	1	0	0	1	0	0	0	1	1
1	0	1	0	1	0	0	0	0	1	0	0	1	1
1	1	0	0	0	1	0	0	0	0	1	0	1	1
1	1	1	0	0	0	1	0	0	0	0	1	1	1

Funciones Booleanas

UNIVERSIDAD DE ANTIOQUIA

Definición formal de un Algebra Booleana

Un **Álgebra Booleana** es una estructura matemática que consiste en un conjunto B, dos operadores binarios (+ y ·), un operador unitario ($\bar{\ }$), y dos elementos distintos (0 y 1).

Se denota formalmente como la 6-tupla:

$$(\boldsymbol{B}, +, \cdot, \bar{\ }, \boldsymbol{0}, \boldsymbol{1})$$

Donde los siguientes axiomas (o postulados) se cumplen para todos los elementos $x, y, z \in B$:

Ley conmutativa

$$x + y = y + x$$
$$x \cdot y = y \cdot x$$

Ley asociativa

$$x + (y + z) = (x + y) + z$$
$$x \cdot (y \cdot z) = (x \cdot y) \cdot z$$

Ley distributiva

$$x + y \cdot z = (x + y) \cdot (x + z)$$
$$x \cdot (y + z) = x \cdot y + x \cdot z$$

Ley de identidad

$$x + \mathbf{0} = x$$
$$x \cdot \mathbf{1} = x$$

Ley de complemento

$$x + x' = \mathbf{1}$$
$$x \cdot x' = \mathbf{0}$$

Funciones Booleanas

UNIVERSIDAD DE ANTIOQUIA

Definición formal de un Algebra Booleana

- La Lógica Proposicional, la Teoría de Conjuntos y el Álgebra de Boole son sistemas matemáticos formalmente equivalentes (**isomórficos**). Esto significa que comparten la misma estructura subyacente.
- Lógica proposicional, teoría de conjuntos y álgebra binaria son representaciones distintas de álgebras booleanas pero se expresan en contextos distintos:
  - **Lógica proposicional**: operaciones sobre proposiciones (verdadero/falso).
  - **Teoría de conjuntos**: operaciones sobre conjuntos (pertenencia).
  - **Álgebra de Boole**: operaciones algebraicas con 0 y 1.
- Cuando se entiende las reglas en un sistema, es posible **traducirlas** a los otros.
- A continuación vamos a mostrar la conexión entre:
  - Símbolos y conceptos.
  - Identidades.

Funciones Booleanas

UNIVERSIDAD DE ANTIOQUIA

Definición formal de un Algebra Booleana

Concepto	Lógica proposicional	Teoría de conjuntos	Algebra de Boole
Elementos	Proposiciones (p, q)	Conjuntos (A, B)	Variables (x, y)
Verdadero / Todo	Tautología $(\mathbf{V})$	Conjunto universal $(\mathbf{U})$	$\mathbf{1}$
Falso / Nada	Contradicción $(\mathbf{F})$	Conjunto vacío $(\emptyset)$	$\mathbf{0}$
Operación AND	Conjunción $(\wedge)$	Intersección $(\cap)$	Producto $(\cdot)$
Operación OR	Disyunción $(\vee)$	Unión $(\cup)$	Suma $(+)$
Operación NOT	Negación $(\neg)$	Complemento (A^C)	Complemento (x')
Relación de Igualdad	Equivalencia $(\equiv)$	Igualdad $(=)$	Igualdad $(=)$

Funciones Booleanas

Definición formal de un Algebra Booleana

Identidad	Lógica proposicional	Teoría de conjuntos	Algebra de Boole
Elemento Neutro	$p \vee \mathbf{F} \equiv p$ $p \wedge \mathbf{V} \equiv p$	$A \cup \emptyset = A$ $A \cap \mathbf{U} = A$	$x + \mathbf{0} = x$ $x \cdot \mathbf{1} = x$
Dominación	$p \wedge \mathbf{F} \equiv \mathbf{F}$ $p \vee \mathbf{V} \equiv \mathbf{V}$	$A \cap \emptyset = \emptyset$ $A \cup \mathbf{U} = \mathbf{U}$	$x \cdot \mathbf{0} = \mathbf{0}$ $x + \mathbf{1} = \mathbf{1}$
Idempotencia	$p \vee p \equiv p$ $p \wedge p \equiv p$	$A \cup A = A$ $A \cap A = A$	$x + x = x$ $x \cdot x = x$
Complemento	$p \vee p' \equiv \mathbf{V}$ $p \wedge p' \equiv \mathbf{F}$	$A \cup A^C = \mathbf{U}$ $A \cap A^C = \emptyset$	$x + x' = \mathbf{1}$ $x \cdot x' = \mathbf{0}$
Conmutativa	$p \vee q \equiv q \vee p$ $p \wedge q \equiv q \wedge p$	$A \cup B = B \cup A$ $A \cap B = B \cap A$	$x + y = y + x$ $x \cdot y = y \cdot x$
Asociativa	$p \vee (q \vee r) \equiv (p \vee q) \vee r$ $p \wedge (q \wedge r) \equiv (p \wedge q) \wedge r$	$A \cup (B \cup C) = (A \cup B) \cup C$ $A \cap (B \cap C) = (A \cap B) \cap C$	$x + (y + z) = (x + y) + z$ $x \cdot (y \cdot z) = (x \cdot y) \cdot z$
Distributiva	$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$ $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$	$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$ $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$	$x + y \cdot z = (x + y) \cdot (x + z)$ $x \cdot (y + z) = x \cdot y + x \cdot z$
De Morgan	$(p \vee q)' \equiv p' \wedge q'$ $(p \wedge q)' \equiv p' \vee q'$	$(A \cup B)^C = A^C \cap B^C$ $(A \cap B)^C = A^C \cup B^C$	$(x + y)' = x' \cdot y'$ $(x \cdot y)' = x' + y'$
Doble Negación	$\neg(\neg p) \equiv p$	$(A^C)^C = A$	$x'' = x$
Absorción	$p \vee (p \wedge r) \equiv p$ $p \wedge (p \vee r) \equiv p$	$A \cup (A \cap B) = A$ $A \cap (A \cup B) = A$	$x + x \cdot y = x$ $x \cdot (x + y) = x$

UNIVERSIDAD DE ANTIOQUIA

Agenda

- Contextualización
- Circuitos lógicos
- Introducción al Algebra Booleana
- Funciones Booleanas
- **Representación de funciones Booleanas**

Representación de funciones booleanas

Representación de funciones booleanas

- Una **función booleana**, que es una regla que asigna una salida (0 o 1) a cada combinación de entradas, esto es:

$$f : \{0{,}1\}^n \rightarrow \{0{,}1\}$$

- Existen diferentes formas de representar una función booleana (algunas de las cuales ya hemos visto), estas son:
    1. **Expresiones Booleanas**: La forma algebraica.
    2. **Tablas de Verdad**: La representación explícita y exhaustiva.
    3. **Formas Canónicas (Suma de Productos y Producto de Sumas)**: Formas estandarizadas cruciales para el diseño y la simplificación.
    4. **Diagramas de Circuitos Lógicos**: La representación física y visual.
- En esta sección vamos a profundizar en la representación mediante **formas canónicas**.

Representación de funciones booleanas

UNIVERSIDAD DE ANTIOQUIA

Formas canónicas - ¿Por qué son necesarias?

- **Problema de la ambigüedad**: una misma función booleana puede ser representada por muchas expresiones algebraicas diferentes.
- **Ejemplo**: Represente mediante una tabla de verificación las funciones:
    - $F_1 = xy' + xy$
    - $F_2 = x(y' + y)$
    - $F_3 = x \cdot \mathbf{1}$
    - $F_4 = x$

x	y	x'	y'	xy	xy'	$y + y'$	F_1	F_2	F_3	F_4
0	0	1	1	0	0	1	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	1	0	1	1	1	1	1	1
1	1	0	0	1	0	1	1	1	1	1

- Todas estas expresiones representan la misma función ($F(x, y) = x$), pero no es obvio a simple vista lo cual dificulta saber si dos circuitos son equivalentes o encontrar la forma "base" de una función.
- **Solución – La estandarización**: Las **formas canónicas** son representaciones estándar y únicas para cada función booleana. "**Canónico**" significa que sigue una regla fija y aceptada.

Representación de funciones booleanas

UNIVERSIDAD DE ANTIOQUIA

Formas canónicas

- Las **formas canónicas** son expresiones algebraicas únicas y normalizadas de una función booleana. Las principales ventajas de estas son:
  - **Unicidad**: Cada función tiene exactamente una forma canónica de cada tipo. Si dos funciones tienen la misma forma canónica, son la misma función.
  - **Derivación Directa**: Se pueden obtener directamente de la tabla de verdad de la función sin necesidad de manipulación algebraica.
  - **Puente al Diseño**: Son el punto de partida para muchas técnicas de simplificación (como los Mapas de Karnaugh) y para el diseño directo de circuitos.
- Existen dos formas de representación canónica:
  - Forma Normal Disyuntiva (FND) o Suma de minterminos.
  - Forma Normal Conjuntiva (FNC) o producto de maxterminos.
- Para construir las formas canónicas, primero es necesario comprender los **bloques fundamentales** para su construcción, esto es, los **minterminos** y los **maxterminos**.

Representación de funciones booleanas

UNIVERSIDAD DE ANTIOQUIA

Minterminos (Minterms)

- **Término producto**: es una expresión lógica que consiste en un conjunto de variables (o sus complementos) unidas por productos lógicos (**AND**). Algunos ejemplos son: $xyz, x'y, xy'z'w$
- Un **mintérmino** es un término producto (**AND**) que incluye todas las variables de la función, ya sea en su forma normal o negada.
- **Notación**: m_i denota el mintermino, donde i es el valor decimal de la fila de la tabla de verdad.
- **Ejemplo**: La representación mediante minterminos para tres variables (x, y, z) se muestra en la siguiente tabla:

Fila (i)	x	y	z	m_i	**expresión**
0	0	0	0	m_0	$x'y'z'$
1	0	0	1	m_1	$x'y'z$
2	0	1	0	m_2	$x'yz'$
3	0	1	1	m_3	$x'yz$
4	1	0	0	m_4	$xy'z'$
5	1	0	1	m_5	$xy'z$
6	1	1	0	m_6	xyz'
7	1	1	1	m_7	xyz

Observación importante: Para los minterminos un 0 en la entrada indica que la variable esta negada, mientras que un 1 implica que la variable esta sin negar.

Representación de funciones booleanas

UNIVERSIDAD DE ANTIOQUIA

Maxterminos (Maxterms)

- **Término suma**: es una expresión lógica que consiste en un conjunto de variables (o sus complementos) unidas por las sumas lógicas (**OR**). Algunos ejemplos son: $x' + y, w + x + z'$
- Un **maxtermino** es un término suma (**OR**) que incluye todas las variables de la función, ya sea en su forma normal o complementada.
- **Notación**: M_i denota el maxtermino, donde i es el valor decimal de la fila de la tabla de verdad.
- **Ejemplo**: La representación mediante maxterminos para tres variables (x, y, z) se muestra en la siguiente tabla:

Fila (i)	x	y	z	M_i	**expresión**
0	0	0	0	M_0	$x + y + z$
1	0	0	1	M_1	$x + y + z'$
2	0	1	0	M_2	$x + y' + z$
3	0	1	1	M_3	$x + y' + z'$
4	1	0	0	M_4	$x' + y + z$
5	1	0	1	M_5	$x' + y + z'$
6	1	1	0	M_6	$x' + y' + z$
7	1	1	1	M_7	$x' + y' + z'$

Observación importante: Para los maxtermino un 0 en la entrada indica que la variable esta sin negar, mientras que un 1 implica que la variable esta negada.

Representación de funciones booleanas

UNIVERSIDAD DE ANTIOQUIA

Formas estándar .vs. No estándar

- No todas las expresiones booleanas se escriben de la misma manera.

- Forma estándar: Cuando cada término (producto o suma) contiene todas las variables.
  - Cada mintermino de la FND incluye todas las variables.
  - Cada maxtermino de la FNC incluye todas las variables.
  - Se construye directamente de la tabla de verdad.
  - **Ejemplos**:
    - $F_1 = ab'c + ab'c' + a'bc = \sum m(3,4,5)$
    - $F_2 = (a+b')(a'+b')(a+b) = \prod M(0,1,3)$
    - $F_3 = abcd + a'bc'd = \sum m(5,15)$
    - $F_4 = (a+b+c)(a'+b'+c') = \prod M(0,7)$

- Forma simplificada o no estándar: Cuando algunos términos no tienen todas las variables
  - Es la forma más libre de escribir una función.
  - Algunos términos ya no tienen todas las variables.
  - Resulta de simplificar la expresión.
  - **Ejemplos**:
    - $F_1 = (ab' + c)(b + d)$
    - $F_2 = a + b\big(c + (d + e)\big)$
    - $F_3 = (a + b)'c + d$
    - $F_4 = ab' + cd$
    - $F_5 = a + b'cd' + ac'd$
    - $F_6 = (a' + b')b$
    - $F_7 = a(b' + c + d')(a + c' + d)$

Representación de funciones booleanas

Forma Normal Disyuntiva (FND) – Suma de productos (SOP)

- Esta forma se construye sumando (**OR**) los minterminos que corresponden a las salidas 1 de la función.
- **Regla**: Si la función es 1 en la fila i, entonces el mintermino m_i forma parte de la suma.
- **Notación abreviada**:

$$F = \sum m(i_1, i_2, \dots)$$

Fila (i)	x	y	z	F
0	0	0	0	0
1	0	0	1	0
2	0	1	0	1
3	0	1	1	0
4	1	0	0	1
5	1	0	1	0
6	1	1	0	1
7	1	1	1	1

Ejemplo: Obtenga la función y su representación como suma de minterminos.

$$F(x,y,z) = x'yz' + xy'z' + xyz' + xyz$$

$$F(x,y,z) = m_2 + m_4 + m_6 + m_7$$

$$F(x,y,z) = \sum m(2,4,6,7)$$

Representación de funciones booleanas

UNIVERSIDAD DE ANTIOQUIA

Forma Normal Conjuntiva (FNC) – Producto de sumas (POS)

- Esta forma se construye multiplicando (**AND**) los maxterminos que corresponden a las salidas 0 de la función.
- **Regla**: Si la función es 0 en la fila i, entonces el mintermino M_i forma parte del producto.
- **Notación abreviada**:

$$F = \prod M(i_1, i_2, \ldots)$$

Fila (i)	x	y	z	F
0	0	0	0	0
1	0	0	1	0
2	0	1	0	1
3	0	1	1	0
4	1	0	0	1
5	1	0	1	0
6	1	1	0	1
7	1	1	1	1

Ejemplo: Obtenga la función y su representación como suma de maxterminos

$$F(x, y, z) = (x + y + z)(x + y + z')(x + y' + z')(x' + y + z')$$

$$F(x, y, z) = M_0 \cdot M_1 \cdot M_3 \cdot M_5$$

$$F(x, y, z) = \prod M(0,1,3,5)$$

Representación de funciones booleanas

Conversión entre formas canónicas

Para convertir una forma canónica en otra, se intercambian los símbolos $\sum$ y $\prod$, y se listan los índices faltantes

Fila $(\boldsymbol{i})$	$\boldsymbol{x}$	$\boldsymbol{y}$	$\boldsymbol{z}$	$\boldsymbol{F}$	$\boldsymbol{m_i}$	$\boldsymbol{M_i}$
0	0	0	0	0		M_0
1	0	0	1	0		M_1
2	0	1	0	1	m_2	
3	0	1	1	0		M_3
4	1	0	0	1	m_4	
5	1	0	1	0		M_5
6	1	1	0	1	m_6	
7	1	1	1	1	m_7	

$$F(x, y, z) = \sum m(2,4,6,7)$$

$$\downarrow \text{Conversión}$$

$$G(x, y, z) = \prod M(0,1,3,5)$$

Representación de funciones booleanas

UNIVERSIDAD DE ANTIOQUIA

Ejemplo 1

La siguiente tabla de verdad describe la función $F(a,b)$, se pide:

- Obtener la representación en las formas canónicas SOP y POS para cada caso.
- Simplique la expresión SOP si es posible.
- Obtener el circuito digital asociado.

$\boldsymbol{a}$	$\boldsymbol{b}$	$\boldsymbol{F(a,b)}$
0	0	0
0	1	1
1	0	1
1	1	1

Representación de funciones booleanas

UNIVERSIDAD DE ANTIOQUIA

Ejemplo 1 - Solución

Formas estándar

indice $(\boldsymbol{i})$	$\boldsymbol{a}$	$\boldsymbol{b}$	$\boldsymbol{F(a,b)}$	$\boldsymbol{m_i}$	$\boldsymbol{M_i}$
0	0	0	0	$a'b'$	$a+b$
1	0	1	1	$a'b$	$a+b'$
2	1	0	1	ab'	$a'+b$
3	1	1	1	ab	$a'+b'$

POS (Producto de sumas)

$$F(a,b) = (a+b) = M_0 = \prod M(0)$$

SOP (Suma de productos)

$$f(a,b) = a'b + ab' + ab = m_1 + m_2 + m_3 = \sum m(1,2,3)$$

Representación de funciones booleanas

UNIVERSIDAD DE ANTIOQUIA

Ejemplo 1 - Solución

Simplificación de la POS

indice (i)	a	b	$F(a,b)$	m_i
0	0	0	0	$a'b'$
1	0	1	1	$a'b$
2	1	0	1	ab'
3	1	1	1	ab

SOP (Suma de productos)

$$f(a,b) = a'b + ab' + ab = m_1 + m_2 + m_3 = \sum m(1,2,3)$$

Simplificación

		Pasos		Razón
1	$F(a,b)$	$=$	$a'b + ab' + ab$	SOP canónico (sin simplificar)
2		$=$	$a'b + a(b' + b)$	Distributividad para el producto $(\cdot)$ en **1** (factor común)
3		$=$	$a'b + a\mathbf{1}$	Complemento para la suma $(+)$ en **2**
4		$=$	$a'b + a$	Identidad para el producto $(\cdot)$ en **3**
5		$=$	$(a' + a)(b + a)$	Distributividad para la suma $(+)$ en **4**
6		$=$	$\mathbf{1}(b + a)$	Complemento para la suma $(+)$ en **5**
7		$=$	$b + a$	Identidad para el producto $(\cdot)$ en **6**
8		$=$	$a + b$	Conmutatividad para la suma $(+)$ en **7**

Expresión simplificada

$$f(a,b) = a + b$$

Representación de funciones booleanas

Ejemplo 1 - Solución

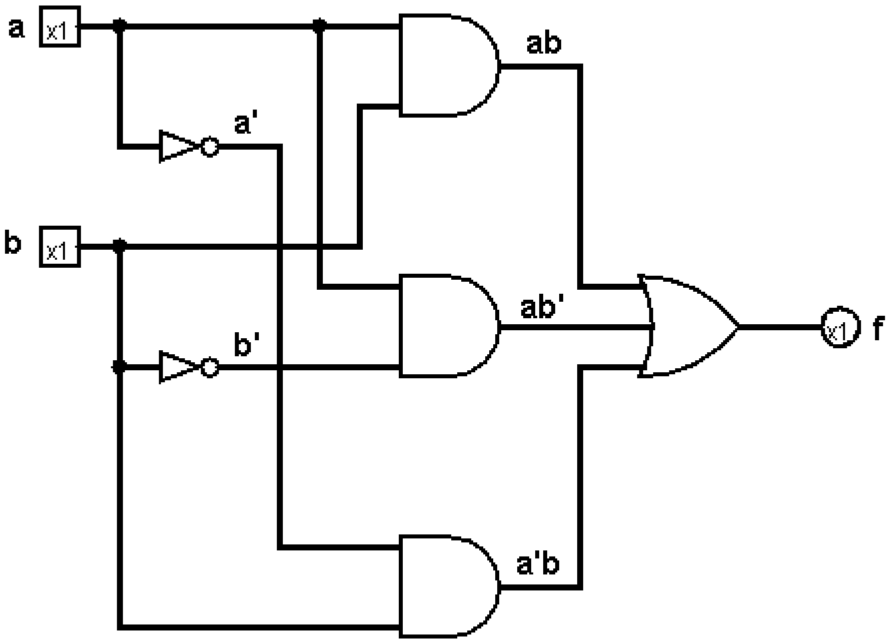
Circuitos digitales equivalentes

Tabla de verdad

a	b	$F(a,b)$
0	0	0
0	1	1
1	0	1
1	1	1

Forma estandar

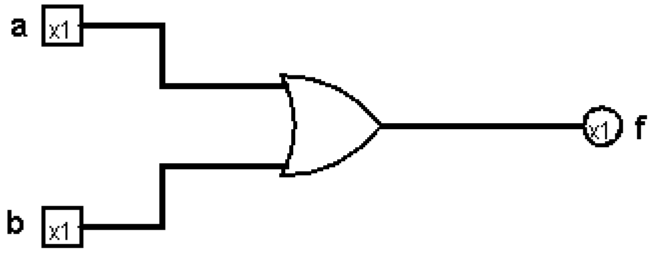
$$f(a,b) = a'b + ab' + ab$$



ejemplo1_f_no_simplificada.cir

Forma no estandar

$$f(a,b) = a + b$$



ejemplo1_f_simplificada.cir

Representación de funciones booleanas

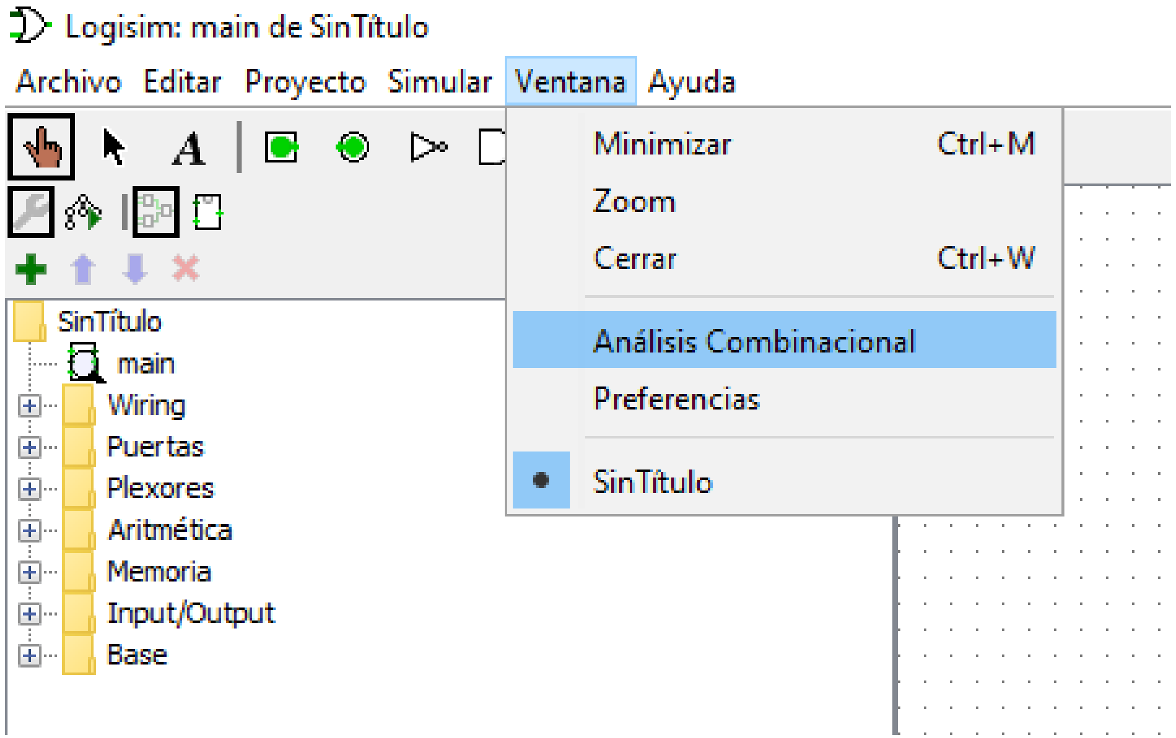
UNIVERSIDAD DE ANTIOQUIA

Ejemplo 1 – Simulación

A continuación vamos a mostrar como obtener el circuito a partir de la tabla de verdad del ejemplo1 en Logisim

1. **Seleccione la opción de Análisis combinacional**: Ventana > Análisis combinacional

a	b	$F(a,b)$
0	0	0
0	1	1
1	0	1
1	1	1



Representación de funciones booleanas

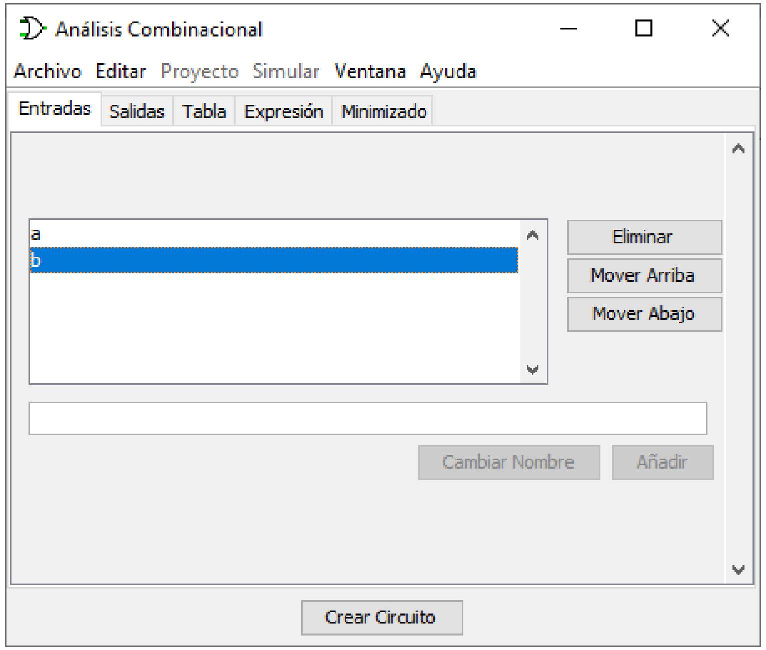
UNIVERSIDAD DE ANTIOQUIA

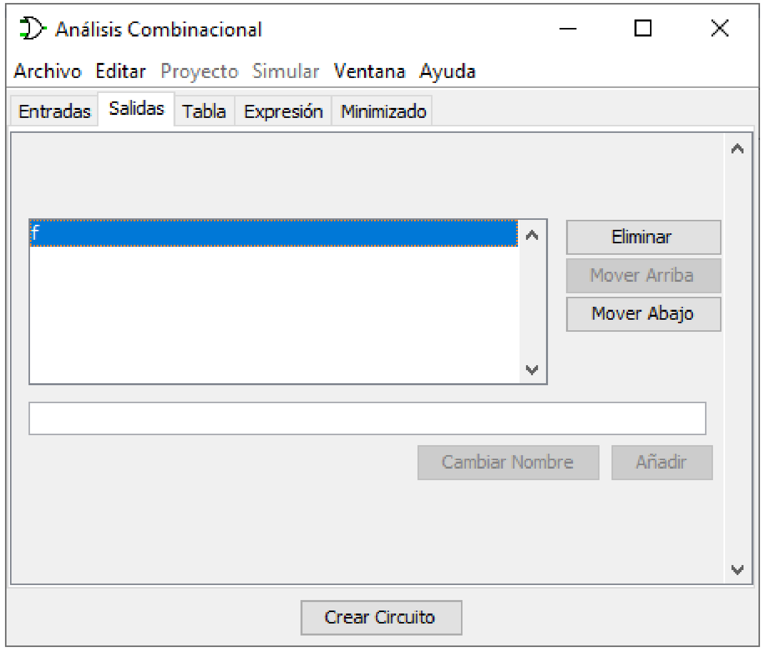
Ejemplo 1 – Simulación

A continuación vamos a mostrar como obtener el circuito a partir de la tabla de verdad del ejemplo1 en Logisim

2. **Defina las entradas y salidas**: En la pestaña de Entradas coloque agregue la entradas de la función y repita el mismo procedimiento para las salidas en la respectiva pestaña.

a	b	$F(a,b)$
0	0	0
0	1	1
1	0	1
1	1	1



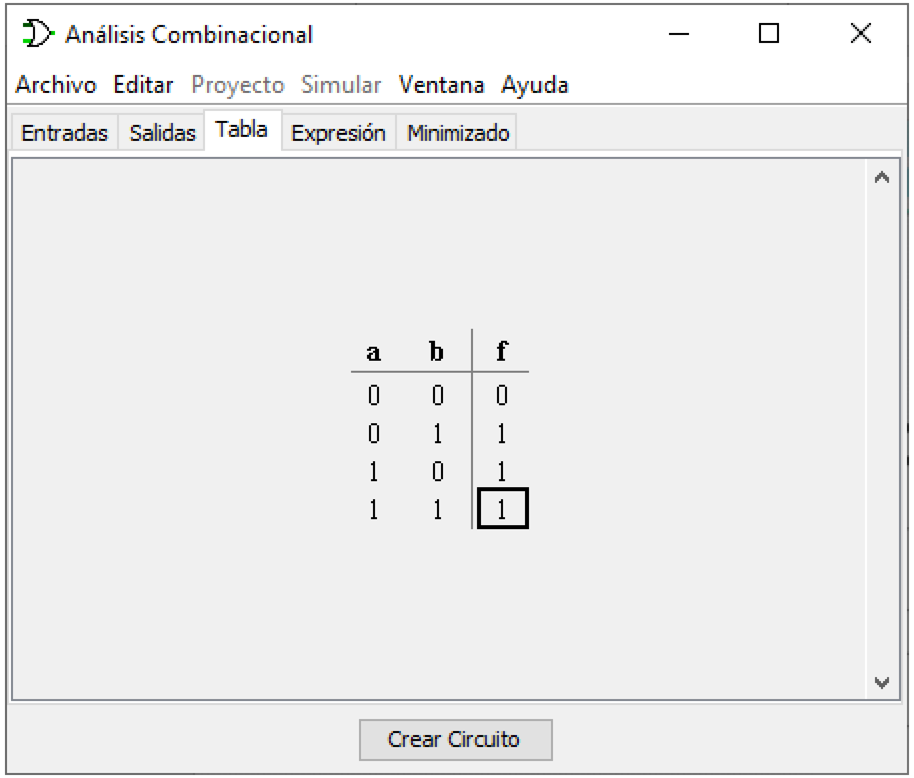


Representación de funciones booleanas

Ejemplo 1 - Solución

3. **Defina la función**: Ingrese los valores de las salidas para definir la función booleana.

a	b	$F(a,b)$
0	0	0
0	1	1
1	0	1
1	1	1

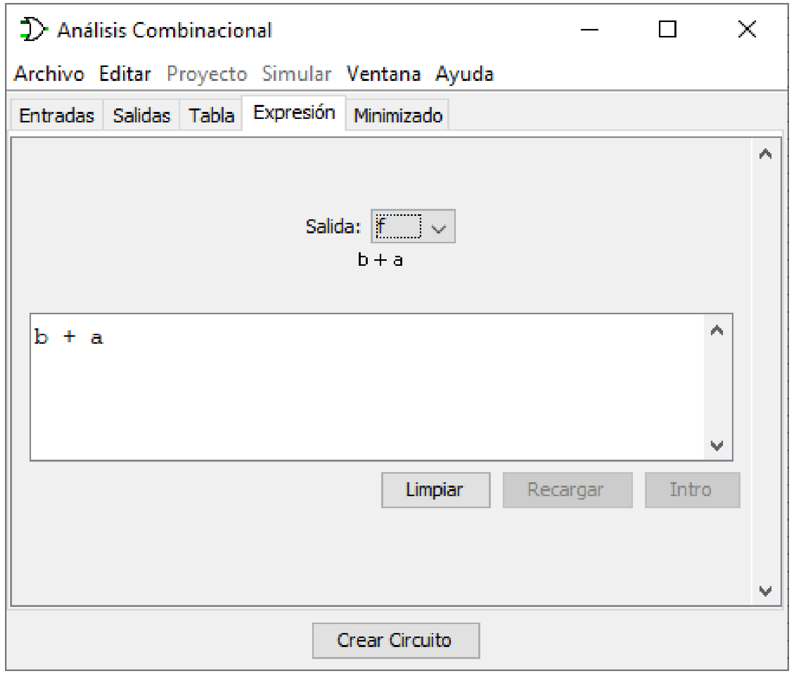


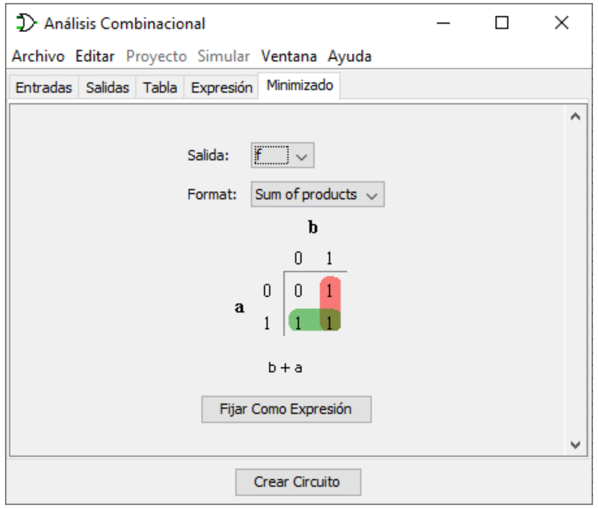
Representación de funciones booleanas

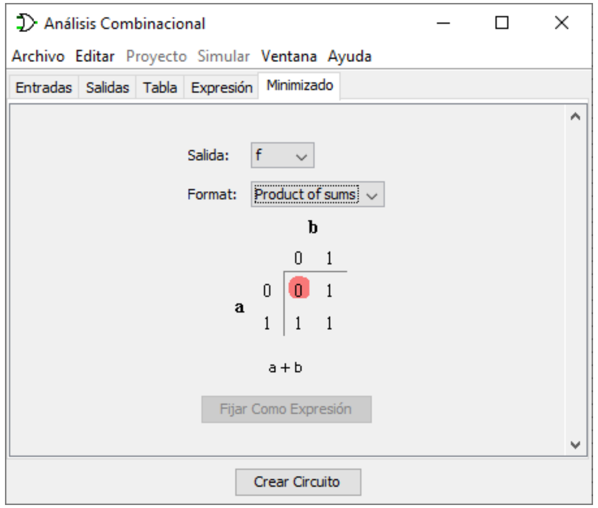
UNIVERSIDAD DE ANTIOQUIA

Ejemplo 1 - Solución

4. **Inspeccione la expresión asociada a la función booleana**: La expresión simplificada así como el mapa de K relacionado con la tabla puede ser visualizado mediante una inspección en las pestañas **Expresión** y **Minimizado** respectivamente.





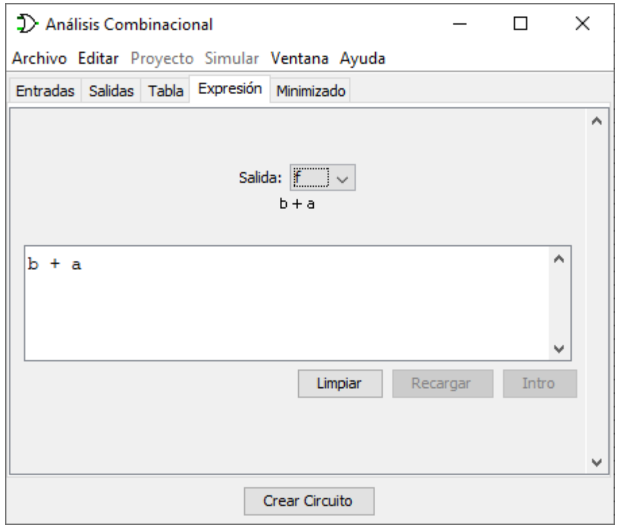


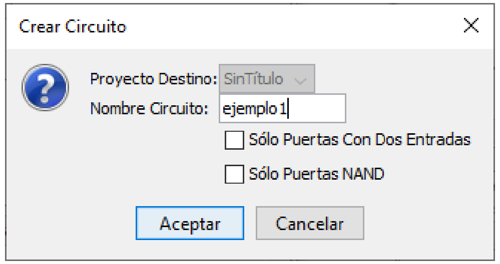
Representación de funciones booleanas

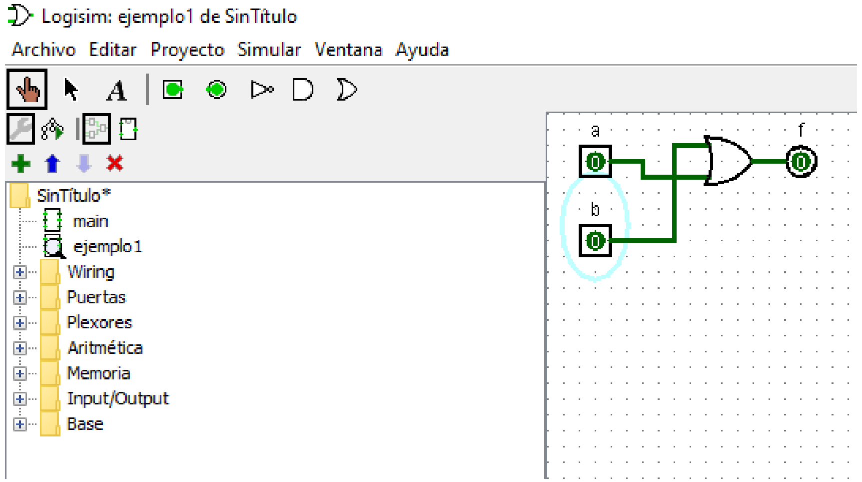
UNIVERSIDAD DE ANTIOQUIA

Ejemplo 1 - Solución

5. **Genere el circuito digital asociado**: Presione el botón **Crear Circuito**, luego coloque el nombre del subcircuito (caja negra) asociado a las compuertas lógicas conectadas. Finalmente, de click en el botón **Aceptar**. Si todo sale de acuerdo a lo esperado, se generara el circuito digital.







Representación de funciones booleanas

Ejemplo 2

Se requiere diseñar un circuito lógico que controle una luz de advertencia **A** en el tablero de un vehículo. Esta luz debe encenderse si ocurre alguna situación de peligro según las siguientes condiciones.

- El motor está encendido y la puerta del conductor está abierta.
- El motor está encendido, es de noche y las luces están apagadas.

Representación de funciones booleanas

Ejemplo 2 - Solución

Se requiere diseñar un circuito lógico que controle una luz de advertencia **A** en el tablero de un vehículo. Esta luz debe encenderse si ocurre alguna situación de peligro según las siguientes condiciones.

- El motor está encendido y la puerta del conductor está abierta.
- El motor está encendido, es de noche y las luces están apagadas.

Paso 1 - Definir las entradas y salidas del sistema:

Entradas:

- **M (motor):** M = 1 el motor esta encendido, M = 0 el motor esta apagado.
- **P (puerta)**: P = 1 la puerta esta abierta, P = 0 la puerta esta cerrada.
- **N (noche)**: N = 1 si es de noche, 0 si es de día.
- **L (luces)**: L = 1 las luces están encendidas, 0 las luces están apagadas.

Salida:

- **A (alarma):** A=1 la luz de advertencia esta encendida; A = 0, el caso contrario.

Representación de funciones booleanas

UNIVERSIDAD DE ANTIOQUIA

Ejemplo 2 - Solución

Paso 2 – Traducir las reglas a una expresión binaria

Para el caso se tiene que $A(M, P, N, L)$ es una función booleana que se activa $A = 1$ cuando:

- El motor está encendido y la puerta del conductor está abierta: $M = 1, P = 1$
- El motor está encendido, es de noche y las luces están apagadas: $M = 1, N = 1, L = 0$

De este modo podemos decir que la expresión inicial para A esta dada por:

$$A = M \cdot P + M \cdot N \cdot L'$$

Representación de funciones booleanas

UNIVERSIDAD DE ANTIOQUIA

Ejemplo 2 - Solución

Paso 3 – Tabla de verdad

Permite no solo determinar el funcionamiento del circuito sino que también sirve para verificar el comportamiento de este para todas las entradas y salidas posibles.

La alarma se actica cuando:

- El motor está encendido y la puerta del conductor está abierta
- El motor está encendido, es de noche y las luces están apagadas

indice (i)	M	P	N	L	$A(M,P,N,L)$	m_i
0	0	0	0	0	0	
1	0	0	0	1	0	
2	0	0	1	0	0	
3	0	0	1	1	0	
4	0	1	0	0	0	
5	0	1	0	1	0	
6	0	1	1	0	0	
7	0	1	1	1	0	
8	1	0	0	0	0	
9	1	0	0	1	0	
10	1	0	1	0	1	$MP'NL'$
11	1	0	1	1	0	
12	1	1	0	0	1	$MPN'L'$
13	1	1	0	1	1	$MPN'L$
14	1	1	1	0	1	$MPNL'$
15	1	1	1	1	1	$MPNL$

Representación de funciones booleanas

UNIVERSIDAD
DE ANTIOQUIA

Ejemplo 2 - Solución

Paso 3 – Tabla de verdad

indice (i)	M	P	N	L	$A(M,P,N,L)$	m_i
0	0	0	0	0	0	
1	0	0	0	1	0	
2	0	0	1	0	0	
3	0	0	1	1	0	
4	0	1	0	0	0	
5	0	1	0	1	0	
6	0	1	1	0	0	
7	0	1	1	1	0	
8	1	0	0	0	0	
9	1	0	0	1	0	
10	1	0	1	0	1	$MP'NL'$
11	1	0	1	1	0	
12	1	1	0	0	1	$MPN'L'$
13	1	1	0	1	1	$MPN'L$
14	1	1	1	0	1	$MPNL'$
15	1	1	1	1	1	$MPNL$

A partir de la tabla de verdad llegamos a la siguiente expresión:

$$A = MP'NL' + MPN'L' + MPN'L + MPNL' + MPNL$$

$$A = m_{10} + m_{12} + m_{13} + m_{14} + m_{15}$$

$$A = \sum m(10,12,13,14,15)$$

Representación de funciones booleanas

UNIVERSIDAD DE ANTIOQUIA

Ejemplo 2 - Solución

Paso 4 – Simplificación

Aplicando las identidades del algebra booleana o métodos como los mapas K procedemos a simplificar la expresión booleana

Simplificación de la expresión derivada del enunciado

	Pasos	Razón
1	$A = MP + MNL'$	Expresión sin simplificar
2	$= M(P + NL')$	Distributividad para el producto $(\cdot)$ en **1** (Factor común)

$$A = M \cdot (P + N \cdot L')$$

Representación de funciones booleanas

UNIVERSIDAD DE ANTIOQUIA

Ejemplo 2 - Solución

Paso 4 – Simplificación

Simplificación de la expresión derivada de la tabla de la verdad

$$A = M \cdot (P + N \cdot L')$$

		Pasos	Razón
1	$A =$	$MP'NL' + MPN'L' + MPN'L + MPNL' + MPNL$	Expresión sin simplificar
2	$=$	$M(P'NL' + PN'L' + PN'L + PNL' + PNL)$	Factor común para el producto $(\cdot)$ en **1**
3	$=$	$M(P'NL' + PN'L + PNL' + PNL + PN'L')$	Conmutatividad para la suma $(+)$ en **2**
4	$=$	$M\big(P'NL' + P(N'L + NL' + NL + N'L')\big)$	Factor común para el producto $(\cdot)$ en **3**
5	$=$	$M\big(P'NL' + P(N'L + N'L' + NL' + NL)\big)$	Conmutatividad para la suma $(+)$ en **4**
6	$=$	$M\Big(P'NL' + P\big(N'(L + L') + N(L' + L)\big)\Big)$	Factor común para el producto $(\cdot)$ en **5**
7	$=$	$M\big(P'NL' + P(N'\mathbf{1} + N\mathbf{1})\big)$	Complemento para la suma $(+)$ en **6**
8	$=$	$M\big(P'NL' + P(N' + N)\big)$	Identidad para el producto $(\cdot)$ en **7**
9	$=$	$M(P'NL' + P\mathbf{1})$	Complemento para la suma $(+)$ en **8**
10	$=$	$M(P'NL' + P)$	Identidad para el producto $(\cdot)$ en **9**
11	$=$	$M\big((P'+P)(NL' + P)\big)$	Distributividad para la suma $(+)$ en **10**
12	$=$	$M\big(\mathbf{1}(NL' + P)\big)$	Complemento para la suma $(+)$ en **11**
13	$=$	$M(NL' + P)$	Identidad para el producto $(\cdot)$ en **12**
14	$=$	$M(P + NL')$	Conmutatividad para la suma $(+)$ en **13**

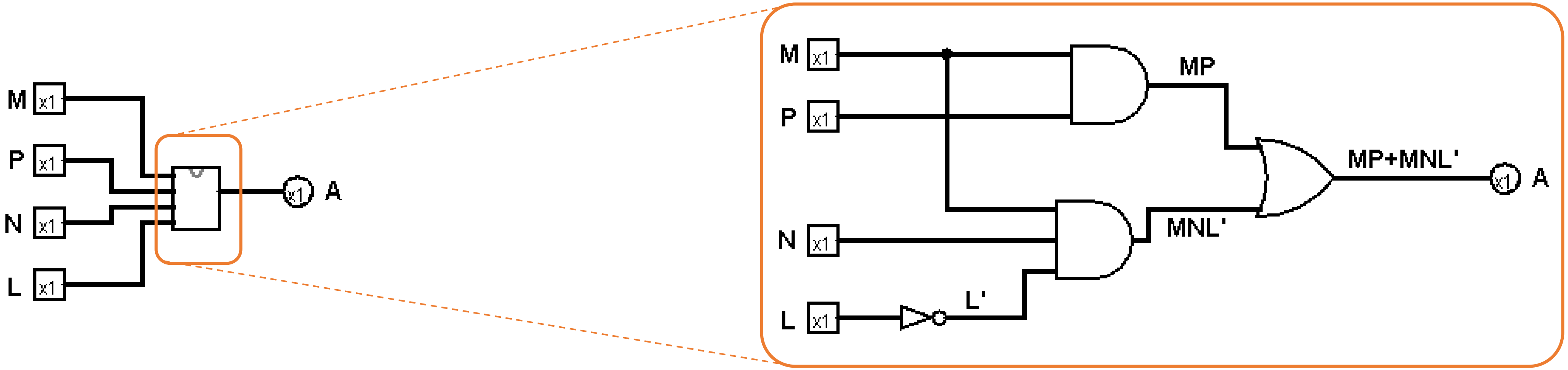
Representación de funciones booleanas

UNIVERSIDAD DE ANTIOQUIA

Ejemplo 2 - Solución

Paso 5 – Circuito digital

$$A = M \cdot P + M \cdot N \cdot L'$$



ejemplo2_f1_no_simplificada.cir

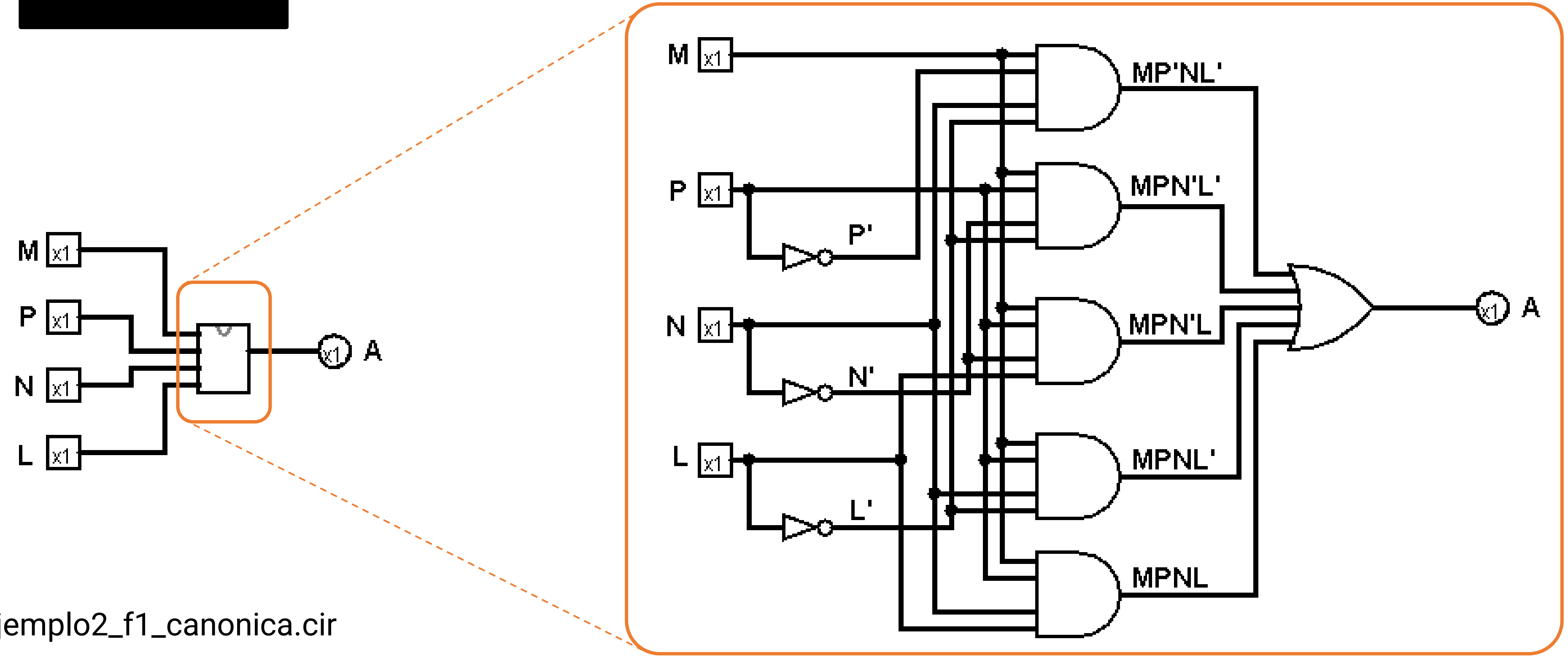
Representación de funciones booleanas

UNIVERSIDAD DE ANTIOQUIA

Ejemplo 2 - Solución

Paso 5 – Circuito digital

$$A = MP'NL' + MPN'L' + MPN'L + MPNL' + MPNL$$



ejemplo2_f1_canonica.cir

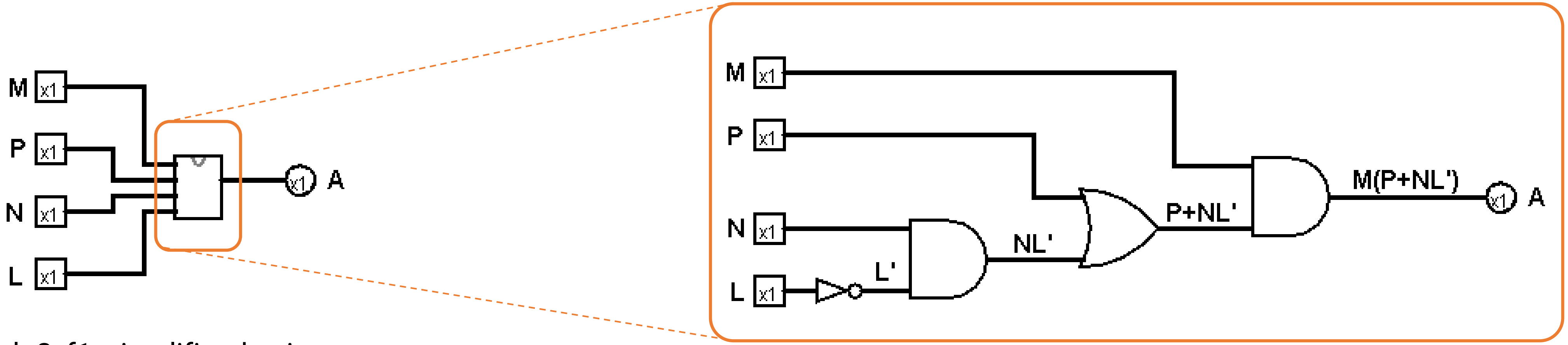
Representación de funciones booleanas

UNIVERSIDAD DE ANTIOQUIA

Ejemplo 2 - Solución

Paso 5 – Circuito digital

$$A = M \cdot (P + N \cdot L')$$



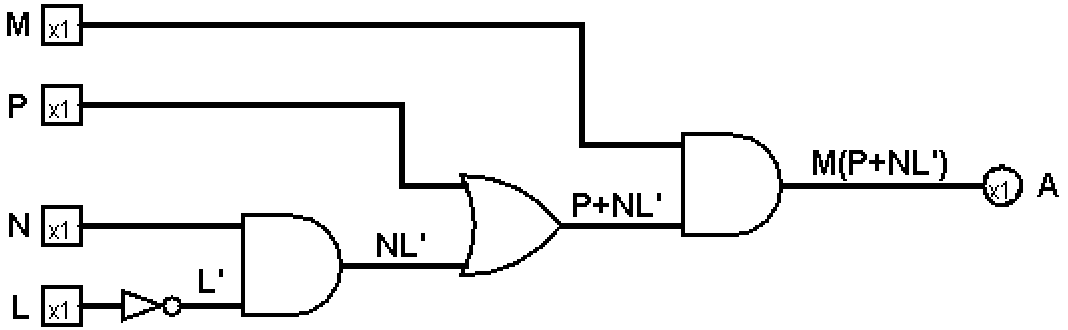
ejemplo2_f1_simplificada.cir

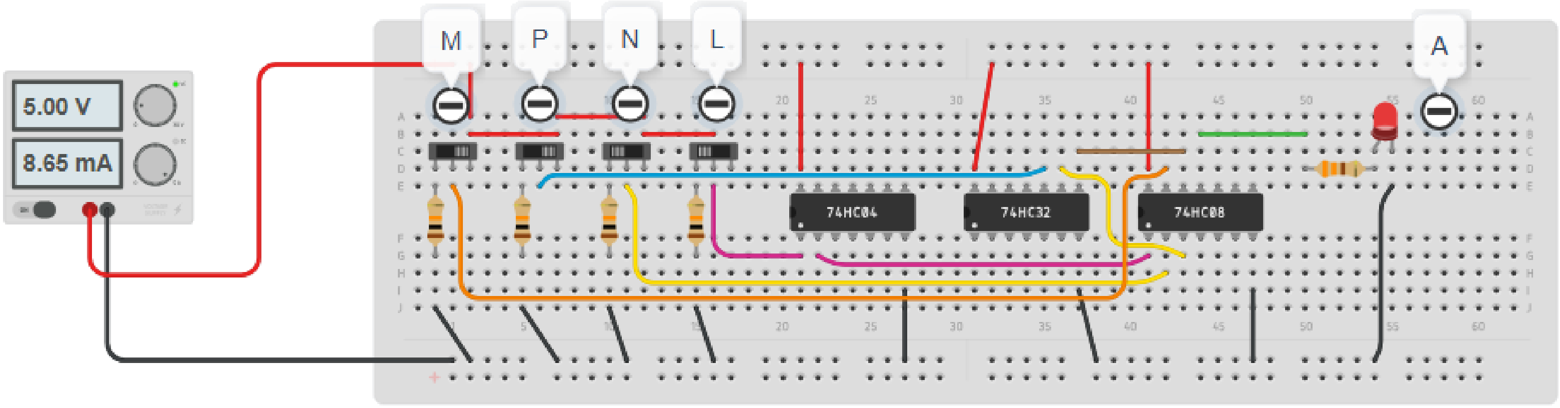
Representación de funciones booleanas

Ejemplo 2 - Solución

Paso 5 – Circuito digital

$$A = M \cdot (P + N \cdot L')$$





Alarma [link]

Representación de funciones booleanas

UNIVERSIDAD
DE ANTIOQUIA

Ejemplo 3

Encontrar la expansión de suma de productos para la función $F(x, y, z) = (x + y)z'$.

Representación de funciones booleanas

UNIVERSIDAD
DE ANTIOQUIA

Ejemplo 3 - Solución

Encontrar la expansión de suma de productos para la función $F(x, y, z) = (x + y)z'$.

Solución:

Para la solución se utilizaran dos métodos, primero utilizando una tabla y segundo utilizando identidades booleanas.

Método 1: Formar la suma de los minterminos correspondientes a cada fila de la tabla que tenga el valor 1.

Fila (i)	x	y	z	$x+y$	z'	$(x+y)z'$
0	0	0	0	0	1	0
1	0	0	1	0	0	0
2	0	1	0	1	1	1
3	0	1	1	1	0	0
4	1	0	0	1	1	1
5	1	0	1	1	0	0
6	1	1	0	1	1	1
7	1	1	1	1	0	0

$$F(x, y, z) = x'yz' + xy'z' + xy'z$$

$$F(x, y, z) = m_2 + m_4 + m_6$$

$$F(x, y, z) = \sum m(2,4,6)$$

Representación de funciones booleanas

UNIVERSIDAD DE ANTIOQUIA

Ejemplo 3 - Solución

Encontrar la expansión de suma de productos para la función $F(x,y,z) = (x+y)z'$.

Solución:

Para la solución se utilizaran dos métodos, primero utilizando una tabla y segundo utilizando identidades booleanas.

Método 2: Usar identidades booleanas para encontrar la forma normal disyuntiva.

		Pasos	Razón
1	$F(x,y,z) =$	$(x+y)z'$	Expresión no estandar
2	$=$	$xz' + yz'$	Distributividad para el producto $(\cdot)$ en **1**
3	$=$	$x\mathbf{1}z' + \mathbf{1}yz'$	Identidad para el producto $(\cdot)$ en **2**
4	$=$	$x(y+y')z' + (x+x')yz'$	Complemento para la suma $(+)$ en **3**
5	$=$	$xyz' + xy'z' + xyz' + x'yz'$	Identidad para el producto $(\cdot)$ en **4**
6	$=$	$xy'z' + xyz' + x'yz'$	Idempotencia para la suma $(+)$ en **5**
7	$=$	$x'yz' + xy'z' + xyz'$	Conmutatividad para la suma $(+)$ en **6**

$$F(x,y,z) = x'yz' + xy'z' + xy'z$$

$$F(x,y,z) = m_2 + m_4 + m_6$$

$$F(x,y,z) = \sum m(2,4,6)$$

Representación de funciones booleanas

UNIVERSIDAD
DE ANTIOQUIA

Ejemplo 4

Simplifique la expresión $F(x, y, z) = xyz + (x + y)(x + z)$ empleando el menor número de variables posible. Encuentre, además, las formas canónicas.

Representación de funciones booleanas

UNIVERSIDAD DE ANTIOQUIA

Ejemplo 4 - Solución

Simplifique la expresión $F(x,y,z) = xyz + (x+y)(x+z)$ empleando el menor número de variables posible. Encuentre, además, las formas canónicas.

Solución: Inicialmente vamos a realizar la simplificación mediante la aplicación de las identidades booleanas.

	Pasos		Razón
1	$F(x,y,z) =$	$xyz + (x+y)(x+z)$	**Expresión inicial**
2	$=$	$xyz + xx + xz + yx + yz$	Distributividad para el producto $(\cdot)$ en **1**
3	$=$	$xyz + x + xz + yx + yz$	Idempotencia $(\cdot)$ en **2**
4	$=$	$x + yz$	Absorción para la suma $(+)$ en **3**

$$F(x,y,z) = x + yz$$

Representación de funciones booleanas

UNIVERSIDAD DE ANTIOQUIA

Ejemplo 4 - Solución

Obtención de las formas canónicas (SOP y POS) – Forma 1: Procedemos a evaluar la expresión simplificada en una **tabla de verdad** y a partir de esta obtenemos las formas canónicas.

$$F(x,y,z) = x + yz$$

Fila (i)	x	y	z	yz	$x+yz$	m_i	M_i
0	0	0	0	0	0		M_0
1	0	0	1	0	0		M_1
2	0	1	0	0	0		M_2
3	0	1	1	1	1	m_3	
4	1	0	0	0	1	m_4	
5	1	0	1	0	1	m_5	
6	1	1	0	0	1	m_6	
7	1	1	1	1	1	m_7	

POS (Producto de sumas)

$$G(x,y,z) = \prod M(0,1,2)$$

SOP (Suma de productos)

$$F(x,y,z) = \sum m(3,4,5,6,7)$$

Representación de funciones booleanas

UNIVERSIDAD
DE ANTIOQUIA

Ejemplo 4

Obtención de las POS – Forma 2: Empleamos las identidades de algebra booleana para llegar a la expresión estándar a partir de la versión simplificada de la función.

	Pasos			Razón
1	$F(x,y,z)$	$=$	$x+yz$	Expresión inicial
2		$=$	$x\mathbf{11}+\mathbf{1}yz$	Identidad para el producto $(\cdot)$ en **1**
3		$=$	$x(y+y')(z+z')+(x+x')yz$	Complemento para la suma $(+)$ en **2**
4		$=$	$x(yz+yz'+y'z+y'z')+xyz+x'yz$	Distributividad para el producto $(\cdot)$ en **3**
5		$=$	$xyz+xyz'+xy'z+xy'z'+xyz+x'yz$	Distributividad para el producto $(\cdot)$ en **4**
6		$=$	$xyz'+xy'z+xy'z'+xyz+x'yz$	Idempotencia para el producto $(\cdot)$ en **5**
7		$=$	$x'yz+xy'z'+xy'z+xyz'+xyz$	Conmutatividad para la suma $(+)$ en **6**
8		$=$	$m_3+m_4+m_5+m_6+m_7$	Expresión de **7** como minterminos
9		$=$	$\sum m(3,4,5,6,7)$	Expresión abreviada de **8** como minterminos

Representación de funciones booleanas

UNIVERSIDAD DE ANTIOQUIA

Ejemplo 4

Obtención de las SOP – Forma 2: Empleamos las identidades de algebra booleana y ciertas definiciones para llegar a la expresión estándar a partir de la versión simplificada de la función.

	Pasos			Razón
1	1	$=$	$F(x,y,z)+F'(x,y,z)$	**Premisa:** Complemento para la suma $(+)$
2	1	$=$	$\sum m(3,4,5,6,7)+F'(x,y,z)$	Reemplazo de la POS de $F(x,y,z)$
3	$F'(x,y,z)$	$=$	$\sum m(0,1,2)$	Deducción de $F'(x,y,z)$ a partir del complemento para la suma $(+)$ en **2**
4	$F'(x,y,z)$	$=$	$m_0+m_1+m_2$	Expansión de la POS abreviada en **3**
5	$F'(x,y,z)$	$=$	$x'y'z'+x'y'z+x'yz'$	Representación de los minterminos de **4** en términos de las variables
6	$G(x,y,z)$	$=$	$(F'(x,y,z))'$	La SOP viene a ser igual al complemento de **5**
7	$G(x,y,z)$	$=$	$(x'y'z'+x'y'z+x'yz'\ )'$	Se reemplaza $F'(x,y,z)$ en **6**
8	$G(x,y,z)$	$=$	$(x'y'z')'(x'y'z)'(x'yz')'$	Ley de Morgan para el producto $(\cdot)$ en **7**
9	$G(x,y,z)$	$=$	$(x+y+z)(z+y+z')(x+y'+z\ )$	Ley de Morgan para la suma $(+)$ en **8**
10	$G(x,y,z)$	$=$	$M_0\cdot M_1\cdot M_2$	Expresión de la SOP en **9** como Maxterminos
11	$G(x,y,z)$	$=$	$\prod M(0,1,2)$	Expresión de la SOP en **10** en forma abreviada.

UNIVERSIDAD
DE ANTIOQUIA

Curso de Matemáticas Discretas 1

Clase 12 – Algebra de Boole