

Lógica Proposicional

En la creación de sistemas computacionales, es clave entender cómo funciona el razonamiento humano, ya que esto permite diseñar procesadores que tomen decisiones lógicas. Para esto se estudia la lógica, una disciplina que analiza cómo se pueden obtener conclusiones válidas a partir de ciertas premisas.[1]

Una parte fundamental de la lógica es la lógica proposicional, que se enfoca en los razonamientos más simples, basados en proposiciones (enunciados que pueden ser verdaderos o falsos, pero no ambos a la vez).[1]

Ejemplos:

- “Dos más dos son cuatro” (Verdadero)
- “Cinco más tres son nueve” (Falso)

Álgebra de Boole

La electrónica ha dado un salto enorme desde los setenta, permitiéndonos usar hoy en día máquinas programables para manejar cualquier dato imaginable. Estas máquinas se basan en señales digitales, que resultan ser más adaptables, veloces y exactas comparadas con las señales analógicas.[2]

Manejar sistemas digitales implica saber cómo utilizar dispositivos electrónicos que cumplen tres funciones primordiales:

- Operan con señales binarias, o sea, secuencias de unos y ceros.
- Ejecutan operaciones siguiendo una lógica matemática estandarizada a nivel global.
- Pueden almacenar los resultados de dichas operaciones y seguir un orden predefinido.

Para diseñar circuitos digitales de manera eficiente, necesitamos herramientas que faciliten la comprensión de su operación y el procesamiento de información.[2]

En 1938, Claude Shannon aprovechó el álgebra desarrollada por George Boole en 1854 para aplicar conceptos matemáticos al diseño de los circuitos eléctricos. Hoy conocemos esta herramienta como álgebra de Boole, y es fundamental para

comprender el funcionamiento de los sistemas digitales.[2]

Previo al uso del álgebra de Boole, es crucial comprender cómo se representan los números en los sistemas digitales, ya que esto mejora la comprensión del funcionamiento de estos dispositivos y la aplicación de sus reglas.[2]

PUERTAS LÓGICAS



Las **funciones lógicas básicas** se representan físicamente mediante **puertas lógicas**, como (AND, OR, NOT, NAND, NOR, XOR y XNOR) cada una tiene su tabla de verdad, su expresión lógica y su símbolo gráfico, que puede variar según el tipo de norma utilizada. Existen dos estilos de simbología:

- Americana (ANSI/IEEE 91-1973)
- Europea (ANSI/IEEE 91-1984)

La simbología europea se usa cada vez más por su claridad, ya que representa las funciones mediante rectángulos con símbolos internos, facilitando su lectura sin necesidad de explicaciones adicionales.[1]



Función Y (AND). Puerta lógica AND

La salida es 1 solo si todas las entradas son 1. Representa un producto lógico.[1]

Símbolo	Tabla de verdad	Función															
 simbología europea	 simbología americana	<table><tr><th>a</th><th>b</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table> $F = a \cdot b$	a	b	F	0	0	0	0	1	0	1	0	0	1	1	1
a	b	F															
0	0	0															
0	1	0															
1	0	0															
1	1	1															


Función O (OR). Puerta lógica OR

La salida es 1 si al menos una entrada es 1. Representa una suma lógica.[1]

Símbolo	Tabla de verdad	Función															
 simbología europea	 simbología americana	<table><tr><th>a</th><th>b</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table> $F = a + b$	a	b	F	0	0	0	0	1	1	1	0	1	1	1	1
a	b	F															
0	0	0															
0	1	1															
1	0	1															
1	1	1															

Función inversora NO (NOT). Puerta lógica NOT

La salida es contraria a la entrada. Se entiende como la negación lógica o el complemento.[1]

Símbolo	Tabla de verdad	Función						
	<table><tr><th>a</th><th>F</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	a	F	0	1	1	0	$F = \bar{a} = a' = a^*$
a	F							
0	1							
1	0							

Microprocesadores

Un microprocesador viene a ser un conjunto de circuitos digitales entrelazados que forman un sistema electrónico, concebido para llevar a cabo tareas muy concretas. Entre sus partes esenciales, figuran los registros, la Unidad Aritmético-Lógica (UAL) y un secuenciador, que se encarga de poner en orden el trabajo del resto de elementos.[1]

Las tareas primordiales que un microprocesador lleva a cabo abarcan el traspaso de información entre los registros y operaciones lógicas o matemáticas dentro de la UAL. Estas acciones, por sí mismas, no representan una gran ventaja, pero al ejecutarse siguiendo una secuencia marcada por un programa guardado en la memoria, hacen posible la ejecución de tareas complejas.[1]

El microprocesador se distingue por dos puntos clave:

- Su arquitectura, que plasma su estructura interna y los circuitos que la componen.
- Su modelo de programación, que es el conjunto de órdenes que es capaz de poner en marcha.

En sistemas más sofisticados, como los ordenadores, el microprocesador se refuerza con circuitos externos y programas que expanden lo que puede hacer. Gracias a los lenguajes de alto nivel, se puede programar sin necesidad de saber al detalle cómo está hecho por dentro, lo que da lugar a la idea de un controlador universal al alcance de casi todo el mundo.[1]

- [1] P. Novo, *Logica digital y microprogramable*. Marcombo, 2008. [Online].
Available: <https://elibro.net/es/lc/uteq/titulos/45920>
- [2] J. L. Duran, H. Martinez, and J. Domingo, *Electronica*. Marcombo, 2020. [Online].
Available: <https://elibro.net/es/lc/uteq/titulos/280313>