Arquitectura de Computadoras

Lógica Digital

J. Irving Vásquez
ivasquez@ccc.inaoep.mx

Centro de Innovación y Desarrollo Tecnológico en Cómputo

2 de marzo de 2016

Table of contents

Introducción

- Álgebra booleana
  - Desarrollada por George Boole Siglo XIX
  - Pretendia representar el pensamiento
  - Representación algebraica de la lógica binaria
  - Codifica verdadero como 1 y falso como 0
- Retomada por Claude Shannon
  - Tesis de Maestría 1937
  - Redes de relevadores con interruptores

Compuertas lógicas

- Funciones de dos variables
- Tablas de verdad

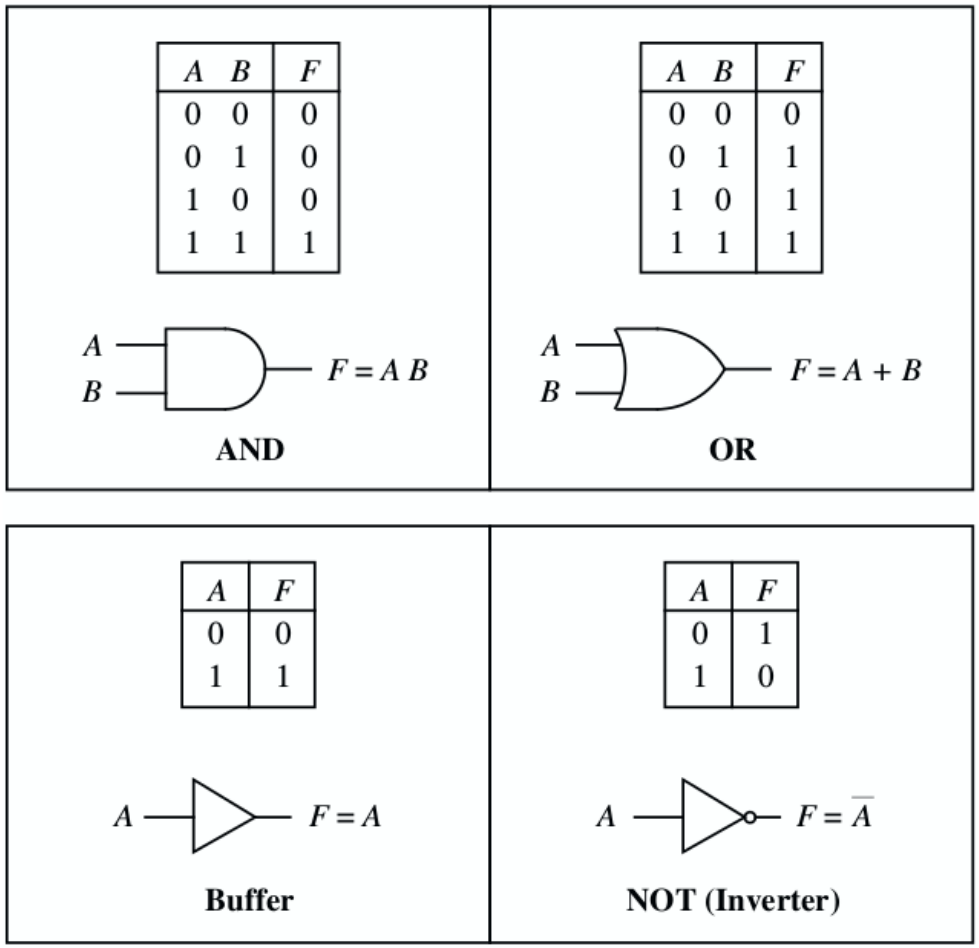


Figure A-5 Logic gate symbols for AND, OR, buffer, and NOT Boolean functions.

Compuertas lógicas

-

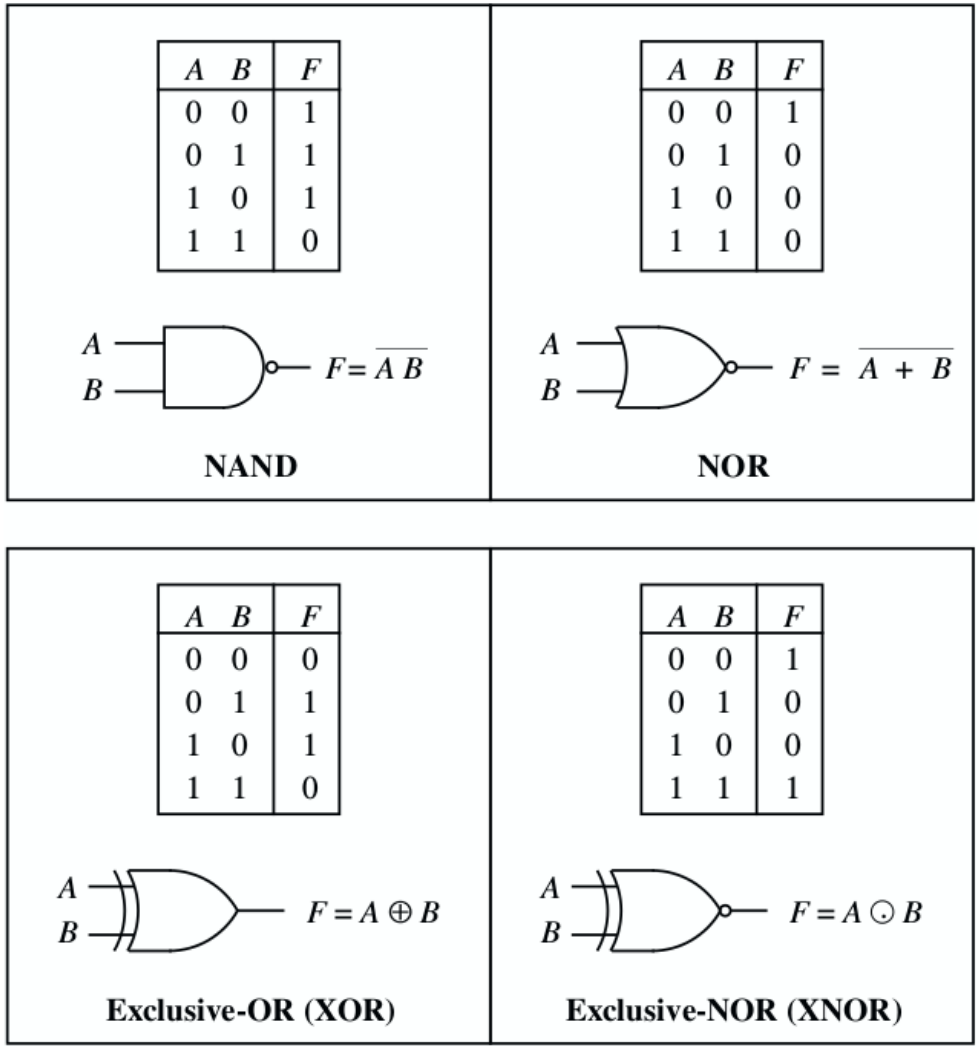


Figure A-6 Logic gate symbols for NAND, NOR, XOR, and XNOR Boolean functions.

Algebra booleana

- OR es la suma, $+$
- AND es el producto
- Complemento
- 0 identidad de la suma
- 1 identidad del producto

Propiedades

	Relationship	Dual	Property
Postulates	$A\,B = B\,A$	$A + B = B + A$	Commutative
	$A\,(B + C) = A\,B + A\,C$	$A + B\,C = (A + B)\,(A + C)$	Distributive
	$1\,A = A$	$0 + A = A$	Identity
	$A\,\overline{A} = 0$	$A + \overline{A} = 1$	Complement
Theorems	$0\,A = 0$	$1 + A = 1$	Zero and one theorems
	$A\,A = A$	$A + A = A$	Idempotence
	$A\,(B\,C) = (A\,B)\,C$	$A + (B + C) = (A + B) + C$	Associative
	$\overline{\overline{A}} = A$		Involution
	$\overline{A\,B} = \overline{A} + \overline{B}$	$\overline{A + B} = \overline{A}\,\overline{B}$	DeMorgan's Theorem
	$AB + \overline{A}C + BC$ $= AB + \overline{A}C$	$(A + B)(\overline{A} + C)(B + C)$ $= (A + B)(\overline{A} + C)$	Consensus Theorem
	$A\,(A + B) = A$	$A + A\,B = A$	Absorption Theorem

Table A.1 Basic properties of Boolean algebra.

Ejemplo Sumador

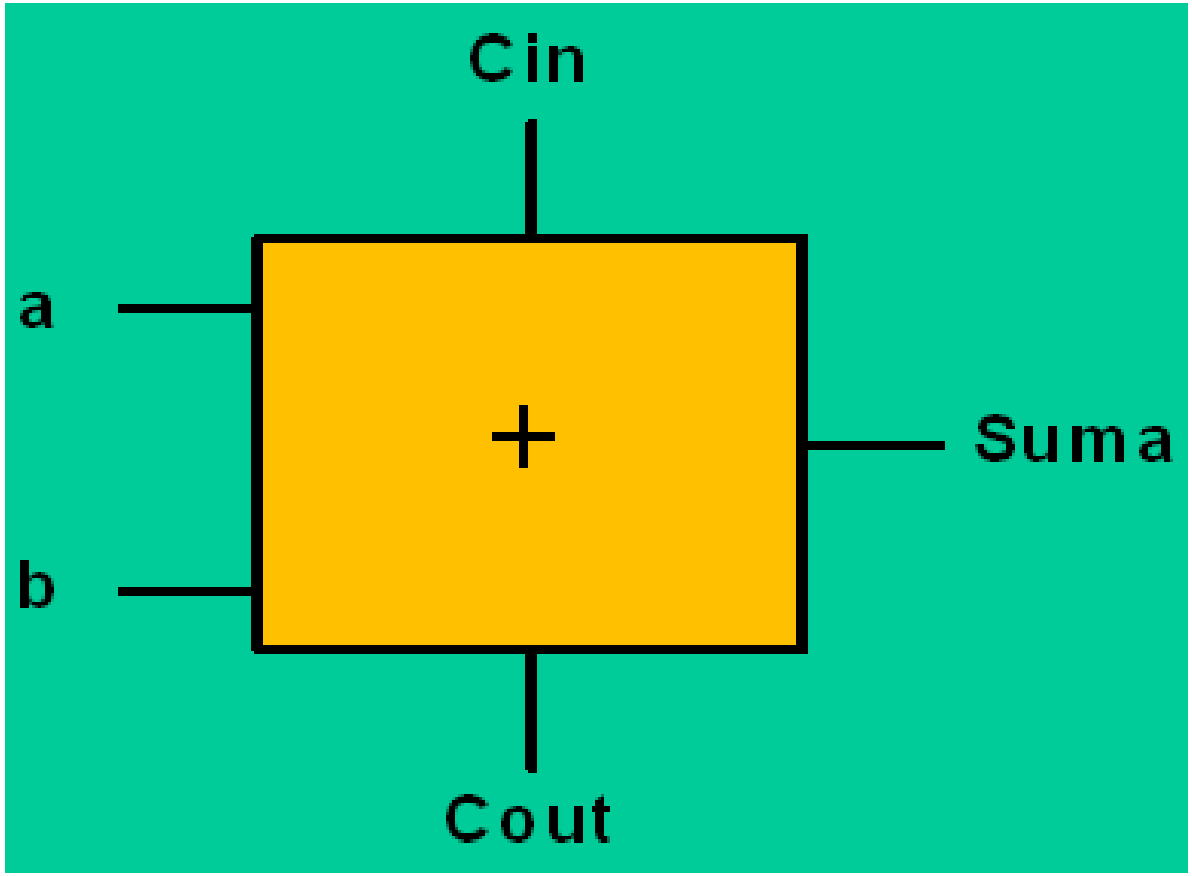


Figura : Sumador

Ejemplo Sumador

Entradas			Salidas		
a	b	Cin	Cout	Suma	Comentarios
0	0	0	0	0	0+0+0=00
0	0	1	0	1	0+0+1=01
0	1	0	0	1	0+1+0=01
0	1	1	1	0	0+1+1=10
1	0	0	0	1	1+0+0=01
1	0	1	1	0	1+0+1=10
1	1	0	1	0	1+1+0=10
1	1	1	1	1	1+1+1=11

Figura : Tabla de verdad del sumador

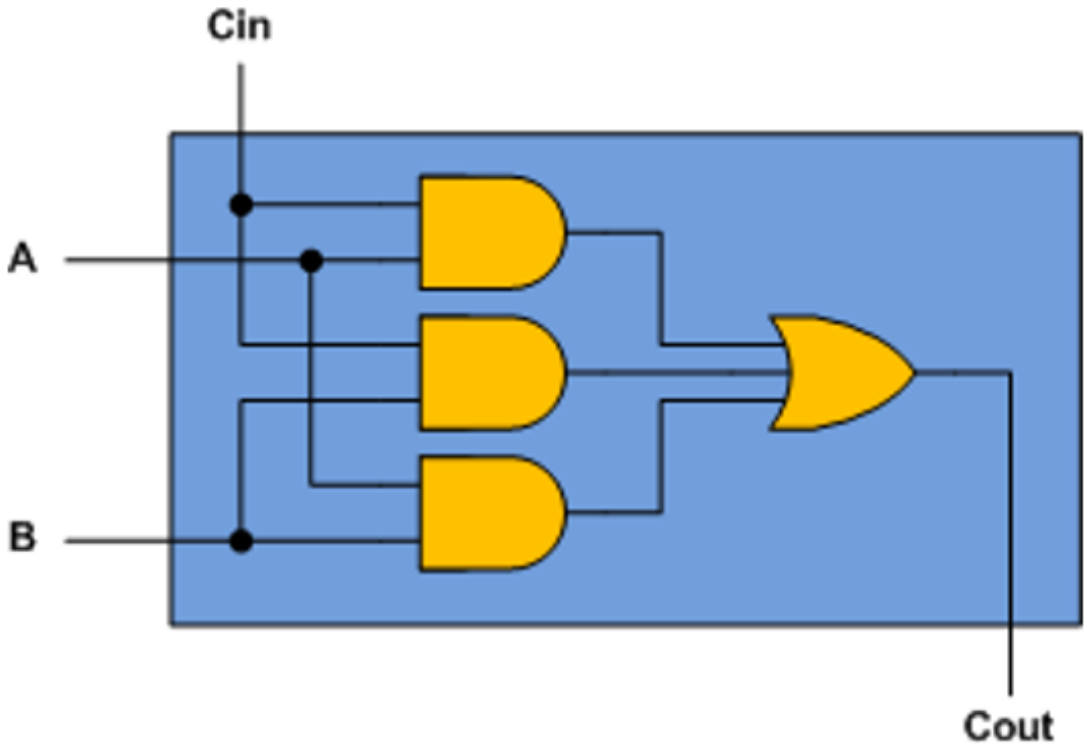
Ejemplo Sumador

Entradas			Salidas		
a	b	Cin	Cout	Suma	Comentarios
0	0	0	0	0	0+0+0=00
0	0	1	0	1	0+0+1=01
0	1	0	0	1	0+1+0=01
0	1	1	1	0	0+1+1=10
1	0	0	0	1	1+0+0=01
1	0	1	1	0	1+0+1=10
1	1	0	1	0	1+1+0=10
1	1	1	1	1	1+1+1=11

Figura : Tabla de verdad del sumador

- $cout = (!a * b * Cin) + (a * !b * cin) + (a * b * cin) + (a * b * Cin)$
- $cout = (cin * b) + (cin * a) + (a * b)$

Ejemplo Sumador



- Tarea (A mano). Demostrar simplificación
- Usar postulados y teoremas de lógica de boole

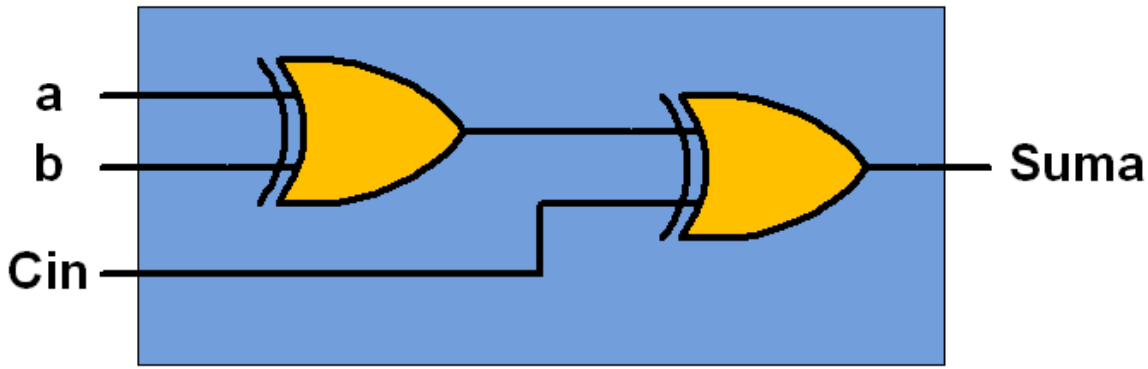
Ejemplo Sumador

Entradas			Salidas		
a	b	Cin	Cout	Suma	Comentarios
0	0	0	0	0	0+0+0=00
0	0	1	0	1	0+0+1=01
0	1	0	0	1	0+1+0=01
0	1	1	1	0	0+1+1=10
1	0	0	0	1	1+0+0=01
1	0	1	1	0	1+0+1=10
1	1	0	1	0	1+1+0=10
1	1	1	1	1	1+1+1=11

Figura : Tabla de verdad del sumador

- $suma = (!a*!b*cin) + (!a*b*!cin) + (a*!b*!cin) + (a*b*cin)$
- $suma = (a \oplus b) \oplus cin$, donde $\oplus$ es or exclusivo
- $a \oplus b = (a*!b) + (!a*b)$

Ejemplo Sumador



- Tarea (A mano). Demostrar la simplificación

Ejemplo Sumador

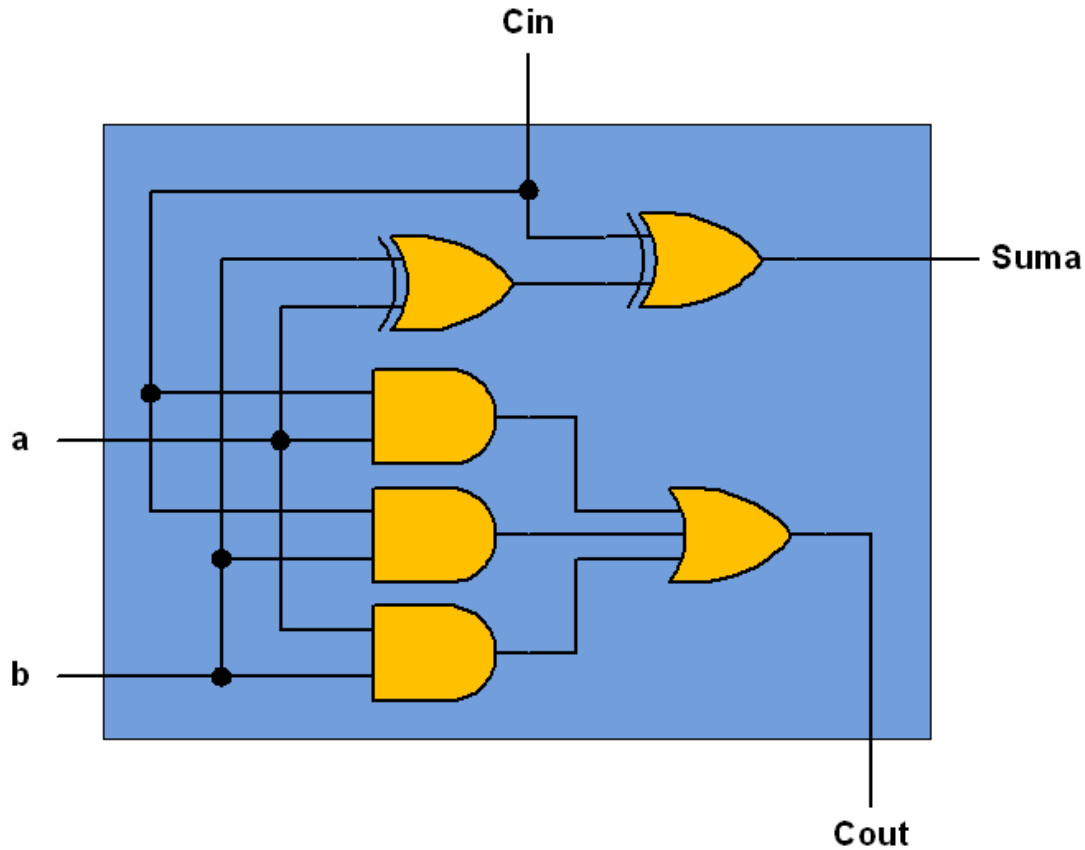


Figura : Sumador completo de 1 bit

ALU (and, or, suma)

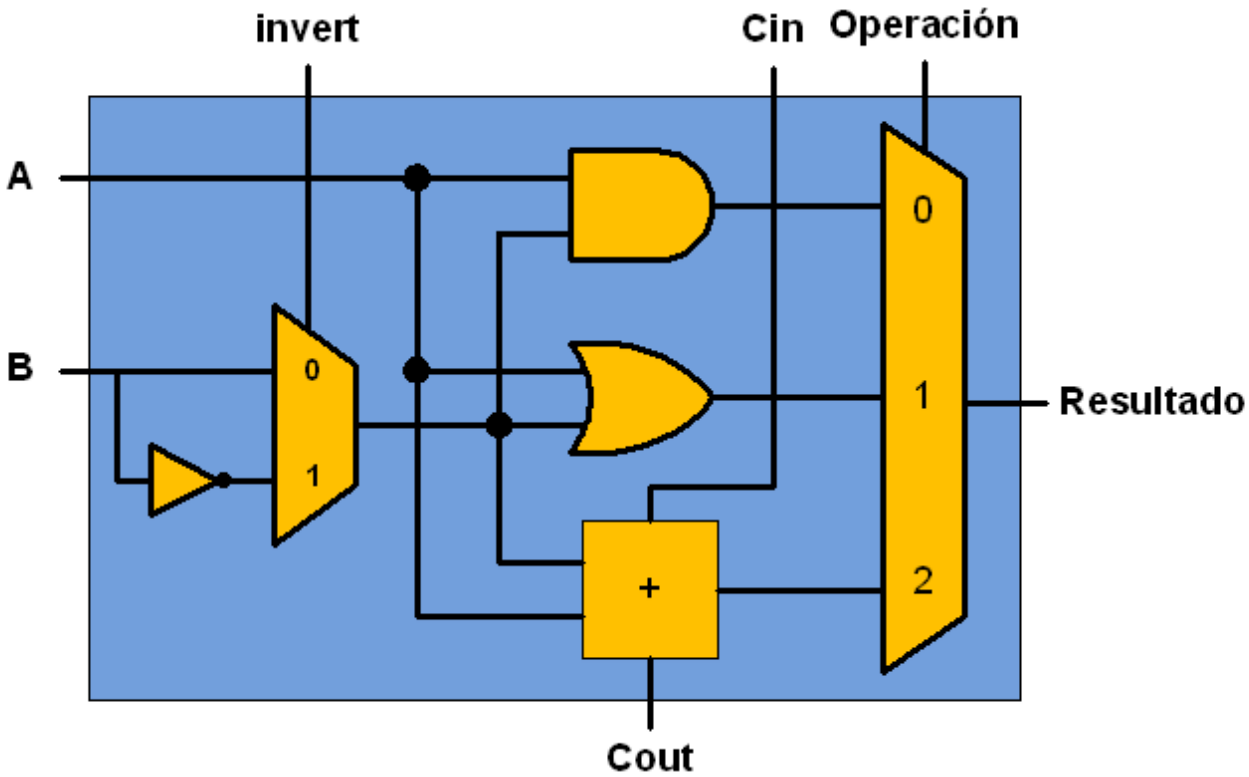


Figura : ALU de 1 bit

Referencias


William Stallings. Computer Organization and Architecture. Prentice Hall. 1993.


Miles J. Murdocca and Vincent P. Heuring. Principios de arquitectura de computadoras. Prentice Hall.