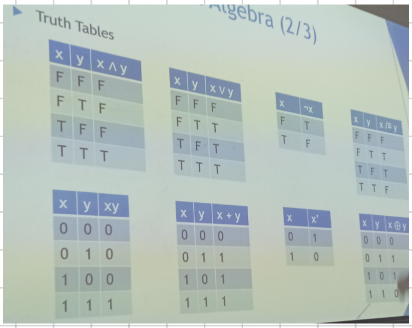
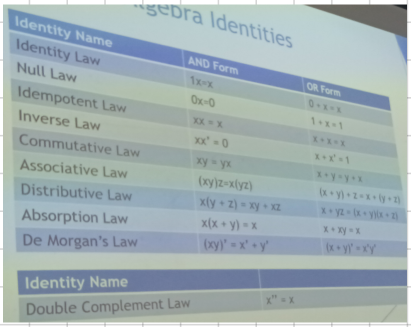
Boolean Algebra

- George Boole
  - The laws of thought (1854
  - Symbolic logic (Boolean Algebra)
- Claude Shannon
  - Two-valued Boolean Algebra could describe the operation of two-valued electrical switching circuits (1938)
- John Vincent Atanasoff
  - Applied Boolean Algebra to computing (1939)
  - Electricity → Computadoras hechas con electricidad
  - Capacitors (condensers) for memory
  - Base 2 numbers
  - Direct logic action (Boolean algebra - trial and error)

Logic vs Boolean Algebra

Logic	Boolean Algebra
B = {F, T}	B = {0, 1}
$p \wedge q$	pq
$p \vee q$	$p + q$
$\neg p$	p' → Complemento
$\not\equiv$	$p \oplus q$ ↳ o exclusivo
Formula/expresión/Proposition	Boolean functions
$\equiv$ theorems/Axioms	Boolean Algebra Identities
• $\equiv, \not\equiv, \neg, \wedge, \vee$	
Derivation	Boolean Function Simplification

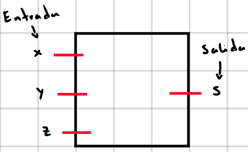




→ y todos los Axiomas/teoremas de LCAT/LYMD

- Problem solving using digital circuits (Hardware)
  - Problem Description
  - Truth table → Vamos en reversa, de tabla a fórmula
  - Boolean function
  - Boolean Function Simplification
  - Digital Circuit construction

- Ej: Given three bits identify if there are more ones than zeros.
  - truth table



x	y	z	s
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

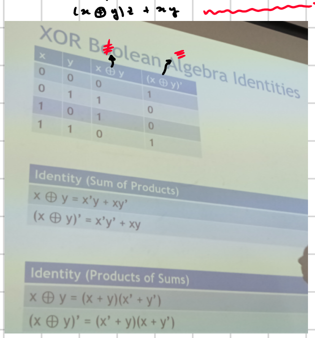
Negar a 0 (Complemento a 1)

$s = x'yz + xy'z + xyz' + xyz$ (14 operaciones)

$s = (x + y + z)(x + y + z')(x + y' + z)(x' + y + z)$ (14 operaciones)

Simplificación → Usan de propiedades

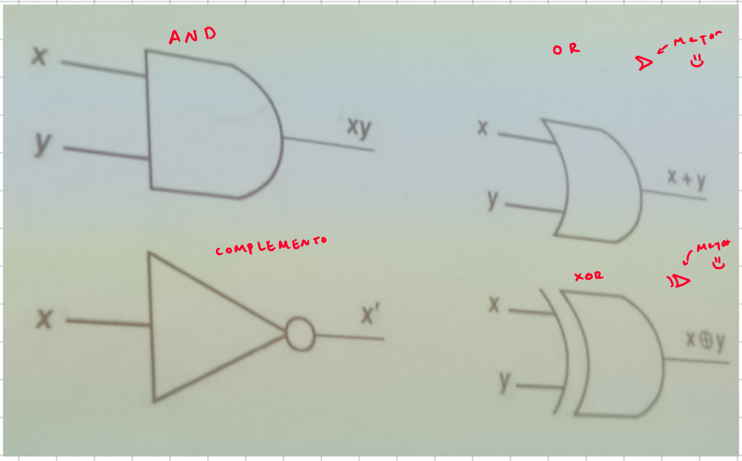
s
$\equiv$ ⟨Suma de productos⟩
$x'yz + xy'z + xyz' + xyz$
$\equiv$ ⟨Distributividad⟩
$(x'y + xy')z + xy(z' + z)$
$\equiv$ ⟨Inverso⟩
$(x'y + xy')z + xy1$
$\equiv$ ⟨Identidad⟩
$(x'y + xy')z + xy$
$\equiv$ ⟨def XOR⟩
$(x \oplus y)z + xy$ → De 14 a 4! (!)

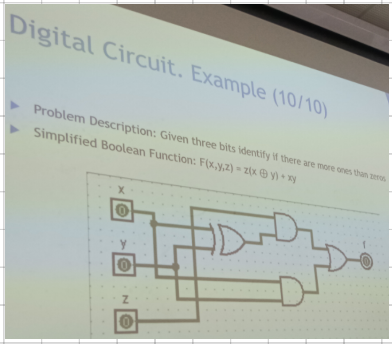


- Formas estandar:
  - Suma de productos:
    - Más ceros que unos en la tabla
    - Focus on rows with value 1
  - Productos de sumas:
    - Mas unos que ceros
    - Focus on rows with value 0

MyScript Notes

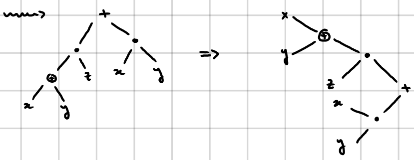
• Construcción - compuertas lógicas

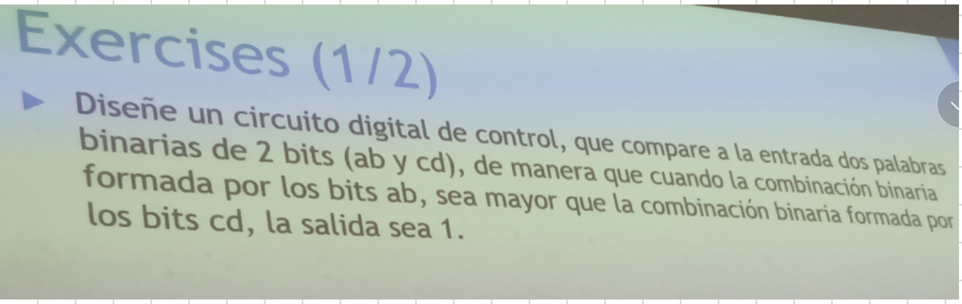




Construcción:

$(x \oplus y)z + xy$ ⇝





a	b	c	d	m
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

$$m = a'bc'd' + ab'c'd' + \underline{ab'c'd} + \underline{abc'd'} + \underline{abc'd} + \underline{abcd'}$$

$$= (a'b + ab')c'd' + ac'(b'd + bd') + ab(c'd + cd')$$

$$= (a \oplus b)c'd' + ac'(b \oplus d) + ab(c \oplus d)$$

$$= (a \oplus b)c'd' + a(c'(b \oplus d) + b(c \oplus d))$$

... que cuando la combinación binaria ... entrada dos palabras ... sea mayor que la combinación binaria formada por ... la salida sea 1.

En una familia de cuatro miembros (padre, madre, hermano y hermana), apasionados de la electrónica, emplean el siguiente procedimiento a la hora de ver la tele: a) Deciden los padres. b) Si no se ponen de acuerdo, deciden los hijos. c) Si tampoco se ponen de acuerdo los hijos, se hará lo que diga la madre.

→ Entrada: 1 de acuerdo; 0 desacuerdo

→ Salida → Quien toma la decisión

p	m	ho	ha	m	h	m
0	0	0	0	0	0	1
0	0	0	1			
0	0	1	0			
0	0	1	1			
0	1	0	0			
0	1	0	1			
0	1	1	0			
0	1	1	1			
1	0	0	0			
1	0	0	1			
1	0	1	0			
1	0	1	1			
1	1	0	0			
1	1	0	1			
1	1	1	0			
1	1	1	1			

Una máquina-herramienta tiene cuatro detectores de seguridad, 2 superiores y 2 inferiores. La máquina se para cuando se accionen, simultáneamente, al menos un detector superior y un detector inferior.

a	b	c	d	e
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

$e = a'bc'd + a'bcd' + a'bcd + ab'c'd + ab'cd' + abc'd + abcd' + abcd + ab'cd$

$e = a'b(c'd + cd') + a'bcd + ab'(c'd + cd') + abc(d' + d) + ab'c(d' + d)$

$e = a'b(c \oplus d) + a'bcd + ab'(c \oplus d) + abc \cdot 1 + ab'c \cdot 1$

$e = (a'b + ab')(c \oplus d) + a'bcd + abc + ab'c$

$e = (a \oplus b)(c \oplus d) + a'bcd + ac(b + b')$

$e = (a \oplus b)(c \oplus d) + a'bcd + ac\,1$

$e = (a \oplus b)(c \oplus d) + c(a'bd + a)$

MyScript Notes