CIRCUITI LOGICI COMBINATORI

1) Sono circuiti che ricevono in input uno o più bit ed in uscita forniscono un bit

2) Sono realizzati utilizzando porte logiche (AND, OR, NOT, XOR NOR, NAND, NXOR)

3) Le uscite dipendono solo dagli ingressi forniti e NON dal valore delle uscite precedenti: cioè sono circuiti logici privi di memoria del passato.

NOTA: (non comprensi
Il punto 3) vuol dire che le uscite delle singole porte NON ritornano come ingressi a porte che rispetto alla direzione di propagazione dell'elaborazione stanno prima

Per realizzare ed analizzare circuiti logici combinatori serve conoscere:

1) Tabelle di verità
2) Algebra Booleana
3) Porte logiche
4) Mappe di Karnaugh

TABELLE DI VERITÀ

Se abbiamo 1 bit al suo posto possiamo scrivere:

bit

0
1

$\left.\begin{matrix}0\\1\end{matrix}\right\}$ 2 possibilità

Se abbiamo 2 bit al loro posto possiamo scrivere:

0	0
0	1
1	0
1	1

4 possibilità

Se abbiamo3 bit

0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

8 casi

In generale se abbiamo N bit possiamo scrivere su di essi 2^N **possibilità**

TABELLE DI VERITÀ

Per scrivere tutte le possibilità degli N bit si procede così:
- il numero di casi è 2^N
- nella prima colonna a destra, 0 e 1 si alternano a uno a uno
- nella seconda colonna 0 e 1 si alternano a 2 a 2
- nella terza colonna 0 e 1 si laternano a 4 a 4
- ecc. ecc. nelle altre colonne, se ci sono

In questo modo si ottiene la "Tabella di Verità"

colonna D si alternano 1 a 1

colonna C si alternano a 2 a 2

colonna B si alternano a 4 a 4

colonna A si alternano a 8 a 8

Per esempio abbiamo 4 bit, chiamiamoli A,B,C e D.

Avremo $2^4 = 16$, quindi 16 casi (righe della tabella)

Colonne

A	B	C	D
0	0	0	0
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

per comodità li abbiamo raggruppato le righe a 4 a 4

16 righe

TABELLE DI VERITÀ - numeri binari

Ogni riga della tabella di verità corrisponde ad un numero binario che rappresenta un valore in decimale.

0 0 0 → Vale 0

0 0 1 → Vale 1

1 0 1 → Vale 5

1 1 0 → Vale 6

Il valore di un numero binario si ottiene sommando i valori dei bit che sono 1.

Un bit 1 vale in base alla posizione partendo da destra e andando a sinistra.

Valore	16	8	4	2	1
posizione	5	4	3	2	1

0 1 0 1 0 → Vale 8 + 2 = 10

0 0 1 1 1 → Vale 4 + 2 + 1 = 7

TABELLE DI VERITÀ - numeri binari

Se il primo bit (quello più a destra) vale 1 allora la quantità è un valore dispari

Se il primo bit è 0 allora il valore è pari

Valore	4	2	1	
bit	3	2	1	
	•	•	•	
	0	1	0	→ Vale 2 pari
	1	1	0	→ Vale 4+2 = 6 pari
	1	0	1	→ Vale 4+1 = 5 dispari
	0	1	1	→ Vale 2+1 = 3 dispari

LOGICA BOOLEANA

E' una matematica dove sono possibili solo due valori 0 e 1.

Si utilizzano delle variabili che possono assumere uno di questi due valori e si indicano in genere con lettere maiuscole: A, B, C,D, X,Y,Z ecc. ecc.

Su questi due valori 0 e 1 sono definite alcune operazioni: AND, OR, NOT, XOR, XNOR, NAND, NOR.

NOT, AND e OR sono operazioni base
Le operazioni XOR, XNOR, NAND e NOR si possono ottenere utilizzando le operazioni base.

Le operazioni binarie o booleane si definiscono mediante tabella di verità

NOT
not di A oppure A negato $\overline{A}$

A	$\overline{A}$
0	1
1	0

OR
A or B oppure A più B $A + B$

A	B	$A+B$
0	0	0
0	1	1
1	0	1
1	1	1

AND
A and B oppure A per B $A \cdot B$

A	B	$A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

LOGICA BOOLEANA

NAND, NOR, XOR e XNOR si possona anche ottenere mediante calcoli con le sole operazioni base (NOT, AND e OR)

NAND $\overline{A \cdot B}$

A	B	$\overline{A \cdot B}$
0	0	1
0	1	1
1	0	1
1	1	0

XOR $A \oplus B$

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

NOR $\overline{A + B}$

A	B	$\overline{A + B}$
0	0	1
0	1	0
1	0	0
1	1	0

XNOR $\overline{A \oplus B}$

A	B	$\overline{A \oplus B}$
0	0	1
0	1	0
1	0	0
1	1	1

LOGICA BOOLEANA -- Espressioni

$$Y = A \cdot B + C + \overline{A \cdot C} + \overline{B}$$

è un'espressione booleana, il risultato si ottiene sostituendo i valori ad A,B e C e facendo icalcoli con le operazioni logiche definite prima

Se fosse $A = 1$, $B = 1$ e $C = 0$ avremmo

$$Y = 1 \cdot 1 + 0 + \overline{1 \cdot 0} + \overline{1} = 1 + 0 + \overline{0} + 0 = 1 + 0 + 1 + 0 =$$

$$= (1+0) + 1 + 0 = 1 + 1 + 0 = (1+1) + 0 = 1 + 0 = 1$$

LOGICA BOOLEANA -- Proprietà

$\overline{\overline{A}} = A$

$A + 0 = A$

$A + 1 = 1$

$A \cdot 0 = 0$

$A \cdot 1 = A$

$A + A = A$

$A + \overline{A} = 1$

$A \cdot A = A$

$A \cdot \overline{A} = 0$

$AB = AB(C + \overline{C}) = ABC + AB\overline{C}$

compare C che non era presente

$ABC + A\overline{B}C = AC(B + \cancel{\overline{B}}) = AC$

cioè scompare B

DISTRIBUTIVITÀ

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

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

ASSORBIMENTO

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

$A + \overline{A} \cdot B = A + B$ $\quad$ $A \cdot (\overline{A} + B) = A \cdot B$

LOGICA BOOLEANA -- De Morgan

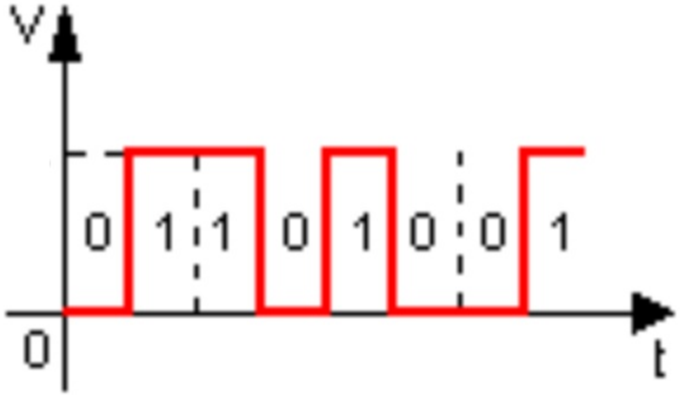
$$\overline{(A \cdot B)} = \overline{A} + \overline{B}$$

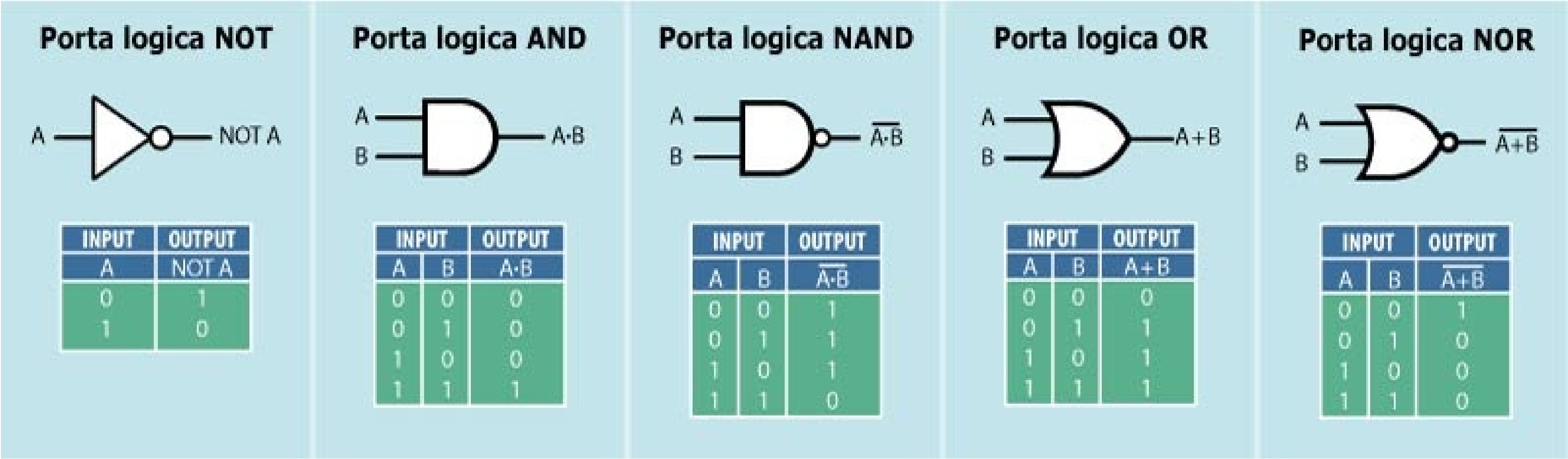
$$\overline{(A + B)} = \overline{A} \cdot \overline{B}$$

PORTE LOGICHE

Sono circuiti che funzionano come le operazioni dell'algebra Booleana.

Questi circuiti ricevono e forniscono 0 e 1 come tensioni elettriche.
Per esempio le portedel tipo(si dice famiglia logica) TTL considerano 1 una tensione di circa 5 V e 0 una tensione di circa 0 V.





INPUT	OUTPUT
A	NOT A
0	1
1	0

INPUT		OUTPUT
A	B	A·B
0	0	0
0	1	0
1	0	0
1	1	1

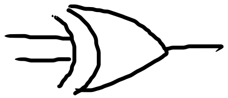
INPUT		OUTPUT
A	B	$\overline{A \cdot B}$
0	0	1
0	1	1
1	0	1
1	1	0

INPUT		OUTPUT
A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

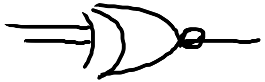
INPUT		OUTPUT
A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

PORTE LOGICHE

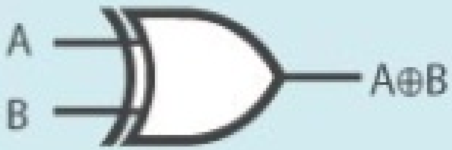
XOR



XNOR

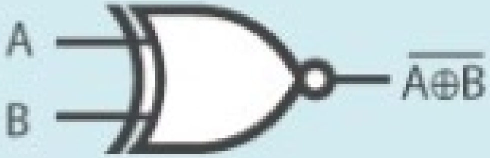


Porta logica XOR



INPUT		OUTPUT
A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

Porta logica XNOR



INPUT		OUTPUT
A	B	$\overline{A \oplus B}$
0	0	1
0	1	0
1	0	0
1	1	1

Metodo dei mintermini

Data una tabella di verità di una espressione logica che non conosciamo (nella la colonna più a destra c'e' il risultato dell'espressione per ogni caso) otteniamo l'espressione.

Ogni riga è una delle possibilità che le variabili hanno.

A	B	C	U	
0	0	0	0	
0	0	1	0	
0	1	0	1	$\rightarrow \overline{A}\cdot B\cdot\overline{C}$
0	1	1	0	
1	0	0	1	$\rightarrow A\cdot\overline{B}\cdot\overline{C}$
1	0	1	0	
1	1	0	1	$\rightarrow A\cdot B\cdot\overline{C}$
1	1	1	0	

3 Variabili

$\Downarrow$

$2^3 = 8$ casi

L'espressione logica corrispondente alla tabella si ottiene considerando le righe che danno risultato 1.

Per ognuna di queste righe si scrive un AND tra le variabili considerate negate se al loro posto c'e' 0 e non negate se c'e' 1.

Tra le and ottenute si fa OR.

$$U = \overline{A}\cdot B\cdot\overline{C} + A\cdot\overline{B}\cdot\overline{C} + A\cdot B\cdot\overline{C}$$

Semplificare espressione Logica

Per semplificare un'espressione logica ci sono due strade:
- proprietà dell'algebra boolena
- Karnaugh