

Laboratorio di Informatica

Dr Carlo Meneghini

Dip. di Fisica "E. Amaldi"

via della Vasca Navale 84

meneghini@fis.uniroma3.it

st. - 134 -

tel.: 06 55177217

IV. Operazioni automatiche

La somma

Sistema decimale

		10 ²	10 ¹	10 ⁰	
176	+	1	7	6	
28	=	0	2	8	
		1	1		Riparti
204		2	0	4	

C. Meneghini - Laboratorio di informatica 2006/07

2

Somma Sistema binario

Somma	Somma
00011001+	00001001+
00100110=	00101110=
00111111	0001000
	00110111
	riporti
dec.	dec.
25+	9+
38=	46=
63	55

C. Meneghini - Laboratorio di informatica 2006/07

3

Somma algebrica Sistema binario

25 - 7 = 18

1) rappresentazione di -7 in complemento a 2

		1	1	1	1	0	0	1	riporti
7	0000	0111							
-7	1111	1000	+						
	0000	0001	=						
	1111	1001							

bit di segno

-m è uguale a $2^N - m$

Il riporto (oltre l'ottavo bit) non viene considerato

C. Meneghini - Laboratorio di informatica 2006/07

4

Automatizzare i calcoli

In un computer le operazioni (matematiche, algebriche, logiche etc...) si possono effettuare combinando componenti elettronici (porte) che realizzano operazioni elementari.

Quali sono le operazioni fondamentali?

C. Meneghini - Laboratorio di informatica 2006/07

5

Numeri interi

Prodotto

12 · 4

12 +

12 +

12 +

12 =

48

Quoziente

12 / 4

12 - 4 = 8 → 1

8 - 4 = 4 → 1

4 - 4 = 0 → 1

3

Mediante somme algebriche è possibile effettuare prodotti, quozienti e elevamento a potenza.

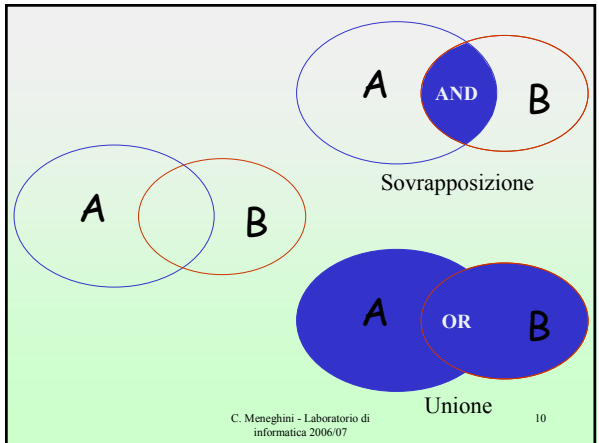
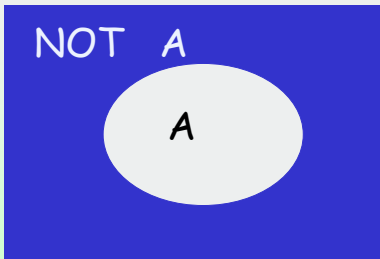
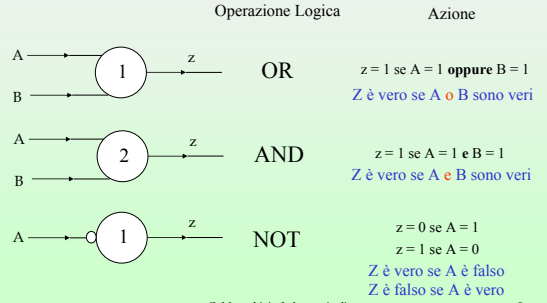
C. Meneghini - Laboratorio di informatica 2006/07

ALGEBRA di BOOLE (~1850)

Simboli	matematica	Logica (proposizioni)
1	1	vero
0	0	falso

Operatori	matematica	Logica (proposizioni)
+	addizione	disgiunzione (OR)
×	prodotto	coniunzione (AND)
-	negativo	negazione (NOT)

Le operazioni logiche (1940)



Le operazioni logiche elementari

NOT(A)
 $\bar{A}$

A OR B
A + B

A AND B
A × B

A	$\bar{A}$
0	1
1	0

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

A	B	$A \times B$
0	0	0
0	1	0
1	0	0
1	1	1

Completezza

Le operazioni logiche fondamentali: NOT, OR, AND costituiscono un sistema funzionale completo di operatori logici, cioè consentono di realizzare qualunque funzione logica possibile.

Priorità: in una sequenza di operazioni logiche

- prima vengono eseguite le operazioni unarie (**NOT**)
- quindi le operazioni **AND**
- e infine gli **OR**

Operazioni logiche complesse

$$\text{NOT } A \times B + A \times \text{NOT } B$$

$$A = 0, B = 1$$

$$\text{NOT } 0 \times 1 + 0 \times \text{NOT } 1 =$$

$$1 \times 1 + 0 \times 0 =$$

$$1 + 0 =$$

$$1$$

$$\text{NOT } A \times (B + A) \times \text{NOT } B$$

$$A = 0, B = 1$$

$$\text{NOT } 0 \times (1 + 0) \times \text{NOT } 1 =$$

$$1 \times 1 \times 0 =$$

$$0$$

$$\text{NOT } A \times (B + A \times \text{NOT } B)$$

$$A = 0, B = 1$$

$$\text{NOT } (0 \times 1 + 0 \times \text{NOT } 1) =$$

$$\text{NOT } (0 \times 1 + 0 \times 0) =$$

$$\text{NOT } (0 + 0) =$$

$$1$$

C. Meneghini - Laboratorio di
informatica 2006/07

13

Altri operatori logici

$$A \text{ XOR } B$$

$$A \oplus B$$

"or esclusivo"

A	B	A⊕B
0	0	0
0	1	1
1	0	1
1	1	0

vero se è vero A o
B ma non tutti e
due veri

$$A \text{ NAND } B$$

$$A \otimes B$$

$$A \times B$$

A	B	A⊗B
0	0	1
0	1	1
1	0	1
1	1	0

NOT(AND)

$$A \text{ NOR } B$$

$$A + B$$

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

NOT(OR)

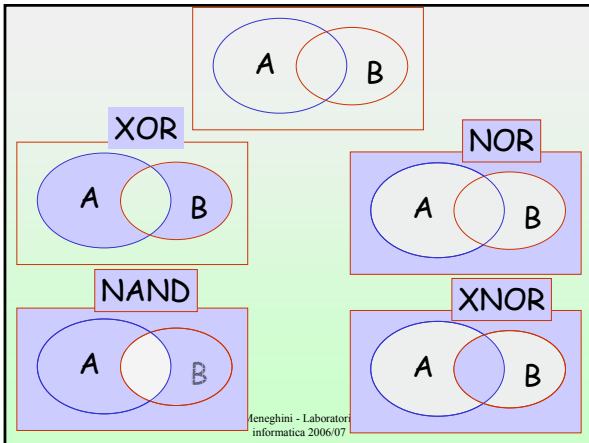
$$A \text{ XNOR } B$$

A	B	A⊙B
0	0	0
0	1	1
1	0	1
1	1	0

NOT(XOR)

C. Meneghini - Laboratorio di
informatica 2006/07

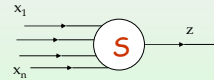
14



Meneghini - Laboratorio di
informatica 2006/07

Le porte logiche

L'uso del termine "porte logiche" utilizzato tuttora si ispira al funzionamento dei modelli di celle neuronali (1931)

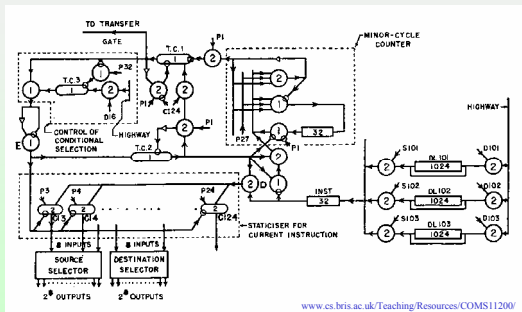


Neurone artificiale: una "cella" cui afferiscono un certo numero di segnali di input ($x_1 \dots x_n$) che possono essere attivi (1) o inattivi (0). Il neurone fornisce un segnale attivo in uscita (**z**) se il numero di segnali attivi in ingresso è maggiore o uguale a **S**.

1 = Vero 0 = Falso

informatica 2006/07

16

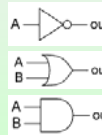


www.cs.bris.ac.uk/Teaching/Resources/COMS11200/

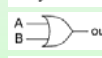
C. Meneghini - Laboratorio di
informatica 2006/07

17

Le porte logiche fondamentali



NOT



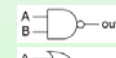
OR



AND



XOR



NAND



NOR



XNOR

C. Meneghini - Laboratorio di
informatica 2006/07

18

Funzioni logiche elementari					
1	0	1	1	1	1
AND			AND		
1					0
0		0			
OR			OR		
0					1
0					
NOT					
1					

1	1	XOR	0
1	1	NAND	0
1	1	NOR	0
1	1	XNOR	1

The screenshot shows a Google search page with the following elements:

- Search Bar:** Contains the text "cinema roma -adriano".
- Buttons:** "Trovare risultati" (Find results) and "Ricerca avanzata" (Advanced search).
- Filters:** "All Sources", "Journals", "Books", "Reference Works", "Abstract Databases", "Scopus".
- Search Criteria:**
 - Term(s):** A text box containing "Enter terms using Boolean connectors (see: cut OR full AND not/and)".
 - Source(s):** Checkboxes for "Journals", "Book Series", "Handbooks", "Reference Works", and "Abstract Databases".
 - Subject:** A dropdown menu showing "Arts & Humanities" selected, with other options like "Agricultural and Biological Sciences", "Life Sciences", "Mathematics", "Medicine", "Physical Sciences", and "Social Sciences".
 - Date:** A date range selector set to "1995" to "Present", with an "All Years" option.
- Buttons:** "Search", "Clear", "Recall Search", and "Search Tips".
- Results:** A list of search results, including "Cinema Roma" and "Cinema Roma - Adriano".

Circuiti logici

XOR

$$A \oplus B = (\overline{A} \times B) + (A \times \overline{B})$$

A	B	A⊕B
0	0	0
0	1	1
1	0	1
1	1	0

Diagram illustrating the implementation of the XOR operation using logic gates. The circuit uses two 3-input AND gates and one 3-input OR gate. The inputs A and B are connected to the first two inputs of both AND gates. The third input of the first AND gate is connected to NOT B (001), and the third input of the second AND gate is connected to NOT A (011). The outputs of the AND gates (000 and 010) are connected to the inputs of the OR gate, which produces the final XOR output (010).

C. Meneghini - Laboratorio di informatica 2006/07

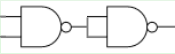
22

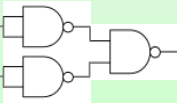
$A \oplus B = (\overline{A} \times B) + (A \times \overline{B})$

NOT (A) x B + NOT (B) x A			
a	b		
1	0	1	0
NOT			NOT
0	0	1	1
AND		AND	
0		1	
0		1	
OR			
1			

NAND e NOR

A  $\overline{A}$ NOT

A  $A \times B$ AND

A  $A+B$ OR

Un solo tipo di circuito: il NAND è sufficiente per effettuare tutte le operazioni logico matematiche: AND, OR, e NOT!

NOT

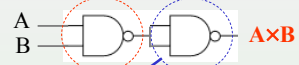


NOT (A AND A)

A	A	A AND A	NOT(A AND A)
0	0	0	1
1	1	1	0

NOT A

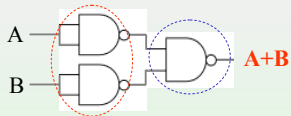
AND



NOT (NOT (A AND B))

A	B	A ⊗ B	$\overline{A \otimes B}$
0	0	1	0
0	1	1	0
1	0	1	0
1	1	0	1

OR



NOT (NOT (A) AND NOT (B))

A	B	$\overline{A}$	$\overline{B}$	A ⊗ B	$\overline{A \otimes B}$
0	0	1	1	1	0
0	1	1	0	0	1
1	0	0	1	0	1
1	1	0	0	0	1

Operazioni aritmetiche con porte logiche

~~OR~~

A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

XOR

A	B	A ⊕ B
0	0	0
0	1	1
1	0	1
1	1	0

Addizione

$$\begin{array}{r} 0+ \\ 0= \\ 0 \end{array} \quad \begin{array}{r} 0+ \\ 1= \\ 1 \end{array} \quad \begin{array}{r} 1+ \\ 0= \\ 1 \end{array} \quad \begin{array}{r} 1+ \\ 1= \\ 10 \end{array}$$

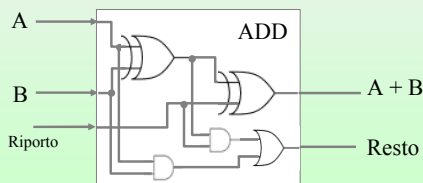
1 1 1 1 1 0 0 1 riporti

25 +	00011001 +
- 7 =	11111001 =
18	100010010

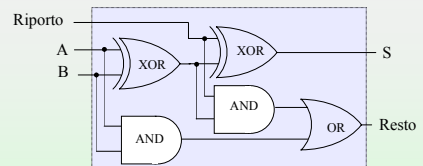
Il sommatore binario

$$\begin{array}{r} 0+ \\ 0= \\ 0 \end{array} \quad \begin{array}{r} 0+ \\ 1= \\ 1 \end{array} \quad \begin{array}{r} 1+ \\ 0= \\ 1 \end{array} \quad \begin{array}{r} 1+ \\ 1= \\ 10 \end{array}$$

$$\begin{array}{r} 1+ \\ 1+ \\ 0= \\ 10 \end{array}$$

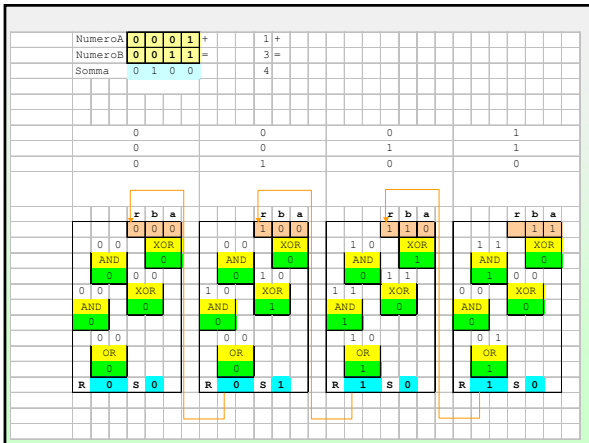
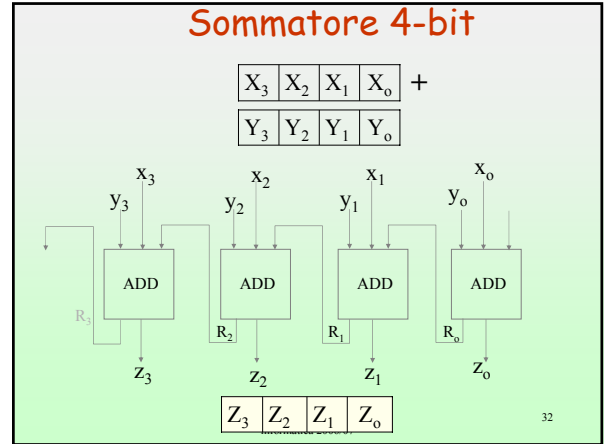
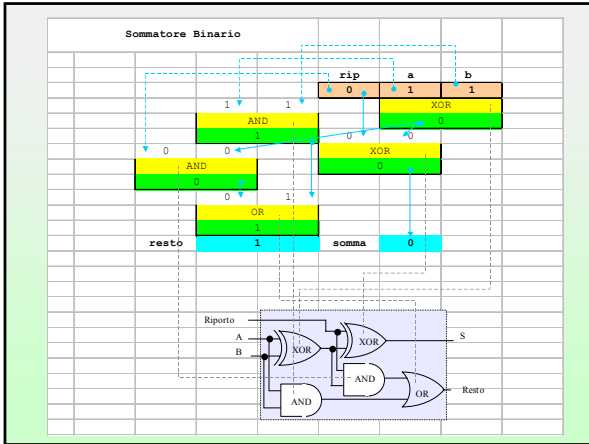


Il sommatore



A	B	Rip.	Rest.	S
0	0	0	0	0
1	0	0	0	1
0	1	0	0	1
1	1	0	1	0

B ₁	B ₂	Rip.	Rest.	S
0	0	1	0	1
1	0	1	1	0
0	1	1	1	0
1	1	1	1	1



Altro materiale

<http://www.math.unipd.it/~frossi/info-base-2003.html>

Iniversità di Padova
Corso di Informatica di Base (sc. biologiche)
Docente: Francesca Rossi
Anno accademico 2003-2004

C. Meneghini - Laboratorio di informatica 2006/07