

# Il computer, II

Giovanni Sartor

Giuseppe Contissa

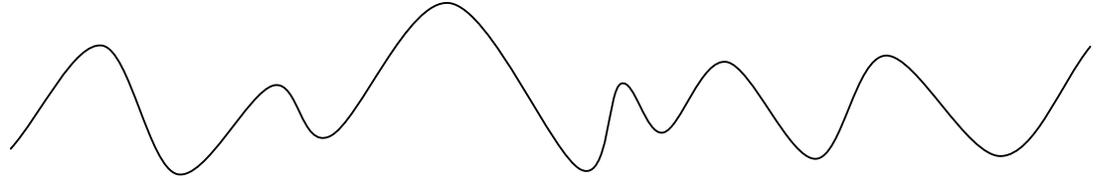
# Le caratteristiche dei calcolatori

I computer sono

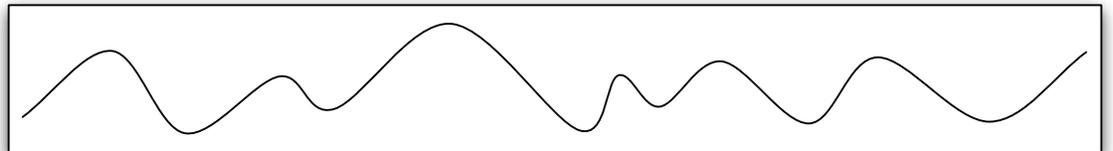
- *digitali*: elaborano informazioni espresse in numeri e, in particolare, in numeri binari.
- *elettronici*: sfruttano il comportamento degli elettroni (le particelle atomiche che ruotano attorno al nucleo dell'atomo), particelle che si spostano sui circuiti elettrici e che determinano la creazione di campi magnetici.
- *programmabili*: operano seguendo le indicazioni di programmi informatici, i quali esprimono, in linguaggi comprensibili da parte del computer, *algoritmi*, cioè precise specificazioni delle elaborazioni da eseguire.
- *universali*: ogni calcolatore è in grado di eseguire, in linea di principio (cioè prescindendo dalle limitazioni del tempo e della memoria di cui dispone) ogni algoritmo.

Rappresentazioni  
analogiche e  
digitali

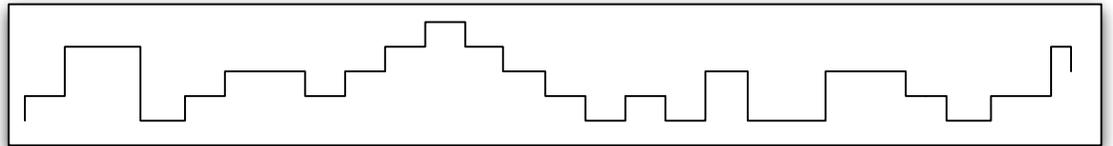
**Il fenomeno da rappresentare, le onde sonore**



**Rappresentazione analogica**



**Rappresentazione discreta**



**Rappresentazione digitale (in numeri decimali)**

0,1, 3, 3, 0, 1, 2, 2, 1, 2, 3, 4, 3, 2, ...

**Rappresentazione digitale (in numeri binari)**

000,001,011,000,001,010,010,001,010,011,100,011,010 ...

# I vantaggi del digitale

- La rappresentazione digitale offre due vantaggi importanti rispetto alla rappresentazione analogica:
  - E' riproducibile con assoluta precisione (la riproduzione è assolutamente identica all'originale);
- Non si danneggia col passare del tempo;
- E' direttamente elaborabile dal computer.

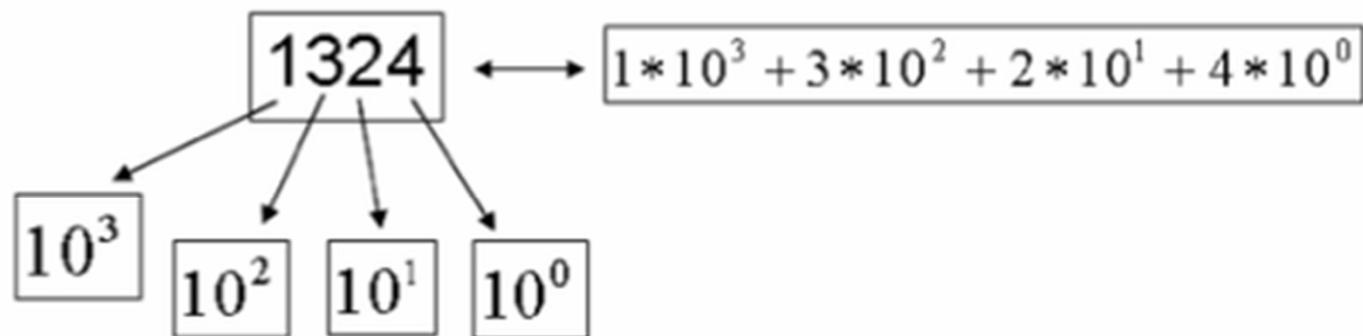
# Il sistema decimale e il sistema binario

- Sono entrambi posizionali: il valore di ogni cifra dipende dalla sua posizione
  - nel sistema decimale la cifra  $x$  nella posizione  $n$ , a partire dalla fine (e contando da 0), vale  $x * 10^n$
  - nel sistema binario la cifra  $x$  nella posizione  $n$ , a partire dalla fine (e contando da 0), vale  $x * 2^n$

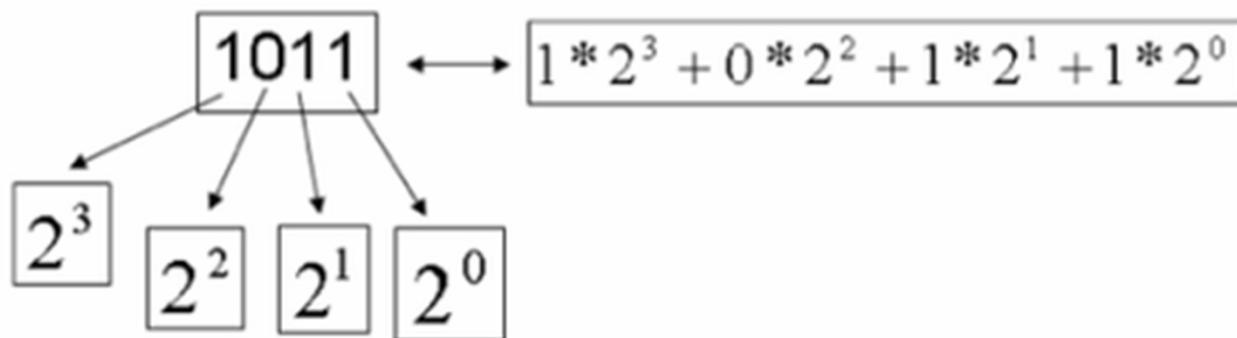
# Binary system: Leibniz's medallion



## Sistema decimale



## Sistema binario



# Il sistema binario e i computer

I computer usano il sistema binario per registrare ogni informazione (sia dati che programmi) in forma digitale

Si adotta il sistema binario (anziché quello digitale) perché è più facile sviluppare componenti hardware capaci di assumere 2 stati alternativi: on / off ( 1/0)

**Bit** = 1 cifra binaria(0 or 1)

**Byte** = 8 bits

1 kilobyte = 1000 byte

1 megabyte = 1 milione di byte

1 gigabyte = 1 miliardo di byte

1 terabyte = 1000 miliardi di byte



# Tabella ASCII

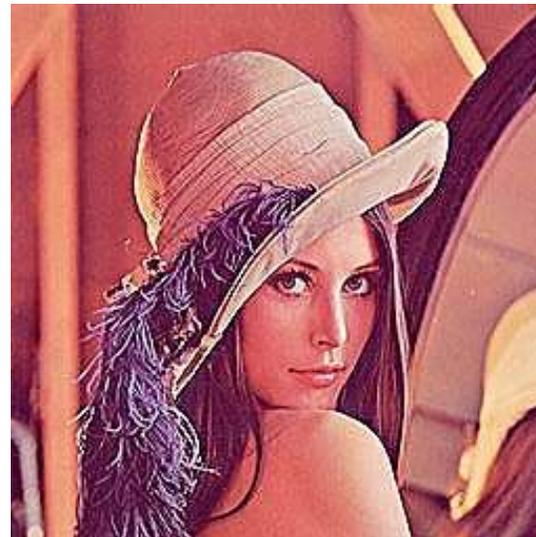
|    |    |                  |    |    |       |    |    |   |     |    |   |
|----|----|------------------|----|----|-------|----|----|---|-----|----|---|
| 0  | 00 | Null             | 32 | 20 | Space | 64 | 40 | @ | 96  | 60 | ` |
| 1  | 01 | Start of heading | 33 | 21 | !     | 65 | 41 | A | 97  | 61 | a |
| 2  | 02 | Start of text    | 34 | 22 | "     | 66 | 42 | B | 98  | 62 | b |
| 3  | 03 | End of text      | 35 | 23 | #     | 67 | 43 | C | 99  | 63 | c |
| 4  | 04 | End of transmit  | 36 | 24 | \$    | 68 | 44 | D | 100 | 64 | d |
| 5  | 05 | Enquiry          | 37 | 25 | %     | 69 | 45 | E | 101 | 65 | e |
| 6  | 06 | Acknowledge      | 38 | 26 | &     | 70 | 46 | F | 102 | 66 | f |
| 7  | 07 | Audible bell     | 39 | 27 | '     | 71 | 47 | G | 103 | 67 | g |
| 8  | 08 | Backspace        | 40 | 28 | (     | 72 | 48 | H | 104 | 68 | h |
| 9  | 09 | Horizontal tab   | 41 | 29 | )     | 73 | 49 | I | 105 | 69 | i |
| 10 | 0A | Line feed        | 42 | 2A | *     | 74 | 4A | J | 106 | 6A | j |
| 11 | 0B | Vertical tab     | 43 | 2B | +     | 75 | 4B | K | 107 | 6B | k |
| 12 | 0C | Form feed        | 44 | 2C | ,     | 76 | 4C | L | 108 | 6C | l |
| 13 | 0D | Carriage return  | 45 | 2D | -     | 77 | 4D | M | 109 | 6D | m |
| 14 | 0E | Shift out        | 46 | 2E | .     | 78 | 4E | N | 110 | 6E | n |
| 15 | 0F | Shift in         | 47 | 2F | /     | 79 | 4F | O | 111 | 6F | o |
| 16 | 10 | Data link escape | 48 | 30 | 0     | 80 | 50 | P | 112 | 70 | p |
| 17 | 11 | Device control 1 | 49 | 31 | 1     | 81 | 51 | Q | 113 | 71 | q |
| 18 | 12 | Device control 2 | 50 | 32 | 2     | 82 | 52 | R | 114 | 72 | r |
| 19 | 13 | Device control 3 | 51 | 33 | 3     | 83 | 53 | S | 115 | 73 | s |
| 20 | 14 | Device control 4 | 52 | 34 | 4     | 84 | 54 | T | 116 | 74 | t |
| 21 | 15 | Neg. acknowledge | 53 | 35 | 5     | 85 | 55 | U | 117 | 75 | u |
| 22 | 16 | Synchronous idle | 54 | 36 | 6     | 86 | 56 | V | 118 | 76 | v |
| 23 | 17 | End trans. block | 55 | 37 | 7     | 87 | 57 | W | 119 | 77 | w |
| 24 | 18 | Cancel           | 56 | 38 | 8     | 88 | 58 | X | 120 | 78 | x |
| 25 | 19 | End of medium    | 57 | 39 | 9     | 89 | 59 | Y | 121 | 79 | y |
| 26 | 1A | Substitution     | 58 | 3A | :     | 90 | 5A | Z | 122 | 7A | z |
| 27 | 1B | Escape           | 59 | 3B | ;     | 91 | 5B | [ | 123 | 7B | { |
| 28 | 1C | File separator   | 60 | 3C | <     | 92 | 5C | \ | 124 | 7C |   |

# Bitmap and Pixmap

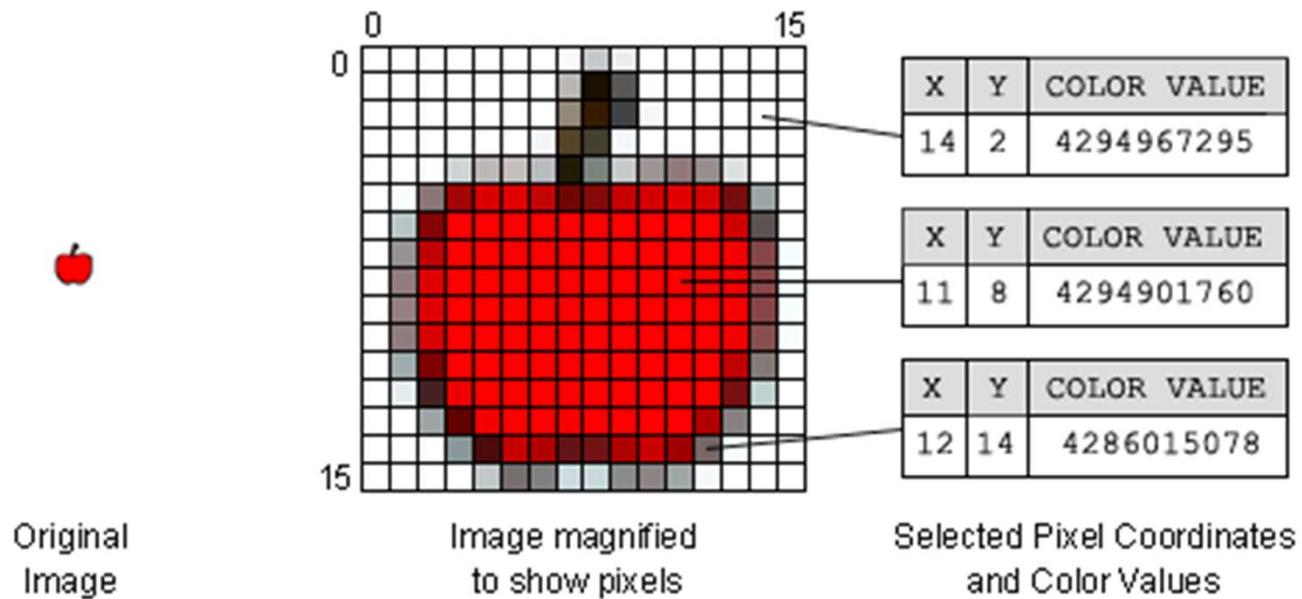
Binary image



Digital colours



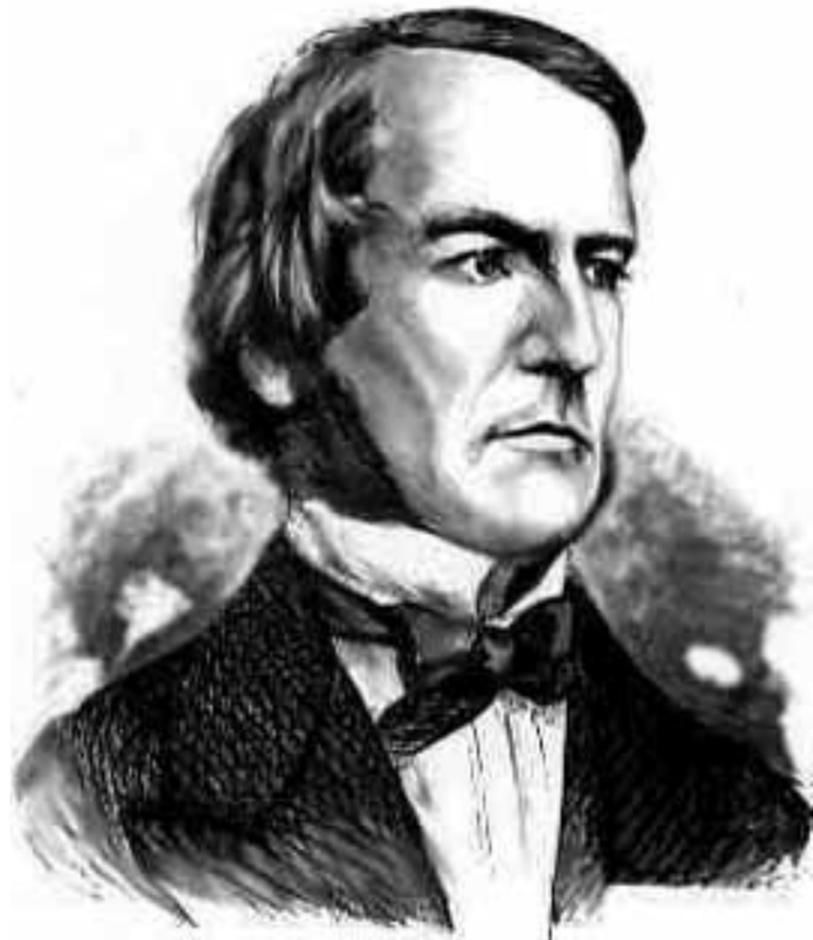
# Bitmap



# George Boole

[1815-1864)

Matematico inglese,  
inventore dell'algebra di  
Boole e della logica  
proposizionale moderna



# Sistema binario e logica

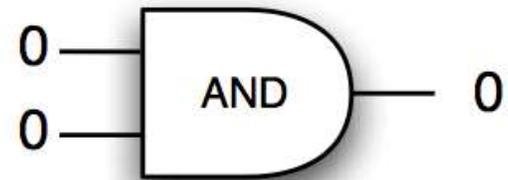
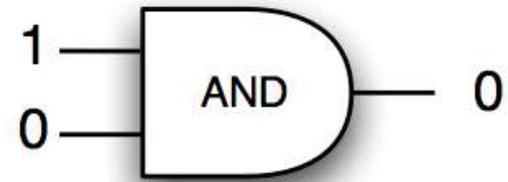
- Si può stabilire un parallelismo tra sistema binario e logica proposizionale:
  - Nel sistema binario , ogni cifra può assumere il valore 1 o il valore 0
  - Nella logica proposizionale, ogni proposizione può assumere il valore “vero” o il valore “falso”
- Nel 1936 Claude Shannon sviluppò dei circuiti elettronici (basati su porte logiche) che operavano secondo la logica

# Il connettivo and

La proposizione  
“**A and B**” è  
vera se e solo  
se le  
proposizioni A  
e B sono  
entrambe vere

| <i>A</i> | <i>B</i> | <i>A and B</i> |
|----------|----------|----------------|
| 1        | 1        | 1              |
| 1        | 0        | 0              |
| 0        | 1        | 0              |
| 0        | 0        | 0              |

# La porta logica AND



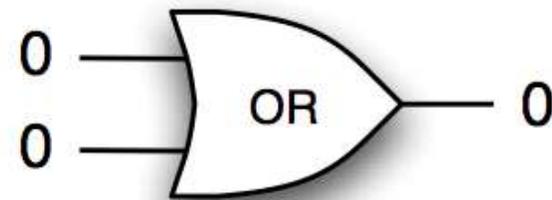
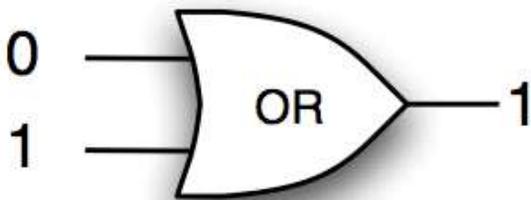
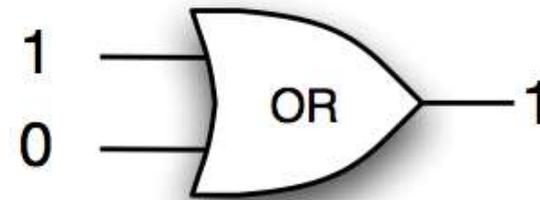
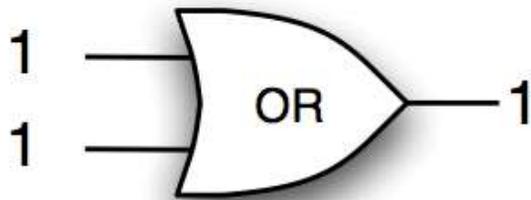
## Il connettivo or (v)

La proposizione “**A or B**” è vera se almeno una delle proposizioni A e B è vera

(disgiunzione debole, come “vel” latino)

| <i>A</i> | <i>B</i> | <i>A or B</i> |
|----------|----------|---------------|
| 1        | 1        | 1             |
| 1        | 0        | 1             |
| 0        | 1        | 1             |
| 0        | 0        | 0             |

# La porta logica OR

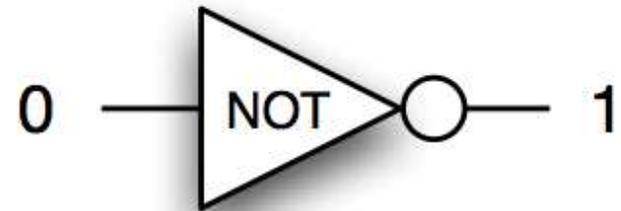
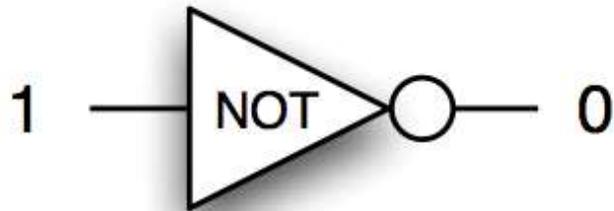


# Il connettivo not

La proposizione “**not A**” è vera se e solo A è falsa

| $A$ | <b>not</b> | $A$ |
|-----|------------|-----|
| 1   | 0          |     |
| 0   | 1          |     |

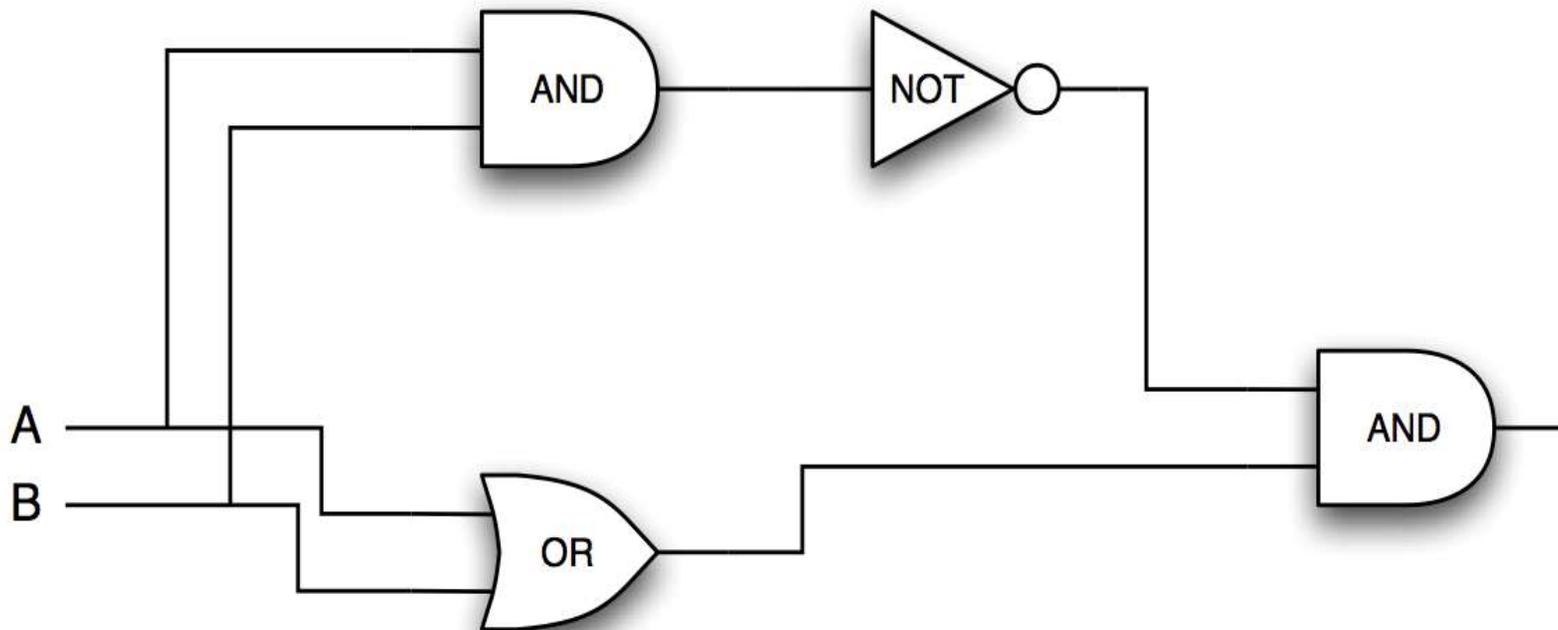
Porta logica NOT.



## Example: XOR

Disgiunzione forte (AUT latino): A XOR B è vera se e solo se A oppure B è vera (ma non entrambe)

**(A OR B) AND (NOT (A AND B))**



Grazie per l'attenzione  
Consultate il sito per materiali e comunicazioni!  
[giovanni.sartor@unibo.it](mailto:giovanni.sartor@unibo.it)