Informatica

Dr. Gianluca Cincotti

Fondamenti

Informatica: Informazione + Automazione

- Si riferisce ai processi e alle tecnologie che rendono possibile l'immagazzinamento e l'elaborazione dell'informazione con un intervento marginale dell'uomo.

Evoluzione della Teoria

- Gli antenati del moderno Computer:
  - Macchina analitica di Babbage (1830)
  - Macchina universale di Turing (Anni '30)
    - Nozione di commutabilità
  - Macchina di von Neumann (anni '40)

La macchina computer

- In generale, un computer:
  - Esegue operazioni logiche e aritmetiche,
  - Ha una memoria per conservare i dati
    - La memoria è il logo fisico in cui vengono immagazzinate le informazioni

- Un programma contiene istruzioni, cioè comandi relativi alle operazioni che l'utente vuole far eseguire alla macchina.

Hardware vs. Software

- L'hardware denota le parti della struttura fisica del computer, costituita di norma da componenti elettronici che svolgono specifiche funzioni nel trattamento dell'informazione.

- Il software denota l'insieme dei programmi che mettono in azione le varie componenti dell'hardware, affinché venga realizzato quanto voluto dall'utente.

Applicazioni nel campo…

- Economico e commerciale (internet: trading on line, shopping on line…);
- Industriale;
- Didattico e della formazione professionale;
- Spettacolo e Arte;
- Ingegneristico;
- Matematico e della scienze;
- Lavorativo e del tempo libero.

La codifica delle informazioni

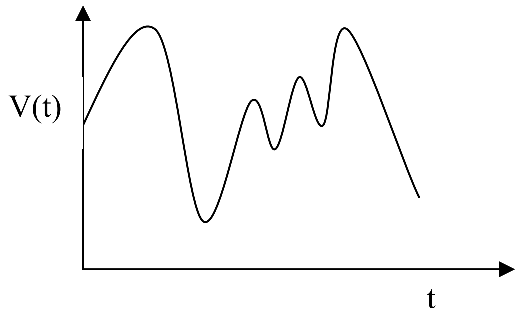
I segnali per comunicare

- Analogico;
- Digitale.

Gli esseri umani ed i computer utilizzano differenti tipi di simboli e segnali per comunicare.

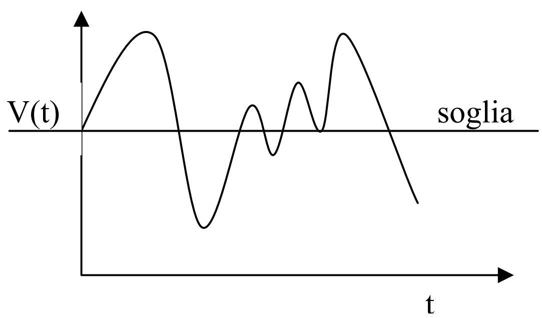
Informazione analogica

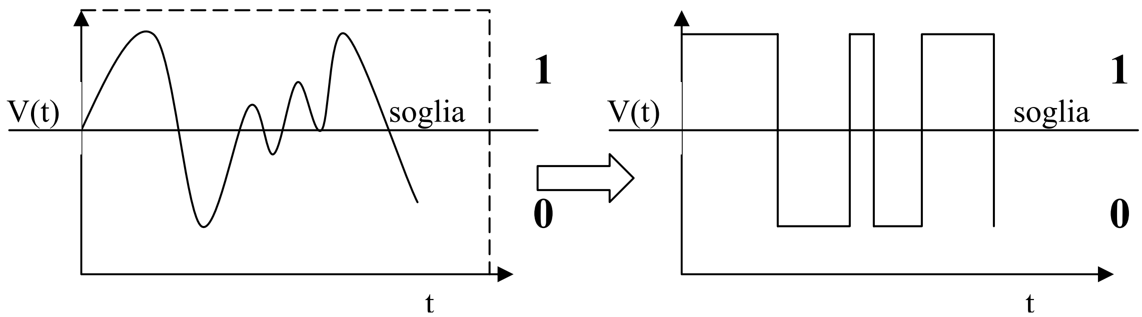
La voce umana e la trasmissione dei programmi di radio e televisione sono comunicazioni di tipo analogico basate su segnali continui di natura generalmente meccanica o tutt'al più elettrica.
Affinché si abbia informazione, occorrono segnali o simboli e un supporto che li contiene.



Informazione digitale

La trasformazione dei segnali nei computer ed in genere nei circuiti elettronici avviene in modo digitale poiché le grandezze sono rappresentate da stati discreti (discontinui).
Nei circuiti di memoria di un computer lo 0 rappresenta un segnale a basso voltaggio che spegne gli interruttori (transistors), mentre l'1 rappresenta un segnale ad alto voltaggio che li accende.



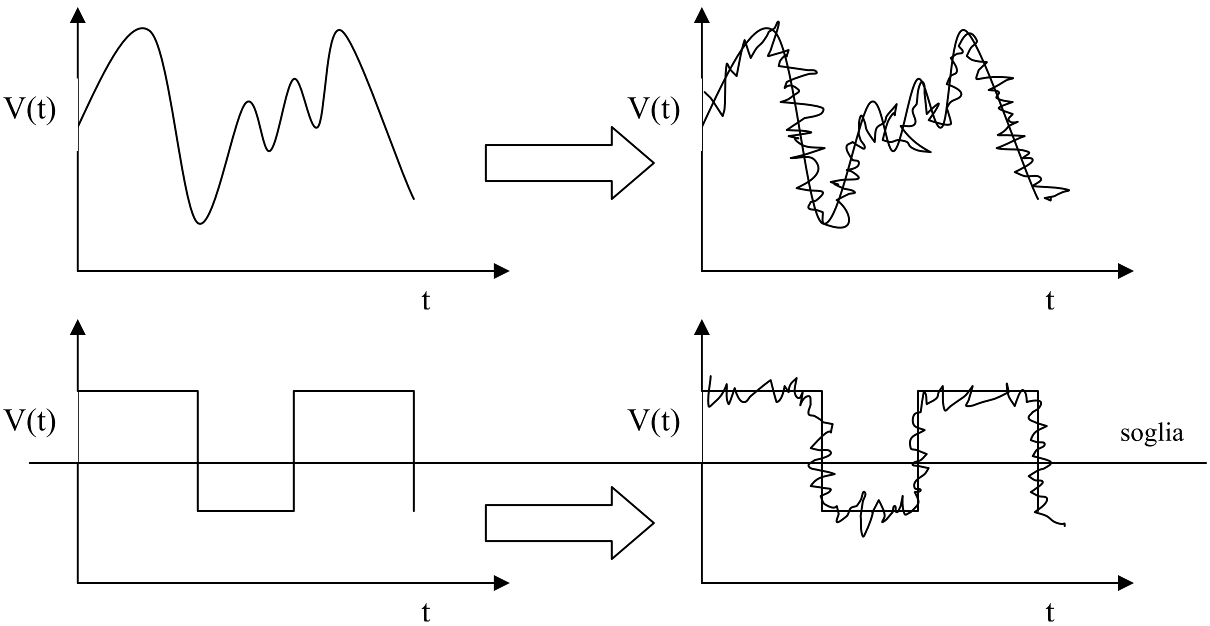


Digitalizzazione dei segnali

- I segnali elettrici continui (analogici) vengono convertiti in segnali digitali.
- La conversione comporta un certo grado di approssimazione.

Precisione dei segnali

- I segnali digitali sono meno affetti da disturbi di trasmissioni.
- La minore sensibilità al rumore consente di replicare perfettamente il segnale.



La rappresentazione delle informazioni

- Tutte le informazioni sono rappresentate in forma binaria o digitale utilizzando due soli simboli: 0 e 1.
- Con una cifra binaria si possono rappresentare solo due tipi di informazioni.
- Le ragioni di questa scelta sono prevalentemente di tipo tecnologico:
    - Due possibili stati di polarizzazione di una sostanza magnetizzabile;
    - Passaggio/non passaggio di corrente attraverso un conduttore;
    - Passaggio/non passaggio della luce attraverso la fibra ottica.

Il bit

- Unità fisica di informazione che vale 0 oppure 1
    - Il nome proviene da Binary Digit.

- Si utilizzano i multipli del bit.

Kilo	Kb	2^{10}	~ un migliaio	1024
Mega	Mb	2^{20}	~ un milione	1024 x 1024
Giga	Gb	2^{30}	~ un miliardo	1Mb x 1024
Tera	Tb	2^{40}	~ mille miliardi	1Gb x 1024

Codifica binaria

- Per poter rappresentare un numero maggiore di informazioni è necessario utilizzare sequenze di bit.
  - Utilizzando due bit, si possono rappresentare quattro informazioni diverse:

    00 01 10 11
- Il processo che fa corrispondere ad un'informazione una configurazione di bit prende il nome di "Codifica dell'informazione".

Sequenze di bit

Numero di bit nella sequenza	**Informazioni rappresentabili**
2	4
3	8
4	16
5	32
6	64
7	128
8	256

I caratteri utilizzati nella comunicazione scritta

- 52 lettere alfabetiche maiuscole e minuscole;
- 10 cifre (0, 1, 2, 3, 4, 5, 6, 7, 8, 9);
- Segni di punteggiatura (!,;.:"….);
- Segni matematici (+ - * / =…);
- Caratteri nazionali (ç, ò, à, ù, ì, è, é);
- Altri segni grafici (@, #, €, ®, ©, ™)
- In totale 220 caratteri circa.

Codice

- Si pone quindi la necessità di codificare in numeri binari almeno 220 caratteri.
- La sequenza di bit necessaria a rappresentare 220 simboli deve essere composta da 8 bit e prende il nome di "codice".

Il Byte

- Un gruppo di 8 bit viene denominato Byte
  - Corrisponde ad un carattere;
  - E' l'unità di misura della capacità di memoria.

Kilo	KB	2^{10}	~ un migliaio
Mega	MB	2^{20}	~ un milione
Giga	GB	2^{30}	~ un miliardo
Tera	TB	2^{40}	~ mille miliardi

- Si utilizzano i multipli del Byte

Rappresentazione di dati alfabetici

- Un codice numerico per ogni carattere.
- Codifiche standard:
  - ASCII, 8 bit per carattere, rappresenta 256 caratteri (ASCII: American Standard Code for Information Interchange);
  - UNICODE, 16 bit per carattere (ASCII e caratteri etnici)
- Codifiche proprietarie:
  - MS Windows, 2 Byte per carattere (Simile a UNICODE)

Sequenze di caratteri ASCII

Dividendo la sequenza in gruppi di Byte è possibile risalire ad ogni singolo carattere.

Numeri e codice ASCII

- Con il codice ASCII, è possibile rappresentare i numeri come sequenza di caratteri;
- Ad esempio il numero 234 sarà rappresentato come:

00110010	00110011	00110100
2	3	4

Con questo tipo di rappresentazione non è possibile effettuare operazioni aritmetiche.

Il sistema di numerazione posizionale decimale

Nella numerazione posizionale, ogni cifra del numero assume un valore in funzione della posizione. Ad esempio, prendendo il numero 221 e trascrivendolo in notazione compatta, verrà:

$$(2 \times 100) + (2 \times 10) + (1 \times 1)$$

o meglio

$$(2 \times 10^2) + (2 \times 10^1) + (1 \times 10^0)$$

con la notazione esplicita

Notazione posizionale

- Ogni numero si esprime come la somma dei prodotti di ciascuna cifra per la base elevata all'esponente che rappresenta la posizione della cifra.
- La notazione posizionale può essere usata con qualunque base, creando così differenti sistemi di numerazione.
  - Per ogni base di numerazione si utilizza un numero di cifre uguale alla base

- In informatica si utilizza prevalentemente la numerazione:
  - Binaria;
  - Ottale;

- Esadecimale.

➢ Il sistema di numerazione romano non è posizionale
- Ad esempio, XIII è diverso da CXII

Sistema di numerazione decimale

La numerazione decimale utilizza una notazione posizionale basata su 10 cifre, da 0 a 9, e sulle potenze di 10.

- Il numero 234 può essere esplicitamente rappresentato come:

$$(2 \times 10^2) + (3 \times 10^1) + (4 \times 10^0)$$

Il sistema di numerazione binario

Il sistema di numerazione binario utilizza una notazione posizionale basata su 2 cifre, 0 e 1, e sulle potenze di 2.

- Il numero 1001 può essere esplicitamente rappresentato come:

$$1001_2 = (1 \times 2^3) + (0 \times 2^2) + (0 \times 2^1) + (1 \times 2^0) = 9_{10}$$

Il sistema di numerazione ottale

Il sistema di numerazione ottale utilizza una notazione posizionale basata su 8 cifre (da 0 a 7) e sulle potenze di 8.

- Ad esempio prendiamo il numero 534:

$$534_8 = (5 \times 8^2) + (3 \times 8^1) + (4 \times 8^0) = 348_{10}$$

Il sistema di numerazione esadecimale

La numerazione esadecimale utilizza una notazione posizionale basata su 16 cifre (da 0 a 9 e poi A, B,C, D, E, F) e su potenze di 16.

Dove:

A = 10
B = 11
C = 12
D = 13
E = 14
F = 15

- Il numero B7FC viene esplicitamente rappresentato come:

$$B7FC_{16} = [(11) \times 16^3] + (7 \times 16^2) + [(15) \times 16^1] + [(12) \times 16^0] = 47100_{10}$$

Conversione da base n a base 10

➢ Per convertire un numero ad una base n qualsiasi, occorre trovare tutti i resti delle successive divisioni per la base n:
- Come esempio si vuole trovare il valore binario del numero 210; per fare ciò bisognerà dividere 210 per la base 2:

$$
\begin{array}{r|lcc}
210 & 2 & \text{resto} & 0 \\
105 & 2 & \text{“} & 1 \\
52 & 2 & \text{“} & 0 \\
26 & 2 & \text{“} & 0 \\
13 & 2 & \text{“} & 1 \\
6 & 2 & \text{“} & 0 \\
3 & 2 & \text{“} & 1 \\
1 & 2 & \text{“} & 1
\end{array}
$$

- Leggendo la sequenza dei resti dal basso verso l’alto, si ottiene il numero: 11010010_2

Verifica di correttezza

Per una verifica di correttezza basta riconvertire il risultato alla base 10.

$$11010010_2 = (1 \times 2^7)+(1 \times 2^6)+(0 \times 2^5)+(1 \times 2^4)+(0 \times 2^3)+(0 \times 2^2)+(1 \times 2^1)+(0 \times 2^0)=210_{10}$$

Costruzione dei numeri binari

Per costruire la successione dei numeri binari si può seguire il seguente schema:

$$
\begin{array}{cccccc}
0 & 0 & 0 & 0 & = & 0 \\
0 & 0 & 0 & 1 & = & 1 \\
0 & 0 & 1 & 0 & = & 2 \\
0 & 0 & 1 & 1 & = & 3 \\
0 & 1 & 0 & 0 & = & 4 \\
0 & 1 & 1 & 0 & = & 5 \\
0 & 1 & 1 & 1 & = & 6
\end{array}
$$

PER NON PERDERE TEMPO ALL’ESAME!

$$
\begin{array}{cccccc}
2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \\
\downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\
32 & 16 & 8 & 4 & 2 & 1 \\
 & & & 1 & 0 & 1
\end{array}
$$

$$
101_2= \begin{array}{ccc}
1 & 0 & 1 \\
\downarrow & \downarrow & \downarrow \\
2^2 & 0 & 2^0 \\
\downarrow & & \downarrow \\
4 & + & 1
\end{array} =5_{10}
\qquad
43_{10}= \begin{array}{ccccccc}
0 & 1 & 0 & 1 & 0 & 1 & 1 \\
2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0
\end{array}
$$

Operazioni binarie

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

```
  0 0 1 1 0 0 1 1+
  0 0 1 1 1 0 0 0=
 ------------------
  0 1 1 0 1 0 1 1
```

0 + 1 = 0
0 + 0 = 0
1 + 1 = 0 con riporto di 1

```
     1 1 0 1x          1 0 0 1 1x
         1 1=                  1 0=
  -------------      --------------
     1 1 0 1           0 0 0 0 0
   1 1 0 1 ---       1 0 0 1 1 ---
  -------------      --------------
 1 0 0 1 1 1         1 0 0 1 1 0
```

Rappresentazione dei numeri

- All'interno dei computer, a causa dei vincoli tecnologici, per codificare qualsiasi tipo di numero, si utilizzano sempre un numero fisso di cifre binarie.
- Su tutti i computer si utilizzano:

  16 bit (2 byte)
  32 bit (4 byte)

  - In alcuni casi si può arrivare a 64 bit (8 byte) o più, a seconda del tipo di processore.

- Tutti i numeri vengono distinti in tre categorie:

  - Interi senza segno (interi positivi)
  - Interi con segno (interi positivi e negativi)
  - Reali (numeri positivi e negativi con virgola)

- Ogni categoria viene rappresentata in modo differente

Rappresentazione con numero fisso di cifre

- Per comprendere il meccanismo alla base della rappresentazione con un numero fisso di cifre, partiamo da un esempio:

  - Qual è il numero più grande rappresentabile con 4 cifre?

    Base 10 = 9999

    Base 2 = 1111 = 15_{10}

    Base 8 = 7777 = 4095_{10}

    Base 16 = FFFF = 65535_{10}

- In generale si avrà:

  b^n-1

  Base 10 = 9999 10^4-1

  Base 2 = 1111 = 2^4-1

Base 8 = 7777 = 8^4-1

Base 16 = FFFF = 16^4-1

Rappresentazione dei numeri interi senza segno

Per calcolare il valore massimo ammesso, occorre applicare la regola

$$2^n-1$$

dove n vale 16 o 32.

- Nella rappresentazione a 16 bit, i possibili valori saranno compresi tra
  0 e 65.535
- Nella rappresentazione a 32 bit, i possibili valori saranno compresi tra
  0 e 4.294.967.295

Numeri interi con segno

- Per rappresentare i numeri con il loro segno (interi positivi e negativi) esistono due possibili modi. Il primo è il seguente:
  - Dati n bit, un bit si riserva al segno e gli altri n-1 sono destinati al numero.
  - Ad esempio, considerando 8 bit e ponendo il primo bit a sinistra 0 per il + e 1 per il meno avremo:

$$0\ 0\ 0\ 0\quad 0\ 1\ 0\ 1_2 = +\ 5_{10}$$

$$1\ 0\ 0\ 0\quad 0\ 1\ 0\ 1_2 = -\ 5_{10}$$

- Questo tipo di rappresentazione prende il nome di "Rappresentazione in modulo e segno"
- Anche se semplice, possiede però un grosso difetto: esistono due zeri.

1 0 0 0	0 0 1 1	=	-3
1 0 0 0	0 0 1 0	=	-2
1 0 0 0	0 0 0 1	=	-1
1 0 0 0	0 0 0 0	=	-0
0 0 0 0	0 0 0 0	=	+0
0 0 0 0	0 0 0 1	=	+1
0 0 0 0	0 0 1 0	=	+2
0 0 0 0	0 0 1 1	=	+3

- Utilizzando n bit e riservandone uno al segno, l'applicazione della formula precedente porterà:

da $-\mathbf{(2^{n-1}-1)}$ a $\mathbf{2^{n-1}-1}$
(dove n vale 16 o 32)

- Seguendo il ragionamento precedente, i possibili valori nel caso di 16 bit saranno quindi compresi fra

-32.767 e +32.767

- Nel caso di 32 bit si avrà:

-2.147.483.647 e +2.147.483.647

- Il secondo modo per rappresentare i numeri con il loro segno (interi positivi e negativi) è quello del *complemento a due*:

    - Dato un numero composto da n bit, la rappresentazione in complemento a due si ottiene complimentando ogni cifra, cioè invertendo gli 1 in 0 e gli 0 in 1, poi sommando 1 al risultato ottenuto.

L'overflow

- Per questioni tecnologiche tutti i computer, senza alcuna eccezione, trattano i numeri sempre con un numero fisso di cifre binarie (ad esempio 16 32 o più).

- Quando l'elaboratore esegue un'operazione il cui risultato eccede il numero di cifre permesso, la computazione si arresta immediatamente e viene segnalato l'errore di overflow.
- Ad esempio se la rappresentazione è a 32 bit senza segno e si vuole eseguire la seguente operazione:

3.000.000.000+
2.000.000.000=
****
Errore di overflow!
La computazione si arresta immediatamente

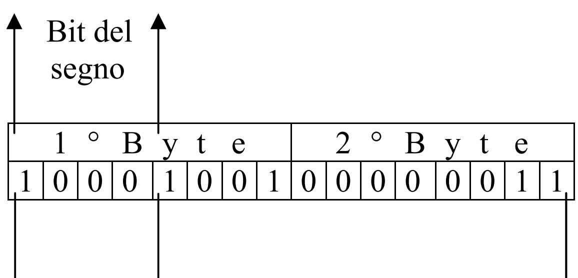
Rappresentazione dei numeri con virgola

- I numeri con la virgola vengono rappresentati mediante la notazione scientifica o in virgola mobile.

- Anche i numeri interi possono sempre essere espressi come numeri con virgola attraverso la notazione scientifica.

Numero	Notazione Scientifica	Parte decimale (mantissa)	Exp
250	$0,25 \times 10^{3}$	25	3
-83,76	$-0,8376 \times 10^{2}$	-8376	2
0,05	$0,5 \times 10^{-1}$	5	-1
55.640.350	$0,5564035 \times 10^{8}$	5564035	8

- Con la notazione scientifica ogni numero viene memorizzato solo come mantissa (parte decimale senza lo 0 e la virgola) e con l'esponente (senza la base): si risparmia spazio.



esponente mantissa

La logica booleana

Proposizioni logiche

- Una proposizione logica è una proposizione grammaticale del tipo:
  "Al soggetto X spetta il predicato Y"
  per cui abbia senso dire che è vera o non vera (ovvero falsa).

- Una proposizione logica esprime un giudizio (predica) su uno o più argomento.

- Sono preposizioni logiche:
  - Tommaso è fratello di Elena;
  - (5+7) è un numero pari;
  - Maria è brava in italiano.

- Non sono preposizioni logiche:
  - I sentimenti (Es.: Ti voglio bene!)
  - Le domande (Es.: Che ora è?)
  - I comandi (Es.: Passami il sale!)
  - Le previsioni (Es.: Domani pioverà)
  - Le definizioni (Es.: Chiamasi cerchio…)

- Teoremi e postulati sono preposizioni logiche.

Principi fondamentali della logica

- Principio di *non contraddizione:*
  - Una proposizione logica non può essere contemporaneamente vera e falsa.

- Principio del *terzo escluso*:
  - Una proposizione logica è vera "o" è falsa.

- Combinando insieme i due principi si ottiene che, per una proposizione logica A, si possono presentare due soli casi:
  1. A è vera;
  2. A è falsa.

  L'uno esclude l'altro.

Preposizioni logiche composte

- Se A e B sono due proposizioni logiche, allora le seguenti sono ancora proposizioni logiche (composte):
  1. $\neg A$ (si legge not A)
  2. $A \land B$ (si legge A and B)

3. $A \vee B$ (si legge A or B)

- I simboli $\neg$, $\wedge$ e $\vee$ si chiamano connettivi preposizionali

Tavole di verità

- Il valore di verità di una proposizione logica composta è determinato dal connettivo in oggetto e dal valore di verità delle proposizioni logiche componenti mediante le tavole di verità.

A	¬A
V	F
F	V

A	B	A∧B	A∨B
V	V	V	V
V	F	F	V
F	V	F	V
F	F	F	F

- Per i principi della logica, per una proposizione logica si possono avere solo 2 casi: vero (V) o falso (F). Per tanto, per due proposizioni logiche ne abbiamo 4, per 3 ne abbiamo 8, ecc.
- Esempio: "Un numero intero può essere multiplo di 2 o multiplo di 3" (può anche esserlo di tutti e due).

Il connettivo ⊕

- Il connettivo ⊕ denota la "o" esclusiva (xor). [Es. un numero intero è pari "o" dispari.]
- Attenzione: l'interpretazione del simbolo ⊕, è diversa rispetto al connettivo ∨.

A	B	A⊕B
V	V	F
V	F	V
F	V	V
F	F	F

Per ricordare!!!

Connettivo ∧ : Se ALMENO UNA delle proposizioni è FALSA,
il risultato dell'espressione sarà FALSO!

Connettivo ∨ : Se ALMENO UNA delle proposizioni è VERA,
il risultato dell'espressione sarà VERO!

Connettivo ⊕ : 2 proposizioni UGUALI hanno risultato FALSO!
2 proposizioni DIVERSE hanno risultato VERO!

Espressioni logiche

- $(\neg A \vee B) \vee (A \wedge C)$
    - Se poniamo: A = Vero, B = Vero, C = Falso,
    il valore di verità sarà Falso.

- $(A \wedge \neg B) \vee (B \wedge \neg C)$

- Se poniamo: A = Vero, B = Vero, C = Falso,

il valore di verità sarà Vero.

Tautologia e contraddizione

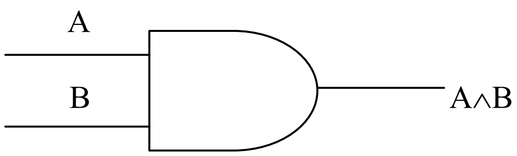
- Una proposizione logica sempre vera si dice Tautologia.
- Una proposizione logica sempre falsa si dice contraddizione.

Osservazioni sulla negazione

- Consideriamo le due proposizioni logiche:
  - A: “Mario è bravo in italiano”
  - B: “Lucia è brava in italiano”
- Quindi $(A \wedge B)$ significa che Mario e Lucia sono bravi in italiano.
- La negazione di $(A \wedge B)$ è:
  - $\neg(A \wedge B)$. Cioè, non è vero che entrambi sono bravi in italiano.
  - Ovvero, almeno uno dei due non è bravo in italiano, dunque $\neg(A \wedge B) = (\neg A \wedge \neg B)$

Porte logiche (“And”)

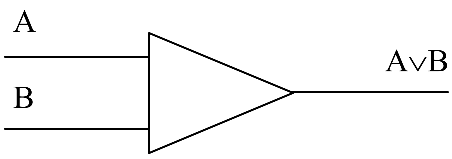
- Definiamo *porta congiuntiva* un dispositivo che ha due (o più) linee di entrata (input) ed una linea in uscita (output) che realizza la congiunzione logica $\wedge$ “and”



- Fa sì che la corrente esca ad alta tensione (1) se e solo se entra a bassa tensione (0) e viceversa.

Porte logiche (“Or”)

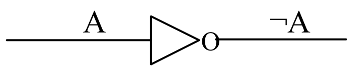
- Definiamo *porta disgiuntiva* un dispositivo che ha due (o più) linee di entrata (input) ed una linea di uscita (output) che realizza la congiunzione logica $\vee$ (or).



- Fa sì che la corrente esca ad alta tensione (1) se e solo se entra ad alta tensione (1) da almeno una delle linee di input.

Porte logiche (“Not”)

- Definiamo *invertitore* un dispositivo che ha una sola linea di entrata (input) ed una linea d’uscita (output) che realizza la negazione logica $\neg$ “not”.



- Fa sì che la corrente esca ad alta tensione (1) se e solo se entra a bassa tensione (0) e viceversa.

Circuiti logici

Si definisce circuito logico un circuito costruito a partire da porte logiche (congiuntive, disgiuntive, invertitori) per realizzare espressioni logiche, cioè per simulare il valore di verità di un'espressione logica.

- $(A \wedge B) \vee (A \wedge \neg C) \vee (B \wedge C)$