# Fondamenti di Informatica

*Rappresentazione dell'informazione*

Corso di Laurea in Ingegneria dell'Informazione

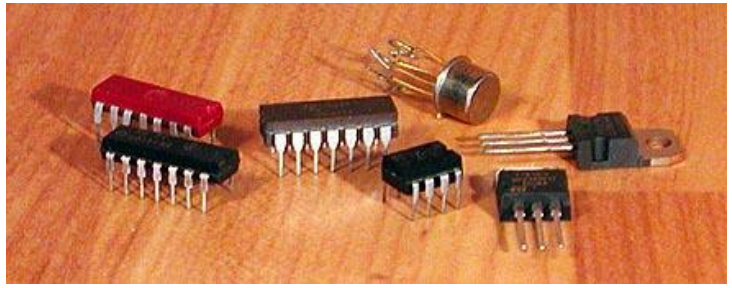
A.A. 2015-2016

Prof. Ing. Domenico Rosaci

Università degli Studi
**Mediterranea**
di Reggio Calabria

# Calcolatori Elettronici

- Con l'invenzione del tubo a vuoto (1904), del transistor (1947) e, infine, dei circuiti integrati (1969), l'evoluzione dei computer divenne inarrestabile

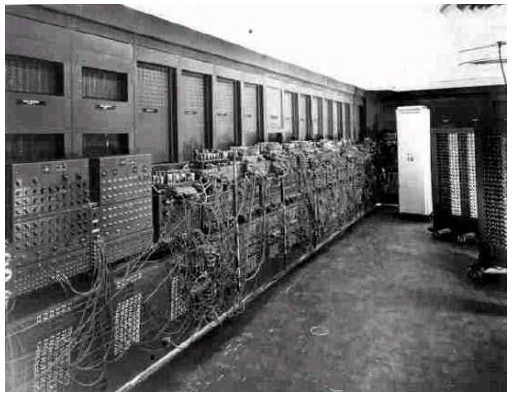



- Attualmente la potenza di calcolo degli elaboratori decuplica ogni 5–6 anni (…ma non può durare, almeno con le tecnologie in uso)

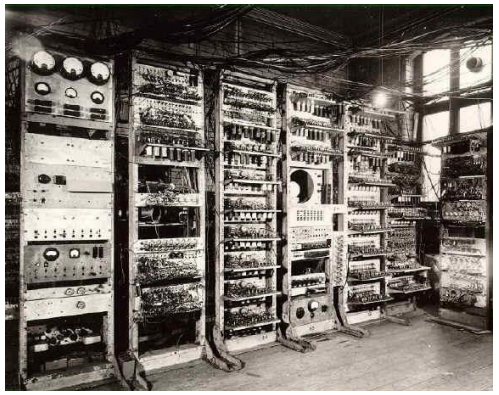
# Primi Calcolatori Elettronici

- La costruzione dei primi calcolatori risale all'inizio degli anni '40, grazie alla tecnologia elettronica; i primi esemplari venivano programmati mediante connessioni elettriche e commutatori (ENIAC, Mark I)
- Il nome di Von Neumann è legato invece ai primi calcolatori a programma memorizzato realizzati alla fine degli anni '40 (EDSAC, Whirlwind, IAS, UNIVAC)
  - Per la prima volta, vige il principio di ***unitarietà di rappresentazione di dati e istruzioni***, che vengono codificati, all'interno dell'elaboratore, in maniera indistinguibile
- La diffusione dei calcolatori a livello mondiale è avvenuta nei decenni '60 e '70

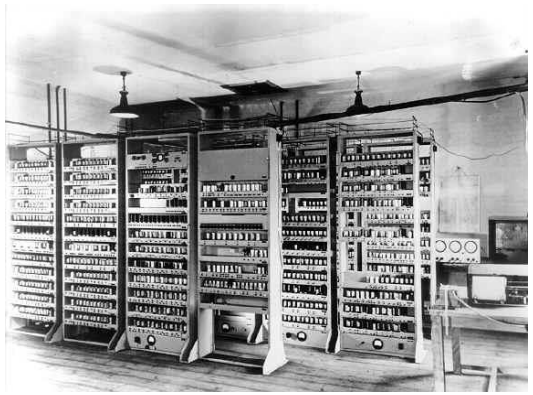
# Primi Calcolatori Elettronici



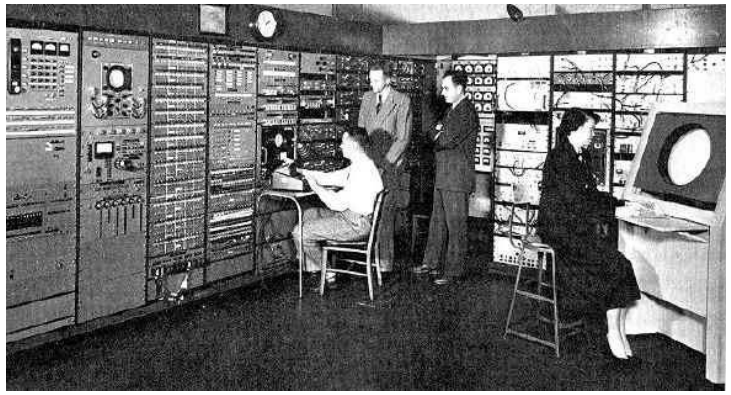
ENIAC (1946)



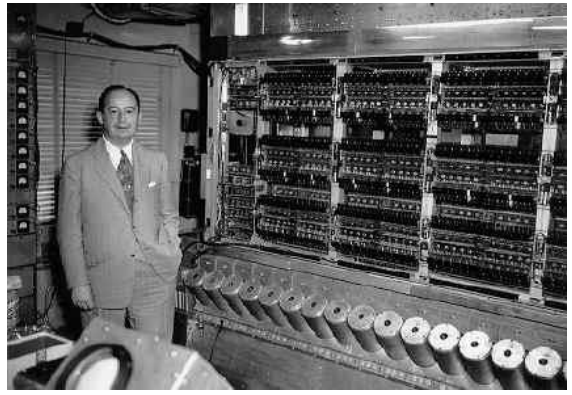
Mark I (1948)



EDSAC (1949)



Whirlwind (1949)



IAS (1952)

UNIVAC (1952)

# Il Personal Computer

- Tuttavia, l'esplosione dell'informatica come fenomeno di massa è datata 1981, anno in cui l'IBM introdusse un tipo particolare di elaboratore: il Personal Computer (PC)
- La particolarità dei PC consisteva nell'essere "assemblati" con componenti facilmente reperibili sul mercato (e quindi a basso costo)
  - Possibilità per qualsiasi casa produttrice di costruire "cloni"
- Attualmente i PC, o meglio il loro componente fondamentale — il microprocessore — è utilizzato in tutti i settori applicativi (non solo per elaborare dati):
  - Telefoni cellulari
  - Ricevitori satellitari digitali
  - Bancomat e carte di credito
  - Lavatrici e forni a microonde
  - Computer di bordo e ABS
  - ...

# Evoluzione

Illiac (1955)

CDC 6600 (1963)

Cray 1 (1976)

Cray X1 (2002)

PC IBM (1981)

Portatile e Palmare (oggi)

# L'architettura di Von Neumann

- La capacità dell'elaboratore di eseguire successioni di operazioni in modo automatico è determinata dalla presenza di un dispositivo di **memoria**
  - Nella memoria sono registrati i **dati** e...
  - ...la descrizione delle operazioni da eseguire su di essi (nell'ordine secondo cui devono essere eseguite): il **programma**, la "ricetta" usata dall'elaboratore per svolgere il suo compito
- Il programma viene interpretato dall'**unità di controllo**

⇓

**Modello di Von Neumann**

# La macchina universale

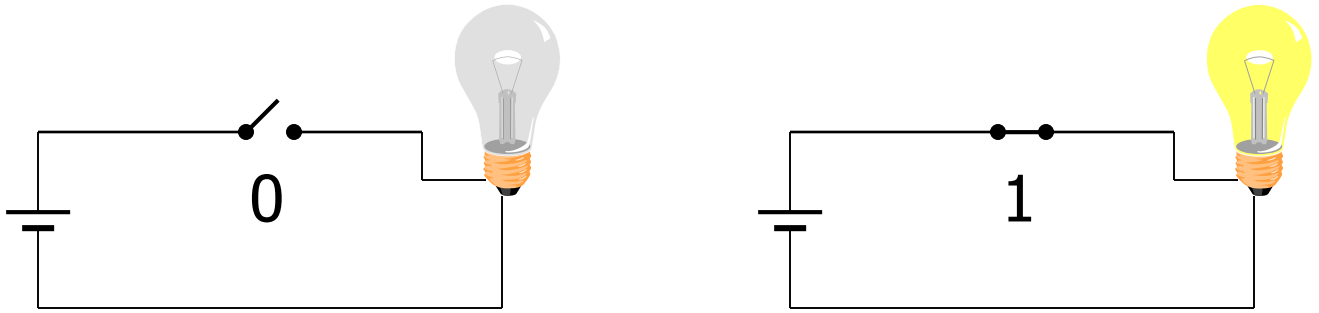
- **Programma**: sequenza di operazioni atte a predisporre l'elaboratore alla soluzione di una determinata classe di problemi
  - Il programma è la descrizione di un **algoritmo** in una forma comprensibile all'elaboratore

- **Algoritmo**: sequenza finita di istruzioni attraverso le quali un operatore umano è capace di risolvere ogni problema di una data classe; non è direttamente eseguibile dall'elaboratore

- L'elaboratore è una **macchina universale**: cambiando il programma residente in memoria, è in grado di risolvere problemi di natura diversa (una classe di problemi per ogni programma)

# Algebra Booleana

- Contempla due costanti 0 e 1 (falso e vero)
- Corrispondono a due stati che si escludono a vicenda
- Possono descrivere lo stato di apertura o chiusura di un generico contatto o di un circuito a più contatti

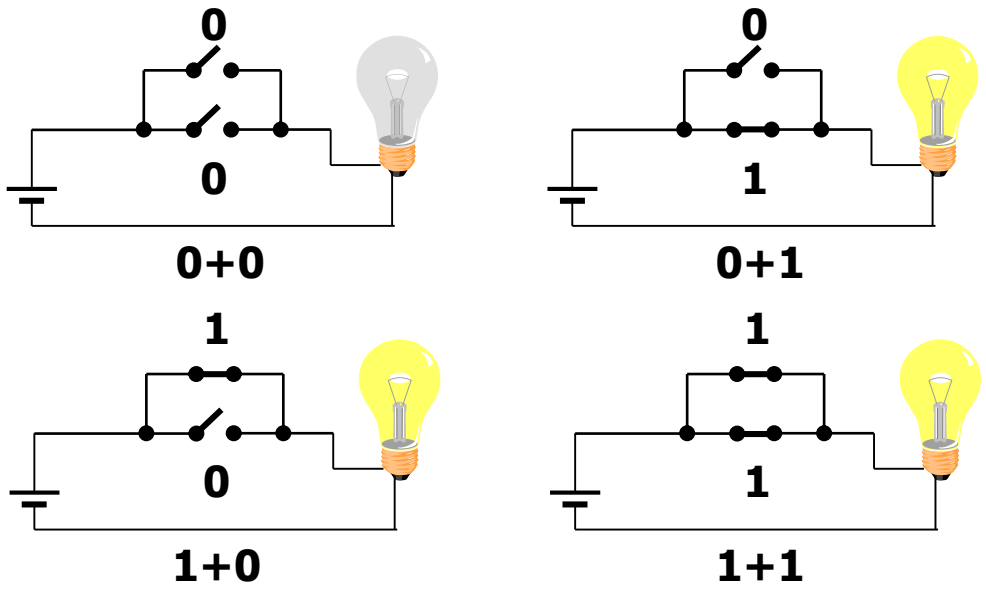



- Si definiscono delle operazioni fra i valori booleani: AND, OR, NOT sono gli operatori fondamentali

# L'operazione di OR

- Si definisce l'operazione di somma logica (OR): il valore della somma logica è il simbolo 1 se il valore di almeno uno degli operandi è il simbolo 1

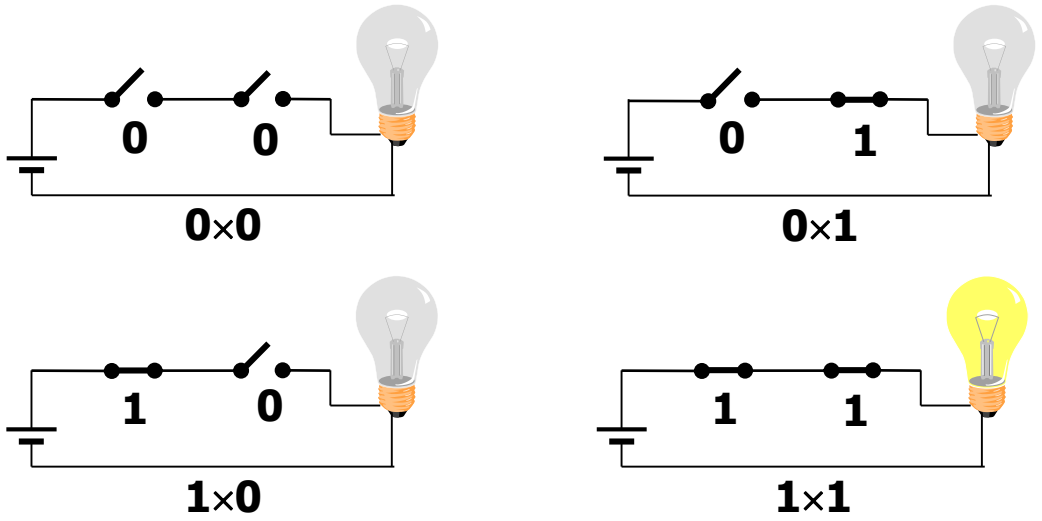
```
0+0 = 0
0+1 = 1
1+0 = 1
1+1 = 1
```



# L'operazione di AND

- Si definisce l'operazione di prodotto logico (AND): il valore del prodotto logico è il simbolo 1 se il valore di tutti gli operandi è il simbolo 1

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



# La negazione NOT

- Si definisce l'operatore di negazione (NOT): l'operatore inverte il valore della costante su cui opera

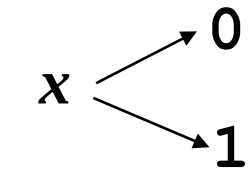
$$\overline{0} = 1$$
$$\overline{1} = 0$$

- Dalla definizione…

$$\overline{\overline{0}} = 0$$
$$\overline{\overline{1}} = 1$$

# Variabili binarie

- Una variabile binaria indipendente può assumere uno dei due valori **0** e **1**




- Date *n* variabili binarie indipendenti, la loro somma logica (OR) è

$$x_1 + x_2 + \ldots + x_n = \begin{cases} \mathbf{1} \text{ se almeno una } x_i \text{ vale } \mathbf{1} \\ \mathbf{0} \text{ se } x_1 = x_2 = \ldots = x_n = \mathbf{0} \end{cases}$$

# AND e NOT con variabili binarie

- Date *n* variabili binarie indipendenti, il loro prodotto logico (AND) è

$$x_1 \times x_2 \times \ldots \times x_n = \begin{cases} 0 \text{ se almeno una } x_i \text{ vale } 0 \\ 1 \text{ se } x_1 = x_2 = \ldots = x_n = 1 \end{cases}$$

- La negazione di una variabile *x* è

$$\overline{x} = 0 \text{ se } x = 1$$
$$\overline{x} = 1 \text{ se } x = 0$$

# Configurazioni delle variabili

- Date $n$ variabili binarie indipendenti $x_1$, $x_2$,…, $x_n$, queste possono assumere $2^n$ configurazioni distinte

Ad esempio per $n$=3 si hanno 8 configurazioni

$x_1x_2x_3$ 

```
000 001 010 011
100 101 110 111
```

# Funzioni logiche

- Una variabile $y$ è una funzione delle $n$ variabili indipendenti $x_1$, $x_2$,…, $x_n$, se esiste un criterio che fa corrispondere in modo univoco ad ognuna delle $2^n$ configurazioni delle $x_i$ un valore di $y$

$$y = \mathrm{F}(x_1, x_2, \ldots, x_n)$$

- Una rappresentazione esplicita di una funzione è la tabella di verità, in cui si elencano tutte le possibili combinazioni di $x_1$, $x_2$, …, $x_n$, con associato il valore di $y$

$y = x_1 + x_2$

| $x_1$ | $x_2$ | $y$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

# Un esempio: lo XOR

- La funzione XOR verifica la disuguaglianza di due variabili

| $x_1$ | $x_2$ | $XOR$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

- L'espressione come somma di prodotti è quindi...

$$XOR = x_1\bar{x}_2 + \bar{x}_1x_2$$

# Un po' di preistoria…

- I primi esempi di utilizzo di sistemi di numerazione risalgono al neolitico, ovvero a circa 50.000 anni fa
- In epoca preistorica, le più utilizzate furono le basi 2, 5, 10, 20, 12, e 60
  - Mentre le basi 2, 5 10 e 20 sono suggerite dalla fisiologia umana, 12 e 60 sembrano suggerite da scopi utilitaristici: 12 è divisibile per 1, 2, 3, 4, 6 e 12 mentre 60 per 1, 2, 3, 4, 5, 6, 12, 15, 20, 30 e 60
  - Da notare che il 7 non compare mai nelle basi di numerazione e, in effetti, ebbe significati particolari, anche religiosi, presso i popoli antichi
  - La base 12 è ancora utilizzata in certe misure di tempo, 60 nella misurazione di angoli e tempo

# ...E di storia

- Tra le prime testimonianze certe dell'utilizzo di concetti numerici avanzati vi sono le tavole numeriche babilonesi, elenchi di numeri utilizzati per calcoli astronomici e di agrimensura, risalenti al X secolo a.C.
- Tuttavia nelle culture dell'antica Mesopotamia esistevano tabelle per le addizioni e le sottrazioni già durante il regno di Sargon I, intorno al 2350 a.C.
- Il documento più significativo dell'antico Egitto è il papiro di Ahmes o Ahmose, dal nome dello scriba che lo compose nel 1650 a.C.
- Lo stesso Ahmes sostiene inoltre che il suo materiale è tratto da un documento anteriore, e fa risalire l'originale ad Imhotep, medico e architetto del faraone Djoser della III dinastia, e quindi al 2650 a.C. circa

# Sistemi di numerazione posizionali

- Sistemi di numerazione **posizionali**:

  La **base** del sistema di numerazione
  Le **cifre** del sistema di numerazione

  Il numero è scritto specificando le cifre in ordine ed il suo valore dipende dalla **posizione relativa** delle cifre

  **Esempio**: Il sistema decimale (Base 10)

Cifre : 0 1 2 3 4 5 6 7 8 9

$$5641 = 5\cdot 10^3 + 6\cdot 10^2 + 4\cdot 10^1 + 1\cdot 10^0$$

Posizione: 3 2 1 0

# Sistemi in base B

- La base definisce il numero di cifre diverse nel sistema di numerazione
- La cifra di minor valore è sempre lo 0; le altre sono, nell'ordine, 1,2,…,B–1; se B>10 occorre introdurre B–10 simboli in aggiunta alle cifre decimali

Un numero **intero** N si rappresenta con la scrittura $(c_n c_{n-1} \ldots c_2 c_1 c_0)_B$

$$N = c_n B^n + c_{n-1} B^{n-1} + \ldots + c_2 B^2 + c_1 B^1 + c_0 B^0$$

$\mathbf{c_n}$ è la **cifra più significativa**, $\mathbf{c_0}$ la **meno significativa**

Un numero **frazionario** N' si rappresenta come $(0,c_1 c_2 \ldots c_n)_B$

$$N' = c_1 B^{-1} + c_2 B^{-2} + \ldots + c_n B^{-n}$$

# Numeri interi senza segno

- Con n cifre in base B si rappresentano tutti i numeri interi positivi da 0 a $B^n-1$ ($B^n$ numeri distinti)

**Esempio**: base 10

2 cifre: da 0 a $10^2-1 = 99$

00
01
02
....
98
99

$10^2 = 100$ valori

**Esempio**: base 2

2 cifre: da 0 a $2^2-1 = 3$

00
01
10
11

$2^2 = 4$ valori

# Il sistema binario (B=2)

- **La base 2 è la più piccola per un sistema di numerazione**

Cifre: 0 1 – **bit** (**binary digit**)

**Esempi**:

Forma polinomia

$$(101101)_2 = 1\times2^5 + 0\times2^4 + 1\times2^3 + 1\times2^2 + 0\times2^1 + 1\times2^0 =$$
$$32 + 0 + 8 + 4 + 0 + 1 = (45)_{10}$$

$$(0,0101)_2 = 0\cdot2^{-1} + 1\cdot2^{-2} + 0\cdot2^{-3} + 1\cdot2^{-4} =$$
$$0 + 0,25 + 0 + 0,0625 = (0,3125)_{10}$$

$$(11,101)_2 = 1\cdot2^1 + 1\cdot2^0 + 1\cdot2^{-1} + 0\cdot2^{-2} + 1\cdot2^{-3} =$$
$$2 + 1 + 0,5 + 0 + 0,125 = (3,625)_{10}$$

# Dal bit al byte

- Un **byte** è un insieme di 8 bit (un numero binario a 8 cifre)

$$b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0$$

- Con un byte si rappresentano i numeri interi fra 0 e $2^8 - 1 = 255$

00000000
00000001
00000010
00000011
……
11111110
11111111

$2^8 = 256$ valori distinti

- È l'elemento base con cui si rappresentano i dati nei calcolatori
- Si utilizzano sempre dimensioni multiple (di potenze del 2) del byte: 2 byte (16 bit), 4 byte (32 bit), 8 byte (64 bit)…

# Dal byte al kilobyte

- Potenze del 2

$2^4 = 16$
$2^8 = 256$
$2^{16} = 65536$

- Cosa sono KB (Kilobyte), MB (Megabyte), GB (Gigabyte)?

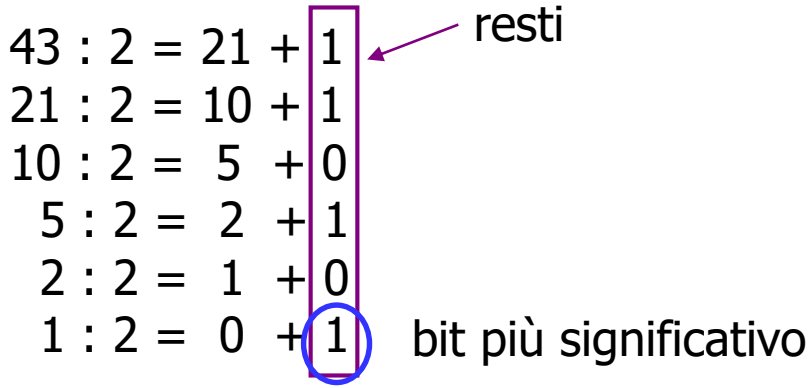
$2^{10} = 1024$ (K=Kilo)
$2^{20} = 1048576$ (M=Mega)
$2^{30} = 1073741824$ (G=Giga)

1 KB = $2^{10}$ byte = 1024 byte
1 MB = $2^{20}$ byte = 1048576 byte
1 GB = $2^{30}$ byte = 1073741824 byte
1 TB = $2^{40}$ byte = 1099511627776 byte (Terabyte)

# Da decimale a binario
## Numeri interi

- Si divide ripetutamente il numero **intero** decimale per 2 fino ad ottenere un quoziente nullo; le cifre del numero binario sono i resti delle divisioni; la cifra più significativa è l'ultimo resto

**Esempio**: convertire in binario $(43)_{10}$




$(43)_{10} = (101011)_2$

# Da binario a decimale

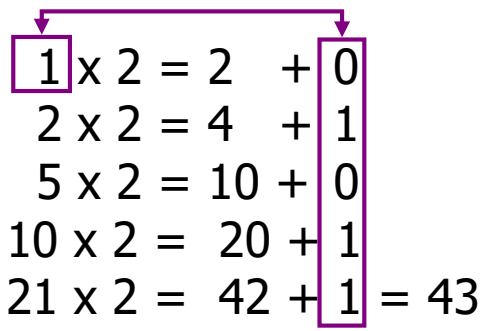
- Oltre all'espansione esplicita in potenze del 2 – **forma polinomia**…

$$(101011)_2 = 1\times2^5 + 0\times2^4 + 1\times2^3 + 0\times2^2 + 1\times2^1 + 1\times2^0 = (43)_{10}$$

- …si può operare nel modo seguente: si raddoppia il bit più significativo e si aggiunge al secondo bit; si raddoppia la somma e si aggiunge al terzo bit… si continua fino al bit meno significativo

**Esempio**: convertire in decimale $(101011)_2$

bit più significativo




$$(101011)_2 = (43)_{10}$$

# Sistema esadecimale

- **La base 16 è molto usata in campo informatico**

Cifre: 0 1 2 3 4 5 6 7 8 9 A B C D E F

La corrispondenza in decimale delle cifre oltre il 9 è

$A = (10)_{10}$ $D = (13)_{10}$
$B = (11)_{10}$ $E = (14)_{10}$
$C = (12)_{10}$ $F = (15)_{10}$

Esempio:

$$(3A2F)_{16} = 3\times16^3 + 10\times16^2 + 2\times16^1 + 15\times16^0 =$$
$$3\times4096 + 10\times256 + 2\times16 + 15 = (14895)_{10}$$

# Esempi – 1

- In qualsiasi base, l'essere il sistema di numerazione ***posizionale*** impone che combinazioni diverse di cifre uguali rappresentino numeri diversi; ad esempio:
  - $(319)_{10} \neq (193)_{10}$
  - $(152)_6 \neq (512)_6$, infatti...

    $(152)_6 = 1\times6^2+5\times6^1+2\times6^0 = 36+30+2 = (68)_{10}$

    $(512)_6 = 5\times6^2+1\times6^1+2\times6^0 = 180+6+2 = (188)_{10}$

- Numeri che hanno identica rappresentazione, in basi diverse, hanno valori diversi:
  - $(234)_{10} \neq (234)_8$ , infatti...

    $(234)_8 = 2\times8^2 + 3\times8^1 + 4\times8^0 = 2\times64 + 3\times8 + 4 = 128+24+4 = (156)_{10}$

Osservazione: più piccola è la base, minore è il valore del numero rappresentato dalla stessa sequenza di cifre

# Numeri interi positivi

- I numeri interi positivi sono rappresentati all'interno dell'elaboratore utilizzando un multiplo del byte (generalmente 4/8 byte)
- Se l'intero si rappresenta con un numero di cifre minore, vengono aggiunti zeri nelle cifre più significative

**Esempio**: 12 viene rappresentato in un byte come…

00001100

# Numeri con segno

- Per rappresentare numeri con segno, occorre utilizzare un bit per definire il segno del numero
- Si possono usare tre tecniche di codifica
  - Modulo e segno
  - Complemento a 2
  - Complemento a 1

# Modulo e segno

- Il bit più significativo rappresenta il segno: 0 per i numeri positivi, 1 per quelli negativi
- Esiste uno zero positivo (00…0) e uno zero negativo (10…0)
- Se si utilizzano n bit si rappresentano tutti i numeri compresi fra $-(2^{n-1}-1)$ e $+2^{n-1}-1$

**Esempio**: con 4 bit si rappresentano i numeri fra $-7$ ($-(2^3-1)$) e $+7$ ($2^3-1$)

| positivi | | negativi | |
|---|---|---|---|
| 0000 | +0 | 1000 | −0 |
| 0001 | +1 | 1001 | −1 |
| 0010 | +2 | 1010 | −2 |
| 0011 | +3 | 1011 | −3 |
| 0100 | +4 | 1100 | −4 |
| 0101 | +5 | 1101 | −5 |
| 0110 | +6 | 1110 | −6 |
| 0111 | +7 | 1111 | −7 |

# Complemento a 2

- Il complemento a 2 di un numero binario $(N)_2$ a n cifre è il numero

$$2^n - (N)_2 = 10\underbrace{\ldots\ldots}_{n}0 - (N)_2$$

- Il complemento a 2 si calcola…
  - Effettuando il complemento a 1 del numero di partenza (negazione di ogni cifra): si trasforma ogni 0 in 1 e ogni 1 in 0
  - Aggiungendo 1 al numero ottenuto

```
01010111
↓↓↓↓↓↓↓↓ ← complemento a 1
10101000

          + 1
----------
10101001
```

# Interi in complemento a 2

- I **numeri positivi** sono rappresentati (come) in modulo e segno
- I **numeri negativi** sono rappresentati in complemento a 2 $\Rightarrow$ la cifra più significativa ha sempre valore 1
- Lo zero è rappresentato come numero positivo (con una sequenza di n zeri)
- Il campo dei numeri rappresentabili varia da $-2^{n-1}$ a $+2^{n-1}-1$

**Esempio**: numeri a 4 cifre

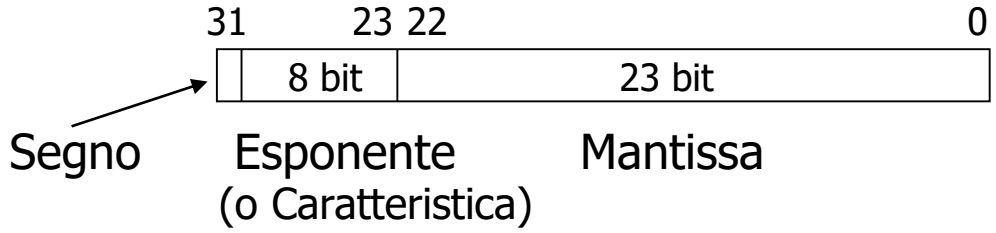
| | | | |
|---|---|---|---|
| 0000 | +0 | 1000 | −8 |
| 0001 | +1 | 1001 | −7 |
| 0010 | +2 | 1010 | −6 |
| 0011 | +3 | 1011 | −5 |
| 0100 | +4 | 1100 | −4 |
| 0101 | +5 | 1101 | −3 |
| 0110 | +6 | 1110 | −2 |
| 0111 | +7 | 1111 | −1 |

Nota: 0111 +7
1000 −8

# Numeri in virgola mobile

- La rappresentazione dei numeri in virgola mobile è in relazione con la **notazione scientifica** (es. $1.2 \times 10^2 = 120$)
- La IEEE ha previsto uno standard per la rappresentazione in virgola mobile
  - **singola precisione** (32 bit = 4 byte)
  - **doppia precisione** (64 bit = 8 byte)
  - **quadrupla precisione** (128 bit = 16 byte)

Singola precisione




Il valore è

$(-1)^S \ 1.M \times 2^{E-127}$ se $E \neq 0$

$(-1)^S \ 0.M \times 2^{-126}$ se $E = 0$

**Eccesso**: vale $2^{t-1}-1$, se t è il numero di cifre riservate alla caratteristica → rappresentazione "interna" dell'esponente sempre positiva

# L’aritmetica degli elaboratori – 1

- L’aritmetica “interna” degli elaboratori differisce notevolmente dall’aritmetica classica
- Sebbene le stesse operazioni possano essere realizzate secondo modalità diverse su elaboratori diversi, si riscontrano alcune caratteristiche comuni:
  - Rappresentazione binaria dei numeri
  - Rango finito dei numeri rappresentabili
  - Precisione finita dei numeri
  - Operazioni espresse in termini di operazioni più semplici

# L’aritmetica degli elaboratori – 2

- Rango finito dei numeri rappresentabili
  - Qualunque sia la codifica utilizzata, esistono sempre il più grande ed il più piccolo numero rappresentabile
  - I limiti inferiore e superiore del rango di rappresentazione dipendono sia dal tipo di codifica, sia dal numero di bit utilizzati
  - Se il risultato di un’operazione non appartiene al rango dei numeri rappresentabili, si dice che si è verificato un overflow (un underflow, più precisamente, se il risultato è più piccolo del più piccolo numero rappresentabile)

# L’aritmetica degli elaboratori – 3

- Precisione finita dei numeri
  - La precisione della rappresentazione di un numero frazionario è una misura di quanto essa corrisponda al numero che deve essere rappresentato
  - Negli elaboratori, i numeri frazionari sono rappresentati in virgola mobile (**floating–point**), utilizzando un numero finito di bit
  - È plausibile che un numero reale non ammetta una rappresentazione finita, quindi dovrà essere codificato in maniera approssimata
  - Negli elaboratori si rappresentano soltanto numeri razionali (fino ad una data precisione)

# L'aritmetica degli elaboratori – 4

- Operazioni espresse in termini di operazioni più semplici
  - La maggior parte degli elaboratori non possiede circuiti in grado di eseguire direttamente tutte le operazioni:
    - La sottrazione si realizza per mezzo di una complementazione e di un'addizione
    - La moltiplicazione si realizza per mezzo di una successione di addizioni e di **shift** (traslazioni)
    - La divisione si realizza per mezzo di una successione di shift e sottrazioni
  - Le operazioni più semplici sono eseguite direttamente da appositi circuiti (in **hardware**); le operazioni più complesse sono realizzate mediante l'esecuzione di successioni di operazioni più semplici, sotto il controllo di programmi appositamente realizzati, e generalmente memorizzati permanentemente (in **firmware**)

# Codifica dei caratteri alfabetici – 1

- Oltre ai numeri, molte applicazioni informatiche elaborano caratteri (simboli)
- Gli elaboratori elettronici trattano numeri
- Si codificano i caratteri e i simboli per mezzo di numeri
- Per poter scambiare dati (testi) in modo corretto, occorre definire uno standard di codifica

| | | |
|---|---|---|
| A | ⟶ | 01000001 |
| 3 | ⟶ | 00110011 |
| $ | ⟶ | 00100100 |

# Codifica dei caratteri alfabetici – 2

- Quando si scambiano dati, deve essere noto il tipo di codifica utilizzato
- La codifica deve prevedere le lettere dell'alfabeto, le cifre numeriche, i simboli, la punteggiatura, i caratteri speciali per certe lingue (æ, ã, ë, è,…)
- Lo standard di codifica più diffuso è il codice ASCII, per American Standard Code for Information Interchange

# Codifica ASCII

- Definisce una tabella di corrispondenza fra ciascun carattere e un codice a 7 bit (128 caratteri)
- I caratteri, in genere, sono rappresentati con 1 byte (8 bit); i caratteri con il bit più significativo a 1 (quelli con codice dal 128 al 255) rappresentano un'estensione della codifica
- La tabella comprende sia caratteri di controllo (codici da 0 a 31) che caratteri stampabili
- I caratteri alfabetici/numerici hanno codici ordinati secondo l'ordine alfabetico/numerico

| | | |
|---|---|---|
| 0 48 | A 65 | a 97 |
| 1 49 | B 66 | b 98 |
| ……. | ……. | ……. |
| 8 56 | Y 89 | y 121 |
| 9 57 | Z 90 | z 122 |
| cifre | maiuscole | minuscole |

# Caratteri di controllo ASCII

- I caratteri di controllo (codice da 0 a 31) hanno funzioni speciali
- Si ottengono o con tasti specifici o con una sequenza Ctrl+carattere

| Ctrl | Dec | Hex | Code | Nota |
|---|---|---|---|---|
| ^@ | 0 | 0 | NULL | carattere nullo |
| ^A | 1 | 1 | SOH | partenza blocco |
| ..... | ... | ... | ..... | ..... |
| ^G | 7 | 7 | BEL | beep |
| ^H | 8 | 8 | BS | backspace |
| ^I | 9 | 9 | HT | tabulazione orizzontale |
| ^J | 10 | A | LF | line feed (cambio linea) |
| ^K | 11 | B | VT | tabulazione verticale |
| ^L | 12 | C | FF | form feed (alim. carta) |
| ^M | 13 | D | CR | carriage return (a capo) |
| ..... | ... | ... | ..... | ..... |
| ^Z | 26 | 1A | EOF | fine file |
| ^[ | 27 | 1 B | ESC | escape |
| ..... | ... | ... | ..... | ..... |
| ^_ | 31 | 1F | US | separatore di unità |

# Caratteri ASCII stampabili

| Dec | Hx | Chr | Dec | Hx | Chr | Dec | Hx | Chr | Dec | Hx | Chr | Dec | Hx | Chr | Dec | Hx | Chr |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 32 | 20 | SPACE | 48 | 30 | 0 | 64 | 40 | @ | 80 | 50 | P | 96 | 60 | ` | 112 | 70 | p |
| 33 | 21 | ! | 49 | 31 | 1 | 65 | 41 | A | 81 | 51 | Q | 97 | 61 | a | 113 | 71 | q |
| 34 | 22 | ” | 50 | 32 | 2 | 66 | 42 | B | 82 | 52 | R | 98 | 62 | b | 114 | 72 | r |
| 35 | 23 | # | 51 | 33 | 3 | 67 | 43 | C | 83 | 53 | S | 99 | 63 | c | 115 | 73 | s |
| 36 | 24 | $ | 52 | 34 | 4 | 68 | 44 | D | 84 | 54 | T | 100 | 64 | d | 116 | 74 | t |
| 37 | 25 | % | 53 | 35 | 5 | 69 | 45 | E | 85 | 55 | U | 101 | 65 | e | 117 | 75 | u |
| 38 | 26 | & | 54 | 36 | 6 | 70 | 46 | F | 86 | 56 | V | 102 | 66 | f | 118 | 76 | v |
| 39 | 27 | ’ | 55 | 37 | 7 | 71 | 47 | G | 87 | 57 | W | 103 | 67 | g | 119 | 77 | w |
| 40 | 28 | ( | 56 | 38 | 8 | 72 | 48 | H | 88 | 58 | X | 104 | 68 | h | 120 | 78 | x |
| 41 | 29 | ) | 57 | 39 | 9 | 73 | 49 | I | 89 | 59 | Y | 105 | 69 | i | 121 | 79 | y |
| 42 | 2A | * | 58 | 3A | : | 74 | 4A | J | 90 | 5A | Z | 106 | 6A | j | 122 | 7A | z |
| 43 | 2B | + | 59 | 3B | ; | 75 | 4B | K | 91 | 5B | [ | 107 | 6B | k | 123 | 7B | { |
| 44 | 2C | , | 60 | 3C | < | 76 | 4C | L | 92 | 5C | \ | 108 | 6C | l | 124 | 7C | \| |
| 45 | 2D | - | 61 | 3D | = | 77 | 4D | M | 93 | 5D | ] | 109 | 6D | m | 125 | 7D | } |
| 46 | 2E | . | 62 | 3E | > | 78 | 4E | N | 94 | 5E | ^ | 110 | 6E | n | 126 | 7E | ~ |
| 47 | 2F | / | 63 | 3F | ? | 79 | 4F | O | 95 | 5F | _ | 111 | 6F | o | 127 | 7F | DEL |

**Nota**: il valore numerico di una cifra può essere calcolato come differenza del suo codice ASCII rispetto al codice ASCII della cifra 0 (es. ‘5’–‘0’ = 53–48 = 5)

# Tabella ASCII estesa

- I codici oltre il 127 non sono compresi nello standard originario

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 128 | Ç | 144 | É | 160 | á | 176 | ░ | 193 | ┴ | 209 | ╤ | 225 | ß | 241 | ± |
| 129 | ü | 145 | æ | 161 | í | 177 | ▒ | 194 | ┬ | 210 | ╥ | 226 | Γ | 242 | ≥ |
| 130 | é | 146 | Æ | 162 | ó | 178 | ▓ | 195 | ├ | 211 | ╙ | 227 | π | 243 | ≤ |
| 131 | â | 147 | ô | 163 | ú | 179 | │ | 196 | ─ | 212 | ╘ | 228 | Σ | 244 | ⌠ |
| 132 | ä | 148 | ö | 164 | ñ | 180 | ┤ | 197 | ┼ | 213 | ╒ | 229 | σ | 245 | ⌡ |
| 133 | à | 149 | ò | 165 | Ñ | 181 | ╡ | 198 | ╞ | 214 | ╓ | 230 | µ | 246 | ÷ |
| 134 | å | 150 | û | 166 | ª | 182 | ╢ | 199 | ╟ | 215 | ╫ | 231 | τ | 247 | ≈ |
| 135 | ç | 151 | ù | 167 | º | 183 | ╖ | 200 | ╚ | 216 | ╪ | 232 | Φ | 248 | ° |
| 136 | ê | 152 | _ | 168 | ¿ | 184 | ╕ | 201 | ╔ | 217 | ┘ | 233 | ⊛ | 249 | · |
| 137 | ë | 153 | Ö | 169 | _ | 185 | ╣ | 202 | ╩ | 218 | ┌ | 234 | Ω | 250 | · |
| 138 | è | 154 | Ü | 170 | ¬ | 186 | ║ | 203 | ╦ | 219 | ▉ | 235 | δ | 251 | √ |
| 139 | ï | 156 | £ | 171 | ½ | 187 | ╗ | 204 | ╠ | 220 | ▄ | 236 | ∞ | 252 | _ |
| 140 | î | 157 | ¥ | 172 | ¼ | 188 | ╝ | 205 | ═ | 221 | ▌ | 237 | ϕ | 253 | ² |
| 141 | ì | 158 | _ | 173 | ¡ | 189 | ╜ | 206 | ╬ | 222 | ▐ | 238 | ε | 254 | ■ |
| 142 | Ä | 159 | ƒ | 174 | « | 190 | ╛ | 207 | ╧ | 223 | ▀ | 239 | ∩ | 255 | |
| 143 | Å | 192 | └ | 175 | » | 191 | ┐ | 208 | ╨ | 224 | α | 240 | ≡ | | |