Fondamenti di Informatica

Ingegneria Clinica

Lezione 13/11/2009

Prof. Raffaele Nicolussi

FUB - Fondazione Ugo Bordoni

Via B. Castiglione 59 - 00142 Roma

Docente	**Raffaele Nicolussi**	*rnicolussi@fub.it* *0654803323*
Lezioni Aula 54 (ex aula 4) Via del Castro Laurenziano, 7	**Lunedì, Giovedì, Venerdì**	**12:00 – 13:30**
Esercitazioni Aula 15 Via Tiburtina, 205	**Giovedì**	**14:00 – 16.30**
Ricevimento:	**Per appuntamento**	**in FUB, per email, per telefono**
Sito web:	**http://w3.uniroma1.it/IngClinFondinf**	

Rappresentazioni numeriche (1)

- Rappresentazione ed interpretazione
- Sistema posizionale, binario, ottale ed esadecimale.
- Notazione
- Conversioni fra basi
  - Binaria, ottale, decimale, esadecimale, …
- La base 16 per etichettare la memoria
- Rappresentazione dei numeri interi negativi con il metodo del complemento
- Aritmetica modulare
- Problema dell'overlow nelle operazioni aritmetiche
- Complemento a 2 per cambiare segno
- Numeri frazionari

Rappresentazioni numeriche (2)

- Rappresentazione dei numeri frazionari in altre basi
- Algoritmo di conversione per moltiplicazioni successive
- Problema della periodicità e della conseguente approssimazione
- Periodicità di una rappresentazione dipendente dalla base
- Rappresentazione in binario
- Mantissa ed esponente
- Forma normalizzata
- Scalamento degli esponenti per l'effettuazione della somma
- Problema dell'underflow

Rappresentazione dell'informazione

- Varie rappresentazioni sono possibili per la medesima informazione
  - Es. Testo scritto su carta o registrato su nastro
- Le rappresentazioni R1 e R2 sono equivalenti se data R1 è possibile ricavare R2 e viceversa
  - Es. Trascrizione del testo data la sua registrazione e viceversa
- Scelta della rappresentazione
  - Spesso convenzionale ...
  - ... ma quasi sempre legata a vincoli
  - Es. Rappresentazione binaria negli elaboratori

Analogico vs Digitale

Informazione **esplicita nel supporto**: per analogia

Si definisce ***analogico*** un procedimento che rappresenta un fenomeno con continuità, per esempio: un orologio classico che con il moto regolare della lancetta segna il trascorrere del tempo in modo continuo

Informazione **implicita nella codifica**

E' **digitale** un procedimento che rappresenta lo stesso fenomeno traducendolo in cifre (dall'inglese **digit**, cifra) e quindi in modo discontinuo, come per esempio avviene in un orologio a cristalli liquidi numerico nel quale la stessa durata temporale viene misurata da una successione di scatti.

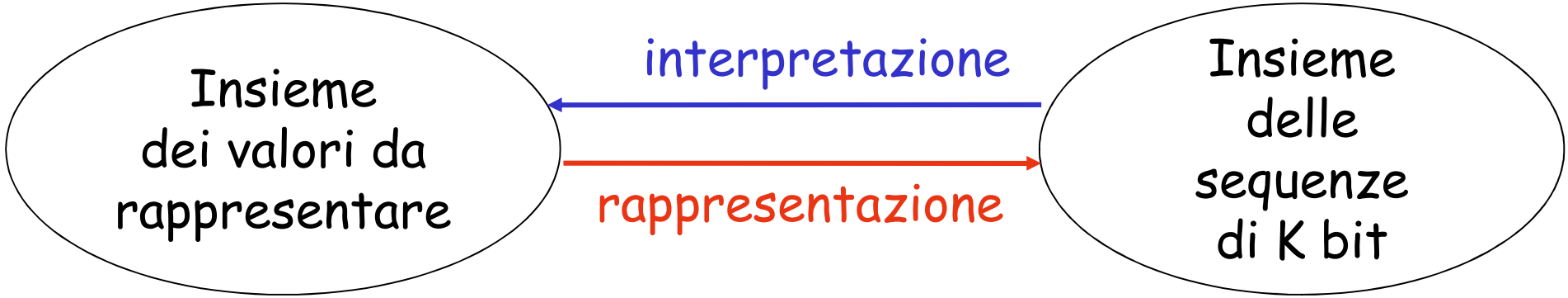
Analogico e digitale

- Contrariamente a quanto si potrebbe credere la registrazione digitale, pur essendo "a salti", **può essere più precisa di quella analogica** in quanto non soggetta ad interferenze e disturbi.
- Occorre però che **il numero di valori utilizzati sia molto elevato**, in modo da cogliere ogni più piccola sfumatura!

Rappresentazione dell'informazione

- Il principio di funzionamento di un computer si basa sulla **logica binaria**.
- Per comprenderne l'idea di base, possiamo pensare a **un interruttore, che può essere aperto o chiuso**, o ad una scheda che può essere forata o meno in un punto, o ad un suono che può essere presente o no, o ancora ad una riflessione ottica che può verificarsi o meno.

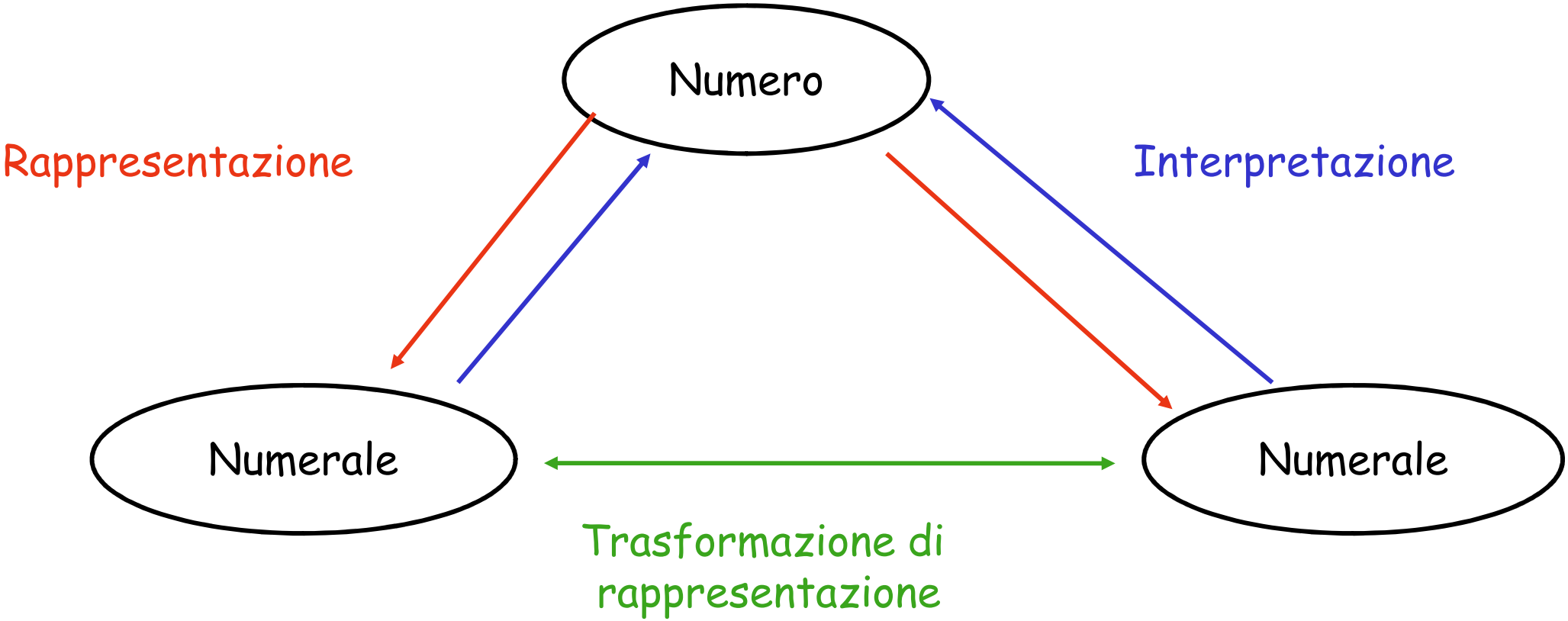
Rappresentazione ed interpretazione



- **Rappresentare** un insieme di valori con sequenze di bit significa associare ad ogni valore una sequenza distinta
- Con sequenze di **K bit** si possono rappresentare **2^K valori**.
- Per rappresentare **M valori**, è necessario **usare k >= $\lceil \log_2 M \rceil$**
- **Interpretare** una sequenza di bit significa ricostruire il valore che essa rappresenta

Valori numerici: numeri e numerali

Numero: concetto astratto
Numerale: rappresentazione del numero
(Esempi: "dodici", 12, XII, 0xC)



BIT (1)

- In informatica ed in teoria dell'informazione, la parola **bit** ha due significati molto diversi, a seconda del contesto in cui rispettivamente la si usa

BIT (2)

1. un bit è l'unità di misura dell'informazione (dall'inglese "**b**inary un**it**"), definita come **la quantità minima di informazione che serve a discernere tra due possibili alternative equiprobabili**.
2. un bit è una cifra binaria, (in inglese "**b**inary dig**it**") ovvero uno dei due simboli del sistema numerico binario, classicamente chiamati *zero* (0) e *uno* (1);

BIT (3)

- La differenza di significato tra le due definizioni, può riassumersi con una frase come: "la ricezione degli ultimi 100 bit (**simboli binari**) di messaggio ha aumentato la nostra informazione di 40 bit (**quantità di informazione**)"
  - differenza tra **numero di bit** e **quantità di informazione rappresentata**
- La quantità di informazione portata da un simbolo dipende dalla probabilità a priori che si ha di riceverlo

Multipli del BIT

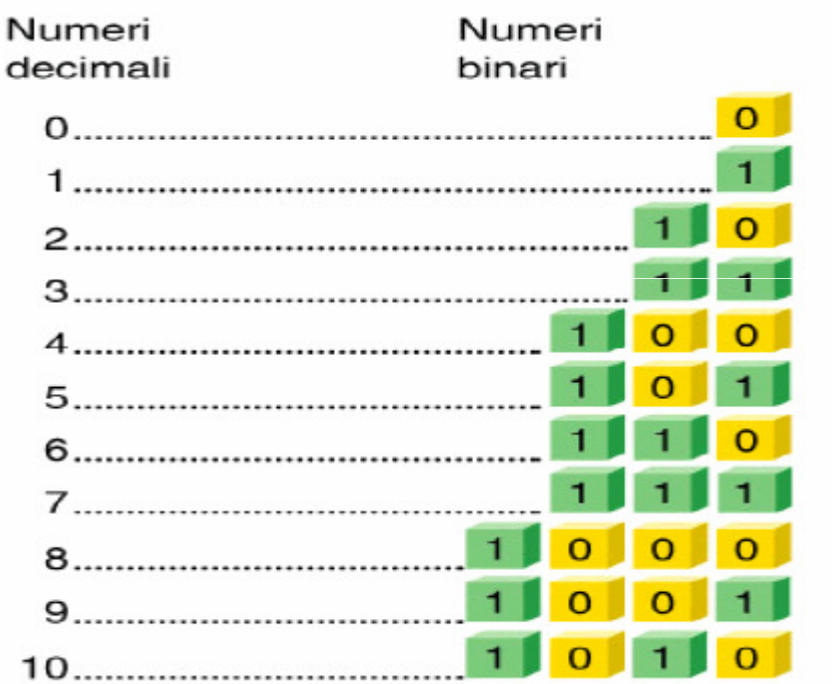
Multipli del bit					
Prefissi SI			Prefissi binari		
Nome	Simbolo	Multipli	Nome	Simbolo	Multipli
kilobit	kb	10^3	kibibit	Kib	2^{10}
megabit	Mb	10^6	mebibit	Mib	2^{20}
gigabit	Gb	10^9	gibibit	Gib	2^{30}
terabit	Tb	10^{12}	tebibit	Tib	2^{40}
petabit	Pb	10^{15}	pebibit	Pib	2^{50}
exabit	Eb	10^{18}	exbibit	Eib	2^{60}
zettabit	Zb	10^{21}	zebibit	Zib	2^{70}
yottabit	Yb	10^{24}	yobibit	Yib	2^{80}

BIT

- Il bit può essere pensato come un interruttore che può assumere dolo due stati:
  - **0 = aperto**
  - **1 = chiuso**
- Questa variabile (**bit** o "binary digit", cifra binaria) che assume solo due stati (0 e 1) è **l'unità minima di informazione e la base dell'algebra binaria.**

Byte

- Un computer "ragiona" unicamente interpretando gruppi di bit, cioè comandi rappresentati da sequenze di "uno" e di "zero" (per esempio, 00101100).
- Convenzionalmente, **8 bit** costituiscono **1 byte**.
  - 10011010
  - 00011101
  - 11001010



Numeri a precisione finita (1)

- I numeri a precisione finita sono quelli rappresentati con **un numero finito di cifre**.
  - Fissate le caratteristiche del numero è determinato anche l'insieme di valori rappresentabili.
- Le operazioni con i numeri a precisione finita causano errori quando il loro risultato non appartiene all'insieme dei valori rappresentabili:
  - **Underflow**: si verifica quando il risultato dell'operazione è minore del più piccolo valore rappresentabile
  - **Overflow**: si verifica quando il risultato dell'operazione è maggiore del più grande valore rappresentabile
  - **Non appartenenza all'insieme**: si verifica quando il risultato dell'operazione, pur non essendo troppo grande o troppo piccolo, non appartiene all'insieme dei valori rappresentabili (divisione tra due interi **può** generare un reale)

Numeri a precisione finita (2)

- Esempio: si considerino i numeri a tre cifre senza virgola e senza segno (es. **159**)
- Non possono essere rappresentati:
  - Numeri superiori a 999
  - Numeri negativi
  - Frazioni e numeri irrazionali
- Alcuni errori possibili in operazioni fra tali numeri:
  - 600+600 = 1200 → Overflow
  - 300-600 = -300 → Underflow
  - 007/002 = 3.5 → Non appartenenza all'insieme

Numeri a precisione finita (3)

- L'algebra dei numeri a precisione finita è diversa da quella convenzionale, poiché alcune delle proprietà non vengono rispettate.
  - A differenza dei numeri interi, i numeri a precisione finita **non rispettano la chiusura** rispetto alle operazioni di somma, sottrazione e prodotto.
  - La proprietà associativa [a + (b - c) = (a + b) – c] e la proprietà distributiva [a × (b - c) = a × b – a × c] non sono rispettate
- Esempi (numeri a precisione finita di 3 cifre senza virgola e senza segno):
  - Chiusura: 050×050=2500 (Overflow)
  - Prop. associativa:
    - (400 + 300) – 500 = 200
    - 400 + (300 – 500) = Underflow
  - Prop. Distributiva:
    - 50 × (50 - 40) = 500
    - 50 × 50 – 50 × 40 = Overflow

Sistema di numerazione posizionale (1)

- Le cifre hanno un valore diverso a seconda della posizione che occupano nel numerale
- Un sistema é definito da una coppia (B, A) dove:
  - B è un **intero > 1** detto **base**
  - A è un insieme di B simboli distinti detti **cifre**
- sistema decimale, B = 10, A = {0,1,2,3,4,5,6,7,8,9}
- binario, B = 2, A = {0,1}
- ottale, B = 8, A = {0,1,2,3,4,5,6,7}
- esadecimale, B = 16, A={0,1,…,9,A,B,…,F}
- Una singola cifra rappresenta univocamente un numero compreso fra 0 e B - 1

Sistema di numerazione posizionale (2)

- In una sequenza di cifre, il valore rappresentato dalla cifra d_K dipende dalla posizione **K** : **valore(d_k) = d_k* B^k**

- Nel numero **100** (base **10**), 1 rappresenta **$1*10^2$ = 100**

$$\text{Valore}(d_k) = d_k * B^k$$

$$100 = 1 * 10^2$$

Sistema di numerazione posizionale (3)

- Per numeri **frazionari** si usa anche una sequenza di cifre con **peso negativo**, il cui inizio è segnalato da un punto "."
- In 0.001 (base 10), 1 rappresenta **$1*10^{-3} = 10^{-3}$**
- Il valore complessivo rappresentato da

$$d_n \ldots d_2 d_1 d_0 \, . \, d_{-1} d_{-2} \ldots d_{-m}$$

è la somma di tutti i valori rappresentati dalle cifre e moltiplicati per il loro peso

$$d_n B^n + d_{n-1} B^{n-1} + \ldots + d_0 + d_{-1} B^{-1} + \ldots + d_{-m} B^{-m}$$

Sistemi di numerazione posizionale (4)

- I sistemi di numerazione più utilizzati in informatica:
  - Sistema **decimale** (b=10) 0 1 2 3 4 5 6 7 8 9
  - Sistema **binario** (b=2) 0 1: ogni cifra, detta bit (**B**inary dig**IT**), può essere rappresentata direttamente tramite un livello elettrico di tensione
  - Sistema **ottale** (b=8) 0 1 2 3 4 5 6 7
  - Sistema **esadecimale** (b=16) 0 1 2 3 4 5 6 7 8 9 A B C D E F: è utilizzato poiché è molto compatto e ben si presta alla traduzione in valori binari, poiché ogni cifra corrisponde esattamente a 4 cifre binarie.

Sistemi di numerazione posizionale (5)

- Ad ogni numero **corrispondono rappresentazioni nelle basi diverse**:
  - Binario: $11010 = 1\times2^4+1\times2^3+0\times2^2+1\times2^1+0\times2^0 = 16+8+0+2+0 = 26$
  - Ottale: $32 = 3\times8^1+2\times8^0 = 24+2 = 26$
  - Decimale: $26 = 2\times10^1+6\times10^0 = 20+6 = 26$
  - Esadecimale: $1A = 1\times16^1+10\times16^0 = 16+10 = 26$
- L'insieme dei simboli utilizzati dalle varie basi non è disgiunto: **è necessario specificare la radice utilizzata**:
  - $11010_2 = 32_8 = 26_{10} = 1A_{16}$

Notazione

- La base B utilizzata in una rappresentazione R si indica tipicamente con $(R)_B$
  - *Esempio*: $(1011)_2$ -> $d_3 = 1$, $d_2 = 0$, $d_1 = 1$, $d_0 = 1$
  - *Valore* =

$$1 * 2^3 + 0 * 2^2 + 1 * 2^1 + 1 * 2^0 =$$

$$8 + 2 + 1 = (11)_{10}$$

- La cifra più a sinistra è detta **cifra più significativa** (most significant digit) e quella più a destra è detta **cifra meno significativa** (least significant digit)
- Se **B = 2**, si usano gli acronimi msb (most significant bit) ed lsb (least significant bit)
- Se **B = 16**, si usa il prefisso 0x per indicare la base
  - Esempio: $0x721 = (721)_{16}$
  - si parla, in questo caso, di numeri esadecimali

Esempi

Numero	Base	Simboli	Rappresentazione
15	2	{0,1}	**1111**
15	8	{0,1,2,…,7}	**17**
15	10	{0,1,2,…9}	**15**
15	16	{0,…9,A,…,F}	**F**

Numerazione binaria

- Affiancando un sufficiente numero di bit **è possibile rappresentare qualunque numero**.
- Ogni posizione rappresenta **una potenza crescente di 2**:

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	**0**	**0**	**1**	**0**	**1**	**1**	**0**

$= 128*1 + 64*0 + 32*0 + 16*1 + 8*0 + 4*1 + 2*1 + 1*0 = 150$ decimale

- Questo procedimento **consente di convertire un numero binario in decimale**

Conversione fra basi

- **Problema**: dato un numero N1 in base B1 trovarne la rappresentazione N2 in base B2

$$(N1)_{B1} \rightarrow (N2)_{B2}$$

- Nel seguito, se chiaro dal contesto, **N denota sia il numero che il numerale**

Conversione da base 10 a base B

- **N** numero da convertire (N > 0, intero)
- Per ogni intero **D** (1 <= D <= N)

$$N = Q \times D + R$$

  - **Q** (Q <= N) **quoziente** della divisione fra gli interi N e D
  - **R** (0 <= R < D) **resto** della divisione fra gli interi N e D

- Notazione
  - Q = N/D
  - R = N mod D

Conversione da base 10 a base B (2)

- Nella nuova base B, N sarà rappresentato dalle sue cifre d moltiplicate per la base B opportunamente elevata a potenza

$$d_{K-1}B^{K-1} + \ldots + d_1B + d_0 = (d_{K-1}B^{K-2} + \ldots + d_1) \times B + d_0$$

- Quindi per analogia con la precedente formula
  - $d_0 = R = N \bmod B$
  - $d_{K-1}B^{K-2} + \ldots + d_1 = Q = N / B$
  - $B = D$

Infatti

$$N = Q \times D + R$$

Algoritmo di conversione : decimale -> binario

25 2

N intero, B base

Poni **h = 0**

Finché N <> 0 **esegui**

d_h = N mod B
N = N / B
h = h + 1
fine

Esempio: $(25)_{10} = (??)_2$

N	N / 2	N mod 2	Cifra
25	12	1	$d_0 = 1$
12	6	0	$d_1 = 0$
6	3	0	$d_2 = 0$
3	1	1	$d_3 = 1$
1	0	1	$d_4 = 1$

$(25)_{10} = (11001)_2$

Esempi

L'algoritmo è lo stesso anche per altre basi

$(30)_{10} = (??)_{16}$

N	N/16	N mod 16	Cifra
30	1	14	$d_0 = E$
1	0	1	$d_1 = 1$

$(30)_{10} = (1E)_{16} = 0x1E$

$(30)_{10} = (??)_2$

N	N/2	R mod 2	Cifra
30	15	0	$d_0 = 0$
15	7	1	$d_1 = 1$
7	3	1	$d_2 = 1$
3	1	1	$d_3 = 1$
1	0	1	$d_4 = 1$

$(30)_{10} = (11110)_2$

Sistema numerico esadecimale (1)

- Il **sistema numerico esadecimale** (spesso abbreviato come **esa** o **hex**) è un **sistema numerico posizionale** in base 16
- **utilizza 16 simboli** invece dei 10 del sistema numerico decimale tradizionale.
- Per l'esadecimale si usano in genere simboli da **0 a 9** e poi **le lettere da A a F**, per un totale **di 16 simboli**

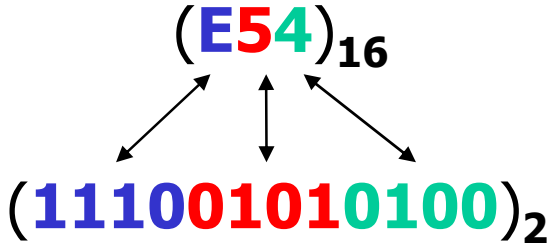
Sistema numerico esadecimale (2)

- Ecco una tabella che confronta le rappresentazioni esadecimali, decimali e ottali dei numeri fino a 15
- il numero **decimale 79**, la cui rappresentazione binaria è **0100 1111**, può essere scritto come **4F** in esadecimale.
- Per fare questo il numero da trasformare è diviso in **gruppi da 4bit** a partire da destra
- **0100 1111** -> **4 F**

$0_{hex} = 0_{dec} = 0_{oct}$	0	0	0	0
$1_{hex} = 1_{dec} = 1_{oct}$	0	0	0	1
$2_{hex} = 2_{dec} = 2_{oct}$	0	0	1	0
$3_{hex} = 3_{dec} = 3_{oct}$	0	0	1	1
$4_{hex} = 4_{dec} = 4_{oct}$	0	1	0	0
$5_{hex} = 5_{dec} = 5_{oct}$	0	1	0	1
$6_{hex} = 6_{dec} = 6_{oct}$	0	1	1	0
$7_{hex} = 7_{dec} = 7_{oct}$	0	1	1	1
$8_{hex} = 8_{dec} = 10_{oct}$	1	0	0	0
$9_{hex} = 9_{dec} = 11_{oct}$	1	0	0	1
$A_{hex} = 10_{dec} = 12_{oct}$	1	0	1	0
$B_{hex} = 11_{dec} = 13_{oct}$	1	0	1	1
$C_{hex} = 12_{dec} = 14_{oct}$	1	1	0	0
$D_{hex} = 13_{dec} = 15_{oct}$	1	1	0	1
$E_{hex} = 14_{dec} = 16_{oct}$	1	1	1	0
$F_{hex} = 15_{dec} = 17_{oct}$	1	1	1	1

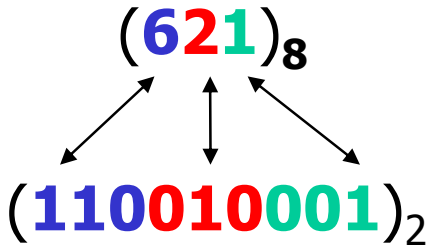
Conversioni fra basi 2/8/16

da HEX a BIN ogni cifra
HEX diventa 4 BIN



Da base 16 a base 2

da OCT a BIN ogni cifra
OCT diventa 3 BIN



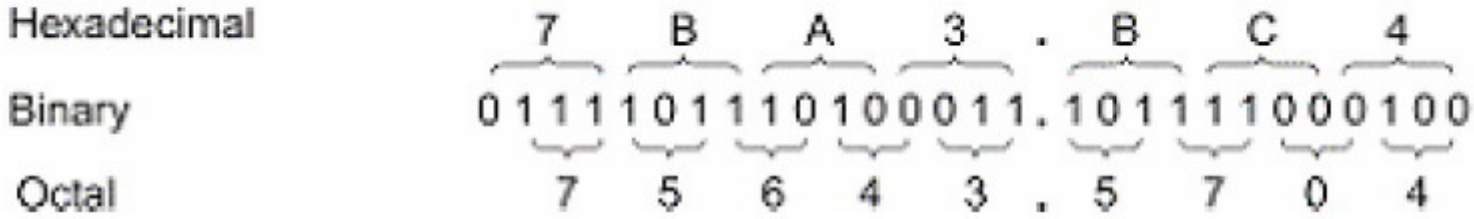
Da base 8 a base 2

$(E54)_{16} \leftrightarrow (111001010100)_2 \leftrightarrow (111001010100)_2 \leftrightarrow (7124)_8$

Da base 16 a base 8

Conversioni fra basi (1)

- **Binario →Ottale**
  - Dato che una cifra del sistema ottale è rappresentabile esattamente con tre cifre del sistema binario, la conversione può essere **ottenuta raggruppando le cifre binarie a 3 a 3 a partire dalla virgola binaria**. L'operazione contraria è ugualmente semplice, ogni cifra ottale viene convertita in esattamente tre cifre binarie.
- **Esadecimale →binario**
  - Il processo di conversione è equivalente a quello binario→ottale ma le cifre binarie devono essere considerate a gruppi di 4.



Conversioni fra basi (2)

- **Decimale →Binario (Metodo generale)**
  - Si procede sottraendo al numero da decomporre **la più grande potenza di 2 minore del numero da decomporre**. Il processo viene applicato ricorsivamente al resto della sottrazione. Il risultato binario si ottiene ponendo a uno le cifre corrispondenti alle potenze che sono state utilizzate nella decomposizione.

Es. conversione in binario di 1492.25_{10}

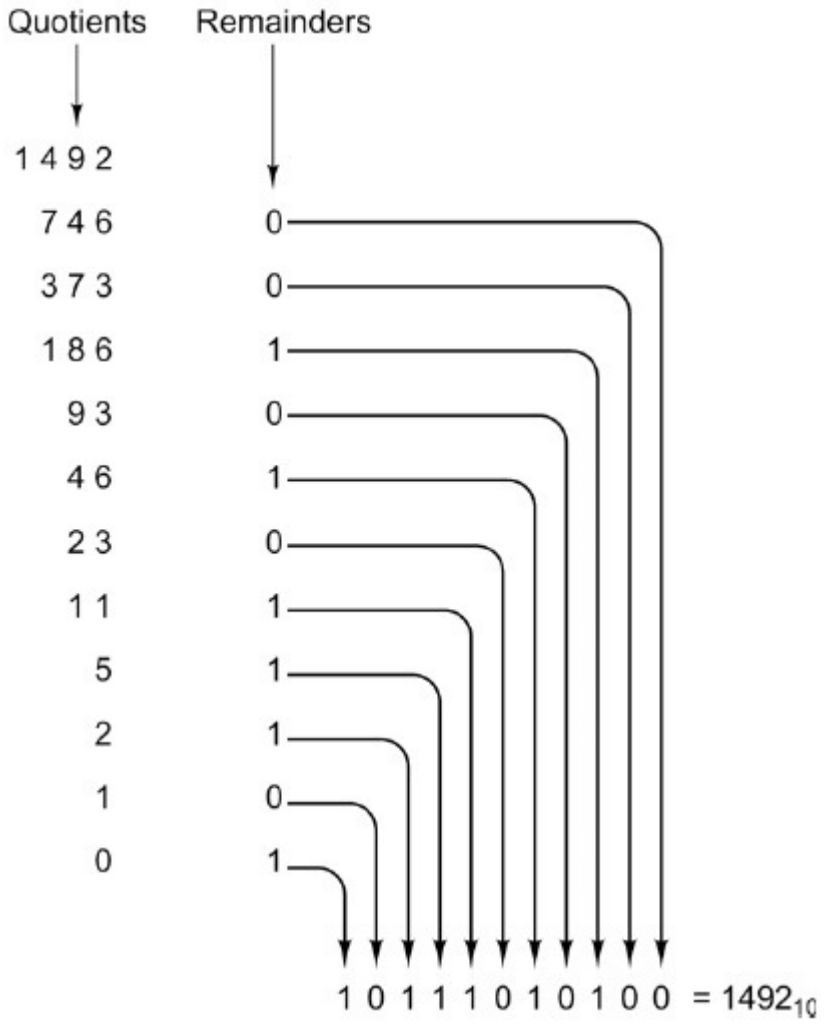
$1492.25 = 2^{10} + 468.25$
$468.25 = 2^{8} + 212.25$
$212.25 = 2^{7} + 84.25$
$84.25 = 2^{6} + 20.25$
$20.25 = 2^{4} + 4.25$
$4.25 = 2^{2} + 0.25$
$0.25 = 2^{-2}$

→ 10111010100.01

$\mathbf{1}*2^{10} + \mathbf{0}*2^{9} + \mathbf{1}*2^{8} + \mathbf{1}*2^{7} + \mathbf{1}*2^{6} + \mathbf{0}*2^{5} + \mathbf{1}*2^{4} + \mathbf{0}*2^{3} + \mathbf{1}*2^{2} + \mathbf{0}*2^{1} + \mathbf{0}*2^{0} + \mathbf{0}*2^{-1} + \mathbf{1}*2^{-2}$

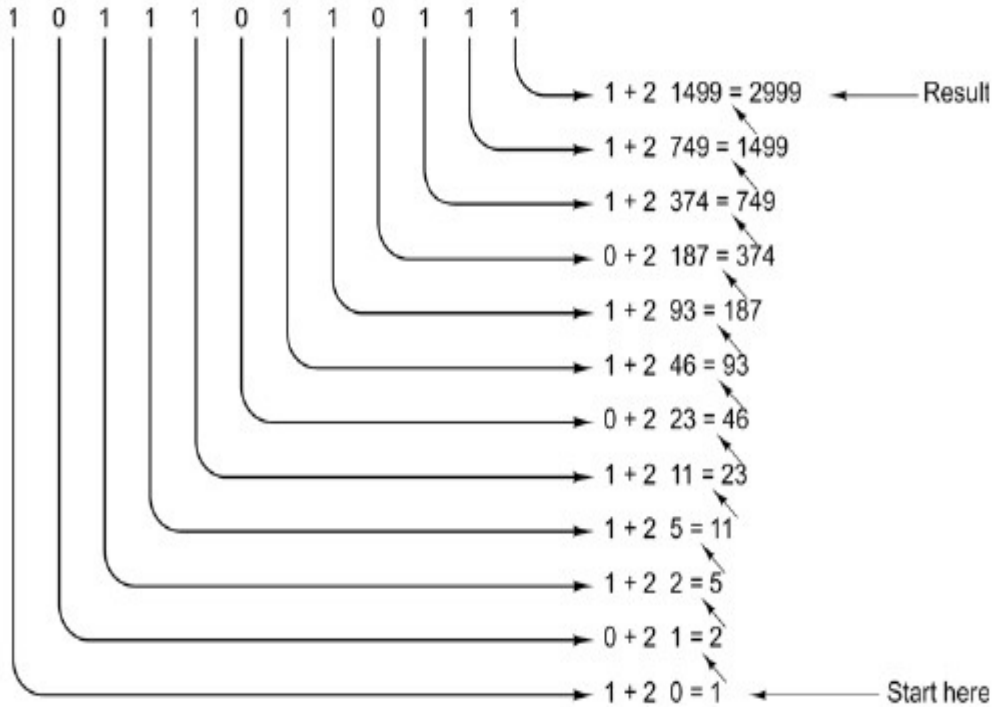
Conversioni fra basi (3)

- Decimale →Binario
  (*solo per numeri interi*)
  - La codifica viene ottenuta direttamente procedendo in modo ricorsivo.
  - **Si divide il numero per 2**
  - Il resto (0 o 1) farà parte del risultato mentre il quoziente verrà utilizzato come input al seguente passo di ricorsione.



Conversioni fra basi (4)

- **Binario → Decimale (Primo metodo)**
  - Si procede traducendo le singole cifre binarie alle corrispondenti potenze di due in base decimale e sommando i risultati parziali:
    - $10111010100.01 \rightarrow 2^{10} + 2^8 + 2^7 + 2^6 + 2^4 + 2^2 + 2^{-2} \rightarrow$ $1024+256+128+64+16+4+0.25 \rightarrow$ 1492.25
- **Binario →Decimale (Secondo metodo)**
  - La codifica viene ottenuta ricreando il valore posizionale di ogni cifra tramite successive moltiplicazioni per due a partire dalle cifre più significative verso quelle meno significative.
- **Altre conversioni**
  - Passando per il sistema binario o applicando i metodi sopra descritti (attenzione al corretto uso delle basi).



Esercizi di conversione

- Convertire da decimale a binario
  - 125, 230, 34
- Da binario a decimale
  - I valori ottenuti sopra
  - 10011, 11, 1000111, 111, 1000110
- Da Binario a ottale
  - 10011, 11, 1000111, 111, 1000110
- Da binario a esadecimale
  - 10011, 11, 1000111, 111, 1000110