

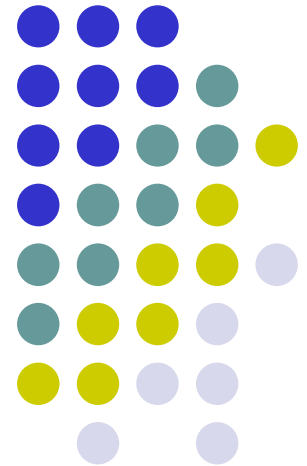
Laboratorio di Calcolatori 1

Corso di Laurea in Fisica

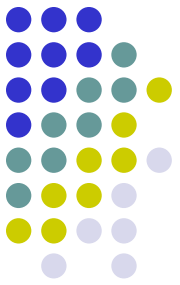
A.A. 2006/2007

Dott. Davide Di Ruscio

Dipartimento di Informatica
Università degli Studi di L'Aquila



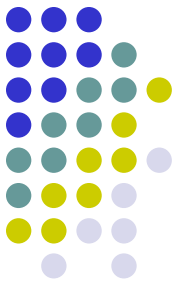
Nota



Questi lucidi sono tratti dal materiale distribuito dalla McGraw-Hill e basati su del materiale fornito dal Prof. Flammini Michele

Rappresentazione dell'informazione

(1/2)

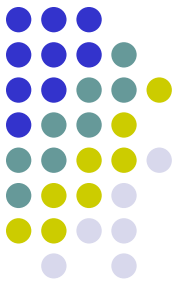


Codifica binaria dell'informazione: modalità di rappresentazione dei dati (numeri, caratteri, immagini, ...) in un elaboratore

- Bit:
 - può avere soltanto valore 0 o 1
 - è l'unità elementare di informazione memorizzabile in un elaboratore
 - il suo valore corrisponde ad un possibile stato di un dispositivo bistabile (es., interruttore aperto o chiuso, cerchietto di semiconduttore magnetizzato o smagnetizzato, presenza o assenza di corrente o tensione, alternanza fra luce e buio nella trasmissione di dati su fibra ottica, ecc....)
- Byte = 8 bit.
- Kbyte = 2^{10} byte = 1024 byte
- Mbyte = $2^{10} \times 2^{10} = 2^{20}$ byte
- Gbyte = 2^{30} byte
- Tbyte = 2^{40} byte
- Es: una RAM di 512 Mbyte contiene 512×2^{20} byte o $512 \times 2^{20} \times 8$ bit
- Poiché $2^{10} \approx 1000$, a volte si approssimano K con mille, M con milione, G con miliardo₃
...

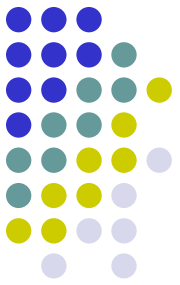
Rappresentazione dell'informazione

(2/2)



- Internamente ad un elaboratore tutto viene codificato con sequenze di bit in modo trasparente all'utente, poiché il procedimento inverso di presentazione delle informazioni secondo i nostri formalismi e percezioni è effettuato in modo automatico dai dispositivi di Input/Output
- Un bit permette di distinguere tra loro solo due elementi o possibilità, rappresentate o codificate rispettivamente dai valori *0* ed *1*
- Per codificare un numero superiore di elementi, è necessario utilizzare sequenze di bit, il cui numero aumenta all'aumentare della loro lunghezza
- Il numero di sequenze di lunghezza k è dato dal numero di possibili configurazioni ottenibili assegnando valori nell'ordine ai k bit, ossia 2^k

Codifica binaria (1/2)

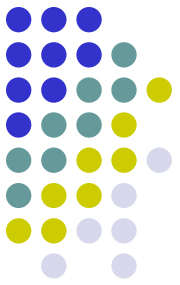


- Dato un insieme U di n elementi ed un numero intero k fissato, una codifica binaria di U a k bit consiste nell'assegnamento di sequenze di k bit agli elementi di U in modo univoco, ossia in modo tale che ad ogni elemento corrisponda un'unica sequenza e ad ogni sequenza un unico elemento
- Più formalmente una codifica è una funzione iniettiva $c:U \rightarrow \{0,1\}^k$

Esempio

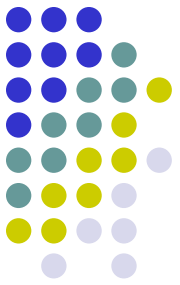
- $U = \{\text{Luca}, \text{Marco}, \text{Alessio}\}$
- Codifica di U a 2 bit: $c(\text{Luca})=00$, $c(\text{Marco})=01$ e $c(\text{Alessio})=10$

Codifica binaria (2/2)



- **Domanda:** quanto deve essere grande k per codificare un insieme U di n elementi?
- Con $k=1$ si hanno soltanto 2 sequenze diverse, ossia 0 e 1, per cui si possono codificare insiemi U con $n \leq 2$, ossia contenenti al più 2 elementi.
- Con $k=2$ si hanno soltanto 4 sequenze diverse, ossia 00, 01, 10 e 11, per cui si possono codificare insiemi U con $n \leq 4$
- Con $k=3$ si hanno soltanto 8 sequenze diverse, per cui $n \leq 8$
- ...
- Per un generico k deve essere $n \leq 2^k$, ossia $k \geq \log_2 n$
- In generale si cerca di scegliere k più piccolo possibile, in modo da risparmiare memoria
- Poiché k è un numero intero, il più piccolo valore possibile è $k = \lceil \log_2 n \rceil$

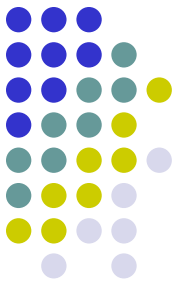
Riepilogando ...



n	$\min k$
2	1
3	2
4	2
5	3
6	3
7	3
8	3
9	4
...	...

n	$\min k$
16	4
17	5
...	...
32	5
33	6
...	...
64	6
65	7
...	..

n	$\min k$
128	7
129	8
...	...
256	8
257	9
...	...
512	9
513	10
...	...



Codifica caratteri

Includono

- caratteri alfanumerici (lettere e cifre)
- simboli (punteggiatura, operatori aritmetici, ...)
- caratteri di controllo (per formattazione, stampe, codici di trasmissione, ...)

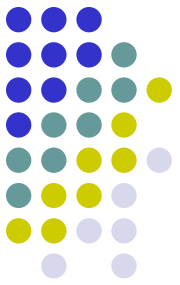
Alcuni codici noti: ASCII, EBCDIC, FIELDATA, ...

ASCII (American Standard Code for Information Interchange):

- 7 bit per carattere
- 1 bit per controllo errore (bit di parità, assegnato in modo da avere un numero pari di bit uguali ad 1 in tutti il byte)
- 128 caratteri rappresentati

Extended ASCII:

- 8 bit per carattere
- 256 caratteri rappresentati



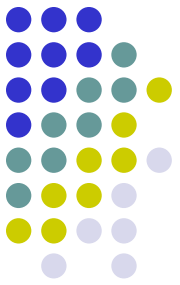
Codifica numeri

Codifiche per

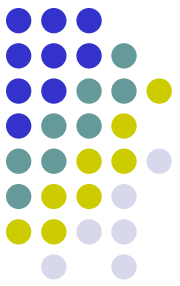
- naturali (interi non negativi)
- interi (anche negativi)
- frazionari (reali compresi tra 0 e 1)
- reali

Le codifiche dei numeri non sono arbitrarie, ma progettate per facilitare lo svolgimento delle operazioni aritmetiche e garantire diversi livelli di precisione

Sistemi di numerazione (1/4)



- Il sistema di numerazione comunemente usato è *l'arabico* che rappresenta i numeri naturali tramite sequenze di cifre
- E' un sistema di numerazione in **base dieci** (le cifre usate sono dieci $\{0, \dots, 9\}$)
- E' un sistema di numerazione **posizionale**, perché il significato attribuito a ciascuna cifra è funzione della posizione che tale cifra occupa nel numero



Sistemi di numerazione (2/4)

Sistema di numerazione posizionale in base 10:

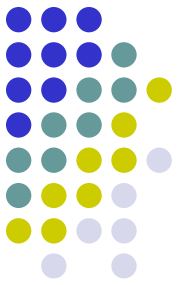
- utilizza 10 cifre (0, ..., 9)
- il valore di ogni cifra dipende dalla posizione all'interno del numero

Es. $357 = 7 + 5 \cdot 10 + 3 \cdot 100$

- in generale

$$\begin{aligned} & a_{n-1} a_{n-2} \dots a_1 a_0 = \\ & = a_0 \cdot 10^0 + a_1 \cdot 10^1 + \dots + a_{n-2} \cdot 10^{n-2} + a_{n-1} \cdot 10^{n-1} = \\ & = \sum a_i \cdot 10^i \quad a_i \in \{0, \dots, 9\} \end{aligned}$$

È possibile estendere a qualsiasi base $b > 0$



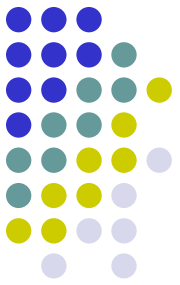
Sistemi di numerazione (3/4)

Sistema di numerazione posizionale in base b :

- utilizza b cifre
- il valore di ogni cifra dipende dalla posizione all'interno del numero:

$$\begin{aligned} & (a_{n-1}a_{n-2}\dots a_1a_0)_b = \\ & = a_0 \cdot b^0 + a_1 \cdot b^1 + \dots + a_{n-2} \cdot b^{n-2} + a_{n-1} \cdot b^{n-1} = \\ & = \sum a_i \cdot b^i \quad a_i \in \{0, \dots, b-1\} \end{aligned}$$

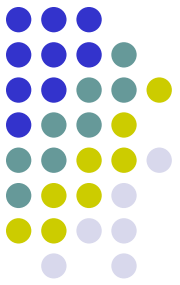
Sistemi di numerazione (4/4)



Quando la base b è maggiore di 10 , per indicare le cifre superiori a 9 per convenzione si utilizzano le lettere maiuscole dell'alfabeto

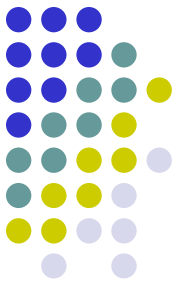
Esempio: $b=16$ (sistema esadecimale)

Cifra in base 16	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Valore in base 10	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15



Conversioni di base

- Come convertire un numero in base b nel suo equivalente in una base $b' \neq b$?
- Vedremo ora due regole generali, la cui distinzione fondamentale consiste nel fatto che effettuano le operazioni aritmetiche coinvolte rispettivamente nella base partenza b e nella base di arrivo b'



Regola 1

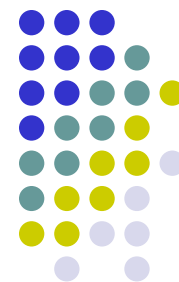
- Svolge le operazioni nella base di arrivo b' , per cui è molto adatta al caso in cui $b'=10$
- Consiste nell'applicare in modo diretto la sommatoria

$$(a_{n-1} a_{n-2} \dots a_1 a_0)_b = \sum a_i \cdot b^i \quad (1)$$

Regola 1:

7. si esprimono le cifre a_i e la base b nella base b' (solitamente banale)
8. si calcola la sommatoria (1)

Esempi



1. $10011_2 =$

$$= 1 \cdot 2^0 + 1 \cdot 2^1 + 0 \cdot 2^2 + 0 \cdot 2^3 + 1 \cdot 2^4 = 1 + 2 + 16 = 19$$

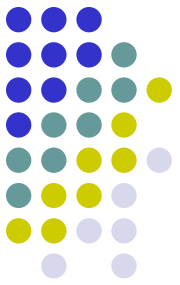
4. $237_8 =$

$$= 7 \cdot 8^0 + 3 \cdot 8^1 + 2 \cdot 8^2 = 7 + 24 + 128 = 159$$

7. $1AF_{16} =$

$$= 15 \cdot 16^0 + 10 \cdot 16^1 + 1 \cdot 16^2 =$$

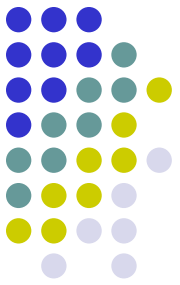
$$= 15 + 160 + 256 = 431$$



0000_2	=	0_{10}	=	0_8	=	0_{16}
0001_2	=	1_{10}	=	1_8	=	1_{16}
0010_2	=	2_{10}	=	2_8	=	2_{16}
0011_2	=	3_{10}	=	3_8	=	3_{16}
0100_2	=	4_{10}	=	4_8	=	4_{16}
0101_2	=	5_{10}	=	5_8	=	5_{16}
0110_2	=	6_{10}	=	6_8	=	6_{16}
0111_2	=	7_{10}	=	7_8	=	7_{16}
1000_2	=	8_{10}	=	10_8	=	8_{16}
1001_2	=	9_{10}	=	11_8	=	9_{16}
1010_2	=	10_{10}	=	12_8	=	A_{16}
1011_2	=	11_{10}	=	13_8	=	B_{16}
1100_2	=	12_{10}	=	14_8	=	C_{16}
1101_2	=	13_{10}	=	15_8	=	D_{16}
1110_2	=	14_{10}	=	16_8	=	E_{16}
1111_2	=	15_{10}	=	17_8	=	F_{16}

Regola 2

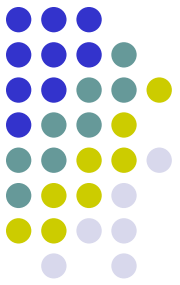
(delle divisioni successive)



- Svolge le operazioni nella base di partenza b , per cui è molto adatta al caso in cui $b=10$
- Si basa sull'osservazione che, dividendo il numero per b' :
 - il resto della divisione corrisponde alla cifra meno significativa del numero nella base b'
 - il quoziente della divisione corrisponde al numero ottenuto cancellando la cifra meno significativa dal numero di partenza espresso in base b'
 - le cifre più significative possono essere quindi determinate riapplicando ricorsivamente lo stesso metodo al quoziente

Esempio

$357:10$ è pari a 35 con resto 7
 $35:10$ è pari a 3 con resto 5
 $3:10$ è pari a 0 con resto 3



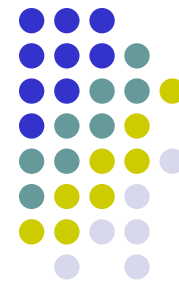
Regola 2:

2. si determinano il quoziente ed il resto della divisione del numero per b'
3. si prosegue come al passo 1. considerando di volta in volta come numero di partenza il quoziente della divisione effettuata nel passo precedente, finché non si determina un quoziente nullo
4. si scrivono tutti i resti ottenuti in ordine inverso, esprimendoli nella base b' (solitamente banale)

NB:

- il primo resto ottenuto (al passo 1.) corrisponde alla cifra meno significativa, mentre l'ultimo a quella più significativa
- se $b' < b$ tutti i resti ottenuti sono già espressi nella base b'

Esempi (1/3)



1) 37 ($b=10$ e $b'=2$)

$$\begin{array}{l} 37:2 = 18 \text{ con resto } 1 \\ 18:2 = 9 \text{ con resto } 0 \\ 9:2 = 4 \text{ con resto } 1 \\ 4:2 = 2 \text{ con resto } 0 \\ 2:2 = 1 \text{ con resto } 0 \\ 1:2 = 0 \text{ con resto } 1 \end{array} \left| \begin{array}{l} \uparrow \\ \uparrow \\ \uparrow \\ \uparrow \\ \uparrow \\ \uparrow \end{array} \right.$$

Quindi $37 = 100101_2$

Controprova:

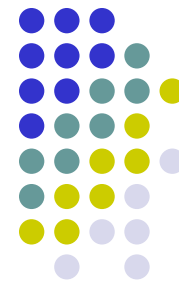
$$1x2^5 + 0x2^4 + 0x2^3 + 1x2^2 + 0x2^1 + 1x2^0 = 37_{10}$$

2) 3226 ($b=10$ e $b'=16$)

$$\begin{array}{l} 3226:16 = 201 \text{ con resto } 10 \text{ (A)} \\ 201:16 = 12 \text{ con resto } 9 \text{ (9)} \\ 12:16 = 0 \text{ con resto } 12 \text{ (C)} \end{array} \left| \begin{array}{l} \uparrow \\ \uparrow \\ \uparrow \end{array} \right.$$

Quindi $3226 = C9A_{16}$

Esempi (2/3)



3) 331 ($b=10$ e $b'=2$)

$$331:2 = 165 \text{ con resto } 1$$

$$165:2 = 82 \text{ con resto } 1$$

$$82:2 = 41 \text{ con resto } 0$$

$$41:2 = 20 \text{ con resto } 1$$

$$20:2 = 10 \text{ con resto } 0$$

$$10:2 = 5 \text{ con resto } 0$$

$$5:2 = 2 \text{ con resto } 1$$

$$2:2 = 1 \text{ con resto } 0$$

$$1:2 = 0 \text{ con resto } 1$$



Quindi $331_{10} = 101001011_2$

4) 331 ($b=10$ e $b'=8$)

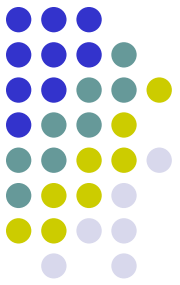
$$331:8 = 41 \text{ con resto } 3$$

$$41:8 = 5 \text{ con resto } 1$$

$$5:8 = 0 \text{ con resto } 5$$

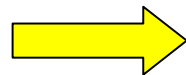


Quindi $331_{10} = 513_8$



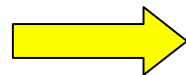
- Le BASI, 8 e 16 sono potenze di 2 ($8=2^3$ e $16=2^4$)
- Data la configurazione binaria c raggruppare le cifre in pacchetti di 3 (base 8) o in pacchetti di 4 (base 16) partendo da destra
- Esempio:

$$\begin{array}{ccccccc} 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ \underbrace{}_5 & \underbrace{}_1 & \underbrace{}_3 & & & & & & & \\ & 5 & 1 & 3 & & & & & & \end{array}$$



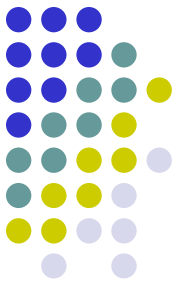
$$101001011_2 = 513_8$$

$$\underbrace{}_4 \quad \underbrace{}_4 \quad \underbrace{}_4$$



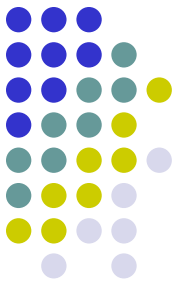
$$101001011_2 = 14B_{16}$$

$$1_{10} = 1_{16} \quad 4_{10} = 4_{16} \quad 11_{10} = B_{16}$$



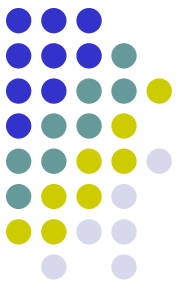
- Se abbiamo la configurazione $14B_{16}$ e vogliamo sapere quella binaria rappresento ogni cifra esadecimale con 4 bit

$$\begin{array}{ccc} & 14B_{16} & \\ & / \quad | \quad \backslash & \\ 0001_2 & & 1011_2 \\ & | & \\ & 0100_2 & \end{array} = 101001011_2$$



- Se abbiamo la configurazione 513_8 e vogliamo sapere quella binaria rappresento ogni cifra esadecimale con 3 bit

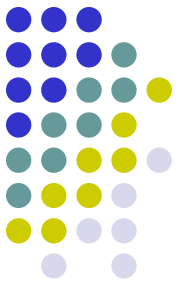
$$\begin{array}{ccc} & 513_8 & \\ & / \quad | \quad \backslash & \\ 101_2 & & 011_2 \\ & | & \\ & 001_2 & \end{array} = 101001011_2$$



Codifica numeri naturali

- Poiché l'unità elementare di informazione in un elaboratore è il bit, che corrisponde alle due cifre 0 e 1 , in modo naturale viene utilizzato il sistema di numerazione posizionale in base 2
- Fissato il numero di bit k da utilizzare nella rappresentazione
 1. si converte il numero di partenza nella base 2
 2. si antepongono bit uguali a 0 al numero determinato fino ad ottenere complessivamente esattamente k bit

NB: l'aggiunta dei bit pari a 0 nel passo 2. è necessaria perché nella memorizzazione del numero bisogna specificare per ogni bit (anche per i più significativi che convenzionalmente non indichiamo quando pari a 0) lo stato del relativo dispositivo bistabile; in caso contrario si potrebbero avere errori di rappresentazione.



Esempio

- 43 (con $k=8$ e $b'=2$)

$$43:2 = 21 \text{ con resto } 1$$

$$21:2 = 10 \text{ con resto } 1$$

$$10:2 = 5 \text{ con resto } 0$$

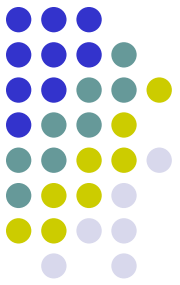
$$5:2 = 2 \text{ con resto } 1$$

$$2:2 = 1 \text{ con resto } 0$$

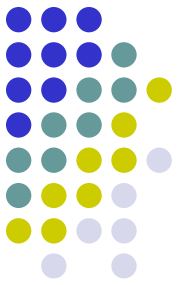
$$1:2 = \mathbf{0} \text{ con resto } 1$$

Quindi $43=101011_2$, la cui rappresentazione a 8 bit è

00101011



- Chiaramente, fissato il numero di bit k , è possibile rappresentare al più 2^k numeri interi, che vanno da 0 ($0\dots 0$) a 2^k-1 ($1\dots 1$)
- Quindi, l'intervallo dei numeri rappresentabile con k bit è $[0, 2^k-1]$
- **Overflow:** errore che si verifica quando si tenta di rappresentare un numero al di fuori dell'intervallo, ad esempio quando il risultato di un'operazione aritmetica è troppo grande



Punti Chiave

- Codifica binaria dell'informazione
- Codifica caratteri
- Codifica numeri
 - Sistemi di numerazione posizionali
 - Conversioni di base
 - Codifica naturali