



UNIVERSITÀ DEGLI STUDI DI MILANO

Rappresentazione dell'Informazione – lez. 1
Numeri naturali (cioè, interi senza segno)
Parte B

Prof. Marco Tarini
(lucidi in collaborazione con prof. Maria Damiani)



15



Sistema di numerazione binario

Il sistema in base 2 è detto binario. Cifre: $\{0,1\}$.

$$110_2 = (1 \times 2^2) + (1 \times 2^1) + (0 \times 2^0) = 6_{10}$$
$$1011_2 = (1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (1 \times 2^0) = 11_{10}$$


UNIVERSITÀ DEGLI STUDI DI MILANO

17

Il familiare algoritmo «con carta e penna» della somma nella comune base 10

$$124 + 86 =$$

$$\begin{array}{r} \textcolor{red}{1} \quad \textcolor{red}{1} \\ 1 \ 2 \ 4 \ + \\ \quad 8 \ 6 \\ \hline 2 \ 1 \ 0 \end{array}$$

18

Esempio in base 2

$$1101_2 + 1001_2$$

$$\begin{array}{r} \textcolor{red}{1} \quad \textcolor{red}{1} \\ 1101 \ + \\ \quad 1001 \\ \hline 10110 \end{array}$$

19

Altre operazioni su numeri codificati in base B

Conosciamo tutti (fin dalle elementari!) alcuni altri algoritmi standard per effettuare operazioni su numeri codificati in base 10, *facilmente adattabili ad una base qualsiasi* (e in particolare, alla base 2):

- Somma (come visto)
- Differenza ($A - B = C$)
- Prodotto ($A \times B = C$) [te lo ricordi?]
- Divisione (intera), cioè trovare il **quoziente** e il **resto**

20

Divisione intera (cioè fra numeri interi): quoziente e resto

Dati il dividendo A e il divisore D, trovare il quoziente Q e il resto R (con $R < D$) tali che
 $A = D \times Q + R$

- $A / B = Q$ (quoziente)
- $A \% B = R$ (resto)

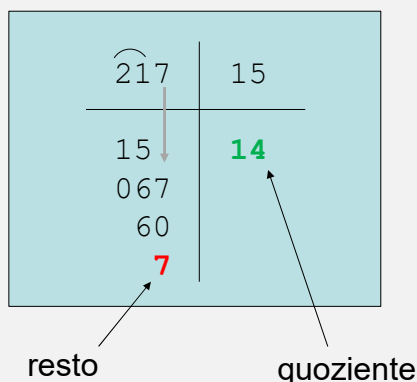
Ad esempio, con $A = 24$ e $D = 10$, diciamo:

- "24 diviso 10 uguale 2 col resto di 4" (infatti $24 = 10 \times 2 + 4$, cioè
- $24 / 10 = 2$ ("il 10 sta nel 24 2 volte...", 2 è il **quoziente** della divisione)
- $24 \% 10 = 4$ ("...col resto di 4", 4 è il **resto** della divisione) anche scritto
- " $24 \text{ rem } 10 = 4$ " (rem = reminder = resto), o
- " $24 \text{ mod } 10 = 4$ ", mod = modulo, "24 modulo 10")

21

Algoritmo per la divisione intera

Chi si ricorda l'algoritmo per la divisione intera (su base 10) quello delle elementari, da eseguire con carta e penna?



“il 15 nel 21 ci sta **1** volta...
 $15 \times 1 = 15$...
21 meno 15 fa 06...
«calo» il 7...
il 15 nel 67 ci sta **4** volte...
 $15 \times 4 = 60$...
67 meno 60 fa **7**...”

quindi
 $217 = 15 \times 14 + 7$

22

Un caso speciale: moltiplicazioni e divisioni per numeri B^k in base B

Osservazione: nella familiare base 10, è estremamente facile moltiplicare e dividere per 10, 100, 1000, o qualsiasi potenza di 10 (quelli che chiamiamo numeri “tondi”). Si tratta di un’operazione banale sulle rappresentazioni dei numeri!

Ad esempio, è banale svolgere (senza bisogno dell’algoritmo “standard”):

$$13 \times 10 = 130$$

$$124 \times 100 = 12400$$

$$540 / 10 = 54$$

$$32452 / 1000 = 32 \text{ col resto di } 452$$

$$352 \times 1,000,000 = 352,000,000$$

Nota: useremo la virgola
per raggruppare le cifre,
(non il punto)
usando la notazione inglese

Regola: “Ogni zero che aggiungo a destra moltiplica per $B=10$ (la base).”
“Ogni cifra che tolgo da destra divide per $B = 10$ (e la cifra tolta è il resto della divisione)”

Vale per ogni base!

23

Un caso speciale: moltiplicazioni e divisioni per 2^k in base 2 (cioè per 2, 4, 8, 16...)

In base 2...

raddoppiare = aggiungere uno **zero** in fondo

$$1101_b = 13_{10} \quad \text{allora} \quad 11010_b = 26_{10}$$

moltiplicare per otto (cioè 2^3) = aggiungere tre **zeri** in fondo

$$1101_b = 13_{10} \quad \text{allora} \quad 1101000_b = 13_{10} \times 8_{10}$$

$$\text{infatti, } 1101_b \times 1000_b = 1011000_b$$

Dividere per due = togliere l'ultima cifra.

Resto della divisione per due: l'ultimo bit (il least significant bit).

Conseguenza: in base due, l'ultimo bit vale 1 sse il numero è dispari!

Per dividere 1100101_b per 8 (dove $8 = 2^3$) è banale:

$$\begin{array}{c} 1100101_b \\ \swarrow \quad \searrow \\ \text{Quoziente} = 1100_b \quad \text{Resto} = 101_b \end{array}$$

24

Rappresentazione di un numero naturale in base qualsiasi B ... usando K cifre

Data una base B, supponiamo di poter utilizzare al massimo K cifre. Quanti e quali numeri posso rappresentare?

- Con K cifre in base B, possiamo rappresentare B^K numeri, da **0** a **$B^K - 1$** .
L'insieme dei numeri rappresentabili costituisce **l'intervallo di rappresentazione:**

$$[0, B^K - 1]$$

Esempio: con $K=3$, $B=10$, l'intervallo di rappresentazione è $[0, 999_{10}]$

- Il numero più grande rappresentabile con K cifre in base B è costituito da cifre tutte uguali a $B-1$.

Esempio: con $K=5$, $B=7$ il numero più grande è 66666_7

25

Rappresentazione dei numeri in un computer

In un computer, non è possibile rappresentare l'insieme infinito dei numeri, perchè il numero di cifre (bit) che possono comporre il numero è limitato.

Un numero viene invece rappresentato da una sequenza di bit di lunghezza prefissata. La lunghezza di questa sequenza è una potenza di 2 (ad esempio 8,16, 32, 64).

0	0	0	1	1	1	1	1
---	---	---	---	---	---	---	---

rappresentazione su 8 bit (1 byte)

1	0	0	0	1	0	0	1	1	0	1	0	1	0	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

rappresentazione su 16 bit (2 byte)

I numeri naturali rappresentabili in un computer vengono anche chiamati
interi senza segno

26

Esempio

Determinare l'intervallo di rappresentazione nei seguenti casi:

$K=8, B=2 \rightarrow [0, 255_{10}] \quad 255=2^8 - 1$

$K=2, B=16 \rightarrow [0, 255_{10}] \quad 255=16^2 - 1$

$K=2, B=7 \quad ?$

$K=11, B=2 \quad ?$

27

Numero minimo di cifre

Dato un numero N e una base B , il numero minimo K di cifre necessarie per rappresentare numeri da 0 a N si calcola come segue:

$$K = \lfloor \log_B N \rfloor + 1$$

funzione **floor** $\lfloor \rfloor$

$\text{floor}(x)$ determina il più grande intero $\text{int} \leq x$

Es: $\text{floor}(1.4) = 1$

funzione **ceiling** $\lceil \rceil$

$\text{Ceiling}(x)$ determina il più piccolo intero $\text{int} \geq x$. Es. $\text{Ceiling}(1.4) = 2$

28

Esempio

bit necessari per rappresentare i numeri fino a 35 e 128 in base 2

$$K = \lfloor (\log_2 35) \rfloor + 1 = 6$$

$$100011_2$$

$$K = \lfloor (\log_2 128) \rfloor + 1 = 8$$

$$1000\ 0000_2$$

Oppure, più lento: a mano.

Esempio, per 35:

«con 4 bit arrivo fino a $2^4 - 1 = 15$, non basta»

«con 5 bit arrivo fino a $2^5 - 1 = 31$, non basta»

«con 6 bit arrivo fino a $2^6 - 1 = 63$ mi basta!»

29

Overflow

Si consideri sempre il caso in cui si possono usare K cifre in base B .
Consideriamo l'operazione di somma.
La somma fra due numeri può non essere rappresentabile con K cifre!

Esempio: $K=2, B=10 \rightarrow 35+80$ non è rappresentabile

In questo caso, si dice che l'operazione di somma determina **overflow** (trabocco)

L'overflow si può verificare in altre operazioni, per es (ancora con $K=2, B=10$) genera overflow

- Successivo(99) (il calcolo del numero successivo di 99)
- $56 - 66$ (i numeri negativi non sono rappresentabili con numeri senza segno!)
- 20×30

31

Esempio: overflow nella somma di binari

Si assuma $B=2$, numero di cifre $K=4$

$1101_2 + 1001_2$

```
  1  1
 11012 +
 10012
 10110
```

$\rightarrow 10110_2$ OVERFLOW!

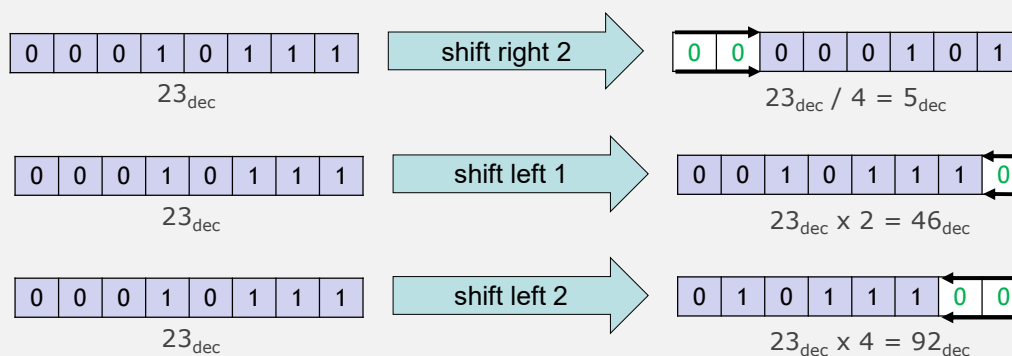
L'overflow viene rilevato perché l'ultimo riporto non è zero)

32

Moltiplicazioni e divisioni «banali» (cioè per multipli di 2) su numeri binari a k cifre: operazione di SHIFT

Su numeri a k cifre, l'operazione di shift (verso sinistra o destra) consiste nel far *scorrere* le cifre di N posti (a destra o a sinistra) (effettuando così moltiplicazioni o divisioni intere per 2^N)

Esempio, su un byte (binario ad 8 bit):



33

Conversione di base

- L'operazione di conversione di base converte un numero da una base ad un'altra.
- Abbiamo già visto la conversione da base B qualsiasi a base 10.
- Ora consideriamo:
 - Conversione da base 10 a base B
 - Conversione da base 16 a base 2, e da base 2 a base 16 (next lecture)

34

UNIVERSITÀ DEGLI STUDI DI MILANO



Conversione da base 10 a base B

Algoritmo di conversione di un numero x in base 10 in base B :

$i = 0$
Divido (div. intera) il numero x per B
Resto della divisione:
cifra i -esima in base B
 $i = i + 1$
Quoziente della divisione $\rightarrow x$
si prosegue fino a che il quoziente $x = 0$

L'ultimo resto è la cifra più significativa del numero in base B

35

UNIVERSITÀ DEGLI STUDI DI MILANO



Esempio

Convertire in base 2 il numero 1492

	Q	R	
1492:2	746	0	← Bit meno significativo
746:2	373	0	
373:2	186	1	
186:2	93	0	
93:2	46	1	
46:2	23	0	
23:2	11	1	
11:2	5	1	
5:2	2	1	
2:2	1	0	
1:2	0	1	← Bit più significativo

→ $1492_{10} \rightarrow 10111010100_2$

36

Verifica : conversione da base Binaria (cioè 2) a base 10

$$\begin{aligned} 101\ 1101\ 0100_2 &= 1 \times 2^{10} + 0 \times 2^9 + 1 \times 2^8 + \\ &\quad 1 \times 2^7 + 1 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + \\ &\quad 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 = \\ &\quad 1024 + 256 + 128 + 64 + 16 + 4 = \mathbf{1492}_{10} \end{aligned}$$

37

Esempio

Convertire $\mathbf{1492}_{10}$ in base 8:

	Q	R
1492 : 8	186	4 ← Cifra meno significativa
186 : 8	23	2
23 : 8	2	7
2 : 8	0	2 ← Cifra più significativa

$$1492_{10} \rightarrow 2724_8$$

38

Esempio

Convertire in base 5 il numero 148

	Q	R	
148:5	29	3	← Bit meno significativo
29:5	5	4	
5:5	1	0	
1:5	0	1	← Bit più significativo

→ $148_{10} \rightarrow 1043_5$

39

Esercizi di riepilogo

1. Specifica l'intervallo di rappresentazione (numero minimo e massimo esprimibile) dei numeri naturali in base 2 con 10 cifre
2. Qual è il numero minimo di cifre necessarie per rappresentare in base 2 i numeri naturali 57_{10} e 1023_{10} ?
3. Convertire 67_{10} in base 3, e verifica il risultato effettuando il passaggio inverso.
4. Quanto fa 10110_2 per 32_{10} , in binario? (hint: scopriilo senza «conti»)
5. Quanto fa $101010111011110101011010_2$ modulo 16_{10} , in decimale?
6. Scrivere un'operazione di Shift a sinistra a piacere che generi un overflow (su un byte). E' possibile generare un overflow con uno shift a destra?
7. Trova e somma fra loro due byte che rappresentano, come *unsigned int*, «ottantanove» e «centocinquanta» (effettuando l'operazione in binario). Si è generato un overflow?
8. Provare: effettua la moltiplicazione fra 1010_2 e 1011_2 in binario (generalizzando alla base 2 il familiare algoritmo «con carta e penna» della moltiplicazione, che sai già applicare, dalle elementari, in base 10). Verifica il risultato.

40