Aritmetica dei Calcolatori 1

Architettura degli Elaboratori e Laboratorio

1 Marzo 2013

1. Sistema di numerazione
   - sistema posizionale
2. rappresentazione binaria
   - cambio di base
   - basi potenze di 2
3. Rappresentazione binaria con segno

Sistema di numerazione posizionale

Il sistema di numerazione che usiamo è in base dieci e posizionale:

$2454 = 2 * 1000 + 4 * 100 + 5 * 10 + 4$

$2454 = 2 * 10^3 + 4 * 10^2 + 5 * 10^1 + 4 * 10^0$

Cosa significa?

- utilizziamo cifre da 0 a 9
- la stessa cifra assume un peso diverso a seconda della posizione

In generale, fissata una base B

- utilizziamo B simboli
- un numero si esprime come somme di potenze della base

Cambio di base: da 10 a B

Dato un numero N in base 10, le cifre di N in base B sono i resti delle divisioni successive di N per la base B.

Un esempio: $(28)_{10} = (11100)_2 = (103)_5$

$$\begin{array}{rcl} 28/2{=}14 & \textit{resto} & 0 \\ 14/2{=}\ 7 & \textit{resto} & 0 \\ 7/2{=}\ 3 & \textit{resto} & 1 \\ 3/2{=}\ 1 & \textit{resto} & 1 \\ 1/2{=}\ 0 & \textit{resto} & 1 \end{array}$$

$$\begin{array}{rcl} 28/5{=}5 & \textit{resto} & 3 \\ 5/5{=}1 & \textit{resto} & 0 \\ 1/5{=}0 & \textit{resto} & 1 \end{array}$$

La cifra meno significativa è il primo resto calcolato!

Cambio di base: da B a 10

Dato un numero di C cifre in base B, calcoliamo la sua rappresentazione in base 10 come:

$$(N)_{10} = \sum_{0}^{C-1} \sigma_i B^i$$

dove σ_i è il valore in base 10 della l'*i*-esima cifra in base B.

Un esempio:

$$(11100)_2 = 1 * 2^4 + 1 * 2^3 + 1 * 2^2 + 0 * 2^1 + 0 * 2^0 = (28)_{10}$$

$$(103)_5 = 1 * 5^2 + 0 * 5^1 + 3 * 5^0 = (28)_{10}$$

Rappresentazione binaria senza segno

Il numero $(28)_{10}$ è quindi rappresentato, utilizzando 32 bit:

0	0	0	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	1	0	0

MSB *LSB*

MSB = *most significant bit*
LSB = *least significant bit*

Considarando solo i numeri positivi, possiamo rappresentare l'intervallo $[0; 2^{32} - 1]$, cioè $[0; 4\,294\,967\,295]$

$$
\begin{aligned}
(0000\ \ 0000\ \ 0000\ \ 0000\ \ 0000\ \ 0000\ \ 0000\ \ 0000)_2 &= (0)_{10}\\
(0000\ \ 0000\ \ 0000\ \ 0000\ \ 0000\ \ 0000\ \ 0000\ \ 0001)_2 &= (1)_{10}\\
(0000\ \ 0000\ \ 0000\ \ 0000\ \ 0000\ \ 0000\ \ 0000\ \ 0010)_2 &= (2)_{10}\\
\dots\\
\dots\\
(1111\ \ 1111\ \ 1111\ \ 1111\ \ 1111\ \ 1111\ \ 1111\ \ 1101)_2 &= (4\,294\,967\,293)_{10}\\
(1111\ \ 1111\ \ 1111\ \ 1111\ \ 1111\ \ 1111\ \ 1111\ \ 1110)_2 &= (4\,294\,967\,294)_{10}\\
(1111\ \ 1111\ \ 1111\ \ 1111\ \ 1111\ \ 1111\ \ 1111\ \ 1111)_2 &= (4\,294\,967\,295)_{10}
\end{aligned}
$$

Osservazioni

Per passare da un numero in base 2 a una base 2^p, possiamo sfruttare il legame tra le due basi: raggruppiamo le cifre binarie a gruppi di p, partendo dal bit meno significativo, poi sostituiamo ogni gruppo di cifre con il relativo valore decimale.

Un esempio:

$$B = 4 = 2^2$$
$$\begin{aligned}(11100)_2 &= 01 \quad 11 \quad 00 \\ &= \ 1 \quad\ 3 \quad\ 0 \quad = (130)_4\end{aligned}$$

$$B = 8 = 2^3$$
$$\begin{aligned}(11100)_2 &= 011 \quad 100 \\ &= \ \ 3 \quad\ \ 4 \quad = (34)_8\end{aligned}$$

$$B = 16 = 2^4$$
$$\begin{aligned}(11100)_2 &= 0001 \quad 1100 \\ &= \ \ \ 1 \quad\ \ \ C \quad = (1C)_{16}\end{aligned}$$

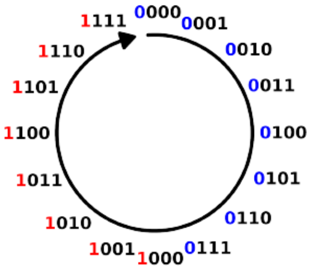
Quando finiscono le cifre?

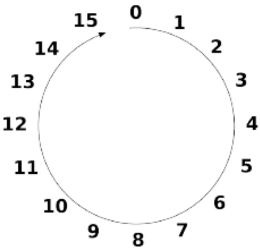
$A = 10$
$B = 11$
$C = 12$
$D = 13$
$E = 14$
$F = 15$
$\vdots$

Rappresentazione binaria (senza segno)

Come si rappresentano i numeri interi senza segno?

Abbiamo già visto come si rappresentano i numeri interi non negativi: con 4 bit possiamo rappresentare 16 numeri: $[0, 2^4 - 1]$

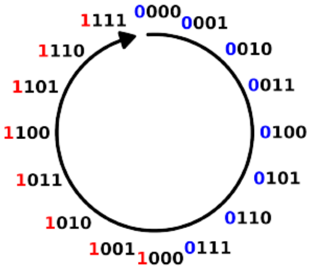


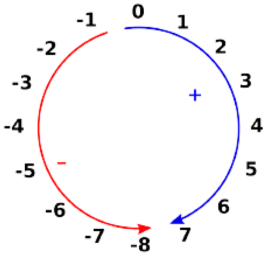


Rappresentazione binaria (con segno)

Come si rappresentano i numeri interi con segno?

Per memorizzare il segno, si utilizza il bit più significativo:
1 per i numeri negativi, 0 per lo zero e i numeri positivi.
In questo modo si possono rappresentare gli interi in $[-2^3, 2^3 - 1]$





Esempio

Per convertire un numero con segno in base 2 utilizzando N bit:

- osservare il segno
- convertire il suo modulo in base 2
- se negativo, calcolare il suo complemento a 2: negare ogni bit e sommare 1.

Esempio:

- $(-28)_{10} = -(28)_{10}$
- $(28)_{10} = (011100)_2$
- $(011100)_2 \rightarrow (100011)_2$
  $(100011)_2 + (000001)_2 = (100100)_2$

Estensione del segno

Immaginiamo di avere due interi con segno, rappresentati su un numero di bit differente, vogliamo calcolarne la somma.

Esempio:
$(30)_{10} = (00011110)_2$
$(-28)_{10} = (100100)_2$

Prendiamo il numero rappresentato con il minor numero di bit: consideriamo il bit più significativo e lo copiamo a sinistra tante volte quanti sono i bit mancanti per arrivare alla rappresentazione dell'altro numero.

Esempio:
$(30)_{10} = (00011110)_2$
$(-28)_{10} = (100100)_2 \rightarrow (11100100)_2$

Ora possiamo fare la somma:

$(00011110)_2$
$(11100100)_2$
$(00000010)_2 \quad = (2)_{10}$

Riassumendo:

- Cambio di segno
  Dato un numero in notazione binaria, per cambiare il segno basta calcolare il complemento a due (negare bit a bit e sommare 1)
- Estensione del segno
  Dato un numero in notazione binaria su N bit, portarlo a M bit, con M>N

Operazione di shift

Data una sequenza di bit, lo shift a destra di s bit equivale a una divisione per 2^s, mentre lo shift a sinistra di s bit equivale a una moltiplicazione per 2^s.

in C:

```
n >> s;  // shift a destra di s posizioni
n << s;  // shift a sinistra di s posizioni
```

Moltiplicazione I

Se osserviamo l'algoritmo di moltiplicazione in colonna:

```
   123
 x 456
 =====
   738  (this is 123 x 6)
  615   (this is 123 x 5, shifted one position to the left)
+ 492   (this is 123 x 4, shifted two positions to the left)
 =====
 56088
```

```
     1011   (this is 11 in decimal)
   x 1110   (this is 14 in decimal)
   ======
     0000   (this is 1011 x 0)
    1011    (this is 1011 x 1, shifted one position to the left)
   1011     (this is 1011 x 1, shifted two positions to the left)
+ 1011      (this is 1011 x 1, shifted three positions to the left)
=========
 10011010   (this is 154 in decimal)
```

Quindi, nel caso binario, la moltiplicazione si può implementare come successione di somme e shift.

Moltiplicazione II

Nel libro di testo trovate un esempio di moltiplicazione $(3)_{10} \times (2)_{10}$:

Iteration	Step	Multiplier	Multiplicand	Product
0	Initial values	0011	0000 0010	0000 0000
1	1a: 1 $\Longrightarrow$ Prod = Prod + Mcand	0011	0000 0010	0000 0010
	2: Shift left Multiplicand	0011	0000 0100	0000 0010
	3: Shift right Multiplier	0001	0000 0100	0000 0010
2	1a: 1 $\Longrightarrow$ Prod = Prod + Mcand	0001	0000 0100	0000 0110
	2: Shift left Multiplicand	0001	0000 1000	0000 0110
	3: Shift right Multiplier	0000	0000 1000	0000 0110
3	1: 0 $\Longrightarrow$ no operation	0000	0000 1000	0000 0110
	2: Shift left Multiplicand	0000	0001 0000	0000 0110
	3: Shift right Multiplier	0000	0001 0000	0000 0110
4	1: 0 $\Longrightarrow$ no operation	0000	0001 0000	0000 0110
	2: Shift left Multiplicand	0000	0010 0000	0000 0110
	3: Shift right Multiplier	0000	0010 0000	0000 0110