Calcolo numerico e **programmazione**
Rappresentazione dei numeri

Tullio Facchinetti
<tullio.facchinetti@unipv.it>

16 marzo 2012

10:54

http://robot.unipv.it/toolleeo

Rappresentazione dei numeri nei calcolatori

- l'unità minima di informazione nei calcolatori digitali è il bit
- il bit corrisponde ad un sistema fisico dotato di stati stabili: passa/non passa corrente, tensione alta/bassa, condensatore scarico/carico, ecc.
- i calcolatori perciò si basano sul sistema numerico binario, ovvero le cifre 0 e 1 (esistono delle eccezioni)
- si utilizzano cioè un insieme di bit per rappresentare le cifre binarie
- il numero di bit utilizzati è generalmente un multiplo di 8 (cioè si utilizzano 1, 2, 4, 8 byte)

Rappresentazione dei numeri nei calcolatori

- si consideri un byte (8 bit)
- un byte permette di rappresentare 2^8 stati differenti
- può memorizzare 256 diverse configurazioni corrispondenti ai primi 256 numeri naturali (0-255):

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

esiste un'interpretazione intuitiva di questa rappresentazione:

$$N = b_7 \cdot 2^7 + b_6 \cdot 2^6 + b_5 \cdot 2^5 + b_4 \cdot 2^4 + b_3 \cdot 2^3 + b_2 \cdot 2^2 + b_1 \cdot 2^1 + b_0 \cdot 2^0$$

esempio $N = 37$:

b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
0	0	1	0	0	1	0	1

I numeri negativi

esiste un problema:
come rappresentare i numeri negativi?

Modulo e segno (binario naturale)

rappresentazione in modulo e segno
detta anche binario naturale

- si utilizza un bit per rappresentare il segno del numero considerato
- $0 \rightarrow +$ (numero positivo)
- $1 \rightarrow -$ (numero negativo)
- se si considera un byte, rimangono 7 bit per il modulo del numero
- i numeri rappresentabili sono perciò $\pm[0 \ldots 127]$

$$\pm \quad b_6 \quad b_5 \quad b_4 \quad b_3 \quad b_2 \quad b_1 \quad b_0$$

Modulo e segno (binario naturale)

consideriamo per semplicità solo 4 bit

in modulo e segno:

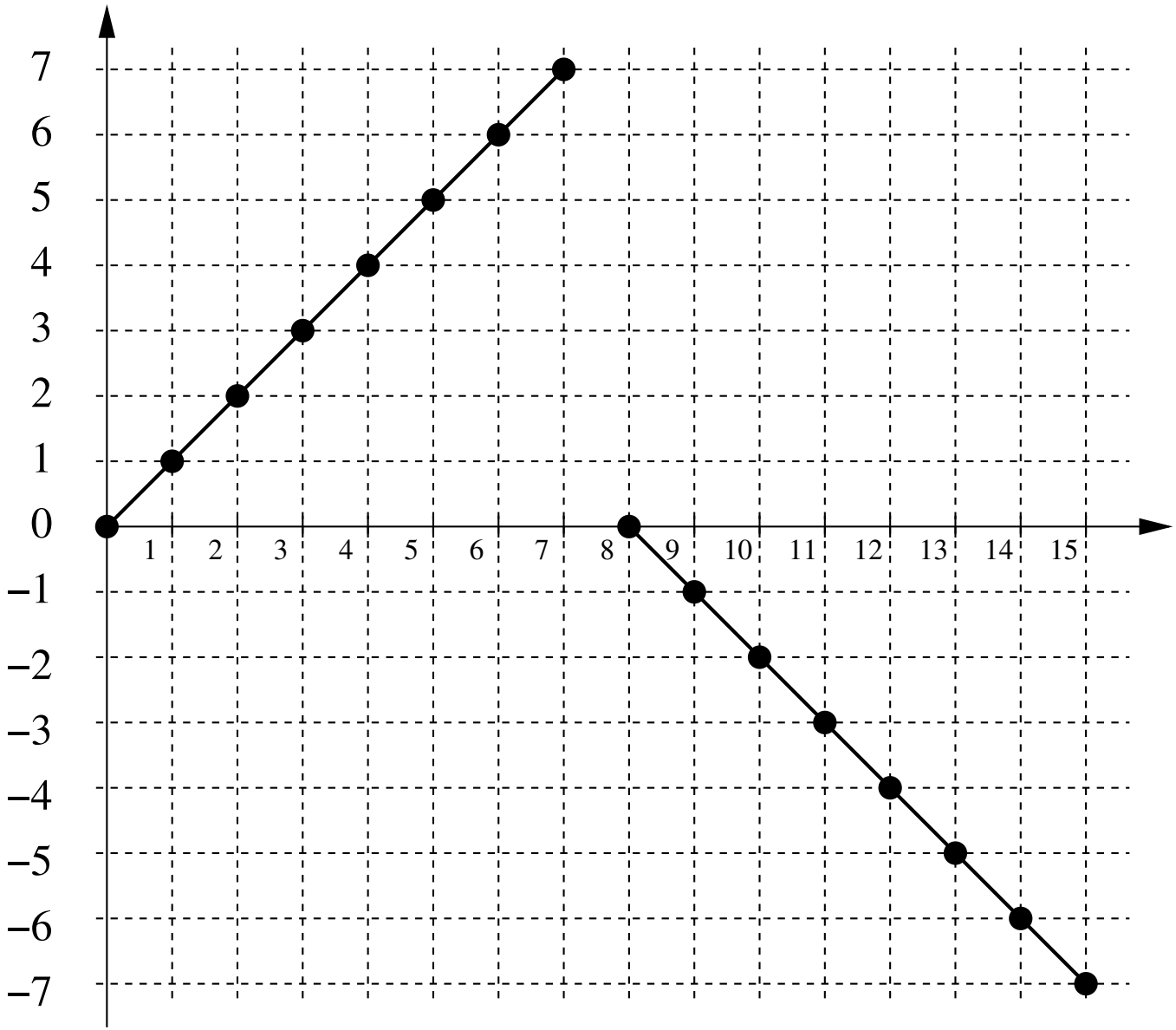
$+/-$ b_2 b_1 b_0

in valore assoluto:

b_3 b_2 b_1 b_0

numero rappresentato	rappresentazione binaria	valore assoluto
+7	0111	7
+6	0110	6
+5	0101	5
+4	0100	4
+3	0011	3
+2	0010	2
+1	0001	1
+0	0000	0
-0	1000	8
-1	1001	9
-2	1010	10
-3	1011	11
-4	1100	12
-5	1101	13
-6	1110	14
-7	1111	15

Modulo e segno (binario naturale)



Modulo e segno (binario naturale)

- esistono due rappresentazioni diverse dello 0 distanti '8' fra di loro
- un incremento nella rappresentazione corrisponde ad un incremento per numeri positivi, ma un decremento per numeri negativi
- numero minimo: $-2^{n-1}+1$
- numero massimo: $2^{n-1}-1$

Rappresentazione modulo e segno: problemi

- le operazioni minime di cui si deve disporre per poter realizzare qualsiasi operazione aritmetica sono addizione e sottrazione
- si supponga che il calcolatore abbia una Unità Aritmetica che realizzi indipendentemente le due operazioni

di fronte ad una somma algebrica, il calcolatore dovrebbe:

- confrontare i due segni
- se uguali, attivare il circuito di addizione
- se diversi, identificare il maggiore (in valore assoluto) ed attivare il circuito di sottrazione
- completare il risultato con il segno corretto

non è evidentemente pratico!

Rappresentazione in complemento

17 - 24 = - (24 - 17) = -7

0	1	7	-
0	2	4	=
9	9	3	

993 è il complemento di 7

questo esempio suggerisce la possibilità di utilizzare il complemento per rappresentare i numeri negativi

Rappresentazione in complemento

consideriamo una diversa situazione

$$\begin{array}{cccc} & 5 & 4 & 1 & + \\ & 6 & 2 & 8 & = \\ \hline 1 & 1 & 6 & 9 & \end{array}$$

sommando due numeri di 3 cifre, si ottiene un risultato di 4 cifre

Rappresentazione in complemento

rappresentazione in complemento a 2:

- i numeri positivi sono rappresentati dal loro modulo e hanno il bit più significativo (segno) posto a 0
- i numeri negativi sono rappresentati dal complemento a 2 del corrispondente numero positivo, segno compreso
- i numeri negativi hanno il bit del segno sempre a 1
- metà delle configurazioni sono riservate a numeri positivi e metà ai numeri negativi
- discorsi analoghi possono essere fatti per basi diverse da 2: in base 10 un numero è negativo se la prima cifra è ≥ 5, in base 8 se ≥ 4, in base 16 se ≥ 8

Rappresentazione in complemento alla base

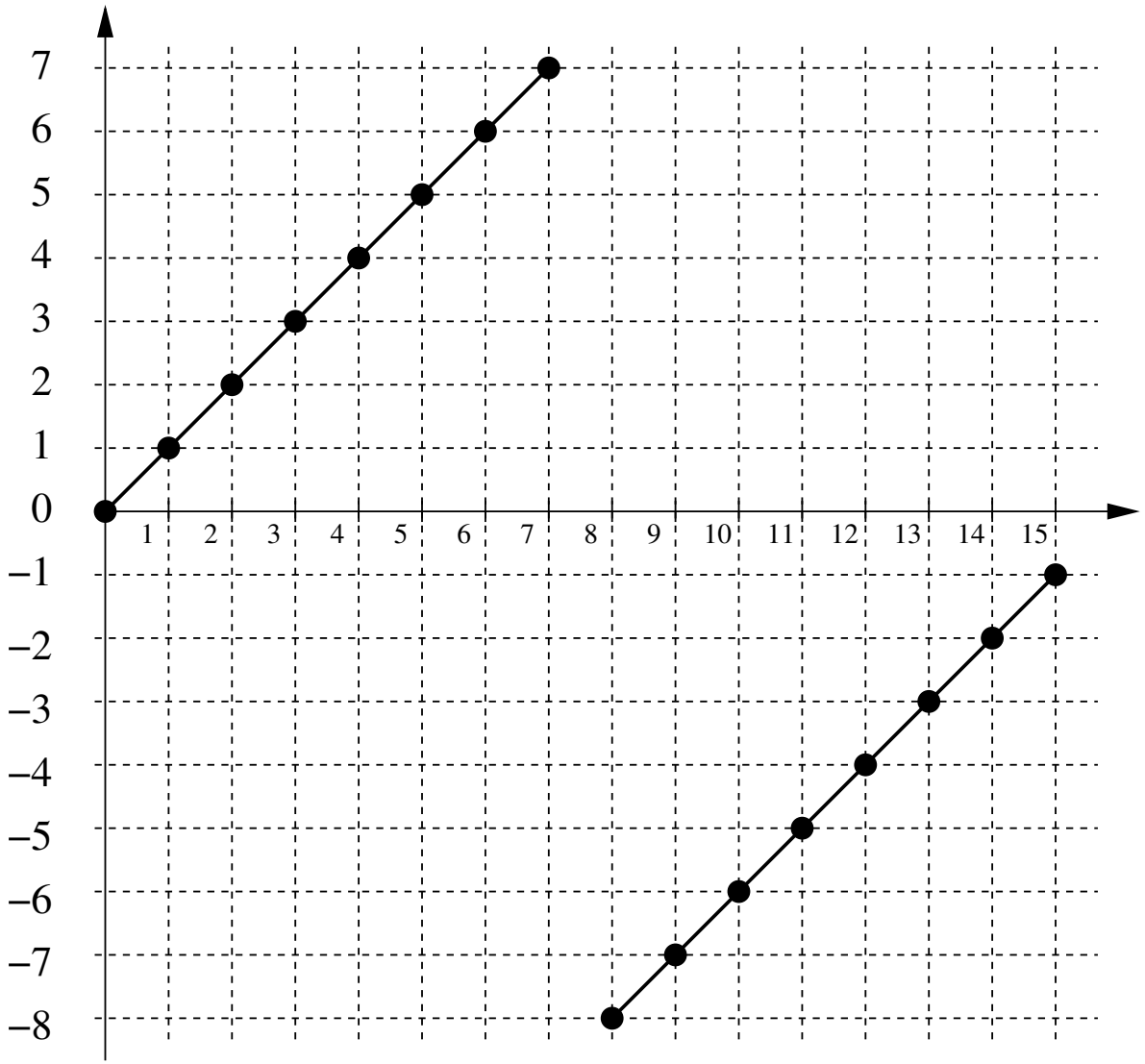
- il numero minimo è -2^{n-1}
- il numero massimo è $2^{n-1} - 1$
- -1 è rappresentato da tutti 1 qualunque sia il numero di bit considerato
- il numero può essere interpretato considerando il bit più significativo con segno negativo

$$N = -b_3 \cdot 2^3 + b_2 \cdot 2^2 + b_1 \cdot 2^1 + b_0 \cdot 2^0$$

Rappresentazione in complemento alla base

numero rappresentato	rappresentazione binaria	valore assoluto
+7	0111	7
+6	0110	6
+5	0101	5
+4	0100	4
+3	0011	3
+2	0010	2
+1	0001	1
0	0000	0
-1	1111	15
-2	1110	14
-3	1101	13
-4	1100	12
-5	1011	11
-6	1010	10
-7	1001	9
-8	1000	8

Rappresentazione in complemento alla base



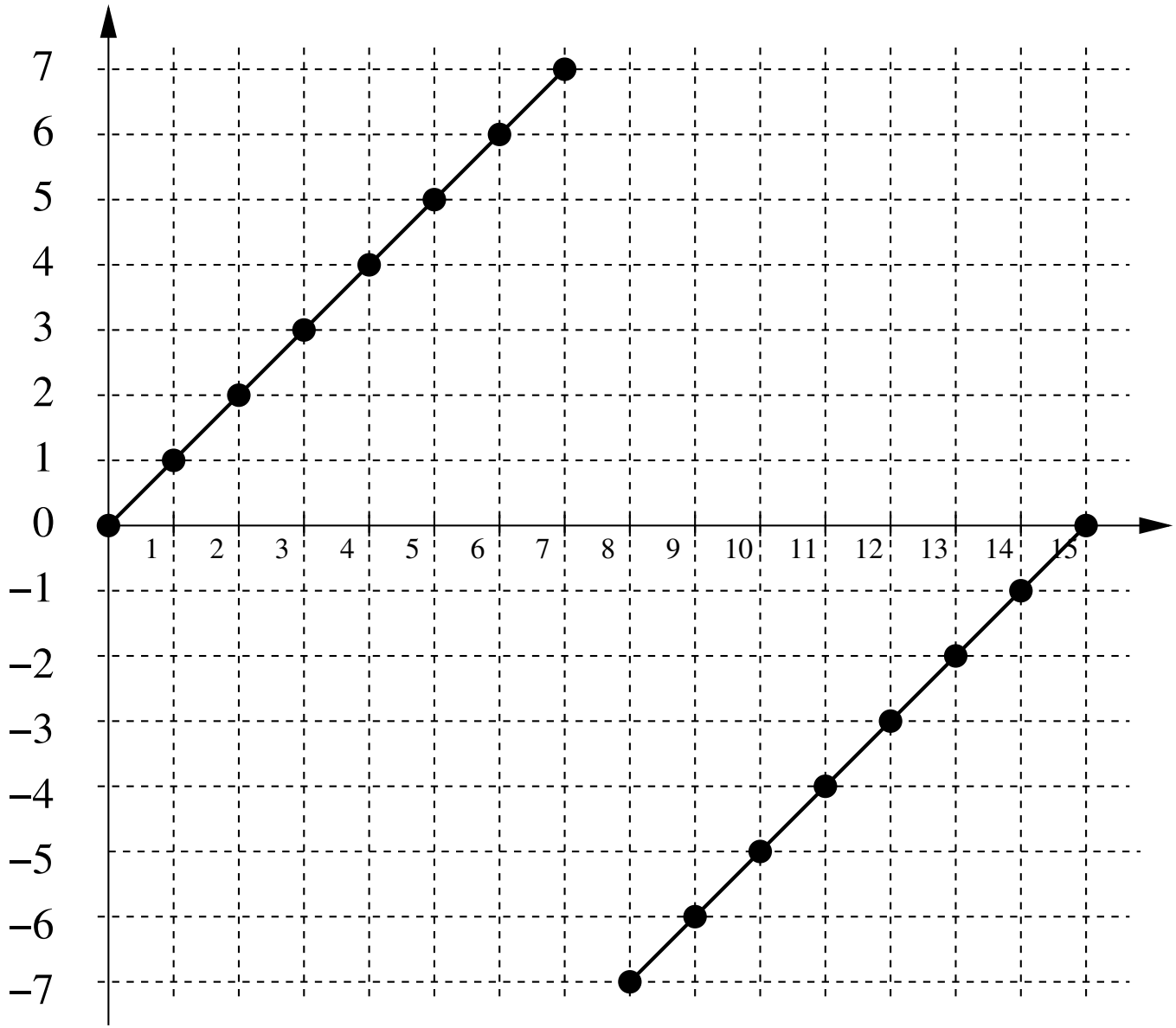
Caratteristiche del complemento alla base

- vi è una sola rappresentazione dello 0 (0000)
- tutti i numeri sono consecutivi
- la configurazione dello 0 segue quella di -1 trascurando il riporto
- il numero minimo è -2^{n-1}
- il numero massimo è $2^{n-1} - 1$

Rappresentazione in complemento alla base -1

numero rappresentato	rappresentazione binaria	valore assoluto
+7	0111	7
+6	0110	6
+5	0101	5
+4	0100	4
+3	0011	3
+2	0010	2
+1	0001	1
+0	0000	0
-0	1111	15
-1	1110	14
-2	1101	13
-3	1100	12
-4	1011	11
-5	1010	10
-6	1001	9
-7	1000	8

Rappresentazione in complemento alla base -1



Rappresentazione in complemento alla base -1

- vi sono due rappresentazioni dello 0 (0000 e 1111)
- nella rappresentazione in complemento a 1 i numeri negativi si ottengono complementando a 1 tutti i corrispondenti numeri positivi segno compreso
- i numeri negativi hanno il bit di segno sempre a 1

Tecnica dell'eccesso

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

- usato raramente per scopi particolari
- al numero viene sommata una costante fissa, ad esempio con n bit 2^{n-1}
- con questa convenzione il bit di segno è invertito

Schema riassuntivo

	modulo e segno	complemento a 2	complemento a 1
zero	$00\ldots0000/10\ldots00$	$00000\ldots00$	$00\ldots0/11\ldots1$
valore massimo	$2^{n-1}-1$	$2^{n-1}-1$	$2^{n-1}-1$
valore minimo	$-2^{n-1}-1$	-2^{n-1}	$-2^{n-1}+1$
bit di segno	0/1	0/1	0/1

Rappresentazione più comune degli interi

singola precisione: 2 byte (16 bit)

- $\max = 2^{15} - 1 = 32,767$
- $\min = -2^{15} = -32,768$

doppia precisione: 4 byte (32 bit)

- $\max = 2^{31} - 1 = 2,147,483,647$
- $\min = -2^{31} = -2,147,483,648$

Esempi

$\beta = 10, N = (14320)_{10}$
$N = 1 \cdot 10^4 + 4 \cdot 10^3 + 3 \cdot 10^2 + 2 \cdot 10^1 + 0 \cdot 10^0$

$\beta = 2, N = (110100)_2$
$N = 1 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^+ 0 \cdot 2^1 = 32 + 16 + 4 = 52$

$\beta = 8, N = (3216)_8$
$N = 3 \cdot 8^3 + 2 \cdot 8^2 + 1 \cdot 8^1 + 6 \cdot 8^0 = 1536 + 128 + 8 + 6 = 1678$

$\beta = 16, N = (AB9E)_{16}$
$N = 10 \cdot 16^3 + 11 \cdot 16^2 + 9 \cdot 16^1 + 15 \cdot 16^0 =$
$40960 + 2816 + 144 + 15 = 43935$

Operazioni in complemento

- con la tecnica del complemento, se si trascura il riporto, si può utilizzare un solo circuito per effettuare sia l'addizione, sia la sottrazione:

$$A - B = A + (10 - B)$$

- analogo discorso con k cifre considerando il complemento a 10^k
- la sottrazione può essere sostituita dall'addizione con il numero complementato

si ricordi sempre di fissare il numero di cifre

Esempi

esempio (base 10):

A	43517
B	26106
A-B	17411

A	43517
Comp B	73894
	117411

esempio (base 2):

A	01001
B	00110
A-B	00011

A	01001
Comp B	11010
	100011

esempio (base 16):

A	C2A61
B	00B02
A-B	C1F5F

A	C2A61
Comp B	FF4FE
	1C1F5F

Operazioni aritmetiche

se si utilizza la tecnica del complemento a 1 occorre sommare il riporto al risultato finale
esempi:

A	43517	A	43517	17410	+
B	26106	Comp B	73893	1	=
A-B	17411		117410	17411	

A	01001	A	01001	00010	+
B	00110	Comp B	11001	1	=
A-B	00011		100010	00011	

A	C2A61	A	C2A61	C1F5E	+
B	00B02	Comp B	FF4FD	1	=
A-B	C1F5F		1C1F5E	C1F5F	

Aritmetica degli interi con tecnica del complemento

esempi con 6 cifre binarie

complemento a 2

19 + (-17)

	0	1	0	0	1	1
	1	0	1	1	1	1
1	0	0	0	0	1	0

(+2)

complemento a 1

19 + (-17)

	0	1	0	0	1	1
	1	0	1	1	1	0
1	0	0	0	0	0	1
						1
	0	0	0	0	1	0

(+2)

Aritmetica degli interi con tecnica del complemento

esempi con 6 cifre binarie

complemento a 2

(-19) + 17

1	0	1	1	0	1
0	1	0	0	0	1
1	1	1	1	1	0

(-2)

complemento a 1

(-19) + 17

1	0	1	1	0	0
0	1	0	0	0	1
1	1	1	1	0	1

(-2)

Aritmetica degli interi con tecnica del complemento

esempi con 6 cifre binarie

complemento a 2

(-17) + (-2)

	1	0	1	1	1	1
	1	1	1	1	1	0
1	1	0	1	1	0	1

(-19)

complemento a 1

(-17) + (-2)

	1	0	1	1	1	0
	1	1	1	1	0	1
1	1	0	1	0	1	1
						1
	1	0	1	1	0	0

(-19)

Aritmetica degli interi con tecnica del complemento

esempi con 6 cifre binarie

$$
\begin{array}{cccccc}
\multicolumn{6}{c}{19 + 17} \\
0 & 1 & 0 & 0 & 1 & 1 \\
0 & 1 & 0 & 0 & 0 & 1 \\
\hline
1 & 0 & 0 & 1 & 0 & 0 \\
\multicolumn{6}{c}{\text{(-28)}}
\end{array}
$$

- si sono sommati due numeri positivi e si è ottenuto un numero negativo
- il fenomeno si chiama traboccamento o overflow

Aritmetica degli interi con tecnica del complemento

esempi con 6 cifre binarie

complemento a 2

(-19) + (-17)

	1	0	1	1	0	1
	1	0	1	1	1	1
1	0	1	1	1	0	0

(28)

complemento a 1

(-19) + (-17)

	1	0	1	1	0	0
	1	0	1	1	1	0
1	0	1	1	0	1	0
						1
	0	1	1	0	1	1

(27)

- si sono sommati due numeri negativi e si è ottenuto un numero positivo
- il fenomeno si chiama traboccamento o overflow

Aritmetica degli interi con tecnica del complemento

esempi con 6 cifre binarie

complemento a 2

(-13) + (-19)

	1	1	0	0	1	1
	1	0	1	1	0	1
1	1	0	0	0	0	0

(-32)

complemento a 1

(-13) + (-19)

	1	1	0	0	1	0
	1	0	1	1	0	0
1	0	1	1	1	1	0
						1
	0	1	1	1	1	1

(31)

- l'operazione in complemento a 2 è corretta, mentre in complemento a 1 ha dato traboccamento
- infatti -32 non è rappresentabile con 6 bit in complemento a 1

Rappresentazione di interi

esercizio:
rappresentare i numeri con 8 bit e complemento alla base [-1]

numero	complemento a 2	complemento a 1
15	00001111	00001111
-15	11110001	11110000
0	00000000	00000000
1	00000001	00000001
-1	11111111	11111110
144	non rappresentabile	non rappresentabile
128	non rappresentabile	non rappresentabile
-128	10000000	non rappresentabile

Moltiplicazione

esempio: $-4 \cdot 6$

$4 \rightarrow 100$

$6 \rightarrow 110$

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

(-24)

- ovviamente ci possono essere problemi di traboccamento
- numeri in virgola fissa N_1 e N_2 sono memorizzati come $N_1 \cdot 2^n$ e $N_2 \cdot 2^n$, quindi il loro prodotto è $N_1 \cdot N_2 \cdot 2^{2n}$, vi sono cioè 4 cifre decimali in eccesso che vanno trascurate

Moltiplicazione

esempio: $-4 \cdot -6$

$4 \rightarrow 100$

$6 \rightarrow 110$

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

(24)

- ci possono essere problemi di traboccamento
- numeri in virgola fissa N_1 e N_2 sono memorizzati come $N_1 \cdot 2^n$ e $N_2 \cdot 2^n$, quindi il loro prodotto è $N_1 \cdot N_2 \cdot 2^{2n}$, vi sono cioè 4 cifre decimali in eccesso che vanno trascurate

Rappresentazione dei numeri in virgola fissa

- il numero complessivo di cifre significative dei numeri che possono essere rappresentate in un calcolatore è limitato dalla capacità di una cella di memoria (con k bit, in modulo e segno, si rappresentano i numeri compresi fra $-2^{k-1}+1$ e $2^{k-1}-1$)
- quando si utilizzano numeri in cui sia presente sia una parte intera, sia una decimale, si può ricorrere alla rappresentazione detta in virgola fissa in cui si fissa la posizione che la virgola deve avere all'interno del numero da rappresentare
- ciò equivale a stabilire a priori il numero di cifre da utilizzare sia per la parte intera, sia per quella decimale
- per i numeri negativi si può utilizzare ancora la tecnica del complemento

Rappresentazione dei numeri in virgola fissa

esempio: memorizzare 72.6 con 12 bit (4 decimali)

72	0	b_0
36	0	b_1
18	0	b_2
9	1	b_3
4	0	b_4
2	0	b_5
1	1	b_6
0	0	b_7

0.6		
1.2	1	b^{-1}
0.4	0	b^{-2}
0.8	0	b^{-3}
1.6	1	b^{-4}

$$(72.6)_{10} = (010010001001)_2$$

Rappresentazione dei numeri in virgola fissa

- rappresentare -72.6: $(-72.6)_{10} = (101101110111)_2$
- in pratica il numero è stato moltiplicato per 2^4, avendo stabilito di avere 4 cifre decimali

problema: ridotto intervallo di rappresentazione dei numeri e ridotta precisione di rappresentazione

Rappresentazione normalizzata dei numeri reali

$$\begin{aligned} 1758.37 &= 0.175837 \cdot 10^4 \\ -0.001 &= -0.1 \cdot 10^{-2} \\ 1 &= 0.1 \cdot 10^1 \\ 5000 &= 0.5 \cdot 10^4 \end{aligned}$$

in generale, qualunque numero A può essere rappresentato da:

1. il segno di A (S = segno, con la convenzione: 0 = +, 1 = -);
2. le cifre significative di A (M = mantissa) rappresentate in una forma normalizzata; se n sono le cifre utilizzate e B è la base del sistema di numerazione, l'intervallo di variabilità della mantissa è: $B^{-n} \leq M < 1$;
3. l'esponente (E) a cui bisogna elevare la base B per ottenere il fattore per cui moltiplicare la mantissa per ottenere A

$$A = S0.M \times B^E$$

Rappresentazione in virgola mobile

per una rappresentazione effettiva si conviene di:

- eliminare i simboli ridondanti
- fissare la lunghezza della mantissa
- fissare la lunghezza dell'esponente
- utilizzare per l'esponente una convenzione che permetta di rappresentare numeri positivi e negativi senza l'indicazione esplicita del segno (complemento alla base o tecnica dell'eccesso, sommando una opportuna costante)
- disporre gli elementi rimasti (segno, esponente, mantissa) in un ordine convenzionale

quindi A viene rappresentato dalla sequenza:

S E M

Rappresentazione in virgola mobile

esempio: in base 10: 8 cifre di mantissa, esponente compreso tra -49 e 49 (si somma 50)

1758.37	$0.175837 \cdot 10^4$	$+ 54\ 17583700$
-0.001	$-0.1 \cdot 10^{-2}$	$- 48\ 10000000$
1	$0.1 \cdot 10^1$	$+ 51\ 10000000$
5000	$0.5 \cdot 10^4$	$+ 54\ 50000000$

S	Ex	Ex	M	M	M	M	M	M	M	M

Formato dei dati numerici

- numeri reali in singola precisione: 4 byte (1 per l'esponente, 3 per la mantissa)
- l'esponente varia quindi tra -127 e 127
- si parla di dinamica per descrivere l'intervallo dei numeri rappresentabili circa $10^{-38} \ldots 10^{38}$
- la mantissa di 23 bit permette circa 7 cifre significative
- per valori inferiori a 10^{-38} si parla di underflow
- per valori superiori a 10^{38} si parla di overflow

Formato dei dati numerici

passi per ottenere la rappresentazione binaria

- trasformazione in binario
- normalizzazione
- memorizzazione

esempio: 75.125 (numero reale positivo > 1)

75	1	0.125	
37	1	0.250	0
18	0	0.500	0
9	1	1.000	1
4	0		
2	0		
1	1		
0			

$$(75.125)_{10} = (1001011.001)_2$$
$$1001011.001 \rightarrow 0.1001011001 \cdot 2^7$$

Formato dei dati numerici

esempio: 0.0375 (numero reale positivo a con $0 < a < 1$)

$$\begin{array}{ll} 0.0375 & 0 \\ 0.0750 & 0 \\ 0.1500 & 0 \\ 0.3000 & 0 \\ 0.6000 & 0 \\ 1.2000 & 1 \\ 0.4000 & 0 \\ 0.8000 & 0 \\ 1.6000 & 1 \end{array}$$

$$(0.0375)_{10} = (0.0000\overline{1001})_2$$
$$(0.0000\overline{1001})_2 = 0.\overline{1001} \cdot 2^{-4}$$

Standard IEEE

- un numero reale $\neq 0$ viene scritto in binario come $1.b_1b_2b_3\ldots \cdot 2^e$
- la mantissa α è perciò $1 \leq \alpha < 2$
- il numero viene memorizzato su 32 bit
  - 1 bit di segno
  - 8 bit di esponente
  - 23 bit per la mantissa
- poiché la prima cifra è sempre 1, questa cifra non viene memorizzata. Di fatto la mantissa ha 24 cifre binarie di precisione
- l'esponente viene memorizzato sommando il valore costante 127

esempio: 1.0
0 01111111 0000000 00000000 00000000

Esempi IEEE

- -1

  $1 \quad 01111111 \quad 0000000 \quad 00000000 \quad 00000000$

- $(2)_{10} \rightarrow (10)_2 = 10 \cdot 2^0 = 1.0 \cdot 2^1$

  $0 \quad 10000000 \quad 0000000 \quad 00000000 \quad 00000000$

- $(-3)_{10} \rightarrow (-11)_2 = -11 \cdot 20 = -1.1 \cdot 2^1$

  $1 \quad 10000000 \quad 1000000 \quad 00000000 \quad 00000000$

- $(0.5)_{10} \cdot (0.1)_2 = 0.1 \cdot 20 = 1.0 \cdot 2 - 1$

  $0 \quad 01111110 \quad 0000000 \quad 00000000 \quad 00000000$

- 0.666666

  $0 \quad 01111110 \quad 0101010 \quad 10101010 \quad 10101010$

- 3.141593

  $0 \quad 10000000 \quad 1001001 \quad 00001111 \quad 11011100$