Rappresentazione dell’informazione

Rappresentazione dell'informazione

- Il calcolatore elettronico é costituito da dispositivi in grado di assumere due soli valori: acceso e spento.

- Le informazioni sono rappresentate mediante due simboli :

{ **0** , **1** }

Sono detti: ***bit***

- *Come rappresentare i numeri e i caratteri nel calcolatore ?*

Mediante sequenze di bit: 01001101011

BYTE = 8 BIT

WORD= n BYTE, n≥2

Rappresentazione dell'informazione

Prof. G. Ascia

- Con N bit sono rappresentabili fino a 2^N simboli

N=2
00
01
10
11

N=3
000
001
010
011
100
101
110
111

Multipli di bit :

1 kilo(K) $=2^{10}$, 1 mega(M) $=2^{20}$,
1 giga(G)$=2^{30}$, 1 tera (T) $=2^{40}$

Rappresentazione dell'informazione

Per rappresentare M simboli sono necessari:

$$N \geq \log_2 M$$

Esempi:

$M=16=2^4$ $N=\log_2 16=4$

$M=256=2^8$ $N=\log_2 256=8$

$M=1024=2^{10}$ $N=\log_2 1024=10$

Sia $m=\log_2 M$

Se N = m la codifica si dice ***IRRIDONDANTE***

Se N > m la codifica si dice ***RIDONDANTE***

Rappresentazione posizionale dei numeri

Prof. G. Ascia

Si fa riferimento a una base ***b***

Valori possibili: 0, 1, .., b-1.

$$D_b: \quad c_{n-1}c_{n-2}\ldots c_1c_0\ c_{-1}c_{-2}\ldots c_{-m}$$

c_i appartiene all'insieme {0, 1, .., b-1}

$$D= c_{n-1}*(b)^{n-1}+c_{n-2}*(b)^{n-2}+..+c_1*(b)^1+c_0*(b)^0+$$
$$+ c_{-1}*(b)^{-1}+c_2*(b)^{-2}+..+ c_m*(b)^{-m}$$

Rappresentazione decimale

$b=10$, $c_i \in \{0,1, ..,9\}$

$(5209)_{10}= 5*(10)^3+2*(10)^2+0*(10)^1+9*(10)^0$

$(345,87)_{10}= 3*(10)^2+4*(10)^1+5*(10)^0+8*(10)^{-1}+7*(10)^{-2}$

Rappresentazione binaria

Prof. G. Ascia

$b=2,\ c_i \in \{0,1\}$

$(10110)_2=\mathbf{1}*(2)^4+\mathbf{0}*(2)^3+\mathbf{1}*(2)^2+\mathbf{1}*(2)^1+\mathbf{0}*(2)^0$

$(101)_2=\mathbf{1}*(2)^2+\mathbf{0}*(2)^1+\mathbf{1}*(2)^0$

Conversione da decimale a binario

- Per la parte intera:
  - ✓ Si procede per divisioni successive per due.
  - ✓ Le divisioni hanno termine quando il quoziente è nullo.
  - ✓ Il resto della i-esima divisione costituisce l'i-esimo bit a partire da destra.

19 |_2 1
9|_2 1
4|_2 0
2|_2 0
1|_2 1
0

$(19)_{10}=(10011)_2$

13 |_2 1
6|_2 0
3|_2 1
1|_2 1
0

$(13)_{10}=(01101)_2$

Conversione da decimale a binario

Prof. G. Ascia

Per la parte decimale si procede per prodotti per due.

La parte intera del i-esimo prodotto è il bit i-esimo a partire da sinistra.

Si prosegue con il prodotto dopo avere sottratto la parte intera.

Il procedimento ha termine non appena è stata raggiunta la precisione desiderata.

Esempio: $(0,225)_{10}=$

0,225*2=0,4500	
0,450*2=0,9	0
0,900*2=1,8	1
0,800*2=1,6	1
0,600*2=1,2	1
0,200*2=0,4	0

Rappresentazione dei numeri positivi

I numeri sono rappresentati in base 2

Con N bit i numeri interi positivi rappresentabili sono

$$0,1,..,2^N-1$$

N=num. di bit	Valori possibili	Num.max (2^N-1)
1	{0,1}	2-1=1
2	{0,1,2,3}	4-1=3
3	{0,1,..,6,7}	8-1=7
4	{0,1,..,14,15}	16-1=15
5	{0,1,..,30,31}	32-1=31
..	..	..
7	{0,1,..,126,127}	128-1=127
..	..	..

Rappresentazione in modulo e segno

- Il bit più significativo indica il segno
- Es.

$+5_{10} \rightarrow$

0	0	0	0	0	1	0	1

$-5_{10} \rightarrow$

1	0	0	0	0	1	0	1

Esistono due rappresentazioni dello zero:

$+0 \rightarrow$

0	0	0	0	0	0	0	0

$-0_{10} \rightarrow$

1	0	0	0	0	0	0	0

Rappresentazione in complemento a 2

I numeri positivi sono rappresentati dal loro modulo e segno zero.

I numeri negativi sono rappresentati dal complemento a 2 del corrispondente numero positivo.

$$\text{Rappr. }(X) = \begin{cases} |X| & \text{se} \quad 0 \leq |X| < 2^{N-1} \\ 2^N - |X| & \text{se} \quad -2^{N-1} \leq |X| < 0 \end{cases}$$

dove N è il numero di bit usati per la rappresentazione dei numeri.

Poiché $2^N=(2^N-1)+1$

$2^N-|X|= \{(2^N-1)- |X|\}+1$

Rappresentazione in complemento a 2

Calcolare $\{(2^N-1)- |X|\}$ equivale a invertire ogni bit di $|X|$

Es. N=7, $|X|= 11$

1	1	1	1	1	1	1	2^7-1
0	0	0	1	0	1	1	11
1	1	1	0	1	0	0	$(2^7-1)-11$

Il complemento a 2 di un numero si ottiene invertendo i bit del corrispondente intero positivo (complemento a 1) e aggiungendo 1.

Con N bit si possono rappresentarsi i numeri da -2^{N-1} a $2^{N-1}-1$

Es. N=8 $\rightarrow$ da $-2^{(8-1)}=-128$ a $2^{(8-1)}-1=127$

Rappresentazione in complemento a 2

Esempi.

$+5 \rightarrow$	**0**	0	0	0	0	1	0	1	
$\{2^8$-1-|5|$\}$ +	**1**	1	1	1	1	0	1	0	+
1 =	**0**	0	0	0	0	0	0	1	=
Rappr.(-5)	**1**	1	1	1	1	0	1	1	
$+19 \rightarrow$	**0**	0	0	1	0	0	1	1	
$\{2^8$-1-|19|$\}$+	**1**	1	1	0	1	1	0	0	+
1 =	**0**	0	0	0	0	0	0	1	=
Rappr.(-19)	**1**	1	1	0	1	1	0	1	

Rappresentazione in virgola fissa

Uso *m* bit e *n* bit per parte intera e frazionaria

Esempio (m=8, n=6, tot. 14 bit): $-123{,}21_{10}$

$-123_{10} = 10000101_2$

$0{,}21_{10} \approx 001101_2$

$-123{,}21_{10} \approx 10000101{,}001101_2$

Rappresentazione in virgola mobile

Prof. G. Ascia

Numero = $\pm M \times 2^{E}$

M=1.xxxxxxxx (Mantissa);

E = e+bias è l'esponente;

Es. bias =-127 $0 \leq e \leq 255$

Singola precisione

S	Esponente (8 bit)	Mantissa (23-bit)

Doppia precisione

Segno (1-bit)	Esponente (11 bit)		Mantissa (52-bit)	
0	1	11	12	63

Rappresentazione in virgola mobile

Es.1

$+1101.01_2 \quad = +1.10101 * 2^{011}$

$M=1.10101 \quad E=011=3=130-127 \quad e=130_{10}$

Rappresentazione dei caratteri

•- lettere, cifre numeriche;

•- simboli speciali, (parentesi, simboli operazionali, segni di interpunzione)

$$\begin{array}{cccccc} S_1 & S_2 & S_3 & S_4 & .. & S_N \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ C_1 & C_2 & C_3 & C_4 & .. & C_N \end{array}$$

Codici a 8 bit:

Ciascun simbolo S_i è rappresentato mediante una sequenza di 8 bit

$N=2^8=256$ SIMBOLI RAPPRESENTABILI

-ASCII (**A**merican **S**tandard **C**ode for **I**nformation **I**nterchange) -EBCDIC (**E**xtended **B**inary **C**oded **D**ecimal **I**nterchange **C**ode)

Esempio di codice ASCII

Prof. G. Ascia

Valore decimale	*Valore binario* **76543210**	*Carattere*
42	00101010	*
43	00101011	+
44	00101100	,
45	00101101	-
48	00110000	0
57	00111001	9
65	01000001	A
..		..
90	01011010	Z
97	01100001	a
..		..
122	01111011	z

Esempio di codice ASCII

Prof. G. Ascia

Codifichiamo

A=(B+C)*2

A	$\rightarrow 65_{10}=$	01000001
=	$\rightarrow 61_{10}=$	00111101
(	$\rightarrow 40_{10}=$	00101000
B	$\rightarrow 66_{10}=$	01000010
+	$\rightarrow 43_{10}=$	00101011
C	$\rightarrow 67_{10}=$	01000010
)	$\rightarrow 41_{10}=$	00101001
*	$\rightarrow 42_{10}=$	00101010