Lezione 1 – Codifiche binarie di valori numerici

Architettura degli elaboratori

Modulo 1 – Fondamenti architetturali

Unità didattica 2 – Rappresentazione binaria delle informazioni

Nello Scarabottolo

Università degli Studi di Milano - Ssri - CDL ONLINE

Le informazioni da rappresentare

Vogliamo dare una rappresentazione binaria ai vari modi in cui si presentano le informazioni che ci troviamo a trattare:

- quantità espresse da **NUMERI**;
- descrizioni testuali espresse mediante **CARATTERI** (lettere dell'alfabeto, simboli di interpunzione, ecc.);
- immagini costituite da **MATRICI BIDIMENSIONALI DI PIXEL** ("mosaici" con un certo numero di "tessere": per es. 1280x1024);
- suoni costituiti da **FORME D'ONDA** che riproducono le variazioni di pressione dell'aria;
- filmati costituiti da **SUONI** e sequenze di **IMMAGINI**

Cominciamo dai numeri interi positivi

In primo luogo, cerchiamo una rappresentazione che ci consenta di riutilizzare le regole di calcolo cui siamo abituati:

- ci rifacciamo alla **notazione decimale**, inventata in India, perfezionata dagli arabi e introdotta in Europa da Fibonacci;
- si tratta di una **notazione posizionale** basata sulle **10 cifre decimali** da 0 a 9;
- ogni cifra concorre al valore finale del numero con un **peso** dato dalla sua posizione nel numero: unità, decine, centinaia, migliaia, … decimi, centesimi, millesimi, …;
- il peso è una **potenza** del numero 10 (**base** della notazione).

Notazione posizionale pesata

Decimale

$(1273)_{10} = 1\times10^3 + 2\times10^2 + 7\times10^1 + 3\times10^0$

Binaria (due cifre: 0 e 1)

$(10010110)_2 = 1\times2^7 + 0\times2^6 + 0\times2^5 + 1\times2^4 +$
$+ 0\times2^3 + 1\times2^2 + 1\times2^1 + 0\times2^0 =$
$= (150)_{10}$

Esadecimale (10 cifre decimali e le lettere da A a F)

$(A59F)_{16} = 10\times16^3 + 5\times16^2 + 9\times16^1 + 15\times16^0 =$
$= (42399)_{10} =$
$= (1010\ 0101\ 1001\ 1111)_2 =$
$= $ xA59F

Quante cifre servono?

Decimale

- con c cifre rappresento tutti i numeri n tali che:

$$0 \le n \le 10^c - 1$$

- per rappresentare un numero n servono almeno:

$$c = \lceil \log_{10} n \rceil = \text{int.sup.}(\log_{10} n) \text{ cifre}$$

Binario

- con b bit rappresento tutti i numeri n tali che:

$$0 \le n \le 2^b - 1$$

- per rappresentare un numero n servono almeno:

$$b = \lceil \log_2 n \rceil = \text{int.sup.}(\log_2 n) \text{ bit}$$

I multipli binari

Una parentela fra le potenze delle due basi, che ha portato a un uso leggermente improprio dei nomi ...

Val.b	Nome	Simb.	Esp.b	Esp.d	Val.d
1.024	kilo	K	2^{10}	10^3	1.000
1.048.576	mega	M	2^{20}	10^6	1.000.000
1.073.741.824	giga	G	2^{30}	10^9	1.000.000.000
...	tera	T	2^{40}	10^{12}	...
...	peta	P	2^{50}	10^{15}	...
...	exa	E	2^{60}	10^{18}	...

Le operazioni si fanno nel modo usuale

Somma

	00110101_2	53_{10}
+	10110001_2	177_{10}
rip.	*0110001*	
=	11100110_2	230_{10}

Sottrazione

	10010001_2	145_{10}
–	00110000_2	48_{10}
pr.	*1100000*	
=	01100001_2	97_{10}

Prodotto

				1	1	0	1	$_2$	13_{10}
×				1	0	0	1	$_2$	9_{10}
				1	1	0	1		
			0	0	0	0			
		0	0	0	0				
	1	1	0	1					
= 0	1	1	1	0	1	0	1	$_2$	117_{10}

Conversione di base

Da binario a decimale, si **applica la notazione posizionale** come visto negli esempi precedenti.

Da decimale a binario, si **divide ripetutamente il numero per 2** fino ad arrivare a quoziente nullo, e si **prendono i resti a partire dall'ultimo**.

$(1017)_{10} = (1111111001)_2$

quoziente	resto
1017	
508	1
254	0
127	0
63	1
31	1
15	1
7	1
3	1
1	1
0	1

Tutto OK anche con i numeri frazionari

Decimale

$(127,3)_{10} = 1\times10^2 + 2\times10^1 + 7\times10^0 + 3\times10^{-1}$

Binaria

$(10010,110)_2 = 1\times2^4 + 0\times2^3 + 0\times2^2 + 1\times2^1 +$
$+ 0\times2^0 + 1\times2^{-1} + 1\times2^{-2} + 0\times2^{-3} =$
$= (18,75)_{10}$

Per avere però

- buona **precisione** (capacità di rappresentare numeri piccoli, o che differiscono per piccoli valori);
- buona **estensione** (capacità di rappresentare numeri grandi)

serve un numero esagerato di bit …

I numeri in virgola mobile

Il numero *n* viene rappresentato in notazione esponenziale:

$$n = m \times b^e$$

dove:

- ***m*** è la **mantissa**, costituita da un numero predefinito di cifre significative;
- ***b*** è la **base** (10 o 2);
- ***e*** è l'**esponente** (positivo o negativo)da dare alla base per definire il peso delle cifre della mantissa. Serve quindi a "muovere" la posizione della virgola rispetto alle cifre della mantissa, e consente di rappresentare valori molto piccoli e molto grandi

E se un numero è relativo ?

Nessun problema: il segno del numero relativo (+ o −) è un'informazione binaria …

- basta associare un bit al **segno**, con la convenzione:

  0 = numero positivo

  1 = numero negativo

  e usare gli altri bit per rappresentare il **modulo** del numero;
- con ***b*** bit (incluso il bit di segno) rappresentiamo tutti i numeri ***n*** tali che:

$$-(2^{b-1} - 1) \leq n \leq 2^{b-1} - 1$$

In sintesi…

La notazione posizionale pesata in base 2 ci consente di:

- rappresentare numeri interi e frazionari;
- utilizzare le regole dell'aritmetica per eseguire operazioni.

Se si vuole aumentare precisione ed estensione della rappresentazione senza richiedere numeri eccessivi di bit, si può ricorrere alla notazione in virgola mobile.

I numeri con segno possono essere rappresentati con la notazione in modulo e segno.

Chiusura

Fine della lezione