Architettura degli Elaboratori

Rappresentazione delle frazioni proprie
Aritmetica in binario

Barbara Masucci
UNIVERSITA DEGLI STUDI DI SALERNO
DIPARTIMENTO DI INFORMATICA
DIPARTIMENTO DI ECCELLENZA

Punto della situazione

- Abbiamo visto
  - Il sistema posizionale pesato, in particolare le rappresentazioni con basi 10, 2, 8 e 16
  - Gli algoritmi di conversione tra le varie basi per i numeri interi
- Oggi vediamo come rappresentare i numeri con la virgola

Sistema posizionale pesato

Ricordiamo che nel

Il peso della cifra cambia sulla base della posizione

Sistema Posizionale Pesato

$$N = (a_{n-1}\, a_{n-2} \ldots a_1\, a_0\, ,\, a_{-1}\, a_{-2} \ldots a_{-s})_b$$

$$(2425{,}295)_{10}$$

$$2 \times 10^3 + 4 \times 10^2 + 2 \times 10^1 + 5 \times 10^0 + 2 \times 10^{-1} + 9 \times 10^{-2} + 5 \times 10^{-3}$$

Rappresentazione decimale

Sistema posizionale pesato

La sequenza $a_{n-1}\,a_{n-2}\ldots a_1\,a_0\,,\,a_{-1}\,a_{-2}\ldots a_{-s}$

di simboli in $\{0, 1, \ldots, b-1\}$

rappresenta il numero

$$N = a_{n-1}\times b^{n-1} + a_{n-2}\times b^{n-2} + \ldots + a_1\times b^1 + a_0\times b^0$$

← parte intera

$$+\; a_{-1}\times b^{-1} + a_{-2}\times b^{-2} + \ldots + a_{-s}\times b^{-s}$$

← parte frazionaria o frazione propria (compresa tra 0 e 1)

$$= \sum_{i=-s}^{n-1} a_i b^i$$

Rappresentazione binaria

$(1011,110)_2$

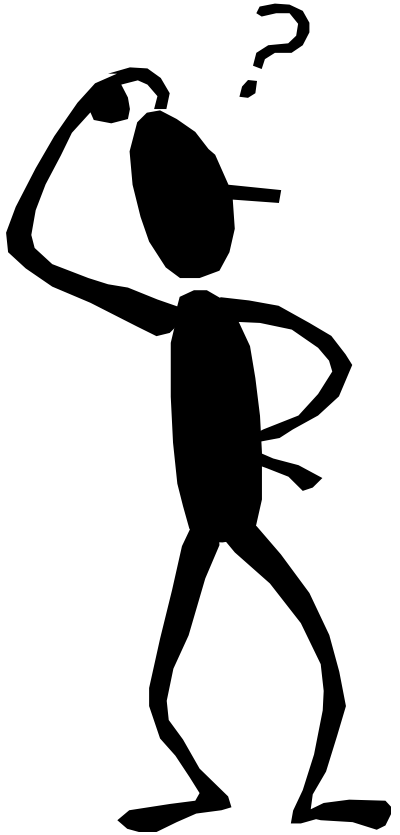
$N = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} + 0 \times 2^{-3}$

Rappresentazione binaria

$N = 11,75$

Conversione Decimale→ Binario

Come convertire una frazione propria da decimale in binario?



Conversione Decimale→ Binario

Come convertire una frazione propria da decimale in binario?

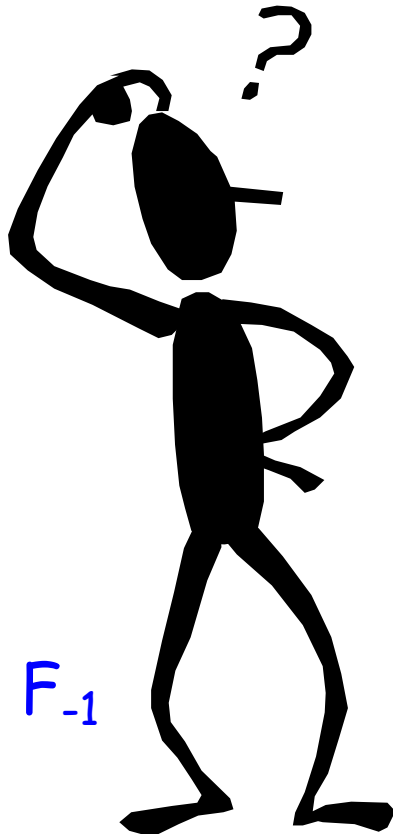
Data la frazione propria in decimale

$$F = a_{-1}\,2^{-1} + a_{-2}\,2^{-2} + \ldots + a_{-s}\,2^{-s}$$

Moltiplicando per 2 si ha

$$2F = a_{-1} + (a_{-2}\,2^{-1} + \ldots + a_{-s}\,2^{-(s-1)})$$
$$= a_{-1} + F_{-1}$$

Abbiamo isolato il bit a_{-1}
e ottenuto un'altra parte frazionaria F_{-1}



Conversione Decimale→ Binario

Possiamo ripetere il procedimento per isolare a_{-2}

$$F_{-1} = a_{-2}\, 2^{-1} + \ldots + a_{-s}\, 2^{-(s-1)}$$

Moltiplicando per 2 si ha

$$2F_{-1} = a_{-2} + (a_{-3}\, 2^{-1} + \ldots + a_{-s}\, 2^{-(s-2)})$$
$$= a_{-2} + F_{-2}$$

Abbiamo isolato il bit a_{-2}
e ottenuto un'altra parte frazionaria F_{-2}

Conversione Decimale→ Binario

Definiamo una sequenza di frazioni

$$F = F_0$$

$$2F_0 = a_{-1} + F_{-1}$$

$$2F_{-1} = a_{-2} + F_{-2}$$

...

$$2F_{-i} = a_{-(i+1)} + F_{-(i+1)}$$

Ad ogni passo, si ottiene un nuovo bit e una nuova parte frazionaria

Conversione Decimale→ Binario

Quando ci fermiamo?

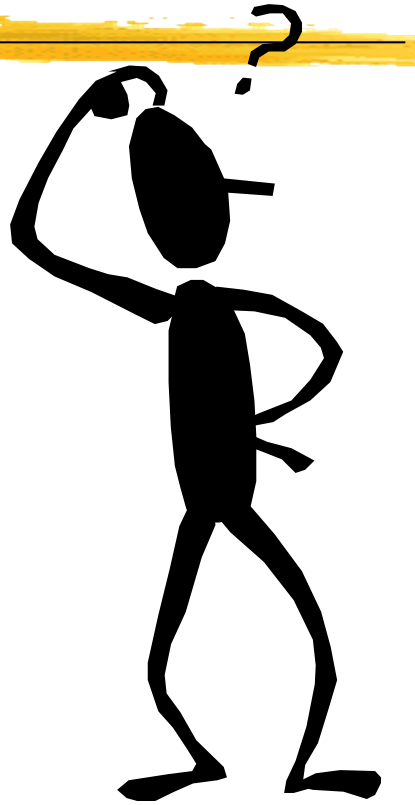
$F = F_0$

$2F_0 = a_{-1} + F_{-1}$

$2F_{-1} = a_{-2} + F_{-2}$

...

$2F_{-i} = a_{-(i+1)} + F_{-(i+1)}$



Quando troviamo un valore di F_{-i} uguale a zero

oppure

Quando abbiamo finito il numero di bit a disposizione per la rappresentazione binaria (approssimazione)

Esempio

$F=0{,}234_{10}$

Decimale->Binario

$2 \times 0{,}234 = 0 + 0{,}468$

$2 \times 0{,}468 = 0 + 0{,}936$

$2 \times 0{,}936 = 1 + 0{,}872$

$2 \times 0{,}872 = 1 + 0{,}744$

$2 \times 0{,}744 = 1 + 0{,}488$

$2 \times 0{,}488 = 0 + 0{,}976$

$2 \times 0{,}976 = 1 + 0{,}952$

.....

$0{,}234_{10}=(0{,}0011101\ldots)_2$

Esempio

$F=0,73_{10}$

Decimale->Binario

2 X 0,73 = 1 + 0,46

2 X 0,46 = 0 + 0,92

2 X 0,92 = 1 + 0,84

2 X 0,84 = 1 + 0,68

2 X 0,68 = 1 + 0,36

2 X 0,36 = 0 + 0,72

2 X 0,72 = 1 + 0,44

2 X 0,44 = 0 + 0,88

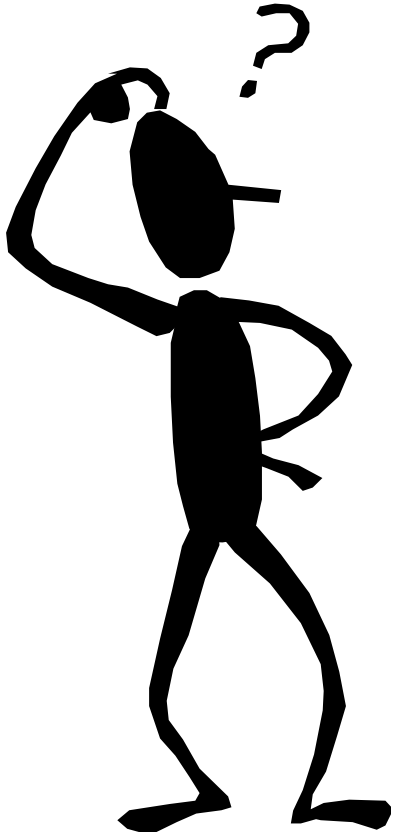
$0,73_{10}=(0,10111010\ldots)_2$

.....

In realtà $(0,10111010)_2$ rappresenta 0,7265625

Conversione Binario → Decimale

Come convertire una frazione propria da binario in decimale?



Conversione Binario → Decimale

Come convertire una frazione propria da binario in decimale?

Data la frazione propria in binario

$$F=(0,a_{-1}a_{-2}\ldots a_{-s})_2$$

Il valore in decimale è

$$F = a_{-1}\,2^{-1} + a_{-2}\,2^{-2} + \ldots + a_{-s}\,2^{-s}$$

Esempio: $F=(0,0011101)_2$ $s=7$

$$F = 2^{-3} + 2^{-4} + 2^{-5} + 2^{-7} = 0{,}2265625$$

Conversione Binario → Decimale

- Data la frazione propria in binario

$$F=(0,a_{-1}a_{-2}\ldots a_{-s})_2$$

Un algoritmo più efficiente consiste nell'utilizzare la sequenza di frazioni $F_{-s+1},\ldots, F_{-1}, F_0$, dove

$$F_{-s+1} = a_{-s}/2$$
$$F_{-s+2} = (a_{-s+1} + F_{-s+1})/2$$
$$F_{-s+3} = (a_{-s+2} + F_{-s+2})/2$$

...

$$F_{-1} = (a_{-2} + F_{-2})/2$$

$$F_0 = (a_{-1} + F_{-1})/2 \qquad F=F_0$$

Esempio

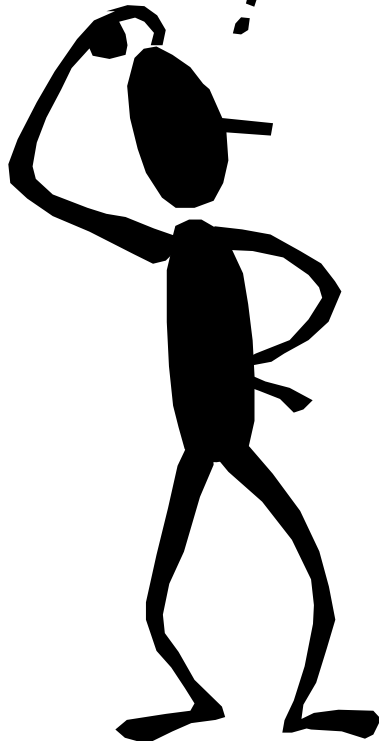
- Convertire in decimale il numero $(0,1101)_2$
  - $s=4$, $a_{-1}=1$, $a_{-2}=1$, $a_{-3}=0$, $a_{-4}=1$

$$F_{-3} = a_{-4}/2 = 1/2 = 0,5$$
$$F_{-2} = (a_{-3}+F_{-3})/2 = (0+0,5)/2 = 0,25$$
$$F_{-1} = (a_{-2}+F_{-2})/2 = (1+0,25)/2 = 1,25/2 = 0,625$$
$$F_0 = (a_{-1}+F_{-1})/2 = (1+0,625)/2 = 1,625/2 = 0,8125$$

$$(0,1101)_2 = 0,8125_{10}$$

Aritmetica in binario

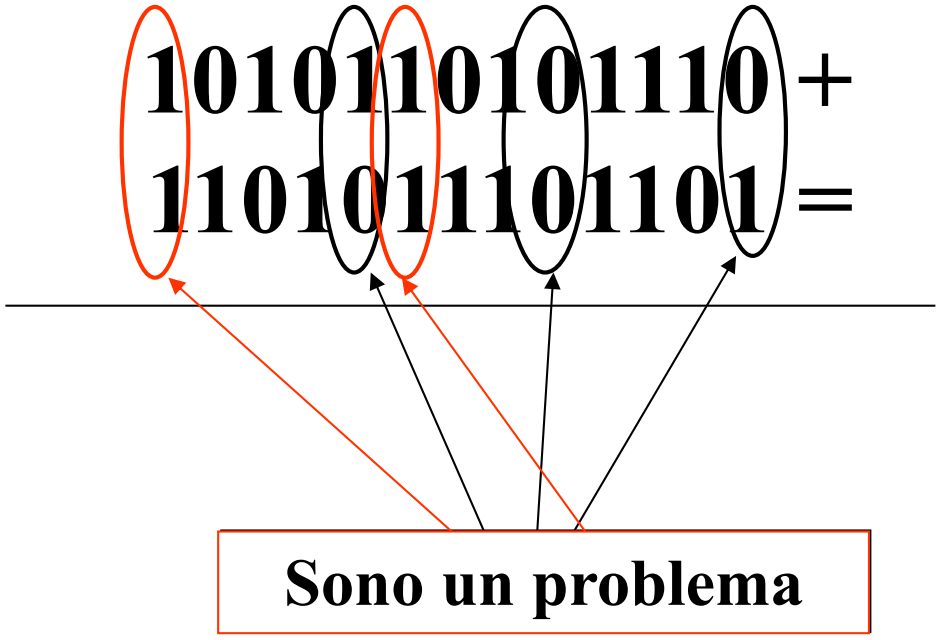
- Come eseguire operazioni aritmetiche su numeri espressi in notazione binaria?
  - Addizione
  - Sottrazione



Addizione in binario

- Si effettua bit per bit, portando il riporto (carry) alla cifra successiva
  - 0 + 0 = 0
  - 0 + 1 = 1
  - 1 + 0 = 1
  - 1 + 1 = 0 con il riporto di 1

La somma in binario



0+0=0

1+1=10

1+0=1

Regole

s	0	1
0	0	1
1	1	0

c	0	1
0	0	0
1	0	1

Somma (sum)

Riporto (carry)

Addizione

$c_n \quad c_{n-1} \quad c_i \quad c_2 \; c_1$

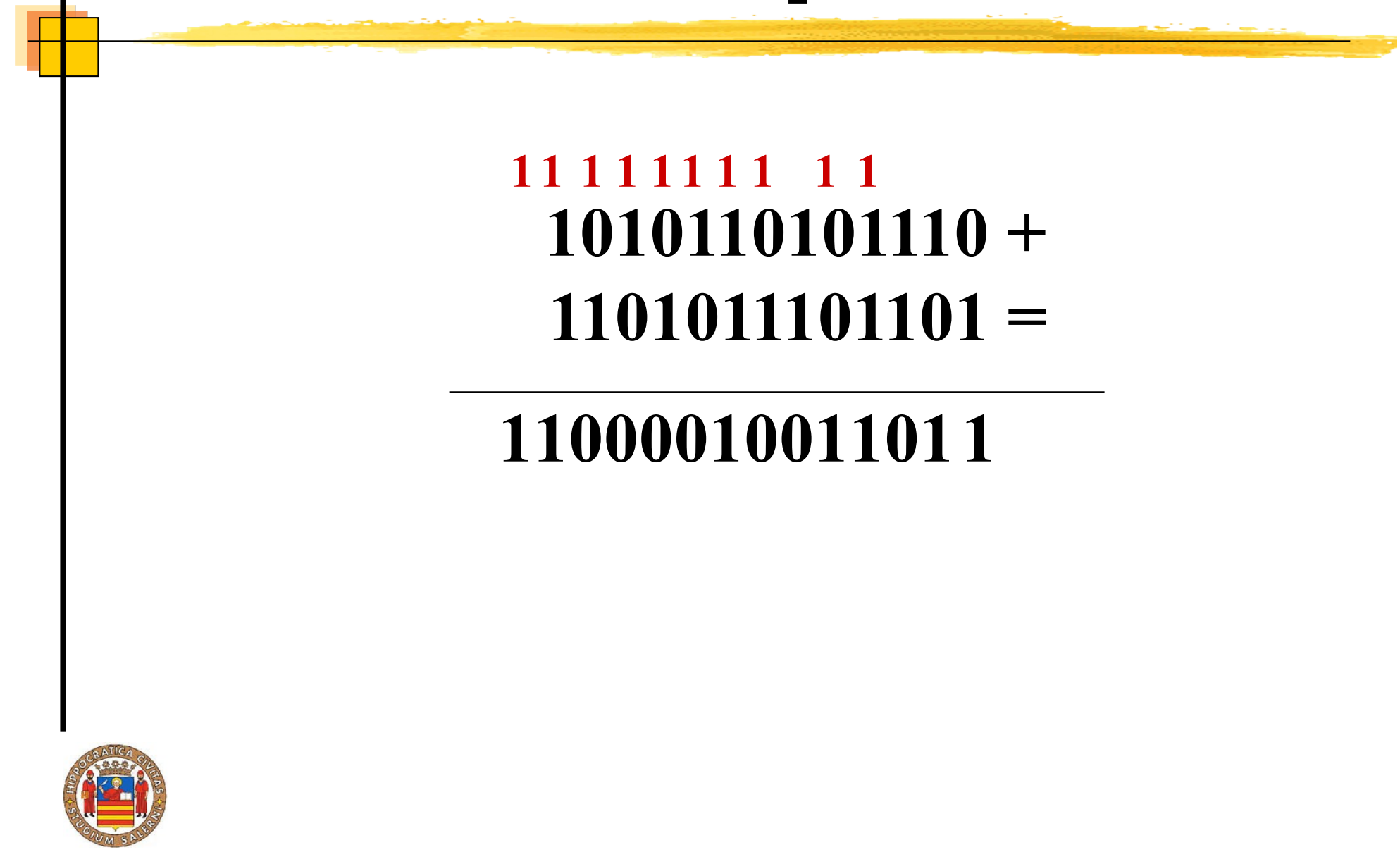
$a_{n-1}\ldots a_i \ldots a_2 a_1 a_0 +$
$b_{n-1}\ldots b_i \ldots b_2 b_1 b_0 =$

$s_n s_{n-1} \ldots s_i \ldots s_2 s_1 s_0$

s_i bit di somma

c_i bit di riporto

Esempio


1 1 1 1 1 1 1 1 1 1
1010110101110 +
1101011101101 =
11000010011011

Un caso particolare

$$\begin{array}{r} 1\ 1\ 1\ 1\ 1\ 1 \\ 1\ 1\ 1\ 1\ 1\ 1\ + \\ 1\ = \\ \hline 1\ 0\ 0\ 0\ 0\ 0\ 0 \end{array}$$

In generale:

$(1\ 1\ 1\ \ldots\ 1\ 1)_2 = 2^{n-1}+2^{n-2}+\ldots+2^1+2^0$

(n)

$(1\ 1\ 1\ \ldots\ 1\ 1)_2 + 1 = (1\ 0\ 0\ 0\ \ldots\ 0\ 0)_2 = 2^n$

(n) ($n+1$)

Da cui: $(1\ 1\ 1\ \ldots\ 1\ 1)_2 = 2^n - 1$

(n)

$2^{n-1}+ 2^{n-2}+ \ldots +2^1 + 2^0 = 2^n-1$

$$\sum_{i=0}^{n-1} 2^i = 2^n - 1$$

Un caso particolare

```
  1 1  1 1  1  1
    1 1 1 1 1  1 +
              1 =
-----------------
  1 0 0 0 0  0 0
```

$(111111)_2$ = 63 può essere rappresentato con **6 bit**

$(1000000)_2 = (111111)_2 + 1$ = 64 necessita di **7 bit**

La somma di due numeri rappresentati con n bit può essere un numero che necessita di n+1 bit per essere rappresentato (cioè un numero che supera $2^n - 1$)

Overflow

- Condizione che si verifica quando la somma di due numeri rappresentati con n bit supera $2^n - 1$
- Esempio:
  - sommiamo 5 e 11 (rappresentati con 4 bit)
  - $(5)_{10} = (0101)_2$
  - $(11)_{10} = (1011)_2$

$$\begin{array}{r} 0101 + \\ 1011 = \\ \hline 10000 \end{array}$$

Overflow!

Sottrazione in binario

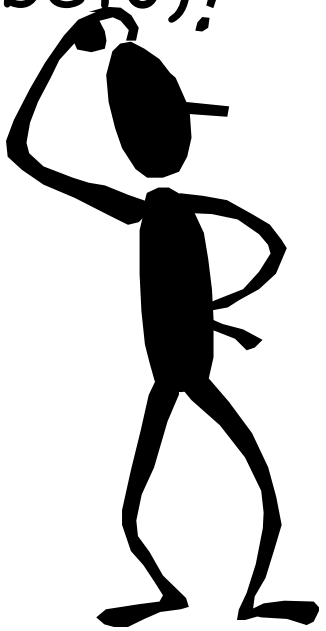
- Si effettua bit per bit, facendosi prestare, se necessario, una cifra dalla cifra precedente
  - 1 - 0 = 1
  - 1 - 1 = 0
  - 0 - 0 = 0
  - 0 - 1 = 1 (perché 10-1 =1)

Esempi

$$\begin{array}{r} 11101 - \\ 1110 = \\ \hline 01111 \end{array} \qquad \begin{array}{r} 1110 - \\ 1011 = \\ \hline 0011 \end{array}$$

Codifica di informazioni non numeriche

- La codifica dei numeri deve essere accurata perchè è necessario effettuare operazioni aritmetiche
- Come codificare le informazioni non numeriche (ad esempio i caratteri dell'alfabeto)?
  - Basta fissare una convenzione per poter riconoscere i dati



Codice ASCII

- American Standard Code for Information Interchange, pubblicato dall'ANSI nel 1968
- Codice alfanumerico, utilizza 7 bit (8 nella versione estesa) per codificare
  - Numeri, lettere maiuscole, lettere minuscole, segni di interpunzione, caratteri speciali
  - I primi 3 bit classificano il tipo di simbolo
    - 011 (0011 nella versione estesa): numeri
    - 100 e 101 (0100 e 0101 nella versione estesa): lettere maiuscole
    - 110 e 111 (0110 e 0111 nella versione estesa): lettere minuscole
    - 010 e 000 (0010 e 0000 nella versione estesa): punteggiatura e caratteri speciali

Codice ASCII Esteso

- Il codice ASCII consente di codificare 128 caratteri distinti, indicizzati da 0 a 127
- Il codice ASCII esteso consente di codificare 256 caratteri distinti
  - Aggiunge ai 128 caratteri precedenti altri caratteri speciali, matematici, grafici e di lingue straniere
  - I nuovi caratteri sono indicizzati da 128 a 255

Codice ASCII Esteso

ASCII control characters		
00	NULL	(Null character)
01	SOH	(Start of Header)
02	STX	(Start of Text)
03	ETX	(End of Text)
04	EOT	(End of Trans.)
05	ENQ	(Enquiry)
06	ACK	(Acknowledgement)
07	BEL	(Bell)
08	BS	(Backspace)
09	HT	(Horizontal Tab)
10	LF	(Line feed)
11	VT	(Vertical Tab)
12	FF	(Form feed)
13	CR	(Carriage return)
14	SO	(Shift Out)
15	SI	(Shift In)
16	DLE	(Data link escape)
17	DC1	(Device control 1)
18	DC2	(Device control 2)
19	DC3	(Device control 3)
20	DC4	(Device control 4)
21	NAK	(Negative acknowl.)
22	SYN	(Synchronous idle)
23	ETB	(End of trans. block)
24	CAN	(Cancel)
25	EM	(End of medium)
26	SUB	(Substitute)
27	ESC	(Escape)
28	FS	(File separator)
29	GS	(Group separator)
30	RS	(Record separator)
31	US	(Unit separator)
127	DEL	(Delete)

ASCII printable characters					
32	space	64	@	96	`
33	!	65	A	97	a
34	"	66	B	98	b
35	#	67	C	99	c
36	$	68	D	100	d
37	%	69	E	101	e
38	&	70	F	102	f
39	'	71	G	103	g
40	(	72	H	104	h
41	)	73	I	105	i
42	*	74	J	106	j
43	+	75	K	107	k
44	,	76	L	108	l
45	-	77	M	109	m
46	.	78	N	110	n
47	/	79	O	111	o
48	0	80	P	112	p
49	1	81	Q	113	q
50	2	82	R	114	r
51	3	83	S	115	s
52	4	84	T	116	t
53	5	85	U	117	u
54	6	86	V	118	v
55	7	87	W	119	w
56	8	88	X	120	x
57	9	89	Y	121	y
58	:	90	Z	122	z
59	;	91	[	123	{
60	<	92	\	124	|
61	=	93	]	125	}
62	>	94	^	126	~
63	?	95	_		

Extended ASCII characters							
128	Ç	160	á	192	└	224	Ó
129	ü	161	í	193	┴	225	ß
130	é	162	ó	194	┬	226	Ô
131	â	163	ú	195	├	227	Ò
132	ä	164	ñ	196	─	228	õ
133	à	165	Ñ	197	┼	229	Õ
134	å	166	ª	198	ã	230	µ
135	ç	167	º	199	Ã	231	þ
136	ê	168	¿	200	╚	232	Þ
137	ë	169	®	201	╔	233	Ú
138	è	170	¬	202	╩	234	Û
139	ï	171	½	203	╦	235	Ù
140	î	172	¼	204	╠	236	ý
141	ì	173	¡	205	═	237	Ý
142	Ä	174	«	206	╬	238	¯
143	Å	175	»	207	¤	239	´
144	É	176	░	208	ð	240	≡
145	æ	177	▒	209	Ð	241	±
146	Æ	178	▓	210	Ê	242	‗
147	ô	179	│	211	Ë	243	¾
148	ö	180	┤	212	È	244	¶
149	ò	181	Á	213	ı	245	§
150	û	182	Â	214	Í	246	÷
151	ù	183	À	215	Î	247	¸
152	ÿ	184	©	216	Ï	248	°
153	Ö	185	╣	217	┘	249	¨
154	Ü	186	║	218	┌	250	·
155	ø	187	╗	219	█	251	¹
156	£	188	╝	220	▄	252	³
157	Ø	189	¢	221	¦	253	²
158	×	190	¥	222	Ì	254	■
159	ƒ	191	┐	223	▀	255	nbsp

Codice ASCII

Usando questo codice ciascun carattere viene rappresentato con 8 bit (1 byte)

O	G	G	I		P	I	O	V	E
01001111	01000111	01000111	01001001	00100000	01010000	01001001	01001111	01010110	01000101

Riepilogo e riferimenti

- Rappresentazione di frazioni proprie in binario e conversioni
  - [P] par. 1.3.2
- Aritmetica in binario
  - [P] par. 1.4
- Codice ASCII
  - [P] par. 1.6.3 o [PH] par. 2.9