Cilc per tutti gli appunti (AUTOMAZIONE – TRATTAMENTI TERMICI ACCIAIO – SCIENZA delle COSTRUZIONI…)

e-mail per suggerimenti

3 SISTEMI DI NUMERAZIONE

3.1 Sistema decimale

Nel sistema decimale qualsiasi numero viene espresso dalla combinazione di 10 simboli (cifre) diversi, i quali assumono un significato e valore differente secondo la posizione occupata nel numero.

I dieci simboli adoperati sono:

0,1,2,3,4,5,6,7,8,9

Il simbolo *"0"* rappresenta la cifra nulla. I primi nove numeri, in ordine crescente, sono quindi: *0,1,2,3,4,5,6,7,8,9.* Un simbolo, nell'ordine scritto, *da sinistra a destra*, ha un valore pari ad un'unità in più di quello precedente. Il decimo numero, dopo il 9, lo si fa corrispondere ad una unità di ordine superiore, alla quale si dà il nome di *"decina"*. Il decimo numero viene espresso come una unità della prima potenza di dieci: $1 \cdot 10^1$; in tal modo l'unità viene pesata rispetto a 10^1

Il simbolo che esprime il decimo numero è composto dalle cifre 0 ,1 poste in ordine da destra verso sinistra , e si scrive: 10, con il significato di zero unità e una decina:

$$10 = 1 \cdot 10^1 + 0 \cdot 10^0$$

Si noti che le prime nove unità possono essere espresse secondo la potenza di ordine zero:

$$1 = 1 \cdot 10^0 \quad 2 \cdot 10^0 \quad 3 = 3 \cdot 10^0 \quad \ldots.$$

I numeri successivi al 10 si ottengono lasciando il numero 1 nella seconda posizione (decina) e cambiando successivamente la cifra nella prima, in sequenza da 1 a 9:

11,12,13,...,19.

ove :

$$11 = 1 \cdot 10^1 + 1 \cdot 10^0$$ una decina + una unità

....

$$19 = 1 \cdot 10^1 + 9 \cdot 10^0$$ una decina + nove unità

10^2	10^1	10^0
		1
		2
		3
		4
		5
		6
		7
		8
		9
	1	0
	1	1
....		
	1	9

Dopo il numero 19 si ottiene un numero corrispondente a una decina + dieci unità; ma 10 unità corrisponde ad una unità di ordine superiore : quindi a due decine

$$20 = 2 \cdot 10^1 + 0 \cdot 10^0$$

La numerazione continua con *21-22..*

Ogni *10* unità si passa da una di ordine "n" ad un'altra di ordine "*n+1*"- Così partendo da *20*, dopo dieci unità *(21,22...,29)* si passa al numero *30* e così via... Un qualsiasi numero può essere espresso con una   combinazione dei dieci simboli, il cui ordine di disposizione da destra verso sinistra rappresenta la potenza di dieci con la quale la cifra indicata si deve moltiplicare.

Così: $348 = 3 \cdot 10^2 + 4 \cdot 10^1 + 8 \cdot 10^0$

Si intende che la prima posizione a sinistra della virgola (l'unità) è quella di ordine zero; segue 10^1...

Come noto a   destra della virgola vengono posti i decimi 10^{-1} ,i centesimi 10^{-2}... che sono potenze negative della base 10

$$25.62 = 2 \cdot 10^1 + 5 \cdot 10^0 + 6 \cdot 10^{-1} + 2 \cdot 10^{-2}$$

Da quanto detto si può concludere che un numero nel sistema decimale è espresso da un polinomio ordinato secondo le potenze decrescenti della base *10*: i coefficienti delle potenze di *10* rappresentano le cifre significative del numero:

$$(N)_b = a_n \bullet 10^n + ... + a_i \bullet 10^i + ... + a_3 \bullet 10^3 + a_2 \bullet 10^2 + a_1 \bullet 10^1 + a_0 \bullet 10^0$$

Dove a_i può assumere i valori *0,1,2,3,4,5,6,7,8,9*

$$0 \le a_i \le 9$$

È da notare che i simboli che contraddistinguono il numero, oltre allo zero, sono dei simboli del tutto convenzionali: possono essere rappresentati anche da segni diversi da quelli usuali

❄

3.2 NUMERAZIONE IN BASE "b"

Un numero può essere espresso da un polinomio ordinato secondo le potenze decrescenti di una qualsiasi base *b* (numero naturale maggiore di 1):

$$(N)_b = a_n \cdot b^n + a_{n-1} \cdot b^{n-1} + ... + a_i \cdot b^i + ... + a_1 \cdot b^1 + a_0 \cdot b^0 + a_{-1} \cdot b^{-1} + a_{-2} \cdot b^{-2}$$

Il coefficiente generico a_i (cifra significativa) può essere rappresentato da *b* simboli diversi, dei quali uno indicherà l'elemento neutro *"0"*- Oltre lo zero , a_i assumerà "*b-1*" simboli diversi.

Così se la base è *5:*

$$b=5$$

oltre allo zero, vi saranno

b-1=5-1=4 simboli da dovere fissare.

Scelti questi tra i numeri naturali, a_i può assumere i caratteri:

0,1,2,3,4

con "*0*" elemento nullo.

------0-----

Consideriamo una numerazione con base 3

Con base $b=3$, oltre allo zero "0" si possono scegliere altri due simboli per le due unità di valore rispettivamente superiore come:

"0,1,2" – oppure – *"0,a b"* – oppure – *0 ,£,$, ecc...*

Si scelgano come simboli la prime tre cifre dei numeri naturali :

0,1,2.

Sono riportati nello schema i numeri espressi secondo la base 3 - (potenza di tre).

3^3	3^2	3^1	3^0	Numero naturale Base 10
			0	0
			1	1
			2	2
		1	0	3
		1	1	4
		1	2	5
		2	0	6
		2	1	7
		2	2	8
	1	0	0	9
	1	0	1	10
	1	0	2	11
	1	1	0	12
	1	1	1	13

I primi tre numeri *(coefficienti di 3^0)* sono: 0 1 2 con il significato di:

$$0 \cdot 3^0 = 0 \qquad 1 \cdot 3^0 = 1 \qquad 2 \cdot 3^0 = 2$$

Dopo il terzo numero si passa dalla unità di ordine inferiore 3^0 a quella di ordine superiore 3^1. Il numero successivo a due sarà: *10* con il significato di:

$$10 = 1 \cdot 3^1 + 0 \cdot 3^0 = 3$$

Seguiranno due numeri con unità crescenti nell'ordine inferiore 3^0 essi sono:

$$11 = 1 \cdot 3^1 + 1 \cdot 3^0 = 4$$

$$12 = 1 \cdot 3^1 + 2 \cdot 3^0 = 5$$

Quindi al terzo numero si passerà ad una unità di ordine superiore:

$$20 = 2 \cdot 3^1 + 0 \cdot 3^0 = 6$$

Segue:

$$21 = 2 \cdot 3^1 + 1 \cdot 3^0 = 7 \qquad 22 = 2 \cdot 3^1 + 2 \cdot 3^0 = 8$$

Dopo quest'ultimo si passa ad una unità di ordine superiore a 3^0, che, aggiungendosi alle due unità di ordine 3^1, farà scattare la cifra di una unità nell'ordine superiore 3^2 ...

Così il numero successivo a

$$22=2 \cdot 3^1+2 \cdot 3^0 =8$$

è:

$$100=1 \cdot 3^2+0 \cdot 3^1+0 \cdot 3^0=9$$

In generale si può affermare che effettuando una numerazione in base *"b"* occorrono *b-1* simboli significativi oltre lo *0*.

Nell'effettuare la numerazione, dopo *b* simboli *(compreso lo 0)* nella potenza b^0, scatta una unità *"1"* nella potenza superiore b^1, mentre inizia di nuovo da *0* la cifra nella potenza inferiore b^0; così, dopo altri *b* simboli nella potenza inferiore b^0 scatterà un'altra unità nella potenza superiore b^1 che, sommandosi alla precedente la porta a b^2, e così via...

Nel procedere della numerazione dopo *b* simboli si passa dalla potenza b^i alla potenza b^{i+1}

3 Trasformazione di un numero dal sistema decimale in uno in base b

3-1 Trasformazione di un numero intero

Sia *N* un numero intero del sistema decimale -

Si divida *N* per la base *b* nella quale si vuole trasformare il numero :

$$N:b$$

Effettuando la divisione si otterrà, in generale, un quoto Q_O e un resto che si indicherà con a_O $(0 \le a_O \le b)$

$$\begin{array}{r|l} N & b \\ \hline a_0 & Q_O \end{array}$$

$N:b=Q_O$ resto a_O ; per cui risulterà:

$$N=Q_O \cdot b+a_O \qquad (3.1.1)$$

Se il quoto *Qo* risulta maggiore della base *b:* $Q_O \ge b$, allora si effettui la divisione tra Q_O e *b:*

$$\begin{array}{r|l} Q_0 & b \\ \hline a_1 & Q_1 \end{array}$$

$$Q_o : b = Q_1 \quad \text{resto} \quad a_1 \quad \text{risulterà:} \qquad Q_o = Q_1 \cdot b + a_1 \quad (3.1.2)$$

Sostituendo nella (3.1) si ottiene:

$$N = (Q_1 + a_1) \cdot b + a_o$$

$$N = Q_1 \cdot b^2 + a_1 \cdot b + a_o \qquad (3.1.3)$$

Sia ancora il quoto $Q_1 \geq b$. Si divida questo ancora per b**.** Si ha:

$$Q_1 : b = Q_2 \text{ resto } a_2; \quad \text{da cui si ha:} \quad \Rightarrow \quad Q_1 = Q_2 \cdot b + a_2$$

Sostituendo nella (3.3) si ha:

$$N = (Q_2 \cdot b + a_2) \cdot b^2 + a_1 \cdot b + a_o$$

$$N = Q_2 \cdot b^3 + a_2 \cdot b^2 + a_1 \cdot b + a_o \quad (3.1.4)$$

Si procederà fino a che risulta $Q_{n-1} < b$

$$N = Q_{n-1} \cdot b^n + a_{n-1} \cdot b^{n-1} ....+ a_2 \cdot b^2 + a_1 \cdot b + a_o \qquad (3.1.5)$$

Allora effettuando ancora la divisione tra il. quoto ottenuto Q_{n-1} e b si avrà:

$$\begin{array}{r|l} Q_{n-1} & b \\ \hline a_n = Q_{n-1} & 0 \end{array}$$

$$Q_{n-1} : Q_{n-1} = 0 \text{ resto: } Q_{n-1} = a_n$$

Il quoto della divisione è zero *"0"* mentre il suo resto a_n risulta uguale al dividendo Q_{n-1}. Quindi risulterà:

$$Q_{n-1} = b \cdot 0 + a_n$$

Sostituendo nella (3.1.5) si ottiene:

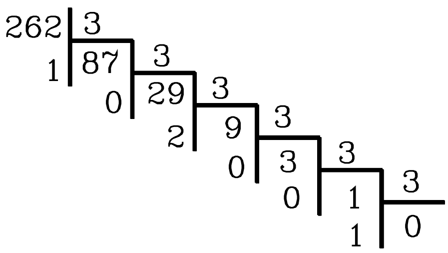
$$(N)_b = a_n \bullet b^n + a_{n-1} \bullet b^{n-1} + ... + a_i \cdot b^i + ... + a_2 \bullet b^2 + a_1 \bullet b + a_o \quad (3.1.6)$$

Dove $a_o, a_1, a_2, ...a_i,...., a_n$ sono i resti delle divisioni successive per la base b.

Il numero N espresso nella base b sarà indicato dai coefficienti del polinomio $(N)_b$, posti in ordine secondo la potenza decrescente da sinistra verso destra

Si debba trasformare il numero *262* in base *10* in un numero in base *b=3*

fig.1.15



Dividendo per 3 i successivi quozienti, si ottengono i resti che vengono posti in ordine da destra verso sinistra.
Dalle successive divisioni si ottiene:

1 0 0 2 0 1

$262 : 3 = 87 \quad resto\ 1 \; => \; 262 = 87 \cdot 3 + 1$

$87 : 3 = 29 \quad resto\ 0 \; => \; 87 = 29 \cdot 3 + 0 \; => \; 262 = (29 \cdot 3 + 0) \cdot 3 + 1 \cdot 3^0 \; \Rightarrow$

$$\Rightarrow \quad 262 = 29 \cdot 3^2 + 0 \cdot 3^1 + 1 \cdot 3^0$$

$29 : 3 = 9 \quad resto\ 2 \; \rightarrow \; 29 = 9 \cdot 3 + 2 \; \rightarrow \; 262 = (9 \cdot 3 + 2) \cdot 3^2 + 0 \cdot 3^1 + 1 \cdot 3^0$

$$\Rightarrow \quad 262 = 9 \cdot 3^3 + 2 \cdot 3^2 + 0 \cdot 3^1 + 1 \cdot 3^0$$

$9:3=3 \qquad resto\ 0 \;\Rightarrow\; 9=3\cdot 3+0 \;\Rightarrow\; 262=(3\cdot 3+0)\cdot 3^3+2\cdot 3^2+0\cdot 3^1+1\cdot 3^0$

$\Rightarrow\; 262=3\cdot 3^4+0\cdot 3^3+2\cdot 3^2+0\cdot 3^1+1^0$

$3:3=1 \qquad resto\ 0 \;\Rightarrow\; 3=3\cdot 1+0 \;\Rightarrow\; 262=(1\cdot 3+0)\cdot 3^4+2\cdot 3^2+0\cdot 3^1+1\cdot 3^0$

$\Rightarrow\; 262=1\cdot 3^5+0\cdot 3^4+0\cdot 3^3+2\cdot 3^2+0\cdot 3^1+1\cdot 3^0$

$1:3=0 \qquad resto\ 1 \;\Rightarrow\; 1=0\cdot 3+1$ quindi:

$262=(0\cdot 3+1)\cdot 3^5+0\cdot 3^4+0\cdot 3^3+2\cdot 3^2+0\cdot 3+1\cdot 3$

$262=0\cdot 3^6+1\cdot 3^5+0\cdot 3^4+0\cdot 3^3+2\cdot 3^2+0\cdot 3^1+1\cdot 3^0$

$0:3=0 \qquad resto\ 0$

Le potenze di 3 superiori a 5 hanno tutti coefficienti nulli. Nel procedimento di divisione ci si ferma all'ultimo resto inferiore a b.

Il numero 262 espresso nella base b=3 sarà:

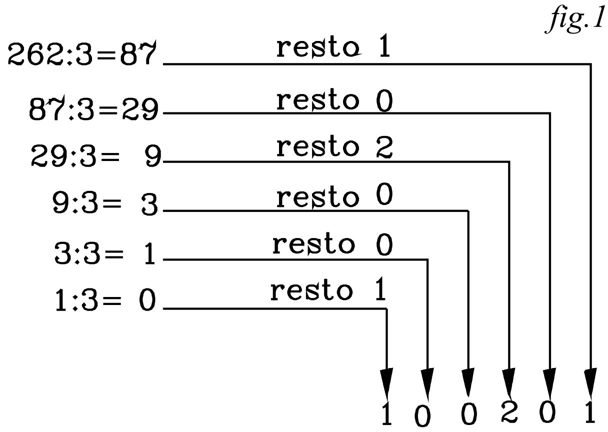
$$262=1\cdot 3^5+0\cdot 3^4+0\cdot 3^3+2\cdot 3^2+0\cdot 3+1\cdot 3^0$$

Il numero si indica omettendo le potenze di 3

$$262=100201$$

Le cifre del numero in base $b=3$ sono i resti delle divisioni multiple. In ordine crescente, da destra verso sinistra, si scrive:

fig.1.16



Numero $(N)_3=100201$

3.2 Trasformazione della parte decimale di un numero espresso in base 10 in un numero espresso in base b

Sia $0,D$ il numero da trasformare; avendo indicato con D la parte decimale a destra della virgola.

Si moltiplichi il numero $0,D$ per la base b.

Dalla moltiplicazione si otterrà un numero costituito da una parte intera I_1 (*che può essere anche nulla*) e una parte decimale restante D_1 (*che può essere nulla*)

$$0,D\cdot b = I_1,D_1 = I_1 + 0,D_1 \qquad (3.2.1)$$

da cui, ricavando *0,D* si ha:

$$0,D = \frac{I_1}{b} + 0,D_1 \cdot \frac{1}{b} \quad (3.2.2)$$

❄❄

Per esempio il numero decimale sia: ***0,D=0,342***

La nuova base nella quale si vuole esprimere il numero dato sia: b=4

Moltiplicando il numero per 4 si ha:

$$0,342x4=1,368$$

$$0,342x4 = 1 + 0,368 \quad (a)$$

dove la parte intera è $I_1=1$ *e quella decimale* $D_1=0,368$

ricavando dalla espressione (a) il numero dato: ***0,342****, si ottiene:*

$$0,342 = \frac{1}{4} + 0,368 \cdot \frac{1}{4}$$

❄❄

Si moltiplichi la parte decimale $0,D_1$ ottenuta nella (3.2.2) ancora per *b, ottenevo un nuovo numero decimale con parte intera* I_2 *e decimale* D_2*:* I_2,D_2

$$0,D_1 \, x \, b = I_2,D_2 = I_2 + 0,D_2$$

da cui:

$$0,D_1 = \frac{I_2}{b} + \frac{0,D_2}{b} \quad (3.2.3)$$

Sostituendo la (3.2.3) nella (3.2.2) si ha:

$$0,D = \frac{I_1}{b} + \left(\frac{I_2}{b} + \frac{0,D_2}{b}\right) \cdot \frac{1}{b}$$

$$0,D = \frac{I_1}{b} + \frac{I_2}{b^2} + \frac{0,D_2}{b^2}$$

In generale, salvo casi particolari , l'operazione può continuare fino all'infinito. Ci si ferma ad un termine n^{mo} **,** dipendente dalla approssimazione che si vuole ottenere.

Considerando le espressioni già ottenute è facile esprimere l'n^{mo} termine

$$0,D = \frac{I_1}{b} + \frac{I_2}{b^2} + \frac{I_3}{b^3} + \cdots + \frac{I_n}{b^n} + \frac{0,D_n}{b^n} \quad (3.2.4)$$

Se l'ultimo termine $0,D_n \neq 0$ allora il rapporto:

$$\varepsilon = \frac{0,D_n}{b^n} \quad (3.2.5)$$

rappresenta l'errore che si commette nell'esprimere il numero in base *b* limitato ad n termini dopo la virgola.

Quindi, con l'approssimazione anzidetta, il numero decimale in base *b* è espresso da un polinomio in detta base, ordinato secondo le potenze decrescenti negative, fino all'ordine *n,* dipendente dalla approssimazione che si vuole ottenere

$$0,D = I_1 \bullet b^{-1} + I_2 \bullet b^{-2} + I_3 \bullet b^{-3} + ... + I_n \bullet b^{-n} \quad (3.2.6)$$

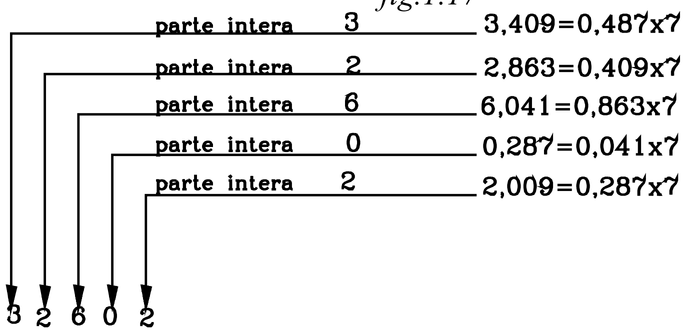
Dove $I_1, I_2, I_3, ..., I_n$ sono gli interi ottenuti dalle moltiplicazioni successive per la base b delle parti decimali. Sottintendendo le potenze negative di b in ordine decrescente da sinistra verso destra, e scrivendo solamente i coefficienti, il numero decimale in base *b* è espresso nella forma:

$$0,D = I_1 I_2 I_3 .....I_n$$

Si debba cosi' trasformare 0,487 dal sistema decimale ad un altro sistema in base $b=7$

In pratica si opera secondo il seguente schema:

fig.1.17



Come precedentemente detto, si moltiplica il numero *0,487* in base 10 per la base *7 (0,487x7=3,409)*, ottenendo la parte intera 3 e la parte decimale *0,409*. La parte intera 3 dà la prima cifra significativa. La parte decimale *0,409* si moltiplica per la base *7* , ottenendo la parte intera *2* che rappresenta la seconda cifra significativa. E così via ....

Il numero decimale approssimato di 4 cifre che si ottiene è:

$$0,32602$$

con il significato di

$$0,32602 = 3 \cdot 7^{-1} + 2 \cdot 7^{-2} + 6 \cdot 7^{-3} + 0 \cdot 7^{-4} + 2 \cdot 7^{-5} = 0,486999464 \cong 0,487$$

L'errore effettuato risulta di:

$$\varepsilon = 0,487 - 0,486999464 = 0,000000536$$

❄

3.3 SOMMA DI NUMERI IN BASE b

Si considerino per semplicità due numeri interi:

$$\text{Siano:} \begin{cases} (N)_b = \sum_{i=0}^{n} a_i \cdot b^i & \textit{primo numero} \\ (N_1)_b = \sum_{i=0}^{m} c_i \cdot b^i & \textit{secondo numero} \end{cases}$$

Sia $n \geq m$. In forma esplicita i due numeri si possono esprimere:

$$(N)_b = a_n \cdot b^n + \ldots + a_m \cdot b^m + a_{m-1} \cdot b^{m-1} + \ldots a_i \cdot b^i + \ldots + a_1 \cdot b^1 + a_0 \cdot b^0$$
$$(N_1)_b = c_m \cdot b^m + c_{m-1} \cdot b^{m-1} + \ldots c_i \cdot b^i + \ldots + c_1 \cdot b^1 + c_0 \cdot b^0$$

La somma $(N)_b+(N_1)_b$ si ottiene sommando i termini simili. Considerando così due termini simili generici $a_i \cdot b^i$, $c_i \cdot b^i$, la somma darà:

$$a_i \cdot b^i + c_i \cdot b^i = (a_i + c_i) \cdot b^i \quad (3.3.1)$$

Si possono distinguere due casi:

1°- $a_i+c_i < b$ -

In tal caso la cifra della i^{ma} posizione sarà quella corrispondente al risultato $\mathbf{a_i+c_i}$

Esempio *base* $b=5$

$(N)_b$	2310	+
$(N_1)_b$	2104	=
$(N)_b+(N_1)_b=$	4414	

❄

2°- $a_i+c_i \geq b$

La somma dei termini simili è uguale o supera b. Si supponga che $a_i+c_i>b$

Allora si divida la somma (a_i+c_i) per la base b: $a_i+c_i : b$

$$\begin{array}{r|l} (a_i+c_i) & b \\ \hline R_i & Q_i \end{array}$$

Si ottiene un quoto Q_i ed un resto R_i ; per cui si può scrivere che il dividendo (a_i+c_i) è uguale al divisore b per il quoto Q_i più il resto R_i:

$$(a_i + c_i) = Q_i \cdot b + R_i$$

Così il termine i^{mo} della somma si scriverà:

$$(a_i + c_i) \cdot b^i = (Q_i \cdot b + R_i) \cdot b^i$$

$$(a_i + c_i) \cdot b^i = Q_i \cdot b^{i+1} + R_i \cdot b^i \quad (3.3.2)$$

Ne viene che: nella posizione i^{ma} del numero risultante dalla somma, si pone, come cifra, il resto della divisione, mentre il quoto va sommato alla cifra di ordine superiore successivo $i+1$.

Così se con la base $b=5$ la somma di due termini risulta $a_i+c_i=7$, allora si effettua la divisione:

7:5=1 con resto di *2*.

In tal caso si procede nella seguente maniera:

1- Si pongono due unità, pari al resto "$R_i=2$", nella posizione i^{ma}

2- Il quoto della divisione "$Q_i=1$" 'verrà sommato alla cifra della posizione successiva $i+1$

❄

Esempio:

Siano dati in base b=5 i tre numeri da sommare: *2343 , 134 , 43*

$$\begin{array}{lr} \textit{riporto}\rightarrow & 122 \\ & \text{------} \\ & 2343\ + \\ & 134\ + \\ & 43\ = \\ & \text{-------} \\ & 3130 \end{array}$$

Somma delle cifre della I colonna: $3+4+3=10$; $10:5=2$ resto$=0$
0 è la cifra risultante della I colonna - si riporta 2 sulla II

Somma delle cifre della II colonna: $2+4+3+4=13$; $13:5=2$ resto$=3$
3 è la cifra risultante della II colonna - si riporta 2 sulla III

Somma delle cifre della III colonna: $2+3+1=6$; $6:5=1$ resto 1
1 è la cifra risultante della *III* colonna - si riporta 1 sulla *IV*

Somma delle cifre della IV colonna: $1+2=3$
La somma risultante "*3*" non supera la base *5*, questa rappresenta la cifra della *IV* colonna

❄

3.4 SOTTRAZIONE DI DUE NUMERI IN BASE b

Si abbiano due numeri in base *b*, rispettivamente di ordine ne *m* . Essi si possono esprimere con i polinomi caratteristici

$$\text{Siano:}\begin{cases} (N)_b = \sum_{i=0}^{n} a_i \cdot b^i & \textit{primo numero} \\ (N_1)_b = \sum_{i=0}^{m} c_i \cdot b^i & \textit{secondo numero} \end{cases}$$

Sia $n \geq m$. In forma esplicita i due numeri si possono esprimere:

$$\begin{aligned} (N)_b &= a_n \cdot b^n + \ldots + a_m \cdot b^m + a_{m-1} \cdot b^{m-1} + \ldots a_i \cdot b^i + \ldots + a_1 \cdot b^1 + a_0 \cdot b^0 \\ (N_1)_b &= \qquad\qquad\qquad c_m \cdot b^m + c_{m-1} \cdot b^{m-1} + \ldots c_i \cdot b^i + \ldots + c_1 \cdot b^1 + c_0 \cdot b^0 \end{aligned}$$

La differenza $(N)_b-(N_1)_b$ si ottiene sottraendo i termini simili

Considerando due termini generici come $a_i \cdot b_i$, $c_i \cdot b_i$ si ha:

$$a_i \cdot b^i - c_i \cdot b^i = (a_i - c_i) \cdot b^i \quad (3.4.1)$$

Si presentano tre casi

$$a_i > c_i \quad a_i = c_i \quad a_i < c_i$$

$a_i > c_i$ Il risultato della differenza è una cifra compresa tra 0 e la base b "$0 < a_i - c_i < b$", esso, in tal caso, si pone come cifra da assegnare alla colonna i^{ma}

$a_i = c_i$ Allora $a_i - c_i = 0$ e si assegnerà la cifra 0 alla colonna i^{ma}

$a_i < c_i$ La differenza $a_i - c_i$ non è effettuabile nel campo dei numeri naturali.

Consideriamo quest'ultimo caso in cui risulti $a_i < c_i$. alla differenza $a_i - c_i$ sommiamo e sottraiamo **b**. Vale l'uguaglianza :

$$a_i - c_i = a_i - c_i + b - b \quad (3.4.2)$$

Per cui, sostituendo la (3.4.2) nella (3.4.1), il coefficiente della potenza i^{ma} si può scrivere nella forma:

$$(a_i - c_i)\cdot b^i = (a_i - c_i + b - b)\cdot b^i = (a_i + b - c_i)\cdot b^i - b^{i+1}$$

Il termine b^{i+1} andrà sottratto alla cifra di ordine $i+1$.

Così nella differenza dei termini simili dei polinomi caratteristici dei due numeri in base b compariranno :

$$\ldots(a_{i+1} - c_{i+1})\cdot b^{i+1} + (a_i + b - c_i)\cdot b^i - 1\cdot b^{i+1}\ldots\ldots =$$
$$= \ldots(a_{i+1} - 1 - c_{i+1})\cdot b^{i+1} + (a_i + b - c_i)\cdot b^i$$

Quando nella posizione i^{ma} il minuendo a_i è inferiore al sottraendo c_i, allora si prende in prestito una unità dalla cifra successiva di ordine superiore *(i+1)*:

$$1\cdot b^{i+1}$$

Tale unità corrisponde a b unità nell'ordine inferiore :

$$1\cdot b^{i+1} = b\cdot b^i$$

Si somma così al minuendo i^{mo}: "a_i" le b unità prese in prestito dalla cifra di ordine superiore: (a_i+b) e al risultato verrà sottratto il sottraendo $c_i<b$.

Si ottiene così nella posizione i^{ma} la cifra risultante

$$(a_i + b - c_i) \qquad ossia: \qquad (a_i + b - c_i)\cdot b^i.$$

Avendo preso in prestito una unità dalla cifra di ordine superiore i+1, nella posizione $i+1^{ma}$ occorre togliere detta unità al minuendo a_{i+1} e poi sottrarre il sottraendo c_{i+1} :

$$(a_{i+1} - 1 - c_{i+1})\cdot b^{i+1}$$

❄

Esempio:

Si debba effettuare la differenza dei due numeri in base b=5 : (10341-2023)

```
riporto      -1  5     -1  5
            ----------
             1  0  3  4  1  -     ⇒in base 10     1·5⁴+0·5³+3·5²+4·5¹+1·5⁰     →721-

                2  0  2  3  =                          2·5³+0·5²+2·5¹+3·5⁰     →263=
             ----------------                    -----------------------------    ----
                3  3  1  3                             3·5³+3·5²+1·5¹+3·5⁰     →458
```

Posti i due numeri in colonna ,ordinati da destra verso sinistra, si effettua la differenza delle cifre di una stessa colonna

- Se la cifra del minuendo è superiore o uguale al sottraendo: si effettua la sottrazione, ottenendo la cifra risultante della colonna considerata.
- Se il minuendo è inferiore al sottraendo, si prende in prestito 1 unità dalla cifra di ordine superiore, che corrisponde a b=5 unità nell'ordine inferiore. In tal caso si riportano 5 unità sulla colonna nella quale si sta effettuando la sottrazione e "-1" unità sulla colonna di ordine superiore.

Differenza della prima colonna a destra:

Si deve effettuare la differenza *1-3* - Occorre prendere in prestito una unità dalla cifra della seconda colonna e si porrà su di essa il riporto *-1*; mentre sulla prima colonna si pongono le *5* unità corrispondenti alla unità di ordine superiore preso in prestito.
Si effettua quindi la seguente operazione: al minuendo **1** si sommano le **5** unità di prestito e si sottrae il sottraendo **-3**, ottenendo la cifra della prima colonna

1+5-3=3 (cifra risultante della I colonna)

Differenza della seconda colonna:
Al minuendo *4* si sottrae la cifra *1* data in prestito e, al risultato, si sottrae il minuendo *2*

4-1-2=1 (cifra risultante della seconda colonna)

e così via...

❄

3.5 MOLTIPLICAZIONE

Siano dati due numeri (per semplicità interi) in base b

$$(N_1)_b = a_n \cdot b^n + \ldots + a_m \cdot b^m + a_{m-1} \cdot b^{m-1} + \ldots a_i \cdot b^i + \ldots + a_1 \cdot b^1 + a_0 \cdot b^0$$
$$(N_2)_b = c_m \cdot b^m + c_{m-1} \cdot b^{m-1} + \ldots c_i \cdot b^i + \ldots + c_1 \cdot b^1 + c_0 \cdot b^0$$

Si debbano moltiplicare i due numeri $(N_1)_b \times (N_2)_b$.

Per fare ciò occorre moltiplicare i due polinomi caratteristici.
Si pongano nella stessa colonna i termini aventi la stessa potenza della base *b*

$$(N_1)_b = a_n \cdot b^n + a_{n-1} \cdot b^{n-1} + \cdots + a_m b^m + \cdots + a_i \cdot b^i + \cdots + a_2 \cdot b^2 + a_1 \cdot b^1 + a_0 \cdot b^0$$

$$(N_2)_b = c_m \cdot b^m + \cdots + c_i \cdot b^i + \cdots + c_2 \cdot b^2 + c_1 \cdot b^1 + c_0 \cdot b^0$$

$$a_n c_0 b^n b^0 + a_{-1} c_0 b^n b^0 + \cdots + a_m c_0 b^m b^0 + \ldots + a_i c_0 b^i b^0 + \ldots + a_2 c_0 b^2 b^0 + a_1 c_0 b^1 b^0$$
$$a_n c_1 b^n b^1 + a_n c_0 b^n b^1 + a_{n-1} c_1 b^n b^1 + \cdots + a_m c_1 b^m b^1 + \cdots + a_i c_1 b^i b^1 + \ldots + a_2 c_1 b^2 b^1$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$
$$a_n c_m b^n b^m + \ldots\ldots\ldots\ldots\ldots\ldots + a_m c_0 b^m b^0$$

$$h_{n+m} b^{n+m} + \ldots\ldots\ldots\ldots\ldots\ldots + h_m b^m + \cdots + h_i b^i + \cdots + h_2 b^2 + h_1 b^1 + h_0 b^0$$

La moltiplicazione va effettuata moltiplicando ciascun termine del primo polinomio per ciascuno del secondo.

Così , partendo dal termine $c_0 . b^0$ del secondo polinomio, questo si moltiplica per ciascuno del primo ; successivamente si prende il *II* termine $c_1 \cdot b^1$ del secondo polinomio lo si moltiplica per tutti i termini del primo.

Le moltiplicazioni effettuate vengono scritte in righe successive, ponendo sulla stessa colonna i termini con uguale potenza.

Ogni riga ,ottenuta dalla moltiplicazione di un termine del secondo polinomio per tutti i termini del primo , risulta spostata di una colonna verso sinistra rispetto alla precedente, essendo il termine moltiplicatore di un grado superiore rispetto a quello precedente.

Si consideri il termine generico del polinomio "$h_i \cdot b^i$ ", nel quale h_i rappresenta la cifra significativa dell'*i^mo^* posto del risultato del prodotto . Si possono presentare due casi.

1°- $h_i<b$ Allora il coefficiente h_i resta come cifra significativa dell'i^{mo} posto

2° - $h_i \geq b$ In tal caso si effettua la divisione $h_i:b$; si ottiene il quoto q_i e resto r_i

$$\begin{array}{r|l} h_i & b \\ \hline r_i & q_i \end{array}$$

Per cui si può scrivere :

$$h_i = q_i \cdot b + r_i$$

Così il termine generico del polinomio si può porre nella forma:

$$h_i \cdot b^i = (q_i \cdot b^i + r_i) \cdot b^i = q_i \cdot b^{i+1} + r_i \cdot b^i$$

Dalla espressione si nota che: il resto r_i è la cifra significativa dell' i^{mo} posto ; mentre il quoto q_i va ad aggiungersi alla cifra successiva di ordine superiore $i+1$

Esempi:
Siano dati i due numeri in base ***b=5*** *: 3413 , 4*
si moltiplichi 3413x4

	base 5		base 10		
riporto	*3 3 1 2*				
	2 4 1 3 x 4	⇒	$3 \cdot 5^3 + 4 \cdot 5^2 + 1 \cdot 5^1 + 3 \cdot 5^0$	→	483
	4 =		$4 \cdot 5^0$	→	4
	------------------		---------------------------------		------
	3 0 2 1 2		$3 \cdot 5^4 + 0 \cdot 5^3 + 2 \cdot 5^2 + 1 \cdot 5^1 + 2 \cdot 5^0$		1932

- *Si inizia con 3x4=12*
  Il risultato *12* della moltiplicazione è superiore alla base $b=5$
  Si effettua la divisione tra il risultato 12 e la base b=5 e risulta:
  Il quoto $q_0=12:5=2$, il resto $r_0=2$: si pone $r_0=2$ come cifra significativa della prima colonna di destra e si riporta il quoto $q_1=2$ in cima alla seconda colonna

- *Si moltiplica 4x1=4*
  Al risultato *4* della moltiplicazione si somma il riporto *2,* in cima alla colonna: *4+2=6*
  Il risultato 6 è maggiore della base $b=5$ quindi : $q_1=6:5=1$,resto r_1=1 .Si pone come cifra significativa della seconda colonna il resto $r_1=1$; mentre il quoto $q_2=1$ si riporta in cima alla terza colonna.

- Si moltiplica 4x4=16

Al risultato *16* della moltiplicazione si somma il riporto 1*,* in cima alla colonna: *16+1=17*
Il risultato 17 è maggiore della base b=5 quindi : q_2=17:5=3 ,resto r_2=2 .Si pone come cifra significativa della terza colonna il resto r_2=2 ; mentre il quoto q_2=3 si riporta in cima alla quarta colonna.

- Si moltiplica *3x4=12*
Al risultato *12* della moltiplicazione si somma il riporto *3,* in cima alla colonna: *12+3=15*
Il risultato 15 è uguale alla base *b=5* quindi : $q_3=15:5=3$,resto $r_3=0$ .Si pone come cifra significativa della terza colonna il resto $r_3=0$; mentre il quoto $q_3=3$ si riporta in cima alla quarta colonna.

- Il riporto *3* sulla quinta colonna determina la cifra significativa di essa.

Il risultato della moltiplicazione è *3 0 2 1 2*

Così si effettui la moltiplicazione dei due numeri in base 5 : 1342x 342

	base 5								base 10
riporto	*1342x3* →	*1*	*2*	*2*	*1*				
riporto	*1342x4* →		*1*	*3*	*3*	*1*			
riporto	*1342x2* →				*1*	*1*			
					1	3	4	2	x
						3	4	2	=
riporto	somme			*1*	*1*	*1*			
prodotto	1342x2 →				3	2	3	4	
prodotto	1342x4 →		1	2	0	2	3		
prodotto	1342x3 →	1	0	1	3	1			
		1	1	4	2	1	1	4	

Si noti che:

$$1\ 3\ 4\ 2 \;=\; 1\cdot 5^3 + 3\cdot 5^2 + 4\cdot 5^1 + 2\cdot 5^0 = \;222$$

$$3\ 4\ 2 \;=\; 3\cdot 5^2 + 4\cdot 5^1 + 2\cdot 5^0 \;=\; 97$$

$$1\ 1\ 4\ 2\ 1\ 1\ 4 \;=\; 1\cdot 5^6 + 1\cdot 5^5 + 4\cdot 5^4 + 2\cdot 5^3 + 1\cdot 5^2 + 1\cdot 5^1 + 4\cdot 5^0 = \;21534 = 222x97$$

Per effettuare l'operazione occorre lasciare uno spazio, limitato da una linea tratteggiata, al di sopra dei due numeri da moltiplicare, posti in colonna, per scrivere i riporti delle moltiplicazione dei singoli numeri.

Un altro spazio occorre lasciare tra i due numeri posti in colonna e i risultati delle moltiplicazioni, ove indicare i riporti delle loro somme

- Si inizia a moltiplicare la prima cifra 2 di 342 per la prima cifra 2 del fattore 1342:

2x2=4

essendo il prodotto inferiore alla base 5 si scrive direttamente il risultato senza riporto in corrispondenza della prima colonna a destra rispondente a 5^0

- Si moltiplica poi la prima cifra 2 del fattore 342 per la seconda cifra 4 del fattore 1342 della colonna 5^1

$$2x4=8$$

Essendo 8 maggiore della base 5, si divide 8 per detta base:

$$8:5=1 \text{ con resto } 3$$

si pone il resto 3 come cifra significativa sulla colonna 5^1 mentre il quoto 1 si pone come riporto sulla riga al di sopra della linea tratteggiata sulla colonna successiva 5^2

- Si moltiplica poi la prima cifra 2 del fattore 342 per la terza cifra “3” del fattore 1342 della colonna 5^2

$$2x3=6$$

al prodotto va sommato il riporto del prodotto precedente posto sulla riga al di sopra della linea tratteggiata

$$6+1=7$$

Essendo 7 maggiore della base 5, si divide 7 per detta base:

$$7:5=1 \text{ con resto } 2$$

si pone il resto 2 come cifra significativa sulla colonna 5^2 mentre il quoto 1 si pone come riporto sulla riga sopra la linea tratteggiata nella colonna successiva 5^3

- Si continua poi a moltiplicare la prima cifra 2 di 342 per tutte le altre cifre di 1342

- Si passa poi a moltiplicare la seconda cifra 4 del fattore 342 per ciascuna cifra di 1342
  Così si inizia con

$$4x2=8$$

Essendo 8 maggiore della base 5, si divide 8 per detta base:

$$8:5=1 \text{ con resto } 3$$

la prima cifra significativa è il resto 3 che va posta in una riga al di sotto dei prodotti precedenti e una colonna a sinistra della prima cifra di detti prodotti: colonna di 5^1 .
Il quoto “1” va riportato in una riga al disopra dei precedenti riporti e nella colonna successiva a sinistra 5^2 .

- E cosi via…

- Alla fine si ottengono su righe successive, dall’alto verso il basso, spostate ciascuna rispetto all’altra di una colonna a sinistra i prodotti di ciascuna cifra di 342 per le cifre di 1342

- Il risultato del prodotto dei due numeri si ottiene sommando le cifre poste su una stessa colonna di tutte le righe ottenute. I riporti della somma sono posti sopra la riga tratteggiata

3.6 DIVISIONE

Siano dati i due numeri in base b≥2 :

$$(N_1)_b = a_n \cdot b^n + \ldots + a_m \cdot b^m + a_{m-1} \cdot b^{m-1} + \ldots a_i \cdot b^i + \ldots + a_1 \cdot b^1 + a_0 \cdot b^0$$

$$(N_2)_b = c_m \cdot b^m + c_{m-1} \cdot b^{m-1} + \ldots c_i \cdot b^i + \ldots + c_1 \cdot b^1 + c_0 \cdot b^0$$

La divisione $(N_1)_b$=$(N_1)_b$:$(N_2)_b$ è quel numero $(N)_b$ che moltiplicato per $(N_2)_b$ dà come risultato $(N_1)_b$.

Si opera come nella divisione normale

Si prenda ad esempio la divisione dei due numeri in base *b=7: "42356 : 625"*

fig.1.18

4 2 3 5 6	6 2 5
3 4 3 6	
− 6 6 6 6	4 5
4 3 6 4	
− 3 0 2	

- Si pongano i numeri nello schema della divisione, nel quale: il dividendo è posto nella parte superiore a sinistra il divisore a destra, il quoto sotto il divisore e il resto sotto il dividendo

- Le prime tre cifre *423* a sinistra del dividendo sono inferiore al divisore *(423<625)*; occorre quindi prendere in considerazione le prime quattro cifre *4235*
  Si può ora dividere *4235* per il divisore *625.*

- Per far ciò, si ricerca quel numero inferiore alla base *b=7* ,tale che, moltiplicato per 625 dia come risultato il numero più prossimo per difetto al dividendo *4235 (la moltiplicazione è tra numeri in base 7)*

Si provi con i numeri *5* e *4*

```
4 1 3
-----------------
  6 2 5 x                                  6 2 5 x
      5 =                                      4 =
________                                 ______
4 3 6 4     risulta 4364>4235          3 4 3 6     inferiore a 4235
```

- La moltiplicazione *625x5* dà come risultato *4364* maggiore del dividendo
  Per la moltiplicazione si procede come si è esposto

5x5=25		si divide per la base 7 prima cifra significativa 4	25:7=3 resto 4 riporto quoto 3
5x2=10	più riporto 10+3=13	si divide per la base 7 prima cifra significativa 6	13:7=1 resto 6 riporto quoto 1

5x6=30	più riporto 30+1=31	si divide per la base 7 prima cifra significativa 3	31:7=4 resto 3 riporto quoto 4

l'ultimo riporto determina l'ultima cifra significativa: 0x5+4=4

- La moltiplicazione *625x4* dà come risultato 4235 inferiore del dividendo: quindi la prima cifra del quoto è 4
- Si riporti il risultato della moltiplicazione per *4 (3436)* sotto le quattro cifre del dividendo e si effettui la sottrazione *4235-3436 = 466*
- Si abbassa il 6 del dividendo, ponendolo accanto a *466* e si ottiene *4666*
- Si ricerca il numero che moltiplicato per *625* approssimi per difetto 4666 : tale numero è 5 ; infatti *625x5=4364*
- Quindi la seconda cifra del quoto è *5*.
- Si riporta *4364* sotto il dividendo *4666* e si effettua la sottrazione *4666-4364 = 302* che rappresenta il resto della divisione

4 SISTEMA BINARIO

Il binario è il sistema di numerazione in base è: $b=2$
Occorrono solo due simboli. Usualmente vengono scelti i caratteri : *0 1*

Nel sistema binario valgono ovviamente le stesse regole studiate per la base *b* qualsiasi.

Per trasformare un numero da decimale a binario si effettuano le divisioni multiple per *2*: i resti delle divisioni danno le cifre significative del numero binario cercato.

Per numeri non elevati si può effettuare la trasformazione a mente.

Esempio
Si debba trasformare il numero 14 in numero binario. Conviene procedere nella seguente maniera:

1- Si determini la potenza di due più prossima a *14* per difetto: essa è $2^3=8$

2- Si effettui la differenza $14-8=6$

3- Si determini la potenza di due più prossima a 6 per difetto: essa è $2^2=4$

4- Si effettui la differenza $6-4=2$

5- *2* coincide con 2^1

6- Il numero è completamente scomposto secondo le potenze di 2. Non si ha nessuna cifra con la potenza 2^0

Il numero *14* è composto dalle seguenti potenze di due

$$1\cdot 2^3 + 1\cdot 2^2 + 1\cdot 2^1 + 0\cdot 2^0$$

$$14 \xrightarrow{corrisponde} 1110$$

4.1 Somma

0+0=0

1+0=1

0+1=1

1+1=0   con riporto di 1 nell'ordine superiore

Al numero binario si può associare una interpretazione fisica, come di un parametro che può assumere due stati opposti.

Ad esempio lo stato dei contatti di un relè o la condizione di accensione e spegnimento di una lampada:

Relè eccitato	⇒	contatto chiuso	= 1
Relè diseccitato	⇒	contatto aperto	= 0
Lampada accasa	⇒		= 1
Lampada spenta	⇒		= 0

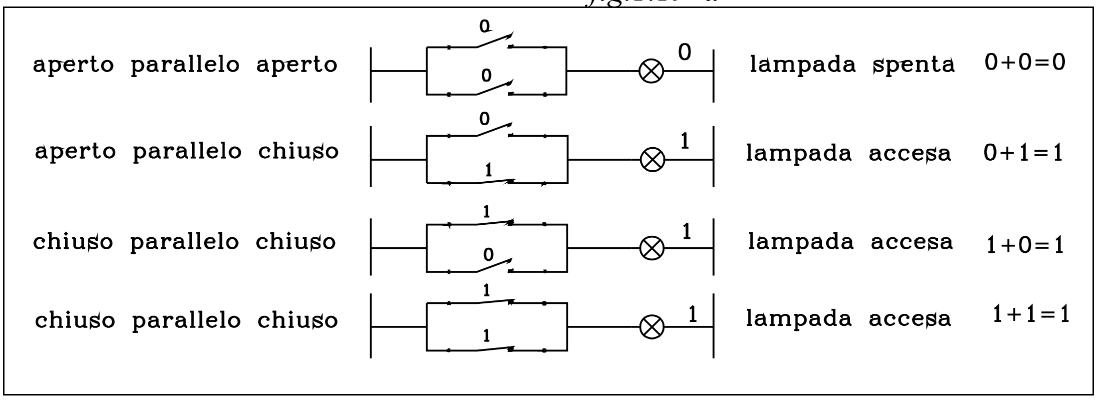
La somma di numeri binari corrisponde al parallelo dei contatti

Infatti, si considerino due contatti in parallelo e si studi l'effetto che il loro stato può avere sull'accensione di un lampada

Si indichi con 1 lo stato che fa passare la corrente da sinistra verso destra e fa accendere la lampadina, con 0 lo stato che interrompe la corrente

Il simbolo "+" indichi il parallelo. Si verifica che:

fig.1.19 a



Come si può constatare vi è una differenza nell'interpretazione della somma 1+1 nel funzionamento circuitale del parallelo dei contatti rispetto alla somma binaria:

1+1=1   nella interpretazione circuitale

1+1=0   nella somma binaria *(ma con riporto di 1 nella cifra con potenza di ordine superiore)*

4.2 Prodotto

0·0=0

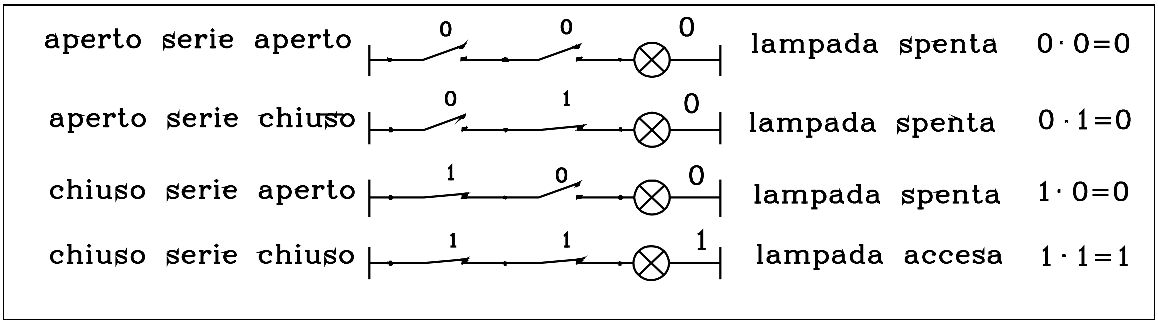
0·1=0

1·0=0

1·1=1

Il prodotto binario corrisponde alla serie di contatti. Infatti, considerando l'effetto sulla accensione della lampada in relazione alla combinazione degli stadi che i contatti in serie possono assumere. Si verifica.:

fig.1.19 b



Differenza
0-0=0
1-0=1
1-1=0
0-1=1   con riporto di -1 nella cifra di ordine superiore

Esempi

❄

Somma

	1 1 1 1 1	*riporti*
	- - - - - - - - - - -	
25 +	1 1 0 0 1 +	
12 +	1 1 0 0 +	
3 =	1 1 =	
40	1 0 1 0 0 0	

Iniziando dalla prima colonna a destra si ha:

Prima colonna   *1+0=1* il risultato va sommato a *1: 1+1 =0* con riporto di *1* sulla colonna successiva a sinistra.

Seconda colonna   Si parte dal *riporto 1* e si sommano ad esso le altre cifre della seconda colonna : *1+0+0+1=0* con riporto di 1 sulla colonna successiva a sinistra.

....

Quinta colonna   Il riporto 1 viene sommato ad 1 della quinta colonna: 1+1=0 con riporto di 1 sulla 6° colonna successiva a sinistra.

Sesta colonna   Si ha solamente il riporto 1

❄

Moltiplicazione

```
                      1  1  1  1  1                     riporto delle somme
                     - - - - - - - - - - - - - - - - -
   25  x                          1  1  0  0  1  x
   13  =                             1  1  0  1  =
  ____                   ______________________
   75                             1  1  0  0  1
  25                           0  0  0  0  0
  _____                     1  1  0  0  1
  325                    1  1  0  0  1
                       ________________________
                       1  0  1  0  0  0  1  0  1
```

- Posti i due numeri in colonna, si inizia a moltiplicare la prima cifra a destra *"1"* del moltiplicatore per tutte le cifre del moltiplicando *11001*.

  11001x1= 11001

  Il risultato si pone al di sotto del moltiplicatore con le cifre in colonna partendo dalla cifra a destra.
- Si moltiplica quindi la seconda cifra *0* del moltiplicatore per il moltiplicando, ottenendo tutte le cifre uguale a *0*.

  11001x0 = 00000

  Le cifre ottenute si pongono spostate di una colonna a sinistra rispetto al risultato della moltiplicazione precedente.

```
   1  1  0  0  1
0  0  0  0  0
```

- Si procede nella stessa maniera per tutte le cifre del moltiplicatore; al termine si sommano le cifre delle colonne ottenute

Nella riga superiore vi sono i riporti delle somme.

❄

Sottrazione

```
                          -1 -1 -1 -1               riporti negativi
                         - - - - - - - - - -
   24  -                  1  1  0  0  0  -
   13  =                     1  1  0  1  =
   __                    _____________
   11                        1  0  1  1
```

Si inizia la sottrazione dalla prima colonna a destra

Prima colonna   Si ha : *0 - 1 = 1* con riporto di *-1* sulla colonna successiva a sinistra. Si ottiene così 1 come risultato sulla prima colonna

Seconda colonna   Si considera la cifra *0* del minuendo e la si sottrae al riporto: *0 - 1 = 1* con riporto di *-1* sulla colonna successiva a sinistra. Il risultato 1 ottenuto va sottratto allo *0* del sottraendo *1 - 0 = 1*; si ottiene così la cifra *1* come risultato sulla seconda colonna.

Terza colonna Si considera la cifra 0 del minuendo e la si sottrae al riporto: $0 - 1 = 1$ con riporto di -1 sulla colonna successiva a sinistra. Il risultato 1 ottenuto va sottratto allo cifra 1 del sottraendo: $1 - 1 = 0$; si ottiene così 0 come risultato della terza colonna.

Quarta colonna Si considera la cifra *1* del minuendo e la si sottrae al riporto: $1 - 1 = 0$ Questo risultato si sottrae alla cifra *1* del sottraendo: $0 - 1 = 1$ con riporto di -*1* sulla colonna successiva a sinistra. Si ottiene così 1 come risultato della quarta colonna.

Quinta colonna Alla cifra 1 del minuendo si sottrae il riporto: $1 - 1 = 0$. Il risultato *0* ottenuto non viene posto sulla quinta colonna, in quanto non è significativo

❄

4.3 Divisione

Si effettua normalmente come si è proceduto per le divisioni in base *b*.

È evidente che nel caso di divisione di due numeri in base binaria le cifre del quoto sono solamente *0* oppure *1*. Da questa osservazione ne viene che i risultati delle divisioni parziali sono o *0* oppure *1*

fig.1.20

```
 3 7 | 7                 1 0 0 1 0 1 | 1 1 1
     |-----                          |----------
   2 | 5                   1 1 1     | 1 0 1
     |                   0 0 1 0 0   |
                             0 0 0   |
                           1 0 0 1   |
                             1 1 1   |
                               1 0   |
```

- Si considerino le prime tre cifre *100*. Il numero che esse esprimono è minore di *111*. Occorre quindi considerare le prime quattro cifre *1001* da dividere per i divisore *111*: si otterrà come quoto 1 con un resto.
- Per ottenere il resto si moltiplica il quoto *1* per il divisore *111*

111x1=111

e il risultato ottenuto si sottrae alle quattro cifre 1001 del dividendo :

```
  -1 -1
- - - - - - - - -
  1  0  0  1
     1  1  1
 ____________
  0  0  1  0
```

Gli zeri davanti alla cifra 1 non sono significativi, per cui il resto della divisione *1001:111* risulta *10*

- Si abbassa la cifra del dividendo, successiva alla quarta: essa è *0*; questa viene posta dopo le due cifre significative del resto *10* e si ottiene *100* . Il numero ottenuto lo si divide per *111:* vi sta *0* volte.

- Si moltiplica *0* per *111 : 111x0=0* e il risultato va sottratto al dividendo 100 ottenendo come resto il numero stesso *(100)*.

- Si abbassa 1 che posto dopo le tre cifre del resto *100* ottenendo *1001*. Il numero ottenuto lo si divide per *111*: vi sta 1 volte con un resto.

- Per ottenere il resto si moltiplica il quoto *1* per il divisore *111*

  111x1=111

  e il risultato ottenuto si sottrae alle quattro cifre 1001 del dividendo :

$$\begin{array}{cccc}
 & -1 & -1 & \\
\text{- - -} & \text{- - -} & \text{- - -} & \text{- - -} \\
1 & 0 & 0 & 1 \\
 & 1 & 1 & 1 \\
\hline
0 & 0 & 1 & 0
\end{array}$$

Se non si procede oltre nella divisione si ottiene:

Quoto della divisione $101 = 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 5$

Resto della divisione $10 = 1 \cdot 2^1 + 0 \cdot 2^0 = 2$

4.1 Trasformazione Binario - Ottale Ottale - Binario

Il sistema di numerazione ottale ha base *8*. Dalla sua conformazione, che si presenta come la terza potenza di due : $8=2^3$, ne deriva la proprietà che ogni gruppo di tre cifre nel sistema binario corrisponde ad una cifra in ottale.

Per trasformare in ottale un numero binario, basta suddividere questo in gruppi di tre cifre, a partire dalla prima a destra, e sostituire ad ogni gruppo il corrispondente valore nella base decimale, la quale corrisponde al numero in ottale, in quanto un qualsiasi gruppo di tre cifre binarie ha un valore inferiore a *8 (massimo 7⇒111)*

È evidente che gli eventuali zeri alla sinistra delle cifre significative del gruppo non sono influenti sul valore numerico: Per esempio $011 = 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 3$

$$011 \xrightarrow{\text{corrisponde}} 11$$

Si abbia così il numero binario:

$$11101001 = 1 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 233$$

Si divida il numero in gruppi di tre cifre a partire da destra e si sostituisca ad ogni gruppo il corrispondente valore in base 10, che corrisponde anche al numero in ottale.

fig.1.21

$$\underbrace{0\ 1\ 1}_{3} \quad \underbrace{1\ 0\ 1}_{5} \quad \underbrace{0\ 0\ 1}_{1}$$

Si sostituisce:

001 ⇒ $2^0 = 1$

101 ⇒ $1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 5$

011 $\Rightarrow$ $1 \cdot 2^1 + 1 \cdot 2^0 = 3$

Il numero in ottale risulta costituito dalle cifre:

351

con il significato: $351 = 3 \cdot 8^2 + 5 \cdot 8^1 + 1 \cdot 8^0 = 233$

Si ottiene lo stesso risultato del numero binario di partenza.

Vale la trasformazione inversa.

> Per trasformare un numero da ottale in binario, basta sostituire , procedendo da destra verso sinistra, ad ogni cifra dell'ottale il corrispondente numero binario, completando, se occorre, il gruppo di tre cifre con zeri alla sinistra delle cifre significative del numero binario.

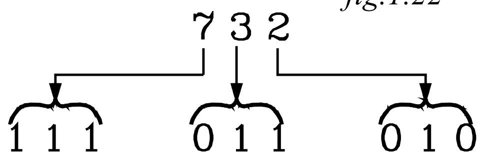
Si debba trasformare il numero ottale *732* nel corrispondente numero binario.

Il numero ottale *732* corrisponde al seguente numero in base 10:

$$732 = 7 \cdot 8^2 + 3 \cdot 8^1 + 2 \cdot 8^0 = 474$$

Si sostituiscano alle cifre *2 , 3, 7* i corrispondenti numeri binari secondo il seguente schema:

fig.1.22



Si sostituisce:

$2 \Rightarrow 10 = 1 \cdot 2^1 + 0 \cdot 2^0 \Rightarrow$ 010

$3 \Rightarrow 11 = 1 \cdot 2^1 + 1 \cdot 2^0 \Rightarrow$ 011

$7 \Rightarrow 111 = 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 \Rightarrow$ 111

Il numero binario corrispondente all'ottale *732* risulta:

111011010

Infatti detto numero, riportato in base 10 dà come risultato lo stesso dell'ottale:

$$111011010 = 1 \cdot 2^8 + 1 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 474$$

4.2 Trasformazione Binario – esadecimale; Esadecimale - Binario

Un numero esadecimale appartiene al sistema di numerazione in base *16*. Questa base è la quarta potenza di due: $2^4=16$, ne viene che il valore di ogni quattro cifre in binario corrisponde ad un numero esadecimale.

> Per trasformare in esadecimale un numero binario, basta suddividere questo in gruppi di quattro cifre, a partire dalla prima a destra, e sostituire ad ogni gruppo il corrispondente valore nella base decimale, la quale corrisponde al numero in esadecimale, in quanto un qualsiasi gruppo di quattro cifre binarie ha un valore inferiore a *16 (massimo 15⇒1111)*

Simboli per la numerazione esadecimale

Per rappresentare in numeri nel sistema esadecimale occorrono *15* simboli oltre quello nullo *(0)*.

Come simboli si assumono, usualmente, oltre allo zero *"0"* le cifre da *1* a *9* e, dopo quest'ultimo, le prime *6* lettere dell'alfabeto.

fig.1.23

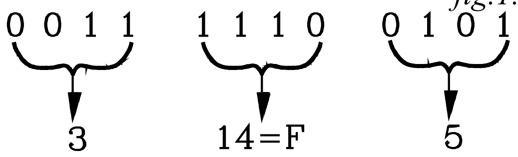
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
										↓	↓	↓	↓	↓	↓
										10	11	12	13	14	15

Si debba trasformare in esedacimale il numero binario :

$$1111100101 = 1 \cdot 2^9 + 1 \cdot 2^8 + 1 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 997$$

Si divida il numero in gruppi di quattro cifre partendo dalla prima a destra.

fig.1.24



Si sostituisce:

$0101 \Rightarrow 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 5$

$1110 \Rightarrow 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 14 \equiv E$

$0011 \Rightarrow 1 \cdot 2^1 + 1 \cdot 2^0 = 3$

Il numero nel sistema esadecimale è rappresentato dai simboli $3\ E\ 5$:

$$3\ E\ 5 = 3 \cdot 16^2 + 14 \cdot 16^1 + 5 \cdot 16^0 = 997$$

Il numero esadecimale ottenuto dà, ovviamente, lo stesso risultato nella trasformazione in base 10 del numero binario di partenza.

Vale la regola inversa per trasformare un numero dal sistema esadecimale a quello binario.

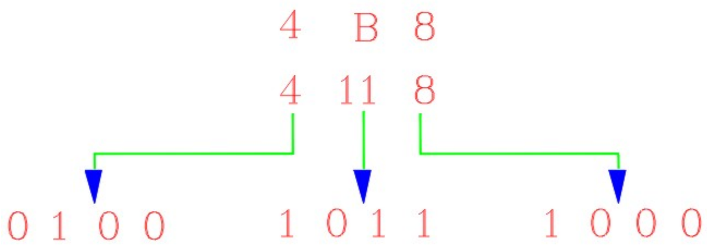
Per trasformare un numero dal sistema esadecimale in quello binario, basta sostituire, da destra verso sinistra, ad ogni cifra del numero esadecimale il corrispondente numero binario, completando, quando occorre, il gruppo di quattro cifre con zeri alla sinistra delle cifre significative del numero binario.

Si abbia così il numero esadecimale $4\,B\,8$:

$$4\,B\,8 = 4 \cdot 16^2 + 11 \cdot 16^1 + 8 \cdot 16^0 = 1208$$

Si sostituisce ad ogni cifra del numero esadecimale il corrispondente numero binario, completando a sinistra, se occorre, il groppo di *4* cifre con zeri.

fig.1.25



Si sostituisce:

$8 \Rightarrow 1000 = 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 \Rightarrow 1000$

$B \Rightarrow 1011 = 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 \Rightarrow 1011$

$4 \Rightarrow 100 = 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 \Rightarrow 0100$

Il numero dato, nel sistema binario è rappresentato da:

$$4\,B\,8 \Rightarrow 10010111000$$

5 CODIFICA DELLE INFORMAZIONI

5.1 Alfabeto

Viene definito *Alfabeto* un insieme finito e non vuoto di simboli, tra loro distinti, denominati *caratteri*

Così $A = \{0,1\}$ costituisce l'alfabeto binario, composto dai simboli *0,1*

5.2 Parola

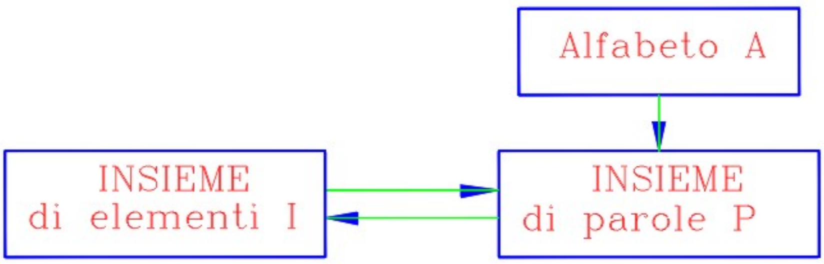
Si definisce *Parola* una *successione finita di caratteri*, anche con ripetizione, appartenenti all'insieme alfabetico prefissato.

Il numero binario *101* costituisce una parola formata con i due simboli dell'alfabeto binario.

5.3 Codifica

Si abbia un insieme *I* di *elementi qualsiasi*, da dover rappresentare e un insieme *P* di *parole* ottenute con un alfabeto *A*.

fig.1.26



Si definisce *codifica* un procedimento che fa corrispondere biunivocamente ad ogni *elemento* dell'insieme *I* una *parola* dell'insieme *P*, ottenuta con caratteri dell'*alfabeto A*.

Per esempio sia *I* l'insieme dei numeri naturali. Si può far corrispondere a ciascuno di essi, biunivocamente, una parola, presa dall'insieme *P* dei numeri binari *(parole)*, i quali sono formati con caratteri dell'alfabeto binario: $A = \{0,1\}$, composto dai due simboli *0,1*.

La corrispondenza si ottiene, come noto con la regola del polinomio caratteristico:

$$0 \to 0 \quad 1 \to 1 \quad 2 \to 10 \quad 3 \to 11 \text{ ecc.}$$

5.4 CODIFICA DEI NUMERI

La codifica dei numeri in binario puro non è vantaggiosa per l'elevato numero di caratteri occorrenti per rappresentare numeri grandi.

Occorre porre in rilievo che la memoria di un calcolatore è organizzata in celle; in tal modo si possono trattare stringhe di bit di limitate dimensioni.

Il primo multiplo del bit è il *byte*, composto da *otto bit*:

$$1byte=8bit$$

La parola per la codifica può essere formata da uno o più byte: usualmente *2* o *4* o *5 byt*e. Si hanno così parole di *16 (2 byte) - 32 (4 byte) - 40 (5 byte)*

5.4.1 Indirizzo

Ogni otto celle di memoria formano un byte. Ogni byte di memoria è numerato con un numero naturale partendo da 0.

Vi è un corrispondenza biunivoca tra byte di memoria e numeri naturali, denominati *indirizzi* : ad ogni byte corrisponde un indirizzo e viceversa.

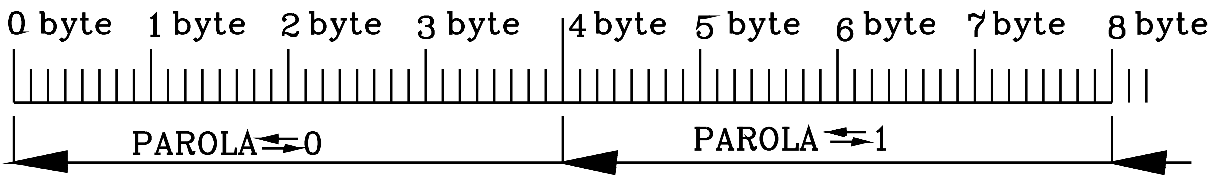
Come si è detto, un certo numero di byte costituisce la parola di *conformazione del sistema di memoria*: vi saranno così memorie conformate a 2 byte (16 bit) - 4 (32 bit) - 5 byte (40 bit).

Essendo i byte di memoria in corrispondenza biunivoca con i numeri naturali, nasce anche una corrispondenza biunivoca tra le *parole* di conformazione del sistema e i *numeri naturali*, multipli del numero di byte costituenti detta parola.

I numeri naturali che sono in corrispondenza biunivoca con le *parole*, costituiscono i loro *indirizzi*

Per chiarire quanto detto si consideri un sistema di memorizzazione a *4byte=32bit*.

fig.1.27



Facendo riferimento alla figura, i byte sono in corrispondenza biunivoca con i numeri naturali che iniziano da *0* : *byte ↔0 - byte ↔1 - byte ↔2* ecc.

Le parole, composte da *4 byte* sono in corrispondenza biunivoca con i numeri naturali multipli di *4*: *parola↔0 - parola↔4 - parola↔8 - parola↔16 ecc.*

5.5 Rappresentazione dei numeri interi

Ogni valore di tipo intero *"integer"* è contenuto in una parola.

In una parola di *4 byte=32 bit* possono essere disponibili $2^{32} = 4.294.967.296$ disposizioni diverse dell'alfabeto binario $A = \{0,1\}$, composto dai due simboli *0,1*; ogni disposizione può rappresentare un diverso numero intero, per cui con la configurazione a *32 bit* possono essere rappresentati *4.294.967.296* numeri interi diversi .

Con la configurazione di memoria, contenente *parole a n bit* si possono rappresentare 2^n numeri interi diversi.

5.5.1 Rappresentazione di numeri interi con segno

Per semplicità di trattazione si considera una conformazione con parole a $2\,byte = 16\,bit$ con la quale si possono rappresentare un totale di $2^{16} = 65.536$ numeri interi.

Per rappresentare i numeri interi con segno vi sono due metodi

5.5.1.1 1° metodo - bit del segno algebrico

Il segno algebrico viene rappresentato dal diverso valore assunto dal primo **bit** a sinistra della parola:

Il segno positivo è rappresentato *dal bit 0* il negativo *dal bit 1*

In tal caso le altre celle di memoria (nell'esempio *16-1=15*) rappresentano i numeri interi in valore assoluto.

In tale rappresentazione vi è una incongruenza nella rappresentazione dello *zer***o**. Questo viene rappresentato, assegnando il *bit=0* a tutte le celle determinanti il numero intero in valore assoluto (15 celle), con la prima cella che può assumere , indifferentemente, i due valori opposti : *0* oppure *1*: si hanno due configurazioni di *zero* *+0* *-0***.**

Da quanto detto in una conformazione di memoria, con parole a *16 bit*, una cella, la prima, è riservata ad indicare il segno, le altre *15* a rappresentare il valore assoluto del numero intero.

Con le *15 cell*e si possono avere $2^{15} = 32768$ disposizioni diverse, delle quali una è riservata a rappresentare lo *zero***,** sia quando il segno è positivo che negativo. Ne viene che:

Con una parola a *16 bit*, oltre lo zero possono essere rappresentati $32768 - 1 = 32767$ numeri interi positivi e altrettanti negativi:

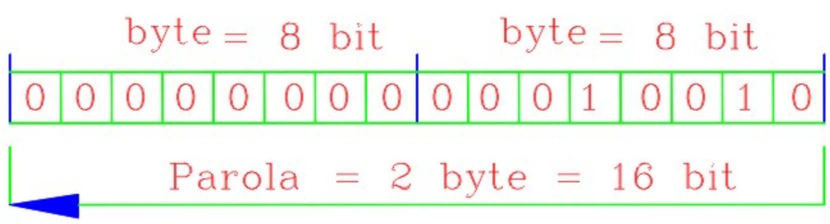
$$-32.767 \leq N \leq +32.767$$

Si debba rappresentare il numero *+18***.** Il numero è positivo, quindi alla prima cella di sinistra è assegnato il *bit=0*. Il numero intero 18 trasformato in binario fornisce la parola:

$$18 \xrightarrow{\text{in binario}} 10010$$

A tutte le celle a sinistra dell'ultima cifra significativa di *10010* si assegna il *bit=0*

fig.1.28

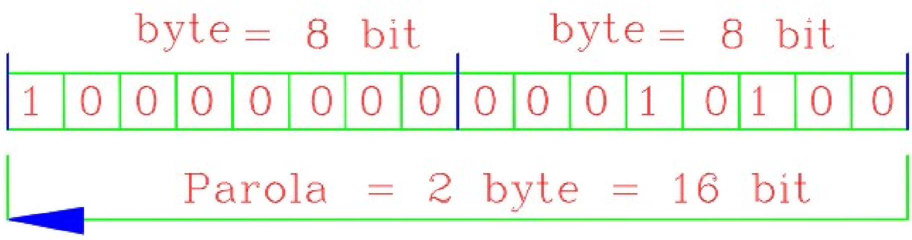


Si rappresenti il numero *-20***.** Il numero è negativo, quindi alla prima cella di sinistra è assegnato il *bit=1*. Il numero intero *20* trasformato in binario fornisce la parola:

$$20 \xrightarrow{\text{in binario}} 10100$$

A tutte le celle a sinistra dell'ultima cifra significativa di *10100* si assegna il *bit=0*

fig.1.29



In una conformazione con parole a *4 byte=32 bit*, la prima cella è adibita a indicare il segno, le altre *31* a rappresentare i valori assoluti dei numeri interi, compreso lo zero. Con *31 celle* oltre lo zero si possono rappresentare i numeri interi fino a $2^{31} - 1 = 2.147.483.647$

$$-2.147.483.647 \leq N \leq +2.147.483.647$$

5.5.1.2 2° metodo - Complemento a due

I numeri positivi si rappresentano con lo stesso metodo addotto nel precedente caso : si pone uno zero nella prima cella rappresentante il segno algebrico positivo e, nelle restanti celle, il numero assoluto trasformato in binario. Si completano le celle alla sinistra dell'ultima cifra significativa con *bit=0*

Per ottenere in codice un numero negativo, rappresentato da una parola a *n bit* (per esempio 16) si procede nella seguente maniera:

1- Si trasforma il valore assoluto del numero intero dal sistema decimale in quello binario, questo occuperà un numero di celle inferiore o tutt'al più uguale a *n-1* (negli esempi $\leq 16 - 1 = 15$), partendo dall'ultima cella di destra.

2- Alla sinistra del numero binario si pongono tanti *bit=0*, fino ad ottenere, in totale n cifre (compreso il numero binario)

3- *Si determina il complemento a due:*
Questo si ottiene effettuando la differenza tra il numero *2*, posto sulla n^{ma} colonna e il numero binario posto sulle precedenti *n-1 colonne*.
Con questa operazione sicuramente la prima cella di sinistra assume il valore **1**, indicante il segno negativo.

Per chiarezza ci si riferisca ad un esempio pratico. Si voglia rappresentare, con il *complemento a due*, il numero negativo -20 in un sistema a *16 bit* *"n=16"*.

- Il numero binario corrispondente al numero assoluto *20* è:

$$20 \xrightarrow{\text{in binario}} 10100$$

- Alla sinistra del numero binario *10100* si pongono tanti *0* fino ad ottenere in totale *16* cifre.

0000000000010100

- Si effettua la differenza tra il numero *2* posto sulla 16^{ma} colonna e il numero binario, completo degli zeri come spiegato al punto precedente.

fig.1.29

$$\begin{array}{cccccccccccccccc}
2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & + \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & = \\
\hline
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 &
\end{array}$$

In pratica, come si può notare dal risultato dell'esempio, il complemento a due si ottiene nella seguente maniera:

Per effettuare il *complemento a due* , partendo dalla prima cifra a destra del numero, si lasciano inalterati gli *0* fino al primo *1*, che resta pur esso invariato, per tutte le rimanenti cifre alla sinistra di questo si scambia *1* con *0* e viceversa.

Esempi

Con il metodo del complemento a due si debbano scrivere in codice i due numeri opposti *+44 , -44*.

- Si trasforma il valore assoluto del numero 44 nel sistema binario:

$$44 \xrightarrow{\text{in binario}} 101100$$

- Il numero positivo in codice si ottiene ponendo *0* sulla prima cella a sinistra $\left(16^{ma}\ \ colonna\right)$ che rappresenta la positività del numero; il *numero binario* del valore assoluto si pone sulle ultime celle a destra e nelle restanti celle alla sua sinistra si assegnano tutti 0.

$$+44 \xrightarrow{\text{in binario}} 0000000000101100$$

- Per ottenere in codice il numero negativo "-44" si deve effettuare il complemento a due del suo valore assoluto, espresso nel sistema binario.

- Si considera così il precedente numero che rappresenta *+44: 0000000000101100*

- Il numero presenta, partendo da destra, due 0 che rimangono inalterati. Si lascia inalterato il primo 1 che si incontra andando da destra verso sinistra; quindi, per tutte le altre cifre alla sinistra di esso, si scambia 1 con 0 e viceversa . Si ottiene così:

$$0000000000101100 \xrightarrow{\text{complemento a due}} 1111111111010100$$

- Il numero -44 è rappresentato in codice con il complemento a due da:

$$-44 \xrightarrow{\text{in codice con complemento a due}} 1111111111010100$$

La codifica dei numeri con il metodo del complemento a due ammette un unico zero costituito da tutte le celle con la cifra *0*.

Per il sistema di codifica dei numeri interi con segno, mediante il complemento a due nella rappresentazione dei negativi, si possono trarre le seguenti conclusioni:

1- In una conformazione di memorizzazione con parole a *n bit* si possono rappresentare un numero totale di numeri interi pari a 2^n

2- Nella rappresentazione con segno si ha un solo zero costituito da *n zeri (0)*

3- I numeri positivi sono rappresentati con uno zero posto sulla prima casella di sinistra; contraddistinguendo questa anche lo *zero (0)*, per la rappresentazione dei numeri positivi rimangono a disposizione le altre *n-1 cifre (celle)*

4- Dal punto precedente ne deriva che con parole a *n bit* possono essere rappresentati numeri interi fino al valore max 2^{n-1}

5- I numeri interi negativi sono rappresentati con un *1* posto sulla prima *cifra (cella)* di sinistra
6- I numeri interi negativi si ottengono effettuando il complemento a due del numero positivo codificato

5.6 Decodifica delle parole

Per decodificare una *parola* riferentesi ad un numero positivo occorre considerare che il primo *0* a sinistra contraddistingue la positività del numero, le restanti cifre rappresentano, nel sistema binario, il numero intero.

Il numero intero si ottiene trasformando le *cifre significative* della *parola* dal sistema *binario* in quello *decimale*: gli 0 alla sinistra delle cifre significative non vengono considerate.

Per decodificare una parola riferentesi ad un numero negativo basta effettuare il complemento a due dell'intera parola e quindi trasformare le cifre significative dal sistema binari in quello decimale

5.7 Traboccamento - Overflow

La conformazione di un sistema di memoria è contraddistinto dal tipo di parola che può essere a *16 - 32 - 40 bit*.

Un numero intero con segno, codificato, è rappresento da una particolare parola composta dalle diverse disposizioni con ripetizione dei due simboli dell'alfabeto binario $A = \{0,1\}$. Si genera una corrispondenza biunivoca tra numeri interi con segno e l'insieme delle parole. Da questo modo di rappresentazione in codice dei numeri interi ne viene che:

Non possono essere rappresentati numeri interi che, trasformati in codice, occupino un numero di cifre (celle) superiori a quelle componenti la parola di conformazione del sistema.

Se con un sistema di memorizzazione a *n bit* si immette un numero intero che codificato occupa più di *n cifre* si ha una errore di *traboccamento (overflow).*

Così può avvenire che, nell'effettuare una somma tra due numeri interi rappresentabili in codice, il risultato ha un valore che, codificato, supera il numero di bit del sistema: si ha in tal caso un errore di *traboccamento (overflow).*

5.8 Operazione di somma tra numeri interi codificati

Per comodità di trattazione si considerano parole composte solamente da *6 cifre (celle)*. Si suppone che il metodo di codifica dei numeri negativi sia quello del complemento a due.

Con il metodo di codifica adottato si tratta indifferentemente la somma tra due numeri interi dello stesso segno o di segni opposti; e precisamente:

1- Si sommano tutte le cifre delle due parole, compreso quella riferita al segno - Nella parola a *6 elementi*, presa come esempio, la cifra indicante il segno è la prima a sinistra: *sesta colonna*, partendo da destra **.**

2- L'eventuale riporto a sinistra, dalla somma delle ultime cifre indicanti il. segno, viene ignorato. Così nell'esempio con parola a 6 elementi, viene ignorato il riporto sulla 7^{ma} colonna .

Esempi

I Esempio

Si debbano sommare i due numeri negativi -7 , -6 :

$$-7+(-6)=-7-6=-13$$

- Si codifichino i due numeri con parole a 6 elementi:

$7 \xrightarrow{\text{in binario}} 111$ codifica +7 $+7 \xrightarrow{\text{parola a 6 elementi}} 000111$ $-7 \xrightarrow{\text{complemento a 2}} 111001$

$6 \xrightarrow{\text{in binario}} 110$ codifica +6 $+6 \xrightarrow{\text{parola a 6 elementi}} 000110$ $-6 \xrightarrow{\text{complemento a 2}} 111001$

- Si effettua ora la somma dei due numeri codificati.

```
riporti        1 1 1
             - - - - - - - - - - - - - -
                 1 1 1 0 0 1+
                 1 1 1 0 1 0=
               ______________
                 1 1 0 0 1 1
```

- Il numero è negativo essendo *1* la prima cifra a sinistra.
- Per ricavare il risultato occorre decodificare il numero negativo: si deve effettuare il complemento a due

$$110011 \xrightarrow{\text{complemento a due}} 001101$$

- Le cifre significative del risultato della somma sono *1101*. La trasformazione da binario in decimale fornisce il numero:

$$1101 = 1\cdot 2^3 + 1\cdot 2^2 + 0\cdot 2^1 + 1\cdot 2^0 = 13$$

II Esempio

Sommare i numeri *-7 , +3*

$$-7+(-3)=4$$

$7 \xrightarrow{\text{in binario}} 111$ codifica +7 $+7 \xrightarrow{\text{parola a 6 elementi}} 000111$ $-7 \xrightarrow{\text{complemento a 2}} 111001$

$3 \xrightarrow{\text{in binario}} 011$ codifica +6 $+3 \xrightarrow{\text{parola a 6 elementi}} 000011$

```
riporti              1 1
             - - - - - - - - - - - - - -
               1 1 1 0 0 1+
               0 0 0 0 1 1=
              ______________
               1 1 1 1 0 0
```

- Il numero risultante è negativo essendo *1* la prima cifra a sinistra. Il valore assoluto, decodificato, si ottiene effettuando il complemento a *2*

$$111100 \xrightarrow{\text{complemento a due}} 000100$$

- Si trasformano poi le cifre significative *100* dal sistema binario in quello decimale:

$$100 = 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 4$$

- Il risultato decodificato è quindi *-4*

5.9 Numeri a virgola mobile

Per codificare i numeri reali occorre rappresentare numeri formati da una parte intera ed una decimale. I numeri irrazionali da codificare si avvicineranno tanto più al loro valore vero quanto maggiore è il numero di cifre decimali che possono essere rappresentate.

Un metodo di codifica che risponde meglio alle esigenze di codificazione dei numeri reali è quello *a virgola mobile*.

Il numero, nella codifica a virgola mobile, viene scritto in forma normalizzata esponenziale, ed è costituito da cifre significative moltiplicate per una potenza di 10.

Il numero a virgola mobile è formato da due parti essenziali

Mantissa Rappresenta la prima parte del numero a virgola mobile ed è formato da un numero fisso di cifre intere e decimali, costituenti le cifre significative.

$$\pm x_1\ x_2\ x_3 \ldots x_h\ ,\ y_1\ y_1 \ldots y_k$$

Caratteristica È la potenza di 10 per la quale deve essere moltiplicata la mantissa per definire la posizione della virgola nel numero.

$$10^{\pm b_1\ b_2 .. b_n}$$

Il numero si presenta nella forma:

$$\pm x_1\ x_2\ x_3 \ldots x_h\ ,\ y_1\ y_1 \ldots y_k \cdot 10^{\pm b_1\ b_2 .. b_n}$$

dove $\pm x_1\ x_2\ x_3 \ldots x_h\ ,\ y_1\ y_1 \ldots y_k$ rappresenta la mantissa e l'esponente $\pm b_1\ b_2\ b_n$ rappresenta la caratteristica.

Esempio

Si fissi una **mantissa** formata da due cifre intere e 8 decimali. La **caratteristica** sia formata da due cifre.

Il numero *+234,562*	si scrive	$+23.45620000 \cdot 10^{+01}$
il numero *-0,32*	si scrive	$-32.00000000 \cdot 10^{-02}$

Vi sono ovviamente vari tipi di normalizzazione a seconda delle diverse forme che possono assumere mantissa e caratteristica.

Una particolare normalizzazione è quella di esprimere la mantissa con una parte intera **0** (zero) e tutte le cifre significative nella parte decimale. Il numero si presenta nella forma:

$$\pm 0.\ y_1\ y_1 \ldots y_k \cdot 10^{\pm b_1\ b_2 .. b_n}$$

Dove il punto sta al posto della la virgola.

Così *32,45* si scrive $0.3245 \cdot 10^{+02}$

Come per la precedente codifica dei numeri interi, il numero a virgola mobile codificato è rappresentato da *una parola*, con i suoi elementi costituenti una particolare disposizione con ripetizione dei due simboli dell'alfabeto binario.

Per la codifica sono possibili diverse rappresentazioni con adozione di particolari convenzioni.

Si prende qui in esame una particolare rappresentazione che adotta le seguenti convenzioni

1- Si sottintende lo zero della parte intera e il punto che esprime la mantissa nella forma normalizzata descritta. Così della forma normalizzata:

$$\pm 0.\, y_1\, y_1 \ldots y_k \cdot 10^{\pm b_1\, b_2 .. b_n}$$

si sottintendono i due caratteri *"0."*

2- Si fissa la lunghezza della mantissa in un numero di cifre costanti

3- Si limita il valore dell'esponente di due che serve a rappresentare il numero nel sistema binario. Tale esponente viene compreso in un intervallo fisso: per esempio $-64 \div +63$

4- Si somma convenzionalmente all'esponente un numero positivo in modo che il suo valore nell'intervallo di esistenza risulti sempre positivo

Così se si assume $-64 \div +63$ come intervallo di esistenza dell'esponente , affinché questo risulti sempre positivo basta sommare ad esso la costante *+64*.

In tal modo si ha:

a- L'esponente minimo *-64* in codice è rappresentato da: $-64 + 64 = 0$

b- L'esponente *0* in codice è rappresentato in codice da: $0 + 64 = 64$

c- L'esponente *+1* in codice è rappresentato in codice da: $+1 + 64 = 65$

Da quanto detto ne viene per esempio che:

- L'esponente *-23* si tramuta in codice nel numero: $-23 + 64 = 21$
- Viceversa il numero in codice *25* corrisponde all'esponente: $25 - 64 = -39$
- Così il numero in codice *+89* corrisponde all'esponente: $89 - 64 = +25$

Si consideri un sistema di conformazione della parola a *4 byte= 32 bit* .

Si adottino le seguenti convenzioni per codificare i numeri a virgola mobile.

Il primo *byte=8 bit* della parola riguarda il segno del numero e il valore dell'esponente, e precisamente:

a- Il primo *bit* rappresenta con *0* i numeri positivi con *1* i numeri negativi

b- Gli altri *7 bit* del primo *byte* della parola sono riservati a indicare l'esponente della base 2 per rappresentare la caratteristica, pari al numero di cifre binarie che la compongono , che si traduce in codice sommando ad esso la costante *+64*

La mantissa è rappresentata dagli altri *tre byte=24 bit* della parola

Con i *7 bit* riservati per codificare l'esponente, nel sistema binario si possono rappresentare $2^7 = 128$ numeri compreso lo zero.

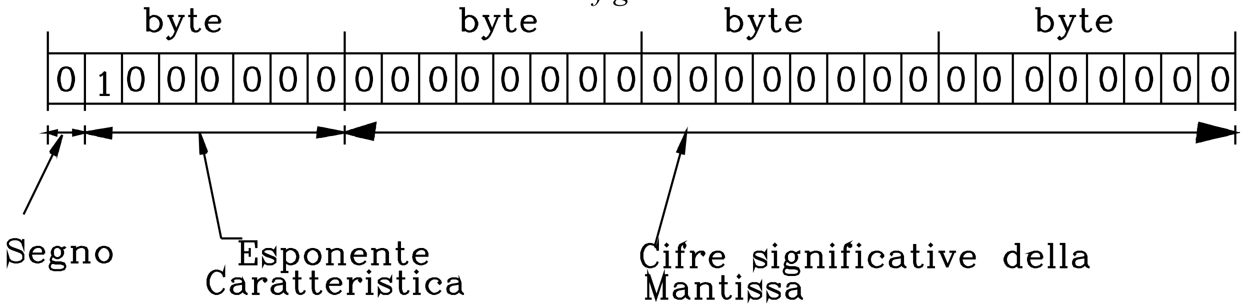
In codice l'esponente zero della base *2* è rappresentato da $0+64=64$. Questo trasformato in numero binario dà *1000000.*

Si osservi che l'esponente *1* della base *2* (2^1) è rappresentato da *1+64=65* che tradotto in numero binario è *1000001*

Il numero *0* è rappresentato in codice nella seguente maniera:

a- Si pone *0* sul primo bit *(prima cella)* rappresentante il segno.
b- Sulle altre 7 cifre del primo *byte* si pone il numero binario *1000000* corrispondente *64*, che, in codice, rappresenta l'esponente *0* $(64-64=0)$
c- Si pone zero *"0"* su tutti gli altri bit dei *tre byte* rappresentanti la mantissa.

fig.1.30



Se l'esponente zero, in codice è rappresentato dal numero 64 trasformato in binario:

1000000

tutti gli esponenti positivi sono rappresentati da numeri binari con *7* cifre aventi la cifra *1 fissa* sul 7^{mo} *bit* a sinistra. Il numero di esponenti positivi che si possono rappresentare sarà la metà meno *1* del numero di disposizioni con ripetizione dei due simboli *0,1* che si possono ottenere con le *7* celle:

$$\frac{2^7}{2}-1=\frac{128}{2}-1=64-1=63$$

occorre togliere *1* tenendo conto che una delle combinazioni *"1000000"* è riservata a rappresentare l'esponente *0*

L'altra metà delle disposizioni, che si possono ottenere con le *7* celle riservate alla codifica dell'esponente, contenti la cifra fissa *0* sul 7^{mo} *bit*, viene utilizzata per rappresentare gli esponenti negativi.

Si possono così rappresentare gli esponenti negativi fino a:

$$-\frac{2^7}{2}=-\frac{128}{2}=-64$$

Il minimo esponente codificabile è quindi -64. Per codificarlo si aggiunge +64 ottenendo:

$$-64+64=0$$

L'esponente -*64* è rappresentato da *0000000*

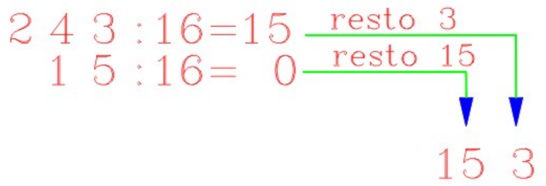
Esempio

Si trasformi il numero *+243,57* nel numero a virgola mobile secondo la codifica considerata con parole a 32 bit.

- Occorre trasformare il numero dal sistema decimale in quello binario.
- Dato che il numero è abbastanza grande, conviene prima convertirlo in esadecimale e da questo poi in binario.

Trasformazione parte intera

fig.1.31

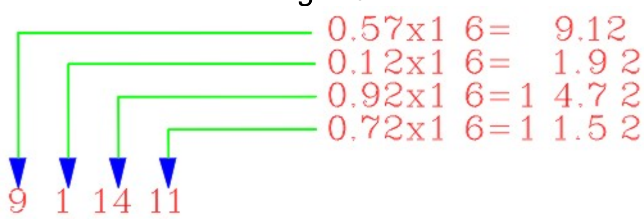


$$15 \cdot 16^{1} + 3 \cdot 16^{0} = 243$$

La parte intera in esadecimale è $243 \xrightarrow{esadecimale} F3$

Trasformazione parte decimale

fig.1.32



Terminando la trasformazione alla 4° cifra si ha il numero approssimato:

$$9 \cdot 16^{-1} + 1 \cdot 16^{-1} + 14 \cdot 16^{-1} + 11 \cdot 16^{-1} = 0.5699992$$

Con approssimazione $0.57 \xrightarrow{esadecimale} 0.91EB$

Il numero 243,57 nel sistema esadecimale è $243{,}57 \xrightarrow{esacecimale} F3.91EB$

- Si trasformi il numero da esadecimale in binario:

F	3.	9	1	E	B
1 1 1 1	0 0 1 1.	1 0 0 1	0 0 0 1	1 1 1 0	1 0 1 1

$$243{,}57 \xrightarrow{in\ binario} 11110011\ ,\ 1001000111101011$$

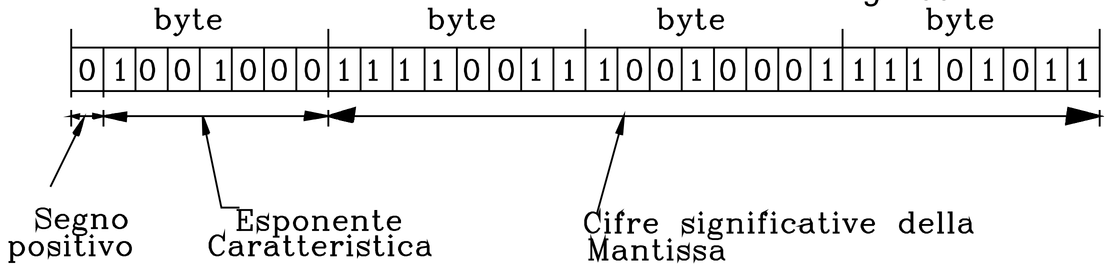
- Tale cifra significativa del numero binario va a riempire le celle dei *tre byte=24 bit* della parola riservati per memorizzare il numero.
- La caratteristica è il numero di cifre della parte intera del numero binario, pari al numero di celle a sinistra della virgola e rappresenta l'ottavo esponente di due compreso lo zero. Nel caso in esame la cifra intera è *11110011* composta da **8** cifre.
- Al numero *8* va sommato *+64*

$8 + 64 = 72$ tradotto in binario si ottiene $72 \xrightarrow{in\ binario} 1001000$

La caratteristica è *100100*

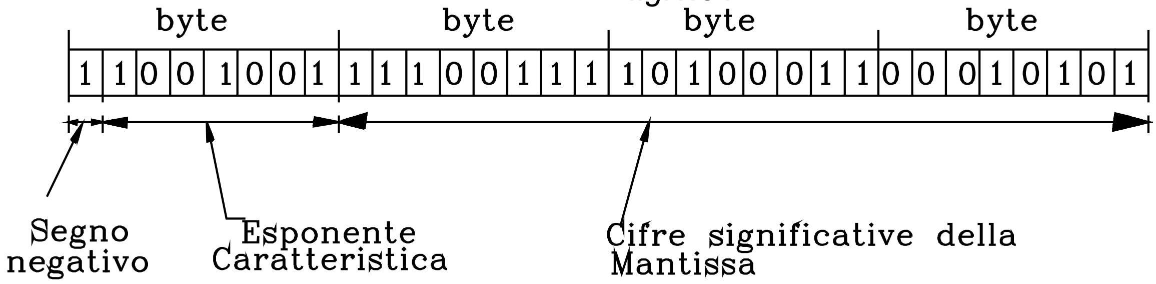
- Il numero da codificare *(+243,57)* è positivo, quindi il primo bit a sinistra è *0*
- Il numero *+243,57* in codice forma la seguente parola:

fig.1.33



Viceversa si debba trasformare in base 10 il seguente numero in codice

fig.1.34



- Il primo bit è 1, quindi il numero è negativo.
- La caratteristica è espressa dal numero

$$1001001 \xrightarrow{\text{corrisponde a}} 73$$

- Il numero di cifre in binario è *73-64=9*

- delle cifre significative se ne prendono 9 a sinistra prima della virgola

$$111001111,010001100010101$$

- Conviene suddividere il numero in gruppi di 4 per trasformarlo in esadecimale

1	1 1 0 0	1 1 1 1,	0 1 0 0	0 1 1 0	0 0 1 0	1 0 1 0
1	C	F,	4	6	2	A

- il numero nel sistema esadecimale è:

-1CF,462A;

trasportandolo nel sistema in base 10 si ottiene:

$$-1CF,462A = 1\cdot 16^{2} + 12\cdot 16^{2} + 15\cdot 16^{2} + 4\cdot 16^{-1} + 6\cdot 16^{-1} + 2\cdot 16^{-1} + 10\cdot 16^{-1} = -463,274078$$

Altro metodo di codifica

Un altro modo di codificare i numeri reali a virgola mobile è il seguente.

Si assuma una parola di *5 byte= 40 bit*

Le regole di codifica possono essere le seguenti:

1- Il primo *byte* è riservato con tutti i suoi 8 bit alla memorizzazione dell'esponente della base 2 del numero in codice binario. Con gli *8 bit* si possono memorizzare $2^8 = 256$ numeri compreso lo zero.

2- Metà dei numeri che si possono memorizzare, *"128"*, rappresentano quelli con segno negativo, partendo da *-1*; l'altra metà rappresentano lo zero *"0"* e i successivi *127* numeri positivi.

3- Affinché l'esponente*(la caratteristica)* sia sempre espresso da un numero positivo, si somma ad esso, convenzionalmente, la costante *+128*

4- Dalle precedenti convenzioni ne viene che lo zero della caratteristica è espresso in binario da:

$$0 \xrightarrow{\text{sommando } 128} 0 + 128 = 128 \xrightarrow{\text{in binario}} 10000000$$

Il minimo numero negativo della caratteristica è -128 e, sommando ad esso la costante +128, è rappresentato da:

$$-128 \xrightarrow{\text{sommando } 128} -128 + 128 = 0 \xrightarrow{\text{in binario}} 00000000$$

Il massimo numero positivo della caratteristica è $+127$ e, sommando ad esso la costante *+128*, è rappresentato da:

$$+127 \xrightarrow{\text{sommando } 128} +127 + 128 = 255 \xrightarrow{\text{in binario}} 11111111$$

La caratteristica è espressa da un numero dell'intervallo

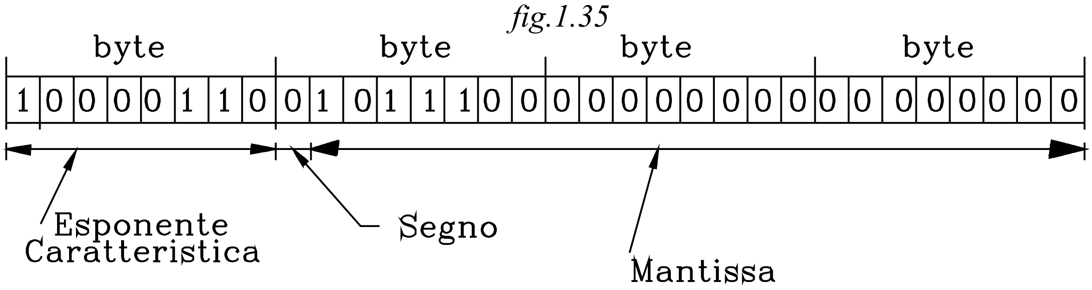
$$-128 \div +127$$

5- Gli altri *4 byte=32 bit* della parola sono riservati alla codifica del segno del numero e alla mantissa e precisamente:
Il primo bit alla sinistra è riservato al segno del numero: 0 per i numeri positivi 1 per i negativi.
I restanti 31 bit sono riservati alla codifica del valore assoluto della mantissa.

6- Convenzionalmente la mantissa viene scritta omettendo il primo *bit=1* significativo a sinistra: lo si sottintende nella codifica occorre poi aggiungerlo nella decodifica.

Si consideri il numero codificato secondo le suddette regole

fig.1.35



- L'esponente in binario 10000110 corrisponde a 134 , togliendo 128 si ottiene:
  Esponente base 2 $134 - 128 = 6$

- I restanti *4 byte* sono composti da *01011100 00000000 00000000 00000000*
- Il primo bit di sinistra è *0*, quindi il numero è positivo.

- Tolto il primo bit indicante il segno *(in questo caso 0)* occorre aggiungere il primo bit significativo sottinteso nella codifica ottenendo:
  11011100 00000000 00000000 00000000

- Occorre ora porre la virgola dopo le prime 6 cifre *(pari all'esponente)*. Si ottiene
  110111,00 00000000 00000000 00000000

- Il numero in binario è *110111* che corrisponde a:
  $$1 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 55$$

5.10Overflow e underflow

Se nel calcolo con numeri a virgola mobile il risultato di una operazione produce un numero la cui rappresentazione comporterebbe una mantissa con numero di bit superiori al sistema, questa viene troncata ottenendo un numero approssimato.

Se nel calcolo si ottiene un numero il cui esponente supera la caratteristica della parola di sistema si ottiene un traboccamento *"overflow"* e il computer non può produrre il risultato: il programma si blocca producendo in uscita un messaggio di errore.

Ciò avviene anche se l'esponente risulta molto piccolo: si ha in tal caso il fenomeno di *underflow*.

5.11CODICI

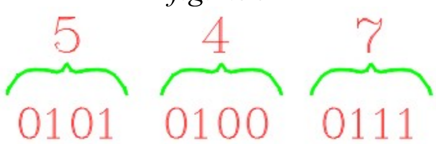
Sono stati studiati vari codici con i quali agli elementi di un *insieme* si fa corrispondere biunivocamente una *parola* composta da elementi *dell'alfabeto*, usualmente binario o esadecimale.

5.11.1 Codice BCD *(Binary Coded Decimal)*

È un sistema di codifica numerico con il quale ad ogni elemento dell'insieme delle *10 cifre* del sistema decimale $\{0,1,2,3,4,.5,6,7,8,9\}$ si fa corrispondere la parola di *4 elementi*, ottenuta cambiando la cifra decimale nella corrispondente binaria. La parola è composta da *4 elementi* dell'alfabeto binario $A = \{0,1\}$

Così il numero *543* viene codificato tramutando ogni sua cifra in una parola di *4 bit* che ne esprime il valore binario:

fig.1.36



$$547 \xrightarrow{\text{nel codice BCD}} 0101\ 0100\ 0111$$

Decimale	B CD
0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	0 1 0 1
6	0 1 1 0

7	0 1 1 1
8	1 0 0 0
9	1 0 0 1

5.11.3 Codice eccesso a tre

Come per il codice *BCD* e cifre da *0 a 9* del sistema decimale vengono codificate con parole di 4 bit dell'alfabeto binario.

Nella codifica con eccesso a tre si somma 3 alla cifre da codificare e poi il risultato lo si tramuta nel corrispondente valore binario. In tal modo, la cifra 0 corrisponde a 3 e l'ultima cifra 9 a 12. Si ottiene così la seguente tabella di codifica

Decimale	Eccesso a tre
0	0 0 1 1
1	0 1 0 0
2	0 1 0 1
3	0 1 1 0
4	0 1 1 1
5	1 0 0 0
6	1 0 0 1
7	1 0 1 0
8	1 0 1 1
9	1 1 0 0

Così il numero *953* in codice diviene: $953 \xrightarrow{\text{codice eccesso a 3}} 1100\ 1000\ 0110$

5.11.4 Codice Gray

Il codice *Gray* ha la caratteristica che ogni parola che esprime un numero differisce dalla precedente per il cambiamento di *un solo bit*.

Questa proprietà verrà utilizzata negli encoder assoluti nei quali si fa corrispondere ad una posizione angolare o lineare una parola di un certo numero di bit. Perché non vi sia ambiguità di lettura dei bit è bene che ogni posizione che può essere letta differisca dalla precedente per un solo bit.

fig.1.37

BINARIO	GRAY
0	0 0 0 0
1	0 0 0 1
2	0 0 1 1
3	0 0 1 0
4	0 1 1 0
5	0 1 1 1
6	0 1 0 1
7	0 1 0 0
8	1 1 0 0
9	1 1 0 1
10	1 1 1 1
11	1 1 1 0
12	1 0 1 0
13	1 0 1 1
14	1 0 0 1
15	1 0 0 0

Si noti che ciò non avviene con la codifica in binario puro. Si considerino per esempio le due posizioni successive contraddistinte dalle due parole corrispondenti ai due numeri $15 \rightarrow 16$: si passa dalla parola $1111 \xrightarrow{\text{alla parola}} 10000$ con il cambiamento contemporaneo di 4 bit.

Si consideri una codifica con parole a 4 bit .

Ogni parola di codifica di un numero decimale in codice Gray si ottiene dal corrispondente numero binario nel quale ogni cifra viene sommata con quella posta alla sua sinistra trascurando il riporto: in tal modo 1+1=0 senza riporto.

Per chiarezza si codifichi il numero 6 nel codice Gray con parole a 4 bit:

$6 \xrightarrow{\text{nel codice binario}} 0110$

Si inizi dalla prima cifra di destra.

1° cifra in binario: *0* → Si sostituisce al posto della cifra *0* la somma di essa con 1 che si trova alla sua sinistra: $0+1=1$

2° cifra in binario: *1* → Si sostituisce al posto della cifra *1* la somma di essa con *1* che si trova alla sua sinistra: $1+1=0$

3° cifra in binario: *1* → Si sostituisce al posto della cifra 1 la somma di essa con la cifra *0* che si trova alla sua sinistra: $1+0=1$

4° cifra in binario: *0* → Resta la cifra 0 in quanto alla sua sinistra non vi sono più cifre

In codice Gray il numero *5* si tramuta: $6 \xrightarrow{\text{in codice Gray}} 0101$

5.11.5 Codice ASCII

È un codice studiato per rappresentare oltre ai numeri, anche altri caratteri : lettere maiuscole e minuscole, segni di operazioni ecc. Inoltre sono codificate particolari istruzioni.

La sigla è una abbreviazione di "*American Standard Code for Information Interchaange*".

L'insieme da codificare è costituito dai simboli alfanumerici e particolari istruzioni; le parole, che biunivocamente si associano agli elementi dell'insieme, sono costituite da *7 bit* essenziali, presi dall'alfabeto binario $A = \{0,1\}$, divisi in due gruppi, di cui: 3 di peso forte a sinistra e 4 di peso debole a destra.

Con i 7 bit si possono formare un numero di parole:

$$2^7 = 128$$

e quindi possono essere codificabili 128 elementi dell'insieme.

I tre bit di peso forte possono assumere i valori che vanno: dal minimo *0*, corrispondente alla combinazione

$$0 \xrightarrow{\text{in binario}} 000,$$

al massimo al valore **7**, corrispondente alla combinazione

$$7 \xrightarrow{\text{in binario}} 111$$

I quattro bit di peso debole possono assumere i valori, che vanno dal minimo

$$0 \xrightarrow{\text{in binario}} 0000$$

al massimo $15 \xrightarrow{\text{in binario}} 1111 \xrightarrow{\text{in esadecimale}} F$

Occorre osservare che nei sistemi elettronici le parole sono suddivise in byte, costituite da *8 bit*. Nel codice *ASCII ai 7 bit* essenziali viene aggiunto un ottavo bit alla sinistra dei precedenti, che viene impiegato come controllo di parità.

Questo bit assume il valore *0* se, nella parola di codifica , vi compare un numero pari di caratteri 1; assume invece il valore 1 se detto carattere compare nella parola in numero dispari.

Così ad esempio nella parola *010 100* vi compare un numero dispari di caratteri *1* *(3 caratteri)*: si aggiunge 1 come ottavo carattere: *1010 1100*

In tal modo nella parola vi compare sempre in numero pari di caratteri *1*

Il codice ASCII viene ampiamente usato nei sistemi di elaborazione elettronica, per la comunicazione tra utente e macchina e tra questa e la memorizzazione su dischi.

Quando viene pigiato un tasto della tastiera viene emessa una parola di codifica in codice ASCII.

I primi 32 numeri di codice e il 127° sono riservati a particolari controlli. Occorre osservare che il codice ASCII è stato in origine studiato per trasmissioni attraverso telescriventi, per cui a queste si riferiscono le descrizioni delle istruzioni, estese poi alla trasmissione tra unità centrale e periferiche di sistemi a microprocessore.

Decimale	ASCII	Binario		Esadecimale	
1	SOH *(inizio testata)*	0 0 0	0 0 0 1	0	1
2	STX *(inizio testo)*	0 0 0	0 0 1 0	0	2
3	ETX *(fine testo)*	0 0 0	0 0 1 1	0	3
4	EOT *(fine trasmissione)*	0 0 0	0 1 0 0	0	4
5	ENQ *(richiesta)*	0 0 0	0 1 0 1	0	5
6	ACK *(conferma)*	0 0 0	0 1 1 0	0	6
7	BEL *(segnale sonoro)*	0 0 0	0 1 1 1	0	7
8	BS *(ritorna indietro)*	0 0 0	1 0 0 0	0	8
9	HT *(tabulazione orizzontale)*	0 0 0	1 0 0 1	0	9
10	LF *(avanzamento linea)*	0 0 0	1 0 1 0	0	A
11	VT *(Tabulazione verticale)*	0 0 0	1 0 1 1	0	B
12	FF *(avanzamento carta)*	0 0 0	1 1 0 0	0	C
13	CR *(ritorno carrello)*	0 0 0	1 1 0 1	0	D
14	SO *(disinserimento Shift)*	0 0 0	1 1 1 0	0	E
15	SI *(inserimento Shift)*	0 0 0	1 1 1 1	0	F
16	DLE *Uscita trasmissione)*	0 0 1	0 0 0 0	1	0
17	DC1 *(controllo periferica 1)*	0 0 1	0 0 0 1	1	1
18	DC2 *(controllo periferica 2)*	0 0 1	0 0 1 0	1	2
19	DC3 *(controllo periferica 3)*	0 0 1	0 0 1 1	1	3
20	DC4 *(controllo periferica 4)*	0 0 1	0 1 0 0	1	4
21	NAK *(conferma negativa)*	0 0 1	0 1 0 1	1	5
22	SYN *(attesa sincronizzazione)*	0 0 1	0 1 1 0	1	6
23	ETB *(fine blocco trasmiss.)*	0 0 1	0 1 1 1	1	7
24	CAN *(annullamento)*	0 0 1	1 0 0 0	1	8
25	EM *(fine supporto)*	0 0 1	1 0 0 1	1	9
26	SUB *(sostituzione)*	0 0 1	1 0 1 0	1	A
27	ESC *(uscita dal codice)*	0 0 1	1 0 1 1	1	B
28	FS *(separazione file)*	0 0 1	1 1 0 0	1	C
29	GS *(separazione di gruppo)*	0 0 1	1 1 0 1	1	D
30	RS *(separazione record)*	0 0 1	1 1 1 0	1	E
31	US *(Separazione unità)*	0 0 1	1 1 1 1	1	F
32	SP *(spazio)*	0 1 0	0 0 0 0	2	0
33	!	0 1 0	0 0 0 1	2	1
34	"	0 1 0	0 0 1 0	2	2
35	#	0 1 0	0 0 1 1	2	3
36	$	0 1 0	0 1 0 0	2	4
37	%	0 1 0	0 1 0 1	2	5
38	&	0 1 0	0 1 1 0	2	6
39	'	0 1 0	0 1 1 1	2	7
40	(	0 1 0	1 0 0 0	2	8
41		0 1 0	1 0 0 1	2	9
42	*	0 1 0	1 0 1 0	2	A
43	+	0 1 0	1 0 1 1	2	B
44	,	0 1 0	1 1 0 0	2	C
45	-	0 1 0	1 1 0 1	2	D
46	.	0 1 0	1 1 1 0	2	E
47	/	0 1 0	1 1 1 1	3	F
48	0	0 1 1	0 0 0 0	3	0
49	1	0 1 1	0 0 0 1	3	1
50	2	0 1 1	0 0 1 0	3	2
51	3	0 1 1	0 0 1 1	3	3

Decimale	ASCII	Binario		Esadecimale	
52	4	0 1 1	0 1 0 0	3	4
53	5	0 1 1	0 1 0 1	3	5
54	6	0 1 1	0 1 1 0	3	6
55	7	0 1 1	0 1 1 1	3	7
56	8	0 1 1	1 0 0 0	3	8
57	9	0 1 1	1 0 0 1	3	9
58	:	0 1 1	1 0 1 0	3	A
59	;	0 1 1	1 0 1 1	3	B
60	<	0 1 1	1 1 0 0	3	C
61	=	0 1 1	1 1 0 1	3	D
62	>	0 1 1	1 1 1 0	3	E
63	?	0 1 1	1 1 1 1	3	F
64	@	0 1 1	0 0 0 0	4	0
65	A	1 0 0	0 0 0 1	4	1
66	B	1 0 0	0 0 1 0	4	2
67	C	1 0 0	0 0 1 1	4	3
68	D	1 0 0	0 1 0 0	4	4
69	E	1 0 0	0 1 0 1	4	5
70	F	1 0 0	0 1 1 0	4	6
71	G	1 0 0	0 1 1 1	4	7
72	H	1 0 0	1 0 0 0	4	8
73	I	1 0 0	1 0 0 1	4	9
74	J	1 0 0	1 0 1 0	4	A
75	K	1 0 0	1 0 1 1	4	B
76	L	1 0 0	1 1 0 0	4	C
77	M	1 0 0	1 1 0 1	4	D
78	N	1 0 0	1 1 1 0	4	E
79	O	1 0 0	1 1 1 1	4	F
80	P	1 0 1	0 0 0 0	5	0
81	Q	1 0 1	0 0 0 1	5	1
82	R	1 0 1	0 0 1 0	5	2
83	S	1 0 1	0 0 1 1	5	3
84	T	1 0 1	0 1 0 0	5	4
85	U	1 0 1	0 1 0 1	5	5
86	V	1 0 1	0 1 1 0	5	6
87	W	1 0 1	0 1 1 1	5	7
88	X	1 0 1	1 0 0 0	5	8
89	Y	1 0 1	1 0 0 1	5	9
90	Z	1 0 1	1 0 1 0	5	A
91	[	1 0 1	1 0 1 1	5	B
92	\	1 0 1	1 1 0 0	5	C
93	]	1 0 1	1 1 0 1	5	D
94	^	1 0 1	1 1 1 0	5	E
95	_	1 0 1	1 1 1 1	5	F
96	`	1 1 0	0 0 0 0	6	0
97	a	1 1 0	0 0 0 1	6	1
98	b	1 1 0	0 0 1 0	6	2
99	c	1 1 0	0 0 1 1	6	3
100	d	1 1 0	0 1 0 0	6	4
101	e	1 1 0	0 1 0 1	6	5
102	f	1 1 0	0 1 1 0	6	6
103	g	1 1 0	0 1 1 1	6	7
104	h	1 1 0	1 0 0 0	6	8

Decimale	ASCII	Binario		Esadecimale	
105	i	1 1 0	1 0 0 1	6	9
106	j	1 1 0	1 0 1 0	6	A
107	k	1 1 0	1 0 1 1	6	B
108	l	1 1 0	1 1 0 0	6	C
109	m	1 1 0	1 1 0 1	6	D
110	n	1 1 0	1 1 1 0	6	E
111	o	1 1 0	1 1 1 1	6	F
112	p	1 1 1	0 0 0 0	7	0
113	q	1 1 1	0 0 0 1	7	1
114	r	1 1 1	0 0 1 0	7	2
115	s	1 1 1	0 0 1 1	7	3
116	t	1 1 1	0 1 0 0	4	4
117	u	1 1 1	0 1 0 1	4	5
118	v	1 1 1	0 1 1 0	4	6
119	w	1 1 1	0 1 1 1	4	7
120	x	1 1 1	1 0 0 0	4	8
121	y	1 1 1	1 0 0 1	4	9
122	z	1 1 1	1 0 1 0	4	A
123	{	1 1 1	1 0 1 1	4	B
124	|	1 1 1	1 1 0 0	4	C
125	}	1 1 1	1 1 0 1	4	D
126	~	1 1 1	1 1 1 0	5	E
127	DEL (Delete)	1 1 1	1 1 1 1	5	F

I primi 32 numeri di codice e il 127° sono riservati a particolari controlli. Occorre osservare che il codice ASCII è stato in origine studiato per trasmissioni attraverso telescriventi, per cui a queste si riferiscono le descrizioni delle istruzioni, estese poi alla trasmissione tra unità centrale e periferiche di sistemi a microprocessore.

Cljc per continuare

clic per precedente

Clic per la pagina iniziale