

## Rappresentazione dei numeri binari

---

I numeri binari vengono usati in due forme differenti: con segno e senza segno. Nella forma senza segno (*unsigned*, in inglese) il loro valore è sempre positivo; in quella con segno (*signed*), invece, il valore può essere sia negativo che positivo. I calcolatori fanno uso di un solo formato per i numeri senza segno e dei tre diversi formati segno e modulo (*sign-and-magnitude*), complemento uno (*one's complement*) e complemento due (*two's complement*) per i numeri con segno (Figura C.1).

---

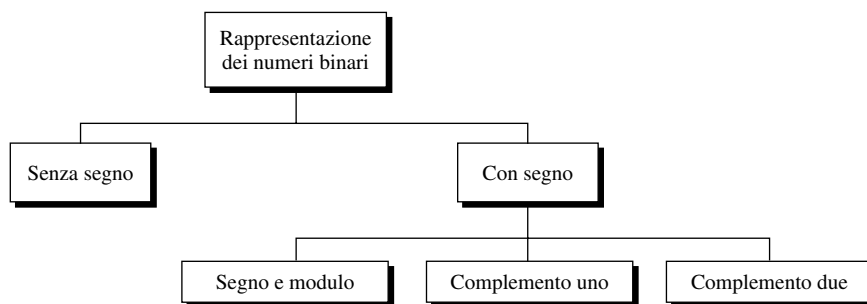
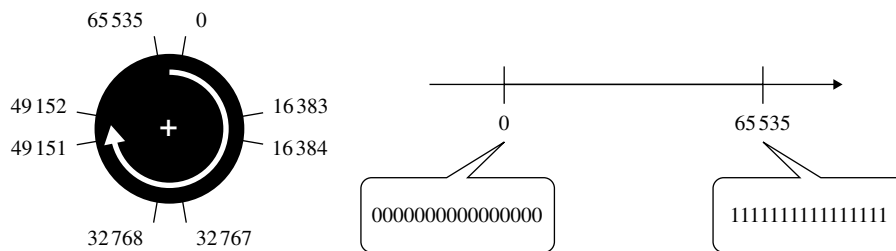


Figura C.1 Rappresentazione dei numeri binari.

### C.1 Numeri binari senza segno

Lo spazio a disposizione per immagazzinare le informazioni relative a un numero binario è limitato dalle dimensioni del buffer. In un buffer a 16 bit tutte le informazioni relative al numero, segno compreso, devono occupare non più di sedici bit. Nel caso di numeri senza segno, però, il numero viene assunto positivo e pertanto i sedici bit possono essere usati tutti per rappresentare le varie cifre del numero; in tal caso, quindi, un buffer a sedici bit permette di trattare un qualsiasi numero tra 0 (0000000000000000) e 65 535 (1111111111111111) (Figura C.2).



**Figura C.2** Senza segno.

Le limitazioni imposte da un buffer di sedici bit possono essere illustrate facendo uso dell'aritmetica modulare (Figura C.2): i numeri da 0 a 65 535 possono essere riportati su una circonferenza e si può assumere che aggiungendo 1 al valore massimo 65 535 si ottiene 0. Un semplice esempio di aritmetica modulare è quello del quadrante di un comune orologio a lancette: aggiungendo 1 ora alle 12 non si ottiene 13, ma 1.

### Da un numero decimale alla forma binaria senza segno

La forma binaria senza segno di un numero decimale può essere costruita effettuando le operazioni seguenti:

- il numero decimale viene scritto in forma binaria;
- tutte le celle vuote a sinistra del numero vengono riempite di zeri (fino a sedici celle nel caso di buffer a sedici bit).

### Esempio C.1

Si trasformi il numero 76 nella sua forma binaria senza segno.

#### Soluzione:

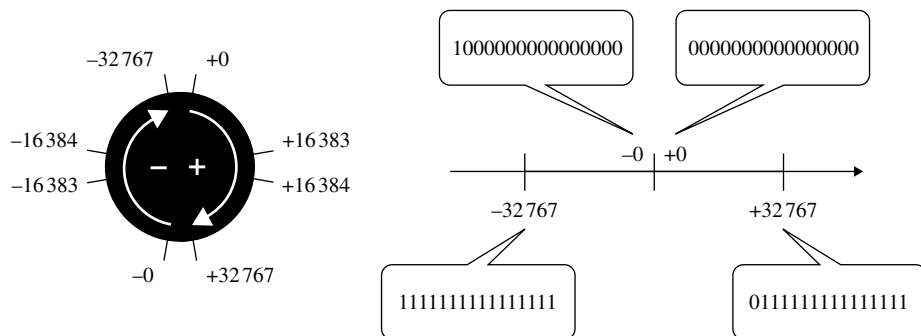
- la forma binaria del numero 76 è 1001100;
- riempendo le nove celle vuote con zeri si ottiene 000000001001100.

## C.2 Numeri binari con segno

La rappresentazione dei numeri binari con segno presenta aspetti più interessanti rispetto a quella dei numeri senza segno, infatti c'è il problema di inserire informazioni relative al segno (+ oppure -) rispettando le limitazioni imposte dal buffer a sedici bit. I tre metodi usati sono: segno e modulo, complemento uno e complemento due.

### Segno e modulo

Nel formato segno e modulo il bit più significativo dell'intera parola a sedici bit viene riservato al segno del numero: 1 per il segno meno e 0 per il segno più. Si osserva che per il numero zero esistono, quindi, due rappresentazioni equivalenti:  $+0$  (0000000000000000) e  $-0$  (1000000000000000). Dal momento che un bit è riservato al segno, i bit restanti per la rappresentazione del modulo sono soltanto quindici e pertanto l'intervallo di numeri possibili va da  $-32\,767$  a  $+32\,767$  (Figura C.3).



**Figura C.3** Formato segno e modulo.

Anche in questo caso si può fare ricorso all'aritmetica modulare per illustrare l'intervallo dei numeri rappresentabili: a partire da  $+0$  (0000000000000000) si aggiungono bit 1 fino al valore massimo  $+32\,767$  (0111111111111111). A questo punto si assume che aggiungendo un ulteriore 1 si ottenga  $-0$ , ovvero  $0111111111111111 + 1 = 1000000000000000$ . Si procede, quindi, con l'aggiunta di bit 1 fino a raggiungere il valore minimo  $-32\,767$  (1111111111111111); aggiungendo un ulteriore 1 si ottiene  $+0$  chiudendo, così, il circolo (Figura C.3).

### Da un numero decimale al formato segno e modulo

La forma binaria segno e modulo di un numero decimale può essere costruita effettuando le operazioni seguenti:

- si ignora il segno;
- il modulo del numero decimale viene scritto in forma binaria;
- tutte le celle vuote a sinistra del numero, tranne una, vengono riempite di zeri (fino a quindici celle nel caso di buffer a sedici bit).
- se il numero è positivo l'ultima cella viene riempita con un bit 0, se il numero è negativo l'ultima cella viene riempita con un bit 1.

### Esempio C.2

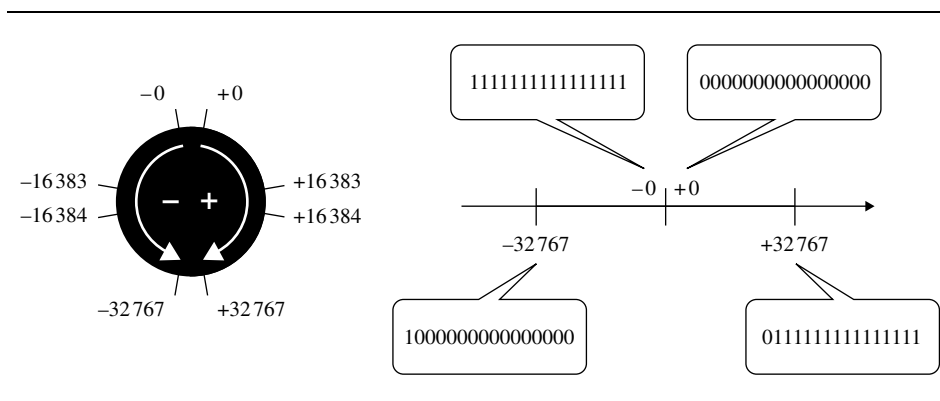
Si trasformi il numero  $-77$  nella sua forma binaria segno e modulo.

**Soluzione:**

- il modulo del numero è 77;
- la forma binaria del numero 77 è 1001101;
- riempendo le otto celle vuote con zeri si ottiene 000000001001101.
- il numero di partenza è negativo, pertanto l'ultima cella vuota viene riempita con un bit 1: 1000000001001101.

**Complemento uno**

Nel metodo complemento uno tutti i bit partecipano alla rappresentazione del segno del numero di partenza. Il complemento uno di un numero binario viene costruito modificando il valore di tutte le cifre che costituiscono il numero stesso; il complemento di un numero rappresenta lo stesso numero decimale cambiato di segno, pertanto la somma di un numero e del suo complemento fa zero. Per esempio +1 è rappresentato da 0000000000000001 e -1 da 1111111111111110. Questa sorta di simmetria vale anche per il numero zero: +0 = 0000000000000000 e -0 = 1111111111111111 (Figura C.4).

**Figura C.4** Complemento uno.

Il formato complemento uno di un numero decimale positivo coincide con quello segno e modulo (Figura C.4). In questo formato, inoltre, il segno di un numero è individuato dal valore del primo bit, infatti tutti i numeri positivi iniziano con il bit 0, mentre quelli negativi iniziano con il bit 1. D'altro canto il formato complemento uno di un numero negativo è completamente diverso da quello segno e modulo dello stesso numero.

Il formato complemento uno viene usato nella trasmissione di dati per controllare l'accuratezza della trasmissione.

**Da un numero decimale al formato complemento uno**

La forma binaria complemento uno di un numero decimale può essere costruita effettuando le operazioni seguenti:

- a. si ignora il segno;
- b. il modulo del numero decimale viene scritto in forma binaria;
- c. tutte le celle vuote a sinistra del numero vengono riempite di zeri (fino a sedici celle nel caso di buffer a sedici bit);
- d. se il numero è positivo non si fa altro, se il numero è negativo si fa il complementare di ogni singola cifra.

**Esempio C.5**

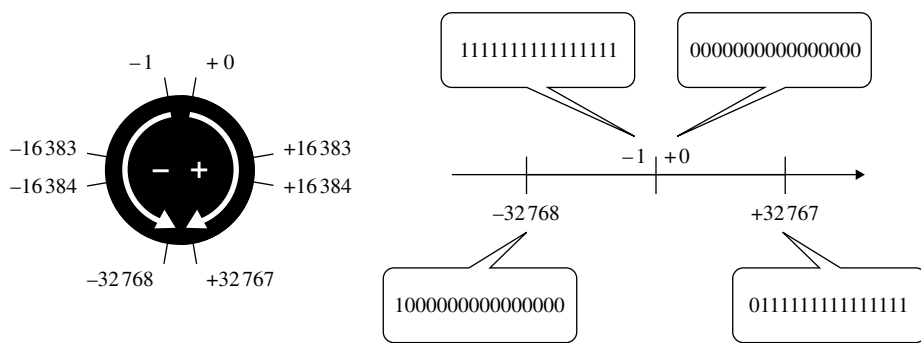
Si trasformi il numero  $-77$  nella sua forma binaria complemento uno.

Soluzione:

- a. il modulo del numero è 77;
- b. la forma binaria del numero 77 è 1001101;
- c. riempiendo le nove celle vuote con zeri si ottiene 000000001001101.
- d. il numero di partenza è negativo, pertanto è necessario calcolare il complementare di ogni singola cifra: 111111110110010.

**Complemento due**

Anche in questo formato tutti i bit partecipano alla rappresentazione del segno del numero di partenza, ma non vi è la medesima simmetria del metodo complemento uno; il numero zero, ad esempio, ha una sola rappresentazione. Per questa ragione l'intervallo dei numeri decimale che possono essere rappresentati con questo metodo è un po' più ampio, infatti va da  $-32\,768$  a  $+32\,767$  (Figura C.5).



**Figura C.5** Complemento due.

Un esame attento della Figura C.5 rivela che in questo formato i numeri 0 e  $-1$  sono uno l'inverso dell'altro, ma non sono complementari, infatti la loro somma è diversa da zero:  $0000000000000000(0) + 1111111111111111(-1) = 1111111111111111(-1)$ . Analogamente i numeri  $-32\,768$  e  $+32\,767$  sono l'uno il complementare dell'altro.

### Da un numero decimale al formato complemento due

La forma binaria complemento due di un numero decimale può essere costruita effettuando le operazioni seguenti:

- a. si ignora il segno;
- b. il modulo del numero decimale viene scritto in forma binaria;
- c. tutte le celle vuote a sinistra del numero vengono riempite di zeri (fino a sedici celle nel caso di buffer a sedici bit);
- d. se il numero è positivo non si fa altro, se il numero è negativo si fa il complementare di ogni singola cifra e si aggiunge 1 al risultato. Se con questa operazione si ottiene un riporto relativo al bit più significativo, tale riporto viene trascurato.

#### Esempio C.7

Si trasformi il numero  $-77$  nella sua forma binaria complemento uno.

#### Soluzione:

- a. il modulo del numero è 77;
- b. la forma binaria del numero 77 è 1001101;
- c. riempiendo le nove celle vuote con zeri si ottiene 000000001001101.
- d. il numero di partenza è negativo, pertanto è necessario calcolare il complementare di ogni singola cifra: 111111110110010. Ora si aggiunge 1 e si ottiene: 111111110110011.

## C.3 Il formato complemento uno: alcuni dettagli

In questa sezione vengono approfonditi alcuni aspetti del formato complemento uno rilevanti per il calcolo del checksum.

### Determinazione del complemento

Il complemento uno di un numero è un altro numero tale che la somma dei due numeri faccia zero. Il complemento uno di  $A$  è quindi  $-A$ . Per determinare il complemento uno di un numero binario è sufficiente scrivere il complementare di ogni sua cifra, ovvero trasformare gli 0 in 1 e gli 1 in 0 (Figura C.6).

Come si è già detto esistono due diverse rappresentazioni del numero 0:  $+0 = 0000000000000000$  e  $-0 = 1111111111111111$ .

**Nell'aritmetica complemento uno esistono due diverse rappresentazioni del numero 0:  $+0 = 0000000000000000$  e  $-0 = 1111111111111111$ .**

A e  $-A$  sono l'uno il complemento dell'altro

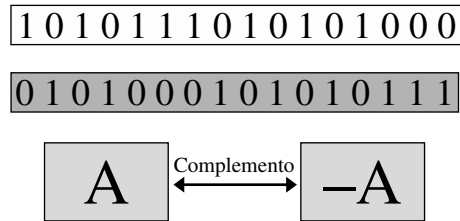


Figura C.6 Complemento uno.

### Somma di due numeri

La somma di due numeri in formato binario complemento uno viene effettuata con lo stesso algoritmo che si usa nel caso della somma tra numeri decimali, ma in base 2. Nel riquadro seguente le diverse azioni da compiere per sommare le cifre in una singola colonna vengono riassunte in quattro semplici regole:

#### Quattro regole per la somma di cifre appartenenti alla stessa colonna:

1. se non ci sono 1 il risultato è 0;
2. se vi è un solo 1 il risultato è 1;
3. se vi sono due 1 il risultato è 0 e viene riportato 1 alla colonna successiva;
4. se vi sono tre 1 il risultato è 1 e viene riportato 1 alla colonna successiva.

Nel riquadro seguente, invece, vengono riportate due semplici regole per la somma di due numeri binari:

1. somma dei bit in ciascuna colonna;
2. se l'ultima colonna genera in riporto viene aggiunto 1 al risultato.

Nella Figura C.7 è riportato un esempio di somma tra due numeri in formato complemento uno che non generano riporto nell'ultima colonna.

Nella Figura C.8, invece, è riportato un esempio di somma tra due numeri in formato complemento uno che generano riporto nell'ultima colonna.

Si osserva, infine, che dalle regole precedenti segue che sommando due numeri complementari si ottiene una parola costituita da soli 1 che, come si è detto, è equivalente a  $-0$  (Figura C.9).

Riporti	1 1 1 1 1 1	1 1 1	
Primo numero	1 0 0 0 0 1 0 1 0 1 0 0 0 0 1 1		+
Secondo numero	0 0 1 0 1 0 1 1 1 1 1 0 1 0 1 0 1		
Risultato	1 0 1 1 0 0 0 1 0 0 0 1 1 0 0 0		

**Figura C.7** Somma con il formato complemento uno.

Riporti	1	1 1 1 1 1 1	1 1 1	
Primo numero	1 0 0 0 0 1 0 1 0 1 0 0 0 0 1 1		+	
Secondo numero	1 0 1 0 1 0 1 1 1 1 1 0 1 0 1 0 1			
Risultato	0 0 1 1 0 0 0 1 0 0 0 1 1 0 0 0		+	
			→ 1	
Risultato finale	0 0 1 1 0 0 0 1 0 0 0 1 1 0 0 1			

**Figura C.8** Somma con il formato complemento uno in caso di riporto associato all'ultima colonna.

+A	1 0 1 0 1 1 1 0 1 0 1 0 1 0 0 0	+
-A	0 1 0 1 0 0 0 1 0 1 0 1 0 1 1 1	
-0	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	

**Figura C.9** Somma di due numeri complementari.

**Sommando due numeri complementari si ottiene -0, ovvero una parola costituita da soli 1.**