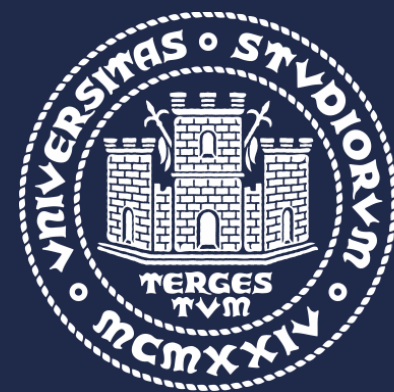




UNIVERSITÀ
DEGLI STUDI
DI TRIESTE

Rappresentazione dell'informazione (3)

Numeri interi negativi, numeri decimali in virgola fissa

Prof.ssa Giulia Cisotto

giulia.cisotto@units.it

Trieste, 25 febbraio 2026

RAPPRESENTAZIONE DI UN **NUMERO INTERO NEGATIVO**

Ci sono tre diverse possibili rappresentazioni:

- RAPPRESENTAZIONE «MODULO E SEGNO»
- Complemento a 1
- Complemento a 2

RAPPRESENTAZIONE DI UN **NUMERO INTERO NEGATIVO**

Ci sono tre diverse possibili rappresentazioni:

- **RAPPRESENTAZIONE «MODULO E SEGNO»**

Il primo bit (MSB) viene usato per il segno: 0 se POSITIVO, 1 se NEGATIVO

I rimanenti bit vengono usati per il modulo del numero

ESEMPIO su 8 bit $-4_{10} = 10000100_2$

Range di rappresentazione con n bit è $[-(2^{n-1} - 1), 2^{n-1} - 1]_{10}$

Problemi: (1) un bit è speso per il segno, (2) lo ZERO ha due diverse rappresentazioni

$$0000_2 = +0_{10}$$

$$1000_2 = -0_{10}$$

RAPPRESENTAZIONE DI UN **NUMERO INTERO NEGATIVO**

Ci sono tre diverse possibili rappresentazioni:

- **Complemento a 1**

Se il numero è positivo, si converte in binario usando il metodo standard, usando 1 bit in meno.

Se il numero è negativo, si converte in binario il modulo del numero, poi si trasforma nel suo **complemento**

Il **COMPLEMENTO** si ottiene «invertendo» tutti i bit: tutti i bit a zero vengono messi a 1 e tutti quelli a 1 vengono messi a 0.

Nota. Il primo bit è comunque per il segno: 0 = positivo, 1 = negativo

ESEMPIO su 4 bit

$$3_{10} = 0011_2$$

$$-3_{10} = \overline{0011_2} = 1100_2$$

Rimane il problema che lo ZERO ha due diverse rappresentazioni

0000 0000 (+0)

1111 1111 (-0)

RAPPRESENTAZIONE DI UN **NUMERO INTERO NEGATIVO**

Ci sono tre diverse possibili rappresentazioni:

- Complemento a 2

Se il numero è positivo, si converte in binario usando il metodo standard.

Se il numero è negativo, si converte in binario il modulo del numero, si passa al CA1, poi si somma +1.

Somma di numeri binari

ADDENDO #1	ADDENDO #2	RIPORTO	SOMMA
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Serve un bit in più!

Risolve il problema dello ZERO: adesso ha un'unica rappresentazione

0000 0000

RAPPRESENTAZIONE DI UN **NUMERO INTERO NEGATIVO**

Ci sono tre diverse possibili rappresentazioni:

- Complemento a 2

ESEMPIO su 4 bit: Trovare la rappresentazione CA2 del numero -7_{10}

$\overbrace{111}_2$

RAPPRESENTAZIONE DI UN **NUMERO INTERO NEGATIVO**

Ci sono tre diverse possibili rappresentazioni:

- Complemento a 2

ESEMPIO su 4 bit: Trovare la rappresentazione CA2 del numero -7_{10}

$$\overbrace{111}_7 \xrightarrow{4\text{bit}} \overbrace{0111}^{+7}_{\text{CA1}}$$

RAPPRESENTAZIONE DI UN **NUMERO INTERO NEGATIVO**

Ci sono tre diverse possibili rappresentazioni:

- Complemento a 2

ESEMPIO su 4 bit: Trovare la rappresentazione CA2 del numero -7_{10}

$$\overbrace{111}_7 \xrightarrow{4\text{bit}} \overbrace{0111}^{+7}_{\text{CA1}} \xrightarrow{\text{CA1}} \overbrace{1000}^{-7}_{\text{CA1}}$$

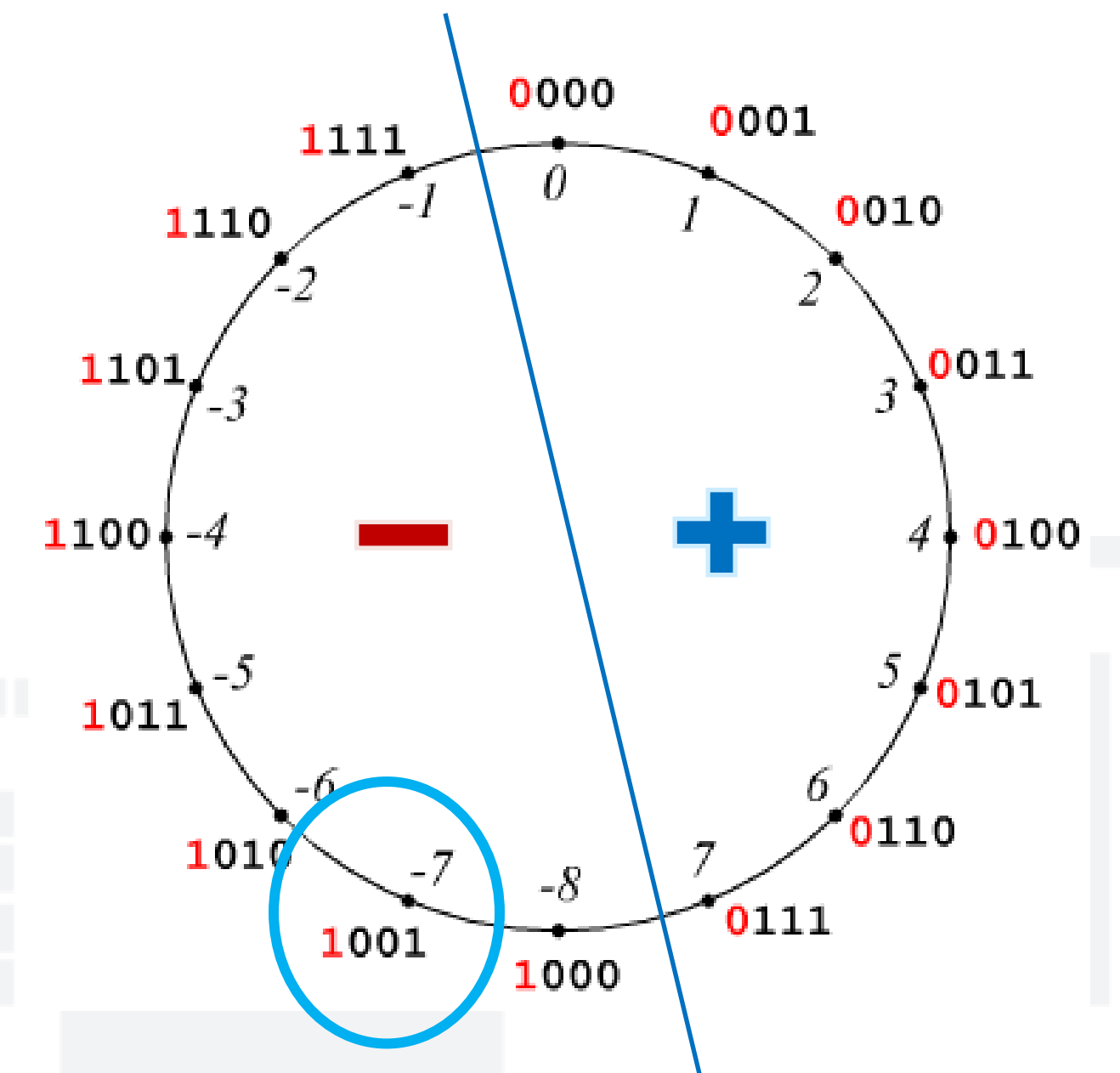
RAPPRESENTAZIONE DI UN **NUMERO INTERO NEGATIVO**

Ci sono tre diverse possibili rappresentazioni:

- **Complemento a 2**

ESEMPIO su 4 bit: Trovare la rappresentazione CA2 del numero -7_{10}

$$\overbrace{111}_7_2 \xrightarrow{4\text{bit}} \overbrace{0111}^{+7}_{CA1} \xrightarrow{CA1} \overbrace{1000}^{-7}_{CA1} \xrightarrow{+1} \overbrace{1001}^{-7}_{CA2}$$



I numeri negativi iniziano comunque con 1 (MSB=1)

Range di rappresentazione con n bit è $[-2^{n-1}, 2^{n-1} - 1]_{10}$

RAPPRESENTAZIONE DI UN **NUMERO INTERO NEGATIVO**

Ci sono tre diverse possibili rappresentazioni:

- **Complemento a 2: COME SI CALCOLA?** **Esistono 3 diversi modi**

(1) $CA2(X) = 2^n - X$ *(serve saper fare la sottrazione binaria: v. lezione di domani)*

(2) Passando per il CA1

- $CA1(X) = 2^n - 1 - X$
- $CA2(X) = CA1(X) + 1$

(3) Regola pratica. Algoritmo:

- Si parte da destra, si trascrivono tutti gli 0 fino ad incontrare il primo 1 e si trascrive anch'esso
- Si complementano a 1 tutti i bit restanti ($0 \rightarrow 1, 1 \rightarrow 0$)

RAPPRESENTAZIONE DI UN **NUMERO INTERO NEGATIVO**

Ci sono tre diverse possibili rappresentazioni:

- **Complemento a 2** **ESEMPIO su 4 bit:** Trovare la rappresentazione CA2 del numero -7_{10}

(1) $2^4 - 7 = 10000 - 111 = 1001_{CA2}$

(2) $\overbrace{111}_7_2 \xrightarrow{4\text{bit}} \overbrace{0111}^{+7}_{CA1} \xrightarrow{CA1} \overbrace{1000}^{-7}_{CA1} \xrightarrow{+1} \overbrace{1001}^{-7}_{CA2}$

(3) $\overbrace{111}_7_2 \xrightarrow{4\text{bit}} \overbrace{0111}^{+7}_{CA2} \xrightarrow{CA2} \overbrace{1001}^{-7}_{CA2}$

ESERCIZI (15')

1. Qual è il range rappresentabile in CA2 con 4 bit? E con MS?
2. Rappresentare il numero -51_{10} in MS, CA1 e CA2 su 8 bit
3. Convertire il numero 1101_2 sapendo che è rappresentato in codifica CA2.

SOLUZIONI

RAPPRESENTAZIONE NUMERI INTERI NEGATIVI	RANGE DI VALORI RAPPRESENTABILI	RANGE SU 4 BIT
«Modulo e segno»	$[-(2^{N-1} - 1), 2^{N-1} - 1]_{10}$	$[-7, +7]$
CA1	$[-(2^{N-1} - 1), 2^{N-1} - 1]_{10}$	$[-7, +7]$
CA2	$[-2^{N-1}, 2^{N-1} - 1]_{10}$	$[-8, +7]$
MSB = 0 → POSITIVO, MSB = 1 → NEGATIVO		

Rappresentare il numero -51_{10} in MS, CA1 e CA2 su 8 bit

Modulo e segno: $-51_{10} = 10110011_2$

Complemento a 1: $51_{10} = 00110011_2 \xrightarrow{\text{CA1}} -51_{10} = 11001100_2$

Complemento a 2: $51_{10} = 00110011_2 \xrightarrow{\text{CA1}} 11001100_2 \xrightarrow{\text{CA2}} 11001100_2 + 00000001_2 = 11001101_2$

Convertire il numero 1101_2 sapendo che è rappresentato in codifica CA2.

Controllo il primo bit: 1 → numero negativo

In alternativa: $1101_2 = -8 + 4 + 0 + 1 = -3_{10}$

Per trovare il valore assoluto devo applicare il CA2: $1101_2 \xrightarrow{\text{CA1}} 0010_2 \xrightarrow{\text{CA2}} 0010_2 + 0001_2 = 0011_2 = 3_{10}$

Il complemento a 2 è un'operazione involutiva: farla due volte riporta al valore originale.

RAPPRESENTAZIONE DI UN NUMERO REALE

Come si fa??

RAPPRESENTAZIONE DI UN NUMERO REALE

I numeri reali possono essere rappresentati:

- in virgola fissa (*fixed point*)
- in virgola mobile (*floating point*)

VIRGOLA FISSA

Nel sistema di numerazione in virgola fissa:

- Si riserva **un numero fisso di bit** per la parte intera e per la parte frazionaria
- La **posizione della virgola** è quindi **implicita**
- La posizione della virgola è **uguale** per tutti i numeri

VIRGOLA FISSA (UNSIGNED)

Quando devono essere rappresentati numeri positivi e lo zero (**unsigned**).

Assumiamo di avere **N bit** a disposizione.

- $L < N$ bit per rappresentare la parte intera del numero
- $D = N - L$ bit per rappresentare la parte decimale del numero

L-1	L-2	...	0	-1	-2	...	-D
PARTE INTERA				PARTE FRAZIONARIA			

VIRGOLA FISSA (UNSIGNED)

Quando devono essere rappresentati numeri positivi e lo zero (**unsigned**).

Assumiamo di avere **N bit** a disposizione.

- $L < N$ bit per rappresentare la parte intera del numero
- $D = N - L$ bit per rappresentare la parte decimale del numero

Non c'è il segno, quindi si possono rappresentare solo numeri dallo zero in su

L'intervallo di numeri rappresentabile è quindi:

$$[0, 2^L - 2^{-D}]$$

SI VEDA ANCHE LEZIONE 26/02/2026 (I set di slide)

VIRGOLA FISSA (UNSIGNED)

Esempi di applicazione

Grafica digitale → pixel (es. livelli di colore 0-255 per immagini a 8 bit).

Misurazioni che non possono essere negative → distanza, velocità (es. GPS, sensori di movimento).

Tempo e frequenze → durata di eventi, campionamento di segnali.

Dati finanziari e contatori → somme di denaro, conteggi.

VIRGOLA FISSA (SIGNED)

SI VEDA LEZIONE 26/02/2026
(I set di slide)

VIRGOLA FISSA (SIGNED)

Supponiamo di poter rappresentare i numeri reali con $N = 8$ bit.

Possiamo dedicare, alternativamente:

- 4 bit alla parte intera (incluso il segno), 4 bit alla parte decimale (Q4.4)
- 6 bit alla parte intera (incluso il segno), 2 bit alla parte decimale (Q6.2)

Come scegliere??

Dipende se si ha maggiore **necessità di precisione numerica** sulla parte intera o su quella decimale. Se si vuole rappresentare *numeri più grandi*, bisogna dare più spazio (più cifre) alla parte intera. Se invece si necessita di *maggior precisione decimale*, allora si dedicheranno più cifre alla parte decimale.

VIRGOLA FISSA (SIGNED)

Esempi di applicazione

Audio digitale (formati PCM a 16 o 24 bit) → i segnali sonori variano tra valori negativi e positivi.

Sensori e misurazioni fisiche → temperatura, tensione, corrente (valori positivi e negativi).

Elaborazione matematica → calcoli con numeri reali dove sono necessari anche valori negativi.

CONVERSIONE BINARIO FRAZIONARIO --> DECIMALE

X.YYYY è un numero frazionario in virgola fissa *unsigned* in base 2.

Convertirlo in base 10.

Regola: si converte esattamente come visto per i numeri interi.

Esempio:

$$N = 101.01_2 = 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2} = 4 + 0 + 1 + 0 + 0.25 = 5.25_{10}$$

Nota: lo stesso vale se il numero di partenza fosse in base 16.

CONVERSIONE DECIMALE --> BINARIO

X.YYYY è un numero frazionario *positivo* in base 10.

Convertirlo in base 2 in virgola fissa *unsigned*.

Algoritmo:

1. Convertire la parte intera in base 2
2. Si considera solo la parte decimale e la si moltiplica per 2. Si tiene da parte la cifra intera del risultato (che sarà 0 oppure 1)
3. Si ripete 2. fino a che il resto è 0, oppure si trova pattern ripetuto, oppure si esauriscono le cifre a disposizione della parte decimale

CONVERSIONE DECIMALE --> BINARIO

Esempio: $N = 3.625_{10}$

$$3_{10} = 11_2$$

0.625	.2	1.25
0.25	.2	0.5
0.5	.2	1
0		0

$$N = (3.625)_{10} = (11.101)_2$$

ESERCIZI (10')

Convertire in binario in virgola fissa su 8 bit i seguenti numeri decimali

$$N = 15.1875_{10}$$

$$N = 9.1_{10}$$

$$N = 9.6234_{10}$$

SOLUZIONI

Convertire in binario in virgola fissa su 8 bit i seguenti numeri decimali

$$N = 15.1875_{10}$$

$$15_{10} = 1111_2$$

0.1875	·2	0.375
0.375	·2	0.75
0.75	·2	1.5
0.5	·2	1
0		

frazioni via via più piccole

$$N = (15.1875)_{10} = (1111.0011)_2$$

CONVERSIONE DECIMALE --> BINARIO

Attenzione: alcuni numeri decimali potrebbero diventare binari periodici.

Esempio: $N = 9.1_{10}$

$$9_{10} = 1001_2$$

$$N = (9.1)_{10} = (1001.000110011..)_2$$

0.1	·2	0.2
0.2	·2	0.4
0.4	·2	0.8
0.8	·2	1.6
0.6	·2	1.2
0.2	·2	0.4
:	:	:

CONVERSIONE DECIMALE --> BINARIO

Alcuni numeri decimali potrebbero non essere rappresentati con precisione.

Esempio: $N = 9.6234_{10}$

$$9_{10} = 1001_2$$

$$N = (9.1)_{10} = (1001.10011111..)_{2}$$

0.6234	·2	1.2468
0.2468	·2	0.4936
0.4936	·2	0.9872
0.9872	·2	1.9744
0.9744	·2	1.9488
0.9488	·2	1.8976
0.8976	·2	1.7952
0.7952	·2	1.5904
:	:	:

Se abbiamo a disposizione 8 bit per la parte decimale, allora la rappresentazione è precisa. Altrimenti, la rappresentazione non è precisa (non si hanno sufficienti cifre per rappresentare la parte decimale).

VIRGOLA FISSA

Svantaggi

- Rigidità nell'assegnazione dello spazio (precisione) per parte decimale
- Trade-off a-priori tra precisione su numeri piccoli e codifica di numeri grandi
- Spreco di bit a 0 quando il numero è intero

VIRGOLA MOBILE



La massa dell'elettrone è
0.0000000000000000000000000000000091 Kg



La massa della terra è
597360000000000000000000000 kg

Requisiti ideali:

- Convenzione per rappresentazione univoca per numeri così diversi
- Costo limitato di rappresentazione nel sistema binario (significa memoria!)

Materiale per la lezione

- Appendici A e B Tanenbaum (numeri binari, virgola mobile, IEEE 754)