

Precisione

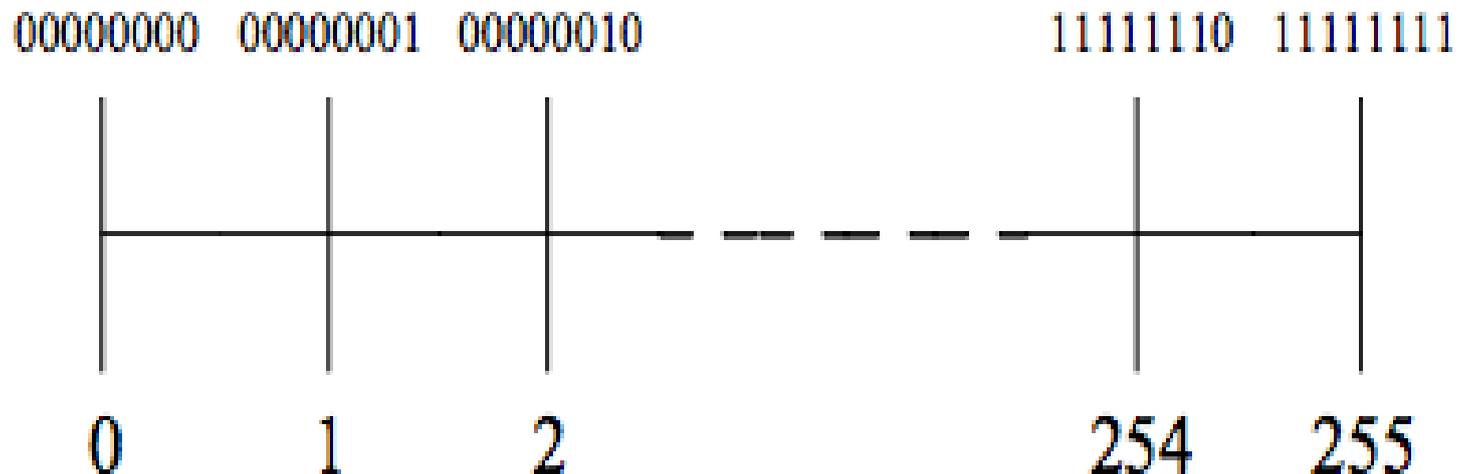
Quantità di cifre binarie utilizzate per rappresentare un'informazione

Es 8 bit = $2^8 = 256$ codifiche

→ 0 – 255

→ non posso rappresentare il numero 256

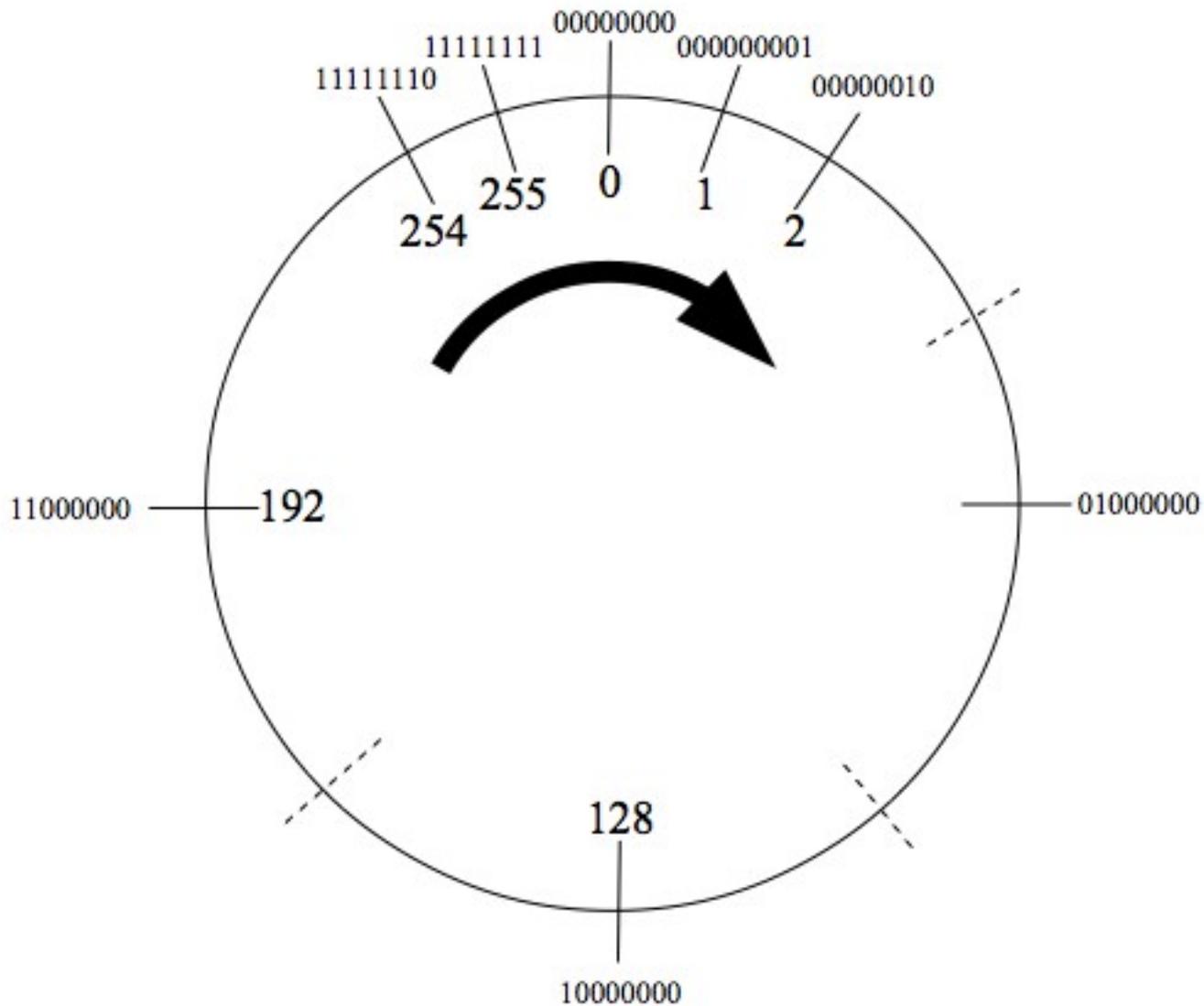
→ informazione finita



- Anche aumentando il numero di bit, gli elementi di codifica restano finiti
- Es 32 bit = 4.294.967.296 numero finito

Overflow

255+1 ?

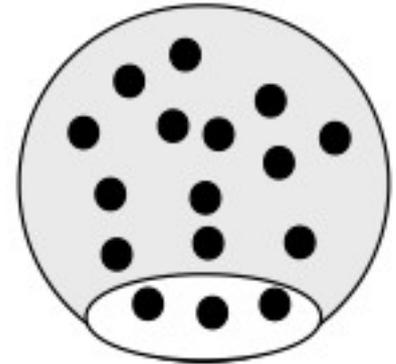


Insieme Z

- È infinito
- Contiene + e -
- Coppie di numeri $(-X, +X)$ t.c. $-X+X=0$
- → serve una codifica diversa da quella binaria

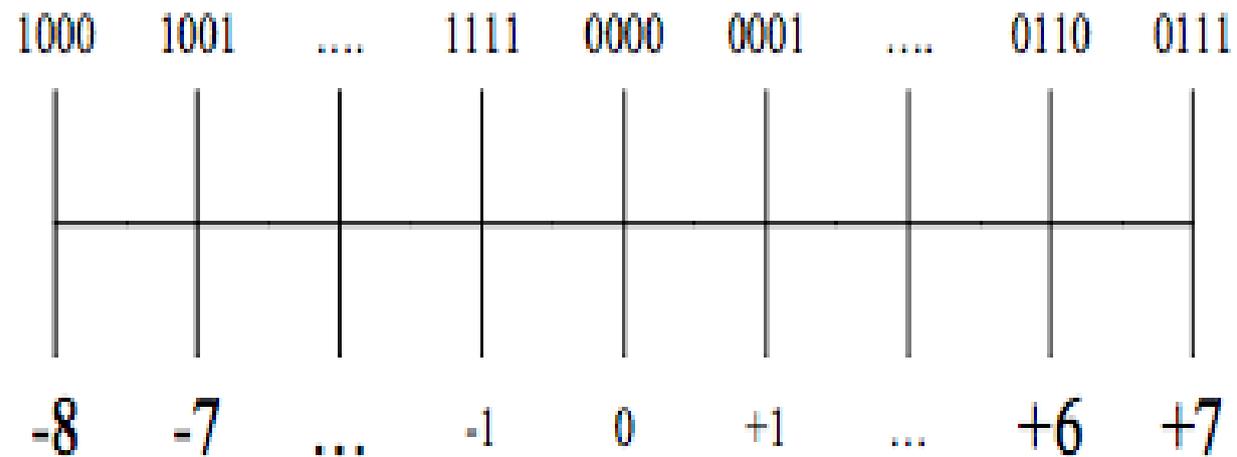
Complemento-a-2

- $X = B^N - X$
- Es $3 = B^N - 3 \rightarrow 4 \text{ bit} \rightarrow 16 - 3 = 13$
- 13 rappresenta il complemento-a-2 di $+3 \rightarrow -3$
- $B^N = 10000$ $3 = 0011$
- $10000 - 0011 = 1101 \rightarrow 13$
- Verifichiamo: $+3 + (-3)$
- $0011 + 1101 = 0000$



Complemento-a-2

<i>Codifica</i>	<i>Significato in N</i>	<i>Significato in Z</i>
0000	0	+0
0001	1	+1
0010	2	+2
0011	3	+3
0100	4	+4
0101	5	+5
0110	6	+6
0111	7	+7
1000	8	-8
1001	9	-7
1010	10	-6
1011	11	-5
1100	12	-4
1101	13	-3
1110	14	-2
1111	15	-1



Complemento-a-2

Vantaggi

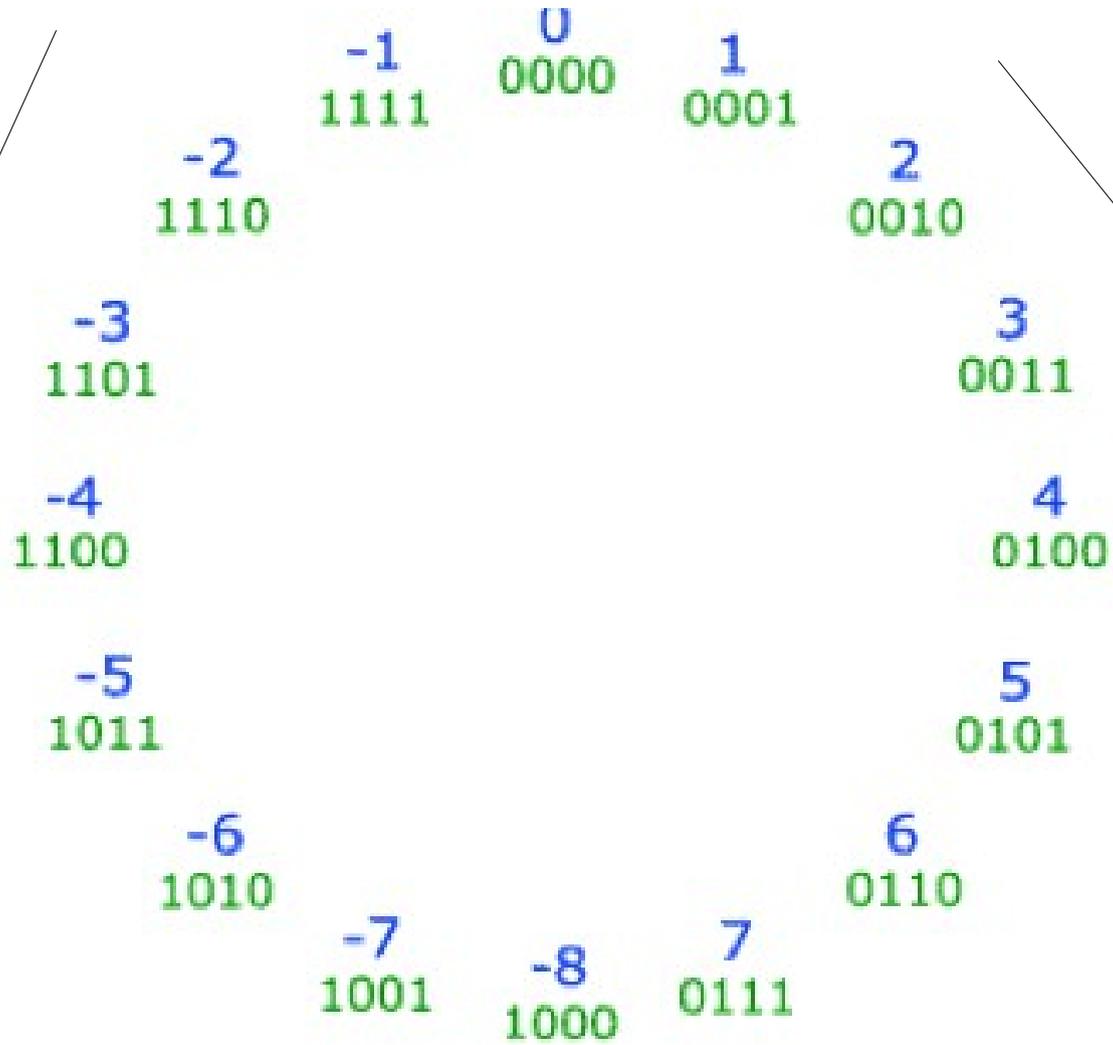
- Le operazioni di sottrazione non sono operazioni diverse dalle somme ma possono essere eseguite come somme del minuendo con il complemento-a-due del sottraendo:

$$\text{Diff} = \text{Minuendo} - \text{Sottraendo} = \text{Minuendo} + C2(\text{Sottraendo})$$

- Esiste una unica codifica per il valore 0
- Il riconoscimento del segno è comunque immediato perché tutti i numeri positivi iniziano per 0 e tutti quelli negativi per 1 come nella codifica modulo-e-segno.

SOTTRAZIONE

SOMMA

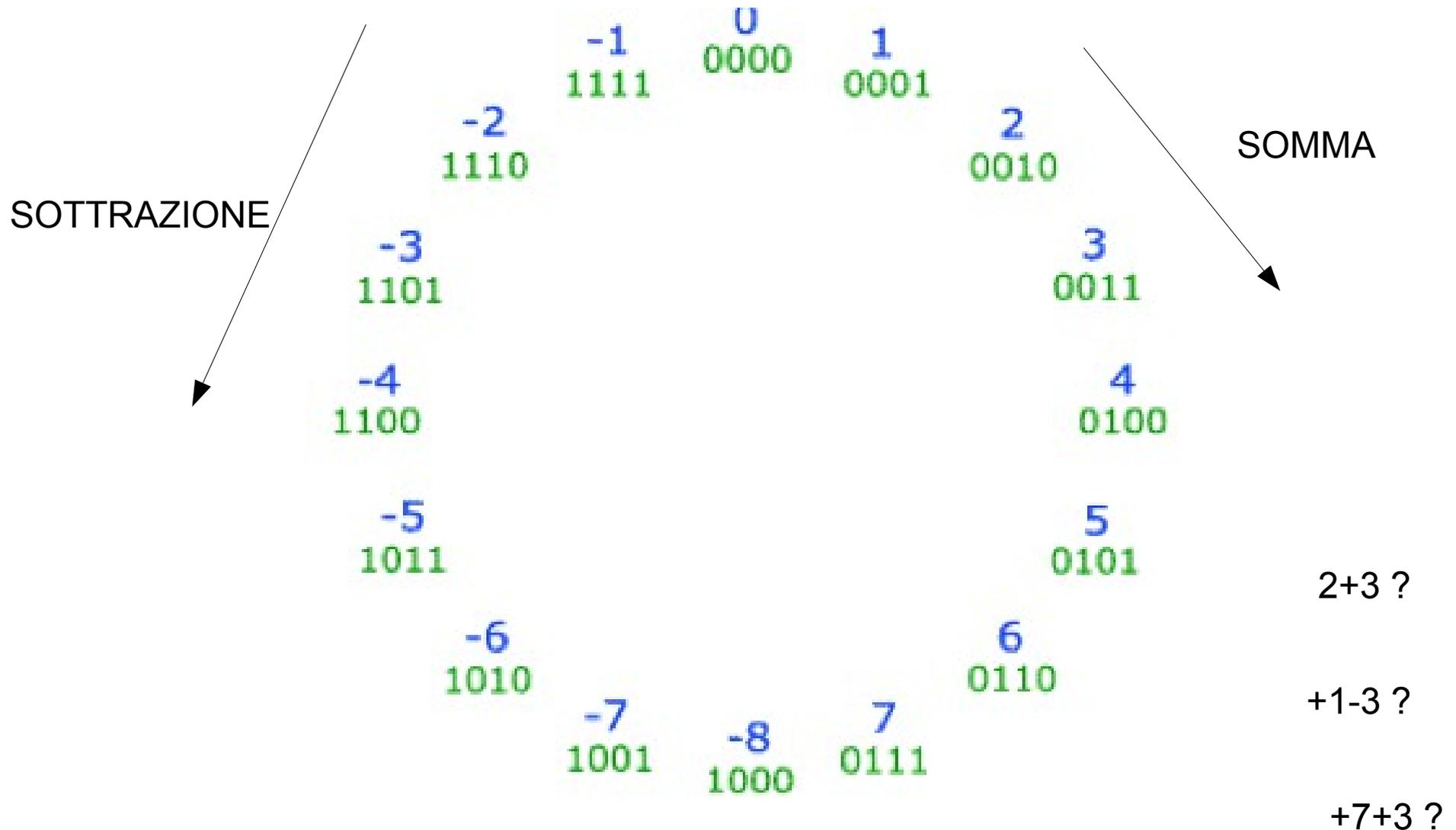


2+3 ?

+1-3 ?

+7+3 ?

Da notare che spostamenti che scavalchino i limiti di
 MAX_VALUE(+7) e MIN_VALUE (-8)
 producono una situazione di overflow.



Calcolo pratico

- Regola del complemento-a-1
- Faccio il complemento di ogni singolo numero
- $1 \rightarrow 0$ e $0 \rightarrow 1$
- Aggiungo 1 e trovo il complemento-a-2

+1

Numero 00000001

Cpl-1 11111110 +

1=

Cpl-2 11111111

-2

Numero 11111110

Cpl-1 00000001 +

1=

Cpl-2 00000010

Tipi interi primitivi

<i>tipo</i>	<i>prec.</i>	<i>MAX_VALUE</i>	<i>MIN_VALUE</i>
byte	8 bit	+127 ($+2^7-1$)	-128 (-2^7)
short	16 bit	+32.767 ($+2^{15}-1$)	-32.768 (-2^{15})
int	32 bit	+2.147.483.647 ($+2^{31}-1$)	-2.147.483.648 (-2^{31})
long	64 bit	+9.223.372.036.854.775.807 ($+2^{63}-1$)	-9.223.372.036.854.775.808 (-2^{63})

Classi per tipi interi

<i>classe</i>	<i>prec.</i>	<i>MAX_VALUE</i>	<i>MIN_VALUE</i>
Byte	8 bit	Come sopra	Come sopra
Short	16 bit	Come sopra	Come sopra
Integer	32 bit	Come sopra	Come sopra
Long	64 bit	Come sopra	Come sopra
BigInteger	arbitraria	Arbitraria	arbitraria