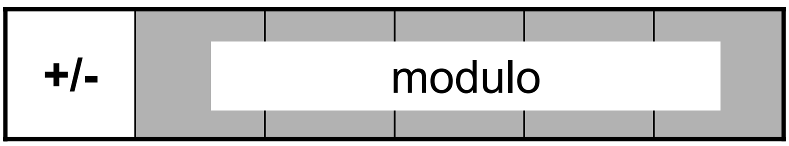
Rappresentazione dei dati

Rappresentazione in segno e modulo
Rappresentazione in complementi alla base
Rappresentazione per eccessi

Rappresentazione dei numeri negativi

- **Soluzione più immediata: segno + modulo**



Possibile convenzione:

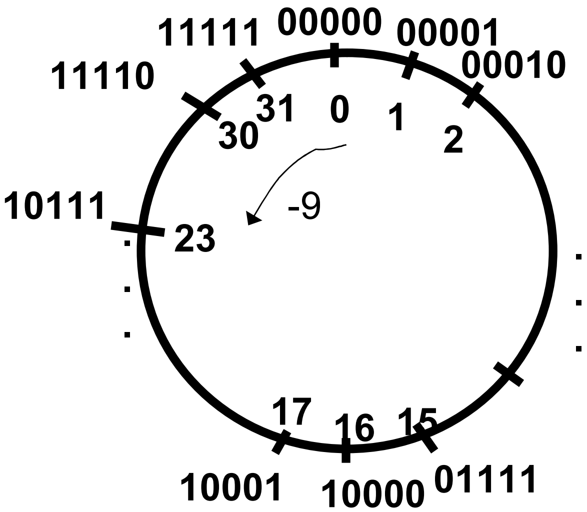
0 → + 1 → -

- **Problemi**
  - dove mettere il segno ?
  - doppia rappresentazione per lo zero (+0, -0)
  - operazioni alquanto complicate

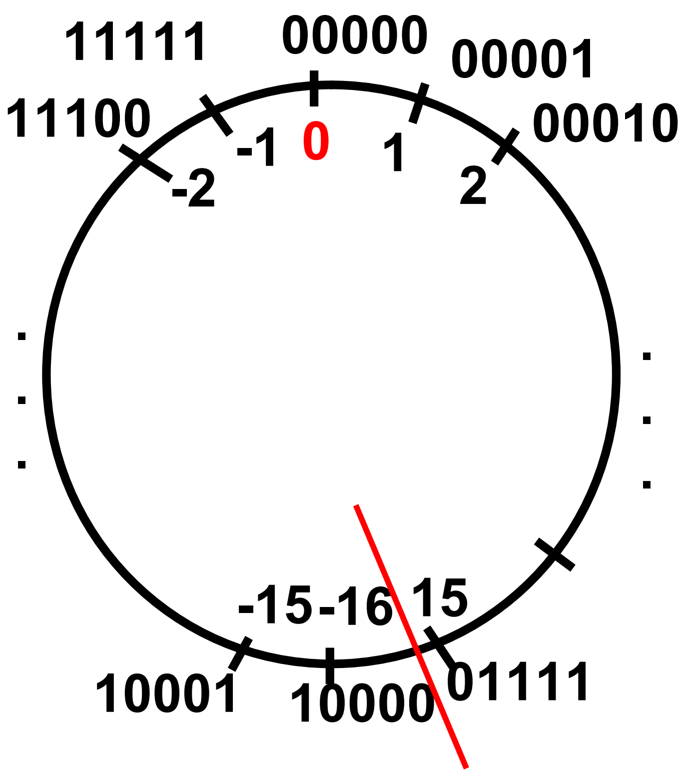
Rappresentazione dei numeri negativi

- **Soluzione alternativa**
  - Che cosa succede in un registro a N bit quando si sottrae un numero da 0 ?

	0	0	0	0	0	-
	0	1	0	0	1	=
1	1	0	1	1	1	



Complementi alla base



Caratteristiche:

- 2^{N-1} non-negativi
- 2^{N-1} negativi
- uno zero
- quanti positivi ?
- confronto ?
- rappr. dello zero

Complementi alla base

- **L'intervallo di numeri rappresentati è $[-2^{N-1} \quad +2^{N-1}-1]$**
- **La rappresentazione di un numero x nell'intervallo è data da $R(x)=(x+2^N) \bmod 2^N$**
- **Il bit più significativo è indicativo del segno ("bit di segno")**

00000	0
00001	+1
00010	+2
00011	+3
.	.
.	.
01110	+14
01111	+15
10000	-16
10001	-15
10010	-14
.	.
.	.
11101	-3
11110	-2
11111	-1

Calcolo rapido del complemento alla base

- **Per ottenere rapidamente la rappresentazione in complemento alla base di un numero negativo su N bit**
  - si estrae la rappresentazione del valore assoluto del numero su N bit
  - si complementano le cifre ad una ad una
  - si aggiunge 1
- **Es.: complemento alla base su 8 bit di -33**

  33_{10} =00100001 11011110+1=11011111

Operazioni in complemento alla base

- Le addizioni si realizzano direttamente sulle rappresentazioni in quanto R(x+y)=R(x)+R(y)
- Anche le sottrazioni si valutano tramite addizioni, ponendo x-y come x+(-y); di conseguenza R(x-y)=R(x)+R(-y)
- Nel caso in cui l'operazione produce un numero al di fuori dell'intervallo di rappresentazione si ha un *overflow*

Operazioni in complemento alla base

+4	0100
+2	+ 0010
+6	0|0110

+4	0100
-2	+1110
+2	1|0010

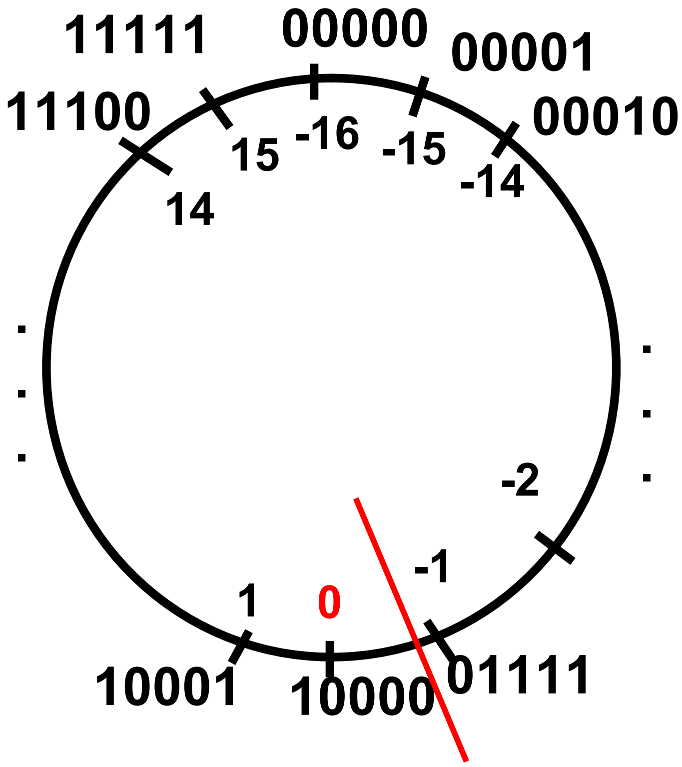
-4	1100
-2	+ 1110
-6	1|1010

+5	0101
+4	+0100
-7	0|1001

-6	1010
-3	+1101
+7	1|0111

overflow

Rappresentazione per eccessi (polarizzata)



Caratteristiche:

- 2^{N-1} non-negativi
- 2^{N-1} negativi
- uno zero
- quanti positivi ?
- confronto ?
- rappr. dello zero

Eccessi

- **L'intervallo di numeri rappresentati è [-2^{N-1} $+2^{N-1}-1$]**
- **La rappresentazione di un numero x nell'intervallo è data da $R(x)=x+2^{N-1}$**
- **Il bit più significativo è indicativo del segno ("bit di segno")**

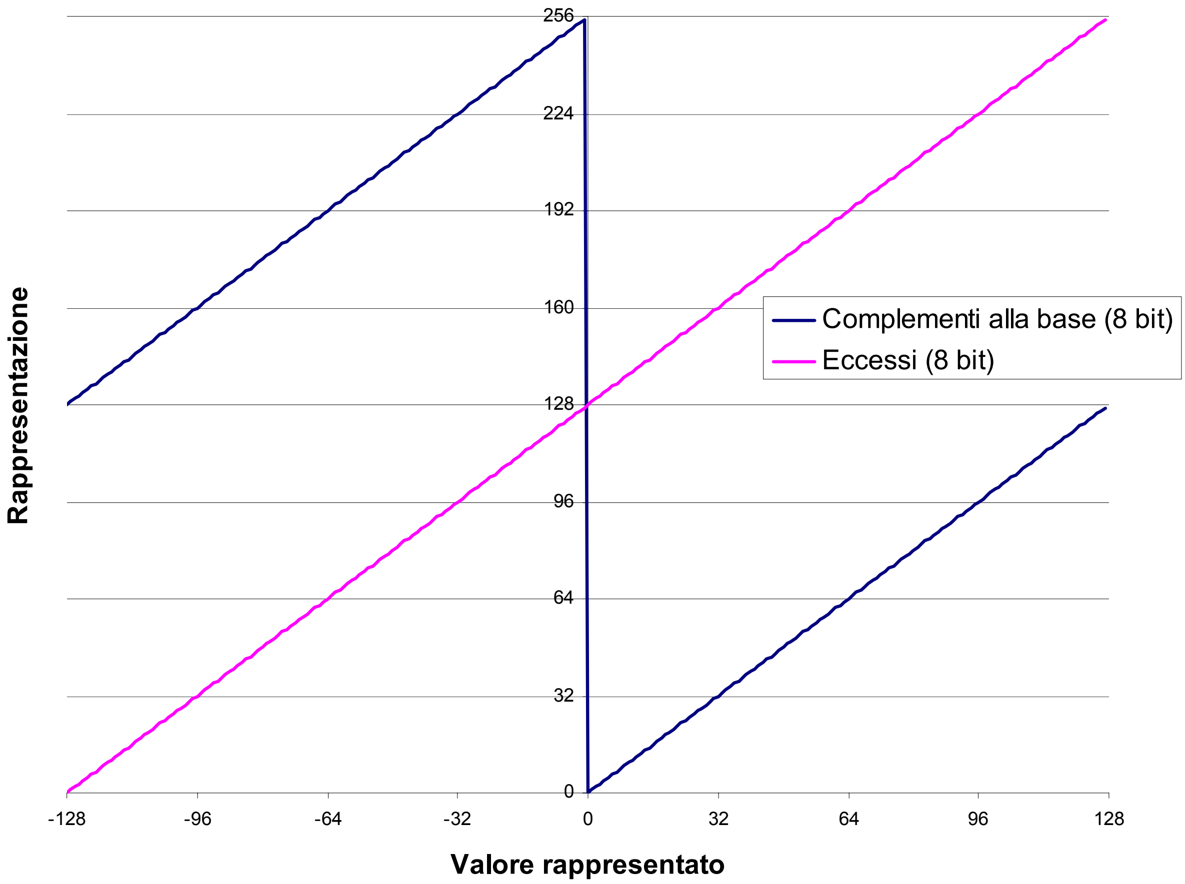
00000	-16
00001	-15
00010	-14
00011	-13
.	.
.	.
01110	-2
01111	-1
10000	0
10001	+1
10010	+2
.	.
.	.
11101	+13
11110	+14
11111	+15

Operazioni in eccessi

- Le addizioni si realizzano direttamente sulle rappresentazioni in quanto R(x+y)=R(x)+R(y)
- Anche le sottrazioni si valutano tramite addizioni, ponendo x-y come x+(-y); di conseguenza R(x-y)=R(x)+R(-y)
- Achtung! Siccome $R(x)+R(y)=x+y+2^{N-1}+2^{N-1}$, il risultato necessita di una correzione
- Nel caso in cui l'operazione produce un numero al di fuori dell'intervallo di rappresentazione si ha un *overflow*

Confronto tra complementi alla base ed eccessi

- Entrambe permettono di realizzare una sottrazione tramite addizione (macchine aritmetiche più semplici)
- Le operazioni in eccessi richiedono un aggiustamento finale
- La rappresentazione in complementi rende più difficile il confronto



Numeri signed e unsigned

- **Un registro di N bit può rappresentare:**
  - Numeri assoluti nel range **[0, 2^N-1]** → numeri ***unsigned***
  - Numeri relativi nel range **[-2^{N-1}, 2^{N-1}-1]** → numeri ***signed***

  } **C**

- **Dalla stringa di bit nel registro non si può risalire al tipo di numero memorizzato. Quali sono le conseguenze ?**
  - Operazioni aritmetiche indipendenti dalla rappresentazione → nessuna conseguenza
  - Confronto dipendente dalla rappresentazione → due tipi di confonto

- **X = 10001 Y = 01110**

  X > Y ?
  - unsigned: SI (17>14)
  - signed: NO (-15<+14)