



ELEMENTI DI INFORMATICA

DOCENTE: FRANCESCO MARRA

INGEGNERIA CHIMICA

INGEGNERIA ELETTRICA

SCIENZE ED INGEGNERIA DEI MATERIALI

INGEGNERIA GESTIONALE DELLA LOGISTICA E DELLA PRODUZIONE

INGEGNERIA NAVALE


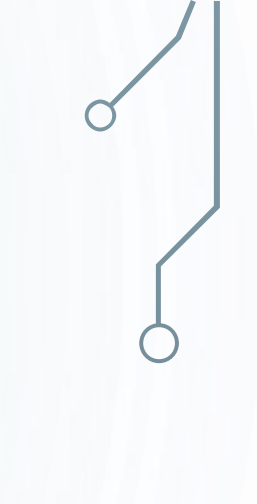
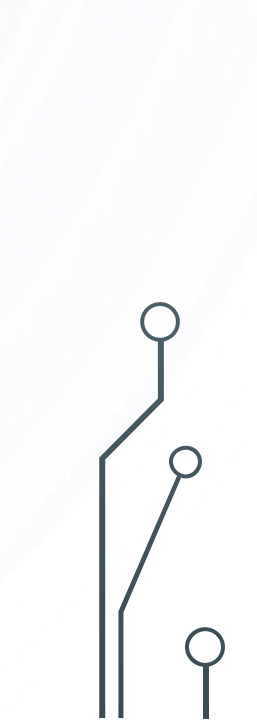
UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE

RAPPRESENTAZIONE NUMERI RELATIVI IN BINARIO





AGENDA

- Rappresentazione dei numeri relativi
 - Segno e modulo
 - Complementi alla base
 - Complemento alla base diminuita
 - Per eccesso
 - Operazioni in rappresentazione binaria
- 
- 
- 

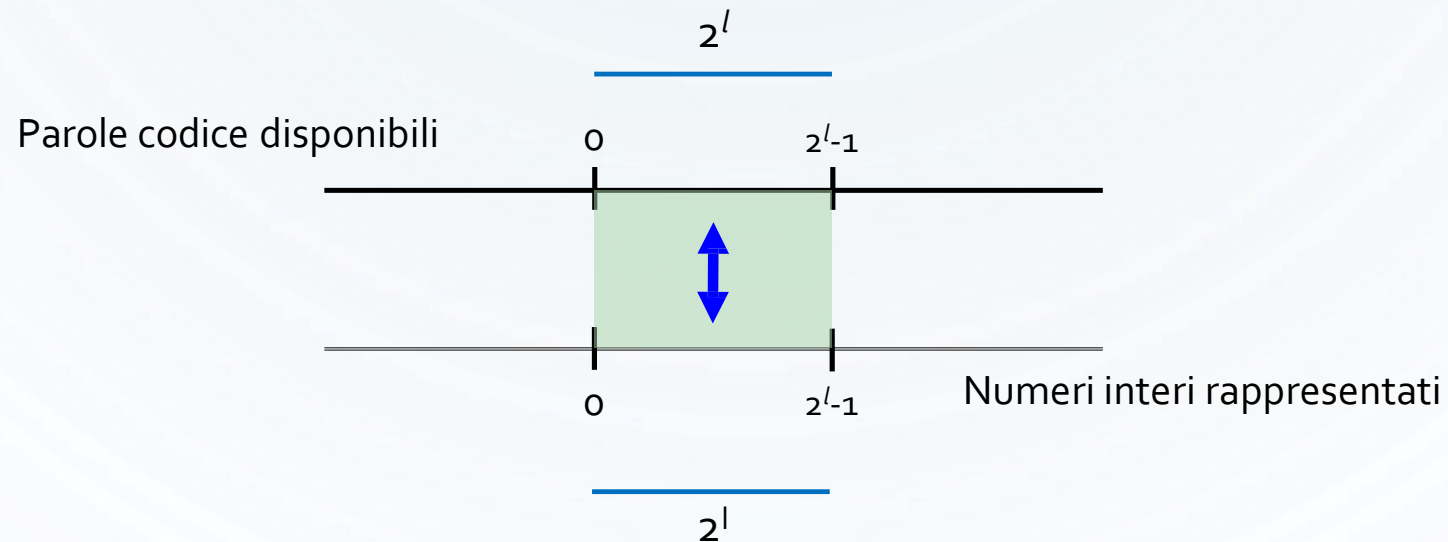
RAPPRESENTAZIONE DEI NUMERI INTERI NON NEGATIVI IN BINARIO

- La rappresentazione finora presentata è detta in “binario puro”
- Avendo l bit a disposizione per le parole codice, tale rappresentazione rende possibile codificare tutti i numeri relativi positivi appartenenti all'intervallo $[0, 2^l - 1]$
- Esempio se $l = 2$

2^l parole codice	Valore decimale rappresentato secondo la codifica in binario puro
00	0
01	1
10	2
11	3

RAPPRESENTAZIONE DEI NUMERI INTERI NON NEGATIVI IN BINARIO

- La rappresentazione finora presentata è detta in “binario puro”
- Avendo l bit a disposizione per le parole codice, tale rappresentazione rende possibile codificare tutti i numeri relativi positivi appartenenti all'intervallo $[0, 2^l - 1]$



LA RAPPRESENTAZIONE DEI NUMERI RELATIVI

- Numeri relativi
 - Numeri interi, positivi e negativi incluso lo zero
- Diverse rappresentazioni
 - Segno e modulo
 - Complemento a due
 - Complemento a uno
 - Per eccesso
- Realizzare circuiti elettronici capaci di effettuare operazioni aritmetiche in un calcolatore in modo **ottimizzato e semplice**

RAPPRESENTAZIONE SEGNO E MODULO

- Data una sequenza di l bit, $l - 1$ bit per rappresentare il modulo del numero intero X
- Bit più a sinistra (MSB) per il segno
 - “0” per indicare un valore positivo
 - “1” per indicarne uno negativo
- Numeri relativi rappresentati nell'intervallo $[-(2^{l-1} - 1), (2^{l-1} - 1)]$
 - $2^{l-1} - 1$ valori positivi, altrettanti negativi, e lo zero
 - $2^l - 1$ valori diversi
- NOTA: è simile alla rappresentazione dei numeri relativi in decimale «classica»

Positivo 01001001
↳ +1001001

Negativo 10101010
↳ -0101010

RAPPRESENTAZIONE *SEGNO E MODULO*

- Esempio $l = 2$

2 ^l parole codice	Valore decimale secondo la codifica binario puro	Valore decimale secondo la codifica segno e modulo
00	0	0
01	1	1
10	2	0
11	3	-1

RAPPRESENTAZIONE SEGNO E MODULO

- Esempio $l = 3$

2 ^l parole codice	Valore decimale secondo la codifica binario puro	Valore decimale secondo la codifica segno e modulo
000	0	0
001	1	1
010	2	2
011	3	3
100	4	-0
101	5	-1
110	6	-2
111	7	-3

RAPPRESENTAZIONE SEGNO E MODULO

- Esempio:

- se $l=8$ e $Y = 73$

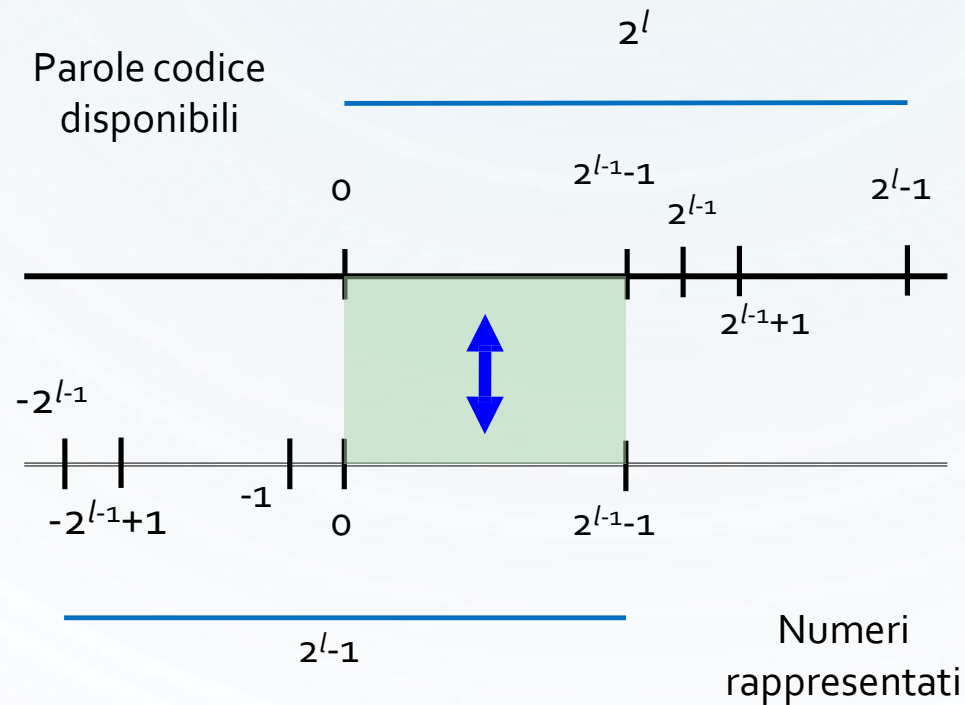
- $|X| = 73 \rightarrow (73)_{10} = (1001001)_2 \rightarrow Y = (\textcolor{red}{0}1001001)_2$

- se $l=8$ e $Y = -73$

- $|X| = 73 \rightarrow (73)_{10} = (1001001)_2 \rightarrow Y = (\textcolor{red}{1}1001001)_2$

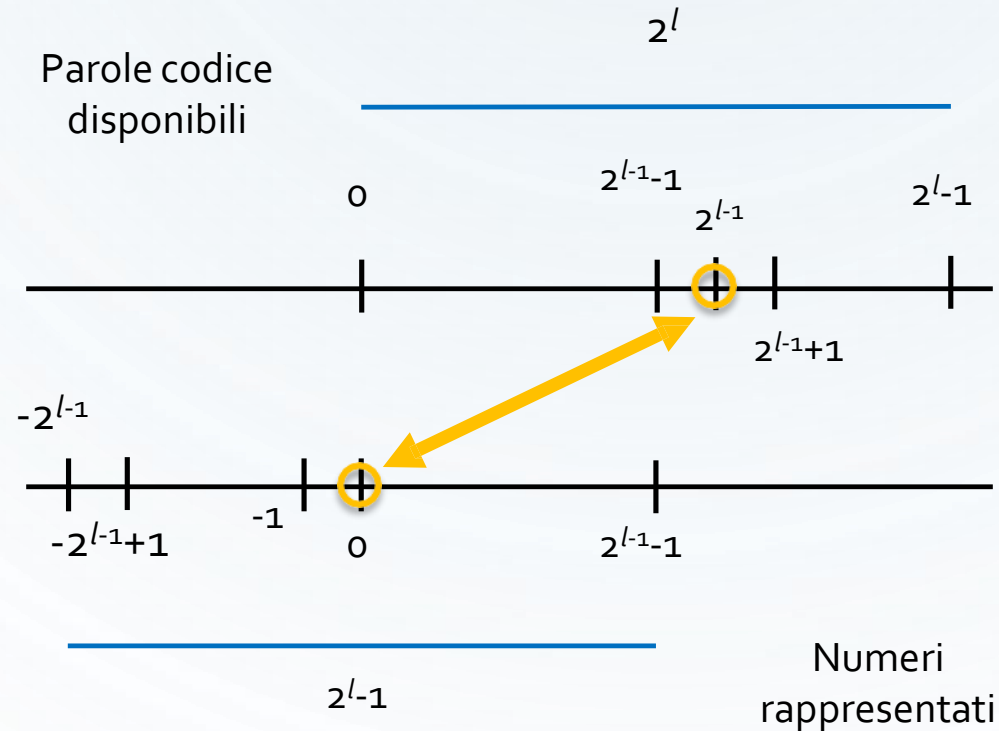
RAPPRESENTAZIONE SEGNO E MODULO

- Numeri relativi rappresentati nell'intervallo $[-2^{l-1} + 1, 2^{l-1} - 1]$
 - 2^l valori diversi includendo i due '0', di cui 2^{l-1} valori positivi e 2^{l-1} negativi



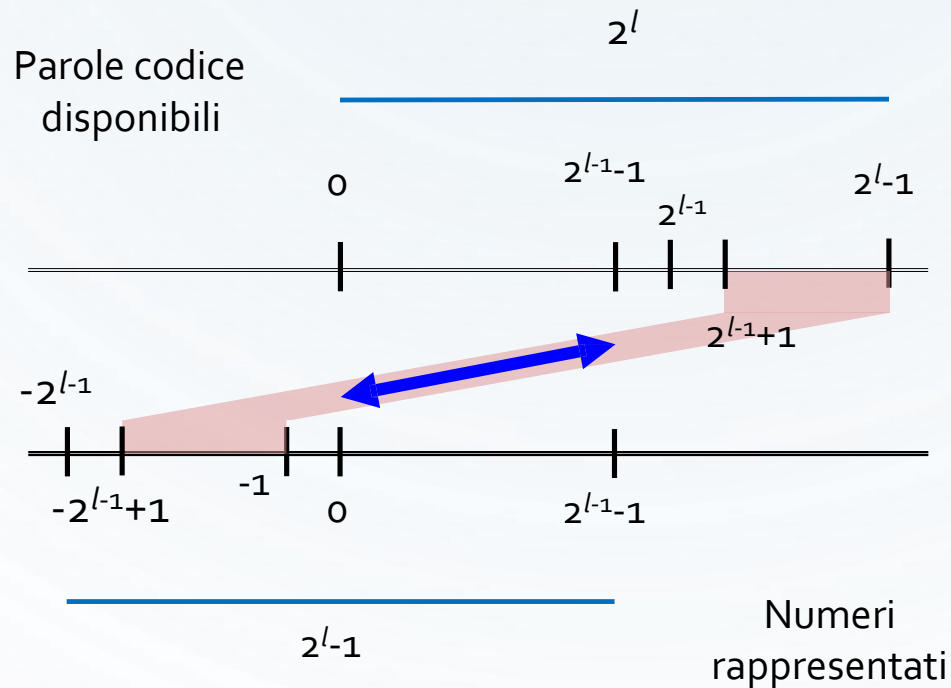
RAPPRESENTAZIONE SEGNO E MODULO

- Numeri relativi rappresentati nell'intervallo $[-2^{l-1} + 1, 2^{l-1} - 1]$
 - 2^l valori diversi includendo i due '0', di cui 2^{l-1} valori positivi e 2^{l-1} negativi



RAPPRESENTAZIONE SEGNO E MODULO

- Numeri relativi rappresentati nell'intervallo $[-2^{l-1} + 1, 2^{l-1} - 1]$
 - 2^l valori diversi includendo i due '0', di cui 2^{l-1} valori positivi e 2^{l-1} negativi



RAPPRESENTAZIONE *SEGNO E MODULO*:ESERCIZIO

- Indicare la rappresentazione in segno e modulo dei seguenti numeri
 - -15 su 5 bit
 - -33 su 6 bit
 - 23 su 7 bit

RAPPRESENTAZIONE *SEGNO E MODULO*:ESERCIZIO

- Indicare la rappresentazione in segno e modulo dei seguenti numeri
 - -15 su 5 bit [R. 11111]
 - -33 su 6 bit [R. Non rappresentabile]
 - 23 su 7 bit [R. 0010111]

RAPPRESENTAZIONE *SEGNO E MODULO*: *PROBLEMI*

- Due configurazioni dello zero
 - “0” positivo (00000000) e “0” negativo (10000000)
- Algoritmi diversi e **complessi** per somma (e sottrazione):
 - Confronto dei segni dei due operandi:
 - Se i segni sono uguali:
 1. Si sommano tutti i bit meno quello del segno
 2. Si aggiunge il bit di segno ai bit ottenuti dalla somma
 - Segni diversi
 1. Si considerano tutti i bit meno quello del segno
 2. Si sottrae il numero più piccolo in modulo dal numero più grande
 3. Si aggiunge ai bit ottenuti dalla sottrazione il bit di segno del numero in modulo più grande

RAPPRESENTAZIONE SEGNO E MODULO: ESEMPIO SOMMA (E SOTTRAZIONE)

- Si vogliono svolgere le seguenti somme in segno e modulo con $l=4$:

$$(-9)_{10} + (-3)_{10}$$

$$(11001)_{2s\&m} + (10011)_{2s\&m}$$

Se i segni sono uguali:

1. Si sommano tutti i bit meno quello del segno
2. Si aggiunge il bit di segno ai bit ottenuti dalla somma

$$1. (1001)_2 + (0011)_2 = (1100)_2$$

$$2. (11100)_2 = (-12)_{10}$$

$$(-4)_{10} + (7)_{10}$$

$$(10100)_{2s\&m} + (00111)_{2s\&m}$$

Se i segni sono diversi:

1. Si considerano tutti i bit meno quello del segno
2. Si sottrae il numero più piccolo in modulo dal numero più grande
3. Si aggiunge ai bit ottenuti dalla sottrazione il bit di segno del numero in modulo più grande

$$1\&2. (0111)_2 - (0100)_2 = (0011)_2$$

$$3. (00011)_2 = (3)_{10}$$

OPERAZIONE DI COMPLEMENTAZIONE

- Data una sequenza di k cifre che rappresenta il numero N in base b , si definisce:

- **complemento alla base** di N il valore

$$C_b = b^k - N$$

- **complemento diminuito di 1** il valore

$$C_d = C_b - 1 = (b^k - 1) - N$$

OPERAZIONE DI COMPLEMENTAZIONE IN BASE 2

- Nel sistema binario ($b=2$)
 - $C_b=C_2$ prende il nome di **complemento a due**
 - $C_d=C_1$ prende il nome di **complemento a uno**
- Note:
 - In alternativa, il complemento a uno di un numero si può ottenere anche invertendo i bit del numero da complementare sostituendo nella sequenza di bit gli "0" con "1" e gli "1" con "0"
 - $C_2 = C_1 + 1$

OPERAZIONE DI COMPLEMENTAZIONE IN BASE 2: ESEMPI

- $N = (001)_2$
 - Complemento a 2
 - $C_2 = (1000)_2 - (001)_2 = (111)_2$
 - Complemento a 1
 - $C_1 = (111)_2 - 1 = (110)_2$

- $N = (100)_2$
 - Complemento a 2
 - $C_2 = (1000)_2 - (100)_2 = (100)_2$
 - Complemento a 1
 - $C_1 = (100)_2 - 1 = (011)_2$

- $N = (111)_2$
 - Complemento a 2
 - $C_2 = (1000)_2 - (111)_2 = (001)_2$
 - Complemento a 1
 - $C_1 = (001)_2 - 1 = (000)_2$

o
p
p
u
r
e

- $N = (001)_2$
 - Complemento a 1
 - $C_1 = (110)_2$
 - Complemento a 2
 - $C_2 = (110)_2 + 1 = (111)_2$

- $N = (100)_2$
 - Complemento a 1
 - $C_1 = (011)_2$
 - Complemento a 2
 - $C_2 = (011)_2 + 1 = (100)_2$

- $N = (111)_2$
 - Complemento a 1
 - $C_1 = (000)_2$
 - Complemento a 2
 - $C_2 = (000)_2 + 1 = (001)_2$

RAPPRESENTAZIONE COMPLEMENTO A UNO

- Per codificare un numero $X \in [-2^{l-1} + 1, 2^{l-1} - 1]$ con una stringa Y di l bit:
 - se $X \in [0, 2^{l-1} - 1]$
 - si usa la codifica *binario puro* di X , se $X \in [0, 2^{l-1} - 1]$
 - se $X \in [-2^{l-1} + 1, -1]$
 - si usa il complemento a uno di X ($2^l - |X| - 1$)
 - in tal caso Y corrisponde alla stringa di bit ottenuta invertendo le cifre della codifica *binario puro* di $|X|$
- N.B. il bit più significativo di Y permette di capire il segno di X

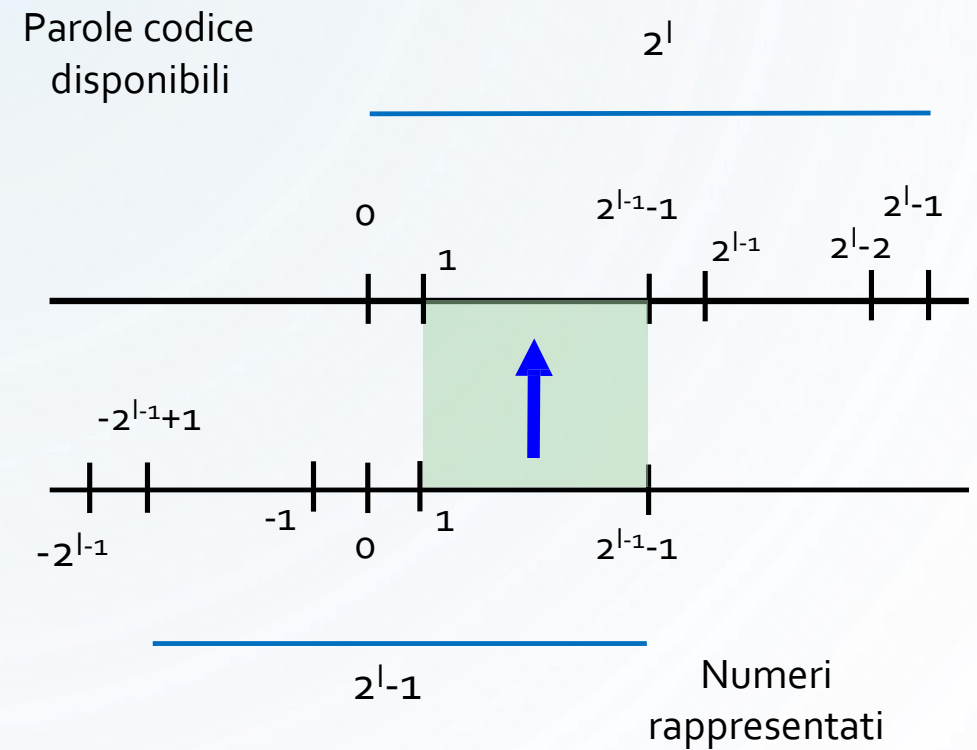
RAPPRESENTAZIONE COMPLEMENTO A UNO

- Esempio

- $X = 7, l=3 \rightarrow X$ non è rappresentabile, in quanto $X \notin [-2^{l-1} + 1, 2^{l-1} - 1] = [-3, 3]$
- $X = 2, l=3 \rightarrow X \in [0, 2^{l-1} - 1] = [0, 3] \rightarrow Y = (010)_2$
- $X = -2, l=3 \rightarrow X \in [-2^{l-1} + 1, -1] = [-3, -1] \rightarrow 2^l - |X| - 1 = 8 - 2 - 1 = 5 \rightarrow Y = (101)_2$

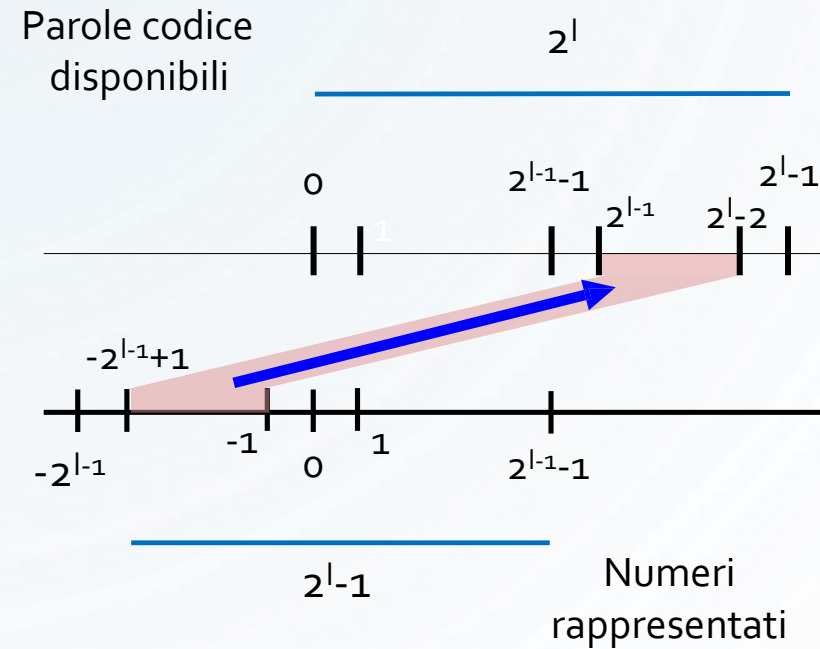
RAPPRESENTAZIONE COMPLEMENTO A UNO

- Numeri positivi
 - Bit più significativo uguale a zero
 - Rappresentati in $[1, 2^{l-1} - 1]$



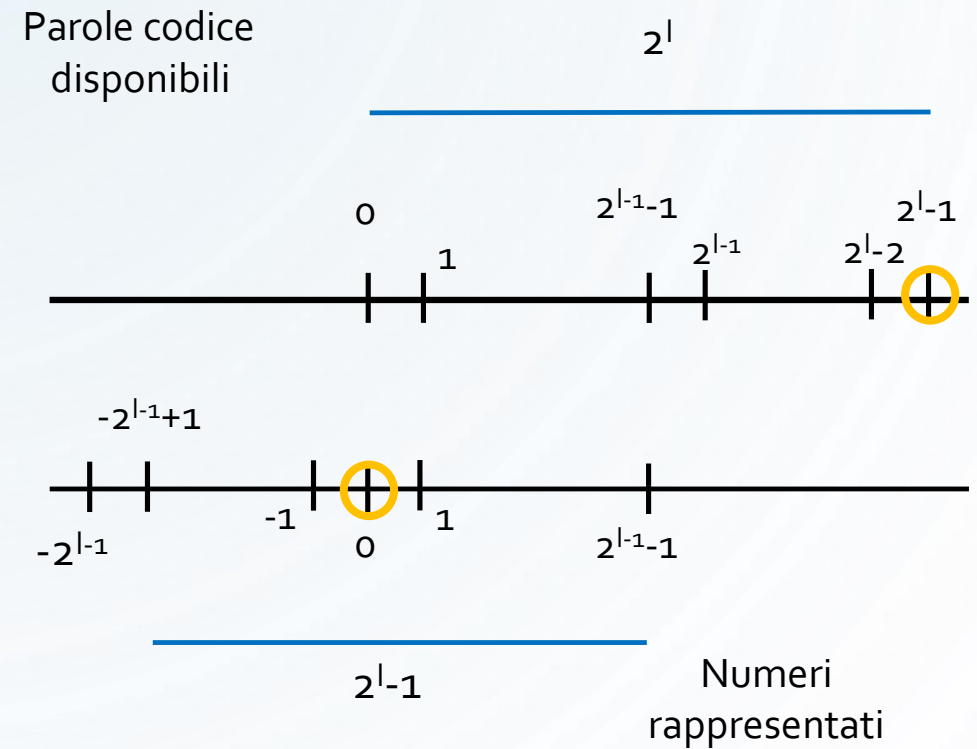
RAPPRESENTAZIONE COMPLEMENTO A UNO

- Numeri positivi
 - Bit più significativo uguale a zero
 - Rappresentati in $[1, 2^{l-1} - 1]$
- Numeri negativi
 - Bit più significativo uguale a uno
 - Rappresentati in $[2^{l-1}, 2^l - 2]$



RAPPRESENTAZIONE COMPLEMENTO A UNO

- Numeri positivi
 - Bit più significativo uguale a zero
 - Rappresentati in $[1, 2^{l-1} - 1]$
- Numeri negativi
 - Bit più significativo uguale a uno
 - Rappresentati in $[2^{l-1}, 2^l - 2]$
- Doppia rappresentazione dello zero
 - 0 e $2^l - 1$



RAPPRESENTAZIONE COMPLEMENTO A UNO

- Esempio $l=2$

2^l parole codice	Valore decimale secondo la codifica binario puro	Valore decimale secondo la codifica segno e modulo	Valore decimale secondo la codifica complemento a uno
00	0	0	0
01	1	1	1
10	2	-0	-1
11	3	-1	-0

RAPPRESENTAZIONE COMPLEMENTO A UNO

- Esempio $l=3$

2 ^l parole codice	Interpretazione binario puro	Interpretazione segno e modulo	Interpretazione complemento a uno
000	0	0	0
001	1	1	1
010	2	2	2
011	3	3	3
100	4	-0	-3
101	5	-1	-2
110	6	-2	-1
111	7	-3	-0

RAPPRESENTAZIONE COMPLEMENTO A UNO

- Esempio $l=4$

2^l parole codice	Interpretazione complemento a uno	2^l parole codice	Interpretazione complemento a uno
0000	0	1000	-7
0001	1	1001	-6
0010	2	1010	-5
0011	3	1011	-4
0100	4	1100	-3
0101	5	1101	-2
0110	6	1110	-1
0111	7	1111	-0

RAPPRESENTAZIONE COMPLEMENTO A UNO

- Considerazioni
 - intervallo simmetrico di numeri rappresentabili $[-2^{l-1} + 1, 2^{l-1} - 1]$
 - il bit più significativo della parola codice indica il segno del numero
 - doppia rappresentazione dello "0" che complica le operazioni di somma e sottrazione
 - semplicità di determinazione della codifica dei numeri negativi
 - ottenuta invertendo i bit della codifica binaria del valore assoluto del numero negativo da codificare

CALCOLO DELLA RAPPRESENTAZIONE IN COMPLEMENTO A UNO DI UN NUMERO NEGATIVO CON UN BYTE

- Esempio: -9
- Metodo 1: $(2^l - |N| - 1)_2 = 2^8 - 9 - 1 = 246$

$(-9)_{c1}$ **1 1 1 1 0 1 1 0**

- Metodo 2: Si fa il complemento a 1 del modulo di N (si invertono i bit di $|N|$)

$(9)_2$	0	0	0	0	1	0	0	1
$(-9)_{c1}$	1	1	1	1	0	1	1	0

DA COMPLEMENTO A UNO A NUMERO DECIMALE

c_7	c_6	c_5	c_4	c_3	c_2	c_1	c_0
-------	-------	-------	-------	-------	-------	-------	-------

- **Numeri non negativi**

- bit di segno c_7 uguale a 0

➡ Normale conversione binario-decimale
 $c_6 * 2^6 + c_5 * 2^5 + \dots + c_1 * 2^1 + c_0 * 2^0$

- **Numeri non positivi**

- bit di segno c_7 uguale a 1

➡ **Metodo 1:** $|N|$ si ottiene sottraendo da 2^l la rappresentazione in complemento a uno e sottraendo 1.

$$(N)_{c1} = 2^8 - |N| - 1$$

$$|N| = 2^8 - 1 - (N)_{c1}$$

➡ **Metodo 2:** il modulo si ottiene facendo il complemento a 1 di tutti i bit

DA COMPLEMENTO A UNO A NUMERO DECIMALE

Rappresentazione in complemento a uno	Numero positivo: <i>Conversione binario-decimale</i>	Modulo numero negativo: <i>Metodo 1</i>	Numero in Decimale
10000000			
00000100			
10010010			
11111111			
00001001			
10101010			

DA COMPLEMENTO A UNO A NUMERO DECIMALE

Rappresentazione in complemento a uno	Numero positivo: <i>Conversione binario-decimale</i>	Modulo numero negativo: <i>Metodo 1</i>	Numero in Decimale
10000000	-	$2^8 - (1*2^7) - 1$	-127
00000100	$1*2^2$		4
10010010		$2^8 - (1*2^7 - 1*2^4 - 1*2^1) - 1$	-109
11111111		$2^8 - (1*2^7 - 1*2^6 - \dots - 1*2^0) - 1$	0
00001001	$1*2^3 + 1*2^0$		9
10101010		$2^8 - (1*2^7 - 1*2^5 - 1*2^3 - 1*2^1) - 1$	-85

DA COMPLEMENTO A UNO A NUMERO DECIMALE

Rappresentazione in complemento a uno	Numero positivo	Modulo numero negativo: Metodo 2		Numero in Decimale
		Complemento a 1	Modulo	
10000000	-	01111111	$1*2^6 + 1*2^5 + 1*2^4 + 1*2^3 + 1*2^2 + 1*2^1 + 1*2^0$	-127
00000100	$1*2^2$		$1*2^2$	4
10010010		01101101	$1*2^6 + 1*2^5 + 1*2^3 + 1*2^2 + 1*2^0$	-109
11111111		00000000	0	-0
00001001	$1*2^3 + 1*2^0$		$1*2^3 + 1*2^0$	9
10101010		01010101	$1*2^6 + 1*2^4 + 1*2^2 + 1*2^0$	-85

RAPPRESENTAZIONE COMPLEMENTO A UNO

ESERCIZIO

- Indicare la rappresentazione in complemento a 1 dei seguenti numeri:
 - -43 su 8 bit
 - 68 su 7 bit
- Indicare a quale numero decimale corrisponde il seguente numero binario codificato mediante rappresentazione in complemento a 1
 - 10101010

RAPPRESENTAZIONE COMPLEMENTO A UNO

ESERCIZIO

- Indicare la rappresentazione in complemento a 1 dei seguenti numeri:
 - -43 su 8 bit [R. 11010100]
 - 68 su 7 bit [R. Non rappresentabile]
- Indicare a quale numero decimale corrisponde il seguente numero binario codificato mediante rappresentazione in complemento a 1
 - 10101010 [R. -85]

RAPPRESENTAZIONE COMPLEMENTO A DUE

- Per codificare un numero $X \in [-2^{l-1}, 2^{l-1} - 1]$ con una stringa Y di l bit:
 - se $X \in [0, 2^{l-1} - 1]$
 - si usa la codifica *binario puro* di X , se $X \in [0, 2^{l-1} - 1]$
 - se $X \in [-2^{l-1}, -1]$
 - si usa il complemento a due di X ($2^l - |X|$)
- N.B. il bit più significativo di Y permette di capire il segno di X

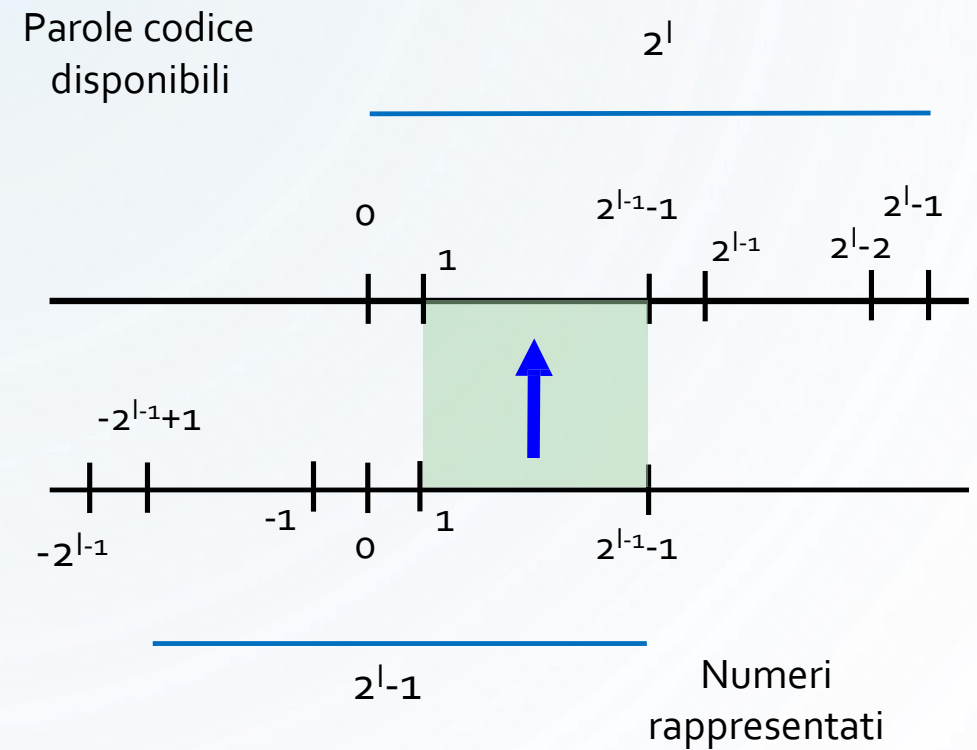
RAPPRESENTAZIONE COMPLEMENTO A DUE

- Esempio

- $X = 10, l=4 \rightarrow X$ non è rappresentabile, in quanto $X \notin [-2^{l-1}, 2^{l-1} - 1] = [-8, 7]$
- $X = 5, l=4 \rightarrow X \in [0, 2^{l-1} - 1] = [0, 7] \rightarrow Y = (0101)_2$
- $X = -5, l=4 \rightarrow X \in [-2^{l-1}, -1] = [-8, -1] \rightarrow 2^l - |X| = 16 - 5 = 11 \rightarrow Y = (1011)_2$
 - Oppure $Y = (|X|)_{C1} + 1 = (0101)_{C1} + 1 = (1010)_2 + 1 = (1011)_2$

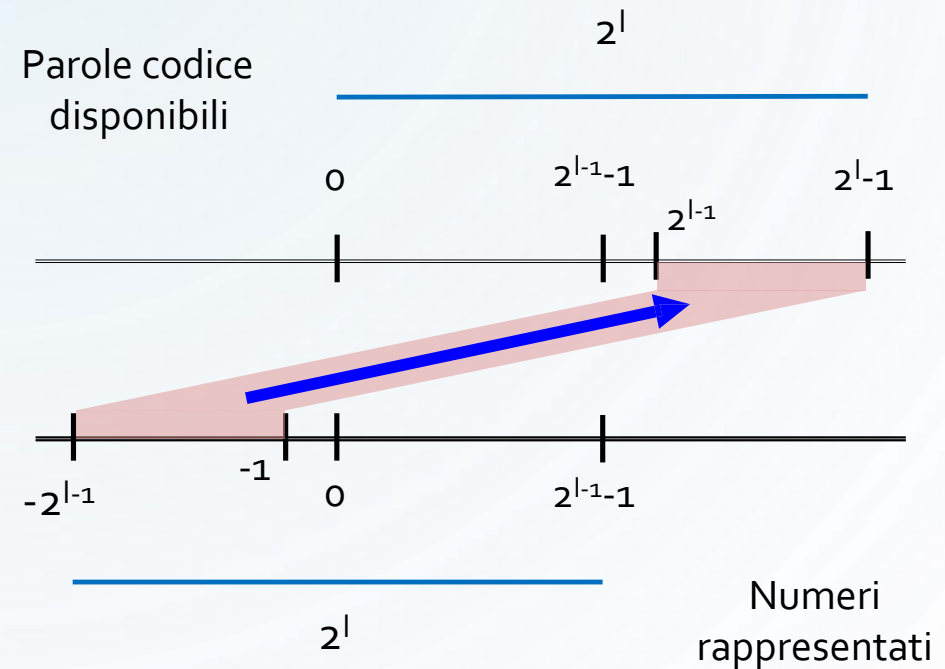
RAPPRESENTAZIONE COMPLEMENTO A DUE

- Numeri positivi
 - Bit più significativo uguale a zero
 - Rappresentati in $[1, 2^{l-1} - 1]$



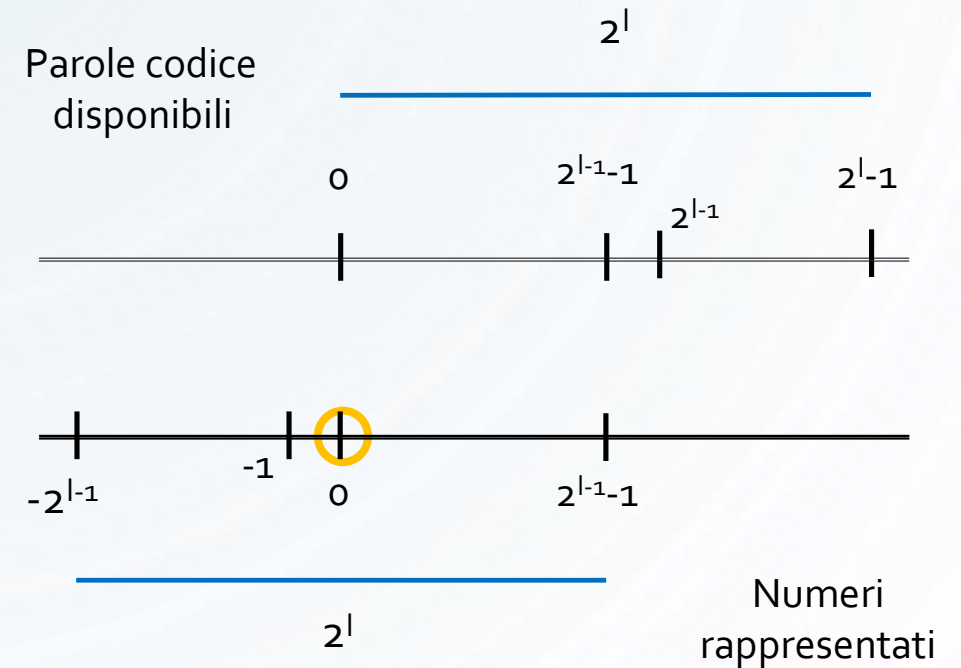
RAPPRESENTAZIONE COMPLEMENTO A DUE

- Numeri positivi
 - Bit più significativo uguale a zero
 - Rappresentati in $[1, 2^{l-1} - 1]$
- Numeri negativi
 - Bit più significativo uguale a uno
 - Rappresentati in $[2^{l-1}, 2^l - 1]$
 - intervallo $[-2^{l-1}, -1]$ traslato a destra di 2^l



RAPPRESENTAZIONE COMPLEMENTO A DUE

- Numeri positivi
 - Bit più significativo uguale a zero
 - Rappresentati in $[1, 2^{l-1} - 1]$
- Numeri negativi
 - Bit più significativo uguale a uno
 - Rappresentati in $[2^{l-1}, 2^l - 1]$
 - intervallo $[-2^{l-1}, -1]$ traslato a destra di 2^l
- Singola rappresentazione dello zero
 - Tutti i bit a 0



RAPPRESENTAZIONE COMPLEMENTO A DUE

- Esempio $l=2$

2^l parole codice	Valore decimale secondo la codifica binario puro	Valore decimale secondo la codifica segno e modulo	Valore decimale secondo la codifica complemento a uno	Valore decimale secondo la codifica complemento a due
00	0	0	0	0
01	1	1	1	1
10	2	-0	-1	-2
11	3	-1	-0	-1

RAPPRESENTAZIONE COMPLEMENTO A DUE

- Esempio $l=3$

2^l parole codice	Interpretazione binario puro	Interpretazione segno e modulo	Interpretazione complemento a uno	Interpretazione complemento a due
000	0	0	0	0
001	1	1	1	1
010	2	2	2	2
011	3	3	3	3
100	4	-0	-3	-4
101	5	-1	-2	-3
110	6	-2	-1	-2
111	7	-3	-0	-1

RAPPRESENTAZIONE COMPLEMENTO A DUE

- Esempio $l=4$

2^l parole codice	Interpretazione complemento a due	2^l parole codice	Interpretazione complemento a due
0000	0	1000	-8
0001	1	1001	-7
0010	2	1010	-6
0011	3	1011	-5
0100	4	1100	-4
0101	5	1101	-3
0110	6	1110	-2
0111	7	1111	-1

RAPPRESENTAZIONE COMPLEMENTO A DUE

- Considerazioni
 - intervallo non simmetrico di numeri rappresentabili $[-2^{l-1}, 2^{l-1} - 1]$
 - il bit più significativo della parola codice indica il segno del numero
 - si elimina la ridondanza nella rappresentazione dello zero
 - notevoli vantaggi, soprattutto per effettuare somme e differenze
 - semplicità di determinazione della codifica dei numeri negativi
 - Calcolo diretto $(2^l - |N|)_2$
 - passando per il complemento a uno:
 - sommare 1 alla stringa di bit ottenuta invertendo i bit della codifica binaria del valore assoluto del numero negativo da codificare

CALCOLO DELLA RAPPRESENTAZIONE IN COMPLEMENTO A DUE DI UN NUMERO NEGATIVO CON UN BYTE

- Esempio: -9
- Metodo 1: $(2^l - |N|)_2 = 2^8 - 9 = 247$

$(-9)_{c2}$

1	1	1	1	0	1	1	1
---	---	---	---	---	---	---	---

- Metodo 2: Si fa il complemento a 1 del modulo di N (si invertono i bit di $|N|$)

$(9)_2$

0	0	0	0	1	0	0	1
---	---	---	---	---	---	---	---

$(-9)_{c1}$

1	1	1	1	0	1	1	0
---	---	---	---	---	---	---	---

 +

$(-9)_{c2}$

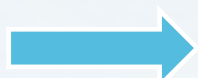
1	1	1	1	0	1	1	1
---	---	---	---	---	---	---	---

DA COMPLEMENTO A DUE A NUMERO DECIMALE

c_7	c_6	c_5	c_4	c_3	c_2	c_1	c_0
-------	-------	-------	-------	-------	-------	-------	-------

- **Numeri non negativi**

- bit di segno c_7 uguale a 0

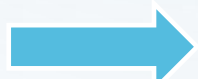


Normale conversione binario-decimale

$$c_6 * 2^6 + c_5 * 2^5 + \dots + c_1 * 2^1 + c_0 * 2^0$$

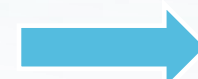
- **Numeri non positivi**

- bit di segno c_7 uguale a 1



Metodo 1: conversione binario-decimale dando al bit di segno un peso negativo nel sistema di numerazione posizionale

$$-1 * 2^7 + c_6 * 2^6 + c_5 * 2^5 + \dots + c_1 * 2^1 + c_0 * 2^0$$



Metodo 2: negativo della somma tra 1 e il numero ottenuto dalla conversione *binario-decimale invertendo tutti i bit*

DA COMPLEMENTO A DUE A NUMERO DECIMALE

Rappresentazione in complemento a uno	Numero negativo: <i>Metodo 1</i>	Numero negativo: <i>Metodo 2</i>	Numero in Decimale
10000000	$-(01111111 + 1)$		
00000100	$1 * 2^2$		
10010010	$-(01101101 + 1)$		
11111111	$-(00000000 + 1)$		
00001001	$1 * 2^3 + 1 * 2^0$		
10101010	$-(01010101 + 1)$		

DA COMPLEMENTO A DUE A NUMERO DECIMALE

Rappresentazione in complemento a uno	Numero negativo: <i>Metodo 1</i>	Numero negativo: <i>Metodo 2</i>	Numero in Decimale
10000000	$-(01111111 + 1)$		-128
00000100	$1*2^2$		4
10010010	$-(01101101 + 1)$		-110
11111111	$-(00000000 + 1)$		0
00001001	$1*2^3 + 1*2^0$		9
10101010	$-(01010101 + 1)$		-86

DA COMPLEMENTO A DUE A NUMERO DECIMALE

Rappresentazione in complemento a uno	Numero negativo: <i>Metodo 1</i>	Numero negativo: <i>Metodo 2</i>	Numero in Decimale
10000000	$-(01111111 + 1)$	$-1*2^7$	-128
00000100	$1*2^2$	$1*2^2$	4
10010010	$-(01101101 + 1)$	$-1*2^7 + 1*2^4 + 1*2^1$	-110
11111111	$-(00000000 + 1)$	$-1*2^7 + 1*2^6 + \dots + 1*2^0$	0
00001001	$1*2^3 + 1*2^0$	$1*2^3 + 1*2^0$	9
10101010	$-(01010101 + 1)$	$-1*2^7 + 1*2^5 + 1*2^3 + 1*2^1$	-86

RAPPRESENTAZIONE COMPLEMENTO A DUE

ESERCIZIO

- Indicare la rappresentazione in complemento a 2 dei seguenti numeri:
 - -43 su 8 bit
 - 68 su 6 bit
- Indicare a quale numero decimale corrisponde il seguente numero binario codificato mediante rappresentazione in complemento a 2
 - 101111

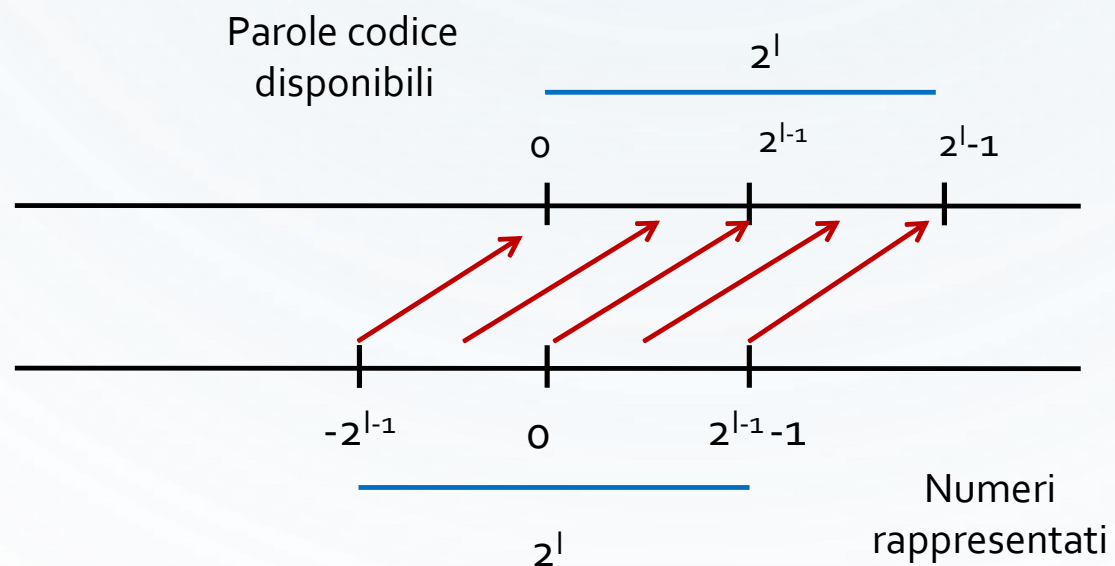
RAPPRESENTAZIONE COMPLEMENTO A DUE

ESERCIZIO

- Indicare la rappresentazione in complemento a 2 dei seguenti numeri:
 - -43 su 8 bit [R. 11010101]
 - 68 su 6 bit [R. Non rappresentabile]
- Indicare a quale numero decimale corrisponde il seguente numero binario codificato mediante rappresentazione in complemento a 2
 - 101111 [R. -17]

RAPPRESENTAZIONE PER ECCESSO 2^{l-1}

- Data una sequenza di l bit:
 - un numero intero X è codificato come codifica binaria della somma di se stesso con 2^{l-1}
 - uguale al complemento a due, ma con il bit di segno invertito
 - numeri compresi in $[-2^{l-1}, 2^{l-1} - 1]$ mappati su $[0, 2^l - 1]$



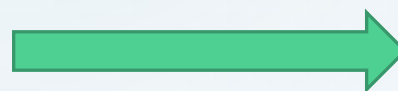
RAPPRESENTAZIONE PER ECCESSO 2^{l-1}

- Esempio

- $X = (+12)_{10}, l = 8 \text{ bit}, 2^{l-1} = 128$

- $((12 + 128) = 140)_{10}$

Binario puro



$$(10001100)_2$$

- $X = (-12)_{10}, l = 8 \text{ bit}, 2^{l-1} = 128$

- $((-12 + 128) = 116)_{10}$



$$(01110100)_2$$

RAPPRESENTAZIONE PER ECCESSO 2^{l-1}

- Numeri relativi rappresentati nell'intervallo $[-2^{l-1}, 2^{l-1}-1]$
 - intervallo non simmetrico
 - 2^l valori diversi
 - una sola rappresentazione dello zero
 - $2^{l-1}-1$ valore del massimo
 - 2^{l-1} valore assoluto del minimo
- Per $l = 8$ i numeri appartenenti a $[-128, 127]$ mappati in $[0, 255]$
 - Numeri negativi mappati in $[0, 127]$, lo zero su 128, e numeri positivi in $[129, 255]$

$(x)_{10}$	$x + 2^{l-1}$ ($l=3$)
-4	000
-3	001
-2	010
-1	011
0	100
1	101
2	110
3	111

RAPPRESENTAZIONE PER ECCESSO 2^{l-1}

- Numeri relativi rappresentati nell'intervallo $[-2^{l-1}, 2^{l-1}-1]$

- intervallo non simmetrico
- 2^l valori diversi
- una sola rappresentazione dello zero
- $2^{l-1}-1$ valore del massimo
- 2^{l-1} valore assoluto del minimo

- Per $l = 8$ i numeri appartenenti a $[-128, 127]$ mappati in $[0, 255]$

- Numeri negativi mappati in $[0, 127]$, lo zero su 128, e numeri positivi in $[129, 255]$

$(x)_{10}$	$x + 2^{l-1}$ ($l=4$)	$(x)_{10}$	$x + 2^{l-1}$ $C_2(l=4)$
-8	0000	0	1000
-7	0001	1	1001
-6	0010	2	1010
-5	0011	3	1011
-4	0100	4	1100
-3	0101	5	1101
-2	0110	6	1110
-1	0111	7	1111

RAPPRESENTAZIONE PER ECCESSO 2^{L-1}

ESERCIZIO

- Indicare la rappresentazione per eccesso 128 del numero -36 su 8 bit
- Indicare la rappresentazione per eccesso 64 del numero -21 su 7 bit
- Indicare a quale numero decimale corrisponde il numero binario 0110100 codificato mediante rappresentazione per eccesso 64

RAPPRESENTAZIONE PER ECCESSO 2^{L-1}

ESERCIZIO

- Indicare la rappresentazione per eccesso 128 del numero -36 su 8 bit
 - 01011100
- Indicare la rappresentazione per eccesso 64 del numero -21 su 7 bit
 - 0101011
- Indicare a quale numero decimale corrisponde il numero binario 0110100 codificato mediante rappresentazione per eccesso 64
 - -12

ESEMPI DI CODIFICA CON 8 BIT NELLE QUATTRO RAPPRESENTAZIONI

Valore decimale di N	N in binario	N in segno e modulo	N in complemento a 1	N in complemento a 2	N per eccesso 2^7
-128					
-127					
-100					
-10					
-1					
0					
1					
10					
100					
127					
128					

ESEMPI DI CODIFICA CON 8 BIT NELLE QUATTRO RAPPRESENTAZIONI

Valore decimale di N	N in binario	N in segno e modulo	N in complemento a 1	N in complemento a 2	N per eccesso 2^7
-128	10000000	Non esiste	Non esiste	10000000	00000000
-127	01111111	11111111	10000000	10000001	00000001
-100	01100100	11100100	10011011	10011100	00011100
-10	00001010	10001010	11110101	11110110	01110110
-1	00000001	10000001	11111110	11111111	01111111
0	00000000	10000000 00000000	11111111 00000000	00000000	10000000
1	00000001	00000001	00000001	00000001	10000001
10	00001010	00001010	00001010	00001010	10001010
100	01100100	01100100	01100100	01100100	11100100
127	01111111	01111111	01111111	01111111	11111111
128	10000000	Non esiste	Non esiste	Non esiste	Non esiste

RAPPRESENTAZIONI PIÙ UTILIZZATE

- Le rappresentazioni in:
 - Complemento a due
 - Eccesso 2^{l-1}
- Sono le più efficienti per svolgere operazioni aritmetiche

OPERAZIONI IN RAPPRESENTAZIONE BINARIA

- Usando la rappresentazione in complemento a due o quella per eccesso 2^{l-1} :
 - Algoritmo unico per somma e sottrazione
 - Sottrazione tra numeri come somma tra numeri di segno opposto
 - $(X - Y) = (X + (-Y))$
- Esempio di addizione in complemento a due

13 +
-20 =
-7

0	0	0	0	1	1	0	1
1	1	1	0	1	1	0	0
1	1	1	1	1	0	0	1

OPERAZIONI IN RAPPRESENTAZIONE BINARIA: COMPLEMENTO A DUE

- Operandi stesso segno (si ignora il «trabocco»):

4 +		0	1	0	0
<u>2 =</u>		0	0	1	0
6	0	0	1	1	0

-4 +		1	1	0	0
<u>-2 =</u>		1	1	1	0
-6	1	1	0	1	0

OPERAZIONI IN RAPPRESENTAZIONE BINARIA: COMPLEMENTO A DUE

- Operandi stesso segno (si ignora il «trabocco»):

4 +		0	1	0	0
<u>2 =</u>		0	0	1	0
6	0	0	1	1	0

-4 +		1	1	0	0
<u>-2 =</u>		1	1	1	0
-6	1	1	0	1	0

- Operandi stesso segno (si ha errore se il risultato ha segno discorde):

6 +		0	1	1	0
<u>3 =</u>		0	0	1	1
9	0	1	0	0	1

-6 +		1	0	1	0
<u>-3 =</u>		1	1	0	1
-9	1	0	1	1	1

OPERAZIONI IN RAPPRESENTAZIONE BINARIA: COMPLEMENTO A DUE

- Operandi segno diverso il risultato è sempre corretto:

4 +		0	1	0	0
<u>-2 =</u>		1	1	1	0
2	1	0	0	1	0

-4 +		1	1	0	0
<u>2 =</u>		0	0	1	0
- 2	0	1	1	1	0

OPERAZIONI IN RAPPRESENTAZIONE BINARIA: ECESSO 2^{l-1}

- Esempio di addizione in rappresentazione per eccesso 2^{l-1} :

$$\begin{array}{r} 13 + \\ -20 = \\ \hline -7 \end{array}$$

1	0	0	0	1	1	0	1
0	1	1	0	1	1	0	0
01	1	1	1	1	0	0	1

dato che durante l'addizione abbiamo sommato due volte l'eccesso 2^{l-1} , dobbiamo sottrarlo dal risultato

00	1	1	1	1	0	0	1
-----------	----------	----------	----------	----------	----------	----------	----------

$$\begin{array}{r} 13 + \\ -20 = \\ \hline -7 \end{array} \quad \rightarrow \quad \begin{array}{r} 13 + 128 + \\ -20 + 128 = \\ \hline -7 + 256 \end{array}$$

$$\begin{array}{r} -7 + 256 + \\ -128 = \\ \hline -7 + 128 \end{array}$$

OPERAZIONI IN RAPPRESENTAZIONE BINARIA

- Esempio di addizione in rappresentazione per eccesso 2^{l-1} :

$$\begin{array}{r} 13 + \\ +20 = \\ +33 \end{array}$$

1	0	0	0	1	1	0	1
1	0	0	1	0	1	0	0
10	0	1	0	0	0	0	1

dato che durante l'addizione abbiamo sommato due volte l'eccesso 2^{l-1} , dobbiamo sottrarlo dal risultato

01	0	1	0	0	0	0	1
----	---	---	---	---	---	---	---

$$\begin{array}{r} 13 + \\ +20 = \\ +33 \end{array} \rightarrow \begin{array}{r} 13 + 128 + \\ +20 + 128 = \\ +33 + 256 \end{array}$$

$$\begin{array}{r} +33 + 256 + \\ - 128 = \\ +33 + 128 \end{array}$$

DOMANDE, DUBBI, PERPLESSITÀ

