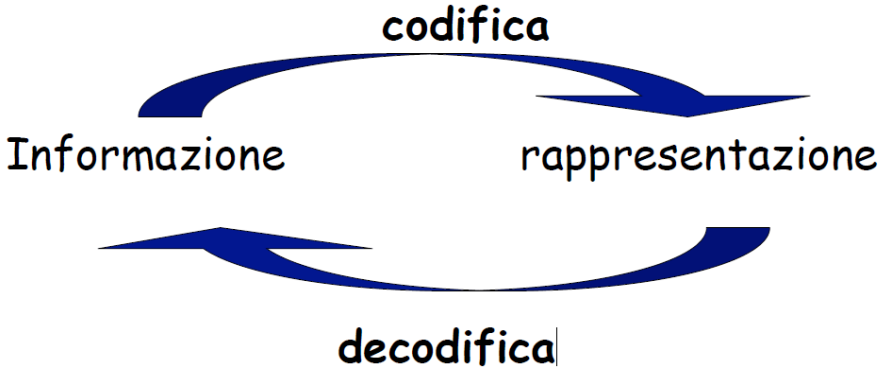
Rappresentazione delle informazioni

Abbiamo informazioni (numeri, caratteri, immagini, suoni, video. . .) che vogliamo rappresentare (e poter elaborare) in un calcolatore. Per motivi tecnologici un calcolatore lavora solo con i valori 0 e 1 e quindi tutte le informazioni sono rappresentate in forma binaria.



CODIFICA: l'operazione di formalizzazione e scrittura dell'informazione su supporto fisico.
DECODIFICA: l'operazione di interpretazione e lettura dell'informazione da supporto fisico.

Quanti oggetti posso codificare con k bit?
->1 bit -> (0, 1) -> 2 oggetti
->2 bit -> (00, 01, 10, 11) -> 4 oggetti
->3 bit -> (000, 001, 010, …, 111) -> 8 oggetti
->…
->k bit -> (…)-> 2^k oggetti

Domanda: quanti diversi valori posso rappresentare con 2 byte?
Risposta: 2 byte = 16 bit, quindi posso rappresentare 2^{16} = 65536 diversi valori.

Quanti bit mi servono per codificare N oggetti?
->Devo trovare quel numero K tale che……N <= 2^K

Domanda: quanti bit mi servono per rappresentare 112 diversi valori?
Risposta: 7 bit ($2^7 = 128$). 6 bit sarebbero stati pochi, mentre 8 bit sarebbero stati troppi!

Rappresentazione dell'informazione numerica

- Numeri naturali (insieme N)
- Numeri interi (insieme Z)
- Numeri razionali (insieme Q)

……..

La precisione con cui i numeri possono essere espressi è finita e predeterminata poiché questi devono essere memorizzati entro un limitato spazio di memoria.
Per rappresentare i numeri si utilizza il sistema binario poiché più adatto a essere maneggiato dal calcolatore.
I numeri a precisione finita sono quelli rappresentati con un numero finito di cifre.

NUMERI NATURALI

Per i numeri naturali si usa la ***rappresentazione binaria posizionale:***
$(101100)_2 = (44)_{10}$

Con n bit si possono rappresentare 2^n diversi numeri naturali. Quali sono?
I numeri rappresentabili appartengono all'intervallo:

$$(0;\ 2^n - 1)$$

n è la dimensione (in bit) della cella di memoria che contiene il numero

Insieme dei valori rappresentabili:
con 1 byte si possono rappresentare i num tra **$(0;\ 2^8 - 1)$ cioè tra (0; 255)**
con 2 byte **$(0;\ 2^{16} - 1)$…..(0; 65535)**
con 4 byte **$(0;\ 2^{32} - 1)$…..$(0;\ 4*10^9)$** *** **la più utilizzata** ***
con 8 byte **$(0;\ 2^{64} - 1)$…….(0; ????)**

Dato che lo spazio disponibile è finito, vi sono dei limiti nella dimensione dei numeri memorizzabili ….

Le operazioni con i numeri a precisione finita causano errori quando il loro risultato non appartiene all'insieme dei valori rappresentabili:

- Underflow: si verifica quando il risultato dell'operazione è minore del più piccolo valore rappresentabile
- Overflow: si verifica quando il risultato dell'operazione è maggiore del più grande valore rappresentabile
- Non appartenenza all'insieme: si verifica quando il risultato dell'operazione, pur non essendo troppo grande o troppo piccolo, non appartiene all'insieme dei valori rappresentabili

Esempio: si considerino i numeri naturali di tre cifre, non possono essere rappresentati:

- Numeri superiori a 999
- Numeri negativi
- Frazioni e numeri irrazionali

Alcuni errori possibili in operazioni fra tali numeri:

- 600+600 = 1200 Overflow
- 300-600 = -300 Underflow
- 007/002 = 3.5 Non appartenenza all'insieme

Valore minimo di una sequenza di n cifre binarie: 000 … 0 (n volte) = 0^{10}
Valore massimo di una sequenza di n cifre binarie: 1111…111 (n volte) = $2^n - 1$

Esempio con n = 3: $111 = 2^2 + 2 + 1 = 7 = 2^3 - 1$

NUMERI INTERI

Modulo e segno (MS)

➔E' indispensabile indicare il numero K di bit utilizzati per la rappresentazione
➔Il bit più a sinistra (il + significativo) rappresenta il segno del numero: 0 = '+' , 1 = '-'
➔I rimanenti k-1 bit rappresentano il modulo

Come si converte da decimale a Modulo e Segno? (10 -> MS)

Si calcola la rappresentazione binaria del valore assoluto del numero senza il bit del segno poi se il numero è negativo si pone = a 1 il bit del segno se il numero è positivo si pone = a 0 il bit del segno.
NB: il bit del segno non ha significato numerico

Esempi:

- se utilizzo 4 bit…..
  +7 = $(0111)_{ms}$, -7 = $(1111)_{ms}$
  +6 = $(0110)_{ms}$, -6 = $(1110)_{ms}$
- se utilizzo 8 bit…..
  -5 = $(10000101)_{ms}$, +5 = $(00000101)_{ms}$

Come si converte da Modulo e Segno a decimale? (MS ->10)

Si elimina il bit del segno e si converte il valore assoluto in notazione decimale. Il risultato sarà il valore assoluto se il bit di segno è 0, oppure il corrispondente numero negativo se il bit di segno è 1.

Esempi:

- se utilizzo 8 bit…..
  Voglio sapere a quale numero decimale corrisponde il numero $(10001011)_{ms}$.
  Il bit più significativo è 1 quindi il numero è negativo.
  Elimino il bit del segno mi rimangono 7 bit 0001011. Converto il numero in decimale…..il numero è 11. Quindi $(10001011)_{ms}$ = -11
- se utilizzo 5 bit…..
  Voglio sapere a quale numero decimale corrisponde il numero $(01111)_{ms}$.
  Il bit più significativo è 0 quindi il numero è positivo.
  Elimino il bit del segno mi rimangono 4 bit 1111. Converto in numero in decimale…..il numero è 15. Quindi $(01111)_{ms}$ = +15

Vediamo la rappresentazione dei numeri da -7 a +7 in MS:

Codice	Nat	MS
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7

Codice	Nat	MS
1000	8	-0
1001	9	-1
1010	10	-2
1011	11	-3
1100	12	-4
1101	13	-5
1110	14	-6
1111	15	-7

Attenzione…….. ci sono due zeri: $+0=0000_{ms}$ e $-0=1000_{ms}$

In modulo e segno, con n bit, possiamo rappresentare gli interi nell'intervallo da:

$$(-(2^{n-1}-1);\ 2^{n-1}-1)$$

Con n=4 bit i valori rappresentabili vanno da $-2^3+1=-7$ a $2^3-1=+7$
Con n=8 bit i valori rappresentabili vanno da $-2^7+1=-127$ a $2^7-1=+127$
Con n=16 bit i valori rappresentabili vanno da -32767 a +32767
Con n=32 bit i valori rappresentabili vanno da $-2X10^9$ a $+2X10^9$

Problemi:

- C'è una doppia rappresentazione per lo zero e si spreca una configurazione
- Le operazioni tra numeri rappresentati in MS richiedono algoritmi complessi altrimenti provocano degli errori.

  ES sommiamo i numeri:

-5 +	0 0 0 0 0 1 0 1 +		
+5=	1 0 0 0 0 1 0 1 =		
0	1 0 0 0 1 0 1 0	→	-10 e non 0!!!!!!

ESERCIZI

Esercizio 1:

Rappresenta i seguenti valori nella notazioni in modulo e segno utilizzando 8 bit.

a = 13_{10}

b = -3_{10}

c = -15_{10}

d = -20_{10}

e = -72_{10}

Esercizio 2;

A quali numeri decimali corrispondono i seguenti numeri binari rappresentati in MS con 4 bit?

a = 0111_{ms}

b = 1101_{ms}

c = 1010_{ms}

Esercizio 3:

Rappresentare in modulo e segno i seguenti numeri negativi su 10 bit:

- -31
- -109
- -321

Esercizio 4:

A quali numeri decimali corrispondono i seguenti numeri binari rappresentati in MS con 8 bit?

- $(10000110)_{ms}$
- $(10001110)_{ms}$
- $(10000011)_{ms}$
- $(00000101)_{ms}$

Ancora esercizi…

Convertire il numero -1310 in modulo e segno a 6 bit.

Convertire i numeri 0111_{MS} e 1111_{MS} in decimale.

Che numeri rappresentano 10100 e 01110 in MS?

Complemento a 2 (C2)

➔E' indispensabile indicare il numero **K** di bit utilizzati per la rappresentazione
➔Il bit più a sinistra rappresenta il segno del numero: 0 = '+' , 1 = '-'
➔La parte restante della rappresentazione NON è il valore assoluto del numero, lo è soltanto per i numeri positivi
➔C'è una sola rappresentazione dello 0. Non ci sono più configurazioni "sprecate":
Con 4 bit $0000_{c2} = 0_{10}$ mentre $1000_{c2} = -8_{10}$

Come si converte da Decimale a C2? (10 -> C2)

- Numeri interi positivi (compreso lo 0): un numero positivo è rappresentato in modo standard su k bit (come abbiamo visto per modulo e segno)

- Numeri interi negativi: si trova la rappresentazione di **-X** a partire da quella di **X**. Effettuare il complemento di ogni bit di X e aggiungere 1….
I tre passi da compiere:
1) rappresentare X in modo standard su k bit
2)complementare tutti i bit (1 ► 0, 0 ► 1)
3)sommare 1 al risultato

Esempio: dati 4 bit trovare la rappresentazione di -6 in C2
rappresentazione di $+6_{10}$= 0110
complemento tutti i bit = 1001
Aggiungo 1 a 1001 ottenendo 1010
Risultato $-6_{10} = 1010_{C2}$

Esempio: dati 4 bit trovare la rappresentazione di +6 in C2
rappresentazione di $+6_{10}$= 0110_{C2}

Esempio: dati 8 bit trovare la rappresentazione di -27 in C2
Rappresentazione di $+27_{10}$ = 00011011
Complemento tutti i bit = 11100100
Aggiungo 1 a 11100100 ottenendo 11100101
Risultato -27_{10}= 11100101_{c2}

Esempio: dati 8 bit trovare la rappresentazione di -1 e 1 in C2
Rappresentazione di $+1_{10}$ = 00000001
Complemento tutti i bit = 11111110
Aggiungo 1 a 11111110 ottenendo 11111111
Risultato -1_{10}= 11111111_{c2}

Vediamo la rappresentazione dei numeri da -7 a +7 in C2:

Codice	Nat	MS	C2
0000	0	0	0
0001	1	1	1
0010	2	2	2
0011	3	3	3
0100	4	4	4
0101	5	5	5
0110	6	6	6
0111	7	7	7

Codice	Nat	MS	C2
1000	8	-0	-8
1001	9	-1	-7
1010	10	-2	-6
1011	11	-3	-5
1100	12	-4	-4
1101	13	-5	-3
1110	14	-6	-2
1111	15	-7	-1

ATTENZIONE: valori opposti es. -4 e +4 hanno rappresentazioni completamente diverse!!!

Come si converte da C2 a Decimale? (C2->10)

- Numeri interi positivi (compreso lo 0): stesso procedimento della codifica modulo e segno
- Numeri interi negativi:
  Prendo in considerazione tutti i bit compreso anche il bit del segno.
  • si sottrae 1 al numero rappresentato in C2
  • si complementano tutti i bit (1 ► 0, 0 ► 1)
  • si converte da binario a decimale e si aggiunge il segno

Esempio:
A quale numero decimale corrisponde il numero 1100101_{C2} rappresentato con 7 bit?
E' un numero negativo perché il bit più significativo è uguale a uno. Prendo in considerazione tutti i bit compreso anche il bit del segno. Sottraggo 1 al numero ottenendo1100100. Complemento tutti i bit ottenendo 0011011=1+2+8+16=27. Quindi il numero è -27

Esempio:
A quale numero decimale corrisponde il numero 00001110_{C2} rappresentato con 8 bit?
E' un numero positivo perché il bit più significativo è uguale a zero. Procedo come per MS. 1110=2+4+8=14. Quindi il numero è +14

Esempio:
A quale numero decimale corrisponde il numero 1101100_{C2} rappresentato con 7 bit?
E' un numero negativo perché il bit più significativo è uguale a uno. Prendo in considerazione tutti i bit compreso anche il bit del segno. Sottraggo 1 al numero ottenendo 1101011. Complemento tutti i bit ottenendo 0010100=4+16=20. Quindi il numero è -20.

In complemento a 2, con n bit, possiamo rappresentare gli interi nell'intervallo da:

$$(-2^{n-1}; 2^{n-1}-1)$$

Con k=4 bit i valori rappresentabili vanno da **-8 a +7**
Con k=8 bit i valori rappresentabili vanno da **-128 a +127**
Con k=16 bit i valori rappresentabili vanno da **-32768 a + 32767**
…….

ESERCIZI

Esercizio 1:
Rappresenta i seguenti valori nella notazioni in complemento a 2 utilizzando 8 bit.
a = 1310
b = -310
c = -1510
d = -2010
e = -7210

Esercizio 2:
Si considerino le seguenti stringhe di 4 bit. Le si interpreti come valori interi espressi in C2.
a = 0111_{C2}
b = 1101_{C2}
c = 1010_{C2}

Esercizio 3:
Rappresentare in C2 i seguenti numeri negativi su 10 bit:

- -31
- -109
- -321

Esercizio 4:
Convertire in complemento a 2 i seguenti numeri:
$(12)_{10}$, $(-12)_{10}$, $(-8)_{10}$, $(1)_{10}$, $(-54)_{10}$

Esercizio 5:
Scrivere i seguenti numeri decimali in binario, ms e c2 (8 bit) : 15,-19,-128, 45, -232