Corso di Calcolatori Elettronici I
A.A. 2011-2012

Rappresentazione dei numeri interi in un calcolatore (parte 2)

Lezione 5

Prof. Roberto Canonico

Università degli Studi di Napoli Federico II
Facoltà di Ingegneria
Corso di Laurea in Ingegneria Informatica (allievi A-DE)
Corso di Laurea in Ingegneria dell'Automazione

Complementi diminuiti

- La rappresentazione in complementi diminuiti costituisce un'ulteriore alternativa per la codifica dei numeri relativi
- Concettualmente è analoga alla rappresentazione in complementi alla base
- La differenza rispetto ad essa è che la legge di codifica dei numeri negativi è leggermente differente:
  - $X=2^n - |x|$; (complementi alla base)
  - $X=2^n - 1 - |x|$; (complementi diminuiti)
- I numeri rappresentabili, se si utilizzano n bit, sono quelli compresi nell'intervallo $[-(2^{n-1} - 1), 2^{n-1} - 1]$.
- I numeri rappresentabili sono $2^n - 1$
- lo zero ha una doppia rappresentazione

Esempio

Rappresentazione in complementi diminuiti su 4 bit

n=4
$V = [-7,7] \cap Z$
Codifica:
Per $0 \le x \le 7$: $X = x$
per $-7 \le x \le -1$: $X = 2^n - 1 - |x|$

x	X_2	X_{10}
7	0111	7
6	0110	6
5	0101	5
4	0100	4
3	0011	3
2	0010	2
1	0001	1
0	0000;1111	0;15
-1	1110	14
-2	1101	13
-3	1100	12
-4	1011	11
-5	1010	10
-6	1001	9
-7	1000	8

Complementi diminuiti: perché?

- Maggiore semplicità con cui è possibile calcolare la rappresentazione dell'opposto di un numero, a partire dalla rappresentazione del numero stesso: basta semplicemente complementare tutti i bit della rappresentazione indistintamente
- Esempi:
  - la rappresentazione in complementi diminuiti su 4 bit di 4 è 0100;
    - complementando tutti i bit si ottiene 1011;
    - 1011 è la rappresentazione in complementi diminuiti su 4 bit di –4
  - la rappresentazione in complementi diminuiti su 4 bit di – 6 è 1001;
    - complementando tutti i bit si ottiene 0110;
    - 0110 è la rappresentazione in complementi diminuiti su 4 bit di 6

Aritmetica in complementi diminuiti

- Componenti:
  - Ancora l'addizionatore modulo-2^n (e non 2^n-1)
    - L'addizionatore modulo-2^n è più semplice da realizzare
  - Un complementatore
- Il risultato però deve essere opportunamente "corretto" per renderlo compatibile con l'aritmetica in modulo 2^{n-1}
- In particolare deve essere aggiunta un'unità al risultato nei seguenti casi:
  - se entrambi gli addendi sono negativi
  - se un addendo è positivo, l'altro negativo e la somma è positiva
- Nei casi suddetti l'aritmetica degli interi positivi (quella sulle rappresentazioni) da overflow
  - L'overflow (carry) quindi può essere interpretato come la necessità di effettuare la correzione

Esempi di somme in complementi diminuiti

```
- 2 +          1101 +
- 3 =          1100 =
-----   --->  -------
- 5           11001 +
                  1 =
              -------
               1010
```

Somma di due numeri negativi.
Si è generato overflow tra le rappresentazioni.
Necessita correzione.

```
  5 +          0101 +
- 2 =          1101 =
-----   --->  -------
  3           10010 +
                  1 =
             --------
               0011
```

Somma di un numero positivo e un numero negativo.
Il risultato è positivo.
Si è generato overflow tra le rappresentazioni.
Necessita correzione.

```
  3 +          0011 +
- 4 =   --->   1011 =
-----          ------
- 1            1110
```

Somma di un numero positivo e un numero negativo.
Il risultato è negativo.
Non si è generato overflow tra le rappresentazioni.
Non necessita alcuna correzione.

Rappresentazione eccesso-k

- La rappresentazione in eccesso-k costituisce un metodo diverso da quello dei resti in modulo per ricondurre i numeri negativi a positivi
- In particolare, tutti i numeri sono traslati "verso l'alto" di k, che viene scelto maggiore o uguale al numero più piccolo da rappresentare

$$X = x + k$$

Rappresentazione eccesso-k: proprietà

- Analogamente al caso dei complementi diminuiti, la somma va corretta aggiungendo o sottraendo la costante k, e quindi in maniera sufficientemente semplice
- Moltiplicazioni e divisioni risultano invece più complesse
- Il vantaggio di tale codifica è che viene conservata la proprietà della disuguaglianza sulle rappresentazioni:

$$x_1 > x_2 \Leftrightarrow X_1 > X_2$$

- Questa rappresentazione, perciò, è utilizzata soltanto laddove siano richieste fondamentalmente somme algebriche e confronti logici fra gli operandi
- Tipicamente si utilizza per rappresentare gli esponenti nella rappresentazione in virgola mobile (prossima lezione)

Esempio

Rappresentazione in eccesso-8 su 4 bit

n=4
$V = [-8,7] \cap Z$
Codifica:
$X = x + k$

x	X_2	X_{10}
7	1111	15
6	1110	14
5	1101	13
4	1100	12
3	1011	11
2	1010	10
1	1001	9
0	1000	8
-1	0111	7
-2	0110	6
-3	0101	5
-4	0100	4
-5	0011	3
-6	0010	2
-7	0001	1
-8	0000	0