Informatica B - Esercitazione 11

Riepilogo

Stefano Cereda
stefano1.cereda@mail.polimi.it

19/12/2017

Politecnico di Milano

Binario - TdE 26 Giugno 2017 i

Si dica e si motivi qual è il minimo numero di bit che permette di rappresentare in complemento a due (CP2) tutti i numeri seguenti:

1. A = + 255
2. B = - 42
3. C = + 7
4. D = - 257

Binario - TdE 26 Giugno 2017 ii

Se si volessero eseguire tutte le possibili somme e sottrazioni tra le coppie di numeri A, B, C, D, qual è il minimo numero di bit in cui dovrebbero essere rappresentati i numeri? Si motivi la risposta

Si calcoli il risultato dell'operazione 7 - 255 utilizzando la rappresentazione in CP2 con il numero di bit stabilito nel punto (a). Si mostrino tutti i passaggi eseguiti e si indichino i bit di carry e di overflow.

Binario - TdE 26 Giugno 2017 iii

Scrivere una funzione Matlab controllaCP2 che prende in ingresso una stringa (ad esempio '0101001') corrispondente ad un numero codificato in CP2 e restituisce due valori:

1. par che vale true se il numero è pari e false altrimenti
2. pos che vale true se il numero è positivo e false altrimenti

(L'esecuzione della funzione nel caso dell'esempio darebbe par = false e pos = true)

Binario - TdE 26 Giugno 2017 - Soluzione i

In CP2 con n bit posso rappresentare da -2^{n-1} a $2^{n-1} - 1$.

Dato un numero negativo x mi serviranno dunque $n = \log_2 -x + 1$ bit per rappresentarlo e dato un numero positivo y me ne serviranno $n = \log_2(y + 1) + 1$

Le potenze di 2 sono: 1 2 4 8 16 32 64 128 256 512 ...

I due numeri che devo controllare sono A e D:

- A=255 $\rightarrow$ $n_A = \log_2 256 + 1 = 9$
- D=-257 $\rightarrow$ $n_B = \log_2 257 + 1 = 10$

Dunque mi servono 10 bit per rappresentare tutti i valori.

Binario - TdE 26 Giugno 2017 - Soluzione ii

Le operazioni che possono dare problemi sono le sottrazioni che coinvolgono A e D. In particolare, A-D darà un numero positivo, mentre D-A darà un numero negativo. Le due operazioni avranno risultato uguale in valore assoluto. Dato che con n bit posso rappresentare da -2^{n-1} a $2^{n-1}-1$ controllo solo A-D che dando un risultato positivo è l'operazione più problematica a causa del -1.

$A - D = 255 - (-257) = 255 + 257 = 256 + 256 = 512 = 2^9$ e per rappresentarlo mi servono $n = \log_2(2^9 + 1) + 1 = 10 + 1 = 11$.

Mi servono dunque 11 bit per eseguire tutte le possibili somme e sottrazioni.

Binario - TdE 26 Giugno 2017 - Soluzione iii

Converto +7 in CPL2. Inizio a convertirlo in base 2:

7	1
3	1
1	1
0	

La rappresentazione è dunque 111. Essendo un numero positivo la notazione in CPL2 è 0111. Dovendo usare 10 bit aggiungo degli zeri a sinistra: 0000000111

Binario - TdE 26 Giugno 2017 - Soluzione iv

Inizio a convertire +255 in base 2:

255	1
127	1
63	1
31	1
15	1
7	1
3	1
1	1
0	

La rappresentazione di +255 è dunque 11111111. Aggiungo uno 0 a sinistra per avere la rappresentazione in CPL2: 011111111. Noi siamo interessati a -255, dunque copiamo da sx verso dx fino al primo 1 compreso, poi complementiamo: 100000001 e aggiungiamo degli 1 a sinistra per ottenere 10 bit: 1100000001

Binario - TdE 26 Giugno 2017 - Soluzione v

7	0000000111	+
-255	1100000001	=
riporti	111	
risultato	1100001000	

Operandi discordi non possono dare overflow.

Controllo il risultato: 1100001000 è negativo, l'opposto è 0011111000, che vale $8 + 16 + 32 + 64 + 128 = 248 \rightarrow -248 = -255 + 7$

Binario - TdE 26 Giugno 2017 - Soluzione vi

La funzione matlab deve semplicemente controllare il bit a sinistra per sapere se il numero è positivo, ed il bit a destra per sapere se è pari (ragionate sul perché un numero positivo pari mantiene il primo bit=0 quando passo al negativo).

```
1  function [par, pos] = controllaCP2(stringa)
2    if stringa(1) == '0'
3      pos = true;
4    else
5      pos = false;
6    end
7    % versione equivalente:
8    % pos = stringa(1) == '0';
9
10   par = stringa(end) == '0';
```

Binario - TdE 26 Giugno 2017 - Soluzione vii

```
11  end
```

Binario

Si considerino $A_{10} = +77$ e $B_{CPL2} = 101110$ (già in complemento a 2). Li si rappresenti entrambi in base 2 notazione complemento a 2, sul numero minimo di bit per rappresentare entrambi gli operandi. Si effettuino quindi, sul numero di bit prima individuato, le operazioni A+B e A–B in complemento a 2, indicando se si verifica overflow oppure no.

Binario - soluzione i

77	1
38	0
19	1
9	1
4	0
2	0
1	1
0	

$77_{10} = 1001101_2 = 01001101_{CPL2}$

Ho rappresentato A con 8 bit, devo aggiungere 2 bit a B. Essendo B negativo (inizia con 1) aggiungo degli 1: $B = 11101110_{CPL2}$

Noto anche che B vale -18, in quanto:
$-B = 010010_2 = 16 + 2 = 18_{10}$

Binario - soluzione ii

Eseguo A+B

A	01001101	+
B	11101110	=
risultato	(1)00111011	

Ignoro il riporto. Il risultato è 00111011. Operandi discordi non possono dare overflow.

$00111011 = 1 + 2 + 8 + 16 + 32 = 59 = 77 - 18$

Binario - soluzione iii

Eseguo A-B = A+(-B).

$-B = 00010010_{CPL2}$

A	01001101	+
-B	00010010	=
risultato	01011111	

Operandi concordi hanno dato un risultato ad essi concorde, quindi non si è verificato overflow.

$01011111 = 1 + 2 + 4 + 8 + 16 + 64 = 95 = 75 - (-18)$

C - Struct

Vedi esercizio 2 TdE 26 Giugno 2017

La difficoltà dell'esercizio è più nella comprensione del testo che nella sua risoluzione, abbiamo visto come affrontandolo un punto alla volta sia di facile risoluzione.

C -Struct e Array i

Un supermercato ha memorizzato il proprio archivio di scontrini nell'array *archivio*, in cui ogni è una struttura dati di tipo *scontrino* con le seguenti informazioni:

- IDcliente: stringa di 10 caratteri che identifica univocamente il cliente
- totale: totale della spesa in Euro
- punti: punti premio associati alla spesa

Ad ogni spesa, vengono assegnati ad ogni cliente un quantitativo di punti premio pari alla somma dei punti raccolti più un ulteriore punto premio per ogni 10 euro spesi.

C -Struct e Array ii

1. Definire il tipo *scontrino* e l'array *archivio*
2. Scrivere il frammento di codice C che, dato il codice identificativo di un cliente, calcoli la somma dei suoi punti premio.
3. Scrivere il frammento di codice C che ordina l'array *archivio* secondo il totale della spesa in ordine decrescente.

Nota: Si faccia attenzione al fatto che un cliente può comparire in più di uno scontrino.

Matlab - Ricorsione i

Un metodo per calcolare il valore approssimato della radice quadrata di un numero reale non negativo z, detto metodo babilonese, utilizza la seguente relazione ricorsiva:

$$x(n) = \frac{1}{2}(x(n-1) + \frac{z}{x(n-1)})$$

$$x(0) = 1$$

Dove n è un intero non-negativo arbitrario, mentre $x(n)$ rappresenta un'approssimazione della radice quadrata del numero z. Per esempio, se z è pari a 2 e n è pari a 3 il valore di $x(n)$ ottenuto applicando la formula indicata sopra è 1.4142 (che è una buona approssimazione della radice di 2).

L'errore di approssimazione associato a $x(n)$ è definito come segue:

$$errore(n) = |x(n) - x(n-1)|$$

e decresce al crescere di n. Per esempio, se si calcola la radice di 2 con n pari a 1 si ottiene il valore 1.5000 con errore 0.5000. Se invece si calcola la radice dello stesso valore con n pari a 2 si ottiene il valore 1.4167 con errore 0.0833.

Matlab - Ricorsione iii

1. Si scriva una funzione ricorsiva *sqrt1(z,n)* per MATLAB/Octave che restituisca un'approssimazione della radice quadrata di z e il relativo errore utilizzando il metodo babilonese. Quando n è pari a zero, si assuma un valore dell'errore pari a inf.
2. Si scriva una funzione *sqrt2(z,err)* per MATLAB/Octave che, utilizzando la funzione *sqrt1*, restituisca un valore approssimato della radice quadrata di z con errore non superiore al valore *err* fornito come parametro.

Nella soluzione di entrambi i quesiti non è permesso utilizzare alcuna funzione di libreria di MATLAB/Octave a parte la funzione *abs(x)* per il calcolo del valore assoluto di x.

Matlab - Grafici

Scrivere una funzione Octave con parametri a, b e p che esegue le seguenti operazioni:

1. calcola i valori $\cos(x^2)$ per valori di x compresi fra a e b, con passo p, e li memorizza in un vettore y
2. visualizza su un grafico l'andamento di y;
3. restituisce al chiamante il massimo valore di $\cos(x^2)$ per $a \leq x \leq b$ ed il valore di x corrispondente. Per effettuare il calcolo in questione si può utilizzare la funzione max che, quando prende come parametro un vettore, restituisce due valori di cui il primo è il valore massimo trovato nel vettore ed il secondo è l'indice della posizione del vettore in cui tale valore è contenuto.

Matlab - Scomposizione in fattori

Scrivere una funzione che scomponga in fattori primi un numero. Il risultato dovrà composto da due vettori: il primo conterrà le basi dei fattori, il secondo conterrà gli esponenti dei fattori, cioè quante volte essi sono ripetuti.

Esempio: il numero 100 è uguale a $2^2 * 5^2$, per cui il vettore delle basi sarà [2 5] mentre quello degli esponenti corrispondenti sarà [2 2].

Un modo semplice per fattorizzare un numero consiste nel dividerlo ripetutamente per tutti i numeri che gli sono inferiori fino ad ottenere 1.