CALCOLATORI ELETTRONICI A.A. 2019/2020
Esercizi sulla rappresentazione in floating point

1. Si consideri una rappresentazione binaria in virgola mobile a 16 bit, di cui (*nell'ordine da sinistra a destra*) 1 per il segno (1=negativo), 5 per l'esponente, che è rappresentato in eccesso 16, e 10 per la parte frazionaria della mantissa. In corrispondenza a tutti valori dell'esponente diversi da 00000 la mantissa è normalizzata tra 1 e 2 ($1 \leq m < 2$). Con l'esponente 00000 si rappresentano invece numeri denormalizzati, con esponente convenzionalmente uguale a -15 e mantissa compresa tra 0 e 1 ($0 < m < 1$):
    a) calcolare il massimo e il minimo numero positivo rappresentabili, sia normalizzati che denormalizzati, specificando anche i rispettivi numerali nella notazione suddetta;
    b) calcolare l'ordine di grandezza in termini di potenze di 10 della differenza fra il minimo positivo normalizzato e il massimo positivo denormalizzato;
    c) calcolare la potenza di 2 che approssima per eccesso il numero *n* rappresentato nella notazione suddetta dai 16 bit espressi in esadecimale da 80CF;
    d) rappresentare in complemento a due col numero minimo di bit il numero $n \cdot 2^{32}$ dove *n* è il numero di cui al punto precedente.

Soluzione

Notazione in virgola mobile a 16 bit:
1 bit per segno (1=negativo)
5 bit per esponente in eccesso 16
10 bit per mantissa

a) Calcoliamo il massimo e minimo numero positivo normalizzato rappresentabile.
Essendo e=5, in eccesso 2^{e-1}, l'intervallo dell'esponente è $[-2^4, 2^4-1] = [-16, 15]$, ma -16 non può essere usato in quanto corrisponde a 00000 (riservato per numeri denormalizzati). Quindi, non essendoci ulteriori configurazioni riservate:
Estremo superiore: $2^1 \cdot 2^{15} = 2^{16}$
Minimo numero positivo: $2^0 \cdot 2^{-15} = 2^{-15}$
Rappresentando i numerali nella notazione:
$2^{16} \approx$ 0 11111 1111111111 (per esponente: 15+16 = 31 = 11111)
2^{-15} = 0 00001 0000000000 (per esponente: -15+16 = 1 = 00001)

Calcoliamo il massimo e minimo numero positivo denormalizzato rappresentabile, ricordando che nella notazione data per i numeri denormalizzati l'esponente 00000 è convenzionalmente uguale a -15 e la mantissa compresa tra 0 e 1 ($0 < m < 1$):
estremo superiore: $2^0 \cdot 2^{-15} = 2^{-15}$
minimo numero positivo: $2^{-10} \cdot 2^{-15} = 2^{-25}$
Rappresentando i numerali nella notazione:
$2^{-15} \approx$ 0 00000 1111111111
2^{-25} = 0 00000 0000000001

b) Calcoliamo l'ordine di grandezza in termini di potenze di 10 della differenza fra il minimo positivo normalizzato e il massimo positivo denormalizzato; dal punto a) sappiamo che:
minimo positivo normalizzato: 2^{-15} = 0 00001 0000000000
estremo superiore positivo denormalizzato: $2^{-15} \approx$ 0 00000 1111111111
La loro differenza è:
$2^{-15} - 2^{-15}(2^{-1} + 2^{-2} + 2^{-3} + 2^{-4} + 2^{-5} + 2^{-6} + 2^{-7} + 2^{-8} + 2^{-9} + 2^{-10}) \approx 2^{-15} - 2^{-15}(2^0 - 2^{-11}) = 2^{-26}$
Calcoliamo l'ordine di grandezza di 2^{-26}:
$2^{-26} = 2^4 2^{-30} \approx 16 \cdot 10^{-9} \approx 10^{-8}$

c) Calcoliamo la potenza di 2 che approssima per eccesso il numero *n* rappresentato nella notazione suddetta dai 16 bit espressi in esadecimale da 80CF.
n = 80CF = 1000 0000 1100 1111 = 1 00000 0011001111
1 bit per segno: negativo (1)

5 bit per esponente in eccesso 16: -15 (è un numero denormalizzato)
10 bit per mantissa: $2^{-3} + 2^{-4} + 2^{-7} + 2^{-8} + 2^{-9} + 2^{-10}$

$n = -2^{-15}(2^{-3} + 2^{-4} + 2^{-7} + 2^{-8} + 2^{-9} + 2^{-10}) = -(2^{-18} + 2^{-19} + 2^{-22} + 2^{-23} + 2^{-24} + 2^{-25}) \approx -2^{-17}$

d) Rappresentiamo in complemento a due col numero minimo di bit il numero $n \cdot 2^{32}$ dove *n* è il numero di cui al punto c).

$n \cdot 2^{32} = -2^{32}(2^{-18} + 2^{-19} + 2^{-22} + 2^{-23} + 2^{-24} + 2^{-25}) = -(2^{14} + 2^{13} + 2^{10} + 2^{9} + 2^{8} + 2^{7})$

Occorrono 16 bit per la rappresentazione in CP2

$n \cdot 2^{32} = -(0110011110000000) = 1001100010000000$

2. Si consideri una rappresentazione binaria in virgola mobile a 24 bit, di cui (*nell'ordine da sinistra a destra*) 1 per il segno (0=positivo), 7 per l'esponente, che è rappresentato in eccesso 64, e 16 per la parte frazionaria della mantissa. In corrispondenza a tutti valori dell'esponente diversi da 0000000 la mantissa è normalizzata tra 1 e 2 ($1 \le m < 2$). Con l'esponente 0000000 si rappresentano invece numeri denormalizzati, con esponente uguale a -63 e mantissa normalizzata tra 0 e 1 ($0 < m < 1$).
    a) calcolare il massimo e il minimo numero positivo rappresentabile (normalizzato e denormalizzato), specificando anche i rispettivi numerali nella notazione suddetta;
    b) calcolare l'ordine di grandezza in termini di potenze di 10 dei numeri calcolati al punto a);
    c) calcolare il valore arrotondato alla più vicina potenza di 2 del numero rappresentato nella notazione suddetta dai 24 bit espressi in esadecimale da 6503F3.

Soluzione

Notazione in virgola mobile a 24 bit:
1 bit per segno (0=positivo)
7 bit per esponente in eccesso 64
16 bit per mantissa

a) Calcoliamo il massimo e il minimo numero positivo rappresentabile (normalizzato e denormalizzato), specificando anche i rispettivi numerali nella notazione data.

Numeri normalizzati:
Essendo e=7 in eccesso 2^{e-1}, l'intervallo dell'esponente è $[-2^6, 2^6-1] = [-64, 63]$ ma -64 non può essere usato in quanto corrisponde a 0000000 (riservato per numeri denormalizzati). Quindi, non essendoci altre configurazioni riservate:
estremo superiore: $2^1 \cdot 2^{63} = 2^{64}$
minimo numero positivo: $2^0 \cdot 2^{-63} = 2^{-63}$
Rappresentando i numerali nella notazione:
$2^{64} \approx$ 0 1111111 1111111111111111 (per esponente: 63+64 = 127 = 1111111)
2^{-63} = 0 0000001 0000000000000000 (per esponente: -63+64 = 1 = 0000001)

Numeri denormalizzati:
Ricordando che nella notazione data per i numeri denormalizzati l'esponente 0000000 è convenzionalmente uguale a -63 e la mantissa compresa tra 0 e 1 ($0 < m < 1$):
estremo superiore: $2^0 \cdot 2^{-63} = 2^{-63}$
minimo numero positivo: $2^{-63} \cdot 2^{-16} = 2^{-79}$
Rappresentando i numerali nella notazione:
$2^{-63} \approx$ 0 0000000 1111111111111111
2^{-79} = 0 0000000 0000000000000001

b) Calcoliamo l'ordine di grandezza in termini di potenze di 10 dei numeri calcolati al punto a):
$2^{64} = 2^4\, 2^{60} = 2^4\, (2^{10})^6 \approx 2^4\, (10^3)^6 = 16 \cdot 10^{18} \approx 10^{19}$
$2^{-63} = 2^{-3}\, 2^{-60} = 2^{-3}\, (2^{10})^{-6} \approx 2^{-3}\, (10^3)^{-6} = 2^{-3} \cdot 10^{-18} \approx 10^{-19}$
$2^{-79} = 2^1\, 2^{-80} = 2^1\, (2^{10})^{-8} \approx 2^1\, (10^3)^{-8} = 2 \cdot 10^{-24} \approx 10^{-24}$

c) Calcoliamo il valore arrotondato alla più vicina potenza di 2 del numero rappresentato nella notazione suddetta dai 24 bit espressi in esadecimale da 6503F3:
6503F3 = 0110 0101 0000 0011 1111 0011 = 0 1100101 0000001111110011
1 bit per segno: positivo (0)
7 bit per esponente in eccesso 64: $2^6 + 2^5 + 2^2 + 2^0 - 2^6 = 2^5 + 2^2 + 2^0 = 37$ (quindi numero normalizzato)
16 bit per mantissa: $2^0 + 2^{-7} + 2^{-8} + 2^{-9} + 2^{-10} + 2^{-11} + 2^{-12} + 2^{-15} + 2^{-16}$ (2^0 perché numero normalizzato)
Valore arrotondato alla più vicina potenza di 2: $(6503F3) \approx 2^{37}$

3. Si considerino due notazioni binarie in virgola mobile a 16 bit, entrambe con (*nell'ordine da sinistra a destra*) 1 bit per il segno (0=positivo), *e* bit per l'esponente, rappresentato in eccesso 2^{e-1}, ed i rimanenti *m* bit per la parte frazionaria della mantissa che è normalizzata tra 1 e 2. Nella prima notazione *e*=4 ed *m*=11, nella seconda *e*=8 ed *m*=7.
    a) dato il numero *n* rappresentato nella prima notazione dalla stringha 35D7, rappresentarlo nella seconda notazione;
    b) calcolare l'errore relativo ed assoluto che si commette nel passaggio di notazione;
    c) dato il numero *n* rappresentato in complemento a 2 dalla stringa B3F742, definire una terza notazione, analoga alle precedenti ma con valore di *e* tale da rappresentare *n* col minimo errore relativo;
    d) calcolare l'ordine di grandezza decimale dell'errore relativo di cui al punto c).

Soluzione

Notazione in virgola mobile a 16 bit:
1 bit per segno (0=positivo)
e bit per esponente in eccesso 2^{e-1}
m=(15-*e*) bit per mantissa
Notazione A: e=4, m=11
Notazione B: e=8, m=7

a) Dato il numero *n* rappresentato nella notazione A dalla stringa 35D7, rappresentarlo nella notazione B.
n = 35D7 = 0011 0101 1101 0111
Nella notazione A: 0 0110 10111010111
1 bit per segno: positivo (0)
4 bit per esponente in eccesso 2^3: 0110 = 6 - 2^3 = -2
11 bit per mantissa = 10111010111 = $2^0 + 2^{-1} + 2^{-3} + 2^{-4} + 2^{-5} + 2^{-7} + 2^{-9} + 2^{-10} + 2^{-11}$ (2^0 perché mantissa normalizzata tra 1 e 2)
35D7 = 2^{-2} ($2^0 + 2^{-1} + 2^{-3} + 2^{-4} + 2^{-5} + 2^{-7} + 2^{-9} + 2^{-10} + 2^{-11}$)
Nella notazione B:
1 bit per segno: positivo (0)
8 bit per esponente in eccesso 2^7: -2 + 2^7 = 126 = 01111110
7 bit per mantissa: $2^0 + 2^{-1} + 2^{-3} + 2^{-4} + 2^{-5} + 2^{-7}$ = 1011101
Quindi:
n = 0 01111110 1011101 = 0011 1111 0101 1101 = 3F5D

b) Calcoliamo l'errore relativo ed assoluto che si commette nel passaggio di notazione.
Errore assoluto e_A
$e_A = 2^{-2} (2^0 + 2^{-1} + 2^{-3} + 2^{-4} + 2^{-5} + 2^{-7} + 2^{-9} + 2^{-10} + 2^{-11}) - 2^{-2} (2^0 + 2^{-1} + 2^{-3} + 2^{-4} + 2^{-5} + 2^{-7}) = 2^{-2} (2^{-9} + 2^{-10} + 2^{-11}) = 2^{-11} + 2^{-12} + 2^{-13}$
Errore relativo e_R
$e_R = 2^{-2} (2^{-9} + 2^{-10} + 2^{-11}) / (2^{-2} (2^0 + 2^{-1} + 2^{-3} + 2^{-4} + 2^{-5} + 2^{-7} + 2^{-9} + 2^{-10} + 2^{-11})) \approx 2^{-10}$

c) Dato il numero *n* rappresentato in complemento a 2 dalla stringa B3F742, definiamo una terza notazione, analoga alle precedenti ma con valore di *e* tale da rappresentare *n* col minimo errore relativo.
B3F742 è rappresentato in CP2:
B3F742 = 1011 0011 1111 0111 0100 0010 = - (0100 1100 0000 1000 1011 1110) = - ($2^{22} + 2^{19} + 2^{18} + 2^{11} + 2^7 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1$) = $-2^{22} (2^0 + 2^{-3} + 2^{-4} + 2^{-11} + 2^{-15} + 2^{-17} + 2^{-18} + 2^{-19} + 2^{-20} + 2^{-21})$
Determiniamo il minimo numero di bit necessari per rappresentare l'esponente:
$16 < 22 < 32 = 2^5$ da cui e=6
Quindi nella notazione C: e=6, m=9

d) Calcoliamo l'ordine di grandezza decimale dell'errore relativo di cui al punto c).
Massimo errore relativo nella notazione C:
$2^{-9} = 2^1 2^{-10} \approx 2 \cdot 10^{-3} \approx 10^{-3}$

4. Si consideri una rappresentazione binaria in virgola mobile a 16 bit, di cui (*nell'ordine da sinistra a destra*) 1 bit per il segno (0=positivo), *e* per l'esponente, che è rappresentato in complemento a 2, ed i rimanenti per la parte frazionaria della mantissa che è normalizzata tra 1 e 2 ($1 \leq m < 2$):
    a) determinare il valore minimo di *e* che consenta di rappresentare nella notazione data il numero *n* rappresentato in notazione eccesso 2^{31} dalla stringa esadecimale 83FA534B;
    b) determinare l'errore di troncamento assoluto e relativo che si commette rappresentando *n* nella notazione in virgola mobile suddetta, esprimendolo in termini della più vicina potenza di 2;
    c) determinare il numerale corrispondente, nella notazione data, al più piccolo numero positivo rappresentabile.

Soluzione

Notazione in virgola mobile a 16 bit:
1 bit per segno (0=positivo)
e bit per esponente in CP2
m=(15-*e*) bit per mantissa, normalizzata

a) 83FA534B è rappresentato in eccesso 2^{31}:
83FA534B = 1000 0011 1111 1010 0101 0011 0100 1011 = $2^{31} + 2^{25} + 2^{24} + 2^{23} + 2^{22} + 2^{21} + 2^{20} + 2^{19} + 2^{17} + 2^{14} + 2^{12} + 2^{9} + 2^{8} + 2^{6} + 2^{3} + 2^{1} + 2^{0} - 2^{31} = 2^{25} (2^{0} + 2^{-1} + 2^{-2} + 2^{-3} + 2^{-4} + 2^{-5} + 2^{-6} + 2^{-8} + 2^{-11} + 2^{-13} + 2^{-16} + 2^{-17} + 2^{-19} + 2^{-22} + 2^{-24} + 2^{-25})$
Determiniamo il minimo numero di bit necessari per rappresentare l'esponente:
$16 < 25 < 32 = 2^5$ da cui e=6
Quindi nella notazione: e=6, m=9

b) Poiché nella notazione data i bit per la mantissa sono 9, si ha:
Errore assoluto e_A:
$e_A = 2^{25} (2^{-11} + 2^{-13} + 2^{-16} + 2^{-17} + 2^{-19} + 2^{-22} + 2^{-24} + 2^{-25}) = 2^{14} (2^{0} + 2^{-2} + 2^{-5} + 2^{-6} + 2^{-8} + 2^{-11} + 2^{-13} + 2^{-14}) \approx 2^{14}$
Errore relativo e_R:
$e_R = 2^{25} (2^{-11} + 2^{-13} + 2^{-16} + 2^{-17} + 2^{-19} + 2^{-22} + 2^{-24} + 2^{-25}) / (2^{25} (2^{0} + 2^{-1} + 2^{-2} + 2^{-3} + 2^{-4} + 2^{-5} + 2^{-6} + 2^{-8} + 2^{-11} + 2^{-13} + 2^{-16} + 2^{-17} + 2^{-19} + 2^{-22} + 2^{-24} + 2^{-25})) \approx 2^{-12}$

c) Essendo *e*=6 e l'esponente rappresentato in CP2, l'intervallo di rappresentazione dell'esponente è $[-2^5, 2^5-1] = [-32, 31]$ e -32 può essere usato in quanto non ci sono configurazioni riservate nella notazione data. Quindi:
minimo numero positivo: $2^{0} \cdot 2^{-32} = 2^{-32}$
Rappresentando il numerale nella notazione:
2^{-32} = 0 000000 000000000
essendo 000000 la rappresentazione di -32 in CP2 con 6 bit.

5. Si consideri una rappresentazione binaria in virgola mobile a 24 bit, di cui (*nell'ordine da sinistra a destra*) 1 bit per il segno (0=positivo), 7 per l'esponente, che è rappresentato in eccesso 64, ed i rimanenti per la parte frazionaria della mantissa che è normalizzata tra 1 e 2 ($1 \leq m < 2$):
    a) rappresentare nella notazione suddetta la somma dei numeri corrispondenti in tale notazione alle stringhe esadecimali 33834B e 443545;
    b) determinare gli errori di troncamento assoluto e relativo che si commettono rappresentando il risultato della somma in questione, esprimendoli in termini della più vicina potenza di 2;
    c) determinare il numerale corrispondente, nella notazione data, al più piccolo numero positivo rappresentabile, indicandone l'ordine di grandezza come potenza di 10.

6. Si considerino due notazione binarie in virgola mobile a 16 bit, entrambe con (*nell'ordine da sinistra a destra*) 1 bit per il segno (0=positivo), e bit per l'esponente, rappresentato in complemento a 2, ed i rimanenti m bit per la parte frazionaria della mantissa che è normalizzata tra 1 e 2. Nella prima notazione e=4 ed m=11, nella seconda e=11 ed m=4.
    a) dati i numeri r ed s rappresentati in complemento a 2 rispettivamente dalle stringhe esadecimali F03 e 653, scegliere per ciascuno la notazione più adatta a rappresentarlo fra le due proposte e calcolare i rispettivi numerali;
    b) specificare se si commette o meno un errore nell'utilizzare per r ed s le rappresentazioni scelte al punto a), e, in caso positivo, valutare l'errore relativo ed assoluto.

7. Si consideri una rappresentazione binaria in virgola mobile a 24 bit, di cui (nell'ordine da sinistra a destra) 1 per il segno (0=positivo), e=9 per l'esponente, che è rappresentato in eccesso 2^{e-1}, ed i rimanenti per la parte frazionaria della mantissa m che è normalizzata tra 1 e 2 ($1 \leq m < 2$).
    a) Calcolare il minimo ed il massimo numero negativo rappresentabili nella notazione data, precisando anche i numerali che li rappresentano ed il loro ordine di grandezza decimale.
    b) Dato il numero r rappresentato in tale notazione dalla stringa A5D424, rappresentare in complemento a 2 con 16 bit il numero intero n che approssima r per difetto.