UVOD U ORGANIZACIJU I ARHITEKTURU RAČUNARA

Zapis brojeva u pokretnom zarezu, standard IEEE754

Danijela Simić

ULP, Relativna greška i Zaokruživanje

Zapis realnih brojeva u računaru je aproksimacija skupa realnih brojeva u određenom intervalu. Pri aproksimiranju na određeni broj decimala može doći do zaokruživanja, a samim tim i greške. Greška pri zaokruživanju se meri na dva načina:
1. Pomocu ulp-a
2. Pomocu relativne greške

ULP (Unit in the last place) je najmanja vrednost za koju se mogu razlikovati dva broja u pokretnom zarezu zapisana u određenoj osnovi i sa određenom preciznošću. Ako je broj z predstavljen u računaru kao $d_0, d_{-1} \dots d_{-(p-1)} \cdot b^e$, onda je greška

$$\|d_0, d_{-1} \dots d_{-(p-1)} - \frac{z}{b^e}\| \cdot b^{p-1} ulp - a \tag{1}$$

Primer :
b=10, p=4. Broj z=0.034869 je predstavljen kao $3.487 \cdot 10^{-2}$. Greška je:
$\|3.487$-$(0.034869/10^{-2})\| \cdot 10^3 =$
$\|3.487$-$3.4869\| \cdot 10^3 =$
$0.0001 \cdot 10^3 = 0.1$ ulp-a

Relativna greška je apsolutna vrednost razlike realnog broja i njegove reprezentacije podeljena apsolutnom vrednošću realnog broja. Uvek se zapisuje u terminima mašinskog $\epsilon = \frac{b^{1-p}}{2}$.

$$relativna\_greska = \frac{\|realan\_broj - reprezentacija\_realnog_b roja\|}{\|realan\_broj\|} = \epsilon \cdot \alpha \tag{2}$$

Primer :
b=10, p=4, e=0.0005. Broj z=0.034869 je predstavljen kao $3.487 \cdot 10_{-2}$. Relativna greška je:

$\|0.034869$-$0.03487\|/\|0.034869\| =$
0.000028678 $\approx$ 0.0574ϵ

Zaokruživanje se vrši kada rezultat operacije ne može biti tačno zapisan. Moguće su sledeće vrste zaokruživanja:

- **Zaokruživanje na najbližu vrednost.** Pri ovoj vrsti zaokruživanja broj se zaokružuje na najbližu predstavljivu vrednost, uz zaokruživanje na parnu cifru kada je broj na sredini intervala između dve predstavljive vrednosti. Ovo je predefinisani način zaokruživanja.
- **Zaokruživanje prema** $+\infty$. Realizuje se u dva koraka:
  (a) Ako je broj pozitivan i postoji bar jedna jedinica na nekoj poziciji desno od poslednje pozicije koja se čuva u zapisu, na poslednju poziciju se dodaje jedinica.

(b) Bez obzira na znak odbacuju se bitovi desno od poslednje pozicije koja se cuva u zapisu.

- **Zaokru zivanje prema** $-\infty$**.** Realizuje se u dva koraka:

  (a) Ako je broj negativan i postoji bar jedna jedinica na nekoj poziciji desno od poslednje pozicije koja se čuva u zapisu, na poslednju poziciju se dodaje jedinica.

  (b) Bez obzira na znak odbacuju se bitovi desno od poslednje pozicije koja se čuva u zapisu.

- **Zaokruživanje prema nuli.** Odbacuju se svi bitovi desno od poslednje pozicije koja se čuva u zapisu.

Zapis brojeva u pokretnom zarezu, standard IEEE754

Broj u pokretnom zarezu **jednostruke tačnosti** u standardu IEEE 754 propisuje zapis:
Broj je zapisan u 32 bita i to:

1. 1 bit za znak
2. 8 bita za eksponent (u zapisu sa uvecanjem 127)
3. 23 značajne binarne cifre

Broj zapisujemo u sistemu sa osnovm 2.

Normalizovani brojevi u jednostrukoj tačnosti imaju eksponent između -126 (00000001) i +127 (11111110). Frakcija ima oblik $1, d_1 ... d_{(p-1)}$, pri čemu se prva jedinica ne zapisuje. Zapis frakcije može da ima bilo koju vrednost.

NaN (Not a number) su specijalne vrednosti koje nisu brojevi i označavaju neke izuzetne situacije prilikom izračunavanja. Npr. 0/0 ili $\sqrt{-1}$. Eksponent NaN vrednosti je maksimalan. U slucaju jednostruke tačnosti to je 128 (11111111). Frakcija mora biti razlicita od nule. Postoje takozvani **Signalni NaN** (SNaN) i **Tihi NaN** (QNaN).

Signalni NaN signalizira izuzeto stanje kod aritmetičkih operacija, poređenja i konverzija. Zapis eksponenta u jednostrukoj tačnosti je 11111111. Prvi bit frakcije je 0. Ostatak frakcije je različit od 0.

Tihi NaN predstavlja pojavu nedozvoljene operacije u programu. Propagira se kroz izračunavanje, tako da ostaje vidljiv na njegovom kraju. Signalizira se pojava izuzeća. Zapis eksponenta u jednostrukoj tačnosti je 11111111. Prvi bit frakcije je 1. Ostali bitovi frakcije su proizvoljni.

IEEE 754 standard omogućava predstavljanje beskonačnih vrednosti. Time se omogucava nastavak izračunavanja u slučaju pojave prekoračenja. Znak odreduje da li se radi o $+\infty$ ili $-\infty$. Eksponent u jednostrukoj tačnosti ima vrednost 128 (11111111). Frakcija je 0.

Da bi se povećala gustina realnih brojeva oko nule i izbegla pojava potkoračenja uvode se takozvani denormalizovani brojevi. U jednostrukoj tačnosti važi:
Zapis eksponenta u jednostrukoj tačnosti je 00000000.
Frakcija je razlicita od nule, a vodeća jedinica se ne podrazumeva. Ako je frakcija f, onda je predstavljeni broj $0.f \cdot 2^{-126}$.

Primeri, standard IEEE754

1. Zapisati broj 13,25 prema standardu IEEE 754 u jednostrukoj tačnosti.

   Rešenje :
   $(13,25)_{10} = (1101.01)_2 = (1.10101)_2 \cdot 2^3$
   Znak: 0 (+)
   Eksponent sa uvećanjem 127: $130 = (10000010)_2$
   Značajni deo sa implicitnom jedinicom: 10101 - $(1.10101)_2$
   Zapis:
   0 10000010 10101000000000000000000

2. Zapisati broj -222.625 prema standardu IEEE 754 u jednostrukoj tačnosti.

   Rešenje :
   $(222.625)_{10} = (11011110.101)_2 = (1.1011110101)_2 \cdot 2^7$
   Znak: 1 (-)
   Eksponent sa uvećanjem 127: $134 = (10000110)_2$
   Zapis:
   1 10000110 10111101010000000000000

3. Zapisati broj 6.625 prema standardu IEEE 754 u jednostrukoj tačnosti.

   Rešenje :
   $(6.625)_{10} = (110.101)_2 = (1.10101)_2 \cdot 2^2$
   Znak: 1 (-)
   Eksponent sa uvećanjem 127: $129 = (10000001)_2$
   Zapis:
   0 10000001 10101000000000000000000

4. Zapisati broj -11.125 prema standardu IEEE 754 u jednostrukoj tačnosti.

   Rešenje :
   $(11.125)_{10} = (1011.001)_2 = (1.011001)_2 \cdot 2^3$
   Znak: 1 (-)
   Eksponent sa uvećanjem 127: $130 = (10000010)_2$
   Zapis:
   1 10000010 01100100000000000000000

5. Pročitati sledeći zapis:
   1 10000110 01001000000000000000000

   Rešenje :
   Znak: -
   Eksponent: $(10000110)_2 = 134$, 134-127=7
   Frakcija: $(1.01001)_2 = (1.28125)_{10}$
   Rešenje: $(-1.28125)_{10} \cdot 2^7 = (-164)_{10}$

6. Pročitati sledeći zapis:
   11000011010010000000000000000000

Rešenje :
1 10000110 10010000000000000000000
Znak: -
Eksponent: $(10000110)_2 = 134$, 134-127=7
Frakcija: $(1.1001)_2$
Rešenje: $(-1.1001)_2 \cdot 2^7 = (-11001000)_2 = (-200)_{10}$

7. Pročitati sledeći zapis:
0 00000000 00000000000000000000101

Rešenje :
Znak: +
U pitanju je denormalizovani broj.
Rešenje: $(0.00000000000000000000101)_2 \cdot 2^{-126} = (0.101) \cdot 2^{-146}$

8. Pročitati sledeći zapis:
0 00000000 00000000000000000000111

Rešenje :
Znak: +
U pitanju je denormalizovani broj.
Rešenje: $(0.00000000000000000000111)_2 \cdot 2^{-126} = (0.111) \cdot 2^{-146}$

9. Pročitati sledeći zapis:
1 10000011 10001000000000000000000

Rešenje :
Znak: -
Eksponent: $(10000011)_2 = 131$, 131-127=4
Frakcija: $(1.1001)_2$
Rešenje: $(-1.1001)_2 \cdot 2^4 = (-11001)_2 = (-25)_{10}$

10. Pročitati sledeći zapis:
11000001011011000000000000000000

Rešenje :
Znak: -
Eksponent: $(10000010)_2 = 130$, 130-127=3
Frakcija: $(1.11011)_2$
Rešenje: $(-1.11011)_2 \cdot 2^3 = (-1110.11)_2 = (-14.75)_{10}$

Sabiranje i oduzimanje u IEEE 754

Prilikom sabiranja i oduzimanja operandi se svode na jednake eksponente. Manji eksponent se povećava, a cifre frakcije koja mu odgovara se pomeraju udesno za onoliko mesta za koliko je povećan eksponent. Ako pri pomeranju frakcija postane 0 rezultat je vrendost drugog operanda.

Sabiranje i oduzimanje frakcija se vrši prema pravilima koja važe za cele brojeve u zapisu znak i apsolutna vrednost. Eksponent rezultata je eksponent operanada posle izjednačavanja. Ako dolazi do prekoračenja rezultat se pomera za jedno mesto udesno uz povećanje vrednosti eksponenta za jedan. Ako povećanje vrednosti eksponenta dovede do prekoračenja rezultat je ∞, ali uzevši u obzir i znak.

Ako rezultat operacije nije normalizovan, pokušava se normalizacija. Cifre frakcije se pomeraju ulevo uz smanjivanje eksponenta. Može se dobiti i denormalizovan rezultat. Na kraju se vrši zaokruživanje, ako je potrebno.

Primeri :

1. Predstaviti brojeve 5.375 i 0.5625 u IEEE 754 zapisu sa binarnom osnovom, i izračunati njihov zbir po algoritmu za sabiranje brojeva zapisanih u IEEE754 zapisu. Rezultat prevesti u dekadni sistem.

   Rešenje :
   $(5.375)_{10} = (101.011)_2 = (1.01011) \cdot 2^2$
   znak: 0 (+)
   eksponent: $2 + 127 = 129 = (10000001)_2$
   0 10000001 01011000000000000000000

   $(0.5625)_{10} = (0.1001)_2 = (1.001) \cdot 2^{-1}$
   znak: 0 (+)
   eksponent: $-1 + 127 = 126 = (01111110)_2$
   0 01111110 00100000000000000000000

   Sada sabiramo:

   (a) Rezultat je pozitivan broj, cifra na mestu za znak je 0

(b) Operandi se svode na jednake eksponente. $10000001 > 01111110$, pa se druga frakcija pomera za tri mesta udesno, pa dobijamo (0.001001). Sada je moguće sabrati frakcije.
1.01011 + 0.001001 = 1.011111
Rezultat je već normalizovan, tako da nema potrebe za normalizacijom. Rešenje:
0 10000001 01111100000000000000000

(c) Odgovarajuća dekadna vrednost je $+(1.011111) \cdot 2^2 = (101.1111)_2 = 5.9375$

2. Predstaviti brojeve 5.375 i 0.5625 u IEEE 754 zapisu sa binarnom osnovom, i izračunati njihovu razliku po algoritmu za oduzimanje brojeva zapisanih u IEEE754 zapisu. Rezultat prevesti u dekadni sistem.

Rešenje :
$(5.375)_{10} = (101.011)_2 = (1.01011) \cdot 2^2$
znak: 0 (+)
eksponent: $2 + 127 = 129 = (10000001)_2$
0 10000001 01011000000000000000000

$(0.5625)_{10} = (0.1001)_2 = (1.001) \cdot 2^{-1}$
znak: 0 (+)
eksponent: $-1 + 127 = 126 = (01111110)_2$
0 01111110 00100000000000000000000

Sada oduzimamo:

(a) Rezultat je pozitivan broj, cifra na mestu za znak je 0

(b) Operandi se svode na jednake eksponente. $10000001 > 01111110$, pa se druga frakcija pomera za tri mesta udesno, pa dobijamo (0.001001). Sada je moguće oduzeti frakcije.
1.01011 - 0.001001 = 1.001101
Rezultat je već normalizovan, tako da nema potrebe za normalizacijom. Rešenje:
0 10000001 00110100000000000000000

(c) Odgovarajuća dekadna vrednost je $+(1.001101) \cdot 2^2 = (100.1101)_2 = 4.8125$

3. Predstaviti brojeve 5.625 i 11.125 u IEEE 754 zapisu sa binarnom osnovom, i izračunati njihov zbir po algoritmu za sabiranje brojeva zapisanih u IEEE754 zapisu. Rezultat prevesti u dekadni sistem.

Rešenje :
$(5.625)_{10} = (101.101)_2 = (1.01101) \cdot 2^2$
znak: 0 (+)
eksponent: $2 + 127 = 129 = (10000001)_2$
0 10000001 01101000000000000000000

$(11.125)_{10} = (1011.001)_2 = (1.011001) \cdot 2^3$
znak: 0 (+)
eksponent: $3 + 127 = 130 = (10000010)_2$
0 10000010 01100100000000000000000

Sada sabiramo:

(a) Rezultat je pozitivan broj, cifra na mestu za znak je 0

(b) Operandi se svode na jednake eksponente. $10000001 < 10000010$, pa se prva frakcija pomera za jedno mesta udesno, pa dobijamo (0.101101). Sada je moguće sabrati frakcije.
0.101101 + 1.011001 = 10.000110
Rezultat je nije normalizovan, eksponent treba uvećati za jedan, tj. dobijamo 10000011 pa je frakcija onda 1.0000110. Rešenje:
0 10000011 00001100000000000000000

(c) Odgovarajuća dekadna vrednost je $+(1.0000110) \cdot 2^4 = (10000.110)_2 = 16.75$

4. Predstaviti brojeve -222.625 i -111.125 u IEEE 754 zapisu sa binarnom osnovom, i izracunati njihovu razliku po algoritmu za oduzimanje brojeva zapisanih u IEEE754 zapisu. Rezultat prevesti u dekadni sistem.

Rešenje :
$(222.625)_{10} = (11011110.101)_2 = (1.1011110101) \cdot 2^7$
znak: 1 (-)
eksponent: $7 + 127 = 134 = (10000110)_2$
1 10000110 10111101010000000000000

$(111.125)_{10} = (1101111.001)_2 = (1.101111001) \cdot 2^6$
znak: 1 (-)
eksponent: $6 + 127 = 133 = (10000101)_2$
0 10000101 10111100100000000000000

Sada oduzimamo:

(a) -222.625 - (-111.125) = -222.625 + 111.125 = -(222.625 - 111.125)

(b) Rezultat je negativan broj, cifra na mestu za znak je 1.

(c) Operandi se svode na jednake eksponente. $10000101 < 10000110$, pa se druga frakcija pomera za jedno mesta udesno, pa dobijamo (0.1101111001). Sada je moguće oduzeti frakcije.
1.1011110101 - 0.1101111001 = 0.1101111100
Rezultat je nije normalizovan, eksponent treba smanjiti za jedan, tj. dobijamo 10000101 pa je frakcija onda 1.1011111. Rešenje:
1 10000101 10111110000000000000000

(d) Odgovarajuća dekadna vrednost je $-(1.1011111) \cdot 2^6 = -(1101111.1)_2 = -111.5$

5. Predstaviti brojeve -27.375 i 58.75 u IEEE754 zapisu sa binarnom osnovom, sabrati dobijene zapise po algoritmu za sabiranje brojeva zapisanih u IEEE754 zapisu i rezultat prevesti u dekadni sistem.

   Rešenje :
   $(27.375)_{10} = (11011.011)_2 = (1.1011011) \cdot 2^4$
   znak: 1 (-)
   eksponent: $4 + 127 = 131 = (10000011)_2$
   1 10000011 10110110000000000000000

   $(58.75)_{10} = (111010.11)_2 = (1.1101011) \cdot 2^5$
   znak: 0 (+)
   eksponent: $5 + 127 = 132 = (10000100)_2$
   0 10000100 11010110000000000000000

   Sada sabiramo:

   (a) -27.375 + 58.75 = 58.75 - 27.375
   (b) Rezultat je pozitivan broj, cifra na mestu za znak je 0.
   (c) Operandi se svode na jednake eksponente. $10000011 < 10000100$, pa se prva frakcija pomera za jedno mesta udesno, pa dobijamo (0.11011011). Sada je moguće oduzeti frakcije.
   1.1101011 - 0.11011011 = 0.11111011
   Rezultat je nije normalizovan, eksponent treba smanjiti za jedan, tj. dobijamo 10000011 pa je frakcija onda 1.1111011. Rešenje:
   0 10000011 11110110000000000000000
   (d) Odgovarajuća dekadna vrednost je $+(1.1111011) \cdot 2^4 = (11111.011)_2 = 31.375$

6. Izvršiti računske operacije nad brojevima predstavljenim u IEEE754 zapisu sa binarnom osnovom i obavezno prevesti rezultat u dekadni zapis:
   0 10000011 11010100000000000000000 + 0 10000001 01100000000000000000000

   Rešenje :

   (a) Sabiramo pozitivne brojeve, rezultat je pozitivan, tj znak je 0 (+).
   (b) Operandi se svode na jednake eksponente. $10000011 - 10000001 = (10)_2 = (2)_{10}$, pa se druga frakcija pomera za dva mesta udesno, pa dobijamo (0.01011). Sada je moguće sabrati frakcije.
   1.110101 + 0.01011 = 10.001011
   Rezultat je nije normalizovan, eksponent treba povećati za jedan, tj. dobijamo 10000100 pa je frakcija onda 1.0001011. Rešenje:
   0 10000100 00010110000000000000000
   (c) $(10000100)_2 = 132$ 132 - 127 = 5, odgovarajuća dekadna vrednost je $1.0001011 \cdot 2^5 = (100010.11)_2 = 34.75$

Množenje i deljenje u IEEE 754

Jednostavnije je od sabiranja i oduzimanja jer se ne vrši svođenje na isti eksponent:
$x = x_s \cdot 2^{x_e}$ i $y = y_s \cdot 2^{y_e}$
U opštem slučaju:
$x \cdot y = (x_s \cdot y_s) \cdot 2^{x_e + y_e}$
$\frac{x}{y} = \frac{x_s}{y_s} \cdot 2^{x_e - y_e}$

Pri izvođenju operacija potrebno je voditi računa o generisanju i propagaciji specijalnih vrednosti.

Osnovni koraci u množenju:

- Proverava se postojanje specijalnih vrednosti. Ukoliko neki od argumenata operacije predstavlja specijalnu vrednost, rezultat se određuje na osnovu odgovarajućih pravila za specijalne vrednosti.
- Ukoliko je bar jedan od činilaca jednak nuli, rezultat je 0.
- Saberu se vrednosti eksponenata i od dobijenog zbira oduzme uvećanje. Ako je došlo do prekoračenja pri ovom sabiranju, krajnji rezultat je $\pm\infty$ u zavisnosti od znaka brojeva x i y. Ako je pak do slo do potkoracenja vrednosti eksponenta, krajnji rezultat je pozitivna ili negativna (u zavisnosti od znaka brojeva x i y) nula.
- Pomnože se frakcije brojeva. Množenje se vrši prema pravilima za množenje celih brojeva zapisanih pomocu znaka i apsolutne vrednosti.
- Dobijeni rezultat se normalizuje sličnim postupkom kao kod sabiranja. Pri tome je moguće dobiti i denormalizovani broj.
- Broj (binarnih) cifara u proizvodu je dvostruko veći od broja cifara vrednosti koje su pomnožene; cifre koje su višak se odbacuju u procesu zaokruživanja.

Primeri :

1. 0 10000011 00100000000000000000000 * 0 10000010 00110000000000000000000.

   Rešenje :

   (a) Ni jedan od operanada nije specijalna vrednost ili nula.
   (b) Pošto se množe dva pozitivna broja, znak rezultata je + (0).
   (c) Vrednosti eksponenata se sabiraju i od dobijenog rezultata se oduzme uvećanje:
   (10000011 + 10000010) - 001111111 = 100000101 - 001111111 = 10000110

   (d) Množe se frakcije brojeva i dobija se normalizovan broj u kome nema potrebe za zaokruživanjem:
   $1.001 \cdot 1.0011 = 1.0101011$

(e) Konačan rezultat je: 0 10000110 01010110000000000000000

(f) $(10000110)_2 = (134)_{10}$, 134-127 = 7, pa je dekadna vrednost rezultata $(1.0101011) \cdot 2^7 = (10101011)_2 = (171)_{10}$.

Osnovni koraci u deljenju:

- Proverava se postojanje specijalnih vrednosti. Ukoliko je neki od argumenata operacije specijalna vrednost, rezultat se određuje na osnovu odgovarajućih pravila za specijalne vrednosti.
- Ako je delilac nula, tada:

  - ako je deljenik $\neq$ 0, količnik je $\pm\infty$ u zavisnosti od znaka deljenika

  - ako je deljenik = 0, tada je rezultat NaN
- Oduzmu se vrednosti eksponenata i na dobijenu razliku doda uvećanje. Ako je došlo do prekoračenja pri ovom sabiranju, krajnji rezultat je $\pm\infty$ u zavisnosti od znaka brojeva x i y. Ako je pak došlo do potkoračenja vrednosti eksponenta, krajnji rezultat je pozitivna ili negativna (u zavisnosti od znaka brojeva x i y) nula.
- Podele se frakcije brojeva. Deljenje se vrši prema pravilima za deljenje celih brojeva zapisanih pomocu znaka i apsolutne vrednosti.
- Dobijeni rezultat se normalizuje sličnim postupkom kao kod sabiranja. Pri tome je moguće dobiti i denormalizovan broj.
- Dobijeni kolicnik se zaokružuje prema pravilima za zaokruživanje.

Primer :

1. 0 10000100 11101110000000000000000 / 0 10000001 10100000000000000000000

   Rešenje :

   (a) Ni jedan od operanada nije specijalna vrednost.

   (b) Ni deljenik ni delilac nisu nule.

   (c) Znak rezultata je pozitivan jer su takva i oba argumenta operacije.

   (d) Eksponent rezultata se dobija kad se od vrednosti eksponenta deljenika oduzme vrednost eksponenta delioca i na dobijeni zbir doda 127 (vrednost uvećanja):
   (10000100 - 10000001) + 01111111 = 00000011 + 01111111 = 10000010

   (e) Frakcija razultata dobija se deljenjem frakcija kao neoznačenih binarnih brojeva:
   1.1110111 / 1.101 = 1.0011

(f) Dobijeni količnik je normalizovan i nema potrebe za dodantom normalizacijom, kao ni za zaokruživanjem jer je rezultat dobijen na odgovarajućem broju cifara.

(g) Rešenje: 0 10000010 00110000000000000000000

(h) $(10000010)_2 = 130$, 130 - 127 = 3, pa je dekadna vrednost broja $(1.0011) \cdot 2^3 = (1001.1)_2 = (9.5)_{10}$

1. Izvršiti računske operacije nad brojevima predstavljenim u IEEE754 zapisu sa binarnom osnovom i obavezno prevesti rezultat u dekadni zapis: 1 10000011 11000000000000000000000 * 0 10000001 00100000000000000000000

   Rešenje :

   - Množimo pozitivan i negativan broj, što znači da je rezultat negativan broj, tj. prvi bit proizvoda je 1.
   - eksponent:
     (10000011 + 10000001) - 01111111 = 100000100 - 01111111 = 10000101
   - frakcija:
     $1.11 \cdot 1.001 = 1.11111$, nema potrebe za normalizacijom
   - Rešenje je: 1 100000101 11111000000000000000000
   - Dekadni ekvivalent:
     eksponent: $(10000101)_2 = 133$ 133-127 = 6
     $-(1.11111) \cdot 2^6 = -(1111110)_2 = -126$

2. Izvršiti racunske operacije nad brojevima predstavljenim u IEEE754 zapisu sa binarnom osnovom i obavezno prevesti rezultat u dekadni zapis: 0 11111111 01101110000000000000000 / 0 10000001 10100000000000000000000

   Rešenje :
   Deljenik je SNaN pa je i rezultat operacije QNaN.

3. Izvršiti računske operacije nad brojevima predstavljenim u IEEE754 zapisu sa binarnom osnovom i obavezno prevesti rezultat u dekadni zapis: 1 10000101 01010100000000000000000 * 0 10000011 10110000000000000000000

   Rešenje :

   - Množimo pozitivan i negativan broj, što znači da je rezultat negativan broj, tj. prvi bit proizvoda je 1.
   - eksponent:
     (10000011 + 10000101) - 01111111 = 100001000 - 01111111 = 10001001
   - frakcija:
     $1.010101 \cdot 1.1011 = 10.0011110111$, treba normalizovati, tj. od eksponenta se sabere sa 1:
     10001001 + 1 = 10001010
   - Rešenje je: 1 10001010 00011110111000000000000
   - Dekadni ekvivalent:
     eksponent: $(10001010)_2 = 138$ 138-127 = 11
     $-(1.00011110111) \cdot 2^{11} = -(100011110111)_2 = -2295$

4. Izvršiti računske operacije nad brojevima predstavljenim u IEEE754 zapisu sa binarnom osnovom i obavezno prevesti rezultat u dekadni zapis: 0 10000101 11100000000000000000000 / 0 10000100 10000000000000000000000

   Rešenje :

   - Delimo pozitivne brojeve, što znači da je rezultat pozitivan broj, tj. prvi bit količnika je 0.
   - eksponent:
     (10000101 - 10000100) + 01111111 = 00000001 + 01111111 = 10000000
   - frakcija:
     $1.010101/1.1011 = 1.01$, ne treba normalizovati
   - Rešenje je: 0 10000000 01000000000000000000000
   - Dekadni ekvivalent:
     eksponent: $(10000000)_2 = 128$ 128-127 = 1
     $(1.01) \cdot 2^1 = (10.1)_2 = 2.5$

5. Izvršiti računske operacije nad brojevima predstavljenim u IEEE754 zapisu sa binarnom osnovom i obavezno prevesti rezultat u dekadni zapis: 0 10000110 01010110000000000000000 / 0 10000011 00100000000000000000000

   Rešenje :

   - Delimo pozitivne brojeve, što znači da je rezultat pozitivan broj, tj. prvi bit količnika je 0.
   - eksponent:
     (10000110 - 10000011) + 01111111 = 00000011 + 01111111 = 10000010
   - frakcija:
     $1.0101011/1.001 = 1.0011$, ne treba normalizovati
   - Rešenje je: 0 10000010 00110000000000000000000
   - Dekadni ekvivalent:
     eksponent: $(10000010)_2 = 130$ 130-127 = 3
     $(1.0011) \cdot 2^3 = (1001.1)_2 = 9.5$

6. Izvršiti racunske operacije nad brojevima predstavljenim u IEEE754 zapisu sa binarnom osnovom i obavezno prevesti rezultat u dekadni zapis: 0 10000010 11010000000000000000000 * 0 11111111 10000000000000000000000

   Rešenje :
   0 11111111 10000000000000000000000 je QNaN, a rezultat množenja u kome ucestvuje QNaN je ponovo QNaN.

7. Izvršiti računske operacije nad brojevima predstavljenim u IEEE754 zapisu sa binarnom osnovom i obavezno prevesti rezultat u dekadni zapis: 0 10000100 11101110000000000000000 / 1 10000001 10100000000000000000000

Rešenje :

- Delimo brojeve različitog znaka, što znači da je rezultat negativan broj, tj. prvi bit količnika je 1.
- eksponent:
  (10000100 - 10000001) + 01111111 = 00000011 + 01111111 = 10000010
- frakcija:
  $1.1110111/1.101 = 1.0011$, ne treba normalizovati
- Rešenje je: 0 10000010 00110000000000000000000
- Dekadni ekvivalent:
  eksponent: $(10000010)_2 = 130$ 130-127 = 3
  $(1.0011) \cdot 2^3 = (1001.1)_2 = 9.5$

Specijalne vrenosti, napomena

Sa izuzetkom operacija koje proizvode QNaN sve operacije koje ukljucuju ∞ takodje imaju ∞ kao rezultat. Međutim i dalje treba paziti na znak.
QNaN se propagira kroz aritmetičke operacije. Može se pojaviti u sledećim slučajevima:

- $(\pm\infty) - (\pm\infty)$
- $0 \cdot \infty$
- $\frac{0}{0}, \frac{\infty}{\infty}$
- $x\%0, \infty\%x$
- $\sqrt{x}$ gde $x < 0$
- Bilo koja operacija čiji je argument SNaN.