Digitalni zapis podataka

Stefan Mišković

5 Zapis mešovitih brojeva

Za zapis mešovitih brojeva u računaru, bez obzira koliko memorije je odvojeno za zapis broja, uvek može da se zapiše konačno mnogo brojeva. Zbog toga razlomljeni deo broja, ukoliko postoji, uvek mora biti zapisan sa konačnim brojem cifara. Svaki zapis određuje opseg brojeva koji se može zapisati, kao i broj dozvoljenih cifara u razlomljenom delu.

U početku nije postojao jedinstven standard za zapis mešovitih brojeva, pa je bilo moguće da različite arhitekture implementiraju zapis na različite načine, zbog čega može doći do grešaka i nepodudaranja pri prenosu podataka. Zbog potrebe sa unifikacijom načina zapisa, nastao je IEEE 754 standard. Njime je precizno propisan način zapisa brojeva, kao i algoritmi za različite operacije nad njima, ponašanje u graničnim situacijama, i slično. Kako danas sve mašine podržavaju ovaj standard, smanjena je verovatnoća greške pri prenosu podataka. IEEE 754 standard podržava zapis brojeva sa dekadnom i binarnom osnovom, a mogući su zapisi sa jednostrukom, dvostrukom i četvorostrukom tačnošću. Oni redom zauzimaju 32, 64 i 128 bitova. Radi jednostavnosti, svi zapisi koji će biti pomenuti u nastavku će biti u jednostrukoj tačnosti, odnosno imaće 32 bita.

Zaokruživanje. U računaru uvek može da se zapiše konačno mnogo brojeva, zbog čega se neki brojevi ne mogu precizno zapisati, već se pamti njihova približna vrednost, pri čemu se vrši zaokruživanje. Postoji nekoliko načina na koji se može izvršiti zaokruživanje, među kojima su:

- Zaokruživanje na parnu cifru – broj se zaokružuje na najbližu pretpostavljenu vrednost, a ako je na sredini intervala, onda se zaokruživanje vrši na parnu cifru. Tako se, na primer, brojevi 1.212, 1.218, 1.225 i 1.235 zaokružuju na dve decimale redom na vrednosti 1.21, 1.22, 1.22 i 1.24.

- Zaokruživanje ka $+\infty$ – broj se zaokružuje tako da mu je zaokružena vrednost uvek veća ili jednaka od stvarne. Tako se, na primer, pozitivni brojevi 1.21 i 1.29 na jednu decimalu zaokružuju redom na 1.3 i 1.3, a negativni brojevi -1.21 i -1.29 redom na -1.2 i -1.2.

- Zaokruživanje ka $-\infty$ – broj se zaokružuje tako da mu je zaokružena vrednost uvek manja ili jednaka od stvarne. Tako se, na primer, pozitivni brojevi 1.21 i 1.29 na jednu decimalu zaokružuju redom na 1.2 i 1.2, a negativni brojevi -1.21 i -1.29 redom na -1.3 i -1.3.

- Zaokruživanje ka nuli – broj se zaokružuje tako da mu je zaokružena vrednost uvek po apsolutnoj vrednosti manja ili jednaka od stvarne. Tako se, na primer, pozitivni brojevi 1.21 i 1.29 na jednu decimalu zaokružuju redom na 1.2 i 1.2, a negativni brojevi -1.21 i -1.29 redom na -1.2 i -1.2.

5.1 IEEE 754 standard sa dekadnom osnovom

Kod IEEE 754 standarda sa dekadnom osnovom, osnova je uvek 10, a frakcija može biti dekadna ili binarna. Ako je frakcija dekadna, radi se o DPD kodiranju, a ako je binarna, o BID kodiranju. Za obe vrste kodiranja će biti opisan zapis u jednostrukoj tačnosti, kada se brojevi kodiraju sa 32 bita.

DPD kodiranje. Ukoliko su i frakcija i osnova dekadne, IEEE 754 standard koristi DPD kodiranje. Najpre ćemo prikazati kako se mogu tri dekadne cifre kodirati sa 10 binarnih cifara, što će kasnije biti iskorišćeno za kodiranje proizvoljnog mešovitog broja. U nastavku će biti prikazana tabela za kodiranje, pomoću koje se od tri dekadne cifre dobija odgovarajući kod od 10 bitova (ova tabela, uključujući i njenu varijantu za dekodiranje ne mora da se uči napamet, već će biti data na ispitu):

aei	pqr stu v wxy
000	bcd fgh 0 jkl
001	bcd fgh 1 00l
010	bcd jkh 1 01l
100	jkd fgh 1 10l
110	jkd 00h 1 11l
101	fgd 01h 1 11l
011	bcd 10h 1 11l
111	00d 11h 1 11l

Neka je sada data uređena trojka celih dekadnih cifara $c_2c_1c_0$. Svaku od pomenutih cifara treba zapisati u BCD kodu u zapisu 8421, gde se dobija zapis $abcdefghijkl$. Na osnovu vrednosti bitova a, e i i treba se pozicionirati u odgovarajući red tabele, na osnovu čega se jednosavno može izračunati vrednost $pqrstuvwxy$, što je 10 bitova koji predstavlja odgovarajući kod.

Na primer, trocifren broj 918 se u zapisu 8421 piše kao

$$abcdefghijkl = 100100011000.$$

Kako je $aei = 101$, sledi da se treba pozicionirati u peti red tabele. Iz njega se vidi da je

$$pqrstuvwxy = fgd01h111l = 0010111110.$$

Obratno, pomoću tabele za dekodiranje se uređena desetorka bitova može dekodirati u 3 dekadne cifre. Dekodiranje se vrši pomoću sledeće tabele:

vwxst	abcd efgh ijkl
0....	0pqr 0stu 0wxy
100..	0pqr 0stu 100y
101..	0pqr 100u 0sty
110..	100r 0stu 0pqy
11100	100r 100u 0pqy
11101	100r 0pqu 100y
11110	0pqr 100u 100y
11111	100r 100u 100y

Neka je, na primer, dat niz od 10 bitova

$$pqrstuvwxy = 0010111110.$$

Kako je $vwxst = 11101$, treba se pozicionirati u peti red tabele. Dobija se da je

$$abcdefghijkl = 100r0pqu100y = 100100011000,$$

pa je traženi broj 918.

U nastavku ćemo kroz primere opisati kako se vrši kodiranje iz dekadne vrednosti u zapis i dekodiranje iz zapisa u dekadnu vrednost. Neka je, na primer, potrebno kodirati broj 123.4 po IEEE 754 standardu u dekadnoj osnovi u jednostrukoj tačnosti koristeći DPD kodiranje. Broj najpre treba napisati kao proizvod celobrojne frakcije i stepenovane osnove. Ovde je $123.4 = 1234 \cdot 10^{-1}$. Celobrojnu frakciju je uvek potrebno zapisati na 7 mesta. Ako ima manje od 7 značajnih cifara, dopunjuje se nulama, a ako ima više, vrši se zaokruživanje. Ovde je prilagođena frakcija oblika 0001234. Ona se sada deli na tri dela, gde prvi deo sadrži jednu cifru, a ostala dva dela po 3 cifre. U našem primeru frakcija se deli na 0, 001 i 234. Poslednja dva dela frakcija se kodiraju sa po 10 bitova koristeći tabelu za kodiranje. Kodirajmo najpre trojku 001. Zapisi 8421 dekadnih cifara 0, 0 i 1 su redom 0000, 0000 i 0001. Ako označimo da je $abcd = 0000$, $efgh = 0000$ i $ijkl = 0001$, budući da je $aei = 000$, iz tabele za kodiranje sledi da je $pqr = bcd$, $stu = fgh$, $v = 0$ i $wxy = jkl$, pa je traženi kod $pqrstuvwxy = 0000000001$. Na sličan način se dobija da se dekadna trojka 234 kodira sa 0100110100. Preostala je još prva cifra frakcije. Ukoliko je ona između 0 i 7, kodira se sa tri binarne cifre, tako što se cifra pretvori u binarni zapis na tri mesta, uz eventualno dodavanje vodećih nula (na primer, 0 se kodira kao 000, 5 kao 101 i 7 kao 111). Ukoliko je cifra frakcije 8 ili 9, ona se kodira sa jednom binarnom cifrom. Osmica se kodira sa 0, a devetka sa 1. U našem primeru je prva cifra frakcije 0, pa se ona kodira kao 000. Ostalo je još da se kodira eksponent. Eksponent -1 je potrebno zapisati u binarnom sistemu na 8 mesta sa uvećanjem 101. Dakle, zapis eksponenta je oblika $-1 + 101 = 100 = (01100100)_2$. Na osnovu koda prve cifre frakcije (3 bita) i eksponenta (8 bitova), formira se kombinacija koja se sastoji od 11 bitova. (Ako je prva cifra 8 ili 9, ona se kodira samo sa jednom binarnom cifrom, pa se kombinacija od 11 bitova dobija na nešto drugačiji način, što će biti objašnjeno u narednom primeru.) Kombinacija se dobija tako što se eksponent razdvoji na dva dela, levi koji sadrži prva dva i desni koji sadrži poslednjih šest bitova, a između njih se ubaci trobitni kod početka frakcije. U našem slučaju, kombinacija je 01000100100. Broj je pozitivan, pa je bit za znak 0. Konačno, prvi bit zapisa je za znak, narednih 11 bitova čini kombinacija, a preostalih 20 ostatak frakcije. Zapis traženog broja je 0 01000100100 0000000001 0100110100.

Pretpostavimo da je potrebno kodirati na isti način broj $-982.5294 \cdot 10^{42}$. Pogodan zapis broja za kodiranje je $-9825294 \cdot 10^{38}$. Kod cifre 9 je 1, a desetobitni kodovi trojki 825 i 294 su redom 1000101101 i 0101011010. Ovi kodovi su dobijeni koristeći tabelu za kodiranje, slično kao u prethodnom primeru. Zapis eksponenta je $38 + 101 = 139 = (10001011)_2$. Kombinacija od 11 bitova se i ovde dobija od ekponenta i koda prve cifre frakcije, ali je ovde problem sto zapis početka frakcije ima samo 1 bit. U slučaju kada je prva cifra frakcije 8 ili 9 (kada se ona ustvari kodira jednim bitom), na početku kombinacije se dodaje 11, a ostatak se dobija analogno, razdvajanjem eksponenta i ubacivanjem cifre frakcije. Dakle, u ovom slučaju kombinacija se sastoji od 11, početka eksponenta 10, početka frakcije 1 i nastavka eksponenta 001011, pa je ona oblika 11101001011. Uzimajući u obzir i da je broj negativan, njegov zapis je 1 11011001011 1000101101 0101011010.

Objasnimo sada obratni proces, kada se vrši prevođenje iz zapisa u dekadni broj na primeru 32-bitnog zapisa 0 01111011110 0000000000 0000110101. Najpre treba posmatrati kombinaciju, jer tu mogu postojati dva slučaja – kada je prva cifra frakcije između 0 i 7, odnosno kada je 8 ili 9. Ukoliko ona počinje sa 11, radi se o početnoj cifri 8 ili 9, a inače, početna cifra je između 0 i 7. U našem primeru prve dve cifre i poslednjih 6 cifara kombinacije su cifre eksponenta, a preostala trojka 111 kodira prvu cifru frakcije čija je vrednost 7. Eksponent iznosi $(01011110)_2 - 101 = 94 - 101 = -7$. Na osnovu tabele za dekodiranje (videti lekciju gde se prvi put pominje DPD), dobija se da kodovi 0000000000 i 000011010 redom predstavljaju trojke dekadnih cifara 000 i 035. Kako je broj pozitivan, konačno se dobija da je tražena vrednost $7000035 \cdot 10^{-7} = 0.7000035$.

Pored zapisa konačnih brojeva, IEEE 754 standard sa dekadnom osnovom (i kod DPD i kod BID kodiranja) podržava zapis specijalnih vrednosti. Kada se vrši dekodiranje, potrebno je prvo razmotriti da li se radi o nekoj specijalnoj vrednosti, pa ako se ustanovi da je u pitanju konačan broj, treba pribeći samom algoritmu. Specijalne vrednosti koje imaju poseban zapis u ovom standardu su:

- Nula se može pojaviti kao konačna vrednost ili kao neka veoma mala vrednost koja se ne može zapisati, kada dolazi do potkoračenja, gde se ta vrednost zapisuje da je približno jednaka nuli. Može postojati pozitivna ili negativna nula, pa joj bit za znak može biti 0 ili 1. Na svim bitovima kojim je označena frakcija je potrebno da stoje nule, a na bitovima gde je eksponent može biti proizvoljna vrednost. To znači da su treći, četvrti i peti bit kombinacije, kao i poslednjih 20 bitova zapisa jednaki 0.

- Beskonačno se može pojaviti kada se neki veoma veliki broj ne može zapisati ili kao rezultat operacija poput $5 - \infty = -\infty$, $\infty + \infty = \infty$, $-5/0 = -\infty$, i slično. U zavisnosti od toga da li se radi o pozitivnoj ili negativnoj beskonačnosti, bit za znak može biti 0 ili 1. Za beskonačno važi da je početak kombinacije oblika 11110.

- NaN vrednosti se javljaju u izuzetnim situacijama. Postoji signalni NaN (sNaN) i tihi NaN (qNaN). Signalni NaN se javlja pri greškama u inicijalizaciji ili konverziji, a tihi NaN pri greškama u aritmetičkim operacijama. Primeri za pojavu qNaN vrednosti su nedefinisane vrednosti poput $\infty - \infty$, $0 \cdot \infty$, $0/0$, itd. Ukoliko je argument aritmetičke operacije sNaN ili qNaN, rezultat je qNaN. Kod sNaN vrednosti kombinacija počinje sa 111111, a kod qNaN vrednosti sa 111110. Ostale cifre zapisa mogu biti proizvoljne.

BID kodiranje. Ukoliko je osnova dekadna, a frakcija binarna, IEEE 754 standard koristi BID kodiranje. U nastavku će biti navedeni primeri kodiranja i dekodiranja u jednostrukoj tačnosti. Specijalne vrednosti se zapisuju na isti način kao u slučaju DPD kodiranja.

Neka je, na primer, potrebno odrediti zapis broja -14.37. Pogodan zapis broja za kodiranje je

$$-14.37 = -1437 \cdot 10^{-2} = (-10110011101)_2 \cdot 10^{-2}.$$

Dakle, broj treba zapisati tako da mu je frakcija celobrojna, a zatim tu frakciju prevesti u binarni sistem. Osnova ostaje sve vreme dekadna. Ukoliko frakcija sadrži manje od 23 bita, potrebno je dopisati vodeće nule, kako bi bila dopunjena do tog broja bitova. U ovom slučaju, zapis frakcije bi bio 00000000000010110011101. Eksponent se i ovde piše na 8 mesta sa uvećanjem 101 i iznosi $-2+101 = 99 = (01100011)_2$. Zapis broja se sastoji iz tri dela, jednobitnog znaka, kombinacije od 11 bitova koja se formira nadovezivanjem cifara eksponenta i prve tri cifre frakcije i nastavka frakcije od 20 bitova. Kako je u pitanju negativan broj, zapis je 1 01100011000 00000000010110011101.

Neka je potrebno izvršiti prevođenje broja 9000001. Pogodan zapis za kodiranje je

$$9000001 \cdot 10^0 = (100010010101010001000001)_2 \cdot 10^0.$$

Kada frakcija ima 24 bita, što je ovde slučaj, njena prva tri bita se ingorišu (podrazumevano su uvek jednaki 100). Kombinacija se dobija dodavanjem para bitova 11 na početak, zatim 8 cifara eksponenta i četvrtog bita frakcije. Ostalih 20 bitova se dodaju u nastavku frakcije. Eksponent je $0 + 101 = 101 = (01100101)_2$, pa je kombinacija 11011001010. Broj je pozitivan, pa je njegov zapis 0 11011001010 10010101010001000001.

Izvršimo sada dekodiranje na primeru zapisa 1 01100110000 00000000000000001011. Nakon što se utvrdi da se ne radi ni o kakvoj specijalnoj vrednosti, treba proveriti da li kombinacija počinje sa 11. Ako počinje, razmatra se slučaj dekodiranja gde frakcija ima 24 cifre, a inače slučaj kada frakcija ima 23 cifre. Ovde prvih 8 bitova kombinacije predstavlja eksponent, a poslednja tri početak frakcije. Eksponent je $(01100110)_2 - 101 = 102 - 101 = 1$, a frakcija $(-1011)_2 = -11$. U pitanju je broj $-11 \cdot 10^1 = -110$.

5.2 IEEE 754 standard sa binarnom osnovom

Kod IEEE 754 standarda sa binarnom osnovom, frakcija je binarna, a osnova 2. Zapis se sastoji iz tri dela – zapisa znaka, eksponenta i frakcije. I ovde će biti razmotrena jednostruka tačnost, gde se broj kodira sa 32 bita. IEEE 754 standard sa binarnom osnovom propisuje zapis normalnih i subnormalnih brojeva, kao i specijalnih vrednosti. Podržane specijalne vrednosti su 0, beskonačno i NaN vrednost. Nula se piše sa svim nulama u eksponentu i frakciji, a bit za znak može biti 0 ili 1, u zavisnosti od znaka nule. Beskonačno se piše sa svim jedinicama u eksponentu i svim nulama u frakciji. Znak za beskonačno takođe može biti 0 ili 1. Obe NaN vrednosti se pišu sa svim jedinicama u eksponentu. Kod sNaN vrednosti, prvi bit frakcije je 0, a kod ostatka frakcije nisu svi bitovi jednaki nuli, jer bi se tada poklapao zapis sa ∞, a ostale kombinacije su dozvoljene. Kod qNaN vrednosti, prvi bit frakcije je 1, a ostali bitovi mogu biti proizvoljni.

U jednostrukoj tačnosti u zapisu se znak kodira pomoću jednog bita, eksponent pomoću 8, a frakcija pomoću 23. Prikažimo algoritam prevođenja na primeru broja 13.25. Pogodan zapis za prevod je $13.25 = (1101.01)_2 = (1.10101)_2 \cdot 2^3$. Dekadni broj je najpre potrebno prevesti u binarni, nakon čega frakciju treba dovesti na oblik $1.f$ (f je ostatak frakcije iza decimalne tačke) i ažurirati po potrebi eksponent. Ekponent se zapisuje na 8 mesta sa uvećanjem 127 i iznosi $3 + 127 = 130 = (10000010)_2$. Pri zapisu frakcije, ceo deo (prva jedinica) se ignoriše, budući da je uvek ista i nema potrebe je pamtiti. Kako je broj pozitivan, njegov zapis je 0 10000010 10101000000000000000000. Primetimo da je ostatak frakcije dopunjen nulama.

Obratno, pretpostavimo da je dat zapis 1 10000010 10101000000000000000000 i neka je potrebno odrediti njegovu dekadnu vrednost. Za vrednost eksponenta broja se dobija

$(10000010)_2 - 127 = 130 - 127 = 3$. Dodavanjem podrazumevane jedinice, i imajući u vidu da je broj negativan, sledi da je frakcija $(-1.10101)_2$. Dekadna vrednost broja je $(-1.10101)_2 \cdot 2^3 = (-1101.01)_2 = -13.25$.

IEEE 754 standard sa binarnom osnovom podržava i zapis subnormalnih brojeva. Subnormalni brojevi su oni brojevi koji su veoma bliski nuli. Prepoznaju se tako što imaju sve nule u eksponentu. Kada se iščitava, njihova frakcija je, za razliku od normalnih brojeva, oblika $0.f$. Za jednostruku tačnost, eksponent iznosi -126. Na primer, na osnovu zapisa 0 00000000 00010000000000000000000, zaključujemo da se radi o subnormalnom broju. Na osnovu početka zapisa frakcije 0001 (ostatak su nule, pa nisu od značaja), zaključujemo da je frakcija oblika 0.0001. Broj je pozitivan, pa je vrednost datog denormalizovanog broja $(0.0001)_2 \cdot 2^{-126} = 2^{-4} \cdot 2^{-126} = 2^{-130}$.

5.3 Aritmetičke operacije

U nastavku će biti opisane operacije sabiranja, oduzimanja, množenja i deljenja u IEEE 754 standardu na primeru zapisa sa binarnom osnovom. Pre izvođenja bilo koje operacije, treba ustanoviti da li je rezultat eventualno neka specijalna vrednost. Ukoliko nije, operacija se može izvršiti odgovarajućih algoritmom. Na kraju izvršene operacije, eventualno može doći do prekoračenja ili potkoračenja, kada se rezultat redom proglašava za beskonačno ili nulu.

Sabiranje. Neka su data dva broja u IEEE 754 standardu u binarnoj osnovi u obliku $f_1 \cdot 2^{e_1}$ i $f_2 \cdot 2^{e_2}$, gde su f_1 i f_2 frakcije, a e_1 i e_2 eksponenti. Za sabiranje je brojeve potrebno dovesti na isti eksponent, a u ovom algoritmu se uzima onaj koji je veći. Pretpostavimo da je $e_1 \geq e_2$ i da je nakon svođenja na eksponent e_1 drugi sabirak oblika $f_2' \cdot 2^{e_1}$. Sada se sabiranje svodi na

$$f_1 \cdot 2^{e_1} + f_2 \cdot 2^{e_2} = f_1 \cdot 2^{e_1} + f_2' \cdot 2^{e_1} = (f_1 + f_2') \cdot 2^{e_1}.$$

Na primer, neka je potrebno izvršiti sabiranje 0 10000010 10011000000000000000000 i 0 10000001 10110000000000000000000. Bit za znak rezultata je 0, budući da se vrši sabiranje dva pozitivna broja. Eksponent rezultata je jednak većem od eksponenata sabiraka i iznosi 10000010. Sada je potrebno sabrati odgovarajuće frakcije. Kako je eksponent rezultata jednak eksponentu prvog sabirka, frakcija prvog sabirka ostaje nepromenjena i iznosi 1.10011. Eksponent drugog sabirka treba povećati za 1, pa je potrebno izvršiti modifikaciju njegove frakcije 1.1011 pomeranjem decimalne tačke za jedno mesto ulevo. Dakle, njegova frakcija je oblika 0.11011, pa je frakcija zbira jednaka $1.10011+0.11011 = 10.0111$. Kada se ne dobije normalizovana frakcija, potrebno je izvršiti njenu normalizaciju i u skladu sa tim promeniti vrednost eksponenta. Važi da je $(10.0111)_2 = (1.00111)_2 \cdot 2^1$, pa eksponent rezultata treba uvećati za 1. On sada iznosi 10000011. Zapis zbira je 0 10000011 00111000000000000000000.

Oduzimanje. Neka su data dva broja u obliku $f_1 \cdot 2^{e_1}$ i $f_2 \cdot 2^{e_2}$. Kod oduzimanja je potrebno eksponent umanjioca dovesti na eksponent umanjenika. Pretpostavimo da je nakon svođenja na eksponent e_1 umanjilac oblika $f_2' \cdot 2^{e_1}$. Sada se oduzimanje svodi na

$$f_1 \cdot 2^{e_1} - f_2 \cdot 2^{e_2} = f_1 \cdot 2^{e_1} - f_2' \cdot 2^{e_1} = (f_1 - f_2') \cdot 2^{e_1}.$$

Na primer, neka je potrebno izvršiti oduzimanje 0 10000001 01011000000000000000000 i 0 01111110 00100000000000000000000. Bit za znak razlike je 0, budući da se vrši oduzimanje dva pozitivna broja, gde se oduzima manji broj od većeg. Eksponent razlike je jednak eksponentu umanjenika. Frakciju umanjioca treba pomeriti za odgovarajući broj mesta ulevo kako bi se eksponent izjednačio sa eksponentom umanjenika. U ovom slučaju, frakcija umanjioca 1.001 postaje 0.001001. Decimalnu tačku je potrebno pomeriti tri mesta ulevo zbog toga što se njegov eksponent povećava za 3, budući da je $(10000001)_2 - (01111110)_2 = 3$. Oduzimanjem frakcija se dobija $1.010110 - 0.001001 = 1.001101$. Frakcija razlike je normalizovana, pa je rezultat 0 10000001 00110100000000000000000.

Množenje. Neka su data dva broja u obliku $f_1 \cdot 2^{e_1}$ i $f_2 \cdot 2^{e_2}$. Njihovim množenjem se dobija

$$(f_1 \cdot 2^{e_1}) \cdot (f_2 \cdot 2^{e_2}) = (f_1 f_2) \cdot 2^{e_1+e_2}.$$

Iz poslednjeg sledi da je frakcija proizvoda jednaka proizvodu frakcija, a eksponent proizvoda zbiru eksponenata. Pri sabiranju eksponenata treba voditi računa da su oni u višku 127, koji nakon sabiranja treba anulirati.

Na primer, neka je potrebno izvršiti množenje 0 10000011 00100000000000000000000 i 0 10000010 00110000000000000000000. Kako se množe dva pozitivna broja, rezultat je takođe pozitivan. Kako su eksponenti oba činioca zapisani u višku 127, a eksponent proizvoda je potrebno takođe napisati u višku 127, pri sabiranju eksponenata činioca, da bi se dobio zapis eksponenta rezultata, potrebno je od dobijenog zbira oduzeti 127. Dakle, dobija se:

```
   10000011
+  10000010
-----------
  100000101
-  01111111
-----------
   10000110
```

Zatim se množenjem frakcija brojeva (postupak množenja je sličan množenju dva dekadna broja) dobija $1.001 \cdot 1.0011 = 1.0101011$. Dobijenu frakciju nije potrebno normalizovati, jer je već svedena na oblik $1.f$. Proizvod je 0 10000110 01010110000000000000000.

Deljenje. Neka su data dva broja u obliku $f_1 \cdot 2^{e_1}$ i $f_2 \cdot 2^{e_2}$. Njihovim deljenjem se dobija

$$\frac{f_1 \cdot 2^{e_1}}{f_2 \cdot 2^{e_2}} = \frac{f_1}{f_2} \cdot 2^{e_1-e_2}.$$

Iz poslednjeg sledi da je frakcija količnika jednaka količniku frakcija, a eksponent količnika razlici eksponenata. Pri oduzimanju eksponenata treba voditi računa da su oni u višku 127, koji nakon oduzimanja treba dodati.

Na primer, neka je potrebno izvršiti deljenje 0 10000100 11101110000000000000000 i 0 10000001 10100000000000000000000. Kako se dele dva pozitivna broja, rezultat je takođe pozitivan broj. Kako su eksponenti i deljenika i delioca zapisani u višku 127, a eksponent količnika je potrebno takođe napisati u višku 127, pri oduzimanju eksponenata deljenika i delioca, da bi se dobio zapis eksponenta rezultata, potrebno je dobijenoj razlici dodati 127. Dakle, dobija se:

```
   10000100
-  10000001
-----------
   00000011
+  01111111
-----------
   10000010
```

Zatim se deljenjem frakcija brojeva (postupak deljenja je analogan deljenju dva dekadna broja) dobija $1.1110111/1.101 = 1.0011$. Dobijenu frakciju nije potrebno normalizovati, jer je već svedena na oblik $1.f$. Dobijeni količnik je

$$0\ 10000010\ 00110000000000000000000.$$