Realni brojevi u pokretnom zarezu

Predstavljaju se pomoću osnove β (koja je uvek parna) i preciznosti p.

Primer:

- β=10, p=4: broj 0.4 se predstavlja kao 4.000×10^{-1}
- β=10, p=4: broj broj 564000000000000000000000000 se predstavlja kao 5.640×10^{26}
- β=10, p=4: broj broj 564000055555555555555555555 se predstavlja kao 5.640×10^{26}
- β=2 i p=10 broj 0.4 se predstavlja kao $1.100110011 \times 2^{-2}$.

Opšti slučaj

$$\pm d_0.d_{-1}d_{-2}...d_{-(p-1)}\beta^e$$

Oznake:

- $d_0.d_{-1}d_{-2}...d_{-(p-1)}$ – *značajni deo* (eng. *significand*). Zapisuje se u brojčanom sistemu sa osnovom β, tj. $0 \leq d_i < \beta$.
- β – *osnova*
- e – *eksponent*
- p – *preciznost*.

Zapis broja za koji važi da je $d_0 \neq 0$ se naziva **normalizovan**.

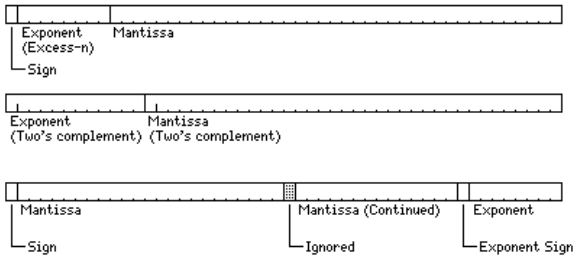
U savremenim računarima β=2 ili β=16.

Zapis brojeva u pokretnom zarezu

Različit zapis realnih brojeva u pokretnom zarezu kroz istoriju:

- pre-IEEE754 format
- IEEE754 format
- IEEE854 format
- IEEE754R format

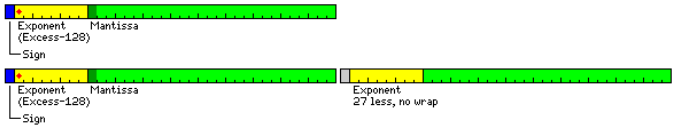
Formati zapisa realnih brojeva u pokretnom zarezu



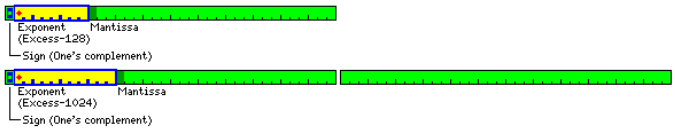
Slika 1: Formati zapisa relanih brojeva u pokretnom zarezu kroz istoriju

Group I

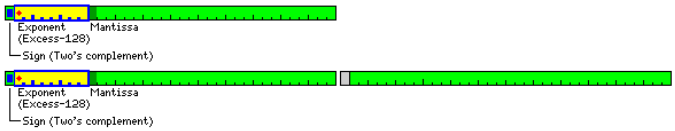
International Business Machines 704, 709, 7040, 7044, 7090, 7094, 7094 II



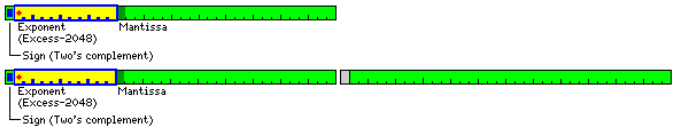
Univac 1107, 1108



Digital Equipment Corporation PDP-6, PDP-10, DECSYSTEM-20

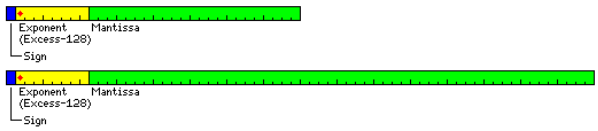


Expanded range (KL-10 only)



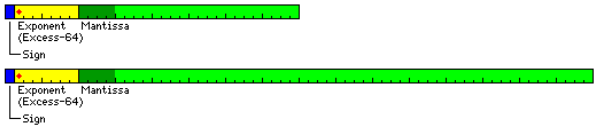
KA-10 Double Precision

Digital Equipment Corporation PDP-11, VAX-11

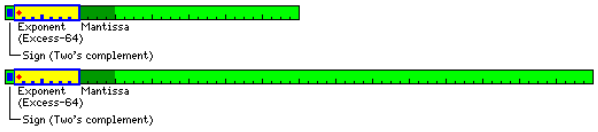


Digital Equipment Corporation PDP-8 Special (8K FORTRAN)

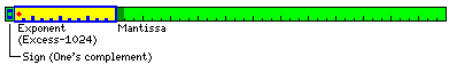
International Business Machines System/360, System/370, ESA/390, z/Architecture



Xerox Data Systems Sigma

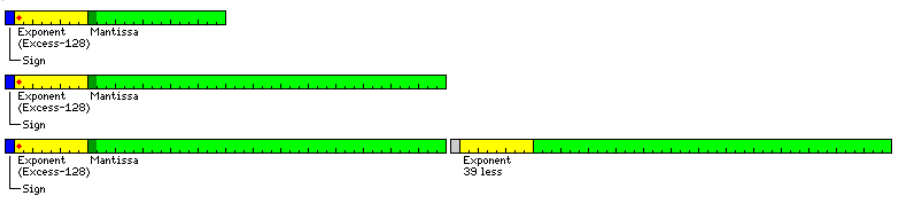


Control Data Corporation 1604, 3600

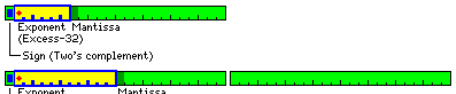


Control Data Corporation 6600

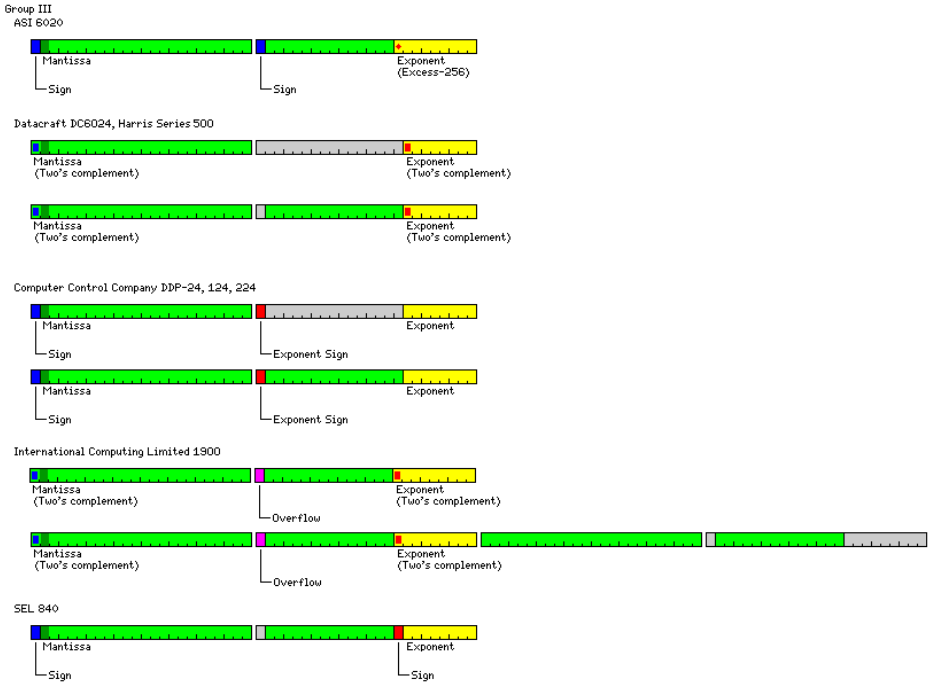
English Electric KDF9



Foxboro FOX-1

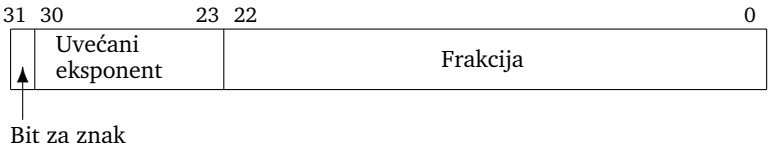


Slika 3: Formati zapisa relanih brojeva u pokretnom zarezu kroz istoriju (nastavak)



Slika 4: Formati zapisa relanih brojeva u pokretnom zarezu kroz istoriju (nastavak)

Primer zapisa sa binarnom osnovom - PDP-11, VAX-11



Slika 5: Format zapisa realnog broja pomoću binarne osnove

Važi

- Vrednost eksponenta se povećava za 1 dok frakcija postaje

$$0.d_0 d_{-1} d_{-2} ... d_{-(p-1)}$$

- Frakcija $d_0 d_{-1} d_{-2} ... d_{-(p-1)}$ se predstavlja preko 24 bita, sa 23 binarne pozicije na mestima 0-22.
- Eksponent se zapisuje u 8 bita na pozicijama 23–30 uz uvećanje od 128.

		Znak	Eksponent	Frakcija
$+15$	=	0	10000100	11100000000000000000000
-15	=	1	10000100	11100000000000000000000
$+1/64$	=	0	01111011	00000000000000000000000
0	=	0	00000000	00000000000000000000000
$(1-2^{-24})\times 2^{+127}$	=	0	11111111	11111111111111111111111
$+1\times 2^{-128}$	=	0	00000001	00000000000000000000000

Provera vrednosti zapisa broja +15:
– Znak = +
– Eksponent = 4 (=132-128)
– Frakcija = $(0.1111)_2 = +1\times2^{-1}+1\times2^{-2}+1\times2^{-3}+1\times2^{-4}$
– Vrednost = Znak frakcija * $2^{eksponent} = (+1\times2^{-1}+1\times2^{-2}+1\times2^{-3}+1\times2^{-4}) * 2^4$
$= +1\times2^3+1\times2^2+1\times2^1+1\times2^0 = 8 + 4+ 2 +1 = +15$

Tabela 1: Zapis realnih brojeva u jednostrukoj tačnosti / binarna osnova

Za veličinu eksponenta e važi

$$-2^7 \leq e \leq 2^7 - 1$$

Za vrednost s kojom se predstavlja frakcija važi

$$2^{-1} \leq |\, s \,| \leq 1 - 2^{-24}$$

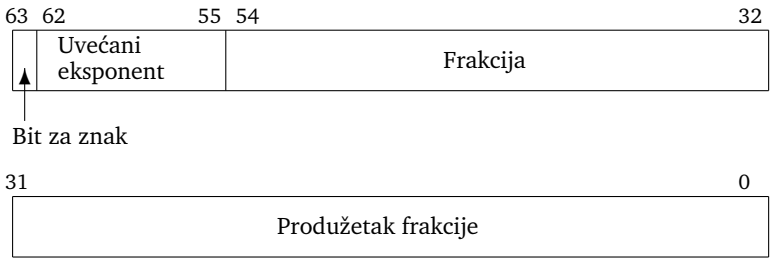
Na osnovu prethodnog, mogu se zapisati brojevi u intervalu

$$2^{-1} * 2^{-128} \leq |\, x \,| \leq (1 - 2^{-24}) * 2^{+127}$$

Medjutim, kako važi da je kod za broj $2^{-1} * 2^{-128}$ predstavlja 0, opseg je

$$2^{-128} \leq |\, x \,| \leq (1 - 2^{-24}) * 2^{+127}, \quad \text{odnosno}$$

$$2.9 * 10^{-39} < |\, x \,| < 1.7 * 10^{+38}$$



Slika 6: Format zapisa realnog broja u dvostrukoj tačnosti pomoću binarne osnove

Interval realnih brojeva koji mogu da se predstave u dvostrukoj tačnosti je

$$2^{-128} \leq |x| \leq (1-2^{-56}) * 2^{+127}$$

Zapis brojeva u pokretnom zarezu pomoću binarne osnove - IEEE754 format)

Karakteristike izračunavanja sa realnim brojevima pre pojave standarda IEEE 754:

1. Slaba prenosivost numerički intenzivnih programa.
2. Postoje razlike u rezultatima u izvršavanju istog programa na različitim računarskim sistemima.
3. IEEE 754 poboljšava prenosivost programa propisujući
    - algoritme za operacije sabiranja, oduzimanja, množenja, deljenja i izračunavanje kvadratnog korena
    - način njihove implementacije
    - načine zaokruživanja i ponašanja u graničnim slučajevima

Neki problemi koji se javljaju pri izračunavanjima sa realnim brojevima su:

1. Veličina greške pri zapisu i zaokruživanju broja

β=10 i p=3. Neka je rezultat izračunavanja 5.76×10^{-2}, matematički tačna vrednost računata sa beskonačnom preciznošću 0.0574. Greška zapisa je veličine 2×jedinična vrednost na poslednjem mestu zapisa broja.

$0.0574367 \longrightarrow 5.74\times10^{-2}$ – greška je 0.367 jedinica na poslednjem mestu.

Veličina "jedinica na poslednjem mestu" se označava sa *ulp* (eng. *unit in the last place*.

U opštem slučaju, ako broj $d_0.d_{-1}...d_{-(p-1)} \times \beta^e$ predstavlja broj z, tada je greška u zapisu $\mid d_0.d_{-1}...d_{-(p-1)} - (z/\beta^e) \mid \beta^{p-1}$ ulp-a.

Relativna greška je apsolutna vrednost razlike izmedju realnog broja i njegove reprezentacije podeljena sa apsolutnom vrednošću realnog broja.

Na primer, relativna greška pri aproksimaciji 5.74367 sa 5.74×10^0 je 0.00367/5.74367≈0.0006.

ulp i relativna greška zavise od tzv. *mašinskog* $\varepsilon=(\beta/2)\beta^{-p}$; Relativna greška uvek zapisuje kao faktor od ε. Na primer, u prethodnom primeru je $\varepsilon=(\beta/2)*\beta^{-p}=5*(10)^{-3}=0.005$. Tako se relativna greška može izraziti kao

$$((0.00367/5.74367)/0.005)\varepsilon \approx 0.12\varepsilon$$

Razlika izmedju *ulp* i relativne greške:

$x = 12.35$ je aproksimiran sa $x_a = 1.24 \times 10^1$.

Greška je $0.5ulp$, a relativna greška je 0.8ε.

$8 * x_a$: tačna vrednost je $8 * x = 98.8$, izračunata vrednost je $8 * x_a = 9.92 \times 10^1$. Greška merena u *ulp* je 8 puta veća, relativna greška je ista.

Ako je realan broj zapisan sa greškom od n *ulp*-a, tada je broj cifara obuhvaćenih greškom $log_\beta n$, a ako je relativna greška $n\varepsilon$ tada je broj obuhvaćenih cifara $\approx log_\beta n$.

2. **Cifre čuvari**. Neka je npr. $p = 6$, β=10 i neka treba naći razliku 10000.1-9999.93:

x =	1.00001×10^4
y =	0.99999×10^4
x-y =	0.00002×10^4

Tačan odgovor je 0.17, greška od oko 30 *ulp*-a;

Operacija sa dodatnim ciframa, npr. sa $P+1$ cifrom:

x =	1.000010×10^4
y =	0.999993×10^4
x-y =	0.000017×10^4

3. **Tačno zaokružene operacije.**

Operacije koje se izvode tako da se izračuna tačna vrednost a zatim zaokruži na najbliži broj u pokretnom zarezu se nazivaju *tačno zaokružene*.

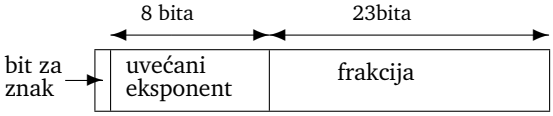
Zaokruživanje:

- na najbližu vrednost (4,48$\longrightarrow$ 4,5; 4,34$\longrightarrow$ 4,3; 4,45$\longrightarrow$?)
- na parnu cifru (4,45$\longrightarrow$ 4,4)

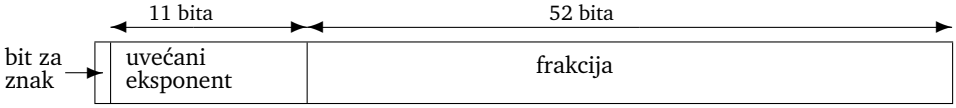
Opis IEEE 754

$\beta=2$

$$\pm 1.d_{-1}d_{-2}...d_{-(p-1)} \times 2^e$$



a) Jednostruka tačnost



b) Dvostruka tačnost

Slika 7: Format zapisa realnog broja prema standardu IEEE 754

		Znak	Eksponent	Frakcija
$+15$	=	0	10000010	11100000000000000000000
-15	=	1	10000010	11100000000000000000000
$+1/64$	=	0	01111001	00000000000000000000000
$+0$	=	0	00000000	00000000000000000000000
-0	=	1	00000000	00000000000000000000000
$(1-2^{-24})\times 2^{+128}$	=	0	11111110	11111111111111111111111
$+1\times 2^{-126}$	=	0	00000001	00000000000000000000000
$+1\times 2^{-149}$	=	0	00000000	00000000000000000000001

Provera vrednosti zapisa broja $+15$:

– Znak = +
– Eksponent = 3 (=130-127)
– Frakcija = $(1.111)_2 = 1\times 2^0+1\times 2^{-1}+1\times 2^{-2}+1\times 2^{-3}$
– Vrednost = Znak frakcija * $2^{eksponent} = (+1\times 2^0+1\times 2^{-1}+1\times 2^{-2}+1\times 2^{-3})$ * 2^3
$= +1\times 2^3+1\times 2^2+1\times 2^1+1\times 2^0 = 8 + 4+ 2 +1 = +15$

Tabela 2: Zapis realnih brojeva u jednostrukoj tačnosti prema IEEE 754 standardu

Parametar	Format				
	Jednostruki	Jednostruki proširen	Dvostruki	Dvostruki proširen	Četvorostruki
Dužina formata	32	≥43	64	≥79	128
Dužina eksponenta	8	≥11	11	≥15	15
Dužina frakcije	23	≥31	52	≥63	112
Preciznost (p)	24	≥32	53	≥64	113
Najveći eksponent (e_{max})	+127	≥+1023	+1023	≥+16383	+16383
Najmanji eksponent (e_{min})	-126	≤-1022	-1022	≤-16382	-16382
Uvećanje eksponenta	127	≥1023	1023	≥16383	16383
N_{max}	$(1\text{-}2^{-24})\times 2^{128}$ $\approx 3.4\times 10^{+38}$	xxx	$(1\text{-}2^{-53})\times 2^{1024}$ $\approx 1.8\times 10^{+308}$	xxx	$(1\text{-}2^{113})\times 2^{16384}$ $\approx 1.2\times 10^{+4932}$
N_{min}	1.0×2^{-126} $\approx 1.2\times 10^{-38}$	xxx	1.0×2^{-1022} $\approx 2.2\times 10^{-308}$	xxx	1.0×2^{-16382} $\approx 3.4\times 10^{-4932}$
D_{min}	1.0×2^{-149} $\approx 1.4\times 10^{-45}$	xxx	1.0×2^{-1074} $\approx 4.9\times 10^{-324}$	xxx	1.0×2^{-16494} $\approx 6.5\times 10^{-4966}$
$Prec_{dek}$	6–9	xxx	15–17	xxx	33–36

D_{min}	Najmanji (po apsolutnoj vrednosti) predstavljivi denormalizovani broj
N_{max}	Najveći (po apsolutnoj vrednosti) predstavljivi broj
N_{min}	Najmanji (po apsolutnoj vrednosti) predstavljivi normalizovani broj
$Prec_{dek}$	Značajnih dekadnih cifara
xxx	Zavisi od implementacije

Tabela 3: Pregled IEEE 754 formata zapisa

Specijalne vrednosti

- Označena nula
  - Obavezno je dodefinisanje $+0 = -0$;
    Ne važi ekvivalencija $x = y \Leftrightarrow 1/x = 1/y$ za $x = +0, y = -0$.
  - Utiče na znak rezultata, npr. +123*(-0) = -0;
  - $\forall x$ važi 1/(1/x) = x (uključujući i $x = \pm\infty$)
  - Olakšana upotreba funkcija koje imaju potkoračenje i prekid u 0. Npr. definiše se `log 0` =-∞ i `log x = NaN` za svakio x<0. Ako x$\longrightarrow$ 0 tada log x= NaN; bez označene nule bi bilo log x =-∞.
- Denormalizovani brojevi
  - Uvek važe relacije oblika $x = y \Leftrightarrow x - y = 0$.
  - Programske naredbe oblika `if (x<>y) then z=1/(x-y)` ne izazivaju prekid zbog deljenja nulom.
  - Smanjuje se broj programskih prekida zbog potkoračenja

  Bez denormalizovanih brojeva prethodni primeri ne bi važili za $x = 1,0002 \times 2^{-125}$ i $y = 1,0001 \times 2^{-125}$.
- Beskonačno
  - Omogućava nastavak izvršavanja programa u slučaju prekoračenja.
- Nan (*Not-a-number*, nije broj)
  - Tihi NaN. Signalizira izuzeto stanje kod aritmetičkih operacija, poredjenja, i konverzije kod operacija koje su deo standarda.
  - Signalni NaN. Označava pojavu nedozvoljene operacije u programu.

  NaN omogućuje nastavak izvršavanja programa u slučaju pojave neke od prethodnih grešaka programu.

Klasa podataka	Znak	Uvećani eksponent	Implicitni bit	Frakcija
Nula	±	e_{min}-1	0	0
Denormalizovani brojevi	±	e_{min}-1††	0	≠0
Normalizovani brojevi	±	$e_{min} \leq e \leq e_{max}$	1	proizvoljno
Beskonačno	±	$e_{max}+1$	xxx	0
Tihi NaN	±	$e_{max}+1$	xxx	$f_0=1$, f_r=proizvoljno
Signalni NaN	±	$e_{max}+1$	xxx	$f_0=0$, $f_r \neq 0$

xxx	Ne primenjuje se
††	U aritmetičkim operacijama se dodaje 1
e_{min}-1	Sadržaj polja za eksponent su sve nule
$e_{max}+1$	Sadržaj polja za eksponent su sve jedinice
NaN	Not-a-number
f_0	Krajnje levi bit frakcije
f_r	Ostali bitovi frakcije

Tabela 4: Klase podataka u IEEE 754

Zaokruživanje u IEEE 754

Zaokruživanje se vrši svaki put kada rezultat operacije nije tačan. Postoje četiri moguća načina zaokruživanja:

1. **Zaokruživanje na najbližu vrednost** uz zaokruživanje na parnu cifru kada je medjurezultat na sredini (predefinisano)
2. **Zaokruživanje prema $+\infty$**:
   - Ako je broj pozitivan i ako postoji bar jedna jedinica na nekoj poziciji desno od poslednje pozicije koja se čuva u zapisu, na tom mestu se dodaje jedinica.
   - Bez obzira na znak broja, odbace se svi bitovi desno od poslednje pozicije koja se čuva u zapisu.
3. **Zaokruživanje prema $-\infty$**:
   - Ako je broj negativan i ako postoji bar jedna jedinica na nekoj poziciji desno od poslednje pozicije koja se čuva u zapisu, na tom mestu se oduzima jedinica.
   - Bez obzira na znak broja, odbace se svi bitovi desno od poslednje pozicije koja se čuva u zapisu.
4. **Zaokruživanje ka 0**

Specijalne vrednosti u aritmetičkim operacijama

Ako podatak predstavlja normalizovani broj, denormalizovani broj ili nulu, rezultat se dobija u skladu sa uobičajenim načinom izračunavanja.

Rezultat operacije koja ima beskonačno kao operand: zameni se ∞ sa konačnim brojem x i odredi granica kada $x \to \infty$.

Na primer, $45/+\infty = 0$ jer $\lim_{x \to +\infty} 45/x = 0$.

Ako granica kada $x \to \pm\infty$ ne postoji, tada je rezultat operacije `NaN`. Važi

$$-\infty < \text{bilo koji konačan broj} < +\infty$$

Kada operacija uključuje ∞ i rezultat nije QNaN, $\longrightarrow$ rezultat je ∞.

45	+	(+∞)	=	+∞
45	+	(-∞)	=	-∞
45	-	(+∞)	=	-∞
45	-	(-∞)	=	+∞
(+∞)	+	(+∞)	=	+∞
(+∞)	-	(-∞)	=	+∞
(-∞)	+	(-∞)	=	-∞
(-∞)	-	(+∞)	=	-∞

Operacija	`QNaN` formiran pomoću
Sabiranje, oduzimanje	$(\pm\infty) \pm (\pm\infty)$
Množenje	$0 \times \infty$
Deljenje	$0/0, \quad \infty / \infty$
Ostatak	x REM $0, \ \infty$ REM y
Kvadratni koren	$\sqrt{x}$ kada je x<0
Svaka operacija	argument operacije = `SNaN`

Tabela 5: Operacije koje proizvode Tihi NaN

Sabiranje i oduzimanje

Neka $x = x_s \times 2^{x_e}, y = y_s \times 2^{y_e}$. Tada važi

$$x \pm y = \begin{cases} (x_s \times 2^{x_e - y_e} \pm y_s) \times 2^{y_e} & \text{ako } x_e \leq y_e \\ (x_s \pm y_s \times 2^{y_e - x_e}) \times 2^{x_e} & \text{ako } y_e < x_e \end{cases}$$

Pri izvodjenju operacija se vodi računa o specijalnim vrednostima. c Osnovni koraci u algoritmu su

1. Provera postojanja specijalnih vrednosti.
2. Oduzimanje $x - y$ se se realizuje kao $x + (-y)$ uz prethodnu promenu znaka argumenta y.
3. Ukoliko je jedan od sabiraka jednak nuli, vrednost drugog sabirka je rezultat sabiranja.
4. Sabirci se svode na jednake eksponente.
5. Saberu se frakcije sabiraka pri čemu se uzimaju u obzir njihovi znaci. Sabiranje se vrši po pravilima za sabiranje celih brojeva u zapisu znak i apsolutna vrednost. Ukoliko je dobijeni rezultat nula tada je ukupan zbir nula. Ako je pak pri sabiranju došlo do prekoračenja, dobijeni rezultat se pomera za jedno mesto udesno uz povećanje vrednosti eksponenta za jedan. Ako ovo povećanje vrednosti eksponenta dovede do prekoračenja, ukupan rezultat sabiranja je $+\infty$ ili -∞ u zavisnosti od znaka broja.
6. Ako je rezultat sabiranja frakcija normalizovan zaokružuje se i fromira zbir kombinacijom zaokružene frakcije i eksponenta. Ako dobijeni rezultat nije normalizovan, pokušava se njegova normalizacija.

Množenje i deljenje

Neka $x = x_s \times 2^{x_e}, y = y_s \times 2^{y_e}$. Tada važi:

$$x * y = (x_s * y_s) \times 2^{x_e + y_e}$$

$$x / y = (x_s / y_s) \times 2^{x_e - y_e}$$

Pri izvodjenju operacija se vodi računa o specijalnim vrednostima. Osnovni koraci u algoritmu za množenje su

1. Provera postojanja specijalnih vrednosti.
2. Ukoliko je bar jedan od činilaca jednak nuli, rezultat je 0.
3. 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 došlo do potkoračenja vrednosti eksponenta krajnji rezultat je pozitivna ili negativna (u zavisnosti od znaka brojeva x i y) nula.
4. Pomnože se frakcije brojeva. Množenje se vrši prema pravilima za množenje celih brojeva zapisanih pomoću znaka i apsolutne vrednosti.
5. Dobijeni rezultat se normalizuje sličnim postupkom kao kod sabiranja.
6. Broj (binarnih) cifara proizvodu je dvostruko veći od broja cifara vrednost koje su umnožene; cifre koje su višak se odbacuju u procesu zaokruživanja.

Osnovni koraci u algoritmu za deljenje su:

1. Provera postojanja specijalnih vrednosti.
2. Ako je delilac nula tada
   - Ako je deljenik $\neq 0$ količnik je $\pm\infty$ u zavisnosti od znaka x.
   - Ako je deljenik $=0$ tada je rezultat `NaN`.
3. 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.
4. Podele se frakcije brojeva. Deljenje se vrši prema pravilima za deljenje celih brojeva zapisanih pomoću znaka i apsolutne vrednosti.
5. Dobijeni rezultat se normalizuje sličnim postupkom kao kod sabiranja.
6. Dobijeni količnik se zaokružuje prema pravilima za zaokruživanje.

Izuzeta stanja, zastavice i zamke

Kada dodje do izuzetog stanja (kao npr. deljenja sa nulom ili prekoračenja) predefinisana akcija po IEEE standardu je predavanje rezultata i nastavak sa radom.

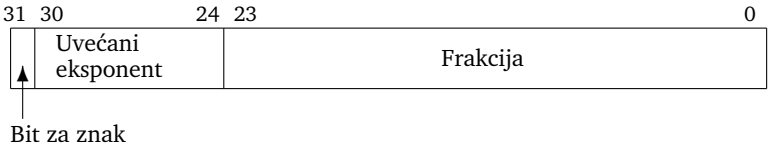
Po IEEE standardu postoji: prekoračenje, potkoračenje, deljenje sa nulom, pogrešna operacija i netačnost. Za svako od ovih izuzeća se postavlja posebna zastavica (eng. *flag*) koju korisnik može programski da ispituje.

IEEE 754 omogućuje rad sa zamkama. Npr.

```
do    uslov
until (x >= 100)
```

Ne bi korektno radio jer poredjenje sa `NaN` uvek vraća netačno pa bi postojala opasnost ulaska u beskonačni ciklus.

Zapis brojeva u pokretnom zarezu pomoću heksadekadne osnove (IBM S/360, S/370, S390, z-series)



Slika 8: Format zapisa realnog broja pomoću heksadekadne osnove

Važi

- Osnova $\beta=16$
- Eksponent se zapisuje u 7 bita na pozicijama 24–30 uz uvećanje od 64. uz uvećanje za 64.
- Frakcija je normalizovana i zapisuje se sa 6 heksadekadnih cifara u 24 bita.

Za eksponent e važi

$$-2^6 \leq e \leq 2^6 - 1$$

Za frakciju s važi

$$16^{-1} \leq |s| \leq 1 - 16^{-6}$$

Interval kome pripada realan broj x koji može da se zapiše uz korišćenje 32 bita je

$$16^{-1} * 16^{-64} \leq |x| \leq (1 - 16^{-6}) * 16^{+63}, \quad \text{odnosno}$$

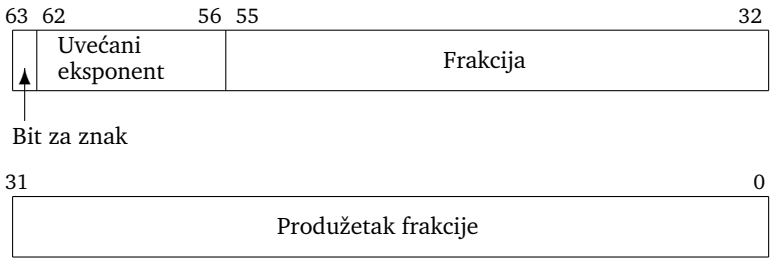
$$5.5 * 10^{-79} < |x| < 7.2 * 10^{+75}$$

Najmanji denormalizovani broj koji može da se zapiše je $16^{-6} * 16^{-64} = 16^{-70} \approx 5.2 * 10^{-85}$

		Znak	Eksponent	Frakcija
$+15$	=	0	1000001	11110000000000000000000
-15	=	1	1000001	11110000000000000000000
$+1/64$	=	0	0111111	01000000000000000000000
0	=	0	0000000	00000000000000000000000
$+16^{-1} \times 16^{-64}$	=	0	0000000	00010000000000000000000
$(1-16^{-6}) \times 16^{+63}$	=	0	1111111	11111111111111111111111
$+16^{-6} \times 16^{-64}$	=	0	0000000	00000000000000000000001

Provera vrednosti zapisa broja +15:
– Znak = +
– Eksponent = 1 (=65 - 64)
– Frakcija = $(0.F)_{16}$ = F$\times 16^{-1}$
– Vrednost = Znak frakcija * $16^{eksponent}$ = +F$\times 16^{-1}$ * 16^{1} = $+F_{16}$ = $+15_{10}$

Tabela 6: Zapis realnih brojeva u jednostrukoj tačnosti / heksadekadna osnova



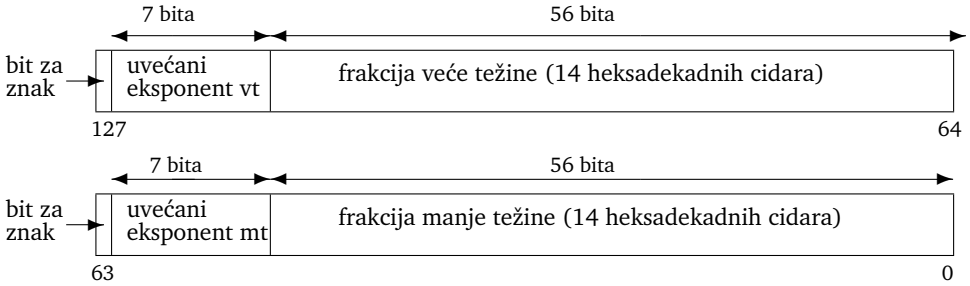
Slika 9: Format zapisa realnog broja u dvostrukoj tačnosti pomoću heksadekadne osnove

Interval realnih brojeva koji mogu da se predstave u dvostrukoj tačnosti je

$$16^{-1} * 16^{-64} \quad \leq |x| \leq \quad (1 - 16^{-14}) * 16^{+63}, \quad \text{odnosno}$$

$$5.5 * 10^{-79} \quad < |x| < \quad 7.2 * 10^{+75}$$

Najmanji denormalizovani broj koji može da se zapiše je $16^{-78} \approx 1.2 * 10^{-94}$



Slika 10: Format zapisa realnog broja u dvostrukoj proširenoj (četvorostrukoj) tačnosti pomoću heksadekadne osnove

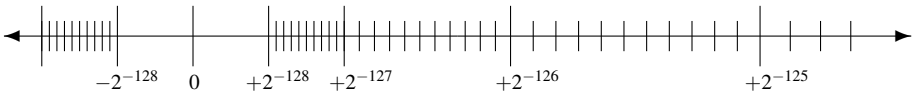
Interval realnih brojeva koji mogu da se predstave u dvostrukoj proširenoj (četvorostrukoj) tačnosti je

$$16^{-1} * 16^{-64} \quad \leq |x| \leq \quad (1 - 16^{-28}) * 16^{+63}, \quad \text{odnosno}$$

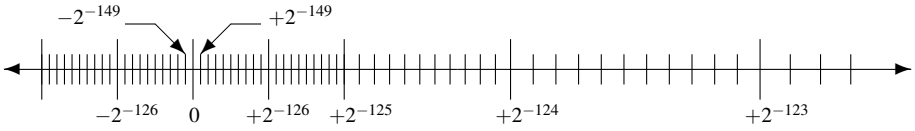
$$5.5 * 10^{-79} \quad < |x| < \quad 7.2 * 10^{+75}$$

Najmanji denormalizovani broj koji može da se zapiše je $16^{-92} \approx 1.7 * 10^{-111}$

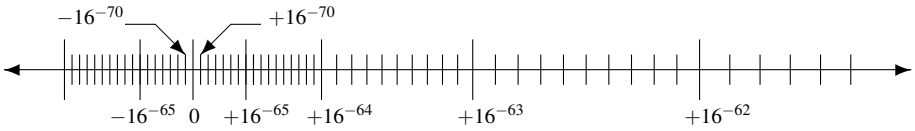
Bez obzira na izabrani način zapisa realnih brojeva, količina realnih brojeva koji mogu da se zapišu je ista, ali je gustina na pojedinim delovima brojčane ose različita



(a) Zapis pomoću binarne osnove, pre-IEEE754 format



(b) Zapis pomoću binarne osnove - IEEE 754 format sa uključenim denormalizovanim vrednostima



(c) Zapis pomoću heksadekadne osnove sa uključenim denormalizovanim vrednostima

Slika 11: Gustina zapisa realnih brojeva (za jednostruku tačnost)