Realni brojevi u pokretnom zarezu zapisani pomoću dekadne osnove

IEEE 754-2008

- Standard IEEE 754-2008 usvojen Avgusta 2008. godine
- Zamenjuje prethodne verzije IEEE 754 i IEEE 854 standarda
- Dorade prethodnog standarda vezane za binarnu osnovu
- Uvodi se mogućnost zapisa brojeva pomoću dekadne osnove

Zašto brojevi u pokretnom zarezu?

Svakodnevne aplikacije su takve da ne mogu da budu u potpunosti podržane brojevima u fiksnom zarezu. Na primer,

- Rad sa operacijama deljenja i stepenovanja
- Izračunavanje kamata
- Izračunavanje cene telefonskom impulsa po iskorišćenim sekundama
- Finansijske analize, statistike, porezi, konverzija valuta, osiguranje, ...

Zašto dekadna osnova?

Zašto nije pogodno koristiti binarnu osnovu?

- U I generaciji računara konstruisani su računari koji su čuvali brojeve i u binarnoj i u dekadnoj osnovi
- U 1950-tim godinama realni brojevi u pokretnom zarezu su bili mnogo pogodniji za upotrebu jer su zahtevali manje memorijskog prostora i manje komponenti za konstrukciju
- Jednostavnije (brže) izvodjenje operacija sa binarnim brojevima u odnosu na operacije sa dekadnim brojevima

... i problemi

Binarna frakcija ne može tačno da prikaže sve dekadne frakcije.

- Na primer, 0.1 se predstavlja kao beskonačan periodičan binaran broj 0.000110011001100...
- 1.2x1.2 u računaru ne mora uvek da proizvede rezultat 1.44. Jer,
  - 1.2 zapisan u pokretnom zarezu sa zapisan u binarnoj osnovi sa jednostrukom tačnošću ima vrednost 1.2000000476837158203125
  - Kada se ova vrednost kvadrira dobije se 1.440000057220458984375
- Javljaju se problemi u realnim aplikacijama. Na primer, neka na cenu telefonskog impulsa treba dodati 5% poreza. Ako je potrošeno 0.70 dinara, iznos sa porezom, zapisan u binarnoj osnovi sa dvostrukom tačnošću, biće jednak 0.73499999999999998667732370449812151491641998291015625 (treba da bude 0.735) odnosno, kada se zaokruži na dve decimale daće 0.73 a ne 0.74

Zašto dekadna osnova?

- Ljudi računaju u dekadnom sistemu
- Finansijske i komercijalne aplikacije, kao i aplikacije koje su orijentisane prema korisnicima (tzv. *human-centric*) imaju legitimne zahteve za aritmetikom sa brojevima u dekadnom sistemu
- Oko 55% numeričkih podataka u bazama podataka su dekadni podaci, a od preostalih još 43% su celi brojevi
- Tradicionalan način rada sa celobrojnom aritmetikom i ručnim skaliranjem nije pogodan za korišćenje i podložan je greškama

Predstavljanja brojeva u dekadnoj osnovi

1. Softverski - softver se stara o zapisu i operacijama sa takvim brojevima. Brizina rada nije zadovoljavajuća

2. Hardverski

   - Pomoću BCD zapisa.
     - fiksni zarez,
     - relativno složeno izvodjenje operacija
     - Znatno sporije od binarne osnove.
   - Pomoću zapisa realnih brojeva u pokretnom zarezu pomoću dekadne osnove (IEEE 754-2008).
     - Trenutno samo neki od procesora hardverski podržavaju ovakav zapis (IBM *z series*, Power, ...)

Zapis u registrima računara

Zahtevi

- Realne brojeve u dekadnoj osnovi u računaru treba zapisati u registrima računara, tj. u prostoru iste veličine kao i pri zapisu realnih brojeva u pokretnom zarezu pomoću binarne osnove.
- Zapis treba da bude u obliku znak, frakcija i eksponent.
- Zapis treba da bude moguć u jednostrukoj, dvostrukoj i četvorostrukoj tačnosti
- Zapis treba da bude takav omogući tačno zapisivanje dekadnih cifara bez konverzije
- Preciznost zapisa treba da bude uporediva sa preciznošću zapisa dekadnih brojeva pomoću binarne osnove

...i problemi pri tom zapisu

- Nedovoljan prostor (npr. u registru veličine 32 bita) za smeštanje svih potrebnih komponenti ako se kodiraju pomoću BCD koda
- Potrebno je primeniti kodiranje koje omogućuje zapis svih potrebnih podataka u dovoljnoj tačnosti
- Zapis treba da bude moguć u jednostrukoj, dvostrukoj i četvorostrukoj tačnosti
- Treba predvideti zapis specijalnih vrednosti
- Za takav zapis treba definisati pravila za izvodjenje operacija

Chen-Ho kodiranje

- Varijanta Hofmanovog kodiranja
- Kodira tri dekadne cifre u 10 bita
- 20% efikasnije kodiranje od BCD zapisa
- Deli cifre na male (0-7) i velike (9,9) na osnovu vrednosti prvog bita u BCD kodu
- Male cifre zahtevaju tri bita a velike jedan bit da bi se medjusobno razlikovale
- Kodiranje razmatra svaku od kombinacija tri cifre
  - Sve tri cifre su male: potrebno je 10 bitova (3+3+3 za cifre, 1 bit da označi ovu kombinaciju)
  - Dve cifre su male: potrebno je 7 bitova za cifre (3+3+1); preostala 3 bita označavaju kombinaciju
  - Jedna cifra je mala: potrebno je 5 bitova za cifre (3+1+1); preostalih 5 bitova označavaju kombinaciju
  - Sve cifre su velike: potrebno je 3 bita za cifre (1+1+1); preostalih 7 (potrebno je samo 5) označavaju kombinaciju

Shematski prikaz kodiranja

Cifre $C_1C_2C_3$ zapisane u BCD zapisu (abcd)(efgh)(ijkl) postaju (p)(qrs)(tuv)(wxy) gde je

aei	p	qrs	tuv	wxy
000	0	bcd	fgh	jkl
100	1	00d	fgh	jkl
010	1	01d	bch	jkl
001	1	10d	fgh	bcl
011	1	11d	00h	bcl
101	1	11d	01h	fgl
110	1	11d	10h	jkl
111	1	11d	11h	00l

Shematski prikaz dekodiranja

Cifre (p)(qrs)(tuv)(wxy) postaju (abcd)(efgh)(ijkl) koje predstavljaju BCD zapis cifara $C_1C_2C_3$

pqrtu	abcd	efgh	ijkl
0....	0qrs	0tuv	0wxy
100..	100s	0tuv	0wxy
101..	0tus	100v	0wxy
110..	0wxs	0tuv	100y
11100	0wxs	100v	100y
11101	100s	0wxv	100y
11110	100s	100v	0wxy
11111	100s	100v	100y

gde tačka ('.') označava da je sadržaj na toj poziciji nebitan.

Softverska realizacija kodiranja

Softverski, kodiranje može da se realizuje preko sledećih logičkih jednakosti:

$$
\begin{aligned}
p &= a \vee e \vee i \\
q &= (b \wedge \neg\, e) \vee i \vee (a \wedge e) \\
r &= (c \wedge \neg i) \vee e \vee (a \wedge i) \\
s &= d \\
t &= (a \wedge e) \vee (f \wedge (\neg a \wedge \neg i)) \vee (b \wedge e \wedge \neg i) \\
u &= (a \wedge i) \vee (c \wedge e \wedge \neg i) \vee (g \wedge \neg e) \\
v &= h \\
w &= j \vee (b \wedge i) \vee (f \wedge a \wedge i) \\
x &= k \vee (c \wedge i) \vee (g \wedge a \wedge i) \\
y &= l
\end{aligned}
$$

Softverska realizacija dekodiranja

Softverski, dekodiranje može da se realizuje preko sledećih logičkih jednakosti:

$$
\begin{aligned}
a &= (p \wedge \neg q \wedge \neg r) \vee (p \wedge q \wedge r \wedge (t \vee u)) \\
b &= (q \wedge \neg p) \vee (t \wedge p \wedge \neg q \wedge r) \vee (w \wedge q \wedge (\neg r \vee (\neg t \wedge \neg u))) \\
c &= (r \wedge (\neg p \vee (\neg q \wedge u))) \vee (x \wedge p \wedge q \wedge (\neg r \vee (\neg t \wedge \neg u))) \\
d &= s \\
e &= (p \wedge r \wedge (\neg q \vee \neg u \vee t)) \\
f &= (t \wedge (\neg p \vee \neg r)) \vee (p \wedge q \wedge r \wedge \neg t \wedge u \wedge w) \\
g &= (u \wedge (\neg p \vee \neg r)) \vee (p \wedge q \wedge r \wedge \neg t \wedge u \wedge x) \\
h &= v \\
i &= (p \wedge q \wedge (\neg r \vee \neg t \vee u)) \\
j &= (w \wedge (\neg p \vee \neg q \vee (r \wedge t))) \\
k &= (x \wedge (\neg p \vee \neg q \vee (r \wedge t))) \\
l &= y
\end{aligned}
$$

Densely Packed Decimal kodiranje

- Slično Chen-Ho kodiranju, ali umesto Hofmanovog koda koristi drugačije preuredjenje bitova
- Kodira tri dekadne cifre u 10 bita
- 20% efikasnije kodiranje od BCD zapisa
- Deli cifre na male (0-7) i velike (9,9) na osnovu vrednosti prvog bita u BCD kodu
- Male cifre zahtevaju tri bita a velike jedan bit da bi se medjusobno razlikovale
- Kodiranje razmatra svaku od kombinacija tri cifre
  - Sve tri cifre su male: potrebno je 10 bitova (3+3+3 za cifre, 1 bit da označi ovu kombinaciju)
  - Dve cifre su male: potrebno je 7 bitova za cifre (3+3+1); preostala 3 bita označavaju kombinaciju
  - Jedna cifra je mala: potrebno je 5 bitova za cifre (3+1+1); preostalih 5 bitova označavaju kombinaciju
  - Sve cifre su velike: potrebno je 3 bita za cifre (1+1+1); preostalih 7 (potrebno je samo 5) označavaju kombinaciju

Shematski prikaz kodiranja

Cifre $C_1C_2C_3$ zapisane u BCD zapisu (abcd)(efgh)(ijkm) postaju (pqr)(stu)(v)(wxy) gde je

aei	pqr stu v wxy	Komentar
000	bcd fgh 0 jkm	Sve cifre su male
001	bcd fgh 1 00m	Krajnje desna cifra je velika
010	bcd jkh 1 01m	Srednja cifra je velika
100	jkd fgh 1 10m	Krajnje leva cifra je velika
110	jkd 00h 1 11m	Krajnje desna cifa je mala (ostale dve su velike)
101	fgd 01h 1 11m	Srednja cifra je mala (ostale dve su velike)
011	bcd 10h 1 11m	Krajnje leva cifra je mala (ostale dve su velike)
111	00d 11h 1 11m	Sve cifre su velike; dva bita se ne koriste

Shematski prikaz dekodiranja

Cifre ((pqr)(stu)(v)(wxy) postaju (abcd)(efgh)(ijkm) koje predstavljaju BCD zapis cifara $C_1C_2C_3$

vwxst	abcd efgh ijkm
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

gde tačka ('.') označava da je sadržaj na toj poziciji nebitan.

Softverska realizacija kodiranja

Softverski, kodiranje može da se realizuje preko sledećih logičkih jednakosti:

$$
\begin{aligned}
p &= b \vee (a \wedge j) \vee (a \wedge f \wedge i) \\
q &= c \vee (a \wedge k) \vee (a \wedge g \wedge i) \\
r &= d \\
s &= (f \wedge (\neg a \vee \neg i)) \vee (\neg a \wedge e \wedge j) \vee (e \wedge i) \\
t &= g \vee (\neg a \wedge e \wedge k) \vee (a \wedge i) \\
u &= h \\
v &= a \vee e \vee i \\
w &= a \vee (e \wedge i) \vee (\neg e \wedge j) \\
x &= e \vee (a \wedge i) \vee (\neg a \wedge k) \\
y &= m
\end{aligned}
$$

Softverska realizacija dekodiranja

Softverski, dekodiranje može da se realizuje preko sledećih logičkih jednakosti:

$$
\begin{array}{lcl}
a & = & (v \land w) \land (\neg s \lor t \lor \neg x) \\
b & = & p \land (\neg v \lor \neg w \lor (s \land \neg t \land x)) \\
c & = & q \land (\neg v \lor \neg w \lor (s \land \neg t \land x)) \\
d & = & r \\
e & = & v \land ((\neg w \land x) \lor (\neg t \land x) \lor (s \land x)) \\
f & = & (s \land (\neg v \lor \neg x)) \lor (p \land \neg s \land t \land v \land w \land x) \\
g & = & (t \land (\neg v \lor \neg x)) \lor (q \land \neg s \land t \land w) \\
h & = & u \\
i & = & v \land ((\neg w \land \neg x) \lor (w \land x \land (s \lor t))) \\
j & = & (\neg v \land w) \lor (s \land v \land \neg w \land x) \lor (p \land w \land (\neg x \lor (\neg s \land \neg t))) \\
k & = & (\neg v \land x) \lor (t \land \neg w \land x) \lor (q \land v \land w \land (\neg x \lor (\neg s \land \neg t))) \\
m & = & y
\end{array}
$$

Prednosti DPD u odnosu na Chen-Ho kodiranje

- Kompresija jedne ili dve dekadne cifre (u optimalno 4 ili 7 bitova respektivno) se vrši kao podskup kodiranja 3-cifrenog dekadnog broja. Posledica ovoga je da se proizvoljan broj dekadnih cifara kodira efikasnije. Na primer, 38 cifara se može kodirati pomoću 127 bita, a 71 dekadna cifra u 237 bita. Kao kontrast, Chen-Ho kodiranje uvek koristi 10 bita za kodiranje 3 dekadne cifre, tako da je broj dekadnih cifara koje se kodiraju uvek umnožak od 3.

- Kodiranje jedne ili dve dekadne cifre je uvek desno poravnato u grupi od 10 bita, dok su ostali bitovi 0. Ovo omogućuje da kodirane dekadne cifre mogu da se zapišu polju veće dužine dopunjavanjem nula sa leve strane, bez potrebe za ponovnim kodiranjem.Kao kontrast, dve dekadne cifre kodirane Chen-Ho algoritmom moraju da budu ponovno kodirane ukoliko treba da budu zapisane u polju veće dužine.

- Pozicija i izbor indikator bita dopuštaju da se svi jednocifreni brojevi (preciznije svi brojevi u intervalu [0,79] kodiraju desno poravnato na isti način kao u BCD (8421) kodu, što olakšava konverziju takvih brojeva. (Chen-Ho kodiranje na ovaj način preslikava brojeve u intervalu [0,7]).

Primer poredjenja Chen-Ho i DPD kodiranja

Cifre	BCD	Chen-Ho	Densely Packed
005	0000 0000 0101	000 000 0101	000 000 0101
009	0000 0000 1001	110 000 0000	000 000 1001
055	0000 0101 0101	000 010 1101	000 101 0101
099	0000 1001 1001	111 000 1001	000 101 1111
555	0101 0101 0101	010 110 1101	101 101 0101
999	1001 1001 1001	111 111 1001	001 111 1111

Model aritmetike u zapisu pomoću dekadne osnove

Komponente modela

- Brojevi. Predstavljaju vrednosti koje mogu da se obradjuju ili da budu rezultati operacija
- Operacije. Osnovne operacije (sabiranje, oduzimanje, množenje, deljenje, ...) koje mogu da se izvode nad brojevima
- Kontekst. Predstavlja okruženje (zavisno od parametara korisnika) u kome se odredjuju rezultati aritmetičkih operacija (npr. preciznost koja se koristi, način zaokruživanja, itd.).

Brojevi

Brojevi mogu biti *konačni brojevi* (tj. brojevi čije vrednosti mogu da budu tačno prikazane) ili *specijalne vrednosti*.

Brojevi su zapisani pomoću

- Znaka broja (0 za pozitivne, 1 za negativne)
- Frakcije. Frakcija se zapisuje kao celobrojna vrednost koja je veća ili jednaka nuli.
- Eksponenta.

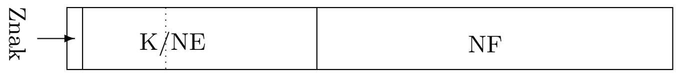
U zavisnosti od komponenti, jedna ista vrednost može biti zapisana na različite načine. Skup zapisa (istog) realnog broja u pokretnom zarezu se naziva *kohorta*. Članovi kohorte su različite reprezentacije istog realnog broja u pokretnom zarezu. Na primer, ako je s znak broja, c umnožak od 10 i q je manje od najveće (predstavljive) vrednosti eksponenta, tada su (s, q, c) i $(s, q + 1, c \div 10)$ su dve reprezentacije istog realnog broja koje pripadaju jednoj kohorti.

Kako broj može biti predstavljen pomoću različitih elemenata kohorte, njegova reprezentacija ne mora da bude normalizovana.

I u slučaju dekadnih brojeva važi da postoje iste klase brojeva kao i za binarnu osnovu:

- Označena nula
- Subnormalni brojevi (u prethodnoj verziji standarda *denormalizovani brojevi*)
- Normalni (u prethodnoj verziji standarda *normalizovani brojevi*brojevi)
- Beskonačno
- NaN (tihi i signalni)

Formati zapisa



- Znak (1 bit): 0 za pozitivne i 1 za negativne
- K/NE - Kombinacija/eksponent (w+5 bitova): za konačne brojeve sadrži uvećani eksponent i cifre najveće težine frakcije; za NaN i beskonačne vrednosti polje sadrži kodove koji ih identifikuju.
  - Prvih 5 bitova označava da li je broj konačan ili je u pitanju neka specijalna vrednost. Ako je broj konačan u ovih 5 bita su kodirane cifra nejaveće tečine frakcije i eksponenta.
  - Ostalih w bitova označava ili nastavak eksponenta ili se koristi za bližu karakterizaciju specijalne vrednosti. Uvećani eksponent E je širine w+2 bita, pri čemu je vrednost prva dva bita uvećanog eksponenta uzeta zajedno 0, 1 ili 2.
- NF - Nastavak frakcije (t bita, pri čemu je $t = J \times 10$): kada se ovo polje kombinuje sa vodećim ciframa frakcije koje se nalaze u kombinaciji, format zapisuje ukupno $p = 3 \times J + 1$ dekadnu cifru.

Veličine w, t(J) i uvećanje zavise od toga da li se broj zapisuje u jednostrukoj, dvostrukoj ili četvorostrukoj tačnosti.

Naziv formata	parametar	decimal32	decimal64	decimal128
Širina polja (bita)		32	64	128
Kombinacija i nastavak eksponenta	w+5	11	13	17
K-Kombinacija		5	5	5
NE-Nastavak eksponenta	w	6	8	12
Nastavak frakcije	t	20	50	110
Uvećanje		101	398	6176

Vrednost broja predstavljenog u pokretnom zarezu pomoću dekadne osnove se odredjuje na sledeći način:

1. Ako su 5 bitova kombinacije u intervalu 00000-11101, tada je u pitanju konačan broj. Tih 5 cifara kodiraju dva bita najveće težine uvećanog eksponenta i cifru najveće težine frakcije. Ostatak od w bita u polju NE predstavlja nastavak eksponenta.

2. Ako su 5 bitova kombinacije jednaki 11110, vrednost broja je $\pm\infty$, u zavisnosti od znaka broja. Ostalih w bitova u polju NE i nastavak frakcije nisu relevantni (bitovi u NE polju mogu da se koriste radi bliže specifikacije beskonačnosti u operacijama, ali nemaju uticaja na odredjivanje vrednosti broja koja je jednaka beskonačno).

3. Ako su 5 bitova kombinacije jednaki 11111, vrednost broja je NaN. Ako je bit najveće težine polja NE jednak 1 tada je vrednost SNaN; u suprotnom je QNaN. Ostalih w-1 bitova polja NE i nastavak frakcije se koriste radi bliže specifikacije NaN vrednosti.

Formati kodiranja

Reprezentacija konačnog broja r je jednaka trojci (znak, E-uvećanje,C), a vrednost je jednaka $v = (-1)^{znak} \times C \times 10^{E-\text{uvećanje}}$, gde je C dobijeno dopisivanjem cifara frakcije iz polja za kombinaciju (kao cifre veće težine) na ostatak frakcije, dok se vrednost eksponenta odredjuje dekodiranjem sadržaja polja za kombinaciju i nastavak eksponenta.

Eksponent se kodira kao ceo broj sa uvećanjem, dok način kodiranja (i dekodiranja) frakcije zavisi od toga da li se koristi dekadno ili binarno kodiranje.

- Dekadno kodiranje, Densely Packed Decimal (DPD) algoritmom ravijenim od strane IBM-a
- Binarno kodiranje, Binary Integer Decimal (BID) algoritmom razvijenim od strane INTEL-a
- Postoje razlike u veličini intervala što otežava punu portabilnost programa.

Dekadno kodiranje

- Koristi se u hardverskim i softverskim implementacijama IBM-a.
- Dekadne cifre u nastavku frakcije su kodirane DPD algoritmom
- Dve cifre najveće težine eksponenta i cifra najveće težine frakcije se odredjuju na sledeći način:
  - Ako su 5 bitova kombinacije u intervalu 0xxxx-10xxx, tada su dva bita najveće težine kombinacije dva bita najveće težine eksponenta, a vrednost cifre najveće težine frakcije je vrednost dekadne cifre zapisane pomoću tri cifre najmanje težine kombinacije.
  - Ako su 5 bitova kombinacije u intervalu 110xx-1110xx, predstavimo bitove u kombinaciji kao $11e_1e_2f$. Tada je dekadna vrednost cifre najveće težine frakcije jednaka 8+f, dok su e_1e_2 cifre najveće težine eksponenta.

Binarno kodiranje

- Koristi se u softverskim implementacijama. Za sada ne postoji hardverska implementacija koja koristi ovaj način kodiranja.
- Dekadne cifre u nastavku frakcije su zapisane kao celobrojne vrednosti u binarnoj osnovi
- Dve cifre najveće težine eksponenta i cifra najveće težine frakcije se odredjuju na sledeći način:
  - Ako su 5 bitova kombinacije u intervalu 0xxxx-10xxx, tada se eksponent formira od w+2 bita najveće težine para K/NE (posmatranih kao jedna celina), a frakcija od preosta 3 bita najmanje težine NE i nastavka frakcije (opet posmatrane kao jedna celina).
  - Ako su 5 bitova kombinacije u intervalu 110xx-1110xx, tada se uvećani eksponent formira od narednih w bitova para K/NE. Frakcija se dobija dopisivanjem četiri bita najveće težine koji se dobijaju kao zbir 8 (1000)+ vrednost bita najmanje težine NE.

Intervali brojeva

DPD			
Naziv formata	decimal32	decimal64	decimal128
Širina polja (bita)	32	64	128
Polje za kombinaciju	5	5	5
Nastavak eksponenta	6	8	12
Nastavak frakcije	20	50	110
Uvećanje	101	398	6176
Interval eksponenta	[-101,+90]	[-389,+369]	[-6176,+6111]
Preciznost	7	16	34
BID			
Naziv formata	decimal32	decimal64	decimal128
Širina polja (bita)	32	64	128
Eksponenta	8	10	14
Frakcija	23	53	113
Uvećanje	101	398	6176
Interval eksponenta	[-95,+96]	[-383,+384]	[-6143,+6144]
Preciznost	7	16	34

Način izračunavanja uvećanja eksponenta

Eksponent je označeni ceo broj koji predstavlja stepen broja 10 kojim se množi frakcija. Neka $E_{granica}$ označava gornju granicu apsolutne vrednosti eksponenta. *Prilagodjen eksponent* je vrednost eksponenta koja odgovara 'normalizovanoj' vrednosti frakcije

Neka je *duzinafm* maksimalan broj dekadnih cifara koji može biti zapisan u frakciji. Tada:

- Potrebno je da gornja granica apsolutne vrednosti eksponenta E_{gmax} bude veća od $5 \times duzinafm$. Preporuka je da važi $E_{gmax} > 10 \times duzinafm$.

- Vrednost prilagodjenog eksponenta je jednaka $eksponent + (duzinaft - 1)$ gde je $duzinaft$ (trenutni) broj dekadnih cifara predstavljen frakcijom.

- Najveća i najamanja vrednost prilagodjenog eksponenta moraju da zadovoljavaju uslov $E_{min} = 1 - Emax$. Odavde, eksponent pripada intervalu $[-E_{granica} - (duzinaft - 1) + 1, E_{granica} - (duzinaft - 1)]$.

Primer: U slučaju zapisa broja u jednostrukoj tačnosti, frakcija ima 7 cifara. Tada je $(duzinaft - 1) = 6$, a broj različitih vrednosti eksponenata koje mogu da se zapiše u 8 binarnih cifara je $(10111111)_2$, tj $(191)_{10}$ (nije dozvoljena kombinacija koja počinje sa 11 jer ona predstavlja beskonačno ili NaN). Kako su ovom vrednošću obuhvaćeni i pozitivni i negativni brojevi, odatle je $E_{granica}$=191/2 +1=96. Odatle je interval kome pripada eksponent [-96-6+1, +96-6], tj. [-101,+90]. Uvećanje je +101 da bi najmanja vrednost uvećanog eksponenta bila kodirana kao sve binarne 0.

Način računanja uvećanja je se može videti na adresi http://speleotrove.com/decimal/dbcalc.html

Primeri zapisa brojeva pomoću dekadnog kodiranja

		Znak	Kombinacija	Nastavak eksponenta	Nastavak frakcije
$+15$	$=$	0	01000	100101	0000 0000 0000 0001 0101
-15	$=$	1	01000	100101	0000 0000 0000 0001 0101
$+15.0$	$=$	0	01000	100100	0000 0000 0000 1101 0000
$+1/64$	$=$	0	01000	011111	0000 0101 0111 0010 0101
$+0$	$=$	0	01000	100101	0000 0000 0000 0000 0000
-0	$=$	1	01000	100101	0000 0000 0000 0000 0000
$+0.0$	$=$	0	01000	100100	0000 0000 0000 0000 0000
$+0.00$	$=$	0	01000	100011	0000 0000 0000 0000 0000
$+(10^7 - 1) \times 10^{+90}$	$=$	0	11101	111111	1111 1111 1111 1111 1111
$+1 \times 10^{-95}$	$=$	0	00000	000110	0000 0000 0000 0000 0001
$+1 \times 10^{-101}$	$=$	0	00000	000000	0000 0000 0000 0000 0001

Provera vrednosti zapisa broja +15:
– Znak = +
– Eksponent = 0 (=101 - 101)
– Frakcija = $(15)_{10}$ = 0000000000 0000010101 (kodirana u dva dekleta)
– Vrednost = Znak frakcija * $10^{eksponent}$ = +15* 10^0 = $+15_{10}$

Uporedni prikaz zapisa nekih brojeva pomoću binarne, heksadekadne i dekadne osnove

Vrednost		S	BE or C	Ostatak frakcije	
1.0	B	0	011 1111 1	000 0000 0000 0000 0000 0000	
	H	0	100 0001	0001 0000 0000 0000 0000 0000	
	D	0	010 0010 0100	0000 0000 0000 0001 0000	[10x10^{-1}]
0.5	B	0	011 1111 0	000 0000 0000 0000 0000 0000	
	H	0	100 0000	1000 0000 0000 0000 0000 0000	
	D	0	010 0010 0100	0000 0000 0000 0000 0101	[5x10^{-1}]
1/64	B	0	011 1100 1	000 0000 0000 0000 0000 0000	
	H	0	011 1111	0100 0000 0000 0000 0000 0000	
	D	0	010 0001 1111	0000 0101 0111 0010 0101	[15625x10^{-6}]
+0	B	0	000 0000 0	000 0000 0000 0000 0000 0000	
	H	0	000 0000	0000 0000 0000 0000 0000 0000	
	D	0	010 0010 0101	0000 0000 0000 0000 0000	[+0x10^{0}]
-0	B	1	000 0000 0	000 0000 0000 0000 0000 0000	
	H	1	000 0000	0000 0000 0000 0000 0000 0000	
	D	1	010 0010 0101	0000 0000 0000 0000 0000	[-0x10^{0}]
-15.0	B	1	100 0001 0	111 0000 0000 0000 0000 0000	
	H	1	100 0001	1111 0000 0000 0000 0000 0000	
	D	1	010 0010 0100	0000 0000 0000 1101 0000	[-150x10^{-1}]
20/7	B	0	100 0000 0	011 0110 1101 1011 0110 1110	
	H	0	100 0001	0010 1101 1011 0110 1101 1011	
	D	0	010 1001 1111	1101 0111 0100 1100 0011	[2857143x10^{-6}]
2^{-126}	B	0	000 0000 1	000 0000 0000 0000 0000 0000	
	H	0	010 0001	0100 0000 0000 0000 0000 0000	
	D	0	000 0111 1001	0011 1101 0110 0101 1010	[1175494x10^{-44}]
2^{-149}	B	0	000 0000 0	000 0000 0000 0000 0000 0001	
	H	0	001 1011	1000 0000 0000 0000 0000 0000	
	D	0	000 0111 0010	1000 0000 0101 0101 1110	[1401298x10^{-51}]
$2^{128} \times F$	B	0	111 1111 0	111 1111 1111 1111 1111 1111	
$F = 1 - 2^{-24}$	H	0	110 0000	1111 1111 1111 1111 1111 1111	
	D	0	100 1100 0101	1000 0000 1001 0010 1101	[3402823x10^{+32}]
2^{-260}	B			Nula (broj je isuviše mali)	
	H	0	000 0000	0001 0000 0000 0000 0000 0000	
	D	0	001 0101 0000	0111 1110 1111 0000 0101	[5397605x10^{-85}]
$2^{248} \times F$	B			Ne može da se predstavi	
$F = 1 - 2^{-24}$	H	0	111 1110	1111 1111 1111 1111 1111 1111	
	D	0	101 0010 1001	1010 1000 1100 1010 1000	[4523128x10^{+68}]

Oznake:

- S - znak broja (frakcije)
- B - Binary Floating Point
- HFP - Hexadecimal Floating Point
- DFP - Decimal Floating Point
- BE ili C Uvećani eksponent BFP ili HFP broja, kombinacije kod DFP broja

Primeri

Primer1: Zapisati broj +123.4 u pokretnom zarezu pomoću dekadne osnove u jednostrukoj tačnosti, koristeći dekadno (DPD) kodiranje.

Rešenje: Broj koga treba zapisati predstavimo u obliku +1234*10^{-1}. Vrednosti cifara u zapisu broja su:

1. Broj koga treba zapisati je pozitivan $\longrightarrow$ znak broja je 0.

2. Frakcija broja je 1234. Dopunimo frakciju do maksimalnog broja cifara za jednostruku tačnost (7). Dobijeni broj koga treba da kodiramo je 0001234. Kodiraju se 6 cifara manje težine u dva dekleta, dok se cifra najveće težine kodira u polju za kombinaciju. Primenom DPD kodiranja dobija se (npr. na osnovu tabele shematskog prikaza kodiranja):

0 0 1	2 3 4	Dekadni zapis
abcd efgh ijkm	abcd efgh ijkm	
0000 0000 0001	0010 0011 0100	BCD zapis
000 000 0 001	010 011 0 100	DPD dekleti
pqr stu v wxy	pqr stu v wxy	

3. Eksponent je jednak -1, i uz uvećanje 101 dobija se 100. Prevod dobijene vrednosti u binarni brojni sistem je 01100100. Odavde je nastavak eksponenta 100100, dok se prva dva bita 01 kodiraju u kombinaciji.

4. Kombinacija treba da kodira cifru 0 (cifra frakcije) i dva bita najveće težine eksponenta (binarne cifre 01). Kako je 0 cifra frakcije, na osnovu pravila za predstavljanje sledi da je kombinacija = 01000.

Odavde, zapis broja +982.5294*10^{42} je 0 01000 100100 0000 0000 0101 0011 0100

Primer 2: Zapisati broj -243/13 u pokretnom zarezu pomoću dekadne osnove u jednostrukoj tačnosti, koristeći dekadno (DPD) kodiranje.

Rešenje: Dobijeni količnik zapišimo pomoću 7 dekadnih cifara u obliku znak, farkcija i eksponent. Kako je 243/13=18.69230769... pretpostavimo da je odabrano takvo pravilo zaokruživanja da je rezultat zaokružen na 7 cifara jednak 18,69230 (primedba: i za zapis pomoću dekadne osnove važe ista pravila zaokruživanja!).

Broj koga treba zapisati predstavimo u obliku -1869230*10^{-5}. Vrednosti cifara u zapisu broja su:

1. Broj koga treba zapisati je negativan $\longrightarrow$ znak broja je 1.
2. Frakcija broja je 1869230. Kodiraju se 6 cifara manje težine u dva dekleta, dok se cifra najveće težine kodira u polju za kombinaciju. Primenom DPD kodiranja dobija se (npr. na osnovu tabele shematskog prikaza kodiranja):

```
  8    6    9         2    3    0            Dekadni zapis
abcd efgh ijkm      abcd efgh ijkm
1000 0110 1001      0010 0011 0000           BCD zapis

110 010 1  111      010 011 0  000           DPD dekleti
pqr stu v  wxy      pqr stu v  wxy
```

3. Eksponent je jednak -5, i uz uvećanje 101 dobija se 96. Prevod dobijene vrednosti u binarni brojni sistem je 01100000. Odavde je nastavak eksponenta 100000, dok se prva dva bita 01 kodiraju u kombinaciji.
4. Kombinacija treba da kodira cifru 1 (cifra frakcije) i dva bita najveće težine eksponenta (binarne cifre 01). Kako je 1 cifra frakcije, na osnovu pravila za predstavljanje sledi da je kombinacija = 01001.

Odavde, zapis broja -243/13 je 1 01001 100000 1100 1011 1101 0011 0000

Primer 3: Zapisati broj +982.5294*10^{42} u pokretnom zarezu pomoću dekadne osnove u jednostrukoj tačnosti, koristeći dekadno (DPD) kodiranje.

Rešenje: Dati broj zapišimo u obliku +9825294*10^{38}. Vrednosti cifara u zapisu broja su:

1. Broj koga treba zapisati je pozitivan $\longrightarrow$ znak broja je 0.
2. Frakcija broja je 9825294. Šest cifara manje težine se koriraju u dva dekleta, dok se cifra najveće težine kodira u polju za kombinaciju. Primenom DPD kodiranja dobija se (npr. na osnovu tabele shematskog prikaza kodiranja):

```
  8    2    5          2    9    4             Dekadni zapis
abcd efgh ijkm       abcd efgh ijkm
1000 0010 0101       0010 1001 0100            BCD zapis

100 010 1  101       010 101 1  010            DPD dekleti
pqr stu v  wxy       pqr stu v  wxy
```

3. Eksponent je jednak +38, i uz uvećanje 101 dobija se 139. Prevod dobijene vrednosti u binarni brojni sistem je 10001011. Odavde je nastavak eksponenta 001011, dok se prva dva bita 10 kodiraju u kombinaciji.
4. Kombinacija treba da kodira cifru 9 (cifra frakcije) i dva bita najveće težine eksponenta (binarne cifre 10). Kako je 9 cifra frakcije, na osnovu pravila za predstavljanje sledi da je kombinacija = 11101.

Odavde, zapis broja -243/13 je 1 11101 001011 1000 1011 0101 0101 1010

Primer 4: Koji dekadni broj je predstavljen brojem u pokretnom zarezu zapisanim u IEEE 754 zapisu pomoću dekadne osnove (DPD kodiranje): 00111101111000000000000000110101

Rešenje: Razdvojimo cifre u zapisu broja da bi se odredile komponente zapisa.
0|01111|011110|0000 0000 0000 0011 0101

Odavde se dobija:

1. Cifra za znak je nula $\longrightarrow$ Broj je pozitivan
2. Prve dve cifre u kombinaciji su 01 $\longrightarrow$ broj je konačan (nije beskonačno ili NaN)
   - Pošto su prve dve cifre kombinacije 01, one su i cifre najveće težine uvećanog eksponenta. Dobijena vrednost uvećanog eksponenta je $(01011110)_2$, odnosno $(94)_{10}$. Odavde je vrednost eksponenta 94-101=-7.
   - Pošto su prve dve cifre kombinacije 01, preostale cifre kombinacije (111) odredjuju 7 kao dekadnu cifru najveće težine frakcije.
3. Dekodiranjem dekleta frakcije se dobija vrednost frakcije. Dekodiranje može da se izvede pomoću tabele koja daje šematski prikaz dekodiranja na sledeći način:

```
pqr stu v  wxy       pqr stu v  wxy
000 000 0  000       000 011 0  101            DPD dekleti

0000 0000 0000       0000 0011 0101            BCD zapis
abcd efgh ijkm       abcd efgh ijkm
  0    0    0          0    3    5             Dekadni zapis
```

   Primedba: u ovom slučaju dekleti su mogli jednostavnije da se dekodiraju na osnovu treće osobine navedene kao prednost DPD kodiranja u odnosu na Chen-Ho kodiranje, odakle se direktno dobija da je dekodirana vrednost dekleta 000 035.

Frakcija koja se dobija je 7000035. Broj koji je zapisan ima vrednost $+7000035*10^{-7}$, odnosno $7.000035*10^{-1}$, tj. 0.7000035

Zadaci za vežbu

1. Zapisati sledeće brojeve u pokretnom zarezu pomoću dekadne osnove u jednostrukoj tačnosti, koristeći dekadno (DPD) kodiranje:

   (a) -245

   (b) +245.00

   (c) +345.678*10^{65}

   (d) 177.36/31

   (e) -778.3456264556*10^{-35}. Zadatak uraditi za različite načine zaokruživanja!

2. Koji dekadni brojevi su predstavljeni brojevima u pokretnom zarezu zapisanim u IEEE 754 zapisu pomoću dekadne osnove (DPD kodiranje):

   (a) 0 00000 000000 0000 0000 0000 0000 1100

   (b) 1 00010 000000 0000 0000 0000 0000 0000

   (c) 0 10001 101101 0000 0000 0000 0000 0000

   (d) 1 01001 011110 1100 0000 0010 0000 1111

   (e) 0 10101 110010 0001 0100 0000 0110 0101

   (f) 1 11011 101011 0000 0000 0000 0000 0000

   (g) 0 11111 011100 0001 0000 0000 0000 0001

   (h) 1 11110 101011 0000 0100 1000 0000 0000