# Programiranje 2

Laboratorijske vježbe

## LV1
## Zapis cijelih i realnih brojeva u računalu

**Fakultet elektrotehnike računarstva i informacijskih tehnologija Osijek**

Kneza Trpimira 2b

**www.ferit.unios.hr**

# Brojevni sustav

Brojevni sustav je način zapisivanja brojeva, njihovo tumačenje. Danas se koristi položajni brojevni sustav kod kojeg položaj znamenke u zapisu određuje njezinu težinsku vrijednost. Svaki brojevni sustav ima određeni broj znamenki, a ukupan broj različitih znamenki naziva se bazom brojevnog sustava.

| Naziv | Baza | Znamenke | Označavanje |
|---|---|---|---|
| **Dekadski** | 10 | 0, 1, ... , 9 | (10) |
| **Binarni** | 2 | 0, 1 | (2) |
| **Oktalni** | 8 | 0, 1, ... , 7 | (8) |
| **Heksadekadski** | 16 | 0, 1, ... , 9, A, B, C, D, E, F | (16) |

Svaka znamenka brojevnog sustava kreće se od 0 do $B - 1$, gdje je $B$ baza brojevnog sustava.
U svakom brojevnom sustavu, svaka znamenka broja ima jedinstvenu težinsku vrijednost koja se dobije tako da se vrijednost baze brojevnog sustava potencira eksponentom čija vrijednost ovisi o položaju znamenke. Krajnji desni eksponent ima vrijednost 0, a krajnje lijevi ima vrijednost $n - 1$, gdje je $n$ broj znamenaka broja.

## Pretvaranje u dekadski brojevni sustav

Moguće je svaki broj iz bilo kojeg brojevnog sustava pretvoriti u broj iz dekadskog brojevnog sustava rastavljanjem na težinske vrijednosti.

*Primjer 1*: Rastavljanje brojeva iz različitih brojevnih sustava na težinske vrijednosti.

$$0101_2 = 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 4 + 1 = 5_{10},$$
$$1234_8 = 1 \cdot 8^3 + 2 \cdot 8^2 + 3 \cdot 8^1 + 4 \cdot 8^0 = 512 + 128 + 24 + 4 = 668_{10},$$
$$\text{C}34_{16} = 12 \cdot 16^2 + 3 \cdot 16^1 + 4 \cdot 16^0 = 3072 + 48 + 4 = 3124_{10}.$$

## Pretvaranje iz dekadskog brojevnog sustava u binarni, oktalni i heksadekadski brojevni sustav

*Algoritam:*

1. Provodi se cjelobrojno dijeljenje, broj koji se preračunava iz dekadskog brojevnog sustava, dijeli se s bazom sustava u koji se želi preračunati broj,
2. Ostatak cjelobrojnog dijeljenja zapisuje se desno od rezultata dijeljenja,
3. Dobiveni rezultat koji se dobio iz prethodnog dijeljenja nastavlja se dijeliti s bazom sustava u koji se želi preračunati broj,
4. Ponavlja se korak 2. i 3. sve dok se za rezultat dijeljenja ne dobije 0.

*Primjer 2:* Pretvaranje broja iz dekadskog brojevnog sustava u heksadekadski, oktalni i binarni sustav.

| $32_{10} \rightarrow ?_2$ | Ostatak | $512_{10} \rightarrow ?_8$ | Ostatak | $1024_{10} \rightarrow ?_{16}$ | Ostatak |
|---|---|---|---|---|---|
| 32:2 = 16 | 0 | 512:8 = 64 | 0 | 1024:16 = 64 | 0 |
| 16:2 = 8 | 0 | 64:8 = 8 | 0 | 64:16 = 4 | 0 |
| 8:2 = 4 | 0 | 8:8 = 1 | 0 | 4:16 = 0 | 4 |
| 4:2 = 2 | 0 | 1:8 = 0 | 1 | | |
| 2:2 = 1 | 0 | | | | |
| 1:2 = 0 | 1 | | | | |
| $\mathbf{32_{10} \rightarrow 100000_2}$ | | $512_{10} \rightarrow 1000_8$ | | $1024_{10} \rightarrow 400_{16}$ | |

## Pretvaranje iz oktalnog brojevnog sustava u binarni

Kako binarni brojevni sustav ima bazu 2, a oktalni sustav ima bazu 8, može se zapisati $8 = 2^3$. Jedna znamenka oktalnog sustava može se zamijeniti s tri znamenke binarnog sustava.

| B = 8 | B = 2 |
|---|---|
| 0 | 000 |
| 1 | 001 |
| 2 | 010 |
| 3 | 011 |
| 4 | 100 |
| 5 | 101 |
| 6 | 110 |
| 7 | 111 |

*Primjer 3*: $512_8 = ?_2$

$$5|1|2_8 = 101|001|010_2$$

## Pretvaranje iz binarnog brojevnog sustava u oktalni

Pretvaranje broja iz binarnog brojevnog sustava u oktalni sustav provodi se tako da se znamenke binarnog sustava grupiraju po tri znamenke, počevši od krajnje desne. Ako krajnje lijeva grupa znamenaka nemaju tri znamenke, onda se grupa nadopuni nulama koje se stavljaju na početak zapisa. Nakon što se grupiraju znamenke, iz tablice se očitaju vrijednosti u oktalnom sustavu, redoslijedom kako su postavljene grupe u binarnom sustavu.

*Primjer 4*: $101001010_2 = ?_8$

$$101|001|010_2 = 5|1|2_8$$

## Pretvaranje iz heksadekadskog brojevnog sustav u binarni

Kako binarni brojevni sustav ima bazu 2, a heksadekadski ima bazu 16, može se zapisati $16 = 2^4$. Jedna znamenka heksadekadskog sustava može se zamijeniti s četiri znamenke binarnog sustava.

| B = 16 | B = 2 |
|---|---|
| 0 | 0000 |
| 1 | 0001 |
| 2 | 0010 |
| 3 | 0011 |
| 4 | 0100 |
| 5 | 0101 |
| 6 | 0110 |
| 7 | 0111 |
| 8 | 1000 |
| 9 | 1001 |
| A | 1010 |
| B | 1011 |
| C | 1100 |
| D | 1101 |
| E | 1110 |
| F | 1111 |

*Primjer 5*: $512_{16} =?_2$

$$5|1|2_{16} = 0101|0001|0010_2$$

## Pretvaranje iz binarnog brojevnog sustav u heksadekadski

Pretvaranje broja iz binarnog brojevnog sustava u heksadekadski sustav provodi se tako da se znamenke binarnog sustava grupiraju po četiri znamenke, počevši od krajnje desne. Ako krajnje lijeva grupa znamenaka nemaju četiri znamenke, onda se grupa nadopuni nulama koje se stavljaju na početak zapisa. Nakon što se grupiraju znamenke, iz tablice se očitaju vrijednosti u heksadekadskom sustavu, redoslijedom kako su postavljene grupe u binarnom sustavu.

*Primjer 6*: $010100010010_2 =?_{16}$

$$0101|0001|0010_2 = 5|1|2_{16}$$

# Zapis cijelih brojeva u računalu

Binarni zapis cijelog broja zapisuje se u memorijske registre koji mogu biti različite duljine, 8, 16, 32, 64 bita. Kako se jedan bajt sastoji od 8 bitova, za pretvorbu dekadskog broja u binarni zapis koristi će se minimalno 8 bita.

Prvi bit u registru (krajnje lijevi) je *MSB* (engl. *most significant bit*), dok zadnji bit u registru (krajnje desni) je *LSB* (engl. *least significant bit*).

*MSB* bit u registru određuje predznak broja:

- MSB bit = 0, broj je pozitivan,
- MSB bit = 1, broj je negativan.

Za zapis negativnih cijelih brojeva u memorijski registar računala potrebno je koristiti drugi komplement, tada MSB bit predstavlja predznak broja i omogućava se zapis negativni i pozitivnih cijelih brojeva iz određenog intervala. Ako se isključivo zapisuju pozitivni cijeli brojevi tada se ne koristi drugi komplement i MSB bit ne predstavlja više predznak broja te se i taj bit koristi za zapis pozitivnog cijelog broja iz određenog intervala.

Raspon cijelih negativnih brojeva za pojedinu duljinu registra su:

- 8 bita ($-2^7$ do $2^7 - 1$),
- 16 bita ($-2^{15}$ do $2^{15} - 1$),
- 32 bita ($-2^{31}$ do $2^{31} - 1$),
- 64 bita ($-2^{63}$ do $2^{63} - 1$).

Raspon cijelih pozitivnih brojeva za pojedinu duljinu registra su:

- 8 bita (0 do $2^8 - 1$),
- 16 bita (0 do $2^{16} - 1$),
- 32 bita (0 do $2^{32} - 1$),
- 64 bita (0 do $2^{64} - 1$).

## Zapis pozitivnog dekadskog broja u 32-bitni registar

Pretvaranje dekadskog broja u binarni broj vrši se prema prethodno objašnjenom načinu.

*Primjer 7*: Zapisivanje dekadskog broja $5_{10}$ u 32 bitni registar.

$$5_{10} = 00000101_2$$

| 31 | | | | | | | | 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |

U slučaju da se binarni broj kraćeg zapisa zapisuje u registar duljeg zapisa, onda se prostor ispred krajnjeg lijevog bita kraćeg binarnog broja nadopunjuje nulama kako bi se registar popunio.

## Zapis negativnog dekadskog broja u 32-bitni registar

Kako bi se moglo ispravno prikazati pretvaranje negativnog dekadskog broja u binarni ekvivalent, potrebno je koristiti drugi komplement.

Postupak pretvaranja negativnog dekadskog broja u binarni ekvivalent:

1. Uzima se apsolutna vrijednost dekadskog broja i pretvara se u binarni ekvivalent,
2. Svaku nulu binarnog broja treba pretvoriti u jedinicu, a svaku jedinicu u nulu (ovim putem se dobiva prvi komplement binarnog broja). Problem prvog komplementa je taj što se pojavljuju dva zapisa za nulu, tj. negativna i regularna nula i time se ograničava širina intervala kojim bi se prikazao dodatni negativni broj.
3. Zato se prvom komplementu binarnog broja pribraja binarna jedinica (ovim putem se dobiva drugi komplement broja).

Drugim se komplementom izbjegava dobivanje negativne 0, već je ona jedinstvena.

***Pretvaranje broja u drugi komplement može se napraviti na jednostavniji način, počevši od najmanjeg značajnog bita invertiraju se svi bitovi nakon prve pojave jedinice***.

*Primjer 8*: Postupak zbrajanja dva binarna broja.

| Prijenos | | 1 | 1 | 1 | - | - | 1 | |
|---|---|---|---|---|---|---|---|---|
| | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 |
| + | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| = | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 |

*Primjer 9*: Postupak pretvaranja negativnog dekadskog broja u 32 bitni binarni ekvivalent:

Zadan je broj -5, potrebno je broj pretvoriti u binarni zapis duljine 32 bita.

1. Uzima se apsolutna vrijednost broja: **|-5| = 5**
2. Broj 5 je potrebno pretvoriti u binarni broj: $\mathbf{5_{10} = 00000101_2}$
3. Pretvoreni binarni broj potrebno je nadopuniti nulama do 32 bita duljine:
   **0000 0000 0000 0000 0000 0000 0000 0101**
4. Potrebno je napraviti prvi komplement nad 32 bitni brojem:
   **1111 1111 1111 1111 1111 1111 1111 1010**
5. Prvom komplementu je potrebno pribrojiti jedan kako bi se dobio drugi komplement, odnosno konačno rješenje:

**1111 1111 1111 1111 1111 1111 1111 1010 + 1 = 1111 1111 1111 1111 1111 1111 1111 1011**

## Zapis negativnog 32-bitnog binarnog broja u dekadski ekvivalent

Postupak pretvaranja negativnog binarnog broja u dekadski ekvivalent:

1. Pamti se predznak negativnog binarnog broja.
2. Svaku nulu binarnog broja treba pretvoriti u jedinicu, a svaku jedinicu u nulu (ovim putem se dobiva prvi komplement binarnog broja).
3. Prvom komplementu binarnog broja pribraja se binarna jedinica (ovim putem se dobiva drugi komplement broja).

4. Iz dobivenog binarnog broja radi se pretvaranje u dekadski ekvivalent.
5. Dobivenom dekadskom broj zapisuje se negativni predznak koji se zapamtio u prvom koraku

*Primjer 10*: Postupak pretvaranja negativnog 32 bitnog binarnog broja u dekadski ekvivalent:

Zadan je 32 bitni negativni broj **1111 1111 1111 1111 1111 1111 1111 1011**, potrebno ga je pretvoriti u dekadski ekvivalent.

1. Potrebno je zapamtiti predznak: **(-)**
2. Nad binarnim brojem radi se prvi komplement:

**0000 0000 0000 0000 0000 0000 0000 0100**

3. Prvom komplementu je potrebno pribrojiti jedan kako bi se dobio drugi komplement:

**0000 0000 0000 0000 0000 0000 0000 0100 + 1 = 0000 0000 0000 0000 0000 0000 0000 0101**

4. Binarni broj se nakon drugog komplementa pretvara u dekadski ekvivalent:

$$\mathbf{1 \cdot 2^2 + 1 \cdot 2^0 = 5_{10}}$$

5. Dekadskom broju se dodaje predznak i dobiva se konačno rješenje:

$$\mathbf{-5_{10}}$$

# Zapis realnih brojeva u računalu

Realni brojevi se mogu zapisati u računalu na dva načina:

- nepomičnom točkom i
- s pomičnom točkom (engl. *floating point*).

Zapis realnog broja u računalu s nepomičnom točkom je nepraktično jer zauzima fiksan dio registra za zapis cijelog broja i fiksan dio registra za zapis decimalnog dijela broja, te predstavlja neprecizan zapis broja. Ovim načinom se raspon brojeva koji se mogu zapisati u računalu poprilično smanjuje jer ovisi o duljini registra. Kako je registar ograničene duljine, da bi se mogao zapisati veći raspon realnih brojeva koristi se zapis realnog broja u računalu s pomičnom točkom.

U širokoj primjeni zapisa broja s pomičnom točkom koristi se standard **IEEE 754**. Dva su načina zapisa broja s pomičnom točkom:

- Jednostrukom preciznošću (širine 32 bita),
- Dvostrukom preciznošću (širine 64 bita).

## IEEE 754 – zapis jednostrukom preciznošću

Za zapis realnog broja jednostrukom preciznošću koristi se 32 bita širina registra.

| 31 | 30      23 | 22      0 |
|---|---|---|
| Predznak (P) | Karakteristika (K) | Mantisa (M) |

Predznak (*P*), predstavlja *MSB*:

- *MSB* bit = 0, broj je pozitivan,
- *MSB* bit = 1, broj je negativan.

Karakteristika (*K*), predstavlja eksponent i za nju se koristi 8 bitova, (bitovi 30 - 23).

$$K = (be + 127)_{10}.$$

Za određivanje karakteristike potrebno je binarni eksponent (*be*) zbrojiti s 127. Ovaj način je odabran kako bi se izbjegao zapis negativnih eksponenata. Dobiveni dekadski rezultat je potrebno pretvoriti u binarni ekvivalent.

Svaki realni broj u binarnom obliku potrebno je normirati prije zapisa u pogodnom obliku. Zapis normiranog broja ima oblik $1.M_2 \cdot 2^{be}$.

*Primjer 11*: Primjer normiranja binarnog broja:

$$101.101_2 = 1.01101_2 \cdot 2^2$$

Mantisa (*M*), predstavlja decimalni dio normirane vrijednosti broja koji ima oblik $1.M_2$. Kod normiranog broja oblika $1.M_2$ vodeća se jedinica ne zapisuje u računalo i naziva se skriveni bit, dok se decimalni dio broja $M_2$ zapisuje u preostala 23 bita, (bitovi 22 - 0).

**IEEE 754 pravilo pretvaranja dekadskog broja u binarni ekvivalent jednostruke preciznosti:**

1. Ako je apsolutna vrijednost broja veća ili jednaka 2:
   a. Broj se dijeli s 2 tako i svaki sljedeći rezultat dijeljenja se dijeli s 2, sve dok je rezultat dijeljenja veći od 2 i pamti se broj dijeljenja ($be$),
   b. Broj se dijeli s $2^{be}$ ,
   c. U dio za karakteristiku upisuje se vrijednost $K = (be + 127)_{10}$, u binarnom zapisu (8 bita),
   d. Mantisa se računa tako da se uzme decimalni dio rezultata iz koraka ***b*** i pomnoži se s 2, tako i svaki sljedeći rezultat množenja se množi s 2, a cjelobrojni dio rezultata se upisuje u dio registra od 23. bita na desno.
2. Ako je apsolutna vrijednost broja veća od 0 i manja od 1:
   a. Broj se množi s 2 tako i svaki sljedeći rezultat množenja se množi s 2 sve dok je rezultat množenja manji od 1 i pamti se broj množenja ($be$),
   b. Broj se dijeli s $2^{-be}$(ili množi s $2^{be}$),
   c. U dio za eksponent upisuje se vrijednost $K = (-be + 127)_{10}$, u binarnom zapisu (8 bitova),
   d. Mantisa se računa tako da se uzme decimalni dio rezultata iz koraka ***b*** i pomnoži se s 2, tako i svaki sljedeći rezultat množenja se množi s 2, a cjelobrojni dio rezultata se upisuje u dio registra od 23. bita na desno.
3. Ako je apsolutna vrijednost broja veća ili jednaka 1 i manja od 2:
   a. *be* = 0, *K* = 127,
   b. Mantisa se računa tako da se uzme decimalni dio rezultata iz koraka ***b*** i pomnoži se s 2, tako i svaki sljedeći rezultat množenja se množi s 2, a cjelobrojni dio rezultata se upisuje u dio registra od 23. bita na desno.

Napomena za korak gdje se popunjava mantisa:

1. Ako se u ovom koraku pretvaranja ne popuni registar od 23 bita, dozvoljeno ga je popuniti nulama s desne strane zadnjeg pretvorenog rezultata jer se na taj način ne mijenja vrijednost broja, a registar se u potpunosti popuni.
2. Ako se dogodi situacija beskonačnog pretvaranja decimalnog dijela, potrebno je pretvoriti rezultat taman do popunjavanja mantise od 23 bita.

**Opće pravilo pretvaranja realnog dekadskog broja u binarni ekvivalent:**

1. Dekadski broj se podijeli na cijeli dio i decimalni dio,
2. Cijeli dio dekadskog broja se dijeli s 2 sve dok je rezultat dijeljenja različit od 0, (pretvaranje dekadskog broja u binarni ekvivalent).
3. Decimalni dio se množi s 2, a cijeli dio rezultata množenja je znamenka u binarnom sustavu koji se pamti,
4. Ako je rezultat množenja veći od 0, nastavlja se množenje decimalnog dijela prethodnog rezultata množenja s 2, a cijeli dio se pamti,
5. Ponavlja se korak 3 i 4 sve dok se za decimalni dio rezultat množenja ne dobije 0.

*Primjer 12*: IEEE 754 pravilo pretvaranja za broj čija je apsolutna vrijednost veća ili jednaka 2, $-5.75_{10}$

Računa se s apsolutnom vrijednosti broja $|-5.75|_{10} = 5.75_{10}$.

Broj je negativan, stoga je P = 1.

a. Broj se dijeli s 2 tako i svaki sljedeći rezultat dijeljenja se dijeli s 2, sve dok je rezultat dijeljenja veći od 2 i pamti se broj dijeljenja (*be*),

| |
|---|
| 5.75 : 2 = 2.875 |
| 2.875 : 2 = 1.4375 |
| ***be*** = 2 |

b. Broj se dijeli s $2^{be}$,

| |
|---|
| 5.75 : $2^2$ = 1.4375 |

c. U dio za karakteristiku upisuje se vrijednost $K = (be + 127)_{10}$, u binarnom zapisu (8 bita),

$$K = (2 + 127)_{10} = 129_{10} = 10000001_2$$

d. Mantisa se računa tako da se uzme decimalni dio rezultata iz koraka ***b*** i pomnoži se s 2, tako i svaki sljedeći rezultat množenja se množi s 2, a cjelobrojni dio rezultata se upisuje u dio registra od 23. bita na desno.

| | |
|---|---|
| 0.4375 · 2 = 0.875 | 0 |
| 0.875 · 2 = 1.75 | 1 |
| 0.75 · 2 = 1.5 | 1 |
| 0.5 · 2 = 1.0 | 1 |
| 0.0 · 2 = 0 | 0 |

Konačni zapis realnog broja s pomičnom točkom u računalu:

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

*Primjer 13*: Opće pravilo pretvaranja za broj čija je apsolutna vrijednost veća ili jednaka 2, $-5.75_{10}$.

Računa se s apsolutnom vrijednosti broja $|-5.75|_{10} = 5.75_{10}$.

Prema općem pravilu pretvaranja realnog dekadskog broja u binarni ekvivalent potrebno je posebno pretvoriti cijeli dio broja i decimalni dio broja u binarni ekvivalent.

| $5_{10} \rightarrow ?_2$ | Ostatak | $0.75_{10} \rightarrow ?_2$ | Ostatak |
|---|---|---|---|
| 5:2 = 2 | 1 | 0.75x2 = 1.5 | 1 |
| 2:2 = 1 | 0 | 0.5x2 = 1.0 | 1 |
| 1:2 = 0 | 1 | 0.0x2 = 0 | 0 |
| $5_{10} \rightarrow \mathbf{101_2}$ | | $0.75_{10} \rightarrow 110_2$ | |

Broj je negativan, P = 1. Dobiveni binarni ekvivalent je potrebno normirati kako bi se dobio oblik $1.M_2 \cdot 2^{be}$ slijedi zapis.

$$101.110_2 = 1.01110_2 \cdot 2^2$$

Da bi broj bio normiran potrebno je imati vodeću jedinicu, zbog toga je bilo potrebno napraviti pomak decimalne točke u lijevu stranu za dva decimalna mjesta te se dobio pozitivan pomak za *be* koji iznosi 2.

Karakteristika $K = (2 + 127)_{10} = 129_{10} = 10000001_2$

Konačni zapis realnog broja s pomičnom točkom u računalu:

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

*Primjer 14*: IEEE 754 pravilo pretvaranja za broj čija je apsolutna vrijednost veća od 0, i manja od 1, $0.16_{10}$

Broj je pozitivan, stoga je P = 0.

a. Broj se množi s 2 tako i svaki sljedeći rezultat množenja se množi s 2 sve dok je rezultat množenja manji od 1 i pamti se broj množenja (*be*),

| |
|---|
| 0.16 · 2 = 0.32 |
| 0.32 · 2 = 0.64 |
| 0.64 · 2 = 1.28 |
| ***be*** = 3 |

b. Broj se dijeli s $2^{-be}$(ili množi s $2^{be}$),

| |
|---|
| $0.16 \cdot \mathbf{2^3} = 1.28$ |
| $0.16 : \mathbf{2^{-3}} = 1.28$ |

c. U dio za eksponent upisuje se vrijednost $K = (-be + 127)_{10}$, u binarnom zapisu (8 bitova),

$$K = (-3 + 127)_{10} = 124_{10} = 01111100_2$$

d. Mantisa se računa tako da se uzme decimalni dio rezultata iz koraka ***b*** i pomnoži se s 2, tako i svaki sljedeći rezultat množenja se množi s 2, a cjelobrojni dio rezultata se upisuje u dio registra od 23. bita na desno.

| | |
|---|---|
| 0.28 · 2 = 0.56 | 0 |
| 0.56 · 2 = 1.12 | 1 |
| 0.12 · 2 = 0.24 | 0 |
| 0.24 · 2 = 0.48 | 0 |
| 0.48 · 2 = 0.96 | 0 |
| 0.96 · 2 = 1.92 | 1 |
| ... | |

Konačni zapis realnog broja s pomičnom točkom u računalu:

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |

*Primjer 15*: Opće pravilo pretvaranja za broj čija je apsolutna vrijednost veća od 0, i manja od 1, $0.16_{10}$

U ovom slučaju dogodit će se beskonačno pretvaranje dekadskog decimalnog dijela broj u binarni ekvivalent, gdje je potrebno množiti više od 23 puta zbog normiranja broja. Potrebno je primijetiti kako se u jednom trenutku pretvaranja ponavlja uzorak množenja, odnosno rezultat pamćenja cijelog dijela broja.

| $0_{10} \rightarrow ?_2$ | Ostatak | $0.16_{10} \rightarrow ?_2$ | Ostatak |
|---|---|---|---|
| 0:2 = 0 | 0 | 0.16x2 = 0.32 | 0 |
| | | 0.32x2 = 0.64 | 0 |
| | | 0.64x2 = 1.28 | 1 |
| | | 0.28x2 = 0.56 | 0 |
| | | 0.56x2 = 1.12 | 1 |
| | | 0.12x2 = 0.24 | 0 |
| | | 0.24x2 = 0.48 | 0 |
| | | 0.48x2 = 0.96 | 0 |
| | | 0.96x2 = 1.92 | 1 |
| | | 0.92x2 = 1.84 | 1 |
| | | 0.84x2 = 1.68 | 1 |
| | | 0.68x2 = 1.36 | 1 |
| | | 0.36x2 = 0.72 | 0 |
| | | 0.72x2 = 1.44 | 1 |
| | | 0.44x2 = 0.88 | 0 |
| | | 0.88x2 = 1.76 | 1 |
| | | 0.76x2 = 1.52 | 1 |
| | | 0.52x2 = 1.04 | 1 |
| | | 0.04x2 = 0.08 | 0 |
| | | 0.08x2 = 0.16 | 0 |
| | | 0.16x2 = 0.32 | 0 |
| | | 0.32x2 = 0.64 | 0 |
| | | ... | |
| $0_{10} \rightarrow \mathbf{0_2}$ | | $0.16_{10} \rightarrow 0010\ldots_2$ | |

Broj je pozitiva, P = 0. Dobiveni binarni ekvivalent je potrebno normirati kako bi se dobio oblik $1.M_2 \cdot 2^{be}$ slijedi zapis.

$$\boxed{0.00101000\ldots0001_2 = 1.01000111\ldots1010_2 \cdot 2^{-3}}$$

Da bi broj bio normiran potrebno je imati vodeću jedinicu, zbog toga je bilo potrebno napraviti pomak decimalne točke u desnu stranu za tri decimalna mjesta te se dobio negativan pomak *be* koji iznosi 3.

Karakteristika $K = (-3 + 127)_{10} = 124_{10} = 01111100_2$

Konačni zapis realnog broja s pomičnom točkom u računalu:

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |

*Primjer 16*: IEEE 754 pravilo pretvaranja za broj čija apsolutna vrijednost veća ili jednaka 1, i manja od 2, $1.125_{10}$

Broj je pozitivan, stoga je P = 0.

a. *be* = 0, *K* = 127, $K = (be + 127)_{10}$, u binarnom zapisu (8 bitova),

$$K = (0 + 127)_{10} = 127_{10} = 01111111_2$$

b. Mantisa se računa tako da se uzme decimalni dio rezultata iz koraka ***b*** i pomnoži se s 2, tako i svaki sljedeći rezultat množenja se množi s 2, a cjelobrojni dio rezultata se upisuje u dio registra od 23. bita na desno.

| | |
|---|---|
| 0.125 · 2 = 0.25 | 0 |
| 0.25 · 2 = 0.5 | 0 |
| 0.5 · 2 = 1.0 | 1 |
| 0.0 · 2 = 0.0 | 0 |

Konačni zapis realnog broja s pomičnom točkom u računalu:

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

*Primjer 17*: Opće pravilo pretvaranja za broj čija apsolutna vrijednost veća ili jednaka 1, i manja od 2, $1.125_{10}$

Broj u dekadskom sustavu već ima normirani oblik koji je potrebno pretvoriti u binarni ekvivalent.

| $1_{10} \to ?_2$ | Ostatak | $0.125_{10} \to ?_2$ | Ostatak |
|---|---|---|---|
| 1:2 = 0 | 1 | 0.125x2 = 0.250 | 0 |
| | | 0.250x2 = 0.5 | 0 |
| | | 0.5x2 = 1.0 | 1 |
| | | 0x2 = 0 | 0 |
| $1_{10} \to \mathbf{1_2}$ | | $0.125_{10} \to 0010_2$ | |

Broj je pozitiva, P = 0. Dobiveni binarni ekvivalent je potrebno normirati kako bi se dobio oblik $1.M_2 \cdot 2^{be}$ slijedi zapis.

$$\boxed{1.0010_2 = 1.0010_2 \cdot 2^0.}$$

Pretvaranjem broj u binarni ekvivalent dobiven je normirani oblik. Stoga nema potrebe za pomicanjem decimalne točke i zato pomak *be* iznosi 0.

Karakteristika $K = (0 + 127)_{10} = 127_{10} = 01111111_2$

Konačni zapis realnog broja s pomičnom točkom u računalu:

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Specijalni slučajevi:**

1. ako je *K* = 0 i svi bitovi *M* = 0, onda se radi o broju 0,
2. ako je *K* = 255 i svi bitovi *M* = 0, onda se radi o -∞ (ako je *P* = 1), odnosno +∞ (ako je *P* = 0).
3. ako je *K* = 255, a neki bit *M* ≠ 0, onda se radi o *NaN* (engl. *Not a number*), nije broj.

## Pretvaranje iz binarnog zapisa jednostruke preciznosti u dekadski

*Primjer 18*: binarni zapis broja $-10.5_{10}$:

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Iz binarnog zapisa može se zaključiti sljedeće:

1. broj je negativan, *P* = 1,
2. karakteristika koja predstavlja eksponent iznosi $K = 10000010_2 = 130_{10}$, iz nje je moguće izračunati binarni eksponent:
   $K = (be + 127)_{10} \rightarrow be = K - 127_{10} = 130_{10} - 127_{10} = 3_{10}$.
   Binarni eksponent iznosi 3 i potrebno je decimalnu točku pomaknuti u desnu stranu za tri decimalna mjesta.
3. konačni broj ima oblik $1.M_2 \cdot 2^{be} = 1.0101_2 \cdot 2^3 = 1010.1_2 = 1 \cdot 2^3 + 1 \cdot 2^1 +. 1 \cdot 2^{-1} = 10.5_{10} \rightarrow -10.5_{10}$

*Primjer 19*: binarni zapis broja $0.25_{10}$:

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Iz binarnog zapisa može se zaključiti sljedeće:

1. broj je pozitivan, *P* = 0,
2. karakteristika koja predstavlja eksponent iznosi $K = 01111101_2 = 125_{10}$, iz nje je moguće izračunati binarni eksponent:
   $K = (be + 127)_{10} \rightarrow be = K - 127_{10} = 125_{10} - 127_{10} = -2_{10}$.
   Binarni eksponent iznosi -2 i potrebno je decimalnu točku pomaknuti u lijevu stranu za dva decimalna mjesta.
3. konačni broj ima oblik $1.M_2 \cdot 2^{be} = 1.0_2 \cdot 2^{-2} = 0.01_2 = .1 \cdot 2^{-2} = 0.25_{10}$

*Primjer 20*: binarni zapis broja $1.125_{10}$:

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Iz binarnog zapisa može se zaključiti sljedeće:

1. broj je pozitivan, *P* = 0,

2. karakteristika koja predstavlja eksponent iznosi $K = 01111111_2 = 127_{10}$, iz nje je moguće izračunati binarni eksponent:
   $K = (be + 127)_{10} \rightarrow be = K - 127_{10} = 127_{10} - 127_{10} = 0_{10}$.
   Binarni eksponent iznosi 0, stoga neće biti pomicanja decimalne točke.
3. konačni broj ima oblik $1.M_2 \cdot 2^{be} = 1.001_2 \cdot 2^0 = 1.001_2 = 1 \cdot 2^0 + .1 \cdot 2^{-3} = 1.125_{10}$

## IEEE 754 – zapis dvostrukom preciznošću

Za zapis realnog broja dvostrukom preciznošću koristi se 64 bita širina registra.

| 63 | 62       52 | 51       0 |
|---|---|---|
| Predznak (P) | Karakteristika (K) | Mantisa (M) |

Za ovaj zapis vrijede sva pravila kao i kod zapisa jednostrukom preciznošću jedino je razlika u broju bitova koji se koriste za zapis karakteristike koji iznosi 11 bita, a za zapis mantise potrebno je 52 bita.

Karakteristika (*K*) se dobije izrazom:

$$K = (be + 1023)_{10}$$

Za određivanje karakteristike potrebno je binarni eksponent (*be*) zbrojiti s 1023. Ovaj način je odabran kako bi se izbjegao zapis negativnih eksponenata. Dobiveni dekadski rezultat je potrebno pretvoriti u binarni ekvivalent.

Svaki realni broj u binarnom obliku potrebno je normirati prije zapisa u pogodnom obliku. Zapis normiranog broja ima oblik $1.M_2 \cdot 2^{be}$.

Mantisa (*M*), predstavlja decimalni dio normirane vrijednosti broja koji ima oblik $1.M_2$. Kod normiranog broja oblika $1.M_2$ vodeća se jedinica ne zapisuje u računalo i naziva se skriveni bit, dok se decimalni dio broja $M_2$ zapisuje u preostala 52 bita, (bitovi 51 - 0).

**IEEE 754 pravilo pretvaranja dekadskog broja u binarni ekvivalent dvostruke preciznosti:**

1. Ako je apsolutna vrijednost broja veća ili jednaka 2:
   a. Broj se dijeli s 2 tako i svaki sljedeći rezultat dijeljenja se dijeli s 2, sve dok je rezultat dijeljenja veći od 2 i pamti se broj dijeljenja (*be*),
   b. Broj se dijeli s $2^{be}$ ,
   c. U dio za karakteristiku upisuje se vrijednost $K = (be + 1023)_{10}$, u binarnom zapisu (11 bita),
   d. Mantisa se računa tako da se uzme decimalni dio rezultata iz koraka ***b*** i pomnoži se s 2, tako i svaki sljedeći rezultat množenja se množi s 2, a cjelobrojni dio rezultata se upisuje u dio registra od 52. bita na desno.
2. Ako je apsolutna vrijednost broja veća od 0 i manja od 1:
   a. Broj se množi s 2 tako i svaki sljedeći rezultat množenja se množi s 2 sve dok je rezultat množenja manji od 1 i pamti se broj množenja (*be*),
   b. Broj se dijeli s $2^{-be}$(ili množi s $2^{be}$),
   c. U dio za eksponent upisuje se vrijednost $K = (-be + 1023)_{10}$, u binarnom zapisu (11 bitova),
   d. Mantisa se računa tako da se uzme decimalni dio rezultata iz koraka ***b*** i pomnoži se s 2, tako i svaki sljedeći rezultat množenja se množi s 2, a cjelobrojni dio rezultata se upisuje u dio registra od 52. bita na desno.
3. Ako je apsolutna vrijednost broja veća ili jednaka 1 i manja od 2:
   a. *be* = 0, *K* = 1023,
   b. Mantisa se računa tako da se uzme decimalni dio rezultata iz koraka ***b*** i pomnoži se s 2, tako i svaki sljedeći rezultat množenja se množi s 2, a cjelobrojni dio rezultata se upisuje u dio registra od 52. bita na desno.

Napomena za korak gdje se popunjava mantisa:
1. Ako se u ovom koraku pretvaranja ne popuni registar od 52 bita, dozvoljeno ga je popuniti nulama s desne strane zadnjeg pretvorenog rezultata jer se na taj način ne mijenja vrijednost broja, a registar se u potpunosti popuni.
2. Ako se dogodi situacija beskonačnog pretvaranja decimalnog dijela, potrebno je pretvoriti rezultat taman do popunjavanja mantise od 52 bita.

*Primjer 21*: IEEE 754 pravilo pretvaranja za broj čija je apsolutna vrijednost veća ili jednaka 2, $-6.36_{10}$

Računa se s apsolutnom vrijednosti broja $|-6.36|_{10} = 6.36_{10}$.

Broj je negativan, stoga je P = 1.

a. Broj se dijeli s 2 tako i svaki sljedeći rezultat dijeljenja se dijeli s 2, sve dok je rezultat dijeljenja veći od 2 i pamti se broj dijeljenja (*be*),

| |
|---|
| 6.36 : 2 = 3.18 |
| 3.18 : 2 = 1.59 |
| ***be*** = 2 |

b. Broj se dijeli s $2^{be}$,

| 6.36 : $2^2$ = 1.59 |
|---|

c. U dio za karakteristiku upisuje se vrijednost $K = (be + 1023)_{10}$, u binarnom zapisu (11 bita),

$$K = (2 + 1023)_{10} = 1025_{10} = 10000000001_2$$

d. Mantisa se računa tako da se uzme decimalni dio rezultata iz koraka ***b*** i pomnoži se s 2, tako i svaki sljedeći rezultat množenja se množi s 2, a cjelobrojni dio rezultata se upisuje u dio registra od 23. bita na desno.

| | |
|---|---|
| 0.59 · 2 = 1.18 | 1 |
| 0.18 · 2 = 0.36 | 0 |
| 0.36 · 2 = 0.72 | 0 |
| 0.72 · 2 = 1.44 | 1 |
| 0.44 · 2 = 0.88 | 0 |
| 0.88 · 2 = 1.76 | 1 |
| ... | |

Konačni zapis realnog broja s pomičnom točkom u računalu:

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |

*Primjer 22*: Opće pravilo pretvaranja za broj čija je apsolutna vrijednost veća ili jednaka 2, $-6.36_{10}$.

Računa se s apsolutnom vrijednosti broja $|-6.36|_{10} = 6.36_{10}$.

U ovom slučaju dogodit će se beskonačno pretvaranje dekadskog decimalnog dijela broj u binarni ekvivalent, gdje je potrebno množiti više od 52 puta zbog normiranja broja. Potrebno je primijetiti kako se u jednom trenutku pretvaranja ponavlja uzorak množenja, odnosno rezultat pamćenja cijelog dijela broja.

| $6_{10} \rightarrow ?_2$ | Ostatak | $0.36_{10} \rightarrow ?_2$ | Ostatak |
|---|---|---|---|
| 6:2 = 3 | 0 | 0.36x2 = 0.72 | 0 |
| 3:2 = 1 | 1 | 0.72x2 = 1.44 | 1 |
| 1:2 = 0 | 1 | 0.44x2 = 0.88 | 0 |
| | | 0.88x2 = 1.76 | 1 |
| | | 0.76x2 = 1.52 | 1 |
| | | 0.52x2 = 1.04 | 1 |
| | | 0.04x2 = 0.08 | 0 |
| | | 0.08x2 = 0.16 | 0 |
| | | 0.16x2 = 0.32 | 0 |
| | | 0.32x2 = 0.64 | 0 |
| | | 0.64x2 = 1.28 | 1 |
| | | 0.28x2 = 0.56 | 0 |
| | | 0.56x2 = 1.12 | 1 |
| | | 0.12x2 = 0.24 | 0 |
| | | 0.24x2 = 0.48 | 0 |

| | | | |
|---|---|---|---|
| | | 0.48x2 = 0.96 | 0 |
| | | 0.96x2 = 1.92 | 1 |
| | | 0.92x2 = 1.84 | 1 |
| | | 0.84x2 = 1.68 | 1 |
| | | 0.68x2 = 1.36 | 1 |
| | | 0.36x2 = 0.72 | 0 |
| | | 0.72x2 = 1.44 | 1 |
| | | ... | |
| $6_{10} \rightarrow \mathbf{110_2}$ | | $0.16_{10} \rightarrow 0101\ldots_2$ | |

Broj je negativan, P = 1. Dobiveni binarni ekvivalent je potrebno normirati kako bi se dobio oblik $1.M_2 \cdot 2^{be}$ slijedi zapis.

$$\boxed{110.01011100\ldots0101_2 = 1.10010111\ldots0001_2 \cdot 2^2}$$

Da bi broj bio normiran potrebno je imati vodeću jedinicu, zbog toga je bilo potrebno napraviti pomak decimalne točke u lijevu stranu za dva decimalna mjesta te se dobio pozitivan pomak za *be* koji iznosi 2.

Karakteristika $K = (2 + 1023)_{10} = 1025_{10} = 10000000001_2$

Konačni zapis realnog broja s pomičnom točkom u računalu:

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |

*Primjer 23*: IEEE 754 pravilo pretvaranja za broj čija je apsolutna vrijednost veća od 0, i manja od 1, $0.25_{10}$

Broj je pozitivan, stoga je P = 0.

a. Broj se množi s 2 tako i svaki sljedeći rezultat množenja se množi s 2 sve dok je rezultat množenja manji od 1 i pamti se broj množenja (*be*),

| |
|---|
| 0.25 · 2 = 0.5 |
| 0.5 · 2 = 1.0 |
| ***be*** = 2 |

b. Broj se dijeli s $2^{-be}$(ili množi s $2^{be}$),

| |
|---|
| 0.25 · $\mathbf{2^2}$ = 1.0 |
| 0.25 : $\mathbf{2^{-2}}$= 1.0 |

c. U dio za eksponent upisuje se vrijednost $K = (-be + 1023)_{10}$, u binarnom zapisu (11 bitova),

$$K = (-2 + 1023)_{10} = 1021_{10} = 01111111101_2$$

d. Mantisa se računa tako da se uzme decimalni dio rezultata iz koraka ***b*** i pomnoži se s 2, tako i svaki sljedeći rezultat množenja se množi s 2, a cjelobrojni dio rezultata se upisuje u dio registra od 23. bita na desno.

| | |
|---|---|
| 0.0 · 2 = 0.0 | 0 |
| ... | 0 |
| | 0 |
| | 0 |
| | 0 |
| | 0 |

Konačni zapis realnog broja s pomičnom točkom u računalu:

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

*Primjer 22*: Opće pravilo pretvaranja za broj čija je apsolutna vrijednost veća od 0, i manja od 1, $0.25_{10}$

Prema općem pravilu pretvaranja realnog dekadskog broja u binarni ekvivalent potrebno je posebno pretvoriti cijeli dio broja i decimalni dio broja u binarni ekvivalent.

| $0_{10} \rightarrow ?_2$ | Ostatak | $0.25_{10} \rightarrow ?_2$ | Ostatak |
|---|---|---|---|
| 0:2 = 0 | 0 | 0.25x2 = 0.5 | 0 |
| | | 0.5x2 = 1.0 | 1 |
| | | 0.0x2 = 0.0 | 0 |
| $0_{10} \rightarrow \mathbf{0_2}$ | | $0.25_{10} \rightarrow 010_2$ | |

Broj je pozitiva, P = 0. Dobiveni binarni ekvivalent je potrebno normirati kako bi se dobio oblik $1.M_2 \cdot 2^{be}$ slijedi zapis.

$$0.010_2 = 1.0_2 \cdot 2^{-2}$$

Da bi broj bio normiran potrebno je imati vodeću jedinicu, zbog toga je bilo potrebno napraviti pomak decimalne točke u desnu stranu za dva decimalna mjesta te se dobio negativan pomak *be* koji iznosi 2.

Karakteristika $K = (-2 + 1023)_{10} = 1021_{10} = 01111111101_2$

Konačni zapis realnog broja s pomičnom točkom u računalu:

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

*Primjer 23*: IEEE 754 pravilo pretvaranja za broj čija apsolutna vrijednost veća ili jednaka 1, i manja od 2, $1.75_{10}$

Broj je pozitivan, stoga je P = 0.

c. *be* = 0, *K* = 1023, $K = (be + 1023)_{10}$, u binarnom zapisu (11 bitova),

$$K = (0 + 1023)_{10} = 1023_{10} = 01111111111_2$$

d. Mantisa se računa tako da se uzme decimalni dio rezultata iz koraka ***b*** i pomnoži se s 2, tako i svaki sljedeći rezultat množenja se množi s 2, a cjelobrojni dio rezultata se upisuje u dio registra od 23. bita na desno.

| | |
|---|---|
| 0.75 · 2 = 1.5 | 1 |
| 0.5 · 2 = 1.0 | 1 |
| 0.0 · 2 = 0.0 | 0 |
| … | 0 |

Konačni zapis realnog broja s pomičnom točkom u računalu:

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

*Primjer 24*: Opće pravilo pretvaranja za broj čija apsolutna vrijednost veća ili jednaka 1, i manja od 2, $1.75_{10}$

Broj u dekadskom sustavu već ima normirani oblik koji je potrebno pretvoriti u binarni ekvivalent.

| $1_{10} \to ?_2$ | Ostatak | $0.75_{10} \to ?_2$ | Ostatak |
|---|---|---|---|
| 1:2 = 0 | 1 | 0.75x2 = 1.5 | 1 |
| | | 0.5x2 = 1.0 | 1 |
| | | 0.0x2 = 0.0 | 0 |
| $1_{10} \to \mathbf{1_2}$ | | $0.75_{10} \to 110_2$ | |

Broj je pozitiva, P = 0. Dobiveni binarni ekvivalent je potrebno normirati kako bi se dobio oblik $1.M_2 \cdot 2^{be}$ slijedi zapis.

$$\boxed{1.110_2 = 1.110\ldots_2 \cdot 2^0.}$$

Pretvaranjem broj u binarni ekvivalent dobiven je normirani oblik. Stoga nema potrebe za pomicanjem decimalne točke i zato pomak *be* iznosi 0.

Karakteristika $K = (0 + 1023)_{10} = 1023_{10} = 01111111111_2$

Konačni zapis realnog broja s pomičnom točkom u računalu:

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Specijalni slučajevi:**

1. ako je *K* = 0 i svi bitovi *M* = 0, onda se radi o broju 0,
2. ako je *K* = 2047 i svi bitovi *M* = 0, onda se radi o -∞ (ako je *P* = 1), odnosno +∞ (ako je *P* = 0).
3. ako je *K* = 2047, a neki bit *M* ≠ 0, onda se radi o *NaN* (*engl*. Not a number), nije broj.

## Pretvaranje iz binarnog zapisa dvostruke preciznosti u dekadski

*Primjer 26*: binarni zapis broja $2.0_{10}$:

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Iz binarnog zapisa možemo zaključiti sljedeće:

1. broj je pozitivan, *P* = 0,
2. karakteristika koja predstavlja eksponent iznosi $K = 10000000000_2 = 1024_{10}$, iz nje je moguće izračunati binarni eksponent:
   $K = (be + 1023)_{10} \rightarrow be = K - 1023_{10} = 1024_{10} - 1023_{10} = 1_{10}$.
   Binarni eksponent iznosi 1 i potrebno je decimalnu točku pomaknuti u desnu stranu za jedno decimalno mjesto.
3. konačni broj ima oblik $1.M_2 \cdot 2^{be} = 1.0_2 \cdot 2^1 = 10.0_2 = 1 \cdot 2^1 = 2_{10}$

*Primjer 27*: binarni zapis broja $-0.626953125_{10}$:

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Iz binarnog zapisa možemo zaključiti sljedeće:

1. broj je pozitivan, *P* = 0,
2. karakteristika koja predstavlja eksponent iznosi $K = 01111111110_2 = 1022_{10}$, iz nje je moguće izračunati binarni eksponent:
   $K = (be + 1023)_{10} \rightarrow be = K - 1023_{10} = 1022_{10} - 1023_{10} = -1_{10}$.
   Binarni eksponent iznosi -1 i potrebno je decimalnu točku pomaknuti u lijevu stranu za jedno decimalno mjesto.
3. konačni broj ima oblik $1.M_2 \cdot 2^{be} = 1.01000001_2 \cdot 2^{-1} = 0.101000001_2 = 0 +.1 \cdot 2^{-1} + 1 \cdot 2^{-3} + 1 \cdot 2^{-9} = 0.626953125_{10} \rightarrow -0.626953125_{10}$

*Primjer 28*: binarni zapis broja $1.0712890625_{10}$:

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Iz binarnog zapisa možemo zaključiti sljedeće:

1. broj je negativan, *P* = 1,
2. karakteristika koja predstavlja eksponent iznosi $K = 01111111111_2 = 1023_{10}$, iz nje je moguće izračunati binarni eksponent:

$K = (be + 1023)_{10} \rightarrow be = K - 1023_{10} = 1023_{10} - 1023_{10} = 0_{10}$.
Binarni eksponent iznosi 0, stoga neće biti pomicanja decimalne točke.

3. Konačni broj ima oblik $1.M_2 \cdot 2^{be} = 1.0001001001_2 \cdot 2^0 = 1.0001001001_2 = 1 \cdot 2^0 + .1 \cdot 2^{-4} + 1 \cdot 2^{-7} + 1 \cdot 2^{-10} = 1.0712890625_{10}$

# Zadaci za vježbu

1. Napisati C program koji s tipkovnice učitava cijeli broj u intervalu [-128, 127]. Potrebno je izračunati i na ekran ispisati binarni zapis učitanog broja duljine 8 bita. Pripaziti na postupak pretvorbe kada je broj negativan i kada je pozitivan.
2. Napisati C program koji s tipkovnice učitava brojeve 0 i 1 u polje cijelih brojeva duljine 8 elemenata. Potrebno je izračunati i na ekran ispisati dekadski zapis učitanog broja. Pripaziti na postupak pretvorbe kada je broj negativan i kada je pozitivan.
3. Napisati C program koji će iz cjelobrojnog polja koje predstavlja binarni zapis duljine 32 elemenata popunjenog pseudo-slučajnim vrijednostima [0, 1], pretvoriti i ispisati dekadski zapis broja. Pripaziti na postupak pretvorbe kada je broj negativan i kada je pozitivan.
4. Napisati C program koji će pseudo-slučajno generirati broj iz intervala za *short* tip podatka, te taj broj pretvoriti u binarni ekvivalent popunjavanjem cjelobrojnog polja duljine 32 elementa. Pripaziti na postupak pretvorbe kada je broj negativan i kada je pozitivan.
5. Napisati C program koji će korisniku omogućiti unos realnog broja koji je veći ili jednak 2 s barem tri decimalna mjesta i pomoću standarda IEEE 754 jednostruke preciznosti pretvoriti broj u binarni ekvivalent popunjavanjem cjelobrojnog polja duljine 32 elementa. Pripaziti na postupak pretvorbe kada je broj negativan i kada je pozitivan.
6. Napisati C program koji će korisniku omogućiti unos realnog broja koji je veći od 0 i manji od 1 s barem tri decimalna mjesta i pomoću standarda IEEE 754 jednostruke preciznosti pretvoriti broj u binarni ekvivalent popunjavanjem cjelobrojnog polja duljine 32 elementa. Pripaziti na postupak pretvorbe kada je broj negativan i kada je pozitivan.
7. Napisati C program koji će korisniku omogućiti unos realnog broja koji je veći ili jedna 1 i manji od 2 s barem tri decimalna mjesta i pomoću standarda IEEE 754 jednostruke preciznosti pretvoriti broj u binarni ekvivalent popunjavanjem cjelobrojnog polja duljine 32 elementa. Pripaziti na postupak pretvorbe kada je broj negativan i kada je pozitivan.
8. Napisati C program koji će korisniku omogućiti unos realnog broja koji je veći ili jednak 2 s barem tri decimalna mjesta i pomoću standarda IEEE 754 dvostruke preciznosti pretvoriti broj u binarni ekvivalent popunjavanjem cjelobrojnog polja duljine 64 elementa. Pripaziti na postupak pretvorbe kada je broj negativan i kada je pozitivan.
9. Napisati C program koji će korisniku omogućiti unos realnog broja koji je veći od 0 i manji od 1 s barem tri decimalna mjesta i pomoću standarda IEEE 754 dvostruke preciznosti pretvoriti broj u binarni ekvivalent popunjavanjem cjelobrojnog polja duljine 64 elementa. Pripaziti na postupak pretvorbe kada je broj negativan i kada je pozitivan.
10. Napisati C program koji će korisniku omogućiti unos realnog broja koji je veći ili jedna 1 i manji od 2 s barem tri decimalna mjesta i pomoću standarda IEEE 754 dvostruke preciznosti pretvoriti broj u binarni ekvivalent popunjavanjem cjelobrojnog polja duljine 64 elementa. Pripaziti na postupak pretvorbe kada je broj negativan i kada je pozitivan.