

Uvod u organizaciju i arhitekturu računara 1

Stefan Mišković

2020/2021.

3 Množenje i deljenje

3.1 Množenje neoznačenih brojeva

Množenje neoznačenih binarnih brojeva je slično množenju dekadnih brojeva, na način kako bi se to radilo pomoću papira i olovke. Posmatrajmo, na primer, množenje dekadnih brojeva 14 i 9. Njihovi binarni zapisi su redom 1110 i 1001. Izračunavanje njihovog proizvoda se odvija na sledeći način:

```
1110 x 1001
-----
      1110
     0000
    0000
   1110
  -----
 1111110
```

Binarni zapis 1111110 u dekadnom sistemu predstavlja broj 126. Mada je operacija množenja komutativna, u računaru se prvi i drugi činilac posmatraju odvojeno. Nazovimo prvi činilac množenikom, a drugi množiocem. U svakom koraku se množenik množi ciframa množioca. Ako je cifra množioca 1, tada se potpiše sâm množenik, a ako je cifra množioca 0, tada se dopiše onoliko nula koliko množenik ima cifara. Možemo primetiti i da se u svakom koraku međurezultat uvlači za pojedno mesto ulevo. Pretpostavimo, radi lakšeg snalaženja, da množenik i množilac imaju jednak broj cifara. Ako je taj broj cifara jednak k , tada proizvod može imati najviše $2k$ cifara. U našem primeru, činioći imaju po 4 cifre, a proizvod 7. Da je postojao prenos na poslednjoj poziciji, proizvod bi imao 8 cifara. Na osnovu svih pomenutih razmatranja, može se konstruisati hardverski algoritam za množenje dva neoznačena cela binarna broja.

Neka su data dva neoznačena cela binarna broja sa po k cifara i neka je potrebno izračunati njihov proizvod. Za zapis njihovog proizvoda će biti dovoljno $2k$ bitova. Prvi činilac nazovimo množenikom i označimo ga sa M , a drugi množiocem i označimo ga sa P . M i P možemo smatrati binarnim registrima koji su dužine k . Registri zapravo predstavljaju niske od k bitova (nula ili jedinica). Za algoritam su nam potrebni još registar A dužine k i jednobitni registar C .

Na početku algoritma u registru M je upisan množenik, a u registru P množilac, kao neoznačeni binarni brojevi sa k cifara. U registre A i C se upisuje onoliko nula kolika je odgovarajuća dužina registra. Neka je poslednji bit registra P označen sa P_0 . Algoritam se izvršava u k koraka od kojih se svaki sastoji iz dva dela:

- U prvom delu, ako je $P_0 = 0$, ne vrši se nikakva akcija. Ako je $P_0 = 1$, vrši se sabiranje brojeva koji su u registrima A i M , a dobijeni rezultat postaje nova vrednost registra A . Eventualni prenos koji se može pojaviti prilikom sabiranja se upisuje u registar C .
- U drugom delu se registri C , A i P posmatraju kao jedinstven registar od $2k + 1$ bitova, tako što se nadovežu jedan na drugi. Pritom se vrši logičko pomeranje pomeranje registra CAP udesno. Logičko pomeranje znači da će se svi bitovi osim poslednjeg pomeriti za jedno mesto udesno, poslednji će se izgubiti, a na mesto prvog bita sleva novodobijenog broja se dodaje 0. Na primer, logičko pomeranje udesno niza od šest bitova 100101 daje kao rezultat nisku 010010.

Vrednost proizvoda je smešten u registar AP , koji posmatramo kao jedinstven registar od $2k$ bitova, dobijen nadovezivanjem registara A i P .

Primer. Dekadne brojeve 112 i 9 zapisati kao neoznačene cele brojeve u binarnom sistemu na 8 mesta, pa izvršiti njihovo množenje hardverskim algoritmom. Dobijeni rezultat prevesti u dekadni sistem.

Prevođenjem brojeva 112 i 9 u binarni sistem i njihovim zapisivanjem na 8 mesta dobijaju se vrednosti $(112)_{10} = (0111000)_2^8$ i $(9)_{10} = (00001001)_2^8$. Prema tome, početne vrednosti registara koji označavaju množenik i množilac su redom $M = 0111000$ i $P = 00001001$. Početne vrednosti ostalih registara se inicijalizuju nulama, odnosno iznose $A = 00000000$ i $C = 0$. Vrednost registra M ostaje nepromenljiva, a vrednosti ostalih registara se menjaju kroz narednih 8 koraka, što je prikazano tabelom ispod.

| Korak | C | A | P | Komentar |
|-------|-----|----------|----------|---|
| 0 | 0 | 00000000 | 00001001 | Vrši se inicijalizacija. |
| 1 | 0 | 01110000 | 00001001 | $P_0 = 1$, vrši se sabiranje $A = A + M$. |
| | 0 | 00111000 | 00000100 | Registar CAP se logički pomera udesno. |
| 2 | 0 | 00111000 | 00000100 | $P_0 = 0$, ne vrši se nikakva akcija. |
| | 0 | 00011100 | 00000010 | Registar CAP se logički pomera udesno. |
| 3 | 0 | 00011100 | 00000010 | $P_0 = 0$, ne vrši se nikakva akcija. |
| | 0 | 00001110 | 00000001 | Registar CAP se logički pomera udesno. |
| 4 | 0 | 01111110 | 00000001 | $P_0 = 1$, vrši se sabiranje $A = A + M$. |
| | 0 | 00111111 | 00000000 | Registar CAP se logički pomera udesno. |
| 5 | 0 | 00111111 | 00000000 | $P_0 = 0$, ne vrši se nikakva akcija. |
| | 0 | 00011111 | 10000000 | Registar CAP se logički pomera udesno. |
| 6 | 0 | 00011111 | 10000000 | $P_0 = 0$, ne vrši se nikakva akcija. |
| | 0 | 00001111 | 11000000 | Registar CAP se logički pomera udesno. |
| 7 | 0 | 00001111 | 11000000 | $P_0 = 0$, ne vrši se nikakva akcija. |
| | 0 | 00000111 | 11100000 | Registar CAP se logički pomera udesno. |
| 8 | 0 | 00000111 | 11100000 | $P_0 = 0$, ne vrši se nikakva akcija. |
| | 0 | 00000011 | 11110000 | Registar CAP se logički pomera udesno. |

Rezultat množenja je sadržan u registru AP . Prevođenjem u dekadni sistem se dobija

$$(1111110000)_2 = (1008)_{10}.$$

3.2 Množenje označenih brojeva (Butov algoritam)

Ovde će biti predstavljen hardverski algoritam za množenje označenih celih binarnih brojeva zapisanih u potpunom komplementu, koji nosi naziv Butov algoritam. Neka su data dva cela binarna broja zapisana sa po k cifara u potpunom komplementu i neka je potrebno izračunati njihov proizvod. Za zapis njihovog proizvoda će biti dovoljno $2k$ bitova. Prvi činilac nazovimo množenikom i označimo ga sa M , a drugi množiocem i označimo ga sa P . M i P možemo smatrati binarnim registrima koji su dužine k . Od pomoćnih registara se koriste A i P_{-1} , od kojih prvi ima k bitova, a drugi 1 bit. Oba pomoćna registra se na početku inicijalizuju nulama. Neka je poslednji bit registra P označen sa P_0 . Algoritam se izvršava u k koraka od kojih se svaki sastoji iz dva dela:

- U prvom delu se posmatra par registara P_0P_{-1} . Ako je $P_0P_{-1} = 00$ ili $P_0P_{-1} = 11$, ne vrši se nikakva akcija. Ako je $P_0P_{-1} = 01$, vrši se sabiranje brojeva koji su u registrima A i M , a dobijeni rezultat postaje nova vrednost registra A . Ako je $P_0P_{-1} = 10$, vrši se oduzimanje brojeva koji su u registrima A i M . Kako se operacije sabiranja i oduzimanja vrše u potpunom komplementu, nije potrebno razmatrati poslednji prenos. Pritom se oduzimanje može izvršiti direktno, ili se svesti na sabiranje, gde se $A - M$ svodi na $A + M'$, pri čemu je M' registar čija je vrednost promenjenog znaka broja koji je u registru M .
- U drugom delu se registri A , P i P_{-1} posmatraju kao jedinstven registar od $2k + 1$ bitova, tako što se nadovežu jedan na drugi. Pritom se vrši aritmetičko pomeranje pomeranje registra APP_{-1} udesno. Aritmetičko pomeranje znači da će se svi bitovi osim poslednjeg pomeriti za jedno mesto udesno, poslednji će se izgubiti, a na mesto prvog bita sleva novodobijenog broja se dodaje cifra koja je prethodno bila prva. Na primer, aritmetičko pomeranje udesno niza od šest bitova 100101 daje kao rezultat nisku 110010.

Vrednost proizvoda je smešten u registar AP , koji se posmatra kao jedinstven registar nadovezanih registara A i P .

Primer. Dekadne brojeve 103 i -13 zapisati kao označene cele brojeve u binarnom sistemu u potpunom komplementu na 8 mesta, pa izvršiti njihovo množenje Butovim algoritmom. Dobijeni rezultat prevesti iz potpunog komplementa u dekadni sistem.

Prevođenjem brojeva 103 i -13 u binarni sistem u potpuni komplement na 8 mesta dobija se $(103)_{10} = (01100111)_2^8$ i $(-13)_{10} = (11110011)_2^8$. Prema tome, početne vrednosti registara koji označavaju množenik i množilac su redom $M = 01100111$ i $P = 11110011$. Početne vrednosti ostalih registara se inicijalizuju nulama, odnosno iznose $A = 00000000$ i $P_{-1} = 0$. Vrednost registra M ostaje nepromenljiva, a vrednosti ostalih registara se menjaju kroz narednih 8 koraka, što je prikazano tabelom ispod.

| Korak | A | P | P_{-1} | Komentar |
|-------|----------|----------|----------|--|
| 0 | 00000000 | 11110011 | 0 | Vrši se inicijalizacija. |
| 1 | 10011001 | 11110011 | 0 | $P_0P_{-1} = 10$, pa se vrši oduzimanje $A = A - M$. |
| | 11001100 | 11111001 | 1 | Registar APP_{-1} se aritmetički pomera udesno. |
| 2 | 11001100 | 11111001 | 1 | Kako je $P_0P_{-1} = 11$, ne vrši se nikakva akcija. |
| | 11100110 | 01111100 | 1 | Registar APP_{-1} se aritmetički pomera udesno. |
| 3 | 01001101 | 01111100 | 1 | $P_0P_{-1} = 01$, pa se vrši sabiranje $A = A + M$. |
| | 00100110 | 10111110 | 0 | Registar APP_{-1} se aritmetički pomera udesno. |
| 4 | 00100110 | 10111110 | 0 | Kako je $P_0P_{-1} = 00$, ne vrši se nikakva akcija. |
| | 00010011 | 01011111 | 0 | Registar APP_{-1} se aritmetički pomera udesno. |
| 5 | 10101100 | 01011111 | 0 | $P_0P_{-1} = 10$, pa se vrši oduzimanje $A = A - M$. |
| | 11010110 | 00101111 | 1 | Registar APP_{-1} se aritmetički pomera udesno. |
| 6 | 11010110 | 00101111 | 1 | Kako je $P_0P_{-1} = 11$, ne vrši se nikakva akcija. |
| | 11101011 | 00010111 | 1 | Registar APP_{-1} se aritmetički pomera udesno. |
| 7 | 11101011 | 00010111 | 1 | Kako je $P_0P_{-1} = 11$, ne vrši se nikakva akcija. |
| | 11110101 | 10001011 | 1 | Registar APP_{-1} se aritmetički pomera udesno. |
| 8 | 11110101 | 10001011 | 1 | Kako je $P_0P_{-1} = 11$, ne vrši se nikakva akcija. |
| | 11111010 | 11000101 | 1 | Registar APP_{-1} se aritmetički pomera udesno. |

Rezultat je $AP = 1111101011000101$. Prevođenjem iz potpunog komplementa u dekadni sistem se dobija

$$(1111101011000101)_2^{16} = (-1339)_{10}.$$

3.3 Modifikovani Butov algoritam

Modifikovani Butov algoritam najčešće omogućava efikasnije množenje dva binarna broja u potpunom komplementu. Postupak ćemo objasniti kroz primer množenja dekadnih brojeva -28 i 111 , zapisanih u potpunom komplementu. Ako se za množilac koristi k bitova, za množenik je potrebno iskoristiti $2k$ bitova. U našem primeru, množenik će biti zapisan na 16, a množilac na 8 bitova.

Najpre se prevođenjem dekadnih brojeva -28 i 111 u binarni sistem u potpuni komplement na odgovarajući broj mesta dobijaju zapisi $(-28)_{10} = (1111111111100100)_2^{16}$ i $(111)_{10} = (01101111)_2^8$. Zapis 01101111 se naziva Butov množilac, a od njega se dobija Butov kodirani množilac na sledeći način:

$$\begin{array}{cccccccc} 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ \hline +1 & 0 & -1 & +1 & 0 & 0 & 0 & -1 \end{array}$$

Za prebacivanje množioca u modifikovani oblik ideja je uočiti uzastopne nizove jedinica krećući se zdesna nalevo. Početak serije jedinica treba obeležiti sa -1 , a prvu pojavu nule nakon serije jedinica sa $+1$. Na svim ostalim pozicijama treba upisati nule.

Neka je Butov kodirani množilac označen sa $a_7a_6a_5a_4a_3a_2a_1a_0$. U nastavku algoritma, parove izdvajamo zdesna nalevo počevši od pozicije najmanje težine: (a_1, a_0) , (a_3, a_2) , (a_5, a_4) i (a_7, a_6) . Svakom paru (a_i, a_{i-1}) treba pridružiti odgovarajuću vrednost sa $(a_{2i+1}, a_{2i}) \rightarrow 2a_{2i+1} + a_{2i}$. Moguće vrednosti parova koje se na ovaj način mogu dobiti su -2 , -1 , 0 , 1 i 2 . Za posmatrani primer se dobijaju sledeće vrednosti parova (parovi su numerisani redom sa $i = 0, 1, 2, 3$):

| i | Par cifara | Vrednost |
|-----|-------------------------|-------------------------|
| 0 | $(a_1, a_0) = (0, -1)$ | $2 \cdot 0 + (-1) = -1$ |
| 1 | $(a_3, a_2) = (0, 0)$ | $2 \cdot 0 + 0 = 0$ |
| 2 | $(a_5, a_4) = (-1, +1)$ | $2 \cdot (-1) + 1 = -1$ |
| 3 | $(a_7, a_6) = (+1, 0)$ | $2 \cdot 1 + 0 = 2$ |

Dalje, za svaku vrednost broja i ($i = 0, 1, 2, 3$) prvo treba pomeriti množenik za $2i$ bita ulevo, a zatim tako dobijeni binaran broj treba pomnožiti vrednošću i -tog para. Pritom važi:

- Ako je vrednost para 2, množenje se svodi na pomeranje množenika za jednu poziciju ulevo.
- Ako je vrednost para 1, množenik se ne menja.
- Ako je vrednost para 0, rezultat je 0.
- Ako je vrednost para -1 , množenje se svodi na menjanje znaka množeniku (komplementiranje i dodavanje jedinice na poziciju najmanje težine).
- Ako je vrednost para -2 , množenje se svodi na menjanje znaka množeniku, a zatim na pomeranje tako dobijenog broja za jednu poziciju ulevo (isti efekat će se postići i ako se promeni redosled promene znaka i pomeranja ulevo).

Konačan proizvod se dobija sabiranjem svih međuproizvoda. Sabiranje se izvodi po pravilima koja važe za sabiranje brojeva u potpunom komplementu. Na taj način se u primeru rezultat dobija sabiranjem međuproizvoda u poslednjoj koloni sledeće tabele:

| i | Vrednost para | Pomeren množenik | Međuproizvod |
|-----|---------------|-------------------|-------------------|
| 0 | -1 | 11111111 11100100 | 00000000 00011100 |
| 1 | 0 | 11111111 10010000 | 00000000 00000000 |
| 2 | -1 | 11111110 01000000 | 00000001 11000000 |
| 3 | 2 | 11111001 00000000 | 11110010 00000000 |

Sabiranjem se dobija:

```

00000000 00011100
00000000 00000000
00000001 11000000
+ 11110010 00000000
-----
11110011 11011100

```

Prevođenjem rezultata 1111001111011100 iz potpunog komplementa u dekadni sistem se dobija

$$(1111001111011100)_2^{16} = (-3108)_{10}.$$

3.4 Deljenje neoznačenih brojeva

Deljenje neoznačenih binarnih brojeva je slično deljenju dekadnih brojeva, na način kako bi se to radilo pomoću papira i olovke. Ovde ćemo govoriti o celobrojnomo deljenju. Posmatrajmo, na primer, deljenje dekadnih brojeva 53 i 5. Njihovi binarni zapisi su redom 110101 i 101. Izračunavanje njihovog količnika i ostatka se odvija na sledeći način:

$$\begin{array}{r} 110101 : 101 = 1010 \\ 101 \\ --- \\ 110 \\ 101 \\ --- \\ 11 \end{array}$$

Količnik je $(1010)_2 = (10)_{10}$, a ostatak $(11)_2 = (3)_{10}$. Ukoliko se delilac sadrži u trenutnim ciframa deljenika koje se razmatraju, količniku se dopisuje cifra 1, a delilac se potpisuje ispod razmatranih cifara i vrši se oduzimanje. Inače, ukoliko se ne sadrži, količniku se dopisuje cifra 0. U oba slučaja se na trenutni ostatak dodaje naredna cifra deljenika. Postupak se ponavlja dok god se ne iskoriste sve cifre deljenika. Na osnovu prethodnog opisa, može se konstruisati hardverski algoritam za deljenje dva neoznačena binarna broja.

Neka su data dva neoznačena cela binarna broja sa po k cifara i neka je potrebno izračunati njihov količnik i ostatak celobrojnomo deljenjem. Označimo deljenik sa P , a delilac sa M . P i M možemo smatrati binarnim registrima koji su dužine k . Za algoritam deljenja nam je potreban i registar A dužine k .

Na početku algoritma u registru P je upisan deljenik, a u registru M delilac, kao neoznačeni binarni brojevi sa po k cifara. U registar A se upisuje k nula. Algoritam se izvršava u k koraka od kojih se svaki sastoji iz dva dela:

- U prvom delu se registri A i P posmatraju kao jedinstven registar od $2k$ bitova, tako što se nadovežu jedan na drugi. Pritom se vrši pomeranje registra AP ulevo. Svi bitovi osim prvog će se pomeriti za jedno mesto ulevo, prvi će se izgubiti, a na mesto poslednjeg bita novodobijenog broja se dodaje 0.
- U drugom delu, ako je $A < M$, ne vrši se nikakva akcija. Inače, ako je $A \geq M$, vrši se oduzimanje $A = A - M$, a u najniži bit registra P se upisuje cifra 1.

Vrednost količnika je na kraju smeštena u registar P , a ostatka u registar A .

Primer. Dekadne brojeve 24 i 9 zapisati kao neoznačene cele brojeve u binarnom sistemu na 8 mesta, pa izvršiti njihovo deljenje hardverskim algoritmom. Dobijeni količnik i ostatak prevesti u dekadni sistem.

Prevođenjem brojeva 24 i 9 u binarni sistem i njihovim zapisivanjem na 8 mesta dobijaju se vrednosti $(24)_{10} = (00011000)_2$ i $(9)_{10} = (00001001)_2$. Prema tome, početne vrednosti registara koji označavaju deljenik i delilac su redom $P = 00011000$ i $M = 00001001$. Početna vrednost pomoćnog registara je $A = 00000000$. Vrednost registra M ostaje nepromenljiva, a vrednosti ostalih registara se menjaju kroz narednih 8 koraka, što je prikazano tabelom ispod.

| Korak | A | P | Komentar |
|-------|----------|----------|--|
| 0 | 00000000 | 00011000 | Vrši se inicijalizacija. |
| 1 | 00000000 | 00110000 | Registar AP se pomera ulevo. |
| | 00000000 | 00110000 | Kako je $A < M$, ne vrši se nikakva akcija. |
| 2 | 00000000 | 01100000 | Registar AP se pomera ulevo. |
| | 00000000 | 01100000 | Kako je $A < M$, ne vrši se nikakva akcija. |
| 3 | 00000000 | 11000000 | Registar AP se pomera ulevo. |
| | 00000000 | 11000000 | Kako je $A < M$, ne vrši se nikakva akcija. |
| 4 | 00000001 | 10000000 | Registar AP se pomera ulevo. |
| | 00000001 | 10000000 | Kako je $A < M$, ne vrši se nikakva akcija. |
| 5 | 00000011 | 00000000 | Registar AP se pomera ulevo. |
| | 00000011 | 00000000 | Kako je $A < M$, ne vrši se nikakva akcija. |
| 6 | 00000110 | 00000000 | Registar AP se pomera ulevo. |
| | 00000110 | 00000000 | Kako je $A < M$, ne vrši se nikakva akcija. |
| 7 | 00001100 | 00000000 | Registar AP se pomera ulevo. |
| | 00000011 | 00000001 | Kako $A \geq M$, važi $A = A - M$ i $P_0 = 1$. |
| 8 | 00000110 | 00000010 | Registar AP se pomera ulevo. |
| | 00000110 | 00000010 | Kako je $A < M$, ne vrši se nikakva akcija. |

3.5 Deljenje označenih brojeva

Ovde će biti predstavljen hardverski algoritam za deljenje označenih celih binarnih brojeva zapisanih u potpunom komplementu. Neka su data dva cela binarna broja zapisana u potpunom komplementu i neka je potrebno izračunati njihov količnik i ostatak. Neka je deljenik zapisan sa $2k$, a delilac sa k bitova. Deljenik se na početku zapisuje u jedinstven registar AP koji ima $2k$ bitova, koji se dobija nadovezivanjem registara A i P , od kojih svaki ima po k bitova. Pritom treba voditi računa da tada, ako je deljenik pozitivan, registar A će počinjati nulama, a ako je negativan, registar A će počinjati jedinicama. Algoritam se izvršava u k koraka od kojih se svaki sastoji iz dva dela:

- U prvom delu se registri A i P posmatraju kao jedinstven registar od $2k$ bitova, tako što se nadovežu jedan na drugi. Pritom se vrši pomeranje registra AP ulevo. Svi bitovi osim prvog će se pomeriti za jedno mesto ulevo, prvi će se izgubiti, a na mesto poslednjeg bita novodobijenog broja se dodaje 0.
- U drugom delu se proverava da li su zadovoljeni odgovarajući uslovi da se izvrši sledeća akcija:
 - Ako su brojevi A i M različitog znaka, vrši se sabiranje $A = A + M$, a u najniži bit registra P se upisuje cifra 1.
 - Ako su brojevi A i M istog znaka, vrši se oduzimanje $A = A - M$, a u najniži bit registra P se upisuje cifra 1.

Za izvršavanje pomenute akcije, dovoljno je da je ispunjen jedan od sledeća dva uslova:

- Nakon izvršene akcije sabiranja ili oduzimanja registar A neće promeniti znak.
- Ako se stiglo do poslednjeg koraka, nakon izvršene akcije sabiranja ili oduzimanja registar A ima vrednost 0.

Inače se opisana akcija neće izvršiti.

Vrednost ostatka je na kraju smeštena u registar A . Ukoliko su deljenik i delilac istog znaka, vrednost količnika je smeštena u registar P , a ako su različitog, za količnik treba uzeti vrednost registra P sa promenjenim znakom.

Primer. Dekadne brojeve 24 i -9 zapisati kao označene cele brojeve u binarnom sistemu u potpunom komplementu na 16 i 8 mesta, redom, pa izvršiti njihovo deljenje hardverskim algoritmom. Dobijeni količnik i ostatak prevesti iz potpunog komplementa u dekadni sistem.

Prevođenjem brojeva 24 i -9 u binarni sistem u potpuni komplement na 16 i 8 mesta dobija se $(24)_{10} = (0000000000011000)_2^{16}$ i $(-9)_{10} = (11110111)_2^8$. Prema tome, početne vrednosti registara koji označavaju deljenik i delilac su redom $AP = 0000000000011000$ i $M = 11110111$. Vrednost registra M ostaje nepromenljiva, a vrednosti ostalih registara se menjaju kroz narednih 8 koraka, što je prikazano tabelom ispod.

| Korak | A | P | Komentar |
|-------|----------|----------|--|
| 0 | 00000000 | 00011000 | Vrši se inicijalizacija. |
| 1 | 00000000 | 00110000 | Registar AP se pomera ulevo. |
| | 00000000 | 00110000 | Ne vrši se nikakva akcija. |
| 2 | 00000000 | 01100000 | Registar AP se pomera ulevo. |
| | 00000000 | 01100000 | Ne vrši se nikakva akcija. |
| 3 | 00000000 | 11000000 | Registar AP se pomera ulevo. |
| | 00000000 | 11000000 | Ne vrši se nikakva akcija. |
| 4 | 00000001 | 10000000 | Registar AP se pomera ulevo. |
| | 00000001 | 10000000 | Ne vrši se nikakva akcija. |
| 5 | 00000011 | 00000000 | Registar AP se pomera ulevo. |
| | 00000011 | 00000000 | Ne vrši se nikakva akcija. |
| 6 | 00000110 | 00000000 | Registar AP se pomera ulevo. |
| | 00000110 | 00000000 | Ne vrši se nikakva akcija. |
| 7 | 00001100 | 00000000 | Registar AP se pomera ulevo. |
| | 00000011 | 00000001 | Vrši se oduzimanje $A = A - M$ i $P_0 = 1$. |
| 8 | 00000110 | 00000010 | Registar AP se pomera ulevo. |
| | 00000110 | 00000010 | Ne vrši se nikakva akcija. |

Ostatak deljenja je sadržan u registru A i iznosi 6. Vrednost registra P je 2. Kako je znak deljenika i delioca različit, prema algoritmu se za vrednost količnika uzima vrednost registra P sa promenjenim znakom, pa je količnik jednak -2 .