UVOD U ORGANIZACIJU I ARHITEKTURU RAČUNARA

Množenje i deljenje neoznačenih i označenih brojeva

Danijela Simić

Množenje neoznačenih brojeva

Proizvod se formira kao zbir delimicnih proizvoda. Ako je cifra množioca 1, delimicni proizvod je jednak množeniku, a ako je 0, onda je i delimični proizvod 0. Pocinje se od cifre najmanje težine. Svaki sledeci delimicni proizvod se pomera za po jedno mesto ulevo.

Primer:

1. $00001110 \cdot 00001001 = 0000000001111110$
2. $1001 \cdot 110011 = 111001011$

Hardverski se ovaj algoritam implementira preko serijskog množioca koji koristi tri registra A, M i P, kao i jednobitni registar C koji sadrži prenos pri sabiranju. Algoritam se može opisati na sledeći način:

1. Na početku množenja se množenik upisuje u registar M, množilac u registar P, dok se u registre A i C upisuje 0.
2. U svakom koraku množenja bit množioca na mestu najmanje težine odredjuje da li će u tom koraku množenik biti sabran sa tekućom vrednošću proizvoda:
   (a) Ako je vrednost bita 1 sabiranje se vrši.
   (b) Ako je vrednost bita 0 ne vrši se nikakva akcija.
3. Vrši se logičko pomeranje udesno sadržaja registara C, A i P, pri čemu se sva tri posmatraju kao jedinstven registar. Logičko pomeranje označava da prilikom pomeranja (šiftovanja) registara se na početku upisuju 0.
4. Korak 3 se ponavlja u ciklusu sve dok se ne obrade svi bitovi u množiocu.
5. Vrednost proizvoda je upisana u registrima A i P, posmatranim kao jedan registar.

Primeri:

1. $9 \cdot 51 = ?$
   $(9)_{10} = (1001)_2$
   $(51)_{10} = (110011)_2$

Korak	M	C	A	P	Komentar
	00001001	0	00000000	00110011	upisane vrednosti
1	00001001	0	00001001	00110011	A = A+M
	00001001	0	00000100	10011001	pomeranje udesno
2	00001001	0	00001101	10011001	A = A+M
	00001001	0	00000110	11001100	pomeranje udesno
3	00001001	0	00000011	01100110	pomeranje udesno
4	00001001	0	00000001	10110011	pomeranje udesno
5	00001001	0	00001010	10110011	A = A+M
	00001001	0	00000101	01011001	pomeranje udesno
6	00001001	0	00001110	01011001	A = A+M
	00001001	0	00000111	00101100	pomeranje udesno
7	00001001	0	00000011	10010110	pomeranje udesno
8	00001001	0	00000001	11001011	pomeranje udesno; kraj

Reyultat je $(0000000111001011)_2 = 459$

2. Zapisati 221 i 131 u 8-bita kao neoznacene binarne brojeve i izvršiti množenje $221 \cdot 131$.
$(221)_{10} = (11011101)_2$
$(131)_{10} = (10000011)_2$

Korak	M	C	A	P	Komentar
	11011101	0	00000000	10000011	početno stanje
1	11011101	0	11011101	10000011	A = A+M
	11011101	0	01101110	11000001	pomeranje udesno
2	11011101	1	01001011	11000001	A = A+ M
	11011101	0	10100101	11100000	pomeranje udesno
3	11011101	0	01010010	11110000	pomeranje udesno
4	11011101	0	00101001	01111000	pomeranje udesno
5	11011101	0	00010100	10111100	pomeranje udesno
6	11011101	0	00001010	01011110	pomeranje udesno
7	11011101	0	00000101	00101111	pomeranje udesno
8	11011101	0	11100010	00101111	A = A + M
	11011101	0	01110001	00010111	pomeranje udesno; kraj

Rezultat je $(0111000100010111)_2 = 28951$

3. Zapisati 201 i 137 u 8-bita kao neoznačene binarne brojeve i hardverski izvršiti množenje: $201 \cdot 137$.

$(201)_{10} = (11001001)_2$
$(137)_{10} = (10001001)_2$

Korak	M	C	A	P	Komentar
	11001001	0	00000000	10001001	početno stanje
1	11001001	0	11001001	10001001	A = A + M
	11001001	0	01100100	11000100	pomeranje udesno
2	11001001	0	00110010	01100010	pomeranje udesno
3	11001001	0	00011001	00110001	pomeranje udesno
4	11001001	0	11100010	00110001	A = A + M
	11001001	0	01110001	00011000	pomeranje udesno
5	11001001	0	00111000	10001100	pomeranje udesno
6	11001001	0	00011100	01000110	pomeranje udesno
7	11001001	0	00001110	00100011	pomeranje udesno
8	11001001	0	11010111	00100011	A = A + M
	11001001	0	01101011	10010001	pomeranje udesno; kraj

Rezultat je: $(0110101110010001)_2 = 27537$

Množenje brojeva u potpunom komplementu - Butov algoritam

Prethodno opisani postupak se ne može primeti na izračunavanje proizvoda dva broja zapisana u potpunom komplementu.

Problem nastaje kada pokušavamo množiti negativne broje. Jedno rešenje je da se oba činioca konvertuju u pozitivne brojeve, ali postoji i lakši i kraći način. Ispod je opisan postupak hardverskog množenja.

U kontekstu modernih procesora i ASIC-a (Application-Specific Integrated Circuits - specijalizovana integrisana kola), može se koristiti raznovrsne tehnike, zavisno o specifičnim zahtevima aplikacije. Ove tehnike mogu uključivati varijacije Butovog algoritma, paralelno množenje, množači korišćenjem stabala i druge. Izbor algoritma za množenje često zavisi o željenom odnosu između brzine, prostora i potrošnje energije.

Međutim, vredi napomenuti da, iako se Butov algoritam predaje na mnogim kursevima računarske arhitekture zbog njegove istorijske važnosti i ilustrativne vrednosti u razumevanju aritmetike, ali nije jedini metod koji se koristi u stvarnim hardverskim dizajnima.

U množenju se koriste četiri registra A, M, P i P_{-1} pri čcemu P_{-1} sadrži samo jedan bit. Rad Butovog algoritma se odvija na sledeći način:

1. Moženik i množilac se upisuju u registre M i P, a u registre A i P_{-1} upisuje se 0. Takođe, postavlja se vrednost brojača koji određuje broj ponavljanja koraka u telu algoritma se postavlja na n koje odgovara dužini registara u kojima se nalaze množenik i množilac.
2. U svakom koraku se porede vrednosti u bitu najmanje težine P_0 registra P i uvedenog dodatka P_{-1}:
   - Ako se bitovi razlikuju, i kombinacija je 01 tada se množenik sabere sa sadržajem registra A.
   - Ako se bitovi razlikuju, i kombinacija je 10 tada se množenik oduzme od sadržaja registru A.
   - Ako su te dve vrednosti jednake (11 ili 00) ne vrši se nikakva akcija.
   - Posle svakog od ova tri slučaja vrši se aritmetičko pomeranje udesno za jednu poziciju sadržaja A, P i P_{-1} pri čemu se oni posmatraju kao jedan registar. Aritmetičko pomeranje označava da prilikom pomeranja (šiftovanja) registara se na početku upisuju znak. Odnosno, upisuje se 0 ukoliko je vrednost počinjala 0 (ako je vrednost bila pozitivna), tj. 1 ako je vrednost u registru počinjala sa 1, tj. ako je bila negativna.
3. Od brojača se oduzme 1.
4. Prethodni korak se ponavlja u ciklusu sve dok ne budu obradjeni svi bitovi u množiocu, odnosno dok brojač ne postane 0.
5. Rezultat množenja je upisan u registre A i P, posmatrane kao jedna reč.

Primeri:

1. Prevesti u 8-bitne označene binarne brojeve u potpunom komplementu i izvršiti mno enje Butovim algoritmom: $103 \cdot (-13)$

   Resenje:
   $(103)_{10} = (01100111)_2^8$
   $(-13)_{10} = (11110011)_2^8$

Korak	M	A	P	P_{-1}	Komentar
	01100111	00000000	11110011	0	početno stanje
1	01100111	10011001	11110011	0	A = A-M
	01100111	11001100	11111001	1	pomeranje udesno
2	01100111	11100110	01111100	1	pomeranje udesno
3	01100111	01001101	01111100	1	A = A+M
	01100111	00100110	10111110	0	pomeranje udesno
4	01100111	00010011	01011111	0	pomeranje udesno
5	01100111	10101100	01011111	0	A = A-M
	01100111	11010110	00101111	1	pomeranje udesno
6	01100111	11101011	00010111	1	pomeranje udesno
7	01100111	11110101	10001011	1	pomeranje udesno
8	01100111	11111010	11000101	1	pomeranje udesno; kraj

Rešenje je $(1111101011000101)_2 = -1339$

2. Prevesti u 8-bitne označene binarne brojeve u potpunom komplementu i izvršiti množenje Butovim algoritmom: $62 \cdot (-47)$

Resenje:
$(62)_{10} = (00111110)_2^8$
$(-47)_{10} = (11010001)_2^8$

Korak	M	A	P	P_{-1}	Komentar
	00111110	00000000	11010001	0	početno stanje
1	00111110	11000010	11010001	0	A = A - M
	00111110	11100001	01101000	1	pomeranje udesno
2	00111110	00011111	01101000	1	A = A + M
	00111110	00001111	10110100	0	pomeranje udesno
3	00111110	00000111	11011010	0	pomeranje udesno
4	00111110	00000011	11101101	0	pomeranje udesno
5	00111110	11000101	11101101	0	A = A - M
	00111110	11100010	11110110	1	pomeranje udesno
6	00111110	00100000	11110110	1	A = A + M
	00111110	00010000	01111011	0	pomeranje udesno
7	00111110	11010010	01111011	0	A = A - M
	00111110	11101001	00111101	1	pomeranje udesno
8	00111110	11110100	10011110	1	pomeranje udesno; kraj

Rešenje je $(1111010010011110)_2 = -2914$

3. Zapisati brojeve 12 i -31 kao osmobitne označene brojeve a potom koristeći Butov algoritam izvršiti množenje $12 * (-31)$.

$(12)_{10} = (00001100)_2$
$(31)_{10} = (00011111)_2$
$(-31)_{10} = (11100001)_2$

Korak	M	A	P	P_{-1}	Komentar
	00001100	00000000	11100001	0	početno stanje
1	00001100	11110100	11100001	0	A = A-M
	00001100	11111010	01110000	1	siftovanje u desno
2	00001100	00000110	01110000	1	A = A+M
	00001100	00000011	00111000	0	siftovanje u desno
3	00001100	00000001	10011100	0	siftovanje u desno
4	00001100	00000000	11001110	0	siftovanje u desno
5	00001100	00000000	01100111	0	siftovanje u desno
6	00001100	11110100	01100111	0	A = A - M
	00001100	11111010	00110011	1	siftovanje u desno
7	00001100	11111101	00011001	1	siftovanje u desno
8	00001100	11111110	10001100	1	siftovanje u desno

Proizvod je $(1111101110101101)_2^{16}$ tj. $(-1107)_{10}$.

4. Prevesti u 8-bitne binarne brojeve i izvršiti množenje: $(-123) \cdot 9$.

$(-123)_{10} \rightarrow (10000101)_2^8$
$(9)_{10} \rightarrow (00001001)_2^8$

Korak	M	A	P	P_{-1}	Komentar
	10000101	00000000	00001001	0	početno stanje
1	10000101	01111011	00001001	0	A = A-M
	10000101	00111101	10000100	1	siftovanje u desno
2	10000101	11000010	10000100	1	A = A+M
	10000101	11100001	01000010	0	siftovanje u desno
3	10000101	11110000	10100001	0	siftovanje u desno
4	10000101	01101011	10100001	0	A = A-M
	10000101	00110101	11010000	1	siftovanje u desno
5	10000101	10111010	11010000	1	A = A+M
	10000101	11011101	01101000	0	siftovanje u desno
6	10000101	11101110	10110100	0	siftovanje u desno
7	10000101	11110111	01011010	0	siftovanje u desno
8	10000101	11111011	10101101	0	siftovanje u desno

Proizvod je $(1111101110101101)_2^{16}$ tj. $(-1107)_{10}$.

Modifikovan Butov algoritam

Operacija množenja izvodi se u više koraka:

1. Formira se *Butov kodirani množilac* tako što se pretraži originalni množilac zdesna ulevo i upiše -1 na svakoj poziciji gde je 1 na početku niske, $+1$ kada se naiđe na prvu sledeću 0, dok se na ostalim mestima upiše 0. Iz kodiranog množioca se vidi uspešnost optimizacije (pojava -1 znači da se primenjuje oduzimanje, $+1$ primenjuje se sabiranje, a 0 da nema akcije).

Primer:
Za množioce $+9$, -9 i $+85$ kodirani množioci su:

0	0	0	0	1	0	0	1	Množilac $(+9)_{10}$
0	0	0	+1	-1	0	+1	-1	Kodirani množilac
1	1	1	1	0	1	1	1	Množilac $(-9)_{10}$
0	0	0	-1	+1	0	0	-1	Kodirani množilac
1 0	1	0	1	0	1	0	1	Množilac $(+85)_{10}$
+1	-1	+1	-1	+1	-1	+1	-1	Kodirani množilac

2. Izdvojiti parove oblika $(a_{2k+1}; b_{2k})$ iz kodiranog množioca, gde su u indeksu označene pozicije na kojima se nalaze vrednosti a i b, pri čemu važi $k \in [0, \frac{n}{2} - 1]$. Težina a_{2k+1} je dvostruko veća od težine b_{2k}. Na osnovu težine se određuje zajednička vrednost svakog para. Na primer, zajednička vrednost para $(+1, -1) = +2 + (-1) = 1$, dok je vrednost para $(0, +1) = 2 \cdot 0 + (+1) = 1$.

3. Za svaki par koji se pojavi u kodiranom množiocu izvrši se pomeranje množenika za $2k$ mesta ulevo. Tako dobijena vrednost se množi sa vrednošću para i dodaje proizvodu. U implementaciji algoritma se određivanje Butovog kodiranog množioca i vrednosti njegovih parova spaja u jedan korak na osnovu trobitne kombinacije bitova.

PRIMERI:

1. Zapisati brojeve -123 i -19 kao osmobitne označene brojeve u potpunom komlementu, a potom koresteći modifikovan Butov algoritam izvršiti množenje -123*(-19) = 2337.

   $(-19)_{10} = (11101101)_2$
   $(-123)_{10} = (10000101)_2$

   parovi:
   (+1, -1) = 1; (0, -1) = -1; (-1, +1) = -1; (0, 0) = 0;

   zbir:

   ```
   1111111110000101
   0000000111101100
   0000011110110000
   0000000000000000
   ------------------
   0000100100100001
   ```

2. Zapisati brojeve 12 i -45 kao osmobitne označene brojeve u potpunom komlementu, a potom koresteći modifikovan Butov algoritam izvršiti množenje $12 * (-45) = -540$.

   $(12)_{10} = (00001100)_2$
   $(-45)_{10} = (11010011)_2$

   parovi:
   (0, -1) = -1; (0, +1) = 1; (+1, -1) = 1; (0, -1) = -1;

   zbir:

   ```
   1111111111110100
   0000000000110000
   0000000011000000
   1111110100000000
   ----------------
   1111110111100100
   ```

3. Zapisati brojeve 62 i -47 kao osmobitne označene brojeve u potpunom komlementu, a potom koresteći modifikovan Butov algoritam izvršiti množenje $62 * (-47) = -2914$.

   $(62)_{10} = (00111110)_2$
   $(-42)_{10} = (11010001)_2$

   parovi: (+1, -1) = 1; (0, 0) = 0; (+1, -1) = 1; (0, -1) = -1;

   zbir:

```
0000000000111110
0000000000000000
0000001111100000
1111000010000000
----------------
1111010010011110
```

4. Zapisati brojeve -51 i -73 kao osmobitne označene brojeve u potpunom komlementu, a potom koresteći modifikovan Butov algoritam izvršiti množenje $(-51) * (-73) = 3723$.

$(-51)_{10} = (11001101)_2$
$(-81)_{10} = (10110111)_2$

parovi:
(0, -1) = -1; (+1, 0) = 2; (0, -1) = -1; (-1, +1) = -1

zbir:

```
0000000000110011
1111111001101000
0000001100110000
0000110011000000
----------------
0000111010001011
```

Deljenje neoznačenih brojeva

Primeri:

1. $110101/101 = 1010$ i ostatak 11
2. $1000110101/1011 = 110011$ i ostatak 100

Hardverska implementacija koristi tri registra P, A i M i brojač čija je inicijalna vrednost jednaka broju bitova u registrima. Inicijalno se u registar P upisuje deljenik, u registar M delilac, a u registar A nula. Deljenje se vrši na sledeći način:

1. U prvom koraku se pomera se sadržaj registara A i P (posmatranih kao jedna binarna reč) ulevo za jedan bit.
2. Sadržaj registra M se oduzima od sadržaja registra A i dobijena vrednost se upisuje u registar A.
3. Ispituje se da li je vrednost u registru $A \geq 0$, tj. da li može da se izvrši oduzimanje i dobije delimični ostatak. Ako može, tada se u bit najmanje težine registra P upisuje 1. U suprotnom se u bit najmanje težine registra P upisuje 0 i sadržaj registra M se sabira sa sadržajem registra A radi restauracije prethodnog sadržaja registra A.
4. Vrednost brojača se smanjuje za 1; ukoliko je vrednost brojača veća od nule, izvršavanje se vraća na korak 1.
5. Na kraju deljenja količnik se nalazi u registru P, ostatak u registru A. Dužina registra određuje broj koraka.

Primeri:

1. Prevesti u 8-bitne neoznačene binarne brojeve i izvršiti deljenje: $219/3$.
   $(219)_{10} = (11011011)_2^8$
   $(3)_{10} = (00000011)_2^8$

Korak	M	A	P	Komentar
	00000011	00000000	11011011	početno stanje
1	00000011	00000001	10110110	pomeranje u levo
	00000011	11111110	10110110	A = A-M, neuspešno
	00000011	00000001	10110110	restauracija sadržaja A
2	00000011	00000011	01101100	pomeranje u levo
	00000011	00000000	01101100	A = A - M; uspešno
	00000011	00000000	01101101	bit najmanje težine registra P postavlja na 1
3	00000011	00000000	11011010	pomeranje u levo
	00000011	11111101	11011010	A = A-M; neuspešno;
	00000011	00000000	11011010	restauracija sadržaja A
4	00000011	00000001	10110100	pomeranje u levo
	00000011	11111110	10110100	A = A-M; neuspešno
	00000011	00000001	10110100	restauracija sadržaja A
5	00000011	00000011	01101000	pomeranje u levo
	00000011	00000000	01101000	A = A-M; uspešno
	00000011	00000000	01101001	bit najmanje težine registra P postavlja na 1
6	00000011	00000000	11010010	pomeranje ulevo
	00000011	11111101	11010010	A = A-M; neuspešno
	00000011	00000000	11010010	restauracija sadržaja A
7	00000011	00000001	10100100	pomeranje ulevo
	00000011	11111110	10100100	A = A-M; neuspešno
	00000011	00000001	10100100	restauracija sadržaja A
8	00000011	00000011	01001000	pomeranje ulevo
	00000011	00000000	01001000	A = A+M; neuspešno
	00000011	00000000	01001001	bit najmanje težine registra P postavlja na 1

Količnik je $P = (01001001)_2 = 73$, a ostatak $A = (00000000)_2 = 0$.

Deljenje brojeva zapisanih u potpunom komplementu

Deljenje označenih brojeva zapisanih u potpunom komplementu odvija se po sledećem algoritmu:

1. Na početku se u registre A i P upisuje deljenik posmatran kao broj u potpunom komplementu dužine $2n$. Delilac se upisuje u registar M.

2. Sadržaj registara A i P se pomera (aritmetičkim pomeranjem) za jedno mesto u levo. Ako M i A imaju isti znak tada se izvršava operacija oduzimanja: $A = A - M$; u suprotnom se izvršava sabiranje $A = A + M$. Izvršena operacija se smatra uspešnom ako je znak registra A nepromenjen posle njenog izvršavanja.

3. - Ako je operacija uspešna ili ($A = 0$ i stiglo se do poslednjeg koraka) tada se bit najmanje težine registra P postavlja na 1.
   - U suprotnom (ako je operacija neuspešna ili $A \neq 0$ i nije se stiglo do poslednjeg koaraka) bit najmanje težine registra P se postavlja na 0 i restaurira se prethodna vrednost registra A.

4. Prethodna dva koraka se ponavljaju n puta, gde je n dužina registra P. Na kraju procesa registar A sadrži ostatak. Ako je znak deljenika i delioca isti tada je vrednost količnika zapisana u registru P. U suprotnom, za vrednost količnika treba uzeti vrednost u registru P sa promenjenim znakom.

Primeri:

1. Prevesti u 8-bitne označene binarne brojeve i izvršiti deljenje: $103/(-7)$.
   $(103)_{10} = (01100111)_2^8$
   $(-7)_{10} = (11111001)_2^8$

Korak	M	A	P	Komentar
	11111001	00000000	01100111	početno stanje
1	11111001	00000000	11001110	pomeranje u levo
	11111001	11111001	11001110	A = A+M, neuspešno
	11111001	00000000	11001110	restauracija sadržaja A
2	11111001	00000001	10011100	pomeranje u levo
	11111001	11111010	10011100	A = A + M. neuspešno
	11111001	00000001	10011100	restauracija sadržaja A
3	11111001	00000011	00111000	pomeranje u levo
	11111001	11111100	00111000	A = A+M; neuspešno;
	11111001	00000011	00111000	restauracija sadržaja A
4	11111001	00000110	01110000	pomeranje u levo
	11111001	11111111	01110000	A = A+M; neuspešno
	11111001	00000110	01110000	restauracija sadržaja A
5	11111001	00001100	11100000	pomeranje u levo
	11111001	00000101	11100000	A = A+M; uspešno
	11111001	00000101	11100001	bit najmanje težine registra P postavlja na 1
6	11111001	00001011	11000010	pomeranje ulevo
	11111001	00000100	11000010	A = A+M; uspešno
	11111001	00000100	11000011	bit najmanje težine registra P postavlja na 1
7	11111001	00001001	10000110	pomeranje ulevo
	11111001	00000010	10000110	A = A+M; uspešno
	11111001	00000010	10000111	bit najmanje težine registra P postavlja na 1
8	11111001	00000101	00001110	pomeranje ulevo
	11111001	11111110	00001110	A = A+M; neuspešno
	11111001	000000101	0001110	restauracija sadržaja A

Znak deljenika i delioca nije isti $\longrightarrow$ rezultat je $-P$.

Količnik je $(11110010)_2$, tj. -14, a ostatak $(00000101)_2 = 5$.

2. Prevesti u 8-bitne označene binarne brojeve i izvršiti deljenje: $(-154)/17$.
$(-154)_{10} = (101100110)_2^8$
$(17)_{10} = (00010001)_2^8$

Korak	M	A	P	Komentar
	00010001	11111111	01100110	početno stanje
1	00010001	11111110	11001100	pomeranje u levo
	00010001	00001111	11001100	A = A+M; neuspešno
	00010001	11111110	11001100	restauracija sadržaja A
2	00010001	11111101	10011000	pomeranje u levo
	00010001	00001110	10011000	A = A+M; neuspešno
	00010001	11111101	10011000	restauracija sadržaja A
3	00010001	11111011	00110000	pomeranje u levo
	00010001	00001100	00110000	A = A+M; neuspešno
	00010001	11111011	00110000	restauracija sadržaja A
4	00010001	11110110	01100000	pomeranje u levo
	00010001	00000111	01100000	A = A+M; neuspešno
	00010001	11110110	01100000	restauracija sadržaja A
5	00010001	11101100	11000000	pomeranje u levo
	00010001	11111101	11000000	A = A+M; uspešno
	00010001	11111101	11000001	bit najmanje težine registra P postavlja na 1
6	00010001	11111011	10000010	pomeranje ulevo
	00010001	00001100	10000010	A = A+M; neuspešno
	00010001	11111011	10000010	restauracija sadržaja A
7	00010001	11110111	00000100	pomeranje u levo
	00010001	00001000	00000100	A = A+M; neuspešno
	00010001	11110111	00000100	restauracija sadržaja A
8	00010001	11101110	00001000	pomeranje u levo
	00010001	11111111	00001000	A = A+M; uspešno
	00010001	11111111	00001001	bit najmanje težine registra P postavlja na 1

Znak deljenika i delioca nije isti $\longrightarrow$ rezultat je $-P$.

Količnik je $(11110111)_2$, tj. -9, a ostatk je $(11111111)_2$, tj. -1.

3. Izvršiti računsku operaciju: $(-148)/17$.

$(-148)_{10} = (101101100)_2$
$(17)_{10} = (00010001)_2$

Korak	M	A	P	Komentar
0	00010001	11111111	01101100	početno stanje
1	00010001	11111110	11011000	pomeranje ulevo
	00010001	00001111	11001100	A = A + M
	00010001	11111110	11011000	restauracija sadržaja A
2	00010001	11111101	10110000	pomeranje ulevo
	00010001	00001110	10011000	A = A + M
	00010001	11111101	10110000	restauracija sadržaja A
3	00010001	11111011	01100000	pomeranje ulevo
	00010001	00001100	00110000	A = A + M
	00010001	11111011	01100000	restauracija sadržaja A
4	00010001	11110110	11000000	pomeranje ulevo
	00010001	00000111	01100000	A = A + M
	00010001	11110110	11000000	restauracija sadržaja A
5	00010001	11101101	10000000	pomeranje ulevo
	00010001	11111110	10000000	A = A + M
	00010001	11111110	10000001	postavljanje najmanjeg bita u P
6	00010001	11111101	00000010	pomeranje ulevo
	00010001	00001100	00000010	A = A + M
	00010001	11111101	00000010	restauracija sadržaja A
7	00010001	11111010	00000100	pomeranje ulevo
	00010001	00001011	00000100	A = A + M
	00010001	11111010	00000100	restauracija sadržaja A
8	00010001	11110100	00001000	pomeranje ulevo
	00010001	00000101	00001000	A = A + M
	00010001	11110100	00001000	restauracija sadržaja A

Znak deljenika i delioca nije isti $\longrightarrow$ rezultat je $-P$.

Količnik je $-P = -(00001000)_2 = -8$, a ostatak: $(11110100)_2 = -12$