Digitalni zapis podataka

Stefan Mišković

3 Množenje celih brojeva

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, činioci 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 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 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 $(103)_{10} = (01100111)^8_2$ i $(-13)_{10} = (11110011)^8_2$. 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

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

Pokažimo na jednostavnom primeru zašto Butov algoritam funkcioniše. Pretpostavimo najpre da je dat proizvoljan množenik M i pozitivan množilac $P = 00011110$, koji se sastoji od jedne grupe uzastopnih jedinica. Imajući u vidu da važi identitet

$$2^n + 2^{n-1} + \cdots + 2^{n-k} = 2^{n+1} - 2^{n-k},$$

sledi da je

$$M \times P = M \times (00011110)_2 = M \times (2^4 + 2^3 + 2^2 + 2^1) = M \times (2^5 - 2^1).$$

Primetimo da ovo odgovara jednom oduzimanju kada se naiđe na par bitova 10 i sabiranju kada se naiđe na par bitova 01. Pretpostavimo sada da množilac ima dva para uzastopnih jedinica, i neka je $P = 01111010$. Isti identitet sve može primeniti dva puta, zbog čega važi

$$M \times P = M \times (01111010) = M \times (2^6 + 2^5 + 2^4 + 2^3 + 2^1) = M \times (2^7 - 2^3 + 2^2 - 2^1).$$

Slično, ovde bi se u algoritmu dva puta izvršilo sabiranje (nailaženjem na par bitova 01) i dva puta oduzimanje (nailaženjem na par bitova 10). Ovo se može uopštiti na množilac koji ima proizvoljan broj uzastopnih jedinica. Situacija je slična i kada je množilac P negativan broj u potpunom komplementu.