

Dodatni zadatak

Implementacija algoritma množenja dva binarna broja

Potrebno je implementirati jednostavan algoritam množenja dva binarna broja pri čemu je prvi broj 16-bitni, a drugi 8-bitni i potom na osnovu toga izračunati faktorijel nekog od prvih 8 prirodnih brojeva.

Uvod

Radi ilustracije algoritma množenja poći ćemo od množenja dva decimalna broja:

$$\begin{array}{r} 456 \\ \times 123 \\ \hline 1368 \\ 9120 \\ +45600 \\ \hline 56088 \end{array} \quad \begin{array}{l} = 456 \times 3 \times 10^0 \\ = 456 \times 2 \times 10^1 \\ = 456 \times 1 \times 10^2 \end{array}$$

Iz prethodnog primera je očigledno da se množenje dva decimalna broja svodi na sumiranje delimičnih proizvoda koji se dobijaju množenjem prvog činioca sa ciframa drugog pomnoženog težinskim faktorom.

Slično imamo i u slučaju binarnog množenja:

$$\begin{array}{r} 1100 \quad (X) \\ \times 0110 \quad (Y) \\ \hline 0000 \\ 11000 \\ 110000 \\ +0000000 \\ \hline 1001000 \end{array} \quad \begin{array}{l} = 1100 \times 0 \times 2^0 \\ = 1100 \times 1 \times 2^1 \\ = 1100 \times 1 \times 2^2 \\ = 1100 \times 0 \times 2^3 \end{array}$$

Ukoliko prvi činioc označimo sa X, a drugi sa Y, može se uočiti da se u suštini X pomera u levo za jedan bit i dodaje ukupnoj sumi ukoliko je odgovarajući bit činioca Y jednak jedinici. Ukoliko je taj bit jednak nuli sabira se nula, dakle nema uticaja. Na osnovu toga se može zaključiti da je za potrebe implementacije algoritma binarnog množenja potrebno sledeće:

- određivanje da li je neki bit činioca Y jednak jedinici ili nuli,
- ukoliko je bit jednak nuli potrebno je preskočiti dodavanje pomerenog činioca X u konačnu sumu,
- ukoliko je bit jednak jedinici, potrebno je izvršiti pomeranje činioca X za odgovarajući broj bita u *levo* i dodati ga krajnjoj sumi.

Na osnovu gornje ilustracije binarnog množenja se takođe može zaključiti da je pomeranje binarnog broja ulevo za jedan, ekvivalentno operaciji množenja sa 2, što je potrebno koristiti jer ne postoji instrukcija za pomeranje sadržaja registra u levo.

Na osnovu svega navedenog se može napraviti algoritam binarnog množenja koji je prikazan na narednoj slici.

Zadatak

- 1) Realizovati algoritam množenja dva binarna broja (jedan 16-bitni, a drugi 8-bitni) čije su vrednosti zadate u DATA delu programa. Uzimajući u obzir da je prvi broj 16-bitni potrebno je rezervisati 2 memorijske lokacije za viših 8 bita i nižih 8 bita. Rezultat množenja smestiti u 2 sukcesivne memorijske lokacije (za viših 8 i nižih 8 bita). Obratiti pažnju na bit prenosa prilikom sabiranja nižih 8 bita! **Algoritam množenja realizovati kao proceduru.**
- 2) Koristeći proceduru u kojoj je implementiran algoritam množenja binarnih brojeva, napraviti glavni program koji izračunava faktorijel nekog od prvih 8 prirodnih brojeva, pri čemu se broj čiji se faktorijel traži nalazi u memorijskoj lokaciji rezervisanoj u DATA delu. Krajnji rezultat (**i viših i nižih 8 bita**) postaviti na stek.

Sledi predlog izgleda DATA dela programa:

```
DATA
    msb_broj1    db 0
    lsb_broj1    db 0
    broj2        db 0
    msb_rez      db 0
    lsb_rez      db 0
    maska        db 1
    broj         db 4
ENDDATA
```

Dakle rezervisano je 7 memorijskih lokacija - dve za prvi broj, jedna za drugi broj, dve za rezultat, jedna za masku i jedna za broj čiji se faktorijel traži(npr. 4).

