

## ARITMETIKA S PLAVAJOČO VEJICO

Zadnje poglavje je namenjeno predvsem tistim, ki nameravate uporabljati strojne programe za reševanje matematičnih in tehničnih nalog. Vseeno najtopleje priporočam, da ga vsaj preletite, tudi če vaši nameni niso takšni.

Gotovo ste opazili, da smo se v knjigi ves čas ukvarjali le s celimi števili. Seveda je računalnik kos tudi realnim - navadno pravimo "decimalnim" - številom. V računalništvu imenujemo takšno aritmetiko aritmetiko s plavajočo vejico. Ukazi, ki smo jih obravnavali v prejšnjih poglavjih, so ukazi procesorja Z80. Zato so enaki za vse računalnike, ki imajo ta CP. Pri delu s plavajočo vejico pa uporabljamo izključno podprograme v ROM-u. Stvari, s katerimi se boste seznanili zdaj, veljajo zato (razen ukaza RST 28H) le za Sinclairjev Spectrum.

Oglejmo si najprej, kako so realna števila zapisana v pomnilniku. Vsako število (razen 0) lahko zapišemo v obliki

$$x = m * 2^n$$

m imenujemo mantisa, n pa eksponent. V Spectrumu zavzema število v plavajoči vejici 5 zlogov. Prvi zlog je namenjen eksponentu, natančneje: vsoti  $128+n$ . V ostalih štirih zlogih je zložena mantisa v dvojiški obliki. Mantisa mora imeti vrednost med  $1/2$  in 1 (lahko je  $1/2$ , ne more pa biti 1). Ker je manjša od 1, ima dvojiško vejico:

1/2 napišemo dvojiško 0.1  
 1/4 " " 0.01  
 3/4 " " 0.11 itd.

Ker mora biti mantisa vedno večja ali vsaj enaka  $1/2$ , je prvi bit drugega zloga vedno 1. Ker se tega zavedamo, uporabljamo ta bit raje za označevanje predznaka - 0 pomeni pozitivno, 1 negativno število. Poskusimo zdaj po teh pravilih zapisati 0.3125.

$$0.3125 = 5/16$$

$$5/16 = m * 2^n$$

$$5/16 = 5/8 * 2^1(-1)$$

$$5/8 = 1/2 + 1/8$$

Prvi zlog =  $128 + \text{eksponent} = 128 + (-1) = 127 = 7FH$   
 Drugi zlog =  $00100000 = 20H$ . Zlogi 3, 4 in 5 so 0. Prvi bit drugega zloga je enak 0, ker je število pozitivno. 0.3125 je v plavajoči vejici enako  $7F200000$ .

Števila v petih zlogih ne moremo, kot smo to počeli do zdaj, spraviti v en register ali register ali register par. Zato realna števila upravljamo v registerski "peterček" AEDCB. S tem porabimo vse registre. Ker bomo gotovo potrebovali več kot eno realno število, moramo imeti na voljo še kakšno mesto za shranjevanje. V RAM-u je v ta namen rezerviran prostor - računski sklad.

Računski sklad je zelo podoben strojnemu skladu, ki ga že dobro poznate. V nekaterih pomembnih podrobnostih pa se oba sklada razlikujeta:

- v pomnilniku se nahajata na različnih mestih;
- računski sklad "raste" navzgor, ne navzdol;
- vsaka vrednost na računskem skladu zavzema 5 zlogov in ne le 2 kot na strojnem;
- računski sklad lahko shranjuje poleg števil tudi znake ter - ukazi CALL in RET nanj ne vplivajo.

Ukaz RST 28H v Spectrumu označuje, da želimo izvesti operacijo - eno ali več - s števili na računskem skladu. Števila, ki sledijo ukazu (kodi operacij), določajo za kakšne operacije gre. Niz teh kodov se mora vedno končati z 38H, ki označuje, da spet sledi strojni ukaz. Med izvajanjem ukaza RST 28H sname CP operande z računskega sklada, izvede zahtevano operacijo ter rezultat porine nazaj na sklad. Primer: zaporedje RST 28 04 38 bo snelo s sklada zgornji dve vrednosti (sklad bo tako za dve števili krajši), ju pomnožilo (kod operacije 04) ter dobljeno vrednost porinilo na sklad. V nizu je seveda lahko tudi več kodov. Koda operacij in njihovi pomeni so zbrani v tabeli na naslednji strani. Omejili smo se le na pogostejše uporabljane in laže razumljive, ker bi bila za ostale potrebna obširnejša razlaga. A videli boste, da vam ukaz RST 28 omogoča več kot le preprosto aritmetiko.

## UPORABA PODPROGRAMOV IN RAČUNSKEGA POMNILNIKA

Preden začnemo računati, moramo nekako poriniti ustrezna števila na računski sklad. Najpreprostejši način za to je uporaba treh podprogramov v ROM-u: