

Kot vidite, smo v programu označili naslov "Zanka" in nato zapisali ukaz "JR NZ, Zanka". To je običajni način zapisa v zbirni obliki in olajšuje prebiranje ter razumevanje programa. Še vedno pa morate v ukazu navesti odmik (in ne naslova, kot bi rekli na prvi pogled). Ob tem še uporaben nasvet, ki vam bo olajšal izračunavanje odkikov v krajših programih. Pri skoku nazaj začnite pri *drugi* zlogu ukaza JR šteti od 255 (= FFH) nazaj. Odmik je vrednost, ki jo boste dobili, ko boste prišli do zloga, na katerem naj se skok konča. Preizkusite to pravilo v našem primeru (kjer smo dobili vrednost odmika FCH = 252). Pri skoku naprej začnete pri *prvi* zlogu naslednjega ukaza šteti od 0. Ko dospete do naslovnega zloga, imate izračunano vrednost odmika.

Procesor Z80 pozna poseben ukaz, ki opravi delo zadnjih dveh vrstic prejšnjega programa. To je ukaz

```
DJNZ dis ,
```

ki ga preberemo "zmanjšaj B in skoči, če ničelna zastavica ni dvignjena, za odmik dis" (angleško DECREASE AND JUMP IF NOT ZERO). Ukaz zaseda dva zloga in nam torej prihrani en zlog v primerjavi z našim prejšnjim zapisom. Zaradi tega ukaza se register B običajno uporablja kot števec. Omejitev ukaza DJNZ (tako kot vseh relativnih skokov) je v številu do 255. Seveda pa lahko ukaze DJNZ "vgnezdimo", če je potrebno:

```
LD B, 10H      ;B = 16
Zanka1 PUSH BC ;spravi vrednost B na sklad
LD B, 0        ;s tem pravzaprav postavimo B na 256
Zanka2 -----
          -----
          ;vmesni izračuni
          ;opravljeno 256-krat ?
          ;vrni prvotno vrednost B
          ;ponovi večjo zanko 16-krat.
```

Seveda raba ukaza DJNZ v tem primeru ni obvezna. Prav lahko bi dali paru BC vrednost 1000H ter dodali nekaj ukazov, ki bi zmanjševali BC in testirali, ali je že enak 0.

## ČAKALNE ZANKE

Včasih želimo v programu iz tega ali onega razloga narediti kratek odmor. V ta namen uporabljamo čakalne zanke, ki jih napravimo s pomočjo ukaza DJNZ:

```
LD B, n
```

```
Čakaj DJNZ Čakaj .
```

Ukaz DJNZ Čakaj bo vrnil CP na samega sebe tolikokrat, da bo vrednost B postala 0. Šele potem bo nadaljeval z izvajanjem programa. To je obenem odgovor na naš OREH (str. 75), kaj se zgodi ob ukazu

```
Čakaj JR Čakaj.
```

Te zanke CP ne bo nikoli zapustil. Morali mu boste pomagati in izključiti računalnik. Najdaljši premor, ki ga povzroči enkratna uporaba čakalne zanke z ukazom DJNZ, traja nekaj manj kot eno tisočinko sekunde (natančneje: 952 mikrosekund).

Ker smo ravno spregovorili o "odmorih" v programu, naj omenimo še ukaz, ki sicer ne sodi v to poglavje. To je

```
HALT ,
```

ki prekine izvajanje strojnega programa do naslednje prekinitve (angleško interrupt). Njegov učinek je enak ukazu PAUSE 1 v basicu. A bodite previdni! Če so prekinitve onemogočene (ukaz DI), bo ukaz HALT nepreklicno ustavil izvajanje – in spet bo treba izvleči vtič iz vtičnice.

UPORABA: Pripravljeni program je zasnovan tako, da boste z njim lahko preizkusili dolge in kratke skoke. Spreminjati bo treba četrto (za kratke skoke) ali četrto in peto vrstico (za dolge skoke). Potrebne kode boste našli v tabeli na koncu knjige. V programu spreminjajte vrednost v registru A (prva vrstica), tako boste vplivali na zastavice. Kadar bo CP izvedel skok, bo rezultat programa 1, sicer pa 0.

```
3EXX      LD A, XX
BB        CP B
010000    LD BC, 0000
2002      JR NZ, Skok
00        NOP
```