

zbirnem zapisu, da z njima označimo podatke. Če pogledate v tabele, boste videli, da je 5A kod znaka "Z", 64 znaka "d", 72 znaka "r" itd. Z DEFB 00 smo označili konec našega "besedila" - se pravi konec podatkov, ki naj jih uporabi S\_PRINT. Seveda CP podatkov ne sme obravnavati kot strojne ukaze. Zato moramo program napisati tako, da se kaj takega ne zgodi. Spomnite se, kaj se dogaja ob klicu strojnega podprograma (o tem smo govorili v poglavju "Podprogrami"). Ker beseda "Zdravo" sledi ukazu CALL, bo na skladu spravljen naslov znaka "Z". Ta naslov lahko snamemo z ukazom POP - npr. POP HL. Če zdaj povečamo HL in napravimo PUSH, se bo CP ob koncu podprograma vrnil na naslednji naslov, tj. na naslov znaka "d". V našem podprogramu se torej lahko ognemo izvajanju podatkov tako, da ponavljamo POP in PUSH dokler ne dosežemo konca podatkov, tj. zloga z vrednostjo 0 (ki nikoli ne more biti del besedila). Pazljivo si oglejte tale podprogram:

```
E1 S_PRINT POP HL
7E LD A, (HL)
23 INC HL
E5 PUSH HL
A7 AND A
C8 RET Z
D7 RST 10
18F7 JR S_PRINT
```

Prve štiri vrstice pogledajo znak, spravljen na trenutnem naslovu za povratek. Ob tem tudi pomaknejo ta naslov za en zlog naprej (INC HL). Naslednji dve vrstici nadzorujeta, ali je že konec podatkov. Zapomniti si velja uporabo ukaza AND A za primerjanje A z 0 (s tem prihranimo 1 zlog). Če ima A vrednost 0 (tj. če smo prišli do konca podatkov), bo CP izvedel ukaz RET Z in se vrnil na zlog, ki sledi zlogu 00. Če A še ni 0, bo CP uporabil ukaz RST 10 in izpisal ustrezní znak na zaslon. Postopek se ponavlja, dokler ni konec besedila.

Zdaj pride na vrsto naš urejevalnik. Za start uporabimo 28672 (7000H) in najprej vpišite podprogram S\_PRINT, takoj za njim pa še nekoliko spremenjen glavni program:

```
AF ST XOR A
323C5C LD (TV FLAG), A
```

```
CD0070 CALL S_PRINT
1106100212014B646F72207669736F6B6F206C6574612C
206E697A6B6F2070616465 DEFM "besedilo"
00 DEFB 00
C9 RET
```

"START ZA USR" naj bo ST (= 28681). Poženite program! Za začetek kar lepo, ali ne?

#### RAZISKOVANJE ZASLONA

Lotimo se naše naloge še z druge strani. Preizkusimo, kaj se zgodi, če v basicu zapišemo:

```
FOR I= 16384 TO 23295: FOKE I, 41: NEXT I: PAUSE 0
```

Se vam zdi nenavadno? Ne le to, da smo vplivali na ekran, ampak predvsem kako se je to dogajalo. Vzrok je predstavitev zaslonu, kakor jo v svojem pomnilniku hrani Spectrum - in ki se zdi bolj zapletena, kot bi bilo potrebno. Zaradi tega je koristno, da zaslon nekoliko raziščemo. Ugotovili bomo, da je njegova predstavitelj v pomnilniku veliko bolj smiselna, kot je videti na prvi pogled.

Oglejmo si sliko 1. Iz nje razberemo naslov kateregakoli "PRINT" položaja na ekranu. Prav tako lahko iz nje razberemo naslove zlogov, ki določajo prilastke (barvo papirja, črnila, svetlost in utripanje) posameznih PRINT položajev (več o prilastkih na strani 116). Poskusimo s pomočjo slike določiti naslov mesta, na katerega bi pisali z ukazom PRINT AT 5,4 v basicu. Prva dva znaka naslova sta 40. Tretji znak je A (ker uporabljamo šesto vrstico), četrti pa 4. Tako pridemo do naslova 40A4. Naslov prilastkov dobimo tako, da za prvi dve številki vzamemo znaka v oklepaju. V našem primeru bi bil torej naslov zloga s prilastki 58A4. Zdaj lahko preizkusimo dva programčka:

```
21A440 LD HL, 40A4
36FF LD (HL), FF
C9 RET
```

```
21A458 LD HL, 58A4
36C7 LD (HL), C7
C9 RET
```