

H = 01, L = 02

naslov	vsebina	naslov	vsebina
30000	00	30000	02
30001	00	30001	01
30002	00	30002	00

V pomnilniku (in izpisih programov) je torej nizki zlog zapisan prvi. Za to spremembo ni prave razlage ampak se moramo z njo kar sprijazniti.

Prosim, prepričajte se, ali vam je zamenjava načinov poploma jasna. Zelo verjetno je, da bo to edini zelo pogosti vir napak v vaših programih. V registrskih parih je prvi spravljen visoki zlog, v pomnilniku in izpisih pa je prvi spravljen nizki zlog. Tega nikakor ne bi smeli lahkomišlno preskočiti. Vsakič, ko boste imeli v strojnem kodu opravka s 16-bitnimi števili, boste morali pazljivo razmisliti o zaporedju visokih in nizkih zlogov.

A ne dajte se prestrašiti - življenje z Z80 bi bilo brez 16-bitnih ukazov skoraj nemogoče. In to je cena, ki jo moramo plačati.

### NALAGANJE 16-BITNIH ŠTEVIL

Najpreprostejši ukaz v tej skupini je polnjenje registrskega para s 16-bitnim številom; splošna krajšava zanj je

LD rr, nn.

Spet uporabljamo zapis z dvema črkama za označevanje 16-bitnih vrednosti. Znak rr pomeni katerikoli registrski par, nn pa katerikoli število.

Če nimate zbirnika - in boste prevajali krajšave v strojni kod s pomočjo tabel na koncu knjige - potem postaja pravilo o zaporedju zlogov za vas zelo pomembno. Celó če imate zbirnik, bi se morali dobro zavedati spremembe zapisa. Le tako boste brez težav prebrali zbirnikov izpis ali "brali" strojni kod s pomočjo ukaza PEEK v basicu.

Poglejmo primer "napolni HL z 258". Krajšava za to je LD HL, 0102H. Kod ukaza LD HL, nn je 21 XX XX. Namesto XX XX moramo vstaviti število - v našem primeru 0102H. Vendar tega ne

zapišemo 21 01 02, ampak je pravilna oblika 21 02 01. V zbirni obliki bo napisano:

21 02 01 LD HL, 0102H.

Z vnašanjem programov iz te knjige torej ne boste imeli težav. Vseeno vam mora postati to domače, da boste lastne programe pisali brez težav.

Prav tako kot v registrske pare naložimo 16-bitna števila v indeksna registra (ki sta oba nogi s po 16 prsti kot se spominjate):

LD IX, nn  
LD IY, nn

Podatke lahko prenašamo tudi iz registrskih parov v dve zaporedni pomnilniški celici in obratno. Splošne krajšave so

LD (nn), rr LD rr, (nn)  
LD (nn), IX LD IX, (nn)  
LD (nn), IY LD IY, (nn)

Kot vedno pomenijo oklepaji "vsebino". Zadnji ukaz bi npr. prebrali "napolni pomnilniško celico nn z vrednostjo registra IY".

Pravkar omenjeni ukazi vedno vplivajo na dve 8-bitni pomnilniški celici (delamo s 16 prsti!). Vzemimo za primer ukaz LD BC, (nn). Predstavljamo si lahko, da je sestavljen iz ukazov

LD C, (nn) ter  
LD B, (nn+1)

(v resnici takšna ukaza ne obstajata). Enako velja za somerne ukaze LD (nn), rr. Vsi delujejo na dve celici. Vendar zadostuje, da navedemo samo naslov "nn", ker "nn+1" CP izračuna sam. Paziti pa je treba, da ne bi zamenjali ukazov za delo z 8- in 16-bitnimi števili.

POZOR! Pogosto ti ukazi programerja zapeljejo, da skuša v registrski par naložiti vsebino ene same pomnilniške celice! To je lahko zahrbtn vir napak v programih.

UPORABA: Udomačite si te ukaze z naslednjimi programi. Prvi bo prebiral vrednosti parov pomnilniških zlogov: