

- zastavica polprenosa (H): bo dvignjena, če se bo zaradi operacije spremenil peti bit (tj. bit 4) števila;
  - zastavica odštevanja(N): bo spuščena (= 0) po ukazih INC in dvignjena (= 1) po ukazih DEC;
  - zastavica prenosa (C): nanjo povečevanje in zmanjševanje ne vpliva. Sicer pa se dvigne, kadar pride do prenosa iz najvišjega bita registra.
- Ukazi povečevanja in zmanjševanja 16-bitnih števil ne vplivajo na zastavice.

UPORABA: Poskusite sami napisati program, s katerim boste v par BC naložili število med 0 in 65535 (tj. neko 16-bitno število) in nato uporabite ukaza za povečevanje in zmanjševanje: Primerjajte zdaj vaš program z našim:

```
01 XX XX LD BC, XX XX ;ne pozabite na pravo zaporedje !
03 INC BC
C9 RET
```

Za zmanjševanje nadomestite drugo vrstico z "OB DEC BC".

## RACUNANJE NA ENI ROKI

Vse operacije, o katerih bomo zdajle spregovorili, uporabljajo zgolj 8-bitna števila in jih lahko opravljamo le na CP-jevi desnici, v registru A.

Kaže, da le desnica ve, kako seštevati in odštevati! S tem so prežete celo krajšave, tako da v njih izpuščamo oznako registra A. Če bi hoteli prišteti npr. B k A, bi pričakovali ukaz

```
ADD A, B
```

(ADD = prištej). V resnici pa je krajšava le

```
ADD B.
```

(Povedati moramo, da nekateri zbirniki zahtevajo zapis ukaza v razširjeni obliki, se pravi na primer ADD A, B).

Kljub tej omejitvi je strojni jezik še vedno zelo vsestranski, saj lahko k številu na desnici prištejemo marsikaj:

```
ADD r    prištej k A katerikoli 8-bitni register
ADD n    prištej k A 8-bitno število
ADD (HL) prištej k A vsebino pomnilniške celice na
          naslovu HL
ADD (IX + dis) prištej k A vsebino pomnilniške celice na
          naslovu IX + dis
ADD (IY + dis) prištej k A vsebino pomnilniške celice na
          naslovu IY + dis.
```

To pomeni že pravo bogastvo možnosti. Številu v registru A lahko prištejemo katerikoli 8-bitno število, katerikoli 8-bitni register (razen F) in vsebino katerekoli pomnilniške celice (seveda moramo v zadnjem primeru prej določiti naslov). Vrednost v operandu (tj. v r, v (HL) itn.) se ob operaciji ne spremeni. Najbrž pogrešate ukaz

```
ADD (nn).
```

Takšnega ukaza ni, zato ga moramo sestaviti

```
LD HL, nn
```

```
ADD (HL).
```

Ker lahko na ta način naslov spreminjamo med izvajanjem programa, je to po svoje še ugodnejše. Register HL je tu spet prednosten. S paroma BC in DE naslova ne moremo določiti.