



## NAČINI NASLAVLJANJA IN FORMAT UKAZA

Stran 3-4

Poseben pomen ima register PC ali programski števec. Tega lahko uporabljamo le kot kazalec na naslednji ukaz. Vsak podatek, ki se zapiše v ta register bo procesor tako tolmačil. Pri uporabi programskega števca za naslavljanje je učinek drugačen kot pri ostalih registrih.

## 3.4 NAČINI NASLAVLJANJA

## 3.4.1 Registrski način (5)

MOVW R1, R2

Pri tem načinu uporabljamo register kot akumulator, to je začasna shramba za podatek. Na ta način ne moremo uporabljati programskega števca. Če je podatek prevelik, da bi ga zapisali v en register, naprimer število s plavajočo vejico z dvojno natančnostjo, uporabi procesor še naslednji register. Na ta način ne smemo uporabljati kazalca sklada, ker se s tem implicitno vključi programski števec.

Kot primer zapišimo registrsko naslavljanje z ukazom CLRW, ki zapiše vrednost 0 v eno besedo:

```
CLRW   R3           ; Zapiše 0 v bite 0 do 15 v
MOVW   26, R3      ; registru R3.
```

## 3.4.2 Posredni registrski način (6)

REG. DEFERRED

Register je pri tem načinu kazalec na podatek. Tudi tega načina ne moremo uporabiti s programskim števcem. Primer:

```
CLRW   (R3)        ; Zapiše 0 v dva byta na naslovu,
ADDW2  25, (R8)    ; ki je zapisan v registru R3.
```

AUTOINCR

## 3.4.3 Prištevalni način (8) - +1 (B) +2 (W) +4 (L) +8 (Q)

Prištevalni način se loči od posrednega registrskega le po tem, da se po dostopu do podatka vrednost v registru poveča za število bytov, ki jih zavzema ta podatek. Na ta način delamo s podatki, ki so zapisani v pomnilniku na zaporednih lokacijah. Ko opravimo s prvim podatkom, je v registru že naslov naslednjega.

Za primer naj bo v registru R5 vrednost 100.

```
CLRW   (R5)+       ; Zapiše 0 na naslova 100 in 101
CLRL   (R5)+       ; in poveča vrednost v R5 na 102.
        ; Zapiše 0 na naslove 102, 103
        ; 104 in 105, v R5 pa zapiše 106.
```

```
MOVW   (25) + (27) +
- 16 -
```