

NAČINI NASLAVLJANJA IN FORMAT UKAZA

Stran 3-6

MOVF #2.25,R5 ; Zapiše realno število 2.25 v R5.

DISPLACEMENT

3.4.7 Naslavljanje z odmikom (A)

^B
^W za displacement
^L - default

Podatek lahko naslovimo tudi tako, da uporabimo vsebino registra kot naslov, torej kot pri posrednem registrskem naslavljanju, temu pa prištejemo še neko vrednost - odmik od naslova. Posredni registrski način lahko torej gledamo kot obliko naslavljanja z odmikom 0.

Odmik lahko zapišemo kot število ali kot simbol, ki smo mu na drugem mestu priredili vrednost. Za odmike, ki se večkrat ponavljajo je presledneje na začetku programa definirati simbol z nazornim imenom in kasneje uporabljati za odmik ta simbol namesto ustreznega števila. Tudi to pripomore k čitljivosti programa.

Če imamo v neki podatkovni strukturi od 12. do 15. byte zapisano starost, naslov te podatkovne strukture pa je v R6, lahko dobimo podatek o starosti na naslednji način:

```
STAROST = 12 ; Definiramo simbol za odmik.
MOVL STAROST(R6),R2 ; Zapiše starost v register R2.
MOVL 12(R6),R2 ; Je pri izvajanju programa
; popolnoma enakovredno prvi obliki.
MOVL L^12(R6),R2 ; Za odmik rezervira dolgo besedo,
; program bo za 3 byte daljši.
```

MOV B^5(24), B^3(23) 24+5 23+5

Odmik je lahko različno velik in ga lahko včasih zapišemo v en byte, včasih pa rabimo celo besedo ali celo dolgo besedo. Če sami nič ne povemo o velikosti prostora, kamor bomo zapisali odmik, izbere prevajalnik najkrajši možni prostor.

3.4.8 Posredno naslavljanje z odmikom (B)

DISPL. DEF.

Obstaja tudi posredna oblika naslavljanja z odmikom, pri kateri je to, kar smo pri neposrednem naslavljanju imeli za operand, naslov operanda.

Urnimo se na primer, ko smo v tekstih brisali prvi znak. Zdaj bi radi samo v tretjem tekstu brisali prvi znak. Kazalec na začetek tabele naslovov teh tekstov naj bo v registru R3.

```
CLRB @2*4(R3) ; Prištejemo naslovu v R3 2*4 byte
; in s tem preskočimo prva dva
; podatka (dolga beseda), tretjesa
; pa uporabimo kot naslov.
```

