

## Osoitusmuodot (RAM, SFR)

Mitä kehittyneempi prosessori sitä enemmän varsinkin epäsuoria osoitusmuotoja.

PIC:issä

1. Vakiodata (constant, literal)
2. Vakio-osoite (direct, absolut)
3. Epäsuora (indirect)

### Vakiodata

-data osana käskyä

-ei voi muuttaa suorituksen aikana (vain lähdekoodissa)

**-Microchip kutsuu "literal" komennoiksi**

`movlw 0xff` ; siirtää luvun 0xff w-rekisteriin

`addlw 0x05` ; lisää  $w=w+5$

### Vakio-osoite

-kiinteä osoite käskyn perässä. Voi muuttaa vain lähdekoodissa

-mikrokontrollereissa hyvin yleinen tapa

Käskyn yhteyteen tallentuu pic18 sarjan kontrollereissa vain osoitteen alimmat 8 bittiä. Muistia on kuitenkin enemmän, joten osoitteen ylemmät bitit laitetaan **BSR-rekisteriin (Bank Select Register)**. Ram muisti on jaettu lohkoihin (BANK) ja **BSR rekisteri kertoo mikä lohko on aktiivinen. Käskyjen perään laitetaan sana BANKED kertomaan, että käytetään BSR-rekisteriä.** Tässä mikrokontrollerissa on viimeisessä BANK15:sta erikoisrekisterit **SFR (Special Function Register)**. Tämä pitää sisällään porttien, ajastimien, keskeytysten, AD-muuntimien, tilatietojen jne. kontrolli- ja datarekisterit. Siis näiden kautta ohjataan näitä toimintoja.

Lisäksi on ns. **ACCESS BANK**. Tämä pitää sisällään viittauksen (mapped) rammin BANK0 alkuosan muistipaikkoihin (0 - 0x7f) ja loppuosan osoitteet (0x80 - 0xff) viittaavat osaan BANK15:n SFR rekistereihin.

ACCESS BANK on tavallaan virtuaalinen bank.

**Laittamalla käskyn perään BANKED sanan tilalle ACCESS sana osoite viittaa silloin ACCESS BANK:iin. BSR ei siis silloin käytetä.** ACCESS bankin rammiin kannattaa laittaa muistipaikat, joihin viitataan useasti esimerkiksi silmukkalaskurit. esim.

movwf 0x10,ACCESS	;siirrä rekisterin w sisältö ;osoitteeseen 0x10 rammin ;access lohkoissa
movf 0x22,W,BANKED	;siirretään muistipaikasta ;[BSR]:0x22 tavu w-rekisteriin. ;Rekisterissä BSR on ;aktiivisen lohkon (BANK) ;numero 0 - 5 tai 0x0f.

BSR rekisterin voi ladata movlb 0x01

## Epäsuora osoite

-Ensi osoite laitetaan osoiterekisteriin ja sitten voidaan käskyissä viitata tähän rekisteriin, josta saadaan osoite, joka viittaa rammiin PIC18F452 on kolme epäsuoran osoitteen rekisteriparia

**FSR0H:FSR0L , FSR1H:FSR1L, FSR2H:FSR2L**

Osoitteeseen tarvitaan tässä mallissa kaksi rekisteriä H(igh) ja L(ow) osa.

Vastaavasti käskyssä viitataan osoiterekisteriparin osoittamaan muistipaikkaan pseudorekisterin **INDF0, INDF1, INDF2** kautta. Epäsuorassa osoitustavassa ei tarvita BSR rekisteriä, koska FSRnH:t (n=0,1,2) vastaavat sitä.

**esim. nollataan osoitteet 0x0120 - 0x0130**

```
movlw    D'17'                ; lukumäärä
movwf    laskuri,ACCESS        ; siirretään laskuri
                                ; muistipaikkaan

movlw    0x01
movwf    FSR0H, ACCESS        ; osoitteen ylempi tavu 1
movlw    0x20                ; osoitteen alempi tavu
                                ; w:hen
movwf    FSR0L ,ACCESS        ; w->FSR0L
silmukka: clrf    INDF0, ACCESS ; nollataan
                                ; muistipaikka.
                                ; Osoite saadaan
                                ; FSR0H:FSR0L :stä
incf     FSR0L, ACCESS        ; kasvatetaan osoite
                                ; laskuria
decfsz   laskuri,f, ACCESS    ; vähennetään laskuri
                                ; muistipaikkaa yhdellä ja
                                ; hypätään seuraavan
                                ; käskyn yli, kun laskuri on
                                ; nolla
goto     silmukka            ; hyppy silmukkaan
```

**toisin:**

```
movlw    D'17'                ; lukumäärä
movwf    laskuri,ACCESS        ; siirretään laskuri
                                ; muistipaikkaan
lfsr      0,0x0120             ; FSR:t voi ladata myös
                                ; LFSR komennolla
```

```
silmukka: clrf    POSTINC0,ACCESS ; nollataan
                                ; muistipaikka.
                                ; Osoite saadaan
                                ; FSR0H:FSR0L :
                                ; sitä ja kasvatetaan osoite-
                                ; laskuria.
                                ; FSR0H kasvaa myös, jos
                                ; tarpeen.
decfsz    laskuri,f, ACCESS    ; vähennetään laskuri
                                ; muistipaikkaa yhdellä ja
                                ; hypätään seuraavan
                                ; käskyn yli, kun laskuri on
                                ; nolla
bra       silmukka            ; hyppy silmukkaan
```

Käskyissä vakio-osoitteen paikalle voi laittaa **INDF<sub>n</sub>** (n=0,1,2). **INDF<sub>n</sub>** on eräänlainen pseudorekisteri, joka toimii niin että ram osoite haetaan vastaavasta FSR<sub>n</sub>H:FSR<sub>n</sub>L rekisteriparista ja operoidaan tähän.

**INDF<sub>n</sub>** tilalla voi käyttää **POSTINC<sub>n</sub>**, **PREINC<sub>n</sub>**, **POSTDEC<sub>n</sub>**, **PREDEC<sub>n</sub>** ja **PLUSW<sub>n</sub>** jotka toimivat kuten **INDF<sub>n</sub>** ja sen lisäksi kasvattavat tai vähentävät FSR<sub>n</sub> rekisteriparia nimensä mukaisesti.

## Bittiosoite

-viitataan muistitavun yksittäiseen bittiin

```
bcf STATUS,C, ACCESS ; asetetaan STATUS-rekisteri bitti  
                        ; nollaksi toisin bcf H'0FD8',0  
bsf STATUS,C, ACCESS ; asetetaan STATUS-rekisteri bitti  
                        ; ykköseksi  
btfsc tavux,6, ACCESS ; Jos tavux muistipaikana bitti  
                        ; numero 6 on nolla hypätään  
                        ; seuraavan käskyn yli muutoin  
                        ; suoritetaan se
```

Muoto: tavun osoite, bittinumero

Bitit numeroitu 0 (lsb) – 7 (msb).

SFR-rekisterien osoitteille on annettu nimet tiedostossa p18f452.inc. On siis syytä laittaa ohjelman alkuun

**#include p18f452.inc**

Mikrokontrollerin nimi pitää olla tietysti saman kuin mille ohjelma on tarkoitettu.

Nyt voi käyttää SFR-rekistereille niitä nimiä, joita on datalehdessä käytetty.

Reunahuomautuksena, että muissa PIC malleissa kuin 18 sarja voi muistiin viittaaminen olla erilaista.

Esimerkiksi 16 sarjan pic:ssä käskyn yhteydessä on vain 7 alinta bittiä osoitteesta ja BANK:in valinta tehdään STATUS-rekisterin RP0, RP1 ja IRP bittien avulla. ACCESS BANK toimintoa ei myöskään ole. Joissain malleissa SFR-rekisterit ovat muistialueen alkupäässä ja joissain on ne hajasijoitettu eri BANK:hin. Myös ohjelmamuisti voi olla lohkottu (BANK:ä vastaa PAGE-termi).

## **EEPROM luku ja kirjoitus**

- kts. esimerkki datalehdestä

## **Datan luku ohjelmamuistista eli TBLRD-käskyt**

- Data laitetaan ohjelmakoodiin
- kts. esimerkit datalehdestä