



1918

**TALLINNA TEHNIKAÜLIKOOL**  
TALLINN UNIVERSITY OF TECHNOLOGY

Elektriamite ja jõuelektroonika instituut

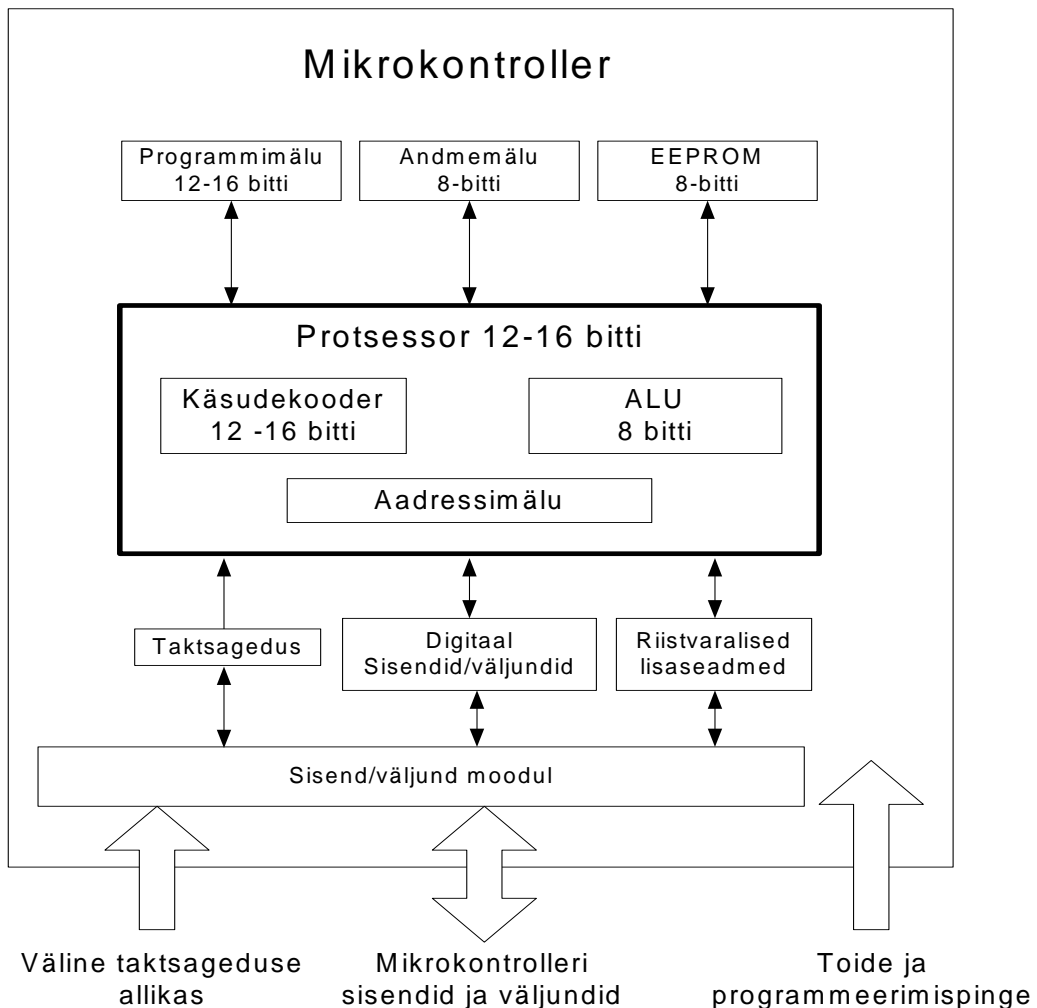
## **PIC mikrokontrolleri ehitus**

Tallinn 2005

## 1 Üldine tutvustus

Mikrokontroller on ühte mikrokiipi mahutatud miniarvuti, mille abil saab sooritada loogikatehteid, arvutusi, juhtida seadmeid, töödelda andmeid. Mikrokontroller koosneb järgnevatest osadest:

- Protsessor
  - ALU (Aritmetic Logic Unit) - aritmeetika loogikaplokk
  - Käsudekooder
  - Aadressimälu
- Mälu
  - Programmimälu
  - Andmemälu (RAM)
  - EEPROM mälu
- Sisend/väljundplokk
- Lisaseadmed



Joonis: 1.1 Mikrokontrolleri plokskeem

Mikrokontrolleri tööle rakendudes hakkab käsudekooder lugema järjest programmimälu sisu, millesse salvestatud käskudele vastavalt kirjutatakse ja loetakse teisi mälusid, kontrollitakse sisendeid ja väljundeid, edastatakse infot

töötlemiseks aritmeetika/loogikaplokkide ning rakendatakse tööle lisaseadmeid. Kuigi 16F877 PIC protsessori programmimälu ja siin on 14 bitine, tuleb ALU toime vaid 8 bitiste arvude liitmise ja lahtutamise (18F seeria mikrokontrolleritel ka korrutamise) ning lihtsamate loogikatehetega. Ka üldiseks kasutamiseks mõeldud mälu iga mälupesa on 8 bitine. See tähendab, et PIC mikrokontrollerid tunnevad vaid arvusid 0 kuni 255. See ei takista siiski PIC mikrokontrollerite programmides suurte arvude kasutamist.

## 2 Mälude struktuur

PIC tüüpi mikrokontrolleritel on programmi, andmete ja vahetulemuste salvestamiseks kolme liiki mälu.

- Flash – tüüpi korduvalt ülekirjutatav programmimälu, kuhu salvestatakse programm. Mällu salvestatud info säilib ka toitepinge kadumisel.
- Muutmälu (RAM), mida kasutatakse tihti muutuvate ja ajutiselt vajalike andmete hoidmiseks (muutujate väärtused, seadistusregistrid, vahetulemused, jne). RAM – tüüpi mälus hoitav info kustub pärast mikrokontrolleri toitepinge kadumist.
- EEPROM – tüüpi mälu kasutaja poolt defineeritud andmete programmiliseks salvestamiseks. Kasutatakse enamasti harva muudetavate seadeväärtuste salvestamiseks, mis peavad olema loetavad ka pärast mikrokontrolleri taaskäivitamist.

PIC 16F877a mikrokontrolleri muutmälu on jaotatud neljaks mälupangaks. Igas mälupangas on 128 8-bitist mälupesa.

PIC mikrokontrolleritel on aadressi pikkus piiratud 7 bitiste arvudega, mis tähendab 128 registrit, ent mälukohtade tegelik aadressivahemik on 00h – 1FFh (kümneksüsteemis 0 – 511). Nii suure arvu kirjeldamiseks oleks vaja juba 9 bitist arvu, ent PIC võimaldab tehteid üleüldse ainult 8 bitiga. Lahendamaks probleemi ning tegemaks 7 bitisest aadressist 9 bitist ongi need 128 registrit jagatud nelja mälupanka ning mälupankade valiku bitid lisatakse üldisele aadressile. Mälupankasid saab valida, omistades väärtuseid STATUS registri bittidele RP0 ja RP1.

- RP0=0, RP1=0 ;Valitakse mälupank nr. 0
- RP0=1, RP1=0 ;Valitakse mälupank nr. 1
- RP0=0, RP1=1 ;Valitakse mälupank nr. 2
- RP0=1, RP1=1 ;Valitakse mälupank nr. 3

Näiteks: me soovime anda registrile aadressiga 20 (kümneksüsteemis 32) väärtust 10 kümneksüsteemis:

```
movlw d '10' ;Omistame tööregistrile väärtuse 10
movwf 0x20 ;Omistame tööregistri väärtuse mälupesale aadressil 20
```

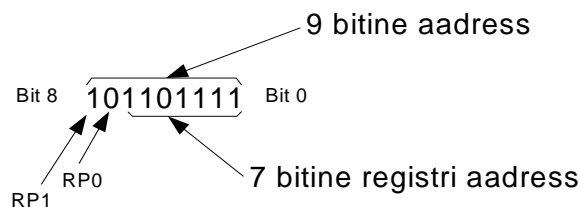
Sellist adresseerimist kutsutakse otseseks adresseerimiseks. Samamoodi saab aadresse kasutada ka kõigi teiste käskudega ning omistada muutujatele mäluaadresse. Samas mäluruumis on PIC mikrokontrolleritel ka mitmesugused seaderegistrid (näiteks STATUS, PORTA jne), millele erinevaid väärtuseid andes on võimalik

mikrokontrollerit ümber seadistada või saada infot kontrolleris toimuvate protsesside kohta. Kasutaja võib oma muutujate väärtuste salvestamiseks kasutada GPR registreid (General Purpose Register). Kõik seaderegistrid ja tähtsamad seadebitid on kirjeldatud vastavale mikrokontrollerile mõeldud include failis, kus aadressidele on omistatud nimed, nii võib numbriliste väärtuste asemel kasutada MPLAB programmeerimistarkvaras ka seaderegistrite nimesid.

File Address	File Address	File Address	File Address
Indirect addr. <sup>(1)</sup> 00h	Indirect addr. <sup>(1)</sup> 80h	Indirect addr. <sup>(1)</sup> 100h	Indirect addr. <sup>(1)</sup> 180h
TMR0 01h	OPTION_REG 81h	TMR0 101h	OPTION_REG 181h
PCL 02h	PCL 82h	PCL 102h	PCL 182h
STATUS 03h	STATUS 83h	STATUS 103h	STATUS 183h
FSR 04h	FSR 84h	FSR 104h	FSR 184h
PORTA 05h	TRISA 85h		
PORTB 06h	TRISB 86h	PORTB 106h	TRISB 186h
PORTC 07h	TRISC 87h		
PORTD <sup>(1)</sup> 08h	TRISD <sup>(1)</sup> 88h		
PORTE <sup>(1)</sup> 09h	TRISE <sup>(1)</sup> 89h		
PCLATH 0Ah	PCLATH 8Ah	PCLATH 10Ah	PCLATH 18Ah
INTCON 0Bh	INTCON 8Bh	INTCON 10Bh	INTCON 18Bh
PIR1 0Ch	PIE1 8Ch	EEDATA 10Ch	EECON1 18Ch
PIR2 0Dh	PIE2 8Dh	EEADR 10Dh	EECON2 18Dh
TMR1L 0Eh	PCON 8Eh	EEDATH 10Eh	Reserved <sup>(2)</sup> 18Eh
TMR1H 0Fh		EEADRH 10Fh	Reserved <sup>(2)</sup> 18Fh
T1CON 10h			
TMR2 11h	SSPCON2 91h		
T2CON 12h	PR2 92h		
SSPBUF 13h	SSPADD 93h		
SSPCON 14h	SSPSTAT 94h		
CCPR1L 15h			
CCPR1H 16h			
CCP1CON 17h			
RCSTA 18h	TXSTA 98h	General Purpose Register 16 Bytes	General Purpose Register 16 Bytes
TXREG 19h	SPBRG 99h		
RCREG 1Ah			
CCPR2L 1Bh			
CCPR2H 1Ch	CMCON 9Ch		
CCP2CON 1Dh	CVRCON 9Dh		
ADRESH 1Eh	ADRESL 9Eh		
ADCON0 1Fh	ADCON1 9Fh		
General Purpose Register 96 Bytes	General Purpose Register 80 Bytes	General Purpose Register 80 Bytes	General Purpose Register 80 Bytes
	accesses 70h-7Fh	accesses 70h-7Fh	accesses 70h-7Fh
Bank 0 7Fh	Bank 1 FFh	Bank 2 17Fh	Bank 3 1FFh

Joonis 2.1 PIC 16F877a mälukaart.

Näide: soovime anda registrile aadressiga 16Fh (kümnendsüsteemis 367) väärtuse 10. 16Fh 9-bitise arvuna binaaris on 101101111 bit 0 on kõige parempoolsem arv ja bit 8 on kõige vasakpoolsem.



Joonis 2.2 9-bitine aadress otsese adresseerimisega.

RP0=0 ja RP1=1 tähendab, et register asub mäluosas number 3. 7 bitti aadressi algusest 1101111 annavad 6Fh (kümnendsüsteemis 111). Järelikult register 16Fh asub mäluosas nr.3 aadressil 6Fh.

```

;Valime mälupanga 3 RP1=1 ja RP0=0
bsf STATUS,RP1 ;RP1=1
bcf STATUS,RP0 ;RP0=0
movlw d '10' ;Omistame tööregistrile väärtuse 10
movwf 0x6F ;Omistame tööregistri väärtuse mälupesale
;adressil 6F

```

### 3 Sisendid ja väljundid seadistamine ning kasutamine

PIC mikrokontrollerite sisendid ja väljundid on jagatud 8-bitistesse portidesse. See tähendab: igas pordis on 8 viiku, mida saab vastavalt vajadusele konfigureerida töötama kas sisendi või väljundina. Pordid on tähistatud tähtede ja üksikud viigud numbritega (porta.4 või lühendatult RA.4 tähendab porti A kuuluvat viiku number 4). Mikrokontrolleri sisendeid ja väljundeid saab seadistada TRIS ja PORT registritega. TRIS registri iga bitt on seotud vastava järjekorranumbriga viiguga.

Näide: seadistame portc kõik viigud väljunditeks ning lülitame väljundid sisse.

```

bsf STATUS, RP0 ;Valime mälupanga 1 RP0=1
movlw b '00000000' ;Tööregistrisse väärtus 0
movwf TRISC ;Tööregistri sisu TRISC registrisse,
;millega muudetakse kõikPortC kõik
;viigud väljunditeks

movlw b '11111111' ;Tööregistrisse väärtus 1
movwf PORTC ;RC0 kuni RC7 väljundisse signaal

```

Samamoodi on võimalik viikusid ka sisenditeks defineerida, kirjutades TRIS registrisse vastavate bittide väärtuseks 1. Sisendi olekut näitavad vastava PORT registri bitid.

Näiteks: andes 5V pinge ainult PORTC viigule 0, mis on defineeritud sisendiks, on registris PORTC väärtus 00000001.

Enamusel mikrokontrolleri viikudest on mitmeid lisafunktsioone ning nendega on seotud riistvaralised integreeritud lisaseadmed, millest on täpsemalt kirjutatud peatükis 7. Enamasti vajavad lisafunktsioonid aktiveerimist välja arvatud analoog/digitaal muundur. Kõigil PIC mikrokontrolleritel on A/D muunduriga seotud PORTA, mis on vaikimisi sisse lülitatud ja võib põhjustada probleeme kui PORTA viikusid üritatakse kasutada digitaalsisendite või –väljunditena. A/D muundurite välja lülitamiseks tuleb kirjutada ADCON1 registrisse väärtus 7.