

MİKRODENETLEYİCİLER

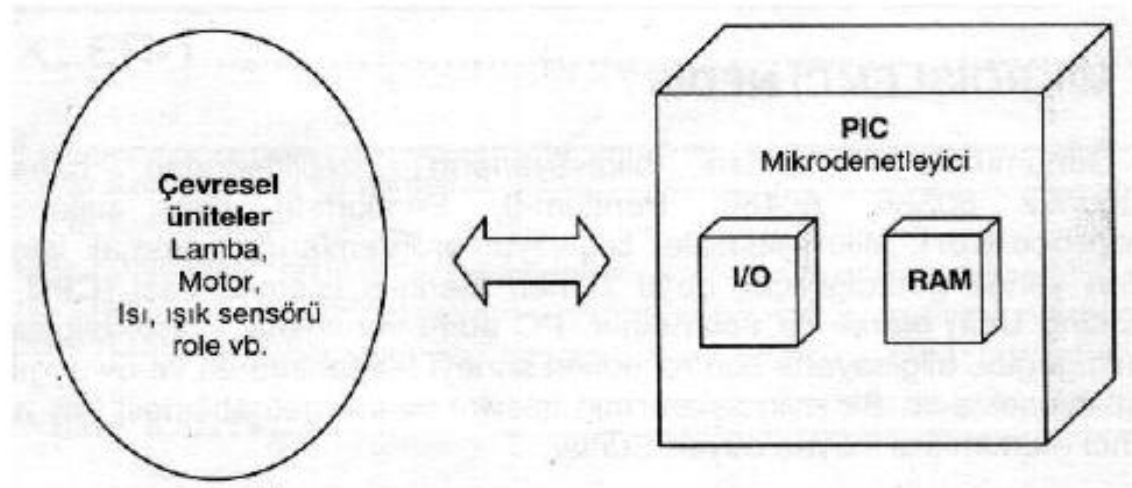
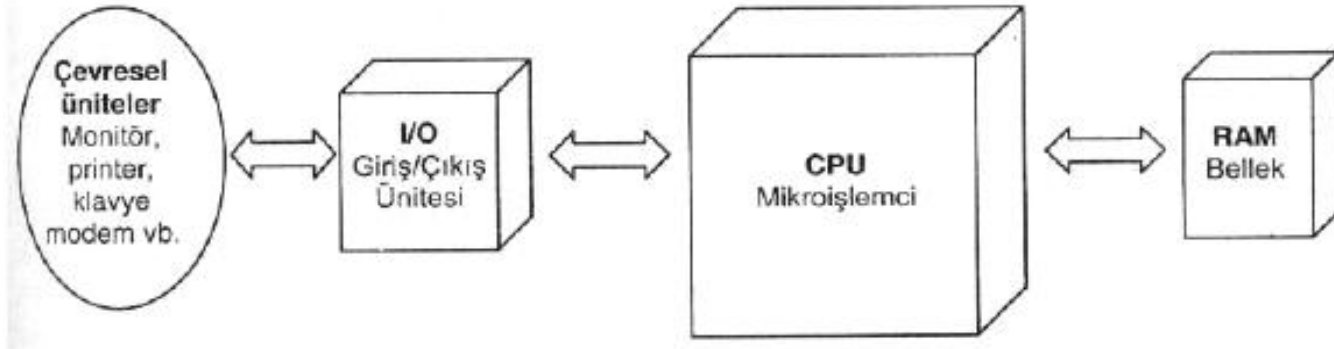
Mikroişlemci: Merkezi işlem biriminin fonksiyonlarını tek bir yarı iletken tümleşik devrede birleştiren programlanabilir sayısal elektronik devre

Mikrodenetleyici: Bir mikroişlemcinin merkezi işlem birimi, bellek ve giriş/çıkış birimlerinin bazı özelliklerinin azaltılarak tek bir entegre içerisinde üretilmiş biçimi

MİKRODENETLEYİCİ KULLANIMININ TERCİH EDİLME SEBEPLERİ

- Küçük boyutlu
- Güç tüketimi düşük
- Düşük maliyetli
- Yüksek performanslı

MİKRODENETLEYİCİ KULLANIM ALANLARI



MİKRODENETLEYİCİ SEÇİMİNDE DİKKAT EDİLMESİ GEREKENLER

- Programlanabilir sayısal paralel giriş/çıkış
- Programlanabilir analog giriş/çıkış
- Seri giriş/çıkış
- Motor veya servo kontrol için tetikleme sinyali çıkışı
- Harici giriş vasıtasıyla kesme

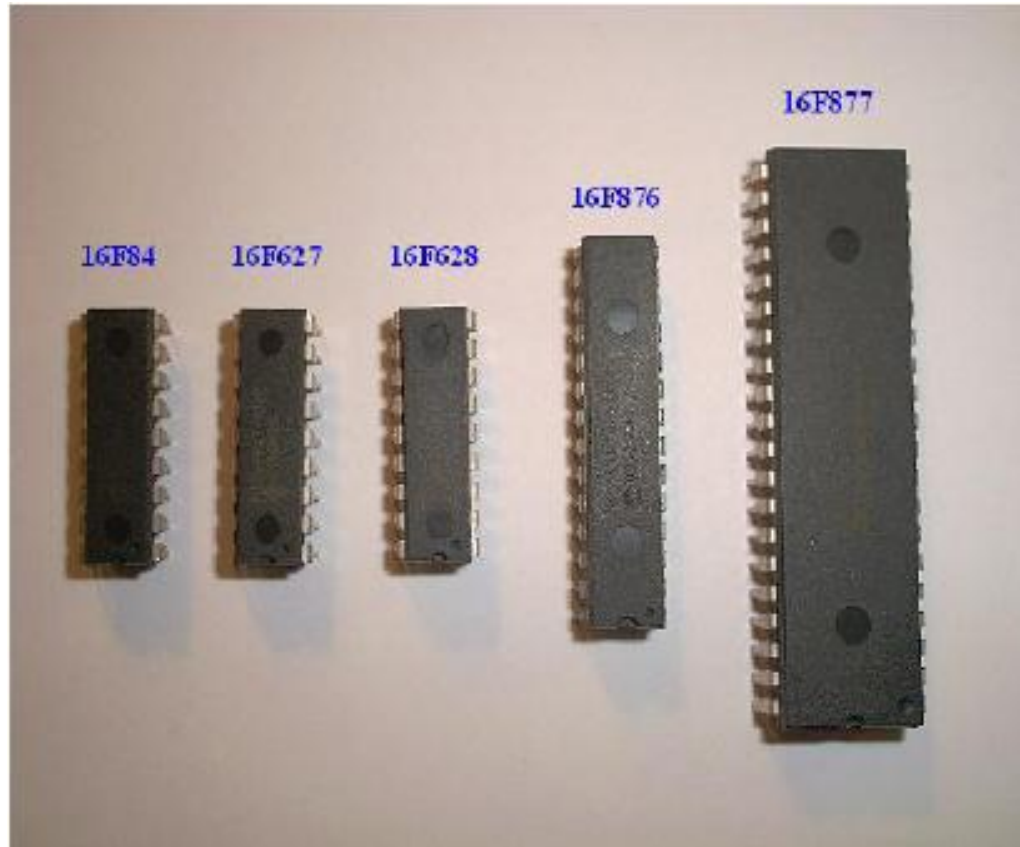
MİKRODENETLEYİCİ SEÇİMİNDE DİKKAT EDİLMESİ GEREKENLER

- Zamanlayıcı vasıtasıyla kesme
- Harici bellek arabirimi
- Harici veri yolu arabirimi
- Dahili bellek tipi seçenekleri (ROM, EEPROM...)
- Dahili RAM seçeneği
- Kayan nokta hesaplaması

PIC NEDİR?

- PIC: Peripheral Interface Controller
= Çevresel Arayüz Birimi Denetleyicisi
- Microchip firmasının ürettiği mikrodeneleyicilerin genel adı
- Amacı: Çok fonksiyonlu mantıksal uygulamaların hızlı ve ucuz bir mikroişlemci ile yazılım yoluyla karşılanması
- İlk PIC: PIC16C54 (1994)
- En yaygın kullanılan PIC: PIC16F(C)84

BAZI 16F SERİSİ MİKRODENETLEYİCİLER



MİKRODENETLEYİCİ MİMARİSİ

İki çeşit mikroişlemci mimarisi vardır.

1. Von Neuman Mimarisi

- Program kodları (komutlar) ve veriler aynı bellek biriminden tek bir yoldan alınıp işlemciye getirilir.
- Önce komut getirilir, daha sonra veri alınıp işlenir.
- Bu mimaride gecikmeler mevcuttur.

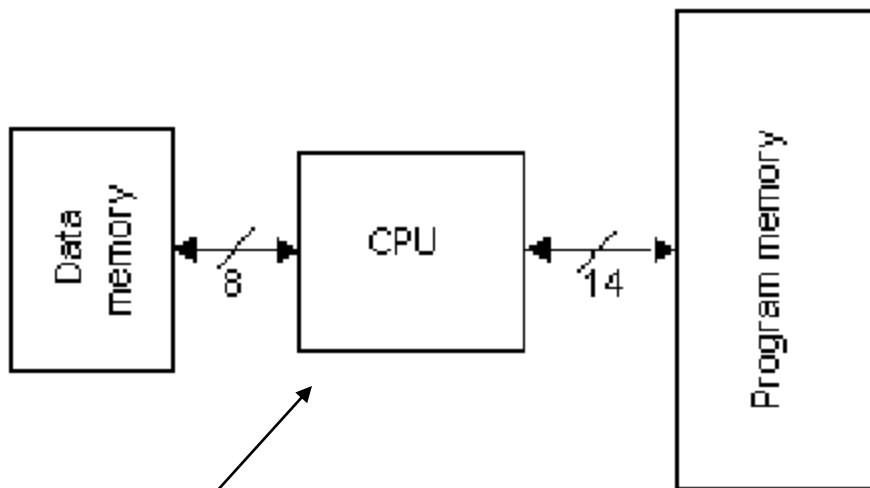
MİKRODENETLEYİCİ MİMARİSİ

2. Harvard Mimarisi

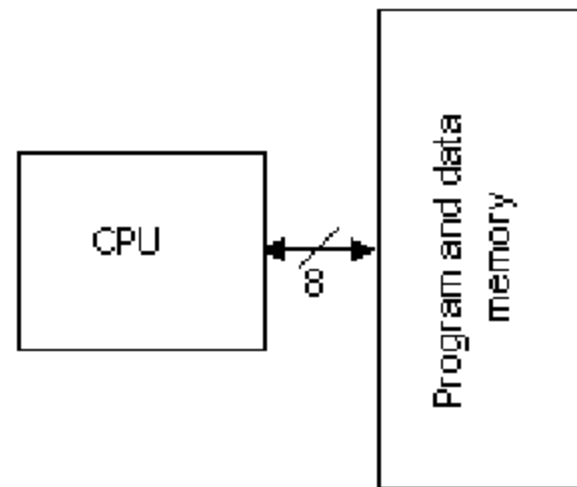
- Program kodları ve verilere ayrı yollardan ulaşılır.
- Çalışması daha hızlıdır.
- Mikrodenetleyicilerde genelde bu mimari tercih edilir.

MİKRODENETLEYİCİ MİMARİSİ

Harvard



von-Neumann



Harvard vs. von Neuman Block Architectures

PIC 16f84

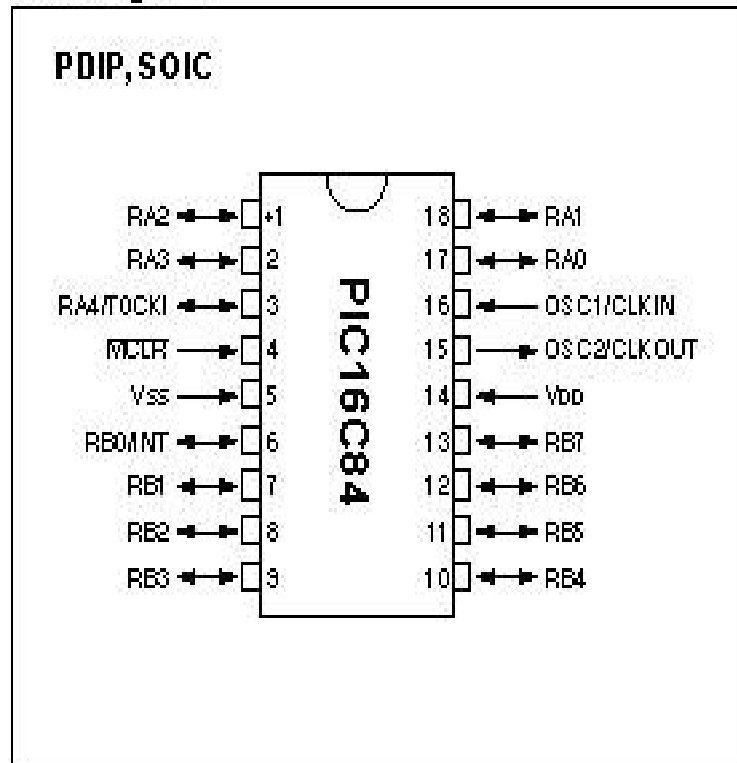
8086 microprocessor

PIC PROGRAMLAMAK İÇİN GEREKENLER

- IBM uyumlu bilgisayar
- Metin editörü
- Derleyici programı (MPLAB, Micropro, PicEQ, Propic, ICProg...)
- PIC programlayıcı donanım ve yazılım
- PIC

PIC16F84 PİN GÖRÜNÜŞÜ

Pin Diagram



5 pin: A portu

8 pin: B portu

1 pin: Reset

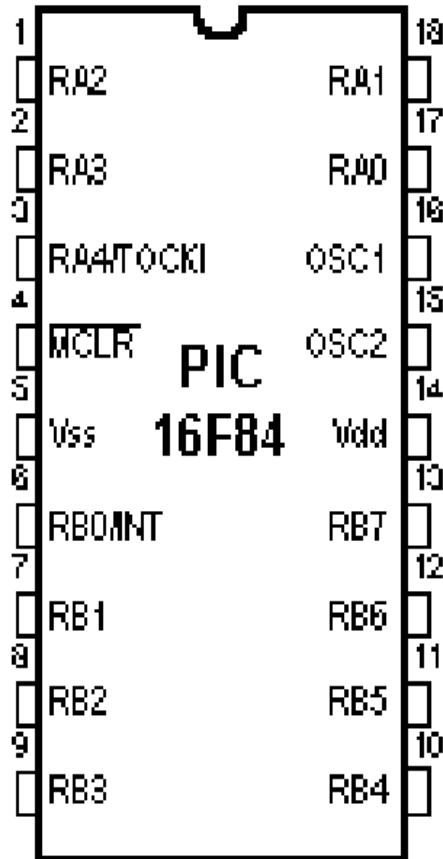
1 pin: Pozitif besleme

1 pin: Toprak

2 pin: Osilatör uçları

+
18 pin

PIC16F84 PIN TANIMLAMALARI



Pin no.1 **RA2** Second pin on port A. Has no additional function

Pin no.2 **RA3** Third pin on port A. Has no additional function.

Pin no.3 **RA4** Fourth pin on port A. TOCK1 which functions as a timer is also found on this pin

Pin no.4 **MCLR** Reset input and Vpp programming voltage of a microcontroller

Pin no.5 **Vss** Ground of power supply.

Pin no.6 **RB0** Zero pin on port B. Interrupt input is an additional function.

Pin no.7 **RB1** First pin on port B. No additional function.

Pin no.8 **RB2** Second pin on port B. No additional function. Pin no.9 **RB3** Third pin on port B. No additional function.

Pin no.10 **RB4** Fourth pin on port B. No additional function.

Pin no.11 **RB5** Fifth pin on port B. No additional function.

Pin no.12 **RB6** Sixth pin on port B. 'Clock' line in program mode.

Pin no.13 **RB7** Seventh pin on port B. 'Data' line in program mode.

Pin no.14 **Vdd** Positive power supply pole.

Pin no.15 **OSC2** Pin assigned for connecting with an oscillator

Pin no.16 **OSC1** Pin assigned for connecting with an oscillator

Pin no.17 **RA2** Second pin on port A. No additional function

Pin no.18 **RA1** First pin on port A. No additional function.

PIC16F84 MİKRODENETLEYİCİ İÇ YAPISI

Memory organization

PIC16F84 has two separate memory blocks, one for data and the other for program. EEPROM memory and GPR registers in RAM memory make up a data block, and FLASH memory makes up a program block.

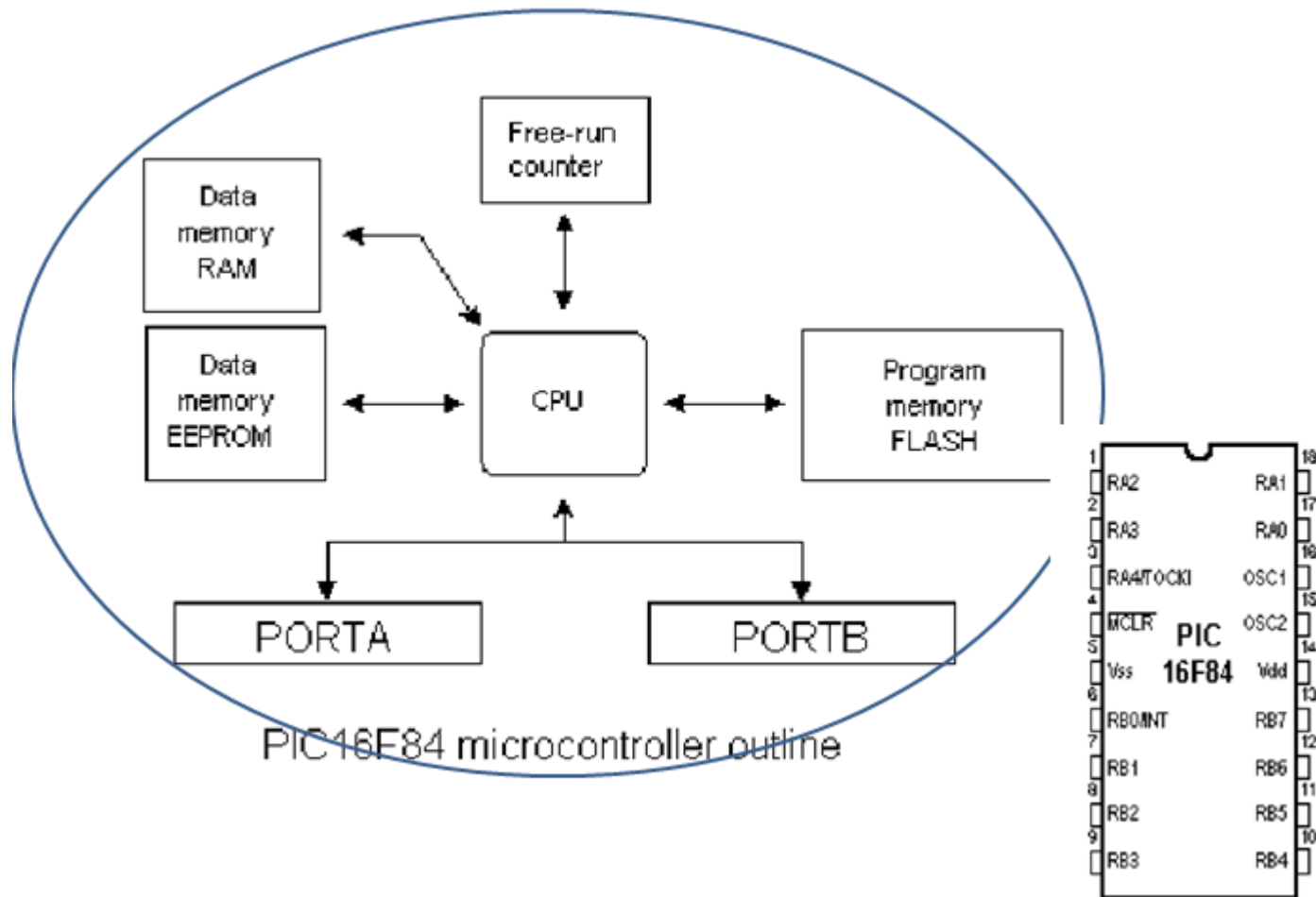
Program memory:

Program memory has been realized in FLASH technology which makes it possible to program a microcontroller many times before it's installed into a device, and even after its installment if eventual changes in program or process parameters should occur. The size of program memory is 1024 locations with 14 bits width where locations zero and four are reserved for reset and interrupt vector.

Data memory

Data memory consists of EEPROM and RAM memories. EEPROM memory consists of 64 eight bit locations whose contents is not lost during loosing of power supply. EEPROM is not directly addressible, but is accessed indirectly through EEADR and EEDATA registers. As EEPROM memory usually serves for storing important parameters (for example, of a given temperature in temperature regulators) , there is a strict procedure for writing in EEPROM which must be followed in order to avoid accidental writing. RAM memory for data occupies space on a memory map from location 0x0C to 0x4F which comes to 68 locations. Locations of RAM memory are also called GPR registers which is an abbreviation for General Purpose Registers. GPR registers can be accessed regardless of which bank is selected at the moment.

PIC16F84 MIKRODENETLEYİCİ İÇ YAPISI



OSİLATÖR UÇLARI VE BAĞLANTILARI

- PIC belleğinde bulunan program komutlarının çalıştırılması için kare dalga sinyale ihtiyaç vardır. Bu sinyale **saat sinyali**, bu sinyali sağlayan devreye de **osilatör** adı verilir.
- PIC16F84'ün osilatör girişi olarak kullanılan iki pini vardır.

OSİLATÖR UÇLARI VE BAĞLANTILARI

- Bu pinlere farklı tipte osilatörlerden elde edilen sinyaller uygulanabilir.

RC-Direnç/Kapasitör (0-4MHz)

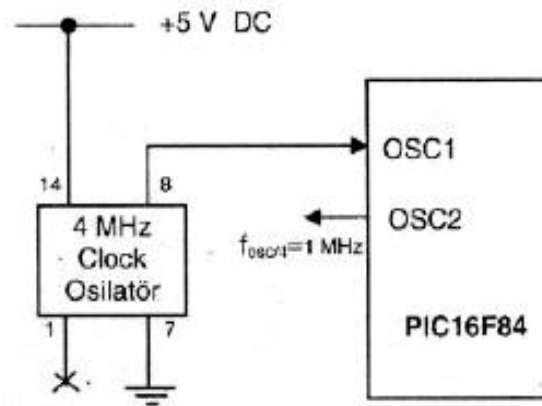
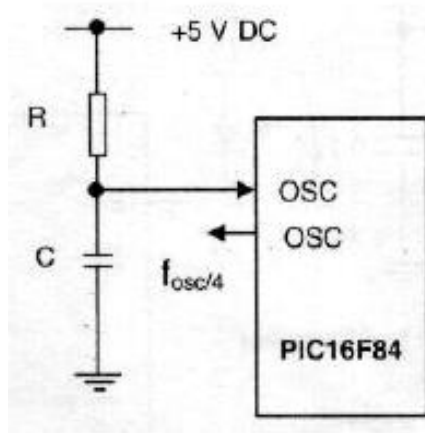
XT-Kristal veya seramik resonatör (XTAL) (0.1-4MHz)

HS-Yüksek hızlı kristal veya resonatör (High Speed) (4-20 MHz)

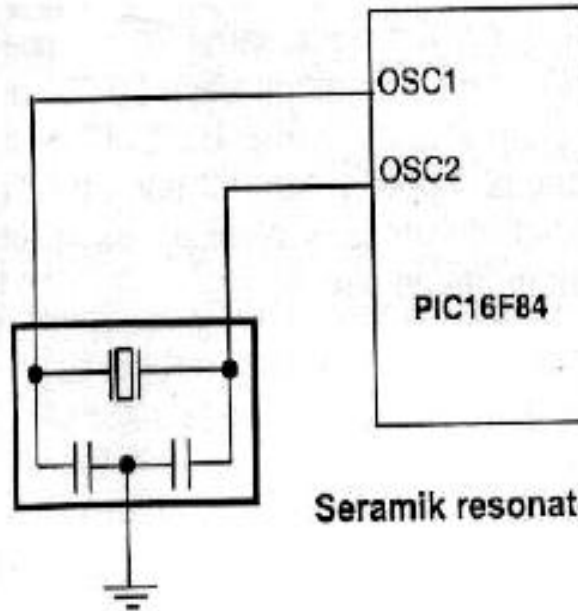
LP-Düşük frekanslı kristal (Low Power) (5-200KHz)

- PIC'e bağlanan osilatörün tipi programlama sırasında konfigürasyon bitlerine yazılır.

OSİLATÖR UÇLARI VE BAĞLANTILARI

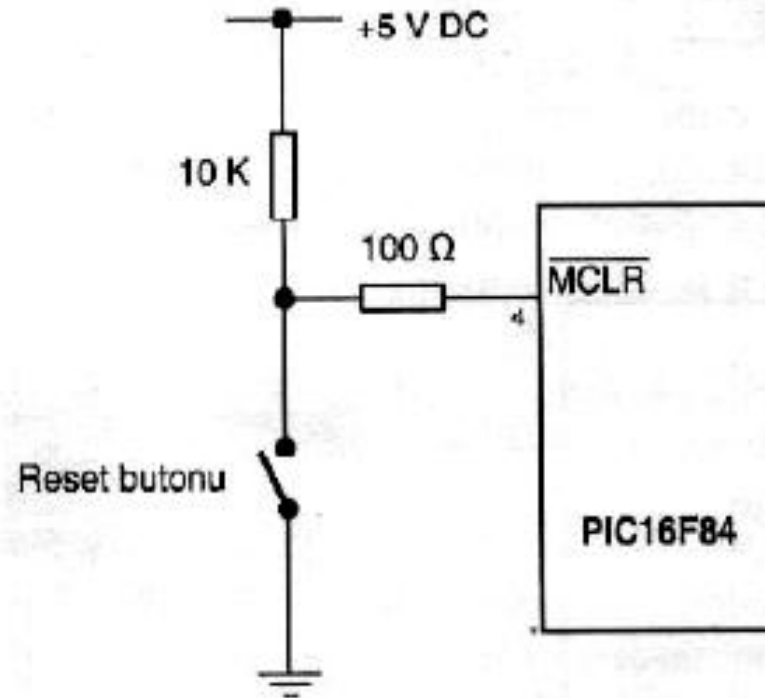


OSİLATÖR UÇLARI VE BAĞLANTILARI



Seramik resonatör osilatörünün PIC'e bağlantısı

RESET DEVRESİ



BANK (16F84)

- 8086'daki segmentler yerine PIC'te Bank'lar ve Bank'ların içinde özel ve genel amaçlı yazmaçlar vardır.
- Bir yazmacı kullanabilmek için, yazmacın ait olduğu Bank'a geçiş yapılmalıdır.

BANK (16F84)

File Address		File Address
00h	Indirect addr. ⁽¹⁾	80h
01h	TMR0	OPTION
02h	PCL	PCL
03h	STATUS	STATUS
04h	FSR	FSR
05h	PORTA	TRISA
06h	PORTB	TRISB
07h		
08h	EEDATA	EECON1
09h	EEADR	EECON2 ⁽¹⁾
0Ah	PCLATH	PCLATH
0Bh	INTCON	INTCON
0Ch		
	68 General Purpose registers (SRAM)	Mapped (accesses) in Bank 0
4Fh		
50h		
7Fh		
	Bank 0	Bank 1
		FFh

■ Unimplemented data memory location; read as '0'.

Note 1: Not a physical register.

BANK (16F877)

File Address	File Address	File Address	File Address
Indirect addr. ⁽¹⁾ 00h	Indirect addr. ⁽¹⁾ 80h	Indirect addr. ⁽¹⁾ 100h	Indirect addr. ⁽¹⁾ 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 ⁽¹⁾ 08h	TRISD ⁽¹⁾ 88h		
PORTE ⁽¹⁾ 09h	TRISE ⁽¹⁾ 89h		
PCLATH 0Ah	PCLATH 8Ah	PCLATH 10Ah	PCLATH 18Ah
INTCON 0Bh	INTCON 8Bh	INTCON 10Bh	INTCON 18Bh
PIR1 0Ch	PIE1 8Ch	EEDATA 10Ch	EEDCON1 18Ch
PIR2 0Dh	PIE2 8Dh	EEADR 10Dh	EECON2 18Dh
TMR1L 0Eh	PCON 8Eh	EEDATH 10Eh	Reserved ⁽²⁾ 18Eh
TMR1H 0Fh		EEADRH 10Fh	Reserved ⁽²⁾ 18Fh
T1CON 10h			
TMR2 11h	SSPCON2 91h		
T2CON 12h	PR2 92h		
SSPBUF 13h	SSPAD0 93h		
SSPCON 14h	SSPSTAT 94h		
CCPR1L 15h			
CCPR1H 16h			
CCP1CON 17h		General Purpose Register 10 Bytes	General Purpose Register 10 Bytes
RCSTA 18h	TXSTA 98h		
TXREG 19h	SPBRG 99h		
RCREG 1Ah			
CCP2CON 1Bh			
CCPR2H 1Ch			
CCP2CON 1Dh			
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
7Fh	accesses 10h-7Fh	accesses 70h-7Fh	accesses 70h-7Fh
Bank 0	Bank 1	Bank 2	Bank 3

Unimplemented data memory locations, read as '0'.
⁽¹⁾ Not a physical register.

Note 1: These registers are not implemented on the PIC18F876.
Note 2: These registers are reserved, maintain these registers clear.

PIC ASSEMBLY KOMUTLARI

- 35 komut mevcuttur.
- Her komut 14 bit uzunluğundadır.
- İstisnai birkaç komut haricinde (CALL, GOTO, BTFSS...) her komut tek bir saat çevrim süresinde çalışır.

PIC ASSEMBLY KOMUTLARI

- 1. Byte Yönlendirmeli Komutlar:** ADDWF, ANDWF, CLRF, CLRW, COMF, DECF, DECFSZ, INCF, INCFSZ, IORWF, MOVF, MOVWF, NOP, RLF, RRF, SUBWF, SWAPF, XORWF)
- 2. Bit Yönlendirmeli Komutlar:** BCF, BSF, BTSFC, BTFSS

PIC ASSEMBLY KOMUTLARI

- 3. Sabit İşleyen Komutlar:** ANDLW, ADDLW, IORLW, MOVLW, RETLW, SUBLW, XORLW

- 4. Kontrol Komutları:** CALL, CLRWDT, GOTO, RETFIE, RETURN, SLEEP

PIC ASSEMBLY KOMUTLARI

Mnemonic, Operands		Description	Cycles	14-Bit Opcode				Status Affected	Notes
				MSb		LSb			
BYTE-ORIENTED FILE REGISTER OPERATIONS									
ADDWF	f, d	Add W and f	1	00	0111	dfff	ffff	C,DC,Z	1,2
ANDWF	f, d	AND W with f	1	00	0101	dfff	ffff	Z	1,2
CLRF	f	Clear f	1	00	0001	1fff	ffff	Z	2
CLRW	-	Clear W	1	00	0001	0xxx	xxxx	Z	
COMF	f, d	Complement f	1	00	1001	dfff	ffff	Z	1,2
DECF	f, d	Decrement f	1	00	0011	dfff	ffff	Z	1,2
DECFSZ	f, d	Decrement f, Skip if 0	1(2)	00	1011	dfff	ffff		1,2,3
INCF	f, d	Increment f	1	00	1010	dfff	ffff	Z	1,2
INCFSZ	f, d	Increment f, Skip if 0	1(2)	00	1111	dfff	ffff		1,2,3
IORWF	f, d	Inclusive OR W with f	1	00	0100	dfff	ffff	Z	1,2
MOVF	f, d	Move f	1	00	1000	dfff	ffff	Z	1,2
MOVWF	f	Move W to f	1	00	0000	1fff	ffff		
NOP	-	No Operation	1	00	0000	0xx0	0000		
RLF	f, d	Rotate Left f through Carry	1	00	1101	dfff	ffff	C	1,2
RRF	f, d	Rotate Right f through Carry	1	00	1100	dfff	ffff	C	1,2
SUBWF	f, d	Subtract W from f	1	00	0010	dfff	ffff	C,DC,Z	1,2
SWAPF	f, d	Swap nibbles in f	1	00	1110	dfff	ffff		1,2
XORWF	f, d	Exclusive OR W with f	1	00	0110	dfff	ffff	Z	1,2

PIC ASSEMBLY KOMUTLARI

BIT-ORIENTED FILE REGISTER OPERATIONS									
BCF	f, b	Bit Clear f	1	01	00bb	bfff	ffff		1,2
BSF	f, b	Bit Set f	1	01	01bb	bfff	ffff		1,2
BTFSC	f, b	Bit Test f, Skip if Clear	1 (2)	01	10bb	bfff	ffff		3
BTFSS	f, b	Bit Test f, Skip if Set	1 (2)	01	11bb	bfff	ffff		3
LITERAL AND CONTROL OPERATIONS									
ADDLW	k	Add literal and W	1	11	111x	kkkk	kkkk	C,DC,Z	
ANDLW	k	AND literal with W	1	11	1001	kkkk	kkkk	Z	
CALL	k	Call subroutine	2	10	0kkk	kkkk	kkkk		
CLRWDT	-	Clear Watchdog Timer	1	00	0000	0110	0100	$\overline{TO}, \overline{PD}$	
GOTO	k	Go to address	2	10	1kkk	kkkk	kkkk		
IORLW	k	Inclusive OR literal with W	1	11	1000	kkkk	kkkk	Z	
MOVLW	k	Move literal to W	1	11	00xx	kkkk	kkkk		
RETFIE	-	Return from interrupt	2	00	0000	0000	1001		
RETLW	k	Return with literal in W	2	11	01xx	kkkk	kkkk		
RETURN	-	Return from Subroutine	2	00	0000	0000	1000		
SLEEP	-	Go into standby mode	1	00	0000	0110	0011	$\overline{TO}, \overline{PD}$	
SUBLW	k	Subtract W from literal	1	11	110x	kkkk	kkkk	C,DC,Z	
XORLW	k	Exclusive OR literal with W	1	11	1010	kkkk	kkkk	Z	

KONFIGÜRASYON BİTLERİNİN AYARLANMASI

Konfigurasyon bit'leri, PIC'e gerilim uygulandığı anda PIC'in uyması gereken koşulları belirler.

Koşul	Yazılacak Tanım Kodu
Kod koruma var	<code>_CP_ON</code>
Kod koruma yok	<code>_CP_OFF</code>
Power-on-reset var	<code>_PWRTE_ON</code>
Power-on-reset yok	<code>_PWRTE_OFF</code>
Watchdog timer devrede	<code>_WDT_ON</code>
Watchdog timer devrede yok	<code>_WDT_OFF</code>
Low Power Osilatör	<code>_LP_OSC</code>
Kristal Osilatör	<code>_XT_OSC</code>
High Speed Osilatör	<code>_HS_OSC</code>
RC Osilatör	<code>_RC_OSC</code>

AKÜMÜLATÖR

- 8086'da kullanılan yamaçlar yerine (AX, BX, CX, DX) PIC yalnızca W yazmacını kullanır.
- Bir değer bir değişkene atanmak istendiğinde W yazmacının kullanılması zorunludur.

AKÜMÜLATÖR

- Bu yazmaç sanal bir saklayıcıdır. İçeriğine doğrudan ulaşmak mümkün değildir. Ancak bütün yükleme işlemleri bu yazmaç aracılığı ile yapılır.
- Bir değişkene, bir yazmaca ya da bir porta bilgi göndermek için önce bu bilgi W yazmacına, daha sonra W yazmacını ilgili porta ya da değişkene yüklemek yolunu izlemek gerekmektedir.

AKÜMÜLATÖR

- Örnek: Bir değişkene bilgi yükleme

MOVLW D'15'; W=15

MOVWF SAYAC; SAYAC=15

STATUS YAZMACI

- Bank'lar arasındaki geçişi sağlar.

R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x	
IRP	RP1	RP0	TQ	PD	Z	DC	C	
bit 7								bit 0

bit 6-5 **RP1:RP0:** Register Bank Select bits (used for direct addressing)

11 = Bank 3 (180h - 1FFh)

10 = Bank 2 (100h - 17Fh)

01 = Bank 1 (80h - FFh)

00 = Bank 0 (00h - 7Fh)

Each bank is 128 bytes

STATUS YAZMACI

- **BSF STATUS, RP0** ;Bit Set f
; (Status'da RP0)
- **BSF STATUS, 5** ;Bit Set f
; (Status'da Bit 5)

R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
IRP	RP1	RP0	\overline{TO}	\overline{PD}	Z	DC	C
bit 7			bit 0				

PROGRAM BÖLÜMLERİ

```
LIST          P=16F877
INCLUDE       P16F877.INC
```

Başlık Bloğu

```
#DEFINE      DEGISKEN_1      d'100'
#DEFINE      DEGISKEN_2      b'10010110'      ;=d'150'=0x96
DEGIS_3      EQU             0x20      ;HAFIZADA SEÇİLEN BİR YERE
DEGIS_4      EQU             0x21      ;ATANAN DEĞİŞKEN
```

Adres ve Değişken
Atama Bloğu

```
ORG          0      ;PROGRAMI 0X00'DAN BAŞLAT
```

Port Durumu Belirleme Bloğu

Açıklama Bloğu

BASLA

```
BSF          STATUS,RP0
MOVLW        b'00000111'
MOVWF        TRISA
CLRf         TRISB
```

```
BCF          STATUS,RP0
```

```
MOVLW        b'00001010'
MOVWF        PORTB
CLRf         PORTB
```

```
;BANK1'E GEÇİŞ
;İSTENEN I/O KONFIGURASYONU SEÇİLİR
;TRISA REGISTER'INA ATANIR
;TRISB ÇIKIŞ MODUNA ATANIR (BÖYLECE
;PORTB'YE ATANACAK BİLGİLER ÇIKIŞ OLARAK
;GÖRÜLÜR)
;BANK0'A GEÇİŞ
```

```
;B PORTUNA 1010 BİLGİSİ GONDER
;
;DİYEREK ÇIKIŞI TAMAMEN SIFIRLAYABİLİRİZ
```

Program Bloğu

DONGU

```
GOTO DONGU
END
```

```
;SONSUZ LOOP
```

Etiket Bloğu

Sonlandırma Bloğu

BAŞLIK BLOKU

- **LIST P=16F877;** Kullanılacak PIC tanımlanır.
- **INCLUDE P16F877.INC;** Kullanılan ;PIC'teki yazmaç tanımlamaları bu dosyada ;saklanır. Böylece yazmaç değerlerinin teker ;teker tanımlanması gerekmez.

BÜYÜK/KÜÇÜK HARF KULLANIMI

- PIC komutlarının büyük veya küçük harfle yazılması önemli değildir.

movlw = MOVLW = MoVlW

- Ancak etiketler harf büyüklüğüne duyarlıdır.

dongu≠DONGU

KULLANILAN SAYI/KARAKTER TİPLERİ

■ Hexadecimal

STATUS	EQU	0x03
		3
		03
		03h
		h' 03'
MOVLW	0x03	
	h' 03'	
	03	
	03h	

■ Binary

MOVLW	b' 00010110'
-------	--------------

■ Decimal

MOVLW	d' 15'
-------	--------

■ ASCII Character

RETLW	'A'
-------	-----

ÖRNEK PROGRAM

```
LIST                P=16F877
INCLUDE            P16F877.INC

ORG                0                ;PROGRAMI 0X00'DAN BAŞLAT

BSF                STATUS,RP0        ;BANK1'E GEÇİŞ
CLRF               TRISD             ;TRISD ÇIKIŞ MODUNA ATANIR
BCF                STATUS,RP0        ;BANK0'A GEÇİŞ

MOVLW              b'01100110'       ;D PORTUNA 01100110 BİLGİSİ GONDER
MOVWF              PORTD

DONGU
GOTO DONGU          ;SONSUZ LOOP
END
```

ÖRNEK PROGRAM

```
LIST P=16F877
INCLUDE P16F877.INC
__CONFIG_CP_OFF & _WDT_OFF & _PWRTE_ON &
_XT_OSC
```

```
#DEFINE      DSAYI      0xFF
CNTKH      EQU      0x25
CNTKL      EQU      0x26
CNTKM      EQU      0x27
```

```

ORG      0x00
BASLA
    CLRF  PORTA
    CLRF  PORTB
    CLRF  PORTC
    CLRF  PORTD
    CLRF  PORTE

    BSF   STATUS,RP0
    CLRF  TRISB
    CLRF  TRISC
    CLRF  TRISD
    CLRF  TRISE
    BCF   STATUS,RP0

```

```
CALL    BEK
MOVLW   b'01010101'
MOVWF   PORTB
CALL    BEK
MOVLW   b'11001100'
MOVWF   PORTC
```

```
CALL    BEK
MOVLW   DSAYI
MOVWF   PORTD
CALL    BEK
MOVLW   0x05
MOVWF   PORTE
CALL    BEK
CALL    BEK
GOTO    DONGU
```

```

BEK
    MOVLW    0X05
    MOVWF    CNTKM
TEKM
    MOVLW    0XF0
    MOVWF    CNTKH
TEK1
    MOVLW    0XFF
    MOVWF    CNTKL
TEK2
    DECFSZ   CNTKL, F
    GOTO     TEK2
    DECFSZ   CNTKH, F
    GOTO     TEK1
    DECFSZ   CNTKM, F
    GOTO     TEK2
    RETURN

```

```
DONGU
      GOTO BASLA
      END
```