

## PIC Mikrodenetleyicileri

Intel 1976 da 8031/51 ailesini piyasaya sürdüğünde dünyanın en popüler mikro işlemcisi olmuştu. Bu işlemci dünya üzerinde 12 den fazla firma tarafından (Intel, Phillips, Dallas, Siemens, Oki, Temic, Harris,...) milyonlarca adet üretildi. 8 bit tabanlı olan 64Kbyte a kadar adresleme yapabilen, 32 adet I/O portuna sahip bu işlemciler kendi endüstri standartını oluşturdu. 8031 ailesinin en önemli avantajı ise Eprom programlamanın zorlukları , çalışması için harici entegrelere ihtiyaç duyması idi. Gerçi 8051 de bu sorun yok ama onda da 8031 e göre fiyat avantajı var.

Bu makalede RISC mimarisi ile üretilen ve 8031 ailesine göre daha üstün özelliklere sahip olan (Fiyat,Programlayıcılar,Derleyiciler,Kaynak yeterliliği,...) Microchip in PIC serisi mikro işlemcilerinden en popüler olan 16C84/16F84 tanıtılacaktır.

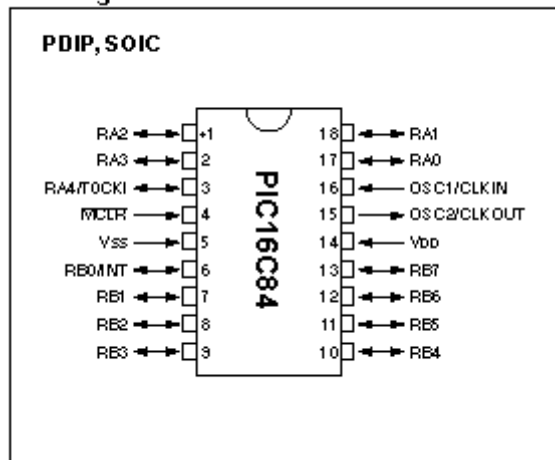
### PIC16F84 Microcontroller:

#### Genel Özellikler

- Sadece 35 komut ile programlama
- 13 adet giriş/çıkış portu
- Çalışma hızı DC-10 MHz
- 8 bit data
- 1Kx14 EEPROM Program Hafızası
- 64 Byte EEPROM Data Hafızası
- Direkt/Dolaylı Adresleyebilme
- 4 adet Kesme Fonksiyonu. (PBO,TMR0,RB Change,EEPROM Write)
- 1 milyondan fazla yazma silme
- Kolay ve ucuz programlayabilme
- Herbir pinden 25 mA e kadar akım verebilme
- 8 bit programlanabilir Timer

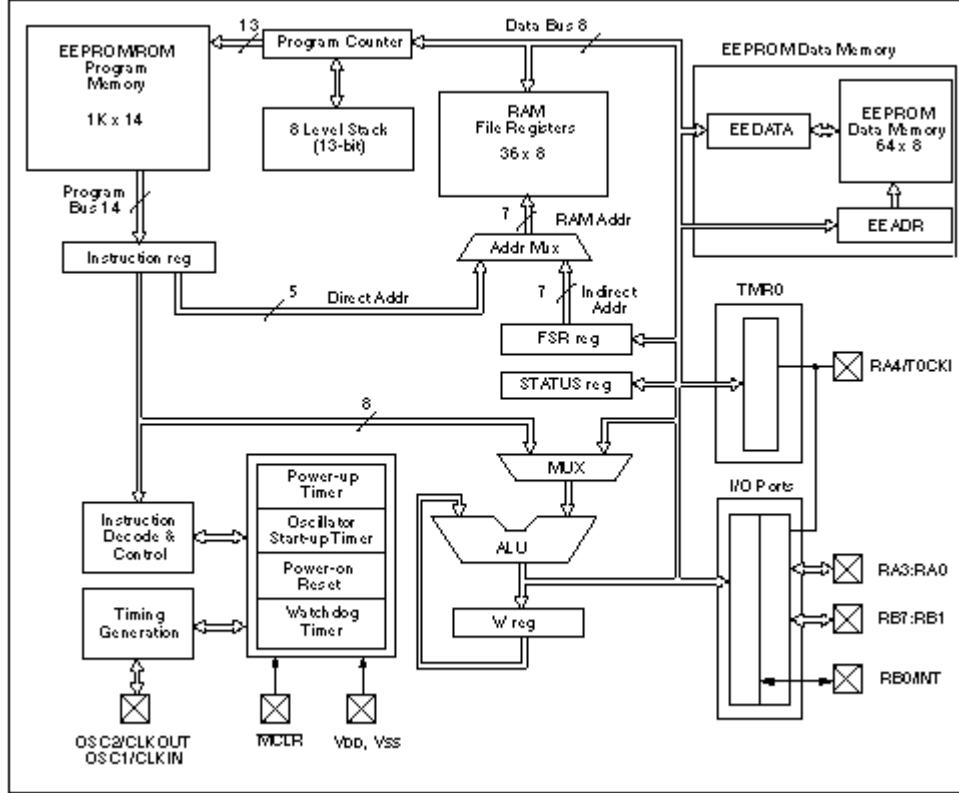
Aşağıda microcontroller in pin diagramı verilmiştir.

#### Pin Diagram



Şekil-1 : 16C84 Pin Diagram

16F84 13 adet giriş çıkış portu bulunmaktadır. Bunlardan 8 bitlik olanına PortB,5 bitlik olanına da PortA denir. PortA nın 4 numaralı pini open kollektör olup giriş çıkış yapmak için bir direnç ile +5 e çekmek gerekmektedir. Diğer portlardan hiçbir ek elemana ihtiyaç duymadan giriş/çıkış yapılabilir. Microcontrollerin mimarisi şekil-2 de verilmiştir.



**Şekil-2 : Mimari**

Microcontroller ı programlamadan önce Register haritasını bilmemiz gerekmektedir.

PIC mimarisinde genelde tüm işlemler W (work) register üzerinden yapılır. Program yazmadan önce hangi porttan giriş hangi portttan çıkış yapılacağını belirlenip bunun microcontroller e anlatılması gerekmektedir. Bu da TrisA ve TrisB register larına degerler atayarak olur. Şekil-3 de 16F84 ün register haritası görülmektedir.

File Address		File Address	
00h	Indirect addr. <sup>(1)</sup>	Indirect addr. <sup>(1)</sup>	80h
01h	TMR0	OPTION	81h
02h	PCL	PCL	82h
03h	STATUS	STATUS	83h
04h	FSR	FSR	84h
05h	PORTA	TRISA	85h
06h	PORTB	TRISB	86h
07h			87h
08h	EEDATA	EECON1	88h
09h	EEADR	EECON2 <sup>(1)</sup>	89h
0Ah	PCLATH	PCLATH	8Ah
0Bh	INTCON	INTCON	8Bh
0Ch			8Ch
	36 General Purpose registers (SRAM)	Mapped (accesses) in Bank 0	
2Fh			AFh
30h			B0h
7Fh			FFh
	Bank 0	Bank 1	

Unimplemented data memory location; read as '0'.  
 Note 1: Not a physical register.

### Şekil-3: Register Haritası

Şekil-3 e dikkat edilirse Bank0 ve Bank1 olarak 2 bölme vardır. Bank1 de bilmemiz gereken en önemli registerlar Option, TrisA ve TrisB dir. Bank0 da ise Status, PortA, PortB , TMR0 ve Intcon dur. Bu register ların bit-bit açıklaması ise şekil-4 de verilmiştir.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-on Reset	Value on all other resets (Note3)		
<b>Bank 0</b>													
00h	INDF	Uses contents of FSR to address data memory (not a physical register)								----	----		
01h	TMR0	8-bit real-time clock/counter								xxxx	xxxx	uuuu	uuuu
02h	PCL	Loworder 8 bits of the Program Counter (PC)								0000	0000	0000	0000
03h	STATUS <sup>(2)</sup>	IRP	RP1	RP0	TÖ	PD	Z	DC	C	0001	xxxx	000q	quuu
04h	FSR	Indirect data memory address pointer 0								xxxx	xxxx	uuuu	uuuu
05h	PORTA	—	—	—	RA4/T0CKI	RA3	RA2	RA1	RA0	---x	xxxx	---u	uuuu
06h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0/MNT	xxxx	xxxx	uuuu	uuuu
07h		Unimplemented location, read as '0'								----	----	----	----
08h	EEDATA	EEPROM data register								xxxx	xxxx	uuuu	uuuu
09h	EEADR	EEPROM address register								xxxx	xxxx	uuuu	uuuu
0Ah	PCLATH	—	—	—	Write buffer for upper 5 bits of the PC <sup>(1)</sup>				---	0000	---	0000	
0Bh	INTCON	GIE	EEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000	000x	0000	000u
<b>Bank 1</b>													
80h	INDF	Uses contents of FSR to address data memory (not a physical register)								----	----	----	----
81h	OPTION	REPT	INTE DG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111	1111	1111	1111
82h	PCL	Loworder 8 bits of Program Counter (PC)								0000	0000	0000	0000
83h	STATUS <sup>(2)</sup>	IRP	RP1	RP0	TÖ	PD	Z	DC	C	0001	xxxx	000q	quuu
84h	FSR	Indirect data memory address pointer 0								xxxx	xxxx	uuuu	uuuu
85h	TRISA	—	—	—	PORTA data direction register				---	1111	---	1111	
86h	TRISB	PORTB data direction register								1111	1111	1111	1111
87h		Unimplemented location, read as '0'								----	----	----	----
88h	EECON1	—	—	—	EEIF	WREERR	WREN	WR	RD	---	xxxx	---	q000
89h	EECON2	EEPROM control register 2 (not a physical register)								----	----	----	----
0Ah	PCLATH	—	—	—	Write buffer for upper 5 bits of the PC <sup>(1)</sup>				---	0000	---	0000	
0Bh	INTCON	GIE	EEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000	000x	0000	000u

Legend: x = unknown, u = unchanged. - = unimplemented read as '0', q = value depends on condition.

Note 1: The upper byte of the program counter is not directly accessible. PCLATH is a slave register for PC<12:8>. The contents of PCLATH can be transferred to the upper byte of the program counter, but the contents of PC<12:8> is never transferred to PCLATH.

2: The TÖ and PD status bits in the STATUS register are not affected by a MCLR reset.

3: Other (non power-up) resets include: external reset through MCLR and the Watchdog Timer Reset.

## Şekil-4 : Özel Register ların açıklamaları.

Şimdi bu register ları kısa kısa açıklayalım.

**PORTA:** 5 adet giriş/çıkış pini vardır.

**PORTB:** 8 adet giriş/çıkış pini vardır.

**TRISA:** PortA dan girişmi yoksa çıkışmi yapılacağını belirleyen register dir. İlgili gözelerle 1 yüklenirse Giriş, 0 yüklenirse Çıkış yapılır. Örnek olarak

```
MOVLW b'00000011'
```

```
MOVWF TRISA
```

Bu komut satırları ile PortA nın 0,1 nolu pinleri Giriş, 2,3,4 nolu pinleri ise çıkış olarak tanımlanmış olur.

**TRISB:** TrisA ile aynı.

## STATUS:

R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
IRP	RP1	RPO	TO	PD	Z	DC	C

bit7 bit0

R = Readable bit  
W = Writable bit  
U = Unimplemented bit,  
read as '0'  
- n = Value at POR reset

- Bit7:IPR            0: Bank0-1  
                      1: Bank2-3
- Bit6-5:RPO-RP1   00:Bank0  
                      01:Bank1  
                      10:Bank2  
                      11:Bank3
- Bit4:TO            Süre Aşım Biti  
                      0:WDT süre aşım işlemi yapmışsa  
                      1:CRWDT yada SLEEP yapılmışsa
- Bit3:PD            Güç Kesme Biti  
                      0:SLEEP den sonra  
                      1:CLRWDT dan sonra
- Bit2:Z             0:Yapılan aritmetik işlemin sonucu sıfırdan farklı  
ise
- 1: Yapılan aritmetik işlemin sonucu sıfır ise
- Bit1:DC            1: 8 bitin Düşük anlamlı ilk 4 bitinden taşma olursa  
                      0: Taşma olmazsa (ADD komutları ile değişir)
- Bit0:C             1:8 bitten taşma olursa  
                      0:Taşma olmazsa(ADD komutları ile değişir)

## OPTION:

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0

bit7 bit0

R = Readable bit  
W = Writable bit  
U = Unimplemented bit,  
read as '0'  
- n = Value at POR reset

- Bit7:RBPU        0: PortB Pull-Up disable  
                      1: PortB Pull-Up enable
- Bit6:INTEDG      0:PB0 düşen kenarda interrupt kabul eder.  
                      1:PB0 yükselen kenarda interrupt kabul eder.
- Bit5:T0CS        0:T0CKI den gelen darbeler Clock kabul edilir  
                      1:Clock kaynağı XTAL in ürettiği dahili darbeler olur.
- Bit4:T0SE        (T0CS=1 ise)  
                      0:T0CKI den gelen darbeler yükselen kenarda clock u  
                      arttırır.  
                      1: T0CKI den gelen darbeler düşen kenarda clock u  
                      arttırır.

Bit3:PSA	0:Prescaler TMR0 için ayarlı		
	1:Prescaler WDT için ayarlı		
Bit2,1,0:	Prescaler set etme		
	PS2,PS1,PS0	TMR0	WDT
	000	1:2	1:1
	001	1:4	1:2
	010	1:8	1:4
	011	1:16	1:8
	100	1:32	1:16
	101	1:64	1:32
	110	1:128	1:64
	111	1:256	1:128

Bu kadar bilgiden sonra artık sizde microcontroller inizi Programlayabilirsiniz. Örnek devreler için örnekler sayfasını görebilirsiniz. Sorularınız için bana mail gönderiniz.

Iyi çalışmalar dileği ile....

**Oktay Sirrik**

Web : <http://www.turkengineers.com>

e-mail : [oktay@turkengineers.com](mailto:oktay@turkengineers.com)