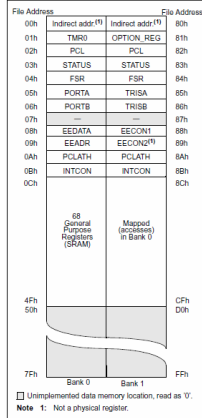


TIMER

- Mikrodenetleyicilerde bir veya birden fazla TIMER olabilir.
- TIMER olup kesmesi (interrupt yani T0IF gibi) olmayabilir. Örneğin 16F54
- 16F84 de T0IF vardır.

16F84

FIGURE 2-2: REGISTER FILE MAP - PIC16F84A



16F628A

OPTION REGISTER (ADDRESS 81h)

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
RBP0	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7							bit 0

- bit 7 **RBP0**: PORTB Pull-up Enable bit
1 = PORTB pull-ups are disabled
0 = PORTB pull-ups are enabled by individual port latch values
- bit 6 **INTEDG**: Interrupt Edge Select bit
1 = Interrupt on rising edge of RB0/INT pin
0 = Interrupt on falling edge of RB0/INT pin
- bit 5 **T0CS**: TMR0 Clock Source Select bit
1 = Transition on RA4/T0CKI pin
0 = Internal instruction cycle clock (CLKOUT)
- bit 4 **T0SE**: TMR0 Source Edge Select bit
1 = Increment on high-to-low transition on RA4/T0CKI pin
0 = Increment on low-to-high transition on RA4/T0CKI pin
- bit 3 **PSA**: Prescaler Assignment bit
1 = Prescaler is assigned to the WDT
0 = Prescaler is assigned to the Timer0 module
- bit 2-0 **PS2:PS0**: Prescaler Rate Select bits
- | Bit Value | TMR0 Rate | WDT Rate |
|-----------|-----------|----------|
| 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 |

- Option registerinin ilk 3 biti dikkate alınacak
- Örneğin ilk üç bit 000 ise her 2 saykılada TMR0 1 artar.
- Örneğin ilk üç bit 101 ise her 64 saykılada TMR0 1 artar.
- Örneğin ilk üç bit 111 ise her 256 saykılada TMR0 1 artar.

- TIMER 0'dan başlayıp 255 e kadar sayar sonra tekrar 0'a döner bu esnada kesme oluşturur. Bu kesme INTCON registerinin 2. bitidir (T0IF biti). Eğer bu bit 1 ise süre aşılmış.
- 0 ise süre henüz aşılmamıştır.

FIGURE 2-3: INTCON REGISTER (ADDRESS 0Bh, 8Bh)

	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
	GIE	EEIE	T0IE	INTE	RBIE	T0IF	INTF
bit 7							bit 0

bit 7 **GIE:** Global Interrupt Enable bit
1 = Enables all unmasked interrupts
0 = Disables all interrupts

bit 6 **EEIE:** EE Write Complete Interrupt Enable bit
1 = Enables the EE Write Complete Interrupts
0 = Disables the EE Write Complete Interrupts

bit 5 **T0IE:** TMR0 Overflow Interrupt Enable bit
1 = Enables the TMR0 Interrupt
0 = Disables the TMR0 Interrupt

bit 4 **INTE:** RB0/INT External Interrupt Enable bit
1 = Enables the RB0/INT external interrupt
0 = Disables the RB0/INT external interrupt

bit 3 **RBIE:** RB Port Change Interrupt Enable bit
1 = Enables the RB port change interrupt
0 = Disables the RB port change interrupt

bit 2 **T0IF:** TMR0 Overflow Interrupt Flag bit
1 = TMR0 register has overflowed (must be cleared in software)
0 = TMR0 register did not overflow

bit 1 **INTF:** RB0/INT External Interrupt Flag bit
1 = The RB0/INT external interrupt occurred (must be cleared in software)
0 = The RB0/INT external interrupt did not occur

bit 0 **RBIF:** RB Port Change Interrupt Flag bit
1 = At least one of the RB7:RB4 pins changed state (must be cleared in software)
0 = None of the RB7:RB4 pins have changed state

- Timer ayarlama bank1'de yapılır. Örneğin 256 saykıl da 1 artması için;

```
bsf    STATUS,5
movlw  0x07
movwf  OPTION_REG
bcf    STATUS,5
```

Önemli 1

- Eğer süre aşıldı ise kesme oluşur yani T0IF biti 1 olur. Timer hala saymaktadır. Biti kullanıcı sıfırlamalıdır. Eğer sıfırlamazsa hep 1 olarak kalır. Yani ikinci süre aşımında gene 1'i 1 yapar.

Önemli 2

- Kesme bayrağını (T0IF) sıfırlayınca timer sıfırlanmaz. Örneğin TMR0 süreyi aştı ve T0IF 1 oldu. TMR0 0'dan başlayıp tekrar saymaya devam eder. Örneğin TMR0 34'te iken biz T0IF i kontrol edip 1 olduğunu görüp süre aştığını anlarız ve tekrar T0IF'i 0 yaparız. Bu arada timer 34-35-36-.. diye saymaya devam edecektir.

Önemli 3

- TMR0 istenildiği zaman sıfırlanabilir. Örneğin 75, 76, 77,... diye sayarken arada "clrf TMR0" komutu ile sıfırlanınca sayma 0,1,2,3,4,... diye devam edecektir.

Soru

- Osilatör frekansı 4 Mhz dir. Option Registerin ilk üç biti 110 olarak ayarlanmıştır. TIMER kaç saniyede bir kesme oluşturur.

Çözüm

- 1 saykıl=1us (4 Mhz için)
- Option register 110 ise 128 saykılda TMR0 1 artar.
- TMR0 0-255 arası sayar yani 256 sayı sayar. Her 128 saykılda 1 artıyorsa 256 sayı için
- $128 \times 256 = 32768$ saykıl = 32768 us
- =32,768 ms yapar

TIMER'ı neden kullanırız.

Soru

- Osilatör frekansı 4 Mhz dir. Option Registerin ilk üç biti 111 olarak ayarlanmıştır. TIMER kaç saniyede bir kesme oluşturur.

Çözüm

- 1 saykıl=1us (4 Mhz için)
- Option register 111 ise 256 saykılda TMR0 1 artar.
- TMR0 0-255 arası sayar yani 256 sayı sayar. Her 256 saykılda 1 artıyorsa 256 sayı için
- $256 \times 256 = 65536$ saykıl = 65536 us
- =65,536 ms yapar

Soru

- Timer kullanarak yaklaşık 1 saniyelik gecikme oluşturunuz.(4 Mhz kristal osilatör bağlı)

Çözüm

- Option Registerin ilk üç biti 111 seçilirse her kesme yaklaşık 65 ms sürüyor. Bir önceki soruda çözüldü. 1 saniye için $1s=1000ms=1000/65=15$ defa kesme üretilmeli (yaklaşık değer)
- Her kesme oluştuğunda 15 den geriye doğru sayacak. Kullanılacak komut decfsz sayac,1 ve btfsc INTCON,2
- Öncelikle sayaca 15 yükleyelim.
movlw .15
movwf sayac

Basla

```
movlw .15  
movwf sayac  
clrf TMR0
```

Dongu

```
btfss INTCON,2  
goto Dongu; kesme oluşmadı  
bcf INTCON,2;kesmeyi sıfırla yoksa  
;sonraki kesmeyi anlayamazsın  
decfsz sayac,1  
goto Dongu  
goto suredoldu
```

Soru

- Kesme bitini kontrol etmeden Timer'ın süreyi aşıp aşmadığını nasıl kontrol ederiz. (çünkü PIC16F54 gibi mikrodenetleyicide Timer var ama kesmesi yoktur)

Çözüm

- Çıkarma işlemi yapılır. Örneğin Timer değeri 250 den çıkarılır. Timer 250 den büyükse sonuç negatif çıkacak ve C biti 0 olacaktır. Buradan Timer ın 250 değerini aştığını anlarız.
- Burada 250 tercih etmemizin nedeni (255 seçmedik) sürekli kontrol etmiyor ve arada başka işlemlerde yapıyorsak Timer'ın değerini kaçırabiliriz, yani timer 0 a dönebilir.

1. Yöntem

```
dongu  
movlw TMR0  
sublw .250; 250-TMR0  
btfsc STATUS,0  
goto dongu  
xxxx
```

2. Yöntem

```
movlw .250  
movwf sayi  
dongu  
movf TMR0,0;TMR0'ı W'ye ata  
subwf sayi; sayi-TMR0  
btfsc STATUS,0  
goto dongu  
xxxx
```