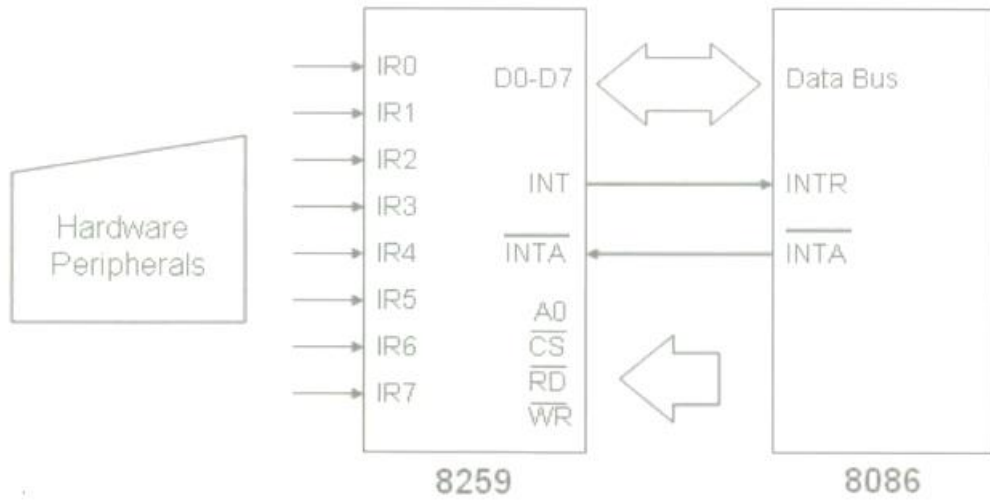


## KESMELİ GİRİŞ/ÇIKIŞ

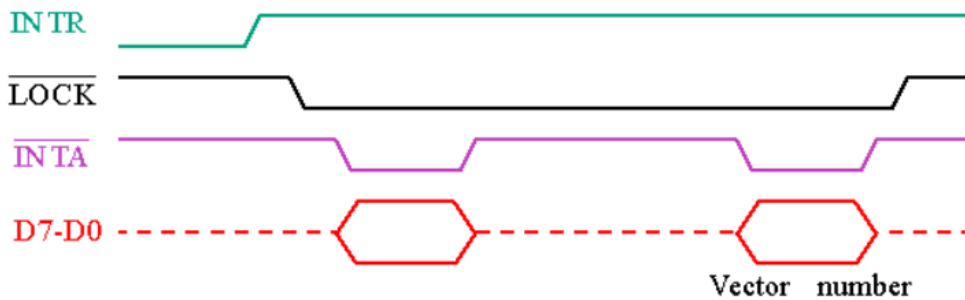
8259 PIC (Programmable Interrupt Controller) ve 8086 CPU tümleşik devrelerin sinyal akışı ve bacak bağlantıları Şekil-1'deki gibidir. 8259'un IR0-IR7 kesme isteği (Interrupt Request) uçlarına bağlanabilen donanımsal çevrebirim, bir servis talebinde bulunduğunda "Interrupt Request" ucu aktif hale gelir. PIC'te kesmeler yetkilendirilmiş ise bu istek 8259'un INT ucundan 8086'ın INTR ucuna aktarılır. 8086 o anda koşmakta olduğu emirin icrasını tamamladıktan sonra INTA ucundan bir darbe göndererek bu kesme isteğini onaylar (acknowledge).

8259, 8086'dan ikinci INTA sinyalini aldığı anda kesme istek numarasını (Interrupt Request Number) data bus üzerinden 8086'ya göndererek hangi cihazın kesme isteğinde bulunduğunu tanımlar. Bu kesme istek numarası 8086'nın içinde 4 ile çarpılır ve 20-bitlik adrese dönüştürüldükten sonra kesme vektör tablosuna uygulanır. Tablodan alınan CS ve IP adres bilgileri işlenerek servis programının fiziksel adresi bulunur.

Servis programından geri dönülmeden önce servis programı, EOI (End-of-Interrupt) komutunu 8259'a göndererek servis programının bittiğini bildirir. Böylece PIC'de bu istekten daha düşük öncelikli istekler yetkilendirir.



**Şekil-1:** 8259 ve 8086 tümleşik devrelerin sinyal akışı ve bacak bağlantıları

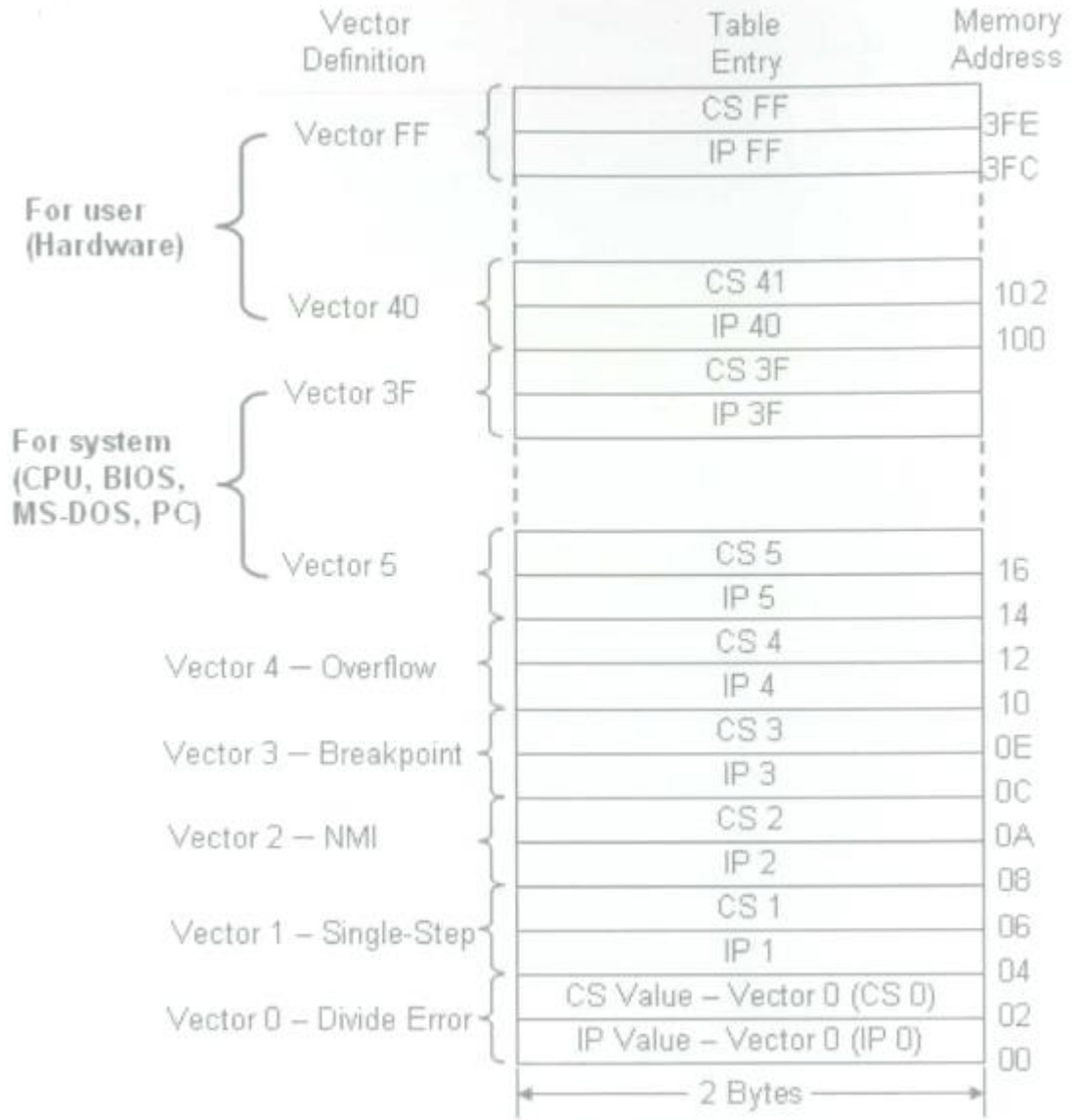


8259'un iki tip kontrol kelimesi vardır: Initialization Control Words (ICWs) ve Operation Control Words (OCWs). Kontrol kelimeleri; A0, CS, RD ve WR uçlarının yardımıyla data bus üzerinden gönderilir. A0=0 yapılarak ICW1, OCW2, OCW3 yazmaçlarına erişilebilir. A0=1 yapılarak ise ICW2, ICW3, ICW4, OCW1 kaydedicilerine erişilebilir ( Tablo-1).

**Tablo-1:** 8259 kaydedicilerinin adreslenmesi

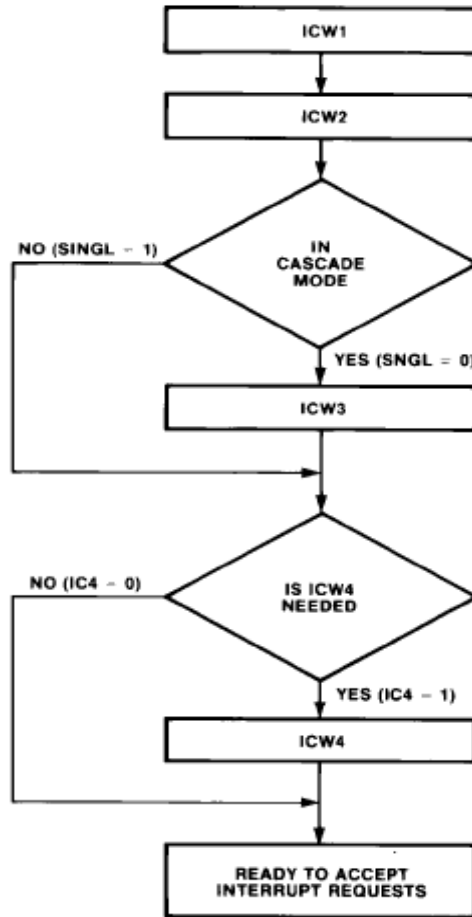
CS	RD	WR	A0	8259 Operation	8086 Operation
0	1	0	0	Data bus to ICW1 register Data bus to OCW2 register Data bus to OCW3 register	ICW1 write OCW2 write OCW3 write
0	1	0	1	Data bus to ICW2 register Data bus to ICW3 register Data bus to ICW4 register Data bus to OCW1 register	ICW2 write ICW3 write ICW4 write OCW1 write

Şekil-2'de gösterildiği gibi kullanıcı interruptlarının kesme vektör tablosundaki yeri 0000:0100 (Type 40 Interrupt, 40H\*4) adresinden başlamaktadır. CPU, IRQ isteğini aldığı anda o anda icra edilen program durdurulur ve IRQ kesme servis rutinini icra edilir. Kesme servis programının başlangıç adresi (CS:IP) kesme vektöründe tutulur.



**Şekil-2:** Kesme vektör tablosu

Şekil-3’de 8259’u ilk kullanıma hazırlama dizisi verilmiştir. İlk kullanıma hazırlama kelimeleri sırasıyla yazılır. Tablo-1’den de görüleceği gibi ICW2-ICW4’nin hepsi aynı port adresine yazılır. Çünkü 8259, bu kaydedicilerin sırasıyla yazılmasını beklemektedir. ICW3 ve ICW4 kaydedicileri opsiyoneldir.



**Şekil-3:** 8259 ilk kullanıma hazırlama dizisi

ICW kelimelerinin formatı Şekil-4~7'de görülmektedir.



**Şekil-4:** ICW1 register format



**Şekil-5:** ICW2 register format



**Şekil-6:** ICW3 register format



**Şekil-7:** ICW4 register format

*Special fully nested mode (SFNM):* Bu mod, "cascade" bağlantı yapısındaki master 8259 tarafından kullanılır. Öncelik yapısı "fully nested mode" daki gibidir, yani IR0 en yüksek önceliğe, IR7 ise en düşük önceliğe sahiptir. Bu modda master sadece, slave'den gelen (slave'in o esnada başka kesmesi hizmet almaktadır) daha yüksek öncelikli kesmelere hizmet eder.

ICW1=00010011: 80x86 modu; kenar tetikleme; tek bir 8259; ICW4'e ihtiyaç var (D7-D5 ve D2 bitleri sadece 8-bitlik 8085 mikroişlemcisi kullanıldığında gereklidir. D1, single mod için 1 yapılır.

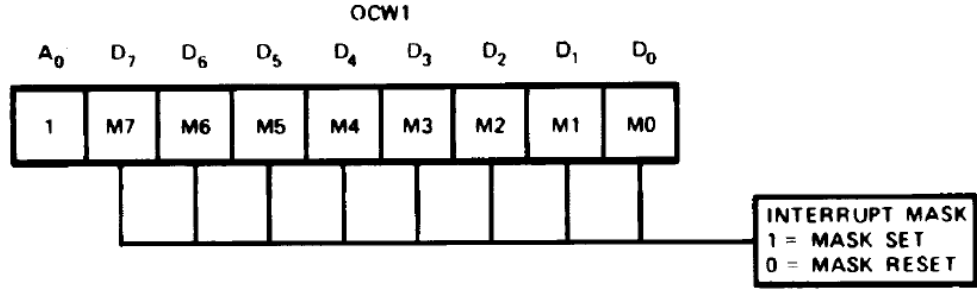
ICW2=01000000: Kesme vektör adresi 40h yapılır.  
(8 dış cihazın hangisinden isteğin (IR0-IR7) geldiğine bağlı olarak ICW2'nin en anlamsız üç biti 000 -111 arasında değerler alır.)

ICW4=00000101: Normal nesting; Non-buffer; Manual EOI; 80x86 mod  
(Normal nesting modda IR0 en yüksek, IR7 ise en düşük önceliğe sahiptir. Manual EOI seçilmişse, kesme servis rutininden "return" ile dönüş yapılmadan önce, diğer kesmelerin kabul edilebilmesi için EOI komutunun yayımlanması gerekir.

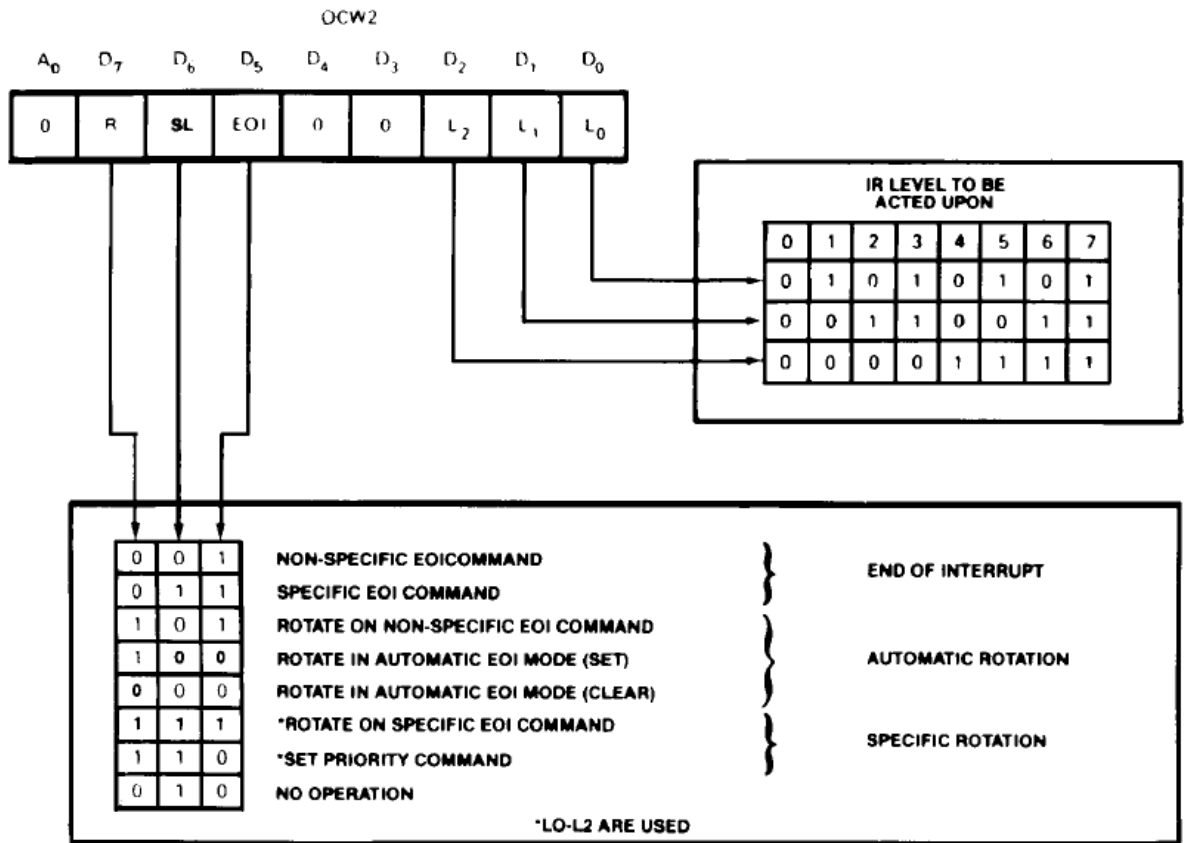
İklendirme tamamlandıktan sonra, yani çalışma moduna geçilince operasyon kontrol kelimeleri herhangi bir anda herhangi bir sırada gönderilebilir. OCW'ler kesme maskesini, dönen öncelik modunu, özel maske modunu, yoklamalı modu ve EOI komutlarını tayin eder. OCW1 ve OCW2'nin formatları Şekil-8-9'de verilmiştir.

OCW1: M7-M0=11111110 kelimesi bit0'ın haricindeki istekleri maskeler. M7-M0, ilgili kesme hattının maske bitleridir. Biti setleme maskeyi etkinleştirir ve dolayısıyla ilgili kesmeyi engeller.

OW2: D7-D0=00100000 kelimesi 8259 denetleyicisinin Nonspecific EOI komutu altında çalışmasını sağlar.



Şekil-8: OCW1 register format



Şekil-9: OCW2 register format

*Automatic rotation mode* (Otomatik döndürme modu): Çeşitli kesme kaynakları aynı önceliğe sahip olduğunda kullanılır. Bu modda, bir cihaza servis verildikten sonra en düşük önceliği alır. Diğer tüm öncelikler buna göre döner. Örnek: IR4'e yeni hizmet verildiyse, en düşük önceliği alacaktır.

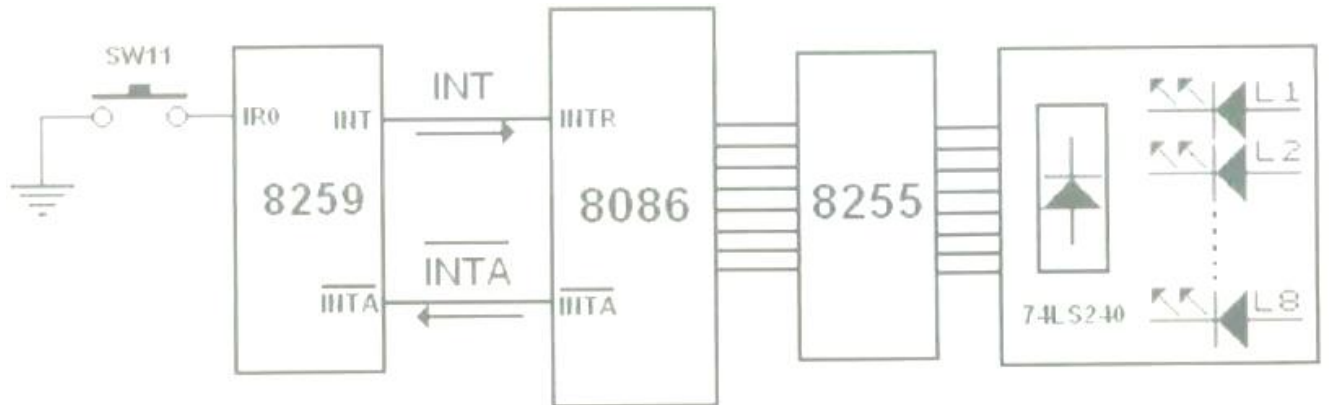
## DENEYİN YAPILIŞI

**Amaç:** Servis programının amacı SW11 butonu ile her kesme isteği gönderildiğinde Şekil-10'da görülen yanan LED i bir bit sola kaydırmaktır.



Şekil-10: Programın çıktısı

**Devre tanımı:** SW11 butonu, 8259'un IR0 ucuna bağlıdır ve harici donanım kesmesi üretmek için kullanılmaktadır. 8259'dan kesme isteğini 8086 her aldığı anda LED'leri yakan bilgiyi bir bit sola kaydırarak 8255' gönderir.



Şekil-11: Fonksiyonel devre tanımı

## Kaynak Program:

```
=====8259.ASM=====
CNT      EQU  3FD6H      ;Define 8255 control word port address
BPORT    EQU  3FD2H      ;Define 8255 portB address
COMM1    EQU  0FFC8H     ;Define 8259 command address for ICW1, OCW2, OCW3
COMM2    EQU  0FFCAH     ;Define 8259 command address for ICW2, ICW3, ICW4, OCW1

CODE      SEGMENT
          ASSUME CS:CODE,DS:CODE
```

```

START:    CLI                ;Disable interrupt
          MOV  SP,4000H      ;Setup of stack pointer
          MOV  AX,CS         ;
          MOV  DS,AX         ; Data Segment= Code Segment

```

*;<Setup IR0 vector>*

```

MAIN:     MOV  AX,0          ;Initialize ES to '0'
          MOV  ES,AX         ;Extra Segment=0
          MOV  BX,40H*4      ;Setup vector address of IR0 to BX=40H*4
          MOV  ES:WORD PTR[BX],OFFSET INTRO
                                   ;IP of ISR write to vector
          MOV  ES:[BX+2],CS  ;CS of ISR write to vector

```

*;<Initialize Command Words>*

*;<ICW1>*

```

          MOV  DX,COMM1      ;Enable command1 port address
          MOV  AL,00010011B  ;Setup ICW1
          OUT  DX,AL

```

*;<ICW2>*

```

          MOV  DX,COMM2      ;Enable command2 port address
          MOV  AL,40H        ;Setup ICW2
          OUT  DX,AL

```

*;<ICW4>*

```

          MOV  AL,00000101B  ;Setup ICW4
          OUT  DX,AL

```

*;<Operation Command Words>*

*;<OCW1>*

```

          MOV  AL,11111110B  ;OCW1(Unmask IR0)
          OUT  DX,AL

```

*;<setup 8255 control word register & initialize LED>*

```

          MOV  DX,CNT        ;Enable 8255 control port
          MOV  AL,80H        ;Setup 8255 control word register
          OUT  DX,AL         ;AL=80h,define portA, portB, portC=>output
          MOV  AL,01H        ;initialize LED
          MOV  DX,BPORT      ;Enable 8255 portB
          OUT  DX,AL         ;Output data, LED=01h

```



```

        STI                                ;Enable Interrupt
        JMP  $                            ;Wait for IR0 interrupt
; <IR0 Interrupt Service Routine>
INTR0:  ROL  AL,1                        ;LED rotate left 1 bit
        OUT  DX,AL                      ;Output data to LED
        MOV  CX,5FFFH                   ;Setup CX register for time delay
        LOOP $
        PUSH AX                         ;Store AX
        PUSH DX                         ;Store DX
        MOV  DX,COMM1                   ;Enable 8259 COMM1 port adres (Setup OCW2)
        MOV  AL,20H                     ;EOI(End of Interrupt) command write to 8259
        OUT  DX,AL                      ;Output command
        POP  DX                         ;Restore AX
        POP  AX                         ;Restore DX
        IRET                            ;Return to Main routine

CODE    ENDS
        END  START

```