

19. 8251 USART (UNIVERSAL SYNCHRONOUS/ASYNCHRONOUS RECEIVER/TRANSMITTER)

8251 evrensel anuyumlu/anuyumsuz alıcı/verici (Universal Synchronous/Asynchronous Receiver/Transmitter - USART) olarak isimlendirilen, 8 bit veri genişliğine sahip, TTL uyumlu, G/Ç birimleri ile seri haberleşme için kullanılan bir programlanabilir arayüz tüm devresidir.

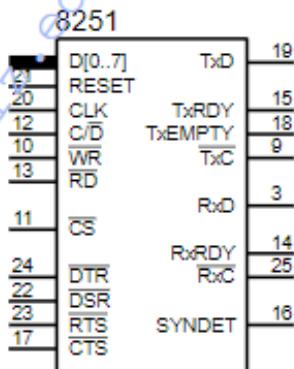
Genel kullanım amacı:

- anuyumsuz seri haberleşme
- anuyumlu seri haberleşme

olarak verilebilir.

19.1 8251 Uç Tanımları

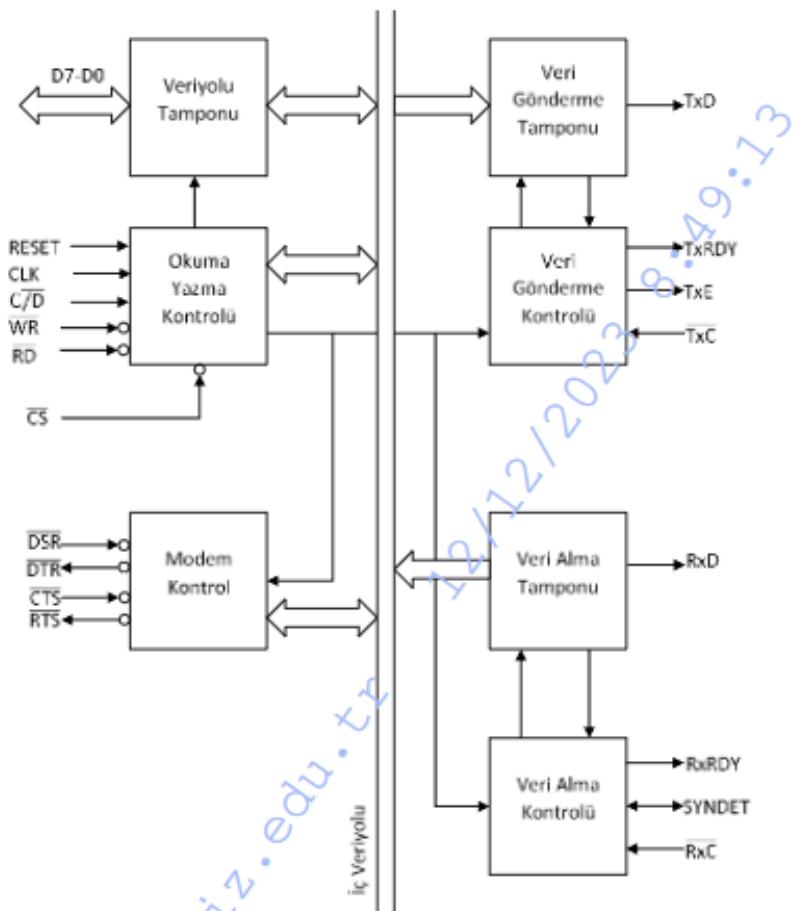
8251 fonksiyonel uç tanımları Şekil 19-1 ile verilmektedir. 8251, sekiz adet veri (D7-D0), bir adet adres (C/D), okuma kontrol (RD), yazma kontrol (WR), yetkilendirme (CS), saat darbesi girişi (CLK), modem kontrol (DTR, DSR, RTS, CTS), veri gönderme (TxD), veri alma (RxD), veri gönderme saat girişi (Tx C), veri alma saat girişi (Rx C), veri gönderme tamponu boş (Tx RDY), veri alma tamponu dolu (Rx RDY) uçlarına sahiptir.



Şekil 19-1: 8251 fonksiyonel uç tanımları

19.2 8251 İç Yapısı

8251 iç yapısı incelendiğinde veri alma tampon ve kontrolü, veri gönderme tampon ve kontrolü ile veri okuma tamponu, okuma/yazma kontrolü ve modem kontrolü blokları görülmektedir. Bu bloklar bir iç veri yolu üzerinden iletişim halindedir. 8251 iç yapısı Şekil 19-2 ile verilmektedir.



Şekil 19-2: 8251 iç yapısı

19.3 8251 Yazaçları ve Görevleri

C/D , RD , WR ve CS uclarında oluşturulan değerlere göre 8251 yazaçlarına erişim sağlanabilmektedir. C/D ucunda lojik 0 değeri oluşturulduğunda veri yazaçlarına, lojik 1 değeri oluşturulduğunda ise kontrol yazaçlarına erişim sağlanabilmektedir. Veri yazaçları giriş ve çıkış yönü olmak üzere iki ayrı yazaçtan oluşur. Veri yazaçları okuma yönü adreslendiğinde veri giriş yazmacına erişilirken, yazma yönü adreslendiğinde ise veri çıkış yazmacına erişilir. 8251'de yazma yönü erişilebilen kontrol yazaçları kip, kontrol, anuyumluluk-1 ve anuyumluluk-2 yazaçlarıdır. Okuma yönü erişilebilen kontrol yazacı ise durum yazacıdır. 8251 yazaçlarına erişim Tablo 19-1 ile özetlemiştir. Tablo 19-1 incelediğinde bir adres ucuna karşı mikroişlemci tarafından erişilebilecek yedi farklı yazaç görülebilir.

Tablo 19-1: 8251 yazmaç erişimi

-CS	C/D	-RD	-WR	Yazmaç Erişimi
1	x	x	x	Veri yolu yüksek empedans gösterir
0	x	1	1	Veri yolu yüksek empedans gösterir
0	1	0	1	Durum yazmaç okunur
0	1	1	0	Kip, Kontrol, Anuyumluluk-1 veya Anuyumluluk-2 yazmaçlarına değer yazılır
0	0	0	1	8251 veri yazmaç okunur
0	0	1	0	8251 veri yazmacına yazılır

8251 haberleşme açısından anuyumlu veya anuyumsuz kipte ayarlanabilir. Anuyumlu haberleşmede iki terminal arasında veri alma, veri gönderme, ortak saat darsbesi ve gerilim referans ucu olmak üzere dört uç üzerinden bağlantı sağlanır. Anuyumsuz haberleşmede ise veri alma veri gönderme ve gerilim referans ucu olmak üzere üç uç üzerinden bağlantı sağlanır. Seri haberleşme açısından terminaller arasında ortak bir saat darsbesi bulunması uzun bit serileri halinde veri aktarımına imkân sağlar. Anuyumlu haberleşmede veri transferinin başlangıç ve bitişleri de anuyumluluk yazmaçları ile ilişkili özel bit dizileri olarak tanımlanabilir. Anuyumsuz haberleşme de ise terminaller kendi iç saatlerini kullanarak veri aktarım uçlarını örnekle yerek peş peşe bit'leri okur. İç saatler arasında oluşabilecek frekans farkı anuyumsuz haberleşmenin kısa bit dizileri olarak kısıtlamasına sebep olur.

Anuyumlu veya anuyumsuz haberleşme 8251 kip yazmaç ile sağlanmaktadır. 8251 anuyumlu kipte kullanılmak istenirse kip yazmacının en düşük anlamlı bit'leri $(00)_2$ olarak ayarlanmalıdır (Tablo 19-2).

Tablo 19-2: 8251 kip yazmaç formatı (anuyumlu haberleşme için)

8251 Kip Yazmaç (Anuyumlu)							
D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
SCS	ESD	EP	PEN	L ₂	L ₁	0	0

SCS: Sync karakter sayısı: 0: 2 sync, 1: 1 sync

ESD: External sync detect: 0: SYNDET output, 1: SYNDET input.

Kitap kapsamında 8251 anuyumsuz haberleşme kipinde kullanılacaktır. Anuyumsuz haberleşmede önce başlangıç bit'i, sonrasında veri bit'leri, kullanılıyorsa eşlik bit'i ve son olarak durma bit(ler)i aktarılır. 8251'in anuyumsuz kip ayarına Tablo 19-3 ile yer verilmiştir. Anuyumsuz kipte 1, 1,5 veya 2 durma bit'ı²¹¹ kullanılabilir. Veri transferinde eşlik kontrolü, çift veya tek eşlik kontrolü olarak kullanılabilir. Veri 5, 6, 7 veya 8 bit üzerinden aktarılabilir. 8251 veri alma saat girişi ve veri gönderme saat girişine bağlı frekans değerinin hangi oranda bölünerek hangi baud oranının oluşturulacağı ise baud oran çarpan bit'leri ile sağlanır.

²¹¹ Durma bit sayısını süreye bağlı olarak düşünmek gereklidir. Örneğin 1,5 durma bit'i (1,5/baud oranı) sn. süreye denktir.

Tablo 19-3: 8251 kip yazmacı formatı (anuyumsuz haberleşme için)

8251 Kip Yazmacı (Anuyumsuz)							
D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
S ₂	S ₁	EP	PEN	L ₂	L ₁	B ₂	B ₁

S ₂	S ₁	Durma bit sayısı
0	0	Gecersiz
0	1	1 durma bit'i
1	0	1,5 durma bit'i
1	1	2 durma bit'i

EP	Eşlik (Parity)
0	Tek eşlik (odd parity)
1	Çift eşlik (even parity)

PEN	Eşlik etkinleştirme
0	Eşlik yok
1	Eşlik var

L ₂	L ₁	Veri bit sayısı
0	0	5
0	1	6
1	0	7
1	1	8

B ₂	B ₁	Baud oran çarpanı
0	0	Anuyumlu kip
0	1	1
1	0	16
1	1	64

8251'de kip yazmacını ayarlandıktan sonra kontrol yazmacının da uygun şekilde ayarlanması gerekmektedir. Kontrol yazmacı ile veri gönderme ve alma aktifleştirilebilir veya kapatılabilir. Yazılımsal olarak 8251 ilk durumuna döndürülebilir (reset). Son olarak daha önceki veri transferlerinde kalan hata bit'leri temizlenebilir (Tablo 19-4).

Tablo 19-4: 8251 kontrol yazmacı formatı

8251 Kontrol Yazmacı							
D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
EH	IR	RTS	ER	SBRK	RxE	DTR	TxE

IR : Yazılımsal reset

ER : Hata bit'lerini temizle

SBRK : Veri göndermeyi durdur ($TxD = 0$)

RxE : Veri almayı etkinleştir

TxE : Veri göndermeyi etkinleştir

Durum yazmacı program kontrollü veri akışında veri gönderme ve alma tamponlarının hazır olup olmadıklarına ilişkin kontrol bit'lerine sahiptir. Ayrıca çerçeve, ezme ve eşlik hata bit'leri de durum yazmacında sunulmaktadır (Tablo 19-5). 8251'in yeni bir veriyi dış dünyaya gönderebilmesi için bir önceki veri gönderme adımda yazılan verinin son bit'inin de gönderilmiş olarak veri gönderme tamponun boş hale gelmemelidir. 8251 vasıtasiyla dış dünyadan gelen yeni bir verinin okunabilmesi için bit dizisinin son bit'inin de okunarak veri alma tamponun dolu hale getirilmesidir.

Tablo 19-5: 8251 durum yazmacı formати

8251 Status Yazmacı							
D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
DSR	SYNDET	FE	OE	PE	TxE	RxDY	TxDY

FE: Çerçeve hatası

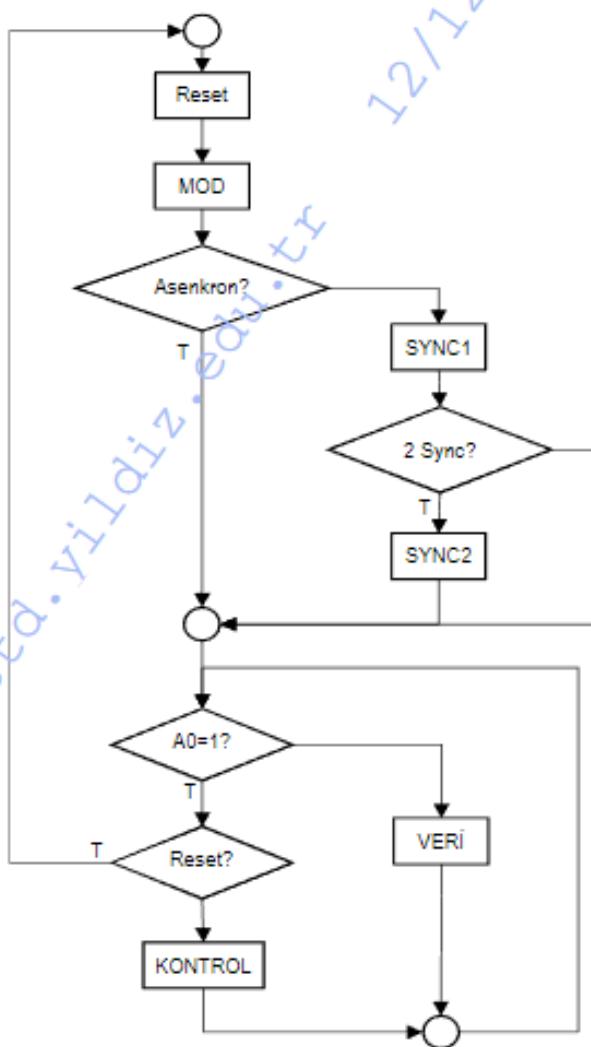
OE: Ezme hatası

PE: Eşlik hatası

RxRDY : veri alma tamponu dolu

TxRDY : veri gönderme tamponu boş

8251'de tek bir adres ucuyla yedi farklı yazmacın nasıl adreslenebileceği Şekil 19-3 ile verilmiştir. Anuyumsuz haberleşme açısından incelediğinde donanımsal veya yazılımsal reset işleminden sonra kontrol grubu yazmaclarına yapılan ilk yazma işlemi ile kip yazmacına, ikinci yazma işlemi ile kontrol yazmacına erişir.



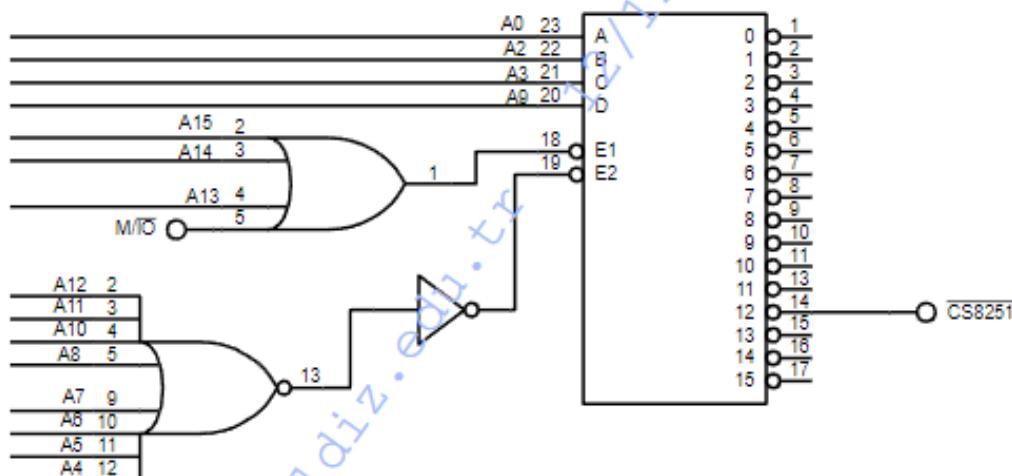
Şekil 19-3: 8251 kontrol akış diyagramı

19.4 8251 Örnek Uygulamaları

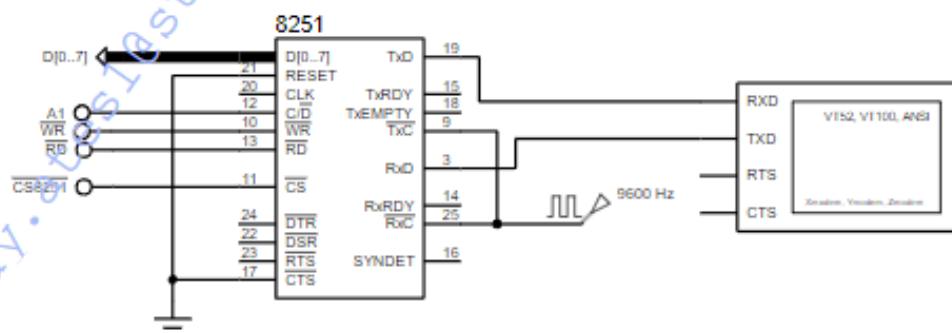
Örnek 19-1

8086 G/C uzayında, 208H adresinden itibaren ardışık çift adreslere 8251 yerleştirilmiştir. 8251'in RxD ve TxD uçları sırasıyla bir seri terminalin TxD ve RxD uçlarına bağlanmıştır. Seri terminal 9600 baud oranını kullanacak şekilde ayarlanmıştır. Buna göre:

- Gerekli adres çözümleme devresini 4x16 kodçözücü ve gerekli basit lojik kapılar kullanarak tasarluyınız.
- 8251'e yapılması gereken bağlantıları (adres çözümleme devresi, mikroişlemci dahil) gösteriniz.



Şekil 19-4: Örnek 19-1 için adres çözümleme devresi



Şekil 19-5: Örnek 19-1 için 8251 bağlantıları

19.4.1 Seri Veri Gönderme

Örnek 19-2

Örnek 19-1 ile verilen adres aralığına yerleştirilen 8251'in RxD ve TxD uçları sırasıyla bir seri terminalin TxD ve RxD uçlarına bağlanmıştır. Seri terminal 9600 baud oranında, sekiz veri bit'i kullanacak şekilde, eşlik bit kullanmayacak bir durma bit'i kullanacak şekilde ayarlanmıştır. Ayrıca bir lojik analizörün iki kanalı sırasıyla 8251 TxRDY ve TxD uçlarına bağlanmıştır. Buna göre:

- 8251 veri gönderme birimi uygun olduğunda peş peşe 'A' ASCII karakterini ('A' = 41H) gönderen assembly kodunu yazınız.
- Lojik analizör çıktısını çiziniz.

```

; PROGRAM : Örnek 19-2.ASM
; 8251 kullanarak sürekli 'A' gönderen
; assembly program
;

PAGE 60,80
TITLE 8251 seri veri gönderme
STAK SEGMENT PARA STACK 'STACK'
DW 20 DUP(?)
STAK ENDS
CODE SEGMENT PARA 'CODE'
ASSUME CS:CODE, SS:STAK
START PROC FAR ; ana yordam
MOV DX, 020Ah
MOV AL, 01001101B
OUT DX, AL ; kip yazmacına erişim
MOV AL, 40H
OUT DX, AL ; kontrol yazmacı ile 8251 reset'leme
MOV AL, 01001101B ; mod:1 stop bit'i, parity kullanmadan,
; 8 bit veri, 1 çarpan
OUT DX, AL ; kontrol: veri gönderme aktif,
MOV AL, 00010001B ; hata bit'lerini temizle
OUT DX, AL
ENDLESS: MOV DX, 020AH ; durum yazmacı adresi
TEKRAR: IN AL, DX ; durum yazmacını oku
AND AL, 01H ; TXRDY'yi kontrol et
JZ TEKRAR ; TXRDY 1 olmadığı sürece tekrarla
MOV DX, 0208H ; veri yazmacı adresi
MOV AL, 41H
OUT DX, AL ; 8251 ile 'A' karakteri gönderimi
JMP ENDLESS ; sonsuz döngü
START
CODE
END START

```

8251 ucunda TxRDY = 1 olmasıyla 'A' ASCII karakteri karşılığı veri gönderme tamponuna yazılır. Veri gönderme tamponuna veri yazılmasıyla TxRDY ucu lojik

0'a düşer. Bundan sonra 9600 baud oranında (~104 μ s aralıklarla) sırasıyla START (lojik 0), D0-D7 (1000 0010)₂, STOP bit'leri TxD ucundan gönderilir.



Şekil 19-6: Örnek 19-2 için lojik analizör çıktıları

19.4.2 Seri Veri Alma

Örnek 19-3

Örnek 19-1 ile verilen adres aralığına yerleştirilen 8251'in RxD ve TxD uçları sırasıyla bir seri terminalin TxD ve RxD uçlarına bağlanmıştır. Seri terminal 9600 baud oranında, sekiz veri bit'i kullanacak şekilde, çift eşlik kullanan, iki durma bit'i kullanacak şekilde ayarlanmıştır. 8251'e seri terminalden gönderilen ASCII karakterleri veri kesiminde tanımlanmış olan RECEIVED isimli beş elemanlı dizide round robin mantığında saklayan assembly kodunu yazınız.

```

; PROGRAM : Örnek 19-3.ASM
; 8251 kullanarak alınan ASCII karakterleri veri kesiminde
; tanımlanmış olan RECEIVED isimli beş elemanlı dizide round robin
; mantığında saklayan assembly program
;

PAGE 60,80
TITLE 8251 seri veri alma
STAK SEGMENT PARA STACK 'STACK'
DW 20 DUP(?)
STAK ENDS
DATA SEGMENT PARA 'DATA'
RECEIVED DB 5 DUP (0)
DATA ENDS
CODE SEGMENT PARA 'CODE'
ASSUME CS:CODE, DS:DATA, SS:STAK
START PROC FAR ; ana yordam
;
; DATA ismiyle tanımlı kesim alanına erişebilmek için gerekli tanımlar
;
MOV AX, DATA
MOV DS, AX
XOR SI, SI
MOV DX, 020Ah
MOV AL, 01001101B
OUT DX, AL ; kip yazmaç erişimi
MOV AL, 40H
OUT DX, AL ; kontrol yazmacı ile 8251 reset'teme
MOV AL, 11111101B ; 2 stop, çift eşlik, 8 bit
OUT DX, AL
MOV AL, 00010100B ; veri alma aktif, hata bit'lerini temizle
OUT DX, AL
ENDLESS: MOV DX, 020AH ; durum yazmacı adresi
TEKRAR: IN AL, DX ; durum yazmacını oku
AND AL, 02H ; RxRDY bit'ini kontrol et
JZ TEKRAR ; RxRDY 1 olmadığı sürece tekrarla
MOV DX, 0208H ; veri yazmacı adresi
IN AL, DX

```

```

        MOV RECEIVED[SI], AL
        INC SI
        CMP SI, 5
        JNE DEVAM
        XOR SI, SI
DEVAM:   JMP ENDLESS      ; sonsuz döngü
START    ENDP
CODE     ENDS
END START

```

19.4.3 Seri Veri Alma ve Gönderme

Örnek 19-4

Örnek 19-1 ile verilen adres aralığına yerleştirilen 8251'in RxD ve TxD uçları sırasıyla bir seri terminalin TxD ve RxD uçlarına bağlanmıştır. Seri terminal 9600 baud oranında, sekiz veri bit'i kullanacak şekilde, eşlik bit kullanmayan, bir durma bit'i kullanacak şekilde ayarlanmıştır. 8251'e seri terminalden gönderilen her bir ASCII karaktere karşılık bir fazlasını geri gönderen assembly kodunu yazınız.

```

; -----  

; PROGRAM : Örnek 19-4.ASM  

; 8251 kullanarak gelen seri ASCII karakterin bir fazlasını  

; gönderen assembly program  

;  

; -----  

; PAGE 60,80  

; TITLE 8251 seri veri alma ve gönderme  

STAK    SEGMENT PARA STACK 'STACK'  

        DW 20 DUP(?)  

STAK    ENDS  

CODE    SEGMENT PARA 'CODE'  

        ASSUME CS:CODE, SS:STAK  

START   PROC FAR          ; ana yordam  

        MOV DX, 020Ah  

        MOV AL, 01001101B  

        OUT DX, AL          ; kip yazmaç erişimi  

        MOV AL, 40H  

        OUT DX, AL          ; kontrol yazmacı ile 8251 reset'leme  

        MOV AL, 01001101B  

        OUT DX, AL          ; kip: 1 stop, parity yok,  

        ; 8 veri bit'i, 1 çarpan  

        MOV AL, 00010101B  

        OUT DX, AL          ; kontrol: veri alma ve gönderme aktif,  

        ; hata bit'lerini temizle  

ENDLESS: MOV DX, 020AH  

TEKRAR1: IN AL, DX  

        AND AL, 02H  

        JZ TEKRAR1  

        MOV DX, 0208H  

        IN AL, DX  

        INC AL  

        MOV BL, AL  

        MOV DX, 020AH  

TEKRAR2: IN AL, DX  

        AND AL, 01H  

        JZ TEKRAR2  

        ; durum yazmaç adresi  

        ; durum yazmacını oku  

        ; TxRDY bit'ini kontrol et  

        ; TxRDY 1 olmadığı sürece tekrarla  

        ; veri yazmacı  

        ; alınan veriyi oku  

        ; AL<-alınan verinin ASCII değer  

        ; olarak bir fazlası

```

```
MOV DX, 0208H ; veri yazmacı
MOV AL, BL
OUT DX, AL ; seri olarak alınan karakterin
JMP ENDLESS ; bir fazlasını gönder
; sonsuz döngü
START
CODE
ENDP
ENDS
END START
```

20. 8254 PIT (PROGRAMMABLE INTERVAL TIMER)

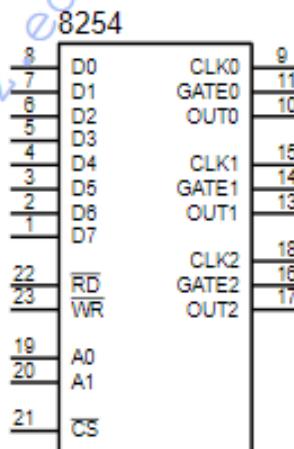
8254 programlanabilir aralık sayacı (Programmable Interval Timer - PIT) olarak isimlendirilen, üç adet programlanabilir 16 bit'lik, geriye doğru sayıci barındıran, sayıcları altı farklı sayma/zamanlama kipi destekleyen, 8 bit veri genişliğine sahip, TTL uyumlu bir programlanabilir arayüz tüm devresidir.

Genel olarak kullanıldığı durumlar aşağıda listelenmiştir:

- yazılım kontrollü, kesin ve doğru zaman gecikmeleri oluşturmak
- olay sayıci
- gerçek zamanlı saat (Real Time Clock - RTC)
- kare dalga üretici
- karmaşık dalga üretici
- tetikleme sinyali üretimi

20.1 8254 Uç Tanımları

8254 sekiz adet veri (D7-D0), iki adet adres (A1, A0), okuma kontrol (\bar{RD}), yazma kontrol (\bar{WR}), yetkilendirme (\bar{CS}) uçlarıyla, her bir sayıci için saat darbesi girişi (CLK0 CLK1, CLK2), kontrol (GATE0, GATE1, GATE2) ve çıkış (OUT0, OUT1, OUT2) uçlarını barındırır (Şekil 20-1). Adres uç değerlerine göre 8254'te erişilen yazmaçlar Tablo 20-1 ile verilmiştir.



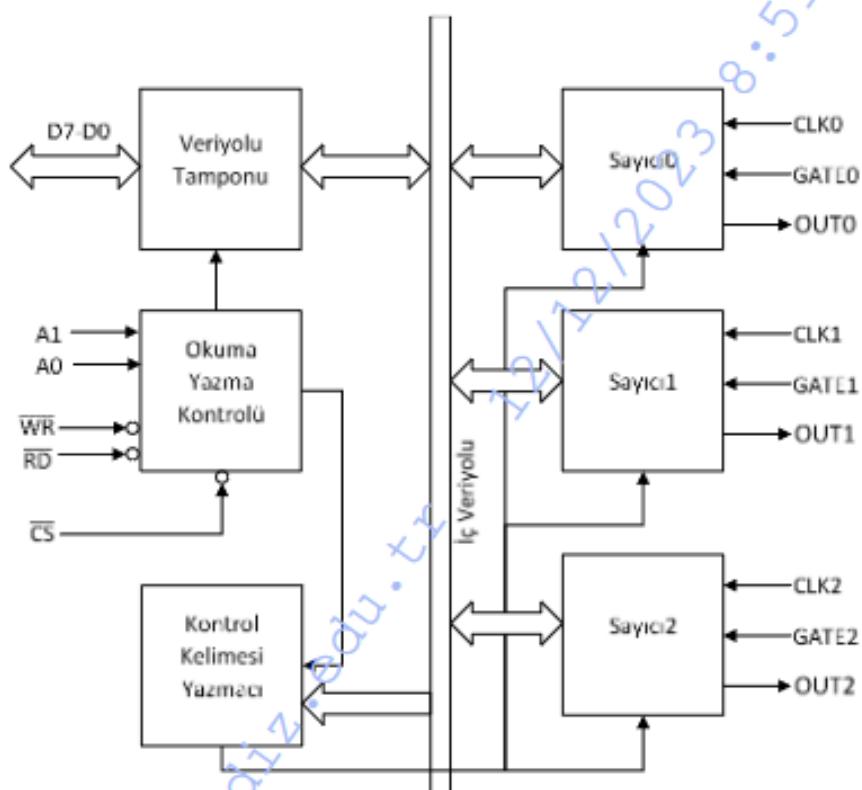
Şekil 20-1: 8254 fonksiyonel uç tanımları

Tablo 20-1: 8254 adresleme ve yazmaç erişimi

A1	A0	Yazmaç
0	0	Sayıci 0
0	1	Sayıci 1
1	0	Sayıci 2
1	1	Kontrol Yazmaç

20.2 8254 İç Yapısı

8254 iç yapısı incelendiğinde 3 adet sayıcıya ilişkin kontrol işaretleri ve değer yüklemelerine ilişkin veri akışını sağlayan yollar görülebilmektedir. Ayrıca veri yolu tamponu ve okuma yazma kontrolü mikroişlemci tarafı ile haberleşme kontrol edilmektedir. 8254 iç yapısı Şekil 20-2 ile verilmektedir.



Şekil 20-2: 8254 iç yapısı

20.3 8254 Kontrol Yazmacı

8254 kontrol yazmacı formatı Tablo 20-2 ile verilmiştir. Kontrol kelimesine yazılan değerler ile temelde sayıcıların kipleri ve sayma formatları ayarlanabilir. Bunula birlikte özel iki komut (geri okuma komutu ve sayıcı tutucu komutu) da kontrol yazmacı ile verilebilmektedir. Sayıcı kipi ve sayma formatı ayarlayabilmek için en yüksek anlamlı iki bit (SC1-SC0) ile sayıcı seçilir, sayma değerine ilişkin format belirlenir (RW1-RW0), kip seçilir (M2-M0) ve sayma formatı (binary, BCD) belirlenir.

Eğer sayma değeri için 0-255 arasında bir değer yeterli ise 16 bit'lik sayıcı için sadece LSB (düşük anlamlı byte) yazmak yeterlidir. Eğer sayıcı için 256-65280 arasında 256'nın katı bir değer yüklenecekse sadece MSB (yüksek anlamlı byte) yazmak yeterlidir. Eğer sayıcı için 0-65535 arasında bir değer gönderilecekse önce LSB sonra MSB değerleri ile 16 bit'lik sayma değer oluşturulabilir. 16 bit'lik sayma de-

ğerinin iki adımda gönderilebilmesinin sebebi ise 8254'ün veri yolunun 8 bit'lik olmasıdır. Kontrol yazmacı için kip ve format yazıldıktan sonra sayma değeri sayıcı adresine yapılan yazma ile gönderilir.

Sayıci ikili formatta saymaya ayarlanmışsa ve ilk değer olarak 0010H yüklenmişse sayıci sırasıyla 0010H, 000FH, 000EH, ..., 0000H değerlerini alır, bunula birlikte sayıci BCD formatta ayarlanmışsa ve ilk değer olarak 0010H yüklenmişse sayıci sırasıyla 0010H, 0009H, 0008H, ..., 0000H değerlerini alır.

Tablo 20-2: 8254 kontrol yazmacı formати

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
SC ₁	SC ₀	RW ₁	RW ₀	M ₂	M ₁	M ₀	BCD

SC ₁	SC ₀	SC - Sayıcı Seçim
0	0	Sayıci 0
0	1	Sayıci 1
1	0	Sayıci 2
1	1	Geri Okuma Komutu

M ₂	M ₁	M ₀	M - Kip
0	0	0	Kip 0
0	0	1	Kip 1
X	1	0	Kip 2
X	1	1	Kip 3
1	0	0	Kip 4
1	0	1	Kip 5

RW ₁	RW ₀	RW - Okuma/Yazma
0	0	Sayıci Tutucu Komutu
0	1	LSB
1	0	MSB
1	1	Önce LSB, sonra MSB

BCD	Sayma
0	İkili
1	İkili kodlu ondalık (BCD)

20.4 8254 Kipleri

8254 altı farklı kipte kullanılabilir. Bu kipler tek seferlik, tekrarlı veya dışardan tespitlenebilen yapıdadırlar.

20.4.1 Kip 0 (Interrupt on Terminal Count)

Kontrol değerinin kontrol yazmacına yazılıması ile OUT ucu lojik 0 olur. OUT ucu sayma değeri kadar saat darbesi süresince lojik 0'da kalır, sayma bitince lojik 1'e çıkar. İlgili sayıciya yeni sayma değeri yüklenirse yeni sayma değeri ile işlem tekrarlanır. GATE ucu lojik 0 yapılrsa sayma duraklar.

20.4.2 Kip 1 (Hardware Retriggerable One Shot)

Kontrol değeri ve sayma değeri yüklenikten sonra OUT ucu lojik 1 yapılır. GATE ucunda lojik 0 → 1 geçisi görüldüğünde OUT ucu lojik 0 yapılır. Sayma boyunca OUT lojik 0'da kalır. Sayma işleminin bitmesiyle OUT ucu lojik 1'e çekilir. GATE ucunda tekrar lojik 0 → 1 geçisi görülmemesiyle işlem tekrarlanır.

20.4.3 Kip 2 (Rate Generator)

Tekrarlı darbe (pulse) üreten bir kiptir. OUT ucu başta lojik 1'dedir. Sayma değeri 1 oluncaya kadar OUT ucu lojik 1'de kalır. Sayma değeri 1 olduğunda OUT ucu lojik 0'a düşer, 1 saat darbesi sonra OUT ucu tekrar lojik 1'e çıkar. İşlem periyodik olarak tekrarlanır.

Örnek 20-1

CLK0 ucunda 2 MHz frekansında saat darbesi bağlı olan, CNTR0 kullanarak 50 μ s periyotlu bir pulse işaretini üretiniz.

Sayma değeri hesaplayabilmek için CLK0 ucundaki saat darbesi frekansını üretilecek işaretin frekansına bölmek yeterlidir.

$$\frac{2 \text{ MHz}}{\frac{1}{50\mu\text{s}}} = \frac{2 \times 10^6}{\left(\frac{1}{50 \times 10^{-6}}\right)} = 100 = 64H$$

Bu durumda sekiz bit'lik bir sayma değerinin yeterli olduğu görülür. Buna göre kontrol kelimesi Tablo 20-3 ile verildiği gibi belirlenir.

Tablo 20-3: Örnek 20-1 için kontrol yazmacı

Alan	Sayıcı Seçimi		Okuma-Yazma		Kip			Sayma	Onaltılık Değer
Değer	0	0	0	1	0	1	0	0	14H
Anlam	Sayıcı 0 seçilir		Sadece LSB		Kip 2			İkili sayma	

```

MOV AL, 14H
OUT CONTROL_ADDRESS, AL
MOV AL, 64H
OUT CNTR0_ADDRESS, AL

```

20.4.4 Kip 3 (Square Wave Generator)

Sayma değerinin yarısı süresinde OUT ucu lojik 1, diğer yarısında ise lojik 0 olur. Periyodik olarak tekrarlanır.

Örnek 20-2

CLK1 ucunda 2 MHz frekansında saat darbesi bağlı olan, CNTR1 kullanarak 1 kHz frekansında kare dalga üretiniz.

Sayma değeri hesaplayabilmek için CLK1 ucundaki saat darbesi frekansını üretilecek işaretin frekansına bölmek yeterlidir.

$$\frac{2 \text{ MHz}}{1 \text{ kHz}} = \frac{2 \times 10^6}{1 \times 10^3} = 2000$$

Bu durumda sayma değeri BCD formatında yüklenmek istenirse kontrol kelimesi Tablo 20-4 ile verildiği gibi belirlenir.

Tablo 20-4: Örnek 20-2 için kontrol yazmacı

Alan	Sayıcı Seçimi		Okuma-Yazma		Kip			Sayma	Onaltılık Değер
Değer	0	0	0	1	0	1	1	1	77H
Anlam	Sayıcı 1 seçilir		Once LSB, sonra MSB		Kip 3			BCD sayma	

```

MOV AL, 77H
OUT CONTROL_ADDRESS, AL
MOV AL, 00H
OUT CNTR1_ADDRESS, AL
MOV AL, 20H
OUT CNTR1_ADDRESS, AL

```

20.4.5 Kip 4 (Software Triggered Strobe)

OUT ucu başta lojik 1'dedir. Sayma değeri 1 olunca OUT ucu lojik 0 yapılır. 1 saat darbesi süre sonra OUT ucu lojik 1 yapılır. Yeni bir sayma değeri yüklenindiğinde işlem bu sayma değeri için tekrarlanır.

20.4.6 Kip 5 (Hardware Triggered Strobe)

OUT ucu başta lojik 1'dedir. GATE ucunda lojik 0 → 1 geçisi görülmesiyle sayma başlar. Sayma değeri 1 olunca OUT ucu lojik 0 yapılır. 1 saat darbesi süre sonra OUT ucu lojik 1 yapılır. GATE ucunda yeni lojik 0 → 1 geçisi görülmesiyle işlem tekrarlanır.

20.5 8254 Sayıcı Tutma Komutu

8254 başlatıldıktan sonra herhangi bir sayıcının sayma değerini almak için tutma komutu kullanılabilir (Tablo 20-5). Tutma komutu yürütüldükten sonra sayıcı adresinden yapılan okuma ile tutulan sayma değerine erişilebilir. Tutma fonksiyonu ile bir seferde sadece bir sayıcıya ilişkin sayma değeri tutulabilir.

Tablo 20-5: 8254 sayıçı tutma komut formatı

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
SC ₁	SC ₀	0	0	X	X	X	X

SC ₁	SC ₀	SC - Sayıcı Seçim
0	0	Sayıci0
0	1	Sayıci1
1	0	Sayıci2

20.6 8254 Geri Okuma Komutu

8254 başlatıldıktan sonra herhangi bir sayıcının sayma değerini ve/veya sayıcının durumunu almak için geri okuma komutu kullanılabilir (Tablo 20-6). Geri okuma komutuyla herhangi bir sayıcıya ilişkin durum tutabilmek için, COUNT kontrol bit'ine lojik 0 yazılmalı ve CNT₀, CNT₁, CNT₂ sayıcılarından hangileri tutulmak isteniyorsa lojik 1 olarak işaretlenmelidirler. Geri okuma komutuyla herhangi bir sayıcıya ilişkin sayma değerini tutabilmek için, STATUS kontrol bit'ine lojik 0 yazılmalı ve CNT₀, CNT₁, CNT₂ sayıcılarından hangileri tutulmak isteniyorsa lojik 1 olarak işaretlenmelidirler. Geri okuma komutuyla istenirse tüm sayıcılar için hem durum hem de sayma değeri tutulabilir. Durum ve sayma değerlerine sayıcının adresinden yapılan okumalar ile erişilebilir. Durum ve sayma değeri okuma ile şu noktaya dikkat edilmelidir: Sayıcıya ilişkin herhangi bir okuma yapılmadan sırasına bakılmaksızın durum ve sayma değeri tutulmuşsa, sayıcı adresinden yapılan ilk okuma durumu, sonraki bir veya iki okuma (sayıcının ayarlandığı moda göre 8 bit veya 16 bit sayma değerine göre) ise sayma değerini verir. Bir başka dikkat edilmesi gereken nokta ise daha önceden yapılan tutma işlem sonuçları okunmadan, yeni geri okuma komutuyla yapılan tutma işlemleri geçersizdir.

Tablo 20-6: 8254 geri okuma komut formatı

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	1	COUNT	STATUS	CNT ₂	CNT ₁	CNT ₀	0

COUNT = 0 : Sayma değeri tut
 STATUS = 0 : Durum tut
 CNT2 = 1 : Sayıcı 2 için tut
 CNT1 = 1 : Sayıcı 1 için tut
 CNT0 = 1 : Sayıcı0 için tut

20.7 8254 Durum Yazmacı

8254 için durum yazmacı sayıcının hangi kipte ayarlanmış olduğunu geri okumak için kullanılabilir. Bunun dışında sayıcının ayarlama aşamasında olup aktif olmadığını gösteren NULL_COUNT durum bit'ini ve sayıcıya ilişkin OUTPUT ucunun durumunu da sunmaktadır (Tablo 20-7).

Tablo 20-7: 8254 durum yazmacı

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
OUTPUT	Null_Count	RW ₁	RW ₀	M ₂	M ₁	M ₀	BCD

20.8 8254 Örnek Uygulamaları

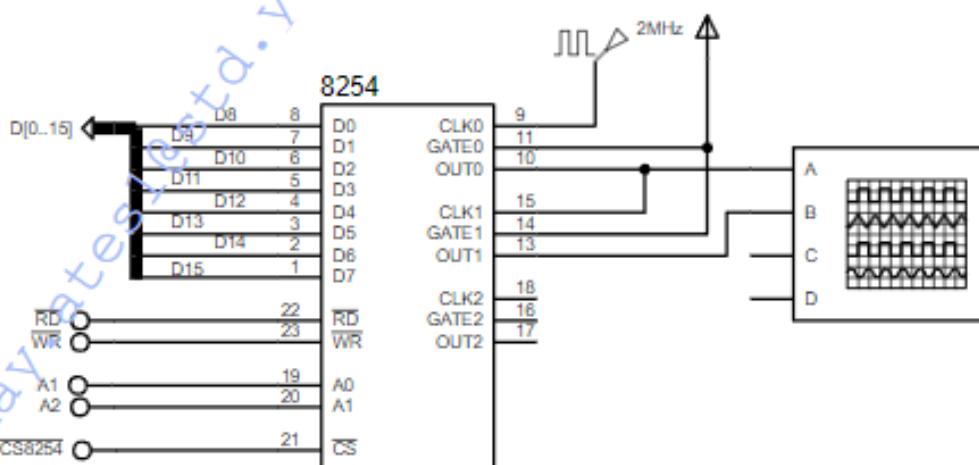
8254'ün zamanlayıcı ve sayıci olarak fonksiyon gösterdiği örneklerde aşağıda yer verilmiştir. Zamanlayıcı olarak darbe üretme (Örnek 20-3), nota üretme (Örnek 20-4) ve Pulse Width Modulation (PWM) (Örnek 20-5) işaretü üretme örneklerine yer verilirken, Örnek 20-6 ile 8254'ün durum sayıcı olarak çalıştığı bir örneğe yer verilmiştir.

20.8.1 Darbe (Pulse) Üretme

Örnek 20-3

8086 G/Ç uzayında 0A9H adresinden itibaren ardışık tek adreslere yerleştirilen 8254 ve 2 MHz frekansında bir kara dalga yardımıyla periyodu 1 sn olan pulse işaretinin üretilmesi istenmektedir. Buna göre:

- 8254'e yapılması gereken bağlantıları gösteriniz.
- 1 sn periyotlu pulse işaretini üretebilmek için gerekli assembly kodunu yazınız.



Şekil 20-3: Örnek 20-3 için 8254 bağlantıları

Şekil 18-21 ile verilen adres çözümleme devresi ve Şekil 20-3 ile verilen bağlantılar sonucunda 8086 G/Ç uzayında 0A9H adresine Sayıcı0, 0ABH adresinde Sayıcı1, 0ADH adresinde Sayıcı2 ve 0AFH adresinde ise kontrol yazmacı bulunmaktadır. Mevcut saat frekansı (2 MHz) ve tek bir sayıcı ile elde edilebilecek en düşük frekans $\frac{2 \cdot 10^6}{65536} \cong 30\text{Hz}$ ’tir. Bu durumda ilkinin CLK ucunda mevcut saat işaretini, ikincisinin CLK ucunda ise birincinin OUT çıkışını bağlı olmak üzere iki sayıcı kullanılabılır. Kullanılacak her iki sayıcı da Kip 2’de kullanılabilir. Sayıcı0 2.000, Sayıcı1 ise 1.000 sayma değeri ile yüklenirse ve OUT0 ucu CLK1 ucuna bağlanırsa elde edilecek frekans $\frac{2 \cdot 10^6}{2000 \cdot 1000} = 1\text{ Hz}$ olur.

```

; PROGRAM : Örnek 20-3.ASM
; 8254'te iki sayıcı kullanarak 2 MHz saat işaretinden 1sn periyotlu
; pulse üreten assembly program
;
PAGE 60,80
TITLE 8254 iki sayıcı birlikte kullanım ile pulse üretme
STAK SEGMENT PARA STACK 'STACK'
DW 20 DUP(?)
STAK ENDS
CODE SEGMENT PARA 'CODE'
ASSUME CS:CODE, SS:STAK
START PROC FAR           ; ana yordam
MOV AL, 00110101B
OUT 0AFH, AL             ; CNTR0, 16 bit, kip 2, BCD
MOV AL, 01110101B
OUT 0AFH, AL             ; CNTR1, 16 bit, kip 2, BCD
XOR AL, AL
OUT 0A9H, AL
OUT 0ABH, AL
MOV AL, 20H               ; CNTR0 sayma değeri 2000
OUT 0A9H, AL
MOV AL, 10H               ; CNTR1 sayma değeri 1000
OUT 0ABH, AL
ENDLESS: JMP ENDLESS     ; sonsuz döngü
START ENDP
CODE ENDS
END START

```

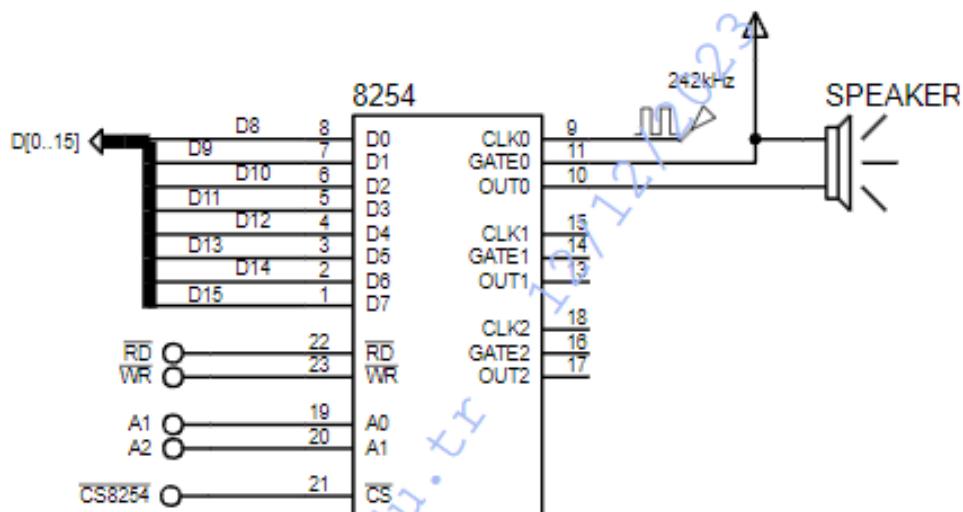


Şekil 20-4: Örnek 20-3 için osiloskop çıktısı

20.8.2 Nota Üretme

Örnek 20-4

8086 G/C uzayında A9H adresinden itibaren ardışık tek adreslere 8254 yerleştirilmiştir. 8254 için Şekil 20-5 ile verilen bağlantılar yapılmıştır. CLK0'ya 242 kHz frekansında kare dalga bağlanmıştır. Buna göre 8254 yardımıyla dördüncü oktav La (A_4) notası frekansında kare dalga üretmek için gerekli assembly kodunu yazınız.



Şekil 20-5: Örnek 20-4 için 8254 bağlantıları

Tablo 20-8: Dördüncü oktav notaların frekans karşılıkları

Nota	Frekans (Hz)
C ₄ (Do)	261,63
D ₄ (Re)	293,66
E ₄ (Mi)	329,63
F ₄ (Fa)	349,23
G ₄ (Sol)	392,00
A ₄ (La)	440,00
B ₄ (Si)	493,88

Verilen CLK0 frekansı (242 kHz) ve A₄ frekansı (440 Hz) dikkate alındığında CNTR0 sayma değerinin $242000/440 = 550$ olduğu bulunur. Tek ton 440 Hz sinyal üretmek için sinyal formunun sinüs dalgası olması gerekmektedir. Fakat burada 440 Hz frekansında bir kare dalga elde edildiği durum yeterli sayılacaktır. Pratikte kare dalga çıktısı filtreden geçirilerek iyileştirilebilir.

```

PROGRAM : Örnek 20-4.ASM
; 8254 kullanarak 440 Hz nota üreten
; assembly program

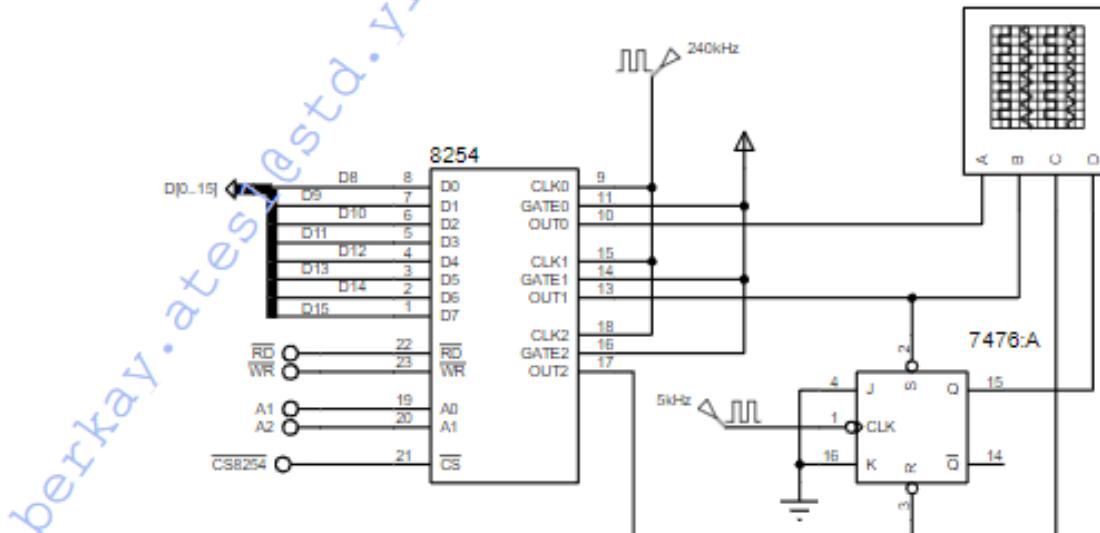
PAGE 60, 80
TITLE 8254 440 Hz nota üretme
STAK SEGMENT PARA STACK 'STACK'
DW 20 DUP(?)
ENDS
CODE SEGMENT PARA 'CODE'
ASSUME CS:CODE, SS:STAK
START PROC FAR ; ana yordam
MOV AL, 00110110B
OUT 0AFH, AL ; CNTR0 16 bit, kip 3, binary
MOV AX, 550
OUT 0A9H, AL
MOV AL, AH
OUT 0A9H, AL ; CNTR0 sayma değeri 550
ENDLESS: JMP ENDLESS ; sonsuz döngü
START ENDP
CODE ENDS
END START

```

20.8.3 PWM Sinyali Üretme

Örnek 20-5

Örnek 20-4 ile verilen adres aralıklarına yerleştirilen 8254 için Şekil 20-6 ile verilen bağlantılar yapılmıştır. JK iki duraklı (flip-flop) kullanarak yüksek hassasiyetli bir PWM işaretini üretmek istenmektedir. Bunun için CLK1 ve CLK2 uçlarına 240 kHz frekansında saat darbesi bağlanmıştır, OUT1 ve OUT2 çıkışları JK iki duraklısının S (set) ve R̄ (reset) uçlarına bağlanmıştır. OUT1 ve OUT2 arasındaki faz farkından yararlanarak 1 kHz frekansında %75 görev oranında PWM işaretini üretmektırm için gerekli assembly kodunu yazınız.



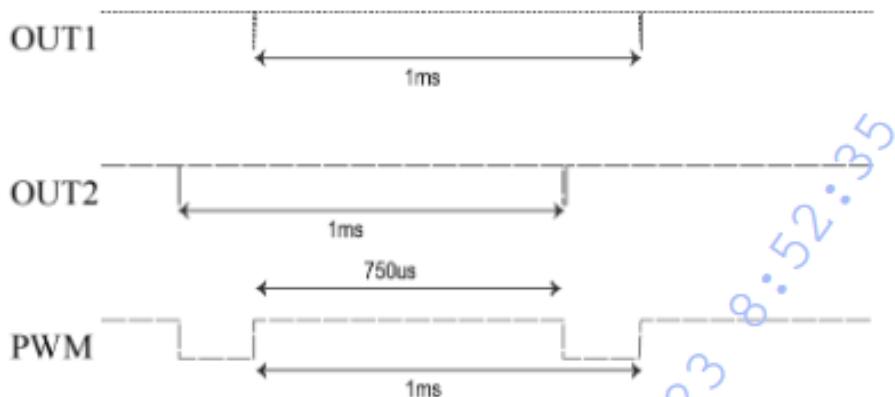
Şekil 20-6: Örnek 20-5 için 8254 bağlantıları

JK iki duraklı çıkışında bir PWM işaretini üretmek isteniyor. JK uçlarına (00)₂ değeri bağlı olduğu için JK iki duraklı durumunu koruyacak şekilde kendi saat frekansında (5 kHz) çalışacaktır. Öyleyse PWM işaretinin üretimi JK iki duraklısının \bar{R} ve \bar{S} anuyumsuz kontrol uçları ile sağlanmalıdır. Aktif sıfır olmaları da dikkate alınarak ($\bar{R} S$)₂ = (01)₂ uygulandığında JK çıkış lojik 0, ($\bar{R} S$)₂ = (10)₂ uygulandığında ise JK çıkış lojik 1 olarak ayarlanacaktır. PWM işaretinin %75 görev oranında 1 kHz'lik bir işaret olması istendiğinden önce \bar{S} ucunun lojik 0 yapılması bundan tam 750 μ s sonra da \bar{R} ucunun lojik 0 yapılması gerekmektedir (240 kHz'lik CLK girişi sayma değeri olarak 240'a bölündürse 1 kHz frekansı elde edilir, 1 kHz'lik sinyalin periyodu 1 ms olup bunun %75'i 750 μ s'dır). \bar{S} ucu OUT1, \bar{R} ucu ise OUT2 ile sürülmektedir. Dolayısıyla CNTR1 ve CNTR2 240 sayma değeri ile Kip 2'de ayarlanmalıdır. CNTR1 ile CNTR2'nin başlatılması (sayma değerlerinin yüklenmesi) arasında 750 μ s fark CNTR0 ile hassas bir şekilde ölçülecektir. CLK0'a 240 kHz bağlıdır, CNTR0'ın Kip 0'da 750 μ s sayılabilmesi için $750 \times 10^{-6} \times 240000 = 180$ sayma değeri ile yüklenmesi gereklidir.

```

; PROGRAM : Örnek 20-5.ASM
; 8254 CNTR0, CNTR1 ve CNTR2 kullanarak 1kHz frekansında %75 görev
; oranında PWM işaretini üreten assembly program
;
PAGE 60,80
TITLE 8254 PWM işaret 1kHz %75 görev oranı
STAK SEGMENT PARA STACK 'STACK'
DW 20 DUP(?)
STAK ENDS
CODE SEGMENT PARA 'CODE'
ASSUME CS:CODE, SS:STAK
START PROC FAR ; ana yordam
MOV AL, 01010100B
OUT OAFH, AL ; CNTR1, 8bit, kip 2, binary
MOV AL, 240
OUT OABH, AL ; CNTR1 sayma değeri 240, 1kHz frekans
MOV AL, 00010000B
OUT OAFH, AL ; CNTR0, 8 bit, kip 0, binary
;
; CNTR0 sayma değeri 180
; CNTR1 ve CNTR2 arasında 750us faz farkı
;
MOV AL, 180
OUT OA9H, AL
MOV AL, 11100010B
OUT OAFH, AL ; CNTR0 readback status
KNTRL: IN AL, 0A9H ; CNTR0 durum yazmacını oku
TEST AL, 80H
JZ KNTRL ; OUT0 1 olunca CNTR2'yi başlat
MOV AL, 10010100B
OUT OAFH, AL ; CNTR2, 8bit, kip 2, binary
MOV AL, 240
OUT OADH, AL ; CNTR2 sayma değeri 240, 1kHz frekans
ENDLESS: JMP ENDLESS ; sonsuz döngü
START ENDP
CODE ENDS
END START

```

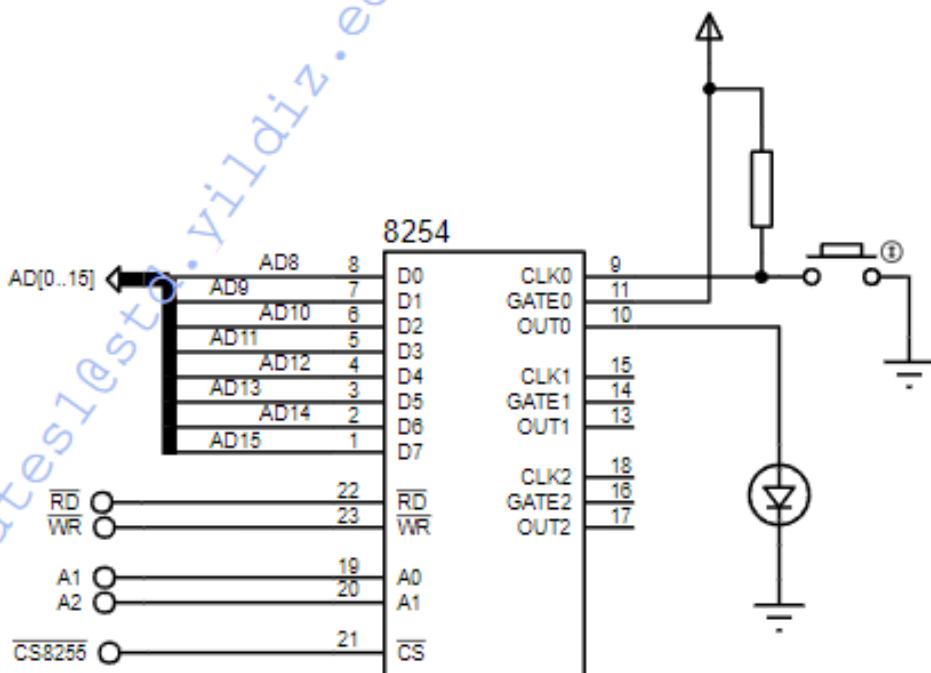


Şekil 20-7: Örnek 20-5 için osiloskop çıktısı

20.8.4 Sayıcı

Örnek 20-6

Bir paketleme bandında 100'er ürünü oluşturmak istenmektedir. Ürünlerin tek sıralı olarak taşınmış olduğu banutta, ürün, sayma sensöründen (bir buton üzerinden geçen ürün butona basılması sağlanmaktadır) gereklilik paket kolisine ulaşır. Mikroişlemci G/C uzayında 0A9H adresinden itibaren ardışık tek adreslere yerleştirilmiş 8254 ve sayma sensörünü kullanarak her yeni 100'üncü ürünü uyarı üreten sistemi tasarlayınız.



Şekil 20-8: Örnek 20-6 için 8254 bağlantıları

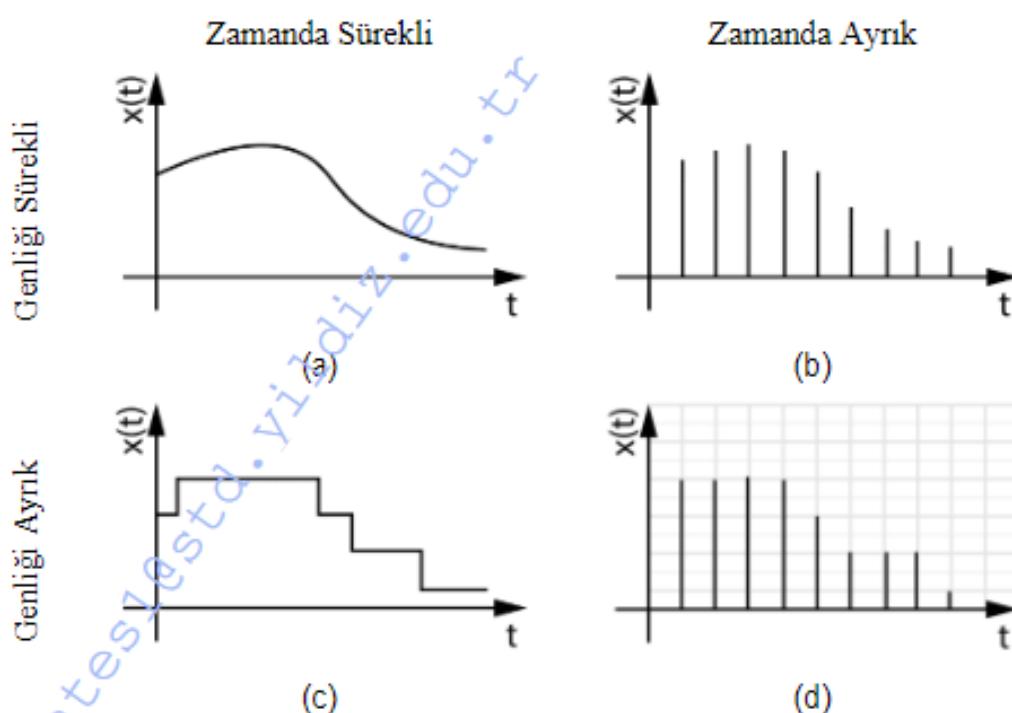
8254'ü durum sayıcı olarak kullanılabilmesi için durum üreten kaynağın, sayıcı CLK ucuna bağlanması gerekmektedir. Örnekte her bir ürün ağırlığı ile butona basılmasını sağladığı için buton çıkışının doğrudan CLK0 ucuna bağlanmıştır (Şekil 20-8). Bu durumda CNTR0 Kip 2'de 100 sayımada bir darbe üretecek şekilde ayarlanırsa her 100. yılında OUT0'a bağlı LED sönerken uyarı üretecektir. Kip 2 tekrarlı bir kip olduğu için yeni bir kontrol değeri veya sayma değeri yüklemeye gerek kalmadan sistem çalışmasına devam edebilecektir.

```
; -----  
; PROGRAM :Örnek 20-6.ASM  
; 8254 sayıcı uygulaması ile tekrarlı olarak  
; 100 sayınca uyarı veren assembly program  
;  
PAGE 60,80  
TITLE 8254 kip 2 sayıcı  
STAK SEGMENT PARA STACK 'STACK'  
DW 20 DUP(?)  
STAK ENDS  
CODE SEGMENT PARA 'CODE'  
ASSUME CS:CODE, SS:STAK  
START PROC FAR ; ana yordam  
MOV AL, 00010100B  
OUT 0AFH, AL ; CNTR0, 8bit, kip 2, binary  
MOV AL, 99  
OUT 0A9H, AL ; CNTR0 sayma değeri 99  
ENDLESS: JMP ENDLESS ; sonsuz döngü  
START ENDP  
CODE ENDS  
END START
```

21. ADC0804 VE DAC0830

Fiziki dünya analog²¹² bir ortamdır. Fiziki dünyadaki sıcaklık, basınç, gerilim, akım, hız, ışık şiddeti gibi parametreler analog değerlere sahiptir. Mikroişlemciler ise sayısal (digital) hatta ikili değerlerle çalışır. Mikroişlemci ile fiziki dünya arasında etkileşim sağlayabilmek için analog değerleri sayısal değerlere ve sayısal değerleri analog değerlere çeviren dönüştürücülere ihtiyaç vardır.

Fonksiyonlar tanım kümelerinin sürekli veya ayrik olmalarına göre zamanda sürekli veya zamanda ayrik işaretler olarak ikiye ayrılmaktadır. Benzer şekilde fonksiyonların görüntü kümelerinde sürekli veya ayrik değerler almaları ile genliği sürekli veya genliği ayrik işaretler olarak ikiye ayrılmaktadır. Şekil 21-1 ile tanım ve görüntü kümelerinde sürekli ve ayrik olan işaret çeşitleri verilmiştir. Zamanda (tanım kümelerinde) ve genlikte (görüntü kümelerinde) sürekli olan işaretler **analog işaret** olarak adlandırılır. Zamanda (tanım kümelerinde) ve genlikte (görüntü kümelerinde) ayrik olan işaretler ise **sayısal işaret** olarak adlandırılır.



Sekil 21-1: Tanım ve görüntü kümelerinde sürekli ve ayrik işaret çeşitleri

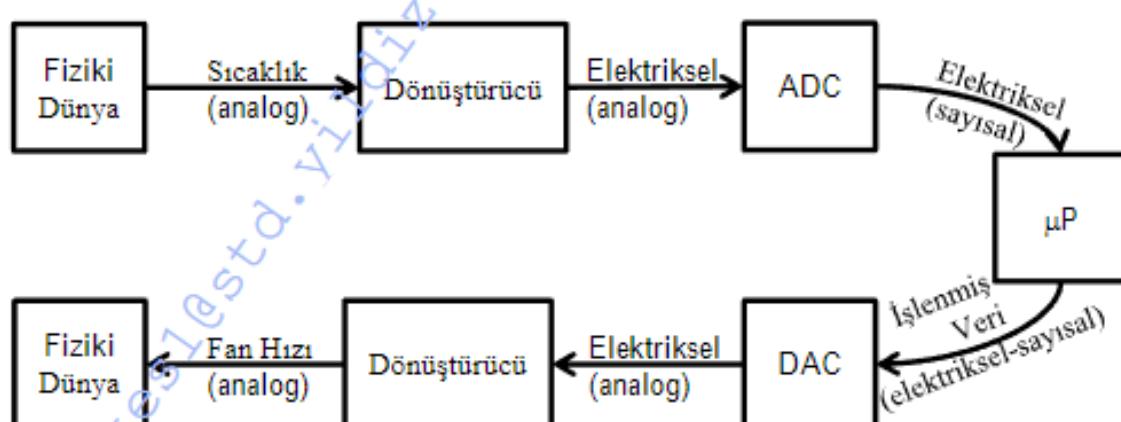
Analog değerlerin sayısal değerlere çevrilmesi analog-sayısal çeviriciler (analog-digital converter - ADC) ile sağlanmaktadır. Sayısal değerlerin analog değerlere çev-

²¹² Analog varlıklar ölçülebilirken. (cm, kg, lt vb. birimler kullanılarak), sayısal varlıklar sayılabilir

rilmesi ise sayısal-analog çeviriciler (digital-analog converter - DAC) ile sağlanmaktadır. Mikroişlemcili sistemlerde ADC modülünün girdi olarak elektriksel bir işaret olan analog gerilim değeri aldığı ve dönüştürme çözünürlüğünə göre bir bit dizisi ürettiği söylenebilir. DAC'ının ise, çözünürlüğe göre bir bit dizisini girdi olarak alarak, çıktıda elektriksel analog bir değer ürettiği söylenebilir.

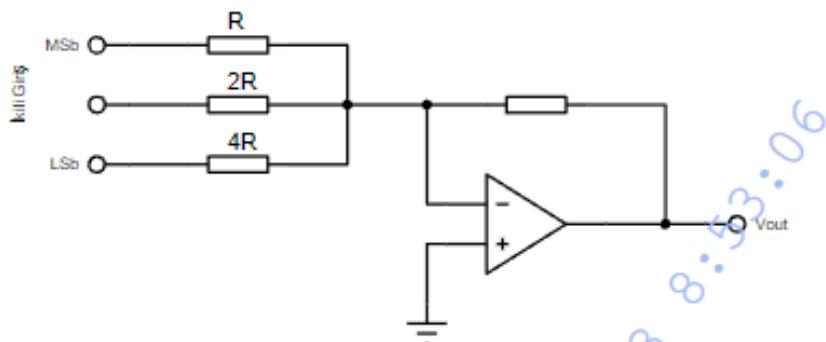
Fiziki dünya ile mikroişlemcili sistemin etkileşim sağlanabilmesi, mikroişlemcili sistem ile fiziki dünyadaki bazı parametrelerin okunabilmesi için çeşitli enerji tiplerini (ısı, basınç vb.) elektrik enerjisine çeviren dönüştürücü (transducer) modüllerine de ihtiyaç olduğu görülmektedir. Yine mikroişlemcili sistemin, ürettiği elektriksel analog çıktılar vasıtıyla, fiziki dünyada kontrol yürütebilmesi için elektrik enerjisini diğer enerji tiplerine dönüştüren ikinci bir dönüştürücüye daha ihtiyaç vardır.

Şekil 21-2 ile ortam sıcaklığını belirli bir aralıkta tutmayı hedefleyen mikroişlemcili bir sisteme ilişkin blok diyagramı verilmiştir. Ortam sıcaklığının ölçülerek elektriksel analog bir değere dönüştürülmesi; örneğin direnç değeri sıcaklıkla değişen; termistörden oluşturulan dönüştürücü ile sağlanabilir. Elektriksel analog işaretin belirli aralıklarla örneklenerek sayısallaştırılması ADC ile sağlanır. İkili bir sayıya dönüştürülen ortam sıcaklık değerine göre mikroişlemcide gerekli işlem yürütülerek ortamda bulunan fanın dönüş hızı belirlenir. İkili bir sayı olarak oluşturulan hız değerine karşılık DAC kullanılarak analog elektriksel gerilim değeri oluşturulur. Motor sürücüden oluşan bir dönüştürücü ile analog gerilim değerine karşılık fanın dönüsü sağlanabilir.



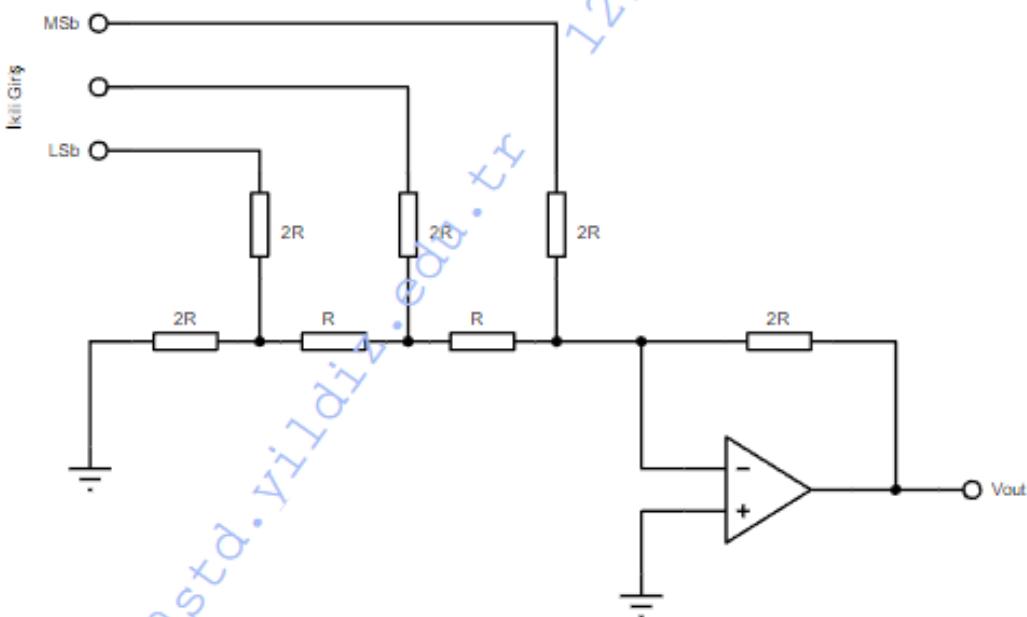
Şekil 21-2: Mikroişlemcili bir sistem ile fiziki dünya etkileşimi

Bu bölümde iki farklı tip DAC yapısı incelenmiştir. İkili-ağırlıklı sayısal-analog çeviriçi (binary weighted DAC) yapısı temelde toplayıcı op-amp devresine benzer. Bu yapıda hassas toleranslarda farklı değerlere sahip direnç değerlerinin kullanımını gerektmektedir. Şekil 21-3 ile ikili-ağırlıklı DAC yapısı görülebilmektedir.



Şekil 21-3: İkili-ağırlıklı sayısal-analog çevirici

R-2R merdiven sayısal-analog çevirici (R-2R ladderAC), ikili-ağırlıklı DAC'ye benzer şekilde birlikte sadece iki farklı direnç değerini kullanan devre şemasına sahiptir. Şekil 21-4 ile R-2R merdiven DAC yapısı verilmiştir.

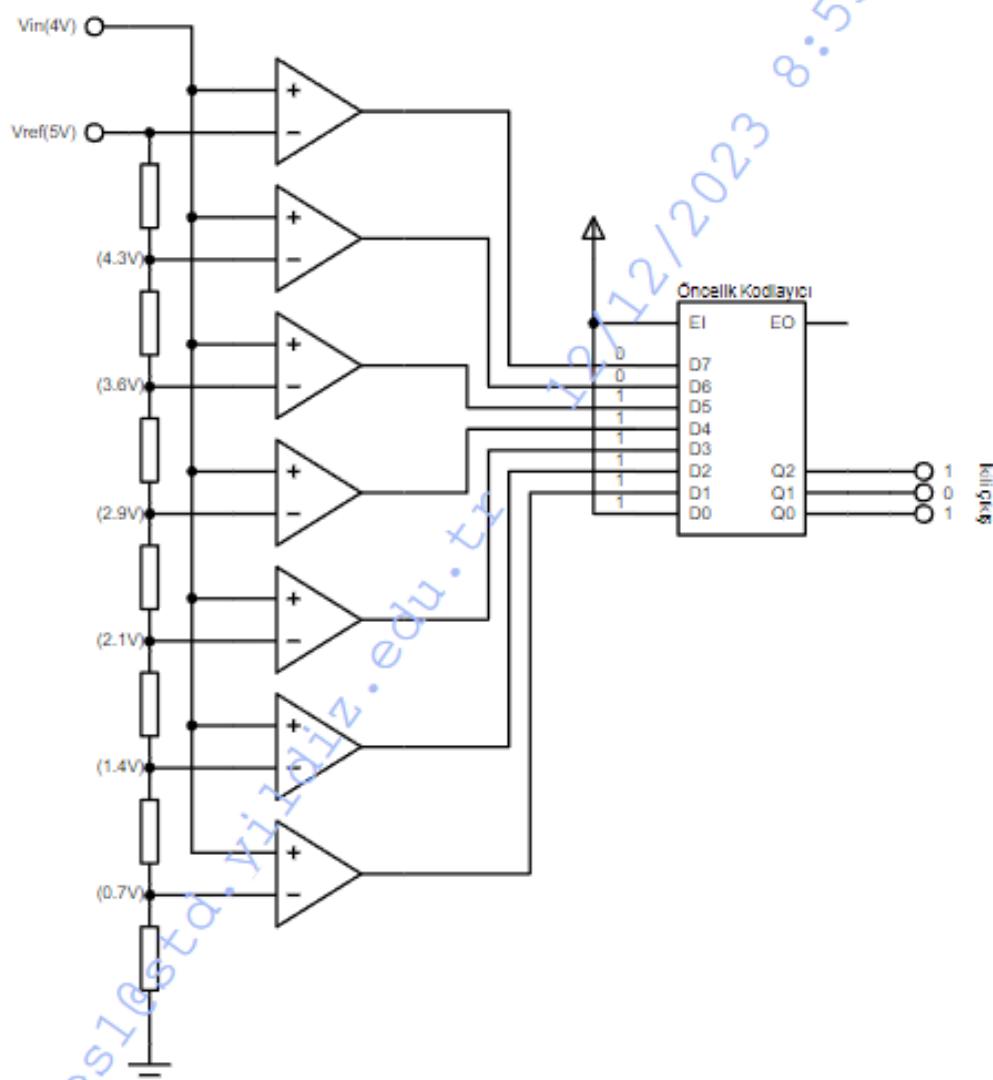


Şekil 21-4: R-2R merdiven sayısal-analog çevirici

Bu bölümde iki farklı tip ADC yapısı incelenmiştir. Paralel ADC yapısında bir referans gerilim değeri (örneğin 5 V) gerilim bölücü direnç merdivenine uygulanır. Gerilim bölücü direnç merdiveninin her basamak çıkışını dönüştürmek istenen analog giriş gerilimi ile karşılaştırılır²¹³. Karşılaştırıcı çıkışları belirli bir basamağa kadar lojik 0 çıkışı üretirken analog giriş gerilim seviyesine göre belirli bir basamaktan sonra

²¹³ Karşılaştırıcılarda, $V_+ > V_-$ ise çıkış lojik 1, $V_+ \leq V_-$ ise çıkış lojik 0 kabul edilir.

lojik 1 üretecektir. Analog giriş geriliminin, referans gerilimin direnç merdivenindeki çıktılarını hangi basamakta geçtiği ise öncelik kodlayıcısı (priority encoder)²¹⁴ ile bulunarak ikili sayıya dönüştürülmektedir. Şekil 21-5 ile paralel ADC yapısı verilmiştir. Örneğin referans geriliminin 5 V, dönüştürülmek istenen analog değerinin de 4 V olduğu durumda DAC sonucu olarak (101)₂ değeri üretilir.

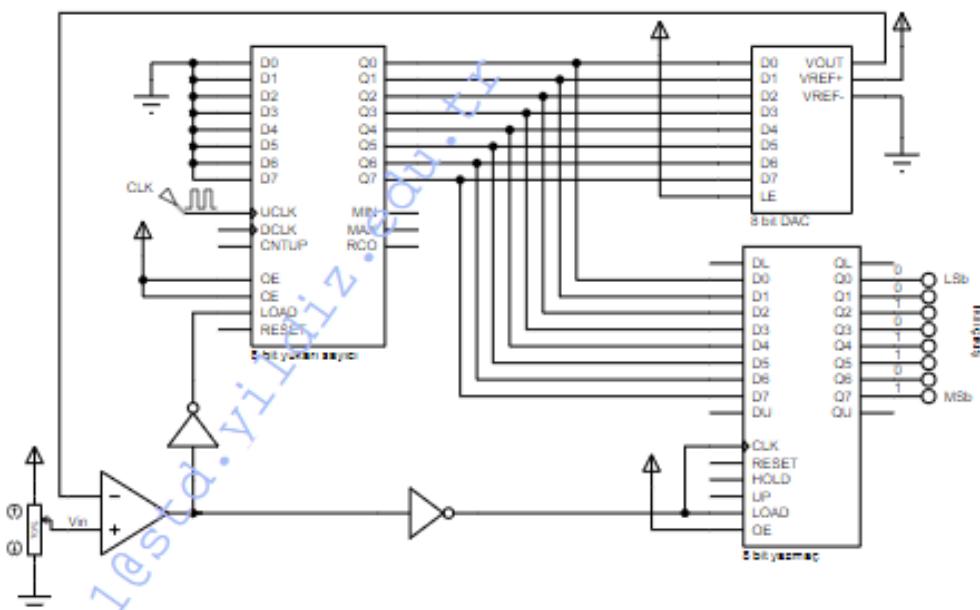


Şekil 21-5: Paralel ADC yapısı

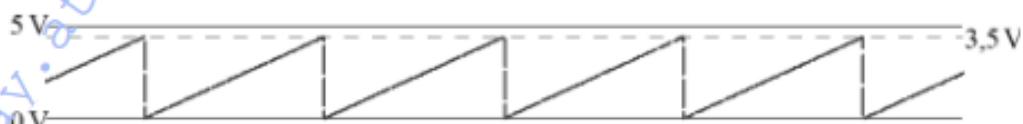
Rampa ADC'de n-bit'lik bir sayıcının sıfırdan itibaren adım adım sayması sağlanır. Sayıcının çıktıları DAC vasıtayıyla analog değere çevrilir. Yöntemin rampa ismi de

²¹⁴ Öncelik kodlayıcısında, lojik 1 girişi en anlamlı girişlerden itibaren ilk hangi girişle mevcutsa o numaralı çıkışa karşılık ikili bir sayı üretilecek ve çıkışta anlamlı veri olduğunu göstermek adına EO ucu lojik 1 yapılır. Eğer hiçbir girişte lojik 1 mevcut değilse EO ucunda lojik 0 üretilecektir.

DAC çıkışındaki analog değerin rampa fonksiyonu şekline artmasından gelir. Analog giriş gerilimi ile DAC çıktıları bir karşılaştırıcıya aktarılır. DAC çıktılarının analog giriş gerilimini aştığı anda bir yazmaç yükleme ucundan tetiklenerek o anki sayıçı değeri tutulmuş olur. Yazmaçta saklanan değer ADC sonucudur. Şekil 21-6 ile rampa ADC yapısı şematik olarak verilmiştir. Örneğin dönüştürmek istenen analog değerin bir potansiyometre yardımıyla sisteme verildiği durum için V_{in} ucunda 3,5 V analog gerilim değeri oluşturulmuştur ($5\text{ V} \times 0,7 = 3,5\text{ V}$). 8 bitlik sayıçı 0 ilk değeri ile yüklenir ve buna karşılık gelen 8 bit DAC sonucu da 0 V'tur. V_{in} değeri başlangıç itibariyle DAC çıkışından büyük olduğu için sayıçaya saat girişi frekansında yukarı doğru sayarken 8 bit yazmaç için yükleme ucu aktif değildir. Sayma değeri ve buna karşılık gelen DAC sonucu V_{in} değerini geçtiği anda yazmaç yükleme ucu aktif olarak o anki sayma değerini çıkışlarında tutar. Sayıçı yeniden 0 değeri ile yüklenerek süreç tekrarlanır. Yazmaç çıkışlarında oluşan değer ADC sonucudur. 3,5 V analog değerine karşılık ADC sonucu olarak $(10110100)_2$ değeri üretilmiştir. Rampa ADC'nin çalışma mantığını ortaya koymak adına, tekrarlanan süreç boyunca DAC çıktıları Şekil 21-7 ile verilmiştir.



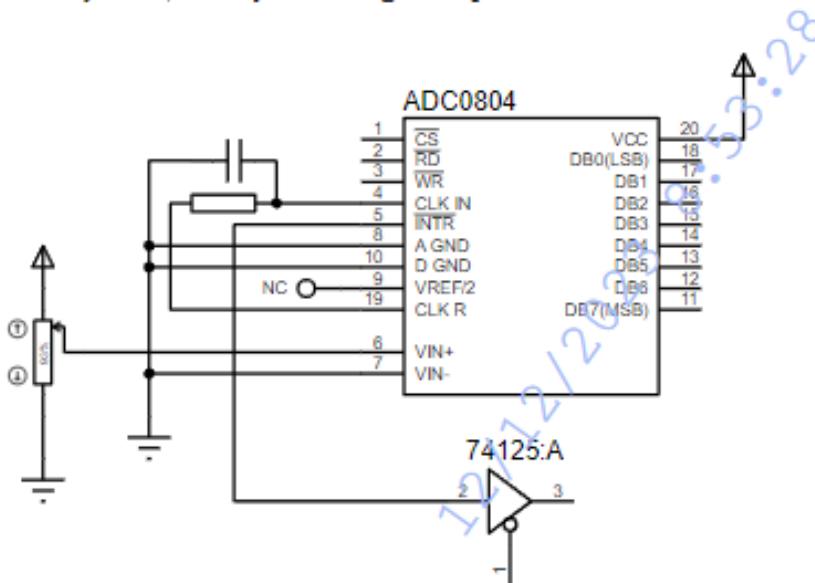
Şekil 21-6: Rampa ADC yapısı



Şekil 21-7: Rampa ADC'de yer alan DAC çıkışı

21.1 ADC0804

ADC0804, 8086 uyumlu, 8-bit çözünürlüğe sahip bir ADC tümdevresidir.



Şekil 21-8: ADC0804 minimum çalışan örnek bağlantıları

Analog girişe karşılık, $(0000\ 0000)_2$ - $(1111\ 1111)_2$ arasında ikili bir sayı üretir. Kitap kapsamında ADC0804 Şekil 21-8 ile verildiği, minimum çalışır bağlantı haliyle kullanılacaktır. Bu sebeple ADC0804'ü, dış dünya ile sadece analog gerilim girişi, INTR, CS, WR, RD ucları ve ikili veri çıkışları ile bağlantı sağlayan bir kara kutu olarak düşünebilirsiniz.

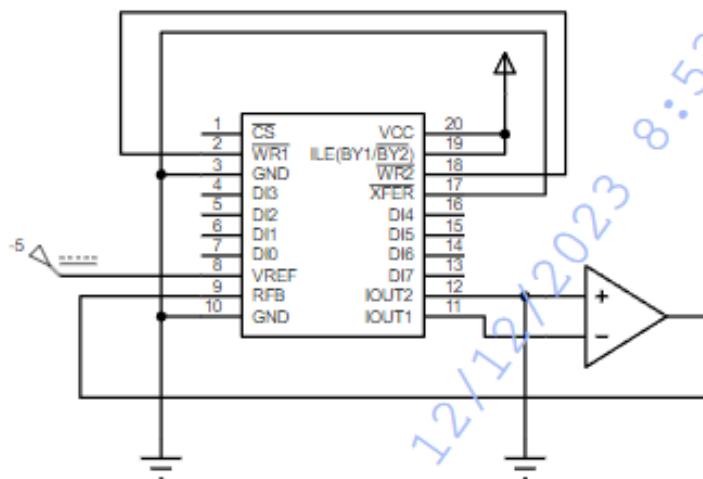
ADC0804'ün kullanımı dönüşüm başlatma ve dönüşüm sonucunu okuma adımlarından oluşur. ADC0804'e yapılan bir yazma işlemi²¹⁵ ile dönüşüm başlatılır. Bu adımda ADC0804'e yazılan değerin bir anlamı yoktur. Dönüşüm tamamlanması ile ADC0804 INTR ucunda lojik 0 değeri oluşturur. INTR=0 değeri yakalandığında ADC0804'ten yapılan okuma ile ADC sonucu elde edilmiş olur. ADC0804 okuma işlemi program kontrollü G/C (program controlled I/O) (bkz. Örnek 21-1) şeklinde yapılabileceği gibi kesme kontrollü G/C (interrupt controlled I/O) mantığında da yapılabilir (bkz. Örnek 22-5).

21.2 DAC0830

DAC0830, 8086 uyumlu 8-bit çözünürlüğe sahip bir DAC tümdevresidir. 8-bit sayısal girişe karşılık 0 V ile $-V_{ref}$ arasında bir analog gerilim değeri üretir. Kitap kapsa-

²¹⁵ ADC0804'de CS=0 ve WR=0 oluşturan durumdur.

mında DAC0830 Şekil 21-9 ile verildiği, minimum çalışır bağlantı haliyle kullanıla- caktır. Bu sebeple DAC0830'ü dış dünya ile sadece analog gerilim çıkışı, CS, WR uçları ve ikili veri girişleri ile bağlantı sağlayan bir kara kutu olarak düşünebilirsiniz.



Şekil 21-9: DAC0830 minimum çalışan örnek bağlantıları

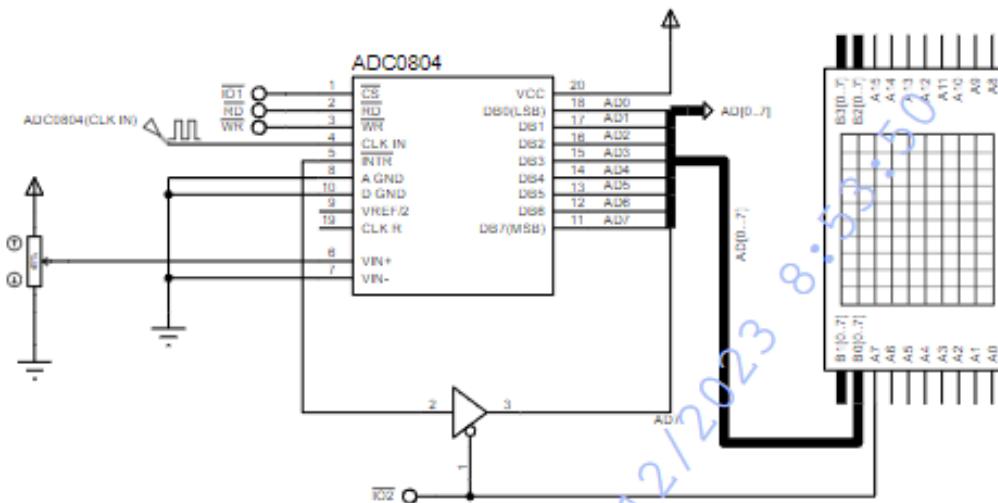
DAC0830'a yapılan bir yazma²¹⁶ işlemini takiben yazılan ikili değere karşılık analog değer Şekil 21-9 ile verilen op-amp çıkışında elde edilir.

21.3 Örnek ADC Uygulaması

Örnek 21-1

8086 G/C uzayında 200H adresine ADC0804 yerleştirilmek istenmektedir. ADC0804'ün INTR ucu ise bir üç durumlu tamponlayıcı (tri state buffer) ile veri hattına bağlanmıştır. INTR ucu için oluşturulan G/C arayüzü ise 400H adresine yerleştirilmek istenmektedir. Bunlara göre analog giriş ucuna bağlı olan potansiyometrede okunan değeri sayısalça çeviren assembly kodunu yazınız.

²¹⁶ DAC0830'da CS=0 ve WR=0 oluşturan durum



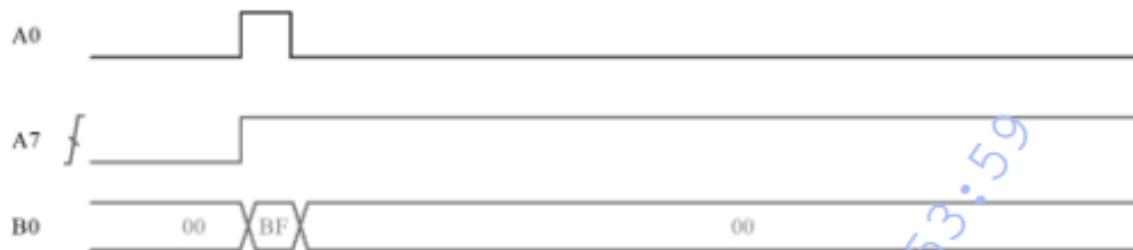
Şekil 21-10: Örnek 21-1 için bağlantı yapısı

PROGRAM : Örnek 21-1.ASM
ADC0804 kullanarak ADC dönüşümü başlatan, INTR ucu ile arayüz oluşturup dönüşüm bitmesini kontrol eden, dönüşüm bitince ADC'yi okuyan assembly program

```

PAGE 60,80
TITLE ADC0804 polling ile ADC okuma
STAK SEGMENT PARA STACK 'STACK'
DW 20 DUP(?)
STAK ENDS
CODE SEGMENT PARA 'CODE'
ASSUME CS:CODE, SS:STAK
START PROC FAR
ENDLESS: MOV DX,0200H ; ana yordam
          MOV AL,00H
          OUT DX, AL ; ADC0804'e boş yazma, dönüşüm başlatır
          MOV DX, 0400H
INTR_KONTROL:
          IN AL, DX ; INTR'yi oku
          TEST AL, 80H
          JNZ INTR_KONTROL ; INTR 0 ise dönüşüm bitmiştür
          MOV DX, 0200H
          IN AL, DX ; ADC0804 dönüşüm sonucunu oku
          CALL DELAY ; bir sonraki ADC dönüşümü öncesi delay
          JMP ENDLESS ; sonsuz döngü
START ENDP
; Bekleme sağlayan altyordam
DELAY PROC NEAR
PUSH CX
MOV CX, 0FH
L1: LOOP L1
POP CX
RET
DELAY ENDP
CODE ENDS
END START

```



Şekil 21-11: Örnek 21-1 için lojik analizör çıktıları

21.4 Örnek DAC Uygulaması

Örnek 21-2

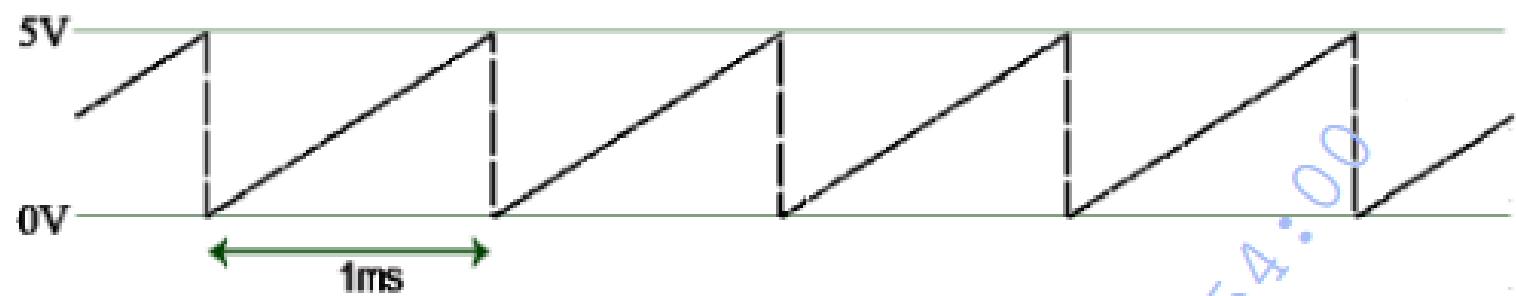
8086 G/Ç uzayında 1000H adresine DAC0830 yerleştirilmiş Buna göre

- 1 Hz frekansında 0 V – 5 V arasında değişen testere dışı dalga üreten assembly kodunu yazınız.
- DAC0830 çıkışını çiziniz.

```

;-----+
; PROGRAM : Örnek 21-2.ASM
; DAC0830 kullanarak 1Hz frekansında testere dışı analog dalga
; üreten assembly program
;-----+
PAGE 60, 80
TITLE DAC0830 testere dışı
STAK SEGMENT PARA STACK 'STACK'
DW 20 DUP(?)
STAK ENDS
CODE SEGMENT PARA 'CODE'
ASSUME CS:CODE, SS:STAK
START PROC FAR ; ana yordam
MOV DX, 1000H
MOV AL, 00H
ENDLESS: OUT DX, AL ; AL değerine karşılık DAC
CALL DELAY ; AL 250'den sonra tekrar 0'a döner
INC AL
CMP AL, 250
JB ENDLESS
XOR AL, AL
JMP ENDLESS ; sonsuz döngü
START ENDP
;-----+
; 4 ms bekleme sağlayan altyordam
;-----+
DELAY PROC NEAR
PUSH CX
MOV CX, 0100H
L1: LOOP L1
POP CX
RET
ENDP
CODE ENDS
END START

```



Şekil 21-12: Testere dışı dalga şekli

12/12/2023
8:54:00

bestd.yildiz.edu.tr

22. 8259 PIC (PROGRAMMABLE INTERRUPT CONTROLLER)

Bu bölümde kavramsal olarak donanım kesmeleri ve programlanabilir kesme kontrolcüsü olan 8259 anlatılacaktır.

22.1 Kesme Kaynakları

Kesme oluşturan iki temel kaynak mevcuttur, bunlardan ilki yazılımsal kaynaklar ikincisi ise donanımsal kaynaklardır. (Yazılım kaynaklı kesmeler için bkz. 14. bölüm.)

Kitabın bu bölümünde donanım kaynaklı kesmeler ele alınacaktır. 8086'ın donanım kesmelerini algılamak üzere iki tane kontrol bacağı mevcuttur. Bunlar INTR (Interrupt Request) ve NMI (Non-Maskable Interrupt) olarak adlandırılmış kullanımlarında belirgin farklılıklar vardır.

22.1.1 Maskelenemez Kesmeler (NMI)

Mikroişlemci üzerindeki NMI bacağı, çevre birimlerinde oluşan kritik durumların mikroişlemciye bildirilmesinde kullanılır. Bu bacağı gelen sinyaller (kesme istekleri) hemen değerlendirilir. Örneğin bellekte oluşan bir hata bu bacak üzerinden gönderilen işaret ile mikroişlemciye iletilmekte ve ekrana “**memory parity error**” gibi bir hata mesajını çıkartmaktadır. Genel olarak bellekte böyle bir hata oluşması durumunda sistemin çalışmaya devam etmesi mümkün değildir. Adından da anlaşılacağı gibi bu gruba dahil olan kesmelerin olması hiçbir şekilde engellenemez.

22.1.2 Maskelenebilir Kesmeler (INTR)

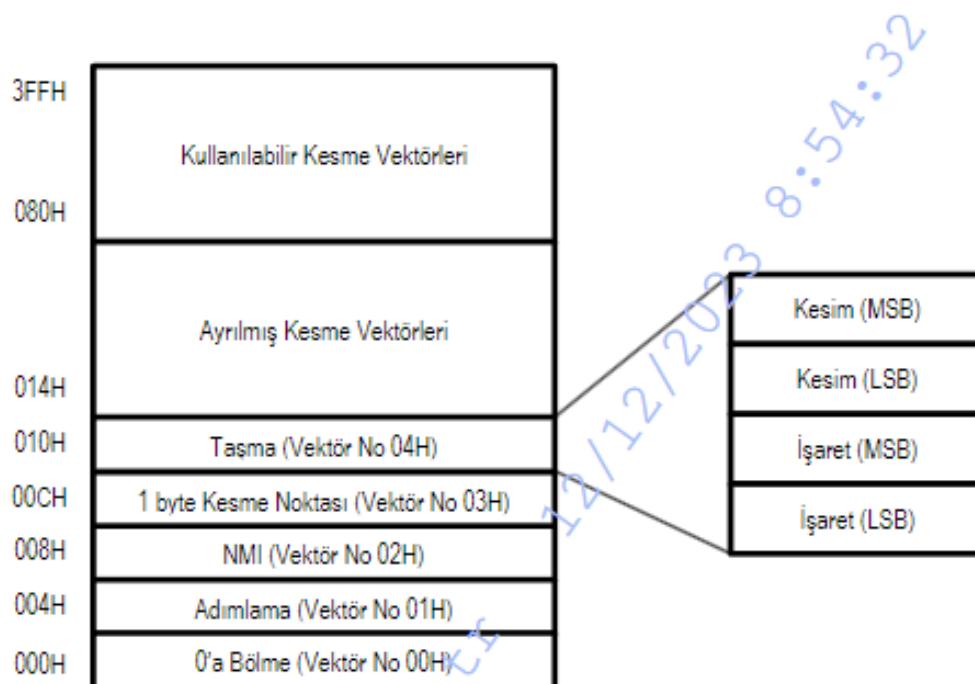
Mikroişlemci üzerindeki INTR (Interrupt Request) bacağı, özel donanımlar ile veya 8259 tümdevresi üzerinden sürülebilmektedir. 8259 tümdevresinin görevi, kullanıcının belirleyeceği durumlara bağlı olarak bazı çevre birimlerden gelen kesmelerin göz ardı edilebilmesini sağlamak, yani gelen istekleri belirlenen işlem politikası dâhilinde mikroişlemciye yansıtmadır. IF (Interrupt Flag) değerine bakarak bu tipteki kesmeler dikkate alınır ($IF \leftarrow 1$ -enable) veya alınmaz ($IF \leftarrow 0$ -disable). INTR ucundan gelen kesme isteklerinin 8086 tarafından karşılanması durumunda INTA ucu lojik 0'a getirilir.

22.2 Kesme Vektör Tablosu

Kesme olduğunda mevcut CS:IP ile gösterilen kodun yürütülmesi bırakılarak, kesmeye ilişkin fonksiyonun adresine ilişkin kesim ve işaret yazmaç değerleri yüklenir.

Her kesmeye ilişkin fonksiyonun hangi bellek adresinde yer aldığı **Kesme Vektör Tablosu** ile tutulur. Kesme Vektör Tablosu bellek uzayında 00000H – 003FFH adres aralığındaki 1024 byte'lık alandadır. Kesme Vektör Tablosu toplamda 256 farklı

kesme için ilgili fonksiyonların işaret ve kesim değerlerini saklayabilir. (bkz. 14.2. bölüm)



Şekil 22-1: Kesme Vektör Tablosunun yerlesimi ve içeriği

22.3 Kesme Yürütme Süreci

Donanım kaynaklı kesmelerin yürütülme süreci yazılım kaynaklı kesme yürütme süreçleri ile aynıdır (bkz. 14.1.4. bölüm).

22.4 Donanımsal Kesmeler

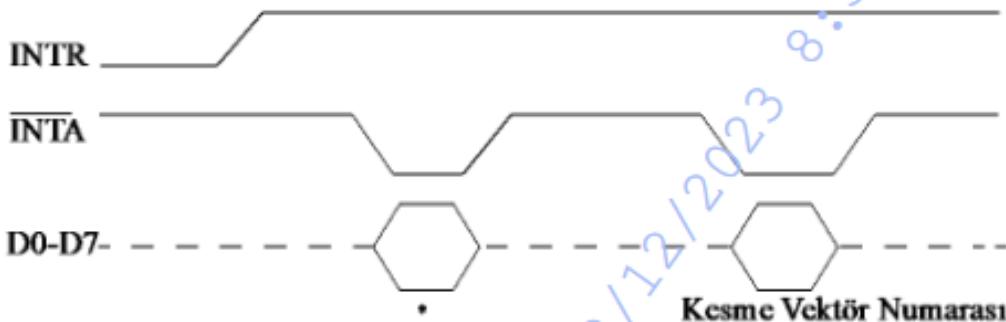
NMI ucu aktif olduğunda²¹⁷ kesme vektör tablosunun 008H adresinde yer alan, vektör numarası 02H olan kesme işlenir. NMI ucuna gelen kesme isteklerinde vektör numarası mikroişlemci tarafından içsel olarak otomatik bir şekilde 02H olarak çözümlenir.

INTR ucu aktif olduğunda²¹⁸ hangi numaralı kesme vektörünün yürütüleceği harici olarak mikroişlemciye bildirilmelidir. Intel 00H-1FH arasında yer alan vektör numaralarını mikroişlemci kullanımı için ayırmıştır. Genel kullanım için 20H- OFFH vektör numaraları kullanılabilir.

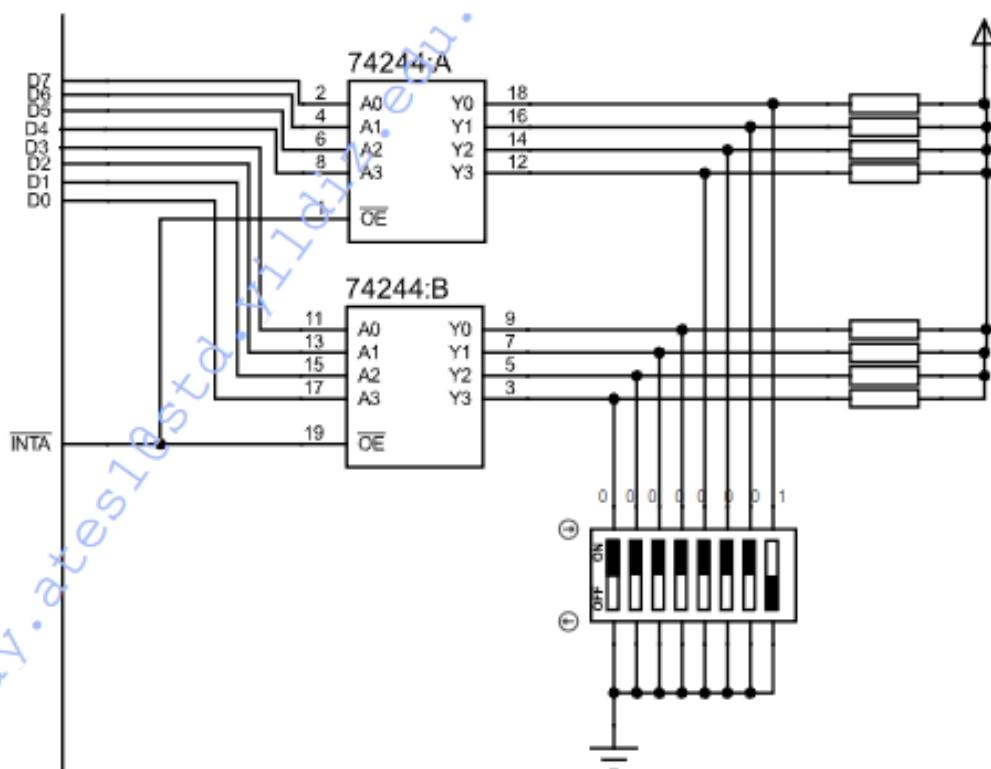
²¹⁷ NMI yükselen kenar ile tetiklenen bir girişir.

²¹⁸ INTR = 1 olduğunda seviye ile tetiklenen bir girişir.

8086'da INTR ucuna gelen kesme istekleri ve bunun karşılanmasıına yönelik dalga formu Şekil 22-2 ile verilmiştir. Burada bir kere daha vurgulamak gerekirse INTA ucunun aktif olmasıyla eşanlı olarak harici bir şekilde D0-D7 uçlarından sağlanan değer kesme servis programı adresi olmayıp, kesmenin vektör numarasıdır. Vektör tablosunda harici olarak verilen vektör numarasına gidildiğinde kesme servis programının adreslerine erişilecektir.



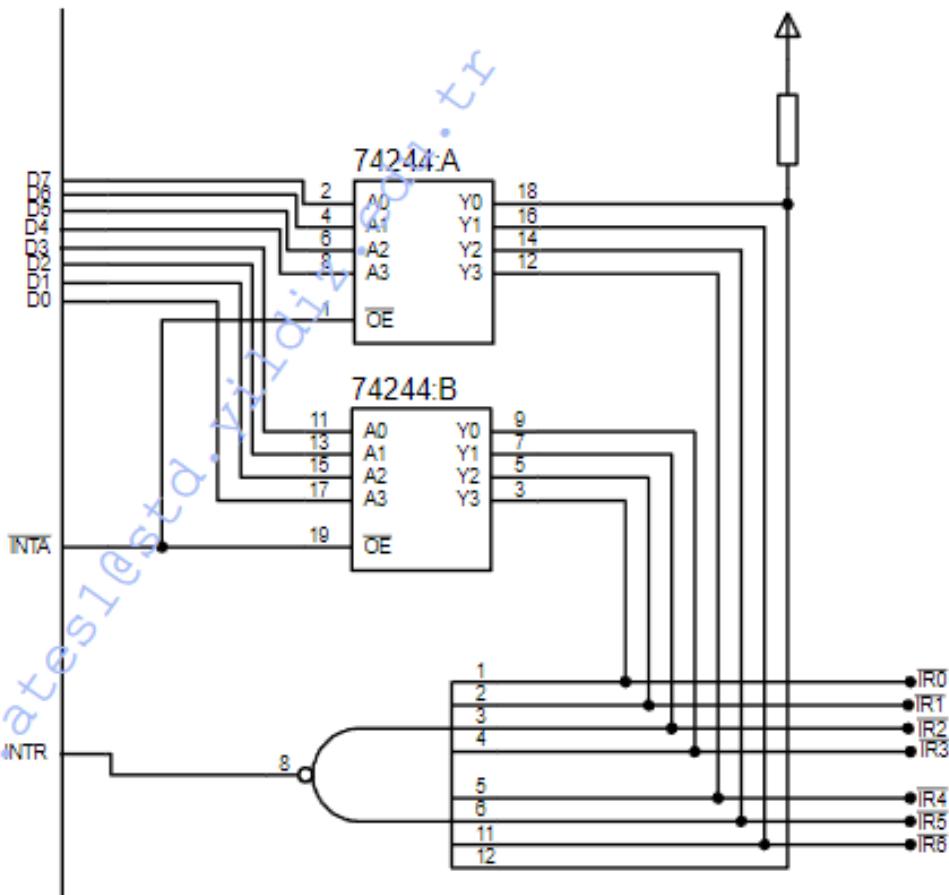
Şekil 22-2: Maskelenebilir kesme isteği ve cevaplanmasına ilişkin dalga formu
8086'da kesme istekleri karşılanırken ilki boş olmak üzere iki kere INTA ucu aktivitesi görülmektedir. 8086 kesme vektör numarasını ikinci INTA geçişinde beklemektedir.



Şekil 22-3: Sabit vektör numaralı basit kesme kontrolcüsü

Lojik devreler ile basit kesme kontrolcüleri oluşturulabilmektedir. Şekil 22-3 ile herhangi bir kaynaktan INTR ucuna gelen tüm kesme isteklerinin 80H vektör numarası ile karşılaşmasını sağlayan devre şemasına yer verilmiştir. Kesme isteği INTA ile karşılaşlığında, 8-bit tamponun girişlerinde kaldırma dirençleri ve dip anahtar (dip switch) ile oluşturulan sabit 8-bit'lik değer vektör numarası olarak D0-D7 uçlarına aktarılmış olur. Verilen devre ile 8086'ya sağlanan sabit vektör numarası 80H olarak hesaplanmaktadır.

Şekil 22-4 ile farklı kaynaklardan gelen kesme isteklerine ilişkin farklı vektör numaralarını sağlayan devre şemasına yer verilmiştir. Tüm kesme istekleri (IR0- IR6) aktif 0 sinyaller olarak verilmiştir. Herhangi bir kesme isteği aktif olduğunda NAND çıkışı lojik 1 değerini alarak INTR ucunda kesme isteği tetikler. Buna karşılık kesme isteği INTA ile karşılaşlığında aktif olan kesme isteklerine göre D0-D7 uçlarında farklı vektör numaraları sağlanmış olacaktır. Farklı kesme isteklerinin aktif olmasınayla D0-D7 uçlarında sağlanacak olan vektör numaraları Tablo 22-1 ile verilmiştir. Burada özellikle birden fazla kesme isteğinin aynı anda aktif olması durumunun da farklı vektör numaraları olarak ele alınabildiğine dikkat edilmelidir.

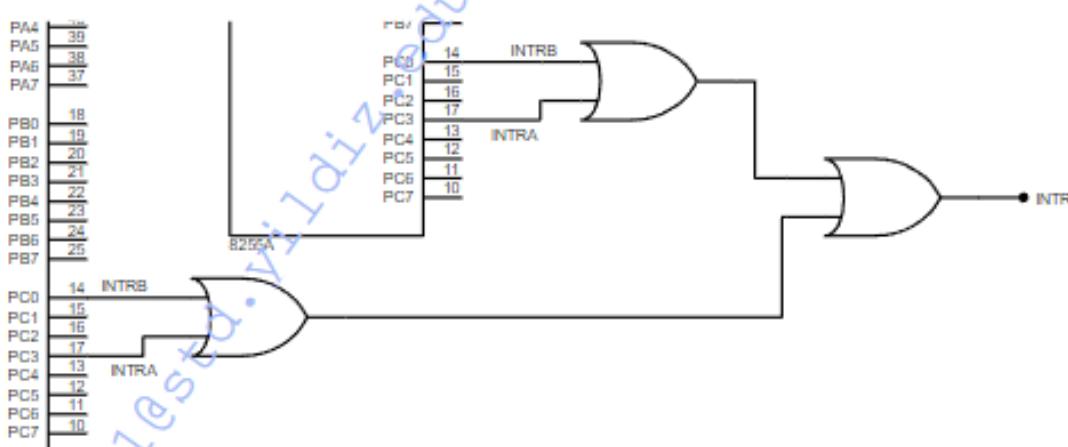


Şekil 22-4: Farklı kesme kaynakları için farklı vektör numaraları üretebilen basit kesme kontrolcüsü

Tablo 22-1: Şekil 22-4 için aktif kesme isteklerine karşılık sağlanan örnek kesme vektör numaraları

IR6	IR5	IR4	IR3	IR2	IR1	IR0	Kesme Vektör Numarası
1	1	1	1	1	1	0	0FEH
1	1	1	1	1	0	1	0FDH
0	1	1	1	1	1	1	0BFH
0	1	1	1	1	1	0	0BEH
1	1	1	1	1	0	0	0FCH

Özellikle, programlanabilir çevre birimlerinin ürettiği farklı kesme isteklerinin önceliklendirilebilmesi ve karşılanması için üçüncü bir yaklaşım ise, kesme istekuçlarının **papatya zinciri (daisy chain)** yapısında birleştirilmesi, tüm kesme isteklerine tek bir vektör numarası ile cevap verilmesi ve kesme servis programında çevre birimlerin durumlarını okuyarak kesme isteklerinin önceliklendirilmesi ve değerlendirilmesinin yazılım ile sağlanmasıdır. **Şekil 22-5** ile her ikisinde de GrupA ve GrupB portları Kip 1 ile ayarlanmış 8255 devre kesiti görülmektedir. Toplamda 4 farklı kesme kaynağı papatya zinciri ile bağlanarak mikroişlemcinin INTR ucunu tetiklemektedir. Kesme servis programında kesmenin hangi kaynaktan geldiği, belirlenen öncelik sırasına göre 8255 durum yazmalarının okunması ile bulunabilir.

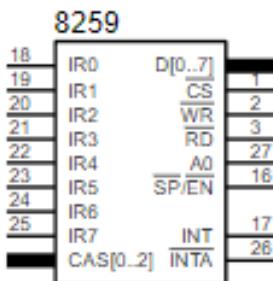


Şekil 22-5: Papatya zinciri ile kesme isteği önceliklendirme ve karşılama

Basit kesme kontrolcüler yanında programlanabilir kesme kontrolcüsü olan 8259 kullanarak, daha üst seviyede önceliklendirme, maskleme ve kaskat bağlantılar sayesinde arttırlılmış kaynak sayısı ile kesme kontrolü sağlanabilmektedir.

22.5 8259 Uç Tanımları

8259'a ilişkin uç yerleşimi ve tanımları sırasıyla **Şekil 22-6** ve **Tablo 22-2** ile verilmektedir.



Şekil 22-6: 8259 fonksiyonel uç tanımları

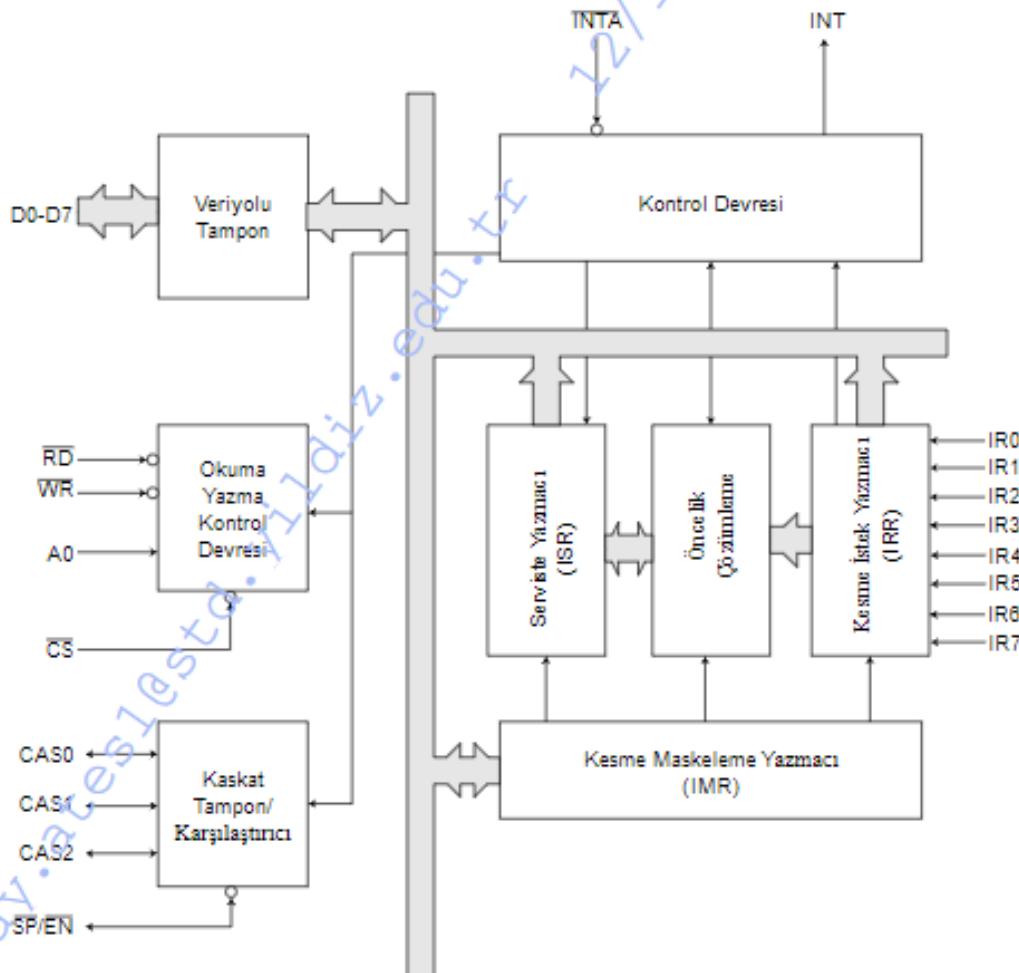
Tablo 22-2: 8259 uç tanımları

Uç İsmi	Tipi	Fonksiyonu
-CS	G	Seçim ucu; aktif 0 bir giriş ucudur. 8259 ile 8086 arasındaki okuma yazma işlemlerini aktif yapar. INTA ile ilgili iletişim bu seçim ucundan etkilenmez.
-WR	G	Yazma ucu; aktif 0 bir giriş ucudur. 8259'un 8086'dan komut kabul etmesini sağlar.
-RD	G	Okuma ucu; aktif 0 bir giriş ucudur. 8259'un veri yoluna durum bilgisini çikmasını sağlar.
D7-D0	G/Ç	Çift yönlü veri yolu. Kontrol komutları, durum bilgisi ve kesme vektör numarası bilgisi bu veri yolu ile aktarılır.
CAS0-CAS2	G/Ç	Kaskat bağlantı uçları. Çoklu 8259 kullanımı için kaskat bağlantı uçları olup, ana 8259 için çıkış, yardımcı 8259'lar için ise giriş yönlü çalışan uçlardır.
-SP/EN	G/Ç	(Slave Program / Enable Buffer) tamponlama kipinde 8259 iç mandallarını aktif eder, çoklu kipte SP = 1 ana 8259'u, SP = 0 yardımcı 8259'u belirler.
INT	Ç	Kesme ucu. 8086'nın INTR ucuna bağlanarak kesme isteğini iletir.
IR0-IR7	G	Kesme istek uçları. Kenar tetikleme ve seviye kipinde kesme isteği üretebilir.
-INTA	G	Kesme onayı. 8086 INTA ucuna bağlıdır. INTA ucunun aktif olması ile 8259 kesme vektör numarasını 8086'ya sağlar.
A0	G	8259 adres ucu. CS, WR, RD ile birlikte 8259 adreslemesinde ve programlanması sırasında kullanılır.

8259 programlanabilir kesme kontrol tümdevresi sekiz adet dış birimin kesme isteklerini karşılamak üzere kullanılmaktadır. 8259 yardımıyla INTR ucunda oluşturulan kesme istekleri, IF bayrağındaki değere göre engellenebilirken, NMI bacagi üzerinden gelen kesmeler üzerinde IF bayrağının hiçbir etkisi yoktur. 8259 kesmeler arasında öncelik belirlemeye kullanılır. Tek bir 8259, 8 farklı kesme isteğine karşılık 8 ardışık kesme vektör numarası sağlayabilir. Kaskat bağlı yapıda ise bir ana 8259 ve bunun her bir kesme istek ucuna bağlı sekiz 8259 daha kullanılarak toplamda 64 farklı kaynaktan gelen kesme isteklerine, kesme vektör numarası sağlanabilir.

22.6 8259 İç Yapısı

8259 iç yapısı; veri yolu kontrol birimi, okuma yazma kontrol birimi, maske yazmacı, kesme istek yazmaçları, kesme kontrol birimi ve 8259 iç yolu ile Şekil 22-7'de verilmektedir.



Şekil 22-7: 8259 iç yapısı

22.7 8259 Kontrol ve Ayar Yazmaçları

Kitap kapsamında tek bir 8259 kullanılarak kesme kontrolüne ilişkin ömekler ele alınacaktır. Bu sebeple örnek uygulamalarda (bkz. 22.9.3. bölüm) SP/EN=1 olarak kullanılmıştır.

8259, dört adet ilklendirme komut yazmacı (Initialization Command Word - ICW) ve üç adet operasyonel komut yazmacı (Operation Command Word - OCW) kullanılarak programlanır. 8259 ICW yazmaçlarının erişimde kullanılan adresler ve erişim akışı sırasıyla Tablo 22-3 ve Şekil 22-8 ile verilmiştir.

Tablo 22-3: 8259 ICW adresleme

$\neg CS$	A0	Erişim
0	0	ICW1
0	1	ICW2, ICW3, ICW4
1	X	8259 adreslenmez

ICW1 (Tablo 22-4), ICW2 (Tablo 22-5), ICW3 (yardımcı – ana kip, Tablo 22-6) ve ICW4 (Tablo 22-7) kontrol bit'leri fonksiyonları ile birlikte verilmiştir.

Tablo 22-4: 8259 ICW1 kontrol bit'leri

ICW1								
D7	D6	D5	D4	D3	D2	D1	D0	
0	0	0	1	LTIM	0	SGNL	ICW4	
8086 için (000) ₂		LTIM=1 → seviye tetiklemeli kesme isteği LTIM=0 → kenar tetiklemeli kesme isteği				SGNL=1 → tek 8259 SGNL=0 → kaskat 8259	ICW4=1 → ICW4 kullanılacak ICW4=0 → ICW4 kullanılma- yacak	

Tablo 22-5: 8259 ICW2 kontrol bit'leri

ICW2							
D7	D6	D5	D4	D3	D2	D1	D0
T7	T6	T5	T4	T3	T2	T1	T0
$(T_7 \dots T_0)_2$ IR0 kesme vektör tipi, diğer IR'ler buna göre sırayla değer alır							

8:55:29

Tablo 22-6: 8259 ICW3 kontrol bit'leri

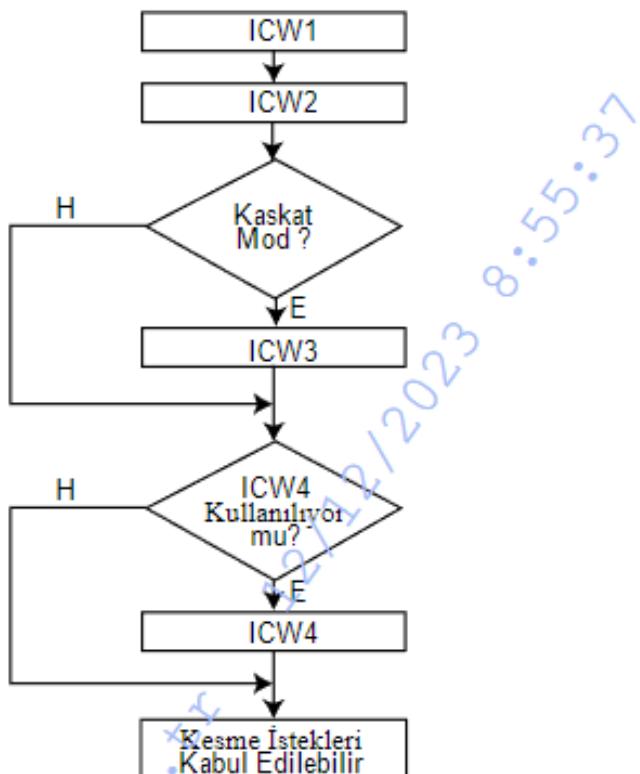
ICW3: kaskat bağlantıda yardımcı 8259 için							
D7	D6	D5	D4	D3	D2	D1	D0
S7	S6	S5	S4	S3	S2	S1	S0
$S_i = 1 \rightarrow$ IR'ya yardımcı 8259 bağlı $S_i = 0 \rightarrow$ IR'ya yardımcı 8259 bağlı değil							
ICW3: kaskat bağlantıda ana 8259 için							
D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	ID2	ID1
$(ID_2 ID_1 ID_0)_2$ yardımcı 8259 ID değeri							

Tablo 22-7: 8259 ICW4 kontrol bit'leri

ICW4							
D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	SFNM	BUF	MS	AEOI	μ PM
			SFNM=1 tamamen iç içe geçmiş özel mod	$(0x)_2 \rightarrow$ tek 8259 $(10)_2 \rightarrow$ kaskat kip yardımcı 8259 $(11)_2 \rightarrow$ kaskat kip ana 8259		AEOI=1 \rightarrow otomatik kesme sonlandırma AEOI=0 \rightarrow normal kesme sonlandırma	μ PM=1 \rightarrow 8086 (ilki boş iki INTA kullanımı) μ PM=0 \rightarrow MSC80 (tek INTA kullanımı)

Tek ana 8259 kullanımı için ICW4 için 03H değeri kullanılacaktır. Bu sayede kesme sonlandırmanın otomatik ele alındığı ve ilki boş olmak üzere iki INTA ile vektör numarası aktarımı sağlanabilir. ICW yazmaçlarına erişime ilişkin takip edilmesi gereken akış Şekil 22-8 ile verilmiştir.

berkay test



Şekil 22-8: 8259 ICW erişim akışı

8259 OCW yazımcılarının erişimde kullanılan adresler Tablo 22-8 ile verilmiştir.

Tablo 22-8: 8259 OCW adresleme

-CS	A0	Erişim
0	0	OCW2, OCW3
0	1	OCW1
1	X	8259 adreslenmez

OCW1 (Tablo 22-9), OCW2 (Tablo 22-10), ve OCW3 (Tablo 22-11) kontrol bit'leri fonksiyonları ile birlikte verilmiştir.

Tablo 22-9: 8259 OCW1 kontrol bit'leri

OCW0							
D7	D6	D5	D4	D3	D2	D1	D0
M7	M6	M5	M4	M3	M2	M1	M0
$M_i = 1 \rightarrow$ kesme maskele $M_i = 0 \rightarrow$ kesme maskeleME							

8:55:49

Tablo 22-10: 8259 OCW2 kontrol bit'leri

OCW2							
D7	D6	D5	D4	D3	D2	D1	D0
R	SL	EOI	0	0	L2	L1	L0
özel kesme sonlandırma komutları							özel kesme sonlandırma komutalarında IR seçimi

Tablo 22-11: 8259 OCW3 kontrol bit'leri

OCW3							
D7	D6	D5	D4	D3	D2	D1	D0
0	ESMM	SMM	0	1	P	RIR	RIS
özel maskeleme kipi							IR, IS yazmaç okuma komutları

Kitap kapsamında 8259 AEOI kipinde kullanıldığından örneklerde (bkz. 22.9.3. bölüm) OCW yazımcıları kullanılmamıştır.

8259'un okunabilir üç adet durum yazmacı bulunmaktadır. Bunlar, kesme istek yazmacı (IRR - interrupt request register), serviste yazmacı (ISR - in-service register) ve kesme maskeleme yazmacı (IMR - interrupt mask register) olarak sıralanabilir. IRR, ISR ve IMR 8 bit'lik yazımcılardır. IRR hangi kesme isteklerinin aktif olduğunu gösterir. ISR hangi kesme isteklerinin servisinin devam etmekte olduğunu gösterir. Örneğin IR6 ucundaki kesme isteği karşılanması tamamlanmadan IRR isteği de aktif olursa ISR = $(01000000)_2$ değerini alır. IR6 kesme isteğinin karşılanması tamamlanmadan IR3 isteği de aktif olursa ISR = $(01001000)_2$ değerini alır. IMR hangi kesme isteklerinin maskeleneceğini belirler. IRR ve ISR okumaları OCW3 D0-D1 bit'leri üzerinden programlanır, programlamayı takip eden bir sonraki okuma çevriminde IRR veya

bertil

ISR'den hangisi seçilmişse A0 = 0 adresinden okunur. IMR okuması A0 = 1 adresinden yapılır.

Kitapta kullanıldığı haliyle 8259 yardımıyla kesmenin ele alınması aşağıdaki süreç göre yürütülür:

- IR7-0 uçlarından biri veya birden fazlası 1'e çıkar
- 8259 istekleri değerlendirip, CPU'ya INTR gönderir
- CPU INTA ile karşılık verir
- INTA 8259'a ulaştığında, otomatik olarak en yüksek öncelikli kesmeye ilişkin ISR bit'i 1, IRR bit'i 0 yapılır
- CPU ikinci INTA darbesini gönderir, 8259 karşılık olarak kesme vektör numarasını veri yoluna koyar
- AEOI kipinde ISR bit'i 0 yapılarak kesme süreci sonlandırılır.

22.8 PC-XT ve PC-AT Bilgisayarlarda 8259 Kullanımı

8259 tümdevresi, 20H ve 21H numaralı iki iskele (port) kullanılarak programlanabilmektedir. 21H numaralı iskele, 8259 için maske bilgisini tutmak üzere kullanılırken, 20H adresindeki iskeleden de kontrol bilgisi yollanmaktadır. PC-XT'lerde bir adet 8259 tümdevresi ile sadece 8 harici kesme kontrol edilebilirken (Tablo 22-12), PC-AT'lerde kaskat (cascade) bağlanmış iki 8259 kullanılarak 15 adet harici kesme kontrol edilebilmektedir (Tablo 22-13).

Tablo 22-12: XT'ler için IRQ ve kontrol ettiğleri dış bağlantılar

Giriş hattı	INT#	Vektör Adresi	Harici birim
IRQ0	08H	0020H	Zamanlayıcı kanalı
IRQ1	09H	0024H	Klavye
IRQ2	0AH	0028H	Renkli grafik adaptörü
IRQ3	0BH	002CH	Seri-2
IRQ4	0CH	0030H	RS-232-C (Seri-1)
IRQ5	0DH	0034H	Sabit disk
IRQ6	0EH	0038H	Disket sürücü
IRQ7	0FH	003CH	Yazıcı

Tablo 22-13: AT'ler için IRQ ve kontrol ettiğleri dış bağlantılar

Giriş hattı	INT #	Vektör Adresi	Harici birim
IRQ0	08H	0020H	Zamanlayıcı kanalı
IRQ1	09H	0024H	Klavye
IRQ2	0AH	0028H	İkinci 8259
IRQ3	0BH	002CH	Seri-2
IRQ4	0CH	0030H	RS-232-C (Seri-1)
IRQ5	0DH	0034H	Yazıcı-2
IRQ6	0EH	0038H	Disket sürücü
IRQ7	0FH	003CH	Yazıcı-1
IRQ8	70H	01C0H	Gerçek zaman saatı-RTC
IRQ9	71H	01C4H	INT 0AH
IRQ10	72H	01C8H	Yedek
IRQ11	73H	01CCH	Yedek
IRQ12	74H	01D0H	Yedek
IRQ13	75H	01D4H	Matematik yardımcı işlemci
IRQ14	76H	01D8H	Sabit disk
IRQ15	77H	01DCH	Yedek

8259 tümdevresini programlamak için ilgili bit değerinin 0 (izinli – enabled) veya 1 (izinsiz – masked) olarak belirlenmesi gereklidir. Buna göre ilgili bit'lerden oluşan byte (PC-XT için) 8259 tümdevresinin 21H adresli iskelesine yazılır. Bir örnek olması açısından klavye ve zamanlama kesmesinin izinli diğerlerinin ise maskelenmiş olduğu bir maske Şekil 22-9 ile gösterilmiştir.

7	6	5	4	3	2	1	0	
1	1	1	1	1	1	0	0	→ FCH

Şekil 22-9: 8259 için maske değerinin belirlenmesi

Şekil 22-9 ile oluşturulan maske değeri 0FCH olup 8259 tümdevresinin 21H numaralı iskelesi üzerinden programlanmalıdır. Bu amaçla aşağıda yer alan kodun çalıştırılması gerekmektedir.

```
MOV AL, 0FCH
OUT 21H, AL
```

Maskenin yerine yazıldığını ve işlemin bittiğini ve bundan sonra verilen maske değerinin geçerli olacağını belirtmek için ise kontrol iskelesi olarak adlandırılan 20H numaralı iskeleye işlemin sona erdiğini bildiren özel kodun (20H) yazılması gereklidir.

```
MOV AL, 20H
OUT 20H, AL
```

22.9 Örnek Uygulamalar

Bu bölümde kesme servis programının (KSP) (Interrupt Service Routine - ISR) değiştirilmesine (Örnek 22-1), NMI kesmesi (Örnek 22-2), çevre birimleri ile 8259'un birlikte kullanıldığı kesme temelli G/Ç örneklerine (8255 için Örnek 22-3, 8251 için Örnek 22-4, ADC0804 için Örnek 22-5, 8254 için Örnek 22-6) yer verilmiştir.

22.9.1 Kesme Servis Programını Değiştirme

Örnek 22-1

8086'da sıfır bölme sonucunda 00H vektör numaralı kesme olmaktadır. Sıfır bölme durumunda AX yazmacında 0FFFFH değerini döndüren kesme servis programını oluşturarak, bu kesme servis programının 00H vektör numarasına bağlanmasılığını sağlayan assembly kodunu yazınız. Not: Sıfır bölme kesmesi sonucunda tekrar kesmeyi oluşturan komut adresine dönülür, dolayısıyla kesme servis programı yürütüldükten sonra ana programda bir sonraki komuta geçilmesi kesme servis programının sorumluluğundadır.

Örnek 22-1.ASM ile verilen akışa göre, 00H vektör numarasına bağlanan kesme servis programının ana yordam çalıştığı sürece aktif olduğuna dikkat edilmelidir (kalıcı şekilde kesme vektör tablosunun değiştirilmesi ile ilgili bkz. 14.2. bölüm)

```
; -----
; PROGRAM : Örnek 22-1.ASM
; DIV0 kesmesi karşılayıp sonuçta 0FFFFH dönen,
; kesme servis programını vektör tablosuna yerleştiren
; örnek assembly program
; -----
PAGE 60,80
TITLE DIV0 interrupt handler
STACKSD SEGMENT PARA STACK 'STACK'
DW 12 DUP(?)
STACKSG ENDS
DATASG SEGMENT PARA 'DATA'
DIZI1 DW 1,2,3,4,5,6,7
DIZI2 DW 1,0,3,4,5,6,7
ELEMAN DW 7
OLDOF DW 0
OLD SG DW 0
DATASG ENDS
CODESG SEGMENT PARA 'CODE'
ASSUME CS:CODESG, DS:DATASG, SS:STACKSG
```

```

; DIV0 kesme servis programı

NEWINT PROC FAR
    PUSH BP
    MOV BP, SP
    MOV AX, [BP+2]           ; AX <- ana yordam dönüş IP
    ADD AX, 2

; DIV0 kesmesini DIV komutu oluşturdu
; DIV komutu 2 byte yer kaplar
; DIV0 kesmesi dönüşünde DIV komutunu geçmek için
; IP bir sonraki komutu işaret eder

    MOV [BP+2], AX
    MOV AX, OFFFFFH          ; DIV0'dan AX <- OFFFFFH dönülür
    POP BP
    IRET
NEWINT ENDP
BASLA PROC FAR

; Dönüş için gerekli olan değerler yığında saklanıyor

    PUSH DS
    XOR AX, AX
    PUSH AX

; DATASG ismiyle tanımlı kesim alamına erişebilmek için gerekli tanımlar

    MOV AX, DATASG
    MOV DS, AX
    XOR AX, AX
    MOV ES, AX

; Vektör tablosundaki mevcut DIV0 CS ve IP değerleri saklanır

    MOV AX, WORD PTR ES:[0]
    MOV OLDOF, AX
    MOV AX, WORD PTR ES:[2]
    MOV OLDSG, AX

; 00H kesme vektör numarasının NEWINT'e bağlanması

    LEA AX, NEWINT          ; DIV0 için yeni KSP ofseti
    MOV WORD PTR ES:[0], AX
    MOV AX, CS                ; DIV0 için yeni KSP segmenti
    MOV WORD PTR ES:[2], AX

; dizi1 ve dizi2'nin karşılıklı elemanları bölünür
; SI=1 iken 2/0 durumda AX=0FFFFH değerini alır

    XOR SI, SI
    MOV CX, ELEMAN
    XOR DX, DX
    MOV AX, DIZI1[SI]
    MOV BX, DIZI2[SI]
    DIV BX
    ADD SI, 2
    LOOP TEKRAR

; Program sonlandırılmadan önce orijinal DIV0 KSP adresleri geri yazılır

    XOR AX, AX
    MOV ES, AX

```

```

        MOV AX, OLDOF
        MOV WORD PTR ES:[0], AX
        MOV AX, OLDSG
        MOV WORD PTR ES:[2], AX
        RETF
BASLA    ENDP
CODESG   ENDS
END BASLA

```

22.9.2 NMI Kesmesi

Örnek 22-2

8086'da NMI ucuna indirme direnci ile sürülen bir buton bağlanmıştır²¹⁹. NMI kesmesini kullanarak butona kaç kere basıldığını bularak sonucu AX yazmacı üzerinden döndüren assembly kodunu yazınız.

```

; -----
; PROGRAM : Örnek 22-2.ASM
; NMI kesmesi kullanarak buton sayma
; assembly program
; -----
PAGE 60,80
TITLE NMI kesmesi kullanarak buton sayma
STAK     SEGMENT PARA STACK 'STACK'
DW 20 DUP(?)
STAK     ENDS
CODE     SEGMENT PARA 'CODE'
ASSUME CS:CODE, SS:STAK
; -----
; NMI kesmesi için KSP
; -----
NEWINT   PROC FAR
; -----
; NMI yükselen kenar tetiklemeli
; Her butona basıldığında NMI kesmesi oluşur
; Her NMI kesmesinde AX = AX+1
; -----
INC AX
IRET
NEWINT   ENDP
START    PROC FAR
XOR AX, AX
MOV ES, AX
MOV AL, 2H
MOV AH, 4
MUL AH
MOV BX, AX
; -----
; 02H kesme vektör numarasının NEWINT'e bağlanması
; -----
LEA AX, NEWINT
MOV WORD PTR ES:[BX], AX
MOV AX, CS
MOV WORD PTR ES:[BX+2], AX

```

²¹⁹ NMI ucu yükselen kenarda tetiklenir.

```

ENDLESS: XOR AX, AX      ; butona kaç kere basıldığı AX'te
          JMP ENDLESS    ; sonsuz döngü
START     ENDP
CODE      ENDS
          END START

```

22.9.3 8259 Uygulamaları

Bu bölümde 8259 yardımıyla 8255, 8251, 8254 ve ADC0804 temelli kesmelerin ele alındığı örnekler anlatılacaktır.

22.9.3.1 8255 Kaynaklı Kesme Uygulaması

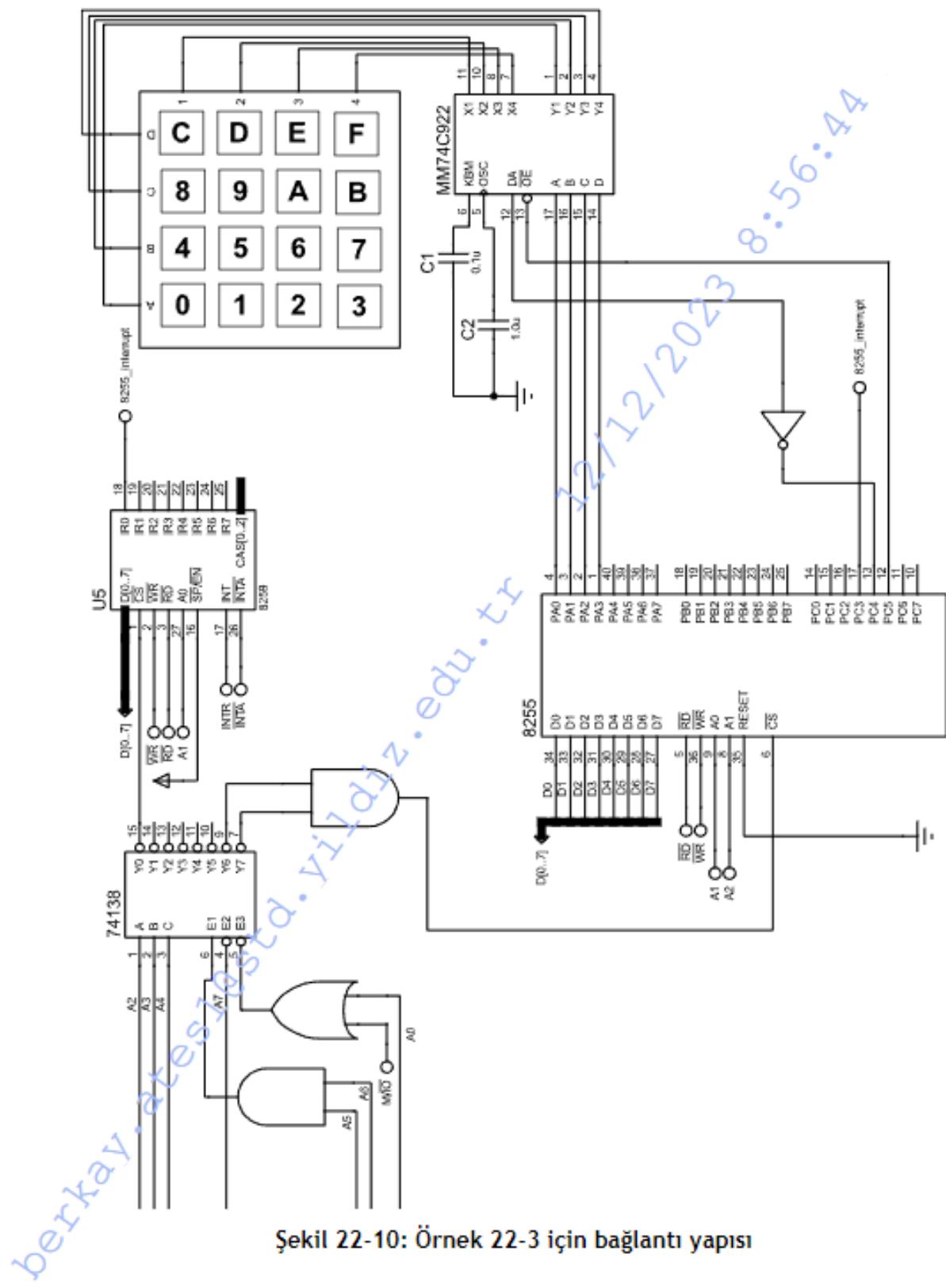
Örnek 22-3

8086 G/C uzayında 78H adresinden itibaren ardışık çift adreslere 8255 ve 60H adresinden itibaren ardışık çift adreslere ise 8259 yerleştirilmek istenmektedir. 8255 GrupA ile kesme temelli olarak 4×4 tuş tarama takımı sürülmek istenmektedir. Bu amaçla PortA düşük anlamlı 4 bit'ine ($PA_3PA_2PA_1PA_0$) tuş tarama tümdevresi (74922 – basılan tuş karşılığı 4 bit değer döner) veri uçları (DCBA) anlam sırasıyla bağlanmıştır. Tuş tarama takımında bir tuşa basıldığında, tuş değerinin, 30H vektör numarasına bağlı kesme servis programı ile okunarak veri kesimindeki TUS isimli değişkene aktarılması istenmektedir. Buna göre:

- Gerekli adres çözümleme devresini 3×8 kodçözücü ve gerekli basit lojik kapılar kullanarak tasarluyız.
- 8255 GrupA için uygun ayarlamayı yapınız
- 8259 için uygun ayarlamayı yapınız
- 8255 GrupA ile gelen verinin okunmasını sağlayan kesme servis programını yapınız
- d şıkkındaki kesme servis programının 30H kesme vektör numarasına bağlanması yapınız

Şekil 22-10 ile Örnek 22-3 için gerekli bağlantılar verilmiştir.

berkay.ates1@su.edu.tr



Şekil 22-10: Örnek 22-3 için bağlantı yapısı

```

;-----+
; PROGRAM : Örnek 22-3.ASM
; 8259 ve 8255 kullanarak kesme tabanlı tuş tarama takımı okuyan
; assembly program
;-----+
        PAGE 60,80
        TITLE 8259 ve 8255 ile kesme tabanlı tuş tarama takımı okuma
STAK     SEGMENT PARA STACK 'STACK'
        DW 20 DUP(?)
STAK     ENDS
DATA     SEGMENT PARA 'DATA'
TUS      DB 0FFH
DATA     ENDS
CODE    SEGMENT PARA 'CODE'
ASSUME CS:CODE, DS:DATA, SS:STAK
;-----+
; 8255 GrupA kip 1 giriş için kesme servis program
;-----+
NEWINT   PROC FAR
        IN AL, 78H          ; tuş değerini okur
        AND AL, 0FH          ; okunan low nibble maskelenir
        MOV TUS, AL          ; okunan değer TUS değişkeninde
        IRET
NEWINT   ENDP
START    PROC FAR
;-----+
; DATA ismiyle tanımlı kesim alanına erişebilmek için gerekli tanımlar
;-----+
        MOV AX, DATA
        MOV DS, AX
;-----+
; 30H kesme vektör numarasının NEWINT'e bağlanması
;-----+
        XOR AX, AX
        MOV ES, AX
        MOV AL, 30H
        MOV AH, 4
        MUL AH
        MOV BX, AX
        LEA AX, NEWINT
        MOV WORD PTR ES:[BX], AX
        MOV AX, CS
        MOV WORD PTR ES:[BX+2], AX
;-----+
; 8255 ayarları
;-----+
        MOV AL, 09H          ; GrupA kip 1 giriş
        OUT 7EH, AL          ; BSR kipte INTR aktif
        MOV AL, 0B8H
        OUT 7EH, AL
;-----+
; 8259 ayarları
;-----+
        MOV AL, 13H          ; kenar tetikleme IR, tek 8259
        OUT 60H, AL
        MOV AL, 30H          ; IRO kesme tipi 30H
        OUT 62H, AL
        MOV AL, 03H
        OUT 62H, AL
        STI                  ; AEOI, 8086
        IF<-1, INTR kesmeleri aktif
ENDLESS: JMP ENDLESS
START    ENDP
CODE    ENDS
END START

```

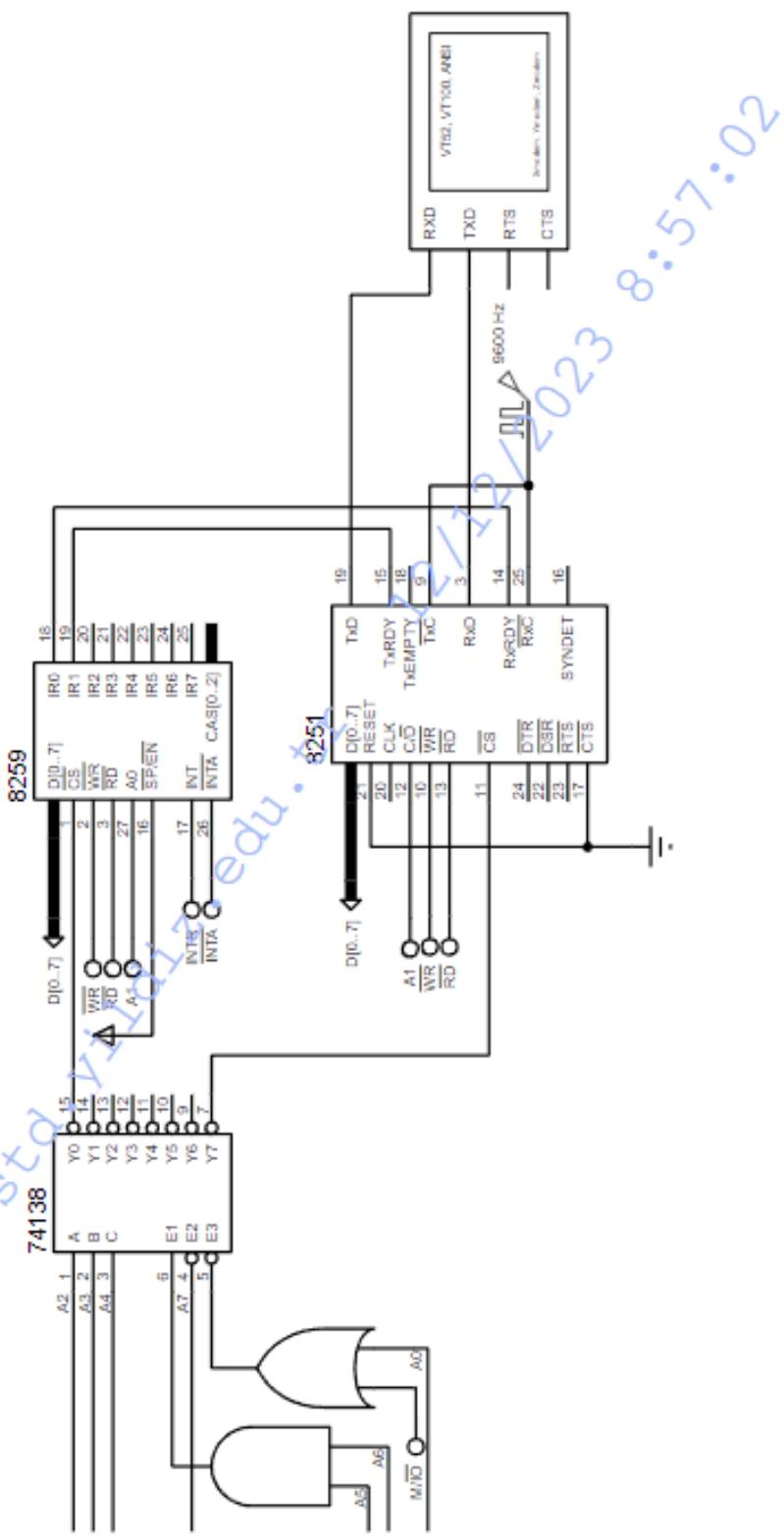
22.9.3.2 8251 Kaynaklı Kesme Uygulaması

Örnek 22-4

8086 G/Ç uzayında 7CH adresinden itibaren ardışık çift adreslere 8251 ve 60H adresinden itibaren ardışık çift adreslere ise 8259 yerleştirilmek istenmektedir. Bir sanal terminalden (9600 baud, sekiz veri bit'i, eşlik kullanmadan, bir durma bit'i kullanacak şekilde ayarlanmış) gelen veriyi kesme mantığında okuyup cevap veren bir program yapılması istenmektedir. Bunu için:

- Gerekli adres çözümleme devresini 3×8 kodçözücü ve gerekli basit lojik kapılar kullanarak tasarılayınız
- Seri veri almaya ilişkin kesme servis programını 78H kesme vektör numarasına, seri veri göndermeye ilişkin kesme servis programını ise 79H kesme vektör numarasına bağlamak için gerekli assembly kodunu yazınız
- 8259'u kesme kullanımı için uygun şekilde ayarlayınız
- 8251'i kesme kullanımı için uygun şekilde ayarlayınız
- Seri veri alma kesme servis programı sanal terminalden gelen ASCII karakterlerin değerce bir büyüğünü bir dizide saklamalı, alınan veri adedi beş olduğunda bu beş elemanı kesme mantığında geri göndermeli. Gerekli assembly kodlarını yazınız (örneğin sanal terminalden sırayla 'a', 'b', 'c', 'd', 'e' karakterleri basılmışsa 'e' karakterinden sonra 8086 cevap olarak 'bcdef' karakter dizisini göndermeli).

Şekil 22-11 ile Örnek 22-4 için bağlantı yapısı verilmiştir.



Şekil 22-11: Örnek 22-4 için bağlantı yapısı

```

; -----[PROGRAM : Örnek 22-4.ASM]-----
; 8259 ve 8251 veri alma ve gönderme kesmeleri yardımıyla alınan karakter
; sayısını 5 olunca topluca bu 5 karakteri ASCII karşılıklarına 1
; ekleyerek gönderen assembly program
;
; -----[PAGE 60,80]
; -----[TITLE 8259 ve 8251 kullanarak kesme tabanlı veri alma gönderme]
STAK    SEGMENT PARA STACK 'STACK'
        DW 20 DUP(?)
;
STAK    ENDS
DATA    SEGMENT PARA 'DATA'
MYDAT   DB 5 DUP(0)
DATASAY  DW 0
DATASEND SAY DW 0
ALLSENT DB 1
DATA    ENDS
CODE    SEGMENT PARA 'CODE'
        ASSUME CS:CODE, DS:DATA, SS:STAK
;
; -----[8251 seri veri alma KSP]
;
NEWINT  PROC FAR
        IN AL, 7CH           ; gelen karakteri oku
        INC AL               ; ASCII bir fazlasını sakla
        MOV SI, DATASAY
        MOV MYDAT[SI], AL
        INC SI
        CMP SI, 5            ; 5 karakter biriktirilmiş mi
        JNE DEVAM
        XOR SI, SI           ; alma indisini sıfırla
        MOV ALLSENT, 0
        MOV AL, MYDAT[0]
        OUT 7CH, AL           ; 5 karakterden ilkini gönder
        INC DATASEND SAY
DEVAM:  MOV DATASAY, SI
        IRET
NEWINT  ENDP
;
; -----[8255 seri veri gönderme KSP]
;
NEWINT2 PROC FAR
        CMP ALLSENT, 1         ; 5 karakterin gönderimi bitti mi
        JE DEVAM2
        MOV DI, DATASEND SAY
        MOV AL, MYDAT[DI]
        OUT 7CH, AL           ; sıradaki karakteri gönder
        INC DI               ; bir sonraki karakter indisı
        CMP DI, 5
        JNE DEVAM2
        MOV ALLSENT, 1         ; 5 karakterin gönderimi bitti
        XOR DI, DI           ; gönderme indisini sıfırla
DEVAM2: MOV DATASEND SAY, DI
        IRET
NEWINT2 ENDP
START  PROC FAR
;
; -----[DATA ismiyle tanımlı kesim alamına erişebilmek için gerekli tanımlar]
;
        MOV AX, DATA
        MOV DS, AX
;
; -----[78H kesme vektör numarasının NEWINT'e bağlanması]
;
```

```

XOR AX, AX
MOV ES, AX
MOV AL, 78H
MOV AH, 4
MUL AH
MOV BX, AX
LEA AX, NEWINT
MOV WORD PTR ES:[BX], AX
MOV AX, CS
MOV WORD PTR ES:[BX+2], AX
; 79H kesme vektör numarasının NEWINT2'ye bağlanması
;-----[8:57:10]
MOV AL, 79H
MOV AH, 4
MUL AH
MOV BX, AX
LEA AX, NEWINT2
MOV WORD PTR ES:[BX], AX
MOV AX, CS
MOV WORD PTR ES:[BX+2], AX
;-----[12:12:2023]
; 8251'in koşullandırılması
;-----[8:57:10]
MOV AL, 01001101B
OUT 7EH, AL
MOV AL, 40H
OUT 7EH, AL
MOV AL, 01001101B
OUT 7EH, AL
MOV AL, 15H
OUT 7EH, AL
;-----[8:57:10]
; 8259'un koşullandırılması
;-----[12:12:2023]
MOV AL, 13H
OUT 60H, AL
MOV AL, 78H ; IRO 78H tipinde kesme sağlar
OUT 62H, AL
MOV AL, 03H
OUT 62H, AL
STI ; IF<-1, INTR kesmeleri aktif
; sonsuz döngü
ENDLESS: JMP ENDLESS
START ENDP
CODE ENDS
END START

```

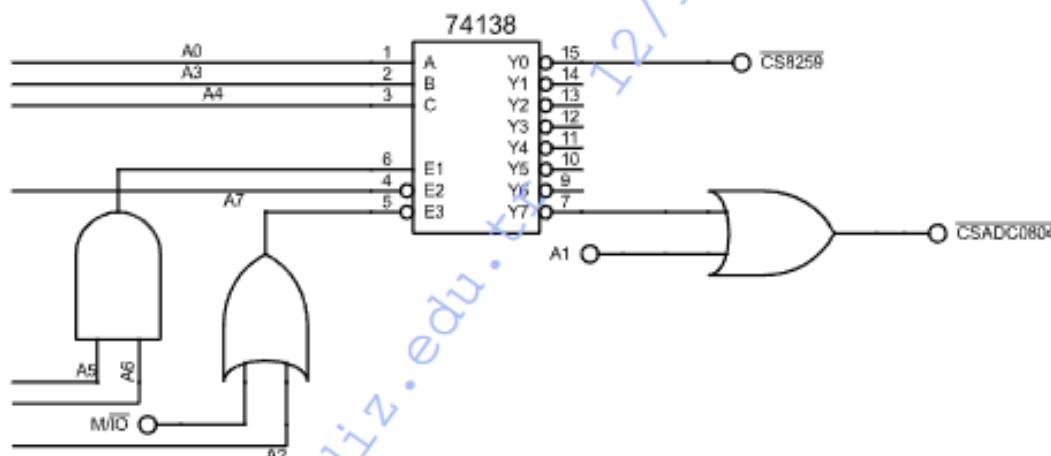
22.9.3.3 ADC0804 Kaynaklı Kesme Uygulaması

Örnek 22.5

ADC0804, 8259 ve bir karşılaştırıcı (comparator) kullanılarak kesme tabanlı bir tuş tarama sistemi oluşturulmak istenmektedir. 4 adet buton bir direnç dizisi yardımıyla Şekil 22-13 ile verildiği gibi bağlanmıştır. Basıldığımda her buton için D isimli düğümde farklı analog gerilim değerleri elde edilmektedir. Karşılaştırıcının + ucuna D düğümü, - ucuna ise 0,3 V bağlanmıştır. D düğümündeki gerilim 0,3 V'tan büyük olduğunda karşılaştırıcı çıkışı lojik 1 (5 V), diğer durumlarda 0 V olmaktadır. ADC'nin analog girişine D düğümü bağlıdır. Karşılaştırıcı çıkışı 8259'da IR7 ucuna

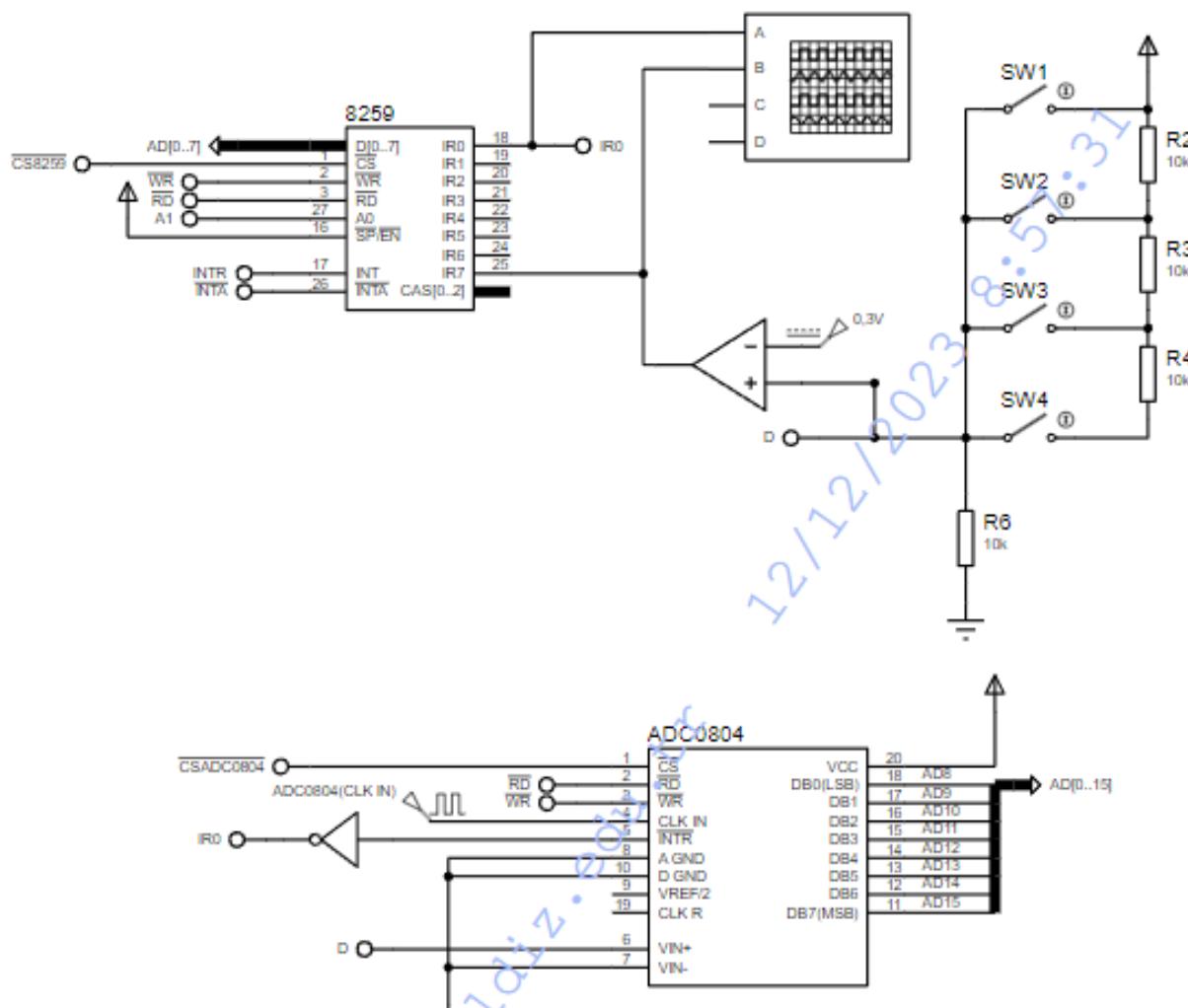
bağlıdır. ADC'nin INTR ucu ise 8259'da IR0 ucuna bağlıdır. Verilenlere ilişkin kesme tabanlı tuş tarama mantığı assembly kodunu aşağıdaki adımları dikkate alarak yazınız:

- ADC0804 ve 8259'nın yerleştirilmiş olduğu adresleri bulun.
- Herhangi bir butona basıldığında 47H vektör numarasına bağlı kesme servis programının çağrılmamasını sağlayan ayarlamayı yapın.
- 47H vektör numarası ile ilgili kesme servis programında ADC dönüşümü başlatın.
- ADC dönüşümü bitince 40H vektör numarasına bağlı kesme servis programının çağrılmamasını sağlayan ayarları yapın.
- 40H vektör numarası ile ilgili kesme servis programında hangi butona basıldığını BL yazmacında saklayın.
- 8259 ayarlarını yapın.



Şekil 22-12: Örnek 22-5 için adres çözümleme devresi

Karşılaştırıcı yardımıyla herhangi bir tuşa basılıp basılmadığı anlaşılabilmektedir. Tuş takımında hiçbir tuşa basılmamışsa karşılaştırıcının V+ ucuna 0 V gelecek olup çıkışında da lojik 0 üretilir. Herhangi bir tuşa basıldığında ise karşılaştırıcı V+ ucu 0,3 V'tan büyük olacağı için karşılaştırıcı çıkışında lojik 1 üretilir. Bu ise IR7'den bir kesme isteğine karşılık gelir. Basılan tuşa göre D ucunda olacak gerilim değeri farklıdır. Tuş basılması ile tetiklenen kesme servis programında bir ADC dönüşümü başlatılır. ADC dönüşümü bittiğinde ise bu sefer ADC'nin INTR ucu vasıtasiyla IR0'dan bir kesme isteği oluşturulur. ADC dönüşümünün bitmesiyle tetiklenen kesme servis programında ise ADC dönüşüm sonucu okunur.



Şekil 22-13: Örnek 22-5 için bağlantı yapısı

PROGRAM :Örnek 22-5.ASM
8259, ADC0804, karşılaştırıcı yardımcıyla analog gerilim değerlerinden faydalananak kesme tabanlı tuş tarama assembly program

```

PAGE 60,80
TITLE 8259, ADC0804, karşılaştırıcı ile kesme tabanlı tuş tarama
STAK SEGMENT PARA STACK 'STACK'
DW 20 DUP(?)
ENDS
DATA SEGMENT PARA 'DATA'
DB 20 DUP(0)
ENDS
CODE SEGMENT PARA 'CODE'
ASSUME CS:CODE, DS:DATA, SS:STAK

```

ADC dönüşümü bitince çağrılacak olan KSP

```

NEWINT    PROC FAR
        PUSH BP
        MOV BP, SP
        IN AL, 79H      ; ADC dönüşüm sonucu okunur
        MOV BL, 1
        CMP AL, 0BFH
        JA BUTTON_READ ; SW1'e basıldıysa ADC sonucu 0BFH'dan büyük
        MOV BL, 2
        CMP AL, 6AH
        JA BUTTON_READ ; SW2'e basıldıysa ADC sonucu 6AH'dan büyük
        MOV BL, 3
        CMP AL, 4AH
        JA BUTTON_READ ; SW3'e basıldıysa ADC sonucu 4AH'dan büyük
        MOV BL, 4
        ; SW4'e basıldıysa ADC sonucu 4AH'dan küçük
BUTTON_READ:
        POP BP
        IRET
NEWINT    ENDP
; -----
; Herhangi bir tuşa basıldığında tetiklenecek KSP
; -----
NEWINT2   PROC FAR
        PUSH BP
        MOV BP, SP
        OUT 79H, AL      ; ADC'ye boş yazma yaparak dönüşüm başlatır
        POP BP
        IRET
NEWINT2   ENDP
START     PROC FAR
; -----
; DATA ismiyle tanımlı kesim alamına erişebilmek için gerekli tanımlar
; -----
        MOV AX, DATA
        MOV DS, AX
; -----
; 40H kesme vektör numarasının NEWINT'e bağlanması
; -----
        XOR AX, AX
        MOV ES, AX
        MOV AL, 40H
        MOV AH, 4
        MUL AH
        MOV BX, AX
        LEA AX, NEWINT
        MOV WORD PTR ES:[BX], AX
        MOV AX, CS
        MOV WORD PTR ES:[BX+2], AX
; -----
; 47H kesme vektör numarasının NEWINT2'ye bağlanması
; -----
        XOR AX, AX
        MOV ES, AX
        MOV AL, 47H
        MOV AH, 4
        MUL AH
        MOV BX, AX
        LEA AX, NEWINT2
        MOV WORD PTR ES:[BX], AX
        MOV AX, CS
        MOV WORD PTR ES:[BX+2], AX

```

```

; 8259'un koşullandırılması
;
    MOV AL, 13H
    OUT 60H, AL
    MOV AL, 40H
    OUT 62H, AL
    MOV AL, 03H
    OUT 62H, AL
    CALL FAR PTR DELAY ; 8259 ve ADC0804 hazır olması için bekleme
    STI ; IF<-1, INTR kesmeleri aktif
ENDLESS: JMP ENDLESS ; sonsuz döngü
START ENDP
;
; Bekleme sağlayan altyordam
;
DELAY PROC FAR
    PUSH CX
    MOV CX, 000FH
L1:   LOOP L1
    POP CX
    RET
DELAY ENDP
CODE ENDS
END START

```

22.9.3.4 8254 ve ADC0804 Kaynaklı Kesme Uygulaması

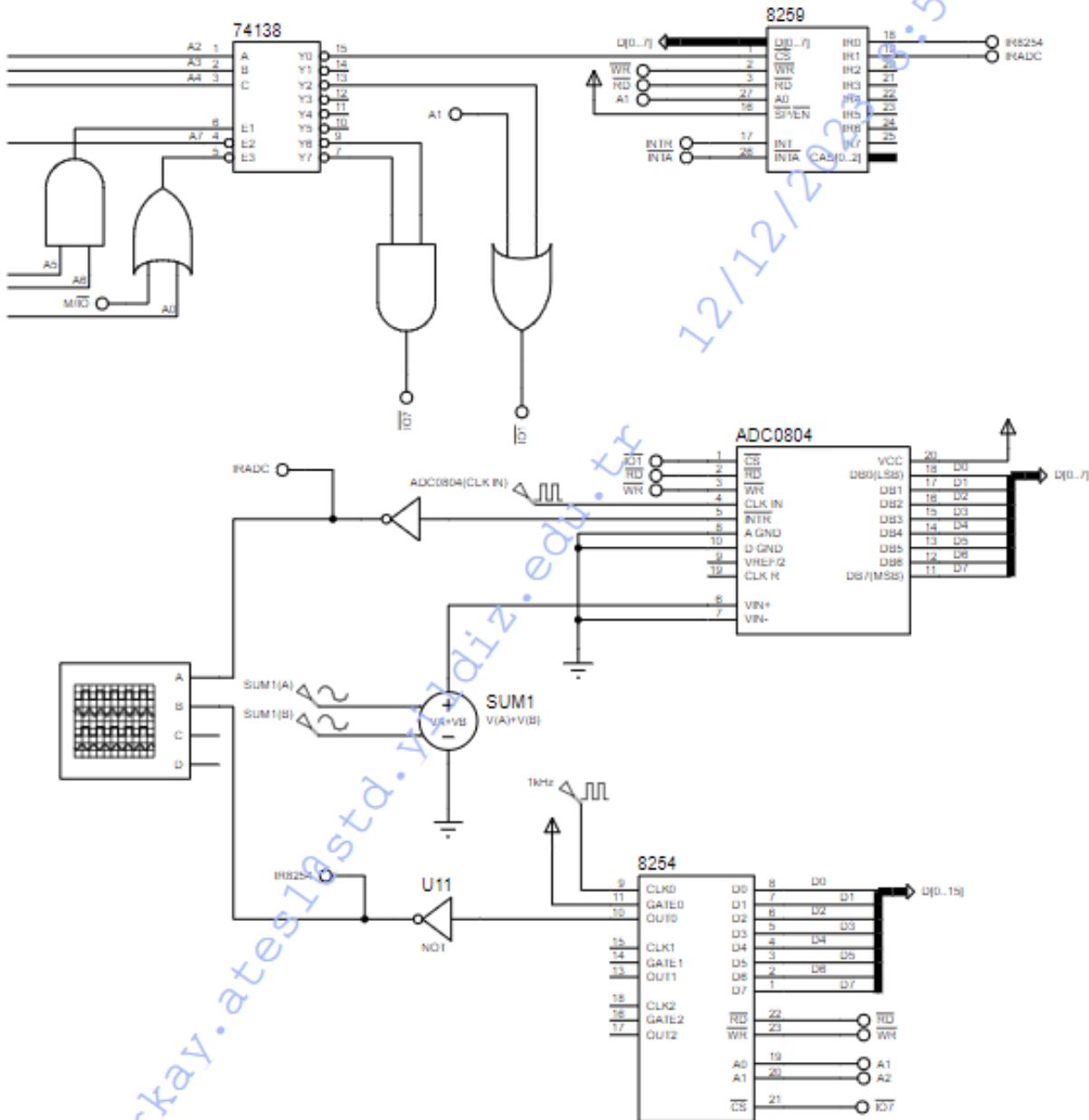
Örnek 22-6

8086 G/Ç uzayında 60H adresinden itibaren ardışık çift adreslere 8259, 68H adresine ADC0804 ve 78H adresinden itibaren çift adreslere ise 8254 yerleştirilmek isteniyor. Analog bir işaretin 200 Hz frekansta örneklenerek sayısallaştırılması istenmektedir. Örnekleme frekansının 8254 ile kesme temelli olarak hassas bir şekilde belirlenmesi, sayısal dönüşümün ise ADC0804 ile yine kesme temelli gerçekleştirilmesi istenmektedir.

8254 CNTR0 kullanarak örnekleme yapılacaktır. Bunun için OUT0 ucu NOT kapısından geçirilerek 8259 IR0 kesme istek ucuna bağlanmıştır. ADC0804 kullanılarak sayısallaştırma yapılacaktır. Bunu için ise ADC0804 INTR ucu NOT kapısından geçirilerek 8259 IR1 ucuna bağlanmıştır. Şekil 22-14 ile verilen devre şeması ve bağlantılar göre aşağıda istenenleri gerçekleyiniz:

- Adres çözümleme devresini 3×8 kodçözücü ve gerekli basit lojik kapılar kullanarak tasarlayınız.
- CNTR0'ı 200Hz frekansta tekrarlı olarak darbe üretecek şekilde ayarlayınız.
- Her OUT0 pulse geçişinde tetiklenecek bir kesme servis programı oluşturarak burada ADC dönüşümü başlatan assembly kodunu yazınız.
- Bir önceki maddede tanımlanan kesme servis programını 0A0H kesme vektör numarasına bağlayan assembly kodunu yazınız.

- e) Her ADC dönüşümü bittiğinde tetiklenecek bir kesme servis programı oluşturarak ADC sonucu olan sayısal değeri okuyan assembly kodunu yazınız.
- f) Bir önceki maddede tanımlanan kesme servis programını 0A1H kesme vektör numarasına bağlayan assembly kodunu yazınız.
- g) 8259'u uygun şekilde ayarlayan assembly kodunu yazınız.



Şekil 22-14: Örnek 22-6 için bağlantı yapısı

Şekil 22-14 ile Örnek 22-6 için bağlantı yapısı Verilmiştir.

```

;-----;
; PROGRAM : Örnek 22-6.ASM
; 8259 yardımcıyla analog işaretin 200Hz'de örnekleyen ve örnekleme
; frekansını hassas bir şekilde kesme tabanlı olarak 8254 kullanımlıyla
; gerçekleştiren assembly program
;-----;

        PAGE 60,80
        TITLE 8259, 8254, ADC0808 ile kesme tabanlı örnekleme
STAK     SEGMENT PARA STACK 'STACK'
        DW 20 DUP(?)
STAK     ENDS
CODE     SEGMENT PARA 'CODE'
        ASSUME CS:CODE, SS:STAK
;-----;
; 8254'un her çevriminde tetiklenen kesmeye ilişkin KSP
;-----;

NEWINT    PROC FAR
        OUT 68H, AL      ; boş yazma ile ADC dönüşümü başlatır
        IRET
NEWINT    ENDP
;-----;
; ADC dönüşümünün bitmesiyle tetiklenen kesmeye ilişkin KSP
;-----;

NEWINT2   PROC FAR
        IN AL, 68H       ; ADC dönüşüm sonucu okunur
        IRET
NEWINT2   ENDP
START    PROC FAR
;-----;
; 0A0H kesme vektör numarasının NEWINT'e bağlanması
;-----;

        XOR AX, AX
        MOV ES, AX
        MOV AL, 0A0H
        MOV AH, 4
        MUL AH
        MOV BX, AX
        LEA AX, NEWINT
        MOV WORD PTR ES:[BX], AX
        MOV AX, CS
        MOV WORD PTR ES:[BX+2], AX
;-----;
; 0A1H kesme vektör numarasının NEWINT2'ye bağlanması
;-----;

        MOV AL, 0A1H
        MOV AH, 4
        MUL AH
        MOV BX, AX
        LEA AX, NEWINT2
        MOV WORD PTR ES:[BX], AX
        MOV AX, CS
        MOV WORD PTR ES:[BX+2], AX
;-----;
; 8254'un koşullandırılması
;-----;

        MOV AL, 34H      ; 34H=0011 0100B, CNTR0, 16bit, kip 2, binary
        OUT 7EH, AL
        MOV AX, 5
        OUT 78H, AL
        MOV AL, AH
        OUT 78H, AL      ; CNTR0 sayma değeri 5, 1000/5=200Hz
;-----;
; 8259'un koşullandırılması
;-----;

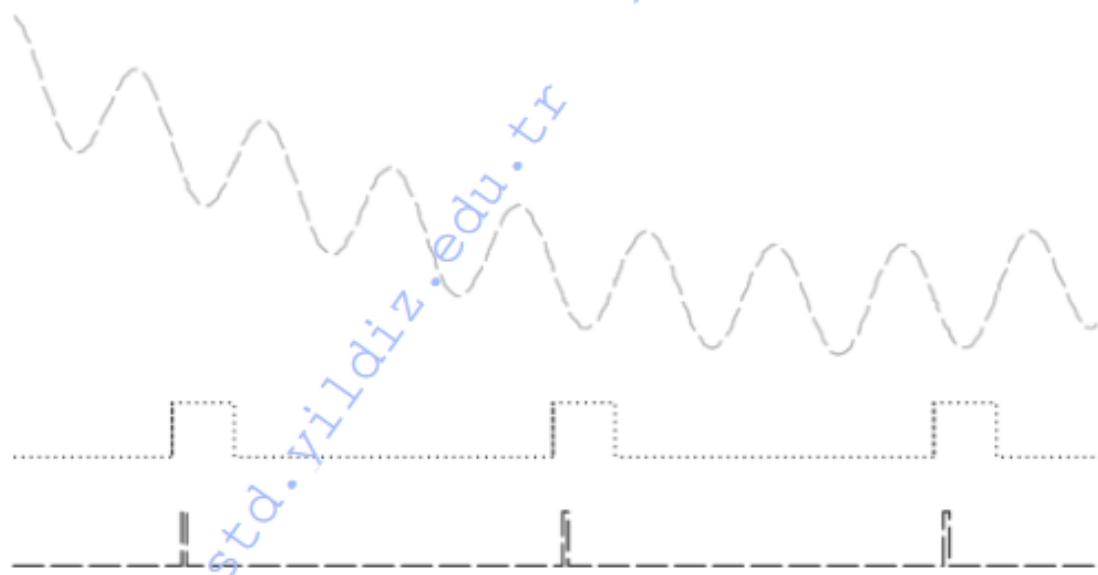
```

```

        MOV AL, 13H
        OUT 60H, AL
        MOV AL, 0A0H      ; IRO 0A0H tipinde kesme isteği tetikler
        OUT 62H, AL
        MOV AL, 03H
        OUT 62H, AL
        STI             ; IF<-1, INTR kesmeleri aktif
        CALL FAR PTR DELAY; arayüzlerin hazır olmasını bekle
ENDLESS:   JMP ENDLESS    ; sonsuz döngü
START      ENDP

; Bekleme sağlayan altyordam
-----
DELAY     PROC FAR
        PUSH CX
        MOV CX, 0F00H
L1:       LOOP L1
        POP CX
        RET
DELAY     ENDP
CODE      ENDS
END START

```



Şekil 22-15: Örnek 22-6 osiloskop çıktısı (yukarıdan aşağıya analog işaret, IR0, IR1 ucu)

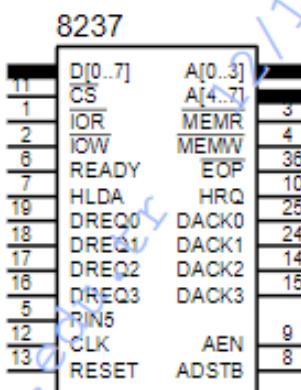
Şekil 22-15 ile Örnek 22-6 için osiloskop çıktısı verilmiştir. Burada IR0 ile kesme tetiklenmesinden (CNTR0 kaynaklı) hemen sonra IR1 kesmesinin (ADC dönüşüm bitmesi kaynaklı) tetiklendiği görülebilmektedir.

23. 8237 DMA (DIRECT MEMORY ACCESS)

Program kontrollü ve kesme temelli G/C haricinde veri aktarımı için üçüncü bir tip G/C formu olarak **doğrudan bellek erişimi (direct memory access - DMA)** verilebilir. DMA G/C ile mikroişlemci geçici olarak devre dışı tutularak doğrudan bellek erişimi sağlanır. Böylelikle G/C birimi ve bellek arasında yüksek hızda veri aktarımı sağlanabilir.

23.1 8237 Uç Tanımları

8257 DMA kontrolcüsüne ilişkin uç tanımları Şekil 23-1 verilmektedir. Burada her bir uç tanımından sonra ucun yönü de belirtilmektedir (I: giriş, O: çıkış):



Şekil 23-1: 8237 fonksiyonel uç tanımları

HLDA (Hold Acknowledge – Tutma Onayı): (I)

8086'daki HLDA ucunun (Şekil 23-1) karşılığıdır. Bu ucta lojik1 değeri 8086'nın system yollarının kontrolünü devrettiği anlamına gelir.

DACK0-DACK3 (DMA Acknowledge – DMA Onayı): (O)

Bir çevre birimine DMA için onay verildiğini belirtmek için kullanılır. Aktif 0 veya aktif 1 olarak programlanabilir. Reset halinde aktif 0 olarak ayarlanır.

DREQ0-DREQ3 (DMA Request – DMA İsteği): (I)

Çevre birimlerinin DMA servislerine ilişkin istek girişleridir. Birbirinden bağımsız arayümsuz kanallar olarak çalışırlar. DREQ0-3 uçlarından gönderilen istekler 8237 tarafından DACK0-3 uçları (Şekil 23-1) ile cevaplandırılır. Geçerli bir istek oluşturmak adına DACK0-3 uçları aktif olana kadar DREQ0-3 işaretini (Şekil 23-1) sağlanmalıdır. DREQ0-3 uçları aktif 1 veya aktif 0 olarak programlanabilmektedir. Reset ile DREQ0-3 aktif 1 olarak çalışmaktadır.

DB0-DB7 (Data Bus – Veri Yolu): (I/O)

Sistem veri yoluna bağlı çift yönlü uçlardır. Programlama durumunda 8237 yazmaçlarını okumak ve yazmaçlara değer yazmak için kullanılır. DMA çevrimlerinde ise ADSTB ucu (Şekil 23-1) yardımıyla harici bir mandalda, adres değerinin yüksek anlamlı byte’ını saklanabilmesi için adres değerlerini sağlar.

IOR (I/O Read - G/Ç Okuma): (I/O)

Çift yönlü, aktif 0 bir kontrol ucudur. DMA boşta bekleme çevrimlerinde mikroişlemci tarafından kontrol yazmaçlarını okumak için kullanılır. DMA aktif çevrimlerinde ise 8237’nin çevre birimlerinden veri okuma kontrol ucu olarak kullanılır.

IOW (I/O Write – G/Ç Yazma): (I/O)

Çift yönlü, aktif 0 bir kontrol ucudur. DMA boşta bekleme çevrimlerinde mikroişlemci tarafından kontrol yazmaçlarına veri yazmak için kullanılır. DMA aktif çevrimlerinde ise 8237’nin çevre birimlerine veri yazma kontrol ucu olarak kullanılır.

EOP (End of Process – İşlem Sonu): (I/O)

Çift yönlü, aktif 0 bir kontrol ucudur. DMA servisinin sonlandırılmasında kullanılır.

A0-A3 (Address - Adres): (I/O)

Çift yönlü uçlardır. DMA boşta bekleme çevrimlerinde mikroişlemcinin 8237 yazmaçlarını adreslemesi için kullanılır. DMA aktif çevrimlerde ise adres çıkış ucu olarak kullanılır.

A4-A7 (Address - Adres): (O)

DMA çevrimlerinde aktif olan adres çıkış uçlarıdır.

HRQ (Hold Request – Tutma İsteği): (O)

Mikroişlemciden sistem yollarının kontrolünü devretmesi için istekte bulunmak amacıyla kullanılır.

AEN (Address Enable – Adres Aktif): (O)

Adres yüksek anlamlı byte değerinin sistem adres yoluna çıkartılmasını sağlar.

ADSTB (Address Strobe – Adres Tut): (O)

Aktif 1 bir uctur. Adres yüksek anlamlı byte değeri için tutma işaretidir.

MEMR (Memory Read): (I/O)

Aktif 0 çıkış ucudur. DMA okuma veya bellekten belleğe işlemlerde okuma sinyali olarak kullanılır.

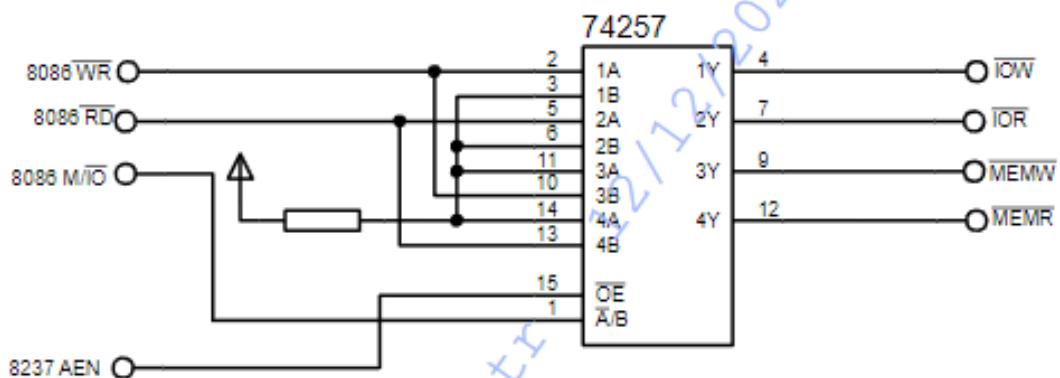
MEMW (Memory Write): (O)

Aktif 0 çıkış ucudur. DMA yazma veya bellekten belleğe işlemlerde yazma sinyali olarak kullanılır.

PIN5: (I)

İşlem sırasında lojik 1'de tutulması gereklidir.

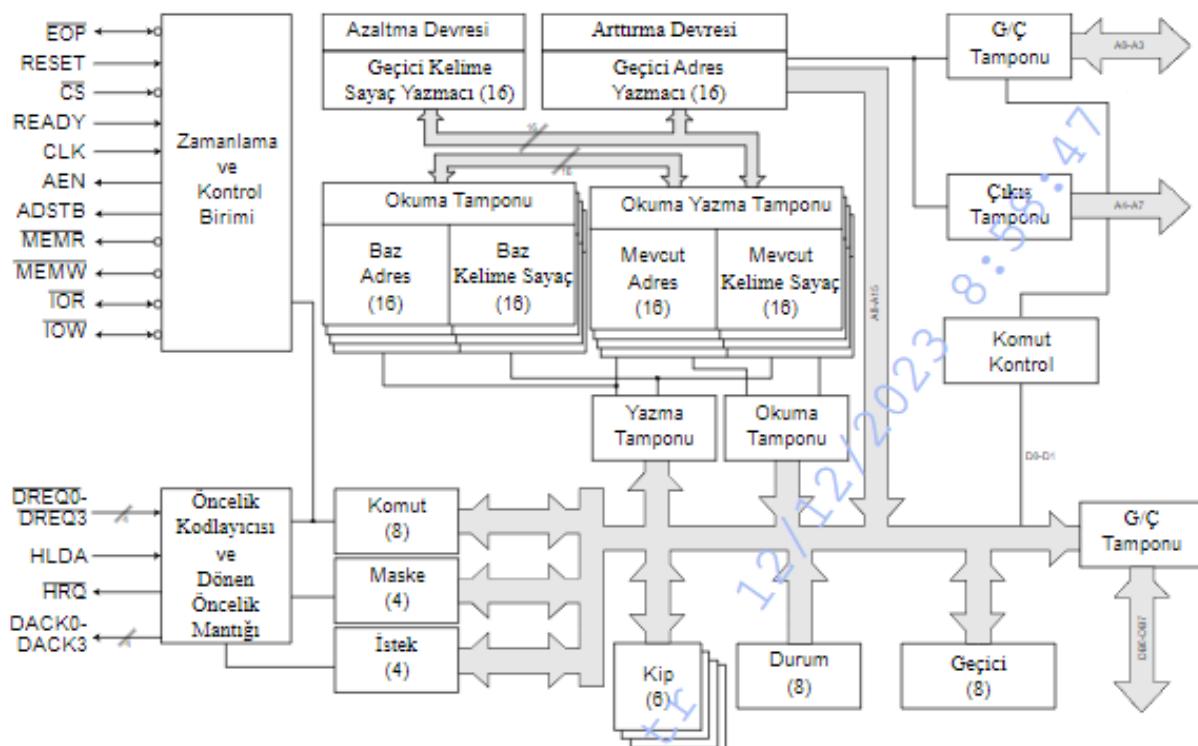
8237'nin ihtiyaç duyacağı IOR, IOW, MEMR ve MEMW sinyallerinin oluşturulması Şekil 23-2 ile verilen devre şeması ile gerçekleştirilebilir.



Şekil 23-2: 8237 okuma yazma kontrol sinyallerinin üretilmesi

23.2 8237 İç Yapısı

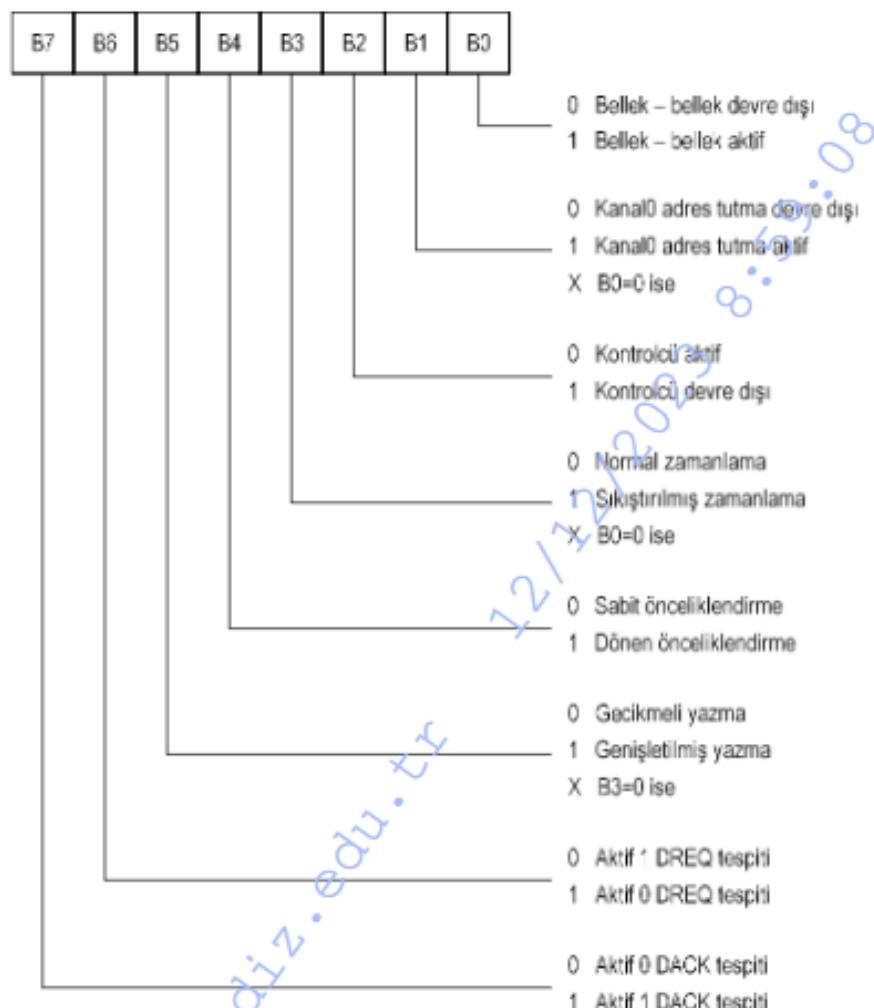
8237 iç yapısı Şekil 23-3 ile verilmiştir. 8237'nin mikroişlemci ile arayüzü sağlayan fonksiyonel bloklar yanında, G/C ve bellek birimleri ile arayüzünü sağlayan fonksiyonel bloklar da bulunmaktadır. 8237 dört kanal DMA veri transferini desteklemektedir.



Şekil 23-3: 8237 iç yapısı

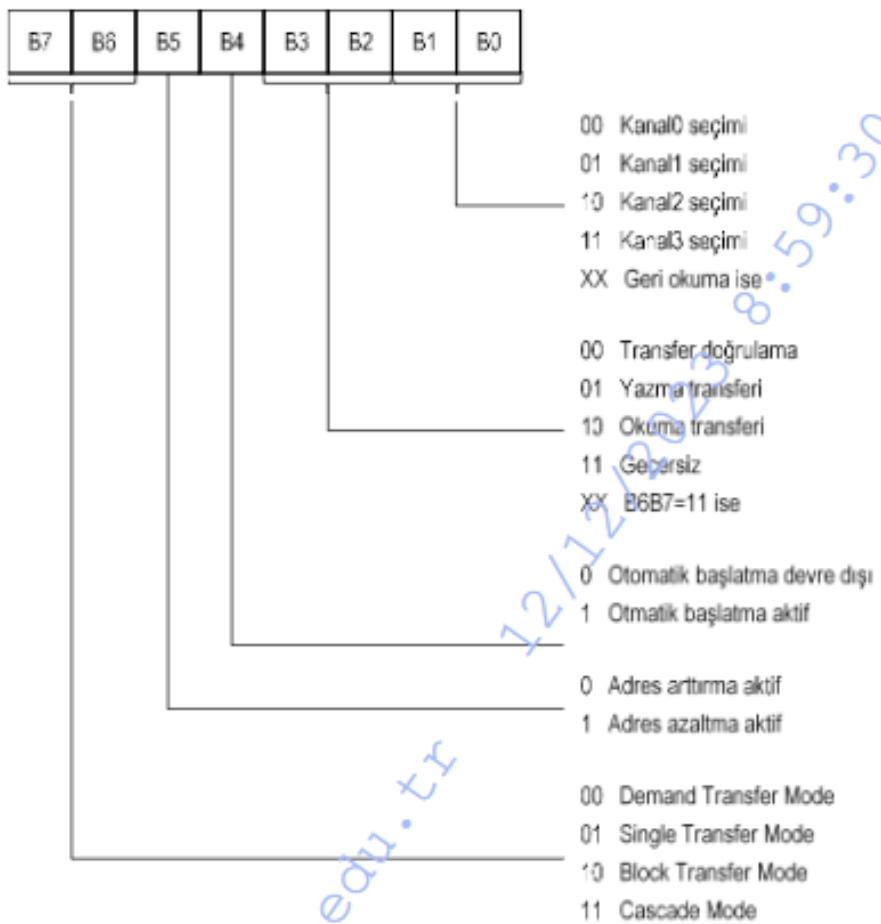
23.3 8237 Yazmaçları

Mevcut adres yazmacı (current address register - CAR) 16 bit'lik DMA transfer adres değerini tutar. Mevcut kelime sayıç yazmacı (current word count register - CWCR) DMA transferinde aktarılacak byte adedini tutar. Baz adres yazmacı (base address register - BAR) ve baz kelime sayıç yazmacı (base word count register - BWCR) otomatik başlatma kipinde CAR ve CWCR yazmacılarına yüklenecek değerleri tutar. 8237 komut yazmacının bit'lerine göre görevleri Şekil 23-4 ile verilmiştir.

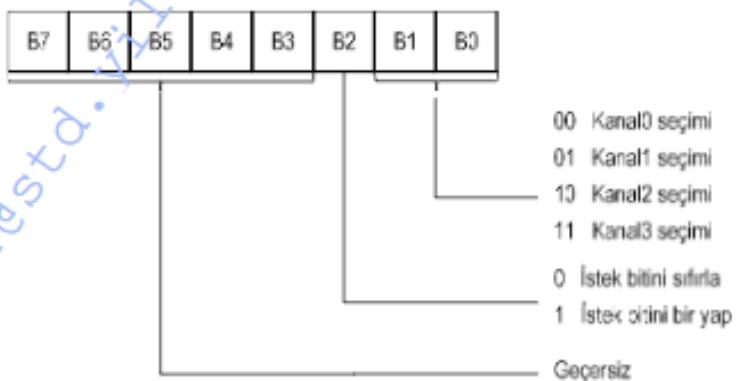


Şekil 23-4: 8237 kontrol yazmacı

8237 kip yazmacının (Şekil 23-5), istek yazmacının (Şekil 23-6), maske yazmacının (Şekil 23-7) ve durum yazmacının (Şekil 23-8) bit'lerine göre fonksiyonları verilmiştir. 8237 yazıcılarının adet ve genişlikleri Tablo 23-1 ile verilmiştir.



Şekil 23-5: 8237 kip yazmacı

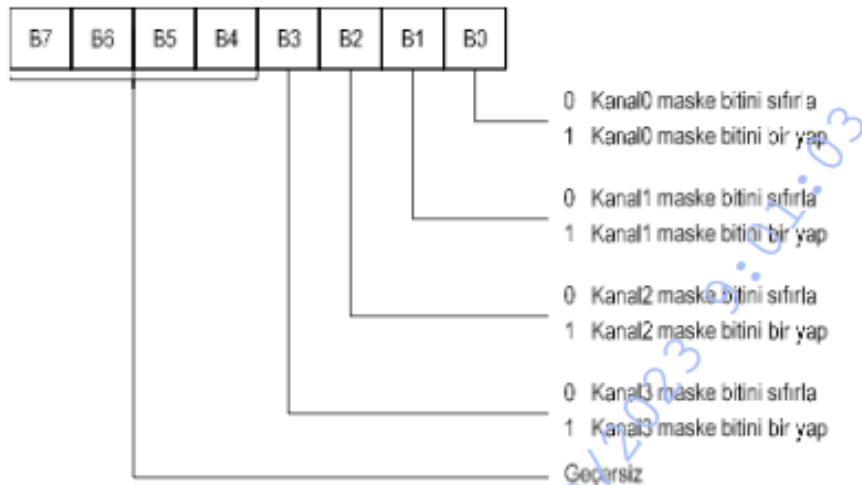


Şekil 23-6: 8237 istek yazmacı

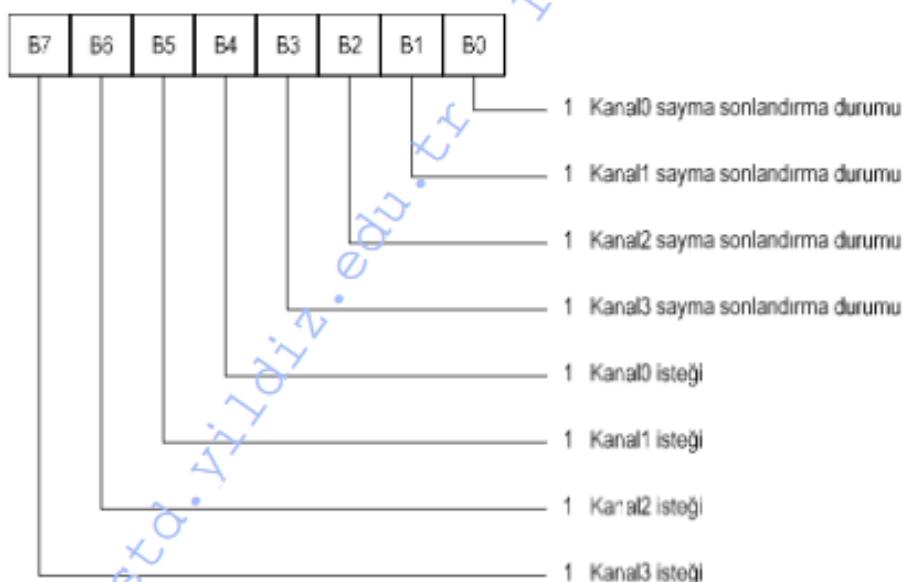
berkay.ates1@sta.edu.tr

8:59:30

12/12/2012



Şekil 23-7: 8237 maske yazmacı



Şekil 23-8: 8237 durum yazmacı

Tablo 23-1: 8237 yazmaçları ve özellikleri

Yazmaç Tipi	Yazmaç Genişliği	Yazmaç Adedi
Baz Adres Yazmacı	16 bit	4 adet
Baz Kelime Sayaç Yazmacı	16 bit	4 adet
Mevcut Adres Yazmacı	16 bit	4 adet
Mevcut Kelime Sayaç Yazmacı	16 bit	4 adet
Geçici Adres Yazmacı	16 bit	1 adet
Geçici Kelime Sayaç Yazmacı	16 bit	1 adet
Durum Yazmacı	8 bit	1 adet
Komut Yazmacı	8 bit	1 adet
Geçici Yazmaç	8 bit	1 adet
Kip Yazmacı	6 bit	4 adet
Maske Yazmacı	4 bit	1 adet
İstek Yazmacı	4 bit	1 adet

8237 kontrol yazmaçlarının 8086 tarafından programlanmak adına adreslenebilmesi için uçlarında oluşturulması gereklilik değerler Tablo 23-2 ile verilmektedir. Mikroişlemci tarafından 8237'ye yapılan erişimlerin bir kısmı yazılım komutları olarak kullanılır. İlk/son iki duraklısının (first/last flip flop) ayarlanmasına ilişkin olarak sıfırlama (clear first/last FF) ve bir yapma (set first/last FF) komutları mevcuttur. Bu iki yazılım komutu kullanılarak iki duraklı bilinen bir lojik seviyeye çekilerek adres ve kelime adedi yazmaçları hangi anlam sırasında erişileceği belirlenmiş olur. Reset komutuyla; komut, durum, istek, geçici yazmaçları ve iki duraklı sıfırlanır, maske yazmacının bit'leri bir yapılır. Maske yazmacının bit'lerini sıfırlamak için ayrı bir yazılım komutu mevcuttur.

Tablo 23-2: 8086 tarafından 8237 kontrol yazmaçlarının adreslenmesi

İşlem	A3	A2	A1	A0	TOR	TOW
Durum yazmacını okuma	1	0	0	0	0	1
Komut yazmacına yazma	1	0	0	0	1	0
İstek yazmacını okuma	1	0	0	1	0	1
İstek yazmacına yazma	1	0	0	1	1	0
Komut yazmacını okuma	1	0	1	0	0	1
Tek maske bit'i yazma	1	0	1	0	1	0
Kip yazmacını okuma	1	0	1	1	0	1
Kip yazmacına yazma	1	0	1	1	1	0
İlk/son iki duraklıyı bir yap (Set first/last FF)	1	1	0	0	0	1
İlk/son iki duraklıyı sıfırla (Clear first/last FF)	1	1	0	0	1	0
Geçici yazmacı okuma	1	1	0	1	0	1
Reset (Master clear)	1	1	0	1	1	0
Kip yazmaç sayacını sıfırla (Clear mode reg. counter)	1	1	1	0	0	1
Maske yazmacını temizleme	1	1	1	0	1	0
Maske bit'lerini okuma	1	1	1	1	0	1
Maske bit'lerine yazma	1	1	1	1	1	0

8237 için kanal G/Ç port adresleri Tablo 23-3 ile verilmiştir.

Tablo 23-3: 8237 kanal G/Ç adresleri

Kanal	Yazmaç	İşlem	IOR	IOW	A3	A2	A1	A0	İki Duraklı	DB0-DB7
0	Baz ve mevcut adres yazmacı	Yazma	1	0	0	0	0	0	0	A0-A7
		Yazma	1	0	0	0	0	0	1	A8-A15
	Mevcut adres yazmacı	Okuma	0	1	0	0	0	0	0	A0-A7
		Okuma	0	1	0	0	0	0	1	A8-A15
	Baz ve mevcut kelime sayaç yazmacı	Yazma	1	0	0	0	0	1	0	W0-W7
		Yazma	1	0	0	0	0	1	1	W8-W15
	Mevcut kelime sayaç yazmacı	Okuma	0	1	0	0	0	1	0	W0-W7
		Okuma	0	1	0	0	0	1	1	W8-W15
1	Baz ve mevcut adres yazmacı	Yazma	1	0	0	0	1	0	0	A0-A7
		Yazma	1	0	0	0	0	1	0	A8-A15
	Mevcut adres yazmacı	Okuma	0	1	0	0	1	0	0	A0-A7
		Okuma	0	1	0	0	0	1	1	A8-A15
	Baz ve mevcut kelime sayaç yazmacı	Yazma	1	0	0	0	0	1	0	W0-W7
		Yazma	1	0	0	0	0	1	1	W8-W15
	Mevcut kelime sayaç yazmacı	Okuma	0	1	0	0	1	1	0	W0-W7
		Okuma	0	1	0	0	1	1	1	W8-W15
2	Baz ve mevcut adres yazmacı	Yazma	1	0	0	1	0	0	0	A0-A7
		Yazma	1	0	0	0	1	0	0	A8-A15
	Mevcut adres yazmacı	Okuma	0	1	0	1	0	0	0	A0-A7
		Okuma	0	1	0	0	1	0	1	A8-A15
	Baz ve mevcut kelime sayaç yazmacı	Yazma	1	0	0	0	1	0	1	0
		Yazma	1	0	0	0	1	0	1	W8-W15
	Mevcut kelime sayaç yazmacı	Okuma	0	1	0	1	0	1	0	W0-W7
		Okuma	0	1	0	1	0	1	1	W8-W15
3	Baz ve mevcut adres yazmacı	Yazma	1	0	0	1	1	0	0	A0-A7
		Yazma	1	0	0	1	1	0	0	A8-A15
	Mevcut adres yazmacı	Okuma	0	1	0	1	1	0	0	A0-A7
		Okuma	0	1	0	1	1	0	1	A8-A15
	Baz ve mevcut kelime sayaç yazmacı	Yazma	1	0	0	1	1	1	0	W0-W7
		Yazma	1	0	0	1	1	1	1	W8-W15
	Mevcut kelime sayaç yazmacı	Okuma	0	1	0	1	1	1	0	W0-W7
		Okuma	0	1	0	1	1	1	1	W8-W15

23.4 8237 Veri Transfer Kipleri

8237 herhangi bir DMA isteği olmadığımda boşta bekleme çevrimindedir. Bir kanal dan DMA isteği geldiğinde 8237 mikroişlemciye HRQ ucu üzerinden tutma isteği gönderir. Mikroişlemci tutma isteğini onaylarsa bunu HLDA ucu üzerinden 8237'ye iletir ve 8237 aktif çevrime geçer. Bu çevrimde aşağıda verilen kiplerin birinde veri transferi gerçekleşir.

23.4.1 Tek Aktarım Kipi (Single Transfer Mode)

Her DMA isteği bir byte veri aktarımı ile sonuçlanır. Her istek için DACK sağlanana kadar DREQ ucunun aktif tutulması gerekmektedir. Her bir aktarımda kelime sayacı bir azaltılarak, adres değeri ise programlamaya göre bir arttırılır veya azaltılır.

23.4.2 Blok Aktarım Kipi (Block Transfer Mode)

DREQ ile veri transferi başlatılarak kelime sayacı sıfırlanana kadar devam eder. DACK sağlanana kadar DREQ ucunun aktif tutulması gerekmektedir.

23.4.3 Talep Aktarım Kipi (Demand Transfer Mode)

G/C biriminin yetişemediği durumda DREQ sinyalinin kaldırılması ile veri transferine ara verilebilir. Veri transferinin devam edebilmesi DREQ sinyalinin yeniden aktif edilmesi ile sağlanabilir.

23.4.4 Kaskat Kip (Cascade Mode)

8237'nin DREQ ve DACK kanallarına dört adet yardımcı 8237'nin HRQ ve HLDA uçları üzerinden bağlanması ile kaskat kip oluşturulur.

23.4.5 Bellek - Bellek Aktarımı (Memory to Memory Transfer)

Bir bellek alanından başka bir bellek alanına blok veri transferi için bellek – bellek transfer kipi kullanılır.

23.5 8237 Programlama

8237'nin DMA veri transferi için ayarlanmasıında aşağıdaki adımların yürütülmesi gerekmektedir:

- Adres ve sayıç yazmaçlarının ayarlanması
 - İlk son iki duraklıyı sıfırla
 - Kanalı devre dışı bırak
 - Adres LSB değerini gönder
 - Adres MSB değerini gönder
 - Kelime sayacı LSB gönder
 - Kelime sayacı MSB gönder
- Transfer kipini belirle
- Kanalı aktif yap

Örnek 23-1

80H-9EH aralığındaki çift adreslere yerleştirilmiş 8237 kullanarak 10000H-13FFFH adres aralığından 140000H-17FFFFH adres aralığına bellek – bellek DMA

veri aktarımını sağlayan assembly kodunu yazınız. Bellek – bellek DMA veri aktarımında kaynak için kanal0, hedef için kanal1 kullanılmasını sağlayınız.

```

; PROGRAM : Örnek 23-1.ASM
; 8237 kullanarak bellek - bellek DMA veri transferi yapan
; assembly program

PAGE 60,80
TITLE bellek - bellek DMA veri transferi

CLEAR_FF EQU 98H ;F/L CLEAR VALUE      1001 1000B
CH0_A    EQU 80H ;CHANNEL 0 ADDRESS   1000 0000B
CH1_A    EQU 84H ;CHANNEL 1 ADDRESS   1000 0100B
CH1_C    EQU 86H ;CHANNEL 1 COUNT    1000 0110B
MODE     EQU 96H ;MODE               1001 0110B
CR       EQU 90H ;COMMAND REGISTER  1001 0000B
MASKS   EQU 9EH ;MASKS              1001 1110B
REQ      EQU 92H ;REQUEST REGISTER 1001 0010B
STATUS  EQU 90H ;STATUS REGISTER  1001 0000B

STAK     SEGMENT PARA STACK 'STACK'
DW 20 DUP(?)
STAK     ENDS
DATA     SEGMENT PARA 'DATA'
DW 1000H
DMA_SEG  DW 0000H
DMA_SRC  DW 4000H
DMA_DEST DW 4000H
DMA_CNT  DW 4000H
DATA     ENDS
CODE     SEGMENT PARA 'CODE'
ASSUME CS:CODE, SS:STAK
START   PROC FAR

; Dönüş için gerekli olan değerler yığında saklamıyor
; PUSH DS
; XOR AX, AX
; PUSH AX

; DATA ismiyle tanımlı kesim alanına erişebilmek için gerekli tanımlar
; MOV AX, DATA
; MOV DS, AX
MOV ES, DMA_SEG ; DMA kaynak segment değeri ES'de
MOV SI, DMA_SRC ; DMA kaynak işaretçi değeri SI'da
MOV DI, DMA_DEST ; DMA hedef işaretçi değeri DI'da
MOV CL, 4
MOV AL, 0
OUT CLEAR_FF, AL ; İlk/son iki duraklıyı sıfırla
MOV AX, ES ; DMA kaynak segment değeri
SHL AX, CL ; AX <- AX*16
ADD AX, SI ; AX <- ES*16+SI, kaynak adresi
OUT CH0_A, AL ; Kanal0 adres LSB
MOV AL, AH
OUT CH0_A, AL ; Kanal0 adres MSB
MOV AX, ES ; DMA hedef segment değeri
SHL AX, CL ; AX <- AX*16
ADD AX, DI ; AX <- ES*16+DI, hedef adresi
OUT CH1_A, AL ; Kanal1 adres LSB
MOV AL, AH

```

```
        OUT CH1_A, AL      ; Kanalı adres MSB
        MOV AX, DMA_CNT    ; Kelime adedi
        DEC AX             ; Kelime adedini bir azalt
        OUT CH1_C, AL      ; Kanalı kelime sayaç LSB
        MOV AL, AH
        OUT CH1_C, AL      ; Kanalı kelime sayaç MSB
;
; Kip yazmacı : Kanalo, blok kip, adres artımlı, okuma yönü
;
        MOV AL, 88H
        OUT MODE, AL
        MOV AL,1            ; Kontrol yazmacı : bellek-bellek transfer
        OUT CR, AL
        MOV AL, 0EH          ; Maske yazmacı : kanalo maskesini aç
        OUT MASKS, AL
;
; İstek yazmacı : kanalo isteği set, DMA transferi başlat
;
        MOV AL, 4
        OUT REQ, AL
        RETF
START  ENDP
CODE   ENDS
END START
```