



İSTANBUL TİCARET ÜNİVERSİTESİ  
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ  
MİKROİŞLEMCİLİ SİSTEM LABORATUVARI



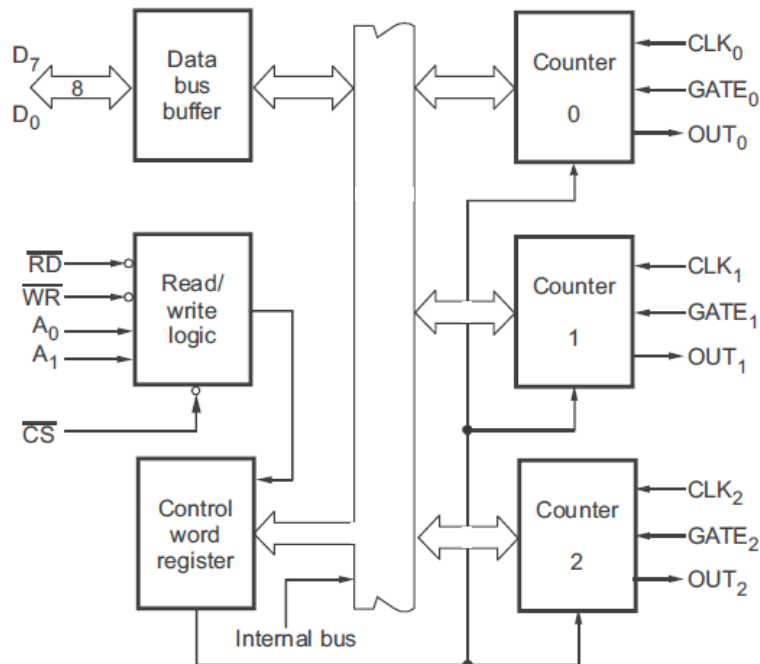
## OLAYLARI ZAMANLAMA

İnsanların işlerini bir takvime ve zamana bağlı olarak yürütmesine benzer şekilde, bilgisayarlar da işlerini bir takvim ve zaman içinde yapar. Örneğin bir program koşulurken klavyeden yapılacak müdahaleleri anlayabilmek için belirli aralıklarla klavyenin yoklanması bir zamanlama işidir. Veya CMOS'ta üretilen gerçek-zamanlı saatten bilgisayarın sistem saatinin hangi aralıklarla güncelleneceği de yine bir zamanlama problemidir. Eski bilgisayarlarda dinamik belleklerin tazelenmesi (refresh) bu zamanlamaya verilecek başka bir örnektir.

Zamanlama işlemi donanım veya yazılımla yapılabilir. Yazılım yöntemi gecikme programı kullanımına dayanır ve bu yüzden CPU saat frekansının bir fonksiyonudur. Yani aynı gecikme programı farklı bilgisayarlarda farklı zamanlamalar doğurur. Bu farklılıkları ortadan kaldırmak ve bilgisayarın donanımından bağımsız zamanlamalar gerçeklemek için bilgisayarlarda özel donanımlar inşa edilmiştir. Bu özel donanımların başında da 1.193180 MHz'lik işaretin ürettiği saat devresi ve bu saatten farklı biçimlere sahip dikdörtgen dalga üretmeyi sağlayan 8254PIT Programlanabilen Aralık Zamanlayıcı (Programmable Interval Timer) gelir. Bu deneyin amacı 8254 PIT yongasını tanıtmak ve onu programlayarak çeşitli uygulamalar gerçeklemektir.

### 8254 PIT'in YAPISI ve ÇALIŞMASI

8254 PIT, CPU'dan bağımsız olarak, bir dış saat işaretinden programlanabilen zaman aralıkları üretir. Altı farklı çalışma moduna sahip olan PIT'in iç yapısı Şekil 1'de gösterilmiştir.

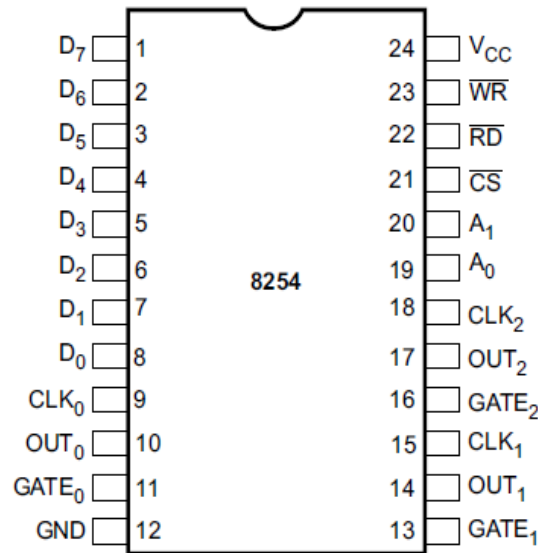


Şekil 1. 8254 PIT'in iç yapısını gösteren blok diyagram

PIT her biri ayrı programlanabilen 16-bit genişliğinde üç adet (Counter 0-2) sayıcılarına sahiptir. Her sayıcı zaman tabanı (time base) olarak hizmet veren kendi saat işaretine (CLK0-CLK2) sahiptir. Saymayı tetiklemek veya başlatmak için GATE0-GATE2 kapı işaretlerinden yararlanır. Sayma moduna bağlı olarak alçak-yüksek veya yüksek-alçak geçişlerinde sayıcı aktif yapılır ve OUT0-OUT2 çıkışlarında modun gerektirdiği biçimde bir işaret üretilir. Her bir sayıcı iki tane giriş (**CLOCK** ve **GATE**) ve bir tane çıkış (**OUT**) sinyaline sahiptir.

#### 8254 PIT'in Özellikleri:

- 3 tane birbirinden bağımsız 16 bitlik aşağı sayıcı
- Çalışma frekansı 0-10 MHz (8253'ün çalışma frekansı 0-2.6 MHz) arasındadır.
- Özdeş 3 sayıcı da Binary veya BCD olarak sayacak şekilde programlanabilmektedir.
- Bütün Intel işlemciler ve diğer işlemcilerin pek çoğu ile uyumludur.
- 8254 önemli bir komut olan READ-BACK işlemini destekler. Bu komut; sayıcıdaki değeri ve sayıcının aktif olan modunu öğrenmeye imkan tanır.

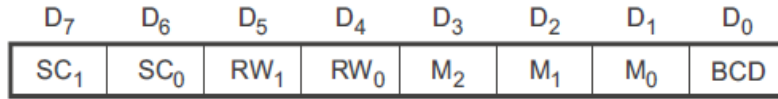


Şekil 2. 8254'ün bağlantı uçlarının isimleri

#### 8254 PIT'in PROGRAMLANMASI

Programlamak için ilk olarak kontrol kelimesi kontrol kaydediciye yazılır, ve daha sonra bir veya iki veri baytı amaçlanan sayıcının portuna yazılır. Kontrol kaydedici bir kere yüklendikten sonra kontrol kaydediciye erişmeden sayıcılara başka değerler yazılabilir. 8253 PIT'in kontrol kaydedicisi sadece yazılan, 8254'ün kaydedicisi ise hem yazılan hem de okunan bir kaydedicidir. A<sub>1</sub> ve A<sub>0</sub> bitlerinin durumları (yani 00,01 veya 10) sırasıyla Counter #0, Counter #1, veya Counter #2 sayıcılarını seçer. A<sub>1</sub>A<sub>0</sub> =11 olduğunda kontrol word register seçilmiş olur.

(CS=0, RD=1)			
WR	A1	A0	Write target
0	0	0	Counter #0
0	0	1	Counter #1
0	1	0	Counter #2
0	1	1	Control word register



**SC - Select counter**

SC<sub>1</sub> SC<sub>0</sub>

0	0	Select counter 0
0	1	Select counter 1
1	0	Select counter 2
1	1	Illegal for 8253 Read -Back command for 8254 (See Read operations)

**M - Mode**

M<sub>2</sub> M<sub>1</sub> M<sub>0</sub>

0	0	0	Mode 0
0	0	1	Mode 1
x	1	0	Mode 2
x	1	1	Mode 3
1	0	0	Mode 4
1	0	1	Mode 5

**RW - Read /Write**

RW<sub>1</sub> RW<sub>0</sub>

0	0	Counter latch command (See Read operations)
0	1	Read / Write least significant byte only
1	0	Read / Write most significant byte only
1	1	Read / write least significant byte first, then most significant byte

**BCD :**

0	Binary counter 16 - bits
1	Binary coded decimal (BCD) Counter (4 Decades)

**Şekil 3:** 8253 Kontrol kelimesinin formatı

PIT'i programlama örneği:

## Example - 1

- Write code to set up three counters located at I/O address 40h as follows:
  - Counter 0: Binary counter operating in mode 0 with a value of 1234h
  - Counter 1: BCD counter operating in mode 2 with a value of 0100h
  - Counter 2: Binary counter operating in mode 4 with a value of 1FFFh

## Example – 1, cont'd

Mode word for counter 0 = 00 11 000 0 = 30h

Mode word for counter 1 = 01 11 010 1 = 75h

Mode word for counter 2 = 10 11 100 0 = B8h

```
MOV     AL,30H           ;SET UP COUNTER 0 MODE
OUT     43H,AL
MOV     AL,75H           ;SET UP COUNTER 1 MODE
OUT     43H,AL
MOV     AL,0B8H          ;SET UP COUNTER 2 MODE
OUT     43H,AL
MOV     AL,34H           ;LOAD COUNTER 0
OUT     40H,AL
MOV     AL,12H
OUT     40H,AL
MOV     AL,00H           ;LOAD COUNTER 1
OUT     41H,AL
MOV     AL,01H
OUT     41H,AL
MOV     AL,0FFH          ;LOAD COUNTER 2
OUT     42H,AL
MOV     AL,1FH
OUT     42H,AL
```

## SAYMA MODLARI

8254 PIT altı farklı sayma moduna sahiptir ve hem ikili hem de onlu modda sayabilir. Çeşitli çalışma modları aşağıdaki şekillerde gösterilmiştir. Tüm modlarda PIT ilk sayma değerinden başlayarak aşağı değerlere kadar sayar. Yeni sayma değerleri herhangi bir anda sayıcılara yazılabilir. Sayıcı 0 değerine eriştiği zaman 0,1,4, 5 periyodik olmayan modlarda sayıcı saymayı durdurmaz, ama FFFFh (BCD=0) veya 9999 (BCD=1) ile devam eder.

**Mod 0:** Nihai değerde kesme üretir. (*Interrupt on terminal count*)

**Mod 1:** Programlanabilen monoflop (*Gate Retriggerable One-shot*)

**Mod 2:** Oran üretici (*Rate generator*)

**Mod 3:** Kare-dalga üretici (*Square Wave generator*)

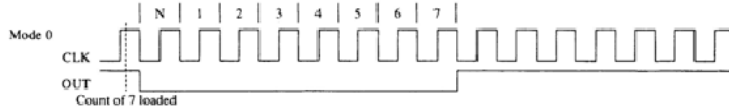
**Mod 4:** Yazılım tetiklemeli darbe (*Software Triggered Strobe*)

**Mod 5:** Donanım tetiklemeli darbe (*Hardware Triggered Strobe*)

## 8254 Modes of Operation

- **MODE 0: Events counter**

- Output becomes logic 0 when control word is written and remains there until N plus the number of programmed counts.
- Nice for generating termination interrupts



Kontrol kelimesi yazıldığında çıkış *lojik-0* olur ve sayıcı registerine değer yüklendikten sonra da çıkış düşük seviyede kalmaya devam eder ve ardından sayıcı saymaya başlar. Sayıciya yüklenen deęer+1 saat darbesi süresince çıkış, *lojik-0* olarak kalır. Sayıcıdaki deęer sıfıra ulaştığında çıkış *lojik-1* olur ve sayıcıya tekrar deęer yüklenene kadar çıkış yüksek seviyede kalır. (Gate=1 ise sayma devam eder, Gate=0 ise sayma durur.)

**Şekil 4.** Mod 0 zamanlama diyagramı

### 5.1 Mode 0: Interrupt at the End of Count

In this mode, the OUT output changes from low to high level when the end of the specified count is reached.

**Table 5-1 Mode 0 Operation**

Initial OUT		Low level
GATE input	High level	Count enable
	Low level	Count disable
Count write		The OUT pin becomes low level (independent of the CLK pulse). In 2-byte read/write mode, count is disabled when the first byte is written, and then the OUT pin becomes low level
Count transfer and operation		<p>When the count is written when GATE is high level:</p> <p>Transfer is performed at the first CLK pulse after the count data is written.</p> <p>The decrement operation of the down counter begins from the first CLK pulse after data transfer. If a count of N is set, the signal at the OUT pin will become low level after N+1 CLK pulses.</p> <p>When the count is written when GATE is low level:</p> <p>Transfer is performed at the first CLK pulse after the count is written.</p> <p>Decrement begins from the first CLK pulse after the GATE signal becomes high level.</p> <p>If a count of N is set, OUT will be low level for a period of N CLK pulses.</p>
Count ZERO		The signal at the OUT pin becomes high level. The count operation itself does not stop and counts down to FFFFH (binary)/9999 (BCD).
Minimum count		1

#### Mode 0 Program Example

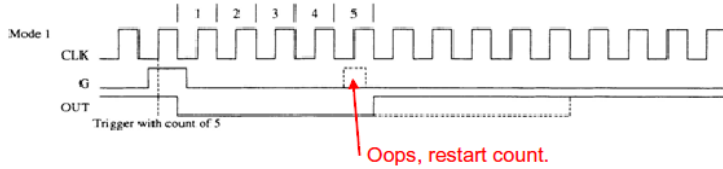
This is a subroutine for a delay of 10000 (2710H) CLK pulses. In this program, counter #2 is set to low/high 2-byte read/write mode and binary count.

```

SUBRO:  MOV    AL, 10110000B } Mode setting
        OUT    PCTRL, AL    }
                                     o Counter #2
                                     o High/low 2-byte read/write mode
                                     o Count mode 0
                                     o Binary
        MOV    AL, 10H      } Write count 10000
        OUT    PCNT2, AL
        MOV    AL, 27H
        OUT    PCNT2, AL
        RET
    
```

## 8254 Modes of Operation

- MODE 1: One-shot
  - G triggers the counter.
  - A pulse appears on OUT pin that remains logic 0 for the duration of the count.



Gate giriři, ıkıřı tetiklemek iin kullanılır. ıkıř bařlangıta yksek (lojik-1) seviyededir. Gate giriřinden uygulanan darbenin ykselen kenarını takiben gelen CLK darbesinde ıkıř alak seviyeye (lojik-0) geer. ıkıř, sayıcıdaki deęer sıfıra ulařtıęında yksek seviyeye geer ve Gate giriřinden uygulanacak bir sonraki darbenin ykselen kenarına kadar bu seviyede kalır.

řekil 5. Mod 1 zamanlama diyagramı

### 5.2 Mode 1: GATE Retriggerable One-Shot

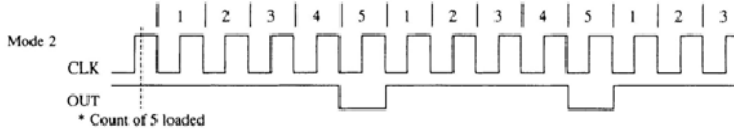
In this mode a low level one-shot pulse (triggered by the GATE input) is output from the OUT pin.

Table 5-2 Mode 1 operation

Initial OUT		High level
GATE	Trigger*	The count data is transferred at the first CLK pulse after the trigger.
Count write		Write count without affecting the current operation
Count transfer and operation		Transfer is performed at the first CLK pulse after the trigger. At the same time, the signal at the OUT pin becomes low level to start a one-shot pulse operation. The count is decremented from the next CLK pulse. If a count of N is set, the one-shot output from the OUT pin will continue for N CLK pulses.
Count ZERO		The signal at the OUT pin becomes high level. The count operation does not stop and counts down to FFFFH (binary)/9999 (BCD).
Minimum count		1

## 8254 Modes of Operation

- MODE 2: Programmable clock generator
  - Produces a pulse, one clock width wide, that is spaced based on the count.
  - Continues until G goes low or a new mode is programmed in.



Bu mod, N'e bölen sayıcı gibi çalışır. Çıkış başlangıçta yüksek seviyededir. Sayıcıya N değeri yüklenmiş olsun. Sayıcıya yüklenen değer sıfıra ulaşmadan önceki bir saat periyodu boyunca çıkış alçak seviyeye gider. Sonra çıkış yüksek seviyeye geçer ve sayıcıya başlangıçtaki sayma değeri (otomatik olarak) yüklenir ve bu süreç tekrarlanır. (Gate=1 ise sayma devam eder, Gate=0 ise sayma durur.)

Şekil 6. Mod 2 zamanlama diyagramı

### 5.3 Mode 2: Rate Generator

In this mode, the signal output from the OUT pin cyclically becomes low level during the last one CLK of the specified count. The counter here operates as a frequency divider.

Table 5-3 Mode 2 Operation

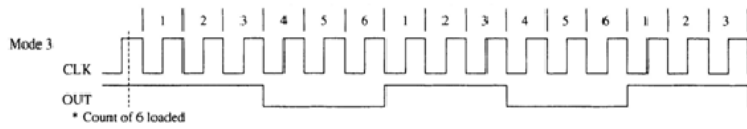
Initial OUT		High level
GATE input	High level	Count enable
	Low level	Count disable. If GATE becomes low level when OUT is low level, OUT will become high level (independent of the CLK pulse).
	Trigger*	Transfer is performed at the first CLK pulse after the trigger.
Count write		Write count data without affecting the current operation.
Count transfer and operation		Transfer operation is performed at the first CLK pulse after the count is written following the mode setting. The count is then decremented. Transfer is again performed at the first CLK pulse after the count becomes 1. When the trigger is present, the transfer operation is repeated at the next CLK pulse.  When the contents of the down counter become 1, OUT becomes low level for one CLK and then returns to high level. Therefore, if a count of N is set, out will repeat this sequence with a period of N CLK pulses.
Count ZERO		Does not occur in this mode
Minimum count		2



Bu modda Gate ucundan lojik-1 uygulandıđı sürece çıkış ucundan sürekli-kare dalga sinyali elde edilir. Sayıcıya yüklenen N değeri çift sayı ise, sayıcı her bir saat darbesinin düşen kenarında 2 azaltılır. Sayıcı sıfır değerine ulaşınca çıkışın durumu değıştirilir ve sayıcıya başlangıçtaki sayma değeri (otomatik olarak) tekrar yüklenir ve tüm süreç tekrarlanır. Eğer sayıcıya yüklenen N değeri tek sayı ise, sayıcı ilk saat darbesinde 1 azaltılır, takip eden saat darbelerinin her birinde ise 2 azaltılır. Sayıcı sıfıra

## 8254 Modes of Operation

- **MODE 3: Square wave generator**
  - Generates a continuous square-wave at OUT as long as G is high.
  - If count is EVEN, OUT is hi for ½ the count, low for the other ½.
  - If count is ODD, the output is high for one extra clock period.



ulaşınca çıkış alçak seviyeye geçer ve sayıcıya başlangıçtaki sayma (değeri otomatik olarak) yeniden yüklenir. (Yeniden yükleme işleminden sonraki) ilk saat darbesinde sayıcı 3 azaltılır ve takip eden saat darbelerinde ise 2 azaltılır. Tüm bu süreç tekrarlanarak sürekli kare dalga üretilir. Çift sayı kullanıldığında üretilen kare dalga sinyalin "Duty Cycle"ı %50 olur. Tek sayıda ise çıkış  $(N+1)/2$  saat darbesi süresince lojik-1,  $(N-1)/2$  saat darbesi süresince lojik-0 olur.

Şekil 7. Mod 3 zamanlama diyagramı

### 5.4 Mode 3: Square Wave Generator

This is a frequency divider like that shown for mode 2. Only the duty cycles of these two modes are different.

Table 5-4 Mode 3 Operation

Initial OUT		High level
GATE input	High level	Count enable
	Low level	Count disable. If GATE becomes low level when OUT is low level, OUT will become low level (independent of the CLK pulse).
	Trigger*	Transfer is performed at the first CLK pulse after the trigger.
Count write		The current operation is not affected. The count is transferred at the end of the half-period of the current square wave. At the same time, the OUT pin becomes high level.
Count transfer and operation		Count data is transferred at the first CLK pulse after the count write following the mode setting. Transfer is performed at the end of the current half cycle and the output of the OUT pin is inverted. Transfer is also performed at the CLK pulse after the trigger is input.  The operation performed will differ depending on whether count N is even or odd. When N is even, the count is decremented by twos. After the count becomes 2, transfer will be performed at the next CLK pulse and the state of the OUT pin will be inverted. This is taken as a half cycle and is repeated thereafter.  When N is odd, N-1 is transferred and the count is decremented by twos. The half cycle when the OUT pin is high level continues until the count value becomes 0 and N-1 is transferred again at the next CLK pulse. The half cycle while OUT is low level continues only until the count becomes 2. For this reason, the half cycle with OUT=1 is one CLK longer than the one with OUT=0.
Count ZERO		Occurs only when the count is odd.
Minimum count		2



### Mode 3 Program example

This is a subroutine that divides the input CLK frequency (5.0688 MHz) by 264 (108H) to obtain an 19,200 Hz clock. Counter #2 is used in the low/high 2-byte read/write mode to perform binary count operation.

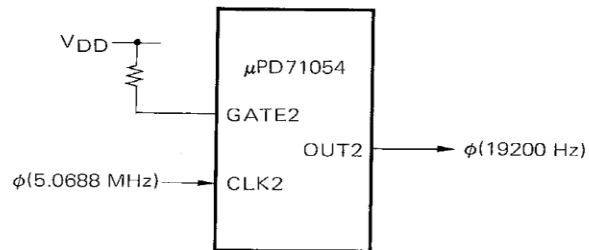
```
SUBR4:  MOV    AL, 10110110B }
        OUT    PCNT2, AL   }

        MOV    AL, 08H    }
        OUT    PCNT2, AL   }
        MOV    A, 01H     }
        OUT    PCNT2, AL   }
        RET
```

Mode specification

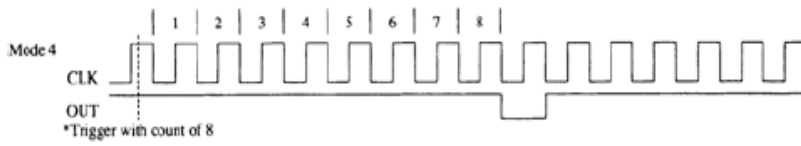
- Counter #2
- Low/high 2-byte read/write mode
- Count mode 3
- Binary

264 frequency division specification



## 8254 Modes of Operation

- **MODE 4: Single-pulse generator (SW one-shot)**
  - Produces a single pulse, one clock width wide, that is delayed based on the count.
  - Starts when count is loaded (SW trigger)



Çıkış başlangıçta yüksek seviyededir. Sayma işlemi sayıcıya değer yüklenmesiyle tetiklenir (software triggered). Sayıcıya yüklenen değer sıfıra ulaştıktan sonra çıkış sinyali, 1 saat periyodu boyunca düşük seviyede kalır. (Gate=1 ise sayma devam eder, Gate=0 ise sayma durur.)

**Şekil 8.** Mod 4 zamanlama diyagramı

### 5.5 Mode 4: Software-Triggered strobe

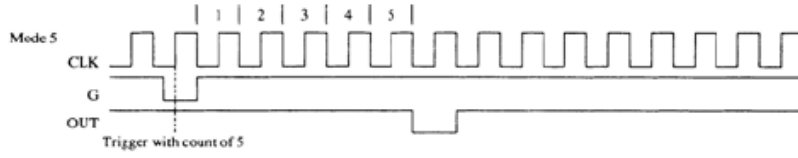
When the specified count is up, OUT will become low level for one CLK pulse only. The strobe is generated only once per a transfer of a count number.

**Table 5-5 Mode 4 Operation**

Initial OUT		High level
GATE input	High level	Count enable
	Low level	Count disable
Count write		When the count is written, the data is transferred at the next CLK pulse. In the 2-byte read/write mode, transfer is performed after the second byte is written.
Count transfer and operation		Data transfer is performed at the first CLK pulse following the count write. If GATE is high level, the down counter will start to decrement from the next CLK pulse, and if GATE is low level, decrement operation will start at the first CLK pulse after GATE becomes high level.
Count ZERO		The OUT pin is low level for one CLK and then becomes high level again. The down counter counts to FFFFH (binary) or 9999 (BCD) without stopping the counter operation.
Minimum count		1

## 8254 Modes of Operation

- **MODE 5: Hardware-triggered one-shot**
  - Same as mode 4 except that the count is started by a triggered pulse on the G pin instead of by software.

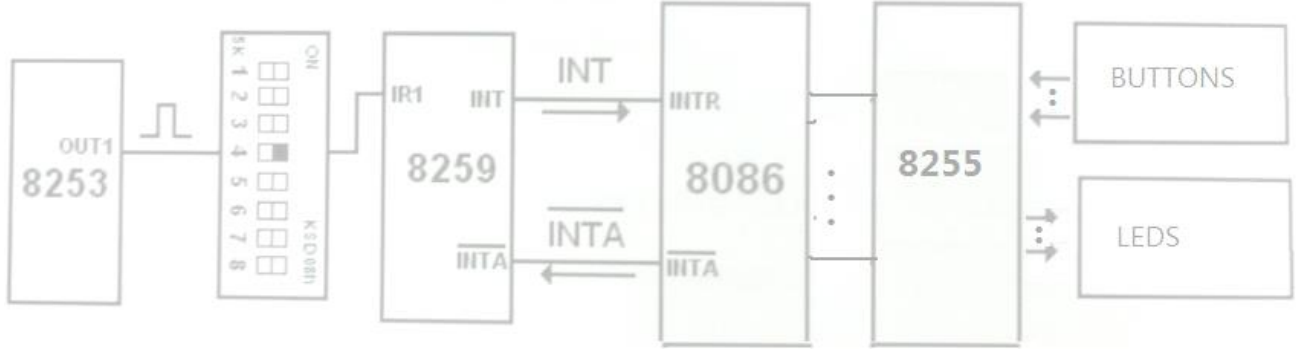


Bu modun çalışması, bir husus dışında Mod 4'ün çalışmasının aynısıdır. Burada sayma işini başlatan yazılım yerine Gate ucundan uygulanan darbedir. Sayıcı programlanır ve sayma değeri registre yazılır. Sayma, Gate ucundan uygulanan darbenin yükselen kenarıyla tetiklenir ve sayıcıya yüklenen değer sifıra ulaştıktan sonra çıkış sinyali, 1 saat periyodu boyunca düşük seviyede kalır.

**Şekil 9.** Mod 5 zamanlama diyagramı

## DENEYİN YAPILIŞI (Deney 1)

**Amaç:** Programın amacı 8253 PIT'in 2. zamanlayıcısını mod 3 te çalıştırarak frekansı değiştirilebilen bir kare dalga üretici yapmaktır. Üretilen kare dalganın frekansını değiştirmek için 8255'in A portuna bağlı butonlar kullanılacaktır. Üretilen kare dalga sinyal 8255'in B Portuna bağlı 8 adet led üzerinden dışarıya yansıtılacaktır.



Deney 1 devre şeması

### Kaynak Program:

```
=====karedalga.ASM=====
COMM1 EQU 0FFC8H ;Define 8259 command address for ICW1, OCW2, OCW3
COMM2 EQU 0FFCAH ; Define 8259 command address for ;ICW2, ICW3,ICW3, OCW1
COUNT1 EQU 0FFDAH ; Define 8253 counter#1 port address
CNT_CSR EQU 0FFDEH ; Define 8253 control word port address
CNT3 EQU 3FD6H ;Define 8255 control word port address

APORT3 EQU 3FD0H ;Define 8255 portA address
BPORT3 EQU 3FD2H ;Define 8255 portB address
CPORT3 EQU 3FD4H ;Define 8255 portC address

CODE SEGMENT
ASSUME CS:CODE,DS:CODE

ORG 0
START: CLI ;CPU'nun kesme isteklerini kabul etmemesi sağlanır

MOV DI,0 ; PORTB'deki LED bilgisini tutan register
MOV SI,0 ; Kesme alt programındaki döngü değişkeni için kullanılan reg.
MOV AX,CS ;CS=DS
MOV DS,AX ;Code segment=Data Segment
MOV SP,0F000H ;Yığın belleğin başlangıç adresi belirlenir.

;<Setup IR1 interrupt vector>
```

```

MOV AX,0 ;Initialize ES to '0'
MOV ES,AX ;Extra Segment =0

MOV BX,41H*4 ;Setup vector address of IR1 to BX=41H*4
MOV ES:WORD PTR[BX],OFFSET INTR1
;IP of ISR write to vector
MOV ES:[BX+2],CS ;CS of ISR write to vector

;<Initialize 8259 Command Words>
;<ICW1>
MOV DX,COMM1 ;Enable command 1 port address
MOV AL,00010011B ; Setup ICW1
OUT DX,AL

;<ICW2>
MOV DX,COMM2 ;Enable command 2 port address
MOV AL,40H ; Setup ICW2
OUT DX,AL

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

;<8259 operation command words>
;<OCW1>
MOV AL,11111101B ;OCW1(Unmask IR1)
OUT DX,AL

;<Setup 8255 control word register>

MOV AL,90H ;AL=90, PortA=input, PortB=PortC=output
MOV DX,CNT3 ;Enable 8255 control port
OUT DX,AL

MOV AL,0FFH ;Timer Counter'larının GATE uçlarına PortC üzerinden lojik-1 uygulamak için
MOV DX,CPORT3 ; 8255 PortC nin adresi DX'e kopyalanır
OUT DX,AL ;PortC ye FfH gönderilerek, 8253 in Gate1 =High yapılır

;<setup 8253 control word register>

I8253: MOV DX,CNT_CSR ;8253 control port registerinin adresi DX' kopyalanır
MOV AL,01110110B ;8253 control word registerini ayarlama bilgisi
OUT DX,AL ; Counter1, mod-3, binary count

;<Output to 8253>
MOV DX,COUNT1 ; 8253 count#1 adresi DX'e kopyalanır
MOV AX,0FFFFH ; count#1 için bölen değeri 65536 yapılır (Divisor set to FFFFh)
OUT DX,AL ;Low Byte of divisor transfer to 8253
MOV AL,AH ;High byte of divisor transfer to AL
OUT DX,AL ; High Byte of divisor transfer to 8253
STI ;Enable interrupt

```

*;<Main code>*

```
MOV    BL,0FFh           ; BL registeri eski buton bilgisini tutan register, ilk degeri FFh olsun

TEKRAR: MOV    DX,APORT3       ; Enable 8255 portA
        IN     AL,DX           ; PortA'daki buton bilgisi okunur

        CMP    AL,BL           ; Buton bilgisinde deęişiklik var mi?
        JZ     TEKRAR         ; Hayır ise, giriři dinlemeye devam.

                                ;Evet ise,
        MOV    BL,AL           ;Eski buton bilgisi güncellenir.

                                ; buton bilgisine göre timer countere yeni sayma degerini gonderilir
        MOV    AH,AL           ; Butonların durumuna göre AH=FF~00 arasında deęerler alır
        MOV    AL,0FFH        ; Divisor'un alt byte ına FF yazılır
        MOV    DX,COUNT1      ; Enable 8253 count#1
        OUT    DX,AL          ; Divisor'un Low Byte'ı 8253'e transfer edilir
        MOV    AL,AH          ; High byte of divisor transfer to AL
        OUT    DX,AL          ; Divisor'un High Byte'ı 8253'e transfer edilir
        JMP    TEKRAR
```

*;<IR1 Interrupt Service Routine>*

```
INTR1:  PUSH   AX              ; AX'i yığına iter
        PUSH   BX              ; BX'i yığına iter
        PUSH   DX              ; DX'i yığına iter

        INC    SI              ; SI bir attırılarak interrupt çağrı adedi sayılır
        CMP    SI,0010H        ; SI==10H?
        JNZ    INTR1_BITIR     ; Hayır ise, INTR1_BITIR etiketine dallanır

        MOV    SI,0            ; Evet ise, SI sıfırlanır (yeniden saymaya başlamak için)
        NOT    DI              ; DI evrilir (DI, PortB deki LED'lere gönderilecek bilgiyi tutar )
        MOV    AX,DI           ; AX=DI yapılır
        MOV    DX,BPORT3      ; Enable BPORT3
        OUT    DX,AL          ; Write AL to BPORT3

INTR1_BITIR: MOV    DX,COMM1      ; Enable 8259 data port address (setup OCW2)
        MOV    AL,20H          ; EOI(End of Interrupt) Command write to 8259
        OUT    DX,AL          ; Output command

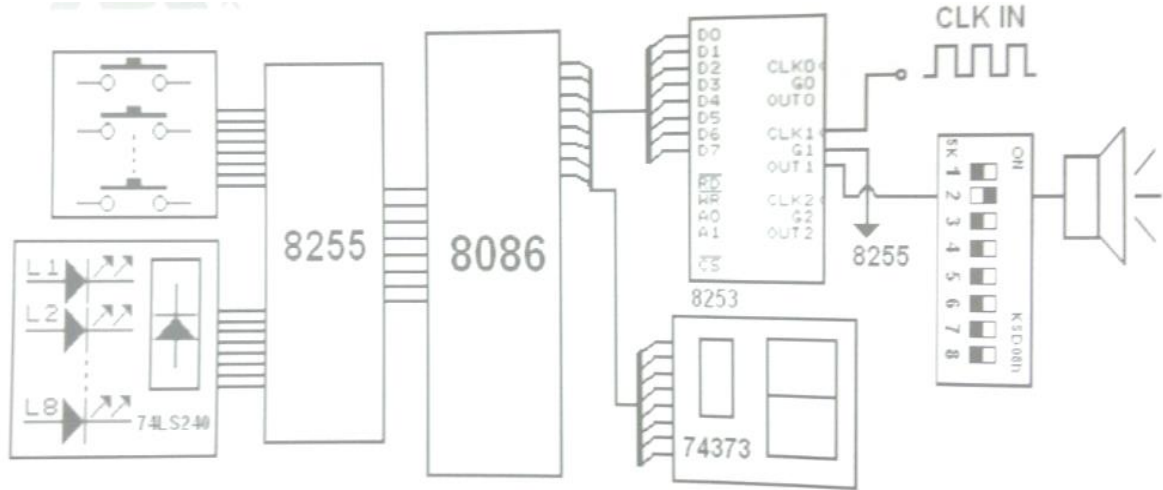
        POP    DX              ; DX yığından çekilir
        POP    BX              ; BX yığından çekilir
        POP    AX              ; AX yığından çekilir

        IRET                   ; Return to main routine

        CODE  ENDS
END      START
```

## DENEYİN YAPILIŞI (Deney 2)

**Amaç:** Program çalıştırıldığında, 8 TACT switch'ten girilen bilgi LED'lerde ve yedi parçalı display'de gösterilecek ve hoparlör aracılığıyla ilgili müzik skalasına dönüştürülecektir. Uygulamaya ilişkin devre şeması aşağıdaki gibidir.



Deney 2 devre şeması

## Kaynak Program:

```
-----piyano.ASM-----
COUNT1 EQU 0FFDAH ;Define 8253 counter#1 port address
CSR EQU 0FFDEH ; Define 8253 control word port address
CNT3 EQU 3FD6H ; Define 8255 control word port address
APORT3 EQU 3FD0H ; Define 8255 portA address
BPORT3 EQU 3FD2H ;Define 8255 portB address
CPORT3 EQU 3FD4H ;Define 8255 portC address
FND EQU 3FF0H ; Define FND port address

ORG 0

CODE SEGMENT
ASSUME CS:CODE,DS:CODE

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

MOV AX,0 ;Initialize ES to '0'
MOV ES,AX ;Extra segment=0
MOV BX,2*4 ;Setup vector address of NMI to BX=02H*4
MOV ES:WORD PTR[BX],OFFSET NMI
MOV ES:2[BX],CS ;IP of ISR write to vector
```



```

MOV DX,CNT3           ;CS of ISR write to vector
MOV AL,90H           ;A PORT=Input, BC PORT=Output
OUT DX,AL            ;Output 90H to 8255 control port

MOV DX,CPORT3        ;Enable 8255 portC
MOV AL,0FFH          ;Setup output to logic High
OUT DX,AL            ;Output FFh so that Gate1 of 8253=High

```

*<Setup 8253 control word register>*

```

18253: MOV DX,CSR      ;Enable 8253 control port
MOV AL,01110110B ;Setup 8253 control Word register
OUT DX,AL          ;Output data to 8253 control port
MOV DX,COUNT1      ;Enable 8253 count#1

```

*<Send Input data to LED>*

```

PLAY:  PUSH DX          ;DX=COUNT1, store to stack
MOV DX,APORT3      ;Enable 8255 PortA (8 bit TACT Switch)
IN AL,DX           ;Input data from TACT switch
NOT AL             ;Reverse the signal from Low to High
MOV DX,BPORT3      ;Enable 8255 PortB (8 bit LED)
OUT DX,AL          ;Output data to LED
POP DX             ;DX=COUNT1, load from stack

```

*<Pressed key changed?>*

```

CMP AL,BL          ;When BL=unknown, ZF=0
                  ;When BL=AL, ZF=1
JZ PLAY           ;If ZF=0 → key change, go next instruction
                  ;If ZF=1 → no key change, jump to PLAY wait
                  ;for next key to be pressed.
MOV BL,AL         ;Copy AL to BL

```

*< Key pressed?>*

```

CMP AL,0          ;When any key pressed → ZF=0
                  ;When no key pressed → ZF=1
JZ PLAY           ;If ZF=0 → key pressed, go to next instruction
                  ; → play music and show in FND
                  ;If ZF=1 → no key pressed, jump to PLAY wait
                  ; for next key to be pressed

```

*< Short period of soundless>*

```

PUSH AX           ;Store input data from TACT switch to stack
MOV AX,10         ;Data (Divisor) for soundless

```

```

OUT    DX,AL                ;The divisor=10, the frequency is too high
MOV    AL,AH                ;to be heard
OUT    DX,AL
MOV    CX,1500              ;Time delay
LOOP   $
POP    AX                    ;Load input data back to AX from stack

```

*< Determine which key is pressed >*

```

PUSH   DX                    ;DX=COUNT1, store to stack
MOV    DX,FND                ;Enable FND port address

TEST   AL,80H                ;"7" key pressed? SW10.7=!7FH=80H
JNZ    DO                    ; If pressed, jump to DO.
TEST   AL,40H                ;"6" key pressed? SW10.6=!BFH=40H
JNZ    RE                    ; If pressed, jump to RE.
TEST   AL,20H                ;"5" key pressed? SW10.5=!DFH=20H
JNZ    MI                    ; If pressed, jump to MI.
TEST   AL,10H                ;"4" key pressed? SW10.4=!EFH=10H
JNZ    FA                    ; If pressed, jump to FA.
TEST   AL,8                  ;"3" key pressed? SW10.3=!F7H=08H
JNZ    SOL                   ; If pressed, jump to SOL.
TEST   AL,4                  ;"2" key pressed? SW10.2=!FBH=04H
JNZ    RA                    ; If pressed, jump to RA.
TEST   AL,2                  ;"1" key pressed? SW10.1=!FDH=02H
JNZ    SY                    ; If pressed, jump to SY.
TEST   AL,1                  ;"0" key pressed? SW10.0=!FEH=01H
JNZ    DO1                   ; If pressed, jump to DO1.

DO:    MOV    AL,11011000B ; ;Number "7" for FND
        OUT    DX,AL                ;Output to FND
        MOV    AX,4697              ;Divisor for "DO"
        JMP    SET8253              ;Jump to SET8253

RE:    MOV    AL,10000010B ; ;Number "6" for FND
        OUT    DX,AL                ;Output to FND
        MOV    AX,4184              ;Divisor for "RE"
        JMP    SET8253              ;Jump to SET8253

MI:    MOV    AL,10010010B ; ;Number "5" for FND
        OUT    DX,AL                ;Output to FND
        MOV    AX,3728              ;Divisor for "MI"
        JMP    SET8253              ;Jump to SET8253

FA:    MOV    AL,10011001B ;4 ;Number "4" for FND
        OUT    DX,AL                ;Output to FND
        MOV    AX,3519              ;Divisor for "FA"
        JMP    SET8253              ;Jump to SET8253

```

```

SOL:      MOV  AL,10110000B ;3
          OUT  DX,AL        ;Output to FND
          MOV  AX,3135      ;Divisor for "SOL"
          JMP  SET8253      ;Jump to SET8253

RA:       MOV  AL,10100100B ;Number "2" for FND
          OUT  DX,AL        ;Output to FND
          MOV  AX,2793      ;Divisor for "RA"
          JMP  SET8253      ;Jump to SET8253

SY:       MOV  AL,11111001B ;Number "1" for FND
          OUT  DX,AL        ;Output to FND
          MOV  AX,2491      ;Divisor for "SY"
          JMP  SET8253      ;Jump to SET8253

DO1:     MOV  AL,11000000B   ;Number "0" for FND
          OUT  DX,AL        ;Output to FND
          MOV  AX,2352      ;Divisor for High "DO"
; <Play Sound>
SET8253:  POP  DX            ;DX=COUNT1, load from stack
          OUT  DX,AL        ;Low Byte of divisor transfer to 8253
          MOV  AL,AH        ;High Byte of divisor transfer to AL
          OUT  DX,AL        ;High Byte of divisor transfer to 8253

          MOV  CX,1500      ;Time delay
          LOOP $
          JMP  PLAY

; <Interrupt Service Routine for NMI>

NMI:     PUSH  DX            ;DX store to stack
          PUSH  AX            ;AX store to stack
          MOV  DX,COUNT1     ;Enable 8253 Count1
          MOV  AX,10         ;Data (Divisor) for soundless
          OUT  DX,AL        ;The divisor= 10, frequency too high to be heard
          MOV  AL,AH
          OUT  DX,AL
          POP  AX            ;AX return from stack
          POP  DX            ;DX return from stack
          IRET

CODE     ENDS
        END  START

```