



İSTANBUL TİCARET ÜNİVERSİTESİ  
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ  
MİKROİŞLEMCİLİ SİSTEM LABORATUARI

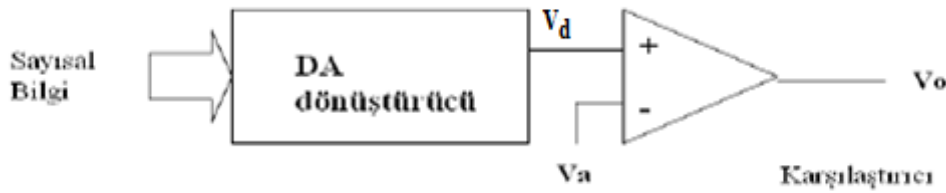


## MİKROİŞLEMCİLİ A/D DÖNÜŞTÜRÜCÜ

### 1. Giriş

Analog işaretler analog donanım kullanılarak işlenebilir. Ama analog donanım üzerinde değişiklik yapmak hem masraflı hem de çok zaman alıcı bir işlemdir. Oysaki yazılım düzeyinde değişiklik yapmak oldukça kolaydır. Bu yüzden yazılımla işlenebilecek işaretler bir A/D dönüştürücü üzerinden sayısal işaretlere dönüştürülür ve değişiklik işlemi sadece yazılım üzerinde yapılır.

Analog işareti sayısal bilgiye dönüştürmenin birçok yöntemi vardır. Şekil 1'de verilen basit bir blok devreyle analog işaret sayısal bilgiye kolayca dönüştürülebilir.

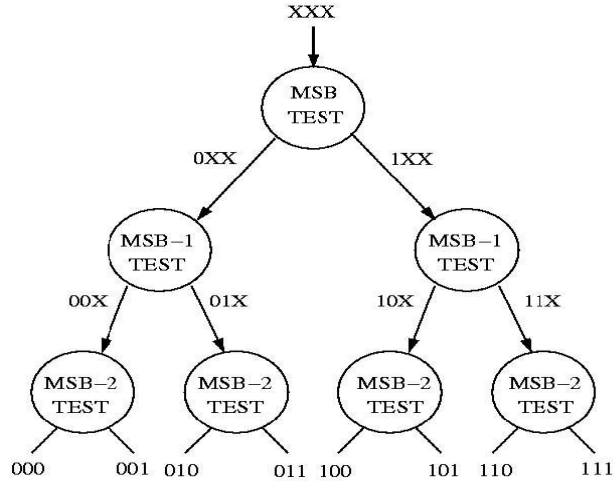


Şekil 1. A/D dönüştürücünün blok şeması

Şekil 1'de D/A dönüştürücüye sıfırdan başlayarak sayısal bilgi sırayla uygulanır.  $V_d$  gerilimi  $V_a$ 'dan küçük olduğu her durum için  $V_o$  0'dan küçük olacaktır.  $V_d$  gerilimi  $V_a$ 'ya eşit gerilimdir ve bu  $V_d$ 'nin sıfırdan büyük ilk  $V_o$  gerilimini veren karşılığı da D/A dönüştürücünün girişindeki sayısal değerdir.

### 2. Successive Approximation Metodu

Şekil 1'deki dönüştürücüye sayısal bilgileri artan sırada uygulamak ve uygun sonuca ulaşmak oldukça yavaş bir yöntemdir. Bunu yerine en anlamlı bitten başlanarak D/A dönüştürücüye uygulanan sayısal bilgilerin bitleri sırayla lojik 1 yapılır ve  $V_o$  gerilimi okunur. Eğer  $V_o \geq 0$  ise o bitin yerine 0, değilse 1 konulur. Bu işlem tüm bitler için tekrarlandığında  $V_a$ 'nın sayısal eşdeğeri elde edilmiş olur. Bunu bir örnekle açıklayalım. Şekil 2'de 3 bitlik bir ADC dönüştürücü için gerekli ağaç yapısı verilmiştir.



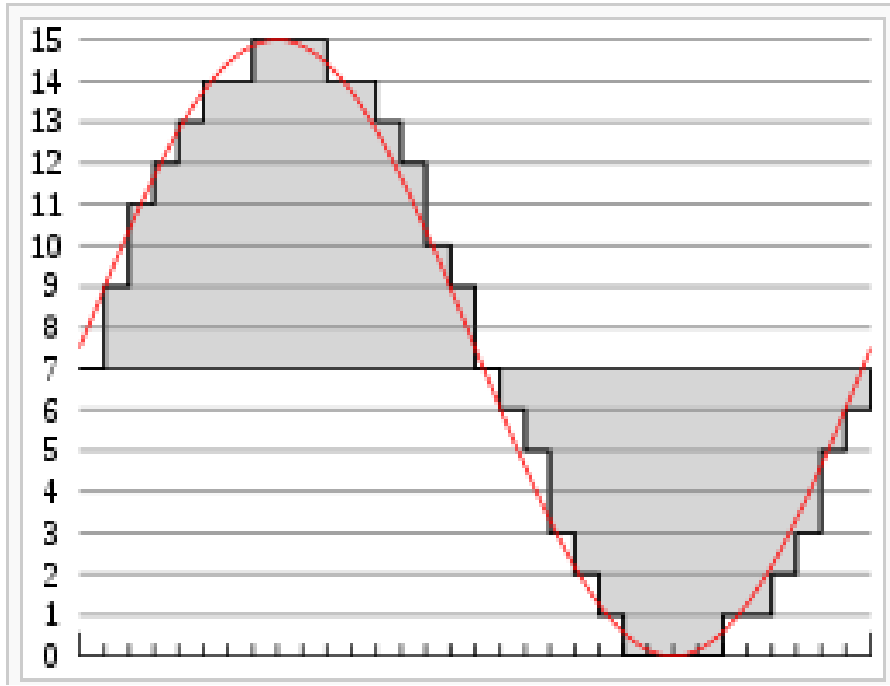
**Şekil 2.** 3 bit için gerekli ikili yaklaşım ağacı

Benzer şekilde deneyde kullanılan 8 bit için gerekli bit testleri de Tablo 1’de verilmiştir.

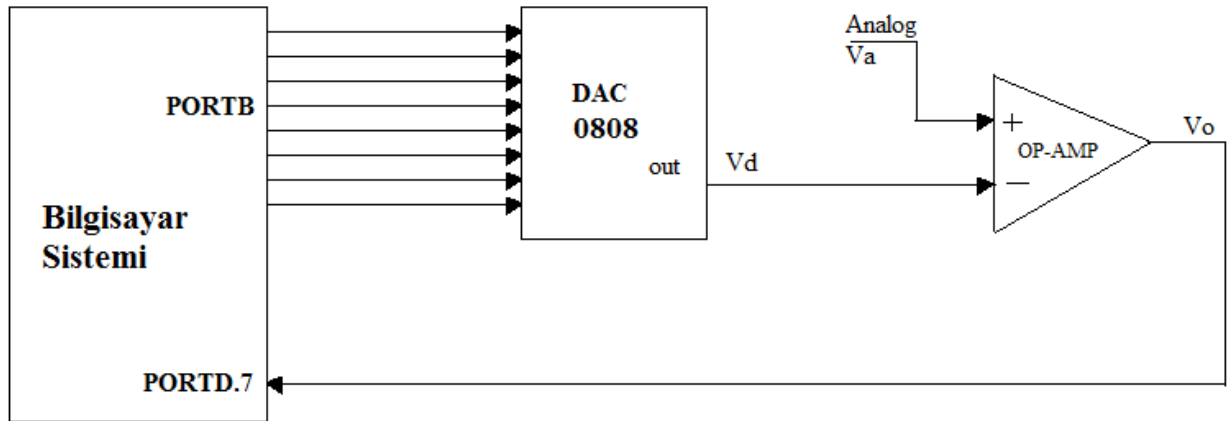
Adım	MSB	LSB	?
1	1 0 0 0 0 0 0 0	$V_o \geq 0$	0
2	0 1 0 0 0 0 0 0	$V_o < 0$	1
3	0 1 1 0 0 0 0 0	$V_o \geq 0$	0
4	0 1 0 1 0 0 0 0	$V_o \geq 0$	0
5	0 1 0 0 1 0 0 0	$V_o < 0$	1
6	0 1 0 0 1 1 0 0	$V_o < 0$	1
7	0 1 0 0 1 1 1 0	$V_o \geq 0$	0
8	0 1 0 0 1 1 0 0	$V_o > 0$	0

**Tablo 1.** 8 bitlik A/D dönüştürücü için denenen sayısal bilgiler

Burada  $V_a$ 'nın sayısal eşdeğeri 8 adım sonunda 01001100 olarak bulunmuştur. Örnekten de görüldüğü gibi elde edilecek sayısal bilgi D/A dönüştürücünün bit sayısına bağlıdır. D/A dönüştürücünün bit sayısı ne kadar az alınırsa sonuç o kadar hızlı bulunmuş olur. Bunun yanında bit sayısı ne kadar fazla alınırsa sonuç o kadar doğru çıkacaktır. Burada dikkate alınması gereken iki kavram karşımıza çıkmaktadır. Bunlar örnekleme (*sampling*) ve kuantalamadır (*quantization*). Şekil 3’te bir sinüs eğrisinin örnekleme ve kuantalaması verilmektedir. Sayısal tasarım bilgilerinizden bu kavramların ne olduğunu araştırınız.



**Şekil 3.** Sinüs İşaretinin örneklenmesi ve 4-bit ile kuantalanması

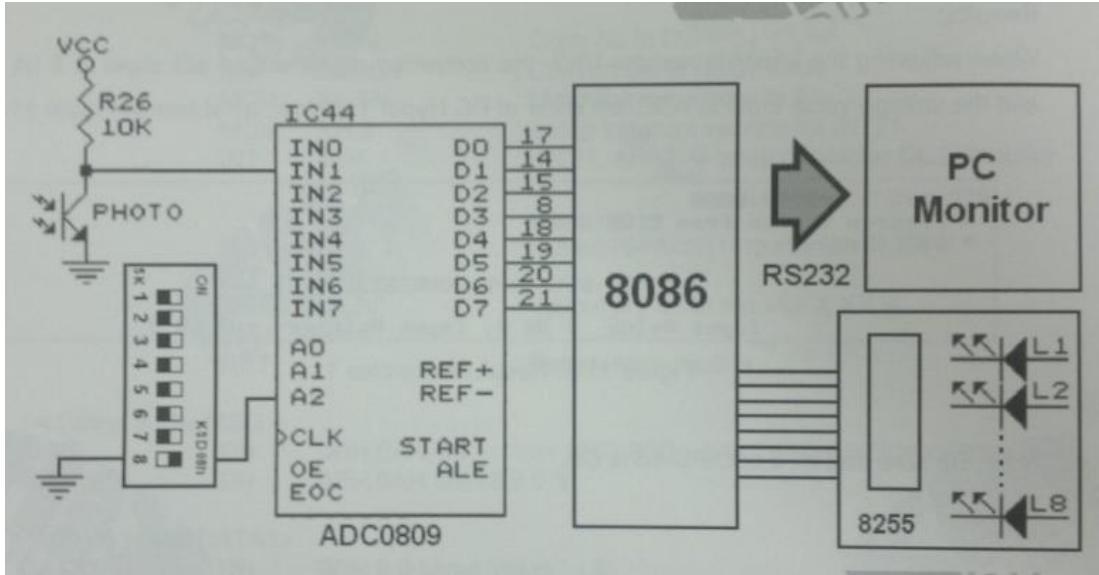


**Şekil 4.** Örnek bir A/D Dönüşüm devresi

## DENEYİN YAPILIŞI

**Amaç:** Bu deneyle farklı elektriksel olmayan büyüklüklerin sayısal olarak ölçülmesi amaçlanmıştır. Büyüklüklerin biri de ışık şiddetidir. Fototranzistörlü devre ışıkla orantılı bir gerilimin üretilmesine yol açar. Program çalıştırıldığında fototranzistörden gelen gerilim bilgisi LED'ler üzerinde binary formatta ve PC üzerinde de hexadecimal formatta görüntülenmektedir.

**Devrenin tanımı:** Fototransistörden akan akım fototransistörün base'inden giren ışık ile orantılıdır ve kolektörde daha az gerilim düşümünü sağlar. Diğer bir deyişle, ADC08092un IN1'ine uygulanan gerilim, fototransistöre giren ışık şiddetiyle ters orantılıdır.



**Şekil 5.** Devre şeması

**Kaynak Program:**

```

;=====ADC.ASM=====
ADC1      EQU  3FCAH          ;Define ADC0809 IN1 port address
CNT3      EQU  3FD6H          ;Define 8255 control word port address
BPORT3    EQU  3FD2H          ;Define 8255 portB address

CODE      SEGMENT
          ASSUME      CS:CODE, DS:CODE

          ORG  0

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

;<Setup 8255 control word register>

          MOV  DX,CNT3         ;Enable 8255 control word
          MOV  AL,91H          ;portA, portC Lower = input
          OUT  DX,AL           ;portB, portC Upper = output

          MOV  DX,OFFSET MSG    ;Setup starting address of buffer MSG
          MOV  AH,9             ;Output string MSG to monitor
          INT  21H              ;Interrupt for general MS-DOS service

```

*;<Enable and initialize ADC0809>*

```
J1:      MOV  AX,0           ;Initialize AX register
        MOV  DX,ADC1      ;Enable ADC0809 ADC1
        OUT  DX,AL        ;Output data 00h to ADC0809 IN1

        MOV  CX,20H      ;Time Delay
        LOOP $
```

*;<Input data and output to LED>*

```
        IN   AL,DX       ;Input data from ADC0809 ADC1

        MOV  DX,BPORT3   ;Enable 8255 portB (LED)
        OUT  DX,AL       ;Output ADC1 data to 8255 portB
```

*;<Output Word AX to monitor>*

```
        MOV  DH,4        ;When DH=2, output =1 byte (AL)
                          ;When DH=4, output =1 word (AX)
        INT  10H         ;Output AX to monitor

        MOV  CX,0        ;Time delay
        LOOP $

        MOV  DX,OFFSET BACK ;Setup starting address of buffer BACK
        MOV  AH,9        ;Output string BACK
        INT  21H        ;Interrupt for general MS-DOS services
        JMP  J1         ;Jump to J1
```

*;<String buffer MSG>*

```
MSG      DB  9,9,'=== MTS-86C A/D CONVERTER TEST ==='
        DB  0DH,0AH,0AH,9,9,9,9,'$'
```

*;<String buffer BACK>*

```
BACK     DB  8,8,8,8,'$'
```

```
CODE     ENDS
        END  START
```