

SIFIR DOKUZ ARASI SATILARIN  
FILTRE ÇIKISLARINA GÖRE TANINMASI

AYHAN EROL

Anadolu Üniversitesi  
Fen Bilimleri Enstitüsü  
Lisansüstü Yönetmeliği Uyarınca  
Elektrik-Elektronik Mühendisliği Anabilim Dalı  
Elektronik Bilim Dalında  
YÜKSEK LİSANS TEZİ  
Olarak Hazırlanmıştır.

Danışman: Prof. Dr. Atalay Barkana

EYLÜL 1990

ANADOLU ÜNİVERSİTESİ  
MERKEZ KÜTÜPHANESİ

AYHAN EROL 'un "YÜKSEK LİSANS" tezi olarak hazırladığı "SIFIRDAN DOKUZ SAYILARIN FİLTRE ÇIKIŞLARINA GÖRE TANINMASI" bu çalışma, jürimizce lisansüstü yönetmeliğinin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir.

22/10/1990

üye : Prof. Dr. Atalay Barkana

üye : Prof. Dr. Atilla Barkana

üye : Doç. Dr. Hamdi Atmaca

Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 24.10.1990  
gün ve 259.-3 sayılı kararıyla onaylanmıştır.

Prof. Dr. Rüstem KAYA  
Enstitü Müdürü

## İÇİNDEKİLER

	Sayfa
ÖZET . . . . .	111
ABSTRACT . . . . .	19
ŞEKİLLER DİZİNİ . . . . .	v
TANIMLAR DİZİNİ . . . . .	vi
TEŞEKKÜR . . . . .	vii
1.0 GİRİŞ . . . . .	1
2.0 SES TANIMA SİSTEMLERİ TARİHÇESİ . . . . .	2
3.0 SES TANIMA METODU VE DİNAMİK PROGRAMLAMA . . . . .	3
4.0 DONANIM . . . . .	7
4.1 GİRİŞ/ÇIKIŞ BİRİMİ . . . . .	7
4.2 DİĞER ELEMANLAR . . . . .	7
4.3 ANALOG GİRİŞ SENTEZLEYİCİ . . . . .	10
4.3.1 Ön-kuvvetlendirici . . . . .	12
4.3.2 Filtre Bankası . . . . .	12
4.3.3 Integrator Bölümü . . . . .	14
4.3.4 Analog Digital Çevirici . . . . .	15
5.0 YAZILIM . . . . .	16
6.0 SONUÇLAR . . . . .	19
7.0 KAYNAKLAR DİZİNİ . . . . .	21
8.0 EKLER . . . . .	22
8.1 Ek-1 0-9 Arası Sayıların Örnek Filtre Çıktıları	23
8.2 Ek-2 Programlar . . . . .	34
8.3 Ek-3 Program Menu Çıktısı . . . . .	49
8.4 Ek-4 Çoklu Giriş/Çıkış Kartının Baskı Devre Şeması	51
8.5 Ek-5 Analog Sentezleyici Kartın Baskı Devre Şeması	53

## OZET

Ses tanıma çalışmaları son yıllarda büyük ölçüde gelişmiştir. Günümüzde ses tanıma sistemleri kullanarak konuşma aracılığıyla her çeşit makina, kontrol panel'i ve bilgisayarlar ile iletişim kurmak mümkündür. Bu tez çalışmasında PC XT/AT tipi bilgisayarlarda kullanılabilen sıfır'dan dokuz'a kadar sayıları tanıyabilecek donanım ve yazılım tasarlanmıştır. Yazılan programla ses incelemesi ve konuşmacılar değişikçe şablonların yenilenmesi gibi özellikler de sağlanmıştır. Gerçekleştirilen çoklu giriş/çıkış kartında 72 port bulunmakta ve bu port'ların 16 tanesi analog giriş sentezleyici kart için kullanılmaktadır. Diğer port'lar boşta bulunmaktadır. Boş kalan port'lar Bilgisayar destekli ses kontrol sistemleri yapmada kullanılabilir.

## ABSTRACT

The speech recognition studies have advanced greatly in the last decade. Nowadays, it is possible to communicate reliably with every kind of machine, control panel and computer by speaking to them, using speech recognition system. In this thesis working, hardware and software have been designed to be used on PC XT/AT computers to recognize numbers which are from zero to nine. The software also provides making speech analysis and updating of templates. The realized multi i/o card has 72 ports where 16 ports of them are used for analog card. Other ports of multi i/o card are not connected to any peripheral. These ports provide possibility to make speech controller system.

## ŞEKİLLER DİZİNİ

Şekil		Sayfa
3.1	Ses Tanıma Devresinin Blok Şeması . . . . .	4
3.2	(m,n) Noktasından Origin'e en Kısa Mesafeli Gidiş Yolu . . . . .	6
4.1	Port Elemanları İçin Kod Çözücü Bölümü . . . . .	8
4.2	Çoklu Giriş/Çıkış Kartının Açık Devre Şeması . . . . .	9
4.3	Analog Giriş Sentezleyici . . . . .	11
4.4	Ön-kuvvetlendirici Devresi . . . . .	12
4.5	Aktif Band Geçiren Filtre . . . . .	13
4.6	Integrator Devre Şeması . . . . .	14

**TANIMLAR DIZINI**

ADC (Analog Digital Convertor)	: Analog Sayısal Çevirici
I/O (Input/Output)	: Giriş/Çıkış
OPAM (Operational Amplifier)	: İşlemsel Kuvvetlendirici
PIO (Parallel Input Output)	: Paralel Giriş Çıkış
MULTI I/O	: Çoklu Giriş/Çıkış

### TEŞEKKÜR

Bu tez çalışmasını bana veren ve araştırmalarımın gerek fikir gerekse kaynak yönünden yardımcı olan sayın hocam Prof. Dr. Atalay Barkana'ya teşekkür ederim.

Ayrıca tez süresince her konuda bana yardımcı olan değerli arkadaşlarım Arş. Gör. Rifat Edizkan'a ve Arş. Gör. Hakan Tora'ya teşekkür ederim.

## 1.0 GİRİŞ

Bilgisayar dünyası geliştikçe buna baęlı olarak kontrol sistemleri de gelişmektedir. Bilgisayarlarda ses tanınmasıyla deęişik amaçlı kontrol sistemleri kurulabilir.

Ses tanınmasında günümüze kadar deęişik yaklaşımlar kullanılmış ve 1980'li yıllardan sonra daha matematiksel işlemlerin yer aldığı yaklaşımların kullanıldığı görülür.

Bu tez çalışmasında seslerin deęişik frekanstaki enerjilerine bakılarak oluşturulan patern'leri Dinamik Programlama Algoritmi kullanarak ses tanıma yoluna gidilmiştir. Seslerin deęişik frekanstaki enerjileri, seslerin mikrofon, yükseltici, filtre, integratör ve ADC'den geçirilerek sayısal olarak elde edilmiştir. Hazırlanan programla ayrıca ses analizi yapmak'ta mümkündür.

Bu tez çalışmasında kullanılan donanım iki kart'tan oluşmaktadır. Bunlar çoklu giriş/çıkış kartı ve analog girişi sentezleyici kart'tır. Analog sentezleyici kart'ta filtrelerin merkez frekansının veya integratör kazançlarının deęistirilmesi mümkündür.

## 2.0 SES TANIMA SİSTEMLERİ TARİHÇESİ

Ses tanıma sistemleri 1930'lu yıllarda başlamıştır. Bu yıllarda bir telefon abonesi başka bir aboneyi aramak istediğinde telefon operatörüyle kontak kurup görüşmek istediği telefon numarasını vermek zorunda idi. Telefon operatörlerini aradan kaldırmak için dijital tanıması denenmiştir. Farklı konuşmacılarda başarısız olmuş ve telefon operatörlerini aradan kaldırmak için başka çözümler aramaya gidilmiştir.

1950'li yıllardan sonra sayısal bilgisayarların işletime girmesiyle ses tanıma çalışmaları çekici olmaya başlamıştır. Ses tanıma çalışmalarında günümüze kadar sırasıyla akustik, patern tanıma, dilci ve pragmatik yaklaşımlar kullanılmıştır.

Pragmatik yaklaşımda, izole edilmiş kelime tanıyıcılarında Dinamik Programlama Algoritması kullanılması büyük önem kazanmıştır. İlk Dinamik Programlama Algoritması, Rusça 200 kelimeyi % 5 hatayla tanıyan bir sistemde Velichko ve Zalgoruyko tarafından kullanılmıştır (1970).

Daha sonraları bu metodun gücü, Dinamik Programlama Algoritmasının bir sayısını optimize eden Sakoe ve Chiba tarafından belirtilmiş ve onların sistemleri Japonca dijitaler için % 0,2 hata oranına erişmiştir (1978).

İkinci düzey Dinamik Programlama Algoritmi, bağlantılı dijitalerin dilimlerinin tanınmasında Sakoe tarafından araştırılmıştır. Oluşturulan sistemin tanıma oranı % 99,6 olarak bulunmuş ve bu algoritma NEC DP200 sisteminin temeli olarak kullanılmıştır. Yine aynı amaçla daha verimli bir algoritma Bridle ve Brown tarafından geliştirilmiştir (1979). (Kaynak 3)

### 3.0 SES TANIMA METODU VE DINAMİK PROGRAMLAMA

Bu tasarımda sesi tanımak için, sekiz band geçiren filtre, integratörler ve ADC kullanılmıştır. Mikrotondan gelen ses sinyali ön kuvvetlendirici kısmında yükseltilir. Kuvvetlendiriciden gelen ses sinyali sekiz band geçiren filtrelerden geçirilir. Filtre çıkışları integratörlere girer ve integratörler vasıtasıyla her filtre çıkışındaki ses sinyalinin enerjisi elde edilir. Integratörler 20 ms'de bir resetlenerek ses sinyalinin 20 ms'lik enerjisi, analog sayısal çevirici (ADC) vasıtasıyla sayısal olarak okunur. 0-9 arası sayıları söyleme süresi yaklaşık 500-800ms arasındadır olduğundan ses kaydında 20 ms'de bir 44 vektör okunur.

$$X_1 = \begin{bmatrix} x_{11} \\ x_{12} \\ x_{13} \\ x_{14} \\ x_{15} \\ x_{16} \\ x_{17} \\ x_{18} \end{bmatrix} \quad X_2 = \begin{bmatrix} x_{21} \\ x_{22} \\ x_{23} \\ x_{24} \\ x_{25} \\ x_{26} \\ x_{27} \\ x_{28} \end{bmatrix} \quad \dots$$

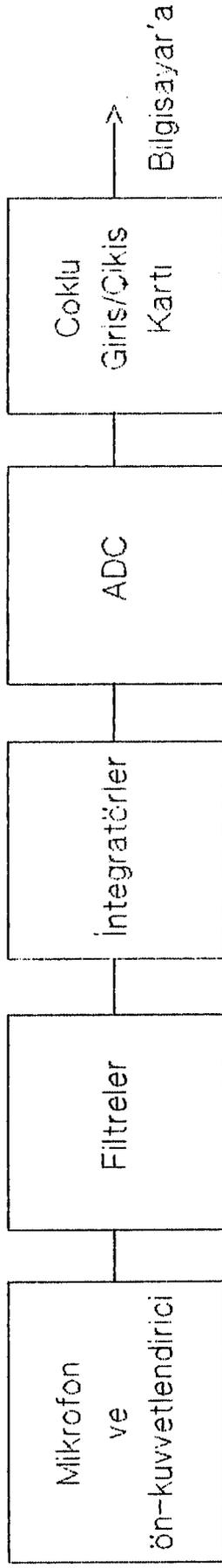
n=1                      n=2

Elde edilen vektörler ( $X_i$ ), şablonlarla dinamik programlama yöntemiyle karşılaştırılır. En yakın şablon seçilerek ses tanınması yapılmış olur. Şekil 3.1'de ses tanımada kullanılan devrenin blok diagramı verilmiştir.

Dinamik programlama ile şablonlar ve giriş patern vektörlerinin bağlı çiftleri arasındaki mesafeler hesaplanır. Bu durumda giriş paterni ve şablonlar arasındaki uzaklık:

$$D_k = \sum_{j=1}^m \sum_{i=1}^n (X_{ij} - Y_{ijk})$$

burada  $X_{ij}$  giriş patern elemanları ve  $Y_{ijk}$  k'inci şablonun elemanlarıdır.



Sekil 3.1. Ses Tanıma Devresinin Blok Şeması

Patern'nin ait olduğu sınıf, patern'den minimum uzaklığa sahip şablonun seçilmesiyle bulunur. Patern ve şablon vektörleri farklı sayıdaki dizilerden oluştuğunda, lineer zaman normalizasyonu uygulanır.

Bu durumda uzaklık matrisi:

$$dk(i, j) = \sum_{q=1}^p (X_i(q) - Y_{jk}(q))$$

burada  $X_i(q)$  bilinmeyen patern'in elemanları,  $Y_{jk}(q)$  n p boyutlu vektörleri içeren k'inci şablonun elemanlarıdır.

Kümülatif uzaklık matrisi  $g(m, n)$  recurrence bağlantısına göre lokal uzaklık matrisinden hesaplanır. Kümülatif uzaklık matrisi:

$$g(i, j) = \min ( g(i-1, j) + d(i, j), g(i-1, j-1) + kd(i, j), g(i, j-1) + d(i, j) )$$

burada k sabitinin optimum değeri bilinmiyor fakat genellikle 1 yada 2 olarak alınır.

Kısa dizili patern'leri kompanse etmek için kümülatif uzaklık matrisi normalize edilir. Toplam kümülatif uzaklık:

$$G(m, n) = \frac{g(m, n)}{m+n}$$

burada m şablon vektörlerinin sayısı ve n bilinmeyen patern vektörlerinin sayısıdır.

Şekil 3.2 (m, n) noktasından origin'e doğru dinamik programlamayla bulunmuş en kısa mesafeli gidiş yolu verilmiştir.



#### 4.0 DONANIM

Ses tanıma sisteminin donanım kısmı, iki karttan oluşmaktadır. İlk kart PC'ye ara bağ sağlayan çoklu giriş/çıkış, ikinci kart ise ön-kuvvetlendirici, filtreler, integratörler ve ADC'nin bulunduğu karttır.

Çoklu giriş/çıkış kartı, IBM PC veya IBM PC uyumlu bilgisayarlarda kullanılabilecek PC ile PC harici sayısal çevre arasında arabag kurmak için kullanılmıştır. Çoklu giriş/çıkış kartı 8255 paralel giriş/çıkış elemanı, 74LS138 kod çözücü ile 74LS85 4-bit'lik karşılaştırıcı tümdevrelerinden oluşmaktadır.

#### 4.1 GİRİŞ/ÇIKIŞ BİRİMİ

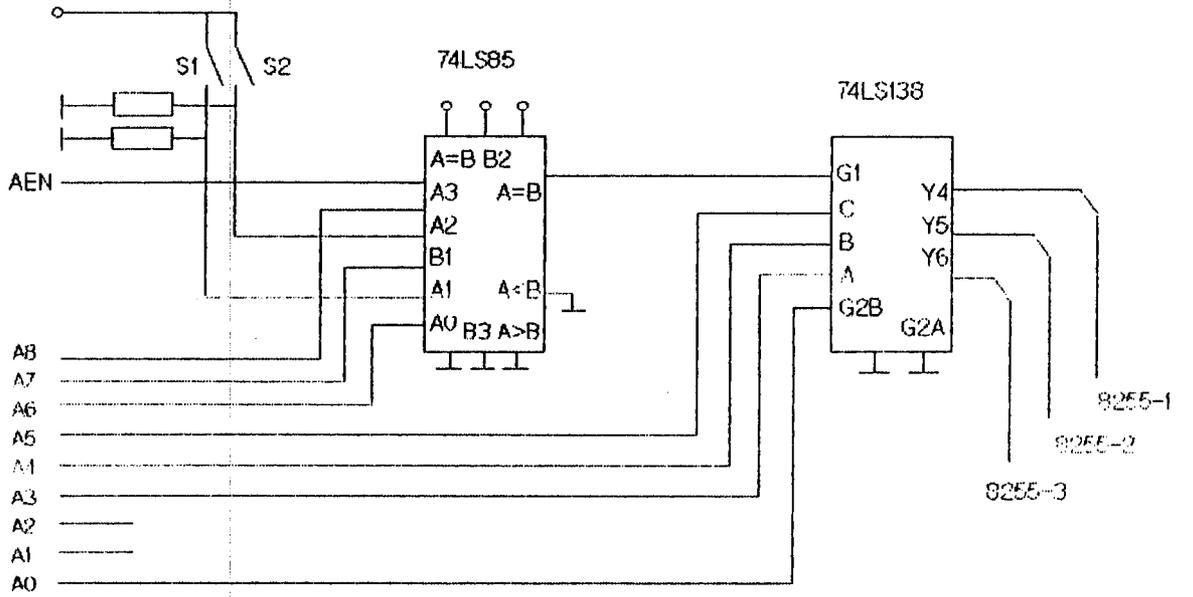
Giriş/çıkış birimi olarak üç adet 8255 PIO (parallel input/output) tümdevresi kullanılmıştır. 8255 PIO iki yönlü veri transferi için programlanabilen 3 adet 8-bit'lik giriş/çıkış portuna sahiptir.

#### 4.2 DİĞER ELEMANLAR

Kod çözücü devresi ikili anahtar, karşılaştırıcı ve kod çözücü tümdevresi ile tasarlanmıştır. Şekil 4.1 de çoklu giriş/çıkış kartında 8255'lerin enable'ı için kullanılan kod çözücü bölümü verilmiştir.

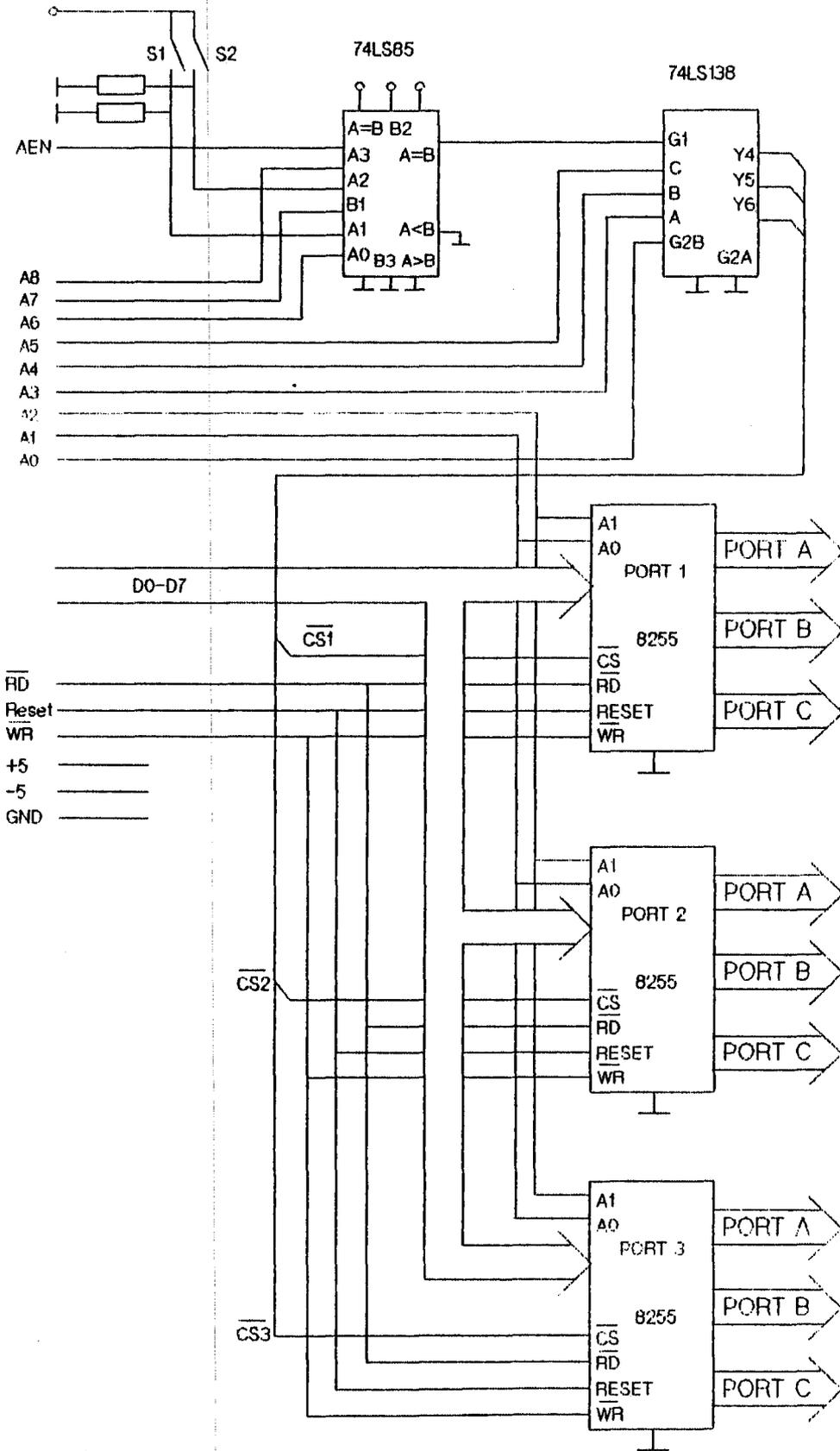
Kod çözücüde S1, S2 anahtarlarının durumuna göre A6 ve A7 adres bit'lerinin durumlarıyla 74LS138 enable edilir. S1-S2'nin durumuna göre port elemanlarına erişim 120H-1FBH adresleri arasında olur.

Giriş/çıkış kartı üzerinde 40 çıkışlı iki konnektör vardır (ST1 ve ST2). İkinci kart ST1 konnektörüne bağlanır.



Sekil 4.1 Port elemanlari için kod çözücü bölümü

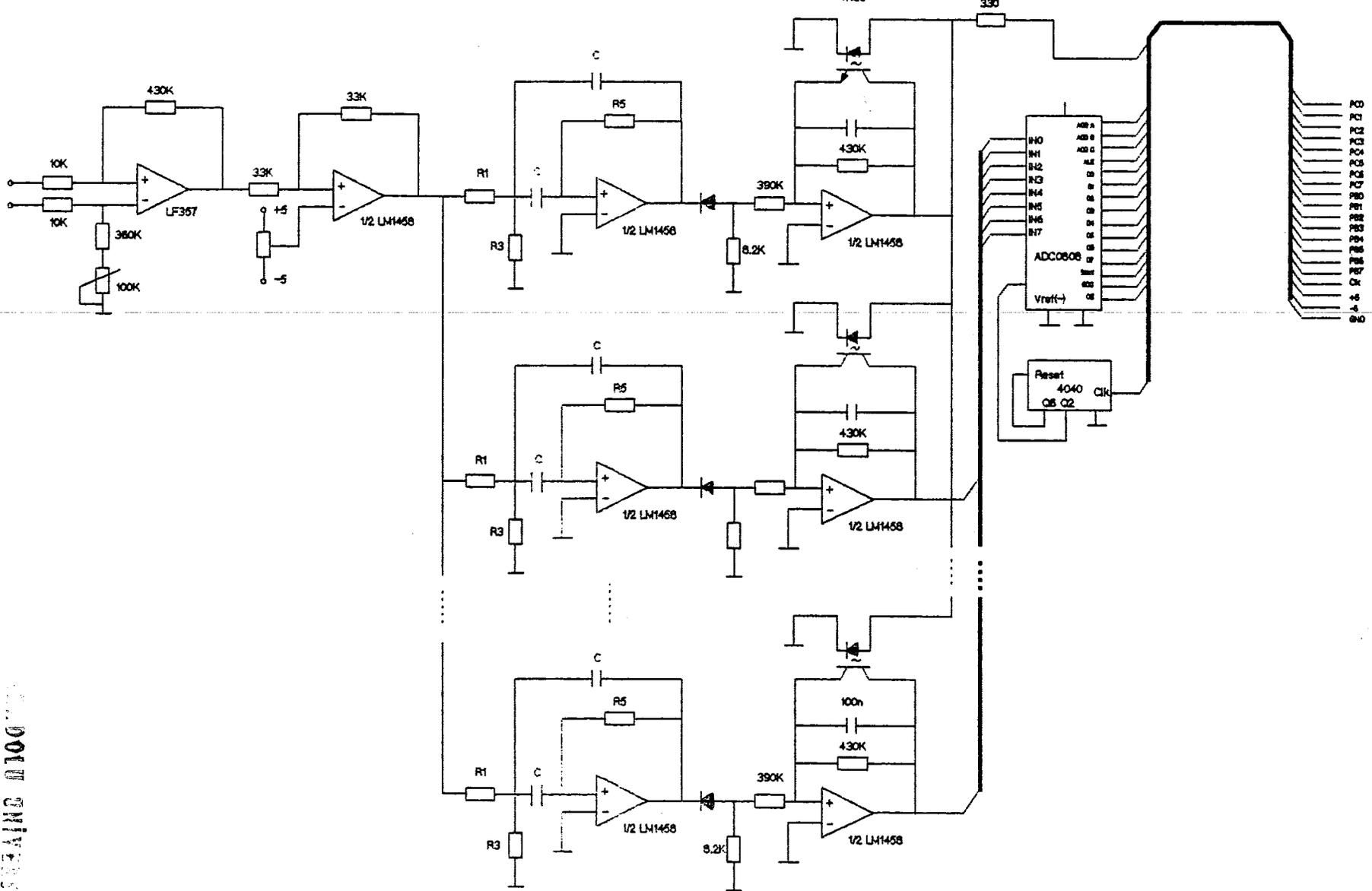
İkinci kart'ın kontrol ve veri hatları için birinci 8255 için B ve C portları kullanılmıştır. Sekil 4.2 de yuklu giriş/cıkış kartının açık devre şeması verilmiştir.



Sekil 4.2. Çoklu giriş/çıkış kartının açık devre seması

#### 4.3 ANALOG GİRİŞ SENTEZLEYİCİ

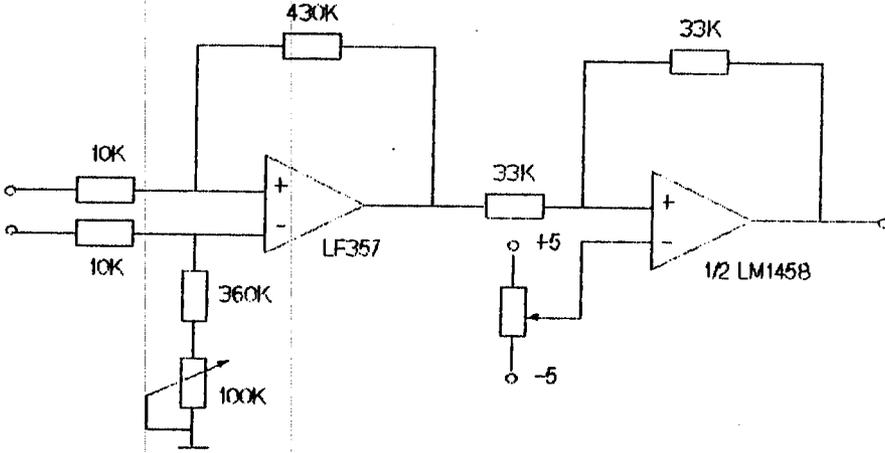
Mikrofondan girilen ses, kazancı 40 olan önkuvvetlendiricide kuvvetlendirilir. Kuvvetlendirici çıkışı sekiz band geçiren filtre girişine uygulanır. Filtre çıkışları bilgisayardan gelecek darbe ile resetlenebilen integratör girişlerine uygulanır. Integratör çıkışları ADC0808'in girişlerine bağlıdır. Bilgisayardan gelen adres'e göre bu girişlerden biri okunur. 20 ms'de bir sekiz elemanlı vektör okunabilir. Vektör eleman sayısı yazılımla 1 ile 8 arasında değiştirilebilir. Şekil 4.3 de kartın açık devre şeması verilmiştir.



Şekil 4.3. Analog Giriş Sentezleyici

#### 4.3.1 Ön-kuvvetlendirici

Bu bölümde JFET girişli OPAM LF357 kullanılarak feedback yoluyla giriş sinyali yükseltilmiştir. Çıkışta DC seviyeyi ayarlayan kazancı 1 olan yükseltici katı vardır. Şekil 4.4 de ön-kuvvetlendirici devre şeması verilmiştir.



Şekil 4.4 Ön-kuvvetlendirici devresi

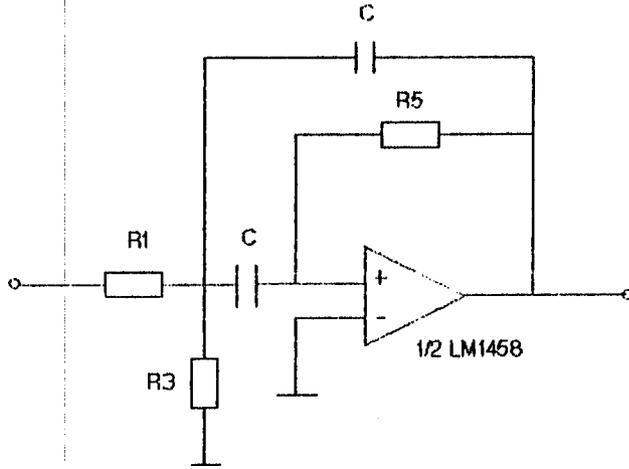
#### 4.3.2 Filtre Bankası

Filtre bankası 8 band geçiren filtreden oluşmaktadır. Filtrelerin merkez frekansları ve band genişliği Tablo 1'de verilmiştir.

f <sub>0</sub>	BW
500	200
800	200
1200	400
1600	400
2100	600
2700	600
3300	600
4900	700

Tablo 1 Filtrelerin merkez frekansları ve band genişlikleri

Aktif band geçiren filtre devresi Şekil 4.5 verilmiştir.



Şekil 4.5 Aktif band-geçiren filtre

Bu devrenin merkez frekansı :

$$\omega_0 = \frac{1}{1/(R_5 \cdot C) (1/R_1 + 1/R_3)} \quad \text{dir.}$$

Kazanç  $G$ , kalite faktörü  $Q$  ve  $C$  değeri verilmiş olsun. Bu durumda:

$$R_1 = \frac{Q}{(GC\omega_0)},$$

$$R_3 = \frac{Q}{(2Q-G)C\omega_0}$$

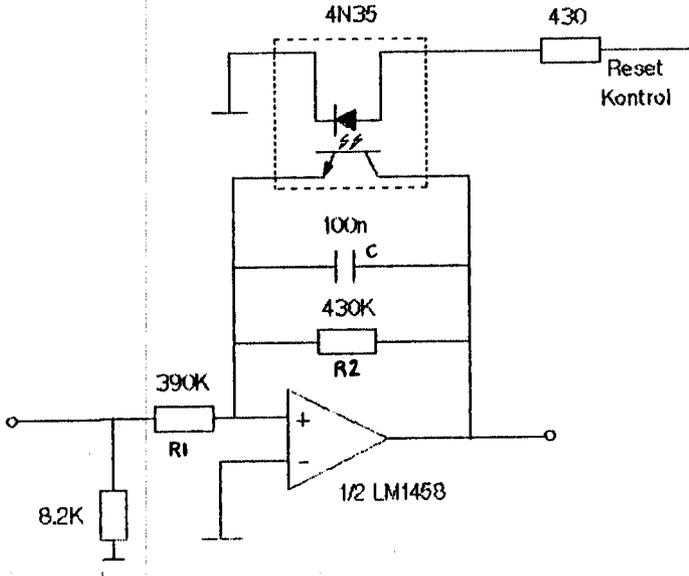
ve

$$R_5 = \frac{2Q}{C\omega_0} \quad \text{olarak hesaplanır.}$$

Devrenin kazancı  $G = R_5/(2R_1)$  dir.

### 4.3.3 İntegratör Bölümü

İntegratörler zaman sabiti (  $t = R2C$  ) 0.043 sn olan, LM1458 OPAM ve kondansatörü resetlemek için opto transistor kullanılarak tasarlanmıştır. Kondansatörün resetlenmesi bilgisayardan gelen darbe ile yapılır. İntegratör devre şeması Şekil 4.6 da verilmiştir.



Şekil 4.6 İntegratörün devre şeması

Bu devrenin transfer fonksiyonu:

$$\frac{V_o(s)}{V_i(s)} = -1/(R1C) \frac{1}{s + (1/R2C)} \text{ dir.}$$

#### 4.3.4 Analog Sayısal Çevirici

ADC integratör çıkışındaki analog sinyali bilgisayarda kullanabileceğimiz formatta sayısal'a çevirmek için kullanılmıştır. ADC olarak 8 analog girişli 8-bit'lik ADC0808 seçilmiştir. ADC0808 üzerindeki A, B, C adresleriyle giriş kanalı seçilir.

## 5.0 YAZILIM

Yazılım ses kaydı, ses verilerinin bir dosyadan yüklenmesi, ses verilerinin bir dosyada saklanması, integratör çıktılarının grafiği, şablonların yenilenmesi, ses tanıma ve şablonların ekranda görüntülenmesi bölümlerini içerir. Program Pascal dilinde yazılmış ve ses kaydı için asembler bölümü Pascal'da unit olarak oluşturulmuştur.

Program klavye'den SPANAL6 girilerek işleme alınır. Program çalıştığında Ek'3 de verilen menu, ekrana gelir. Kursor alt üst oklar'a basılarak hareket ettirilebilir ve aynı anda kursor'un bulunduğu seçenek hakkında kısa bilgi verir. Herhangi bir seçeneği işleme almak için kursor o seçeneğin üzerine getirilerek Enter tuşuna basılır.

Program çalıştığında işleme alınabilecek sekiz seçenek vardır. Bunlar:

- 1- RECORD
- 2- LOAD
- 3- SAVE
- 4- GRAPHIC
- 5- UPDATE TEMPLATE
- 6- SPEECH RECOGNIZATION
- 7- DISPLAY TEMPLATE
- 0- QUIT

### 1- RECORD

Bu kısım mikrofondan ses kaydı yapmak için kullanılır. Kursor bu seçeneğin üzerine getirilip enter tuşuna basıldığında ses kaydına geçmek için herhangi bir tuşa basılması mesajı gelir. Herhangi bir tuşa basıldıktan sonra

mikrofondan ses girilmesi beklenir. Ses kaydı yapmak için bir eşik değeri vardır. Mikrofondan gelen ses eşik değerini aştığında ses kaydı yapılır. Sekiz integratör çıkışından 44'er örnek alınır ve ana menüye dönlür. Ses PC RAM'inin \$6000:\$4000-\$6000:\$4108 adreslerine kayıt edilir.

## 2- LOAD

Bu seçenek ile ses verileri verilen bir dosyadan RAM'in \$6000:\$4000-\$6000:\$4108 adreslerine yüklenir. İşlem tamamlandıktan sonra ana menü'ye dönlür.

## 3- SAVE

Bu seçenekle RAM'deki ses verileri verilen bir dosyada saklanır.

## 4- GRAPHIC

Mikrofondan veya bir dosyadan yüklenmiş sekiz filtreye ait ses verilerinin normalize edilmiş şeklini ekrana verir. Bu seçenekten çıkmak için ESC tuşuna basılmalıdır.

## 5- UPDATE TEMPLATE

Şablonları yenilemek için kullanılır. Bu seçenek işleme alındığında yenilenecek şablon numarası verilerek o sayıya ait şablon normalize edilmiş yenişablon ile yenilenir.

## 6- SPEECH RECOGNIZATION

Kayıt edilmiş veya bir dosyadan yüklenmiş patern'lerin dinamik programlama metoduyla ait olduğu şablon bulunur ve ekrana kısa bir mesajla bilinmeyen patern'in hangi şablona ait olduğu verilir.

## 7- DISPLAY TEMPLATE

Herhangi bir şablonu ekran'da görüntülemek için kullanılır.

## 8- QUIT

Programdan çıkılır.

Bu programın çalışması için aşağıdaki dosyaların disk veya disket'te aynı directory'de olması gerekir.

SPANAL6.EXE  
TEMPLATE.DAT  
SPANAL8.DAT  
TEMPLAT8.DAT  
CGA.BGI  
EGAVGA.BGI  
HERC.BGI

SPANAL6 Programı altı filtre çıkışına göre çalışır ve bu program'a ait şablon TEMPLATE.DAT isimli dosyada saklanır. SPANAL8 Programı sekiz filtre çıkışına göre çalışır ve bu program'a ait şablon TEMPLAT8.DAT isimli dosyada saklanır. Extension'ı BGI ile biten dosyalar çeşitli grafik kartlarına ait grafik sürücülerdir. Bu dosyalar olmadan GRAPHIC ve DISP. TEMP. bölümleri işleme alınamaz ve program çöker. Ek'2 de programların dokümü verilmiştir.

## 6.0 SONUÇLAR

Mikrofondan girilen sesin iyi algılanabilmesi ve ses kaydının başlangıcını tespit etmek için ses kaydı esik değeri verildi. Programdan ses kaydı seçeneğine girildikten sonra ilk 6 filtre çıkışı toplamları \$0e veya 0.237V değerini geçtikten sonra ses kaydına geçilecek şekilde ses kaydı programı yazıldı. Bu duruma göre sessiz harfle başlayan 0-9 arası sayıların ilk harfleri iptal edilmiş oldu.

Filtre merkez frekansları ve kazançlarının değiştirilmesiyle ses tanıma oranlarına bakılarak en iyi filtre bankası kurulmaya çalışıldı. Bu duruma göre filtrelerin merkez frekansları ve kazançları:

1-	500 Hz	Kazanç 1
2-	800 Hz	Kazanç 1
3-	1200 Hz	Kazanç 1
4-	1600 Hz	Kazanç 1
5-	2100 Hz	Kazanç 2
6-	2700 Hz	Kazanç 5
7-	3300 Hz	Kazanç 10
8-	4900 Hz	Kazanç 10

Ses tanımada ilk denemelerde kendi sesime bağlı olarak oluşturulan şablon'lar kullanılarak ses tanıma oranları, değişik filtre grublarına bakılarak çıkartıldı. Mikrofondan girilen sesin süresi ve enerjisi değiştikçe ses tanıma oranları düştü. Kişiyeye bağımlı ses tanıma oranı sayılar iyi bir şekilde söylendiğinde % 100'e varan sonuçlar alındı. Farklı kişilerde deneme yapmak için 10 kişinin ses kaydı yapıldı. Bu farklı kişilerde sesin süresi, vurgusu ve enerjisi eşit tutulmaya çalışıldı. Kişiyeye bağımlı ve bağımsız ses tanıma oranları, sıfır'dan dokuz'a kadar olan sayılar için aşağıdaki tabloda verilmiştir.

Kelime	Deneme	Kişiye Bağımlı sayısı Tanıma oranı	Kişiden Bağımsız sayısı Tanıma Oranı
0	6	83.3 %	10 90.0 %
1	6	100.0 %	10 90.0 %
2	6	100.0 %	10 100.0 %
3	6	100.0 %	10 100.0 %
4	7	100.0 %	10 100.0 %
5	7	100.0 %	10 90.0 %
6	6	83.3 %	10 90.0 %
7	6	83.3 %	10 80.0 %
8	6	100.0 %	10 80.0 %
9	6	100.0 %	10 100.0 %
GENEL	62	96.7 %	10 92.0 %

Ses tanınmasında filtre çıkışlarına göre daha iyi sonuçlar alabilmek için özellikle otomatik kazanç kontrolünün olması gerekir. Filtre sayılarının arttırılması ve düşük enerjiye sahip sessiz harflerin daha iyi tanınması için yüksek frekanslardaki filtre'lerin kazançlarının arttırılması ses tanıma oranlarının artmasını sağlar.

Dinamik Programlama metodu ile ses tanınması sablon sayısı, vektörlerin sayısı ve vektör elemanlarının sayısına göre tanıma süresi değişmektedir. Tanıma süresi yaklaşık 3 ile 7 sn arasında değişmektedir. Bilgisayarda " Coprocessor" kullanılması durumunda ses tanıma süresi daha da kısaltılabilir.

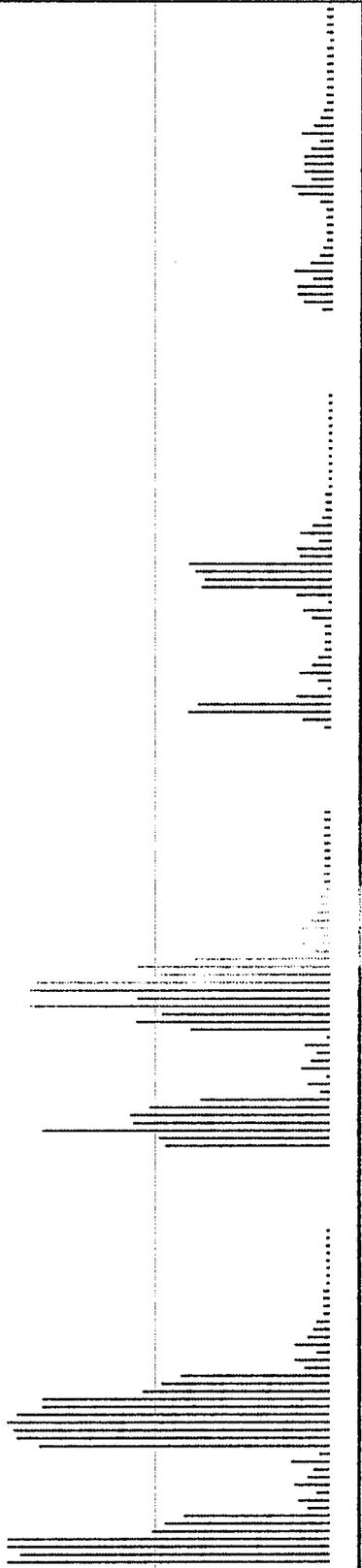
## 8.0 KAYNAKLAR DIZINI

1. Trends in Speech Recognition - Wayne, Lea - Prentice Hall, 1980
2. Digital Processing of Speech Signals - Lawrence R. Rabiner, Ronald W. Schafer, 1984
3. Electronic Speech Recognition Techniques, Technology and Applications - Geoff Bristow, 1986
4. Micro, June 1987
5. Byte, 1987
6. Linear Databook

**EK-1**

0-9 ARASI SAYILARIN ÖRNEK FİLTRE ÇIKIŞLARI

Sifir



F5

F6

F7

F8



BIR

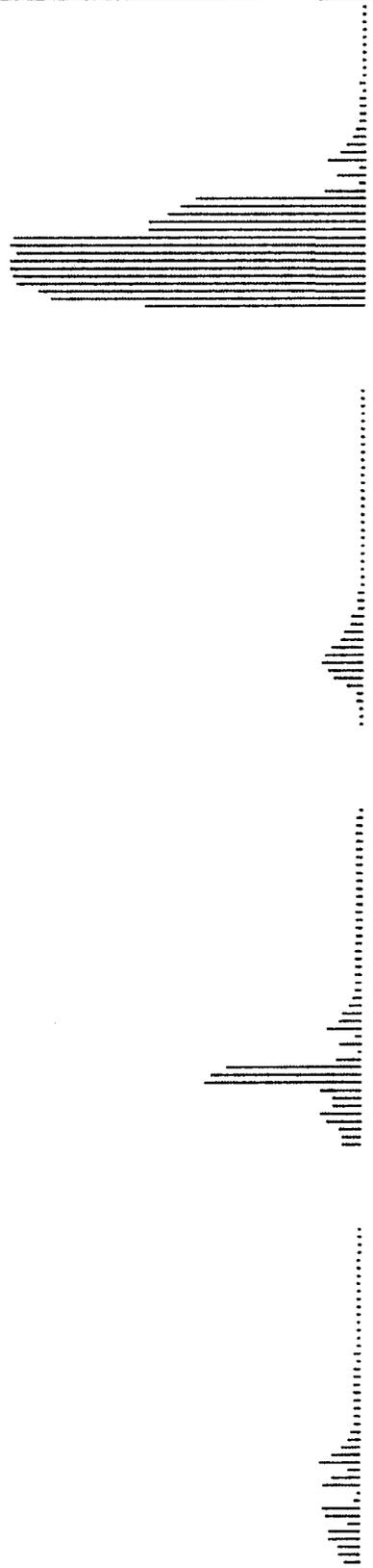


F5

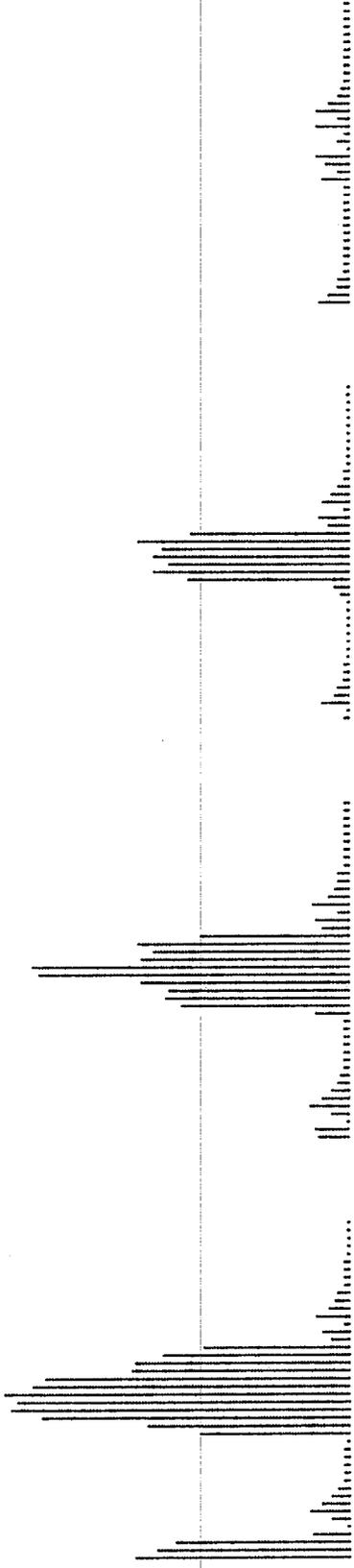
F6

F7

F8



iki

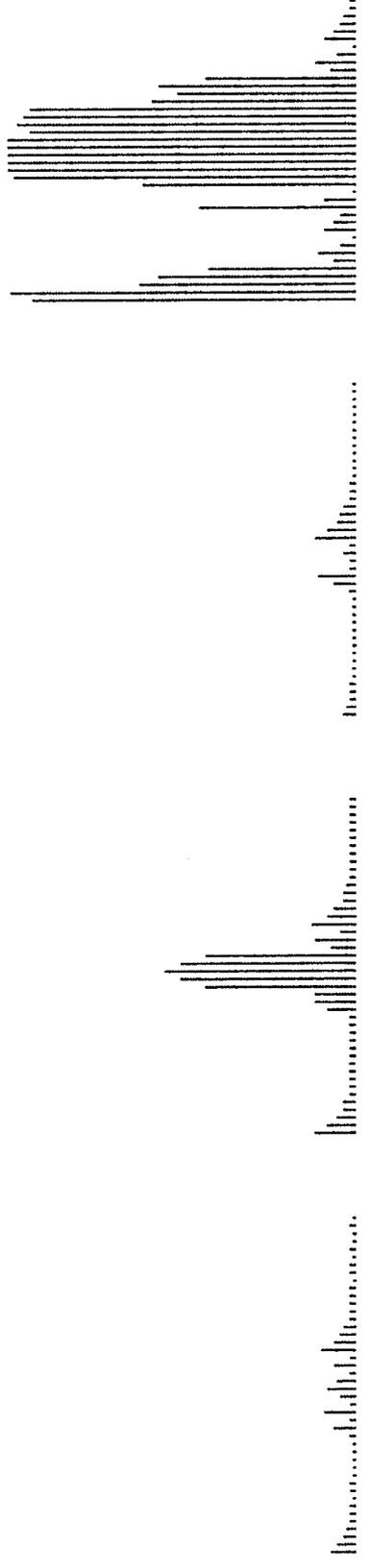


F5

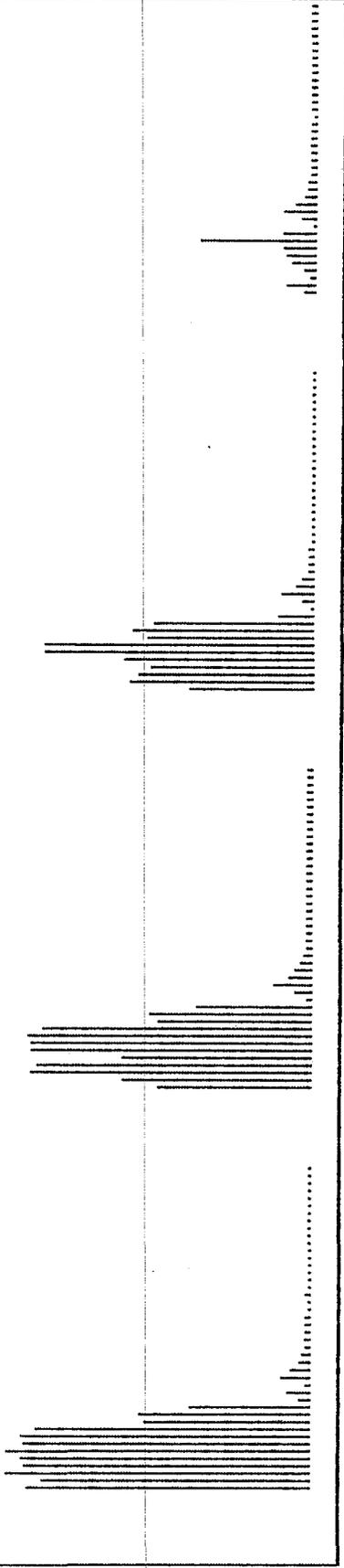
F6

F7

F8



ÜÇ

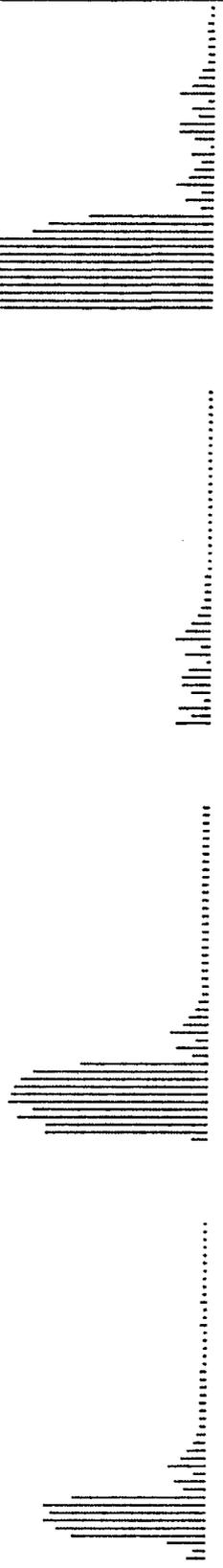


F5

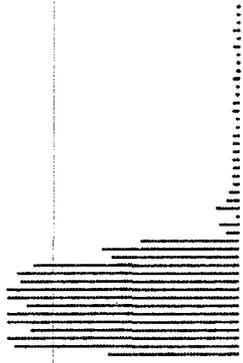
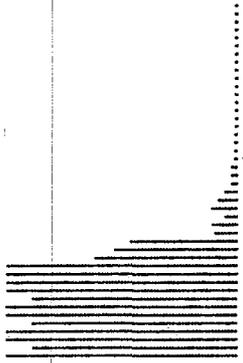
F6

F7

F8



DÖRT

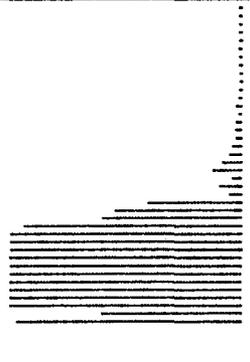
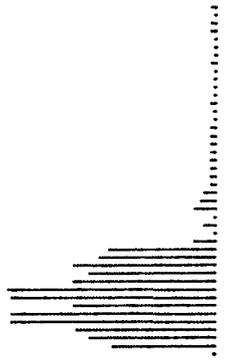
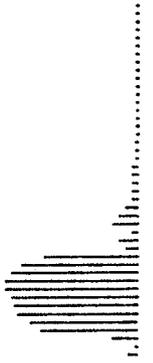


F5

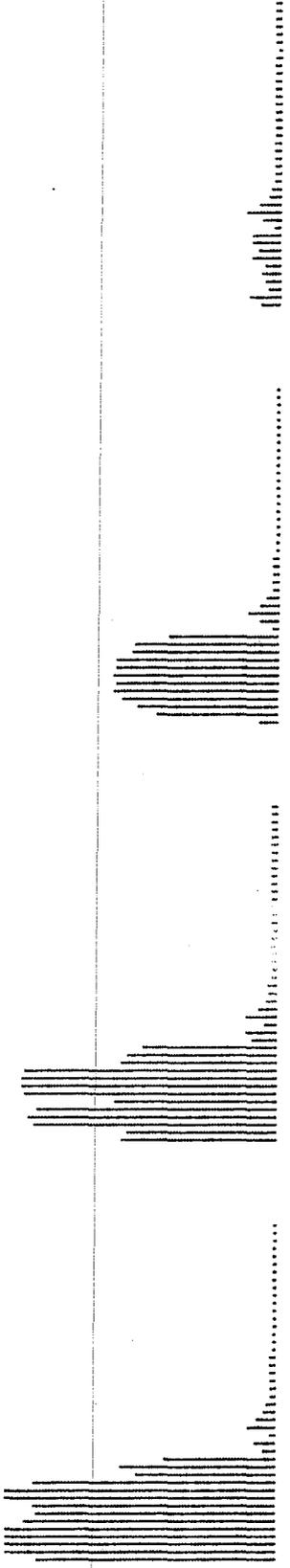
F6

F7

F8



BES



F5

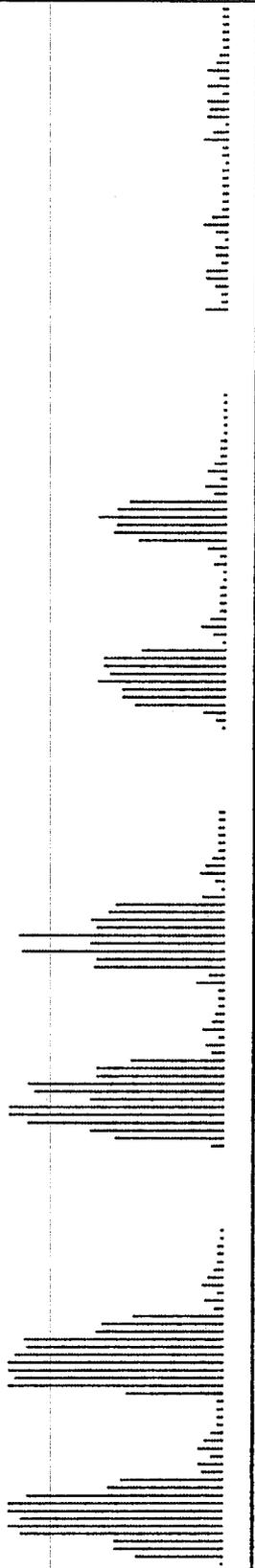
F6

F7

F8



ALTI



F5

F6

F7

F8



YEDI



F5

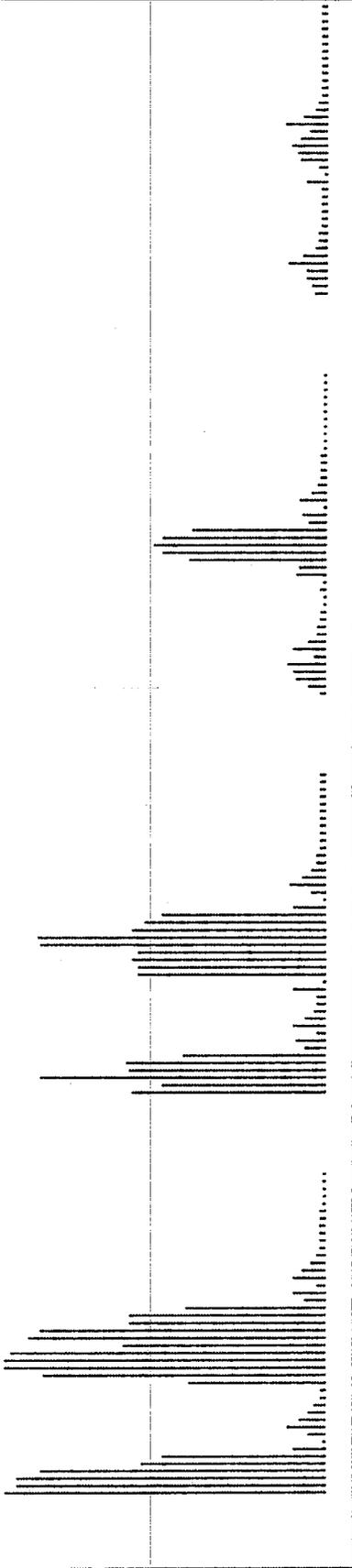
F6

F7

F8



SEKİZ

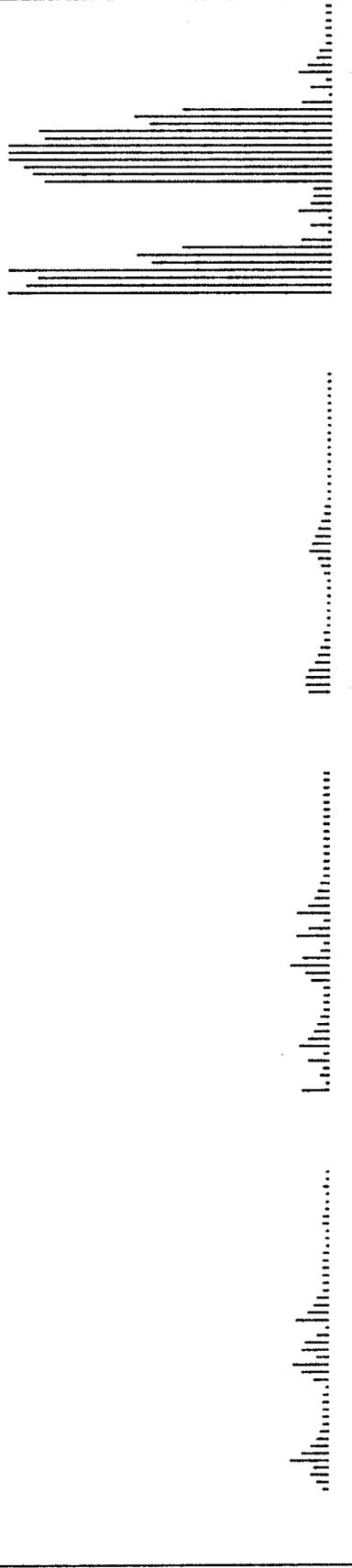


F5

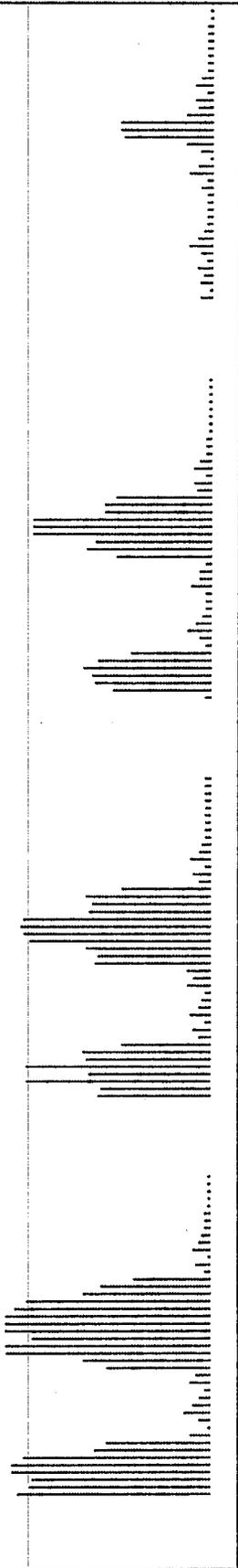
F6

F7

F8



DOKUZ

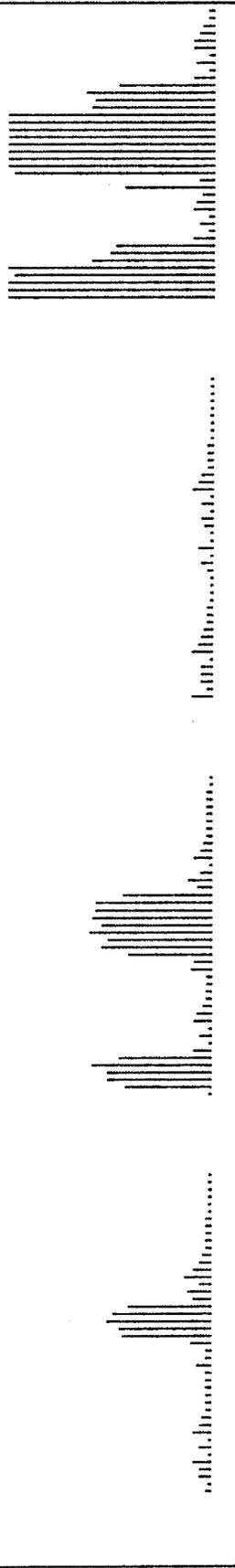


F5

F6

F7

F8



**EK-2**

PROGRAMLAR

```
program speech_analysis;
uses dos, crt, graph, srecord;
```

```
type
```

```
outext = string[25];
fntyp = string[11];
reclin = record
    temp1: integer;
    temp2: integer;
    temp3: integer;
    temp4: integer;
    temp5: integer;
    temp6: integer;
end;
```

```
var
```

```
tempfile: file of reclin;
temprec: reclin;
grp:array[0..265] of integer;
x:array[1..44,1..6] of integer;
y:array[0..11,1..44,1..6] of integer;
fndm:array[0..11] of longint;
i,k,j1,j2,m,barline,l1line,nup:integer;
ch,rp:char;
Done,done1,done2,done_end,done_end1:boolean;
choose:integer;
mo1,mo2,vo1,vo2:string[12];
y1,y2:integer;
file:string[8];
filename:fntyp;
```

```
procedure updatemp;
```

```
var
```

```
i, j, pnr, pnrr, pnrup:integer;
```

```
begin
```

```
Assign(tempfile,'TEMPLATE.DAT');Reset(tempfile);
clrscr;
gotoxy(10,10);j:=0;
write('Enter number of word to be prepared template ');
readln(pnr);
while pnr in [0..11] do
begin
    pnrr:=pnr*44;pnrup:=(pnr+1)*44-1;
    for i:=pnrr to pnrup do
    begin
        Seek(tempfile,i);read(tempfile,temprec);
        with temprec do
        begin
            write(' temp1 = ', temp1);temp1:=grp[j];
            write(' temp2 = ', temp1);temp2:=grp[j+1];
            write(' temp3 = ', temp1,' ',j);temp3:=grp[j+2];
            write(' temp4 = ', temp1);temp4:=grp[j+3];
            write(' temp5 = ', temp1);temp5:=grp[j+4];
            writeln(' temp6 = ', temp1);temp6:=grp[j+5];
        end;
        j:=j+6;
        seek(Tempfile,i);
        write(Tempfile,temprec);
    end;
end;
```

```

    pnr:=12;
end;
Close(tempfile);
end;

procedure prep_temp;
var i,j,k,pnr:integer;

begin
Assign(tempfile,'TEMPLATE.DAT');Reset(tempfile);
clrscr;
for k:=0 to 11 do
begin
writeLn(' ',k,'th template is being prepared..');
for j:=1 to 44 do
begin
pnr:=k*44+j-1;
Seek(tempfile,pnr);read(tempfile,temprec);
with temprec do
begin
Y[k,j,1]:=temp1;
Y[k,j,2]:=temp2;
Y[k,j,3]:=temp3;
Y[k,j,4]:=temp4;
Y[k,j,5]:=temp5;
Y[k,j,6]:=temp6;
end;
end;
end;
Close(tempfile);
writeLn('template matrix prepared.');
```

```

end;

procedure disptemp;
var dt,cc:integer;

begin
clrscr;
gotoxy(10,10);write('Enter number of the word to be displayed. ');
read(k);
for dt:=1 to 44 do
for cc:=1 to 6 do
grp[(dt-1)*6+cc-1]:=y[k,dt,cc];
end;
end;

procedure findn(var nn:integer);
var jii:integer;

begin
jii:=265;
repeat
jii:=jii-1;
until (grp[jii]+grp[jii-1]+grp[jii-2]+grp[jii-3] > 15) or (jii=6);
nn:=trunc(jii/6-0.5);
end;

procedure findm;
var tmp:array[0..265] of integer;
fm, pn, jl, jjk:integer;

```

```

begin
Assign(tempfile,'TEMPLATE.DAT');Reset(tempfile);
for fm:=0 to 11 do
begin
pn:=0;writeln('fm.....',fm);
for jl:=(fm*44) to ((fm+1)*44-1) do
begin
Seek(tempfile,jl);read(tempfile,temprec);
with temprec do
begin
tmp[pn]:=temp1;
tmp[pn+1]:=temp2;
tmp[pn+2]:=temp3;
tmp[pn+3]:=temp4;
tmp[pn+4]:=temp5;
tmp[pn+5]:=temp6;
end;
pn:=pn+6;
end;
jkk:=264;
repeat
jkk:=jkk-1;
until (tmp[jkk]+tmp[jkk-1]+tmp[jkk-2]+tmp[jkk-3] > 15) or (jkk=6);
fndm[fm]:=trunc(jkk/6);
end;
close(tempfile);
end;
end;

procedure normalization;
var
ll, dd, kk:integer;
value, norm:real;
temp:longint;
begin
findn(nup);while nup=0 do
begin
srec5c;
findn(nup);
end;
temp:=0;writeln('nup...',nup);
for dd:=0 to 264 do ((nup*6) )
begin
grp[dd]:=abs(mem[$6000:$4000+dd]);
temp:= temp+mem[$6000:$4000+dd];
end;
if temp=0 then
begin
writeln('there is no data.');
```

```

ar
d:array[0..11,1..44,1..44] of integer;
cost:array[0..11] of longint;
i, j, k, sayi, kk, dd, cst, ll, q, d1, d2, d3 : integer;
m, n: integer;

begin
  findn(n);
  for kk:=1 to n do
    begin
      for dd:=1 to 6 do
        begin
          ll:=(kk-1)*6+dd-1;
          x[kk, dd]:=grp[ll];
        end;
      end;
    end;
  (normalization);
  writeln('n.....',n);
  writeln('normalization finished..');
  for k:=0 to 9 do
    begin
      m:=fndm[k];
      i:=n; j:=m;
      d1:=0;
      for q:=1 to 6 do
        begin
          d[k, i, j]:=trunc(sqr((x[i, q]-y[k, j, q])/10))+d1;
          d1:=d[k, i, j];
        end;
      cost[k]:=D[k, n, m];
      repeat
        d1:=0; d2:=0; d3:=0;
        for q:=1 to 6 do
          begin
            d[k, i-1, j]:=trunc(sqr((x[i, q]-y[k, j, q])/10))+d1;
            d[k, i, j-1]:=trunc(sqr((x[i, q]-y[k, j, q])/10))+d2;
            d[k, i-1, j-1]:=trunc(sqr((x[i, q]-y[k, j, q])/10))+d3;
            d1:=d[k, i-1, j]; d2:=d[k, i, j-1]; d3:=d[k, i-1, j-1];
          end;
        if (D[k, i-1, j]<D[k, i-1, j-1]) and (D[k, i-1, j]<=D[k, i, j-1]) then
          begin
            cost[k]:=cost[k]+D[k, i-1, j];
            i:=i-1;
          end
        else
          begin
            if (D[k, i, j-1]<D[k, i-1, j-1]) and (D[k, i, j-1]<=D[k, i-1, j]) then
              begin
                cost[k]:=cost[k]+D[k, i, j-1];
                j:=j-1;
              end
            else
              begin
                cost[k]:=cost[k]+d[k, i-1, j-1];
                i:=i-1; j:=j-1;
              end;
            end;
          end;
        until (i=1) or (j=1);
    end;
  end;
end;

```

```

        if (i>5) or (j>5) then
            cost[k]:=50000;
end:
cst:=10000;sayi:=15;
for k:=0 to 9 do
    begin
        (cost[k]:=trunc(cost[k]/(n+fndm[k])+0.5);)
        writeln('cost[',k,']= ',cost[k]);
        if cost[k]<=cst then
            begin
                cst:=cost[k];
                sayi:=k;
            end;
        end;
    if sayi=15 then writeln('pattern could not be found..')
        else
            writeln('pattern belongs to ',sayi,' ');
end;

procedure graphics;
var graphd,graphm,cntr,xcor,sbs,flt:integer;
    maxX,maxY,maxY2:integer;
begin
    DetectGraph(graphd,graphm);Initgraph(graphd,graphm,'c:\tp');
    maxX:=getmaxX;maxY:=getmaxY;maxY2:=trunc(maxY/2);
    Rectangle(1,10,maxX,maxY2);Rectangle(1,maxY2+11,maxX,maxY);
    setviewport(10,10,maxX,maxY,true);
    xcor:=20;sbs:=6;
    outtextxy(20,10,'F1');outtextxy(230,10,'F2');outtextxy(440,10,'F3');
    outtextxy(20,maxY2+10,'F4');outtextxy(230,maxY2+10,'F5');
    outtextxy(440,maxY2+10,'F6');
    for flt:=1 to 3 do
        begin
            for cntr:=1 to 44 do
                begin
                    xcor:=xcor+4;
                    moveto(xcor,maxY2-20);
                    lineto(xcor,maxY2-20-trunc(grp[cntr*6-sbs]/2+0.5));
                end;
            xcor:=xcor+40;sbs:=sbs-1;
        end;
    setviewport(10,maxY2+11,maxX,maxY,true);
    xcor:=20;
    for flt:=1 to 3 do
        begin
            for cntr:=1 to 44 do
                begin
                    xcor:=xcor+4;
                    moveto(xcor,maxY2-20);
                    lineto(xcor,maxY2-20-trunc(grp[cntr*6-sbs]/2+0.5));
                end;
            xcor:=xcor+40;sbs:=sbs-1;
        end;
    repeat rp:=readkey;until (rp=#27);
    (repeat until keypressed;)
    ClearDevice;closegraph;
    Restorecrtmode;
end;
procedure rev_write(var y1,y2:integer;

```

```

                                outext1,outext2:outext);

begin
  gotoxy(35,y1);write(outext1);
  textcolor(black);textbackground(white);
  gotoxy(35,y2);write(outext2);
  textcolor(white);textbackground(black);
end;

procedure create_data_file(var filename:fntyp);
var
  i, j:integer;
begin
  Assign(tempfile,filename);
  Rewrite(tempfile);clrscr;
  j:=0;
  for i:=1 to 44 do
  begin
    with temprec do
    begin
      temp1:=mem[$6000:$4000+j];
      temp2:=mem[$6000:$4000+j+1];
      temp3:=mem[$6000:$4000+j+2];
      temp4:=mem[$6000:$4000+j+3];
      temp5:=mem[$6000:$4000+j+4];
      temp6:=mem[$6000:$4000+j+5];
    end;
    j:=j+6;
    write(Tempfile,temprec);
  end;
  Close(tempfile);
end;

procedure load_data_file(var filename:fntyp);
var i, j:integer;
begin
  Assign(tempfile,filename);
  Reset(tempfile);clrscr;
  j:=0;
  for i:=1 to 44 do
  begin
    seek(tempfile,i-1);read(tempfile,temprec);
    with temprec do
    begin
      memw[$6000:$4000+j]:=temp1;
      memw[$6000:$4000+j+1]:=temp2;
      memw[$6000:$4000+j+2]:=temp3;
      memw[$6000:$4000+j+3]:=temp4;
      memw[$6000:$4000+j+4]:=temp5;
      memw[$6000:$4000+j+5]:=temp6;
    end;
    j:=j+6;
  end;
end;

```

```

Close(tempfile);
nd;

procedure main;

label lab.main_lab;

begin
main_lab:
clrscr;
done2:=False;done_end:=false;
extcolor(black);textbackground(white);
gotoxy(35,5);write('1- RECORD');
extcolor(white);textbackground(black);
gotoxy(33,2);write('SPEECH ANALYSIS');
gotoxy(1,3);for i:=1 to 80 do write(#205);
gotoxy(1,1);write(#218);for i:=1 to 78 do write(#196);write(#191);
gotoxy(1,24);write(#192);for i:=1 to 78 do write(#196);write(#217);
gotoxy(1,2);for i:=1 to 22 do writeln(#179);
for i:=2 to 23 do begin gotoxy(80,i);write(#179);end;
gotoxy(1,3);write(#198);gotoxy(80,3);write(#181);
gotoxy(1,21);write(#195);for i:=1 to 78 do write(#196);write(#180);
gotoxy(35,7);write('2- LOAD');
gotoxy(35,9);write('3- SAVE');
gotoxy(35,11);write('4- GRAPHIC');
gotoxy(35,13);write('5- UPDATE TEMPLATE');
gotoxy(35,15);write('6- SPEECH RECOGNIZATION');
gotoxy(35,17);write('7- DISPLAY TEMPLATE');
gotoxy(35,19);write('0- QUIT');
gotoxy(3,22);
write('                Records speech data which you will say to microphone.
');

gotoxy(43,5);
repeat
lab:
ch:=ReadKey;
case ch of
#0:
begin
ch:=ReadKey;
case ch of
#72:  (up)
begin
if wherey=19 then
begin
y1:=19;y2:=17;
gotoxy(3,22);write('                Display any template which you
t.                ');
rev_write(y1,y2,'0- QUIT','7- DISPLAY TEMPLATE');
goto lab;
end;
if wherey=17 then
begin
y1:=17;y2:=15;
gotoxy(3,22);write(' Recognize unknown speech pattern by using Dyna
programming method. ');

```

```

    rev_write(y1,y2,'7- DISPLAY TEMPLATE','6- SPEECH RECOGNIZATION');
    goto lab;
end;
if wherey=15 then
begin
    y1:=15;y2:=13;
    gotoxy(3,22);write('
t.                Updates any template which you
                ');
    rev_write(y1,y2,'6- SPEECH RECOGNIZATION','5- UPDATE TEMPLATE');
    goto lab;
end;
if wherey=13 then
begin
    y1:=13;y2:=11;
    gotoxy(3,22);write('
outputs.        Display recorded or loaded 6 inte
                ');
    rev_write(y1,y2,'5- UPDATE TEMPLATE','4- GRAPHIC');
    goto lab;
end;
if wherey=11 then
begin
    y1:=11;y2:=9;
    gotoxy(3,22);write('
ed.            Saves speech data which recorded
                ');
    rev_write(y1,y2,'4- GRAPHIC','3- SAVE');
    goto lab;
end;
if wherey=9 then
begin
    y1:=9;y2:=7;
    gotoxy(3,22);write('
le.           Loads speech data from a g
                ');
    rev_write(y1,y2,'3- SAVE','2- LOAD');
    goto lab;
end;
if wherey=7 then
begin
    y1:=7;y2:=5;
    gotoxy(3,22);write('
o microphone. Records speech data which you wil
                ');
    rev_write(y1,y2,'2- LOAD','1- RECORD');
    goto lab;
end;
if wherey=5 then
begin
    y1:=5;y2:=19;
    gotoxy(3,22);write('
                Exits program.
                ');
    rev_write(y1,y2,'1- RECORD','0- QUIT');
    goto lab;
end;
end;

B0:    (down)
begin
if wherey=19 then

```

```

begin
  y1:=19;y2:=5;
  gotoxy(3,22);write('
a microphone.      ')
  rev_write(y1,y2,'0- QUIT','1- RECORD');
  goto lab;
end;
if wherey=17 then
begin
  y1:=17;y2:=19;
  gotoxy(3,22);write('
                                Exits program.
                                ');
  rev_write(y1,y2,'7- DISPLAY TEMPLATE','0- QUIT');
  goto lab;
end;
if wherey=15 then
begin
  y1:=15;y2:=17;
  gotoxy(3,22);write('
                                Display any template which you
                                ');
  rev_write(y1,y2,'6- SPEECH RECOGNIZATION','7- DISPLAY TEMPLATE');
  goto lab;
end;
if wherey=13 then
begin
  y1:=13;y2:=15;
  gotoxy(3,22);write(' Recognize unknown speech pattern by using Dyna
ogramming method.  ');
  rev_write(y1,y2,'5- UPDATE TEMPLATE','6- SPEECH RECOGNIZATION');
  goto lab;
end;
if wherey=11 then
begin
  y1:=11;y2:=13;
  gotoxy(3,22);write('
                                Updates any template which you
                                ');
  rev_write(y1,y2,'4- GRAPHIC','5- UPDATE TEMPLATE');
  goto lab;
end;
if wherey=9 then
begin
  y1:=9;y2:=11;
  gotoxy(3,22);write('
                                Display recorded or loaded 6 inter
outputs.      ');
  rev_write(y1,y2,'3- SAVE','4- GRAPHIC');
  goto lab;
end;
if wherey=7 then
begin
  y1:=7;y2:=9;
  gotoxy(3,22);write('
                                Saves speech data which recorded
ed.          ');
  rev_write(y1,y2,'2- LOAD','3- SAVE');
  goto lab;
end;
if wherey=5 then
begin

```

```

y1:=5;y2:=7;
gotoxy(3,22);write('
le. ');
rev_write(y1,y2,'1- RECORD','2- LOAD');
goto lab;
end;
end; {label 80}
end; {case 2}
end; {label 0}
#13: begin
case wherey of
5:
begin
window(1,1,80,25);clrscr;
clrscr;gotoxy(20,20);write('Press any key to speech recording..');
repeat until keypressed;
srec5c;
normalization;
graphics;
done2:=true;done_end:=true;
end;
7:
begin
window(1,1,80,25);clrscr;
gotoxy(20,20);write('Enter FILE NAME : ');
repeat;readln(filen);until filen<>' ';
filename:=concat(filen,'.DAT');
load_data_file(filename);
normalization;
done2:=true;done_end:=true;
end;
9:
begin
window(1,1,80,25);clrscr;
gotoxy(20,20);write('Enter FILE NAME : ');
repeat;readln(filen);until filen<>' ';
filename:=concat(filen,'.DAT');
create_data_file(filename);
done2:=true;done_end:=true;
end;
11:
begin
graphics;
done2:=true;done_end:=true;
end;
13:
begin
window(1,1,80,25);clrscr;
updatemp;
findm;
prep_temp;
done2:=true;done_end:=true;
end;
15:
begin
window(1,1,80,25);clrscr;rp:='s';
rectem;repeat rp:=readkey;until (rp=' ');
done2:=true;done_end:=true;

```

Loads speech data from a g

```

end;
17:
begin
window(1,1,80,25);clrscr;
disptemp;graphics;
done2:=true;done_end:=true;
end;
19:
begin
done2:=true;
end;
end;
end;
end;
until done2;

if done_end then goto main_lab;
clrscr;
end;

begin
for i:=0 to 264 do
memw[$6000:$4000+i]:=0;
end;
findm;
prep_temp;
for i:=1 to 44 do
for k:=1 to 6 do
grp[(i-1)*6+k-1]:=y[0,i,k];
end;
end;
main;
window(1,1,80,25);clrscr;
writeln(' Speech Analysis Program, 1990, Created by A.EROL');
end.

```

```

title      '8 bit ses kaydi .....(sr.asm)'
;*****
.stack    100h
include  dos.inc
include  bios.inc
.model   small

program   segment
slem      proc           far
assume   cs:program,ss:program,ds:program,es:program
push    bx
push    cx
push    dx
push    si
push    bp
push    ds
push    es
mov     di,sp
mov     si,ss:[di+10]
push   si
mov     ax,6000h
mov     ds,ax
mov     cx,2b00h      ;sets speech record period
mov     bx,0
mov     dx,0126h
mov     al,92h
out     dx,al
mov     dx,0124h      ; resets all integrators
mov     ax,0080h      ;
out     dx,al        ;
mov     dx,100
reset:   dec          dx
         jnz         wreset
         nop
         nop
osp:     mov          cx,2b00h
         mov          bx,0
ero:     mov          dx,0124h
         mov          al,cl
         out          dx,al
         mov          al,08h
         add          al,cl
         out          dx,al
         mov          al,40h
         out          dx,al
         mov          al,0
         out          dx,al
         mov          dx,0126h
         mov          al,9ah
         out          dx,al
         mov          dx,0124h
ga:      in           al,dx
         shl          al,1
         shl          al,1
         shl          al,1
         jnc          aga
         mov          dx,0126h
         mov          al,92h

```

```

out      dx,al
mov      dx,0124h
mov      al,10h
out      dx,al
mov      dx,0122h
in       al,dx
add      bx,ax
mov      dx,0124h
mov      al,0
out      dx,al
inc      cl
mov      al,cl
sub      al,05
jnz     zero
mov      al,80h
out      dx,al
nop
mov      dx,32h
bek1:   dec      dx
        jnz     bek1
mov      dx,0124h
mov      al,0
out      dx,al
delay:  mov      dx,16a8h
        dec      dx
        jnz     delay
sub      bx,0090h
jc       nosp
mov      bx,4000h
mov      cx,2b00h
jmp      basla
loop:   mov      cl,0
basla:  mov      dx,0124h
mov      al,cl
out      dx,al
mov      al,08h
add      al,cl
out      dx,al
mov      al,40h
out      dx,al
mov      al,0
out      dx,al
mov      dx,0126h
mov      al,9ah
out      dx,al
mov      dx,0124h
ine:    in       al,dx
        shl     al,1
        shl     al,1
        shl     al,1
        jnc     yine
mov      dx,0126h
mov      al,92h
out      dx,al
mov      dx,0124h
mov      al,10h
out      dx,al
mov      dx,0122h

```

```

in          al,dx
mov         ss:[bx],al
mov         dx,0124h
mov         al,0
out         dx,al
inc        bx
inc        cl
mov         al,cl
sub         al,06
jnz        basla
mov         al,80h
out         dx,al
mov         dx,2000h
bek2:      dec         dx
jnz        bek2
mov         dx,0124h
nop
mov         al,0
out         dx,al
mov         dx,5
delay1:   dec         dx
jnz        delay1
dec        ch
jnz        gloop
pop         si
mov         di,sp
mov         ss:[di+10],si
pop         es
pop         ds
pop         bp
pop         si
pop         di
pop         dx
pop         cx
pop         bx
@exit    0
isllem   endp
program  ends
end      isllem

```

```
unit srecord;

interface
procedure srec5c;

implementation
procedure srec5c;

begin
  inline(
    $53/
    $51/
    $52/
    $57/
    $56/
    $55/
    $1e/
    $06/
    $8b/$fc/
    $36/$8b/$75/$0a/
    $56/
    $b8/$00/$60/
    $8e,$d8/
    $b9/$00/$2b/
    $bb,$00/$00/
    $ba/$26/$01/
    $b0/$92/
    $ee/
    $ba,$24/$01/
    $b8/$80/$00/
    $ee/
    $ba/$00/$10/
    $4a/
    $75,$fd/
    $90/
    $90/
    $b9,$00/$2b/
    $bb,$00/$00/
    $ba,$24/$01/
    $8a,$c1/
    $ee/
    $b0/$08/
    $02,$c1/
    $ee/
    $b0/$40/
    $ee/
    $b0/$00/
    $ee/
    $ba/$26/$01/
    $b0/$9a/
    $ee/
    $ba,$24/$01/
    $ec/
    $d0/$e0/
    $d0/$e0/
    $d0/$e0/
    $73,$f7/
    $ba/$26/$01/
```

\$b0/\$92/  
\$ee/  
\$ba/\$24/\$01/  
\$b0/\$10/  
\$ee/  
\$ba/\$22/\$01/  
\$ec/  
\$03/\$d8/  
\$ba/\$24/\$01/  
\$b0/\$00/  
\$ee/  
\$fe/\$c1/  
\$8a/\$c1/  
\$2c/\$06/  
\$75/\$bd/  
\$b0/\$80/  
\$ee/  
\$90/  
\$ba/\$32/\$00/  
\$4a/  
\$75/\$fd/  
\$ba/\$24/\$01/  
\$b0/\$00/  
\$ee/  
\$ba/\$a0/\$16/  
\$4a/  
\$75/\$fd/  
\$81/\$eb/\$0e/\$00/  
\$72/\$9b/  
\$bb/\$00/\$40/  
\$b9/\$00/\$3b/  
\$eb/\$03/  
\$90/  
\$b1/\$00/  
\$ba/\$24/\$01/  
\$8a/\$c1/  
\$ee/  
\$b0/\$08/  
\$02/\$c1/  
\$ee/  
\$b0/\$40/  
\$ee/  
\$b0/\$00/  
\$ee/  
\$ba/\$26/\$01/  
\$b0/\$9a/  
\$ee/  
\$ba/\$24/\$01/  
\$ec/  
\$d0/\$e0/  
\$d0/\$e0/  
\$d0/\$e0/  
\$73/\$f7/  
\$ba/\$26/\$01/  
\$b0/\$92/  
\$ee/  
\$ba/\$24/\$01/  
\$b0/\$10/

(threshold value : 0007H )

```
$ee/  
$ba/$22/$01/  
$ec/  
$88/$07/  
$ba/$24/$01/  
$b0/$00/  
$ee/  
$43/  
$fe/$c1/  
$8a/$c1/  
$2c/$06/  
$75/$bc/  
$b0/$80/  
$ee/  
$ba/$0a/$00/  
$4a/  
$75/$fd/  
$ba/$24/$01/  
$90/  
$b0/$00/  
$ee/  
$ba/$a8/$0f/      ( wait for 20ms delay   )  
$4a/                (                               )  
$75/$fd/           ( '$0e'                               )  
$fe/$cd/  
$75/$a0/  
$5e/  
$8b/$fc/  
$36/$89/$75/$0a/  
$07/  
$1f/  
$5d/  
$5e/  
$5f/  
$5a/  
$59/  
$5b/  
$b8/$00/$4c);
```

end;

```
begin  
  writeln('ok...');  
end.
```

## **EK-3**

PROGRAM MENÜ ÇIKTISI

---

SPEECH ANALYSIS

---

- 1- RECORD
- 2- LOAD
- 3- SAVE
- 4- GRAPHIC
- 5- UPDATE TEMPLATE
- 6- SPEECH RECOGNIZATION
- 7- DISPLAY TEMPLATE
- 0- QUIT

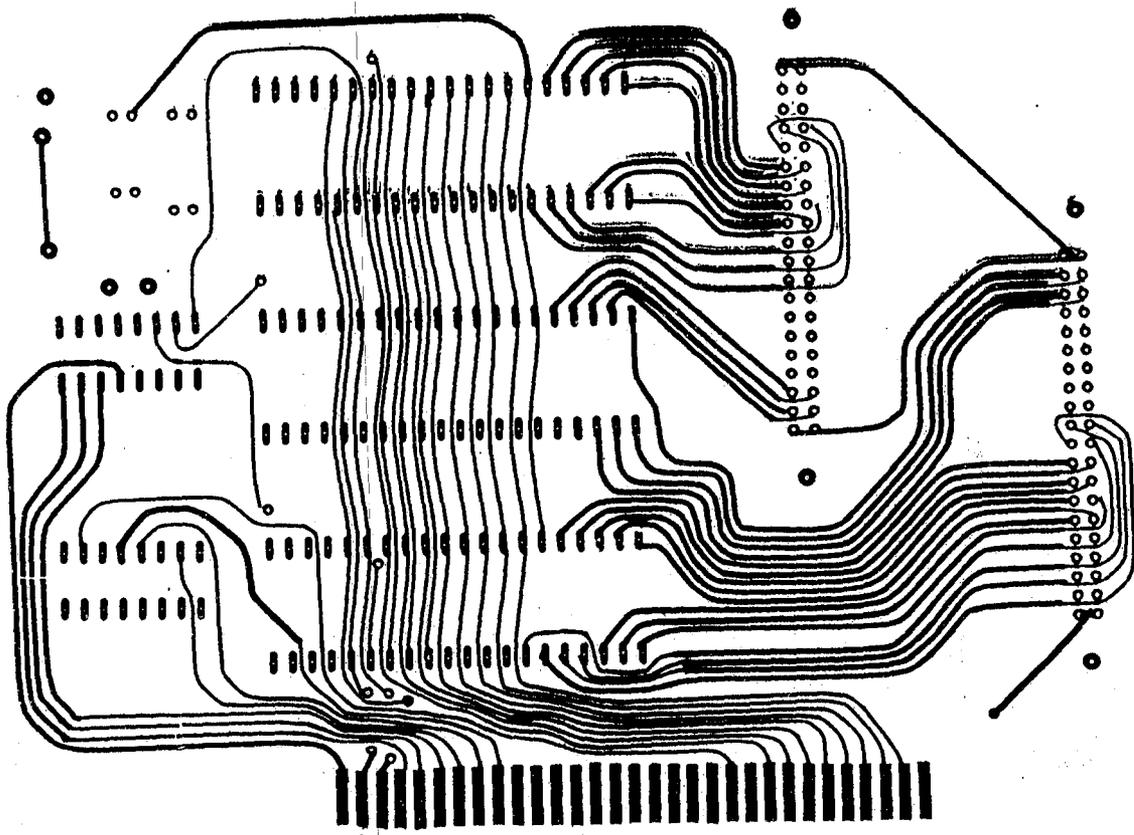
---

Records speech data which you will say to microphone.

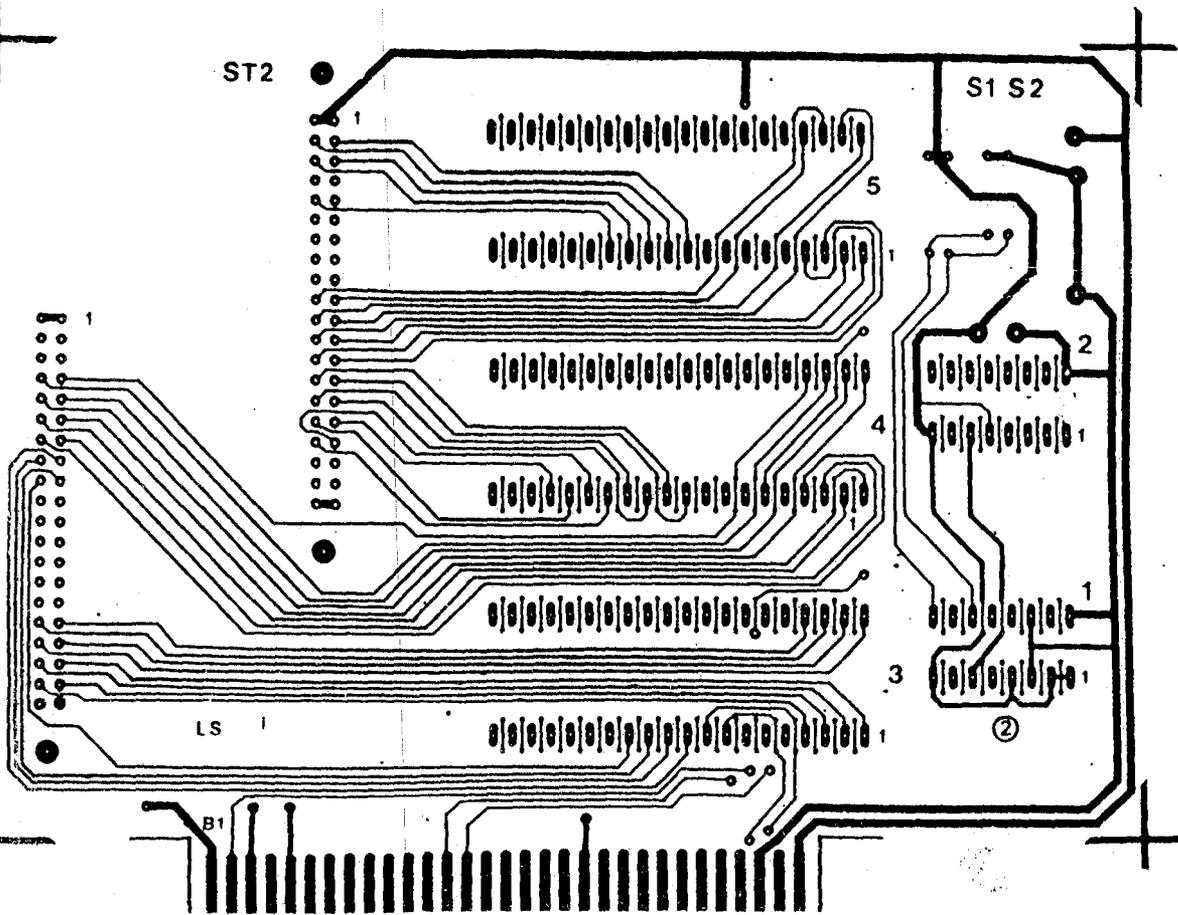
---

## **EK-4**

**ÇOKLU GİRİŞ - ÇIKIŞ KARTININ BASKI DEVRE ŞEMASI**



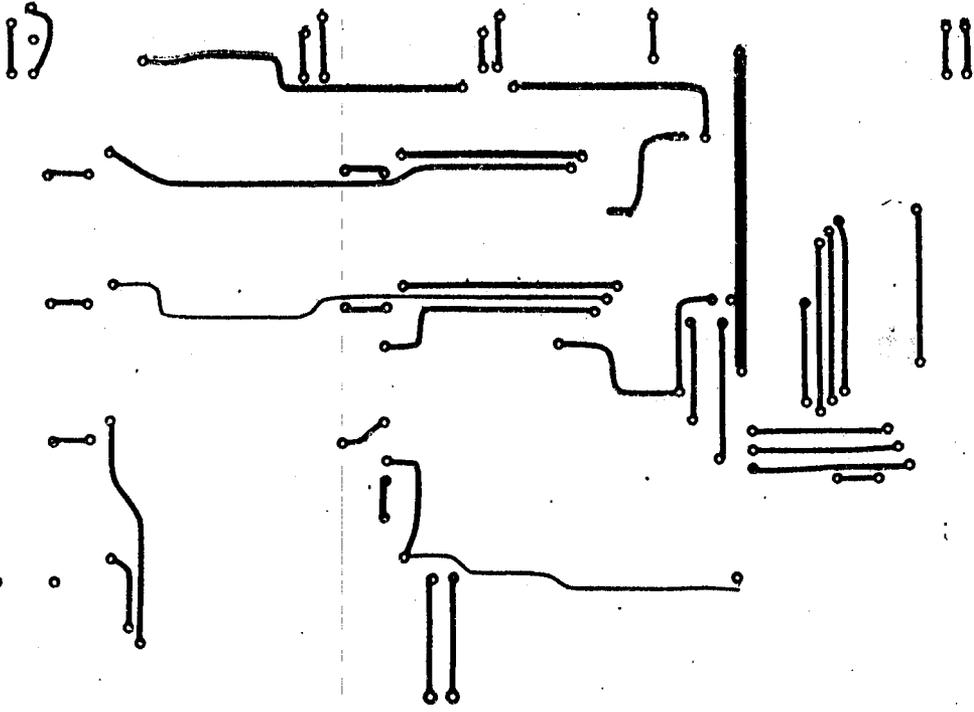
*Baskı Devrenin Üstten Görünüsü*



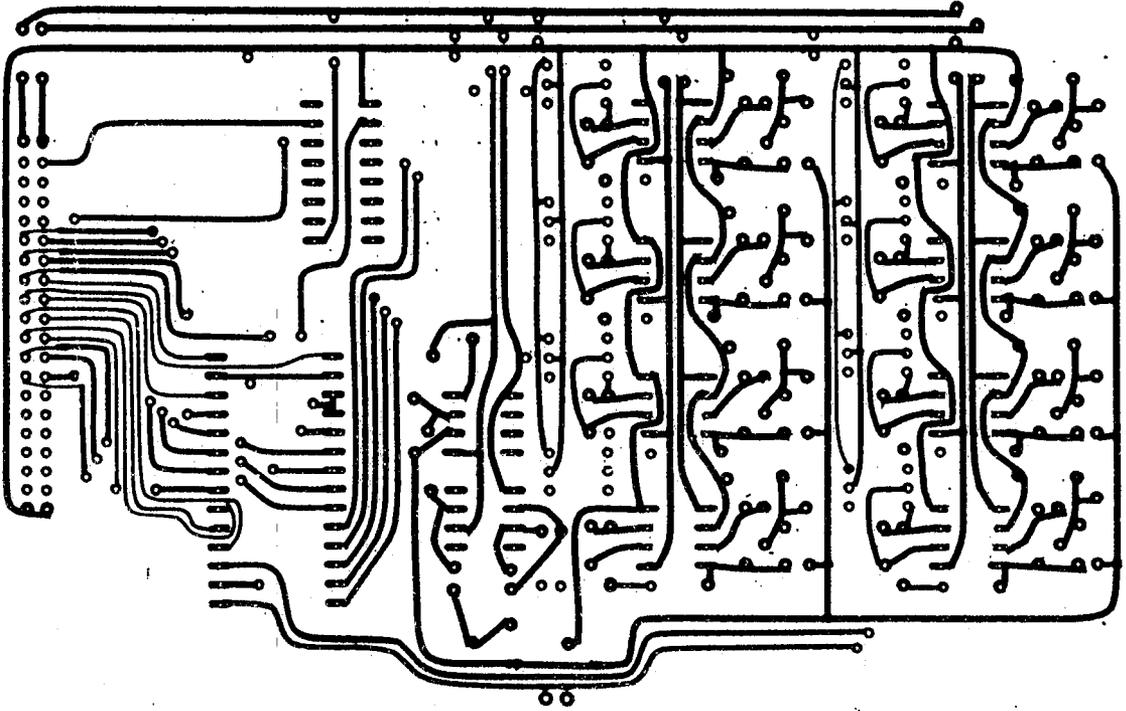
*Baskı Devrenin Alttan Görünüsü*

## **EK-5**

ANALOG SENTEZLEYİCİ KARTI BASKI DEVRE ŞEMASI



*Baskı Devrenin Üstten Görünüsü*



*Baskı Devrenin Alttan Görünüsü*