

SIFIRDAN DOKUZA KADAR SAYILAR VE AÇ , KAPAT ,  
KANAL KELİMELERİNİ TANIYAN KONTROL SİSTEMİ

Gökhan DINDİŞ

Anadolu Üniversitesi  
Fen Bilimleri Enstitüsü  
Lisans Üstü Yönetmeliği Uyarınca  
Elektrik-Elektronik Müh. Anabilim Dalında  
YÜKSEK LİSANS TEZİ  
Olarak Hazırlanmıştır.

Danışman: Dr. Erdener İLDİZ

Eylül - 1987

Gökhan Dındış'ın YÜKSEK LİSANS tezi olarak hazırladığı " Sıfırdan Dokuza Kadar Sayılar ve Aç , Kapat , Kanal Kelimelerini Tanıyan Kontrol Sistemi " başlıklı bu çalışma , jürimizce lisansüstü yönetmeliğinin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir.

16.9.1987

Üye : Dr. Erdener İLDİZ

Üye : Prof. Dr. Atalay BARKANA

Üye : Doç. Dr. Atıla BARKANA

---

Fen Bilimleri Enstitüsü Yönetim Kurulu'nun  
28.9.1987... gün ve .155./1..... sayılı kararıyla  
onaylanmıştır.

Rüstem KAYA  
Enstitü Müdürü

---

## ÖZET

Ses ve ses tanıma çalışmalarına aktif bir şekilde girmek , aynı zamanda bunu uygulamaya dökmek amacıyla bir ses tanıma sisteminin yapımı düşünülmüştür. Hedef kelimeler olarak , sıfır-dokuz arası sayılar ve aç , kapat , kanal komutları seçilmiştir. Sistemin bu ses girdilerini tanıyarak , ilgili cihaz için açma-kapama kontrolü yapması sağlanmıştır.

Ses tanıma sistemi , tip olarak boşluklarla sınırlandırılmış (izoleli) kelime tanıyan bir sistemdir ve sıfır geçiş analizi kullanmaktadır. Parametreler tek bir konuşmacıya göre ayarlanmıştır. Kullanılan algoritmalar , uygun bir mikrobilgisayar yapıldığında sürekli kelime tanıyan bir sistem için aynen kullanılabilir.

SUMMARY

Speech and speech recognition studies <sup>are</sup> is aimed. It is also thought that a speech recognition system design is accomplished. As target words ; the numbers between zero and nine , on , off and channel orders are chosen. The system is designed so that it recognizes these voice data and it controls the related equipment by on and off.

The speech recognition system is an Isolated-<sup>W</sup> Word Recognition System. It uses zero crossing analyse <sup>is-</sup> method. The parameters are calibrated according to a single speaker. The algorithms used can be implemented to a system which recognizes words continuously if an appropriate microcomputer is designed. ??

## TESEKKUR

Makineye ses tanıma zekası verilerek sesle kontrol edilmesi modern yaşantıda artık bir ihtiyaç niteliğindedir. Özellikle elleri , ayakları tutmayan hastalar için iyi bir yardımcı olabilecek robot makinelerin kontrolü , çocuk oyuncakları , güvenlik gerektiren özel uygulamalar , evlerde , arabalarda , iş yerlerinde kullanılabilecek lüks cihazlar bu ihtiyaçların başlıcaları olarak sayılabilir. Bu arada kullanımı yüksek sorumluluk gerektiren uçaklarda pilotlara oldukça yardımcı olabilecek uygulamaları gözardı etmemek gerekir. Ses tanıma çalışmalarının hedefi belki de tek deyişle insanoglunun yaşantısını kolaylaştırmak...

Böylesine popüler bir çalışmada yardımlarını esirgemeyen ve disiplin içinde çalışmaya teşvik eden tez yöneticim sayın Hv. Bnb. Dr. Erdener ILDIZ'a , konuyla ilgili yardımlarırdan dolayı bölüm başkanımız sayın Prof. Dr. Atalay BARKANA ve sayın Doç. Dr. Atila BARKANA'ya teşekkür ederim.

## İÇİNDEKİLER

	Sayfa
ÖZET .....	iv
SUMMARY .....	v
ŞEKİLLER DİZİNİ .....	ix
1. GİRİŞ .....	1
2. MAKİNE İLE SES TANIMA ÇALIŞMALARI .....	3
2.1. Geçmişte Yapılan Teorik Çalışmalar ....	3
2.2. Geçmişte Gerçekleştirilen Sistemler ...	7
2.2.1. Kelime Tanıyan Sistemler .....	7
2.2.2. Sürekli Ses Tanıyan Sistemler ...	9
2.2.3. Konuşma Anlayan Sistemler .....	10
3. SES TANIMA ÇALIŞMALARINA GİRİŞ .....	11
3.1. Yapılacak Sistemin Seçimi .....	11
3.2. Türkçe Seslerin Tabiatının İncelenmesi .	12
3.2.1. Zaman Ekseninde Analiz .....	12
3.2.2. Frekans Ekseninde Analiz .....	18
3.3. Yapılan İncelemelere Göre Sistemin	
Tutarlılığı .....	24
4. SİSTEMİN TASARIMI - DONANIM VE YAZILIM .....	25
4.1. Analog Donanım Çalışmaları .....	25
4.1.1. Ses Kuvvetlendiricisi .....	27
4.1.2. Band Geçiren Filtre .....	28
4.1.3. Yüksek Geçiren Filtre .....	29
4.1.4. Zarf Seçici ve Karşılaştırıcı ..	30

## İÇİNDEKİLER ( devam )

Sayfa

4.1.5. Sayısal Seviye Ayarlama ve Koruyucu .....	31
4.2. Amstrad CPC6128 Bilgisayarı İçin Uyum devresi .....	31
4.3. Cihaz Kontrolü İçin Gerekli Ara Devre ve Çalışma Sekli .....	32
4.4. Yazılım Çalışmalarına Giriş .....	33
4.5. Sıfır Geçiş Verilerinin Ön İşlemesi ...	34
4.6. Ses Tanımda Kullanılabilecek Modellemeler .....	36
4.7. Tanıma Alt Programlarına Ait Akış Şemaları .....	39
5. SONUÇ VE ÖNERİLER .....	52
KAYNAKLAR DİZİNİ .....	54

## EKLER

1. Tipik Ses Dalga Şekilleri
2. 8255 Paralel Uyum Devresi
3. Bilgisayar Programları

## SEKİLLER DİZİNİ

	Sayfa
3.1 Türkçede tipik 'bir' kelimesi .....	13
3.2 Türkçede tipik 'iki' kelimesi .....	13
3.3 Türkçede tipik 'üç' kelimesi .....	13
3.4 Türkçede tipik 'dört' kelimesi .....	14
3.5 Türkçede tipik 'beş' kelimesi .....	14
3.6 Türkçede tipik 'altı' kelimesi .....	14
3.7 Türkçede tipik 'yedi' kelimesi .....	15
3.8 Türkçede tipik 'sekiz' kelimesi .....	15
3.9 Türkçede tipik 'dokuz' kelimesi .....	15
3.10 Türkçede tipik 'sıfır' kelimesi .....	16
3.11 Türkçede tipik 'aç' kelimesi .....	17
3.12 Türkçede tipik 'kapat' kelimesi .....	17
3.13 Türkçede tipik 'kanal' kelimesi .....	17
3.14 'a' sesi .....	20
3.15 'e' sesi .....	20
3.16 'i' sesi .....	21
3.17 'ı' sesi .....	21
3.18 'ö' sesi .....	22
3.19 'o' sesi .....	22
3.20 'ü' sesi .....	23
3.21 'u' sesi .....	23
4.1 Analog Donanım blok şeması .....	26
4.2 Ses kuvvetlendiricisi .....	27
4.3 Band geçiren filtre .....	28
4.4 Yüksek geçiren filtre .....	29



## SEKİLLER DİZİNİ ( devam )

	Sayfa
4.5 Zarf seçici ve karşılaştırıcı .....	30
4.6 Sayısal seviye ayarı ve koruma devresi ...	31
4.7 Uyum devresi .....	31
4.8 Cihaz kontrol devresi .....	32
4.9 Sistemin çalışma modeli .....	37
4.10 Kelime tanıma durum modeli .....	37
4.11 'bir' Kelimesine ait akış şeması .....	39
4.12 'iki' Kelimesine ait akış şeması .....	40
4.13 'üç' Kelimesine ait akış şeması .....	41
4.14 'dört' Kelimesine ait akış şeması .....	42
4.15 'beş' Kelimesine ait akış şeması .....	43
4.16 'altı' Kelimesine ait akış şeması .....	44
4.17 'yedi' Kelimesine ait akış şeması .....	45
4.18 'sekiz' Kelimesine ait akış şeması .....	46
4.19 'dokuz' Kelimesine ait akış şeması .....	47
4.20 'sıfır' Kelimesine ait akış şeması .....	48
4.21 'aç' Kelimesine ait akış şeması .....	49
4.22 'kapat' Kelimesine ait akış şeması .....	50
4.23 'kanal' Kelimesine ait akış şeması .....	51

## 1. GİRİŞ

İnsanoğlu eskiden beri pek çok işi , makinelere uzaktan sesle kumanda etmek suretiyle yaptırabilmeyi hayal etmiştir. 'Ali Baba ve Kırk Haramiler' masalındaki 'açıl susam açıl' denildiğinde açılan kapıyı buna güzel bir örnek olarak vermek mümkündür. Bilimsel ses tanıma çalışmaları ise bu tür hayal ürünleri ile başlamış ve gelişmiştir.

Su an için ses tanıma çalışmaları insan ve makine diyalogunu oldukça geliştirmiş bir düzeydedir. Bugün konuyla uğraşan bilim adamları daha çok , ses tanıma sistemlerindeki kelime sayısını artırmak , maliyet ve hata oranını düşürmek peşindedirler. Daha ilginç kelime tanımaktan ziyade makine zekasını artırarak kelimeleri birleştirip bundan anlam çıkarabilecek türde sistemler geliştirmeye çalışmaktadırlar.

Ses tanıma konusunda çalışan yabancı bilim adamlarının çalışmalarına bakıldığında , genelde ingilizce veya benzeri dillerde uygulamalar yapıldığı gözlenmektedir. Bu çalışmada , Türkçe ses tanıma konusuna eğilinmiş ve mikrobilgisayar kontrollü bir ses tanıma sistemi gerçekleştirilmiştir. Sistem tanıdığı kelimelere göre ilgili bir cihazı açıp kapayabilecek

özelliğindedir. Tanınacak kelimeler kod olarak ta kullanılabilir türden olması bakımından 'sıfır' - 'dokuz' arası sayılar ve cihaz kontrolü için 'aç', 'kapat', 'kanal' komutları olarak düşünülmüş ve uygulamaya bu şekilde geçilmiştir.

Maliyetinin düşüklüğü ve parçalarının bulunabilirliği sıfır geçiş analizi yapabilecek bir sistemin tasarlanmasını gerektirmiştir. Böylece analiz metodu da belirlenmiştir. Öncelikle tanınacak seslerin karakteristik dalga biçimleri ve <sup>F</sup>fourier spektrumları incelenmiş daha sonra uygulamaya geçildiğinde frekans bileşenlerinin yoğun olarak dağıldıkları bölgeler filtrelerle ayrılmıştır. Filtre çıkışlarından elde edilen pozitif kenar sıfır geçişler arasındaki süreler ölçülerek, bulunan periyotlara göre istatistiksel bir tanıma algoritmi hazırlanmıştır.

Söz edilen inceleme, geliştirme ve uygulama basamakları AMSTRAD CPC6128 bilgisayarını ile yapılmıştır. Kullanılan algoritma 3-4 MHz. arası bir saat ile çalışan tek bir mikrobilgisayarın hız bakımından altından kalkabileceği şekilde düzenlenmiştir.

## 2. MAKİNE İLE SES TANIMA ÇALIŞMALARI

Makine ile ses tanıma çalışmalarına ilk olarak 1950 'li yıllarda rastlanmaktadır. Halen kullanılmakta olan bir çok ses analizi tekniği 1950'li ve 1960'lı yıllarda geliştirilmişlerdir. Ünlü ses bilimcisi Flanagan'ın kitabı , bu çalışmaları , konu ile ilgili tanımları ve kullanılan ses analiz tekniklerini (kısa-zaman spektrumu , frekans analizi , sıfır geçiş analizi) özetlemektedir.

Bugün ses tanıma araştırmalarında ortak kullanılan teknikler eskiye göre pek değişmemiştir. Bazı darbe kodlu modülasyon ( PCM ) teknikleri , sayısal elektronığın gelişmesiyle uygulamaya dönük hale gelebilmiştir. Doğrusal öngörülü kodlama ( LPC ) ve kısa-zaman spektrumlarının çabuk hesaplanabildiği hızlı <sup>F</sup>fourier dönüşümleri ( FFT ) gibi metotlar geliştirilmiştir.

### 2.1. Geçmişte Yapılan Teorik Çalışmalar

1950 ve 1960'lı yıllarda yapılan daha çok teoriye dayalı çalışmalar 1970'li yıllarda oldukça hızlanmıştır. İlk defa 1971 yılında Advanced Research Projects Agency'nin aktif çalışmaları ile ses tanıma ve anlama çalışmaları yoğunluk kazanmaya başlamıştır. Çeşitli sürekli ses tanıma sistemleri geliştirilmiş

ve sergilenmiştir .Bu alanda akustik ses bilimlerinin , yazım ( syntax ) ve anlam ( grammer ) bilimlerinin rolü net bir şekilde görülmüştür. Bilgisayar programları göstermiştir ki verilen girişler belirsiz , hatalarla dolu ve gramatiklikten uzak olabilmektedir. Ancak ele alınan kısıtlı , basit gramatik yapılara sahip belirli sayıda kelimenin sadece tek bir konuşmacıyla ve kontrol edilebilen gürültüsüz bir ortamda tanıtımı sağlanabilmiştir. Günümüze kadarki çalışmalar genelde kontrol edilmemiş çevre şartlarında , farklı konuşmacılar için kısıtsız bir ses tanıma üzerinedir.

Bu alandaki araştırmalarda ana amaç sistemlerin ve tekniklerin makineye ses kontrol girişi için geliştirilmesidir. 1950'li yıllardan bu yana , birkaç yıl öncesine kadar yapılan sistemler kısmi olarak başarılı olmuş , fakat kullanılan teknikler istenilen sonuç için yeterli olamamıştır. Bu sistemlerin kendilerine has zorlukları tablo 2.1'de verilmiştir. Zorlukları azaltacak çalışmalar ise halen sürmektedir.

izoleli kelime tanıma sistemleri günümüzde halen kullanılmaktadır. Bu yöntemde , kelimeler arasına sessizlikler konarak kelimelerin birbirinden ayrılması sağlanmaktadır. Böylece ardarda gelerek birleşebilecek kelimeleri ayırma zorluğu ortadan

kalkmakta aynı zamanda boşluklar sayesinde sisteme ek bir tanıma süresi kazandırılmaktadır.

Sürekli ses tanıma ve konuşma anlama için sistemler geliştirilse de ancak kısıtlı kelime sayısı için iyi sonuçlar alınabilmiştir. Sınırsız kelime hazinesine sahip, sürekli ses tanıma ve konuşma anlama sistemleri yapılabilir gözükse de epey zaman alacaktır. Sürekli ses tanıma sistemlerinde karşılaşılan güçlük kelimenin başının ve sonunun ayırilebilmesidir. Halbuki izoleli kelime tanıma sistemlerinde bu problem kelime arası boşluklarla çözülmüştür.

Sistemlerin kendilerine has özelliklerinden birisi kelime hazinesidir. Kelime sayısı arttıkça sistemlerin kullandığı metotlar değiştiği gibi tanı süreleri de artmaktadır. Maliyet ise sistemlerin günlük yaşantıda kabul edilebilir veya edilemezliğini ortaya koyar.

Sürekli konuşma anlama sistemlerinde, yazım ve anlam bilim devreye girer. Söylenenlerin düzgün ve gramatik olup olmadığı da incelenir. Bu tür sistemler, henüz gerçek anlamda, istenilen tasarımdan diğerlerine oranla daha uzaktadır.

Tablo 2.1 Ses tanıma sisteminde karşılaşılan zorluklar.

SİSTEM	KELİME SAYISI	ÖZEL BİLGİ	DİL	KONUŞMACI	ÇEVRE
İzoleli kelime tanıma	10-300	sınırlı kullan.	---	sisteme özel	---
Sürekli bağlantılı konuşma tanıma	30-500	sınırlı kullan.	sınırlı komut	sisteme özel	gürültüsüz
Sürekli bağlantılı konuşma anlama	100-2000	tam kullan.	ing. ve benzeri	bağımsız	---
Konuşulana yazan makine	1000-10000	sınırlı kullan.	ing. ve benzeri	sisteme özel	gürültüsüz
Limitsiz sürekli bağ. kon. anlama	sınırsız	tam kullan.	ing.	bağımsız	---
Limitsiz sürekli bağ. kon. tanıma	sınırsız	----	ing.	bağımsız	gürültüsüz

(---): Sistemine göre değişiklik gösterdiğini ifade etmektedir.

## 2.2. GemiŖte GerekleŖtirilen Sistemler

Ses tanıyan sistemlerin farklı tiplerinin yapıları hakkında fikir edinmek amacıyla uygulamaya konmuş sistemlere göz atmakta fayda vardır.

Ortak test verilerine karşılık deęişik sistemlerden deęişik ve pek çok yönden karşılaştırılabilir sonuç alınabilmektedir. Dizaynı mümkün oldukça çok sistem vardır ve bunların herhangi birini diğerleriyle karşılaştırıp 'bu en iyisidir' demek mümkün değildir. Ancak doğruluk , hızlılık , kelime hazinesi gibi açılardan bazı üstünlükleri karşılaştırmalar yapılabilir.

Geçmişte gerçekleştirilmiş sistemleri üç ana başlıkta toplamak yerinde olacaktır.

### 2.2.1. Kelime Tanıyan Sistemler

Bu tür sistemlere en tipik örnekler Itakura , Martin ve White adlı ses bilimcilere ait sistemlerdir. 10-300 Arası bir kelime grubuna ait kelimeyi bilinen bir şahıs için tanımaya çalışan bu sistemler %99 doğruluğa ulaşabilmektedir. Kelime veya konuşmacının deęişmesi halinde sistemin kelimeleri tanıyabilmesi için tekrar yeni konuşmacının ifadelerine göre düzenlenmesi gerekmektedir. Bu sistemler klasik patern tanıma metotlarını kullanmaktadır. İzlendiği yol verilen ifadelerdeki bazı parametre ve özellikleri referanslarla karşılaştırıp sonuca ulaşmaktadır.



Mikrofon veya konuşmacı değiştiğinde bazı parametreler de paralel olarak değişeceğinden çevredeki gürültü karakteristiklerinde de değişiklik yaratacaktır. Oluşan veya oluşabilecek gürültüler kaynaklarına göre değişik gruplara ayrılmış ve isimlendirilmişlerdir. Oda gürültüsü , mikrofon gürültüsü, konuşmacıdan kaynaklanan öksürük gürültüsü gibi. Bu gürültülerin bir kısmı karakteristikleri bilindiği için genlik normalizasyonuna tabi tutulabilmektedir. Örneğin White , filtre örneklerini toplam enerjiye bölerek değişik bir normalizasyon yapmıştır.

Itakura'nın kullandığı dogrusal öngörülü kodlama katsayıları yaklaşımı , Martin'in kullandığı band geçiren filtre çakışları , White'in kullandığı 1/3 oktavlık filtre dizisi olaya ne tür parametreler ile yaklaşıldığını göstermektedir.

Bir sistem , bilinmeyen bir paterni referans paternleriyle ancak belli bir hızda karşılaştırabilmektedir. Yani referans paternler arttıkça tanıma süresi de lineer olarak artacaktır. Dolayısıyla metot , 30 kelimelik referans grubuna sahip bir mikrobilgisayar sisteminde saniyede 200 bin ile 500 bin komut icra etmesine rağmen limit bir hız vermektedir. Kelimelerin artması birbirine benzer kelimelerin sayısını da artıracığından hata oranını yükseltecektir.

### 2.2.2. Sürekli Ses Tanıyan Sistemler

Geçmişte , bu metotla ses tanıyan sistemler için en tipik örnekler Carnegie-Mellon Üniversitesinde geliştirilen Hearsay-1 ve Dragon sistemi , MIT Lincoln laboratuvarlarında geliştirilen Lincoln sistemi ve IBM laboratuvarlarında geliştirilen IBM sistemidir.

Hearsay-1 konuşma anlama sistemi olarak geliştirilmiş ancak anlama işini yapabilmesi için gerekli zeka modülleri sonradan çıkarılmıştır. Oluşan yeni sistem ses tanıma sistemi olarak kullanılmıştır. Dragon ve Lincoln sistemleri ise doğrudan ses tanıma sistemleri olarak tasarlanmıştır. Bütün bu sistemler %97 'ye ulaşan bir doğruluk verebilmiştir.

Sürekli ses tanıyan sistemlerde , daha önce belirtilen kelime başı ve sonu ayırımı probleminin yanısıra akustik karakteristiklerin çok fazla değişiklik göstermesi yüzünden yüksek doğruluklara erişememe sorunu vardır. Buna rağmen normal bir şekilde konuşulan kelimeyi veya cümleyi tanıyabildiğinden , %99 doğruluğa ulaşabilen fakat boşluklar arasında söylenen kelimeleri tanıyan izoleli kelime tanıyan sistemlere göre tercih edilirler.

Sürekli ses tanıyan sistemlerde kullanılan bir metot seslerin sembollere dönüştürülmesidir. Bu şekilde gramatik olarak kelime başı ve sonu ayırımı

yapılabilmektedir. Dragon sistemi 10 ms. lik konuşma bölümleri içinde 12 genlik ve sıfır geçiş parametresinden mümkün 33 fonemik sembolü çıkarabilmektedir.

Lincoln sistemi hızlı sayısal işlemci ile LPC spektrumu çıkarmakta ve belirlediği formant frekanslarıyla sonuca ulaşmaktadır.

### 2.2.3. Konuşma Anlayan Sistemler

Konuşma anlayan sistemlere örnek olarak Carnegie-Mellon Üniversitesinde geliştirilen Speechlis , System Development Corporation'da geliştirilen VDMS ve Standford Research Enstitüsü'nde geliştirilen Hearsay-II 'yi verebiliriz. Bu sistemler anlam bilim ve yazın bilim gibi yapay zeka modülleriyle donatılmışlardır. Verilen zekanın mükemmelliğine göre değişik doğruluklara ulaşabilmektedirler. Konuşma anlayan sistemlerde ana problem çok iyi bir zekanın sisteme kazandırılabilmesidir. Sistem bu sayede çok iyi formda ve çok düzgün olmayan konuşmaları dahi , konuşmaya benzer öksürük , mırıltı gibi gürültülerden ayırabilmelidir. Bu konudaki çalışmalar daha uzun yıllar sürecek gibi görünmektedir.

### 3. SES TANIMA CALIŞMALARINA GIRIŞ

Kısım 2'de , daha önce yapılan ses analizi teknikleri ve sistemlerinden bahsedilmişti. Ses tanıması yapabilecek teknik ve sistemi düşünmeden önce sesleri incelemek , tanımak gerekir. Kuşkusuz ki sesleri tanımadan, makineye tanıma algoritmaları hazırlamak , tanıma zekası vermek olanaksızdır. Hepsinden önce bu çalışmalarda kullanmak üzere bir geliştirme sistemine ihtiyaç vardır. Böyle bir sistem olarak Anadolu Üniversitesi'nde 1987-Haziran döneminde hazırlanmış bulunan " AMSTRAD CPC6128 Bilgisayarında Ses Sinyallerinin İncelenebilmesi İçin Ara Devre ve Programların Hazırlanması " adlı bitirme ödevinden yararlanılmıştır.

#### 3.1. Yapılacak Sistemin Seçimi

Çalışmada ilk hedef sürekli ses tanıması yapabilecek bir sistem olduğundan , uzun matematiksel işlemler gerektirerek tanıma hızını düşürecek LPC , FFT gibi tekniklerden kaçınılmıştır. Bu teknikler aynı zamanda hızlı ve piyasada kolayca bulunamayan aritmetik işlemciler gerektirmektedir. Sözü edilen aritmetik işlemciler ayrıca sistemin maliyetini de artıracak niteliktedir. Sıfır geçiş analizi ise pratik ucuz ve kullanılabilir malzemeler piyasada kolayca bulunabilecek türdendir. Bu sebepler sıfır geçiş analizinin seçimini gerektirmiş ve incelemeler bu yönde sürdürülmüştür.

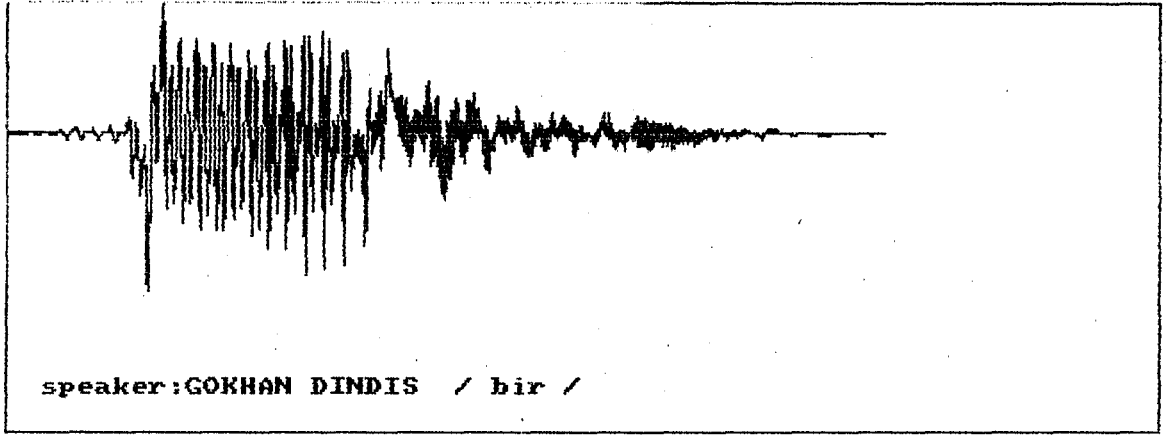
### 3.2. Türkçe Seslerin Tabiatının İncelenmesi

Ses analizine , öncelikle zaman alanında girmektedir. Zaman ekseninde inceleme , sesler hakkında kaba bazı fikirler verecektir. Hangi seslerde yüksek genlikli , hangilerinde düşük genlikli, hangilerinde titreşim daha fazla gibi sorulara cevap verebilmek için bu gereklidir.

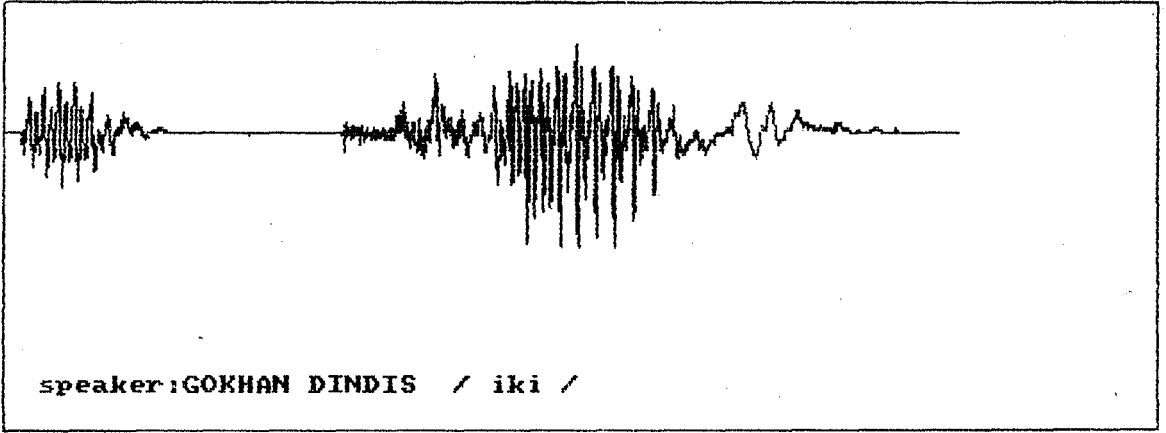
Daha sonra frekans analizine geçerek özellikle sesli harflere ait frekans bileşenlerini incelemek mantıklı olacaktır. Çünkü sesli harflerde oldukça kararlı ve belirgin frekans bileşenleri mevcuttur.

#### 3.2.1. Zaman Ekseninde Analiz

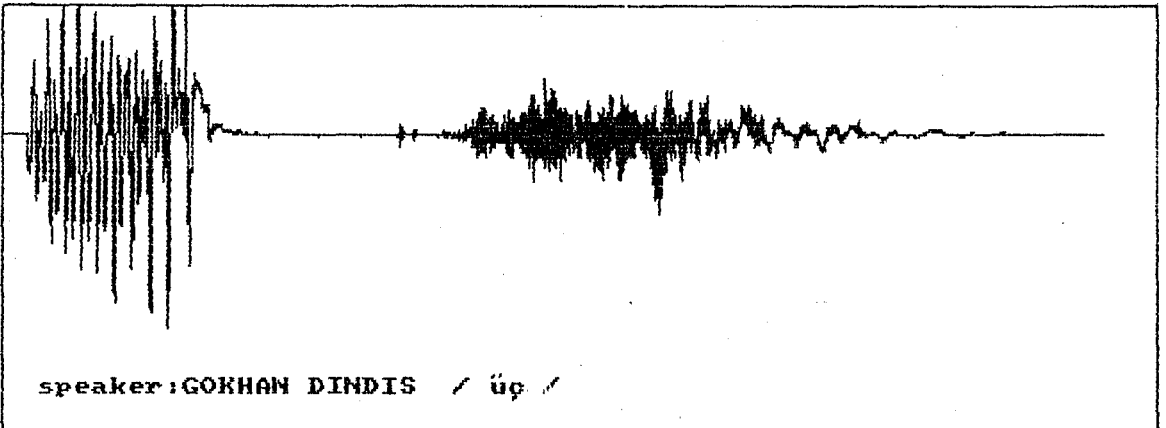
Hedef olarak seçilen onüç kelimeye ait dalga şekillerine ( şekil 3.1 ... 3.13 ) ilk bakışta , bazı karakteristik özellikler hemen göze çarpmaktadır. Bunlardan biri , ses dizilerinin grup grup oluşudur. İnsan ağzının bazı sesleri bir çırpıda arka arkaya çıkarmaması ve hece aralarında özellikle yaratılan anlamlı boşluklar bu grupları birbirinden ayıran iki önemli sebeptir. Göze çarpan ikinci özellik , sesli harflerin diğerlerine oranla daha yüksek genliğe sahip oluşudur. Sesli harfler çıkarılırken ses telleri titreşmekte ve ciğerlerden şiddetli bir şekilde hava gönderilmektedir.



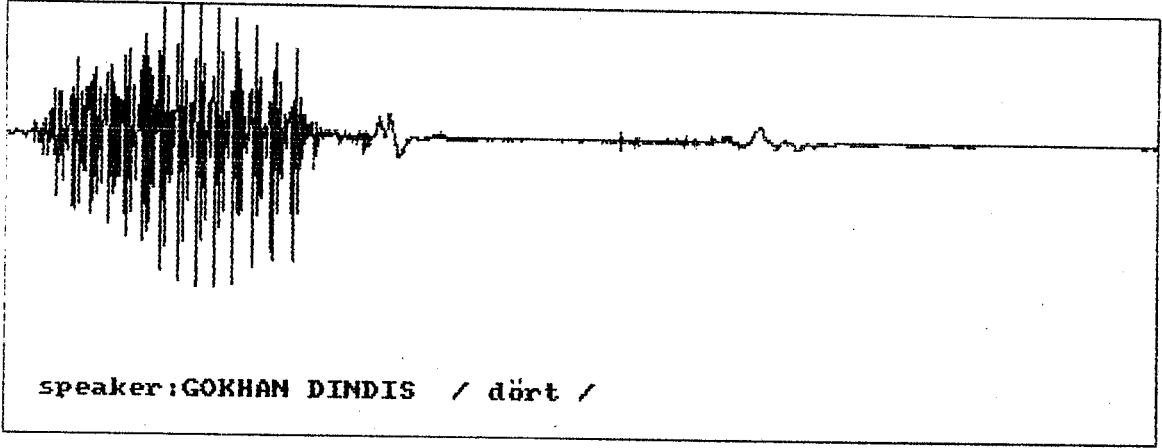
Sekil 3.1 Türkçede tipik 'bir' kelimesi



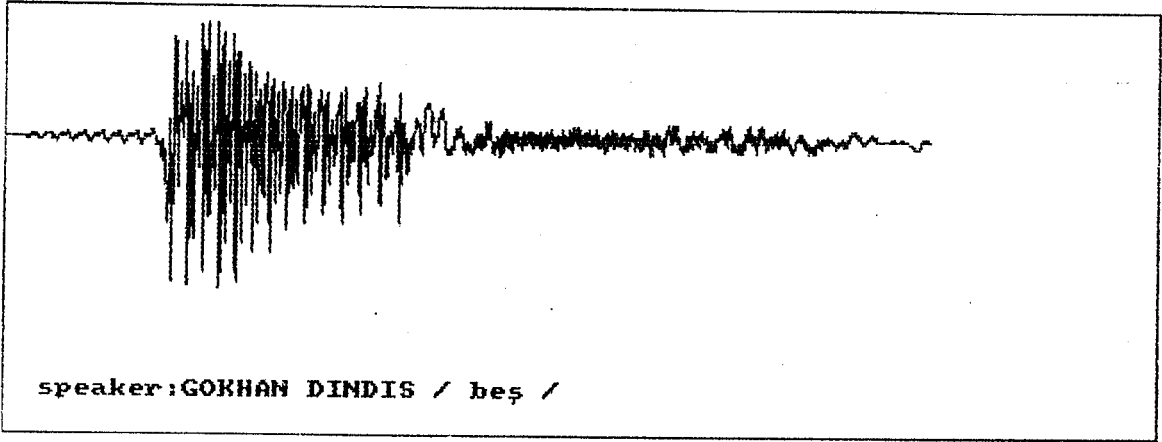
Sekil 3.2 Türkçede tipik 'iki' kelimesi



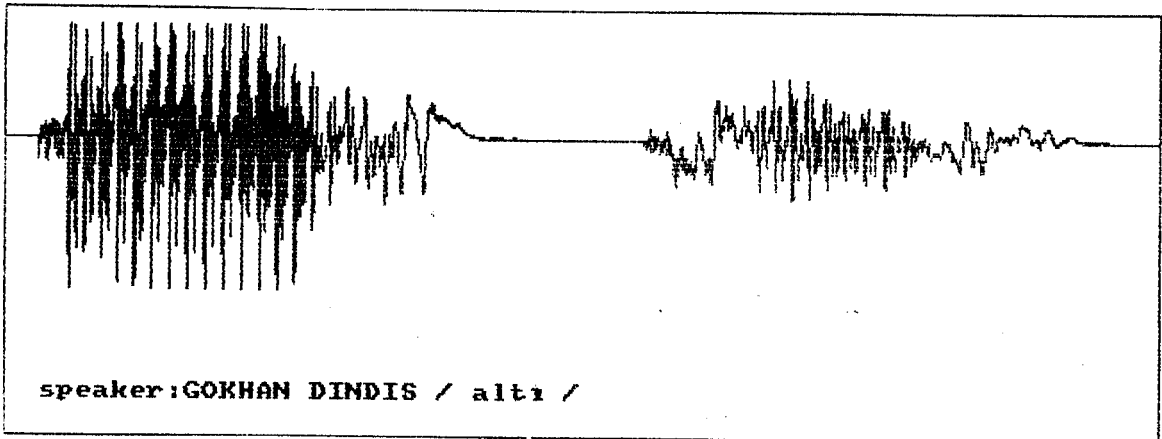
Sekil 3.3 Türkçede tipik 'üç' kelimesi



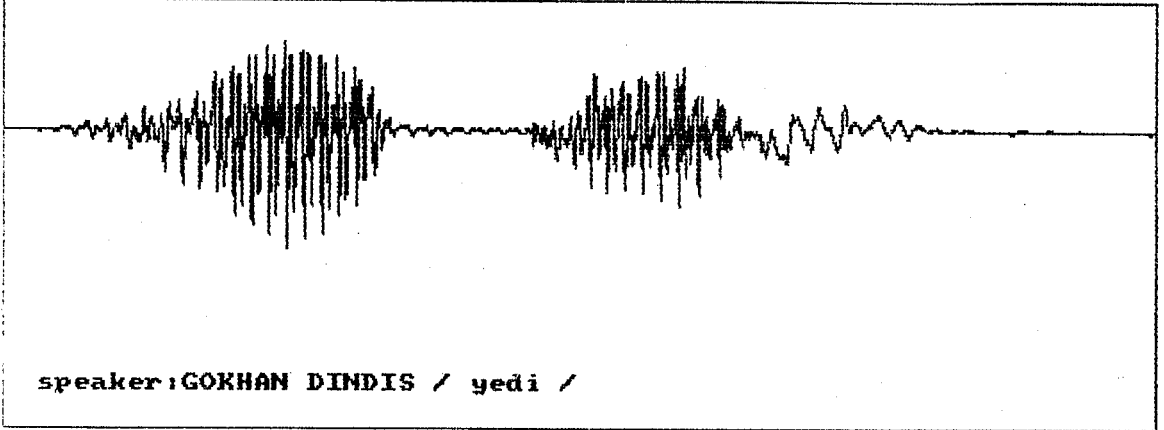
Sekil 3.4 Türkçede tipik 'dört' kelimesi



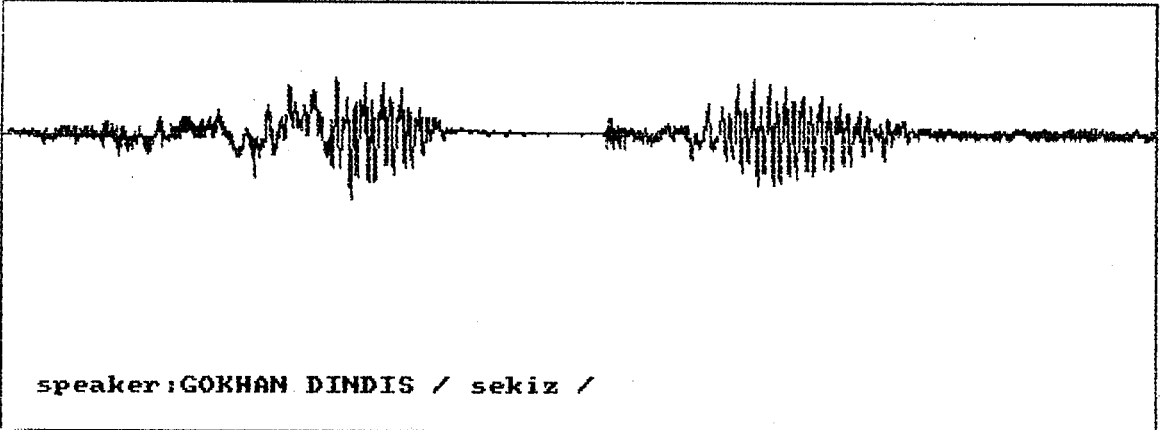
Sekil 3.5 Türkçede tipik 'beş' kelimesi



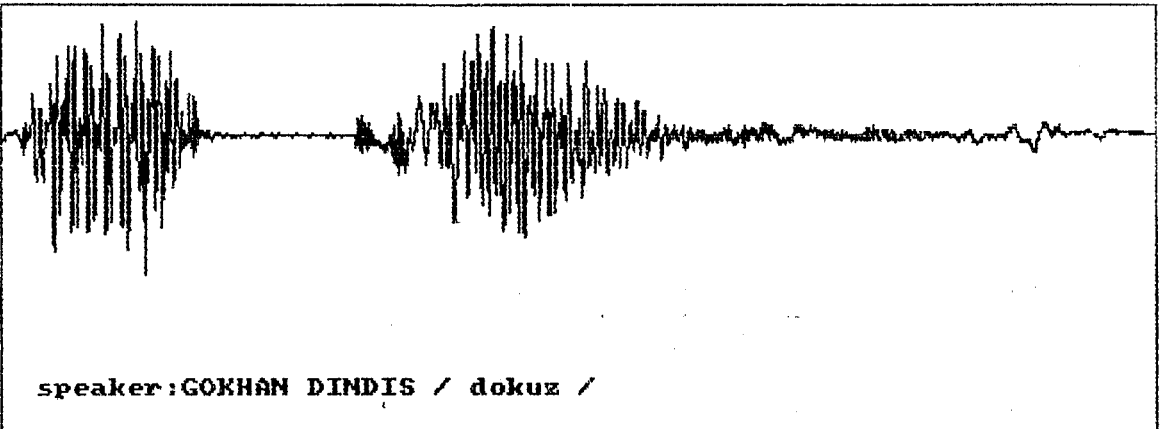
Sekil 3.6 Türkçede tipik 'altı' kelimesi



Sekil 3.7 Türkçede tipik 'yedi' kelimesi

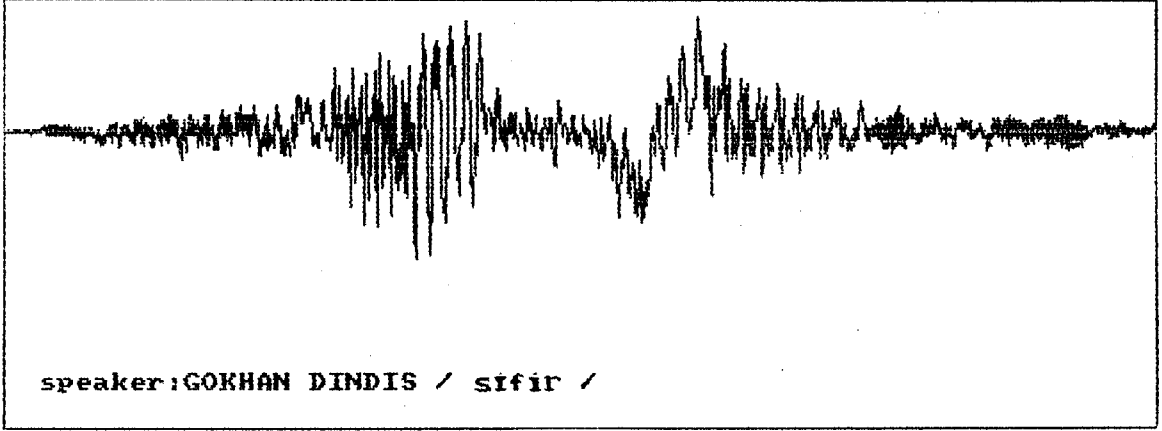


Sekil 3.8 Türkçede tipik 'sekiz' kelimesi

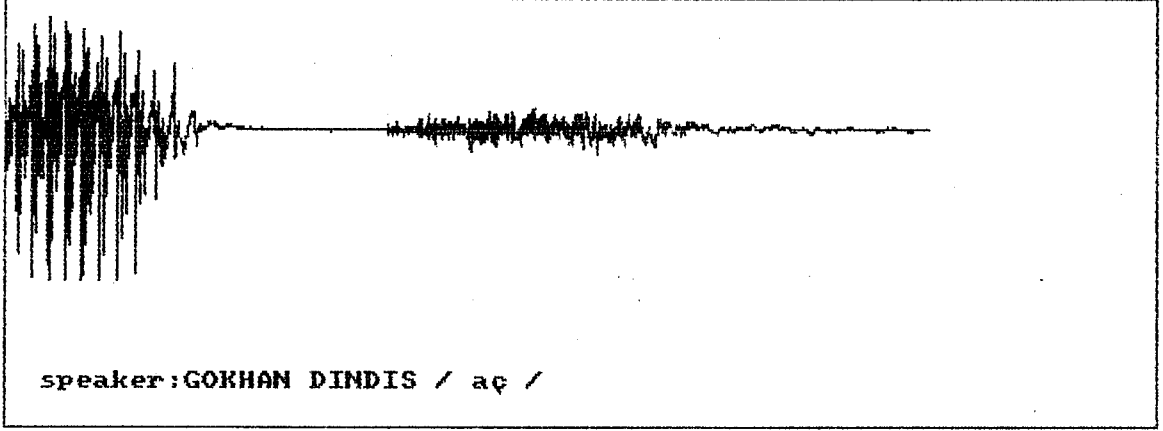


Sekil 3.9 Türkçede tipik 'dokuz' kelimesi

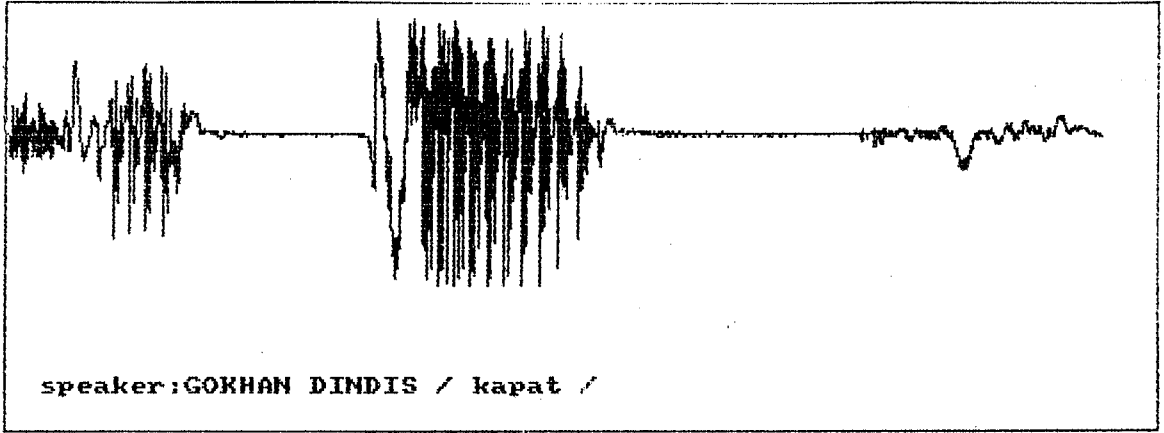




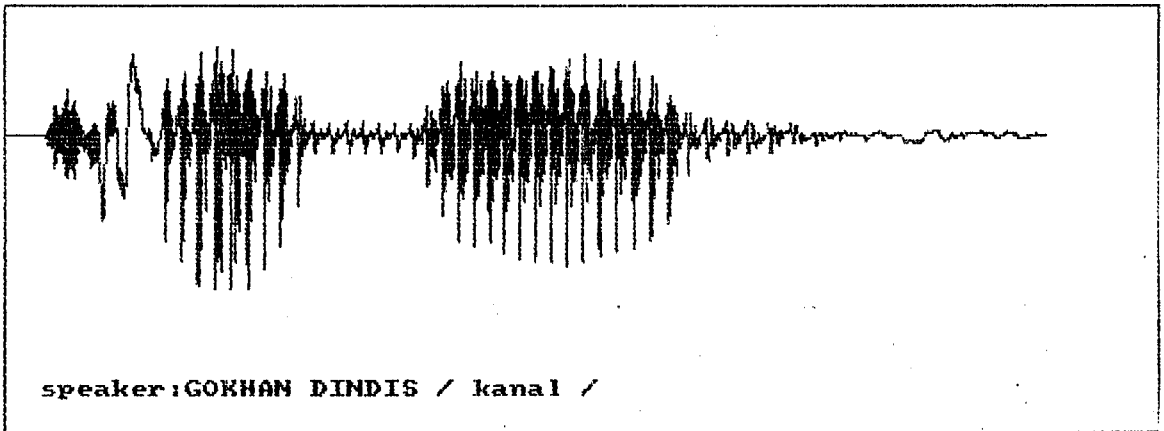
Sekil 3.10 Türkçede tipik 'sıfır' kelimesi



Şekil 3.11 Türkçede tipik 'aç' kelimesi



Şekil 3.12 Türkçede tipik 'kapat' kelimesi



Şekil 3.13 Türkçede tipik 'kanal' kelimesi

Başka bir sesten hemen sonra çıkarılamayan seslere örnek olarak k , t , ç , p harfleri verilebilir. Bunların kelime içindeki yerleşimleri 'iki' , 'üç' , 'dört' kelimelerinin dalga şekillerinde açıkça görülmektedir.

Sessiz harflerden bazıları düşük frekanslı titreşimlere ( b , p , d , l , n gibi ) , bazıları ise tam tersine yüksek frekanslı titreşimlere ( s , ş , r , z , k , t gibi ) sahiptir. Sessiz harfler birbirinden ayırdedilirken sıfır geçiş analizi tatminkar sonuç verecek şekilde görünmektedir.

Sözü edilen tipik özellikler dışında , başka ayırdedici özellik olup olmadığı hedef onuç kelimeye ait dalga şekillerinden kabaca incelenebilir. Ekler kısmında ayrıca büyültülmüş ölçekte dalga şekilleri verilmiştir.

### 3.2.2. Frekans Ekseninde Analiz

Amstrad bilgisayarında yapılan frekans analizlerinde sesli harflerin oldukça düzenli , kararlı sonuç verdiği görülmüştür. Düşünülen onuç kelime için ses tanıma algoritmalarında, frekans bakımından ince analizin sesli harflere göre yapılması uygun bulunmuştur. Kararlı ve belirgin frekans bileşenlerinin bulunması , tanıma olayında sesli harflerin kuvvetli bir faktör olduğunu belirtmektedir.

Sessiz harfleri belirlerken , frekans bileşenlerinin çok kararlı olmaması yüzünden dinamik özelliklere de bakılmıştır.

Frekans analizlerinde , sesli harflerin tipik örnekleri için elde edilen periyodik fourier serisi bileşenleri 3.14 ... 3.21 şekillerinde verilmiştir.

Bir  $f(t)$  fonksiyonu için fourier serisi;

$$f(t) = a_0 + \sum_{n=-\infty}^{\infty} c_n e^{jn\omega_0 t} \dots \dots \dots (3.1)$$

biçimindedir. Burada ,

$$c_n = 1/2 \cdot (a_n - jb_n) \dots \dots (n= 1,2,3 \dots) \dots (3.2)$$

$$|c_n| = 1/2 \cdot \sqrt{a_n^2 + b_n^2} \dots \dots \dots (3.3)$$

$$a_n = 2/T \int_0^T f(t) \cos n\omega_0 t dt \dots \dots \dots (3.4)$$

$$b_n = 2/T \int_0^T f(t) \sin n\omega_0 t dt \dots \dots \dots (3.5)$$

ifadeleriyle belirlenir.

Kullanılan örnekleme frekansı 10 kHz 'tir.

$\Delta T = 100$   $\mu s$  olarak elde edilir.

$$T = \text{periyodun sonu} - \text{periyodun başı} = (b-a) \cdot \Delta T$$

ifadesinden hesaplanır. Burada ,

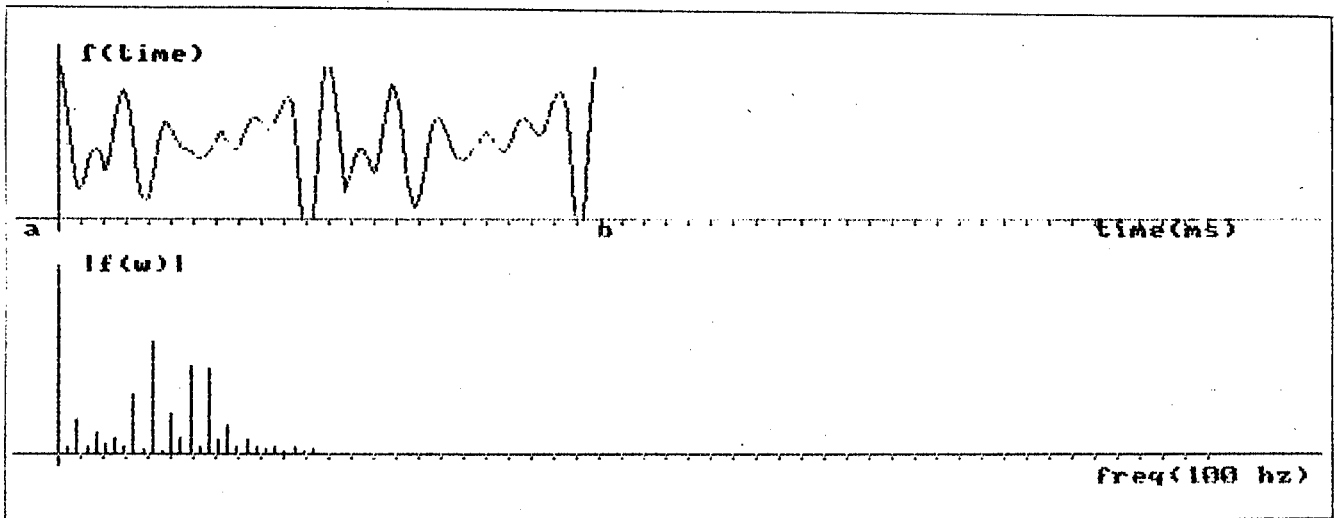
$b-a$  ; periyot içindeki veri sayısını göstermektedir. Dolayısıyla kesikli zamandaki yeni katsayılar ,

$$a_n = 2/(b-a) \sum_{k=a}^b f(k) \cos(n2\pi k/(b-a)) \dots \dots (3.6)$$

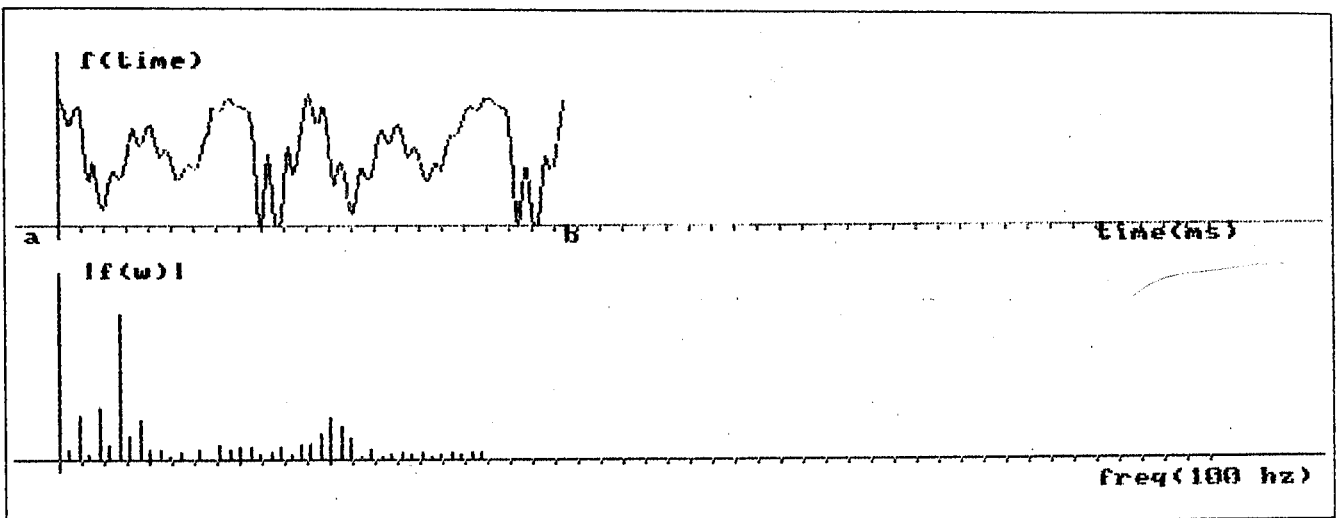
$$b_n = 2/(b-a) \sum_{k=a}^b f(k) \sin(n2\pi k/(b-a)) \dots \dots (3.7)$$

olacaktır.

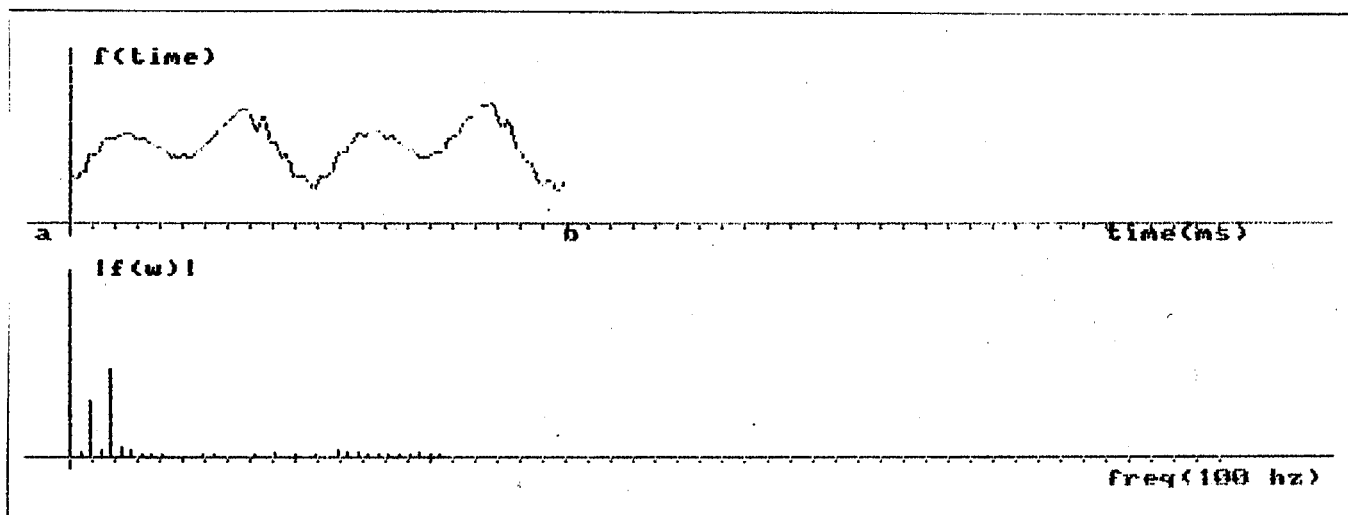
Frekans analizlerinde kullanılan program ekler kısmında verilmiştir.



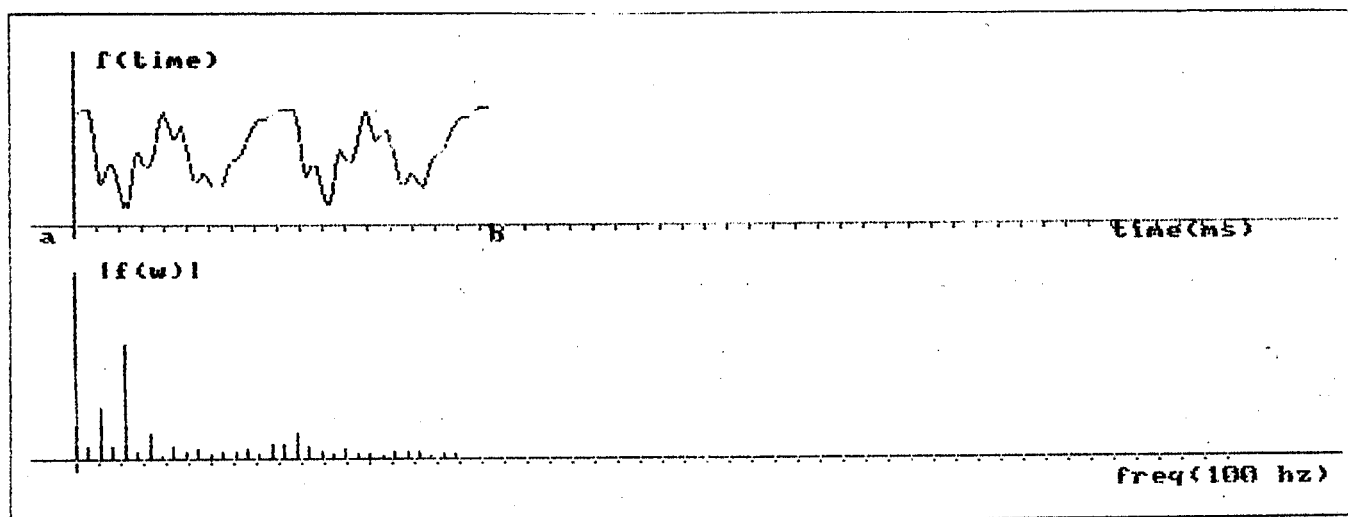
Sekil 3.14 'a' sesi



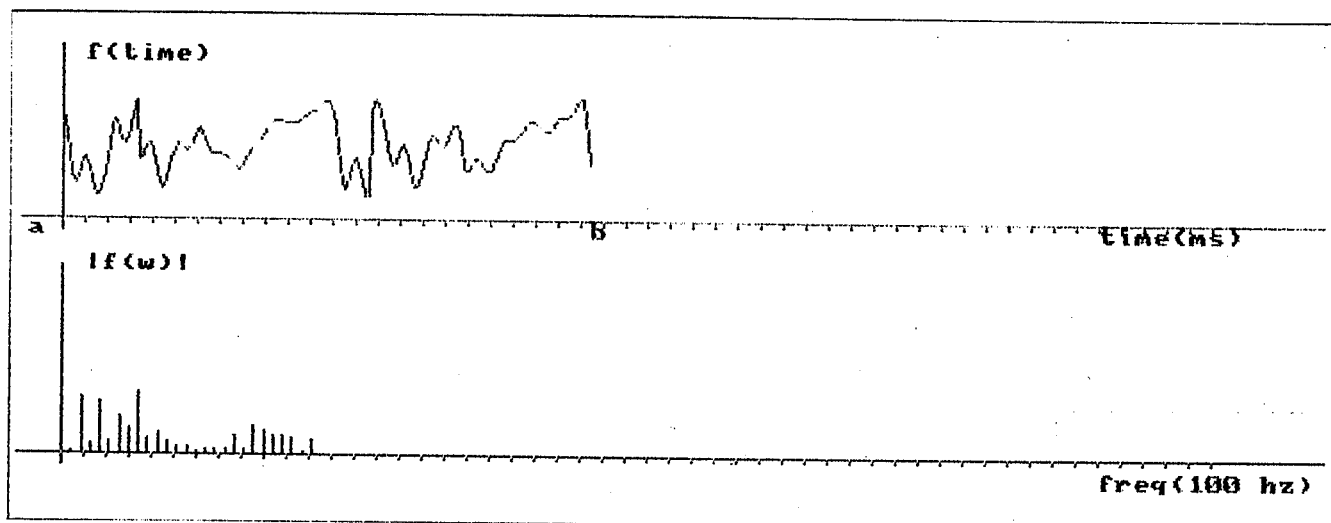
Sekil 3.15 'e' sesi



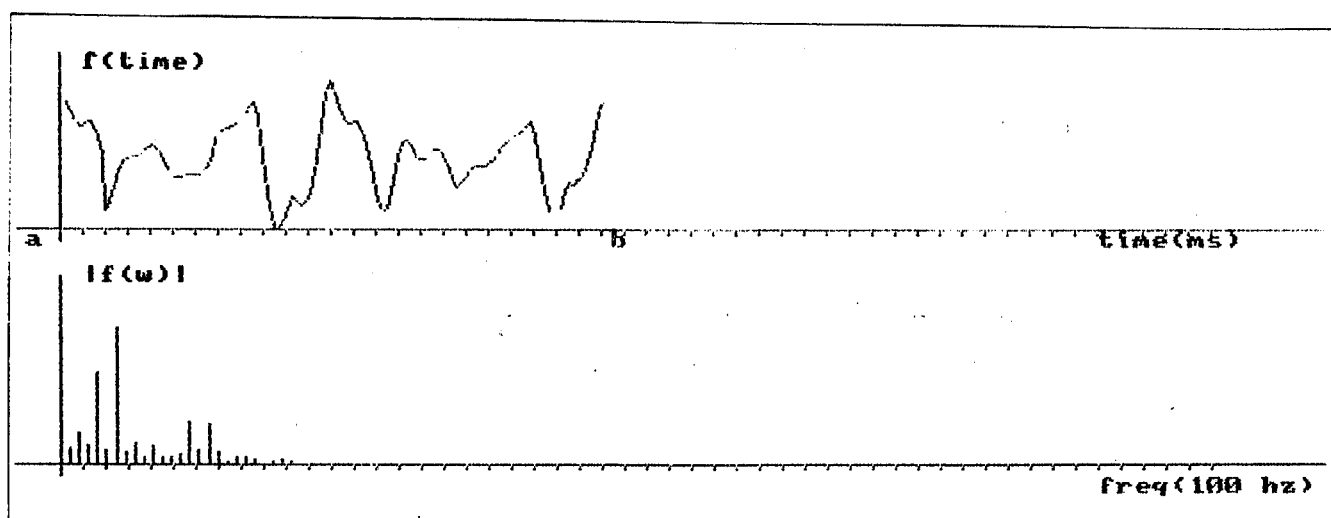
Sekil 3.16 'i' sesi



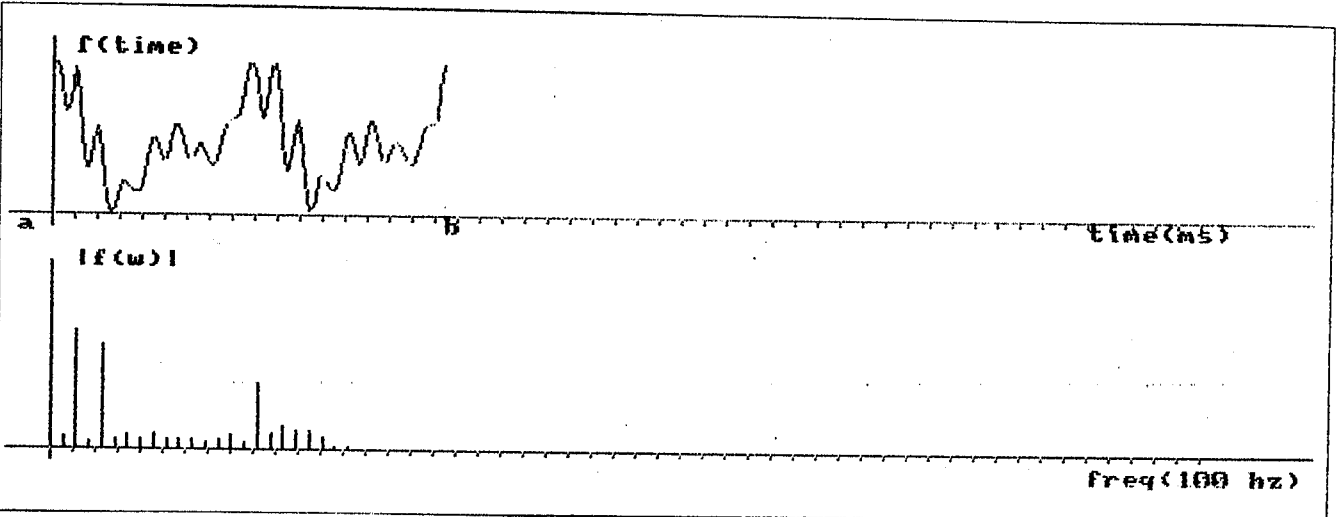
Sekil 3.17 'i' sesi



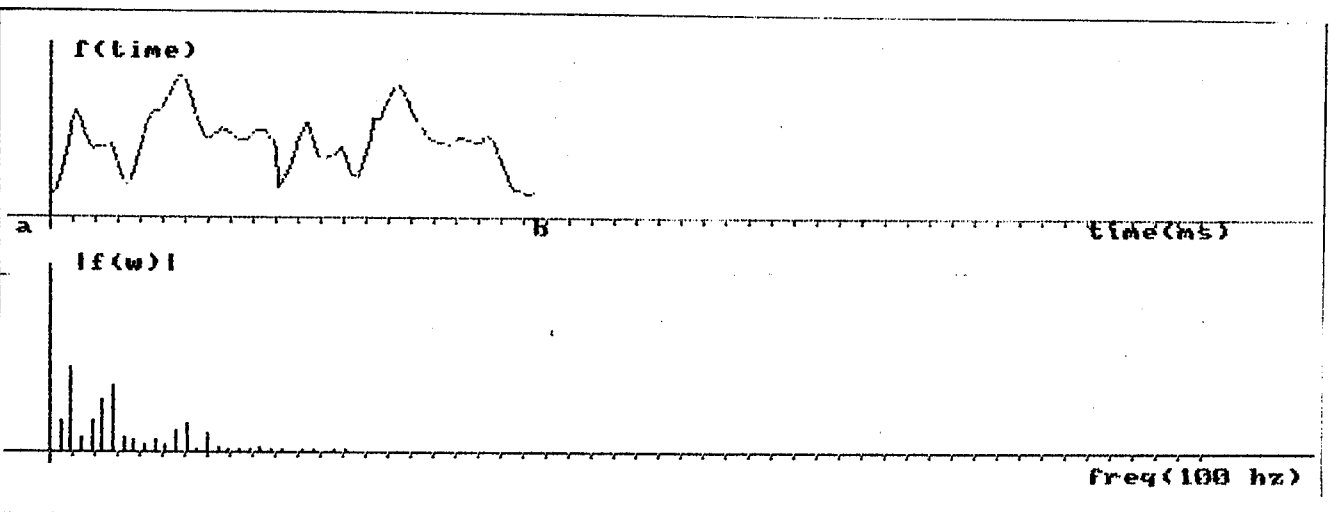
Sekil 3.18 'o' sesi



Sekil 3.19 'o' sesi



Sekil 3.20 'ü' sesi



Sekil 3.21 'u' sesi



### 3.3. Yapılan İncelemelere Göre Sistemin Tutarlılığı

Frekans ve zaman alanında yapılan incelemelere göre hedef onüç kelime için sıfır geçiş analizinin yeterli olduğu görülmüştür. Ancak genelde yüksek genlikli ve düşük frekanslı bileşenler düşük genlikli yüksek frekanslı bileşenleri bastırmaktadır. Bu nedenle sistemde düşük ve yüksek frekanslı bileşenlerin ayrı ayrı incelenebilmesi için filtreler kullanılması gerekmiştir.

Sesli harflerin belirlenmesinde kullanılacak bir devre sistemin doğruluğunu artıracaktır. Sesli harflerin yüksek genlikli ilk frekans bileşenlerini içine alacak bir frekans bandını geçiren filtre bu devrenin ilk birimi olacaktır. Filtre çıkışı sırayla zarf seçici ve karşılaştırtıcı birimlerinden geçirilerek sesli-sessiz ayrımı için kullanılacaktır.

#### 4. SİSTEMİN TASARIMI - DONANIM VE YAZILIM

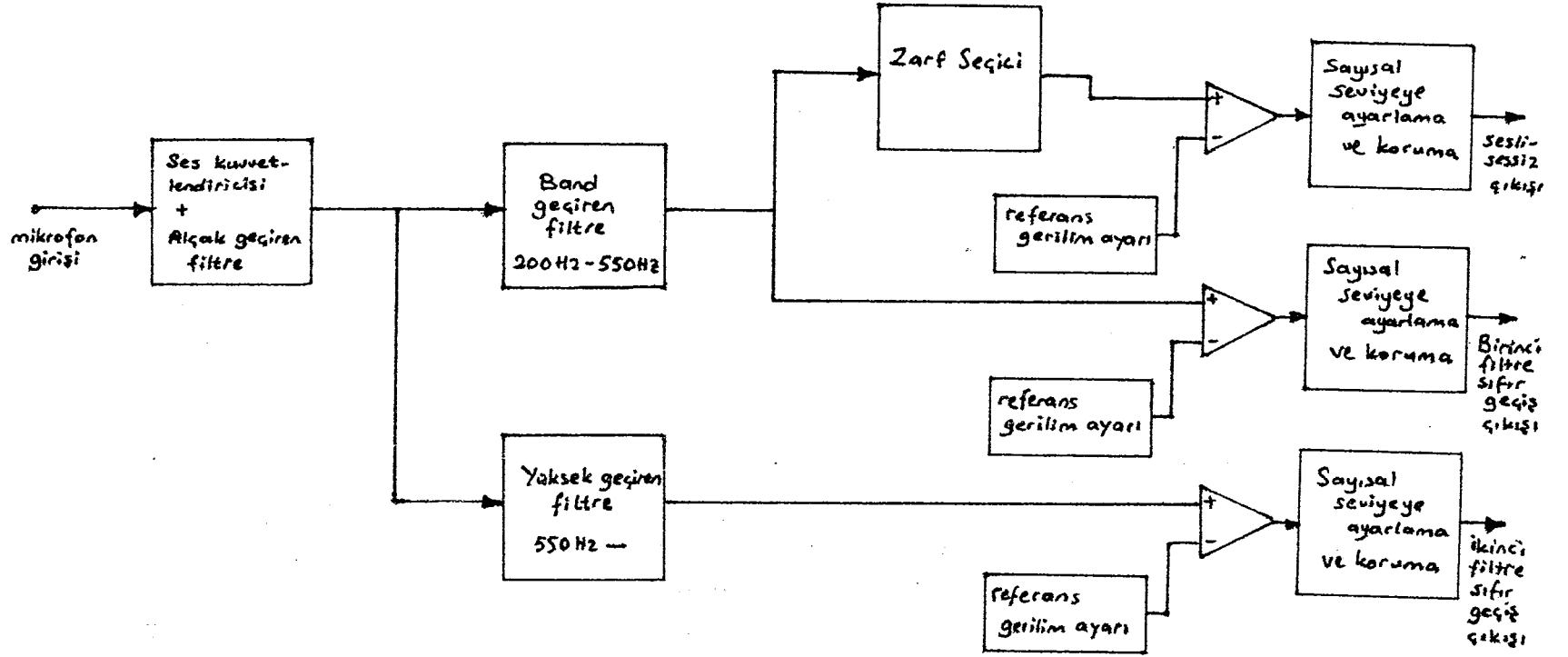
Donanım çalışmaları üç kısımdan oluşmaktadır. Bunlardan biri analog filtre devrelerini , diğer biri cihaz kontrolünü sağlayacak devreyi içermektedir. Üçüncü kısım ise bu devrelerin bilgisayarla bağlantısını sağlayan çalışmadır.

Yazılım çalışmaları akış şemalarına kadar kısmı içerilmiş , hazırlanan programlar ekler kısmında verilmiştir.

##### 4.1. Analog Donanım Çalışmaları

Analog devre mikrofon girdisini üç sayısal çıktıya çevirebilecek şekilde tasarlanmıştır. Bu üç sayısal çıkışın ikisi , daha önceki bölümde bahsedildiği gibi özellikle frekans bileşenlerinin yığıldığı iki frekans bandı için hazırlanmış filtrelerdir. Filtre çıkışlarındaki işaretler düzeltilerek "0", "1" mantık seviyelerine dönüştürülmüştür. Üçüncü çıkış sesli-sessiz çıkışıdır. Sesli harflerin yüksek enerjili ilk frekans bileşenlerinin zarf seçici devreden geçirilmesiyle elde edilmiştir. Birinci filtrenin geçirdiği band 200 Hz - 550 Hz, ikinci filtrenin geçirdiği band ise 550 Hz - 5000 Hz arasındadır. Devrenin blok şeması şekil 4.1 'de verilmiştir.

Mikrofondan gelen ses sinyali kuvvetlendiriciden geçerek yükselir ve filtre devrelerinde süzülür. Birinci filtre çıkışı sesli işaretlerin belirlenebilmesi için zarf seçici devreye girer. Zarf seçici

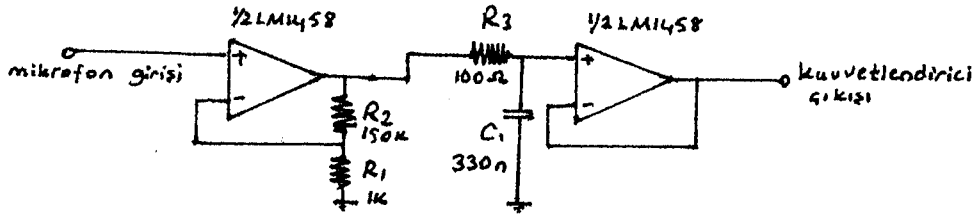


Şekil 4.1 Analog Donanım blok şeması

devre çıkışı ise bir referans seviyeye karşılaştırılır. Karşılaştırıcı çıkışı mantık "1" seviyesinde ise işaret sesli harfe ait demektir. Birinci filtre çıkışı aynı zamanda ikinci filtre çıkışı gibi bir referans seviyeye karşılaştırılır ve sıfır geçiş analizi için kullanılacak diğer veriler oluşturulur.

#### 4.1.1. Ses Kuvvetlendiricisi

Ses kuvvetlendiricisi bir LM1458 işlemsel kuvvetlendirici entegre devresi ile tasarlanmıştır. Pozitif kuvvetlendiricidir ve kazancı 40 dB civarındadır.  $R_2$  ve  $C_1$  elemanları kesim frekansını 5 KHz civarına ayarlamak için kullanılmıştır.



Sekil 4.2 Ses kuvvetlendiricisi

Burada ;  
 $f = \frac{1}{2\pi R_3 C_1} = \frac{1}{2\pi 100 \cdot 330 \cdot 10^{-9}} \approx 4800 \text{ Hz}$   
 ve ,  
 $\text{kazanç} = \frac{R_2}{R_1} + 1 = \frac{150k}{1k} + 1 \approx 150$   
 $= 20 \log 150 \text{ dB} \approx 43.5 \text{ dB}$   
 olarak hesaplanır.

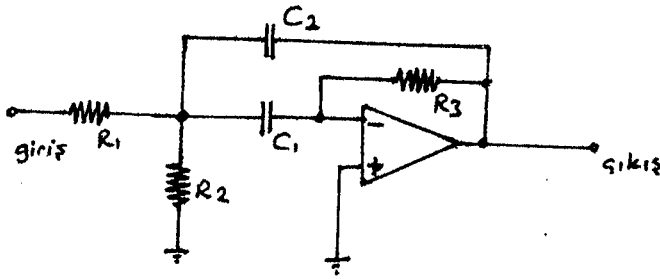
Devrede 5 KHz'lik bir alçak geçiren filtre kullanılmasının sebebi bilgisayarın örnekleme frekansının 10 KHz olmasıdır.

$$f_{\text{örnek}} \geq 2f \quad , \quad f: \text{max ses frekansı} \dots \dots \dots (4.1)$$

olması ile örnekleme teorilerine sadık kalınmış ve bu yüzden ortaya çıkabilecek hatalar önlenmiştir.

#### 4.1.2. Band Geçiren Filtre

LM1458 işlemsel kuvvetlendirici entegre devresi ile tasarlanmış olup gerilim kontrollü gerilim kaynağı tipinde ikinci dereceden bir band geçiren filtredir. Merkez frekansı 350 Hz olup band genişliği 400 Hz olarak belirlenmiştir. (Şekil 4.3)



Şekil 4.3 Band geçiren filtre

Devrede kullanılacak elemanları belirlemek için ;

$$\omega_0 = 2200 \text{ rad/s}, G = 3, Q = 2, C_1 = C_2 = 100 \text{ nF}$$

olarak seçilir. Direnç değerleri ise ;

$$R_1 = \frac{Q}{G C_1 \omega_0} \dots \dots \dots (4.2)$$

$$R_2 = \frac{Q}{(2Q^2 - G) C_1 \omega_0} \dots \dots \dots (4.3)$$

$$R_3 = \frac{2Q}{C_1 \omega_0} \dots \dots \dots (4.4)$$

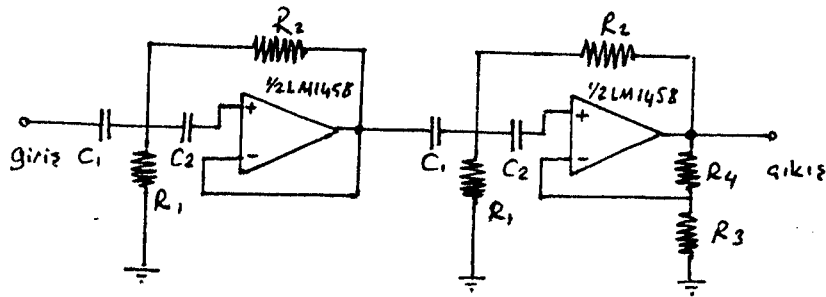
denklemlerinden ,

$$R_1 \approx 3 \text{ k}\Omega, R_2 \approx 1.8 \text{ k}\Omega, R_3 \approx 18 \text{ k}\Omega$$

olarak bulunur.

### 4.1.3. Yüksek Geçiren Filtre

Dördüncü dereceden bir yüksek geçiren filtre 550 Hz 'ten yukarıdaki frekans bileşenlerinin sıfır geçişlerini tesbit için kullanılmıştır. Daha önceki ses kuvvetlendiricisi katındaki, kesim frekansı 5 KHz olan alçak geçiren filtre sayesinde devrenin çıkışında 550 Hz - 5 KHz arası bileşenler görülmektedir.



Sekil 4.4 Yüksek geçiren filtre

Burada ,

$$f_c = \frac{1}{2\pi\sqrt{R_1 R_2 C_1 C_2}} \quad \dots \dots \dots (4.5)$$

olarak verilmiştir.

$$C_1 = C_2 = 100 \text{ nF}$$

seçilirse  $R_1 = R_2$  şartı için ;

$$R_1 = R_2 \approx 2.8 \text{ k}\Omega$$

olarak hesaplanır. Filtrenin yüksek frekans kazancı ,

$$G = \frac{R_4}{R_3} + 1 \quad \dots \dots \dots (4.6)$$

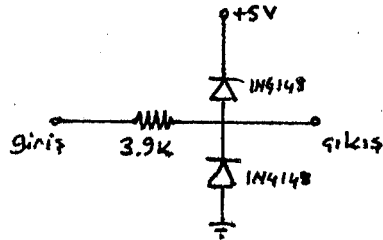
eşitliği ile verilmiş ve buna göre ;

$$G = \frac{4.7\text{K}}{1\text{K}} + 1 \approx 6$$

olmuştur.

#### 4.1.5. Sayısal Seviye Ayarlama ve Koruyucu

Bu birimde kullanılan direnç , çıkışı kısa devre edecek tür kazalardan , diyotlar ise çıkışa takılacak bilgisayar uyum devresini 0 - 5 volt dışında gelebilecek gerilimlerden korumak için kullanılmıştır. Dolayısıyla çıkış "0" veya "1" mantık seviyesinde olabilmektedir.

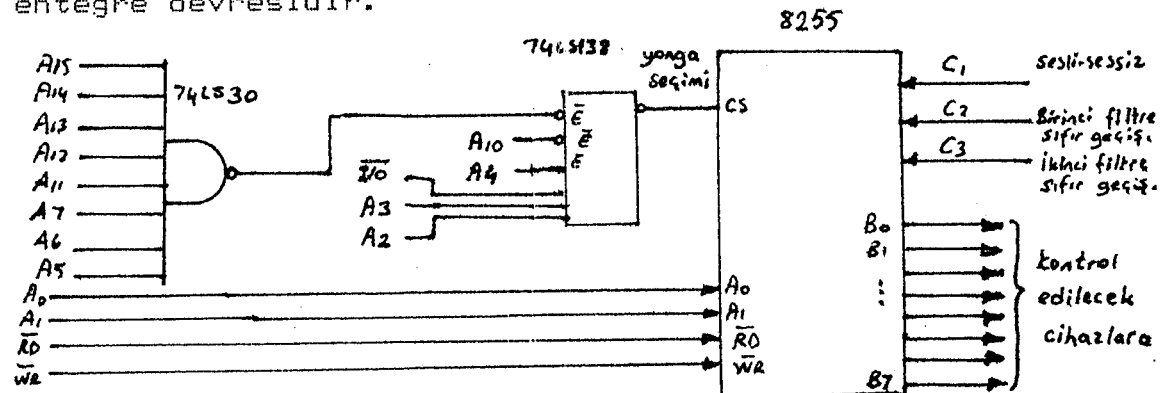


Şekil 4.6 Sayısal seviye ayarı ve koruma

devresi

#### 4.2. Amstrad CPC6128 Bilgisayarı İçin Uyum Devresi

Amstrad CPC6128 bilgisayarında hazır giriş/çıkış portu bulunmadığından özel bir giriş/çıkış portunun hazırlanması gerekmiştir. Bu amaç için bilgisayarda , kullanıcı için ayrılmış adreslere ( F8F8H - F8F3H ) bir adres çözücü ile giriş/çıkış uyum birimi yerleştirilmiştir. Bu birim 8255 paralel giriş/çıkış entegre devresidir.



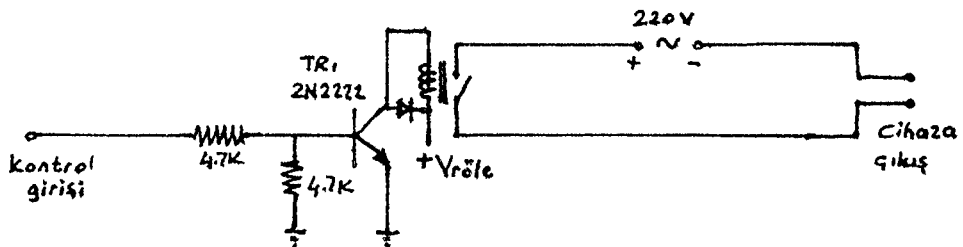
Şekil 4.7 Uyum devresi

Kullanılan adreslerden biri seçildiğinde yonga seçimi hattı "0" mantık seviyesine düşmekte ve 8255 devresinin aktif hale geçmesiyle mikroişlemci, devre ile irtiba kurabilmektedir.  $A_0$  ve  $A_1$  adres girişleri 8255'in içindeki yazaçlardan birini seçmek için kullanılır. 8255 entegre devresine ait bilgi ekler kısmında verilmiştir.

$C_1$ ,  $C_2$ ,  $C_3$  girişleri analog devreden gelen sıfır geçiş hatları,  $B_0 - B_7$  çıkışları ise herbiri bir cihaz kontrol edebilecek kontrol çıkışlarıdır.

#### 4.3. Cihaz Kontrolü İçin Gerekli Ara Devre ve Çalışma Şekli

Cihaz açma-kapama kontrolü için şekil 4.8' de verilen devre kullanılmıştır. Kontrol edilecek her cihaz için 8255'e ait B portu çıkışlarından birine bu devrelerden takılabilir. Ses tanıyan sistem, cihazın açılması istendiğinde çıkışa "1" mantık seviyesi gönderecek ve röle yoluyla 220 voltluk şebeke gerilimini cihaza verecektir. Açma-kapama dışında daha değişik kontroller için sistem programlarına ilaveler yapılabilir.



Şekil 4.8 Cihaz Kontrol Devresi



#### 4.4. Yazılım Çalışmalarına Giriş

Ses tanıma sisteminde kullanılacak algoritmalara geçmeden önce , kullanılacak kelimelere has belirgin özelliklere bakmakta fayda vardır. Daha önceki bölümde de bahsedildiği gibi sesli harfler kararlı ve yüksek genlikli frekans bileşenlerine sahip olduğundan birbirlerinden kolayca ayrılabilirlerdir. Sessiz harflerin bir kısmı çok yüksek frekanslı bileşenlere sahip olduğundan (s, ş, t gibi) diğerlerinden kolayca ayırdedilebilirlerdir. Bazı sessizler ise ( b,d gibi ) çok düşük frekanslarda bileşene sahiptir. Referans seviyesini aşacak genlikte bir yüksek frekans çıktısı oluşturamazlar.

C , k , t , p gibi sessizler insan ağzının fiziksel yapısından dolayı başka bir sesin arkasından hemen çıkarılamazlar . Ağız bu sesleri çıkarmak için hazırlık yaparken kısa bir sessizlik oluşur. İki , Üç ve dört kelimelerinin dalga şekillerinde ( bölüm 3 ) daha önce incelenmişti. Sözü edilen sessiz harflerden önce sıfır geçiş bilgileri gelmeyecek , dolayısıyla frekans bileşenlerinin birbirine yakın olabildiği , birbirine benzer sesliler kelime ayırımına yetmediginde , kullanılabilirlerdir. Beş ve dört kelimelerinde dört'ün içindeki " r " sesi çok kısa sürmekte ve "ö" sesinin içinde kaybolmaktadır. Buna rağmen "T" sesi , "ş" gibi yüksek frekans bileşenlerine sahip olsa bile önünde boşluk olduğundan tanınacak ve "ş" sesinden

ayrılacaktır. Böylece kelimenin beş değil dört olduğu anlaşılacaktır.

Sistem durum makinesi ( state machine ) şeklinde tasarlanmıştır. Gelen seslere göre değişik durumlara girmektedir. Kelimelerin son sesleri ise genelde son durumları ifade etmekte , makine bu durumlara geldiğinde kontrol işlerini icra etmektedir .

#### 4.5.Sıfır Geçiş Verilerinin Ön İşlemesi

Sıfır geçiş verileri daha önce sözü edilen üç bilgi hattından gelmektedir. "0" mantık durumundan "1" mantık durumuna geçen bir hat , gelen sinyalde bir periyodun bittiğini yeni periyodun başladığını ifade etmektedir. İki "0" 'dan "1" 'e geçiş arasındaki periyodun ölçülmesini sağlamak amacıyla veriler bir ön işlemden geçmelidir . Tanıma işlemi sırasında arka arkaya birbirine yakın periyotlar gelip gelmediği ve gelen periyotların istatistiksel aritmetik ortalaması bu işi yapacak programla sağlanmaktadır . Ayrıca aynı program son birkaç periyodun saklanması işini de yüklenmektedir.

Programın tam listesi ekler bölümünde verilmiştir. Programda kullanılan ve tanımda gerekecek değişkenler açıklamalarıyla beraber tablo 4.1 'deki gibidir.

Tablo 4.1 Ön işlemede kullanılan değişkenler

VCFLAG	Ses sinyalinin sesli harfe ait olup olmadığını belirtir.
BPFLAG	Birinci filtreden gelen işaretin pozitif mi negatif mi olduğunu belirtir.
BPPER	Birinci filtreden gelen son iki negatiften pozitive geçiş arasındaki periyodu verir . Birimi 100 $\mu$ s ' dir. Değeri 255 ise periyot çok uzun demektir. Yani uzun bir sessizlik vardır.
HPFLAG	ikinci filtreden gelen işaretin pozitif mi negatif mi olduğunu belirtir.
HPPER	ikinci filtreden gelen son iki negatiften pozitive geçiş arasındaki periyodu verir . Birimi 100 $\mu$ s ' dir. Değeri 255 ise periyot çok uzun demektir. Yani uzun bir sessizlik vardır.
BPBUF	Birinci filtreden gelen son üç periyodun saklandığı yerdir . Üç yazaçtan oluşur . Birimleri 100 $\mu$ s ' dir.
HPBUF	ikinci filtreden gelen son sekiz periyodun saklandığı yerdir . Sekiz yazaçtan oluşur birimleri 100 $\mu$ s ' dir.
BPTOT	Birinci filtreden gelen son üç periyodun ortalamasını belirtir . Birimi 33.3 $\mu$ s ' dir.
HPTOT	ikinci filtreden gelen son sekiz periyodun ortalamasını belirtir . Birimi 12.5 $\mu$ s ' dir.

#### 4.6. Ses Tanımadaki Kullanılabilecek Modellemeler

Sıfır geçişleri yöntemi temelde frekans analizi yerine periyot analizlerine dayanan bir tekniktir.

Tanıma işlemi yapacak sistem her 10 ms'lik süre içinde sıfır geçiş verilerine bakacaktır. 10 ms ise insan sesinin en kısa sürdüğü bir vakittir. 10 ms içinde band geçiren filtre çıkışından ortalama üç periyot gelmektedir. Bu nedenle analizde son üç periyodun ortalamasının kullanılmasına karar verilmiştir.

$$T_{\text{bandgeçiren}} = 1/3 \sum_{i=1}^3 T_{1-i}, \quad \begin{array}{l} T_0 = \text{son periyot} \\ T_1 = \text{bir önceki periyot} \\ T_2 = \text{iki önceki periyot} \end{array}$$

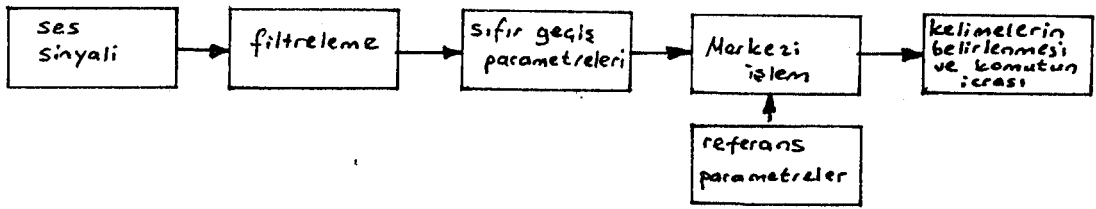
Yüksek geçiren filtre çıkışından ise son sekiz periyoda ait ortalamalar incelenmiştir. Filtre çıkışından tanıma işlemi yapılacak 10 ms içinde, sesliler için ortalama dokuz-on periyot gelmektedir. Ön işlemede program hızını fazla düşürmemek için son sekiz periyot ortalamasının incelenmesinde karar kılınmıştır. Bu sonucu pek fazla değiştirmeyecektir.

$$T_{\text{yüksek geçiren}} = 1/8 \sum_{i=1}^8 T_{1-i}, \quad \begin{array}{l} T_0 = \text{son periyot} \\ \dots \\ T_7 = \text{yedi önceki periyot} \end{array}$$

Bant geçiren ve yüksek geçiren filtrelere ait sıfır geçiş ( periyot ) ortalamalarının , kelime analizinde hangi sınırlarda hangi seslere ait olduğu akış şemalarından izlenebilir. Sesli ve sessiz harflere ait sınırların kelimedenden kelimeye değişiklik gösterdiği ilk bakışta anlaşılabilir. Bunun sebebi ise insan

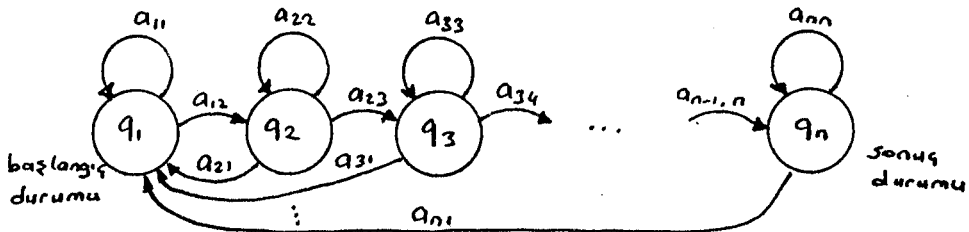
sesinin , yerine göre frekans ve genlik bakımından değişiklik göstermesidir. Zaten ses analizinde zorlukların başlıca kaynağı da budur .

Sistemin teorik çalışma modeli şekil 4.9' da verildiği gibidir. Gelen sinyallerden elde edilen sıfır geçiş periyotları parametre olarak kullanılır ve referans parametrelerle karşılaştırılarak sonuca ulaşmaya çalışılır .



Şekil 4.9 Sistemin Çalışma Modeli

Sistemin tanıma olayını gerçekleştirirken durum makinesi olarak çalıştığına değinilmişti . Kelime tanıma işleminde kullanılan durum modeli şekil 4.10 'daki gibidir. Şekilde  $q_i$ 'ler kelimenin hangi sesi olduğu durumları ,  $a_{ij}$ 'ler ise durum değişikliği yaratacak parametrelerin varlığını veya yokluğunu belirtmektedir. Sonuç durumuna gelinmesi ise kelimenin tanındığını gösterir .



Şekil 4.10 Kelime tanıma durum modeli

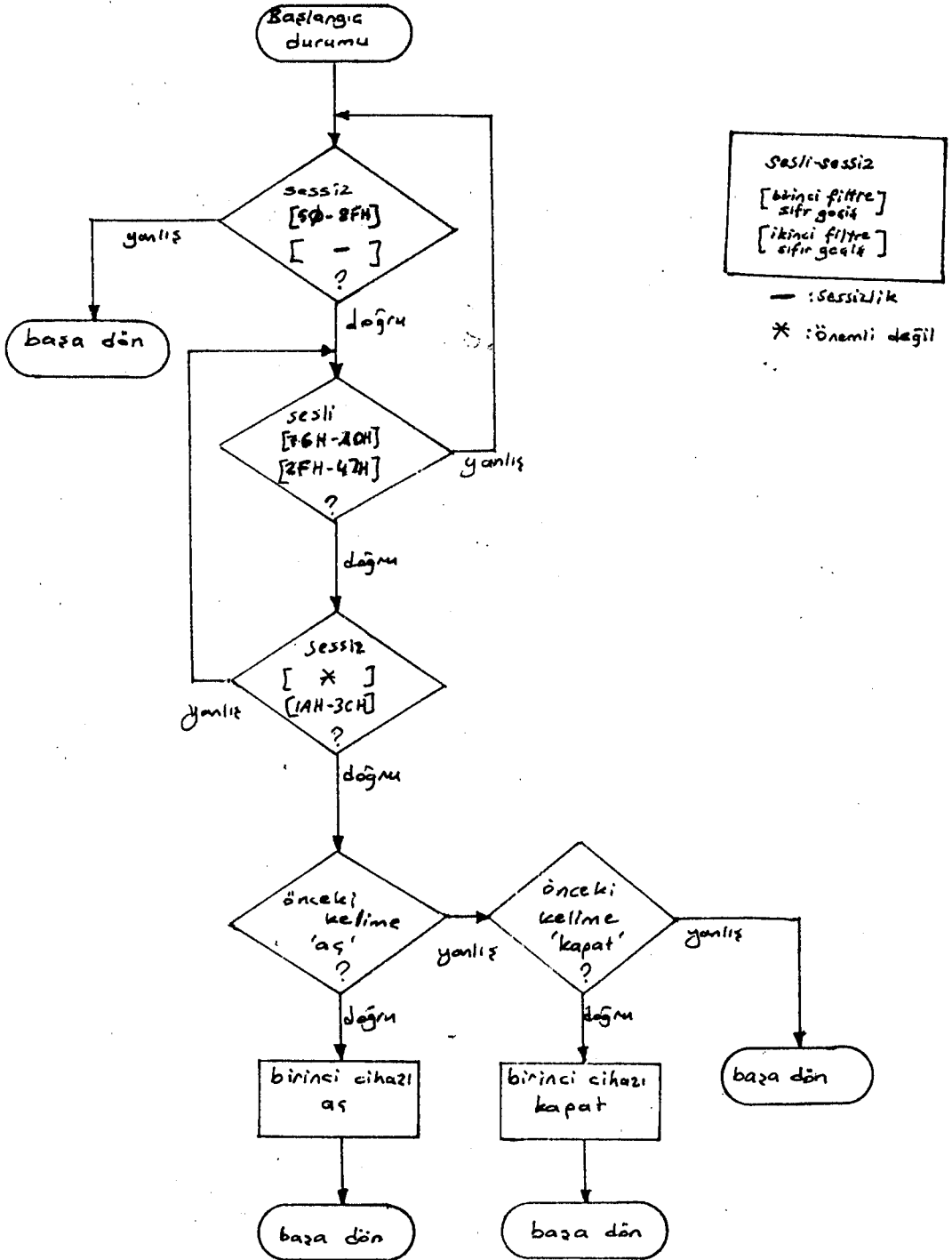
$$a = \begin{cases} 1, & \text{minimum fark} < \left| \begin{array}{l} \text{referans} \\ \text{parametre} \end{array} - \begin{array}{l} \text{elde edilen} \\ \text{parametre} \end{array} \right| < \text{maksimum fark} \\ 0, & \text{diğer durumlar} \end{cases}$$

olmak üzere ;

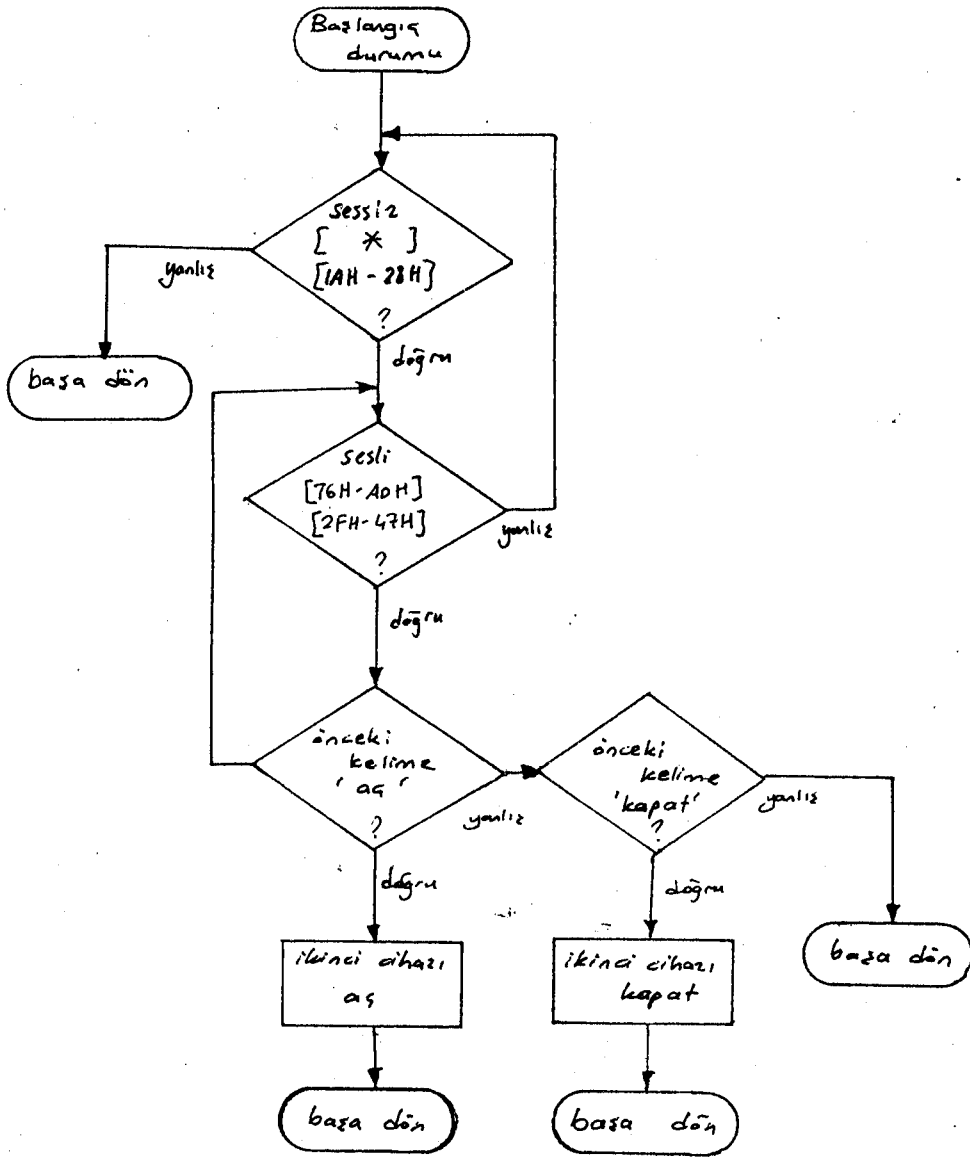
$$q_i = q_{i-1} \cdot a_{i-1} + q_i \cdot a_{ii} + q_1 \cdot a_{i1} , \quad ( i, j=1, \dots, n )$$

Bu ifade , son durumun bir önceki durumlarla olan ilişkisini göstermektedir .

## 4.7. Tanıma Alt Programlarına Ait Akış Şemaları

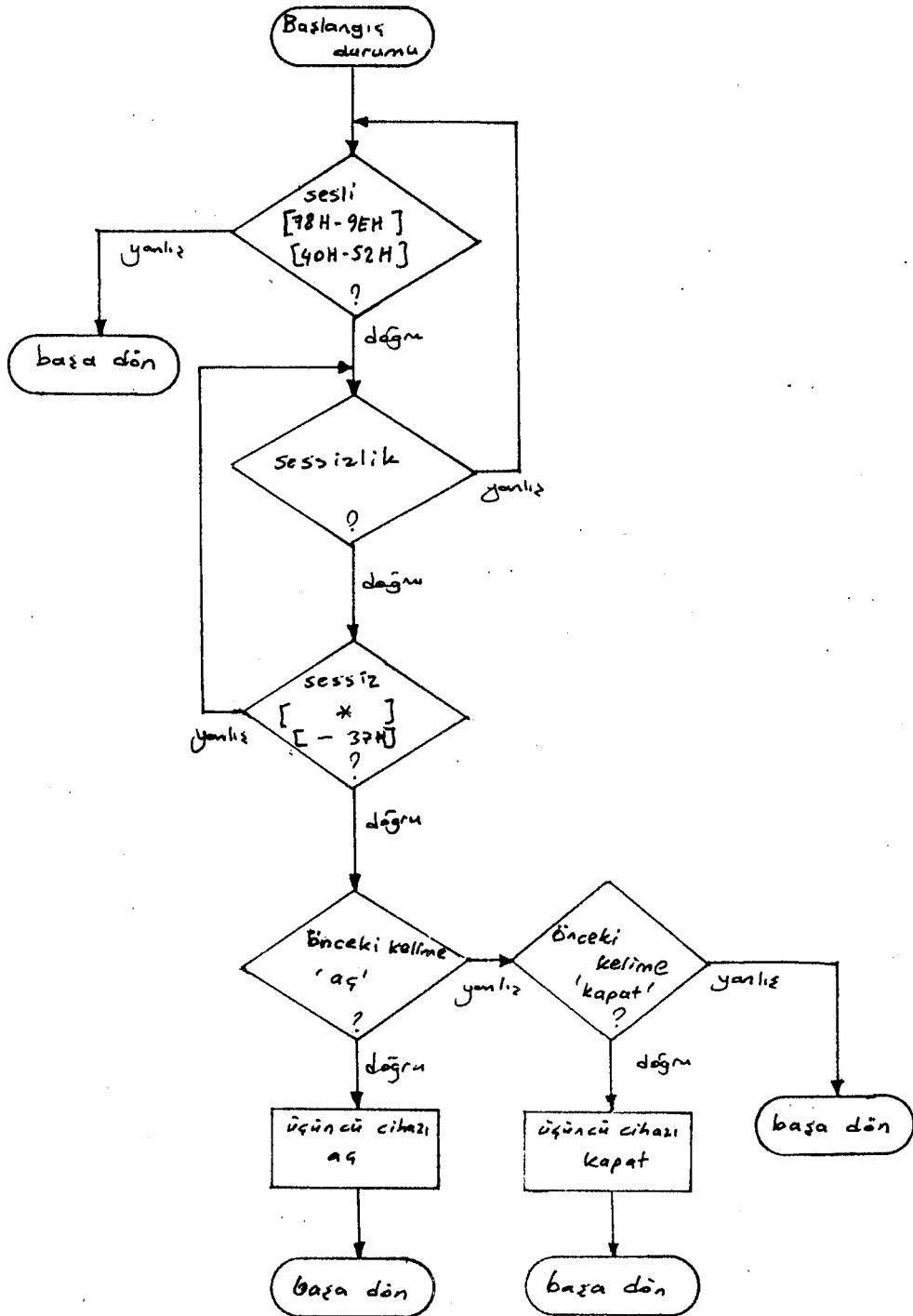


Şekil 4.11 'bir' kelimesine ait akış şeması

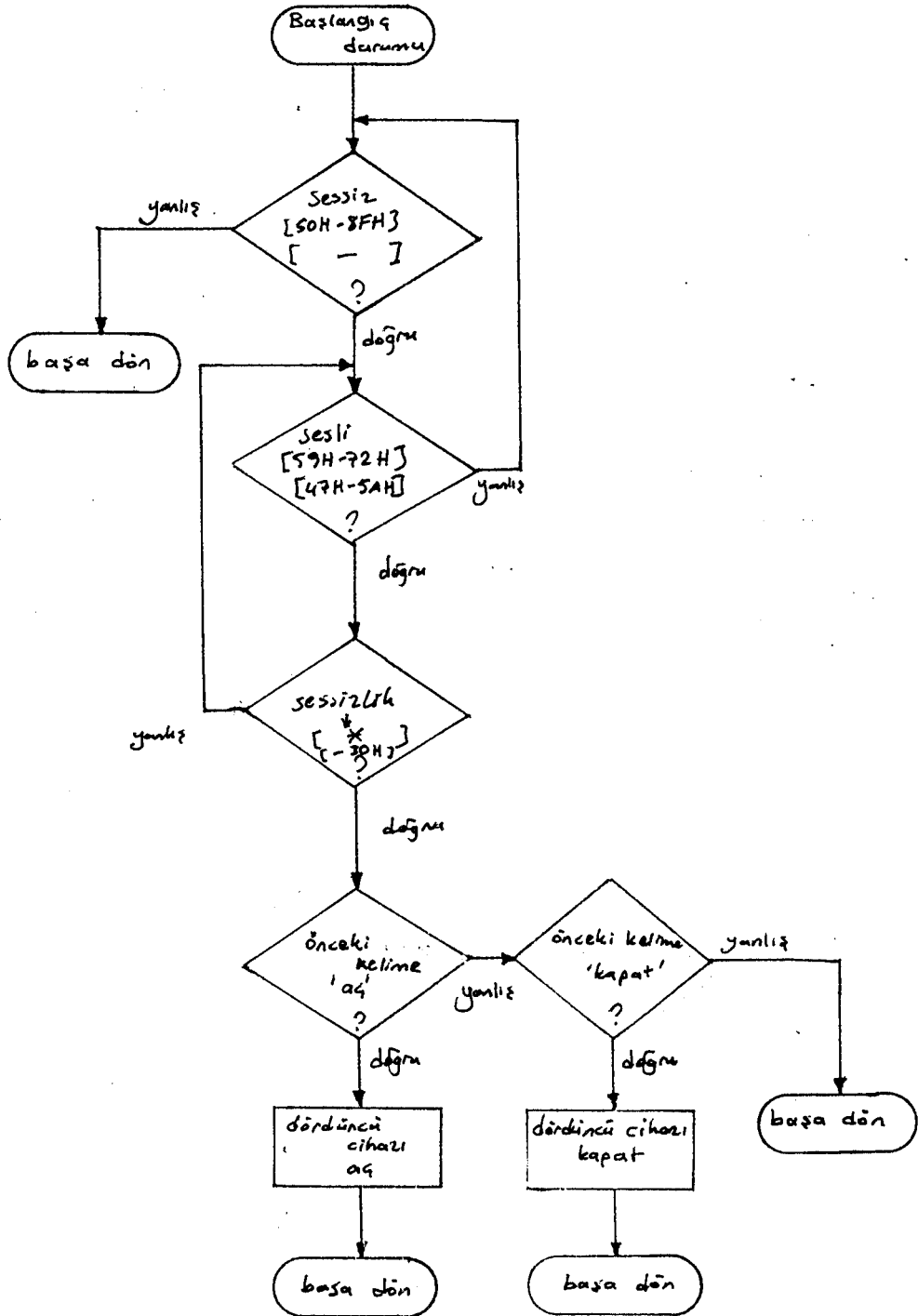


Sekil 4.12 'iki' Kelimesine ait akış şeması

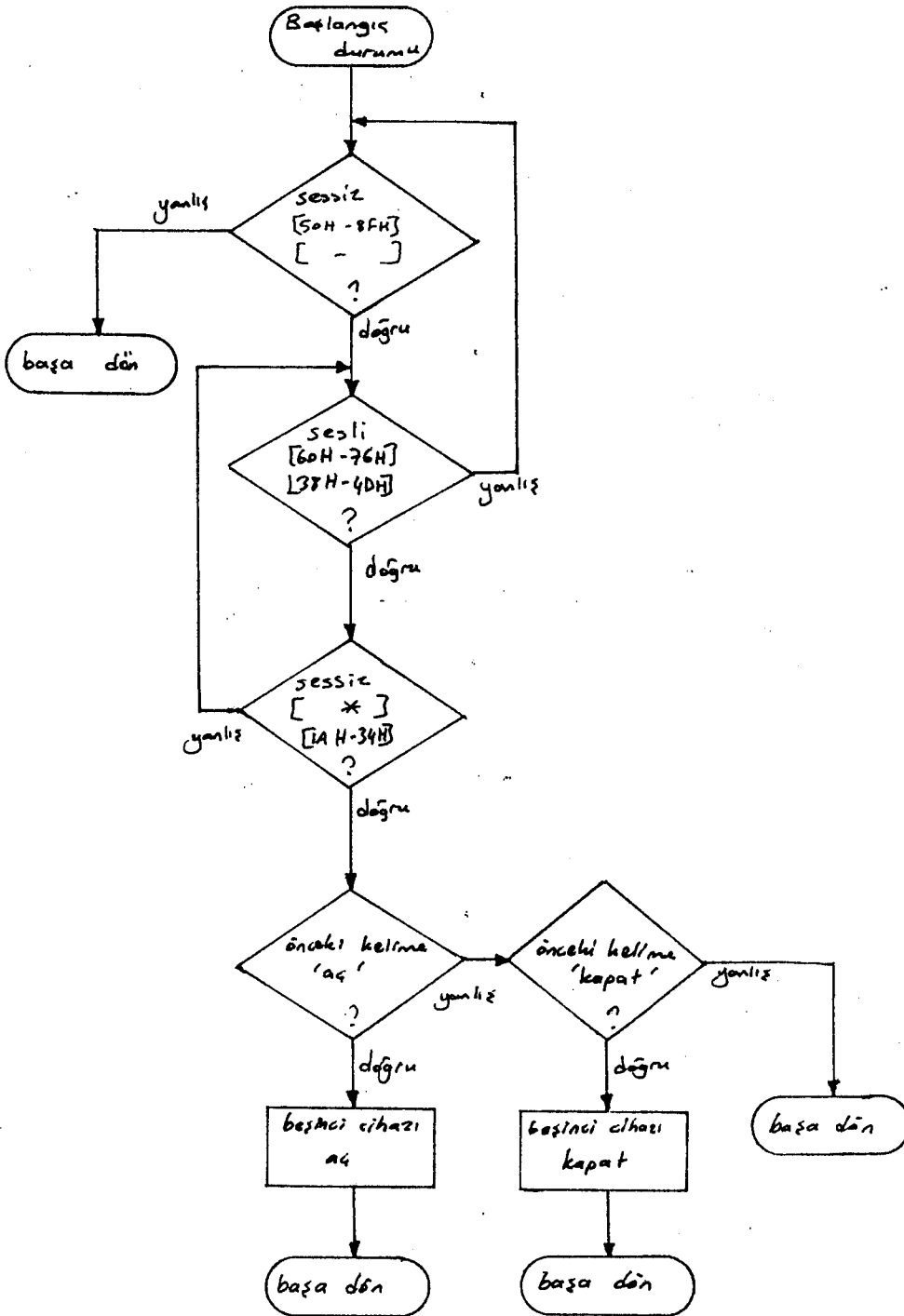




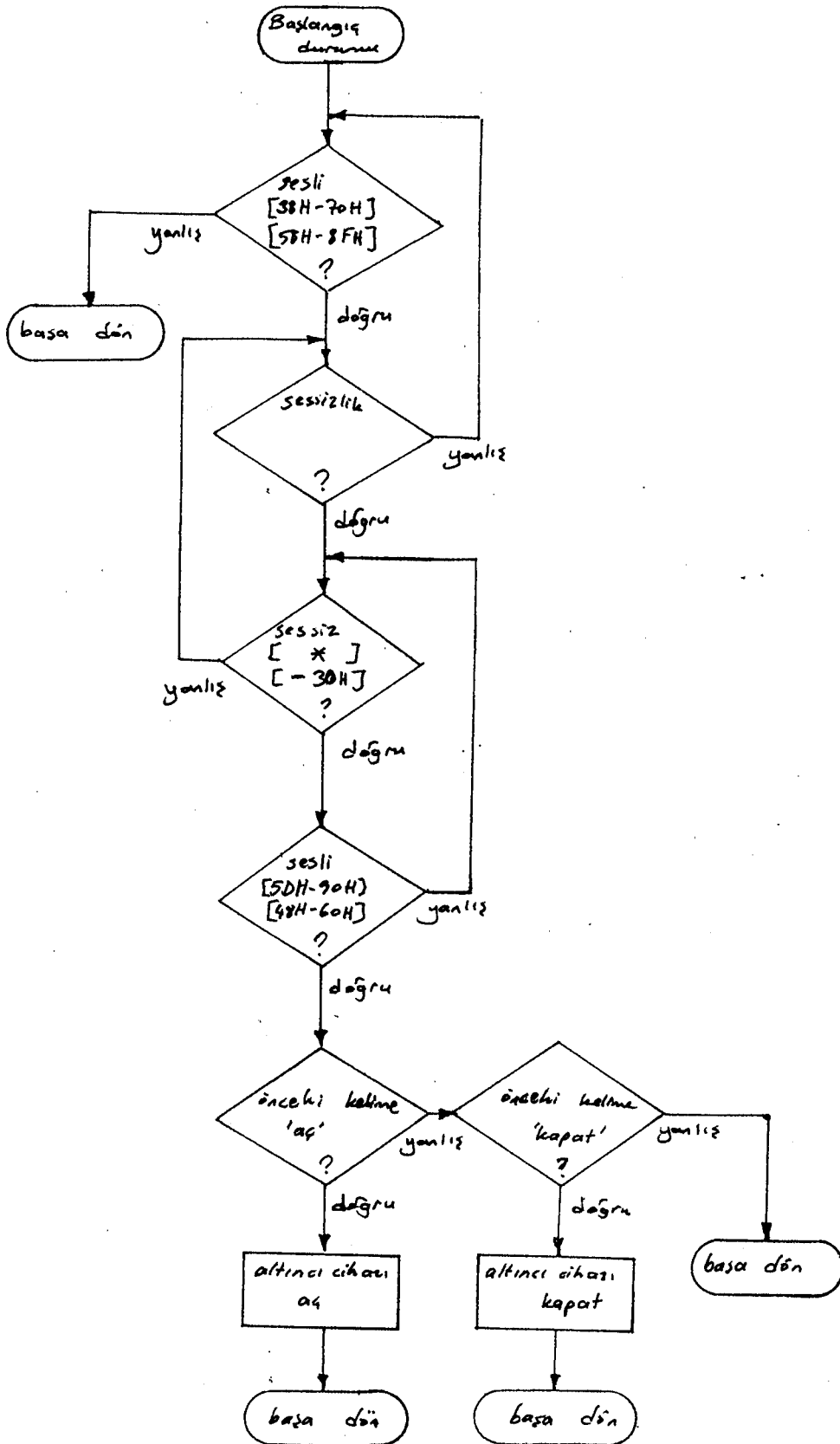
Şekil 4.13 'uç' Kelimesine ait akış şeması



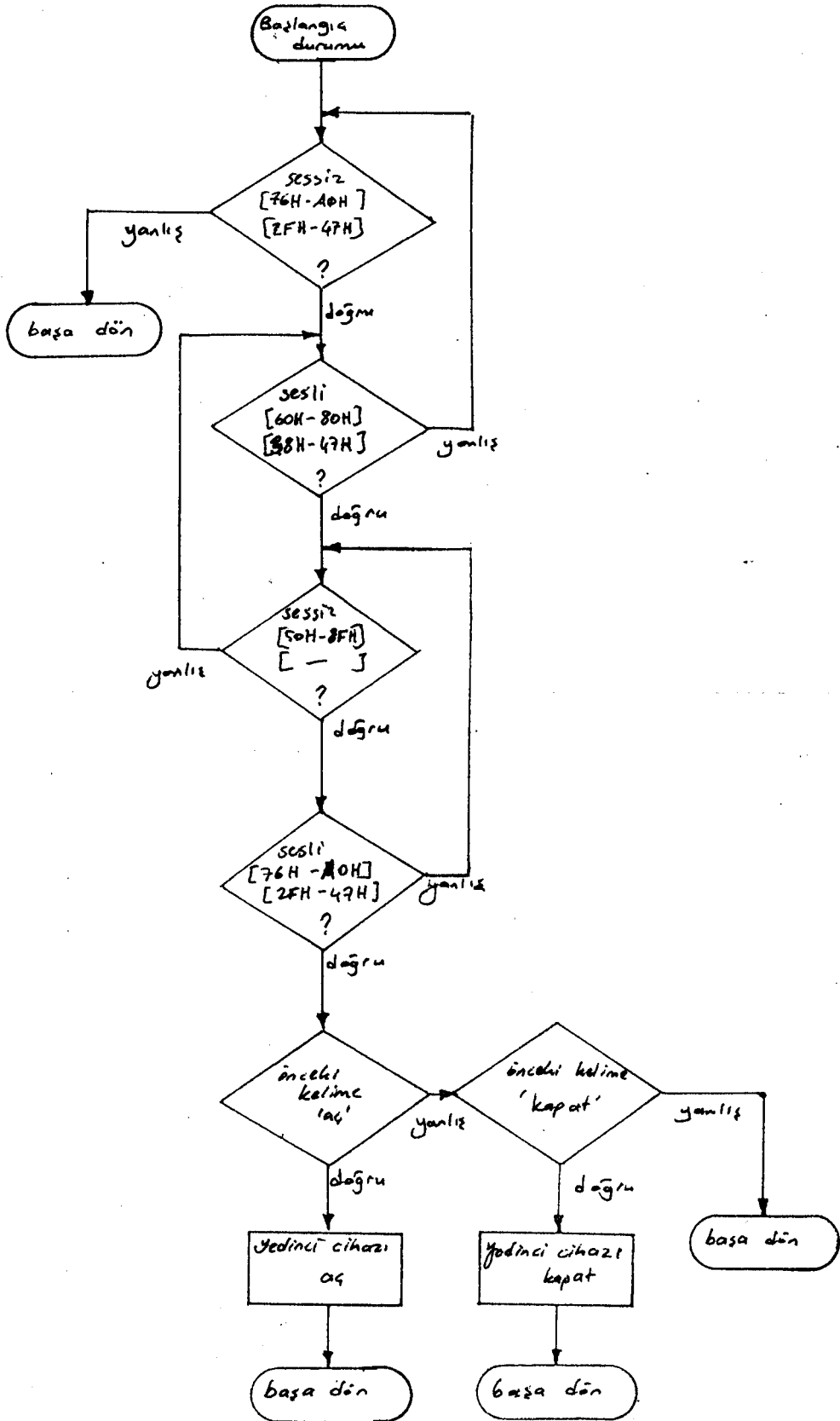
Sekil 4.14 'dört' kelimesine ait akış şeması



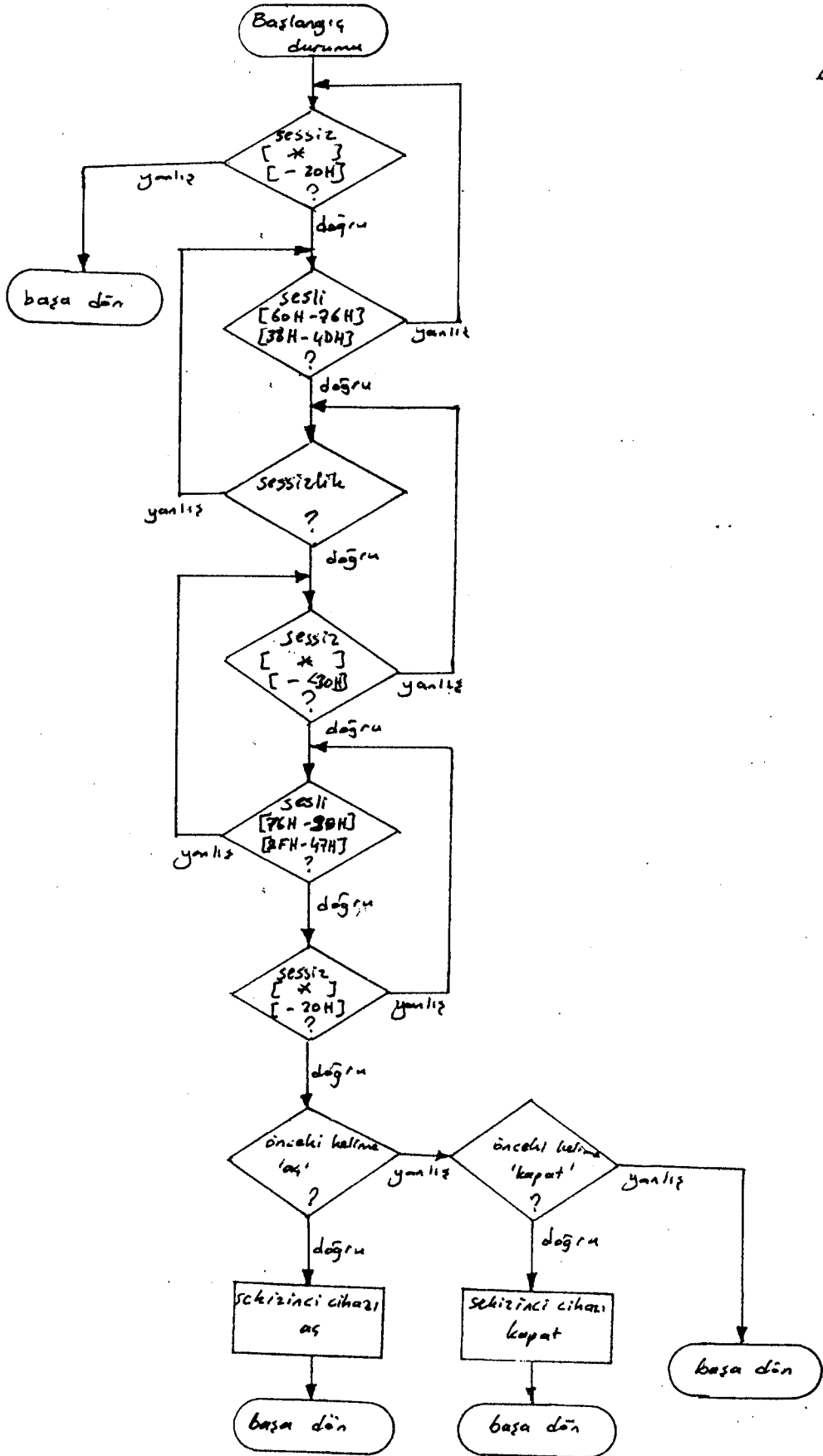
Sekil 4.15 'beş' Kelimesine ait akış şeması



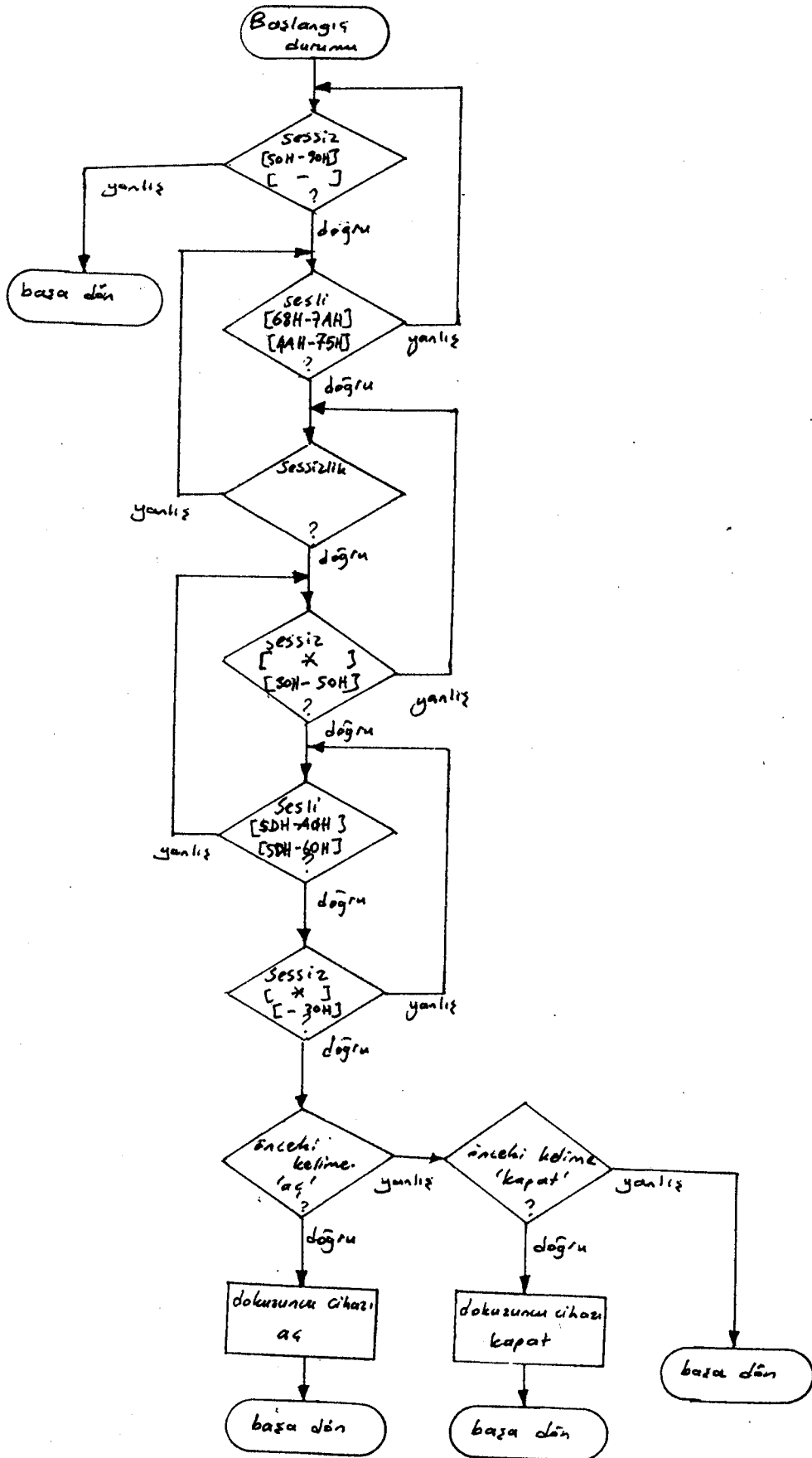
Sekil 4.16 'altı' kelimesine ait akış şeması



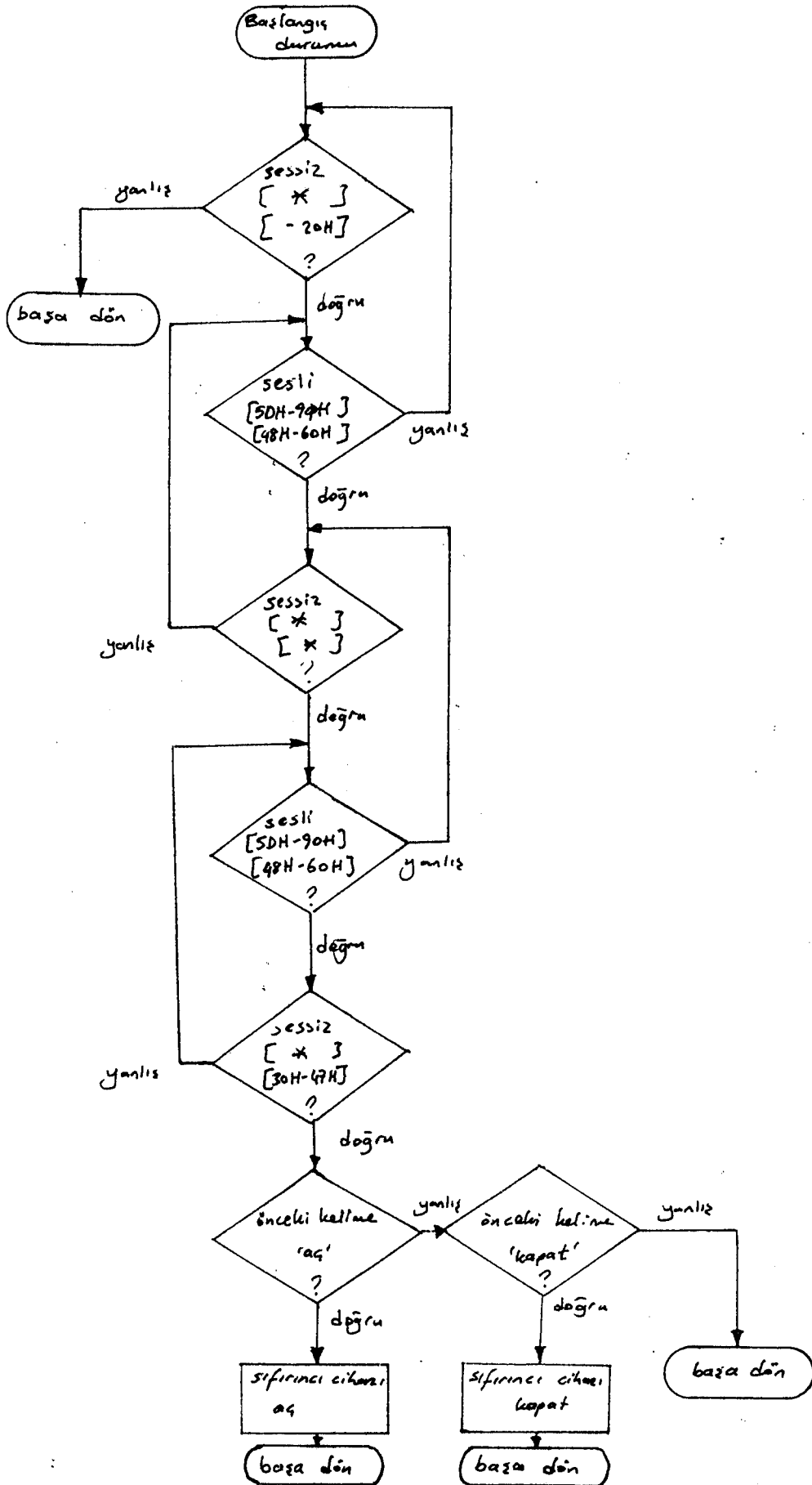
Şekil 4.17 'yedi' kelimesine ait akış şeması



Sekil 4.18 'sekiz' Kelimesine ait akış şeması

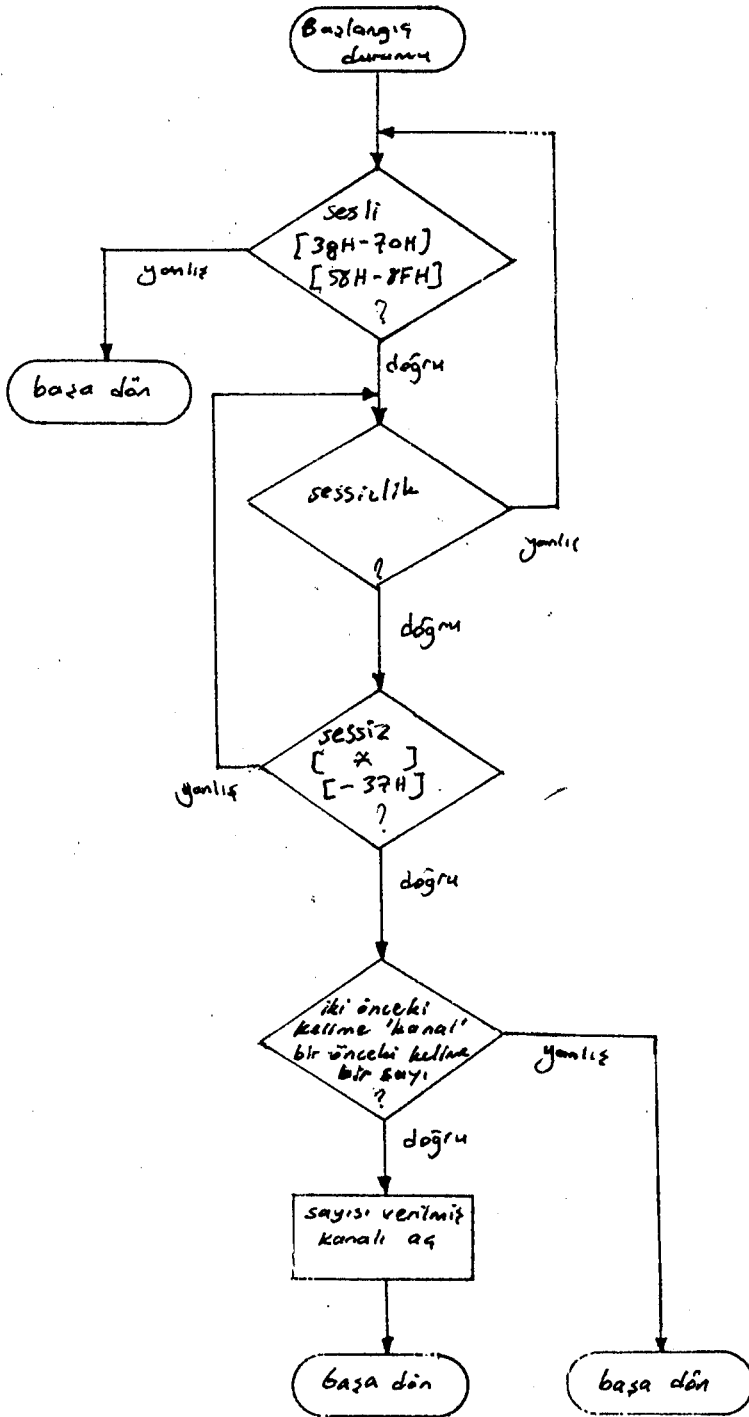


Sekil 4.19 'dokuz' Kelimesine ait akış şeması

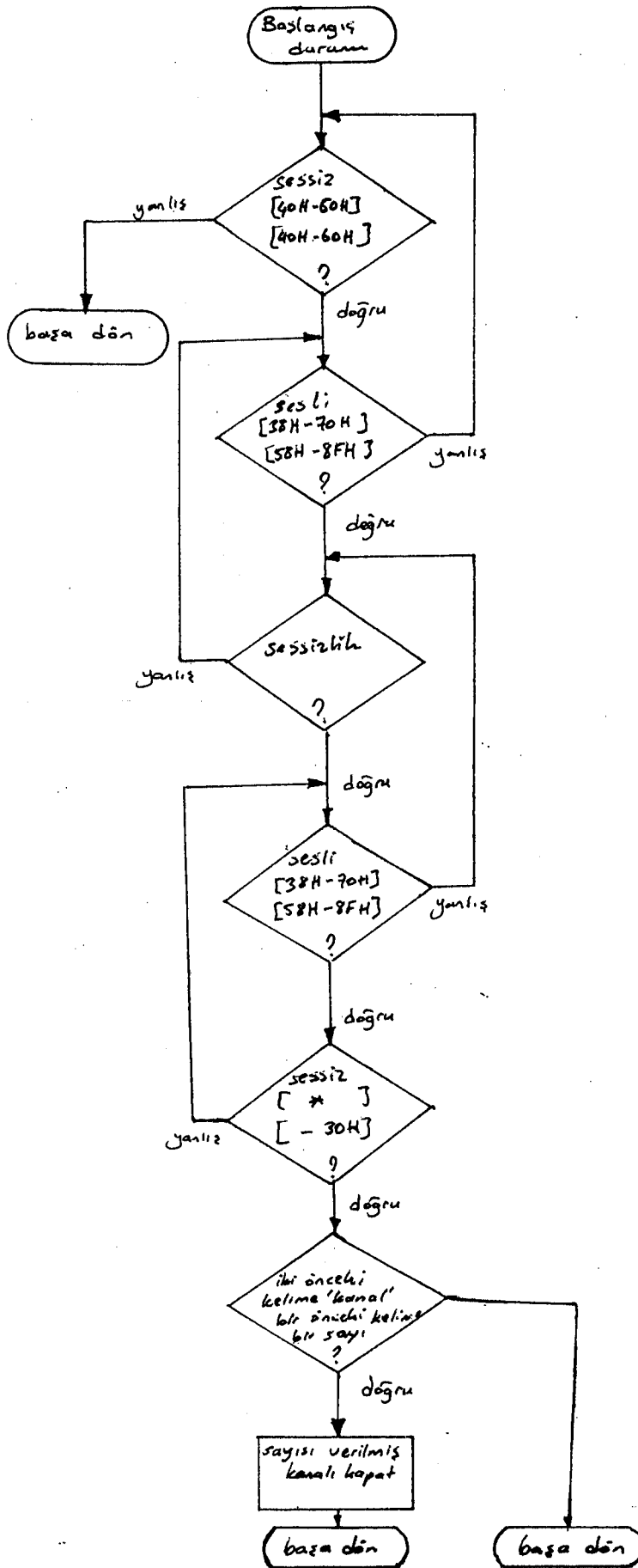


Sekil 4.20 'sıtır' kelimesine ait akış şeması

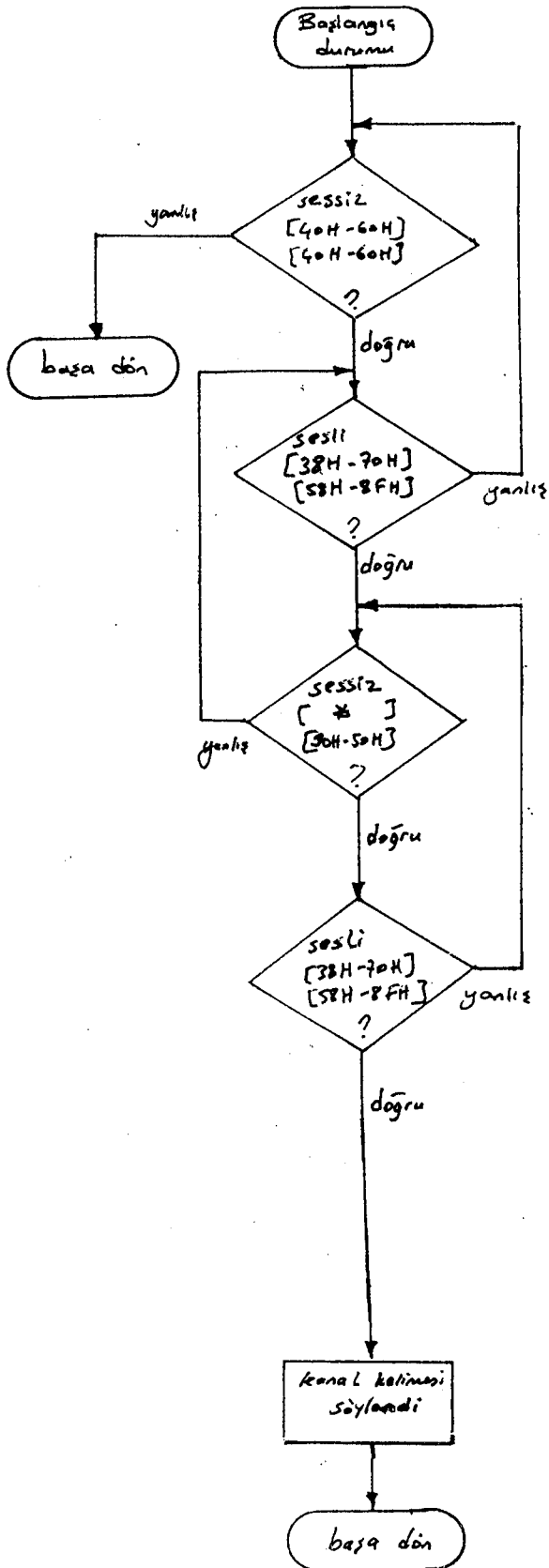




Sekil 4.21 'aç' Kelimesine ait akış şeması



Sekil 4.22 'kapat' Kelimesine ait akış şeması



Sekil 4.23 'kanal' Kelimesine ait akış şeması

## 5 . SONUÇ VE ÖNERİLER

Yapılan çalışmada sıfır geçiş analizi tekniği ile ses tanıma problemi çözülmeye çalışılmış , konuşmacıya bağımlı bir izoleli kelime tanıyan sistem hazırlanmıştır .

Sistem 220 volt ile çalışan bir cihaz için açma , kapama kontrolü yapabilmektedir . Gerekli donanımlar sağlanır, programda ilaveler yapılırsa kontrol edilecek noktaların sayısı arttırılabilir .

Sistemin konuşmacıdan bağımsız olabilmesi için ortalama sıfır geçiş sınırları genişletilebilir. Bu , hata oranını artırabilecektir. Çünkü analog donanım gerekli hassasiyette değildir . Gerekli hassasiyeti sağlamak mümkündür. Patlamalı ve düşük frekanssa sahip b , p , f gibi sesleri daha iyi tanımak amacıyla bir alçak geçiren ( 0 - 150 Hz ) filtreye, yüksek frekansa sahip bileşenlerin en baskınını değil de tümünü tanıyabilmek için ise , birkaç band geçiren filtreye ihtiyaç vardır. Bu band geçiren filtreler 550 Hz - 5000 Hz arasında birbirini takip edecek şekilde düzenlenmelidir. Böylece sıfır geçişlerin geldiği veriler artacak daha ayrıntılı bir analize geçilecektir.

Sistemin sürekli kelime tanıyan bir sisteme dönüştürülebilmesi için uygun hızda ( 3 -4 MHz saatle çalışan ) bir mikrobilgisayar kullanılabilir.

Sistemi sadece kendi sesine ayarlamak isteyen kiři parametre sınırlarıyla oynayabilir. Basic dilinin yavaşlığından kurtulmak için alt programlar tek tek makine dilinde yazılabilir.

## KAYNAKLAR DİZİNİ

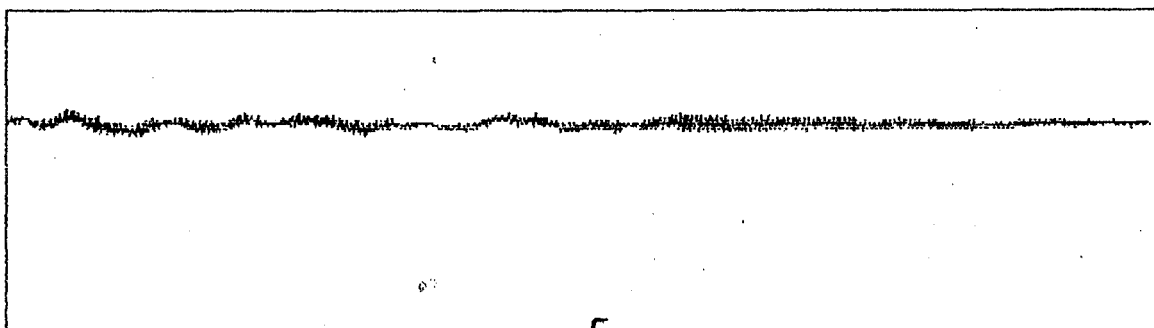
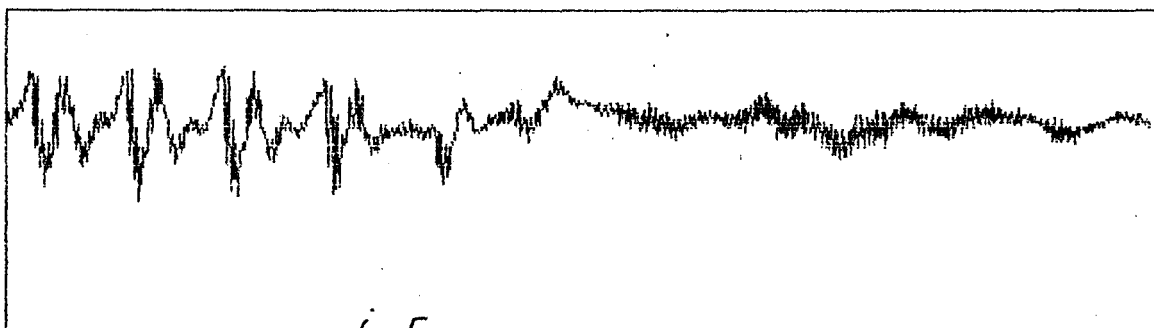
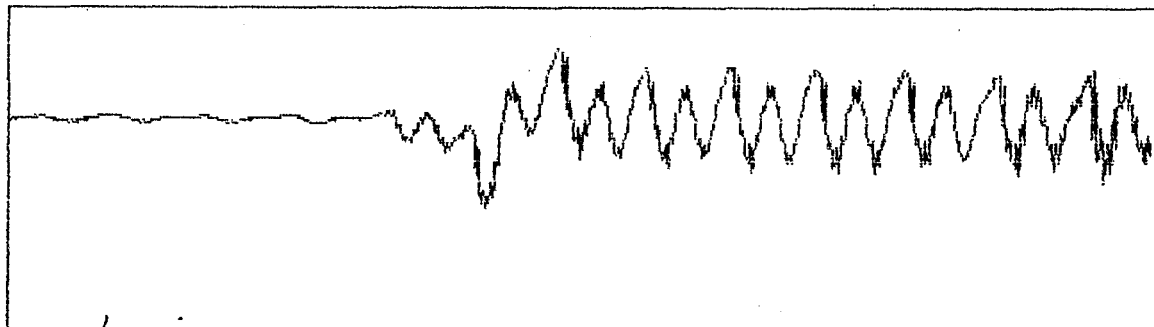
1. Berlin , H.M. , 1981 , Design of Active Filters , With Experiments , Howard W. Sams & Co., Inc.
2. Edizkan , R. , 1987, Amstrad CPC6128 Bilgisayarında Ses Sinyallerinin İncelenebilmesi İçin AraDevre ve Programların Hazırlanması , Bitirme Ödevi , Anadolu Üniversitesi - ESKİŞEHİR
3. Fallside , F., Woods , W.A. 1985 , Computer Speech Processing , Prentice-Hall.
4. Jelinek , F. , March - 1976 , Continuous Speech Recognition by Statistical Methods, IEEE Proc.
5. Martin , T.B., March - 1976, Practical Applications of Voice Input to Machines , IEEE Proceedings.
6. Reddy , D.R., March - 1976 , Speech Recognition by Machine : A Review , IEEE Proceedings.
7. Rosenberg , F. , March - 1976 , Automatic Speaker Verification : A Review , IEEE Proceedings.

## EKLER

EK.1. Tipik Ses Dalga Sekilleri

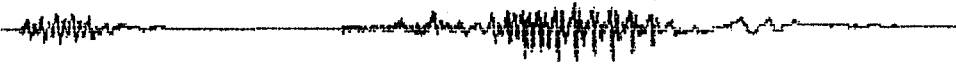
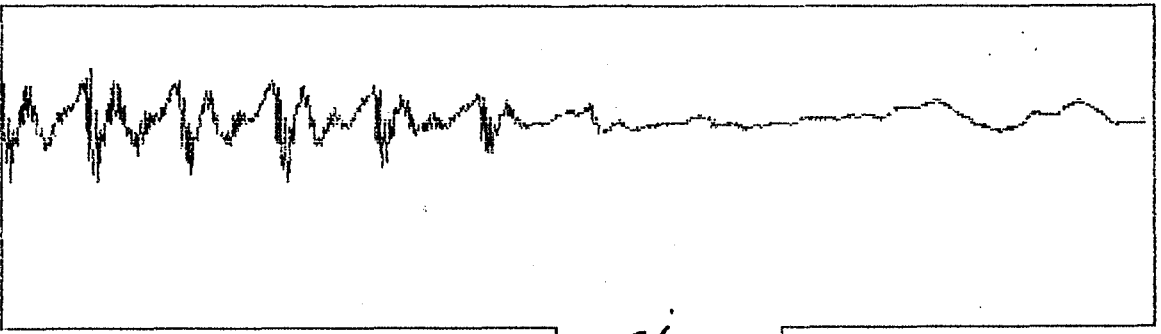
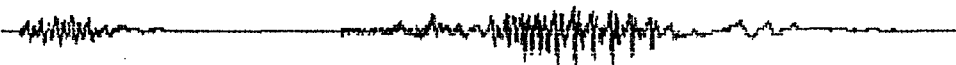
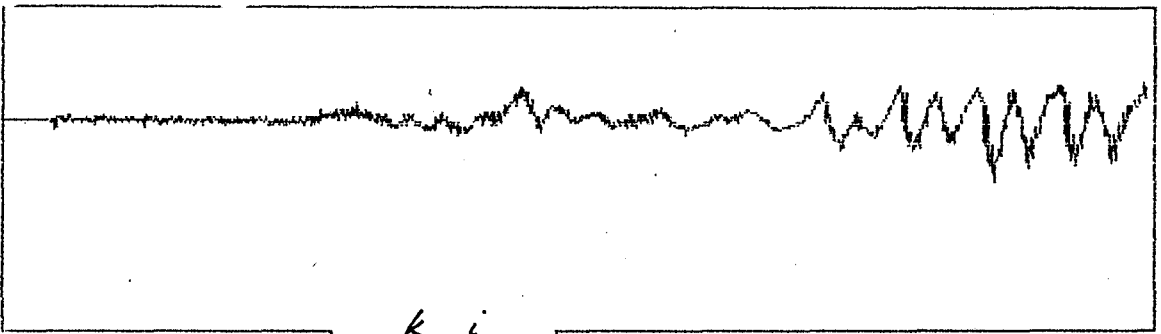
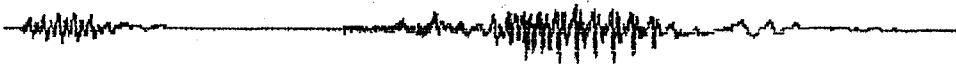
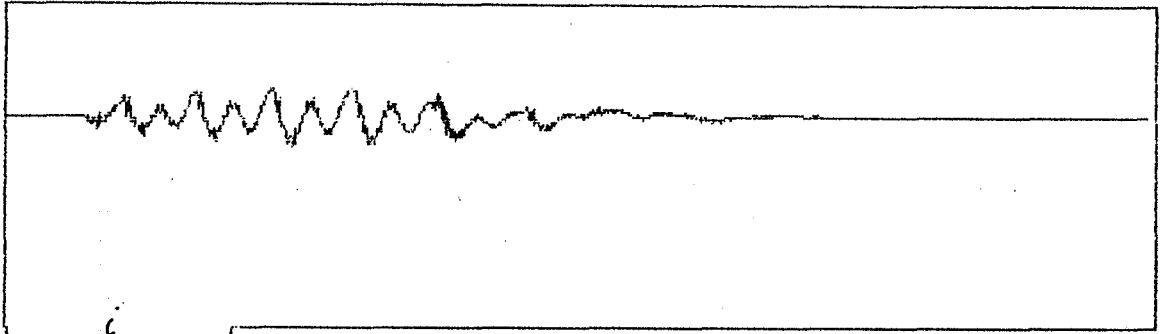
EK.2. 8255 Paralel Uyum Devresi

EK.3. Bilgisayar Programları

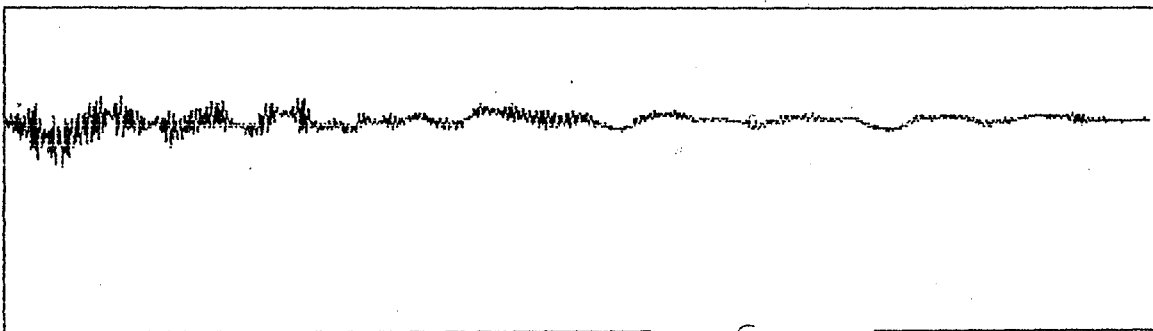
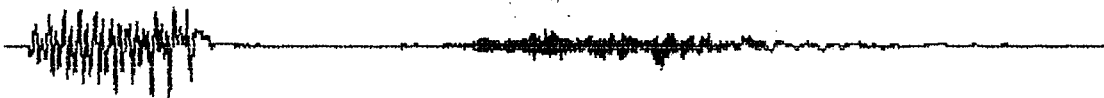
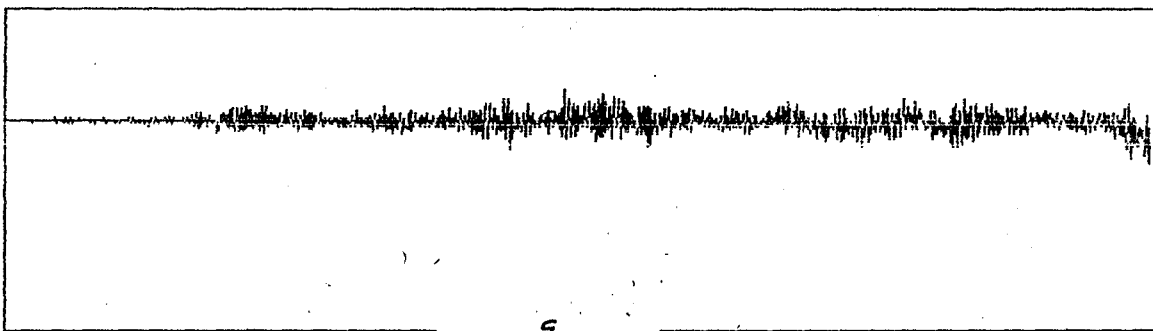
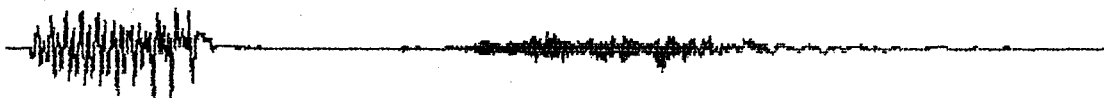
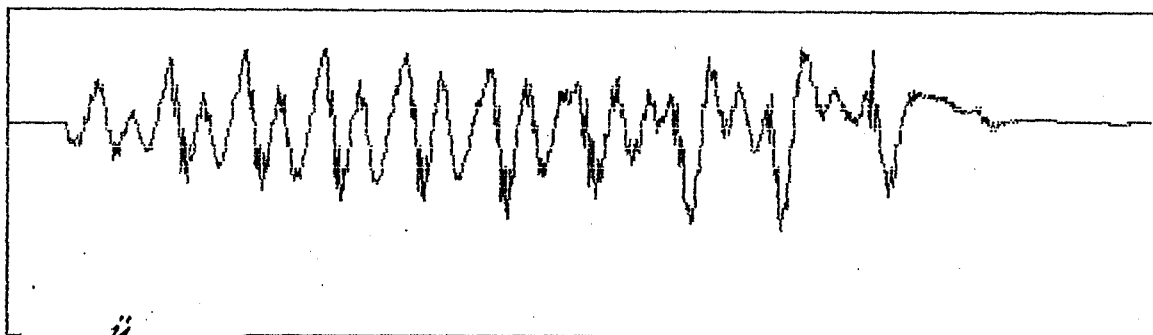


/bir/

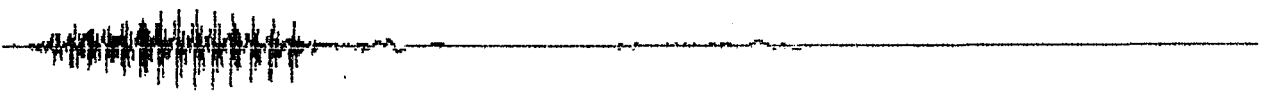
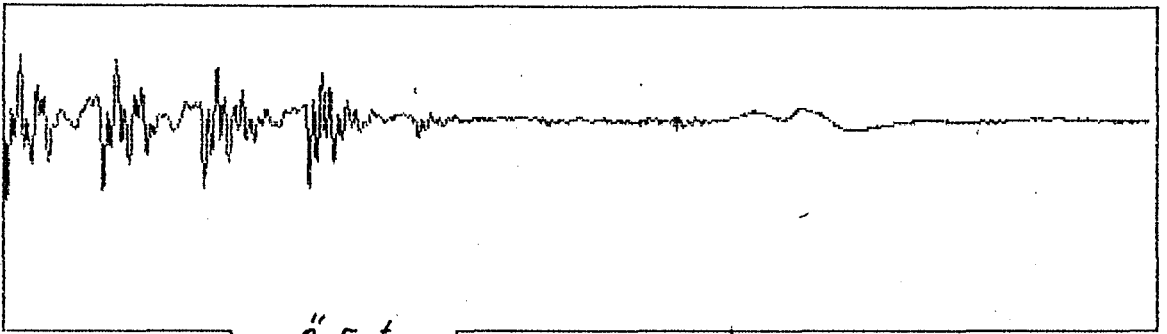
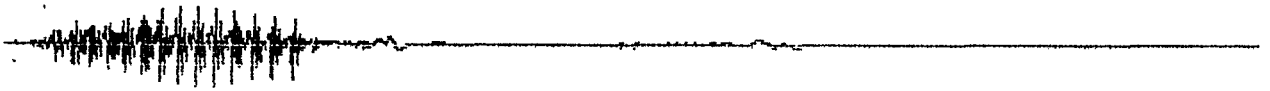
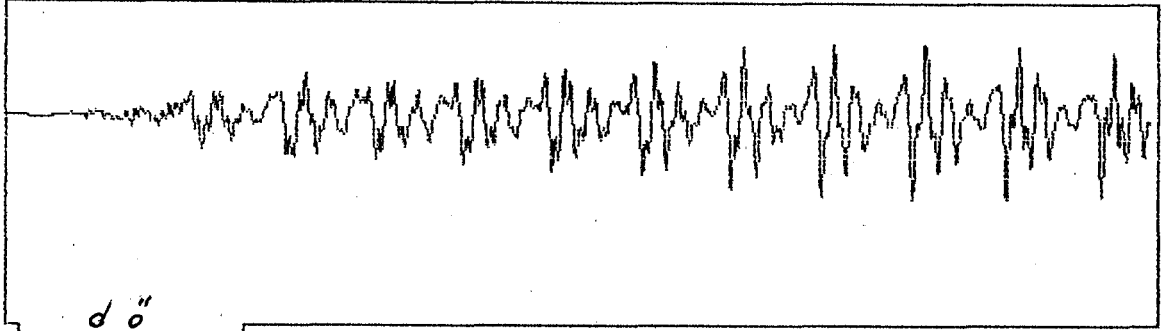




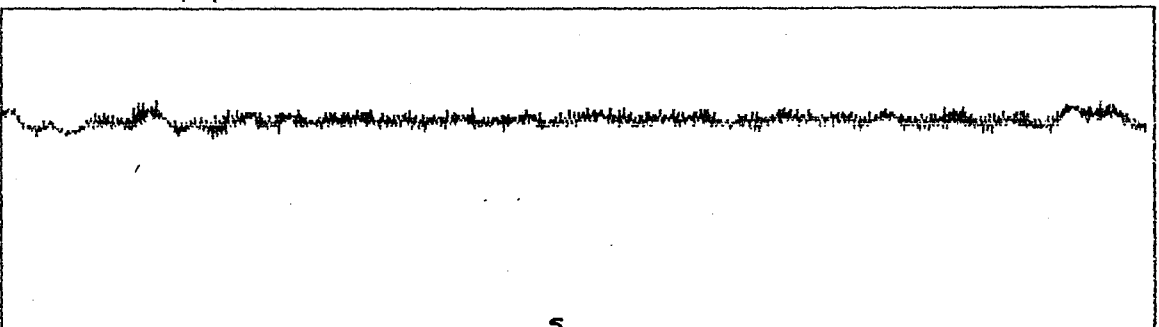
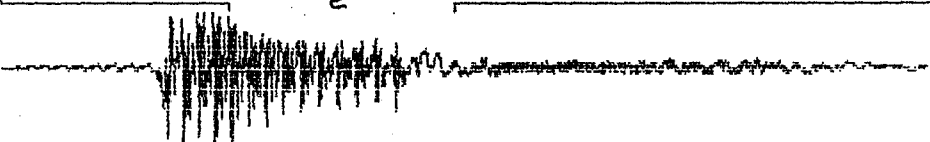
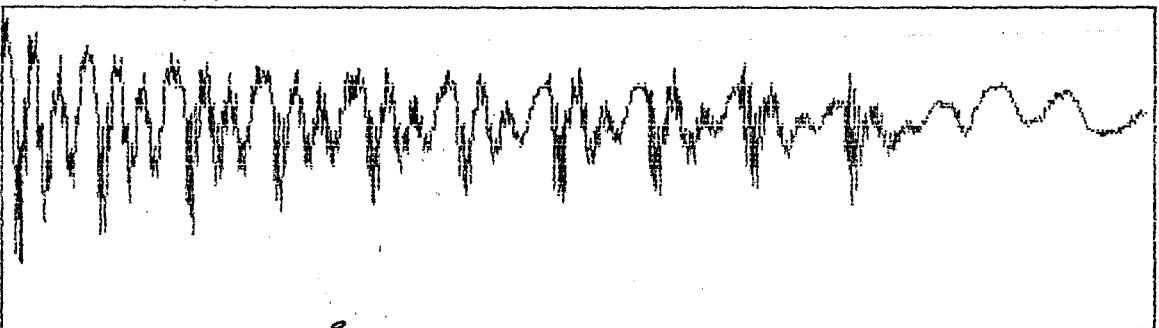
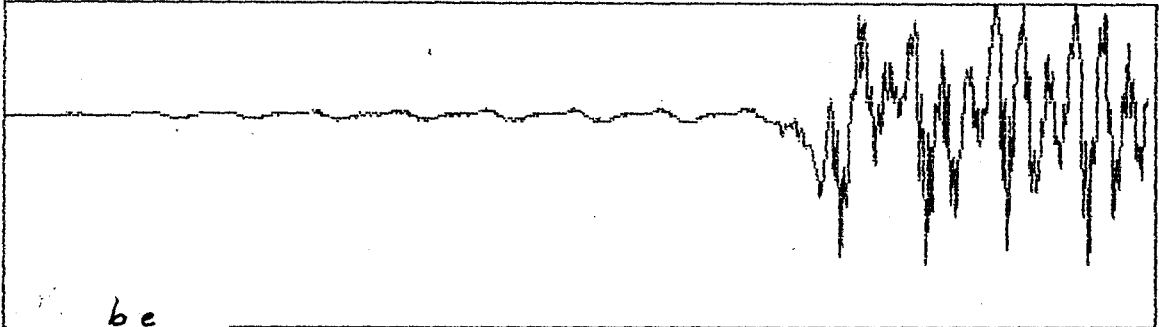
/iki/



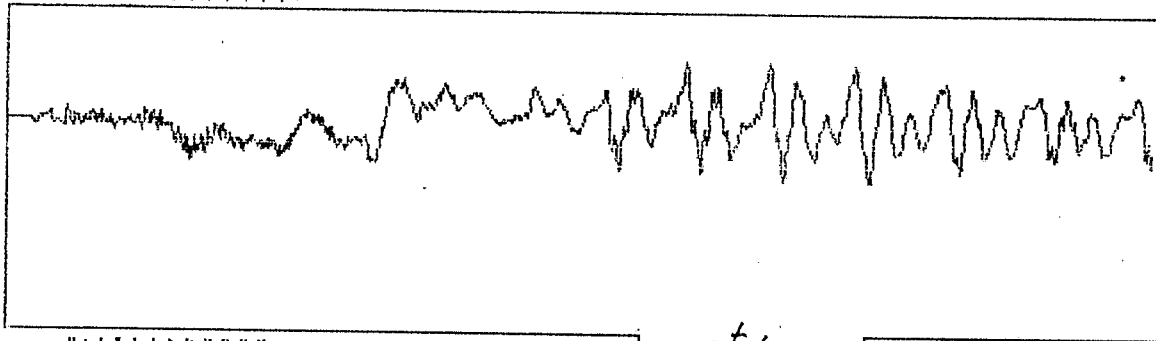
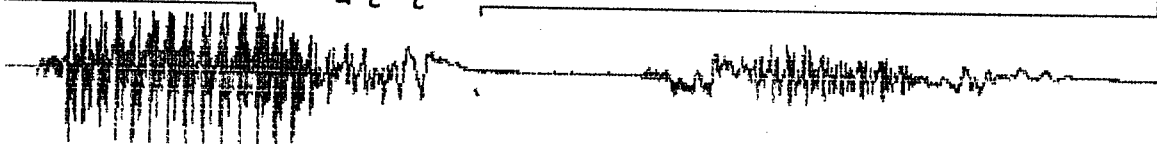
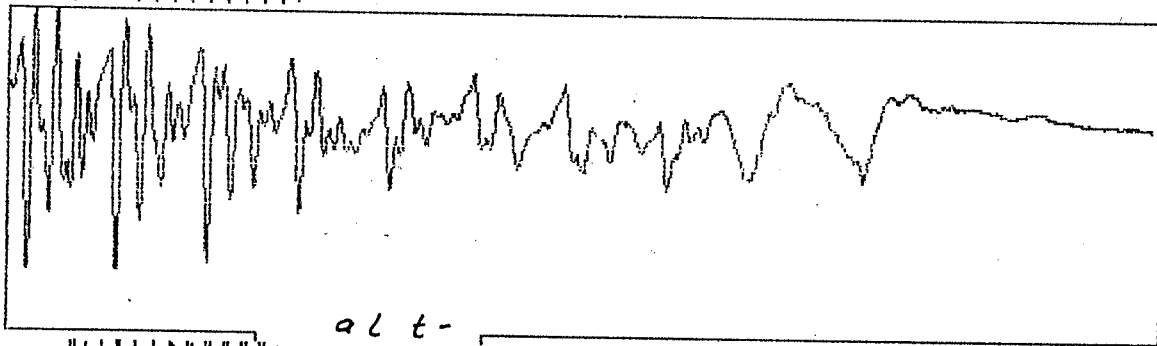
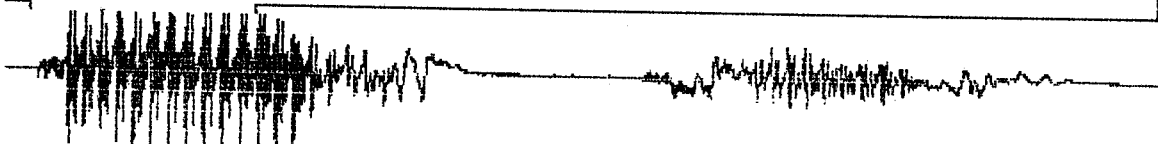
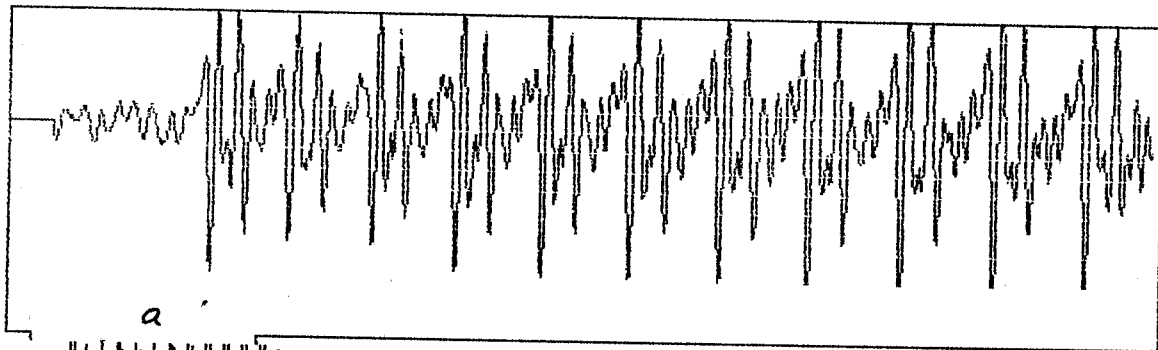
| ü 9 |



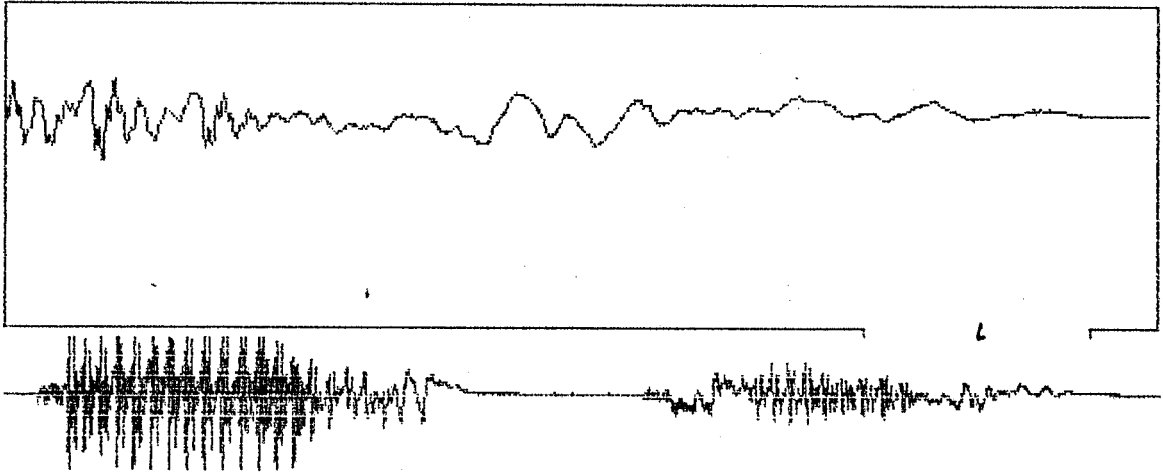
/dört/



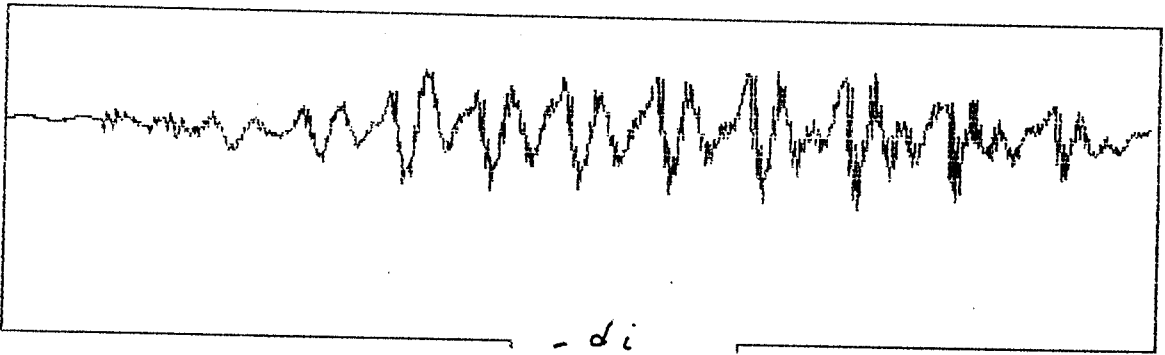
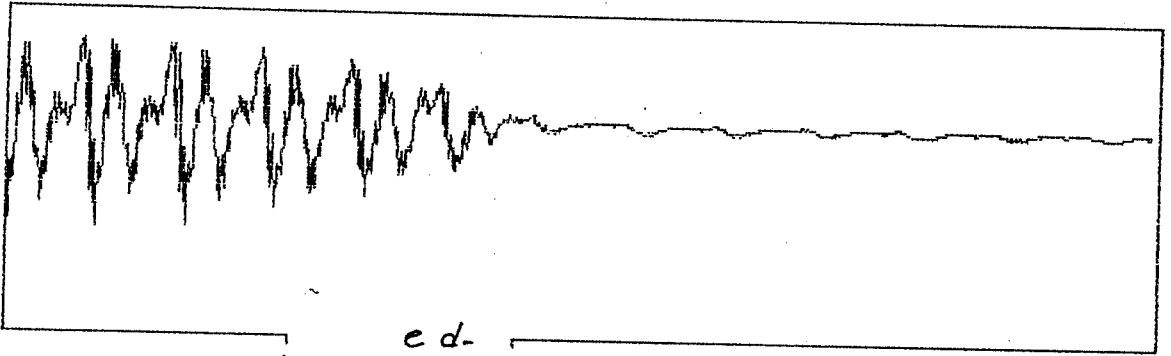
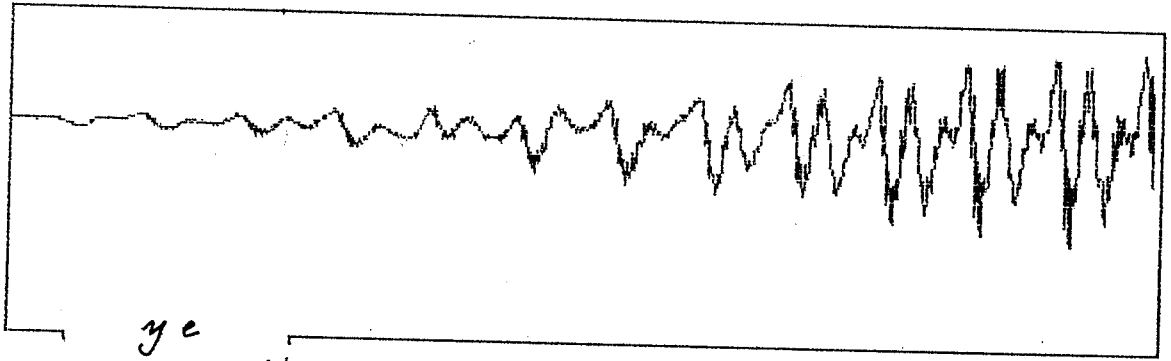
/ bes /



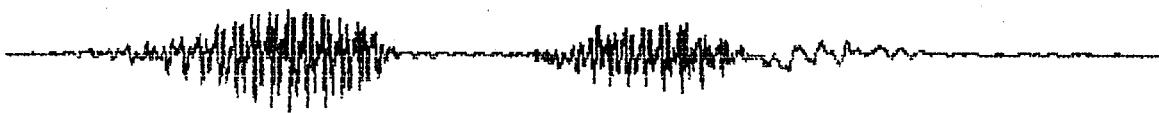
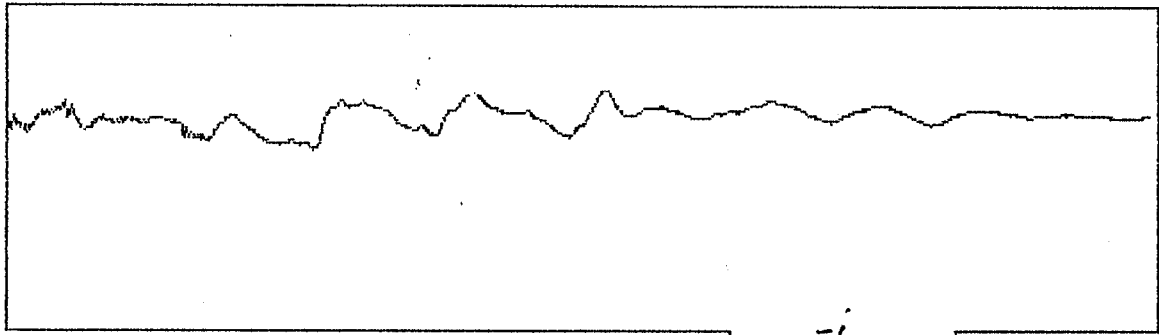
/alt1/



/alt/

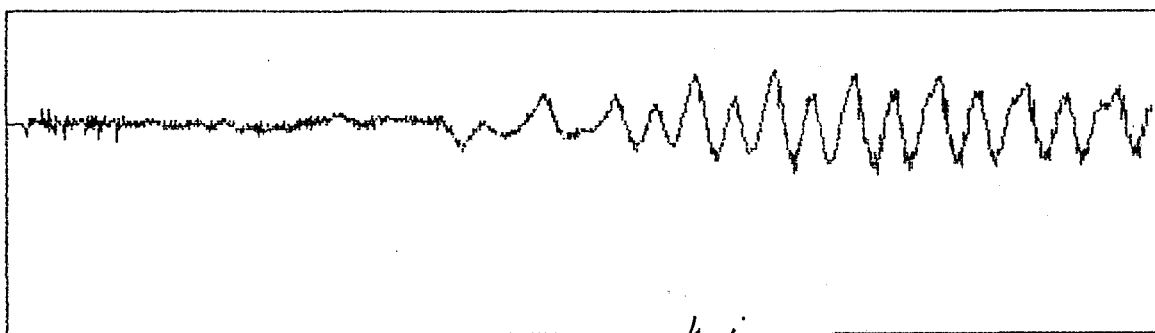
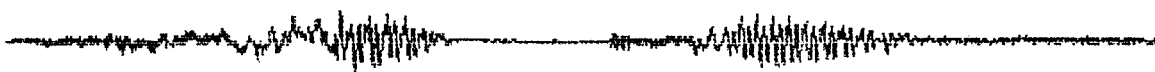
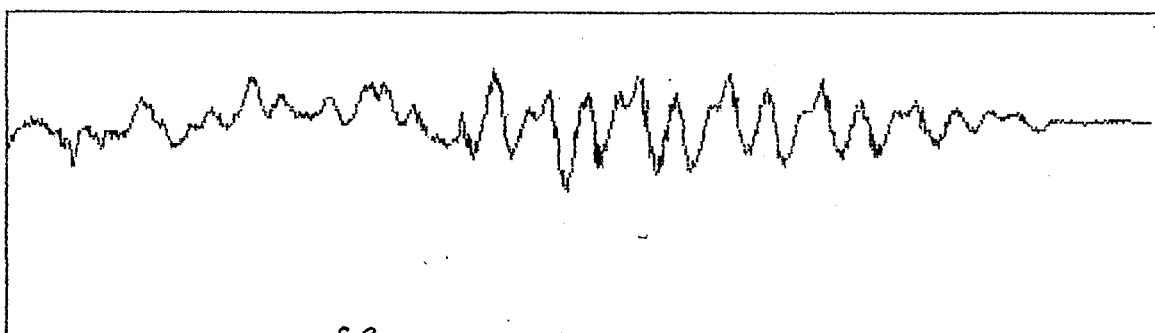
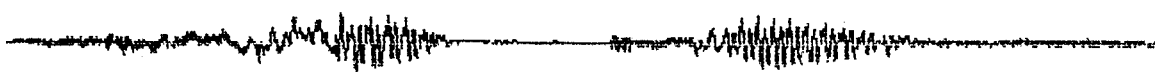
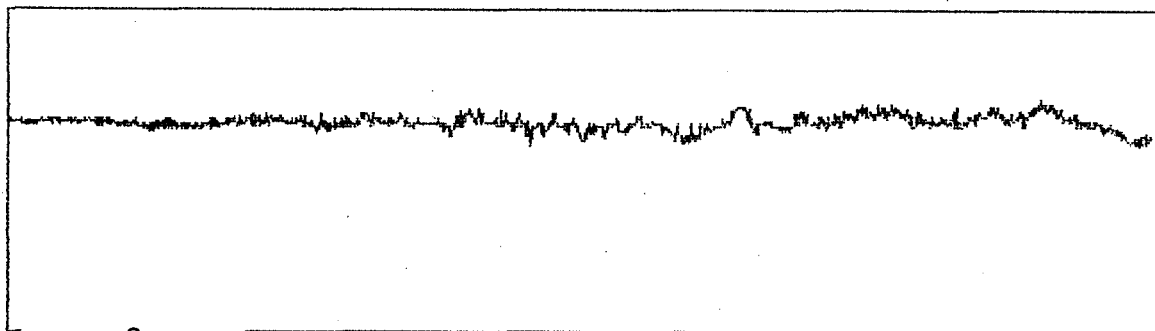


/yedi/

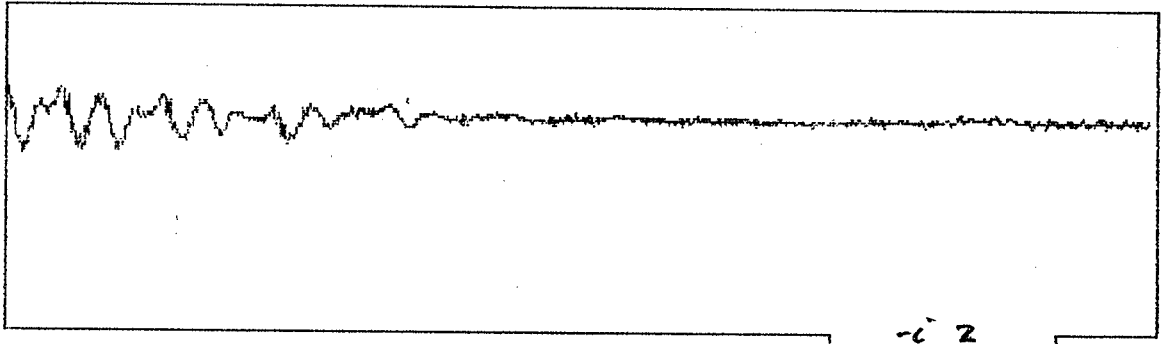


/yedi/



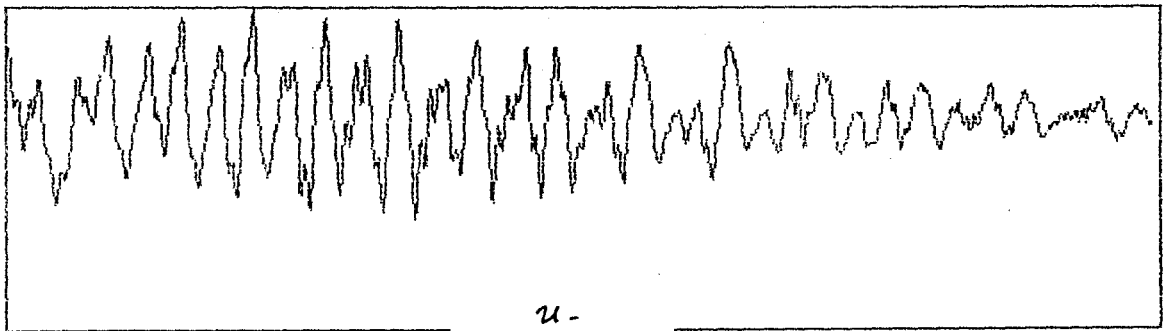
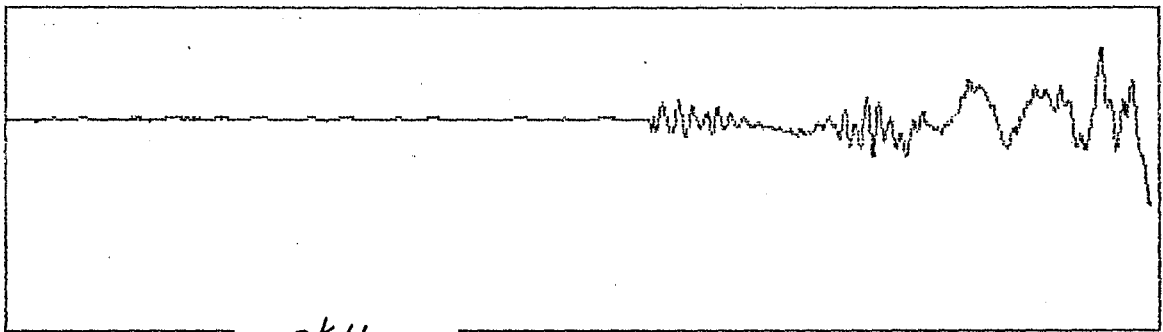
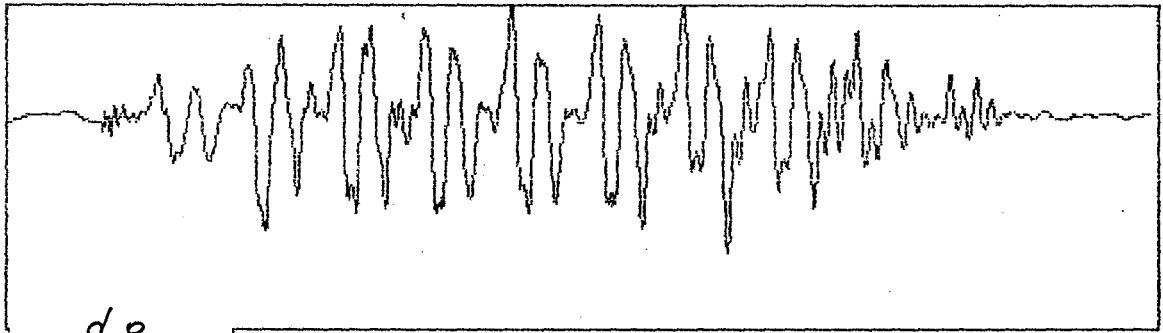


/sekiz/

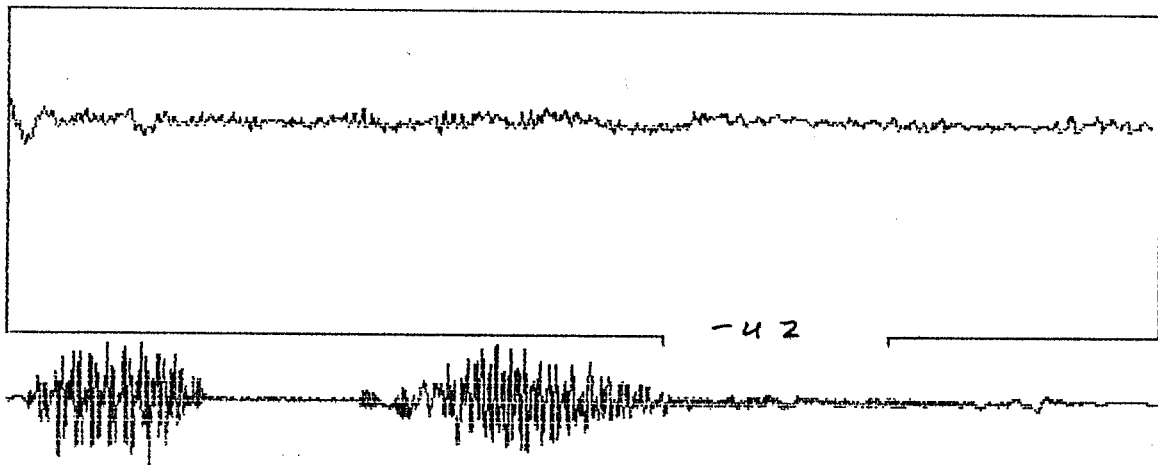


2

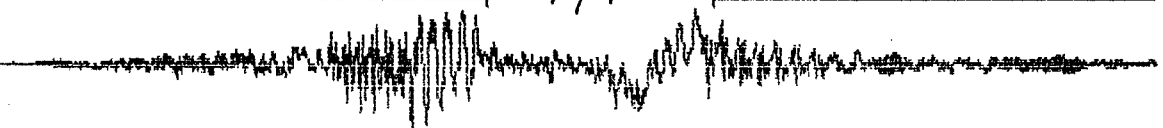
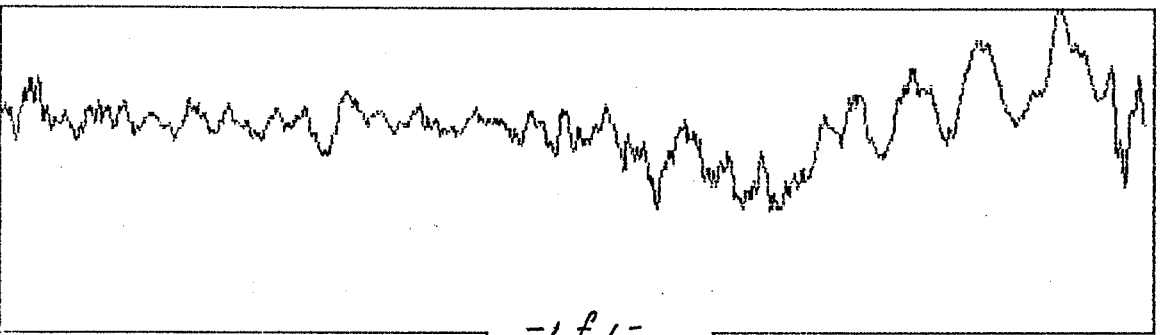
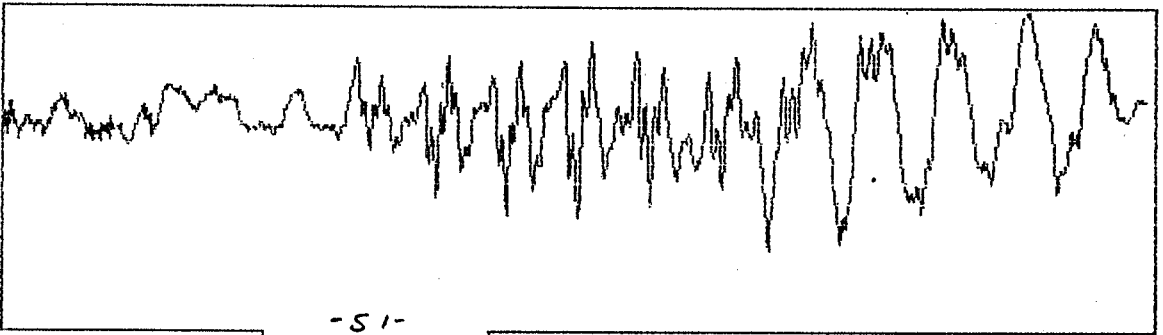
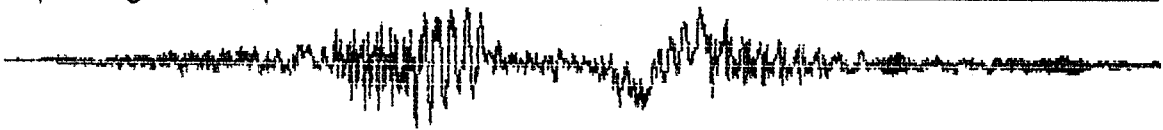
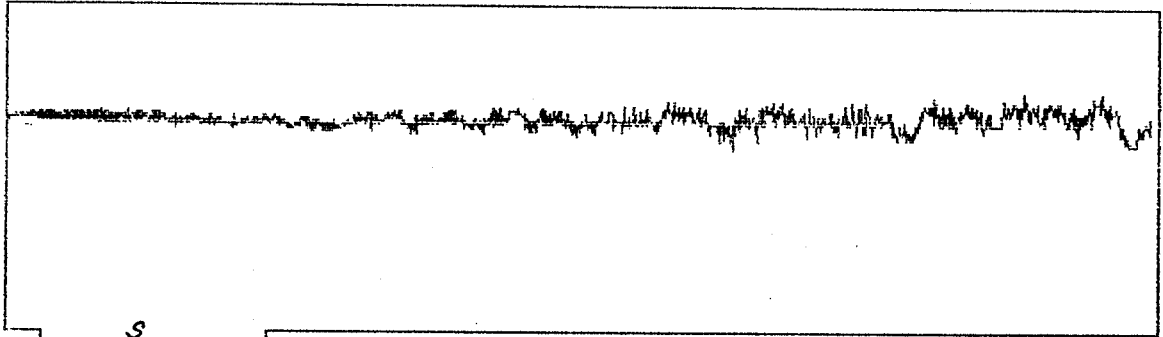
/sekiz/



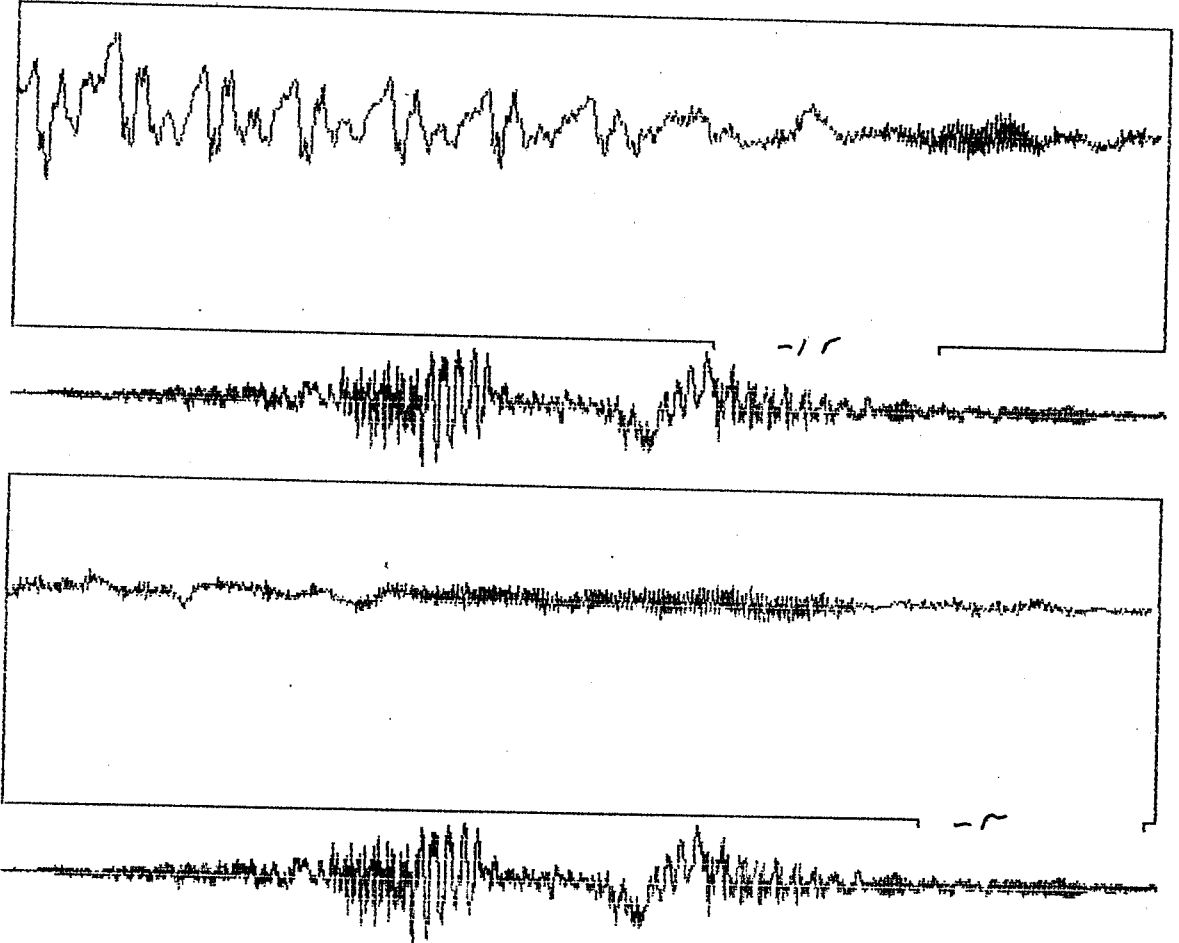
/dokuz/



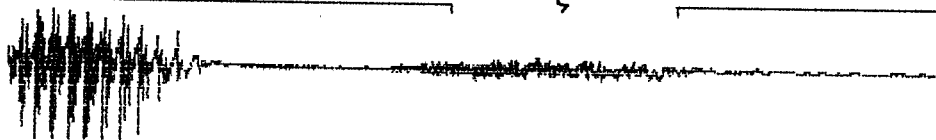
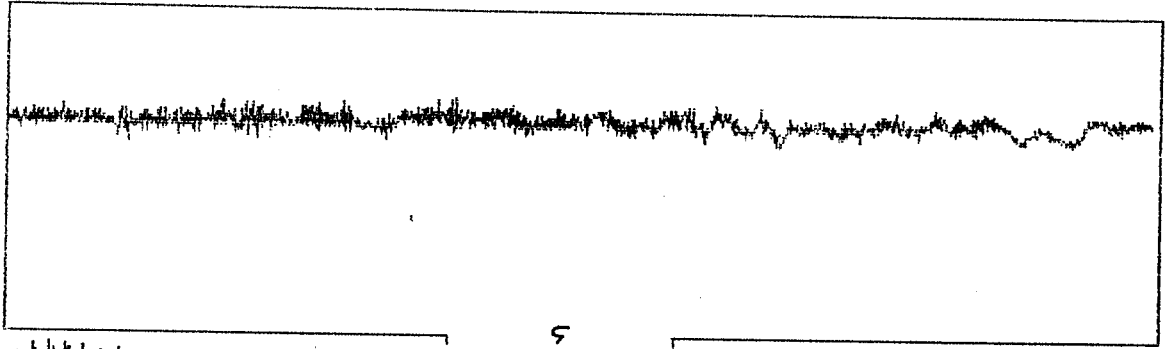
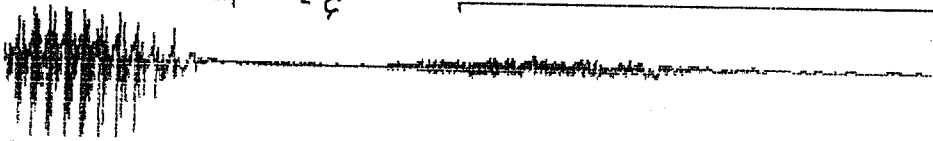
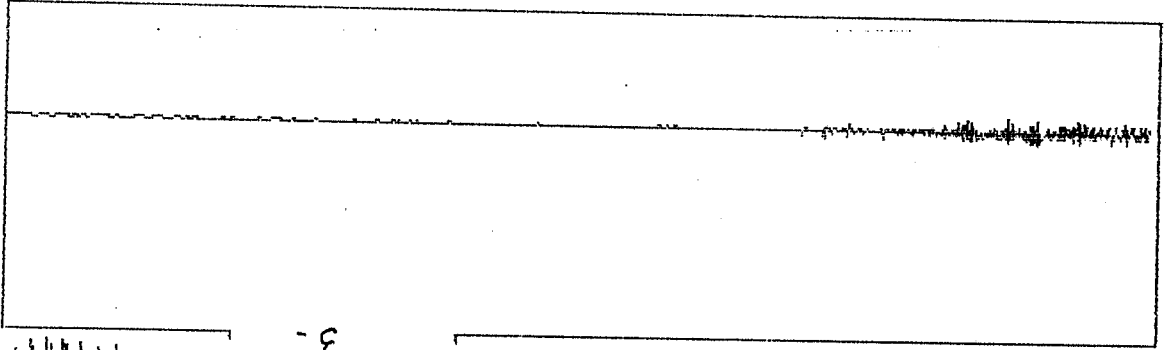
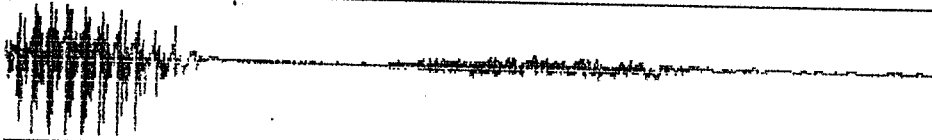
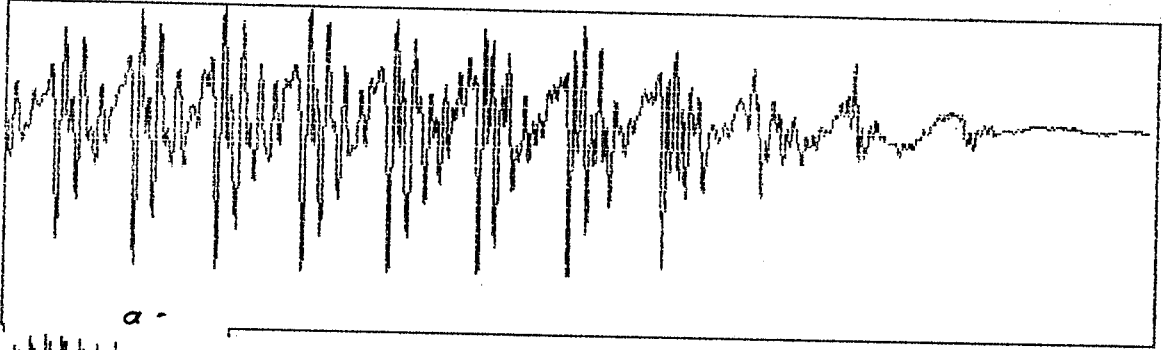
/dokuz/



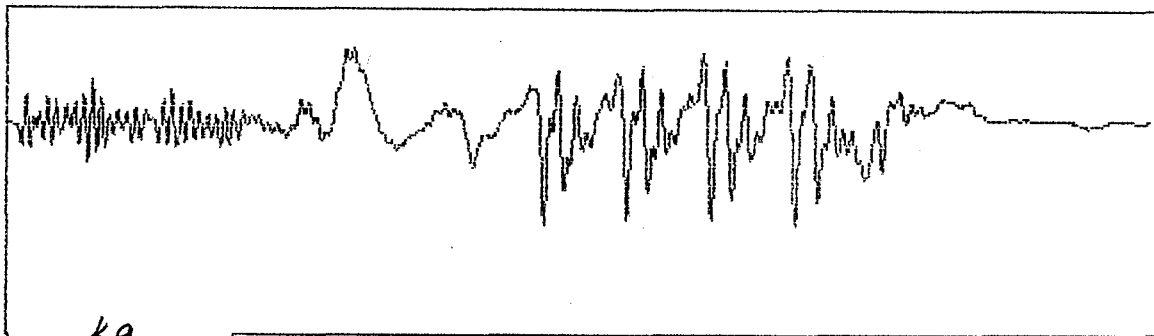
|sifir|



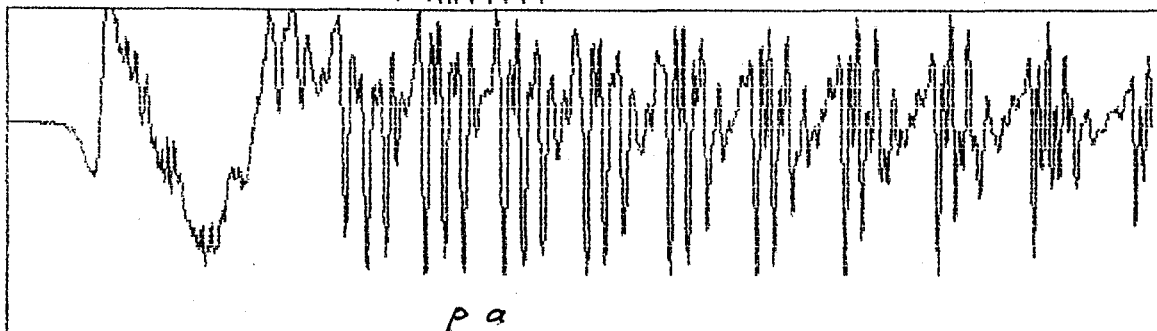
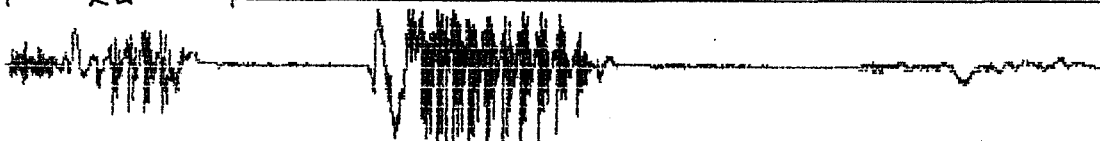
|sifir|



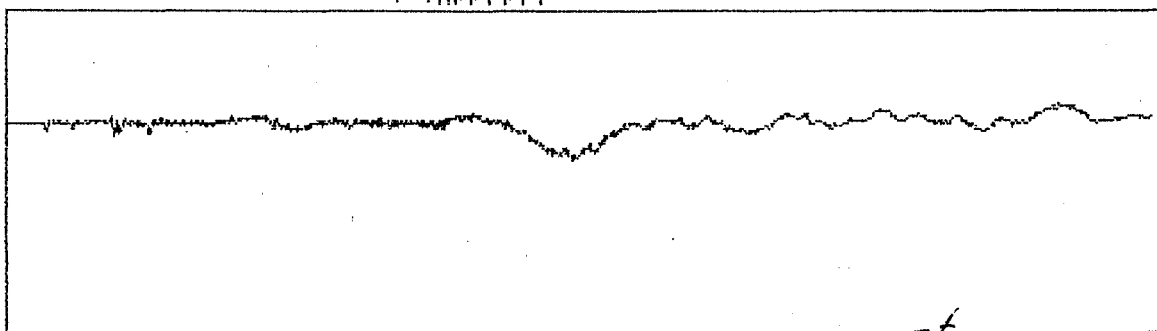
/ a 9 /



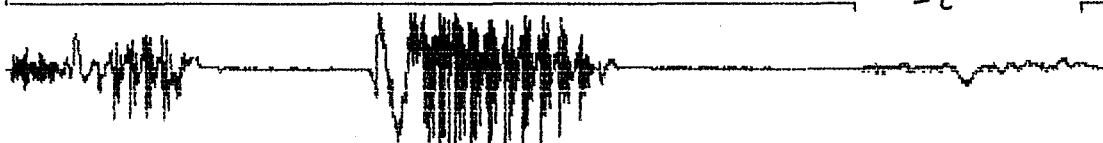
ka



pa

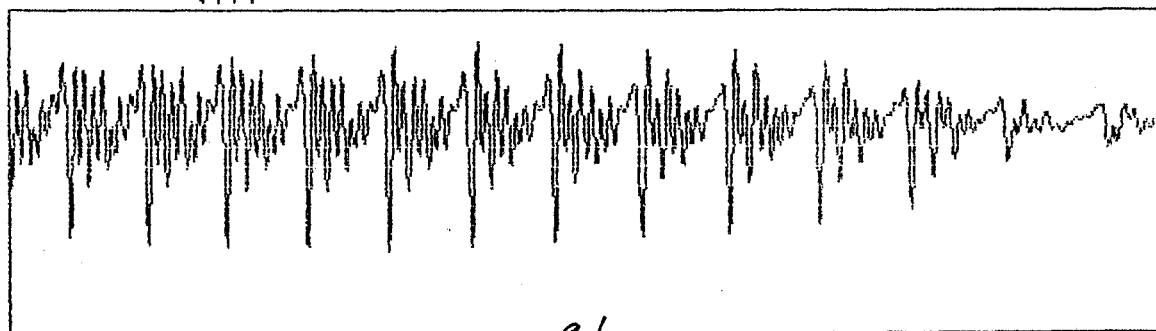
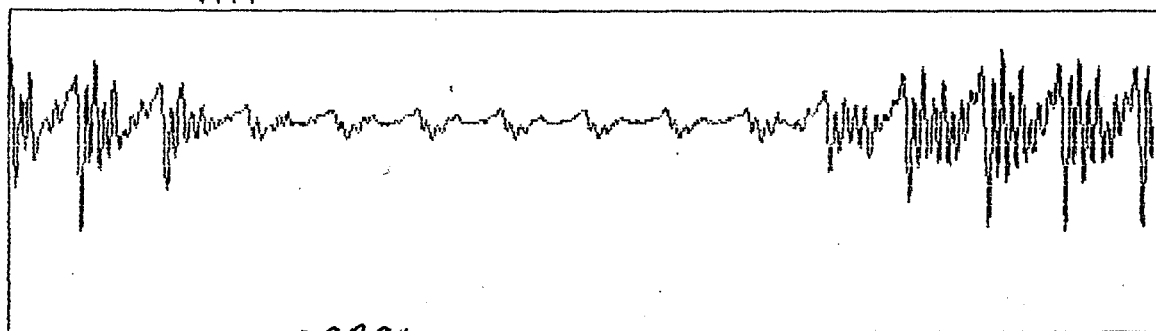
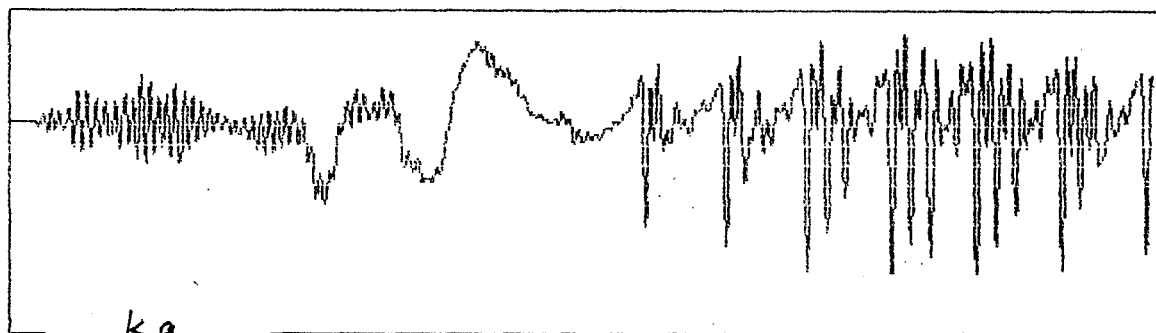


-t

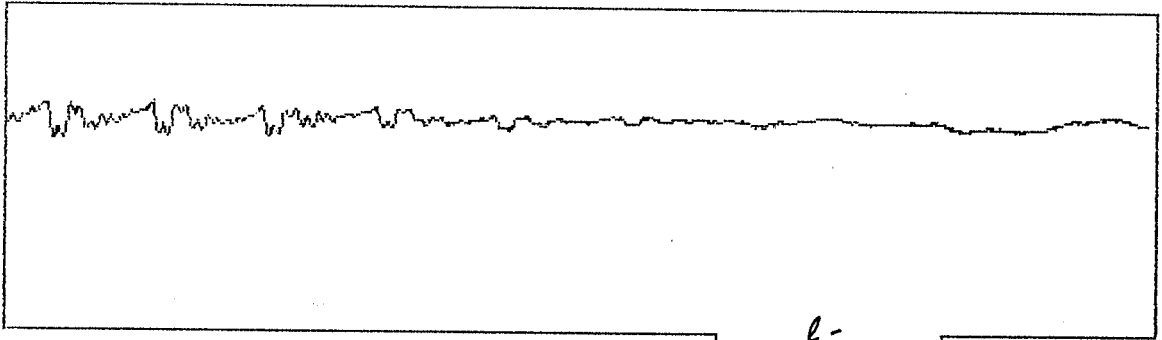


/ kapat /





/kanal/



l-



/kanal/



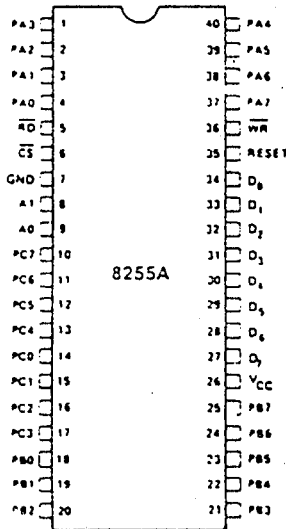
**PRELIMINARY**  
 Notice: This is not a final specification. Some data and pin numbers are subject to change.

# 8255A/8255A-5 PROGRAMMABLE PERIPHERAL INTERFACE

- MCS-85™ Compatible 8255A-5
- 24 Programmable I/O Pins
- Completely TTL Compatible
- Fully Compatible with Intel® Micro-processor Families
- Improved Timing Characteristics
- Direct Bit Set/Reset Capability Easing Control Application Interface
- 40-Pin Dual In-Line Package
- Reduces System Package Count
- Improved DC Driving Capability

The Intel® 8255A is a general purpose programmable I/O device designed for use with Intel® microprocessors. It has 24 I/O pins which may be individually programmed in 2 groups of 12 and used in 3 major modes of operation. In the first mode (MODE 0), each group of 12 I/O pins may be programmed in sets of 4 to be input or output. In MODE 1, the second mode, each group may be programmed to have 8 lines of input or output. Of the remaining 4 pins, 3 are used for handshaking and interrupt control signals. The third mode of operation (MODE 2) is a bidirectional bus mode which uses 8 lines for a bidirectional bus, and 5 lines, borrowing one from the other group, for handshaking.

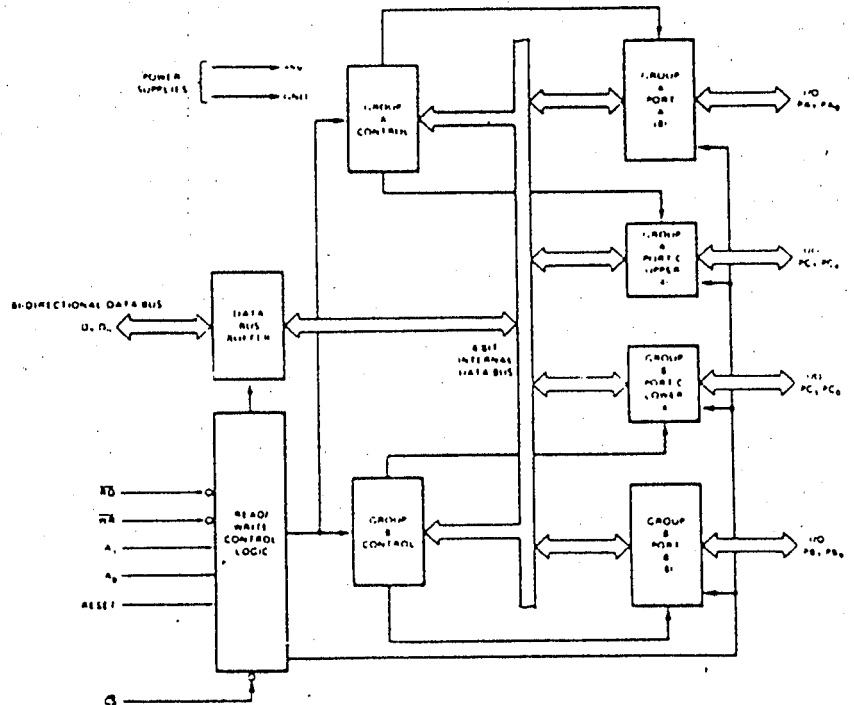
### PIN CONFIGURATION



### PIN NAMES

D <sub>7</sub> -D <sub>0</sub>	DATA BUS (BI DIRECTIONAL)
RESET	RESET INPUT
CS	CHIP SELECT
RD	READ INPUT
WR	WRITE INPUT
A <sub>0</sub> , A <sub>1</sub>	PORT ADDRESS
PA <sub>7</sub> -PA <sub>0</sub>	PORT A (BIT)
PB <sub>7</sub> -PB <sub>0</sub>	PORT B (BIT)
PC <sub>7</sub> -PC <sub>0</sub>	PORT C (BIT)
V <sub>CC</sub>	+5 VOLTS
GND	0 VOLTS

### 8255A BLOCK DIAGRAM



## 8255A FUNCTIONAL DESCRIPTION

### General

The 8255A is a programmable peripheral interface (PPI) device designed for use in Intel® microcomputer systems. Its function is that of a general purpose I/O component to interface peripheral equipment to the microcomputer system bus. The functional configuration of the 8255A is programmed by the system software so that normally no external logic is necessary to interface peripheral devices or structures.

### Data Bus Buffer

This 3-state bidirectional 8-bit buffer is used to interface the 8255A to the system data bus. Data is transmitted or received by the buffer upon execution of input or output instructions by the CPU. Control words and status information are also transferred through the data bus buffer.

### Read/Write and Control Logic

The function of this block is to manage all of the internal and external transfers of both Data and Control or Status words. It accepts inputs from the CPU Address and Control busses and in turn, issues commands to both of the Control Groups.

### (CS)

Chip Select. A "low" on this input pin enables the communication between the 8255A and the CPU.

### (RD)

Read. A "low" on this input pin enables the 8255A to send the data or status information to the CPU on the data bus. In essence, it allows the CPU to "read from" the 8255A.

### (WR)

Write. A "low" on this input pin enables the CPU to write data or control words into the 8255A.

### (A<sub>0</sub> and A<sub>1</sub>)

Port Select 0 and Port Select 1. These input signals, in conjunction with the RD and WR inputs, control the selection of one of the three ports or the control word registers. They are normally connected to the least significant bits of the address bus (A<sub>0</sub> and A<sub>1</sub>).

## 8255A BASIC OPERATION

A <sub>1</sub>	A <sub>0</sub>	RD	WR	CS	INPUT OPERATION (READ)
0	0	0	1	0	PORT A ← DATA BUS
0	1	0	1	0	PORT B ← DATA BUS
1	0	0	1	0	PORT C ← DATA BUS
					OUTPUT OPERATION (WRITE)
0	0	1	0	0	DATA BUS → PORT A
0	1	1	0	0	DATA BUS → PORT B
1	0	1	0	0	DATA BUS → PORT C
1	1	1	0	0	DATA BUS → CONTROL
					DISABLE FUNCTION
X	X	X	X	1	DATA BUS → 3-STATE
1	1	0	1	0	ILLEGAL CONDITION
X	X	1	1	0	DATA BUS → 3-STATE

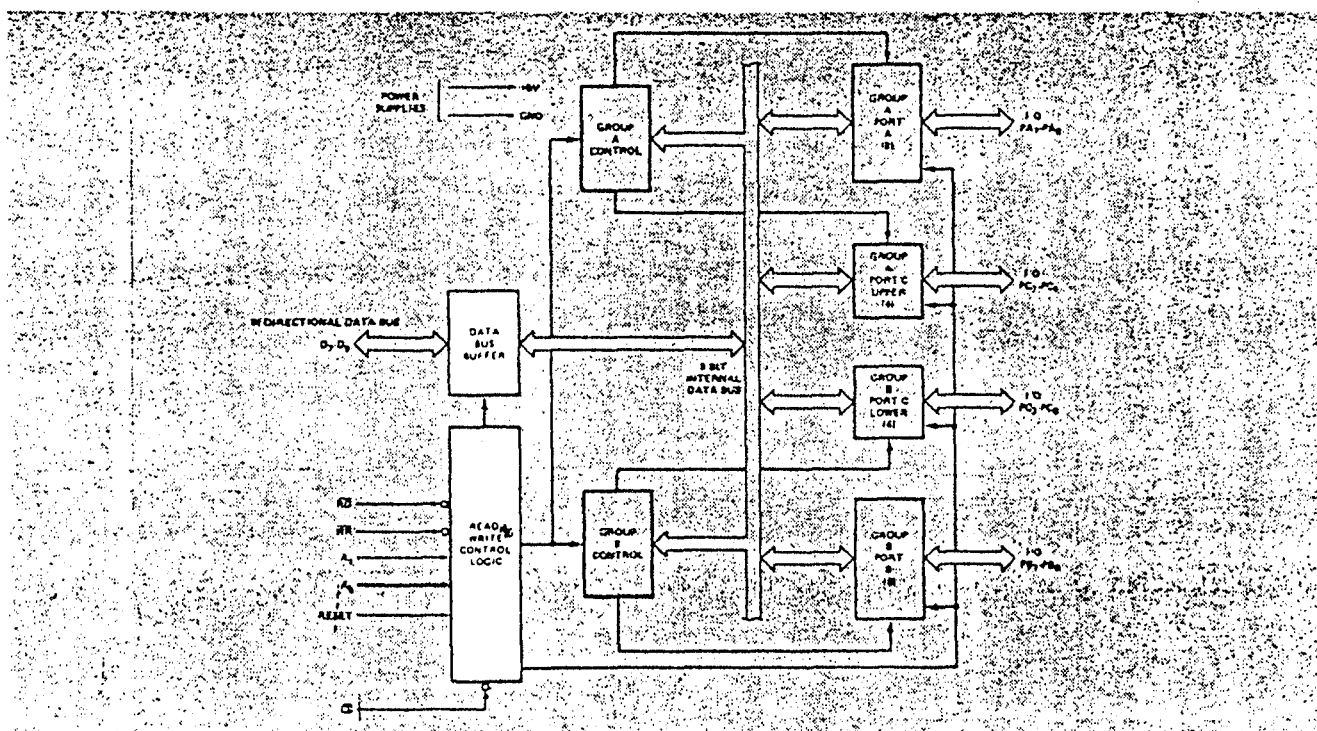


Figure 1. 8255A Block Diagram Showing Data Bus Buffer and Read/Write Control Logic Functions

**(RESET)**

**Reset.** A "high on this input clears the control register and all ports (A, C, C) are set to the input mode.

**Group A and Group B Controls**

The functional configuration of each port is programmed by the systems software. In essence, the CPU "outputs" a control word to the 8255A. The control word contains information such as "mode", "bit set", "bit reset", etc., that initializes the functional configuration of the 8255.

Each of the Control blocks (Group A and Group B) accepts "commands" from the Read/Write Control Logic, receives "control words" from the internal data bus and issues the proper commands to its associated ports.

Control Group A – Port A and Port C upper (C7-C4)

Control Group B – Port B and Port C lower (C3-C0)

The Control Word Register can Only be written into. No Read operation of the Control Word Register is allowed.

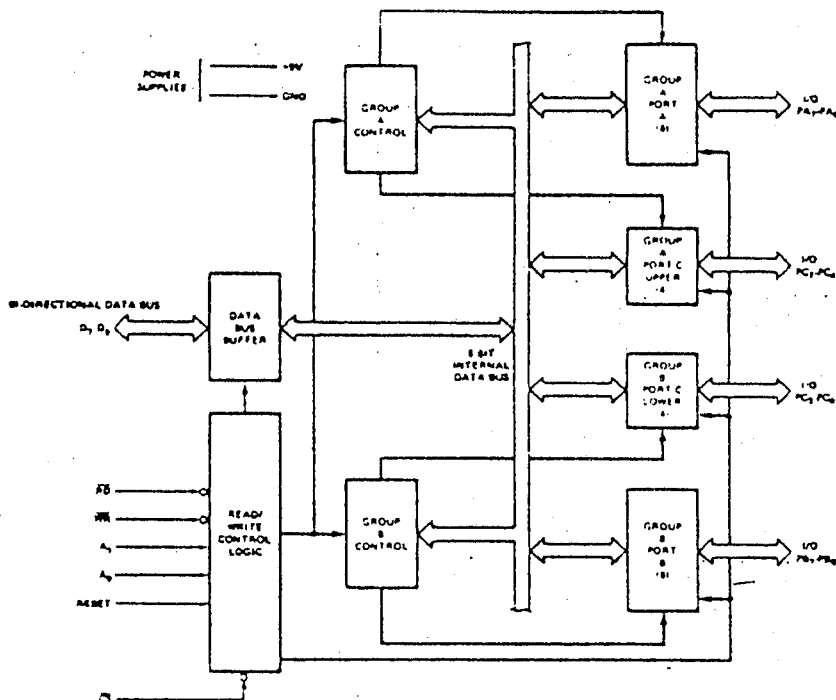
**Ports A, B, and C**

The 8255A contains three 8-bit ports (A, B, and C). All can be configured in a wide variety of functional characteristics by the system software but each has its own special features or "personality" to further enhance the power and flexibility of the 8255A.

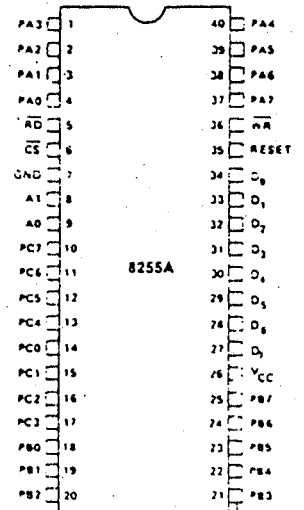
**Port A.** One 8-bit data output latch/buffer and one 8-bit data input latch.

**Port B.** One 8-bit data input/output latch/buffer and one 8-bit data input buffer.

**Port C.** One 8-bit data output latch/buffer and one 8-bit data input buffer (no latch for input). This port can be divided into two 4-bit ports under the mode control. Each 4-bit port contains a 4-bit latch and it can be used for the control signal outputs and status signal inputs in conjunction with ports A and B.



**PIN CONFIGURATION**



**PIN NAMES**

D <sub>7</sub> -D <sub>0</sub>	DATA BUS (BI DIRECTIONAL)
RESET	RESET INPUT
CS	CHIP SELECT
RD	READ INPUT
WR	WRITE INPUT
A0, A1	PORT ADDRESS
PA7-PA0	PORT A (8BIT)
PB7-PB0	PORT B (8BIT)
PC7-PC0	PORT C (8BIT)
Vcc	+5 VOLTS
GND	0 VOLTS

Figure 2. 8255A Block Diagram Showing Group A and Group B Control Functions

## 8255A OPERATIONAL DESCRIPTION

### Mode Selection

There are three basic modes of operation that can be selected by the system software:

- Mode 0 – Basic Input/Output
- Mode 1 – Strobed Input/Output
- Mode 2 – Bi-Directional Bus

When the reset input goes "high" all ports will be set to the input mode (i.e., all 24 lines will be in the high impedance state). After the reset is removed the 8255A can remain in the input mode with no additional initialization required. During the execution of the system program any of the other modes may be selected using a single output instruction. This allows a single 8255A to service a variety of peripheral devices with a simple software maintenance routine.

The modes for Port A and Port B can be separately defined, while Port C is divided into two portions as required by the Port A and Port B definitions. All of the output registers, including the status flip-flops, will be reset whenever the mode is changed. Modes may be combined so that their functional definition can be "tailored" to almost any I/O structure. For instance; Group B can be programmed in Mode 0 to monitor simple switch closings or display computational results, Group A could be programmed in Mode 1 to monitor a keyboard or tape reader on an interrupt driven basis.

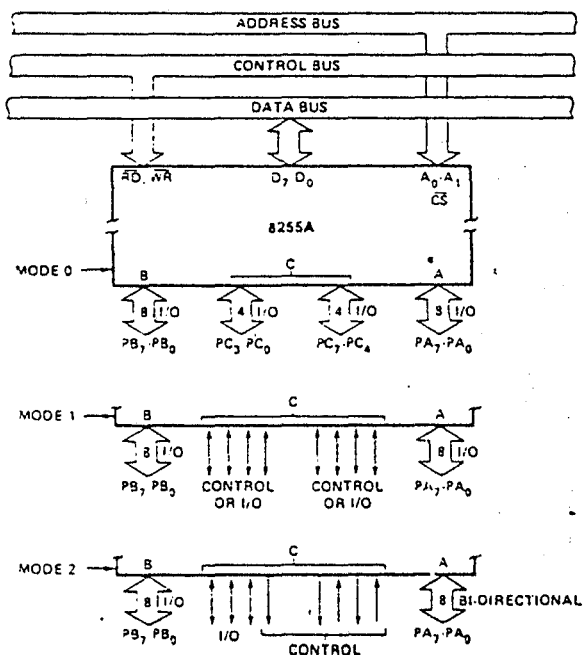


Figure 3. Basic Mode Definitions and Bus Interface

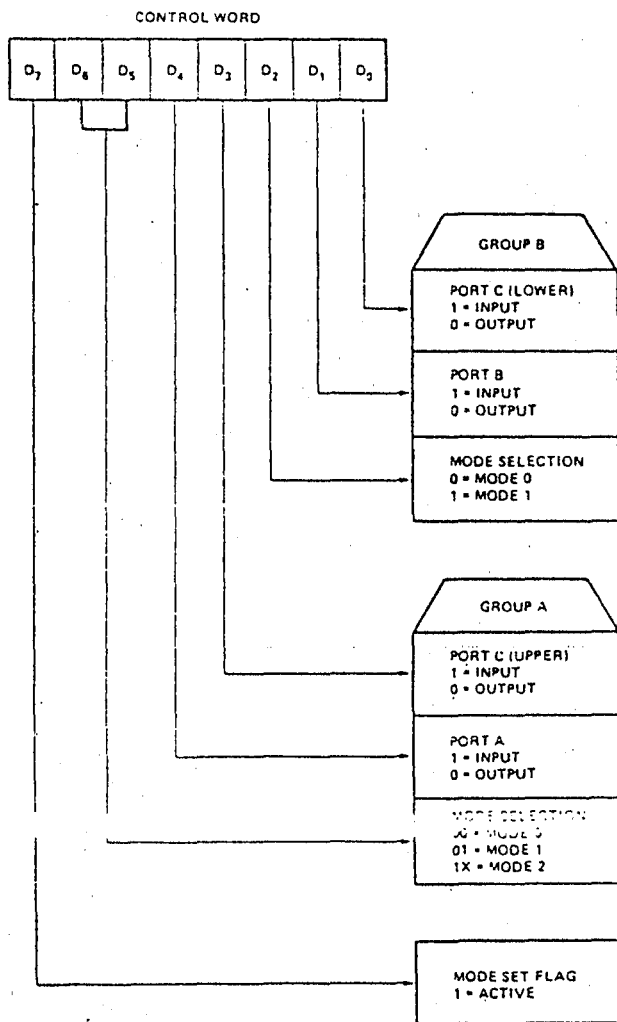


Figure 4. Mode Definition Format

The mode definitions and possible mode combinations may seem confusing at first but after a cursory review of the complete device operation a simple, logical I/O approach will surface. The design of the 8255A has taken into account things such as efficient PC board layout, control signal definition vs PC layout and complete functional flexibility to support almost any peripheral device with no external logic. Such design represents the maximum use of the available pins.

### Single Bit Set/Reset Feature

Any of the eight bits of Port C can be Set or Reset using a single OUTPUT instruction. This feature reduces software requirements in Control-based applications.

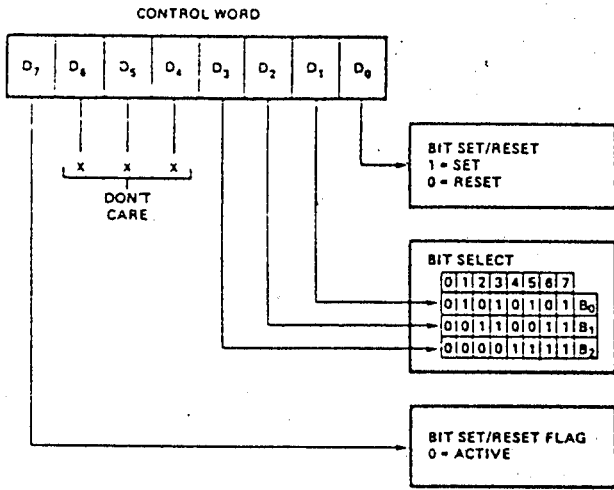


Figure 5. Bit Set/Reset Format

When Port C is being used as status/control for Port A or B, these bits can be set or reset by using the Bit Set/Reset operation just as if they were data output ports.

**Interrupt Control Functions**

When the 8255A is programmed to operate in mode 1 or mode 2, control signals are provided that can be used as interrupt request inputs to the CPU. The interrupt request signals, generated from port C, can be inhibited or enabled by setting or resetting the associated INTE flip-flop, using the bit set/reset function of port C.

This function allows the Programmer to disallow or allow a specific I/O device to interrupt the CPU without affecting any other device in the interrupt structure.

INTE flip-flop definition:

- (BIT-SET) – INTE is SET – Interrupt enable
- (BIT-RESET) – INTE is RESET – Interrupt disable

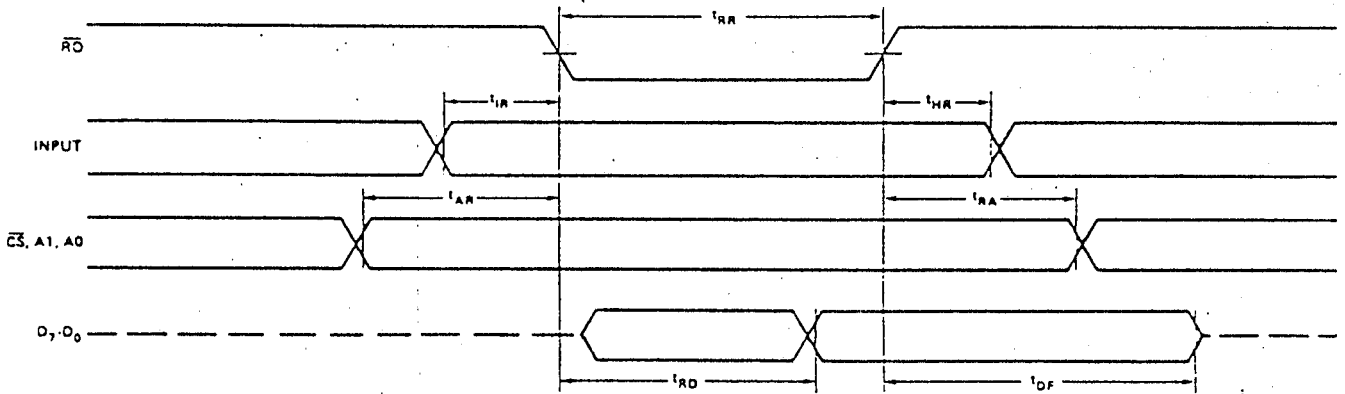
Note: All Mask flip-flops are automatically reset during mode selection and device Reset.

**Operating Modes**

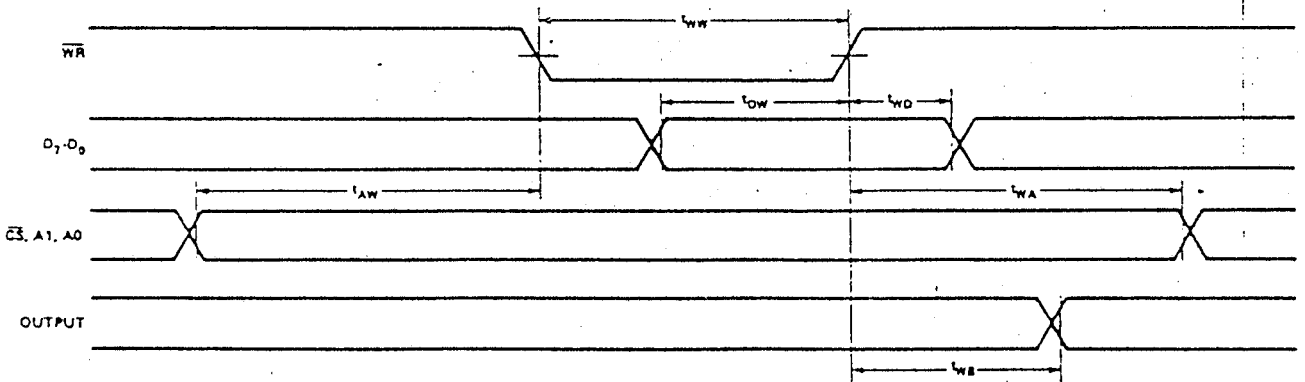
**MODE 0 (Basic Input/Output).** This functional configuration provides simple input and output operations for each of the three ports. No "handshaking" is required, data is simply written to or read from a specified port.

Mode 0 Basic Functional Definitions:

- Two 8-bit ports and two 4-bit ports.
- Any port can be input or output.
- Outputs are latched.
- Inputs are not latched.
- 16 different Input/Output configurations are possible in this Mode.



**MODE 0 (Basic Input)**



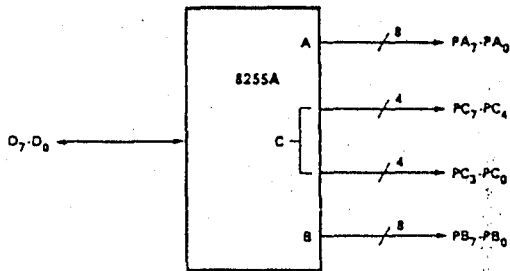
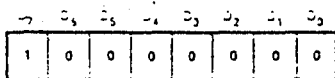
**MODE 0 (Basic Output)**

MODE 0 Port Definition

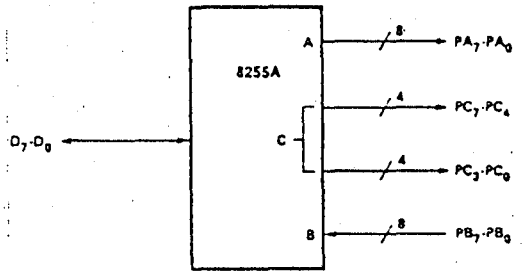
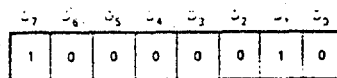
A		B		GROUP A			GROUP B		
D <sub>4</sub>	D <sub>3</sub>	D <sub>1</sub>	D <sub>0</sub>	PORT A	PORT C (UPPER)	#	PORT B	PORT C (LOWER)	
0	0	0	0	OUTPUT	OUTPUT	0	OUTPUT	OUTPUT	
0	0	0	1	OUTPUT	OUTPUT	1	OUTPUT	INPUT	
0	0	1	0	OUTPUT	OUTPUT	2	INPUT	OUTPUT	
0	0	1	1	OUTPUT	OUTPUT	3	INPUT	INPUT	
0	1	0	0	OUTPUT	INPUT	4	OUTPUT	OUTPUT	
0	1	0	1	OUTPUT	INPUT	5	OUTPUT	INPUT	
0	1	1	0	OUTPUT	INPUT	6	INPUT	OUTPUT	
0	1	1	1	OUTPUT	INPUT	7	INPUT	INPUT	
1	0	0	0	INPUT	OUTPUT	8	OUTPUT	OUTPUT	
1	0	0	1	INPUT	OUTPUT	9	OUTPUT	INPUT	
1	0	1	0	INPUT	OUTPUT	10	INPUT	OUTPUT	
1	0	1	1	INPUT	OUTPUT	11	INPUT	INPUT	
1	1	0	0	INPUT	INPUT	12	OUTPUT	OUTPUT	
1	1	0	1	INPUT	INPUT	13	OUTPUT	INPUT	
1	1	1	0	INPUT	INPUT	14	INPUT	OUTPUT	
1	1	1	1	INPUT	INPUT	15	INPUT	INPUT	

MODE 0 Configurations

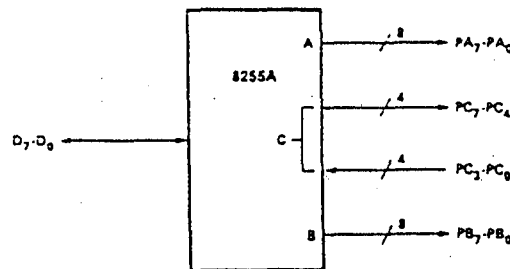
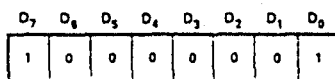
CONTROL WORD #0



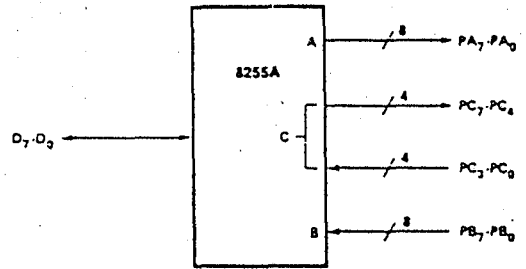
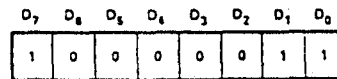
CONTROL WORD #2



CONTROL WORD #1



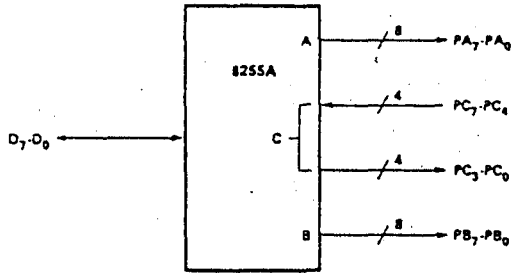
CONTROL WORD #3





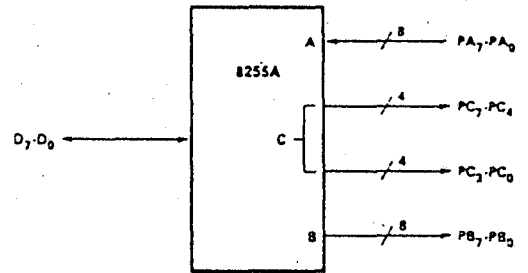
CONTROL WORD #4

D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
1	0	0	0	1	0	0	0



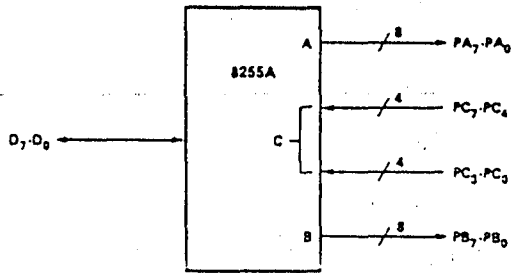
CONTROL WORD #8

D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
1	0	0	1	0	0	0	0



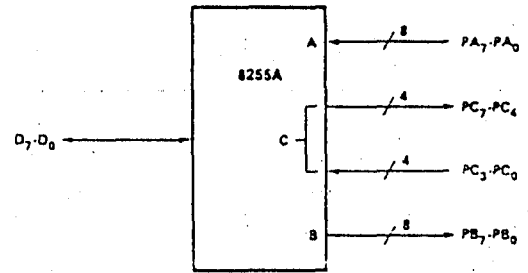
CONTROL WORD #5

D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
1	0	0	0	1	0	0	1



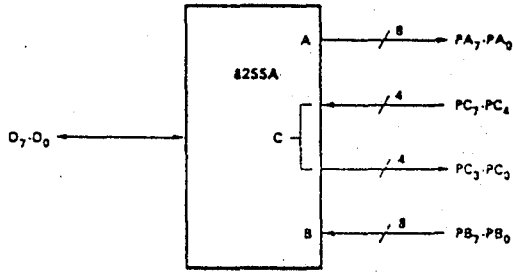
CONTROL WORD #9

D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
1	0	0	1	0	0	0	1



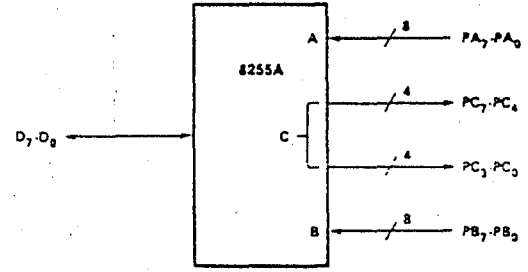
CONTROL WORD #6

D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
1	0	0	0	1	0	1	0



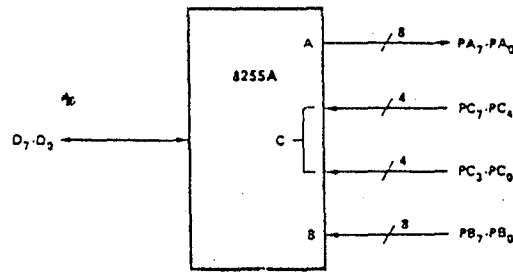
CONTROL WORD #10

D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
1	0	0	1	0	0	1	0



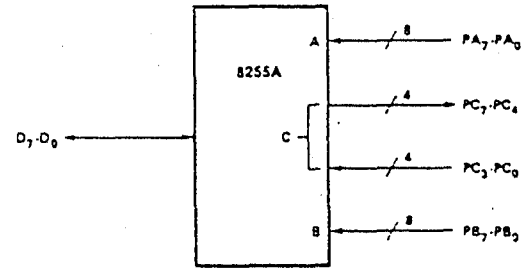
CONTROL WORD #7

D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
1	0	0	0	1	0	1	1

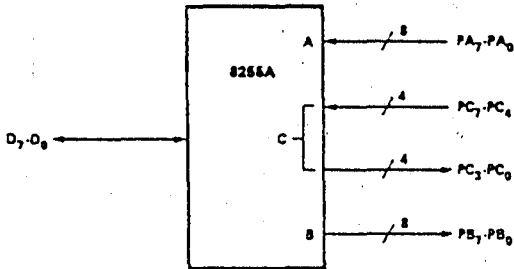
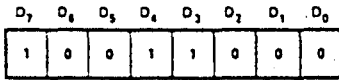


CONTROL WORD #11

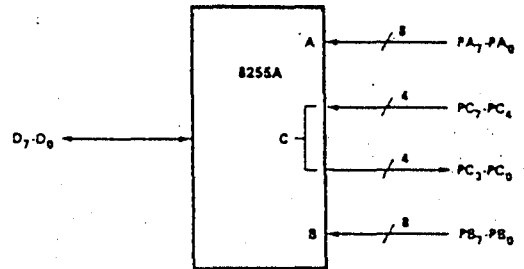
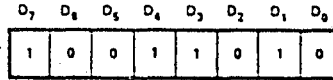
D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
1	0	0	1	0	0	1	1



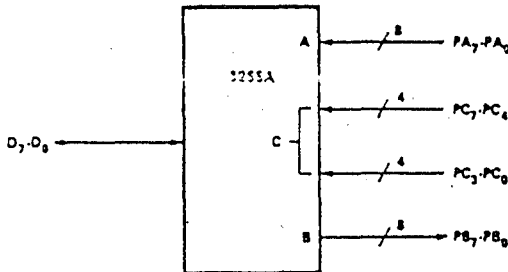
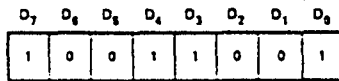
CONTROL WORD #12



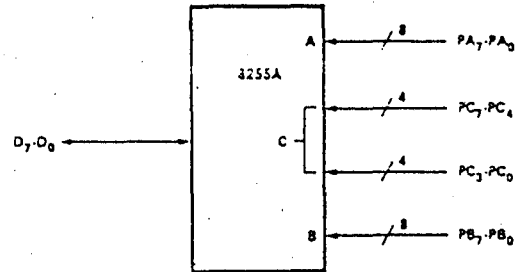
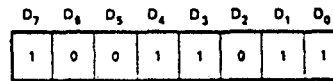
CONTROL WORD #14



CONTROL WORD #13



CONTROL WORD #15



### Operating Modes

**MODE 1 (Strobed Input/Output).** This functional configuration provides a means for transferring I/O data to or from a specified port in conjunction with strobes or "handshaking" signals. In mode 1, port A and Port B use the lines on port C to generate or accept these "handshaking" signals.

#### Mode 1 Basic Functional Definitions:

- Two Groups (Group A and Group B)
- Each group contains one 8-bit data port and one 4-bit control/data port.
- The 8-bit data port can be either input or output. Both inputs and outputs are latched.
- The 4-bit port is used for control and status of the 8-bit data port.

```

10 A$=CHR$(23)+CHR$(1):REM XOR GRAPHICS MODE
20 B$=CHR$(23)+CHR$(3):REM OR GRAPHICS MODE
30 MENU0$="Record Printscreen Waveform save load pattern"
40 MENU1$="circuit copyall Clear-screen directory Other"
50 MODE 2:MEMORY &57FF
60 WINDOW 10,1,80,1,25:PAPER 10,0:PEN 10,1
70 WINDOW 11,1,80,25,25:PAPER 11,1:PEN 11,0
80 WINDOW 12,1,80,3,24:PAPER 10,0:PEN 10,1
90 GOSUB 360 INITIALIZE 8255
100 CLSF1
110 INPUT 11,"states y/n";st$:st%=UPPER$(st%)
120 PRINT 11,menu0$
130 H$=INKEY$:IF H$="" THEN 130
140 IF UPPER$(H$)="V" THEN GOSUB 270:GOTO 120
150 IF UPPER$(H$)="R" THEN GOSUB 680:GOTO 120
160 IF UPPER$(H$)="P" THEN CALL &A500:GOTO 120
170 IF UPPER$(H$)="W" THEN GOSUB 420:GOTO 120
180 IF UPPER$(H$)="C" THEN CLS 10:CLS 12:GOTO 120
190 IF UPPER$(H$)="G" THEN GOSUB 470:GOTO 120
200 IF UPPER$(H$)="O" THEN GOSUB 250:GOTO 130
210 IF UPPER$(H$)="E" THEN GOSUB 300:GOTO 120
220 IF UPPER$(H$)="D" THEN GOSUB 320:GOTO 120
230 IF UPPER$(H$)="N" THEN GOSUB 740:GOTO 120
240 GOTO 130
250 CLS 11:IF MENU=0 THEN MENU=1:PRINT 11,MENU1$; ELSE MENU=0:PRINT 11,MENU0$
260 RETURN
270 CLS 11:INPUT 11,"FILE NAME ",FILEN$
280 SAVE FILEN$,B,&6000,&3F00
290 RETURN
300 WINDOW 10,1,80,5,24:PAPER 10,0:PEN 10,1:CLS:CAT
310 WINDOW 10,1,80,1,4:PAPER 10,0:PEN 10,1:WINDOW 12,1,80,5,24:PAPER 12,0:PEN
,1:RETURN
320 CLS 11:INPUT 11,"FILE NAME ",FILEN$
330 LOAD FILEN$
340 RETURN
350 END
360 '===== INITIALIZE =====
370 OUT &FBF3,&91:OUT &FBF2,&80:OUT &FBF1,&80
380 IF PEEK(&A0FF)=175 THEN RETURN ELSE POKE &A0FF,175
390 LOAD "!asmALL"
400 LOAD "!rcgobj"
410 RETURN
420 '===== WAVE-FORM OF SPEECH =====
430 CLS 10:CLS 12
440 PRINT b$;
450 CALL &A180
460 PRINT b$;:CLS 11:PRINT 11,"eXit Wave Left Right "
470 PRINT A$;
480 MOVE PAGEX,380:DRAW PAGEX+25,380:MOVE PAGEX,382:DRAW PAGEX+25,382
490 H$=INKEY$:IF H$="" THEN 490
500 IF UPPER$(H$)="P" THEN CALL &A500

```

```

510 MOVE PAGEX,300:DRAW PAGEX:25,300:MOVE PAGEX,302:DRAW PAGEX:25,302
520 IF UPPER$(H$)="X" THEN PRINT B$::RETURN
530 IF H#=CHR$(13) OR H#=CHR$(10) THEN GOSUB 600:GOTO 400
540 IF UPPER$(H$)="W" THEN GOSUB 600:GOTO 400
550 IF UPPER$(H$)="L" THEN PAGEX=PAGEX+5:GOTO 400
560 IF UPPER$(H$)="R" THEN PAGEX=PAGEX+5:GOTO 400
570 IF H#=CHR$(8F3) THEN PAGEX=PAGEX+5:GOTO 400
580 IF H#=CHR$(8F2) THEN PAGEX=PAGEX+5:GOTO 400
590 GOTO 400
600 CLS L2:PRINT B$::MOVE 0,127
610 i%=i:page%=pagex
620 FOR I%=PAGE%X25 TO PAGE%X25+637
630 MOVE I%-25*PAGE%,300:DRAW I%-25*PAGE%,300+10*(PEEK(I%+6000)AND 0)/2
640 MOVE I%-25*PAGE%,280:DRAW I%-25*PAGE%,280+10*(PEEK(I%+6000)AND 4)/4
650 MOVE I%-25*PAGE%,260:DRAW I%-25*PAGE%,260+10*(PEEK(I%+6000)AND 2)/2
660 NEXT:PRINT A$::RETURN
670 '
680 '*** LOADING ***
690 CLS L1:'GOSUB 680
710 INPUT L1,"PRESS RETURN KEY AND SPEAK IMMEDIATELY FOR SPEECH RECORD ",H$
720 CLS L1:PRINT L1,"LOADING PLEASE WAIT":CALL &A140:CLS L1:'RETURN
730 '
740 '* RECOGNITION *
750 PRINT B$:CLS:CALL &A180
760 PRINT L1,"exit "
770 BYTEADR=PAGEX*25+6000
780 POKE &5B1F,INT(BYTEADR/256)
790 POKE &5B1E,BYTEADR-256*INT(BYTEADR/256)
800 scrx=PAGEX
810 status=&5B00:vcflag=&5B01:vcper=&5B02:vcctr=&5B03:onasm=&5B00
  820 bpflag=&5B04:bpper=&5B05:bpctr=&5B06:bpff=&5B07:bpctot=&5B08:bpbuf=&5B0A
830 hpflag=&5B0E:hpper=&5B0F:hpctr=&5B10:hpff=&5B11:hptot=&5B12:hpbuf=&5B14
840 POKE bpff,4
850 scrline=8:LOCATE 40,scrline:PRINT "state":scrline=scrline+1
860 '
870 IF st$="Y" THEN LOCATE 40,scrline:PRINT "1":scrline=scrline+1
875 POKE bpbuf+1,&FF
880 nvc=0:nha=0:v=0:h=0:b=0
890 CALL onasm:PLOT scrx,380:scrx=scrx+4:IF scrx>630 THEN RETURN
900 IF PEEK(vcflag)=0 AND PEEK(bpbuf)<&6F AND PEEK(bpbuf+1)<&6F AND PEEK(bpbuf
)&50 AND PEEK(bpbuf+1)&50 AND (PEEK(hpctr)&FE OR P
EEK(hpff)>1) AND PEEK(bpff)<1 THEN 970
910 IF PEEK(hpff)>0 OR PEEK(hpctr)&F0 THEN nvc=0:nha=0
915 IF PEEK(vcflag)=1 THEN v=v+1:nvc=0:b=b+PEEK(hptot):h=h+PEEK(hptot)
924 IF v>7 OR(v>3 AND PEEK(vcflag)=0) THEN LOCATE 1,scrline:PRINT HEX$(L/v,2);"
";HEX$(h/v,2);" ";HEX$(v,2)
925 IF v>7 OR(v>3 AND PEEK(vcflag)=0) THEN IF b/v>&30 AND L/v<&70 AND h/v>&50 AN
D h/v<&6F THEN en$="a":GOTO 7000
926 IF v>7 OR(v>3 AND PEEK(vcflag)=0) THEN IF b/v>&70 AND L/v<&70 AND h/v>&40 AN
D h/v<&52 THEN en$="u":GOTO 7000
927 IF v>7 OR(v>3 AND PEEK(vcflag)=0) THEN IF b/v>&50 AND L/v<&A0 AND h/v>&20 AN
D h/v<&48 THEN en$="e":GOTO 6000

```

```

928 IF PEEK(VCFIAG)=0 THEN nVC=nVC+1:nHA=PEEK(HPTOT):nHA:L=0:h=0:v=0
929 IF nVC>4 THEN LOCATE 1,scrline:PRINT HEX$(nHA/nVC,2); " ";HEX$(nvc,2)
940 IF nVC>4 THEN IF nHA/nVC<&24 THEN en$="s":GOTO 2030
945 IF nVC>4 THEN IF nHA/nVC<&30 THEN en$="k":GOTO 2030
950 IF nVC>4 THEN LOCATE 1,scrline:PRINT HEX$(nHA/nvc,2); " ";HEX$(nvc,2) :GOTO 0
80
960 GOTO 890
970 '
980 IF st$="Y" THEN LOCATE 40,scrline:PRINT "2":scrline=scrline+1
990 V=0:L=0:h=0
1000 CALL onasm:PLOT scrx,380:scrx=scrx+4
1020 IF PEEK(BPCTR)=&FF AND PEEK(HPCTR)=&FF THEN 860
1030 IF PEEK(VCFIAG)=1 AND V>0 THEN V=v+1:L=L+PEEK(BPTOT):h=h+PEEK(HPTOT)
1040 IF PEEK(VCFIAG)=1 AND V=0 AND PEEK(BPTOT)<&B0 AND PEEK(HPCFF)=0 THEN V=1:B=P
EEK(BPTOT):H=PEEK(HPTOT)
1050 IF V>7 OR (PEEK(VCFIAG)=0 AND V>1) THEN 1060
1060 IF PEEK(VCFIAG)=0 THEN V=0:L=0:h=0
1070 IF SCRXX<&30 THEN 1000:ELSE RETURN
1080 LOCATE 1,scrline:PRINT HEX$(D/V,2); " ";HEX$(H/V,2); " ";HEX$(V,2)
1090 IF B/V<&A0 AND B/V>&70 AND H/V<&47 AND H/V>&2F THEN en$="bi":GOTO 7000
1100 IF B/V<&6D AND B/V>&60 AND H/V<&4D AND H/V>&38 THEN en$="be":GOTO 7000
1110 IF B/V<&72 AND B/V>&57 AND H/V<&5A AND H/V>&47 THEN GOTO 1470
1120 IF B/V<&7A AND B/V>&68 AND H/V<&75 AND H/V>&4A THEN GOTO 1640
1130 GOTO 1000
1460 '
1470 IF st$="Y" THEN LOCATE 40,scrline:PRINT "7   do":scrline=scrline+1
1480 nVOICECTR=0:PLOT scrx,380:scrx=scrx+4:IF SCRXX>&30 THEN RETURN
1490 CALL onasm:PLOT scrx,380:scrx=scrx+4:IF SCRXX>&30 THEN RETURN
1500 IF (PEEK(BPCTR)>&F0 OR PEEK(BP(f))>0) AND (PEEK(HPCTR)>&F0 OR PEEK(hp(f))>0)
THEN 1520
1510 IF PEEK(vcfIag)=1 THEN nVOICECTR=0:GOTO 1490:ELSE nVOICECTR=nVOICECTR+1:IF
nVOICECTR=9 THEN en$="be":GOTO 7000 ELSE GOTO 1490
1520 '
1530 IF st$="Y" THEN LOCATE 40,scrline:PRINT "8   dor ":scrline=scrline+1
1540 ffctr=0
1550 CALL onasm:PLOT scrx,380:scrx=scrx+4:IF SCRXX>&30 THEN RETURN
1560 IF PEEK(BPCTR)=&FF AND PEEK(HPCTR)=&FF THEN ffctr=ffctr+1:nffctr=0:IF ffctr
>20 THEN RETURN ELSE 1550
1580 CALL onasm:PLOT scrx,380:scrx=scrx+4:IF SCRXX>&30 THEN RETURN
1590 IF PEEK(vcfIag)=1 THEN 1760 'may be dokuz
1600 IF PEEK(hptot)<&38 AND PEEK(hptot)>&1C THEN 1620 ELSE 1580
1610 '
1620 LOCATE 40,scrline:PRINT "9   dorl":scrline=scrline+1
1630 GOTO 860
1640 '
1650 IF st$="Y" THEN LOCATE 40,scrline:PRINT "10  do(k)":scrline=scrline+1
1660 WHILE PEEK(vcfIag)=1 :CALL onasm:PLOT scrx,380:scrx=scrx+4:IF SCRXX>&30 THEN
RETURN
1670 WEND
1680 ffctr=0
1690 CALL onasm:PLOT scrx,380:scrx=scrx+4:IF SCRXX>&30 THEN RETURN

```

```

1700 IF PEEK(BPCTR)=&FF AND PEEK(HPCTR)=&FF THEN ffcctr=ffctr+1:fffctr=0:IF ffcctr
>20 THEN RETURN ELSE 1670
1710 fffctr=fffctr+1:IF fffctr<2 THEN 1670
1720 CALL onasm:PLOT scrx,380:scrx=scrx+4:IF SCR>630 THEN RETURN
1730 IF PEEK(vcf1ag)=1 THEN 1760
1740 IF PEEK(hptot)<&2C AND PEEK(hptot)>&1C THEN 1620 ELSE 1720
1750
1760 IF st$="Y" THEN LOCATE 40,scrline:PRINT "11   Jo(ku)":scrline=scrline+1
1770 VOICECTR=0:BPADD=0:HPADD=0
1780 CALL onasm:PLOT scrx,380:scrx=scrx+4:IF SCR>630 THEN RETURN
1790 IF PEEK(BPCTR)=&FF AND PEEK(HPCTR)=&FF THEN RETURN
1800 IF PEEK(VCF1AG)=1 AND VOICECTR>0 THEN VOICECTR=VOICECTR+1:BPADD=PEEK(BPTOT)
+BPADD:HPADD=PEEK(HPTOT)+HPADD
1810 IF PEEK(VCF1AG)=1 AND VOICECTR=0 AND PEEK(BPTOT)<&B0 AND PEEK(HPTOT)=0 THEN
VOICECTR=1:BPADD=PEEK(BPTOT):HPADD=PEEK(HPTOT)
1820 IF VOICECTR>7 OR (PEEK(VCF1AG)=0 AND VOICECTR>4) THEN 1850
1830 IF PEEK(VCF1AG)=0 THEN VOICECTR=0:BPADD=0:HPADD=0
1840 GOTO 1780
1850 LOCATE 1,scrline:PRINT HEX$(BPADD/VOICECTR,2); " ";HEX$(HPADD/VOICECTR,2); "
";HEX$(VOICECTR,2)
1860 IF BPADD/VOICECTR<&A0 AND BPADD/VOICECTR>&5D AND HPADD/VOICECTR<&60 AND HPA
DD/VOICECTR>&40 THEN en$="doku":GOTO 2000:ELSE 1780
2030 IF st$="Y" THEN LOCATE 40,scrline:PRINT en$:scrline=scrline+1
2040 VOICECTR=0:BPADD=0:HPADD=0
2050 CALL onasm:PLOT scrx,380:scrx=scrx+4:IF SCR>630 THEN RETURN
2060 IF PEEK(HPCTR)>&F0 THEN 660
2070 IF PEEK(VCF1AG)=1 THEN VOICECTR=VOICECTR+1:BPADD=PEEK(BPTOT)+BPADD:HPADD=PE
EK(HPTOT)+HPADD
2080 IF VOICECTR>7 OR (PEEK(VCF1AG)=0 AND VOICECTR>3) THEN 2110
2090 IF PEEK(VCF1AG)=0 THEN VOICECTR=0:BPADD=0:HPADD=0
2100 GOTO 2050
2110 PRINT HEX$(BPADD/VOICECTR) ; " ";HEX$(HPADD/VOICECTR)
2120 IF BPADD/VOICECTR<&98 AND BPADD/VOICECTR>&5D AND HPADD/VOICECTR<&60 AND HPA
DD/VOICECTR>&41 THEN 2150
2130 IF BPADD/VOICECTR<&98 AND BPADD/VOICECTR>&50 AND en$="s" AND HPADD/VOICECTR
<&49 AND HPADD/VOICECTR>&30 THEN 2330
2135 IF BPADD/VOICECTR<&A0 AND BPADD/VOICECTR>&50 AND en$="k" AND HPADD/VOICECTR
<&48 THEN fst$="iki":GOTO 6170
2140 GOTO 2050
2150 *X state 15 *
2160 IF st$="Y" THEN LOCATE 40,scrline:PRINT "15   sI ":scrline=scrline+1
2170 WHILE PEEK(vcf1ag)=1:CALL onasm:PLOT scrx,380:scrx=scrx+4:IF SCR>630 THEN
RETURN
2180 WEND:c=0
2190 WHILE c<8:c=c+1:CALL onasm:PLOT scrx,380:scrx=scrx+4:IF SCR>630 THEN RETUR
N
2200 WEND
2210 Vc=0:b=0:h=0
2220 CALL onasm:PLOT scrx,380:scrx=scrx+4:IF SCR>630 THEN RETURN
2230 IF PEEK(HPCTR)>&F0 THEN 660
2240 IF PEEK(VCF1AG)=1 THEN Vc=Vc+1:B=PEEK(BPTOT)+B:H=PEEK(HPTOT)+H
2250 IF Vc>7 OR (PEEK(VCF1AG)=0 AND Vc>3) THEN 2260
2260 IF PEEK(VCF1AG)=0 THEN Vc=0:B=0:H=0

```

```

2270 GOTO 2220
2280 PRINT HEX$(B/VC) ; " "; HEX$(h/vc)
2290 IF B/vc<&90 AND B/vc>&5D AND H/vc<&60 AND H/vc>&48 THEN en$="sIfI":GOTO 700
0
2300 GOTO 2220
2330 '* state 18*
2340 IF st$="Y" THEN LOCATE 40,scrline:PRINT "18 se ":scrline=scrline+1
2350 WHILE PEEK(vcflag)=1 :CALL onasm:PLOT scrx,380:scrx=scrx+4:IF scrx>630 THEN
RETURN
2351 WEND:fc=0
2352 CALL onasm:PLOT scrx,380:scrx=scrx+4:IF scrx>630 THEN RETURN
2353 IF PEEK(BPCTR)=&FF AND PEEK(HPCTR)=&FF THEN fc=fc+1:hfc=0:IF fc>20 THEN G60
ELSE 2352
2354 'IF fc<2 THEN 2352
2355 'CALL onasm:PLOT scrx,380:scrx=scrx+4:IF scrx>630 THEN RETURN
2356 'IF not(PEEK(hptot)<&2C AND PEEK(hptot)>&1C and PEEK(vcflag)=0) then 2355
2357 'IF PEEK(BPCTR)=&FF AND PEEK(HPCTR)=&FF THEN G60
2360 'IF PEEK(vcflag)=1 THEN 2370:ELSE 2355
2370 v=0:b=0:h=0:GOTO 2390
2380 CALL onasm:PLOT scrx,380:scrx=scrx+4:IF scrx>630 THEN RETURN
2390 IF PEEK(BPCTR)=&FF AND PEEK(HPCTR)=&FF THEN fc=fc+1:IF fc>20 THEN G60
2400 IF PEEK(VcFLAG)=1 AND PEEK(BPTOT)<&B0 AND PEEK(HPTOT)<&80 AND PEEK(HPTOT)
THEN v=v+1:B=b+PEEK(BPTOT):h=h+PEEK(HPTOT)
2405 'IF v>0 THEN PRINT HEX$(B/v,2) ; " "; HEX$(H/v,2) ; " "; v
2410 IF v>7 OR (PEEK(VcFLAG)=0 AND v>3) THEN PRINT HEX$(B/v,2) ; " "; HEX$(H/v,2) :I
F B/v<&A0 AND B/v>&70 AND H/v<&44 AND H/v>&28 THEN e
n$="seki":GOTO 7000
2420 GOTO 2380
7000 IF st$="Y" THEN LOCATE 40,scrline:PRINT en$:scrline=scrline+1:'be-bi-doku-s
IfI-seki-a-u
7001 WHILE PEEK(VcFLAG)=1 :CALL onasm:PLOT scrx,380:scrx=scrx+4:IF scrx>630 THEN
RETURN
7002 WEND
7010 nV=0:h=0:f1=0:va=0
7020 CALL onasm:PLOT scrx,380:scrx=scrx+4:IF scrx>630 THEN RETURN
7030 IF (PEEK(BPCTR)>&F0 OR PEEK(BPff)>0) AND (PEEK(HPCTR)>&F0 OR PEEK(Hpff)>0)
AND en$="be" THEN 1520
7035 IF PEEK(BPCTR)=&FF AND PEEK(HPCTR)=&FF AND (en$="be" OR en$="Li" OR en$="do
ku" OR en$="sIfI" OR en$="seki") THEN G60
7036 IF (PEEK(BPCTR)>&B0 OR PEEK(HPCTR)>&B0) THEN:nv=0:h=0:f1=f1+1:IF (en$="a"
OR en$="u") AND f1>0 THEN en$=en$+" "
7040 IF PEEK(VcFLAG)=0 AND nv>0 THEN nv=nv+1:h=PEEK(HPTOT) III
7050 IF PEEK(vcFLAG)=0 AND nv=0 AND PEEK(BPTOT)<&B0 AND PEEK(HPTOT)=0 THEN nv=1:II
=PEEK(HPTOT)
7065 IF PEEK(VcFLAG)=1 AND en$="a " THEN 7200
7066 IF PEEK(VcFLAG)=1 AND en$="a" AND f1<1 AND nv>1 THEN en$="kan":GOTO 7200
7067 IF nv>5 THEN 7100
7080 GOTO 7020
7100 PRINT HEX$(H/nV,2) ; " "; HEX$(nV,2)
7110 IF H/nV<&34 AND H/nV>&1A AND en$="be" THEN fst$="bes":GOTO 7130
7111 IF H/nV<&56 AND H/nV>&1A AND en$="bi" THEN fst$="bir":GOTO 7130
7112 IF H/nV<&35 AND H/nV>&1A AND en$="doku" THEN fst$="dokuz":GOTO 7130

```

```

7113 IF h/nv<&40 AND en#="sIfI" THEN fst#="sIfIr":GOTO 7130
7114 IF h/nv<&38 AND h/nv>&10 AND en#="seki" THEN fst#="sekiz":GOTO 7130
7115 IF h/nv<&28 AND (en#="a" OR en#="a ") THEN fst#="ac":GOTO 7130
7116 IF h/nv<&40 AND f1>2 AND en#="a " THEN fst#="al(l)":GOTO 7200
7117 IF h/nv<&37 AND (en#="u" OR en#="u ") THEN fst#="uc":GOTO 7130
7118 IF h/nv<&70 AND h/nv>42 AND f1=0 AND en#="a" THEN fst#="kanal":GOTO 7130
7120 nv=0:h=0:GOTO 7020
7130 IF cc#="ac" AND fst#="bir" THEN cc#="" :OUT &F0F6,&FF
7131 IF fst#="ac" THEN cc#="ac" :ELSE cc#=""
7132 IF fst#="sIfIr" THEN OUT &F0F6,&0
7133 IF fst#="kapat" THEN OUT &F0F6,&0
7135 LOCATE 40,scrline:PRINT fst#:scrline=scrline+1:GOTO 860
7200 LOCATE 40,scrline:PRINT "al(l) - kap(a) - kan(a) ":scrline=scrline+1
7205 WHILE PEEK(VCF6)=0 :CALL onasm:PLOT scrx,380:scrx=scrx+4:IF scrx>630 THEN
RETURN
7210 WEND
7215 v=0:h=0:b=0
7220 CALL onasm:PLOT scrx,380:scrx=scrx+4:IF scrx>630 THEN RETURN
7230 IF PEEK(VCF6)=1 THEN v=v+1:nvc=0:b=b+PEEK(LPTOT):h=h+PEEK(HPTOT)
7240 IF V>7 OR(V>3 AND PEEK(VCF6)=0) THEN LOCATE 1,scrline:PRINT HEX$(b/v,2);"
";HEX$(h/v,2);" ";HEX$(v,2)
7250 IF V>7 OR(V>3 AND PEEK(VCF6)=0) THEN IF b/v>&38 AND b/v<&78 AND h/v>&58 A
ND h/v<&8F AND en#<>"kan" THEN fst#="kapat" :GOTO 81
70
7255 IF V>7 OR(V>3 AND PEEK(VCF6)=0) THEN IF b/v>&38 AND b/v<&78 AND h/v>&58 A
ND h/v<&8F AND en#="kan" THEN fst#="kanal" :GOTO 817
0
7260 IF V>7 OR(V>3 AND PEEK(VCF6)=0) THEN IF b/v>&70 AND b/v<&AA AND h/v>&48 A
ND h/v<&60 THEN fst#="altI" :GOTO 8170
7270 IF V>7 THEN 7215:ELSE GOTO 7220
8000 IF st#="Y" THEN LOCATE 40,scrline:PRINT "ye":scrline=scrline+1
8010 WHILE PEEK(VCF6)=1 :CALL onasm:PLOT scrx,380:scrx=scrx+4:IF scrx>630 THEN
RETURN
8015 WEND
8016 CALL onasm:PLOT scrx,380:scrx=scrx+4:IF scrx>630 THEN RETURN
8020 IF PEEK(vcf6)=0 AND PEEK(bpbuf)<&6F AND PEEK(lpbuf+1)<&6F AND PEEK(bpbuf
)>&50 AND PEEK(lpbuf+1)>&50 AND PEEK(lpff)<2 THEN 81
00
8030 'bosluk
8040 GOTO 8016
8100 '
8110 v=0:h=0:b=0
8120 CALL onasm:PLOT scrx,380:scrx=scrx+4:IF scrx>630 THEN RETURN
8130 IF PEEK(VCF6)=1 THEN v=v+1:nvc=0:b=b+PEEK(LPTOT):h=h+PEEK(HPTOT)
8140 IF V>7 OR(V>3 AND PEEK(VCF6)=0) THEN LOCATE 1,scrline:PRINT HEX$(b/v,2);"
";HEX$(h/v,2);" ";HEX$(v,2)
8150 IF V>7 OR(V>3 AND PEEK(VCF6)=0) THEN IF b/v>&70 AND b/v<&AA AND h/v>&20 A
ND h/v<&48 THEN fst#="yedi" :GOTO 8170
8160 GOTO 8120
8170 WHILE PEEK(VCF6)=1 :CALL onasm:PLOT scrx,380:scrx=scrx+4:IF scrx>630 THEN
RETURN
8180 WEND:GOTO 7130

```



```

10 'frekans analizi
20 INPUT "a,b":ia,ib
30 CLS f0:CLS f2
40 MOVE 20,250:DRAW 600,250:MOVE 40,240:DRAW 40,400
50 MOVE 20, 50:DRAW 600, 50:MOVE 40, 40:DRAW 40,210
60 MOVE 40,250:FOR i=&6000:ia TO &6000:ib
70 MOVE 25,248:TAG:PRINT "a":MOVE ib ia:40,248:PRINT "L":MOVE 500,248:PRINT "L
ime(ms)":MOVE 50,398:PRINT "f(Lime)":MOVE 50,218:PR
INT "f(w)":MOVE 500,38:PRINT "freq(100 hz)":TAGOFF
80 MOVE 25,248:TAG:PRINT "a":MOVE ib ia:40,248:PRINT "L":MOVE 500,248:PRINT "L
ime(ms)":MOVE 50,398:PRINT "f(time)":MOVE 50,218:PR
INT "f(w)":MOVE 500,38:PRINT "freq(100 hz)":TAGOFF
90 FOR i=50 TO 550 STEP 10:MOVE i,248:DRAW i,250:MOVE i,48:DRAW i,50:NEXT
100 PRINT E#:MOVE 40,50
110 FOR n=1 TO 50 STEP 1 :ofset=&6000:ia:c=n*2*PI/(ib-ia):an=0:bn=0
120 FOR t=0 TO ib-ia
130 IF INKEY#("<") THEN RETURN
140 x=2*PI*n*157/(ib-ia)+40:MOVE x,50:DRAW x,(100/(ib-ia)*(an^2+bn^2))^0.5/40:50
' 100 hz icin 10 birim ise 10/(2*pi/100)=157 ile car
pilacak
150 x=2*PI*n*157/(ib-ia)+40:MOVE x,50:DRAW x,(100/(ib-ia)*(an^2+bn^2))^0.5/40:50
' 100 hz icin 10 birim ise 10/(2*pi/100)=157 ile
carpilacak
160 NEXT
170 RETURN

```

5 ;On isleme assembler listesi

```

10 STATUS: EQU 24B00
20 VCFLAG: EQU 24B01
30 VCPER: EQU 24B02
40 VCCTR: EQU 24B03
50 BPFLAG: EQU 24B04
60 BPPER: EQU 24B05
70 BPCTR: EQU 24B06
80 BFFF: EQU 24B07
90 BPTOT: EQU 24B08
100 BPBUF: EQU 24B0A
110 HPFLAG: EQU 24B0E
120 HPPER: EQU 24B0F
130 HPCTR: EQU 24B10
140 HPFF: EQU 24B11
150 HPTOT: EQU 24B12
160 HPBUF: EQU 24B14
170 BYTEAD: EQU 24B1E
180 ORG 24800
190 LD B,100
200 LP100: LD HL,(BYTEAD)
210 LD A,(HL)
220 LD (STATUS),A
230 INC HL
240 LD (BYTEADR),HL
250 LD A,(BPCTR)
260 CF ZFF
270 JR Z,UPHCTR
280 INC A
290 LD (BPCTR),A
300 UPHCTR: LD A,(HPCTR)
310 CF ZFF
320 JR Z,UPVC
330 INC A
340 LD (HPCTR),A
350 UPVC: LD A,0
360 LD (VCFLAG),A
370 LD A,(STATUS)
380 AND B
390 JF Z,BF
400 LD A,1
410 LD (VCFLAG),A
420 BP: LD A,(STATUS)
430 AND 4
440 JR Z,RTBPF
450 LD A,(BPFLAG)
460 AND 1
470 JF NZ,HP
480 CALL BPIHT
490 JR HP
500 RTBPF: LD A,0
510 LD (BPFLAG),A
520 HP: LD A,(STATUS)
530 AND Z
540 JR Z,RTHPF
550 LD A,(HPFLAG)

```

```

1040 HPINT: LD A,1
1050 LD (HPFLAG),A
1060 LD A,(HPCTR)
1070 LD (HPPER),A
1080 LD A,0
1090 LD (HPCTR),A
1100 LD A,(HPBUF+8)
1110 LD (HPBUF+9),A
1120 LD A,(HPBUF+7)
1130 LD (HPBUF+8),A
1140 LD A,(HPBUF+6)
1150 LD (HPBUF+7),A
1160 LD A,(HPBUF+5)
1170 LD (HPBUF+6),A
1180 LD A,(HPBUF+4)
1190 LD (HPBUF+5),A
1200 LD A,(HPBUF+3)
1210 LD (HPBUF+4),A
1220 LD A,(HPBUF+2)
1230 LD (HPBUF+3),A
1240 LD A,(HPBUF+1)
1250 LD (HPBUF+2),A
1260 LD A,(HPBUF)
1270 LD (HPBUF+1),A
1280 LD A,(HPPER)
1290 LD (HPBUF),A
1300 LD HL,0
1310 LD DE,0
1320 LD A,(HPBUF)
1330 LD E,A
1340 ADD HL,DE
1350 LD A,(HPBUF+1)
1360 LD E,A
1370 ADD HL,DE
1380 LD A,(HPBUF+2)
1390 LD E,A
1400 ADD HL,DE
1410 LD A,(HPBUF+3)
1420 LD E,A
1430 ADD HL,DE
1440 LD A,(HPBUF+4)
1450 LD E,A
1460 ADD HL,DE
1470 LD A,(HPBUF+5)
1480 LD E,A
1490 ADD HL,DE
1500 LD A,(HPBUF+6)
1510 LD E,A
1520 ADD HL,DE
1530 LD A,(HPBUF+7)
1540 LD E,A
1550 ADD HL,DE
1560 LD (HPTOT),HL

```

```
1570          LD   A, (HPPER)
1580          CP   1FF
1590          JR   NZ, HPNOFF
1600          LD   A, 0
1610          LD   (HPFF), A
1620          RET
1630 HPNOFF: LD   A, (HPFF)
1640          CP   0
1650          JR   Z, HPFF0
1660          DEC  A
1670          LD   (HPFF), A
1680 HPFF0:  RET
```