

**İSTANBUL TEKNİK ÜNİVERSİTESİ**  
**ELEKTRİK – ELEKTRONİK FAKÜLTESİ**

**BİR GSM MODEMİN FPGA ÜZERİNDE GERÇEKLENMESİ**

**BİTİRME ÖDEVİ**

**SERCAN TUNÇAY**

**040080328**

**Bölümü: Elektronik ve Haberleşme Mühendisliği Bölümü**

**Programı: Elektronik Mühendisliği**

**Danışmanı: Doç. Dr. Sıddıka Berna ÖRS YALÇIN**

**MAYIS 2012**

## **ÖNSÖZ**

Bana sayısız değer katan okuluma, her konuda anlayışlı olan ve yardımını esirgemeyen Berna Hocam'a, İTÜ Güneş Arabası Ekibi'ne, bugünlere gelmemi sağlayan aileme ve yoğun çalışmalarım esnasında yokluğuma sabır gösteren tüm sevdiklerime içten saygı ve teşekkürlerimi sunarım.

Mayıs 2012

Sercan Tunçay

## İÇİNDEKİLER

	<b><u>Sayfa</u></b>
ÖNSÖZ .....	ii
İÇİNDEKİLER.....	iii
KISALTMALAR.....	v
ŞEKİL LİSTESİ .....	vi
ÖZET.....	viii
SUMMARY.....	ix
1. GİRİŞ.....	1
2. GSM ÜZERİNDEN VERİ TRANSFERİ.....	3
3. DOĞRUSAL ÖNGÖRÜ METODU İLE SESİN KODLANMASI.....	7
3.1 İnsan Konuşmasının Modellenmesi.....	7
3.2 Sesin Örneklenmesi ve Çerçeveleme.....	8
3.2.1 Örnekleme.....	8
3.2.2 Çerçeveleme.....	12
3.3 Ses Çerçevesinden LPC Parametrelerinin Elde Edilmesi.....	15
3.3.1 Öz ilişki Parametrelerinin Elde Edilmesi.....	16
3.3.2 Doğrusal Öngörü Katsayılarının Elde Edilmesi .....	19
3.4 Perde (Periyot) Bilgisinin Eldesi ve Seslilik Sınıflandırması.....	20
3.5 LPC – LSF Parametre Dönüşümü .....	22
3.6 Modemin Verici Olarak Kullanılması .....	26
3.7 Modemin Alıcı Olarak Kullanılması .....	28
4. GSM MODEMİN FPGA ÜZERİNDE GERÇEKLENMESİ.....	30
4.1 Xilinx EDK.....	30
4.1.1 Xilinx Platform Studio.....	31
4.1.2 Software Development Kit .....	32
4.2 MicroBlaze.....	33
4.3 Donanım Tasarımı.....	33
4.3.1 Fiziksel donanımlar.....	34

4.3.2 IP Çekirdekleri .....	36
4.4 Donanım – Yazılım Haberleşmesi.....	37
4.5 MicroBlaze ve Kesmeler .....	39
4.5.1 Donanıma Kesme Özelliği Eklenmesi .....	40
4.5.2 Yazılım Rutinlerinde Kesme İdaresi.....	41
4.6 Yazılım Tasarımı .....	42
4.6.1 Program Ana Rutinleri.....	42
4.6.2 Kesme Rutinleri .....	44
5. GERÇEK ZAMAN TESTLERİ .....	45
6. SONUÇLAR VE TARTIŞMA.....	48
KAYNAKLAR .....	49
ÖZGEÇMİŞ.....	51

## **KISALTMALAR**

<b>GSM</b>	: Global System for Mobile Communications
<b>LPC</b>	: Linear Predictive Coding
<b>FPGA</b>	: Field Programmable Gate Array
<b>DSP</b>	: Digital Signal Processor
<b>PCM</b>	: Pulse Code Modulation
<b>GPRS</b>	: General Packet Radio Service
<b>MS</b>	: Mobile StationS
<b>BSS</b>	: Base Station Subsystem
<b>GSM FR</b>	: GSM Full Rate
<b>CDMA</b>	: Code Division Multiple Access
<b>RAM</b>	: Random - Access Memory
<b>LSF</b>	: Linear Spectral Frequencies
<b>RF</b>	: Radio Frequency
<b>EDK</b>	: Embedded Development Kit
<b>XPS</b>	: Xilinx Platform Studio
<b>IP</b>	: Intellectual Property
<b>SDK</b>	: Software Development Kit
<b>RISC</b>	: Reduced Instruction Set Computing
<b>RTL</b>	: Register Transfer Logic
<b>FIFO</b>	: First In First Out
<b>ADC</b>	: Analog to Digital Converter
<b>DAC</b>	: Digital to Analog Converter
<b>DDR2</b>	: Double Data Rate

## ŞEKİL LİSTESİ

	<u>Sayfa</u>
Şekil 2.1 : GSM hattı genel şeması[2].....	3
Şekil 2.2 : GSM ağında şifreleme kullanımına bir örnek[2]. ....	4
Şekil 2.3 : GSM ses kodlayıcılarında ses aktivitesi analizi[5]. ....	5
Şekil 2.4 : GSM Modem verici(sol) ve alıcı(sağ) şemaları [6]. ....	6
Şekil 3.1 : İnsanda sentezlenen ses çeşitleri [8]. ....	7
Şekil 3.2 : İnsanda sesin modellenmesi.İnsan ses sentez yapısı ile LPC bloklarının karşılaştırılması(üst) [9]. ....	8
Şekil 3.3 : Zamanda sürekli bir sinyalin örneklenmesi [10].....	9
Şekil 3.4 : Zaman bölgesinde örnekleme teoremi[10].....	10
Şekil 3.5 : Frekans bölgesinde örnekleme teoremi.Örnek sinyalin frekans tayfı(a), darbe katarı(b), örnekleme teoremine uygun frekans ile yapılan örnekleme sonucu oluşan tayf(c), örneklemenin yeterince hızlı olmaması durumunda tayfin geri dönüştürülemeyecek biçimde bozulması(d) [10].....	11
Şekil 3.6 : İngilizce ‘Should’ kelimesine ait bir ses kaydının bölümlere ayrılışı [10].....	12
Şekil 3.7 : 30ms’lik çerçevelerin 1/3 oranında örtüştürülmesine bir örnek [9].....	13
Şekil 3.8 : Ses çerçevelerinin işlenmesinin ardından yeniden birleştirilmeleri sırasında meydana gelen süreksizlikler [11]. ....	13
Şekil 3.9 : Çerçeveleme, örtüştürme ve yeniden birleştirme şeması [11]. ....	14
Şekil 3.10 : Diktörge(n) (üst) ve Hamming (alt) ile çerçevelenmiş sinyaller(sol) ve tayfları(sağ) [5]. ....	15
Şekil 3.11 : LPC Modeli [6]. ....	16
Şekil 3.12 : Öz ilişki denklem takımı ve doğrusal öngörü katsayıları [5]. ....	19
Şekil 3.13 : İnsan konuşmasının frekans tayfı [11]. ....	21
Şekil 3.14 : Özilişki değerleri yardımıyla perdenin hesaplanması.Ötelenmiş sinyalin kendisiyle olan öz ilişki değeri(üst) ile sinyalin zamandaki ifadesi(alt) [5]. .	21
Şekil 3.15 : LSF değerleri(dikey çizgiler) ve LPC tayfı arasındaki ilişki [11].....	22
Şekil 3.16 : Çift segment yöntemi ile kök bulma[15]. ....	25
Şekil : 3.17 : Adım sayısının görece büyümesinden ötürü kök atlanması durumu [15]. ....	25
Şekil 3.18 : Gönderilmek üzere modeme gelen bir bit dizisinin ayrıştırılması. ....	26
Şekil 3.19 : Sentez filtresi [16]. ....	27
Şekil 4.1 : Xilinx araçlarının gömülü sistem tasarım aşamalarında kullanımı. ....	30
Şekil 4.2 : SDK programına ait ekran görüntüsü. ....	32
Şekil 4.3 : ATLYS geliştirme kartı.....	33
Şekil 4.4 : Ses giriş- çıkışları, ses kodlayıcısı ve FPGA bağlantıları [18].....	34
Şekil 4.5 : AC-Link protokolü [17]. ....	35
Şekil 4.6 : FPGA ve DDR2 Ram bağlantı şeması [18]. ....	35
Şekil 4.7 : ATLYS kartı üzerindeki tuş ve anahtarların bağlantı şeması [18].....	36
Şekil 4.8 : Mimaride bulunan donanımların adres ve genişlikleri.....	37
Şekil 4.9 : Ledler için oluşturulan öncül tanımlamalar. ....	38
Şekil 4.10 : MicroBlaze ile IP'lere erişim(üst) ve Led IP'sine erişim(alt).....	38

<b>Şekil 4.11</b> : AC-Link protokolündeki bölümlerin bir kısmı(üst) ile IP tarafından çözümlenen sağ ses giriş örneğinin okunmasına bir örnek(alt). .....	38
<b>Şekil 4.12</b> : Kesme kontrol donanımının fonksiyonu [19]. .....	39
<b>Şekil 4.13</b> : Donanımın kesme kontrolörüne bağlanması.....	40
<b>Şekil 4.14</b> : Kesme bağlantıları ve öncelik sıralamasının yapılması. ....	40
<b>Şekil 4.15</b> : Kesme kontrolörüne ait port bağlantıları. ....	40
<b>Şekil 4.16</b> : MicroBlaze yazılımsal kesme yapılandırması. ....	41
<b>Şekil 4.17</b> : Yazılım akış diyagramı. ....	42
<b>Şekil 4.18</b> : Modemin alıcı olarak kullanılması.....	43
<b>Şekil 5.1</b> : Süreç hesaplama için kurulan yapı(a) ve zaman kesmesinde işletilen sayaçlar(b).....	45
<b>Şekil 5.2</b> : Süreç hesaplama yapılarının kullanımı. ....	46
<b>Şekil 5.3</b> : Öz ilişki parametreleri fonksiyonundan bir parça.....	47

## BİR GSM MODEMİN FPGA ÜZERİNDE GERÇEKLENMESİ

### ÖZET

Mobil Haberleşme için Küresel Sistem (Global System for Mobile Communications- GSM) günlük hayatta sivil ve askeri uygulamalarda yaygın olarak kullanılır hale gelmiştir. GSM ses kanalı yalnızca ses aktarımı için özelleştirilmiştir ve bu kanal üzerinden şifreli bir konuşma veya resim dosyası gibi genel amaçlı veri transfer etmek mümkün değildir.

Bu tez içeriğinde Doğrusal Öngörülü Kodlama (Linear Predictive Coding- LPC) tekniği ile insan sesini taklit eden bir sistem tasarlanarak, GSM ağı üzerinden aktarılması arzu edilen verilere ilişkin sentezlenen yapay sesler ve bu seslerin alıcı tarafında geri elde edilmesi için gerekli olan aşamalar anlatılmıştır

Gerçekleme aşamasında GSM şebekesinin genel amaçlı veri iletiminde kullanılabilmesi için insan konuşmasının modellenmesine ilişkin teorik altyapı incelenip bu işlemlerin gömülü bir sistemde gerçekleştirilmesi için gerekli algoritmalar oluşturularak C dilinde ön testleri yapılmıştır. İkinci aşamada bu fonksiyonların gerçek zamanlı testlerinin yapılabilmesi için Spartan6 Sahada Programlanabilir Kapı Dizisi (Field Programmable Gate Array- FPGA) içeren Atlys geliştirme kartında MicroBlaze mikroişlemcisi kurulup yazılım - donanım ortak bir çalışma yürütülmüştür.

Sonuç olarak bir GSM ses hattının şifreli görüşme veya genel amaçlı veri transferi için kullanılabilmesi adına tasarlanacak modem için sahip olması gereken temel özellikler ile, gerçek zamanlı çalışma ihtiyacı sebebiyle sistemin hangi parçalarının donanımsal olarak gerçekleştirilmesi gerektiği belirlenmiştir.



## **FPGA IMPLEMENTATION OF A GSM MODEM**

### **SUMMARY**

Global System for Mobile Communications (GSM) has become widely used in both civil and military applications. GSM is specialized only to recognize and transfer human voice, so it is not possible to send an encrypted speech or a picture over GSM because these data will not be recognized by the system as a human speech.

This study contains the model that mimics human voice with Linear Predictive Coding (LPC) to send&receive desired data over GSM without any constraints, which is called a GSM Modem. All steps from data input to artificial voice output and voice input to data output from modem is explained step by step.

First, basic audio processing and speech synthesis fundamentals are examined. After that, algorithmic functions are written and pre-tested in C. As this study is based on a real-time system, an Atlys board with Spartan6 is used with a MicroBlaze to enable a hardware and software co-design implementation.

After the implementation, real-time tests are performed to obtain operation times of each function and process. Test results are examined to conclude which parts of the system should be implemented as hardware or software.

As a result, a GSM Modem is implemented in FPGA with MicroBlaze and worked as expected. With collected results and feedbacks, a model with some function partitioning in hardware is proposed to realize the implementation in real-time mode.

## 1. GİRİŞ

GSM teknolojisi herhangi fiziki bir kabloya dahi ihtiyaç duymadan dünya çapında, insanların birbirleriyle haberleşebilmelerine olanak sağlamıştır ancak, artan şebeke kullanımıyla beraber hat kalitesinin de yüksek tutulabilmesi için GSM ses kanalı yalnızca insan konuşmasını algılamak ve iletmek üzere özelleştirilmiştir[1]. Bu açıdan bakılacak olursa askeri veya banka bilgilerinin iletiildiği sivil bir görüşmede örnek verilebileceği gibi şahıslar arasındaki haberleşmenin yüksek önem arz ettiği durumlarda ses örnekleri yalnızca şifrelenip GSM hattına verilemez çünkü şifreleme işlemi sonrası oluşacak veri dizisi konuşmaya benzemeyecektir. Şuanda mevcut olan GSM sistemi de yalın haliyle son kullanıcılar arasında güvenli iletişime olanak tanımamaktadır [2].

Gelişen sayısal teknolojiler sayesinde geçmişte hayata geçirilmesine ihtimal verilmeyen birçok uygulama bugün günlük hayatta kullanılır hale gelmiştir. Sayısal teknolojinin daha da gelişmesiyle, üst düzey ve daha karmaşık işlemlerin de gerçekleştirilmesi mümkün olmaktadır. Algoritmik veya matematiksel hesaplamalar içeren bir fonksiyon veya sistem bugün Sayısal İşaret İşleyici (Digital Signal Processor- DSP)'ler, ve/veya FPGA'lar kullanılarak gerçekleştirilmektedir. Yalnızca mikrodenetleyiciler yüksek işlem kapasitesi gerektiren uygulamalarda yetersiz kalmakta, paralel işlem gücüne sahip donanımlara ihtiyaç duyulmaktadır.

Gerçeklenmesi arzu edilen GSM üzerinden genel veri aktarım sistemi, hem yüksek karmaşıklıkta hem de gerçek zamanlı olduğundan ötürü paralel iş gücüne sahip donanımlara ihtiyaç duyulmaktadır. Bu sebeple gerçekleştirme için Spartan6 FPGA'sı içeren Atlys geliştirme kartı tercih edilmiştir.

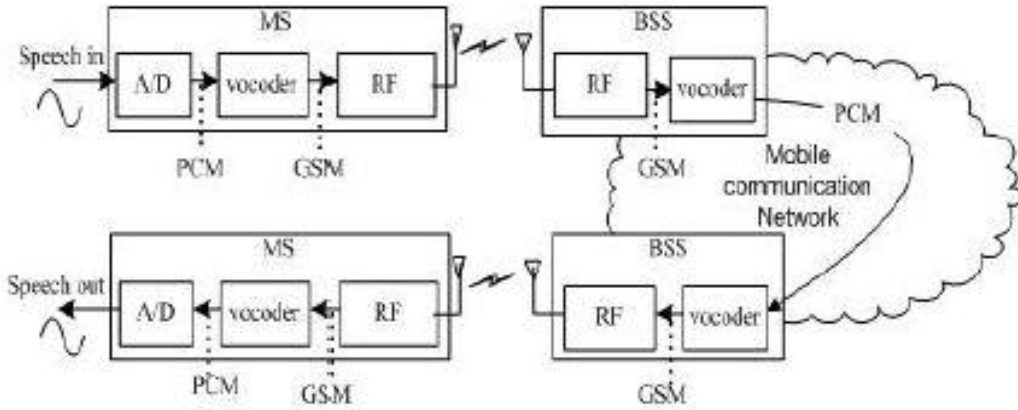
Bu FPGA içerisinde MicroBlaze sanal işlemcisi kullanılarak, yazılım ve donanım ortak bir çalışma yapılması hedeflenmiş, C dilinde fonksiyonların işlevselliği gerçekleştirildikten sonra çeşitli testler yapıp sistemin gerçek zamanlı çalıştırılabilmesi için hızlandırılması gereken fonksiyonlar belirlenmiştir.

Gerçeklenen sistem, karşı tarafa ulaştırılması arzu edilen veriden sentetik bir ses sentezleyip daha sonra bu sesi analiz edip gönderilen orijinal veriyi elde edecektir. Bu noktada önemli olan, gönderilecek olan verinin örneklenmiş bir ses değil arzu edilen herhangi sayısal bir veri olacaktır. Bu sayede, orijinal veri ne olursa olsun arzu edilen haberleşme sağlanabilecektir. Gönderilecek verinin orijinal sesten bağımsız olması, şifreli bir uygulama için şifrelenen ses örneklerinin insan konuşmasına benzememesi veya verinin bir video ya da resime ait olması iletişim kurulmasına engel olmayacaktır.

GSM hattı üzerinden iletişime ait genel bilgilendirme ikinci bölümde, gerçekleştirilmesi yapılacak sisteme ilişkin teorik altyapı üçüncü bölümde, Atlys geliştirme kartı üzerinde MicroBlaze kullanımı ve gerçekleştirme aşamaları ise dördüncü bölümde sunulmuştur. Beşinci bölümde sistemin gerçek zamanlı çalışır hale getirilmesi için yapılan analizler ve elde edilen sonuçlardan bahsedilmiştir.

## 2. GSM ÜZERİNDEN VERİ TRANSFERİ

Günümüzde giderek daha da yaygınlaşan ve gelişen kablosuz teknolojiyi hayatın her alanında kullanmaktayız. Haberleşme ağlarından en yaygın ve kabul görmüş olanlarından biri şüphesiz GSM'dir. GSM hattına eklenen genel veri transfer paketi olan Genel Veri Radyo Servisi (General Packet radio Service- GPRS) ile GSM hattının kendisi aynı radyo kanalını kullanmaktadır. GSM'de öncelik ses iletimi olduğundan ötürü servis kalitesinin korunması amacıyla ses iletişimi dışında kalan hat kapasitesi GPRS'e ayrılmış durumda olup, bu da GPRS kalitesinin zamana bağlılığına yol açmaktadır [1,3,4]. GPRS kalitesinin artırılması için servis sağlayıcılarına da çoğunlukla ek ücret ödenmesi gerekmektedir.



Şekil 2.1 : GSM hattı genel şeması[2].

GSM şebekesi için genel şema Şekil-2.1'de gösterilmiştir. Genel olarak şebeke, Mobil İstasyon (Mobile Station- MS) ve Baz İstasyon Altsistemi (Base Station Subsystem- BSS) olarak ikiye ayrılabilir. MS'e örnek olarak cep telefonları, BSS'ye de baz istasyonları verilebilir. Şemadan görülebildiği gibi ses (veya veri) hem MS'de hem de BSS'de iki kez ard arda kodlanmaktadır. Bu sisteme peş peşe kodlama (tandeming) adı verilmektedir[2].

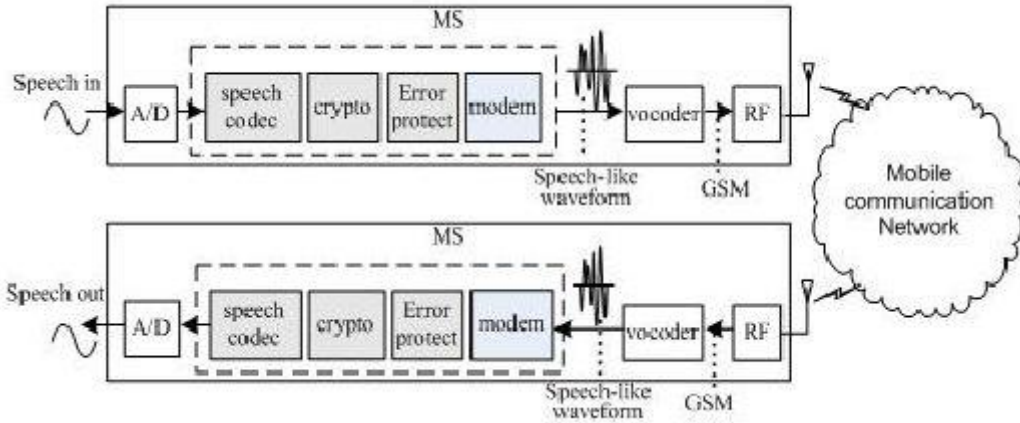
Bu çalışmanın amacı daha önce belirtildiği gibi GSM kullanılarak genel veri transferi sağlanmasıdır. Ancak kriptolanmış bir ses veya bir resme ait veriler insan sesinden elde edilen verilere paralellik göstermeyecektir. GSM ise, bant genişliğinin yüksek verimlilikte değerlendirilebilmesi için yalnızca konuşmaya benzer sesleri ayırt edip gönderen doğrusal olmayan bant sınırlı bir sistemdir [1,4].

Bu bilgiler ışığında GSM hattından veri transfer edebilmek için gerekli parametreler iki ana bölümde toplanabilir;

1-) Ses (sentetik veya gerçek), kanaldan , ses aktivasyon detektörü ve ses bandına sığacak biçimde sistem filtrelerine maruz kalmadan geçebilmelidir. [1,2].

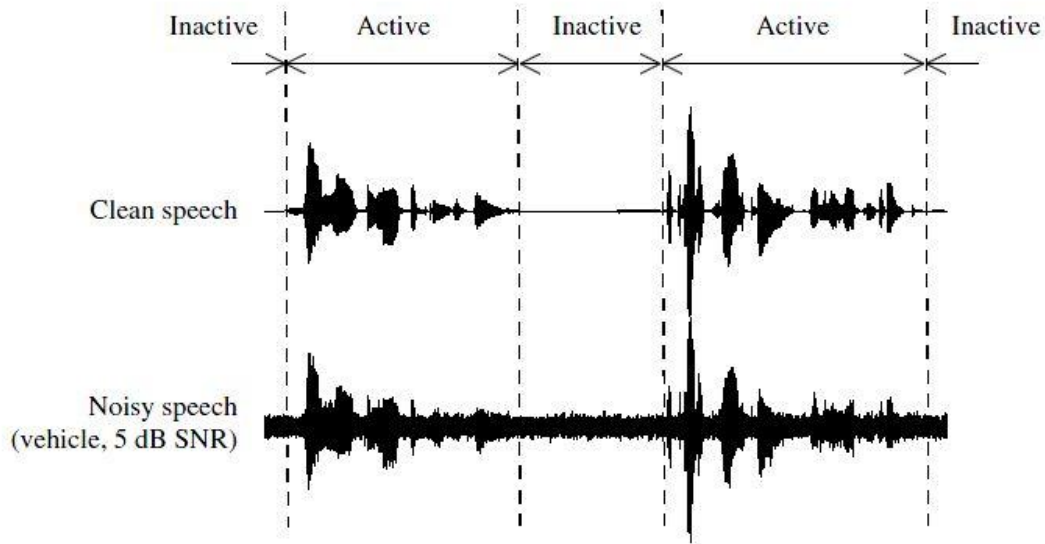
2-) GSM ses kodlayıcısına karşı sağlam olmalıdır.Sağlamlıktan kasıt kodlanan sesin GSM çözücüsünden başarıyla geri çözümlenebilmesidir [1,2].

GSM ağında meydana gelen konuşma dışı seslerin bastırılması problemi iki yöntem ile çözülebilir. İlk yöntem gönderilecek verinin(veya bir ses şifreleyicisinin) MS’de bulunan GSM kodlayıcısının çıkışına yerleştirilmesidir. Bu yöntemin sorunu, BSS’de iletimin ancak GSM Tam Kapasite (GSM Full Rate- GSM FR) ‘dan PCM’e olacak biçimde tasarlanmış olmasıdır. Bunun için , gönderilen veri şifreli bir ses ise BSS’de çözülmesi veya bir veri ise sese benzetilmesi gerekmektedir. Her baz istasyonunda böyle bir değişiklik yapılamayacağı aşikardır.



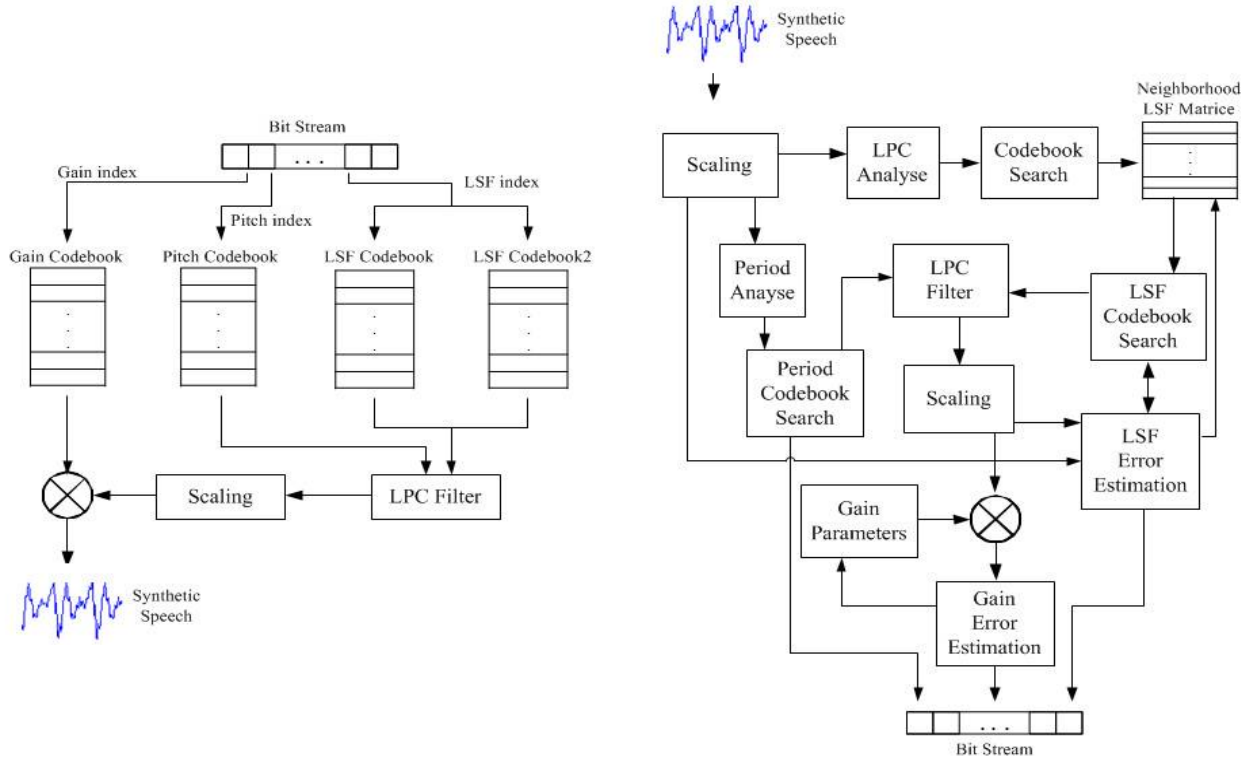
Şekil 2.2 : GSM ağında şifreleme kullanımına bir örnek[2].

İkinci yöntem ise Şekil 2.2’de gösterildiği gibi gönderilecek verinin MS’de GSM kodlayıcısının girişine konmasıdır. Ancak bu sefer de gönderilecek verinin iletimine (resim,video veya şifreli konuşma) , daha önce belirtildiği gibi GSM kodlayıcısı yalnızca konuşma verilerini iletmek üzere tasarlandığı için hatalar meydana gelecektir[2]. Ayrıca peş peşe kodlamadan ötürü verileri taşıyan sinyal birkaç kez kodlama ve çözümlenmelerden geçecektir. Sinyallerde meydana gelen ufak hatalar, iletilen verilerde daha büyük hatalara yol açabilir[2]. Bu sebeple sentetik ses tasarımında ileriki bölümlerde ele alınacak konulara hassasiyet gösterilmesi gerekmektedir.



**Şekil 2.3 :** GSM ses kodlayıcılarında ses aktivitesi analizi[5].

GSM hattındaki ses aktivitesi ve benzerliği tespiti; ortalama gönderim gücünü azaltarak kanallar arası karışmanın bastırılması, Kod Bölmeli Çoklu Erişim (Code Division Multiple Access- CDMA) sisteminin kanal kapasitesinin artırılması, mobil istasyon ve vericilerin akü enerjilerinin verimli kullanılması, paket bazlı sistemlerde paket kaybının en aza indirilmesi ve iletimde kullanılan bit oranının azaltılması gibi önemli sebeplerle uygulanmaktadır [5].



**Şekil 2.4 :** GSM Modem verici(sol) ve alıcı(sağ) şemaları [6].

Tez kapsamında tasarlanan modemın genel şeması Şekil 2.4’de verilmiştir. Aktarılabacak olan veri sayısal bit dizisi formunda olacaktır. Tasarlanan sistemde bit dizisinin 10 biti sentez filtresi, 4 biti periyot bilgisi, 2 biti de enerji ve genlik bilgisi için ayrılmış, böylece gönderilecek bir adet veri paketi 2 bayt olarak belirlenmiştir. Bu girdiler daha sonra GSM üzerinden aktarımda ikinci bölümde değinilen problemlere ve risklere karşı geliştirilmiş özel veritabanlarına aktarılır ve her girdi tıpkı bir Random - Access Memory (RAM) işaretçisi gibi, veritabanında bir adres olarak algılanır ve o adreslerdeki, periyot, Doğrusal Tayf Frekansları (Linear Spectral Frequencies- LSF) ve enerji verilerinden sentetik ses sentezi gerçekleştirilmiş olur.

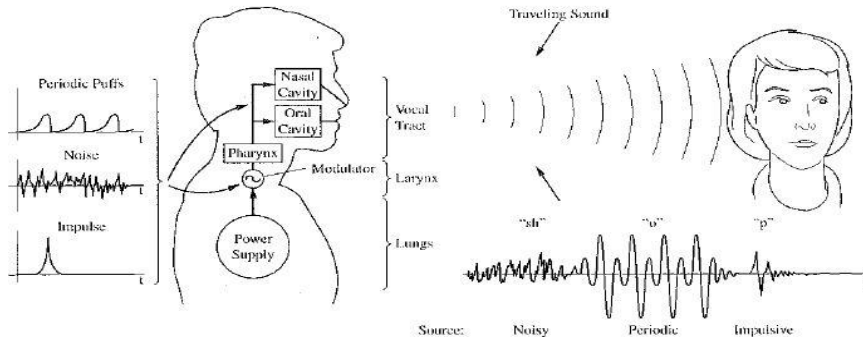
Artık analog olan ses GSM kodlayıcılarından modüle edilip tekrar demodüle edildikten sonra örnekleme ile dijital hale geçirilir ve paralel olarak sentezdeki işlemlerin tersi, periyot analizi, LPC analizi ve enerji analizleri yapılarak bu parametrelerin veritabanlarındaki adresleri kontrol edilir ve hata başarımını arttırmak için veritabanlarındaki belirli komşuluktaki adreslerden de sentez yapılarak orjinal ses ile karşılaştırılır, en başarılı sonucu veren adresler, gönderimi yapılan veriler olarak kabul edilir.

### 3. DOĞRUSAL ÖNGÖRÜ METODU İLE SESİN KODLANMASI

LPC bir ses kodlama metodudur. Dalga biçimi kodlayıcıların aksine ses kodlayıcılarında, sesin analiz edilip parametrelerinin çıkartılması ve işlenmesi gerekir. Bu aşamalar sisteme ek karmaşıklık ve işlem yükü getirse de, dalga şekli kodlayıcılarına göre daha yüksek sıkıştırma oranları elde etmek mümkündür[7]. Bir ses çerçevesinin LPC ile kodlanması için, çerçevenin selsiliği, periyodikliği ve model filtre parametreleri hakkında bilgiye ihtiyaç vardır. İlerleyen bölümlerde, insan konuşmasının analog haldeki ses sinyalinden dijital veri haline geçirilmesi, parametrelerine ayrıştırılması, sıkıştırılması, modellenmesi, yeniden sentezlenmesi konularına değinilecektir.

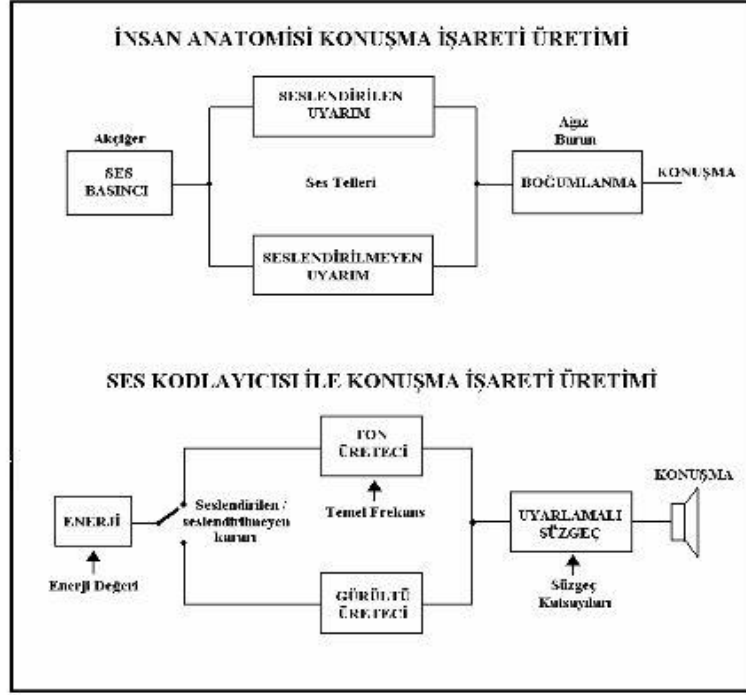
#### 3.1 İnsan Konuşmasının Modellenmesi

İnsanda periyodik ve düzenli, gürültü benzeri ve darbe tipli üç ana ses üretimi mevcuttur. Periyodik ve düzenli seslere 'u', 'o', 'a' örnek olarak verilebilir. Alfabemizdeki sesli harfler periyodik ve düzenli ses üretimi sınıfındadır. Sessiz harfler kendi içinde iki sınıfta incelenebilir, ilk sınıf yüksek frekanslı beyaz gürültü ile modellenebilecek yapıda olan 's', 'ş', 'h' gibi harflerdir, ikinci sınıf ise sürekli olarak çıkarılamayan, darbe kaynağı gibi zamanın belirli bir anında yüksek güç ihtiva eden 'p', 't', 'g' gibi harflerdir [8]. Akciğerler devrenin güç kaynağıdır, harflere göre gürültü, darbe veya düzgün periyodik etki fonksiyonları, akciğerlerden gelen hava boğaz, dil ve burundan oluşan vokal sistemde şekillendirilerek üretilir[6,8].



Şekil 3.1 : İnsanda sentezlenen ses çeşitleri [8].





**Şekil 3.2 :** İnsanda sesin modellenmesi. İnsan ses sentez yapısı ile LPC bloklarının karşılaştırılması(üst) [9].

Şekil 3.3’ de paylaşılan şemadan görülebileceği gibi, LPC ile insan ses üretimleri büyük benzerlik göstermektedir, akciğer bir enerji kaynağı olarak, ses telleri, düzgün periyodik ses ve gürültü üreteçleri olarak, ağız ve burun ise LPC sentez filtre katsayıları ile modellenmektedir.

### 3.2 Sesin Örneklenmesi ve Çerçeveleme

Orijinal formunda analog olarak bulunan ses veya konuşmanın doğru analiz edilebilmesi için öncelikle doğru biçimde sayısal forma geçirilmelidir. Bu geçişte iki önemli alt blok, aşağıdaki bölümde incelenecek olan örnekleme ve çerçeveleme aşamalarıdır.

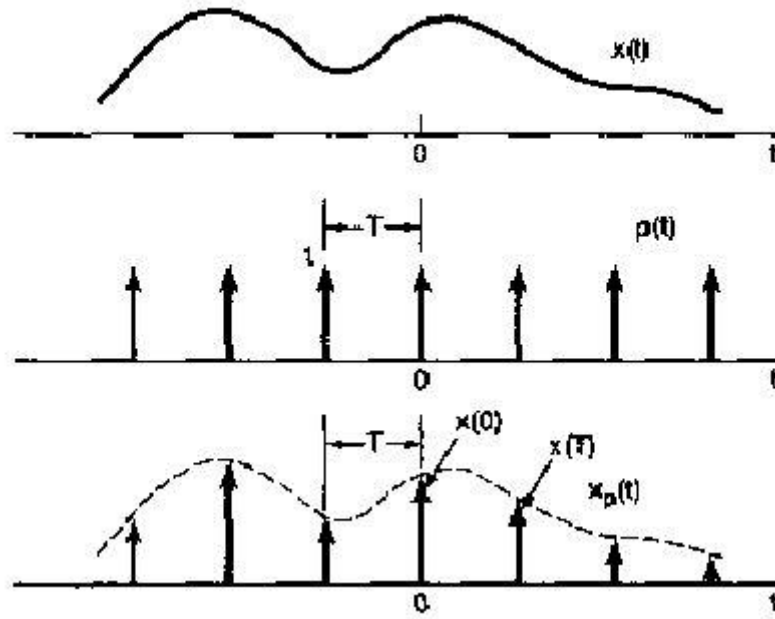
#### 3.2.1 Örnekleme

Örnekleme, analog bir sinyalden zamanda eşit aralıklarla örnekler alınması, böylece zamanda sürekli olan sinyalin, zamanda süreksiz hale getirilerek, bir örnekler dizisi elde edilmesidir.

Doğru olarak örneklenmiş bir sinyal yeniden hatasız biçimde zamanda sürekli hale getirilebilir. Buna en güzel örneklerden biri video sistemleridir. Videolarda da zaman

içinde belirli aralıklarda resim parçaları ardı ardına sıralanır, insan gözü bir filtre gibi davranarak zamanda ayırık olan resimleri sanki akıcı bir dizgeymiş gibi birleştirir [10].

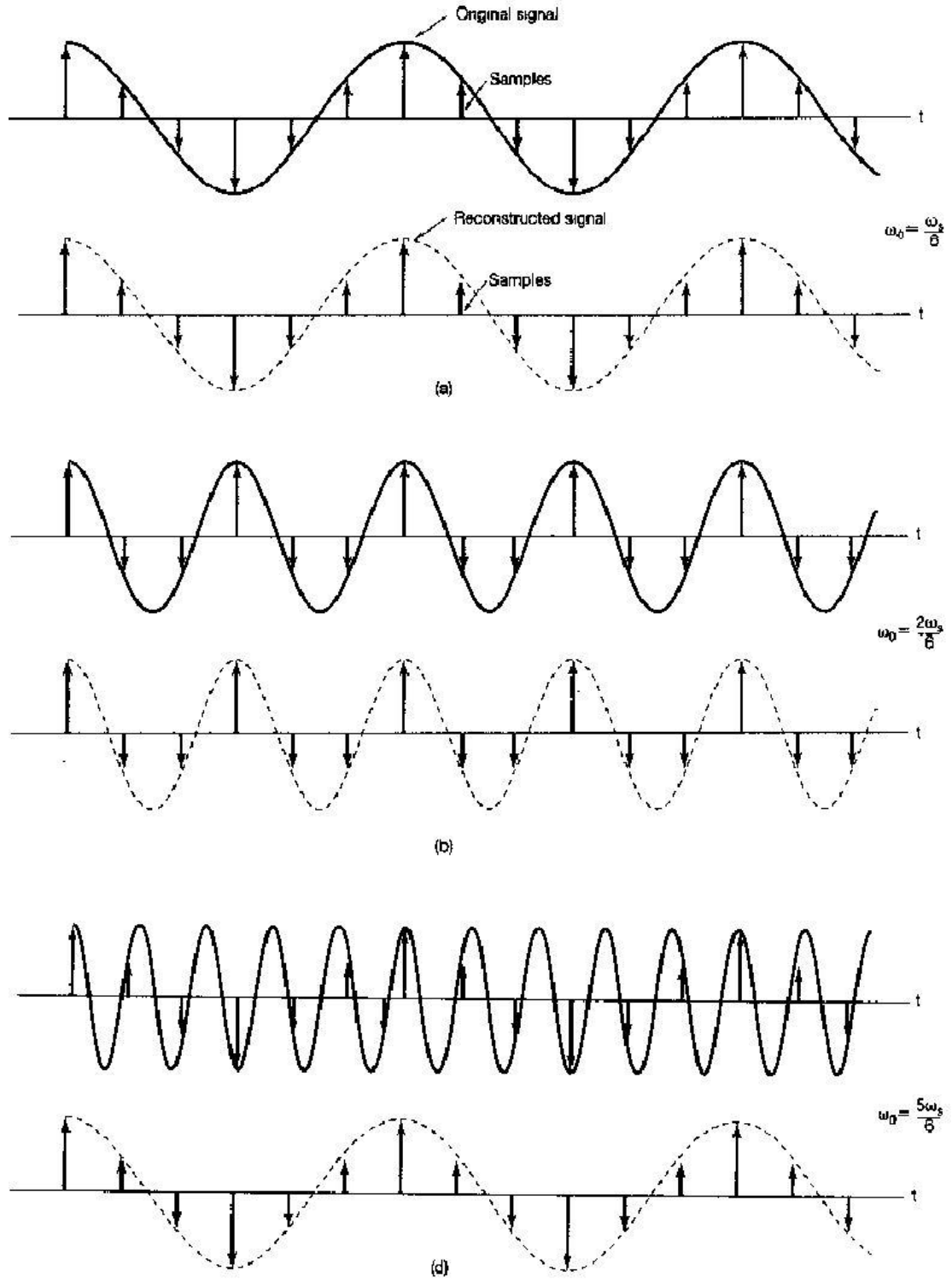
Örnekleme teorisi için yukarıda belirtilen doğru örnekleme kavramı hem zaman hem de frekans bölgesinde ayrı ayrı anlaşılabilir. Zamanda sürekli bir sinyalin örneklenmesine ilişkin gösterim Şekil 3.4'de gösterilmiştir.



Şekil 3.3 : Zamanda sürekli bir sinyalin örneklenmesi [10].

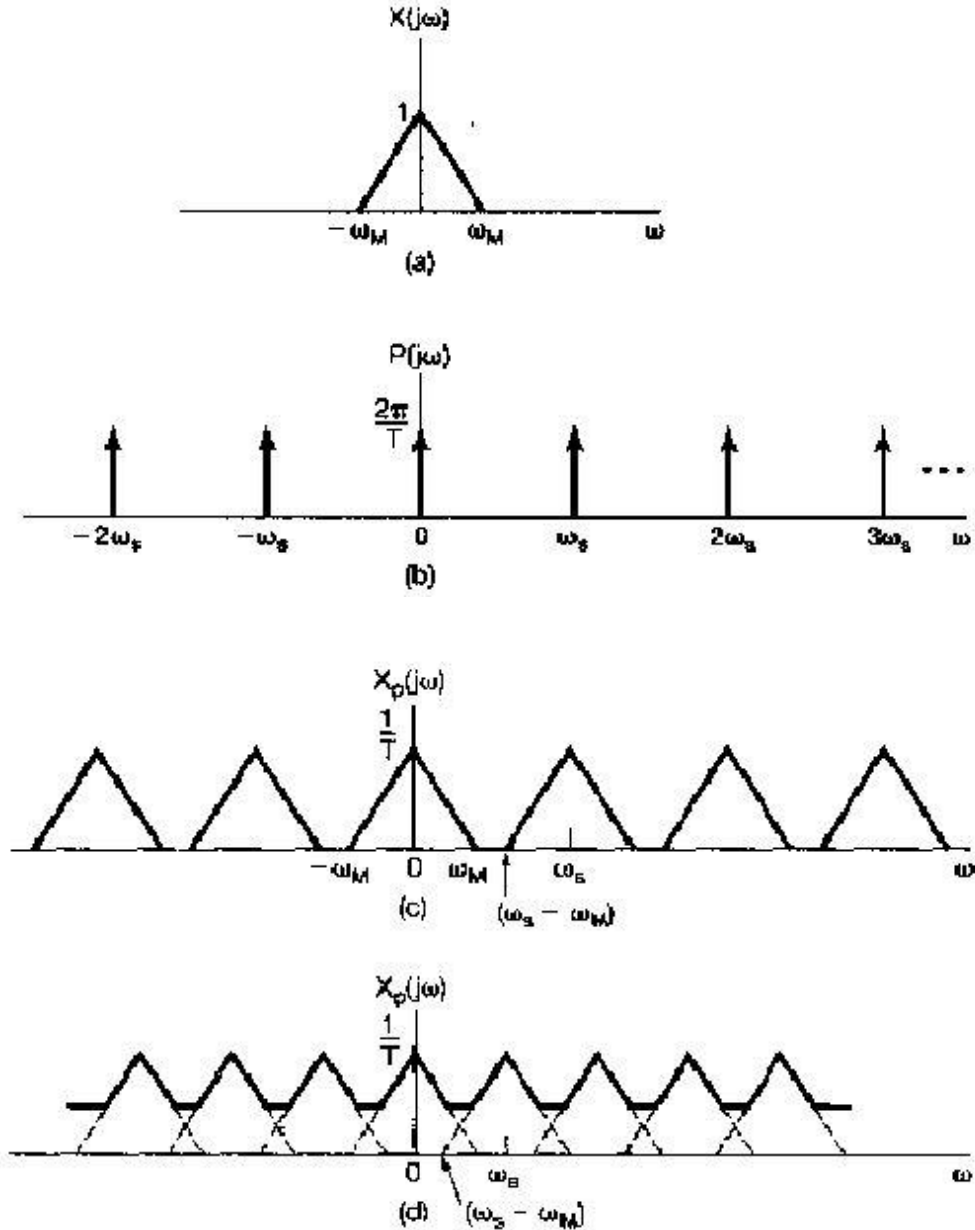
Örneklenen bir sinyalin, yeniden zamanda sürekli orijinal formuna getirilebileceği belirtilmişti, Şekil 3.4'de darbe katarı ile sinyal örneklenmektedir, kayıpsız geri dönüşüm için darbe katarı frekansının sinyalin frekans spektrumundaki en yüksek frekanstan en az iki kat büyük olması gerekir [10]. O halde  $F_s$  örnekleme frekansı,  $F_m$  de frekans spektrumundaki maksimum frekans ise, aralarındaki bağıntı  $F_s \geq 2 * F_m$  ile gösterilebilir.

Bu teoremin doğruluğunu zaman bölgesinde incelemek için Şekil 3.5'den faydalanılabilir. Bu şekilde inceleme kolaylığı açısından sinyal tek ton sinüsdür. Örnekleme frekansı sinyal frekansına göre ilk iki örnekte yeterli iken üçüncü durumda yetersizdir ve bu sebeple geri dönüşüm hatalı gerçekleşmektedir.



Şekil 3.4 : Zaman bölgesinde örnekleme teoremi[10].

Frekans bölgesinde örnekleme teoremini anlamak için ise Şekil 3.5' incelenebilir. Burada sinyalin ve darbe katarının frekans tayfları görülmektedir. Zamanda iki sinyalin çarpılması, frekans bölgesinde tayfların konvolüsyonuna karşılık gelir[10]. Sinyalin tayfı, örnekleme hızının düşük olması durumunda Şekil3.5' de (d) bölümünde görüldüğü gibi iç içe geçer ve geri dönüştürülemeyecek biçimde hata meydana gelir.



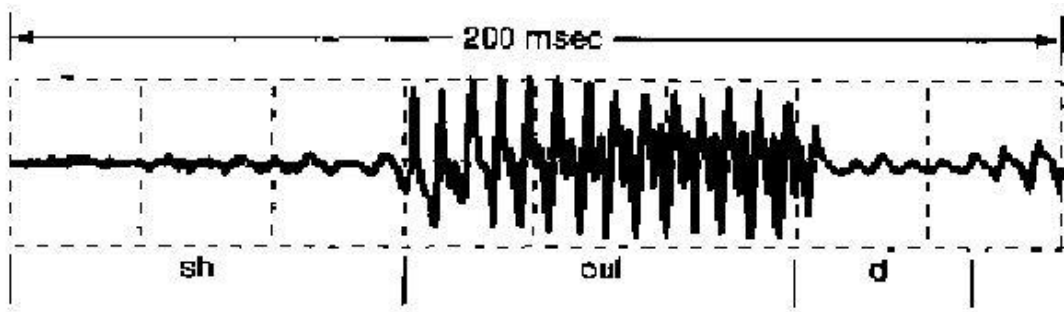
**Şekil 3.5 :** Frekans bölgesinde örnekleme teoremi.Örnek sinyalin frekans tayfı(a), darbe katarı(b), örnekleme teoremine uygun frekans ile yapılan örnekleme sonucu oluşan tayf(c), örnekleme hızının yeterli olmaması durumunda tayfin geri dönüştürülemeyecek biçimde bozulması(d) [10].

İnsan kulağı yaklaşık olarak 22KHz 'lik bir algıya sahiptir. CD'lerdeki ses formatlarının 44KHz olmasının sebebi de buradan anlaşılabilir. Günlük konuşmalarda ise ses frekansı 3KHz in üzerine çıkmaz.

Örneklemede bir ikincil parametre zamanda olduğu gibi genlikte de süreksizleşme, kuantalamadır. Analog, genlikte sürekli bir sinyalin dijital bir sistemde ancak sınırsız bit uzunluğu ile ifade edilebilir. Bu sebeple genlikte de süreksizleşme elzemdir. Ancak genlikte ayrıklaşma, zamanda ayrıklaşmanın aksine sinyalin hatasız geri dönüşümüne olanak tanımaz. Bu sebeple örnekleme teoremine uyulsa dahi, bir sinyal sayısallaştırılıp tekrar sürekli hale getirildiğinde genlik hatasız olarak geri elde edilemeyecektir.

### 3.2.2 Çerçeveleme

Konuşmanın durağan bir sinyal olarak kabul edilip işlenebilmesi için, sesin kısa (20ms mertebesinde) çerçevelere alınması gerekir. İşlenmesi gereken sesin büyüklüğü arttıkça, işlem karmaşıklığı üssel olarak artış gösterir. Gerçek zamanlı sistemlerde, konuşmanın tamamının bitmesi beklenip daha sonra işlem yapılamaz, uzun bir konuşmanın tamamen kaydedilip işleme tabi tutulması yüksek hafıza alanlarına da gereksinim oluşturacaktır. Bu sebeple konuşmanın çerçeveleme gereklidir [11]. Her ne kadar sesler veya heceler kısa zaman aralıklarında durağan olsa da, ses ve hece geçişleri arasındaki bölümler analizlerde istenmeyen etkilere sebebiyet verebilir. Şekil 3.6'da İngilizce 'Should' kelimesine ait bir ses kaydı gösterilmiştir.

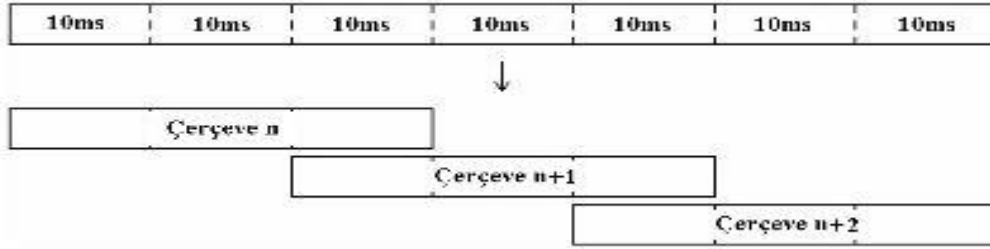


Şekil 3.6 : İngilizce 'Should' kelimesine ait bir ses kaydının bölümlere ayrılışı [10].

Şekilden görüldüğü gibi 'sh' beyaz gürültü, 'o' ve 'u' periyodik ve düzenli, 'd' darbe cevabı şeklinde modellenebileceği bölüm 3.1'de de değinildiği gibi görülmektedir. Şekil 3.6'da görülen 3 ana bölüm haricinde, çerçevelerin geçiş bölgelerine denk geldiği durumlarda, konuşmaya ait özelliğin yarısının bir çerçevede diğer yarısının bir

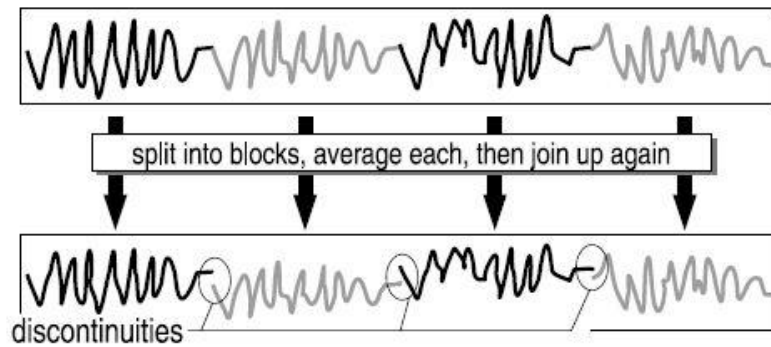
sonraki çerçevede kalma ihtimali ortaya çıkar. Bu durumda o özellik iki çerçevede de tamamıyla kaybolabilir, bu durumun önüne geçmek için çerçevelerin örtüştürülmesi gerekir.%25 ve %50 yaygın kullanımlardır [11].

Ancak %50 oranında örtüştürme, %50 daha fazla işlenecek veri anlamına da gelmektedir. Örtüştürmenin görsel betimi Şekil 3.7’de yapılmıştır. Şekilden görülebileceği gibi, 70ms’lik ses örneği örtüştürme sebebiyle, toplamda işlenmesi gereken 90ms’lik bir veri haline gelmiştir.

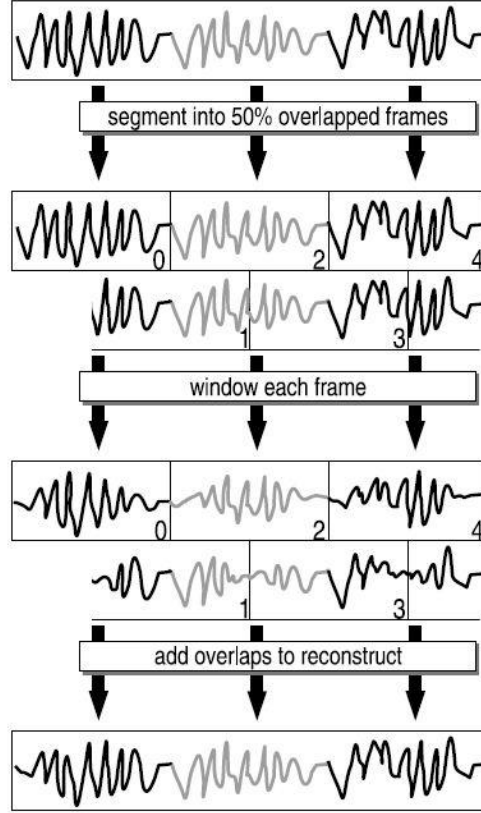


**Şekil 3.7 :** 30ms’lik çerçevelerin 1/3 oranında örtüştürülmesine bir örnek [9].

Analiz kalitesini arttırmak amacıyla örtüşme kullanıldığında, bu verilerden tekrar ses sentezlenmesi de önemli bir konudur. Çerçeveler basitçe arka arkaya eklenemez, bu durumda örneklenenden uzun süren bir konuşma sentezlenmiş olur. Örtüşme kullanılsa dahi, orijinal konuşma çerçevelere ayrılıp işlenip yeniden birleştirilmek istendiğinde komşu çerçeveler arasında seviye farkları sebebiyle süreksizlikler meydana gelecektir ve kulağa rahatsızlık verici sesler oluşturacaktır(Şekil 3.8) [11].



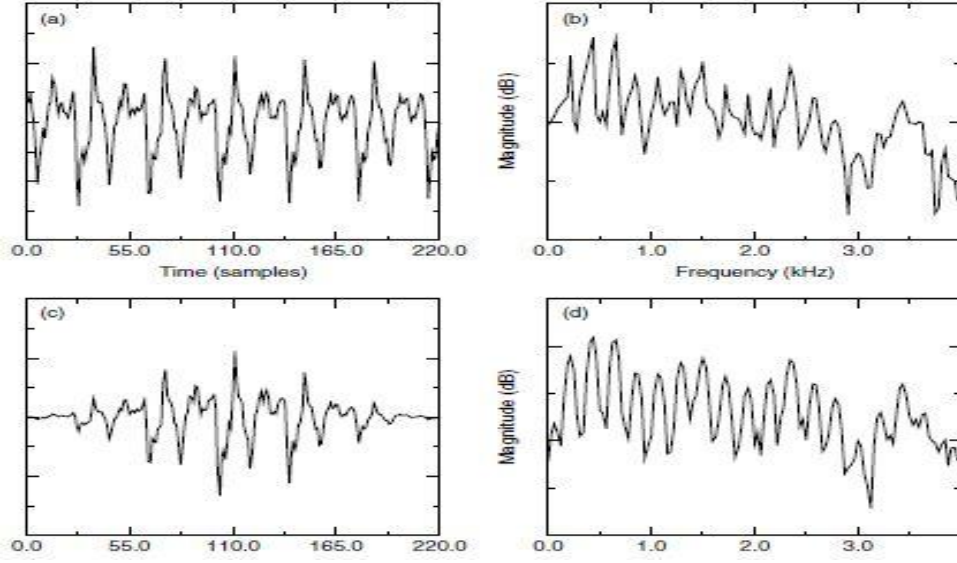
**Şekil 3.8 :** Ses çerçevelerinin işlenmesinin ardından yeniden birleştirilmeleri sırasında meydana gelen süreksizlikler [11].



**Şekil 3.9 :** Çerçeveleme, örtüştürme ve yeniden birleştirme şeması [11].

Bu problemlerin aşılabilmesi için, şekil 3.9’da verilen akış şemasında olduğu gibi, örtüştürülerek analiz edilen ve işlenen ses çerçevelerinin uygulamada arzu edilen özelliklere göre Hamming, Bartlett, Blackmann, Hanning gibi çeşitli çerçeve filtrelerinden geçirilip bundan sonra yeniden birleştirilip sentezlenmesi gerekli olacaktır. Böylece istenmeyen harmoniklerin önüne geçilebilir.

En temel çerçeveleme olan dikdörtgen fonksiyonu işlem kolaylığı sağlasa da, istenmeyen harmonikleri de sinyalle beraber getirdiğinden genellikle konuşmanın tayf analizlerinde kullanılmaz [5]. Hamming ile çerçevenilmiş bir sinyal Şekil 3.10’da verilmiştir. Hamming çerçevesinin etkisinin, dikdörtgen çerçeveye nazaran frekans tayfında daha yumuşak geçişler olduğu gözlemlenmektedir. Nasıl alçak geçiren bir filtre zaman bölgesinde bir sinyalde daha yumuşak geçişler sağlıyor ise, zamanda çerçevelemenin de bu yönüyle frekans bölgesindeki tayfta benzer bir etki gösterdiği görülmektedir.



Şekil 3.10 : Diktörge(üst) ve Hamming (alt) ile çerçevenilmiş sinyaller(sol) ve tayfları(sağ) [5].

### 3.3 Ses Çerçevesinden LPC Parametrelerinin Elde Edilmesi

Doğrusal öngörülü kodlama yöntemi, Birleşik Devletler Savunma Departmanı tarafından 1015 federal standardı içerisinde 1984 yılında önerilmiştir ve özünde bir sinyalin anlık değerinin, o andan daha önceki değerler yardımıyla bulunabilmesi, bir başka tabirle öngörülebilmesi olarak tanımlanır ve ayrık zaman bölgesindeki matematiksel tanımı (3.1) denklemindeki gibi ifade edilebilir [5,12].

$$s(n) \approx a_1 * s(n - 1) + a_2 * s(n - 2) + \dots a_p * s(n - p) \quad (3.1)$$

Bu bir fark denklemdir ve Z bölgesinde incelenecek olursa esasen bir filtredir ve Şekil 3.3’de daha önce gösterildiği gibi, insanda ağız ve burun tarafından boğumlanma ile sese şekil verilmesinin(vokal sistemin) matematiksel modelidir.

LPC10 kodlamasının kullanıldığı sistemler incelendiğinde, insan konuşmasından alınan 8KHz’de 8 bitlik örnekler kodlanmadığı takdirde 64kb/s ‘lik bir veri aktarımı gerekirken, LPC sayesinde veri iletiminin 2.4Kb/s’ye sıkıştırıldığı görülmektedir.LPC10 ismini (1) denklemdeki öngörü derecesini ifade eden ‘p’ değişkenininin 10 olmasından almaktadır.

Bu bölümde sesin analiz edilebilmesi için gerekli olan 3 ana parametre, öz ilişki ve perde (periyot) bilgileri ile seslilik sınıflandırmasına değinilecektir.

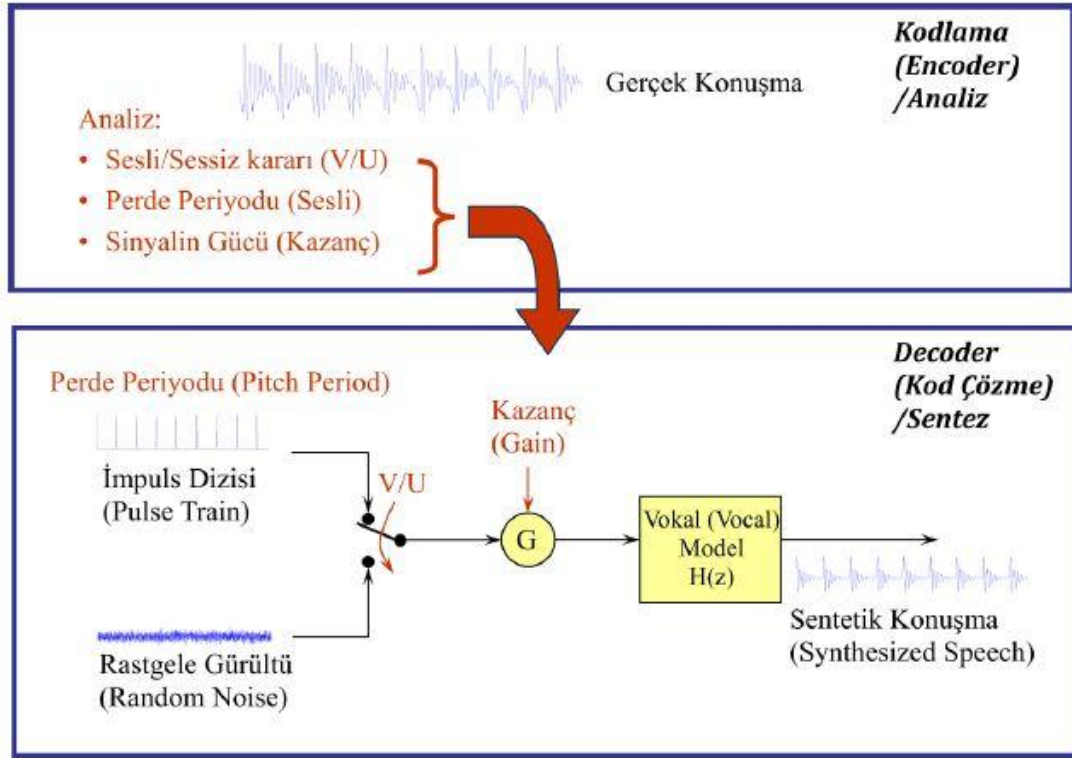


### 3.3.1 Öz ilişki Parametrelerinin Elde Edilmesi

Bir önceki bölümde (1) denklemi ile verilen doğrusal öngörü ifadesi (2) denklemine aşağıdaki gibi genişletilebilir [13].

$$s(n) = Gu(n) + \sum_{k=1}^p a_k * s(n - k) \quad (3.2)$$

Burada 'n', ayrık zamanda o anki değerlere, 'p' öngörü derecesine, 's(n)' modellenen sinyale,  $a_k$  öngörü katsayılarına, u(n) Şekil 3.11 'de verilen LPC modelindeki rastgele gürültüye, 'G' 'de kazanç karşı gelmektedir. Sesli (a,e,i vb.) sesler için u(n) darbe katarıyla, sessiz (s,h,f vb.) sesler ise beyaz gürültüyle modellenir [6].



Şekil 3.11 : LPC Modeli [6].

Genel halini alan (2) denklemde 'Gu(n)' tahrik fonksiyonunu (ciğerler), ikinci kısım ise vokal modeli oluşturacaktır.

Bulunması gereken iki parametre olduğu açıktır, bunlardan biri özilişki parametreleri olan ' $a_k$ ' lar diğeri ise tahrik fonksiyonudur. Bu bölümde  $a_k$  öngörü katsayılarının elde edilmesinde temel adımlar gösterilmiş, tezin kapsamı bu matematiksel denklemlerin çıkarılışından çok gömülü sistemlerde gerçekleşmesi ve problemlerin

tespit edilmesi olduğundan dolayı detaylı anlatıma girilmemiştir. Teorik çıkarımların daha detaylı incelemesi literatürde bulunan kaynaklardan yapılabilir [5,6,13]. Bu tez kapsamında LPC parametrelerinin LSF parametrelerine kuvantalanması kısmında teorik çalışmaya ağırlık verilmiştir.

Daha önce verilen (3.2) denkleminde vokal kısım öngörü ile sentezlenen yapay ses olarak nitelendirilip (3.3) denklemini olarak tanımlansın [13].

$$\hat{s}(n) = \sum_{k=1}^p a_k * s(n - k) \quad (3.3)$$

O halde orijinal (3.2) ses ile, öngörü ile sentezlenmiş ses (3.3) arasındaki fark bize hatayı verecektir(3.4).

$$e(n) = s(n) - \hat{s}(n) = s(n) - \sum_{k=1}^p a_k * s(n - k) \quad (3.4)$$

Bu denklemin, Z bölgesinde hata için transfer fonksiyonu çıkarılırsa (3.5) denklemini elde edilir.

$$A(z) = \frac{E(z)}{S(z)} = 1 - \sum_{k=1}^p a_k * z^{-k} \quad (3.5)$$

Öngörü katsayılarının belirlenebilmesi için kısa-zaman zarfında periyodik olarak kabul edeceğimiz sinyal ve hata fonksiyonları (3.6) ve (3.7) denklemlerindeki gibi tanımlanır.

$$s_n(m) = s(n + m) \quad (3.6)$$

$$e_n(m) = e(n + m) \quad (3.7)$$

Hedeflenen, herhangi bir 'n' anında (3.8)'de verilen hata bağıntısını minimize etmektir.

$$E_n = \sum_m e^2(m) \quad (3.8)$$

Bu denklemini genişleterek, n. sentezlenen sinyalin hatası için genel ifade (3.9) bağıntısı ile elde edilir.

$$E_n = \sum_m \left[ s_n(m) - \sum_{k=1}^p a_k * s_n(m-k) \right]^2 \quad (3.9)$$

Hatayı en aza indirmek için, (3.9) ifadesinde (3.10) ifadesindeki gibi her ‘ $a_k$ ’ değerine göre türev alıp sıfıra eşitlenerek (3.11) eşitliği elde edilir.

$$\frac{dE_n}{da_k} = 0, \quad k = 1, 2, \dots, p \quad (3.10)$$

$$\sum_m s_n(m-i) * s_n(m) = \sum_{k=1}^p a_k \sum_m s_n(m-i) * s_n(m-k) \quad (3.11)$$

Burada  $\sum_m s_n(m-i) * s_n(m-k)$  özel bir ifade olup, kovaryans bağıntısı olarak tanımlanır (3.12).

$$\varphi(i, k) = \sum_m s_n(m-i) * s_n(m-k) \quad (3.12)$$

Böylece (3.12) ve (3.11) denklemleri birleştirilerek, ifade (3.13) ‘deki gibi kovaryanslar cinsinden aşağıdaki gibi gösterilebilir.

$$\varphi_n(i, 0) = \sum_{k=1}^p a_k * \varphi_n(i, k) \quad (3.13)$$

Bulunan eşitlikler, (3.9)’da yerine koyularak hatayı en aza indiren eşitlik (3.14)’deki gibi elde edilir.

$$E_n = \varphi_n(0,0) - \sum_{k=1}^p a_k * \varphi_n(0, k) \quad (3.14)$$

Çözümün elde edilebilmesi için,  $\varphi_n(i, k)$ ,’nin  $1 \leq i \leq p$ ,  $0 \leq k \leq p$  için oluşturulup ‘ $p$ ’ kadar denklem setinin çözümünden  $a_k$  katsayılarının hesaplanması gerektiği anlaşılmıştır [13].

### 3.3.2 Doğrusal Öngörü Katsayılarının Elde Edilmesi

Bir önceki bölümde, (3.11)'in içerisinde (3.12) ile tanımlanan kovaryans bağıntısında geçen 'm' için bir limit olmadığı görülmektedir. Sınırı olmayan bir işlem kümesinin gerçek hayatta ifade edilemeyeceği aşikardır. Bunu çözümlerin bir yolu, sinyalin  $0 \leq m \leq N - 1$  bölgesi dışında sıfır olduğunun varsayılması ile 'm' parametresine bir sınır getirmektir. Bu yöntem öz ilişki (autocorrelation) metodu olarak adlandırılır. Gerçekte yukarıda tanımlanan çerçeve dışında  $N \leq m \leq N + p$  bölgesinde, 'sıfırların' öngörüldüğü ve aslında sinyal sıfır olmayacağından ötürü, çerçeve köşelerinde hatalar sıfıra indirgenemeyecektir [5].

Sınırlı çerçeve yaklaşımı ile öz ilişki bağıntısı (3.15) 'deki gibi, öz ilişki ve kovaryans eşitliklerinin birbiriyle olan bağıntısı ise (3.16) denkleminde tanımlanmıştır [6].

$$r_n(i) = \sum_{m=1}^p s_n(m) * s_n(m + i) \quad (3.15)$$

$$r_n(\tau) = r_n(|i - k|) = \varphi_n(i, k) \quad (3.16)$$

(3.15) ve (3.16)'dan faydalanılarak (3.13) bağıntısı öz ilişki cinsinden aşağıdaki (3.17) gibi ifade edilir.

$$\sum_{k=1}^p a_k * r_n(i - k) = r_n(i) \quad (3.17)$$

(3.17) denklem takımı, Şekil-3.12'de gösterildiği gibi matris formunda açılarak incelenirse, matrisin simetrik ve köşegen üzerindeki elemanlarının eşit olması sebebiyle  $A_{i,j} = A_{i+1, j+1}$  Toeplitz yapısında olduğu görülebilir.

$$\begin{bmatrix} R_n(0) & R_n(1) & \dots & R_n(p-1) \\ R_n(1) & \dots & \dots & R_n(p-2) \\ \vdots & \vdots & \vdots & \vdots \\ R_n(p-1) & \dots & \dots & R_n(0) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_p \end{bmatrix} = \begin{bmatrix} R_n(1) \\ R_n(2) \\ \vdots \\ R_n(p) \end{bmatrix}$$

**Şekil 3.12 :** Öz ilişki denklem takımı ve doğrusal öngörü katsayıları [5].

(3.17) denklem takımı teorik olarak her ne kadar  $p \times p$ 'lik bir matrisin tersi ile çözümlenip doğrusal öngörü katsayıları elde edilebilse de, sayısal sistemlerde sınır bit sayısı,

dolayısıyla sınırlı ondalık sebebiyle hatalara sebebiyet verebilmektedir. Öte yandan Toeplitz karakteristiklerinden faydalanılarak, doğru ve hızlı biçimde çözüme ulaşılabilir. En çok tercih edilen yöntem Levinson – Durbin algoritmasıdır [5,6]. Aşağıda sunulan adımlar takip edilerek çözüme ulaşılır [13].

1.  $s_0 \equiv r(0)$ , olarak tanımlanır. (3.18)

2.  $i = 0, 1, \dots, p - 1$  için aşağıdaki işlemler tekrar edilir

- a.  $\rho_{i+1} = \frac{r(i+1) + \sum_{k=1}^i a_{i,k} * r(i+1-k)}{s_i}$  (3.19)

- b.  $s_{i+1} = s_i(1 - \rho_{i+1}^2)$  (3.20)

- c.  $a_{i+1,k} = a_{i,k} - \rho_{i+1} * a_{i, i+1-k}, \quad 1 \leq k \leq i$  (3.21)

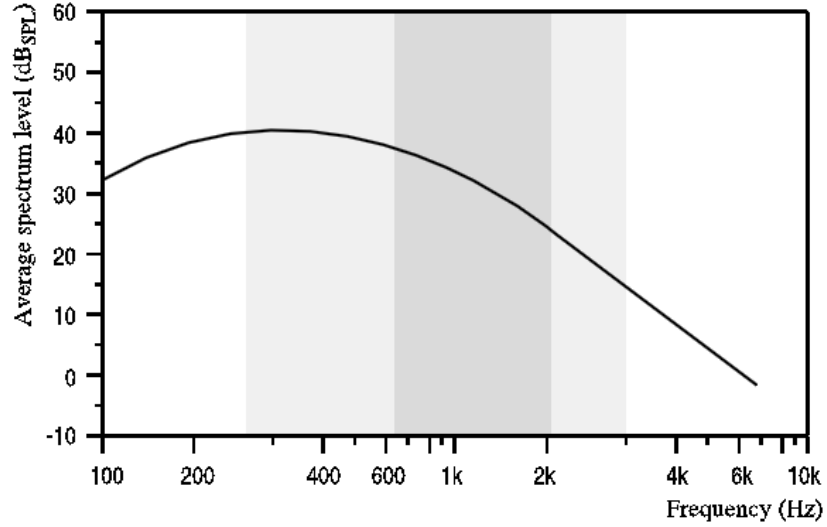
- d.  $a_{i+1,i+1} = -\rho_{i+1}$  (3.22)

Bu algoritma ile elde edilen  $a_{i+1,i+1}$  katsayıları, (3.5) denkleminde verilmiş olan  $a_k$  katsayılarına eşittir. Bu katsayıların eldesiyle, ilgili ses kümesi için insandaki vokal bölüme denk düşen LPC analiz filtresi modellenmiş olur.

Bu katsayıların en önemli özelliği, modellenen filtrenin kararlı olmasıdır [5,6]. Gerçekleme konusunda bir diğer önemli nokta, çözümün yaklaşık  $p^2$  adet çarp-topla işlemi içermesidir. Matris tersi ile çözüme gidilmesi durumuna nazaran, Levinson-Durbin algoritması ile onuncu dereceden ( $p=10$ ) bir sistem için 25 kat daha az aritmetik işleme ihtiyaç duyulmaktadır [13].

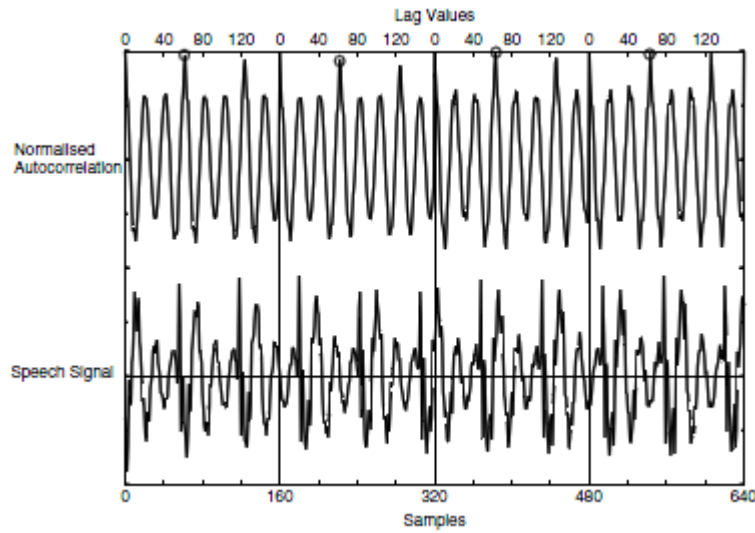
### 3.4 Perde (Periyot) Bilgisinin Eldesi ve Seslilik Sınıflandırması

İnsan kulağı 22KHz'e kadar olan sesleri algılayabilmektedir. Ancak daha önce değinildiği gibi günlük konuşmalarda ses 2KHz'i geçmezken, çerçevelenmiş bölümlerde var olan sesli harflerin temel frekans bileşeni 500Hz'i geçmez [11]. Harmonikler ve diğer frekans bileşenleri 2KHz band genişliğini oluşturmaktadır. Temel frekans bileşeni olarak adlandırılan perde(pitch)'nin hesaplanması için çeşitli yöntemler mevcut olup, güvenilirlik ve işlem yükü bakımından birbirine karşı avantaj ve dezavantajlara sahiptirler.



**Şekil 3.13** : İnsan konuşmasının frekans tayfı [11].

Özilişki bağıntısı, sinyalin kendi örnekleri arasındaki benzerliğin bir ölçütü olduğundan dolayı, sinyalin kendisi ile zamanda ötelenmiş (lag) halinin özilişki katsayıları elde edilerek perde tahmin edilebilir. Özilişki değerinin en yüksek olduğu öteleme, aranan periyot değerine karşılık düşecektir. Her öteleme için özilişki hesaplanması sisteme yük getirmesine rağmen, ileriki bölümlerde bahsedileceği gibi başarımından ötürü diğer yöntemlere göre tercih edilmektedir.



**Şekil 3.14** : Özilişki değerleri yardımıyla perdenin hesaplanması. Ötelenmiş sinyalin kendisiyle olan öz ilişki değeri(üst) ile sinyalin zamandaki ifadesi(alt) [5].

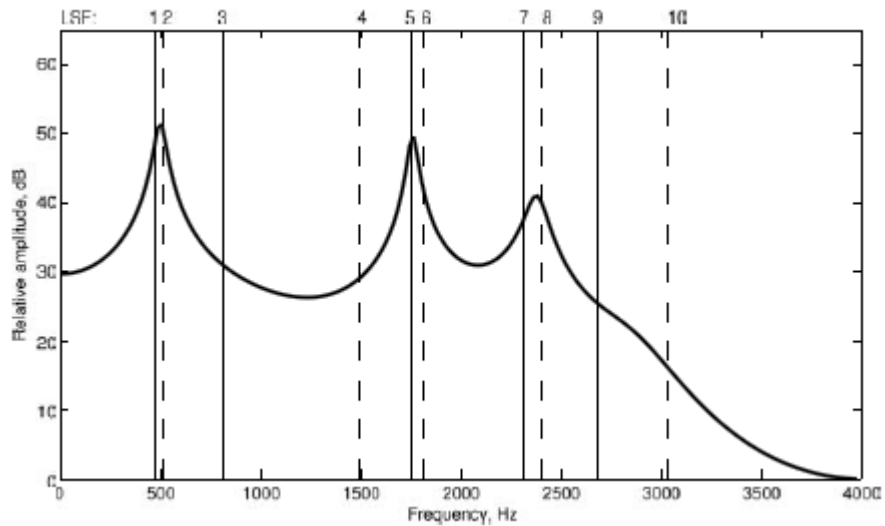
İnsan konuşmasının frekans aralığı ve yapılan gerçeklemede örnekleme hızının 8Khz olması dolayısıyla, öz ilişki değerlerinin 20 ila 150 örnek öteleme arasında aranması

uygun olacaktır. Periyodiklik bu bölgenin dışına taşıyor ise, incelenilen sesin sessiz (beyaz gürültü modeli) olduğu kanaatine varılabilir [6].

### 3.5 LPC – LSF Parametre Dönüşümü

LPC parametrelerinin LSF parametrelerine dönüştürülmesindeki temel amaç, LSF katsayılarının kuantalamaya karşı daha uygun olmasından kaynaklanmaktadır. LPC değerlerinin kuantalama sebebiyle kırılan ve kaybolan bitleri, sesi modelleyen sentez filtresinin karasızlaşmasına veya seste kulağı rahatsız edecek parazitlerin oluşmasına sebebiyet verebilir [5,11,14]. LSF parametrelerinde meydana gelecek kayıplar birazdan değinileceği gibi bu gibi etkilere daha dayanıklı olduğundan ötürü sisteme getirdiği yoğun işlem yüküne rağmen veri aktarımında tercih edilirler.

İnsan konuşmasının modellenmesinde Şekil 3.2 ile verilen şemada yapılan kabul, aynı anda sesli ve sessiz seslerin birlikte çıkarılmadığıdır. Bir seçici ya sesli sesler için periyodik ve düzenli bir ton üretir veya sessiz sesler için gürültü üretilir. Yani gırtlak ya kapalı ya da açıktır. Gerçekte ise gırtlak konuşma sırasındaki hızlı değişimler sebebiyle ne tam kapalı ne de açıktır. Bu sebeple hakiki rezonans frekansları LSF parametrelerinin Şekil 3.15’de gösterildiği gibi ara bölgelerinde oluşmaktadır [11]. Bu şekil aynı zamanda LPC parametreleriyle oluşturulmuş filtrenin frekans yanıtını ve LSF frekanslarının ilişkisini de göstermektedir.



Şekil 3.15 : LSF değerleri(dikey çizgiler) ve LPC tayfı arasındaki ilişki [11].

LPC-LSF parametre dönüşümü, daha sonraki bölümlerde detaylandırılacağı gibi sistemde en yoğun iş yüküne sahip fonksiyon olmuştur. Bunun sebebi yüksek dereceli bir polinomun köklerinin bulunmasına olan ihtiyaçtır.

P ve Q olmak üzere aşağıdaki gibi 2 fonksiyon tanımlansın.

$$P(z) = A_0 * z^p + A_1 * z^{(p-1)} + \dots + A_p \quad (3.23)$$

$$Q(z) = B_0 * z^p + B_1 * z^{(p-1)} + \dots + B_p \quad (3.24)$$

Bu denklem takımında;

$$A_0 = 1 \quad (3.25)$$

$$B_0 = 1 \quad (3.26)$$

$$A_k = (a_k - a_{p+1-k}) + A_{k-1} \quad k = 1 \dots p \quad (3.27)$$

$$B_k = (a_k + a_{p+1-k}) - B_{k-1} \quad k = 1 \dots p \quad (3.28)$$

olarak ifade edilebilir. Bu eşitliklerde  $a_k$  'lar LPC katsayılarını ifade etmektedir. Aranılan LSF parametreleri, P ve Q polinomlarının kökleridir. Bu kökler iki önemli özelliğe sahiptir;

1. P ve Q polinomlarının tüm kökleri Z bölgesinde birim çember üzerindedir
2. P ve Q polinomlarının kökleri birim çember üzerinde sırasıyla birbiri ardına gider ve sıfır ile pi arasındadır,  $0 \leq w_{q,0} < w_{p,0} < w_{q,1} < w_{p,1} < \dots \leq \pi$

P ve Q polinomlarının eldesi hakkında detaylı bilgi, ilgili kaynaklardan edinilebilir[5].

Görüldüğü gibi P ve Q polinomlarının köklerinin bulunması LPC-LSF dönüşümünün temelini oluşturmaktadır. Polinomlar simetrik olmalarından dolayı dereceleri p/2'ye (3.29) ve (3.30)'da verildiği gibi indirgenip, çözüm gerçek kök yöntemiyle aranabilir.

$$P(z) = z^{p/2} [ A_0 \left( z^{\left(\frac{p}{2}\right)} + z^{\left(-\frac{p}{2}\right)} \right) + A_1 \left( z^{\left(\frac{p}{2}-1\right)} + z^{\left(-\frac{p}{2}+1\right)} \right) + \dots + A_{p/2} ] \quad (3.29)$$

$$Q(z) = z^{p/2} [ B_0 \left( z^{\left(\frac{p}{2}\right)} + z^{\left(-\frac{p}{2}\right)} \right) + B_1 \left( z^{\left(\frac{p}{2}-1\right)} + z^{\left(-\frac{p}{2}+1\right)} \right) + \dots + B_{p/2} ] \quad (3.30)$$



Bütün köklerin birim çember üzerinde olduğundan ötürü (3.31) aşağıdaki gibi tanımlanabilir.

$$z = 1 * e^{j\omega}, \quad z^1 + z^{-1} = 2 * \cos(\omega) \quad (3.31)$$

(3.31) ile (3.29) ve (3.30) birleştirilirse;

$$P(z) = 2e^{jp\omega/2} [A_0 * \cos\left(\frac{p}{2}\omega\right) + A_1 \left(\frac{p-2}{2}\omega\right) + \dots + \frac{1}{2}A_{p/2}] \quad (3.32)$$

$$Q(z) = 2e^{jp\omega/2} [B_0 * \cos\left(\frac{p}{2}\omega\right) + B_1 \left(\frac{p-2}{2}\omega\right) + \dots + \frac{1}{2}B_{p/2}] \quad (3.33)$$

(3.32) ve (3.33) ile verilen denklem takımları elde edilir.  $x = \cos(\omega)$  değişken atamasıyla ve tez kapsamında belirlenen öngörü derecesi olan  $p=10$  için denklemler güncellenir ise (3.34) ve (3.35) elde edilir.

$$P_{10}(x) = 16A_0x_5 + 8A_1x_4 + (4A_2 - 20A_0)x_3 + (2A_3 - 8A_1)x_2 \\ + (5A_0 - 3A_2 + A_4)x + (A_1 - A_3 + 1/2A_5) \quad (3.34)$$

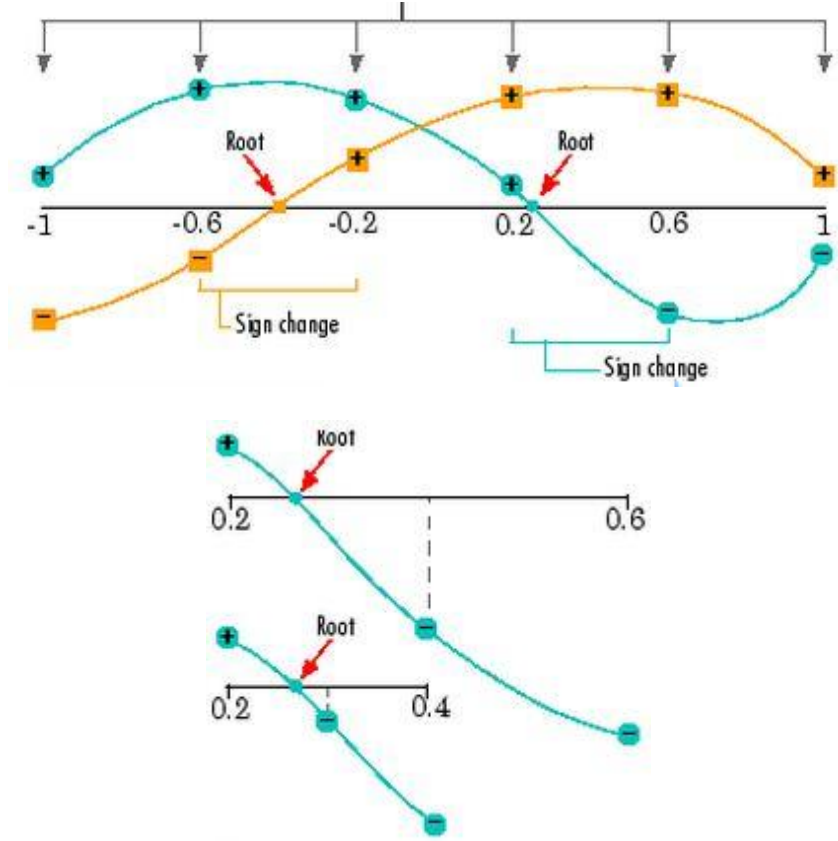
$$Q_{10}(x) = 16B_0x_5 + 8B_1x_4 + (4B_2 - 20B_0)x_3 + (2B_3 - 8B_1)x_2 \\ + (5B_0 - 3B_2 + B_4)x + (B_1 - B_3 + 1/2B_5) \quad (3.35)$$

Bu durumda aranan LSF değerleri aşağıdaki gibi(3.36) elde edilir:

$$LSF(i) = \frac{\cos^{-1}(x_i)}{2\pi T} \quad (3.36)$$

Bu durumda yapılması gereken işlem, kökleri -1 ile 1 arasında ve gerçek olan iki adet beşinci dereceden denklemin çözülmesidir. Kökler gerçek ve de çözüm aralığı belirli olduğundan ötürü kök bulmak için çift segment(bisection) yöntemi seçilmiştir.

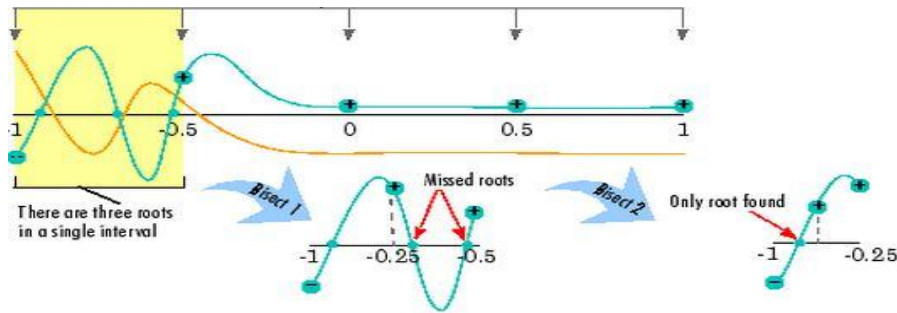
Çift segment yöntemi ismini, iki farklı adımda ve de koordinat sisteminin pozitif ve negatif bölümlerindeki değişimi aramasından ötürü almaktadır ve genel işleyiş şeması Şekil 3.16 ile verilmiştir. Burada hedef belirli değer aralıklarında fonksiyonun işaretini kontrol edip, işaret değişimi durumunda fonksiyonun x ekseninden geçtiği, yani o değer aralığında bir kökü olduğu anlaşılacaktır. Bu ilk aşamanın ardından, işaret değişimi bulunan aralıkta iterasyon yapılarak köke doğru adım adım yaklaşılır.



Şekil 3.16 : Çift segment yöntemi ile kök bulma[15].

Yöntemin kullanılmasında dikkat edilmesi gereken noktalardan ilki, kökün bulunması durumunu tanımlamak için sayısal olarak hata veya sıfır'ın sisteme tanıtılması gerekliliğidir. Tasarlanan kod kitaplarındaki hassasiyet göz önüne alınarak  $Sıfır(veya\ hata) = \pm 0.0001$  olarak tanımlanmıştır.

Yöntemin kullanılmasında dikkat edilmesi gereken noktalardan ikincisi ise, adım sayısının seçimidir. Şayet adım sayısı çok küçük seçilir ise işlem yükü de aynı oranda artacak ve işlem süresi uzayacaktır. Kök bulma işlemini hızlandırmak için adım aralığı görece büyük seçilmesi durumunda ise Şekil 3.17'de gösterildiği gibi kök atlanması meydana gelebilir.



Şekil : 3.17 : Adım sayısının görece büyümesinden ötürü kök atlanması durumu [15].

Bu sebeplerden ötürü Çift segment yaklaşımında hata değerinin tanımlanması ve de adım sayısının belirlenmesi, çözüm etkinliğini doğrudan etkilemektedir.

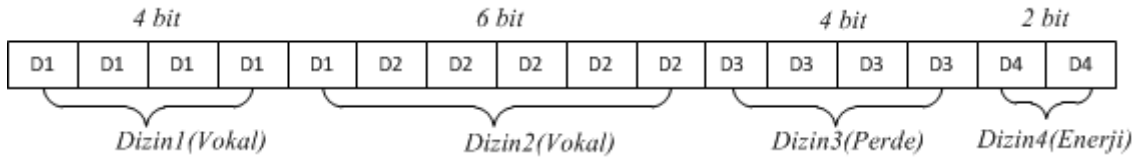
### 3.6 Modemin Verici Olarak Kullanılması

Modem, GSM ağını kullanarak arzu edilen herhangi bir bit dizisini göndermek üzere tasarlanmıştır. Şekil 2.4 ve GSM üzerinden iletim konusunda değinilmiş olan problemler hatırlanacak olursa, bit dizilerinin doğrudan Radyo Frekansı (Radio Frequency- RF) katına verilemeyeceği aşıkardır. Bu sebeple bit dizilerine, daha önce değinildiği gibi gönderimdeki uygunluğu belirlenmiş farkı perde ve yapılardaki ses kitapları karşı düşürölüp, RF katına bu bilgiler iletilecektir.

Modeme, iletilmesi arzu edilen 16 bitlik bir veri ulaştığında sırasıyla aşağıdaki işlemler gerçekleşecektir:

#### 1. Ayırıştırma

16 bitlik dizi, ilgili içeriklere ayırıştırılır. Bu içerikler vokal model, perde ve enerji olmak üzere üç sınıfa ayrılır. Ayırıştırma Şekil 3.18’de verilmiştir.



Şekil 3.18 : Gönderilmek üzere modeme gelen bir bit dizisinin ayırıştırılması.

#### 2. Vokal Modelin Oluşturulması

Vokal modelin, yani ses sentez filtresinin öncül (LSF) parametreleri toplamda 10 bit (öngörü derecesi sebebi) ile belirlenmektedir. Seçimin dört ve altı bitlik iki ayrı dizine ayrılmasının sebebi kod kitap uzunluğunun makul seviyede tutulmak istenmesidir. Doğrudan 10 bit girişli bir kod kitabı için  $2^{10} = 1024$  adet farklı vokal yapı gereklidir. Dört ve altı bitlik yapı için ise  $2^6 + 2^4 = 80$  adet vokal yapı yeterli olacaktır. Sonuç olarak birinci adım ile ayırıştırılmış vokal modele ilişkin LSF parametreleri, ilgili kod kitaplarından ikinci adımda seçilmiş olur.

#### 3. LSF – LPC Dönüşümü

İkinci adım ile seçilen LSF parametrelerinden sentetik ses sentezlenebilmesi için, ilgili LPC parametrelerine ihtiyaç vardır. LSF parametreleri, yani (3.34) ve (3.35) ‘in

kökleri denklem (3.37) ve (3.38)'deki gibi polinom  $z$  bölgesinde çarpılarak yeniden oluşturulabilir.

$$P_{p+1}(z) = z^{-(p+1)}[(1 - z) * (z - r_0) * (z - r_0^*) \dots (z - r_{p/2}) * (z - r_{p/2}^*)]$$

$$= S_0 + S_1 z^{-1} + \dots S_p z^{-p} + S_{p+1} z^{-(p+1)} \quad (3.37)$$

$$Q_{p+1}(z) = T_0 + T_1 z^{-1} + \dots + T_p z^{-p} + T_{p+1} z^{-(p+1)} \quad (3.38)$$

Bu eşitliklerde;

$$S_0 = 1 \quad (3.39)$$

$$T_0 = 1 \quad (3.40)$$

$$S_{p+1} = -1 \quad (3.41)$$

$$T_{p+1} = 1 \quad (3.42)$$

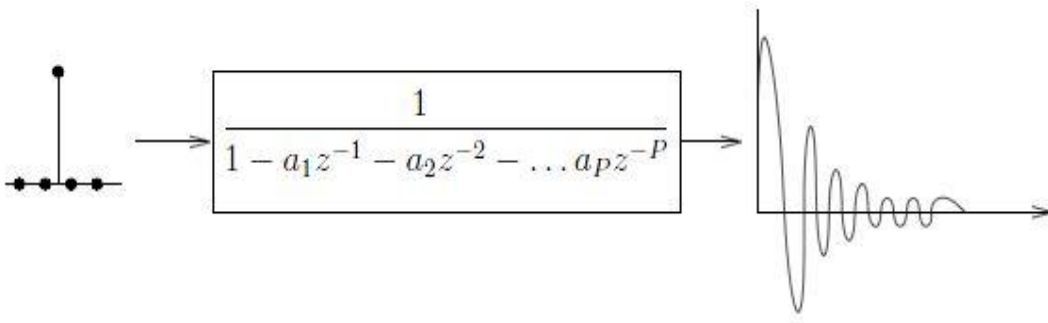
$$a_i = \frac{1}{2}(T_i + S_i), \quad i = 1, \dots, p/2 \quad (3.43)$$

$$a_{p+1-i} = \frac{1}{2}(T_i - S_i), \quad i = 1, \dots, p/2 \quad (3.44)$$

olarak verilir. (3.43) ve (3.44) ile LPC parametreleri ses sentezi için elde edilmiş olur.

#### 4. Ses sentezi

Belirlenen LPC parametreleri yardımıyla ses çerçevesi sentezlenmesi mümkündür. Bu parametreler tüm kutup bir filtrenin katsayılarıdır ve sentez akışı Şekil 3.19'da verildiği gibi özetlenebilir. LPC katsayıları sese şekli verir. Sentezlemeler sırasında tüm ses kitaplığı, GSM hattındaki gürültü bastırımlarından kaçınılmak için sesli seslerden oluşturulmuştur. Bir başka anlatımla, sentez filtresinin girişi periyodik darbe katarları biçimindedir. Darbe katarlarının periyodu, çerçevedeki perde bilgisine karşılık düşmektedir.



Şekil 3.19 : Sentez filtresi [16].

## 5. Enerji Bilgisi

Son aşama, sentezlenen sese enerji bilgisinin eklenmesidir. Sentezlenen sinyal öncelikle taşmaların engellenmesi için normalize edilir. Normalizasyon işleminin ardından kazanç kitaplığından ilgili bit dizisine karşılık düşen kazanç, sentezlenen ses dizisi ile çarpılarak, 16 bitlik bir verinin GSM üzerinden gönderimi için oluşturulan sentetik ses RF katına verilmeye hazır hale gelir.

### 3.7 Modemin Alıcı Olarak Kullanılması

GSM üzerinden gönderim için özel olarak sentezlenmiş ses RF katından geçip modeme ulaştığında, bu sesin analiz ve işlenmesi sonucu gönderilen 16 bit'in geri eldesi için sırasıyla aşağıdaki adımlar takip edilir:

#### 1. Örnekleme ve Çerçeveleme

RF katından gelen ses verileri, 8KHz'de 20ms'lik çerçevelerde örneklenir.

#### 2. Normalizasyon

Ses çerçevesi, işlemlerin doğru ilerlemesi için normalize edilir.

#### 3. LPC Parametrelerinin Eldesi

Ses çerçevesinin vokal olarak modellenmesi için gerekli olan LPC parametreleri daha önceki bölümlerde açıklandığı gibi elde edilir.

#### 4. LPC – LSF Dönüşümü

Elde edilen 10 adet LPC katsayısı, daha önce değinilen yöntemler vasıtasıyla LSF parametrelerine dönüştürülür. Elde edilen 10 adet LSF katsayısı, tasarım gereği 4 ve 6 bite ayrılarak, bu değerlerin kod kitaplığındaki karşılıkları aranır. Karşılığın bulunduğu dizin değeri, aranan 4 ve 6 bitlik değerlerdir.

#### 5. Perde'nin Eldesi

Ses çerçevesine ait perde bilgisi sinyalin farklı gecikmelerine ilişkin öz ilişki değerleri hesaplanılarak elde edilir. Böylece 4 bit daha elde edilmiş olur.

#### 6. Enerjinin Eldesi

Geriye kalan 2 bit için, çerçevelenmiş orijinal sinyal ile normalize edilmiş hali 4 farklı kazanç değeri için karşılaştırılarak hatanın en az olduğu kazancın dizin değeri aranan

son iki bite karşılık gelir. Böylece, verici modem tarafından 16 bit için gönderilen ses çerçevesinin analizi tamamlanmış ve 16 bit geri elde edilmiş olur.

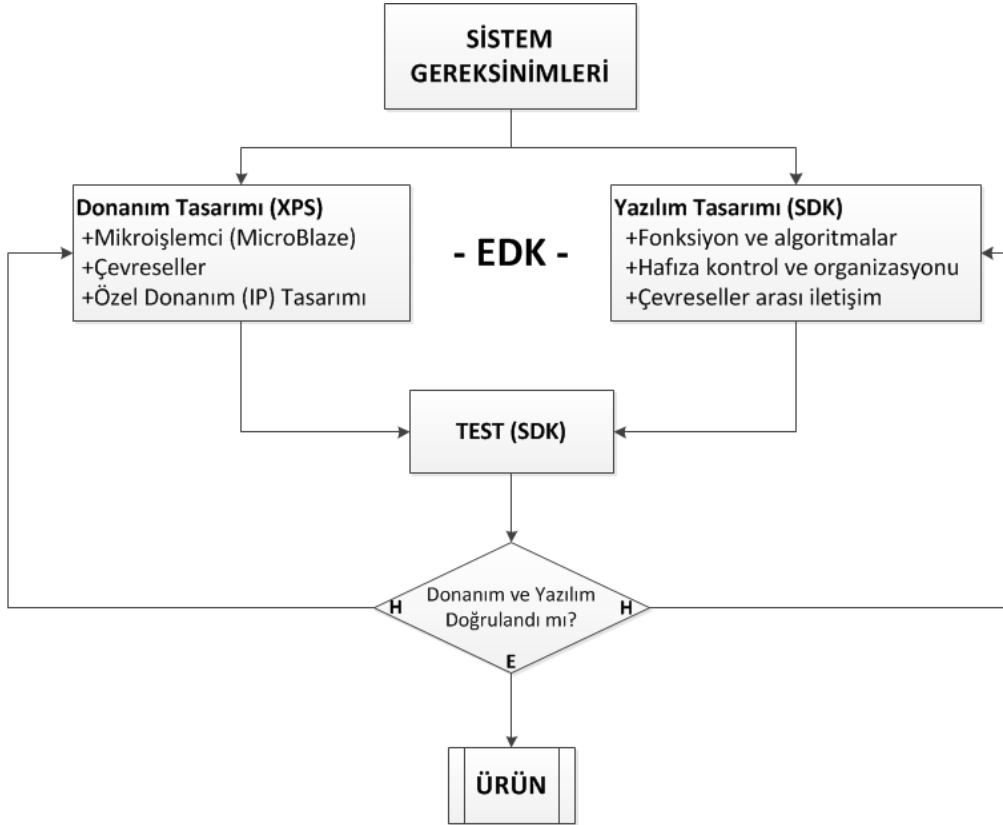
#### 4. GSM MODEMİN FPGA ÜZERİNDE GERÇEKLENMESİ

Bu bölümde, tasarlanan GSM Modem'in FPGA üzerinde Xilinx firmasının araçlarından faydalanılarak nasıl yazılımsal ve donanımsal olarak tasarlandığına ve gerçekleştirilmesine değinilecektir.

##### 4.1 Xilinx EDK

Xilinx Gömülü Sistem Geliştirme Aracı (Embedded Development Kit- EDK), Xilinx firması tarafından geliştirilmiş olup gömülü sistem tasarımı için tümleştirilmiş bir bilgisayar programıdır. Bu program ile gömülü sistem tasarımında gerekli olan yazılım ve donanım tasarımları bir arada yapılabilmektedir.

Günümüzde gömülü sistem tasarımlarında daha önce bahsedildiği gibi, gereksinimler üzerine sistem donanımı (Mikrokontrolör, DSP, FPGA vb.) belirlenip bu donanım üzerine yazılım geliştirilmesi yapılmaktadır.



Şekil 4.1 : Xilinx araçlarının gömülü sistem tasarım aşamalarında kullanımı.

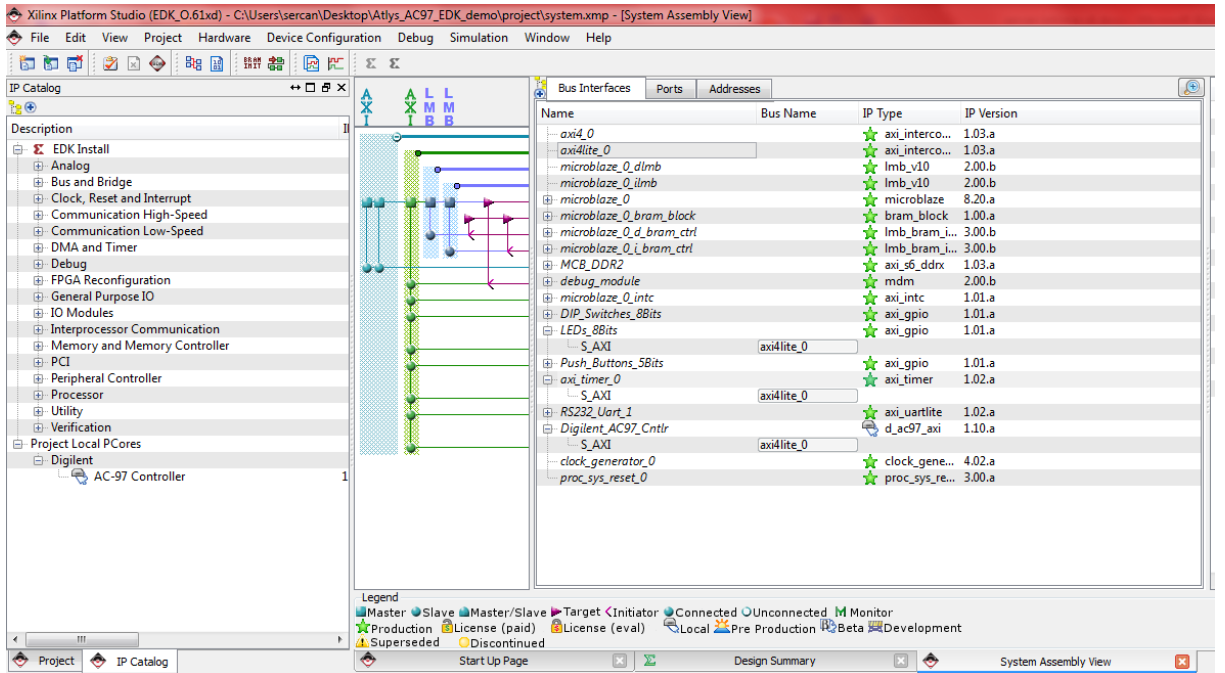
Xilinx EDK, bir FPGA üzerinde hem donanım hem de yazılım geliştirilmesine olanak tanımaktadır. Xilinx tasarım araçları ile yapılan bir gömülü sistem tasarımı şekil 4.1’de verilmiştir.

Bu şekilde, gömülü sistem tasarımında kat edilen aşamalar ve bu noktalarda hangi araçların kullanılabileceği gösterilmiştir. Xilinx EDK’nın sağladığı bir başka fayda, Şekil 4.1’de gösterilen test aşamasında donanımın yetersiz bulunması durumunda, esnek ve kolay biçimde donanım güncellemesine olanak sağlamasıdır. Standart bir tasarımda, test aşamasında donanım tarafında karşılaşılan kritik bir problemde, işlemci yetersiz bulunur ise, güncelleme yapmanın çok daha zor olacağı ortadadır. Hem yeni bir işlemci(DSP,Mikrokontrolör) seçilmesi, hem de elektronik kartın yeniden çizilip bastırılması gerekli olacak, süreç oldukça uzayacaktır.

Bu yönüyle Xilinx EDK, prototip aşamasındaki projelerde, donanım ve yazılım ortak tasarımı sağlaması ve özellikle esnek biçimde donanımsal güncellemeye olanak tanınması sebebiyle tasarım ve test süreçleri için oldukça elverişli bir geliştirme ortamıdır.

### 4.1.1 Xilinx Platform Studio

Xilinx Platform Studio (XPS), mikroişlemci mimarisinin oluşturulduğu tasarım aracıdır. Bu araç ile, sistemin sahip olacağı çevreseller, işlemci ve özel tasarım donanımlar oluşturulur. Programın ekran görüntüsü şekil 4.2’de verilmiştir.



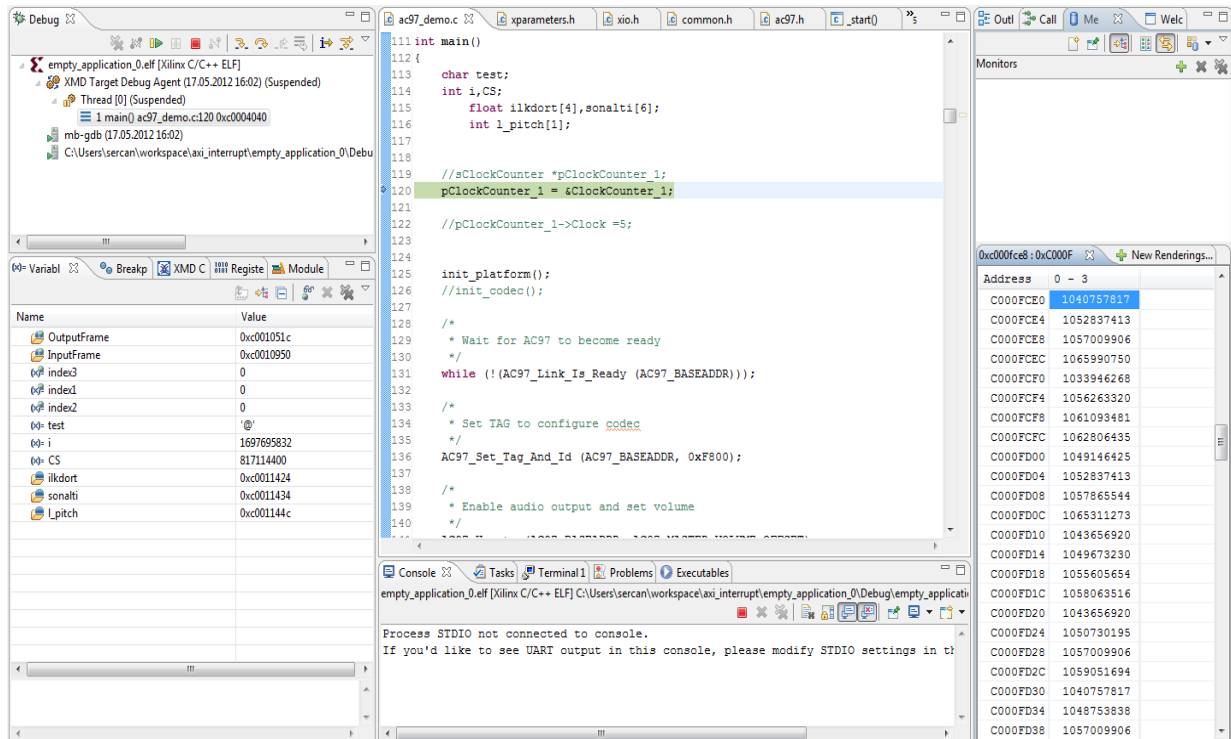
Şekil 4.2 : XPS ekran görüntüsü.



Şekil 4.2’de programın sol tarafındaki bölüm Özel Tasarım Donanım (Intellectual Property-IP) kataloğunu göstermektedir ve bu bölümden işlemcide olması arzu edilen; zamanlayıcı (timer), seri haberleşme (UART, USB vb.) gibi temel çevreseller ve de kullanıcıya ait donanımlar bulunmaktadır. Tez kapsamında mimariye eklenen donanımlar ilerleyen bölümlerde açıklanacaktır.

#### 4.1.2 Software Development Kit

Software Development Kit (SDK) programı, XPS ile tasarlanan donanım mimarisi üzerinde yazılım geliştirilmesine imkan tanımaktadır. Geliştirme dili olarak C kullanılmıştır. SDK hem yazılım geliştirme hem de test ortamı olarak kullanılmıştır.



Şekil 4.2 : SDK programına ait ekran görüntüsü.

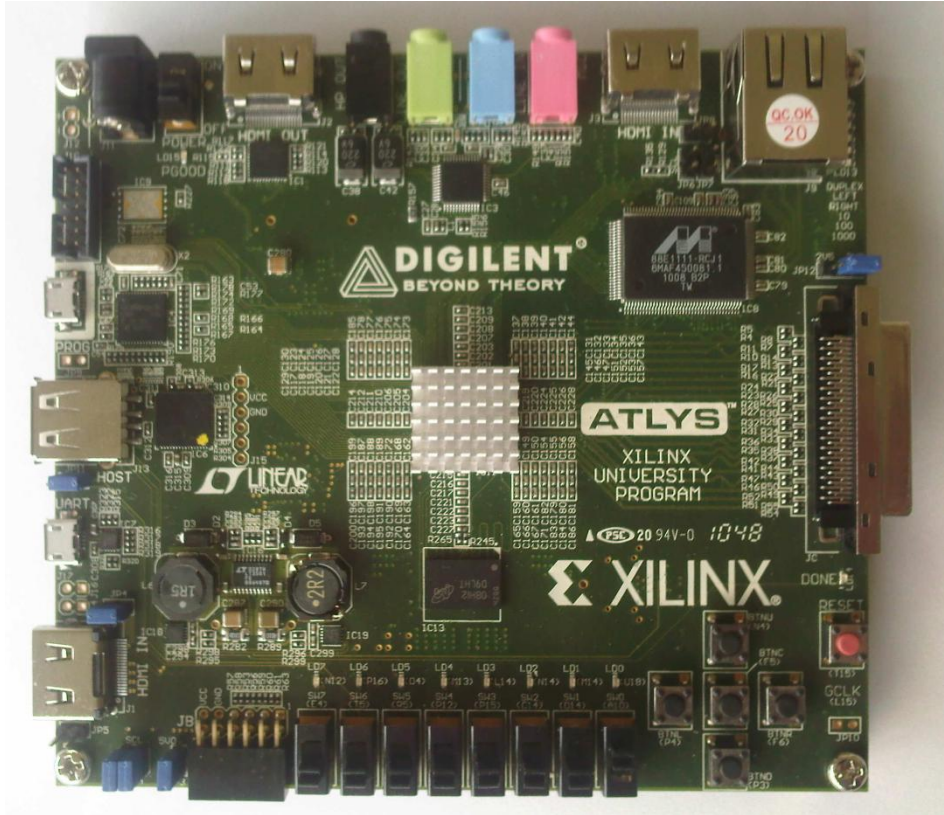
Yukarıda şekil 4.3’de ekran görüntüsü verilen program vasıtasıyla, tasarlanan mimarideki donanımların yazılım tarafında kullanılabilmesi için otomatik olarak birçok kütüphane derlenmekte ve bu sayede donanımların kullanımı oldukça kolaylaşmaktadır. SDK ile ayrıca geliştirilen yazılımların çözümü (Debug) de yapılabilmektedir.

## 4.2 MicroBlaze

MicroBlaze, Xilinx firmasının geliştirdiği 32-bit İndirgenmiş Komut Setli Hesaplama (Reduced Instruction Set Computing-RISC) yapısına sahip sanal bir işlemci mimarisidir. Burada sanal'dan kasıt, mimarinin fabrikasyon ile fiziksel olarak üretilmemiş olup Kütük Transfer Lojigi (Register Transfer Logic-RTL) bazındaki tasarımının FPGA üzerinde gerçekleştirilebilir olmasıdır. MicroBlaze'in olumsuz bir yanı, yalnızca Xilinx FPGA'ları için özelleştirilmiş olup her FPGA için derlenememesidir. Bu tez kapsamında, yazılım geliştirmelerinde 100MHz'de MicroBlaze işlemci mimarisi kullanılmıştır.

## 4.3 Donanım Tasarımı

Bu bölümde, tez kapsamında kullanılan fiziksel ve sanal donanımların tanıtımı yapılacaktır. Test ve gerçekleştirme düzeneği olarak kullanılan Digilent – ATLYS geliştirme kartı şekil 4.3'de gösterilmiştir. Gerçeklemler sırasında kart üzerinde, SPARTAN6 FPGA'sı, LM4550 ses kodlayıcısı, tuşlar ve RAM, kullanılan fiziksel donanımlardır.

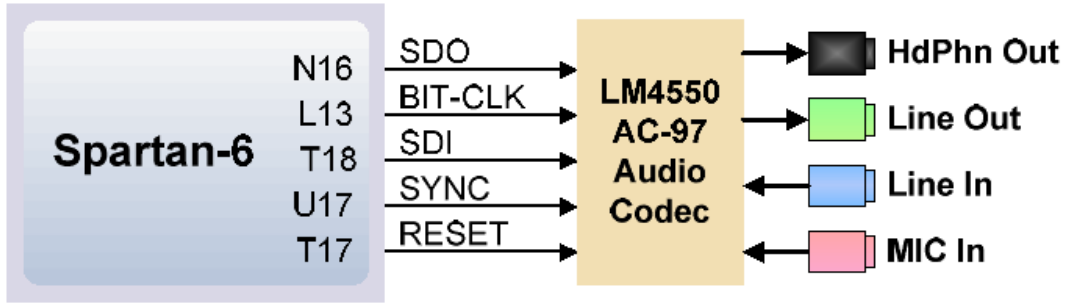


Şekil 4.3 : ATLYS geliştirme kartı.

Bu fiziksel donanımların IP'leri, ses kodlayıcısı için tasarlanan İlk Giren İlk Çıkar (First In First Out-FIFO) bellek ve zamanlayıcı çevresine ait IP'ler sistemde kullanılan sanal donanımlar olmuştur.

#### 4.3.1 Fiziksel donanımlar

Gerçeklemede kullanılan fiziksel donanımlardan LM4550, AC'97 mimarisine sahip bir ses kodlayıcısıdır ve 1Hz çözünürlükte, 4KHz ila 48KHz arasında Analog – Dijital Çevirimi (Analog to Digital Conversion-ADC) ve Dijital - Analog Çevirimi (Digital to Analog Conversion-DAC) yapabilmektedir [17]. Kart üzerinde FPGA ile kodlayıcı arasındaki bağlantı şeması şekil 4.4'de verilmiştir.

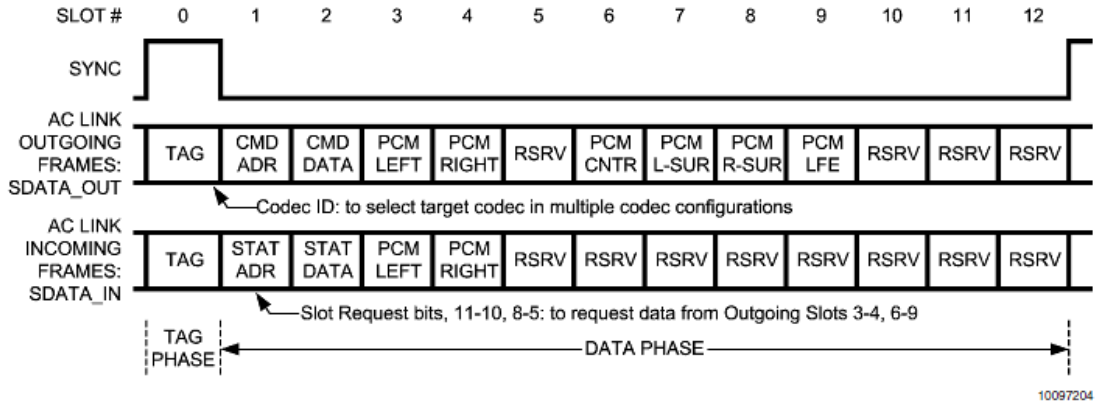


Şekil 4.4 : Ses giriş- çıkışları, ses kodlayıcısı ve FPGA bağlantıları [18].

Gerçeklemeler sırasında kart üzerinden Şekil 4.4'te gösterilen bağlantılar kullanılarak ses çıkışı verilir ve bu çıkış tekrardan örneklenebilir, böylece alıcı ve verici kısmı modellenmiştir. Hat çıkışı (Line Out) ile hat girişi (Line In) birbirine bağlanmış, hat çıkış düğümünden bir paralel daha alınarak, üretilen ve analiz sonrası yeniden sentezlenen seslerin hoparlör vasıtasıyla dinlenebilmesi sağlanmıştır.

Sayısal ve analog dünyalar arasında köprü görevini üstlenen LM4550 entegresi, uygulamada sentetik ses sentezi ve bu sesin yeniden örneklenmesinde kullanılmıştır. Daha önce belirtildiği gibi, kodlayıcı 8KHz hızında ADC ve DAC işlemlerini yerine getirecek biçimde ayarlanmıştır.

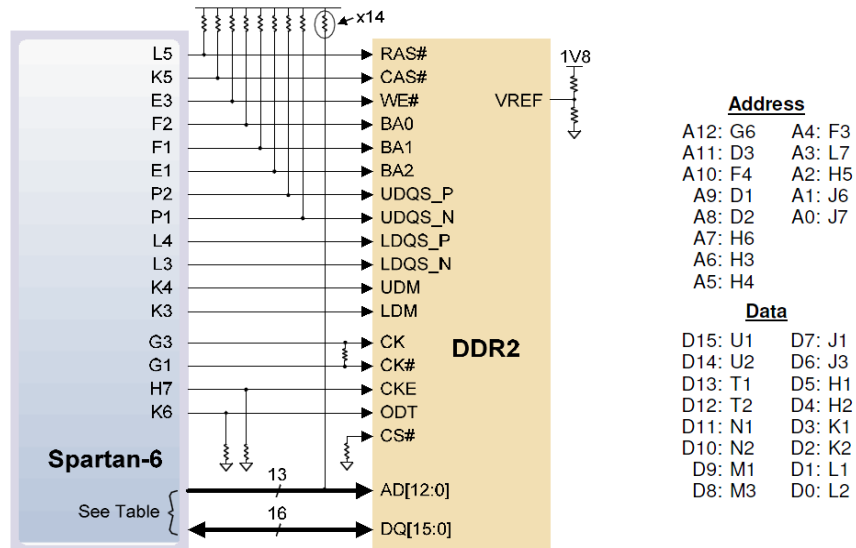
LM4550, ses örneklerini PCM formatında alıp AC-Link haberleşme protokolünde FPGA'ya göndermektedir. Uygulamada kodlayıcı efendi (master) olarak kullanılmıştır ve kendisi 24.576MHz'de saat işaretini üretir. Bir veri paketi 12 bölümden ve toplamda 256bit'den oluşmaktadır. Bu durumda paket gönderim hızının 48KHz, yani en yüksek örnekleme hızına eşit olduğu görülmektedir [17].



Şekil 4.5 : AC-Link protokolü [17].

AC-Link protokolü şekil 4.5’de verilmiştir. LM455 entegresine her pakette, arzu edilen konfigürasyon bilgileri ile, sağ ve sol (stero) ses çıkışları için PCM verileri gönderilip entegreden arzu edilen durum bilgileri ile, sağ ve sol giriş sinyal örneklerine ait PCM verileri alınmaktadır.

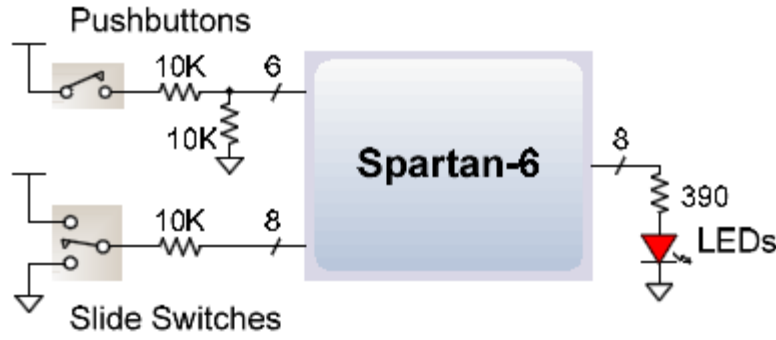
LM455 tarafından örneklenen ses verileri 18 bit ile ifade edilmektedir. Yapılan çalışmalarda, örnekler 10 bite kadar kırıldığında, günlük konuşma tonları için seste bir bozulma gözlemlenmemiş, 8 bite yapılan kırılmalarda parazitlenmeyle karşılaşılmasına rağmen sesin ana formunda bozukluk görülmemiş, alçak geçiren bir filtre ile parazitliğin giderilebildiği görülmüş ve kodlamanın 8 bit ile yapılmasına karar verilmiştir.



Şekil 4.6 : FPGA ve DDR2 Ram bağlantı şeması [18].

Gerçeklemede kullanılan bir diğer donanım DDR2(Double Data Rate, Çiftte Aktarım Hızlı) Ram’dir. Kart üzerinde 800MHz’e kadar FPGA ve Ram arasında haberleşme

sağlanması mümkündür [18]. Tasarımda haberleşme hızı 600MHz olarak seçilmiştir. FPGA ve Ram arasındaki bağlantı hatları şekil 4.5'te verilmiştir.



**Şekil 4.7 :** ATLYS kartı üzerindeki tuş ve anahtarların bağlantı şeması [18].

Kart üzerindeki 5 adet tuş, testler esnasında farklı senaryoların başlatılması için donanımsal kesme özelliği ile birlikte kullanılmıştır. Tuşların bağlantı şeması şekil 4.7'de verildiği gibi olduğundan ötürü, kesmelerin yükselen kenarda tetiklenmesi sağlanmıştır.

#### 4.3.2 IP Çekirdekleri

Tasarımda mikroişlemciye 5 temel IP çekirdeği eklenmiştir. Bunlardan LM455, Ram ve tuşlar için olanlar dış dünya ile bağlantı kurarken, zamanlayıcı ve FIFO IP'leri ise mikroişlemci içerisinde işlevseldir.

LM455 entegresi ile AC-Link hattı üzerinden haberleşme sağlayan IP, daha önce bahsedildiği gibi gelen ses verilerini veya durum bilgilerini protokol içerisinden ayıklayıp yazılım tarafından bu bilgilerin kolaylıkla iç kütüklerden okunmasını sağlar. Benzer şekilde, entegreye gönderilecek ses çıkışları veya konfigürasyonlarını yazılımdan kendi iç kütüklerine alarak, protokole uygun biçimde AC-Link hattına göndermektedir. Bu donanım sayesinde, sistemin dış dünya ile ana bağlantısı olan ses giriş ve çıkışları için mikroişlemcinin, donanımın kütüklerine veri yazıp okumak dışında hiçbir ek iş yapmaması sağlanmış, paralelleştirme sayesinde sistemin yükü dağıtılmıştır.

AC-Link hattındaki giriş ve çıkış ses verilerinin, bir çerçevenin işlenmesi için geçecek süre zarfında saklanması gereklidir. Gelen verileri mikroişlemci tarafından yazılımla LM455 IP'sinden okuyup Ram'e yazarak saklamak mümkündür. Ancak bu işlemin de

hızlı gerçekleşebilmesi için bir FIFO bellek tasarlanarak mikroişlemci mimarisine eklenmiştir.

Bir adet zamanlayıcı (timer)IP'si kesme özelliği ile birlikte sisteme eklenmiş böylece yazılım tarafında periyodik olarak işlenmesi gereken fonksiyonlar ve süre hesabı gerektiren işlemler için altyapı hazırlanmıştır.

DDR2 Ram'ler ile haberleşme ve kontrol adımları, statik Ram'lere nazaran oldukça karmaşıktır. DDR2 Ram'in yazılım tarafından statik bir Ram'miş gibi görülebilmesi için, yalnızca kendi iç kütüklerine okunacak veya yazılacak veri ile adresi alıp gerekli işlemleri kendisi yapan DDR2 kontrol IP'si sisteme eklenmiştir.

Tuşlardan gelen sinyal değişimlerinin algılanması ve kesme üretilip mikroişlemciye bildirilmesi için bir adet tuş IP'si kullanılmıştır.

#### 4.4 Donanım – Yazılım Haberleşmesi

Bu bölümde mikroişlemci mimarisine eklenen IP'lerin yazılım tarafında nasıl erişildiğine değinilecektir. XPS'de yapılan tasarım tamamlanıp yazılım geliştirme amacıyla SDK'ya aktarıldığında, otomatik olarak her donanıma XPS'de atanan adresler kullanılarak erişim sağlanabilmesi için öncül (header) dosyalar oluşturulur. Şekil 4.8'da mimariye bağlı olan donanımların adresleri gösterilmiştir.

microblaze_0's Address Map					
microblaze_0_d_bram_ctrl	C_BASEADDR	0x00000000	0x00003FFF	16K	SLMB
microblaze_0_i_bram_ctrl	C_BASEADDR	0x00000000	0x00003FFF	16K	SLMB
Push_Buttons_5Bits	C_BASEADDR	0x40000000	0x4000FFFF	64K	S_AXI
LEDs_8Bits	C_BASEADDR	0x40020000	0x4002FFFF	64K	S_AXI
DIP_Switches_8Bits	C_BASEADDR	0x40040000	0x4004FFFF	64K	S_AXI
RS232_Uart_1	C_BASEADDR	0x40600000	0x4060FFFF	64K	S_AXI
microblaze_0_intc	C_BASEADDR	0x41200000	0x4120FFFF	64K	S_AXI
axi_timer_0	C_BASEADDR	0x41C00000	0x41C0FFFF	64K	S_AXI
debug_module	C_BASEADDR	0x74800000	0x7480FFFF	64K	S_AXI
Digilent_AC97_Cntlr	C_BASEADDR	0x7D000000	0x7D00FFFF	64K	S_AXI
MCB_DDR2	C_S0_AXI_BASE...	0xC0000000	0xC7FFFFFF	128M	S0_AXI

Şekil 4.8 : Mimaride bulunan donanımların adres ve genişlikleri.

Bu donanımlardan, örneğin ledlerin yazılım tarafında nasıl kullanıldığını incelemek için ilk önce oluşturulan öncül tanımlamalara göz atmak gerekir. Tüm donanımların tanımlandığı 'xparameters.h' dosyası içerisinde ledler için bu tanımlamalar aşağıda Şekil 4.9'da verildiği biçimde yapılmıştır.



```

#define XPAR_LEDS_8BITS_BASEADDR 0x40020000
#define XPAR_LEDS_8BITS_HIGHADDR 0x4002FFFF
#define XPAR_LEDS_8BITS_DEVICE_ID 1
#define XPAR_LEDS_8BITS_INTERRUPT_PRESENT 0

```

**Şekil 4.9 :** Ledler için oluşturulan öncül tanımlamalar.

Burada 4.9 ve 4.8 karşılaştırıldığında XPS’de led donanımı için tanımlanan adres aralığının SDK’da otomatik olarak tanımlandığı, böylece kullanıcıya kolaylık sağladığı görülmektedir.

Mikroişlemcinin en temel görevi, bir adresten veri okuyup bir başka adrese bu verinin farklı işlemlerden geçirilmiş halini yazmaktır. MicroBlaze sayesinde tasarlanan her IP’ye baz adresi sayesinde erişilip iç kütüklerine veri yazılıp okunabilir. Bu işlemi gerçekleştiren fonksiyon ve ledlerin yakılmasına örnek bir kullanım Şekil 4.10’da gösterilmiştir.

```

#define Xil_Out32(Addr, Value) \
    (*(volatile u32 *) ((Addr)) = (Value))

Xil_Out32(LED_BASEADDR, bitRecLED); //Kayıt ledini yak

```

**Şekil 4.10 :** MicroBlaze ile IP’lere erişim(üst) ve Led IP’sine erişim(alt).

Şekil 4.10’da verilen led yakma örneğinden daha karmaşık donanımlarda, örneğin LM455 entegresindeki AC97 mimarisine erişip daha önce açıklanan AC-Link haberleşme hattına veri gönderip almak için, baz adresinden iç kütükler arası öteleme ile erişim için yapılan ön tanımlama ve kullanıma örnek Şekil 4.11’de verilmiştir.

```

#define AC97_ADDR_OFFSET                0x00
#define AC97_CONFIG_DATA_OFFSET        0x04
#define AC97_PCM_OUT_L_OFFSET          0x08
#define AC97_PCM_OUT_R_OFFSET          0x0C
#define AC97_PCM_IN_L_OFFSET           0x10
#define AC97_PCM_IN_R_OFFSET           0x14
#define AC97_TAG_AND_ID_OFFSET         0x18
#define AC97_CONTROL_OFFSET            0x18
#define AC97_STATUS_OFFSET             0x1C

Sample_R = XIo_In32(AC97_BASEADDR + AC97_PCM_IN_R_OFFSET);

```

**Şekil 4.11 :** AC-Link protokolündeki bölümlerin bir kısmı(üst) ile IP tarafından çözümlenen sağ ses giriş örneğinin okunmasına bir örnek(alt).

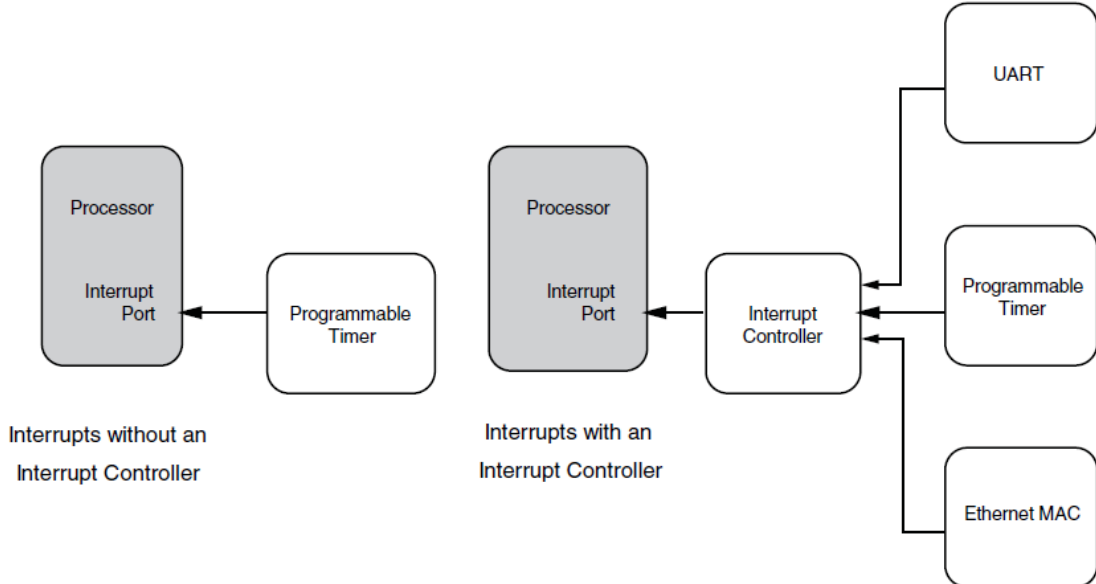
MicroBlaze’in sunduğu bu imkan ile, özel olarak tasarlanan IP’leri yazılım ile kolaylıkla haberleştirilebilmek, bir kart üzerinde ayrık işlemci veya kontrolör ile

FPGA'nın haberleştirilmesi gibi görece zaman alıcı ve zor bir sürece ihtiyaç kalmamaktadır.

#### 4.5 MicroBlaze ve Kesmeler

Kesmeler, gömülü sistemlerde kullanılan en önemli özelliklerden biridir. Kesme, belirlenen bir durum için (örneğin belirli bir sürenin periyodik olarak dolması veya bir sinyalin yükselen kenarı) mikroişlemcinin işletmekte olduğu 'main' fonksiyonunu bırakıp kesme vektörünün durumuna göre ilgili işlemlerin yerine getirilebilmesini sağlar. Örneğin bir tuş takımından yapılacak kontrol veya seri haberleşme portundan gelecek veri için kesme kullanılarak, işlemcinin sürekli bu sinyalleri incelemesi ve beklemesine gerek kalmaz. Kesmeler sayesinde yalnızca beklenen işlem gerçekleştiğinde mikroişlemci gerekli fonksiyonların işletilmesi için yönlendirilmiş olur.

MicroBlaze' de birden kesme kontrolü ve işlemciye bildirim için bir adet kesme kontrol IP'si bulunmaktadır [19]. Bu IP'nin en önemli görevi birden çok kesme oluşması durumunda, yalnızca bir bitlik kesme bildirim portu bulunan işlemcinin, belirlenen önem sıralamasına göre tüm kesmeleri sırasıyla işleyebilmesini sağlamaktır. Kesme kontrolörünün işlevselliği Şekil 4.12 ile gösterilmiştir.



Şekil 4.12 : Kesme kontrol donanımının fonksiyonu [19].



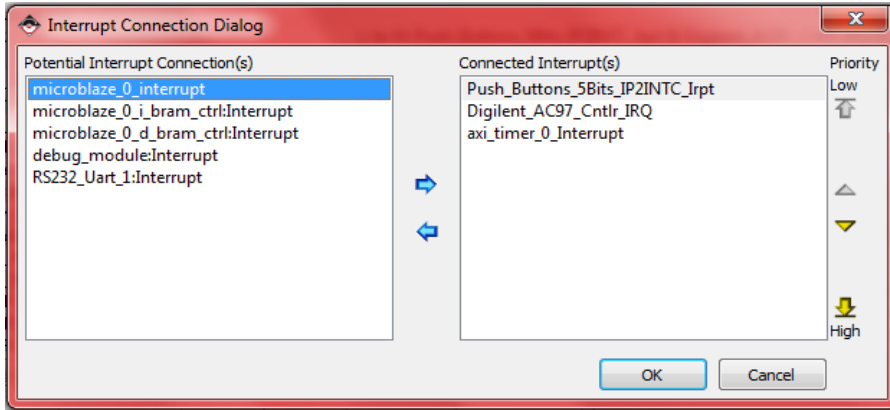
#### 4.5.1 Donanıma Kesme Özelliği Eklenmesi

Tasarlanan bir donanıma kesme özelliği eklenebilmesi için hem XPS’de hem de SDK’da ayarlamaların yapılması gerekmektedir. Daha önce bahsedildiği gibi her donanıma ait adres ve port bilgileri XPS’de mevcut ve konfigüre edilebilir haldedir. Bir donanım tasarlanırken öncelikle kesme özelliğine de sahip olacağı seçilir. Ardından donanım portları bölgesinde Şekil 4.13’de tuşlar için örnek verildiği gibi, donanımın kesme pini kesme kontrolörüne bağlanmalıdır.

Push_Buttons_5Bits	
IP2INTC_Irpt	Push_Buttons_5Bits_IP2INTC_Irpt
(BUS_IF) S_AXI	Connected to BUS axi4lite_0
(IO_IF) gpio_0	Connected to External Ports

Şekil 4.13 : Donanımın kesme kontrolörüne bağlanması.

Bu işlemin ardından kesme kontrolörünün port ayarlarında, izlenecek kesme olaylarına ait pinler (zamanlayıcı, tuşlar, haberleşme vb.) arzu edilen öncelik derecesine göre Şekil 4.15’deki gibi sıralanır.



Şekil 4.14 : Kesme bağlantıları ve öncelik sıralamasının yapılması.

İşlemler tamamlandığında, mimari konfigürasyon dosyasında Şekil 4.15’de kesme kontrolör IP’si için verilen satırlar oluşmaktadır. Tuş, AC97 ve zamanlayıcı IP’lerinin kesme pinlerinin, kesme kontrolörüne ‘INTR’ portu ile bağlandığı görülebilir.

```
BEGIN axi_intc
PARAMETER INSTANCE = microblaze_0_intc
PARAMETER HW_VER = 1.01.a
PARAMETER C_BASEADDR = 0x41200000
PARAMETER C_HIGHADDR = 0x4120ffff
BUS_INTERFACE S_AXI = axi4lite_0
PORT IRQ = microblaze_0_interrupt
PORT S_AXI_ACLK = clk_100_0000MHzPLL0
PORT INTR = Push_Buttons_5Bits_IP2INTC_Irpt & Digilent_AC97_Cntlr_IRQ & axi_timer_0_Interrupt
END
```

Şekil 4.15 : Kesme kontrolörüne ait port bağlantıları.

## 4.5.2 Yazılım Rutinlerinde Kesme İdaresi

Kesme talebinde bulunan bir donanım öncelik sıralamasındaki yeri geldiğinde kesme portu üzerinden Microblaze'e bildirilir. Mikroişlemcide bir adet kesme vektörü bulunduğundan, kesme talebinin hangi donanımdan geldiğinin yazılımsal olarak çözümlenmesi gerekir. Bu çözümlenmeyi, Şekil 4.16'daki komut dizisinden *microblaze\_register\_handler(My\_Handler,INTC\_DEVICE\_ID)* ile verilen fonksiyon gerçekleştirir.

```
//MicroBlaze Yazılımsal Kesme Yapılandırması

/* AŞAMA-1 : Ön Yapılandırma
 * Tuşlar ve Zamanlayıcı kesmelerinin sürücü(driver) fonksiyonlarını ata.
 * Bu atama, SDK tarafından sağlanan kesme kontrol fonksiyonlarının bu donanımlar için
 * kullanılabilmesini sağlar
 */
XIntc_Initialize(&intCtrl, INTC_DEVICE_ID); //XPS'de, kesme kontrol donanımını 'intCtrl' yapısına ata
XIntc_SetOptions(&intCtrl, XIN_SVC_ALL_ISRS_OPTION); // Tüm kesmelerin işlenmesini sağla

/* AŞAMA-2 : Kesme Servislerini Ata
 * Hangi kesme sonucunda hangi fonksiyonun işletilmek üzere çağırılacağını ata.
 */
XIntc_Connect(&intCtrl, BTNS_IRPT_ID, (XInterruptHandler) PushBtnHandler, &pshBtns);
XIntc_Connect(&intCtrl, XPAR_MICROBLAZE_0_INTC_AXI_TIMER_0_INTERRUPT_INTR, (XInterruptHandler)my_timer_handler, (void *)0);

/* AŞAMA-3 : Etkinleştirme.
 * Kesmelerin, kesme kontrolörü tarafından işletilmesini etkinleştir.
 */
XIntc_Enable(&intCtrl, BTNS_IRPT_ID);
XIntc_Enable(&intCtrl, XPAR_MICROBLAZE_0_INTC_AXI_TIMER_0_INTERRUPT_INTR);

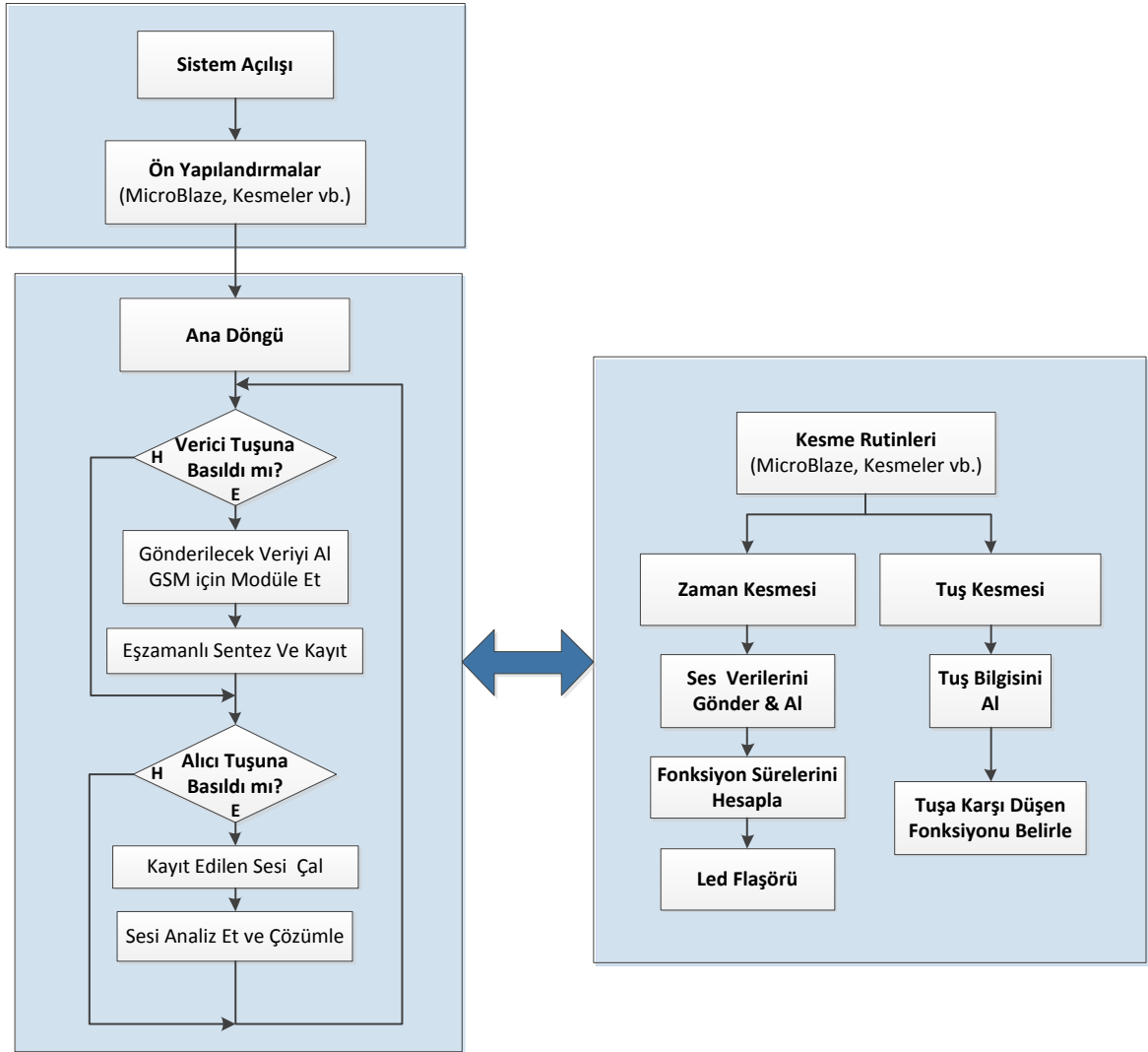
/* AŞAMA-4 : Başlatma.
 * Kesme gözleme ve işletilmesini kontrolörde ve MicroBlaze işlemcisinde başlat.
 */
XIntc_Start(&intCtrl, XIN_REAL_MODE);
microblaze_register_handler(My_Handler, INTC_DEVICE_ID);
microblaze_enable_interrupts();
```

Şekil 4.16 : MicroBlaze yazılımsal kesme yapılandırması.

Kesme kontrolörü öncelikle yapılandırılmalı, ardından hangi kesme için hangi fonksiyonların işletileceği eşleştirilmelidir. Daha sonra kesmelerde arzu edilen özellikler (örneğin zamanlayıcı için periyot veya tuşlar için düşen veya yükselen kenar kesmesi) seçilir. Son adım olarak kesme kontrolöründe ve mikroişlemcide kesme gözlemi ve işletilmesi etkinleştirilir.

## 4.6 Yazılım Tasarımı

İşletim sistemine sahip olmayan mikroişlemci sistemlerinde, görev sistemi uygulanır. Özellikle zamanlayıcıların(timer) ve diğer çevresellerin kullanımı karmaşık sistemlerin tasarımını oldukça kolaylaştırır. Bu tez kapsamında daha önce değinildiği gibi tuşlardan ve zamanlayıcı donanımından gelen uyarılar ile işlemci kesmeye girmektedir. Yazılımdaki ana döngü ve kesme rutinlerine ilişkin akış diyagramları Şekil 4.17 ile verilmiştir.



Şekil 4.17 : Yazılım akış diyagramı.

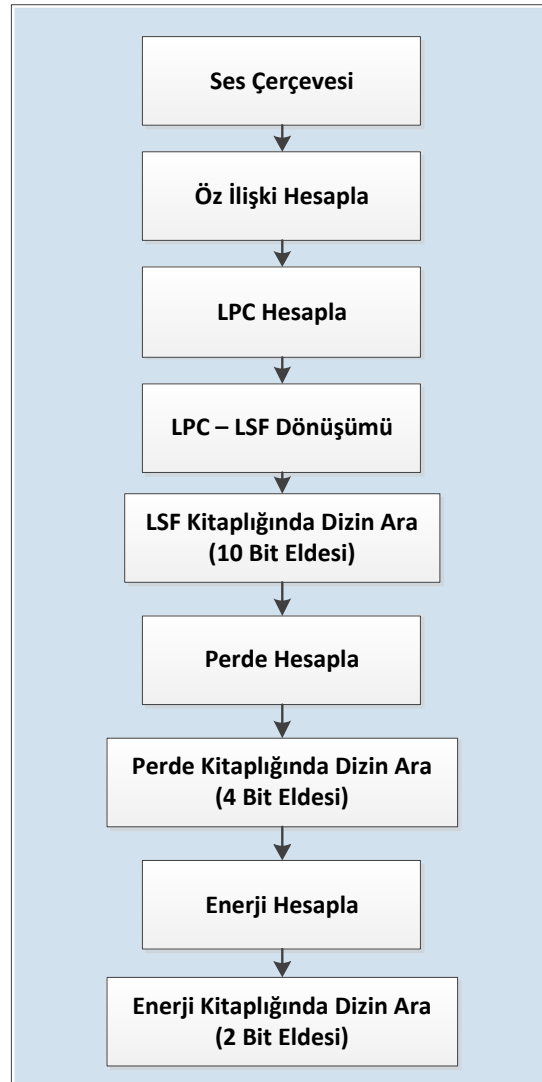
### 4.6.1 Program Ana Rutinleri

Main fonksiyonu içerisinde tez kapsamında tasarlanan iki ana test senaryosu mevcuttur. Bu senaryolar, tuşlardan gelen kesme uyarılarına göre çalışmaya başlar.

İlk senaryo modem verici olarak kullanılmasıdır. Gönderilmesi arzu edilen 16 bitlik dizi sisteme girilerek, bu bite ilişkin modem sentetik bir ses çerçevesi sentezlemesi ve bunu hoparlörden çalması sağlanır. 20ms olan çerçeve süresi anlaşılır bir ses duyulamayacak kadar kısadır bu sebeple 20 defa üst üste çalınarak bir saniyelik bir ses çıkışı sağlanmıştır. Tek FPGA kullanılmasıyla, ses çıkışıyla beraber paralel olarak örnekleme de yapılır.

Örneklenen bu ses modem alıcı olarak kullanılmasını başlatan tuşa basılana kadar hafızada tutulur. Test başlatılırsa hafızadaki ses analiz edilerek sırasıyla bölüm 3.7’de açıklanan süreçlerden Şekil 4.18’de gösterildiği gibi geçerek, gönderilen bit dizisi geri elde edilir.

### Sesi Analiz Et ve Çözümle



Şekil 4.18 : Modemin alıcı olarak kullanılması.

#### 4.6.2 Kesme Rutinleri

Zamanlayıcı kesmesi, AC97 kodlayıcısının periyodik ve eşzamanlı olarak ses çıkışı üretip , kendi çıkışlarını örnekleme sağlanmıştır. Test senaryolarının işletildiğini kullanıcıya bildirmek amacıyla kart üzerinde bulunan ledler de algılanabilir biçimde yakıp söndürülür.

SDK'da saat (clock) sayıcı bulunmadığından, yazılımda işletilen analiz ve sentez fonksiyonlarının işlenmesi için ne kadar zaman harcadığının belirlenmesinde zaman kesmesi sayaç olarak kullanılmıştır. Bu sürelerle ilişkin detaylı bilgi bir sonraki bölümde verilecektir.

Kesme rutinleri, daha önce bahsedildiği gibi işlemcinin sürekli kontrolüne gerek kalmaması ve yapı itibariyle kesme kontrolüne çok uygun olması sebebiyle, tuşların bağlı olduğu bacalarda sinyalin yükselen kenarında(tuşa basılması durumunda) tetiklenecek biçimde ayarlanmıştır. Tuş kesmesine girilmesi durumunda, öncelikle hangi tuşa basıldığı belirlenip, ardından bu tuşun hangi fonksiyona karşılık geldiği (bu uygulama için modemin alıcı veya verici olarak çalıştırılması) işletim kütüğüne kayıt edilir. İşletim kütüğü Şekil 4.17'de gösterildiği gibi ana döngüde kontrol edilerek test senaryoları başlatılır.

## 5. GERÇEK ZAMAN TESTLERİ

Bu bölümde, modemın alıcı ve verici olarak kullanılmasında gerekli olan fonksiyonların işletilmesi için ne kadar zaman geçtiği tespit edilmeye çalışılmıştır. Gerçek zamanlı bir sistemde modem'in göndereceği veya alacağı ses çerçevesini işlemesi için yalnızca 20ms' si vardır. Bu tez kapsamında fonksiyonellik C dilinde test edilip, işlem süresinin 20ms' ye çekilebilmesi için hangi parçaların donanıma geçirilebileceği belirlenmeye çalışılmıştır.

Daha önce bahsedildiği gibi süre hesabında Şekil 5.1'de verildiği gibi zamanlayıcı kesmesinden faydalanılmış ve kesme periyodu 100us olarak seçilmiştir.

```
typedef struct
{
    char ProcessReg;
    int ProcessPeriod[5];
} sClockCounter;
```

(a)

```
//Süreç Hesaplama
if (pClockCounter_1->ProcessReg & Process0_Mask )
    pClockCounter_1->ProcessPeriod[Process0]++;

if (pClockCounter_1->ProcessReg & Process1_Mask )
    pClockCounter_1->ProcessPeriod[Process1]++;

if (pClockCounter_1->ProcessReg & Process2_Mask )
    pClockCounter_1->ProcessPeriod[Process2]++;

if (pClockCounter_1->ProcessReg & Process3_Mask )
    pClockCounter_1->ProcessPeriod[Process3]++;

if (pClockCounter_1->ProcessReg & Process4_Mask )
    pClockCounter_1->ProcessPeriod[Process4]++;
```

(b)

**Şekil 5.1** : Süreç hesaplama için kurulan yapı(a) ve zaman kesmesinde işletilen sayaçlar(b).

Süresi ölçülmek istenen bir fonksiyona, Şekil 5.2'de ses çerçevesinden LPC parametrelerine geçiş için örnek verildiği gibi eklemeler yapılır.

```

//Çerçeve - LPC
pClockCounter_1->ProcessPeriod[Process1] = 0;
pClockCounter_1->ProcessReg = pClockCounter_1->ProcessReg | Process1_Mask;
autocorrelation(speech_frame);
pClockCounter_1->ProcessReg = pClockCounter_1->ProcessReg & (~Process1_Mask);

pClockCounter_1->ProcessPeriod[Process1] = 0;
pClockCounter_1->ProcessReg = pClockCounter_1->ProcessReg | Process2_Mask;
levinson(acf_coeff);
pClockCounter_1->ProcessReg = pClockCounter_1->ProcessReg & (~Process2_Mask);

```

**Şekil 5.2 :** Süreç hesaplama yapılarının kullanımı.

Testler sırasında bölüm 3.6 ve 3.7’de verilen adımlar, Şekil 4.17’de verilen akış diyagramı ile işletilerek Tablo 5.1’de verilen sonuçlar elde edilmiştir.

**Tablo 5.1 :** Fonksiyonların işlem süreleri

Fonksiyon	Süre(ms)
<b>Modemin Alıcı Olarak Kullanımı (16 bit)</b>	<b>980</b>
Ses Çerçevesi - Özilişki Parametreleri	27,6
Özilişki Parametreleri- LPC Katsayıları	1,9
<b>LPC - LSF Dönüşümü</b>	<b>925</b>
P ve Q Polinomlarının Oluşturulması	0,5
P Polinomunun Köklerinin Eldesi	465
Q Polinomunun Köklerinin Eldesi	455
Köklerden LSF katsayılarına Geçiş	2,3
Kod Kitaplığında İlk dört LPC taraması (4 bit eldesi)	0,9
Kod Kitaplığında Son altı LPC taraması (6 bit eldesi)	1,2
<b>Perdenin Eldesi</b>	<b>1,3</b>
Perdenin hesabı	0,9
Perde kod kitaplığında dizin taraması (4 bit eldesi)	0,4
<b>Enerjinin Eldesi (2 bit eldesi)</b>	<b>4,5</b>
<b>Modemin Verici Olarak Kullanımı (16 bit)</b>	<b>28,6</b>
<b>LSF - LPC Dönüşümü ( 10 bit girdisi)</b>	<b>3,7</b>
<b>Ses Sentezi (4 bit girdisi)</b>	<b>24,9</b>
<b>Enerji Sentezi (2 bit girdisi)</b>	<b>0,6</b>

Süre hesaplama testleri esnasında perdenin eldesi için genlik kontrolü kullanılmıştır. Bölüm 3.4’de belirtilen 20 ile 150 örnek öteleme için öz ilişki değerleri hesaplanması durumunda Tablo5.1’den görülebileceği gibi,  $130 \cdot 27,6 = 3588\text{ms}$  harcanması gerekecektir.

Süreç hesaplarında hassasiyet için 100us tutulan kesme periyodunun, fonksiyonların gerçek süreleri üzerinde bozucu etkisi olduğu açıktır. Bu etkiyi gözlemlemek adına,

modemin verici olarak kullanıldığı test, zamanlayıcı periyodu 10ms' ye çıkartılarak yeniden incelenmiş ve sonucun 980ms'den 880ms'ye indiği görülmüştür.

Sistemde en çok zaman harcayan fonksiyonun LSF-LPC dönüşümü ve bu fonksiyon içerisinde de kök bulma alt rutinleri olduğu anlaşılmıştır. LSF-LPC dönüşümünü ikinci en yüksek zaman ile öz ilişki parametrelerinin hesaplanması takip etmektedir. Şekil 5.3'de öz ilişki parametrelerinin hesaplanmasına ilişkin bir program kesiti gösterilmiştir. Bir diğer yüksek işlem süresine sahip ses sentez filtresi de, Şekil 5.3'deki gibi iç içe 'for' döngüleri barındırmaktadır.

```

if(autocorrelation_start){
    for ( p=0; p<predictor+1; p++) {
        for ( i=0; i<(frame_length-p); i++) {
            acf_coef[p] = speech_frame[i] * speech_frame[i+p] + acf_coef[p];
        }
        acf_coeff[p] = acf_coef[p];
    }

    autocorrelation_done = 1;
}

```

**Şekil 5.3 :** Öz ilişki parametreleri fonksiyonundan bir parça.

Yüksek işlem süreleri sebebiyle öz ilişki parametreleri hesabı ve kök bulma işlemleri için sisteme özel donanımlar eklenmesi gerektiği sonucuna varılmıştır. Bu verilerden elde edilebilecek bir diğer sonuç, modemin hangi süreçte bit başına daha çok işlem yaptığıdır.

**Tablo 5.2 :** Fonksiyon sürelerinin bit sentez ve analiz adetlerine oranı.

<b>Fonksiyon</b>	<b>Veri Eldesi (zaman(ms)/bit)</b>
<i>Vokal Yapı Analizi</i>	954,5 / 10 = <b>95,45</b>
<i>Perde Analizi</i>	3588/4 = <b>897</b> (öz ilişki) - 1,3/4 = <b>0,325</b> (genlik)
<i>Enerji Analizi</i>	4,5 / 2 = <b>2,25</b>
<i>Vokal Yapı Sentezi</i>	3,7 / 10 = <b>0,37</b>
<i>Ses Sentezi</i>	24,9 / 4 = <b>6,22</b>
<i>Enerji Sentezi</i>	0,6 / 2 = <b>0,3</b>

Tablo 5.2'de 16'bitlik bir veri paketinin sentez ve analizinde harcanan süreler karşılığında elde edilen bitler gösterilmiştir. Göze çarpan önemli bir sonuç, genlik kontrolünden öz ilişki karşılaştırmasına uzanan çeşitlilikte perde analiz süresinin oldukça değişkenlik gösterdiği. Bu değişkenlikle beraber elbette başarı oranı da değişmektedir. Bir diğer önemli bulgu modemin analizde, sentezde olduğundan yaklaşık 150 kat daha uzun işlem süresine ihtiyaç duyduğu olmuştur.



## 6. SONUÇLAR VE TARTIŞMA

Bu tezde, doğrusal öngörü ile yapay ses sentezi ve GSM hattının ses iletimindeki kriterleri incelenerek, GSM üzerinden çalışabilecek bir modem tasarlanmıştır. Tasarlanan sistem, Digilent firmasına ait ATLYS geliştirme kartı üzerindeki SPARTAN6 FPGA'sı üzerine MicroBlaze mikroişlemcisi kurularak donanımsal ve yazılımsal ortak bir çalışma ile gerçekleştirilmiştir.

Yapılan çalışmalar esnasında, insan konuşmasına ait bir ses doğrusal öngörü metodu ile modellenilip sıkıştırılabildiği gibi, GSM üzerinden iletme müsait ses kitaplığından da sentetik konuşmalar sentezlenebilmiş, böylece yalnızca insan konuşması için özelleşmiş bir sistem üzerinden, konuşmaya benzemeyen resim, ses, kriptolu konuşma ve video gibi genel formatlı verilerin aktarımı da mümkün kılınmıştır.

Çalışma neticesinde, modem fonksiyonları C dilinde algoritmik olarak test edilmiş ve modemin alıcı olarak çalıştığı durumlarda kayda değer bir iş gücüne ihtiyaç duyduğu görülmüştür. Sistemde en büyük iş yükünün LPC parametrelerinin LSF parametrelerine dönüştürülmesinde harcandığı belirlenmiştir. Bu büyük iş yükünün karşılığında model sentez filtrelerinin kararlılığı garantilenmiş ve LSF'ler ile betimlenen vokal model hataya karşı daha dayanımlı olmuştur. Modemin gerçek zamanlı olarak çalışabilmesi için kök alma ve öz ilişki belirleme fonksiyonlarının donanımsal destek ve daha hızlı algoritmalarla gerçekleştirilmesi gerektiği sonucuna varılmıştır.

## KAYNAKLAR

- [1] **LaDue, C.K.; Sapozhnykov, V.V.; Fienberg, K.S.**, "A Data Modem for GSM Voice Channel," *Vehicular Technology, IEEE Transactions on* , vol.57, no.4, pp.2205-2218, July 2008.
- [2] **Yucun Yang; Suili Feng; Wu Ye; Xinsheng Ji**, "A Transmission Scheme for Encrypted Speech over GSM Network," *Computer Science and Computational Technology, 2008. ISCSCT '08. International Symposium on* , vol.2, no., pp.805-808, 20-22 Dec. 2008.
- [3] **N.N. Katugampala, K.T.Al-Naimi, S. Vilette, A. Kondoç**, "Real time data transmission over GSM voice channel for secure voice and data applications," *Secure Mobile Communications Forum: Exploring the Technical Challenges in Secure GSM and WLAN, The 2nd IEE (Ref. No. 2004/10660)* ,Lodon,23 Sept. 2004.
- [4] **N.N. Katugampala, S. Vilette, A. Kondoç**, "Secure voice over GSM and other low bit rate systems," *Secure GSM and Beyond: End to End Security for Mobile Communications, IEE Seminar on (Digest No. 2003/10059)*, Lodon,11 Feb. 2003.
- [5] **A.M.Kondoç**, "Digital Speech:Coding for Low Bit Rate Communication Systems", John Wiley&Sons, Second edition 2004.
- [6] **Özkan A.**, 2011. GSM Ağı Üzerinden Güvenli Ses İletimi, *Lisans Tezi*, İ.T.Ü. Elektrik Elektronik Fakültesi, İstanbul.
- [7] **N. S. Jayant and P. Noll**, "Digital Coding of Waveforms: Principles and Applications to Speech and Video", Prentice-Hall, Englewood Cliffs, New Jersey, 1984.
- [8] **Alan V. Oppenheim and Ronald W. Schafer**, "Digital Signal Processing", Prentice Hall, 1975.
- [9] **Kırçiçek Y.**, 2007. Doğrusal Öngörü İle Konuşma İşareti Kodlayıcısı Tasarımı, *Yüksek Lisans Tezi*, Y.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.

- [10] **Alan V. Oppenheim and Alan S. Willsky**, “Signals and Systems”, Prentice Hall, 1997.
- [11] **Ian McLoughlin**, “Applied Speech and Audio Processing”, Cambridge University Press, New York, 2009.
- [12] **Bradbury J.S.**, 2000. “Linear Predictive Coding”, Teknik Rapor, Queen’s University, Ontario.
- [13] **Gomez, P.**, 2004. Speech Coding & Linear Predictive Coding (LPC), *PhD Thesis*, Florida International University, Miami.
- [14] **Thierry Dutoit and Ferran Marques**, “Applied Signal Processing”, Springer, 2009.
- [15] **Signal Processing Blockset – LPC to LSF/LPS Conversion**, Matlab Tutorial, 2007.
- [16] **Jones D., Appadwedula S., et al.**, “Speech Processing: Theory of LPC analysis and Synthesis”, Connexions Project, 2009.
- [17] **LM4550**, veri yaprağı, 2003, National Semiconductor.
- [18] **Atlys Kartı**, Referans el kitabı, 2011, Digilent.
- [19] **Embedded System Tools Reference Guide - EDK 11.3.1**, 2009, Xilinx.

## **ÖZGEÇMİŞ**

**Ad Soyad:** Sercan Tunçay

**Doğum Yeri ve Tarihi:** İstanbul, 1989

**Adres:** Erenköy Mah. Alpler Sok. Çakınay Apt. No: 10/12 Erenköy/İSTANBUL

**Lisans:** İstanbul Teknik Üniversitesi, Elektronik Mühendisliği; 2008 - 2012