

**YILDIZ TEKNİK ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

139680

**SAYISAL İŞARET İŞLEYİCİYLE  
ŞEBEKE HARMONİKLERİNİN  
ANALİZİ**

139680


Elektronik Müh. Özcan DURAHİM

**FBE Elektronik ve Haberleşme Mühendisliği Anabilim Dalı Elektronik Programında  
Hazırlanan**

**YÜKSEK LİSANS TEZİ**

Tez Danışmanı : Yrd. Doç. Dr. Tuncay UZUN

Prof. Dr. Celal KOCATEPE



Prof. Dr. Vedat Tansuğul



İSTANBUL, 2003

**TC YÜKSEK ÖĞRETİM KURULU  
DOKÜMANTASYON MERKEZİ**

# İÇİNDEKİLER

	Sayfa
SİMGE LİSTESİ.....	iv
KISALTIMA LİSTESİ.....	v
ŞEKİL LİSTESİ.....	vi
ÇİZELGE LİSTESİ.....	viii
ÖNSÖZ.....	ix
ÖZET.....	x
ABSTRACT .....	xi
1. GİRİŞ.....	1
2. HARMONİK ANALİZİ .....	3
2.1 Harmonik Kavramı .....	3
2.2 Harmonik Üreten Kaynaklar .....	4
2.2.1 Harmoniklerin Yük Tarafında Üretilmesi.....	4
2.2.2 Harmoniklerin Şebeke Tarafında Üretilmesi .....	5
2.3 Harmoniklerin Ortaya Çıkardığı Problemler .....	5
2.4 Harmonik Analizi .....	6
2.4.1 Çoklu Konik Spektrum Kestirimi.....	7
2.4.2 Parametrik Olmayan Spektrum Kestirimi.....	7
2.4.3 Parametrik Spektrum Kestirimi.....	7
2.4.4 Belirli Spektrum Analizi .....	7
2.4.4.1 Doğrusal Filtreleme İşlemi.....	9
3. SAYISAL İŞARET İŞLEYİCİ .....	11
3.1 ADSP 2181'in Genel Mimarisi: .....	11
3.1.1 Hesaplayıcı Birimler .....	12
3.1.1.1 Aritmetik Mantık Birimi .....	12
3.1.1.2 Çarpıcı/Toplayıcı.....	14
3.1.1.3 Öteleyici.....	16
3.1.1.4 Öteleyici İşlemleri .....	16
3.1.2 Program Kontrol Birimleri.....	17
3.1.2.1 Program Sıralayıcı .....	17
3.1.3 Veri Transferi .....	18
3.1.3.1 Veri Adres Üreteçleri.....	19
3.2 Sayısal İşaret İşleyicinin Seri Portu.....	19
3.3 Zamanlayıcılar.....	21
3.4 Bellek Arabirimi .....	21
3.4.1 Program Belleği Arabirimi.....	22
3.4.2 Veri Belleği Arabirimi .....	23
3.4.3 Ön Yükleme Belleği Arabirimi .....	24
4. FREKANS SPEKTRUMUNUN HESAPLANMASI .....	25
4.1 Belirli Spektrum Analizi .....	25

4.1.1	FFT Algoritmasının Genel Akış Diyagramı .....	26
4.1.2	Zamanda İndirgeme .....	28
4.1.3	Frekansta İndirgeme .....	30
4.1.4	DIF FFT - DIT FFT tercihi: .....	30
4.1.5	DIF veya DIT Seçildikten Sonra FFT Gerçeklenmesi: .....	30
4.2	Matematiksel İşlemler.....	32
4.2.1	Sabit Nokta Aritmetiği.....	32
4.2.2	Kayan Nokta Aritmetiği.....	33
4.2.3	Taşma Engelleme Yöntemleri.....	33
4.2.4	Giriş Verisini Dengeleme Yöntemi .....	33
4.2.5	Şartsız Blok Kayan Nokta Dengeleme Yöntemi.....	34
4.2.6	Şarh Blok Kayan Nokta Dengeleme Yöntemi .....	34
4.2.7	Bit Tersleme Özelliği.....	34
4.2.8	Blok Kayan Nokta Dengeleme İşlemi .....	35
5.	HARMONİK ANALİZ DONANIM VE YAZILIMI .....	37
5.1	Sistemin Temel Yapısı.....	37
5.2.	Sistemin Sayısal İşaret İşleme Mantığı.....	37
5.2.1	Örneksel-Sayısal Dönüşüm .....	37
5.2.1.1	İşaretin Örneklenmesi .....	38
5.2.1.2	Kuantalama ve Kodlama.....	39
5.3	Yazılım Ve Donanım Geliştirme Aşamaları.....	43
5.3.1	ADSP-2181 İşlemcisinde Program Yazılması ve Derlenmesi.....	43
5.3.1.1	Derleyicinin Çalıştırılması .....	43
5.3.1.2	Birleştiricinin Çalıştırılması .....	43
5.3.1.3	Simülatörün Çalıştırılması .....	44
5.3.1.4	EPROM'a Program Yükleme.....	44
5.3.2	Geliştirme Seti Hakkında Özet Bilgi .....	45
5.3.2.1	AD1847 Kodlayıcı ve kod çözücü tümleşik devresi .....	47
5.3.3	Sistemin Örneklem Frekansı Seçimi.....	48
5.3.4	Örtüşme Sorunu ve Giderilmesi .....	49
5.3.5	Spektrum Sızıntısı Sorunu ve Giderilmesi.....	50
5.3.6	Pencere Fonksiyonu Seçimi .....	52
5.3.7	Devrenin Çalışması.....	53
5.3.9	Harmonik Analiz Programı Akış Diyagramı .....	55
5.4	Harmonik Analiz Arayüz Programı Akış Diyagramı .....	56
5.5	Deneysel Sonuçlar .....	57
5.5.1	Deney-1 .....	57
5.5.2	Deney -2.....	57
5.5.3	Deney -3 .....	58
5.5.4	Deney -4.....	58
5.5.5	Deney -5.....	59
5.5.6	Deney -6.....	59
5.5.7	Sonuçların Frekans Bakımından Değerlendirilmesi.....	60
5.5.8	Sonuçların Genlik Bakımından Değerlendirilmesi .....	61
6.	SONUÇLAR VE ÖNERİLER .....	63
	KAYNAKLAR.....	64
	ÖZGEÇMİŞ.....	65

## SİMGE LİSTESİ

$a$	Kuantalama aralığı
$F$	Frekans
$F_a$	A-eğrisi fonksiyonu
$F_\mu$	$\mu$ -eğrisi fonksiyonu
$f_s$	Örnekleme frekansı
$N$	Örnekleme sayısı
$T$	Periyot
THD	Toplam harmonik bozulması
$X(k)$	Giriş dizisinin spektrumu
$X_s$	Örneklenmiş işaret
$W$	Açısal hız
$w$	Twiddle faktör



## **KISALTIMA LİSTESİ**

<b>ALU</b>	<b>Arithmetic Logic Unit</b>
<b>ASHIFT</b>	<b>Arithmetic Shift</b>
<b>ASTAT</b>	<b>Arithmetic Status</b>
<b>BMS</b>	<b>Boot Memory Select</b>
<b>BR</b>	<b>Bus Request</b>
<b>CLKIN</b>	<b>Clock In</b>
<b>CLKOUT</b>	<b>Clock Out</b>
<b>DAG</b>	<b>Data Address Generator</b>
<b>DFT</b>	<b>Discrete Fourier Transform</b>
<b>DIF</b>	<b>Decimation-In-Frequency</b>
<b>DIT</b>	<b>Decimation-In-Time</b>
<b>DMA</b>	<b>Data Memory Address</b>
<b>DMD</b>	<b>Data Memory Data</b>
<b>DMS</b>	<b>Data Memory Select</b>
<b>DR</b>	<b>Data Recieve</b>
<b>DSP</b>	<b>Digital Signal Processor</b>
<b>DT</b>	<b>Data Transmit</b>
<b>EXP</b>	<b>Exponential</b>
<b>EXPADJ</b>	<b>Exponential Adjustment</b>
<b>FFT</b>	<b>Fast Fourier Transform</b>
<b>GND</b>	<b>Ground</b>
<b>ICNTL</b>	<b>Interrupt Control</b>
<b>IDE</b>	<b>Integrated Development Environment</b>
<b>IEEE</b>	<b>International Electrics Electronics Engineering Society</b>
<b>IRQ</b>	<b>Interrupt Request</b>
<b>LSHIFT</b>	<b>Logic Shift</b>
<b>MAC</b>	<b>Multiplier / Accumulator</b>
<b>MMAP</b>	<b>Memory Map</b>
<b>NORM</b>	<b>Normalization</b>
<b>PAM</b>	<b>Pulse Amplitude Modulation</b>
<b>PMA</b>	<b>Program Memory Address</b>
<b>PMD</b>	<b>Program Memory Data</b>
<b>R</b>	<b>Result</b>
<b>RAM</b>	<b>Random Access Memory</b>
<b>RFS</b>	<b>Receive Frame Synchronization</b>
<b>RTI</b>	<b>Return From Interrupt</b>
<b>SCLK</b>	<b>Serial Clock</b>
<b>SMPS</b>	<b>Swicthed Mode Power Supply</b>
<b>SPORT</b>	<b>Serial Port</b>
<b>TDM</b>	<b>Time Division Multiplexing</b>
<b>TFS</b>	<b>Transmit Frame Synchronization</b>
<b>UPS</b>	<b>Uninterruptible Power Supplies</b>

## ŞEKİL LİSTESİ

	Sayfa
Şekil 2.1	Harmonikli işaret..... 4
Şekil 2.2	Harmonikli işaretin zaman ve frekans düzlemi gösterimi ..... 8
Şekil 2.3	DFT-FFT karşılaştırması ..... 9
Şekil 2.4	Goertzel algoritması ..... 10
Şekil 3.1	ADSP-21XX blok diyagramı ..... 11
Şekil 3.2	1.15 Sayı formatı ..... 12
Şekil 3.3	ALU blok diyagramı..... 13
Şekil 3.4	MAC blok diyagramı..... 15
Şekil 3.5	Öteleyici blok diyagramı ..... 16
Şekil 3.6	Program sıralayıcı blok diyagramı ..... 18
Şekil 3.7	Veri adres üretici blok diyagramı ..... 19
Şekil 3.8	Seri port yapısı ..... 20
Şekil 3.9	Timer blok diyagramı ..... 21
Şekil 3.10	ADSP-2181 sistemi ..... 22
Şekil.3.11	ADSP-2181 program bellek haritası..... 23
Şekil 3.12	Veri belleği yerleşimi ..... 23
Şekil 3.13	EPROM'dan iç program belleğine programın yüklenme sırası..... 24
Şekil 4.1	İşaretlerin türlerine göre Fourier dönüşüm tipleri ..... 25
Şekil 4.2	Örnek bir işaretin Fourier dönüşümü ..... 26
Şekil 4.3	N noktalı DIT FFT karakteristikleri ..... 26
Şekil 4.4	Radix-2 DIT FFT genel akış diyagramı..... 27
Şekil 4.5	8 noktalı DIT FFT ..... 29
Şekil 4.6	Kelebek yapısı ..... 29
Şekil 4.7	8 noktalı DIF FFT ..... 30
Şekil 4.8	16 Bitlik sayının işaretli ve işaretsiz gösterim biçimleri ..... 33
Şekil 4.9	Bit tersleme özelliğine bir örnek ..... 35
Şekil 4.10	Üst ayarlanması ..... 36
Şekil 5.1	Sistemin sayısal işaret işleme mantığı blok diyagramı..... 37
Şekil 5.2	Örneksel-sayısal dönüşümün modellenmesi..... 38
Şekil 5.3	İşaretin darbe katarıyla çarpımı..... 39
Şekil 5.4	Düzgün kuantalama eğrisi..... 40
Şekil 5.5	Sıkıştırma ve genişletme eğrileri..... 41
Şekil 5.6	A ve u tipi sıkıştırma eğrileri ..... 42
Şekil 5.7	Geliştirme setinin blok şeması ..... 45
Şekil 5.8	AD1847 CODEC tümleşik devresi blok diyagramı ..... 47
Şekil 5.9	AD1847 tümleşik devresi veri giriş ve çıkış zamanlama diyagramları..... 48
Şekil 5.10	26 kHz'lik üçgen dalganın frekans düzlemi gösterimi..... 49
Şekil 5.11	Örtüşen frekans bileşenleri..... 50
Şekil.5.12	Zaman kaydı gösterimi ..... 50
Şekil 5.13	Spektrum sızıntılı zaman kaydı ..... 51
Şekil 5.14	Spektrum sızıntısının frekans düzlemi gösterimi ..... 51
Şekil 5.15	Pencerelemiş spektrumun frekans karakteristikleri..... 52
Şekil 5.16	Hanning penceresi ve etkisi ..... 52
Şekil 5.17	Flatop penceresinin frekans yanıtı..... 53
Şekil 5.18	Devre şeması ..... 54
Şekil 5.19	Harmonik analiz programı akış diyagramı ..... 55
Şekil 5.20	Harmonik analiz arayüz programı akış diyagramı ..... 56
Şekil 5.21	Matlab'e göre bağıl frekans hataları..... 60
Şekil 5.22	Matlab'e göre bağıl genlik hataları..... 61



## ÇİZELGE LİSTESİ

	Sayfa
Çizelge 3.1 ALU durum bayrakları.....	14
Çizelge 3.2 Seri port uç tanımlamaları.....	21
Çizelge 4.1 Algoritmaların örnek sayısına göre $\mu$ sn cinsinden karşılaştırılması.....	31
Çizelge 4.2 Örnek sayısı 1024 iken algoritmalara göre byte olarak bellek kullanımı .....	32
Çizelge 5.1 Deney-1' in sonuçları.....	57
Çizelge 5.2 Deney-2' nin sonuçları.....	57
Çizelge 5.3 Deney-3' ün sonuçları.....	58
Çizelge 5.4 Deney-4' ün sonuçları.....	58
Çizelge 5.5 Deney-5' in sonuçları.....	59
Çizelge 5.6 Deney-6' nın sonuçları.....	59
Çizelge 5.7 Frekans bakımından sonuçlar.....	60
Çizelge 5.8 Genlik bakımından sonuçlar.....	61



## **ÖNSÖZ**

**Bu tezin hazırlanmasında yardımlarını esirgemeyen tez danışmanım Sayın Yrd. Doç. Dr. Tuncay Uzun'a , yazılım ve donanım bakımından destek veren Analog Devices firması yetkililerine, değerli iş arkadaşlarıma ve bana daima destek olan aileme teşekkürü bir borç bilirim.**



## ÖZET

Bir Sayısal İşaret İşleyici kullanılarak, ayırık zamanda Fourier Dönüşümüyle şebeke harmoniklerinin analizi gerçekleştirilmiş ve bir yazılım aracılığıyla analiz sonuçları ölçekli olarak görselleştirilmiştir.

Enerji sistemlerinde, akım ve gerilimin 50 Hz temel frekansta ve sinüzoidal dalga şeklinde olması idealdir. Ancak işletmelerdeki V-I karakteristiği doğrusal olmayan yükler, sinüs olmayan akımlar çekerler ve bu akımlar besleme noktasındaki gerilimin dalga şeklini bozarlar. Tüketicilerin kullandığı cihazları besleyen gerilimin kalitesi, cihazın verimini ve ömrünü direkt olarak etkilediğinden harmoniklere karşı bazı önlemler alınmak zorunda kalmıştır. Çeşitli gerilim ve akımlar için harmoniklerin sınırlandırılmış değerleri “Toplam Harmonik Bozulması (THD)” ölçüt alınarak belirlenmiştir.

Şebekedeki sinüzoidal değişimden uzaklaşmış işaretlerin, içerdiği 0 ile 22 kHz arasındaki harmonikler ve genlikleri hakkında bilgi edinilebilecek bir sistem geliştirilmiştir. Bu sistem elde ettiği bilgileri, seri port üzerinden özel bir yazılım aracılığıyla bilgisayara aktararak kullanıcının analizi daha rahat bir şekilde incelemesi sağlanmaktadır.

**Anahtar Kelimeler:** Harmonik analiz, sayısal işaret işleyici, spektrum analizi

## **ABSTRACT**

The analysis of power system harmonics is realized by a Digital Signal Processor, using Discrete Fourier Transform Method. The results, are visualized by a host program on a personal computer.

In the energy systems, it is required that the current and the voltage are ideally sinusoidal and 50 Hz. Because of some loads which have non-linear V-I characteristic, the sinusoidal main wave is formed as a non-sinusoidal one. As the power quality, supplies consumer goods, affects the life time and productivity of them, it's become a big necessary to take some precautions against harmonics. Some countries are determined "Harmonic Standarts" as Total Harmonic Distorsion (THD). By this way "Harmonic Analysis" has become a big requirement. The limited harmonic values for several currents and voltages are determined by considering total harmonic distorsion (THD) as a reference.

A new system is developed, to get exact information about the non-sinusoidal signals contain harmonics, from 0 to 22 [kHz] . This system sends the harmonic details to a personal computer to show the data.

**Keywords:** Harmonic analysis, digital signal processor, spectrum analysis



## 1. GİRİŞ

Bu çalışmada bir sayısal işaret işleyici (DSP) kullanılarak, ayrık zamanda Fourier Dönüşümüyle (DFT) şebeke harmoniklerinin analizi gerçekleştirilmiş ve bir yazılım aracılığıyla analiz sonuçları ölçekli olarak görselleştirilmiştir.

Modern güç sistemleri çok çeşitli yükler içermektedirler. Bu yüklerin bir çoğunun, sinüzoidal işarete cevapları doğrusal değildir. Doğrusal olmayan yükler, akımı anahtarlayan ve yüksek harmonik seviyelerine sebep olan elektronik devrelerdir. Oluşan bu harmonikler, sistem endüktans, kapasitans ve dirençlerine etki ederek, şebekenin akım ve gerilim dalga şeklini bozarlar.

Hız kontrol sürücüleri ve güç çeviricileri tarafından oluşturulan harmonik akımlar, gerilim bozulmaları yaratacak şekilde sistem empedansına etki ederler. Aşırı gerilim bozulmaları elektronik cihazların yanlış çalışmasına sebep olurlar. Ayrıca temel frekans dışında sisteme ilave olan harmonikler enerji kayıplarını da artırırlar. Güç sistemlerindeki bu harmonikler, özel filtreleme devrelerince engellenirler.

Sinüzoidal olmayan ancak Dirichlet koşulları olarak bilinen; bir periyot içinde sonlu sayıda süreksizlik bulunması, sonlu sayıda maksimum ve minimum bulunması ve ortalamasının sonlu değerde olması koşullarını sağlayan bu dalgalar, Fransız matematikçi J. Fourier tarafından tanımlanmış seriye açıldığında genlik, faz açısı ve frekans olarak birbirinden farklı sinüzoidal dalgaların toplamı olarak ifade edilebilmektedir. Temel dalga frekansındaki dalganın dışındaki dalgalara, "harmonik" dalgalar adı verilmektedir.

Bu harmoniklerin sistemlerde meydana getirdiği; kontrol, kumanda ve koruma devrelerinde işlev bozuklukları, elektrik makinelerinde titreşim, gürültü, aşırı ısınma ve güç kaybı gibi problemleri önlemek için harmonikler hakkında bilgi edinme gereksinimi doğmuştur bu da harmonik analizini gerekli kılmıştır.

Şebeke harmoniklerinin sınırlandırılmış değerleri, şebeke harmonik standartları ile belirlenmiştir. Ölçüt olarak, Toplam Harmonik Bozulması (THD) alınmaktadır. Bu standartlardan EN 61000-3-2, Avrupa Birliği'nde 1 Haziran 1998'den beri uygulanmaktadır. Bu tarihten itibaren hiçbir cihaz bu standart limitlerin üstünde şebekeyi kirletecek düzeyde harmonik üretecek şekilde piyasaya sürülemez. En son oluşturulan standart ise EN 5160 standardıdır.

Bu çalışmanın benzeri bir çalışma, Eylül 1992'de Gerald McGuire tarafından ADSP-2101 sayısal işaret işleyicisi kullanılarak gerçekleştirilmiştir. Çalışmada, 0 ile 4 kHz arasındaki işaretlerin 256 noktalı FFT'si alınarak harmonik bilgileri elde edilmiş ve standart bir osiloskop kullanılarak bu bilgilerin gösterilmesi yoluna gidilmiştir.

Harmonik bilgilerini elde etme anahtar işlev olan FFT, ile ilgili bazı çalışmalar incelenmiş ve FFT tümleşik devreleri tasarlanırken dikkat edilen noktalar bu çalışmanın yazılımı gerçekleştirilirken göz önünde bulundurulmuştur. Bu çalışmalar şunlardır; Mayıs 1997'de Westminster Üniversitesi'nden Ediz Çetin, İzzet Kale ve Richard Morling, 256 noktalı FFT alan CMOS teknolojisi kullanan bir tüm devre geliştirmişlerdir. Yine aynı paralelde İspanya Cantabria Üniversitesi'nden, Jesus Garcia, Juan Michell ve Angel Buron 2048 noktalı karmaşık taban (split radix) kullanan bir tümleşik devre geliştirmişlerdir.



## 2. HARMONİK ANALİZİ

Harmonik Analizi'nin temel kavramları; harmonik tanımı, analizi ve yöntemleri irdelenecektir.

### 2.1 Harmonik Kavramı

Doğrusal olmayan elemanların doğasından kaynaklanan en önemli bozucu etki akım ve gerilim dalga biçimlerinin periyodik olmakla birlikte, sinüzoidal değişimden uzaklaşmalarıdır. Sinüzoidal olmayan ancak Dirichlet koşulları olarak bilinen; bir periyot içinde sonlu sayıda süreksizlik bulunması, sonlu sayıda maksimum ve minimum bulunması ve ortalamasının sonlu değerde olması koşullarını sağlayan bu dalgalar, Fransız matematikçi J. Fourier tarafından tanımlanmış seriye açıldığında genlik, faz açısı ve frekans olarak birbirinden farklı sinüzoidal dalgaların toplamı olarak ifade edilebilmektedir. Temel dalga frekansındaki dalganın dışındaki dalgalara, "harmonik" dalgalar adı verilmektedir. Herhangi bir  $f(t)$  işareti, Fourier serisine açıldığında aşağıdaki ifade elde edilir (Kocatepe,2002).

$$\begin{aligned}
 f(t) &= \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos nt + \sum_{n=1}^{\infty} b_n \sin nt \\
 &= \frac{a_0}{2} + a_1 \cos t + a_2 \cos 2t + a_3 \cos 3t + \dots \\
 &\quad + b_1 \sin t + b_2 \sin 2t + b_3 \sin 3t + \dots
 \end{aligned} \tag{2.1}$$

İfadeyi yeniden düzenlersek;

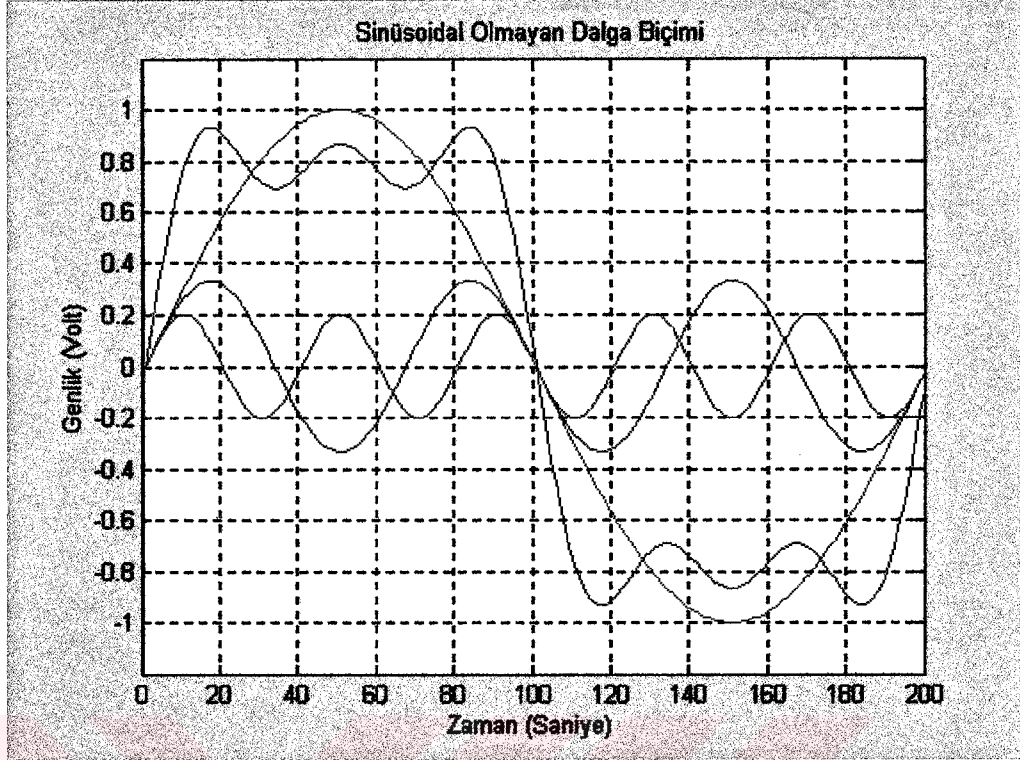
$$\begin{aligned}
 f(t) &= \frac{a_0}{2} + (a_1 \cos t + b_1 \sin t) + (a_2 \cos 2t + b_2 \sin 2t) + \\
 &\quad (a_3 \cos 3t + b_3 \sin 3t) + \dots
 \end{aligned} \tag{2.2}$$

şeklini alır ve buradan;

$(a_1 \cos t + b_1 \sin t)$  terimi temel bileşen

$(a_2 \cos 2t + b_2 \sin 2t)$  terimi 2. harmonik

$(a_3 \cos 3t + b_3 \sin 3t)$  terimi 3. harmonik olacaktır.



- Sinüsoidal olmayan dalga
- - - Temel bileşen
- · · Üçüncü harmonik
- · - Beşinci harmonik

Şekil 2.1 Harmonikli işaret

## 2.2 Harmonik Üreten Kaynaklar

Enerji sistemlerinde, enerjinin üretilmesi, dağıtılması ve tüketilmesi sırasında akım ve gerilimin 50 Hz temel frekansta ve sinüsoidal dalga şeklinde olması idealdir. Bu koşul elektrik enerjisinin kalitesini belirleyen ana faktörlerden biridir. Ancak işletmelerdeki V-I karakteristiği doğrusal olmayan yükler, sinüs olmayan akımlar çekerler ve bu akımlar şebekede sinüs olmayan gerilim düşümleri oluşturarak besleme noktasındaki gerilimin dalga şeklini bozarlar. Magnetik devrelerin aşırı doyması, elektrik arkları ve güç elektroniği devrelerindeki gerilimin anahtarlanması ve kıyılması doğrusal olmayan (OHM Kanunu'nun geçerli olmadığı) olaylardır. Harmonikler yük tarafından oluşturulabileceği gibi şebeke tarafından da oluşturulabilir. Harmonik kaynakları aşağıdaki gibi sınıflanabilir;

### 2.2.1 Harmoniklerin Yük Tarafında Üretilmesi

- Güç Elektroniği Devreleri ve Doğru Akım Kullanımı

- Motor Hız Kontrol Devreleri, Akü Şarj Devreleri
- DC Çeviriciler (Doğrultucu Devreler)
- Kesintisiz Güç Kaynakları (UPS), foto voltaik sistemler
- Anahtarlama Güç Kaynakları ( SMPS içeren bilgisayarlar)
- Kaynak Makineleri, Elektrikli Ulaşım Vasıtaları, Doğru Akımla Enerji Taşıma
- Ark İlkesiyle Çalışan Yükler
- Ark Fırımları, Statik VAR Dengeleyiciler
- Flüoresan ve Elektronik Balastlar
- Deşarj Armatürleri (Cıva buharlı neon, ksenon, yüksek basınçlı sodyum lambalar)
- Elektro mekanik cihazlar
- Transformatörler
- Asenkron ve Senkron Motorlar

### **2.2.2 Harmoniklerin Şebeke Tarafında Üretilmesi**

- Senkron Üreteç (generatör) ve Transformatör
- Yıldırım
- Şebeke Kararsızlıkları ve Sınımları

### **2.3 Harmoniklerin Ortaya Çıkardığı Problemler**

- Hatlarda ve elektrik tesislerindeki kayıplarda artış
- Yalıtımı zayıflattıklarından, tesis elemanlarının ömürlerinde azalma
- Güç faktöründe bozulma
- Seri veya paralel rezonanstan oluşan aşırı akım ve gerilimler
- Kontrol, kumanda ve koruma devrelerinde işlev bozuklukları
- Ölçme hataları
- Telefon ve veri hatlarında gürültü ve girişim

- Elektrik makinelerinde titreşim, gürültü, aşırı ısınma ve güç kaybı

Harmoniklerin ortaya çıkardığı bu problemlerden dolayı ve tüketicilerin kullandığı cihazları besleyen gerilimin kalitesi, cihazın verimini ve ömrünü direkt olarak etkilediğinden harmoniklere karşı bazı önlemler alınmak zorunda kalmıştır. Bu önlemler standartları doğurmuş ve çeşitli ülkeler tarafından, çeşitli gerilimler için harmoniklerin sınırlanmış değerleri “Toplam Harmonik Bozulması (Total Harmonic Distortion) ” ölçüt alınarak belirlenmiştir. Örneğin bu ölçüt gerilim için,  $U_1$ = temel bileşenin etkin değeri,  $U_n$ = n.harmoniğin etkin değeri olmak üzere;

$$\text{Toplam Harmonik Bozulması (THD}_v\text{)} = \frac{\sqrt{\sum_{n=2}^{\infty} U_n^2}}{U_1} \quad (2.3)$$

şeklinde tanımlanır. Bu standardizasyon, harmonik ölçümlerini ve analizini gerekli kıldı. Bu çalışmada bir sayısal işaret işleyici kullanılarak ayrık zamanda Fourier Dönüşümüyle harmonik ölçümü gerçekleştirilmiştir.

#### 2.4 Harmonik Analizi

Harmonik analizi, kompleks periyodik bir işaretteki harmoniklerin faz, genlik ve frekans değerlerinin belirlenmesidir. Aynı zamanda spektrum analizinin özel bir halidir bu sebeple önce spektrum analizi irdelenecektir.

**Spektrum (Tayfsal) analizi:** Bir işaretin gücünün, frekanslar üzerine nasıl dağıldığını tahmin etmek veya belirlemek anlamına gelir. Spektrum analizinde göz önünde bulundurulması gereken 3 önemli kavram vardır;

1. Çözünürlük (Resolution) = Birbirine yakın frekans bileşenlerini ayırma yeteneği
2. Taban (Bias)= Gerçek spektrum ile belirlenen spektrum arasındaki ortalama fark
3. Değişim (Variance) = Gerçek spektrum ile belirlenen spektrum arasındaki ortalama karesel fark

Spektrum analizi 4 farklı biçimde gerçekleştirilebilir;

1. Çoklu Konik Spektrum Tahmini (Multiple Taper Spectral Estimation)
2. Parametrik Olmayan Spektrum Tahmini (Non-Parametric Spectral Estimation)

### 3. Parametrik Spektrum Tahmini (Parametric Spectral Estimation)

#### 4. Belirli Spektrum Analizi (Deterministic Spectral Analysis)

##### 2.4.1 Çoklu Konik Spektrum Kestirimi

Bu metot işaretin spektrumunun yüksek dinamikliği olduğu durumlarda bir çok bilim alanında kullanılmaktadır.

##### 2.4.2 Parametrik Olmayan Spektrum Kestirimi

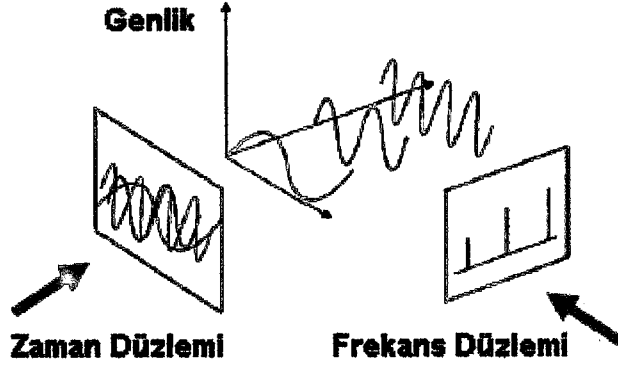
Bu yöntem, Periodogram ve Correlogram Metotlarını içerir. Periodogram, bir işaretin Fourier Dönüşümünün her bir frekanstaki gücüdür. Correlogram, bir işaretin otokorelasyonunun Fourier Dönüşümüdür.

##### 2.4.3 Parametrik Spektrum Kestirimi

Bu metot analiz edilecek işaretin oransal veya dizi şeklinde bir spektrumunun olduğunu farz eder. Oransal spektrum durumunda, veriye bir fark eşitliği uydurulur. Dizi spektrum durumunda ise işaretin fazı, genliği ve frekans değerleri bilinmeyen sinüzoidler içerdiği farz edilir ve problemin bundan sonrası bir tahmin problemi gibi çözülür.

##### 2.4.4 Belirli Spektrum Analizi

Bir işaret osiloskop ile gözlemlendiğinde zaman düzleminde görüntülenir. Dikey eksen gerilim, yatay eksen ise zamandır. Birçok işaret için bu gözleme tekniği yeterlidir ancak ilgilenilen nokta işaretin içerdiği frekans bileşenleri olduğunda, işareti frekans düzleminde gözlemlemek gerekecektir. Frekans düzlemine geçişin asıl amacı, kompleks bir işaretin içerdiği frekans bileşenleriyle işlem yapmanın bu düzlemde daha kolay olmasıdır. Frekans düzleminde, dikey eksen hala gerilimdir (o frekanstaki genlik değeri) fakat yatay eksen frekanstır. Frekans düzlemi, işaretin enerjisinin ne kadarının hangi frekansta olduğunu gösterir.



Şekil 2.2 Harmonikli işaretin zaman ve frekans düzlemi gösterimi

Sürekli bir  $x(t)$  işaretinin frekans düzlemi gösterimi, aşağıdaki formülle ifade edilir;

$$X(\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t} dt \quad (2.4)$$

Sayısal ölçme sistemlerinde, ölçme işlemi belirli aralıklarla örneksel işaretten örnek alınarak yapılır. Bu anlamda, örneklenmiş  $T$  periyotlu bir  $x_s(t)$  işareti düşünülürse;

$$x_s(t) = \sum_{-\infty}^{\infty} x(nT)\delta(t - nT) \quad (2.5)$$

ve bunun sürekli zaman Fourier Dönüşüm alınırsa;

$$X_s(\omega) = \int_{-\infty}^{\infty} \sum_{n=-\infty}^{\infty} x(nT)\delta(t - nT)e^{-j\omega t} dt \quad (2.6)$$

$$X_s(\omega) = \sum_{n=-\infty}^{\infty} x(nT)e^{-j\omega nT}$$

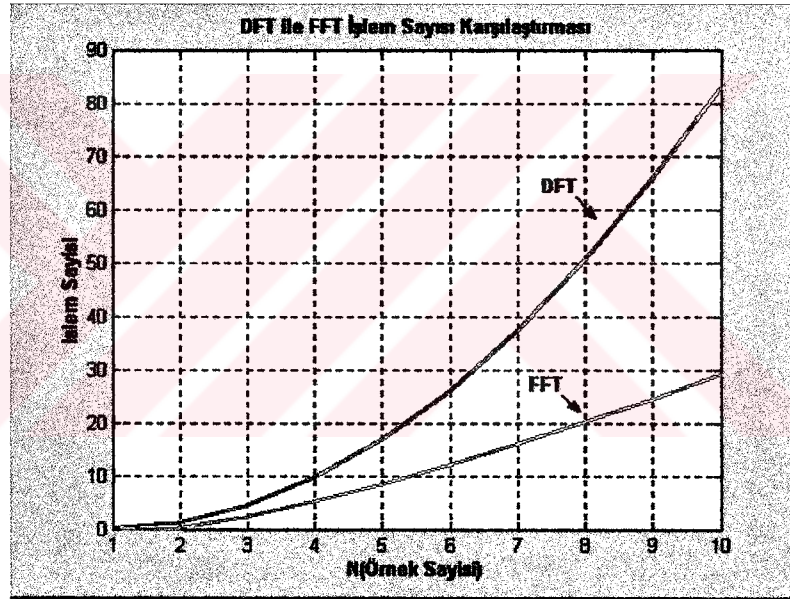
şeklinde olacaktır. Aşağıdaki eşitlik,  $\omega$ ; açısal frekansı,  $f_s$ ; örnekleme frekansı olmak üzere;

$$\Omega = \omega T = 2\pi f T = 2\pi \frac{f}{f_s} \quad (2.7)$$

$x(n)$ 'in Ayırık Zamanlı Fourier Dönüşümü (DTFT) aşağıdaki gibi elde edilir;

$$X(e^{j\Omega}) = \sum_{n=-\infty}^{\infty} x(n)e^{-jn\Omega} \quad (2.8)$$

Ayrık Zamanlı Fourier Dönüşümü (DFT), sayısallaştırılmış zaman düzlemi verisini alır ve frekans düzlemi bilgisini hesaplar. Ayrık Fourier Dönüşümünün doğrudan hesaplanmasında, frekans spektrumundaki her bir değer için,  $N^2$  (örnek sayısı) tane işlem yapılmaktadır. Bu işlemler spektrumdaki frekans adedi arttıkça, fazlalaşacak ve çok zaman gerektirecektir. Bu nedenle Ayrık Fourier Dönüşümü hesaplanmasında, etkin olarak kullanılan 2 farklı yaklaşım kullanılır. Bunlardan biri Hızlı Fourier Dönüşümü, diğeri ise veri üzerinde doğrusal filtreleme işlemi yapan ve 2 tip algoritması olan (Goertzel ve Chirp-z algoritmaları) bir dönüşümdür. 4. Bölümde ayrıntılı olarak incelenecek olan Hızlı Fourier Dönüşümünde ise işlem sayısı  $N \log_2 N$ 'dir. Örnek sayısı arttıkça Hızlı Fourier Dönüşümünün etkinliği daha fazla ortaya çıkacaktır. Aşağıdaki grafikte, bu noktayı görsel olarak ispatlamaktadır;



Şekil 2.3 DFT-FFT karşılaştırması

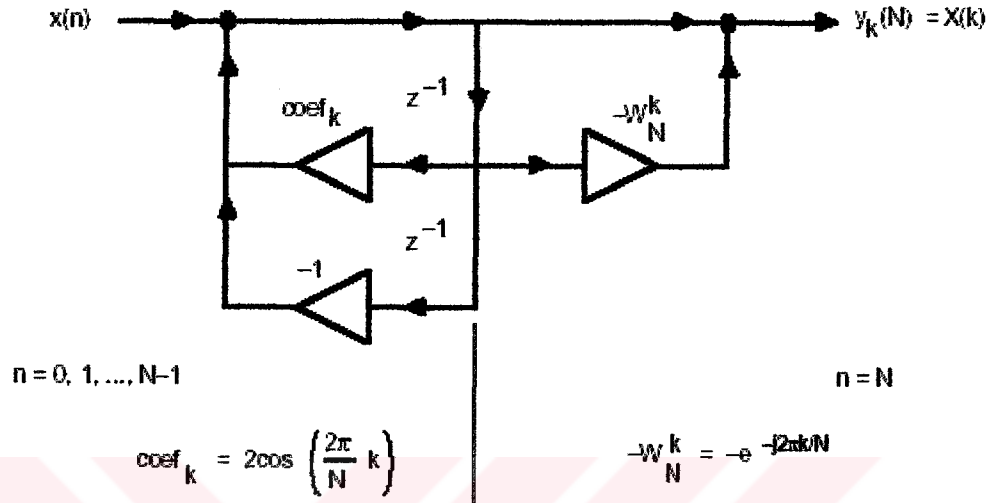
#### 2.4.4.1 Doğrusal Filtreleme İşlemi

Bazen spektrumum hesaplarken bütün frekans bileşenleri yerine sadece bazı frekans bileşenleri bulunmak istenebilir. Bu gibi durumlar için Goertzel algoritması kolaylık sağlar. Goertzel algoritmasında daha az kompleks işlem yaparak sadece istenen frekans bileşenleri bulunabilir. Uygulamadaki en önemli örneklerden birisi de telefon santrallerindeki DTMF ton algılama sistemleridir.

N adet örnek alınan bir işaretin k.ıncı frekans bileşeninin gücünü bulmak için

$$y_k(n) = W_N^{-k} \cdot y_k(n-1) + x(n) \quad (2.9)$$

Fark denklemi  $n=0$ 'dan  $N$ 'e kadar uygulanarak bulunabilir. Direk Form yapısında gösterimi şekildeki gibidir.



Şekil 2.4 Goertzel algoritması (Kayran, 1990)

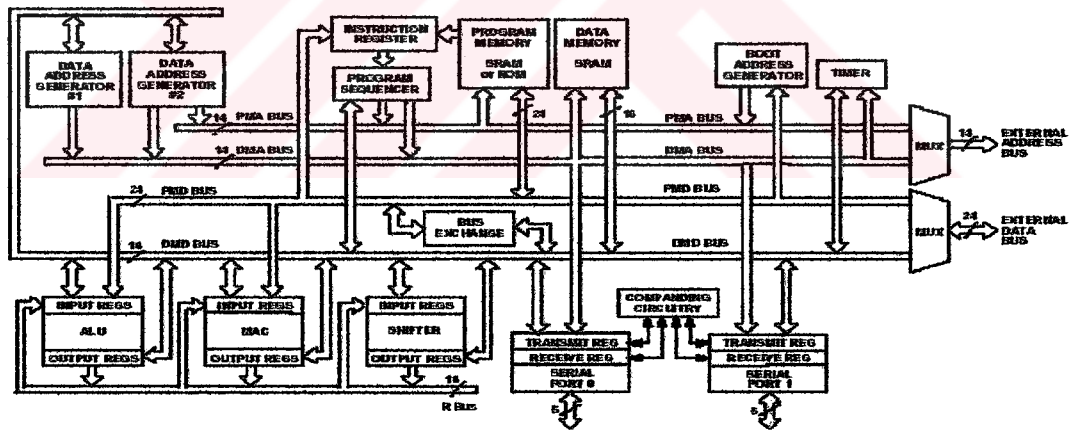
Goertzel Algoritması frekansın bazı bileşenleri bulunmak istendiğinde kullanılır. FFT ye göre bellek kapasitesi yönünden ve hız bakımından avantaj sağlar.

### 3. SAYISAL İŞARET İŞLEYİCİ

Sayısal işaret işleme sistemlerinde kullanılan hesaplamaların, hızlı bir şekilde yapılabilmesi için DSP'lerde özel donanımsal üstünlükler bulunmaktadır. Örneğin; çarpma işleminin gerçekleştirilmesi için DSP'lerde çarpıcı-toplayıcı (MAC) mantık birimi yer almaktadır. MAC sayesinde işlemci, çarpma ve çarpmanın sonucunu başka bir değişkenle toplayarak sonuç verme işlemini sadece tek iş (makine) çevriminde yapmaktadır. Bu klasik mikroişlemci ve mikrodenetleyicilerde olmayan bir özelliktir. Çarpma ve bölmeyi hızlı bir şekilde yapabilme yeteneği sayesinde pek çok, hız gerektiren uygulamalarda klasik mikroişlemci yerine, DSP'ler kullanılmaktadır. Bu çalışmada da, sayılan sebeplerden dolayı, Analog Devices firmasının ADSP 2181 Sayısal işaret işleyicisi seçilmiştir.

Sayısal işaret işleyicinin yapısı ve çalışma biçimi ayrı bir çalışma olabilecek kadar geniş olduğundan burada yalnızca genel olarak bahsedilecek ve seri port biraz daha ayrıntılı incelenecektir;

#### 3.1 ADSP 2181'in Genel Mimarisi:



Şekil 3.1 ADSP-21XX blok diyagramı (Vinay, 1995)

Şekilde ADSP-21xx mimarisinin blok diyagramını görülmektedir. İşlemci, birbirinden bağımsız üç hesaplayıcı birimden oluşmaktadır. Bunlar: Aritmetik Mantık Birimi (ALU), Çarpıcı/Toplayıcı (MAC) ve Kaydırıcı (SHIFTER)'dir. Aritmetik Mantık Birimi, genel amaçlı aritmetik ve mantık işlemleri içindir. Çarpıcı/Toplayıcı, tek makine çevriminde çarpma/toplama, çarpma/çıkarma işlemlerinin yapılabilmesini sağlar. Kaydırıcı, ise kaydırma ve üstel hesaplama gibi işlemlerde kullanılır. İç sonuç (R) yolu, doğrudan hesaplama birimlerine bağlanır. Bu sayede herhangi bir birimin çıkışı diğer birimin girişi olarak

kullanılabilir. Program Sıralayıcı; programda döngü oluşturulmasını kolaylaştırır. Koşullara bağlı atlamaları destekler ve alt programları çağırır. Veri Adres Üreteçleri, program belleği ve veri belleği için sıralı adresler üretirler.

Etkili veri transferi, aşağıdaki beş iç veri yolunun kullanılması ile gerçekleşir;

- Program Bellek Adres (PMA) Yolu
- Program Bellek Veri (PMD) Yolu
- Veri Bellek Adres (DMA) Yolu
- Veri Bellek Veri (DMD) Yolu
- Sonuç (R) Yolu

### 3.1.1 Hesaplayıcı Birimler

ADSP-21XX serisi işlemciler 16-bit sabit noktalı (Fixed Point) işlemcilerdir. Sabit noktalı işlemcilerde sayılar tamsayı veya ondalıklı olarak işlem görür. İşaretli sayılar ise daima ikiye tümleyen sayı biçimi şeklinde işlem görürler. Ondalık sayılar ise 1.15 sayı biçimindedir.

1.15 SAYI	DESİMAL EŞDEĞERİ
0x0001	0.000031
0x7FFF	0.999969
0xFFFF	-0.000031
0x8000	-1.000000

$2^0$	$2^{-1}$	$2^{-2}$	$2^{-3}$	$2^{-4}$	$2^{-5}$	$2^{-6}$	$2^{-7}$	$2^{-8}$	$2^{-9}$	$2^{-10}$	$2^{-11}$	$2^{-12}$	$2^{-13}$	$2^{-14}$	$2^{-15}$
-------	----------	----------	----------	----------	----------	----------	----------	----------	----------	-----------	-----------	-----------	-----------	-----------	-----------

Şekil 3.2 1.15 Sayı formatı (Vinay, 1995)

#### 3.1.1.1 Aritmetik Mantık Birimi

Aritmetik mantık birimi (ALU), her mikroişlemcide bulunan klasik toplama, çıkarma mantık işlemleri vs. gibi işlemleri yürüten birimdir. Aşağıda blok diyagramı görülen ALU'nun Doymalı Çalışma Durumu, Tutmalı Çalışma Durumu gibi çeşitli çalışma durumları bulunmaktadır.



Çizelge 3.1 ALU durum bayrakları

Bayrak	İsim	Tanımlama
AZ	Sıfır Biti	ALU çıkış kaydedicisi sıfır ise lojik "1" olur.
AN	Negatif Biti	ALU çıkış kaydedicisi negatif ise lojik "1" olur.
AV	Taşma Biti	En ağırlıklı bitten taşma varsa lojik "1" olur.
AC	Elde Biti	En ağırlıklı bitten Elde gelmişse lojik "1" olur.
AS	İşaret Biti	X giriş portunun işaretini belirtir.
AQ	Kalan Biti	DIVS ve DIVQ komutlarıyla yapılan bölme işleminde kalanı belirtir.

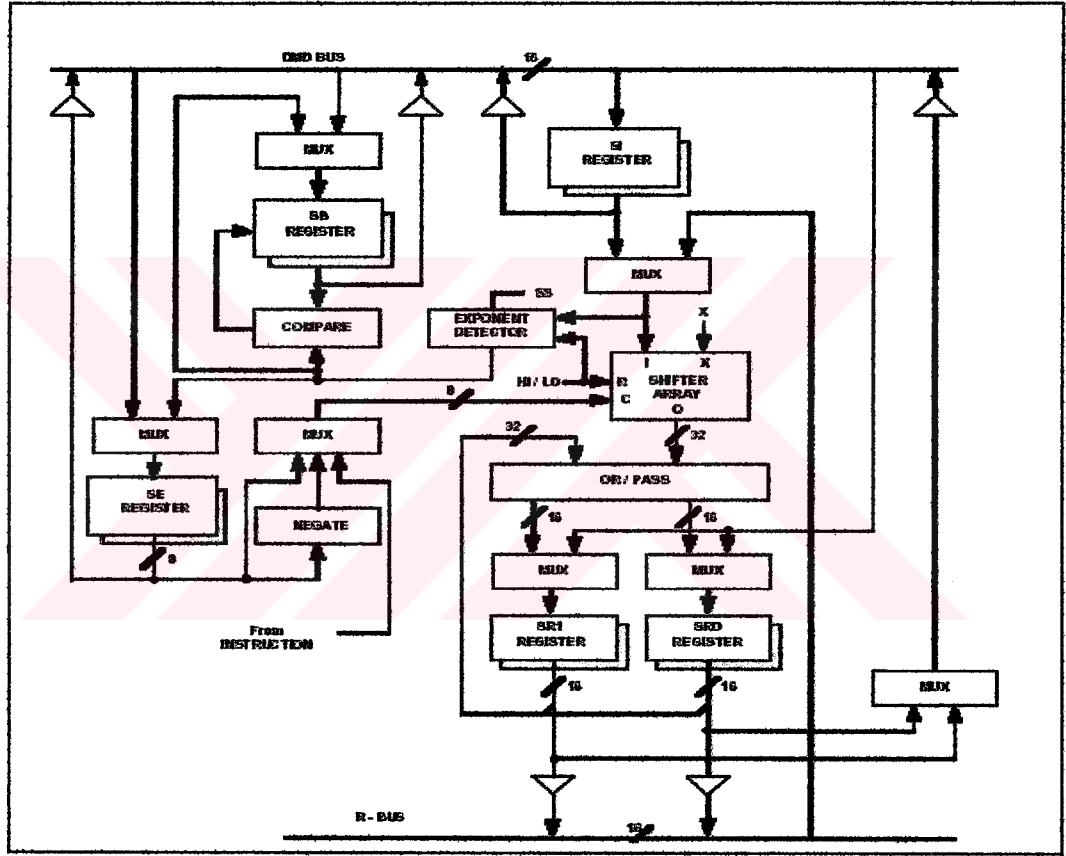
### 3.1.1.2 Çarpıcı/Toplayıcı

Tamsayı ve ondalıklı sayıların bir makine çevriminde çarpılıp toplanmasını sağlar. DSP işlemcilerini, mikroişlemcilerden ayıran en önemli özelliklerden biridir. Çarpma ve Toplama işlemini tek komut çevriminde yapabilme yeteneği sayesinde pek çok uygulamanın hızlı bir şekilde gerçekleştirilmesine olanak verir. Giriş kaydedicileri MX, MY olup çıkış kaydedicileri de MR kaydedicileridir. Yapılacak işlemin tamsayı mı kesirli mi olduğu MSTAT kaydedicisindeki 0.bit ile kontrol edilir. Eğer ondalıklı durum seçilmiş ise çarpım sonucu bir bit sola kaydırılarak doğru sonuç elde edilir çünkü 2 adet işaret biti oluşmaktadır.



### 3.1.1.3 Öteleyici

Aritmetik ve mantık kaydırma işlemlerinde ve blok kayan nokta işlemlerinde kullanılır. Sabit noktalı işlemcilerde bütün sayılar sabit bir yapıda olduğu için o yapının dışına çıkan sayıların gösterilmesi ve bu sayılarla işlem yapılabilmesi için öteleyici kullanılır. Girilen bir dizinin en büyüğü tespit edilerek diziyeye ortak bir üs atanır ve SB kaydedicisinde saklanır. Bu şekilde sabit noktalı işlemcilerle, kayan noktalı işlemcilerin yaptığı işlemlere yakın işlem yapılması sağlanır.



Şekil 3.5 Öteleyici blok diyagramı (Vinay, 1995)

### 3.1.1.4 Öteleyici İşlemleri

- Aritmetik Kaydırma (ASHIFT Komutu)
- Mantıksal Kaydırma (LSHIFT Komutu)
- Normalizasyon (NORM Komutu)
- Üst Tespiti (EXP Komutu)

- Blok Üsteli Ayarlaması (EXPADJ Komutu)

### **3.1.2 Program Kontrol Birimleri**

Program kontrolünde kullanılan birimler tanıtılacaktır.

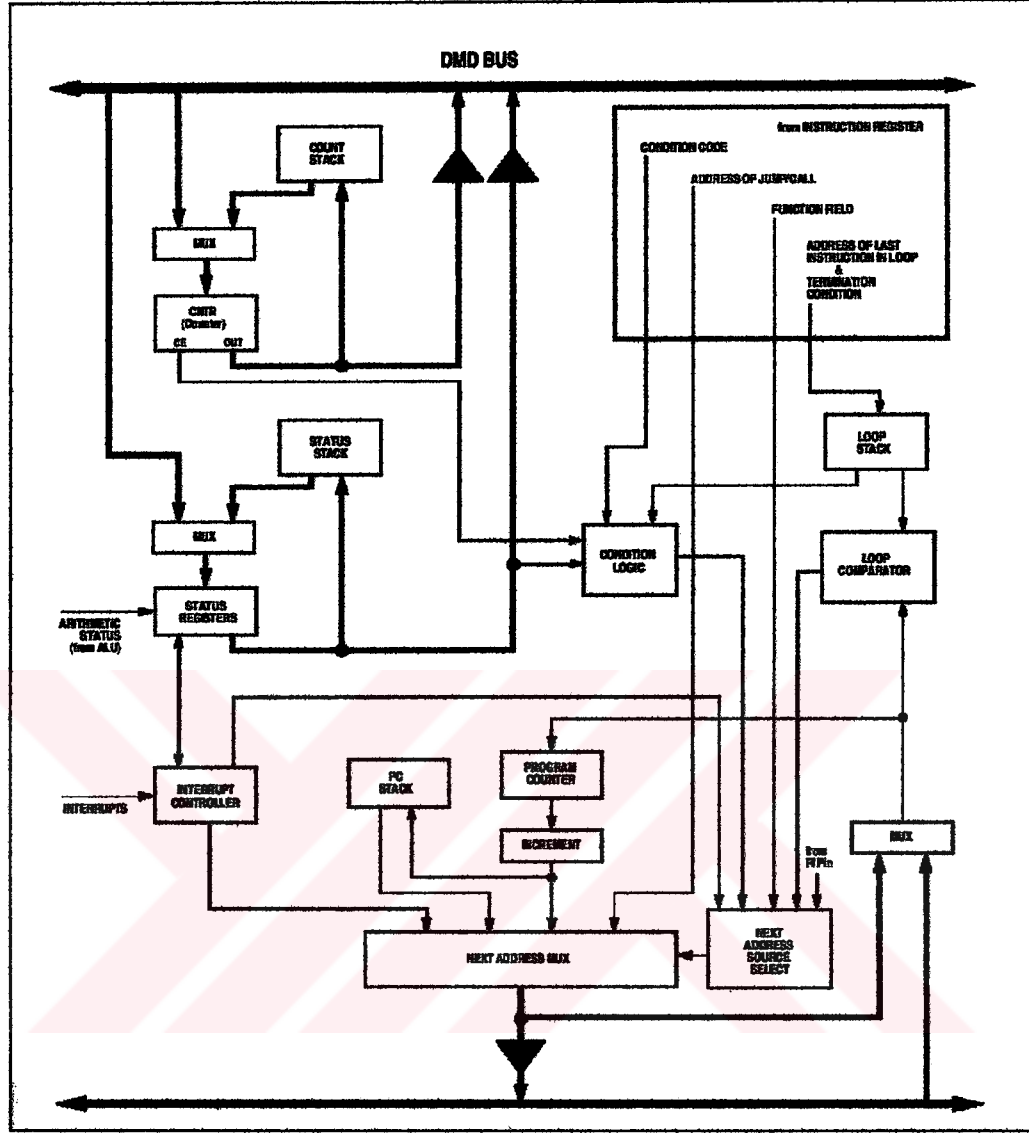
#### **3.1.2.1 Program Sıralayıcı**

Program sıralayıcı program içerisinde döngülerin yapılabilmesi için gerekli adresleri sağlar.

Bu sayede program içerisinde şartlı döngüler oluşturulabilir.

Program Sıralayıcı Mantık Birimi yardımıyla uygulanan komutlar şunlardır;

- DO UNTIL (Şart gerçekleşene kadar yap)
- JUMP (Sıçra)
- CALL (Çağır)
- RTS (Alt rutinden geri dön)
- RTI (Kesmeden geri dön)
- IDLE (Bekleme konumuna geç)



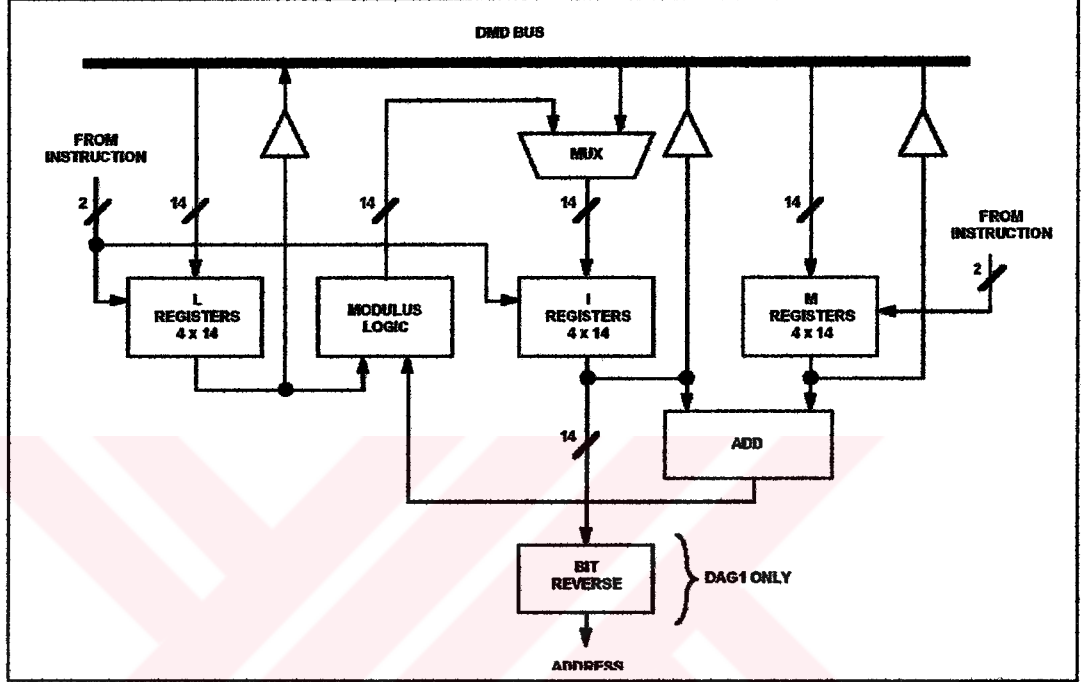
Şekil 3.6 Program sıralayıcı blok diyagramı (Vinay, 1995)

### 3.1.3 Veri Transferi

ADSP-21xx ailesi işlemcilerde veri transferi Doğrudan Adresleme ve Dolaylı Adresleme olmak üzere iki yolla yapılır. Dolaylı Adresleme, Veri Adres Üreteçleri yardımıyla yapılmaktadır.

### 3.1.3.1 Veri Adres Üreteçleri

ADSP-21xx ailesi işlemcilerde 2 adet Veri Adres Üreteci bulunmaktadır. Her bir adres üreticinde 4 adet DAG kaydedicisi bulunmaktadır. DAG1 sadece veri belleği adresi üretme yeteneğine sahipken, DAG2 hem veri hem de program adresi üretme yeteneğine sahiptir.



Şekil 3.7 Veri adres üretici blok diyagramı (Vinay, 1995)

Bir Veri Adres Üretecinde 3 adet kaydedici bulunmaktadır. (M) kaydedicisi komut işlendikten sonra işaretçinin ne kadar arttırılacağını gösterir. (I) kaydedicisi okunacak veya yazılacak bellek bölgesinin adresini tutar. (L) kaydedicisi ise dairesel adreslemede kullanılmaktadır. Eğer (L) kaydedicisi 0'a eşitlenirse, bu dairesel adreslemenin olmayacağı anlamına gelmektedir. Aşağıda basit bir dolaylı adresleme örneği görülmektedir;

$$I3=0x3800;$$

$$M2=0, L3=0;$$

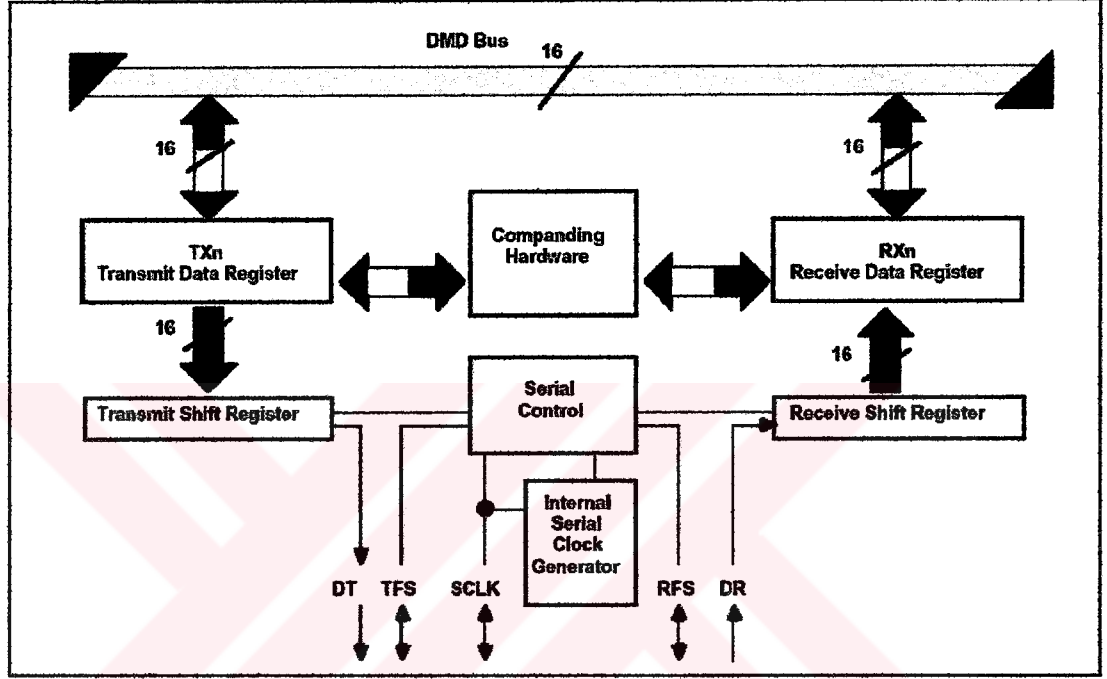
$$AX0=DM(I3,M2);$$

Bu program parçacığı AX0 kaydedicisine 3800h adresindeki veriyi kaydetmektedir.

### 3.2 Sayısal İşaret İşleyicinin Seri Portu

ADSP-21xx serisi işlemcilerinin dış çevre birimler ile iletişim kurabilmesi için 1 veya 2 adet

seri portu bulunmaktadır. Seri portlara genellikle CODEC tümleşik devresi bağlanarak her Sayısal İşaret İşleme uygulaması için gerekli olan D/A ve A/D dönüşümleri gerçekleştirilir. Ayrıca seri portlar aracılığıyla RS-232 arabirimi gibi arabirimlerle de iletişim kurulabilir. Seri portlar A-yasası (A-Law) veya  $\mu$ -yasası ( $\mu$ -Law) tipi sıkıştırılmaları-genişletmeleri de desteklerler.



Şekil 3.8 Seri port yapısı (Vinay, 1995)

Seri port arabirimi, senkron olarak çalışan bir arabirimdir. Gönderme ve alma kesme yeteneğine sahiptir. Otomatik Tampon Bellek (Autobuffering) özelliği de seri portun yeteneklerinden birisidir. Bu özellik sayesinde blok halinde bir bilgi geldiği zaman kesme üretir. Bu özellik veri adres üreteçlerinden birini kullanarak işlev görür. Öncelikle bir dairesel tampon bellek (buffer) tanımlamak gereklidir. Seri porttan gelen bilgiler bu bölgeye baştan başlayarak yazılır. Tampon bellek dolduğunda Kesme (interrupt) üretilir. Daha sonra gelen bilgiler okunduktan sonra yüklemeye devam edilir. Gönderme özelliği de benzer şekilde işlev görür. Ayrıca Çok Kanallı Durum özelliği de mevcuttur. Bu özellik sayesinde TDM modülasyonu rahatlıkla yapılabilir.

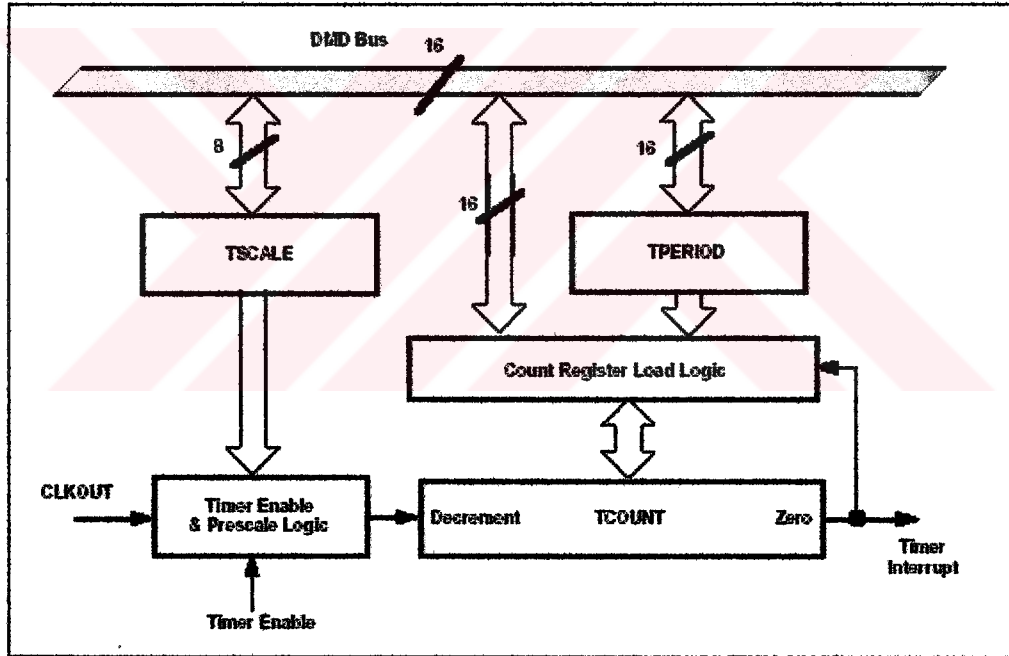
Seri port uçları ve işlevleri aşağıdaki tabloda belirtilmiştir.

Çizelge 3.2 Seri port uç tanımlamaları

İşaret Adı	Fonksiyon
SCLK	Seri Saat (Giriş/Çıkış)
RFS	Alıcı Çerçeve Eş-zamanlama (Giriş/Çıkış)
TFS	Gönderici Çerçeve Eş-zamanlama (Giriş/Çıkış)
DR	Seri Veri Girişi

### 3.3 Zamanlayıcılar

Periyodik olarak kesme üretmek için zamanlayıcılar (timer) kullanılmaktadır. Aşağıda zamanlayıcı blok diyagramı görülmektedir.



Şekil 3.9 Timer blok diyagramı (Vinay, 1995)

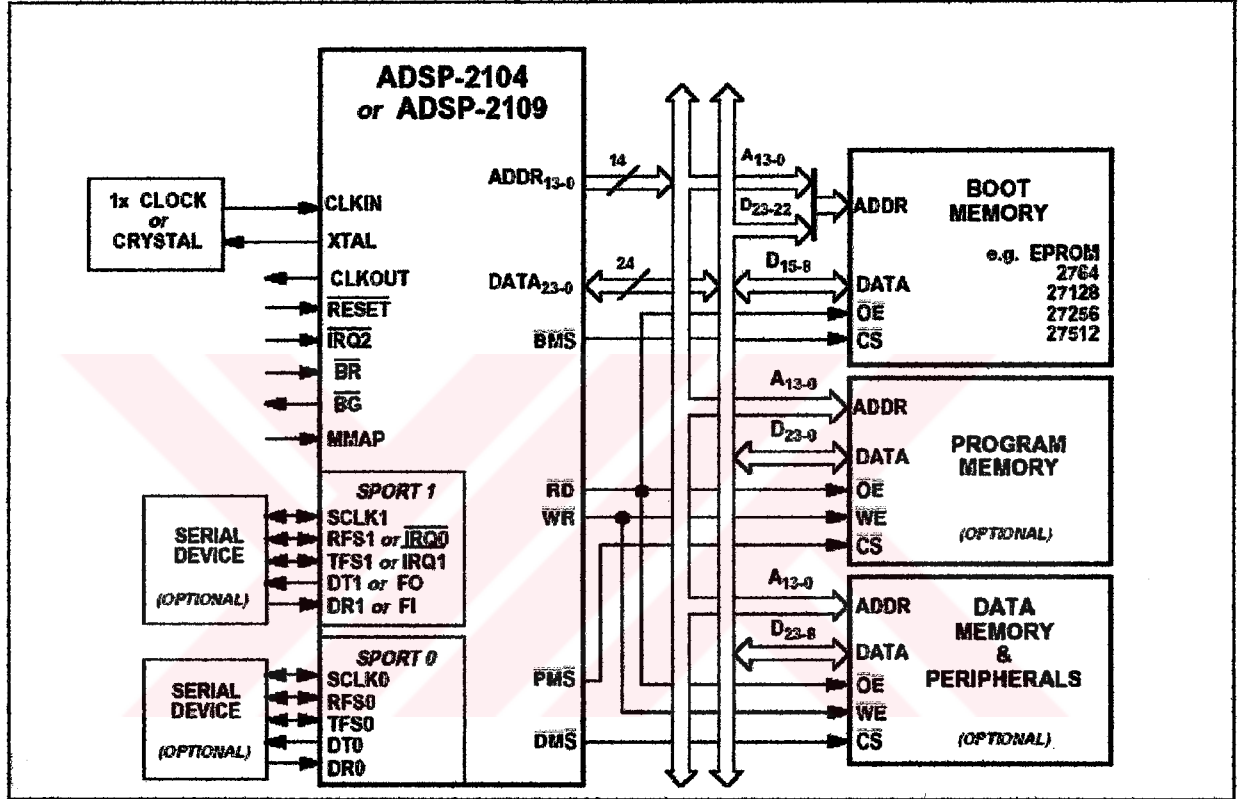
### 3.4 Bellek Arabirimi

Şekil 3-10 da ADSP-2181 işlemci sistemi verilmiştir. Şekilden de anlaşılacağı üzere ADSP-2181 işlemcisi 14 bit adres yoluna ve 24 bit veri yoluna sahiptir. İşlemci içerisinde 512 byte program belleği ve 256 byte veri belleği bulunmaktadır. Fakat blok diyagramda da görüldüğü gibi dışarıdan veri belleği ve program belleği bağlanabilir. Ayrıca bir Ön Yükleme Belleği Birimi (Boot Memory) yardımıyla düşük fiyath bir EPROM kullanılarak iç veri

belleğine istenen program yüklenebilir.

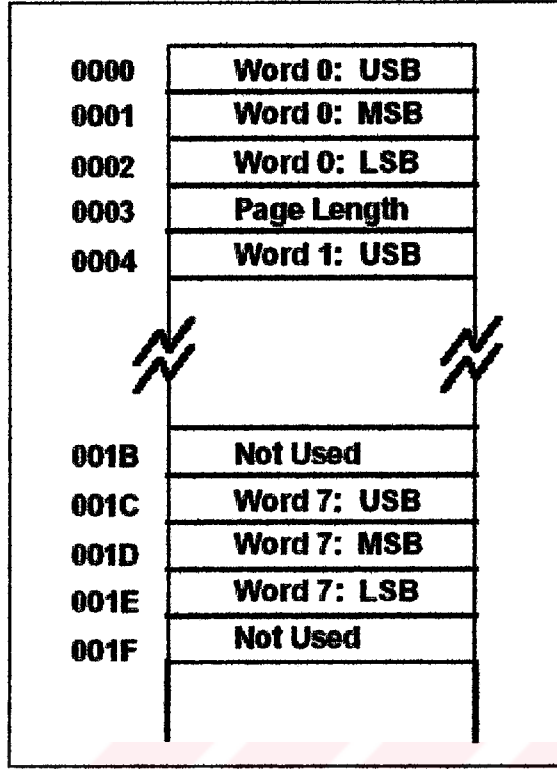
### 3.4.1 Program Belleği Arabirimi

İşlemci 16K'ya kadar 24 bitlik program belleğini maksimum 2K iç bellek olmak üzere adresleyebilir. Komutlar 24 bitlik PMD veri yolu aracılığı ile transfer edilebilir. Dış program belleğine ulaşım için işlemcinin PMS, RD ve WR uçları kullanılır.



Şekil 3.10 ADSP-2181 sistemi (Vinay, 1995)

Şekil 3-11 de ADSP-2181 işlemcisinin Program Bellek haritası gösterilmiştir. Program Bellek Haritasında MMAP ucunun '0' ya da '1' olmasına bağlı olarak iki durum yer almaktadır. Eğer MMAP ucu '0' a çekilmişse iç program belleği bellek haritasında 0'ncı adresten başlayarak yerleşmiştir. Bu durumda işlemci yeniden başlatıldığında harici EPROM dan otomatik olarak programı yüklemeye başlar. MMAP ucu '1' ise bu sefer program belleği yeniden başlama anında otomatik olarak İç Belleğe (RAM) yüklenmez. İç program belleği sadece program esnasında okunur veya yazılabilir.



Şekil.3.11 ADSP-2181 program bellek haritası (Vinay, 1995)

Ayrıca PMOVLAY yazmacındaki bitlerin konumuna göre Program Belleği Dahili ya da Harici olarak konumlandırılabilir.

### 3.4.2 Veri Belleği Arabirimi

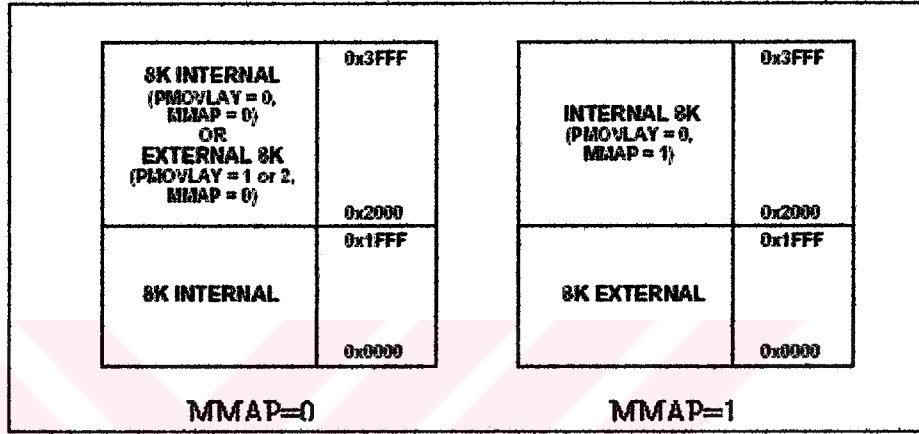
İşlemci veri belleği üzerinde 14 bit adres sağlar. Veriler ise 24 bitlik veri yolunun en ağırlıklı 16 bitinden işlemciye transfer edilir.

<b>32 MEMORY-MAPPED REGISTERS</b>	<b>0x3FFF</b>
	<b>0x3FEO</b>
<b>INTERNAL 8160 WORDS</b>	<b>0x3FDF</b>
	<b>0x2000</b>
<b>8K INTERNAL (DMOVLAY = 0) OR EXTERNAL 8K (DMOVLAY = 1, 2)</b>	<b>0x1FFF</b>
	<b>0x0000</b>

Şekil 3.12 Veri belleği yerleşimi (Vinay, 1995)

### 3.4.3 Ön Yükleme Belleği Arabirimi

ADSP-2181 işlemcisinde 24 bitlik veri yolundan 8 biti ön yükleme bellek birimi (boot memory) için ayrılmıştır. Aynı zamanda 14 bit adres yolu da doğrudan bağlanabilir. Bu veri yoluna ucuz bir EPROM konularak DSP de çalıştırılmak istenen program buraya yüklenebilir. Yeniden Başlama (RESET) anında, MMAP ucu '0' a çekilmişse 8 bitlik EPROM'dan 24 bit genişliğindeki iç Program Belleğine şekilde görünen sırada çalıştırılmak istenen program yüklenebilir.







Şekil 3.13 EPROM'dan iç program belleğine programın yüklenme sırası (Vinay, 1995)

#### 4. FREKANS SPEKTRUMUNUN HESAPLANMASI

Bölüm 2'de tanımlanan belirli spektrum analizi, ayrıntılı olarak incelenecektir.

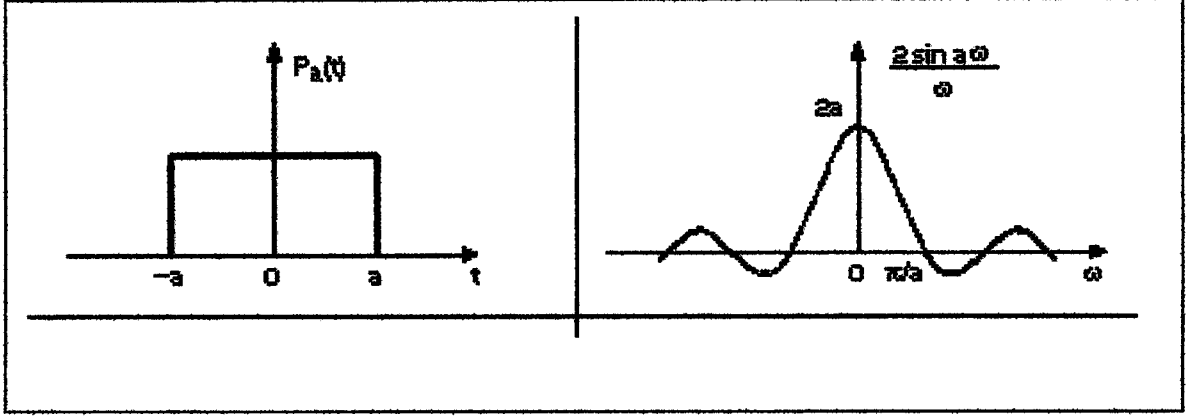
##### 4.1 Belirli Spektrum Analizi

Tüm periyodik ya da periyodik olmayan (aperiyodik) işaretler aslında değişik frekanslardaki sinüzoidal işaretlerin toplamından oluşmaktadır. İşaretlerin Periyodik-Periyodik Olmayan ya da Sürekli-Ayrık olmasına göre çeşitli Fourier Dönüşüm türleri mevcuttur.

Dönüşüm Türü	Örnek İşaret
Fourier Dönüşümü Sürekli ve Aperiodyk İşaretler	
Fourier Serileri Sürekli ve Periyodik İşaretler	
Ayrık Zamanlı Fourier Dönüşümü Ayrık ve Aperiodyk İşaretler	
Ayrık Fourier Dönüşümü Ayrık ve Periyodik İşaretler	

Şekil 4.1 İşaretlerin türlerine göre Fourier dönüşüm tipleri (Wistman, 1997)

Şekilden de görüldüğü gibi işaretin yapısına göre değişik tipte Fourier dönüşümleri mevcuttur. Bölüm 2'de de bahsedildiği gibi sayısal sistemlerle hesaplanmaya uygun olan tek dönüşüm yöntemi Ayrık Fourier Dönüşümü(DFT) dir.



Şekil 4.2 Örnek bir işaretin Fourier dönüşümü (Wistman, 1997)

#### 4.1.1 FFT Algoritmasının Genel Akış Diyagramı

Gerçekleştirilen sistemde kullanılan DSP programının FFT algoritması, Radix-2, 1024 noktalı Zamanda İndirgemeli (DIT) ve şartlı blok kayan nokta dengeleme yöntemi ile gerçekleştirilmiştir. Programda üç adet döngü bulunmaktadır:

- Kelebek Döngüsü
- Grup Döngüsü
- Bölüm Döngüsü

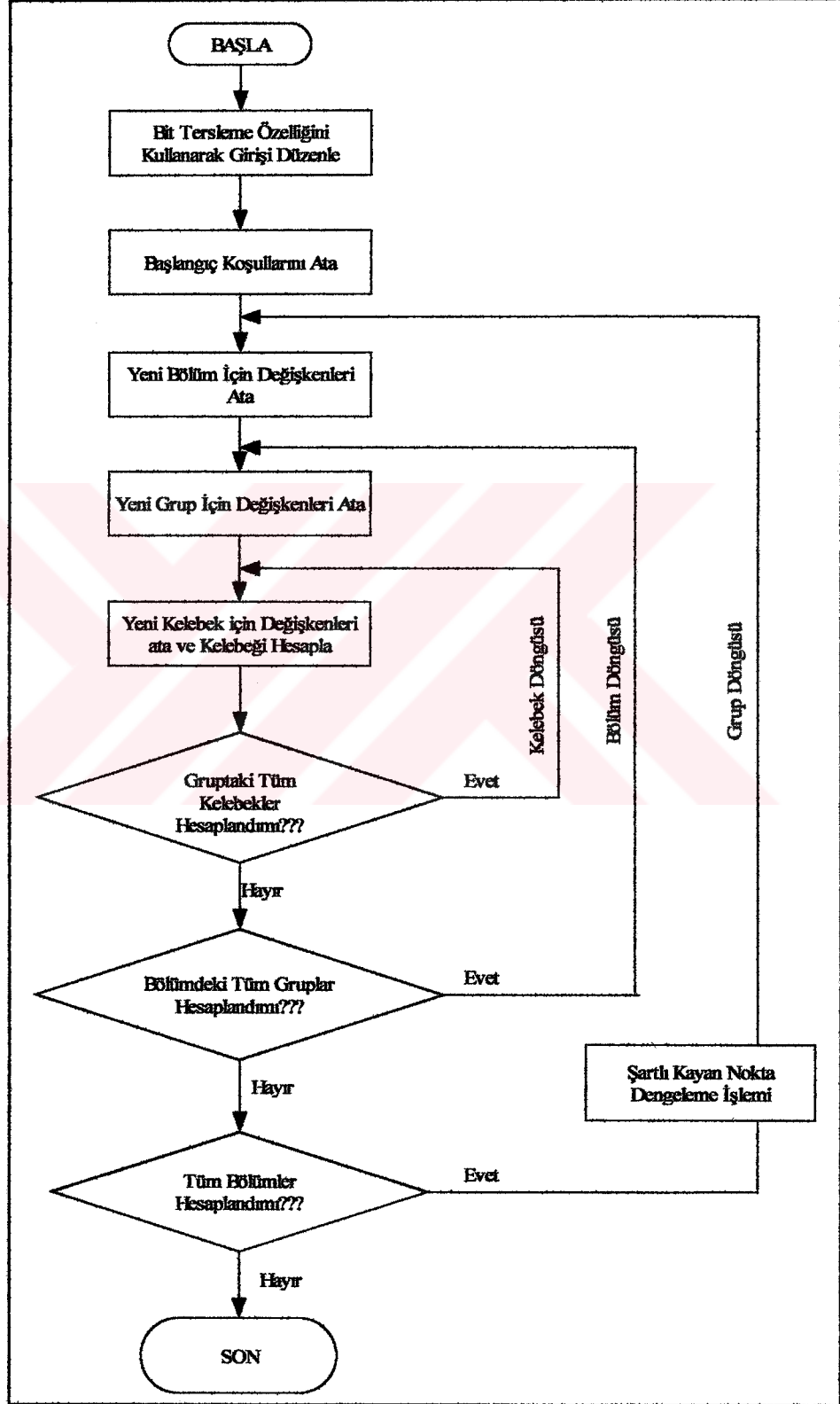
Grup döngüsüne girmeden öncede o grupta kaç adet hesaplanacak kelebek olduğu, kelebeklerdeki düğümler arası boşluğun ne olduğu girilir. Şekil 4.3 da bit terslenmiş girişler uygulanan DIT FFT algoritmasının özet karakteristikleri görülmektedir.

	Bölüm 1	Bölüm 2	Bölüm 3	Bölüm $\log_2 N$
<i>Grup Sayısı</i>	$N/2$	$N/4$	$N/8$	1
<i>Herbir Gruptaki Kelebek Sayısı</i>	1	2	4	$N/2$
<i>Kelebeğin İki Girişi Arası Boşluk</i>	1	2	4	$N/2$
<i>Twiddle Faktör Üstelleri</i>	$(N/2)k,$ $k=0$	$(N/4)k,$ $k=0, 1$	$(N/8)k,$ $k=0, 1, 2, 3$	$k,$ $k=0 \dots N/2-1$

Şekil 4.3 N noktalı DIT FFT karakteristikleri (Vinay, 1995)

Bu şekilde bir tablo yardımıyla FFT programı yapmak daha da kolaylaşmaktadır. Grup Sayısı, her bir Gruptaki Kelebek Sayısı, Kelebeğin İki Girişi Arasındaki Boşluk ve Twiddle Faktör

Üstleri Veri Belleğindeki bellek gözlerinde değişkenler olarak saklanır. Her yeni grup döngüsüne girildiğinde değiştirilir. Her bir Gruptaki Kelebek Sayısı ve Kelebeğin İki Girişi Arasındaki Boşluk değişkenleri her yeni grup döngüsüne girildiğinde iki katına çıkarılır.



Şekil 4.4 Radix-2 DIT FFT genel akış diyagramı

### 4.1.2 Zamanda İndirgeme

Zamanda İndirgeme (Decimation-in-time), işaretin zaman düzleminde onu oluşturan ve işlem kolay olan küçük parçalara indirgenmesi demektir. Vinay, (1995)'de belirtildiği üzere N noktalı zaman düzlemindeki giriş  $x(n)$ 'nin frekans cevabı,  $X(k)$  hesaplanırsa;

$$X(k) = \frac{1}{N} \sum_{n=0}^{N-1} x(n)W_N^{Nk} \quad k = 0,1,2,\dots,N-1 \quad (4.1)$$

Bu N noktalı DFT'yi alır ve tek ve çift numaralı örnekler olmak üzere N/2 noktalı 2 tane DFT'ye bölersek;

$$X(k) = \frac{1}{N} \sum_{m=0}^{N/2-1} x(2m)W_N^{2mk} + \frac{1}{N} \sum_{m=0}^{N/2-1} x(2m+1)W_N^{(2m+1)k}$$

$$\text{ÇİFT NUMARALI ÖRNEKLER} \quad \text{TEK NUMARALI ÖRNEKLER} \quad (4.2)$$

$$x_1(m) = \text{Çift numaralı örnekler} = x(2m)$$

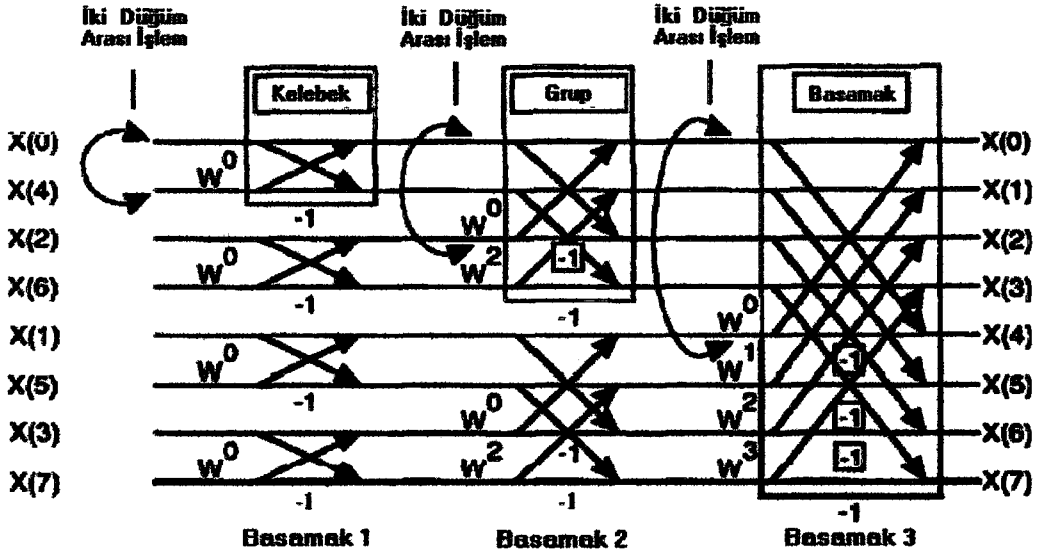
$$x_2(m) = \text{Tek numaralı örnekler} = x(2m+1)$$

$$X(k) = \sum_{m=0}^{N/2-1} x_1(m)W_{N/2}^{mk} + \sum_{m=0}^{N/2-1} x_2(m)W_{N/2}^{(2m+1)k} \quad (4.3)$$

Eşitlikten de görüldüğü üzere, tek ve çift numaralı örneklerin "Twiddle Faktörleri" aynı değildir. Eşitliği basitleştirmek amacıyla, DFT'deki  $W_N^{2mk} = W_{N/2}^{mk}$  eşitliğine dayanarak twiddle faktörü toplam dışına alırsak;

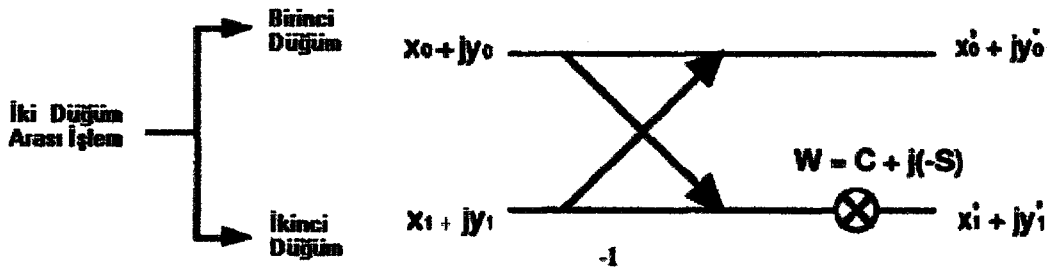
$$X(k) = \sum_{m=0}^{N/2-1} x_1(m)W_{N/2}^{mk} + W_N^k \sum_{m=0}^{N/2-1} x_2(m)W_{N/2}^{mk} \quad (4.4)$$

Artık elimizde N noktalı DFT'den daha kısa sürede hesaplanabilecek 2 tane N/2 noktalı DFT vardır. Bu indirgeme işlemi, 2 noktalı DFT'lere ulaşmıncaya kadar sürdürülür.



Şekil 4.5 8 noktalı DIT FFT (Vinay, 1995)

Yukarıdaki şekilde 8 noktalı bir DIT FFT gösterilmiştir. Girişler, çıkış verilerinin düzgün bir sırayla elde edilmesi için bit-terslenerek girilmiştir. Bit tersleme gereksinimi, FFT'nin veri dizilerini sürekli indirgemesinin doğal bir sonucu olarak ortaya çıkmaktadır. Programın basitleşmesi için twiddle faktörlere bit-terslenmesi yapılmamıştır. Her bir ok çifti bir butterfly (kelebeği) 'ı temsil etmektedir. Dikkat edilecek olursa FFT hesaplamasının tümü grup ve basamaklar diye farklı farklı organize edilen kelebeklerden oluşmaktadır. Her bir kelebek (2-radix DIT FFT'de) aşağıda gösterildiği üzere birinci ve ikinci düğüm olarak adlandırılan girişlere sahiptir.



Şekil 4.6 Kelebek yapısı (Vinay, 1995)

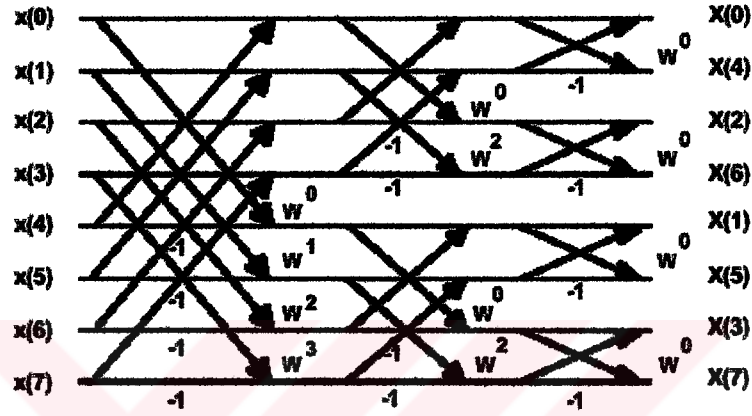
Buradaki x ve y değişkenleri sırasıyla reel ve sanal kısımlarını ifade etmektedir. Her bir kelebeğin grup ve basamağına bağlı olarak ayrı birer "twiddle faktörü" vardır. Twiddle faktörler için de aynı mantık geçerlidir çünkü;

$$W_N = e^{-j2\pi/N} = \cos(2\pi/N) - j\sin(2\pi/N) \quad (4.5)$$

şeklinde. Bu çalışmada yazılan programda da twiddle faktör değerleri bellekte “cosine” ve “-sine” biçiminde depolanmıştır.

#### 4.1.3 Frekansta İndirgeme

Frekansta İndirgeme (Decimation-in-frequency), yönteminde, işaret frekans düzleminde indirgenir ve zaman düzleminde tekrar birleştirilir. Burada da işlem 2 noktalı DFT’leri elde edinceye kadar sürdürülür.



Şekil 4.7 8 noktalı DIF FFT

Yukarıdaki şekilde 8 noktalı bir DIF FFT görülmektedir. Dikkat edilecek olursa, girişler düzgün bir sırayla dizili, ancak çıkışlar bit terslenmiş (scrambled) biçimdedir. DIF FFT’de de girişler bit-terslenmiş olarak verilir, çıkışlar düzgün sırada alınabilir ancak böyle bir durumda twiddle faktörlere de bit-terslenmesi gerekeceğinden ve bu da programı zorlaştıracığından genellikle tercih edilmemektedir.

#### 4.1.4 DIF FFT - DIT FFT tercihi:

Bu iki yaklaşım rutini arasındaki tek fark; DIF normal sıralı veri ile işe başlar ve bit terslenmiş çıkış üretir. DIT ise, tam zıddı olarak bit terslenmiş giriş ile başlar ve normal sıralı çıkış üretir. Bu sebeple işlem basitliği bakımından, ileri yönlü dönüşümlerde (in forward transforms) DIT, ters dönüşümlerde (in inverse transforms) ise DIF kullanılması daha mantıklıdır.

#### 4.1.5 DIF veya DIT Seçildikten Sonra FFT Gerçeklenmesi:

FFT, için uygulamaya göre ister DIF ister DIT seçilsin gerçekleştirilmede birçok yöntem mevcuttur ancak biz burada kısaca birkaç tanesinden bahsedeceğiz. Bunlar; Radix2, Radix4, Split Radix, Hızlı Hartley Dönüşümü (Fast Hartley Transform), Hızlı Fourier Dönüşümü

(Quick Fourier Transform) ve Zaman - Frekans Düzleminde İndirgeme (Decimation-in Time Frequency)'dir. Radix2 ve Radix4 yaklaşımları, N-noktalı DFT hesaplamasını, sırasıyla 2 veya 4 noktalı DFT'lere indirgeyerek gerçekleştirirler. Örnek sayısı 4'ün katı olduğunda Radix4 tabanlı kelebekteki hesaplamalar, Radix2 tabanlı kelebektekiye göre daha az karmaşık çarpma içerir ve verimlilik daha yüksek olur. Bu avantajdan yararlanmak için, Split-radix her iki ayrıştırmayı da kullanır. Hızlı Hartley Dönüşümü, DFT'deki karmaşık üstel terimi, gerçel değişkenleri kullanan bir çekirdek ile değiştirerek hesaplama yoğunluğunu azaltır. Hızlı Fourier Dönüşümü, karmaşık hesaplamaların sayısını azaltmak için sinüs ve kosinüs terimlerinin simetrikliğini kullanır. Zaman - Frekans Düzleminde İndirgeme algoritması, Radix2 tabanlı DIT ve DIF FFT'nin avantajlı yanlarını kullanır. DIT algoritmasında karmaşık işlemlerin yoğunluğu son çevrimlere doğrudur. DIF algoritmasında ise başlardadır. DITF ise işleme DIT ile başlar son çevrimde DIF'e anahtarlanır. Bu yöntemle karmaşık işlem sayısı azalırken çevirme terimleri işin içine girer. Kullanım yerlerine göre düşünüldüğünde farklı algoritmaları seçmek yerinde olacaktır. Seçimde kullanılacak 2 kritere göre aşağıda karşılaştırma tabloları verilmiştir.

Çizelge 4.1 Algoritmaların örnek sayısına göre  $\mu$ sn cinsinden karşılaştırılması (Garcia, 1999)

ALGORİTMA	ÖRNEK SAYISI				
	64	256	1024	4096	16384
RAD-2	238	952	3952	17857	77714
RAD-4	191	762	3476	14714	60333
SRFFT	238	905	3810	17429	74524
FHT	48	286	1333	7143	31905
QFT	143	762	3476	15952	78000
DITF	238	1000	4191	18714	80333

Çizelge 4.2 Örnek sayısı 1024 iken algoritmaların byte olarak bellek kullanımı (Garcia, 1999)

ALGORİTMA	BELLEK
RAD-2	24616
RAD-4	8344
SRFFT	24656
FHT	32832
QFT	49152
DITF	24616

Bu çalışmada, FHT tercih edilmemiştir çünkü kayan noktalı (floating point) işlem yapan bir DSP gerektirmektedir bu da maliyeti arttırmaktadır. 2181 DSP'si sabit noktalı (fixed-point) bir DSP olduğundan 2 alternatif söz konusu olabilir; Radix2 ve Radix4 . Radix4 algoritmasında örnek sayısı arttıkça veri adresleme mekanizması karmaşıklaştığından Radix2 tercih edilmiştir.

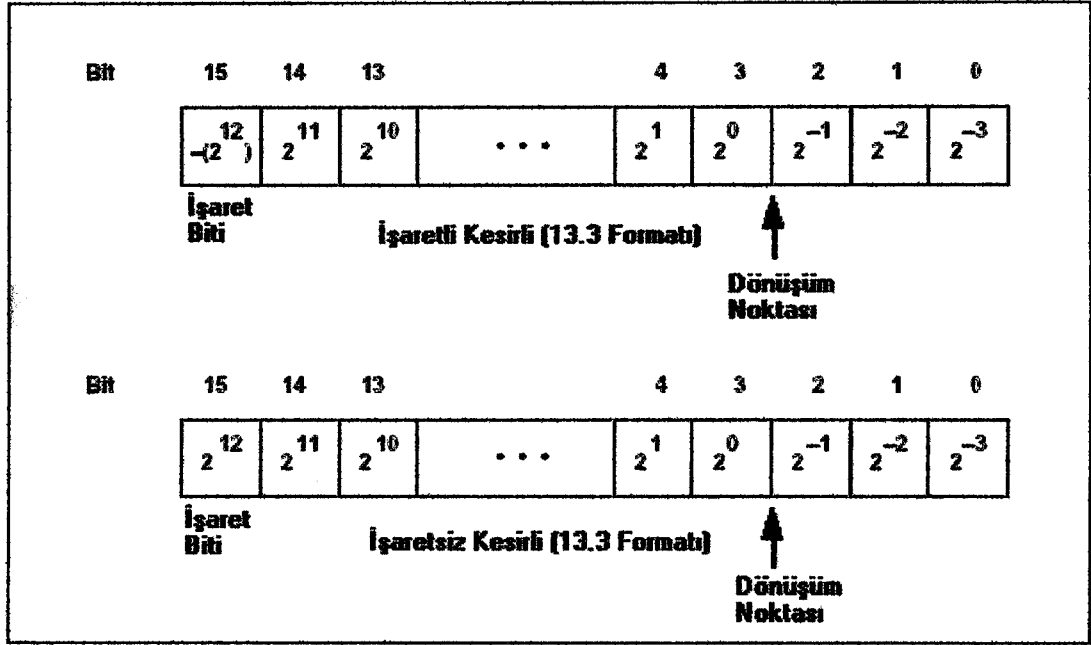
## 4.2 Matematiksel İşlemler

DSP programı yazılırken kullanılan matematiksel işlemler tanımlanacaktır.

### 4.2.1 Sabit Nokta Aritmetiği

İkili sayılar genellikle üç ana kısımdan oluşmaktadır. İşaret biti, dönüşüm noktası ve sayının değeri. İşaret biti sayının negatif veya pozitif oluşunu belirtir. Dönüşüm noktası sayının tamsayı ve kesirli kısmını ayıran noktadır.

Sabit nokta aritmetiğinde sayıların gösteriminde a.b formatı kullanılır. a harfi tamsayı olan bitlerin sayısını b ise kesirli bitlerin sayısını göstermektedir. Örneğin 4.12 formatı denildiğine soldan başlayarak 1 işaret biti 3 bit tamsayı kısım ve 12 bit kesirli kısım olarak algılanır. Sayı ise 16 bit uzunluğundadır. Genellikle Sabit Noktalı DSP işlemcilerinde ikiye tımleyen aritmetiği ve tamsayı aritmetiği kullanılmaktadır. İşlemciler ya 1.15 formatında kesirli işlem yapmakta ya da tamsayı işlem yapabilmektedir. İkiye tımleyen aritmetiğinin kullanılma sebebi ise bu aritmetikte sadece bir adet sıfır bulunmasıdır. Oysa bire tımleyen aritmetikte hem pozitif hem de negatif sıfır bulunmaktadır. Bu da karışıklığa yol açmaktadır.



Şekil 4.8 16 Bitlik sayının işaretli ve işaretsiz gösterim biçimleri (Vinay, 1995)

#### 4.2.2 Kayan Nokta Aritmetiği

Sabit nokta aritmetiğinde Dönüşüm Noktasının yeri sabit olduğundan temsil edilebilecek sayılar ya 1.15 formatında (-1 ile +1 arası) yada 16.0 formatında yani tamsayı formatındaki sayılardır. Örneğin 4.12 formatında bir sayıyla işlem yapmak için çeşitli kaydırma işlemleri, Normalizasyon ve Denormalizasyon işlemleri yapmak gerekmektedir. Oysa Kayan Nokta Aritmetiğinde Dönüşüm Noktasının yerini belirtmek içinde sayının sonunda bitler ayrılmıştır. Örneğin IEEE 754 standardında kayan noktalı sayı formatında sayılar 32 bitle temsil edilmektedir. 32 bitlik kelime soldan sağa 1 işaret biti 8 üstel biti 23 bit ise tamsayı kısmını belirlemek için kullanılmaktadır.

#### 4.2.3 Taşma Engelleme Yöntemleri

FFT hesabını Sabit Noktalı İşlemcilerle yaparken taşmayı engellemek için üç adet yöntem geliştirilmiştir.

#### 4.2.4 Giriş Verisini Dengeleme Yöntemi

Bu yöntemde FFT'nin kaç noktalı olduğuna bakılarak ve her bir kelebekten 2 bit taşma olacağı varsayılarak toplam koruma bitleri önceden hesaplanır. Örneğin 32 noktalı bir FFT hesabında Bölüm sayısı  $\log_2 32 + 1 = 6$  olduğu için ardarda iki kelebekte iki bitten fazla taşma olamayacağı varsayımıyla toplam 6 bit taşma olabilir. Dolayısıyla giriş dizisinde 6 adet koruma biti bırakılarak (Girişteki her değeri 6 bit sağa kaydırmak anlamına gelir) FFT hesabı yapılır. Fakat FFT'nin nokta sayısı büyüdükçe koruma bitlerinin de sayısı artacak ve kayıplar

fazla olacaktır. Bu yöntem hızlı olmasına rağmen kayıplar fazladır. Fakat düşük noktalı FFT'lerde tercih edilebilir.

#### **4.2.5 Şartsız Blok Kayan Nokta Dengeleme Yöntemi**

Bu yöntemde önce giriş dizisinde taşma olmaması için 2 koruma biti bırakılır (Giriş verisi 2 kez sağa kaydırılır). Daha sonra her bölümün çıkışında oluşan dizi şartsı olarak 2 ye bölünür. Bu yöntemde kayıplar Giriş Verisini Dengeleme Yöntemi'nde oluşan kayıplarla aynıdır. Fakat sonuç daha doğrudur çünkü girişte girilen veri daha az kayıplıdır. Bu yöntem ilk yöntemle göre daha yavaştır ama daha doğru sonuç verir. Düşük noktalı FFT'lerde tercih edilebilir.

#### **4.2.6 Şarh Blok Kayan Nokta Dengeleme Yöntemi**

En çok kullanılan yöntemdir. Çünkü kayıpların en az olduğu yöntemdir. Bu yöntemde giriş dizisinde iki adet koruma biti alınır. Bir gruptaki bütün kelebekler hesaplandıktan sonra tüm dizide koruma bitlerine taşma olup olmadığına bakılır. Bu işlem EXPADJ komutuyla gerçekleştirilir. Bu komut yardımıyla çıkış dizisinin blok üsteli bulunur. Eğer koruma bitlerinden taşma olmuşsa taşma sayısı kadar kaydırma yapılarak koruma bitleri boşaltılır. Toplam kaydırma sayısı da blk\_exponent bellek bölgesinde saklanır. Bu yöntem diğer yöntemlere göre yavaş olmasına rağmen kayıpları az olduğu için en çok tercih edilen yöntemdir. Özellikle nokta sayısı fazla olan FFT hesaplarında yüksek doğruluk sağlamaktadır.

#### **4.2.7 Bit Tersleme Özelliği**

Zamanda İndirgemeli Fourier Dönüşümünde giriş dizisi bit terslenmiş şekilde FFT programına verilmelidir. İleride bahsedilecek olan Frekansta İndirgemeli Fourier Dönüşümünde (DIF FFT) ise tam tersi bir durum söz konusudur. Bu sefer giriş dizisi sıralı fakat çıkış dizisi düzensizdir. Bunun içinde çıkıştaki diziyi bit tersleme özelliğini kullanarak düzenli hale getirilmelidir.

Hızlı Fourier Dönüşümünün bu özelliği ADSP-21xx işlemcilerinde donanım olarak sağlanmıştır. ENA BIT\_REV komutuyla DAG2'de adresleri ters olarak okumasına neden olur. Bu şekilde tüm veriler bit\_terslenmiş bir şekilde inplacereal bellek bölgesine kaydedilir. İşlem bittikten sonra DIS BIT\_REV komutuyla tekrar normal moda döndürülür.

<b>Örnek Giriş Dizisi</b>	<b>Sıralı Yerleşim</b>		<b>Bit Terslenmiş Yerleşim</b>		<b>Örnek Bit Terslenmiş Sıralama</b>
	Desimal	İkili	Desimal	İkili	
x(0)	0	000	0	000	x(0)
x(1)	1	001	4	100	x(4)
x(2)	2	010	2	010	x(2)
x(3)	3	011	6	110	x(6)
x(4)	4	100	1	001	x(1)
x(5)	5	101	5	101	x(5)
x(6)	6	110	3	011	x(3)
x(7)	7	111	7	111	x(7)

Şekil 4.9 Bit tersleme özelliğine bir örnek (Vinay, 1995)

#### 4.2.8 Blok Kayan Nokta Dengeleme İşlemi

Blok Kayan Nokta Dengeleme İşlemi Sabit Nokta İşlemcilerin hesap yeteneğini güçlendirmek için kullanılan bir yöntemdir. Sabit noktalı işlemciler sadece 1.15 formatında yada tamsayı formatında işlem yapabilme kabiliyetine sahip oldukları için diğer sayı formatlarında işlemleri yapabilmek için kaydırıcıya ihtiyaç duyarlar. Kaydırma işlemini hızlı bir şekilde yapabilmek için DSP'de KAYDIRICI (SHIFTER) Bloğu bulunmaktadır. Kaydırıcının komutları yardımıyla tek saat darbesinde istenildiği kadar kaydırma işlemi yapılabilmektedir. Blok Kayan Nokta işlemi bir dizi bloğunun ortak üstelini bularak SB yazmacında blok üstelini tutar. EXPADJ komutu bir sayının üstelini bulmak için kullanılır. Ancak sadece üstelini bulmakla kalmaz SB yazmacındaki bir önceki değerle karşılaştırarak eğer eski değerden daha büyükse SB yazmacına yeni bulunan üstel değerini atar. Değilse SB yazmacındaki eski değer korunur. Bu şekilde tüm bir dizinin ortak üsteli bulunmuş olur. Ancak kaydırma işlemi yapmaz sadece blok üstelini bularak SB yazmacına atar.

Radix-2 kelebeği 2 bit taşma ihtimaline sahip olduğundan FFT'de bir bölüm hesaplandıktan sonra SB yazmacına -2 değeri atanır. Bütün çıkış dizisi tek tek kontrol edilir. Aşağıda Blok Üsteli Bulma ile ilgili bir örnek verilmiştir. (SB nin ilk değeri=-2)

<i>Çıkış Dizisi</i>	<i>EXPADJ Komutunun uygulanmasından sonra SB yazmacının değeri</i>
1111 0000 0000 0000	SB= -2
1110 0000 0000 0000	SB= -2
1100 0000 0000 0000	SB= -1
1110 0000 0000 0000	SB= -1
1000 0000 0000 0000	SB= 0

Şekil 4.10 Üst ayarlanması (Vinay, 1995)



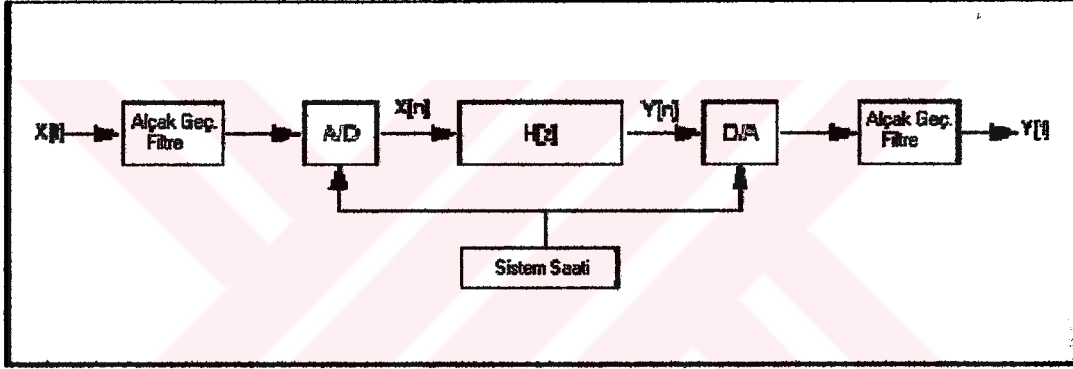
## 5. HARMONİK ANALİZ DONANIM VE YAZILIMI

Gerçekleştirilen sistem ve gerekli donanım, yazılım ayrıntılı olarak ele alınacaktır.

### 5.1 Sistemin Temel Yapısı

Temeli entegral hesabına dayanan Fourier dönüşümü, sürekli zamandan ayrık zamana dönüştürülüp, dolayısıyla entegrasyon da toplam ve çarpımlar haline getirildiğinde sayısal sistemlere kolaylıkla uygulanabilmektedir. Bu sayede örneksel bir işaret, A/D çeviriciden geçirilerek, örnek değerler kaydedilmek suretiyle, yinelenen sürekli toplam ve çarpımlarla işaretin frekans bileşenleri rahatlıkla bulunabilir.

### 5.2. Sistemin Sayısal İşaret İşleme Mantığı



Şekil 5.1 Sistemin sayısal işaret işleme mantığı blok diyagramı

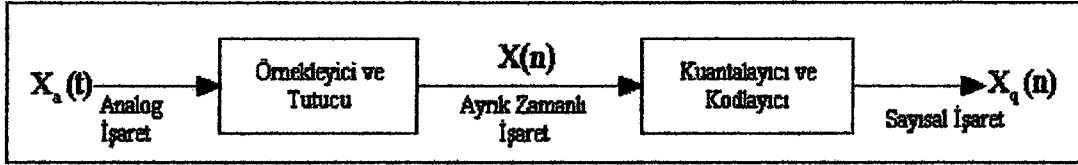
Bir sayısal işaret işleme sistemi; girişte bir A/D çevirici, çıkışta bir D/A çevirici, bir geri besleme bloğu ve denetleyiciden oluşmaktadır. Giriş ve çıkışta ise örtüşme olayının ortaya çıkmaması için alçak geçiren filtre bulunmaktadır.  $H(z)$  ile temsil edilen blok ise denetleyiciyi göstermektedir. Bu denetleyici, bir mikroişlemci, mikrodenetleyici ya da Sayısal İşaret İşleyici (DSP) olabilir. Bu çalışmada, denetleyici olarak, Sayısal İşaret İşleyici (DSP) kullanılmıştır.

#### 5.2.1 Örneksel-Sayısal Dönüşüm

Örneksel işaretin doğrudan örneksel olarak işlenmeyip, sayısal işaret işleyicide işlenmesinin birtakım nedenleri vardır Bunlar:

- Sayısal programlanabilir sistemlerin, programdaki bazı değişikliklerle sayısal işaret işleme operasyonunu düzenleyebilme esnekliğine sahip olması.

- Sayısal sistemlerin örneksel sistemlere oranla daha fazla gürültüye karşı doğruluk derecesinin yüksek olması. Örneksel sistemlerde kullanılan elemanların toleranslarının kontrolünün zor olması.
- Sayısal sistemlerle gerçekleştirilen devrelerde, aynı işi yapacak olan örneksel sisteme göre çok daha az yer kaplaması.
- Sayısal sistemlerin programlanabilir olması ve bellekli olması.

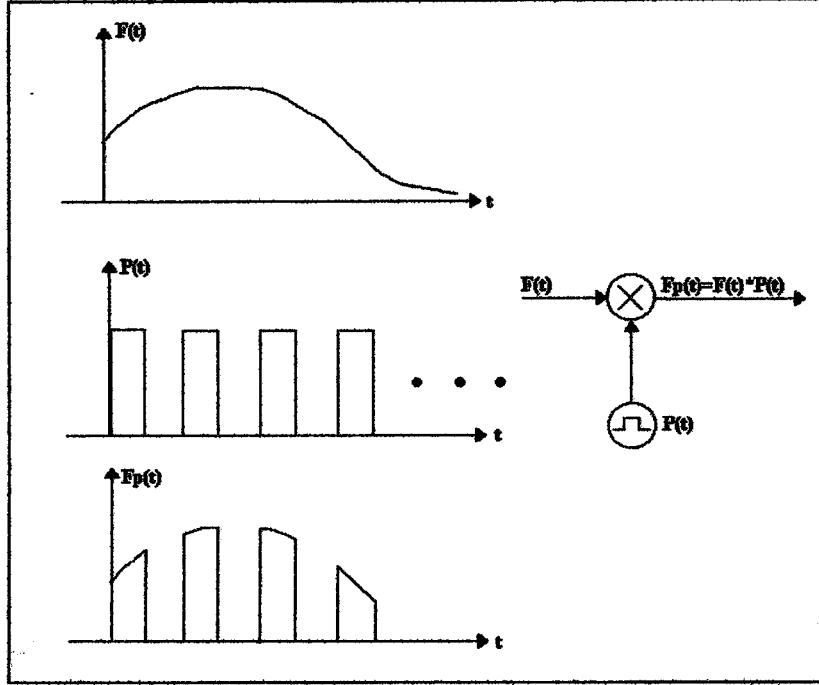


Şekil 5.2 Örneksel-sayısal dönüşümün modellenmesi

Örnekselden sayısal çevrim iki basamakta gerçekleştirilir;

#### 5.2.1.1 İşaretin Örnekleme

Örneksel işaretin örneklemeinde pek çok yöntem vardır. Burada pratikte en çok kullanılan periyodik örnekleme tipi ile ilgilenecektir.



Şekil 5.3 İşaretin darbe katarıyla çarpımı (Kayran, 1990)

Şekilden de görüleceği üzere örneksel-sayısal çevriminin ilk aşaması örneksel işareti periyodik bir darbe dizisi ile çarpmaktır. Darbe dizisinin frekansı örnekleme frekansına karşılık gelmektedir. Mesaj işareti periyodik bir darbe dizisi ile çarpıldığında bölünmüş yani örneklenmiş olur. Fakat bu örneksel bilgiyi sayısala çevirmek için yeterli değildir. A/D çeviricinin girişine bir seferde uygulanan işaretler sabit DC seviyede olmalıdır. Bunun için örneklenen işaretin ilk değerinin tutularak örnek değeri sonuna kadar aynı seviyede kalması gerekmektedir. Bu işlem yapıldıktan sonra işaret A/D çevirici girişine verilmeye hazır hale getirilmiş olmaktadır.

Bu işleme Darbe Genlik Modülasyonu (PAM) adı verilmektedir. Bu modülasyon türünün başka bir avantajı da aynı anda birden fazla bilginin gönderilmesine imkan vermesidir. Örneklenen işaret şekilden de görüleceği gibi asıl işaretin bölünmüş şeklidir ve darbe katarıyla çarpıldığında arada boşluklar oluşmaktadır. Eğer darbe işaretinin darbe boşluk oranı daha da küçük tutulursa aradaki boşluklar daha da büyüyecektir. Bu boşluklarda başka mesaj işaretlerinin de gönderilmesine imkan tanımaktadır. Anahtarlama yöntemiyle aralara başka işaretlerde konularak aynı anda birden fazla mesajın iletilmesi olayına Zaman Bölmeli Çoğullama (TDM) adı verilmektedir.

### 5.2.1.2 Kuantalama ve Kodlama

Darbe genlik modülasyonunda örneklenmiş değerler belirli kuantalama seviyelerine yuvarlatılmadan iletilmektedir. Ancak, bu işlem işaretin gürültüye olan bağışıklığı açısından bir yarar sağlamayacaktır. Bunun yerine, işaret genliğini belirli kuantalama seviyelerine yuvarlatmak

ve her kuantaya seviyesine uygun bir kod kelimesi karşı düşürmek daha uygun olmaktadır.  $X(t)$  işaretinin maksimum ve minimum genlikleri  $A_{max}$  ile  $-A_{max}$  arasında değişiyorsa ve bu aralıkta değişen genlik değerleri  $Q=2^n$  adet eşit kuantaya seviyesine bölünmek isteniyorsa, kuantalama aralığı veya adımı

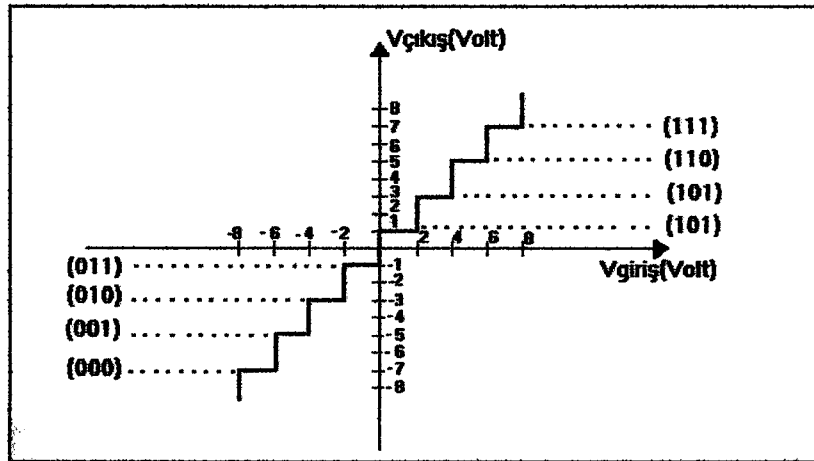
$$a = \frac{2A_{max}}{2^n} \quad (5.1)$$

olarak tanımlanmaktadır. Kuantalama işleminde örnek değerlerin bulunduğu dilim belirlenir. Örneğin, -8 ve +8 volt arasında değişen bir  $x(t)$  işaretini ele alalım. Bu aralık 8 kuantaya seviyesine ayrılırsa, kuantalama aralığı  $a=(16/8)=2$  birim olacaktır.

#### 5.2.1.2.1 Düzgün Kuantalama

Şekil 5.4'de gösterilen türden kuantalamaya düzgün kuantalama adı verilir. Düzgün kuantalama işlemi olasılık yoğunluk fonksiyonu düzgün olan işaretler için kullanılır. Bunun anlamı işaretin (+,-) sınırları arasında her voltaj değerinin gelme oranının aynı olması demektir. 8 seviyeli düzgün bir kuantalayıcıya ilişkin giriş-çıkış eğrisi Şekil 5.5'de gösterilmiştir.

-3, -2, -1, -0, +0, +1, +2, +3'deki 8 kuantaya seviyesi sırasıyla  $M_0, M_1, M_2, \dots, M_7$  olarak simgelenmektedir. Bu simgeler ikili kod kelimeler kullanılarak kodlandırılır. Bu kodlama işleminde, üretilecek kod kelime uzunluklarının en kısa uzunlukta olması arzu edilir. Verilen örnekte 8 kuantaya seviyesi için  $8=2^3$  olduğundan, kodlamada kelime uzunluğu  $n=3$  olacaktır.



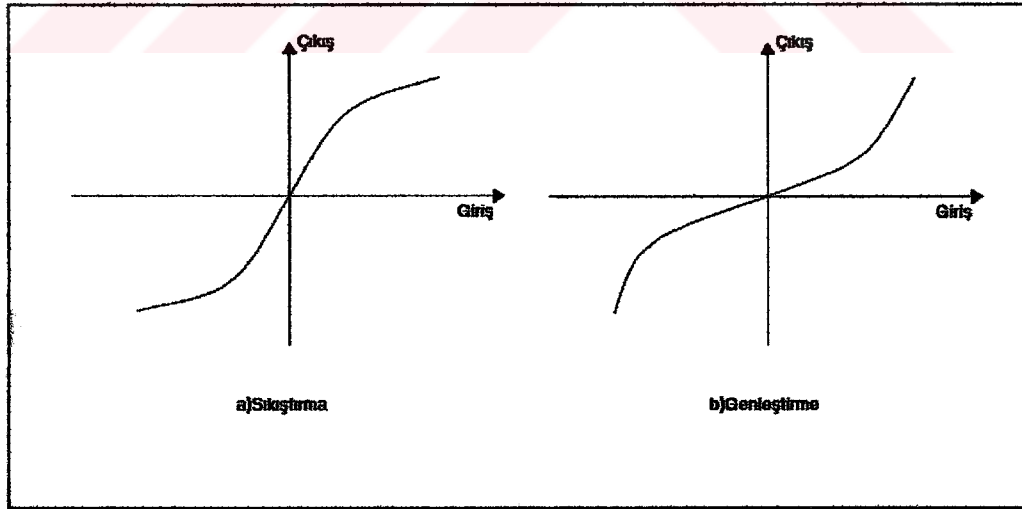
Şekil 5.4 Düzgün kuantalama eğrisi (Vinay, 1995)

### 5.2.1.2.2 Düzgün Olmayan Kuantalama

Ses işaretlerinin istatistikleri incelendiğinde küçük genliklere daha sık rastlandığını görmekteyiz. Oysa, yukarıda küçük işaretlerde kuantalama gürültüsünün rahatsız edici boyutlarda olacağını göstermiş bulunmaktayız.

Bu gürültüyü azaltmak için başvurulacak ilk yöntem, adım büyüklüğünün azaltılması veya dilim sayısının artırılmasıdır. Ancak, bu durumda her bir örneği göstermek için kullanılması gereken bit sayısı artacağından bu yöntem her zaman uygun ve ekonomik değildir. Diğer taraftan, çok seyrek olarak ortaya çıkan yüksek genlikli işaretler için gereksiz yere bir miktar dilim ayrılmış olacaktır. Eğer en büyük genliği küçük tutarsak, bu defa da kırılmalar meydana gelecektir.

Bu mahsurları ortadan kaldırmak için işaretlerin olasılık yoğunluk fonksiyonlarının analizleri yapılmış ve bu olasılık yoğunluk fonksiyonlarını düzgün hale getirebilmek için A ve  $\mu$  tipi sıkıştırma eğrileri standart olarak kabul edilmiştir. Olasılık yoğunluk fonksiyonu düzgün olmayan işaretler için önce Sıkıştırma işlemi yapılarak olasılık yoğunluk fonksiyonu düzgün hale getirilebilir. Daha sonra bu işaret düzgün kuantalama işlemine tabi tutularak kodlanabilir. Alıcı tarafta ise Genleştirme işlemine tabi tutularak kod çözülür. Bu şekilde işareti göndermek için gerekli bit sayısından tasarruf edilmiş olur.



Şekil 5.5 Sıkıştırma ve genleştirme eğrileri (Vinay, 1995)

Bu işlemler DSP işlemcisinin seri portunda donanımsal olarak yapılabilmektedir. ADSP işlemcisinin seri portunun kontrol yazmaçlarındaki uygun bitler set edilerek A tipi veya  $\mu$  tipi sıkıştırma desteklenebilmektedir. Eğer gönderim yapılacaksa 14 bitlik bilgi Seri Porta gönderildiğinde 8 bitlik sıkıştırılmış veriye çevrilerek iletilebilmektedir.

### 5.2.1.2.3 A Tipi Sıkıştırma

A-eğrisi denklemleri olarak adlandırılan bir denkleme uygun sıkıştırma işlemidir.

$$F_a(x) = \begin{cases} \text{Sgn}(x) \cdot \left( \frac{A|x|}{1 + \ln(A)} \right) & ; 0 \leq x \leq 1/A \\ \text{Sgn}(x) \cdot \left( \frac{1 + \ln(A|x|)}{1 + \ln(A)} \right) & ; 1/A \leq x \leq 1 \end{cases}$$

(5.2)

Sesin doğallığını koruyabilmesi için küçük genliklerde eğrinin doğrusal olması ve büyük genliklerde ise logaritmik olması gerekir. A-tipi eğri (2.2) tanım gereği bu koşulu sağlamaktadır.

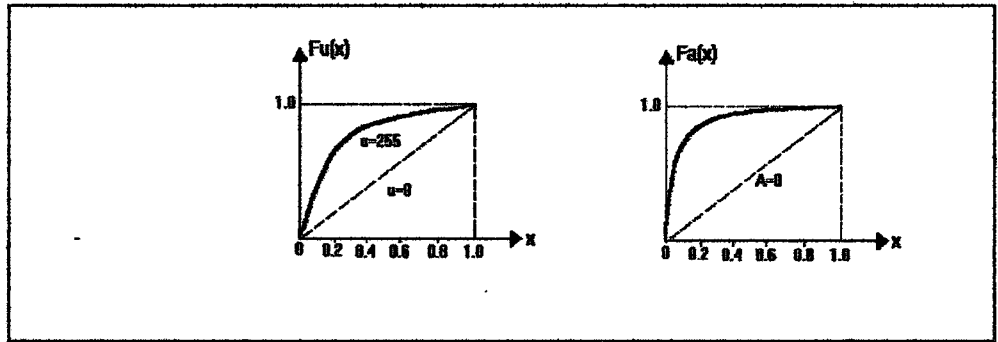
### 5.2.1.2.4 $\mu$ Tipi Sıkıştırma

$\mu$  -eğrisi denklemleri:

$$F_\mu(x) = \text{Sgn}(x) \cdot \frac{\ln(1 + \mu|x|)}{\ln(1 + \mu)}$$

(5.3)

olarak verilmektedir. Bu ifadede sıkıştırma oranı olarak  $\mu=255$  alınır. 7-bit kelime uzunluğundaki kuantalama da, kuantalama gürültü seviyesi  $-77\text{dB}$  olur. Oysa düzgün kuantalamada aynı kelime uzunluğu için kuantalama gürültü seviyesi  $-53\text{dB}$  olmaktadır. Yani 24dB'lik bir iyileştirme söz konusudur.



Şekil 5.6 A ve  $\mu$  tipi sıkıştırma eğrileri (Vinay, 1995)

### 5.3 Yazılım Ve Donanım Geliştirme Aşamaları

Geliştirme seti hakkında özet bilgi ve set üzerinde bulunan AD1847 CODEC tümleşik devresi hakkında bilgi verilecektir.

#### 5.3.1 ADSP-2181 İşlemcisinde Program Yazılması ve Derlenmesi

Program yazımı herhangi bir metin editöründe yapılabilir. Program yazımı yapılırken derleyici direktiflerine ve programdaki blokların yerleşim sırasına dikkat edilmelidir. Örneğin her program .MODULE direktifiyle başlayarak .ENDMOD direktifiyle bitmelidir. Programın başında .CONST direktifiyle programda yer alan sabitler tanımlanır. VAR direktifiyle programda bozulan bellek bölgeleri tanımlanır. INIT direktifiyle bellek bölgelerinin ilk koşulda alması gereken değerler tanımlanır. 0x0000 adresi RESET Vektörünü göstermektedir. Program bu adresten başlanarak yazılmaya başlanır. Ancak 0x0004 - 0x002F adresleri arası Kesme Servis Programı adresleri olduğundan programın reset vektör adresinde JUMP komutu kullanarak kesme vektör adreslerine atlayarak ana programa dallanması sağlanmalıdır. Kesme vektör adreslerinde her bir kesme için 4 byte'lık yer ayrılmıştır. Bu kesmeler eğer kullanılacaksa kesme servis programında ne yapacağı yazılır. Eğer kullanmayı düşünülüyorsa RTI (Kesmeden Geri Dön) veya NOP (İşlem Yapma) komutu kullanarak bu adresler doldurulabilir. Program yazıldıktan sonra .DSP uzantısıyla kaydedilir.

##### 5.3.1.1 Derleyicinin Çalıştırılması

Program yazılarak .DSP uzantısıyla kaydedildikten sonra derleyici çalıştırılarak programın derlenmesi gerekmektedir. ASM21 programı yardımıyla derleme yapılmalıdır. Eğer hata var ise derleyici bu aşamada hataları belirtmektedir. Örneğin fftproje.dsp ismi altında programın kaydedildiği varsayılırsa;

Asm21 fftproje -2181

yazılarak program derlenebilir. Bu aşamada derleyici .obj ve .lst dosyalarını oluşturur.-2181 deyimi ise yazılan programın 2181 işlemcisinin komut kurallarına uygun bir şekilde derlenmesi için gerekmektedir.

##### 5.3.1.2 Birleştiricinin Çalıştırılması

Derleyici tarafından oluşturulan nesne modüllerinden derlenmiş dosyayı hazırlar. EXE uzantılı dosyayı oluşturur. Bu dosya simülasyon ve geliştirme kitinde programın çalışması için yüklenmesi gereken dosyadır.

Ld21 fitproje -a adsp2181 -e proje

Yazıldığında eğer program hatasız yazılmışsa proje.exe adı altında bir derlenmiş bir dosya oluşturacaktır. Mimari tanımlama dosyası -a direktifiyle belirtilir. Bu örnekte mimari dosyası olarak adsp2181.ach dosyasının kullanılacağını belirtmektedir.

### **5.3.1.3 Simülatörün Çalıştırılması**

Program yazılıp derledikten sonra programın adım adım işleyişini görebilmek için simülatörün çalıştırılması gerekmektedir.

Sim2181 -a adsp2181 -e proje

Yazıldığında simülatörü adsp2181.ach mimari dosyasıyla ve proje.exe programını yüklemiş şekilde açılır.

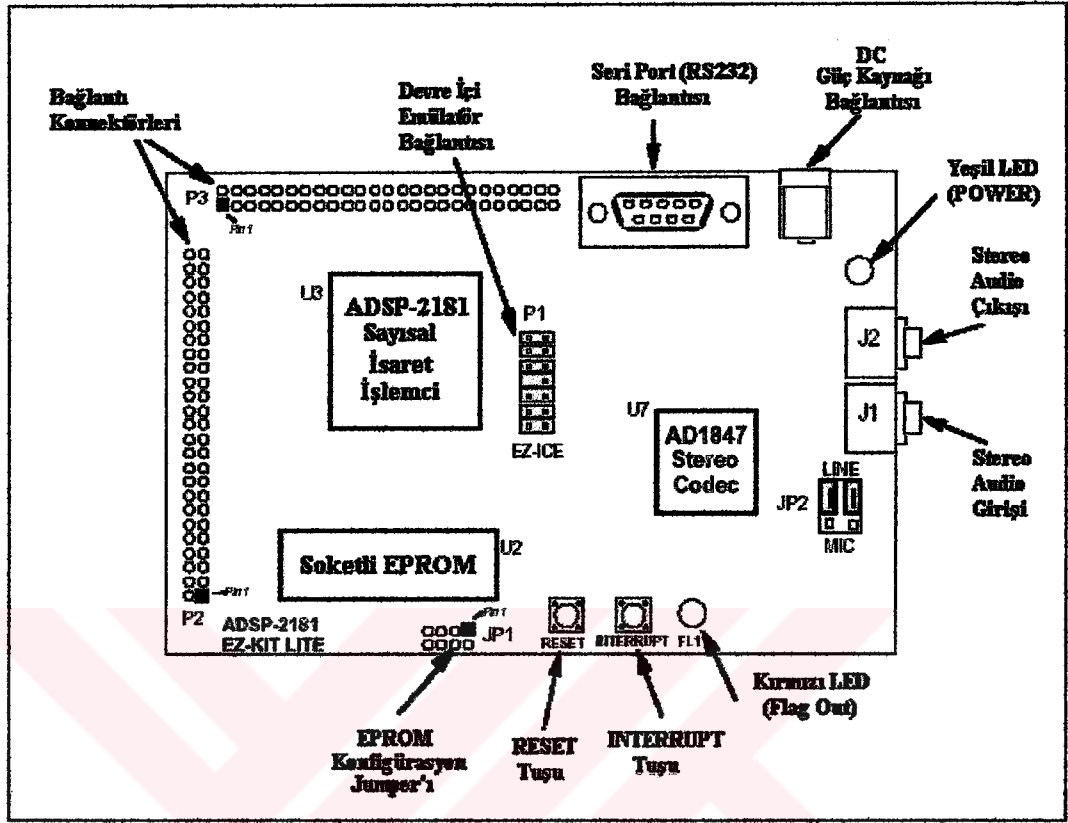
### **5.3.1.4 EPROM'a Program Yükleme**

Yazılan programın doğruluğu onaylandıktan sonra programın EPROM'a kaydedilerek çalıştırılması gerekmektedir. Bunun için Prom Splitter programı yardımıyla yazılan program 8-bit EPROM'a yüklenmeye uygun şekle getirilir.

Spl21 proje projeprom -loader -2181

Yazıldığında proje.exe dosyası alınarak projeprom.bnm dosyası oluşturulur. Bu dosya alınarak bir EPROM programlayıcı yardımıyla EPROM'a yazılır.

### 5.3.2 Geliştirme Seti Hakkında Özet Bilgi



Şekil 5.7 Geliştirme setinin blok şeması (Vinay, 1995)

Her mikroişlemci üreticisi firma kendi ürettiği mikroişlemci, mikrodenetleyici ya da DSP işlemcilerinde hızlı uygulama geliştirilmesini sağlayabilmek çeşitli olanaklar sunar. Bunlar Simulatör, Emülatör, Geliştirme Seti ya da yüksek seviyeli dillerde yapılmış derleyiciler (C derleyicileri, Basic Derleyicileri) olabilir. EZ-KIT Lite geliştirme seti de Analog Devices Firması tarafından üretilen ADSP-2181 işlemcisiyle hızlı uygulama geliştirmek amacıyla tasarlanmıştır. Tasarlanacak proje üzerinde geliştirme çalışmaları önce geliştirme seti yardımıyla yapılır. Proje yazılımı ve geliştirme tamamlandıktan sonra kullanıcı kendi baskı devresini tasarlama aşamasına geçer.

Geliştirme setinde bulunan devre elemanları minimum DSP sistemi kurmak için gerekli elemanlardır. Bu geliştirme seti yardımıyla, derlenen ve simülasyonu yapılan program DSP Program Belleğine yüklenerek çalıştırılır ve sonuçlar gözlenir. Geliştirme seti üzerindeki EPROM'da monitor programı bulunmaktadır. Bu monitor programı Seri Port üzerinden DSP ile bilgisayar arasındaki bağlantıyı sağlayan bir programdır. Monitor programı yardımıyla Program Belleği ve Veri Belleğine yükleme yapılabilir. Aynı zamanda Program Belleği ve

Veri Belleğindeki veriler okunabilir. İstenen adresten program tekrar başlatılabilir. Burada dikkat edilmesi gereken önemli bir husus kullanıcı DSP'ye kendi programını yüklediği zaman bu program EPROM'da bulunan monitor programı tarafından alt program olarak çağırılmaktadır.

Veri Belleği ya da Program Belleği okunmak istenirse RTS komutuyla monitor programına geri dönme zorunluluğu vardır. Tekrar programa geri dönmek istenirse Seri Porttan \$GO[adres(en ağırlıklı 8 bit)][adres(ağırlıksız 8 bit)] komutu gönderilebilir.

Ayrıca kullanıcı herhangi bir programlama dilinde yaptığı programla Seri Port üzerinden komutlar göndererek veya alarak kendi programını oluşturabilir.(Tüm Komutlar ASCII Karakterler olarak gönderilir ya da alınır). Örneğin DSP Veri Belleğini istenen bir adresten başlayarak belli uzunlukta okumak için şu komut gönderilir.

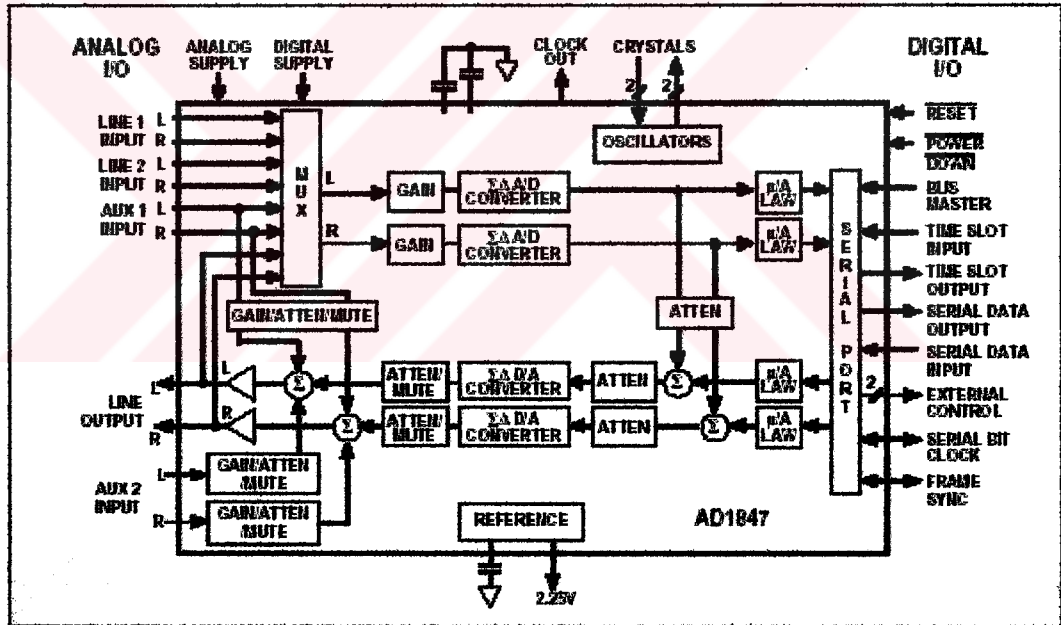
- **Gönderilen Komut:**\$UD[başlangıç adresi(en ağırlıklı 8 bit)] [başlangıç adresi(ağırlıksız 8 bit)][uzunluk(en ağırlıklı 8 bit)][uzunluk(ağırlıksız 8 bit)]
- **Alınan Bilgi:**[ağırlıklı 8 bit][ağırlıksız 8 bit].....

Komut seri porttan gittiği anda veriler sırayla gelmeye başlamaktadır. İletişim hızı 9600bps olmalıdır.

### 5.3.2.1 AD1847 Kodlayıcı ve kod çözücü tümleşik devresi

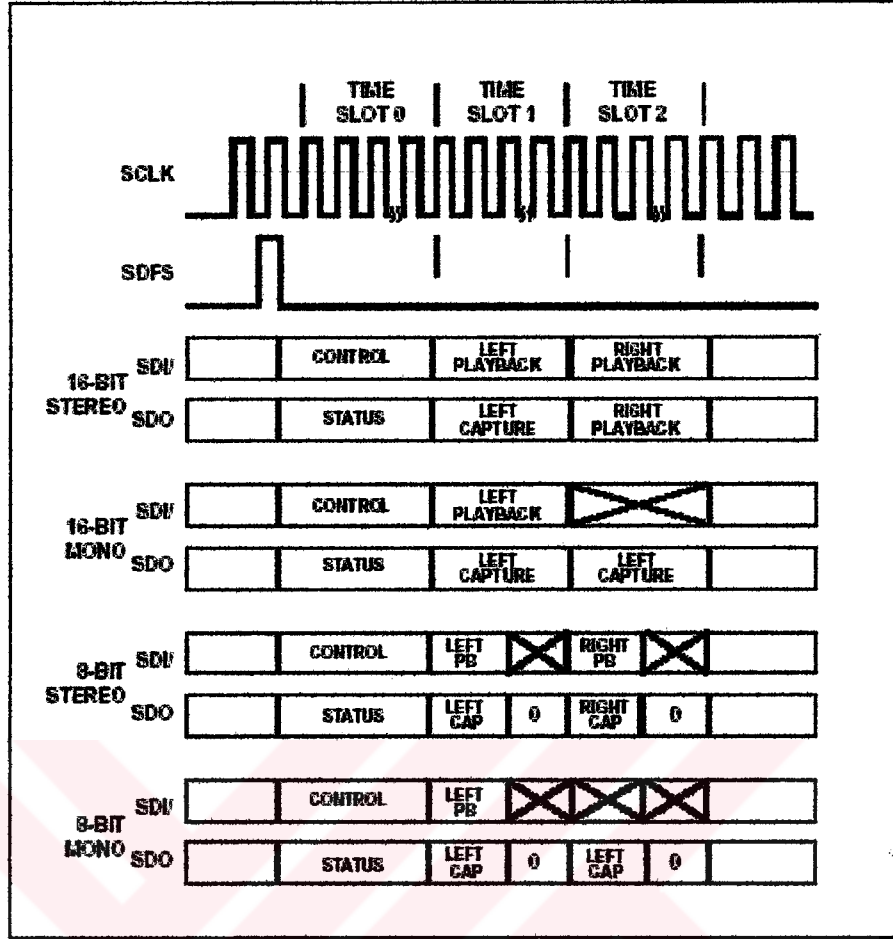
Kodlayıcı ve kod çözücü (CODEC) tümleşik devresi, A/D ve D/A çevrimlerini gerçekleştirir. Her DSP sisteminde mutlaka olması gereken bir elemandır. CODEC tümleşik devresi A/D ve D/A çevrim işlemlerini seri olarak yapmaktadır. Sayısal çevrilen *örneksel* İşaret Seri olarak DSP işlemcisine giriş olarak gitmektedir. Çıkış olarak ise, CODEC tümleşik devresine gelen *sayısal* bilgi seri olarak gelmekte ve *örneksel* bilgiye dönüştürülerek çıkışa verilmektedir. DSP işlemcisindeki iki Seri Porttan birine bağlanarak CODEC ile DSP arasındaki iletişim sağlanmaktadır.

AD1847 CODEC Tümleşik devresi iki kanallı Stereo CODEC'tir. A ve  $\mu$  tipi sıkıştırmayı destekler aynı zamanda 8 bit ve 16 bit tek biçim (uniform) dönüşümü de destekler. İçindeki ADC ve DAC'lar sigma delta tipi dönüştürücüdür. Örnekleme frekansı, 5.5 kHz'den 48 kHz'e kadar değiştirilebilir.



Şekil 5.8 AD1847 CODEC tümleşik devresi blok diyagramı (Vinay, 1995)

AD1847 Tümleşik devresi içerisinde 16 adet İndeks (Index) Yazmacı bulunmaktadır. Bu yazmaçlar yardımıyla örnekleme frekansı değişimi, çıkış ya da giriş işaretinin istenilen oranda zayıflatılması, veri formatı seçimi gibi bazı işlemler yapılabilir.



Şekil 5.9 AD1847 tümleşik devresi veri giriş ve çıkış zamanlama diyagramları (Vinay, 1995)

Şekil 5.9 den görüldüğü gibi AD1847 tümleşik devresinde veri giriş ve çıkış işlemi SDI ve SDO uçları üzerinden ve üçlü veri paketleri şeklinde gönderilir ve alınır. İlk bilgi Kontrol Yazmaç bilgisi, ikinci bilgi sol kanal bilgisi, üçüncü bilgi de sağ kanal bilgisidir. DSP'de ise CODEC ile iletişim kurmak için otomatik tampon bellek (Autobuffering) özelliğinden yararlanılmaktadır. Veri belleğinde 3 bellek bölgesi uzunluklu dairesel tampon bellek rx\_buf adıyla gelen bilgiyi almak içindir ve yine 3 bellek bölgesi uzunluklu dairesel tampon tx\_buf adı ile veri göndermek içindir. rx\_buf bellek bölgesine gelen bilgilerin üçü de geldiği zaman işlemci kesme üretir. Gelen bilgiler okunur ve tekrar aynı çevrim devam eder. tx\_buf bellek bölgesi için de benzer durum söz konusudur.

### 5.3.3 Sistemin Örnekleme Frekansı Seçimi

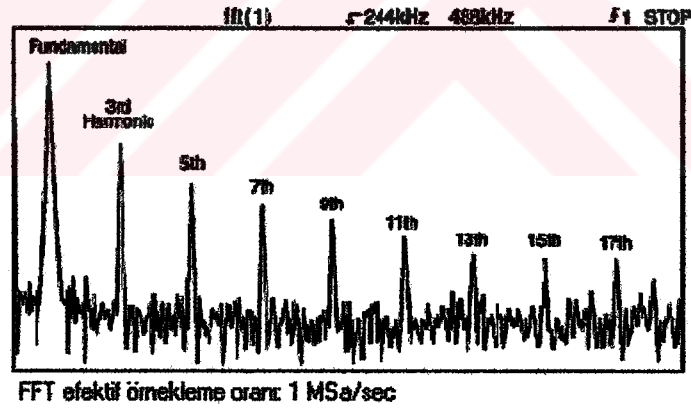
ADSP 2181 temelli EZ-KIT LITE kullanılarak gerçekleştirilen sistem, zaman düzlemi işaretini 48000 nokta ile kayıt etmektedir. Yazılan FFT rutini ise bu noktalardan  $N=1024$  tanesini kullanarak 512 noktalı bir frekans düzlemi gösterimi oluşturmaktadır. Oluşturulan bu frekans düzlemi gösterimi, 0 [Hz] ile  $f_{eff}/2$  [Hz] arasındaki frekans değerlerini gösterebilmektedir. Burada  $f_{eff}$  zaman kaydının (time record), efektif örnekleme frekansıdır.

Frekans düzlemi eksenlerinden yatay eksenin, frekans aralığı ve çözünürlüğü örnekleme frekansına ve FFT’de alınan örnek sayısına(N) bağlıdır. Örnekleme frekansı  $f_{eff}=10,24$  [kHz] olsa ve FFT için örnek sayısı,  $N = 1024$  olsa gösterimdeki frekans bileşenleri  $\Delta f=f_{eff}/N=10$ [Hz] aralıklarla DC(0 [Hz]) ile 5,11 [kHz] arasında dağılacaklardır. Bu çalışmada, örnekleme frekansı 48 kHz seçildiğinden frekans bileşenleri 46,875 Hz’lik artımlarla DC – 24 kHz aralığında dağılır.

Ayrıca Agilent 54600 serisi çok amaçlı osiloskoplar incelendiğinde, örnekleme frekansının değiştirilmeden rasgele tekrarlamalı örnekleme tekniği(random-repetitive sampling technique) ile spektrum bant genişliğinin 1000 kat arttırıldığı görülmüştür.

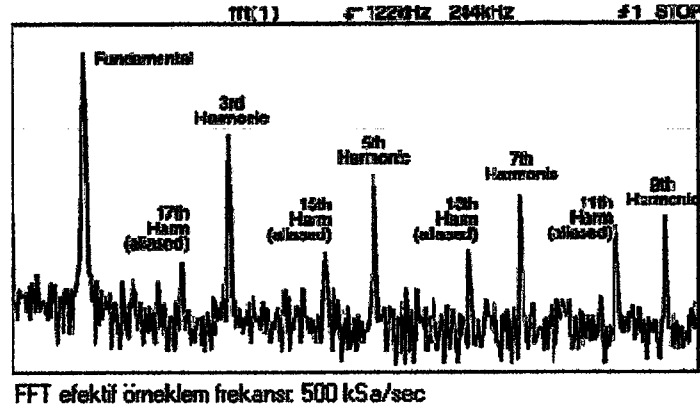
### 5.3.4 Örtüşme Sorunu ve Giderilmesi

$f_{eff}/2$  katlanma frekansı(folding frequency) olarak ta bilinir. Frekans düzlemi gösteriminde  $f_{eff}/2$ ’den daha yüksek frekansa sahip bileşenler diğer bileşenler üzerine katlanacaktır. İşte bu istenmeyen frekans bileşenlerine, örtüşen,katlanana (aliasing) bileşenler denir. Bu sorunun giderilmesi için efektif örnekleme frekansı, ölçülecek işaretin bant genişliğinin 2 katı olmalıdır.



Şekil 5.10 26 kHz’lik üçgen dalganın frekans düzlemi gösterimi (Brigham, 1998)

26 kHz’lik bir üçgen dalga 1 MSa/sec örnekleme oranıyla, frekans düzleminde gösterilmiştir. En soldaki uzun spektrum çizgisi, temel bileşeni ve sonraki büyük spektrum çizgileri ise sırasıyla diğer bileşenleri göstermektedir. Görüldüğü üzere, daha yüksek mertebeli harmoniklerin genlikleri daha küçüktür. FFT gürültüsü üzerinde en son görülebilen 17.harmoniktir ( $17 \times 26 \text{kHz} = 442 \text{kHz}$ ) ve bu harmonik katlanma frekansının ( $f_{eff}/2 = 500 \text{kSa/sec}$ ) içindedir. Bu sebeple değeri büyük bir örtüşme ortaya çıkmamıştır. Ancak örnekleme oranı, 500 kSa/sec’a düşürülürse aşağıda görülen örtüşmeler ortaya çıkar.

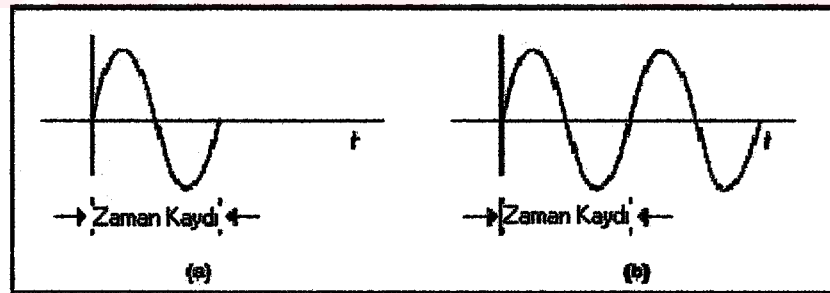


Şekil 5.11 Örtüşen frekans bileşenleri (Brigham, 1998)

Ölçme sistemlerinde örtüşen frekans bileşenlerinden kurtulmak için band sınırlı işaretlerle de çalışılabilir bu da sistemin girişine alçak geçiren bir filtre koyarak sağlanır. Bu çalışmada, gerçekleştirilen sistem ile bant genişliği 24[kHz]'e kadar olan işaretler ölçülebilir.

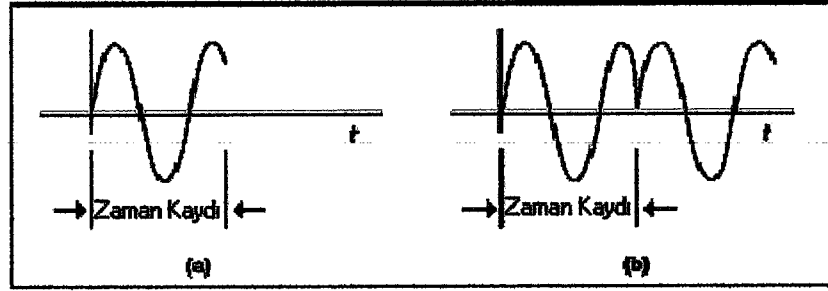
### 5.3.5 Spektrum Sızıntısı Sorunu ve Giderilmesi

FFT, zaman üzerinden entegral alan Fourier dönüşümünü gerçekleştirmek için sonlu bir zaman kaydı (time record) üzerinde çalışır ve sonlu zaman kaydını kopyalayarak tekrarlar. Bu şekilde Fourier entegraline yaklaşır. Aşağıdaki şekilde (a)'da, sonlu zaman kaydı gerçek işareti oldukça iyi temsil etmektedir bu nedenle FFT sonucu Fourier entegraline çok iyi yaklaşır. (b)'de de, işaret kopyalanmıştır ve geçiş noktaları oluşmamıştır.



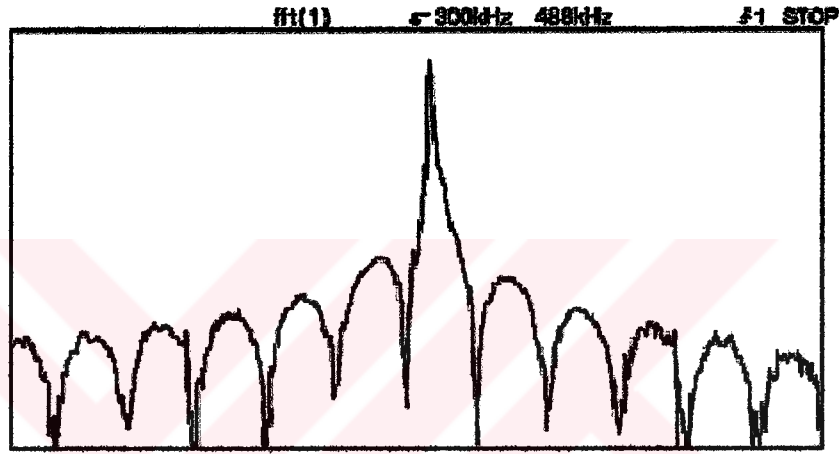
Şekil.5.12 Zaman kaydı gösterimi (Brigham, 1998)

Zaman kaydı aşağıdaki gibi olur ve tam olarak bir kayıt süresine sığmaz ise kopyalandığında, geçiş noktaları oluşur ve frekans düzleminde spektrum sızıntıları (Spectral Leakage) meydana gelir.



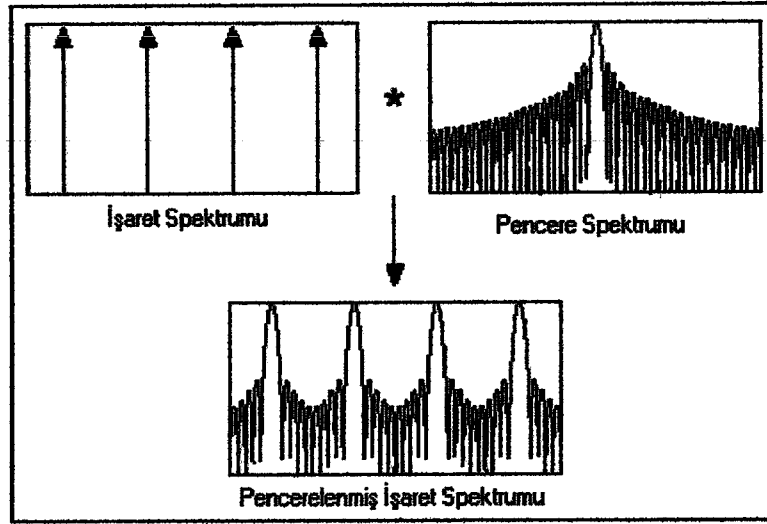
Şekil 5.13 Spektrum sızıntılı zaman kaydı (Brigham, 1998)

Spektrum sızıntısı, ince bir çizgi halinde frekans düzlemin aşağıdaki gibi görünür;



Şekil 5.14 Spektrum sızıntısının frekans düzlemi gösterimi (Brigham, 1998)

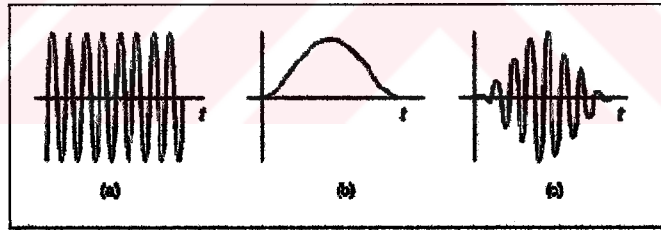
Spektrum sızıntısı problemini çözmek amacıyla; zaman kaydının başında ve sonunda işaret kopyalandıkça geçişlerin oluşmaması için işareti sıfır değerini almaya zorlamak gerekir. Bu olay, zaman kaydını bir PENCERE fonksiyonuyla çarpmak (konvüle etmek) anlamına gelir. Çarpmanın doğal bir sonucu olarak zaman kaydı değişecek ve frekans düzleminde o pencere fonksiyonunun etkisi hissedilecektir. Aşağıda bir pencere fonksiyonunun frekans karakteristiği görülmektedir. Bu sürekli spektrum bir ana lob(lobe) ve bir çok yan loblardan oluşmaktadır. Ana lob, zaman düzlemindeki işaretin her bir frekans bileşenini merkezlemektedir ve yan loblar da  $\Delta f$  ile 0'a yaklaşmaktadır.



Şekil 5.15 Pencerelemiş spektrumun frekans karakteristikleri (Brigham, 1998)

Genelde kullanılan 4 tane pencere fonksiyonu vardır bunlar; Hanning, Flattop, Dikdörtgensel (Rectangular), ve Üstel (Exponential) pencere fonksiyonlarıdır.

Hanning penceresi; zaman kaydının başında ve sonunda düzgün bir sıfır geçişine sahiptir. Aşağıdaki şekilde, (a)'da zaman düzleminde bir sinüs, (b)'de zaman düzlemi verisine uygulanacak Hanning penceresi ve (c)'de pencerelemiş zaman düzlemi kaydı görülmektedir.



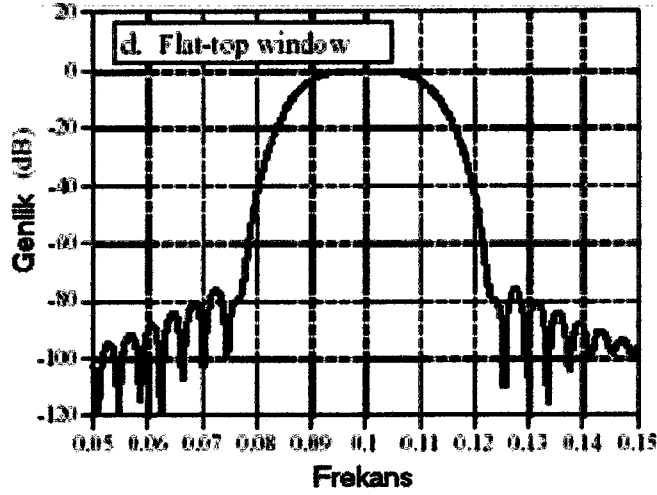
Şekil 5.16 Hanning penceresi ve etkisi (Brigham, 1998)

Pencerelelendikten sonra zaman düzlemi işaretinin şekli tamamen değişmesine rağmen, frekans bileşenleri aynıdır.

### 5.3.6 Pencere Fonksiyonu Seçimi

Pencere fonksiyonu tercihi, genlikteki doğruluk ve frekans çözünürlüğü arasında yapılacak seçime bağlıdır. Pencerelemiş işaretin frekans çözünürlüğü, pencere spektrumunun ana lobunun genişliğiyle sınırlıdır. Ana lob daraltıldıkça, sistemin birbirine yakın frekans bileşenlerini ayırma yeteneği artarken, pencerenin enerjisi yan loblara dağılacığından spektrum sızıntısı kötüleşecek ve genlik doğruluğu azalacaktır. *Hanning penceresi*, diğer pencere fonksiyon tipleriyle karşılaştırıldığında daha az genlik doğruluğu sağlarken çok daha

iyi bir frekans çözünürlüğü sunar. *Flat-top penceresi* ise; en geniş ana loba sahiptir bu sebeple frekans çözünürlüğü o kadar iyi değildir ancak mükemmel bir genlik doğruluğu sağlar.

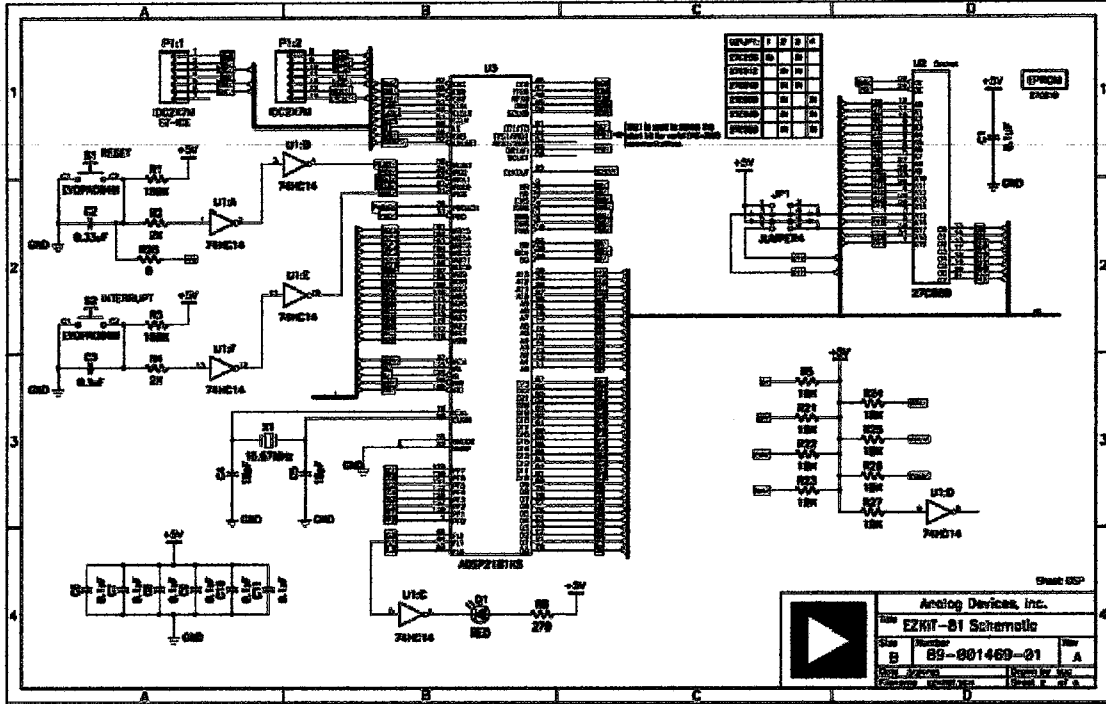


Şekil 5.17 Flat-top penceresinin frekans yanıtı (Brigham, 1998)

Dikdörtgen pencere ise, hiç pencere fonksiyonu uygulanmamış gibi örneklerin değerlerinin değişmediği bir yöntemdir. Bir çok geçiş durumu ortaya çıkabilir bu nedenle darbe, sönümlenen sinüs gibi sızıntı etkisinin ortaya çıkmayacağı garanti edilen özel durumlarda kullanılır. Üstel pencere ise, bazı özel durumlarda kullanılır. Bu çalışmada, Hanning penceresi kullanılması düşünülmektedir, şimdilik sadece dikdörtgen pencere kullanılmaktadır.

### 5.3.7 Devrenin Çalışması

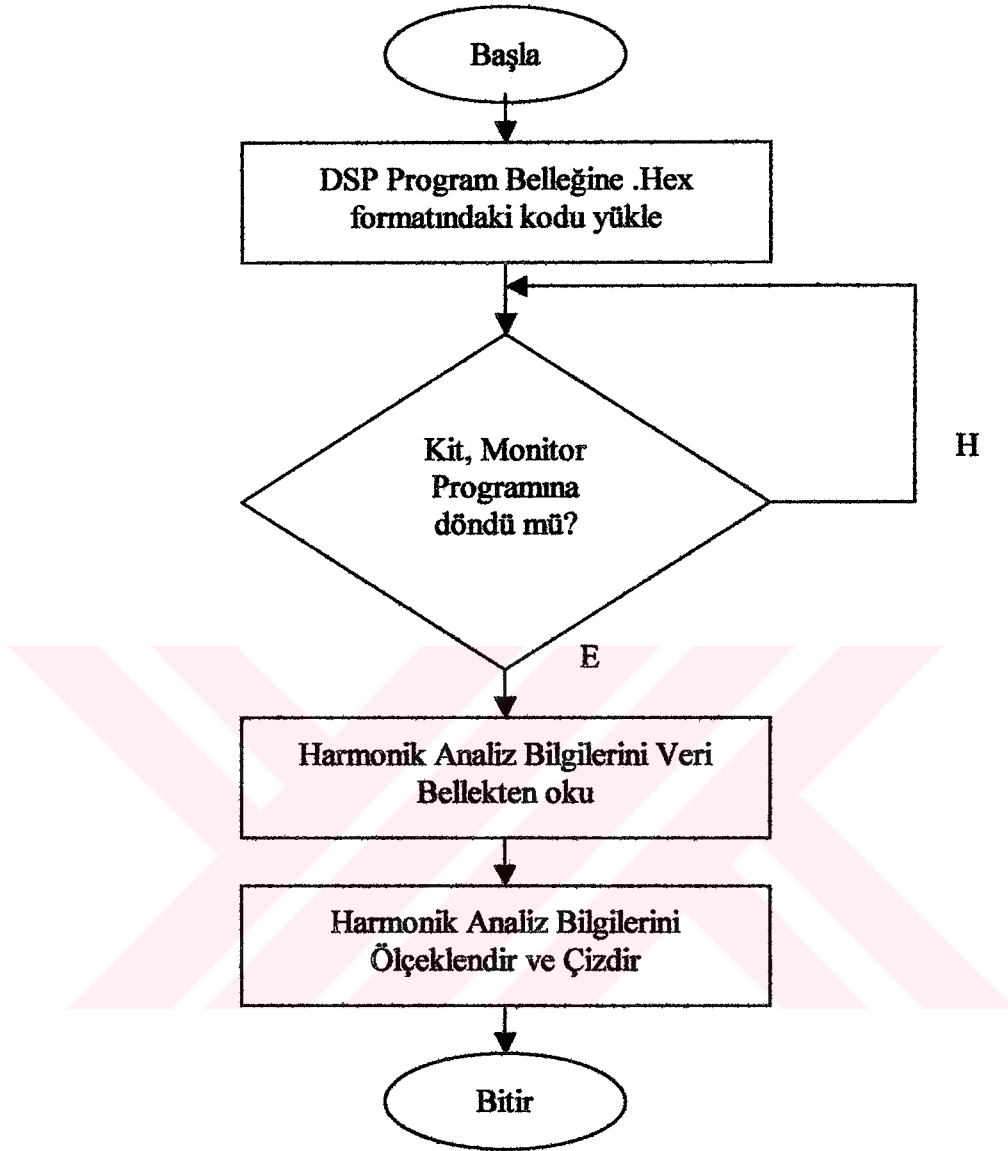
Devreye gerilim verildiğinde, reset devresi yaklaşık 30 msn sayısal işaret işleyiciyi “reset” konumunda tutar. 2181’in BMODE ve MMAP uçları toprağa bağlı olduğundan ilk başlamasını (booting) EPROM üzerinden yapar. EPROM üzerinde yüklü monitör programı sayesinde DSP, SPORT0 TX üzerinden AD1847 kodlayıcı ve kod çözücüsüne, bir dizi test emirleri gönderir ve bu emirlere gelen cevap işaretlerine göre AD1847’nin hazır olup olmadığı anlaşılır. Monitör programı ayrıca dahili RAM’i de kontrol ederek bir problem olmadığını, kit üzerindeki bir ledi aktif hale getirerek gösterir. Böylelikle devre, hatasız bir şekilde başlatılmış olur ve yazılan harmonik analiz programı koşturulabilir hale gelir.



Şekil 5.18 Devre şeması (Vinay, 1995)



#### 5.4 Harmonik Analiz Arayüz Programı Akış Diyagramı



Şekil 5.20 Harmonik analiz arayüz programı akış diyagramı

## 5.5 Deneysel Sonuçlar

MATLAB programına, Analog Devices'in Visual DSP++ 2.0 programı kullanılarak 44.1kHz'te örneklenen ve kitin bellek bölgesinden okunan veriler girilmiştir.

### 5.5.1 Deney-1

Kit Giriş Gerilimi:  $V_{max} = \pm 0.8$  [V], Kare Dalga

$f = 400$  [Hz]

Çizelge 5.1 Deney-1' in sonuçları

n. Harmonik	Matlab Genlik [Volt]	Deneysel Genlik [Volt]	Genlik Hatası %	Matlab Frekans [Hz]	Deneysel Frekans [Hz]	Frekans Hatası %
Temel Bileşen	0,8418	0,8411	0,08	387,59	387,5	0,02
3. Harmonik	0,3038	0,3036	0,06	1205,86	1205	0,07
5. Harmonik	0,1773	0,1787	0,78	2024,12	2024	0,005
7. Harmonik	0,1107	0,1131	2,16	2842,38	2842	0,01

### 5.5.2 Deney -2

Kit Giriş Gerilimi:  $V_{max} = \pm 0.8$  [V], Kare Dalga

$f = 800$  [Hz]

Çizelge 5.2 Deney-2' nin sonuçları

n. Harmonik	Matlab Genlik [Volt]	Deneysel Genlik [Volt]	Genlik Hatası %	Matlab Frekans [Hz]	Deneysel Frekans [Hz]	Frekans Hatası %
Temel Bileşen	0,8267	0,8282	0,18	818,26	818,2	0,007
3. Harmonik	0,2973	0,2979	0,2	2411,71	2411	0,029
5. Harmonik	0,1798	0,1799	0,05	4005,71	4005	0,017
7. Harmonik	0,1210	0,1206	0,33	5598,63	5598	0,011

### 5.5.3 Deney -3

Kit Giriş Gerilimi:  $V_{max} = \pm 0.8$  [V], Kare Dalga

$f = 1200$  [Hz]

Çizelge 5.3 Deney-3' ün sonuçları

n. Harmonik	Matlab Genlik [Volt]	Deneysel Genlik [Volt]	Genlik Hatası %	Matlab Frekans [Hz]	Deneysel Frekans [Hz]	Frekans Hatası %
Temel Bileşen	0,9106	0,9116	0,109	1205,86	1205	0,07
3. Harmonik	0,2914	0,2931	0,58	3617,57	3617	0,01
5. Harmonik	0,1607	0,1630	1,43	6029,29	6029	0,004
7. Harmonik	0,1060	0,1038	2,07	8441,94	8441	0,011

### 5.5.4 Deney -4

$V_{max} = \pm 0.8$  [V], Kare Dalga

$f = 2000$  [Hz]

Çizelge 5.4 Deney-4' ün sonuçları

n. Harmonik	Matlab Genlik [Volt]	Deneysel Genlik [Volt]	Genlik Hatası %	Matlab Frekans [Hz]	Deneysel Frekans [Hz]	Frekans Hatası %
Temel Bileşen	0,8262	0,8280	0,2	2024,12	2024	0,005
3. Harmonik	0,2844	0,2800	1,54	6029,29	6029	0,004
5. Harmonik	0,1596	0,1633	2,31	10034,47	10034	0,004
7. Harmonik	0,0935	0,0957	2,35	14039,64	14039	0,002

### 5.5.5 Deney -5

$V_{max} = \pm 0.8$  [V], Kare Dalga

$f = 4000$  [Hz]

Çizelge 5.5 Deney-5' in sonuçları

n. Harmonik	Matlab Genlik [Volt]	Deneysel Genlik [Volt]	Genlik Hatası %	Matlab Frekans [Hz]	Deneysel Frekans [Hz]	Frekans Hatası %
Temel Bileşen	0,9286	0,9266	0,215	4091,3	4091	0,007
3. Harmonik	0,2523	0,2577	2,14	12230,85	12230	0,006
5. Harmonik	0,1282	0,1302	1,56	20413,47	20413	0,002
7. Harmonik	---	---	---	---	---	---

### 5.5.6 Deney -6

$V_{max} = \pm 0.8$  [V], Kare Dalga

$f = 6000$  [Hz]

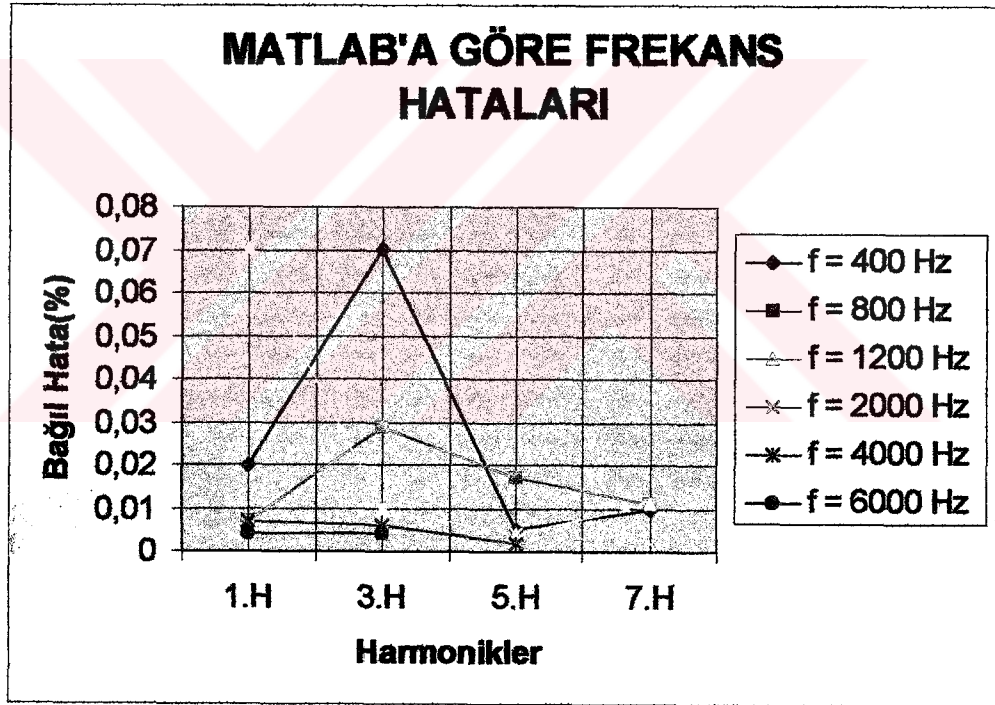
Çizelge 5.6 Deney-6' nın sonuçları

n. Harmonik	Matlab Genlik [Volt]	Deneysel Genlik [Volt]	Genlik Hatası %	Matlab Frekans [Hz]	Deneysel Frekans [Hz]	Frekans Hatası %
Temel Bileşen	0,8674	0,8645	0,33	6029,29	6029	0,004
3. Harmonik	0,2673	0,2672	0,03	18044,82	18044	0,004
5. Harmonik	---	---	---	---	---	---
7. Harmonik	---	---	---	---	---	---

### 5.5.7 Sonuçların Frekans Bakımından Değerlendirilmesi

Çizelge 5.7 Frekans bakımından sonuçlar

	1.H	3.H	5.H	7.H
f= 400 Hz	0,02	0,07	0,005	0,01
f= 800 Hz	0,007	0,029	0,017	0,011
f= 1200 Hz	0,07	0,01	0,004	0,011
f= 2000 Hz	0,005	0,004	0,004	0,002
f= 4000 Hz	0,007	0,006	0,002	
f= 6000 Hz	0,004	0,004		

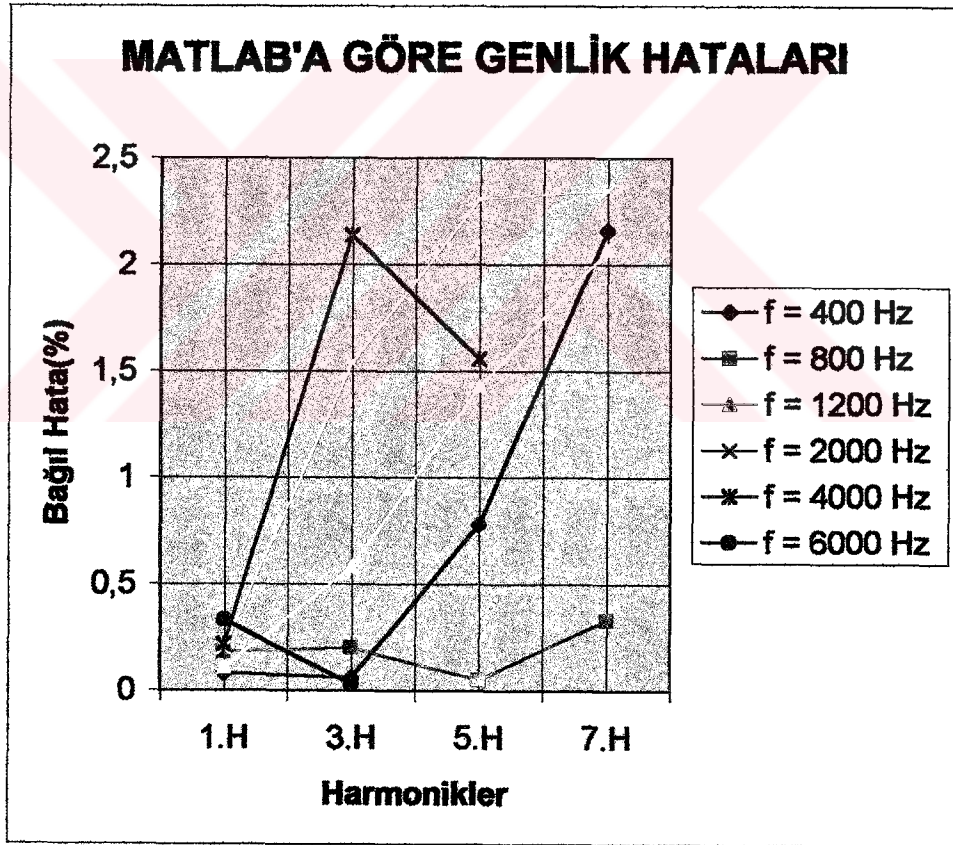


Şekil 5.21 Matlab'e göre bağıl frekans hataları

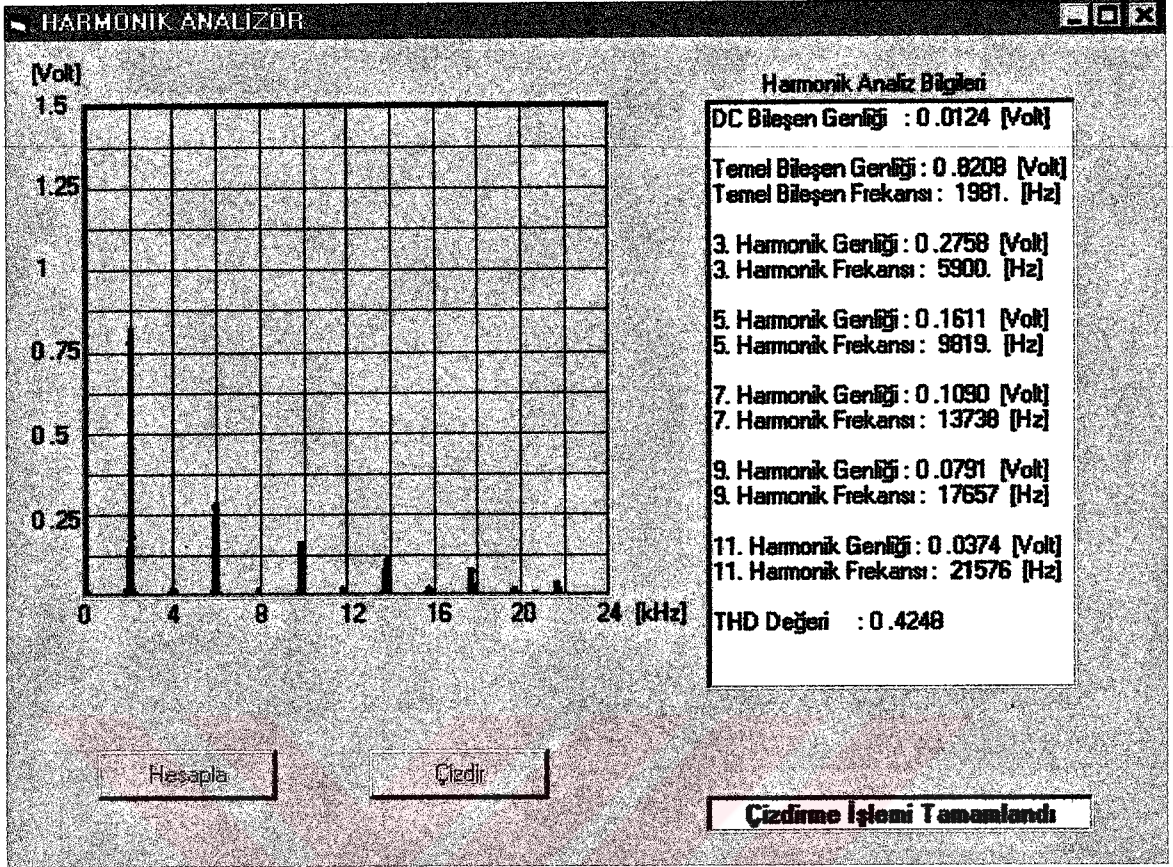
### 5.5.8 Sonuçların Genlik Bakımından Değerlendirilmesi

Çizelge 5.8 Genlik bakımından sonuçlar

	1.H	3.H	5.H	7.H
f = 400 Hz	0,08	0,06	0,78	2,16
f = 800 Hz	0,18	0,2	0,05	0,33
f = 1200 Hz	0,109	0,58	1,43	2,07
f = 2000 Hz	0,2	1,54	2,31	2,35
f = 4000 Hz	0,215	2,14	1,56	
f = 6000 Hz	0,33	0,03		



Şekil 5.22 Matlab'e göre bağıl genlik hataları



Şekil 5.23 Arayüz örnek ekran görüntüsü

## 6. SONUÇLAR VE ÖNERİLER

Bir sayısal işaret işleyici (DSP) kullanılarak, ayırık zamanda Fourier Dönüşümüyle (DFT) şebeke harmoniklerinin analizi gerçekleştirilmiş ve bir yazılım aracılığıyla analiz sonuçları ölçekli olarak görselleştirilmiştir.

Harmonik analizi, kompleks periyodik bir işaretteki harmoniklerin faz, genlik ve değerlerinin belirlenmesidir. Bunu gerçekleştirmek için, sayısallaştırılmış zaman düzlemi verisini alan ve frekans düzlemi bilgisini hesaplayan Ayırık Zamanlı Fourier Dönüşümü (DFT) kullanılmıştır. Ayırık Zamanlı Fourier Dönüşümünün hesaplanması için de, 1024 noktalı 2 tabanlı (radix2) Zamanda İndirgemeli Hızlı Fourier Dönüşüm (DIT FFT) yöntemi tercih edilmiştir.

Çalışma sonucunda, şebekedeki sinüzoidal değişimden uzaklaşmış işaretlerin, içerdiği 0 ile 22 kHz arasındaki harmonikler ve genlikleri hakkında bilgi edinilebilecek 3,9 msn cevap süreli bir sistem geliştirilmiştir. Bu sistem elde ettiği bilgileri, seri port üzerinden özel bir yazılım aracılığıyla bilgisayara aktararak kullanıcının analizi daha rahat bir şekilde incelemesi sağlanmıştır. İleriye yönelik olarak, analiz edilebilecek frekans aralığının genişletilmesi için çalışmalar yapılabilir.

**KAYNAKLAR**

Kayran, A., (1990), Sayısal İşaret İşleme, İTÜ Matbaası.

Kocatepe, C., (2002), Güç Sistem Harmonikleri Yüksek Lisans Ders Notları.

Vinay K., (1995), Digital Signal Processing Laboratory Using The ADSP-2101 Microcomputer, Prentice Hall.

Sedef H., Attar F., Uzun T., (1997), "Enerji Sistemlerinde Meydana Gelen Harmoniklerin Bir Mikrodenetleyici Kullanılarak İşlenmesi".

Wistman G., (1997), "An Integrated 256-Point Complex FFT Processor for Real Time Spektrum Analysis and Measurements", IEEE Proceedings of Instrumentation and Measurement Technology Conference, vol.1, pp. 96-101.

Garcia J., Juan A., Angel M., (1999), "VLSI Configurable Delay Commutator for a Pipeline Split Radix FFT Architecture".

Brigham, E., (1998), The Fast Fourier Transform and Its Applications, Englewood Cliffs, Prentice Hall

**ÖZGEÇMİŞ**

<b>Doğum Tarihi</b>	28 / 11 / 1977	
<b>Doğum Yeri</b>	Sakarya	
<b>Lise</b>	1991-1996	Fatih Anadolu Teknik Lisesi Elektronik Bölümü
<b>Lisans</b>	1996-2000	Karadeniz Teknik Üniversitesi Mühendislik Fak. Elektronik Mühendisliği Bölümü
<b>Yüksek Lisans</b>	2001-	Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Elektronik ve Hab Müh Anabilim Dalı, Elektronik Programı

**Çalıştığı kurumlar**

2000-Devam ediyor Atasan Metal Ltd. Şti., Bakım Mühendisi