YILDIZ TEKNİK ÜNİVERSİTESİ * FEN BİLİMLERİ ENSTİTÜSÜ

Spektrum Analizor

Cengiz Say

Yuksek Lisans Tezi

YILDIZ UNIVERSITESI
FEN BILIMLERI ENSTITUSU

SPEKTRUM ANALIZOR

YUKSEK LISANS TEZI

MUH. CENGIZ SAY

ISTANBUL 1990

YILDIZ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

SPEKTRUM ANALİZÖR

YÜKSEK LİSANS TEZİ

MÜH. CENGİZ SAY

İSTANBUL 1990

ÖNSÖZ

Değerli önerileri ve yapıcı eleştirileriyle çalışmalarıma katkıda bulunan Sayın Yr. Doç. Dr. Osman Sirel'e teşekkür ederim.

Ayrıca tezimin hazırlanışında değerli katkılarından dolayı Yük. Müh. Kemal SUNER, Müh. Ahmet SAKALLIOGLU, Enis KIRIMLI, ELEKTROPANÇ LTD ŞTİ ve Kısmet YILDIRIM'a teşekkür ederim.

İÇİNDEKİLER

ÖZET

BİLGİSAYAR DESTEKLİ SPEKTRUM ANALİZÖR

Bu çalışmada önce bir işaretin frekans spektrumu hakkında genel bir bilgi verilmiştir. Daha sonra frekans spektrumunun bileşenlerini incelenmiş ve spektrum analizörlerde sıkça kullanılan ve işaretin frekans spektrumu alınırken, kenar taşmalarını önlemek için kullanılan sınırlı sayıda örneklenmiş bir işaretin pencerelenmesi olayı, klasik pencere fonksiyonlarının karşılaştırılması şeklinde anlatılmıştır. Bunları takiben, elde edilen bu örneksel verilerin bilgisayar desteği ile Hızlı Fourier Çevrimlerinin bulunması için gerekli donanım ve yazılım kavramları açıklanmıştır. Yazılım, örneklenmiş işaretlerin Hızlı Fourier Çevrimi'ni ve bu işlemin sonucundaki verilerin incelenmesini sağlayan QUICK BASIC'de yazılmış bir programdan oluşmuştur. Değişik işaretlerden alınan işaretler ve bunların frekans spektrumlarının ayrı ayrı printer çıktıları verilerek oluşturulan programın pratik sonuçları gözler önüne serilmiştir. Son olarak, varılan sonuçlar ve gelecekte bu konuda yapılabilecek çalışmalar anlatılmıştır.

SUMMARY

In this Thesis, first given a general explanation of the frequency spectrum of anf signal even periodic or not and than an introducting of spectrum analyzers. Because the windowing of infinitive samples of a signal with a square function before Discrete Fourier Transform (DFT) caused leakage (adjanced bands due to side-lobes of transform of the square window), some of well-known window functions are compared with each other by their performance. So that, a hardware and software necessary to accomplish the Fast Fourier Transform (FFT) which is a fast algorithm rather than DFT and time and frequency domain graphics, is introduced. The software is written in QUICK BASIC language.

Some deterministic signals are used as inputs of the analyzer and results (i.e. their spectrums) are showed in a monochrome monitor with HERCUL GRAPHIC CARD and also in printer outs.

I. BÖLÜM

GİRİŞ

Elektriksel haberleşme tekniğinde kullanılan işaretler akım ve gerilim gibi zamanın bir fonksiyonu olarak değişen büyüklüklerdir. Ekseri, x(t), y(t),... ile gösterilen işaretler "zaman domeninde" tanımlanmış olup, t, bağımsız değişkeni zamanı göstermektedir. Ancak haberleşme tekniğinde, işaretleri "frekans domeninde" tanımlamak daha uygun olmakta ve bir çok yararlar sağlamaktadır. X(f), Y(f),... ile gösterilen işaretlerin frekans domeni tanımlarında, f bağımsız değişkeni frekansı göstermektedir. Kabaca bir x(t) işaretinin, her biri uygun genlik ve fazda olan, belli sayıdaki frekans bileşenlerinden oluştuğu düşünülebilir. Bu nedenle her ne kadar bir işaret zaman domeninde var ise de, "spektrum" diye adlandırılan frekans domenindeki bileşenlerinden oluşacaktır.

Bu çalışmada, sözü edilen bileşenlerin bilgisayar yardımı ile görüntülenmesi sağlanmıştır.

II. BÖLÜM

TASARIMDA KULLANILAN YÖNTEM

2.1. Fazörlerden Spektruma Geçiş

Verilen bir x(t) işaretinin frekans bileşenlerini bulmak için önce elektriksel haberleşmede kullanılan en basit işaret tipini, yani sinüzoidal bir işaret biçimini ele alalım [1].
Örneğin,

$$x(t) = A.\cos(\omega t+\Phi)$$

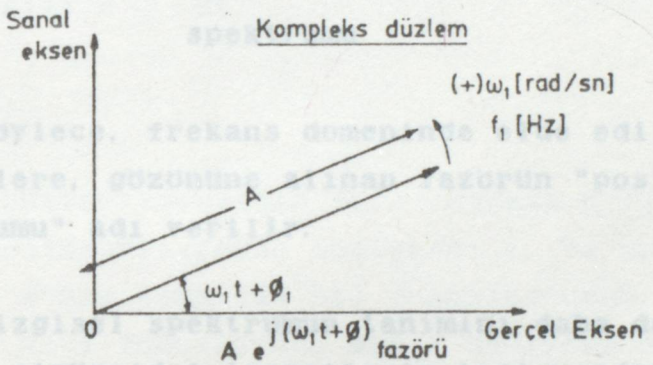
olsun. Euler teoreminden yararlanarak x(t),

$$x(t) = \mathrm{Re}\{ A.e^{j(\omega t+\Phi)} \}$$

biçiminde yazılabilir. Bu eşitlikteki parantez içindeki terime "kompleks düzlemde dönen bir vektör" veya "fazör" adı verilir. Şekil 2.1.'den de görüleceği gibi bir fazör üç özelliği ile tanımlanır.

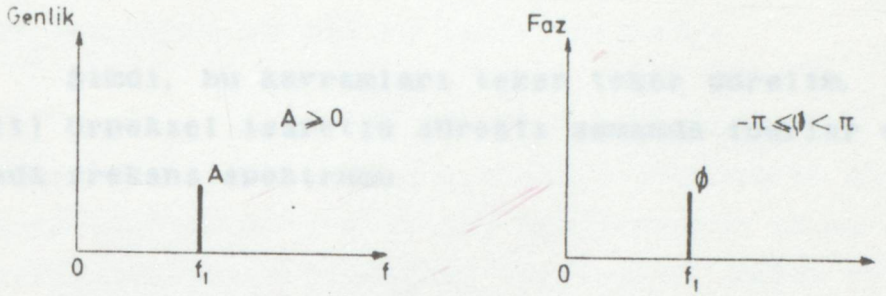
1) A: Fazörün genliği olup, pozitif bir büyüklüktür. ($A \geq 0$)
2) Φ: Fazörün fazı, fazörün t=0 anında gerçel eksenle yaptığı açı olarak tanımlanır ve $-\pi \leq \Phi < \pi$ arasındadır.

3) $\omega \triangleq 2.\pi.f$: Fazörün açısal dönme hızı. Fazörün saat ibrelerinin tersi yönünde ω rad/sn açısal hızı ile döndüğü varsayılmakta olup bu yön pozitif olarak seçilmiştir.



Şekil 2.1. Fazör Diyagramı

Aynı fazörü frekans domeninde belirlemek için, fazörün yalnız f frekansı için tanımlanmış olduğuna dikkat etmemiz gerekir. Bu nedenle, verilen bir fazör frekans domeninde ancak iki ayrı grafikle belirlenebilir. Bunlar Şekil 2.2.'den de görüleceği gibi, fazörün genliğini ve fazını frekansa bağlı olarak gösteren grafiklerdir.



Şekil 2.2. $x(t) = A.Cos(\omega t+\Phi)$ nin çizgisel spektrumu

Böylece, frekans domeninde elde edilen bu grafiklere, gözönüne alınan fazörün "positif çizgisel spektrumu" adı verilir.

Çizgisel spektrumun tanımını daha da genelleştirerek, sinüzoidal işaretlerin toplamından elde edilen işaretlerin positif çizgisel spektrumları da kolayca elde edilebilir.

2.2. Ayrık Fourier Çevrimi (DFT) ve Hızlı Fourier Çevrimi (FFT)

Bilindiği gibi Fourier çevrimi zaman domeninden frekans domenine geçmek için kullanılan bir yöntemdir ve akustik, optik, sismoloji, telekomünikasyon, işaret işleme, görüntü (image) işleme tekniklerinde önemli bir yer tutar. Ayrık fourier çevrimi (DFT) ise, adından da anlaşılacağı gibi sürekli-zamandaki fourier çevriminin ayrık zamanda (örneklenmiş) eşdeğeridir [2] [3] [4]. Benzer şekilde hızlı fourier çevrimi ise (FFT), DFT'nin hesaplanması sırasında yapılan işlem miktarının (dolayısıyle işlem zamanının) azaltılması için geliştirilmiş bir algoritmadır ve DFT'nin bir alt

sınıfı olarak nitelendirebilir [5].

Şimdi, bu kavramları teker teker görelim. Bir x(t) örneksel işaretin sürekli zamanda fourier çevrimi yada frekans spektrumu

$$X(w) = \int_{-\infty}^{\infty} x(t).e^{-j\omega t}.dt \qquad (1)$$

şeklinde gösterilmektedir [6]. Burada x(t) ve X(ω), sırasıyle, sürekli-zaman değişkeni t, ve frekans değişkeni ω'nın kompleks bir fonksiyonudur. Sürekli zamandaki x(t) fonksiyonu, her T saniyede örneklenerek x(nT) şeklinde bir ayrık zaman işaretine dönüştürülür. Örnekleme periyodunun değeri belli olduğundan T notasyonunu kaldırırak ayrık işareti x(n) ile gösterebiliriz. Buna göre ayrık işaretin fourier çevrimi

$$X(w) = \sum_{n=-\infty}^{\infty} x(n)\, e^{-j\omega n} \qquad (2)$$

şeklindedir [7]. Burada ω, değeri 0 ile 2π arasında değişen normalize frekansı gösterir. Dolayısıyla X(ω)'nın da değeri 0 ile 2π arasında değişecektir. X(ω), örneklenmiş bir işaret olan x(n)'nin doğrudan bir sonucudur. Genelde zaman domenindeki örnekleme, frekans domenindeki periyodiklilikle ilişkilidir ve

bunun karşıtı da doğrudur. Bu fourier teorisinin temel bir sonucudur ve DFT'yi tanımlamaktadır.

Farzedelim ki x(n) N adet örnek içersin. Bu örneklemeler aralığının dışındaki değişimler göz önüne alınmaksızın, bu aralıktaki değişimlerin zaman içerisinde tekrarlandığını varsayalım. Kabul edilen varsayım bizi, yukarıda sözü edilen örnekleme ile periyodiklik arasındaki ilişkiye götürecek, bu da fourier çevriminin örnekler arasındaki aralığın zaman domenindeki işaretin frekansına eşdeğer durumuna gelmesiyle ayrık hale dönüşmesine yol açacaktır. Bu aralığın değeri normalize frekans biriminde $2\pi/N$'dir. Sonuçta DFT,

$$X(k) = \sum_{n=0}^{N-1} x(n)\, W_N^{nk} \qquad k = 0, 1, 2, \ldots, N-1 \qquad (3)$$

biçimini alır [8]. Burada

$W_N = e^{-j2\pi/N}$ ve W_N faz veya twiddle faktör olarak bilinir ve (3) nolu denklem N noktada DFT olarak adlandırılır.

DFT, örneklenmiş işaretlerin fourier çevrimini almak için uzun zamandır kullanılmaktadır. Ancak, DFT

işlemi sırasında, N çevrim nokta sayısı olmak üzere 4xNxN gerçel çarpma ve Nx(4xN-2) gerçek toplama yada NxN kompleks çarpma ve Nx(N-1) kompleks toplama yapılması gerekmektedir. Ancak bu işlem sayılarının azaltılması çalışmaları 1965 yılından beri yapılmakta ve ortaya çıkan algoritmalar hızlı fourier çevrimi olarak bilinmektedir.

Çok çeşitli FFT yöntemlerinden bazıları şunlardır; Goertzel Algoritması [9], zamanda ve frekansta ayrıştırma [10], Winograd fourier çevrim algoritması (WFTA) [9], fourier çevriminin simetri özelliğinin kullanılması [11], Radix-2; DFT'nin hızlandırılmış algoritması [12]. Bu tezde kullanılacak yöntem zamanda ayrıştırma'dır. Aşağıda bu yöntem anlatılacaktır.

Hesaplamalarda belirli bir verim elde etmek için, DFT işlemlerinin, daha küçük işlemlere bölünmesi gerekmektedir.

Bunun için, $W_N^{kn} = e^{-j(2\pi/N)kn}$ kompleks üstel ifadesinin

periyodik simetrik olma özelliğinden yararlanılacaktır. x(n) giriş veri dizisinin, daha ufak alt dizilere ayrıştırma prensibine dayanan bu algoritmaya "zamanda ayrıştırma" adı verilir. Zamanda ayrıştırma yönteminde, N genelde 2'nin bir katıdır. (Radix-2).

Temel DFT ifadesi olan (3) nolu denklem dikkate

alırsak, X(k) ifadesindeki tek ve çift numaralı toplamları şu şekilde ifade edebiliriz:

$$X(k) = \sum_{n\ \text{çift}} x(2r)\, W_N^{nk} + \sum_{n\ \text{tek}} x(n)\, W_N^{nk} \qquad (4)$$

burada n çift iken n yerine 2.r ve n tek iken n yerine 2.r+1 koyarak,

$$X(k) = \sum_{r=0}^{N/2-1} x(2r)\, W_N^{2rk} + \sum_{r=0}^{N/2-1} x(2r+1)\, W_N^{(2r+1)k} \qquad (5)$$

$$= \sum_{r=0}^{N/2-1} x(2r)[W_N^{2}]^{rk} + W_N^{k} \sum_{r=0}^{N/2-1} x(2r+1)[W_N^{2}]^{rk} \qquad (6)$$

bununla beraber, $W_N^2 = W_{N/2}$ olduğundan,

$$X(k) = \sum_{r=0}^{N/2-1} x(2r)\, W_{N/2}^{rk} + W_N^{k} \sum_{r=0}^{N/2-1} x(2r+1)\, W_{N/2}^{rk} \qquad (7)$$

$$= G(k) + W_N^k H(k) \qquad (8)$$

Böylece, (8) eşitliğinde her toplam N/2 noktadaki DFT'yi göstermektedir. Her ne kadar, k indisi N değer alıyor görünsede, her toplamın k'nın 0 ile N/2-1 arasındaki değerleri için hesaplanması yeterlidir. Çünkü (8) eşitliğindeki G(k) ve H(k) değişkenleri, periyod N/2 olmak şartı ile, tüm k değerleri için periyodik olma özelliğini taşımaktadırlar. Bu iki N/2 noktadaki DFT 'nin toplamı hesaplandıktan sonra, N noktasındaki DFT bulunmuş olunur. N/2 noktadaki DFT değerleri ise, tekrar N/4 noktadaki DFT toplamları ile hesaplanabilir. Böylece (8) eşitliğindeki G(k) ve H(k) değerleri aşağıda gösterildiği gibi hesaplanabilir:

$$G(k) = \sum_{l=0}^{N/4-1} g(l)\, W_{N/2}^{2lk} \qquad (9)$$

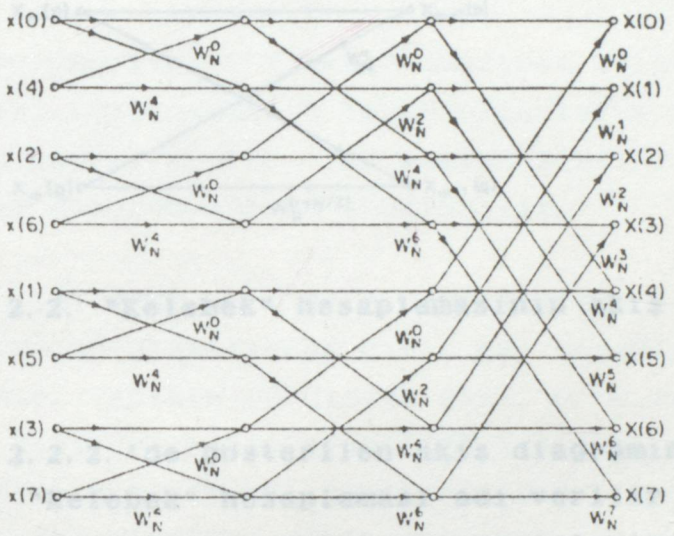
$$= \sum_{l=0}^{N/4-1} g(2l)\, W_{N/2}^{2lk} + \sum_{l=0}^{N/4-1} g(2l+1)\, W_{N/2}^{(2l+1)k} \qquad (10)$$

ya da,

$$G(k) = \sum_{l=0}^{N/4-1} g(2l)\, W_{N/4}^{lk} + W_{N/2}^{k} \sum_{l=0}^{N/4-1} g(2l+1)\, W_{N/4}^{lk} \qquad (11)$$

Genelleştirmek gerekirse, (10) ve (11) eşitlikleri, bölme işlemlerine devam edilerek, 2 noktadaki DFT haline getirilebilir. Böylece bu bölme işlemleri v=log N adımda tamamlanır ve DFT'deki NxN kompleks çarpma işlemine karşılık bu yöntem ile Nxlog N çarpma ve toplama işlemi yapılmaktadır. N=8 nokta için, zamanda ayrıştırma yöntemi ile yapılmış FFT akış diyagramı Şekil 2.2.1.'de verilmiştir. Şekil 2.2.1.'de görülen diyagramdan da anlaşılacağı gibi, her hesaplama adımında, N kompleks sayı, bir başka N kompleks sayıya çevrilmekte ve bu işlem v=log N kere tekrar etmektedir. m'inci hesaplama adımda elde edilen kompleks sayı dizisini Xm(i) ile gösterelim; burada i = 0, 1, ... N-1 ve m=1, 2 ... v'dir. Buna dayanarak, giriş örneklerimizi, Xo(i) değişkeni ile gösterebiliriz. Şekil 2.2.1.'de gösterildiği gibi, N=8 için Xo(i) şu şekilde olacaktır:

$$\begin{aligned} Xo(0) &= x(0) \\ Xo(1) &= x(4) \\ Xo(2) &= x(2) \\ Xo(3) &= x(6) \\ Xo(4) &= x(1) \\ Xo(5) &= x(5) \\ Xo(6) &= x(3) \\ Xo(7) &= x(7) \end{aligned} \qquad (12)$$

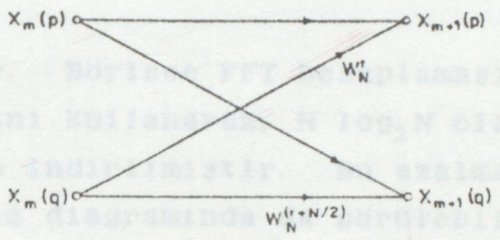


Şekil 2.2.1. Sekiz noktadaki FFT'nin (zamanda ayrıştırma yöntemi ile) akış diyagramı

Bu notasyona dayanarak, Şekil 2.2.1.'deki akış diagramının temel hesaplama diagramı Şekil 2.2.2.'de göaterilmiştir. Bu temel hesaplama diagramının gösterdigi eşitlik şöyle yazılır;

$$X_{m+1}(p) = X_m(p) + W_N^r X_m(q) \qquad (13)$$

$$X_{m+1}(q) = X_m(p) + W_N^{r+N/2} X_m(q)$$



Şekil 2.2.2. "Kelebek" hesaplamasının akış diyagramı

Şekil 2.2.2.'de gösterilen akış diagramının şekil nedeni ile, "Kelebek" hesaplaması adı verilir [13][14]. (12) eşitliğinde bulunan üstel çarpma işleminin sadeleştirilmesi, kompleks çarpma işleminin sayısının da iki katlık bir azalma meydana getirmektedir. Aşağıda bu sadeleşme gösterilmiştir:

$$W_N^{N/2} = e^{-j(2\pi/N)N/2} = e^{-j\pi} = -1 \qquad (14)$$

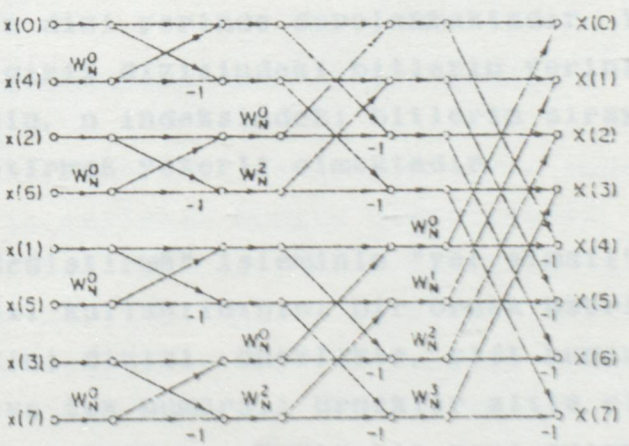
Böylece, (12) eşitliğinde üstel ifade yerine (13) eşitliğini kullanarak, (12) eşitliği

$$X_{m+1}(p) = X_m(p) + W_N^r X_m(q)$$

$$X_{m+1}(q) = X_m(p) - W_N^r X_m(q) \qquad (15)$$

halini alır. Böylece FFT hesaplamasında kelebek hesaplamasını kullanarak, $N \log_2 N$ olan işlem sayısı $N/2 \log_2 N$'e indirilmiştir. Bu azalma Şekil 2.2.3.'de verilen akış diagramında da görülebilir.

Eşitlik (14)'deki p, q ve r değişkenleri Şekil 2.2.3.'den de görülebileceği gibi, her hesaplama adımda değişmektedir. Açıkça görüldüğü gibi, p, ω ve (m+1)'inci dizileri hesaplayabilmek için, p, q ve m'inci dizilerin bilinmesi gerekmektedir. Böylece sadece bir tane N-bitlik saklayıcı, Xm+1(p) ve Xm+1(q) ile Xm(p) ve Xm(q) dizileri aynı saklayıcıda depolandığından, yeterli olmaktadır. Bu tip hesaplama, "yeralmalı" hesaplama olarak bilinmektedir. Her yeni dizinin, bir önceki dizinin depolandığı yere depolanabilmesi, önemli bir avantajdır.



Şekil 2.2.3. Kelebek hesaplamanın kullanıldığı, N=8 nokta için FFT akış diagramı.

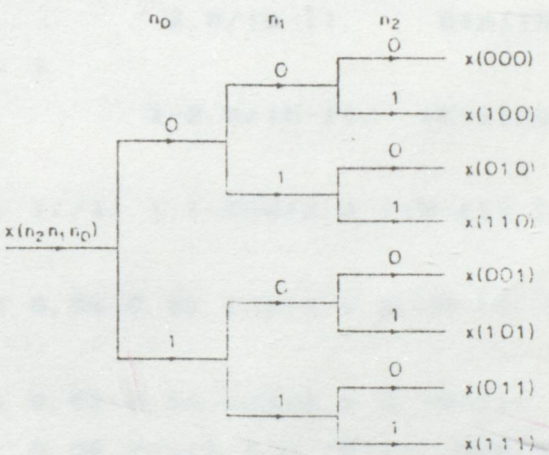
Hesaplamaların yeralma prensibinde çalışabilmesi için, giriş verilerinin sırasız biçimde depolanabilmesi gerekmektedir. Aslında giriş verilerinin sırasız biçimde depolanma işlemi, "bit yer değiştirme" işlemidir. Bit yer değiştirme mantığını anlamak için, daha önce anlatılan sekiz noktadaki akış diagramları örnek alalım. Bu diagramlardaki veriler üç adet ikili hane ile indekslenmiştir. Eşitlik (12)'deki değişkenleri ikili formda yazarsak, şu eşitliği elde ederiz:

$$
\begin{aligned}
Xo(000) &= (000)\\
Xo(001) &= (100)\\
Xo(010) &= (010)\\
Xo(011) &= (110)\\
Xo(100) &= (001)\\
Xo(101) &= (101)\\
Xo(110) &= (011)\\
Xo(111) &= (111)
\end{aligned}
\qquad (15)
$$

Eğer (n2, n, n0), x(n) dizisinin ikili indeks yazılışı olarak alınırsa, o zaman x(n2, n, n0) değeri, Xo(n0, n1, n2)'nin dizi yerinde depolanmaktadır. Yani, x(n2, n1, n0)'ın giriş dizisindeki bitlerin yerinin belirlenmesi için, n indeksindeki bitlerin sırayla yerlerini değiştirmek yeterli olmaktadır.

"Bit yer değiştirme" işleminin "yer almalı" hesaplamada nasıl kullanıldığını bir örnek üzerinde açıklayalım. x(n) dizisi, öncelikle, çift numaralı örnekler üstte ve tek numaralı örnekler altta olmak üzere ikiye ayrılmaktadır. Böyle bir veri ayırma işleminin bilgisini, bit yerdeğiştirmesinde kullanılacak indeksin en az anlamlı bitinde

saklayabiliriz. Eğer en az anlamlı bit "0" ise, Xo(i) dizisi çift numaralı değerlere ve eğer en az anlamlı bit "1" ise, aynı dizi tek numaralı değerlere eşitlenecektir. Bu işlem $v = \log_2 N$ kere devam edecek ve son adımda indeksin en anlamlı bitine gelinecektir. Böylece indeksin bit sayısı da $v = \log_2 N$'e eşit olacaktır. Bu tek ve çift olarak indeksin alt dizileri ayırma işlemi Şekil 2.2.4.'de verilen ağaç diagramında görülebilir.



Şekil 2.2.4. "Bit yer değiştirme" işlemindeki ayırmaları gösteren ağaç diyagramı

2.3. Örneklenmiş İşaretlerin Pencerelenmesi :

Ayrık Fourier Çevrimi işleminden önce, sonsuz sayıdaki veri ile çalışmak mümkün olmadığından, örneklenen verilerin pencerelenmesi kaçınılmazdır. Bir veri parçası (N sayıda) analiz için seçildiğinde, orjinal veri pencerelenmelidir. Bir başka deyişle, orjinal veri analiz bölgesinde '1', bu bölgenin dışında '0' olan kare bir pencere foksiyonu ile çarpılmaktadır. Ancak, kare pencerenin kenar kulaklarının yüksek olması istenmeyen bir durumdur. Bu yüzden kullanılacak

pencere fonksiyonun, kenarlarda yumuşak geçişli olması gereklidir [10].

Sık kullanılan pencere fonksiyonları Şekil 2.3.1.'de verilmiştir. Bu tezde Hamming penceresi kullanılacaktır. Kare penceresinin kenar kulakları -13dB iken, Hamming penceresinin -41dB'dir.

Kare : $\omega(n) = 1 \qquad 0 \le n \le N-1$

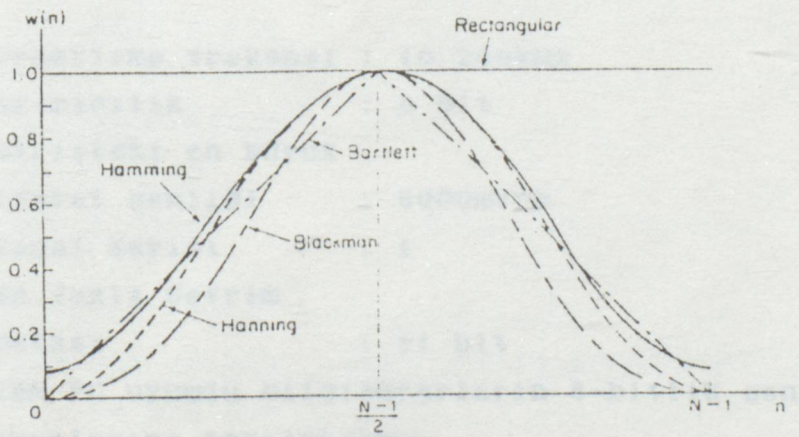
Bortlett : $\omega(n) = \begin{cases} 2.n/(n-1), & 0 \le n \le (N-1)/2 \\ 2-2.n/(N-1), & (N-1)/2 \le n \le N-1 \end{cases}$

Hanning : $\omega(n) = (1/2).[\,1-\cos(2.n./(N-1))\,], \quad 0 \le n \le N-1$

Hamming : $\omega(n) = 0,54-0,46.\cos(2.\pi.n./N-1), \quad 0 \le n \le N-1$

Blackman : $\omega(n) = 0,42-0,50.\cos(2.\pi.n./N-1)+ 0,08.\cos(4.\pi.n./N-1), \quad 0 \le n \le N-1$

Tablo 2.3.1. Sıkça kullanılan pencere fonksiyonları



Şekil 2.3.1. Sıkça kullanılan pencere fonksiyonlarının eğrileri.

III. BÖLÜM

BİLGİSAYAR SİSTEMİNİN DONANIMI

3.1. Giriş

Bu bölümde IBM PC/XT yada PC/AT uyumlu bir bilgisayar ve Ö/S (Örneksel/Sayısal) çeviriciden oluşan bilgisayar sisteminin donanımı Bölüm 3.2.'de, sistemin yazılımı ise Bölüm 3.3.'de açıklanacaktır.

3.2. Bilgisayar Sisteminin Donanımı

Sistemde hazır örneklenmiş işaretlerin alındığı Ö/S çeviricinin örnekleme frekansı 10.240kHz olarak seçilmiştir. Gerek donanımda kullanılan bilgisayar IBM PC/AT uyumlu olduğundan gerekse basitlik ve güvenirlilik açısından, PC uyumlu hazır bir Ö/S çevirici kartı, bilgisayar sisteminde kullanılmıştır. Ö/S çeviricinin gerekli teknik özellikleri Tablo 3.1.'de verilmiştir.

A. Örnekleme frekansı : 10.240kHz
B. Ayırıcılık : 8 Bit
C. Girişteki en büyük işaret genliği : 5000mVpp
D. Kanal sayısı : 1
E. En fazla Çevrim Hatası : ±1 bit
F. IBM PC uyumlu bilgisayarların 8-bitlik genişleme yuvalarına takılabilme

Tablo 3.1. Ö/S Çeviricinin Teknik Özellikleri

Sistemde kullanılan IBM PC/AT uyumlu kişisel bilgisayar aşağıda verilen özelliklere sahiptir.

A) İşlem Birimi :
   20MHz'de çalışan, 32 Bitlik 80386 mikroişlemci birimi

B) Kullanıcıya ait 2MByte RAM

C) Disket Sürücü :
   Bir adet 5-1/4 " floppy disket sürücü, 1.2MByte
   İki adet 40MByte 28msn erişim hızlı hard disk sürücü
   Bir adet 3-1/2" floppy disket sürücü, 1.44MByte

D) Giriş / Çıkış Bağlantıları :
   Tamamen programlanabilir iki adet RS-232 birim
   Bir adet paralel port çıkışı

E) Genişleme Yuvaları :
   Dört adet 16 bit
   İki adet 8 bit

F) Sistem Yazılımı
   MS-DOS 3.3
   QUICK BASIC

G) Görüntü Kartı ve Monitör :
   IBM-EGA uyumlu, renkli grafik adaptörü
   TVM-14 Monitör

3.3. Bilgisayar Sisteminin Yazılımı

Yazılım, Quick Basic version 4.00'de yazılmıştır ve 360kbyte'lık 5" 1/4 DS/DD disket içerisine sığmaktadır.

Yazılımın çalıştırılması IBM/DOS veya MS/DOS 2.0 veya daha büyük versiyonlarla mümkündür ve Hercul Graphic Adaptor ve 640kbyte RAM hafıza ile çalışılacak sistemin desteklenmesi gerekmektedir.

3.3.1. Programın Özellikleri

Program aşağıdaki adımlardan oluşmuştur.

10-30	ana menu
40-50	işaretin zaman domeninde genel bir incelenmesi
1000-1145	işaretin zaman domeninde ayrıntılı incelenmesi
1500-2910	işaretin frekans domeninde ayrıntılı incelenmesi
5000-6900	işaretin pencerelenmesi ve FFT'sinin alınması

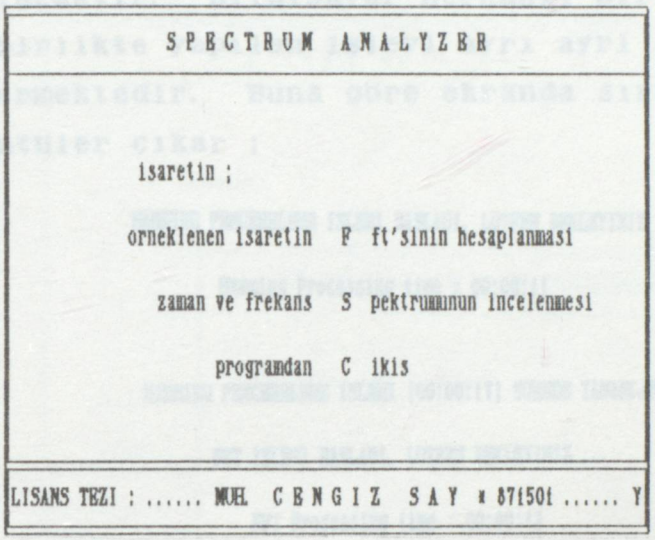
Program çalıştırıldığında ilk olarak ekranda Şekil 3.3.1. belirmektedir. Buna göre eğer,

'F' tuşuna basılırsa örneklenmiş işaretin FFT'sini almak üzere ilgili alt programa,

'S' tuşuna basılırsa örneklenmiş ve FFT'si alınmış işaretin incelenmesi için gerekli alt programa

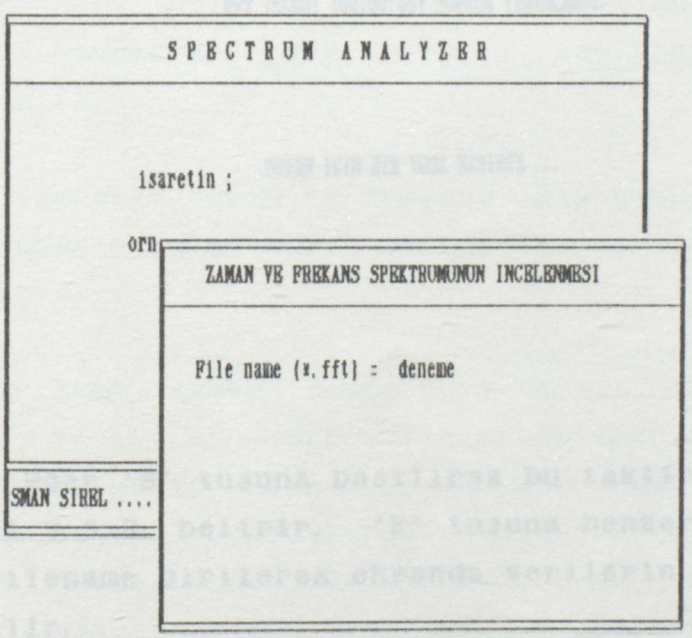
'C' Programdan cıkış alt programına

dallanılacaktır.



Şekil 3.3.1. Ana Menü

'F' tuşuna basıldığında ekrana Şekil 3.3.2 gelir. bu durumda sadece filename yazılarak FFT'si alınacak olan veri dizisi işleme konur.

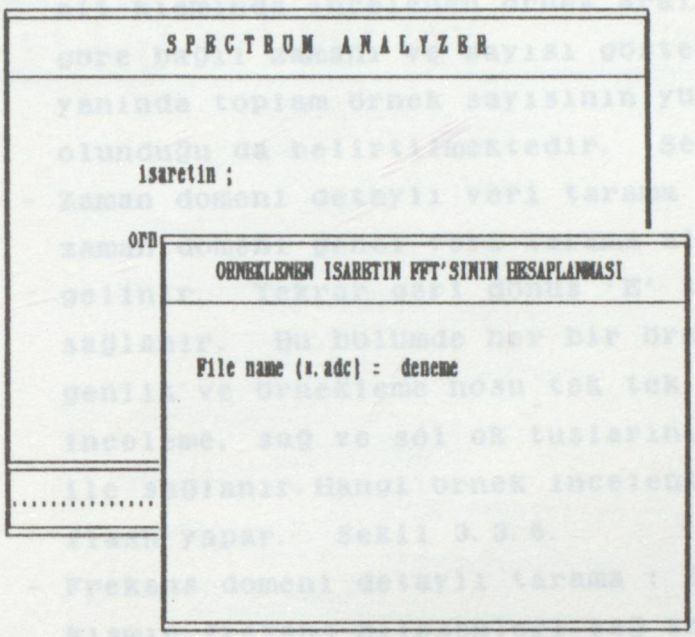


Şekil 3.3.2. FFT hesaplama alt programına geçiş.

Şimdi sırayla pencereleme ve fft işlemleri yapılacaktır. Bilgisayar herhangi bir anda süreleri ile birlikte yapılan işlevi ayrı ayrı ekranda göstermektedir. Buna göre ekranda sırasıyle aşağıdaki görüntüler çıkar ;

```
HAMMING PENCERELEME ISLEMI BASLADI, LUTFEN BEKLEYINIZ ...

Hamming Processing time : 00:00:11


HAMMING PENCERELEME ISLEMI (00:00:17) SUREDE TAMAMLANDI.

FFT ISLEMI BASLADI, LUTFEN BEKLEYINIZ ...

FFT Processing time : 00:00:12


HAMMING PENCERELEME ISLEMI (00:00:17) SUREDE TAMAMLANDI.

FFT ISLEMI (00:00:45) SUREDE TAMAMLANDI.


HAMMING PENCERELEME ISLEMI (00:00:17) SUREDE TAMAMLANDI.

FFT ISLEMI (00:00:45) SUREDE TAMAMLANDI.


DEVAM ICIN BIR TUSA BASINIZ ...
```

Eğer 'S' tuşuna basılırsa bu taktirde ekranda Şekil 3.3.3. belirir. 'F' tuşuna benzer olarak burada da filename girilerek ekranda verilerin incelenmesine geçilir.



Şekil 3.3.3. Zaman ve frekans spektrumunun incelenmesi

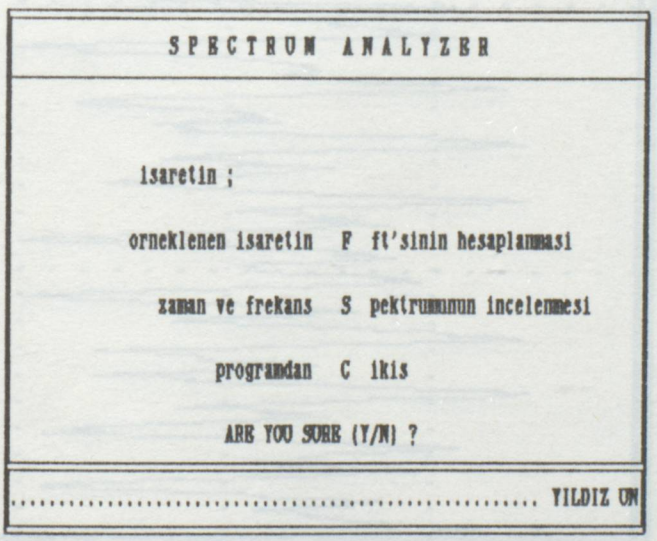
Eğer 'C' tuşuna basılırsa bu defa da ekranda Şekil 3.3.4. belirir. Bilgisayar, eminmisiniz ? diye ikinci kez sorar, eğer cevap 'Y' ise program terkedilir.

işaretin zaman ve frekans spektrumunun incelenmesi sırasında meydana gelen görüntüler de sırasıyle şu şekildedir;

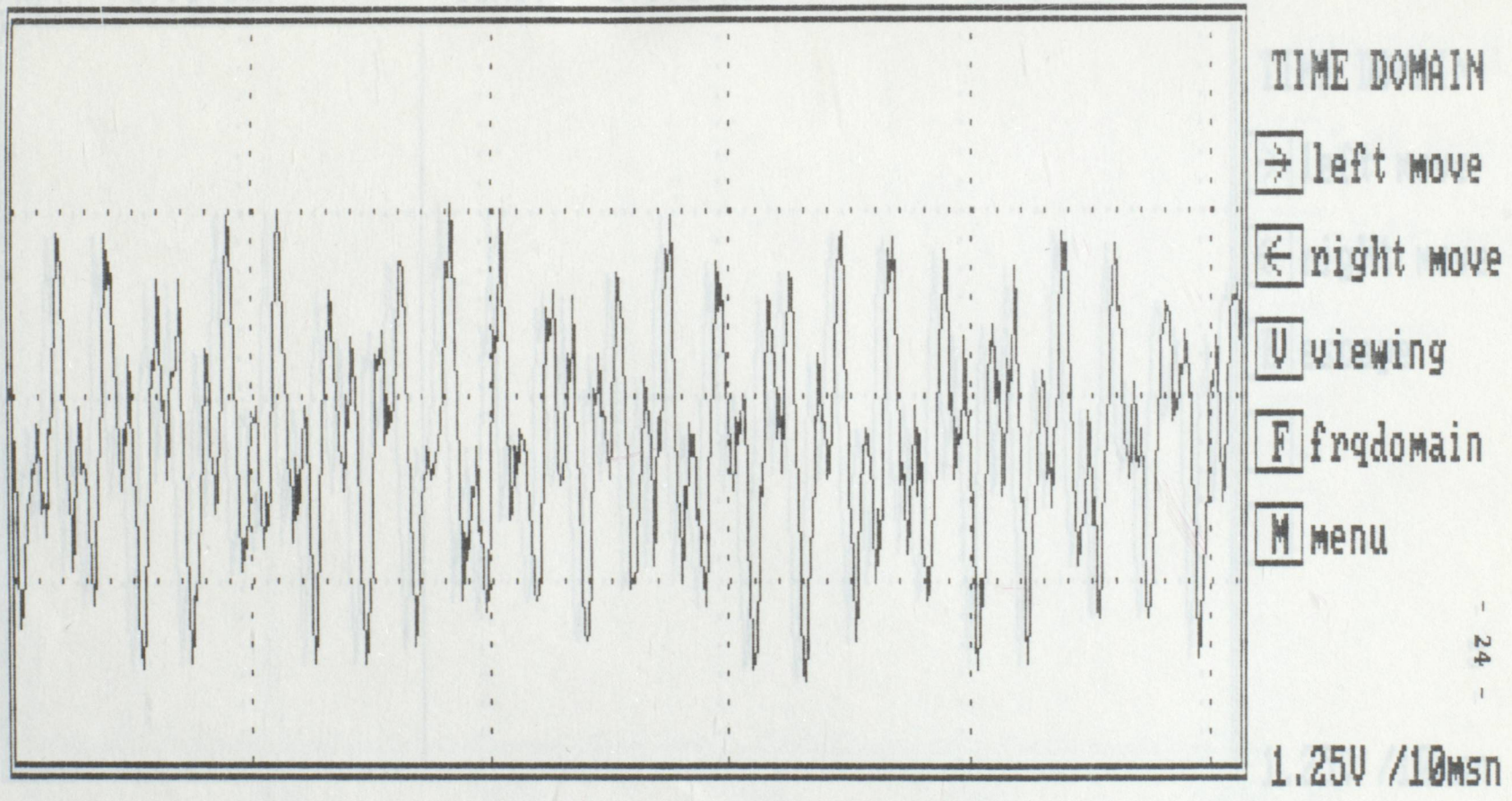
- Zaman domeni genel veri tarama: Bu bölümde sağ ve sol ok tuşları ile örneklenen işaretin istenilen bölümü ekrana görüntülenir. Bir anda bakılan örnek sayısı 512 adettir. Dolayısıyle bu 512 nokta için frekans spekturumu 'F' tuşuna basılarak görülebilir. istenildiğinde ana menüye 'M' tuşuyla geri dönülebilir. Ekrana

gelen 512 örnek üzerinde daha detaylı inceleme olanağı 'V' tuşuna basılarak sağlanır. Ekranın alt kısmında incelenen örnek aralığının başa göre bağıl zamanı ve sayısı gösterilmekte, bunun yanında toplam örnek sayısının yüzde kaçında olunduğu da belirtilmektedir. Şekil 3.3.5.

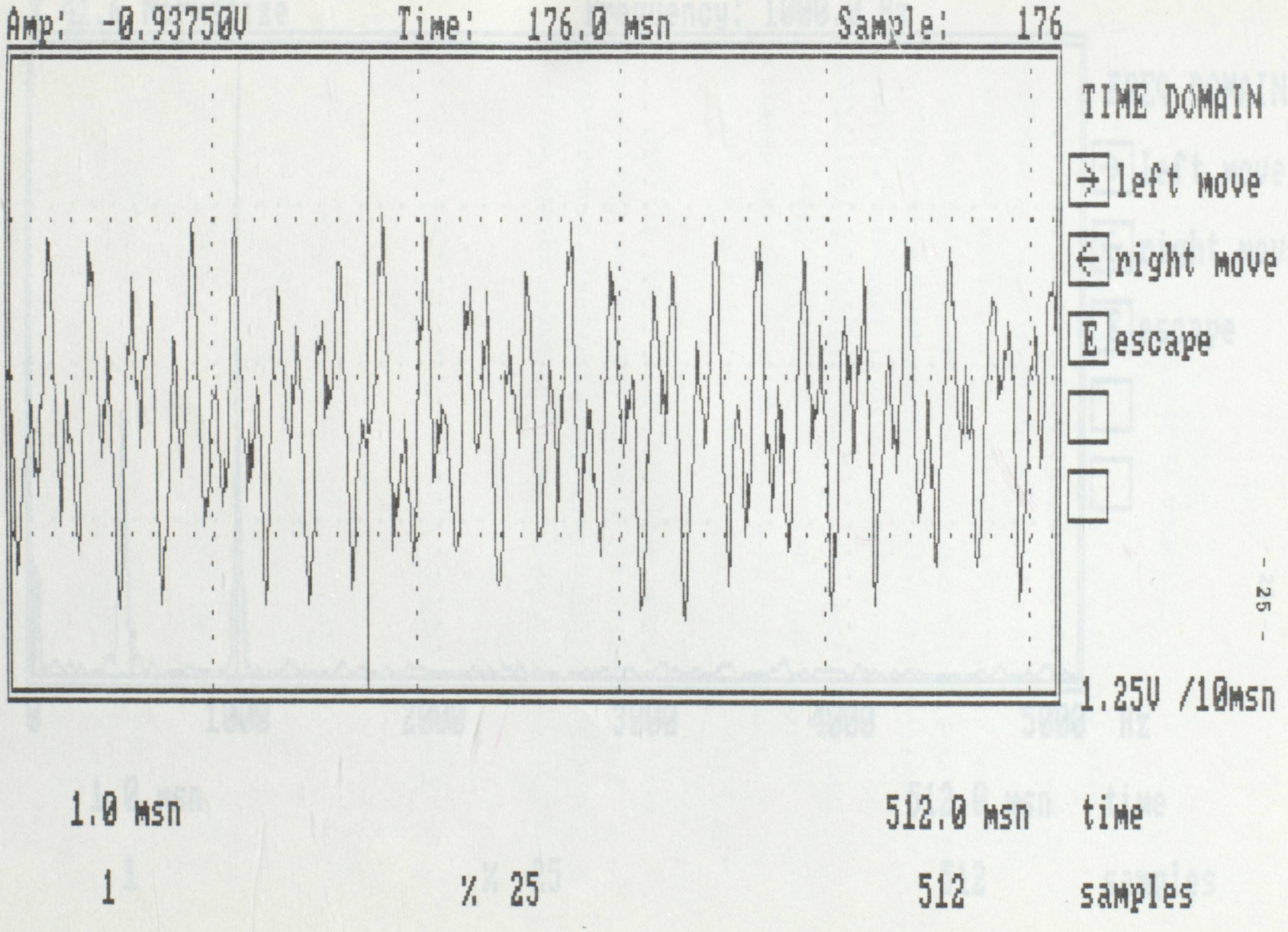
- Zaman domeni detaylı veri tarama : Bu bölüme zaman domeni genel veri tarama alt programından gelinir. Tekrar geri dönüş 'E' tuşuna basılarak sağlanır. Bu bölümde her bir örneğe ait zaman, genlik ve örnekleme nosu tek tek incelenebilir. inceleme, sağ ve sol ok tuşlarının kullanılması ile sağlanır. Hangi örnek inceleniliyorsa o bölge flash yapar. Şekil 3.3.6.
- Frekans domeni detaylı tarama : 512 örneklik kısmın frekans bileşenleri sağ ve sol ok tuşları kullanarak incelenebilir. Zaman domenine dönüş 'E' tuşuna basılarak sağlanır. Şekil 3.3.7.



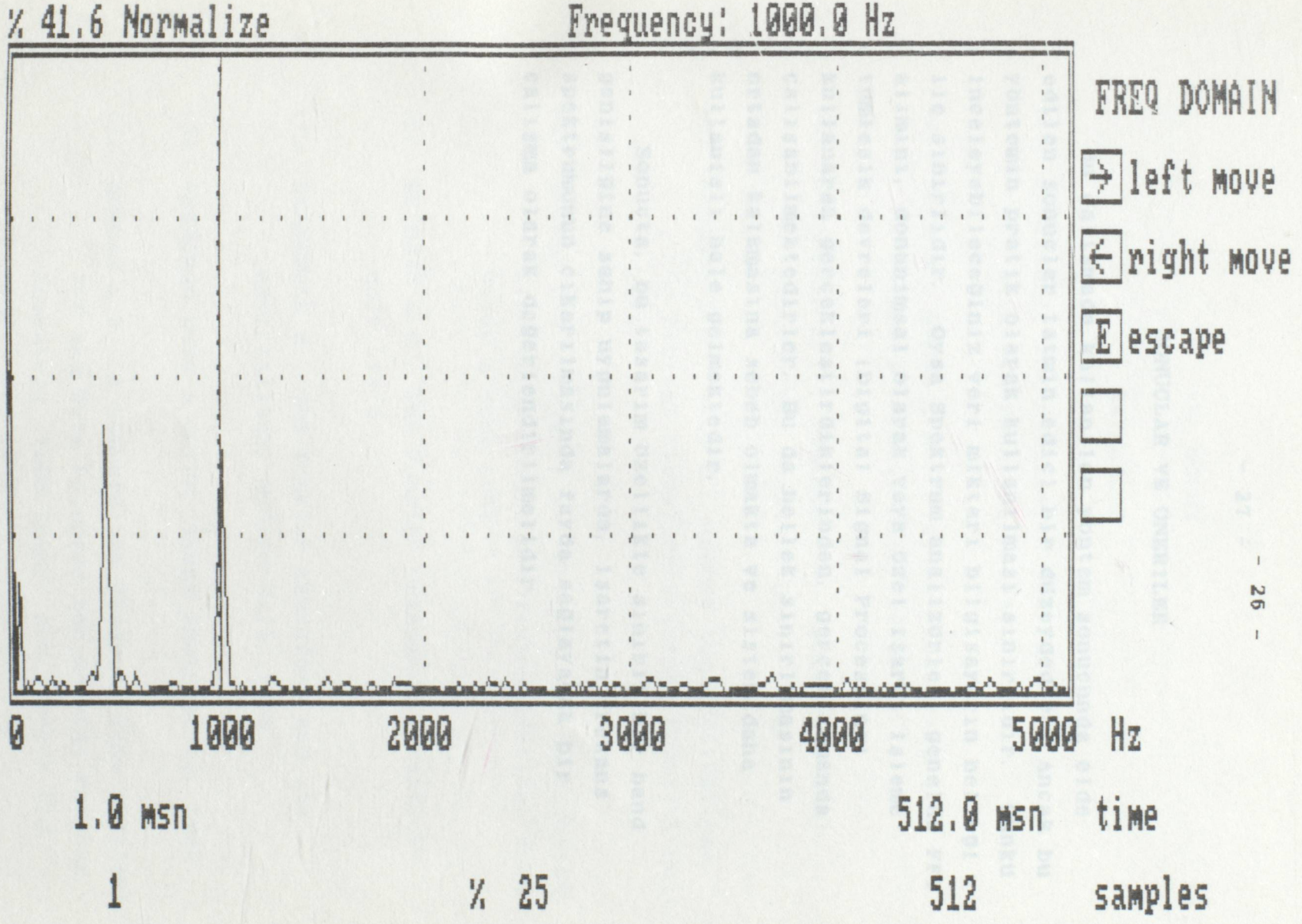
Şekil 3.3.4. Programdan çıkış



Sekil 3.3.5. Zaman domeninde genel tarama



Sekil 3.3.6. Zaman domeni detayli veri tarama



Şekil 3.3.7. Frekans domeni detaylı tarama

SONUÇLAR VE ONERİLER

Bu çalışmada kullanılan yöntem sonucunda elde edilen sonuçlar tatmin edici bir düzeydedir. Ancak bu yöntemin pratik olarak kullanılması sınırlıdır. Çünkü inceleyebileceğiniz veri miktarı bilgisayarın belleği ile sınırlıdır. Oysa Spektrum analizörler, genelde FFT alımını, donanımsal olarak veya özel işaret işleme tümleşik devreleri (Digital Signal Processing) kullanarak gerçekleştirdiklerinden, gerçek zamanda çalışabilmektedirler. Bu da bellek sınırlamasının ortadan kalkmasına sebeb olmakta ve sistem daha kullanışlı hale gelmektedir.

Sonuçta, bu tasarım özellikle sınırlı bir band genişliğine sahip uygulamalarda, işaretin frekans spektrumunun çıkarılmasında fayda sağlayacak bir çalışma olarak değerlendirilmelidir.

IV. BOLUM

KAYNAKLAR

(1) PANAYIRCI, E., Modülasyon teorisi, I.T.U., Elektrik Fakültesi, 1985.

(2) GOLD B., RADER C.M., Digital Processing of Signals, McGraw-Hill, 1969.

(3) OPPENHEIM A.V., SCHAFER R.W., Digital Signal Processing, Prentice-Hall, 1975.

(4) RABINER L.R., GOLD B., Theory And Applications Of Digital Signal Processing, Prentice-Hall, 1975.

(5) BURRUS C.S., PARKS T.W., DFT/FFT And Convolution Algorithms-Theory And Implementation, John Wiley & Sons, 1985

(6) BRIGRAM E.O., The Fourier Transform, Prentice-Hall, 1974.

(7) BRACEWELL R., The Fourier Transform and Its Applications, McGraw-Hill, 1965

(8) TMS32020 USER'S GUIDE, Texas Instruments, 1985

(9) MORRIS L.R., A Comporative Study of Time Efficent FFT and WFTA Programs for General Purpose Computers, IEEE Trans. Acu. Speech Sig. Proc., Ap. 1978

(10) OPENHEIM A.V., Digital Signal Processing Applications, New Jersey, 1979

(11) RABINER L.R., On Use of Symmetry in FFT Computition, IEEE Trans. Acu. Speech Sig. Proc., Aug. 1982.

(12) PREUSS R.D., Very Fast Computition of Radix-2 DFT, IEEE Trans. Acu Speech Sig. Proc., Aug. 1982.

(13) BERGLAND G.D., A Fast Fourier Transform Algorithm using Base-8 Iterations, Mathematics of Computition, Vol 22, No 102, 275-279 (April 1968).

(14) COOLEY J.W., LEWIS P.A.W., WELCH P.D., The Fast Fourier Transform Algorithm - Programming Considerations in the Calculation Sine, Cosine, and Laplace Transforms, Journal of Sound Vibration, Vol 12, 315-337, July 1969.

EK-A
QUICK BASIC DİLİNDE YAZILMIŞ
SPEKTRUM ANALİZOR PROGRAMI

```
1     DIM BUF1%(10000)
2     DIM BUFFER%(10000)
3     DIM BUF2%(10000)
      DIM H AS STRING * 1
      DIM L AS STRING * 1

10    SCREEN 0: CLS : LOCATE , , 0
      A = 1: A$ = STRING$(65, 46) + "..... YILDIZ UNIVERSITESI "
            A$ = A$ + "..... FEN BILIMLERI ENSTITUSU "
            A$ = A$ + "..... YUKSEK LISANS TEZI : "
            A$ = A$ + "..... MUH. C E N G I Z  S A Y  * 871501 "
            A$ = A$ + "..... YONETICI: Yr. Doc. Dr. OSMAN SIREL ..."
            A$ = A$ + STRING$(65, 46)
      PRINT : PRINT
      PRINT "
      PRINT "                    S P E C T R U M  A N A L Y Z E R                    "
      PRINT "
      PRINT "
      PRINT "
      PRINT "                 isaretin :                                             "
      PRINT "
      PRINT "                orneklenen isaretin "; : COLOR 0, 7: PRINT " F "; : COLOR 7, 0:
      PRINT " ft'sinin hesaplanmasi       "
      PRINT "
      PRINT "                  zaman ve frekans "; : COLOR 0, 7: PRINT " S "; : COLOR 7, 0:
      PRINT " pektrumunun incelenmesi     "
      PRINT "
      PRINT "                       programdan "; : COLOR 0, 7: PRINT " C "; : COLOR 7, 0:
      PRINT " ikis                        "
      PRINT "
      PRINT "
      PRINT "
      PRINT "
      PRINT "

20    A = A + 1: IF A = 235 THEN A = 1
      P$ = MID$(A$, A, 65): LOCATE 18, 8: PRINT B$
      FOR Q = 1 TO 300: NEXT Q
      Q$ = INKEY$: IF Q$ = "" THEN GOTO 20
      IF Q$ = "F" OR Q$ = "f" THEN GOTO 30
      IF Q$ = "S" OR Q$ = "s" THEN GOTO 40
      IF Q$ = "C" OR Q$ = "c" THEN LOCATE 16, 30: PRINT "ARE YOU SURE (Y/N) ? ": GOTO 25

23    SOUND 100, 10: GOTO 20

25    Q$ = INKEY$: IF Q$ = "" THEN GOTO 25
      IF Q$ = "Y" OR Q$ = "y" THEN CLS : PRINT "  SPECTRUM ANALYZER terminated....  ": SYSTEM
      LOCATE 16, 30: PRINT "                    "
      GOTO 23
```

```
30    LOCATE 10, 23: PRINT "╔═══════════════════════════════════════╗"
      LOCATE 11, 23: PRINT "║ ORNEKLENEN ISARETIN FFT'SININ HESAPLANMASI ║"
      LOCATE 12, 23: PRINT "╠═══════════════════════════════════════╣"
      LOCATE 13, 23: PRINT "║                                       ║"
      LOCATE 14, 23: PRINT "║                                       ║"
      LOCATE 15, 23: PRINT "║                                       ║"
      LOCATE 16, 23: PRINT "║                                       ║"
      LOCATE 17, 23: PRINT "║                                       ║"
      LOCATE 18, 23: PRINT "║                                       ║"
      LOCATE 19, 23: PRINT "║                                       ║"
      LOCATE 20, 23: PRINT "║                                       ║"
      LOCATE 21, 23: PRINT "║                                       ║"
      LOCATE 22, 23: PRINT "╚═══════════════════════════════════════╝"
      LOCATE 14, 26: PRINT " File name (*.adc) : ?": LOCATE 14, 47, 1, 0, 7
      Y = 14: Y = 47: GOSUB 8100: ON ERROR GOTO 9500: GOSUB 5000: GOTO 10

40    LOCATE 10, 23: PRINT "╔═══════════════════════════════════════╗"
      LOCATE 11, 23: PRINT "║ ZAMAN VE FREKANS SPEKTRUMUNUN INCELENMESI ║"
      LOCATE 12, 23: PRINT "╠═══════════════════════════════════════╣"
      LOCATE 13, 23: PRINT "║                                       ║"
      LOCATE 14, 23: PRINT "║                                       ║"
      LOCATE 15, 23: PRINT "║                                       ║"
      LOCATE 16, 23: PRINT "║                                       ║"
      LOCATE 17, 23: PRINT "║                                       ║"
      LOCATE 18, 23: PRINT "║                                       ║"
      LOCATE 19, 23: PRINT "║                                       ║"
      LOCATE 20, 23: PRINT "║                                       ║"
      LOCATE 21, 23: PRINT "║                                       ║"
      LOCATE 22, 23: PRINT "╚═══════════════════════════════════════╝"
      LOCATE 14, 26: PRINT " File name (*.fft) : ?": LOCATE 14, 47, 1, 0, 7
      Y = 14: Y = 47: GOSUB 8100: ON ERROR GOTO 9500

34    SCREEN 2
      OPEN V$ + ".ADC" FOR BINARY AS #1
      MAXUZ = 512 * INT((LOF(1) - 1) / 512): FREQ = 1 / 10240: SPLFIRST = 1: SPLLAST = 512: BOL = 1
      TIMFRST = 1: TIMLAST = BOL * 512: AMP = 0: SPL = 1: TIM = SPL * FREQ: YUZDE = SPLLAST * 100 / MAXUZ
35    OLCUM = 1.25
      LINE (0, 8)-(517, 8): LINE -(517, 141): LINE -(0, 141): LINE -(0, 8)
      LINE (2, 10)-(515, 10): LINE -(515, 139): LINE -(2, 139): LINE -(2, 10)
      LOCATE 3, 67: PRINT "TIME DOMAIN"
      LOCATE 5, 67: PRINT "  left move":
      LOCATE 7, 67: PRINT "  right move"
      LOCATE 9, 67: PRINT "V viewing"
      LOCATE 11, 67: PRINT "F frqdomain"
      LOCATE 13, 67: PRINT "M menu"
      LOCATE 18, 66: PRINT USING "##.##"; OLCUM; : PRINT "V /"; : PRINT USING "##"; FREQ * 100000; : PRINT "msn"
      LINE (521, 30)-(540, 30): LINE -(540, 40): LINE -(521, 40): LINE -(521, 30)
      LINE (521, 46)-(540, 46): LINE -(540, 56): LINE -(521, 56): LINE -(521, 46)
      LINE (521, 62)-(540, 62): LINE -(540, 72): LINE -(521, 72): LINE -(521, 62)
```

```
    LINE (521, 78)-(540, 78): LINE -(540, 88): LINE -(521, 88): LINE -(521, 78)
    LINE (521, 94)-(540, 94): LINE -(540, 104): LINE -(521, 104): LINE -(521, 94)
    LINE (525, 35)-(535, 35): LINE -(529, 32): LINE (535, 35)-(529, 38)
    LINE (535, 51)-(525, 51): LINE -(531, 48): LINE (525, 51)-(531, 54)
    FOR AX = 2 TO 502 STEP 100: FOR BX = 8 TO 142 STEP 5: PSET (AX, BX): NEXT BX: NEXT AX
    FOR AX = 43 TO 139 STEP 32: FOR BX = 3 TO 515 STEP 10: PSET (BX, AX): NEXT BX: NEXT AX
    IF OX = 0 THEN GET (2, 10)-(515, 139), BUFFERX
    GOSUB 46:
41  KEY(12) OFF: KEY(13) OFF
    ON KEY(12) GOSUB 7500:
    ON KEY(13) GOSUB 7000: KEY(12) ON: KEY(13) ON
43  Q$ = INKEY$: IF Q$ = "" THEN 43
    FOR A = 1 TO 500: NEXT A
    IF Q$ = "M" OR Q$ = "m" THEN CLOSE #1: SOUND 1700, 5: GOTO 10
    IF Q$ = "V" OR Q$ = "v" THEN SOUND 1700, 5: GOSUB 7510: GOTO 1000
    IF Q$ = "F" OR Q$ = "f" THEN CLOSE #1: SOUND 1700, 5: GOTO 1500
    GOTO 43
46  LOCATE 21, 1:
    PRINT USING "#####.#"; TIMFRST; : PRINT " msn"; TAB(53);
    PRINT USING "#####.#"; TIMLAST; : PRINT " msn   time": PRINT : PRINT " ";
    PRINT USING "#######"; SPLFIRST; : PRINT TAB(29); "X ";
    PRINT USING "###"; YUZDE; : PRINT TAB(54);
    PRINT USING "#######"; SPLLAST; : PRINT "       samples"
    ZX = 3
    PSET (1, 74)
    FOR UQ = SPLFIRST TO SPLFIRST + 511
    GET #1, UQ, L
    LIX = ASC(L)
    SAYIX = INT(LIX / 2) + 11
50  LINE -(ZX, SAYIX): ZX = ZX + 1
    NEXT UQ
    LOCATE 21, 1:
    PRINT USING "#####.#"; TIMFRST; : PRINT " msn"; TAB(53);
    PRINT USING "#####.#"; TIMLAST; : PRINT " msn   time": PRINT : PRINT " ";
    PRINT USING "#######"; SPLFIRST; : PRINT TAB(29); "X ";
    PRINT USING "###"; YUZDE; : PRINT TAB(54);
    PRINT USING "#######"; SPLLAST; : PRINT "       samples"
    RETURN


1000 KEY(12) OFF: KEY(13) OFF
    GET (2, 10)-(515, 156), BUF1X
    LOCATE 9, 67: PRINT "E escape  "
    FOR GEC = 10 TO 14: LOCATE GEC, 67: PRINT "          ": NEXT GEC
    LINE (521, 62)-(540, 62): LINE -(540, 72): LINE -(521, 72): LINE -(521, 62)
    LINE (521, 78)-(540, 78): LINE -(540, 88): LINE -(521, 88): LINE -(521, 78)
    LINE (521, 94)-(540, 94): LINE -(540, 104): LINE -(521, 104): LINE -(521, 94)
    LINE (525, 35)-(535, 35): LINE -(529, 32): LINE (535, 35)-(529, 38)
    LINE (535, 51)-(525, 51): LINE -(531, 48): LINE (525, 51)-(531, 54)
    SPL = SPLFIRST + 255: TIM = SPL: SPLFIR1 = SPL: ZX = 256 + 2
```

```
1010  GOSUB 1145
      ON TIMER(1) GOSUB 7900: TIMER ON: FLAG = -1
      ON KEY(12) GOSUB 7800:
      ON KEY(13) GOSUB 7600: KEY(12) ON: KEY(13) ON

1043  Q$ = INKEY$: IF Q$ = "" THEN 1043
      FOR A = 1 TO 500: NEXT A
      IF Q$ = "E" OR Q$ = "e" THEN SOUND 1700, 5: GOTO 1045
      GOTO 1043

1045  KEY(12) OFF: KEY(13) OFF
      TIMER OFF: PUT (2, 10), BUFFER%, PSET: LOCATE 1, 1
      PRINT STRING$(70, 32); OX = 1: GOTO 35

1145  GET #1, SPLFIR1, H
      HI% = ASC(H)
      SAYI% = INT(HI% / 2) + 11
      AMP = (128 - HI%) * 2.5 / 64
      LOCATE 1, 1:
      PRINT "Amp: "; : PRINT USING "+##.#####"; AMP; : PRINT "V"; :
      PRINT TAB(25); "Time: "; : PRINT USING "#####.#"; TIM; : PRINT " msn";
      PRINT TAB(52); "Sample: "; : PRINT USING "######"; SPL;
      RETURN

1500  KEY(12) OFF: KEY(13) OFF
      OPEN V$ + ".FFT" FOR BINARY AS #3

1535  CLS
      LOCATE 21, 1:
      PRINT USING "#####.#"; TIMFRST; : PRINT " msn"; TAB(53);
      PRINT USING "#####.#"; TIMLAST; : PRINT " msn   time": PRINT : PRINT " ";
      PRINT USING "######"; SPLFIRST; : PRINT TAB(29); "% ";
      PRINT USING "###"; YUZDE; : PRINT TAB(54);
      PRINT USING "######"; SPLLAST; : PRINT "      samples"
      LOCATE 5, 67: PRINT "  left move";
      LOCATE 7, 67: PRINT "  right move"
      LOCATE 3, 67: PRINT "FREQ DOMAIN"
      LOCATE 9, 67: PRINT "E escape  "
      LOCATE 19, 1: PRINT "0          1000        2000        3000        4000        5000 Hz"
      LINE (521, 30)-(540, 30): LINE -(540, 40): LINE -(521, 40):
      LINE -(521, 30): LINE (521, 46)-(540, 46): LINE -(540, 56):
      LINE -(521, 56): LINE -(521, 46): LINE (521, 62)-(540, 62):
      LINE -(540, 72): LINE -(521, 72): LINE -(521, 62)
      LINE (521, 78)-(540, 78): LINE -(540, 88): LINE -(521, 88):
      LINE -(521, 78): LINE (521, 94)-(540, 94): LINE -(540, 104):
      LINE -(521, 104): LINE -(521, 94): LINE (525, 35)-(535, 35):
      LINE -(529, 32): LINE (535, 35)-(529, 38): LINE (535, 51)-(525, 51):
      LINE -(531, 48): LINE (525, 51)-(531, 54)
      FOR AX = 2 TO 502 STEP 100: FOR BX = 8 TO 142 STEP 5
      PSET (AX, BX): NEXT BX: NEXT AX
```

```
    FOR AX = 43 TO 139 STEP 32: FOR BX = 3 TO 515 STEP 10
    PSET (BX, AX): NEXT BX: NEXT AX
    FFTFIRST = ((SPLFIRST - 1) / 2) + 1
    LINE (0, 8)-(517, 8): LINE -(517, 141): LINE -(0, 141): LINE -(0, 8)
    LINE (2, 10)-(515, 10): LINE -(515, 139): LINE -(2, 139): LINE -(2, 10)
    ZX = 3
    PSET (1, 74)
    FOR UQ = FFTFIRST TO FFTFIRST + 256
    GET #3, UQ, L
    LIX = ASC(L)
    SAYIX = INT((256 - LIX) / 2) + 11
    LINE -(ZX, SAYIX): ZX = ZX + 2
    NEXT UQ

1800
    GET (2, 10)-(515, 139), BUF2X
    LINE (521, 62)-(540, 62): LINE -(540, 72): LINE -(521, 72): LINE -(521, 62)
    LINE (521, 78)-(540, 78): LINE -(540, 88): LINE -(521, 88): LINE -(521, 78)
    LINE (521, 94)-(540, 94): LINE -(540, 104): LINE -(521, 104): LINE -(521, 94)
    LINE (525, 35)-(535, 35): LINE -(529, 32): LINE (535, 35)-(529, 38)
    LINE (535, 51)-(525, 51): LINE -(531, 48): LINE (525, 51)-(531, 54)
    FFT = FFTFIRST + 127: FFTFREQ = 0: ZX = 255 + 2

1810 GOSUB 1945
    ON TIMER(1) GOSUB 2900: TIMER ON: FLAG = -1
    ON KEY(12) GOSUB 2800:
    ON KEY(13) GOSUB 2600: KEY(12) ON: KEY(13) ON

1843 Q$ = INKEY$: IF Q$ = "" THEN 1843
    FOR A = 1 TO 500: NEXT A
    IF Q$ = "E" OR Q$ = "e" THEN SOUND 1700, 5: GOTO 1845
    GOTO 1843

1845 KEY(12) OFF: KEY(13) OFF
    TIMER OFF: PUT (2, 10), BUFFERX, PSET: LOCATE 1, 1
    PRINT STRING$(70, 32): OX = 1: CLOSE #3:
    LOCATE 19, 1: PRINT STRING$(70, 32)
    OPEN V$ + ".ADC" FOR BINARY AS #1: GOTO 35

1945 GET #3, FFT, H
    HIX = ASC(H)
    SAYIX = INT((256 - HIX) / 2) + 11
    DB = (HIX / 2.55)
    HZ = (ZX - 2) * 10
    LOCATE 1, 1
    PRINT USING "X###.#"; DB; : PRINT " Normalize"; :
    PRINT TAB(35); "Frequency: "; : PRINT USING "####.#"; HZ - 10; : PRINT " Hz";
    RETURN
```

```
2600  KEY(13) OFF: KEY(12) OFF: TIMER OFF: FFT = FFT + 1: ZX = ZX + 2
      IF FFT = FFTFIRST + 256 THEN FFT = FFTFIRST: ZX = 3

2610  GOSUB 1945: TIMER ON: KEY(12) ON: KEY(13) ON: RETURN


2800  KEY(13) OFF: KEY(12) OFF: TIMER OFF: FFT = FFT - 1: ZX = ZX - 2
      IF FFT < FFTFIRST THEN FFT = FFTFIRST + 255: ZX = 513

2810  GOSUB 1945: TIMER ON: KEY(12) ON: KEY(13) ON: RETURN

2900  KEY(12) OFF: KEY(13) OFF:

      FLAG = FLAG * -1
      IF FLAG = -1 THEN PUT (2, 10), BUF2%, PSET:  GOTO 2910
      LINE (ZX, 11)-(ZX, 138): PRESET (ZX, SAYI%)

2910  KEY(13) ON: KEY(12) ON: RETURN

4000  LOCATE 15, 19: COLOR 0, 7: PRINT " Hamming Processing time = "; TIME$; " ": COLOR 7, 0: RETURN
4010  LOCATE 17, 20: COLOR 0, 7: PRINT " FFT Processing time = "; TIME$; " ": COLOR 7, 0: RETURN

5000  CLS
      DIM D AS STRING * 1
      TIME$ = "00:00:00:"
      ON TIMER(1) GOSUB 4000:
      TIMER ON: BEEP: GOSUB 4000

'*****************************************************************************************************************
'                          P E N C E R E L E M E
'*****************************************************************************************************************

5010  LOCATE 13, 10: PRINT " HAMMING PENCERELEME ISLEMI BASLADI, LUTFEN BEKLEYINIZ ..."
      OPEN V$ + ".ADC" FOR BINARY AS #1
      OPEN V$ + ".PCR" FOR BINARY AS #2
      LIM = 512 * INT(LOF(1) / 512)
      FOR IX = 1 TO LIM STEP 512
      KX = 0
      FOR JX = IX TO IX + 511
      GET #1, JX, D
      SX = ASC(D)
      SX = SX - 128
      S1 = SX * (.54 - .46 * COS(2 * 3.14159 * KX / 511))
      S1 = S1 + 128
      S1% = INT(S1)
      D = CHR$(S1%)
      PUT #2, JX, D
      KX = KX + 1
      NEXT JX
      NEXT IX
      CLOSE #1: CLOSE #2
```

```
    TIMER OFF: BEEP
    CLS : LOCATE 13, 10: PRINT "HAMMING PENCERELEME ISLEMI ("; TIME$; ") SUREDE TAMAMLANDI."
    TIME$ = "00:00:00"
    ON TIMER(1) GOSUB 4010
    TIMER ON: GOSUB 4010

'*************************************************************************
'                          F F T   I S L E M I
'*************************************************************************

    LOCATE 15, 17: PRINT "FFT ISLEMI BASLADI, LUTFEN BEKLEYINIZ ..."
    MX = 9: NX = 512: PI = 3.14159: NV2% = NX / 2: NM1% = NX - 1: SX = 1: DIM XI(512), XR(512)
    OPEN V$ + ".PCR" FOR BINARY AS #2
    OPEN V$ + ".FFT" FOR BINARY AS #3
    D = CHR$(0)
    PUT #3, 1, D
    LIM = 512 * INT(LOF(2) / 512)
    FOR Q = 1 TO LIM STEP 512
    VC% = 1
    FOR A% = Q TO Q + 511
    GET #2, A%, D
    XI(VC%) = ASC(D)
    XR(VC%) = XI(VC%)
    VC% = VC% + 1
    NEXT A%
    J% = 1
    FOR I% = 1 TO NM1%
    IF I% >= J% THEN GOTO 5020
    TR = XR(J%)
    XR(J%) = XR(I%)
    XR(I%) = TR
    XI(J%) = XI(I%)
    XI(I%) = TR

5020 K% = NV2%

5030 IF K% >= J% THEN GOTO 5070
    J% = J% - K%
    K% = K% / 2
    GOTO 5030

5070 J% = J% + K%
    NEXT I%
    FOR WJ% = 1 TO MX
    LE% = 2 ^ WJ%
    LE1% = LE% / 2
    UR = 1: UI = 0
    WR = COS(PI / LE1%): WI = -SIN(PI / LE1%)
    FOR J% = 1 TO LE1%
    FOR I% = J% TO NX STEP LE%
```

```
    IPX = IX + LEIX
    TR = XR(IPX) * UR - XI(IPX) * UI
    TI = XR(IPX) * UI + XI(IPX) * UR
    XR(IPX) = XR(IX) - TR
    XI(IPX) = XI(IX) - TI
    XR(IX) = XR(IX) + TR
    XI(IX) = XI(IX) + TI
    NEXT IX
    TR = UR * WR - UI * WI
    TI = UR * WI + UI * WR
    UR = TR: UI = TI
    NEXT JX
    NEXT WJX
    IF Q <> 1 THEN D = CHR$(0): PUT #3, , D
    FOR HQX = 2 TO 256
    ZX = INT((((XR(HQX) ^ 2) + (XI(HQX) ^ 2)) ^ .5) * .015625#)
    IF ZX > 255 THEN ZX = 255
    IF ZX < 0 THEN ZX = 0
    D = CHR$(ZX)
    PUT #3, , D
    NEXT HQX
    NEXT Q
    CLOSE
    TIMER OFF
    LOCATE 17, 20: PRINT "                                    "
    LOCATE 15, 17: PRINT "FFT ISLEMI ("; TIME$; ") SUREDE TAMAMLANDI.          "
    BEEP
    COLOR 0, 7: LOCATE 20, 20: PRINT " DEVAM ICIN BIR TUSA BASINIZ ... "

6900 MB$ = INKEY$: IF MB$ = "" THEN 6900
    COLOR 7, 0: RETURN

7000
7100 KEY(13) OFF: KEY(12) OFF: IF MAXUZ <= SPLFIRST + 512 THEN GOTO 7103
    SPLFIRST = SPLFIRST + 512: BOL = BOL + 1: GOTO 7105

7103 SPLFIRST = 1: BOL = 1

7105 SPLLAST = SPLFIRST + 511: TIMFRST = SPLFIRST: TIMLAST = SPLLAST
    YUZDE = SPLLAST * 100 / MAXUZ: PUT (2, 10), BUFFERX, PSET
    GOSUB 46: GOSUB 7510: KEY(12) ON: KEY(13) ON: RETURN

7500 KEY(12) OFF: KEY(13) OFF: IF 512 >= SPLFIRST THEN GOTO 7503
    SPLFIRST = SPLFIRST - 512: BOL = BOL - 1: GOTO 7506

7503 SPLFIRST = MAXUZ - 512: BOL = 1 + INT(MAXUZ / 512)

7506 SPLLAST = SPLFIRST + 511: TIMFRST = SPLFIRST: TIMLAST = SPLLAST
    YUZDE = SPLLAST * 100 / MAXUZ: PUT (2, 10), BUFFERX, PSET: GOSUB 46:
    GOSUB 7510: KEY(12) ON: KEY(13) ON: RETURN
```

```
7510  FOR A = 0 TO 150: Q$ = INKEY$: NEXT A: RETURN

7600  KEY(13) OFF: KEY(12) OFF: TIMER OFF: SPL = SPLFIRI + 1: ZX = ZX + 1
      IF SPL = SPLFIRST + 512 THEN SPL = SPLFIRST: ZX = 3: GOTO 7620
      TIM = SPL: SPLFIRI = SPL
7610  GOSUB 1145: TIMER ON: KEY(12) ON: KEY(13) ON: RETURN
7620  TIM = SPL: SPLFIRI = SPL: GOTO 7610

7800  KEY(13) OFF: KEY(12) OFF: TIMER OFF: SPL = SPLFIRI - 1: ZX = ZX - 1
      IF SPL < SPLFIRST THEN SPL = SPLFIRST + 511: ZX = 513: GOTO 7820
      TIM = SPL: SPLFIRI = SPL
7810  GOSUB 1145: TIMER ON: KEY(12) ON: KEY(13) ON: RETURN

7820  TIM = SPL: SPLFIRI = SPL: GOTO 7810

7900  KEY(12) OFF: KEY(13) OFF: FLAG = FLAG * -1
      IF FLAG = -1 THEN PUT (2, 10), BUF1%, PSET:  GOTO 7910
      LINE (ZX, 11)-(ZX, 138): PRESET (ZX, SAY1%)

7910  KEY(13) ON: KEY(12) ON: RETURN

8100  V$ = " ": E = 1
8200  W$ = INKEY$: IF W$ = "" THEN 8200
8210  LOCATE , , 0
8220  IF W$ = CHR$(13) AND V$ = " " THEN 8100
8300  IF W$ = CHR$(13) THEN V$ = RIGHT$(V$, LEN(V$) - 1): RETURN
8350  IF W$ = CHR$(27) THEN GOTO 10
8400  IF W$ <> CHR$(8) THEN GOTO 8800
8500  E = E - 1: IF E = 0 THEN 8100
8700  V$ = LEFT$(V$, E): GOTO 9100
8800  IF E = 22 THEN GOTO 9100
8900  E = E + 1: V$ = V$ + W$: IF LEN(V$) > 22 THEN E = 22
9100  LOCATE X, Y: PRINT V$; STRING$(23 - LEN(V$), 32): GOTO 8200
9500  SOUND 100, 10: ON ERROR GOTO 0: GOTO 10
```

VII. BÖLÜM

ÖZGEÇMİŞ

Yazar, 1966 yılında İstanbul'da doğdu. Lise öğrenimini İSTANBUL / Fenerbahçe Lisesi'nde, lisans öğrenimini, İ.T.Ü. Elektronik ve Haberleşme Bölümü'nde tamamladı. 1987-1988 Öğretim yılında Yıldız Üniversitesi, Fen Bilimleri Enstitüsü, Elektronik Programı, Elektronik Anabilim Dalı'nda Yüksek Lisans'a başladı.