

**T.C.
MİLLÎ EĞİTİM BAKANLIĞI**



MEGEP

**(MESLEKİ EĞİTİM VE ÖĞRETİM SİSTEMİNİN
GÜÇLENDİRİLMESİ PROJESİ)**

**ENDÜSTRİYEL OTOMASYON
TEKNOLOJİLERİ**

SAYISAL İŞARET İŞLEME-1

Ankara 2007

Milli Eğitim Bakanlığı tarafından geliştirilen modüller;

- Talim ve Terbiye Kurulu Başkanlığının 02.06.2006 tarih ve 269 sayılı Kararı ile onaylanan, Mesleki ve Teknik Eğitim Okul ve Kurumlarında kademeli olarak yaygınlaştırılan 42 alan ve 192 dala ait çerçeve öğretim programlarında amaçlanan mesleki yeterlikleri kazandırmaya yönelik geliştirilmiş öğretim materyalleridir (Ders Notlarıdır).
- Modüller, bireylere mesleki yeterlik kazandırmak ve bireysel öğrenmeye rehberlik etmek amacıyla öğrenme materyali olarak hazırlanmış, denenmek ve geliştirilmek üzere Mesleki ve Teknik Eğitim Okul ve Kurumlarında uygulanmaya başlanmıştır.
- Modüller teknolojik gelişmelere paralel olarak, amaçlanan yeterliği kazandırmak koşulu ile eğitim öğretim sırasında geliştirilebilir ve yapılması önerilen değişiklikler Bakanlıkta ilgili birime bildirilir.
- Örgün ve yaygın eğitim kurumları, işletmeler ve kendi kendine mesleki yeterlik kazanmak isteyen bireyler modüllere internet üzerinden ulaşabilirler.
- Basılmış modüller, eğitim kurumlarında öğrencilere ücretsiz olarak dağıtılır.
- Modüller hiçbir şekilde ticari amaçla kullanılamaz ve ücret karşılığında satılamaz.

İÇİNDEKİLER

AÇIKLAMALAR	ii
GİRİŞ	1
ÖĞRENME FAALİYETİ-1	3
1. A/D VE D/A DÖNÜŞÜM	3
1.1. Analog sinyal ve Dijital sinyal	3
1.2. Mikroişlemci	4
1.3. A/D VE D/A Dönüşüm İçin Gerekli Mikroişlemci Seçimi	5
1.3.1. A/D Dönüşümün Rolü	7
1.3.2. D/A Dönüşümün Rolü	15
1.4. Mikroişlemcinin Programlanması	27
1.4.1. DSP' nin Tanımlanması	27
1.4.2. DSP' nin Mimarisi	28
1.4.3. Sayısal İşaret İşleme İçin Gereken Hesaplamalar	30
1.4.4. TMS320C5000 DSP serisi	31
UYGULAMA FAALİYETİ	41
ÖLÇME VE DEĞERLENDİRME	44
ÖĞRENME FAALİYETİ-2	45
2. FIR VE IIR FİLTRE NEDİR?	45
2.1. FIR tip filtre	45
2.1.1. FIR Tip Filtrenin Oluşumu	45
2.2. IIR tip filtre	46
2.2.1. IIR Tip Filtrenin Yapısı	46
2.3. FIR ve IIR Filtrenin Kullanım Alanları Nelerdir	49
2.3.1. Filtre çeşitleri	49
2.3.2. Filtrelerin genlik karakteristikleri	50
2.3.3. FIR ve IIR Filtre Nasıl Gerçekleştirilir	52
2.3.4. Alçak-geçiren filtre, filtre katsayıları ve karakteristikleri	54
2.3.5. Kesim frekansına nasıl karar verilir	61
UYGULAMA FAALİYETİ	69
ÖLÇME VE DEĞERLENDİRME	75
MODÜL DEĞERLENDİRME	76
CEVAP ANAHTARLARI	77
KAYNAKÇA	78

AÇIKLAMALAR

KOD	523EO0382
ALAN	Endüstriyel Otomasyon Teknolojileri
DAL/MESLEK	Ortak alan
MODÜLÜN ADI	Sayısal İşaret İşleme-1
MODÜLÜN TANIMI	Sayısal işaret işleme setini kullanma ve programlama becerisinin kazandırıldığı bir öğrenme meteryalidir.
SÜRE	40/32 Saat
ÖN KOŞUL	Ortak dersleri başarmış olmak
YETERLİK	Sayısal işaret işleme uygulamaları yapmak
MODÜLÜN AMACI	Genel Amaç Sayısal işaret işleme devresini teknik özelliklerine ve programlama tekniklerine uygun olarak kullanabileceksiniz. Amaçlar <ol style="list-style-type: none">1. Mikroişlemci kullanarak D/A ve A/D dönüşümünü doğru bir biçimde yapabileceksiniz.2. FIR ve IIR filtre uygulamasını istenen değerlere uygun olarak yapabileceksiniz.
EĞİTİM ÖĞRETİM ORTAMLARI VE DONANIMLARI	Ortam: Elektronik Laboratuvarı Donanım : Bilgisayar, sayısal işaret işleme deney setleri
ÖLÇME VE DEĞERLENDİRME	Her faaliyetin sonunda ölçme soruları ile öğrenme düzeyinizi ölçeceksiniz. Araştırmalarla, grup çalışmaları ve bireysel çalışmalarla öğretmen rehberliğinde ölçme ve değerlendirmeyi gerçekleştirebileceksiniz.

GİRİŞ

Sevgili Öğrenci,

Sayısal İşaret İşleme 1 modülü ile endüstriyel otomasyon teknolojileri alanında kullanılan mikro işlemcileri programlamaya yönelik bilgi ve teknolojiye ait temel yeterlilikleri kazanacaksınız.

Bu modülü başarılı bir şekilde tamamladığınızda Mikro işlemci kullanarak D/A ve A/D dönüşümünü doğru bir biçimde yapmayı kavrayarak detaylı bilgiye sahip olacaksınız. Ayrıca FIR ve IIR filtre uygulamasını istenen değerlere uygun olarak yaparak kendinizi geliştireceksiniz.

ÖĞRENME FAALİYETİ-1

AMAÇ

Mikroişlemci kullanarak D/A ve A/D dönüşümünü doğru bir biçimde yapabileceksiniz.

ARAŞTIRMA

Ø D/A ve A/D dönüşümü hakkında bir araştırma yaparak rapor haline getiriniz.

1. A/D VE D/A DÖNÜŞÜM

1.1. Analog sinyal ve Dijital sinyal

Analog sistemde; devamlı, yapılan ölçümlerde ve sistemin çalışması esnasında kullanılan fiziksel değerler sürekli değişmektedir, Elektriksel devreler, voltaj vb. gibi.

Dijital sistem ise, rakam ya da karakter formunda görüntülenebilen farklı sinyalleri ifade etmektedir.

Bir sözlüğe başvuracak olursak da, benzer tanımlar göze çarpar.

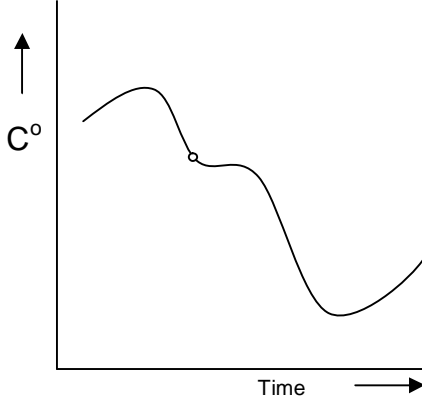
Dijital kelimesinin Latin kökenli olduğu ve Latince’de parmak anlamına geldiği ve buradan İngilizce’ye girerek zamanla “rakam” kelimesi yerine kullanılmaya başladığı görülmektedir. Şöyle bir benzetme yapabiliriz; Analog, devam eden bir süreci ifade ederken; Dijital, 1’ e, 2’ ye, 3’ e şeklinde saymakta ve saymaya devam etmektedir, iki tanım arasındaki fark budur. Eğer Analog ve Dijital sinyalleri şekilsel benzeşimlerle karşılaştırmak istersek bu Tablo 1.1’ deki gibi gösterebiliriz.

	Analog	Dijital
Su miktarı	Kaç litre?	Kaç bardak?
İngilizce	Miktar sayılamaz	Sayılabilen miktar
Zaman gösterimi	İbre ile gösterim	Figür olarak gösterim
Eğrinin eğimi	Eğim ile	Adımla

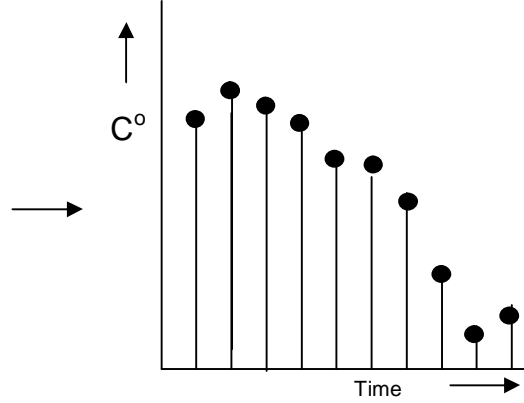
Tablo 1.1: Analog ve dijital

Dünyadaki doğal olaylar incelendiğinde çoğunun analog bir değişme gösterdiğini söyleyebiliyoruz. Örneğin hava değişimini düşünün; hava sıcaklığı sabahleyin birden-bire değişmemesine rağmen insanlar “Bu sabah hava birden soğudu.” derler. Oysaki bir önceki gün ile arada bir gece vardır ve sıcaklık kademeli olarak düşmüştür. Ne zaman ki basınç

değişir o anda kaydedici araç tarafından kayıt altına alınan değer düzleşmiş bir kavis görünümündedir (Şekil 1.1). Bu analog bir değişimdir (devamlı).



Şekil 1.1: Analog kayıt



Şekil 1.2: Dijital kayıt

Fakat, eğer data için yapılan gözlemlerden belli aralıklarla sabitler elde ediliyorsa, örneğin her saat başı alınan sabitler bir grafikte gösteriliyor ise bu grafikteki değişimin Dijital bir değişim olduğunu söyleriz (Şekil 1.2).

Eğer Şekil 1.1 ve Şekil 1.2' yi karşılaştıracak olursa, hangisinde ısının devamlı olarak değiştiği buradaki grafiklerden anlaşılabilir. Buna ek olarak, gözlemlenen grafiklerden, dijital sözüyle ifade edilenin kademeleri daha belirgin bir sinyal olduğu ve Analog adı verilen sinyalle azalma sürecinde aralarındaki farklılıklar gözlenebilir.

Buradan yola çıkarak şu söylenebilir; dijital olarak işlenen analog bir sinyalin incelenmesi mümkündür. Örneğin her insan farklı frekansta bir analog sese sahiptir. Bu ses, ses kayıt cihazlarında dijitale çevrilerek kaydedilir. Kayıtlı sesi dinlemek istediğimiz zaman ise dijital sinyal analog sinyale çevrilir.

1.2. Mikroişlemci

Mikroişlemci; gerek yaptığı işlemlerin mikro saniyeler düzeyinde olması gerekse içindeki elektronik devrelerin ve bölümlerin mikron boyutlarında olması nedeniyle bu adı almıştır.

Mikroişlemci; bir bilgisayar sisteminin en önemli 3 donanımından biridir ve bu 3 donanım arasında adı en çok geçendir; diğerleri, hafıza (RAM-ROM) ve giriş-çıkış (I/O) birimleridir. Mikroişlemci dünyasındaki gelişmelerin yanında diğer donanımların zaman içinde gelişmesi epey yavaş kalır. İnsanlar bilgisayarlarını birbirlerine tarif ederlerken önce mikroişlemcisini söyler “Bende Pentium III 500 var, senin sistem nedir?” gibi.

Bilgisayarlarda bu kadar önemli bir yere sahip olan mikroişlemcilerin tabi ki sadece bir tek adı olması düşünülemez . Mikroişlemcinin CPU (sıpiu diye okunur - Central Processing

Unit), MİB (CPU nun Türkçe karşılığı - Merkezi İşlem Birimi), μ P (mikro processor-mikro prosesor diye okunur) ve genelde işlemci olarak bildiğimiz isimlerini de kullanıyoruz.

Adından da anlaşılacağı gibi mikroişlemci (veya işlemci) matematiksel işlemleri yapabilen bir elektronik yonga (chip) dır. Boyutları çok küçük olmasına rağmen içinde binlerce, yüz binlerce veya milyonlarca elektronik devre elemanı bulunduran mikroişlemci aslında matematiksel işlemleri, elektriğin var olması ya da olmaması temelinden yararlanarak hesaplar.

Matematikçilere soracak olursanız kendi bilim dallarının temelinde aslında sadece toplama işleminin olduğunu söylerler. Mikroişlemci de aslında sadece toplama işlemi yapar. Mikroişlemci için çok kaba olmakla beraber toplama işlemini çok hızlı yapan bir elektronik devredir de diyebiliriz. Sadece toplama işlemini yapması pek çekici görünmese de asıl ününü buradan almaz zaten, mikroişlemciyi mikroişlemci yapan matematiksel işlemleri çok kısa bir zamanda hatasız olarak gerçekleştirebilmesidir. Saniyede milyonlarca işlem yapabilir.

Sonuç olarak mikroişlemci matematiksel, aritmetik ve mantık işlemlerini çok kısa sürelerde yapabilen bir elektronik devredir, bir bilgisayar sisteminin beynidir (kalbi diyenler de var), diyebiliriz. Şayet sizin de bir bilgisayarınız varsa kapağını açıp içindeki mikroişlemciyi görebilirsiniz.

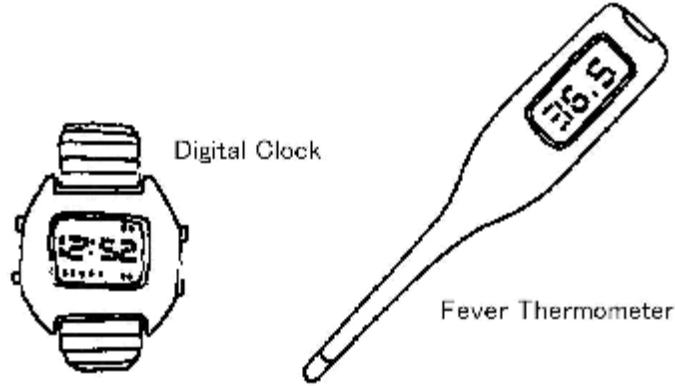
Bir bina yapılırken nasıl çimento, kum ve çakıl kullanılıyorsa mikroişlemciler yapılırken de bazı elektronik devre elemanları kullanılır, **transistör** dediğimiz cihaz ise çoğu kişinin yabancı olmadığı bir devre elemanıdır. Günümüz mikroişlemcileri milyonlarca transistörü bir arada barındırır. Transistörlerle ilgili yazımızı okuyanlar anlayacaklardır ki bir transistör sadece bir olay gerçekleştirir, birkaç tanesi bir araya gelerek bir iş yapar, şayet güzel ve kayda değer işler yapmak istiyorsanız binlercesini veya milyonlarcasını bir araya getirmeniz lazım. Bu arada küçük bir transistör bir nohut tanesi kadardır, milyonlarca transistör çok fazla yer kaplar ama günümüz teknolojisi bu kadar devre elemanını santimetrelere sığdırmayı başarmıştır.

1.3. A/D VE D/A Dönüşüm İçin Gerekli Mikroişlemci Seçimi

Şimdi, dijital sözcüğünün anlamına ve kullanım alanlarına bir göz atalım.

Dijital denilince ilk dijital saat gelir aklımıza, bunun yanında dijital ateş ölçme aletleri de artık bugün marketlerde yerini alıyor. Ancak, bu aletler, eğer biz her değişimin adımlarının neye bağlı olduğunu vermezsek, bir insanın ateşinin ya da her geçen saniyenin nasıl anlaşılacağını, nasıl değiştiğini gösteremez.

Bu dijital dünyada; analog değişimler hem düzgün hem de çok özgün bir şekilde gerçekleşir. Bununla birlikte, bu aletler sadece bir saniyenin ya da bir derecenin ayrımına varabilir, çünkü göstergenin üzerinde ifade edilebilecek basamak sayısı oldukça azdır (Şekil 1.3).

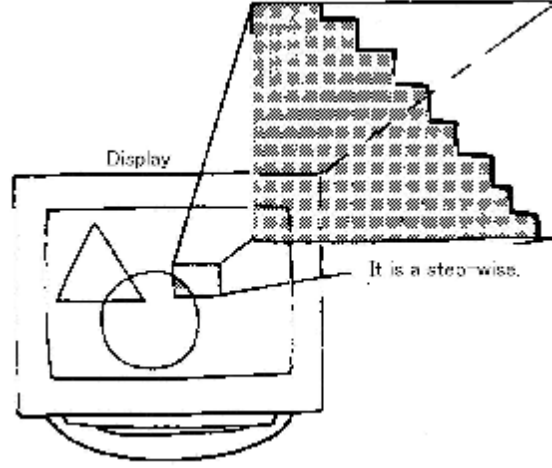


Şekil 1.3: Dijital göstergeler

Çevrenize bir bakın ve dijital bir değişim bulmayı arzulayın, siz de göreceksiniz ki bunu bulmak oldukça güç, baktığınız her yerde analog değişimler göreceksiniz. Eğer ki **Dijital İşaret İşleme** direk olarak işin içine girmiş ve her şeyi çözümlemeye çalışmış olsaydı, her yerde analog değişimlerin yaşandığı bir dünyada bu ne kadar güç ve faydasız olurdu, öyle değil mi?

Peki, niçin dijital İşaret işleme bu kadar hızla yükseldi ve nasıl bu kadar popüler olabildi? Sebepsiz! Sebebi yok! Tek sebep analog işaretlerin dijital olarak işlenebileceğinin keşfedilmiş olması. Dünya üzerinde her dakika analog ve dijital sinyaller karşılaştırılır, bunun nedeni her ikisi arasında oluşabilecek farklılıkların, azalmaların, hata payının önemszenmesidir. Örneğin, biz daima suyun miktarını analog olarak CC vb. ile ölçeriz, ölçüm kapları ile kaç kap geldiğine bakarız. Bunun yanı sıra, sudaki moleküllerin miktarı da gösterilebilir ve suyun miktarı çubuklarla ifade edildiğinde biz bunları sayarak sonucu elde edebilir ya da okuyabiliriz. Dahası, bunu bir çubuktan diğerine yavaşça değişim gösteren bir eğri ile göstermek de mümkündür.

Eğri ekranda plotterda vb. çizildiğinde büyütülebilir de, böylece anlamak daha da kolay olacaktır (Şekil 1.4).



Şekil 1.4: Eğrinin ekranda büyütülmüş görüntüsü

Yukarıda anlatıldığı üzere, işlemin esasında analog ve dijital farklı şeylerdir, esasında bir bütünün farklı parçaları gibidir, buradan önemli olan noktaya ulaşmak mümkündür.

1.3.1. A/D Dönüşümün Rolü

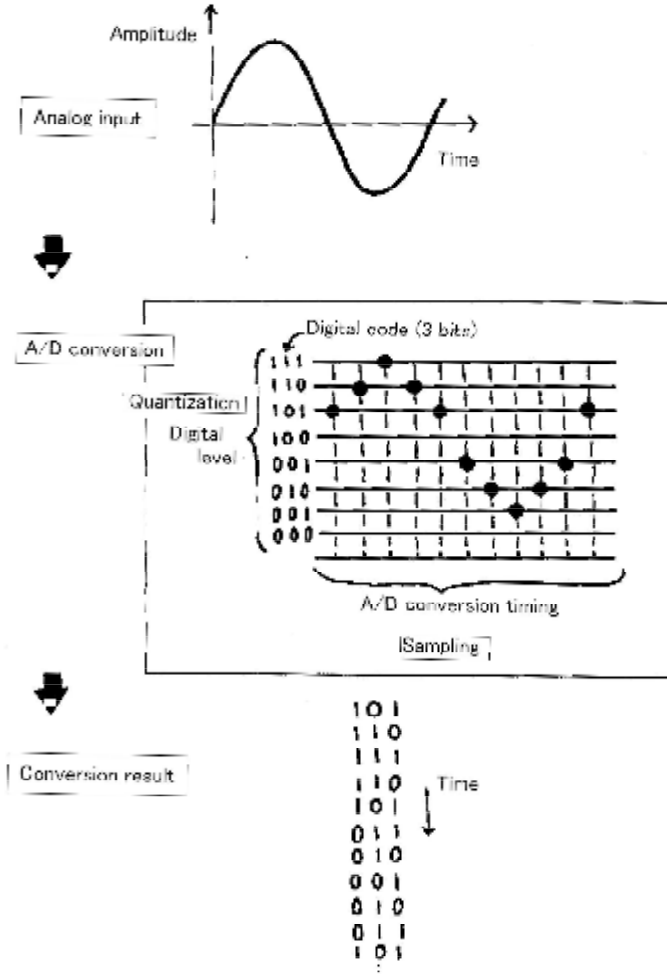
1.3.1.1. A/D dönüşüm

Ø Analog Bilginin Girilme Yöntemi

Bunun anlaşılabilmesi için ilk önce mikroişlemcinin içerisinde dijital sinyalin nasıl bir işleme tabi tutulduğunun anlatılması gerekir. Diğer yandan, şu da bir gerçek ki çevremiz devamlı ve düzgün değişen analog bilgiyle kuşatılmış durumda Bu nedenle, analog bilgi mikroişlemcinin çevremiz ile uyum içinde çalışması sağlanarak ele alınabilir.

İlk olarak, analog bilginin nerelerde değişim gösterdiğini tespit edebilmek için sinyal, eşit aralıklara bölünür (çözümленir). İlâveten, Analog Bilgi Giriş Miktarı hangi bölünmüş aralıkta düzenli aralıklarla değişmektedir, bu dikkatle gözden geçirilir.

Dijital bilginin Binary Rakamlarla karşılaştırmalı olarak ifadesi mümkündür (Şekil 1.5).



Şekil 1.5: A/D dönüşüm işlem

Örneğin, 0-5V aralığındaki analog voltaj değeri dijital formata sekiz bit binary sayılar kullanılarak dönüştürülebilir. Dijital seviyenin kalitesi ya da hassasiyeti ise $5/(2^8-1)$ V seviyesindedir.

Analog bilgiyi dijital bilgiye dönüştürmek için kullanılan bu ve benzeri yöntemlerin tümüne **A/D dönüşüm** adı verilir.

Bu A/D dönüşümde, iki eksen dijitalize edilmektedir, bunlar zaman eksen (yatay eksen) ve genlik eksenidir (dikey eksen) (Şekil 1.5). A/D dönüşüm genelde düzenli zaman aralıkları ile çalıştırılır. Bu, zaman tabanlı dijitalleştirme (Time Based Digitalization) ve buna **örnekleme** adı verilir.

Şu halde, sabit süreli zaman aralıklarını gösteren frekanslara da **örnekleme frekansı** (Sampling Frequency) denir.

Analog sinyal girişinin genlik yönünde uygulanan dijital derecelendirme işlemine **dijitalleştirme**(Digitalization) ve **ölçeklendirme**(Quantization) adı verilir.

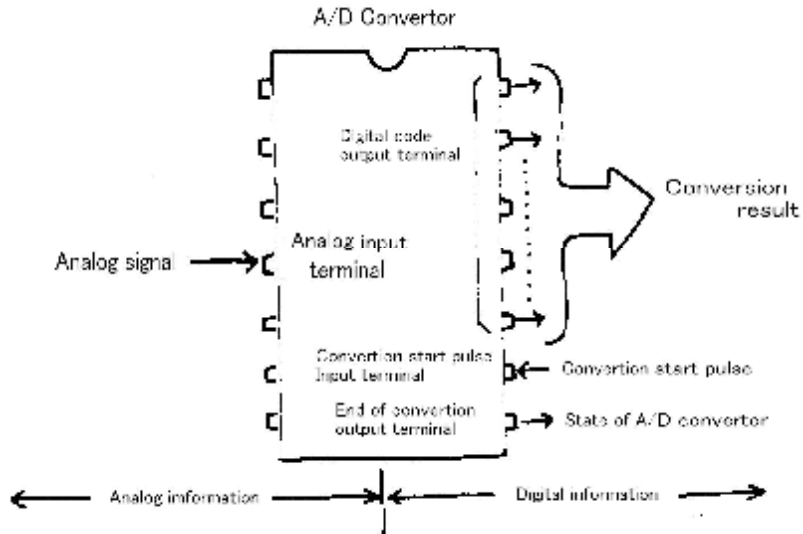
Uygulamada elde edilen ölçeğin her dijital derecesine karşılık gelen dijital kod, Analog'dan Dijital'e dönüşümün elde edilen karşılığıdır.

Şekilde gösterilen Analog giriş sinyali dalgalarının genliği zamanla değişmektedir. A/D Dönüşüm işlemi sırasında yapılan iş sadece düzenli olarak belli aralıklarla yani belli örnekleme aralıklarında bu dalgaların genlik değerlerinin alınmasıdır. Bu durumda şuna dikkatinizi çekmek isterim ki; bilgi kaybı kaçınılmazdır.

Ø A/D Dönüşüm Fonksiyonu

Burada Analog Bilgiyi Dijital Bilgiye çeviren bir A/D dönüştürücü (Converter) gösterilmektedir. A/D dönüşüm işleminin bazı yöntemleri vardır.

Temel işlevi anlayabilmek; Şekil 1.6'da gösterilen dört çeşit işaretin çalışmasını bilmek ile mümkün olacaktır.



Şekil 1.6: A/D dönüşümün temel işaret yönü

Ø Analog Giriş Terminali

Dijital bilgiye dönüşmesi istenen Analog Bilgi girişinin uygulandığı noktadır. Girişe uygulanabilecek voltaj aralığının bir sınırı vardır. Zaten, giriş voltajının aralığı girişe bağlanacak çok basit devrelerle değiştirilebilir de. Eğer dönüştürülmesi istenen Analog bilginin voltaj aralığı sınır değerlerimizden çok ama çok farklı ise, bu durumda voltaj aralığımızın amplifikatörlerle ve dirençlerle uygulanacak kısmi basınçlar vasıtası ile adaptasyonu (uyumlu hale getirilmesi) gerekecektir.

1.3.1.2. Bit Sayısı ve Çözünürlüğü

A/D ya da D/A dönüştürücümüzün çözünürlüğü Dijital bilginin sahip olduğu N kadarlık bit sayısına bağlıdır ve tüm skala (ölçek) değerine (Full Scale Value) 2^N e bölünerek ulaşılır. Tablo 1.2’de bit ve çözünürlük (N) rakamlarının orantılı ilişkileri görülmektedir.

<u>N</u> <u>Bit Sayısı</u>	<u>2^N</u> <u>Miktarı</u>	1 LSB <u>Doğruluk Değeri</u>
4	16	6.25
8	256	0.3906
12	4096	0.0244
16	65536	0.001526

Tablo 1.2: Bit ve çözünürlüğün N rakamları

Ø Dijital Çıkış Terminali

Dönüşümün sonucunda elde edilen dijital işaret, çıkış işareti yolunda kümelenir. Burada çözünürlüğe bağlı olarak 8, 10, 12 ve 16 bit vb. şeklinde yerleşir. Bu, asıl amacımız olan devamlı ve sürekli bilgileri bir bite dönüştürüp çıkışa aktarabilmek için uyguladığımız bir yöntemdir.

Ø Dönüşüm Başlatma İşaret Terminali

Ne zaman ki bu genlik işareti gelir, bu işaret verilir, dönüşümü gerçekleştirilecek olan Analog bilgi Analog Giriş Terminalinden alınmaya başlar.

Ø Dönüşüm Sonu İşaret Terminali

Bu işaret A/D dönüştürücünün içindeki dönüşüm işleminin tamamlanıp çıkışa yollanmaya hazır olduğunu belirtir. Buradan da anlaşıldığı gibi bu adımın gerçekleşmesinin ardından A/D dönüşüm bitmiştir. Oluşan dönüşüm dizisi, RAM’in o anki erişim değerine, yüksek hızlı olmasına bağlı olarak değişebilir de. Sonuç olarak; varolan, elde edilen dönüşüm ve işaret değerinin içinde kaybolan hatta silinen değerlerde olacaktır.

1.3.1.3. Tipik Şifreleme Yöntemi

Şifreleme; sinyalin örnekleme ve miktarlandırma yapılarak dijital kodlarla ifade edilmesi yani dijital formata döndürülmesidir. Aşağıdaki tablo tipik şifreleme yöntemlerinden birisini göstermektedir.

Voltaj değeri kullanılıyor ve sadece artı yönlü değişimlerde (unipolar) binary düz karşılık (straight binary) kullanılıyor. Sıranın içerisinde eksi ve artı yönlü değişimlerin olduğu (bipolar) durumda Offset binary(düzeltilmiş binary karşılık), offset binary ya da twos complement (ikiliye tümler) kullanılabilir. Dijital kod A/D dönüşüm ya da D/A dönüşümden alınabilir. Bu yöntemi her iki duruma da uyarlamak mümkündür.

Giriş V [V]	Binary düz karşılık	Giriş V [V]	Düzeltilmiş binary	İkiliye tümleyen
+10.000	-	+10.000	-	-
+9.961	11111111	+9.922	11111111	01111111
+9.922	11111110	+9.844	11111110	01111110
+5.039	10000001	+0.078	10000001	00000001
+5.000	10000000	+0.000	10000000	00000000
+4.961	01111111	-0.078	01111111	11111111
+0.039	00000001	-9.922	00000001	10000001
+0.000	00000000	-10.000	00000000	10000000

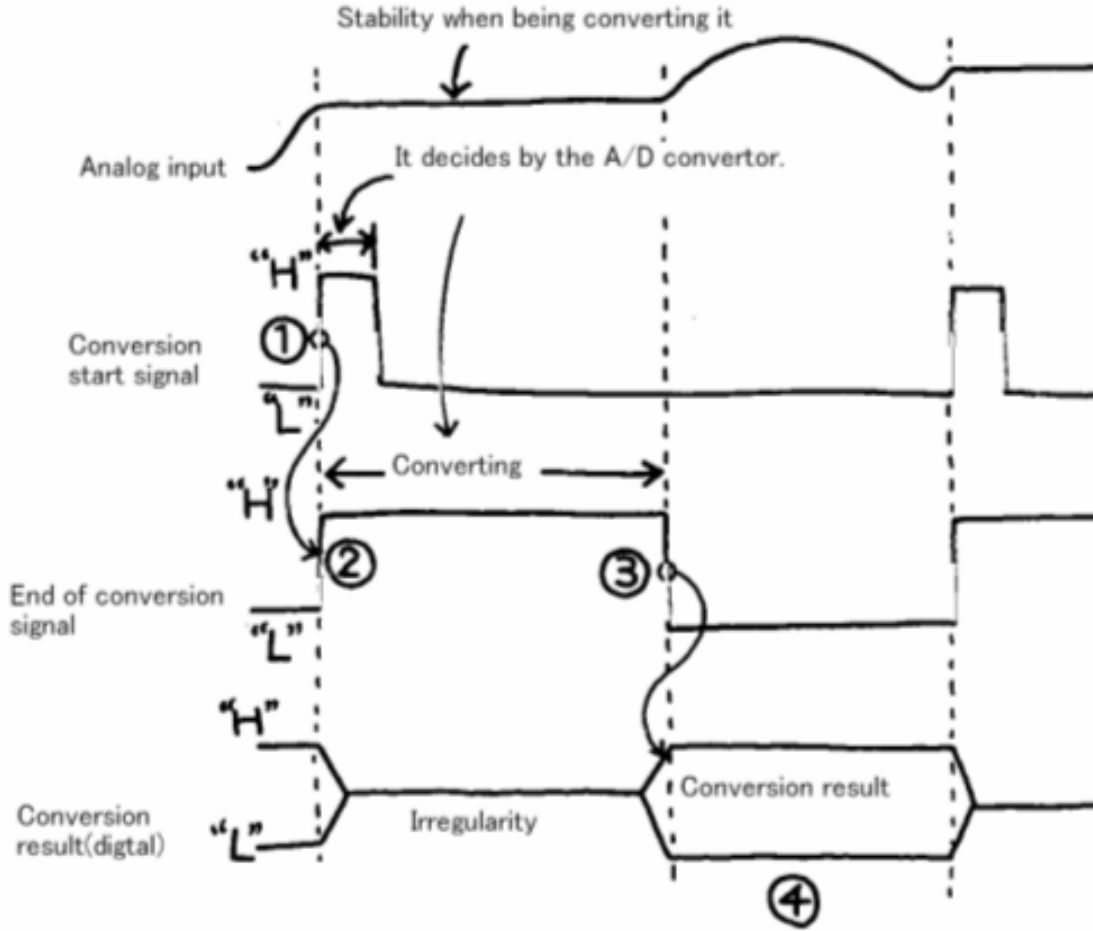
(1) Unipolar giriş
(0 a +10V)

(2) Bipolar giriş
(-10V a +10V)

Tablo 1.3: Bir işaret yöntemi

1.3.1.4. A/D Dönüşüm Nasıl Kullanılır

Şekil 1.7’de A/D dönüşüm süreci (başlangıçtan bitişe tüm adımlar) gösterilmektedir, Analog giriş terminaline gelmekte olan A/D dönüşüm giriş voltajını görüyorsunuz. Buradan sonraki A/D dönüşüm işlem basamakları aşağıdaki sırayla şekilde yer almaktadır.



Şekil 1.7: A/D dönüşüm işlemi

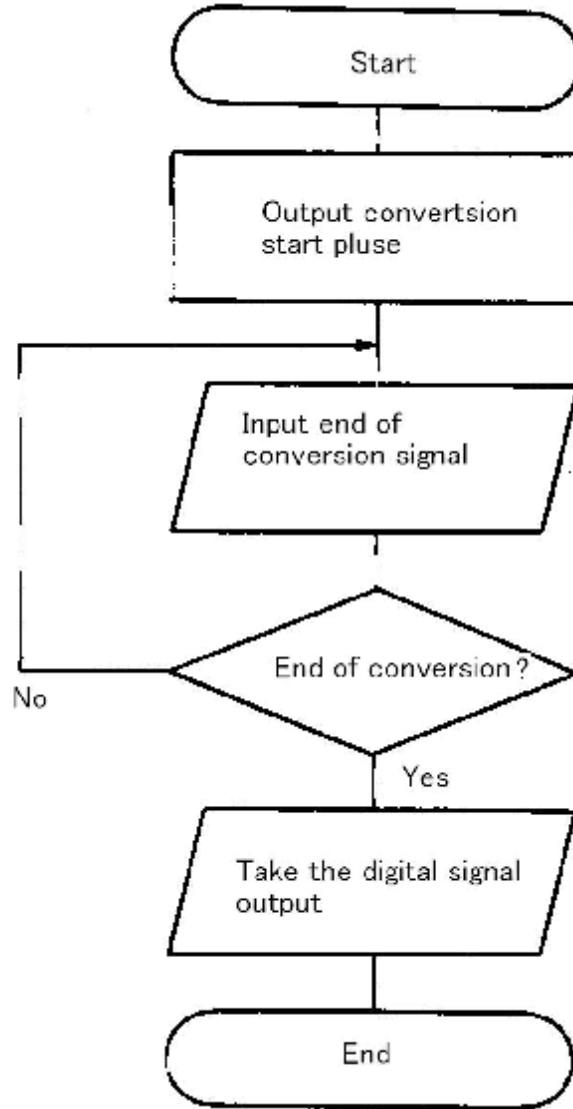
- İlk olarak, dışarıdan bir giriş işaretinin verilmesi ile dönüşüm başlar.
- , A/D dönüşüm, dönüşüm başlama işareti (pulse) alındığı anda başlar.

Şekilde gösterildiği üzere dönüşüm bitiş işareti alındığı anda sinyal içinde tespit edilen "L" ve "H" bitleri çıkışa aynı anda gönderilir. Dönüşüm bitiş işareti "H" dönüşüme devam ederken alınır. Dönüşüm için gerekli zamana **dönüşüm zamanı** denir. Bu A/D dönüşümün performansını etkileyen değişkenlerden birisidir. Analog giriş bu dönüşüm zamanı içerisinde sabit kalabilir.

f Dönüşüm sona erdiğinde kesinleşmiş Dijital sinyali, dönüşüm sonucu olarak, "H" dan "L" ye olmak üzere gösterir.

” “H” dan “L” ye bölümlendirilmiş sinyalin doğrulaması yapılır ve Dijital işaret çıktı olarak dışarıya aktarılır.

A/D dönüşüm işlemi yaklaşık olarak yukarıda bahsedildiği şekilde kontrol edilir ve Dijital çıktıyı veren programın blok diyagramı ise Şekil 1.8’de gösterilmektedir.

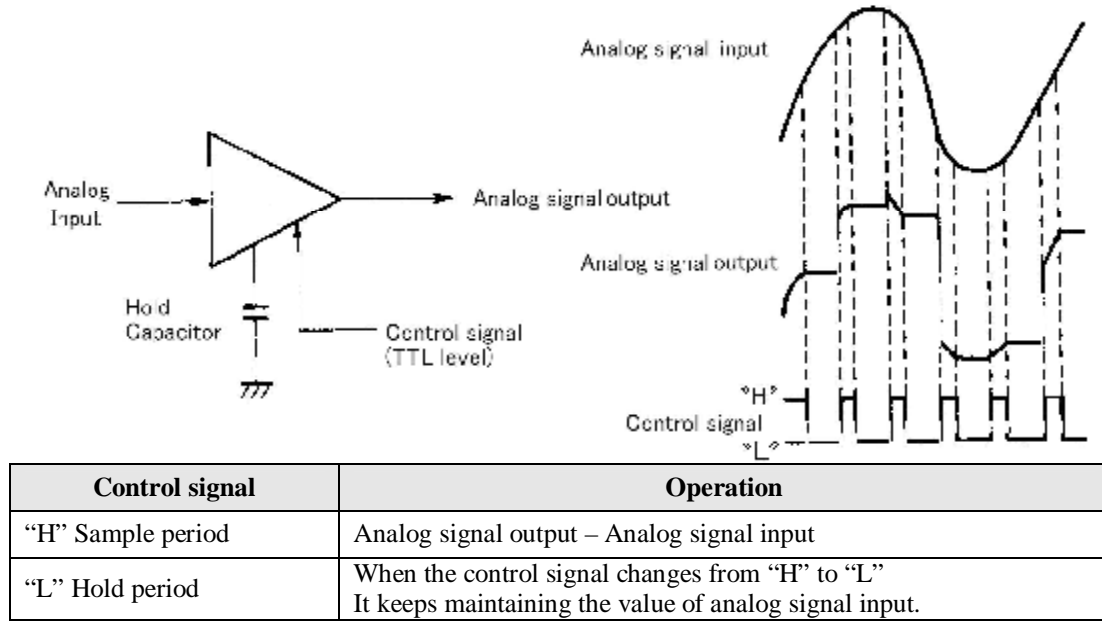


Şekil 1.8: A/D dönüşüm programı örnek blok diyagram modeli

1.3.1.5. A/D Dönüşüm Dalga Şeklindeki İçsel Kesmeler

Girişe uygulanan analog sinyalin genliğinde hızlı değişiklikler gözleniyorsa ve A/D dönüştürücü çalışır durumdaysa, kesin doğru bir A/D dönüşümden söz edilemez. Sonra,

periyodun içerisinde A/D çevrim sırasında analog girişin değişmediği anlarda alınan kesin değerler korunur. Bu şekilde yapılarak A/D dönüşüm gerçekleştirilebilir. Bu örnek için kullanılan devre bir **tutucu devre**dir (hold circuit). **Örnek yakalama devresi** (sampling hold circuit) analog giriş ile analog sinyal giriş terminali arasına yerleştirilir ve A/D dönüştürücüye yollanıp kullanılır.

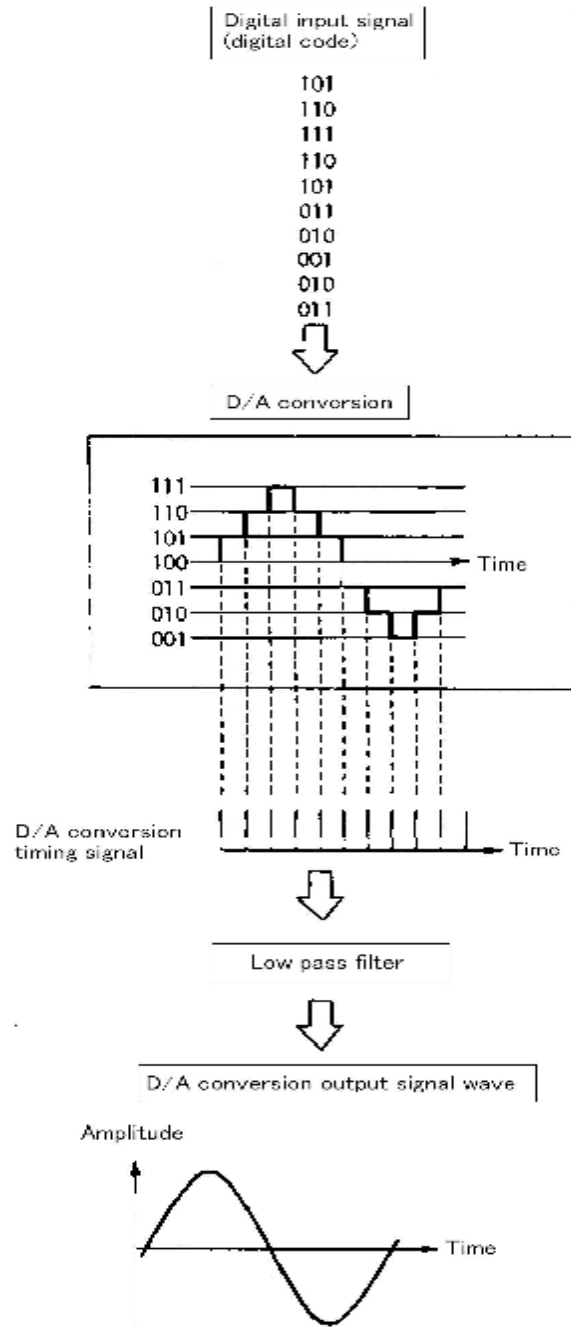


Şekil 1.9: Örnek yakalama devresinin çalışması

Şekil 1.9’da bir yakalama devresinin çalışması örnek olarak verilmiştir. Kontrol işareti “H” olduğunda, giriş voltajındaki değişiklik çıkışa aktarılır (**örnekleme işlemi**). Bu yakalama işlemi kontrol sinyalinin her yükselen ucunda tekrarlanarak örnekleme alınır ve kontrol işareti “L” durumunda iken de voltaj değeri korunur (**yakalama işlemi**).

1.3.2. D/A Dönüşümün Rolü

1.3.2.1. D/A dönüştürücü

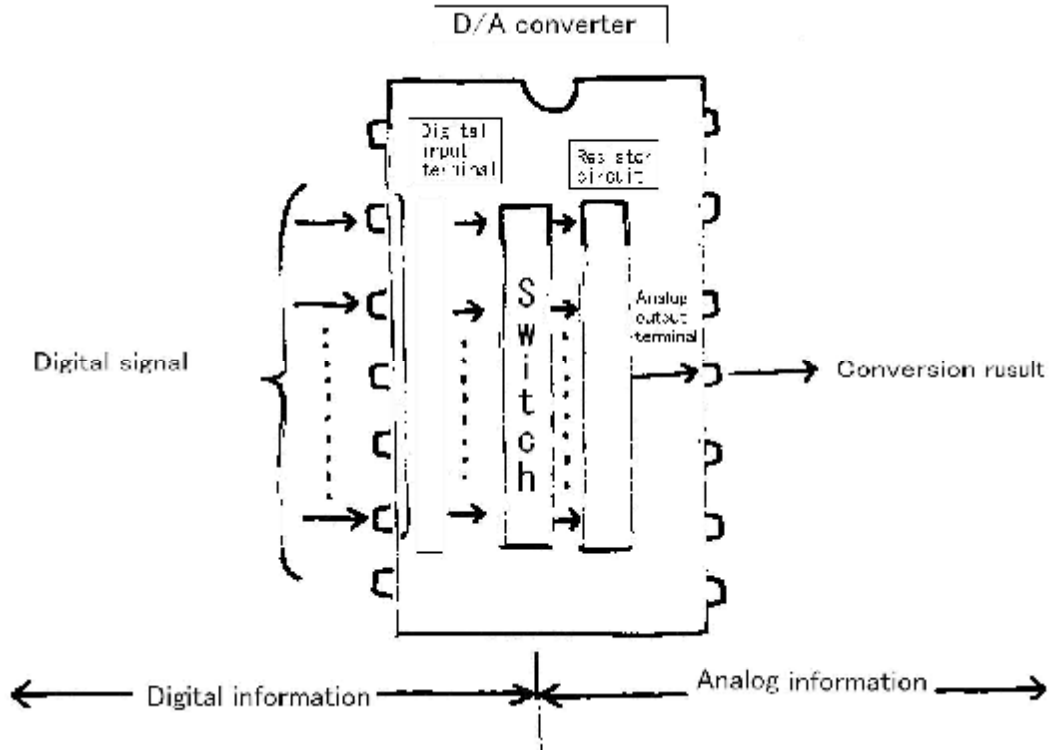


Şekil 110: Dijitalden analoga dönüşüm işlemi

Dijital işaret işleme yolu ile ileri düzeyde matematiksel işlemler yapabilseniz bile, eğer sonuçlar yaşadığımız yer olan gerçek dünyaya (analog) indirgenemiyorsa, anlamlı bir sonuç elde edildi denilemez. Bu nedenle, dijital işaret işlemede dijital bilginin analog bilgiye dönüştürülmesi için çeşitli yollara ve yöntemlere başvurulabilir. D/A dönüşüm temel olarak A/D dönüşümün tersidir (Şekil 1.10).

A/D dönüşüm vb. yollarla elde edilen dijital kodlardan D/A çevrim sonucu elde edilecek olan analog değerler ilk değerlerle orantılı olmalıdır. Dahası, D/A dönüşümde kullanılan zaman aralıklarına bağlı olarak, elde edilen analog dalga frekansı ilk kullanılmış olan analog giriş sinyalinin frekansı ile aynı olabilecektir. Bununla birlikte, edindiğimiz analog sinyal, D/A dönüşüm devresinden detaylı bir şekilde adımlama yöntemiyle elde edilmiş ve içerisinde çok miktarda gereksiz harmonik içeriklerde taşıyan bir sinyaldir. Alçak geçiren filtreden geçerek yumuşakça engebesiz bir hale dönüşmektedir. Bu D/A dönüşüm işleminde kullanılan alçak geçiren filtre dönüşüm zaman frekansının $\frac{1}{2}$ sini veya daha da azını geçirecek şekilde ayarlanabilir.

Aşağıda dijital bilgiyi analog bilgiye çeviren bir D/A dönüştürücü devresi görülmektedir.



Şekil 1.11: D/A dönüştürücünün sinyal iletim devresi

D/A dönüştürücüdeki çevirme işlemi anlık yapıldığından, çevrim işlemi sonunda A/D dönüşümdeki bitiş sinyali gibi bir kontrol sinyaline ihtiyaç yoktur. Bu nedenle, bir sinyal iletim devresi basitçe Şekil 1.11 deki gibi gösterilebilir.

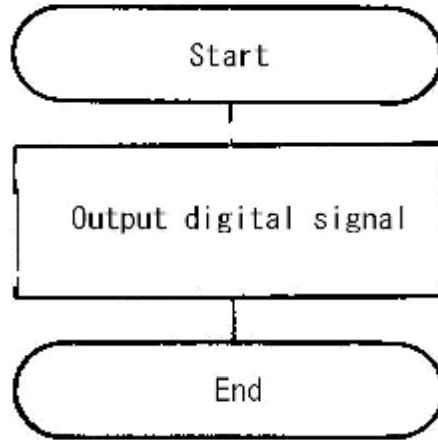
Sinyal iletim devresi; dönüştürülmesi istenen dijital bilginin sinyal yolu demetidir. Ayırma gücüne bağlı olarak 8, 10, 12, ve 16 bit vb. olabilir. Amacımıza dönük olarak, buradaki giriş metodu bu bitleri birer birer seri olarak ele almayı benimsemektedir.

Dönüşüm sonucunda çıktı olarak elde edilen analog sinyallerin izlediği sinyal yoludur. İki çeşit çıkıştan söz edilebilir; birincisi voltaj güç çıkışı, ikincisi ise elektrik akım güç çıkışıdır. Her D/A dönüştürücü bu çıkış sırasına kendisi karar verir.

1.3.2.2. D/A Dönüşümü Nasıl Kullanılır

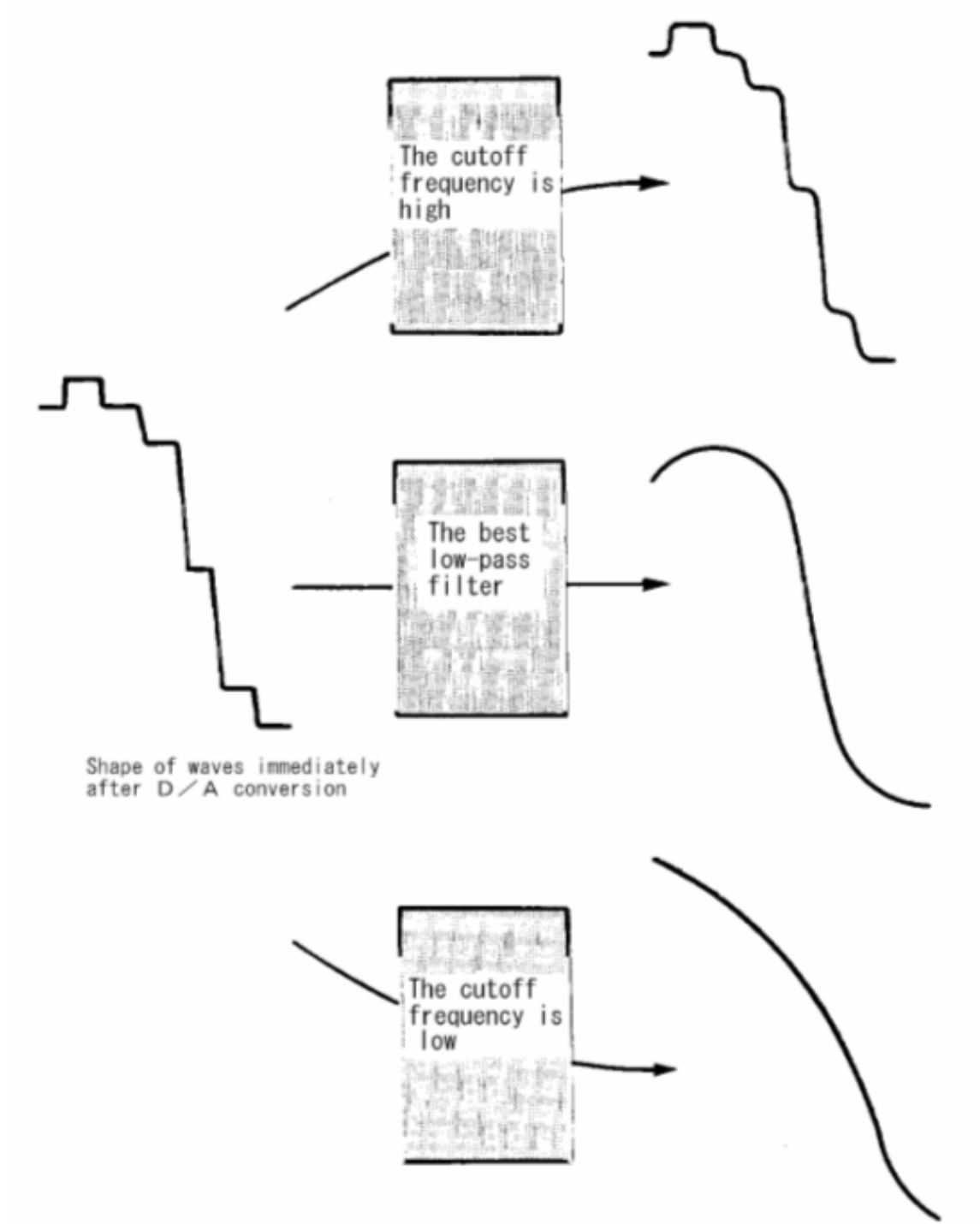
Dijital bilgi D/A dönüştürücüye verildiğinde, bu dijital bilgiye karşılık gelen analog değer güç çıkışından neredeyse hemen elde edilir. Bu nedenle, A/D dönüşüm işlemi sırasında geçen işlem zamanına karşılık gelen değerle bunun mukayesesi gereksizdir. D/A dönüşümün dijital giriş işareti değişmediği sürece, dönüşüm sonucu dijital kodu verilen analog bilgi çıkış olarak korunur.

Bu nedenle, D/A dönüşümü kontrol eden ve çıkış olarak analog işaret veren program oldukça basittir ve Şekil 1.12' deki akış diyagramı ile gösterilebilir.



Şekil 1.12: D /A dönüşüm programının temel fikri

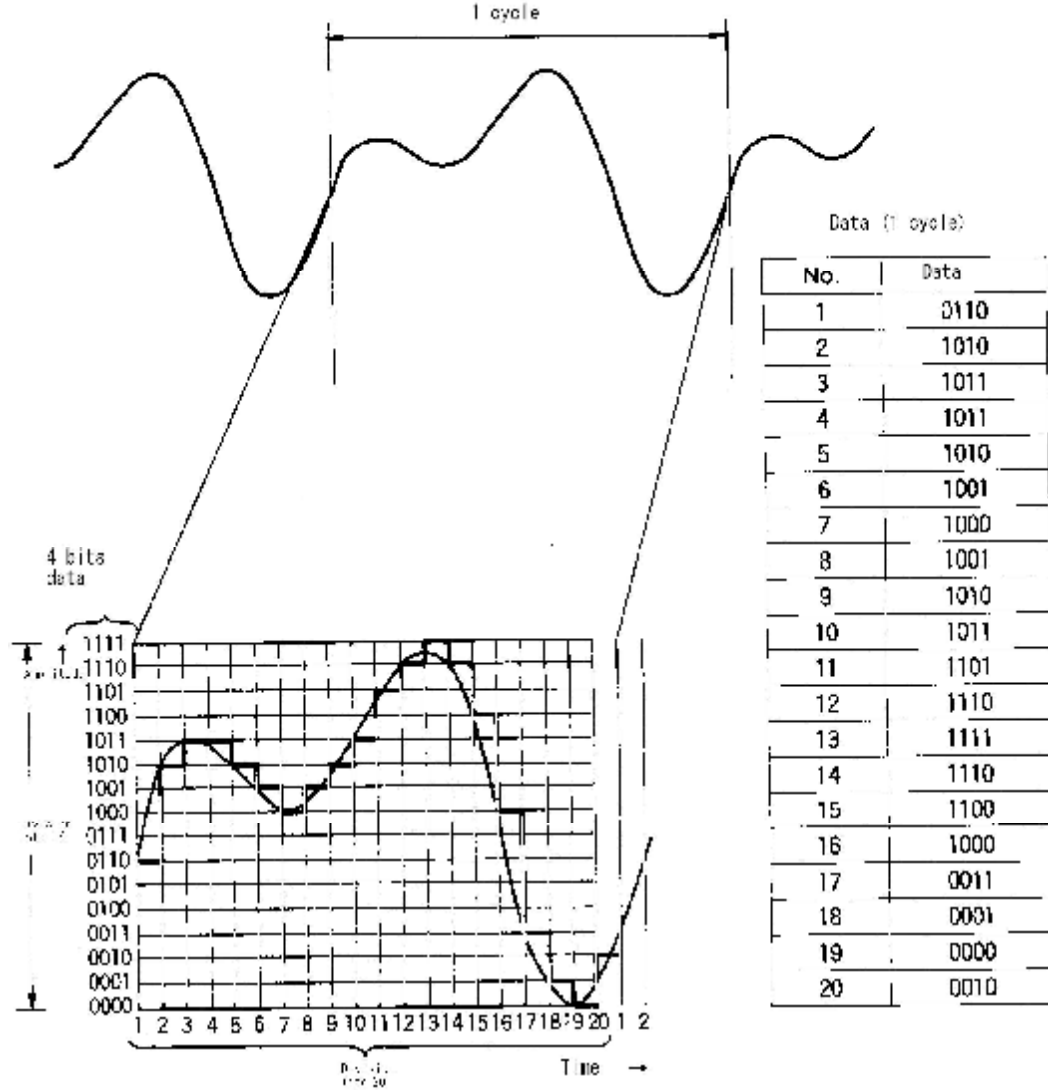
D/A dönüşümdeki güç dalga formunun dönüşümü adımlamalı bir işlem olduğu için elde edilen analog sinyalin dalga formunda bazı bozukluklar olacaktır, bu bozuklukların giderilmesi için kesme frekansı dönüşüm zaman aralığının iki ya daha fazla katı büyüklüğünde olan alçak-geçiren filtre uygulanır (Şekil 1.13.).



Şekil 1.13: D/A dönüŖüm güç çıkıŖı kesme frekansı ile alçak geiren filtre arasındaki iliŖki

1.3.2.3. D/A Dönüşüm ile Tekrarlanan Dalga Şekillerinin Üretilmesi

D /A dönüşüm sonucunda bir çevrim süresi kadar dalga şekli çıkış tablosuna tekrar tekrar aktarılır ve yine bu bilgi defalarca çıkışa yollar. Bir çevrimin veri tablosunun oluşturulabilmesi için, dalga şeklini tekrarlar ve bir çevrim dalga şeklinden kesilerek alınır. Ayrıca; D/A dönüşümün çözme gücünü kullanarak ve bir çevrimi zaman tabanlı veri numaralandırması yoluyla genlik yönünde bölümlendirme, parçalama yapmak da mümkündür (Şekil 1.14.).



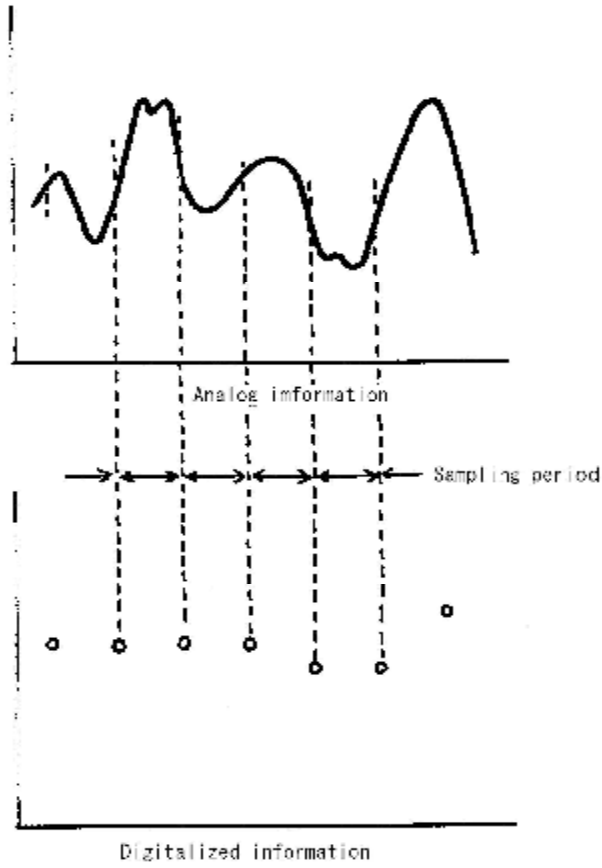
Şekil 1.14: Bir çevrimim tekrarlanması ile dalgasal bilgi nasıl elde edilir (4 bit veri için)

Tekrarlama işlemi devam ettikçe düzenli zaman aralıkları ile elde edilmiş bir veri tablosu yolu ile dalga şekilleri oluşmaktadır. Düzgün dalga şekillerinden oluşmuş bir güç dalga formu yakalamanın yolu ise daha önce de söz ettiğimiz alçak geçiren filtreden geçmektedir.

1.3.2.4. Örnekleme Teoremi

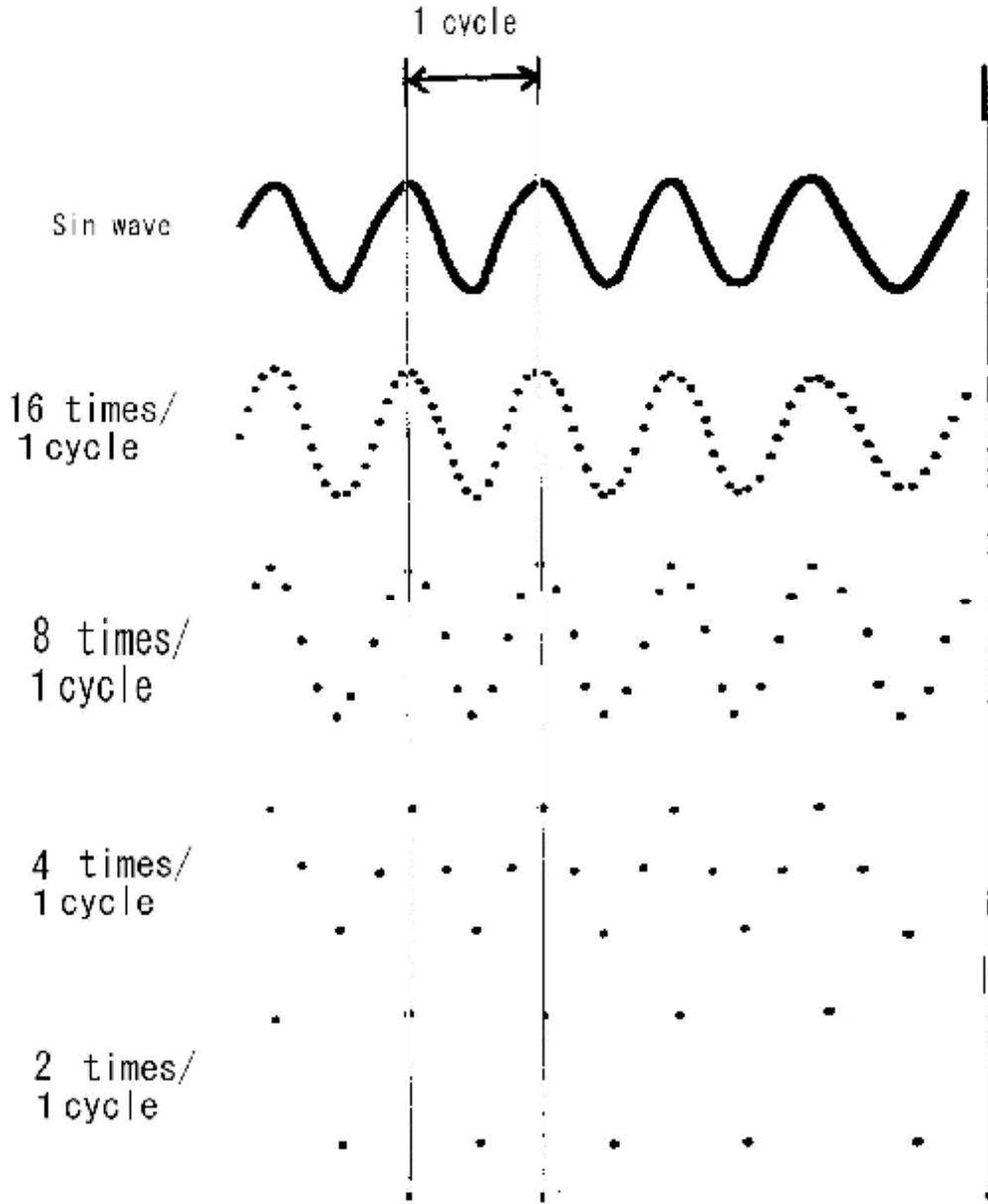
Daha öncede sürekli değişen analog bilginin dijital olarak elde edilmesi anlatılırken bu konunun üzerinde durulmuştu. İşlem hemen hemen aynıdır. Bununla birlikte, A/D dönüşüm işleminin kıstas alınmasıyla A/D dönüşüm işleminden de bir beklentinin oluştuğunu söylemek sanırım yanlış olmaz.

Örneğin, Şekil 1.15'e bakarsak, bu durumda, analog bilginin düzgün bir şekilde dijital bilgiye çevrilmesinin imkânsız olduğunu görürüz. Bununla birlikte, eğer örnekleme zaman aralığı daralırsa elde dijital bilginin doğru ya da doğruya yakın olarak elde edilmesinin yolu açılmış olacaktır (Şekil 1.16).



Şekil 1.15: A/D dönüşümün zamanlaması önemlidir.

Sonuçta, dijital bilginin alınması esnasında ne büyüklükte aralık kullanılıyorsa elde edilen analog bilgi o oranda aslına yaklaşacaktır. Nyquist (Amerika Bell Laboratuvarları 1930 civarı, elektrik mühendisi)'nin matematiksel olarak açıkça gösterdiği bir teorem kullanılmaktadır (Örnekleme teoremi). Ne zaman ki, A/D dönüştürücü ve D/A dönüştürücü bilgisayara bağlansa ve işaretlerin sayısal olarak işlenmesi sağlansa bu teorem kullanılmaktadır. Bu teorem bu alanda bilinen ve kullanılabilen en yaygın araçtır.

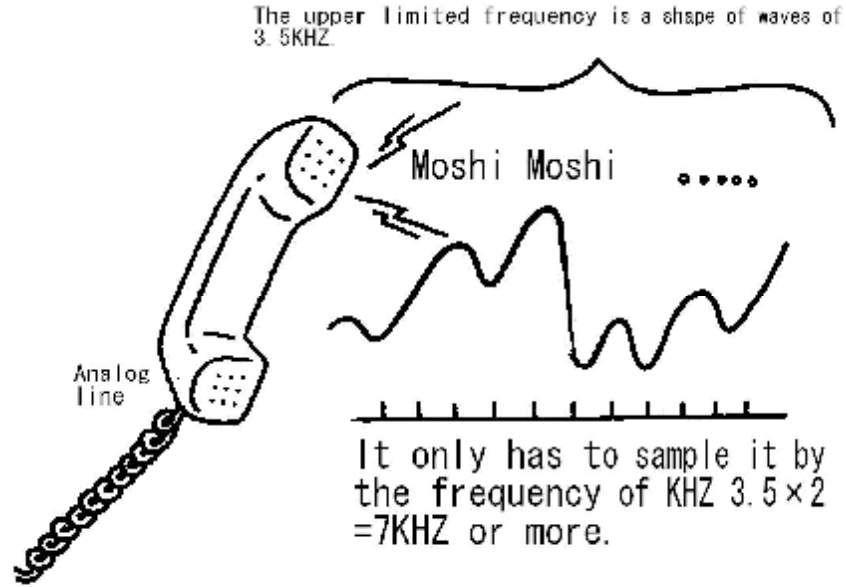


Şekil 1.16: A/D dönüşümün frekans ve dalga şekli

Örnekleme Teoremi: Sürekli değişen bir sinyalin içinden doğru bir örnekleme verisi elde edebilmek için, örnekleme frekansının değerinin, sürekli işaretin frekans üst sınır değerinin iki ya da daha fazla katı bir değer olarak alınmasında fayda vardır.

Bir saniye için elde edilecek A/D dönüşüm değerinin örnekleme frekansını (f_s) değişen işaretin frekans bileşeninin (f) üst sınırının iki katı olarak ayarlanır ($f_s \geq 2f$). Eğer bu teoreme uygun olarak hareket edilmezse, elde edilen işaretimizde gerçek işarete olmayan birtakım bozulmalar (gürültü) oluşacaktır. Bunlara takma gürültü (Alias noise) denir. Alçak

geçiren filtre kullanılarak ki buna Anti-Alias-Filter yani Takıntı-Kesen-Filtre de denilir bu durum önlenabilir. Bu A/D dönüşümü yüksek gereksiz kısımlarını çıkarıp yapacaktır (Örnekleme frekansının 1/2 si aşılma kaydı ile); frekans bileşeni bu alçak geçiren filtre ile taşınır ve böylelikle dijital bilgi gerçekleşmiş yani gerçeğe daha yakın bir bilgi elde edilmiş olur. Örneğin, 3.5 kHz'lik ya da daha düşük bir sinyalin analog olarak gelirken dijitala dönüştürülebilmesi, ancak 7 kHz yani iki katı (saniyede 7,000 kez) örnekleme tabii tutulmasıyla olur. Bu üst sınır frekansdır ve sesin kalitesi için ve dijitala çevrildiğinde akıcı bir veri elde edilebilmesi için gereklidir (Şekil 1.17.).



Şekil 1.17: Sinyalin sahip olduğu üst sınır frekansının iki katı olacak şekilde örnekleme alınır.

Bundan başka, 20 kHz ve üzerinde gelen sinyallerin iki ya da daha fazla bir değerle çarpılmasıyla ancak insanoğlu bu sesleri duyabilir ve A/D dönüştürücü tarafından elde edilen dijital işaret yaklaşık 44 kHz civarında CD'ye kaydedilir.

1.3.2.5. Gerçek A/D ve D /A Dönüşüm Devresinin Bileşenleri

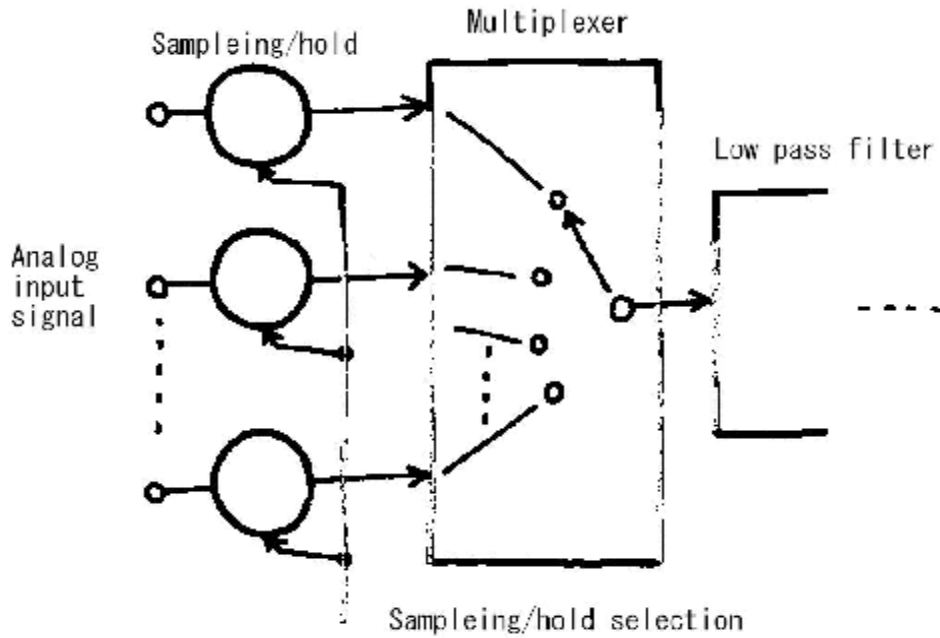
Bu bölümde Dijital İşaret İşleme’ de kullanılan A/D dönüşüm devresi ve D /A dönüşüm devresinin bileşenleri anlatılmaktadır.

Genelde A/D dönüşüm devresi Şekil 1.18’de gösterilen bileşenlerden meydana gelmektedir. İlk olarak, bir çoklayıcı (multiplexer) göze çarpmaktadır ki bu gelen bir ya da daha fazla analog işaret bilgisinden dönüşüme girecek olanın seçimini sağlamaktadır.

Bu bir entegredir ve dijital işaretin birden çok analog giriş biti içerisinde giriş seçimi ile analog giriş sinyalinin belirlenmesini sağlar.

Onu takip eden Alçak-Geçiren-Filtre ise bir Takıntı-Kesen-Filtre’dir ve kesme frekansı değişebilir. Yükselteç (Amplifier) zayıf analog sinyalleri A/D dönüştürücüde işlenebilecek seviyeye kadar yükseltir. Kazanç bu dijital işaretin birkaç bitinde verilerek ayarlanabilir.

Bu sözünü ettiğimiz üç elemanın bağlantı sırası kararlaştırılmamıştır. Her ne kadar, eğer alçak-geçiren-filtre ve yükselteç çoklayıcının önüne konulunca, her biri ayrı birer analog işaret oluşturacaktır ve bu durum bir maliyetle ilgili olarak problem oluşturmaktadır.



Şekil 1.18: Gerçek A/D dönüşüm devresinin bileşenleri

Örnek yakalama devresi analog işareti yakalar ve A/D dönüşüm de sabit değerleri dönüşümünü sağlar. Bu A/D dönüşümün kontrol özelliğinin sağlanması için gereklidir.

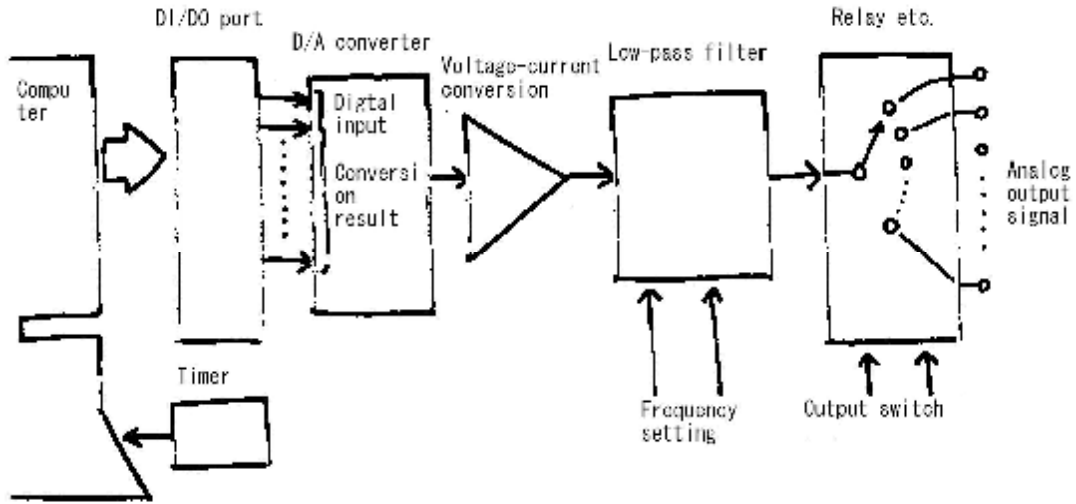
Şekil 1.18’de verilen devre örneğinde, iki ya da daha fazla analog girişin anahtarlama ve A/D dönüştürücünün bunları tek tek ele alıp işlemesi gösterilmektedir. İki ya da daha fazla analog işaretin düzenlemesiyle beraber A/D dönüştürücünün çoklayıcı (multiplexer) ile aynı anda dönüşümü gerçekleyebilmesi için, örnekleme devresinin de önceden yerleştirilmiş olması gerekir (Şekil 1.19).

Analog işaret örnekleme devresince korunurken, A/D dönüşüm işlemi çoklayıcı anahtarlama işlemini yaparken sıra ile gerçekleştirilecektir.

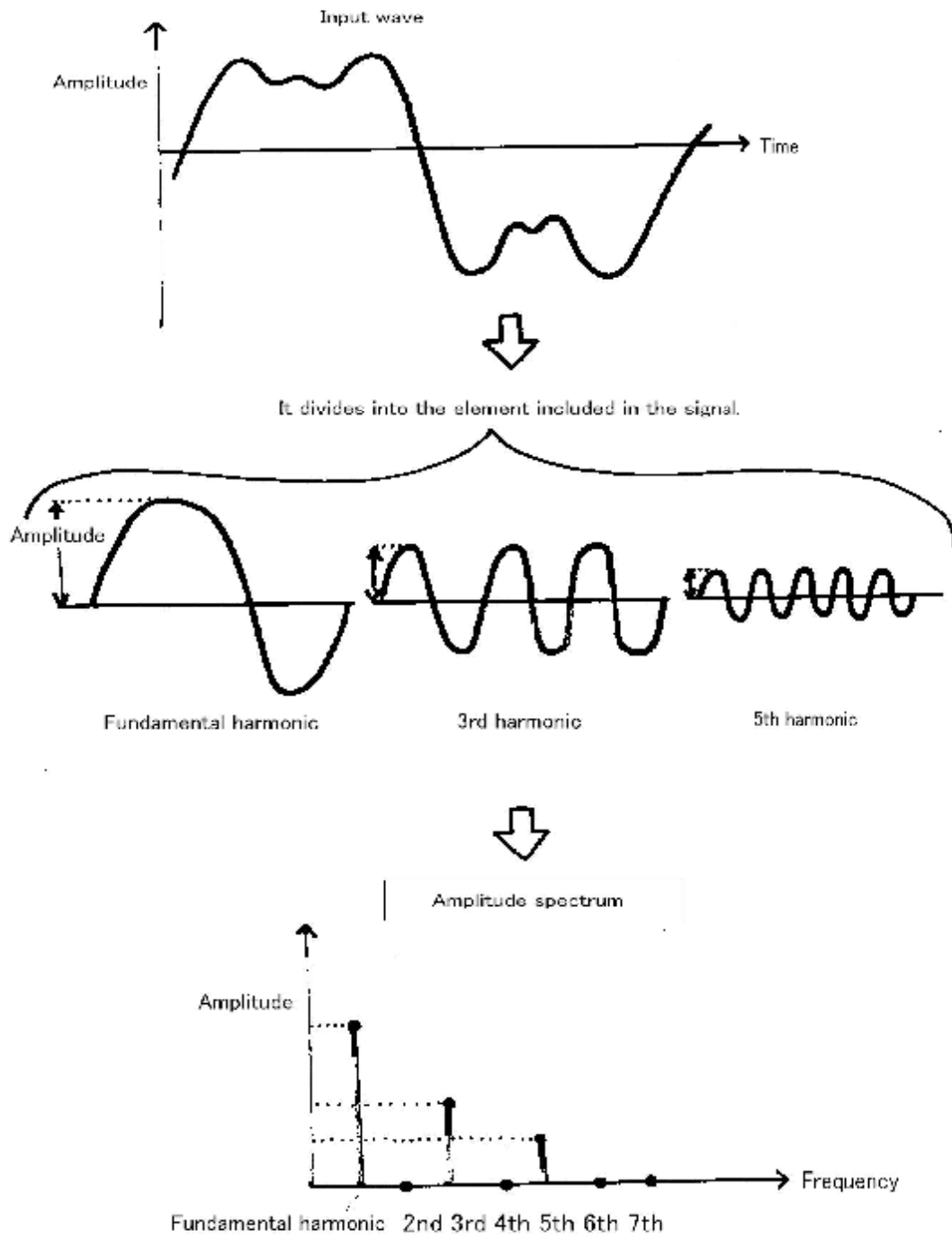
DI/DO portu, özel bir düğüm devresidir ve kişisel bilgisayarımızdan A/D dönüşümün güç çıkışının değişimini ve dönüşüm işleminin çıkış işaretlerinin kontrolünü sağlamaktadır. DI/DO portunca sağlanan işaret çoklayıcı, yükselteç (Amplifier) vb. tarafından kullanılır.

Zamanlayıcı (timer), örnekleme periyodunu üretir. Burada, zamanlayıcı güç çıkışının bilgisayardan uygulanmasının amacı dönüşüm işleminin başlangıç kontrolünün de bir kesme (interrupt) işlemi gibi varsayılmasının sağlanmasıdır. Başka bir yöntem de, A/D dönüşüm başlangıç işaretinin yerine zamanlayıcı çıkışını kullanarak, dönüşümün bitiş işaretiyle birlikte bilgisayara bir kesme yollamaktır.

D/A dönüşüm devre şeması A/D dönüşümün tam tersidir (Şekil 1.20). D/A dönüştürücü tarafından dönüşümü yapılacak olan sayısal kod bilgisi DI/DO portu kullanılarak bilgisayardan verilmektedir. Eğer D/A dönüşüm sonucunun doğruluğunun yüksek olması istenirse, güvenilirliği o derece yüksek olan bir gerilim dönüşüm devresine ihtiyaç duyulur. Bunun yanında; bir işlevsel yükselteç (Operational Amplifier) kullanılarak bu devrenin yapılması gayet kolaydır.



Şekil 1.19: İki ya da daha fazla analog girişe ve eşzamanlı örneklemeye imkân veren D/A dönüşüm devresi

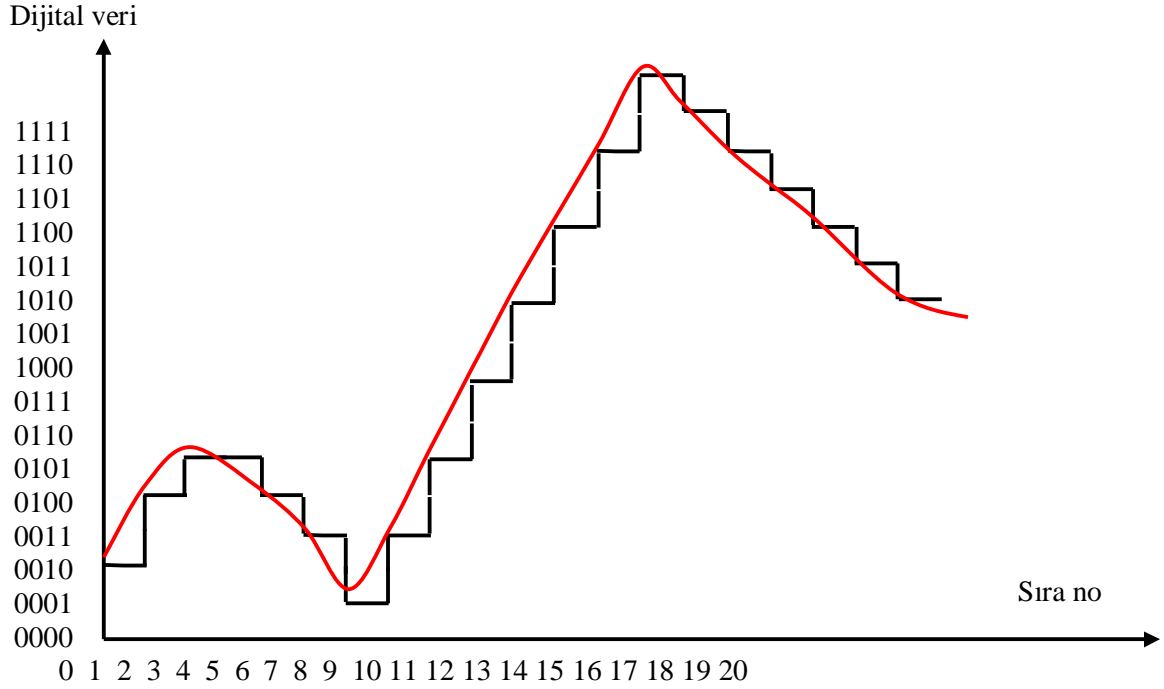


Şekil 1.20: Gerçek D/A dönüşüm terkibi

ÖRNEK 1

Aşağıda bir analog sinyalin dijital dönüşüm verileri görülmektedir. Bu dijital veriler bir DAC entegresine verilip entegre çıkışı bir alçak geçiren filtreden geçirince elde edilecek sinyalin gerçeğe en yakın şeklini belirtin.

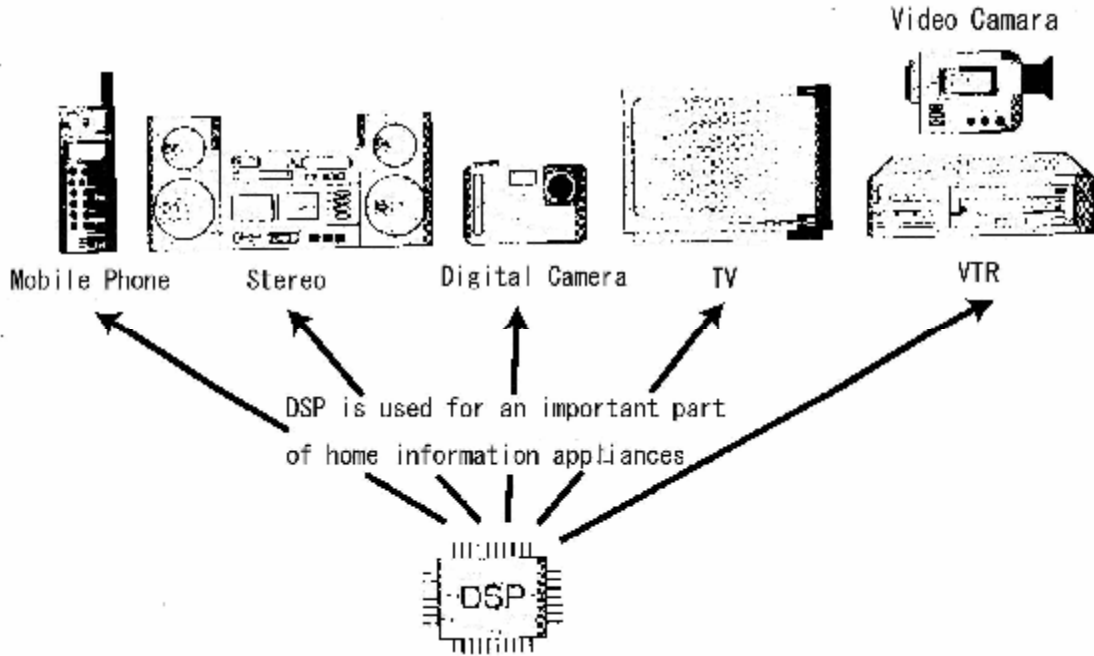
Sıra No:	Dijital veri:	Sıra No:	Dijital veri:	Sıra No:	Dijital veri:
1	0010	9	0101	17	1100
2	0100	10	0111	18	1011
3	0101	11	1001	19	1010
4	0101	12	1011	20	1001
5	0100	13	1101		
6	0011	14	1111		
7	0001	15	1110		
8	0011	16	1101		



1.4. Mikroişlemcinin Programlanması

1.4.1. DSP' nin Tanımlanması

Özellikle bilim ve mühendislik alanında, dijital sinyal işlem teknolojisinin önemi artmaktadır. Bilim dünyası ve mühendislik dijital sinyallerle icra edilmektedir. Sayısal İşaret İşleme, cep telefonu, müzik seti, televizyon, dijital kamera ve VTR gibi çok geniş çeşitlilikteki uygulamalarda kullanılır. Sayısal İşaret İşlemenin kullanım alanları aşağıda gösterilmiştir. Bu bölüm sizin DSP teknolojisini kolayca anlamanıza yardımcı olacaktır.

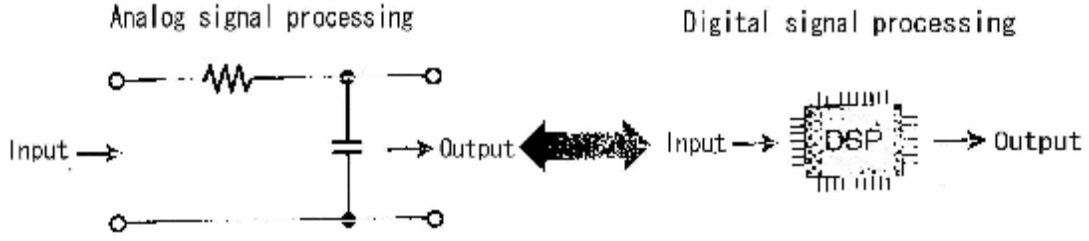


Şekil 1.21: DSP ve ev iletişim araç gereçleri

"DSP" nin anlamı "Dijital Signal Processing (Sayısal İşaret İşleme)" olmasının yanında "Dijital Signal Processor (Dijital Sinyal İşlemci)" olarak da ifade edilir. Bu bölümde, "DSP" ifadesinin anlamını "Dijital Signal Processor (Dijital Sinyal İşlemci)" olarak kullanacağız.

DSP yüksek performans gösteren ucuz işlemcilerdir. Mikroişlemcisinde genel amaçlı teknoloji yoğunlaştırılmıştır. Ayrıca, DSP yüksek hızda çalışır.

Şekil 1.22'de gösterildiği gibi, dirençler, reaktanslar, kondansatörler, transistörler ve opamplar gibi analog devre elemanlarının yerini DSP alır. Örneğin, reaktans yuvarlak bir elektrik kablosundan yapıldığı halde, reaktansın yaptığı iş bilgisayarda aritmetik işlemlerle yapılabilmektedir. DSP, gelişmiş dijital sinyal işlemlerinde iyidir. Bunun için DSP kullanarak analog devreler yapmak kolaydır.



Şekil 1.22: Analog devre ve DSP

DSP kullanımının günlük yaşamda kullandığımız eşyalarda birçok rahatlık getirdiği kolayca görülebilir. Örneğin, dijital cep telefonunun içine DSP yerleştirilmiştir.

Genel amaçlı bir mikroişlemcinin (örneğin Pentium 4) yerleştirilmesiyle oluşan bir ürünün fiyatı pahalı olur. Fakat yüksek performanslı ürünler basit bir mikro işlemciden (örneğin, Z-80) yapılamaz. Bunun için, DSP kullanımı gereklidir. DSP, küçük bir devrede yüksek performans ve düşük üretim maliyeti elde etmek için sinyal işlemlerinde etkin olarak çalışabilir. Böylece, yüksek performanslı ve düşük ücretli cep telefonu küçük bir devreye sahip olabilir.

Ayrıca, DSP internette aktif olarak kullanılır. İnternette, dijital veriler ağ üzerinden (network) iletilir. Halka açık internet ağ hattı ile bilgisayarı bağlamak için modem gereklidir. Tabii ki, tercih edilen internet hattının erişim hızı arttıkça modemle daha rahat bir internet erişimi sağlanır. İnternet erişim hızı yüksek olursa görüntüler daha hızlı alınır ve hemen görüntülenir. Bu tür erişim hızı yüksek modemlere DSP yerleştirilmiştir. DSP'nin internet için çok kullanışlı olduğunu belirtmeliyiz. Hatta internet dünya çapında bu hızlı yayılımını DSP'ye borçludur.

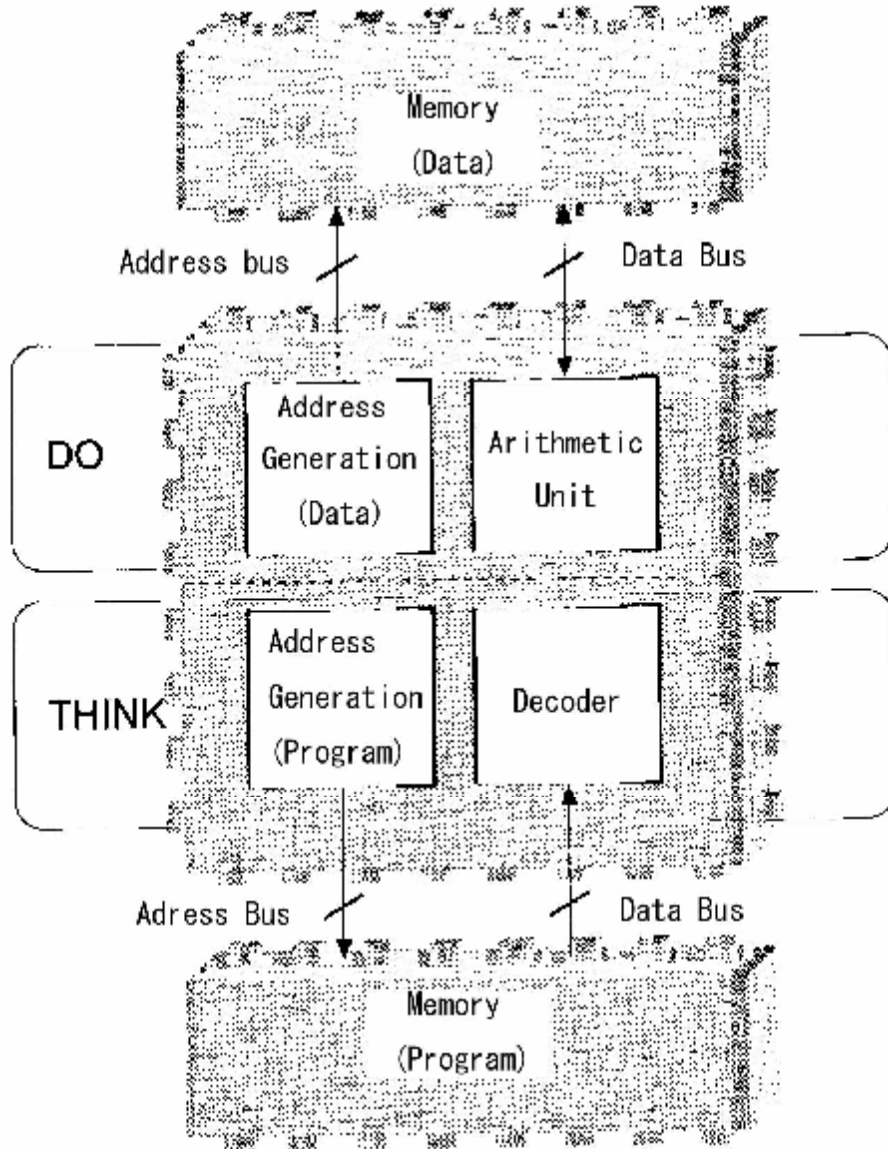
1.4.2. DSP'nin Mimarisi

1.4.2.1. Basit Bir İşlemcinin Mimarisi

İşlemci, bir sistemde "Beyin" olarak görev yapar. Bu "Beyin" in görevleri şunlardır:

- 1) Çevre ünitelerinden gelen veya giden verileri iletir
- 2) Mantıksal veya aritmetik işlemleri uygular.

Bu mekanizmaları elde etmek için, işlemci iki fonksiyonla donatılmıştır. Bu donanım kısımları Şekil 1.23'te de gösterildiği gibi "DO (veri işleme)" bölümü ve "THINK (program)" bölümü olarak isimlendirilir.



Şekil 1.23: İşlecimin içeriği

DO (Veri İşleme)

Veri işleme bölümü, veri alma mekanizması önemlidir. Bir işlemcide çevre ünitelerinden gelen veya giden verilerin iletimi için bir veri yolu "data bus" ve bir adres yolu "address bus" bulunur.

Address bus ve Data bus dijital verilerin iletim yoludur ve fiziksel olarak metal bir hattır. **Address bus**, kesin belirtilmiş iki veya daha fazla bilginin verilerini belirtmek için bir yoldur. **Adres jeneratörü** okunan veri için belirli bir adres üretir. Bu adres verisi adres

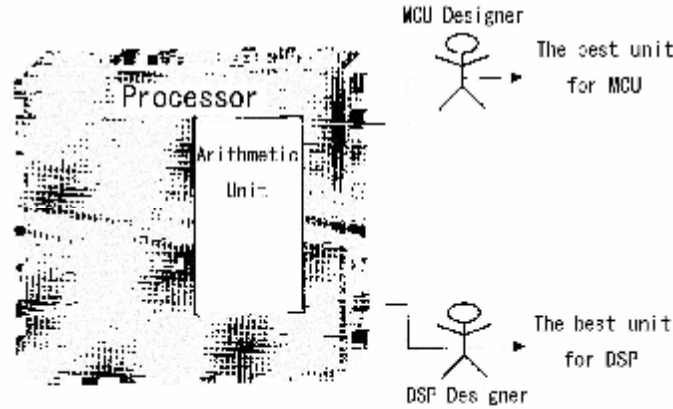
yolunu sürmek için gereklidir. Ayrıca, **aritmetik ünite** alınan verilere aritmetik veya mantıksal işlem uygular. Tabii ki bu da gereklidir.

THINK (Program)

İşlemcinin bir programa ihtiyacı vardır. Hafıza alanına bir program depolanır ve bu program hafızadan okunur. Bu mekanizma hafızadan okunan verilerle aynıdır. Kısaca, adres yolu ve veri yolu kullanılarak, program hafızadan okunur. Ancak, adres jeneratörünün durumu burada DO kısmındakinden farklıdır. Genel olarak, programın yeri ve verilerin alanı farklıdır. Her bir adres jeneratörü program ve veri için bir adres yolu sürer. THINK bölümünde, programın adres jeneratör yazmacına Program Counter (PC) denir. Programda işlenen komutun adresini saklar. Hafızadaki programın biçimi "0" ve "1" 'in birleşiminden oluşur. Burada verilerin biçimi tamamen aynıdır. Bu nedenle, bir işlemci için veri veya program olup olmadığı gerçekten farklı değildir. Bu nedenle, çevirme mekanizmasına ihtiyaç vardır. Bu bir "decoder (kod çözücü)"' dir. Buna kod çözücü veya denetleyici denebilir.

1.4.2.2. DSP'nin İşlemci Mimarisi

Herhangi bir işlemcide, işlemci mimarisinin temel kavramı daha önce açıkladığımız gibidir. Öyleyse farklılık nedir? İşlemcinin temel kavramı aynı olsa bile, her bir ögenin nasıl tasarlandığı "aritmetik bölümde" veya "adres jeneratöründe" olduğu gibi somut bir şekilde farklıdır (Şekil 1.24'te gösterildiği gibi). Eğer bir işlemci tasarımcısı olursanız, hangi ögeyi tasarlıyorsunuz? Bu soru önemlidir. Her bir işlemcinin cevabı farklıdır.



Şekil 1.24: Ögeler amaca göre tasarlanır

Bir işlemcinin amaçlarının en iyi şekilde olması için tasarımcı işlemcinin her bir ögesini tasarlar. DSP, dijital sinyal işlemleri için özel bir tasarıma sahiptir. Bu nedenle, DSP'nin tasarımı Sayısal İşaret İşlemenin ne olduğunu anlamak için gereklidir. Bu işlemler aşağıda açıklanacaktır.

1.4.3. Sayısal İşaret İşleme İçin Gereken Hesaplamalar

Dijital sinyal işlemleri için aşağıda gösterilen ifade kullanılır.

$$Y = \sum_{n=1}^{100} A_n \cdot X_n \dots\dots\dots(i)$$

Formülde n için 1’den 100’e kadar değerlerini yazarsak aşağıdaki açılımı elde ederiz:

$$Y = \sum_{n=1}^{100} A_n \cdot X_n = A_1 \cdot X_1 + A_2 \cdot X_2 + A_3 \cdot X_3 + \dots + A_{100} \cdot X_{100} \dots\dots\dots(ii)$$

İfademiz yalnız çarpımların toplamı haline dönüşür. Kısaca, dijital sinyal işlemleri çarpımların toplanmasıyla oluşur.

1.4.4. TMS320C5000 DSP serisi

DSP içinde dijital sinyal işlemleri verimli bir şekilde işler. Çarpılan değerlerin toplanarak birleştirilmesiyle hesaplanır. Bu kitapta, biz Texas aletlerinin bir serisi olan TMS320C5000DSP (C5000) serisini kullanarak DSP’nin mimarisini öğreneğiz.

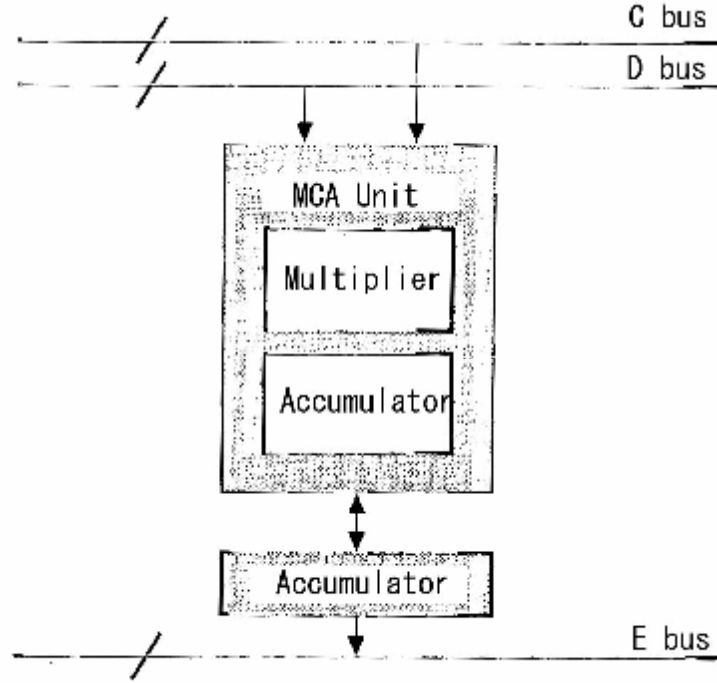
1.4.4.1. MAC ünitesi, C bus ve D bus

Dijital sinyal işlemlerinin anlamı çarpanlar ve toplananlar ünitesidir.Bu üniteye kısaca MAC denir ("Multiply and Accumulation" MAC). MAC ünitesinde bu iki fonksiyonda birlikte yapılır.Çünkü çarpmak için en az iki veri gereklidir ve aynı zamana ait iki verinin alınması dijital sinyal işlemlerinin verimini artırır. Böylece, iki veri yolu MAC ünitesi için hazır olacaktır. Bu veri yollarının isimleri C bus ve D bus’tır.

1.4.4.2. Accumulator (Akümülatör) ve E bus

İşlemlerin sonuçları geçici olarak saklanır. Genellikle, bu alana (register) "Accumulator (akümülatör)" denir . Bu register(akümülatör) toplamların sonuçlarını saklar. Ayrıca akümülatör içindeki değer bus içinde son hafızada da saklanır. Aksi halde, akümülatörün içindeki işlem sonuçları kaybolur.Çünkü bir sonraki işlem akümülatörün üzerine yazılır. Doğal olarak, veri yolu bu iş için gereklidir.

Çünkü C bus ve D bus hafıza ile bağlantılıdır.Bu nedenle hafızaya veri yazılımı için bu veri yollarının kullanıldığı kesindir. Ancak,DSP’de okunan ve yazılan veriler aynı zamanda alınır. DSP’ nin yüksek hızlı olması için,DSP içine E bus yerleştirilir. Şekil. 1.25’te gösterildiği gibi, E bus diğer veri yollarından bağımsızdır ve yalnız yazma yönüne sahiptir.



Şekil 1.25: Aritmetik ünitesi ve veri yolları

1.4.4.3. Aritmetik lojik ünitesi (Toplama Ünitesi)

Dijital sinyal işlemleri için, belirli bir yere kadar gerekli bir fonksiyonla donatılmıştır. Çeşitli dijital sinyal işlemlerinin değişiminin algoritması araştırılır, bu nedenle bu diğer toplama ünitesini(Aritmetik lojik ünitesi) eklemek için etkindir.

Örneğin, FIR filtresinin tipik simetrik ifadesi (iii) ile bu açıklanır:

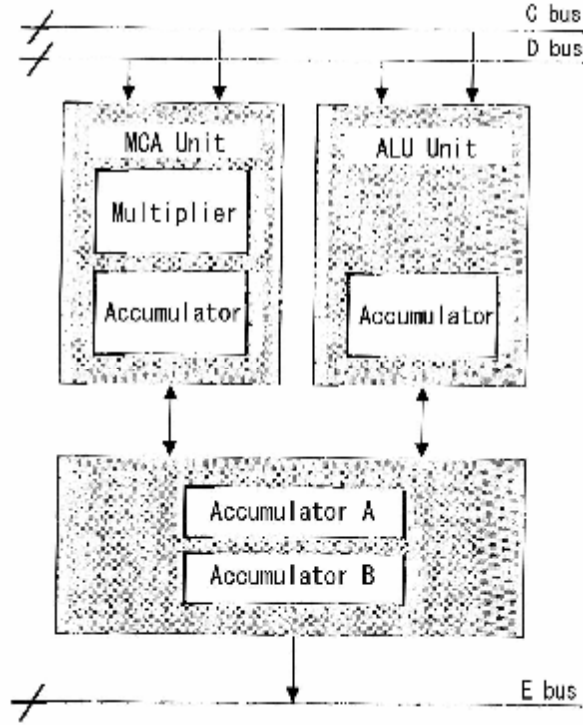
$$Y(n) = a_0 \cdot x_0 + a_1 \cdot x_1 + a_2 \cdot x_2 + a_3 \cdot x_3 + a_4 \cdot x_4 + a_5 \cdot x_5 + a_6 \cdot x_6 + a_7 \cdot x_7 \quad \text{.....(iii)}$$

Ancak, $a_4 = a_3, a_5 = a_2, a_6 = a_1, a_7 = a_0$,

Eşitliği yerine koyulursa aşağıdaki ifade elde edilir:

$$\begin{aligned} Y(n) &= a_0 \cdot x_0 + a_1 \cdot x_1 + a_2 \cdot x_2 + a_3 \cdot x_3 + a_3 \cdot x_4 + a_2 \cdot x_5 + a_1 \cdot x_6 + a_0 \cdot x_7 \\ &= a_0 \cdot (x_0 + x_7) + a_1 \cdot (x_1 + x_6) + a_2 \cdot (x_2 + x_5) + a_3 \cdot (x_3 + x_4) \quad \text{.....(iv)} \end{aligned}$$

Bu uzun mesafeli infrared redrasyon tip alçak geçiren filtrenin filtre katsayılarıdır ve merkezin etrafında bu katsayılar simetriktr. Şekil.1.26'da gösterildiği gibi yalnız MAC ünitesinde hesaplamalar yapıldığı halde diğer aritmetik ünitenin kullanımı ile, toplama yapılırken çarpmanın da yapılması mümkündür. Bu mekanizma tarafından hesaplama süresi yarıya düşer. Diğer algoritmanın içinde de benzer işlerin yapıldığı düşünülebilir.



Şekil 1.26: C5000 Serisi Aritmetik ünitesi ve Bus

Çünkü aritmetik ünitenin sayısı 2'ye artmıştır. İşlem sonuçlarının saklandığı register sayısı da 2'ye artmıştır. Bu DSP'nin C5000 serisinin temel bir yapısıdır. C5000 serisi DSP'nin yüksek performans, düşük güç ve düşük ücret dengesini sağlayabilmektedir.

ÖRNEK 2

CCS Programı Kullanılarak Mikro İşlemci Nasıl Programlanır

İlk olarak "Hello, DSP !" cümlesini görüntüleyen program hakkında düşünelim. Kullanacağımız işlemci DSP olmasına rağmen yazacağımız program klasik C programlama dilinde olacaktır. Program aşağıdaki gibidir:

```
#include <stdio.h>

void main()
{
    printf("Hello,DSP !\n");
}
```

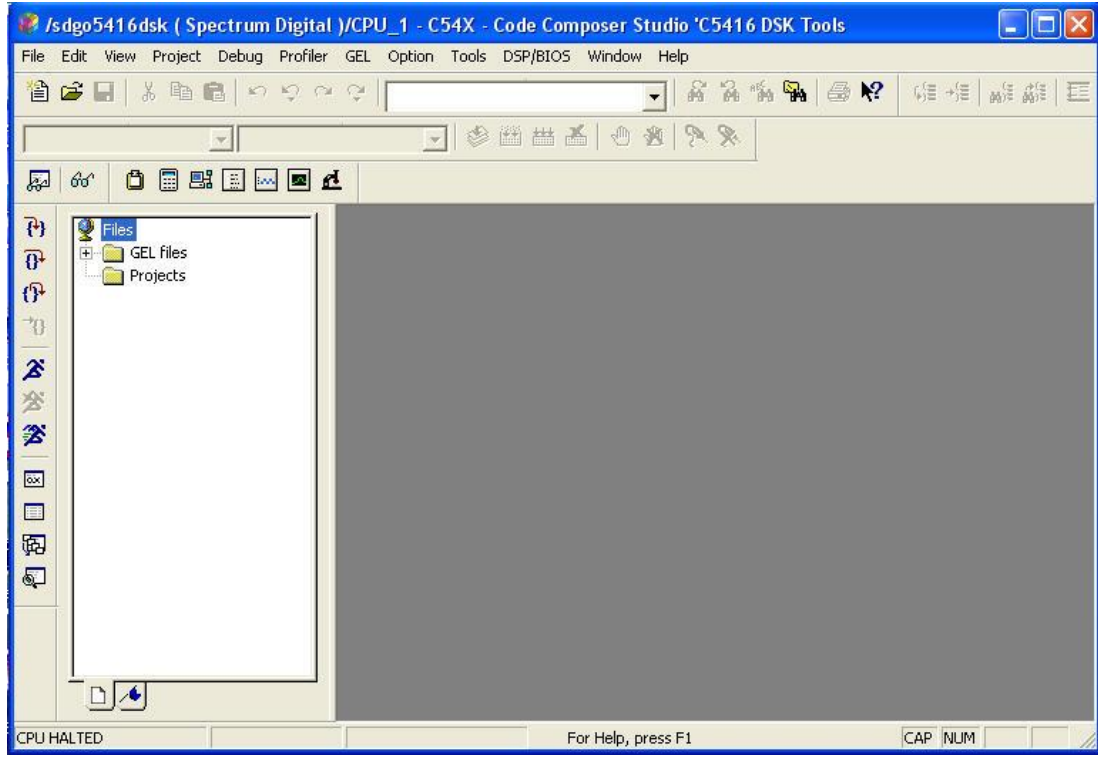
Programı çalıştırma yolu kullandığınız sisteme bağlıdır. Program seçtiğiniz herhangi bir sistem üzerinde yazılır, kaydedilir ve makine diline derlenir. Örneğin Windows'ta, dosya isminin uzantısı ".c" olur. Yani "hello. c" gibi. Programı derledikten sonra, çalıştırılabilir dosya tipi "hello. exe" oluşturulur. DOS'un komut satırından bu program çalıştırıldığı zaman ekranda "Hello, DSP!" gözükür.

Şimdi, DSP sistemi içinde bunun nasıl yapıldığına bakalım. DSP sistemi Windows gibi bir ekrana sahip değildir. Ancak DSP bir hata ayıklayıcı (Geliştirilmiş çevre üniteleri programlayıcısı) ile birlikte çalışır. C5000 çevre ünitesinin programı olan "Code Composer Studio" ile DSP kontrol edilir. Kısacası , "Hello, DSP!" cümlesi "Code Composer Studio (CCS) " program editörü üzerinde görüntülenir. Bu kitapta, Texas çevre ünitesinin programı olan "Code Composer Studio 'C5416 Version 2.10.05" (biz ona kısaca CCS diyeceğiz) kullanılmaktadır.

CCS'nin Çalıştırılması

Eğer CCS bilgisayarınızda kurulu ise, bilgisayarınızın masa üstünde CCS ikonu gözükecektir. Bu ikonu çift tıklayın ve CCS programını başlatın. veya, Windows- Start Menu-Programlar'dan da CCS seçilerek program çalıştırılır.





Şekil 1.27: Code Composer Studio program ekranının açılması


Yeni Bir Proje Yapmak

Hazırlık (Dosyaların Düzenlenmesi)

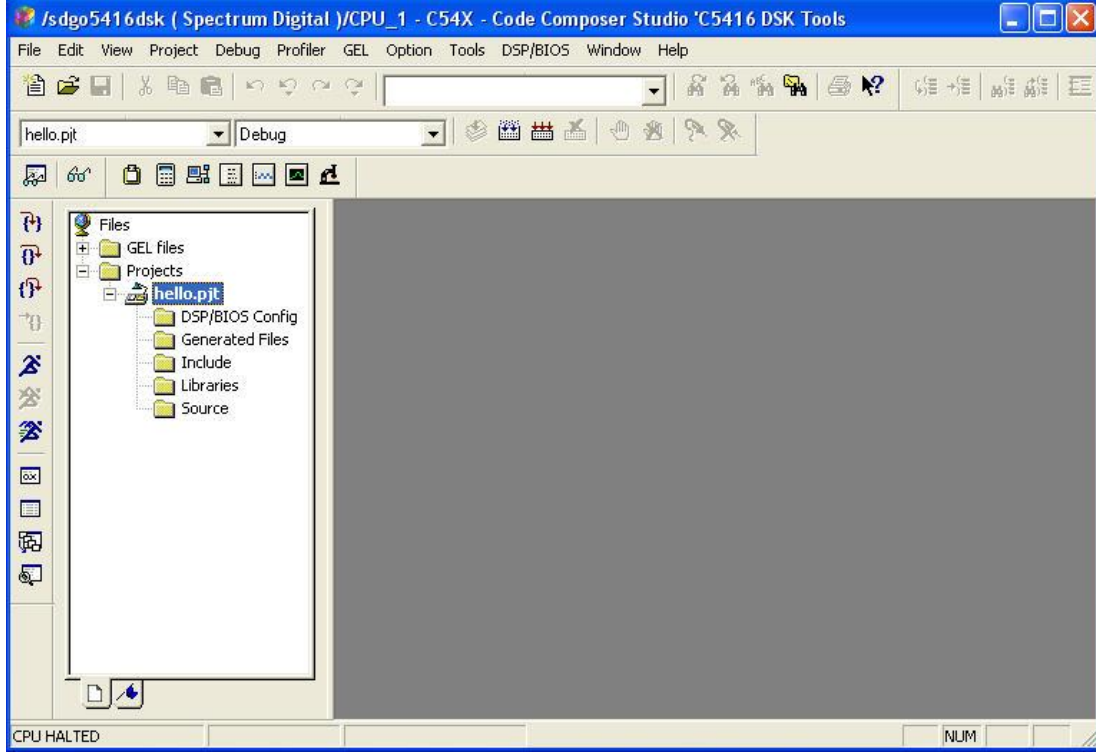
CCS programının kurulumu sırasında "C:" sürücüsünün kök dizininde "ti" dosyası oluşturulur. "ti" dosyası içinde de "myprojects" dosyası oluşturulur. Yazdığımız programların dosyalarını bu dosyanın içine kaydedip proje oluşturacağız. Şimdi "hello" dosyası için projemizi yapalım.

"Hello.PJT" nin Yapılışı

"Project" menüsünden "New" komutunu seçin. "Project Creation" isimli pencere açılır. Sonra, projenin ismi ve yeri belirtilir (şimdi proje ismi olarak , "hello" girin). Projenin ismi girilince yeri otomatik olarak "Location" kutucuğunda belirir. Eğer proje yeri otomatik belirtilmemişse, doğru adresi girin (bu durumda doğru adres "c:\ti\myproject\hello\" olacaktır). Projenin uzantısı otomatik olarak ".pjt" olacaktır. Böylece, projemiz "hello.pjt" yapılmış olur.

CCS program editörünün sol tarafında "Project View" penceresi gözüktür. Bu pencerede oluşturduğunuz "hello.pjt" proje dosyasının yanındaki  işaretine tıklayarak proje dosyasının içerdiği 5 tane alt dosyayı görüntüleyin. Yeni bir proje oluşturulduğunda bu dosyalar otomatik oluşturulur ve başlangıçta içlerinde hiçbir şey yoktur. Bu dosyaların

isimleri sırasıyla şöyledir: "DSP/BIOS Config", "Generated Files", "Include", "Libraries" ve "Source".



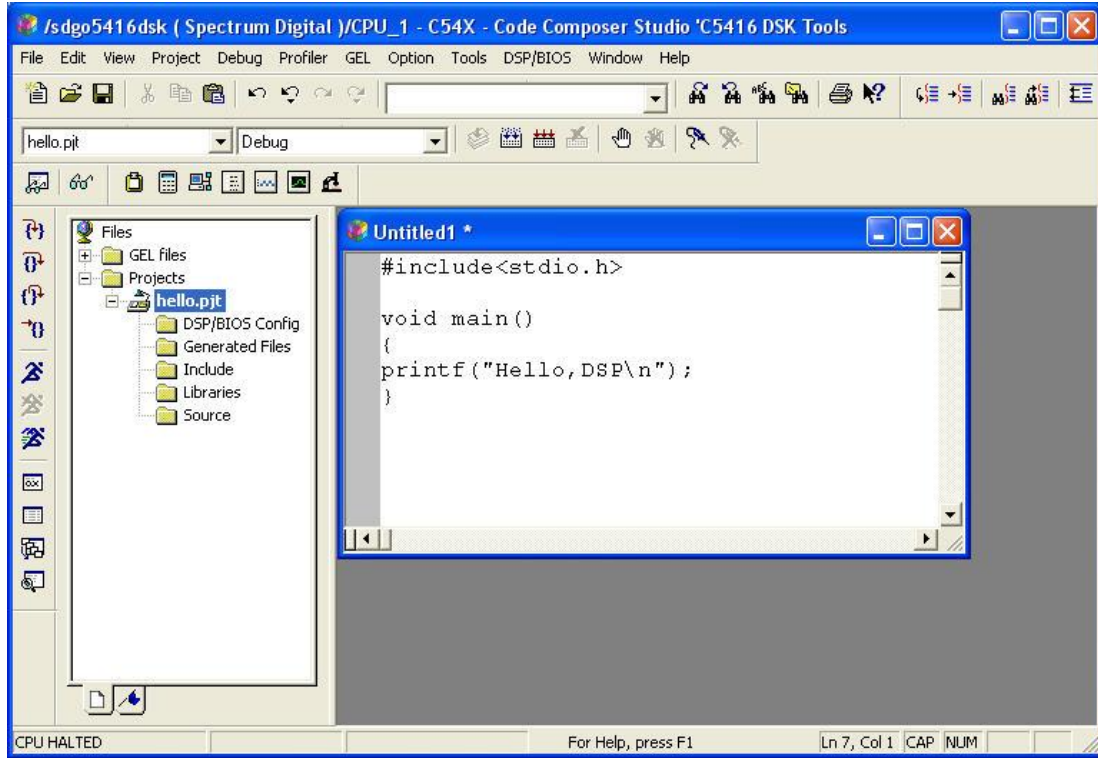
Şekil 1.28: "hello.pjt" 'nin yapısı

Projeye Bazı Dosyaların Eklenmesi

Projenin, kaynak dosya, konfigürasyon (yapılandırma) dosyası gibi bazı dosyalara ihtiyacı vardır.


Kaynak Dosya (Source File)

Yeni bir kaynak dosya yapmak için (örneğimizdeki "hello.c" gibi), menüden "File" -> "New" -> "Source file" seçildiği zaman "Untitled1" program penceresi açılır, buraya Liste 6-1'deki C dilindeki programı yazın. Bu programı kaydetmek için File menüden "Save As" seçeneğini seçin ve dosyanın ismini "hello.c" olarak girin ve Save butonunu tıklayın.




Şekil 1.29: Kaynak programın girilmesi

Projeye Kaynak Dosyanın Eklenmesi

Projeye kaynak program "hello.c" 'nin eklenmesi için, Project menüden "Add Files to Project" seçeneğini seçin. Sonra, ekrana gelen pencerenin içinden "hello.c" seçin ve open butonunu tıklayın. Hello.pjt dizini altındaki source dosyasının yanındaki  işaretine tıklayarak "hello.c" kaynak dosyasının eklendiğini görebilirsiniz.

Bir Konfigürasyon Dosyasının Yapılması Ve Eklenmesi

Yeni bir konfigürasyon dosyası oluşturmak için menüden "File" -> "New" -> "DSP/BIOS Configuration..." seçin. Kullandığımız DSP'nin konfigürasyon dosyası olan "dsk5416.cdb" dosyasını seçin ve "OK" tıklayın. Ekranda "Config1" isimli yeni bir pencere gözükecektir. Bu dosyayı "hello.cdb" olarak kaydettiğimizde oluşturulan konfigürasyon dosyasını projemize eklemiş oluruz. File menüden "Save As" seçeneğini seçin ve konfigürasyon dosyasının ismini "hello.cdb" olarak girin ve Save butonunu tıklayın. Şimdi oluşturulan bu konfigürasyon dosyasını (.cdb uzantılı dosyayı) projeye eklemeliyiz. Menüden "Project" -> "Add Files to Project" seçeneğini tıklayın. Ekrana gelen pencerede "File of type" kutucuğundan konfigürasyon dosyası seçeneğini "Configuration File (*.cdb)" seçin. konfigürasyon dosyası gözükecektir (örneğimizdeki "hello.cdb" dosyası gözükür). Dosyayı seçin ve "open" butonunu tıklayın, "hello.cdb" dosyasının projeye eklendiğini "Project Creation" penceresinden "DSP/BIOS Configuration..." dosyasının yanındaki  işaretine tıklayarak görebilirsiniz.

Compiler ve Linker Seçenekleri

Konfigürasyon için kullandığımız "far call" seçeneğine değiştirmemiz gerekir. Derleyici programı derlerken eğer "Use Far Calls" seçeneği seçili olursa geniş bir program yapılabilir. Bu seçeneği seçmek için menüden "Project" -> "Build Option" seçeneğini tıklayın gelen menüde "Compiler" sekmesini tıklayın ve Category 'nin içinden "Advanced" seçin. Buradan "Use Far Calls" seçeneğini tıklayın. Yine aynı menüden bu sefer "Linker" sekmesini seçin ve Category 'nin içinden "Basic" seçin. Buradaki "Include Libraries" kutucuğuna "dsk5416f.lib" yazın ve "OK" tıklayın.


Linker Komut Dosyası

Konfigürasyon dosyası yapıldığı zaman linker komut dosyası otomatik olarak yapılır. Burada, sadece linker dosyasını (.cmd uzantılı dosyayı) projeye eklemeliyiz. Menüden "Project" -> "Add Files to Project" seçeneğini tıklayın. Ekranı gelen pencerede "File of type" kutucuğundan linker komut dosyası seçeneğini "Linker Command File(*.cmd)" seçin. Komut dosyası gözükecektir (örneğimizdeki "hellocfg" dosyası gözüktür). Dosyayı seçin ve "open" butonunu tıklayın , "hellocfg.cmd" dosyası projeye eklendiğini "Project Creation" penceresinden de görebilirsiniz.

Otomatik Olarak Yüklenen Seçenekler

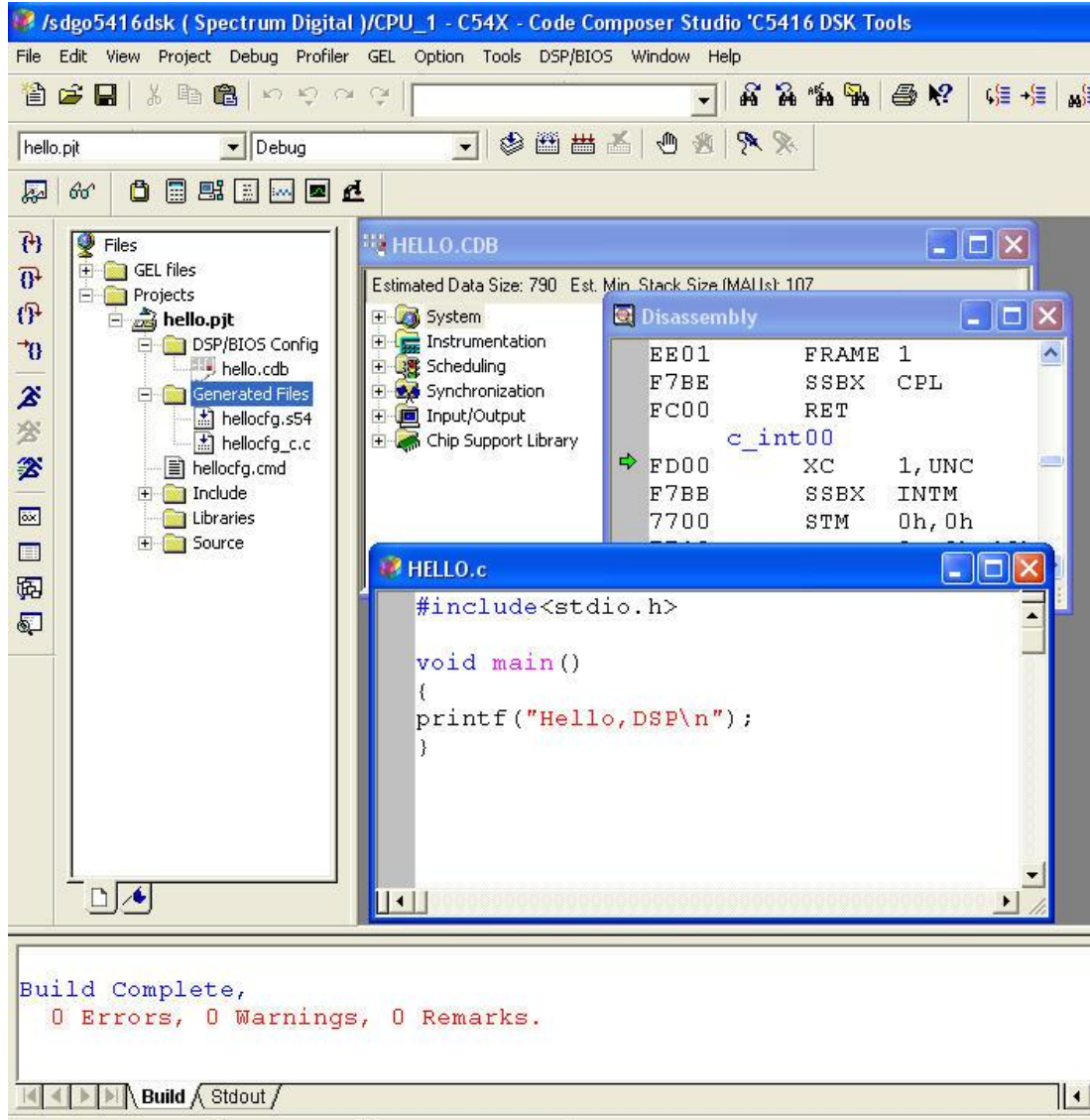
Proje derlendikten sonra otomatik olarak programın DSP borduna yüklenmesi için CCS programında bazı seçenekler aktif yapılır. Bunun için "Option" menüden -> "customize" -> seçeneğini seçin ve ekrana gelen menüde "Program Load Options" sekmesini tıklayın. Burada "Load Program After Build" seçeneğinin aktif olması için yanındaki kutucuğu tıklayın.

Programın Derlenmesi

Buraya kadar yukarıdaki işlemlerin hepsi yapılarak bütün gerekli dosyalar tamamlanmış olur. Projeyi tamamlamak için ise kaydetmek zorundayız. Project -> menüden Save seçeneğini tıklayın. Çalışabilir bir dosya yapmak için derleme "Build" yapılması gerekir. CCS program editöründe, sırasıyla "Projects" -> menüden "Rebuild All" seçeneğini seçin veya araç çubuğu üzerindeki  (Rebuild All) ikonunu tıklayın. Bir süre sonra , eğer bir hatanız yoksa, çalışabilir dosya (hello.out) oluşturulur. Hem de çalışabilir dosya (hello.out) DSP borduna gönderilmiş olur. Şekil 6.4'te gösterildiği gibi CCS editörünün en altındaki "Build" ekranında :


Build Complete,
0 Errors, 0 Warnings, 0 Remarks.

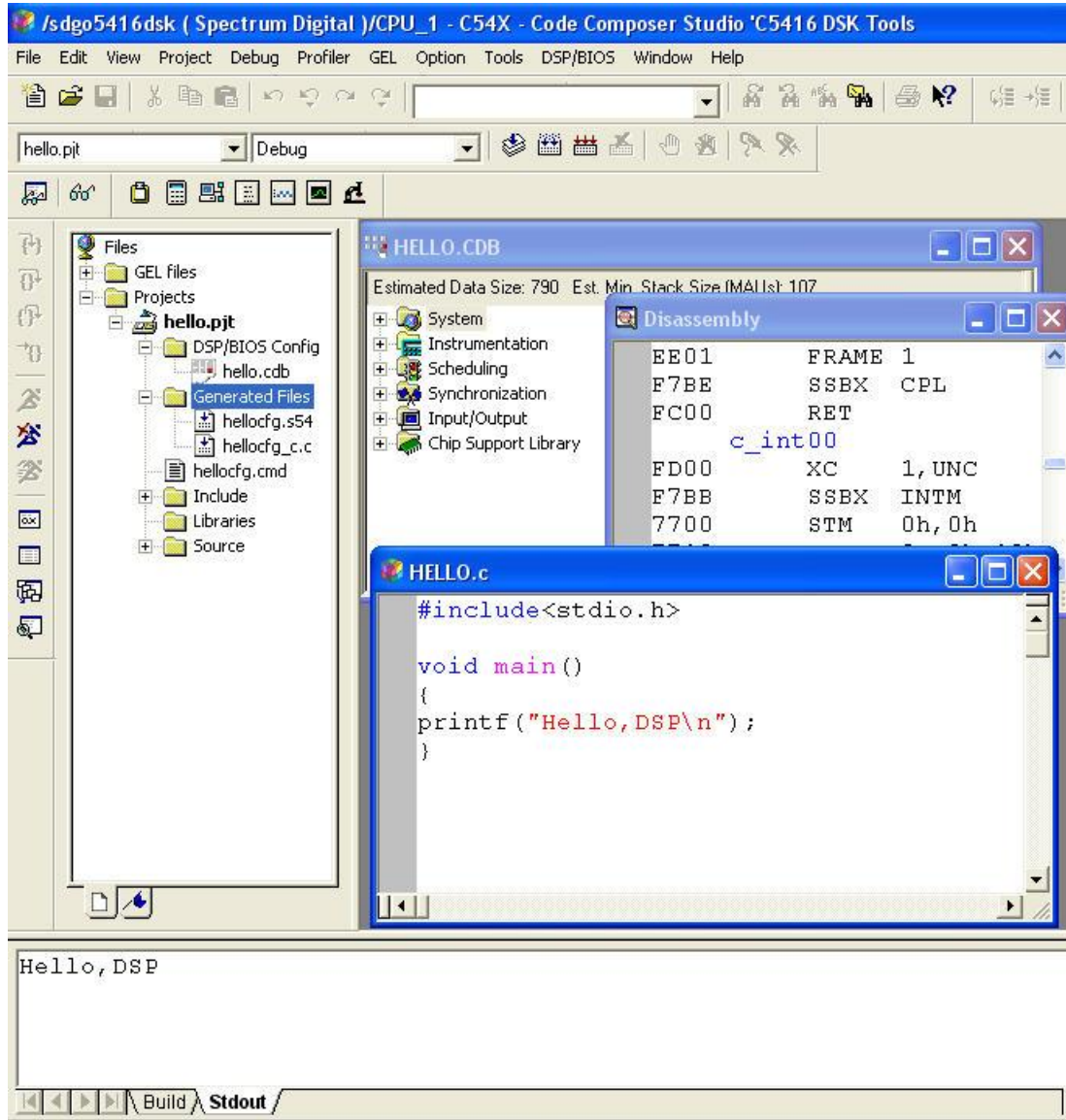
mesajını görürsünüz. Eğer programınızda hata varsa derleme yapıldıktan sonra program DSP kartına gönderilmez ve hatalar yine "Build" ekranında sırasıyla yazar.




Şekil 1.30: Derlemenin hatasız olarak tamamlandığı durum

Programın Çalıştırılması

Programı çalıştırmak için, araç çubuğu üzerindeki  (Run) butonu kliklenir. CCS editörünün en altındaki "Stdout" ekranında "Hello ,DSP! " mesajı gözükecektir.




Şekil 1.31: Uygulamanın sonucu



Programın görüntüsü bitmiş gibi olsa da otomatik olarak programın çalışması durmaz. Programın çalışmasını sonlandırmak için araç çubuğu üzerindeki  (Halt) butonunu tıklayın.

Eğer programı tekrar baştan çalıştırmak isterseniz, "Debug" menüden "Restart" seçeneğini tıklayın (Burada "Reset CPU" seçeneğini sakın seçmeyin). Restart seçeneği ile Program Counter içeriği programın başlangıç adresi "c_int00" olur ve program tekrar baştan çalışmaya başlar. CCS editörünün en altındaki "Stdout" ekranında "Hello ,DSP! " mesajının diğer satıra tekrar yazıldığı görülerek, programın tekrar çalıştırıldığı daha iyi anlaşılır.

UYGULAMA FAALİYETİ

CCS Program Editörüne Adını Soyadını Yazdıran Program	
İşlem basamakları	Öneriler
<p>Ø CCS programını açınız.</p> <p>Ø “Project” menüden “New” komutunu seçiniz.</p> <p>Ø “Project Creation” isimli açılan pencereye proje ismi olarak “isim” yazınız.</p> <p>Ø “File” menüden “New” -> “Source File” seçerek ekrana gelen “Untitled1” program penceresine C dilinde adınızı ve soyadınızı ekranda yazacak programı yazınız.</p> <p>Ø “File” menüden “Save As” seçeneğini seçerek dosyanın ismini “isim.c” olarak yazın ve “Save” butonunu tıklayınız.</p> <p>Ø “Project” menüden “Add Files to Project” seçeneğini seçiniz. Ekrana gelen pencerenin içinden “isim.c” klikleyerek kaynak dosyayı projenize ekleyiniz.</p>	<p>Ø “Project Creation” penceresinde projenin ismi girilince otomatik olarak “Location” kutucuğunda belirir.Eğer proje yeri otomatik belirtilmemişse doğru adresi aşağıdaki gibi giriniz.</p> <p>“c:\ti\myproject\isim\”</p> <p>Ø CCS program editörünün sol yanındaki “Project view” penceresinde “isim.pjt” proje dosyasının yanındaki + işaretine klikleyerek proje dosyasının içerdiği "DSP/BIOS Config", "Generated Files", "Include", "Libraries" ve "Source" isimli içi boş 5 alt dosyanın varlığını teyit ediniz.</p> <p>Ø “isim.c” kaynak dosyasının projenize eklendiğini “isim.pjt” dizini altındaki “source” dosyasının yanındaki + işaretine klikleyerek gözleyiniz.</p>

CCS Program Editörüne Adını Soyadını Yazdıran Program	
İşlem basamakları	Öneriler
<p>Ø Yeni bir konfigürasyon dosyası oluşturmak için menüden "File" -> "New" -> "DSP/BIOS Configuration..." seçiniz. Kullandığımız DSP'nin konfigürasyon dosyası olan "dsk5416.cdb" dosyasını seçin ve "OK" tıklayınız. Ekranda "Config1" isimli yeni bir pencere gözükecektir. Bu dosyayı "hello.cdb" olarak kaydettiğimizde oluşturulan konfigürasyon dosyasını projemize eklemiş oluruz. File menüden "Save As" seçeneğini seçin ve konfigürasyon dosyasının ismini "hello.cdb" olarak girin ve Save butonunu tıklayınız.</p> <p>Ø Menüden "Project" -> "Add Files to Project" seçeneğini tıklayın. Ekrana gelen pencerede "File of type" kutucuğundan konfigürasyon dosyası seçeneğini "Configuration File (*.cdb)" seçin. Konfigürasyon dosyası "isim.cdb" gözükecektir. Dosyayı seçiniz ve "open" butonunu tıklayınız.</p> <p>Ø "Project" -> "Build option" seçeneğini tıklayın gelen menüde "Compiler" sekmesini tıklayınız ve Category 'nin içinden "Advanced" seçin. Buradan "Use Far Calls" seçeneğini tıklayın. Yine aynı menüden bu sefer "Linker" sekmesinde Category 'nin içinden "Basic" seçiniz. Burada "Include Libraries" kutucuğuna "dsk5416f.lib" yazarak "OK" tıklayınız.</p>	<p>Ø "isim.cdb" dosyasının projeye eklendiğini "Project Creation" penceresinden "DSP/BIOS Configuration..." dosyasının yanındaki  işaretine tıklayarak görebilirsiniz.</p> <p>Ø Konfigürasyon dosyası yapıldığı zaman linker komut dosyası otomatik olarak yapılır. Burada, sadece linker dosyasını (.cmd uzantılı dosyayı) projeye eklemeliyiz.</p>

CCS Program Editörüne Adını Soyadını Yazdıran Program	
İşlem basamakları	Öneriler
<p>Ø Menüden "Project" -> "Add Files to Project" seçeneğini tıklayın.Ekrana gelen pencerede "File of type" kutucuğundan linker komut dosyası seçeneğini "Linker Command File(*.cmd)" seçin. Komut dosyası gözükecektir (örneğimizdeki "hellocfg" dosyası gözüktür). Dosyayı seçin ve "open" butonunu tıklayınız.</p> <p>Ø "Option" menüden -> "customize" -> seçeneğini seçin ve ekrana gelen menüde "Program Load Options" sekmesini tıklayın.Burada "Load Program After Build" seçeneğinin aktif olması için yanındaki kutucuğu tıklayınız.</p> <p>Ø Project -> menüden save seçeneğini tıklayınız.</p> <p>Ø "Projects" -> menüden "rebuild All" seçeneğini seçin veya araç çubuğu üzerindeki  (Rebuild All) ikonunu tıklayınız.</p> <p>Ø Programı çalıştırmak için, araç çubuğu üzerindeki  (Run) butonu tıklayınız.</p>	<p>Ø "hellocfg.cmd" dosyası projeye eklendiğini "Project Creation" penceresinden de görebilirsiniz.</p> <p>Ø Proje derlendikten sonra otomatik olarak programın DSP borduna yüklenmesi için CCS programında bazı seçenekler aktif yapmalısınız.</p> <p>Ø Projeyi tamamlamak için ise kaydetmek zorundasınız.</p> <p>Ø Çalışabilir bir dosya yapmak için derleme "Build" yapmalısınız.</p> <p>Ø eğer bir hatanız yoksa, CCS editörünün en altındaki "Build" ekranında :</p> <p>“Build Complete, 0 Errors,0 Warnings,0 Remarks.”</p> <p>mesajını görürsünüz.</p> <p>Ø Eğer programınızda hata varsa derleme yapıldıktan sonra program DSP borduna gönderilmez ve hatalar yine build ekranında sırasıyla yazar.</p> <p>Ø CCS editörünün en altındaki "Stdout" ekranında isminizi görmelisiniz.</p>

ÖLÇME VE DEĞERLENDİRME

OBJEKTİF TESTLER (ÖLÇME SORULARI)

Aşağıdaki soruları doğru “D” veya yanlış “Y” olarak cevaplayınız.

- (.....) 1. Sürekli değişen fiziksel değerler bir analog sinyali oluşturur.
- (.....) 2. Rakam veya karakter formunda görüntülenen sinyaller bir dijital sinyali oluşturur.
- (.....) 3. Dijital bir ses sinyalini analog sinyal olarak kayıt edebiliriz.
- (.....) 4. 0-100V aralığındaki analog voltaj değeri dijital formata sekiz bit binary sayılar kullanılarak dönüştürülmektedir. Dijital seviyenin kalitesi ya da hassasiyeti $100/(2^8-1) = 0,39V$ seviyesindedir.
- (.....) 5. ADC (Analog / Dijital Convertör) entegresinde analog giriş terminalinden gelen analog sinyal dönüşüm başlatma sinyali ile dijital sinyale dönüştürülür ve dönüşüm sonu sinyali ile dijital çıkış terminaline aktarılır.
- (.....) 6. Kesin doğru bir A/D dönüşüm yapmak için bir periyot içerisinde analog sinyalin değişmediği durumlarda alınan kesin değerler korunur. Bu işlemi yapan devrelere tutucu devre denir.
- (.....) 7. Tutucu devre analog giriş ile dijital sinyal çıkış terminali arasına yerleştirilir ve A/D dönüştürücüye yollanıp kullanılır.
- (.....) 8. D/A dönüşüm temel olarak A/D dönüşümün tersidir.
- (.....) 9. DAC (Dijital / Analog Convertör) entegresindeki çevirme işlemi anlık yapıldığından, çevrim işlemi sonunda A/D dönüşümde ki bitiş sinyali gibi bir kontrol sinyaline ihtiyaç yoktur.
- (.....) 10. Bir DAC çıkışındaki sinyal alçak geçiren filtreden geçirilerek düzgün bir analog sinyal elde edilir.

DEĞERLENDİRME

Cevaplarınızı cevap anahtarı ile karşılaştırınız. Doğru cevap sayınızı belirleyerek kendinizi değerlendiriniz. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt yaşadığınız sorularla ilgili konuları faaliyete dönerek tekrar inceleyiniz.

ÖĞRENME FAALİYETİ-2

AMAÇ

FIR ve IIR filtre uygulamasını istenen değerlere uygun olarak yapabileceksiniz.

ARAŞTIRMA

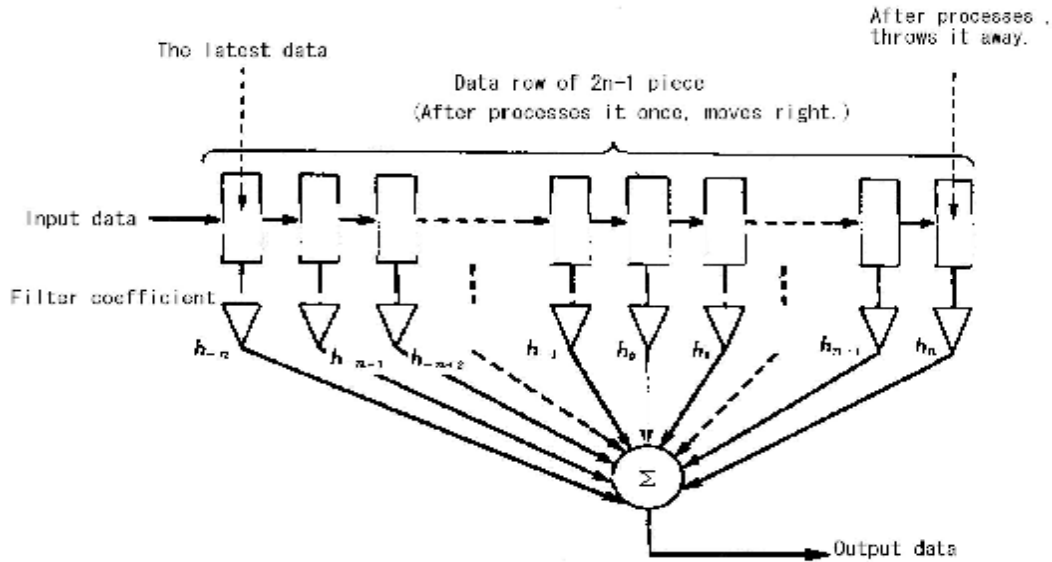
Ø Sayısal filtreler hakkında bir araştırma yaparak rapor haline getiriniz.

2. FIR VE IIR FİLTRE NEDİR?

2.1. FIR tip filtre

2.1.1. FIR Tip Filtrenin Oluşumu

Şekil 2.1’de FIR tipi filtrenin oluşumu gösterilmektedir. D/A konvertörün örnekleme zamanına bağlı olan verileri ve A/D konvertörün önceki ve dosyada depolanan giriş verilerinin bilgileri kabul edilir. Ne zaman en son veri şeklin en solundan girerse bu bir veri işlenir ve bir veri sağa doğru kaydırılır. Ve en son veri tekrar en son tarafta giriş olur. Filtre katsayıları (h_n) sırasıyla hazırlanarak yerleştirilir, her bir veri karşılaştırılır, sonuçlar toplanır ve filtre çıkışında gösterilir. Bu zamanda, çeşitli filtrelerin karakteristiklerine, filtre katsayılarının değerlerine ve sayılarına bağlı olarak serbestçe karar verilebilir.



Şekil 2.1: FIR tip filtrenin temel bileşenleri

Burada her bir verinin işlenmesinden ve biriktirilmesinden önce filtre katsayısı yerleştirilmiştir. Ayrıca, seviye işlemleri ve birikme işlemleri, filtre katsayılarının hepsinin aynı olduğu zaman 2 bölümün zamanında 3'ün 4 bölümü, 0.5 ve 0.33 süresindeki bir sözcüğün 0.25 olduğu kabul edildiği aynı sonuçlar olur. FIR tip filtrede, her bir filtre katsayısı yüksek geçiren filtre ve bant geçiren filtre için bunu yapmak veya bir aktif filtre yapıp yapmamak için yalnız uygun seçim yapar.

2.2. IIR tip filtre

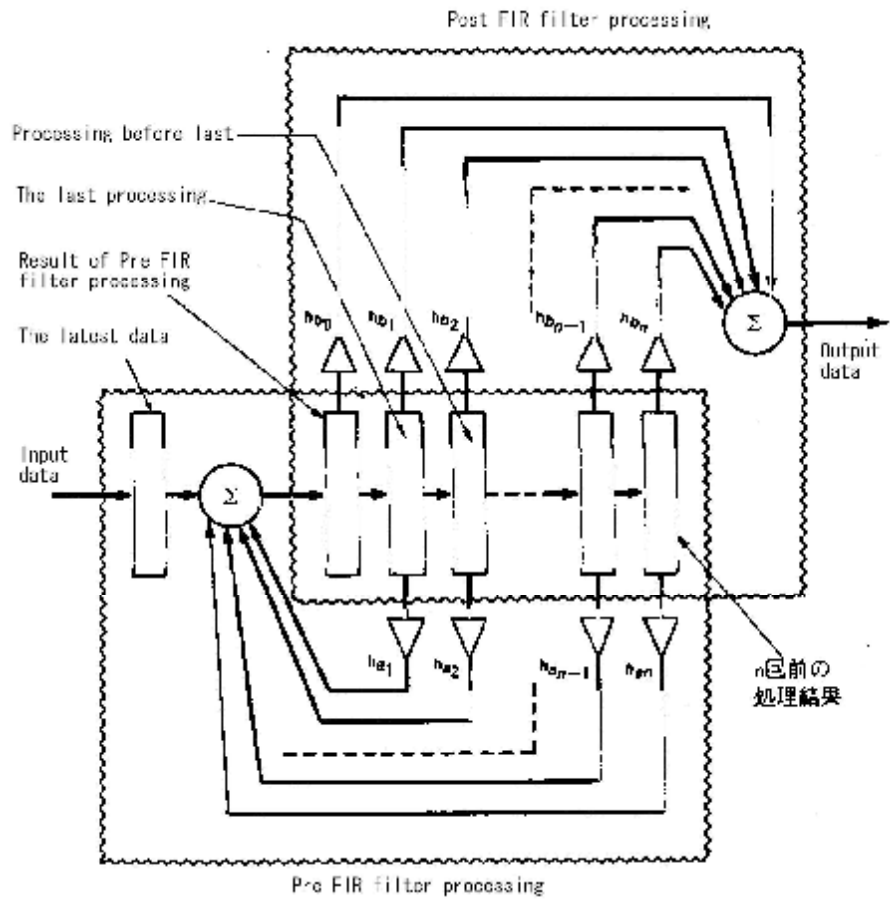
Daha önce IIR tip filtreden ve sayısal filtre çıkışı olduğundan bahsetmiştik. IIR tip filtrede, filtre katsayılarına analog filtre devresinin geçiş donanımının dizayn metodu olarak benzer tekniklere bağlı olarak karar verilir. Birkaç çeşit analog filtre devre yapıları vardır. Bu filtrelerin her bir özelliğini en iyi şekilde ve tam anlamıyla kullanmak gerekir. Burada, filtre katsayıları kullanılarak yapılan kolay bir uygulama sonuçları şekillerle gösterilerek, yalnız bir analog filtrenin karakteristiklerinden elde edilen filtre katsayıları hakkında bilgi verilecektir. Yalnız filtre katsayılarının değiştirilmesiyle çeşitli filtre karakteristiklerinin elde edilebileceğini öğrenebiliriz.

2.2.1. IIR Tip Filtrenin Yapısı

Şekil 2.2'de IIR tip filtrenin bileşenleri gösterilmektedir. Sırayla iki tane uzun mesafeli infrared redrasyon tip filtre bağlanmıştır. Filtre katsayılarının değerleri (h_{a1}, h_{an}), ilk öndeki uzun mesafeli infrared redrasyon tip filtrede en son giriş verileri koyularak toplanır ve bir önceki işlem sonuçları hafızaya alınır. Sonuçlar ilk öndeki uzun mesafeli infrared redrasyon tip filtrenin bir güç çıkışı olarak hafızada tutulur. Son bölümdeki uzun mesafeli infrared redrasyon tip filtrede bir önceki işlem sonuçları işlenerek toplanır. Hazırlanan filtre katsayıları ($h_{b0}-h_{bn}$) kullanılır. Uzun mesafeli infrared redrasyon tip filtreden dolayı bir sonraki adım bir önceki adımın yanındadır. Şekil 2.3'te yukarıdaki anlatılan işlemleri gerçekleyen program gösterilmektedir. Şekil 2.3'te, Butter Worth tip alçak geçiren filtrenin filtre katsayılarının kullanılması durumu açıklanarak gösterilmiştir.

İlk öndeki uzun mesafeli infrared redrasyon tip filtrede işlem sonuçlarının kullanılması ve son bölümdeki uzun mesafeli infrared redrasyon tip filtrede bir sonraki nesnenin işlenmesi IIR filtrenin özelliğidir. Burada işleme tabi tutulan veri sonuçlarından biri anlaşılabilir, çünkü ilk öndeki uzun mesafeli infrared redrasyon tip filtrenin her bir filtre katsayısı yerine koyulur ve şeklin solundaki ve arkasındaki veri sonuçlarının akışı Şekil 2.2'deki nesnenin bir işlemi olarak tekrar kullanılır.

Daha önce işlenmiş bir veri sonucunun kaydırılması ve bu verilerin tekrar bir sonraki işlemde kullanılması işlemleri geri besleme olarak bilinir. Bu geri besleme bulunan mekanizmada sayısal filtreleme işleminin 2 dakikada yapılması ile uzun mesafeli infrared redrasyon tip filtrenin ve IIR tip filtrenin bileşenlerinde büyük bir fark vardır.



Şekil 2.2: IIR tip filtrenin yapısı

```

#include <stdio.h>
#include <stdlib.h>
#include <alloc.h>

main()
{
    double sum, *d;
    float dd;
    int hn=5, dn, i, j;

    double sum, *d, dd;
    int hn, dn, i, j;

    double ha[5]={ 1.0000, 0.0000,
                  -0.4860, 0.0000, -0.0177 };
    double hb[5]={ 0.0940, 0.3759,
                  0.5639, 0.3759, 0.0940 };

    scanf( "%d", &dn );
    d = malloc( sizeof(double)*dn );
    for( i=0; i<dn; i++ ) d[i] = 0.0;
    printf( "%d\n", dn );

    for( i=0; i<dn; i++ )
    {
        for( j=hn-2; j>=0; j-- ) d[j+1] = d[j];
        scanf( "%f", &dd );

        sum = dd;
        for( j=1; j<=hn; j++ ) sum += d[j]*ha[j];
        d[0] = sum;

        sum = 0.0;
        for( j=0; j<=hn; j++ ) sum += d[j]*hb[j];
        printf( "%e\n", sum );
    }

    free( d );
}

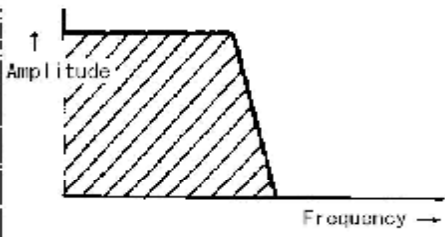
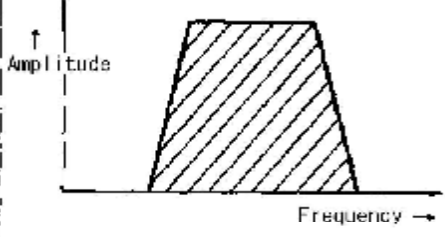
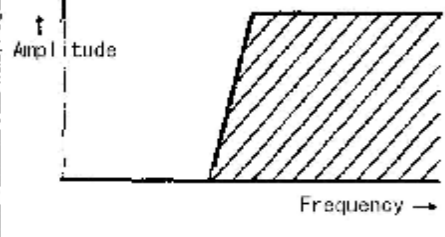
```

Şekil 2.3: IIR tip filtre programı (IIR.C)

2.3. FIR ve IIR Filtrenin Kullanım Alanları Nelerdir

2.3.1. Filtre çeşitleri

Çeşitli sinyaller arasından yalnız istenilen sinyali süzerek veren elektronik devrelere filtre denir. Bir kahvenin telvesini süzen kâğıt filtre veya delikli süzgeç gibi filtreler istenilen frekanstaki sinyalleri geçirir istenmeyen frekanstaki sinyalleri geçirmez. Dijital filtreler sayısal veri sinyali üretir ve bu sinyalleri temel matematik işlemlerine dayalı donanımla birlikte programda işler.

Characteristics	Name of filter	
The shaded portion passes.	For the signal that passes	For the removed signal
	Low-pass filter	High-cut filter
	Band-pass filter	Band elimination filter
	High-pass filter	Low-cut filter

Tablo 2.1: Çeşitli filtreleri ve isimlerini gösterir.

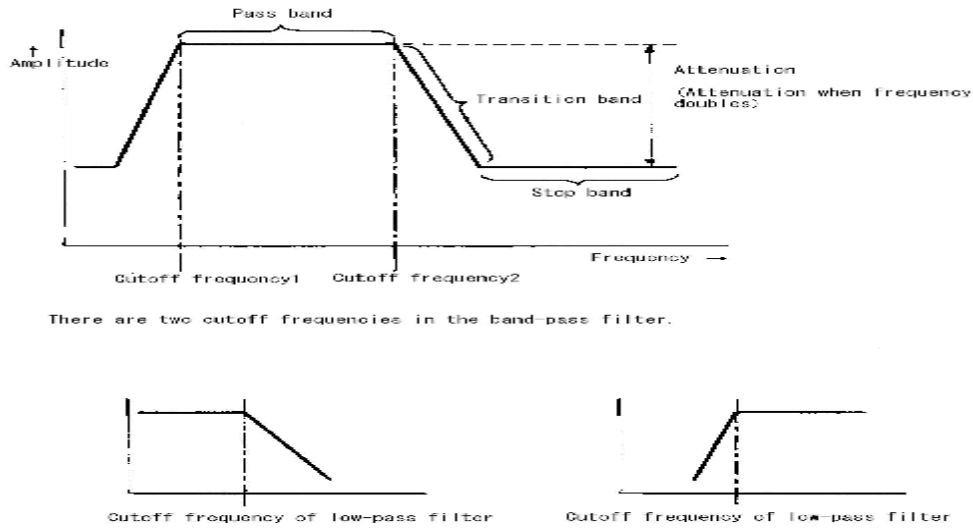
Filtreler çeşitli karışık sinyaller arasından "yalnız gerekli bir sinyalin geçmesi" veya "yalnız gereksiz bir sinyalin geçmemesi" işlemlerinde kullanılır. Yani alçak frekanslı sinyallerin, yüksek frekanslı sinyallerin veya istenilen bant aralığındaki sinyallerin geçirilmesinde filtreler kullanılır.

Genelde, filtrelerin nominal büyüklükteki benzer sinyalleri elde etmek için geçirdiği

görülür. Bir dijital sinyal işlemlerini yapmak için donanım ihtiyacı duyulduğu zaman birçok alçak geçiren filtreler görülür ve Anti-Alias filtre gibi filtrelerin önemli bir rolünün yükü alınır. Ayrıca bu Tablo 2.1’de gösterilen üç çeşit filtrede dijital filtre işlemcisi dijital sinyal işlemleri olarak açıklanır.

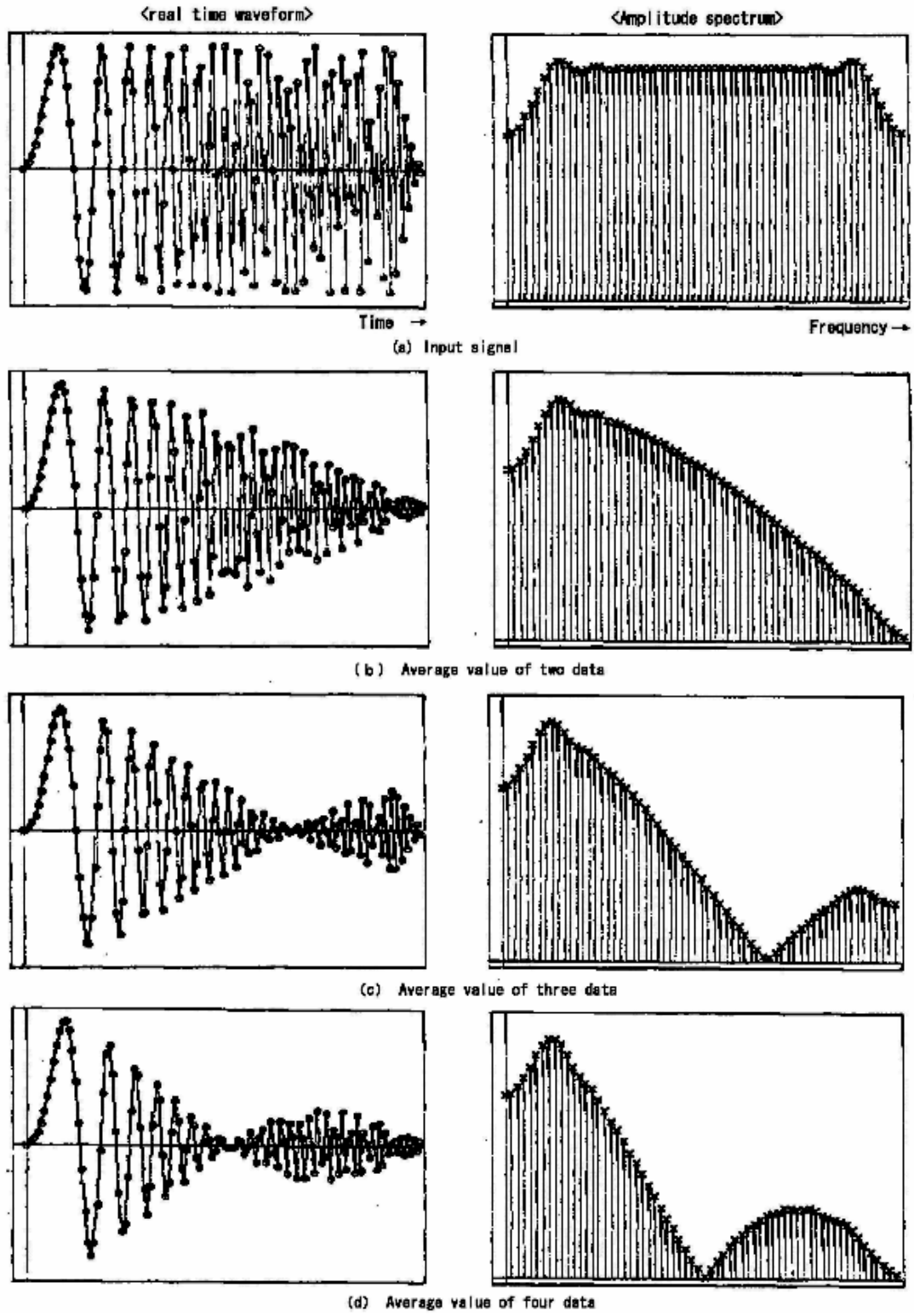
2.3.2. Filtrelerin genlik karakteristikleri

Filtre dizaynı ve karakteristikleri hakkındaki açıklamaların daha iyi anlaşılması için, filtrelerde ihtiyaç duyulan genlik karakteristiklerinin her bir bölümünün isimleri aşağıdaki Şekil 2.4’te gösterilmiştir. Geçiş bandının (pass band) frekans aralığında zayıflatılmış frekansların dışındaki birbirine benzer sinyaller geçirilir. Zayıflatılmış frekans aralığı ise kesim bandı (stop band) olarak bilinir. Kesim bandı (stop band) ile geçiş bandı (pass band) arasındaki banda geçiş bandı (Transition band) denir. Genelde, bu iyi bir filtredir, çünkü bu filtrede istenmeyen sinyaller zayıflatılarak geçmesi engellenir ve istenen sinyaller yalnız olarak ayrılabilir.



Şekil 2.4: Filtrelerin genlik karakteristiklerinin her bir bölümünün isimleri

Geçiş bandının başlangıç ve bitiş frekanslarına kesim frekansı "Cutoff frequency" denir. Yukarıdaki şekilde de görüldüğü gibi iki tane kesim frekansı vardır. Bunlardan biri geçiş bandının başlangıç noktasındaki f_1 kesim frekansı "Cutoff frequency1" diğeri ise geçiş bandının bitip transition (geçiş) bandının başladığı noktadaki f_2 kesim frekansıdır "Cutoff frequency2". Bu iki kesim frekansı geçiş bandı filtresinde (band-pass filter) ve sırasıyla birer birer alçak geçiren filtre (low-pass filter) ve yüksek geçiren filtrede (high-pass filter) yer alır.



Şekil 2.5: Bir alçak geçiren filtrede (low-pass filter) seviyelendirme işlemleri

Şekil 2.5 (a)' da kullanılan giriş sinyalinin gerçek zaman dalga formu değişen zaman ekseninde, genlik spektrumu da frekans ekseninde gösterilmiştir ve buna "Chirp signal" denir. Ayrıca, frekans aksının her iki son ucunda bu sinyallerin genlik spektrumları zayıfladığı halde orta bölmede düzgün bir hale gelir. Bu durum filtre için eklenen sinyallerin filtre karakteristiği olarak bilinir ve geçirilen çıkış sinyallerinin genlik spektrumları görünür. Çeşitli filtre karakteristikleri, "Chirp signal" kullanılmasıyla şekildeki metod tarafından değerlendirilir.

Giriş sinyalinin verileri konusunda, (Biri ile 1 tane yenisinin yenilediği zaman), Şekil 2.5 (b)' de iki parçanın ortalaması alınmıştır. Genlik spektrumu, "Chirp signal" işlemlerinin sonunda alçak geçiren filtrede yavaş yavaş gelir. Çünkü Tablo 2.1'deki bölmelere dikkat ederseniz en sağdaki en zayıflatılmıştır.

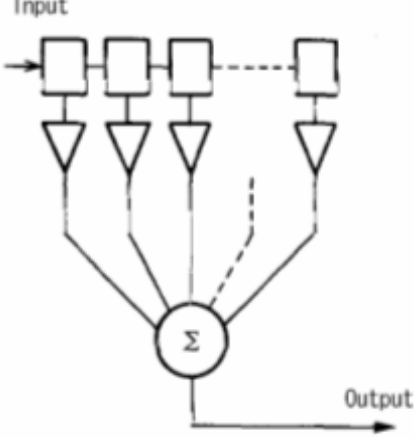
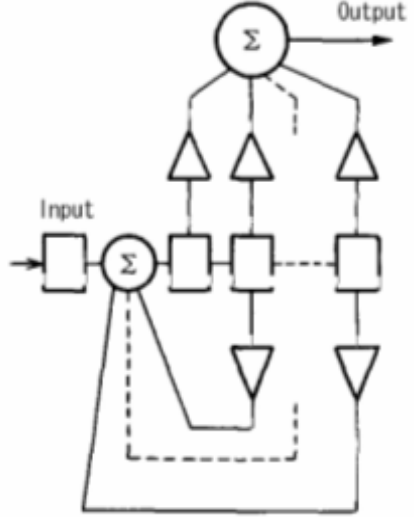
Ayrıca, Şekil (c) ve (d)'de 3 ve 4 verinin seviyelendirilmesiyle oluşan genlik spektrumu gösterilmektedir. Böylece filtre karakteristiği gösterilen transition (geçiş) bandındaki eğilim, artışın zayıflaması ve veri sayılarının seviyeleri hakkında iyi veya kötü olan artışın aniden gelmesi daha iyi anlaşılır. Ancak, pratikte alçak geçiren filtre olarak kullanıldığında bu olmaz çünkü karakteristikte bu azalmadan hemen sonar stop bandı tekrar gelir ve bu gözükür. Bu gerçek zaman dalga formunun filtreden geçirildikten sonra doğrulanabilir.

Bu şekilde, örneğimizdeki bazı giriş sinyallerini kullanarak uygun giriş sinyali ve görünümü işlemleri için filtre işlemlerinin verdiği eşit bir sonuç elde edilmesi daha iyi anlaşılır. Bu sayısal filtrenin bu sürede nasıl en uygun işlemi yapacağı hakkında bir düşüncedir.

2.3.3. FIR ve IIR Filtre Nasıl Gerçekleştirilir

Geri besleme devresiz FIR tip filtre ve geri besleme devreli IIR tip filtre Tablo 1.5'te iki ayrı bölmede gösterilmiştir.

FIR (Finite Impulse Response) tip filtre sadece bir sayısal sinyal işlemi yapan bir filtredir. Çünkü işlemleri sürekli, kolayca kullanılması mümkündür. Ancak, elde edilmesi beklenen azaltılmış karakteristik için, bir çok sayıda aşamalı olarak filtre katsayılarına ihtiyaç duyulur. O nedenle, programda uygulama süresi uzun sürer ve kullanılan işlem takımı tarafından ortalananır.

Kind of filter	The basic component	Feature
FIR filter		<ul style="list-style-type: none"> ○ The operation is steady. ○ The phase property straight line. ✗ A lot of filter coefficients are necessary for obtaining with a sharp characteristics.
IIR filter		<ul style="list-style-type: none"> ○ The design methodology of the analog filter can be used. ○ A sharp characteristics is comparatively obtained by few filter coefficients. ✗ When it makes a mistake in the design, it is an unstable operation.

Tablo 2.2: Sayısal filtrenin özellikleri ve oluşumu

Diğer IIR (Infinite Impulse Response) tip filtre, bir elektronik devre tarafından sayısal sinyal işlemleri ile değiştirilen analog bir filtredir. Bu filtrenin özelliği birkaç filtre katsayısının aşamalı değerleri ile karşılaştırma yaparak büyük bir zayıflama karakteristiği elde etmesidir. Gerçek zaman işlemleri kullanılarak uygulama süresini kısaltmak mümkündür.

Sayısal filtrelerde filtre katsayısının aşamalı değerleri artırılır ve azaltılır. Çünkü bu sadece programdan meydana gelmektedir. Filtrenin azalan karakteristiği iptal edilebilir ve bu çeşitli filtrelerle filtre katsayılarının değiştirilmesiyle mümkün olur. Birçok değer vardır. Örneğin lehimleme işi, yalnız bir elektronik devrenin oluşumu zamanında belirli bir yere kadar bir aktif filtre ile karşılaştırma yapmak gereksiz olmaz ayrıca, hesaplama gibi benzer işleri de kolayca yerine getirir.

Fakat, sayısal filtrenin tek problemi başka bir el üzerindeki programın işlem hızıdır. IIR tip filtre ise, kullanışlı bir zayıflama karakteristiği göreceli olarak uzun mesafeli infrared radyasyon tip filtre ile birkaç filtre katsayısının aşamalı değerleri ile karşılaştırma yaparak elde edilir. Bununla birlikte, ses dalgasının bandı (20Hz-20kHz) kişisel bilgisayar işlemleri ile işlenebilen yüksek ses frekans alanının üst seviyesini işlemek oldukça zordur. Eğer sayısal sinyal işlemleri özel işlemcisi uyum sağlasa bile elektronik devrenin oluşumundaki filtrenin işlem oranı buna yetişemez.

2.3.4. Alçak-geçiren filtre, filtre katsayıları ve karakteristikleri

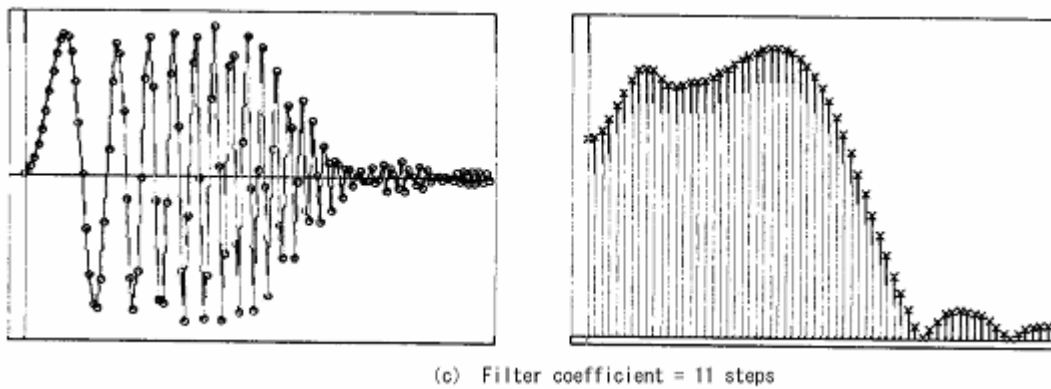
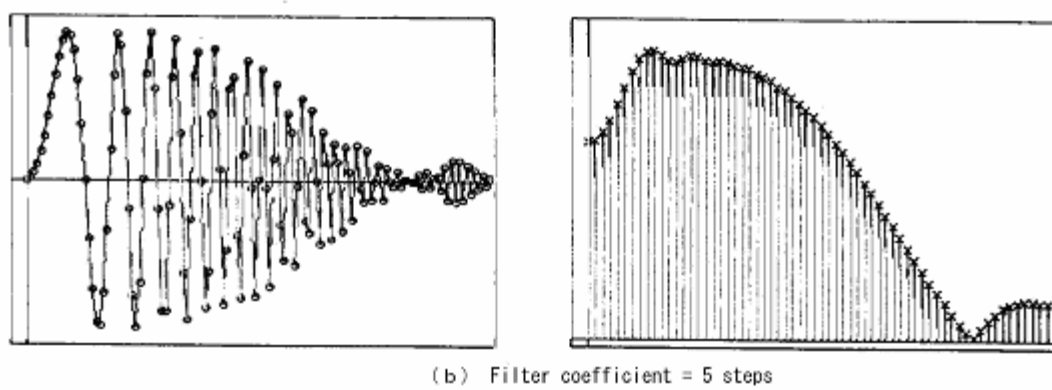
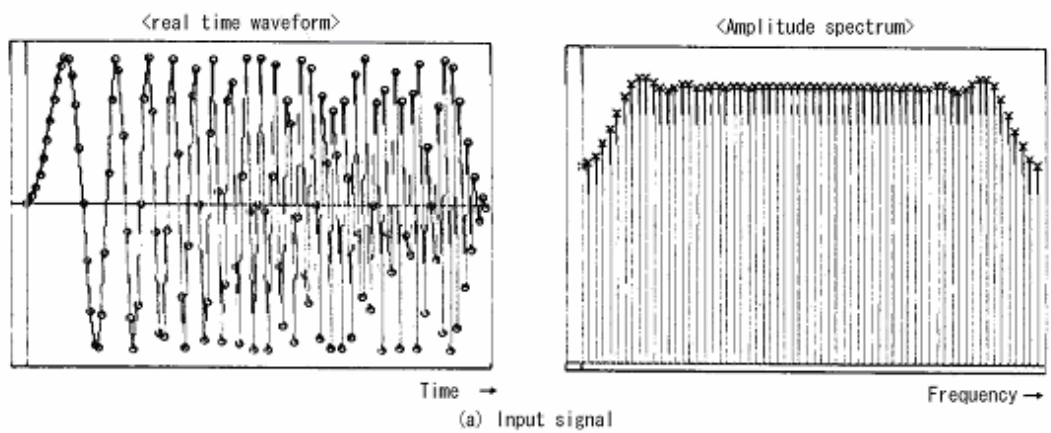
Filtre katsayılarının nasıl isteneceği hususunda bazı metodlar vardır. Burada Fourier serisinin basit bir yuvarlamayla alçak geçiren filtrenin katsayılarının örnekleri Tablo 2.3'te gösterilmektedir. Ve bu katsayılar kullanılarak Chirp sinyal işleminin örnekleri Şekil 2,6'da gösterilmektedir. Filtre katsayılarının sayıları karşılaştırılır ve üç çeşit (5 adım, 11 adım ve 21 adım) kabul edildiği zaman karakteristikle karşılaştırılır.

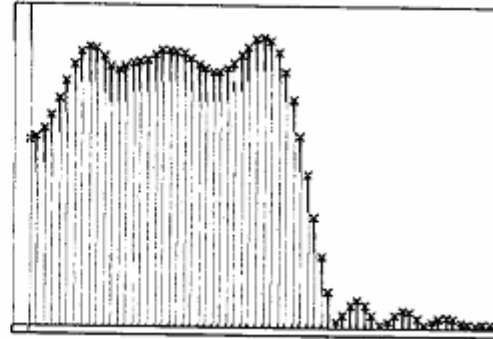
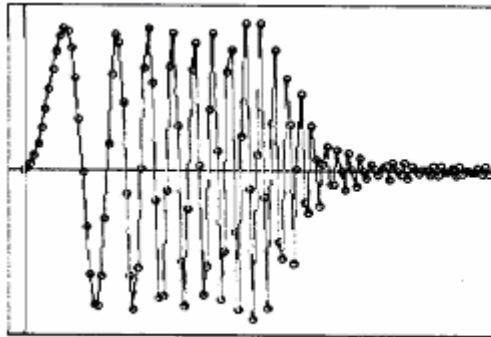
Şekil 2,5(a)'nın giriş sinyali bir Chirp sinyalidir. Her bir filtrenin karakteristiği, her bir filtrenin geçirdiği ve genlik spektrumundan sonraki dalga şekli ve genlik spektrumları dalgalarının şekline bakılarak yapılan karşılaştırmayla doğrulanır.

Şekil 2,5(b) filtre katsayıları konusunda beş adım ile ($n=-2,-1,0,+1,+2$) kullanılan bir alçak geçiren filtrenin karakteristiğidir. Sağdaki yarım ise, bir çeşit filtre olan alçak geçiren filtredir. Çünkü dalga şeklinde de ve genlik spektrumunda da azalma olduğu halde giriş sinyalinin sol yarısı olarak aynıdır.

n	Coefficient
0	6.000000e-01
± 1	3.027307e-01
± 2	-9.354893e-02
± 3	-6.236595e-02
± 4	7.568267e-02
± 5	2.338826e-17
± 6	-5.045512e-02
± 7	2.672826e-02
± 8	2.338723e-02
± 9	-3.363675e-02
± 10	-2.338826e-17

Tablo 2.3: Uzun mesafeli infrared redrasyon tip alçak geçiren filtrenin filtre katsayıları örnekleri

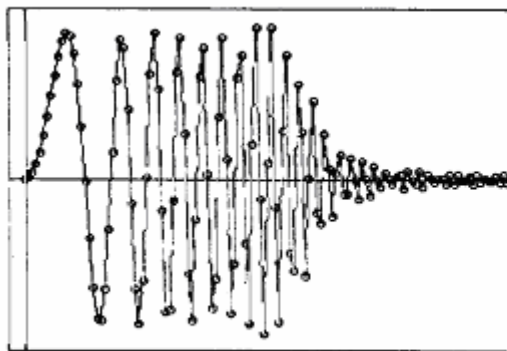




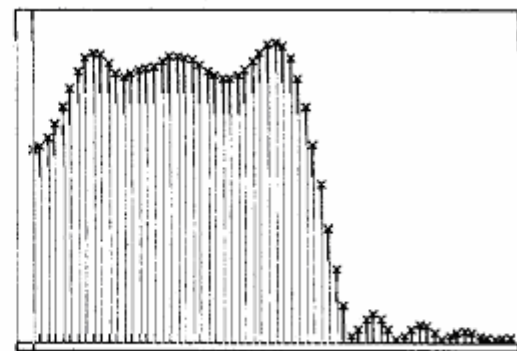
(d) Filter coefficient = 21 steps

<real time waveform>

<Amplitude spectrum>

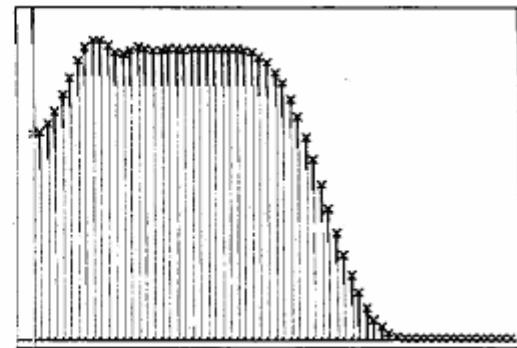
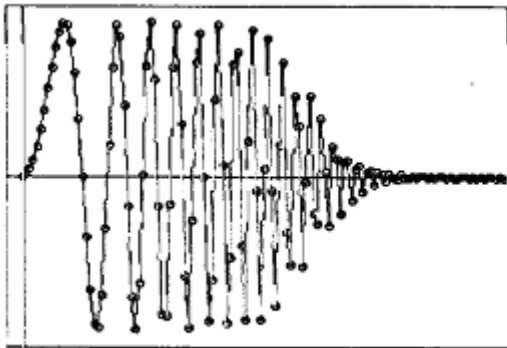


Time →

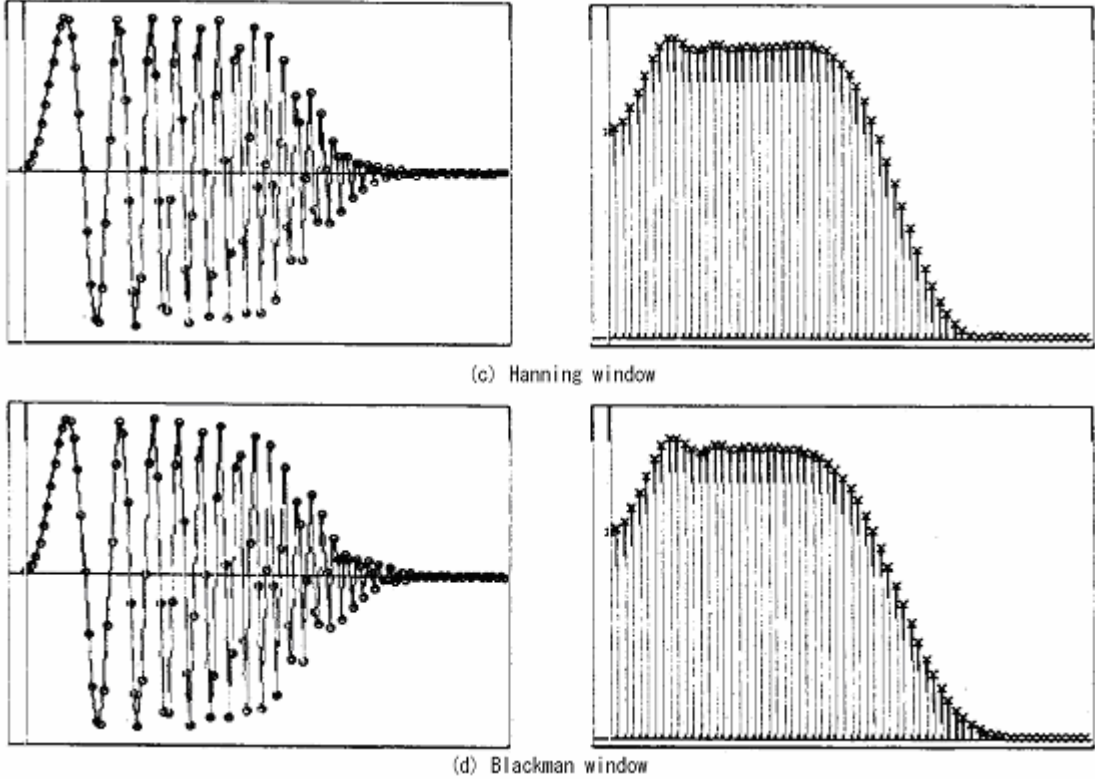


Frequency →

(a) Square-wave window



(b) Hamming window



Şekil 2.6: Tablo 2.3'teki filtre katsayılarını kullanan alçak geçiren filtrenin karakteristiği

Transition (geçiş) bandının eğiminin büyümesi ani olur ve 11 adım ve 21 adım filtre katsayısının aşamalı değerleri ile filtre düzeltilmesi olarak anlaşılması işlemi gözükür. Fakat geçiş bandının engebелiliği, filtre katsayısı artışının aşamalı değerleri olarak bir tarafta artar. Programın kullandığı bu zaman Şekil 2.6'da gösterilmiştir. Burada, $n=0$ dışındaki filtre katsayıları simetriklerdir. Aşağıdaki programda çerçeve içindeki sayılara dikkat ederseniz $n=-1$ ve $n=+1$ için filtre katsayısı aynıdır. Yani 11 adım filtre katsayısı için $n=0$ 'dan $+5$ değerlerinin Tablo 2.3'ten alınarak kullanıldığı görülmektedir. Aynı şekilde, eğer n 'nin değeri 21'e değişirse $n=0$ için 21 filtre katsayısı kullanan programı değiştirmek mümkündür.

```

#include      <stdio.h>

main()
{
    int      num, n=11, i, k;
    double sum, d[11]={ 0.0, 0.0, 0.0, 0.0, 0.0,
                        0.0, 0.0, 0.0, 0.0, 0.0, 0.0 };
    double h[11]={
        2.338826e-17, /* -5 */
        7.568267e-02, /* -4 */
        -6.236595e-02, /* -3 */
        -9.354893e-02, /* -2 */
        3.027307e-01, /* -1 */
        6.000000e-01, /* 0 */
        3.027307e-01, /* 1 */
        9.354893e-02, /* 2 */
        -6.236595e-02, /* 3 */
        7.568267e-02, /* 4 */
        2.338826e-17 /* 5 */
    };

    scanf( "%d", &num );
    printf( "%d\\n", num );

    for( k=0; k<num+n/2; k++ )
    {
        for( i=n-2; i>=0; i-- ) d[i+1] = d[i];
        if( k<num ) scanf( "%lf", &d[0] );
        else d[0]=0.0;

        sum = 0.0;
        for( i=0; i<n; i++ ) sum += d[i]*h[i];

        if( k>=0 ) printf( "%lf\\n", sum );
    }
}

```

filter coefficient =11 steps

One data is moved right.

The latest data is input.

The filter coefficient and data are multiplier and it accumulates.

Şekil 2.7: FIR tip program (adım adım 11 filtre katsayısı için.)

Şekil 2.5'te gösterildiği gibi geçiş bandı pürüzlüdür. Eğer birçok filtre katsayısı kullanılırsa transition (geçiş) bandın eğimi hızlandırılır. Fourier serisinde istenilen filtre katsayıları kullanılır ve sonuç basit bir yuvarlama yapılarak alınır. Buna Gibb's olayı denir. Bu olaydaki indirgeme, istenilen filtre katsayıları için yalnız pencere fonksiyonu uygulamasına sahiptir. Daha önce Tablo 2.3'te filtre katsayıları verilen alçak geçiren filtrenin üç çeşit pencere fonksiyonunun Hamming Window, Panning Window ve Blackman Window için filtre katsayıları Tablo 2.4'te verilmiştir. Somut bir yöntem olarak, Tablo 2.3'teki filtre katsayıları düzenli olarak $n=-10-0-+-10$ için 21 tane verinin toplanmasıyla yapılır. Daha

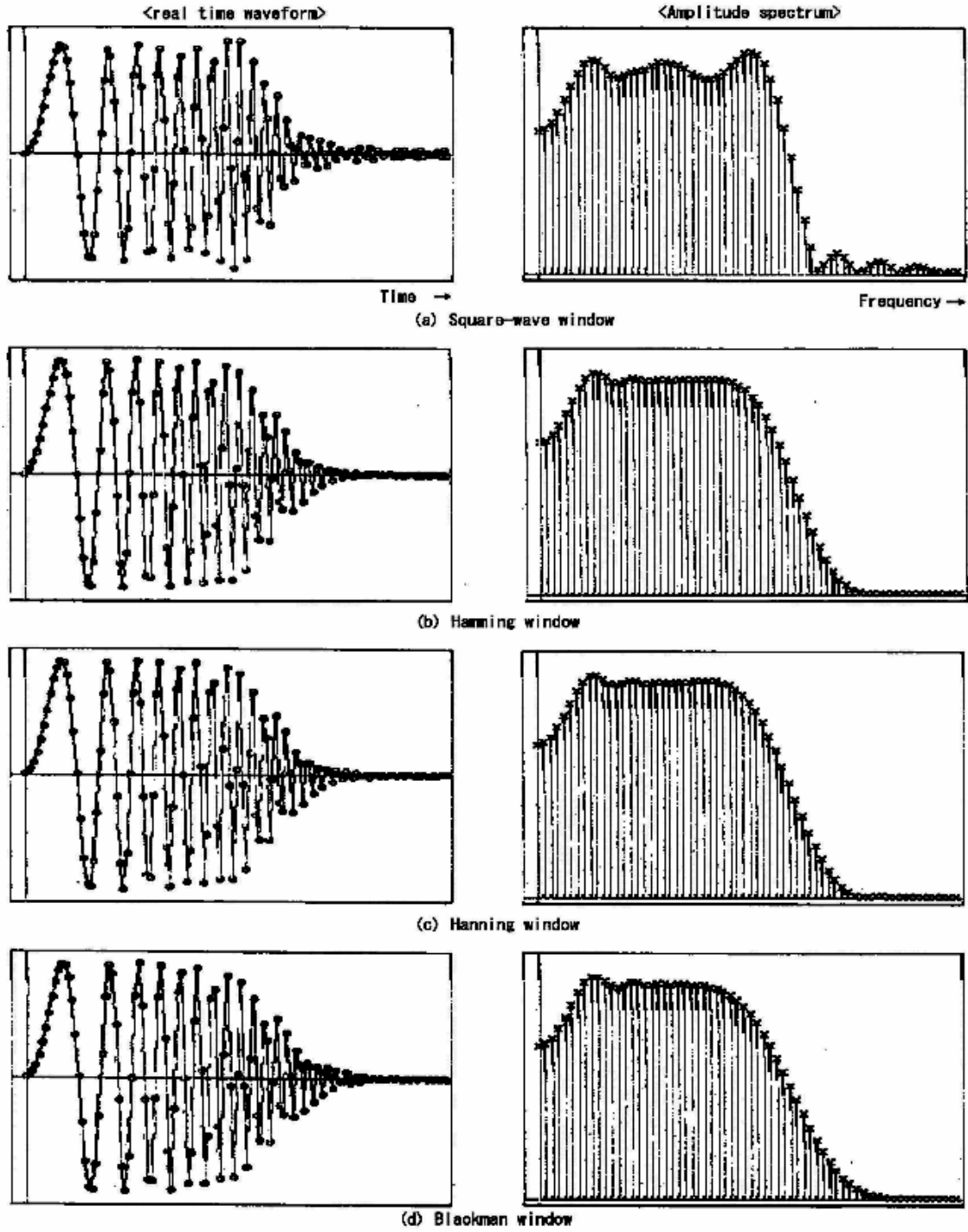
sonra, her bir pencere fonksiyonu programı için kullanılan I/O (giriş/çıkış) nun yeni adrese gönderilmesi işlemleri yapılır. Şekil 2.6'daki programda, Tablo 2.4'teki karşılıklı aynı olan filtre katsayılarının bir diğeri ile yenilenmesiyle filtre karakteristiği değiştirilir. Şekil 2.7'de Tablo 2.3'te verilen filtre katsayıları kullanılarak bu pencere fonksiyonu işlemleri sonucu oluşan alçak geçiren filtre karakteristiklerini göstermektedir.

Şekil 2.7(a)'da bu filtre katsayıları kullanılarak transition (geçiş) bandının eğiminin hızlandırıldığı bir karakteristik görülmektedir. Şekil 2.5(d)'de 21 adım filtre katsayısı kullanılmıştır. Şekil 2.7 (b) - (d)' deki her bir karakteristik her bir pencere fonksiyonunun işlemlerinin bir sonucudur. Çerçeve içindeki şekillerin hepsi birbirinin benzeri görünmesine rağmen transition (geçiş) bandının eğiminde ufak farklılıklar vardır. Fakat, bunu resimden fark etmek zordur.

Fakat böylece, yalnız geçiş bandında değil ayrıca stop bandında da pencere fonksiyonu uygulanmasıyla karakteristiğin fazlasıyla azalışındaki düzensizlik daha iyi anlaşılır.

n	Filter coefficient for low-pass filter			
	Square-wave window	Hanning window	Hanning window	Blackman window
0	6.000000e-01	6.000000e-01	6.000000e-01	6.000000e-01
± 1	3.027307e-01	2.959150e-01	2.953224e-01	2.906970e-01
± 2	-9.354893e-02	-8.533045e-02	-8.461580e-02	-7.944454e-02
± 3	-6.236595e-02	-5.054020e-02	-4.951187e-02	-4.298082e-02
± 4	7.568267e-02	5.162677e-02	4.953495e-02	3.858205e-02
± 5	2.338826e-17	1.262966e-17	1.169413e-17	7.952008e-18
± 6	-5.045512e-02	-2.007368e-02	-1.743181e-02	-1.012988e-02
± 7	2.672626e-02	7.206442e-03	5.508893e-03	2.709872e-03
± 8	2.338723e-02	3.925598e-03	2.233282e-03	9.404675e-04
± 9	-3.363675e-02	-3.448237e-03	-8.231497e-04	-3.092260e-04
± 10	-2.338826e-17	-1.871061e-18	-0.000000e+00	3.245773e-34

Tablo 2.4: Pencere fonksiyonunda kullanılan alçak geçiren filtrenin filtre katsayıları (kesim frekansı 0.2 normal değeri için)



Şekil 2.8: Tablo 2.3'teki filtre katsayılarını kullanan alçak geçiren filtrenin karakteristiği

2.3.5. Kesim frekansına nasıl karar verilir

Bir filtre karakteristiğinde geçiş bandının bitip, transition (geçiş) bandının ve zayıflamanın başladığı sınırdaki frekans, kesim frekansı olarak bilinir. Burada, sayısal filtrede kesim frekansına nasıl karar verileceğini açıklayacağız.

Sayısal filtrelerde, kesim frekansı frekans örnekleme ile yakından ilgilidir. Frekans örneklemlerine bağlı olan giriş verileri belirli gerçek zamandaki giriş verilerinin dosyasından ve belirli uygulamayla dahi elde edilir. Sayısal filtre işlemlerinin toplama işlemlerinin sonuçları filtre katsayılarının yerine koyularak temel örnekleme verilerinin geçirilmesi vb. dir.

Bu yüzden, sayısal filtrelerin kesim frekansı, giriş verisinin örnekleme frekansının bir nesne olarak işleme tabi tutulduğu ve **1** olduğu kabul edilirse **oran** (= kesim frekansı / örnekleme frekansı) olarak belirlenir. "**Normal hale getirilmiş**" olarak ifade edilen bu kesin değer orana göre 1 olur.

Bundan başka, bu oran 0.5 veya daha da azaldığı görülür. Bunun nedeni örnekleme teoreminin donanım tarafından anlaşılması yüzündendir ve Anti-Alias-filtrenin kesim frekansı örnekleme frekansının yarım altındadır.

Şekil 2.8’de 5 çeşit kesim frekansı (normalleştirilmiş kesim frekansları = 0.5, 0.4, 0.3, 0.2, ve 0.1) kullanılan filtre karakteristiklerinde kesim frekansının değişimine dikkat edin. Şekil 2.6’daki programın sadece filtre katsayıları bölümünü Şekil 2.8’deki karakteristiklerin her bir durumu için Tablo 2.5’teki bir duruma ait değerlerle değiştirerek bu karakteristikleri görmeyi mümkün olacaktır.

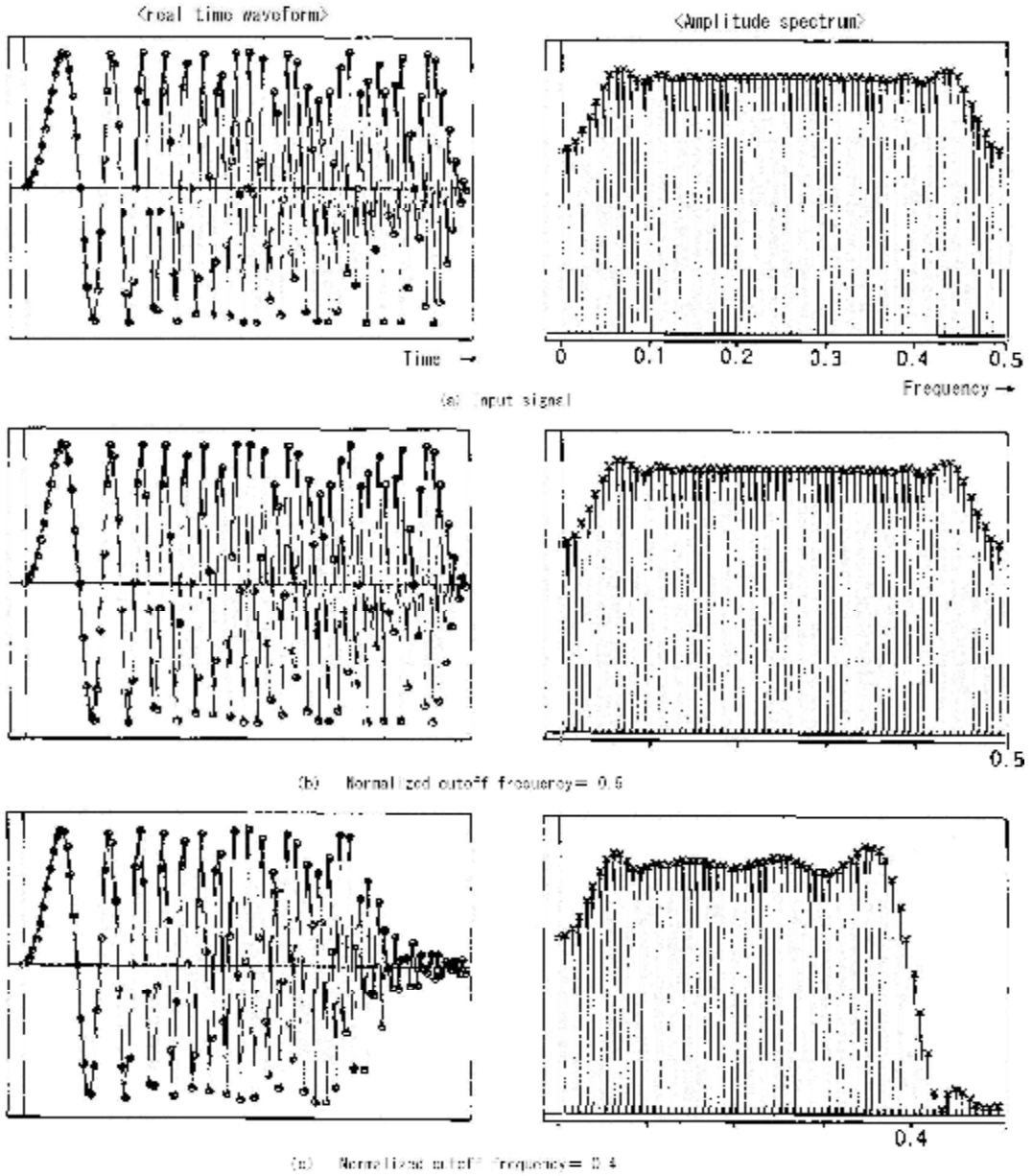
n	Normalized cutoff frequency				
	0.5	0.4	0.3	0.2	0.1
0	1.000000e+00	8.000000e-01	6.000000e-01	4.000000e-01	2.000000e-01
± 1	3.898043e-17	1.870979e-01	3.027307e-01	3.027307e-01	1.870979e-01
± 2	-3.898043e-17	-1.513653e-01	-9.354893e-02	-9.354893e-02	-1.513653e-01
± 3	3.898043e-17	1.009102e-01	-6.236595e-02	-6.236595e-02	1.009102e-01
± 4	-3.898043e-17	-4.677447e-02	7.568267e-02	-7.568267e-02	4.677447e-02
± 5	3.898043e-17	-3.118435e-17	2.338826e-17	-1.559217e-17	7.796086e-18
± 6	-3.898043e-17	3.118298e-02	-5.045512e-02	5.045512e-02	-3.118298e-02
± 7	3.898043e-17	-4.324724e-02	2.672826e-02	-2.672826e-02	4.324724e-02
± 8	-3.898043e-17	3.784133e-02	-2.338723e-02	2.338723e-02	-3.784133e-02
± 9	3.898043e-17	-2.078865e-02	-3.363675e-02	-3.363675e-02	-2.078865e-02
± 10	-3.898043e-17	-3.118435e-17	-2.338826e-17	-1.559217e-17	-7.796086e-18

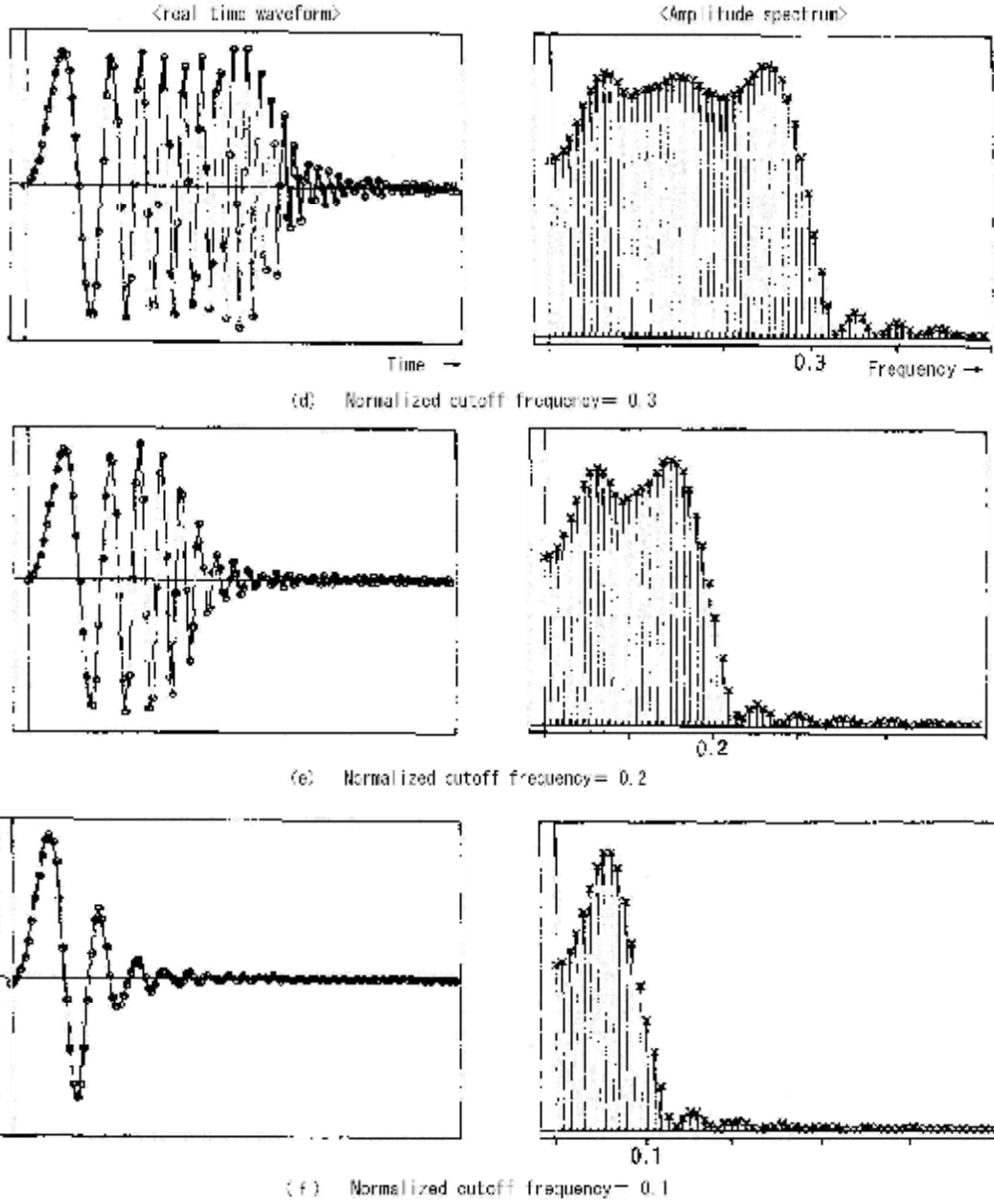
$$\text{Normalized cutoff frequency} = \frac{\text{Cutoff frequency}}{\text{Sampling frequency}}$$

Tablo 2.5: Normalleştirilmiş farklı kesim frekansları için filtre katsayıları örnekleri

Burada, en yüksek frekansın bir sađ köşesi frekans ekseninde gösterilen genlik spektrumunun çeyrek eksenini olsa dahi bu örnekleme frekansının bir yarısıdır. Bu yüzden, her bir kesim frekansı (zayıflamanın başladığı frekans) bu çeyrek eksenin 5 eşit parçaya bölünmesiyle elde edilen frekans olacaktır.

Şekil 2.8(b)'de örnekleme frekansının yarısı kadar olan kesim frekansının 0.5 olduğu karakteristik gösterilmiştir. Aktif frekans bileşenleri örnekleme frekansının yarısının altındaki örnekleme verilerine dahil edilir. Bu yüzden, örnekleme frekansının yarısı (0.5) için alçak geçiren filtrenin kesim frekansının belirtilmesi anlamsızdır. Çünkü burada bütün veriler geçiriliyor. Bu Şekil 2.8(a)'nın giriş sinyali olarak tamamen aynıdır ve diğer sonuçlar deneylerden gelen bir etki değildir.





Şekil 2.9: Tablo 2.5'teki filtre katsayılarının kullanılmasıyla kesim frekansının değişimi

Şekil 2.8.(c)'den (f)'ye kadarki karakteristiklerde yüksek frekansın her bir kesim frekansında genlik spektrumundaki hızlı azalış daha iyi anlaşılmaktadır. Bundan başka, band geçiren filtrenin karakteristiğindeki düzensizliği kesim frekansı için küçük ve gelişerek gözüktüğü anlaşılr.

Yüksek geçiren filtre, filtre katsayıları ve karakteristikleri

Yüksek geçiren filtrenin filtre katsayısı, alçak geçiren filtrenin filtre katsayısının kullanılmasıyla kolayca bulunabilir. Bunun için yapılan işlemler aşağıda gösterilmektedir.

Ü Alçak geçiren filtrenin filtre katsayısının, normalleştirilmiş 0.5 kesim frekansında olduğu kabul edilir ve istenen yüksek geçiren filtrenin kesim frekansının normalleştirilmiş değeri 0.5'den çıkartılır.

Ü n 'nin değerleri istenilen alçak geçiren filtrenin filtre katsayıları için eski sayıların paragrafının sinyali terslenir.

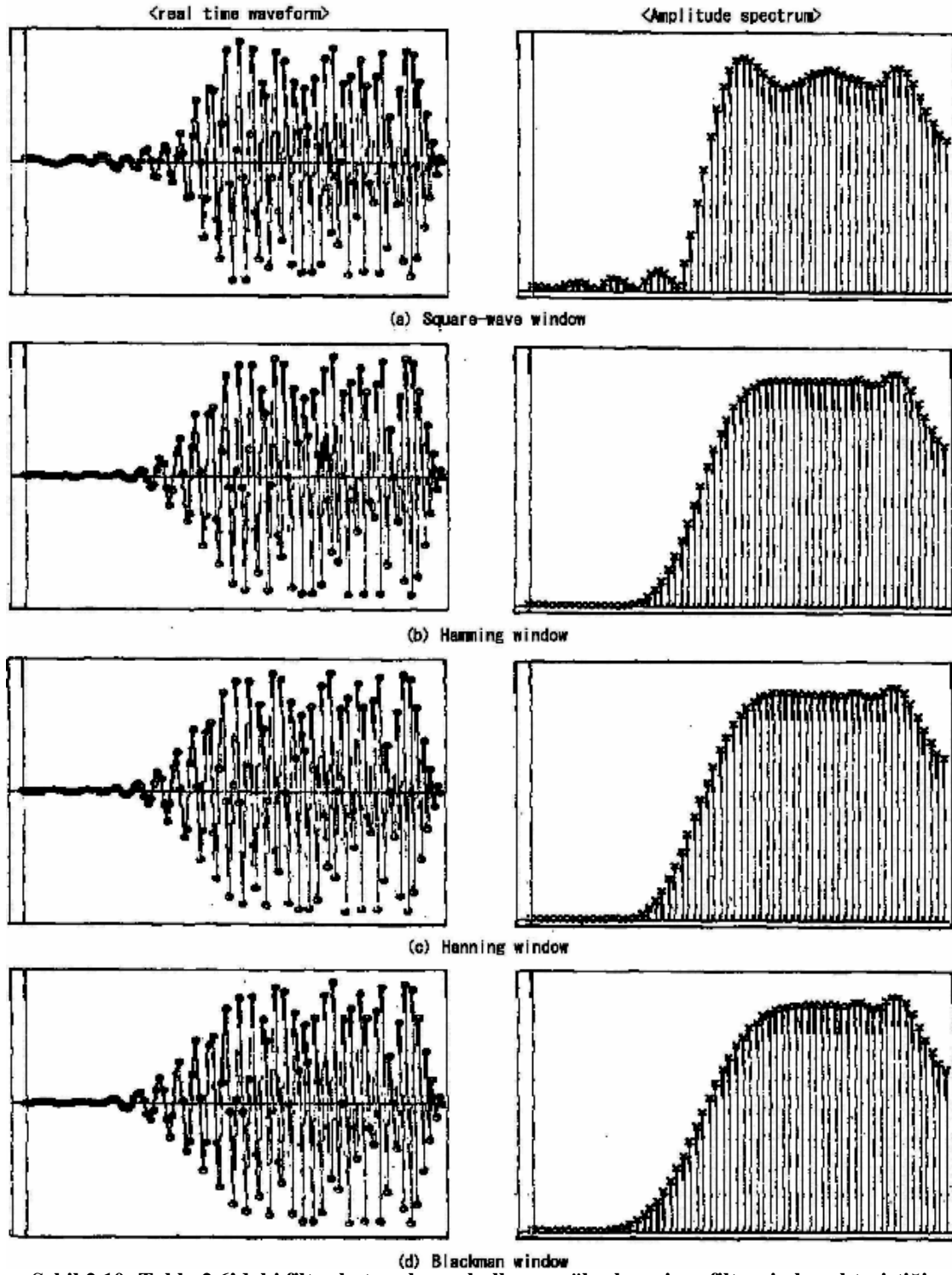
Ü Eğer gerekirse filtre katsayısının elde edilmesinde pencere fonksiyonu koyulur.

Yukarıda anlatılan maddelere bağlı olarak yapılan işlemlerle yüksek geçiren filtrenin istenilen filtre katsayıları örnekleri Tablo 2.6'da gösterilmiştir. Tablo 2.5'ten alınan ve işlenen $0.3(=0.5-0.2)$ normalize edilmiş kesim frekansı değeri alçak geçiren filtrenin filtre katsayısı 1 tanedir, çünkü bu durumda kesim frekansının normalize edilmiş kesim frekansı 0.2'dir. Çünkü alçak geçiren filtrenin filtre katsayıları, kesim frekansı normalize değeri 0.3 olan pencere fonksiyonu uygulanmıştır. Daha önce verilen Tablo 2.4'te daha önce verilen katsayıların yalnız işaretlerinin değiştirilmesi ile de bu mümkündür.

Şekil 2.9 Tablo 2.6'daki her bir filtre katsayısının kullanılmasıyla elde edilen işlemlerin sonuçlarıdır. Bu karakteristiklerde alçak geçiren filtrenin yüksek frekans aralığındaki karşılığı ihmal edilmiştir.

n	Filter coefficient for high-pass filter			
	Square-wave window	Hanning window	Hanning window	Blackman window
0	6.000000e-01	6.000000e-01	6.000000e-01	6.000000e-01
± 1	-3.027307e-01	-2.959150e-01	-2.953224e-01	-2.906970e-01
± 2	-9.354893e-02	-8.533045e-02	-8.461580e-02	-7.944454e-02
± 3	6.236595e-02	5.054020e-02	4.951187e-02	4.298082e-02
± 4	7.568267e-02	5.162677e-02	4.953495e-02	3.858205e-02
± 5	-2.338826e-17	-1.262966e-17	-1.169413e-17	-7.952008e-18
± 6	-5.045612e-02	-2.007368e-02	-1.743181e-02	-1.012988e-02
± 7	-2.672826e-02	-7.206442e-03	-5.508893e-03	-2.709872e-03
± 8	2.338723e-02	3.925598e-03	2.233282e-03	9.404675e-04
± 9	3.363675e-02	3.448237e-03	8.231497e-04	3.092260e-04
± 10	-2.338826e-17	-1.871061e-18	-0.000000e+00	3.245773e-34

Tablo 2.6: Pencere fonksiyonunda kullanılan yüksek geçiren filtrenin filtre katsayıları (kesim frekansı 0.2 normalize değeri için)



Şekil 2.10: Tablo 2.6'daki filtre katsayılarını kullanan yüksek geçiren filtrenin karakteristiği

Band geçiren filtre, filtre katsayıları ve karakteristikleri

Band geçiren filtrenin filtre katsayılarının istenmesi yüksek geçiren filtrenin filtre katsayılarının istenmesi kadar kolay değildir. Burada, Tablo 2.7'deki filtre katsayılarının kullanılmasıyla elde edilen band geçiren filtrenin karakteristiği Şekil 2.10'da örnek olarak gösterilmiştir. Bu şekiller bize band geçiren filtrenin, alçak geçiren filtre ile yüksek geçiren filtrenin birleşimi olduğunu açıklar.

Şekil 2.11 normalize kesim frekansı 0.1 – 0.3 arasındaki değerlerin üzerinde geçiren band geçiren filterde kare dalga pencere (square-wave window) fonksiyonu ile hamming pencere fonksiyonunun kullanıldığı karakteristikler gösterilmektedir

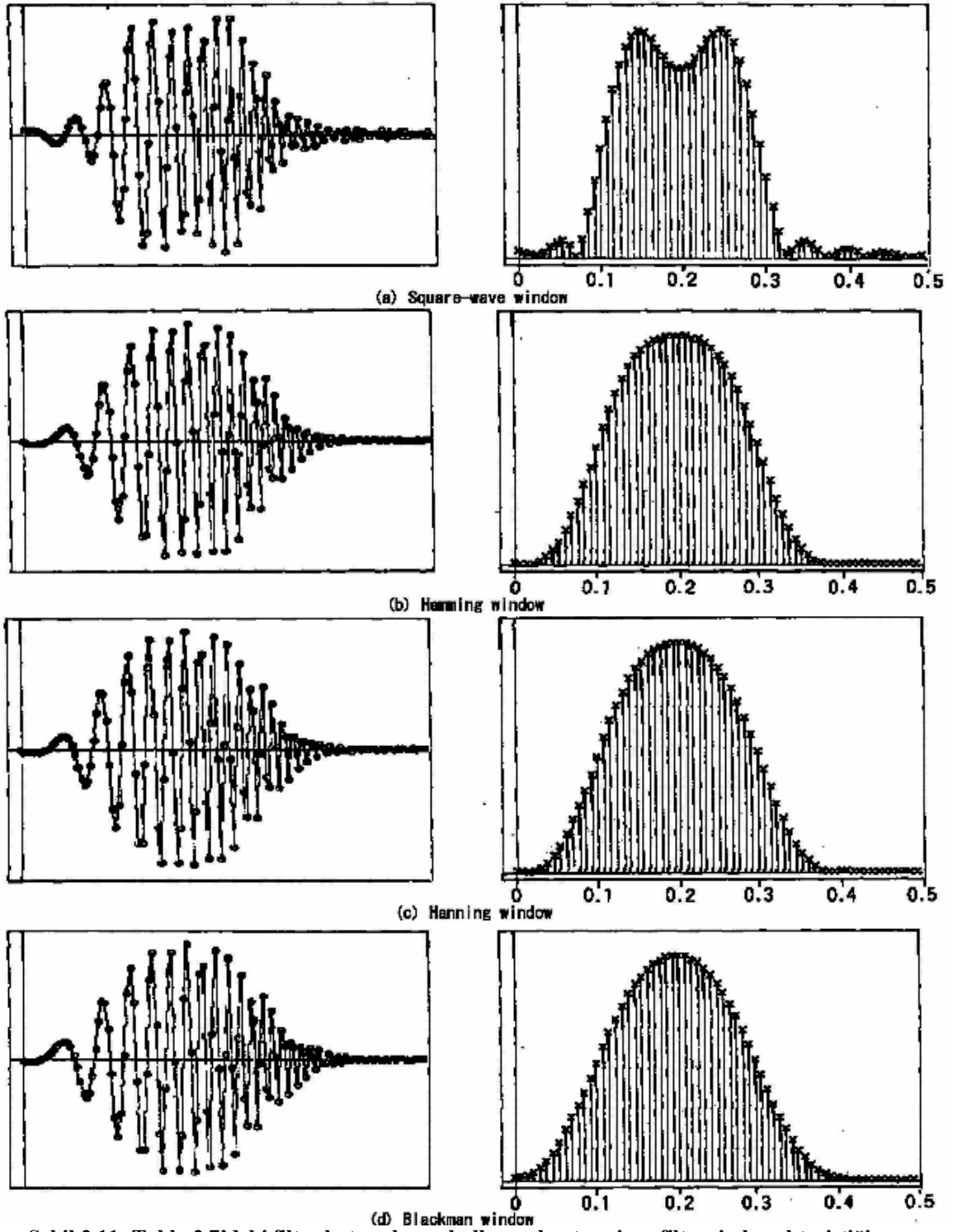
Şekil 2.11 (a) normalize kesim frekansı 0.3 olan alçak geçiren filtrenin karakteristiğini gösterir. Şekil 2.11 (b) ise normalize kesim frekansı 0.1 olan yüksek geçiren filtrenin karakteristiğini gösterir. Band geçiren filtrenin istenilen karakteristikleri yukarıda anlatılan iki filtrenin eşlenik karakteristikleridir.

Şekil 2.11 (a) ve (b) deki sinyallerin band geçiren filtre ile sentezlenmesi sonucu elde edilen dalga şekli Şekil 2.11 (c)'de üst tarafta ve genlik spektrumları Şekil 2.11 (c)'de alt tarafta görülmektedir.

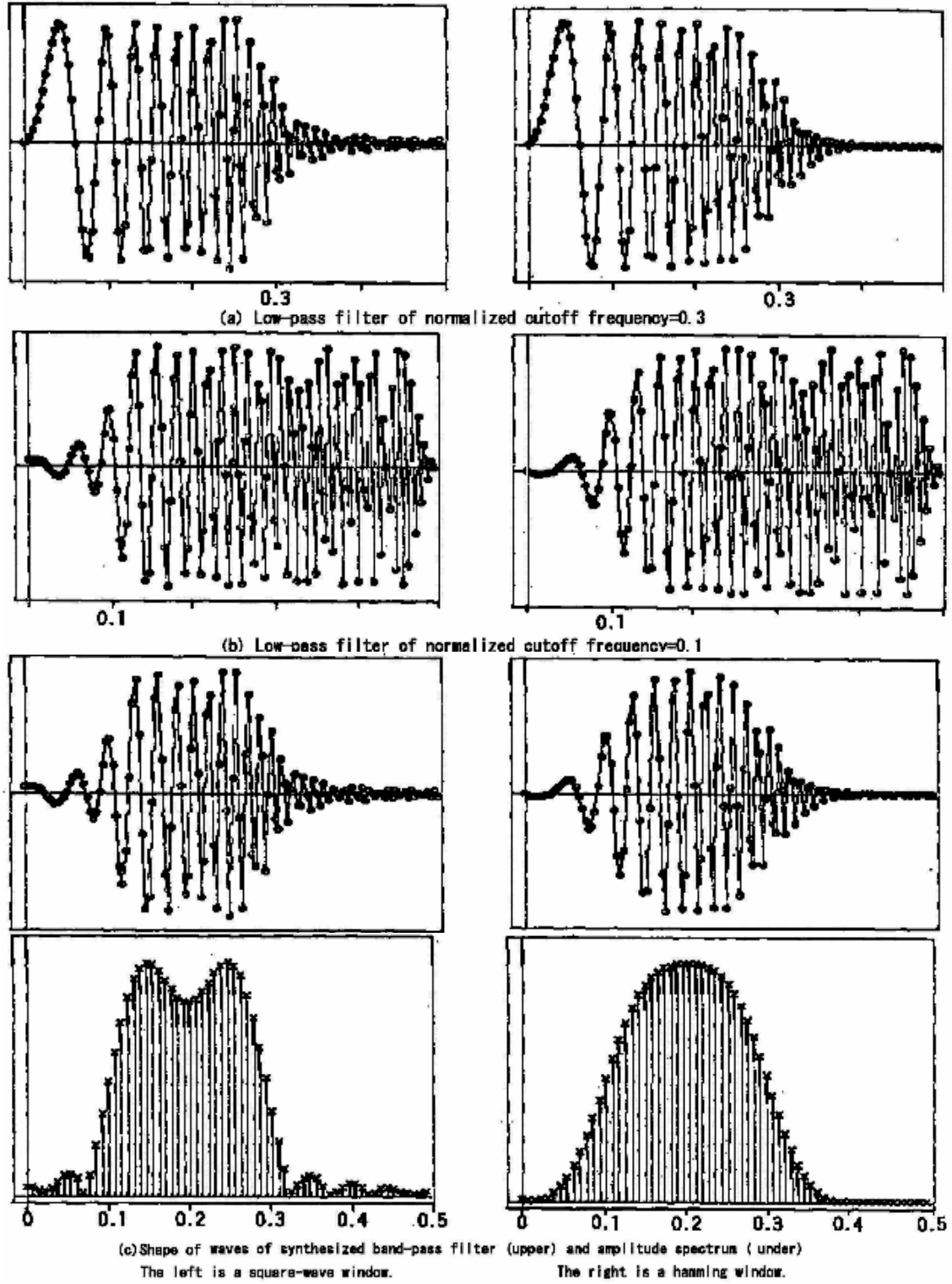
Ayrıca, Şekil 2.11 (c)'deki alt taraftaki band geçiren filtrenin sentezlediği dalganın genlik spektrumları karşılaştırılarak hangi pencere fonksiyonunun kullanıldığı anlaşılabilir.

n	Filter coefficient for band-pass filter			
	Square-wave window	Hamming window	Hamming window	Blackman window
0	4.000000e-01	4.000000e-01	4.000000e-01	4.000000e-01
± 1	1.156329e-01	1.130295e-01	1.128031e-01	1.110364e-01
± 2	-2.449142e-01	-2.233981e-01	-2.215270e-01	-2.079886e-01
± 3	-1.632761e-01	-1.323159e-01	-1.296238e-01	-1.125253e-01
± 4	2.890821e-02	1.971967e-02	1.892067e-02	1.473703e-02
± 5	1.559217e-17	8.419774e-18	7.796086e-18	5.301338e-18
± 6	-1.927214e-02	-7.667466e-03	-6.658358e-03	-3.869270e-03
± 7	6.997550e-02	1.886671e-02	1.442247e-02	7.094537e-03
± 8	6.122856e-02	1.027735e-02	5.846808e-03	2.462175e-03
± 9	-1.284809e-02	-1.317110e-03	-3.144152e-04	-1.181138e-04
± 10	-1.559217e-17	-1.247374e-18	-0.000000e+00	2.163848e-34

Tablo 2.7: Pencere fonksiyonunda kullanılan band geçiren filtrenin filtre katsayıları (Kesim frekansları 0.1 ve 0.3 normal değerlerinde iken)



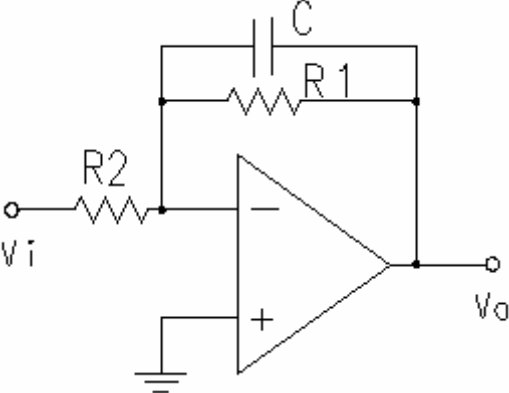
Şekil 2.11: Tablo 2.7'deki filtre katsayılarını kullanan bant geçiren filtrenin karakteristiği

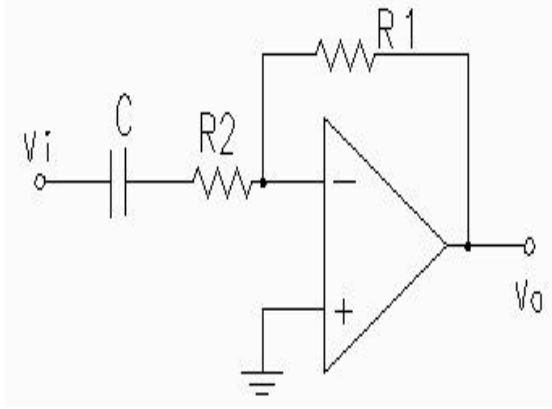


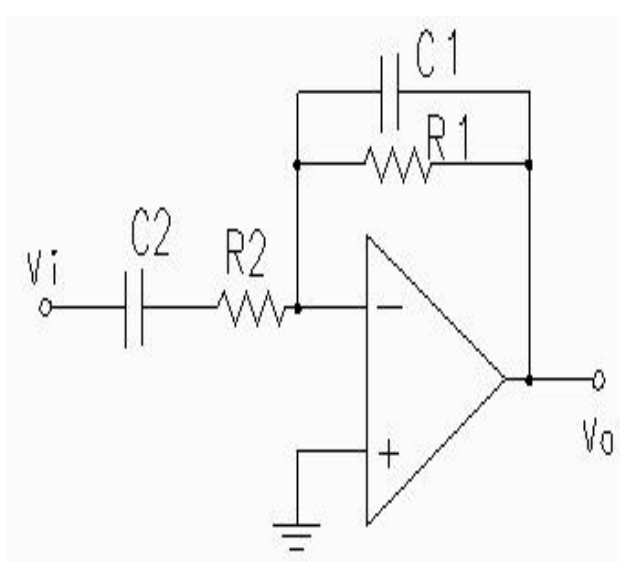
Şekil 2.12: Bant geçiren filtre (Band-pass filter) ile Alçak geçiren filtre (low-pass filter) ve Yüksek geçiren filtre (high-pass filter)

UYGULAMA FAALİYETİ

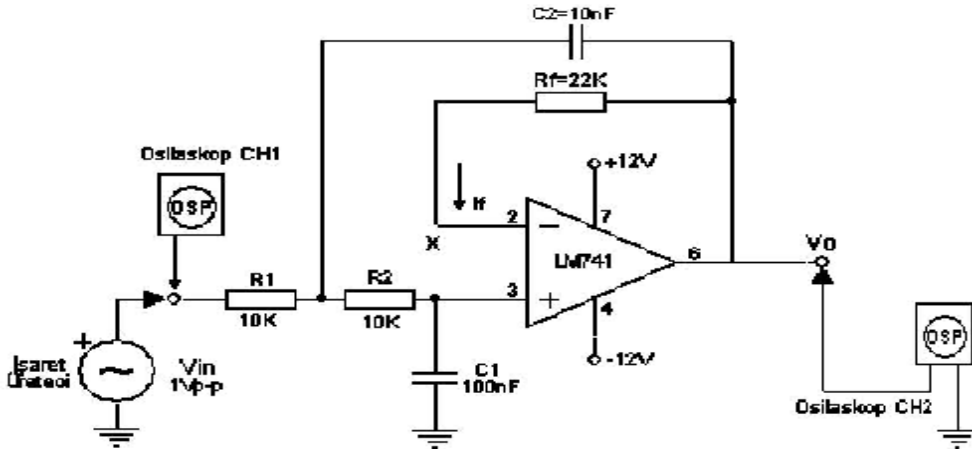
UYGULAMA.1

OP-amp'la 1.Düzye Filtre Devrelerini Kavrama	
İşlem Basamakları	Öneriler
<p>Ø Aşağıdaki OP-AMP'LA 1. düzey alçak geçiren filtrenin devresini defterinize çizerek çalışma prensibi hakkında düşününüz.</p>  $\frac{V_o}{V_i} = \frac{H_o \omega_o}{s + \omega_o}, \quad \text{or} \quad \left \frac{V_o}{V_i} \right = H_o \frac{\omega_o}{\sqrt{\omega^2 + \omega_o^2}}$	<p>Ø Alçak geçiren filtre için yazdığımız transfer fonksiyonu 1. düzey alçak geçiren filtrelerin genel formudur. Yüksek frekanslarda ($\omega \gg \omega_o$), kapasitör kısa devre gibi davranacağından yükseltici kazancı sıfıra gidecektir. Çok düşük frekanslarda ise ($\omega \ll \omega_o$), kapasitör açık devre gibi davranacağından devrenin kazancı H_o'a eşit olacaktır. Peki alçak yada yüksek frekans denilince ne anlıyoruz?</p> <p>Yüksek frekanslarda akımın büyük çoğunluğu kapasitörden geçer. Örneğin; kapasitörün empedans büyüklüğünün R_1'den çok küçük olduğu zaman. Diğer bir ifadeyle $1/\omega C \ll R_1$ yada $\omega \gg 1/R_1 C = \omega_o$'sa yüksek frekansa sahibiz. R_1'in devreye az bir etkisi olacak ve bu etkiden dolayı devre integral alıcı gibi davranacaktır. Ve de $\omega \ll 1/R_1 C$ ise alçak frekans meydana gelir ve devre kazancı $H_o = -R_1/R_2$ olan bir yükseltici etkisi gösterir.</p>

OP-amp'la 1.Düzy Filtre Devrelerini Kavrama	
İşlem Basamakları	Öneriler
<p>Ø Aşağıdaki OP-AMP'LA 1. düzey yüksek geçiren filtrenin devresini defterinize çizerek çalışma prensibi hakkında düşününüz.</p>  $\frac{V_o}{V_i} = H_0 \frac{s}{s + \omega_0}, \quad \text{or} \quad \left \frac{V_o}{V_i} \right = H_0 \frac{\omega}{\sqrt{\omega^2 + \omega_0^2}}$	<p>Ø Alçak geçiren filtreler için kullanılan yöntemleri kullanarak 1. düzey yüksek geçiren filtre için transfer fonksiyonunu yandaki gibi elde edebiliriz. Yüksek frekanslarda ($\omega \gg \omega_0$) kapasitör kısa devre gibi davranacağından yükseltecin kazancı $H_0 = -R_1/R_2$ olacaktır. Çok alçak frekanslarda ise ($\omega \ll \omega_0$) kapasitör açık devre gibi davranır ve kazanç H_0 olur. Bu devre için merkez frekansı yazacak olursak $\omega_0 = 1/R_2C$ dir. Bu yüzden bu devre yüksek geçiren bir filtredir yani yüksek frekanslı sinyalleri geçirir, alçak frekanslı sinyalleri bloke eder.</p>

OP-amp'la 1.Düzy Filtre Devrelerini Kavrama	
İşlem Basamakları	Öneriler
<p>Ø Aşağıdaki OP-AMP'LA 1. düzey basit bant geçiren filtrenin devresini defterinize çizerek çalışma prensibi hakkında düşününüz.</p> 	<p>Ø Alçak geçiren filtrelerin yanı sıra, diğer genel filtre tipleri yüksek geçiren (sadece yüksek frekansları geçirir), band durduran filtreler (belli frekanstaki sinyalleri bloklar) ve band geçiren filtrelerdir (Alçak ve yüksek frekansları söndürür, sadece belli frekans aralığındaki sinyalleri geçirir). Basit bant geçiren filtrede daha önce gördüğümüz alçak geçiren ve yüksek geçiren filtrelerin kombinasyonu ile elde edilir. Bu devreyi alçak frekansları ($\omega \ll 1/R_2 C_2$) ve yüksek frekansları ($\omega \gg 1/R_1 C_1$) durduracak fakat kazancın $-R_1/R_2$ olduğu orta frekansları geçirecektir. Bunun yanı sıra, bu devre çok dar bantlı bir filtre yapmak için kullanılamaz.</p>

UYGULAMA.2

Alçak Geçiren Filtre Devresi	
İşlem Basamakları	Öneriler
<ol style="list-style-type: none"> 1. Aşağıdaki alçak geçiren filtre devresini kurunuz. Sizden istenilenleri hesaplayarak alçak geçiren filtre devresinin özelliklerini ve çalışma karakteristiklerini inceleyiniz. 2. Şekil 2.13'te Opamplar gerçekleştirilmiş alçak geçiren filtre devresini deney seti üzerine kurunuz. İşaret üreticini başlangıçta devreye bağlamayınız. İşaret üreticinin çıkış genliğini tepeden tepeye 1V_{pp} sinüsoidal bir gerilim verecek şekilde ayarlayınız. İşaret üreticinin çıkış frekansını ise Tablo 2.8'de belirtilen 250 Hz değerine ayarlayınız. 	<p>Aşağıdaki donanımları temin ediniz.</p> <ol style="list-style-type: none"> 1) Multimetre (Sayısal veya Analog) Osilaskop 2) Güç kaynağı (± 12 V DC) 3) Opamp (LM 741 tipi) 4) Direnç : 2x10 KΩ, 22 KΩ, 100KΩ 5) Kondansatör :10 nF, 100 nF
 <p>Şekil 2.13: Alçak geçiren filtre devresi</p>	
<ol style="list-style-type: none"> 3. Giriş ve çıkış işaretlerini ölçmek için gerekli Osilaskop bağlantılarını yapınız. İşaret üreticini alçak geçiren filtre devresinin girişine bağlayınız. Çıkış işaretinin tepeden tepeye değerini (V_{opp}) Osilaskop ile ölçerek sonucu yine Tablo 2.8'deki ilgili sütuna kaydediniz. 4. Deneyi işaret üreticinin çıkış genliği her durumda aynı kalmak koşuluyla Tablo 2.8'de verilen her frekans değeri için tekrarlayınız. Her frekans için elde ettiğiniz çıkış geriliminin tepeden tepeye değerini ilgili yere kaydediniz. 	

5. Tablo 2.8'deki verilerden yararlanarak her frekans değerinde -40dB/Dekad'lık alçak geçiren filtre devresinin gerilim kazancını hesaplayarak sonuçları Tablo 2.8'deki ilgili yerlere kaydediniz.
6. Uygulamasını yaptığınız alçak geçiren filtre devresinin kesim frekansı (F_c) aşağıdaki formülden bulunur. Bu formülü kullanarak kesim frekansını bulunuz. Bulduğunuz sonucu kaydediniz.

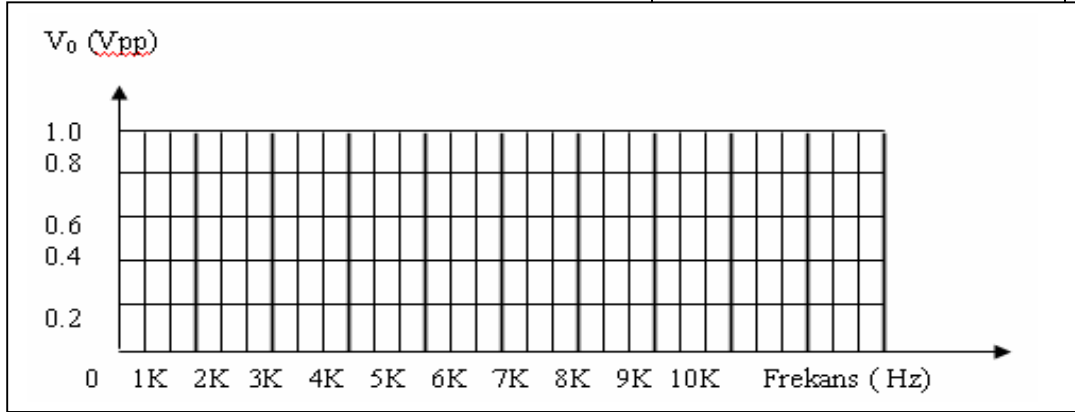
$$F_c = \frac{1}{2p\sqrt{R_1 R_2 C_1 C_2}}$$

- Ø Alçak geçiren filtre devresinin kazancını ($A=V_o/V_{in}$) her bir frekans değeri için ayrı ayrı hesaplayarak tabloya kaydediniz.

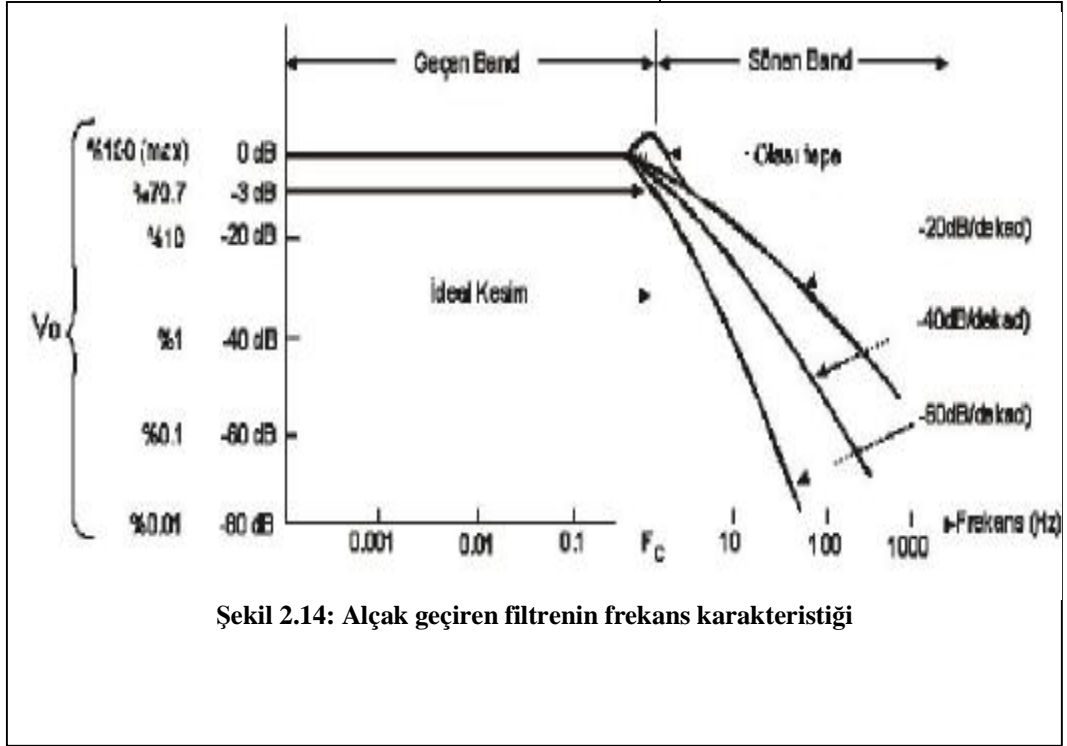
Alçak geçiren filtre devresi $V_{in} = 1 \text{ Vpp}$ Sinüsoidal										
Fin(Hz)	250	500	750	1K	1.25K	1.5K	2K	3K	5K	10K
$V_o(V_{pp})$										
$A=V_o/V_{in}$										

Tablo 2.8: -40dB/Dekad'lık Alçak geçiren filtre devresinin bilgi tablosu

- Ø Tablo 2.8'deki sonuçlardan yararlanarak alçak geçiren filtrenin kazanç-frekans karakteristiğini alta yer alan diyagrama orantılı olarak çiziniz.



Ø Alçak geçiren filtrenin frekans karakteristiğini aşağıdaki gibi elde etmelisiniz.



Şekil 2.14: Alçak geçiren filtrenin frekans karakteristiği

ÖLÇME VE DEĞERLENDİRME

OBJEKTİF TESTLER (ÖLÇME SORULARI)

Aşağıdaki soruları doğru “D” veya yanlış “Y” olarak cevaplayınız.

- (.....) 1. FIR tip filtrenin çalışma prensibi: en son gelen veri şeklin en solundan girdiğinde bu veri işlenir ve bir veri sağa doğru kaydırılır. Ve en son veri tekrar en son tarafta giriş olur. Filtre katsayıları (h_n) sırasıyla hazırlanarak yerleştirilir, her bir veri karşılaştırılır, sonuçlar toplanır ve filtre çıkışında gösterilir.
- (.....) 2. IIR tip filtrede, birkaç çeşit analog filtre devre yapıları vardır. Bu filtrelerin her bir özelliğini en iyi şekilde ve tam anlamıyla kullanmak gerekir.
- (.....) 3. IIR tip filtre sayısal bir filtredir.
- (.....) 4. Bir filtre karakteristiğinde geçiş bandının bitip transition (geçiş) bandının ve zayıflamanın başladığı sınırdaki frekans kesim frekansı olarak bilinir.
- (.....) 5. Alçak geçiren bir filtre devresi kesim frekansının altında kalan alçak frekanslara sahip sinyalleri geçirip yüksek frekanslı sinyalleri bloke ederler. Yüksek geçiren bir filtre devresi kesim frekansının üstünde kalan yüksek frekanslara sahip sinyalleri geçirip alçak frekanslı sinyalleri bloke eder.
- (.....) 6. Filtreler sadece amfileri rahatlatmak için değil de hoparlörleri gereksiz yere çıkarmayacağı sesleri içeren sinyallerden uzak tutmak için kullanılır. Bir de, kabin içinde birden çok hoparlör varsa, hoparlörlerin toplam empedansının amfi empedansına uydurmak için kullanılır. Yani hem anfiyi hem de hoparlörleri rahatlatır.
- (.....) 7. Bant geçiren bir filtre devresi iki kesim frekansının arasında kalan frekanslara sahip sinyalleri geçirip alçak ve yüksek frekanslı sinyalleri bloke ederler.
- (.....) 8. Sayısal filtrelerde, kesim frekansı frekans örneklemeyle yakından ilgilidir.
- (.....) 9. Sayısal filtrelerde filtre katsayısının aşamalı değerleri artırılır ve azaltılabilir. Çünkü sayısal filtreler sadece programdan meydana gelmektedir. Filtrenin azalan karakteristiği iptal edilebilir ve bu çeşitli filtrelerle filtre katsayılarının değiştirilmesiyle mümkün olur.
- (.....) 10. Band geçiren filtre, alçak geçiren filtre ile yüksek geçiren filtrenin birleşimidir.

DEĞERLENDİRME

Cevaplarınızı cevap anahtarı ile karşılaştırınız. Doğru cevap sayınızı belirleyerek kendinizi değerlendiriniz. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt yaşadığınız sorularla ilgili konuları faaliyete dönerek tekrar inceleyiniz.

MODÜL DEĞERLENDİRME

YETERLİK ÖLÇME

Modülde yaptığınız uygulamaları tekrar yapınız. Yaptığınız bu uygulamaları aşağıdaki tabloya göre değerlendiriniz.

Açıklama: Aşağıda listelenen davranışları gözlediyseniz EVET , gözlemediyseniz HAYIR kutucuğuna (X) işareti koyunuz.			
Değerlendirme Ölçütleri		Evet	Hayır
Ø	Analog ve dijital bilginin ne olduğunu öğrendiniz mi?		
Ø	Analog bilginin dijital bilgiye nasıl dönüştürüldüğünü anladınız mı?		
Ø	Dijital bilginin analog bilgiye nasıl dönüştürüldüğünü anladınız mı?		
Ø	Mikroişlemcinin ne demek olduğunu biliyor musunuz?		
Ø	Sayısal İşaret İşleme işlemleri için gerekli mikroişlemcinin seçiminde nelere dikkat edilmelidir, öğrendiniz mi?		
Ø	Bir DSP'nin içerisinde hangi birimlerin olduğunu öğrendiniz mi?		
Ø	Sayısal Filtrenin ne anlama geldiğini öğrendiniz mi?		
Ø	Örneklemenin ne anlama geldiğini öğrendiniz mi?		
Ø	FIR ve IIR filtrelerin ne ifade ettiğini anladınız mı?		
Ø	Kesim Frekansının ne anlama geldiğini öğrendiniz mi?		
Ø	Yüksek geçiren, Alçak geçiren, Band geçiren filtreleri öğrendiniz mi?		

DEĞERLENDİRME

Hayır cevaplarınız var ise ilgili uygulama faaliyetini tekrar ediniz. Cevaplarınızın tümü Evet ise bir sonraki modüle geçebilirsiniz.

CEVAP ANAHTARLARI

ÖĞRENME FAALİYETİ-1 CEVAP ANAHTARI

1	D
2	D
3	Y
4	D
5	D
6	D
7	Y
8	D
9	D
10	D

ÖĞRENME FAALİYETİ-2 CEVAP ANAHTARI

1	D
2	D
3	Y
4	D
5	D
6	D
7	D
8	D
9	D
10	D

KAYNAKÇA

- Ø NAZMAN Mustafa, Melek TOTAN, **Sayısal İşaret İşleme**, ETOGM-JICA, İzmir, 2005.
- Ø http://www.maxim-ic.com/appnotes.cfm/an_pk/2081
- Ø <http://www.discovercircuits.com/O/o-sine.htm>
- Ø <http://picproje.org/main/modules.php?name=News&file=print&sid=11>
- Ø <http://www.elektronikekibi.com/durmus/af.html>
- Ø <http://sutef.selcuk.edu.tr/~otomasyon/images/stories/Analog/deneyler/anal6.pdf>