

SAYISAL İŞARET İŞLEME (DSP), BİR DSP MİMARİSİNİN İNCELENMESİ VE UYGULAMALARI

Türker CAMBAZOĞLU (*)

Sayısal İşaret İşleme (DSP) Nedir?

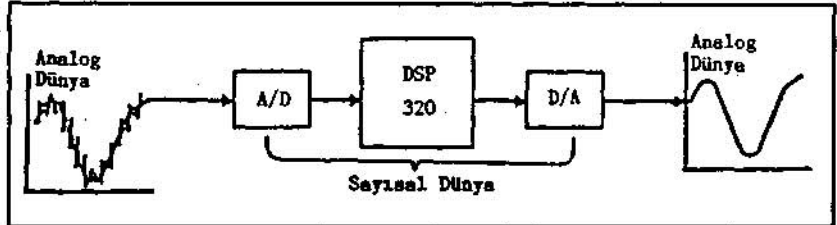
Neden DSP?

Teorik bir disiplin olarak DSP, f¹ uzun yıllar önce ortaya çıkmış bir konudur. Kısaca, sayısal işaret işleme; sinyallerin sayı dizileri şeklinde temsil edilmesi ve bu sayı dizilerinin, nümerik hesaplama yöntemleri ile dönüştürülmesi (transformasyon) veya işlenmesi ile ilgilidir. O halde genel anlamda bir DSP sisteminde, dış dünya ile haberleşebilmek için A/D ve D/A çeviriciler bulunacaktır (Şekil -1). Analog işaret dururken, neden sayısal domene geçerek sinyal işlemeye gerek duyulduğu ise ayrı bir konudur. Analog sinyal işlemede, sinyal, sürekli zaman ve sürekli genlik seviyeleri ile işlenir. Bir DSP sisteminde ise, analog sinyal artık zaman aralıklarında örneklenir. Örnek alınmış analog sinyal genlikleri bir A/D ile sayısal değerlere çevrilir ve sinyalin sayısal eşdeğeri, sayısal domende sayı dizileri halinde, DSP'nin kendisi tarafından işlenir. Bu işlemler; sinyallerin analiz edilmesi veya sentezlenmesi (FFT'ler, konuşma sentezi gibi), sinyallerin frekans veya genlik yanıtlarının değişikliğe uğratılması (süzgeçler, denetleyiciler gibi) ve hatta sistemlerin veya sinyallerin kestirimi (tahmin edilmesi) (Kalman filtreleri, adaptif filtreler, denetleyiciler, vs.) gibi şeyler olacaktır. DSP ile üzerinde işlem yaptıktan sonra, veri, D/A çeviriciler yardımı ile analog biçime dönüştürülür.



performans sağlarlar. Bir takım karmaşık işaret (sinyal) işleme teknikleri vardır ki, bunları analog yöntemlerle halletmek ya çok zordur ya da tümünden olanaksızdır. DSP'ler programlanabilir olduğundan, sistemler üzerinde değişiklikler veya yukarı doğru güncelleştirmelerle kolayca başedilebilir. Bundan başka, tek bir DSP'yi, farklı sayıda sinyal işleme için zaman içinde paylaşımak mümkündür.

DSP sistemleri sorunsuz değildir. Bu tür sistemlerde, analog sinyaller işlenmeden önce sayısal biçime çevrildiğinden; zaman ve genliğin sürekli durumdan ayrık duruma çevrilmesi, çeşitli kısıtlamaları da beraberinde getirir. Eğer bu kısıtlamalara uyulmazsa, oldukça ciddi sorunlarla karşılaşılır. Gözönüne alınması gerekli en önemli nokta, örnekleme aralığının seçimidir. Bu seçimi yapabilmek için genel bir kural; "analog işaret içindeki bilgiyi kaybetmemek için, işaret en azından en yüksek frekanslı bileşenin 2 katı frekans ile örneklenir" biçiminde ifade edilir. Fakat, örneğin sayısal kontrol sistemlerinde, örnekleme hızı, sistem band genişliğinin 6 ila 10 katı bir değerde seçilir.



Şekil 1: EN GENEL bir DSP Sistemi

Sinyallerin, DSP teknikleri ile sayısal domende işlenmesinin, analog domende işlenmesine göre birçok üstünlükleri vardır. Eskime, sıcaklık ile parametre değerlerinin kayması gibi ortam koşullarına bağlı analog elemanların tersine, DSP elemanları ortam koşullarından daha az etkilenirler ve bu halleriyle daha kararlı ve daha güvenilir bir

Gerçek zamanda işlemler yaparken, sayısal işaret işlemci bir sonraki örnek değerini almadan önce, tüm işlemlerini bitirip, hazır hale gelmelidir. Bu nedenle, DSP, son derece hızlı olmalıdır. Bu yüksek performansı sağlayabilmek için DSP'lerin iç mimarileri donanımsal olarak birbirine bağlı birimlerden oluşur Halbuki geleneksel mikroişlemci

(*) EMPA AŞ. Araştırma Bölümü

ve mikrodenetleyicilerde, iç mimari "mikrokod" denilen yöntemle oluşturulur, ve bu yüzden bunlar genel olarak yavaşlardır, işte, bu "donanımsal olarak birbirine bağlı birimlerden oluşan mimarisi" sayesinde DSP'ler komutların birçoğunu tek bir makina çevriminde yürütürler. Örneğin Texas Ins TMS320CIX ailesi için, bu çevrim süresi ya 160ns ya da 200 ns olabilmektedir.

Birçok analog işaret işleme sistemi, PID² algoritmalarına benzer bir biçimde diferansiyel denklemleri gerçekleştirebilirler. Bu tür algoritmalar sayısal işaret işlemede:

$$y(n) = a_1 y(n-1) + a_2 y(n-2) + \dots + b_0 x(n) + b_1 x(n-1) + \dots$$

şeklinde fark denklemlerine dönüşür. Burada $y(n)$, çıkış sinyali örneklerini $x(n)$ ise giriş sinyali örneklerini göstermektedir, (o anda güncel olan örnekler). Benzer şekilde $y(n-k)$ ve $x(n-k)$ k zaman önce güncel olan örnekleri, $a_1, a_2, \dots, b_0, b_1, \dots$ ise kazanç katsayıları gösterir.

Yukarıdaki denklemlerden de anlaşılacağı üzere, işaret işlemede yaygın bir biçimde kullanılan işlem, çarpım-biriktirme (akümülayon) işlevidir. Bu tür işlemlere bazı örnekler; süzme, korelasyon, pencere fonksiyonları ile çarpma³¹, hızlı Fourier dönüşümü (FFT⁴¹) sayılabilir. O halde, tek yongadan ibaret iyi bir sayısal işaret işlemcide, çok hızlı çarpımlar yapmak üzere, yonga üzerinde donanımsal bir çarpma birimi bulunması çok gereklidir. Ö-

neğin, TMS320CIX ürünlerinde yer alan 16x16 bit paralel çarpma birimi, çarpma neticesini 32-Bit şeklinde tek bir makina çevriminde, yani 160 veya 200 ns'de çıkarır. Buna karşılık, örneğin 68020 gibi 32-Bit'lik bir mikro işlemcide aynı iş için 36 tane CPU⁵¹ (Merkezi İşlem Birimi) çevrimi gerekecektir. TMS320CIX ürünlerinde, çarpma birimine bağlı kaydedicilerden (saklayıcılardan) biri, daha önceden yüklenmesi gerekeceğinden, bir tablodaki elemanlar çiftler çiftler 400ns'lik (veya 320 ns'lik) hızlarda çarpılarak toplanırlar (biriktirme işlemi).

Analog işaretlerin sürekli genliklerinin, ayırık genlik seviyelerine dönüştürülmesi, ayırık genlik seviyelerinin kaybolması anlamına gelmektedir. Bu olay, kuantalama hatası olarak da bilinmektedir. Kuantalama hatasına ek olarak, sayısal domende çalışmak başka hataları da yol açar. Örneğin 8x8 Bit'lik bir çarpmanın sonucu 16 Bit'dir. Eğer sadece 8 - Bit'lik bir saklama gözü mevcutsa, o vakit daha az anlamlı olan 8-Bit atılır. Sözcük uzunluğunun kısıtlı oluşu nedeniyle ortaya çıkan bu hataya kesme hatası⁷¹ adı verilmektedir.

Kuantalama ve kesme hataları özellikle 8-Bit'lik mikro işlemcilerde önemli sorunlara yol açarlar. Aynı sorunlar, biraz daha hafiflemiş olarak 16-Bit'lik mikro işlemcilerde de mevcuttur. Kesme ve kuantalama hataları hem işaret örnekleri üzerinde hem de kazanç katsayıları üzerinde

Yaklaşın	Üstün Yanı	Zayıf Yanı
Bit - Slice	<ul style="list-style-type: none"> • Çok Hızlı • Donanımsal Çarpıcı 	<ul style="list-style-type: none"> • Yonga sayısı/Güç tüketimi/maliyet yüksek • Karmaşık Tasarım • Programlamak zor
İsmlenmiş Yongalar (ASIC)	<ul style="list-style-type: none"> • Üretim miktarları yeterince yüksek ise, en düşük üretim maliyeti • iyi tanımlanmış performans 	<ul style="list-style-type: none"> • Sadece çok yüksek hacimli uygulamalar için geçerli • tik geliştirme maliyetleri yüksek. • Yüksek risk.
Genel amaçlı mikro işlemciler	<ul style="list-style-type: none"> • Kolay programlanabilirlik • Kolay tasarım • Düşük maliyet 	<ul style="list-style-type: none"> • Donanımsal çarpıcı mevcut değil • Hız düşük • Özel DSP işlevleri içermez
Genel amaçlı DSP İşlemcileri (TMS320)	<ul style="list-style-type: none"> • Kolay programlanabilirlik • Kolay tasarım • Düşük maliyet • Hızlı • Donanımsal/Çarpıcı içerir • Özel DSP işlevleri • Belirli bir uygulama ile sınırlı değildir. 	<ul style="list-style-type: none"> • Orta Hız (Bit-Slice'a göre) • Yonga sayısı ASIC çözüme göre biraz fazla • Alçak/Orta hacimlerde üretim

Tablo 1: DSP UYGULAMALARI için çeşitli çözümler

etkilidir. Bu etkiler, kendisini kararsızlıklar ve osilasyonlar⁸¹ şeklinde gösterir.

Alışıl gelmiş 8 ve 165-Bit'lik mikro işlemcilerde kesme (veya yuvarlatma) hatasının üstesinden gelebilmek için bir yol, çok sayıda Byte'i peşpeşe uzatarak, nümerik duyarlılığı (presisyon) arttırmaktır. Tabii bunun sonucu olarak, bir işlemin yürütülme hızı düşecektir.

Sayısal işaret işlemede, aritmetik değerlerin üstten taşması çeşitli sorunlar yaratmaktadır. Üstten taşmalar sonucunda, çok büyük sayılarla çok küçük sayılar arasında salınımlar ortaya çıkmakta ve bu, sistemin hatalı davranmasına yol açmaktadır. Bu tür davranışları gidermek üzere, örneğin DSP ailesinde TMS320 özel bir üstten taşma modu mevcuttur. Eğer SOVM komutu kullanılarak üstten taşma modu yetkili kılınmışsa (izin verilmişse), bir taşma oluşması halinde, AMB (Aritmetik Mantık Birimi) tarafından

temsil edilen en büyük/en küçük sayısal eğer akümülatöre yüklenir. Bu özellik, analog sistemlerdeki "doyma (satürasyon)" karakteristiğine benzemektedir. Üstten taşıma-modu ROVM komutu ile yasaklanabilir. Bu durumda, taşıma değerleri hiçbir değişikliğe uğratılmadan akümülatöre yüklenir (2'nin tümleyenini sayı sisteminde dairesel simetri nedeniyle katlanma etkisi ortaya çıkarır).



"Eğer yazılım ve donanım, arabağlaşım devreleri hedeflenmiş olan sisteminkinden farklı bir ortamda test ediliyorsa, sözkonsu test ortamında geçerli olan zamanlama sınırlamalarının bilinmesi gerekir."

Sayısal işaret işleme açısından oldukça önemli olabilecek bir diğer özellik, işlemcinin bir veri örneği üzerine gecikme (Z domende Z^{-1} operatörü) uygulayabilme yeteneğidir. Tanım olarak; bir $x(n-k)$ örnekler dizisinin, $x(n-k-1)$ örnekler dizisine dönüştürülmesi işlemi "birim örnek gecikmesi" olarak adlandırılır ve $x(n)$ 'nin $X(z)$ Z^{-1} dönüşümünü Z^{-1} ile çarparak elde edilir. Birim gecikme işlemi özellikle süzme, konvolüsyon ve korelasyon gibi fonksiyonların gerekli olduğu uygulamalarda çok kullanışlıdır. Örneğin TMS320C1X'de, DMOV komutu, veriyi RAM'daki bir sonraki saklama gözüne ötelileyerek, birim gecikme işlemi gerçekleştirir. Bu arada LTD diye tek bir komut; DMOV komutu ile LT (T kaydedicisini yükle) ve APAC (P kaydedicisinin değerini biriktirir) komutlarını paralel olarak aynı anda icra eder. Bu sayede saniyede 2.5 milyon örnek hızında konvolüsyon işlemleri yürütülebilir.

DSP Uygulamalarında, TMS320 gibi DSP yongaları dışında 3 tane daha yaklaşım mevcut olup, 4 çözümün üstün ve zayıf yanları Tablo-1'de özetlenmiştir.

Sayısal İşaret İşlemci Ürünlerin Seçiminde Rol Oynayan Etkenler

Sayısal işaret işlemci (DSP) yongalarının birçoğu genel amaçlı mikroişlemcilerle benzemektedir. Bu yüzden mikroişlemcilerle tecrübe kazanmış bir tasarımcı için, bu tecrübe DSP ürünlerinin seçimi ve kullanılmasında yararlı olacaktır. Fakat bir kere, dışardan bağlanacak bellekler, anabilgisayarlar ve giriş/çıkış birimleri ile ilgili basit arabağlaşım sorunları halledildikten sonra benzerlikler kaybolur. Uygun DSP yongası seçiminin yapılabilmesi için bellek yapıları, "performans düzeylerini"¹⁰ gösteren programlar, "pipeline'lar, kesmeler"¹⁰, gibi kavramların bilinmesi gerekir.

DSP'lerde, çoğunlukla veri ve program belleklerinin aynı olduğu Harvard-tipi mimari yapı kullanılır, ilk DSP yongalarının içinde, program adımlarının saklandığı bir ROM bulunmaktaydı. Yeni DSP ürünlerinde ise, program dış bellekte saklanabilmektedir. Göze çarpan bir diğer nokta ise, daha yeni DSP yongalarında, VERİ de dışardan bağlanan bellekte tutulmaktadır.

Her ne kadar DSP üreticileri, komut ve veriye ilişkin bellek alanlarının ayrı olması gerektiğinde anlaşılıyorsa da, bu bellek yapılarının nasıl gerçekleştirileceği konusunda iki ayrı eğilim mevcuttur. Eğilimlerin birinde, dışardan bağlanacak bellek ve giriş/çıkış birimleri gibi elemanlarla haberleşebilmek üzere tek bir adres ve veri yolu¹¹ sağlanmaktadır. Tek adres/veri yolunu destekleyen üreticiler,

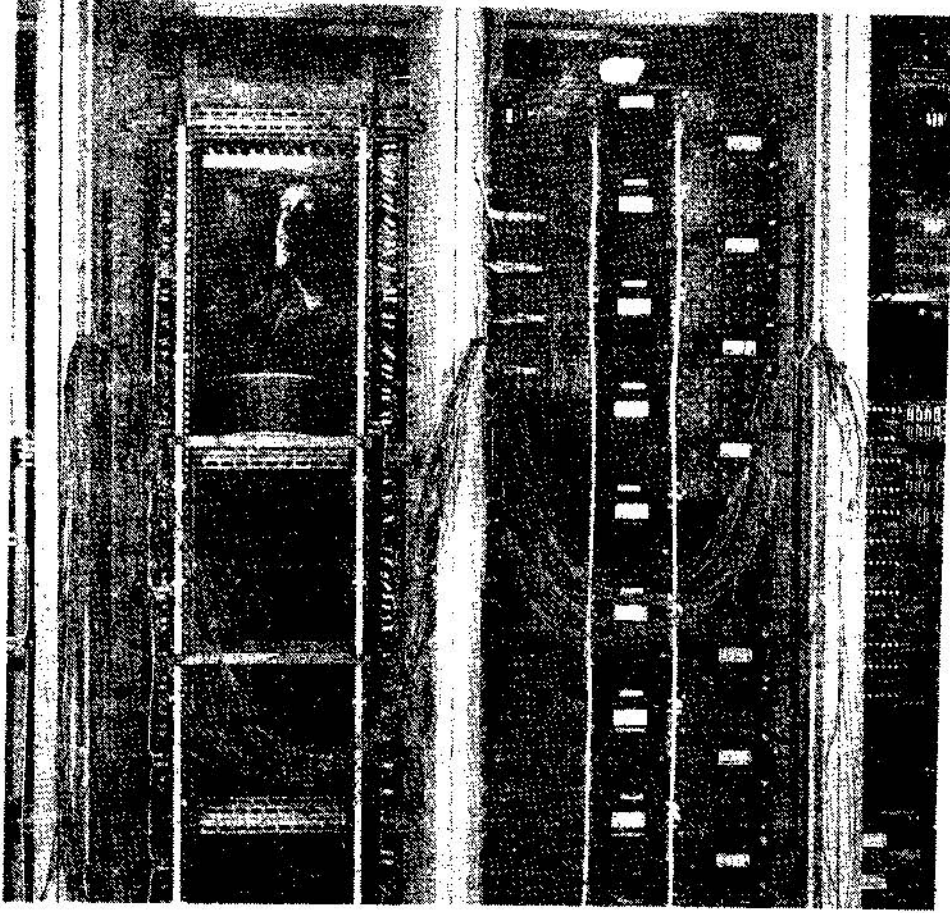
"Basitliğin" gözönüne alınması gereken başlıca özellik olduğunu söylemesine karşılık, "çok-yol" taraftarları ürünlerinin belirli bir zaman diliminde CPU'ya daha fazla bilgi gönderebileceğini söylemektedir.

Belleğe erişme ile ilişkili zamanlama bilgisi kolayca farkedilemez. Bir DSP üzerinde test programları koşturmaya imkan tanıyan simülatör, emülatör veya bir değerlendirme kartı kullanılırken bu nokta gözden uzak tutulmamalıdır. Örneğin, DSP gerçekte veri ve komutlarını yavaş bir dış bellekten elde ediyordur, buna karşılık siz programınızın yonganın üst limit hızında koşturduğunu varsayıyor olabilirsiniz.

Eğer yazılım ve donanım, arabağlaşım devreleri hedeflenmiş olan sisteminkinden farklı bir ortamda test ediliyorsa, söz konusu test ortamında geçerli olan zamanlama sınırlamalarının bilinmesi gerekir.

CPU'lara hızlı olarak veri ve komutlar gönderen yüksek hızlı belleklere ek olarak, günümüzdeki DSP'lerin birçoğu, yüksek işlem hızlarına çıkabilmek için "Pipeline" kavramından yararlanır. Fakat "Pipeline" kavramı sadece DSP'ler için kullanılmamaktadır; tüm yüksek hızlı bilgisayarlarda bu kavrama başvurulmaktadır. "Pipeline" kavramı, CPU'ya gönderilen komutların, CPU'ya vardıklarında kısmen kodçözülmüş ve CPU tarafından yürütülmek üzere hazır durumda bulunmasını garanti eder. Bununla beraber, "Pipeline", bir komutun yonga tarafından alınması ile o komutu yürütmesi arasında gecikme yarattığından, pipeline ile başedebilmek programcılar için zaman zaman sorunlar yaratmaktadır.





bölge sunmasına ve yuvarlatma hatalarını azaltmasına karşılık, sabit-nokta aritmetiği de, birçok DSP problemi çözebilecek yeterlilikte dinamik bölge sunmaktadır. Fakat sabit-nokta ile aritmetik işlemlerin bazı sakıncaları vardır. Örneğin, iki 16-Bit'lik değerın çarpma neticesi 32-Bitlik bir değerdir. Bu değer bilgisayarda saklanmadan önce 16-Bit olarak ya yuvarlatılmalı ya da kesilmelidir. Bu yöntem alternatif olarak, çarpma sonucunda oluşan 32-Bit'lik ara netice, daha sonraki bir matematiksel işlemde kullanılmak üzere, geçici olarak bir biriktiricide (akümülatör) saklanır. Ara neticeleri yonga içindeki kaydedicilerde saklamakla, yuvarlatma işlemi nedeniyle doğabilecek bazı sorunlardan kaçınılmış olur. Böylece her bir ara neticeyi yu-

Her ne kadar, "Pipelinelar ve bellek mimari yapıları DSP'ye sürdürülen işlemleri hızlandırmakta yararlı ise de, bu yongalar giriş/çıkış yongaları ve bazen de ana bilgisayarlar ile haberleşmeden veri üzerinde işlemler yapamazlar.

Bir takım DSP ürünleri yalnızca dış giriş/çıkış kapılarına¹²¹ bel bağlamışken, diğerleri içlerindeki paralel ve seri yolların birleşimlerine dayanır.

DSP yongaları, G/Ç birimlerine giden ve G/Ç birimlerinden gelen bilgi akışını, yürüttükleri işaret-işleme görevleri ile eşzamanlı kılmalı veya koordine etmelidir. Mevcut DSP yongalarından birçoğu, dışardan bağlanan birimlerin hizmet isteklerine izin veren "kesme" işaretleri sağlar.

Bu "kesmeye - hizmet verme" mimari yapısı ilk mikroişlemcilerin mimari yapısını hatırlatmaktadır. Yongaların birçoğunda, "kesmeler" üzerinde işlem yapıldıktan sonra, dönüş adreslerinin saklandığı sınırlı sayıda iç yığıt¹³¹ yerleri mevcuttur, içerde bulunan yığıtın dışarda (bellek üzerinde) yer alan yığıttan üstünlüğü vardır; içteki yığıt daha hızlıdır.

Kesmeler, bellek mimari yapıları ve G/Ç kontrolü, DSP ürününün seçiminde önemli etkenlerdir; fakat aynı zamanda veri üzerinde nasıl işlem yapılması gerektiğine de karar verilmelidir. Yani, tam sayı aritmetiği mi^(U), yoksa kayan-nokta aritmetiği mi¹⁵¹ kullanılacaktır?

Her ne kadar kayan-nokta aritmetiği geniş bir dinamik

varlatacak yerde, ardışık işlemlerin en sonunda oluşan netice yuvarlatılır.

Yuvarlatma işlemi sonsuz impuls yanıtı (IIR)¹⁶¹ filtrelerde ve geribesleme kullanılmış veya yinelemeli¹⁷¹ sistemlerde kararsızlıklara yol açabilir. Yuvarlatma işlemi özellikle küçük sayılarla çalışırken önem kazanır; çünkü küçük bir değerde göreceli olarak son derece büyük değişiklikler oluşabilir. Örneğin 2.51 değerini 3 olarak yuvarlatmak %16'lık bir hata getirir ve çok küçük sayıların yuvarlatılması sıfır neticesini yaratabilir. Bir değerın, çok küçük bir sayı ile veya sıfır ile bölünmesi arasında büyük farklar vardır. Kayan-nokta ile aritmetik yapan bilgisayarlarda ise, yuvarlatma hatası, yuvarlatılan sayının mutlak değerine bağlı değildir.

Unutulmamalıdır ki, kayan-nokta işlemleri sabit-nokta işlemlerinden daha uzun yer işgal eder. CPU, neticeleri normalize hale getirmenin yanında aynı zamanda diğer kontrol işlemlerini de yürütmelidir. Uzunca bir program, kayan-nokta aritmetiği nedeniyle ortaya çıkan ilave süre, hızla birbirine eklenerek artar. Ayrıca, A/D, D/A çeviriciler gibi birçok çevre birimi elemanları tam sayılarla çalışır ve aradıkları ayırma duyarlılığı daha fazla değildir. Bu yüzden belirli bir uygulamada gereğinden fazla duyarlık derecesi aranmaya özen gösterilmelidir. O halde bir matematiksel format seçmeden önce yapılması gereken iş, başlangıç ara ve son değerler açısından gerekecek duyarlık derecesinin gözden geçirilmesidir.

DSP üreticileri, mikroişlemcilerdeki ne benzer biçimde, DSP ürünlerinin veri işleme yeteneklerini gösterebilmek amacı ile, "DSP performans düzeyi programları" sağlarlar. Bu programlar sayesinde tasarımcı bir yonga üzerinde yürüttüğü işlemleri diğerlerinde de yaptırarak, karşılaştırma olanağı bulur. Fakat unutulmaması gereken nokta performans düzeyi programlarının öznel kaldığıdır.

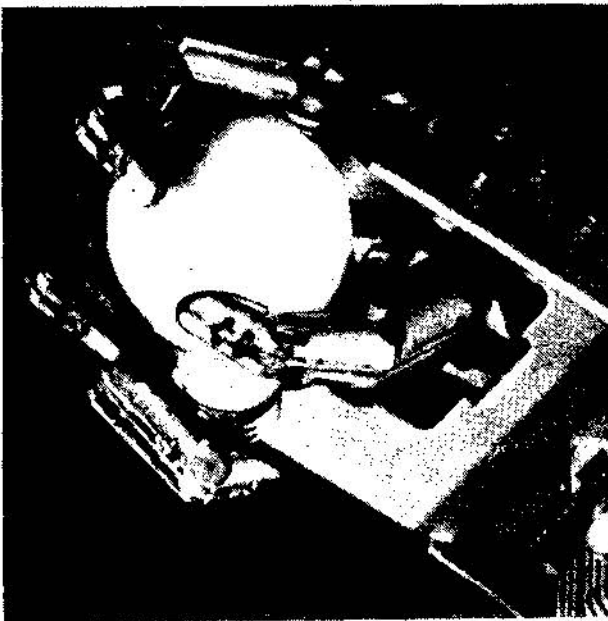


"DSP yongalannu derhal akla gelen uygıdamalan, sayısal işaret işleme konulan ile uğraşanların çok iyi bildiği genel amaçlı birtakım DSP işlevlerini gerçekleştirir."

Örneğin, belirli bir ürünün yüksek hızda çalıştığına dair bir iddia, tüm verinin DSP içindeki RAM'da yerleşmiş olduğu bazı özel durumlara uygulanabilir. Bu nedenle, hız ile ilgili spek'ler ve performans düzeyi değerlerini biraz temkinli değerlendirmek gerekir.

Hızlı Fourier dönüşümü (FFT) çok yaygın kullanılan bir performans düzeyi programıdır ve üreticilerin birçoğu en azından bir tane FFT programı için yürütme hızı değerini verir. Her ne kadar birçok verinin bir yerden diğerine hareket ettirilmesi ve üzerinde işlem yapılması nedeniyle, bir FFT programı iyi bir performans düzeyi programı gibi gözükmeye karşılık, ne tüm FFT'ı programları, nede verilen FFT performans düzeyi değerleri aynıdır. Örneğin FFT programı 64 tane, 1024 veya ara bir değerde veri noktası üzerinde uygulanabilir. FFT algoritması, radix-2 veya radix-4 işlemleri kullanılıyor olabileceği gibi, program ya doğrusal olarak ya da çevrim oluşturulacak şekilde oluşturuluyor olabilir. Ayrıca, performans düzeyi değerleri karmaşık bir problemin sadece bir parçasını yansıtabilir. Örneğin, grafik veri işlemede, "pixel (resim elemanları)" konvolüsyon işlemleri pixel başına 9 adet çarpma işlemi gerektirir. Halbuki, olayı gösteren tipik programlar, adreslerin nasıl ele alındığı göstermeyeceği gibi, "görüntü - sınır koşullarını" gözönüne alan özel işlemleri de belirtmeyecektir.

DSP ürünlerinin seçiminde gözönünde bulundurulması



gereken diğer noktalar:

- Üreticinin gerçekten, ilan edildiği üzere DSP yongalarını üretip üretmediğinden emin olunuz (elde edilebilirlik).
- Seçilen yonganın ikinci bir kaynak (üretici) tarafından sağlanıyor olması, kullanıcı da güven yaratacaktır.
- Uygulama-mühendisliği desteğinin, uygulama bilgisinin, teknik bilgilerinin olup olmadığını kontrol ediniz.
- Ürünün tanıtımına yönelik bir seminer programının ve/veya uygulamalı kurslarının (workshop), olup olmadığını öğreniniz.
- Yazılım, değerlendirme kartı, geliştirme sistemi, gibi destek yordamlarının varlığını araştırınız.

şeklinde sıralanabilir.

DSP UYGULAMALARI

DSP yongalarının derhal akla gelen uygulamaları, sayısal işaret işleme konuları ile uğraşanların çok iyi bildiği genel amaçlı birtakım DSP işlevlerini gerçekleştirir. Bu işlevlerden en çok kullanılanları:

- (i) Sayısal süzme (filtreleme)
- (ii) Konvolüsyon
- (iii) Korelasyon (ilintililik)
- (iv) Hilbert Dönüştürücüleri
- (v) Hızlı Fourier Dönüşümleri (FFT)
- (vi) Adaptif (uyarlamalı) süzme
- (vii) Pencere fonksiyonları ile çarpma
- (viii) Dalgaşekli üretilmesi

SAYISAL FİLTRELER, KONVOLÜSYON, KORELASYON

Frekans seçimi özellikleri gösteren herhangi bir sistem veya devre filtre (süzgeç) olarak nitelenebilir, örneğin birçok uygulamada, bir işarete ilişkin frekans spektrumunun değişikliğe uğratılması, tekrar biçim verilmesi veya istenilen spek'lere uyacak şekilde elden geçirilmesi istenebilir. Bir işareti bozan gürültünün giderilmesi, iletim ortamından kaynaklanan işaret distorsiyonunun giderilmesi, bir iletim kanalından en fazla yararlanabilmek amacı ile, karıştırılmış 2 işaretin birbirinden ayrılması, işaretlerin demodülasyonu ve ayrık zaman işaretlerinin sürekli zaman işaretlerine çevrilmesi, sayısal görüntü işleme, patern (örüntü) tanıma ve spektrum analizi, sayılabilecek diğer bazı filtre kullanım alanlarıdır. Sayısal filtreler analog

benzerlerine göre : (i) Daha yüksek güvenilirlik (ii) Daha fazla doğruluk (iii) Eleman değerlerinin zamanla kayması nedeniyle sistem performansının etkilenmesi (iv) Eleman toleranslarının kritik olmaması gibi nedenlerle üstünlük sağlarlar. Ayrıca TMS320 gibi programlanabilir bir DSP işlemcisinin kullanılması ile, filtre karakteristiklerini değiştirmek için gerekli filtre parametre değişiklikleri kolayca gerçekleştirilebilir. Sayısal filtre tasarımlarında çoğu kez şu 4 adımdan geçilir:

a) Yaklaşıklık yapılması

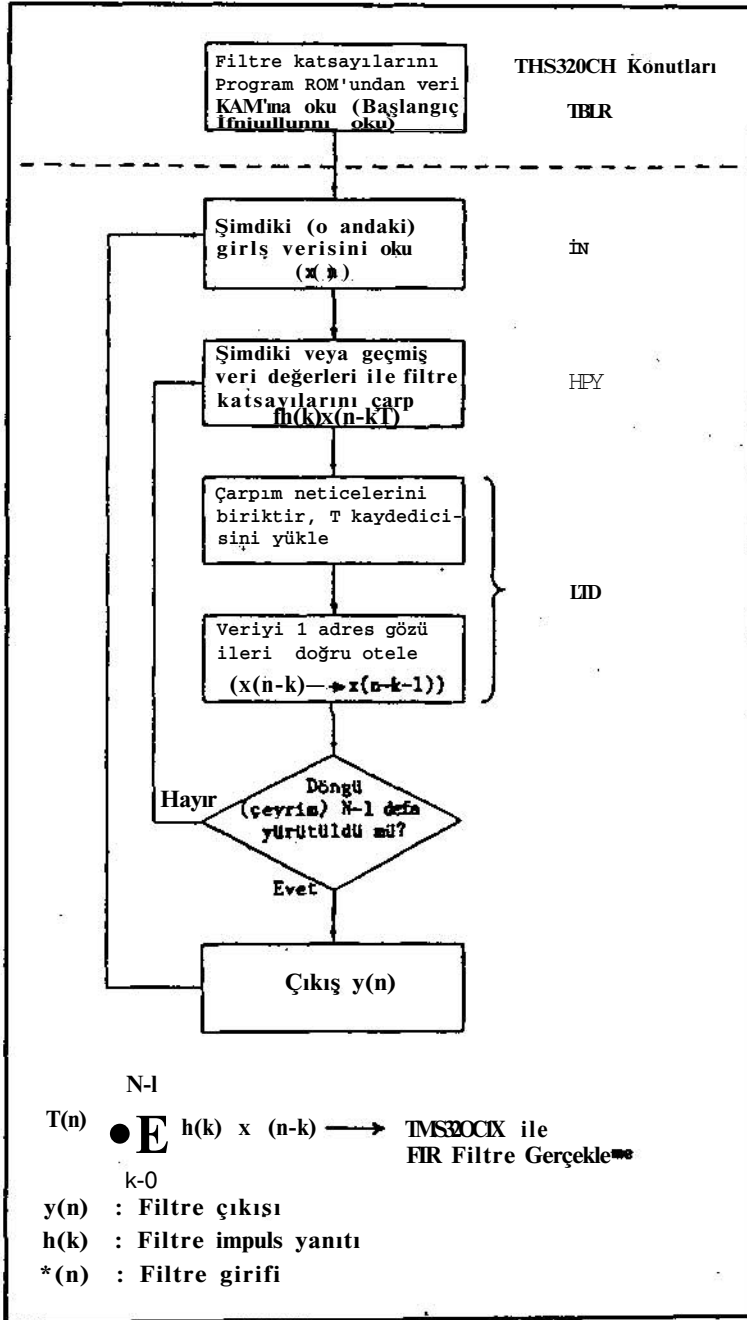
b) Gerçekleşmeye yönelik sayısal filtre yapılarının seçilmesi

c) Sonlu aritmetik duyarlık nedeniyle aritmetik hataların etkisinin incelenmesi

d) Pratik gerçekleştirme

Yaklaşık yapılmasından kasıt, istenilen spek'leri sağlayan bir transfer fonksiyonunun oluşturulmasıdır. Bu işlem sırasında, filtrenin hem zaman hem de frekans yanıtlarından herikisi veya teki için içine girebilir.

Gerçekleşmeye yönelik filtre yapıları seçerken genellikle ya.



$$y(n) = \sum_{k=0}^M b_k X(n-k)$$

Şekilde verilen sonlu impuls Yanıtlı (FIR) yada,

$$y(n) = \sum_{k=0}^M a_k X(n-k) + \sum_{i=1}^N a_i y(n-i)$$

şeklinde verilen Sonsuz İmpuls Yanıtlı (IIR) filtreler seçilir.

Burada amacımız, sayısal filtre teorisini anlatmak olmayıp; TMS320 DSP ailesinin hangi tür uygulamalarda uygun olduğunu belirtmek; mesaj ve fikirler vermektir.

Anlaşılabileceği üzere, yukarıdaki denklemler birçok defa yinelenen çarpma ve toplama (biriktirme) işlemleri gerektirir. TMS320, yüksek hızı ve özel birtakım DSP'ye yönelik komutları ile, bu tür denklemleri hızla sıralayabilir ve diğer ek birtakım işlerin daha yapılabilmesi için yeterli süreyi bırakır.

Şekil -2'de TMS320C1X ile bir FIR gerçekleştirilmesi halinde, geçilecek başlıca adımlar gösterilmiştir. Başlangıçta, ROM'da tutulan h(k) katsayıları TBLR (tablo okuma) komutu yardımı ile RAM'a aktarılmalıdır. Sonra, o andaki x(n) giriş değeri ve RAM'ına okunur ve peşpeşe, çarpma, biriktirme ve veri öteleme işleri gerçekleştirilir. Veri öteleme işlemi, z⁻¹ (birim gecikme) işlemine karşı düşmek üzere, x(n) değerlerinin yerini ayarlar ve böylece bir sonraki y(n) değerinin hesaplanabilmesine olanak tanır.

Katsayıları TBLR komutu yardımı ile RAM'a okutmaya alternatif bir yöntem, MPYK (ivedi çarpma) komutunun kullanılmasıdır. MPYK komutu, çarpılacak değer-

Şekil 2:

lerden birisini (13-Bit'lik bir sabit), komut kodu içinde içerir. Bu yaklaşım hem süreden hem de ROM alanından tasarruf sağlar. Buna karşılık, aritmetik duyarlıktan (presiyon) kaybedilir.

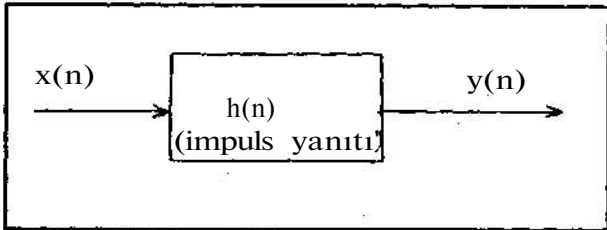
DSP'de sıkça kullanılan KORELASYON işlemi de filtreleme işlemlerine benzer. Korelasyon; 2 sinyal arasındaki benzerliğin bir ölçüsü olarak, örneğin rastgele bir gürültü içinde kaybolmuş zayıf bir sinyali elde etmekte kullanılabilir. En genel durumda, $x(n)$ ve $y(n)$ gibi 2 fonksiyon arasındaki korelasyon (çapraz korelasyon),

$$R_{xy}(m) = \frac{1}{N} \sum_{n=0}^{N-1} x(n)y(n+m)$$

şeklinde tanımlanır. Benzer şekilde, tek bir $x(n)$ dalgaşekli için de korelasyon (ortakorelasyon) fonksiyonu tanımlanabilir.

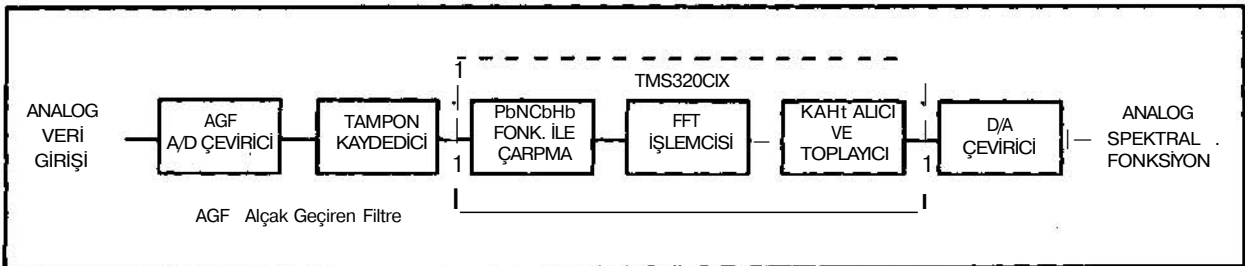
$$R_{xx}(m) = \frac{1}{N} \sum_{n=0}^{N-1} x(n).x(n+m)$$

Korelasyon işlemleri örneğin sonar sinyali işlemede, hedef tayin etmek için kullanılır ve 2 frekanslı bir korelasyon fonksiyonu şeklinde kendini gösterir. Jeofizik uygulamalarında, sismik dalga yayılımını analiz etmek için de genetik korelasyon katsayılarına başvurulur.



Şekil 3 : Lineer zamanla değişmeyen sistem

Konvolüsyon işlemi de yukarıda yazılan denklemlere benzer bir karakter gösterir. Konvolüsyon; lineer, zamanla değişmeyen sistemleri tanımlamak için başvurulan bir işlemdir (Şekil -3) ve



Şekil 4: FFTye dayalı Spektrum Analizi



$$y(n) = \sum_{m=-M}^N h(m)x(n-m)$$

bağıntısı ile tanımlanır. Anlaşılacağı üzere, yukarıda sözü edilen tüm işlemler, TMS320 gibi DSP mimarileri ile etkin bir biçimde yürütülebilir.

SPEKTRUM ANALİZİ, FFT, PENCERE FONKSİYONLARI İLE ÇARPMA

Spektrum analizi; tıp, sonar ve radar gibi uygulamalarda kullanılmaktadır. Bu tür uygulamalarda; alışlagelmış analog yöntemler yerine sayısal tekniklerin tercih edilmesinin bazı nedenleri:

- Gürültü olması halinde bile daha ince frekans çözümü yapılabilmesi
 - Daha büyük dinamik bölge sağlanabilmesi
- olmuştur. Aslında, sismik işaret işleme halinde olduğu gibi, çok alçak frekansların (1 Hz'den küçük) sözkonusu olduğu uygulamalar, sadece sayısal sistemlerle gerçekleştirilebilir.

FFT hesaplama yöntemine dayalı spektrum analizi, Şekil -4'deki bloklardan oluşan bir sisteme gerek duyar. Şekil -4'den anlaşılacağı üzere, TMS320C1X, spektrum analizi

zinin can damarı olan, yoğun hesaplama gerektiren işlemleri yürütür. Pencere fonksiyonu ile çarpma; bir veri dizisindeki (çerçevesindeki) her bir veri noktasını bir ağırlaştırma katsayısı¹⁹ ile çarpmaya karşı düşer. Pencere fonksiyonu ile çarpma işlemi, FFT hesaplama tekniğinin sonlu uzunlukta bir veri çerçevesi üzerine uygulanması nedeniyle yapılmaktadır. Çünkü; anlamlı sonuçlar verebilmesi için, teorik olarak FFT'nin sonsuz uzunlukta bir çerçeve üzerinde hesaplanması gerekir. Halbuki FFT hesapları sonsuz uzunlukta çerçevelerle yapılmaktadır ve eğer pencere fonksiyonları ile gerçekleştirilmezse, çok fazla distorsiyon oluşur ve spektrum yanlış ölçülmüş olur. Spektrum analizinin en karmaşık bloğu FFT işlemcisidir. FFT hesaplarında, en genel durumda her veri noktası için hem reel hem de sanal bileşene ilişkin hesaplama yapılmaktadır. Tek yonga modunda (yonga üzerindeki ROM'dan çalışma) TMS320C1X ürünleri, 64-sanal nokta için (a+jb) FFT hesabını 590 us'de (200 ns'lik çevrim süresi için) tamamlamaktadır. 64'den daha fazla nokta için; yonga dışında yer alan RAM'dan çalışarak, FFT hesaplanabilir. FFT sonuçları sanal sayılar çıkardığından, FFT bloğunun çıkışındaki verinin karesi alınır, toplanır ve D/A'den geçirilerek analog spektral fonksiyonu elde edilir.

Günümüzde DSP yongaları birçok alanda uygulama bulmuş olup, bu yazımızda bunların tümünden söz etmek mümkün olmayacaktır. Tablo-2 TMS320 yongalarının günümüzde uygulandığı alanları özetlemektedir.

- (1) Digital Signal Processing (DSP)
- (2) Proportional Integral Derivative (PID)
- (3) Windowing
- (4) Fast Fourier Transform (FFT)
- (5) Central Processing Unit (CPU)
- (6) Resolution
- (7) Truncation Error
- (8) Limit Cycles
- (9) Benchmark
- (10) Interrupt
- (11) Bus
- (12) Port
- (13) Stack
- (14) Integer Math
- (15) Floating-Point Math
- (16) IIR (Infinite Impulse Response)
- (17) Recursive
- (18) Image Boudry Conditions
- (19) Weighting Coefficient

TMS 320 Yongalarının Günümüzde Uygulandığı Alanlar

GENEL AMAÇLI DSP

- Sayısal Filtreler
- Konvolüsyon
- Nİbert Dönüştürücüler
- Adaptif Filtreler
- Pencere Fonksiyonları İle Çarpma (Windowing)
- Dalgaşekli Üretim

ENSTRÜMANTASYON

- Speknm Analı
- Fonksiyon Üretici
- Patem Uyduma
- Sismik İşleme
- Geçia hal Analizi
- Sayısal Süzme
- Sayısal Faz Kenetlenen Çevnm (PLL) oluşuma

KONTROL

- DskDenetmi
- Servo Denetimi
- Robot Deneten
- Lazer/Yazıcı Denetimi
- Araç, Taşıt Motoru Denetimi

TELEKOMÜNİKASYON

- Eko Giderme
- EDPCM
- Sayısal Pb»
- Kanal Çoklama (Multiplexer)
- 19200 Bitse Kadar Modemler
- Adaptı Dengeleyiciler
- DTMF Kodamatodçözme)

OTOMOTİV

- Araç/taşıt Motoru Denetmi
- Titreşim Analizi
- Patinaj Yapmayı Önleyen Frenler (Antskid Brakes)
- Adaptı Brnş Denetmi
- Klavuzluk Sistemleri
- Ses Komutan
- Ses Frekanslı (Audio) İşaretlerin İşlenmesi

ENDÜSTRİYEL

- Robotk
- Nümerik Kontrol
- Güvenlik Erişimi
- Güc Hatlarının Gözlenmesi
- Motor Kontrott
- Programlanabilir Deneteyicier

GRAFIK/GORUNTU

- 3-Boyutu Döndürme
- Robotk Göm» Sisten»
- Görüntü Dönüşümü/sıkıştırma
- Patem Tanıma
- Görüntü İyileştirme
- Homomorphc İşleme
- İş İstasyonları (Works Statons)
- Sayısal Haritalar

KONUŞMA/SES

- Konuşma Postası (Voice Mail)
- Vocoder
- Konuşma Tanıma
- Konuşucu Yoklama/doğrulama
- Konuşma İ/ileştirme
- Konuşma Sentezi
- Metnden Konuşmaya Geçiş (Text-to-speech)

ASKERİ ELEKTRONİK

- Gizli İletişim
- Radar İşleme
- Sonar İşleme
- Görüntü İşleme
- Klavuzluk Sistemleri (Naviganon)
- Güdümlü Mermi Yönlendirme
- RF Modemleri

TELEKOMÜNİKASYON

- Veri Şifreleme
- Fax
- Hücresel Telefonlar (CeMar Telephone)
- Sayısal Konuşma İnterpolasyonu (Speech InterpocJaton)
- X 25 Paket Anahtarlama
- Video Konlerans Sistemleri
- Yayılmış Spektrum Tekniği İle İletişim (Spread Spectrum)

TÜKETİCİ EKTRONİĞİ

- Radar Detektörü
- Motorlu Aygıtlar
- Sayısal AudioTV
- Müz* Sentezleyia
- Eğitsel Oyuncaklar

TIBBİ ELEKTRONİK

- İşitme Cihazları
- Hasta Gözlemeleme/monitörleri
- Ultrasonik Cihazlar
- Cerm Gözlemeleme/monitörleri
- Tanı(Diagnostk)aygıtları
- Protez Sistemlen

Tablo 2