

YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

106256

SAYISAL İŞARET İŞLEYİCİ YONGALARIN YAPISI
VE MOTOR KONTROLÜ KONUSUNDAKİ
UYGULAMALARIN İNCELENMESİ

Elektrik Müh. İzzet Yılmaz ÖNEL

F.B.E Elektrik Mühendisliği Anabilim Dalında
Hazırlanan

Doç. Dr. İbrahim ŞENOL ^{is} YÜKSEK LİSANS TEZİ

Prof. Oktay AYBAR ^{AA}

Doç. Dr. Herman SEDEF. ^{SeDEF.}

Tez Danışmanı: Doç. Dr. İbrahim ŞENOL

106256

T.C. YÜKSEKÖĞRETİM KURULU
DOKÜMANTASYON MERKEZİ

İSTANBUL, 2001

İÇİNDEKİLER

	Sayfa
SİMGE LİSTESİ	v
KISALTMA LİSTESİ	vi
ŞEKİL LİSTESİ.....	vii
ÇİZELGE LİSTESİ	ix
ÖNSÖZ	x
ÖZET	xi
ABSTRACT	xii
1. GİRİŞ	13
1.1 Dijital Denetleyici Yapılarının Analog Denetleyici Yapıları ile Karşılaştırılması	16
2. MİKROİŞLEMCİLER VE KONTROL SİSTEMLERİNDEKİ UYGULAMALARI.....	20
2.1 Mikroişlemcilerin Yapısı	20
2.2 Mikroişlemcilerin Kontrol Sistemlerinde Uygulamaları	25
2.2.1 Donanım Tasarımı	27
2.2.2 Yazılım Tasarımı	30
3. DİJİTAL İŞARET İŞLEMENİN TEMELLERİ.....	32
3.2 Dijital Yapılarda Sayı Formatı.....	33
3.2.1 Sabit Nokta ve Kayan Nokta Aritmetiği.....	36
3.2.2 Sabit Nokta Aritmetiği.....	36
3.2.3 Kayan Nokta Aritmetiği.....	37
3.3 Sonlu Kelime Genişliği Etkileri.....	39
3.3.1 Taşma.....	41
3.3.2 Yuvarlatma ve Kesme.....	42
3.3.3 Ölçeklendirme.....	44
3.3.4 Analog Dijital Dönüştürücüler.....	45
3.3.4.1 Kuantalama	46
3.3.4.2 Örnekleme Teoremi	50
3.3.4.3 Dijital Analog Dönüşüm	56
3.3.4.4 Veri Dönüşümü İçin Analog Filtreler	59
4. DSP GENEL YAPISI	65
4.1 DSP Genel Özellikleri	65
4.2 DSP Mimarisini Oluşturan Üniteler	71
4.2.1 Çarpım Ünitesi	71

4.2.2	Çarpım - Akümülatör Ünitesi (MAC)	72
4.2.3	Aritmetik- Lojik İşlem Ünitesi (ALU).....	73
4.2.4	Yığın Kaydırma Ünitesi (BS)	75
4.2.5	Veri Adres Generatörü Ünitesi (DAG).....	77
4.2.6	Komut Sıralama Ünitesi (SEQ)	78
5.	ÖRNEK DSP YONGALARIN KARŞILAŞTIRILMASI	82
5.1	TMS320C240 DSP Entegresinin Mimarisi	97
5.1.1	Durum ve Kontrol Kütükleri.....	97
5.1.2	Merkezi İşlem Birimi	99
5.1.2.1	Giriş Ölçekleme Kaydırıcısı	100
5.1.2.2	Çarpım Ünitesi.....	100
5.1.2.3	Akümlatör	103
5.1.2.4	Yardımcı Kütük ve Yardımcı Kütük Aritmetik Ünitesi	104
5.1.3	Dahili Hafıza	105
5.1.3.1	Veri/Program İki Çeşit İşlem Yapabilen RAM (DARAM).....	105
5.1.3.2	Sadece Okunabilir Program Hafızası (ROM).....	105
5.1.4	TMS320C240'ın Yapısında Dahili Olarak Bulunan Çevresel Birimler	106
5.1.4.1	Harici Hafıza Arayüzü	106
5.1.4.2	Durum Yöneticisi Modülü	107
5.1.4.2.1	Genel Amaçlı (GP) Zamanlayıcılar	107
5.1.4.2.2	Tam Karşılaştırma Birimleri.....	109
5.1.4.3	Programlanabilir DeadBand (ÖlüBand) Üretici.....	109
5.1.4.4	Basit Karşılaştırıcılar	109
5.1.4.5	Karşılaştırma / PWM Dalga Şekli Üretimi	109
5.1.4.5.1	Karşılaştırma / PWM Dalga Şekli Üretme Biriminin Karakteristikleri.....	109
5.1.4.6	Örnekleme Birimi	110
5.1.4.7	Kod Çözücü Darbe Devresi	111
5.1.4.8	Analog Dijital Dönüştürücü Modülü (ADC)	111
5.1.5	Seri Çevresel Arayüz Modülü (SPI)	112
5.1.6	Seri Haberleşme Arayüz Modülü (SCI).....	114
5.1.7	Watchdog ve Eşzamanlı Kesme Modülü.....	116
6.	DSP ve MİKROİŞLEMCİLERİN KARŞILAŞTIRILMASI	119
7.	DSP UYGULAMA ALANLARI	126
8.	SONUÇLAR VE UYGULAMA ÖRNEĞİ	140
8.1	DSP Tabanlı Kontrolör Örneği : Bir D.C. Motorun Hız Kontrolünün TMS320C50 Tabanlı Bir Kontrol Sistemi Kullanılarak Gerçekleştirilmesi	142
KAYNAKLAR.....		147
EKLER		148
EK 1	TMS320C240 YONGASININ TERMİNAL AÇIKLAMALARI.....	149

EK 2 : TMS320C240 YONGASININ TERMİNAL YERLEŐİMİ VE BLOK DİYAGRAMI167

ÖZGEÇMİŐ..... 170



SİMGE LİSTESİ

A,B,C,D	Durum matrisleri
b	Ondalık noktanın yeri
b_{m-1}	İşaret bilgisi
e(t)	Hata terimi
f	İşaretin frekansı
f_0	Örnekleme frekansı
g	Referans ağırlık terimi
I(t)	Entegratör çıkış işareti
j	İkili basamaklar (bitler)
K_c	Kazanç değeri
m	Toplam kelime genişliği
N	Maksimum türev kazancı
r	İkinin tümleyeni sayısı
R^{ph}	Fiziksel sistemin sayı aralığı
R^{Pr}	İşlemcinin sayı aralığı
S_u	Kontrol işareti ölçekleme matrisi
S_y	Çıkış işareti ölçekleme matrisi
T_d	Türev zaman sabiti
T_j	Entegral zaman sabiti
U	Kontrol işareti
V(k)	k. gürültü değeri
X	Giriş işareti
Y	Çıkış işareti
Y(k)	Gerçek k. değer
y_r	Referans değeri
Z(k)	Ölçülen k. değer

KISALTMA LİSTESİ

ADC	Analog Digital Converter
ALU	Arithmetic Logic Unit
ARM	Anahtarlamalı Relüktans Motoru
DAG	Data Adress Genetator
DARAM	Dual Access RAM
DSP	Digital Signal Processing
EPROM	Erasable Programmable ROM
EEPROM	Electrically Erasable Programmable ROM
EV	Event Manager
MİB	Merkezi İşlem Birimi
PS	Program Sayacı
PWM	Pulse Width Modulation
RAM	Random Access Memory
ROM	Read Only Memory
RTI	Real Time Interrupt
SP	Stack Pointer
SPI	Serial Peripheral Interface
SCI	Serial Communications Interface
QEP	Quadrature Encoder Pulse
WD	Watch Dog

ŞEKİL LİSTESİ

Şekil 1.1	Analog PID denetleyici yapısı.....	17
Şekil 1.2	Dijital denetleyici yapısı.....	19
Şekil 2.1	Mikrodenetleyicinin yapısı.....	21
Şekil 2.2	Mikroişlemcinin alt birimleri.....	23
Şekil 3.1	Sayısallaştırma işlemi esnasındaki dalga şekilleri.....	48
Şekil 3.2	Dithering ya da titreşim etkisi olarak ifade edilen durumun aşamaları.....	49
Şekil 3.3	Çeşitli dalga şekilleri için yapılan örnekleme.....	51
Şekil 3.4	Örnekleme sırasında analog frekansın dijitale dönüşümü.....	53
Şekil 3.5	Zaman ve frekans domenlerinde örnekleme teoreminin ifadesi.....	55
Şekil 3.6	Dijital – analog dönüşümün analizi.....	58
Şekil 3.7	Basit bir dijital işaret işleme sisteminin blok diyagramı.....	59
Şekil 3.8	Sallen – Key filtresinin yapısı.....	60
Şekil 3.9	Altı kutuplu bir Bessel filtresinin yapısı.....	61
Şekil 3.10	Alçak geçiren RC filtresi ve anahtarlamalı kondansatör filtresi.....	62
Şekil 3.11	Çeşitli filtrelerin lineer ve logaritmik ölçekle çizilmiş frekans cevapları.....	64
Şekil 3.12	Çeşitli filtrelerin lineer ve logaritmik ölçekle çizilmiş frekans cevapları.....	64
Şekil 4.1	Üç Seviyeli iş hattı işlemesi.....	68
Şekil 4.2	MAC 'ın basit yapısı.....	73
Şekil 4.3	ALU 'nin basit yapısı.....	74
Şekil 4.4	Normalize ve denormalize işlemleri.....	76
Şekil 4.5	DAG ' ın basit yapısı.....	78
Şekil 4.6	Sıralama ünitesi ile diğer ünitelerin ilişkisi.....	80
Şekil 5.1	TMS32010 fonksiyon blok diyagramı.....	87
Şekil 5.2	TMS320C25 fonksiyon blok diyagramı.....	90
Şekil 5.3	ADSP-2101 fonksiyon blok diyagramı.....	92
Şekil 5.4	TMS320C30 fonksiyon blok diyagramı.....	96
Şekil 5.5	Durum ve kontrol kütükleri düzeni.....	97
Şekil 5.6	Durum yöneticisinin (Event Manager) blok diyagramı.....	108
Şekil 5.7	Analog dijital dönüştürücü modülü.....	112
Şekil 5.8	4 bacaklı Seri çevresel arayüz birimi blok diyagramı.....	114
Şekil 5.9	Seri haberleşme arayüzü (SCI) modülü blok diyagramı.....	116
Şekil 5.10	WD/RTI modülü blok diyagramı.....	118
Şekil 7.1	Motor kontrol sisteminin birimleri.....	127
Şekil 7.2	Örnek bir kontrol algoritmasına ilişkin akış diyagramı.....	129
Şekil 7.3	Geliştirilmiş PID yapısı.....	133
Şekil 7.4	Gözlemleyici ve durum geri besleme kontrolü.....	137
Şekil 7.5	Birinci dereceden kontrol edilen sistem ve modeli.....	138
Şekil 8.1	TMDS3200051 DSP starter kit'i blok diyagramı.....	143
Şekil 8.2	TMDS3200051 DSP starter kit'i yerleşim planı.....	143
Şekil 8.3	Ana akım devresi.....	144
Şekil 8.4	İlgili referans değerlere karşılık gelen hız bilgileri.....	146
Şekil EK2.1	TMS320C240 yongasının terminal bağlantı uçları.....	168
Şekil EK2.2	TMS320C240 yongasının blok diyagramı.....	169

ÇİZELGE LİSTESİ

Çizelge 1.1	Analog ve dijital denetleyicilerin karşılaştırılması	18
Çizelge 3.1	İkinin tümleyeni sayı formatı gösterimi	34
Çizelge 3.2	Sayı formatları.....	35
Çizelge 3.3	Sabit ve kayan nokta sayı formatı özellikleri.....	38
Çizelge 3.4	Yuvarlatma ve taşmanın etkileri	42
Çizelge 3.5	Kesme ve yuvarlatma karakteristiği.....	43
Çizelge 3.6	Çeşitli filtrelerin k_1 ve k_2 sabitleri	62
Çizelge 4.1	DSP ve mikroişlemcinin komut işleme sayıları.....	69
Çizelge 5.1	Değişik DSP sistemlerinin genel özellikleri	85
Çizelge 5.2	PM bitlerinin durumuna göre yapılacak işlemlerin listesi.....	98
Çizelge 5.3	PM bitlerinin durumuna göre yapılacak kaydırma miktarı	101
Çizelge 6.1	Çeşitli DSP ve mikroişlemcilerin işlem hızlarının karşılaştırılması	120
Çizelge 6.2	Çeşitli DSP yongaların bellek ve adres özellikleri.....	123
Çizelge 8.1	Referans giriş ile hız ilişki tablosu	145



ÖNSÖZ

Yapmış olduğum bu tez çalışması esnasında araştırmalarımı yönlendirmem konusunda bana yol gösteren danışmanım Doç. Dr. İbrahim ŞENOL'a, sevgili arkadaşım ve meslektaşım Araş. Gör. Elk. Yük. Müh. İbrahim Beklan KÜÇÜKDEMİRAL'a , araştırmalarım sırasında yardımlarını esirgemeyen oda arkadaşım Arş. Gör. Elk. Yük. Müh. Mustafa Gürkan AYDENİZ'e, bana maddi, manevi destek olan aileme ve elektrik makinaları laboratuvarı çalışanlarına teşekkür ederim.



ÖZET

Geçtiğimiz yüzyılın ikinci yarısından sonra, elektronik alanında baş döndürücü gelişmeler yaşanmıştır. Bunun bir sonucu olarak elektronik cihazlar günlük hayatta sıkça karşımıza çıkmaktadır. Günümüzde artık elektronik cihazlar sayısal elektronik olarak da adlandırılan ve ikilik sayı sistemine dayanan bir prensiple çalışmaktadırlar.

Dijital elektroniğin en önemli gündemi ise tüm işaretlerin sayısal olarak işlenebilmesidir. Bu amacı gerçekleştirebilmek için çok çeşitli işlemciler tasarlanmıştır. Ancak Texas Instruments firmasının 1983 yılında TMS320C10 adını verdiği yeni bir işlemci, sayısal sinyallerin işlenmesi konusunda yeni bir çığır açmıştır. Öncelikle ses ve görüntü işleme alanlarında kullanılmaya başlanan DSP'ler artık günümüzde karmaşık matematik hesaplamalar gerektiren tüm uygulamaların gerçekleştirilmesinde sıklıkla kullanılmaktadır. Bu işlemcilerin motor kontrolü uygulamalarına girmesi ise çok daha yakın bir geçmişe dayanır. Klasik kontrol yöntemlerinin artık yetersiz kalmasıyla birlikte fuzzy, adaptif kontrol gibi yeni kontrol algoritmaları geliştirilmiştir. Ancak bu algoritmaların hayata geçirilmesi için yoğun matematiksel işlemleri çok hızlı yapabilen işlemcilere ihtiyaç duyulmuş ve bu noktadan sonra DSP'ler motor kontrolünde kullanılmaya başlanmıştır.

Bu tez çalışmasında DSP'lerin yapısı ve mikroişlemcilerden farkları anlatılmıştır. Birinci bölümde dijital denetleyici yapılarının gerekliliği, analog denetleyicilerle karşılaştırılarak, farkları ve benzerlikleri ifade edilmiştir. İkinci bölümde mikroişlemcilerin yapısından kısaca bahsedildikten sonra üçüncü bölümde dijital işaret işlemenin temel kavramları üzerinde durulmuştur. Dördüncü bölümde ise bir DSP'de bulunması gereken minimum donanım yapısı detaylı olarak anlatılmıştır. Beşinci bölüm çeşitli DSP yongalarının karşılaştırılmasını içerir. Ayrıca bu bölümde özellikle TMS320C240 işlemcisinden bahsedilmiştir. Altıncı bölümdeki mikroişlemci – DSP karşılaştırılmasından sonra son olarak yedinci bölümde DSP'lerin uygulama alanlarına değinilmiştir. Sekizinci bölümde ise sonuçlar ve bir servo motorun hız kontrolünün TMS320C50 tabanlı bir denetleyici ile nasıl gerçekleştirildiği açıklanmıştır.

Anahtar Kelimeler : Sayısal İşaret İşleme, Mikroişlemciler, Sabit ve Kayan Nokta Aritmetiği, TMS320C240

ABSTRACT

The electronic's industry has been developed sharply in the second half of the twentieth century. As a result of this, electrical devices are appearing in our daily life. Nowadays, almost all electrical devices which are called digital electronics as well, works with the principle of the binary system.

The most important aim of the digital electronics is that the all signals can be processing digitally. In order to realise this aim many kinds of processors were being designed. But the Texas Instruments Company, developed a new processor which called TMS320C10, began a new era in the digital signal processing area. Earlier DSP's were used in sound and image processing, now they are using in not only sound and image processing but also all applications which require complex mathematical algorithms. Taking these processors place in the motor control applications is very near past. Because the insufficiencies of the classical control methods has being developed such as fuzzy, adaptive control, etc. But to make these algorithms useful, there is a necessity about processors which can make complex mathematical operations and after this point DSP's have being used for the motor control

In the study of this thesis, DSP's structures and the differences between the microprocessors are mentioned. In the first part, necessity of the digital controllers structure with comparing the analog controllers is mentioned. After the shortly mentioning on the microprocessors' structure in the second part, basic features of the digital signal processing are explained in the third part. In the next part, the minimum configuration of a DSP is expressed with details. In the fifth part, DSP chips are mentioned and also especially TMS320C240 processors' features is expressed. After the comparison between microprocessor and DSP in the sixth part, DSP's application fields are mentioned in the seventh part. In the eight part, results and an application which is a servo motor control using TMS320C50 chip, are told.

Keywords: Digital Signal Processing, Microprocessors, Fixed and Floating Point Arithmetic, TMS320C240

1. GİRİŞ

Yüzyılın başında endüstriyel alanda gerçekleşen devrim ile insanlık tarihinde yeni bir dönem açılmıştır. Yüzyılın sonunda elektronik alanında yaşanan devrim ile insanlık tarihi yeni bir dönemin başlangıcındadır.

Son yıllarda elektronik alanında, eleman veya devre olarak geliştirilen sistemler, endüstrinin yeniden yapılanma ihtiyacı göstermesine neden olmuştur. Elektroniğin bütün alanlarında gelişmeler baş döndürücü bir hızla sürerken, motor kontrolü de bu gelişmelerden yararlanmıştır.

Mikroişlemcilerin kontrol alanında kullanılmaya başlanması, kontrol sistemlerine yeni bir boyut kazandırmıştır. Analog denetleyici yapıları ile kontrol yapılırken, sistemin belirli koşulları sağlama zorunluluğu, dijital yapılar ile aşılmıştır. Özellikle değişken çevre ve çalışma koşullarında analog devrelerin yetersiz kalması, dijital yapıya yönelmeyi hızlandırmıştır. Kontrol algoritmalarının karmaşık bir hal alması, analog devreler ile gerçekleştirilmesini imkansızlaştırmıştır. Gerçek zamanda işleyen algoritmaların motor kontrolüne uygulanmaları, sistemlerin beklentilerini arttırmış, daha az enerji ile daha büyük sistemlerin kontrolünün gerçekleştirilmesi istenmiştir.

Motor kontrol sistemlerinde, mikroişlemcilerin kullanımı ile, kapalı çevrim kontrole geçilmiştir. Böylece sistemin tam kontrolünün yapılabilmesi sağlanmıştır. Kapalı çevrim kontrolü sağladıktan sonra sistemden beklenenler artmıştır. Örneğin, motorun sadece hız kontrolü yapılmasıyla aynı zamanda akım ve moment kontrolünün de gerçekleştirilmesi sağlanmıştır. Sistemin klasik kontrol yöntemleri ile kontrolü yerine, adaptif veya bulanık yapıların kurulmasıyla, kontrol bütün sistemin en önemli parçası haline gelmiştir. Aynı sistemlerin, kontrol ile çok değişik davranışlar göstermesi, bu alandaki çalışmaların önemini göstermiştir.

Mikroişlemcilerin getirdiği bütün yararlı özelliklere rağmen, kontrol sisteminin yetersiz kaldığı durumlar ortaya çıkmıştır. Kontrol edilecek sistem karmaşıklaştıkça sistemin parametreleri artmış, bu da kontrol algoritmasının matematiksel olarak yüklü bir hale gelmesine yol açmıştır. Motor kontrol sistemlerinin gerçek zamanda işleme zorunluluğu, sistemin çalışmasını olumsuz yönde etkilemiştir. Gerçek sistem ile haberleşmenin ancak belirli bir hata ve gecikme ile mümkün olması, sistemin en zayıf noktası olmuştur. Son yıllarda üretilen sistemlerde bu hatalar, ihmal edilebilecek değerlere çekilmişse de, kontrol

edilecek sistemin parametrelerinin çok hızlı deęişmesi durumunda hala belirli bir hata payı söz konusu olmaktadır. Bu yüzden kontrol sistemleri yeni işlemcilerin arayışına başlamıştır.

İlk olarak görüntü ve ses uygulamalarında kullanılmaya başlanılan DSP'ler kontrol sistemlerine yeni bir bakış açısı kazandırmıştır. İşlem yapma kapasiteleri, üstün matematik işlem hacmi ve en önemlisi veri işleme hızı, DSP'nin kullanım alanının genişlemesini sağlamıştır. Mimari yapı olarak klasik mikro işlemcilerden farklı olan DSP sistemleri, matematik işlemleri donanım olarak gerçekleştirerek, üstün bir hız özellięi kazanmıştır. Bütün bu özelliklerin kontrol uygulamalarında kullanılmaya başlanması, dijital sistemlerin vazgeçilmez denetleyiciler olmalarını sağlamıştır. Analog denetleyicilerin üstün olduęu alanlarda, aynı özellikleri sağlayıp tam bir üstünlük kazanmışlar ve kontrol edilecek sistemin gerçek zamanda kontrolünü gerçekleştirmişlerdir. Dijital kontrol sistemlerinde matematiksel olarak en çok kullanılan yapı, çarpma ve toplama yapısıdır. Mikro işlemcilerin toplama işlemini donanım olarak gerçekleştirmesine rağmen, çarpma işlemi, kullanıcıya gözükmeyen bir yazılım ile yapılmaktadır. Bu da, sistemin toplam işleyiş süresinin artmasına yol açmaktadır. DSP sistemlerinde, işlem yapan ünitelerin birbirlerinden bağımsız olması ve bellek birimlerini ayrı olması, işlemlerin donanım yapısı olarak yapılmasını yani bir komut saat çevriminde tamamlanmasını sağlamaktadır.

DSP sistemlerinin kontrol alanında uygulanmaya başlaması, dięer alanlara göre geç olmuştur. Bunun nedeni, tasarlanan ve uygulamaya konan sistemde maliyet faktörünün hala en önemli etken olmasıdır. Mikro işlemciler geçen yıllarda DSP'lere göre daha ucuz olmasına rağmen, yan iletken teknolojisindeki son gelişmeler, DSP 'nin de aynı oranda ucuzlamasına neden olmuştur. Bu aşamadan sonra, DSP ile tasarlanan kontrol sistemlerinin daha çok uygulama alanı bulacaęı açıktır.

Güç elektronięi ve motor kontrolunda, kontrol yapısının seçimi bir çok faktöre baęlıdır. Tasarlanan sistemin ihtiyaçlarının mikro işlemci veya DSP ile sağlanması durumunda, hangisinin tercih edilmesi gerektiğini sistemin kendisi belirleyecektir. İki yapının da üstün olduęu ve zayıf kaldığı alanlar bulunmaktadır. Kontrol sisteminin optimum çözümü sunması için, tasarımın çeşitli koşulların göz önüne alınarak yapılması gerekmektedir. Prototip olarak gerçekleştirilen bir sistemin endüstri alanında uygulanabilir olması, ancak belirli kriterlerin gerçekleşmesi ile mümkündür. Bu kriterle daha sonraki bölümlerde ayrıntılı olarak incelenecektir.

Endüstride kullanılan motor yapıları, güç elektronięi alanındaki gelişmelere paralel olarak

yerini, yeni alternatiflere bırakmaya başlamıştır. Özellikle küçük güçlü servo uygulamalarında, yeni geliştirilen motor türleri önemli bir yer edinmeye başlamıştır. Fırçasız doğru akım motoru, sürekli mıknatıslı senkron makine ve anahtarlamalı relüktans motoru, geniş kullanım alanı olan motorlara alternatif oluşturmaktadır. Bu tür motorların ancak güç elektroniği devresi yardımıyla çalıştırılabilmesi, bir sakınca olarak gözüktse de, genel anlamda kontrol için aynı yapının diğer motorlar içinde kurulması gerektiği açıktır.

Asenkron motorun üstün yapı özelliklerini, doğru akım motorunun üstün elektriksel özellikleri ile birleştiren bu motorlar ile değişik alanlarda uygulamalar gerçekleştirmek mümkündür. Bu motorlar arasında, üzerinde en çok çalışma yapılan motor, anahtarlamalı relüktans motorudur.

ARM'nin yapı olarak diğer motorlara göre daha basit yapıda olması, daha kolay ve ucuza imal edilebilmesi ve daha üstün elektriksel özellikleri olması tercih sebebi olmasını sağlamaktadır. Güç elektroniği devresinin asenkron motora göre daha basit yapıda olması bir diğer üstünlüktür. Bunun yanında, sistemin kapalı çevrim çalıştırılması için algılayıcı gereksinimi duyması en önemli sakıncasıdır. Kontrol için kurulması gereken algoritmanın gerçek zamanda işleme zorunluluğu, kontrol sisteminin tasarımını daha da önemli kılmaktadır. Sistemin değişkenleri arasında doğrusal olmayan fonksiyonlar, matematiksel modelin kurulmasını ve sabit bir transfer fonksiyonu elde edilmesini önlemektedir. Sistemin, doğru akım motoru gibi transfer fonksiyonu elde edilerek veya asenkron motor gibi d ve q eksenleri takımına indirgenmiş eşdeğer devresi çıkarılarak kontrolü mümkün değildir. Bu yüzden, değişkenlerin gerçek zamanda ve gerçek değerlerle işlemesi gerekmektedir.

ARM, kontrol yapısı gerektiği gibi kurulduğunda diğer motor türlerine göre daha üstün olan bir yapıdır. Aynı şekilde yanlış bir kontrol algoritması veya değişkeninin kullanılması, motorun çok düşük bir performans ile çalışmasına neden olacaktır. Bu yüzden, tasarlanan sistemin en önemli bölümünü kontrol yapısı oluşturmaktadır. Kontrol yapısında ne tür bir işlemcinin kullanılması gerektiği ve algoritmaların sağlanması gereken koşullar sistem tasarımının önemli bir bölümüdür.

Dijital işaretin yapısı daha sonraki bölümlerde incelenmiş ve işaretin işlenmesi sırasında karşılaşılan sorunlar, dijital sistemin işareti değerlendirirken karşılaştığı sayısal hatalar ve bu sayısal hatalardan dolayı oluşan davranış bozukluklarının düzeltilmesinin yolları açıklanmıştır. Denetleyici tasarımı sırasında bu tür hataların ihmal edilebilecek düzeye indirilmesi temel problemlerden birini oluşturmaktadır. Bunun için matematik işlemlerin

sonularını dzelten bir algoritma yapısı kurulmalıdır. Dijital denetleyici sisteminin tasarımı incelendikten sonra, bu yapının kurulmasında mikroişlemcilerin zayıf kaldığı alanlar irdelenmiş ve yeni bir sistemin gerekliliğı ortaya konmuştur.

Drdnc blmde, DSP sistemlerinin genel tanımı, zellikleri ve mimari yapısı incelenmiştir. Dijital denetleyici sistemine kazandırdığı zellikler ve eksiklikleri araştırıldıktan sonra, mimari yapısını oluşturan niteler ayrı ayrı incelenmiştir. DSP 'nin hızlı işlem yapmasını sağılayan bağımsız mimari yapılar, temel zellikleri ve işleme yapıları temel alınacak şekilde incelenmiştir.

Beşinci blmde, DSP yongaların temel zellikleri ele alınarak, retilmiş DSP entegrelerin temel yapısı ve mimari zellikleri, sabit nokta ve kayan nokta işlemcilerinin açıklanmıştır. Bu blmde ayrıca Texas Instruments firmasının motor kontrol uygulamaları iin rettiğı TMS320C240 DSP'si ayrıntılı olarak incelenmiştir.

Altıncı blmde, DSP ve mikroişlemci sistemleri genel yapı ve uygulamaya ynelik olarak karşılaştırılmıştır. Gerek sistemlerin karşılaştırılmasıyla elde edilen bilgiler aktarılarak ile sistem tasarımı sırasında yapılacak tercihlere yol gsterilmeye alışılmıştır.

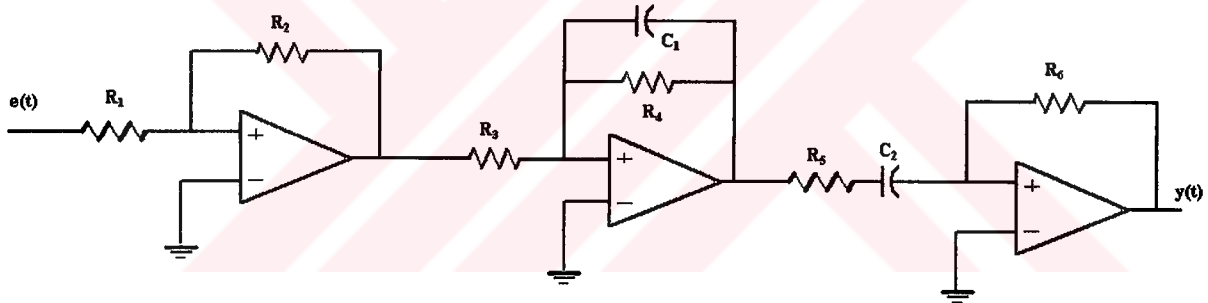
Yedinci blmde, DSP 'lerin uygulama alanları, g elektronik ve motor kontrolu temel alınacak şekilde irdelenmiştir. Kontrol sisteminin tasarımı sırasında kontrol algoritmasının kurulması aşamaları incelenerek, DSP sistemine uygunluğı incelenmiştir. Kontrol sistemlerinde klasik bir yapı haline gelen PID yapısının DSP ile gerekleştirilmesi aşamalar halinde incelenmiş ve gerekli algoritma kurulmuştur. Son olarak sekizinci blmde sonular açıklanmış ve bir servo motorun dinamik hız kontrolnn TMS320C50 DSP'si tarafından nasıl gerekleştirildiğı anlatılmıştır.

1.1 Dijital Denetleyici Yapılarının Analog Denetleyici Yapıları İle Karşılaştırılması

G elektronik ve motor kontrolunda kullanılan sistemlerin daha ok parametre zerinde işlem yapma ve kontrol etme isteğı, kullanılmakta olan denetleyici yapılarının yetersiz kalmasına neden olmuştur. Dijital denetleyicilerin gelişimi, bu konudaki byk bir aığın kapanmasını sağılamıştır. Analog denetleyici, pasif ve aktif devre elemanları kullanılarak sistem parametrelerinin ancak belirli değışim aralığına izin veren yapılardır. Lojik kapılar, işlemsel kuvvetlendiriciler ve pasif devre elemanları ile tasarlanan sistemin kullanımının getirdiğı sorunlar, mikroişlemci tabanlı dijital denetleyicilerin gelişimi ile ortadan kalkmış, ancak bu yapılarda kendilerine zgn sakıncaları beraberinde getirmişlerdir.

Dijital denetleyici olarak adlandırılan sistem, mikroişlemci tabanlı, kontrol edilen sistem ile gerekli olan haberleşmeyi sağlayan donanım özelliklerine sahip ve sisteme uygun bir yazılım desteği olan bir yapıdır. Dijital denetleyicilerin sağladığı en önemli fayda esneklik özelliğidir. Değişik kontrol problemlerine aynı donanım fakat değişik yazılımlar ile kolaylıkla uyum sağlayabilirler. Çözüm olanaklarının bu şekilde program ile çözümlenebilmesi, kontrol edilecek sistemin parametreleri değiştiğinde, analog yapının devresindeki elemanları değiştirmenin zorluğu yanında çok büyük bir avantajdır.

Kontrol edilen sistemin güvenilirliği, denetleyici yapısının güvenilirliği ile doğru orantılıdır. Analog denetleyicinin yapısındaki elemanların birbiri ile bağlantısı, sistem büyüdükçe karmaşıklaşacak, böylece hata olasılığı artacaktır. Dijital denetleyicilerin çevre koşullarından etkilenmesi minimum düzeydedir. Özellikle zor doğa şartlarının bulunduğu çalışma ortamlarında, dijital denetleyici, emniyetli çalışma koşulu aşılmadıkça, analog denetleyiciden çok daha iyi bir performans göstermektedir. Analog devre elemanının sıcaklığa çok bağımlı olması, sistemin hatalı davranma olasılığını arttıracaktır.



Şekil 1.1 : Analog PID denetleyici yapısı

Son yıllarda geliştirilen modern kontrol algoritmalarının analog bir yapı ile gerçekleştirilmesi neredeyse imkansız hale gelmiştir. Kontrol edilen sistem karmaşıklaşmış, parametreleri arttıkça analog denetleyicinin kablo bağlantılı devre yapısı iflas etmiştir. İşlemlerin sayısının artması ve karmaşık hesapların fazlalığı dijital denetleyiciden başka bir çözüme olanak sağlamamaktadır. Mikrobilgisayarların daha büyük bilgisayarlarla bağlanması ile tek bir yapı ile bütün sistemin kontrolü daha duyarlı ve doğru bir şekilde sağlanabilir. Hata fonksiyonlarının işlenmesi, veri işleme, uyarı verme ve görüntüleme özellikleri sayesinde sistemin kontrolü çok kolaylaşmıştır.

Fiziksel sistemler ile haberleşirken, verinin uzak mesafelere taşınması sırasında bilgi kaybı olmaktadır. Analog işaretler, akım ve gerilim seviyesi olarak değişik değerlerde taşınırken,

dijital işaretlerde gerilim seviyesinin 1 veya 0 olması söz konusudur. Çok küçük değerli işaretler, magnetik etkileşim ve parazitik etkilerden dolayı anlaşılabilir hale gelerek, sistemin yanlış davranmasına neden olabilir. Dijital sistemlerde bu hata minimum düzeydedir.

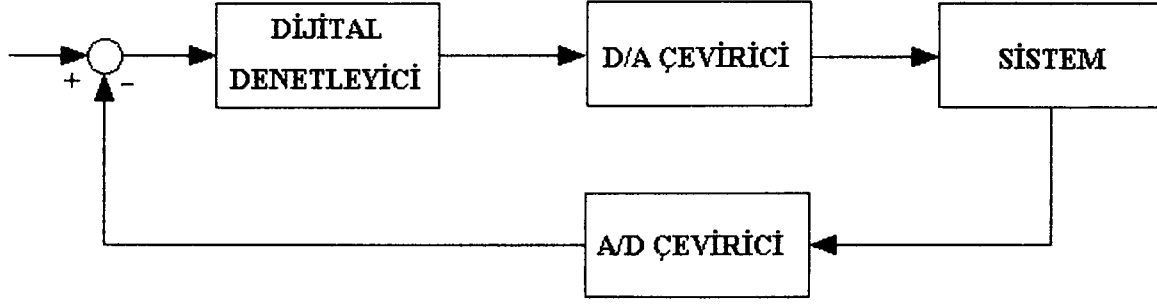
Buna karşılık; fiziksel sistemler analog yapıda olduklarından, dijital denetleyici yapısının kullanılabilmesi için, işaretlerin A/D (analog-dijital) ve D/A (dijital-analog) çeviriciler yardımıyla dönüştürülmesi gerekmektedir. Bu da, belirli bir bilgi kaybına neden olacaktır.

Çizelge 1.1 : Analog ve dijital denetleyicilerin karşılaştırılması

	Analog Denetleyici	Dijital Denetleyici
Avantaj		
	Yüksek doğruluk	Çevre koşullarından etkilenmeme
	Tasarım kolaylığı	Doğru kesin davranış
	Gerçek zamanda çalışma	Karmaşık kontrol algoritmalarına uygun
	Sonsuz örnekleme zamanı	Ek fonksiyonlara olanaklı
Dezavantaj		
	Elemanların yaşlanması	Sayısal hatalar
	Sıcaklığa bağımlılık	Örnekleme zamanının etkisi
	Kablolu bağlantı	Hız problemleri

Dönüşüm işlemi sırasında oluşan anahtarlama kaybı ve kuantalama hatasından dolayı oluşan yanlışlıklar, dijital bilgi işleme sisteminin bit sayısının ve örnekleme frekansının artırılması ile azaltılabilir.

Dijital denetleyicinin işaret işlemesi sırasında sayısal hatalar nedeniyle kontrol sisteminde kalıcı bir hata bulunmaktadır. Analog denetleyicide bu şekilde kalıcı bir hata, devredeki eleman değerlerinin doğru seçilmesi halinde bulunmamaktadır. Mikroişlemcili bir kontrol sisteminin cevabı, fiziksel kontrol sistemin cevabı yanında yavaş kalabilir. Bunun nedeni analog denetleyicinin işaretleri paralel yollardan ihmal edilebilecek bir gecikme ile yollamasına rağmen, dijital denetleyici, seri yol üzerinden haberleşerek, daha uzun bir süre harcamaktadır. Bilgi işlemedeki bu gecikme, kapalı çevrim kontrol sistemlerinde kararlılık problemlerine ve işlenen işaretlerde sorunlara neden olmaktadır.



Şekil 1.2 : Dijital denetleyici yapısı

Dijital denetleyici kullanılarak gerçekleştirilen sistem, yazılım maliyeti yüzünden daha pahalı olabilir. Fakat kontrol edilen sistem büyüyüp, parametreleri arttıkça, denetleyici sisteminin gerçekleştirilme maliyeti, dijital sistemler için bir avantaj haline gelmektedir. Özellikle karmaşık kontrol algoritmalarının uygulanmasında, mikroişlemcili kontrol yapılarının fiyatlarının çok düşmesi ve kontrol edilen sisteme kolayca uyum sağlayabilmesi nedeniyle geniş uygulama alanları bulmasını kolaylaştırmaktadır. Son yıllarda işlemci kullanımı arttıkça firmalar rekabet amacıyla değişik özellikteki işlemciler üreterek, fiyatları aşağı çekmişlerdir. Analog yapılar ancak belirli uygulamalara yönelik üretildiklerinden çok daha pahalı hale gelmişlerdir.

2. MİKROİŞLEMCİLER VE KONTROL SİSTEMLERİNDEKİ UYGULAMALARI

Mikroişlemcilerin kontrol sistemlerinde kullanılmaya başlaması, denetleyici tasarımına yeni boyutlar kazandırmıştır. Denetleyici yapısının tasarlanması için, kontrol edilen sistemin çok iyi tanınması ve parametrelerinin belirli hata sınırları içinde tanımlanmış olması gerekmektedir. Sisteme uygulanan girişe uygun çıkış cevabını vermesi için denetleyici yapısı kullanılmaktadır. Sistemi tanımlayan parametrelerin doğru belirlenmesi cevabın doğruluğunu etkilemektedir.

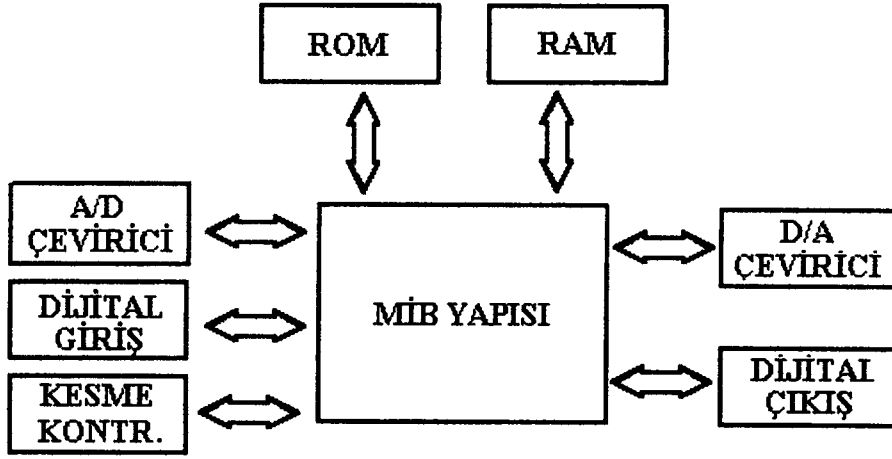
Kontrol sistemlerinde kullanılan mikrodenetleyici yapısı, mikroişlemci biriminin çevre birimleri ile beraber ele alınması ile oluşmaktadır. Fiziksel sistem ile haberleşebilmesi için A/D çevirici birimine ve yazılan programın saklanabilmesi için bellek birimlerine gerek duymaktadır. Program ve veri üzerinde gerekli işlemlerin yapılabilmesi için tekrar tekrar çağırılabilir ve saklanabilmelidir. Sistem kendisine verilen veya dışarıdan gelen veriler ile gerekli bütün matematik ve mantık işlemleri gerçekleştirebilmelidir. Bu verileri değerlendirerek, programın akışına ilişkin gerekli kararları alabilmeli ve değerlendirme yapabilmelidir. Sistemin işlemesi için gerekli bilgi için dış dünya ile gerekli veri iletişimi sağlanmalıdır. Bu özellikleri sağlayan bir mikrodenetleyici, yazılım desteği ile kontrol edilen sistemin ihtiyaçlarına tam olarak cevap verecektir.

Mikroişlemci ile sistemin kontrolüne ilişkin algoritmanın geliştirilmesi sırasında, hızlı cevap veren bir yapı geliştirmek için işlemcinin makine kodu kullanılmalıdır, makine kodu, işlemcinin mimari yapısı kullanılarak programlamayı sağlayan düşük seviyeli bir dildir. Fakat, mikroişlemcilerin donanım özellikleri aynı olmadığından, her biri için geliştirilen makine kodu da farklıdır. Makine kodu için belirli bir standart belirlenemediğinden, mikroişlemci üreticileri, kendi işlemcilerinin yapısına uygun kodlar kullanmaktadırlar. Bu da her mikroişlemci için ayrı bir dil anlamına gelip, tasarımcı için büyük sorun yaratmaktadır. Buna rağmen, işlemcilerin donanım özellikleri genel yapı olarak birbirlerine benzer olduğundan, kullanılan makine kodu yapı olarak aynı, fakat komut ve işleme sistemi olarak farklıdır. Bu yüzden, makine kodunda programlama ile uğraşacak bir tasarımcının işlemcinin donanım yapısını çok iyi kavraması gerekmektedir. (Çiprut,1994)

2.1 Mikroişlemcilerin Yapısı

Bir mikroişlemciden istenen özelliklerin gerçekleşmesini sağlayan mimari yapı şu alt birimlerden oluşmuştur : Merkezi işlem birimi (MİB, Central Processing Unit, MİB), sadece

oku bellek (ROM, Read Only Memory), oku yaz bellek (RAM, Read Access Memory), analog-dijital (A/D) çevirici, dijital-analog (D/A) çevirici, dijital giriş-çıkış birimi (Digital G/Ç) ve kesme denetleyici (Interrupt Controller).



Şekil 2.1 : Mikrodenetleyicinin yapısı

Bir mikroişlemcinin temel birimi, MİB denilen merkezi işlem birimidir. Sistemin beyni olarak adlandırılan bu birim, işlemcinin bütün matematik ve mantık işlemleri ile karar verme işlemlerini gerçekleştirmekte veya işlemlerini kontrol etmektedir. MİB yapı olarak çeşitli alt birimlerin birleşmesinden oluşmuştur. Bu birimler; bellek adres kütüğü, bellek veri kütüğü, akümülatör, aritmetik lojik birim, ve denetleyici yapısıdır. MİB, sistemin diğer birimlerine yollar ile bağlıdır. Sistemde veri ve program akışının sağlanabilmesi için kullanılan bu yollar, adres yolu, veri yolu ve kontrol yoludur. MİB alt birimleri, bellek ve giriş-çıkış birimleri ile haberleşebilmek için bu yollardan yararlanmaktadır.

MİB, verileri saklamak ve üzerinde işlem yapabilmek için kütüklerden yararlanmaktadır. Bu kütükler yardımıyla, diğer birimlere veri yollanmakta ve işleme girecek veri saklanmaktadır. Bellek veri kütüğü, MİB 'de işlenen verilerin, belleğe ve giriş çıkış birimine, veri yolu üzerinden aktarılabilmesi için kullanılmaktadır. Bellek veri kütüğü üzerinden MİB 'ye ulaşan veriler, akümülatör birimine alınmaktadır. Akümülatör birimi, verilerin birimler arasında aktarımını sağlamakta ve işlemin gerektirdiği birimler ile gerekli olan bağlantıyı kurmaktadır. İşlem yapılacak verinin bu birime alınması kullanılan adresleme moduna bağlıdır. Mikroişlemcide, veri işlemek için çeşitli adresleme modlarından yararlanılmaktadır. Adresleme modları verinin, nasıl ve nereden aktarılacağını belirlemektedir. Her işlemcinin temel adresleme modları aynı olmasına rağmen, mimari yapıdaki değişiklikler doğrultusunda işlemciye özgü modlarda bulunmaktadır.

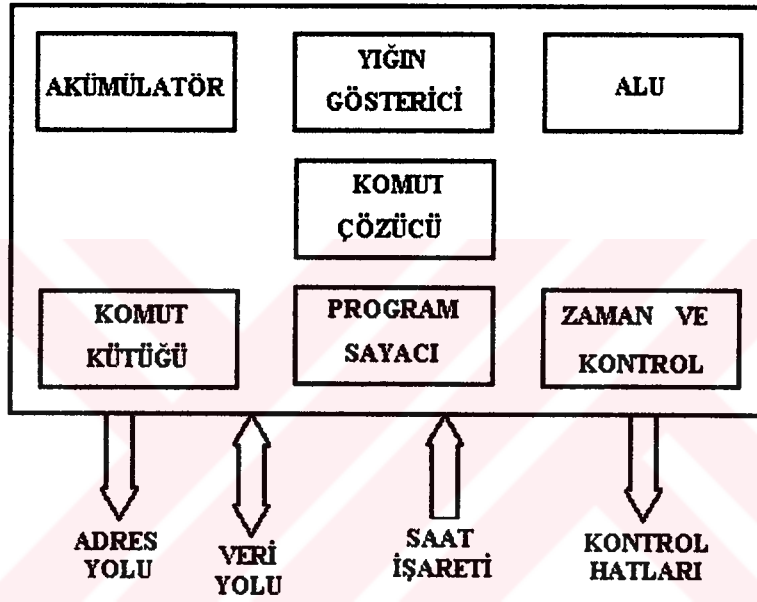
ALU (Aritmetik Lojik Birim), işlemcinin matematik ve mantık işlemlerini gerçekleştirmektedir. ALU, toplama, çıkarma, elde biti ile toplama ve çıkarma, VE, VEYA, evirme gibi temel işlemleri yapmaktadır. Çarpma ve bölme işlemleri, toplama ve öteleme işlemleri kullanılarak yazılım olarak gerçekleştirilebilir. Kontrol sisteminin hızlı ve verimli çalışabilmesi, akümülatör ve ALU 'nun işlem kapasitesine ve hızına bağlıdır.

Mikroişlemcilerde program adım adım işlemektedir ve bu adımları oluşturan komutlar bellekte birbiri ardına yerleştirilmiştir. İşlem yapılacak verinin MİB 'e ulaşmasını veri yolu sağlamaktadır. Bellek adres kütüğü, MİB ile bellek veya giriş-çıkış birimleri arasında işlenecek verilerin bulunduğu yerin adresini belirtme işlemini yerine getirmektedir. MİB, algoritmadaki komutların öngördüğü işlemi gerçekleştirmek için, komut çözücünden yararlanmaktadır. Komutun öngördüğü işlem, komut kütüğünde saklanarak, işlemi gerçekleştirecek birim tarafından gerektiğinde kullanılmak üzere saklanmaktadır. Program sayacı (PS), bir sonra işlenecek komutun adresi ile otomatik olarak yüklenmektedir. Sistem, bir sonraki aşamada hangi satırdaki komutu işleyeceğini program sayacından öğrenir. PS, komut işlendikten sonra bir artar. Bu durum, ancak bir alt program veya kesme işlemi ile karşılaşıldığında bozulmaktadır. Programın işleyişi sırasında, bir karşılaştırma işlemi sonunda "ATLA" komutu ile karşılaşıldığında, PS, "ATLA" komutunun belirlediği yeni bir bellek adresine dallanmaktadır. Bu adresten başlayarak komutları işledikten sonra, daha önce kaldığı yere geri dönmektedir. Program işleyişi sırasında, herhangi bir alt program çağırıldığında, PS, ana programın bir sonraki adresini yığına saklayarak, alt programın bulunduğu adrese gidip işlemi gerçekleştirmektedir. Tekrar ana programa dönme durumunda, yığın göstergesinden daha önce saklanan adres alınmakta ve program kaldığı yerden normal işleyişine devam etmektedir.

İşlemci içinde bulunan zaman ve kontrol birimi, sistemin işleyişini kontrol etmektedir. Birimler arasındaki iletişimin nasıl olacağına bu birim karar vermektedir. Sistemin işlemlerini belirli bir sıra ve zaman içinde yapmasını zaman birimi sağlamaktadır. Yolların paylaşımını gerçekleştirerek, verilerin zamanında işleneceği birime ulaşmasını düzenlemektedir. Veri yolundaki verilerin akış yönünün belirlenmesi, adreslerin ve verilerin gerekli zamanlarda yollara yerleştirilmesi işlemlerini kontrol birimi gerçekleştirmektedir. Mikroişlemcinin saati, dışardan bir osilatör devresi veya kendi içindeki bir ünite yardımıyla uygulanabilir. İşlemcinin bütün işlemleri, saat girişi ile senkron olacak şekilde gerçekleşir. Saatin yüksek frekansta olması , işlemcinin hızlı işlem yapmasını sağlayacaktır.

Veri yolunun genişliği, mikroişlemcinin bit sayısına eşittir. 8 bitlik bir işlemcinin veri yolu 8,

adres yolu 16 hattan oluşmaktadır. Bir programı oluşturan komutlar MİB 'nin iç yapısına uygun yazılmıştır ve ROM 'de saklanmaktadır. Bu komutlar, veri yolu üzerinden MİB 'e yollanarak, MİB 'nin anlayabileceği şekilde kod çözülmekte ve komutun öngördüğü işlem gerçekleştirilmektedir. Böylece program normal çalışmasını sürdürmektedir. Mikroişlemci, kesme işlemi ile karşılaştığında normal çalışmasını bırakarak, kesme alt programını çalıştırmaktadır. Programın işlemesi sonucu üretilen veriler normal çalışma durumunda RAM 'de depolanmaktadır. Eğer, üretilen veri ile fiziksel bir sistem kontrol ediliyorsa, veriler D/A çeviriciye yollanarak fiziksel sistemin anlayacağı hale getirilmektedir. Programda kullanılacak veri de aynı işlemde geçirilerek işlenebilir.



Şekil 2.2 : Mikroişlemcinin alt birimleri

MİB 'nin işlem yapısında, gerektiğinde verilerin saklanıp, daha sonra tekrar kullanıma alınabileceği yığın yapısı bulunmaktadır. Veriler yığına aktarılıp, yığından teker teker geri alınabilir. Yığına son giren veri, ilk olarak geri alınmaktadır. İşlemci içinde yığın, bellek içine yerleştirilmekte ve nerede olduğunu belirtmek için yığın göstergesinden (SP) yararlanılmaktadır. Program çalıştırılmadan önce, yığın göstergesi, yığının başlangıç adresi ile yüklenmektedir. Yığına her yeni veri atıldığında, göstergenin değeri bir azalarak, yığına yeni atılacak verinin yerleşeceği adres gösterilmektedir. Yığın işlemi, sıfır adresli komutlar, kesme işleminde ve alt program işleyişinde, kütüklerin içeriklerini ve dönüş adresini saklamak amacıyla kullanılmaktadır. Alt program işlendikten sonra, ana programın hangi adresine döneceği ve hangi komutun işleneceği, yığında saklanan adres bilgisi ile sağlanmaktadır.

Program işleyişinin artarda gelen yapısı, bir kesme ile bozulabilir. Kesmenin işlevi, işlemcinin dikkatini normal program işleyişinden daha önemli fonksiyonlara çekmektir. Herhangi bir kesme işareti alındığında, mikroişlemci ana programın yürütülmesini durdurur, bir sonraki komutun bulunduğu adresi yığına atar ve kesme işleminde kullanılmak üzere bellek adreslerini işler. İşlemci, kesme işaretine ilişkin bir alt programı çalıştırır ve sonra tekrar ana programa geri döner. Kesme isteği işlemciye dışardan donanım desteği ile veya iç yapıdan donanım veya yazılım desteği ile iletilebilir. Kesme işlevi, sistemin normal çalışmasına aykırı bir durum olduğunda devreye girerek, gerekli önlemlerin alınması için kullanılabilir.

Bellek, veri ve programların saklanması ve gerektiğinde tekrar kullanım amacıyla ortaya çıkarılmasını sağlamak için kullanılır. Mikroişlemcilerde veri ve adresleri saklamak için aynı bellek kullanılmaktadır. Bellek birbirinin aynı yapıda bellek birimlerinin üst üste yığılması ile oluşmaktadır. Her bir bellek biriminde bit denilen bellek birimi bulunur. Aynı sırada bulunan bitler birbirine bağlanarak, veri yolunda, bir yola atanır. Veri yolundaki yol sayısı, bit sayısı kadardır. Aynı yollara bağlı bellek gözlerine erişebilmek için, adresinin bilinmesi yeterlidir.

Mikroişlemcilerde kullanılan bellek ROM (Sadece oku bellek) ve RAM (Oku-yaz bellek) yapısındadır. İşlemcide sürekli kalması istenen bilgilerin saklanabilmesi için kullanılan bellek ROM bellektir. Üretim sırasında veya özel yöntemlerle programlanan bu tür belleklere doğrudan bilgi yazmak söz konusu değildir. ROM, sabit parametreler ve tablolar gibi değişmez parametreleri saklamak için ideal bir bellektir. Çünkü ROM 'un içeriği güç kesintisi olması durumunda bile silinmez, sabittir. Bu tür belleğin içeriği sadece MİB tarafından okunabilir. Bu tür bellekler, EPROM (Elektrikle programlanabilir ROM) ve EEPROM (Elektrikle programlanılıp silinebilir ROM) türünde olabilir.

RAM (Oku yaz bellek) türü bellek, program tarafından kullanılacak veriyi saklamakta kullanılmaktadır. Programın geliştirilmesi sırasında, RAM 'de program tamamlandıktan sonra, ROM türü bir bellek ile sabitlenerek saklanabilir. RAM, silinebilir olduğundan, içeriği disk gibi taşınabilir bir bellek birimine kayıt edilerek, güç kesintisi halinde içeriğinin yok olması önlenemez. RAM 'in MİB ile bağlantısı belirli komutlar ile sağlanabilir. RAM, statik veya dinamik yapıda olabilir. Dinamik yapıda olan RAM saklanacak içerik için periyodik olarak tazelenmelidir, fakat kullanıcı için şeffaf yapıdadır, istenildiğinde ulaşılabilir. Dinamik RAM daha az güç harcadığından daha büyük boyutlarda kolaylıkla kullanılabilir.

MİB 'nin dış dünya ile haberleşmesini sağlayacak olan ana birim giriş çıkış birimidir. G/Ç elemanlarında bilgi giriş çıkışı, bellek elemanlarında olduğu gibi “OKU-YAZ” komutları ile

sağlanmaktadır. Dijital işaretler, dış bir yapı ile haberleşmek için kullanılabilir. Dijital giriş işareti kabul eden herhangi bir birim, doğrudan doğruya bu işaretler ile sürülebilir. Dijital veri transferi, seri olarak universal senkron/asenkron alıcı/iletici üzerinden veya paralel olarak, paralel G/Ç portları üzerinden yapılmaktadır. Klavye, ekran, disk gibi elemanlar dijital olarak işlemciye bağlanarak, bilgi girişi ve sonuçların gösterilmesi için program geliştirme safhasında kullanılabilir.

Kontrol uygulamalarında işlemcinin, analog sinyalleri, A/D ve D/A çeviriciler yardımıyla işlemesi gerekmektedir. Analog giriş işaretleri unipolar veya bipolar yapıda olabilir ve n bit çeviriciler üzerinden dijital kelimelere dönüştürülebilir. Bit sayısının uzunluğu, dijital işaretin kesinliğini ve dönüşüm zamanı işlemcinin maksimum çıkışını belirlemektedir. Çok kanallı girişlere gerek duyulduğunda, birbirinden bağımsız A/D çeviriciler veya analog çoğullayıcı bir karşılaştırıcı kullanılabilir. D/A çeviriciler, dijital bir kelimeyi unipolar veya bipolar bir işarete dönüştürebilir ve bit sayısının uzunluğu analog sinyalin kesinliğini belirlemektedir. Mikrodenetleyicinin işaret işlerken bit uzunluğu, A/D ve D/A çeviricilerin bit uzunluğu ile uyumlu olmalıdır.

2.2 Mikroişlemcilerin Kontrol Sistemlerinde Uygulamaları

Kontrol sistemlerinde mikroişlemcilerin kullanılmaya başlanması, analog denetleyiciler tarafından gerçekleştirilmesi mümkün olmayan birçok uygulamanın hayata geçmesine olanak sağlamıştır. Mikroişlemci ile bir sistemin kontrolünün tasarımı aşamasında analog sistemlerden farklı bir yapı söz konusudur. Sistem tasarımının en kısa sürede en iyi biçimde yapılabilmesi, kontrol edilecek sistemin davranışının ve kontrol parametrelerinin doğru ve eksiksiz olarak belirlenmesine bağlıdır.

Mikroişlemciler kontrol sistemlerinde değişik uygulamalara kolaylıkla uyum sağlamışlardır. Gereken donanım ve yazılım desteği ile birlikte; karmaşık uygulamalara gereken çözümleri sunarak uygulama alanları katlanmıştır. Güç elektroniği ve motor kontrolü sistemlerinde mikroişlemciler, kontrol sisteminin değişik kademelerinde kullanılmaktadır. Sistemlerin kapalı çevrim kontrolünde geri besleme çevrimlerinin gerçekleşmesine olanak sağlamaktadırlar. Geri besleme işaretlerinin kestirimi ve işlenmesinin hızlı bir şekilde yapılabilmesi ancak mikroişlemciler ile mümkündür. Optimal veya adaptif kontrol uygulamaları ancak dijital bir yapı ile gerçekleştirilebilir. Mikroişlemci ile gerçekleştirilen denetleyicilerin koruma, hata belirleme, işaret görüntüleme ve uyarma gibi işlevleri yapması , önemli bir avantajdır. Veri toplama işlemi, test işlemlerinin doğru sonuç vermesi için yeterli

hızda ve doğrulukta yapılmalıdır. Bu genel fonksiyonlar yanında mikroişlemciler, özel uygulamalarda da kullanılmaktadır. Örneğin, kontrollü doğrultucularda ateşleme açılarının kontrolü tamamen işlemci tarafından hesaplanarak uygulanabilir. Eviricilerde PWM kontrolü, mikroişlemcilerle diğer sistemlerden çok daha kolay gerçekleştirilebilir. Mikroişlemcilerin bu genel uygulama alanlarından sonra, güç elektroniğinde özel uygulama alanları şu şekilde sıralanabilir. DC motor kontrolünde, dört bölgeli çevirici kontrolü, dört bölgeli DC kıyıcı kontrolü, yüksek gerilimli çeviricilerde ateşleme açısı ve hata analizi kontrolü, kesintisiz güç kaynaklarının her aşaması, asenkron ve senkron motorlarda eviricilerin PWM kontrolü, statik VAR düzenleri, asenkron motora yumuşak yol verme düzenleri, AC kıyıcılarda ateşleme açıları kontrolü ısıtıcı ve aydınlatma sistemlerinin kontrolü, asenkron ve senkron motorda hız kontrolü, senkron makinenin uyarma akımı kontrolü, akü şarj devrelerinde enerji akışı kontrolü. Bu kadar değişik alanlarda uygulama alanı bulan mikroişlemcili bir kontrol düzeni, yazılımın getirdiği esneklik özelliği sayesinde, birçok uygulamaya donanım yapısında bir değişiklik olmadan uyum sağlayabilir. Bu da mikroişlemcilerin neden bu kadar popüler olduğunu ispatlamaktadır.(Çiprut,1994)

Mikroişlemci ile denetleyici tasarımı belirli aşamalara ayrılmıştır. Bu aşamaların her biri, teker teker ele alınarak, sisteme en uygun yapının kurulması yoluna gidilmelidir. Denetleyici tasarımındaki aşamalar; algoritma geliştirme ve sistem tasarımı, donanım tasarımı, yazılım tasarımı ve sistem entegrasyonu ait başlıklarında toplanabilir.

Sistem tasarımı, kontrol edilecek sistemin analizi ile başlamalıdır. Kontrol parametreleri, gerekli analiz yöntemleri kullanılarak doğru olarak belirlenmelidir. Geliştirilmiş simülasyon programları yardımıyla, sistemin durum uzay modeli ve durum matrisleri belirlenebilir. Böylece, sistemin nasıl bir kontrol algoritması ile kontrol edileceği saptanabilir. Kullanılacak olan kontrol algoritması, istenen performans kriterlerini gerçeklemelidir. Sistemin hangi algoritmayı kullanacağı tamamıyla istenen cevabın özelliklerine bağlıdır. Klasik kontrol sistemlerinde kullanılan PID denetleyici yapısı bir çok sistemin kontrolünde geçerliliğini hala korumaktadır. Parametrelerinin doğru seçilmesi, sistemin istenen davranışı göstermesini sağlamaktadır. Denetleyici parametrelerinin nasıl seçileceği, hangi kriterlerin geçerli olduğu ve sistemin kararlı çalışmasını nasıl sağlayacağı gibi sorulara gereken cevaplar bulunduğu tam bir denetleyici tasarımı yapılmış demektir. Geliştirilmiş PID kontrolör yapıları ile klasik anlamlı bir denetleyici yerine, parametreleri değişken bir yapı ile tam bir kontrol yapılması mümkündür. Bunun yanında, örneğin sistemin aşırı istenen değere oturmasını sağlayan sonlu zaman denetleyici kullanılabilir. Bu yapı, sistemin salınımsız istenen değere oturmasını

sağlarken, sistem parametrelerine çok bağımlı olduğundan, en küçük bir parametre değişiminde denetleyici parametrelerinin elden geçirilip düzeltilmesini gerektirmektedir.

Klasik kontrol algoritmaları yerine sistemin dinamik davranışını da kontrol altında tutabilmek için adaptif kontrol kullanılabilir. Adaptif kontrol yapısı, klasik denetleyici algoritmaları ile karşılaştırıldığında karmaşık gibi gözükse de, sistemin davranışının tam ve doğru olarak kontrolünü sağlamaktadır. Yapı, sistemin nominal parametrelerinin belirlenmesi ve gerçek parametreler ile karşılaştırılarak ve aradaki farkın değerlendirilerek, yeni parametrelerin belirlenmesi prensibine göre işlemektedir. Bu gibi algoritma ve kontrol yapılarının kullanılması sistemden istenen özelliklere ve çalışma koşullarına bağlıdır. Kontrol algoritması seçildikten sonra, akış diyagramı olarak hazırlanıp, yazılım tasarımında kullanılacak duruma getirilmektedir.

2.2.1 Donanım tasarımı

Donanım tasarımı, sistem tasarımının en önemli bölümünü meydana getirmektedir. Amaç sistemin hızını ve doğruluğunu arttırmaktır. Donanım ile yazılım arasındaki ilişki, uygulamanın isteklerine bağımlıdır. Donanım özellikleri, sistemin hızlı çalışması, uyum sağlaması, test edilebilir ve gözlemlenebilir olması ve geliştirmeye yatkın olması olarak sıralanabilir. Donanım tasarımında emniyet, ekonomiklik, gürültü duyarlılığı ve yazılım desteği önemli etkenlerden birkaçıdır.

Mikroişlemcinin seçimi, donanımın tasarımının en önemli bölümüdür. İşlemci seçilirken, kontrol yapısının istekleri ve ihtiyaçları ön planda tutulmalıdır. Bu şekilde istenen kriterleri sağlayan işlemci seçilmiş olacaktır. Mikroişlemci seçiminde göz önüne alınması gereken ilk kriter, işlemcinin kelime genişliğidir. Kelime genişliği uygulamaya göre, 8 bit, 16 bit veya 32 bit olarak seçilebilir. Kelime genişliğinin seçimine kontrol sisteminin duyarlılığına ve işlem hatalarının büyüklüğüne bağlı olarak karar verilmelidir. Kelime genişliğinin büyük olması sayısal hataları azaltacağı gibi, uygulanabilirliğini arttıracaktır. Örneğin, 8 bitlik bir işlemcide sayıların dinamik aralığı kısıtlı olduğundan, büyük sayılar için yazılım desteğine ihtiyaç olmaktadır.

İşlemciden beklenen en önemli özellikten biri de hızdır. Komut işleme süresinin mümkün olduğu kadar kısa olması, sistemin seçiminde önemli bir rol oynayacaktır. İşlemcinin hızlı olması, uygulamalara kolay uyumu beraberinde getirmektedir. Matematik işlemleri yeterince hızlı yapması hatanın azalmasını sağlayacaktır. Sistem saatinin işlemcinin içinde olması veya dışarıdan bir kaynak ile beslenmesi mümkündür. Dışarıdan saat kristali bağlantısı ek bir

donanım getirdiğinden maliyeti arttıracığı gibi, hata olasılığını da arttırmaktadır. Sistem saatinin mikrodenetleyici yapısında olması birçok sorunu ortadan kaldırmaktadır. Mikroişlemci yapısında hafıza, giriş çıkış ve diğer yardımcı birimler bulunmadığından, dışarıdan sisteme eklenmelidir. Bu da maliyeti arttırdığı gibi, sistemin tasarımını zorlaştırmaktadır. Özellikle motor kontrol uygulamalarında gerek duyulan A/D ve D/A çeviricilerin sistemin bir parçası olması, yani aynı tüm devre yapısında bulunması birçok sorunu ortadan kaldırmaktadır.

Mikroişlemcinin bellek ve G/Ç birimlerini desteklemesi tercih sebebi olmaktadır. Uygulamanın gerekliliğine göre, birkaç kesme vektörü bulunması tasarımı kolaylaştıracaktır. Yazılan programın ve elde edilen verilerin saklanması için yeterince büyük boyutta bir bellek yapısı bulunmalıdır. G/Ç birimi ise sistemin dijital olarak dış dünya ile haberleşmesini sağlamaktadır. G/Ç pinlerinin çok sayıda olması, çoğullama yapmadan, yani aynı birimi hem giriş hem de çıkış olarak kullanma gereği duyulmadan, tasarımı sağlamaktadır. Ayrıca işlemcinin kullanılmayan pinlerinin genel amaçlı G/Ç pini olarak kullanılması da aranan bir özelliktir. İşlemci, değişik sistemlere uyumlu çalışabilmesi için, değişik adresleme modlarına destek vermelidir. Adresleme modlarının çeşitli olması, belleğe ve kütüklere aktarılan verilerin değişik ve daha verimli yollarla aktarılmasını sağlamaktadır. Örneğin, işlemcinin bellek biriminde, verilerin devamlı işlendiği bir veri tablosu (look-up table) yapısını en az komut ve en verimli şekilde desteklemesi gerekmektedir. Kütük dosyasının yeterince geniş olması, işlenen verilerin saklanmasını ve kolayca tekrar çağırılarak işleme sokulmasını sağlamaktadır.

İşlemcinin bir diğer önemli özelliği de yeterli yazılım desteğine sahip olmasıdır. Komut takımının yeterince geniş olması, birkaç aşamada tamamlanacak bir işlemin tek komut ile tamamlanmasını sağlamaktadır. Komut işleme süresinin kısa olması ve birkaç komutta yapılan bir işlemin tek bir komutta tamamlanması işlemciye büyük bir avantaj kazandırmaktadır. Örneğin, komut takımının çarpma ve bölme işlemlerini işleyen komutları bulundurmasının, bu işlemlerin toplama ve kaydırma işlemleri ile ayrı bir yazılım olarak gerçekleştirilmesinden çok daha çabuk ve verimli olacağı açıktır. Komut takımında birkaç komutun yaptığı işi tek başına çözümleyen komutların bulunması, algoritmanın kurulması aşamasını kısaltacaktır.

İşlemcinin geliştime sistemlerinin desteğini alması tasarımda önemli bir sorunu ortadan kaldırmaktadır. İşlemcide makine kodunda yazılan programın gözlenmesi, işletilmesi, kontrol edilmesi gibi işlemleri destekleyen geliştirme ünitelerinin bulunması seçimi etkileyecektir.

Makine kodunda program yazımı yanında yüksek seviyeli bir bilgisayar dilini desteklemesi ve işletebilmesi, makine kodunu bilmeyen tasarımcılara kolaylık sağlayacaktır. Ana bir bilgisayar ile iletişim içinde bulunması, sistemin kontrol edilebilmesi ve gözlemlenebilmesi için gereklidir. Uygulamanın karmaşıklığına göre, tek bir işlemci ile çözüm bulunamadığında, birkaç işlemci ile paralel çalışabilecek yapıya destek olması beklenmektedir. Böylece çok daha karmaşık algoritmalar işlenebilir. Yazılım desteği yanında, donanım desteği de tam olmalıdır. Seçilen sistem yeterli bellek birimini ve giriş çıkış birimini bünyesinde bulundurmalıdır.

İşlemci seçiminde bir diğer etken de maliyettir. İşlemcinin kurulan sistemden daha pahalı olması, sistemin kullanılamaz hale gelmesine neden olacaktır. Mikroişlemci veya mikrobilgisayar yapılarından hangisinin seçileceğini uygulama belirleyecektir. Bunun yanında tek bir tüm devre yapısının içinde bütün yardımcı birimlerin bulunması, tasarımı kolaylaştıracak ve az elemanlı bir sistem kurulmasını sağlayacaktır. Bu da maliyetin düşmesi anlamına gelmektedir. Yardımcı birimlerin eklenmesi maliyeti, kullanılan alanı ve tasarımı etkileyecektir. Bunun yanında gerçekleştirilen sistemin geliştirilmeye açık olması, sistem değişimi gerektiğinde, bütün sistemin değişmesi yerine var olan sisteme ekleme yapılarak sorunun çözümlenmesini sağlamaktadır. Mikroişlemci üreten birçok firma olmasına rağmen, sistemin tek bir üretici desteği ile kurulması tavsiye edilmektedir. Bunun nedeni, sistemin gelişimini ve tasarım desteğini her üreticinin değişik ürünlerle desteklemesidir. Üretilen işlemcilerin, makine kodları birbirinden farklı olduğundan, geliştirme sistemleri de farklı olmaktadır. Firmaların işlemcinin tasarımını kolaylaştırmak için sunduğu, geliştirme birimleri, tasarıma geçmeden işlemcinin anlaşılması için önemli bir destek olmaktadır. Bu birimler üzerinde, ek üniteler içinde yer olması, sistemin gelişimini desteklemektedir.

Donanım tasarımı sırasında dikkat edilmesi gereken bir diğer nokta da bellek ve G/Ç birimlerinin tasarımıdır. Program ve veri belleği olarak hangi boyutların yeterli olduğunu algoritma belirleyecektir. Program belleğinin boyutunun belirlenmesi, akış diyagramı üzerinden zor olmaktadır. Genel kontrol uygulamaları için 4K boyutunda bellek yeterli olmaktadır. Yalnız bu, makine kodunda program için geçerlidir. Yüksek seviyeli bir dil kullanıldığında, çok daha büyük bir bellek alanı gerekmektedir. Seçilen ROM ve RAM boyutları ihtiyaca cevap verecek kadar geniş olmalıdır. Programın işlenmesini kolaylaştırmak, EPROM bellek kullanımı ile sağlanabilir. Küçük boyutlu uygulamalarda EPROM çok daha ekonomik olmaktadır. Bu tür bellekler silinebilir olduklarından, tekrar tekrar kullanımı mümkün olmaktadır. Dış bellek ve G/Ç birimleri ile haberleşme işlemciye,

zaman kaybettireceğinden, seçiminde ulaşım süreleri dikkate alınmalıdır. G/Ç biriminin tasarımı için, sistemin işaretlerinin dijital veya analog olanları belirlenmelidir. İşaretler işlemci toprağından izole edilmelidir. Yoksa sistem, hatalı çalışacak veya zarara uğrayacaktır. Analog işaretler bipolar veya unipolar yapıda olabileceklerinden, gereken doğruluk oranına göre A/D ve D/A çeviricilerin bit sayısı seçilmelidir. Gürültü problemlerini en aza indirmek için gerekli önlemler alınmalıdır.

2.2.2 Yazılım tasarımı

Mikroişlemcinin yazılım tasarımı, gerçek zamanda çalışmaya uygun olarak seçilmelidir. İşlemcide bir programın yazılması, genel olarak makine kodunda olmalıdır. Bu işlemcinin hızını arttıracığı gibi, bellek alanını da küçültecektir. Güç elektroniği sistemlerinin kapalı çevrimde çalışması sırasında hız önemli bir etken olmaktadır. Makine kodunda yazılan bir programın işlemesi, diğer yüksek seviyeli dillere göre çok daha hızlı olmaktadır. Bununla beraber, bu dil ile program geliştirmesi, yüksek seviyeli dillere göre daha zordur. Çok daha uzun zaman gerektirdiği gibi, bir çok iterasyona gerek duymaktadır. Yüksek seviyeli dillerden biri ile yazılan programın işleme süresi çok daha uzun olmasına rağmen, geliştirme süresi daha kısa olmaktadır. Son geliştirilen işlemci destek yazılımlarında hem makine kodu hem de yüksek seviyeli bir dil aynı anda karışık olarak kullanılabilir. Bu da sistemin çok daha kısa bir sürede geliştirilmesine olanak sağlamaktadır. Yazılan programın doğru işlemesi için, geliştirme ortamı, adım işleme, parça işleme gibi özellikler ile tasarıma destek olmalıdır. Geliştirilen programın çok karmaşık olması halinde, program parçalara bölünerek, basitleştirilmelidir. (Çiprut,1994)

Örnekleme zamanının seçimi ve doğruluğunun artırılması, denetleyicinin tasarımında önemli bir etkidir. Sistemin örnekleme zamanının seçimi, analog işaretin dijital veriye dönüştürülmesi sırasında doğruluğu etkilemektedir. Analog işaretin dijital olarak işlenmesi sırasında karşılaşılan çeşitli problemler tasarım sırasında sorun yaratmaktadır. Örnekleme frekansı, analog işaretin dijital işarete dönüştürülmesi sırasında, herhangi bir bilgi kaybı olmadan veri işlemesi için dikkatle seçilmelidir. Analog işaretin, dijital bilgiye dönüşmesi sırasında, örnekleme periyodunun düşük seçilmesi, işaretin yanlış yorumlanmasına neden olacaktır. Analog işaret sürekli, fakat dijital veri 2^n 'den oluşmuş seviyeler ile kuantalanmıştır. n bit sayısını gösterdiğinden, bilgi ancak belirli ve sabit bir sayısal hata ile işlenebilir. Kesin sonuç almak için, işlemcinin bit sayısının artırılması yoluna gidilmelidir. Ayrıca, işlenen veride istenmeyen yüksek frekans değerinde bileşenler filtre edilmelidir. İşlenen veri sadece belirli bir bölgeye kaydedilebilir, buna pencere adı verilir. Örneklenen

dalganın frekansı, mükemmel olarak ancak çok uzun bir sürede ölçülebileceğinden; pencere, frekans domeninde, sistemin kararlılığını etkileyecektir. Veri işleme sırasında, pencere verilerin tümünü içine alacak kadar geniş bir aralıkta değilse, anlamlı bilgilerin kaybı söz konusu olacaktır. Özellikle filtreleme ve görüntü analizi uygulamalarında bu hataya rastlanmaktadır.(Çiçrut,1994)

Analog işaretin işlenmesi sırasında, bilginin taşınması ve aktarılması, eğer bilgi çok geniş ise sorunlar yaratmaktadır. Dijital veri işleme sırasında, bilginin sıkıştırılarak, çok daha kısa sürede taşınması mümkün olmaktadır. Özellikle iletişim uygulamalarında veri iletişim hızı, bit oranı (saniyede iletilen bit sayısı veya baud) ile orantılıdır ve kanal sayısı ile sınırlıdır.



3. DİJİTAL İŞARET İŞLEMENİN TEMELLERİ

Mikroişlemcilerin yapısal özellikleri ve uygulama alanları incelendikten sonra, bu bölümde, genel olarak dijital denetleyici tasarımında göz önüne alınacak noktalar üzerinde durulacak, dijital denetleyicilerin sayısal hatalarının algılanması ve azaltılması için alınması gereken önlemler incelenecektir.

Dijital işaret işleme, fiziksel dünyada analog yapıda bulunan sistemlerin dijital olarak kontrol edilmesi amacıyla, sistem modelinin dijital yapıya çevrilmesi ve dijital denetleyicinin belirlenmesi olarak tanımlanabilir. Sistemin kontrolünün yapılabilmesi için ilk aşama olarak, matematiksel modelinin elde edilmesi gerekmektedir. Sistemin yapısının ve çeşitli giriş işaretlerine verdiği cevabı belirleyip, kararlı hale getirmek için gereken analiz yöntemi ortaya konmalıdır. Sistemin kararlı yapıda bir modeli elde edildikten sonra, kontrol etmek için gereken dijital denetleyici yapısı belirlenmelidir. Sistemin kontrolunda izlenecek kontrol yöntemi belirlenerek, bu yapıya uygun algoritma ve denetleyici tasarımı yapılmalıdır. Bu yol izlenirken, sistemin ihtiyaçları tam olarak belirlenip, performans analizi yapılarak optimum denetleyici tasarımı gerçekleştirilmelidir.

Analog bir yapının dijital bir denetleyici ile kontrol edilmesi için yapılar arası dönüşüm gerekmektedir. Analog bir yapının dijital bir yapıya uyumlu hale getirilmesi sistemin ayrıklaştırılması ile sağlanmaktadır. Analog yapıdaki işaret ayrık zamanda örneklenmektedir. Sistemin örnekleme frekansı ile belirlenen bu işlemde sonra, ayrıklaştırılmış işaretin genlikleri belirlenmektedir. Dijital genlik seviyelerinin, analog yapıdaki işareti doğru olarak yansıtması, işlemcinin kelime genişliği ile doğru orantılıdır.

Dijital denetleyici tasarımında dikkate alınması gereken en önemli etkenler; hız, doğruluk ve fiyat olarak sıralanabilir. Seçilen yapının hızı, doğrudan doğruya uygulamaya bağlıdır. İşlem hızı, mimari yapıya, işlem yapma biçimine ve işlemcinin aritmetik kapasitesine bağlıdır. Doğruluk faktörü, sistemin işlem yaparken hata oranını etkilemektedir. İşlem yaparken, sayısal hataların azaltılması, çeşitli önlemler alınarak sağlanabilir. Fiyat faktörü ise, sistemin bütünü göz önüne alınarak, donanım ve yazılım maliyeti olarak ele alınmalıdır.

Dijital denetleyicide uygun işlemci seçilirken, mimari yapı, performans ve çevre birimler ile uyumlu olması göz önüne alınmalıdır. Kontrol sistemi, gerçek zamanda işlem yapan ve belirli bir kontrol algoritmasını işleyen bir yapıdır. Sistemin gerçek zamanda işlem yapması, denetleyicinin performansını etkileyecek sorunları da beraberinde getirmektedir. Denetleyicinin performansını belirleyen diğer bir etken de, örnekleme zamanıdır. İşlemci

ayrık zamanda işlem yaparken, işareti belirli bir frekansta örnekleyerek, işlemektedir. Örnekleme frekansı, kontrol edilen sistemin band genişliğinin en az 10 ile 20 katı olmalıdır. Bu şekilde, sistemin davranışı minimum hata ile algılanabilir. Sistem, işaret işlemeyi bir sonraki örnek işaret gelmeden bitirmelidir, yoksa bilgi kaybı kaçınılmazdır. İşlemcinin bir diğer özelliği de, çevre birimleri ile desteklenmesidir. Kontrol sistemlerinde sıkça gerek duyulan, zamanlayıcı ünitesi, A/D çevirici, PWM üretici gibi birimler sistem seçiminde önemli birer etken olmaktadır. (Çiprut,1994)

Bir işlemcinin işlem yaparken hızı iki faktör ile ölçülebilir: MIPS (Millions of Instructions Per Second), saniyede kaç milyon komut işleyebildiğini göstermektedir. MFLOPS (Millions of Floating Point Operations Per Second), saniyede kaç milyon kayan nokta işlemi gerçekleştirebildiğini göstermektedir. (Texas Instruments, 1993)

Bir işlemcinin hızı genel olarak bu iki birim tarafından belirtilmektedir. Dijital denetleyici yapılarının en çok kullanılan işlemi olan, $X=A*B+C$ 'nin 10000 adımlı bir çevrime sokulması durumunda, eğer sistem, problemi yazılım ile çözüyorsa, birçok öteleme ve toplama işlemi yapması gerekecektir. Mikroişlemcilerde çarpma işlemi, yazılım olarak gerçekleştirilmek zorunda olduğundan, sistemin hızı 0.01 ile 0.1 MFLOPS arasında olacaktır. Eğer sistem, çarpma işlemini donanım olarak gerçekleştirebiliyorsa, 200 ns komut işleme süresi olan bir yapı, bu işlemi 2.5 MFLOPS hızında çözecektir. Donanım olarak gerçekleştirilen çarpma ve toplama işlemleri birbirine paralel olarak aynı komut işleme süresinde gerçekleştirilebilirse, sistemin hızı 5 MFLOPS olacaktır. Eğer, 10000 adet işlemci birbirine paralel olarak çalıştırılırsa, sistemin hızı 50000 MFLOPS olacaktır. 10000 adet işlemcinin paralel olarak çalışması, şimdilik mümkün olmadığından, tek işlemcili çözümlerde kayan nokta ile çalışan sistemler en hızlı sistemler olmaktadır. (Çiprut,1994)

3.2 Dijital Yapılarda Sayı Formatı

Dijital denetleyici tasarımında seçilecek olan işlemcinin sayı formatı, bütün sistemin doğruluğuna etkimektedir. DSP sistemleri, sabit nokta veya kayan nokta formatlarına göre işlemlerini yapmaktadırlar. Dijital sistemler sayıları ikilik düzende işlemektedirler. İkilik düzende sayılar 0 ve 1 ile gösterildiklerinden bütün birimler girişlerinin yüksek veya alçak seviyede olmasına göre işlem yapmaktadırlar. İkilik düzende, pozitif sayılar olduğu gibi gösterilmesine rağmen, negatif sayıları göstermek için özel bir yapıya gerek duyulmuştur. Sayıyı oluşturan bitlerden en anlamlısı, bir diğer deyişle en solda bulunanı işaret biti olarak adlandırılmaktadır. Bu bitin 0 olması pozitif sayıya, 1 olması negatif sayıya karşılık

gelmektedir. Dijital sistemlerin ikilik sayı düzenini kullanmalarının bir diğer şekli de ikinin tümleyeni gösterimidir. (Smith W. Steven, 1997)

İkinin tümleyeni gösteriminde, pozitif sayılarda bir değişiklik olmamaktadır. Sayı, doğrudan doğruya ikilik düzene çevrilmektedir. Negatif sayıların gösteriminde ise, sistem sayıyı ilk önce evirmekte, daha sonra sonuca bir eklemektedir. Son yıllarda gerçekleştirilen işlemcilerin çoğunda ikinin tümleyeni sayı formatı kullanılmaktadır. Bunun nedeni, negatif sayılar ile işlem yapmak için ek bir donanıma gerek duymaması, sayıların işaretini ve genliğini öğrenmek için ayırım yapmamasıdır. Pozitif ve negatif sayıları işlerken sadece bir ekleme birimi ve evirme devresi kullanılmaktadır. Bu da çok daha hızlı ve kolay bir donanım yapısı anlamına gelmektedir. Toplama ve çıkarma işlemleri sadece ekleme birimi ile gerçekleştirildiklerinden daima hatasız sonuç vermektedir. İşlem yapılan sayıların işlemcinin sayı aralığında olması koşulu ile sonuçta taşma olmamakta veya ihmal edilecek düzeyde olmaktadır. İkinin tümleyeni sayı gösteriminde çarpma işlemi, karmaşık yapıda gözükmesine rağmen, geliştirilen değişik algoritmalar ile toplama işlemi kadar kolay ve hızlı gerçekleştirilebilmektedir. (Smith W. Steven, 1997)

Çizelge 3.1 : İkinin tümleyeni sayı formatı gösterimi

67 (Des) = 43 (Hex) = 01000011(Bin)	Pozitif Sayı
-67(Des) = -43(Hex) = 11000011 (Bin)	Negatif Sayı
01000011	İkinin tümleyeni pozitif sayı
Tümleyen al	10111100
Topla	1
10111101 = -67	İkinin tüm. Negatif sayı

8 bitlik bir format ile gösterilebilen en büyük pozitif sayı, 128, negatif sayı ise -127 dir. İşlemci 16 bitlik olduğunda kullanabildiği sayılar 65536 ile -65535 arasında olmaktadır. Dijital sistemlerde sayıları göstermek için belirli formatlardan yararlanılmaktadır. Eğer ondalık sayılar ile işlem yapılmakta ise, bu duruma uygun bir format kullanılmalıdır. Sabit

nokta ile çalışan bir sistemde, sayının kesir bölümü bulunmamakla beraber, uygulamanın türüne göre özel bir gösterim ile tanımlanabilir. İşaret işlemede sayıların ondalık olarak gösterimi hataların azalmasını sağlamaktadır. Örneğin işlenebilen en büyük sayı 0.99 ile ölçeklendirilirse, iki sayının çarpımı hiçbir zaman 1 'den büyük olamayacağı için taşma hatası olması mümkün değildir. Sayının ondalık formatta gösterimi, aritmetik işlemlerin ve çarpım ünitesinin çalışma performansını ve hızını etkilememektedir. Sadece elde edilen sonucun doğruluğunu ve hangi adresten okunacağını etkilemektedir. Ondalık sayı formatı kullanılıyorsa, sonuç yüksek anlamlı 16 bitten, tamsayı formatı kullanılıyorsa, taşma durumu olmadığı varsayılarak düşük anlamlı 16 bitten okunmaktadır. Ondalık sayı formatı ile gösterim, elde edilen sonucun doğruluğunun azalmasına neden olmakla birlikte, taşmalardan korunmayı sağlamaktadır. Tamsayı formatı ile gösterim ise, kesin bir sonuç vermekte ancak taşma durumuna karşı bir koruma yapmamaktadır. Ondalık sayılar ile işlemlerde, çarpma işleminin aksine toplama ve çıkarma işlemleri taşmaya neden olabilir. Bu durum, elde edilen koruyucu bitlerin kullanımı ile ortadan kaldırılmaktadır.

DSP sistemlerinde kullanılan sayı formatı Q15 formatıdır. Q sayının tamsayı bölümünü temsil etmekte, sonra gelen sayı ise kaç bitin kesir sayılara ayrıldığını belirtmektedir. Q15 formatında, 1 işaret biti, 15 kesir biti bulunmakta, tamsayı gösteren bit bulunmamaktadır. Q13 formatında ise, 1 işaret biti, 13 kesir biti ve 2 tamsayı biti bulunmaktadır.

Çizelge 3.2 : Sayı formatları

Desimal	Q15	Q13
+0,5	0,100 0000 0000 0000	000,1 0000 0000 0000
+0,25	0,010 0000 0000 0000	000,0 1000 0000 0000
+0,875	0,111 0000 0000 0000	000,1 1100 0000 0000
-0,5	1,100 0000 0000 0000	100,1 0000 0000 0000
-0,25	1,110 0000 0000 0000	100,1 1100 0000 0000
-0,875	1,001 0000 0000 0000	100,0 0010 0000 0000
-1,00	1,000 0000 0000 0000	100,0 0000 0000 0000

3.2.1 Sabit Nokta ve Kayan Nokta Aritmetiği

Dijital işaret işleme sistemlerinde kullanılan işlemciler aritmetik işlemleri sabit ve kayan nokta olmak üzere iki ayrı yapı ile gerçekleştirebilirler. DSP sistemleri ilk olarak sabit nokta aritmetiğine uygun üretilmelerine rağmen, sonraki nesillerde kayan nokta ile işlem yapabilen kayan nokta işlemcileri de üretilmiştir. İki tür aritmetiğin de birbirlerine göre üstün yanları bulunmaktadır. Kayan nokta aritmetiği çok büyük bir dinamik aralık sağlamasına rağmen, sabit nokta aritmetiği çok daha doğru sonuç vermektedir. Genel tanımlar verildikten sonra, iki yapının karşılaştırılması yapılacaktır.

3.2.2 Sabit Nokta Aritmetiği

Sabit nokta aritmetiği, sayıların normal olarak üs ile ifade edilmeden gösterimi anlamına gelmektedir. İşlemcilerde ikinin tümleyeni sayı formatı kullanıldığında, sabit nokta veya kayan nokta aritmetiğine uyum sağlayabilmektedir. İkinin tümleyeni sayı şu şekilde gösterilebilir.

$$r = 2^{-B} (-b_{m-1} 2^{m-1} + b_j 2^j) \quad [3.1]$$

[3.1]'deki yapı ile bir sayının desimal gösterimi mümkündür. b_j , $j = 0, 1, \dots, m-2$ olmak üzere ikilik basamakları yani bitleri, m toplam kelime genişliğini, b_{m-1} işaret bilgisini, B ise ondalık noktanın yerini göstermektedir. İki özel durum söz konusudur. $B = 0$ ise r tamsayı, $B = m-1$ ise kesir sayı anlamına gelmektedir. Kayan nokta sayılarda B her sayı için değişik değerler alabilirken, sabit nokta sayılarda B sabittir. İkinin tümleyeni formatında sayıların dinamik aralığı simetrik yapıda değildir. $-1.0 < r < 1.0 \cdot 2^{-(m-1)}$ koşulunu sağlayan sayılar kullanıldığında, 1.0 sayısı hiçbir zaman gösterilememektedir. Bu durumda işlemci -1.0 sayısını algılayarak yanlış çözüm yapmaktadır. Bunu önlemek için, dinamik aralığın üst limiti 1.0 sayısına ölçeklendirilmektedir.

Sabit nokta ile sayıların gösterimi 1234.5678 şeklinde olmaktadır. Dijital kontrol sistemlerinde ikinin tümleyeni ve ondalık sayılar tercih edildiğinden, bu gösterim tarzı ile dinamik yapı sınırlı olmaktadır.

Yazılabilen en büyük sayı, işlemcinin kelime genişliği ile aynı boyutta olmaktadır. Örneğin 8 bitlik bir kütük yapısında ondalık sayı gösterimi ile yazılabilecek en büyük ve en küçük sayı, 0.99999999 ve 0.00000001 'dir. Sabit nokta ile işlem sonucunda, iki adet m bitlik sayı işleme girdiğinde sonuç $m-1$ bit olmaktadır. Bunun nedeni sayılarda bir bitin işaret biti olarak kullanılması ve sonuçta sadece bir işaret bitinin yeterli olmasıdır.

3.2.3 Kayan Nokta Aritmetiği

Kayan nokta aritmetiği sayıların üs ve taban gösterimi olarak açıklanabilir. Sabit nokta ile gösterim sonucu ele alınan sayıların dinamik aralığı yeterli büyüklükte olmadığından, çok daha büyük aralıktaki sayıların gösterimine izin veren kayan nokta aritmetiği yapısı ortaya çıkmıştır. IEEE 754 standartlarına göre tanımlanan kayan nokta sayılar, üs ve taban kısımlarından oluşmaktadır. $Y = m \cdot 10^E$ şeklindeki gösterimde, m taban kısmını, E ise üs kısmını göstermektedir. Örneğin, 4 basamak ile gösterilen bir sayının en büyük değeri, $9.999 = 0.999 \cdot 10^9 = 999000000$ olarak verilmektedir.

32 bit uzunluğundaki bir sayıda, en yüksek anlamlı bit işaret bitini, 8 bit üs kısmını, 23 bit kesir kısmını göstermektedir ki, kesir kısmı ve işaret biti taban olarak tanımlanmaktadır. Sistemler, hesaplama sonucu sabit nokta sayı değerlerini aştığı anda kayan nokta gösterime geçecek yapı ile donatılabilirler.

Kayan nokta aritmetiği kullanımı sistemdeki dinamik sayı aralığının yetmemesinden dolayı oluşan sorunlara çözüm getirmesine rağmen, sayıların doğruluğunu olumsuz yönde etkilemektedir. Kayan nokta sayılarda, tabandaki bit sayısı, sayının doğruluğunu göstermektedir. Dinamik aralığın kazancı, kelime genişliği ile büyümektedir. Dinamik aralık ise, kelime genişliği ile, sabit nokta sayılarda lineer olarak artmasına rağmen, kayan nokta sayılarda üstel olarak artmaktadır. Kayan noktalı bir sayının ikilik düzende gösteriminde $y = m \cdot 2^E$ formatına uymak şartı ile, E , üssün alabileceği en büyük değer, N ikilik düzende bit sayısı olmak üzere, $2^N - 1$ dir. Bu yüzden, sayı N biti üs bölümü için, M biti de taban bölümü için ayırmalıdır. Böylece sayının toplam uzunluğu, üs kısmı önce gelecek şekilde, $N+M$ olmalıdır. Örneğin 16 bitlik kayan nokta sayının gösterimi 4 bitin üs, 12 bitin taban olarak kullanılması ile $4E, 12M$ şeklinde olacaktır. Bu format ile yazılabilecek en büyük ve en küçük sayılar;

	Taban	Üs
En büyük :	1111 1111 1111	$1111 = (2^{12} - 1) \cdot 2^{15}$
En küçük :	0000 0000 0001	$0000 = 1 \cdot 2^0$

olarak tanımlanabilir.

Kayan nokta aritmetiği ile gösterilen sayılar, sabit noktalı sayılara göre çok daha geniş dinamik sayı aralığında olmalarına rağmen, sonlu kelime genişliğinin etkilerini tam anlamıyla yok edememektedirler. Sayının dinamik aralığının büyük olması, doğruluğunun düşük olması

sonucunu beraberinde getirmektedir. Dinamik aralık, üs kısmındaki bit sayısı ile, doğruluk ise tabandaki bit sayısı ile orantılıdır. Dinamik aralık ve doğruluk kelime genişliğine bağlıdır. Kayan noktalı sayının formatının yapısı, dinamik aralığı ve doğruluğu değiştirmektedir.

Çizelge 3.3 : Sabit ve kayan nokta sayı formatı özellikleri

Kelime Genişliği	16	32	64
SABİT NOKTA			
Dinamik Aralık	$6,6 \cdot 10^4$	$4,3 \cdot 10^9$	$1,8 \cdot 10^{19}$
Maksimum Doğruluk	4 basamak	9 basamak	19 basamak
KAYAN NOKTA			
	4E,12M	8E,24M	1E,51M
Üs Dinamik Aralığı	$3,3 \cdot 10^4$	$5,8 \cdot 10^{76}$	$1 \cdot 10^{1419}$
Taban Doğruluğu	3 basamak	7 basamak	15 basamak

Çizelge 3.3 'den anlaşıldığı gibi, 32 bitlik bir sayı, sabit nokta formatından 8E, 24M kayan nokta formatına değiştirilirse, 2 basamak doğruluk kaybı olması yanında, dinamik aralık 10^{67} katına çıkmaktadır.

Dinamik aralık ve doğruluk, ikilik düzendeki kayan noktalı bir sayının formatına bağlıdır. 32 bit kelime genişliğindeki sayı 8E, 24M formatı ile ele alındığında, dinamik aralığı $2^{255} = 5 \cdot 10^{1076}$ ve doğruluğu 7 basamaktan büyük olmaktadır. Sayı, 11E, 21 M formatında ise, dinamik aralık, $2^{2047} = 10^{1419}$, doğruluk 6 basamaktan büyük olmaktadır.

Aritmetik işlemler için, sabit nokta ve kayan nokta ile çalışan işlemcilerin yaptığı hatalar değişiktir. Sabit nokta ile çalışan bir işlemci, toplama ve çıkarma işlemi sonucunda oluşabilecek olan taşma hatasını önlemek için elde biti kullanmaktadır. İşlem sonucu, kütüğün bit sayısını aşacak bir değere geldiğinde, elde bitine yazılarak hata önlenmektedir. Sabit noktalı aritmetikte çarpma işleminin hatasının önlenmesi daha fazla elde biti gerektirmektedir. m bit uzunluğundaki iki sayının çarpımının sonucu 2m bit uzunluğunda olacaktır. Sayının tamamen saklanması için 2m uzunluğunda kütük boyu gerekmektedir. Bu yüzden sonuç, yuvarlatma ve kesme işleminden geçirilip, belirli bir hata kabul edilmektedir. Kayan noktalı

aritmetikte toplama işleminde çok küçük sayılar için denormalize işlemine gerek vardır. Üslü sayıların toplanması için üslerinin aynı yapılması gerekmektedir. Bu işleme denormalize denilmektedir. Denormalize işlemi sırasında, verinin kaybı kaçınılmazdır. Çarpma işleminde, kayan noktalı sayıların dinamik aralığının büyük olması hatayı ihmal edilecek düzeyde tutmaktadır. Üslü sayıların çarpımında, tabanlar çarpılıp, üsler toplandığından, taşma olmayacaktır. Buna rağmen, çift kelime genişliğindeki sayılarla işlemlerde, normal kelime genişliğine yuvarlatılırken oluşan hata, üs kısmının kazanç faktörü gibi çalışmasından dolayı, çok çabuk artabilir. Kayan noktalı işlemlerde hatayı daha da azaltmak için, blok kayan nokta işlemler kullanılmaktadır. Kelime genişliğinin sınırlı olduğu durumlarda, yuvarlatma hatasını minimum düzeyde tutmak için, tek bir üs kütüğü yerine üs bloğu kullanılmaktadır. Bu da, fazladan bir kütük ve ölçeklendirme lojiği ile, sayıların tam kapasite ile kullanılmasına olanak sağlamaktadır.

Sabit nokta ya da kayan nokta ile çalışan işlemcilerin seçimi sırasında etkili bir faktör de fiyattır. Geliştirilen sabit nokta DSP sistemleri, mikroişlemciler ile rekabet edebilecek düzeyde olmasına rağmen, kayan nokta DSP sistemleri, hala pahalıdır. Bu yüzden, özellikle görüntü ve ses işleme gibi, hesaplamaların karmaşık ve sabit noktalı bir sistemin başa çıkamayacağı kadar geniş aralıktaki sayıların söz konusu olduğu uygulamalarda kullanılmaktadır. Motor ve güç elektroniği kontrolunda sabit nokta DSP' ler üstün performans ve ucuz fiyatı nedeniyle tercih edilmektedirler. Kayan nokta DSP 'lerin gelecek nesilleri, daha ucuz olduğunda, uygulamalara daha fazla gireceklerdir. Bundan sonra, işlemcilerde sonlu kelime genişliği ve oluşturduğu hatalar üzerinde durulacaktır.

3.3 Sonlu Kelime Genişliği Etkileri

Sabit noktalı DSP 'lerde, aritmetik işlemlerin hatası, kelime genişliğine bağlıdır. İşaretlerin ve katsayıların saklanması için kullanılan kütükler ve mimari yapıdaki birimlerin boyları sınırlıdır. İşlemciler 8, 16, 32 veya 64 bit kelime genişliğinde olabilirler. Sayıların saklanması için kullanılan alanların sınırlı olması, sonlu kelime genişliği olarak adlandırılmaktadır. Alanların sınırlı olması sonucu hataların minimum düzeyde tutulması, işlemcinin dinamik aralığı ve kelime boyuna uygun olarak ölçeklendirme ile sağlanmaktadır. Sonlu kelime genişliği, sistemde gürültü, hatalı işlem yapma ve kararsız çalışma gibi sorunlar çıkmasına neden olmaktadır. Aritmetik işlem sonucu, veri yolundan büyük olduğunda, kelime genişliğini arttırarak veya bazı bitleri atarak işlemin en az hatalı olması sağlanabilir. Bu işlem, kesme veya yuvarlatma şeklinde olabilir. İşlem sonunda bilgi kaybı olmaması için, bit sayısı

iki türlü arttırılabilir. m bit iki sayının çarpımını saklarken, tam doğru sonuç için kelime genişliği $2m$ yapılmalıdır. Fakat bu akümülatör boyu ve hızı için iyi bir çözüm değildir. Bunun yerine, geliştirilmiş doğrulama olarak adlandırılan akümülatöre ek bitler eklenerek, taşmanın etkileri en az seviyeye indirgenebilir. Eklenen bu bitlere koruyucu bit adı verilmektedir. Bu sayede, işlem sonucunun pozitif veya negatif olmasından bağımsız, en az hata ile, g adet koruyucu bit için, 2^g adet işlem gerçekleştirilebilir. Örneğin, 16 bitlik iki sayının çarpımında 8 bit koruyucu biti bulundurmak, sonucun en düşük seviyeli bölümünü atmakta ve yeterli doğrulukta olmasına yetmektedir.

Sonlu kelime genişliği, dijital denetleyicide kuantalama hatasına neden olacaktır. Dijital bir işaretin, ayrıık zamanda genlik seviyeleri ile gösterimi sırasında ortaya çıkan bu hata, işaretin kazanç sabitlerinin, maksimum ve minimum değerlerinin kayba uğramadan temsil edilmesi ile ortadan kaldırılabılır. Kuantalama hatasının azaltılması için, işlemcinin bit sayısının yani kelime genişliğinin arttırılması gerekmektedir. Kuantalama hatası, katsayı kuantalaması ve işaret kuantalaması olarak ele alınabilir. İşaret kuantalaması, üç değişik şekilde ele alınabilir. Bunlar; A/D ve D/A çevrim kuantalama etkisi, taşma etkisi ve kesme, yuvarlatma etkisidir.

A/D ve D/A çevrimlerinde, sürekli işaretin dijital yapıya dönüştürülmesi sırasında hata yapılmaktadır. A/D ve D/A çeviricilerin kelime genişliği genelde 8 veya 12 bit uzunluğundadır. Kelime genişliğinin fazla olması, işaretin kararlılığını arttırarak hatayı azaltmaktadır. Örneğin 8 bitlik bir çeviricide yoğunluk 256 'da 1, ve hata %0.4 iken, 10 bitlik bir çeviricide yoğunluk 1024 'te 1 , hata %0.1 dir. Bunun yanında; çevrimlerin neden olduğu hatalar, kontrol sisteminin cevabına doğrudan etkimemektedir. Uygulamada çeviricilerin bit sayısının işlemciden küçük olması çok büyük bir sorun yaratmamaktadır. Kullanılan sensörlerin doğruluğu da göz önüne alınmalıdır. Örneğin, 5 V 'luk bir sensörde 5mV 'luk bir gürültü varsa, A/D çeviricinin 10 bitten daha büyük seçilmesine gerek yoktur. A/D ve D/A çevrim aynı zamanda denetleyicinin işleme gecikmesine etkimektedir. (Pastacı H. ,1996)

İşleme gecikmesi, işlemcinin kontrol sırasında yaptığı gecikme olarak tanımlanabilir. İşleme gecikmesi, sistem ile olan faz gecikmesi olarak gösterilerek, sistemde negatif faz itilmesine neden olmaktadır. Mikroişlemci sistemlerinin, işleme gecikmesinin büyük olması, birçok uygulamada yavaş kalmasına neden olmuştur. DSP sistemlerinin hızı sayesinde bu gecikme en az düzeydedir. Sistemin negatif faz gecikmesi şu şekilde hesaplanabilir.

$$\text{Faz gecikmesi} = (\text{işleme gecik.}) (\text{Band genişliği frekansı}).360^\circ \quad [3. 2]$$

Örneğin, 1 kHz band genişliği olan bir sistemin kontrolunda, 100 μ s'lik bir işleme gecikmesi,

36° derecelik negatif yönde bir faz kaymasına neden olacaktır. İşleme gecikmesinin etkileri, sistemin işleminin yapısal bir hale getirilmesi ile engellenebilir. İşleme gecikmesinin etkileri, A/D ve D/A çevrimleri sırasında belirginleşmektedir. A/D çevrimi sırasında gecikme, negatif faz kayması yaratarak, sistemin faz aralığını ve kararlılığını etkileyecektir. D/A çevrimi için kullanılan tutucu elemanı da bir örnekleme periyodu kadar gecikme oluşturacaktır. Bu nedenle tasarım sırasında, A/D ve D/A çevirici yerleşme zamanı hesaba katılmalıdır. (Çiprut,1994)

Katsayı kuantalaması, kelime genişliğine uyum sağlamak için katsayıların yeniden ayarlanması olarak tanımlanabilir. Bunun için, katsayılara yuvarlatma veya kesme işlemi uygulanmaktadır. Bu kuantalama sistemin transfer fonksiyonunu değiştirerek, kutup ve sıfırlarının yerlerini belirlemektedir. Katsayı kuantalaması, kelime genişliği kadar sistemin örnekleme zamanına da bağlıdır. Örnekleme zamanı arttıkça, sistem daha kararlı bir hale gelecek şekilde yapısını değiştirmektedir. Kuantalamadan kurtulmak için sistem yapıları bölünerek işletilmektedir. Sistemin yapıları parçalanarak işletilmesi, genel denetleyici yapısını etkilememekle beraber, sonlu kelime genişliğinin etkilerinin azaltması bakımından etkili bir yöntemdir. Yapıların değişik cevaplarının olmasının nedeni, bazı katsayılarının kesme işlemine daha az duyarlı olması ve daha küçük sayı aralığı ile daha kolay ölçeklendirilmesidir. Aynı zamanda, daha düşük dereceden yapılar oluşturduklarından, tasarlanması ve analiz edilmesi daha basit olmaktadır. Kullanılan bazı yapı türleri, direkt yapı, kaskat yapı, paralel yapı ve durum uzay modeli yapıdır.

3.3.1 Taşma

Toplama ve çıkarma işlemleri sonunda, sonucun işlemcinin kelime genişliğinden büyük olması halinde taşma oluşmaktadır. İkinci tümleyeni ile çalışan işlemcilerde toplama değeri maksimum değerden sonra ulaşılan sayı, negatif sayı olmaktadır. Bu da sistemin pozitif sayıların toplamından negatif bir sayı elde etmesi anlamına gelmektedir. İkinci tümleyeni aritmetik kullanıldığında taşma kontrolü yapılarak, sonuç kabul edilebilir sınırlar içinde ise, ihmal edilebilmektedir. Taşmanın etkilerini kontrol altında tutmak için, sistem devamlı her işlem sonunda durum kütüğünde bulunan taşma bitini kontrol ederek, önlem almaktadır. DSP sistemlerinde, işlemcinin taşmaya karşı duyarlılığını kontrol altında tutan doyma modu bulunmaktadır. İşlemin maksimum doğrulukta olması için, yuvarlatma ve ölçeklendirme gibi önlemler alınabilir. Örneğin iki sayının toplamında, işlemden önce yuvarlatılmış ve ölçeklendirilmiş sonuçlar karşılaştırıldığında, ölçeklendirmenin daha hassas sonuçlar verebileceği ortaya çıkmaktadır. Aritmetik ve akümülatör işlemleri sırasında, doğruluğu

artırmak için kelime genişliğinin arttırılması yoluna gidilmelidir. İşlem sırasında taşma olmaması için, en kötü durum göz önüne alındığında, N adet toplama için, $\log_2 N$ adet fazladan koruyucu bit eklenmelidir.

Çizelge 3.4 : Yuvarlatma ve taşmanın etkileri

YUVARLATILMIŞ	ÖLÇEKLENDİRİLMİŞ
0000 0010	010,0 1010
+ 0000 0001	001,0 1111
= 0000 0011	011,1 1001
2+1 = 3	2,3125 + 1,46785 = 3,78

Doyma modunda çalışma, işlemin bir taşma ile karşılaştığındaki davranışını belirlemektedir. Doyma modunda çalışmada, eğer işlem sonucu kelime genişliğini aşıyorsa, yani taşma söz konusu ise, sayı maksimum sayı değerinde tutulur. Bu özellik, analog işaretlerin modellenmesi sırasında, işaretin anlamını kaybetmesini önlemektedir. Eğer, sistem doyma modunda değilse, sonuç, taşma bitine yansiyacak ve sayı yuvarlatılacaktır. Bu da gerçek değerden uzaklaşmak anlamına gelmektedir. Sisteme gereken koşullara göre, doyma modu işleme sokulabilir veya çıkarılabilir. Taşma etkileri, ölçeklendirme faktörünün doğru seçilmesi ve işlem sonunda taşmanın etkisini yok etmek için koruyucu bitler eklenmesi ile azaltılabilir.

3.3.2 Yuvarlatma ve Kesme

İşaret işleme sırasında sinyallerin belirli değerleri aşması durumunda yuvarlatma ve kesme yapılarak hatanın azaltılması yoluna gidilmektedir. İşlemlerin doğruluğunun artması için bu hataların mümkün olduğu kadar düşük seviyede tutulması gerekmektedir. 16 bitlik iki sayının çarpımı sonucu 32 bit uzunluğunda olmaktadır. Eğer sayının tamamını saklayacak kütük boyu mevcut değilse, sayının en yüksek anlamlı bitleri alınarak sonuç kütük boyuna uygun hale getirilmektedir. Eğer, işlem sonrasında düşük anlamlı 16 bit atılırsa, buna kesme hatası denilmektedir. Eğer, LSB (en düşük anlamlı byte) ayrılmadan önce, bir sonraki değere yuvarlatılırsa, buna yuvarlatma hatası denilmektedir. Çarpma işlemi sonunda sayının LSB 'si atılarak sadece MSB' leri (yüksek anlamlı byte) saklanmaktadır.

Çizelge 3.5 : Kesme ve yuvarlatma karakteristiği

MSB	LSB	MSB	
0010	1001	0010	KESME
0010	1001	0011	YUVARLATMA

Kesme işlemi, sayının son bitlerinin değerlerine bakmadan kesmektedir. Bu da hatanın fazla olması anlamındadır. Örneğin, 1,39 sayısı kesme işleminden sonra 1,3 olarak algılanmaktadır. Yuvarlatma ise, sayının LSB ' si, yarı değerinden küçük ise, küçüğe yuvarlatmakta, büyükse, büyük sayıya tamamlamaktadır. Örneğin, 1,44 sayısı 1,4 ve 1,34 sayısı 1,3 olarak algılanmaktadır. Tarafsız yuvarlatma adı verilen bir diğer yapı ise, sayı tam orta değerde ise, sayının hangi değere yuvarlanacağına rastlantısal olarak karar vermektedir. Normal yuvarlatma işlemi orta değerlerle karşılaştığında hep büyük sayıya yuvarlatırken, tarafsız yuvarlatma işlemi %50 oranında sayıyı küçüğe veya büyüğe yuvarlamaktadır. Böylece yapılan hata bir ölçüde azaltılmış olmaktadır.

Kesme işlemi, hatanın yüksek olmasından dolayı tercih edilmemektedir. Yuvarlatma, dijital işaret işleme sistemlerinde daha doğru sonuçlar vermektedir. Tarafsız yuvarlatma, b bit uzunluğundaki sayıda yuvarlatılmamış değere en yakın biti seçip, sonucu yuvarlatarak, hatanın sıfır ile 2^{-b} arasında olmasını sağlamaktadır. Filtre uygulamalarında, kesme işlemi genişlik olarak aynı hatayı vermesine rağmen, bütün küçük değerli değişiklikler atılarak, sayının en düşük değere ötelenmesine neden olmaktadır. Bu sebeple yuvarlatma filtre uygulamalarında tercih edilmektedir.

$$\varepsilon = \frac{y(yuv.) - y(ger)}{y(ger)} \quad \frac{1}{2}2^{-b} < \varepsilon < -\frac{1}{2}2^{-b} \quad [3.3]$$

Yuvarlatma ve kesme hataları sistemde gürültü ve kararsız çalışmaya neden olmaktadır. Sınırlı salınım oluşturmasından dolayı, sistemi nonlineer bir çalışma durumuna sokan bu hatalar, çeşitli gürültü ve hata modelleri ile azaltılabilir. Bu gürültü modelleri, gürültü yoğunluğu cinsinden gürültünün olasılık dağılımını hesaplayarak, sistemin minimum hata ile çalışmasını sağlamaktadır. Hataların sistem üzerindeki olumsuz etkileri sistemin yapıları parçalanması ile önlenir. Örneğin, dördüncü dereceden bir denetleyicinin birbirine paralel ayrı elemanlara ayrılması, yuvarlatma ve kesme hatasının etkisini büyük oranda azaltacaktır.

3.3.3 Ölçeklendirme

Sonlu kelime genişliğinin etkilerini azaltmak için, kontrol yapısında kullanılan sayılar, belirli bir ölçeklendirme faktörü ile çarpılarak, doğru işlem yapılması sağlanmaktadır. Ölçeklendirme faktörünün doğru seçimi; sayıların dinamik aralıklarını aşmalarına engel olmaktadır. Ölçeklendirme faktörünün gereğinden daha büyük seçilmesi durumunda, taşma durumu oluşabilir. DSP 'lerde taşmaya karşı koruma sağlanmasına rağmen, mümkün olduğu kadar olasılık da ortadan kaldırılmalıdır. Bu yüzden daha küçük ölçeklendirme faktörü seçilmelidir. Fakat, bu durum da kuantalama sırasında oluşan gürültünün artmasına neden olacaktır.

Ölçeklendirmenin başarı ile yapılabilmesi için, ilk olarak denetleyicinin davranışı tam olarak belirlenmelidir. Denetleyici belirli alt yapılara parçalandığında, eğer kontrol edilen yapının cevabını tam olarak düzenliyorsa, ölçeklendirmeye gerek kalmamaktadır. Sistem cevabının değişmesine karşın, her alt denetleyici birimi için ayrı ölçek faktörü kullanılabilir.

Ölçek faktörünün doğru seçilmesi için sistemin en kötü şartlardaki cevabı belirlenmektedir. Bu cevap, sistemde kararlılık sınırını ve taşma durumunu zorlayarak, en kötü çalışma koşullarını temsil etmektedir. Daha sonra ölçek faktörünün doğru seçilebilmesi için, giriş ve çıkış işaretlerinin değerleri dikkate alınmalıdır. Sistemin taşma ve doyma moduna girmesini önleyecek bir faktör belirlenmelidir. Bunun için çeşitli simulasyon programları yardımıyla sistemin cevabı üzerinde değişik ölçek faktörlerinin etkileri gözlenmelidir. Ölçeklendirme, giriş ve çıkış birimleri ve durum vektörlerinin ölçeklendirilmesi olarak ayrılabilir. Giriş-çıkış birimlerindeki işaretlerin fiziksel karşılıklarının sabit sayılara karşılık gelmesi, işlemleri kolaylaştıracaktır. Örneğin A/D ve D/A birimlerinde maksimum ve minimum gerilim değerleri, ondalık sayı formatı kullanıldığında +0.999 ile -1.000 arasında olması istenmektedir. Sayıların değişimi, gerilim değerlerinin değişimini tam olarak yansıtmalıdır. Örneğin, kontrol edilecek sistemin durum modeli;

$$\begin{aligned} X_{n+1} &= A X_n + B U_n & X: \text{Giriş}, Y: \text{çıkış}, U: \text{kontrol} & [3.4] \\ Y_n &= C X_n + D U_n & A, B, C, D: \text{Durum matrisleri} & \end{aligned}$$

olarak verildiğinde, durum matrislerinin ölçeklendirmesi gerekmektedir. Bu yüzden B, C, D matrisleri, $(S_y)^{-1}$ ve $(S_u)^{-1}$ diagonal ölçeklendirme matrisleri olmak üzere şu hale gelmektedir.

$$S_{u,i} = \frac{R^{pr}}{R_{u,i}^{ph}} \quad S_{y,i} = \frac{R_{y,i}^{ph}}{R^{pr}} \quad [3.5]$$

$$B_x = B[(S_u)^{-1}] \quad C_x = [(S_y)^{-1}]C \quad D_x = [(S_y)^{-1}]D[(S_u)^{-1}] \quad [3.6]$$

Ölçeklendirme matrisleri diyagonal bir yapıdadır. R^{pr} , işlemcinin sayı aralığını belirtmektedir. Ondalık sayı formatında 2 'ye eşittir. R^{ph} ise, fiziksel sistemin sayı aralığını göstermektedir ki, bu değer durum matrislerinin orijinal değerlerinden elde edilmektedir. Örneğin fiziksel bir sistemde 10 V 'luk gerilim seviyelerinin -1.000 ile 1.000 arasındaki sayı aralığında olması için; $B_x = 10B$, $C_x = 0.1C$, $D_x = D$ olmalıdır.

Durum vektörü ölçeklendirmesinde ise, sistemin durum uzay modeli kurulduğunda, her bir durum değişkeni, işlemcinin sayı aralığında kalacak şekilde bir faktör seçilmelidir. Böylece sistem cevabı istenilen değere getirilebilmektedir. Sisteme uygun ölçek faktörü matrislerinin seçilmesi için değişik yollar izlenebilir. S_x matrisini belirlemek için, sistemin en kötü çalışma koşullarının kapalı çevrimde simülasyonu yapılmalıdır. Bu durumda, her toplama işleminde ve sistemin her düğümünde taşma kontrolü yapılmalıdır. Sistemin en kötü durumu, durum değişkeninin en büyük değeri olarak alınmalıdır.(ÇiPrut,1994)

$$S_{x,i} = maks(X_{n,i}) \quad S_x = diag\left(\frac{1}{S_{x,i}}\right) \quad [3.7]$$

$$X_{n+1} = (S_x^{-1}A.S_x)X_n + S_x^{-1}B.U_n \quad Y_{n+1} = C.S_x.X_n + D.U_n \quad [3.8]$$

Ölçek faktörü matrisinin hesaplanmasında izlenen bir diğer yol da, simülasyon yapmak yerine her düğümde taşma olma olasılığını istatistiksel yöntemlerle hesaplayarak analiz yapmaktır. Ölçeklendirme faktörünün yanlış seçilmesi veya sistem cevabına uygun olacak şekilde belirlenememesi, sistemin davranışını kötü yönde etkileyerek, kararsız ve gürültülü çalışma durumu yaratacaktır.

3.3.4 Analog Dijital Dönüştürücüler

Bilim ve mühendislik alanında karşılaşılan işaretlerin bir çoğu sürekli işaretlerdir. Bu türdeki işaretlerin dijital bilgisayarlar ya da işlemciler tarafından işlenebilmesi için mutlaka ayrık işaretlere dönüştürülmesi gerekmektedir. Sayısallaştırılmış bilgi ile sürekli işaret iki önemli açıdan farklılık gösterir : Örnekleme ve kuantalama. Bu iki kısıtlama ayrık işaretin içerdiği bilgiyi bir miktar azaltacaktır. Bu noktada önemli olan ne kadar bilginin tutulması

gerektiğinin ve ne kadar bilginin ihmal edilebileceğinin belirlenmesidir. Bunun için örnekleme frekansının değeri, bit sayısı ve dijital alan ile analog işaret arasında bulunması gereken analog filtrenin cinsi tespit edilmelidir. (Smith W. Steven, 1997)

3.3.4.1 Kuantalama

Aşağıdaki şekilde tipik bir analog – dijital dönüşümü dalga şekilleri ile birlikte verilmiştir. Şekil3.1(a)'da orjinal analog sinyal gösterilmiştir. Buradaki işaret gerilimin zamana göre değişimidir. İşlem kolaylığı bakımından gerilimin 0 ile 4,095V arasında değiştiğini varsayalım. Buna bağlı olarak dijital değerler de 12 bitlik bir dönüştürücü için 0 ile 4095 arasında olacaktır. Blok diyagram örnekle-tut (sample-hold) ve analog-dijital dönüştürücü olmak üzere iki kısımdan oluşmuştur. Örnekle-tut ünitesi ADC dönüşüm işlemi yaparken giriş işaretinin sabit kalmasını sağlar.

(a) ile (b) şekilleri arasındaki farktan da anlaşılacağı gibi örnekle-tut devresinin çıkışı periyodik aralıklarla değişir. Dolayısıyla bu periyot süresince giriş işaretinde oluşacak olan değişimler tamamen gözardı edilmiş olacaktır. Bu periyotun boyutunu sistemin özellikleri belirleyecektir. Örnekleme işleminin sonunda bağımsız değişken sürekliden halden ayrık hale getirilmiş olacaktır. Örneğimizdeki bağımsız değişken zaman olmaktadır.

Şimdi de (b) ile (c) şekilleri arasındaki farka dikkat edelim. ADC, örnekle-tut devresinin çıkışındaki her düz bölge için 0 ile 4095 arasında bir tamsayı üretmektedir. (b)'deki her düz bölge 0V ile 4,095V arasındaki herhangi bir değer olabilir. Bir başka deyişle, bu değer 12 bitlik dijital işaret ile tam olarak ifade edilemediği durumlar da mevcuttur. Örneğin ADC, örnekle ve tut devresinin çıkışında gördüğü 2,56000V ve 2,56001V değerlerini aynı bir dijital tamsayı (2560) olarak dönüştürecektir. Buradan da anlaşılacağı gibi kuantalama, bağımlı değişkeni, ki örneğimizde bu gerilimdir, sürekli halden ayrık hale dönüştürmektir. Bu işlem bir miktar hataya neden olacaktır.

(a) ile (c) şekillerini karşılaştırmaktan kaçındığımızı dikkat ediniz. Çünkü bu durumda örnekleme ve kuantalama bir arada olacaktır. Dolayısıyla her bir işlem esnasında olacak değişimleri fark etmek oldukça zorlaşacaktır. Her iki işlemin oluşturacağı etkileri birbirinden ayırmak oldukça önemlidir. Çünkü her iki işlemin işaret üzerinde yaptığı bozucu etki birbirinden farklıdır. Bu iki işlemin tek tek kullanılabileceği uygulamalar da vardır. Örneğin : anahtarlamalı kondansatör filtrelerinde kuantalama yapılmaksızın örnekleme işlemi yapılır.

Öncelikle kuantalamanın etkisini inceleyelim. Örneklenen her işaret maksimum $\pm 1/2\text{LSB}$

kadar bir hataya neden olabilir. Şekil (d)'de örneğimizdeki kuantalama hatası grafik olarak verilmiştir. Bu hata Şekil (b)'den (c)'nin, bir başka deyişle kuantalanmış işareten örneklenmiş işaretin çıkarılması ile elde edilmiştir. Buradan hareketle dijital çıkışın (c), örneklenmiş sürekli giriş (b) ile kuantalama hatasının (d) toplamından oluştuğunu söyleyebiliriz. Bu analize bakarak kuantalama hatasının Şekil 3.1(d)'den de görüldüğü üzere rastgele gürültü gibi davrandığını ifade edilebilir.

Örneğin genliği 1,0V ve rastgele gürültü genliği de 1,0mV olan bir işareti ele alalım. Bu işareti 8 bitlik bir dijital bilgi hale getirdiğimizde, 1,0V dijital olarak 255 sayısına ve 1,0 mV'luk gürültü de 0,255 LSB'ye karşılık gelecektir. Hata, kuantalama gürültüsü ile işaretin kendi gürültüsünün aşağıdaki şeklinde toplamı ile bulunabilir.

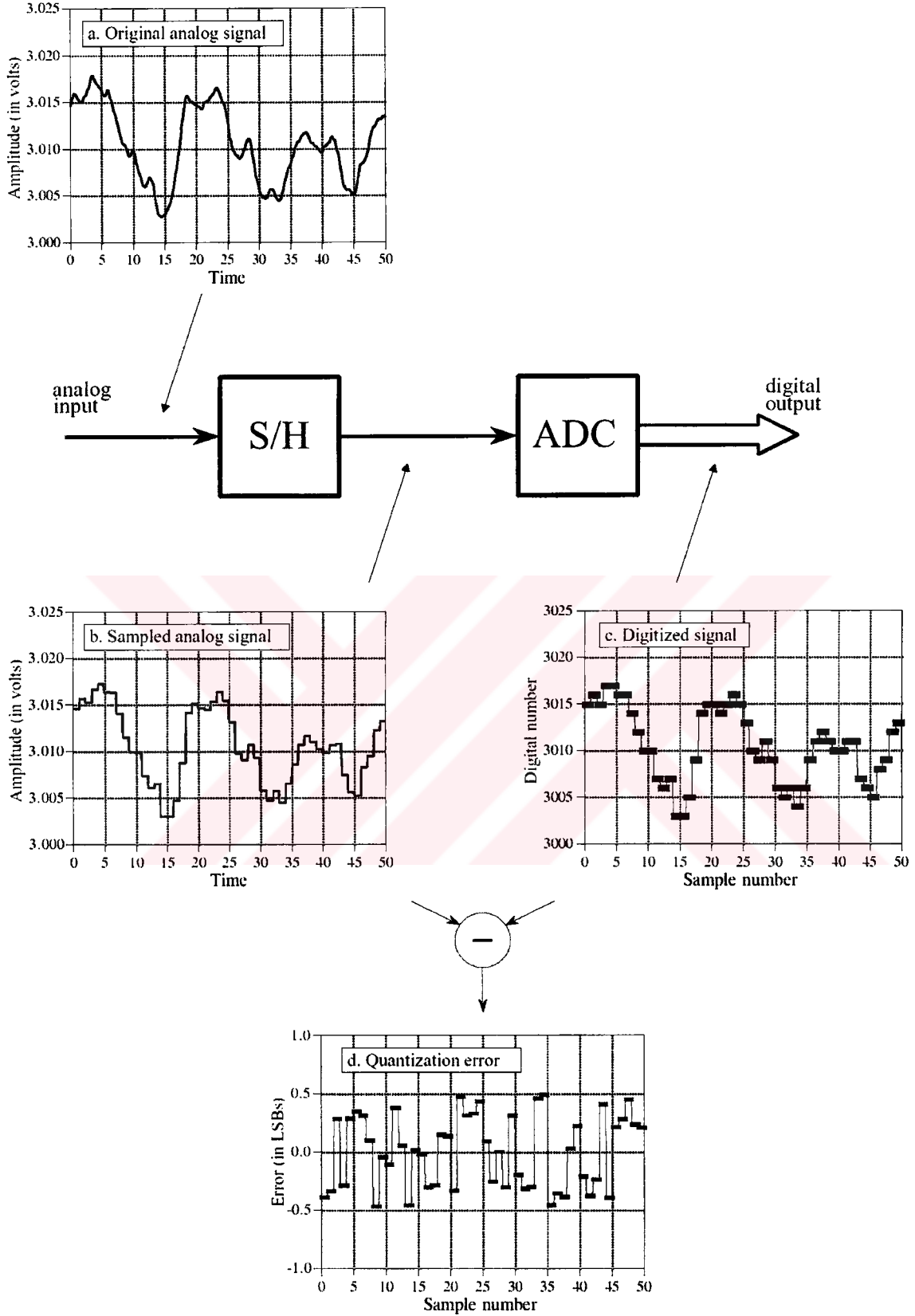
$$\sqrt{A^2 + B^2} = C \quad [3.9]$$

Sayısal değerler yerine yazılırsa toplam hata :

$$\sqrt{0,255^2 + 0,29^2} = 0,386 \text{ LSB} \quad [3.10]$$

olacaktır. Bu da analog sinyalde var olan gürültünün dijitale dönüştürüldüğünde %50 oranında artması anlamındadır. Aynı işaret 12 bitlik dijital işarete dönüştürülmüş olsaydı, gürültü miktarında bir artış olmayacaktı ve kuantalama nedeniyle herhangi bir bilgi kaybı meydana gelmeyecekti. Dolayısıyla bir analog işareti dijitale dönüştürürken dikkat edilecek iki nokta vardır. Analog işarete ne kadar gürültü vardır ve dijital işarete ne kadar gürültüye izin verilebilir.

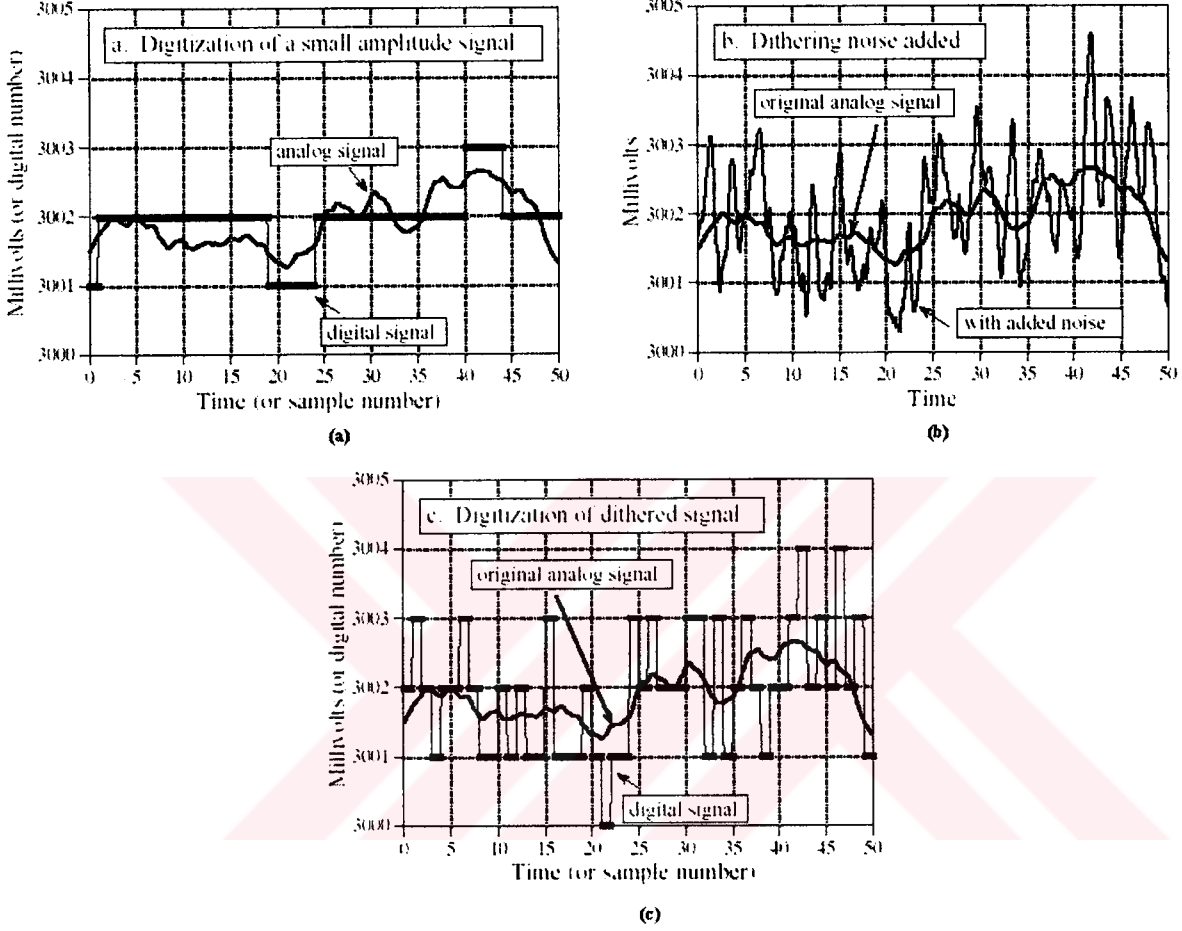
Aşağıdaki şekilde sayısallaştırma işlemi sırasındaki işaretin değişimi verilmiştir. Dönüşüm süreci, örnekleme ve kuantalamının etkilerinin daha iyi ayırt edilebilmesi için iki kısımda incelenmiştir. (a) şeklinde verilen işaret orjinal analog sinyal olup bu işaretin örnekle-tut devresinin çıkışındaki şekli (b)'de verilmiştir. Bu devre sayesinde ADC, dönüşüm işlemi tamamlayana kadar girişindeki işaret sabit tutulur. ADC'nin çıkışındaki dalga şekli ise (c)'deki gibidir. ADC bloğunun yaptığı işlem girişindeki sabit analog işaretin genliğine bağlı olarak çıkışında sabit bir tamsayı üretmektir. ADC girişindeki analog işaretin genliği, sonlu kelime genişliğinin etkisi dolayısıyla, her zaman sıfır hata ile sayısallaştırılmaz. Böyle bir durumla karşılaştığında ADC en yakın dijital değeri çıkışa atar dolayısıyla oluşabilecek maksimum hata $\pm 1/2\text{LSB}$ 'dir. (d)'de ise kuantalama hatası verilmiştir. Bu hata rastgele bir gürültü şeklinde modellenebilir.



Şekil 3.1 : Sayısallaştırma işlemi esnasındaki dalga şekilleri

Bahsedilen kuantalama modeli, kuantalama hatasının rastgele olmadığı durumlarda

geçerliliğini yitirir. Örnek olarak ardışık örneklemeelerde aynı değerin alındığı aşağıdaki şekildeki (Şekil3.2a) gibi uygulamalar verilebilir. Bu durumda dijital çıkış uzun bir süre boyunca, analog sinyal $\pm 1/2\text{LSB}$ arasında değişse bile, aynı değerde kalır. Kuantalama hatası rastgele gürültü yerine eşik etkisi ya da distorsiyon gibi görünmeye başlar.



Şekil 3.2 : Dithering ya da titreşim etkisi olarak ifade edilen durumun aşamaları

Dithering yavaş değişen işaretlerin dijitale dönüştürülmesini iyileştirmek için yaygın olarak kullanılan bir metoddur. Örneğimizde; eklenen gürültü $2/3\text{ LSB}$ standart sapmasıyla dağıtılır. Dolayısıyla tepeden tepeye genlik 3 LSB olur. Şekil 3.2(b)'de analog işarete küçük bir miktar rastgele gürültü eklendiği durum gösterilmiştir. Şekil 3.2(c)'de elde edilen yeni işaretin dijitale dönüşümünü gösterilmiştir. Orjinal analog sinyal $\pm 1/2\text{LSB}$ 'den az değiştiğinde, eklenen gürültü dijital çıkışta ardışık seviyeler arasında bir salıma neden olduğu şekilden görülebilmektedir. Böylece çıkış sabit bir seviyede sıkışıp kalmamış olur.

Yapılan işlemin, dönüşümü nasıl daha iyi bir hale getirdiğini daha iyi anlayabilmek için bir örnek üzerinde düşünelim. Sabit analog sinyalimizin genliği $3,0001\text{V}$ olsun. Eğer dithering

metodu uygulanmadan 10000 örnek alınsaydı, 10000 adet özdeş ve 3000 olan değer elde edilecekti. Fakat, aynı işlem küçük bir miktar dithering gürültüsü eklenmiş olarak yapılırsa 10000 adet değerlerin %90'ı 3000 ve geri kalan %10'u da 3001 değerini alacaktır. Tüm değerlerin ortalaması alındığında sonuç 3000,1 olacaktır. Daha fazla sayıda örnekleme yapılırsa, metot sayesinde, elde edilen sonucun daha fazla bilgi içerdiği açıkça görülür. Sonuç olarak gürültü eklemenin daha fazla bilgi sağlayacağı ortaya çıkar ki bu beklenenden farklıdır.

Bu metod için kullanılan devreler oldukça karmaşık olabilir. Örneğin rastgele sayılar üretmek için bilgisayar kullanılır ve sonra bu sayılar bir DAC aracılığıyla analog işarete dönüştürülerek eklenecek olan gürültü elde edilmiş olur. Sayısallaştırma işleminden sonra bilgisayar dijital işareten rastgele sayıları kayan nokta aritmetiğini kullanarak çıkarabilir. Bu teknik subtractive dither olarak adlandırılır ve sadece karmaşık sistemlerde kullanılır. En basit metod ise dithering için analog sinyalin kendisinde zaten bulunan gürültünün kullanılması prensibine dayanır.

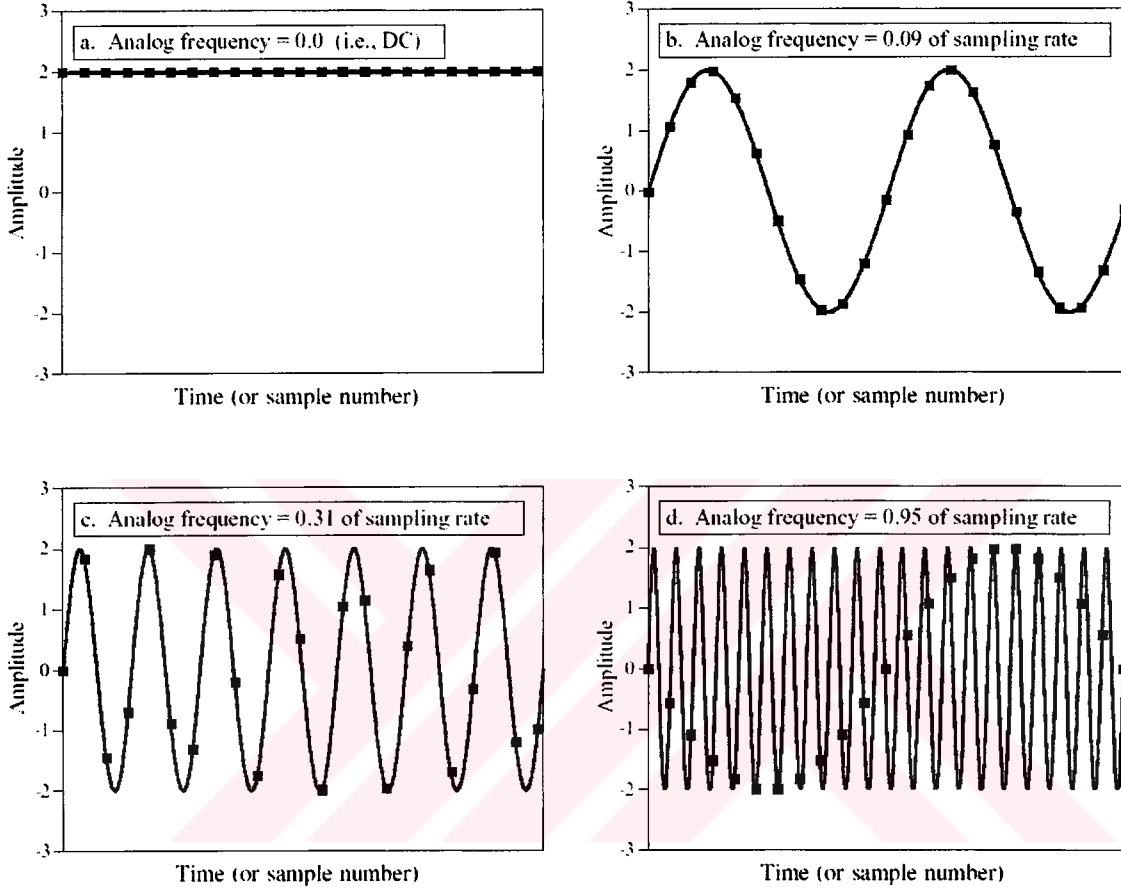
3.3.4.2 Örnekleme Teoremi

Tam ve doğru örneklemenin tanımı oldukça basittir. Eğer alınan örneklerden analog sinyal tekrar elde edebiliyorsa, doğru örnekleme yapılmış demektir. Aşağıdaki şekilde (şekil 3.3) çeşitli işaretlerin dijital hale dönüştürülmeden önceki ve sonraki durumları gösterilmiştir. Sürekli çizgiler ADC girişindeki analog işareti ve kare şeklindeki noktalar ise ADC çıkışındaki dijital işareti belirtmektedir. Şekil 3.3(a)'da analog işaret sabit genlikli bir DC ya da bir başka deyişle sıfır frekanslı kosinüs gerilimdir. Örneklenen noktaların hepsi aynı genliktedir yani işareten alınmış örnekler, analog sinyalin tüm karakteristik özelliklerini aynen taşımaktadır ve bu örnekler kullanılarak analog işarete geri dönülebilir. Yukarıda yaptığımız tanıma göre yapılan örnekleme doğru bir örneklemedir.

Şekil 3.3(b)'de gösterilen sinüs dalga şekli 0,09 örnekleme frekansı oranına sahiptir. Buna göre 90 periyot/saniye'lik bir sinüs dalgasından saniyede 1000 örnek alınacaktır. Bir başka deyişle sinüs dalgasının bir periyodu boyunca 11,1 adet örnekleme yapılmıştır. Bu örneğin bir öncekinden daha karmaşık olduğu açıktır. Çünkü alınan örnekler doğrularla birleştirilirse analog işaret tam olarak elde edilemez. Ancak bu örnekleme de doğru örnekleme olarak kabul edilebilir. Çünkü, elde edilen örneklerden başka türlü bir sinüs ya da sinüs türevi sinyal elde edilemez. Dolayısıyla yapılan örneklemeden elde edilen değerler bir sinüs dalgasına karşılık gelmektedir ve doğru bir örnekleme yapılmıştır.

Şekil 3,3(c)'de sinüs dalgasının frekansı daha da artırılarak örnekleme daha zor bir işaret

elde edilmiştir. Şekildeki örnekleme frekansı oranı 0,31'dir. Bu da periyot başına 3,2 örneğe karşılık gelir. Alınan örnekler oldukça seyrek ve analog sinyal trendini tam olarak izlememektedir. Ancak bu örnekleme de bir öncekine benzer nedenden dolayı doğru örnekleme olarak kabul edilebilir.



Şekil 3.3 : Çeşitli dalga şekilleri için yapılan örneklemler

Elde edilen örnekler analog sinyalin aynısını üretmek için gerekli tüm bilgileri içermektedir.

Şekil 3.3(d)'de analog işaret örnekleme frekansı oranı 0,95'dir. Yani her periyotta 1,05 örnek alınmıştır. Ancak yapılan bu örnekleme doğru bir örnekleme değildir. Çünkü orjinal sinüs dalgasının frekansı ile örnekleme sonusunda elde edilen işaretin frekansı birbirinden farklı ve örnekleme sonucu elde edilen işaretin frekansı daha küçüktür. Elde edilen örneklerden analog sinyali elde etmek mümkün değildir.

İşte yukarıda bahsedilen manik yani örnekleme teoremi, dijital işaret işlemenin yapı taşı oluşturur. Bu teorem genellikle Shannon örnekleme teoremi ya da Nyquist örnekleme teoremi olarak adlandırılır ve 1940'lı yıllarda ortaya atılmıştır. Örnekleme teoremine göre : Eğer

analog sinyalin frekansı örneklenen noktaların (örnekleme) frekansının %50'sinden daha büyük değilse sürekli bir sinyal uygun şekilde örneklenebilir. Örneğin : saniyede 2000 örnek alıyorsak analog işaretin frekansı 1000Hz'den küçük olmalıdır. ($f_0 \geq f$)

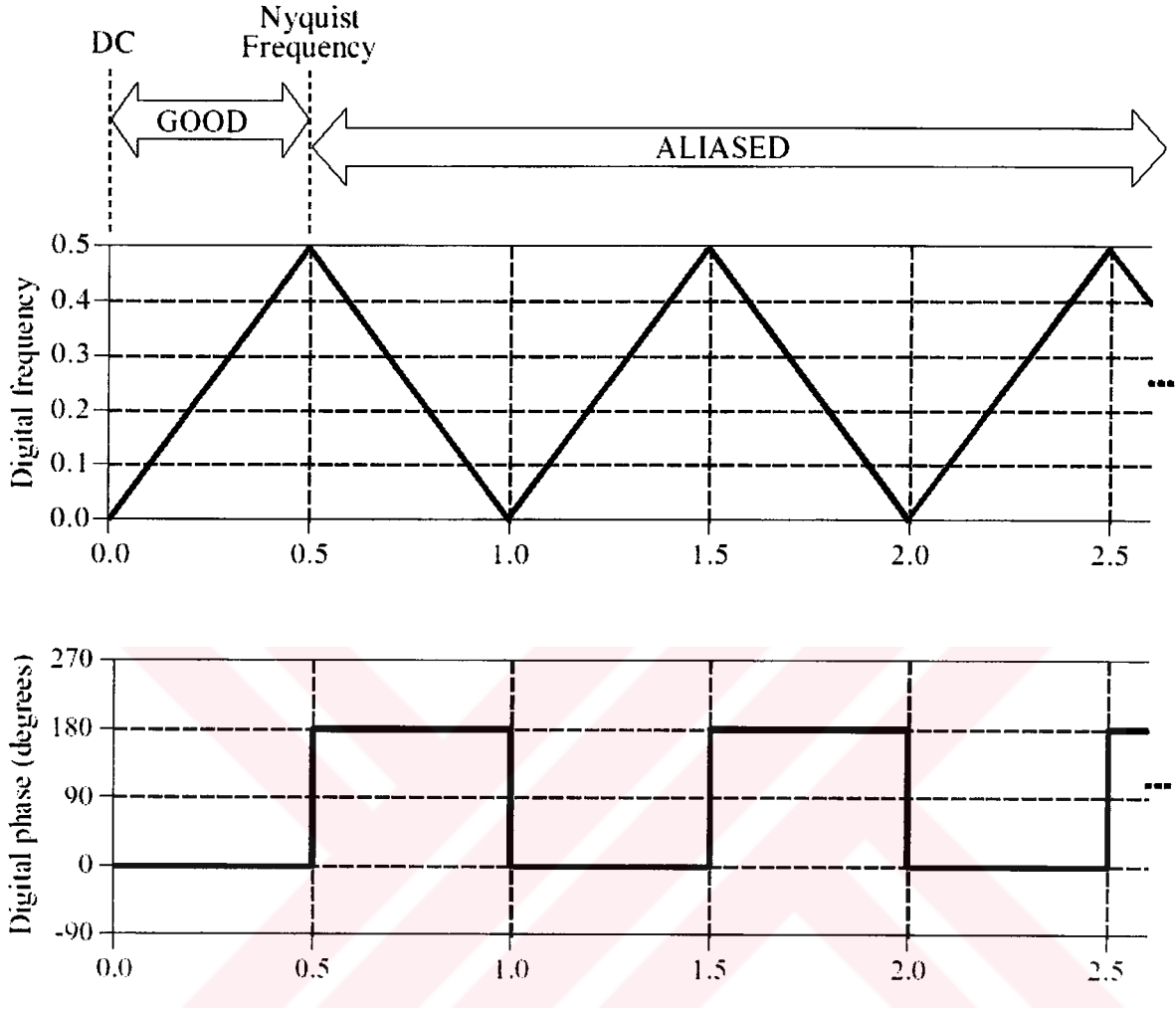
Örnekleme teoreminden bahsedilirken Nyquist frekansı ve Nyquist oranı sıkça kullanılan terimlerdir. Ancak ne yazık ki bu terimlerin anlamları standart değildir. Terimleri daha iyi anlayabilmek için DC ile 3kHz arasında frekansa sahip olan bir analog işareti ele alalım. Bu işareti doğru olarak örnekleymek için örnekleme frekansının en az 6000 örnekleme/saniye ya da örnekleme oranının 2 olması gerekir. Ancak örnekleme frekansını 8000 örnekleme/saniye olarak seçtiğimizi düşünelim. Bu durumda dört önemli frekans söz konusudur.

- 1) İşaretin maksimum frekansı 3kHz
- 2) Maksimum frekansın iki katı 6kHz
- 3) Örnekleme frekansı 8 kHz
- 4) Örnekleme frekansının yarısı 4 kHz

Bunlardan hangisinin Nyquist frekansı ve hangisinin Nyquist oranı olduğu neyi aradığınıza bağlıdır. Yukarıda tüm olası değerler verilmiştir. Biz burada her ikisi için de örnekleme oranının yarısı tanımını kullanacağız.

Bir işaretin frekansının örnekleme ya da bir nonlineer olay sonucu olarak değişmesi sürecine Aynalanma adı verilir. Aşağıdaki şekilde (şekil 3-4) hangi frekansların aynalanma esnasında değiştirildiği gösterilmiştir. Burada hatırlanması gereken nokta dijital işaretin, örnekleme oranının yarısından daha büyük bir frekansı içermemesi gerektiğidir. Sürekli işaretin frekansı , Nyquist oranının altında olduğunda örneklenen verinin frekansı uygundur. Bununla birlikte örneklenen işaretin frekansı Nyquist oranının üzerinde olduğunda frekans uygundur. Aşağıdaki şekilde zikzak biçimindeki çizgi ile gösterildiği gibi, Nyquist oranın üzerinde bir frekansı olan her sürekli işaret, sıfır ile örnekleme oranının yarısı arasındaki bir dijital frekansa karşılık gelir. Eğer zaten bu düşük frekansta bir sinüsoidal işaret varsa, bilgi kaybından dolayı aynalanma işareti ona eklenecektir. Aynalanma'nın iki kötü tarafı vardır. Yüksek ya da düşük frekansta da bilgi kaybına neden olur. 0,2 oranında örneklenmiş bir dijital işaret verildiğini düşünelim. Eğer bu işaret doğru olarak örneklenmiş olsaydı, orjinal analog işaret de 0,2 frekansına sahip olurdu. Ancak örnekleme sırasında aynalanma oluşursa dijital işaretin frekansı olan 0,2 analog sinyalin sonsuz değişik frekansından kaynaklanmış

olabilir.



Şekil 3.4 : Örnekleme sırasında analog frekansın dijitale dönüşümü

Sadece aynalanma, örnekleme sırasında frekans ve fazı değiştirebilir. Örneğin şekil 3.3d deki aynalanmış işarete tekrar bakalım. Bu işaret orijinal analog sinyalin tam tersidir. Biri sinüs dalgası şeklinde iken diğeri negatif sinüs şeklindedir. Diğer bir deyişle aynalanma frekansını değiştirmekle kalmaz 180° 'lik bir faz kaymasına da neden olabilir. Sadece iki değişik faz kayması durumu mümkündür : 0° ve 180° . 0° 'lik faz kayması analog frekansın 0 ile 0,5 , 0,1 ile 1,5 , 2,0 ile 2,5 vs. arasında olduğunda meydana gelir. 180° 'lik faz kayması ise analog frekansın 0,5 ile 1,0 , 1,5 ile 2,0 ve 2,5 ile 3,0 v.s. arasında olduğu durumlarda oluşur.

Şimdi de aynalanmanın nasıl oluştuğuna ve sürekli işaretin ayrık işarete dönüştürüldüğünde bilginin nasıl değiştiğine bir göz atalım. Öncelikle sürekli bir işaret ile sayı dizileri oldukça farklı şeylerdir. Dolayısıyla bunları karşılaştırmak, problemi oldukça zorlaştırmaktadır. Bunun için impuls ya da darbe dizisi adı verilen teorik kavramı incelememiz gerekir.

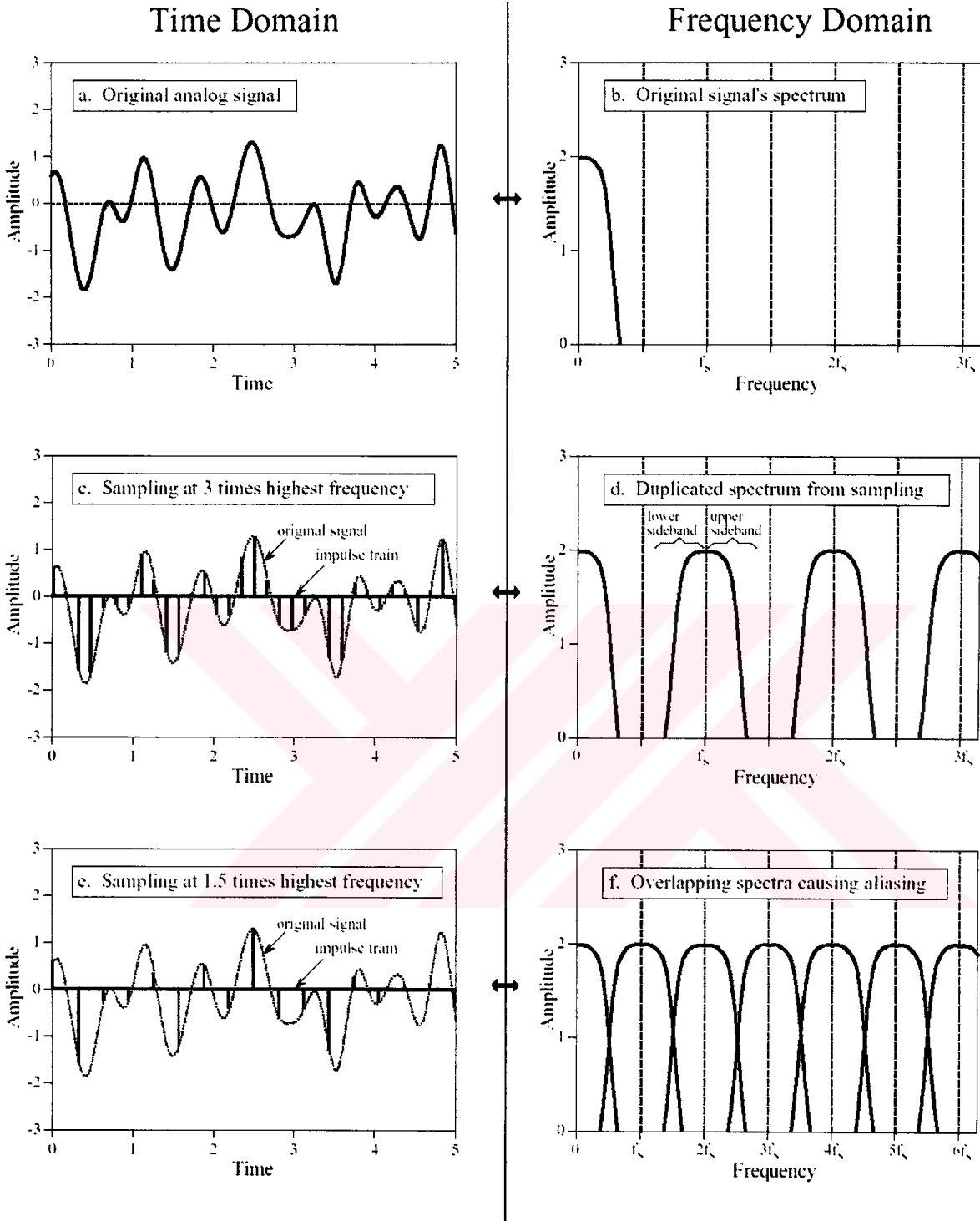
Şekil 3.5a'da analog bir işaret örnek olarak verilmiştir. Şekil 3.5(c)'de ise darbe dizisi ile işaret örnekleme gösterilmiştir. Darbe dizisi birbirine yakın ve orjinal işaretin örnekleme anında aldığı değere eşit genlikteki impulslardan oluşur. Her darbe birbirine sonsuz küçük uzaklıktadır. Her bir örnekleme alınırken geçen süre boyunca dalga şeklinin değişimi sıfırdır. Bu arada darbe dizisinin teorik bir kavram olduğunu unutmamalıyız, uygulamada böyle bir elektronik devre yoktur. Hem orjinal sinyal hem de darbe dizisi sürekli bir dalga şekline sahiptir. Dolayısıyla artık bu ikisi aynı cinsten olduğuna göre karşılaştırmak oldukça kolaylaşmıştır.

Şimdi ayrık işaret ile darbe dizisi arasındaki ilişkiyi incelememiz gerekir. Bilgi içeriği bakımından ikisi özdeştir. Birisi bilindiğinde ötekini hesaplamak çok kolaydır. Böylece şekil 3.5(a)'dan şekil 3.5(c)'ye nasıl geçildiğini açıklamış olduk.

Şekil 3.5'in sol tarafında üç adet sürekli dalga şekli verilmiştir ve bunlara ait frekans spektrumu sağ taraftadır. Bu elektronik derslerinden bildiğimiz bir kavramdır. Her dalga şekli değişik frekans ve genlikteki sinüs dalgalarının toplamı olarak ifade edilebilir.

Şekil 3.5(a)'da örnekleme istenilen analog işaret verilmiştir. Şekil 3.5(b) ise analog işarete ait frekans spektrumudur. Frekans bileşenleri 0 ile $0,33f_s$ arasındadır. Burada f_s kullanmak istenilen örnekleme frekansıdır. Örneğin ; bu 3,3kHz'in üzerinde frekansa sahip işaretleri filtrelenmiş olan bir konuşma sinyali olabilir. Buna karşılık olarak f_s örnekleme frekansımız 10 kHz olacaktır.

Şekil 3.5(a)'daki işareti darbe dizisi yöntemini kullanarak örneklediğimizde şekil 3.5(c)'deki işareti elde ederiz ve buna ait frekans spektrumu Şekil 3.5(d)'deki gibi olacaktır. Bu spektrum orjinal analog işaretin frekans spektrumunun bir kopyasıdır. f_s 'in tam sayı katı olan her örnekleme frekansıda ($f_s, 2f_s, 3f_s, 4f_s, \dots$) orjinal analog işaretin sağa ya da sola ötelenmiş bir spektrumunu oluşturacaktır. f_s için elde edilen kopya üst yanband ve ötelenmiş kopyalar ise alt yanbandlar olarak adlandırılır. Örnekleme işlemi yeni frekanslar üretmiştir. Bunun yanında yapılan bu örnekleme doğru bir örneklemedir. Çünkü, Şekil 3.5(c)'deki işaret 3.5(a)'daki analog işarete $1/2f_s$ 'nin üzerindeki tüm frekanslar filtrelenerek ederek dönüştürülebilir. Bu da analog bir alçak geçiren filtre ile darbe dizisinin analog işarete dönüştürülebileceği anlamına gelir.



Şekil 3.5 : Zaman ve frekans domenlerinde örnekleme teoreminin ifadesi

Şekil 3.5(e)'de çok düşük örnekleme oranı nedeniyle doğru olmayan örnekleme için bir örnek verilmiştir. Analog işaret hala 3,3kHz'in üzerindeki frekansları içermektedir. Ancak örnekleme frekansı 5kHz'in altındadır. f_s , $2f_s$, $3f_s$ boyunca yatay eksenin şekil 3.5(d)'de şekil 3.5(f)'ye nazaran daha yakın konumlandığına dikkat ediniz. Frekans spektrumu, (f), sorunu açıkça ortaya koymaktadır. Spektrumun kopyalanan kısımları bandı örnekleme frekansının 0

ile $\frac{1}{2}$ 'si arasında işgal etmiştir. Bununla birlikte şekil 3.5(f)'ten de görüleceği gibi kesişen frekanslar kendi özelliklerini kaybetmemişlerdir. Ancak kesişen frekansları ayırmak için herhangi bir yöntem mevcut değildir. Analog işaretin içerdiği bilgi kaybolmuştur ve orjinal işaret yeniden oluşturulamaz. Frekansların kesişmesi, analog işaretin frekansı örnekleme frekansının yarısından büyükse meydana gelir. Bu da örnekleme teoremini ispatlar niteliktedir.

3.3.4.3 Dijital Analog Dönüşüm

Teorik olarak dijital – analog dönüşüm için en kolay yol, örnekleri hafıza elemanından okumak ve onları darbe dizisine Şekil 3.6(a)'da görüldüğü gibi çevirmektir. Şekil 3.6(b)'de analog işarete karşılık gelen frekans spektrumu verilmiştir. Daha önce tanımlandığı gibi orjinal analog işaret, darbe dizisini alçak geçiren bir filtreden geçirmek ve örnekleme frekansının yarısına eşit frekanstaki işaretleri kesmekle mükemmel bir şekilde yeniden oluşturulabilir. Diğer bir deyişle, orjinal işaret ve darbe dizisi Nyquist frekansının altında özdeş frekans spektrumlarına sahiptir. Daha yüksek frekanslarda ise orjinal analog işaret herhangi bir bilgi içermezken darbe dizisi bu bilginin kopyasını içerir.

Bu metod matematiksel olarak kusursuz görünse de gerekli darbeleri gerçek hayatta üretmek oldukça zordur. Bu sorunu bertaraf edebilmek için tüm DAC'ler diğer bir örnek alınana kadar son değer üzerinde işlem yaparlar. Buna zeroth – order hold denir. ADC'lerde ise buna karşılık sample and hold prensibi kullanılır. Bir first – order hold noktalar arasını doğrularla birleştirirken second – order hold bu işlem için parabolleri kullanır. Zeroth – order hold merdiven şeklinde bir dalga şekli oluşmasına neden olur. Bu şekil 3.6(c)'de açıkça görülebilir.

Frekans domeninde, zeroth – order hold'un sonucu olarak darbe dizisinin spektrumu çoklu hale gelmiştir. Şekil 3.6(d)'de kalın çizgilerle bu etki gösterilmiş ve aşağıdaki eşitlik ile de ifade edilmiştir.

$$H(f) = \left| \frac{\sin(\pi f / f_s)}{\pi f / f_s} \right| \quad [3.11]$$

yüksek frekans genlik indirilmesi nedeniyle zeroth – order hold. Bu eğri şekil 3.6(d)'de çizilmiştir. Örnekleme frekansı f_s olarak ifade edilmiştir. $f = 0$ ise $H(f) = 1$ dir.

Bu eşitlik $\sin(\pi x) / (\pi x)$ 'nin genel biçimidir ve sinc işlevi ya da sinc(x) olarak adlandırılır. Sinc işlevi dijital işaret işlemede oldukça yaygındır. Bunun yanında zeroth – order hold eşit örnekleme periyoduna sahip bir darbe dizisi ile bir üçgen darbenin birlikte convolution'u

olarak da düşünülebilir. Bu sonuç frekans domeninde üçgen darbenin fourier dönüşümü ile çarpılır. Şekil 3.6(d)'de açık renkli olarak çizilen eğri darbe dizisinin frekans spektrumu ve koyu rekte çizilen eğri ise sinc'nin değişimidir. Zeroth – order hold işaretin frekans spektrumu, bu iki eğrinin çarpımına eşittir.

Analog filtreler Zeroth – order hold işaretinin, Şekil 3.6(c), yeniden oluşturulmuş analog işarete, Şekil 3.6(f), dönüştürülmesinde,

- 1) Örnekleme frekansının yarısından daha büyük frekanstaki işaretlerin kesilmesinde
- 2) İki taraflı Zeroth – order hold etkisiyle frekansların (örneğin: $1/\text{sinc}(x)$) arttırılmasında kullanılır.

Şekil 3.6(e) 'de analog filtrenin frekans cevabı verilmiştir. $1/\text{sinc}(x)$ oranındaki frekans arttırımının etkisi dört yolla ortadan kaldırılabilir.

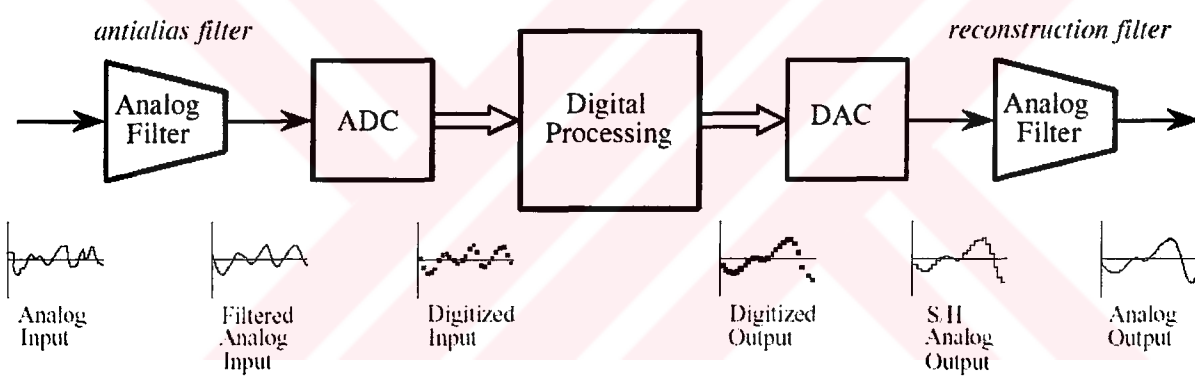
- 1) Bu arttırım yokmuş gibi davranıp sonuçlarını kabul etmek
- 2) $1/\text{sinc}(x)$ oranında genlik azaltıcı bir analog filtre tasarlamak
- 3) Burada bahsettiğimiz çok oranlı (pahalı) tekniği kullanmak
- 4) DAC'den önce yazılım yardımıyla düzeltme yapmak

Belirli bir miktar bilgiyi dijital işaretle taşımamanın iki yolu vardır :

- 1) Alınan her örnekteki bit sayısı bağımlı değişkenin çözünürlüğü tarafından sınırlanır . Bu metod işaret üzerinde küçük değişikliklere neden olur ve kuantalama gürültüsünü (hatasını) yok edebilir.
- 2) Örnekleme frekansı (oranı) bağımsız değişkeni sınırlar. Bu yöntem kullanıldığında analog işarettaki birbirine yakın bazı değişiklikler, iki örnekleme anı arasında kalıp kaybolabilir. Bu da örnekleme frekansının yarısından büyük frekansa sahip işaretlerin kaybolacağı anlamına gelir.

Analog sinyallerin sürekli parametreleri kullandığı ve böylece hem bağımsız hem de bağımlı değişkenlere göre son derece iyi bir çözünürlüğü olduğu düşünülür. Ancak bu yanlıştır. Analog işaretler de dijital işaretlerde karşılaşılan gürültü ve band genişliği (işaretteki en büyük frekans değeri) sorunları nedeniyle sınırlanmıştır. Analog işarettaki gürültü, sinyalin genliğinin ölçümünü sınırlar. Dijital işarete ise bunu sadece kuantalama gürültüsü yapar. Keza analog işarete var olan birbirine çok yakın değişiklikleri ayırma kabiliyeti işarettaki en büyük frekans değerine bağlıdır. Bunu daha kolay anlayabilmek için bir örnek üzerinde düşünelim. Bir analog işaret üzerinde birbirine çok yakın iki adet darbe bulunuyor olsun. Biz eğer işareti alçak geçiren bir filtre üzerinden (yüksek frekanslı işaretleri yok edebilmek için) süzersek, darbeler bulanıklaşacak ve bir darbe gibi görünecektir. Bununla birlikte DC ile 10 kHz arasında değişen bir analog işaret 20 kHz frekansında örneklenmiş dijital işaretle tamamen aynı çözünürlüğe sahiptir. Bu örnekleme teoremi ile sabittir.

3.3.4.4 Veri Dönüşümü İçin Analog Filtreler

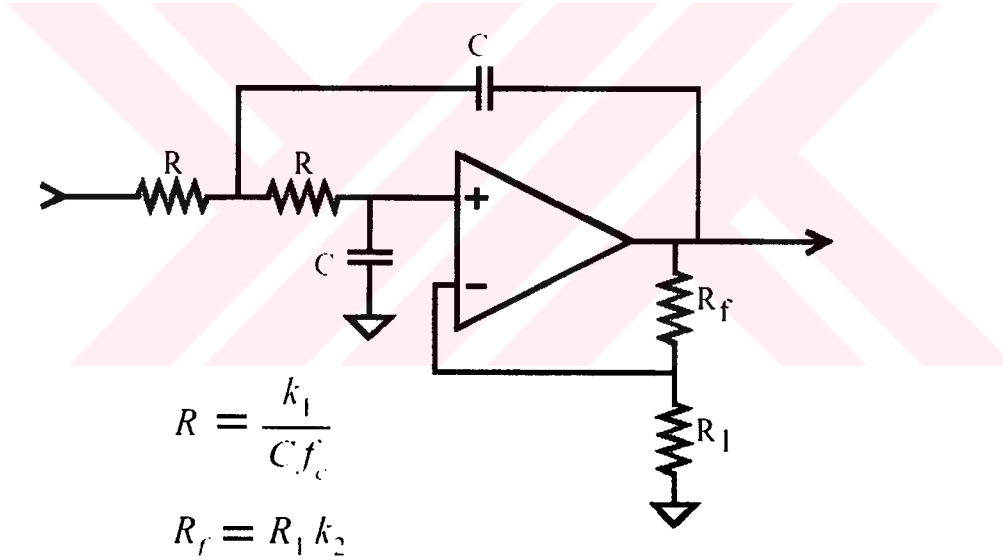


Şekil 3.7 : Basit bir dijital işaret işleme sisteminin blok diyagramı

Yukarıdaki şekilde (Şekil 3.7) örnekleme teoremine uygun bir DSP sistemine ait blok diyagram verilmiştir. Analog işaret dönüşüm işlemine girmeden önce elektronik bir alçak geçiren filtre ile Nyquist frekansı üzerinde kalan işaretlerin süzülmesi sağlanmıştır. Bu işlem örnekleme esnasında aynalanma oluşmasını engellemek için yapılmıştır ve filtre antialiasing filtre olarak adlandırılır. Diğer taraftan sayısallaştırılmış işaret bir DAC'den geçirilmiştir ve daha sonra alçak geçiren bir filtreden süzülerek Nyquist frekansına ayarlanmıştır. Buradaki çıkış filtresi de yeniden düzenleme (reconstruction) filtresi olarak adlandırılır. Ayrıca bu filtre daha önce bahsedilen zeroth – order –hold frekans artırımını da sağlayabilir. Ancak bu basit sistemde kullanılan elektronik filtrelerin meydana getirdiği kısıtlamalar önlemeye çalışıklarından çok daha büyük ve ciddi olabilir.

Bu nedenle analog filtrelerin yapılarının bilinmesi ve bunların uygulanması, başarılı bir sistem oluşturabilmek için önemlidir. Öncelikle karşılaştığımız her sayısallaştırılmış işaret karakteristiği onu elde ederken kullandığımız antialiasing filtresinin çeşidine bağlı olacaktır. Eğer bu filtrelerin yapısı yeteri kadar bilinmezse dijital işaretin yapısı anlaşılabilir. İkinci olarak, dijital işaret işlemenin geleceği artık donanımdan yazılıma doğru kaymaya başlamıştır. Ancak donanımı bilmeden yazılımı geliştirmek imkansızdır. Üçüncü olarak, bir çok DSP günümüzde dijital filtrelerde kullanılmaktadır. Dolayısıyla genel yöntem öncelikle bir analog filtre ile başlamak ve daha sonra bununla aynı işi yapabilecek bir yazılım tasarlamaktır.

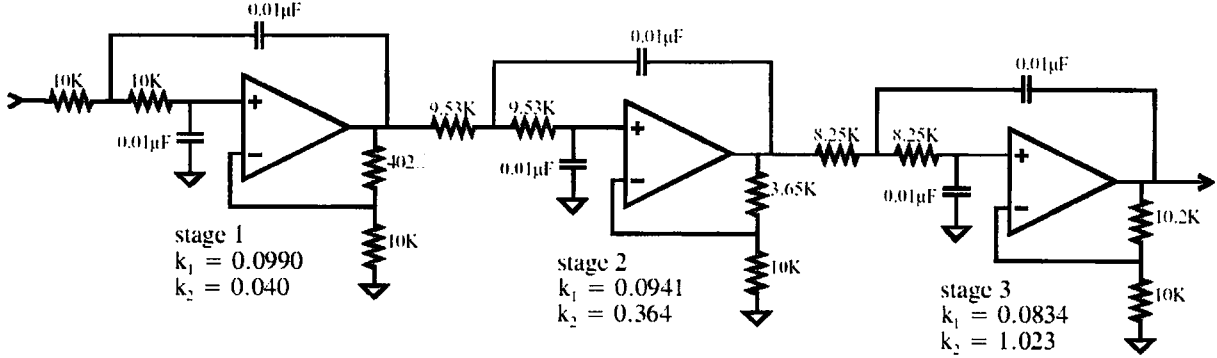
Üç çeşit analog filtre yaygın olarak kullanılır : Chebyshev, Butterworth ve Bessel (ya da Thompson). Bu filtrelerden her biri değişik performans parametrelerini optimize etmekte kullanılır. Her bir filtrenin karmaşıklığı kutup ve sıfırlarının sayısı ile ayarlanabilir. Filtredeki daha çok kutup daha fazla elektronik eleman kullanılmasını gerektirir fakat, daha iyi bir performans sağlar.



Şekil 3.8 : Sallen – Key filtresinin yapısı

Şekil 3.8’de yaygın olarak kullanılan bir analog filtre yapısı verilmiştir. Sallen – Key filtresi olarak adlandırılan bu filtre iki kutuplu alçak geçiren bir filtredir. Bu filtre temelde üç değişik tipte yapılandırılabilir. Çizelge 3.6’da kullanılan direnç ve kondansatörlerin seçimi için gerekli bilgiler verilmiştir. Örneğin iki kutuplu 1 kHz’lik bir Butterworth filtresi dizayn etmek için Tablo 3.1 ‘den $k_1=0,1592$ ve $k_2=0,586$ olarak bulunabilir. Buradan da $R_1=10k$ ve $C = 0,01\mu F$ olarak seçilebilir. R ve R_f değerleri de 15,95K ve 5,86K olarak hesaplanabilir. R ve R_f dirençleri %1 toleranslı olmalıdır. Pratik olarak $R=15,8K$ ve $R_f = 5,90K$ olarak kullanılabilir.

Tüm devrenin frekans kazancı filtrenin kesim frekansından 30 ile 100 kat daha büyük olduğu sürece özel bir op-amp kullanmak gerekli değildir. Bu filtrenin kesim frekansı 100kHz'in altında bulunduğu sürece sağlanması kolay bir şarttır.



Şekil 3.9 : Altı kutuplu bir Bessel filtresinin yapısı

Dört, altı ve sekiz kutuplu filtreler bu devrelerin iki, üç ve dördünün kaskad olarak bağlanmasıyla teşkil edilebilir. Örneğin şekil 3.9'da altı kutuplu bir Bessel filtresi üç adet devrenin kaskad olarak bağlanmasıyla oluşturulmuştur. Her bir devrenin k_1 ve k_2 sabitleri farklı olup çizelge 3.1'den elde edilmiştir.

Bu tip devreler küçük miktardaki üretimler ve R&D uygulamalarında yaygın olarak kullanılır. Bununla birlikte büyük miktardaki üretimlerde filtrelerin entegre devre şeklinde üretilmesi gerekir. Burada dirençleri silikon malzemeden imal etmek aşılması gereken bir problem olarak karşımıza çıkar. Anahtarlamalı kondansatör filtreleri kullanılarak bunun üstesinden gelinebilir. Şekil 3.10'da böyle bir filtrenin yapısı ve karakteristiği klasik RC filtresiyle karşılaştırması olarak verilmiştir.

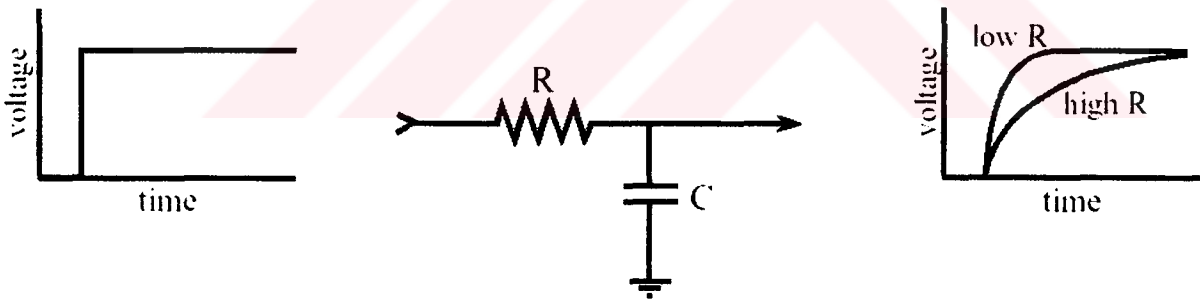
Anahtarlamalı kondansatör filtresi, klasik direç kondansatör filtresinin görevini iki kondansatör ve bir elektronik anahtarlama elemanı kullanarak gerçekleştirir. Devredeki ikinci kondansatör değerinin 0,01 katı değerindedir. Anahtarlama elemanı kesim frekansının yaklaşık 100 katı bir frekansla açma – kapama yaparak küçük değerli kondansatör üzerinden giriş ile çıkışı birbirine bağlar. Anahtarlama elemanı giriş ile küçük kondansatörü kısa devre ettiğinde, bu kondansatör girişteki gerilim seviyesine kadar hızla şarj olur. Anahtarlama elemanı çıkış ile küçük kondansatörü kısa devre ettiğinde ise küçük kondansatör yükünü hızla diğer kondansatöre aktarır. Direnç kullanıldığında bu transferin hızı ve miktarı direncin değerine bağlı olduğu halde anahtarlama elemanı kullanıldığında bu parametreleri küçük değerli kondansatör ve anahtarlama frekansı belirleyecektir. Bu sonuçlar anahtarlamalı

kondansatör filtresi için oldukça önemlidir. Yani bu filtrenin kesim frekansı direkt olarak, anahtarlama elemanını tetikleyen darbenin frekansı ile orantılıdır. Anahtarlama kondansatör filtresinin bu özelliği onu birden fazla örnekleme frekansının kullanıldığı veri toplama sistemleri için oldukça kullanışlı kılar. Ayrıca bu filtreler ucuz ve kullanımı da son derece kolaydır. Örneğin sekiz kutuplu bir filtre, sekiz bacaklı bir entegre devre şeklinde ve yaklaşık

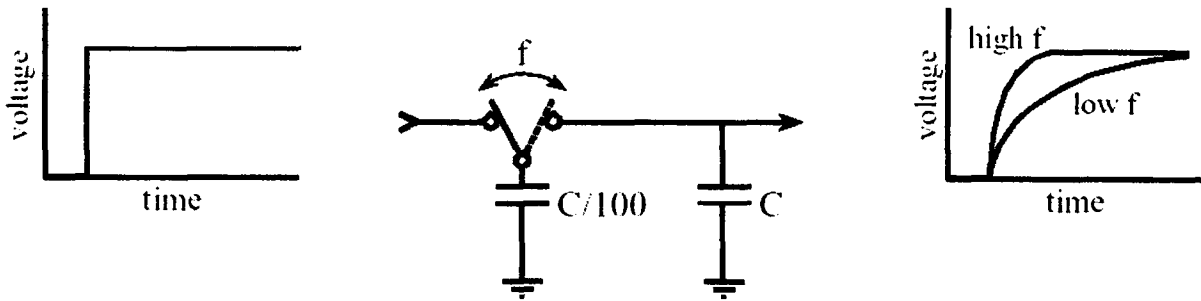
Çizelge 3.6 : Çeşitli filtrelerin k_1 ve k_2 sabitleri

# poles		Bessel		Butterworth		Chebyshev	
		k_1	k_2	k_1	k_2	k_1	k_2
2	stage 1	0.1251	0.268	0.1592	0.586	0.1293	0.842
4	stage 1	0.1111	0.084	0.1592	0.152	0.2666	0.582
	stage 2	0.0991	0.759	0.1592	1.235	0.1544	1.660
6	stage 1	0.0990	0.040	0.1592	0.068	0.4019	0.537
	stage 2	0.0941	0.364	0.1592	0.586	0.2072	1.448
	stage 3	0.0834	1.023	0.1592	1.483	0.1574	1.846
8	stage 1	0.0894	0.024	0.1592	0.038	0.5359	0.522
	stage 2	0.0867	0.213	0.1592	0.337	0.2657	1.379
	stage 3	0.0814	0.593	0.1592	0.889	0.1848	1.711
	stage 4	0.0726	1.184	0.1592	1.610	0.1582	1.913

Resistor-Capacitor



Switched Capacitor



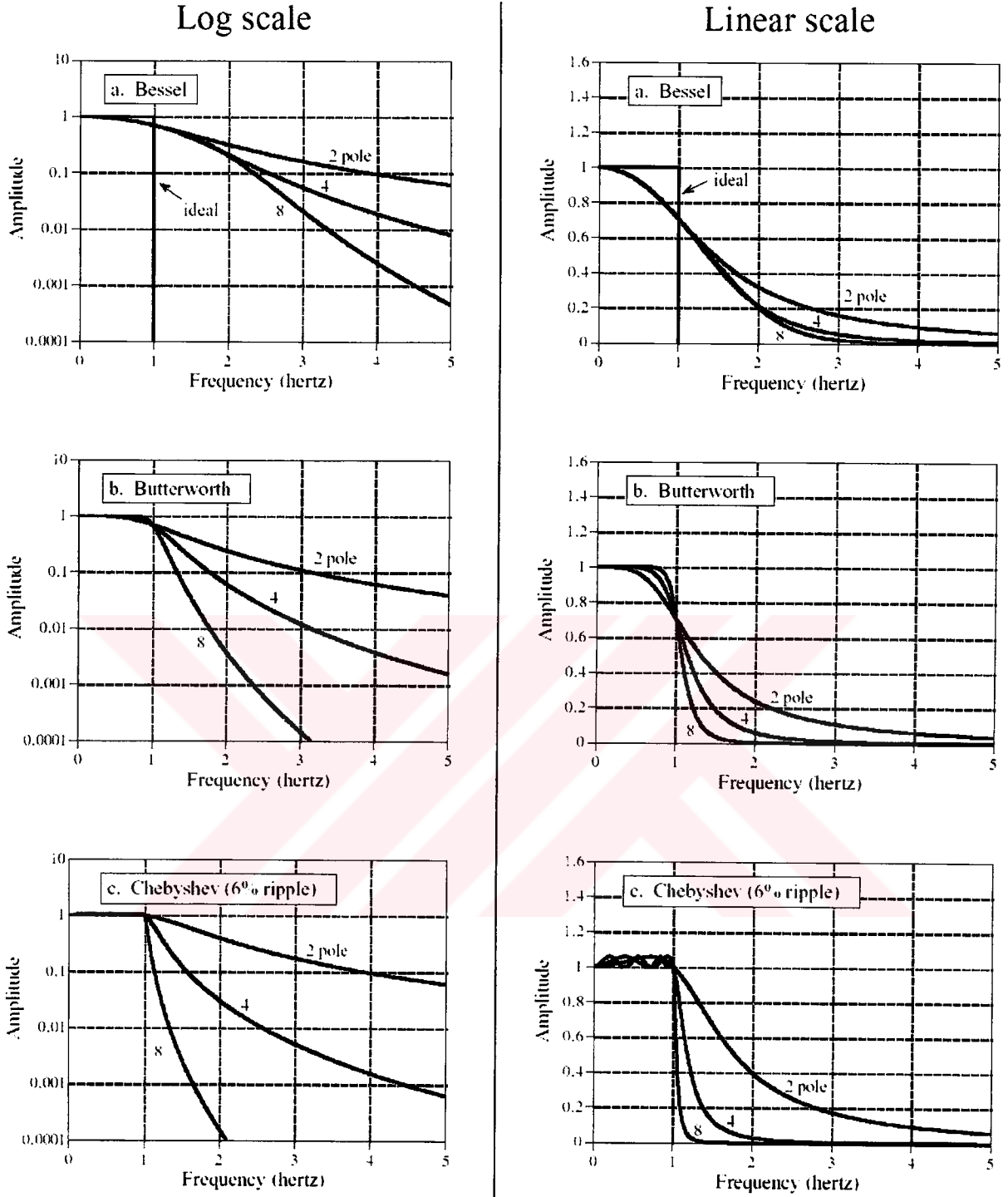
Şekil 3.10 : Alçak geçiren RC filtresi ve anahtarlama kondansatör filtresi

5 – 10 USD fiyatına satılabilmektedir.

Şimdi de bu üç klasik tip filtrenin karakteristiklerine bir göz atalım. Öncelikle kesme frekansının keskinliğine göre karşılaştıralım. Bir alçak geçiren filtre kesim frekansının üzerinde frekansa sahip tüm işaretleri engellemek, stopband, ve bunun altında frekansa sahip işaretleri de zayıflatmadan geçirecek, passband, şekilde tasarlanır. Şekil 3.11’de bahsedilen üç filtrenin frekans cevabı logaritmik düzlemde (dB) verilmiştir. Karşılaştırılan tüm filtreler 1Hz’lik kesme frekansına sahiptir. Fakat grafikler kullanılacak kesim frekansına göre ölçeklenmiştir. Sonuçlar incelenirse Chebyshev fitresinin açıkça en iyisi olduğunu görebiliriz. Daha sonra sıra ile Butterworth ve Bessel fitreleri gelmektedir. Bunun nedeni Chebyshev fitresinin roll-off olarak adlandırılan genliği mümkün olduğunca hızlı azaltma amacıyla tasarlanmış olmasıdır.

Ayrıca 12 bitlik bir sistemde 4096 adet ayrı durum vardır. 100 olarak seçilen zayıflatma faktörü bu durumda yetersiz kalabilir. Bir çok sistemde örnekleme frekansının 0,4 ile 0,5’i arasında kalan band filtrenin işaret zayıflatmasını arttırmak için kullanılamaz.

İdeal bir alçak geçiren filtrenin frekans cevabı tüm passband boyunca düz olmalıdır. Şekil 3.11’deki tüm filtreler bu açıdan bakıldığında mükemmel gözükmektedir. Ancak , bunun nedeni dikey eksenin logaritmik ölçekle ifade edilmiş olmasıdır. Grafikler Şekil 3.12’deki gibi lineer skalaya çevrildiğinde Chebyshev fitresinin passband dalgalanması açıkça görülmektedir. Bunun yanında filtredeki passband dalgalanmalarına daha fazla izin verilirse istenmeyen frekansların zayıflatılması daha hızlı gerçekleştirilir. Çizelge 3.1’deki veriler kullanılarak tasarlanan tüm Chebyshev filtreleri %6 civarında (0,5dB) bir passband dalgalanmasına sahiptir.



Şekil 3.11 : Çeşitli filtrelerin lineer ve logaritmik ölçekle çizilmiş frekans cevapları

Şekil 3.12 : Çeşitli filtrelerin lineer ve logaritmik ölçekle çizilmiş frekans cevapları

4. DSP GENEL YAPISI

4.1 DSP Genel Özellikleri

Kontrol algoritmalarının gerçek zamanda işlem yapabilmesi, dış dünyadan elde edilen fiziksel büyüklüklerin değerlendirilip aynı anda gerekli olan kontrol, parametrelerinin hesaplanması anlamına gelmektedir. Sistemler karmaşık bir yapıya bürünüp, algoritmalar daha kapsamlı hale geldikçe mikroişlemciler çözüm için yeterli olmamaya başlamışlardır. Tüketiciden gelen geri besleme doğrultusunda üretici firmalar uygulamaya özel bazı işlemciler üretmişlerdir. Bu işlemciler genel amaçlı işlemcilerden daha hızlı, fakat uygulamaya özel olduklarından kullanım alanı açısından sınırlı olmuşlardır. Örneğin son birkaç yıl içinde, Motorola firmasını ürettiği 80C196 işlemcisi motor kontroluna yönelik üretilmiş işlemcilerden biridir. Asenkron ve fırçasız doğru akım motorlarının kontrolu için özel olarak tasarlanmış bu işlemci, açık ve kapalı çevrim kontrol yapabilecek şekilde gerekli giriş ve çıkış birimlerini içermektedir. Kapalı çevrim kontrol için enkoder tarafından gönderilen işaretleri doğrudan doğruya işleyerek gereken PWM çıkışlarını üretmektedir. Yalnız motor kontrolunda gösterdiği bu yüksek performansı genel uygulamalar için gösterememektedir. Bu da yapı olarak yeni tip işlemcilerin ortaya konması gerekliliğini ortaya çıkarmıştır. Amaç, mikroişlemcilerin yararlarını yapısında bulunduran ve yetersiz kaldığı durumlarda yeni olanaklar sunan bir işlemci geliştirilmesi olmuştur.

Mikroişlemcilerin üretilmesinde kullanılan yarı iletken teknolojisi, gün geçtikçe yeni gelişmelere tanık olmaktadır. VLSI (Very Large Scale Integration) denilen üretim tekniği yardımıyla, çok daha hızlı ve güçlü yapılar, çok daha küçük alanlara sığdırılarak, üretilmeye başlanmıştır. İşlemcilerin güç kaybının düşük değerde olmasını sağlayan bu sistem, tasarımcıya karmaşık yapıların kolayca işleyebilmesi rahatlığını getirmiştir. Mikroişlemcilerin mimari yapısı, geliştiren kişinin adıyla anılarak Von Nuemann olarak adlandırılmıştır. Bu mimari yapıda veri ve adres aynı yollar üzerinden taşınmaktadır. Veri ve komutların saklanması için aynı bellek biriminin kullanılması, şişe ağzı olarak adlandırılan veri sıkışması problemlerine neden olacaktır. Komut, işleme girmek için çözümlendiğinde, işlenecek olan veri de aynı yol üzerinden taşınmaktadır. Bu da işlemlerin çok daha uzun sürede tamamlanmasına ve daha fazla saat çevrimi sürmesine neden olmaktadır. Örneğin 16 bitlik sayılarla işlem yaparken, 8 bitlik bir işlemci, sayıları iki parça olarak alıp işlemektedir. Bu da 16 bitlik bir işlemcinin yarı hızında çalışması anlamına gelmektedir. Sonraki nesil işlemcilerde, üniteler bağımsız olarak çalıştırılarak, hızı artırma yoluna gidilmiş ve kısıtlı bir

paralel çalışma sağlanmıştır. Veri taşınması, eklenen yeni komut ve yapı değişikliği ile tek saat çevriminde gerçekleştirilebilmektedir.

DSP sistemlerin mimari yapısı ise, Harvard mimarisi olarak anılmaktadır. Bu mimari yapının en önemli özelliği, işlemcinin veri ve adres yollarının birbirinden ayrı olması nedeniyle, sistemin çok daha hızlı işlem yapabilmesidir. Gerçek zamanda veri işleme için gereken hızlı işlem yapabilme yeteneği, bu sayede DSP 'lerde mimari yapıdan gelen bir özellik olarak sağlanmıştır. Bunun yanında, paralel işlem yapabilme özelliği, aynı anda birkaç komutun işlenerek hızının katlanmasını sağlamaktadır. Bir mikroişlemcinin işlem yapabilme kapasitesini gösteren komut sayısı ve adresleme modları DSP sistemlerinde diğer işlemcilerle göre daha kısıtlı gibi gözükse de, sistemin özelliklerini tam olarak kullanmayı sağlayan özel komutlar sayesinde, birçok işlem tek bir çevrim süresinde yapılabilmektedir. Özellikle kontrol sistemlerinde çok kullanılan çarpma ve toplama işlemlerinin donanım olarak gerçekleştirilmesi, hız yönünden büyük bir avantaj sağlamaktadır. Bir algoritma içinde tekrarlanan döngü işlemleri, DSP 'de bu işlemler için öngörülen komutlar sayesinde çok daha hızlı ve verimli olarak çözümlenmektedir. Donanım olarak yapıların birbirinden bağımsız olması ve ayrı yollar üzerinden veri ve adres iletilebilmesi, sistemin üstün özellikleri olarak tanımlanabilir.

DSP sistemi, hızlı ve doğru işlem yapabilme niteliklerini sağlaması için belirli özellikleri yapısında bulundurmalıdır. Bu özelliklerden ilki, sistem içinde bulunan üniteler arasındaki iletişimi sağlayan yolların yapısıdır. Adres ve veri yollarının ayrı olması yanında, bu yolların genişliği, sistemin matematik işlem yapma hızını ve doğruluğunu en çok etkileyen faktördür. Uygulamaya yönelik olarak, işlemlerde kullanılan sayıların yapısal özellikleri doğrudan doğruya yolların genişliğine bağlıdır. Genel olarak sabit nokta aritmetiği ile çalışan sistemlerde 16 bit, daha büyük sayılar kullanan kayan nokta aritmetiği ile çalışan sistemlerde 32 bit yol genişliği bulunmaktadır. Adres ve veri yollarının ayrı olması, sistemin paralel çalışmasına olanak sağladığı gibi, komutun gerektirdiği ünitelerin diğerlerinden bağımsız olarak işlemlerini sağlamaktadır.

Dijital sistemlerde işlem yaparken karşılaşılan en önemli problem, işlemcinin sayısal hatalarıdır. Çarpma ve toplama işlemleri sonrasında oluşan yuvarlatma ve kesme hataları, kontrol edilen sistem üzerinde büyük hatalara neden olabilir. Bunu önleyebilmek için, işlemcinin işlem yapabilme kapasitesinin yeterince büyük olması gerekmektedir. DSP 'lerde donanım olarak bulunan çarpım ünitesinin çıkışı; çarpma işleminden sonra bit sayısının iki katı bir sonuç oluşturmaktadır. Veri yolunun iki katı büyüklüğünde olan bu sayı ile, belirli bir

yuvarlatma hatası ile tek bir kelime boyutunda veya doğrudan çift-kelime boyutunda işlem yapılabilir. Çarpım veya MAC (Çarpım-akümülatör) ünitesinin taşma hatalarını önleyebilmesi için yeterli sayıda koruyucu biti olması gerekmektedir.

DSP 'nin program çevrimleri sırasında hangi adresteki işlemi gerçekleştireceğini otomatik olarak belirlemesi istenen bir özelliktir. Bu işlem aritmetik-lojik ünitesi (ALU) tarafından gerçekleştirilmesi mümkün iken, belirli bir gecikmeye neden olacağından, DSP içinde ayrı bir veri adres generatörü (DAG) bulunmaktadır. Adresleme işleminin önem kazandığı, sıralı adreslemenin yapılmadığı uygulamalarda bu ünite, bağımsız çalışan bir işlemci gibi, adresleri üretmektedir.

Algoritmanın işlemesi sırasında komutların sıralı ve doğru olarak işlemesi, komut sıralama ünitesi sayesinde olmaktadır. Bu ünitenin varlığı ile, komutların işlemesi sırasında, işlemcinin mikrokod olarak çevrim ve döngüleri kullanmasına gerek kalmamaktadır. Bu da komutların daha az çevrim ile daha kısa sürede tamamlanmasını sağlamaktadır.

DSP 'nin dinamik işlem aralığının geniş olması, algoritmanın işlemesi sırasında ölçekleme sorununu ortadan kaldırmaktadır. Fakat dinamik aralık yeterince büyük değilse ve algoritmanın işlem aralığı çok geniş ise, ölçeklendirme yapılmalıdır. Ünitelerde yapılan işlemler sonunda elde edilen verilerin genişliğinin, veri yolunun genişliğinden fazla olmasından dolayı oluşan hatalar, kaydırma ünitesi yardımıyla, ana işlemciye gerek duyulmadan, tek bir çevrim süresinde çözümlenmektedir. Kaydırma ünitesi aynı zamanda, çok bitli tek çevrim kaydırmalar, normalize / denormalize işlemleri, kayan nokta aritmetiği ile işlemlerde büyük kolaylık sağlamaktadır.

Daha önceki bölümlerde sözü edilen, dijital işaret işleme sırasında karşılaşılan sorunları çözmek için, DSP'ler çeşitli yöntemlerle en az hata ile işlem yapabilmelidir. Komut işlemesi sırasında komutların aynı anda aynı veri yollarını kullanma isteğinden doğan veri sıkışması, şişe ağzı olarak adlandırılmaktadır. Sıvının şişeden akarken, ancak şişenin ağzının genişliği kadar hızla akabileceği açıktır. Aynı sorunla işlemcilerin veri işleminde de karşılaşılmaktadır. Bu sorunu ortadan kaldırmak için işlemcinin, iş hattı denilen parçalara ayırarak çalışma veya paralel çalışma sağlayacak yapının kurulması gerekmektedir. Bu sayede komut çevrim süresi en az düzeyde olacak ve aynı anda birden fazla komut işlenebilecektir. İş hattı çalışma, işlemin parçalar ayrılarak, her bir parçanın bağımsız olarak işlenmesi prensibine dayanmaktadır. Parçalı çalışma aynı işlemin, parça sayısı ile ters orantılı olacak oranda kısa sürede tamamlanmasını sağlamaktadır. Bunun yanında, işlemlerin optimal olarak

parçalanması her zaman söz konusu olmayabilir. Fazladan eklenen kütükler işlemin hızını arttırmaya yarayacaktır. Örneğin çarpma işleminde, çarpanların ayrı yollar üzerinden getirilmesi, çarpma işleminin süresini yarı yarıya azaltmaktadır. Sistemde iş hattı denilen yapının olması aynı anda birden fazla komutun işlenmesine izin vermektedir. Komut saat çevrimini en düşük seviyesine indirirken, işlemcinin çıkışından en yüksek performansın alınmasına olanak sağlamaktadır. Birinci nesil DSP 'lerde iki seviyeli bir iş hattı çalışması varken, ikinci nesil işlemcide, üç seviyeli, üçüncü nesil işlemcide dört seviyeli çalışma söz konusudur. Bir komut saat çevriminde üç komutun aynı anda işleyebilmesi şekil 4.1 'de gösterilmiştir. Aynı anda üç değişik komut aktif durumdadır. N. komutun işlemeye başlanması sırasında, bir önceki komut N-1 çözümlenmiş olmakta ve daha önceki N-2 komutu bu sırada işlemektedir.

Fetch (Alma)	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8
Decode (Çözme)	—	I_1	I_2	I_3	I_4	I_5	I_6	I_7
Execute (yürütme)	—	—	I_1	I_2	I_3	I_4	I_5	I_6

I_n : n. komut

Şekil 4.1 : Üç Seviyeli iş hattı işlemesi

İş hattı işleminin yetersiz kaldığı durumlarda paralel çalışma alternatif bir çözüm olmaktadır. Bir işlemin alt parçalara ayrılıp işlenmesi yerine, paralel işlemciler ile aynı işlemin yapılması çok daha iyi sonuç vermektedir. Bu tür çalışma ise, donanımın karmaşıklaşmasına ve fiyatının katlanmasına neden olacaktır. İşlemlerde paralel çalışmanın uygulanabilmesi, ancak işlemler birbirinin yerine geçebilir yapıda ise verimli olmaktadır. Bunun yanında, paralel çalışma sırasında faz gecikmesi veya paralel çalışan hatların birinin yanlış işlem yapması, bütün sistemi etkileyecektir. Bu çeşit problemleri ortadan kaldıracak olan protokol yapılarının tam olarak kurulması gerekmektedir. Paralel çalışma ve iş hattı çalışma koşullarında, komut işlenmesi sonucunda oluşan verilerin kayba uğramaması için, paylaşımlı veri yolunun yeterince geniş olması gerekmektedir.

Komut sıralama ve adres üretme özellikleri, veri işleme sırasında, oluşabilecek genel problemleri ortadan kaldıracaktır. Bütün ünitelerin aynı anda birbiri ile uyumlu çalışabilme özelliklerinden dolayı, karmaşık algoritmalar hızlı bir şekilde işlenebilmektedir. Mimari yapıyı oluşturan ünitelerin birbirinden bağımsız olarak çalışabilmesi, sistemin hızını katlamaktadır. Komut çevrim süresinin kısa olması, sayılan bu özelliklerden çok daha verimli

olarak yararlanabilmeyi sağlamaktadır. DSP 'lerde arzu edilen özellik, tüm komutların tek bir saat çevriminde işletilmesidir. Bunun mümkün olmamasıyla beraber, gelişmiş mikroişlemcilerdeki gibi, komutların mikrokod olarak çözümlenmesi yerine, mimari yapı, komutların gerektirdiği işlemleri doğrudan yapabilen lojik bir yapı olarak kurulmuştur. Sistem, mümkün olduğu kadar çok işlemi mümkün olan en az saat çevriminde gerçekleştirdiğinde, en verimli mimari yapıya sahip demektir.

Çizelge 4.1 : DSP ve mikroişlemcinin komut işleme sayıları

İŞLEMCİ	TAŞI	TOPLA	MAC	DALLAN
80186 Mikroişl	2	3-4	20	13
TMS320C10 DSP	1	1	2	2
ADSP-2100	1	1	1	1

Çizelge 4.1 'den görüldüğü gibi, mikroişlemcilerin temel işlemleri gerçekleştirmesi için gereken saat çevrim süresi DSP 'ye göre çok daha fazladır. DSP 'nin çarpma işlemini donanım olarak gerçekleştirebilmesi, komutların çok kısa sürede işlenmesini sağlamaktadır. DSP'lerin hızından tam olarak faydalanabilmek için, matematik işlemleri mümkün olan en kısa yoldan yapma yoluna gidilmelidir. Çarpma, toplama ve kaydırma işlemlerinin sırası işlemin sonucunu etkilemezken, hızını büyük ölçüde etkilemektedir. Bu yüzden, mümkün olan en kısa yol kullanılarak, en kısa sürede işlem, tamamlanmalıdır.

Sistem tasarımı aşamasında, DSP sisteminin seçimi sırasında belirli faktörlerin önemi bulunmaktadır. Bunlar; mimarisini oluşturan ünitelerin iç yapısı, işlem yapabilme kapasiteleri, komut setleri, ve geliştirme üniteleridir. Genel amaçlı işlemcilerle benzer yapıları bulunmasına rağmen, DSP 'lerin kendilerine özgü bir sınıflandırması bulunmaktadır. Sistem mimarisi, kontrol edilecek sisteme çok bağımlıdır. Gerekli olan yapı ancak kontrol edilecek sistemin çok iyi tanınması ile mümkündür.

Seçilecek mimari yapı, kontrol algoritmasını en az hata ile, en hızlı biçimde en yüksek performansı sağlayacak şekilde gerçekleştirmelidir. Bunun yanında, başka bir sisteme ne kadar kolay uyum sağladığı etkili bir faktördür. Kontrol sisteminin isteğine göre; gerekli olan yardımcı birimler, bellek birimleri ve giriş çıkış birimlerini bünyesinde bulundurmalı, veya bu birimler ile kolaylıkla iletişim kuracak yapıda olmalıdır.

Donanım yapısının önemi yanında, DSP sisteminin seçiminde etkili olan bir diğer faktör de kolay programlanabilme özelliğidir. Komutların basit bir yapıda olması, algoritmanın kolay gerçekleştirilmesini sağlamaktadır. Paralel işlem yapabilme özelliğine uyumlu, mümkün olduğu kadar kısa çevrim süresinde işlemini tamamlayabilen komut setine sahip DSP, tercih edilecektir.

Sistemin gelişmesi durumunda, yeni sistemlere uyumlu olarak değiştirilmesi mümkün olmalıdır. Gün geçtikçe gelişen yeni sistemler, kullanılan DSP sistemlerinin yerini alacaklardır. Fakat yeni sistemlerin hem donanım hem de yazılım olarak eski sistemlere uyumlu olmamaları durumunda, uygulama alanına girmeleri zor olacaktır. Ayrıca sistemin öğrenilmesi, geliştirilmesi ve kontrol sistemine uygulanması sırasında harcanacak sürenin, mantıklı bir düzeyde olması gerekmektedir. Geliştirme süresinin uzun olması, genel sisteme uyum sağlama bakımından önemli bir faktördür.

Sistemin kurulması sırasında göz önüne alınacak diğer bir faktör de performans/fiyat ve performans/verim oranlarının yüksek olmasıdır. Bu oranların yüksek olması sayesinde DSP 'nin genel kontrol sistemine uygulanabilirliği ve kullanım alanı artacaktır.

DSP 'nin mimarisini oluşturan ana üniteler şunlardır: Çarpım veya Çarpım-Akümülatör ünitesi (MAC), Aritmetik-lojik işlem ünitesi (ALU), Yığın kaydırma ünitesi (BS), Veri adres generatörü ünitesi (DAG), Komut sıralama Ünitesi (SEQ).

DSP ünitelerinin teker teker ele alınması sırasında belirli bir sıra izlenecektir. İlk olarak elemanın fonksiyonel tanımı yapılacak, birbirleri ile ve diğer işlemciler ile olan benzerlikleri, farklılıkları, özellikleri ele alınacaktır. Ünitelerin nerede ve nasıl kullanıldığı, DSP içinde hangi işlevi sağladığı ve önemi incelenecek, ve gerçek bir sistem üzerindeki örneği verilecektir.

4.2 DSP Mimarisini Oluşturan Üniteler

4.2.1 Çarpım Ünitesi

Diğer mikroişlemcilerden farklı olarak, DSP içinde donanım olarak bulunan çarpım ünitesi sayesinde, çarpma işlemleri hızlı bir şekilde çözülmektedir. Çarpım ünitesi, tek çevrim süresinde, girişine uygulanan sayıların paralel dizi çarpma işlemini gerçekleyen sistem olarak tanımlanabilir. Girişine uygulanan sayıların format kontrolünü yaparak, işaretli ve işaretli ve karışık moddaki sayılar üzerinde çarpma işlemini gerçekleştirmektedir. Sonuç üzerinde yuvarlatma yaparak veri kaybının minimum düzeyde olmasını sağlamaktadır. Mikroişlemcilerden farklı olarak donanımsal bir yapı olarak işlevini gerçekleştiren çarpım ünitesi, DSP 'ye tartışılmaz bir üstünlük kazandırmaktadır. Mikroişlemcilerde çarpma işleminin yazılım olarak gerçekleştirilmesi sırasında MİB (Merkezi İşlem Birimi) ile haberleşme sırasında olan zaman kaybı, adres ve verilerin aynı yol üzerinden yollanmak zorunda olmasından dolayı daha da artmaktadır. Çarpım işleminin tek saat çevriminde tamamlanması, diğer üniteler ile paralel olarak işleme ve veri transferini mümkün kılmaktadır.

Çarpım ünitesinin hızı, RAM belleğe erişim hızı ile karşılaştırmaya uygun ve paralel işlem yapabilmeye yatkın olmalıdır. Çarpım ünitesinden istenen bir diğer özellik de girişindeki sayıların format kontrolünü yapabilmesidir. Bilgi işlem sistemlerinde kullanılan sayılar, işaretli veya işaretsiz, tamsayı veya kesirli yapıda olmaktadır. Bu tür sayıların birbirine dönüşümünü en az hata ile sağlamalıdır. Sonuçta elde edilen değerin yine istenen formatta, tamsayı veya kesirli olması gerekir. Ayrıca sonucun genişletilmiş veya yuvarlatılmış olması seçeneği ile hata kontrolünü elinde bulundurmalıdır. Kullanıcı, sabit nokta veya kayan nokta aritmetiğinde istediği şekilde işlem yapabilmelidir. Çarpım işleminin tek saat çevriminde sonuçlanması, paralel çalışma ve iş hattı çalışmasına olanak sağlamak, ve kullanılan kütükler, sistemin minimum gizlilik içinde, kullanıcıya görünür bir yapıda çalışmasını kolaylaştırmaktadır.

Çarpma işleminde, işaret kontrolünün yapılması, işlemin doğruluğuna etkilemektedir. İşaretli ve işaretli sayıların çarpımında sonucun en çok anlamlı bölümü, işaretli, en az anlamlı bölümü ise, işaretli olarak çarpılmalıdır. Karışık modda çarpım denilen bu işlem, her iki giriş sayılarının bağımsız format kontrolünü gerektirmektedir.

Çarpma işlemi sonucunda elde edilen verinin sadece en çok anlamlı bölümünün saklanması gerekiyorsa, bu belirli bir hataya neden olacaktır. Bu hatanın kontrol altında tutulabilmesi, sonucun yuvarlatılması ile sağlanmaktadır. Yuvarlatma yerine kesme işlemi, hatanın

artmasına neden olmaktadır.

Bir çarpım ünitesinin verimli olması için, hız, işlem doğruluğu ve fiyatının uyum içinde olması gerekir. Çarpma işleminin hızı, işlemcinin kelime genişliğine bağlıdır. Çarpma işleminin sonucunun minimum hatalı olması için yapılan ölçeklendirme, sonucun iyileşmesini sağlamaktadır. Çarpma işleminin çok olduğu karmaşık uygulamalarda, kelime genişliği büyük DSP 'lerin hem hız hem de doğruluk olarak daha iyi sonuç vereceği açıktır. Bu da kayan nokta aritmetiği ile çalışan işlemciler ile sağlanabilir. Yalnız bu aşamada fiyat faktörü dikkate alınmalıdır. Kayan noktalı işlemcilerin hala sabit noktalıya göre daha pahalı olması, yalnız çok yüklü matematik işlem gerektiren uygulamalarda kullanılmasını verimli yapmaktadır.

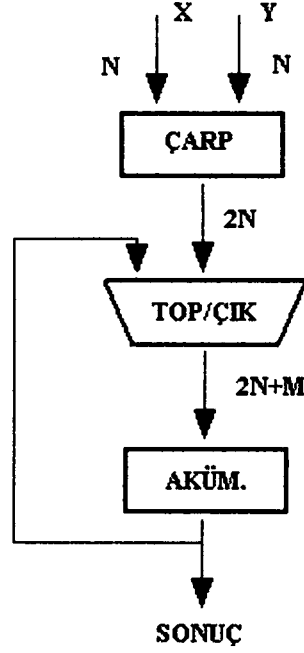
4.2.2 Çarpım - Akümülatör Ünitesi (MAC)

İlk üretilen DSP 'lerde tek başına bulunan çarpım ünitesi, daha sonraki nesillerde akümülatör ile birleştirilerek, çarpım-akümülatör ünitesi olarak adlandırılmıştır.

Dijital filtreler, FFT algoritmaları ve vektörel işlemler gibi birçok uygulamada karşılaşılan

$$\sum a(n)x(n-k) \quad [4.1]$$

işleminin kolaylıkla gerçekleştirilmesi için MAC ünitesi kullanılmaktadır. Mikroişlemcilerde matematiksel işlemlerin gerçekleştirilmesini sağlayan akümülatör ünitesi, merkezi işlem birimi içindeki diğer birimlerle haberleşerek, yazılım şeklindeki işlemleri gerçekleştirmektedir. DSP 'lerde ise, donanım yapısında olan çarpım üniteleri ile birleşerek çarpma-biriktirme işleminin tek saat çevriminde gerçekleşmesini sağlamaktadır. Böylece çok daha hızlı işlem yapabilme yeteneğine kavuşulmuştur. MAC 'in yapısında bulunan geri besleme yolları ve iş hattı işleminin gerçekleşmesini sağlayan kütükler yardımıyla, tekrarlanan yapıdaki işlemler kolayca gerçekleştirilmektedir. MAC ünitesinin en önemli özelliği çarpma-biriktirme işleminin tek saat çevriminde gerçekleştirmesidir. Biriktirme işlemi sırasında, sonucun doğruluğunu arttırmak için akümülatörün çıkış kütüklerinde, büyütme bitleri adı verilen koruma bitleri bulunmaktadır. Çarpım ünitesinin çıkışının genişliği ile uyumlu bit sayısı bulunmaktadır. İşlem sonunda taşma durumu oluşmakta ise, büyütme bitleri yardımıyla, bu durum MAC 'a bildirilerek, gerekli ölçeklendirme işleminin yapılması sağlanmalıdır. Eğer çarpım ünitesinin çıkışından elde edilen sayının sadece MSB (en çok anlamlı yarısı) kullanılacak ise, yuvarlatma ve kesme işlemlerinin en az hata ile gerçekleşmesi sağlanmalıdır. Ayrıca, MAC ünitelerinde bulunan yardımcı geri besleme yolları ve kütükler yardımıyla, DSP işlemlerinin en az saat çevriminde gerçekleştirilmesi sağlanmalıdır.



Şekil 4.2 : MAC 'ın basit yapısı

MAC ünitesine N bitlik iki adet sayı girildiğinde, çarpım ünitesinden 2N bitlik çıkan sonuç, TOP / ÇIK ünitesine gelerek, işlem yapılır ve biriktirme biriminin çıkışından sonuç elde edilir. C geri besleme yolu üzerinden tekrar TOP/ÇIK birimine gelerek, işlemin tekrarını sağlar. TOP/ÇIK biriminin çıkışı, M adet büyütme biti ile genişletilmiştir. Bu sayede, en az 2M adet çarpma biriktirme işlemi, taşma olmadan gerçekleştirilebilir. Örneğin, 40 bitlik bir akümülatör ve 32 bitlik bir çarpım ünitesi 256 komut saat çevrimine, taşma riski olmadan izin vermektedir.

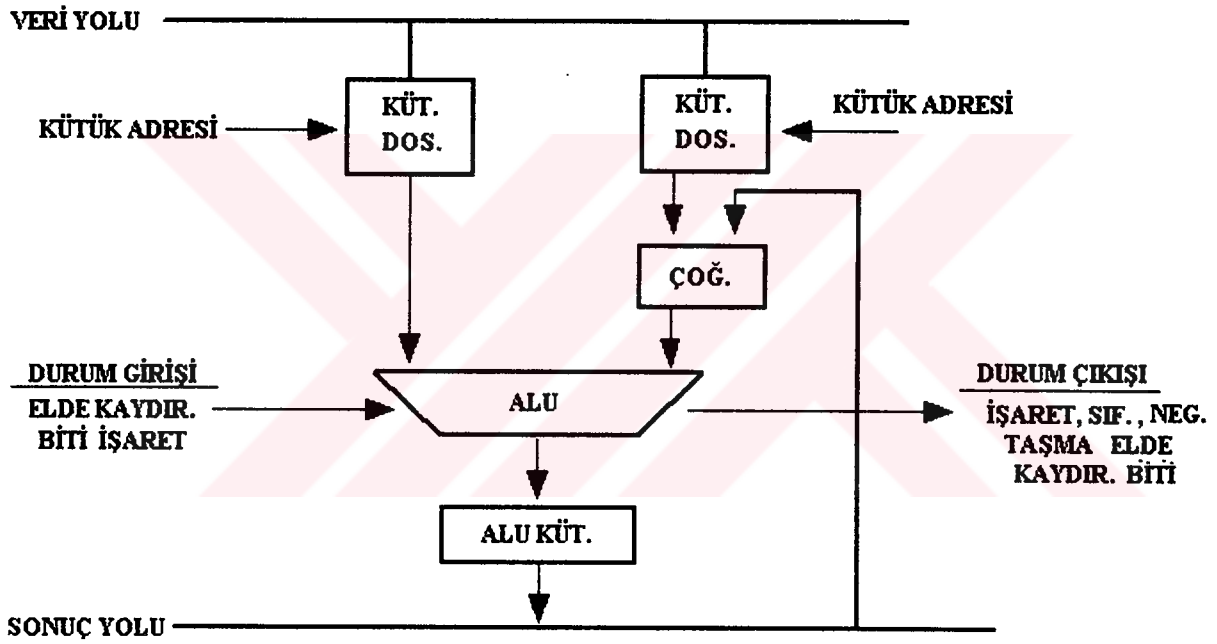
MAC ünitesinde yapılan işlemlerde, istenen bir diğer özellik de taşma ve yuvarlama kontrolü yapabilmesidir. Taşmanın daha önceden algılanabilmesi, sisteme eklenecek bir lojik birim ile sağlanabilir. Eğer bütün bitler eşit değilse, yani hepsi bir veya sıfır değilse, akümülatörün taşması, büyütme kütüğünde taşma olmadan algılanabilir. MAC ünitesine, işlemler sonucunu doğru olarak verebilmesi için kaydırma birimleri eklenmiştir. Bu sayede, hata minimuma indirilmektedir.

4.2.3 Aritmetik- Lojik İşlem Ünitesi (ALU)

DSP sistemlerinde ALU ünitesi, mikroişlemcilerde olduğu gibi, aritmetik ve lojik işlemleri gerçekleştiren birimdir. Bilgi işlemede önemli bir yeri olan lojik işlemler bu birim tarafından gerçekleştirilmektedir. DSP'lerdeki ALU ünitesinin farkı ise, tek saat çevriminde işlem yapabilmesidir. Bunun yanında, MAC ünitesinde olduğu gibi, tekrarlanan işlemlerde, iş hattı

denilen yapıya izin vererek, verilerin sıkışmasını önlemektedir. DSP içindeki diğer ünitelerle paralel çalışabilme özelliği yanında, ölçeklendirme birimleri ile desteklenerek, verinin maksimum dinamik aralığını korumasını sağlamaktadır.

ALU ünitesinden istenen en önemli özellik, bit işletme ve karakter tutma işlemlerini gerçekleştirirken, hızlı işlem yapmasıdır. DSP 'deki ALU birimi, mikroişlemci ile karşılaştırıldığında, daha kısıtlı komut seti ve adresleme modları olmasına rağmen, tek saat çevriminde işlem yapma ve paralel çalışma sayesinde, temel işlemleri, mümkün olan en kısa sürede gerçekleştirilmektedir. ALU ünitesinin gerçekleştirdiği aritmetik işlemlerden bazıları toplama, çıkarma, elde biti ile toplama, borç biti ile çıkarma, mutlak değer alma, lojik işlemlerden bazıları ise, VE, VEYA, evirme vb. dir.



Şekil 4.3 : ALU 'nin basit yapısı

ALU 'dan istenen bir diğer özellik de, işlem yapılmadan önce, genel durum hakkında, durum bitleri yardımıyla, bilgi vermesidir. Sonucun pozitif, negatif veya sıfır olması, taşma oluşup oluşmadığı, elde olup olmadığı, durum kütüğünde bulunan bayraklar yardımıyla belirtilmektedir. DSP 'deki ALU'nin bayraklarının diğer üniteler tarafından kullanılabilir olması, kayan nokta aritmetiği ile yapılan işlemlerde ölçeklendirmeyi kolaylaştırmaktadır. ALU ile bölme işlemlerinin gerçekleştirilmesi, diğer ünitelere göre daha hızlı olduğundan tercih edilebilir. Veri yollanması ve işlenmesi sırasında, ALU 'daki bir diğer özellik de verimli çalışabilmesidir. Veri yollanması ve aritmetik işlemlerin aynı çevrimde gerçekleşmesi ile aynı saat çevrimi içinde ayrı iki sayının yüklenebiliyor olması, ünitenin hızını katlamaktadır. ALU

'nin, işlem yaparken, aynı saat çevrimi içinde, girişteki bayrakların durumuna göre, sonucu kaydırabilmesi, DSP 'nin hızını arttırmaktadır.

ALU 'nin bu işlemleri gerçekleştirebilmesi, tam anlamıyla yeterli sayıda olan kütük dosyasına bağlıdır. Kütük dosyasının yeterli genişlikte olması, program tarafından devamlı kullanılan verinin saklanması ve sonuçların saklanmasında bellek kullanımını minimum düzeye indirmeye yaramaktadır. Veri yollama ve aritmetik işlem yapma işlemlerinin aynı saat çevriminde gerçekleşiyor olması, yazma ve okuma işlemlerini aynı anda gerçekleştirebilen kütükler sayesinde olmaktadır. Giriş işlemlerinde kullanılan kütüklerin, program akışı sırasında gözlemlenebilir olması, sistemin kontrolünü kolaylaştırmaktadır. Girişte kullanılan veri yolları ve giriş çıkış yollarının esnek yapıda olması, diğer ünitelerle işlem yaparken, kolaylık sağlamaktadır. Geri besleme yollarının kullanımı, ünitenin bağımsız bir işlemci olarak çalışmasına olanak sağlamaktadır. Örneğin, ALU ve Yığın Kaydırma ünitesi, paralel çalıştıklarında, veri işlemede hızlı ölçeklendirme için ideal bir yapı oluşturmaktadırlar. Her ünite de veri yan yollarının bulunması, ana yola gerek kalmadan her ünitenin birbiri ile haberleşmesini sağlamaktadır.

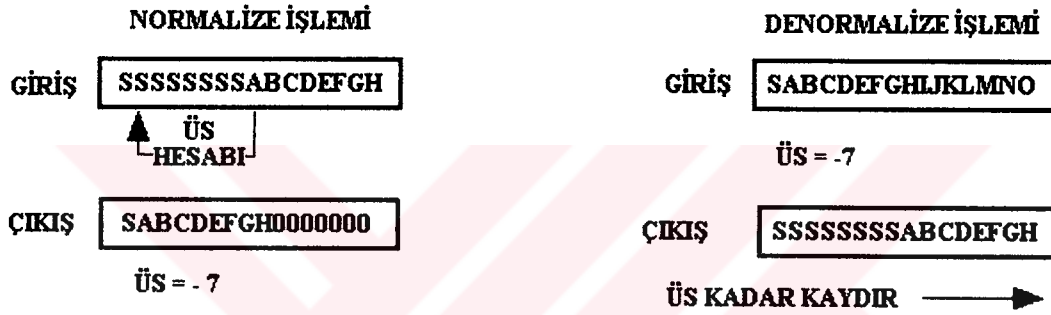
4.2.4 Yığın Kaydırma Ünitesi (BS)

Kaydırma ünitesi, işleme sırasında taşma ve veri kaybı sorunlarını önlemek amacıyla, sayıları ölçeklendirme için ve sabit nokta ile kayan nokta arasında dönüşüm yapabilmek için kullanılan bir ünite dir. Mikroişlemcilerdeki kaydırma ünitesi, her bir çevrimde bir bit kaydırma yaparken, DSP 'deki ünite, tek saat çevrimi içinde birçok bitin kaydırılmasına olanak sağlamak suretiyle, hızı arttırmada etkili bir rol oynamaktadır.

Kaydırma ünitesi, aritmetik ve lojik kaydırma işlemleri yapabilir. Aritmetik kaydırma işlemi, sağa ve sola kaydırma ile sayıların ölçekleşmesini sağlar. Sağa kaydırma işlemi, sonucun işaretini kaybetmesini önleyerek, solda açılan yeni yeri işaret bitinin değeri ile doldurur. Sağa kaydırma işleminin arka arkaya tekrarlanması, sonuçta kesme hatasına yol açar. Sola kaydırma işlemi, sayının boşalan yerlerini sıfır ile doldurur. Taşma olayının oluşmasını önlemek için lojik bir birim eklenmiştir. Mikroişlemcilerde sola kaydırma işlemi, eğer MSB (en anlamlı bit) işaret bitine eşit değilse, taşma bayrağını işlemektedir. DSP 'deki ünite ise, MSB değerini saklayarak, taşma durumunun daha önceden anlaşılıp, önlenmesinde kullanılır. Lojik kaydırma işlemi, veriyi işaretsiz olarak kabul eder, ve dönüş yönüne göre, sayıyı sıfır ile doldurur. Yığın kaydırma ünitesi, mimari yapısından dolayı, uzun kelimelerin iki çevrim içinde döndürme işlemini tamamlayabilir. DSP 'deki özel komutlar yardımıyla, hızlı kaydırma

işlemi, komutta belirtilen bir kontrol değeri kadar taşıma işlemi gerçekleştirilmektedir.

Kayıdırma ünitesinin gerçekleştirdiği en önemli işlemler, normalize, denormalize, kayan nokta bloklama olarak sıralanabilir. Normalize işlemi, sabit noktadaki bir sayının, taban ve üs değerlerini belirleyip, kayan noktaya çevirmeye yaramaktadır. Normalize işlemi iki aşamalıdır; ilk olarak üs alma detektörü, gerekli kaydırma değerini hesaplar, ve bu değer kaydırma işlemi için, kaydırma kontrol bitine yüklenir. Kaydırılacak olan yerin sayısı, üs detektörü tarafından ikilik düzendeki bir kelimeye yüklenir ve üs kütüğü denilen, sayının üs değerini saklayan bir kütüğe yazılır. Üslü sayıya dönüştürülecek sayının bitlerini dönüştüren donanım yapısı, öncelik algılayıcısı denilen bir yapıdır. Lojik bir işlem gerçekleyen bu yapı, 2^b şeklindeki girişe karşılık, b çıkışını üretmektedir.



Şekil 4.4 : Normalize ve denormalize işlemleri

Denormalize işlemi, kayan nokta bir sayıyı sabit noktaya çevirmektedir. Veri yolundan üs kütüğüne yüklenen bir sayı, kaydırma işleminin kontrol kodunu belirlemektedir. Kaydırılan sayıdaki tabanın doğruluk değerinden daha büyük bitler kaybolacaktır. Denormalize işlemini kontrol eden birim, lojik kaydırmaya, kelime dönüşümlerine ve kelime yürütmeye izin veren bir yapıdadır.

Blok kayan nokta işlemi ise, bir veri dizisi içinde en büyük elemanın doğal logaritmik üssünü almaktadır. Bu üs daha sonra bütün dizi ile birleşerek, sayının ölçeğini kontrol etmektedir. Blok kayan nokta üsleri üs algılama lojigi ile algılanmaktadır. Blok kayan nokta işlemi, bir veri bloğunun maksimum dinamik aralıkta olmasını, taban kısmının doğruluğunu tehlikeye atmadan sağlamaktadır. Bunu gerçekleştirirken, gerçek kayan nokta işlemlerinin gerektirdiği gibi veri kelimelerinden üssü belirtmek için bitlerin alınmasına gerek kalmamaktadır. Gerçek kayan nokta, verinin blok olarak işleyişine izin verilmediğinde ve 32 veya 64 bit gibi daha yüksek doğrulukta veri işleme gerektiğinde kullanılmaktadır.

4.2.5 Veri Adres Generatörü Ünitesi (DAG)

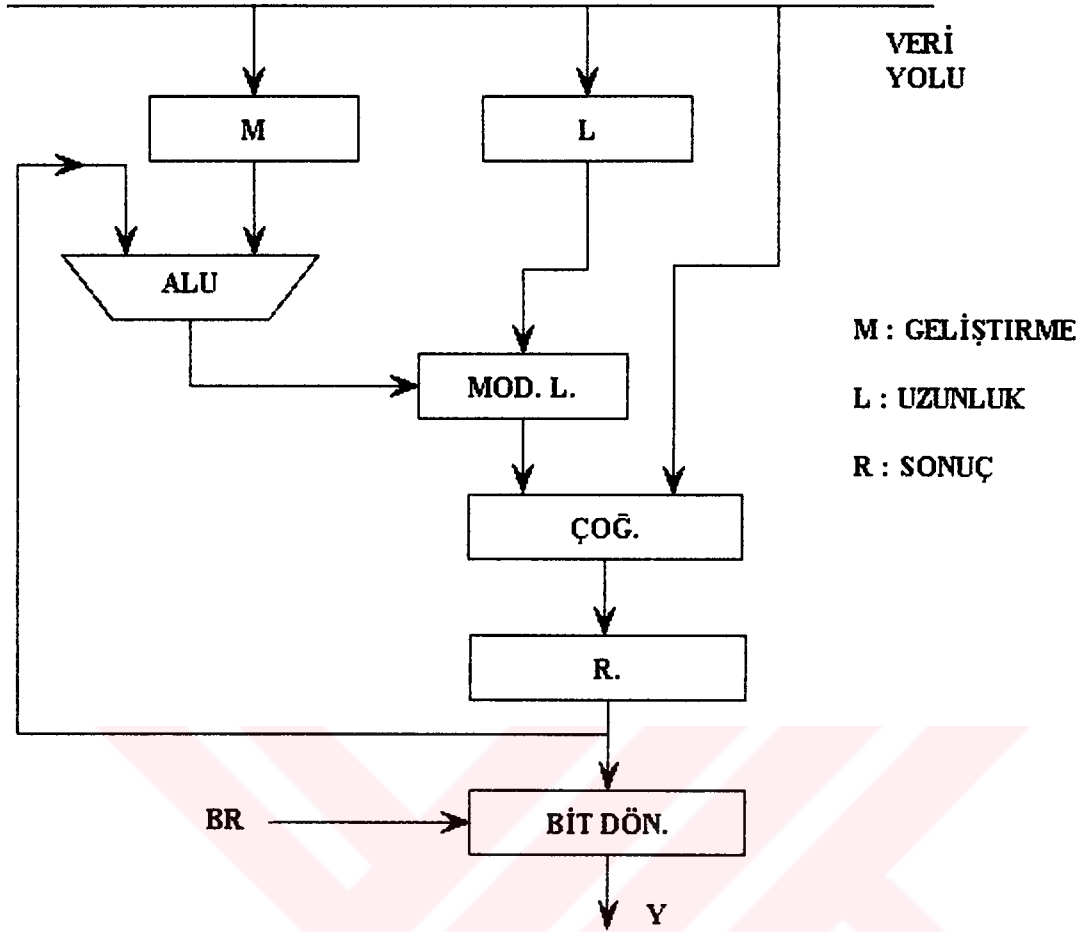
Veri adres generatör ünitesi, bellek üzerindeki adreslere belirli sayı değerleri atayan kütükler içeren bir yapı görünümündedir. Bu sayede, veri okunup yazılması, çok daha hızlı olmaktadır. DSP 'lerde adres generatörünün ayrı bir yapı olması, algoritmanın işleme sırasında, aynı anda birkaç verinin işlenebilir olması ve ünitelerin birbirlerinden bağımsız paralel bir yapıda çalışabilmesini sağlamaktadır.

Adres generatörü yapısı, kendi içinde bir işlemci gibi işlemektedir. Mikroişlemcilerde MİB içinde bulunan bu yapı, karmaşık bir algoritma ve dallanmış çevrimler ile karşılaştığında, hızını yitirmekte, bu da sistemin performansını düşürmektedir. Fakat, DSP 'lerde adres generatörü gibi, işlemleri gerçekleştiren ünitelerden bağımsız bir yapı olması, sistemin hem hızını katlamakta, hem de karmaşık algoritmalar ile işlemler daha kolay ele alınabilmektedir.

Veri adres generatörünün en önemli özellikleri, veri belleğine adres gönderebilmesi, hesaplanmış bir başlangıç değerinden adres üretebilmesi, lojik işlemler ve kaydırma işlemleri yapabilmesidir. Bu ünite, kendine özgü komut kümesi yardımıyla, çıkış kütüklerini en verimli şekilde kullanarak, algoritmanın işleyiş hızını arttırmaktadır.

Adres generatörünün bağımsız bir işlemci gibi işleyebilmesi, bir çok özellik sağlamaktadır. Ünitenin, kaydırma, bit dönüştürme, lojik fonksiyon ve maskeleyme işlemlerini gerçekleyebilmesi, FFT (Hızlı Fourier Dönüşümü) gibi karmaşık adresleme gerektiren uygulamalarda birçok kolaylık sağlamaktadır.

DAG ünitesinden beklenen veri adres ve katsayı adres üretimi için Şekil 4.5 'de gösterilen geliştirme, uzunluk ve sonuç alt birimlerinin sayısı, değişik uygulamalara uyumlu olması için birden fazladır. DAG ünitesinin kaydırma kabiliyeti, FFT gibi değişik adresleme özellikleri gerektiren uygulamalarda, look-up table adı verilen veri tablolarına ulaşmak için ideal bir özelliktir. Bit dönüşüm birimi, üretilen adreslerin, gerektiğinde bitlerinin yerini ters döndürerek değiştiren lojik bir yapıdır. Veri tabloları oluştururken, DAG ünitesinin lojik işlem ve maskeleyme yapabilmesi, sisteme çok büyük bir avantaj sağlamaktadır. DAG kütüklerine döngü değerlerinin yüklenmesi ile, program akışı bozulmadan çevrimlerin işleme mümkündür.



Şekil 4.5 : DAG ' ın basit yapısı

4.2.6 Komut Sıralama Ünitesi (SEQ)

Komut sıralama ünitesi, program akışını belirleyen komut adreslerini üretir. Bu ünite, Program Sayacının (PS) artırılması, alt program çağırılması ve dış kesmelerin işletilmesi görevlerini üstlenmiştir. Örneğin, program bir döngünün tamamlanıp tamamlanmadığını kontrol etmek için, sayaçlardan yararlanabilir. Bu tür fonksiyonlar, SEQ 'deki kütük yığınları yardımıyla işletilmektedir. Alt program ve kesme programından ana programa döndüğünde, hangi adrese dallanılacağı PS içinde saklanmaktadır. SEQ aynı zamanda, karşılaştırma komutları tarafından üretilen işaret, taşma, yığın boş gibi durum bilgilerini işleyebilir.

SEQ 'nin ana fonksiyonları; her komuttan sonra PS 'yi bir artırarak, program akışını sağlamak, alt program adreslemesini düzenlemek ve geri döndüğünde adreslerin işlemlerini yönetmek, DSP donanımındaki döngü sayaçlarının kontrolü altındaki çevrimleri düzenlemek, veri taşmasının düzeltilmesi veya verinin yeniden ölçeklendirilmesi gerektiğinde gerekli olan alt program parçalarını çağırarak, dış G/Ç elemanları ile haberleşebilmek için gereken kesme

servisini sağlamak, kesme servis programlarına dallanmak ve program tamamlandığında geri dönüş adresini belirlemek, olarak sıralanabilir.

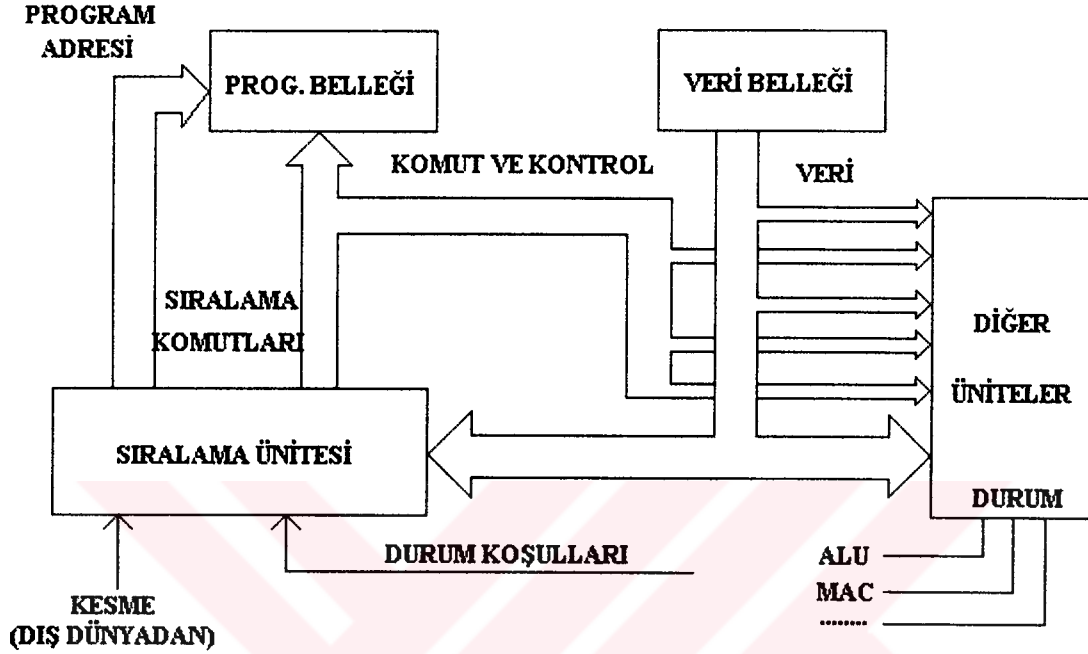
SEQ 'ni oluşturan ana birimler; Program Sayacı (PS), lojik durum algılayıcı, yığın (alt program ve kesme adresleri, sayaç değerleri ve atlama adresleri için), döngü sayaçlarıdır. SEQ 'nin tasarımındaki amaç, program akış kontrolü yapabilmesi yanında, yeterli yığın boyu, gerek yazılım döngülerinde, gerekse donanım tabanlı kesme işlemlerinde programın dallanmasını sağlamaktır. Ünitenin özellikleri, veri yolundan doğrudan doğruya atlama adresinin belirlenmesi, yığın sınırlarının kesin olarak belirtilmesi ve durumsal atlama sağlanması olarak sıralanabilir. Ünitenin yeterli yığın boyuna sahip olması, alt programları ve değişik düzeyde kesmeleri işlenebilmesini sağlamaktadır. Program akışı sırasında, bir dallanma komutu veya kesme önceliğinin belirlenmesi gibi bir durum ile karşılaşıldığında, akışın hızla sağlanması için, bayraklar yardımıyla durum değerlendirilmesi yapılmaktadır. Yüksek düzeyli bir kesme isteği, bayrak testi gibi daha düşük düzeyli işlemler karşısında öncelikli olmalıdır. Kesme maskeleye gibi özelliklere olanak sağlanmaktadır.

Sıralama işlemi sırasında, iş hattı biçiminde işlem yapma tavsiye edilmemektedir. İşlemin parçalara bölünüp, sıralı olarak işletilmesi olarak adlandırılan iş hattı işlemi, işlem hızını arttırmak ve veri kaybını önlemek için kullanılmaktadır. SEQ 'de bu özelliğin olmaması, iş hattı işlemi sırasında olan gecikmeler ve ünitenin veri tablolarından veri işlemesi sırasında sorun çıkmasını önlemektedir. Dış bir kesme isteği olduğunda veya taşma algılandığında, gerekli olan işlemler birkaç saat çevrimi içinde gerçekleşemezse, sistemde karışıklık olacak ve hızı belirli oranda düşecektir.

Sistemin normal işlemesi sırasında, adresler SEQ 'de saklanırken, doğrudan adresleme özelliği ile komutların işlenmesi sağlanmaktadır. Dış porttan adres değerinin doğrudan okunabilmesi, sistemin işlemesi sırasında kolaylık olmaktadır. Yığın, programın işleyişi sırasında belirli değerlerin saklanması için kullanılmaktadır. Alt program, kesme ve atlama geri dönüş adresleri aynı yığın içinde saklanmaktadır. Yığın boyu sınırlı olduğundan, uygulamanın özelliğine göre, bütün durumlar göz önüne alınarak, yer ihtiyacına cevap verecek bir yapı sağlanmıştır. Yığın taşması, yığınların birbiri ile çakışması, en yüksek dereceden maskelenemez kesme tarafından önlenmektedir. Döngü kontrolü için gereken karmaşık kod yapısını azaltıp, iki veya üç yöllü atlama işlemlerini tek bir komut ile işleyerek, sistemin kolay programlanmasını ve çalışma hızını arttırmaktadır.

SEQ ünitesi, program akışı sırasında bir sonraki adresin seçimini, komut kodları, durum

lojiji, kesme ve döngü işleyişinden gelen bilgi ile gerçekleştirmektedir. Kesme vektöründe saklı olan adres, normal işleyiş sırasında PS ile arttırılmış değerdeki adres, alt program veya kesme işleminden dönüşte PS yığımında saklı olan adres ve komut kodu tarafından üretilen adres sistemin işleyişini belirlemektedir. Bir diğer alternatif de ; indeks atlama olarak adlandırılan, adresin dışarıdan program adres yoluna konup PS 'ye aktarılmasıdır.



Şekil 4.6 : Sıralama ünitesi ile diğer ünitelerin ilişkisi

Programın işleyişi sırasında, normal çalışma, PS ve PS yığımından yararlanılmaktadır. İşlemekte olan komutun adresi, PS 'de saklanır ve çıkış bir arttırılır. Bu değer, bir sonraki komutun adresini veya, alt program ya da kesme program vektörü dönüş adresini içermektedir. İndeks adres işlemlerinde, DAG tarafından üretilen yeni bir PS değeri, PS 'ye yüklenmek amacıyla, sıralama ünitesinin yoluna gönderilir.

Sıralama ünitesi, durum kütüğünde yer alan bilgiye göre konum değiştirebilir. Durum kütüğünde, ALU veya MAC 'tan gelen aritmetik durum bilgisi, çalışma durum bilgisi, yığın bilgisi, kesme kontrol bilgisi ve kesme maskeleyme bilgisi yer almaktadır. Bu bilgiler ışığında, ünite sistemin işlemesi için, diğer ünitelere gereken bilgiyi yollamaktadır. Döngü ve atlama komutlarında, veri kaybını önlemek için yığın kullanılmakta, döngüde işletilecek komutlar döngü yığımında, o andaki PS değeri de PS yığımına atılmaktadır. Döngü veya çevrimin sona ermesi için gerekli koşul sağlandığında, program tarafından kullanılan değerler, SEQ tarafından tekrar yüklenerek, işleme devam edilmektedir. Kesme isteği ile

karşılařıldığında, öncelik ve maskelenme durumuna göre, gerekli programı çalıştırmakta, veri kaybını önlemek için, durum bilgisini durum kütüğünde saklamaktadır



5. ÖRNEK DSP YONGALARIN KARŞILAŞTIRILMASI

Daha önceki bölümlerde ele alınan DSP sistemlerinin genel mimari yapısı, bu bölümde gerçek sistemler üzerinde tartışılacak, böylece kontrol sistemi tasarlanırken, kullanılacak yapının daha bilinçli seçilmesi sağlanacaktır.

Kontrol edilecek sisteme uygun bir denetleyici seçimi, bir tasarımcının bütün sistemin kurulması aşamasında göz önüne alması gereken en önemli bölümdür. Kontrol edilen sistemin karakteristiğine göre, DSP sisteminin yapısal özellikleri seçilmelidir. İsteğe cevap vermeyen bir kontrol yapısı, yanlış kontrole neden olacak, sistemin davranışını bozacak ve maliyetin gereksiz olarak artmasına neden olacaktır. Örneğin sabit nokta ile çalışabilecek yapıdaki bir sistemde, kayan nokta işlemci kullanılması, sistemin maliyetini katlayacaktır. Kontrol edilen sistemin dinamik sayı aralığının büyük olması veya sayısal hataların getireceği sorunlara karşı aşırı duyarlı olması durumunda, kayan nokta işlemcinin kullanılması, sistemin daha yüksek bir performansta çalışmasını sağlayacak, maliyet unsurunu dengeleyecektir.

DSP sistemleri, mikroişlemci ve mikrodenetleyici olmak üzere iki ana yapıda toplanabilir. Mikroişlemci yapısı, sadece DSP yapısını içermekte, yapının içinde herhangi bir bellek birimi veya seri port, A/D birimi gibi giriş çıkış birimi bulunmamaktadır. Yapısında herhangi bir yan birim bulunmaması, hızlı işlem yapma açısından büyük bir yarar sağlamaktadır. Bellek birimine veri yazılıp okunması, sistemin hızını düşürmektedir. DSP ne kadar hızlı veri işlese işlesin, veriyi aktarırken erişebildiği hız, ancak bellek biriminin hızı kadardır. Bu yüzden kullanılan yardımcı birimler genel kontrol yapısına uygun seçilmelidir. Bu yardımcı birimlere erişim süresi, sistemin hızını olumsuz olarak etkilemektedir. Bu yüzden mikroişlemci yapıdaki DSP 'lerin, kütük dosyaları ve komut setleri yeterince büyük yapılarak, bellek birimlerinin verimli kullanılması yoluna gidilmektedir. Böylece sistemin hızını azaltacak yapısal değişiklikler mümkün olduğu kadar aza indirgenecektir.

Mikrodenetleyici yapıdaki DSP sistemleri ise, bellek ve yardımcı birimleri bünyesinde bulundurmaktadır. Bu sayede belirli boyuttaki programlar dışarıdan bir bellek birimine gerek duyulmadan saklanabilmektedir. Yapı içine konulan bellekler, çok kısıtlı boyutta olmasına rağmen, basit uygulamalar için ideal bir çözüm sunmaktadır. Daha geniş bellek ihtiyacı, dış bellek birimleri ile karşılanabilir. Kontrol edilen sistemin parametrelerinin gözlemlenebilmesi ve haberleşme amacıyla yardımcı birimlere gerek duyulmaktadır. Yapısında seri port bulunduran DSP sistemleri, kontrole daha uygun bir alternatif oluşturmaktadırlar. Analog bir büyüklük, dijital bir veriye dönüştürülerek, sistemin üzerinde işlem yapabileceği bir hale

getirilmelidir. A/D çevirici, özellikle motor kontrolunda gerek duyulan bir yapıdır. Yapısında A/D çevirici bulunduran sistemlerde, örneğin kontrol edilen motorun akımı belirli bir dönüşüm ile algılanarak, kontrol altında tutulabilir. Bu gibi alt birimleri yapısında bulunduran bir DSP , dış birimlere minimum gereksinme duyacak şekilde çalışmaktadır. Mikrodenetleyici yapısındaki DSP, mikroişlemci türüne göre daha yavaş işlem yapsa da, mikroişlemcinin dış birimlerle haberleşirken harcadığı zaman bu açığı kapatacaktır. Yalnız içinde bellek birimi bulundurmayan yapı, ek mimari yapılar için gerekli alana sahip olacağından, daha kuvvetli bir işlem kapasitesi sağlanabilir.

DSP sistemlerinin kontrol edilen sistemin gereksinimini karşılayabilmesi için mimari yapısının belirli özellikleri taşıması gerekmektedir. Kontrol edilen sistemin ihtiyacının karşılanması sistemin tanınması ile başlamaktadır. DSP 'den yeterli verimin alınması, mimarisinde belirli yapıların bulunması ile sağlanabilir. Sistemin maksimum verim ile çalışabilmesi için aritmetik işlemleri yürüten ALU, MAC ve Kaydırma Ünitesi, veri akışının kontrolünü yapan DAG ve SEQ üniteleri bulunmaktadır. Bu ünitelerden birinin eksik olması veya yeterli performansta çalışmaması sistemin performansını büyük oranda düşürecektir. Örneğin; kaydırma ünitesinin etkili çalışmaması kayan nokta işlemlerinin düşük verim ile çözümlenmesi anlamına gelmektedir. MAC ünitesinin performansında bir azalma, çarpma-ekleme işleminin ana yapıyı oluşturduğu algoritmaların işleyiş hızını büyük ölçüde düşürecektir. Veri aktarımı ve akış kontrolünü sağlayan üniteler, sistemin hızını doğrudan etkilemektedirler. Bu ünitelerin yapısındaki bazı eksikler, hızın birkaç kat azalmasına neden olmaktadır. Bu ünitelerle beraber sistemin çalışmasına etki eden bir diğer etken de birimleri bağlayan yollardır. Yolların yeterince geniş olmaması, hızı azaltacağı gibi, aritmetik işlemler sonunda verinin tam olarak iletilmemesi hatayı arttıracaktır.

Genel kontrol uygulamalarında mikroişlemci yerine DSP 'lerin kullanılması uygulamanın ihtiyacının değişmesi ile açıklık kazanmıştır. DSP 'lerin mikroişlemcilere göre tercih edildiği uygulamalar, matematik işlem yönünden çok kapsamlı olmaktadır. Uygulanan algoritmaların işlem karmaşıklığı, mikroişlemcilerin yeterli çözüm olanaklarını sunmasını engellemektedir. Kontrol algoritmalarında en fazla karşılaşılan algoritma, ağırlık oranlarının toplamı olarak adlandırılan işlemdir. Dijital filtrelerin temel yapısını oluşturan bu işlem, giriş işaretini belirli bir ağırlık katsayısı ile çarparak ve bir önceki işaretleri kullanarak bir sonraki çıkış işaretini üretmektedir. Temel yapıyı kuran bu işlem, çok sayıda çarpma ve toplama işlemi gerektirmektedir. Dijital işaret işleminin temel işleminin algoritma içinde bir kaç defa kullanıldığı göz önüne alınırsa, işlemcinin komut çevrim süresinin kısa olmasının yararları

ortaya çıkacaktır.

Dijital işaret işleme sistemlerinin bir diğer özelliği de gerçek zamanda işlem ihtiyacı göstermeleridir. Kullanıcıya belirli bir gecikme ile ulaşan bir işaret, sistemin hatalı çalışmasına neden olacaktır. Örneğin, ses işleme sisteminde, konuşulan kelime ile tanınan kelime arasında bir gecikme olması, kullanıcının yanlışlık yapmasına neden olacaktır. Motor kontrol sisteminde, yük üzerindeki ani bir değişikliğin kontrol sistemi tarafından zamanında algılanamaması, sistemin yanlış cevap vermesine neden olacak ve motorun tahrip olmasına kadar gidebilecek sorunlar doğuracaktır. Bu yüzden, DSP sistemleri, gerçek zamanda veri işleyecek yapıda olmalıdır. (Çiprut,1994)

Dijital işaret işleminin temel problemi, işaretin örneklenmesidir. İşaretin doğru olarak algılanması, örnekleme frekansının yeterince büyük olmasına bağlıdır. Kontrol uygulamalarında kullanılması gereken işlemci, sistemin band genişliğinin en az on katı bir hızla örnekleme yapabilmelidir. Ancak bu şekilde, işaretin anlamını kaybetmemesi sağlanabilir. DSP sistemleri, işareti yeterince yüksek hızda örnekleyebilmeli ve elde edilen verileri saklayabilmelidir. Uygulamanın yapısına göre örnekleme frekansı da artmaktadır. Görüntü ve ses işleme sistemlerinde çok yüksek örnekleme frekanslarına ihtiyaç duyulurken, kontrol uygulamalarında daha düşük frekanslar yeterli olmaktadır. Kontrol uygulamalarında 1 kHz örnekleme frekansı olmasına karşılık, telekomünikasyon uygulamalarında 8 kHz, ses işlemede 8-10 kHz, müzik işleme uygulamalarında 40-48 kHz, video görüntü işlemede ise 14 MHz örnekleme frekanslarına gerek duyulmaktadır. (Küçükdemiral,1999)

Kontrol işaretlerini gerektiği gibi işleyen bir DSP 'nin geliştirmeye uygun bir yapısı olmalıdır. Esnek yapısı sayesinde yeni geliştirilen algoritmalara ve sistemlere uyumlu çalışabilmelidir. DSP sistemleri hala geliştime aşamasında olduklarından uygulamalara özel değişik yazılımlar geliştirilmekte ve uygulama sahasına girmektedir. Bu nedenle, DSP sistemleri, yeni algoritmaları işleyebilecek bir yapıda olmalıdır.

Çizelge 5.1 : Değişik DSP sistemlerinin genel özellikleri

	SAAT FREKANSI	ÇEVİRİM SÜRESİ	İŞLEM KAPASİTESİ	DAHİLİ		BİT UZ.
				RAM	ROM	
TMS320C10	20 MHz	200ns	5 MIPS	144	1,5K	16 Sabit
TMS320C25	40MHz	100ns	10MIPS	544	4K	16 Sabit
TMS320C30	33MHz	60ns	16,6MIPS	2K	4K	32Kayan
TMS320C50	40MHz	50ns	20MIPS	10K	2K	16Sabit
TMS320C240	20MHz	50ns	20MIPS	544	16K	16Sabit
TMS320C6201	200MHz	5ns	1600MFLOPS	512K	512K	32 Sabit
TMS320C6701	167MHz	6ns	1GFLOPS	512K	512K	32 Kayan

Teknolojinin ilerlemesi ile, işaret işleme amacıyla, mikroişlemcilerin zayıf kaldığı alanları kapatan sistemlerin arayışları başlamıştır. Bu amaca yönelik değişik işlemci yapıları geliştirilmiştir. Dizi işlemci olarak adlandırılan yapı, birçok işlemcinin paralel olarak çalışması ile kurulmuş bir yapıdır. Fakat çok pahalı bir çözüm sunduklarından, genel sistemlere uygulanmaktan çok özel sistemlerde kullanılmışlardır. VLSI teknolojisindeki gelişmelere paralel olarak, tek bir entegre yapısında sistemlerin gerçekleştirilmesi arayışları başlamıştır. DSP sistemlerinin geliştirilmesi ile işaret işleme çok daha kolay ve ucuz bir çözüm olanağına kavuşmuştur.

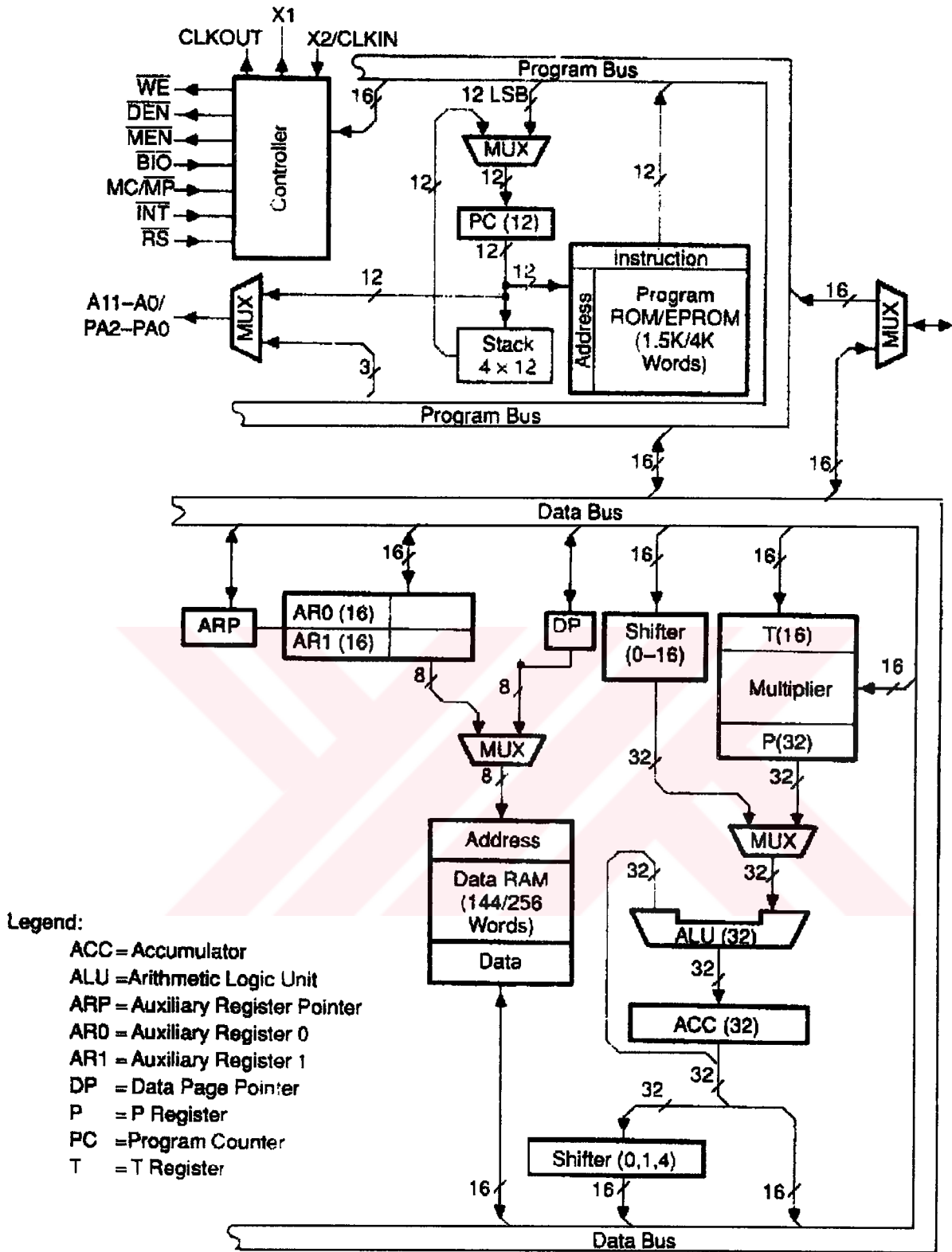
İlk olarak Texas Instruments firması tarafından üretilen işlemciler, DSP sistemlerinin ilk örnekleri olmuşlardır. TMS320 ailesi olarak adlandırılan DSP işlemcisi, ilk nesil olarak TMS32010 adı ile, 1983 yılında ticari uygulamalara yönelik satılmaya başlanmıştır. Daha sonraki yıllarda yeni nesil DSP'ler, sabit nokta ve kayan nokta işlemciler olarak üretilmişlerdir. TMS32010 DSP işlemcisi, türünün ilk temsilcisi olarak daha önce açıklanan temel unsurlara yapısında bulundurarak, mikroişlemcilerin yetersiz kaldığı birçok alanda uygulama bulmuştur. 2,4µm NMOS teknolojisi ile üretilmiştir. Donanım olarak çarpım ünitesi bulundurması işlem hızının katlanmasına neden olmuştur.

Komut saat çevrimi 200 ns olan TMS32010, 16 bitlik iki sayının çarpımı sonucu 32 bitlik sayıyı tek bir çevrimde üretebilmektedir. ALU 'ne veri aktarılmadan önce,yığın kaydırıcı ünitesi

sayesinde, veriler ötelenebilmektedir. Bütün üniteler ile haberleşmek için bulunan kütükler yanında, belleğe veri tablolarından tek çevrim süresinde veri aktarımını sağlayan yardımcı kütükler eklenmiştir. Bellek kapasitesi olarak iç bellek hacmi olarak, 1.5K kelime (3K byte) genişliğinde program ROM belleği, 144 kelime genişliğinde veri RAM belleği bulunmaktadır. Dış bellek olarak 4K kelime bellek alanını adresleyebilmektedir. İşlemcide mikroişlemci veya mikrobilgisayar çalışma modu seçilebilir. Mikrobilgisayar modunda, 1.5K kelimelik ROM belleği yanında, 2,5K kelimelik dış bellek ile tam hızında işleyebilmektedir. Mikroişlemci modunda, 4K dış bellek birimi ile tam hızda çalışabilmektedir. Maksimum saat frekansı 20.5 MHZ 'dir ve bir saniyede beş milyon komut işleme oranı bulunmaktadır.

Daha sonra yine birinci nesil olarak adlandırılan TMS320C1x DSP 'leri üretilmiştir. Değişik saat frekanslarında ve bellek kapasitelerinde üretilen bu nesil, CMOS teknolojisi kullanılarak üretildiğinden, C kodu ile adlandırılmaya başlanmıştır. Bu nesil işlemcilerin temel özellikleri şu şekilde sıralanabilir: değişik frekanslar için 160 ns, 200 ns, 280 ns komut saat çevrimi süresi, veri RAM belleği 144 kelime veya 256 kelime genişliği, program ROM belleği 1.5K veya 4K kelime, (4K EPROM bulunan türleri de bulunmaktadır.) 4K dış bellek tam hızda adresleme yapabilme, donanım olarak 16*16 paralel çarpma ve 32 bit sonuç elde etme, ALU ya giren verileri ötelemek için yığın kaydırma ünitesi, 4*12 bitlik yığın, endirekt adresleme için iki yardımcı kütük, seri port.

Birinci nesil TMS320 ailesi, 5 MIPS hızı ile program ve veri belleğinin ayrı olması ile Harvard mimarisinin temel ilk örneğidir. Mimari yapıyı oluşturan ünitelerin birbirinden bağımsız olması sistemin hızını katlamaktadır. ALU ünitesi, 32 bit kelimeler ile toplama, çıkarma ve lojik işlemleri gerçekleştirmektedir. Akümülatör ünitesi 32 bit kelime genişliğindedir ve ALU ile veri giriş çıkış bağlantısı bulunmaktadır. İşlem yaptığı veriyi, 16 bitlik yüksek ve düşük anlamlı iki bölüme ayırmaktadır. DSP 'nin hızının katlanmasını sağlayan çarpım ünitesi, 16*16 bitlik işlemi gerçekleyebilir. Bu üniteye üç kütük bulunmaktadır. T, P ve çarpma dizi kütükleri. 16 bitlik T kütüğü çarpanı saklarken, 32 bitlik P kütüğü işlemin sonucunu saklamaktadır. Çarpma işlemi için, çarpan sayısı RAM bellekten T kütüğüne yüklenir ve çarpma komutu ile işlem gerçekleştirilir. Çarpma işlemi iki saat çevrimi süresinde tamamlanmaktadır. Veriyi işlemek için iki adet kaydırma ünitesi bulunmaktadır. Bunlar, yığın kaydırma ve paralel kaydırma üniteleridir. Yığın kaydırma ünitesi, 0 ile 16 bit sola kaydırma işlemini gerçekleştirir. Paralel kaydırma ünitesi ise, ikinin tümleyeni aritmetik işleminin sonucunun işaretine göre, 0, 1 ve 4 bitlik kaydırma işlemini gerçekleştirmektedir.



Şekil 5.1 : TMS32010 fonksiyon blok diyagramı

TMS320 ailesinin ikinci nesil DSP 'leri ilk nesil ile aynı temel yapı üzerine kurulmuşsa da,

çok daha geniş işlem olanakları ile donatılmıştır. İkinci nesil DSP ailesinin temel elemanı TMS320C25 'dir. Tasarımcıya değişik uygulamalarda kullanım esnekliği getirmesi amacıyla, değişik frekans ve bellek kapasitelerinde üretilmiştir. Bu DSP ailesinin temel özellikleri şunlardır: 100 ns komut çevrim süresi, 32 bit ALU ve akümülatör ünitesi, tek çevrimde çarp-yükle işlemi, 4K 16 bit kelime ROM bellek, 544 kelime veri RAM bellek, 128K kelime program ve veri bellek alanı, sekiz yardımcı kütük, sekiz birimli donanım yığını, seri port, zamanlayıcı birimi, 16 bit giriş, 16 bit çıkış kanalı, 68 pinli yapısıdır. (Texas Instruments, 1993)

TMS320C25 sistemi, 100 ns çevrim süresi ve 10 MIPS işlem hızı ile DSP sisteminin bütün yararlarından tam olarak faydalanmayı sağlamaktadır. Veri sıkışmasını önlemek için iş hattı denilen parçalı çalışma sayesinde aynı anda komutun alınması, çözümlenmesi ve işletilmesi gerçekleştirilebilir. Komutların büyük bir çoğunluğu bir komut çevrim süresinde işletilebilmektedir. Program ve bellek alanı arasında veri transferini sağlayan özel komut seti bulunmaktadır. Program ve veri bellekleri aynı yol üzerinde çoğullama işleminden geçirilerek, adres aralığının artırılması sağlanmaktadır. Program ve veri yollarının ayrı olması ile tam hızda çalışma sağlanmaktadır. (Texas Instrument, 1993)

TMS320C25 'in bellek alanı, 4K kelime ROM ve 544 kelime RAM 'dan oluşmaktadır. RAM bellek üç ayrı bölüme ayrılmıştır. 256 kelime bölümü veri veya program belleği olarak seçilebilirken, 288 kelime bölümü sadece veri belleğidir. 544 kelime genişliğindeki veri belleği, 512 kelime dizilerin saklanması için izin verirken, 32 kelime alanı diğer ihtiyaçlar için saklamaktadır. Bunun yanında, diğer birimlerin kontrolünü sağlayan kütük yapıları da bulunmaktadır. Sekiz adet yardımcı kütük ile indirekt adresleme yapılabilmektedir. Bu kütükler, bir komut yardımıyla doğrudan veya üç bitlik yardımcı kütük göstergesi ile indirekt olarak adreslenebilir. Yardımcı kütük dosyası, yardımcı kütük aritmetik birimine bağlanmaktadır. Bu aritmetik birim, adres üretimini desteklemek görevi yanında bağımsız bir aritmetik birim olarak da işlev görebilir. Doğrudan veri belleğine bağlı olduğundan 16 bitlik işaretsiz işlem yapabilmektedir. İkinci nesil DSP 'nin temel üniteleri; 16 bit kaydırma ünitesi, 16*16 bit çarpma ünitesi, 32 bit ALU ve 32 bit akümülatörden oluşmaktadır. Kaydırma ünitesi, 16 bitlik giriş ile veri belleğine, 32 bitlik çıkış yolu ile ALU 'ne bağlıdır. Girişindeki verinin 0 ile 16 bitlik sola kaydırma işlemi komutuna bağlı olarak gerçekleştirilmektedir. En düşük anlamlı bitler, sıfır ile doldurulurken, en yüksek anlamlı bitler durum kütüğünün işaret durumuna bağlı olarak, sıfır veya bir ile doldurulabilir. Akümülatör ve çarpma ünitesinin çıkışında bulunan kaydırma birimleri, sayısal ölçekleme, taşma önleme, bit çıkarma ve geliştirilmiş doğrulukta aritmetik işlemlerde kullanılmaktadır. İş hattı çalışması sırasında,

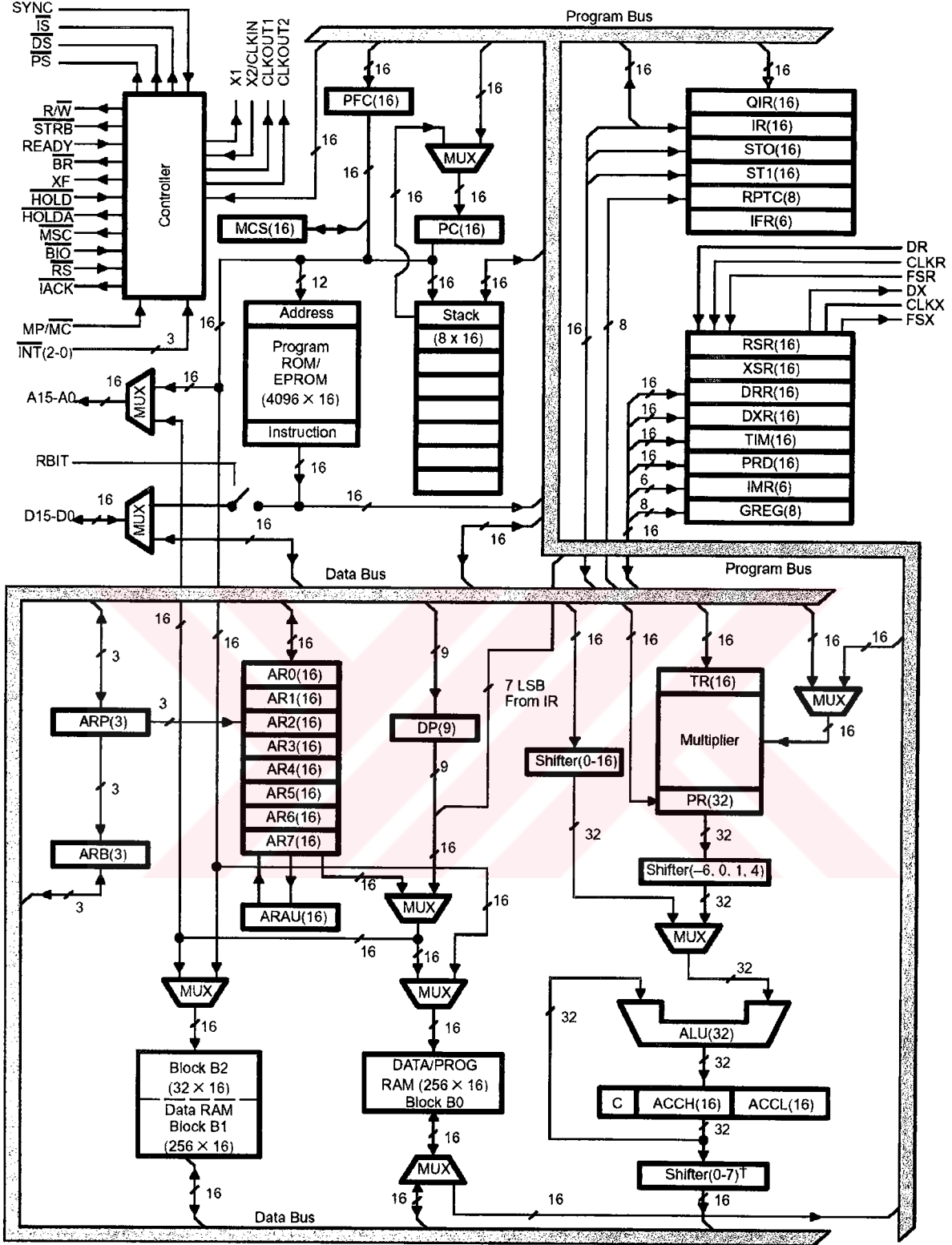
kaydırma işlemi komutun bir parçası olarak ele alınmakta, böylece işletim için ek çevrim süresine gerek kalmamaktadır. (Texas Instrument, 1993)

32 bitlik ALU ve akümülatör üniteleri, lojik ve aritmetik işlemleri gerçekleştirmektedir. Akümülatör biriminde, taşma doyma modu ile, taşma olayı algılandığında, taşma bayrağını açarak, sayının en büyük pozitif veya negatif sayı ile yüklenmesini sağlamaktadır. ALU 'ya veri girişi akümülatörden ve çarpım ünitesinin sonuç kütüğü ile veri belleğinden yüklenen ölçekleme kaydırma biriminden olmaktadır. 32 bitlik akümülatör birimi veriyi, 16 bitlik iki parça olarak işlemektedir. Çıkışında bulunan kaydırma üniteleri 0 ile 7 bitlik sola kaydırma işlemi gerçekleştirmektedir. Akümülatörde bulunan elde biti, geliştirilmiş aritmetik ve dallanma işlemlerine izin vermektedir.

TMS320C25, 16*16 bitlik çarpma işlemlerini 32 bitlik sonuç ile tek komut çevrim süresinde gerçekleştirmektedir. Çarpım ünitesinde iki kütük bulunmaktadır. Çarpma işleminin bir sayısını saklayan 16 bitlik geçici kütük ve 32 bitlik sonucu saklayan sonuç kütüğüdür. Sonuç kütüğünün çıkışı bir veya dört bit sola kaydırma işlemi gerçekleştirmektedir. Bu kesirli sayılarla işlem yaparken , doğruluğun artırılmasını sağlamaktadır. Aynı zamanda çıkış, altı bit sağa kaydırma işlemi gerçekleştirerek, 128 çarpma-yükleme işleminin taşma olmadan gerçekleşmesine olanak sağlamaktadır. (Texas Instrument, 1993)

Çarpım ünitesi işaretli ve işaretsiz çarpma işlemi, özel komut seti yardımıyla tek saat çevriminde gerçekleştirebilmektedir.

İşlemcinin parçalı çalışma ile, komutları, alma, çözme ve işlemeyi tek saat çevriminde gerçekleştirmesi, 100 ns'lik bir sürede komutun işlemlerini sağlamaktadır. Komut setinde bulunan 133 komuttan 97 tanesi tek saat çevriminde işlemektedir. 36 komut işleme için birden fazla saat çevrimine gereksinme duymasının sebebi, dallanma, alt program çağırma veya dönme gibi işlemler, program sayacının değerinin değişmesi ve parçalı çalışma modunun bozulmasıdır. Komutların paralel bir yapıda çalışabilmesi, çarpma, yükleme ve veri iletme işlemlerinin aynı komut çevrimi içinde gerçekleşmesini sağlamaktadır.



Şekil 5.2 : TMS320C25 fonksiyon blok diyagramı

Örneğin MACD komutu, çarpma, yükleme ve veri iletme işlemlerini 100 ns sürede bir saat çevriminde tamamlamaktadır. Bu komutun işlenmesi şu aşamalar ile olmaktadır: 32 bitlik P kütüğü, çıkış kaydırma birimi tarafından ötelenir. 32 bitlik ALU ötelenmiş sonucu, 32 bitlik

akümülatör içeriği ile işler. Veri belleğinin 16 bitlik içeriği, T kütüğüne yüklenir. Bu adres yardımcı kütükler ile indirekt olarak adreslenir. Program belleğindeki adresin 16 bitlik içeriği, çarpım ünitesine girilir ve 16*16 bitlik çarpma işlemi gerçekleştirilir. 32 bitlik sonuç bir sonraki çevrimde kullanılmak üzere P kütüğüne yüklenir. Veri belleği adresinin 16 bitlik içeriği, bir sonraki adrese kopyalanır. Durum kütüğündeki taşıma ve elde bayrakları işletilir. Yardımcı kütük gösterici tarafından işaret edilen 16 bitlik adres, bir sonraki çevrimde veri belleği tarafından kullanılmak üzere hazırlanır. Sayacın 16 bitlik içeriği artırılarak, bir sonraki çevrimde program belleği adresi olarak kullanılmak üzere hazırlanır. Tekrar sayıcısının değeri bir azaltılır. Bütün bu işlemlerin tek komut çevrimi süresinde gerçekleşmesi, veri belleğinden veri alındığı anda, paralel olarak program belleğinden de veri alınması ile sağlanmaktadır. (Küçükdemiral, 1999)

Texas Instruments firmasının sabit nokta ile işlem yapan birinci ve ikinci nesil DSP sistemlerinden sonra, diğer firmalar da bu alanda yeni ürünler çıkararak, pazardan pay kapmaya çalışmışlardır. Özellikle Analog Devices firması, ikinci nesil DSP ile rekabet edebilecek özellikleri olan ADSP-2101 ve ADSP-2105 ürünlerini sunmuştur.

ADSP-2101 , mikrobilgisayar yapısında olup, ADSP-2100 işlemcisine bellek, seri port ve zamanlama birimlerinin eklenmesi ile elde edilmiştir. İkinci nesil C2x DSP'lerine rakip olarak üretilmişlerdir. Temel özellikleri şu şekilde sıralanabilir: 16.67 MHZ kristal, 60 ns komut çevrim süresi, 2K 24 bitlik program RAM bellek, 1 K veri RAM bellek, ayrı program ve veri yolları, bağımsız ALU, MAC ve kaydırma üniteleri, iki adet veri adres generatörü, program sıralama ünitesi, iki adet seri port, zamanlama birimi, üç dış kesme işareti, 68 pinli yapıdır. ADSP-2105 işlemcisi de yine aynı ailenin bir elemanı olup, 10 MHZ kristal, 100 ns komut çevrim süresi, 1K program RAM bellek, 512 kelime veri RAM bellek, bir seri port özellikleri dışında, ADSP-2101 ile aynı mimari yapıdadır. (Analog Devices, 1996)

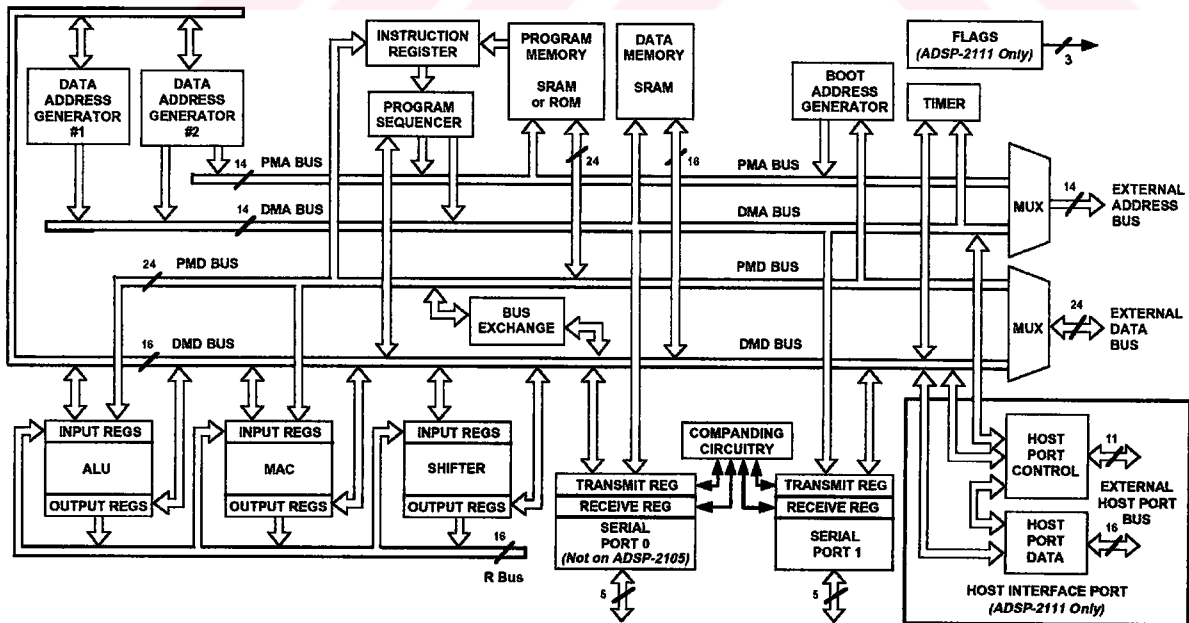
ADSP-2101'in esnek mimari yapısı ile, paralel işleme yapabilmektedir. Tek çevrim süresinde, bir sonraki program adresini üretebilmekte, bir sonraki komutu alabilmekte, bir veya iki veri iletimi yapılabilmekte, bir veya iki veri adres gösterici düzeltebilmekte, aritmetik bir işlem gerçekleştirebilmekte, iki seri porttan veri alabilmekte veya yollayabilmektedir.

ADSP-2101 'in mimari yapısı birbirinden tamamen bağımsız işlem yapabilen ALU, MAC ve kaydırma ünitelerinden oluşmuştur. Bu üniteler 16 bit veri ile işlem yapmaktadır. ALU standart aritmetik ve lojik işlemleri, ayrıca basit bölme işlemlerini gerçekleştirmektedir. MAC ünitesi, bir saat çevriminde, çarpma, çarpma-toplama ve çarpma-çıkarma işlemlerini

gerçekleştirmektedir. (Analog Devices, 1996)

Kayıdırma ünitesi, lojik ve aritmetik kaydırma işlemleri, normalize ve denormalize işlemleri yapmaktadır. Ayrıca üs kontrolü ile, sayı format kontrolü yaparak, kayan nokta gösterimi sağlamaktadır. Bu üniteler, giriş ve çıkışlarında bulunan kütükler yardımıyla, veri bellek veri yoluna bağlanmaktadır. İşlem sırasında, veri ilk olarak giriş kütüklerine gelmektedir. Sonuçlarda çıkış kütükleri üzerinden sonuç yoluna gönderilmektedir. Kütükler üzerinden verinin iletimi, giriş ve çıkışta bir seviyeli iş hattı çalışmasına izin vermektedir.

Veri transferini verimli kılmak için beş adet yol kullanılmaktadır. Bunlar; program bellek adres yolu (PMA), program bellek veri yolu (PMO), veri bellek adres yolu (DMA), veri bellek veri yolu (DMD) ve sonuç yolu (R) 'dur. Sonuç yolu, üniteleri birbirine bağlayarak, herhangi bir ünitenin çıkışının diğer ünitenin girişi olması sağlamaktadır. Program sıralama ünitesi ve veri adres generatörü, işlem ünitelerinin verimli kullanımını sağlamaktadır. Sıralama ünitesi, koşullu atlama, alt program çağırma ve geri dönme işlemlerini tek saat çevriminde gerçekleştirmektedir. Sıralama ünitesi, işlemekte olan komutu saklayan komut kütüğü tarafından sürülmektedir. Komut kütüğü, program akışında tek seviyeli iş hattı çalışması gerçekleştirmektedir. Komutlar, tek saat çevriminde alınarak, komut kütüğüne yüklenmekte ve çözümlenmektedir. Bir sonraki çevrimde işletilmekte iken diğer komut alınmaktadır. (Analog Devices, 1996)



Şekil 5.3 : ADSP-2101 fonksiyon blok diyagramı

DAG ünitesi, program ve veri belleğinden veri aktarımı için adres üretimini sağlamaktadır.

Her bir DAG ünitesinin, dört adres göstericisi bulunmaktadır. Bu göstericiler kullanılarak, değişik adresleme modlarının kullanımı mümkün olmaktadır

DMA yolu 14 bit genişliğindedir ve 16K kelime genişliğinde veriyi adresleyebilir. DMD yolu ise, 16 bit genişliğindedir. DMD yolu, işlemcinin herhangi bir kütüğünden bir başka kütüğe veya dış veri belleğine veri iletimini tek saat çevriminde gerçekleştirmektedir. Veri bellek adresleri, iki ayrı kaynaktan gelebilir. Bunlar; doğrudan adresleme olarak adlandırılan bir komutun çözümlenmesi ile elde edilen değer ve indirekt adresleme denilen veri adres generatöründen gelen değerdir. Program belleğinden veri iletimi için sadece indirekt adreslemeye izin verilmektedir. PMA ve DMA yolları, tek bir yol ile dış adres yoluna bağlanmakta ve dış bellek biriminin adreslenmesini sağlamaktadır. PMD ve DMD yolları da tek bir yol ile dış veri belleğinin kullanılmasını sağlamaktadır. Program belleği, hem komut ve hem de veriyi saklayarak, tek çevrim içinde, biri program belleğinden diğeri veri belleğinden alınan iki verinin işlenmesini sağlamaktadır. (Analog Devices, 1996)

Sabit nokta işlemcilerin üretiminden sonra, daha büyük dinamik sayı aralığı ve daha hızlı çalışan işlemcilere ihtiyaç duyulması, kayan nokta işlemcilerin üretilmesini sağlamıştır. İlk olarak TI firması tarafından üretilen bu işlemciler, üstün hızları ve paralel işlem yapabileme özellikleri sayesinde uygulama alanına girmişlerdir.

TMS320 ailesinin üçüncü nesil işlemcisi, TMS320C30, kayan nokta işlem yapan DSP sisteminin ilk örneğidir. 33 MFLOPS işlem kapasitesi ile yüksek performans işlem yapma yeteneğine sahiptir. Tek saat çevriminde kayan nokta işlem yapabilmesi ve paralel çalışma, sisteme üstün bir hız kazandırmıştır. TMS320C30 'un temel özellikleri; 60 ns saat çevrim süresi, 16 MIPS işleme oranı, iki 1K 32 bitlik RAM bellek, 4K 32 bitlik ROM bellek, 64*32 bit komut bellek, 32 bit komut ve veri kelimeleri, 24 bit adres, 32/40 bit kayan nokta ve tamsayı çarpma ünitesi, 32/40 bit kayan nokta, tamsayı ve lojik ALU, 32 bit yığın kaydırma ünitesi, 8 geliştirilmiş aritmetik kütüğü, 8 yardımcı kütük ile iki adres generatörü, 180 pinli yapıdır. (Texas Instruments, 1997)

TMS320C30 'un üniteleri; kayan nokta ve tamsayı ile işlem yapabilmektedirler. Çarpma ünitesi tamsayı ve kayan nokta sayı formatlarındaki sayılar ile işlem yapabilmektedir. Kayan nokta çarpma işleminde, girişler 32 bit, çıkış 40 bit genişliğindedir. Tamsayı çarpma işleminde, giriş 24 bit iken, sonuç 32 bit genişliğinde olmaktadır. ALU ünitesi, 32 bit tamsayı 32 bit lojik ve 40 bit kayan nokta sayılarla işlem yapabilmektedir. (Texas Instruments, 1997)

Çarpım ve ALU ünitelerinin çıkış sayı formatı, daima, tamsayı kullanıldığında 32 bit, kayan

nokta kullanıldığında 40 bit genişliğindedir. İşlemci tek çevrimde çarpma veya çarpma toplama/çıkarma işlemlerini sabit veya kayan nokta sayılar için gerçekleştirebilmektedir. Kayan nokta ile işlemler, sayıların dinamik aralığını arttırdığından, sayısal işlemler sonunda oluşan hataların ihmal edilebilecek düzeyde olmasını sağlamaktadır. Kütük dosyası, 28 kütükten oluşmuştur ve çarpım ve ALU üniteleri ile işlemlerde kullanılmaktadır. Bu kütüklerden ilk 8 tanesi, geliştirilmiş doğrulukta işlem için kullanılan kütüklerdir. Bu kütükler, 40 bit kayan nokta ve 32 bit tamsayı işlemleri desteklemektedir. Daha sonraki 8 kütük, yardımcı kütüklerdir ve adres üretimini sağlamaktadır. İki adet yardımcı kütük aritmetik işlem ünitesi, tek saat çevriminde iki adres üretebilmekte ve ALU ve çarpım üniteleri ile paralel çalışabilmektedir.

TMS320C30 DSP 'si değişik sayı formatları ile uyumlu çalışabilir. Tamsayı işlemlerde iki format kullanılmaktadır. İşaretsiz ve işaretli sayılar için, acele tamsayı işlemlerde 16 bit, diğer tamsayı işlemlerde 32 bit sayı formatı kullanılmaktadır. Kayan nokta işlemlerde ise, üç ayrı format bulunmaktadır. Acil işlemler için kullanılan 16 bitlik kısa kayan nokta formatında 4 bit üs, 11 bit kesir ve 1 işaret biti bulunmaktadır. Normal işlemlerde kullanılan formatta 8 bit üs, 23 bit kesir ve 1 işaret biti bulunmaktadır. Geliştirilmiş doğrulukta işlem yapmak için kullanılan format ise, 8 bit üs, 31 bit kesir ve 1 işaret olmaktadır. (Texas Instruments, 1997)

TMS320C30 işlemcisinde doğrudan bellek erişimini sağlayan bir denetleyici bulunmaktadır. Bu denetleyici, bellek haritasının istenilen bir adresine, ana işlem birimi ile haberleşmeye gerek kalmadan, veri yazılıp okumayı sağlamaktadır. Ayrıca, bellek veya diğer çevre birimleri ile bağlantı kurulurken, işlemcinin MİB 'si ile ilişkiye gerek kalmadığından sistemin hızında bir düşme olmayacaktır. Doğrudan bellek erişim denetleyicisinin kendi adres generatörü, kütük yapısı ve sayacı bulunmaktadır. Adres ve veri yolları, MİB ile denetleyici arasında ilişki olmadan işlem yapılabilir. (Texas Instruments, 1997)

Üçüncü nesil DSP 'lerde iş hattı işlemi beş ana birim ile kontrol edilmektedir. Program sayacının artmasını ve bellekten gelen komutların işlenmesini kontrol eden alma ünitesi, komutu çözen ve adres üretimini kontrol eden çözme ünitesi, bellekten veri okunmasını kontrol eden okuma ünitesi, kütük dosyasından değerler okuyup, gereken işlemi yapan ve daha sonra sonucu kütük dosyasına veya belleğe geri yazan işletme ünitesi, ve bellekten bağımsız olarak veri okuyup yazabilen doğrudan belleğe erişim kanalı iş hattı işlemini gerçekleştirmektedir. Bütün komutlar bu ünitelerin gerçekleştirdiği işlemlerden geçerek çalışmaktadır. Tam anlamıyla bir hız eldesi için bütün üniteler aynı anda değişik komutları gerçekleştirebilirler. Değişik komutlarda bu işlemlerin çakışmasına iş hattı denilmektedir.

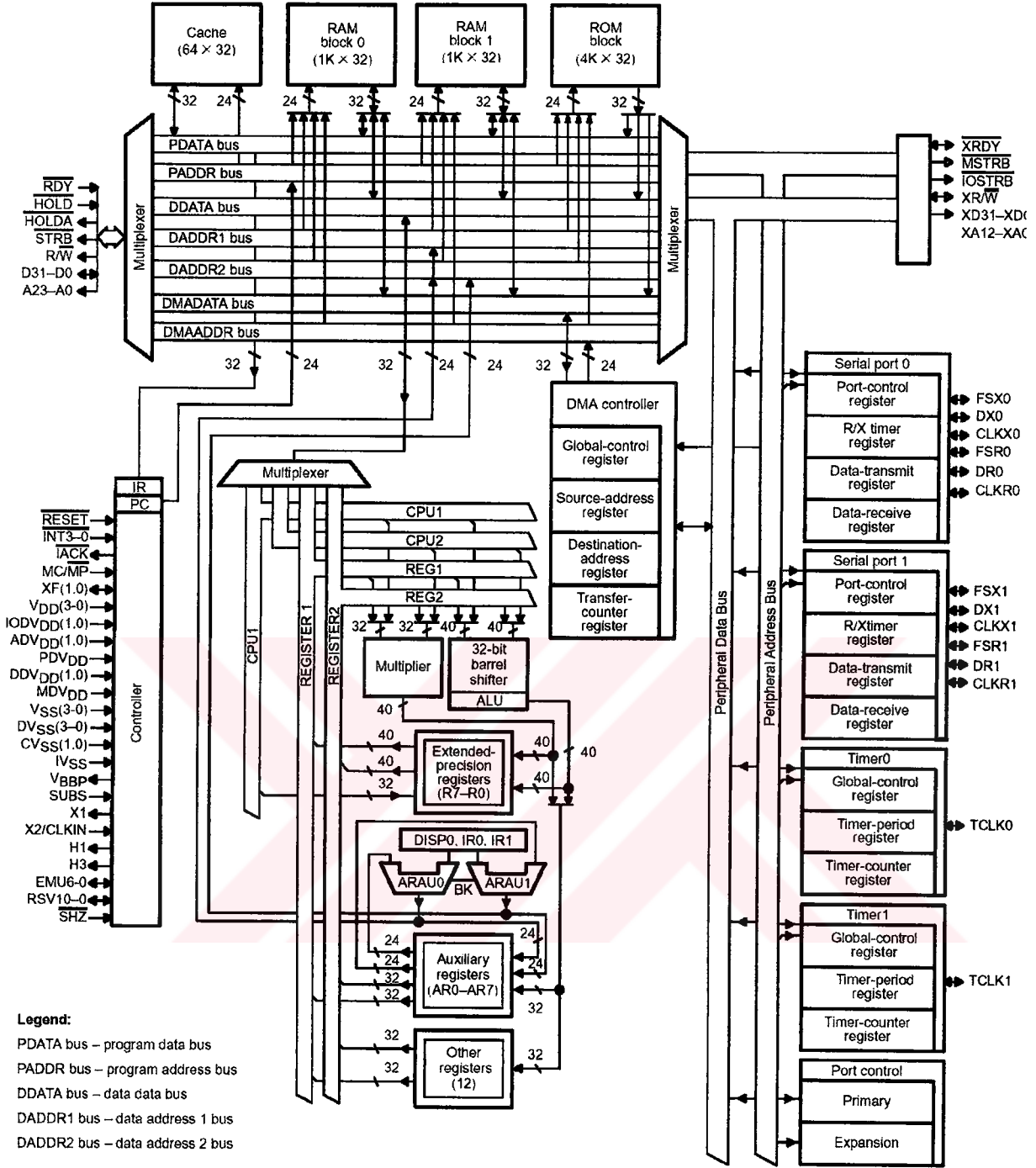
Doğrudan erişim kanalı bütün bu ünitelerden bağımsız işlemektedir. Tek bir saat çevriminde işleyen komutların kontrolü ile çok daha yüksek performanslara erişilmektedir.

Analog Devices firması tarafından üretilen kayan nokta işlemcisi ise, ADSP-21010 ve ADSP-21020 'dir. ADSP-21010 sabit nokta DSP 'de olduğu gibi, bellek, çevre birimleri ve hız olarak daha az fakat daha ucuz bir modeldir. ADSP-21010 temel özellikleri; 80 ns komut çevrim süresi, 12,5 MIPS işleme oranı, 37,5 MFLOPS tepe ve 25 MFLOPS ortalama performans değeri, 32 bit kayan nokta formatı, 80 bitlik akümülatörler ile 32 bit tamsayı ve kesir sabit nokta formatı, çarpma, ALU ve kaydırma üniteleri, iki veri adres generatörü, tek çevrim çarpma ve ALU işlemleri, program sıralama ünitesi, 304 pinli yapıdır. ADSP-21 020 ise, 33,3 MHZ 30 ns komut çevrim süresi, 40 bit gelişmiş doğrulukta kayan nokta işlemi ve 224 pinli bir yapıdır. Örneğin 1K kompleks FFT işlemini 21020 0,58 ms de tamamlarken, 21010 1,54 ms 'de tamamlamaktadır. (Analog Devices, 1996)

DSP 'nin mimari yapısını oluşturan ALU, çarpma ve kaydırma ünitesi birbirine paralel işlem ve tek çevrimde işlem yapabilmektedir. Bir komut ile ALU ve çarpım ünitelerinin paralel ve aynı çevrim içinde çalışması mümkündür. Bu üniteler, 32 bit sabit nokta ve 32 bit kayan nokta sayı formatını desteklemektedirler.

ADSP-21010 'in Harvard mimarisinden dolayı, veri belleği veri, program belleği ise hem komut hem de veri saklamaktadır. Program, veri ve saklı bellek alanlarının ayrı olmasından dolayı, işlemci aynı komut çevrimi içinde, veri belleği, program belleği ve saklı bellek ile işlem yapabilir.

Aritmetik işlemler birbirinden bağımsız üç ünite tarafından gerçekleştirilmektedir. Bu üniteler, 32 bit kayan nokta ve 32 bit sabit nokta sayı formatını kabul etmektedir. Çarpma ünitesi, kayan nokta ve sabit nokta çarpma ile sabit nokta çarp-topla ve çarp-çıkart işlemlerini gerçekleştirmektedir. Tamsayı sonuçlar 64 bit genişliğinde iken, akümülatör 80 bit genişliğindedir. ALU 45 değişik aritmetik ve lojik işlemi yapmaktadır. Kaydırma ünitesi de 32 bitlik sayılar üzerinde 19 değişik işlemi gerçekleştirmektedir. Bu üniteler birbirlerine paralel bağlı olduklarından, bir sonraki çevrimde bir ünitenin çıkışı diğerinin girişi olabilmektedir.



Şekil 5.4 : TMS320C30 fonksiyon blok diyagramı

Dördüncü nesil DSP ailesi yine TI firması tarafından üretilmiştir. TMS320C4x kayan nokta işlemcilerin en gelişmiş olanıdır ve tam anlamıyla paralel çalışma sağlamaktadır. 275 MFLOPS ve 320 Mbyte/s işlem kapasitesi , 40-50 ns komut çevrim süresi, on bir komutun aynı anda işlenebilmesini sağlayan bu paralel çalışma, 6 iletişim portu, 6 kanal DMA yardımcı işlemci bu işlemcinin temel özellikleridir. Beşinci nesil DSP ise, TMS320C5x serisi olarak çıkartılmış ve sabit nokta işlemcilerin oldukça gelişmiş olanıdır. Temel özellikleri şu

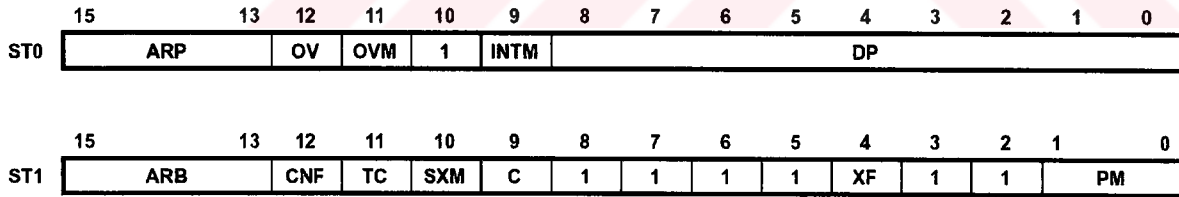
şekilde sıralanabilir. 50/35 ns komut işleme süresi, 8K kelime program/veri RAM bellek, 2K ROM bellek, 544 kelime veri/program RAM bellek, 128K adreslenebilir toplam bellek ve 84 pinlik yapıdadır.

5.1 TMS320C240 DSP Entegresinin Mimarisi

TMS320C240 yongası program ve veri yolu olmak üzere iki ayrı yol içeren Harvard Mimarisi ile üretilmiştir. Yonganın program ve veri için ayrı yolları kullanması işlemcinin veri ve komutları aynı anda okuyup , icra etmesine imkan sağlamaktadır. Ayrıca işlemci 4 seviyeli iş hattı (pipe-line) çalışmasına uygundur. Bu da aynı anda birden fazla komutun icra edilebilmesini sağlar.

5.1.1 Durum ve Kontrol Kütükleri

TMS320C240, çeşitli çalışma modları ve durumlarını içeren iki adet durum kütüğüne sahiptir. Bu kütükler veri hafızasına depo edilir ve yine oradan okunur. ST0 ve ST1 adı verilen bu kütüklere yazmak için LST komutu kullanılır. Ancak INTM biti LST komutundan etkilenmez. INTM bitini değiştirebilmek için SETC ve CLRC komutları kullanılabilir. Aşağıdaki şekilde durum kütüklerinin yapısı basit olarak gösterilmiştir. Durum kütüklerinin bir kısmı lojik 1 sabit değerindedir.



Şekil 5.5 : Durum ve kontrol kütükleri düzeni

Aşağıda ST0 ve ST1 kütüklerindeki kontrol bitlerinin tanımları verilmiştir.

ARB : Yardımcı kütük işaretleyici bufferi. ARP içeriği ST0 kütüğünden çağırıldığında, eski ARP değeri bu alana depolanır. ARB içeriği LST #1 komutuyla çağırıldığında aynı değer ARP alanına da kopyalanır.

ARP :Yardımcı kütük işaretlecisi. ARP, yardımcı kütüğün (AR) endirekt adreslenmesi sırasında kullanılır. ARP içeriği çağırıldığında eski ARP içeriği ARB kütüğüne kaydedilir. ARP endirekt adresleme ve LARP, MAR, LST gibi komutlar kullanılarak değiştirilebilir. Bir

LST #1 komutu icra edildiğinde her iki kütüğün içeriği de aynıdır.

C : Elde (Carry) biti . Bir toplama işlemi sonucunda elde varsa C bayrağı çekilir. Eğer bir çıkarma işlemi sonucunda bir borç varsa C biti lojik 0 olur. Bahsedilen durumların dışında C biti her toplama işleminden sonra resetlenir ve her çıkarma işlemiden sonrada 1'lenir. Ancak 16 bitlik ADD ve SUB komutları istisnai durumlardır. ADD, C bitini sadece set ve SUB komutu ise sadece resetleyebilir. Bunun dışındaki durumlarda elde biti etkilenmez. Yalnız bit kaydırma ve döndürme komutları (SETC, CLRC ve LST #1 gibi komutlar) elde bitini etkiler. Elde biti, yonga resetlendiğinde 1 değerini alır.

CNF :Dahili RAM kontrol biti. Eğer CNF sıfırlanmış ise yeniden programlanabilir, yapılandırılabilir DARAM (iki çeşit işlem yapabilen RAM) veri alanı olarak kullanılır. Bunun haricinde bahsedilen hafıza alanı program hafızası olarak da kullanılabilir. CNF biti SETC, CNF, CLRC, CNF ve LST #1 komutları ile değiştirilebilir. \overline{RS} biti aktif hale getirildiğinde CNF biti 0 değerini alır.

DP : Veri hafıza sayfa işaretçisi. DP alanı LST ve LDP komutları yardımıyla değiştirilebilir.

INTM : Kesme modu bayrağı. INTM bitinin sıfırlanması tüm maskelenmiş kesmelere izin verir. Bu bitin 1'lenmesi durumunda ise hiçbir maskelenebilir kesmeye izin verilmemiş olur. INTM biti, SETC INTM ve CLRC INTM komutları ile değiştirilebilir. Ayrıca \overline{RS} ve \overline{IACK} uçları ile de bu bit 1'lenebilir. INTM biti maskelenemez \overline{RS} , NMI kesmelerini etkilemez. Bunun yanı sıra INTM bitinin LST komutu tarafından etkilenmeyeceğine dikkat edilmelidir. Yonganın resetlenmesi ile bu bit 1 değerini alır.

OV : Taşma bayrağı . Bu bit ALU da bir taşma olduğunda 1 konuma gelir. Bir kez taşma bayrağı çekildiğinde, bayrak resete ya da bit temizlenene kadar 1 konumunda kalır.

OVM : Taşma modu bayrağı. OVM sıfırlandığında taşma modu iptal edilir ve işlem sonucunda elde edilen sayının değeri değiştirilmeden akümülatöre aktarılır. OVM birlendiğinde ise taşma doyum modu aktif duruma geçer. Taşmalı sonuç maksimum pozitif ya da maksimum negatif değere eşitlenerek akümülatöre aktarılır. SETC ve CLRC komutları kullanılarak bu bitin içeriği değiştirilebilir.

PM : Sonuç kaydırma modu seçim bitleri. Bu bitlerin durumuna göre çarpım ünitesinin çıkışı belirli oranda kaydırılır ya da hiç kaydırılmadan ALU'ya aktarılır.

Çizelge 5.2 : PM bitlerinin durumuna göre yapılacak işlemlerin listesi

PM	Yapılacak işlem
00	Çarpım ünitesinin 32 bitlik çıkışı kaydırma olmaksızın ALU'ya aktarılır
01	PREG çıkışı 1 bit sola kaydırılır ve LSB sıfır ile doldurulur
10	PREG çıkışı 4 bit sola kaydırılır ve 4 adet LSB sıfırla doldurulur
11	PREG çıkışı 6 bit sağa kaydırılır ve 1 bitlik işaret genişletmesi yapılır

PM bitleri ne olursa olsun PREG kütüğünün içeriği korunur. Kaydırma işlemi PREG içeriğinin ALU'ya aktarımı sırasında yapılır. PM bitleri SPM ve LST#1 komutları ile çağırılır ve \overline{RS} ucu ile temizlenebilir.

SXM : İşaret genişletme modu kontrol biti. Bu bit yardımı ile aritmetik işlemlerdeki 1 bitlik işaret genişlemesi aktif hale getirilebilir. SXM = 1 ise işaret genişlemesi aktif hale gelir. Veri , ölçekleyici üzerinden akümülatöre aktarılırken işaret genişlemesi gerçekleştirilir. SXM = 0 ise bu mod iptal edilir.

TC : Test/Kontrol bayrağı. Bu bit BIT, BITT, CMPR, LST #1 ve NORM komutları ile değiştirilebilir. Eğer BIT ya da BITT komutlarıyla test edilen bit 1 değerine sahip ise TC bayrağı çekilir. Bunun dışında AR ve AR0 arasında CMPR komutu ile bir karşılaştırma yapıldığında koşul sağlanmış ise ya da akümülatör içeriğindeki iki MSB biti arasında EX-OR işlemi yapıldığında sonuç doğru (yani 1) olduğunda da TC bayrağı çekilir. Koşula bağlı dallanma, alt program çağırma ya da çağırılan alt programdan geri dönme işlemleri TC bayrağının durumuna bağlı olarak icra edilebilir.

XF : XF bacağı durum biti. XF, genel amaçlı çıkış bacağı olan XF'in durumunu belirtir. Bu bit SETC XF komutu ile kurulup, CLRC XF komutu ile temizlenebilir. Yongaya reset atılması durumunda XF, 1 değerini alır.

5.1.2 Merkezi İşlem Birimi

TMS320C240'ın merkezi işlem birimi şu üniteleri içerir : 16 bitlik ölçekleme kaydırıcısı , 16x16 bitlik paralel çarpıcı, 32 bitlik merkezi aritmetik lojik ünite (CALU), 32 bitlik akümülatör, akümülatör ve çarpıcı çıkışında birer adet kaydırıcı. Merkezi işlem birimini

oluşturan üniteler aşağıda tek tek açıklanmıştır.

5.1.2.1 Giriş Ölçekleme Kaydırıcısı

TMS320C240 yongası 16 bitlik giriş ile veri yolu arasında ve 32 bitlik çıkış ile Merkezi Aritmetik Lojik Üniteyi birleştiren ölçekleme kaydırıcısına sahiptir. Bu kaydırıcı ünitesi CALU'nun program ya da veri alanından gelen veri yolunun bir parçası olarak çalışır ve ek bir zaman kaybı üretmez. Bu ünite 32 bitlik CALU'nun hafızasından gelen veriyi düzenlemek için kullanılır. Ölçekleyici ünitesi giriş verisini sıfırdan 16 bite kadar sola kaydırılabilir ve LSB'leri sıfır ile doldurulur. Ayrıca MSB'ler de sıfır ya da işaret genişleme bitinin değeri ile doldurulur. Bu ST1 durum kütüğündeki SXM bitinin değeri ile belirlenir. Kaydırılacak bit sayısı komut kelimesindeki sabit sayı değeri ile ya da TREG değerine göre belirlenir. Komut ile kaydırılacak bit sayısının belirlenmesi özel ölçeklemelere ve düzenleme işlemlerine imkan verir. TREG tabanlı kaydırma ise ölçekleme faktörünün sistem performansına göre uyarlanmasını sağlar.

5.1.2.2 Çarpım Ünitesi

TMS320C240 DSP'si 16x16 bit'lik işaretli ve işaretsiz çarpma işlemini bir makina çevriminde gerçekleştirebilen bir çarpım ünitesine sahiptir. Sonuç ise 32 bit genişliğinde olmaktadır. MPYU (işaretsiz çarpım) komutu dışındaki tüm çarpma komutları işaretli çarpım yaparlar. Çarpılan her iki sayı da 2'ye tümlenmeli sayılardır ve sonuç da 32 bit genişliğinde 2'ye tümlenmeli bir sayıdır. Çarpım ünitesi ile ilgili iki kütük mevcuttur. Bunlardan birincisi 16 bit genişliğindeki geçici kütük TREG, ikincisi ise 32 bitlik sonuç PREG kütüğüdür. TREG kütüğü çarpanlardan bir tanesini içerir. PREG kütüğü ise çarpım sonucunu saklar.

Çarpım işleminin gerçekleştirilmesi sırasında önce çarpanlardan birisi hafızada adreslenmiş geçici kütüklerden TREG'a getirilir ve diğer çarpan ise veri ya da program yolundan sağlanır. Elde edilen 32 bitlik sonuç PREG adı verilen kütükte ALU işlemleri için hazır bekletilir. PREG çıkışında (PSCALE) 4 farklı sonuç kaydırma modu bulunmaktadır. Bu sayede PREG çıkışı çarpma/depolama işlemleri ile kesirli aritmetik ve kesirli çarpma işlemlerini yapabilir. Bahsedilen 4 adet kaydırma modu ST1 durum kütüğündeki PM bitleri ile belirlenir. PM bitlerinin durumlarına göre hangi kaydırma modunda çalışılacağı aşağıdaki çizelgede belirtilmiştir.

Çizelge 5.3 : PM bitlerinin durumuna göre yapılacak kaydırma miktarı.

PM	KAYDIRMA	TANIM
00	Kaydırma işlemi yapılmaz	Sonuç doğrudan ALU'ya aktarılır
01	PREG çıkışı sola bir bit kaydırılır	LSB 0 ile doldurulur. Bu mod yardımı ile 16 bitlik iki sayı çarpıldığında oluşan 1 bit boyutundaki fazlalık işaret bitinin etkisi ortadan kaldırılmış olur.
10	PREG çıkışı 4 bit sola kaydırılır	4 adet LSB 0 ile doldurulur. Bu kaydırma işlemiyle MPY çarpma komutunun kısa ani değerler ile birlikte (13 bit veya daha az) kullanıldığında ve diğer çarpanın 16 bit uzunluğunda olması durumunda elde edilen 4 bitlik fazlalık işaret bitlerinin atılması sağlanır.
11	PREG çıkışı 6 bit sağa kaydırılır	6 LSB biti kaybolur. Bu mod yardımı ile 128'e kadar arka arkaya çarpma/akümülatöre yükleme işlemi taşma problemi olmaksızın gerçekleştirilebilir. Bu sırada PREG'deki bilgi , kaydırma sırasında değişmez

LT komutu ile TREG'a veri yolu ile gelen bilgi yüklenir. MPY komutu ile de çarpma işlemleri için gerekli ikinci çarpan sağlanmış olur. Kısa ya da uzun ani değer operatörü ile çarpım yapıldığında , MPY komutu bir ani değer operatörü ile birlikte kullanılır. DSP bir çok çarpma ve sonuçla toplama işlemini ardarda yaptığında MİB iş hattı çalışmasını mümkün kılar . Basit olarak iş hattı çalışması şu şekilde gerçekleştirilir. Akümülatör PREG ile yüklenir, PREG akümülatör ile toplanır, ve TREG giriş verisi veri hafıza bloğunda sonraki adrese kaydırılır ve akümülatör içeriğinden PREG çıkarılır.

Mevcut 4 değişik çarpma /yükleme komutu (MAC, MACD, MADD, MADS) yardımı ile her türlü çarpma işlemi gerçekleştirilebilir. Bu komutlar sayesinde her iki operatör üzerinde de anda işlemler yapılabilir. Bu operatörler için gerekli olan veri , her çevrimde , veri ve program yollarından çarpıcılara sağlanır. Bu komutlar , tekrar komutları olan RPT ve RPTZ ile birlikte kullanıldıklarında , bir tek çevrimlik çarpma/yükleme komutu olarak işlem görürler. Bu tip tekrarlamalı komutlarda katsayı adresleri PS (program sayacı) ile sağlanırken , veri adresleri

ARAU (yardımcı kütük veya kaydedici aritmetik ünitesi) tarafından sağlanır. RPTZ komutu yapacağı işlemlere başlamadan önce ACC'yi ve PREG'in içeriğini temizler. Örneğin 10 x 10 boyutundaki iki matristen birinin satırı ile diğerinin sütununu çarpalım. MTRX1 birinci matrisin başlangıç adresinin , INDX = 10 matris boyutunu göstermek üzere , o andaki AR ikinci matrisin başlangıç adresinin ifade etsin. Bu durumda program ifadesi aşağıdaki gibi olur

```
RPTZ #9           ; FOR I=0 , I<10 , I++
MAC MTRX1,*0+    ; PREG=DATA(MTRX1+I)XDATA[MTRX2+(I+INDX)]
APAC             ; ACC←PREG
```

16 bitlik iki sayının çarpımı ile elde edilen 32 bitlik sonuç yine 32 bitlik PREG kütüğüne kaydedilir. Bu kütükte depolanan sonuç CALU ya da veri hafızasına SPH ve SPL (16 bitlik iki kısımda) olmak üzere gönderilebilir.

TMS320C240 yongası bir çoğu bir tek çevrim süresinde tamamlanabilen çok çeşitli işlemleri merkezi aritmetik lojik birimi (CALU) yardımıyla kolayca gerçekleştirebilir. CALU tarafından bir işlem gerçekleştirildiğinde sonuç daha sonra yapılacak (kayıdırma gibi) işlemlerin yapılabilmesi için akümülatöre (ACC) aktarılır. CALU'ya gelen veri ölçeklenebilir. Eğer veri, veri yolu ile geliyorsa bunu ISCALE, çarpım ünitesinin çıkışından geliyorsa PSCALE bu işlemi yapar.

CALU 16 bit genişliğinde genel amaçlı aritmetik ya da lojik işlemleri yapan birimdir. CALU'nun işlem yapacağı verilerden bir tanesi mutlaka akümülatör üzerinden sağlanır. Diğer veri ise çarpım birimi çıkışından ya da ölçekleyiciden elde edilir. Sonuç ise mutlaka akümülatöre aktarılır.

TMS320C240 geniş bir dinamik aralık gerektiren uygulamalara cevap verebilmek için kayan nokta aritmetiğini de kullanabilir. NORM komutu akümülatörde saklanan sabit nokta'lı verinin sola kaydırma yapmak suretiyle normalize edilmesini sağlar. TREG kütüğünün bitlerinden dört tanesi ölçekleyici tarafından yapılacak kaydırma işleminin sayısını (LACT/ADDT/SUBT komutları için) tanımlar. Bu komutlar (sabit noktalı) sayıların denormalize edilmesinin gerektiği kayan nokta aritmetiği için oldukça kullanışlıdır. Ayrıca sözü edilen komutlar filtre uygulamalarında otomatik kazanç kontrolü sağlamak için kullanıldığında işimizi oldukça kolaylaştırmaktadır. BITT (bit test) komutu veri hafızasındaki bir kelimenin bitlerini tek tek kontrol edebilmemizi sağlar.

CALU'nun taşması durumuna karşı doyum modunun aktif hale getirilmesi (ya da yetkinin kaldırılması) ST0 kütüğündeki OVM bitinin birleşmesi (ya da sıfırlanması) ile

gerçekleştirilir. Bu mod aktif iken bir taşma oluşursa taşma bayrağı çekilir ve akümülatör içeriği en büyük pozitif ya da en küçük negatif değere atanır. Bu değer pozitif taşmada 07FFFFFFh ve negatif taşma için ise 08000000h'dir. Eğer taşma modu aktif değilken bir taşma oluşursa; taşmalı sonuç akümülatöre yüklenir. Bunun yanında lojik işlemler de taşma oluşmaz.

CALU çok çeşitli dallanma komutlarını CALU ve akümülatörün durumuna bağlı olarak icra edebilir. Örneğin aşma yönetiminde OV(taşma varsa dallan) ve EQ(akümülatör sıfıra eşitse dallan) komutları dallanma için kullanılabilir. Ek olarak, BACC (akümülatördeki adrese dallan) komutu önceden belirlenebilen bir adres alanına dallanmayı sağlar. Akümülatörü etkilemeyen bit test komutu (BIT ve BITT) veri hafızasındaki bir kelimenin bitlerini tek tek kontrol edilebilmesine imkan sağlar.

CALU bir elde bitine sahiptir. Bu bit sayesinde doğruluğu artırılmış sonuç kelimeleri ile yapılan işlemlerde ve ayrıca toplama ve çıkarma işlemlerinde daha hızlı ve verimli bir çalışma elde edilir. Ayrıca elde biti taşma yönetiminde de oldukça kullanışlıdır. Bu bit kaydırma ve döndürme komutlarının yanı sıra bir çok aritmetik işlemlerin sonucunda da etkilenir. Fakat lojik işlemlerde, kontrol komutlarının icraasında ve aritmetik olmayan işlemlerde elde biti etkilemez. ADDC (eldeli toplama) ve SUBB (eldeli çıkarma) komutları, işlem esnasında elde bitinin önceki değerini kullanırlar.

Elde biti, 16 bitlik kaydırma ile birlikte ADD komutunun kullanılması durumunda (sadece ACCH ya da ACCL ile işlem yapılmak istendiğinde) istisnai bir durum sözkonusudur. Böyle bir durumda ADD ya da SUB komutunun icrası sırasında sadece elde biti gerekiyorsa bu bit etkilenir, aksi durumda ise elde biti söz konusu komutlardan etkilenmez. İki koşullu bitler C ve NC çağırma , geri dönme ve koşullu icra gibi komutların kullanılabilmesini sağlar . SETC, CLRC ve LST #1 komutları da elde bitini çağırarak ve içeriğini değiştirmek için kullanılabilir. Elde biti cihaza donanım reseti atıldığında 1 değerini alır.

5.1.2.3 Akümülatör

32 bit boyutundaki akümülatör, CALU'nun çıkışında bulunmaktadır. Akümülatör veri hafızasına depolanırken ACCH ve ACCL olmak üzere 16 bitlik iki kısma ayrılabilir. Akümülatörün çıkışında bulunan kaydırıcılar akümülatör içeriğini 0 ile 7 bit arasında sola kaydırabilirler. Bu kaydırma işlemi içerik, veri yolu ile depolanmak üzere gönderilirken gerçekleştirilir. Ancak akümülatör içeriği aynen korunur. Kaydırma ünitesi akümülatörün yüksek değerli kelimesinde (16 – 31 arası bitler) işlem yaptığında MSB'ler kaybolur ve bu

kaydırılan bitler yerine LSB'ler (0-15 arası bitler) getirilir. Eğer kaydırma ünitesi akümülatör içeriğinin düşük değerli kelimesi üzerinde işlem yapıyorsa bu durumda kaydırılan bitlerin yeri 0'larla doldurulur.

SFL ve SFR (bir bit sola/sağa kaydır) komutları ile ROL ve ROR (sola/sağa döndür) komutları akümülatör içeriğini kaydırma ya da döndürme işlemlerini Elde (carry) biti aracılığıyla yapar. SXM bitinin değeri SFR (akümülatörü sağa ötele) komutunun tanımını değiştirir. SXM = 1 olduğunda , SFR akümülatörün işaretini korumak suretiyle aritmetik öteleme ve SXM = 0 olduğunda ise LSB'leri yok edip MSB'lerin yerlerini 0'larla doldurarak lojik öteleme işlemi yapar. Ancak SFL (akümülatör içeriğini sola ötele) komutu SXM bitinin durumundan etkilenmez. Akümülatör içeriğini birden fazla sola ya da sağa ötelemek/döndürmek için ise RPT(repeat) komutundan faydalanılabilir.

5.1.2.4 Yardımcı Kütük ve Yardımcı Kütük Aritmetik Ünitesi

TMS320C240'ın yapısında 8 adet AR0 – AR7 şeklinde adlandırılan kütük bulunmaktadır. Bu kütük kümesi yardımcı ile dolaylı adresleme yapılabilir ve gerektiği takdirde veri saklanabilir. Dolaylı adreslemede, bu kütüklere işlem yapılacak verinin adresi girilir. İlgili kütük ARP (Auxiliary Register Pointer) adı verilen 3 bitlik yardımcı kütük işaretleyicisi aracılığıyla seçilir. Yardımcı kütükler ve ARP'ler veri hafızasından, akümülatörden, sonuç kütüğünden (PREG) ya da komut satırından girilen ani bir operatörle yüklenebilir. Bu iki kütüğün içerikleri ayrıca veri hafızasına depolanabilir ya da CALU tarafından giriş olarak kullanılabilir.

Yardımcı kütük dosyası (AR0 – AR7), yardımcı kütük aritmetik ünitesine (ARAU) bağlıdır.

MİB veri hafıza yerleşimi ile adresleniyor iken, ARAU, ilgili AR'yi istediği gibi değiştirebilir. Yani ya, +1 , -1'le veya indeks kaydedicinin içeriğiyle değiştirme işlemi yapar. Sonuç olarak tablo işlemleri yapılırken adres güncelleştirmenin yapılabilmesi için CALU'ya ihtiyaç yoktur.

Yardımcı kütük Aritmetik ünitesinin yapmış olduğu işlemleri şu şekilde sıralayabiliriz.

- 1- İlgili AR + INDX \Rightarrow İlgili AR
- 2- İlgili AR – INDX \Rightarrow İlgili AR
- 3- İlgili AR + 1 \Rightarrow İlgili AR
- 4- İlgili AR - 1 \Rightarrow İlgili AR
- 5- İlgili AR \Rightarrow İlgili AR

- 6- İlgili AR + 8 bitlik değer \Rightarrow İlgili AR
- 7- İlgili AR - 8 bitlik değer \Rightarrow İlgili AR
- 8- İlgili AR + rc(INDX) \Rightarrow İlgili AR
- 9- İlgili AR - rc(INDX) \Rightarrow İlgili AR

Ayrıca ilgili AR değerinin ARCR (Auxiliary Register Compare register) ile karşılaştırılması ile TC (Test/Kontrol Bayrağı) değerinin değiştirilmesidir. (Küçükdemiral, 1999)

5.1.3 Dahili Hafıza

TMS320C240 yongası iki çeşit hafıza birimi ile yapılandırılmıştır. Bunlardan biri iki çeşit (veri/program) işlem yapabilen RAM (DARAM) ve diğeri ise Maskelenmiş ROM'dur.

5.1.3.1 Veri/Program İki Çeşit İşlem Yapabilen RAM (DARAM)

TMS320C240 yongası üzerinde 16 bitlik 544 kelimeden oluşan ve hem program hem de veri hafızası olarak kullanılabilen RAM (DARAM – Dual Access RAM) bulunmaktadır. Bu hafıza birimi üzerinden aynı makine çevriminde hem okuma hem de yazma yapılabilir. DARAM, 3 adet ayrı ayrı seçilebilen hafıza bloğundan oluşmuştur.

B0 : 256 kelimelik veri ya da program DARAM bloğu

B1 : 256 kelimelik veri DARAM bloğu

B2 : 32 kelimelik veri DARAM bloğu

DARAM bloklarının asıl amacı veri kümelerinin saklanması olduğu halde; gereken durumlarda program saklanması amacıyla da kullanılabilirler. SETC CNF ve CLRC CNF komutları ile hafıza üzerinde (B0 veri hafızası olarak kullanıldığında) istenilen birimin içeriğinin değiştirilmesi sağlanabilir. B0, veri hafızası olarak kullanıldığında komutlar harici program hafızasından bu alana yüklenip daha sonra icra edilebilir. Yukarıda da belirtildiği gibi, B0 bloğu veri ve program bilgilerinin saklanması için kullanılabilirken , B1 ve B2 blokları sadece veri bilgilerinin saklanması amacıyla kullanılmaktadır. DARAM aynı zamanda yonganın işlem yapma hızını da arttırmaktadır.

5.1.3.2 Sadece Okunabilir Program Hafızası (ROM)

TMS320C240 DSP'si üzerinde 16K kelimelik maskelenebilir ve programlanabilir sadece okunabilir (ROM) program hafıza alanı bulunmaktadır. Kullanıcı $\overline{MP/MC}$ kontrol girişini

kullanmak suretiyle bu hafıza alanını iptal edebilir ya da kullanıma açabilir. Bu alan kullanıldığında program hafızasının en alt bloğunu teşkil eder bir başka deyişle küçük adres numarasına sahiptir. ROM hafızası iptal edildiğinde ise onu adreslemek için kullanılan adresler harici hafızanın adreslenmesi için kullanılır.

5.1.4 TMS320C240'ın Yapısında Dahili Olarak Bulunan Çevresel Birimler

Harici hafıza arayüzü

İki adet Analog - Dijital çevirici

Seri çevresel arayüz (SPI)

Seri haberleşme arayüzü (SCI)

Watchdog timer (WD)

Durum yöneticisi (EVENT MANAGER, EV)

5.1.4.1 Harici Hafıza Arayüzü

TMS320C240 yongası 16 bit genişliğindeki 64K kelimelik bir alanı adresleyebilir. Dahili hafıza kullanıldığında bu sayı biraz daha düşer. Veri hafızasındaki yüksek değerli 32K kelime GREG kütüğünü kullanmak suretiyle işaretlenebilir. Bir veri hafızasına ancak \overline{BR} low olduğunda erişilebilir ve hafıza alanı üzerinde işaretlemeler yapılabilir. MİB aynı makine çevriminde dahili hafızadan veri okuma, veri yazma ve program alma işlemlerini yapabilir. Harici arayüz, dahili bus'ı bir adres ve bir veri yolu olmak üzere çoklama işlemi yaparak her ikisi için de kullanılabilmesini sağlar. Bunun için arayüz öncelikle veriyi yazar , sonra veriyi okur ve son olarak da programı okur. TMS320C240 geniş bir arayüz desteğine sahiptir. Program, veri, G/Ç adresleme alanları hafıza ve G/Ç portları kullanılarak haberleşilen harici birimler için bir arayüz sağlar. 16 bit genişliğindeki adres ve veri yolu, \overline{PS} , \overline{DS} ve \overline{IS} alan seçim sinyalleri ile birlikte kullanılırsa 16 bitlik 64K kelimelik bir program ya da G/Ç alanı adresleyebilir. G/Ç üniteleri, işlemcinin harici adres ve veri yolunu kullanmak suretiyle G/Ç adres alanına yazmak için kullanılabilir. Bu sırada bazı kontrol ve gösterge bitlerinin kullanılması hata olasılığını en aza indirir. Örneğin R/\overline{W} çıkış sinyali o andaki çevrimde okuma işleminin mi yoksa yazma işleminin mi yapıldığını belirtir. Ayrıca \overline{STRB} çıkış işareti de harici elemanın çevrim süresi için bir referans teşkil eder ve senkronizasyonu sağlar. READY girişi kullanılarak değişik hızlardaki hafıza ve G/Ç birimleri ile çalışma sağlanabilir. Yongadan

daha yavaş bir harici birim ile çalışılması gerektiğinde MİB , harici birim işini bitirene kadar bekler. Bunu da READY girişinin durumuna göre anlar. Yol isteği (Bus Request) anlamına gelen \overline{BR} işareti diğer arayüz işaretlerinin global hafızaya erişimi sırasında oluşabilecek hataları ortadan kaldırmak için kullanılır. Bir harici global hafıza birimi \overline{BR} sinyalini aldığı anda hafıza erişimi düzenlendikten ve erişim tamamlandıktan sonra hazır sinyalini göndererek cevap verir.

5.1.4.2 Durum Yöneticisi Modülü

Bu modül genel amaçlı zamanlayıcıları, tam karşılaştırıcı birimini örnekleme birimini ve dörtlü encoder (QEP) devresini içerir. Her bir birim daha aşağıda ayrı ayrı incelenecektir..

5.1.4.2.1 Genel Amaçlı (GP) Zamanlayıcılar

TMS320C240 yongasında 3 adet genel amaçlı zamanlayıcı bulunmaktadır. Bunlar aşağıdaki özelliklere sahiptir:

16 bitlik aşağı/yukarı sayıcı. Yazma ve okuma işlemleri için TxCNT (x = 1,2,3)

16 bitlik zamanlayıcı – karşılaştırma kütüğü. Okuma ve yazma işlemleri için TxCMPR

16 bitlik zamanlayıcı – peryot kütüğü. Okuma ve yazma işlemleri için TxCON

Dahili ya da harici clock kaynağını seçebilme

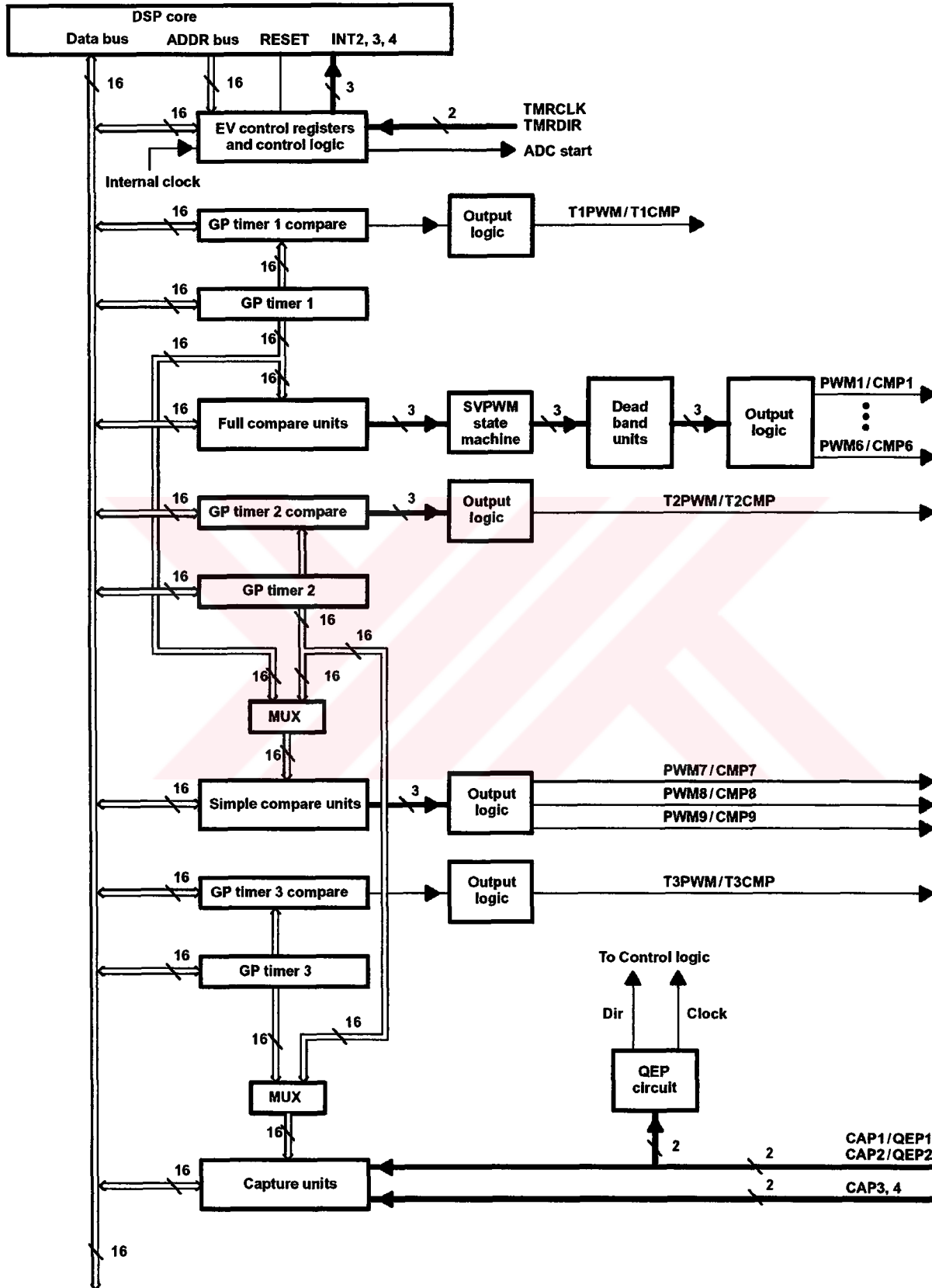
Dahili ve harici clock için ölçekleyici

4 adet maskelenebilir kesme

Aktif Low ya da aktif High olarak ayarlanabilen bir zamanlayıcı – karşılaştırma çıkış bacağı

Bir yön seçim (aşağı/yukarı) kontrol bacağı TMRDIR

Yukarıda özellikleri belirtilen zamanlayıcılar birbirleri ile senkronize olarak çalışabilecekleri gibi bağımsız olarak da çalışabilirler. Ayrıca iki 16 bitlik zamanlayıcı birlikte kullanılarak 32 bitlik bir zamanlayıcı elde edilebilir. Genel amaçlı zamanlayıcıların yukarı ya da aşağı sayma işlemleri iki adet single (tek) ve üç adet sürekli çalışma modunda icra edilebilir. Ayrıca bu zamanlayıcılar dahili ve harici saat darbelerini bir programlanabilir bölücü üzerinden kullanabilir. Her zamanlayıcı/karşılaştırıcı çıkışı GPTCON (genel amaçlı zamanlayıcı kütüğü) kütüğü yardımıyla kontrol edilebilir. Zamanlayıcıların görevlerinden biri de durum yöneticisi modülleri için senkronizasyonu (zaman referansını) sağlamaktır.



Şekil 5.6 Durum yöneticisinin (Event Manager) blok diyagramı

5.1.4.2 Tam Karşılaştırma Birimleri

TMS320C240 yongası 3 adet karşılaştırıcı birimine sahiptir. Bu karşılaştırma birimleri zamanlayıcı 1 'i referans olarak kabul eder ve programlanabilir DeadBand (ölüband) devresini de kullanarak 6 adet PWM çıkışı üretir. Bahsedilen çıkışlar birbirlerinden bağımsız olarak da yapılandırılabilir.

5.1.4.3 Programlanabilir DeadBand (ÖlüBand) Üreteci

ÖlüBand kontrol edilen fonksiyonda otomatik kontrol çalışmaksızın izin verilen tolerans anlamına gelir. Bu devre 3 adet 8 bitlik sayıcı ve 8 bitlik karşılaştırıcı kütüğünü içerir. Üç karşılaştırma biriminin çıkışları için istenilen ölüband süresi karşılaştırma kütükleri kullanılmak suretiyle programlanabilir. Ölüband süresi 0 ile 102 μ s arasındaki bir değer seçilebilir. Ayrıca her bir karşılaştırıcı çıkışı için bireysel olarak ÖlüBand aktif hale getirilebilir ya da iptal edilebilir. ÖlüBand üreteci her karşılaştırma birimi çıkışı için iki adet işaret oluşturur. ÖlüBand üreticinin çıkış durumu ACTR kütüğü kullanılmak suretiyle değiştirilebilir ya da yeniden yapılandırılabilir.

5.1.4.4 Basit Karşılaştırıcılar

TMS320C240 yongası 3 adet basit karşılaştırıcı birimini yapısında bulundurur. Bu 3 birim ek bağımsız karşılaştıma ya da yüksek duyarlıklı PWM dalga şekli üretmek için kullanılır. Basit karşılaştırıcılar için genel amaçlı zamanlayıcı 1 ya da 2 referans olarak seçilebilir. Bu karşılaştırıcı birimlerinin çıkış durumları (çift tamponlanmış – double buffered) SACTR kütüğü kullanılmak suretiyle değiştirilebilir ya da yeniden yapılandırılabilir.

5.1.4.5 Karşılaştırma / PWM Dalga Şekli Üretimi

TMS320C240 yongası 12 adete kadar karşılaştırma / PWM dalga şeklini eşzamalı olarak üretebilir. Bunlar; 3 tam karşılaştırma biriminde bulunan (3 adet) bağımsız karşılaştırıcı çiftler (6 adet çıkış), basit karşılaştırma biriminde bulunan (3 adet) bağımsız karşılaştırıcılar ve genel amaçlı zamanlayıcı vasıtasıyla elde edilen (3 adet) bağımsız çıkışlardır.

5.1.4.5.1 Karşılaştırma / PWM Dalga Şekli Üretme Biriminin Karakteristikleri

16 bit ve 50ns'lik çözünürlük

PWM çıkış çiftleri için 0 ile 102 μ s arasında programlanabilir ÖlüBand zamanı

50ns'lik minimum ölü zaman genişliği

PWM frekans modülasyonu kullanıldığında değiştirilebilir PWM taşıyıcı frekansı

Gerektiğinde her PWM peryodundan sonra değiştirilebilen PWM darbe genişlikleri

Harici olarak maskelenebilir güç ve sürücü koruma kesmeleri

Uzay PWM vektör dalga şekillerini üretmek için programlanabilir şablon darbe üretici devresi

Karşılaştırma ve peryot kütüklerinin otomatik olarak yeniden yüklenmesi ile daha da azaltılmış MİB kullanımı

5.1.4.6 Örneklem Birimi :

Örneklem birimi değişik olayları ve değişimleri kayıt etmek için kullanılır. Örneklem giriş bacaklarında (CAP x , x = 1,2,3,4) daha önce belirlenen değişimler saptandığında genel amaçlı zamanlayıcı (sayıcı) 2 ve/veya genel amaçlı zamanlayıcı 3 değerleri örneklenir ve iki seviyeli FIFO (birinci giriş birinci çıkış) yığınlara kaydedilir. TMS320C240'ın örneklem birimi 4 adet örneklem devresinden oluşur. Bu birimler aşağıdaki özelliklere sahiptir.

Okuma ve yazma işlemleri için bir adet 16 bitlik CAPCON adı verilen örneklem kontrol kütüğü

Bir adet yüksek değerli 8 biti sadece okuma işlemlerine ve düşük değerli 8 bitinin sadece yazma işlemlerine ayrıldığı ve CAPFIFO olarak adlandırılan örneklem FIFO kütüğü

İki adet 16 bitlik çoklayıcı vasıtasıyla isteğe bağlı olarak genel amaçlı zamanlayıcı 2 ya da 3 'ün seçimi. Çoklayıcılardan biri 3 ve 4 numaralı örneklem birimi için kullanılan genel amaçlı zamanlayıcıyı referansı olarak seçer, diğeri ise 1 ve 2 numaralı örneklem birimi için kullanılan zamanlayıcıyı referans olarak alır.

4 adet 16 bitlik x 2 FIFO yığın kütüğü. Her örneklem birimi için bir adet iki seviyeli FIFO yığın kütüğü vardır. Her yığının en üst kütüğü yalnız okunabilen bir kaydedicidir. FIFOx , x = 1,2,3,4

4 Schmitt-trigger'li ve her örneklem birimine bir adet düşecek şekilde örneklem giriş bacakları CAPx , x = 1,2,3,4

CAP1 ve CAP2 giriş bacaklarının aynı zamanda QEP (4'lü kod çözücü darbe) devresi için de kullanılabilme özelliği

Giriş bacaklarında kullanıcının tanımlayabildiği kenar tetikleme (yükselen ya da düşen kenar)

modları

4 adet maskelenebilir kesme/bayraklar, CAPINT_x , x = 1,2,3,4

5.1.4.7 Kod Çözücü Darbe Devresi

İki örnekleme birimi (CAP1 ve CAP2), dahili QEP (quadratura-encoder pulse circuit) devresi için kod çözücü darbe ile birlikte arayüz olarak kullanılabilir. Bu girişlerin tam senkronizasyonu dahili olarak gerçekleştirilir. Genel amaçlı zamanlayıcının sayma yönü bu iki geçiş işaretinin yükselen ve düşen kenarı ile belirlenir.

5.1.4.8 Analog Dijital Dönüştürücü Modülü (ADC)

TMS320C240 yongasının içeriği ADC modülünün basitleştirilmiş blok diyagramı aşağıda verilmiştir. Bu modül iki adet 10 bitlik ADC ile iki adet örnekle – tut (sample – hold) devresinden oluşur. TMS320C240 yongasında 16 analog giriş kanalı mevcuttur. 8:1 analog çoklayıcı sayesinde her ADC 8 analog girişi dijitale dönüştürebilir. Her ADC birimi için minimum dönüştürme süresi 6,1 µs'dir. Ayrıca dönüştürücünün doğruluğu ±1,5 LSB'dir. ADC modüllerinin ihtiyacı olan referans gerilimi harici olarak V_{REFHI} ve V_{REFLO} bacakları aracılığıyla sağlanır. Dijital sonuç aşağıdaki gibidir.

$$\text{Dijital Sonuç} = 1023 \times \frac{\text{Giriş Gerilimi}}{V_{\text{REFHI}} - V_{\text{REFLO}}} \quad [5.1]$$

TMS320C240 yongasının sahip olduğu ADC modülü aşağıda belirtilen özelliklere sahiptir.

Her bir ADC birimi aynı anda iki girişi örneklebilir ve dijital işarete dönüştürebilir.

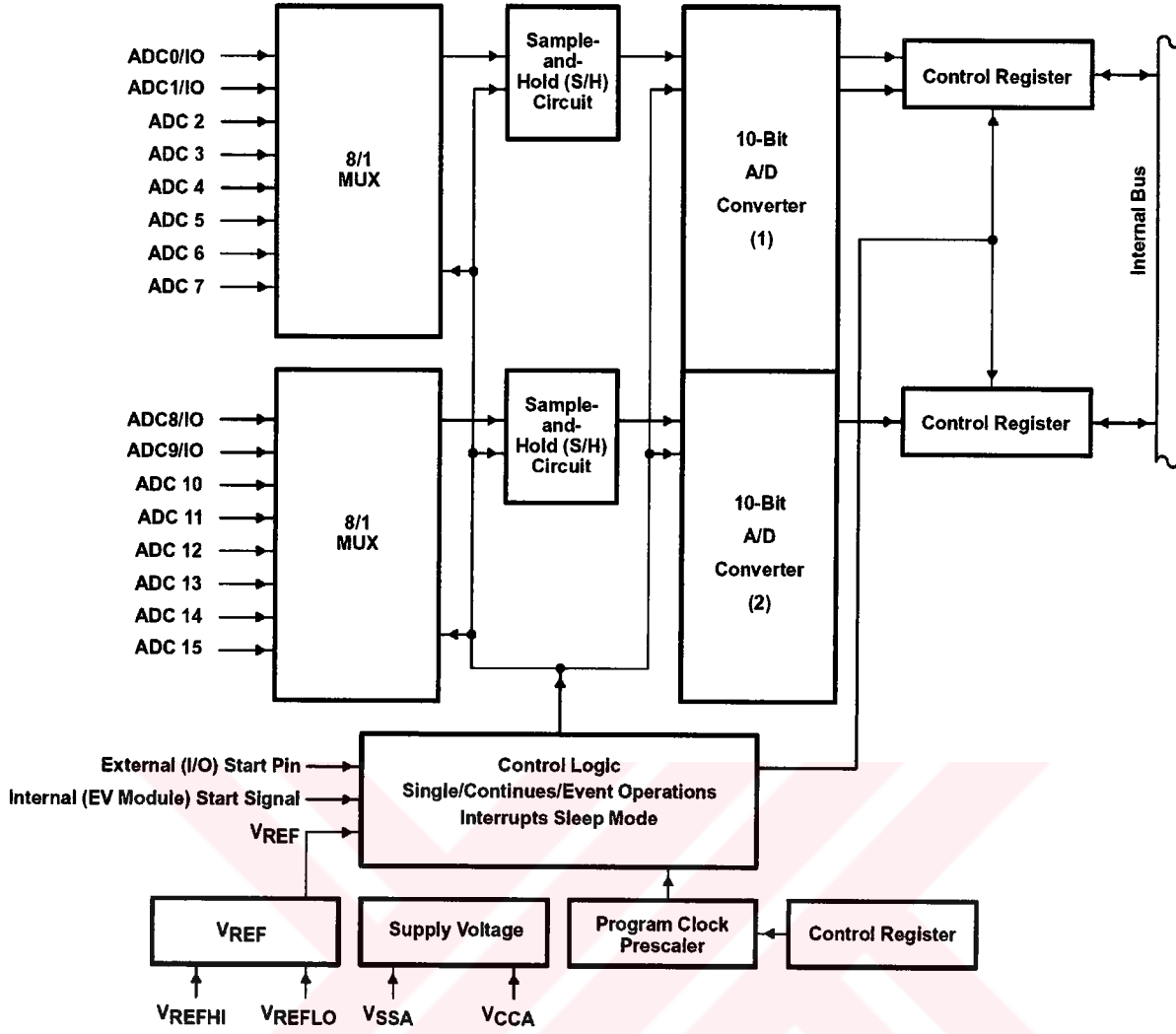
Her ADC birimi tek ya da sürekli S/H (örnekle-tut) ve dönüşüm işlemlerini yapabilir.

İki adet 2 seviyeli FIFO sonuç kütüğünü içerir.

ADC modülü yazılımla, yonga bacaklarından birinden uygulanan işaret ile, örnekleme 4 bacağı aracılığıyla işlem yapmaya başlayabilir.

ADC kontrol kütüğü (double- buffered) çift tamponlanmıştır. Yeni bir dönüşüm bir önceki dönüşüm işlemi tamamlandığında derhal başlayabilir ya da kontrol kütüğünün durumuna göre bekleyebilir.

Eğer kesmeler kaldırılmamış ya da maskelenmemiş ise her dönüşüm prosesinin sonunda bir kesme üretilir.



Şekil 5.7 : Analog dijital dönüştürücü modülü

5.1.5 Seri Çevresel Arayüz Modülü (SPI)

TMS320C240 yongası dört bacaklı seri çevresel arayüz (SPI) modülüne sahiptir. SPI modülü yüksek hızlı senkron bir seri G/Ç portudur. Bu port daha önceden belirlenmiş boyuttaki seri bitin cihaz içine ya da cihazdan çevre birimlere kontrol edilebilir bir hızda akışına izin verir. SPI portu genellikle DSP'nin harici birimlerle ya da diğer bir işlemci ile haberleşmesi sırasında kullanılır. Tipik kullanım alanları olarak ; harici G/Ç , çevresel genişleme (shift register'lar ,display sürücüler) ya da başka bir işlemci (örneğin fuzzy işlemciler) ile birlikte çalışma sayılabilir. SPI'nın ana/yardımcı çalışma modları ile birden fazla DSP yonga ile birlikte çalışılabilir. Bu modül aşağıda belirtilen özelliklere sahiptir.

4 adet harici uç :

SPISOMI : SPI yardımcı çıkış / ana giriş bacağı ya da genel amaçlı iki yönlü G/Ç bacağı

SPISIMO : SPI yardımcı giriş/ ana çıkış bacağı ya da genel amaçlı iki yönlü G/Ç bacağı

SPISTE : SPI yardımcı aktarım yetki verme bacağı ya da genel amaçlı iki yönlü G/Ç bacağı

SPICLK : SPI seri – saat bacağı ya da genel amaçlı iki yönlü G/Ç bacağı

İki işlem modu : Ana ve yardımcı

Bilgi gönderim hızı (Baud) : 125 farklı programlanabilir hız / 10 MHz'lik SYSCLK'da 2,5Mbps

Bilgi kelime biçimi : Birden sekize kadar değişebilen bilgi bit genişliği

4 değişik saat tutma düzeni, saat polaritesi ile kontrol edilir ve saat durum bitleri aşağıdaki özellikleri içerir.

Faz gecikmesiz düşen kenar : SPICLK aktif high'dır. SPI veriyi SPICLK işaretinin düşen kenarında iletir ve aynı işaretin yükselen kenarında da veriyi alır.

Faz gecikmeli düşen kenar : SPICLK aktif high'dır. SPI veriyi SPICLK işaretinin düşen kenarından bir yarı çevrim süresi kadar önce iletir ve işaretin düşen kenarında da alır.

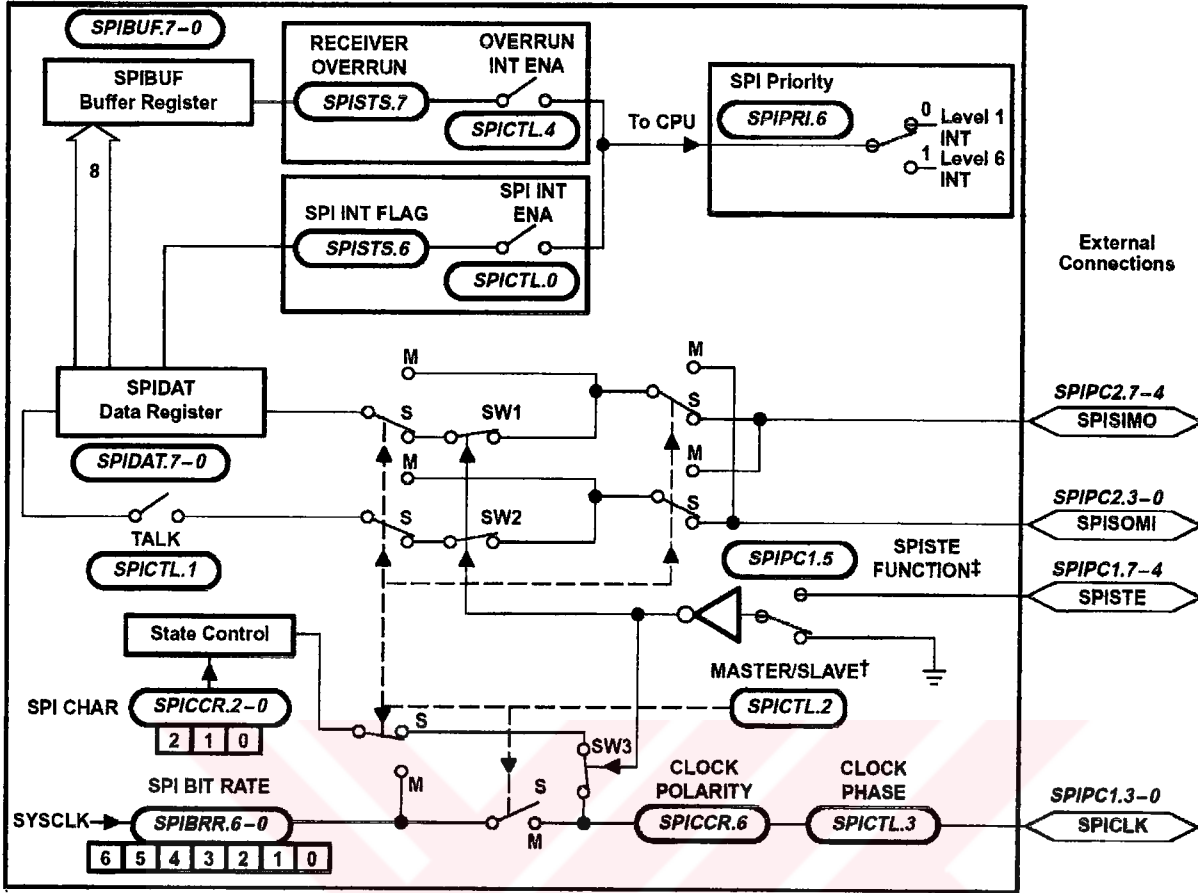
Faz gecikmesiz yükselen kenar : SPICLK aktif olmayan (inactive) low'dur. SPI veriyi SPICLK işaretinin yükselen kenarında iletir ve aynı işaretin düşen kenarında da yeni veriyi alır.

Faz gecikmeli yükselen kenar : SPICLK aktif olmayan low'dur. SPI veriyi iletmeye SPICLK işaretinin düşen kenarından yarım çevrim süresi kadar önce başlar ve aynı işaretin yükselen kenarında da veriyi iletir.

Eşzamanlı iletme ve alma işlemleri yapabilme

Kontrol kütüğü yapısında 7040h adresinden başlamak üzere konumlandırılmış olan 10 ayrı SPI kontrol kütüğü

Aşağıdaki şekilde SPI'nin yardımcı çalışma modundaki blok diyagramı verilmiştir.



Şekil 5.8 : 4 bacaklı Seri çevresel arayüz birimi blok diyagramı

Yukarıdaki şekilde SPISTE bacağı etkisizdir. Bu nedenle bu modda çalışılırken veri iletimi mümkün değildir. Ayrıca SW1 , SW2 ve SW3'ün kapalı olduğuna dikkat ediniz.

5.1.6 Seri Haberleşme Arayüz Modülü (SCI)

TMS320C240 yongası bir seri haberleşme arayüz modülünü dahili olarak yapısında bulundurmaz. SCI modülü MIB ile diğer asenkron çevresel birimler arasında standart non-return-to-zero (NRZ) formatını kullanarak dijital haberleşmeyi sağlar. SCI alıcısının ve ileticisinin her biri bağımsız yetkilendirme ve kesme bitlerine sahip iki tamponu vardır. Bunların her ikisi de full-dupleks modda bağımsız ya da eşzamanlı olarak çalışabilir. Bit hızı, 16 bitlik Baud seçim kütüğü yardımıyla yaklaşık 65000 kadar farklı değere programlanabilir. SCI modülü aşağıdaki özelliklere sahiptir.

İki harici bacak :

SCITXD : SCI iletim çıkış bacağı ya da genel amaçlı iki yönlü G/Ç bacağı

SCIRXD : SCI alım- giriş ya da genel amaçlı iki yönlü G/Ç bacağı

10 MHz'lik SYSCLK darbesinde maksimum 625Kbps hızına erişebilen ve 64K değişik değere programlanabilen Baud hızı

Veri kelime biçimleri : bir başlangıç biti , veri kelime uzunluğunun 1 ile 8 arasında istenilen değere ayarlanabilme özelliği. İsteğe bağlı olarak parite kontrolü (tek, çift ya da hiç) biti , bir ya da iki durdurma biti

4 adet hata tespit bayrağı : Parite , Overrun , framing ve break

Tam ya da yarım dubleks çalışabilme

Çift ara bellekli iletim ve alım işlevleri

İletim ve alım işlemlerini, kesilme sürümlü algoritmalarında dahi durum bayrakları yardımıyla hatasız olarak tamamlayabilme

İletici : TXRDY (İletici –tampon kütüğü yeni bir veriyi iletmek için hazır) bayrağı ve TX EMPTY (iletim –kayıdırma kütüğü hazır) bayrağı

Alıcı : RXRDY (alıcı tampon kütüğü yeni bir veriyi almak için hazır) bayrağı ve RX ERROR (4 kesme durumu için görüntüleme) bayrağı

BRKDT dışında iletilen ve alıcı için ayrı ayrı kesme yetkilendirme bitleri

NRZ (non return to zero) biçimi

7050h adresinden başlama üzere konumlandırılmış 11 SCI modülü kontrol kütüğü

Aşağıdaki şekilde SCI modülüne ait blok diyagram verilmiştir.

Bir adet WD zamanlayıcısının bir sistem reseti üretmek üzere olduğunu belirten WD bayrağı (WD FLAG)

WD'un kontrol kütüğüne yanlış değer girildiğinde bir sistem reseti başlatacağını gösteren bitleri kontrol etme özelliği. (WDCR)

Sisteme reset atıldıktan hemen sonra otomatik olarak devreye giren WD zamanlayıcısı

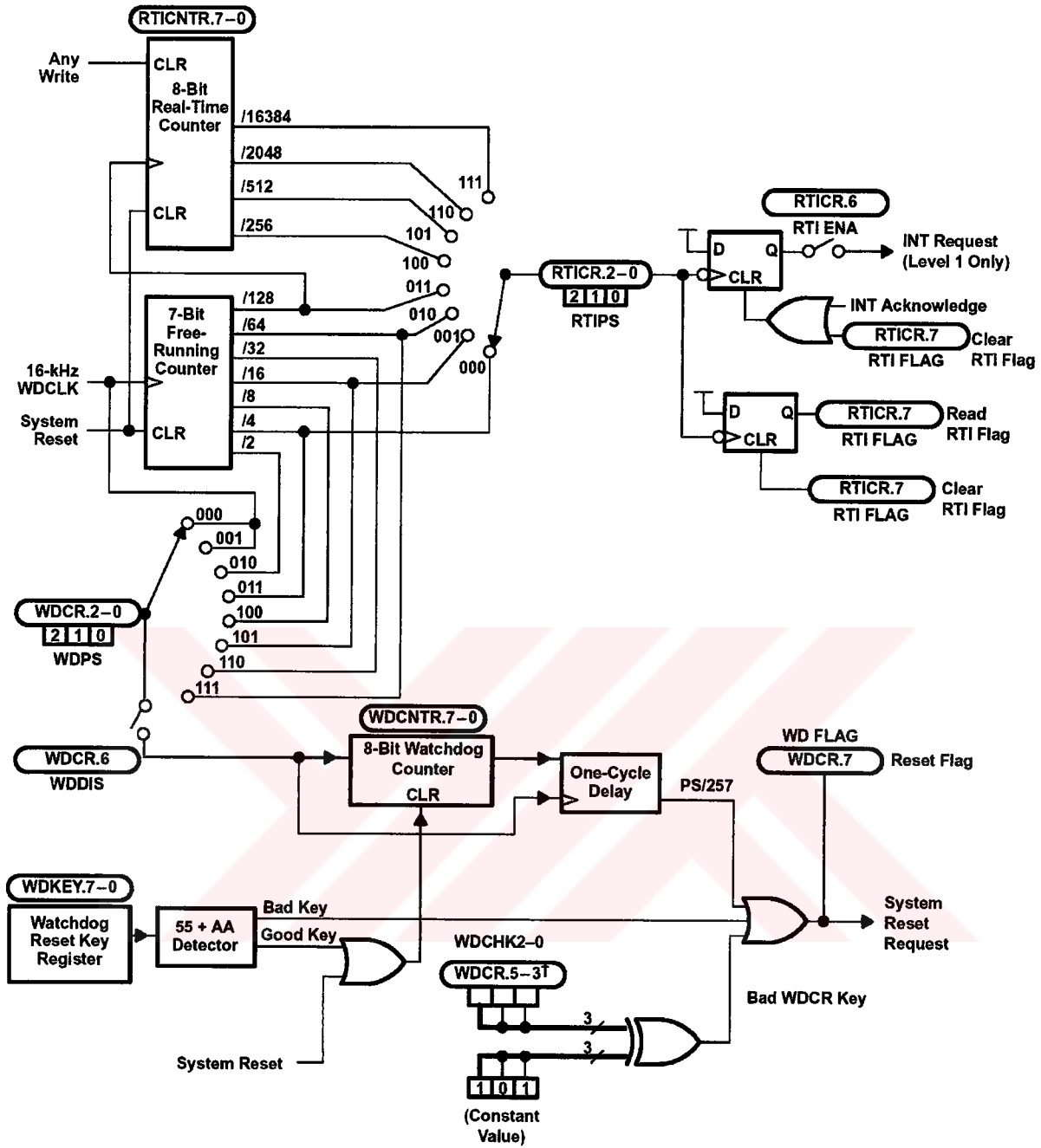
3 adet WD kontrol kütüğü, ana kontrol kütüğü yapısının 7020h adresinden itibaren başlayacak şekilde yerleştirilmiştir.

Eşzamanlı kesme (RTI) biriminin özellikleri ise,

Kesme üretimi saniyede 1 ile 4096 arasındaki programlanabilir bir frekansla gerçekleştirilebilir.

İki RTI kontrol kütüğü ana kontrol kütüğü yapısının 7020h adresinden itibaren başlayacak şekilde yerleştirilmiştir.

Aşağıdaki şekilde WD/RTI modülünün blok diyagramı verilmiştir.



Şekil 5.10 : WD/RTI modülü blok diyagramı

6. DSP ve MİKROİŞLEMCİLERİN KARŞILAŞTIRILMASI

DSP ve mikroişlemcilerin karşılaştırılmasında, uygulamada kullanıldıkları sistemin özellikleri dikkate alınmalıdır. İki sistemin de kendine göre üstün özellikleri olduğu açıktır. Motor kontrolu uygulamaları için sistemler karşılaştırıldığında DSP hızının sağladığı avantaj ile çok daha üstün bir performans göstermektedir. Bunun yanında çevre birimleri ve bellek kapasitesi olarak mikroişlemci daha üstün konumdadır.

DSP ve mikroişlemcilerin mimari yapıdan farklı olmaları, herhangi bir işlemi yapma biçimlerinden gözlemlenebilir. Bir probleme işlemci ile çözüm bulunduğunda, bu çözümün ne kadar kolay programlanabileceği ve ne kadar kısa sürede işleyebileceği mimari yapının özellikleri ile belirlenecektir.

Mikroişlemcilerin ilk olarak göze çarpan mimari özellikleri; basit yapıları, geniş adresleme alanları, adresleme modlarının çeşitliliği ve geniş komut takımıdır. DSP 'lerin mimari özellikleri ise, küçük adres alanları, sınırlı adresleme modları, özel komut takımı, ayrı veri ve program yolları ve komutların tek saat çevrimi süresinde işlenebilmesidir.

Mikroişlemciler Von Neumann mimari yapısında işlemcilerdir. Bu yapının özelliği, veri ve adres iletimi için ayrı yolları kullanması ile komut ve veri belleklerinin ortak olmasıdır. Bundan dolayı, veri ve komutlar ancak bir sıra dahilinde işlenebilir. İkisinin aynı anda işlenmesi mümkün değildir. Bu durum, veri sıkışması olarak adlandırılan sorunun oluşmasına neden olacaktır. Veri ve adres için aynı yolların kullanılması, sistemin hızının düşmesine neden olmaktadır. Çünkü, adres ve verinin aynı anda yollanması mümkün olmamakta, ancak bir zamanlama diyagramını takip edecek şekilde belirli gecikmelerin sağlanması ile adres ve veri artarda gönderilmektedir.

Bu da sistemin hangi adres ile haberleşebileceğini öğrendikten sonra veriyi işlemesi anlamına gelmektedir. DSP 'lerde ise bu şekilde bir sorunla karşılaşılmamaktadır.

DSP 'lerin mimari yapısı Harvard yapısı olarak adlandırılmaktadır. Von Neumann mimarisine rakip olarak 1940 'larda geliştirilen bu yapı ancak VLSI denilen entegrasyon biçimi ile üretime geçilmesi ile tekrar geçerliliğini kazanmıştır. Harvard mimarisinin en önemli özelliği adres ve veri yollarının ayrı ve bağımsız olmasıdır. DSP 'ye üstün hız özelliğini kazandıran bu özelliktir. Komut ve veriler ayrı bellek birimlerinde saklanmaktadır, böylece ulaşılması daha kolay ve daha kısa sürede olmaktadır. Program ile veri belleklerinin ve yollarının ayrı olması, aynı komut saat çevriminde her ikisinin alınıp, işlenebilmesine olanak sağlamaktadır. Hem komutun hem de verinin aynı anda işlenebilmesi sistemin hızının katlamaktadır. Örneğin

Motorola 56000 DSP 'de iki ayrı veri belleği bulunmaktadır ve tek saat çevriminde her ikisi üzerinde işlem yapılabilir. Belleklerin ayrı olması DSP 'lerin her kullanıcı için ayrı bir eleman gibi davranmasına neden olacaktır. Çünkü her kullanıcı belleği kendi uygulamasına özgü yorumlama özgürlüğüne sahiptir. DSP 'lerdeki veri belleği, bir mikroişlemcinin kütük dosyasının genişletilmiş olarak düşünülebilir, çalışması aynı mantık süreci ile olmaktadır. Sadece program ve veri belleklerinin ayrı olması bile sistemin hızını ikiye katlamaktadır.

Çizelge 6.1 : Çeşitli DSP ve mikroişlemcilerin işlem hızlarının karşılaştırılması

	TMS320C14	TMS320C25	80C196	68000	68020	Birim
Saat Çevrimi	160	100	333	400	120	ns
Frekans	25	40	12	10	24	MHz
Çarpma 16*16	0,16	0,1	2,2	7,0	1,0	µs
Matris Çarpımı (3*3(3*1))	4,3	2,7	24,3	65,2	9,5	µs

Mimari yapı olarak mikroişlemcilerin basit olması işleme hızını etkilemektedir. Mikroişlemcilerde bütün işlemler MİB 'nin kontrolü altındadır. İşlemci içindeki bütün alt birimler ile haberleşmeyi ve sayısal işlemleri, MİB gerçekleştirmektedir. MİB içinde, sayısal ve lojik işlemlerin gerçekleşmesini sağlayan ALU ve kaydırma ünitesi gibi birimler olmasına rağmen, bu birimlerin bağımsız olarak çalışması söz konusu değildir. Bu da mikroişlemcilerin hesaplamalar sırasında zaman kaybetmesine neden olmaktadır. DSP 'lerin mimari yapısı, tek bir ana kontrol birimi yerine, birbirinden bağımsız ve aynı anda çalışabilecek ünitelerden oluşmuştur. Bu, DSP 'ye üstün hızlı işlem yapma özelliğini getirmektedir. Sayısal ve lojik işlemleri ele alan üniteler ayrı ayrıdır. MAC ve ALU ünitesi, aynı anda hem lojik hem de sayısal işlemin gerçekleşmesine olanak sağlamaktadır. Kaydırma ünitesi, DSP 'lerde, mikroişlemcilerdeki gibi, tek tek bitlerin kaydırılma işlemi yerine aynı anda birden çok bitin kaydırılmasını da gerçekleştirmektedir. Böylece sistem, tek bir komut ile bir kelimenin gerekli bitlerini kaydırabilmektedir. İkilik sayı düzeninde çalışırken, toplama işleminin sonucunu akümülatörde saklamadan önce aynı saat çevrimi içinde, kaydırma işlemi gerçekleştirilebilir.

Ayrıca, bu ünite, ikilik düzendeki sayıların yuvarlatma işleminin çok daha hızlı olmasını sağlamaktadır. Bu gibi, ünitelere kazandırılan üstün özellikler sayısal işlemlerde DSP 'leri çok daha üstün hale getirmiştir.

Bununla beraber, DSP 'lerde sayısal işlemlerde asıl üstünlüğü sağlayan ünite çarpım ünitesidir. Donanım olarak çarpım veya MAC (Çarpım-Akümülatör) ünitesinin bulunması sayısal işlemlerin hızının katlanmasını sağlamaktadır. Mikroişlemcilerde çarpma işlemi yazılım ile gerçekleştirilmektedir. Bu da birçok toplama ve kaydırma işleminin kullanılması anlamına gelmektedir. DSP 'lerde ise, çarpma işlemi doğrudan donanım ile gerçekleştirilmektedir. Bu da çarpma işleminin birkaç saat çevrimi süresinde tamamlanmasını olanaklı kılmaktadır. Örneğin, Intel firmasının ürettiği 8096 mikroişlemcisi, 16*16 bitlik bir çarpma işlemini 7 μ s sürede gerçekleştirirken, ilk nesil DSP 'de bu işlem 0.16 μ s sürmektedir. Kontrol algoritmalarında, toplama ve çarpma işlemlerinin önemli bir yer tuttuğu göz önüne alındığında DSP 'nin bu konudaki üstünlüğü çok daha açık olarak görülebilir. Mimari yapıdan ve donanım özelliğinden gelen bu hız, DSP 'nin özellikle karmaşık kontrol algoritmalarının kullanıldığı uygulamalardaki yerini sağlamlaştırmıştır. Örneğin, adaptif kontrol, optimal kontrol veya bulanık kontrol ile çalışan sistemlerin matematik işlem yükünün ağırlığından dolayı, mikroişlemci kullanılması zor olmasına rağmen, DSP ile çok daha kolay ve verimli bir çözüm olanağı mümkün olmaktadır.

Matematik işlemler sırasında oluşan hatanın algılanması ve elimine edilmesi DSP 'lerde çok daha verimli olmaktadır. İkilik düzende sabit nokta ile işlem yapan işlemcilerde, taşma olayı olduğunda, pozitif sayı, negatif sayıya dönüştürülmektedir. Normal çalışmada pozitif olması gereken bir sayının, kütükte saklanamayacak kadar büyük olduğunda negatif yapılması, kontrolün alt üst olması anlamına gelmektedir. Bu aşamada yapılması gereken, sistemin doyma moduna girmesi sağlanarak, sayı, kütüğe sığmayacak kadar büyüdüğünde en büyük pozitif veya negatif sayı olmasını sağlamaktır. Bu işlemin mikroişlemcilerde yapılması için, her işlem sonunda taşma kontrolü yapılarak, gerekli görüldüğünde kütüklerin sıfırlanması gerekmektedir. Bu da çok uzun bir süre almaktadır. DSP 'lerde doyma modunda çalışma bulunduğundan, işlem kontrolü kendiliğinden yapılmaktadır. Kütüklerin kontrolü ile ayrıca zaman kaybedilmesine gerek kalmamaktadır.

Mikroişlemcilerin işleyebilecekleri adres alanı, DSP 'lere göre çok daha büyüktür. Bu sayede çok daha geniş programların saklanması ve işlenmesi mümkün olmaktadır. Kontrol algoritmalarında çok kullanılan veri tabloları, bellek alanının yeterince geniş olması nedeniyle hem veri aktarımı hem de sayısal işlemlerden kurtulmak için kullanılmaktadır. Örneğin,

trigonometrik bir fonksiyonun sayısal yöntemler ile hesaplanması yerine bir veri tablosuna değerlerin girilerek, kullanılan değere karşılık fonksiyonun değerinin belirlenmesi, çok daha kolay bir yol olmaktadır. Adresleme modlarının özel uygulamalara cevap verecek kadar çeşitli olması, bu şekildeki çözümlerin uygulanabilirliğini arttırmaktadır. DSP 'lerde ise, bellek alanının ve adresleme modlarının kısıtlı olması sayısal çözümlerin popüler olmasına neden olmuştur. Örneğin, trigonometrik bir fonksiyon için veri tablosu oluşturulması yerine sayısal yöntemlerle hesaplanması çok daha verimli bir çözüm olanağı sunmaktadır. Hem bellekte veri tablosu oluşturularak kullanılan alanı daraltmaktan kaçınmak hem de sistemin hızını arttırmak için, bütün sayısal işlemler mümkün olan en kısa yoldan yapılmalıdır. Matematiksel olarak aynı sonucu vermesine rağmen, aynı işlem, işletim süresini kısaltmak amacıyla çok daha verimli programlanabilir.

DSP 'lerde bellek alanının kısıtlı olması, mikroişlemcilerde birçok işlevi yerine getiren yapıların kısıtlı boyda olmasına neden olmuştur. Örneğin yığın yapısı, mikroişlemcide ana programdan dallanma olduğunda, alt program ve kesme işlemleri ile karşılaşıldığında işlenmekte olan verileri saklayarak, kaybını önlemektedir. DSP 'lerde yığın mikroişlemciye göre çok daha kısıtlı boyuttadır. Donanım yapısı olarak belirlenen yığın, ancak birkaç verinin saklanacağı kadar küçük boyuttadır. İlk nesil DSP 'lerde 4 adres boyunda iken daha sonra 16 adres boyuna kadar çıkmıştır. Mikroişlemcide ise, belleğin belirli bir bölümü, istenilen başlangıç değerinden istenilen boyutta yığın olarak tanımlanabilir. Aynı şekilde kütük yığını DSP 'lerde kısıtlı sayıdadır. Kütük sayısı mikroişlemci ile karşılaştırıldığında küçük olmasına rağmen, DSP 'lerde özel kütükler yardımıyla çok amaçlı kullanım sağlanmaktadır.

Çizelge 6.2 : Çeşitli DSP yongaların bellek ve adres özellikleri

DSP	Veri Alanı	Program Alanı	Veri Kelime Genişliği
TMS320C10	256KW	4KW	16
TMS320C25	64KW	64KW	16
TMS320C50	96KW	64KW	16
TMS320C240	64KW	64KW	16
TMS320C6201	64KW	16KW	32
ADSP- 2101	16KW	16KW	16
M56000	128KW	64KW	24

Mikroişlemcilerde her ortama en verimli şekilde veri iletimi için çok değişik adresleme kodları tanımlanmıştır. Bunun yanında DSP 'lerin adresleme modları kısıtlı sayıdadır. Ancak temel adresleme modları ile çözüm sağlanabilmektedir.

Mikroişlemcilerin komut kümesi, DSP 'ye göre çok daha geniştir. Mikroişlemcilerde komut seti, birçok işlevi en kısa ve en verimli görecektir şekilde kurulmuştur. Bunun yanında, DSP 'nin komut seti mikroişlemci ile karşılaştırıldığında daha az komut içermekte ve daha basit yapıda görünmektedir. Buna rağmen, DSP 'nin mimari yapısından en iyi şekilde yararlanacak komutların varlığı sayesinde komut setinin o kadar geniş olmasına gerek duyulmamıştır, Komutların sayısının fazla olması, kullanıcıya çok fazla kolaylık getirmemektedir. Mikroişlemcilerde komutlar mikrokod olarak işlemektedir. Mikrokod, her komutun ayrı ve sadece mimari yapıya özgü bir yazılım ile işlemesi anlamına gelmektedir. Her komut, işlemci içinde bir mikrokodun çalışmasına neden olmaktadır. Bu, işlemcinin daha ucuz olmasını ve daha kolay işlenmesini sağlamaktadır. Aynı zamanda komut setinde fonksiyonel değişikliklere olanak vermektedir. Bunun yanında komutların işlemesi, daha uzun süre almaktadır. Mikroişlemcilerde komut işleme süresi, işlemciye bağlı olarak, 500 ns ile birkaç yüz mikrosaniye sürmektedir.

DSP 'de komutların işlemesi, tamamıyla lojik bir yapı ile sağlanmaktadır. Komutların işlenmesini donanım özelliği haline getiren bu yapı ile en karmaşık komutların işlemesi bile

birkaç saat çevrimi sürmektedir. Bu, bütün çarpma ve kaydırma işlemleri içinde geçerlidir. Böylece, komut işleme süresi, 25 ns ile 200 ns arasında değişen değerlerde olmaktadır.

DSP 'lerin mikroişlemcilerle göre zayıf olduğu bir alanda çevre birimleri ile sağladığı entegrasyondur. Mikroişlemciler, uygulamaya yönelik, bütün giriş çıkış birimlerini bünyesinde bulundurarak, mikrobilgisayar yapısı haline gelmiştir. Böylece ek donanım devrelerine gerek olmadan doğrudan doğruya sisteme uyum sağlayabilmektedir. Örneğin, A/D çevirici, seri port, paralel port gibi çevre elemanları ile haberleşmeyi ve veri iletimini sağlayan birimlerin tek bir tüm devre içinde toplanması, sistemin çok basit bir donanım yapısı olmasını sağlamaktadır. Motor kontrolüne yönelik mikrobilgisayarlarda enkoderden gelen işaretleri doğrudan değerlendirilecek lojik devrenin ve güç elektroniği elemanlarını sürmek için gereken PWM işaret üreticinin bulunması seçimde tercih sebebi olmaktadır. Bu gibi birimler yardımıyla mikroişlemciler sisteme çok daha kolay ve kısa sürede uyum sağlayabilmektedir.

DSP 'nin çevre birimleri ile donatılması hızını etkileyeceğinden, aynı yapı içinde boyutlandırılmaktan kaçınılmaktadır. Giriş çıkış birimlerinin ek devreler yardımıyla dışarıdan donanım desteği olarak sunulması yoluna gidilmektedir. Tasarlanan işlemcilerde, sınırlı sayıda çevre birimi eklenmiştir. Örneğin, zamanlayıcı birimi, seri port gibi birimlerin eklenmesi kontrol sisteminde kullanılırken daha az donanım yapısı olmasını sağlamaktadır. Örneğin TMS320C14 DSP 'sinde dört adet zamanlayıcı, seri port ve PWM üretici olarak kullanılabilen D/A çevirici yapıları entegre devre içine eklenmiştir. Yalnız kullanımı, mikroişlemcilerde olduğu gibi basit komut ve çevrimler ile verimli olmamaktadır.

Son yıllarda geliştirilen yazılım desteği ile hem mikroişlemciler, hem de DSP 'ler kolayca programlanmaktadır. Yüksek seviyeli dillerin kullanımına olanak sağlayan ortamlarda çalışılması, tasarımcıya büyük kolaylıklar getirmektedir. Bununla beraber, özellikle motor kontrol algoritmalarının programlanması sırasında makine kodunun karşılığı olan Assembler dili tercih edilmelidir. Bu tür uygulamalarda işlem hızı önem kazandığından, makine kodunda program avantaj sağlamaktadır. Yüksek seviyeli bir bilgisayar dilinin kullanılması programın yapısını kolaylaştırırken, programın işlemesi için dilin kendi ortamı gerektiğinden zaman kaybı çok artacaktır. Uygulamada, sistemi gerçekleştirmek için kısıtlı zaman var ise, mikroişlemci tercih edilmelidir. Çünkü, mikroişlemcinin programlanması, DSP ile karşılaştırıldığında, aynı anda sadece bir tek komutun işlemesi mümkün olduğundan daha kolaydır. DSP sisteminin geliştirme aşaması ve verimli bir şekilde programlanması, mikroişlemciden daha uzun bir süre gerektirecektir.

Mikroişlemci ile DSP'nin fiyat olarak karşılaştırılmasında uygulama alanı büyük önem kazanmaktadır. DSP 'nin kurulup geliştirme ve program yazma süresi daha uzundur. Son yıllarda çıkan yazılım desteği ile bu sorun belirli oranda azalsa da maliyeti arttırıcı bir unsur olmaya devam etmektedir. DSP 'ler ilk üretildiklerinde, mikroişlemcilerden çok daha pahalı olmalarına rağmen, şu anda tüm devre fiyatları, birbiriyle özdeş yapılar için çok farklı değildir. Özellikle kayan nokta DSP 'ler sabit noktalarına göre daha pahalıdır. Her uygulamanın DSP ile çözülmesi matematiksel olarak tam bir sonuç verirken, mali bakımdan aşırı bir artış getirecektir. Bununla beraber, örneğin konum kontrolu uygulamalarında, kontrol yapısının yanında çok kapsamlı bir donanım yapısı bulunduğundan, DSP tercih edilmektedir. Gün geçtikçe DSP 'lerin ucuzlaması, kullanım alanlarını arttıracak, bu da sistemin özellikle büyük miktarlar söz konusu olduğunda mikroişlemcilere ucuz bir alternatif olmasını sağlayacaktır.



7. DSP UYGULAMA ALANLARI

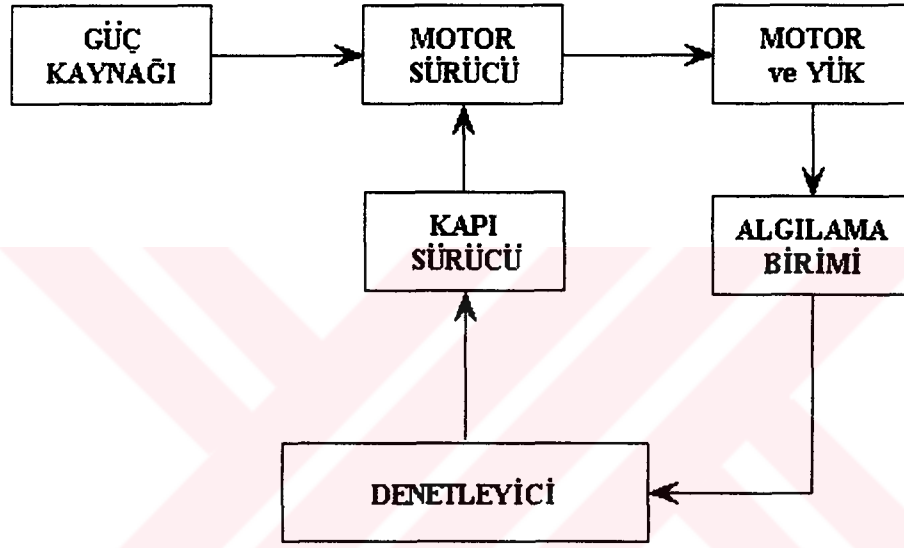
VLSI teknolojisindeki gelişmeler ve son yıllarda Texas Instruments firmasının geliştirmiş olduğu Veloci TI teknolojisi, DSP sistemlerinin çok kısa bir süre içinde kontrol uygulamalarının bütün alanlarına girmesine olanak sağlamıştır. Çok daha küçük alanlara daha fazla işlevi olan yapıların sığdırılması ile tasarım ve uygulama değişik boyutlar kazanmıştır. Özellikle karmaşık algoritmaların ve hızlı işaret işlemenin gerektiği bütün uygulamalarda DSP mikroişlemcilerin yerini almaya başlamışlardır. Mikroişlemciler ile karşılaştırılacak düzeyde bir fiyat seviyesine ulaşıldığında, bütün uygulamalarda kullanılmaya başlayacağı açıktır.

DSP 'nin genel işaret işleme alanında, dijital filtreleme, konvolusyon, korelasyon, Hilbert dönüşümleri, FFT (hızlı fourier dönüşümü), adaptif filtreleme ve sinyal üretimi gibi geniş bir uygulama alanları bulunmaktadır. Grafik ve görüntü işleme alanında; üç boyutlu grafik dönüşümü, görme sistemleri, görüntü analizi, uydu görüntü işleme, görüntü iletimi, görüntü tanıma ve algılama, ölçme alanında; spektrum analizi, fonksiyon generatörü, sismik işaret işleme, geçici olay analizi, dijital görüntü algılama, ses ve konuşma algılama alanında; ses iletimi, ses tanıma, ses algılama, ses sentezi, teksin sese dönüşümü, kontrol alanında; bilgisayar disk sürücü, servo, robot, lazer yazıcı, motor ve taşıt kontrolü, ordu sistemlerinde; radar işleme, sonar işleme, görüntü işleme, radyo frekansı modemler, füze yörünge tespiti, telekomünikasyon alanında; yankı azaltma, modemler, adaptif ekolayzer, faks, telefon ve santral sistemleri, cep telefonu, video konferans, otomobil endüstrisinde; ateşleme kontrolü, titreşim analizi, ABS fren sistemi, adaptif yol kontrolü, yer ve yön tespiti, ses ile kumanda, tıp alanında; işitme cihazları, hasta görüntüleme, ultrason cihazları ve bu alanda kullanılan bütün sistemlerde kullanılmaktadır. Bütün bu örneklerden anlaşılacağı gibi, dijital sistemlerin olduğu her alanda DSP kullanılmaya başlanmıştır.

DSP sistemleri ilk olarak ses ve görüntü işleme ile telekomünikasyon uygulamalarında kullanılmaya başlanmıştır. Bunun nedeni, bu uygulamalarda kontrol edilen sistemin çok hızlı işlemesi ve işlem olarak oldukça yüklü olmasıdır. Uygulamaya özel sistemler, çok daha pahalıya mal olduklarından, DSP sistemleri, genel uygulamalara yönelme açısından bir devrim yaratmıştır. DSP 'nin bu tür uygulamalarda kullanılması, işlem kapasitesini ve çalışma hızını arttırdığından, maliyet artışı dengelenmektedir.

Motor kontrol sistemlerinde DSP 'nin kullanılması diğer uygulama alanları ile karşılaştırıldığında oldukça geç olmuştur. Bunun nedeni, mikroişlemcilerin yeterli düzeyde

bir performans göstermesi ve maliyeti düşürerek, her türlü uygulamaya uyumlu olabilmesidir. Ancak motorların klasik kontrol algoritmaları yerine karmaşık kontrol algoritmaları ile kontrolü söz konusu olduğunda, mikroişlemciler zayıf kalmaktadır. İşlem sayısı arttıkça mikroişlemci yavaşlamakta, bu da kontrol edebileceği sistemlerin performansını etkilemektedir. Bu gibi sorunlara çözüm olarak, mikroişlemcilerin paralel çalışması, kontrol edilen sistemin belirli aşamalarında ayrı işlemcilerin kullanılması yoluna gidilmişse de, hem böyle bir sistemin kontrolünün ve uygulanabilirliğinin zor olması, hem de maliyet artışı DSP 'leri üstün hale getirmiştir.



Şekil 7.1 : Motor kontrol sisteminin birimleri

Motor kontrol sistemini oluşturan üniteler Şekil 7.1'de, blok şema olarak gösterilmiştir. Kontrol edilecek sistemin özelliklerine uygun bir güç elektroniği devresi kullanılmalıdır. Alternatif akım şebekesinden beslenme durumunda güç akışını kontrol etmek amacıyla güç kaynağı katı kullanılabilir. Alternatif gerilim, doğrultucu ile doğrultulduktan sonra, eğer gerekiyorsa, gerilimin genliğini ayarlamak için doğru akım kısıyıcısı kullanılabilir. Daha sonra kullanılan motora özgü güç elektroniği güç katı devresi gelmektedir. Asenkron ve fırçasız doğru akım motor kontrolünde evirici kullanılırken, adım motoru ve anahtarlama relüktans motorunda değişik güç katları uygulanabilir. Güç katının yapısı nasıl olursa olsun, devrede kullanılan güç elektroniği anahtarlarını sürmek için sürücü devreye gerek vardır. Kullanılan anahtarlama elemanları, güç tranzistörü, MOSFET veya IGBT olabilir. Özellikle MOSFET ve IGBT yarı iletken elemanlarının yüksek akım ve gerilim değerlerine ulaşması ve anahtarın iletme ve kesime sokulmasının daha kolay olması kullanım alanını arttırmıştır. Gerilim kontrollü olan bu elemanların yüksek anahtarlama frekanslarına ulaşması, devredeki PWM

işaretinin daha yüksek frekanslarda elde edilmesini sağlamaktadır.

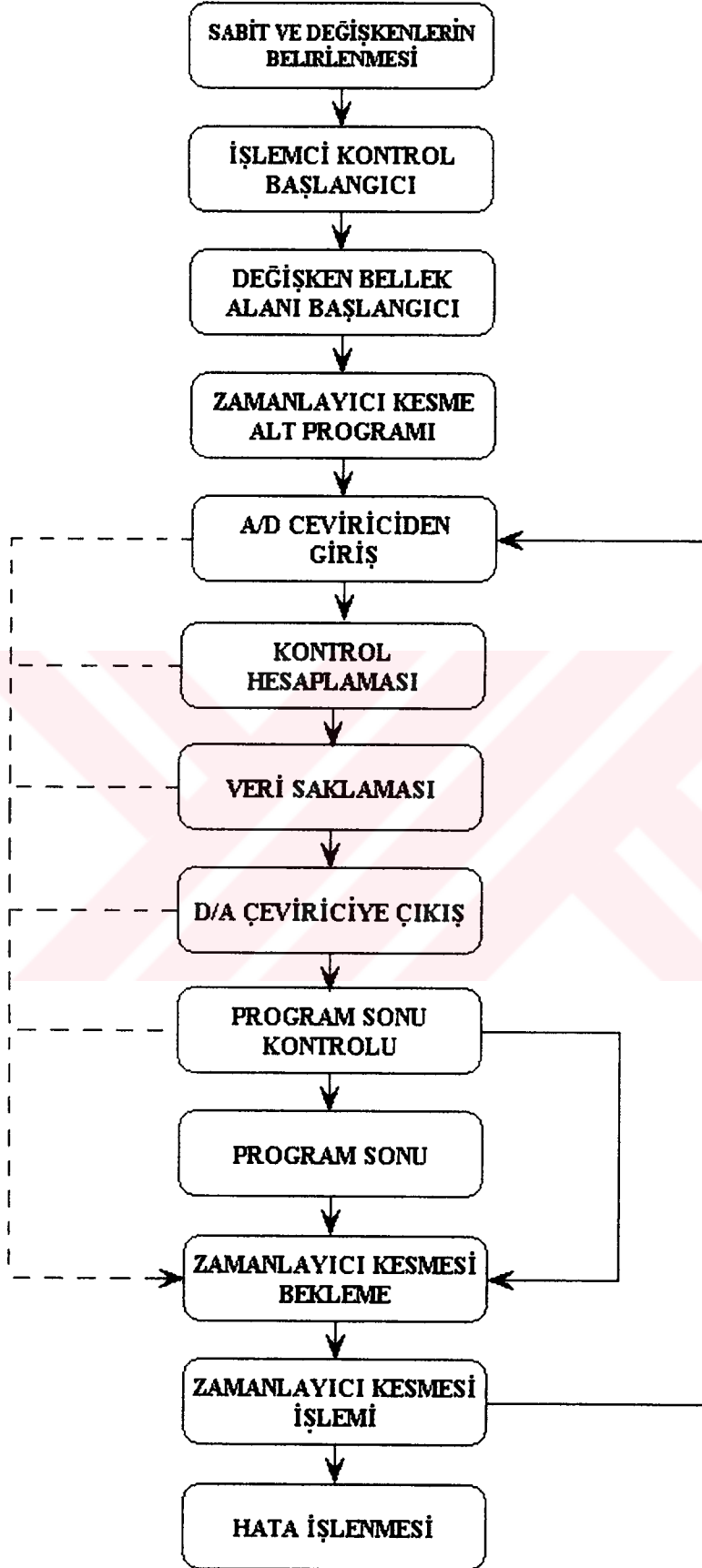
Kontrol sistemindeki bir diğer önemli eleman da algılayıcılardır. Motorda hız veya konum kontrolü yapıldığında, kapalı çevrim bir kontrol için denetleyicinin gerekli olan bilgiyi almasını algılayıcılar sağlamaktadır. Hız ve konum bilgisi analog işaretler olmasına rağmen, denetleyicinin dijital yapıda olması nedeniyle kullanılan algılayıcıların çıkışı dijital seviyededir. Örneğin hız algılamak için, enkoder kullanıldığında, çıkış, motorun hızıyla orantılı frekansta bir işaret olmaktadır.

Denetleyici yapısı ise, sistemden bağımsızdır, sisteme göre değişen sadece işlemcinin programıdır. Bu esnek yapı sayesinde, değişik uygulamalarda donanım yapısında bir değişiklik yapmadan sadece yazılımı değiştirerek kullanılabilir.

Tasarlanan kontrol sisteminin yapısından bağımsız olarak, gerçekleştirilen yazılımın belirli bölümleri aynı kalmaktadır. Motor kontrolünde, sistemin kontrol sistemine göre davranışını incelemek amacıyla, simülasyonu yapılarak, cevap analizi yapılabilir. Bu sayede, gerçek sisteme geçmeden önce, sistemin davranışı saptanmış olacaktır.

Kurulan algoritmanın denetleyici sistemine hangi dil kullanılarak programlanacağı bir diğer tasarım aşamasıdır. DSP yazılım sistemleri, makine kodu yanında C dilini desteklemekte, böylece programın bu dilde yazılıp, işletilmesine olanak sağlamaktadır. Yalnız sistemin hızından tam olarak yararlanma durumu için, gerçek zamanda kontrol uygulamalarında işlemcinin makine dilinde programlanması tercih edilmelidir. Düşük seviyeli diller yardımıyla, işlemcinin bütün mimari üstünlükleri göz önüne alınıp, program gerçekleştirilmelidir.

Akış diyagramı belirlenen program, ilk olarak sistemin sabit ve değişkenlerinin tanımlanması aşamasını gerçeklemelidir. İşlemcinin bellek ve kütük yapıları gereken şekilde tanımlandıktan sonra, zamanlayıcı ve kesme işlemlerine izin verilmelidir. İşlemcinin zamanlama ünitesi gerçek zamandaki işaretleri algılamak ve işlemek amacıyla değişik kesme işlemlerini gerçekleyebilmektedir. Kapalı çevrim motor kontrol sisteminde, algılayıcı ünitesinden gelen işaretler değerlendirilmekte ve gereken işlemler yapılmaktadır. Zamanlama ve kesme işlemleri birer vektör yapısında olup, alt program gibi işlemektedirler.



Şekil 7.2 : Örnek bir kontrol algoritmasına ilişkin akış diyagramı

Sistem veri işleyecek ve saklayacak duruma geldiğinde, A/D çeviriciden gelen işaretler değerlendirilmeye alınmaktadır. Bu veriler bellekte veya kütüklerde saklanarak, kontrol işleminin hesabında kullanılmaktadır. Kullanılan kontrol yöntemine göre, veriler işlenip, gereken kontrol işareti üretildiğinde, D/A çeviriciye yollanarak fiziksel dünya ile gereken dilde anlaşma sağlanmaktadır. Kurulan program yapısı sonsuz döngü şeklinde çalışırken, herhangi bir birimden gelen dallanma emri ile normal çalışmasına ara vermektedir.

Kontrol sistemlerinde, kontrol edilen sistemin davranışını düzenlemek amacıyla denetleyiciler kullanılmaktadır. Klasik kontrol yapısı olarak sistemlerde en çok kullanılan yapı PID yapısıdır. Sistemin tam ve doğru olarak kontrolünü gerçekleştiren bu yapı, istenen kontrol cevabını parametrelerini değiştirerek sağlamaktadır. Yapının bu kadar geniş uygulama alanı bulmasının nedeni, basit bir yapıda olması ve kolaylıkla sisteme uyumlu hale getirilebilmesidir.

$$u(t) = K_c \left(e(t) + \frac{1}{T_i} \int e(s) ds + T_d \frac{de(t)}{dt} \right) \quad [7.1]$$

Gerçek zamanda [7.1] 'deki yapıda olan PID denetleyicide, y çıkış işareti ve y_r referans değeri olmak üzere, $e(t) = y_r - y$ hata terimi, K_c kazanç, T_i entegral zaman sabiti ve T_d türev zaman sabiti olarak tanımlanmaktadır. Sistemde kazanç, işaretin kuvvetlendirilmesini sağlamaktadır. Entegral terimi, düşük frekans kazancını artırarak, kararlı hal hatalarının azalmasını; türev terimi ise, faz kayması oluşturarak, sistemin kararlılığının düzelmesini ve band genişliğinin artmasını sağlamaktadır. PID sisteminin ayrıık zamanda kullanılabilmesi için elemanlarının ayrııklaştırılması gerekmektedir.

Orantı teriminde, gerçek zamandan ayrıık zamana geçilirken, zamana bağlı değişkenler yerine, örneklenmiş eşdeğerleri yazılmaktadır. Yalnız sistemin işleyişini geliştirmek amacıyla, g , referans ağırlık terimi tanımlanmıştır. Bu sayede orantı terimi, referans işaretin g katı olarak işleyecektir. Böylece orantı terimi,

$$P(t_k) = K_c (g y_r - y(t_k)) \quad [7.2]$$

şekline girmektedir. $t_k = kT$, T örnekleme frekansı olmak üzere, örnekleme zamanlarını göstermektedir. g , referans noktasının ve değişken yük cevabının bağımsız olarak düzeltilmesini ve işlemlerini sağlamaktadır.

Kontrol edilen sistemde, denetleyicinin geniş bir aralıkta çalışma durumunda ise, kontrol değişkeni belirlenen sınır değerleri aşabilir. Bu durumda, sistem kapalı çevrim çalışırken, geri

besleme çevrimi kırılıp, sistem açık çevrim çalışmaya başlayabilir. Bu durum söz konusu iken, entegral yapısı, hata işaretinin entegralini almaya devam eder ve entegral terimi çok büyük değerlere ulaşabilir. Bu duruma, entegral sisteminin tükenmesi denilmektedir. Hata terimi, entegral yapısının eski haline gelmesi ve normal çalışma durumuna dönmesi için uzun bir süre boyunca işaret değiştirmelidir. Bu olay, sonlu kelime genişliği olan bir işlemcide sorun yaratacaktır. İşlemci sadece belirli genlikteki sayıları saklayabildiğinden, kontrol değişkeninde taşma oluşacaktır. Bunun önlenmesi sistemin doyma modunda çalıştırılması ile mümkündür. Entegral teriminin bu sorununu önlemenin çeşitli yolları bulunmaktadır. Sistem içinde fazladan bir geri besleme çevrimi kurup, sistemin çıkışını ölçmek ve denetleyici çıkışı v ile sistemin çıkışı u arasındaki farkı bularak bir hata işareti üretilebilir. Eğer sistemin çıkışının ölçülmesi mümkün değilse, sistemin matematik modeli kurularak işaret belirlenmelidir. Hata işareti entegral yapısının girişine $1/T_i$ kazancı ile uygulanmaktadır. $1/T_i$ takip zaman sabiti olarak adlandırılmaktadır. Fazladan kurulan geri besleme, denetleyici çıkışı, doymuş değerdeki çıkışı takip edene kadar entegralin değerinin değişmesini sağlamaktadır. Çıkışın takip etmesi işlemi, T_i zaman sabiti ile yerine getirilmektedir. Bu yöntem ile entegral teriminin tükenmesinin önlenmesi durumunda yapı ,

$$I(t) = \frac{K_c}{T_i} \int e(s) ds + \frac{1}{T_i} \int (u(s) - v(s)) ds \quad [7.3]$$

şeklini almıştır. Sistemin, ayırık zamanda gösterilmesi için, $e_s(t) = u(t) - v(t)$ olmak üzere, türevi alınmakta ve

$$\frac{dI(t)}{dt} = \frac{K_c}{T_i} e(t) + \frac{1}{T_i} e_s(t) \quad [7.4]$$

şekline sokulmaktadır. Sistem ayrıklaştırılarak düzenlendiğinde, h örnekleme periyodu olmak üzere, entegral terimi;

$$I(t_k + 1) = I(t_k) + \frac{K_c h}{T_i} e(t_k) + \frac{h}{T_i} e_s(t_k) \quad [7.5]$$

şeklini almaktadır.

PID sistemindeki türev elemanı, işaretin tam anlamıyla türevini aldığı anda, yüksek frekanslarda kazanç çok büyük olacağından ve bu yüksek frekanstaki gürültülerin kuvvetlenmesine neden olacağından tercih edilmemektedir. Sistemdeki sT_d türev elemanı yerine yaklaşık olarak,

$$\frac{sT_d}{1+sT_d/N} \quad [7.6]$$

kullanılabilir. Bu yaklaşıklık, frekans değerleri N/T_d 'nin altında olan işaretler için doğru olmaktadır. N maksimum türev kazancı olarak adlandırılmaktadır. Analog denetleyicilerde N sabit bir değer olup, 5 ile 20 arasında değişmektedir. Bu durumda, sürekli zamanda türev işareti,

$$D(t) + \frac{T_d}{N} \cdot \frac{dD(t)}{dt} = -K_c T_d \frac{dy}{dt} \quad [7.7]$$

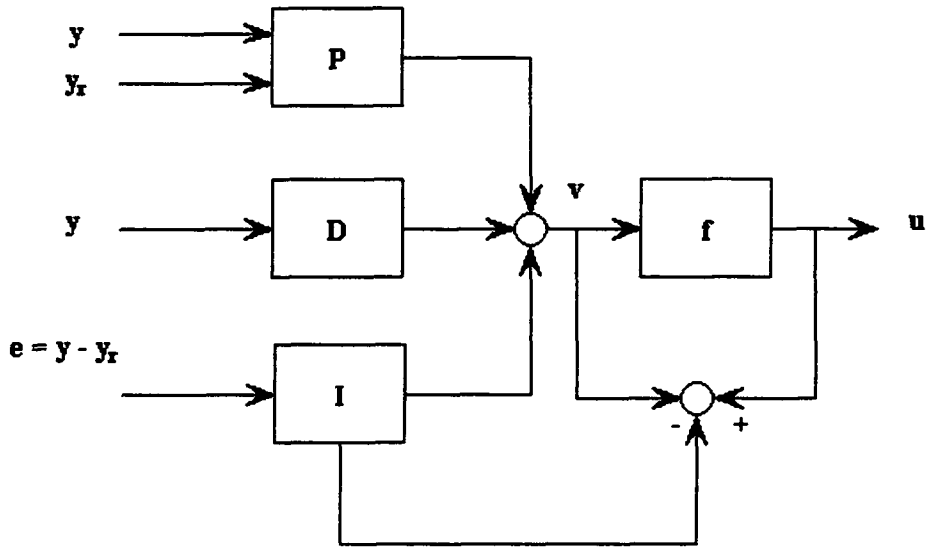
şeklini almaktadır. Bu terimin ayrıklaştırılması ile;

$$D(t_k) = aD(t_k - 1) - b(y(t_k) - y(t_k - 1)) \quad [7.8]$$

$$a = \frac{T_d}{T_d + N_h} \quad b = \frac{K_e T_d N}{T_d + N_h} \quad [7.9]$$

şeklinde türev elemanı elde edilmektedir. Böylece klasik PID kontrol algoritması yanında sistemin hata yapmasının mümkün olduğu alanlardaki davranışını düzelteren bir yapı elde edilmiştir. Birçok durumda, türev elemanı sistem için gerekli olmamaktadır. PI denetleyici yapısı sistemin kontrolüne yetmektedir. Bunun için, türev elemanı sistemden çıkarılarak, sadece oran ve entegral terimleri kullanılmaktadır. Ayrık zamanda elde edilen formüller yardımıyla, sistem birbirine paralel ve bağımsız yapılar şeklinde kurulabilir.

Sistemin dijital bir yapı olarak programlanabilmesi için, her terimin kaç işlem den oluştuğunun belirlenmesi, tasarıma yardımcı olacaktır. Özellikle sabit nokta ile işlem yapıldığında, hesaplamadan çok sayıların ölçeklendirilmesi ve taşma önlenmesi işlem sayısını ve süresini arttırmaktadır. PID ve PI algoritmaları ayrık zamanda programlandığında toplama ve çarpma işlemleri ile temsil edilebilirler. PID algoritması toplam 15 işlem, PI algoritması ise 10 işlem ile gerçekleştirilebilir. Bu işlemlerin kaç komut çevrimi sürdüğü ve bir komut saat çevrim süresi çarpılarak, algoritmanın ne kadar sürede işleyeceği belirlenebilir.



Şekil 7.3 : Geliştirilmiş PID yapısı

Sistem tasarımı sırasında örnekleme periyodunun seçimi sistemin davranışını etkileyecektir. PI yapısı kullanıldığında, örnekleme periyodu, entegral zaman sabitine bağlıdır. h örnekleme periyodu olmak üzere, h/T_i 0.1 ile 0.3 arasında olmalıdır. PID yapısı için gereken örnekleme periyodu daha kısa olmalıdır. Bu durumda, hN/T_d değeri, 0.2 ile 0.6 arasında olacak şekilde örnekleme periyodu seçilmelidir.

PID yapısının kolay programlama ve kontrol edilebilmesi için paralel bir yapı haline getirilmesi işlemleri kolaylaştırmaktadır. Gereken çıkış işaretini elde etmek için her bir yapı ayrı ayrı ölçeklendirilebilir. Sayıların geniş bir aralıkta değişimi ve değerlerinin büyük olması, katsayıların genliklerinin ölçeklenmesini gerektirmektedir.

Katsayılar incelendiğinde, en büyük değer, türev teriminden gelmektedir. Bu yüzden K_c kazancı ile N yüksek frekans türev kazanç katsayılarının sınırlandırılması gerekmektedir. Eğer K_c ve N , 16 değerine sınırlanırsa, $K_c N = 256$ ve $K_c \leq 16$ olacaktır. Bu değerler saklanmadan önce, sırasıyla 256 ve 16 'ya bölünmelidir. İşaretin genliğini düzeltmek için, türev terimi 8 bit sola, orantı terimi 4 bit sola kaydırılmalıdır. Diğer katsayılar, sayı aralığının içinde olmasına rağmen çok küçük değerler almasını önlemek amacıyla, bir alt sınır konulmaktadır.

Denetleyici parametrelerinin işlenmesi sırasında taşma olmaması istenmektedir. Paralel yapıdan, v çıkış işaretini elde ederken her terimin toplanması sırasında taşma önlenmelidir. Orantı terimi daima sayı aralığının içinde olacaktır. Çünkü ondalık sayıların çarpımı daima ondalık ve 1'den küçük sonuç verecektir. Ancak K_c , 1'den büyük iken, işaretin genliğinin

düzeltilmesi sırasında taşma oluşabilir. Bu durumu önlemek için, doyma modu kullanılmalıdır. Entegral teriminde, tükenme önleyici sıfırlama kullanılması, entegral teriminin sayı aralığı içinde kalmasını sağlamaktadır. Bu yüzden doyma modunda çalışma gereksizdir. Türev terimi sistemin çıkışına bağlı olduğundan, analitik ölçekleme yöntemlerinin kullanılması doğru sonuç vermemektedir. Bu yüzden sistemin kapalı çevrim cevabının benzeşimi yapılarak, türev teriminin davranışı incelenebilir. Türev terimi, saklanmadan önce doyma modu kullanılarak değeri düzeltilmelidir.

BAŞLANGIÇ

Program belleğinden Veri belleğine sabitleri yükle Değişkenleri temizle

yr ve $y(n-1)$ değerlerini yükle

Dış birimleri sıfırla

PID

$y(n)$ girişini bekle

D türev yapısını işle

Yuvarla, taşmayı kontrol et ve D 'yi sakla

P orantı yapısını işle

D, P ve I 'yı topla

Yuvarla, taşmayı kontrol et ve $v(n)$ içinde sakla doyma fonksiyonundan $u(n)$ 'i işle

$u(n)$ çıkışını üret

I entegral yapısını işle

Taşmayı kontrol et ve kelime olarak I 'yı sakla

PID 'ye GiT

Yüksek değerde bir kazanç $K_c > 1$ elde etmek için, algoritmaya bir katsayı eklenebilir veya sistemin kazancı, kontrol yapısının dışına alınabilir.

Lineer bir devre elemanı ile gerçekleştirilebilen bu sistem ile taşma olması önlenmektedir. Bu, daha kısa bir yazılım ve daha hızlı bir yapı elde edilmesini sağlamaktadır. Bunun yanında, sınırlı salınımların oluşmasına neden olmaktadır. Ayrıca normal çalışma durumunda hata

düşük değerde olacağından, kontrol değerindeki küçük bir değişiklik kuvvetlenerek sisteme yansiyacak ve sistem cevabını bozacaktır. Bu yüzden, yüksek bir kazanç değeri istendiğinde, sistem doyma modunda çalıştırılmalıdır.

PID algoritmasının geliştirilmesi için akış diyagramının kurulması, sistemin parametrelerinin ve kontrol değişkenlerinin belirlenmesi gerekmektedir. Kontrol algoritması iki temel yapıdan oluşmuştur. İlk olarak, DSP sisteminin başlangıç değerlerinin yüklenmesi, daha sonra PID yapısının hesaplanması ve işletilmesidir.

DSP sisteminin programlanması sırasında bu şekilde bir yapının kurulması sistemin tasarımını kolaylaştıracaktır. Algoritmanın başlangıç bölümünde, program, kesme izinlerini kaldırmakta, taşma modunu işletmekte ve program belleğinden veri belleğine katsayılar yüklemektedir. Daha sonra, denetleyicinin değişkenleri temizlenmekte ve referans değer ile bir önceki çıkış değeri okunmaktadır. Daha sonra, program sonsuz döngüye girerek, PID algoritmasını işlemektedir. Sistemin değişkenlerinin saklanması öncesinde, gerekli olan taşma kontrolü ve yuvarlatma işlemi yapılmalıdır. Değerler saklanırken, ölçeklendirme katsayıları ile çarpılarak, hatanın minimum olması sağlanmaktadır. Bütün bu işlemlerin ne kadar çevrimde tamamlanacağı işlemler teker teker ele alınarak hesaplanabilir. Böylece, algoritma için gereken en az süre hesaplanmış olmaktadır.

PID yapısının kontrol sistemlerinde yaygın olarak kullanılmasını sağlayan basit kullanım özelliği olmuştur. Sistem gereksinimleri değiştikçe yeni kontrol sistemleri geliştirilmiştir. Örneğin, kontrol edilen sistemin aşırı ve minimum yerleşme zamanında istenen değere yerleşmesi için ölü zaman kontrolü denilen yapı kullanılmaya başlanmıştır. Sistem parametrelerinin tam olarak bilindiği varsayımı ile sisteme yeni kutuplar tanımlayan bu yapı basit ve bozucu etkileri bulunmayan sistemlerin kontrolünde kullanılmaktadır. Bunun yanında, değişkenlerin algılanması sırasında karşılaşılan problemler nedeniyle, değişkenlerin kestirimi ve böylece çok daha kapsamlı ve doğru bir kontrol sisteminin geliştirilmesi yoluna gidilmiştir. Tasarlanan denetleyici yapısı, durum geri besleme kontrolü ile sistemin yapısını değiştirerek, sistemin parametre değişimlerinden ve dış bozucu etkilerden etkilenmesini önlemektedir. Adaptif kontrol olarak adlandırılan bu yapı ile karmaşık sistemler kolaylıkla işlenebilmektedir. Adaptif bir sistem, çevre etkilerinden bağımsız olarak, gerek sistemin kendisinden gerek dış ortamdan gelen bozuculara karşı sistemi uyumlu hale getirebilmektedir.

Adaptif bir sistemde klasik kontrol sistemlerindeki gibi sistemin matematiksel modeli tam olarak bulunmamaktadır. Bunun yanında, sisteme uygulanan deneyler yardımıyla, bir

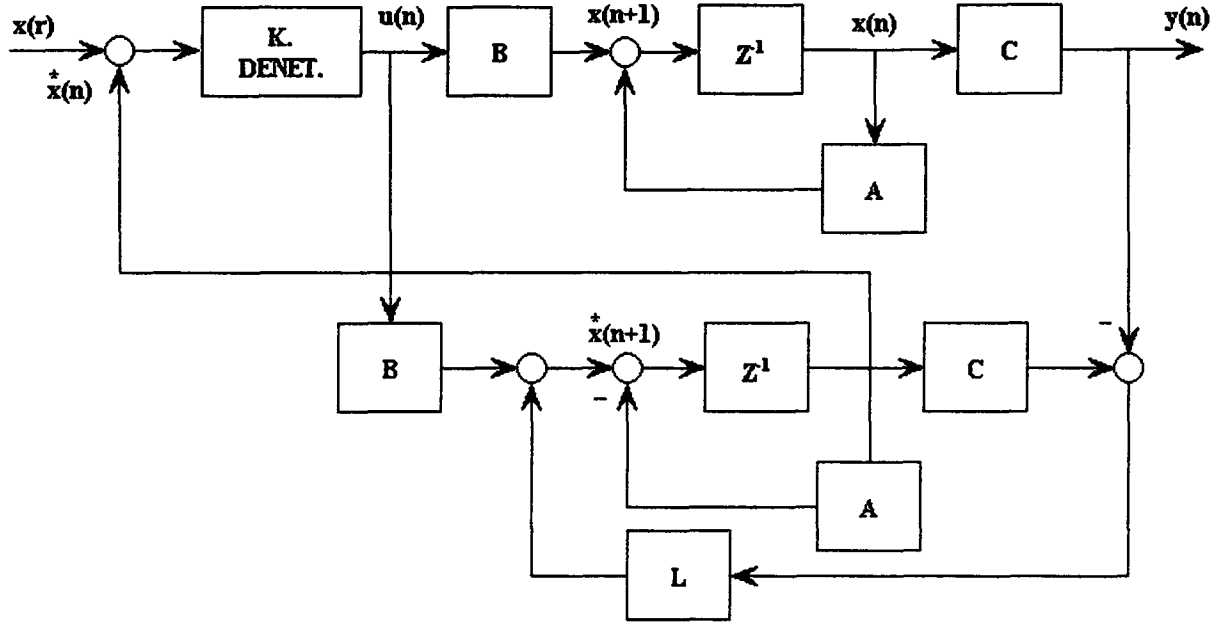
denetleyici yapısı belirlenmektedir. Sistemin kontrolunun sağlanabilmesi için, bir veya daha fazla değişkenin kestirimi ve gürültü ile bozulacak yapıdaki değişkenlerin yerine konulması gerekmektedir. Kestirim işlemi, kontrol edilen sistemden gereken ölçüm bilgisinin tam olarak alınamaması durumunda bilginin hesaplanması olarak adlandırılabilir. Elde bulunan veri, belirli bir hata içerebilir veya dış bozucu etki sayesinde değerini devamlı değiştirebilir. Bu durumda bir gözlemleyici veya kestirim sistemi ile kontrol değişkenlerinin tam olarak belirlenmesi sağlanabilir. Kalman filtresi, bu gibi sorunları çözmek için geliştirilmiş optimum bir gözlemleyici yapısıdır.

Şekil 7.4 'teki gözlemleyici yapısında, A, B, C durum matrislerini, L geri besleme matrisini, K gözlemleyici kazancını göstermektedir. u kontrol, x durum, y çıkış işaretini belirtmektedir.

$$\dot{X}(n+1) = A\dot{x}(n) + Bu(n) + KC \left[\dot{x}(n) - x(n) \right] \quad [7.16]$$

$$\text{Hata} : e(n) = \dot{x}(n) - x(n) \quad e(n+1) = (A + KC)e(n) \quad [7.17]$$

Adaptif kontrol sisteminde model referans tekniği denilen yapı kullanılarak sistemin tanımlanma, kestirim ve kontrolü gerçekleştirilmektedir. Lineer olmayan bir sistemin doğrusallaştırılması işlemine benzer bir yapı ile sistemin durum uzay modelinden sınır koşulları tanımlanarak nominal yörünge elde edilmektedir. Daha sonra jakobyen matrisler kullanılarak sistem doğrusal bir hale getirilmektedir. Gözlemleyici, algılayıcılar tarafından belirlenemeyen ve ölçülemeyen sistem parametrelerinin eldesi için kullanılan dijital bir filtre yapısıdır. Kontrol edilen sistemi iyileştirmek için düzeltilmiş durum girişine geri beslenerek kutup yerleştirme veya model izleme gibi yapıların kurulmasına olanak sağlanabilir.

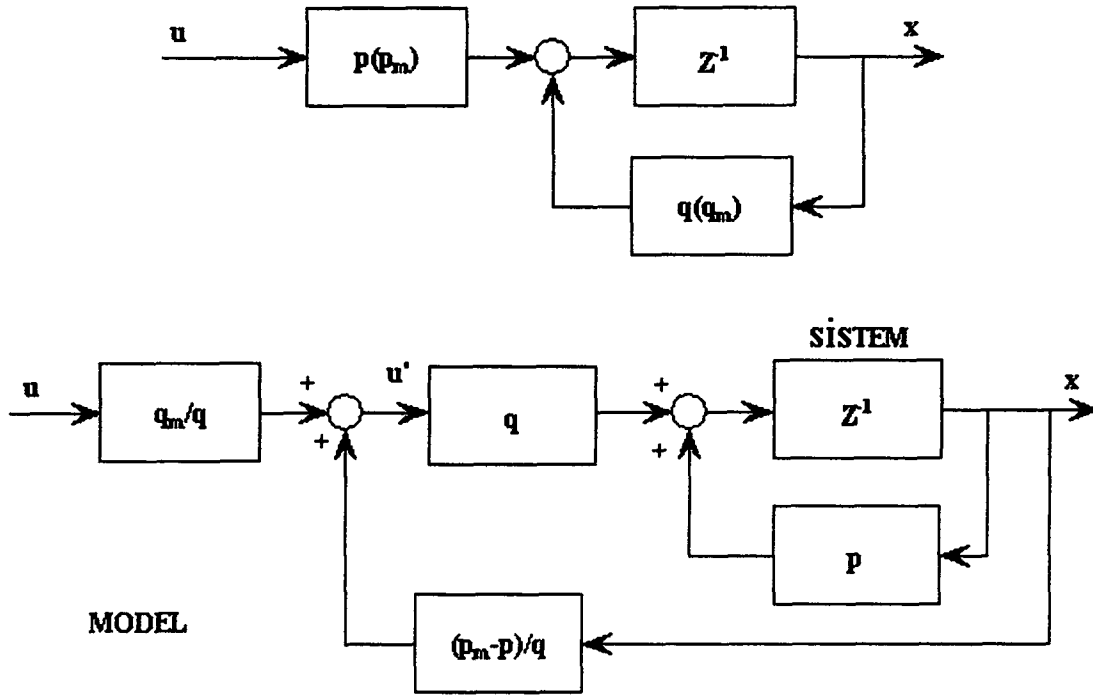


Şekil 7.4 : Gözlemleyici ve durum geri besleme kontrolü

Eğer sistemin kestirimi ve gürbüz kontrol yapısı kullanılmakta ise, sistemin tanımına yani modellenmesine gerek duyulmaz iken, model referans adaptif kontrol yapısında bu yapı kurulmalıdır. Kurulan model, ölçülebilir giriş, çıkış ve fiziksel büyüklükler yardımıyla veya Newton ve Kirchoff yasaları ile hesaplanan değerler ile model kurulmaktadır. En küçük kareler yöntemi ile tanımlama şu yapıdadır: $z(k)$ ölçülen değer, $y(k)$ gerçek değer, $v(k)$ gürültü ise,

$$z(k) = y(k) + v(k) \quad y(k) = \sum h(m)u(k-m) \quad [7.17]$$

şeklinde hesaplanmaktadır; $h(m)$ terimi, $z(0), z(1), \dots, z(N)$ değerleri kullanılarak $z = Uh + v$ yapısı kurularak belirlenebilir. Bu değerden ters matris işlemi ile; $h = (u^T V)^{-1} v^T z$ belirlenmektedir. DSP ile bu değer kolaylıkla hesaplanabilir. Modelleme sırasında, giriş ve çıkış işaretlerinin yeterli olması, modelin basit fakat bütün durumları içermesi ve model için tanımlama tekniğinin geliştirilmesi gerekmektedir.



Şekil 7.5 : Birinci dereceden kontrol edilen sistem ve modeli

Adaptif kontrol sisteminde birinci dereceden denetleyici ve sistemin modeli Şekil 7.5 'de olduğu gibidir. Burada;

$$p = e(-T_c / T_s) \quad q = K(1 - p) \quad [7.18]$$

$$pm = e(-T_{cm} / T_s) \quad qm = Km(1 - pm) \quad [7.19]$$

olarak tanımlanmaktadır. K ve Km kazanç, T_c ve T_{cm} sistemin ve referans modelin zaman sabitleri, T örnekleme periyodudur. Kontrol edilen sistemden beklenen, cevabının referans modelin cevabını takip etmesidir. p ve q kontrol edilen sistemin parametreleri, bilinmeyen veya değişken yapıdadır. Bu yüzden kompanzasyon sisteminin katsayıları da değişken olmalıdır. Sistemin kestirimini gerçekleyen sistem p(k) ve q(k) değerlerini doğru ve hızlı bir şekilde belirlediğinde, adaptif modelin doğruluğundan bahsedilebilir. Model referans sistemin kestirim algoritması;

$$p(k+1) = p(k) + K_p x(k) e^*(k+1) \quad [7.19]$$

$$q(k+1) = q(k) + K_q U(k) e^*(k+1) \quad [7.20]$$

olarak tanımlanmaktadır. $e^*(k)$ adaptif hata işareti, ve U(k) kontrol edilen sistemin referans giriş işareti olarak tanımlandığında,

$$e^*(k+1) = \frac{P_m e^*(k) + x(k+1) - (p(k)x(k) + q(k)U'(k))}{1 + K_p x^2(k) + K_q U'^2(k)} \quad [7.21]$$

$$U(k) = \frac{(P_m - P(k+1))x(k+1) + q_m U(k+1)}{q(k+1)} \quad [7.22]$$

formülleri yardımıyla hesaplanabilir. Adaptif kontrol sistemi tasarımında, model referans kontrol veya Kalman filtresi kullanılması, sistemin cevabını etkilemeyecektir. Sistem durum uzay modelinin kurulması ve işletilmesi, matris ve vektör hesaplarına dayandığından çok sayıda çarpma ve toplama işlemi yapılmaktadır. Sistemin gerçek zamanda kontrolü söz konusu olduğundan bu işlemlerin en kısa sürede gerçekleştirilmesi gerektiğinden, DSP sistemleri tam çözüm olanağı sunmuştur. Sabit nokta DSP kullanıldığında, sayıların taşmaya uğramaması için ölçeklendirme gerekmektedir. Böylece herhangi bir kontrol sisteminin adaptif kontrolü DSP ile yüksek bir performansla gerçekleştirilmektedir.

8. SONUÇLAR VE UYGULAMA ÖRNEĞİ

Önceki bölümlerde de detaylı şekilde anlatıldığı gibi bir sistemi kontrol edecek denetleyicinin seçimi sırasında bir çok faktör gözönünde bulundurulmalıdır. Denetleyicinin cinsi ve yapısı uygulama ile doğrudan ilişkilidir. Özellikle motor kontrolü uygulamalarında, optimum kontrol algoritmasının ve buna bağlı olarak da denetleyicinin seçimi, sistemin maliyetini önemli ölçüde etkileyecektir. Sıradan bir analog PID yapısı ile gerçekleştirilebilecek bir kontrol prosesinin, kayan nokta işlemcisi ile gerçekleştirilmesi sonucunda denetleyici maliyeti kontrol edilecek sistemin maliyetini dahi aşacaktır. Bununla birlikte dijital bir kontrolörün, sistem yapısındaki değişikliklere, analog denetleyicilere nazaran daha kolay uyum sağlayabildiği gözönünde bulundurulmalıdır.

Günümüzde motor kontrolünde kullanılan algoritmalar, karmaşık matematiksel işlemlerin denetleyici yapısı tarafından gerçekleştirilmesini gerektirmektedir. Bu nedenle uygulamalarda mikroişlemciler ya da DSP'ler sıkça kullanılmaya başlanmıştır. Özellikle sabit noktalı DSP fiyatlarının mikroişlemciler ile aynı seviyeye gerilemesi ve motor kontrolü için TMS320C240 gibi özel DSP'lerin üretilmesi kontrolör yapılarının bu işlemciler ile gerçekleştirilmesini son derece cazip hale getirmiştir.

DSP'lerin yüksek bant genişliğinde işlem yapabilme yetenekleri sayesinde; dayanıklı, adaptif ve algılayıcısız dijital kontrol sistemleri çok düşük maliyetlerle gerçekleştirilebilmektedir. DSP'ler mikrokontrolörlerin sunduğu olanakların yanısıra, yüksek hızlarda işlem yapabilme, yüksek çözünürlük ve paralel çalışma gibi özellikleri de kullanıcıya sunarlar. DSP'lerin hızlarının asıl kaynağı, iki yollu Harvard mimarisinden ve bir tek makine çevriminde çarpma ve toplama yapabilen komutlarından gelir. Bu mimari tipinde yollardan biri veri iletimi için kullanılırken, diğer yol program komutlarının aktarılmasında kullanılır. İşlemci her iki yola da aynı anda müdahale edebileceğinden, işlemlerde zamandan kazanılmış olur. Örneğin çarpılacak iki sayının da ayrı ayrı yollardan çarpıcı girişine getirilmesi dahi, işlem hızını iki katına çıkarır. (Küçükdemiral,1999)

Geçmiş yıllarda DSP fiyatlarının çok yüksek olması DSP'lerin sanayide geniş ölçüde kullanılmasını engelleyen yegane faktördü. Fakat hızla ilerleyen mikroelektronik teknolojisi ile birlikte azalan maliyetler DSP fiyatlarını da oldukça aşağıya çektiği gibi, DSP'lerin yapılarının mükemmelleşmesine neden olmuştur. Örneğin, TMS320C62x ve TMS320C67x gibi yeni nesil DSP'ler VelociTI™ mimarisi ile üretilmiştir. VelociTI mimarisi yüksek performanslı, gelişmiş, çok uzun komut kelimesine (VLIW – very long instruction word)

sahip bir mimaridir ve Texas Instruments firması tarafından geliştirilmiştir. Bu mimari sayesinde DSP'ler çok kanallı ve çok fonksiyonlu uygulamaların vazgeçilmez bir parçası olmuşlardır. Saniyede 6000 milyon komut işleme yeteneği ile TMS320C6x işlemcisi yüksek performans gerektiren tüm problemlere çözüm getirebilmektedir. TMS320C6x DSP'si sistem mühendislerine ürünlerini sınırsız geliştirme imkanı getirmiştir. Yüksek performans, kolay kullanım, düşük maliyet TMS320C6x jenerasyonunu çok kanallı, çok fonksiyonlu uygulamaların tartışmasız en iyi çözümü halini almıştır.

Arttırılmış komut işleme paralelliği ise işlemcinin hızına hız katmıştır. C62x DSP'ler 200Mhz (5ns çevrim süresi) çalışırlar. 'C67x işlemciler ise 167Mhz de (6ns çevrim süresi) çalışırlar. Her iki DSP de, her çevrim süresi içerisinde 8 adet 32 bitlik komutu işleyebilirler. DSP'nin çekirdeği 32 adet genel amaçlı, 32 bit kelime uzunluğunda kaydedici ve sekiz adet fonksiyonel ünite içermektedir. Ayrıca 2 çarpıcı; 6 adet Aritmetik lojik ünite (ALU) içermektedir. Ayrıca tasarımcıya çok hızlı işlem yapabilen RISC benzeri kodlar üretme seçeneğini sunmaktadır. Tüm komutlar dallanma yeteneğine sahiptirler. Böylece performansı düşüren dallanma işlemi en aza indirilmiş olur ve paralellik arttırılmış olur. Kodlar bir birinden bağımsız fonksiyonel bloklar içerisinde çalışabilmektedirler. 8/16/32 bit veri desteği sayesinde hafıza en kullanışlı hale getirilmektedir. 40 bitlik aritmetik işlem seçeneği bazı hassas hesaplama gerektiren uygulamalar için işlemciye eklenmiştir. Bunların yanı sıra doyum ve normalizasyon desteği de sağlamaktadır.

DSP'ler, mikrokontrolörlere göre çok daha hızlı işlem yapabilirler. Örneğin bir mikrokontrolör veya analog kontrolör motor hızını okumak için takogeneratör kullanırken, DSP hızı tahmin edebilmektedir.

DSP kullanıldığında, FFT (fast fourier transform) gibi dönüşümler çok az komutla gerçekleştirilerek spektrum analizi yapılabilir. Mekanik titreşimlerin frekans spektrumlarının incelenmesi sonucunda erken hata tespiti de yapılabilir. Sistemin çalışması esnasında, sistemin ağırlığının, dengesinin, yükünün değişmesi neticesinde meydana gelen titreşim modlarının FFT verileri kullanılarak izlenmesi ile filtre katsayıları sürekli olarak , sistem çalışır durumdayken, ayarlanabilir.

DSP tabanlı sistemler yardımı ile uygun kontrolörler ve algılayıcısız yapılar kullanılarak, sistem maliyetleri minimuma indirgenebilir.

Ayrıca geliştirilmiş algoritmalar ile daha az harmonikli sistemler oluşturulabilir. Programlama sırasında sıkça karşılaşılan hafıza yetersizliği problemi, çok işlevli DSP komutlarının look-up

tablolarını ortadan kaldırması ile çözülmüştür.

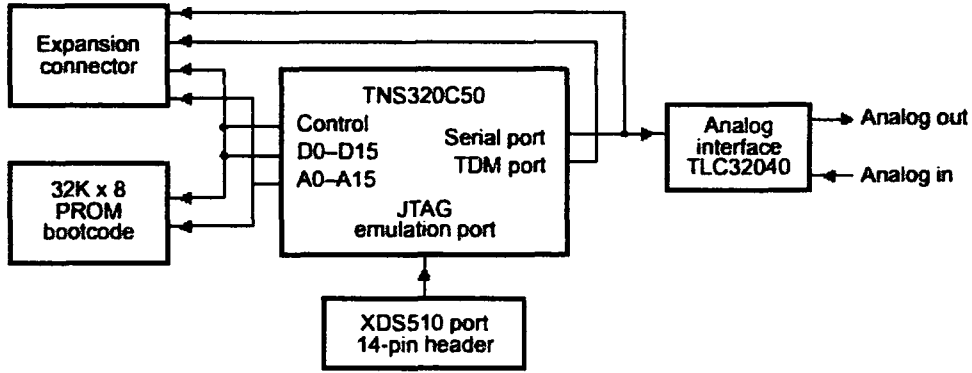
Genel kontrol uygulamalarında mikroişlemci yerine DSP'lerin kullanılması kontrol algoritmalarının gitgide karmaşıklaşması ile belirginleşmiştir. DSP'lerin mikroişlemcilere göre tercih edildiği uygulamalar, matematik işlemlerin ağır ve yoğun olarak kullanıldığı uygulamalardır.

Dijital işaret işleme sistemlerinin bir diğer özelliği de gerçek zamanda işlem ihtiyacı göstermeleridir. Kullanıcıya belirli bir gecikme ile ulaşan bir işaret, sistemin hatalı çalışmasına neden olacaktır. Örneğin motor kontrol sisteminde yük üzerindeki ani bir değişikliğin kontrolör tarafından zamanında algılanamaması, sistem cevabının yanlış olmasına neden olacak ve motorun tahrip olmasına kadar gidebilecek sorunlar ortaya çıkacaktır. Dolayısıyla, eğer işaret dijital formatta işlenecekse DSP gibi hızlı işlem yapabilen işlemcilere ihtiyaç duyulacaktır.

Dijital işaret işlemenin temel problemi işaretin örneklenmesidir. İşaretin doğru olarak algılanması, örnekleme frekansının yeterince büyük seçilmesine bağlıdır. Kontrol uygulamalarında kullanılması gereken işlemci, bant genişliğinin en az on katı bir hızla örnekleme yapabilmelidir. Ancak bu şekilde işaretin anlamını kaybetmesi önlenemez. DSP sistemleri işareti yeterince yüksek bir hızda örnekleymeli ve elde edilen verileri saklayabilmelidir. Uygulamanın yapısına göre örnekleme frekansı da değişim göstermektedir. Örneğin, kontrol uygulamalarında 1kHz'lik örnekleme frekansı yeterli olurken, haberleşme uygulamalarında 8 kHz, ses işleme uygulamalarında 8-10 kHz, müzik işleme uygulamalarında 40-48 kHz ve görüntü işleme uygulamalarında ise 14 MHz'lik örnekleme frekansına ihtiyaç duyulmaktadır. (Küçükdemiral,1999)

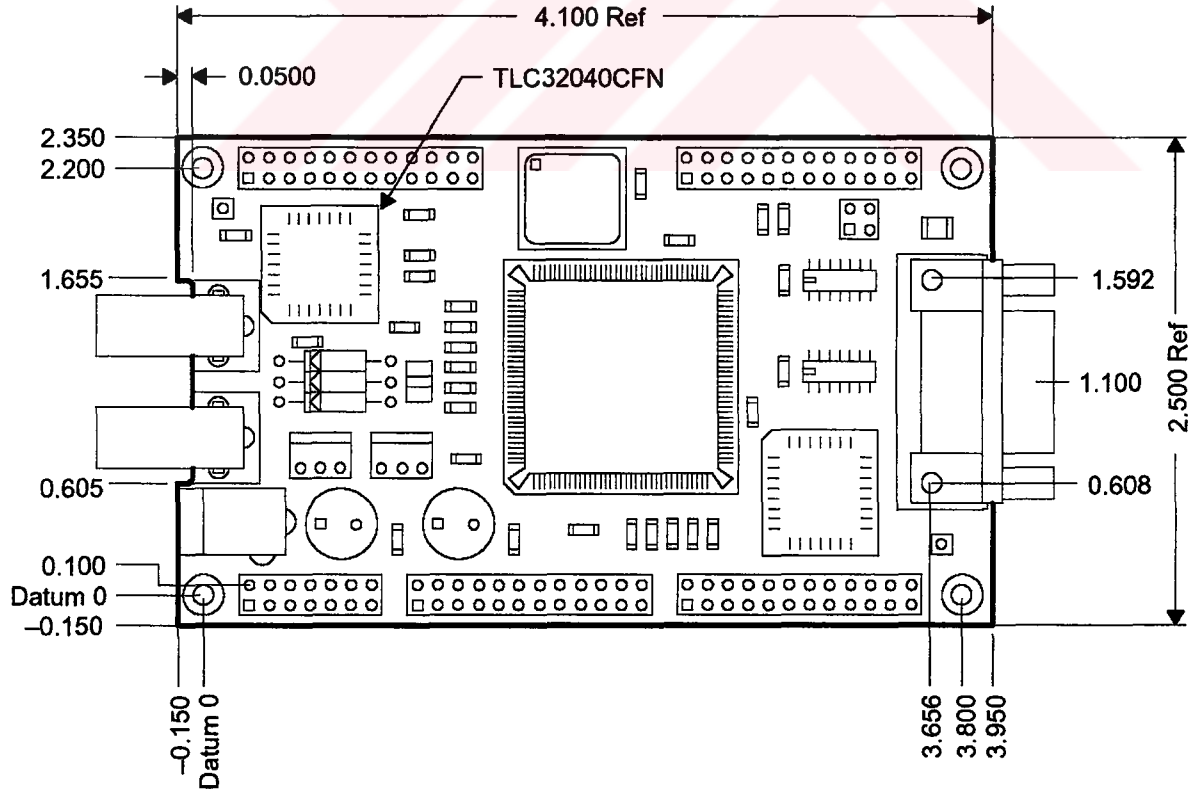
8.1 DSP Tabanlı Kontrolör Örneği : Bir D.C. Motorun Hız Kontrolünün TMS320C50 Tabanlı Bir Kontrol Sistemi Kullanılarak Gerçekleştirilmesi

Aşağıda açıklanan uygulamada TMS320C50 DSP'si ile birlikte TMS3200051 DSP starter kit'i de kullanılmıştır. Bu kartın özellikleri ve blok diyagramı aşağıda verilmiştir.



Şekil 8.1 : TMS3200051 DSP starter kit'i blok diyagramı

TMS3200051 kartı C5x DSP ailesi kullanılarak yapılacak uygulamaların geliştirilmesi için gerekli donanım alt yapısını sağlamaktadır. Kart, RS232 portunu kullanarak bir PC ile haberleşebilir ve Windows uyumludur. Çalışma frekansı 40MHz'dir ve yine aynı frekans değerindeki bir osilatör kart üzerinde bulunmaktadır. 14 bitlik çözünürlüğe sahip TLC32040 analog arayüz yongası kartın önemli özelliklerinden biridir. Kartın beslemesi için 9V, 250mA'lık AC güç kaynağı yeterlidir.

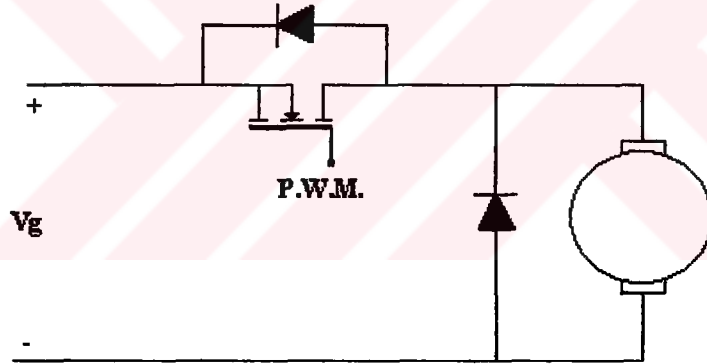


Şekil 8.2 : TMS3200051 DSP starter kit'i yerleşim planı

Starter Kit üzerinde harici hafıza birimi olmamasına rağmen DSP üzerinde bulunan 10K boyutundaki RAM hafıza birimi bir çok uygulama için yeterlidir. Kernel programı 32K ve 8 bitlik PROM hafızasında saklanmaktadır. Bu hafıza sadece açılış sırasında gerekli işlemlerin yapılması için kullanılır ve daha sonra erişimi mümkün değildir.

TLC32040 AIC yongası tek kanallı analog giriş-çıkış için arayüz desteği sağlar. 14 bitlik çözünürlüğe sahiptir. Ayrıca örnekleme frekansı istenilen bir değere ayarlanabilir. AIC C50'nin seri portunu kullanır ve C50'deki 10MHz'lik çıkışı referans alır.

Örneğimizde 20 Voltluk bir doğru akım motorunun TMS320C50 sayısal işaret işleyicisi ile dinamik hız kontrolü gerçekleştirilmiştir. Kontrol edilecek olan motorun miline 60 Voltluk ve 24 Voltluk birer D.C. motor daha bağlanmıştır. 60Voltluk D.C. motor bir takogeneratör olarak kullanılmış, uçlarından alınan gerilim bir gerilim bölücü devre ve alçak geçiren filtreden geçirilerek DSP'ye geri besleme işareti olarak uygulanmıştır. Diğer 24 Voltluk motor ise yük olarak kullanılmıştır. Takogeneratör olarak kullanılan motorun geriliminin diğerlerine göre büyük olması ile düşük gerilimlerde dahi lineer bir çalışma elde edilmiştir.



Şekil 8.3 : Ana akım devresi

Hız kontrol devresi bir DC kıyıcıdır ve ana akım devresi yukarıdaki şekilde verildiği gibidir. Bu devrede anahtarlama elemanı olarak MOSFET (Metal-Oxide-Semiconductor-Field-Effect-Transistor) kullanılmıştır. MOSFET'in kapısına uygulanan kontrol işareti +12V ile -12V arasında değişen bir dikdörtgen dalga şeklindedir. +12Voltluk işaret MOSFET'e uygulandığı zaman eleman iletime girmekte ve DC motoru şebekeye bağlamakta; -12Voltluk kapı gerilimi ise motoru 20Voltluk şebekeden ayırmaktadır. Hızı kontrol edilen motora uygulanan PWM (Pulse Width Modulation) işaretinin periyodu sabit tutulmaktadır. Örnekleme frekansı 8 kHz'dir. DSP 125µs'de bir kesme sinyali üreterek motorun hızını takogeneratör vasıtasıyla okumakta ve kontrolör programını devreye sokarak bir sonraki kontrol işaretini üretmektedir.

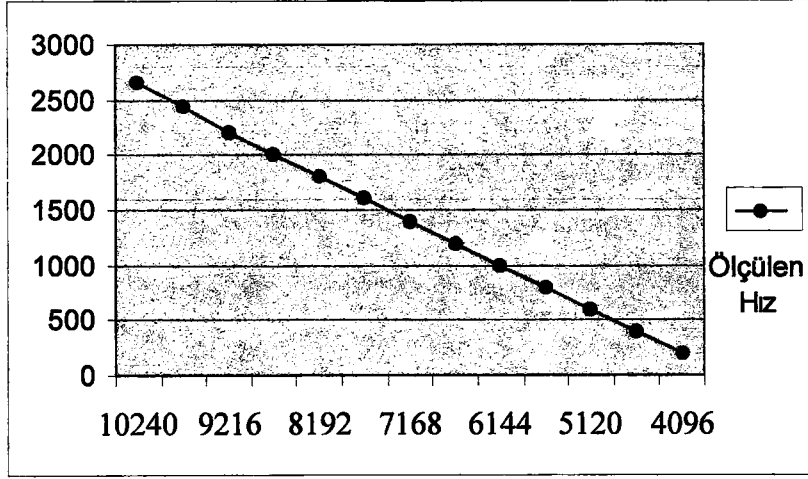
Kontrol işareti (PI denetleyicinin çıkışı) MOSFET'in kapısına uygulanan sürme işaretinin +12V'lık kısmının süresidir.

Kapalı çevrim kontrol sistemine uygulanan referans hız bilgisi yazılım ile sağlanmış ve heksadesimal formatta DSP programına yazılmıştır. Değişik referans hızlara karşı düşen hızlar aşağıdaki tabloda verilmiştir. Bu tablodan da görüleceği gibi referans hızlar ile motor hızı arasında lineer bir ilişki vardır. (Küçükdemiral,1999)

Örnekte PI denetleyici yapısı kullanılmıştır. Ancak DSP yazılımında yapılacak bir kaç değişiklikle kontrol algoritması kolayca değiştirilebilir. PI yapısındaki entegral denetleyicisi ile kararlı hal hatası sıfıra götürülmüştür. Oran kontrolörü ise sistemin ilgili referans hızla hızlı bir şekilde oturması sağlanmıştır. (Küçükdemiral,1999)

Çizelge 8.1 : Referans giriş ile hız ilişki tablosu

REFERANS HIZ (desimal)	Ölçülen Hız (d/dk)	Hız (heksadesimal)
10240	2650	2800
9728	2445	2600
9216	2200	2400
8704	2000	2200
8192	1800	2000
7680	1600	1e00
7168	1400	1c00
6656	1200	1a00
6144	1000	1800
5632	800	1600
5120	600	1400
4608	400	1200
4096	200	1000



Şekil 8.4: İlgili referans değerlere karşılık gelen hız bilgileri

KAYNAKLAR

Analog Devices, (1996), "ADSP-2100 Family DSP Microcomputers" , Analog Devices, Norwood

Çiprut İ. (1994) "DSP Tabanlı bir sistem ile Anahtarlama Relüktans Motorunun Kontrolü", Yüksek Lisans Tezi, İ.T.Ü

Küçükdemiral İ. B. (1999) "TMS320C50 Sayısal İşaret İşleyici Tabanlı Bir Kontrol Sistemi kullanarak Doğru Akım Motorunun Dinamik Hız Kontrolü" , Yüksek Lisans Tezi, Y.T.Ü

Pastacı H. ve Abbasoğlu H. İ., (1996), "Dijital Ölçmeler", Yıldız Teknik Üniversitesi, İstanbul

Smith W. Steven, (1997), "Scientist and Engineer's Guide to Digital Signal Processing", California Technical Publishing, California

Texas Instruments, (1998), "TMS320C240 User's Guide", Texas Instruments, Missouri

Texas Instruments, (1997), "TMS320C5x User's Guide", Texas Instruments, Missouri

Texas Instruments, (1998), "TMS320C5x DSP Starter Kit User's Guide", Texas Instruments, Missouri

Texas Instruments, (1991), "TMS320C1x User's Guide", Texas Instruments, Missouri

Texas Instruments, (1993), "TMS320C2x User's Guide", Texas Instruments, Missouri

Texas Instruments, (1997), "TMS320C30 User's Guide", Texas Instruments, Missouri

EKLER

EK1 TMS320C240 YONGASININ TERMİNAL AÇIKLAMALARI

EK 2 : TMS320C240 YONGASININ TERMİNAL YERLEŐİMİ VE BLOK DİYAGRAMI



EK 1 TMS320C240 YONGASININ TERMİNAL AÇIKLAMALARI



Terminal		Terminal Tipi	Terminal Tanımı
Adı	No.		
Harici Arayüz Bilgi/Haberleşme İşaretleri			
A0 (LSB)	110	Ç/Z	Paralel adres yolu [A0 (LSB) ve A15 (MSB) olmak üzere]. A15 – A0 harici veri/program hafızasını ya da G/Ç adreslemek için (multiplexed) çoklanmıştır. EMU/OFF aktif low olduğunda A15 – A0 yüksek empedans konumuna geçer ve düşük enerji modundan önceki değerini korur
A1	111		
A2	112		
A3	114		
A4	115		
A5	116		
A6	117		
A7	118		
A8	119		
A9	122		
A10	123		
A11	124		
A12	125		
A13	126		
A14	127		
A15 (MSB)	128		
D0 (LSB)	9		Paralel veri yolu [D0 (LSB) ve D15 (MSB) olmak üzere] . D15 – D0 , TMS320C240 ile harici veri / program hafızası ya da G/Ç alanı arasında veri transferi
D1	10		
D2	11		
D3	12		

D4	15	G/Ç/Z	yapması için çoklanmıştır. D15 – D0 düşük enerji modunda ya da (RS) aktif olduğunda ya da EMU1/OFF aktif low olduğunda ya da çıkış yoksa yüksek empedans konumundadır		
D5	16				
D6	17				
D7	18				
D8	19				
D9	22				
D10	23				
D11	24				
D12	25				
D13	26				
D14	27				
D15 (MSB)	28				
\overline{DS}	129			Ç/Z	Bilgi , program ya da G/Ç alanı seçim sinyali. Bu pinler low seviyesine çekilmedikçe (özel bir harici elemanla haberleşmek için) her zaman High seviyesinde bulunurlar. Ayrıca reset , enerji tasarruf modunda ve EMU1/OFF aktif low ise yüksek empedans konumundadırlar.
\overline{PS}	131				
\overline{IS}	130				

READY	36	G	Bilgi hazır sinyali. READY , harici ünitenin bus transaction için hazır olduğunu belirtir. Eğer bu pin low seviyesinde ise işlemci bir çevrim daha bekler ve bu ucu tekrar kontrol eder.
R / \overline{W}	4	Ç/Z	Oku/ Yaz sinyali. R/W , harici ünite ile haberleşme esnasında transfer yönünü belirtir. Bu pin normalde High (okuma) konumundadır ve reset , enerji tasarrufu modu ve EMU1/ OFF aktif low olduğunda yüksek empedans konumundadır.
\overline{STRB}	6	Ç/Z	Strobe. STRB harici bus çevrimini göstermediği zamanlarda daima High seviyesindedir. Reset , enerji tasarruf modu ve EMU1/ OFF aktif low olduğunda yüksek empedans konumundadır.

\overline{WE}	1	Ç/Z	Yazmayı mümkün kılan pindir. WE'nin düşen kenarı , cihazın harici bilgi bus'ı tarafından sürüldüğüne işaret eder (D0-D15). WE'nin yükselen kenarı sırasında bilgi harici program , bilgi ve G/Ç yazımı sırasında daima aktiftir. WE pini reset , enerji tasarruf modu ve EMU1 / OFF aktif low olduğunda yüksek empedans konumundadır.
W/\overline{R}	32	Ç/Z	Yazma/okuma pini R/\overline{W} 'nin ters şekli olup direkt olarak harici ünitenin çıkışa izin veren ucuna bağlanabilir. W/\overline{R} pini reset , enerji tasarruf modu ve EMU1/ \overline{OFF} aktif low olduğu durumlarda yüksek empedans gösterir
\overline{BR}	5	Ç/Z	Bus isteği sinyali. \overline{BR} ucu harici global veri alanına erişim esnasında aktif (low)tir.Bu pin veri hafıza adresleme alanını 32K kelimeye kadar arttırmak için kullanılır.

WDDIS	50	G	Flash hafıza programlama gerilim kaynağı. Eğer $V_{ccp} = 5V$ ise DSP üzerindeki tüm dahili flash hafıza bloğuna yazma ya da okuma yapılabilir. $V_{ccp} = 0V$ ise bellek içeriği değiştirilemez. WDDIS pini aynı zamanda donanım watchdog zamanlayıcısının yetkisini kaldırmaya yarar. Bunun için $V_{ccp}/WDDIS = 5v$ ve WDCR'nin 6. bitinin set edilmesi gerekir.
Paylaşımsız ADC girişleri			
ADCIN2	74	G	İlk ACD'nin analog girişleri
ADCIN3	75	G	
ADCIN4	76	G	
ADCIN5	77	G	
ADCIN6	78	G	
ADCIN7	79	G	
ADCIN10	89	G	
ADCIN11	88	G	
ADCIN12	83	G	
ADCIN13	82	G	
ADCIN14	81	G	

ADCIN15	80	G	
BIT G/Ç ve Paylaşımlı Fonksiyon Pinleri			
ADCINO/IOPA0	72	G/Ç	İki yönlü dijital , ilk ADC'nin analoggirişi ADCINO/IOPA0 Tüm ünite resetleri tarafından dijital giriş olarak yapılandırılmıştır. Bu uç tüm cihaz resetleri tarafından dijital giriş olarak kabul edilir.
ADCIN1/IOPA1	73	G/Ç	İki yönlü dijital , ilk ADC'nin analoggirişi ADCIN1/IOPA1 Tüm ünite resetleri tarafından dijital giriş olarak yapılandırılmıştır. Bu uç tüm cihaz resetleri tarafından dijital giriş olarak kabul edilir.
ADCIN9/IOPA2	90	G/Ç	İki yönlü dijital , ilk ADC'nin analoggirişi ADCIN9/IOPA2 Tüm ünite resetleri tarafından dijital giriş olarak yapılandırılmıştır. Bu uç tüm cihaz resetleri tarafından dijital giriş olarak kabul edilir.

T1PWM/T1CPM/IOPB3 105	G/Ç/Z	İki yönlü dijital G/Ç. Zamanlayıcı 1 karşılaştırma çıkışı. Bu pin maskelenmemiş \overline{PDPINT} aktif low olduğunda yüksek empedans duruuna geçer . Bu uç tüm cihaz resetleri tarafından dijital giriş olarak kabul edilir.
T2PWM/T2CMP/IOPB4 106	G/Ç/Z	İki yönlü dijital G/Ç. Zamanlayıcı 2 karşılaştırma çıkışı. Bu pin maskelenmemiş \overline{PDPINT} aktif low olduğunda yüksek empedans duruuna geçer . Bu uç tüm cihaz resetleri tarafından dijital giriş olarak kabul edilir.
T3PWM/T3CMP/IOPB5 107	G/Ç/Z	İki yönlü dijital G/Ç. Zamanlayıcı 3 karşılaştırma çıkışı. Bu pin maskelenmemiş \overline{PDPINT} aktif low olduğunda yüksek empedans duruuna geçer . Bu uç tüm cihaz resetleri tarafından dijital giriş olarak kabul edilir.

TMRDIR/IOPB6	108	G/Ç	İki yönlü dijital G/Ç.Zamanlayıcılar için yön sinyali . Bu pin Low ise sayma yönü yukarı doğru ve High ise aşağı doğru olacaktır.
TMRCLK/IOPB7	109	G/Ç	İki yönlü dijital G/Ç. Genel amaçlı zamanlayıcılar için harici saat girişi. Bu uç tüm cihaz resetleri tarafından dijital giriş olarak kabul edilir.
ADCSOC/IOPCO	63	G/Ç	İki yönlü dijital G/Ç. ADC'nin işleme başlamasını sağlayan harici pin. Bu uç tüm cihaz resetleri tarafından dijital giriş olarak kabul edilir.
CAP1/QEP1/IOPC4	67	G/Ç	İki yönlü dijital G/Ç. Örnekleme 1 ya da QEP1 giriş pini. Bu uç tüm cihaz resetleri tarafından dijital giriş olarak kabul edilir.
CAP2/QEP2/IOPC5	68	G/Ç	İki yönlü dijital G/Ç. Örnekleme 2 ya da QEP2 giriş pini. Bu uç tüm cihaz resetleri tarafından dijital giriş olarak kabul edilir.

CAP3/IOPC6	69	G/Ç	İki yönlü dijital G/Ç. Örneklemeye 3 giriş pini. Bu uç tüm cihaz resetleri tarafından dijital giriş olarak kabul edilir.
CAP4/IOPC7	70	G/Ç	İki yönlü dijital G/Ç. Örneklemeye 4 giriş pini. Bu uç tüm cihaz resetleri tarafından dijital giriş olarak kabul edilir.
XF/IOPC2	65	G/Ç	İki yönlü dijital G/Ç. Harici bayrak çıkış pini. XF genel amaçlı çıkış ucu ya da birden fazla işlemci kulaanılan uygulamalarda diğer işlemciye sinyal göndermek için kullanılır. Bu uç tüm cihaz resetleri tarafından dijital giriş olarak kabul edilir.
\overline{BIO} /IOPC3	66	G/Ç	İki yönlü dijital G/Ç. Bu pin dallanma kontrol girişi . \overline{BIO} , BIOZ komutu tarafından yönlendirilir. Eğer \overline{BIO} low seviyesinde ise MİB dallanma komutunu icra eder. Bu pin kullanılmazsa High seviyesine çekilmelidir.

CLKOUT/IOPC1	64	G/Ç	İki yönlü dijital G/Ç ve Saat çıkışı. Bu çıkış SYSCR kütüğündeki CLKSCR biti tarafından seçilir. Bu pin Power-on reseti tarafından DSP saat çıkışı olarak yapılandırılmıştır.
Seri Haberleşme Arayüzü (SCI) ve Bit G/Ç pinleri			
SCITXD/IO	44	G/Ç	SCI asenkron seri port bilgi aktarımı ya da genel amaçlı iki yönlü G/Ç pini olarak kullanılır. Bu uç tüm cihaz resetleri tarafından dijital giriş olarak kabul edilir.
SCIRXD/IO	43	G/Ç	SCI asenkron seri port bilgi alımı ya da genel amaçlı iki yönlü G/Ç pini olarak kullanılır. Bu uç tüm cihaz resetleri tarafından dijital giriş olarak kabul edilir
Seri Çevresel Arayüzü (SPI) ve Bit G/Ç pinleri			
SPISIMO/IO	45	G/Ç	SPI yardımcı girişi , ana çıkışı ya da genel amaçlı iki yönlü G/Ç pini olarak kullanılır. Bu uç tüm cihaz resetleri tarafından dijital giriş olarak kabul edilir.

SPISOMI/IO	48	G/Ç	SPI ana girişı , yardımcı çıkışı ya da genel amaçlı iki yönlü G/Ç pini olarak kullanılır. Bu uç tüm cihaz resetleri tarafından dijital giriş olarak kabul edilir.
SPICLK/IO	49	G/Ç	SPI saat girişı ya da iki yönlü dijital G/Ç olarak kullanılabilir. Bu uç tüm cihaz resetleri tarafından dijital giriş olarak kabul edilir.
SPISTE/IO	51	G/Ç	SPI yardımcı işaret gönderimi kontrol pini ya da ki yönlü dijital G/Ç olarak kullanılabilir. Bu uç tüm cihaz resetleri tarafından dijital giriş olarak kabul edilir.

Karşılaştırma Sinyalleri			
PWM1/CMP1	94	O/Z	Karşılaştırma üniteleri ya da PWM birimleri olarak kullanılabilir. Bu pinin nasıl kullanılacağına ACTR kütüğü karar verir. CMP1-CMP6 , \overline{PDINT} aktif low olduğunda ve \overline{RS} aktif low olduğunda yüksek empedans seviyesine gelir.
PWM2/CMP2	95		
PWM3/CMP3	96		
PWM4/CMP4	97		
PWM5/CMP5	98		
PWM6/CMP6	99		
Kesme ve Diğer İşaretler			
\overline{RS}	35	G/Ç	Reset girişi. Bu pin TMS320C240'ın komut icraasını durdurmasını ve PC= 0 olmasını sağlar . \overline{RS} High seviyesine getirildiğinde komutların icraası program hafızasının sıfırcı adres biriminden itibaren başlar. Bu pin bir çok kütük ve durum bitini etkiler
MP/ \overline{MC}	37	I	Mikroişlemci/mikrobilgisayar çalışma seçim biti. Eğer pin Low seviyesinde ise dahili program hafızası ve High seviyesinde ise harici program hafızası kullanılır.

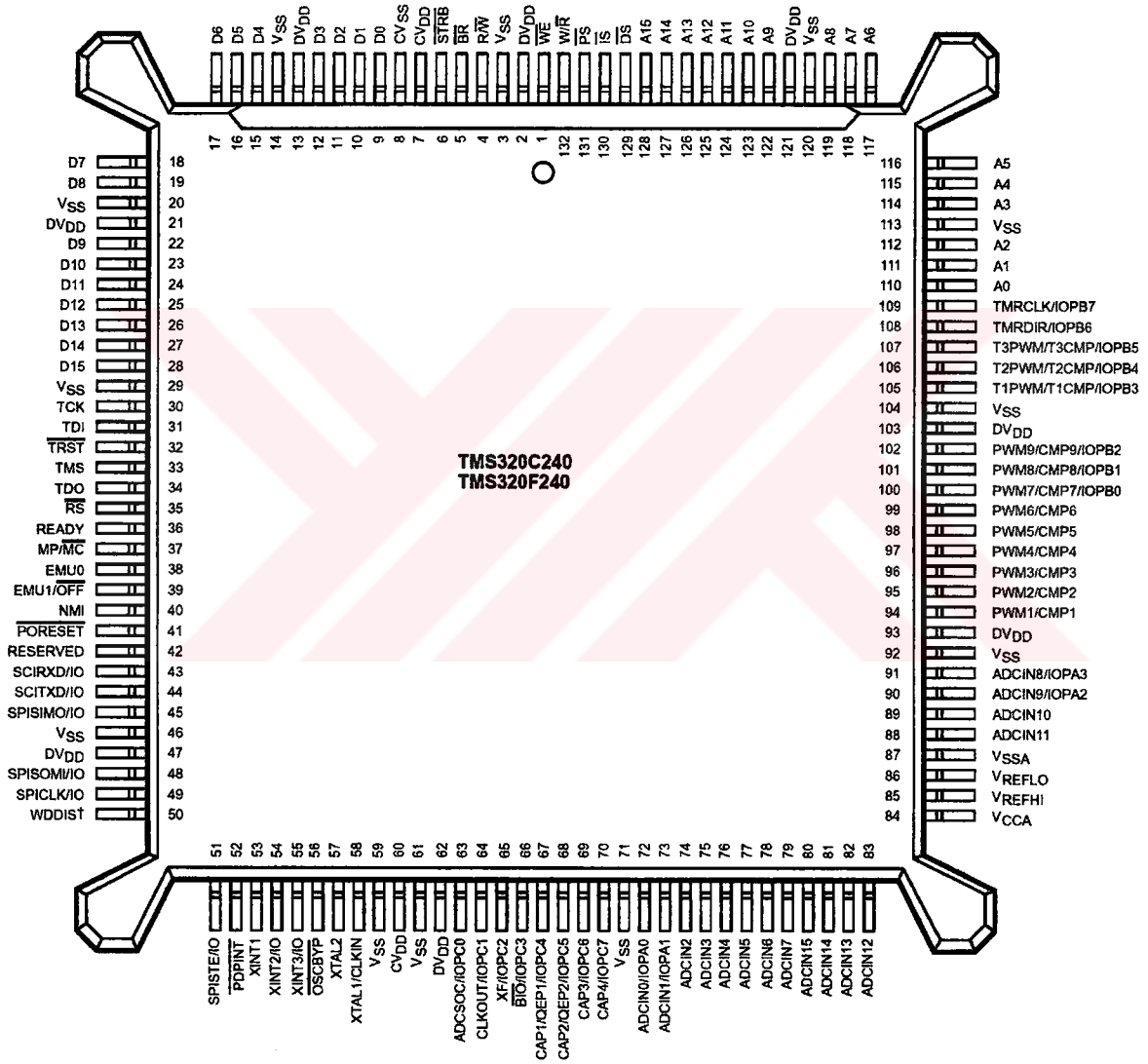
NMI	40	I	Maskelenemez kesme giriş pini. NMI aktif olduğunda yonga durum kaydedici kütüğü INTM bitinin konumuna bakmaksızın kesme programını icraa etmeye başlar. Bu pinin polaritesi programlanabilir.
$\overline{PORESET}$	41	I	Power-on reset pini. Bu pin TMS320C240'ın komut icraasını durdurur ve Program sayacını sıfıra çeker. $\overline{PORESET}$ pini High seviyesine getirildiğinde, komutların icraası sıfırcı program belleği adresinden itibaren yeniden başlar. Ayrıca bu pin \overline{RS} ucu ile aynı kütük ve durum bitlerini etkiler. Ek olarak $\overline{PORESET}$ PLL kontrol kütüğünü de başlangıç konumuna getirir.
XINT1	53	I	1 numaralı Harici kullanıcı kesme pini
XINT2/IO	54	G/Ç	2 numaralı Harici kullanıcı kesme pini ve genel amaçlı iki yönlü G/Ç. Bu pin tüm yonga resetleri tarafından dijital giriş olarak kabul edilir.

XINT3/IO	55	G/Ç	3 numaralı Harici kullanıcı kesme pini ve genel amaçlı iki yönlü G/Ç. Bu pin tüm yonga resetleri tarafından dijital giriş olarak kabul edilir.
\overline{PDPINT}	52	I	Maskelenebilir güç-sürme koruma kesme pini. Eğer \overline{PDPINT} maskelenmemiş ise ve aktif low ise zamanlayıcı karşılaştırma çıkışları derhal yüksek empedans seviyesine gelir.
Saat İşaretleri			
XTAL2	57	O	PLL osilatör çıkış pini. Yonga PLL modunda ise bu pin kristalin bir ucuna bağlanır. Osilatörün Bypass edildiği çalışma modunda bu uç boş olarak da bırakılabilir. Bu pin EMU/\overline{OFF} low seviyesinde olduğunda yüksek empedans konumuna geçer.
XTAL1/CLKIN	58	I/Z	PLL osilatör giriş pini. Yonga PLL modunda ise bu pin kristalin diğer ucuna bağlanır. Bypass modunda ise pin harici bir saat darbe üreticine bağlanabilir.

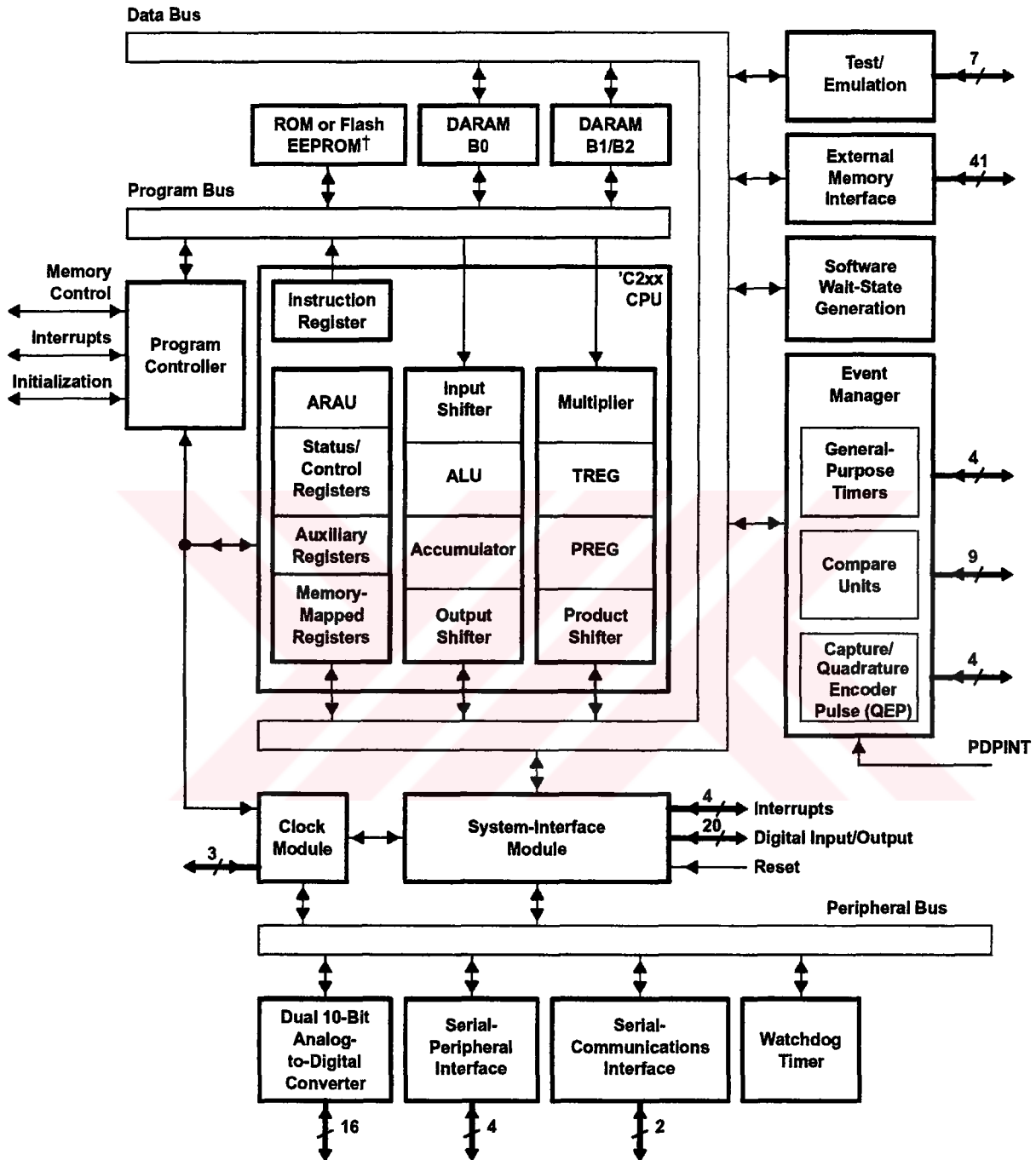
\overline{OSCBYP}	56	I	Eğer bu pin low seviyesinde ise osilatör bypass edilir.
Kaynak İşaretleri			
CV_{SS}	8	I	Dijital çekirdek toprak referansı
3			
14			
20			
29			
46			
V_{SS}	59	I	Dijital lojik toprak referansı
61			
71			
92			
104			
113			
120			
V_{SSA}	87	I	Analog toprak referansı

2			
13			
21			
DV _{DD}	47	I	Dijital G/Ç lojik besleme kaynağı
62			
93			
103			
121			
V _{CCA}	84	I	Analog besleme kaynağı
V _{REFHI}	85	I	ADC Analog gerilim üst referansı
V _{REFLO}	86	I	ADC Analog gerilim alt referansı
Test İşaretleri			
TCK	30	I	IEEE standart test saat işareti giriş pini.
TDI	31	I	IEEE standart test veri giriş pini
TDO	34	I	IEEE standart test veri çıkışı. Daha önceden seçilmiş olan kütüğün içeriği TCK pininden uygulana işaretin düşen keanrında bu pin aracılığıyla dışarı aktarılır.
TMS	33	I	IEEE standart test modu seçim biti

**EK 2 : TMS320C240 YONGASININ TERMİNAL YERLEŐİMİ VE BLOK
DİYAGRAMI**



Şekil EK2.1 : TMS320C240 yongasının terminal bağlantı uçları



Şekil EK2.2 : TMS320C240 yongasının blok diyagramı

ÖZGEÇMİŞ

Doğum tarihi	27.09.1978	
Doğum yeri	İstanbul	
Lise	1991-1994	Bostancı Hayrullah Kefoğlu Lisesi
Lisans	1994-1998	Yıldız Üniversitesi Elektrik – Elektronik Fak. Elektrik Mühendisliği Bölümü
Yüksek Lisans	1999-2001	Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Elektrik Müh. Anabilim Dalı

Çalıştığı kurumlar

1998-1999 AKTAŞ elektrik A.Ş. Şebeke Bakım ve Onarım Müh.

1999-Devam ediyor Yıldız Teknik Üniversitesi
Elektrik – Elektronik Fakültesi
Araştırma Görevlisi