

39798

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

39798

## DSP TABANLI BİR SİSTEM İLE ANAHTARLAMALI RELÜKTANS MOTORUNUN KONTROLU

YÜKSEK LİSANS TEZİ

Müh. İzel ÇİPRUT

Tezin Enstitüye Verildiği Tarih : 13 Haziran 1994

Tezin Savunulduğu Tarih : 1 Temmuz 1994

Tez Danışmanı : Prof. Dr. R. Nejat TUNÇAY

Diğer Jüri Üyeleri : Prof. Dr. Emin TACER

Yrd. Doç. Dr. A. ÖZDEMİR

**Y.İ. YÜKSEK LİSANS ENSTİTÜSÜ**  
**DOKÜMANTASYON MERKEZİ**  
TEMMUZ 1994

## ÖNSÖZ

Tez çalışmasını gerçekleştirerek bu konuda araştırma yapmamı sağlayan ve değerli yardımlarını esirgemeyen Sayın Prof. Dr. R. Nejat TUNÇAY 'a, en derin saygılarımla teşekkürlerimi sunarım. Tezin hazırlanması sırasında destek olan tüm arkadaşlarıma teşekkürü bir borç bilirim.

Haziran 1994, İstanbul

İzel ÇİPRUT

## İÇİNDEKİLER

|                                                                                                   |    |
|---------------------------------------------------------------------------------------------------|----|
| ÖZET .....                                                                                        | v  |
| SUMMARY.....                                                                                      | vi |
| BÖLÜM 1. GİRİŞ .....                                                                              | 1  |
| 1.1. Dijital Denetleyici Yapılarının Analog Denetleyici<br>Yapıları İle Karşılaştırılması : ..... | 6  |
| BÖLÜM 2. MİKROİŞLEMCİLER VE KONTROL SİSTEMLERİNDEKİ<br>UYGULAMALARI.....                          | 10 |
| 2.1. Mikroişlemcilerin Yapısı : .....                                                             | 11 |
| 2.2. Mikroişlemcilerin Kontrol Sistemlerinde<br>Uygulamaları : .....                              | 16 |
| 2.2.1. Donanım Tasarımı : .....                                                                   | 19 |
| 2.2.2. Yazılım Tasarımı : .....                                                                   | 22 |
| BÖLÜM 3. DİJİTAL İŞARET İŞLEME VE DİJİTAL DENETLEYİCİ<br>TASARIMI .....                           | 24 |
| 3.1. Dijital Yapılarda Sayı Formatı : .....                                                       | 26 |
| 3.2. Sabit Nokta ve Kayan Nokta Aritmetiği : .....                                                | 29 |
| 3.2.1. Sabit Nokta Aritmetiği : .....                                                             | 29 |
| 3.2.2. Kayan Nokta Aritmetiği : .....                                                             | 30 |
| 3.3. Sonlu Kelime Genişliği Etkileri : .....                                                      | 33 |
| 3.3.1. Taşma : .....                                                                              | 35 |
| 3.3.2. Yuvarlatma ve Kesme : .....                                                                | 36 |
| 3.3.3. Ölçeklendirme : .....                                                                      | 38 |

|                                                                     |            |
|---------------------------------------------------------------------|------------|
| <b>BÖLÜM 4. DSP GENEL YAPISI VE UYGULAMA ALANLARI.....</b>          | <b>41</b>  |
| <b>4.2. DSP Mimarisini Oluşturan Üniteler.....</b>                  | <b>48</b>  |
| 4.2.1. Çarpım Ünitesi : .....                                       | 48         |
| 4.2.2. Çarpım - Akümülatör Ünitesi ( MAC ) : .....                  | 49         |
| 4.2.3. Aritmetik - Lojik İşlem Ünitesi ( ALU ) : .....              | 51         |
| 4.2.4. Yığın Kaydırma Ünitesi ( BS ) : .....                        | 53         |
| 4.2.5. Veri Adres Generatörü Ünitesi ( DAG ) : .....                | 55         |
| 4.2.6. Komut Sıralama Ünitesi ( SEQ ) : .....                       | 57         |
| <b>BÖLÜM 5. ÖRNEK DSP SİSTEMLERİNİN KARŞILAŞTIRILMASI.....</b>      | <b>60</b>  |
| <b>BÖLÜM 6. DSP ve MİKROİŞLEMCİLERİN KARŞILAŞTIRILMASI .....</b>    | <b>77</b>  |
| <b>BÖLÜM 7. DSP UYGULAMA ALANLARI .....</b>                         | <b>85</b>  |
| <b>BÖLÜM 8. ANAHTARLAMALI RELÜKTANS MOTORUNUN KONTROLÜ.....</b>     | <b>100</b> |
| 8.1. Anahtarlamaalı Relüktans Motoru : .....                        | 101        |
| 8.2. Güç Elektroniğı Devresi ve Sürücü Devrenin Özellikleri : ..... | 103        |
| 8.3. Kontrol Sistemi : .....                                        | 107        |
| <b>BÖLÜM 9. SONUÇ .....</b>                                         | <b>116</b> |
| <b>EK-A : .....</b>                                                 | <b>118</b> |
| <b>KAYNAKLAR : .....</b>                                            | <b>119</b> |
| <b>ÖZGEÇMİŞ .....</b>                                               | <b>122</b> |

## ÖZET

DSP sistemlerinin kontrol alanında kullanılmaya başlanması, sistem tasarımı ve uygulama alanına yeni bir boyut kazandırmıştır. Dijital kontrol sistemlerinde denetleyici yapısı olarak mikroişlemcilerin kullanılmasının getirdiği üstün performans, DSP 'nin uygulama alanına girmesi ile kullanımının hızla artmasına neden olmuştur. Sistemlerden beklentiler arttıkça tasarım mühendisleri, yeni elemanların getirdiği yararları kullanıp, isteklere cevap verme yolunu seçmişlerdir. Motor kontrolünde, yüksek performanstan enerji tasarrufuna kadar geniş beklentiler ancak yeni kontrol algoritmaların kullanılması ile karşılanabilmektedir. Bu gibi kontrol algoritmalarını gerekli koşulları sağlayarak gerçekleştiren denetleyiciler ancak DSP işlemcileri ile gerçekleştirilebilir.

Mikroişlemci kontrolünün güç elektroniğinin bütün alanlarına girmesi ile daha önce yapı olarak bilinmesine rağmen kullanıma geçirilemeyen motorların tekrar gündeme gelmesi gerçekleşmiştir. Bu motorların kontrol alanındaki gelişmeleri izleyerek, yaygın olarak kullanılan motorlara alternatif oluşturacakları açıktır. Anahtarlamalı relüktans motoru, bu motorlar arasında değişik yapısı, kolay tasarımı, ucuz maliyeti ile bir yer edinmiştir. Kontrolünün doğru yapılması durumunda, motorun elektriksel ve mekanik bakımdan üstün özellikleri bulunmaktadır. Sistemin doğru kontrol edilmesi, ancak yeterince hızlı ve doğru işlem yapabilen işlemciler ile mümkündür.

Bu çalışmada, DSP sistemi kullanılarak dijital bir denetleyici tasarımında dikkate alınması gereken koşullar ve mikroişlemciler ile DSP sistemlerinin karşılaştırılması kontrol uygulamaları ön planda olmak üzere gerçekleştirilmiştir. Dijital denetleyici yapısı tasarımında göz önüne alınması gereken noktalar incelenerek kontrol algoritmalarının gerçekleştirilmesi sırasında karşılaşılan sorunlar irdelenmiştir. Anahtarlamalı relüktans motorunun kontrolü için güç elektroniği devresi tasarlanarak, gereken sürücü devre tasarımı yapılmış ve her iki kontrol sistemine uygun yapı kurulmuştur.

## **SUMMARY**

### **DSP BASED CONTROL OF SWITCHED RELUCTANCE MOTOR**

In recent years, the revolution on the electronics, brings the new devices into the area of control. Opportunities have grown rapidly over past years to solve the problems of control applications. All the control systems of a few years were designed by a general purpose microcomputer. But, VLSI design technology brings DSP processor to digital signal processing area with a powerful architectural design. High performance of DSP make it possible to implement a wide range of digital control algorithms for sophisticated applications.

A control system is built by a controller and the circuits which provide to connect these parts together. The control system must achieve a desired output to command a process. In the control applications of power electronics and motors, the system is preferred to be a closed loop. The closed loop system takes the reference value and feedback value which came from the sensors and actuators. The controller must compute a desired response from the entire system.

The controller can be either analog or digital. The function is to ensure the system to be stable in all conditions and respond as quick as possible. Analog controller can be produced with operational amplifiers and passive circuit elements. Analog controllers continuously process a signal and can be used for very high bandwidth systems. The disadvantages of this controller is hard wired design, suffering from temperature drift and component aging. The system is flexible only for simple design approaches. With the new control algorithms like adaptive or fuzzy control, it is impossible to design a controller using analog devices.

The high performance and reliability of microcomputers put them directly to the control applications. The programmable solution and flexibility is the main advantages of the digital controller. It is easy to make upgrade and maintain design investment. To control nonlinear systems with sophisticated techniques from modern control algorithm is possible. Digital controller is not affected by environmental conditions and shows precise behavior. The main problem is that, the processes in nature is analog. So to control these processes with a digital controller, the signals must convert to a kind that both parts understand each other. The digital controller require A/D (analog to digital) and D/A (digital to analog) converters to obtain control of real world data. The accuracy of the

approximation determines the performance of the digital controller. The conversion between two system can provide by sampling the signal at discrete time intervals. This limits the bandwidth that can be handled by the controller. There will be always an error because of the loss of information at the conversion cycle. To avoid this error the resolution or word length of the processor must be increased.

Digital controller is built onto a general purpose processor and peripheral devices to make the system flexible for usage in different applications. The choice of the processor is the main problem to determine the behavior of control system. Some tradeoffs have to be made in selecting a processor. The poor performance of digital system is caused by the wrong selection of the processor. In practical area, speed of execution, accuracy and cost determine the choice of processor. Architecture, performance and peripheral integration is the factors which must be considered.

Processing signals in real time raises numerical problems, and to ensure the performance of the controller acceptable, they must resolve these numerical problems correctly. The architecture of the processor must provide all the requirements of the controlled system. Because of the inadequate processor architecture, quantization, truncation, limit cycles and overflow handling errors appear, and several techniques developed to preset these errors.

Quantization effect is result by representing a signal in discrete time. The analog signals are continuous so to compute these signals quantized magnitude levels must be used. The system must support a large word length and scaling shifters to provide the resolution and dynamic range needed. If the gain coefficients and signals are time varying and have large dynamic range, floating point processors must be used. Scaling can be used to minimize the effects of quantization.

The arithmetic operations always produce an error because of the finite word length. The processor takes the value, compute the operation and then throw away bits when necessary. Because of the finite amount of storage length, the processor can not produce a result using all bits, so it made an approach to make the result understandable to computational units. Rounding and truncation is two ways to do this. If the processor compute  $16 \times 16$  bit multiplication, the result will be 32 bit length. To store and send the result, needs 32 bit bus length and registers. If only 16 bit is available, least meaningful 16 bits will be thrown away. Truncation will throw away before storage, and rounding will round and then throw away before store the result. Rounding makes the result closer to the real value, so it is better than truncation. There are several ways to minimize the effects of rounding and truncation. Limit cycles can produced from

quantization and truncation. Insufficient resolution of the output causes to oscillate around the actual value without being able to reach it.

The control system can use fractional arithmetic or integer arithmetic. The binary arithmetic is used for representation of numbers. Two's complement is the better alternate which is used by digital signal processing elements. Fraction numbers are better when the algorithm is multiplication based. If the highest number is 0.99, it is impossible to occur an overflow. But addition calculation can cause overflow. The registers which are effected by overflow, wrap around and change magnitude from most positive to most negative. The processor must take care of with these registers to avoid wrong control behaviour. The saturation mode is provided to prevent the content of registers to wrap around and change signal when an overflow occurs. This is done automatically in DSP.

The overflow can be avoided with increasing the register size, but it is better solution to put some guard bits to protect the result from overflow. To optimize the dynamic range, data arrays are kept scaled so that the largest value fill all the bits of the data word. Precision and dynamic range of numbers must be balanced to find an optimum solution. The processor must finish processing the signal before the arrival of the next sample or the information will be lost. Also peripheral integration is important from a system cost, ease of design and interface.

DSP is the digital signal processor which provide new features to control system applications. DSP is designed by using Harvard architecture, while the microprocessor is designed by Von Neumann architecture. The most important part, the units are independent so can compute same time in parallel. The second main difference is, the memory of data and program is independent from each other so to compute and to execute can be done in single cycle. All functions are performed internally in hard wired logic so that it takes a single cycle to execute most of the mathematical operations. The hardware multiplier can multiply the contents in a single instruction cycle thus the speed is excellent comparing with microprocessors. The multiple bus architecture provides better computing speed performance. High sampling rates can be provided so the precision improve better. DSP gives the real time application flexibility and performance which is predictable, accurate, repeatable and adaptable. Computational speed matched from unit to unit so no bottlenecks develop. To remove bottlenecks pipelining and parallel operation can be used.

Bottleneck is a computational delay produced because of the data can not use the buses and units efficiently. In pipelining, the operations broken into subunits and these subunits compute part of operation

independently. Parallel operation solves the problem from performing operation in different units in the same time. These features makes the computational speed much faster and improves architecture for higher throughput. Accuracy issues is the main part of a processing unit. So to provide better accuracy, the architecture can be designed for avoid numerical errors. The shared memory for both instructions and data and single instruction data bus is significant bottleneck and will cause errors in real time applications. Architecture, accuracy and speed interact, so an efficient and optimized control solution can only be find with taken care of this issues. The fundemental primitive operation in numerical computation is multiply accumulate step and it is desired to do this in a single clock cycle. DSP can do it with pipelining and parallel operation speed processing by doing more in fewer cycles. The desirable features of a DSP is bus width, expanded arithmetic, data-address generation, instruction sequencing and scaling. To minimize the effect of numerical operations the units must have additional guard bits, bus width must large enough to operate correctly with double precision option.

The basic units of DSP system is as follows : Multiplier-accumulator (MAC), Arithmetic logic unit (ALU), Barrel shifter (BS), Data address generator (DAG), Instruction sequencer (SEQ). These units perform rapid execution of arithmetic and logic operations and provides protection against errors with diffirent solutions. All these units can operate in parallel and connect to buses which gives the result in a single cycle. The main disadvantage of DSP is the memory area. For providing speed only limited memory area is put on chip. Thus, large amount of programs can not been stored into memory.

There are several differences between microcomputer and DSP systems. Both system has its own advantages and disadvantages. The designer must learn the options of controlled system to make the right decision between DSP and microcomputer. Speed, memory area, software support, peripheral devices and cost will be the main points to select which processor must be chosen.

In applications, especially using a sophisticated control algorithm, math operations makes longer the execution time. Microcomputers use look up table to decrease the execution time. But DSP is produced to opareta at this conditions, so it is better to use numerical methods to solve a function. However, there is no enough memory to create look up tables and store program. This is just a sample how DSP and microcomputer solve main problems.

Switched reluctance motor (SRM) is one of the alternative motors to asynchronous and DC motors. SRM is a motor which both stator and rotor is salient pole. There is only winding on stator to create flux, no winding

on rotor. The system must operate normally in closed loop control, with shaft position sensors and encoders. It is necessary to know which stator phase will be excited in a time period. The commutation of the phase currents with rotor position must be in synchronism. The motor can be 8/6, 6/4, 12/8 or another stator rotor pole numbers with different phase numbers. The torque is produced by moving rotor to position where the inductance of excited winding is maximized. The aligned position of motor is defined as, stator and rotor poles are aligned exactly, and airgap is minimum. The magnetic reluctance of flux path is at its lowest, so inductance is maximum. The control of the phase currents gives the best performance. The requirements of control are to produce smooth torque with minimum ripples, good stability and smooth reversing capability. Different kinds of power electronic circuits are defined to take maximum performance. The control of system must be in real time, because torque, current, speed and firing angles react as nonlinear functions and vary with load conditions.

In this thesis, the survey of digital controller design has been made. The microcomputer and DSP systems are introduced from a practical point of view, architecture, main units, differences and applications of the systems are explained. The switched reluctance motor control design is explained with power electronic circuit, gate drive circuit and controller circuit realized with microcomputer and DSP, respectively.

## BÖLÜM 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 alanına girmesi, 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 yapıya bürünmesi, analog devreler ile gerçekleştirilmesini imkansızlaştırmıştır. Gerçek zamanda işleyen algoritmaların motor kontrolüne uygulanmaları, sistemlerin beklentilerini arttırmış, daha az enerji ile daha büyük sistemlerin kontrolünün gerçekleştirilmesi arzulanmıştır.

Motor kontrol sistemlerinde, mikroişlemcilerin kullanımı ile, kapalı çevrim kontrole geçilmiştir. Böylece sistemin tam kontrolünün yapılabilmesi sağlanmıştır. Kapalı çevrim kontrolü sağlandıktan sonra sistemden beklentiler artmıştır. Örneğin, motorun sadece hız kontrolü yapmasıyla 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 karmaşıklıklaştıkça sistemin parametreleri artmış, bu da kontrol algoritmasının matematik 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 değişmesi durumunda hala belirli bir hata payı söz konusu olmaktadır. Bu yüzden kontrol sistemleri yeni işlemcilerin arayışına başlamıştır.

İlk olarak görüntü ve ses uygulamalarında kullanılan DSP'ler kontrol sistemlerine yeni bir bakış açısı kazandırmıştır. İşlem yapma kapasiteleri, üstün matematik işlem hacmi ve en önemlisi işleme hızı, DSP'nin kullanım alanının hızla genişlemesini sağlamıştır. Mimari yapı olarak klasik mikroişlemcilerden farklı olan DSP sistemleri, matematik işlemleri donanım olarak gerçekleştirerek, üstün bir hız özelliği kazanmıştır. Bütün bu özelliklerin kontrol uygulamalarında kullanılmaya başlanması, dijital sistemlerin vazgeçilmez bir denetleyici olmasını sağlamıştır. Analog denetleyicilerin üstün olduğu alanlarda, aynı özellikleri sağlayarak 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. Mikroişlemcilerin toplama işlemini donanım olarak gerçekleştirmesine rağmen, çarpma işlemi, kullanıcıya gözükmeyen yazılım ile yapılmaktadır. Bu da, sistemin toplam işleyiş süresinin artmasına yol açmaktadır. DSP sistemlerinde, işlem yapan ünitelerin birbirinden bağımsız olması ve bellek birimlerinin ayrı olması, işlemlerin donanım yapısı olarak yapılmasını yani bir komut saat çevriminde tamamlanmasını sağlamaktadır. DSP'nin üstün hız özelliği bu şekilde sağlanmaktadır.

DSP sistemlerinin kontrol alanında uygulanmaya başlaması, diğer alanlara göre geç olmuştur. Bunun nedeni, tasarlanan ve uygulamaya

konan sistemde maliyet faktörünün hala en önemli etken olmasıdır. Mikroişlemcilerin geçen yıllarda daha ucuz olmasına rağmen, yarıiletken teknolojisindeki son gelişmeler, DSP `nin de aynı oranda ucuzlamasına neden olmuştur. Bu aşamadan sonra, DSP ile tasarlanan kontrol sistemlerinin daha çok uygulama alanı bulacağı açıktır.

Güç elektroniği ve motor kontrolunda, kontrol yapısının seçimi bir çok faktöre bağlıdır. Tasarlanan sistemin ihtiyaçlarının mikroişlemci veya DSP ile sağlanması durumunda, hangisinin tercih edilmesi gerektiğini sistemin kendisi belirleyecektir. İki yapının da üstün olduğu 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 elektroniği 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 makina ve anahtarlama 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, anahtarlama 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  $q$  eksen takımına indirgenmiş eşdeğer devresi çıkarılarak kontrolü mümkün değildir. Bu yüzden, değişkenlerin gerçek zamanda ve gerçek değerler ile işlemesi gerekmektedir.

ARM, kontrol yapısı gerektiği gibi kurulduğunda diğer motor türlerine göre daha üstün özellikleri olan bir yapıdır. Aynı şekilde yanlış bir kontrol algoritması veya değişkenin 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 algoritmanın sağlaması gereken koşullar sistem tasarımının önemli bir bölümüdür.

Bu çalışmada; ilk bölümde dijital denetleyici yapılarının özellikleri ve analog denetleyiciler ile karşılaştırılması ele alınmıştır. Analog yapıların üstün olduğu alanlarda dijital sistemlerin iyileştirilmesi ile performansın yükseltilmesinin yolları gösterilmiştir.

Dijital denetleyici yapılarının özellikleri açıklandıktan sonra, yapısının temel birimi olan işlemcilerin mimari ve yapısal özellikleri, ikinci bölümde ele alınmıştır. Temel mimari özelliklerinin mikroişlemcilerle kazandırdığı yararlar, uygulama alanında kullanımı sırasında karşılaşılan sorunlar ve çözümleri bu bölümün alt konularıdır.

Üçüncü bölümde, genel anlamda dijital denetleyici yapısının tasarımı incelenmiştir. Dijital işaret işlemenin temelleri ele alındıktan sonra, işaretlerin işlenmesi sırasında karşılaşılan sorunlar, dijital sistemin işareti değerlendirirken karşılaştığı sayısal hatalar, ve bu sayısal hatadan dolayı oluşan davranış bozukluğunun düzeltilmesinin yolları incelenmiştir. Denetleyici sisteminin tasarımı sırasında bu tür hataların ihmal edilecek düzeye indirilmesi temel problemlerden birini oluşturmaktadır. Bunun için,

matematik işlemlerin sonuçlarını düzelten bir algoritma yapısı kurulmalıdır. Dijital denetleyici sisteminin tasarımı incelendikten sonra, bu yapının kurulmasında mikro işlemcilerin zayıf kaldığı alanlar irdelenmiş ve yeni bir sistemin gerekliliği ortaya konmuştur.

Dördüncü bölümde, DSP sistemlerinin genel tanımı, özellikleri ve mimari yapısı incelenmiştir. Dijital denetleyici sistemine kazandırdığı özellikler ve eksiklikleri araştırıldıktan sonra, mimari yapısını oluşturan üniteler ayrı ayrı incelenmiştir. DSP 'nin hızlı işlem yapmasını sağlayan bağımsız mimari yapılar, temel özellikleri ve işleme yapıları temel alınacak şekilde incelenmiştir.

Beşinci bölümde, DSP sistemlerinin temel özellikleri ele alınarak, üretilmiş DSP sistemlerinin temel yapısı ve mimari özellikleri, sabit nokta ve kayan nokta işlemcilerinin gerçek sistemler üzerinde açıklanması gerçekleştirilmiştir.

Altıncı bölümde, DSP ve mikro işlemci sistemleri genel yapı ve uygulamaya yönelik olarak karşılaştırılmıştır. Gerçek sistemleri karşılaştıran bilgiler ile sistem tasarımı sırasında yapılacak tercihlere yol gösterilmeye çalışılmıştır.

Yedinci bölümde, DSP 'lerin uygulama alanları, güç elektroniği ve motor kontrolü temel alınacak şekilde irdelenmiştir. Kontrol sisteminin tasarımı sırasında kontrol algoritmasının kurulması aşamaları incelenerek, DSP sistemine uygunluğu incelenmiştir. Kontrol sistemlerinde klasik bir yapı haline gelen PID yapısının DSP ile gerçekleştirilmesi aşamalar halinde incelenmiş ve gerekli algoritma kurulmuştur. Daha kapsamlı bir kontrol olanağı sağlayan adaptif kontrol yapısının temel olarak işlenmesi ve DSP 'de uygulanabilirliği ele alınmıştır.

Sekizinci bölümde ise, gerçekleştirilen sistemin tanıtımı yapılmıştır. Anahtarlamalı relüktans motorunun yapısı, temel özellikleri ve kontrolü incelendikten sonra, güç elektroniği alanında kurulan sistem tanıtılmıştır. Motor kontrolünde gereken yapılar, sıra ile ele alınarak gerekli bilgiler

verilmiştir. Güç anahtarlarının seçimi, kullanım alanları, anahtarları sürmek için gereken sürücü devrenin yapısı, temel özellikleri ele alınmıştır. Daha sonra gerçekleştirilen kontrol sistemi tanıtılmıştır. Motorun kontrolü için gereken akış diyagramı ve algoritma yapısı verilerek, programın temel özellikleri verilmiştir. Kullanılan işlemciler temel özellikleri ile tanıtıldıktan sonra, bu uygulamaya yönelik olarak nasıl kullanıldıkları incelenmiştir.

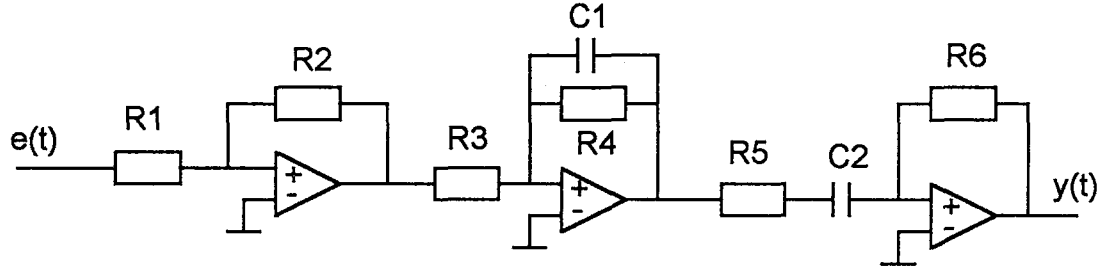
### **1.1. Dijital Denetleyici Yapılarının Analog Denetleyici Yapıları İle Karşılaştırılması :**

Güç elektroniği ve motor kontrolunda kullanılan sistemlerin daha çok parametre üzerinde işlem yapma ve kontrol etme isteği, kullanılmakta olan denetleyici yapılarının yetersiz kalmasına neden olmuştur. Dijital denetleyicilerin gelişimi, bu konudaki büyük bir açığın kapanmasını sağlamıştır. Analog denetleyici, pasif ve aktif devre elemanları kullanılarak sistem parametrelerinin ancak belirli değişim aralığına izin veren yapılardır. Lojik kapılar, işlemsel kuvvetlendiriciler ve pasif devre elemanları ile tasarlanan sistemin kullanımının getirdiği sorunlar, mikroişlemci tabanlı dijital denetleyicilerin gelişimi ile ortadan kalkmış, ancak bu yapılarda kendilerine özgün 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 elemanlarının sıcaklığa çok bağımlı olması, sistemin hatalı davranma olasılığını arttıracaktır.



**ŞEKİL 1.1 : Analog PID Denetleyici Yapısı**

Son yıllarda geliştirilen modern kontrol algoritmalarının analog bir yapı ile gerçekleştirilmesi neredeyse imkansız hale gelmiştir. Kontrol edilen sistem karmaşılaşp, 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 anlaşılamaz 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

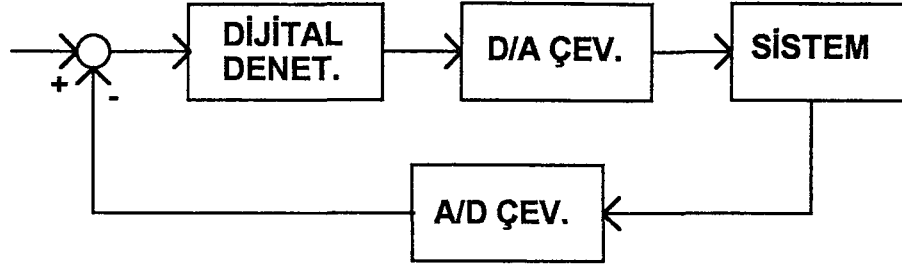
D/A (dijital-analog) çeviriciler yardımıyla dönüştürülmesi gerekmektedir. Bu da, belirli bir bilgi kaybına neden olacaktır.

**TABLO 1.1 : Analog ve Dijital Denetleyicilerin Karşılaştırılması**

|                   | <b>ANALOG DENETLEYİCİ</b>      | <b>DİJİTAL DENETLEYİCİ</b>                    |
|-------------------|--------------------------------|-----------------------------------------------|
| <b>AVANTAJ</b>    |                                |                                               |
|                   | <b>Yüksek doğruluk</b>         | <b>Çevre koşullarından etkilenmeme</b>        |
|                   | <b>Tasarım kolaylığı</b>       | <b>Doğru kesin davranış</b>                   |
|                   | <b>Gerçek zamanda çalışma</b>  | <b>Karmaşık kontrol algoritmalarına uygun</b> |
|                   | <b>Sonsuz örnekleme zamanı</b> | <b>Ek fonksiyonlarına olanaklı</b>            |
| <b>DEZAVANTAJ</b> |                                |                                               |
|                   | <b>Elemanların yaşlanması</b>  | <b>Sayısal Hatalar</b>                        |
|                   | <b>Sıcaklığa bağımlılık</b>    | <b>Örnekleme zam.nın etkisi</b>               |
|                   | <b>Kablolu bağlantı</b>        | <b>Hız problemleri</b>                        |

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.



**ŞEKİL 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.

## BÖLÜM 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.

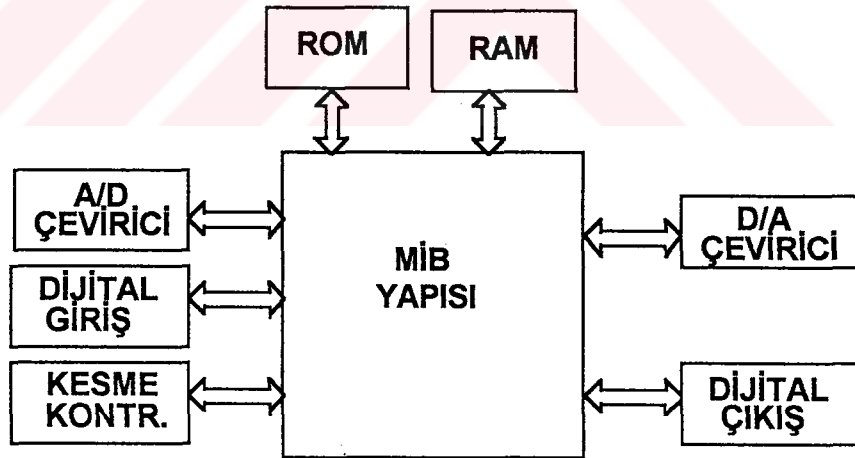
Kontrol sistemlerinde kullanılan mikrodenetleyici yapısı, mikroişlemci biriminin çevre birimleri ile beraber ele alınması ile oluşmaktadır. Fiziksel sistem ile haberleşebilmesi için A/D çevirici birimine ve yazılan programın saklanabilmesi için bellek birimlerine gerek duymaktadır. Program ve veri üzerinde gerekli işlemlerin yapılabilmesi için tekrar tekrar çağırılabilmesi 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 kontroluna 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, herbiri için geliştirilen makine kodu 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.

## 2.1. Mikroişlemcilerin Yapısı :

Bir mikroişlemciden istenen özelliklerin gerçekleşmesini sağlayan mimari yapı şu alt birimlerden oluşmuştur : Merkezi işlem birimi (MİB, Central Processing Unit, CPU), 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 I/O) ve kesme denetleyici (Interrupt Controller).



**ŞEKİL 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

oluşturmuştur. Bu birimler; bellek adres kütüğü, bellek veri kütüğü, akümülatör, aritmetik lojik birimi, 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.

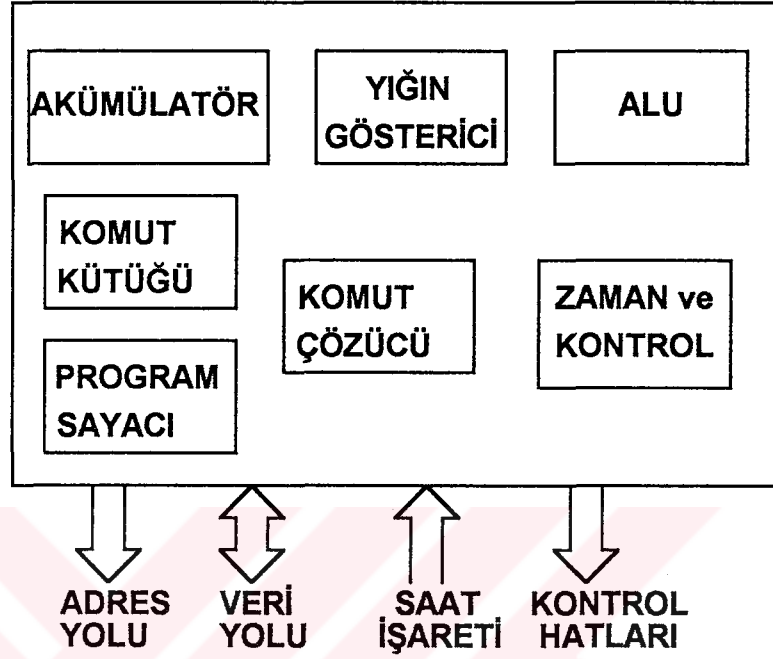
Mikroişlemcilerde program adım adım işlemektedir ve bu adımları oluşturan komutlar bellekte birbiri ardına yerleştirilmiştir. İşlem yapılacak verinin MİB `e ulaşmasını veri yolu sağlamaktadır. Bellek adres kütüğü, MİB ile bellek veya giriş-çıkış birimleri arasında işlenecek verilerin bulunduğu yerin adresini belirtme işlemini yerine getirmektedir. MİB, algorithmadaki 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 yığında 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 yönünü, 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 veride aynı işlemten geçirilerek işlenebilir.



**ŞEKİL 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 göstergeden 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 döneleceği ve hangi komutun işleneceği, yığında saklanan adres bilgisi ile sağlanmaktadır.

Program işleyişinin ard arda 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 işletir. 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 iletilir. 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. Kullanılan 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 programlanıp silinebilir ROM) türünde olabilir.

RAM ( Oku yaz bellek ) türü bellek, program tarafından kullanılacak veriyi saklamakta kullanılmaktadır. Programın geliştirilmesi sırasında, RAM 'da 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

önlenebilir. RAM 'ın 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, 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 kontrolunda geribesleme 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, kontrollu 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 kontrolunda, 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.

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

Sistem tasarımı, kontrol edilecek sistemin analizi ile başlamalıdır. Kontrol parametreleri, gerekli analiz yöntemleri kullanılarak doğru olarak belirlenmelidir. Geliştirilmiş simulasyon 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 gerçekleştirmelidir. Sistemin hangi algoritmayı kullanacağı tamamıyla istenen cevabın özelliklerine bağlıdır. Klasik kontrol sistemlerinde kullanılan PID denetleyici yapısı bir çok sistemin kontrolunda 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 anlamalı 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 kontrolunu sağlamaktadır. Yapı, sistemin nominal parametrelerinin belirlenmesi ve gerçek parametreler ile karşılaştırılarak, 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 bağlıdır. 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 arttıracacağı gibi, hata olasılığını da arttırmaktadır. Seçilen sistemin 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 saklanması 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, 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 ile 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 geliştirme 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 bulundurmaktadır.

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 sistemleride 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 uğruyacaktır. 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 küçültücektir. 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 kullanılabilmektedir. Bu da sistemin çok daha kısa bir sürede geliştirilmesine olanak sağlamaktadır. Yazılan programın doğru işlemesi için, geliştirme ortamı, adım adım işleme, parça işleme gibi özellikler ile tasarıma destek olmalıdır. Geliştirilen programın çok karmaşık olması halinde, program parçalara bölünerek, basitleştirilmelidir.

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 pencereleme 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.

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 iletim hızı, bit oranı ( saniyede iletilen bit sayısı veya baud ) ile orantılıdır ve kanal sayısı ile sınırlıdır.

## BÖLÜM 3

### DİJİTAL İŞARET İŞLEME VE DİJİTAL DENETLEYİCİ TASARIMI

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 yapısını 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 yansıtmaması, işlemcinin kelime genişliği ile doğru orantılıdır.

Dijital denetleyici tasarımı 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 sorunlar da beraberinde getirmektedir. Denetleyicinin performansını belirleyen diğer bir etken de, örnekleme zamanıdır. İşlemci ayırık 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.

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.

Bir işlemcinin hızı genel olarak bu iki birim tarafından belirtilmektedir. Dijital denetleyici yapılarının en çok kullanılan işlemi olan,  $X=A*B+C$  'nın 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.

### 3.1. 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 deyimle 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 İkinin tümleyeni gösterimidir.

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 1 eklemektedir. Son yıllarda gerçekleştirilen işlemcilerin çoğunda İkinin tümleyeni sayı formatı kullanılmaktadır. Bunun nedeni, negatif sayılar ile işlem yapmak için ek bir donanıma gerek duymaması ve 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. İkinci tümleyeni sayı gösteriminde çarpma işlemi, karmaşık yapıda gözükmesine rağmen, geliştirilen değişik algoritmalar ile toplama işlemi kadar kolay ve hızlı gerçekleştirilebilmektedir.

**TABLO 3.1 : İkinci Tümleyeni Sayı Formatı Gösterimi**

|                                                    |                                      |
|----------------------------------------------------|--------------------------------------|
| <b>67 (Des.) = 43 (Heks.) = 01000011 (Bin.)</b>    | <b>Pozitif sayı</b>                  |
| <b>- 67 (Des.) = - 43 (Heks.) = 11000011 (Bin)</b> | <b>Negatif sayı</b>                  |
| <b>01000011</b>                                    | <b>İkinci tümleyeni pozitif sayı</b> |
| <b>10111100</b>                                    | <b>Tümleyeni al</b>                  |
| <b>1</b>                                           | <b>Topla</b>                         |
| <b>10111101 = - 67</b>                             | <b>İkinci tüm. negatif sayı</b>      |

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 biti denilen 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.

**TABLO 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 |

Q15 formatı ile işlem yapıldığında, sonuç yine ondalık sayı formatında olacaktır. Q15 formatından iki sayının çarpımı Q30 formatında bir sayı oluşacaktır. Bu sayıda 2 işaret biti, 30 ondalık sayı biti bulunmakta, tamsayı biti bulunmamaktadır. Sayının Q15 formatında saklanabilmesi için, sola bir kaydırma işlemi ile işaret bitlerinden biri, ve en düşük anlamlı 16 bit seçilmektedir. Bu sayı formatında çarpma işlemi hiçbir zaman taşmaya neden olmamaktadır. Dijital sistemlerde ondalık sayı tercih edilmesinin nedeni, sayıların işlenmesi ve saklanması sırasında en düşük anlamlı m-1 adet bit atılarak sayı anlamlı hale getirilmektedir. Böylece ondalık sayı formatı, sayıların aralığının büyümesi durumunda doğruluğunu korumaktadır. Tamsayı formatında bu durum hatalara neden olmaktadır. Denetleyici yapılarında tamsayı formatı kullanılması, hataları azaltmak amacıyla, ölçeklendirme, öteleme ve buna benzer işlemleri beraberinde getirmiştir.

### 3.2. 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 işlemcileride üretilmiştir. İki tür aritmetiğinde 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.

#### 3.2.1. 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.

$$r = 2^{-B} (-b_{m-1} 2^{m-1} + \sum_{j=0}^{B-1} b_j 2^j) \quad [1]$$

[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.  $-1.0 < r < 1.0-2^{-(m-1)}$  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 sonusunda, 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.

### 3.2.2. 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 tanımı yapılan kayan nokta sayılar, üs ve taban kısımlarından oluşmaktadırlar.  $y=m.10^E$  şeklindeki gösterimde, m taban kısmını, E ise üs kısmını göstermektedir. Örneğin, 4 basamak ile gösterilen bir sayının en büyük değeri,  $9.999 = 0.999.10^9 = 999000000$  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  $y=m2^E$  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 bitide 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 = $(2^{12}-1).2^{15}$ |
| En küçük : | 0000 0000 0001 | 0000 = $1.2^0$             |

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.

**TABLO 3.3 : Sabit ve Kayan Nokta Sayı Formatı Özellikleri**

| Kelime Genişliği:  |                    | 16               | 32                  | 64                  |
|--------------------|--------------------|------------------|---------------------|---------------------|
| <b>Sabit Nokta</b> |                    |                  |                     |                     |
|                    | Dinamik Aralık     | $6.6 \cdot 10^4$ | $4.3 \cdot 10^9$    | $1.8 \cdot 10^{19}$ |
|                    | Maksimum doğruluk  | 4 bas.           | 9 bas.              | 19 bas.             |
| <b>Kayan Nokta</b> |                    |                  |                     |                     |
|                    |                    | 4E,12M           | 8E,24M              | 1E,51M              |
|                    | Üs Dinamik Aralığı | $3.3 \cdot 10^4$ | $5.8 \cdot 10^{76}$ | $10^{1419}$         |
|                    | Taban Doğruluğu    | 3 bas.           | 7 bas.              | 15 bas.             |

Tablo 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 genişliği olan sayı, 8E,24M formatı ile ele

alındığında, dinamik aralığı  $2^{255} = 5 \cdot 10^{76}$ , ve doğruluğu 7 basamaktan büyük olmaktadır. Sayı, 11E,21M 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 lojisi ile, sayıların tam kapasite ile kullanılmasına olanak sağlamaktadır.

Sabit nokta ile kayan nokta ile çalışan işlemcilerin seçimi sırasında etkili bir faktörde 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 fiyatları 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 bulundurmamak, sonucun en düşük seviyeli bölümünü atmakta ve yeterli doğrulukta olmasına yetmemektedir.

Sonlu kelime genişliği, dijital denetleyicide kuantalama hatasına neden olacaktır. Dijital bir işaretin, ayrık zamanda genlik seviyeleri ile gösterimi sırasında ortaya çıkan bu hata, işaretin kazanç sabitlerinin, maksimum ve minimum değerlerinin kayba uğramadan temsil edilmesi ile ortadan

kaldırılabilir. Kuantalama hatasının azaltılması için, işlemcinin bit sayısının yani kelime genişliğinin artırılması 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 ve 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 artırarak 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.

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.

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

Örneğin, 1 kHz band genişliği olan bir sistemin kontrolunda, 100 µs '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 elemanında bir örnekleme periyodu kadar gecikme oluşturacaktır. Bu nedenle tasarım sırasında, A/D ve D/A çevirici yerleşme zamanı hesaba katılmalıdır.

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 yapılara 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 oluşturdıklarından, tasarlanması ve analiz edilmesi daha basit olmaktadır. Kullanılan bazı yapı türleri, direkt yapı, kaskat yapı, paralel yapı ve durum uzay modeli yapısıdır.

### 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 artırılması 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.

**TABLO 3.4 : İşlemden Önce 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, s istemin 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 işletime 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 ( en yüksek anlamlı byte ) saklanmaktadır.

**TABLO 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, 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 yuvarlarken, 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.

$$\varepsilon = \frac{y(\text{yuvar.}) - y(\text{gerçek})}{y(\text{gerçek})} \quad \frac{1}{2}2^{-b} < \varepsilon < -\frac{1}{2}2^{-b} \quad [3]$$

Yuvarlatma ve kesme hataları sistemde gürültü ve kararsız çalışmaya neden olmaktadır. Sınırlı salınım oluşturmamasından 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 yapılara parçalanması ile önlenebilir. Örneğin, dördüncü dereceden bir denetleyicinin birbirine paralel ayırık 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 olasılıkta 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 simülasyon 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;

$$\begin{aligned} x_{n+1} &= A x_n + B u_n & x : \text{giriş, } y : \text{çıkış, } u : \text{kontrol} & [4] \\ y_n &= C x_n + D u_n & A, B, C, D : \text{durum matrisleri} & \end{aligned}$$

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 şekile gelmektedir.

$$S_{u,j} = \frac{R^{pr}}{R^{ph}_{u,j}} \quad S_{y,j} = \frac{R^{ph}_{y,j}}{R^{pr}} \quad [5]$$

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

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'lık 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.

$$S_{x,i} = \max(|x_{n,i}|) \quad S_x = \text{diag}(1/S_{x,i})$$

$$\begin{aligned} x_{n+1} &= (S_x^{-1} A S_x) x_n + S_x^{-1} B u_n & [6] \\ y_n &= C S_x x_n + D u_n \end{aligned}$$

Ölçek faktörü matrisinin hesaplanmasında izlenen bir diğer yolda, simulasyon 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.



## BÖLÜM 4

### DSP GENEL YAPISI VE UYGULAMA ALANLARI

#### 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 yarıiletken 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 yapı, 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, bu sayede DSP `lerde mimari yapıdan gelen bir özellik olarak sağlanmıştır. Bunun yanında, paralel işlem yapabilme özelliği, aynı anda birkaç komut işleyerek hızının katlanmasını sağlamaktadır. Bir mikroişlemcinin işlem yapabilme kapasitesini gösteren komut sayısı ve adresleme modları, DSP sistemlerinde diğer işlemcilere 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 iletebilmesi, sistemin üstün özellikleri olarak tanımlanabilir.

DSP sistemi, hızlı ve doğru işlem yapabilme niteliklerini sağlaması için belirli özellikleri yapısında bulundurmaktadır. 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 tarafından gerçekleştirilmesi mümkün iken, belirli bir gecikmeye neden olacağından, DSP içinde ayrı bir Veri adres generatörü bulunmaktadır. Adresleme işleminin önem kazandığı, sıralı adreslemenin yapılmadığı uygulamalarda bu ünite, bağımsız çalışan bir işlemci gibi, adresleri üretmektedir.

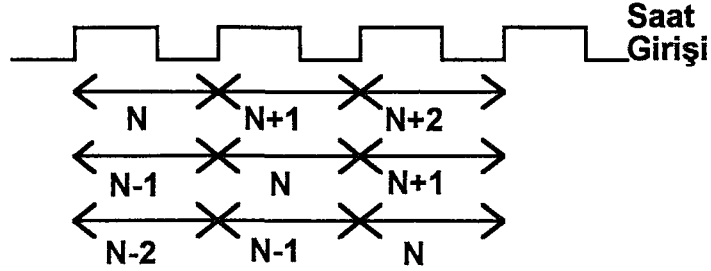
Algoritmanın işlemesi sırasında komutların sıralı ve doğru olarak işlemesi, Komut sıralama ünitesi sayesinde olmaktadır. Bu ünitenin varlığı ile, komutların işlemesi sırasında, işlemcinin mikrokod olarak çevrim ve

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, boru 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. Boru 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 iki katı oranında azaltmaktadır. Sistemde boru hattı denilen yapının olması aynı anda birden fazla komutun işlemesine izin vermektedir. Komut saat çevrimini en düşük seviyesine indirirken, işlemcinin çıkışından en yüksek performansın alınmasına olanak sağlamaktadır. Birinci nesil DSP `lerde iki seviyeli bir boru 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.



**ŞEKİL 4.1 : Üç Seviyeli Boru Hattı İşlemesi**

Boru 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 boru 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.

**TABLO 4.1 : DSP ve Mikroişlemcinin Komut İşleme Sayıları**

| İŞLEMCİ        | TAŞI | TOPLA | MAC* | DALLAN |
|----------------|------|-------|------|--------|
| 80186 Mikroiş. | 2    | 3 - 4 | 20   | 13     |
| TMS32010 DSP   | 1    | 1     | 2    | 2      |
| ADSP-2100      | 1    | 1     | 1    | 1      |

\* : Çarp ve bir önceki sonuca ekle

Tablo 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 sırasında 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 faktörde kolay programlanabilme özelliğidir. Komutların basit bir yapıda olması, algoritmanın kolay gerçekleşmesini 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), Arimetik - 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 ile 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 mikro işlemcilerden 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, işaretli ve karışık moddaki sayılar üzerinde çarpma işlemini gerçekleştirmektedir. Sonuç üzerinde yuvarlatma yaparak veri kaybının minimum düzeyde olmasını sağlamaktadır. Mikro işlemcilerden farklı olarak donanımsal bir yapı olarak işlevini gerçekleştiren çarpım ünitesi, DSP 'ye tartışılmaz bir üstünlük kazandırmaktadır. Mikro işlemcilerde 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 özellikte girişindeki sayıların format kontrolünü yapabilmesidir. Bilgi işlem sistemlerinde kullanılan sayılar, işaretli veya ikinin tümleyeni, tamsayı veya kesirli yapıda olmaktadır. Bu tür sayıların birbirine dönüşümünü en az hata ile sağlamalıdır. Sonuçta elde edilen değer yine istenen formatta, tamsayı veya kesirli olması gerekir. Ayrıca sonucun genişletilmiş veya yuvarlatılmış 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 boru hattı çalışmasına olanak sağlamakta, ve kullanılan kütükler, sistemin minimum gizlilik içinde, kullanıcıya görünür bir yapıda çalışmasını kolaylaştırmaktadır.

Çarpma işleminde, işaret kontrolünün yapılması, işlemin doğruluğuna etkilidir. İşaretli ve işaretli sayıların çarpımında sonucun en çok

anamlı bölümü, işaretli, en az anlamlı bölümü ise, işaretsiz 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 noktalılara göre daha pahalı olması, yalnız çok yüklü matematik işlem gerektiren uygulamalarda kullanılmasını verimli yapmaktadır.

#### 4.2.2. Çarpım - Akümülatör Ünitesi ( MAC ) :

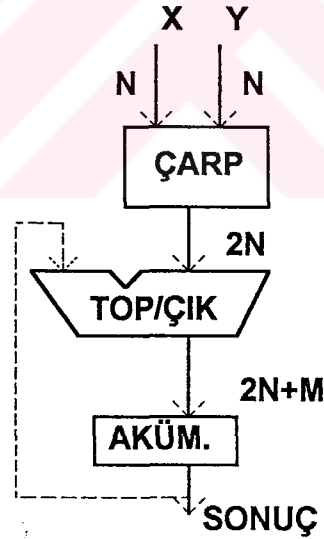
İlk üretilen DSP'lerde tek başına bulunan çarpım ünitesi, daha sonraki nesillerde akümülatör ile birleştirilerek, Çarpım-akümülatör ünitesi olarak adlandırılmıştır.

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

$$\sum b(n)x(n-k) \quad [7]$$

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 `ın yapısında bulunan geri besleme yolları ve boru 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 MSH (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.



**ŞEKİL 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 geribesleme yolu üzerinden tekrar TOP/ÇIK birimine gelerek, işlemin tekrarını sağlar. TOP/ÇIK biriminin çıkışı,

M adet büyütme biti ile genişletilmiştir. Bu sayede, en az  $2^M$  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.

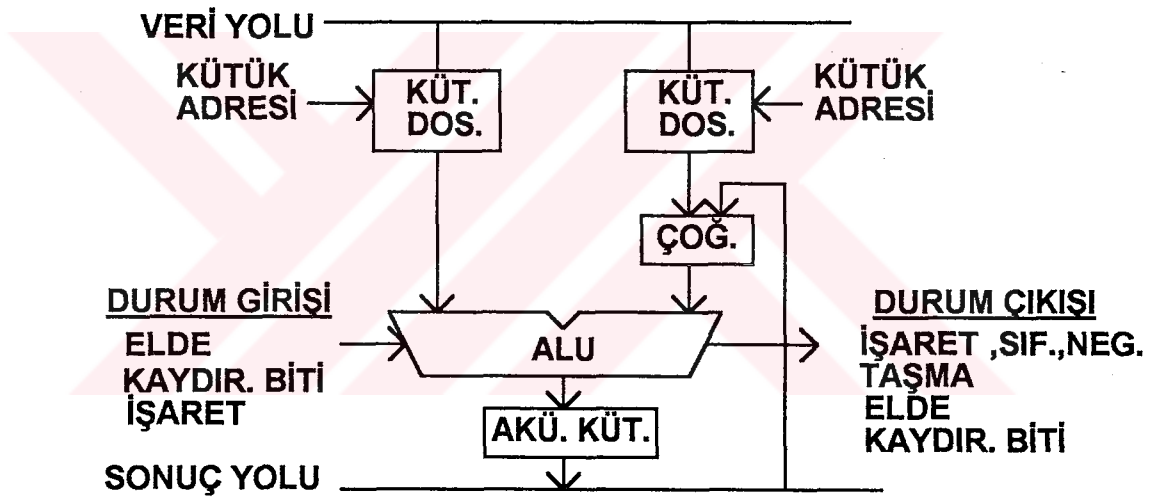
MAC ünitesinde yapılan işlemlerde, istenen bir diğer özellikte taşma ve yuvarlama kontrolü yapılabilmesidir. Taşmanın daha önceden algılanabilmesi, sisteme eklenecek bir lojik birim ile sağlanabilir. Eğer bütün bitler eşit değilse, yani hepsi bir veya sıfır değilse, akümülatörün taşması, büyütme kütüğünde 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, boru 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.

ALU `dan istenen bir diğ er  zellikte, iřlem yapılmadan  nce, genel durum hakkında, durum bitleri yardımıyla, bilgi vermesidir. Sonucun pozitif, negatif veya sıfır olması, tařma oluřup oluřmadıėı, elde olup olmadıėı, durum k t ğ nde bulunan bayraklar yardımıyla belirtilmektedir. DSP `deki ALU `nin bayraklarının diğ er  niteler tarafından kullanılabilir olması, kayan nokta aritmetiėi ile yapılan iřlemlerde  l eklendirmeyi kolaylařtırmaktadır. ALU ile b lme iřlemlerinin ger ekleřtirilmesi, diğ er  nitelere g re daha hızlı olduėundan tercih edilebilir. Veri yollanması ve iřlenmesi sırasında, ALU `daki bir diğ er  zellikte verimli  alıřabilmesidir. Veri yollanması ve aritmetik iřlemlerin aynı  evrimde ger ekleřmesi ile aynı saat  evrimi i inde ayrı iki sayının y klenebiliyor olması,  nitenin hızını katlamaktadır. ALU `nin, iřlem yaparken, aynı saat  evrimi i inde, giriřteki bayrakların durumuna g re, sonucu kaydırabilmesi, DSP `nin hızını arttırmaktadır.



**ŐEKİL 4.3 : ALU `nin Basit Yapısı**

ALU `nin bu iřlemleri ger ekleřtirebilmesi, tam anlamıyla yeterli sayıda olan k t k dosyasına baėlıdır. K t k dosyasının yeterli geniřlikte olması, program tarafından devamlı kullanılan verinin saklanması ve sonu ların saklanmasında bellek kullanımını minimum d zeye indirmeye yaramaktadır. Veri yollama ve aritmetik iřlem yapma iřlemlerinin aynı saat  evriminde ger ekleniyor olması, yazma ve okuma iřlemlerini aynı anda ger ekleřtirebilen k t kler sayesinde olmaktadır. Giriř iřlemlerinde kullanılan k t klerin, program akıřı sırasında g zlemlenebilir 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 üniteye veri yollarının bulunması, ana yola gerek kalmadan her ünitenin birbiri ile haberleşmesini sağlamaktadır.

#### 4.2.4. Yığın Kaydırma Ünitesi ( BS ) :

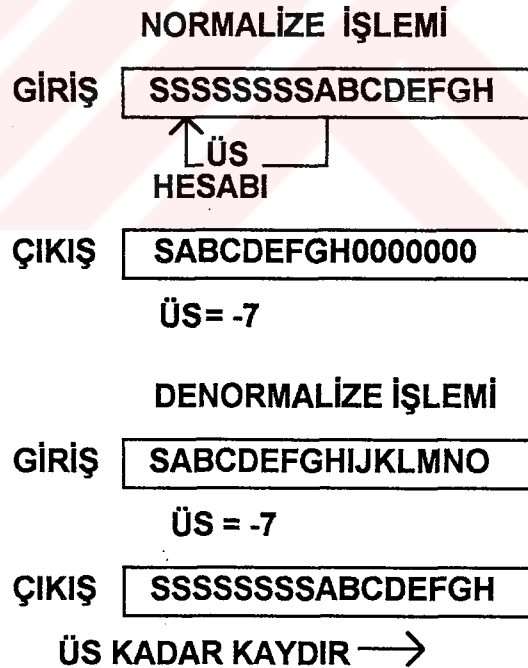
Kaydırma ünitesi, işleme sırasında taşma ve veri kaybı sorunlarını önlemek amacıyla, sayıları ölçeklendirmek için ve sabit nokta ile kayan nokta arasında dönüşüm yapabilmek için kullanılan bir ünite dir. Mikro işlemcilerdeki kaydırma ünitesi, her bir çevrimde bir bit kaydırma yaparken, DSP `deki ünite,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

Kaydırma ünitesi, aritmetik ve lojik kaydırma işlemleri yapabilir. Aritmetik kaydırma işlemi, sağa ve sola kaydırma ile sayıların ölçeklenmesini 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. Mikro işlemcilerde 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, Hemen Kaydırma işlemi, komutta belirtilen bir kontrol değeri kadar taşıma işlemi gerçekleştirmektedir.

Kaydırma 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.

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.



**ŞEKİL 4.4 : Normalize ve Denormalize İşlemleri**

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 lojiği 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 ) :**

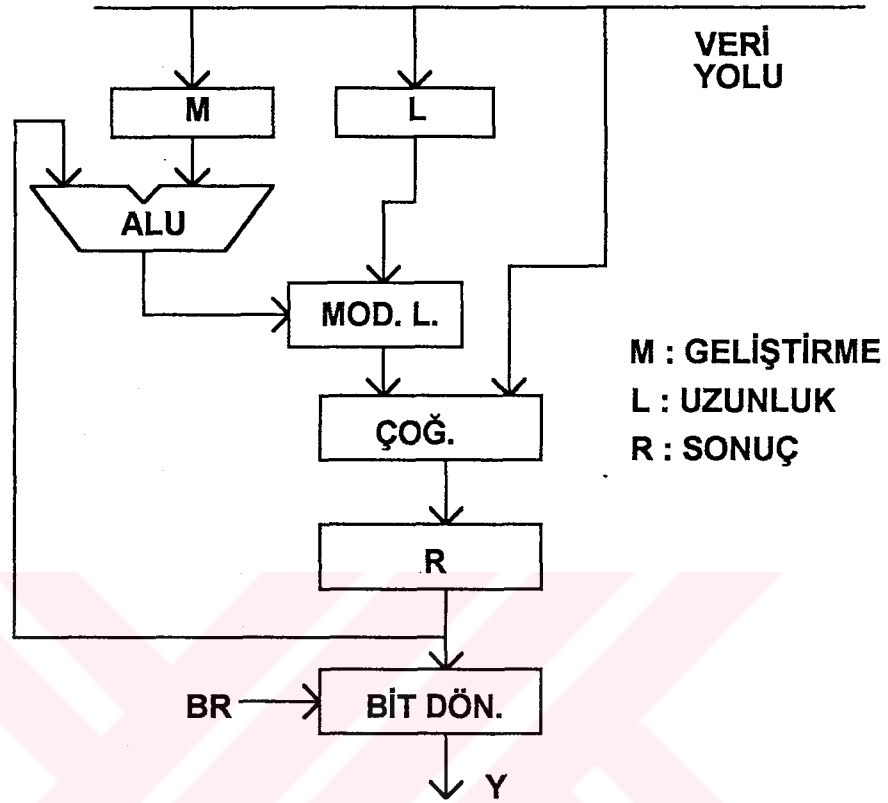
Veri adres generatör ünitesi, bellek üzerindeki adreslere belirli sayı değerleri atayan kütükler içeren bir yapı görünümündedir. Bu sayede, veri okunup yazılması, çok daha hızlı olmaktadır. DSP 'lerde adres generatörünün ayrı bir yapı olması, algoritmanın işlemesi sırasında, aynı anda birkaç verinin 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, birçok özellik sağlamaktadır. Ünitenin, kaydırma, bit dönüştürme, lojik fonksiyon ve

maskeleme işlemlerini gerçekleyebilmesi, FFT gibi karmaşık adresleme gerektiren uygulamalarda birçok kolaylık sağlamaktadır.



**ŞEKİL 4.5 : DAG'ın Basit Yapısı**

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 maskeleme yapabilmesi, sisteme çok büyük bir avantaj sağlamaktadır. DAG kütüklerine döngü değerlerinin yüklenmesi ile, program akışı bozulmadan çevrimlerin işlemesi mümkündür.

#### 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, proram 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 arttırarak, 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 çağırmak, dış I/O elemanları ile haberleşebilmek için gereken kesme servisini sağlamak, kesme servis programlarına dallanmak ve program tamamlandığında geri dönüş adresini belirlemek, olarak sıralanabilir.

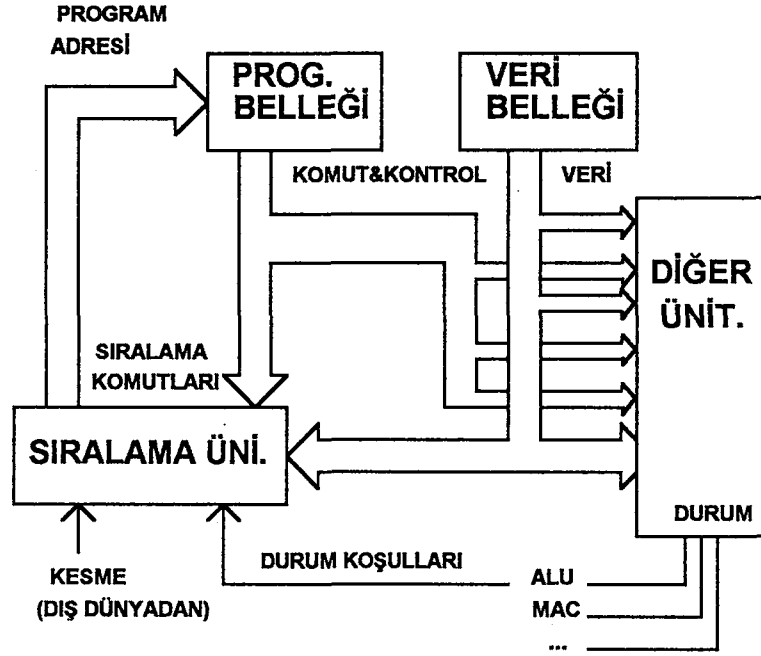
SEQ `ni oluşturan ana birimler; Program Sayacı (PS), lojik durum algılayıcı, yığın ( altprogram ve kesme adresleri, sayaç değerleri ve atlama adresleri için ), döngü sayaçlarıdır. SEQ `nin tasarımındaki amaç, program akış kontrolu yapabilmesi yanında, yeterli yığın boyu, gerek yazılım döngülerinde, gerekse 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, boru hattı biçiminde işlem yapma tavsiye edilmemektedir. İşlemin parçalara bölünüp, sıralı olarak işletilmesi olarak adlandırılan boru hattı işlemi, işlem hızını arttırmak ve veri kaybını önlemek için kullanılmaktadır. SEQ `de bu özelliğin olmaması, boru hattı işlemi sırasında olan gecikmeler ve ünitenin veri tablosundan 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 deceden maskelenemez kesme tarafından önlenmektedir. Döngü kontrolü için gereken karmaşık kod yapısını azaltıp, iki veya üç yollu 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 lojigi, 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 yığnında saklı olan adres ve komut kodu tarafından üretilen adres sistemin işleyişini belirlemektedir. Bir diğer alternatifte; indeks atlama olarak adlandırılan, adresin dışarıdan program adres yoluna konup PS `ye aktarılmasıdır.



**ŞEKİL 4.5 : Sıralama Ünitesi ile Diğer Ünitelerin İlişkisi**

Programın işleyişi sırasında, normal çalışma, PS ve PS yığınının yararlanılmaktadır. İşlemekte olan komutun adresi, PS 'de saklanır, ve çıkış bir artırılır. Bu değer, bir sonraki komutun adresini veya, alt program veya 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 yığınının, o andaki PS değeri de PS yığınının atılmaktadır. Döngü veya çevrimin sona ermesi için gerekli koşul sağlandığında, program tarafından kullanılan değerler, SEQ tarafından tekrar yüklenerek, işleme devam edilmektedir. Kesme isteği ile karşılaşıldığında, öncelik ve maskelenme durumuna göre, gerekli programı çalıştırmakta, veri kaybını önlemek için, durum bilgisini durum kütüğünde saklamaktadır.

## BÖLÜM 5

### ÖRNEK DSP SİSTEMLERİNİN 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 işlerse 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 Üni., veri akışının kontrolunu yapan DAG ve SEQ üniteleri bulunmaktadır. Bu ünitelerden birinin eksik olması veya yeterli performansta çalışmaması sistemin performansını büyük oranda düşürecektir. Örneğin; Kaydırma Üni. `nin 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 etkende birimleri bağlayan yollardır. Yolların yeterince geniş olmaması, hızı azaltacağı gibi, aritmetik işlemler sonunda verinin tam olarak iletilememesi 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, 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, motorun tahrip olmasına kadar gidebilecek sorunlar doğuracaktır. Bu yüzden, DSP sistemleri, gerçek zamanda veri işleyecek yapıda olmalıdır.

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.

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

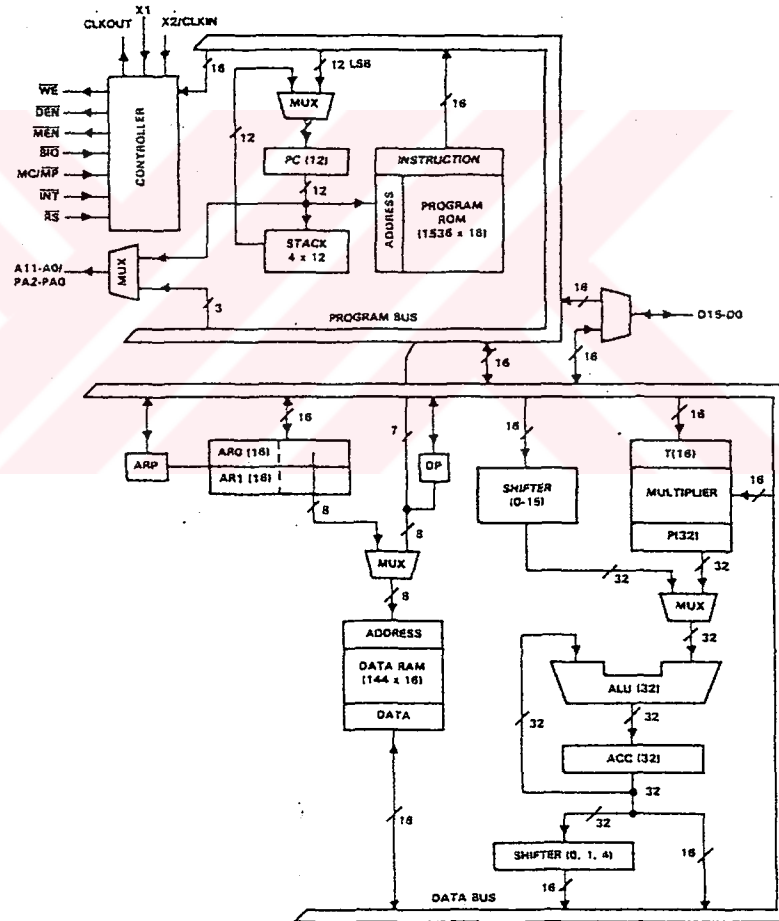
**TABLO 5.1 : DSP Sistemlerinin Genel Özellikleri**

|                  | Bit Uzun.       | Saat Fre.<br>MHz | İşlem Kap.                    | MAC İşl.      | Tranzistör     |
|------------------|-----------------|------------------|-------------------------------|---------------|----------------|
| <b>TMS32010</b>  | <b>16 Sabit</b> | <b>20</b>        | <b>5 MIPS</b>                 | <b>400 ns</b> | <b>58.000</b>  |
| <b>TMS320C25</b> | <b>16 Sabit</b> | <b>40</b>        | <b>10 MIPS</b>                | <b>100 ns</b> | <b>160.000</b> |
| <b>TMS320C30</b> | <b>32 Kayan</b> | <b>40</b>        | <b>20 MIPS/<br/>40 MFLOPS</b> | <b>50 ns</b>  | <b>695.000</b> |

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 unsurları yapısında bulundurarak, mikroişlemcilerin yetersiz kaldığı birçok alanda uygulama bulmuştur. 2,4  $\mu$ m NMOS teknolojisi ile üretilmiştir. Donanım olarak carım.ünitesi bulunması, işlem hızının katlanmasına neden olmuştur.



|                            |                                 |
|----------------------------|---------------------------------|
| ACC : Akümülatör           | ARP : Yardımcı Kütük Göstergesi |
| AR0 : Yardımcı kütük 0     | AR1 : Yardımcı kütük 1          |
| DP : Veri sayfa göstergesi | PC : Program Sayacı             |
| P, T : P ve T kütükleri    | MUX : Çoğullayıcı               |

ŞEKİL 5.1 : TMS32010 Blok Diyagramı

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 türleride 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, indirekt 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.

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 kelimelik ROM bellek, 544 kelime veri RAM bellek, 128K kelimelik 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.

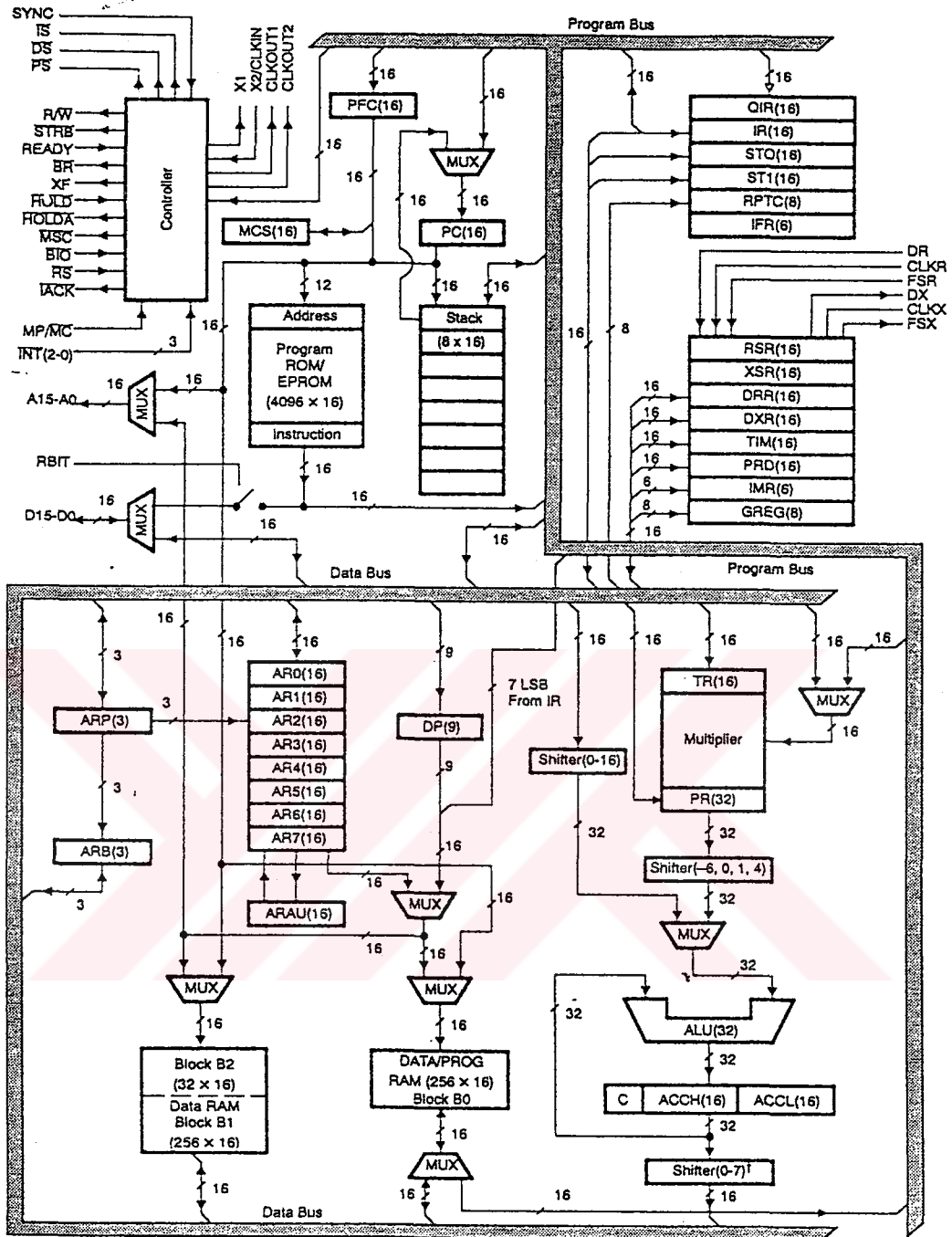
TMS320C25 sistemi, 100 ns çevrim süresi ve 10 MIPS işlem hızı ile DSP sisteminin bütün yararlarından tam olarak faydalanmayı sağlamaktadır. Veri sıkışmasını önlemek için boru hattı denilen parçalı çalışma sayesinde aynı anda komutun alınması, çözümlenmesi ve işletilmesi gerçekleştirilebilir. Komutların 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.

TMS320C25 'in bellek alanı, 4K kelimelik ROM ve 544 kelimelik RAM 'dan oluşmaktadır. RAM bellek üç ayrı bölüme ayrılmıştır. 256 kelimelik bölümü veri veya program belleği olarak seçilebilirken, 288 kelimelik bölümü sadece veri belleğidir. 544 kelime genişliğindeki veri belleği, 512 kelimelik dizilerin

saklanması için izin verirken, 32 kelimelik 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 olarakta işlev görebilir. Doğrudan veri belleğine bağlı olduğundan 16 bitlik işaretli 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 işlemini komutuna bağlı olarak gerçekleştirmektedir. 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. Boru hattı çalışması sırasında, kaydırma işlemi komutun bir parçası olarak ele alınmakta, böylece işletim için ek çevrim süresine gerek kalmamaktadır.

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



**ŞEKİL 5.2 : TMS320C25 Fonksiyon Blok Diyagramı**

\* : Birimlerin açık isimleri Ek-A`da verilmiştir.

ve 32 bitlik sonucu saklayan sonuç kütüğüdür. Sonuç kütüğünün çıkışı bir veya dört bit sola kaydırma işlemini gerçekleştirmektedir. Bu kesirli sayılarla işlem yaparken, doğruluğun artırılmasını sağlamaktadır. Aynı zamanda çıkış, altı bit sağa kaydırma işlemi gerçekleştirerek, 128 çarpma-yükle işleminin taşma olayı olmadan gerçekleşmesine olanak sağlamaktadır.

Çarpım Ünitesi işaretli ve işaretli çarpma işlemini, 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.

Örneğin MACD komutu, çarpma, yükleme ve veri iletme işlemlerini 100 ns sürede bir saat çevriminde tamamlamaktadır. Bu komutun işlemesi 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 taşma 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 arttırılarak, 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.

TI firmasının sabit nokta ile işlem yapan birinci ve ikinci nesil DSP sistemlerinden sonra, diğer firmalarda 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, 1K 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.

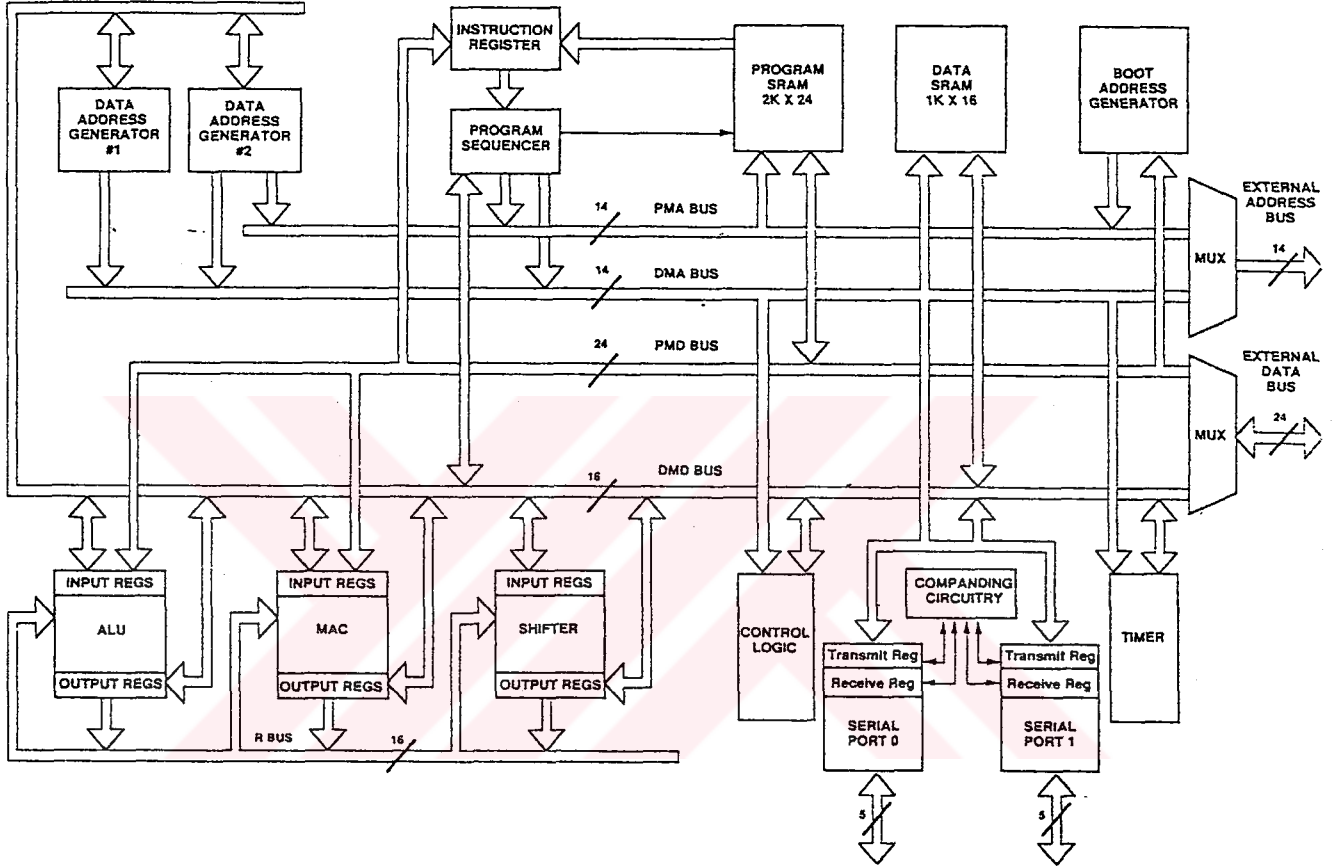
ADSP-2101 'in esnek mimari yapısı ile, paralel işleme yapabilmektedir. Tek çevrim süresinde, bir sonraki program adresini üretebilmekte, bir sonraki komutu alabilmekte, bir veya iki veri iletimi yapılabilen, bir veya iki veri adres gösterici düzeltebilmekte, aritmetik bir işlem gerçekleştirebilmekte,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

Kaydırma 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 boru hattı çalışmasına izin vermektedir.

Veri transferini verimli kılmak için beş adet yol kullanılmaktadır. Bunlar; program bellek adres yolu (PMA), program bellek veri yolu (PMD), 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 sağlanmaktadır. Program sıralama ünitesi ve veri adres generatörü, işlem ünitelerinin verimli kullanımını sağlamaktadır. Sıralama ünitesi, koşullu atlama, alt program çağırma ve geri dönme işlemlerini tek



**ŞEKİL 5.3 : ADSP-2101 Blok Diyagramı**

saat çevriminde gerçekleştirmektedir. Sıralama ünitesi, işlemekte olan komutu saklayan komut kütüğü tarafından sürülmektedir. Komut kütüğü, program akışında tek seviyeli boru hattı çalışması gerçekleştirmektedir. Komutlar, tek saat çevriminde alınarak, komut kütüğüne yüklenmekte ve çözümlenmektedir. Bir sonraki çevrimde işletilmekte iken diğer komut alınmaktadır. DAG ünitesi, program ve veri belleğinden veri aktarımı için adres üretimini sağlamaktadır. Her bir DAG ünitesinin, dört adres göstercisi bulunmaktadır. Bu gösterciler 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.

**TABLO 5.2 : ADSP Sabit Nokta DSP 'lerin Karşılaştırılması**

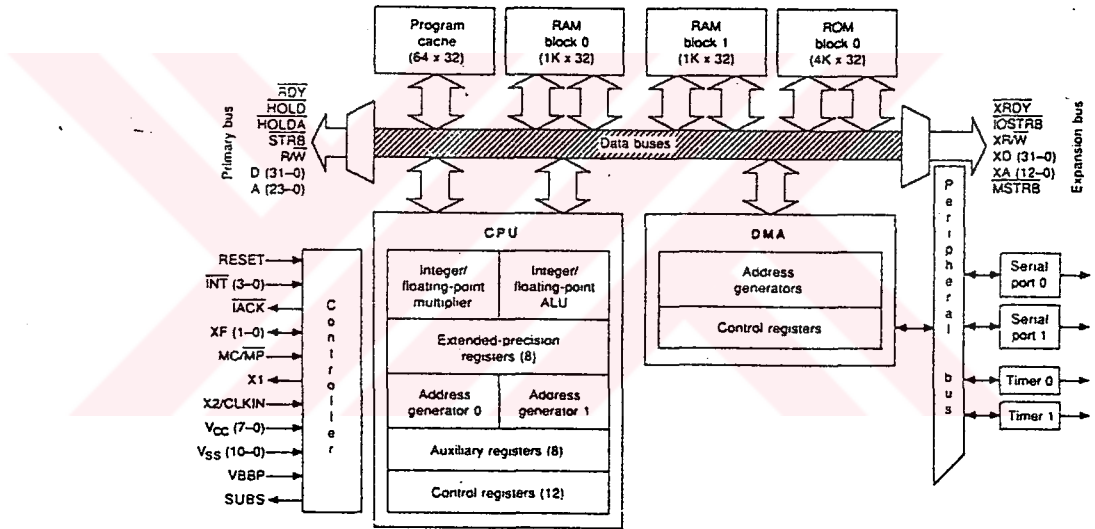
| Algoritma          | Çevrim<br>Saat Sayısı | ADSP-2105<br>100 ns | ADSP-2101<br>60 ns |
|--------------------|-----------------------|---------------------|--------------------|
| PID Çevrim         | 8                     | 0.8 $\mu$ s         | 0.48 $\mu$ s       |
| FIR Filtre         | 1                     | 100 ns              | 60 ns              |
| 2. Der. IIR Filt.  | 12                    | 1.2 $\mu$ s         | 0.72 $\mu$ s       |
| 6. Der. IIR Filtr. | 28                    | 2.8 $\mu$ s         | 1.68 $\mu$ s       |
| Matris Çarpımı     | 52                    | 5.2 $\mu$ s         | 3.12 $\mu$ s       |
| 256 N.Komp. FFT    | 6750                  | 0.675 ms            | 0.405 ms           |

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 yapabilme özellikleri sayesinde uygulama alanına girmişlerdir.

TMS320 ailesinin üçüncü nesil işlemcisi, TMS320C30, kayan nokta İşlem 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.

TMS320C30 'ün 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.



**ŞEKİL 5.3 : TMS320C30 Blok Şema Yapısı**

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. İşaretili ve işaretsiz 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.

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.

Üçüncü nesil DSP 'lerde boru 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 boru 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 çıkışına boru hattı denilmektedir. Doğrudan erişim kanalı bütün bu ünitelerden bağımsız işlemektedir. Tek bir

saat çevriminde işleyen komutların kontrolü ile çok daha yüksek performanslara erişilmektedir.

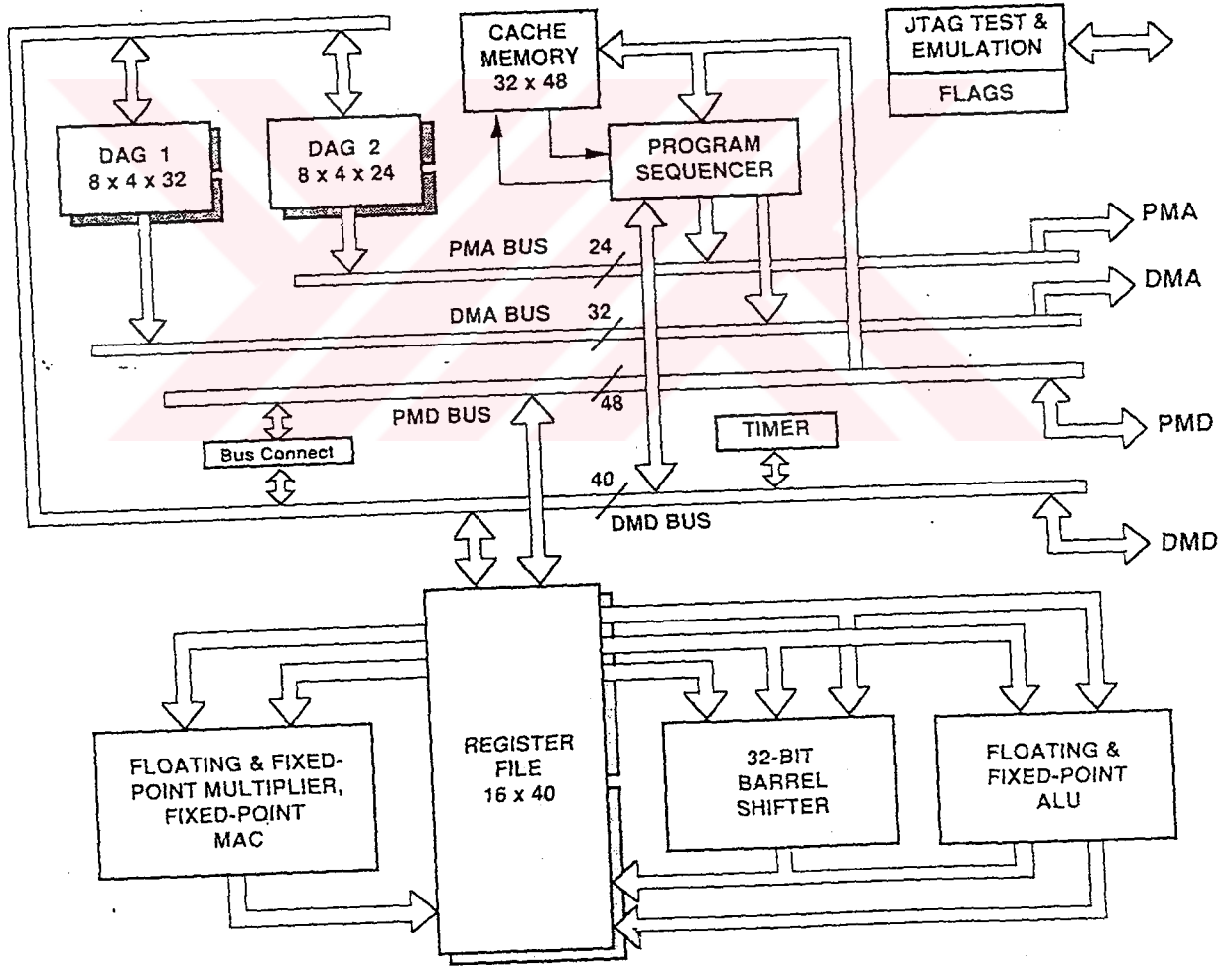
Analog Devices firması tarafından üretilen kayan nokta işlemcisi ise, ADSP-21010 ve ADSP-21020 'dir. ADSP-21010 sabit nokta DSP 'de olduğu gibi, bellek, çevre birimleri ve hız olarak daha az fakat daha ucuz bir modeldir. ADSP-21010 temel özellikleri; 80 ns komut çevrim süresi, 12.5 MIPS işleme oranı, 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-21020 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.

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 gerçeklemektedir. 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.

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, onbir komutun aynı anda işlenmesini sağlayan paralel çalışma, 6 iletişim portu, 6 kanal DMA yardımcı işlemci bu işlemcinin temel özellikleridir. Beşinci nesil DSP ise, TMS320C5x serisi olarak çıkarılmış ve sabit nokta bir işlemcidir. Birinci ve ikinci nesiller ile kod uyumlu olan bu DSP, sabit nokta işlemcilerin en gelişmiş olanıdır. Temel özellikleri şu şekilde sıralanabilir: 50/35 ns komut işleme süresi, 8K kelimelik program/veri RAM bellek, 2K ROM bellek, 544 kelime veri/program RAM bellek, 128K adreslenebilir toplam bellek, 84 pinli yapıdır.



ŞEKİL 5.4 : ADSP-21000 Temel Mimari Yapısı

## BÖLÜM 6

### DSP ve MİKROİŞLEMCİLERİN KARŞILAŞTIRILMASI

DSP ve mikroişlemcilerin karşılaştırılmasında, uygulamada kullanıldıkları sistemin özellikleri dikkate alınmalıdır. İki sisteminde 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 aynı yolları kullanması ve 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 yollarına mümkün olmamakta, ancak bir zamanlama diyagramını takip edecek şekilde belirli gecikmelerin sağlanması ile adres ve veri ard arda 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 integrasyon 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.

**TABLO 6.1 : DSP ve Mikroişlemcilerin Özellikleri**

| Özellik                      | 320C14 | 320C25 | 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    |
| PID Çevrimi                  | 2.2    | 1.3    | 27.0   | 25.0  | 4.8   | µs    |
| Matris Çarpımı<br>(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  $\mu$ s sürede gerçekleştirirken, ilk nesil DSP 'de bu işlem 0.16  $\mu$ 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 oluşturarak 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.

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

**TABLO 6.2 : DSP Bellek ve Adres Özellikleri**

| DSP       | Adresleme Modları | Yığın Boyu | Veri Alanı | Program Alanı | Kelime G. Veri/Prog. |
|-----------|-------------------|------------|------------|---------------|----------------------|
| 320C1x    | 4/4               | 4          | 256 W      | 4 KW          | 16/16                |
| 320C2x    | 4/4               | 8          | 64 KW      | 64 KW         | 16/16                |
| 320C5x    | 5/8               | 16         | 96 KW      | 64 KW         | 16/16                |
| ADSP 2101 | 9/9               | 16         | 16 KW      | 16 KW         | 16/24                |
| M56000    | 5/18              | 16         | 128 KW     | 64 KW         | 24/24                |

Mikroişlemcilerin komut kümesi, DSP `ye göre çok daha geniştir. Mikroişlemcilerde komut seti, birçok işlevi en kısa ve en verimli görecektir. 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 işlemlerini 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 mikroişlemcilere göre zayıf olduğu bir alanda çevre birimleri ile sağladığı integrasyondur. 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 tümdevre 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.

**TABLO 6.3 : DSP `lerin bazı fonksiyonları işleme süreleri ( $\mu$ s)**

| <b>İŞLEM</b>              | <b>320C14</b> | <b>320C50</b> | <b>80C196</b> | <b>68000</b>  |
|---------------------------|---------------|---------------|---------------|---------------|
|                           | <b>26 MHz</b> | <b>40 MHz</b> | <b>12 MHz</b> | <b>16 MHz</b> |
| <b>PID ( 3. der.)</b>     | <b>3.2</b>    | <b>1.0</b>    | <b>27.0</b>   | <b>17.0</b>   |
| <b>4. der. Polinom</b>    | <b>4.6</b>    | <b>1.45</b>   | <b>29.7</b>   | <b>45.4</b>   |
| <b>Kayan N. Çarpma</b>    | <b>5.2</b>    | <b>1.75</b>   | <b>19.8</b>   | <b>25.6</b>   |
| <b>Heks. Des. Çevrimi</b> | <b>91.3</b>   | <b>26.2</b>   | <b>90.8</b>   | <b>113.0</b>  |
| <b>Kesme İş. (Maks.)</b>  | <b>7.0</b>    | <b>0.1</b>    | <b>14.8</b>   | <b>15.1</b>   |
| <b>Kesme İş. (Min.)</b>   | <b>1.6</b>    | <b>0.1</b>    | <b>6.7</b>    | <b>4.0</b>    |

Tablo 6.3 `te DSP ve mikroişlemcilerin bazı fonksiyonları işleme süreleri görülmektedir. Bu fonksiyonlar; üçüncü dereceden PID denetleyici, dördüncü dereceden polinom, kayan nokta aritmetiğinde iki sayının çarpımı, heksadesimal bir sayının desimal bir sayıya çevrilmesi, ve kesme isteği

geldiğinde, işlemcinin yapması gereken işlemlerin maksimum ve minimum işleme sürelerinin karşılaştırılmasıdır. Görüldüğü gibi, C1x birinci nesil ve C5x beşinci nesil DSP, mikroişlemciler ile karşılaştırıldığında, çok daha kısa sürede işlemlerini tamamlamaktadırlar. Donanım yapısının getirdiği özellikler ile, çarpma ve toplama işleminin ağırlıkta olduğu fonksiyonlarda sistem çok daha hızlı işlem yapmaktadır.

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 artırıcı bir unsur olmaya devam etmektedir. DSP `ler ilik üretildiklerinde, mikroişlemcilerden çok daha pahalı olmalarına rağmen, şu anda tümdevre fiyatları, birbiriyle özdeş yapılar için çok farklı değildir. Özellikle kayan nokta DSP `ler sabit noktalılara göre daha pahalıdır. Her uygulamanın DSP ile çözülmesi matematiksel olarak tam bir sonuç verirken, mali bakımdan aşırı bir artış getirecektir. Bununla beraber, örneğin konum kontrolü 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.

## BÖLÜM 7

### DSP UYGULAMA ALANLARI

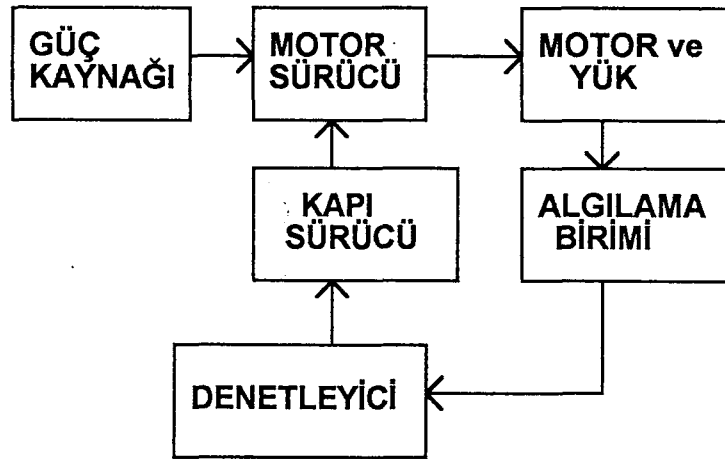
VLSI teknolojisindeki gelişmeler, DSP sistemlerinin çok kısa bir süre içinde kontrol uygulamalarının bütün alanlarına girmesine olanak sağlamıştır. Çok daha küçük alanlara daha fazla işlevi olan yapıların sığdırılması ile tasarım ve uygulama değişik boyutlar kazanmıştır. Özellikle karmaşık algoritmaların 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

anlaşılabileceği 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, mikro işlemcilerin 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, mikro işlemciler zayıf kalmaktadır. İşlem sayısı arttıkça mikro işlemci yavaşlamakta, bu da kontrol edebileceği sistemlerin performansını etkilemektedir. Bu gibi sorunlara çözüm olarak, mikro işlemcilerin 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.



**ŞEKİL 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 kullanılmaktadır. 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 anahtar elemanlar, güç tranzistörü, MOSFET veya IGBT olabilir. Özellikle MOSFET ve IGBT anahtarlarının 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, devrenin PWM işaretinin daha yüksek frekanslarda elde edilmesini sağlamaktadır.

Kontrol sistemindeki bir diğer önemli elemanda algılayıcılarıdır. 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.

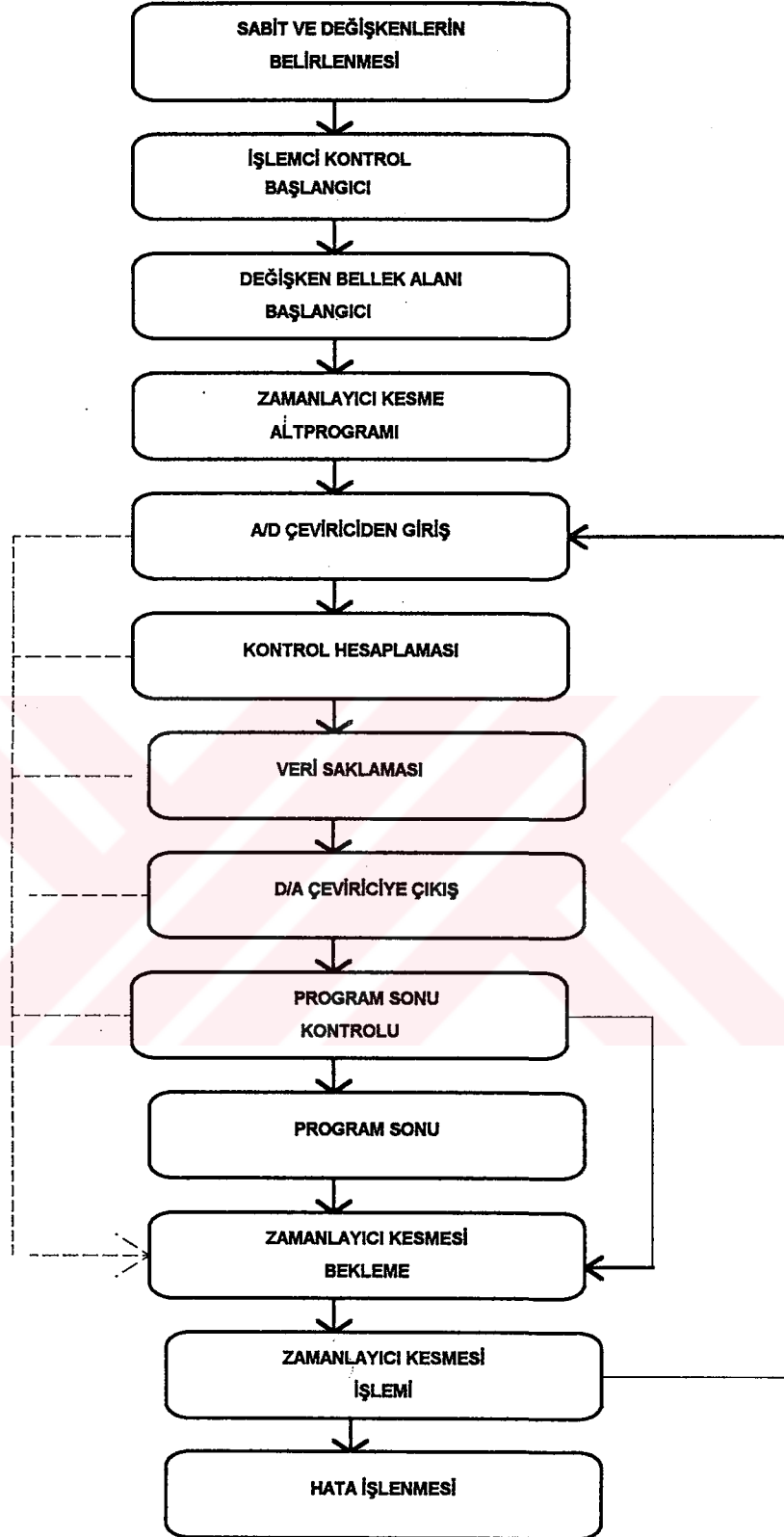
Tasarlanan kontrol sisteminin yapısından bağımsız olarak, gerçekleştirilen yazılımın belirli bölümleri aynı kalmaktadır. Motor kontrolünde, sistemin kontrol sistemine göre davranışını incelemek amacıyla, simulasyonu yapılarak, cevap analizi yapılabilir. Bu sayede, gerçek sisteme geçmeden önce, sistemin 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 zamanlayıcı ile değerlendirilmekte ve gereken işlemler yapılmaktadır. Zamanlama ve kesme işlemleri birer vektör yapısında olup, alt program gibi işlemektedirler.

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 kurulmaktadır. 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.



**ŞEKİL 7.2 : Kontrol Algoritmasına İlişkin Akış Diyagramı**

$$u(t) = K_C(e(t) + \frac{1}{T_i} \int e(s) ds + T_d \frac{de(t)}{dt}) \quad [8]$$

Gerçek zamanda [8] '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,  $T_i$  integral zaman sabiti ve  $T_d$  türev zaman sabiti olarak tanımlanmaktadır. Sistemde kazanç, işaretin kuvvetlendirilmesini sağlamaktadır. İntegral terimi, düşük frekans kazancını artırarak, kararlı hal hatalarının azalmasını sağlamaktadır. Türev terimi ise, faz kayması oluşturarak, sistemin kararlılığının düzelmesini ve band genişliğinin artmasını sağlamaktadır. PID sisteminin ayrık zamanda kullanılabilmesi için elemanlarının ayrıştırılması gerekmektedir.

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

$$P(t_k) = K_C ( b y_r - y(t_k) ) \quad [9]$$

şekline girmektedir.  $t_k = kT$ ,  $T$  örnekleme frekansı olmak üzere, örnekleme zamanlarını göstermektedir.  $b$ , 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, integral yapısı, hata işaretinin integralini almaya devam eder, ve integral terimi çok büyük değerlere ulaşabilir. Bu duruma, integral sisteminin tükenmesi denilmektedir. Hata terimi, integral 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 genliktaki sayıları saklayabildiğinden, kontrol değişkeninde taşma oluşacaktır. Bunun önlenmesi sistemin doyma modunda çalıştırılması ile mümkündür. İntegral teriminin bu sorununu önlemenin çeşitli yolları bulunmaktadır. Sistem içinde fazladan bir geri besleme çevrimi kurup,

sistemin çıkışı 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 integral yapısının girişine  $1/T_t$  kazancı ile uygulanmaktadır.  $T_t$  takip zaman sabiti olarak adlandırılmaktadır. Fazladan kurulan geri besleme, denetleyici çıkışı, doymuş değerdeki çıkışı takip edene kadar integralin değerinin değişmesini sağlamaktadır. Çıkışın takip etmesi işlemi,  $T_t$  zaman sabiti ile yerine getirilmektedir. Bu yöntem ile integral teriminin tükenmesinin önlenmesi durumunda, yapı ,

$$I(t) = \frac{K_c}{T_i} \int e(s) ds + \frac{1}{T_t} \int (u(s) - v(s)) ds \quad [10]$$

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

$$\frac{dI(t)}{dt} = \frac{K_c}{T_i} e(t) + \frac{1}{T_t} e_s(t) \quad [11]$$

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

$$I(t_{k+1}) = I(t_k) + \frac{K_c h}{T_i} e(t_k) + \frac{h}{T_t} e_s(t_k) \quad [12]$$

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,

$$\frac{sT_d}{1 + sT_d/N} \quad [13]$$

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,

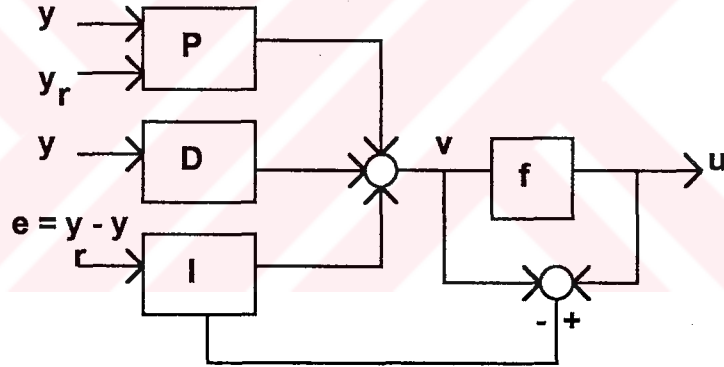
$$D(t) + \frac{T_d}{N} \frac{dD(t)}{dt} = -K_c T_d \frac{dy}{dt} \quad [14]$$

şeklini almaktadır. Bu terimin ayrıklaştırılması ile;

$$D(t_k) = a D(t_{k-1}) - b (y(t_k) - y(t_{k-1})) \quad [15]$$

$$a = \frac{T_d}{T_d + Nh} \quad b = \frac{K_e T_d N}{T_d + Nh}$$

şeklinde türev elemanı elde edilmektedir. Böylece klasik PID kontrol algoritması yanında sistemin hata yapmasının mümkün olduğu alanlardaki davranışını düzelten 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 integral terimleri kullanılmaktadır. Ayrık zamanda elde edilen förmüller yardımıyla, sistem birbirine paralel ve bağımsız yapılar şeklinde kurulabilir.



**ŞEKİL 7.3 : Geliştirilmiş PID Yapısı**

Sistemin dijital bir yapı olarak programlanabilmesi için, her terimin kaç işlemden oluştuğunun belirlenmesi, tasarıma yardımcı olacaktır. Özellikle sabit nokta ile işlem yapıldığında, hesaplama çok sayıların ölçeklendirilmesi ve taşma önlenmesi işlem sayısını ve süresini artırmaktadır. 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.

Sistem tasarımı sırasında örnekleme periyodunun seçimi sistemin davranışını etkileyecektir. PI yapısı kullanıldığında, örnekleme periyodu, integral 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.

**TABLO 7.1 : PID Yapısı İçin Komut Çevrim Sayıları**

| İŞLEM                | TMS32010 | TMS320C25 |
|----------------------|----------|-----------|
| Türev                | 9        | 9         |
| DYKS*                | 43       | 45        |
| Orantı               | 7        | 7         |
| İntegral Öteleme     | 12       | 8         |
| v`de DYKS*           | 23       | 23        |
| İnteg. tük. önleme   | 12       | 12        |
| İntegral             | 15       | 13        |
| İntegral doyma arit. | 10       | 10        |
| G/Ç ve diğer işlem.  | 14       | 14        |
| TOPLAM               | 145      | 141       |

DYKS \* : Doyma, Yuvarlatma, Kaydırma, Saklama

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 sınırlanması gerekmektedir. Eğer  $K_C$  ve  $N$  16 değerine sınırlanırsa,  $K_CN = 256$  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.

Denetleyicinin 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. İntegral teriminde, tükenme önleyici sıfırlama kullanılması, integral 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 fonsiyonundan  $u(n)$  `i işle  
 $u(n)$  çıkışını üret  
I integral yapısını işle  
Taşmayı kontrol et ve kelime olarak I `i sakla

#### PID `ye GİT

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 katsayıları 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.

**TABLO 7.2 : PI ve PID Kontrol Çevrim Sayıları ve Maksimum Örnekleme Frekansı**

| ELEMEN    | PI            |     | PID           |     |
|-----------|---------------|-----|---------------|-----|
|           | Çevrim Sayısı | KHz | Çevrim Sayısı | KHz |
| TMS32010  | 94            | 53  | 145           | 34  |
| TMS320C14 | 94            | 66  | 145           | 43  |
| TMS320C25 | 89            | 112 | 141           | 70  |

PID kontrol 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.

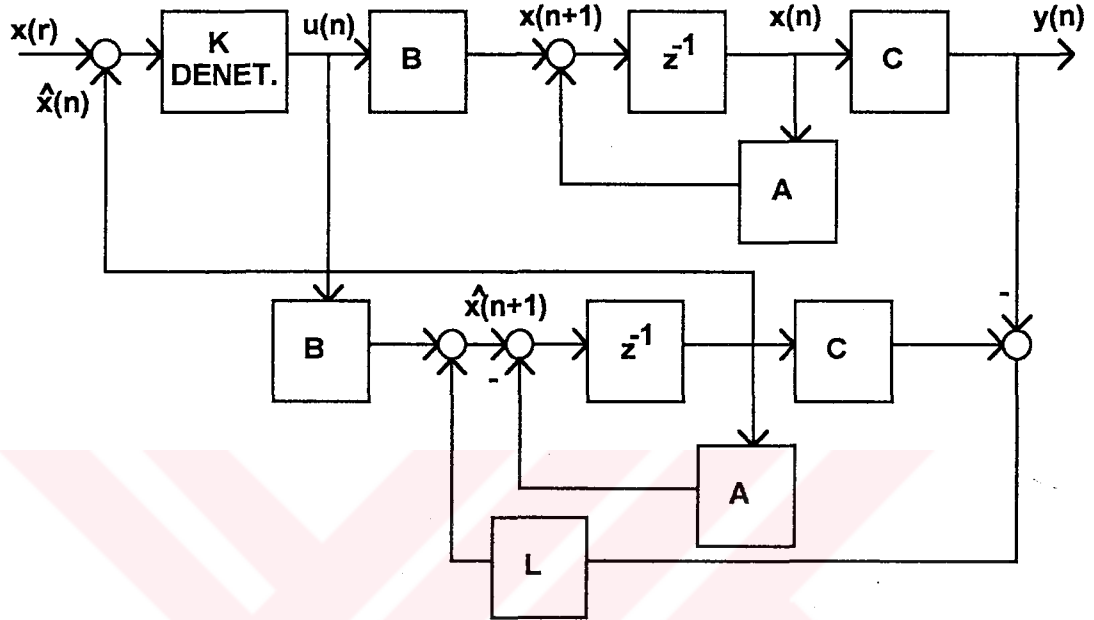
Adaptif bir sistemde klasik kontrol sistemlerindeki gibi sistemin matematiksel modeli tam olarak bulunmamaktadır. Bunun yanında, sisteme uygulanan deneyler yardımıyla, bu deneylerin sonuçları kullanılarak bir denetleyici yapısı belirlenmektedir. Sistemin kontrolünün sağlanabilmesi için, bir veya daha 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.

$$\mathbf{x}^{\wedge} (n+1) = \mathbf{A} \mathbf{x}^{\wedge} (n) + \mathbf{B}u(n) + \mathbf{K}C [\mathbf{x}^{\wedge} (n) - \mathbf{x}(n)] \quad [16]$$

$$\text{Hata : } \mathbf{e}(n) = \mathbf{x}^{\wedge} (n) - \mathbf{x}(n) \quad \mathbf{e}(n+1) = (\mathbf{A} + \mathbf{K}C) \mathbf{e}(n)$$

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



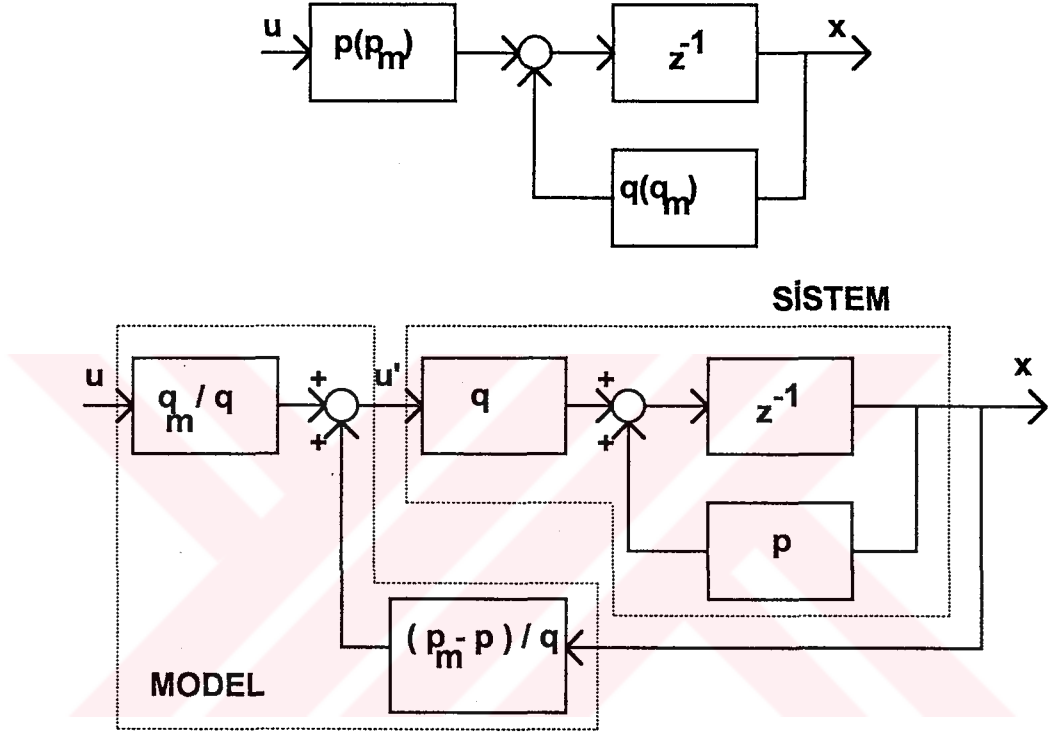
**ŞEKİL 7.4 : Gözlemleyici ve Durum Geri Besleme Kontrolü**

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.

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, ölçülebilen 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,

$$z(k) = y(k) + v(k) \quad y(k) = \sum h(m)u(k-m) \quad [17]$$

şeklinde hesaplanmaktadır.  $h(m)$  terimi,  $z(0)$ ,  $z(1)$ , .....  $z(N)$  değerleri kullanılarak  $z = Uh + v$  yapısı kurularak belirlenebilir. Bu değerden ters matris işlemi ile;  $h = (uTv)^{-1} vTz$  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.



**ŞEKİL 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;

$$p = e^{-T_c/T_s} \quad q = K(1 - p) \quad [18]$$

$$p_m = e^{-T_{cm}/T_s} \quad q_m = K_m(1 - p_m)$$

olarak tanımlanmaktadır.  $K$  ve  $K_m$  kazanç,  $T_c$  ve  $T_{cm}$  sistemin ve referans modelin zaman sabitleri,  $T$  örnekleme periyotudur. 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;

$$p(k+1) = p(k) + K_p x(k) e^*(k+1) \quad [19]$$

$$q(k+1) = q(k) + K_q U'(k) e^*(k+1)$$

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

$$e^*(k+1) = \frac{p_m e^*(k) + x(k+1) - (p(k)x(k) + q(k)U'(k))}{1 + K_p x^2(k) + K_q U'^2(k)} \quad [20]$$

$$U(k) = \frac{(p_m - p(k+1)x(k+1) + q_m U(k+1))}{q(k+1)}$$

formülleri yardımıyla hesaplanabilir. Adaptif kontrol sistemi tasarımı, 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.

## BÖLÜM 8

### ANAHTARLAMALI RELÜKTANS MOTORUNUN KONTROLÜ

Son yıllarda elektrik makinaları ve güç elektroniğindeki gelişmeler, uzun yıllardır kullanılmakta olanlar yerine geçebilecek yeni motorların tasarımını ve kontrolünü sağlamıştır. Elektronik alanındaki gelişmeler, yapıları uzun zaman önce bilinen fakat gerektiği gibi kontrol edilemeyen sistemlerin geliştirilmesine olanak sağlamıştır. Yeni tür motorların kullanıma girmesi ile, doğru akım motorlarının ve asenkron motorların geniş uygulama alanı azalarak, yeni alternatifler ortaya çıkmıştır.

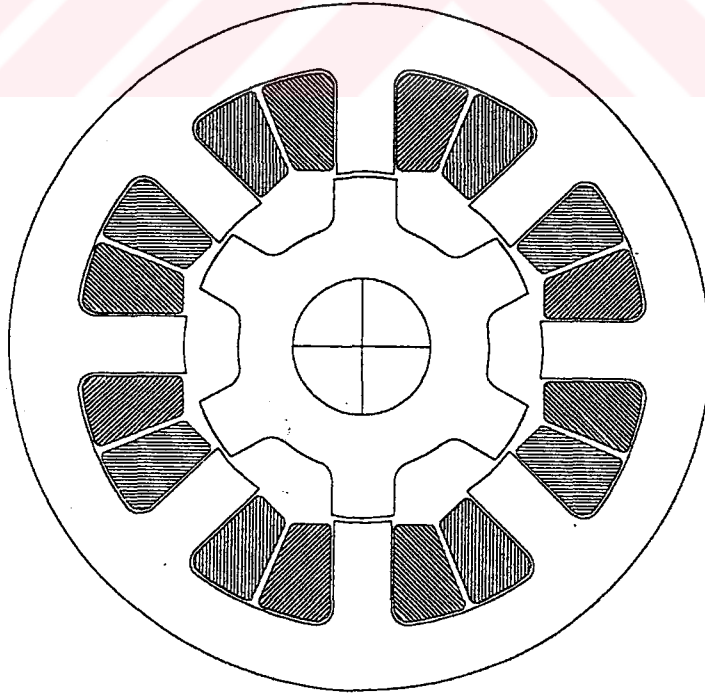
Doğru akım makinalarının üstün özellikleri; kolay kontrolü, kararlı yapısı ve moment ile akım arasındaki doğrusal ilişki bulunması iken, fırça kollektör düzeneğinin getirdiği problemler kullanım alanını kısıtlamaktadır. Asenkron motor, doğru akım motorlarına göre çok daha sağlam bir yapıda olmasına rağmen, doğrusal olmayan yapısından dolayı kontrolü çok daha karmaşık olmaktadır. Moment hız karakteristiğini, doğru akım motoruna benzer şekile getirmek için, frekans kontrolü yerine vektör kontrolü uygulanmakta, böylece çok daha kapsamlı ve zor uygulanabilir bir kontrol sağlanmaktadır.

Doğru akım motoru ve asenkron motorun üstün özelliklerini bünyesinde toplayan fırçasız doğru akım motoru ( FDAM ), sürekli mıknatıslı senkron motor ( SMSM ), ve anahtarlama relüktans motoru ( ARM ) güç elektroniği ve kontrol sistemlerindeki gelişmeler ile geniş uygulama alanı bulmaya başlamıştır. FDAM ve SMSM, doğru akım motoru gibi, doğrusal moment hız karakteristiği içermekte ve fırça kollektör düzeni bulunmadığından çok daha kararlı ve uzun ömürlü bir çalışma göstermektedir. Kendisine özgü yarar ve sakıncaları bulunan bu motorların, güç elektroniği elemanları ve kontrol devreleri geliştikçe daha geniş uygulama alanları bulacakları açıktır.

### 8.1. Anahtarlama Relüktans Motoru :

Anahtarlama relüktans motoru (ARM), yapı olarak diğer motor türlerinden ayrılmaktadır. Hem statoru hem de rotoru çıkık kutuplu yapıda bulunan bu motorun, diğer motorlardan farklı bir çalışma prensibi bulunmaktadır. Statorda bulunan çıkık kutuplar etrafına yerleştirilen sargılara gerilim uygulandığında, stator, rotor kutbunu kendine doğru çekecektir. Stator fazlarına belirli bir sıra ile anahtarlama yapılırsa, rotorda bu sıraya uygun yönde dönüş hareketi yapacaktır. Statora yerleştirilen faz sargıları, karşılıklı kutuplara yerleştirilerek, oluşturdıkları akı birbiri ile toplanmaktadır.

ARM bir, üç veya dört fazlı olabilir. Stator ve rotor kutup sayısı birbirine eşit olacağı gibi, en çok kullanılan yapılar, 6/4, 8/6, 4/2, 12/8 kutuplu olanlardır. Bununla beraber uygulamaya yönelik değişik stator ve rotor kutup sayıları da bulunmaktadır. Rotor ve statorun çıkık kutuplu olmasından dolayı, sistemin endüktansı rotor konumuna ve faz akımına bağlı bir fonksiyon şeklindedir. Bu durum da sistemin matematiksel modelinin elde edilmesini zorlaştırmaktadır.



ŞEKİL 8.1 : 8/6 Anahtarlama Relüktans Motorunun Yapısı

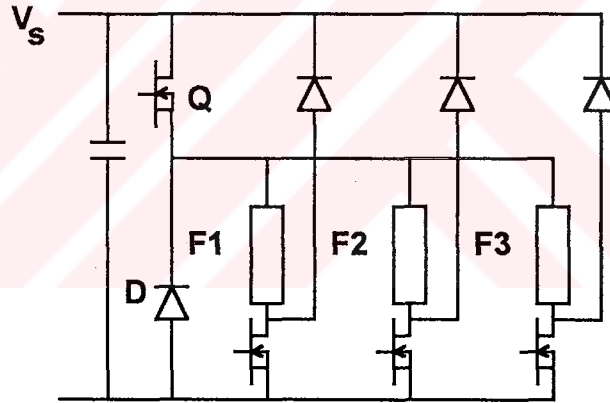
Stator kutbuna yerleştirilen faz sargısı, maksimum akıyı oluşturacak şekilde karşılıklı yerleştirilmiştir. Sargıya gerilim uygulandığında, stator kutbu, rotor kutbunu kendine çekmekte, hava aralığı minimum olmakta ve maksimum endüktans elde edilmektedir. Stator ve rotor kutupları karşılıklı çakışmaktadır. Bu durumda akı, minimum yoldan tamamladığından, magnetik relüktans en küçük değerinde olmakta ve moment elde edilememektedir. Motor, endüktansın yükselmesi yönünde moment üretmektedir. Rotor dönerken faz akı yoğunluğu üçgen veya testere dişi şeklini almaktadır fakat akıma bağlı olarak değişmemektedir.

Çakışma durumunda, akı, doyma moduna yakın çalışmaktadır. Doyma moduna girme sırasındaki akım, akı yolunun uniform olduğu kabulü ile, kestirilebilir. Rotor ile stator kutupları birbirinden tam ayrılmış durumda iken, faz endüktansı minimum seviyededir. Bunun nedeni, stator ve rotor arasındaki hava aralığı en büyük değeri aldığı anda, akı yolunun magnetik relüktansının en yüksek değerinde olmasıdır. Tam bu konumda moment minimum değerdedir. Rotor, endüktansın dönüş yönünde azaldığı bir konumda iken, akım akması durumunda, moment negatif olacaktır. Motordan pozitif moment, rotor maksimum hava aralığı durumundan, bir sonraki kutupların çakışması durumuna geçiş sırasında elde edilmektedir. Eğer, rotor ve stator kutupları simetrik ise, her faz ancak rotor kutup adımının yarı uzunluğunda moment üretebilir.

ARM `nun çalışma prensibi ve yapısı, sistemin kapalı çevrim kontrolünü zorunlu kılmaktadır. Sistemden gereken momentin alınabilmesi için herhangi bir anda rotorun nerede olduğu ve hangi fazın ateşleneceğinin bilinmesi gerekmektedir. Bu da, rotorun konumunun algılanması ile mümkündür. Rotordan alınan konum geri beslemesi ile, rotor konumuna bağlı olarak, faz akımlarının ateşlenmesi eş zamanlı hale getirilmektedir. Rotorun miline bağlanan algılayıcı yardımıyla, rotorun pozisyon ve hız bilgisi algılanarak, gereken ateşleme işaretleri üretilmektedir. Kullanılan algılayıcı tipine göre, eğer gerekiyorsa, belirli bir donanım yapısı ile işaretlerin daha anlamlı olması sağlanabilir.

## 8.2. Güç Elektroniği Devresi ve Sürücü Devrenin Özellikleri :

Motor momentinin yönü, akım ve akı yoğunluğunun işaretine ve değerine bağlı değildir. Sadece  $dL/d\theta$  'nin yani endüktansın rotor ile değişim oranının işaretine bağlıdır. Bunun yararı, akımın ve akı yoğunluğunun unipolar yapıda olmasına izin vermesidir. ARM 'yi kontrol etmek için değişik güç elektroniği devre yapıları önerilmektedir. Kullanılan minimum anahtar sayısı, faz sayısının bir fazlası olmalıdır. Güç elektroniği devrelerinin en basit ve ucuz olanı, faz sargılarını birer anahtarın ve PWM işaretini tek bir anahtarın kontrol etmesidir. Örneğin 3 fazlı bir motor için, her bir faza bir anahtar, ve kontrol için bir anahtar sistemi denetleyebilir. Bu tür bir devre, ucuz olmasına rağmen, moment ve akımda oluşturduğu salınımlar nedeniyle uygulamada problem yaratmaktadır. Q anatarı diğer anahtarlara göre üç katı fazla çalışma oranı ile, çok daha fazla yüklenecek ve hata olasılığını arttıracaktır.



**ŞEKİL 8.2 : 3 Fazlı ARM Güç Elektroniği Devresi**

Bu devre yerine, klasik evirici modeli benzeri bir devre tercih edilmektedir. Bu devrenin özelliği, her bir fazın iki adet anahtar ile kontrol edilmesidir. Sargı, anahtarların arasına yerleştirilerek, aynı anda aynı faz üzerindeki iki anahtar da ilettime sokulup kontrol sağlanmaktadır. Alt kolda bulunan anahtarlar, PWM işareti üretmekte iken, üst koldaki anahtarlar iletim veya kesim kararını vermektedir. Bu tür bir devre, moment ve akım açısından en iyi performansı göstermesine rağmen, her bir faz için iki adet anahtar gerektiğinden, pahalıya mal olmaktadır. Bu devre ile en yüksek kontrol

1. *Journal of Management Studies*, 1996, 33, 1, 1-14.

tranzistörde olduğu gibi, IGBT `nin iletim kayıpları düşüktür. Bu da MOSFET `e göre daha yüksek akım ve gerilim değerlerinde üretilmesini sağlamaktadır. Buna karşılık, anahtarlama frekansları MOSFET `e göre daha düşüktür. Bu elemanlar yanında, yeni geliştirilmekte olan MCT ( MOS Controlled Thyristor ) yüksek anahtarlama frekanslarına ulaşmakta, çok yüksek akm ve gerilim değerlerinin çok düşük kapı gerilimi ile kontroluna olanak sağlamaktadır.

MOSFET elemanlarının iletim ve kesime geçirilmesi eleman kapasitelerinin dolma ve boşalma hızlarına bağlıdır. Bu yüzden anahtar elemanı sürmek için tasarlanan devrenin belirli şartları sağlaması gerekmektedir. Yüksek hızda anahtarlama, anahtarlama kayıplarını azaltmasına rağmen, elektromanyetik gürültü ve geri besleme diyodunda ters iletim akımının artması gibi sorunlara neden olacaktır. Kontrol devresinden elde edilen işaretler, belirli bir seviyenin üstüne çıkamadığından, anahtarın sürülmesi mümkün olmamaktadır. Bu yüzden kontrol devresinden sonra, sürücü devre ile anahtar için gereken akım ve gerilim değerleri elde edilmelidir.

Uygulamaya göre, kullanılması gereken sürücü devrenin yapısı değişmektedir. Çok yüksek frekansta anahtarlama frekansı gerektiğinde, tasarlanan devre, iletim ve kesim işlemlerini yeterli hızda yapabilecek kapasitede olmalıdır. Kontrol devresinden gelen işaret, kuvvetlendirilerek, kapı devresine uygulanmalıdır. Referans gerilim değerinin, kontrol işareti ile karşılaştırılması ile elde edilen işaret, anahtarı ilettime geçirecektir. Bununla beraber, anahtarın kesime geçirilmesi için, kapı devresinin geriliminin kesilmesi ve akımın önlenmesi yeterli olmamaktadır. Kapı ile kaynak arasına kısa süreli ters gerilim uygulanması, elemanın kesime geçmesini garanti altına alacaktır. Bunu sağlamak amacıyla, ortak bazlı, ard arda bağlanmış npn ve pnp tipi tranzistörlerden yararlanılmaktadır. (Şekil 8.4). Güç elemanın ilettime geçirilmesi sırasında, kontrol işareti, npn tranzistör ile kuvvetlendirilip, kapı direnci üzerinden elemana uygulanmaktadır. Kesime geçirmek için kontrol işareti kesildiğinde, pnp tipi tranzistör ilettime geçerek, kapı direnci üzerinden gerilimin boşalmasını sağlayarak, elemanın hızla kesime gitmesini sağlamaktadır.



Gerçekleştirilen sürücü devrede, lojik yapıyı ayrı elemanlardan kurmak yerine, ters gerilim uygulayabilen ve kapı gerilimini sabit tutan bir eleman kullanılmıştır. MDC1000A olarak adlandırılan bu eleman, ek pasif elemanlara gerek kalmadan, MOSFET 'e kapı işaretlerini uygulamakta ve kesime geçirmektedir.

Kontrol işareti, lojik TTL seviyesinde bir işaret olduğundan, sürmek için gereken akım ve gerilim değerlerini elde etmek için kuvvetlendirilmelidir. Kontrol işareti ile güç sürücü devre arasında elektriksel olarak yalıtımı sağlamak için optokuplör elemanı kullanılmıştır. Böylece, kontrol ve sürücü devreleri ayrı kaynaklardan beslenmekte ve ayrı toprakları bulunmaktadır. Kontrol işareti bir seviyesine ulaştığında, işaret evirildiğinden, tranzistörden baz akımı akmamakta, böylece optokuplör iletimde bulunmaktadır. Bu durumda MOSFET iletime geçmektedir. Kontrol işareti sıfır olduğunda, evirilerek, tranzistörden baz akımı akmasını sağlayacak, böylece optokuplör kesimde kalacaktır.

### 8.3. Kontrol Sistemi :

ARM motorunu sürmek için kullanılan güç elektroniği devresi tanıtıldıktan sonra, kontrolüne ilişkin özellikler verilecek ve kurulan yapı incelenecektir.

ARM, diğer motorlarla karşılaştırıldığında, çalışma prensibi daha basit gibi gözükse de, kontrol yönünden daha karmaşık bir sistemdir. ARM 'de matematiksel modelin elde edilmesi diğer motorlara göre çok daha karmaşık işlemler gerektirmektedir. Doğru akım motorunun kontrolü sırasında yük ile motorun matematiksel modeli kurularak, kontrol yapısı tasarlanmaktadır. Asenkron motorun vektör kontrolünde ise, zamanla değişen rotor değişkenleri, d ve q eksenleri tanımlanarak, doğrusal yapıya uygun bir model haline getirilmektedir. Motorun d ve q eksenlerine indirgenmesi, kontrol değişkenlerini karmaşık doğrusal olmayan yapıdan kurtararak, doğrusal yapıya sokmaktadır. ARM 'de ise, moment ile akım arasında doğrusal bir ilişki olmadığından ve asenkron motorda olduğu gibi dolaylı yoldan da elde edilemediğinden, tasarlanan kontrol, karmaşık bir hal almaktadır. Bunun nedeni, moment, akım, hız ve ateşleme açıları arasındaki lineer olmayan

ilişki, ve her bir değişkenin yükün ve hızın bir fonksiyonu olmasıdır. Bu yüzden sistemin kontrolü, faz akım ve geriliminin gerçek değerleriyle işlem yapabilen, gerçek zamanda kontrole uygun bir yapının kullanılması ile gerçekleştirilmektedir.

ARM kontrolunda sistemden beklenen özellikler, çok düşük moment darbeleri, dinamik cevabının hızlı olması, kararlılığının iyi olması, sıfır hızda çalışabilmesi ve dört bölgeli çalışma karakteristiğine izin vermesidir. Motor değişkenlerinin yüke ve hıza çok bağımlı olması, sistem tasarımı sırasında yükün hız ve moment bilgisinin belirlenmesi ve algoritmanın bu yönde kurulmasını gerektirmektedir. Motor, yüksek hızlara çıkarılmak istendiğinde, akım kontrolü gereken cevabı vermemektedir. Bunu sağlamak için, ateşleme açısının kontrolü kullanılmaktadır. Moment kontrolü için, ateşleme açıları, fazın ateşlenmesi gereken zamandan önce uygulanmaktadır. Algılayıcıdan rotorun konumuna dair gelen bilgi değerlendirilerek, ilişkin faz erken ateşlenmekte böylece yüksek hızlara düşük moment salınımları ile çıkılabilmektedir. Hızın fonksiyonu olarak, ateşleme açılarının değerlerinin belirlenmesi lineer olmayan ve karmaşık bir işlemdir. Optimum ateşleme açılarının belirlenmesi ancak yükleme deneyleri ile mümkün olmaktadır. Bunun yanında, ARM motor için geliştirilmiş simulasyon programları, bu alanda büyük destek sağlamaktadır.

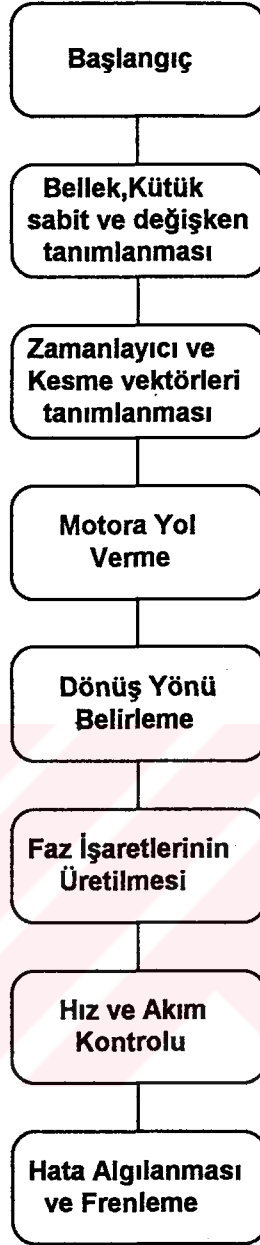
ARM sistemin kontrolü için özel tümdevreler geliştirilmiştir. Bu devreler yardımıyla, motor belirli hız aralıklarında ve moment değerleri ile çalışması sağlanmaktadır. Bu kontrol devreleri kullanılarak, motorun temel kontrolü kolaylıkla gerçekleştirilebilir. Analog yapıda olan bu devreler yanında, dijital tüm devreler de geliştirilmiştir. Dijital bir giriş çıkış birimi üzerinden haberleşebilen sistem, motorun tam kontrolünü gerçekleştirmektedir. Böyle bir sisteme örnek, Hewlett Packard firması tarafından geliştirilen HCTL-1100 tümdevresidir. HCTL-1100, doğru akım, step, FDAM ve ARM motorlarının kontrolünü gerçekleştirebilmektedir. Motor hızı istenilen aralıkta ayarlanabileceği gibi, belirli bir hız profilini takip etme işlemi de kolaylıkla gerçekleştirilmektedir. Motordaki moment salınımlarını minimum seviyeye indirmek ve gürültü problemlerini ortadan kaldırmak amacıyla, gereken düzeltmeler ateşleme açıları ayarlanarak yapılabilir. Buna rağmen, sistemin iç yapısına karışmadığından, değişik uygulamalarda sorun olabilir. Bu

durum, yazılan program ile aşılabılır. Örneğin, sistem, motorun iki yöne gereken hareketini gerçekleştirirken, kontrol yapısı dört bölgeli çalışma için uygun değildir.

Sistemin tam olarak kontrolunu istenen özelliklerde sağlamak için, dijital denetleyici yapısından yararlanılabilir. Mikroişlemci ile gerçekleştirilen bir kontrol devresi, sistemin yük ve hız değişimlerine gereken uyumu sağlayabilecek durumda olacaktır. Sistem istekleri değiştiğinde, esnek yapısından dolayı, kolaylıkla uyumlu hale getirilebilir. Sistem kontrolunun karmaşıklaşması, modern kontrol algoritmalarının uygulanabilmesi ancak bu tür bir yapı ile mümkün olmaktadır.

Motor denetleyici birimi, ister mikroişlemci, ister DSP tabanlı bir sistem olsun, gereken yapının kurulması ancak belirli bir algoritmanın programlanması ve işletilmesi ile sağlanmaktadır. Kurulan algoritma, sistem isteklerini tam anlamıyla karşılayabilecek kadar esnek yapıda olmalıdır.

ARM kontrolunu gerçekleyen algoritma, sistem değişkenlerinin tanımlanması, motora yol verme, PWM işaretinin üretilmesi ve hız kontrolunun sağlanması alt bölümleri olarak tanımlanabilir. Motor kontroluna başlamadan önce denetleyici birimi buna hazırlanmalıdır. Denetleyicinin bellek alanı tanımlanarak, programın ve kullanılacak verilerin saklanması için yer ayrılmalıdır. Bellek biriminin yapısına göre, program örneğin, EPROM'a programlanabilir. Kullanılacak sabitler belirli bir tablo oluşturularak, saklanabilir. Eğer sabitler sınırlı sayıda ise ve işlemcinin kütük dosyası yeterince geniş ise bu alanda saklanabilir. Program, işlemciye yüklendikten sonra, başlangıç değerlerinin tanımlanması gerekmektedir. Örneğin, yığının boyu ve belleğin hangi alanında yer alacağı, zamanlayıcı birimlerinin çalışma şekli, kesme vektörleri ve kesmenin hangi koşullarda kabul edileceği gibi bilgiler işlemciye bildirilmelidir. Daha sonra, zamanlayıcı ünitesi kullanılarak, PWM işareti üretecek zamanlayıcı alt programı gelecektir. Sistem kapalı çevrim çalıştığından, algılayıcıdan gelen bilginin işlemcinin anlayacağı hale getirilmesi gerekmektedir. Ana program, ARM'nin çalışma özelliklerine göre belirlenecektir.



**ŞEKİL 8.6 : ARM Kontroluna İlişkin Akış Diyagramı**

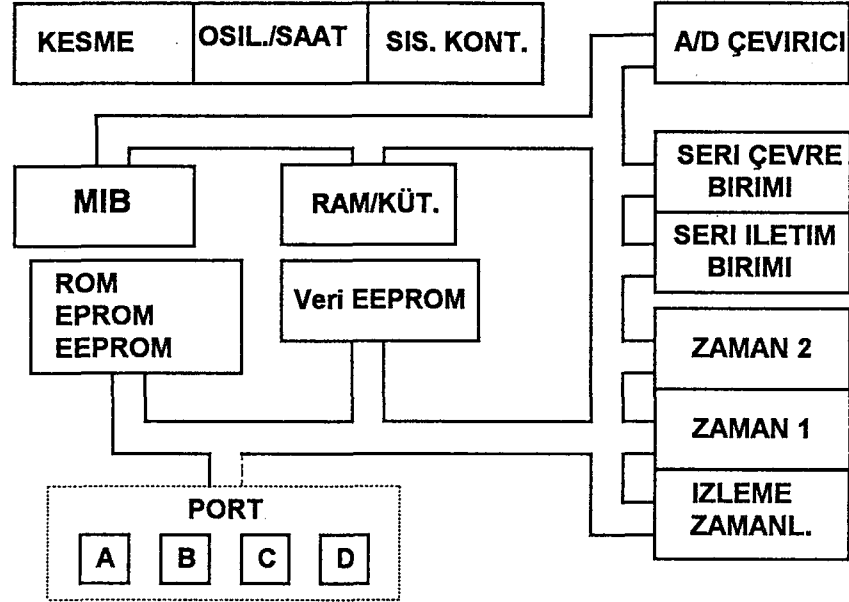
Motora yol verme, kontrolün ana problemlerinden biridir. ARM 'nın çalışması için, hangi fazın ateşleneceğinin bilinmesi gerekmektedir. Motor normal işlerken, miline bağlı algılayıcı sayesinde konumu bilmektedir. Fakat ilk harekete başlarken, rotorun hangi stator kutbuna yakın olduğu bilinmediğinden, harekette bir kararsızlık olacaktır. Bunu önlemek amacıyla, algılayıcıya rotorun konumu öğretilmelidir. Bunun için, motor ilk olarak açık çevrim çalıştırılarak, belirli bir faza gerilim uygulanmalıdır. Daha sonra faz sayısına göre, son faza gerilim uygulanarak rotor kilitlenmelidir. Daha sonra

tekrar ilk faza gerilim uygulanarak, algılayıcının rotor konumunu öğrenmesi sağlanmaktadır.

Motoru kontrol eden PWM işaretleri, zamanlayıcı birimi tarafından üretilmektedir. PWM değerine göre, ana program kesme isteği alarak, zamanlayıcı biriminin PWM işareti üretmesini sağlamaktadır. Kesme isteğinin kullanımı kontrol edilerek, işaretin periyodu ayarlanabilir. Motorun faz sayısına göre, PWM işareti, hangi fazın iletme geçeceğini belirleyen işaretler ile lojik işleme sokularak, motoru sürecektir kontrol işaretleri elde edilmektedir. Motor kontrolü gerçek zamanda ve gerçek işaretler ile yapıldığından, kurulan algoritmanın işleme süresinin ve kontrol işaretlerinin üretilme süresinin büyük önemi vardır. Bu yüzden, algılayıcı işaretini değerlendiren ve PWM işaretini üreten kesmelerin bu durum göz önüne alınarak programlanması gerekmektedir.

Sistemin yardımcı birimleri, geri besleme işaretlerinin kolaylıkla değerlendirilmesini sağlamaktadır. A/D çevrici yardımıyla, gerekli fiziksel veriler denetleyici birimi tarafından algılanmaktadır. Motorun akım bilgisi, A/D çeviriciye girilerek, akım değerinin belirli bir değeri aşmaması sağlanabilir. Bunun için, akımın belirli bir katsayı ile ölçeklendirilmesi gerekmektedir. Denetleyici sistemi kurulduğunda, kullanıcının değişkenleri gözlemlemesi ve gerektiğinde müdahale etmesi için, bir şekilde değişkenlerin gösterimi gerekmektedir. Bu da seri ve paralel port kullanılarak, yardımcı birimlerin kontrol edilmesi ile sağlanabilir. Denetleyici biriminin, bu şekilde yardımcı birimleri kontrol edecek işaretleri üretmesi gerekmektedir. Sistemde bir hata oluştuğunda, sisteme zarar gelmesini önlemek için, kontrol işaretlerini kesmeli ve gereken uyarıyı yapan düzeneği harekete geçirmelidir. Bu özellikleri yapısında toplayan işlemcinin uygulama alanında sağlam bir yer edineceği açıktır.

ARM motorunun dijital denetleyici ile kontrolü için mikroişlemci veya DSP sisteminden yararlanılabilir. Mikroişlemci yerine çevre birimlerini yapısında bulunduran mikrobilgisayar yapısı tercih edilmiştir. Böylece ek tümdevreler ile destek sağlamaya gerek duyulmadan tek işlemci ile, bellek, zamanlayıcı ve A/D çevirici birimleri sisteme kazandırılmıştır.



**ŞEKİL 8.7 : TMS370 Mikroişlemci Blok Diyagramı**

Sistemin gerçekleştirilmesinde kullanılan mikroişlemci, TI tarafından üretilen TMS370 mikroşlemcisidir. Çevre ve bellek birimlerini tek bir tümdevre içinde toplayan bu mikrobilgisayar yapısı, 8 bitlik işlemciler içinde en gelişmiş olanıdır. 8 bitlik diğer işlemciler ile karşılaştırıldığında, çok daha hızlı, daha kapsamlı çevre birimlerine sahip ve yazılım desteği en iyi olan işlemci olduğu görülmektedir. 200 ns komut çevrim süresi ile, diğer mikroşlemcilerden çok daha hızlı işlem yapabilmektedir. Üretilen 370 ailesinin her elemanı değişik bir isteğe cevap vermekte, böylece genel kontrol yapısı değiştirilmeden sadece uygulamanın gereklerini karşılayacak yapı kurulabilmektedir. Bellek birimleri, her modelde değişik biçimlerde ve değişik boyutlardadır. RAM bellek 256 veya 512 byte genişlikte iken, program belleği, ROM, EPROM, EEPROM yapıda ve 4K, 8K veya 16K uzunluğunda olabilir. İşlemcinin MIB yapısı, durum kütüğü, program sayacı ve yığın göstergesini içermektedir. 8 bit uzunluğundaki yol ile bellek ve çevre birimlerine ulaşılabilir. Mikrobilgisayar özelliğinden dolayı çok geniş ve kapsamlı çevre birimleri bulunmaktadır.

Kontrolün temel birimlerinden biri olan zamanlayıcı birimi, iki temel zamanlayıcı ve bir gözlemleyici zamanlayıcı ile gerçek zamanda kontrolü desteklemektedir. Zamanlayıcı birimleri, olay sayma, değer ile karşılaştırma, darbe üretme, köşe algılama gibi işlemlerde kullanılmaktadır. Bu işlemlerin

sonucuna bağılı olarak, kesme işareti üretilebilir, durum bayrağı ayarlanabilir, zamanlayıcı sayıcısını sıfırlayabilir, G/Ç işareti yollayabilir veya PWM çıkışı üretebilir. 370 işlemcisinin Cx3x ailesinde PACT (Programmable Acquisition and Control Timer) adı verilen çok amaçlı bir zamanlayıcı birimi de bulunmaktadır. Bu yapı, çok daha kapsamlı fonksiyonları ile tek başına bir işlemci gibi çalışmaktadır.

Tümdevre içinde bulunan bir diğer yapıda A/D çeviricidir. Bu birim yardımıyla, işlemcinin fiziksel dünya ile doğrudan haberleşmesi mümkündür. 8 kanallı 8 bitlik çeviricinin kaynak ve giriş kanalı seçilebilmektedir. İşlemcide bulunan diğer birimlerde SCI (Seri İletişim Bağlantı Birimi) ve SPI (Seri Çevre Bağlantı Birimi) birimleridir. Bu birimler, diğer yapılarla seri port üzerinden haberleşerek, veri işlenmesini ve görüntülenmesini mümkün kılmaktadır.

İşlemcinin komut seti, 73 adet komuttan oluşmakta ve 14 adet adresleme modu bulunmaktadır. Birkaç komut ile gerçekleştirilecek işlemleri tek komutta işleyen özel komutlar programlamayı kolaylaştırmaktadır. Değişik adresleme modlarının kullanılması, sistemin en verimli şekilde veri işlenmesini ve veri aktarımını sağlamaktadır. Özellikle 16 bitlik verilerin işlenmesi için geliştirilen adresleme modları veri kaybını ve hatayı önlemektedir. İşlemcinin 55 adet giriş çıkış pini bulunmaktadır. G/Ç pinlerinin bu kadar fazla olmasını sağlayan, normal işlevlerinde kullanılmayan birimlerin pinlerinin genel amaçlı G/Ç pini olarak tanımlanabilmesidir. Çevre birimlerinin kendi kesme işaretlerinin yanında üç adet dış kesmeye olanak sağlamaktadır. Ayrıca işlemcinin tam kapasite yerine düşük güçte çalışma modları bulunmaktadır.

Bu özellikleri ile TMS370 mikroişlemcisi, kontrol sistemlerinde geniş uygulama alanı bulmaktadır. Ayrıca, geliştirme modülünün sağladığı yazılım desteği ile örnek sistemlerin tasarımı mümkün olmaktadır. Programların koşturulabileceği ve gerçek zamanda denenebileceği işletim ortamı sayesinde tasarım kolaylaşmıştır. Programlama assembler veya C dilinde yapılabilmektedir. Programın işlenmesi sırasında bellek ve kütük değerlerinin gözlemlenebilmesi, hataları ortaya çıkarmakta ve programlamayı kolaylaştırmaktadır.

Uygulama aşamasında ARM kontroluna ilişkin program hazırlanarak, yazılım ortamında koşturulmuş, gerçek zamanda sisteme uygulanarak motorun ateşleme açıları üretilmiş, PWM işareti değiştirilerek motorun hız kontrolu gerçekleştirilmiştir. Mikroişlemcinin kütük yapısı motora ilişkin değişkenlerin saklanması ve gözlenmesini sağlamaktadır. Oluşturan veri tabloları, bellek içinde saklanarak, gerekli işlemlerde tekrar tekrar üretilme yerine, bellekten alınarak işletilmesi yoluna gidilmiştir. İşlemcide iki adet birbirinden bağımsız zamanlayıcı biriminin bulunması tasarımı kolaylaştırmıştır. İlk zamanlayıcı ile algılayıcıdan gelen işaretler işlenerek, motorun hız ve konum bilgisi elde edilmiş, ikinci zamanlayıcı ise PWM işaretini üretmek amacıyla kullanılmıştır. A/D çevirici ile motorun akım bilgisi algılanarak, belirli değeri aştığında sistemin ateşleme açısı üretme işlemini durdurması sağlanmıştır. Herhangi bir hata durumu, dış kesme işaretlerini tetikleyerek sistemin susmasını sağlamaktadır. Böylece motorda olası bir hataya karşı önlem alınarak güvenli çalışma gerçekleştirilmektedir.

ARM 'nın kontrolunda kullanılan DSP yapısı ise, TMS320C14 işlemcisi kullanılarak gerçekleştirilmiş Power-14 kontrol modülüdür. TMS320C14 DSP, 25.6 MHz, 160 ns komut saat çevrim süresi, 6.4 MIPS işlem kapasitesi, 256 byte RAM, 4K ROM bellek, 4 zamanlayıcı/sayıcı, ve seri port ile kontrol sistemlerine uygun bir işlemcidir. 32 bitlik sayılar ile işlem yaparken, taşma koruması ve ölçeklendirme işlemlerini gerçekleştirmektedir. Sistem tasarımında, tek başına bir işlemciden çok bu şekilde çevre birimleri ile donatılmış bir işlemci, ek donanım desteğini ortadan kaldırdığından tercih edilmektedir. Özellikle kapsamlı zamanlayıcı üniteleri, motor kontrolunda gereken işaretlerin üretilmesine olanak sağlamaktadır. Yapısındaki olay yönetici ile, yakalama ve karşılaştırma işlemleri kontrol edilmektedir. Yakalama birimi, uygulamalarda çok kullanılan bir yapıdır. Örneğin, algılayıcıdan gelen geri besleme işareti, 160 ns doğruluk oranı ile işlenerek, konum ve hız bilgisinin elde edilmesinde kullanılmaktadır. Karşılaştırma birimi ise çıkış olarak işleyerek, örneğin 6 kanallı, 40 ns doğruluk oranı bulunan PWM denetleyici olarak kullanılabilir.

Power-14 modülü, kontrol uygulamalarında kullanılmak üzere TMS320E14 işlemcisini kullanmaktadır. Bu DSP, C14 işlemcisinin EPROM bellek içeren versiyonudur ve diğer bütün özellikleri C14 ile aynıdır. Power-

14 modülü, DSP sistemi ile, ana bilgisayar ile bağlantı için RS-232 portu, giriş çıkış lojji, 12 bit A/D çevirici ve güç katının birleşmesi ile oluşmuştur. C14 işlemcisinin özellikleri yanında, 8 kanal, 12 bit A/D çevirici yardımıyla, algılayıcı işaretlerinin işlenmesi mümkündür. 20  $\mu$ s çevirici ile %0.02 oranında yüksek doğruluk değerleri elde edilebilmektedir. Dört giriş ve dört çıkış modu sayesinde çok değişik uygulamalarda kolaylıkla kullanılabilir. Bu sayede birçok değişik motorun kontrolüne uyumlu hale getirilebilir. PWM çıkış komutundaki bit sayısı, anahtarlama hızını belirlemektedir. Bit sayısının artması, daha yüksek doğruluk oranı fakat daha düşük anahtarlama oranı anlamına gelmektedir. 10 bitlik bir doğruluk ile 25 KHz anahtarlama frekansı elde edilebilmektedir. Sistemin güç katı ise MOSFET anahtarlar kullanılarak, küçük güçlü uygulamalarda kullanılmak üzere tasarlanmıştır. Altı adet anahtarın kullanıldığı devre, akım algılama elemanları içermektedir. 12 bit uzunluğundaki akım geri besleme işaretleri herhangi bir hata algılanması durumunda gerekli önlemleri almakta ve moment kontrolü için gereken bilgiyi sağlamaktadır. Bu geliştirme modülü kullanılarak, doğru akım motoru, asenkron motor, adım motoru, FDAM veya ARM kontrolü gerçekleştirilebilir.

## BÖLÜM 9

### SONUÇ

Bu çalışmada, anahtarlama relüktans motorunun kontrolü için gereken güç katı, sürücü katı ve kontrol katı tasarlanmıştır. 6/4 kutup sayısında üç fazlı motorun kontrolü için MOSFET elemanları kullanılarak, 6 anahtarlı, bir fazın iki anahtar ile kontrol edildiği konfigürasyon tercih edilmiştir. Anahtarları sürmek için Şekil 8.5 'deki devre kurulmuş ve anahtarları gerektiği şekilde iletim ve kesime sokulabildiği gözlenmiştir. Bu devre modelinde anahtarların fazla olması maliyeti arttırmasına rağmen, tam ve esnek bir kontrol yapısına izin verdiğinden tercih edilmiştir.

Motorun hız ve konumunu algılamak için, motorun miline bağlı, 360 darbeli algılayıcı kullanılmıştır. Artan algılayıcı olarak adlandırılan bu yapı, motorun hızı ve konumu ile orantılı üç işaret üretmektedir. Algılayıcının çıkışı, birbirine 90° kaymış kare dalga iki işaret şeklindedir. Bu işaretler doğrudan konumu vermek yerine, dönüş açısı ile orantılı bir frekansta olduklarından ancak hesap yöntemi veya ek donanım ile gereken bilgi edilebilir. Diğer işaret ise, motorun bir tam turunda bir çıkış vermektedir. Bu da işaretin motorun hızı ile aynı frekansta olduğu anlamına gelmektedir.

Motorun kutup sayısına bağlı olarak, güç elektroniği anahtarları geometrik olarak 30° iletimde kalarak ilgili fazdan akım akmasını sağlamaktadırlar. Anahtarların iletimde kalacağı açı değerleri, bir veri tablosu şeklinde işlemcinin belleğine yerleştirilmiştir. Kullanılan motor 6/4 kutup sayısı yerine 8/6 kutup sayısında olduğunda 15° iletim süresine karşılık gelen değerler veri tablosuna yerleştirilerek, ana programın değiştirilmeden kullanılması sağlanabilir.

Moment salınımlarını önlemek ve daha iyi performans elde etmek için fazın ateşlenme süresi ve yeri ayarlanabilmelidir. Bu da algılayıcıdan gelen indeks işaretinin değerlendirilmesi ile mümkündür. İndeks işareti, motor bir tam tur attığında oluşmaktadır. Bu işaret, denetleyiciye girilerek, sistemin kesme işareti alması ve motor hızının algılanması sağlanmaktadır.

ARM `nın kontrolunda mikroişlemcinin veya DSP `nin yapılarından hangisinin tercih edilmesi gerektiği sistemin isteklerine göre belirlenmelidir. Matematik olarak çok yüklü bir algoritma kullanılmadığında, mikroişlemci doğru seçim olacaktır. Mikrobilgisayar yapısındaki sistemle ek donanım ihtiyacı duyulmadan kontrol sağlanabilir. Mikroişlemci, DSP `ye göre daha ucuz ve programlanması daha kolay olduğundan, sistem gereksinimini karşıladığı sürece doğru çözümü sunacaktır. Bununla beraber, sistemin istekleri değiştiğinde ve karmaşılaştığında, mikroişlemcinin, DSP `nin üstün performansını göstermesi beklenmemelidir. Örneğin, motorun dört bölgesi, hız, akım ve moment kontrolunun yapıldığı bir yapı kurulmak istendiğinde, mikroişlemci gereken çözüm performansını yakalamaktan uzak kalacaktır. Bu aşamada DSP doğru çözüm alternatifi olmaktadır.

Sistemin en pahalı bölümlerinden biri, algılayıcı birimidir. Motorun algılayıcı kullanılmadan kontrolu yapılarak daha ucuz ve kapsamlı kontrolu sağlanabilir. Akım ve endüktans değişimi bilgisi değerlendirilerek, motorun konumunun ve hızının algılanması mümkündür. Bu işlem, motorun yüksek hızlarda çalıştırılması söz konusu ise, ancak DSP kullanılarak gerçekleştirilebilir. Matematik işlemlerdeki hızı DSP `yi bu gibi uygulamalarda üstün kılmaktadır.

ARM diğer motorlara göre farklı yapısı ile değişik uygulamalara uyum sağlayabilmektedir. Yüksek hızlara ulaşabilmesi, düşük akım değerlerinde yüksek moment motorun üstün özellikleridir. Bununla beraber moment salınımları ve gürültü en önemli sorunu oluşturmaktadır. Motordan yüksek bir performans alınması ancak iyi bir kontrol düzeneği ile mümkün olabilir. Diğer motorlara göre, daha az maliyeti bulunması, motorun uygulama alanını arttıran bir etkidir. Kontrol algoritmasının geliştirilmesi ve özellikle algılayıcısız sistemlerin kontrolu üzerine daha alınacak çok yol bulunmaktadır.

**EK-A :****TMS320C25 Fonksiyon Üniteleri :**

ACCH : Akümülatör Yüksek Byte  
 ACCL : Akümülatör Düşük Byte  
 ALU : Aritmetik Lojik Ünitesi  
 IFR : Kesme Bayrak Kütüğü  
 IMR : Kesme Maskeleme Kütüğü  
 PC : Program Sayacı  
 PFC : Sayaç  
 RPTC : Tekrar Komut Sayacı  
 IR : Komut Kütüğü  
 ARAU : Yardımcı Kütük Aritmetik Ünitesi  
 ARB : Yardımcı Kütük Göstergesi Kuvvetlendirici  
 ARP : Yardımcı Kütük Göstergesi  
 DP : Veri Bellek Sayfa Göstergesi  
 GREG : Dış Bellek Adresleme Kütüğü  
 MCS : Çağırma Kütüğü  
 QIR : K. Komut Kütüğü  
 PR : Sonuç Kütüğü  
 PRD : Periyot Kütüğü  
 TIM : Zamanlayıcı Birimi  
 TR : Geçici İşlem Kütüğü  
 AR0-AR7 : Yardımcı Kütük  
 ST0-ST1 : Durum Kütükleri  
 DRR : Seri Port Veri Alma Kütüğü  
 DXR : Seri Port Veri Yollama Kütüğü  
 RSR : Seri Port Alma Kaydırma Kütüğü  
 XSR : Seri Port Gönderme Kaydırma Kütüğü

**KAYNAKLAR :**

- [1] AHMED, Irfan ( Edited by ), " Digital Control Applications with the TMS320 Family ", Printed by Texas Ins.Inc.,1991
- [1.1] AHMED, Irfan, BUCELLA, T. " Taking Control with DSPs ", Machine Design. ( bkz. Kaynak [1], s : 19. ), 1989,
- [1.2] HANSELMANN, H. " Implementation of Digital Controllers - A Survey" Automatica, Vol.23, No: 1, ( bkz. Kaynak [1], s : 145.), 1987
- [1.3] MESHKAT, S. " Applications of Kalman Filtering in Motion Control Using TMS320C25 ". ( bkz. Kaynak [1], s : 185. )
- [1.4] ASTRÖM, K.J., " Implementation of a PID Controller on a DSP ", ( bkz. Kaynak [1], s : 205 )
- [2] ASH Dan, " Considering For Selecting a DSP Processor, (ADSP-21020 vs. TMS320C30) ", Application Report AN-235, Analog Devices., 1990
- [3] BOSE K. B., MILLER T. J., SZCZESNY P. M., BICKNELL W.H. " Microcomputer Control of Switched Reluctance Motor ", IEEE Annual Meeting, pp 542-547., 1985
- [4] BOSE K. B., " Power Electronics and AC Drives ", Prentice Hall, 1986
- [5] BOSE K. B., ( Edited by ), " Microcomputer Control of Power Electronics and Drives ", IEEE Press., 1987
- [6] BUCELLA T., " Comparing DSPs to Microcomputers in Motion Control Systems- Some Real World Data " , Intertec Comm. Inc., 1990

- [7] DHAOUADI R., MOHAN N., NORUM L. " Design and Implementation of an Extended Kalman Filter for the State Estimation of a PMSM ", Vol. 6, No: 3, IEEE Transactions on Power Electronics., July, 1991,
- [8] DOTE Yasuhiko, " Servo Motor and Motion Control Using Digital Signal Processors " , Texas Ins., Prentice Hall., 1990
- [9] HANSELMANN H., " Motion Control with DSP ", dSPACE GmbH., 1993
- [10] HARRIS W. D., LANG J. H., " A Simple Motion Estimator for Variable Reluctance Motors ", Vol. 26, IEEE Trans. on Industry App., March/April, 1990
- [11] HIGGINS R. J. " Digital Signal Processing in VLSI ", 1990, Prentice Hall
- [12] INGLE V. K., PROAKIS J., " Digital Signal Processing Laboratory Using the ADSP-2101 Microcomputer ", Prentice Hall., 1990
- [13] MAR Amy, ( Edited by ), " Digital Signal Processing Applications Using the ADSP-2100 Family ", Prentice Hall., 1990
- [14] MATSUI N., AKAO N. , WAKINO T., " High Precision Torque Control of Reluctance Motors ", IEEE Industry App. Soc. Conf., 1989
- [15] MATSUI N., OHASHI H., " DSP-Based Adaptive Control of Brushless Motor ", Vol. 28, No:2 IEEE Ind. App., March/April, 1992
- [16] MILLER, T. J. E., "Switched Reluctance Motors and their Control", Clarendon Press Oxford, 1993

- [17] MOHAN, N., "Power Electronics: Converters, Applications and Design", John Wiley & Sons, 1989
- [18] SEPE R. B., LANG J. H., " Real Time Adaptive Control of Permanent Magnet Synchronous Machines ", Vol. 27, IEEE Ind. App., July/Aug., 1991
- [19] ADSP-2100 User `s Manual, Architecture, Analog Devices, 1991
- [20] ADSP-2101 Manual, Analog Devices., 1990
- [21] ADSP-2105 Manual, Analog Devices., 1990
- [22] ADSP-21020 Manual, Analog Devices., 1992
- [23] Digital Signal Processing Applications with the TMS320 Family: Theory, Algorithms, and Implementations, Vol. 1, 1989, Vol. 2, 1990, Vol. 3, 1990, printed by Texas Instruments Inc.
- [24] POWER-14 User Manual , Teknic Inc.,1989
- [25] TMS320 Educators Conference, Proceedings, sponsored by Texas Ins., 1991
- [26] TMS320C1x Users Guide, Texas Instruments, 1990
- [27] TMS320C2x Users Guide, Texas Instruments., 1994
- [28] TMS320C3x Users Guide, Texas Instruments., 1993

## ÖZGEÇMİŞ

İzel Çiprut, Ocak 1971 yılında İstanbul `da doğdu. İlk, orta ve lise öğrenimini Özel Yıldız Lisesinde tamamlamıştır. 1987 yılında İstanbul Teknik Üniversitesi Elektrik Elektronik Fakültesi Elektrik Mühendisliği Bölümünde lisans eğitimine başladı. 1991 yılında mezun olarak, İTÜ Fen Bilimleri Enstitüsü, Elektrik Anabilim dalı programına katıldı. 1992 yılında, İTÜ Elektrik Müh. Elektrik Makinaları Anabilim dalında araştırma görevlisi olarak görev yaptı. 1994 yılından itibaren Arçelik A. Ş. Araştırma Bölümünde, araştırma mühendisi olarak çalışmaya başladı ve halen bu göreve devam etmektedir.



**T.C. YÜKSEKÖĞRETİM KURULU**  
**DOKÜMANTASYON MERKEZİ**