

MİKROİŞLEMCİLER

Mikroişlemciler, bilgisayar sisteminin kalbidir. Bilgisayar operasyonlarını kontrol ederek veri işleme işlevlerini yerine getirir. Kısaca işlemci veya CPU, kullanıcı ya da programcı tarafından yazılan programları meydana getiren komutları veya bilgileri yorumlamak ve yerine getirmek için gerekli olan tüm mantıksal devreleri kapsar.

İlk mikroişlemci 1971 yılında hesap makinası amacıyla üretilen [Intel firmasının 4004](#) adlı ürünüdür. Bir defada işleyebileceği verinin 4-bit olmasından dolayı 4-bitlik işlemci denilmekteydi.

Bir anda ele alabildiği bit sayısına bakılarak güçlü olup olmadığı anlaşılan işlemcilere daha sonra kısa bir süreç için sınırlı sayıda işlem yapabilen [8-bitlik 8008](#) işlemcisi eklenmiştir. 1974 yılında [Intel 8080](#) adlı işlemcisini, hemen ardından önceki işlemci ile pek farkı olmayan [Motorola 6800](#) adlı işlemcisini piyasaya sürmüşlerdir. Birbirleri arasında küçük farklılıklar olan iki işlemci daha piyasaya sürülmüştür. Bunlar, [MOS Technology firması tarafından üretilen 6502](#) ve [Zilog firması tarafından üretilen Z-80](#) işlemcileridir.

Mikroişlemciler	Üretim Yılı	Kaydedici Sayısı	Kaydedici Büyüklüğü	Veriyolu Genişliği	Adres Yolu Genişliği	Adresleme Kapasitesi
Intel 8085	1974	8 2	8 16	8	16	64K
Motorola 6800	1975	3 3	8 16	8	16	64K
Zilog Z-80	1975	17 1 4	8 7 16	8	16	64K
Mostek 6502	1976	1 16	8 16	8	16	64K

Şekil - 8-bitlik popüler mikroişlemcilerin teknik özellikleri

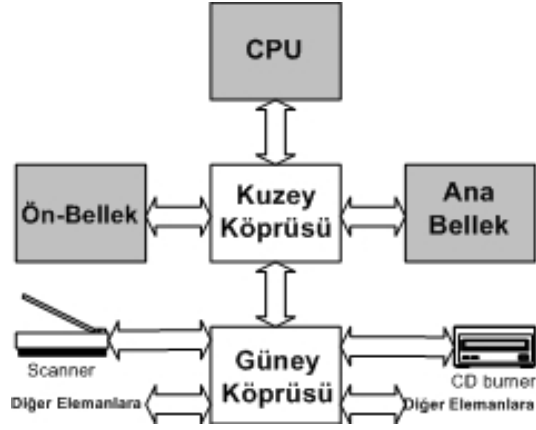
8-bitlik 8080 ve Z-80 mikroişlemcilerinde hesaplama yapmak maksadıyla bol miktarda kaydedici vardır. Bundan dolayı bu işlemcilere [kaydediciye dayalı işlemciler](#) denilmektedir.

Diğer 8 bitlik işlemciler 6800 ve 6502, anlaşılır komutlar ve daha fazla adresleme modu kullanmaları, kaydedicilerinin fazla olmamasından dolayı veri manevrasında sık sık belleği kullanmalarından dolayı [belleğe dayalı işlemciler](#) olarak anılırlar. Bu gruplar birbirlerinin bellek ve G/Ç yongalarını kullanabilmektedirler.

Mikroişlemci Esasları

Bilgisayar terimi, daha önce sözü edildiği gibi, içerisinde bir mikroişlemci, bellek, G/Ç birimi ve bunları birbirine bağlayan iletişim yollarının dahil olduğu bir sistemin tanımlanmasında kullanılmaktadır. Böylece bilgisayar, içerisinde büyük kapasiteli bellek ve disk bulunan geniş boyutlu **çok-kullanıcı** ve **çok-görevli** sistemlerden (Mainframe) otomatik çamaşır makinaları gibi ev aletlerinin denetlenmesinde kullanılan tek-yongalı işlemcilere kadar herşey olabilir.

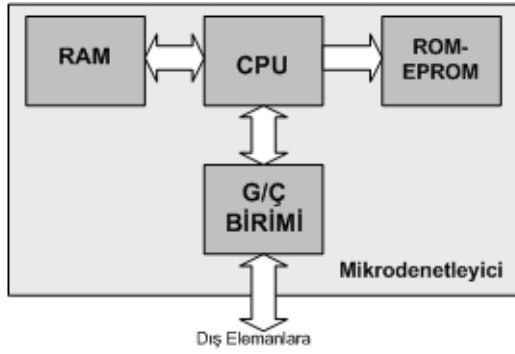
Bir bilgisayar sistemi fiziksel açıdan büyük ebatlı olduklarından sabit disk, CD-ROM vb. hariç tutulursa; işlemci, bellek ve G/Ç birimleri ayrı ayrı yonga gruplarından meydana gelmektedir. Bu elemanların kapasite ve büyüklükleri kullanıldıkları sistemlere göre artış göstermektedir. Mikroişlemci sistemler genelde **çok-yongalı sistemler** olarak anılırlar.



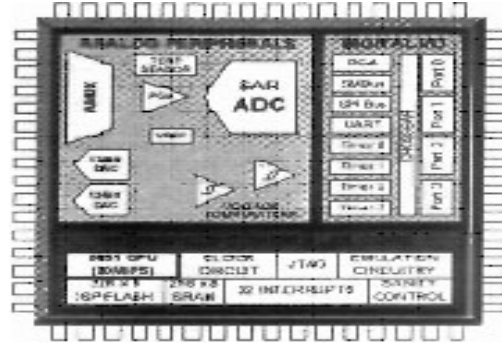
Şekil - Çok-yongalı sistemlere örnek bir bilgisayar sistemi

Ev, işyeri, fabrika gibi çevre denetiminde ve bunlara benzer aletlerin işlevlerinin denetlenmesi söz konusu olursa, bir bilgisayarın bu işler için kullanılması kapasite ve maliyet yönünden uygun olmayacaktır. Bunun yerine yeterli miktarda bellek, kaç eleman denetlenecek ise ona göre G/Ç birimi ve hızı-performansı yeterli bir işlemci bu aletlerin denetiminde yeterli olacaktır.

İşlemci, bellek, G/Ç ve bazı gerekli ek elemanlarında bulunduğu ve tek bir yonga içerisine yerleştirilmiş özel amaçlı yapıya **tek-yongalı mikroişlemciler** denilmektedir.



Şekil - Tek - yongalı mikroişlemci sistemi



Şekil - Tek-yongalı mikrodenetleyici yapısı

Mikroişlemcinin çeşitli fonksiyonel birimlerinin tek bir birimde toplanmasıyla daha fazla performans artışı sağlanmak istenmiştir. Sistemin merkezinden verilerin alınıp getirilmesi ve tekrar dışarı gönderilmesi sırasında mikroişlemcili sistemlerde olduğu gibi hız düşmesi olur. İç iletişim yolları dış iletişim yollarına nazaran daha hızlıdır.

Mikrodenetleyicilerin tarihi birbirine paralel iki aşamalı gelişimi izlenmektedir. Bunlardan birisi Intel diğeri Texas Instruments firmalarındandır. [Intel 4004](#) ile başlayan denemesini [4040](#), [4048](#) ve daha sonra popüler [8051](#) mikrodenetleyici ile sürdürürken, aslında ilk mikrodenetleyici [TMS1000](#) ile [Texas](#) tarafından üretilmiştir.

Bu tek yongaya, dahili saat, işlemci, RAM, ROM ve G/Ç dahil edilmiştir. İçerisindeki farklı birimler, kullandığı bit ve çalışma frekansları üreticiden üreticiye değişebilmektedir. [Gömülü mikroişlemci](#) de denilen bu denetleyicilerin amacı, belirli bir çevrede ve çoğunlukla özel bir görev için süreci denetlemektir. Ana elemanlarının dışında, zamanlama sayacı, iletişim hatları, analog-digital dönüştürücüler ve bazı özelleştirilmiş çıkışlar yerleştirilmiştir.

Mikrodenetleyicilerin yüksek entegrasyon yapısı, düşük enerji tüketimi ve özelleşmeyi sağlayabilmesi yerleşik sistem süreçlerinin görevleri kolaylaştırmaktadır. Watchdog zamanlayıcısı ve suspend mod özellikleri artırıldığında güvenilirlik ve kesin başarı istenen denememiş makinalarda önemli arızaların bulunmasında kullanılabilir.

Mikrodenetleyiciler, karşılıklı denetleme işlemlerinin avantaj sağladığı robotik uygulamalarında yaygın olarak kullanılmaktadır. Tek bir sistemle çok sayıdaki kontrol arasında belirlenecek özel bölümler sağlar.

Bilgi bitleri mikroişlemcinin tipine göre [sekizlik](#), [onaltılık otuzikilik](#) veya günümüzde [altmışdörtlük](#) gruplar halinde işlenir. Normal bilgisayarlarda bütün bilgiler [sekiz bit \(Bayt\)](#), [onaltı bit \(Word\)](#) veya [otuziki bit \(Doubleword\)](#) olarak işlenirler.

Kelime uzunluđu büyük olan işlemcide yapılan aritmetik işlemlerde doğruluk oranları kısa uzunluklu kelimelere nazaran çok yüksektir (4-bit %6, 8 bit %0.4 ve 16 bit %0.001).

2. Mikroişlemcinin tek bir komutu işleme hızı.

Bir mikroişlemcinin hızı saat frekansıyla doğrudan ilgilidir. Fakat saat frekansı her zaman gerçek çalışma frekansını yansıtmaz. İşlemci hızını belirleyen bir çok yol vardır. Bunlar, çalışma çevriminin uzunluğudur ki , bu ölçüm fazla kullanışlı değildir. Başlıca mikroişlemci hızları mikrosaniye olarak 1, 66, 100 MHz veya MIPS'tir(Saniyede Milyom Adet Komut İşleme).

Bir mikroişlemciyi diğerinden daha hızlı yapan unsurlar şunlardır:

- CPU'nun devre teknolojisi ve planı.
- İşlemcinin bir defada işleyebileceği kelime uzunluğu. Daha uzun kelime daha hızlı işlem demektir
- İşlemci komut kümesi. Bir işlemcide bir işlem tek bir komutla yapılırken diğerinde daha çok komutla yapılabilir.
- Genel olarak denetim düzeni.
- Kesme altyordam çeşitleri.
- Bilgisayar belleğine ve G/Ç cihazlarına erişim hızı.

3. Mikroişlemcinin doğrudan adresleyebileceği bellek büyüklüğü.

Bilgisayar sistemlerindeki ana bellek mikroişlemci tarafından adres yolu vasıtasıyla adreslenir. Adres yolu hattı ne kadar çoksa adresleme kapasitesi de ona göre büyük olur. İşlemci içerisindeki adres işaretçisi kaydedicilerin büyüklüğü, işlemcinin adres çıkışında bir kaydıran kaydedici yardımıyla artırılabilirken ve adres yolu da çoğaltılmış olur.

XT tipi bilgisayarlarda adres kaydedicisi (MAR) 16-bitlik olmasına rağmen adres bilgisi dört bit sola kaydırılarak 20-bitlik hatta verilirken 1 MB'lık bellek adreslenebilmektedir. AT tipi bilgisayarlarda 24 ve 32-bitlik adres hattı kullanılarak 4 GB'lık bellek adreslenebilmektedir.

4. Programcının üzerinde çalışabileceği kaydedici çeşitleri.

Kaydedici sayısının fazla olması manevra kolaylığı ve esneklik sağlar. Kaydediciler üç gruba ayrılır; genel amaçlı kaydediciler (A, B ve X), özel amaçlı kaydediciler (PC, SP, ve PSW) ve gizli kaydedicilerdir (İR, MAR, MDK. DAR, DR).

5. Programcının kullanabileceği değişik türdeki komutlar.

Mikroişlemci hızını etkileyen komutlar, veri manevra komutları, giriş/çıkış komutları, aritmetik komutlar, mantık kumulları ve test komutları gruplarından birisine dahildir.

6. Programcının bellek adreslerken gerek duyacağı farklı adresleme modları.

Doğrudan adresleme, dolaylı adresleme ve indeksli adresleme gibi adresleme türleri programcıya ekstra kolaylıklar sağlar. Adresleme modları üzerinde çalışılan bir verinin belleğe nasıl ve ne şekilde yerleştirileceği veya üzerinde çalışılacak bir verinin bellekten nasıl ve hangi yöntemle çağrılacağıdır.

İleri Mikroişlemci Özellikleri

Son zamanlarda mikroişlemcili sistemlerdeki hız ve performans artışı, büyük ve esaslı düzenler sayesinde sağlanmıştır. Bu düzenlerden belli başlıları, ön-bellek sistemleri, is-hattı teknolojileri, üstün dallanma tahmini yürüten sistemler ve yüksek akışkanlık sağlayan sistemlerdir.

Ön-Bellek Sistemi

Mikroişlemcilerin sistemdeki en büyük yardımcı birimlerinden birisi bellektir. Bellek, komut ve verileri üzerinde geçici veya kalıcı olarak tutan bir elektronik elemandır. Tasarıma göre komutlar ve veriler istenirse bellekte ayrı bölümlerde tutulabilmektedir. Buda neyin nerede bulunacağını bilmesini sağladığından daha fazla hız demektir. Mikroişlemcilerin ilk üretim yıllarında mikroelettronik tasarım teknolojisinden dolayı bellekler işlemcilerden daha hızlıydı. Fakat, mikroişlemci mimarisinin tasarımındaki iyileştirmeler bellekten daha hızlı yol almıştır.

Mikroişlemcilerin hızını artırmak için elden ne geldiyse yapıldığı halde bellek mimarisi yavaş ilerlemiştir. Bu sebepten, daha sonraki yıllarda mikroişlemcinin çalışma hızı bellekleri geçmiştir. Bu da ortaya hız uyumsuzluğu denilen bir problem çıkarmıştır.

Ön-bellekler mikroişlemci dolayısıyla bilgisayar performansını artıran önemli elemanlardan birisidir. Sisteme sadece belli bir yük bindiren ön-bellekleri işletmek ön-bellek altsistemleridir. Bu sistemler, verinin hangi bellekte olduğunu, bu veriye nasıl ve kaç koldan ulaşılacağını ve verinin hangi bellekte yenilenip (update) hangisinde yenilenmediğinin bilgisini tutmaktadır.

İş-Hattı ve Süperölçekli İşlem

Mikroişlemci tarafından işlenecek komutlar sırasıyla ana belleğin kod bölümünden alınarak getirilir. Bir sonraki adımda getirilen bu kodun ne demek istediği kod-çözücü bölümünde deşifre edilir. Daha sonra bu deşifre edilen bilgiye göre işlem gerçekleştirilir. Basit olarak bir işlemcinin çalışması kaç adımda gerçekleşir; *komutu algetir, kodunu çöz ve çalıştır*.

Bilgisayar sistemlerinde gelişmeler işlemci gelişmesine de yansıdığından, üç adımda ve tek tek işlenen komutlar günümüzde beş ve daha fazla adımda bir defada yapabilmektedir. Orta hızlı bir işlemcide komut aşağıdaki aşamalardan geçer:

- İşlenecek komutun bellekten alınarak komut kaydedicisine getirilmesi (Algetir safhası).
- Getirilen komutun, mikrokod veya donanımsal yönteme göre kodun çözülmesi (Kod-çözme safhası).
- Üzerinde işlem yapılacak birimin adresinin belirlenmesi (adres üretim safhası).
- Komutun kodunun çözülmesiyle ne yapılmak istendiği anlaşılacak gerçekleştirilmesi (icra/işlem safhası).
- Elde edilen sonucun belleğe yazılması (geriye yazım safhası).

	Algetir Birimi	Kod-özme Birimi	Adr. Üretim Birimi	İcra Birimi	Geri Yazma Birimi
0.saat çevrimi	Komut K	Komut K-1	Komut K-2	Komut K-3	Komut K-4
1.saat çevrimi	Komut K+1	Komut K	Komut K-1	Komut K-2	Komut K-3
2.saat çevrimi	Komut K+2	Komut K+1	Komut K	Komut K-1	Komut K-2
3.saat çevrimi	Komut K+3	Komut K+2	Komut K+1	Komut K	Komut K-1
4.saat çevrimi	Komut K+4	Komut K+3	Komut K+2	Komut K+1	Komut K

Bir komutun yukarıda sıralanan beş adımda gerçekleşmesi sırasında diğer birimlerin işini bitirdikten sonra yeni bir işe başlaması sistem performansını önemli ölçüde artıracaktır.

Algetir biriminin bir komut getirildikten sonra kod-özme bölümüne göndermesi ve hemen ardından başka bir komutu işlemek üzere getirmesi ve diğer birimlerinde işlerini bitirdikten sonra sıradaki işleme dalmasına **iş-hattı sistemi** denir.

Şimdi beş kademeli bir mikroişlemci mimarisine diğer bir veya iki ya da üç beş kademe daha eklenirse ne olur? Bu sistem iki yollu veya üç yolu beş kademeli bir sistem olur ki buna **Süperölçekli mimariye sahip mikroişlemci** denir. Bu üç yollu beş kademeli işlemcide aynı anda tüm birimlerde işlem yapıldığında çok kısa bir sürede büyük işlemler yapılabileceği ortaya çıkmaktadır.

Tek bir montaj hattından 30 dk sonra bir otomobil çıkar. Fakat bir yerine iki veya üç montaj hattı yerleştirirse her 30 dakikada iki yada üç otomobil üretilmiş olacaktır. Aynı zamanda montaj hattının hızı uygun şartlarda artırılırsa bu süre daha da kısacaktır. Mikroişlemcilerde bir kaç is-hattıyla oluşturulan süperölçekli mimari, işlemci çalışma frekansının artırılmasıyla yüksek performanslara erişir. Bazı durumlarda bazı kademeler aşağıda sıralanan tehlikelerden dolayı işsiz kalabilir.

- **Yapısal tehlikeler:** aynı anda iki farklı yoldan aynı kaynakların kullanılmasına teşebbüs edilmesi.
- **Veri tehlikeleri:** daha veri hazır olmadan kullanılmaya kalkışılması. Verinin bir parçası ikinci kademede diğer parçası üçüncü kademede olduğu gibi ikinci kademede verinin işi bitirilip üçüncü kademeye yollanmadan veri hazır değildir.

- **Kontrol tehlikeleri:** şartlar ve durumlar değerlendirilmeden önce karar vermeye teşebbüs edilmesi. Bir komutun islenmesi sonucunda dallanmanın olmadığına bakılmadan sıradaki komutun işlenmesi gibi.

Mikroişlemci denetim sistemi öyle bir yapılandırılmalıdır ki, iş-hattı kontrol mekanizması tehlikeleri sezebilsin. Gecikme hareketleri bu tehlikelerin çözülmesi yönelik olabilir. Günümüz işlemcilerinden Pentium'da 5, P-6 ailesinde 10 ve P4'de 20 kademeli iş-hattı vardır.

Dallanma Tahmini

Mikroişlemci tarafından işlenecek komutlar programcı tarafından yazıldığı biçimde bellekte sıralı olarak dururlar. Fakat işlenecekleri zaman derleyicinin de yardımıyla eş olup olmadıklarına bakılarak sisteme yollanırlar. Günümüz mikroişlemcilerinde yukarıda belirtildiği gibi, üç veya dört yollu iş-hattı mevcuttur.

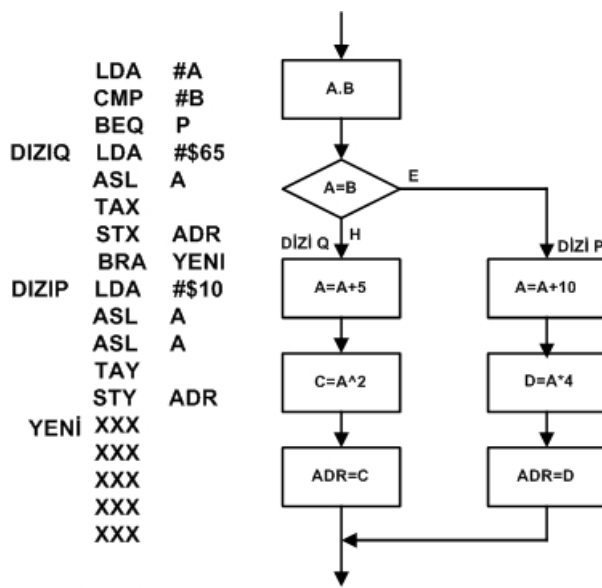
Komutlar içerisinde en güçlüleri **kontrol komutlarıdır**. Bunlar **şartlı ve şartsız** olmak üzere ikiye ayrılırlar. **Şartsız dallanmalara** JMP (jump-dallan), CALL ve INT (interrupt-kesme) gibi komutlar girerken, **şartlı dallanmalara** JNE/RNE, JE/BEQ, JA/JNEB gibi komutlar girer.

Şartlı dallanma komutlarından birisiyle karşılaşıldığında, bir önceki işlemin sonucuna bakılarak işlem yönü belirlenir. Bu durumda çok yollu iş-hatlarına işlenmek üzere alınan komutlardan birisi dallanma komutu ise ortaya büyük bir sorun çıkabilecektir. Sistemin dallanacağı ve işleteceği yeni komutlar iş-hatlarında olmayabilecek ve tüm iş-hatları boşaltılarak bu yeni komut dizisi çalıştırılacaktır. Böyle bir durumda da büyük zaman (performans) kaybı olacaktır.

Intel firmasına göre, **kontrol komutlarının** komut setindeki oranı yaklaşık %20'dir. Buna göre komut işlenirken mutlaka sık sık karşılaşılabilecek olan dallanma komutlarını sisteme zarar vermeden önceden belirleyip tedbir almaktır. Yani, dallanmanın gerçekleşip gerçekleşmeyeceğini. şartı belirleyen mantıksal ve aritmetik işlemin sonucunu hesaplamadan önce tahmin etmektir.

Bu yoldan çıkılarak işlemci tasarımcıları, komutları illâki programdaki sıraya bakarak işlemiyorlar. Bunun yerine değişik satırlardaki komutların birbirleriyle ilişkisini çözümleyip gerektiğinde program sırasını değiştirerek ve bazı komutları öne alarak aynı anda paralel olarak birden fazla komutun çalıştırılabilmektedir. Her ne kadar sıralı komut çalıştırılırsa, dallanma komutu geldiğinde sapılacak adresteki yeni komutların getirilmesi ve işlenmesi zaman alabilecektir.

Mikroişlemci içerisindeki iş-hatları yanda verilen program kodlarıyla doldurulduğu varsayılırsa, **Dizi Q** ve **Dizi P** birbirine alternatiftir. ikisi aynı anda çalışamayacak ve **CMP #B'nin** sonucuna göre ikisinden birisi ele alınacaktır.



Şekil - Dallanmaya sahip basit bir program akış diyagramı

Buna göre, iş-hatlarının **Dizi P** ile dolu olduğu farzedilsin.

Eğer **A=B** şartı doğru çıktığı taktirde bir sorun ortaya çıkmayacak işlem tüm hızıyla sürecektir.

Eğer **A=B** şartı doğru çıkmazsa ne olacak? Bu durumda **Dizi P**'deki komutlarla dolu olan iş-hatları boşaltılarak **Dizi Q**'daki komut dizileriyle doldurulacaktır.

Bundan dolayı sistemde performans azalması olacaktır.

Eğer şart doğru çıkarda iş-hattındaki komut dizisinin işlenmesine devam edilirse bu durumda o kadar da zaman/performans kazanılmış olur.

Programcı tarafından yazılan komutların içerisinde her 7 veya 9 satırda bir dalma komutlarının olduğu ve tahmin edilen komut dizisinin tutma oranının da düşük olduğu varsayılırsa performansdaki azalma gözle görülebilir olacaktır. Bunu önlemenin yolu en basit olarak, [ya fazladan iş-hattı yolları](#) (var olanın aynısı) eklenmesidir ki bu ikinci bir (paralel) işlemci demektir, ya da [dalınacak olan dizinin iyi tahmin edilebilmesi için mükemmel mekanizmaların geliştirilmesidir](#).

Bunlardan birincisi, iş-hattının bir kopyasının oluşturulması, oldukça maliyetli olacak ve işlememin ebadının büyümesini sağlayacaktır. Bu tip bir yaklaşım mümkün olmayacaktır. Tasarımcılar ikinci yaklaşımı uygun görerek, her iki diziden birisini sanki çalıştırılacakmış gibi iş-hatlarına almak, eğer gerçekleşirse kolayca çalıştırmak, eğer gerçekleşmezse iş-hattını boşaltarak yeni diziyi iş-hattına alarak çalıştırmaktır. Bu tahminsel yaklaşımda, başarı random olarak %20 veya %80 arasında değişebilir.

[Statik ve dinamik algoritmalar](#) olarak ortaya atılan bu sistemde, eğer programın akışı sırasında tahminlerini değiştirmeyen algoritmalardır. Meselâ, "[hiç bir zaman dallanma](#)" statik bir algoritma olabilir. Rastladığı her dallanma komutunun şartını her zaman dallanmama yönünde tahmin eder ve ona göre program akışını yönlendirir.

Matematiksel olarak doğru çıkma şansı %50 olsa da, bu uygulamaya göre daha yüksek değerler alabilir. Bundaki mantık, program yazarın kişilerin genelde alışkanlık veya zihinsel eğilim sonucunda, genellikle koddaki istisnai durumlarla ilgilenen kısımlara erişirken dallanma şartını kullandıkları görülmüştür.

Dinamik algoritmalar adından da anlaşılacağı gibi, programın akışına veya yaptıkları hatalara bakarak geleceği tahmin etme yeteneğine sahiptirler. Karmaşık bir yapıya sahip olan bu algoritmalar, programın başından itibaren dallanma komutlarının bir bakıma tarihçesini kaydeder ve programın sonraki bölümlerinde rastlayacağı ve aynı veya benzer dallanma komutlarının tahmin edilmesinde bu bilgilerden faydalanır.

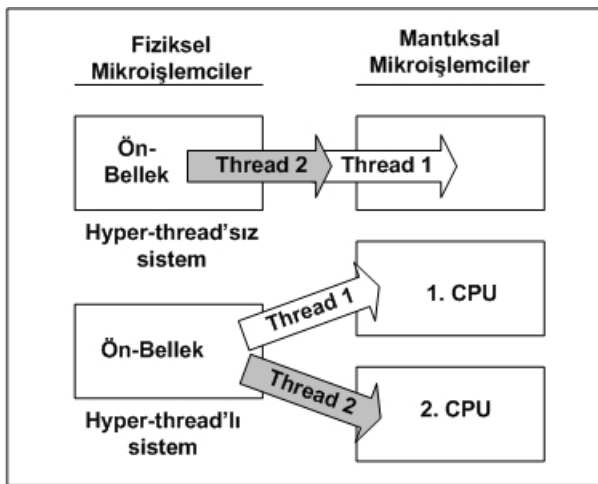
Tarihçe ne kadar geriye yönelik ve ayrıntılı ise tahminlerdeki doğruluk o kadar artar. Bu da, işlemci üzerinde bu işe yönelik bellek ve birim ([BTB-Dallanma adreslerinin tutulduğu yer](#)) artışı demektir. Programdaki dallanmaların hangi yönde olduğu ve hangi şartlarda dalındığı hakkında bilgi bu birimlerde tutulur. Program ilerleyişinde dallanmalarda tahmin %90 veya %98'lere çıkmaktadır.

Çoklu Program İşleme

X86 tabanlı gelişmiş işlemcilerin performansının artırılmasındaki diğer bir adımda, [Hyper-Threading](#) denilen bir teknolojidir. Bu teknoloji ile tek bir işlemci, iki adet işlemci gibi çalıştırılmaktadır. Çift işlemcili bir sistemin aksine hyper-threading kullanan tek işlemcili bir sistemde ön-bellek, veri yolu ve firmware, tanımlanan iki adet mantıksal işlemci tarafından ortaklaşa kullanılmaktadır.

Bu özellik, çok işlemcili sistemlerde görülen [Symmetric Multiprocessing](#) teknolojisine benzemektedir. Fakat arada birkaç fark vardır, simetrik çok işlemcili sistemlerde her işlememin kendisine ait veri yolu, ön-belleği varken, Hyper-threading teknolojisinde mantıksal işlemciler aynı veri yolunu ve ön-belleği paylaşmak zorundadır. İşletim sistemi ile ilgili yazılım arasındaki ilişki sonucunda gerekli işlem gücü, birden çok parçaya ayrılmaktadır.

Mulli-thread destekli yazılımlar birden çok çalışma yüzeyine sahiptirler. Yani bir yüzeye birden fazla thread'i alırlar ve toplu halde işlemci içerisindeki kontrol işlevlerine yollarlar. Fakat burada gerçekleşen olayların birbirinden bağımsız olması çok önemlidir.



Hali hazırda var olan Multi-threading işlevinin geliştirilmiş bir modeli olarak Hyper-threading teknolojisi yeni işlemcilere ve ilgili donanımlara konulmuştur.

Bu kavramdaki mantık, thread değişiminin sadece işletim sistemi tarafından yapılmaması ve dolayısıyla daha yüksek bir kararlılık ve performans elde edilmesidir.

Burada işlemler ayrı ayrı paralel veri kanallarından aynı anda işlenmektedir. Hyper-threading sisteminde, işletim sistemi ve yazılımların tek bir fiziksel işlemciyi iki mantıksal işlemci olarak algılaması sağlanmaktadır. Dolayısıyla sürekli olarak en az iki adet thread görünmekte ve işlemlere anlatıldığı şekilde devam edilmektedir.

Hyper-threading destekli işlemcilerin çalışmasında ilk olarak, işlemci iki adet mantıksal işlemci olarak tanımlanarak hazır hale getirilmektedir.

Diğer İleri İşlemci Özellikleri

Yukarıda bahsedilen yüksek performanslı ileri işlemci özelliklerine ilave olarak pek çok özellikler katılabilir. Bu özellikler artık tüm ileri işlemcilerde yaygın olarak kullanılmaktadır. Belli başlıları aşağıda sıralanmıştır:

- **Multimedya işlemi:** İşlemcilerin; resim, ses, grafik ve iletişim gibi işlemlerde performansının düştüğü bir gerçektir. Tamamına multimedya özellikleri denilen ve MMX ve 3D NOW gibi adlarla anılan bu özellikler X86 tabanlı işlemcilerin performansında büyük artış sağlamıştır.
- **Komutların işlenmesi:** MMX komutları kullanılabildiği dek gelişmiş işlemciler, aynı anda tek bir komutla tek bir veri üzerinde oynayabilme özelliğine (SISD- Single Instruction Single Data-Tek komut Tek Veri) sahipti.

MMX komut kümesiyle birlikte işlemciler zorunlu olarak tek komutla bir çok veri üzerinde çalışılabilen (SIMD-Single Instruction Multiple Data-Tek komutla Çok Veri) bir yeteneğe sahip olmuşlardır. Daha sonraları SSE ve SSE-II denilen bu sistemde SIMD yeteneğine daha da akışkanlık sağlanarak, multimedya işlemleri daha hızlı bir şekilde yürütülmesi hedeflenmiştir.

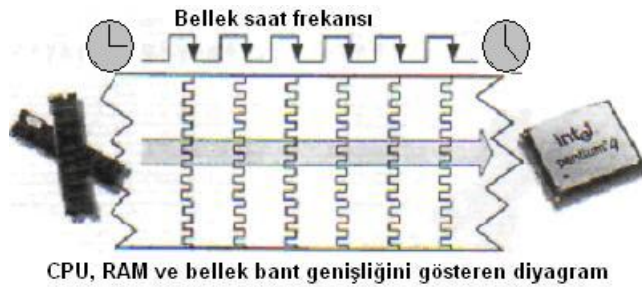
- **Komutların mikro-operasyonlara bölünerek sırasız çalıştırılması:** Bazı RISC tipi mikroişlemcilerde daha önce bazılarında daha sonra devreye sokulan gelişmiş komut işleme mantığı, X86 tipi işlemcilere P6 ailesiyle birlikte girmiştir.

Eski CISC tipi işlemcilerde komutlar bellekten getirilip kodu çözülerek işleniyordu, işlemci sadece X86 komutlarının kodunu çözüp icra birimine göndermek yerine, onları mikro-operasyon (uop) dizilerine dönüştürerek süperölçekli iş-hattında çalıştırılmak üzere bekletir.

- **Çalışma izleme belleği:** Çok kademeli iş-hattında dallanmalardan dolayı meydana gelebilecek gecikmeleri ortadan kaldırmak için çeşitli adlar altında çalışma izleme belleği denilen birim

eklenmiştir. Ön-bellekte bekleyen komutları alıp kodlarını çözerek anlaşılabilir formlara dönüştürdükten sonra onları icra birimine hazır hale getiren bir havuzdur.

- **Yüksek bantgenişliği ve gecikme:** Bantgenişliği bir veri yolundan bit olarak belli bir zamanda ne kadar veri iletilebileceğini gösterir. Yani kabaca, bu yol veya hat ne kadar veri taşıyabilir. Genellikle bu tabir mikroişlemcili sistemlerde CPU ile RAM arasındadır. İşlemcinin çalışma frekansının (bu yükselen veya düşen kenarında olabilir) her bir tıklamasında bu veri yolundan ne kadar bitlik veri gönderilebileceğidir.



Şekilde, ana belleğin CPU'ya 8-baytlık veri bloğunu 6 defa göndermesini göstermektedir. Burada 64-bitlik bellek veri yolu gözönüne alınmıştır. Eğer bellek veri yolu genişliğinin yarı yarıya azaldığı düşünülürse, her bir saat darbesinde veri taşıma oranı yarıya düşer (yani 4-bayta). Eğer iki misline çıkarılırsa, 128-bit, her saat darbesinde 16-bayt olacaktır.

Her 8-baytlık blok gönderimi bellek frekansının düşen kenarında gerçekleşmektedir. **Her bir 8-baytlık blok word (sözcük)** olarak anılır. Böylece şekildeki sistem bellekten işlemciye ardarda 6 sözcük göndermektedir. Bu açıklamalar ışığında, bantgenişliğini artırmak için iki yol vardır.

Birincisi, veri yolu saat frekansını artırmaktır. Her bir düşen kenarla yükselen kenar arasında mutlaka boş kullanılmayan alan vardır. Bu alanı mümkün olduğunca sık tutmaktır. Bu durumda, düşen belli bir zaman dahilinde düşen kenarların sayısı artacağından veri taşıma oranı da yükselecektir.

İkincisi, yukarıda bahsedildiği gibi bellek ile CPU arasında döşenmiş veri yolunu genişletmektir. Bu ikisinin birlikte yapılması elbette ek maliyet getirecektir. Bantgenişliği ile birlikte anılan diğer bir terim gecikmedir (latency). Bu işlem, bir verinin bir birimden diğer birime geçişine kadar ki geçen

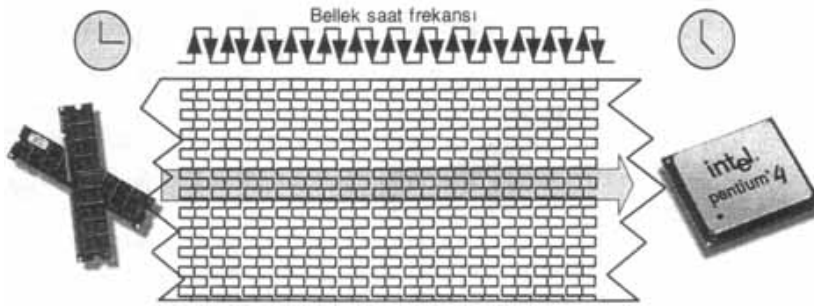
süreyle tanımlanabilir. Bunlar işlemci ile belleğin yapısal olarak ne kadar birbirine yakın olduğu ve bunların saat frekanslarıyla ilgilidir.

Bir örnek verilecek olursa, eğer bir veri yolu 100 MHz'de çalışıyorsa (saniyede 100 milyon saat çevrimi), ve her bir çevrimde 8-baytlık veri taşıyabiliyorsa bu sistemin bantgenişliği 800 MB/s olacaktır.

Bantgenişliği = veri yolu saat frekansı * veri miktarı

Bantgenişliği = 100 * 8 = 800 MegaBayt/saniye

Eğer daha hızlı bir veri yolu sağlamamışsa, meselâ 200 MHz, bantgenişliği 200*8=1.6 GB/s olacaktır. Veri yolunun iki misline genişlediği düşünülürse, 200*16=3.2 GB/s olacaktır. Bu değerleri yükseltmenin yolu daha öncede belirtildiği gibi, saat frekansını yükseltmek ve veri yolunu genişletmektir.



Şekil-Saat frekansı ve veri yolu artırılmış bantgenişliğini gösteren diyagram

Bantgenişliğinin yükseltmenin diğer bir yolu Şekilde görüldüğü gibi, son zamanlarda gelişmiş işlemcilerde uygulanmaktadır. Veri yolundan tek bir frekansla değil birden fazla frekansla ve bir kenarda değil hem yükselen hem de düşen kenarlarda veri aktarmaktır (veri akışkanlığını artırmaktır).

Mikroişlemci Performansı

Performans, verilen bir görevin gerçekleştirilmesi için harcanan zamanla doğrudan ilgili bir kavramdır. Bir çok mikroişlemci sabit oranda çalışan bir saat (sabit frekanslı saat sinyali) kullanılarak tasarlanır. Böylece, tasarımcı onun frekansının veya periyodunun oluşturduğu saat sinyalini referans alır. Meselâ 100 ns'lik zaman periyodunda çalışan bir işlemcinin çalışma frekansı 10 MHz olacaktır. Frekans, $f=1/t$ formülünden yola çıkılarak bulunur.

Performansa etki eden ana unsurlar şunlardır;

- **Verilen görevin yapılma zamanı:** Bunları; işlem zamanı, cevap verme süresi ve gecikmeler oluşturur.
- **Belli bir zamanda belirli bir işin yapılması:** Burada zaman; hafta, gün, saat, dakika veya saniye olabilir. Süreç ve bantgenişliği ile ifade edilir.

Bir işlemcinin belli bir programı çalıştırması için harcadığı zaman aşağıdaki formül ile bulunur:

$$\text{CPUzamanı} = \text{Program için gerekli çevrim sayısı} * \text{Saat çevrimi (periyot)}$$

Saat çevrimine işlenmiş komut sayısında eklenirse, komut başına düşen çevrim sayısı (CPI) kolayca bulunabilir.

$$\text{CPI} = \frac{\text{Programın tamamı için harcanan saat çevrimi sayısı}}{\text{Programın işlenmiş komut sayısı (I)}}$$

Veya $\text{Harcanan saat çevrimi sayısı} = \text{CPI} * I$

Harcanan saat çevrim sayısı 1. formülde yerine konulacak olursa : $\text{CPUzamanı} = I * \text{CPI} * I$ olacaktır.

Eğer bu formül ölçüm birimlerine sokularak yeniden düzenlenirse onaya basit ve gerçek sadelikte bir işlemci zamanı çıkacaktır.

$$\text{CPUzamanı} = \frac{\text{Komutlar}}{\text{Program}} * \frac{\text{Çevrim sayısı}}{\text{Komutlar}} * \frac{\text{Saniye}}{\text{Çevrim Sayısı}}$$

Formül sadeleştirildiğinde $\text{CPUzamanı} = \text{Saniye} / \text{Program}$ çıkmaktadır.

Komut oranına göre performansın (P) bulunması için CPUzamanı çevrilirse :

$$P = \frac{1}{\text{CPUzamanı}} = \frac{1}{I * CPI * I} \text{ veya } P = \frac{F}{I * CPI}$$

Burada F, T' nin tersini gösteren saat frekansdır.

P = MIPS (million instruction Per Second-milyon olarak saniyedeki komut sayısı),

I = Programı çalıştırmak için gerekli olan komut sayısı,

CPI = Her komut için gerekli olan ortalama çevrim sayısı,

F = Saat frekansı (oranı)

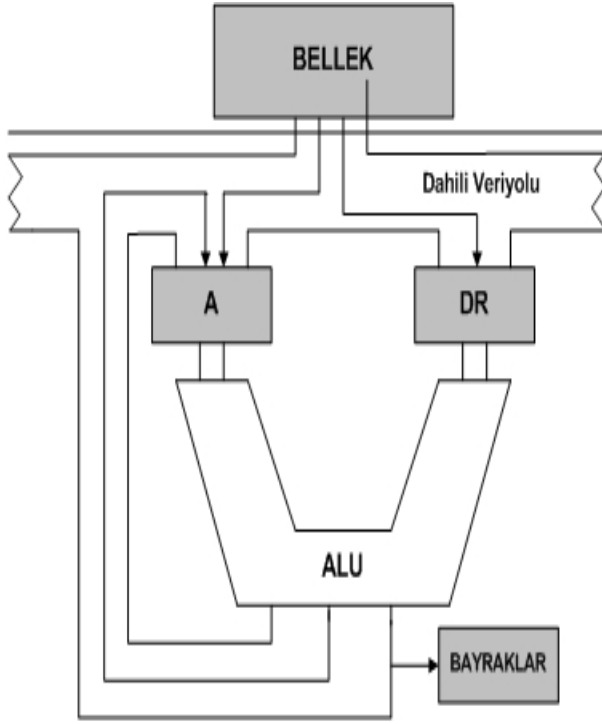
Basitten Karmaşığa Mikroişlemciler

1.8-Bitlik Mikroişlemciler

1.2. Aritmetik ve Mantık Birimi

ALU mikroişlemcide aritmetik ve mantık işlemlerinin yapıldığı en önemli bölümlerden birisidir. Bu birimdeki işlemler, akümülatörle bellekten alınan veri arasında veya akümülatörle kaydediciler arasında iki veya tek elemana dayalı olarak akümülatörde, kaydedicide veya bellekten bir kelime üzerinde olabilir.

Aritmetik işlemler denilince akla başta toplama, çıkarma, bölme ve çarpma gelir. **İşlemcide çarpma**, akümülatördeki verinin sola bir bit kaydırılarak iki ile çarpılması demek ve **bölme**, verinin bir bit sağa kaydırılarak akümülatördeki verinin ikiye bölünmesi demektir. Komutlarla birlikte bu işlemleri, mantık kapıları, bu kapıların oluşturduğu toplayıcılar, çıkarıcılar ve kaydıran kaydediciler gerçekleştirirler. Bloklaştırılmış bu devreler bir dahili veri yolu vasıtasıyla birbirlerine, bir harici veri yolu ve tamponlar vasıtasıyla kaydedicilere ve zamanlama-kontrol birimine bağlanmışlardır.



Şekil - Aritmetik ve mantık birimi

Aritmetik ve mantık biriminin çalışması bir toplama işlemi ile anlatılırsa;

Birinci veri **LDAA KLV** komutuyla birlikte klavyeden girilen herhangi bir veri doğrudan akümülatöre yüklenir. Daha sonra **ADCA \$0100** komutuyla birlikte **[0100]** bellek adresindeki veri ALU'daki geçici veri kaydedicisine DR'ye alınır ve bir sonraki çevrimle birlikte toplayıcının bir girişinden klavye verisi diğer girişinden bellekten alınıp getirilen veri girerek toplanırlar ve sonuç yine dahili veri yolundan doğrudan akümülatöre giderken **N**, **Z** ve **C** bayrakları etkilenebilir.

ALU'da gerçekleştirilen tek işlenenli işlemlere akümülatörü silme, tamlama veya tersleme, sağa - sola kaydırma ve yönlendirme, indis kaydedicileri artırma ve azaltma işlemleri örnek verilirken iki işlenenli işlemler toplama, çıkartma, karşılaştırma, mantıksal AND, OR, EXOR ve BCD aritmetik işlemleri örnek verilebilir.

Bu birimi meydana getiren devrelerin yapısı mantık komutlarından daha iyi anlaşılır. Buna göre devrede AND, OR, EXOR, NOT gibi kapılar bulunmaktadır.

Yukarıda iki işlenenli çalışma anlatılırken çok karmaşık aritmetik işlemlerden söz edilmemiştir. Bu işlemler doğrudan bir komutla çarpma, bölme, karekök olabilir. Buna benzer işlemleri doğrudan gerçekleştirmek için ayrı bir alt-yordam gruplarına veya ek elektronik devrelere ihtiyaç vardır. Eğer ek devre konulmamışsa, mevcut devrelerle bu işlemleri gerçekleştirmek için birbiri ardına aynı komutu defalarca işlemek gereklidir, bu da zaman kaybı demektir.

Gelişmiş mikroişlemcilerde bu devreler yerleşik vaziyettedir. Ayrıca bahsedilen aritmetik işlemlerde yuvarlanmış kesirli sayılar yerine tam sayılar üzerinde durulmuştur. Yuvarlanmış sayılarla 8-bitlik işlemcilerde iş yapmak oldukça zordur. Günümüz işlemcili sistemlerde mikroişlemci yanına birlikte çalışabilen bir ortak işlemci konulmakta veya mikroişlemci içerisine yerleştirilmektedir.

1.3. Zamanlama ve Denetim Birimi

Merkezi İşlem Biriminin üçüncü bölümünü meydana getiren bu kısım, sistemin tüm işleyişinden ve işlemin zamanında yapılmasından kurumludur. Zamanlama ve denetim birimi, bellekte program bölümünde bulunan komut kodunun alınıp getirilmesi, kodunun çözülmesi, ALU tarafından işlenmesi ve sonucun alınıp geri belleğe konulması için gerekli olan denetim sinyallerini üretir. Bilgisayar sisteminde bulunan dahili ve harici bütün durumlar bu denetim sinyalleri ile denetlenir.

6502 mikroişlemcisinin bu bölümü üç değişik işlevi yerine getirir:

1. Zamanlama denetimi : 6502 harici bir saat sinyali üreten birimden giriş alan iç-saat devresine sahiptir. Bu sinyal alınarak zamanlama sinyallerine çevrilir ve komut kod çözücüsüne gönderilir.

2. Komut kod çözücüsü : Bu devre komut kaydedicisinde (IR) tutulan komutları yorumlar ve ALU'ya kaydedicilerle çalışması için uygun sinyaller gönderir. (kastedilen zamanlama ve kesme sinyalleri).

3. Kesme mantık birimi : Gerekli durumlarda kesme sinyallerini alarak işlemciyi uyarırlar.

1.4. İletişim Yolları

Her ne kadar mikroişlemci mimarisine girmese de işlemciyle ayrılmaz bir parça oluşturan iletişim yolları gerçekleştirdikleri göreve göre kendi aralarında üçe ayrılırlar.

Adres yolu; komut veya verinin bellekte bulunduğu adresten alınıp getirilmesi veya tersi işlemlerde adres bilgisinin konulduğu yoldur. 16-bitlik hatta sahip adres yolu tek yönlü yapıya sahiptir. Çünkü, sadece işaretçi olarak vazife görür.

Adres sadece tarif edilir, gelmez. Adres yolunu meydana getiren hatlar aynı zamanda adresleme kapasitesini de gösterir. Maksimum bellek kapasitesi 2^{üssü n}'dir. Burada n, adres hattı sayısıdır. Eğer bir sistemde adres hattı 16-bit ise o sistemin bellek büyüklüğü 2 üzeri 16=65536, kısaca 64 KB olacaktır.

Veri yolu ; işlemciden belleğe veya G/Ç birimine veri yollamada ya da tersi işlemlerde kullanılır. Eğer kaydediciler 8-bitlikse veri yolları da 8-bitliktir. Diğer durumlarda veri iki parça halinde iki kerede getirilecek ve dolayısıyla zaman kaybı olacak veya kapasite uyuşmazlığı baş gösterecektir. Veri yolları bilginin çift yönlü taşınmasında (yükle ve sakla işlemleri) kullanılmaktadır.

Kontrol Yolu ; Sisteme bağlı birimlerin denetlenmesini sağlayan özel sinyallerin oluşturduğu bir yapıya sahiptir. **R/W** (Read/Write), **CS** (Chip Select), **CE** (Chip Enable), **Halt** (işlemci durdurma) gibi sinyaller birer kontrol sinyalidir. Kontrol yolunu meydana getiren sinyaller üç gruba ayrılır:

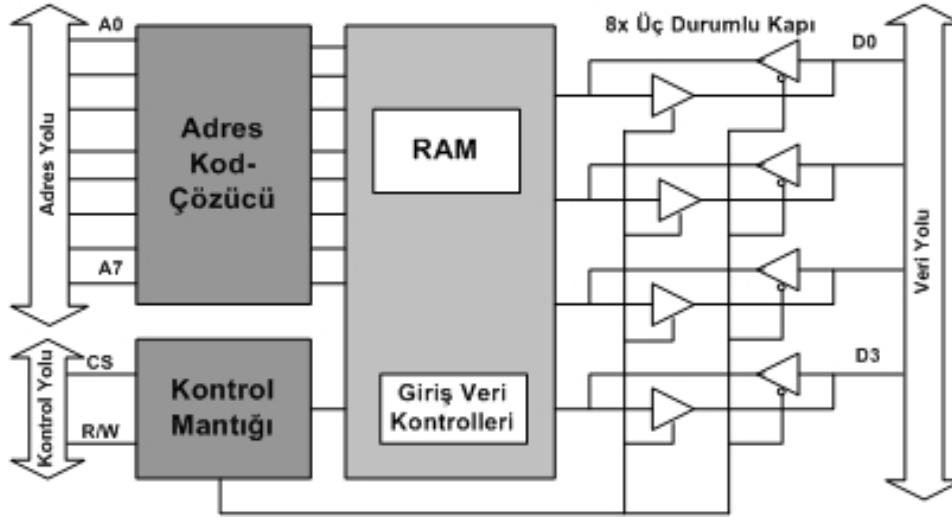
- **Kesme Sinyalleri** : Dış dünyadan (çevre elemanlarından) veya işlemci dışarisından gelebilecek kesme sinyallerinin kullandığı hatlardır. Bunlar; IRQ, NMI veya RESet gibi sinyallerdir.
- **Yön Belirleme Sinyalleri** : Verinin hangi yöne gideceğini ve hangi yonganın seçileceğini belirleyen sinyallerdir (Bellekten okuma veya yazma gibi).
- **Zamanlama Sinyalleri** : Bu hatları kullanan sinyaller hangi zamanda ne yapılacağını tayin ederler. Bunlar saat darbeleri ve işlemci içerisinde veya dışarisında bir elemanı tetiklemek üzere gönderilen sinyallerdir. Meselâ, A kaydedicisine bir veri yükleneceği zaman "in" ucuna kontrol birimi tarafından bir sinyal gönderilmelidir. Ya da bellekten okunan bir verinin veri yolu üzerinden sisteme girdiğinde hangi birime gideceği bu yoldan gönderilen sinyalle belirlenir. Aksi durumda bu veri tüm birimlere yüklenecektir.

1.5. Veri Yolu Bağdaştırma Devreleri

Veriler iletim yolunda gezerken hangi birimle ilgiliyse o birime doğrudan giremez veya çıkamaz. Mutlaka ilgili birimlerin tampon veya sürücü devrelerinde sorgulandıktan sonra işlem görürler. Bu tip uygunlaştırıcı veya bağdaştırıcı devreler üç durumlu elektronik kapılar dizisinden oluşturulur.

RAM tipi belleklerden veya G/Ç biriminden hem okuma hem de yazma yapılabildiğinden iki yönlü veri akışını sağlayan düzenler kullanılır. ROM tipi bellekten sadece okuma yapıldığından tek yönlü üç-durumlu devreler kullanılır. **Üç-durumlu devreler** adından da anlaşılacağı gibi bir uca sahiptir. Bu uçlardan ikisi **giriş ve çıkış** olarak diğeri de **yetkilendirme**(kontrol) ucu olarak düzenlenmiştir.

Üç-durumlu kapılardan bazılarının yetkilendirme uçlarında **invertör** kullanılırken bazılarının çıkış uçlarında kullanılır. Veri yolundaki her bir hatta bir adet üç-durumlu bağlanır.



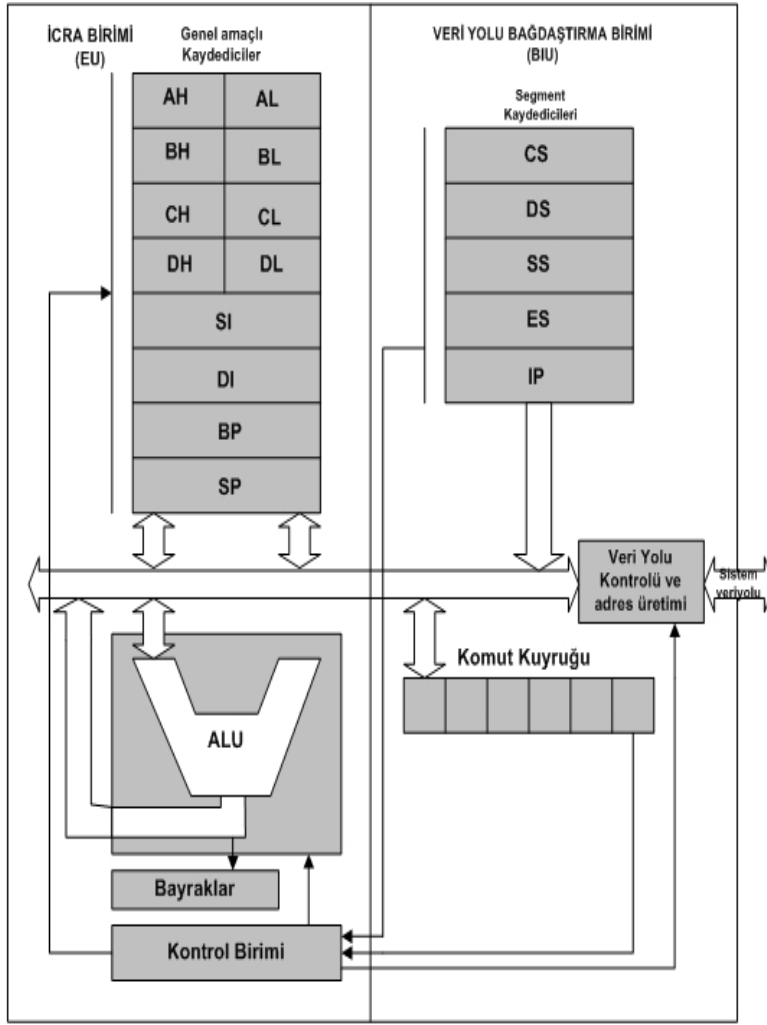
Şekil - Bellek bağdaştırma ve veri yolu devrelerinin uyarlanması

Şekilde A0-A8 ile gösterilen hatlar adres yoluna bağlıyken, D0-D3 hatları veri yoluna bağlıdır. Mikroişlemci tarafından ana bellekten bir veri okunması gerekiyorsa, zamanlama ve kontrol birimi tarafından kontrol yolundaki CS ve R/W hattına gerekli yongayı seç ve oku (mantıksal 1) sinyali gönderilirken, kontrol mantığı devresi üç-durumlu mandallara verinin dışarı çıkacağını bildirir, yani sol yandaki mandal grubuna yetki verilir.

Böylece okunacak verinin adresi belli, kontrol sinyalleri de gönderildi, son olarak mandallar açılarak veri otomatik olarak veri yoluna çıkar. Veri yoluna çıkan verinin aynı yoldan tekrar belleğe dönmesi mümkün değildir. Bundan dolayı veri yolundaki bilgiler mikroişlemci tarafından alınır veya eğer sistemde [Doğrudan Bellek Erişim \(DMA\) devresi](#) bulunuyorsa çıkışa gönderilebilir.

2 . 16-Bitlik Mikroişlemciler

16-bitlik mikroişlemciler basit olarak 8-bitlik işlemcilerde olduğu gibi, [Kaydediciler](#), [ALU](#) ve Zamanlama-Kontrol Birimine sahiptir. 16-bitlik X86 tabanlı işlemciler [Veri Yolu Bağdaştırma Birimi \(BIU\)](#) ve [İcra Birimi \(EU\)](#) olmak üzere iki ana bölümde toplanabilir.



Şekil - Basit 16 - bitlik mikroişlemci mimarisi

BIU birimi, EU birimini veriyle beslemekten sorumluyken, **icra birimi** komut kodlarını çalıştırılmasından sorumludur. BIU bölümüne segment kaydedicileriyle birlikte IP ve komut kuyrukları ve veri alıp getirme birimleri dahilken, EU bölümüne genel amaçlı kaydediciler, kontrol birimi, aritmetik ve mantıksal komutların işlendiği birim dahildir.

Mantıksal iki ana bölümdeki bütün bu birimlerin, yonga üzerinde birbirlerine fiziksel olarak bağlı bulundukları unutulmamalıdır.

2.1. Veri Yolu Bağdaştırma Birimi

İşlenecek komutların kodları, sistem tarafından, bellekte ilgili segmentlerdeki adreslere yerleştirilir. Bellekteki bu komutlar çalıştırılacağı zaman, doğrudan veri yolu bağdaştırma birimi tarafından bellekten alınarak kod-çözme birimine getirilirler. Aynı şekilde, icra edilen bir komutun sonucu belleğe yazılacağı zaman, veri yolu bağdaştırma biriminden geriye yazma talep edilir.

Bellekten verinin alınıp getirilmesi veya belleğe tekrar depolanması gibi işlemlerde en çok çalıştırılan komutlar, **yükle ve depola (LOAD ve STORE)** komutlarıdır. Bellekten okuma veya yazma yapılırken, bellek kullanımından veya iş-hattı yapısından dolayı, bazen atıl (bekleme) durumlar ortaya çıkabilir. Bunun için çeşitli mimari geliştirmeler ya da komut eklemeleri yapılmıştır.

İcra biriminin komutlarıyla verinin alınıp getirilmesi veya belleğe depolanması sırasında, veri yolu bağdaştırma birimi, otomatik olarak getirilen verilerdeki komutları, adına **tampon** da denilen, işlemcinin tipine göre kapasitesi değişen komut kuyruğuna yerleştirir.

Bellekten alınıp getirilecek ve işlenecek bu komutların yeri, **CS : IP** kaydedici ikilisi tarafından belirlenir. Bellekle işlenecek programın ilk komutunun bulunduğu adres, bu birim tarafından otomatik olarak CS:IP kaydedicilerine yerleştirilir. Bu ilk komutla birlikte, sıradaki komutlar, komut kuyruğunun kapasitesine göre, veri yolu bağdaştırma birimi tarafından alınıp, getirilip, sıraya konurlar. Sıradaki bu komutların işlenmesiyle birlikte, bellekten veri alıp getirmek için kuyruk veya kanal kapasitesi kadar beklenmeyecek, hemen sıradan alınıp icra birimi tarafından işlenecektir.

Bellekten veya I/O birimlerinden, her ne yöntemle olursa olsun bir şekilde komut kuyruğuna getirilen komut kodları ve operand bilgileri, icra birimi tarafından işlenmek için hazır durumdadır.

2.2 . İcra Birimi

Bu birim, BIU ile birlikte paralel çalışarak, komut kuyruğuna sürülen makina dilindeki komutların kodunun çözülmesi ve işlemci içerisinde her bir komutun doğru bir biçimde ele alınarak işlenmesinden sorumludur. Eğer komutun işlenmesinde bir veriye gerek duyulursa ve veri genel amaçlı kaydedicilerden birindeyse alınıp getirilmesini, eğer gerek duyulan veri harici ortamdaysa , BIU'den bu verinin talep edilmesi gibi işlemleri EU gerçekleştirir.

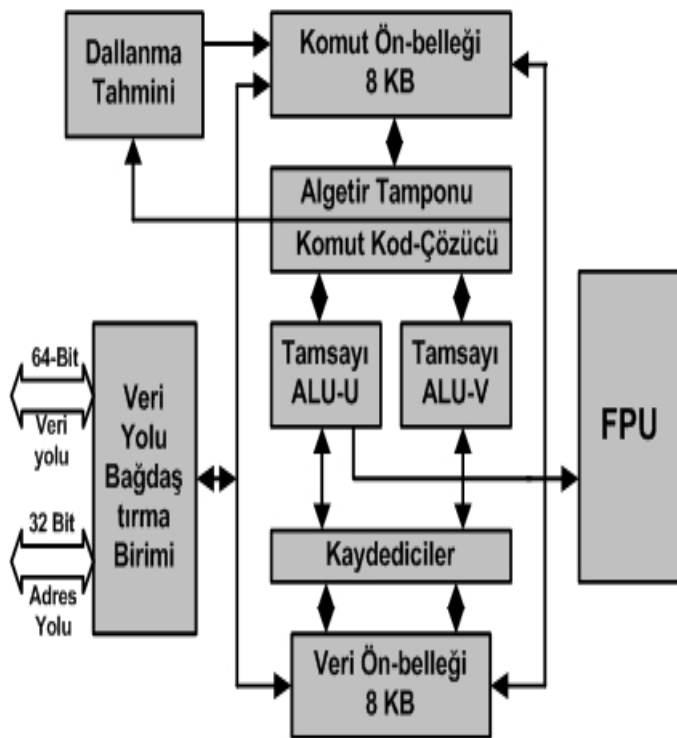
İcra birimi komut kuyruğunun en üstündeki komut kodunu alıp getirdikten sonra, kodlar bir komut kod-çözücüsü vasıtasıyla, ilerisinde komutların mikro karşılıkları olan (**mikro-kod ROM**) ve adına **mikroprogram** denilen bir mikro-kod sıralayıcısının kontrolü altında çözülerek ALU birimine sürülür.

Eğer kodu belirlenen komut bir aritmetik ya da mantık komutuysa, icra birimindeki adres üretici vasıtasıyla BIU uyarılarak, bellekten veri alıp getirerek icra birimindeki kaydedicilere veya doğrudan

ALU'ya yollar ve kontrol biriminin denetiminde, bu veriyle ne yapılması gerekiyorsa o yapılır. Eğer işlemci **FPU** birimi içeriyorsa, büyük değerlikli ve kesirli sayılar bu birime sürülerek orada işlenmesi sağlanır. Eğer mantıksal ya da küçük değerlikli sayılarla aritmetik işlemler yapılıyorsa, ana işlemci yeterli olacaktır. Bir komutun işlenmesinden sonra, komut kuyruğunda bulunan sıradaki komutun ele alınması için kontrol birimi **sin-yal** gönderir.

İlk mikroişlemcilerde bir komutun işlenmesi üç aşamada yapılmaktaydı. Bunlar, **komut kodunun bellekten alınıp getirilmesi, kodunun çözülmesi ve işlenmesi adımlarıydı**. Bir komut kodu bu üç aşamalı iş-hattına girerek belli saat çevrimlerinde işlenmekle-ydi. Bir komut bir kademede işlendikten sonra diğer kademeye geçtiğinde önceki kademe boş duruyordu ve böylece işlemci saat çevrimleri boşa harcanı-yordu. Bu tip çalışma sistemine , **tek kademeli iş-hattı tekniği** denilmek-tedydi.

Daha sonraları, mikroişlemci mimarisindeki gelişimlerle (32-bit işlemciler) birlikte, iş-hattına alınan bir komut bir sonraki sayfaya geçtiğinde, boşalan sayfaya başka bir komut alınmaktadır. Böylece her saat çevriminde iş-hattındaki tüm kademeler bir işle meşgul olmakta ve toplamda birden çok işlem yapılmaktadır. Bu tip işlemci çalışma sistemine, **süper iş-hatlı sistem** denilmektedir.



Şekil - Pentium işlemcinin basitleştirilmiş blok diyagramı

3 . 32 Bitlik Mikroişlemciler

X86 tabanlı mikroişlemci mimarisinde üçüncü kuşağa dahil olan ilk 32-bit işlemci 386 kodlu işlemcisidir. Daha sonraları işlemci içerisine **FPU(Floating Point Unit-Kayan nokta birimi)** denilen ve matematik işlemlerinden sorumlu bir birimin eklendiği ve ilk gerçek ön-bellek sisteminin uygulandığı 32-bit işlemci 486 kodlu işlemciler olmuştur.

Bir sonraki işlemci üretim sürecinde, [AMD ve CYRIX](#) gibi firmaların benzer kodlarla işlemciler piyasaya sürmeleri, Intel firmasının yeni bir adla işlemci üretmesine sebep olmuştur. [Pentium işlemciler](#), adından da anlaşılacağı gibi, beşli demektir. yani 486 serisinin devamı olarak 586 demektir. Bu işlemci 64-bitlik geniş bir harici veri yoluna sahiptir. Geniş veri yolu, işlemcinin bir çevrimlik zamanda daha çok veri taşıması ve dolayısıyla yapacağı görevi daha kısa zamanda yapması demektir.

Bu, işlemcinin bir tıklamasıyla, işlemci ile bellek arasında veya işlemci ile G/Ç birimleri arasında, 8-bitlik bir işlemciye göre 8 kat fazla bilgi taşınması demektir. Dahili veri yolları 128 ile 256 bit genişliğinde olup, veri ve komut kodları çok çabuk olarak, işlemci içerisinde birimler arasında taşınmaktadır.

Şekilde görüldüğü gibi, işlemci dış dünyaya, veri yolu bağdaştırma birimi tarafından, 32-bitlik adres yolu ve 64-bitlik veri yolu vasıtasıyla bağlanmaktadır. Aynı zamanda, bu birim yavaş dış dünyayı hızlı iç dünyadan ayırmaktadır. Bu bağlantılar yine aynı birim üzerindeki kontrol yolu vasıtasıyla denetlenmektedir.

Veri yolu vasıtasıyla alınan veriler, işlemci çekirdeğindeki ayrışık 8'er KB'lık ön-belleklere gönderilir. 8 KB veri ön-belleği iki porta sahiptir, bunlardan her birisi bir iş-hattına bağlıdır. Ön-bellekler adanmış bir [TLB birimine \(Translation Lookaside Buffer\)](#) sahiptir ve bu birim veri ön-belleği tarafından, lineer adreslerin fiziksel adreslere dönüştürülmesinde kullanılmaktadır. Ön-bellekler bu fiziksel adresleri kullanırlarken, veri ön-belleği, gerekli olduğu durumlarda doğrudan yazma ve geriye yazma stratejisini kullanarak çalışabilirler. Her ön-bellek kendi TLB birimine sahiptir.

Komut ön-belleği, [BTB \(Branch Target Buffer\)](#) ve [Prefetch Buffer](#) birimleri, Pentium işlemcinin icra birimlerine ham komutların getirilmesinden sorumludurlar. Komutlar harici veri yolu vasıtasıyla ana bellekten veya L2 ön-bellekten, ya da dahili veri yolu vasıtasıyla L1 ön-belleğinden getirilir. Komut kullanımında eğer dallanma ortaya çıkarsa, bu adresler BTB birimi tarafından hatırlanırlar.

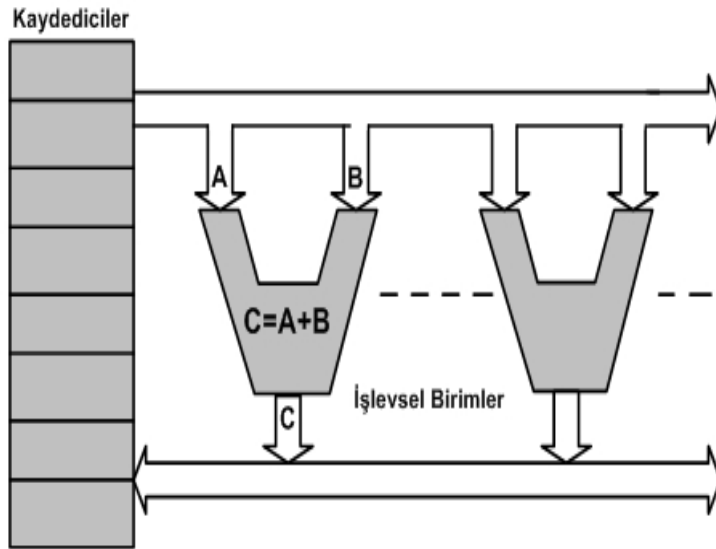
Sistemdeki kontrol biriminin gözetiminde, komut kodları komut ön-belleğinden alınarak [U ve V](#) denilen iki adet tamsayı komut iş-hattına taşınır. [U iş-hattı](#), tamsayı veya kayan nokta sayılara sahip tüm komutları işlerken, [V iş-hattı](#), sadece basit tamsayı komutları ve [FXCH](#) gibi kayan noktalı komutu işleyebilir. Kontrol birimi, büyük kayan noktalı sayıları doğrudan FPU birimine göndererek burada işlenmesini sağlar. Kod-çözme birimleri, iki paralel iş-hattındaki sıralı komutların kodunun çözülmesinde görevlidir. Sistemdeki [Mikrokod ROM](#), işlemci tarafından gerçekleştirilen ardışık operasyonların kontrolünü sağlayan mikro-kodları(uop) içerir.

4 . 64 - Bit Mikroişlemciler

Mikroişlemcilerin performansını yalnızca veri yolunun genişliğine bağlamak doğru değildir. Bir işlemcinin MHz/GHz oranı, saniyede kaç **çevrim (saykıl)** gerçekleştirebileceğini tanımlasa da, her çevrimde ne hacimde işlem kapasitesine sahip olduğunu belirlemez. Bit kapasitesi ile işlemci saat hızı tam burada ayrılmaktadır. Bu, 64-bitlik bir işlemcinin 32-bitlik bir işlemciye nazaran daha fazla işlem yapacağı demektir. Böylece, 64-bitlik bir 1GHz hızındaki **Itanium2 işlemcisi**, 32-bitlik 2.4 GHz hızındaki P4 işlemciye göre dahil hızlı olacak demektir.

32-bitlik mimari yapısındaki kısıtlamalar, veri yollarının ve eskiden kalma mimari düzenin yetersizliği, **IA - 32 bit** teknolojisine dayanan bir komut işleme tekniğinin artık günümüz veri tabanı ve multimedya uygulamalarında yavaş kalmasıdır.

64-bit işlemcileri (Itanium ve Itanium2) 32-bit işlemcilerden ayıran en büyük özellik, mimarisinin 64-bitlik komutları işleyebilmesidir. Gelişmiş işlemciler, **EPIC**, 64-bit komut işleme, dallanma tahmini, spekülasyon ve süperskalar gibi mimari özelliklere sahiptir. Bu özellikler aşağıda sıralanan ana unsurlar içerisine dahildir;



Şekil - 64 - bit modern süperskalar mimarili işlemci

- **Basitlik** : CISC işlemcilerin belleği etkin kullanmak için gösterdiği karmaşıklığı ortadan kaldıran RISC işlemcilerin de gittikçe artan karmaşıklığı basite indirmiştir. CISC ve RISC mimarilerinin yerine, bu sistemlerin üstün özellikleri de alınarak gerçekleştirilen EPIC mimarisinin amacı, işlemciadaki veri akışını basit ve daha akışkan hale getirmektir.

- **Geniřleyebilirlik** : 64-bit iřlemcilerin dayandıęı IA-64 komut kmeli mimari (ISA), 128 adet genel amalı kaydedici ve 128 adet kayan noktalı kaydediciye sahiptir. Bu, iřlemcinin 64-bitlik sperskalar bir mimariye sahip olmasına yol amakladır. řimdiki 4-yollu sperskalar mimari, daha sonraları 8-yollu ve 12-yollu veya n-yollu iplerin geliřine bir iřaret olmaktadır.

- **Paralellik** : EPIC mimarisi, komut kodlarının bir ok koldan paralel iřlenmesine fırsat vermektedir. Komutların birden ok iřlevsel birimlerde aynı anda iřlenmesini saęlamaktadır.

- **Derleyiciye dayalılık** : Derleyici teknolojisindeki iyi arařtırmalara raęmen, yeni mimariler bunun zerinde yeterince avantaj saęlayamamaktadır. EPIC, derleyicilerin tm zelliklerinden faydalanmaktadır.

- **64-bit alıřma** : 64-bitlik mimari aynı zamanda 64-bitlik bilgi iřleme evresi gerektirmektedir. Bunlardan iki ana zellik; iřaretiler iin veri tipleri ve 64-bit geniřlięinde uzun tamsayılardır.