

# MİKROİŞLEMCİLER (BLM202)

## HAFTA - 2

Dr. Bilgin YAZLIK, RTTP, PMP



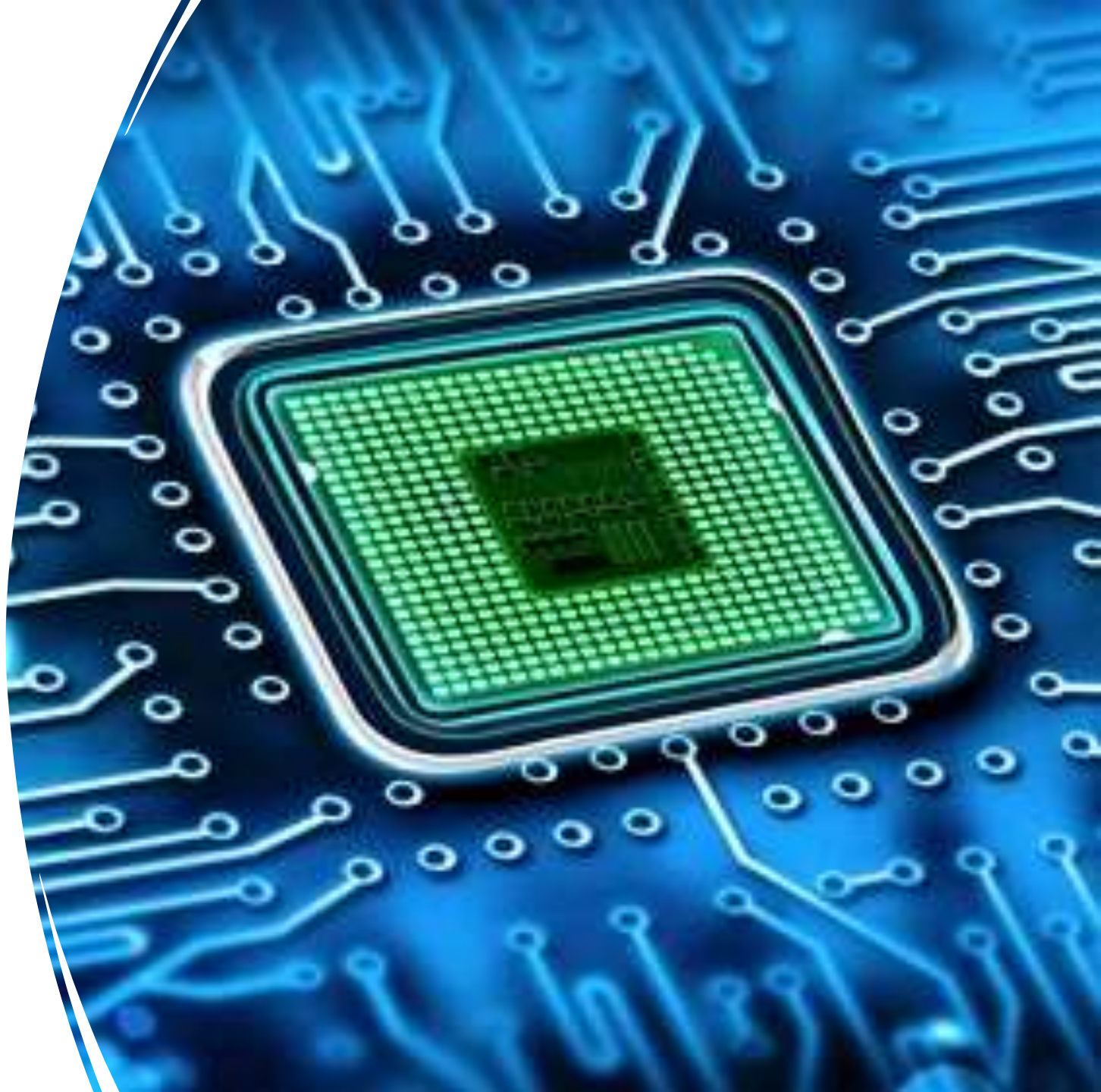
**BİLGİSAYAR MÜHENDİSLİĞİ**



# 2. HAFTA

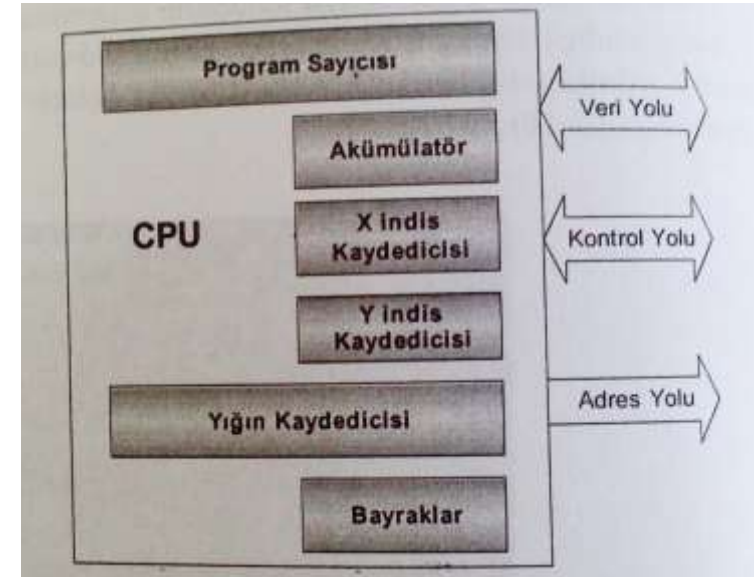
---

- Mikroişlemcinin İç Yapısı:
  - Kaydediciler
  - Aritmetik ve Mantık Birimi (ALU)
  - Zamanlama ve Kontrol Birimi (CU)



# KAYDEDİCİLER (REGISTERS)

- İşlemci içerisinde ham bilgi girdisinin hızlı biçimde işlenerek kullanılabilir çıktıya dönüştürülmesi için sistemde verileri **geçici olarak** üzerinde tutacak bir grup veri saklayıcıya ihtiyaç duyulmaktadır. Kaydediciler verinin **manevrasında** ve geçici olarak tutulmasında görevlidirler.

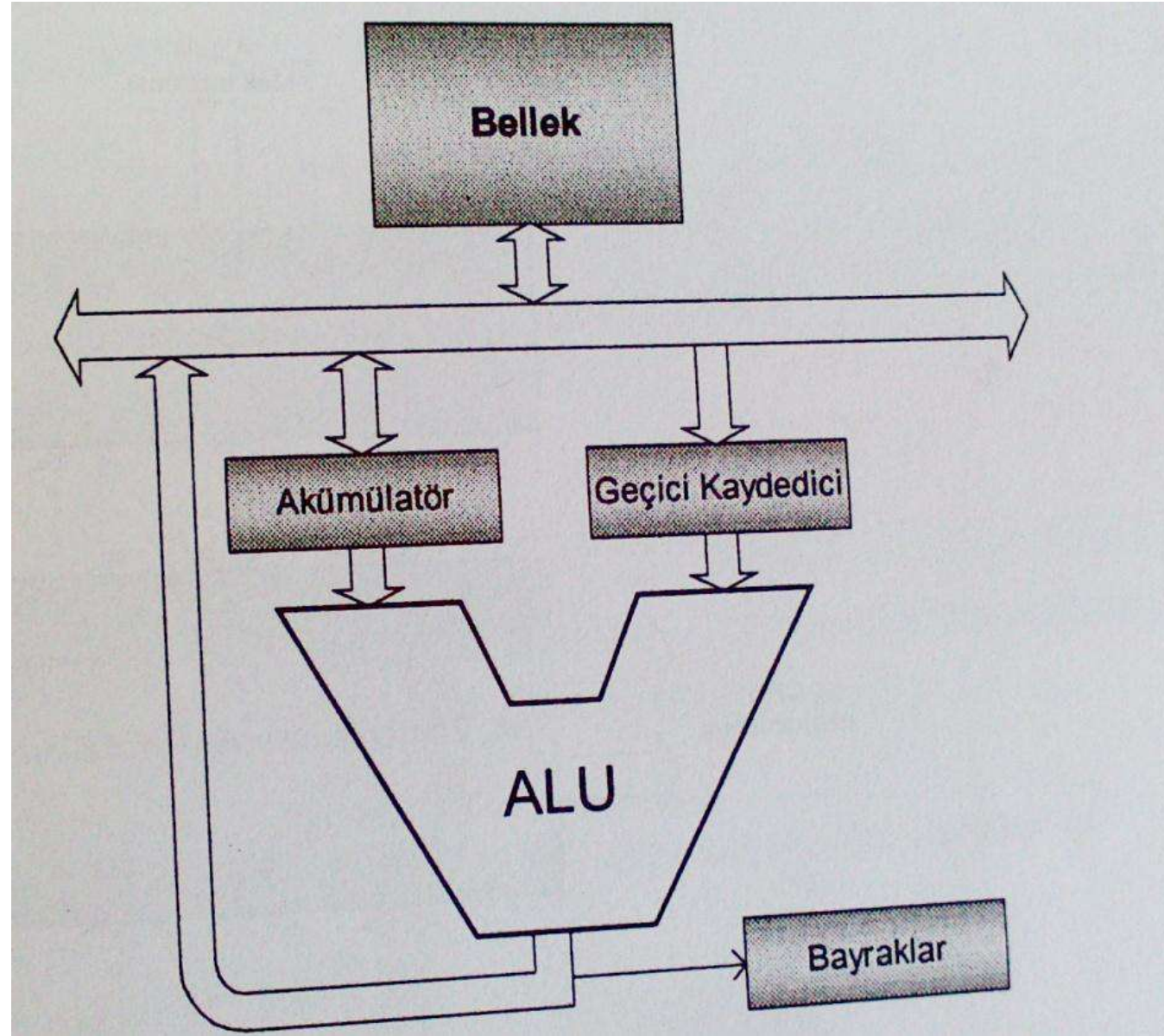


# ARİTMETİK VE MANTIK BİRİMİ (ALU)

- ALU mikroişlemcide **aritmetik ve mantık işlemlerinin** yapıldığı en önemli birimlerden birisidir. Aritmetik işlemler denilince başta **toplama, çıkarma, çarpma, bölme**, mantık işlemleri denilince **AND, OR, EXOR ve NOT** gibi işlemler akla gelir.
- Komutlarla birlikte bu işlemleri mantık kapılarının oluşturduğu toplayıcılar, çıkarıcılar ve kaydıran kaydediciler gerçekleştirirler. ALU'da gerçekleşen bütün bu işlemler kontrol sinyalleri vasıtasıyla Zamanlama ve Kontrol Biriminin gözetiminde **eşzamanlı** olarak yapılır.



# ALU



# ALU

- ALU'da **basit matematik komutlar zorlanmadan işlenebilir** fakat karmaşık aritmetik işlemleri (**doğrudan bir komutla çarpma, bölme, karekök alma**) gerçekleştirmek için ayrı alt yordam gruplarına veya **ek elektronik devrelere** ihtiyaç duyulur. 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.
- Gelişmiş işlemlerde kayan noktalı aritmetik işlemleri gerçekleştirmek üzere **FPU (Floating Point Unit)** bir işlemci daha yerleştirilmiştir. Bu sayede **küçük haneli veya küçük kesirli** sayılarla işlem yapılabilir. Eğer sistemde FPU mevcutsa ağır matematiksel işlemler bu işlemci tarafından yapılırken ana işlemci diğer işlemlerle meşgul olacağından sistemde yavaşlama yok denecek kadar az olur.

# ALU

- İcra birimindeki komutun kodu çözüldükten sonra yapılacak işlem aritmetik ya da mantık işlemi olacaktır. Bu işlemin yürütülmesi işi ALU (Arithmetic and Logic Unit) birimine verilmiştir. Bu birimde bayt veya word olarak basit dört işlemin yanında verinin bit olarak artırılması ya da azaltılması, sağa sola kaydırılması veya yönlendirilmesi işlemleri yapılabilir. Aynı zamanda ALU bayt veya word verisi üzerinde bit-bit mantıksal işlemlerin yapılması için de çok elverişlidir. AND, OR ve XOR komutlarıyla, iki bayt veya iki word verisi arasında bit eşlemesi işlemleri yapılabilir.
- İlk mikro işlemcilerde çarpma bölme gibi aritmetik komutların yerine getirilmesinde sağa veya sola kaydırma komutlarından faydalanılmaktaydı, fakat 16 bitlik işlemcilerle birlikte mimarinin gelişmesine paralel olarak komut kümelerindeki artış, bu işlemleri doğrudan yapabilecek MUL ve DIV gibi komutları da beraberinde getirmiştir. Hangi komut kullanılırsa kullanılsın toplayıcı ve çıkarıcı devreleri çarpma ve bölme işlemlerinde çarpma için üstüste toplama, bölme için üstüste çıkarma şeklinde kullanılmaktadır.

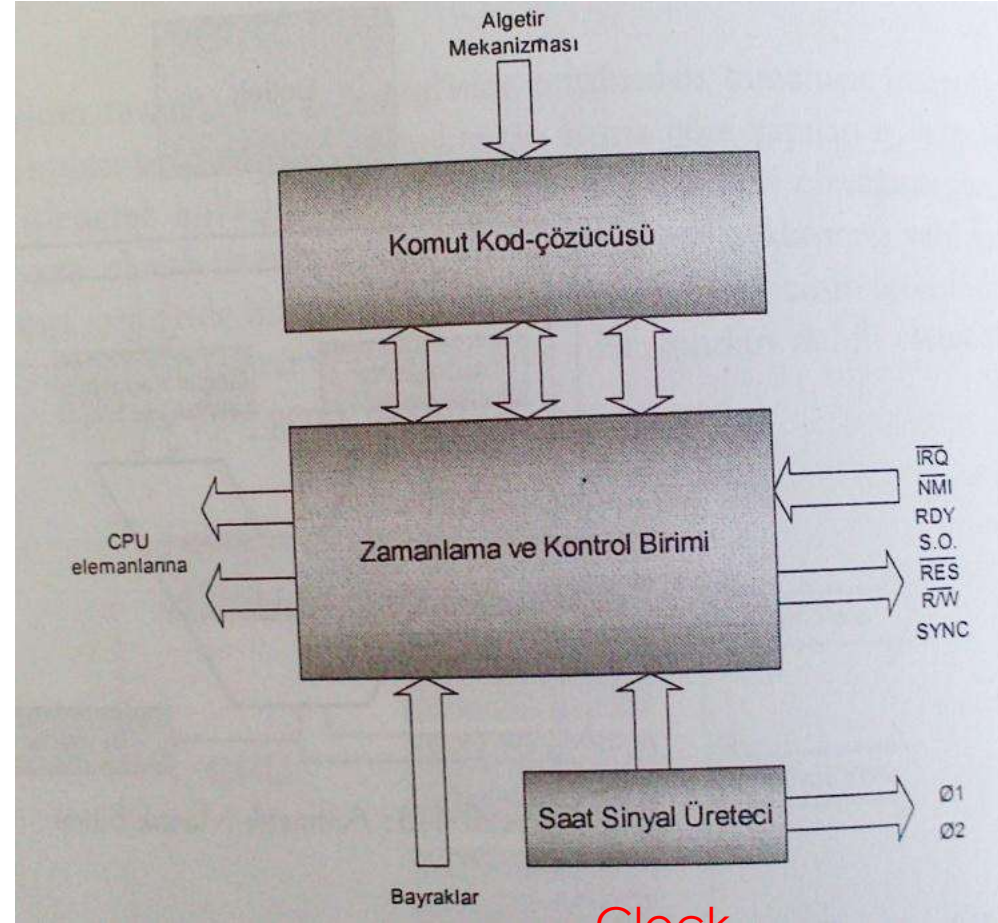
# ALU

- ALU gerekli işlemi bitirdikten sonra adına **bayrak kaydedicisi** denilen hücreye yazılacak durum bitlerini kontrol eder. Bu bitler bir sonraki işlenecek komuta etki edebilir. Mikroişlemci tüm bu işlemlerin düzenli bir şekilde yürütülebilmesi için veri yolu bağdaştırma birimi ile icra biriminin paralel olarak çalışması gerekir. Herhangi bir kopuklukta makine kilitlenebilir.



# ZAMANLAMA VE KONTROL BİRİMİ

Fetch



Clock

# ZAMANLAMA VE KONTROL BİRİMİ

Zamanlama ve kontrol birimi, bellekte program bölümünde bulunan komut kodunun alınıp getirilmesi, kodunun çözülmesi, ALU tarafından işlenilmesi ve sonucunun alınıp belleğe geri konulması için gerekli olan kontrol sinyalleri üretir. Bilgisayar sisteminde bulunan dahili ve harici bütün elemanlar bu kontrol sinyalleri ile denetlenir.

Basit bir mikroişlemcide bu bölüm 3 değişik işlevi yerine getirir:

1. Zamanlama Kontrolü: İşlemci, harici saat sinyali üreten bir birimden giriş alan iç-saat devresine sahiptir. Bir sinyal alınarak talebe göre zaman sinyallerine çevrilerek sisteme dağıtılır.
2. Komut kod çözücüsü: Bu devre komut kaydedicisinde tutulan komutları yorumlar ve ALU'ya kaydedicilerle çalışması için uygun sinyaller gönderilir.
3. Kesme ve Mantık Birimi: Bu birim diğer kontrol elemanlarına benzer. Gerekli durumlarda kesme sinyallerini alarak işlemciyi uyarır.



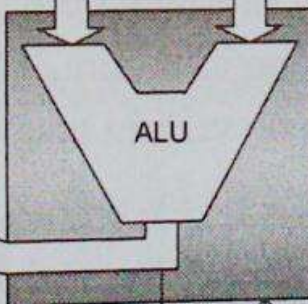
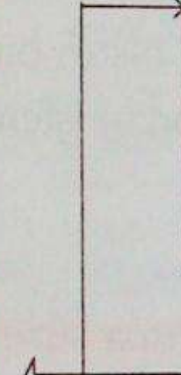
# 16 BİTLİK MİKROİŞLEMCİLER

16 bitlik mikroişlemciler basit olarak 8 bitlik işlemcilerde olduğu gibi Kaydedici bölümü, ALU ve Zamanlama-Kontrol birimine sahiptir. Fakat mimari yapısı çoklu görev ortamına uygun hale getirildiğinden, işlemci içerisindeki bölümler de fonksiyonel açıdan iki mantıksal ana bölüm halinde daha ayrıntılı olarak açıklanmaktadır. 16 bitlik x86 tabanlı işlemciler veri yolu bağdaştırma birimi (BIU) ve icra birimi (EU) olmak üzere iki ana bölümde toplanabilir. BIU birimi, EU birimini veriyle beslemekten sorumluyken, icra birimi komut kodlarının 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. Bu birimler çip üzerinde birbirine fiziksel olarak bağlıdır.

## İCRA BİRİMİ (EU)

Genel Amaçlı  
Kaydediciler

AH	AL
BH	BL
CH	CL
DH	DL
SI	
DI	
BP	
SP	



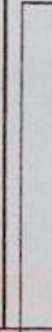
Bayraklar

Kontrol Birimi

## VERİ YOLU BAĞDAŞTIRMA BİRİMİ (BIU)

Segment  
Kaydedicileri

CS
DS
SS
ES
IP

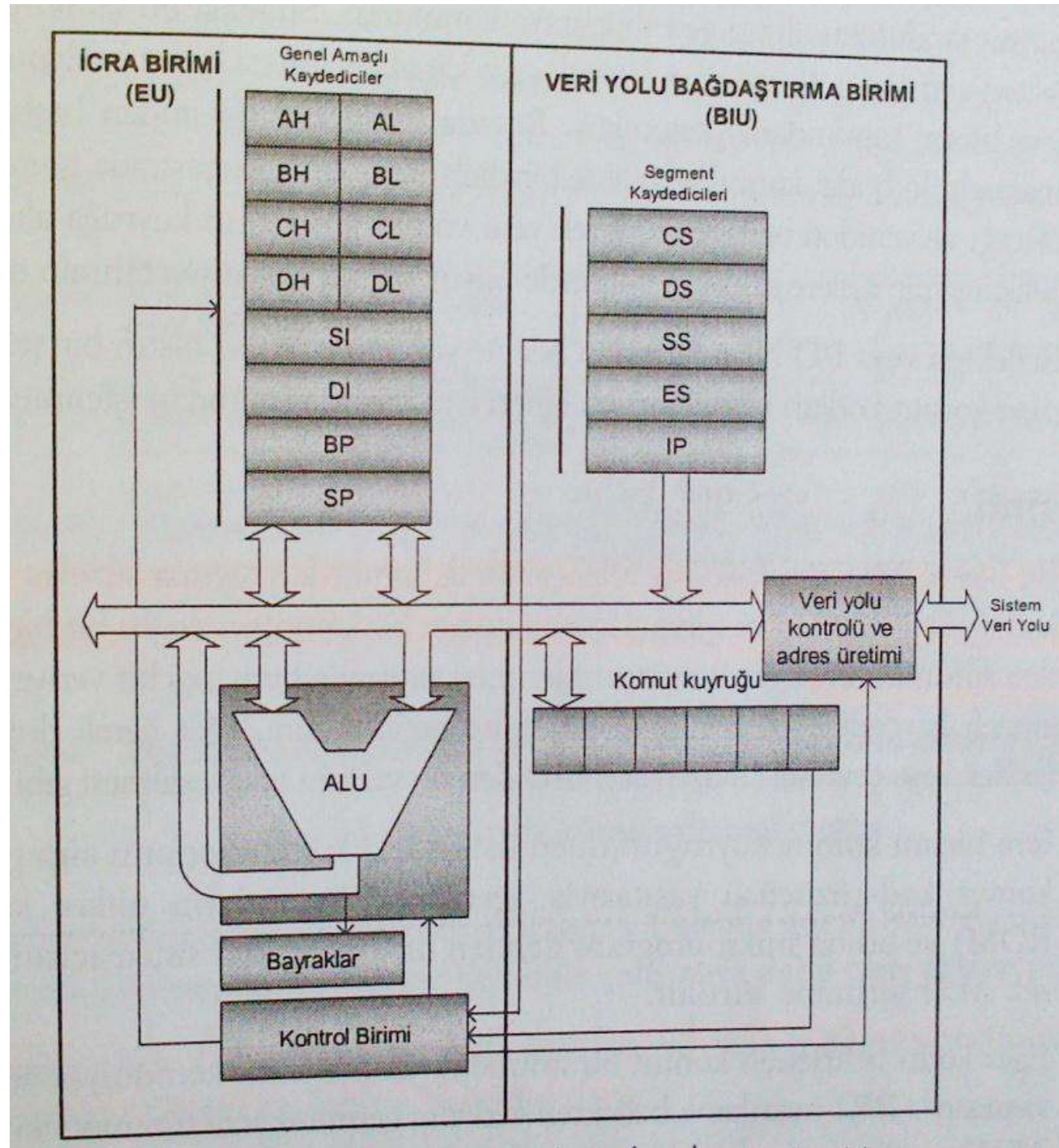


Veri yolu  
kontrolü ve  
adres üretimi

Komut kuyruğu



Sistem  
Veri Yolu





# VERİ YOLU BAĞDAŞTIRMA BİRİMİ (BIU)

- İşlenecek komutların kodları sistem tarafından bellekle ilgili segmentlerdeki adreslere yerleştirilir. Bellekteki bu komutlar çalıştırılacağı zaman doğrudan veri yolu bağdaştırma birimi tarafından **bellekten alınarak** kod çözme birimine getirilirler. Ayrıca icra edilen bir komutun sonucu belleğe yazılacağı zaman veriyolu bağdaştırma biriminden **geriye yazma** talep edilir.
- İcra biriminin komutlarıyla verinin alınıp getirilmesi veya belleğe depolanması sırasında veri 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** ikilişi tarafından belirlenir. Bellekte işlenecek programın ilk komutunun bulunduğu adres bu birim tarafından otomatik olarak CS:IP kaydedicilerine yerleştirilir.
- 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.



# İCRA BİRİMİ (EU)

- Bu birim, BIU ile birlikte paralel çalışarak komut kuyruğuna sürülen makine dilindeki komutların kodunun çözülmesi ve işlemci içerisinde her bir komutun doğru bir biçimde ele alınarak işlenmesinden sorumludur. Eğer komutun işlenmesi sırasında herhangi bir veriye gerek duyulursa ve veri genel amaçlı kaydedicilerden birindeyse alınıp getirilmesi, eğer veri harici ortamdaysa (bellek veya çevresel cihazlarda) **BIU'dan bu verinin talep edilmesi gibi işlemleri EU gerçekleştirir.**
- İcra birimi komut kuyruğunun en üstündeki komut kodunu alıp getirdikten sonra, kodlar bir komut-kod çözücü vasıtasıyla, içerisinde komutların mikro karşılıkları olan (mikro-kod ROM) ve adına **mikroprogram denilen bir mikro-kod sıralayıcısının kontrolü altında çözülerek ALU birimine sürülür.**
- Eğer kodu belirlenen kod bir aritmetik ya da mantık komutuysa, icra birimindeki adres üretici vasıtasıyla BIU uyarılarak **bellekten veri alıp getirilerek icra birimindeki kaydedicilere veya doğrudan ALU'ya** yollar ve kontrol biriminin denetiminde bu veriyle ne yapılması gerekiyorsa o yapılır.

# İCRA BİRİMİ (EU)

- Komutlar işlendikten sonra sonuç bilgisi nereye depolanacaksa (bir kaydediciye veya bellek alanına) oraya gönderilir. Bir komutun işlenmesinden sonra komut kuyruğunda bulunan sıradaki komutun ele alınması için kontrol birimi sinyal gönderir.

- İlk mikroişlemcilerde bir komutun işlenmesi üç aşamada yapılmaktaydı.

1. Komut kodunun bellekten alınıp getirilmesi
2. Kodunun çözülmesi
3. İşlenmesi

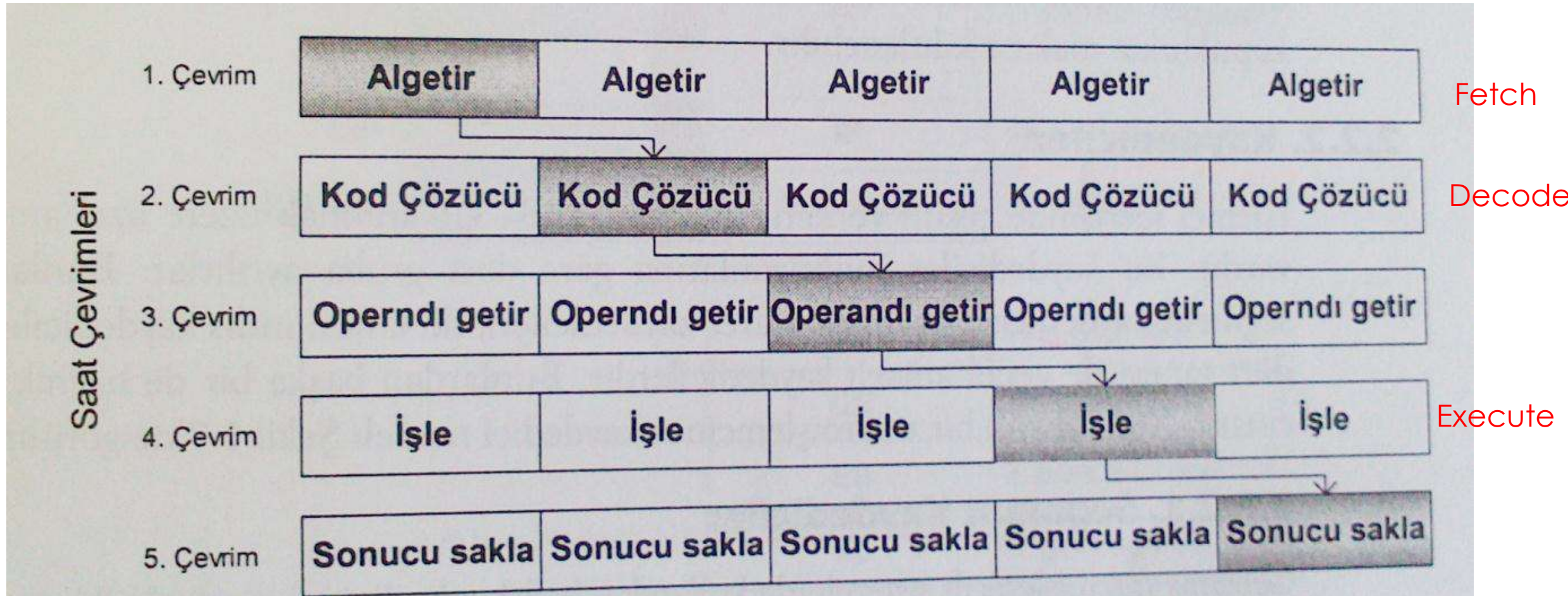
- Bir komut kodu üç aşamalı iş hattına girerek belli saat çevrimlerinde işlenmekteydi. Bu teknikte iş hattına giren bir komut icra edilmeden ikinci bir komut hatta alınmıyordu. Bir komut bir kademede işlendikten sonra diğer kademeye geçtiğinde önceki kademe boş duruyordu ve böylece işlemci çevrimleri boşa harcanıyordu. Bu tip çalışma sistemine tek kademeli iş-hattı tekniği denmektedir.

# İCRA BİRİMİ (EU)

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

- Kısaca iş hattı tekniği, mikroişlemcinin farklı kaynaklar kullanarak komutları ardışık adımlara bölmesi işlemine denir. Bundan dolayı işlemci bir çok işlemi paralel olarak aynı anda yapar.

# İCRA BİRİMİ (EU)



# İCRA BİRİMİ (EU)

- Mikroişlemcinin her defasında komut alıp getirmesi ve onu işlemesi sistemi yavaşlatır. Bu yavaşlığı gidermek için **iş hattı tekniği** yeterlidir. Fakat tekrar belleğe erişilmek istendiğinde bu defada **belleğin işlemci hızında olmaması** ve ona gerektiği hızda cevap vermemesi işlemci tasarımcısına bir sorun daha ortaya çıkarmıştır. Burada verinin akışını hızlandıran ve sistemin bir bütün halinde aynı hızda çalışmasını sağlayan bir dizi tedbirler alınmıştır.
- Günümüz işlemcilerinde bu tip sorunlar Dinamik Çalışma adı altında işlemciye kazandırılan yeteneklerle yok edilmeye çalışılmıştır. Dinamik çalışmaya dahil olan öğeler şunlardır:
- Mikroişlemcinin bir sonraki komutla birlikte ele alacağı komut gurubunun tahmin edilmesi işlemine Çoklu **Dallanma Tahmini** denir.
- Komutlar arasındaki bağımlılıkları analiz** eden bir veri akış analizi ve
- İlk iki öğenin sonuçlarını kullanarak komutları **spekülatif çalıştırma** işlemidir.
- Sistemdeki elemanların hızlarının birbirine yakın olmasının yanında ön bellekler ve süperskalar mimarisi hızda önemli rol oynamaktadır.



# KAYDEDİCİLER (REGISTERS)

- İşlemci içerisinde çeşitli verilerin manevrasında kullanılmak üzere özel amaçlı **14 kaydedici** vardır. Bu kaydediciler fonksiyonlarına göre dört gruba ayrılırlar: Bunlardan
  - 4 tanesi segment kaydedicisi
  - 3 tanesi işaret kaydedicisi
  - 2 tanesi indis kaydedicisi
  - 5 tanesi genel amaçlı kaydedicilerdir.
- Bunlardan başka bir de bayraklar mevcuttur.

# SEGMENT KAYDEDİCİLER

- **Büyük kapasiteli** belleklerde bilginin yönetimi (yüklenmesi, saklanması ve sırasını bekleme) oldukça **karmaşıktır**. Bu sebeple büyük bellekler belli amaçlarla **64 KB'lık küçük gruplara (segmentlere)** ayrılarak daha kolay yönetilirler. Bellekte bu bölümlerin başlangıç adresi segment kaydedicileri tarafından tutulurlar. Bu bölümdeki verilerin adresleri ise, segment kaydedici içeriğine uzaklığıdır ve ofset adres olarak anılırlar.

- Gösterdikleri bellek alanlarının oluşumuna göre segment adları özel amaçları olarak verilmiştir. Buna göre programcı tarafından yazılan komut kodlarının assemblara bağlı olarak sistem tarafından bellekte saklandığı bölüme kod segment adı verilir. Kod segmentteki komutlarla ilgili olan veya bu kodların işleyeceği verilerin saklandığı başka bir bölüme **data segmenti** denilir. Verilerin çok büyük olduğu istatistiksel bilgilerin tutulduğu data segmentinin yetmediği durumlarda ikinci bir bölüm vardır ki buna ekstra segment adı verilir. Networking ve korumalı mod işlemlerinde bu data segmentlerinin büyüklüğü de yetmezse, 386 ve daha sonraki işlemcilerde F segment ve G segment denilen bölümler tanımlanmıştır. Bunun anlamı bellek miktarının mikroişlemci mimarisi ile birlikte paralel olarak artması demektir.

# SEGMENT KAYDEDİCİLER

- Mikroislemcilerde komutların işlenmesi sırasında, kaydedicilerin **yetersiz gelmesi durumunda** verinin geçici olarak bellekte atıldığı özel bölüme **Yığın Segmenti** denilir. Bellekteki bu bölümlerin başlangıç adresi kendi adlarıyla işlemci içerisinde yer alan kaydedicilerde tutulur.
- **Kod segment kaydedicisi**: Kod segment bellekte **çalıştırılacak komutların** sıralı bir şekilde bulunduğu bölümdür.
- **Data Segment Kaydedicisi**: Normalde işlenecek verinin depolandığı bellek alanının başlangıç adresini gösterir. Diğer segment kaydedicileri gibi tam adresin segment tarafını gösterir.
- **Ekstra segment kaydedicisi**: ES olarak adlandırılan bu kaydedici, programcı tarafından tanımlanmadıkça işlemci bunu kullanmaz. Genellikle string işlemlerinde hedef adresi olarak algılanır.

# SEGMENT KAYDEDİCİLER

- **Yığın Segment Kaydedicisi:** Kısaca SS olarak bilinen bu kaydedicinin gösterdiği bellek alanına verilen ad adından da anlaşılacağı gibi bir takım veri işlenirken **yer yokluğundan** veya kaydedici yetersizliğinden dolayı verinin geçici olarak yerleştirildiği yerdir. Yığına veriler geçici olarak atılabildiği gibi, program içerisinde altyordam çağrılmasında veya diğer yüksek düzeyli dillere parametre geçişleri yapılmasında çok sık kullanılmaktadır.

# HEXADECIMAL

DECIMAL	HEX	BINARY
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

4 BİT VERİ YOLU  
D<sup>H</sup>

8 BİT VERİ YOLU  
1E<sup>H</sup>

16 BİT VERİ YOLU  
F02D<sup>H</sup>

32 BİT VERİ YOLU  
A390110A<sup>H</sup>

4 BİT VERİ YOLU  
1101

8 BİT VERİ YOLU  
0001 1110

16 BİT VERİ YOLU  
1111 000 0010 1101

32 BİT VERİ YOLU  
1010 0011 1001 0000  
0001 0001 0000 1010



# Kaynaklar

---

- Feza Buzluca, İTÜ Ders Notları, Bilgisayar Mimarisi
- Wikipedia
- Emel Soylu, Kadriye Öz, Karabük Üniversitesi, Mikroişlemciler Ders Notları

