

MİKROBİLGİSAYAR SİSTEMLERİ



Teknik Bilimler Meslek Yüksekokulu

Dersin Amacı

 **Mikroişlemciler**

 **Mikrodenetleyiciler**

 **PIC Mikrodenetleyiciler**

 **Micro BASIC Programlama**

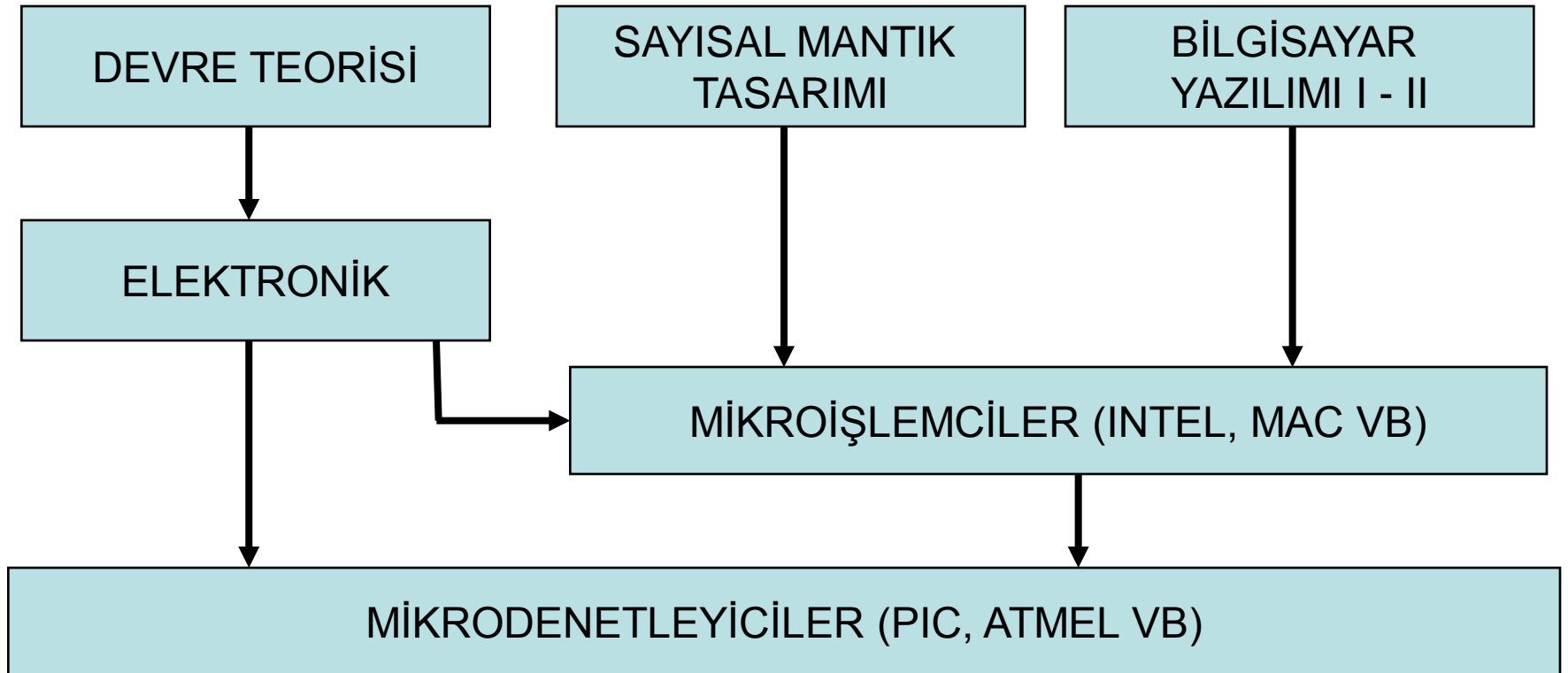
Kullanılacak Programlar

■ MSDOS DEBUG

■ PROTEUS ISIS 7

■ Micro BASIC Editör

MİKROİŞLEMCİLER / MİKRODENETLEYİCİLER İLE İLGİLİ DERSLER



MİKROİŞLEMCİLER AÇIKLAMA

Assembly Programı	C Programı	Pascal Programı
MOV AH,02 MOV DL,41 INT 21 INT 20	<pre>#include <stdio.h> main() { printf("A"); }</pre>	<pre>begin write('A') end.</pre>
Assembler Bilgileri MS-DOS DEBUG	Derleyici Bilgileri MS-DOS için Turbo C 2.01	Derleyici Bilgileri MS-DOS için FreePascal 0.9
Boyut : 8 bayt	Boyut : 8330 bayt	Boyut : 95644 bayt

MİKROBİLGİSAYAR MİMARİLERİ

1. Bellek Organizasyonu açısından

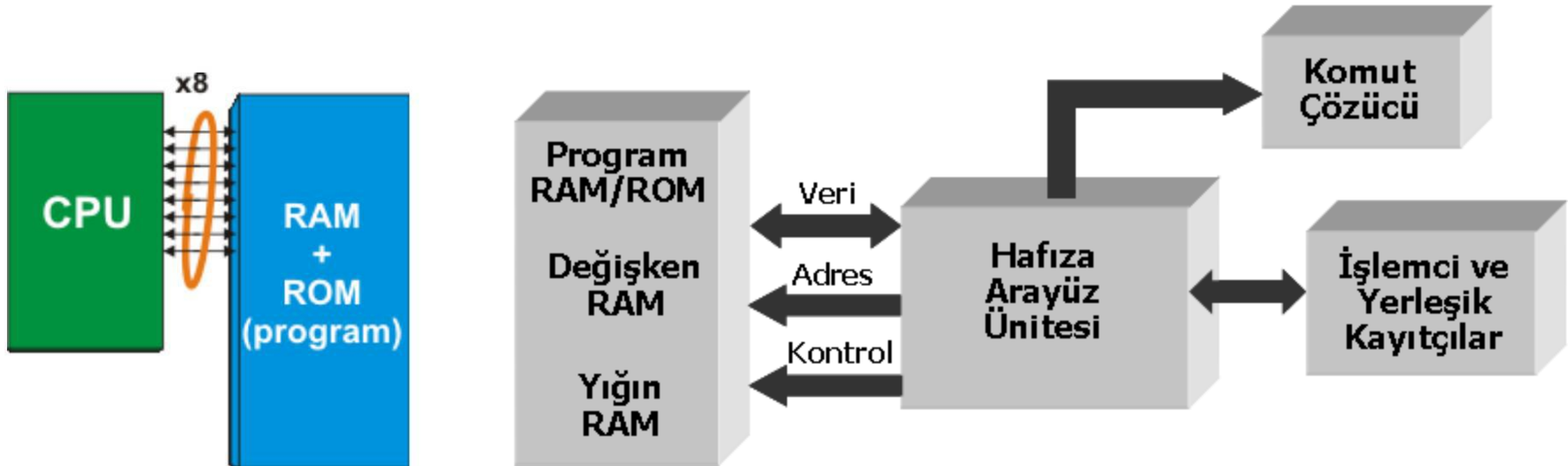
- Von Neuman Mimarisi
- Harvard Mimarisi

2. Komut İşleme Tekniği açısından

- CISC (**C**omplex **I**nstruction **S**et **C**omputer)
- RISC (**R**educed **I**nstruction **S**et **C**omputer)

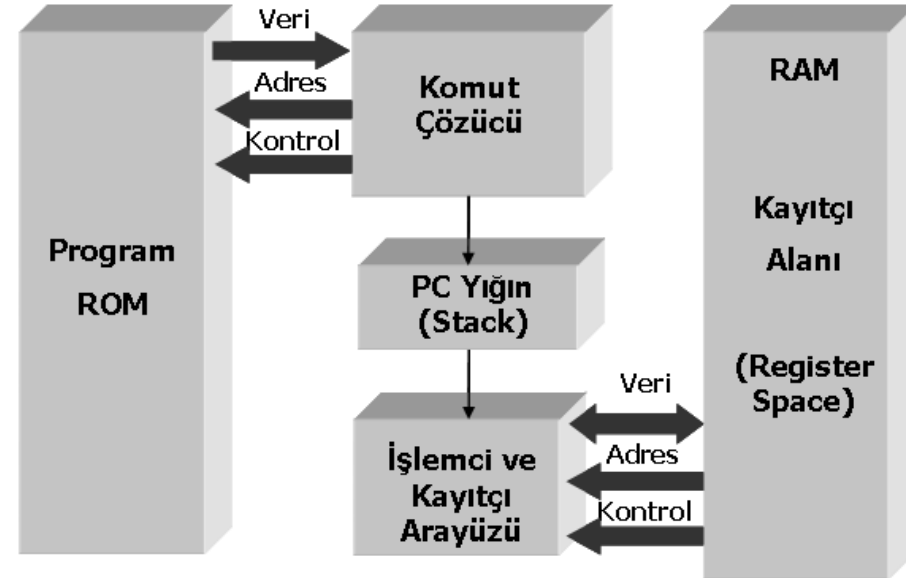
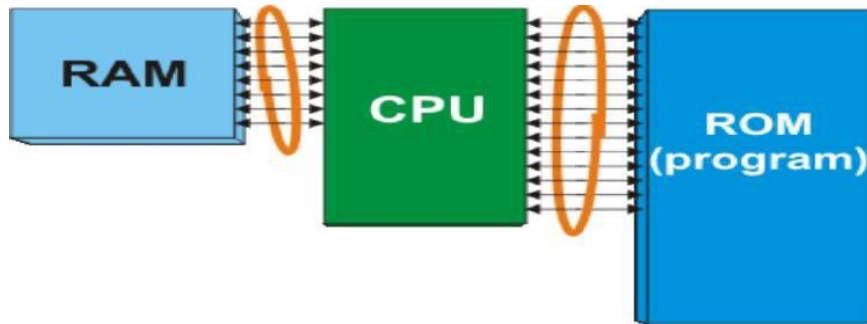
VAN NEUMAN MİMARİSİ

- Veri ve Program alanı aynı hafıza haritası üzerinde bulunur
- Kullanan işlemciler: 80X86, 68HC11, v.b.
- Von Neuman mimarisi PC olarak bilinen kişisel bilgisayarlar arasında standarttır.



HARVARD MİMARİSİ

- Veri ve Program alanı ayrı hafıza haritası üzerindedir
- Güvenilirdir
- Kullanan işlemciler: 8051 , PIC



CISC ÇEKİRDEK

- Mikroişlemci çok sayıda komut içerir ve her eylem için bir komut tanımlanmıştır.
- Temel prensip: donanım her zaman yazılımdan hızlıdır
- Her işlem için farklı bir komut kullanmak işlemleri hızlandırır ancak donanımın yükü artar
- Intel x86 işlemciler(Pentium,AMD gibi), Motorola68000 ailesi, VAX, PDP-11, IBM360

CISC ÇEKİRDEK



- Çeşitli olan komutları çalıştırmak için mikro-kod kullanılmaktadır.
- Komutların çözümünde oldukça karmaşık devrelere (kod çözücülere) ihtiyaç vardır.

RISC ÇEKİRDEK

- Karmaşık assembly dili komutlarının tamamı kullanılmamaktadır.
- Daha az, basit ve hızlı komutlar, uzun, karmaşık ve daha yavaş CISC komutlarından daha verimlidir.
- RISC mimari, daha basit komutlar kullanarak çip karmaşıklığını azaltmaktadır.
- SPARC Alpha, ARC, ARM, AVR, MIPS, PA-RISC, PIC, PowerPC gösterilebilir

RISC ÇEKİRDEK



- RISC komutlarının daha kısa olması belirli bir görevin tamamlanabilmesi için daha fazla komuta gereksinim duyulmasına yol açabilir.
- RISC mimariler için üretilen derleyiciler daha önce CISC mimarisinde bulunan donanım birimlerinin görevini üstlenmek üzere ekstra komutlar kullanmaktadır.

MİKROİŞLEMCİ NEDİR?

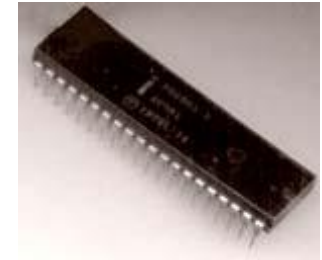
- ❑ Mikroişlemciler kontrol ettiği sistemin beynidirler
- ❑ Bellekler ve Giriş/Çıkış portları gibi birimleri kullanır ve denetler
- ❑ Fakat denetlediği birimlerin hepsi işlemci dışındadır
- ❑ Bu birimlere ADRES, VERİ ve KONTROL yolları ile bağlıdırlar
- ❑ Bu durum, mikroişlemciyi kapasite olarak esnek yaparken diğer birimlere bağımlı hale getirir
- ❑ Mikroişlemciler enerji uygulansa bile diğer birimler olmadan çalışamazlar
- ❑ Çalıştıracağı komut ve bilgilerin tümünü dışarıdan alırlar.

MİKROİŞLEMCİLERİN EVRİMİ

Name	Date	Transistors	Microns	Clock speed	Data width	MIPS
8080	1974	6,000	6	2 MHz	8 bits	0.64
8088	1979	29,000	3	5 MHz	16 bits 8-bit bus	0.33
80286	1982	134,000	1.5	6 MHz	16 bits	1
80386	1985	275,000	1.5	16 MHz	32 bits	5
80486	1989	1,200,000	1	25 MHz	32 bits	20
Pentium	1993	3,100,000	0.8	60 MHz	32 bits 64-bit bus	100
Pentium II	1997	7,500,000	0.35	233 MHz	32 bits 64-bit bus	~300
Pentium III	1999	9,500,000	0.25	450 MHz	32 bits 64-bit bus	~510
Pentium 4	2000	42,000,000	0.18	1.5 GHz	32 bits 64-bit bus	~1,700
Pentium 4 "Prescott"	2004	125,000,000	0.09	3.6 GHz	32 bits 64-bit bus	~7,000



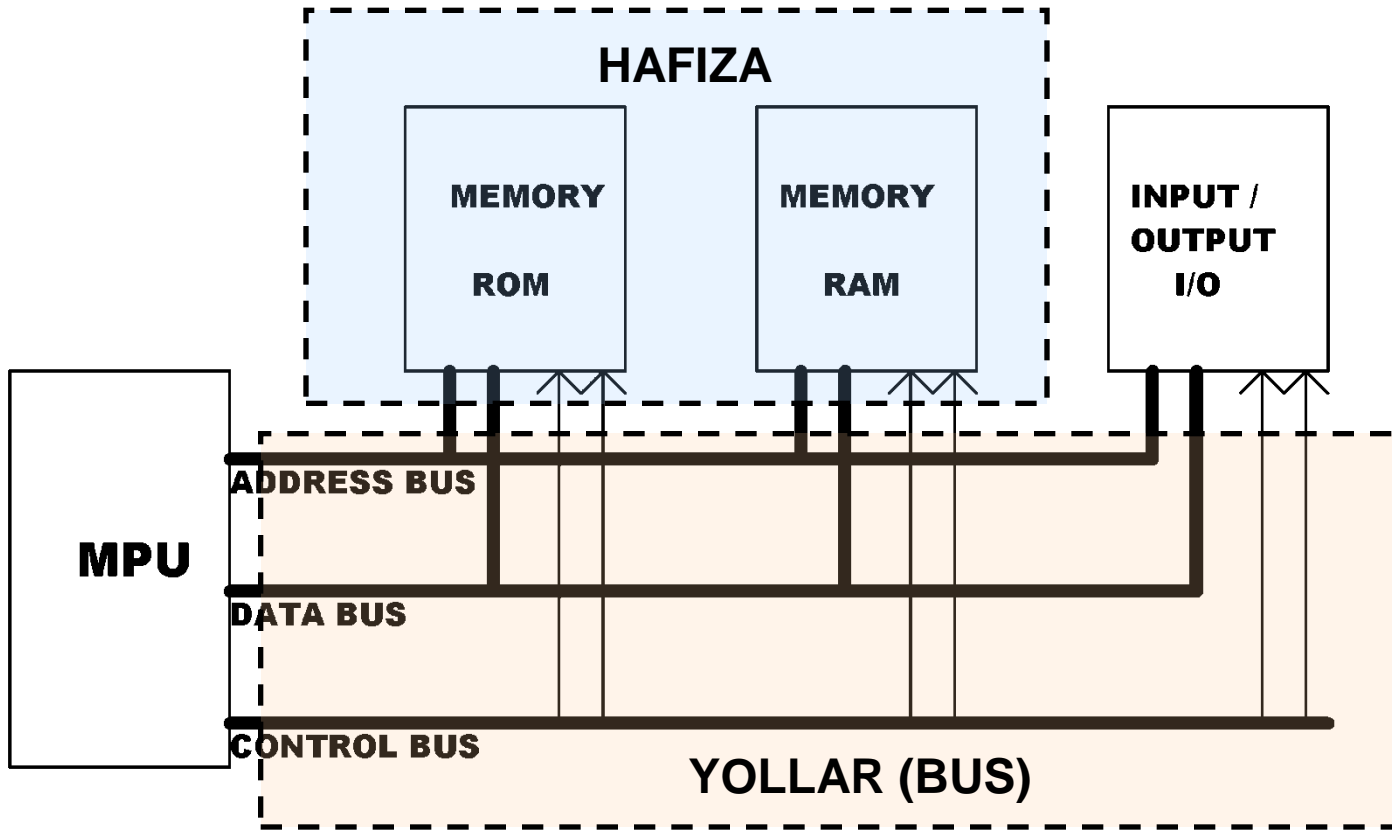
Intel 4004



Intel 8080



MİKROİŞLEMCİ TABANLI BİR SİSTEMİN BLOK ŞEMASI



BELLEK (HAFIZA)

Belleğin Görevleri

1. Program kodunu ve verileri saklar.
2. Mikroişlemciye veri gönderir.
3. Mikroişlemciden gelen veriyi kabul eder.

BELLEK (HAFIZA)

Ana Hafıza Tipleri

- ROM: (Read Only Memory)

İçinde bir program kodu bulundurur, değişmeyen hafıza diye adlandırılabilir, elektrik kesilse bile veri kaybolmaz, mikroişlemci ROM'dan sadece veri okur.

- RAM: (Random Access Memory)

İçinde program kodu ve veri bulundurur, bu hafızaya veri yazılabilir ve okunabilir, elektrik kesilince içeriği kaybolur.

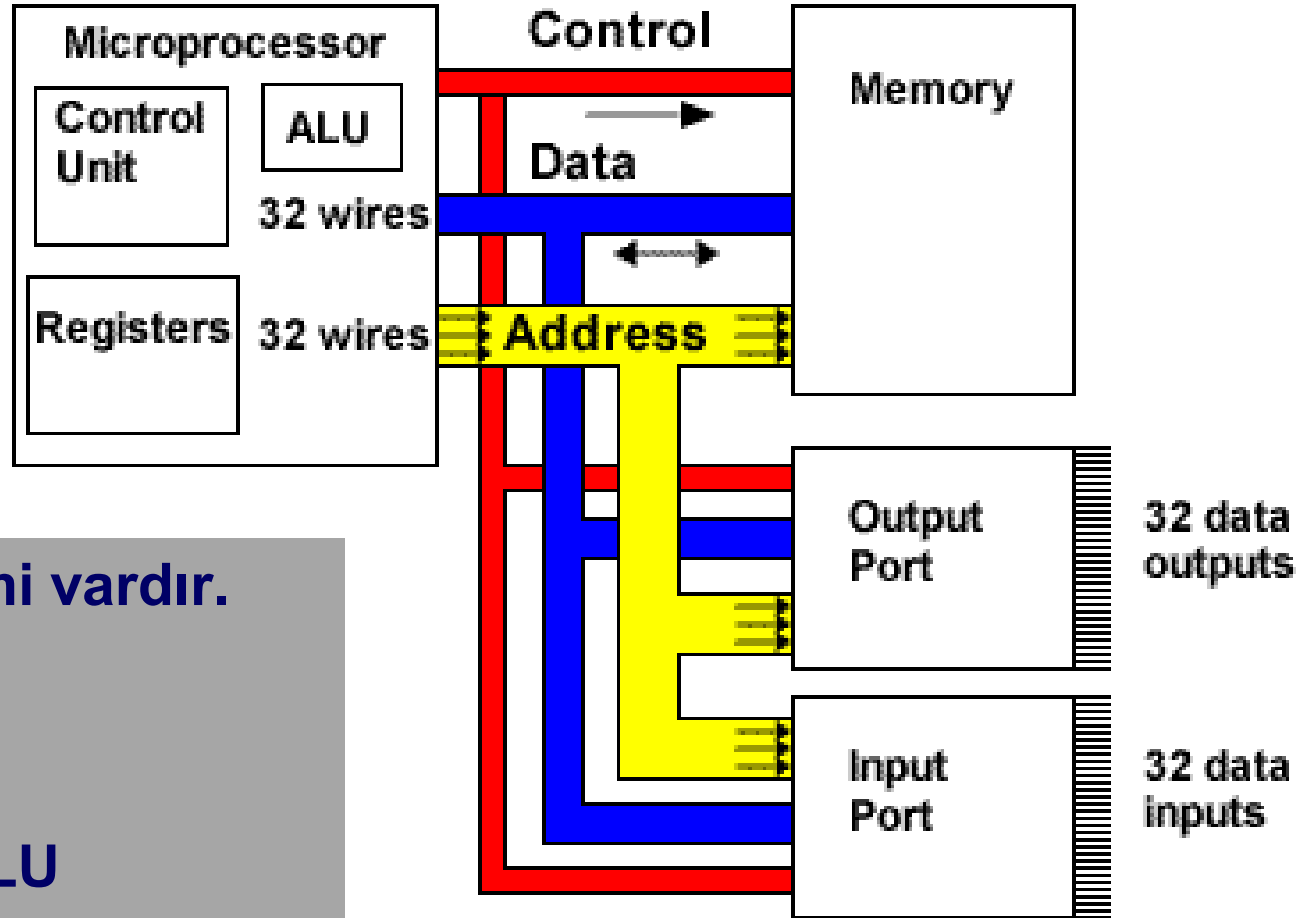
YOL (BUS) SİSTEMLERİ

■ Mikroişlemci tabanlı bir sistemi oluşturan elemanların birbirleriyle bağlantısını sağlayan, bu elemanların arasındaki veri alışverişini gerçekleştiren devrelere YOL(BUS) Sistemleri denir.

■ YOL (BUS) kelimesi baskı devre üzerine döşenmiş bir grup kabloya verilen isimdir. YOL Sistemleri bu kablolar ve gerekli devre elemanlarından oluşan sisteme denir.



YOL (BUS) SİSTEMLERİ



Üç tane YOL Sistemi vardır.

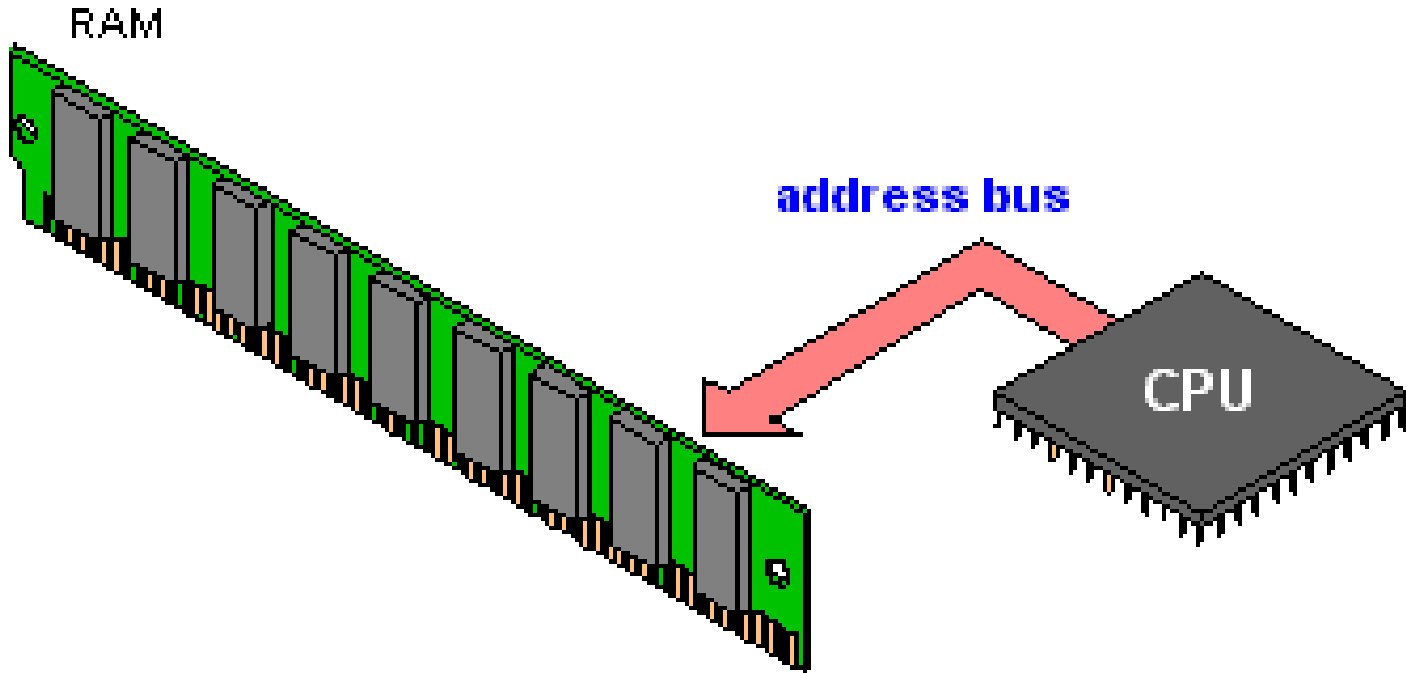
1. ADRES YOLU

2. DATA (VERİ) YOLU

3. KONTROL YOLU

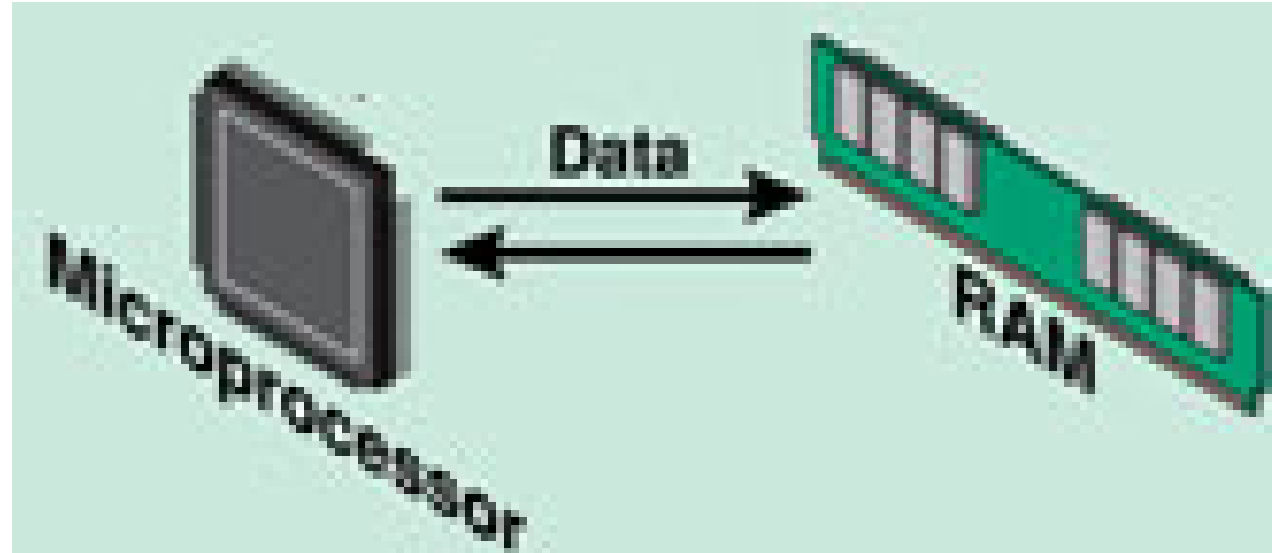
ADRES YOLU

- İşlemcinin haberleşmeyi planladığı hafıza yerini veya I/O cihazlarını tanımlamada kullanılır. Hedef ve kaynak verinin adresini taşır. 16 bitlik adres yolu 64 KB'lık hafıza sağlar.



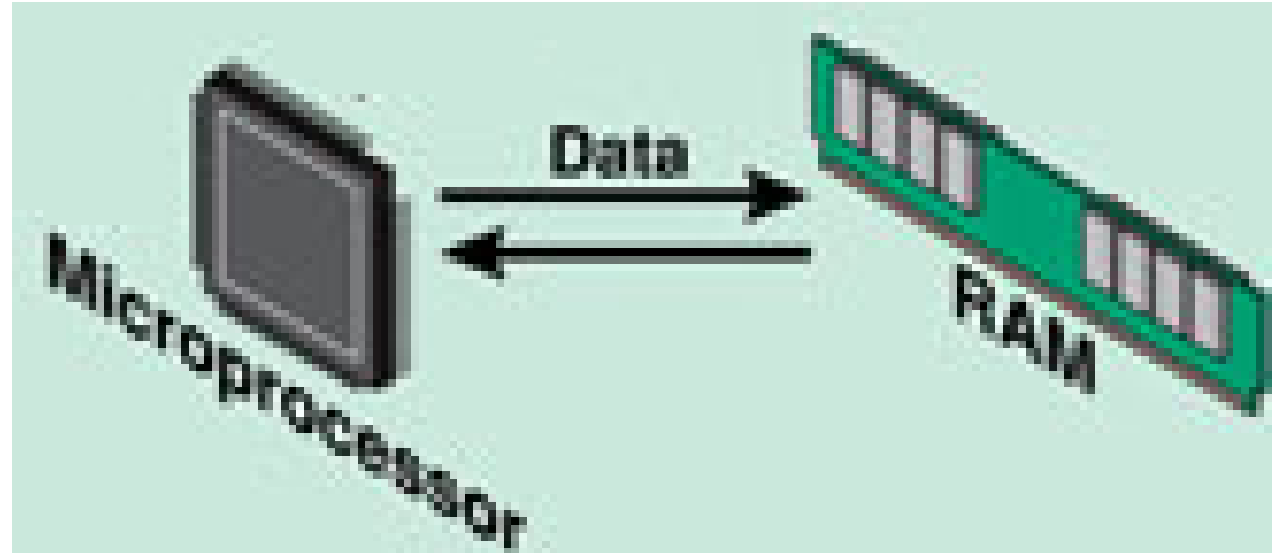
DATA YOLU

- Mikroişlemci tarafından bellek veya çıkış birimlerine veri göndermekte ya da bellekten veya giriş birimlerinden veri almakta kullanılır.
- Bu veri komut veya data olabilir. Veri yolu genişliği (bağlantı sayısı) bilgisayar performansını çok etkiler.
- Intel mikroişlemcilerinde veri yolu genişliği 8 bitten 64 bite kadar değişmektedir.



DATA YOLU

- Mikroişlemci tarafından bellek veya çıkış birimlerine veri göndermekte ya da bellekten veya giriş birimlerinden veri almakta kullanılır.
- Bu veri komut veya data olabilir. Veri yolu genişliği (bağlantı sayısı) bilgisayar performansını çok etkiler.
- Intel mikroişlemcilerinde veri yolu genişliği 8 bitten 64 bite kadar değişmektedir.

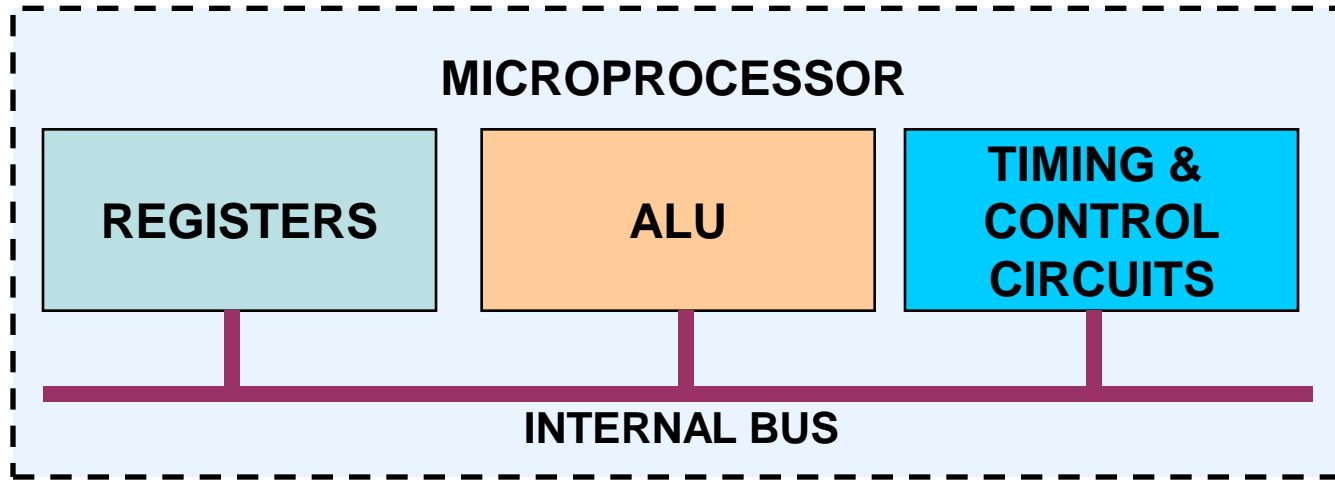


KONTROL YOLU

➤ Yol üzerindeki bir adresin hafıza adresi mi, I/O cihazının adresi mi olduğunu kontrol yolu tespit eder. Burada, hafıza ve I/O cihazı yazma ve okuma sinyallerinden birini aktif eder. Bu 4 kontrol yolu sinyali:

- Memory Read,
- Memory write,
- I/O read,
- I/O write'tır.

MİKROİŞLEMCI İÇ YAPISI



MİKROİŞLEMCI İÇ YAPISI

REGISTERS (YAZMAÇ): Mikroişlemcinin içindeki ikili tabandaki sayıları tutmaya yarayan geçici hafızadır, bu sayılar program kodu yada veri olabilir.

ARITHMETIC LOGIC UNIT (ALU, Aritmetik ve Mantık Birimi):

Bu kısım aritmetik ve mantık işlemlerini yapan hesaplama birimidir.

TIMING & CONTROL CIRCUITS (Zamanlama ve Kontrol

Devreleri) : Mikroişlemci tabanlı bir sistemin tüm elemanlarının senkronizasyon içinde çalışmasını sağlayan devrelerdir.

MİKROİŞLEMCİNİN ÇALIŞMA ŞEKLİ

Makine Dili (Machine Language): Mikroişlemcilerin direkt olarak anlayabildiği kod sistemine Makine Dili denir. Makine kodları sadece 1 ler ve 0 lardan oluşur. Her mikroişlemcinin kendi Makine kodu vardır.

Çevirme Dili(Assembly Dili): Makine dillerinin, insanların anlayabildiği haline çevirme dili denir. Hem 0 ve birlerden hem de bazı kelimelerden oluşur.

MİKROİŞLEMCİNİN ÇALIŞMA ŞEKLİ

Örneğin, Bir Intel Mikroişlemcili bilgisayar Bu kodu anlar,
Intel Mimarisinde yazılmış Makine Kodu:

1011 0000 0110 0001

Bu kodun kullanıcıların anlaya bileceği daha basit gösterimi ise
Çevirme dilinde yazılmış Çevirme Kodu:

MOV AX, 61H

MİKROİŞLEMCİNİN ANA ÇALIŞMA DÖNGÜLERİ



Bellek (RAM)

4. Evre: Store

Sonuçlar hafızaya geri yazılır

3. Evre: Execute

Komutlar işlenir

İşlemci

İşlem Birimi (ALU)

Kontrol Birimi

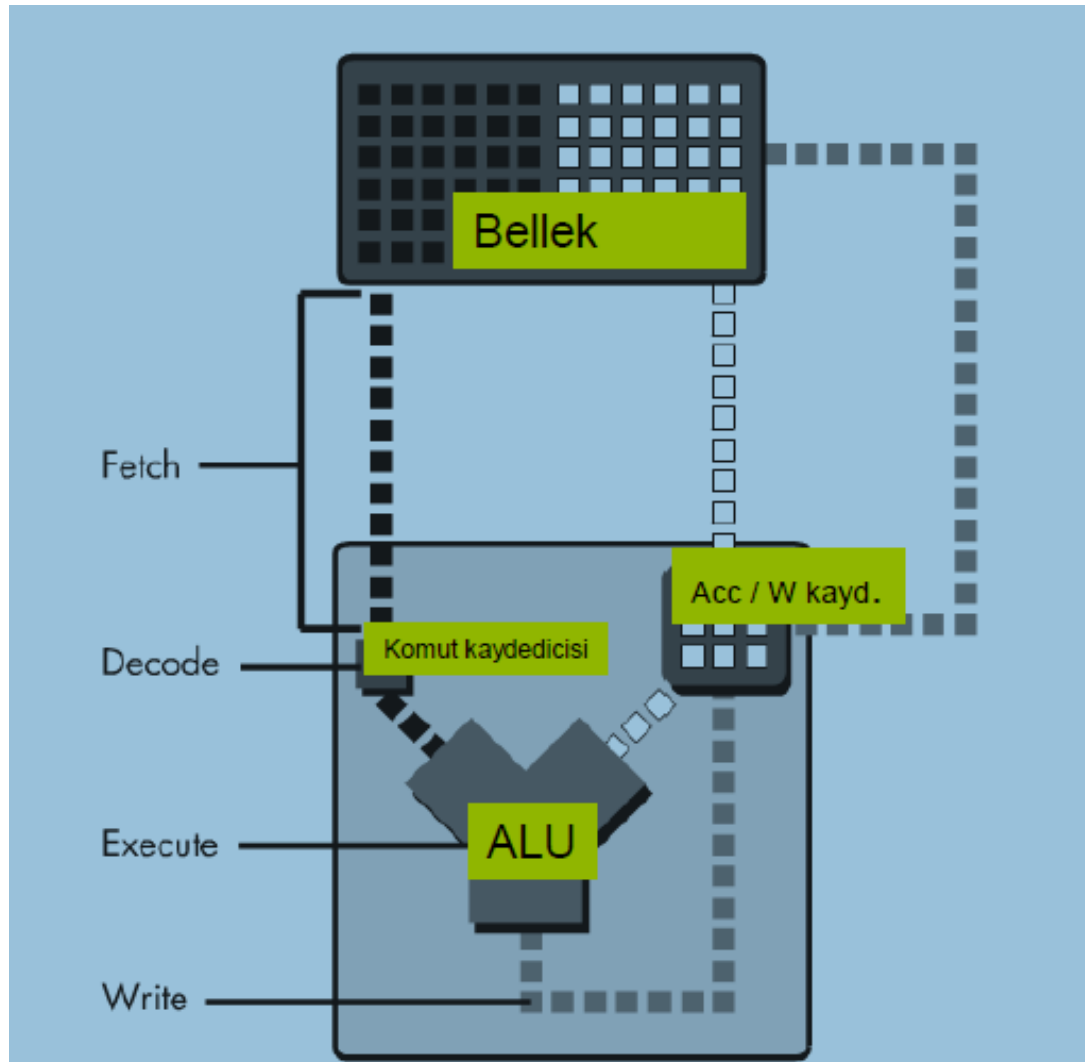
2. Evre: Decode

Alınan komutlar yorumlanır

1. Evre: Fetch

Veri yada program komutları hafızadan alınır

MİKROİŞLEMCİNİN ANA ÇALIŞMA DÖNGÜLERİ



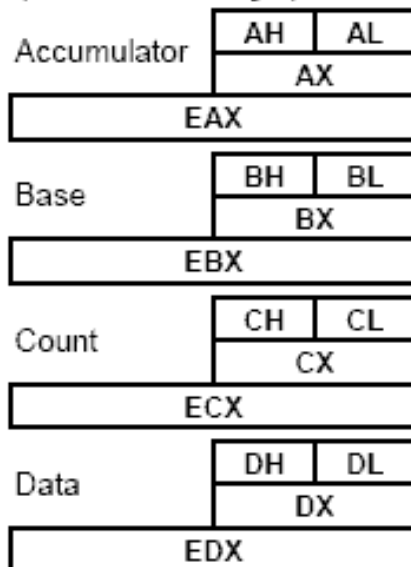
PROGRAMLAR MİKROİŞLEMCİDE NASIL ÇALIŞIR?

- Programlar, çok basit bir şekilde sıralanmış komutlardır
- CPU'lar temelde makine dili ile yazılmış komutları işleyebilirler
- Üst seviye diller tarafından yazılan programlar derleyiciler ile makine diline, yani işlemcinin tanıyacağı komutlar dizisi haline getirilirler
- İşletim sistemleri sınırlı kaynakları zaman içerisinde dağıtarak CPU'nun belirli bir düzen içinde bu programları çalıştırmasını sağlar

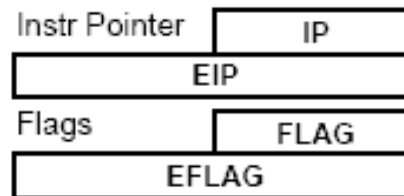
KAYDEDİCİLER

x86 Registers (x86 kaydedicileri)

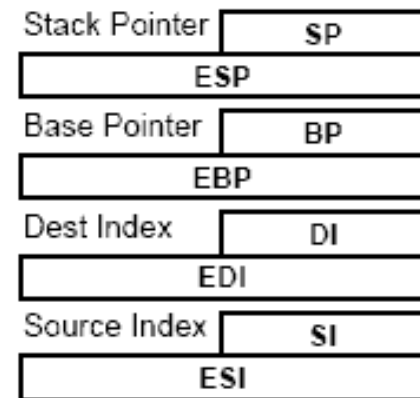
General Purpose (Genel Amaçlı)



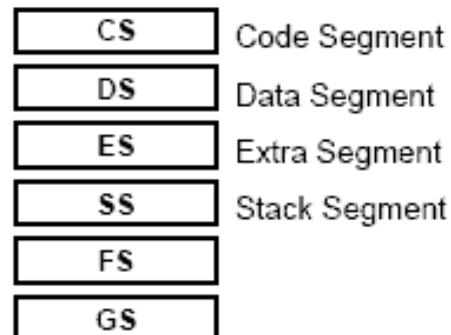
Special Registers (Özel kaydediciler)



Index Registers (indeks kaydedicileri)



Segment Registers (Segment kaydedicileri)



Accumulator: Akümülatör

Base: Taban

Count: Sayıcı

Instr Pointer : Komut işaretçisi

Flags: Bayraklar

Stack : Yığın

Dest: Hedef

Source: Kaynak

Code: Kod

Data : Veri

KAYDEDİCİLER

Basic Program Execution Registers

Sixteen 64-bit
Registers

General-Purpose Registers

Six 16-bit
Registers

Segment Registers

64-bits

RFLAGS Register

64-bits

RIP (Instruction Pointer Register)

FPU Registers

Eight 80-bit
Registers

Floating-Point
Data Registers

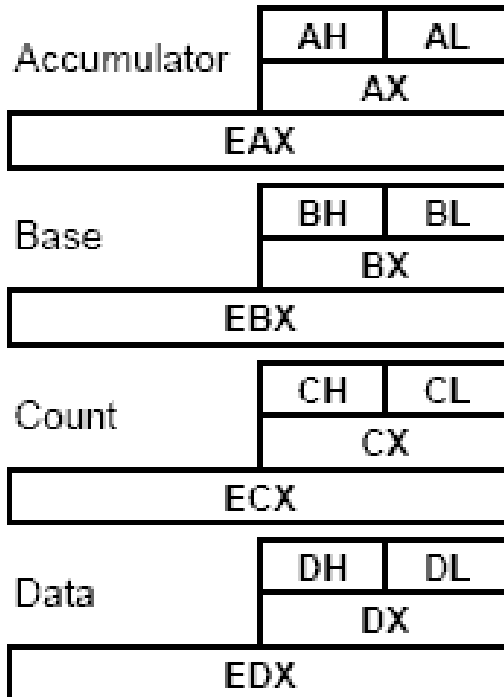
Address Space

$2^{64} - 1$

0

PROGRAMLAR MİKROİŞLEMCİDE NASIL ÇALIŞIR?

General Purpose (Genel Amaçlı)

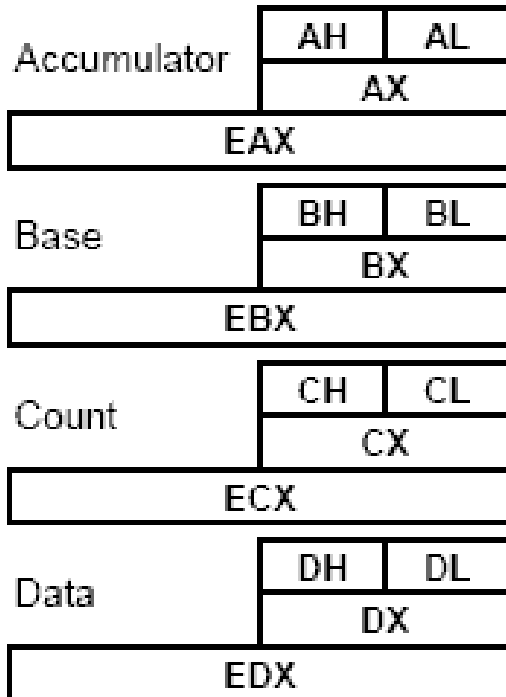


Accumulator (EAX, AX, AH, AL):

- Çok genel kullanım alanına sahip, değişik amaçlarla kullanılır.
- Bu kaydedici birçok giriş/çıkış işleminde ve aritmetik işlemlerde (çarpma, bölme ve taşıma gibi) kullanılır.
- Ayrıca bazı komutlar işlenmeden önce accumulator' den parametre alır veya bu kaydediciye işlemin sonucunu kaydeder.

GENEL AMAÇLI KAYDEDİCİLER

General Purpose (Genel Amaçlı)

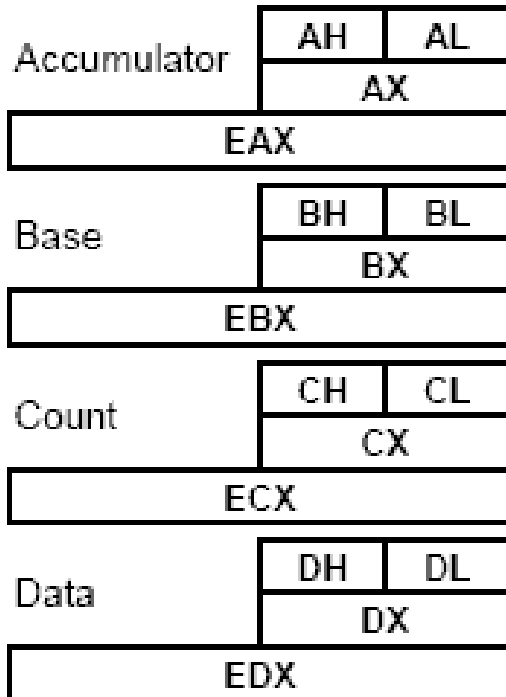


Base (EBX, BX, BH, BL):

- Accumulator gibi genel amaçlı ve hafıza erişiminde indexleri göstermede kullanılır. Bir başka kullanım alanı da hesaplamalardır

GENEL AMAÇLI KAYDEDİCİLER

General Purpose (Genel Amaçlı)



Counter (ECX, CX, CH, CL):

- Özel amaçlar ve hesaplamalarda kullanılacağı gibi genellikle bu kaydedici, sayıcı olarak kullanılır
- Örneğin, loop komutu CX kaydedicisini otomatik olarak değiştirir.

GENEL AMAÇLI KAYDEDİCİLER

General Purpose (Genel Amaçlı)

Accumulator	AH	AL
	AX	
EAX		
Base	BH	BL
	BX	
EBX		
Count	CH	CL
	CX	
ECX		
Data	DH	DL
	DX	
EDX		

Data (EDX, DX, DH, DL):

- Bazı giriş/çıkış komutlarında bu kaydedicinin kullanılması gerekir, ayrıca çarpma ve bölme işlemlerinde accumulator ile birlikte büyük sonuçlar bu kaydedicide tutulur.

SEGMENT KAYDEDİCİLER

Segment Registers (Segment kaydedicileri)

CS	Code Segment
DS	Data Segment
ES	Extra Segment
SS	Stack Segment
FS	
GS	

Code Segment Kaydedicisi (CS):

- DOS işletim sisteminde programları oluşturan kodlar code segment'e yüklenir.
- CS kaydedicisi ise IP kaydedicisi ile birlikte programın çalışma sürecinde, programı oluşturan kodların adreslerini gösterirler.

SEGMENT KAYDEDİCİLER

Segment Registers (Segment kaydedicileri)

CS	Code Segment
DS	Data Segment
ES	Extra Segment
SS	Stack Segment
FS	
GS	

Data Segment Kaydedicisi (DS):

- .exe türündeki bir programda kullanılacak olan veriler data segment denilen hafıza bölümünde tutulur.
- DS kaydedicisi ise bu bölgedeki verilerin konumlarını gösterir.

SEGMENT KAYDEDİCİLER

Segment Registers
(Segment kaydedicileri)

CS	Code Segment
DS	Data Segment
ES	Extra Segment
SS	Stack Segment
FS	
GS	

Extra Segment Kaydedicisi (ES):

■ Bazı string işlemlerinde DI kaydedicisi ile birlikte karakterlerin bulunduğu hafıza adreslerini tutarlar.

SEGMENT KAYDEDİCİLER

Segment Registers (Segment kaydedicileri)

CS	Code Segment
DS	Data Segment
ES	Extra Segment
SS	Stack Segment
FS	
GS	

Stack Segment Kaydedicisi (SS):

- Tüm programlar stack segment denilen bir hafıza alanını geçici depolama alanı olarak kullanmak zorundadırlar (örneğin dallanma işlemlerinde).
- SS kaydedicisi ise SP kaydedicisi ile birlikte bu verilerin adreslerini referans eder.

SEGMENT KAYDEDİCİLER

Segment Registers (Segment kaydedicileri)

CS	Code Segment
DS	Data Segment
ES	Extra Segment
SS	Stack Segment
FS	
GS	

FS ve GS Kaydedicileri:

- 80386 ve sonrası CPU'larda bulunurlar ve diğer segment kaydedicilerinin yetersiz kaldığı durumlarda kullanılırlar.

ÖZEL KAYDEDİCİLER

Special Registers (Özel kaydediciler)

Instr Pointer	IP
EIP	
Flags	FLAG
EFLAG	

IP ve EIP kaydedicileri:

- IP 16 bitlik DOS programlarının EIP ise 32 bitlik programların işlenmesi sürecinde, işlenecek olan bir sonraki komutun offset adresini gösterir.

ÖZEL KAYDEDİCİLER

Special Registers (Özel kaydediciler)

Instr Pointer	IP
EIP	
Flags	FLAG
EFLAG	

FLAG ve EFLAG kaydedicileri:

■ Flag kaydedicisi 16 Eflag kaydedicisi ise 32 bitten oluşur. mikroişlemciler matematiksel işlem yapar, bu kaydedicilerde her işlemde sonra o işleme özel sonuçları gösterirler. İşlemci durum kaydedicisi olarakta bilinen bu kaydediciler sonucun sıfır, pozitif veya negatif olduğunu veya işlemin sonucunda elde üretilip üretilmediği gibi birçok önemli veriyi bitsel olarak programcıya bildirirler.

INDEX KAYDEDİCİLER

Index Registers
(indeks kaydedicileri)

Stack Pointer	SP
ESP	
Base Pointer	BP
EBP	
Dest Index	DI
EDI	
Source Index	SI
ESI	

- Bu kaydedicilerin E ile başlayanlar 32 bitlik programlarda, diğerleride 16 bitlik programlarda kullanılır.
- Hepsi de verilerin offset adreslerini tutmada kullanılır. SP ve BP, SS kaydedicisi ile birlikte SI ve DI, DS ve ES kaydedicileri ile birlikte hafıza adreslerine erişmek için kullanılır.