

# MİKROİŞLEMCİLER (BLM202)

## HAFTA - 3

Dr. Bilgin YAZLIK, RTTP, PMP



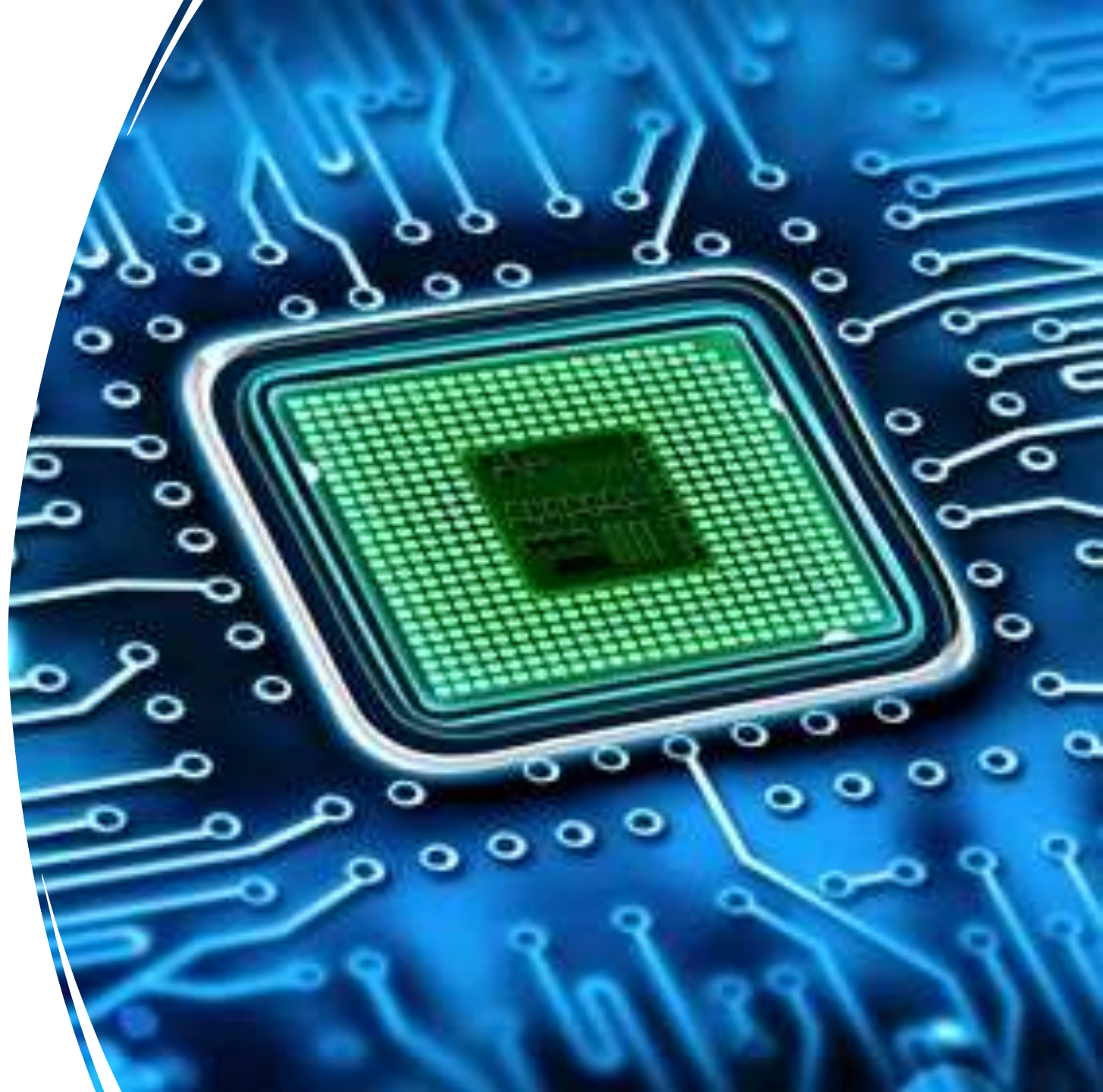
**BİLGİSAYAR MÜHENDİSLİĞİ**



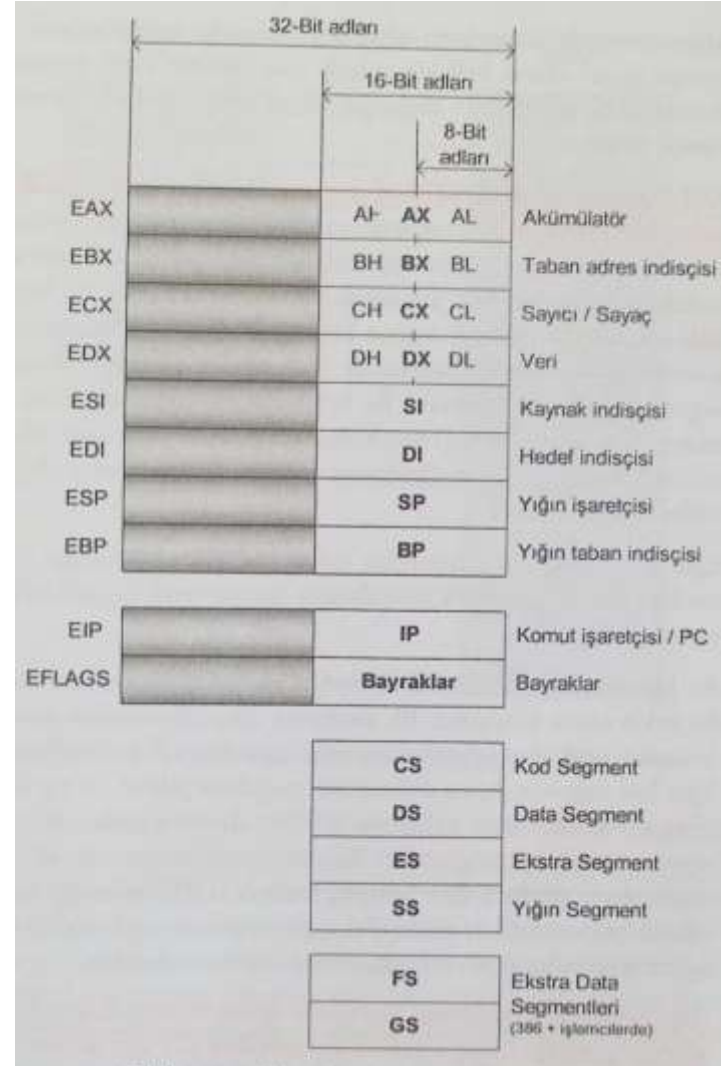
# 3. HAFTA

---

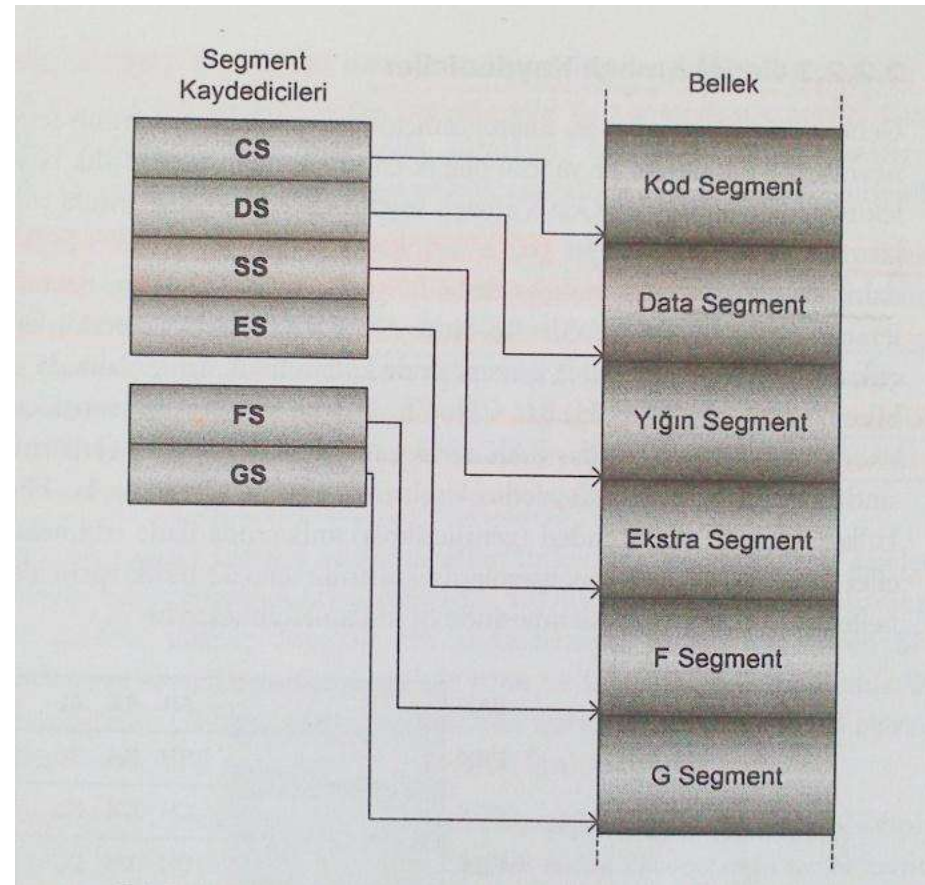
- Mikroişlemcinin İç Yapısı:
  - Kaydediciler
  - Bayraklar



# SEGMENT KAYDEDİCİLER



# SEGMENT KAYDEDİCİLER





# GENEL AMAÇLI KAYDEDİCİLER

- Genel amaçlı kaydediciler, mikroişlemcide program komutlarının icrası sırasında verinin manevrasında kullanılan ve yapısal olarak en küçük bölümü 8 bitlik bellek hücresine benzeyen elektronik elemanlardır. Genel amaçlı kaydediciler kendi aralarında yaptıkları işe göre iki gruba ayrılırlar. Birinci grupta çok amaçlı kaydedicilere EAX, EBX, ECX, EDX, ESI, EDI ve EBP dahilken, ikinci özel amaçlı kaydediciler grubuna ESP, EIP ve Bayrak kaydedicisi girer. 386'ya kadar bu kaydediciler 16 bitlik AX, BX, CX ve DX olarak işlem görmüşlerdir. Daha küçük 8 bitlik verilerin (bayt) işlenmesinde kullanılmak üzere daha da ufak parçalarla tanımlanabilmektedir. AH, AL, BH, BL, CH, CL, DH ve DL gibi. AX serisi kaydediciler 16 bitlik verilerin saklanması EAX serisi kaydediciler 32 bitlik verilerin saklanması kullanılmaktadır. Kaydedici kısaltmasındaki X'in manası H (High) ve L (Low)'un birlikte kullanımı E ise Extenden (genişletilmiş) manasına gelmektedir.

# GENEL AMAÇLI KAYDEDİCİLER

EAX		AH	AX	AL
EBX		BH	BX	BL
ECX		CH	CX	CL
EDX		DH	DX	DL
ESI		SI		
EDI		DI		
ESP		SP		
EBP		BP		
EIP		IP		
EFLAGS		Bayraklar		

# GENEL AMAÇLI KAYDEDİCİLER

- **AX kaydedicisi:** Akümülatör AX koduyla tanımlanır ve verilerin ilk ele alınmasında başrol oynadığından **baş kaydedici** olarak düşünülebilir. 8, 16 ve 32 bitlik verilerle çarpma, bölme bazı I/O işlemlerinde ve bazı harf dizi işlemlerinde etkin bir biçimde kullanılmaktadır.
- **BX kaydedicisi:** **Taban adres kaydedicisi** olarak bilinen ve BX koduyla tanımlanan kaydedici, bellekteki veri gruplarının ofsetinin tutulmasında bir indisçi gibi davranır. Ayrıca hesaplamalarda ve 32 bitlik işlemcilerde bellekteki verinin adreslenmesinde de kullanılmaktadır.
- **CX Kaydedicisi:** **Sayaç kaydedicisi** olarak bilinen CX, string işlemlerinde bir sayaç elemanı veya döngü işlemlerinde tekrarlama sayıcısı gibi işlevleri yerine getirir.
- **DX kaydedicisi:** **Data kaydedicisi** diye tanımlanan DX kaydedicisi, genellikle akümülatöre yardımcı olan bütün işlemlerde bir tampon gibi davranan kaydedicidir.

# GENEL AMAÇLI KAYDEDİCİLER

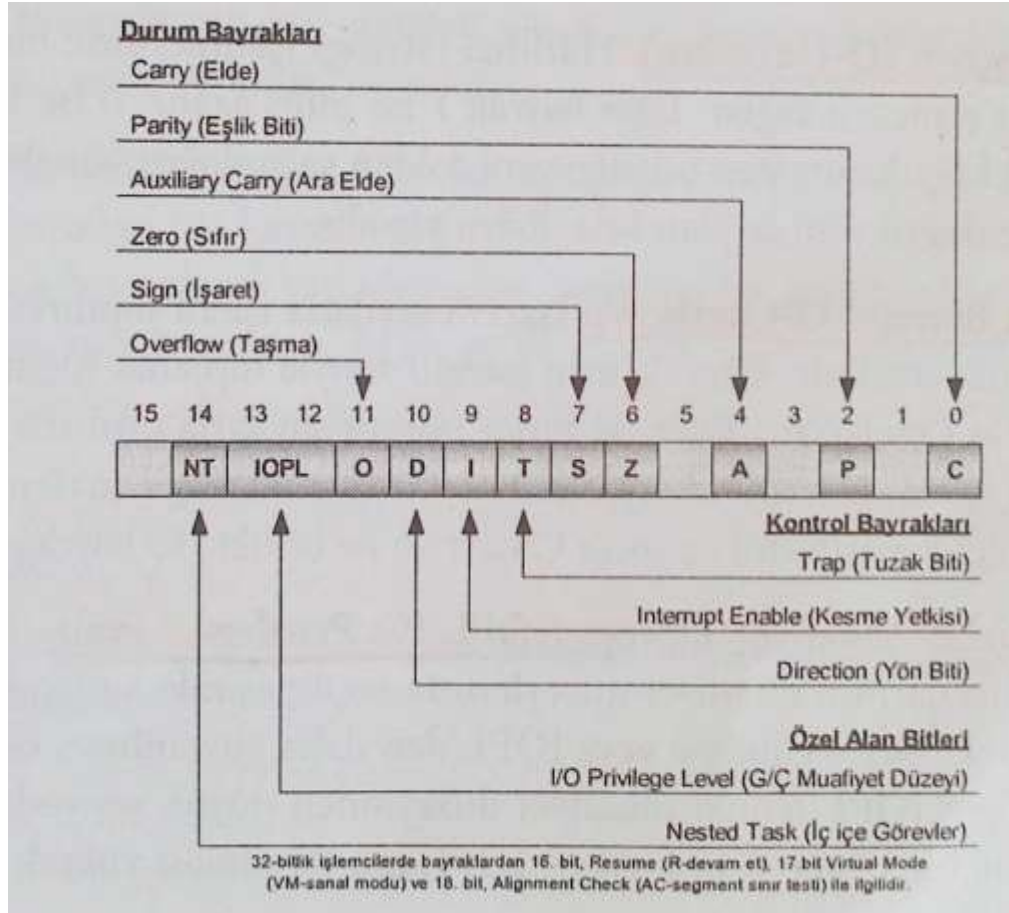
- İşaretçi ve İndis Kaydedicileri: Mikroişlemcili sistemlerde bellekteki ara adresleri gösteren kaydedicilere işaretçi adı verilir.
- Bayrak kaydedicisi: Bayrak kaydedicisi bir işleminde sonucun ne olduğunu kaydedici bitlerine yansıtan bir bellek hücresini oluşturur. Bu kaydediciye bayrak denmesinin sebebi, karar vermeye dayalı komutların yürütülmesinde sonuca göre daha sonra ne yapılacağını bit değişimiyle bu kaydedicinin 1 bitlik hücrelerine yansıtmasıdır. Kaydedici bitlerindeki mantıksal 1 bayrak kalktı, 0 bayrak indi demektir. Karşılaştırma ve aritmetik komutların çoğu bayraklara etki eder.



# BAYRAKLAR (FLAGS)

- **Elde Bayrağı (C-Carry):** Eğer toplama sonucunda elde, çıkarma sonucunda borç ortaya çıkıyorsa  $C=1$  aksi takdirde 0 olur. Aynı zamanda C bayrağı kaydırma ve yönlendirme işlemleri sonucunda kaydedicinin MSB bitinden veya LSB bitinden düşen verileri üzerinde tutar ve karşılaştırma işlemlerinin sonucunu yansıtır. Ayrıca C bayrağı çarpma işlemi için sonuç göstericisi gibi hareket eder.
- **Eşlik biti (P-Parity):** İşlemin sonucunda kaydedicideki mantıksal birlerin sayısı çift ise  $P=1$  aksi halde  $P=0$  olur. Eşlik biti genelde veri iletişiminde karşılıklı verilerin güvenli iletilip iletilmediğinin kontrolünde kullanılır.
- **Yardımcı Elde Bayrağı: (AC-Auxiliary Carry):** Elde bayrağı ile aynı işlemi görür fakat sadece 3. bitten bir fazlalık ortaya çıkarsa bu bayrak 1 aksi durumda 0 olur. AC bayrağı paketlenmiş ondalık verilerin işlenmesinde çok kullanışlıdır.

# BAYRAKLAR



- (1) "Carry flag" C that represents the carry and borrow of the digit
- (2) "Zero flag" Z that the calculation result represents 0

```

[A]  0011 1110
[B]  + 1110 0000
-----
[A]  10001 1110
    
```

Stored in A register

C = 1 Carry flag

```

  29H
+ 4CH
-----
 75H
    
```

$$9 + C(12) = 16(\text{carry}) + 5$$

Auxiliary Carry Flag in Hexadecimal Representation

Let us consider the same example in binary representation.

29H = 0010 1001

+4CH = 0100 1100

75H = 0111 0101

^ here there is carry generated and forwarded to next nibble, so the auxiliary carry flag is set to one.

# BAYRAKLAR

- **Sıfır Bayrağı (Z-Zero):** İşlem sonunda sonuç 0 ise  $Z=1$  aksi halde  $Z=0$  olur. Mesela bu işlem sonunda AX kaydedicisindeki değer 0000 ise sıfır bayrağı 1 olur diğer durumlarda bayrak 0 kalır.
- **İşaret bayrağı (S-Sign):** İşaretili sayılarla yapılan işlemlerde bu bayrak anlam ifade etmektedir. Eğer aritmetik mantık, kaydırma ve yönlendirme işlemleri negatif sonuç üretiyorsa  $S=1$  aksi halde  $S=0$  olur. Diğer bir deyimle S bayrağı sonucun 8 bit veya 16 bit olmasına bakılmaksızın MSB bitini yansıtır.
- **Tuzak Bayrağı (T-Trap):** Hata ayıklama işlemlerinde komutların adım adım işlenmesi maksadıyla kullanılır. Bayrak 1 yapıldığında Debug işlemi yapmak için komutlar tek tek çalıştırılır.
- **Kesme Yetkilendirme bayrağı (I-Interrupt Enable):** Sisteme bağlı harici cihazlardan gelen kesme taleplerine izin verir. I bayrağının 0 olması kesme isteklerine cevap verilmemesini sağlar. Ancak 1=1 olduğunda tekrar istekler göz önüne alınır.

# BAYRAKLAR

- **Yön Bayrağı (D-Direction):** String işlemlerinde indis kaydedicisinin ileri yada geri hareket etmesini sağlar. Eğer bayrak 1 ise indis azalır, 0 ise indis değeri artar. Eğer D=0 ise, işlemci **küçük adresten büyüğe** yani soldan sağa doğru yönelir. Eğer D=1 ise, **büyük adresten küçüğe** doğru yani sağdan sola doğru yönelir. Eğer D=1 ise büyük adresten küçük adrese yani sağdan sola doğru yönelir.
- **Taşma bayrağı(O-Overflow):** İşaretili sayılarla işlem yapılırken bir hatanın ortaya çıkması durumunda gözükür. Eğer iki aynı sayıyla toplama işlemi yapılıyor ve sonuç farklı işaretili çıkıyorsa 0=1 olur. **Eğer matematik bir işlem sonucunda sonuç kaydedici kapasitesini aşıyorsa C bayrağı ile birlikte O bayrağı da 1 olur.**
- **Giriş/Çıkış Muafiyet düzeyi (IOPL-IO Privilege Level):** Korumalı mod operasyonlarında G/Ç cihazlarının muafiyet düzeylerinin seçilmesinde kullanılır. Eğer o andaki muafiyet düzeyi yüksek seçilmişse veya IOPL'den daha güvenilirse, G/Ç herhangi bir engellemesiz çalışır.

# BAYRAKLAR

- **İççe Geçmiş Görevler (NT-Nested Task):** Korumalı mod operasyonlarında o andaki görevin başka bir görevle iç içe girmesi işlemidir. Görev başka bir görevle yazılım tarafından iç içe girdirildiğinde bu bayrak 1 olur.
- **İşleme devam (R-Resume):** Hata ayıklama işlemlerinde (Debug), bir sonraki işlenecek komuta devam edilmesinin kontrolünde kullanılır.
- **Sanal Mod (VM-Virtual Mode):** Korumalı mod sisteminde sanal mod işleminin seçilmesinde kullanılır. Sanal mod, DOS sisteminde belleğin birkaç parçaya bölümlenmesini sağlar.
- **Segment Sınır Tespiti (AC-Alignment Check):** Eğer word veya doubleword tanımlamaları kendilerine uygun adres sınırlarında değilse bu bayrak 1'e kurulur. Bu bayrak sadece 486SX işlemcide kullanılmaktadır.



# BELLEK ADRESLEMESİ

- Mikroişlemciye dayalı sistemlerde adres uzayı **fiziksel veya mantıksal bellek** tanımlamasından birisiyle ilişkilendirilir.
- Çoğu durumlarda mantıksal bellek yapısı fiziksel bellek yapısından farklıdır. **Fiziksel bellek**, bellek sisteminin gerçek donanım yapısıyken mantıksal bellek, bunun programcıya gözüken tarafıdır.

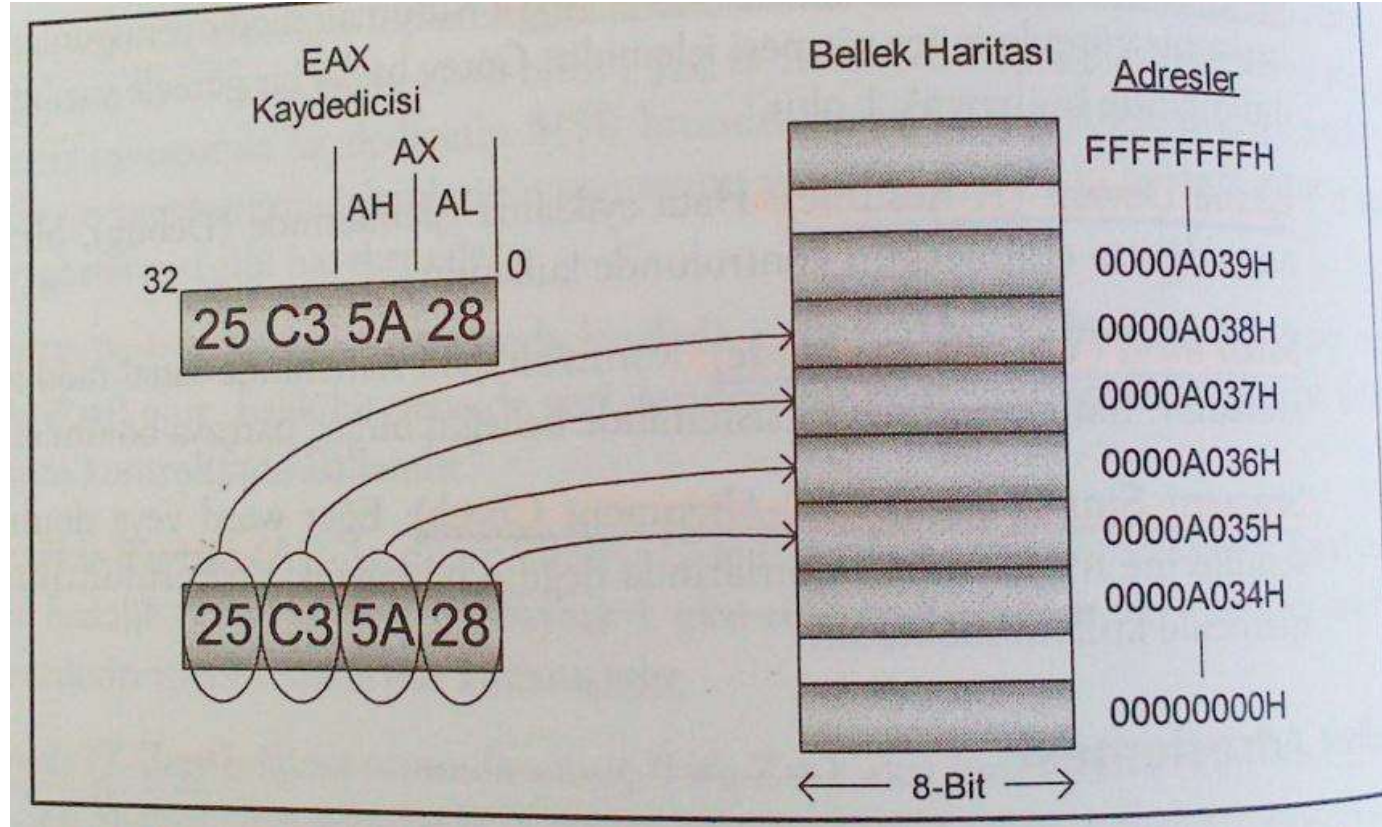
# MANTIKSAL BELLEK TANIMLAMA

- Mantıksal bellek tanımlamasında bütün adresler byte olarak numaralandırılır ve programcı buna göre programını yazar ve uygular. 16 adres hatlı işlemcilerin adres numaraları 0000H ile başlar ve FFFFH ile sona erer. Burada tanımlanan adres uzayı 64 KB'dır. 32 adres hattına sahip işlemcilerin adres numaraları 00000000H ile başlar ve FFFFFFFFH ile sona erer. Bu işlemcileri kullanan sistemin adresleme kapasitesi böylece 4 GB olur.

- Programcı mantıksal bellek yapısına göre bir byte'lık veriyi doğrudan bir adres göstererek oraya depolar veya oradan bir kaydediciye yükleyebilir. Fakat bir word'lük bir verinin üzerinde tanımlanması, bellekte iki ardışık adrese ulaşacak demektir. Bu durumda verinin az değerlikli kısmı küçük nolu adrese, çok değerlikli kısmı büyük nolu adrese yerleştirilir. Bu işleme ters bayt sıralaması denir.

**A byte is eight bits, a word is 2 bytes (16 bits), a doubleword is 4 bytes (32 bits), and a quadword is 8 bytes (64 bits)**

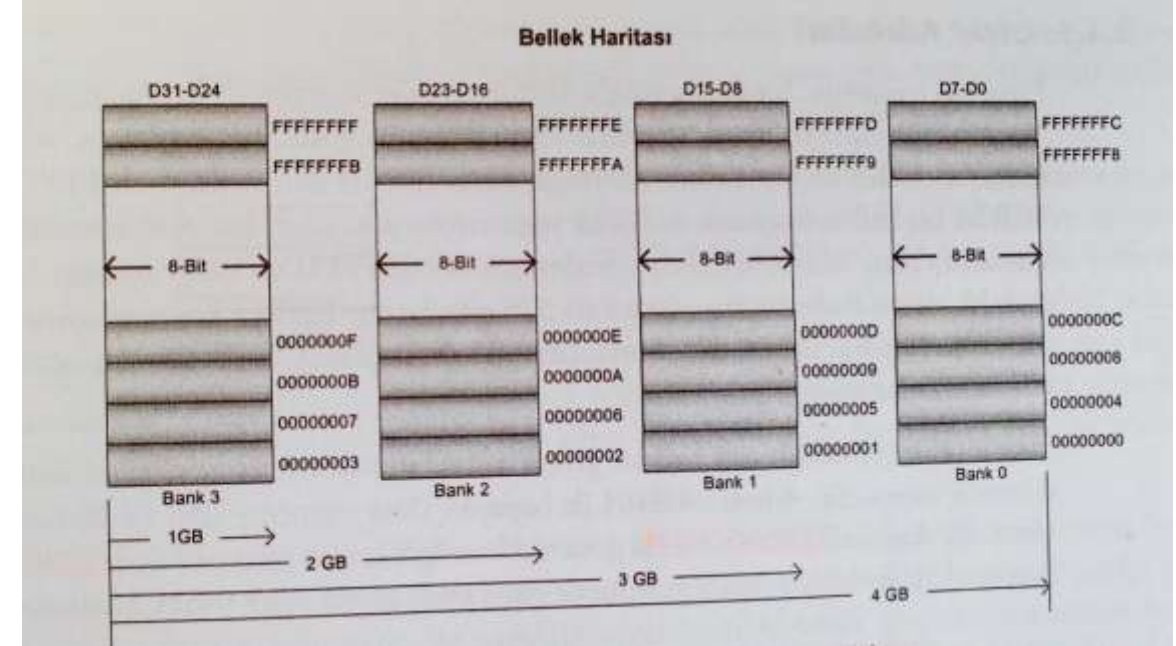
# MANTIKSAL BELLEK TANIMLAMA



32 bitlik EAX kaydedicisindeki **25C35A28H** verisi bellekte **0000A035H** adresine saklanmak istenirse, **dört ardışık bellek adresine** erişim yapılması gereklidir. Bu işlem **otomatik olarak işlemci tarafından** yapılır programcı yalnızca **başlangıç adresini** belirtir.

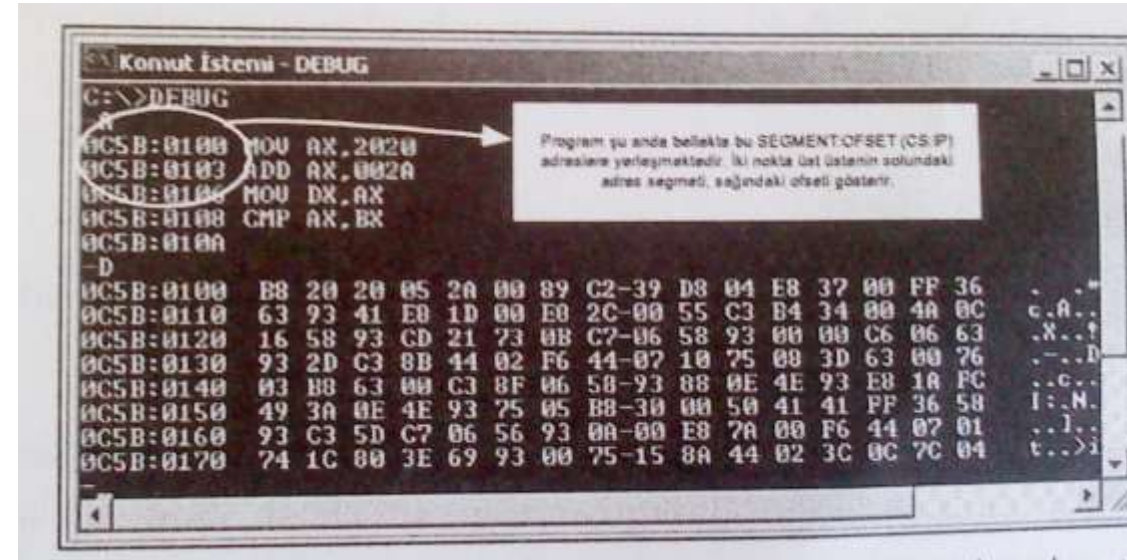
# FİZİKSEL BELLEK TANIMLAMA

- Belleklerin fiziksel tanımlanması donanımsal bir yaklaşımdır ve işlemci mimarisine bağlıdır. Günümüz işlemcilerinin bellek organizasyonu 8'er bitlik banketler halinde (sıralar) yapılır. 16 bitlik bellek düzeninde iki adet 8 bitlik banketle adresleme bayt yada word olarak yapılırken, 32 bitlik bellek düzeninde 4 adet 8 bitlik banketle adreslemeler bayt, word veya doubleword olarak yapılır.



# SEGMENT SINIRLARI

Segmentler, adreslerin eşit bir şekilde ondalık olarak 16 ile bölünebilen ve adına paragraf sınırı denilen adresle başlar. Segment adres kaydedicileri daima 0H ile başlar. Mesela herhangi bir segment kaydedicisi içeriği 255AH ise, 64 KB'lık bellek alanında bu segmentin başlangıç adresi 255 A0H olacaktır.

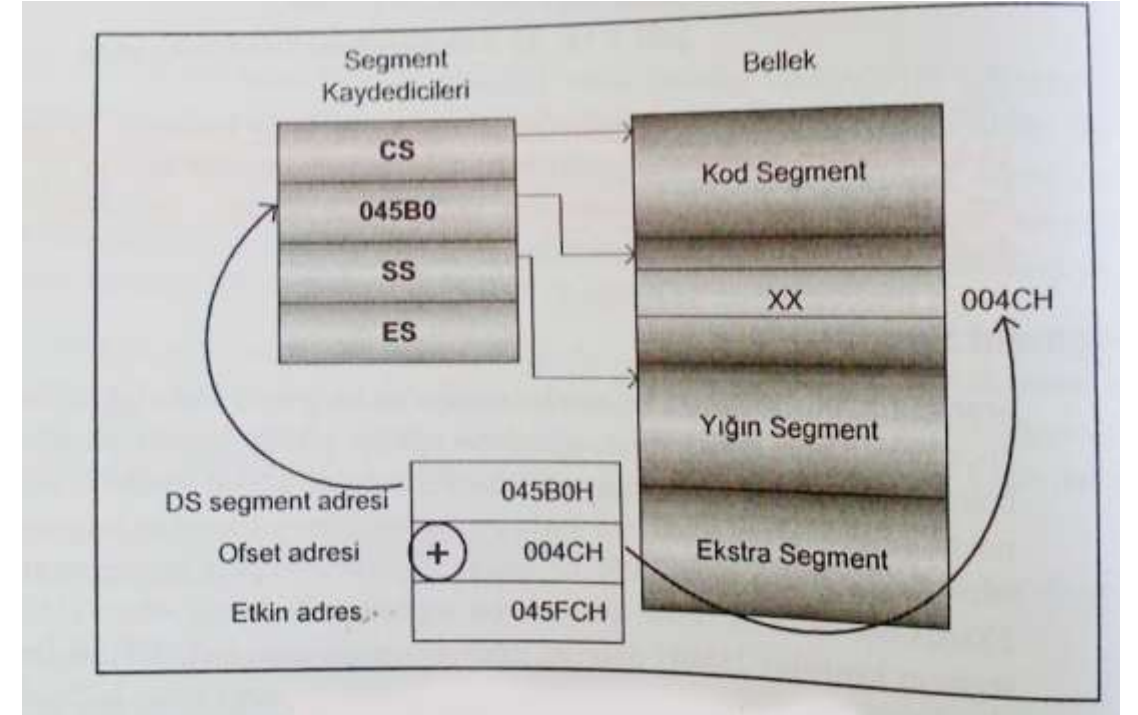




# OFSET ADRESLERİ

Program içerisinde bir segmentteki bütün adresler o segment adresine göre görecelidir. Verinin bulunduğu adres, segment kaydedicisinin gösterdiği başlangıç adresinden uzaklığı kadardır. 64 KB'lık bir bellek uzayında 0000H'dan başlayarak FFFFH'e kadar gider. Şekilde bellekte gerçek adresi bulmak için segment adres ile ofset adresi belli bir düzende birleştirilir.

Adresi 045B0H ile başlayan data segmentindeki bir komutun adresi 004CH olsun. Bu durumda DS:045B0H'yi gösterecek ve ilgili komut ofset olarak da 004CH'yi gösterecektir. Bu durumda komutun data segmentinde işaret ettiği gerçek adres 045FCH olacaktır.



Hex value:

$$045B0 + 004C = \mathbf{45FC}$$

# Kaynaklar

---

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

