

# ALT SEVİYE PROGRAMLAMA

Hafta 1

Dr. Öğr. Üyesi Erkan USLU

# MİKROİŞLEMCİLERİN TARİHSEL GELİŞİMİ

# Mikroişlemci Çağı

- İlk mikroişlemci Intel firmasının geliştirdiği 4004'tür
  - 4 bitlik mikroişlemci
  - Adresleme kapasitesi: 4096 x 4 bit
  - Komut seti 45 komuttan oluşuyor
  - 30 gram ağırlığında
  - Saniyede 50000 işlem (30 ton ENIAC saniyede 100000 işlem)
  - Oyun ve küçük kontrol sistemlerinde kullanıldı
  - RTL (direnç –transistör lojisi ile tasarlanmış)
- Sonrasında daha yüksek frekanslı 4040 mikroişlemci geliştirildi

# Mikroişlemci Çağı

- 1971'de Intel 8008 mikroişlemciyi tanıttı
  - 8-bitlik bir mikroişlemci
  - 16KB adresleme kapasitesi
  - Toplamda 48 farklı komut yürütebiliyordu
- Mikroişlemcilerin daha karmaşık sistemlerde kullanımı mümkün oldu

# Mikroişlemci Çağı

- 1973 yılında Intel 8080 mikroişlemciyi tanıttı
- İlk modern 8 bitlik mikroişlemci olarak kabul edilir
- 8080
  - 64KB adresleme kapasitesi
  - 8008'e göre yaklaşık 10 kat daha hızlı
  - TTL (transistör- transistör lojiği ile tasarlamış)

# Mikroişlemci Çağı

- 8080'in sunumundan 6 ay sonra Motorola MC6800 mikroişlemciyi sundu
- Diğer firmalar tarafından da 8 bitlik mikroişlemciler piyasaya sunuldu
- Fairchild – F8, MOS tech – 6502, National Semiconductors – IMP8, Zilog – Z8
- 1974'te MITS Altair 8800 sunuldu
  - 1975'te Bill Gates ve Paul Allen Altair 8800 için BASIC dilini geliştirdi

# Mikroişlemci Çağı

- 1977 yılında Intel 8085 mikroişlemciyi sundu
- Intel'in son 8 bitlik mikroişlemcisi
- Saniyede 769230 işlem
- Dahili saat üretici kullanımı
- Entegre komponent sayısında artış

# Modern Mikroişlemciler

- 1978 yılında 8086 ve bir yıl sonra 8088 mikroişlemciler tanıtıldı
- 16 bitlik mikroişlemciler
- Komut yürütme süresi 400 ns (saniyede 2,5 milyon işlem)
- Adresleme kapasitesi 1MB
- 4 veya 6 byte'lık komut kuyruğu mevcut (sıradaki birkaç komutun birlikte okunması)
- Çarpma bölme gibi komutların sunulması
- Varyasyonları ile 20000'i bulan komut sayısı



# Modern Mikroişlemciler

- 8086/8088 CISC (complex instruction set computers) mimarisindedir
- Yazmaç sayısında artış söz konusu
- 8086 ve 8088: 20 adet adres ucuna sahip
- 8086: 16 veri ucuna sahip
- 8088: 8 veri ucuna sahip

# Modern Mikroişlemciler

- 1983 yılında 80286 tanıtıldı
- 16MB adresleme kapasitesine sahip
- Komutlar 8086'ya benzer şekilde olmakla birlikte 16MB hafıza için komutlarda güncelleme var
- Saat frekansı 8MHz → saniyede 4 milyon işlem

# Modern Mikroişlemciler

- 1986 yılında 80386 sunuldu
- 32 bit adres yolu, 32 bit veri yolu
- 4GB adresleme kapasitesi

# Modern Mikroişlemciler

- 1989 yılında 80486
- 32 bit adres yolu, 32 bit veri yolu
- Cache kullanımı
- DX modelinde matematik işlemci ana işlemci ile birleştirilmiştir.
- Pentium, Celeron, Itanium, Core, Core 2, Core i

# SAYI SİSTEMLERİ VE DÖNÜŞÜM KURALLARI

# SAYI SİSTEMLERİ

- Mikroişlemciler ile çalışılırken sayı sistemlerinin
  - İkili taban
  - Onluk taban
  - Onaltılık taban

Ve bunlar arasındaki dönüşümlerin nasıl sağlandığının bilinmesini gerektirir

# Rakamlar

- Onluk tabanda kullanılan rakamlar 0-9 arasındadır
- Sekizli tabanda kullanılan basamaklar 0-7 arasındadır
- İkili tabanda kullanılan basamaklar 0,1'dir
- Onluk tabandan daha büyük tabanlarda eklenen her bir basamak alfabe A ve sonraki harfler ile gösterilir
  - Oniki tabanında kullanılan basamaklar 0-9,A,B'dir
- Bilgisayar sistemlerinde onluk, ikili ve onaltılı tabanlar yaygın olarak kullanılır

# Sayıların Gösterimi

- Sayı gösterimlerinde iki önemli kavram var
  - Basamak
  - Basamakların sayıdaki konumsal gösterimi
- Onluk tabanda yazılan 132 sayısı;
  - Soldan sağa 1,3,2 basamaklarından oluşmakta
  - 2 sayısı 1'ler basamağında ( $10^0$ )
  - 3 sayısı 10'lar basamağında ( $10^1$ )
  - 1 sayısı 100'ler basamağında ( $10^2$ )
  - $132 = 10^2 * 1 + 10^1 * 3 + 10^0 * 2$



# Sayıların Gösterimi

- İkili tabanda
  - $(11)_2 = 2^1 * 1 + 2^0 * 1 = 2 + 1 = 3$
- Sekizli tabanda
  - $(11)_8 = 8^1 * 1 + 8^0 * 1 = 8 + 1 = 9$
- Onluk tabanda
  - $(11)_{10} = 10^1 * 1 + 10^0 * 1 = 10 + 1 = 11$

# Sayıların Gösterimi

- Sayıların gösteriminde virgülden sonraki sayılar ise negatif üs ile ifade edilir
- Onluk tabanda
  - $(0.1)_{10} = 10^0 * 0 + 10^{-1} * 1 = 0 + 0.1 = 0.1$
- İkili tabanda
  - $(0.1)_2 = 2^0 * 0 + 2^{-1} * 1 = 0 + 0.5 = 0.5$

# Sayı Tabanları Arası Dönüşüm

- Onluk tabandan ikili tabana dönüşüm
  - $(10)_{10} = (?)_2$
  - $10/2 = 5$ , kalan 0
  - $5/2 = 2$ , kalan 1
  - $2/2 = 1$ , kalan 0
  - Kalan 1
  - $\rightarrow (1010)_2$

# Sayı Tabanları Arası Dönüşüm

- Onluk tabandan sekizli tabana dönüşüm
  - $(10)_{10} = (?)_8$
  - $10/8 = 1$ , kalan 2
  - Kalan 1
  - $\rightarrow (12)_8$

# Sayı Tabanları Arası Dönüşüm

- Onluk tabandan onaltılık tabana dönüşüm (onaltılık tabanda 0-9,A,B,C,D,E,F basamakları kullanılır)
  - $(109)_{10} = (?)_{16}$
  - $109/16 = 6$ , kalan D
  - Kalan 6
  - $\rightarrow (6D)_{16}$

# Sayı Tabanları Arası Dönüşüm

- Virgüllü sayılarda tabanlar arası dönüşüm
  - Sayı basamak değeri ile çarpılır
  - 0 elde edilene kadar işlem tekrarlanır
  - Bazı virgüllü sayılar için sonuçta 0 elde edilemeyebilir

# Sayı Tabanları Arası Dönüşüm

- $(0.625)_{10} = (?)_2$
- 0.625, tam kısım 0
- $0.625 * 2 = 1.25$ , tam kısım 1
- $0.25 * 2 = 0.5$ , tam kısım 0
- $0.5 * 2 = 1.0$ , tam kısım 1
- Virgülden sonra 0 değerine ulaşıldı
- $\rightarrow (0.101)_2$

# Sayı Tabanları Arası Dönüşüm

- $(0.2)_{10} = (?)_2$
- 0.2, tam kısım 0
- $0.2 * 2 = 0.4$ , tam kısım 0
- $0.4 * 2 = 0.8$ , tam kısım 0
- $0.8 * 2 = 1.6$ , tam kısım 1
- $0.6 * 2 = 1.2$ , tam kısım 1
- $0.2 * 2 = 0.4$ , tam kısım 0
- ...
- Virgülden 0 değerine asla ulaşamaz
- $\rightarrow (0.00110011 \dots)_2$



# İşaretli ve İşaretsiz Sayılar

- İşaretsiz sayılarda tüm basamaklar konumsal gösterimine göre değerlendirilerek sayının genliği hesaplanır
- İkili işaretli sayılarda 3 farklı gösterim söz konusudur
  - İşaretli genlik gösterimi
  - 1'e tümleyen işaretli sayı
  - 2'ye tümleyen işaretli sayı

$b_3b_2b_1b_0$	Sign and magnitude	1's complement	2's complement
0111	+7	+7	+7
0110	+6	+6	+6
0101	+5	+5	+5
0100	+4	+4	+4
0011	+3	+3	+3
0010	+2	+2	+2
0001	+1	+1	+1
0000	+0	+0	+0
1000	-0	-7	-8
1001	-1	-6	-7
1010	-2	-5	-6
1011	-3	-4	-5
1100	-4	-3	-4
1101	-5	-2	-3
1110	-6	-1	-2
1111	-7	-0	-1

# Bilgisayarlarda Veri Formatları

- Bilgisayarlarda alfanümerik verinin tutulması için farklı formatlar kullanılmaktadır
- ASCII
- Unicode
- BCD
- İşaretli, işaretsiz sayılar
- Kayan noktalı sayılar

# Bilgisayarlarda Veri Formatları

- ASCII (American Standard Code for Information Interchange)
- Alfanoümeric karakterler 7 bitlik gösterim ile ifade edilirler

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0 ^@  NUL  NULL	1 ^A  SOH  START OF HEADING	2 ^B  STX  START OF TEXT	3 ^C  ETX  END OF TEXT	4 ^D  EOT  END OF TRANSM.	5 ^E  ENQ  ENQUIRY	6 ^F  ACK  ACKNOWL- EDGE	7 ^G  BEL  BELL	8 ^H  BS  BACKSP.	9 ^I  HT  CHARACT. TAB'TION	10 ^J  LF  LINE FEED	11 ^K  VT  LINE TAB'TION	12 ^L  FF  FORM FEED	13 ^M  CR  CARRIAGE RETURN	14 ^N  SO  SHIFT OUT	15 ^O  SI  SHIFT IN
1	16 ^P  DLE  DATALINK ESCAPE	17 ^Q  DC1  DEVICE CONTROL1	18 ^R  DC2  DEVICE CONTROL2	19 ^S  DC3  DEVICE CONTROL3	20 ^T  DC4  DEVICE CONTROL4	21 ^U  NAK  NEG. ACK- NOWLEDGE	22 ^V  SYN  SYNCHR. IDLE	23 ^W  ETB  END OF TRANS.	24 ^X  CAN  CANCEL	25 ^Y  EM  END OF MEDIUM	26 ^Z  SUB  SUBS- TITUTE	27 ^[  ESC  ESCAPE	28 ^\  FS  INFO. SEP. 4	29 ^]  GS  INFO. SEP. 3	30 ^^  RS  INFO. SEP. 2	31 ^_  US  INFO. SEP. 1
2	32   SPACE	33 excl;  !	34 quot;  "	35 num;  #	36 dollar;  \$	37 percnt;  %	38 amp;  &	39 apos;  '	40 lpar;  (	41 rpar;  )	42 ast;  *	43 plus;  +	44 comma;  ,	45  -	46 period;  .	47 sol;  /
3	48   DIGIT ZERO	49   DIGIT ONE	50   DIGIT TWO	51   DIGIT THREE	52   DIGIT FOUR	53   DIGIT FIVE	54   DIGIT SIX	55   DIGIT SEVEN	56   DIGIT EIGHT	57   DIGIT NINE	58 colon;  :	59 semi;  ;	60 lt;  <	61 equals;  =	62 gt;  >	63 quest;  ?
4	64 commat;  @  COMM'IAL AT	65   A	66   B	67   C	68   D	69   E	70   F	71   G	72   H	73   I	74   J	75   K	76   L	77   M	78   N	79   O
5	80   P	81   Q	82   R	83   S	84   T	85   U	86   V	87   W	88   X	89   Y	90   Z	91 lsqb;  [	92 bsol;  \  REVERSE SOLIDUS	93 rsqb;  ]	94 hat;  ^	95 lowbar;  _
6	96 grave;  ,  GRAVE ACCENT	97   a	98   b	99   c	100   d	101   e	102   f	103   g	104   h	105   i	106   j	107   k	108   l	109   m	110   n	111   o
7	112   p	113   q	114   r	115   s	116   t	117   u	118   v	119   w	120   x	121   y	122   z	123   {  L. CURLY BRACKET	124      VERTICAL LINE	125   }  R. CURLY BRACKET	126   ~  TILDE	127 ^?  DEL  DELETE

# Bilgisayarlarda Veri Formatları

- Unicode: alfanümerik karakterler 16 bitlik bir sayı ile kodlanır
- Farklı alfabeler ve yazı sistemlerini destekler
- *<http://www.unicode.org>*

# Bilgisayarlarda Veri Formatları

- BCD (Binary-Coded Decimal): Onluk gösterimdeki her bir basamak 4-bit ile ifade edilir
- BCD gösterim üzerinden işlemleri destekleyen assembly komutları mevcuttur
- Karmaşık işlemler için çok uygun değildir
- $(526)_{10} = (0101\ 0010\ 0110)_{BCD}$

# Bilgisayarlarda Veri Formatları

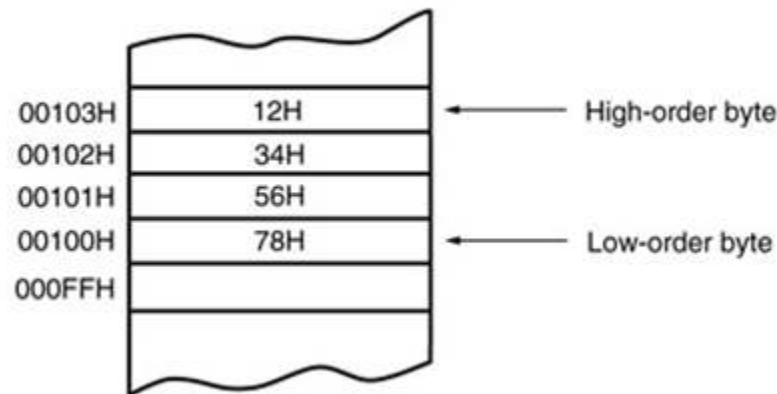
- 8-bitlik ikili sayılar
  - İşaretli, işaretsiz sayıların gösteriminde kullanılır
- En anlamlı bit
  - işaretsiz sayılarda  $2^7 = 128$  kuvvetindedir
  - İşaretli sayılarda  $-2^7 = -128$  kuvvetindedir
- 8 bit işaretsiz sayılar 0 ile 255 arasındaki değerleri gösterebilir
- 8 bit işaretli sayılar -128 ile 127 arasındaki değerleri gösterebilir
- Bilgisayarlarda işaretli sayılar 2'ye tümleyen yapıdadır



# Bilgisayarlarda Veri Formatları

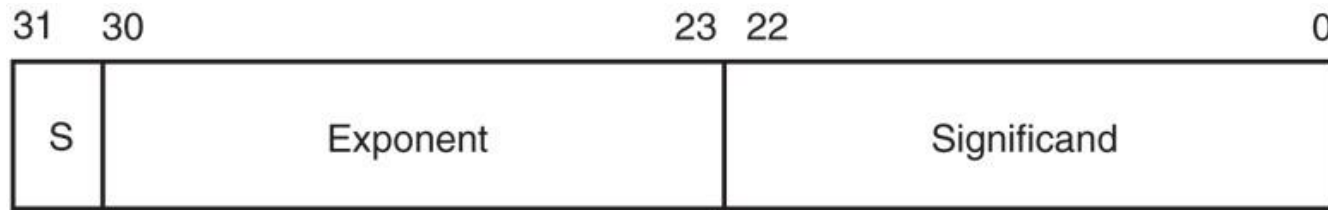
- 16-bitlik ikili sayılar (Word sized data)
- 32-bitlik ikili sayılar (Double word sized data)
- 16-bit, 32-bit sayılarda her bir byte değerinin saklanma yerine göre → little endian, big endian

Little endian →

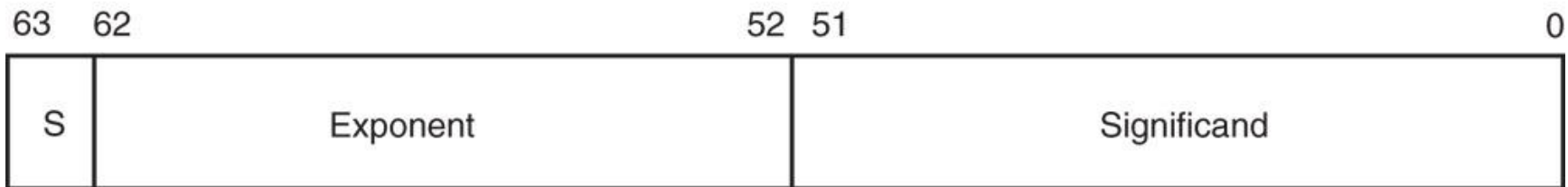


# Bilgisayarlarda Veri Formatları

- Gerçel sayılar aşağıdaki 2 hassasiyette ifade edilebilir:
  - 4-byte single precision
  - 8-byte double precision



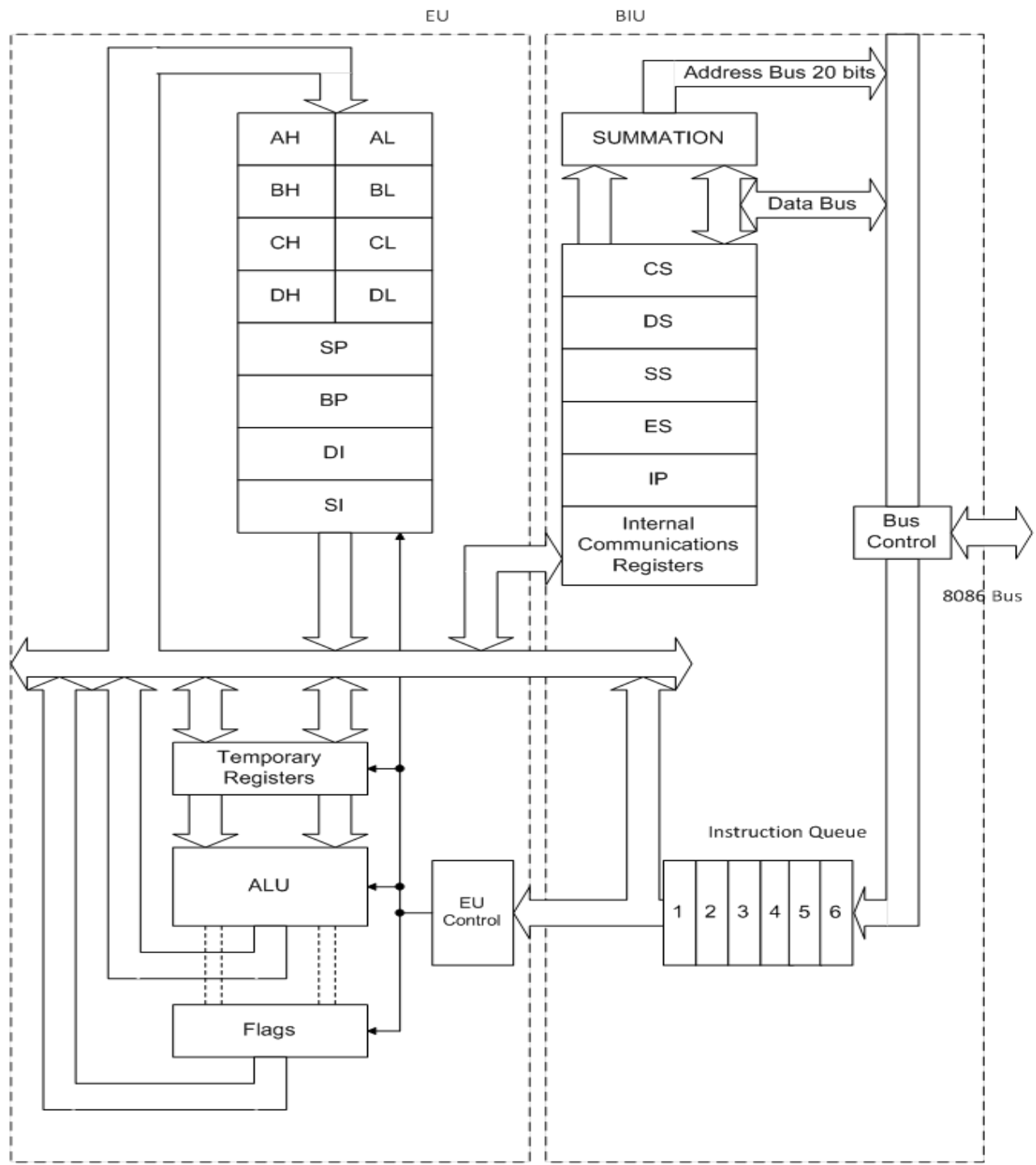
(a)



(b)

# 8086 İÇ YAPISI

# 8086 iç Yapısı



# 8086 Yazmaçları

- AX
  - AL
  - AH
- BX
  - BL
  - BH
- SI
- DI
- CX
  - CL
  - CH
- DX
  - DL
  - DH
- SP
- BP

Genel amaçlı yazmaçlar

Segment yazmaçları

- SS
- CS
- DS
- ES
- IP

Özel amaçlı yazmaç

Bayrak yazmacı

- FLAGS
  - CF
  - PF
  - AF
  - ZF
  - SF
  - TF
  - IF
  - DF
  - OF

# 8086 Yazmaçları – AX, AL, AH

- AX : 16 bitlik akümülatör yazmaç
- AH, AL : 8 bitlik akümülatör yazmaçlar
- Aritmetik, lojik ve veri transferi işlemlerinde kullanılabilir
- Çarpma ve bölme işlemlerinde gizli operand olark kullanılır
- Giriş çıkış komutlarında kullanılır

# 8086 Yazmaçları – BX, BL, BH

- BX : 16 bitlik genel amaçlı yazmaç, (base register)
- BL, BH : 8 bitlik genel amaçlı yazmaçlar
- Dizi şeklindeki veri erişiminde kullanılır

# 8086 Yazmaçları – CX, CL, CH

- CX : 16 bitlik genel amaçlı yazmaç
- CL, CH : 8 bitlik genel amaçlı yazmaçlar
- Tekrarlı işlemlerde tekrar sayısını saklar (CX)
- Öteleme ve kaydırma işlemlerinde tekrar sayısını saklar (CL)



# 8086 Yazmaçları – DX, DL, DH

- DX : 16 bitlik genel amaçlı yazmaç
- DL, DH : 8 bitlik genel amaçlı yazmaçlar
- Çarpma ve bölme komutlarında bölünen sayıyı oluşturmak için kullanılır
- Giriş çıkış işlemlerinde port numarasını saklar

# 8086 Yazmaçları – SP

- SP : yığın yazmacı (stack pointer)
- Yığının en üst adresini işaretlemek için kullanılır
- SS ile birlikte kullanılır
- Her zaman çift bir değer gösterir
- WORD tipinde veriyi gösterir

# 8086 Yazmaçları – BP

- BP : Base pointer
- Fonksiyona parametre aktarılırken kullanılır
- SS ile birlikte kullanılır

# 8086 Yazmaçları – SI

- SI : kaynak indisi yazmacı (source index)
- Dizi komutlarında kaynak indisini tutar
  - DS ile birlikte kullanılır

# 8086 Yazmaçları – DI

- DI : hedef indisi yazmacı (destination index)
- Dizi komutlarında hedef indisini tutar
  - ES ile birlikte kullanılır

# 8086 Kesim (Segment) Yazmaçları

- CS : Kod segment, IP ile kullanılır
  - DS : Data segment, BX, SI, DI ile kullanılır
  - ES : Extra segment, DS gibi
  - SS : Stack segment, BP ve SP ile kullanılır
- 
- DS=1230H, SI=0045H ikilisi ile erişilen fiziki adres  
 $12300H + 0045H = 12345H$

# 8086 Yazmaçları – IP

- IP : Instruction pointer
  - Sıradaki işlenecek komutu gösterir
  - CS ile birlikte kullanılır
- 
- Efektif program adresi :
  - $CS \times 10H + IP$

# 8086 Bayrak Yazmacı

- **Carry Flag (CF)** : İşaretsiz işlemlerde taşma olursa 1 değerini alır
- **Parity Flag (PF)** : İşlem sonucunda 1 olan bitlerin sayısı tek ise 0, çift ise 1 değerini alır
- **Auxiliary Flag (AF)** : 4 bitlik kısımların toplama-çıkarma sonucu elde değerini tutar
- **Zero Flag (ZF)** : İşlem sonucu 0 ise ZF=1 olur
- **Sign Flag (SF)** : İşlem sonucu negatif ise SF=1 olur



# 8086 Bayrak Yazmacı

- **Trap Flag (TF)** : Her komuttan sonra kesme oluşmasını sağlar
- **Interrupt enable Flag (IF)** : Kesme kaynaklarının kesme oluşturmalarına izin verir
- **Direction Flag (DF)** : Dizi işlemlerinde başlangıç adresinden itibaren arttırarak/azaltarak sıradaki göze erişimi belirler
- **Overflow Flag (OF)** : İşaretli işlemlerde taşma durumunda 1 değerini alır

# 32 bit $\mu$ P Yazmaçları (80386)

- EAX
  - AX
    - AL
    - AH
- ECX
  - CX
    - CL
    - CH
- SS
- CS
- DS
- ES
- EIP
- FS
- GS
- EFLAGS
- EBX
  - BX
    - BL
    - BH
- EDX
  - DX
    - DL
    - DH
- ESI
- ESP
- EDI
- EBP

# 64 bit $\mu$ P Yazmaçları

- RAX
  - EAX
    - AX
      - AL
      - AH
- RBX
  - EBX
    - BX
      - BL
      - BH
- RSI
- RDI
- RCX
  - ECX
    - CX
      - CL
      - CH
- RDX
  - EDX
    - DX
      - DL
      - DH
- RSP
- RBP
- SS
- CS
- DS
- ES
- FS
- GS
- RIP
- R8-R15
- RFLAGS

REAL – PROTECTED MOD

# Real Mod Hafıza Adresleme

- 8086 real modda hafıza adresleme yapar
- Real modda sadece 1MB alan adreslenebilir
- 8086 hafıza uzayı 1MB (20 adres ucu → 1MB)
- Tüm bilgisayarlar açıldığında real modda açılır

# Real Mod Hafıza Adresleme

- Real modda segment adres ve ofset adres değerlerinin birleşimiyle hafızada istenen alana erişilir
- Bir segment değeri 64KB'lık alanı gösterir (NEDEN?)
- Ofset değeri 64KB'lık alan içinde bir yeri gösterir
- Örnek: Segment değeri 1000H, ofset değeri 2000H ise mikroişlemcide erişilen fiziki adres ne olur?
- $1000H \times 10H + 2000H = 12000H$  (1000H:2000H)

# Real Mod Hafıza Adresleme

- Varsayılan Segment ve Ofset yazmaçları
- Program hafızasına erişimde CS:IP birlikte kullanılır
- Yığın (stack) erişiminde SS:SP veya SS:BP kullanılır
- Veri erişiminde DS:BX, DS:DI, DS:SI kullanılır
- String işlemlerinde ES:DI ile kullanılır

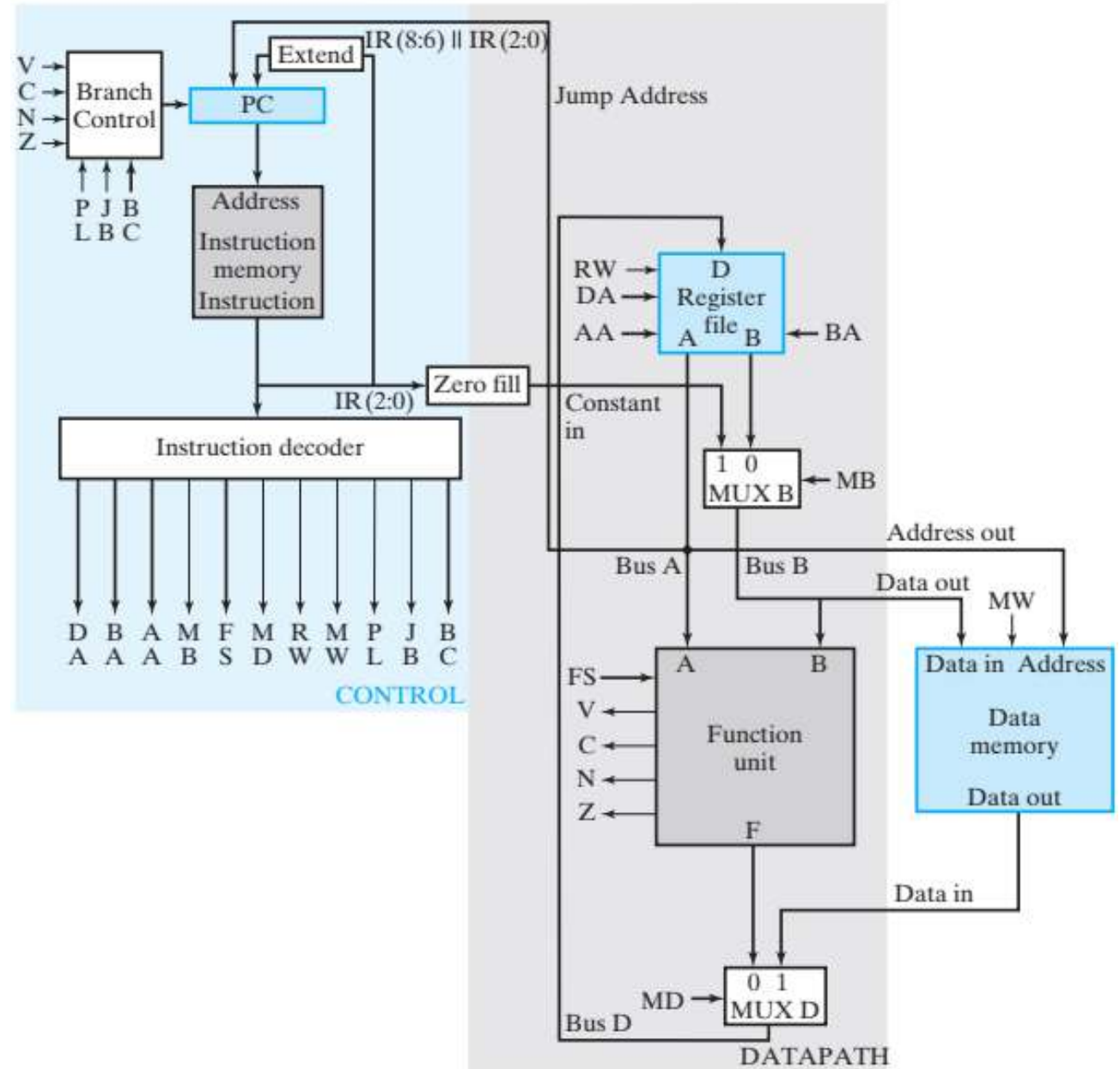
# Protected Mod Hafıza Adresleme

- 1MB'tan daha geniş hafızayı adreslemede kullanılır
- Segment adres değeri bir tablonun (descriptor table) bir satırını gösterir
- Bu tablonun herbir gözünde kullanım için ayrılmış segmentin boyu, yeri, erişim izinleri yazılıdır
- Tabloların global ve yerel versiyonları vardır
- Global descriptor tablosunda tüm programlar için kullanılabilecek segment bilgisi yer alır
- Yerel descriptor tablosunda ise uygulamaya özel segment bilgisi yer alır



EK – Komut nasıl değerlendirilir:

# Basit bir işlemci iç yapısı



# Basit bir işlemciye ilişkin komut çözümleme devresi

