

Microprocessors

Hafta 1

Bölüm 1: Ümumaralandırma ve Kodlama Sistemleri

Onluk (Decimal) Sistem: İnsanların hesaplama yaparken kullandığı sistemdir. Her basamak 0'dan 9'a kadar bir rakam alabilmektedir. Her basamak 10'un kuvvetine göre artar. Decimal sayılar $(sayı)_{10}$ olarak gösterilir.

$$345_{10} \Rightarrow 3 \times 10^2 + 4 \times 10^1 + 5 \times 10^0$$

İkili (Binary) Sistem: Binary sayı sistemi, 0 ve 1 rakamlarının kullandığı bir sayı sistemiidir. Binary sisteme her basamak 2'nin üssü olacak şekilde artar. $(sayı)_2$ olarak gösterilir.

Most Significant Bit (MSB) Least Significant Bit (LSB)

$$1011_2 \Rightarrow 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$$

Decimal Binary Dönüşümü

25_{10} decimal sayıyı binary sisteme çevirelim. Bunu yapmak için 25 sayısını bölüm 0 olana kadar böleceğiz. bölme sırasında kalan sayıyı da tutacağız. İlk kalan sayı least significant bit (LSB), son kalan sayı ise most significant bit (MSB). Bölüm 0 olunca kalan sayılar MSB'den LSB'ye doğru sırayla yazılır ve dönüşüm tamamlanır.

Cözüm:

	Bölüm	kalan
1)	25/2	12
2)	12/2	6
3)	6/2	3
	3/2	1
	1/2	0 <small>=> işlem burada biter.</small>

böylece 25_{10} decimal sayısının binary dönüşümü sonucu 11001_2 idir.

Binary Decimal Dönüşümü

Binary sayıyı decimal sayıya çevirirken en sağdaki sayıdan başlanılarak 2^0 ile çarpılır ve soldaki her bir sayı, 2 nin üssü 1 artırılarak çarpılır ve bulunan sonuçlar toplanarak decimal sayı bulunur.

Binary	Decimal	Binary
110101_2	1×2^0	1
	0×2^1	00
	1×2^2	100
	0×2^3	0000
	1×2^4	10000
	1×2^5	$+ 100\ 000$
	<u>$+ 32$</u>	<u>110101</u>
	53	

Hexadecimal Sistem: 0'dan F'ye kadar 16 tane sayı, gösteren sayı sistemidir. 0'dan 9'a kadar sayı ile gösterilirken 10'dan 15'e kadar harf ile gösterilir.

Binary Hexadecimal Dönüşümü

Binary sayıya hexadecimal sayıya çevirmek için binary sayı sağdan sola doğru dörtlü gruplanır. Gruplanan dörtlü binary sayılar hexadecimal karşılıklarına göre yazılır.

Decimal Hexadecimal Dönüşümü

2 şekilde yapılabilir.

1) Önce decimal sayı binary'ye çevrilir. Binary'ye çevrilen sayıya dönüşüm yapılır.

2) Decimal Sayı, 16'ya bölüm 0 olana kadar tekrahlı böl ve kalanları sırayla yaz.

Table 0-1: Decimal, Binary, and Hex

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

Hexadecimal Decimal Dönüşümü

2 şekilde yapılabilir:

1) Hexadecimal sayıyı binary sayıya çevirdikten sonra decimal sayıya çevirmek

2) Hexadecimal sayıları ağırlıkları ile toplayarak elde edilir.

Örnek 1

1001 1111 0101₂ Sayısını hexadecimal sayıya çeviriniz

★ Binary sayıları sağdan sola doğru dörtlü grupta ve hex sonucunu yazın

$$\begin{array}{c} 1001 \\ \frac{2^3 + 2^2 + 2^1 + 2^0}{9} \end{array} \quad \begin{array}{c} 1111 \\ \frac{2^3 + 2^2 + 2^1 + 2^0}{F} \end{array} \quad \begin{array}{c} 0101 \\ \frac{2^3 + 2^2 + 2^1 + 2^0}{5} \end{array}$$

$$1001 1111 0101_2 = 9F5_{16}$$

Örnek 2:

$29B_{16}$ hex sayısının binary dönüşümünü yapınız

$$\begin{array}{ccc} 2 & 9 & B \\ 0010 & 1001 & 1011 \end{array}$$

$$29B_{16} = 0010 1001 1011_2$$

Örnek 3

45_{10} sayısını hexadecimal sayıya çeviriniz

$$\begin{array}{r} 45/16 \\ 2/16 \end{array} \quad \begin{array}{l} \text{bölüm} \\ 2 \\ 0 \end{array} \quad \begin{array}{l} \text{kalan} \\ \text{yazma} \\ \text{Sırası} \end{array} \quad \begin{array}{l} 13(D) \rightarrow \text{Least Significant Digit (LSD)} \\ 2 \rightarrow \text{Most Significant Digit (MSD)} \end{array}$$

Örnek 4:

1714_{10} sayısını hexadecimal sayıya çeviriniz

	bölüm	kalan
$1714/16$	107	2 (LSD)
$107/16$	6	11 (Hex B)
$6/16$	0	6 (MSD)

$$1714_{10} = 2B6_{16}$$

Örnek 5:

$6B2_{16}$ sayısını decimal sayıya çeviriniz

$$\begin{array}{r} 6B2_{16} \\ \text{---} \\ \text{---} \end{array}$$

Calculation steps:

- $2 \times 16^0 = 2$
- $11 \times 16^1 = 176$
- $6 \times 16^3 = 1536$
- $\underline{+ 176}$
- 1714

$$6B2_{16} = 1714_{10}$$

Bölüm 2: Bilgisayarın İç Yapısı

Önemli Kavramlar

Bit

0
0000

Nibble

00000000

Byte

0000000000000000

Word

$$1 \text{ Kilobyte} = 2^{10} \text{ byte}$$

$$1 \text{ Megabyte} = 2^{20} \text{ byte}$$

$$1 \text{ Gigabyte} = 2^{30} \text{ byte}$$

$$1 \text{ Terabyte} = 2^{40} \text{ byte}$$

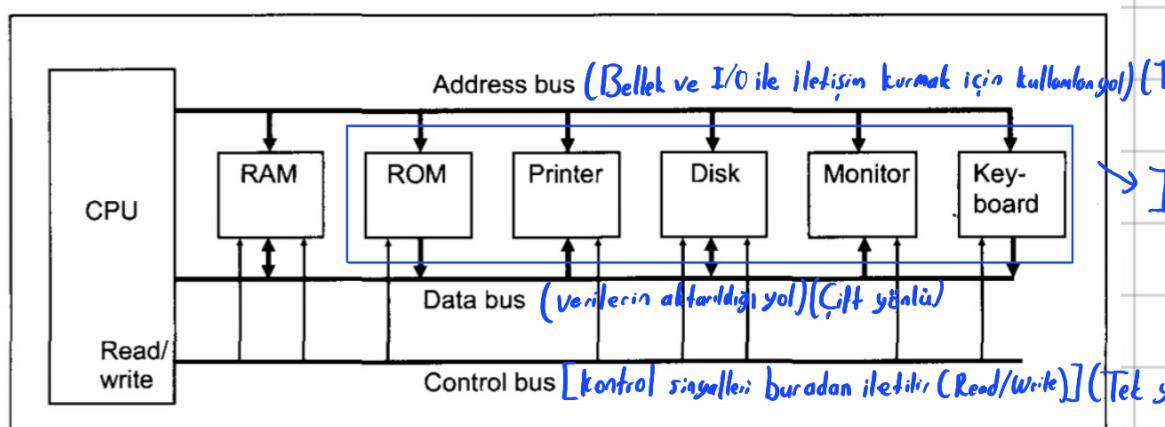


Figure 0-4. Internal Organization of Computers

Bir CPU'nun 16 data yolu, 24 adresleme yolu varsa işlemci 16 bittir ve 16 megabyte (2^{24}) bellek kullanabileceğini gösterir.

CPU'nun İç İşleyisi

A, B, C ve D registerlara sahip; 8 bit data bus ve 16 bit address bus'a sahip bir işlemci olsun. Bu işlemci bellege 0000 ile FFFFH (yani 10000 H (buradaki H "Hexadecimal" ifade etmektedir) konum bulunmaktadır). Burada yapılmak istenen 21H değerini A registerine taşımaktır. Daha sonra ise register A'daki değeri sırayla 42H ve 12H değerleri ile toplamaktır.

Action	Code	Data	
Move value 21H into register A	B0H	21H	Register A $\Rightarrow 21H$
Add value 42H to register A	04H	42H	Register A $\Rightarrow 63H$
Add value 12H to register A	04H	12H	Register A $\Rightarrow 75H$

Yukarıdaki program 1400H numaralı bellek adresinde çalışsaydı aşağıdaki tablo bellek adresinin içindeki değerleri içerecektir.

Memory address	Contents of memory address
1400	(B0) the code for moving a value to register A
1401	(21) the value to be moved
1402	(04) the code for adding a value to register A
1403	(42) the value to be added
1404	(04) the code for adding a value to register A
1405	(12) the value to be added
1406	(F4) the code for halt

- 1) CPU'nun program counter, 0000 ile FFFFH arasında değer alabilmektedir. Yukarıdaki programın çalışması için program counter'a 1400H adresi yüklenmiş olmalıdır. Program counter'a ilk instruction'un adresi yükledikten sonra CPU execute için hazırdır.
- 2) CPU 1400H değerini address bus'a gönderir. CPU Read Singalini gönderdikten sonra belleğin içerisindeki veri data bus'a aktarılır ve CPU'ya gönderilir.
- 3) CPU, instruction decoder aracılığıyla B0 instruction'ını decode eder. Decode işleminden sonra CPU, register A'ya bir sonraki bellek adresindeki değeri taşıması gerektiğini bilir. 1401 bellek adresindeki 21H değerini register A'ya taşıır. Taşıma işlemi sonrasında program counter bir sonraki instruction'u fetch eder (burada 1402H)

4) 1402H bellek konumundaki 04H komutu fetch edilir. Decode işleminden sonra CPU register R_A'yi, bir sonraki bellek adresinde (1403H) bulunan değer ile toplaması gerekligi bilir. 1403 adresindeki değeri (42H) getirdiğten sonra ALU üzerinde toplama işlemi yapılır. ALU'daki işlem sonucu register R_A'ya yazılır ve program counter 1404H olur.

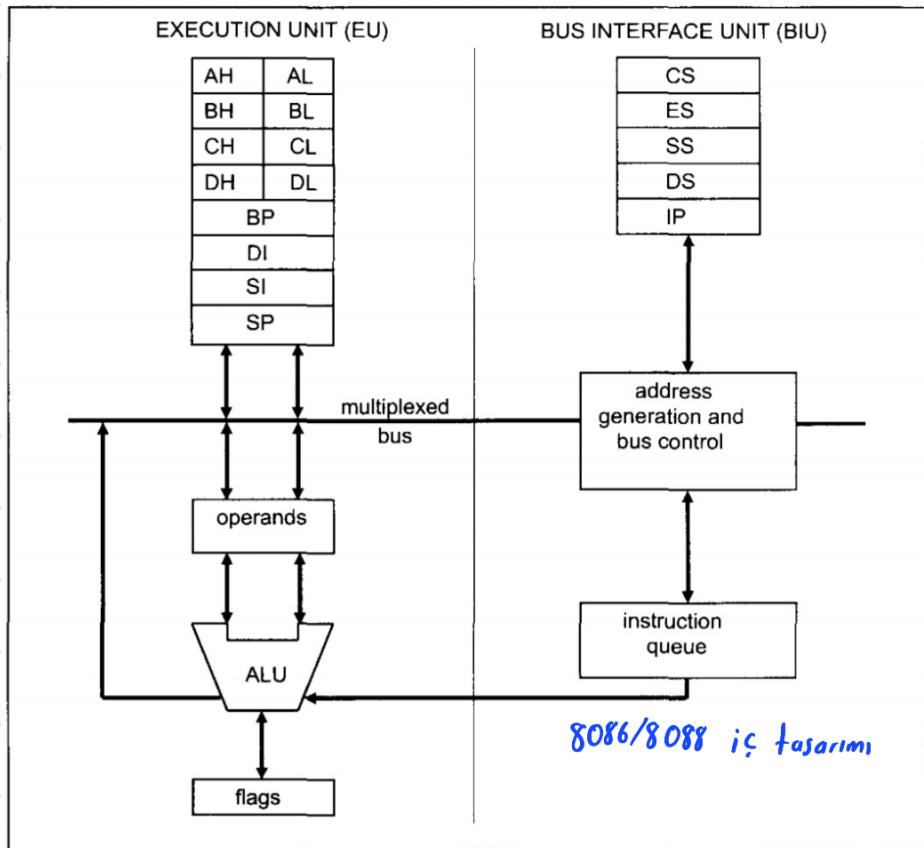
5) 1404H değeri adres basa aktarılır ve kod fetch, decode ve execute edilir. Bu kod geçenki toplama işlemi ile aynıdır. 1405H adresindeki değer ile register R_A'daki değer ALU'da toplanır ve program counter 1406H olur.

6) En sonunda 1406H adresindeki kod fetch, decode ve execute edilir. HALT kodu CPU'ya program Counter'i artttmasını durdurur ve bir sonraki instruction'u bekler.

Table 1-1: Evolution of Intel's Microprocessors

Product	8080	8085	8086	8088	80286	80386	80486
Year introduced	1974	1976	1978	1979	1982	1985	1989
Clock rate (MHz)	2 - 3	3 - 8	5 - 10	5 - 8	6 - 16	16 - 33	25 - 50
No. transistors	4500	6500	29,000	29,000	130,000	275,000	1.2 million
Physical memory	64K	64K	1M	1M	16M	4G	4G
Internal data bus	8	8	16	16	16	32	32
External data bus	8	8	16	8	16	32	32
Address bus	16	16	20	20	24	32	32
Data type (bits)	8	8	8, 16	8, 16	8, 16	8, 16, 32	8, 16, 32

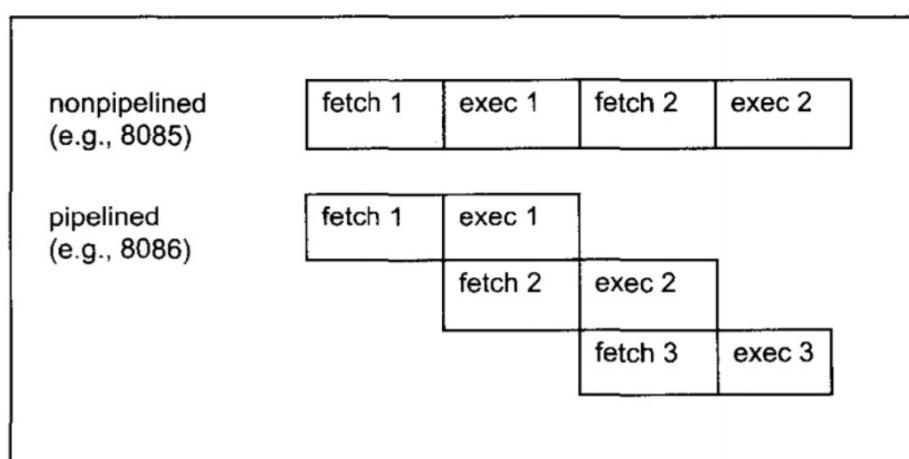
- ★ 8086, 16 bit bir işlemcidir. İlk defa pipelining bu işlemciye kullanılmıştır.
- ★ 80286; real mode ve projected mode olmak üzere 2 moda sahiptir. real mode standart kullanım için kullanılırken protected mode işletim sistemini koruma amacıyla oluşturulmuştur.



8086/8088 iç yapısı

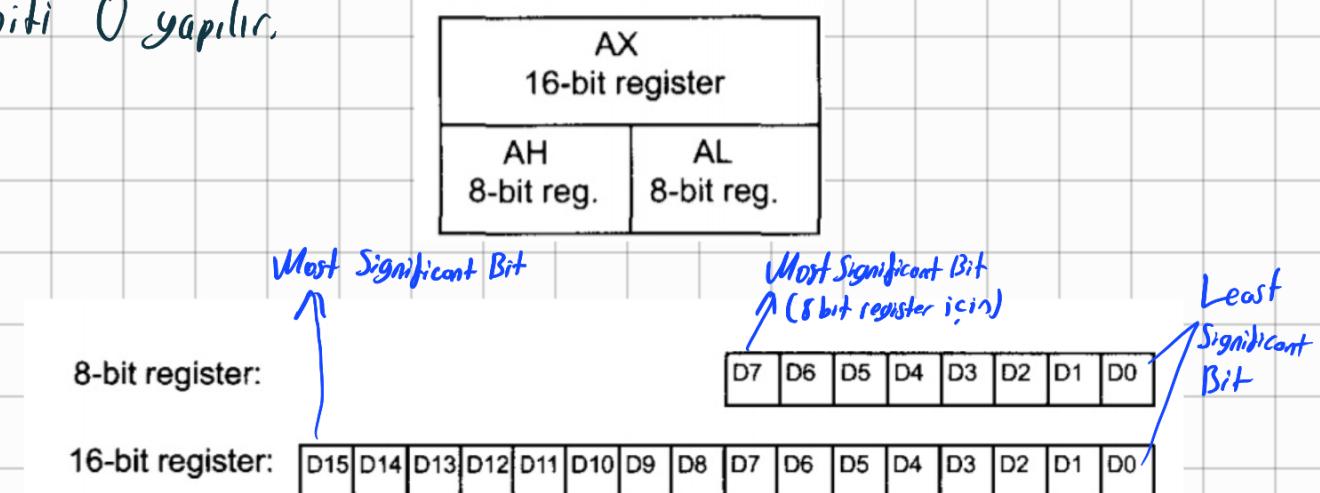
Pipelining

Intel 8086, kendinden önceki işlemcilerden farklı olarak pipeline'lı sahiptir.



Registers

Registers, CPU içerisinde verilerin geçici olarak tutulduğu kısımdır. 8086/8088'de registerler 6 kategoriye ayrılmaktadır. Genel amaçlı registerler 16 bit ve 8 bit olarak erişilebilirdir. Kalan tüm registerler 16 bit olarak erişebilirler. 8086/88'de data tipleri 8 ya da 16 bittir. 12 bitlik bir dataya erişmek istenildiğinde en yüksek değerli 4 biti 0 yapılır.



8086/88'deki forklu registerlar forklu fonksiyonlar için kullanılır. Bazı instructionlar görevlerini yerine getirebilmek için spesifik registerleri kullanır. Register isimlendirmelerindeki ilk harf register'in hangi amaçla kullanıldığını belirtir. AX akümülatör için kullanılır. BX base addressing register olarak kullanılır. CX looplarda counter olarak kullanılır. DX ise I/O işlemlerinde datayı göstermek için kullanılır.

Table 1-2: Registers of the 8086/286 by Category

Category	Bits	Register Names
General	16	AX, BX, CX, DX
	8	AH, AL, BH, BL, CH, CL, DH, DL
Pointer	16	SP (stack pointer), BP (base pointer)
Index	16	SI (source index), DI (destination index)
Segment	16	CS (code segment), DS (data segment), SS (stack segment), ES (extra segment)
Instruction	16	IP (instruction pointer)
Flag	16	FR (flag register)

Not: Genel amaçlı registerler 16 bit bütün olarak (AX gibi), sadece yüksek bit (AH) olarak ve sadece düşük bit olarak (AL) erişilebilir.

Assembly Programlama

MOV Instruction

MOV instruction'ı bir datayı başka bir yere kopyalar

- MOV hedef, kaynak ; kaynagi hedefe kopgolar

MOV	CL,55H	;move 55H into register CL
MOV	DL,CL	;copy the contents of CL into DL (now DL=CL=55H)
MOV	AH,DL	;copy the contents of DL into AH (now AH=DL=55H)
MOV	AL,AH	;copy the contents of AH into AL (now AL=AH=55H)
MOV	BH,CL	;copy the contents of CL into BH (now BH=CL=55H)
MOV	CH,BH	;copy the contents of BH into CH (now CH=BH=55H)

The use of 16-bit registers is demonstrated below.

MOV	CX,468FH	;move 468FH into CX (now CH=46,CL=8F)
MOV	AX,CX	;copy contents of CX to AX (now AX=CX=468FH)
MOV	DX,AX	;copy contents of AX to DX (now DX=AX=468FH)
MOV	BX,DX	;copy contents of DX to BX (now BX=DX=468FH)
MOV	DI,BX	;now DI=BX=468FH
MOV	SI,DI	;now SI=DI=468FH
MOV	DS,SI	;now DS=SI=468FH
MOV	BP,DI	;now BP=DI=468FH

★ "MOV AL,AX" komutu hatalıdır. Çünkü 8 bitlik bir register'in içine 16 bitlik veri kopyalanamaz.

msb LSB

MOV	AX,58FCH	;move 58FCH into AX	(LEGAL)
MOV	DX,6678H	;move 6678H into DX	(LEGAL)
MOV	SI,924BH	;move 924B into SI	(LEGAL)
MOV	BP,2459H	;move 2459H into BP	(LEGAL)
MOV	DS,2341H	;move 2341H into DS	(ILLEGAL)
MOV	CX,8876H	;move 8876H into CX	(LEGAL)
MOV	CS,3F47H	;move 3F47H into CS	(ILLEGAL)
MOV	BH,99H	;move 99H into BH	(LEGAL)

★ Segment registerlerine (CS, DS, SS, ES) adres üzerinden doğrudan Veri kopyalanmaz. Yasaktır.

Segment registerlara veri yüklemek için non segment registerlara değer yüklenir. Daha sonra ise non segment registerdaki veri segment register'a kopyalanır.

MOV AX, 2345H

MOV DS, AX

MOV DI, 1400H

MOV ES, DI

★ 16 bit register'a kopyalandan değer FFH değerinden küçük ise kalan değerler "0" olarak kabul edilir. Örneğin "MOV BX, 5" komutunun sonucunda BX=0005 olur. Yani BH=00 iken BL=05'tir.

★ Register'a alabileceği değerden büyük değer kopyalamak hataya sebep olur.

MOV BL, 7F2H ;ILLEGAL: 7F2H is larger than 8 bits
MOV AX, 2FE456H ;ILLEGAL: the value is larger than AX

Notumuzı beğeniyorsanız beğenmeyi ve kanalımıza abone olmayı unutmayın