



**Microprocessor**

# **Introduction to Microprocessor**

Dr. Cahit Karakuş  
Esenyurt Üniversitesi

# Günümüz Bilgisayar Mühendisliği

- 1- Quantum Computer (Quantum Fiziği) and Quantum Computing (Matematik, Olasılık, İstatistiksel analiz)
- 2- Veri Bilimi, Veri tabanı yönetimi, Veri hazırlama
- 3- Yapay Zeka: Algoritma, matematik ve senaryo tabanlı modüller üretme (%75), Makine öğrenmesi algoritmalarından veri analitiği (%3), Son kullanıcılara yönelik uygulamalar geliştirme (3), Kullanıcı davranışlarını belirleme (%20)
- 4- Otomasyon /Otonom & Computer science: Bilgisayar sistemli makineler (Bilgisayar Organizasyonu, Mikroişlemci, Assembly)
- 5- Algoritma ve Matematiksel modeller
- 6- Yazılım Dilleri: Python, Java Script, C++, Matlab, Assembly
- 7- Uygulamalı Matematik: Lineer Cebir, Olasılık, İstatistik, türev, integral, matris, vektör, öz vektör, öz değerler

# Common Powers

Prefix	Symbol	Power of 10	Power of 2	Prefix	Symbol	Power of 10
Kilo	K	1 thousand = $10^3$	$2^{10} = 1024$	Milli	m	1 thousandth = $10^{-3}$
Mega	M	1 million = $10^6$	$2^{20}$	Micro	$\mu$	1 millionth = $10^{-6}$
Giga	G	1 billion = $10^9$	$2^{30}$	Nano	n	1 billionth = $10^{-9}$
Tera	T	1 trillion = $10^{12}$	$2^{40}$	Pico	p	1 trillionth = $10^{-12}$
Peta	P	1 quadrillion = $10^{15}$	$2^{50}$	Femto	f	1 quadrillionth = $10^{-15}$
Exa	E	1 quintillion = $10^{18}$	$2^{60}$	Atto	a	1 quintillionth = $10^{-18}$
Zetta	Z	1 sextillion = $10^{21}$	$2^{70}$	Zepto	z	1 sextillionth = $10^{-21}$
Yotta	Y	1 septillion = $10^{24}$	$2^{80}$	Yocto	y	1 septillionth = $10^{-24}$

$2^n$ : Byte, Bellek kapasitesi

$10^n$ : Bit, veri işleme hızı veya veri transfer hızı

# Laws of Exponents

- Let  $a$  and  $b$  be positive numbers and let  $x$  and  $y$  be real numbers. Then,

1.  $b^x \cdot b^y = b^{x+y}$

2.  $\frac{b^x}{b^y} = b^{x-y}$

3.  $(b^x)^y = b^{xy}$

4.  $(ab)^x = a^x b^x$

5.  $\left(\frac{a}{b}\right)^x = \frac{a^x}{b^x}$



# **Bilgisayarın Gelişimi**

## Herşey 1947 transistör geliştirilmesiyle başladı.

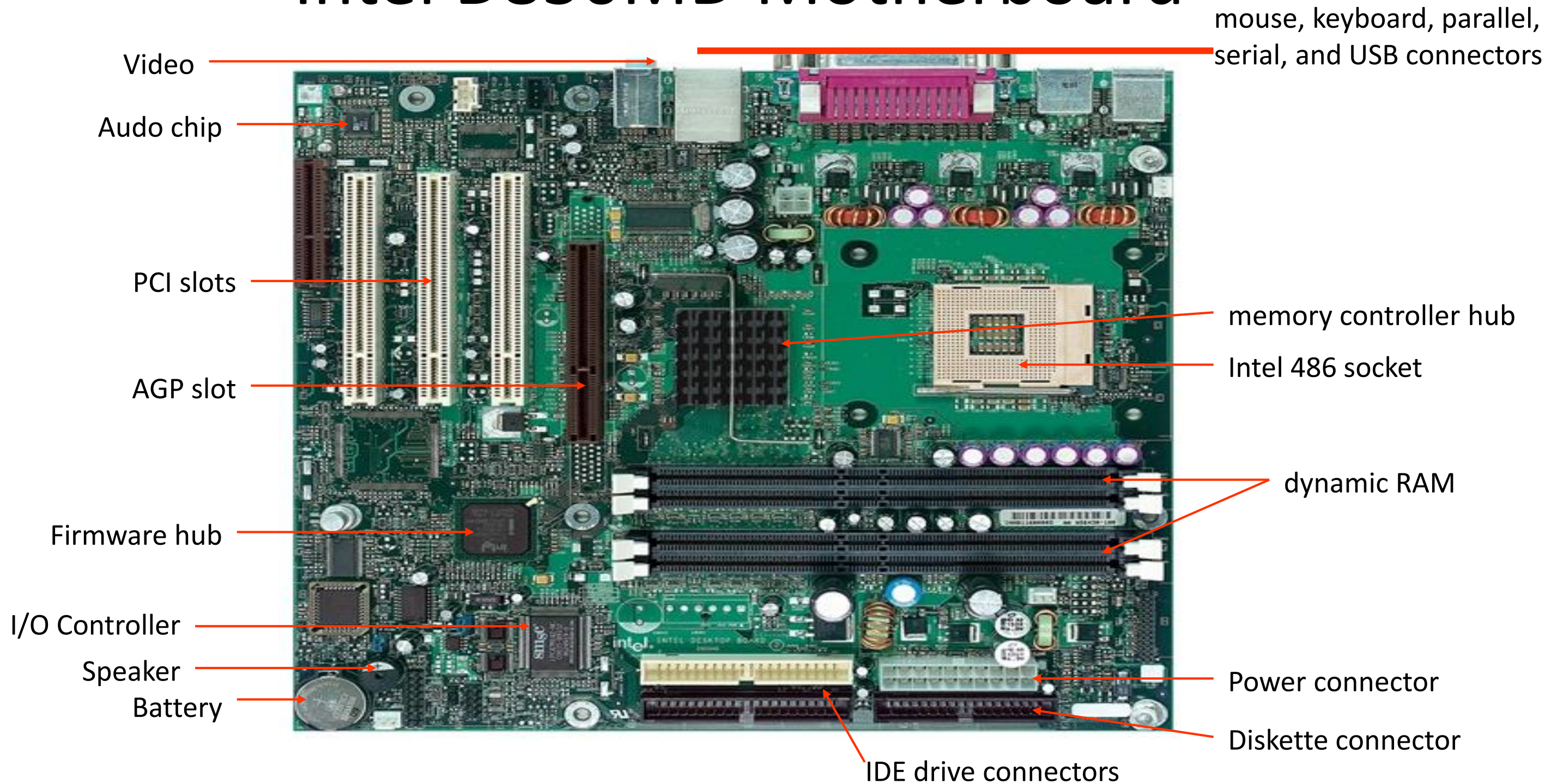
<http://www.computerhistory.org/exhibits/microprocessors/index.page>

Year	name	Data size	memory size	#instructions	
1971	4004	4	4096 4-bit	45	first microprocessor
1973	8008	8	16K bytes	48	1st 8-bit $\mu$ P
1973	8080	8	64K bytes		10 times faster than 8008
1973	MC6800	8	64K bytes		1st Motorola $\mu$ P
1977	8085	8	64K bytes	246	Intel's most successful 8-bit general-purpose $\mu$ P due to its low cost
	Z80	8			Zilog's most successful microprocessor
1978	8086	8,16	1M bytes	>20,000	1st 16-bit $\mu$ P
1979	8088	8,16	1M bytes		prefetch instruction using cache
1981	IBM decided to use 8088 in its personal computer				
1983	80286	8,16	16M		
1986	80386	8,16,32	4G		
1989	80486	8,16,32	4G		
1993	Pentium	8,16,32	4G		
1995	Pentium Pro	64	64G		
1997	Pentium II	64	64G		
1999	Pentium III	?	?		
2000	Pentium 4	?	?		

# Motherboard

- CPU socket
- External cache memory slots
- Main memory slots
- BIOS chips
- Sound synthesizer chip (optional)
- Video controller chip (optional)
- IDE, parallel, serial, USB, video, keyboard, joystick, network, and mouse connectors
- PCI bus connectors (expansion cards)

# Intel D850MD Motherboard





# Input-Output Ports

- Ethernet Port, WiFi, HDMI
- USB (universal serial bus)
  - intelligent high-speed connection to devices
  - up to 12 megabits/second
  - USB hub connects multiple devices
  - *enumeration*: computer queries devices
  - supports *hot* connections
- Parallel
  - short cable, high speed
  - common for printers
  - bidirectional, parallel data transfer
  - Intel 8255 controller chip
- Serial
  - RS-232 serial port
  - one bit at a time
  - uses long cables and modems
  - 16550 UART (universal asynchronous receiver transmitter)
  - programmable in assembly language



**What is computer?**

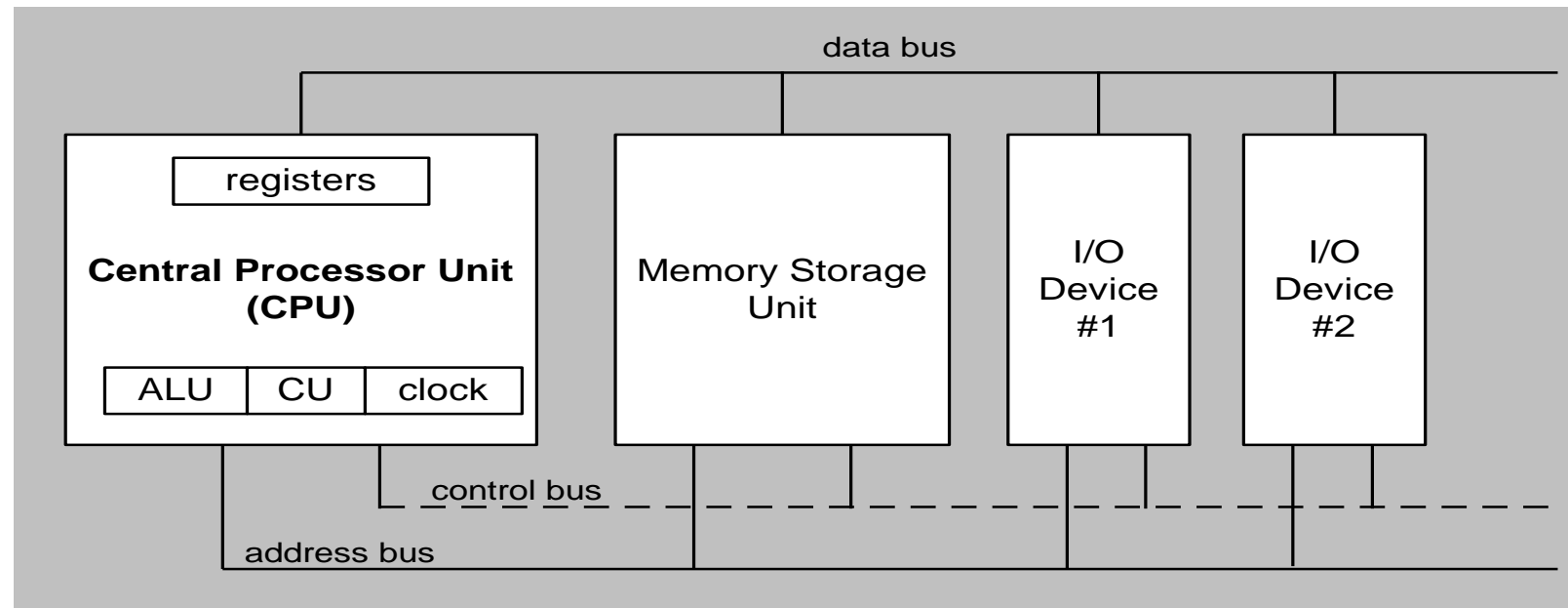
# What Is A Computer?

- Bilgisayar, veri işleme yeteneği olan bir yapıdır.
- Bilgisayar, otonom olarak bir dizi aritmetik veya mantıksal işlem gerçekleştirmek amacıyla programlanan genel amaçlı bir aygıttır.
- Yakın gelecekte binlerce işlemci, verileri çok hızlı işleyebilmesi için birarada çalışırken kuantalama hesaplama kullanacaktır. Bitler elektronlar ya da fotonlar ile temsil edilecektir.
- Information System: Verileri alan, depolayan, işleyen ve çıktı olarak bilgi sağlayan bir sistem

# Bilgisayar - Mikroişlemci

- Bilgisayar: Girdi olarak verileri alan, verileri depolayan, işleyen ve kullanışlı bir biçimde çıktı verisi sağlayan programlanabilir bir makinedir.
  - Input: Data
  - Instructions: Software, Programs
  - Output: Information (numbers, words,sounds, images)
- Mikroişlemci: Komutları alan (bellekten), kod çözen ve yürüten program kontrollü yarı iletken cihazdır (Semiconductor Device, IC: Integrated Circuit ). Bilgisayarlarda CPU (Central Processing Unit: Merkezi İşlem Birimi) olarak kullanılır.

# Bilgisayar Bisteminin Bileşenleri



## ○ Block diagram of a computer system

- Basic components of a computer system using block diagrams:

- CPU

- Memory

- Input and output unit

## ○ Bus system: data bus, address bus and control bus.

## ○ Microprocessor clock & timing system

# CPU İç Yapısı

Internal structure and basic operation of a microprocessor

- Arithmetic and logic unit (ALU),
- Flag
- Control unit,
- Clock and Timing,
- Register sets,
  - accumulator,
  - condition code register,
  - program counter,
  - stack pointer,
  - segment registers



# **İkili Sayı Sistemi**

# Veri

- Veri: Anlam kazanmamış, ilişkilendirilmemiş, özümsememiş kısaca işlenmemiş olgu veya bilgi parçalarıdır.
- Veri: Ses, Metin, Görüntü, Resim. Veri semboller ile temsil edilir.
- Bilgisayarda, tüm işlevler 1/0 ikili sayı sisteminde yapılır. Bit: 1 ve 0 lardan oluşan dizi var.
- Sinyal (Akustik, Elektriksel, Elektromanyetik, ısı, titreşim) mesaj taşır, değişimi temsil eder, matematiksel fonksiyonlar ile ifade edilir.



# Bilgisayar Sistemlerinde Veri Manipülasyonu

- Veri Giriş ve Çıkış
- Veri Transferi
- Veri Saklama
- Veri İşleme:
  - Aritmetik (Toplama, Öteleme),
  - Karşılaştırma ( $>$ ,  $<$ ,  $=$ ,  $\neq$ ,  $\geq$ ,  $\leq$ ),
  - Mantıksal (AND, OR, NOT, XOR, XNOR);
  - Kontrol işlemleri, kesme, ...

# Number Systems

- **Sayı sistemi:** Onlu(10:0,1,2, .. ,9), ikili (bit: 0/1), sekizli (Oktal: 3bit), onaltılı(Hexadesimal: 4bit), Karakter(ASCII: 8bit)
- **Onlu sayı sistemi:** pozitif, negatif, kompleks; kesirli, tam sayı
- **Bit:** bir bilgisayardaki en küçük bilgi birimidir. 4 bit hex ile temsil edilir.
- **Byte:** 8 bit'lik bellek gözünün işaret eder, bellek kapasitesi byte ile belirtilir,  $2^n$  byte. Burada n, adres bus hat sayısıdır.
- **Data boyutu (data bus hat sayısı):** Bir clock periyodunda bellek gözlerine yazılacak bit sayısıdır. Nibble (4 bit), Byte (8bit), Word(16 bit), long Word(32bit), Quad Word(64 bit), Double Quad Word (128 bit)
- **Bit/sec:** 1 saniyede transfer edilen bit sayısı ya da 1 saniyede işlenen bit sayısı,  $10^3$ ,  $10^6$ , ...
- **Transistör:** Klasik bilgisayarlarda bit transistörlerde elektrik sinyal ile temsil edilir.
- **Qubit:** Quantum bilgisayarda temel veri birimi
- **Elektron:** Quantum bilgisayarlarda qubit elektron ile temsil edilir.

# İKİLİ (BINARY) SAYI SİSTEMİ

- Bilgisayarlar sadece 0 ve 1 rakamlarından oluşan ikili sayı sistemini kullanırlar. Algılanan sinyaller analog sinyallere dönüştürülür. Analog sinyaller sayısal sinyallere (ikil sayı sistemi) dönüştürülür.
- Bit ismi Binary Digit, ikilik rakamdan türetilmiştir.
- 8 bite bir byte denir.
- 1 varlığı temsil eder, 0 ise yokluğu temsil eder. 1 gerilim var (5Volt), 0 ise gerilim yok demektir.
- Klavyeden yazılan rakamlar ve harfler bilgisayarın anlayacağı şekle çevrilirken ikili sayı sistemi devreye girer (ASCII).
- Binary sayı sistemi, Türkçesi ikili sayı sistemi anlamına gelen kavramdır. İkili sayılar 2 tabanında yazılarak elde edilir. Dolayısı ile ikilik sistemdeki tüm sayılar 1 ve 0'dan ibarettir.
- İkili sayı sistemi, günümüz bilgisayarlarının neredeyse tamamında kullanılmaktadır.
- Günlük hayatımızda kullandığımız rakamlar ise onluk tabanda, bir başka isimle decimaldir.
- Decimal sistemi oluşturan rakamlar bildiğimiz gibi 0,1,2,3,4,5,6,7,8,9 şeklindedir.

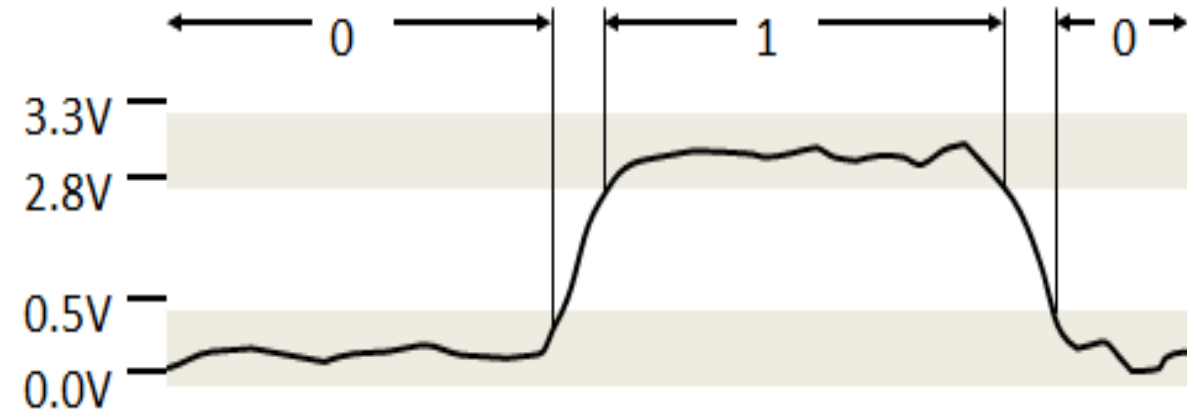
# İkili Sinyal- Bit

**İkili sinyal** (Binary signal), iki durum sinyali: İki durumlu veriler (0/1).

- off & on
- Elektriksel sinyaller ile taşınır, saklanır. low voltage & high voltage; 0v & 5v

**Bit:** Sadece matematiksel bir kavram değil, aynı zamanda fiziksel dünyada karşılığı vardır.

- İkili sayı sistemi 0 veya 1 değerine sahip olur ve başka hiçbir şey olmaz.
- Bir bit, bir bilgisayardaki en küçük veri birimidir

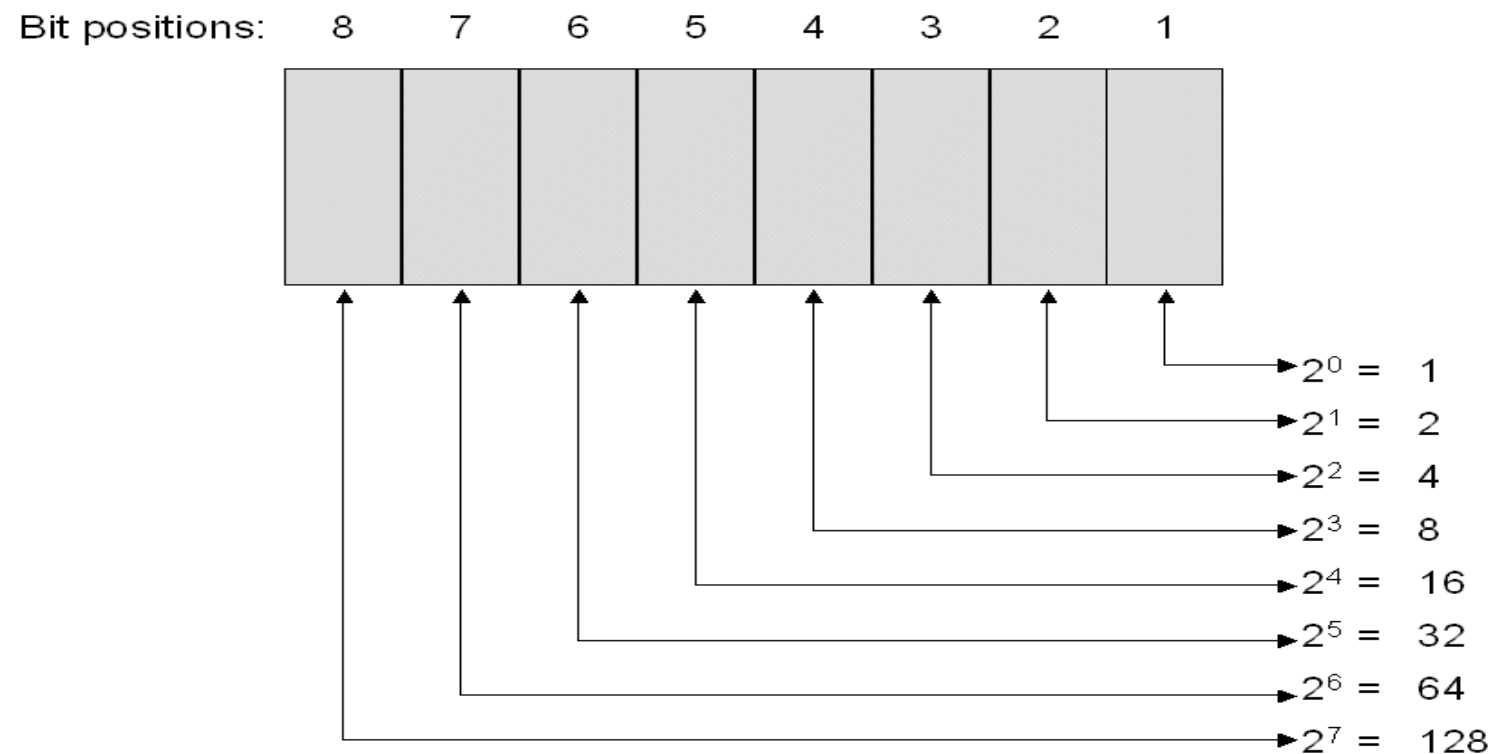
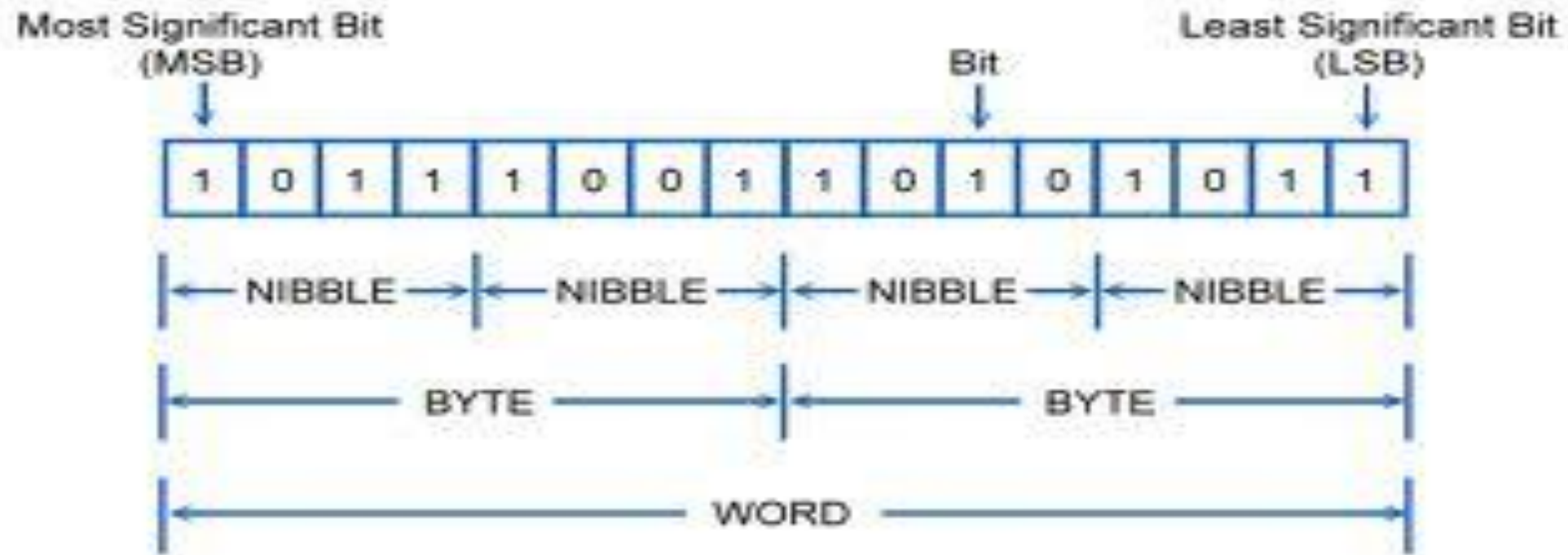


# Bits and Bytes are Slightly Different

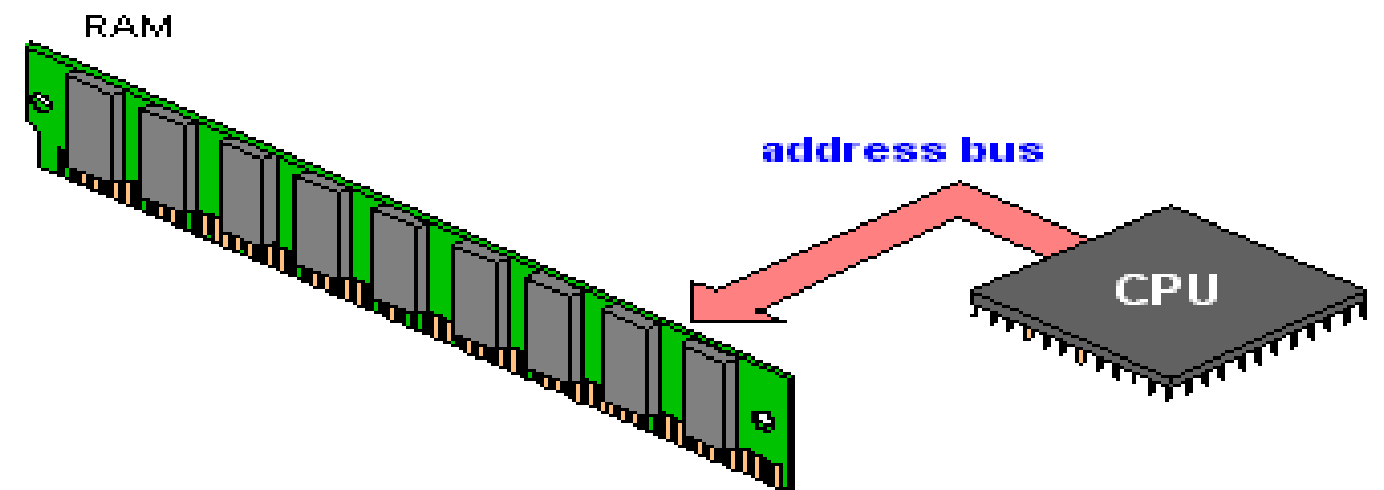
“Kilo” or “Mega” have slightly different values when used with bits per second or with bytes.

- When Referring to Bytes (as in computer memory)
  - Kilobyte (KB)  $2^{10} = 1,024$  bytes
  - Megabyte (MB)  $2^{20} = 1,048,576$  bytes
  - Gigabyte (GB)  $2^{30} = 1,073,741,824$  bytes
  - Terabyte (TB)  $2^{40} = 1,099,511,627,776$  bytes
- When Referring to Bits Per Second (as in transmission rates): Veri işleme ya da veri transfer hızıdır.
  - Kilobit per second (Kbps) = 1000 bps (thousand)
  - Megabit per second (Mbps) = 1,000,000 bps (million)
  - Gigabit per second (Gbps) = 1,000,000,000 bps (billion)
  - Terabit per second (Tbps) = 1,000,000,000,000 bps (trillion)

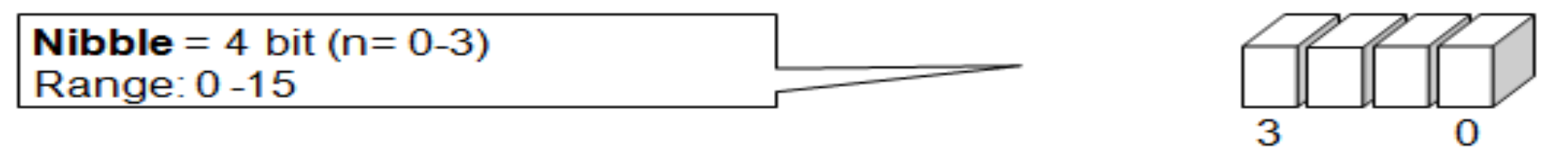

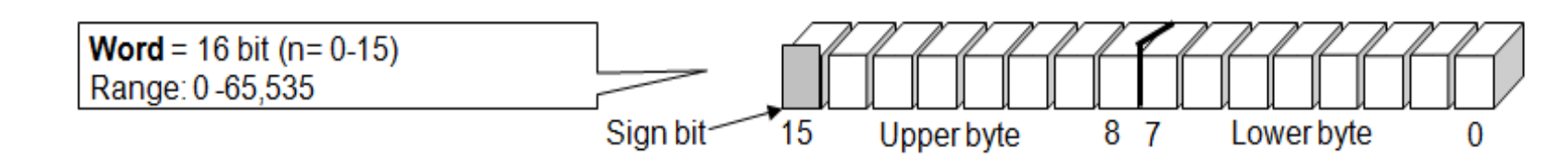
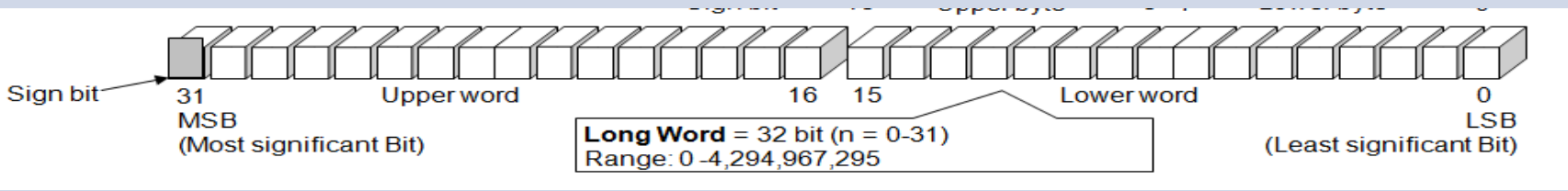
# Bits and Bytes



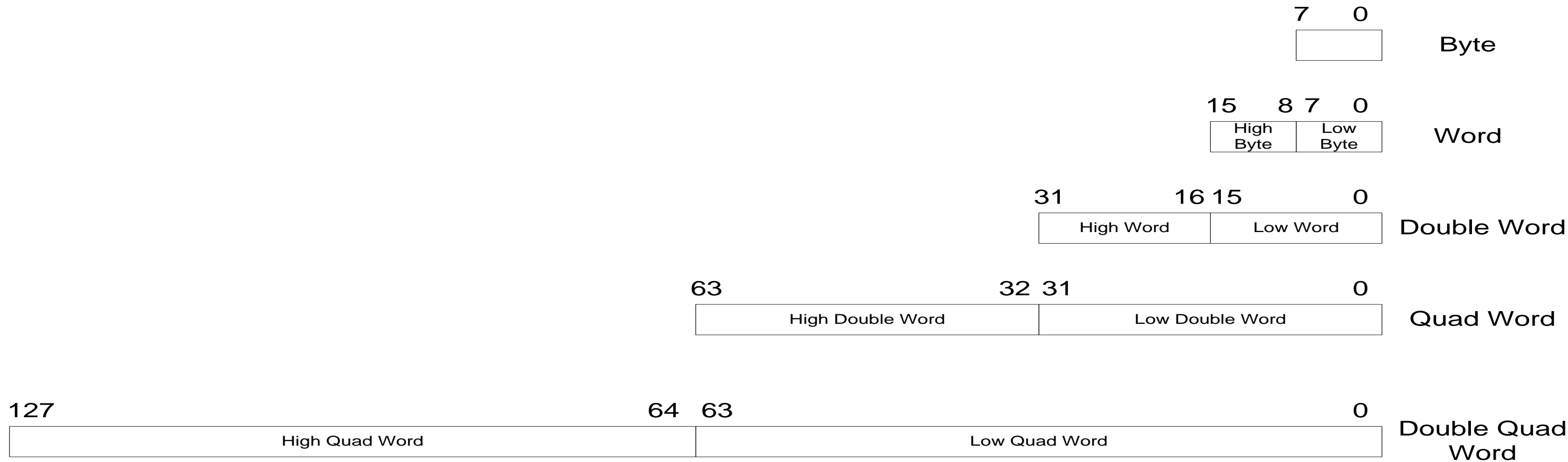
From Computer Desktop Encyclopedia  
© 2004 The Computer Language Co. Inc.



# DATA SIZE

<p><b>Nibble</b></p>	<p><b>4 bit</b></p>	<p><b>Nibble</b> = 4 bit (n= 0-3) Range: 0 -15</p> 
<p><b>Byte</b></p>	<p><b>8 bit</b></p>	<p><b>Byte</b> = 8 bit (n = 0-7) Range: 0 -255</p> 
<p><b>Word</b></p>	<p><b>16 bit</b></p>	<p><b>Word</b> = 16 bit (n= 0-15) Range: 0 -65,535</p> 
<p><b>Long word</b></p>	<p><b>32 bit</b></p>	<p><b>Long Word</b> = 32 bit (n = 0-31) Range: 0 -4,294,967,295</p> 

# Data Types - size



Always take care of the type of data an instruction accesses!!!!



# Decimal / Binary / Hexidecimal

Decimal	Binary	Hexidecimal
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

## Conversions:

1000 1110 (binary)

8 E (hex)

## Notations for hex:

- 0x8E
- 8Eh
- 8E<sub>16</sub>
- 0FEh

# Bits as ASCII Codes

- ASCII: Her karakter benzersiz bir 8 bitlik kod ile temsil edilir.
- Özel karakterler için 256 benzersiz kod
- Klavyedeki her tuşun 8 bitlik bir karşılığı vardır.
- Her harf, rakam ve özel karakterleri temsil eder.

## ASCII Character Codes

Hex	Char	Hex	Char	Hex	Char	Hex	Char	Hex	Char	Hex	Char
20	(Space)	30	0	40	@	50	P	60	`	70	p
21	!	31	1	41	A	51	Q	61	a	71	q
22	"	32	2	42	B	52	R	62	b	72	r
23	#	33	3	43	C	53	S	63	c	73	s
24	\$	34	4	44	D	54	T	64	d	74	t
25	%	35	5	45	E	55	U	65	e	75	u
26	&	36	6	46	F	56	V	66	f	76	v
27	'	37	7	47	G	57	W	67	g	77	w
28	(	38	8	48	H	58	X	68	h	78	x
29	)	39	9	49	I	59	Y	69	i	79	y
2A	*	3A	:	4A	J	5A	Z	6A	j	7A	z
2B	+	3B	;	4B	K	5B	[	6B	k	7B	{
2C	,	3C	<	4C	L	5C	\	6C	l	7C	
2D	-	3D	=	4D	M	5D	]	6D	m	7D	}
2E	.	3E	>	4E	N	5E	^	6E	n	7E	~
2F	/	3F	?	4F	O	5F	_	6F	o	7F	DEL

# Quantum Mekanikinin Temel Prensipleri

Quantum mekaniğin temeli atom altı parçacıklarının davranışlarının öngörülmesi fikrine dayanmaktadır. 1901 ile 1932 tarihleri arasında quantumun 4 temel prensibi çalışılmıştır:

- Birincisi enerjinin parçacıklar halinde bulunması
  - Quantum mekaniğinin ikinci prensibi enerji aynı zamanda hem parçacık ve hemde dalgadır.
  - Quantum mekaniğinin üçüncü prensibi ise parçacığın ne olduğunun anlaşılabilmesi için bir olasılıklar bütününden bahsedilmesi gerekmektedir.
  - Quantum mekaniğinin dördüncü prensibi ise Heisenberg'in belirsizlik prensibine göre bir parçacığın konumunun ve hızının tespit edilememesidir.
- Quantum hesaplamada kullanılan fiziksel özellikler: Quantum süperpozisyonu, Quantum dolanıklığı, Belirsizlik ilkesi, Dalga karışması
  - **Dolanıklık:** Elektron ve foton birbirlerine dolanıktır, **Süperpozisyon:** Aynı anda paralel durumlarda olma olasılığı, **Çökme:** Ölçme yapıldığında paralel olma durumlarından birine çöker.

# Quantum lojik kapısı bir matristir

- Quantum lojik kapısı bir matristir.
- Bra (  $\langle l$  ) satır vektör; Ket (  $| l \rangle$  ) sütun vektör.
- $|\psi\rangle$  ve  $|\varphi\rangle$  birer durum vektörü olmak üzere  $A|\psi\rangle=|\varphi\rangle$  olur, burada  $A$  matrisi bir lojik quantum kapısı olmak üzere,  $|\psi\rangle$  ve  $|\varphi\rangle$  kompleks vektör uzayında birer baz vektördürler.
- Qubit olarak tanımlanan  $|0\rangle$  ve  $|1\rangle$ , quantum durumları kompleks vektör uzayında birer sütun vektör ile gösterilir.  $|0\rangle=\begin{pmatrix} 1 \\ 0 \end{pmatrix}$  ve  $|1\rangle=\begin{pmatrix} 0 \\ 1 \end{pmatrix}$ ; Bunlara base ketler denir.



# What is a qubit?

- A bit has two possible states  $|0\rangle$  or  $|1\rangle$
- Unlike bits, a qubit can be in a **state other than**  
 $|0\rangle$  or  $|1\rangle$
- We can form linear combinations of states  
 $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$
- A qubit state is a unit vector in a two-dimensional *complex vector space*

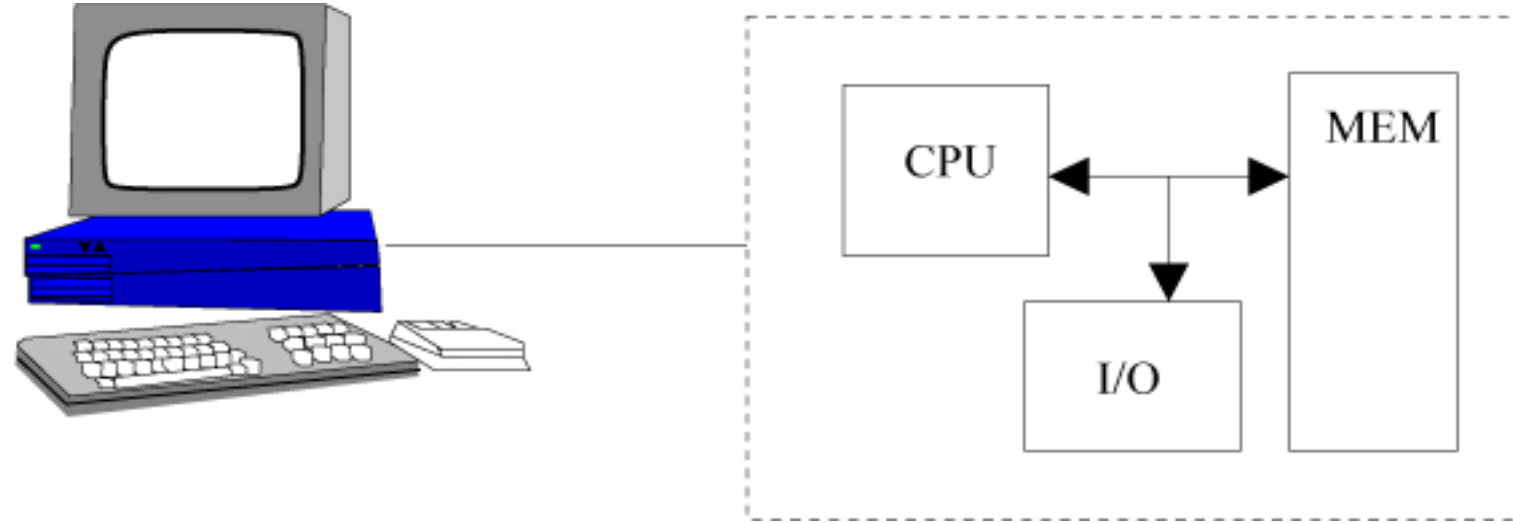
# Dirac Gösterimi (The Dirac Notation)

- Quatum temel prensipleri:
- Quantum hesaplama ile birlikte qubit, Dirac tarafından geliştirilen bir gösterimle karşılanabilmektedir. Bra-ket olarak da geçer.
- $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$  gösteriminde, ket  $\psi$  değeri,  $|\alpha|^2$  değeri kadar ket 0 ve  $|\beta|^2$  değeri kadar ket 1'dir. **Süperpozisyon söz konusudur: Aynı anda ket 0 ya da ket 1 olma ihtimali.** Ket  $\psi$ , iki değer arasında bir yerde kabul edilen bir sütun vektörüdür. Olasılığın temel yasası gereği  $|\alpha|^2 + |\beta|^2 = 1$  olmalıdır.
- Bra-ket gösterimi  $\langle | \rangle$  şeklinde sembolize edilebilir. Buradaki **bra** kısmı  $\langle |$  olurken **ket** kısmı  $| \rangle$  olmuş olur. Yani İngilizcedeki parantez anlamına yakın bir kelimeyi parçalara bölerek (aslında barcket kelimesi, İngilizcede parantez anlamına gelir), parantez iki alt parçada gösterilir.
- $\langle |$  , bra gösterimi ulaşmak istediğimiz hali, veya beklediğimiz durumu göstermeye yarar. Örneğin  $\langle x=1.5 |$  gösterimi bize, parçacığın, 1.5 konumunda bitmesini istediğimizi veya böyle bir beklentimiz olduğunu gösterir. Bu durumda, örneğin  $\langle x=1.5 | x=3 \rangle$  gösterimi, parçacığın 3 konumunda başlayarak 1.5 konumunda bitmesi anlamına gelir.



# **Bilgisayar Temel Bileşenleri**

# Bilgisayar Temel Bileşenleri



## Bilgisayar Sisteminin Temel Bileşenleri:

- CPU – Central Processing Unit (Mekezi İşlemci Birimi - Mikroişlemci)
- Memory: Yazılıp okunacak verileri ve programları saklar.
- Input and output (I/O) unit
- System Bus: data, address and control.
- Clock & Timing
- Yazılım

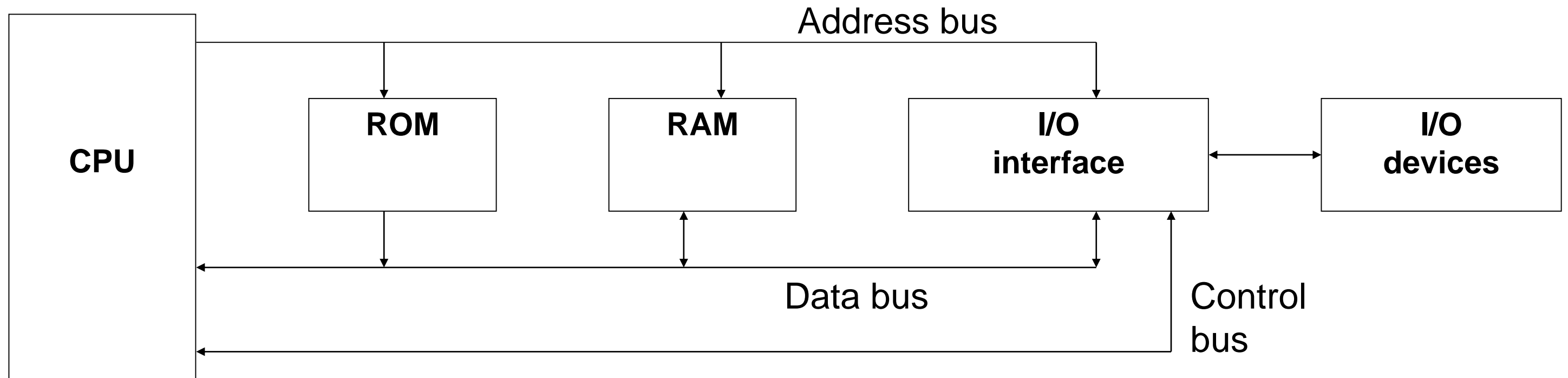
## Bir Bilgisayar Sisteminin İşlevleri:

- İşlemci Ana İşlev Döngüsü (Fetching and execution cycles)
- Address Decoding: Bellek gözlerinin çakışmasını ya da üst üste gelmesini önlemek amacıyla bellek seçer.
- Clock and Timing Signals: Verilerin senkronize (eş zamanlı) işlenmesini sağlar.
- Pipelining – Verimi artırmak amacıyla, komutlar dizisinin ardışık düzenlendiği ve eşzamanlı yürütüldüğü işlevlerdir.



# Block diagram of a basic microcomputer

Basic computer system consist of a Central processing unit (CPU), memory (RAM and ROM), input/output (I/O) unit.



**Block diagram of a basic computer system**

# A Computer System

## ○ Bir bilgisayar sisteminin temel bileşenleri

- CPU – Central Processing Unit (Mekezi İşlemci Birimi - Mikroişlemci)
- Memory
- Input and output unit
- System Bus: data bus, address bus and control bus.
- Clock and Timing signals

## ○ Evolution of microprocessor address lines: 8,16, 20, 32 dan 64 bit; Adres hatları bellek, bellek gözü veya I/O birimi seçmede kullanılır. Paralel hatlardır. Grup olarak çalışırlar. Üzerlerinde bitleri (1 / 0) temsil eden elektrik sinyalleri taşınır.

- Toplam bellek erişim kapasite=  $2^n$  byte olarak hesaplanır. Buradaki n ifadesi belleğin adres hattı sayısıdır.
- Örneğin bir belleğin adres hattı sayısı,  $n=30$  ise bilgisayar sistemin adresleme kapasitesi= $2^{30}$  byte =1 GByte'dır.

## ○ Data lines: Nibble, byte, word ... Longword. Belleklerin gözlerine ve I/O birimlerinin register ya da bellek gözlerine veri yazma ve okumada kullanılır. Paralel hatlardır. Grup olarak çalışırlar. Üzerlerinde 1 ya da 0'ı temsil eden elektrik sinyalleri taşınır. 1 clock periyodunda yazılan ya da okunan bit sayısını temsil eder.

- Örneğin data bus hat sayısı 64 bit ise, 1 clock periyodunda yazılacak ya da okunacak bit sayısı=64 bit=64/8=8 byte'dır.

# Basic components of Computer

## 1. CPU - Central Processing Unit

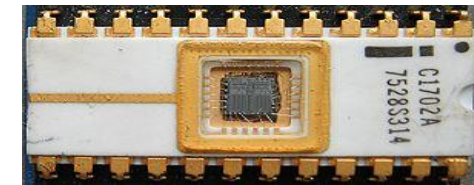
- Bilgisayar sistemindeki bir bilgisayar programının komutlarını yerine getiren kısmıdır.
- Bilgisayarın işlevlerini yerine getiren birincil ögedir.
- Program komutlarını okuyan, yürüten ve yazan birimdir.
- Komut veriler (Program) işlemciye ne yapması gerektiğini söyler.



Pentium D dual core processors

## 2. Memory

- Verileri veya programları (talimat dizileri) geçici veya kalıcı olarak depolamak için kullanılan fiziksel cihazlar.
- Bilgisayar ana belleği iki ana çeşitte gelir: rasgele erişimli bellek (RAM) ve salt okunur bellek (ROM); ayrıca pilli CMOS belleklerde kullanılır..
- RAM, CPU komut verdiği her an okunabilir ve yazılabilir, ancak ROM, hiçbir zaman değişmeyen veri ve yazılımla önceden yüklenmiştir, bu nedenle CPU yalnızca ondan okuyabilir.
- ROM, genellikle bilgisayarın ilk başlatma talimatlarını depolamak için kullanılır.
- Genel olarak, bilgisayarın gücü kapatıldığında RAM içeriği silinir, ancak ROM verilerini süresiz olarak tutar.
- Bir bilgisayarda, ROM, bilgisayar açıldığında veya sıfırlandığında bilgisayarın işletim sisteminin sabit disk sürücüsünden RAM'e yüklenmesini düzenleyen BIOS adlı özel bir program içerir.
- CMOS pil içeren bellekler



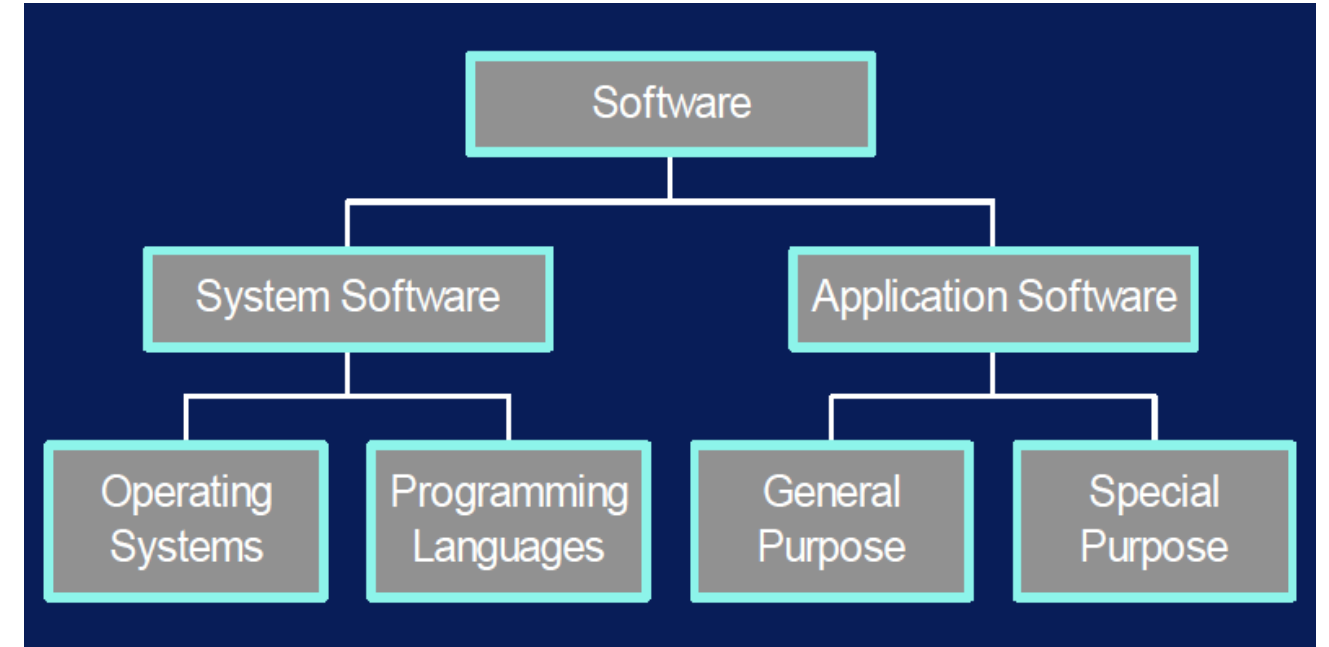
# 3. I/O Unit

- Girdi/çıkıtı (I/O), bir bilgi işleme sistemi (bilgisayar gibi) ile dış dünya, muhtemelen bir insan veya başka bir bilgi işleme sistemi arasındaki iletişimi ifade eder.
- Girişler, sistem tarafından alınan sinyaller veya verilerdir ve çıkışlar, sistemden gönderilen sinyaller veya verilerdir.
- Bilgisayara giriş veya çıkış sağlayan aygıtlara çevre birimleri denir.
- Tipik bir kişisel bilgisayarda çevre birimleri, klavye ve fare gibi giriş aygıtlarını ve ekran ve yazıcı gibi çıktı aygıtlarını içerir.
- Sabit disk sürücüleri, disket sürücüleri ve optik disk sürücüleri, hem giriş hem de çıkış aygıtı olarak işlev görür.
- Bilgisayar ağı, başka bir G/Ç biçimidir.

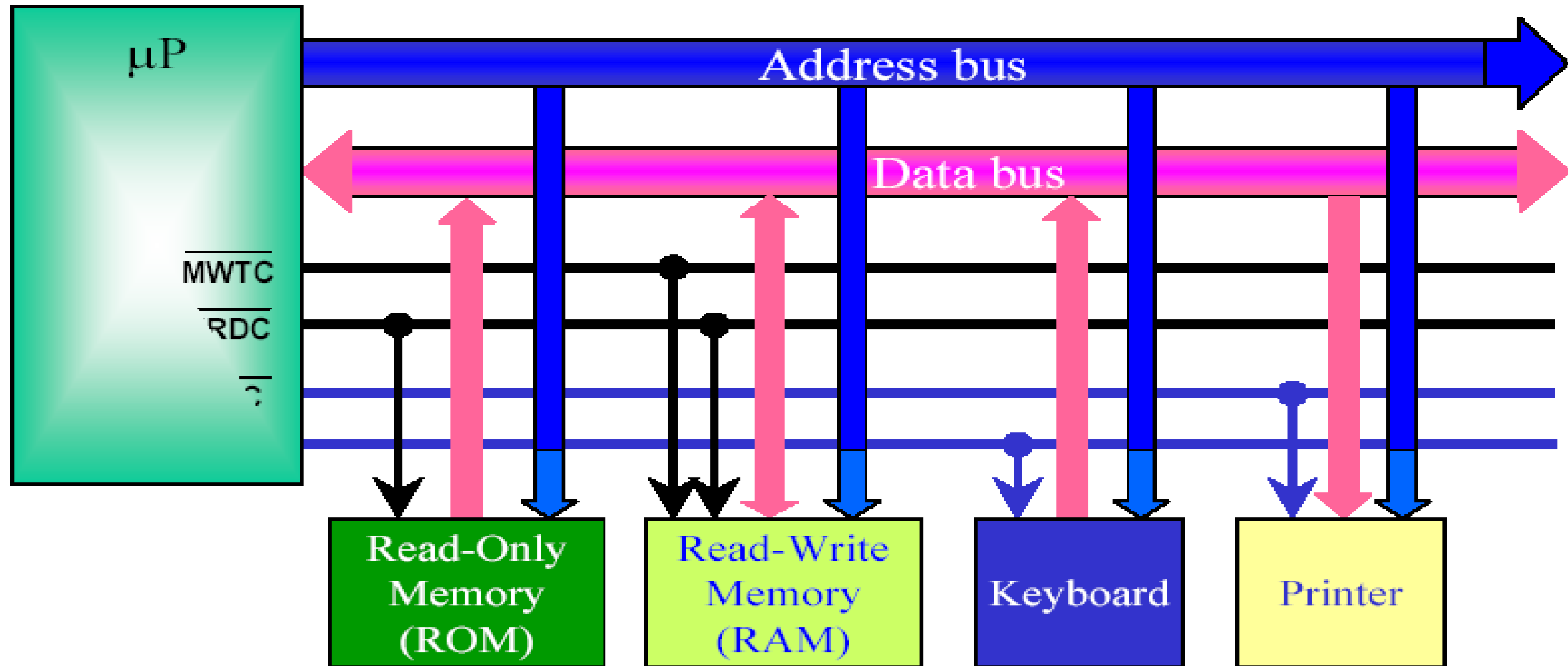
# 4. Software

## Operating Systems:

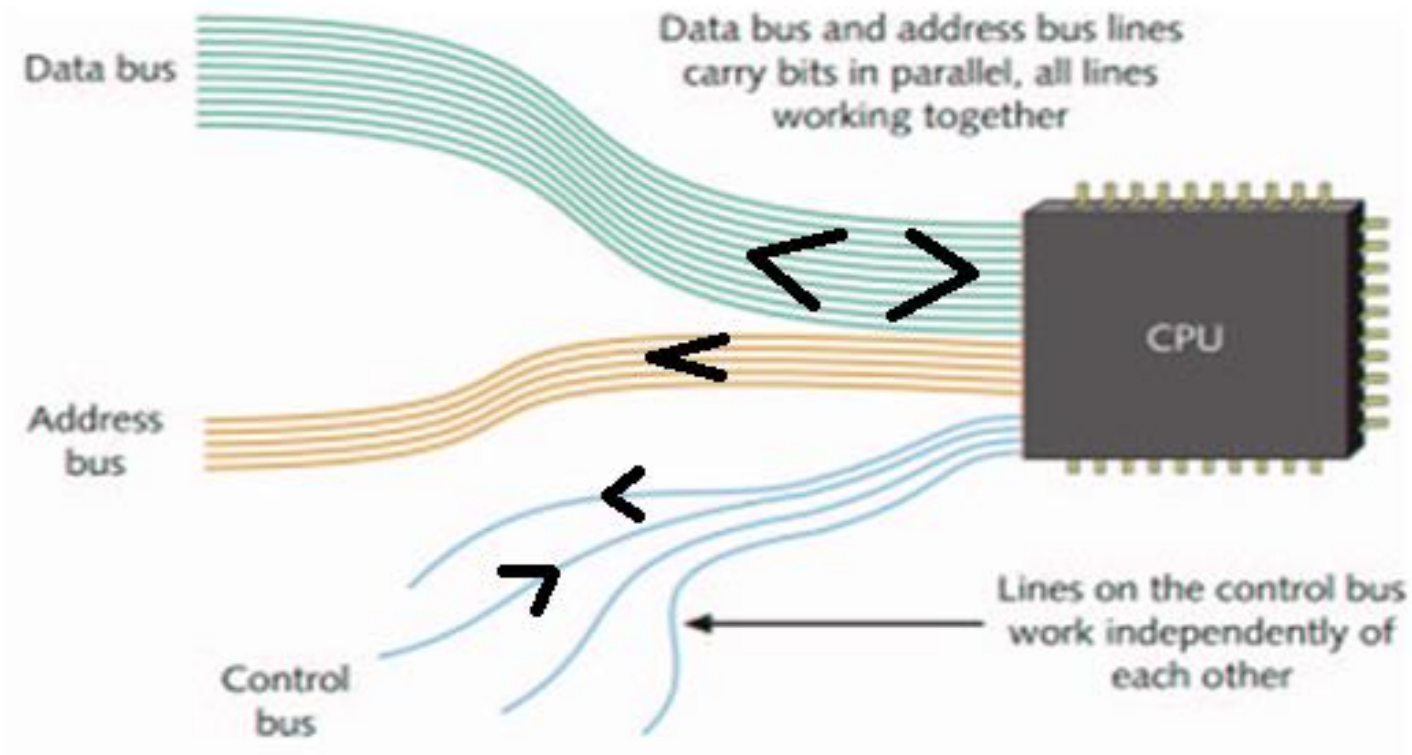
- İşletim Sistemi, bilgisayar sisteminin genel işleyişini yöneten yazılımdır.
- Temel amaç uygulama programlarını desteklemektir.
- Uygulama programlarından cihazların ayrıntıları gizlenir.
- DOS (Disk İşletim Sistemi) Windows: Intel 80X86 işlemciler için tasarlandı
- Macintosh OS: Motorola ve PowerPC işlemciler için tasarlanmıştır.
- Kullanıcı arayüzleri: Programı kullanan kişiden (kullanıcı) bilgi iletmekten sorumlu yazılım. Bilgisayarla iletişim kurar ve bilgisayarı kontrol eder. Üç tür kullanıcı arayüzü:
  - Graphic user interfaces
  - Menu driven interfaces
  - Command driven interfaces



# Block Diagram of a Microprocessor-based computer system



# System Bus Components



- System Bus – wires connecting memory & I/O to microprocessor
  - Address Bus
    - Unidirectional
    - Identifying peripheral or memory location
  - Data Bus
    - Bidirectional
    - Transferring data
  - Control Bus
    - Synchronization signals
    - Timing signals
    - Control signal

Adres Bus ve Data Bus, grup olarak çalışırlar. Üzerlerinde bitleri (1 / 0) temsil eden elektrik sinyalleri taşınır.



# Sistem Bus

**Sistem Bus:** Bir bilgisayar sisteminde CPU ile diğer temel bileşenler (CPU'nun ve Bilgisayar sisteminin) arasında haberleşmenin organize edildiği hatlardır. Bus is data path. Adres Bus ve Data Bus, grup olarak çalışırlar. Üzerlerinde bitleri (1 / 0) temsil eden elektrik sinyalleri taşınır. Bus: Herbir hat üzerinde 0 /1 (bit) bulunur. Elektriksel sinyal ile temsil edilir. Dijital sayı sistemidir.

## ○ Adres Bus:

- Bir bilgisayar sisteminde bellek ve bellek gözü veya I/O birimini seçmede kullanılan tek yönlü paralel hatlardır. Grup olarak çalışırlar. Adres Bus tek yönlüdür. Hat sayısı belirlemede: bellek sayısı, bellek kapasitesi ve I/O sayısı göz önüne alınır.

## ○ Data Bus:

- Bellekler ve I/O birimlerine veri yazmada ya da okumada kullanılan iki yönlü paralel hatlardır. Grup olarak çalışırlar. Data bus hat sayısını belirlemede bir clock periyodunda yazılacak ya da okunacak bellek içeriği, I/O register boyutu ile belirlenir. (8,16,32,64,128, 256, ...)

## ○ Control Bus:

- CPU ile Bellek ya da I/O birimleri arasındaki iş akışını yönetir. Verileri kağınağından hedefe kadar kontrol ederler.
- Yaygın kullanılan Control Bus hatları: Yaz, Oku, Dur, Clock, Reset (Yeniden başlat), Kesme...
- Tek yönlüdür. Bir kısmı CPU'dan bellek veya I/O birimlerine; bir kısmı ise bellek veya I/O birimlerinden CPU'ya gider. Grup olarak çalışmazlar. Bireysel çalışırlar. Paralel hatlardır. Sinyalleri kontrol eder, zamanlama ve kontrol amaçlıdır. Control bus checks the data to the destination.



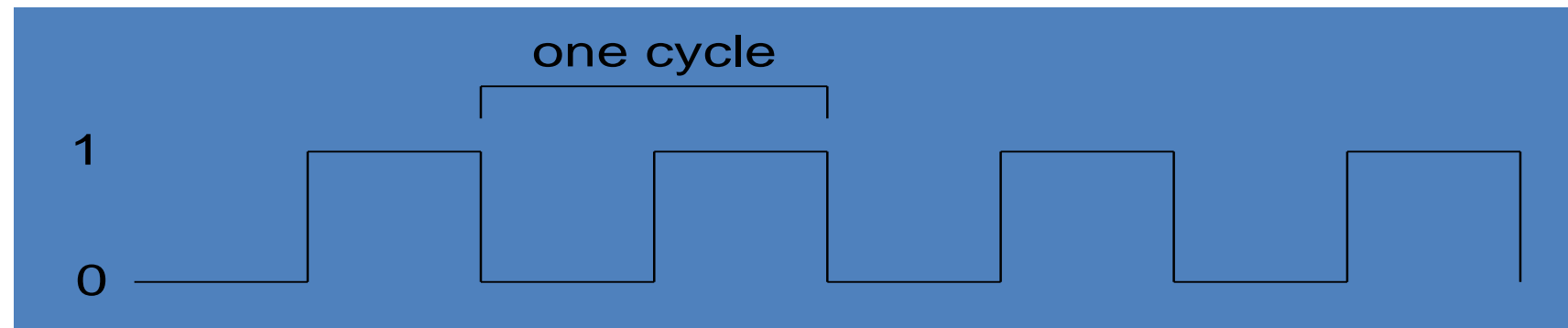
# **Clock & Timing**

# Microprocessor clock

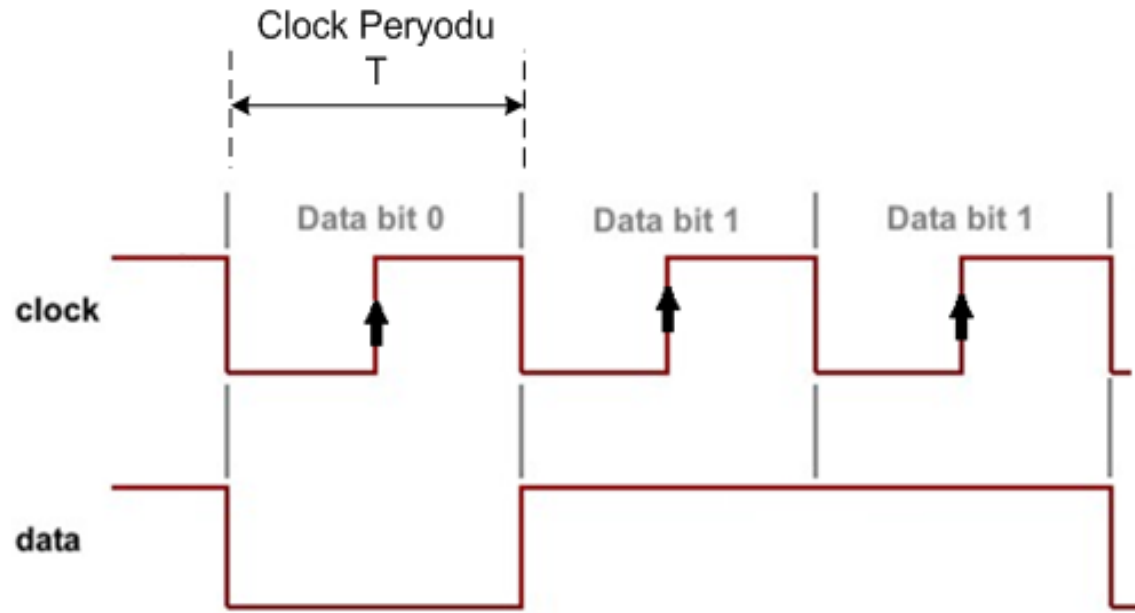
- Also called clock rate, the speed at which a microprocessor executes instructions. Every computer contains an internal clock that regulates the rate at which instructions are executed and synchronizes all the various computer components.
- The CPU requires a fixed number of clock ticks (or clock cycles) to execute each instruction. The faster the clock, the more instructions the CPU can execute per second. Clock speeds are expressed in megahertz (MHz) or gigahertz ((GHz).
- Some microprocessors are superscalar, which means that they can execute more than one instruction per clock cycle.
- Like CPUs, expansion buses also have clock speeds. Ideally, the CPU clock speed and the bus clock speed should be the same so that neither component slows down the other. In practice, the bus clock speed is often slower than the CPU clock speed, which creates a bottleneck. This is why new local buses, such as AGP, have been developed.

# Clock

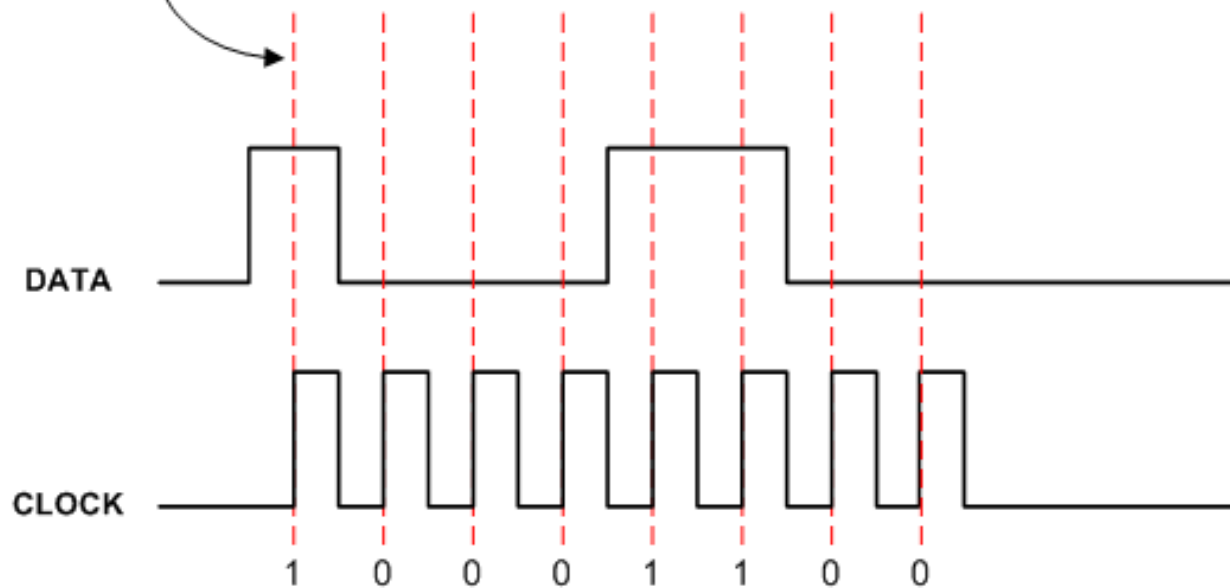
- synchronizes all CPU and BUS operations
- machine (clock) cycle measures time of a single operation
- clock is used to trigger events



# Clock and Data



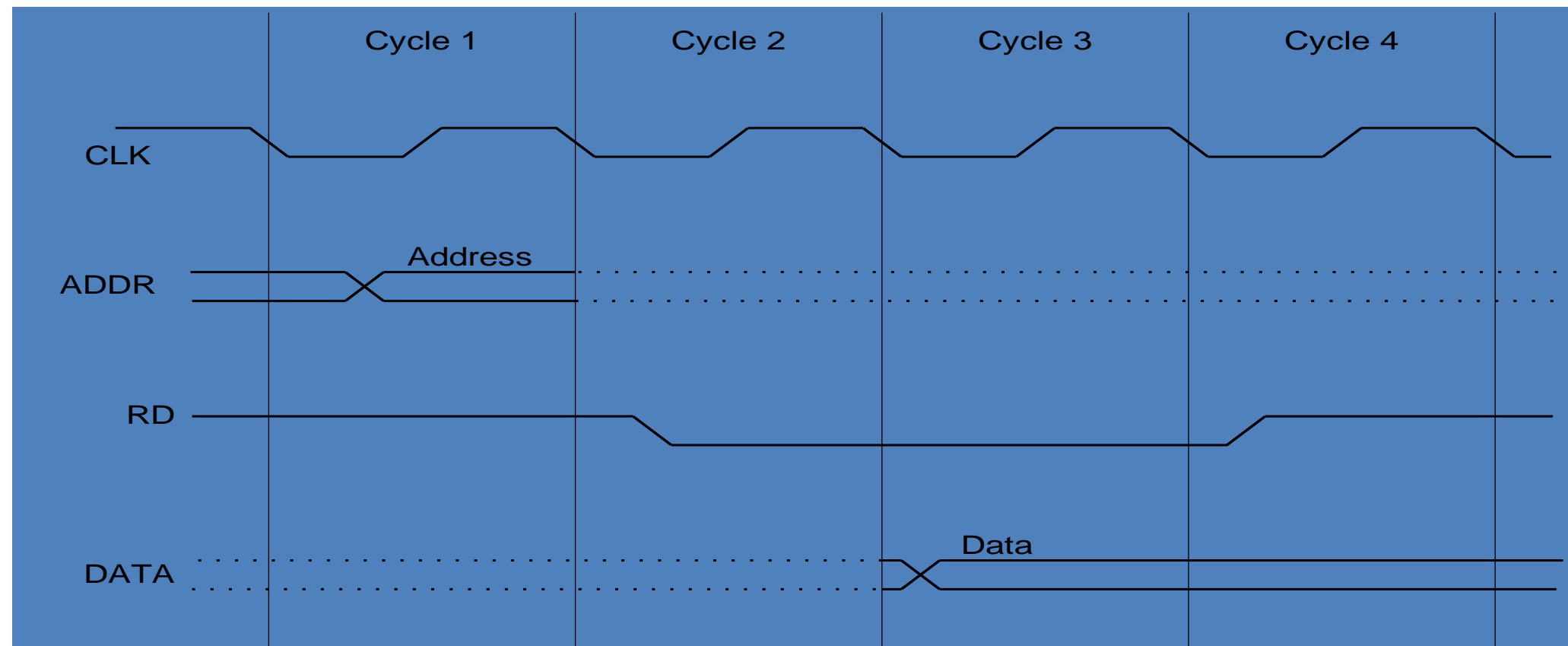
Data sampled on rising edge of clock



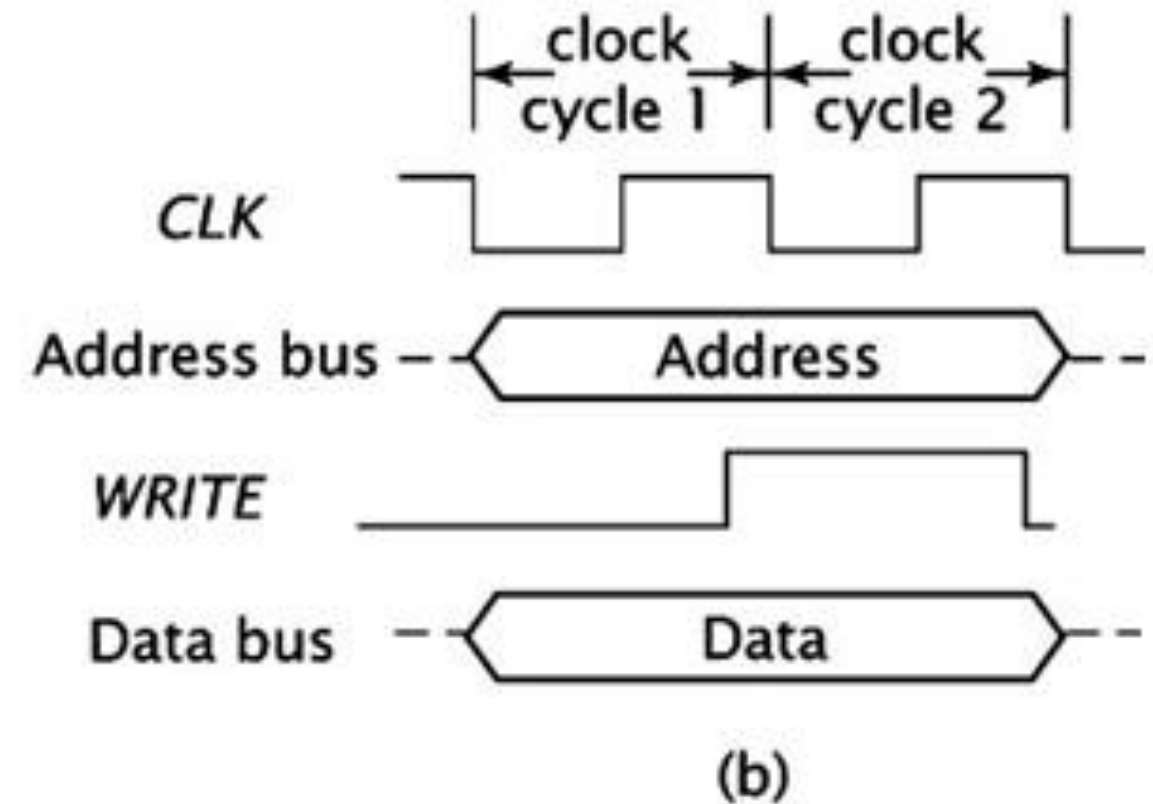
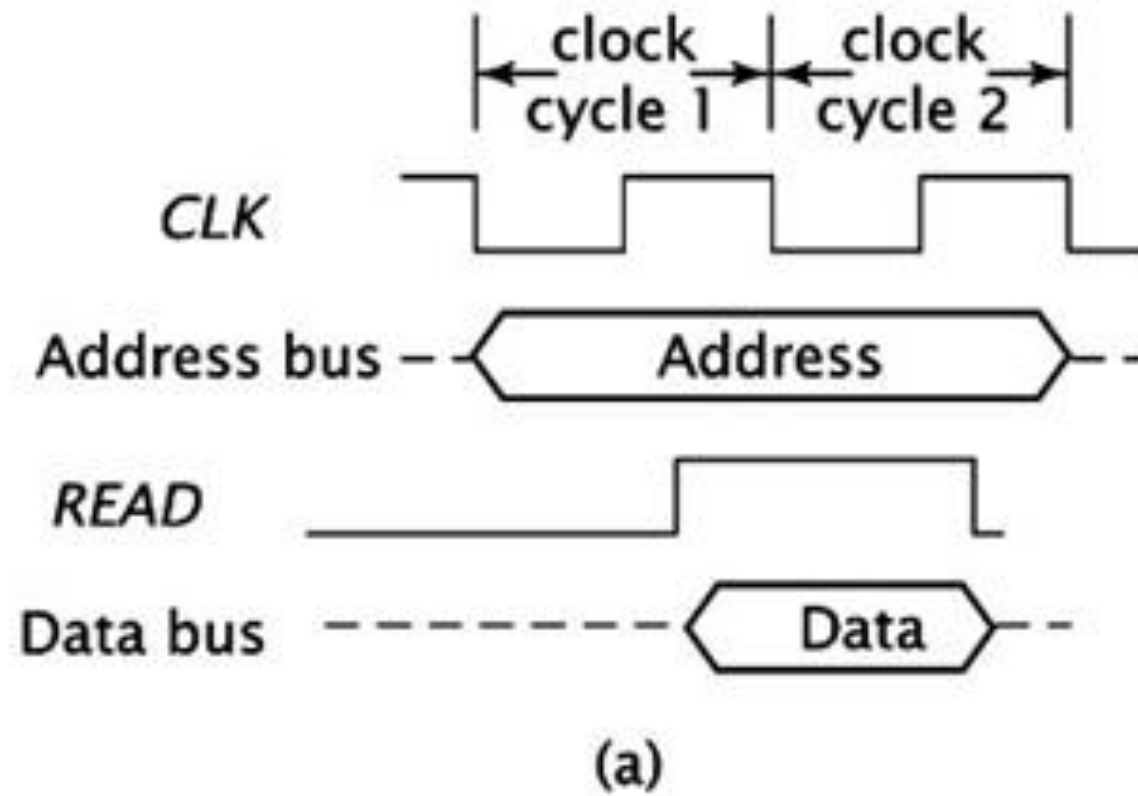
- Her bilgisayarın bir sistem clock darbe sinyali vardır.
- Sonsuzdan gelip sonsuza gider; süreklidir.
- Clock darbe sinyali 1 ve 0 lardan oluşan darbe katarıdır (Elektriksel sinyaldir).
- Clock periyodu, data uzunluğuna eşittir.
- Bit olarak tanımlanan verileri tetiklemede yani işlemede kullanılır.
- Kişisel bilgisayar hızları genellikle gigahertz (GHz) cinsinden ifade edilir.

# Reading from Memory

- Multiple machine cycles are required when reading from memory, because it responds much more slowly than the CPU. The steps are:
  - address placed on address bus
  - Read Line (RD) set low
  - CPU waits one cycle for memory to respond
  - Read Line (RD) goes to 1, indicating that the data is on the data bus

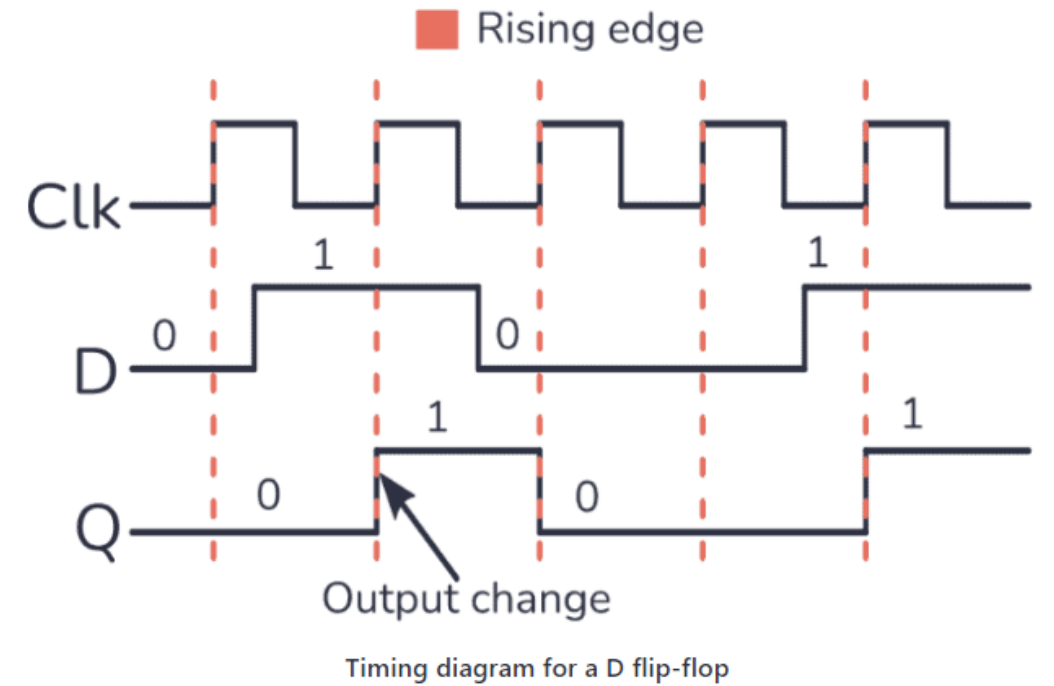
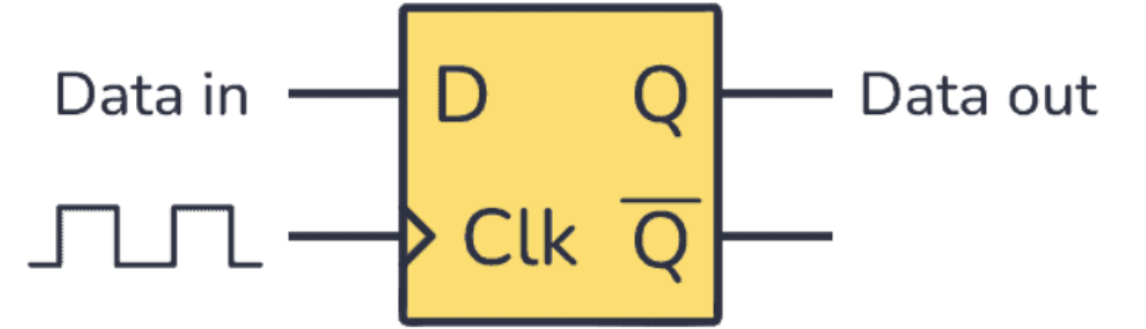


# Instruction Fetch (WR / RD)



# D Flip-Flop

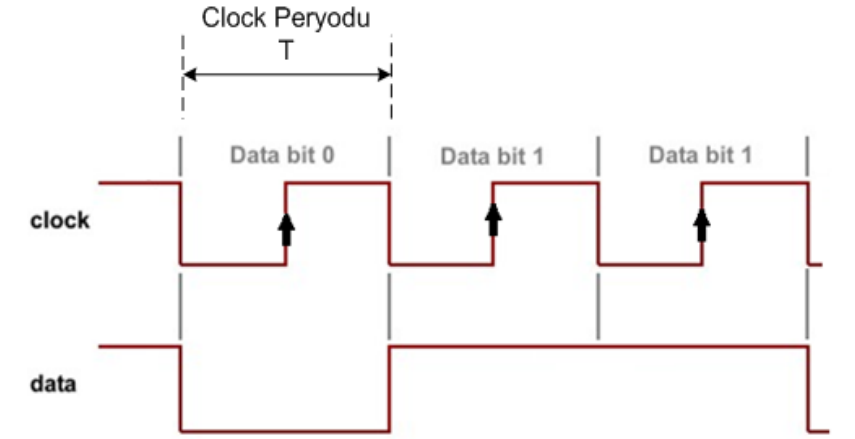
- D flip-flop bir clock sinyaline (Clk) ihtiyaç duyan senkron devrelerdir. D Flip-Flop, clock sinyali 0'dan 1'e (yükselen kenar: tetikleme) gittiğinde Q çıkışı D girişinden yeni bir değer depolayacaktır. Clock sinyali değişmediği sürece Q çıkışın durumu değişmez.





# Örnek:

- Clock sinyali: Yüksel ya da düşen kenar ile tetikleme
- $f=2\text{GHz}$ , Data bus hat sayısı=32 bit ise 1 saniyede belleğe yazılacak bit miktarını belirleyiniz.
- Burada,  $f$ =frekans, mikroişlemci çalışma frekansıdır. Birimi:  $\text{Hz}=1/\text{saniye}$
- Clock peryodu=clock cycle,  $T=1/f=1/2\text{GHz}=1/(2*10^9\text{Hz})$
- $T=0.5*10^{-9}$  saniye
- Bellek gözlerine  $0.5*10^{-9}$  saniyede 32 bit yazılırsa 1 saniye kaç bit yazılır?
- $X=32/(0.5*10^{-9})=16*10^9\text{bit/saniye}=16\text{ Gigabit/saniye}$

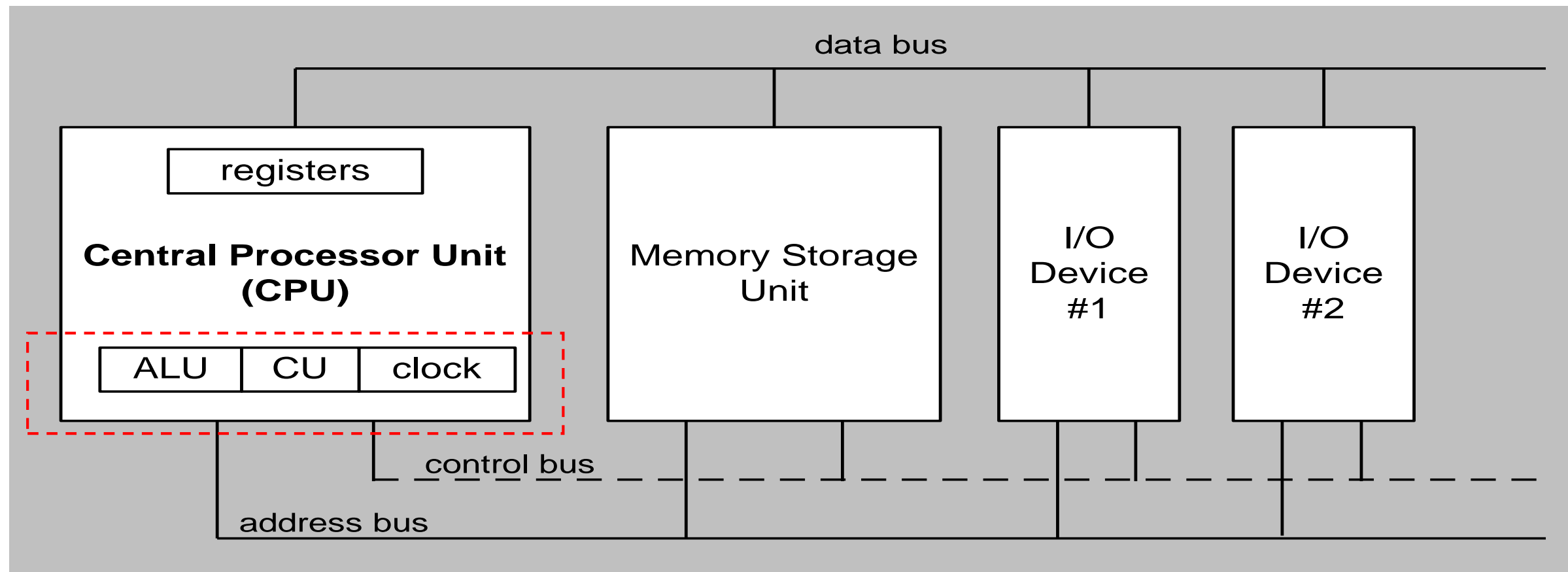




**Mikoişlemci**

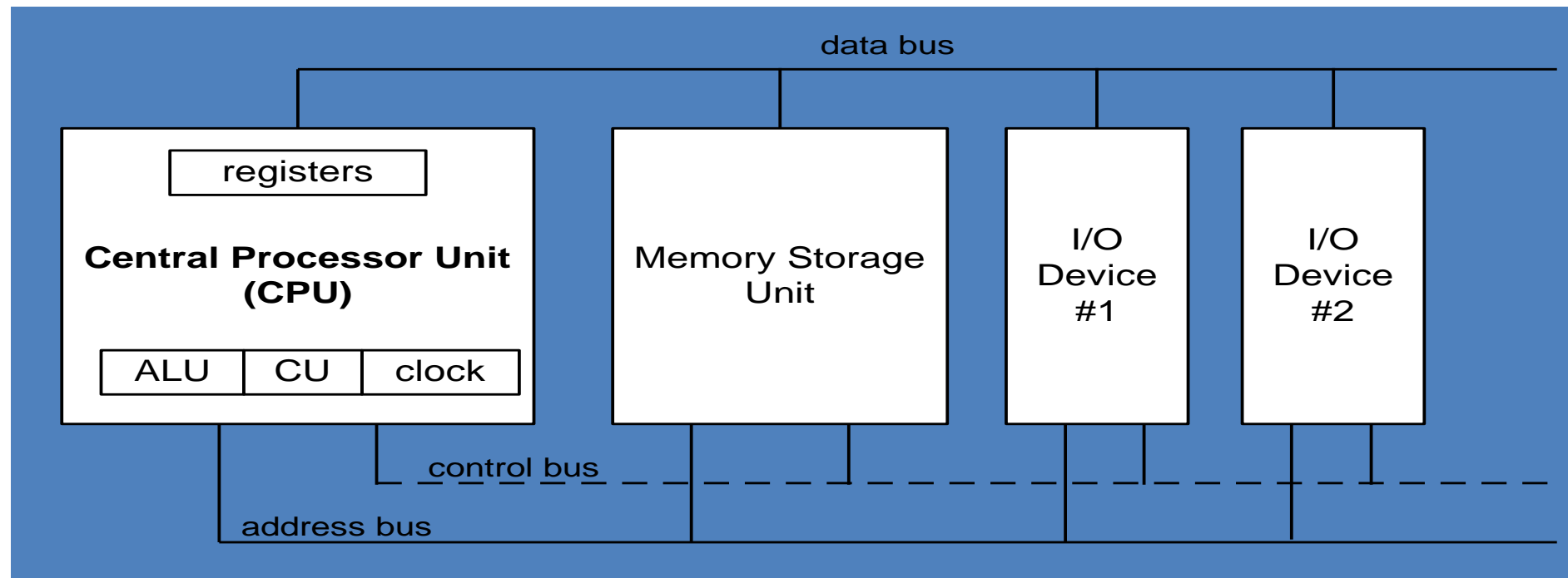
# CPU

- **control unit (CU)**: controls operation timing / sequencing
- **arithmetic logic unit (ALU)**: performs all math & logic operations, receives command sequence from CU
- **clock**: synchronizes CPU operation



# Basic Microcomputer Design

- clock synchronizes CPU operations
- control unit (CU) coordinates sequence of execution steps
- ALU performs arithmetic and bitwise processing



# The structure of a microprocessor

○ Arithmetic and logic unit

○ Control Unit

○ Clock

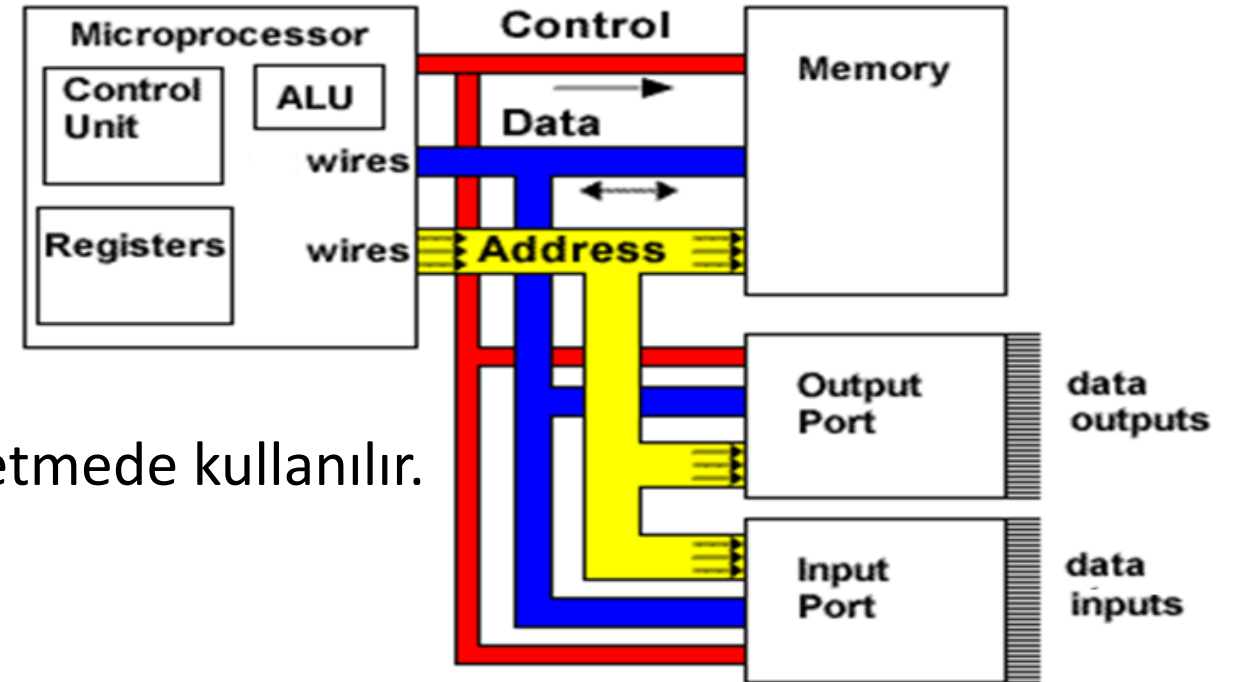
○ Registers:(Özel Amaçlı Saklayıcılar):

Özel amaçlı saklayıcıdır. Yüksek hızlı veri işlemde ve transfer etmede kullanılır.

Geçici depolama alanıdır. CPU içinde bulunur.

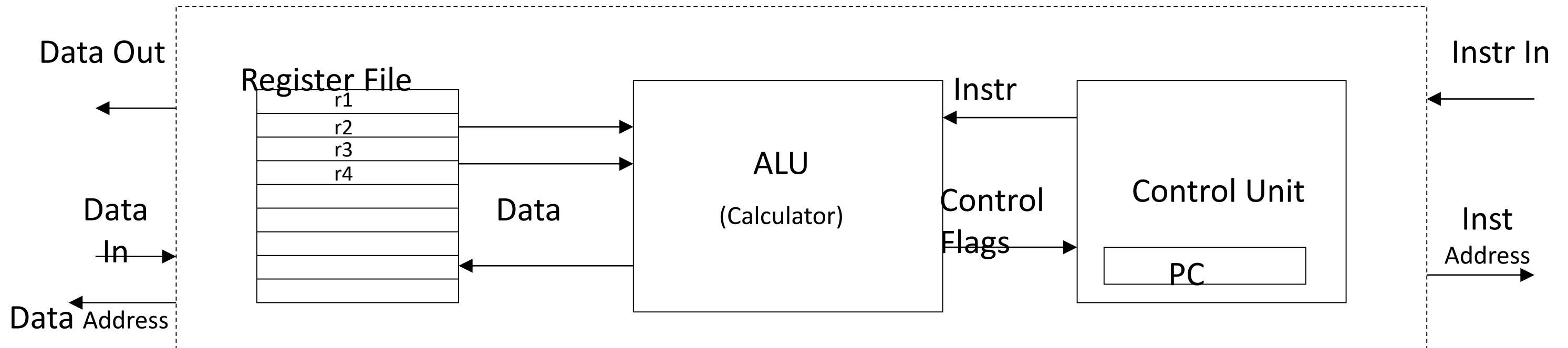
İşlenecek komutların transfer edildiği geçici saklayıcılardır:

- Data Registers: Aritmetik, karşılaştırma, mantıksal işlemler; veri transferi
  - Segment registers: belleklerin başlangıç adreslerini tanımlar
  - Pointer and Index registers: verinin başlangıç adresinden itibaren nerede olduğunu belirtir.
  - Program counter: yazılan program kodunun işleme aşamsını tanımlar
  - Flags: işlemler esnasında değişen durumları tanımlar.
- Kontrol birimi ve aritmetik/mantık birimi. Bu iki bileşen işlemci içinde sistem veriyolu adı verilen bir çeşit elektronik bağ ile birleştirilir. Veriyolu aynı zamanda bu bileşenleri bilgisayar sisteminin bellekleri ve I/O birimleri ile birleştirir.

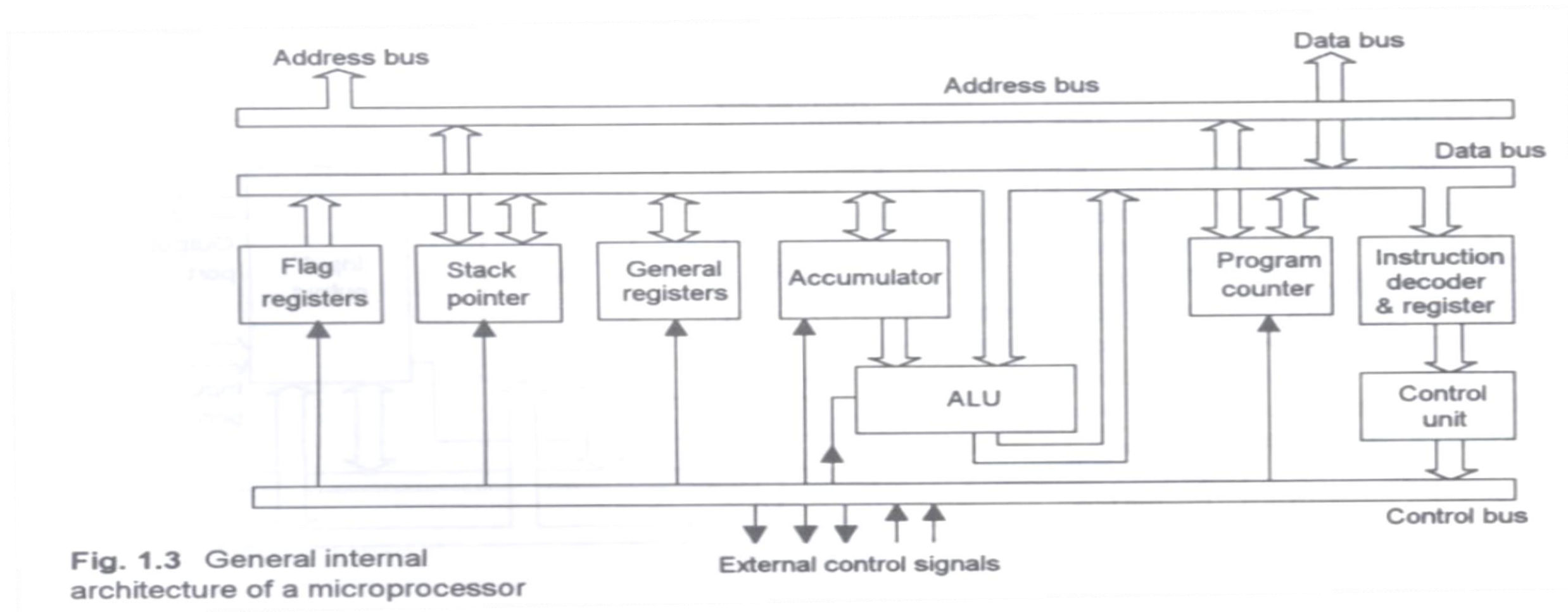


# Internal Structure of the Processor

- Control Unit
  - Fetches instructions from memory, Interprets them, Controls ALU
- ALU
  - Does all computations
- Register File
  - Stores variables



# Internal Components of Microprocessors



# Mikroişlemci Nedir?

- Mikroişlemciler, bilgisayar sisteminin kalbidir. Bilgisayar operasyonlarını kontrol ederek veri işleme işlevlerini yerine getirir. Kısaca işlemci veya CPU (Central Process Unit-Merkezi İşlem Birimi), kullanıcı ya da programcı tarafından yazılan programları meydana getiren komutları veya bilgileri yorumlamak ve yerine getirmek için gerekli olan tüm mantıksal devreleri kapsar.
- İlk mikroişlemci 1971 yılında hesap makinası amacıyla üretilen Intel firmasının 4004 adlı ürünüdür. Bir defada işleyebileceği verinin 4-bit olmasından dolayı 4-bitlik işlemci denilmekteydi.



# Mikroişlemcileri Birbirinden Ayıran Özellikler

- 1) Mikro işlemcinin her saat darbesinde işlem yapabileceği bit sayısı:** İşlenen veriler işlemcinin özelliğine göre 4-bit, 8-bit, 16-bit, 32-bit ve 64-bit uzunluğunda olabilir. Veri yolu uzunluğuna eşittir.
- 2) Komut İşleme Hızı:** Mikro işlemcilerin çalışması için saat sinyallerine ihtiyaç vardır. İşlemci (CPU) her saat sinyalinde bir sonraki işlem basamağına geçer. Saat frekansı mikro işlemciye dışardan uygulanan ya da işlemcinin içinde bulunan osilatörün frekansıdır. Komut çevrim süresi ise herhangi bir komutun görevini tamamlayabilmesi için geçen süredir.
- 3) Adresleme Kapasitesi:** Bir işlemcinin adresleme kapasitesi, adresleyebileceği veya doğrudan erişebileceği bellek alanının büyüklüğüdür. Bu büyüklük işlemcinin adres hattı sayısına bağlıdır. Bu hattın sayısı tasarlanacak sistemde kullanılacak bellek miktarını da belirlemektedir.
- 4) Özel amaçlı saklayıcıların Sayısı:** Mikro işlemcilerde kaydediciler, genel amaçlı kaydediciler ve özel amaçlı kaydediciler olmak üzere iki grupta toplanır. Bu kaydediciler 8, 16, 32 ve 64-bitlik olabilir. Kaydedicilerin sayısının programcının işinin kolaylaştırmasının yanında programın daha sade ve anlaşılır olmasını da sağlar. Her mikro işlemcinin kendine has yapısı ve kaydedici isimleri vardır. Herhangi bir mikro işlemciyi programlamaya başlamadan önce mutlaka bu kaydedicilerin isimlerinin ve ne tür işlevlere sahip olduklarının iyi bilinmesi gerekir.

# Mikroişlemcileri Birbirinden Ayıran Özellikler

**5) İlave Edilecek Devrelere Uyumluluk:** Mikro işlemcili sisteme eklenecek devrelerin en azından işlemci hızında çalışması gerekir. Sisteme ilave edilecek bellek entegrelerinin hızları işlemci ile aynı hızda olması tercih edilmelidir. Aynı şekilde sisteme takılan giriş çıkış birimlerinin hızları ve performansları mikro işlemci ile aynı veya çok yakın olmalıdır. Sisteme takılan birimlerin hızları mikro işlemciye göre düşükse mikro işlemcinin hızı diğer elemanlardaki yavaşlıktan dolayı düşer. Örneğin bir sensör ile ortam sıcaklığını ölçtüğümüzü düşünelim. Eğer sensörümüz geç ısınıp, soğuyorsa mikroişlemci ile sensörümüzü hızlı okumanın bir anlamı olmaz.

**6) Farklı Adresleme Modları:** Bir komutun işlenmesi için gerekli verilerin bir bellek bölgesinden alınması veya bir bellek bölgesine konulması ya da bellek–kaydedici veya kaydedici–kaydedici arasında değiştirilmesi için farklı erişim yöntemleri kullanılır. Mikro işlemcinin işleyeceği bilgiye farklı erişim şekilleri, "adresleme yöntemleri" olarak ifade edilir. Kısaca adresi tarif yollarıdır.

Adresleme türleri: Doğrudan, Dolaylı, Veri tanımlı, Özel amaçlı saklayıcı, Mutlak, Göreceli, İndisli ve göstergeli adresleme

# Specialized Processors

- GPU
- FPU
- DSP - Digital Signal Processors
  - Image processing; sound, speech
- Math co-processors
  - Real number arithmetic
- ASICs - Application-Specific Integrated Circuits
  - Microwave controller
  - Engine management controller

# CPU'yu Oluşturan Ana Bileşenler -1

İşlemcilerin içerisindeki donanım blokları, komutları uygun sırayla bellekten okur ve gerekli verilere erişip, bu veriler üzerinde komutta tanımlanan işleri yürütür.

- **Özel Amaçlı Saklayıcılar (Register):** Mikroişlemcide işlenecek ya da transfer edilecek verinin geçici olarak saklandığı (RAM), Kontrol ve ALU birimlerinin doğrudan bağlandığı bellek birimidir. Birinci önceliklidir. Özel Amaçlı Saklayıcılar verinin manevrasında ve geçici olarak tutulmasında görevlidirler. Veri işleme ve veri iletişim ara yüzünde kullanılır. Yüksek hızlı veri işlemede ve transfer etmede hızlı rol oynar. Geçici depolama alanıdır.
- **Önbellekler (Cach):** İşlenecek verinin önceden getirilip CPU'da hazır edilmesinde kullanılır. Performans artırıcı özelliği vardır. Çünkü yazma ve okuma sürelerini hızlandırır. İkinci önceliklidir.
- **Aritmetik ve Mantık Birimi (Arithmetic Logic Unit -ALU):** Tüm matematik ve mantık işlemlerini gerçekleştirir, CU'dan komut dizisini alır. Mikro işlemcinin birinci derecede önem taşıyan bir birimidir

# Mikroişlemcilerin İşlevsel Bölümleri -2

- **Kontrol Ünitesi ( Control Unit -CU):** Donanımların çalışmasını düzenler. İşlem akışını düzenler, komutları yorumlar ve bu komutların yerine getirilmesini sağlar. CU, işlenen komuta göre mikro işlemci içerisindeki operasyon zamanlamasını / sıralamayı kontrol eder.
- **Haberleşme Yolları:** İşlemci iç mimarisindeki blokları birbirine bağlayan veri yolu yapılarıdır. Mikro işlemci ile bilgisayarın diğer birimleri arasındaki bağlantıları sağlayan iletkenlerdir. Veri yolları (data bus), Adres yolları (address bus), Denetim yolları (control bus)
- **Sayıcılar (Counter):** Sayıcılar işlemi yapılacak komut ve verilerin adreslerini taşıyarak bilgisayarın çalışması sırasında hangi verinin hangi sırayla kullanılacağını belirlerler.
- **Giriş-Çıkış Devreleri:** Bu devreler mikroişlemcinin, yalnız giriş ve yalnızca çıkış yapan veya giriş-çıkış yapan birimleri ile bağlantı kurduğu devrelerdir.

# Mikroişlemcilerin İşlevsel Bölümleri -3

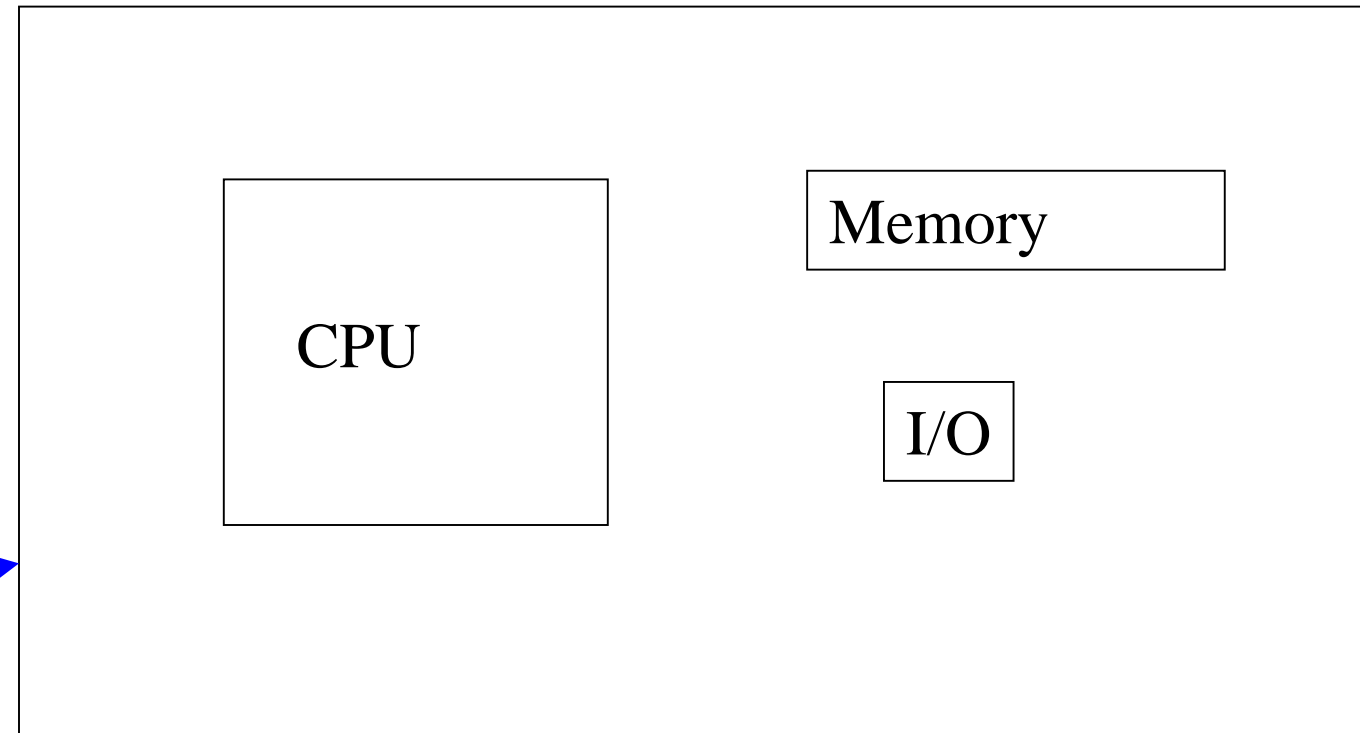
- **Clock Darbe Katarı:** Mikro işlemcide temel parametre clock hızıdır. Clock darbe katarı, CPU işleminde bitleri senkronize (eş zamanlı) eder. Her bilgisayarın bir sistem clock sinyali vardır. Clock sinyali 1 ve 0 lardan oluşan darbe katarıdır. Bit olarak tanımlanan verileri tetiklemede yani işlemede kullanılır. Kişisel bilgisayarın hızları genellikle gigahertz (GHz) cinsinden ifade edilir. Clock'un periyodu veri uzunluğunda olmalıdır. Clock süreklidir, sonsuzdur. 1 ve 0 ın süresi (periyodu), 1 bit uzunluğundadır. Darbe katarı, ezelden ebede gider.
- **Kayan Nokta Birimi (FPU):** Matematik işlemcisi olarak da bilinir. Mikroişlemcide yoğun matematik işlemleri yapan birimdir. Mikro işlemcinin işlem gücünü belirlemektedir.
- **Grafik İşlem Birimleri (GPU).**



# **Embedded System**

# Basic Components of Digital Computer

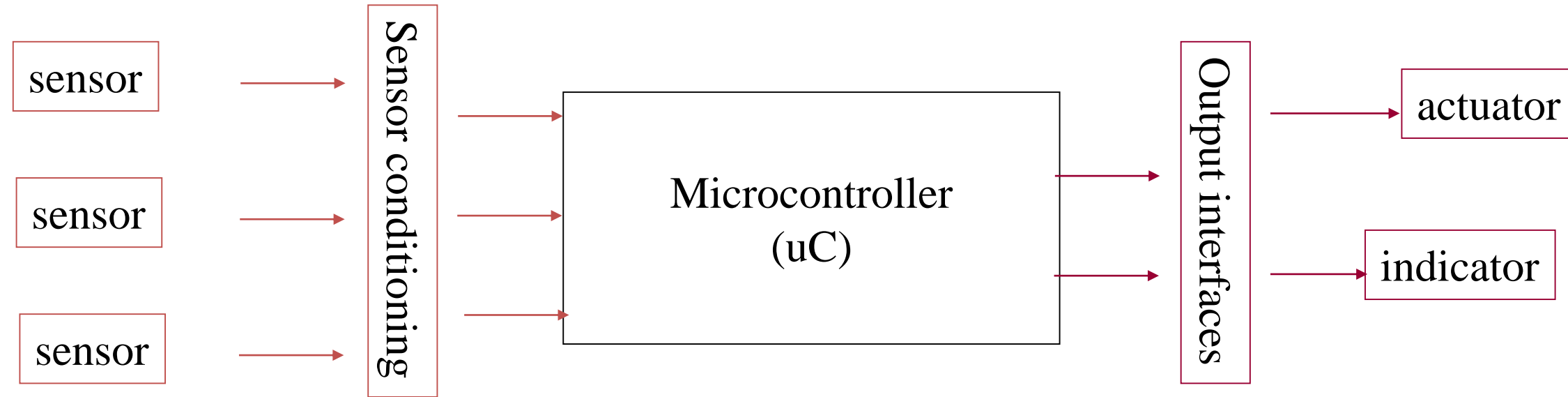
- CPU
- Memory
- I/O



Could be a chip, a board, or several boards



# Embedded System General Block Diagram



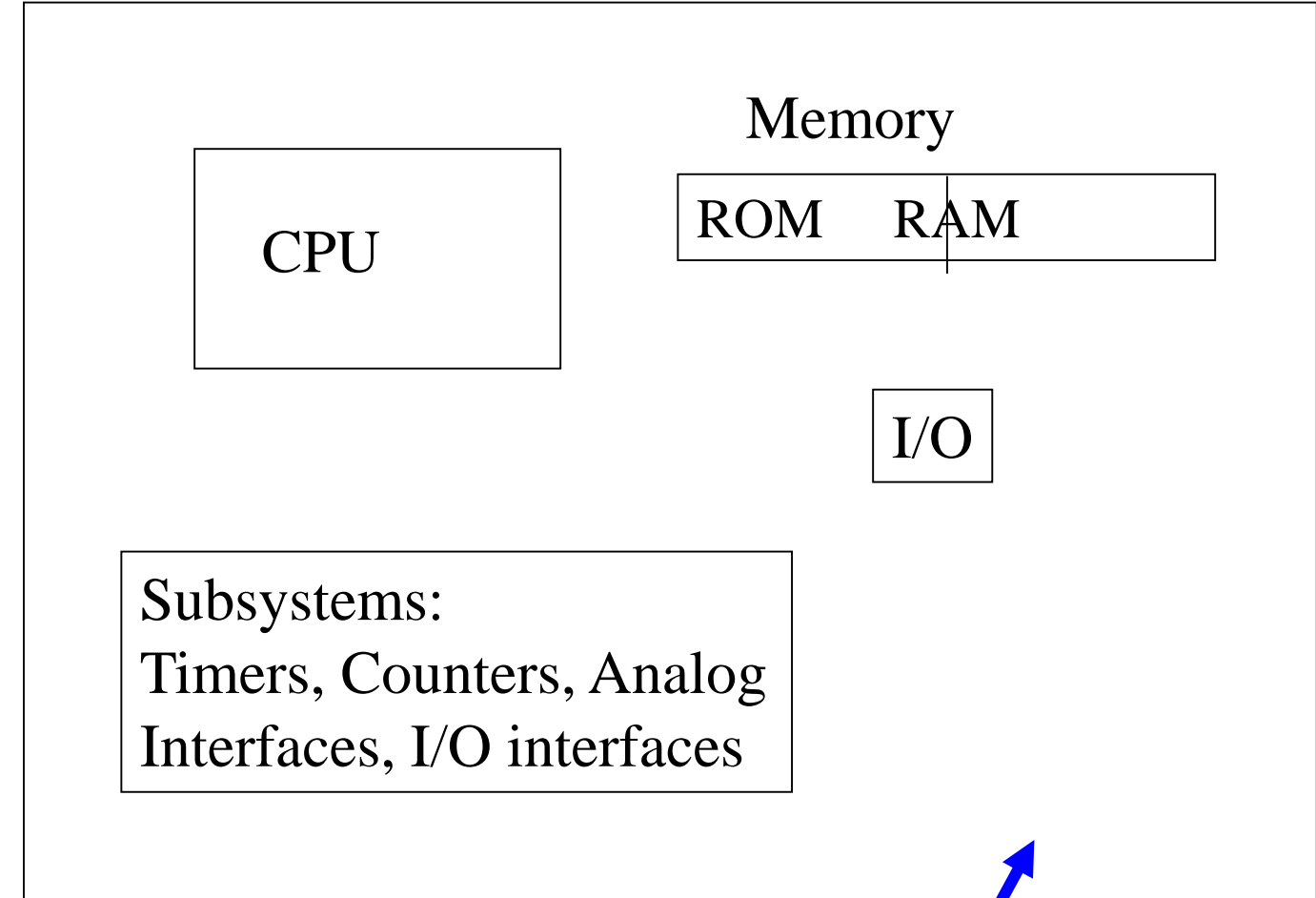
**Aktüatör ya da eyleyici:** Bir mekanizmayı veya sistemi kontrol eden veya hareket ettiren bir tür motordur. Bir enerji kaynağı tarafından çalıştırılır. Bu kaynak genellikle elektrik akımı, hidrolik akışkan basıncı veya pnömatik basınçtır ve bazı tür hareketlerle enerjiye dönüşür.

# Mikrodenetleyiciler

Bir mikroişlemci çekirdeğine ilave olarak, ortak bellek alanlarını kullanan, özelleştirilmiş görevler ile donatılmış çevrebirimlerin eklenmesi ile ortaya çıkan yapıya mikrodenetleyici denir. Denetim teknolojisi gerektiren uygulamalarda kullanılmak üzere tasarlanmış olan mikrodenetleyiciler, mikro işlemcilere göre çok **daha basit ve ucuzdur**.

Mikroişlemcili sistemin tasarımı ve kullanımı mikrodenetleyicili sisteme göre daha karmaşık ve masraflıdır.

Mikrodenetleyicili bir sistemin çalışması için elemanın kendisi ve bir osilatör kaynağının olması yeterlidir. Mikrodenetleyicilerin küçük ve ucuz olmaları, bunların tüm elektronik kontrol devrelerinde kullanılmasını sağlamaktadır.



A single chip

# Mikrodenetleyici Seçimi

Mikrodenetleyiciler ile tasarım yapmadan önce tasarlanan sisteme uygun bir denetleyici seçmek için o denetleyicinin taşıdığı özelliklerin bilinmesi gereklidir. Mikrodenetleyicinin hangi özelliklere sahip olduğu kataloglarından anlaşılabilir. Aşağıda sıralanan özellikler bunlardan bazılarıdır;

- Programlanabilir dijital paralel giriş/çıkış.
- Programlanabilir analog giriş/çıkış.
- Seri giriş/çıkış (senkron, asenkron ve cihaz yönetimi).
- Motor veya servo kontrol için pals sinyali çıkışı.
- Harici giriş vasıtasıyla kesme.
- Harici bellek arabirimi.
- Harici veri yolu arabirimi.
- Dahili bellek tipi seçenekleri (ROM, EPROM, PROM, EEPROM).
- Dâhilî RAM seçeneği.
- Kayan nokta hesaplaması.

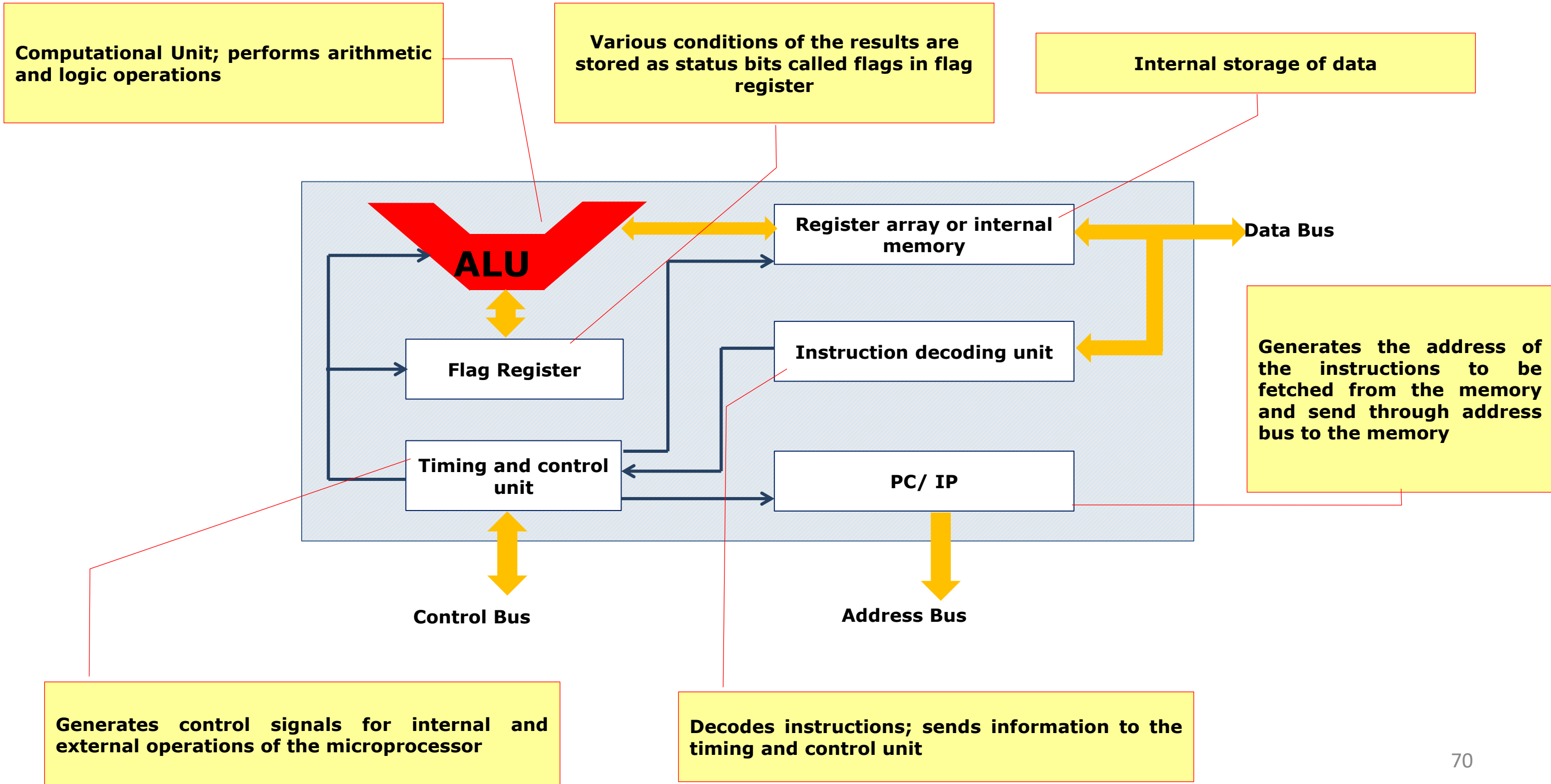
# Mikroişlemci ve Mikrodenetleyiciler Arasındaki Farklar

- Bir mikroişlemci görevini yerine getirebilmesi için mutlaka, verilerin saklanacağı bellek birimine, dış dünyadan veri alışverişinin düzenli yapılmasını sağlayan giriş/çıkış birimine ihtiyaç duyar. Bunlar bir mikroişlemcili sistemde ayrı ayrı birimler (entegreler) şeklinde yerini alır. Bundan dolayı mikroişlemcili sistemlere çok entegreli sistemler denilir.
- Bilgisayar gibi mikroişlemcili sistemlere verilen bir örnekte, bir bilgisayarın bir çamaşır makinesinde veya cep telefonunda kullanılması elbette mümkün olmayacaktır.
- Bilgisayar aynı anda milyonlarca işi yapabildiğinden ve çok yer kapladığından böyle yerlerde kullanılması mantıklı olmaz ve maliyetli olur. Bundan dolayı, sistemi meydana getiren elemanların birçok özelliklerinden feragat edilerek ve bir entegrede birleştirilerek mikroişlemcilerin yeni türevleri (mikrodenetleyiciler) oluşturulmuştur. Bir saydırma veya PWM sinyali üreteceğimizi düşünürsek mikroişlemci ile bunu yazılımsal olarak yapmamız gerekecektir. Ancak mikrodenetleyicinin özelleştirilmiş modülleri sayesinde bu işlemleri programa paralel olarak modüllerle yapabiliriz. Böylece ana programdaki yoğunluk azalır ve işlemcimiz hızlanmış olur.

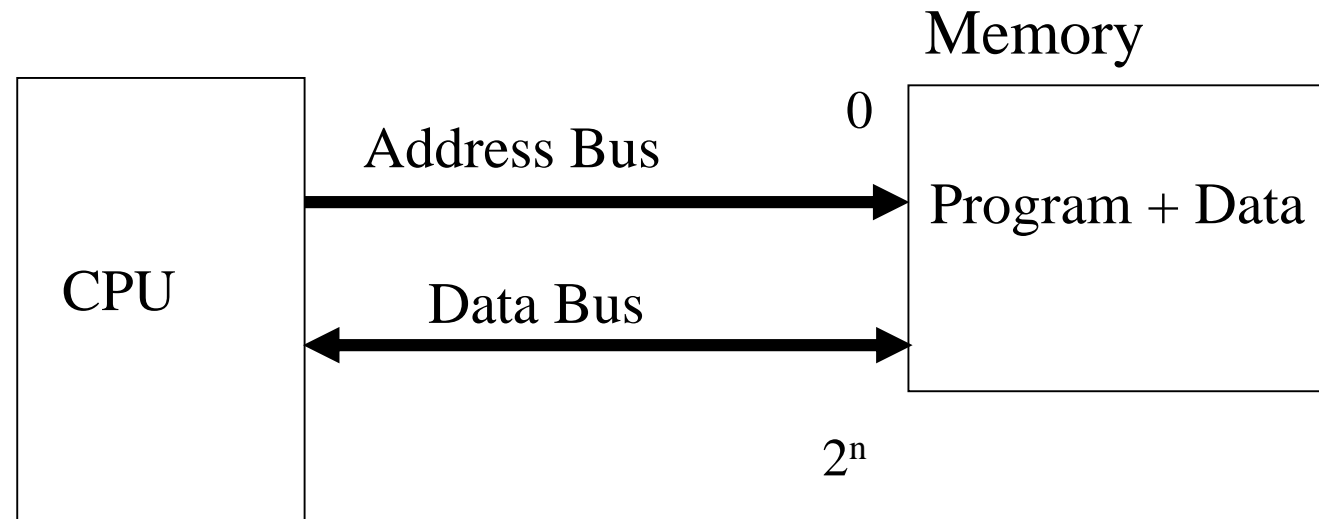


# **“Mikroişlemci Mimari Yapısı”**

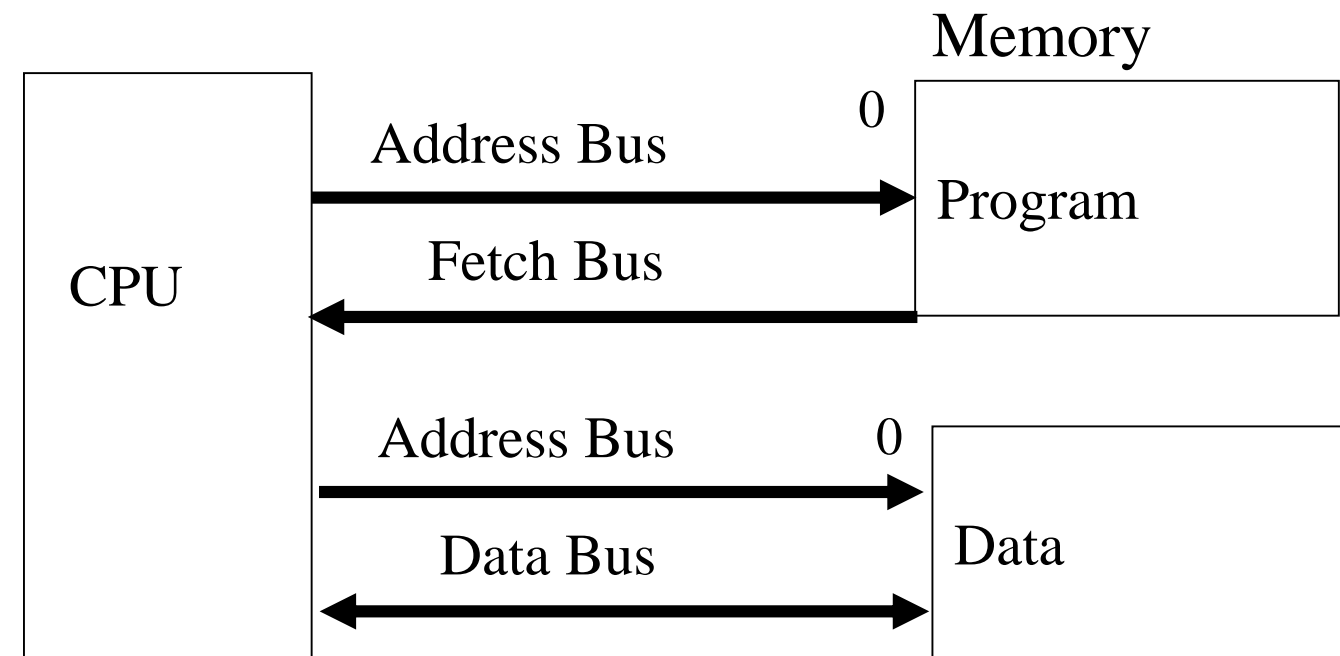
# Functional blocks



# CPU Architectures



Von Neumann  
Architecture



Harvard  
Architecture

# CISC – Complex Instruction Set Computers

- Refers to number and complexity of instructions
- Improvements was: Multiply and Divide
- The number of instruction increased from
  - 45 on 4004 to:
  - 246 on 8085
  - 20,000 on 8086 and 8088

# RISC – Reduced Instruction Set Computer

- Executes one instruction per clock

# Newer RISC - Superscaler Technology

- Execute more than one instruction per clock

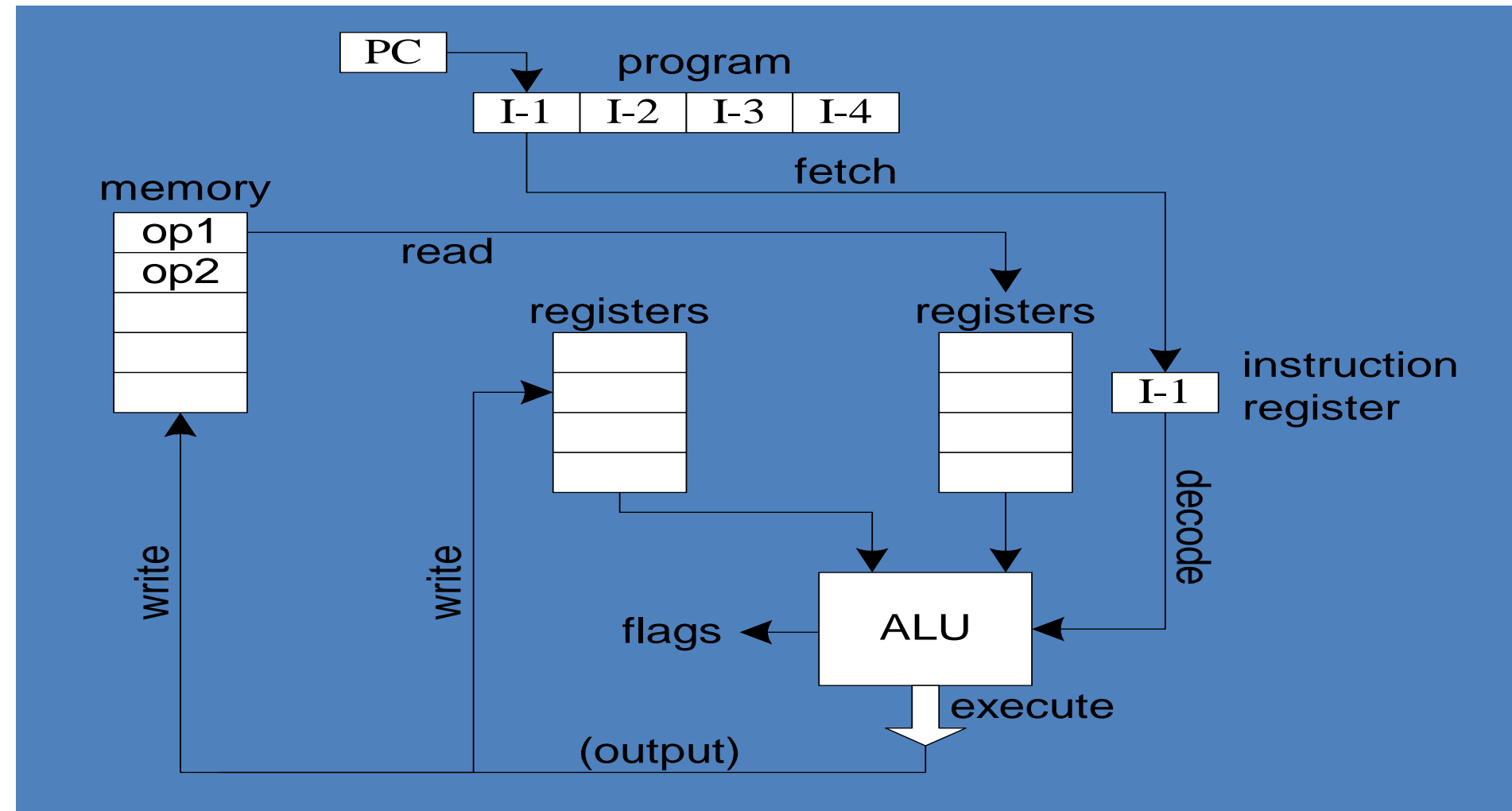


# CISC and RISC

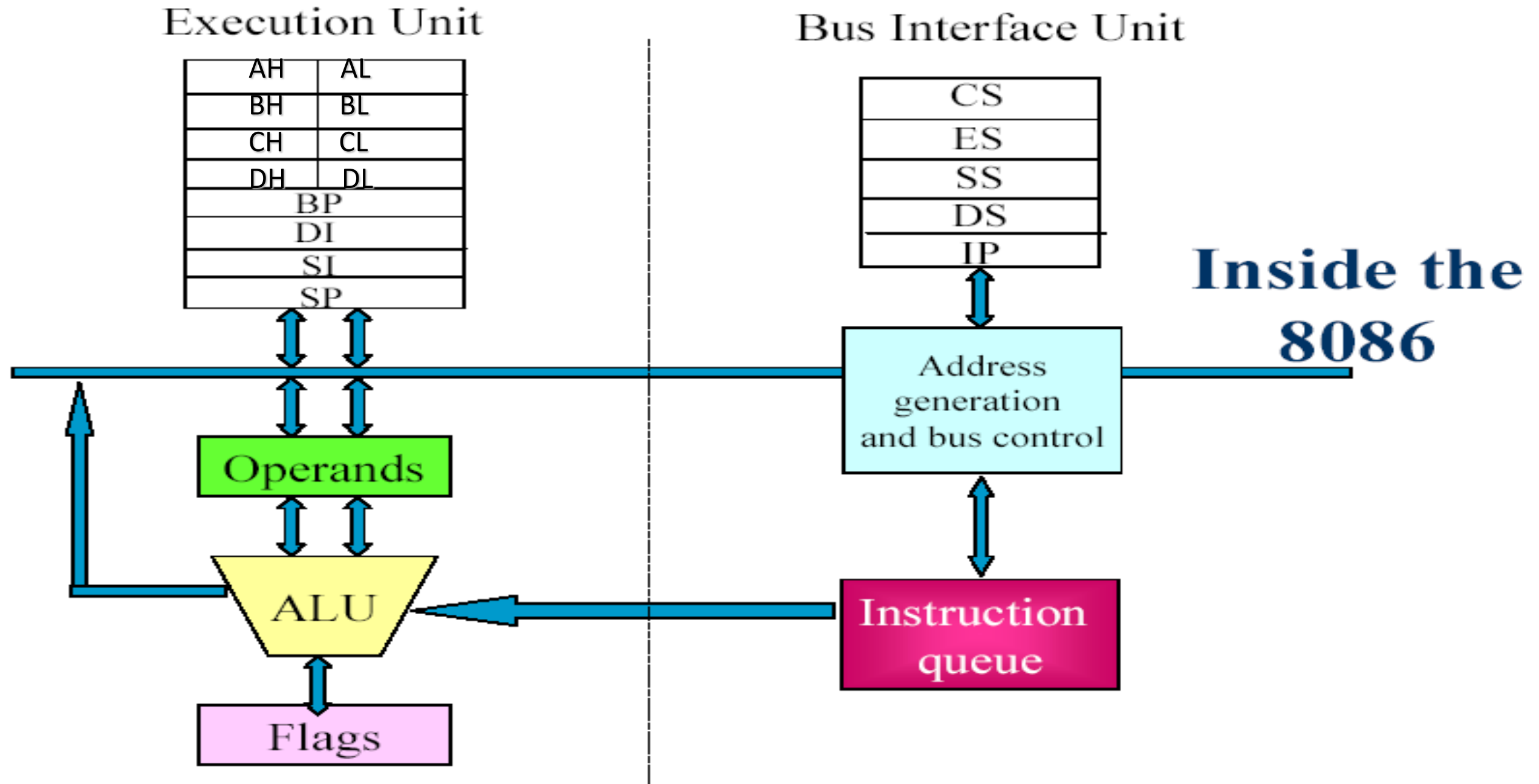
- CISC – complex instruction set
  - large instruction set
  - high-level operations
  - requires microcode interpreter
  - examples: Intel 80x86 family
- RISC – reduced instruction set
  - simple, atomic instructions
  - small instruction set
  - directly executed by hardware
  - examples:
    - ARM (Advanced RISC Machines)
    - DEC Alpha (now Compaq)

# Instruction Execution Cycle

- Fetch: veri al getir; veri al götür (Ram Bellek, I/O ), komut al getir (Kod bellek)
- Komut çözme (decode)
- Decoding: Bellek ya da I/O birimi seçme
- Bellek gözü seçme
- Execute: veri işleme
- Veri transfer
- Veri saklama



# Inside X86



## Inside x86...*pipelining*

Intel implemented the concept of pipelining by splitting the internal structure of the x86 into two sections that works simultaneously:

- **Execution Unit (EU)** – executes instructions previously fetched
- **Bus Interface Unit (BIU)** – accesses memory and peripherals

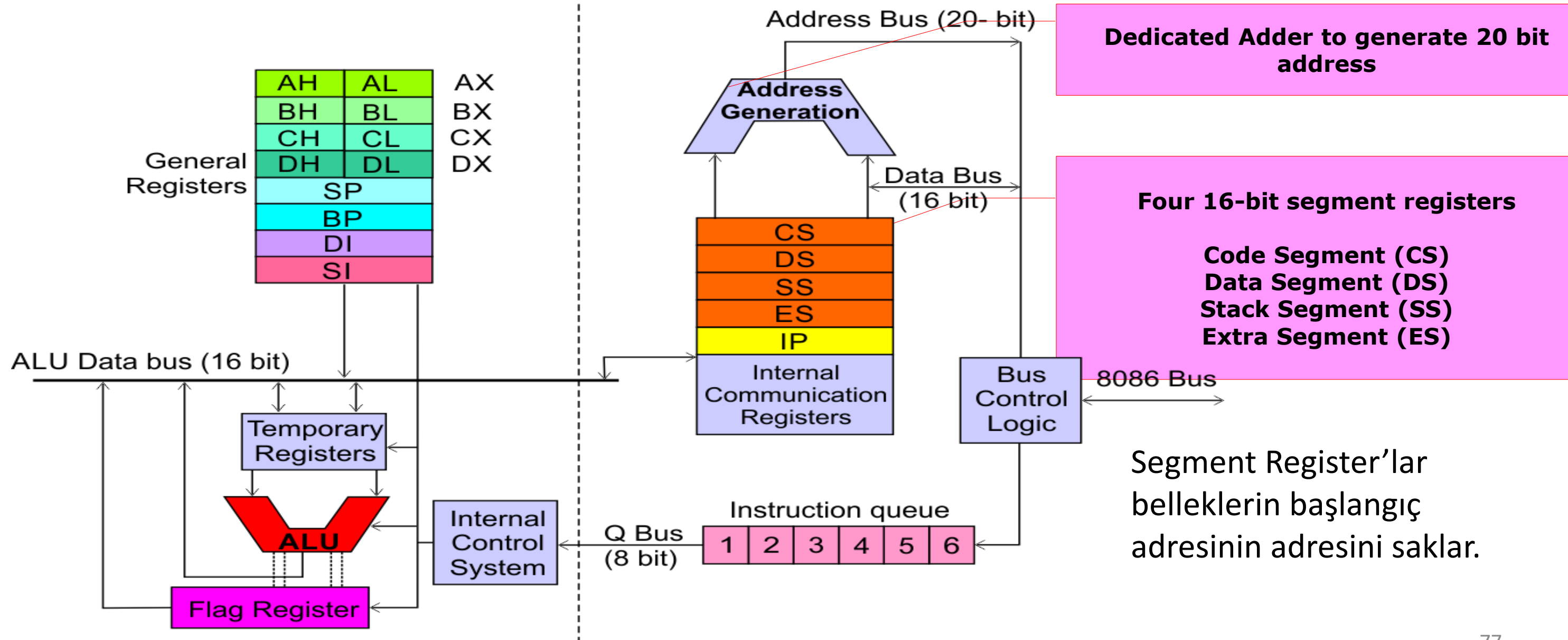
## Execution Unit (EU)

EU executes instructions that have already been fetched by the BIU.

BIU and EU functions separately.

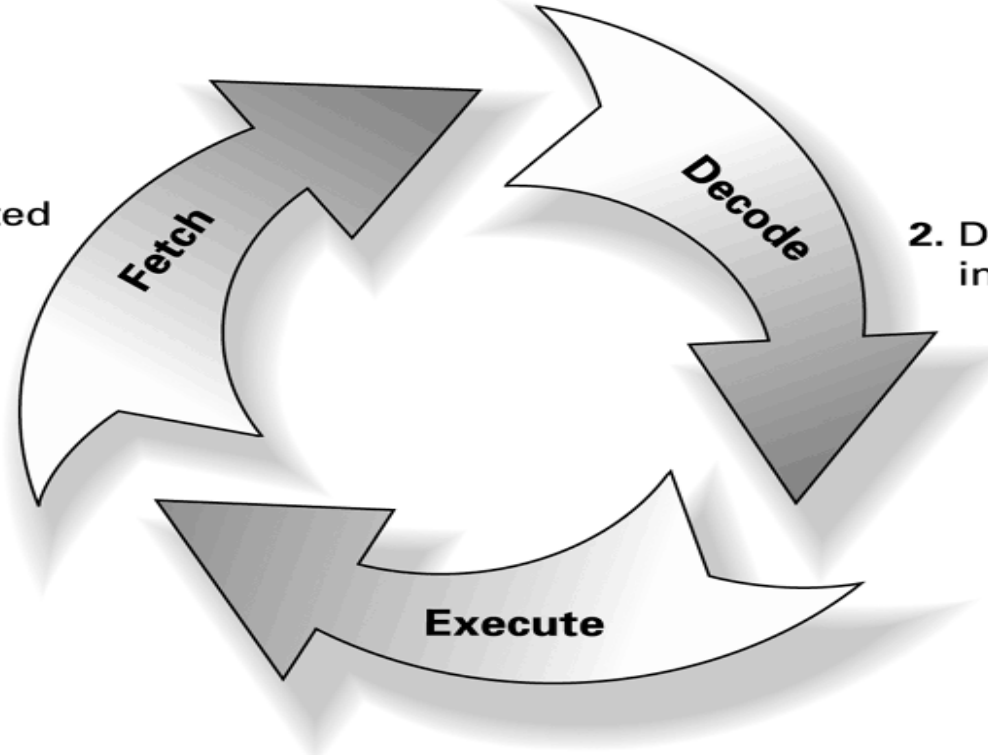
## Bus Interface Unit (BIU)

BIU fetches instructions, reads data from memory and I/O ports, writes data to memory and I/O ports.



# Mikroişlemci işlev döngüsü

1. Retrieve the next instruction from memory (as indicated by the program counter) and then increment the program counter.



2. Decode the bit pattern in the instruction register.

3. Perform the action required by the instruction in the instruction register.

- İşlev döngüsünü yerine getiren birimler ayrı ayrı çalışır.

## Mikroişlemci ana işlev döngüsünde,

### Fetch Process:

- Veriyi bellek ya da I/O biriminden Git – Al – Getir.
- Veriyi bellek ya da I/O birimine Al - Götür.

**Execute process** : Uygula, işle

**Decode**: Bellek ve ilgili gözünü seçer.

# Fetch-Execute Process

---

- **Program Counter (PC) or Instruction Pointer (IP)**
  - Holds address of next instruction to fetch
- **Instruction Register (IR)**
  - Stores the instruction fetched from memory
- **Fetch-Execute process**
  - Read an instruction from memory addressed by PC
  - Increment program counter
  - Execute fetched instruction in IR
  - Repeat process

# Inside x86

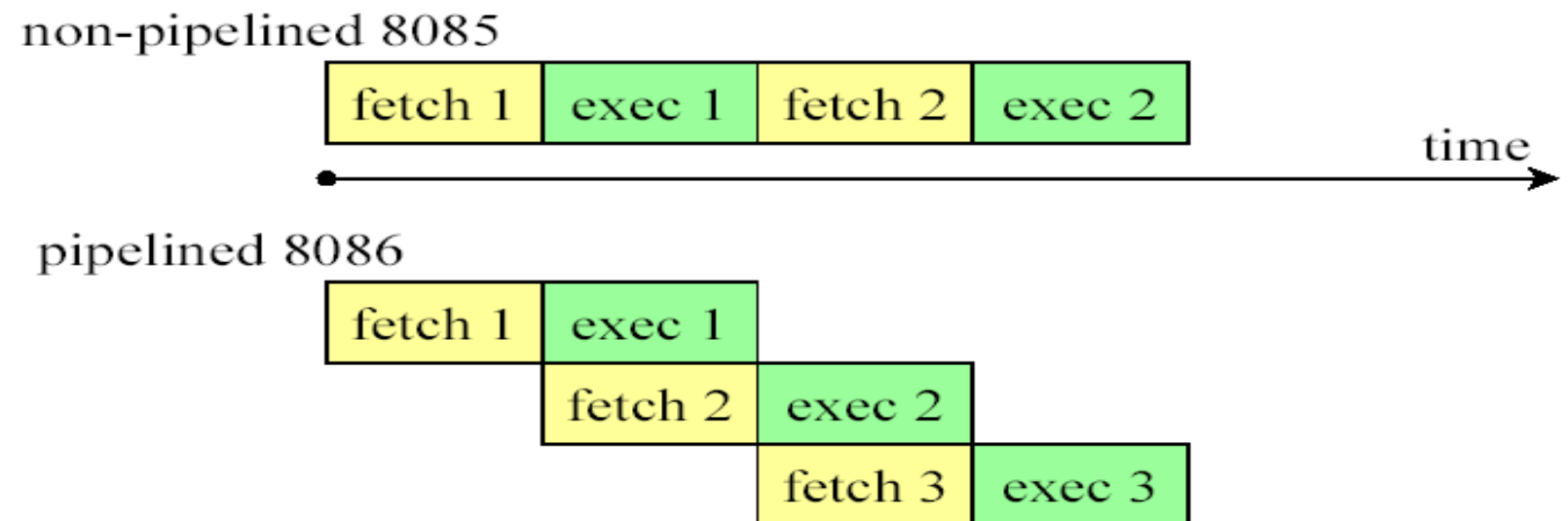
## Concepts important to the internal operation of x86

- Pipelining: İşlevsel döngüleri yerine getiren tüm birimlerin aynı anda işlem yapması, çalışması için çok sayıda komut parçalarının aynı anda manipüle edilmesidir.



# Inside X86...*pipelining*

- **Pipelining**
  - Two ways to make CPU process information faster:
    - Increase the working frequency – technology dependent
    - Change the internal architecture of the CPU
  - Pipelining is to allow CPU to fetch and execute at the same time



# Without Pipelining

- Pipeline, ardışık düzende, işlemcinin komutları paralel olarak yürütmesini mümkün kılar
- S1 .. S6: Cpu Birimleri
- Cycles: Clock periyotlarından oluşur. Herbir işlevin yerine getirildiği süreyi tanımlar.
- Ayrık aşamalara bölünmüş komut yürütme
- Sağda, ardışık düzenlenmemiş bir işlemci örneği. Birçok boşa giden döngü.

		Stages					
		S1	S2	S3	S4	S5	S6
Cycles	1	I-1					
	2		I-1				
	3			I-1			
	4				I-1		
	5					I-1	
	6						I-1
	7	I-2					
	8		I-2				
	9			I-2			
	10				I-2		
	11					I-2	
	12						I-2

# Pipelined Execution

- Döngülerin (Cycles) daha verimli kullanımı, daha fazla talimat çıktısı:

		Stages					
		S1	S2	S3	S4	S5	S6
Cycles	1	I-1					
	2	I-2	I-1				
	3		I-2	I-1			
	4			I-2	I-1		
	5				I-2	I-1	
	6					I-2	I-1
	7						I-2

k durum ve n komut için  
gerekli döngü sayısı:

$$k + (n - 1)$$

- Pipelining uygulamasında tüm birimler aynı anda farklı işlevlerde (Komutlar) çalışmaya başlar.

# Wasted Cycles (pipelined)

- Komut işleme döngüsü (Decoding, Fetching, Executing, ...) aşamalardan biri, iki veya daha fazla saat döngüsü gerektirdiğinde (Clock periyodu), saat döngülerinin bir kısmı boşa harcanır.

		Stages					
		S1	S2	S3	S4	S5	S6
Cycles	1	I-1					
	2	I-2	I-1				
	3	I-3	I-2	I-1			
	4		I-3	I-2	I-1		
	5			I-3	I-1		
	6				I-2	I-1	
	7				I-2		I-1
	8				I-3	I-2	
	9				I-3		I-2
	10					I-3	
	11						I-3

For  $k$  states and  $n$  instructions, the number of required cycles is:

$$k + (2n - 1)$$

# Superscalar

Bir süperskalar işlemcinin birden çok yürütme işlem hattı vardır. Aşağıda, Aşama S4'ün sol ve sağ boru hatlarına (u ve v) sahip olduğuna dikkat edin.

		Stages						
		S1	S2	S3	S4		S5	S6
					u	v		
Cycles	1	I-1						
	2	I-2	I-1					
	3	I-3	I-2	I-1				
	4	I-4	I-3	I-2	I-1			
	5		I-4	I-3	I-1	I-2		
	6			I-4	I-3	I-2	I-1	
	7				I-3	I-4	I-2	I-1
	8					I-4	I-3	I-2
	9						I-4	I-3
	10							I-4

For  $k$  states and  $n$  instructions, the number of required cycles is:

$$k + n$$



# Registers

# Registers

- Special-purpose
- High-speed
- Temporary storage
- Located inside CPU

Mikroişlemcinin iç yapısında,

- Veri işleme ve Veri iletişim ara yüzünde kullanılır.
- Özel amaçlı saklayıcıdır.
- Yüksek hızlı veri işleme ve transfer etme hızlı rol oynar.
- Geçici depolama alanıdır.
- CPU içinde bulunur. CPU'nun ana bileşenidir.

# Registers

- Registers are used to store information temporarily
- 8086, 8088, 80286 contains 8-bit, 16-bit registers
- 80386, 80486, Pentium, Pentium Pro, and Pentium II contain 8, 16, and 32-bit registers.
- How about Pentium 4?



# Overview

- Registers
  - General purpose registers
    - Operands for logical and arithmetic operations
    - Operands for address calculations
    - Memory pointers
  - Segment registers
  - EFLAGS register
  - The instruction pointer register
- The stack

# CPU Registers

---

- **Fourteen 16-bit registers**
- **Data Registers**
  - AX (Accumulator Register): AH and AL
  - BX (Base Register): BH and BL
  - CX (Count Register): CH and CL
  - DX (Data Register): DH and DL
- **Pointer and Index Registers**
  - SI (Source Index)
  - DI (Destination Index)
  - SP (Stack Pointer)
  - BP (Base Pointer)
  - IP (Instruction Pointer)

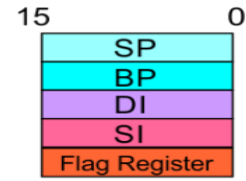
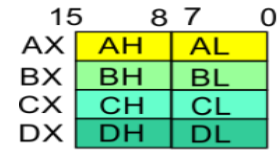
# CPU Registers

---

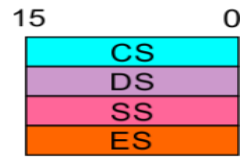
- **Segment Registers**
  - CS (Code Segment)
  - DS (Data Segment)
  - SS (Stack Segment)
  - ES (Extra Segment)
- **FLAGS Register**
  - Zero flag
  - Sign flag
  - Parity flag
  - Carry flag
  - Overflow flag

## Architecture

8086 registers categorized into 4 groups



EU



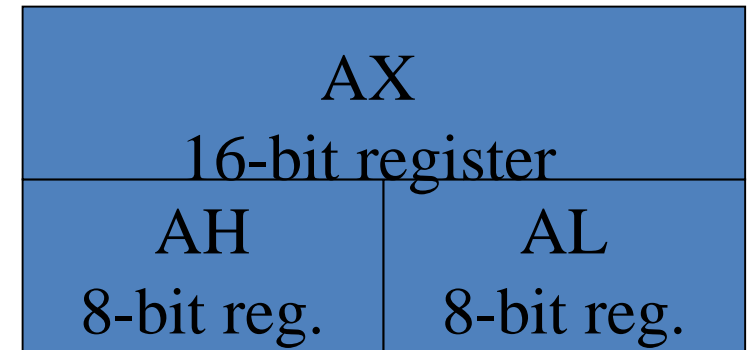
BIU



Sl.No.	Type	Register width	Name of register
1	General purpose register	16 bit	AX, BX, CX, DX
		8 bit	AL, AH, BL, BH, CL, CH, DL, DH
2	Pointer register	16 bit	SP, BP
3	Index register	16 bit	SI, DI
4	Instruction Pointer	16 bit	IP
5	Segment register	16 bit	CS, DS, SS, ES
6	Flag (PSW)	16 bit	Flag register

# Inside X86...registers

- **Registers**
  - To store information temporarily



<b>Category</b>	<b>Bits</b>	<b>Register Names</b>
General	16	AX, BX, CX, DX
	8	AH, AL, BH, BL, CH, CL, DH, DL
Pointer	16	SP (stack pointer), BP (base pointer) ; Stack segment
Index	16	SI (source index), DI (destination index): DS (BX), ES (BX)
Segment	16	CS (code segment), DS (data segment) SS (stack segment), ES (extra segment) points to the exception handling chain
Instruction	16	IP (instruction pointer, code segment)
Flag	16	FR (flag register)

Register	Name of the Register	Special Function
AX	16-bit Accumulator	Stores the 16-bit results of arithmetic and logic operations
AL	8-bit Accumulator	Stores the 8-bit results of arithmetic and logic operations
BX	Base register	Used to hold base value in base addressing mode to access memory data
CX	Count Register	Used to hold the count value in SHIFT, ROTATE and LOOP instructions
DX	Data Register	Used to hold data for multiplication and division operations
SP	Stack Pointer	Used to hold the offset address of top stack memory
BP	Base Pointer	Used to hold the base value in base addressing using SS register to access data from stack memory
SI	Source Index	Used to hold index value of source operand (data) for string instructions
DI	Data Index	Used to hold the index value of destination operand (data) for string operations

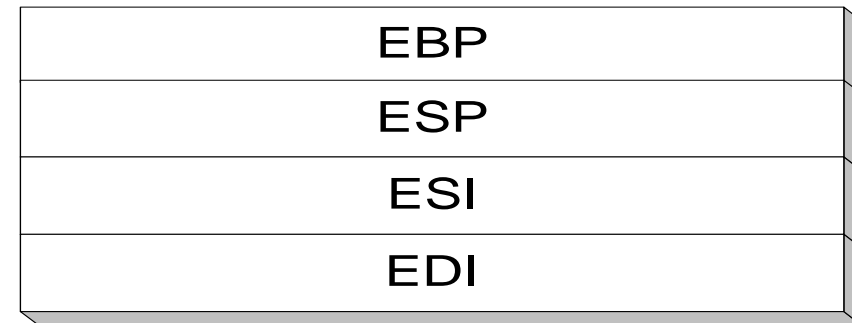
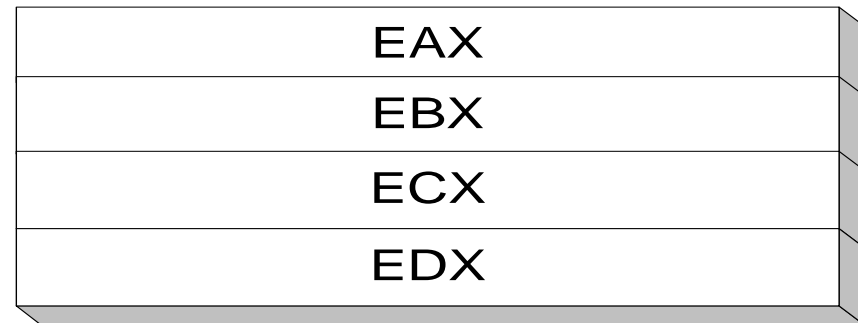


# **32 Bit Registers**

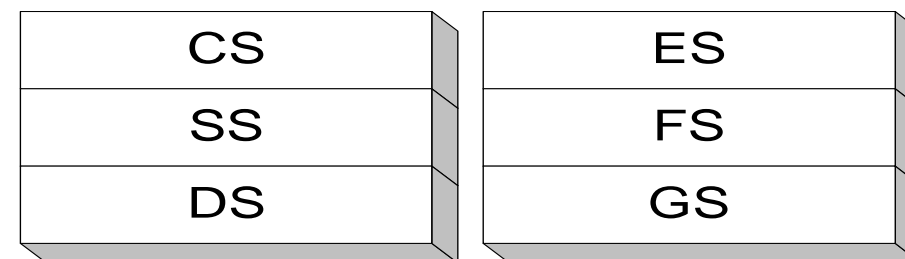
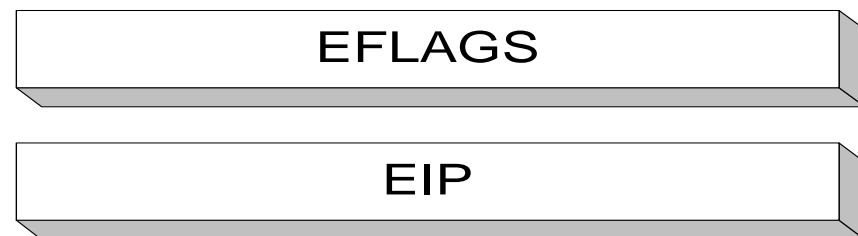
# 32 Bit General-Purpose Registers

Named storage locations inside the CPU, optimized for speed.

## 32-bit General-Purpose Registers



## 16-bit Segment Registers



# Registers

## 6 Category

- General
- **Pointer**
- Index
- Segment
- Instruction
- Flag

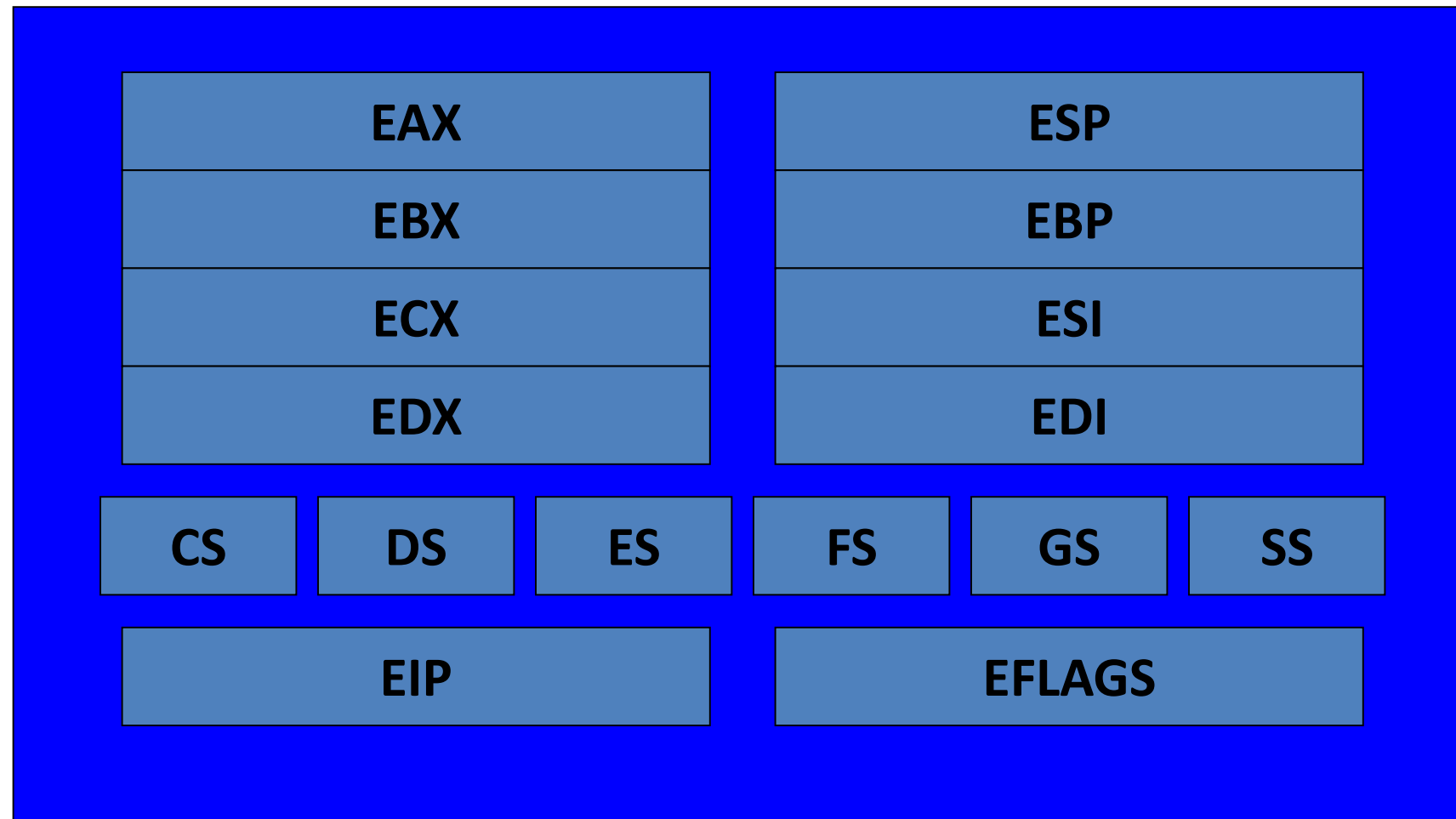
<b>EAX</b>		AH	<b>AX</b>	AL
<b>EBX</b>		BH	<b>BX</b>	BL
<b>ECX</b>		CH	<b>CX</b>	CL
<b>EDX</b>		DH	<b>DX</b>	DL
<b>EBP</b>		<b>BP</b>		
<b>ESI</b>		SI		
<b>EDI</b>		DI		
<b>ESP</b>		<b>SP</b>		
<b>EIP</b>		IP		
<b>EFLAGS</b>		FLAGS		
		CS		
		DS		
		ES		
		SS		
		FS		
		GS		



# Some Specialized Register Uses

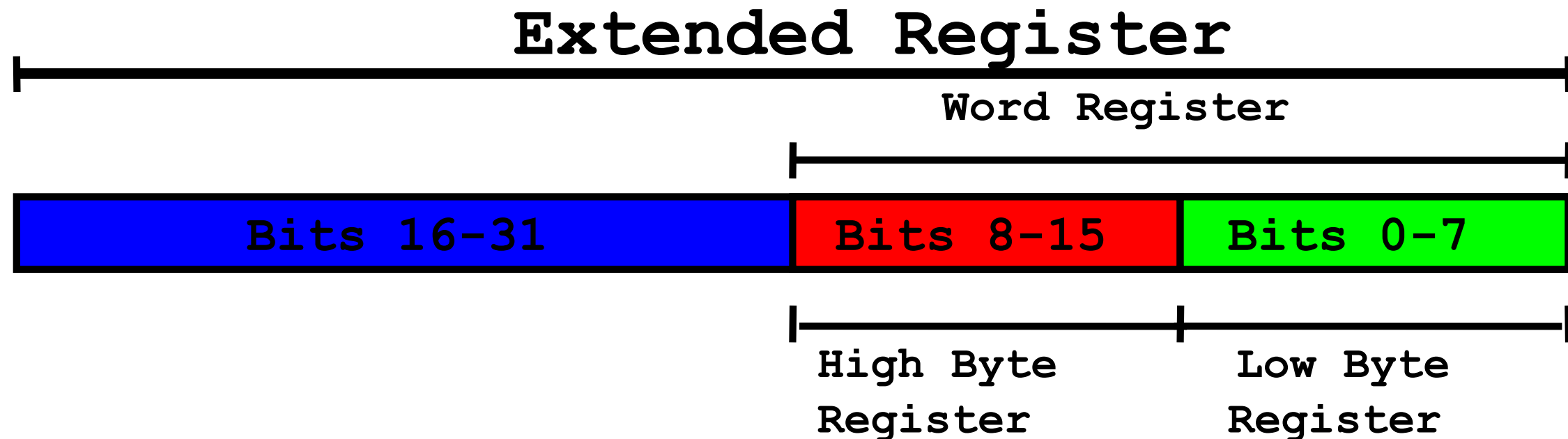
- General-Purpose
  - EAX – accumulator
  - ECX – loop counter
  - ESP – stack pointer
  - ESI, EDI – index registers
  - EBP – extended frame pointer (stack)
- Segment
  - CS – code segment
  - DS – data segment
  - SS – stack segment
  - ES, FS, GS - additional segments

# 32 bit registers (16 adet)



Intel Pentium processor

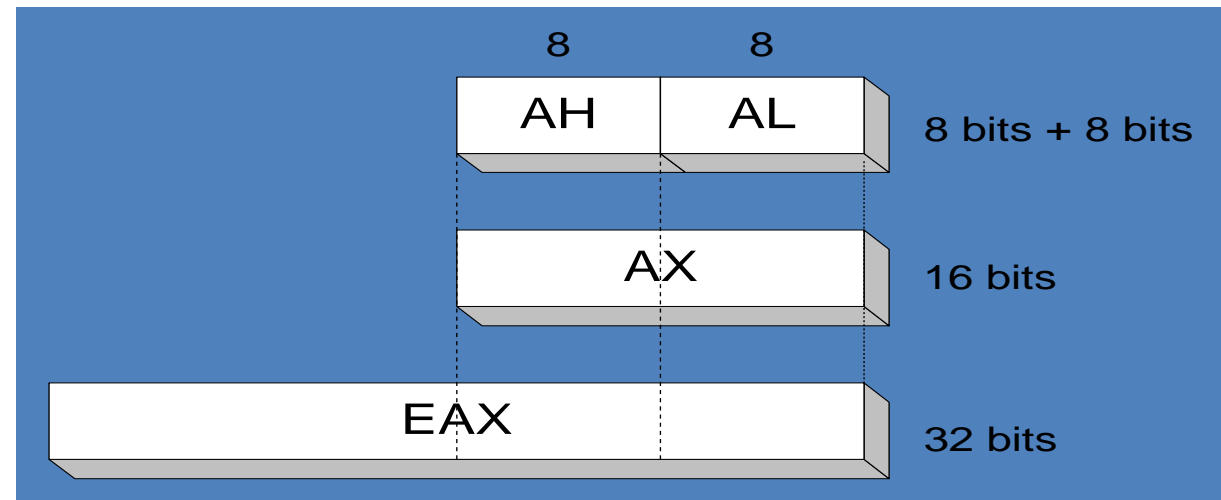
# Anatomy of a 32 bit Register



In today's 32-bit architecture, each register has 32 bits that can be used. However, some registers can be accessed as a single byte, and all registers can be accessed as a 16-bit value, or a 32-bit value. This is further explained on the next slide.

# Accessing Parts of 32bit Registers

- Use 8-bit name, 16-bit name, or 32-bit name
- Applies to EAX, EBX, ECX, and EDX



32-bit	16-bit	8-bit (high)	8-bit (low)
EAX	AX	AH	AL
EBX	BX	BH	BL
ECX	CX	CH	CL
EDX	DX	DH	DL

# General Registers I

- **EAX** – ‘Accumulator’
  - accumulator for operands and results data
  - usually used to store the return value of a procedure
- **EBX** – ‘Base Register’
  - pointer to data in the DS segment
- **ECX** – ‘Counter’
  - counter for string and loop operations
- **EDX** – ‘Data Register’
  - I/O pointer

# General Registers II

- **ESI** – ‘Source Index’
  - source pointer for string operations
  - typically a pointer to data in the segment pointed to by the DS register
- **EDI** – ‘Destination Index’
  - destination pointer for string operations
  - typically a pointer to data/destination in the segment pointed to by the ES register

# General Registers III

- **EBP** – ‘Base Pointer’
  - pointer to data on the stack
  - points to the current stack frame of a procedure
- **ESP** – ‘Stack Pointer’
  - pointer to the top address of the stack
  - holds the stack pointer and as a general rule should not be used for any other purpose

# Index and Base Registers

- Some registers have only a 16-bit name for their lower half:

32-bit	16-bit
ESI	SI
EDI	DI
EBP	BP
ESP	SP



# Segment Registers

- **CS** – ‘Code Segment’
  - contains the segment selector for the code segment where the instructions being executed are stored
- **DS (ES , FS , GS)** – ‘Data Segment’
  - contains the segment selectors for the data segment where data is stored
- **SS** – ‘Stack Segment’
  - contains the segment selector for the stack segment, where the procedure stack is stored

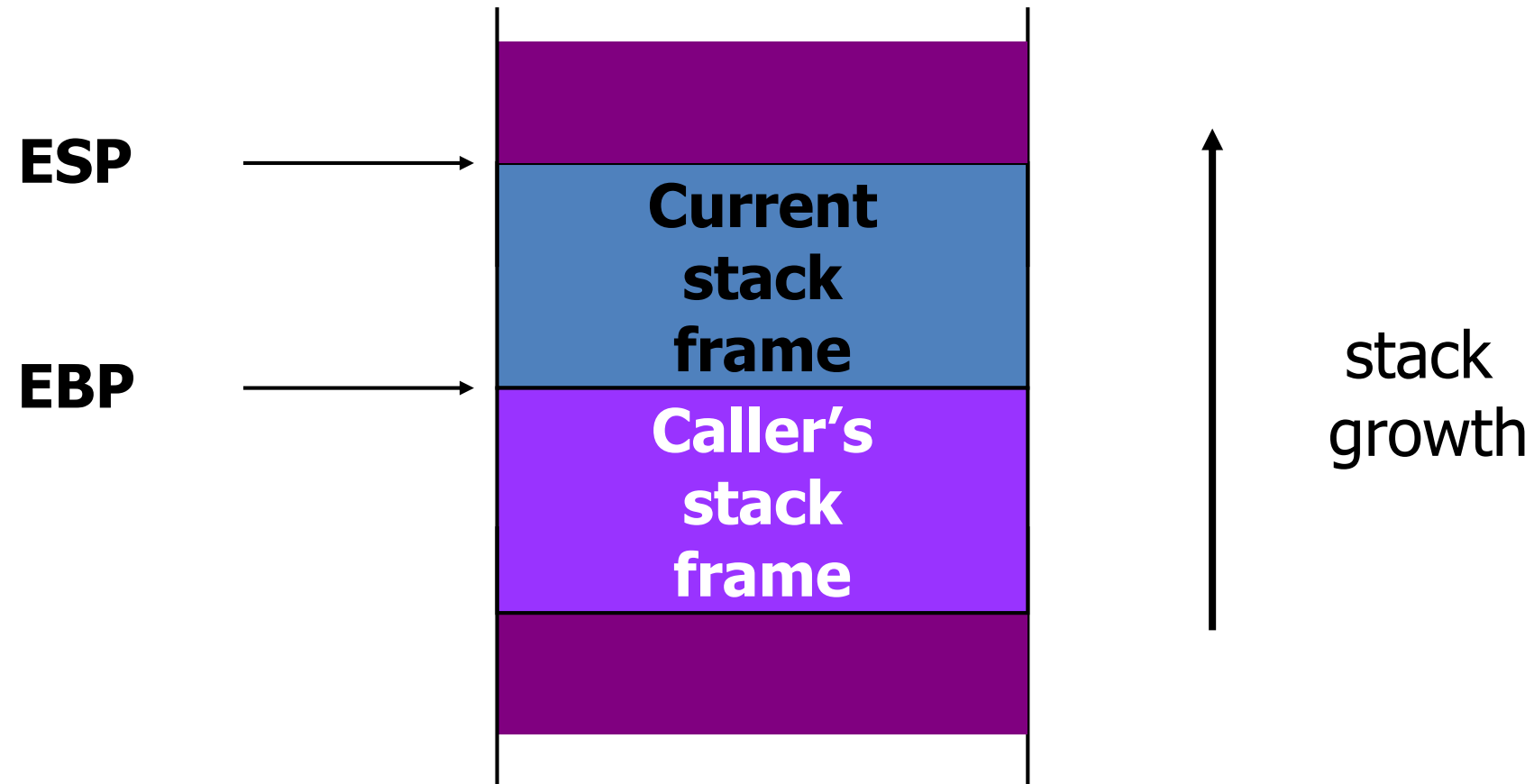
# Instruction Pointer

- **EIP**
  - ‘Instruction Pointer’
  - Contains the offset within the code segment of the next instruction to be executed
  - Cannot be accessed directly by software
  - The processor fetches instructions from the code segment, using a logical address that consists of the segment selector in the CS register and the contents of the EIP register. The EIP register contains the offset within the code segment of the next instruction to be executed.
  - The EIP register cannot be accessed directly by software; it is controlled implicitly by control transfer instructions (e.g., procedure calls or jump statements), interrupts, and exceptions.
  - The only way to read the EIP register is to execute a CALL instruction and then read the value of the return instruction pointer from the procedure stack.
  - The EIP register can be loaded indirectly by modifying the value of a return instruction pointer on the procedure stack and executing a return instruction.

# The Stack

The stack starts in high memory and grows toward low memory

- In the Intel architecture, the stack grows toward low memory (i.e., from addresses with larger numbers to addresses with smaller numbers).
- At the beginning of a routine, the base pointer is saved on the stack and then it is set to equal the current stack pointer. The stack pointer is then adjusted to make room for the subroutine's local variables. Such an area on the stack is called a "stack frame".





**Flags**

# Flags

- The status flags (bits 0, 2, 4, 6, 7, and 11) of the EFLAGS register indicate the results of arithmetic instructions, such as the ADD, SUB, MUL, and DIV instructions. The EFLAGS register is modified automatically by the CPU after mathematical operations. Generally you cannot access these registers directly.
- We will see that some instructions such as conditional jumps rely on the contents of the EFLAGS register to make decisions.
- The carry flag indicates an overflow condition for unsigned-integer arithmetic. It is also used in multiple-precision arithmetic.
- The adjust flag is used in binary-coded decimal (BCD) arithmetic.
- Of the six status flags we define on the slides, only the CF flag can be modified directly, using the STC, CLC, and CMC instructions. The bit instructions (BT, BTS, BTR, and BTC) copy a specified bit into the CF flag. See the Intel manual for details.

# The EFLAGS Register I

- Carry Flag –  $CF$  (bit 0)
  - **Set** if an arithmetic operation generates a carry or a borrow out of the most-significant bit of the result; **cleared** otherwise.
- Parity Flag –  $PF$  (bit 2)
  - **Set** if the least-significant byte of the result contains an even number of 1 bits; **cleared** otherwise.
- Adjust Flag –  $AF$  (bit 4)
  - Set if an arithmetic operation generates a carry or a borrow out of bit 3 of the result; **cleared** otherwise.

# The EFLAGS Register II

- Zero Flag –  $ZF$  (bit 6)
  - **Set** if the result is zero; **cleared** otherwise
- Sign Flag –  $SF$  (bit 7)
  - **Set** equal to the most-significant bit of the result, which is the sign bit of a signed integer
- Overflow Flag –  $OF$  (bit 11)
  - **Set** if the integer result is too large a positive number or too small a negative number (excluding the sign-bit) to fit in the destination operand; **cleared** otherwise

# Flags

- With respect to the SF flag, 0 indicates a positive value and 1 indicates a negative value.
- The OF flag indicates an overflow condition for signed-integer (two's complement) arithmetic.
- There are other flags contained in the EFLAGS register are:
- Bit 9 – Interrupt enable flag
- Bit 10 – Direction flag. This flag determines whether the EDI/ESI registers will be incremented or decremented during string operations.
- Bit 11 – Overflow flag
- Bits 12-13 – IOPL I/O Privilege level
- Bit 14 – NT Nested task flag
- Bit 16 – RF Resume flag
- Bit 17 – VM Virtual 8086 mode flag
- Bit 18 – AC Alignment check flag (486+)
- Bit 19 – VIF Virtual interrupt flag
- Bit 20 – VIP Virtual interrupt pending flag
- Bit 21 – ID flag
- Bits in the EFLAGS register that we have not listed are reserved by Intel.





**Memory**

# İşlemci Bellek İlişkisi

- İşlemci içerisinde önbellekte fazla miktarda bilgi saklanamaz ve geçici saklayıcılarda (kayıt ediciler) ise veri manipule (Transfer, Kontrol, ALU) işlevleri yerine getirilir.
- İşlemci gerekli verileri RAM'den okur ve işlem sonuçlarını RAM belleğe yazar.
- İşlemcinin işleyeceği verilerin RAM'in hangi bellek gözünden okunacağı ve işlem sonuçlarının hangi bellek gözüne yazılacağı programda verilmiş adreslerle belirlenir.
- Ram bellekte herbir veri programlarda belirlenen bir adreste bulunur.
- Ram bellek işlemcinin hesap defteri gibidir. Devamlı buradan bir şeyler okur, siler, yazar...
- CPU, hangi verinin hangi RAM bellek ile ilişkili olduğu yazılan program tarafından verilen adres hatları üzerinden Address Decoding Unit tarafından belirlenir.

# Byte - Memory sizes

- Byte: 8bit'lik veriyi temsil eder.
- 8-bitlik bellek gözünü refere (işaret) eder.
- We refer to this as a  $k \times n$  memory.
  - There are  $k$  *address lines*, which can specify one of  $2^k$  addresses.
  - Each address contains an  $n$ -bit word.

For example: a  $2^6 \times 16$  RAM contains  $2^{26} = 64\text{M}$  words, each memory cell is 16 bits long.

- The RAM would need 26 address lines.
- The total bit *storage capacity* is  $2^{26} \times 16 \text{ bit} = 2^{27} \times 8 \text{ bit} = 2^{27} \text{ byte} = 2^{30} \text{ bits}$ .
- Adres bus hat sayısı=26 adet.
- Hücresel uzunluğu word (16bit) olan belleğin indislenmesi : A25, A23, A22, ..., A1, A0

# Bellek

- **Bellek:** Mikroişlemcinin işlediği veya manipüle ettiği verileri saklayan transistör devre elemanlarından oluşan veri saklama özelliği olan ardışıl lojik kapılardır. Bellekler ikili sayı sisteminde 1 veya 0 formatlarından oluşan komut ya da veri gibi bilgileri saklar.
- **Bellekler: Verileri saklamak için kullanılır.** Mikroişlemcilere komutları ya da verileri transfer etmek için gerekli devrelerle birlikte bir depolama hücreleri koleksiyonudur.
- **Transistör:** Farklı bir noktadan elektron akışını kontrol ederek (akım, gerlimi) sinyalleri anahtarlama yada kuvvetlendirme işlevlerini yerine getiren, yarı iletken teknolojisinde üretilen bir devre elemanıdır. Transistörler atomik yapıda üretilebilmektedir.
- **Register:** Geçici özel amaçlı saklayıcılardır. Mikroişlemcinin içinde en temel saklayıcı birimdir.
- **ROM bellek:** Değiştirilmeyen komut ve verileri saklamak için kullanılır. Sadece okunur bellektir.
- **RAM bellek:** Hem yazılan hem de okunan bellektir.
- CMOS bellek
- **Ön Bellek (Cache):** Mikroişlemcinin sonraki adımlarda işleyeceği verilerin önceden transfer edilip hazırlandığı kendi ön belleğidir (SRAM).

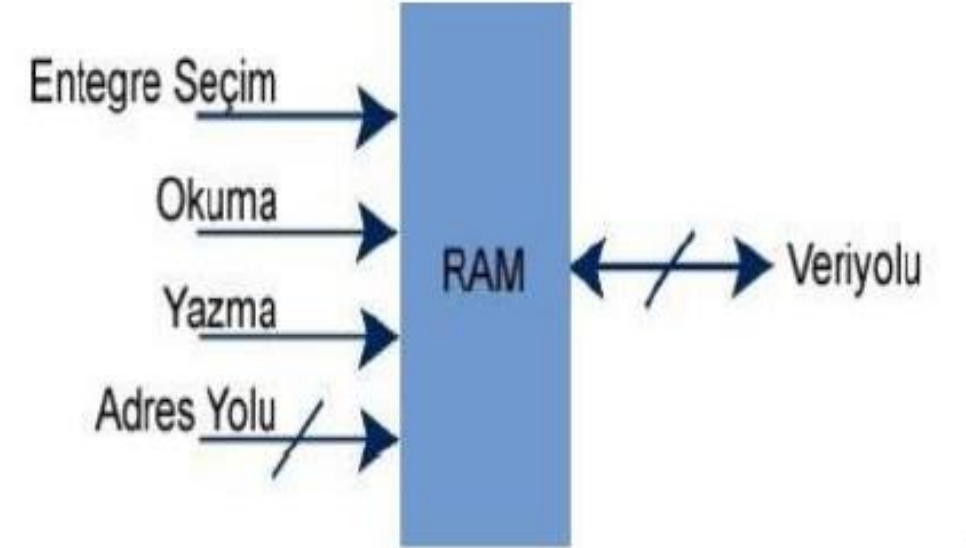
# Memory

- ROM
  - read-only memory
- EPROM
  - erasable programmable read-only memory
- Dynamic RAM (DRAM)
  - inexpensive; must be refreshed constantly
- Static RAM (SRAM)
  - expensive; used for cache memory; no refresh required
- Video RAM (VRAM)
  - dual ported; optimized for constant video refresh
- CMOS RAM
  - complimentary metal-oxide semiconductor
  - system setup information

# Ram Bellekler

RAM Bellekler : Mikro işlemcinin çalışması esnasında her türlü değişkenin üzerinde yer aldığı ve

- Geçici işlemlerin yapıldığı birimi RAM belleklerdir. Özel bir sıra takip etmeden herhangi bir adrese erişildiği için **rastgele erişimli bellek** (Random Access Memory)–RAM olarak isimlendirilir. RAM tipi entegreler hem yazmada hem okumada kullanıldıklarından CPU, bu entegreleri kontrol ederken okuma R (Okuma) ve W (yazma) sinyalleri göndermesi gerekir. Ayrıca entegrenin istendiği zaman aktif duruma geçmesinin sağlayacak entegre seçimi (CS =Chip Select) pini bulunmaktadır ve active low (aktif düşük, 0 Volt) ile çalışır. Her bir biti bir flip-flop devresi olan bu bellekler, yeni bir tetikleme işareti gelinceye kadar içindeki bilgiyi (0 veya 1'i) saklayabilme özelliği sebebiyle **çok düşük güç tüketimi** ile çalışmaktadır. Dışardan devreye bağlanan bir pil yardımıyla içindeki bilgileri çok uzun süreler boyunca saklayabilme imkânı vardır. Yüksek maliyetli olmaları sebebiyle çok yüksek kapasitelerde üretilmez.



# Rom Bellekler

1)ROM Bellekler : **Yalnız okunabilen** birimlere ROM (Read Only Memory) bellekler denir. Bu bellek elemanlarının en büyük özelliği enerjisi kesildiğinde içindeki bilgilerin silinmemesidir. ROM belleklere bilgiler üretim aşamasında yüklenir. Kullanıcıların bellek içindeki bilgileri değiştirmesi mümkün değildir.

2)PROM (Programlanabilir ROM) Bellekler : PROM'lar **bir kez programlanabilir**. Bu bellek elemanı entegre şeklindedir. Kaydedilen bilgiler enerji kesildiğinde silinmez. Üzerine program kodlarını veya verileri yazmak için PROM programlayıcı cihazlara ihtiyaç vardır. Bu bellek elemanının yapısında küçük sigorta telleri bulunur. Bellek hücrelerinde hepsi sağlam durumda bulunan sigortalar "1"i temsil eder. Yazılacak olan bilginin bit düzeninde "0"lara karşılık gelen hücredeki sigorta, küçük bir elektrik akımı ile aktarılır. Bu şekilde PROM programlanır.

3)EPROM (Silinebilir Programlanabilir ROM) Bellekler : "EPROM"lar bellek hücrelerine elektrik sinyali uygulanarak programlama işlemi yapılır. Kaydedilen bilgiler enerji kesildiğinde silinmez. "EPROM" içindeki programın silinmemesi için cam pencereli kısım ışık geçirmeyen bantla örtülmelidir. EPROM belleğe yeniden yazma işlemi yapmak için "EPROM" üzerindeki bant kaldırılıp ultraviyole altında belirli bir süre tutmak gerekir. Bu şekilde içindeki bilgiler silinebilir. Böylece tekrar programlanabilir hâle gelen ürün tekrar tekrar farklı programların denenmesi ve cihazın çalıştırılması için kullanılabilir.

4)EEPROM (Elektriksel Yolla Değiştirilebilen ROM) Bellekler : Üzerindeki bilgiler, **elektriksel olarak yazılabilen ve silinebilen** bellek elemanlarıdır. "EEPROM"u besleyen enerji kesildiğinde üzerindeki bilgiler kaybolmaz. "EEPROM"daki bilgilerin silinmesi ve yazılması için özel silme ve yazma cihazlarına gerek yoktur. Programlayıcılar üzerinden gönderilen elektriksel sinyalle programlanır. "EEPROM"la aynı özellikleri taşıyan fakat yapısal olarak farklı ve daha hızlı olan, elektriksel olarak değiştirilebilir "ROM"lara Flash Bellek denir.

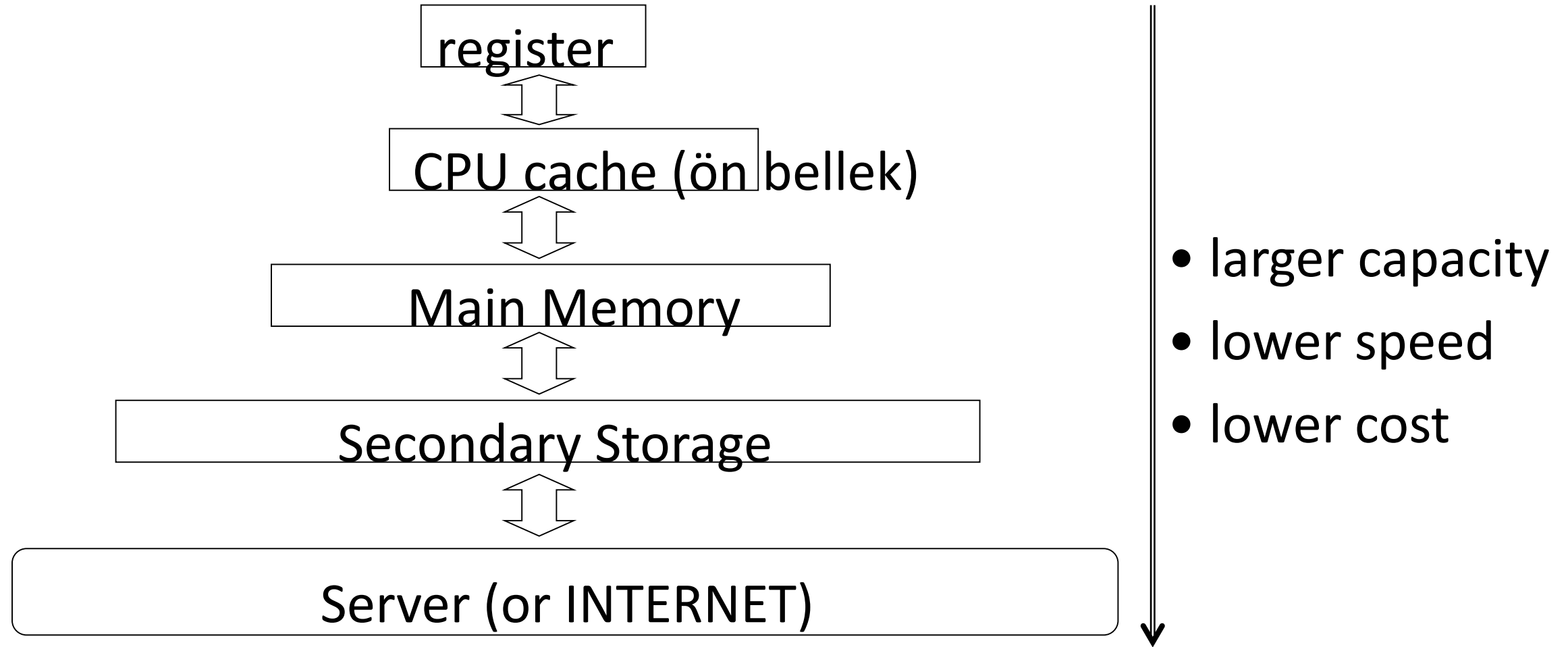
# Comparison of Memory Types

---

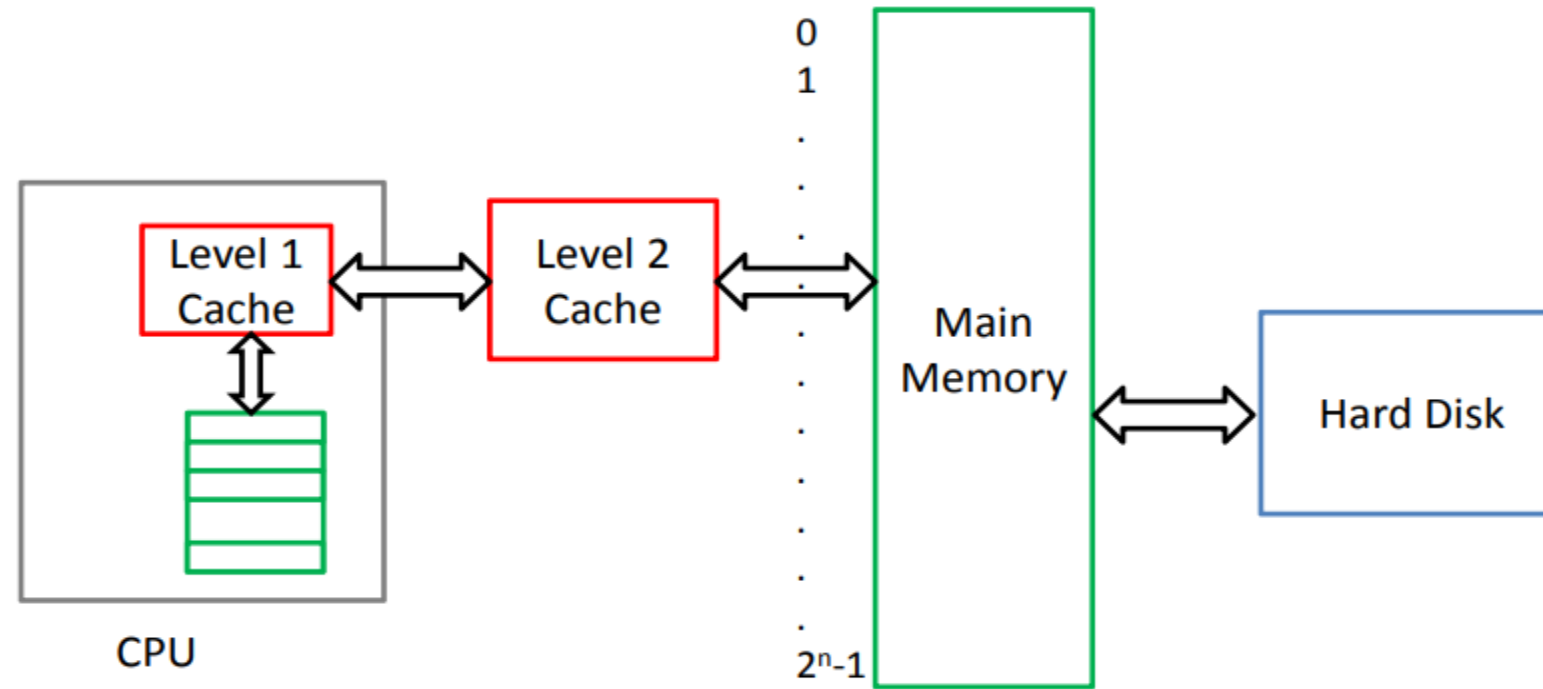
- DRAM
  - very high density → cheap data cache in computers
  - must be periodically refreshed → slower than SRAM
  - volatile; no good for program (long term) storage
- SRAM (basically a Latch)
  - fastest type of memory
  - low density → more expensive
    - generally used in small amounts (L2 cache) or expensive servers
- EEPROM
  - slow/complex to write → not good for fast cache
  - non-volatile; best choice for program memory
- ROM
  - hardware coded data; rarely used except for bootup code
- Register (flip flop)
  - functionally similar to SRAM but less dense (and thus more expensive)
  - reserved for data manipulation applications



# CPU – Bellek erişim özelliğine göre belleklerin sıralanması



# Memory Hierarchy



# Cache Memory

- High-speed, expensive, static RAM both inside and outside of the CPU.
  - Level-1 cache: inside the CPU
  - Level-2 cache: outside the CPU
- Cache hit: when data to be read is already in cache memory
- Cache miss: when data to be read is not in cache memory.

# Latch

- Yazma anında Clock sinyalini yüksel kenar (Tetikleme) gelmediği sürece çıkış değerini korur.
- Bir elektronik devreye çalışma gerilimi uygulandığı sürece durumunu ve buna bağlı olarak çıkışındaki değeri devamlı olarak koruyan multivibratör çeşidi Flip Flop olarak isimlendirilir. FF olarak sembolize edilir. Lojik kapılar ile oluşturduğumuz flip floplar lojik devrelerde en önemli bellek elemanlarıdır.
- FF'ler için çift kararlı multivibratör (bistable multivibratör) terimi de kullanılır. FF'lerin tetikleme girişine uygulanan kare veya dikdörtgen şeklindeki sinyaller, tetikleme sinyali/palsi olarak adlandırılır. FF devresi tetikleme sinyalinin pozitif kenarında tetikleniyorsa pozitif kenar tetikleme negatif kenar tetiklemeli devre olarak tanımlanır.
- Birçok FF türü vardır. Bunlardan en çok kullanılanları:
  - R-S(reset-set) tipi FF
  - Tetiklemeli(clocked) R-S FF
  - J-K Tipi FF
  - Master Slave tipi FF
  - D (data) tipi FF
  - T(Toggle) tipi FF'dir

# Memory Management

- Real-address mode
- Calculating linear addresses
- Protected mode
- Multi-segment model
- Paging
- Gerçek adres modu
- Doğrusal adresleri hesaplama
- Korumalı mod
- Çok segmentli model
- Sayfalama

# Memory Types

- Ana Bellek: Ram, Rom, CMOS (BIOS)
- Cache
- Dynamic ram
- Static ram
- Flash memory
- Memory sticks
- Virtual memory
- Video memory
- Bios
- Hard Disk
- Belleklerin üç görevi vardır:
  - İşlenecek veriyi depolar.
  - Veriyi işleyen komutları (programları) depolar.
  - İşlenmiş, iletişim veya çıkış aygıtlarına gönderilmek için bekleyen veriyi depolar.

# Storage Systems

- Units of Storage:
  - 1 bit
  - 8 bits = 1 byte
  - 1kbyte =  $2^{10}$  = 1024 bytes
  - 1Mbyte =  $2^{20}$  = 1048576 bytes
- Memory (RAM, ROM)
- Optical Disks
- CD ROM
- Magnetic Disk
- Floppy Disk
- Flash Bellek
- Hard Disk
- SSD

# Memory Segmentation

---

- A memory segment is a block of  $2^{16}$  (64K) bytes.
- Each segment is identified by a segment number
  - Segment number is 16 bits .
- A memory location is specified by an offset within a segment.
- Logical address: segment:offset
  - A000:4872h means offset 4872h within segment A000h.
- Physical address: segment \* 10H + offset
  - $A000h * 10h + 4872h = A4872h$  (20-bit address)



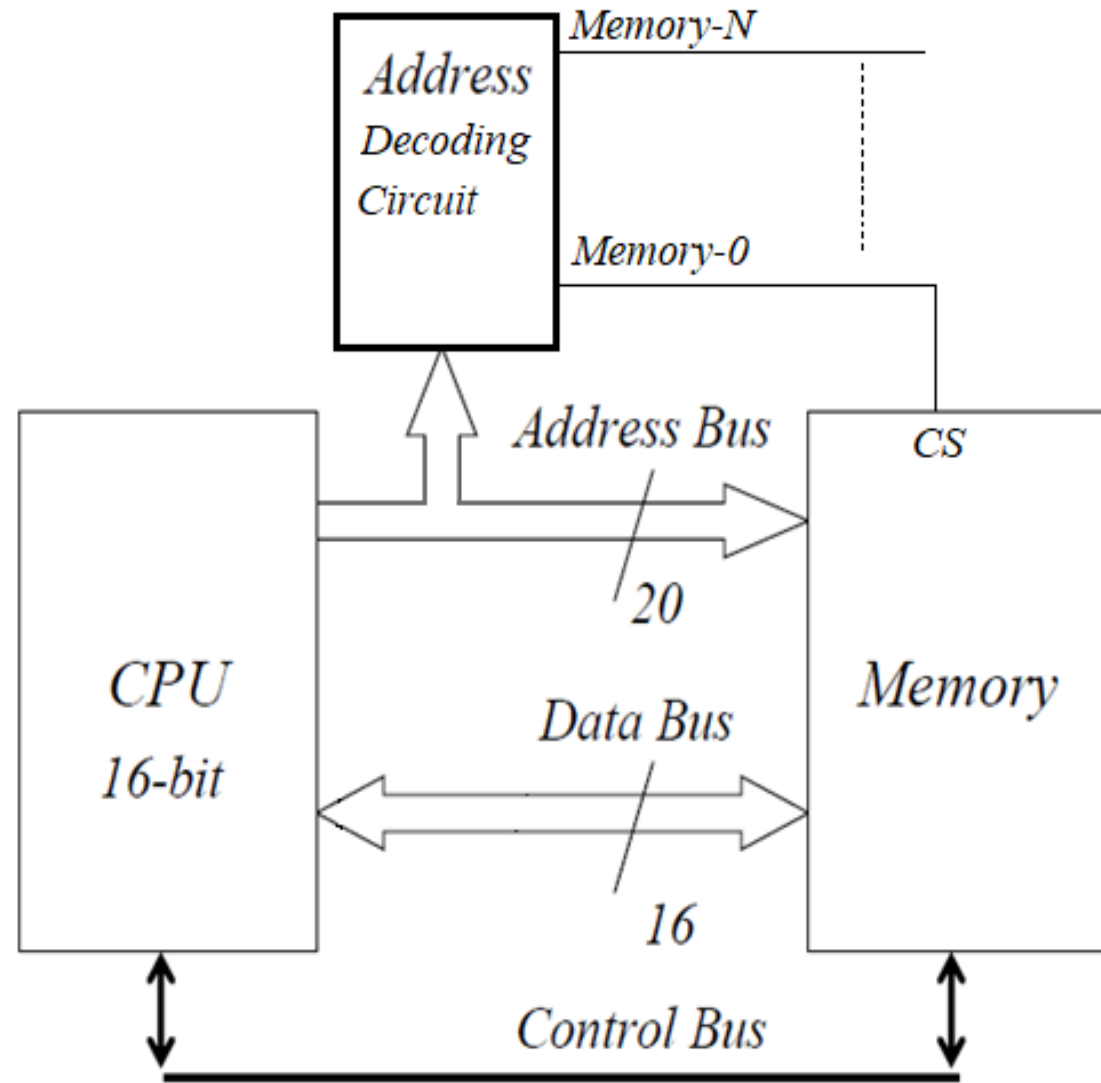
# Paging

- Supported directly by the CPU
- Divides each segment into 4096-byte blocks called **pages**
- Sum of all programs can be larger than physical memory
- Part of running program is in memory, part is on disk
- **Virtual memory manager** (VMM) – OS utility that manages the loading and unloading of pages
- **Page fault** – issued by CPU when a page must be loaded from disk



# **Address Decoding Unit** **(Bellek Seçme)**

# CPU – Address Decoding Unit



- Adres Bus CPU'dan belleklere ve Address Decoding Devresine giden adres bus'a ait hatlardan oluşur.

Adres Bus Hatları paraleldir, grup halinde çalışır; herbir hat üzerinde elektriksel sinyal olarak ikili durum (bit) 0 ya da 1 değeri mevcuttur.

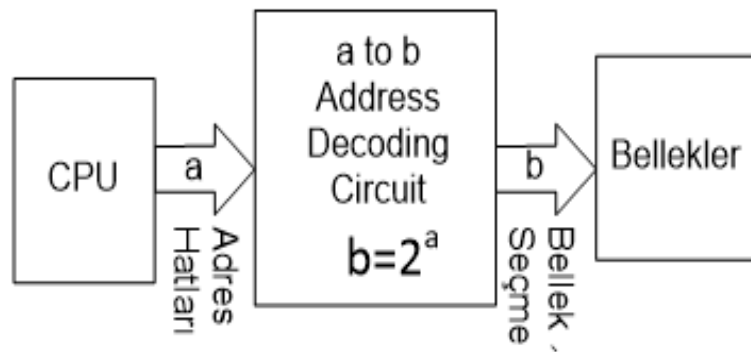
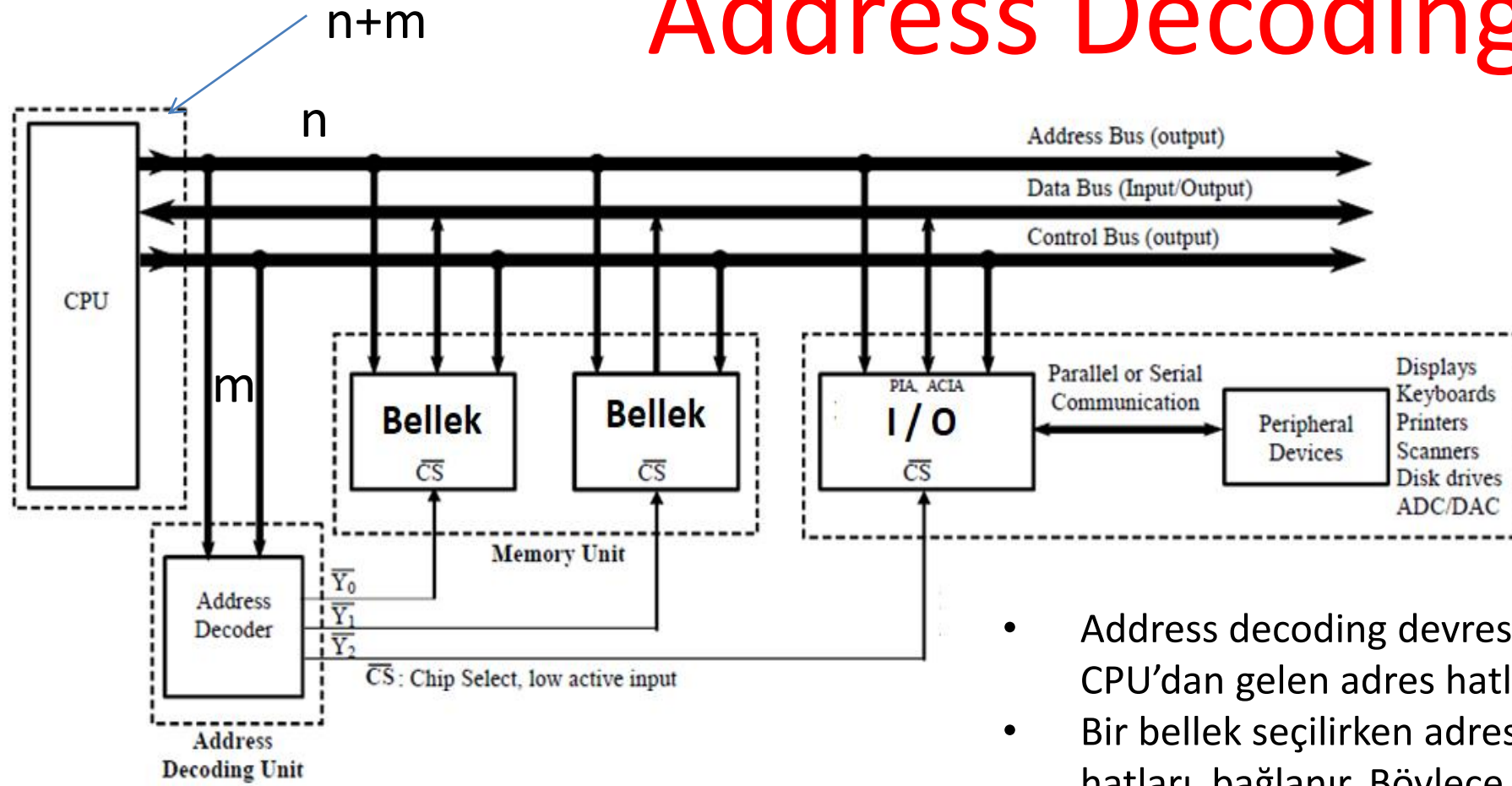
CPU'dan çıkan Adres Bus hat sayısı CPU'nun erişebileceği maksimum bellek kapasitesini verir. Buradaki 20 adet adres bus hat sayısı,  $n=20$  ise adreslenecek maksimum bellek kapasitesi  $=2^{20} = 1\text{Mbyte}$  dir.

Address Decoding Devresi erişilecek bellekleri seçer.

Doğrudan belleklere giden adres bus hatları belleğin kapasitesini belirler, bellek gözü seçer.

Bellekler adres dekoding devresi üzerinden seçildiği için bellek ve bellek gözlerinin çatışması mümkün olmaz.

# Address Decoding Unit



- Address decoding devresinin çıkışları bellek seçer. Seçilen belleğin gözüne CPU'dan gelen adres hatları yardımıyla erişilir.
- Bir bellek seçilirken adres decoding devresinin girişine CPU'dan gelen adres hatları bağlanır. Böylece adres decoding devresinin çıkışları bellekleri seçmede kullanılır.
- Seçilecek bellek sayısı=Address decoding devresi çıkış sayısı= $2^m$  dir. Burada m CPU'dan address decoding devresinin girişine gelen adres hattı sayısıdır.
- Herbir belleğin kapasitesini belirleyen (Kapasite= $2^n$ ) n adet hat CPU'dan gelir.
- Amaç aynı anda bir belleğin ilgili veri gözünün seçilmesidir.
- Adres decoding devresinin çıkış sayısı bellek sayısına eşit ya da büyük olmak zorundadır. ( $2$ 'nin üssüne eşit olmayan bellek sayılarında)

# Örnek:

CPU'dan adres dekoding devresine 4 adet adres bus hattı, maksimum belleğe ise 16 adet adres bus hattı gitmektedir.

- CPU'dan çıkan toplam adres hattı sayısı nedir?  $4+16=20$
- CPU'nun maksimum adresleme kapasitesi kaç byte'dır?  $2^{20}$  byte=1Mbyte
- CPU'dan adres dekoding devresine 4 adet adres hattı gidiyorsa adres dekoding devresi üzerinden kaç adet bellek ya da I/O birimi seçilir?  $2^4=16$ ; Belleklerin ya da I/O birimleri indisleri: U0, U1, U2, ... , U14, U15
- Maksimum bellek kapasitesi olan belleğin adres bus hat sayısı nedir? 16; indis: A15, A14, ... , A1, A0
- Maksimum bellek kapasitesi olan belleğin kapasitesi kaç byte'dır?  $2^{16}=64$ Kbyte
- 11. belleğin 1047. gözünün adres bus hat durum nedir?  
U11=(1011) - A19, A18, A17, A16 Not: maksimum belleğinden maksimum indisinden sonraki indisinden devam eder.  
 $1047=2^{10}+2^4+2^2+2^1+2^0$   
seçilen bellek gözünün adres indisi: (0100 0001 0111)b; indis:A11, A10, A0, ... , A1, A0
- CPU'nun bellek ve bellek gözünü seçmede kullanacağı adres bus indisi:  
A19,A18,A17,A16, A15,A14,A13, A12, A11, A10, A9,A8,A7,A6,A5,A4,A3, A2, A1, A0  
(1011 0000 0100 0001 0111)b = (B0417)h
- DS değeri nedir? (B000)h; fiziksel adres hesaplanırken, segment adresin sonuna 0h konur. B000 elde edilir.
- Data transfer komutunu yazınız,  
Mov Ax, B000h  
Mov Ds, Ax; Data segmente doğrudan veri aktarılmaz.  
Mov AL, DS: [417h]; bellek gözünün fiziksel adresi= DS+417h= B0000+417h=B0417h

# Örnek

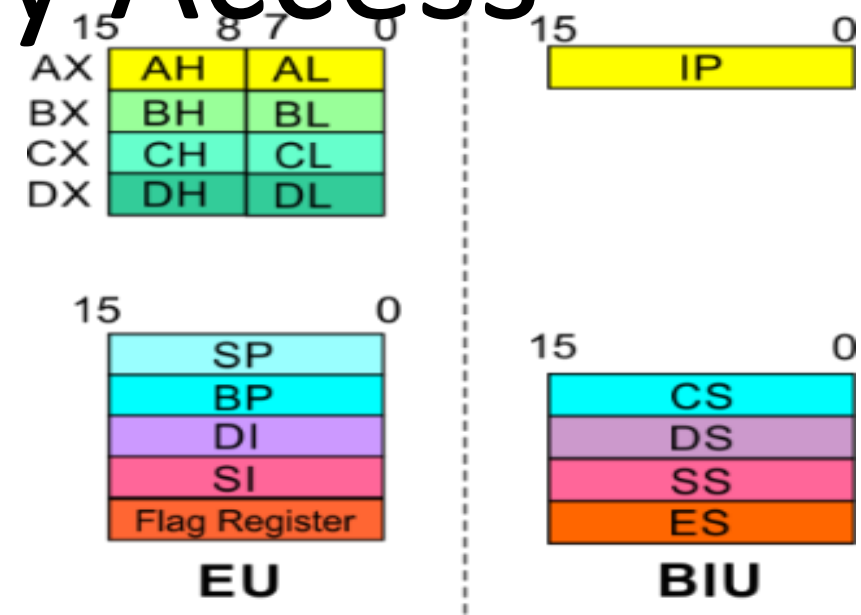
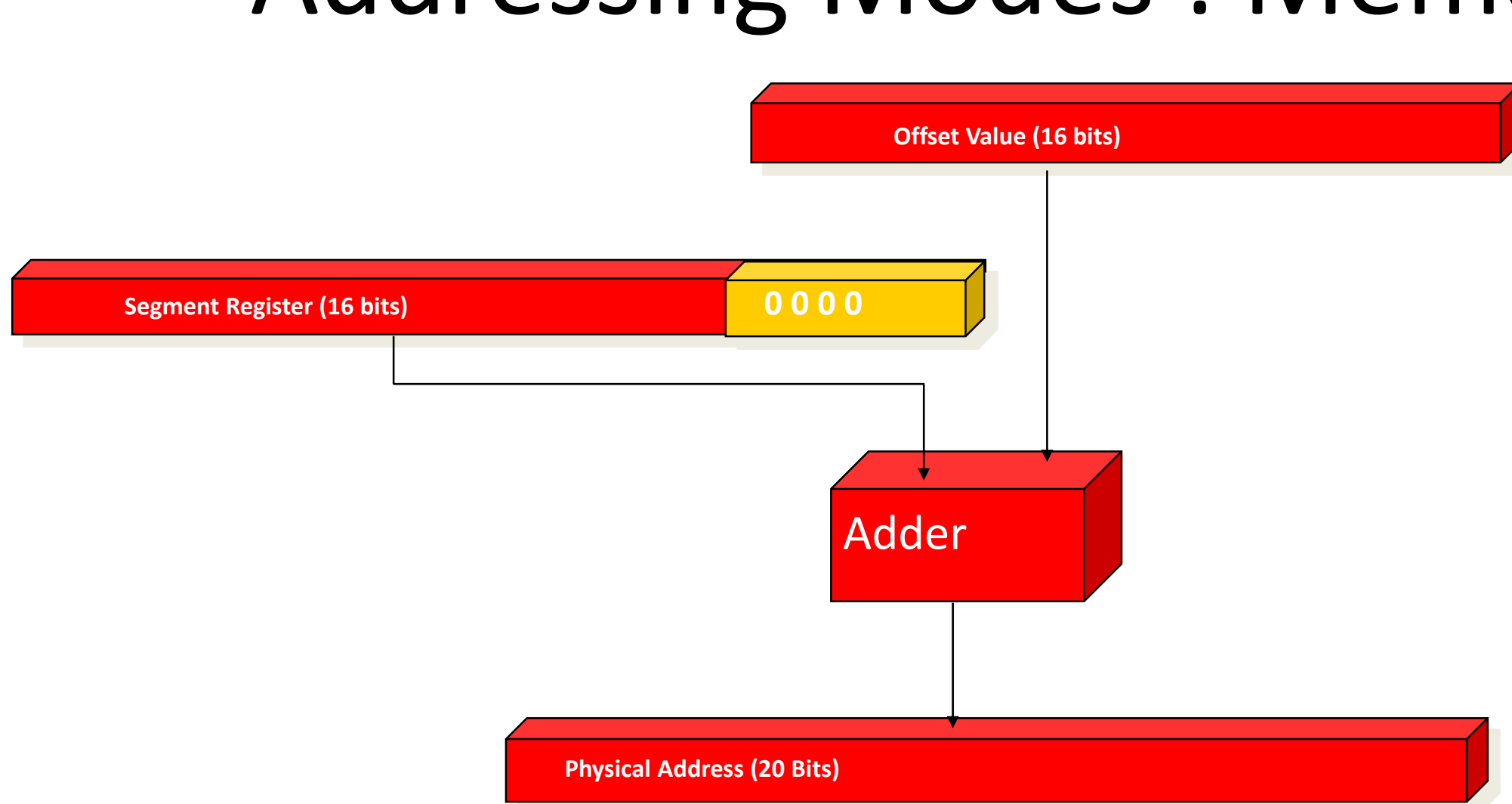
- I/O sayısı: 21, Bellek sayısı: 32, Maksimum bellek kapasitesi: 37Gbyte ise CPU'dan çıkacak adres hattı sayısı nedir?
- Toplam birim sayısı,  $N=21+32=53$ , U0, ..., U52
- Bir anda I/O ve Belleklerden birini seçmek için CPU'dan çıkacak Adres Hattı Sayısı  $m$  ise, toplam birim sayısı,  $N=2^m$  dir. Bu ifadeden  $m=6$  bulunur.
- Maksimum Bellek Kapasitesi=37Gbyte, 64Gbyte alınır.
- $64\text{Gbyte}=2^6 * 2^{30}=2^{36}$  Byte
- Maksimum belleğe bellek gözlerini seçmek için gidecek adres hattı sayısı,  $k=36$  dır.
- İndisleme: A35, A34, ... , A1, A0. Not: Diğer bellekler ne kadar adres hattına ihtiyaç duyarlarsa o kadar adres hattı paralel gider.
- Toplam adreshattı sayısı = Maksimum belleğe bellek gözlerini seçmek için gidecek adres hattı sayısı + Bir anda I/O ve Belleklerden birini seçmek için CPU'dan çıkacak Adres Hattı Sayısı
- Toplam adres hattı sayısı =  $k+m= 36 + 6 =42$
- İndisleme: A41, A40, ... , A1, A0

A41	-	-	-	A36	A35	-	-	-	A1	A0
Toplam 6 adet					Toplam 36 adet					
Bir anda I/O ve Belleklerden birini seçmek					Maksimum bellek ve diğer bellek gözlerini seçmek					

## Adres Dekoding Circuit:

- Giriş - Çıkış
- 000000 – U0
- 000001 – U1
- .....
- 110100 – U52
- .....
- 111111 – U63

# Addressing Modes : Memory Access



# Örnek:

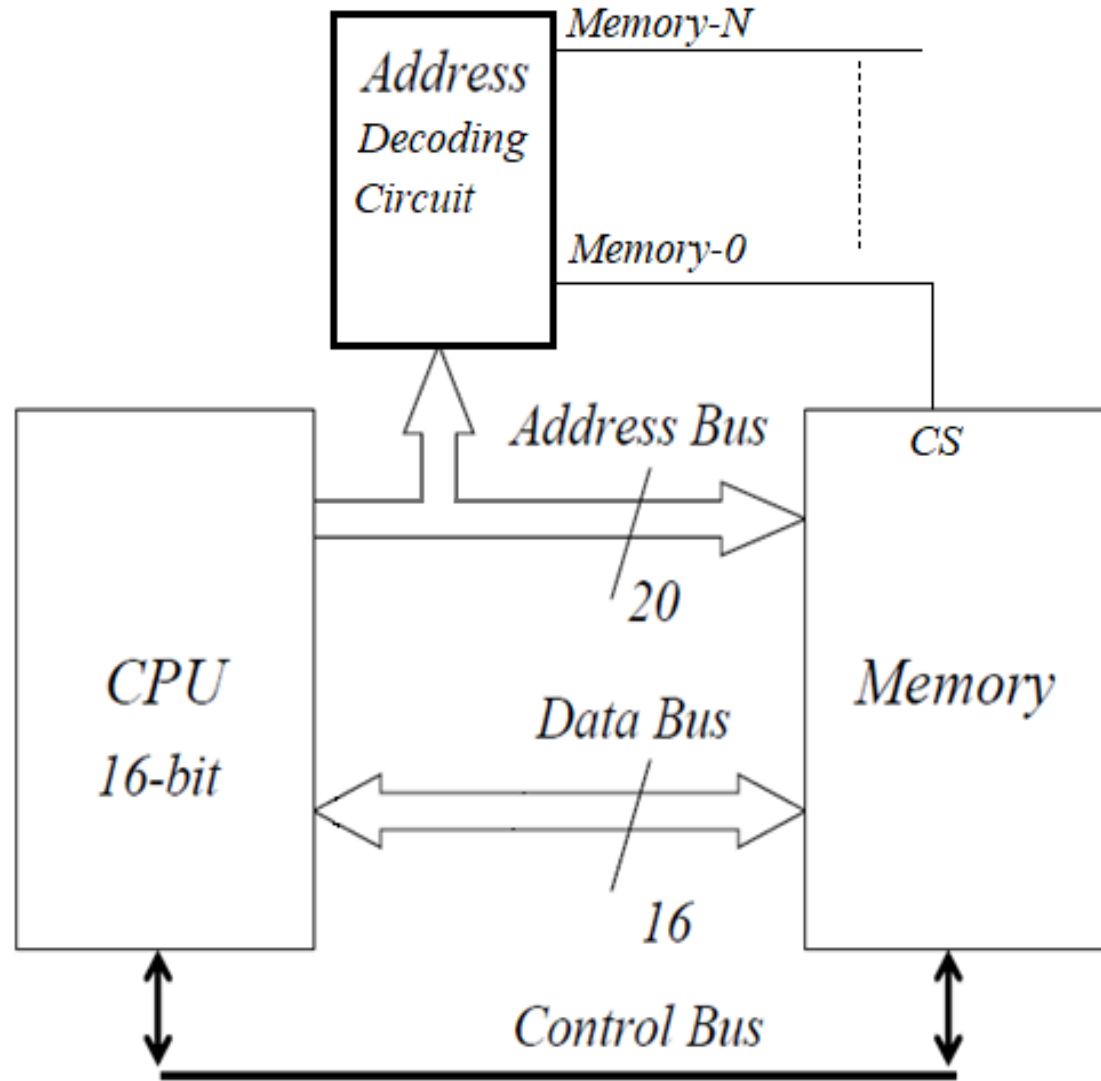
- Indis ve Gösterge:
  - CS: IP
  - DS, ES: BX, SI, DI
  - SS: SP, BP
- 0AB234h fiziksel adresi olan bellek gözüne 55h değeri yazılacaktır. DS 16 bittir, 20bit'lik fiziksel adresin anlamlı 4 bit'i alınacaktır.
- DS'in içeriği nedir? DS: A000h
- İndis ya da ofset adresi BX register'ın içeriği nedir? BX: B234h
- Adresleme yazılımı nedir?
  - Mov AX, 0A000h
  - Mov DS, Ax
  - Mov Bx, 0B234h
  - Mov [Bx], 55h





# Data Bus

# CPU – Data Bus



- CPU'dan çıkıp belleklere gelen Adres Bus hatları ile bellek ve bellek gözü ya da I/O birimi seçildikten sonra, CPU Data bus üzerinden, bellek gözüne 1 clock periyodunda data bus hat sayısı kadar veri yazar ya da okur. Hatlar paraleldir ve grup olarak çalışır. Herbir hat üzerinde 0 ya 1 ikili durum mevcuttur.

Data Bus hat sayısı CPU'nun veri transfer özelliğini verir. CPU, 16 bit dendiğinde Data Bus hat sayısı 16 olan bir CPU'dan bahseder.

Word olarak 16 bit yazılıp okunur. 2 Byte'a denk gelir.

Control bus hatları da paraleldir. Grup olarak çalışmaz. Ayrık çalışır. Herbir hat üzerinde 0 ya 1 ikili durum mevcuttur.

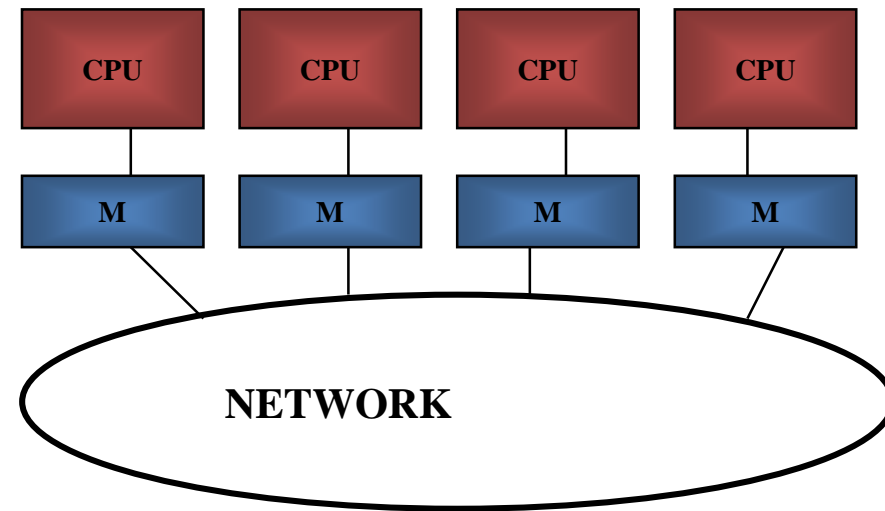


# **Parallel Computing**

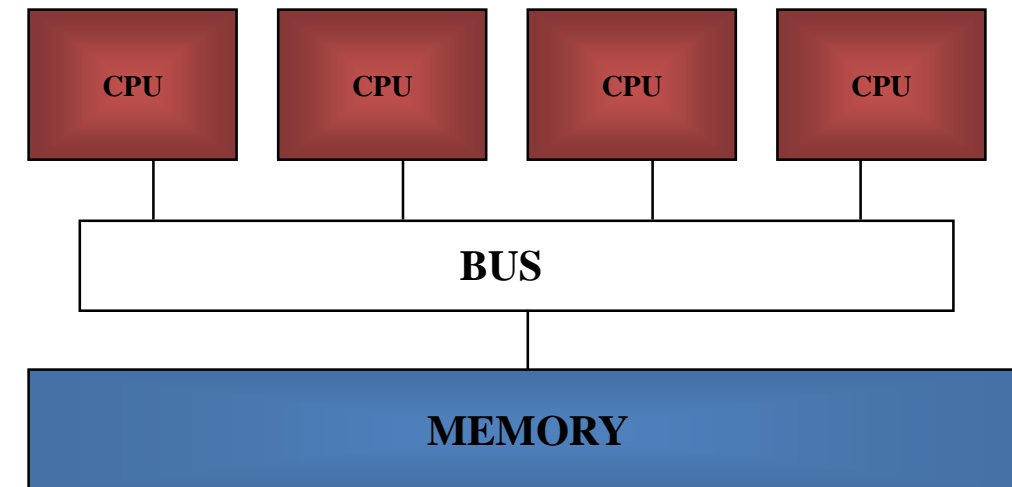
# What is Parallel Computing ?

- Verileri işlemek için birden fazla işlemcinin eş zamanlı kullanımı
  - Aynı programın birçok işlemcide çalıştırılması.
  - Her işlemcide birçok program çalıştırılıyor.

# Shared/Distributed-Memory Architecture



**Distributed memory** - each processor has its own local memory. Must do message passing to exchange data between processors.  
(examples: Emerald, Topsail Clusters)



**Shared memory** - single address space. All processors have access to a pool of shared memory. (examples: Chastity/zephyr, happy/yatta, cedar/cypress, sunny)  
Methods of memory access : Bus and Crossbar

# Advantages of Parallelization

- Fiyat/Performans Oranı Açısından Daha Ucuz
- Eşdeğer pahalı tek işlemcili makinelerden daha hızlı
- Daha büyük sorunları ele alın
- Daha ölçeklenebilir: Belirli bir programın performansı, büyük bir makinede çalıştırılarak geliştirilebilir
- Daha güvenilir: Teorik olarak işlemciler arızalanırsa başkalarını kullanabiliriz

# Multitasking

- İşletim sistemi aynı anda birden fazla programı çalıştırabilir.
- Aynı program içinde birden çok yürütme iş parçacığı.
- Zamanlayıcı yardımcı programı, çalışan her programa belirli bir miktarda CPU zamanı atar.
- Görevlerin hızlı geçişi
  - tüm programların aynı anda çalıştığı yanılsaması verir
  - işlemci görev değiştirmeyi desteklemelidir.



# **Assembly Language Coding**



# Assembly Language Programming

- Assembly Language
  - Processor-specific, low-level programming language
    - exposes most of processor features to the programmer
  - Describes
    - Data types and operations
      - Size: Byte, Word, etc
      - Meaning: integer, character
    - Storage
      - Registers
      - Memory
    - Specific instructions

# Machine/Assembly Language

---

## ■ Machine Language:

- Set of fundamental instructions the machine can execute
- Expressed as a pattern of 1's and 0's

## ■ Assembly Language:

- Alphanumeric equivalent of machine language
- Mnemonics more human-oriented than 1's and 0's

## ■ Assembler:

- Computer program that transliterates (one-to-one mapping) assembly to machine language
- Computer's native language is machine/assembly language

# Assembly Language

- High level language
  - C, C++, Python, Java language
  - Abstraction
  - Human friendly
    - `a = b + c;`
- Assembly language
  - X86 Assembly language for Intel machine
  - Hard for human to read
    - `Mov eax, b`
    - `Mov ebx, c`
    - `Add eax, ebx`
    - `Mov a, eax`
  - High efficiency to use registers
- Machine language
  - Machine code for a specific machine
  - Machine readable
    - `00110100100100100100111010101010`
    - `10100100100100100111010101010111`
    - `10101010101111101010101011110010`

# Kritik Noktalar

- Registers
- Data Types and Representation
- Variable Declaration
- Memory Segmentation
- Memory Addressing Modes
- Instructions
  - Data flow instructions
  - Arithmetic instructions
  - Bit manipulation instructions
  - Flow control instructions



# **Data Representation Basics**

# Negatif Sayılar

- Byte: 8 bit data tanımlar; 8 bitlik bellek gözünü işaret eder.

- Örnek db -4

$$(4)_d = (0000\ 0100)_b$$

$$(-4)_d = \text{Binary } (4) \text{ Tersisi } +1 :$$

$$(-4)_d = 1111\ 1011 + 1 = 1111\ 1100 = \text{FC}$$

- Örnek dW -4

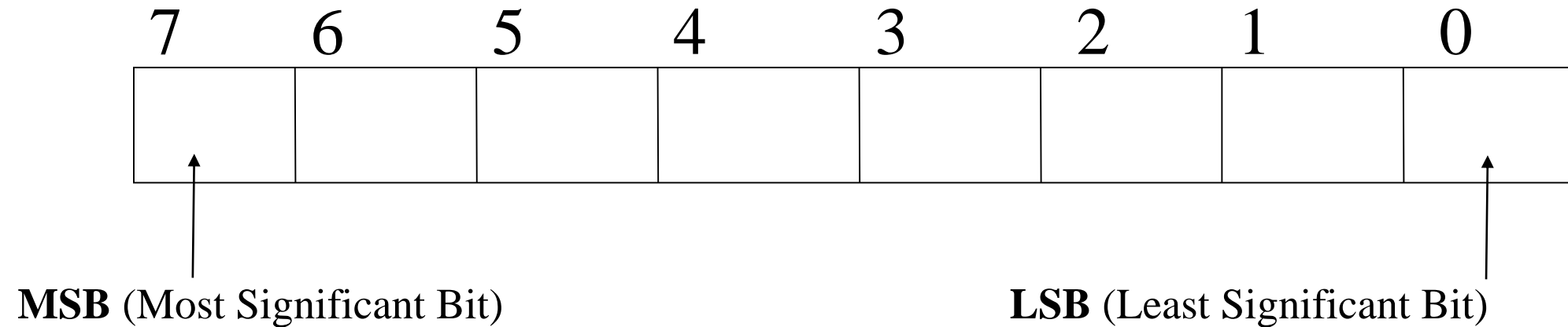
$$(4)_d = 0000\ 0000\ 0000\ 0100$$

$$(-4)_d = \text{Word}(4) \text{ Tersisi } +1$$

$$(-4)_d = 1111\ 1111\ 1111\ 1011 + 1 = 1111\ 1111\ 1111\ 1100 = \text{FFFC}$$

# Data Representation Basics

- **Bit** - the basic unit of information:  
(true/false) or (1/0)
- **Byte structure**: A byte has 8 bits



# Data Representation

## Numbers

- Eğer hiçbir tanım yapılmamış ise ondalık sayı sistemi kullanılıyor demektir.
- 11011 decimal
- (11011)b binary
- 64223 decimal
- (-21843)D decimal
- 1,234 illegal, contains a nondigit character
- (1B4D)H hexadecimal number
- (1B4)D illegal hex number, does not end with “H”
- (FFFF)H illegal hex numbe, does not begin with with digit
- (0FFFF)H hexadecimal number

**Signed numbers represented using 2's complement.**



# Data Representation

## Characters

- must be enclosed in single or double quotes
- e.g. “Hello”, ‘Hello’, “A”, ‘B’
- encoded by ASCII code
- **‘A’ has ASCII code 41H**
- **‘a’ has ASCII code 61H**
- **‘0’ has ASCII code 30H**
- **Line feed has ASCII code 0AH**
- **Carriage Return has ASCII code 0DH**
- **Back Space has ASCII code 08H**
- **Horizontal tab has ASCII code 09H**

# Intrinsic data types

- **BYTE, SBYTE**
  - 8-bit unsigned integer; 8-bit signed integer
- **WORD, SWORD**
  - 16-bit unsigned & signed integer
- **DWORD, SDWORD**
  - 32-bit unsigned & signed integer
- **QWORD**
  - 64-bit integer
- **TBYTE**
  - 80-bit integer

# Initialized data

Pseudo-instruction	<size> filed	<size> value
DB	byte	1 byte
DW	word	2 bytes
DD	double word	4 bytes
DQ	quadword	8 bytes
DT	tenbyte	10 bytes
DDQ	double quadword	16 bytes
DO	octoword	16 bytes

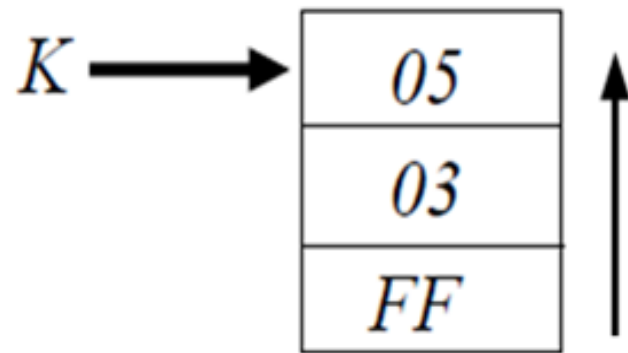
## Examples:

```
var1:    db    0x55    ; define a variable 'var' of size byte, initialized by 0x55
var2:    db    0x55,0x56,0x57; three bytes in succession
var3:    db    'a'      ; character constant 0x61 (ascii code of 'a')
var4:    db    'hello',13,10,'$' ; string constant
var5:    dw    0x1234          ; 0x34 0x12
var6:    dw    'A'           ; 0x41 0x00 – complete to word
var7:    dw    'AB'          ; 0x41 0x42
var8:    dw    'ABC'         ; 0x41 0x42 0x43 0x00 – complete to word
var9:    dd    0x12345678     ; 0x78 0x56 0x34 0x12
```

Soru: Var1 ile Var2 arasındaki fark nedir? Var1: tek değişken, Var2: 3 elemanlı dizi

# Byte Variables

- **Assembler directive format defining a byte variable**
  - name DB initial value
  - a question mark (“?”) place in initial value leaves variable uninitialized
- **I DB 4 define variable I with initial value 4**
- **J DB ? Define variable J with uninitialized value, bellekte deęişken yer ayırır.**
- **Name DB “Course” allocate 6 bytes for Name**
- **K DB 5, 3, -1 allocates 3 bytes**



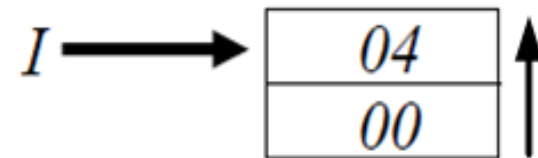
# Word Variables

---

## ■ Assembler directive format defining a word variable

- name DW initial value

### ■ I DW 4



### ■ J DW -2



### ■ K DW 1ABCH



### ■ L DW "01"



Değişken tanımında kaç byte? Kaç adet 4 bit?

- Hex: 4 bit tanımlar.
- 0 yüzden 4 bir bir hex sayı sistemi ile tanımlanır.

• Örnek:

- DW: 16 bit. 2 byte, 4 adet 4 bit.

- I DW 4: (0004)h

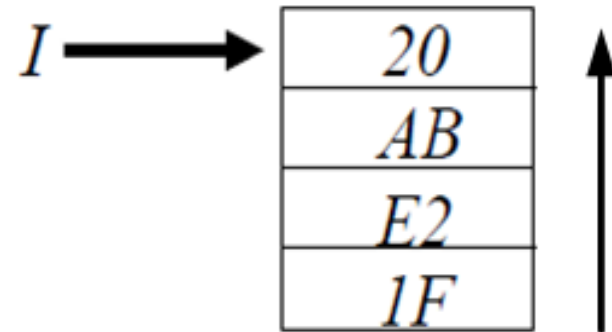
# Double Word Variables

---

- Assembler directive format defining a word variable

- name DD initial value

- I DD 1FE2AB20H



- J DD -4



# **Instruction Types**

# INSTRUCTION SET OF 8086

- The 8086 instructions are categorized into the following main types
- **Data copy /transfer instructions:** These type of instructions are used to transfer data from source operand to destination operand. All the store, load, move, exchange input and output instructions belong to this category.
- **Arithmetic and Logical instructions:** All the instructions performing arithmetic , logical, increment, decrement, compare and ASCII instructions belong to this category.
- **Branch Instructions:** These instructions transfer control of execution to the specified address. All the call, jump, interrupt and return instruction belong to this class.
- **Loop instructions:** These instructions can be used to implement unconditional and conditional loops. The LOOP, LOOPNZ , LOOPZ instructions belong to this category.
- **Machine control instructions:** These instructions control the machine status. NOP, HLT, WAIT and LOCK instructions belong to this class.
- **Flag manipulation instructions:** All the instructions which directly effect the flag register come under this group of instructions. Instructions like CLD, STD, CLI, STI etc., belong to this category of instructions.
- **Shift and Rotate instructions:** These instructions involve the bit wise shifting or rotation in either direction with or without a count in CX.
- **String manipulation instructions:** These instructions involve various string manipulation operations like Load, move, scan, compare, store etc..



# Instruction Types

- **Data transfer instructions**
  - Transfer information between registers and memory locations or I/O ports.
  - MOV, XCHG, LEA, PUSH, POP, PUSHF, POPF, IN, OUT.
- **Arithmetic instructions**
  - Perform arithmetic operations on binary or binary-codeddecimal (BCD) numbers.
  - ADD, SUB, INC, DEC, ADC, SBB, NEG, CMP, MUL, IMUL, DIV, IDIV, CBW, CWD.
- **Bit manipulation instructions**
  - Perform shift, rotate, and logical operations on memory locations and registers.
  - SHL, SHR, SAR, ROL, ROR, RCL, RCR,
  - Logical operations: NOT, AND, OR, XOR, TEST.

# Instruction Types

- **Control transfer instructions**

- Control sequence of program execution; include jumps and procedure transfers.
  - JMP: kontrolsuz istenilen yere gider.
  - CMP: İki değişkenin çıkarma işlemi sonucunda bayraklar etkin olur: zero, sign, carry, ...
- JG, JL, JE, JNE, JGE, JLE, JNG, JNL, JC, JS, JA, JB, JAE, JBE, JNB, JNA, JO, JZ, JNZ, JP, JCXZ,

- **Loop instructions:** LOOP, LOOPE, LOOPZ, LOOPNE, LOOPNZ; Cx register değeri her loop işleminin sonunda 1 eksilir, Cx=0 olduğunda Loop'dan çıkar.
- Alt Program: CALL, RET.

- **String instructions**

- Move, compare, and scan strings of information.
- MOVS, MOVSB, MOVSW, CMPS, CMPSB, CMPSW. SCAS, SCASB, SCASW, LODS, LODSB, LODSW, STOS, STOSB, STOSW.

# Instruction Types

- **Interrupt instructions**
  - Interrupt processor to service specific condition.
  - INT, INTO, IRET.
- **Processor control instructions**
  - Set and clear status flags, and change the processor execution state.
  - STC, STD, STI.
- **Miscellaneous instructions**
  - NOP, WAIT.

# Komut Hedef (Destination), Kaynak (Source)

- Kaynak, hedef ile manipule edilir sonuç hedeftedir.
- Kaynak, anlık bir değer, genel amaçlı kayıt edici (Register) veya bellek konumu olabilir.
- hedef, genel amaçlı bir kayıt edici (Register) veya bellek konumu olabilir.
- Kaynak ve hedef aynı boyutta olmalıdır.
- Örneğin:
  - Mov ah, bl
  - Mov ax, bx

# Usage Notes

- A lot of slides are adopted from the presentations and documents published on internet by experts who know the subject very well.
- I would like to thank who prepared slides and documents.
- Also, these slides are made publicly available on the web for anyone to use
- If you choose to use them, I ask that you alert me of any mistakes which were made and allow me the option of incorporating such changes (with an acknowledgment) in my set of slides.

Sincerely,

Dr. Cahit Karakuş

**cahitkarakus@esenyurt.edu.tr**