

# Temel C Programlama

Eğitimi

# Neset Aydin

Kıdemli Gömülü Yazılım Mühendisi

Öğrencilik yıllarımmdan beri 8 senenin üzerinde bir süredir yazılım ile uğraşıyorum mezu olduktan sonra, Saha Robotik, Koka Mühendislik, Conrad Mühendislik firmalarında 5 senenin üzerinde piyasa tecrübe kazanma fırsatı buldum. Bare-metal, RTOS ve Linux uygulamaları başta olmak üzere ürün odaklı uygulamalar geliştirdiyorum.



# Conrad

Conrad Mühendislik; 2018 yılından beri hobi dedektörleri, derin arama dedektörleri ve 3D zemin tarama dedektörleri tasarıımı ve üretimi yapmaktadır. Conrad ve Groundtech markaları adı altında tasarımlarını gerçekleştirmektedir.

Ürünler başta Amerika olmak üzere Peru, Çin, Güney Kore, Katar, Filipinler, Malezya, Uruguay, Endonezya, ülkelerine direkt ve bayiiler aracılığıyla ihraç edilmektedir.

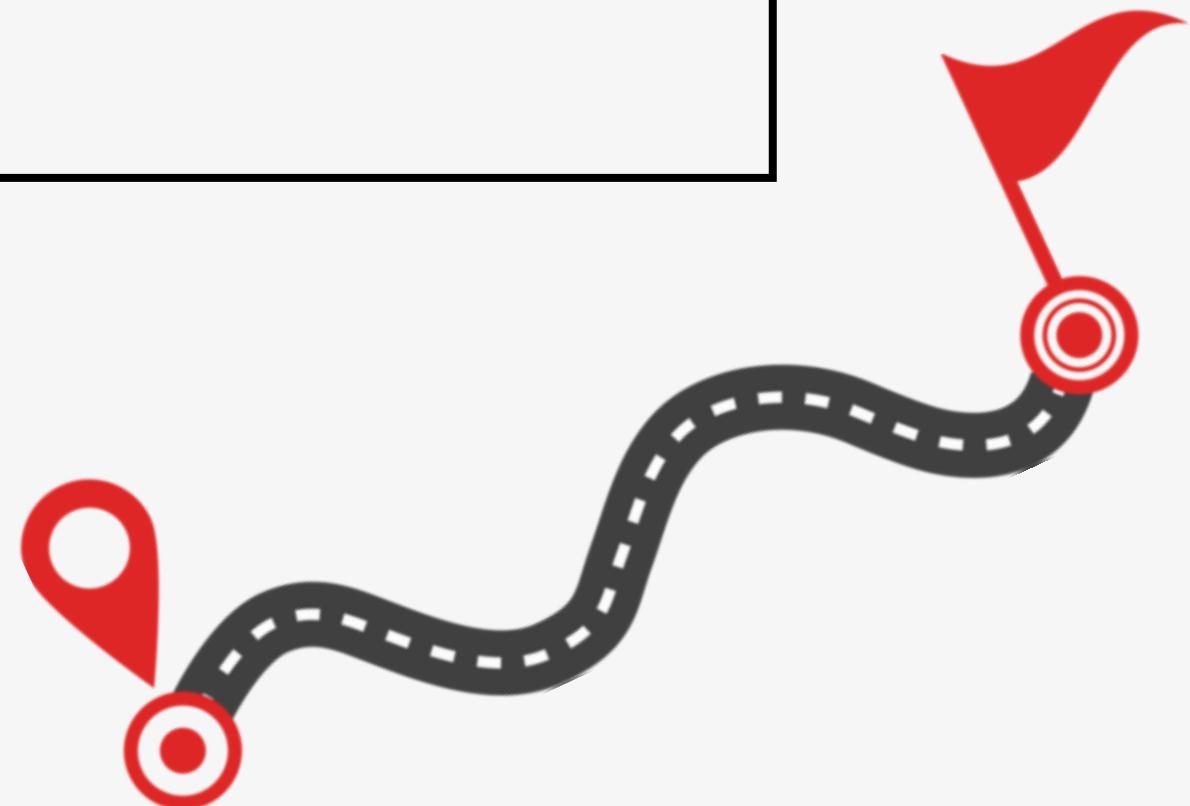
# Eğitim içeriği ve Yol Haritamız

**Temel Sistem Mimarisi, İşlemci Aritmetiği, Sayı Sistemleri ve Kodun Çalıştırılabilir Hale Gelmesi**

**Veri Tipleri, Operatörler, Koşullar, Döngüler, Fonksiyonlar**

**Diziler, İşaretçiler, Sık kullanılan önişlemci komutları, Enum ve Structure**

**Proje Çalışması ve Kapanış**



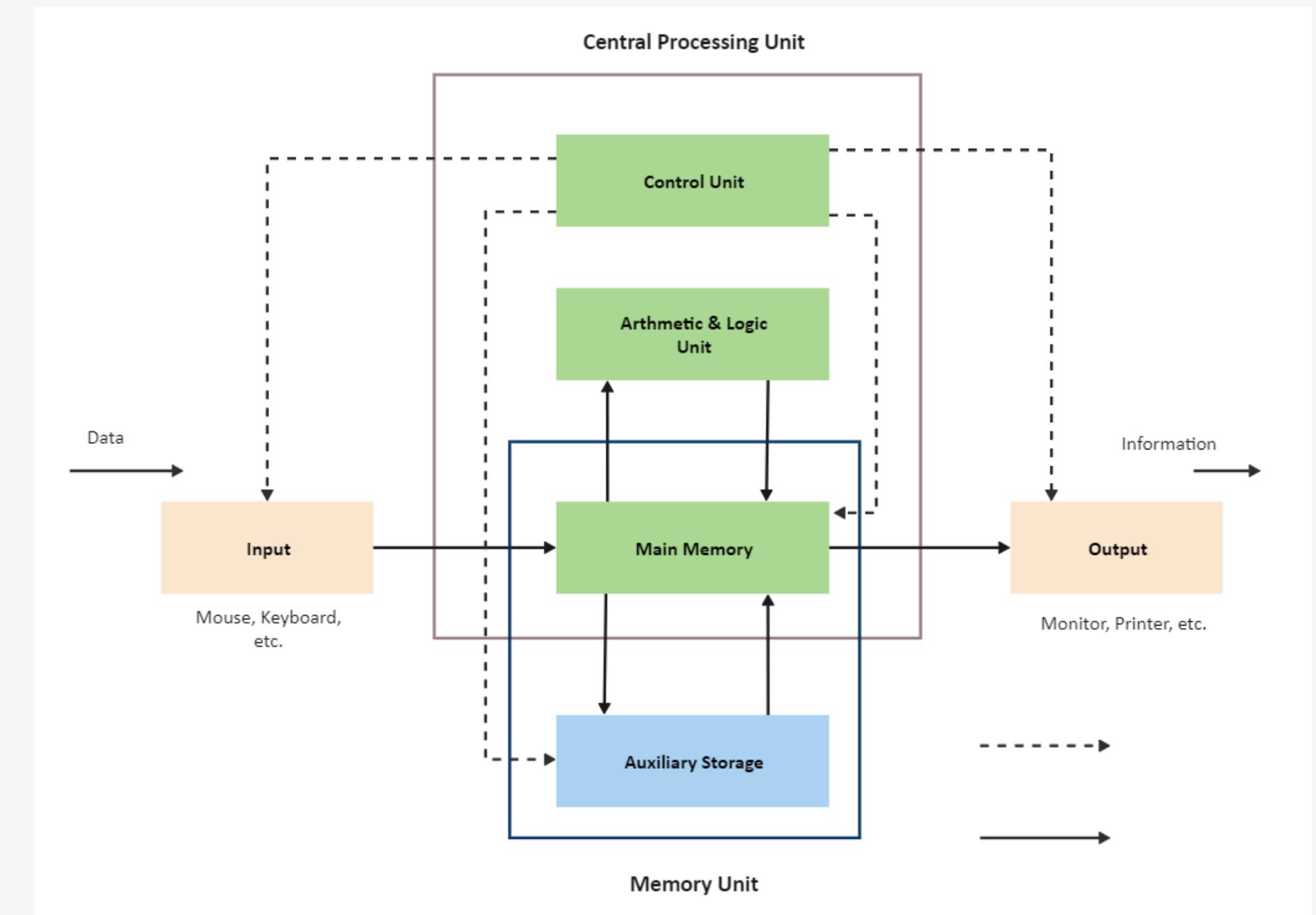
# Temel Sistem Mimarisi

# En temelde ne var?

Bir yazılım'ın çalışabilmesi için en temelde hangi donanım parçacıklarına ihtiyacımız var ?

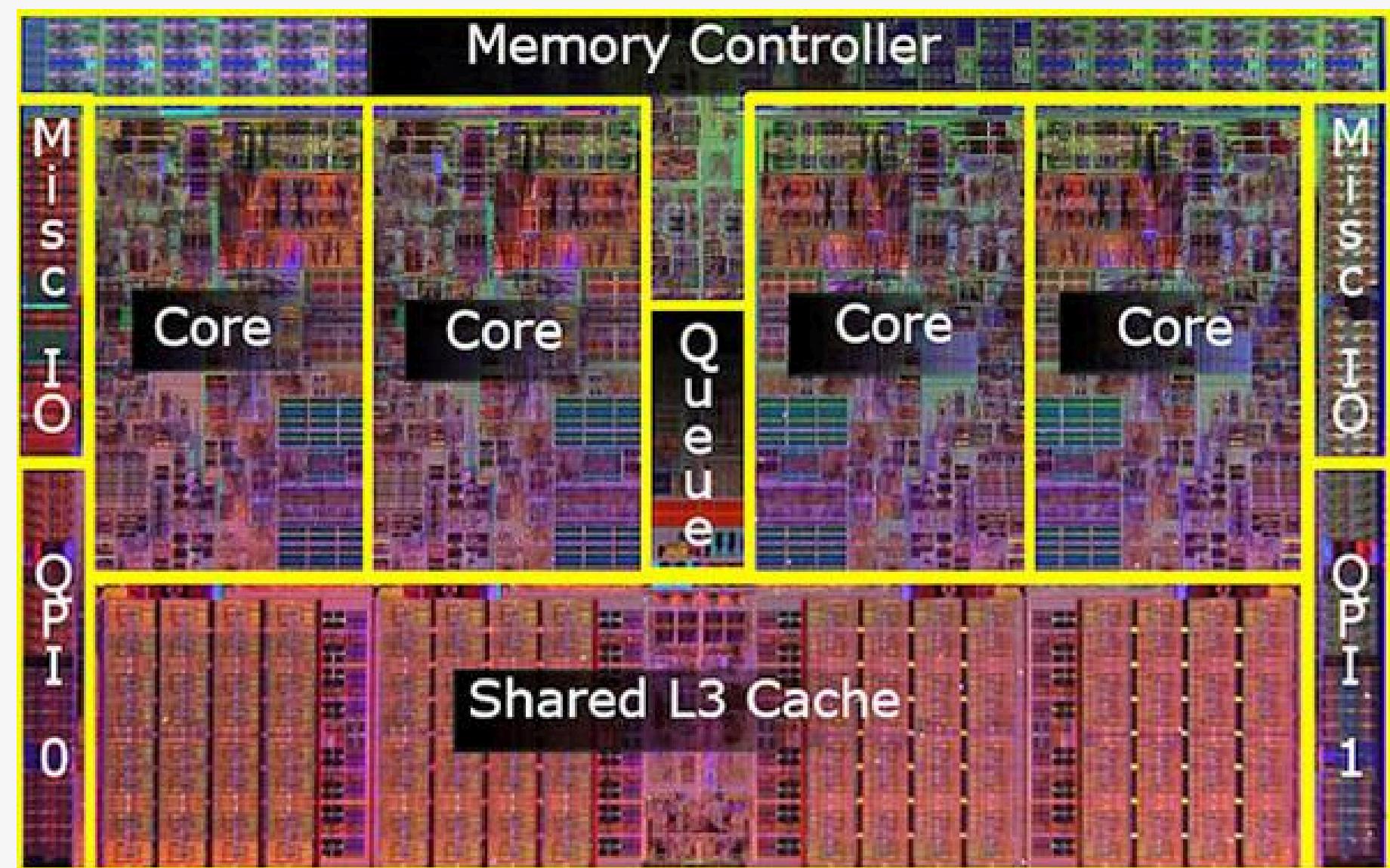
# En temelde ne var?

- CPU,
- Volatile Memory,
- Non-Volatile Memory,
- Bus systems,
- I/O

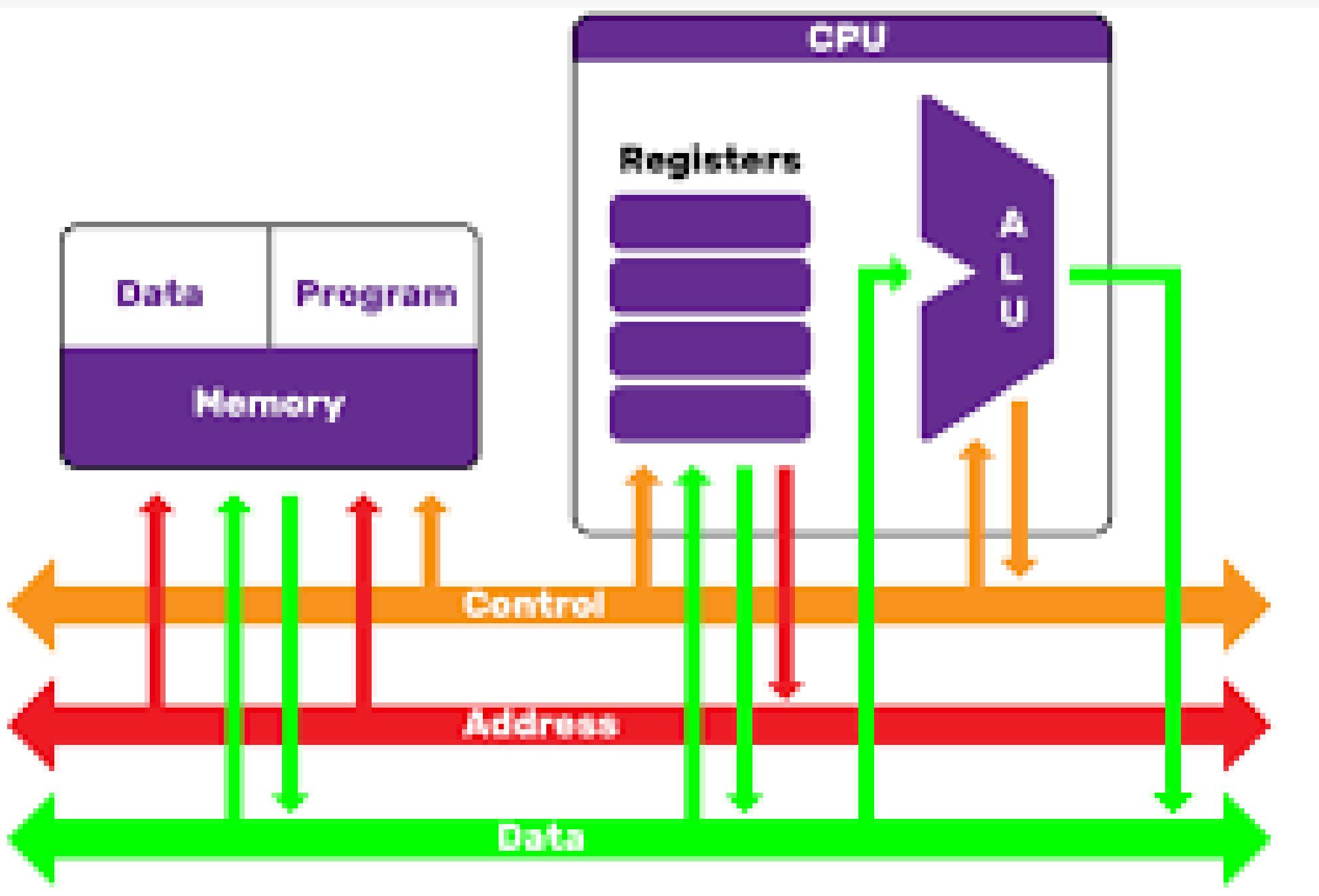


# Unutmadan

Eğer elektrik olmasaydı, bu bileşenlerin hiç bir anlamı olmazdı.



# İşlemci (CPU - Central Processing Unit)



Temelde dört ana birimden oluşur;

- Aritmetik ve Mantık Birimi (ALU - Arithmetic Logic Unit),
- Kontrol Birimi (CU - Control Unit),
- Register'lar (Özel Hızlı Bellekler),
- Önbellek (Cache Memory)

# CPU

## ALU

- Toplama, çıkarma, çarpma, bölme ve mantıksal işlemleri yapar.
- Bit seviyesinde işlemler: Kaydırma, maskeleme, AND, OR işlemleri.
- Örnek: İki sayının toplanması ADD R1, R2, R3 ( $R1 = R2 + R3$ )

## CU

- Program komutlarını sırasıyla yürütür.
- Saat sinyali (Clock) ile işlemleri senkronize eder.
- Komut kodlarını (Opcode) işler ve uygun birimlere yönlendirir.
- Örnek: MOV R1, #5 komutu ile register'a veri yazma işlemi.

# CPU

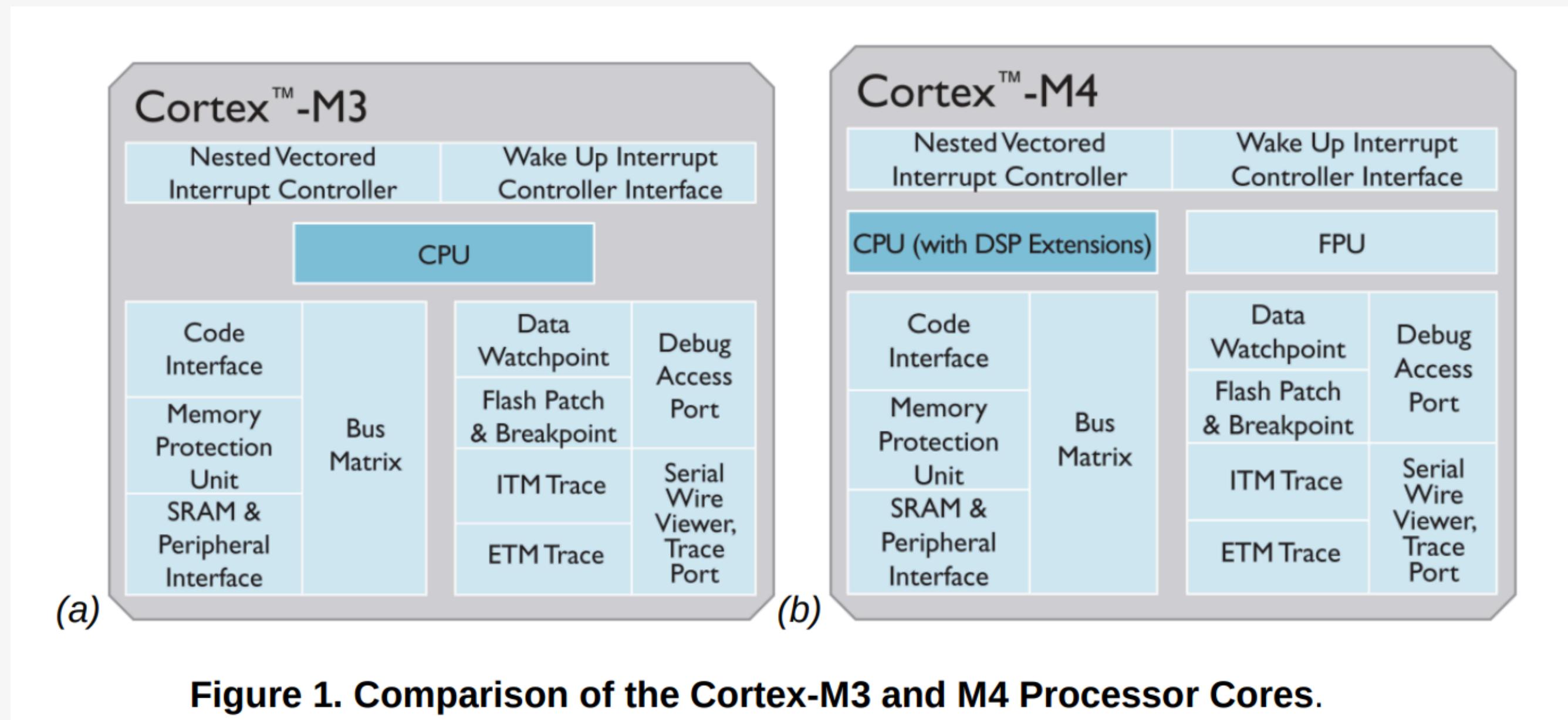
## Register

- Genel Amaçlı Register'lar: Geçici veri depolar (R0, R1, R2...)
- Program Sayacı (PC - Program Counter): Sıradaki komutun adresini tutar.
- Komut Register'i (IR - Instruction Register): Yürüttülecek komutu saklar.
- Stack Pointer (SP): Bellekteki stack bölgesini takip eder.
- Örnek: Bir değişkenin taşınması MOV R2, R1

## Cache Memory

- L1, L2, L3 cache seviyeleri vardır.
- CPU'nun sık kullanılan verileri sakladığı hızlı bellek bölgesi.
- Örnek: İşlemcinin RAM yerine önbellekten veri okuması işlemi.

# CPU



**"Bir GPU değiğildim ama bende çok  
veri işledim."**

– CPU

# Ama nasıl?

Tamam CPU diye bir şey var o olmazsa olmuyor ama bizim yazdığımız kod üzerinde nasıl çalışıyor ?



# CPU ve diğer birimlerin haberleşmesi

## İşlemci ve Bellek

- İşlemci, adres yolunu (Address Bus) kullanarak RAM'e erişir.
- Veri yolu (Data Bus) üzerinden veri okur/yazar.
- Kontrol yolu (Control Bus), veri akışını yönlendirir.
- Örnek: LOAD R1, [1000] RAM'den bir veri okuma işlemi.

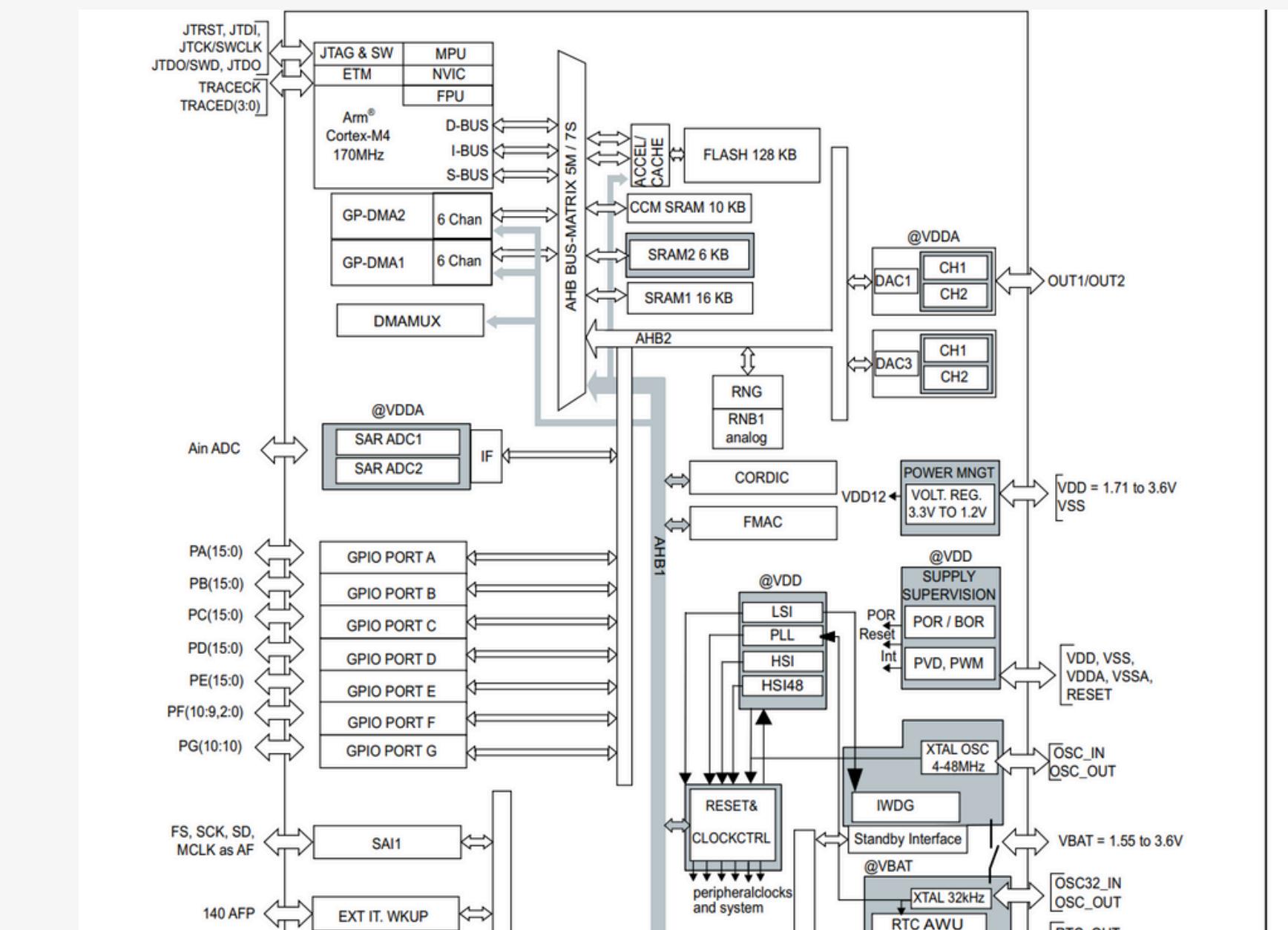
## Çevre Birimleri ile İletişim

- Harici veri yolları (USB, PCI, SATA, I2C, SPI) kullanılarak çevre birimlerine bağlanır.
- Kesme (Interrupt) mekanizması ile çevre birimleri işlemciye sinyal gönderir.
- Örnek: Klavyeden veri okuma işlemi ve işlemcinin tepki vermesi.

# CPU ve diğer birimlerin haberleşmesi

## İşlemci ve Bellek

- I-bus (Instruction Bus – Talimat Yolu)
- D-bus (Data Bus – Veri Yolu)
- S-bus (System Bus – Sistem Yolu)



# Non-Volatile Memory (NVM)

## Özellikler

- Güç kesilse bile veri kaybolmaz
- Uzun süreli depolama sağlar
- RAM'e göre daha yavaş ama kalıcıdır

Examples of Non-Volatile Memory



floppy disk



compact disc



internal hard drive



external hard drive



flash drive

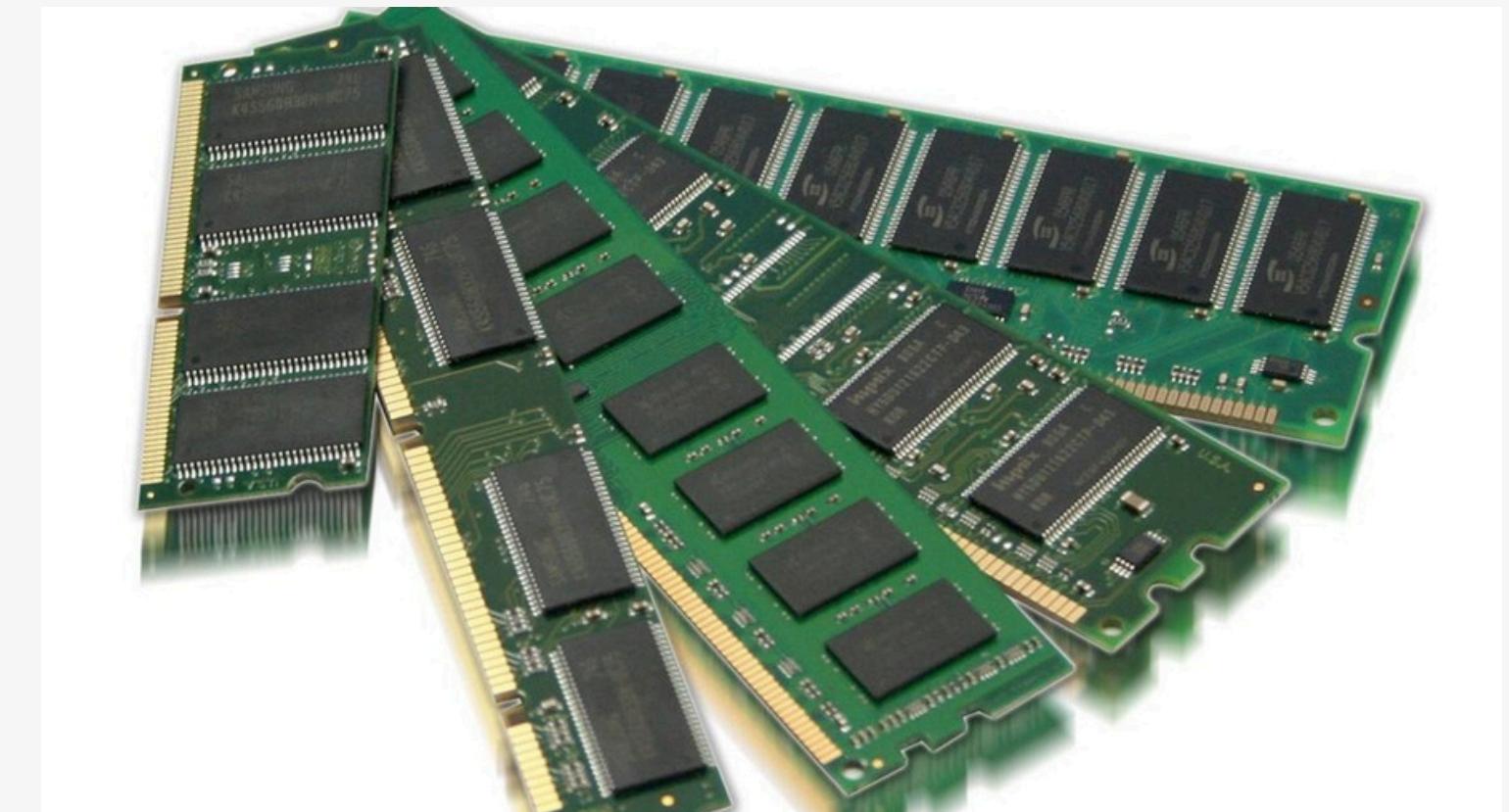


SD card

# Volatile Memory (VM - RAM)

## Özellikler

- Hızlı veri erişimi sağlar.
- Geçici veri depolama için kullanılır.
- Enerji kesildiğinde veriler silinir.



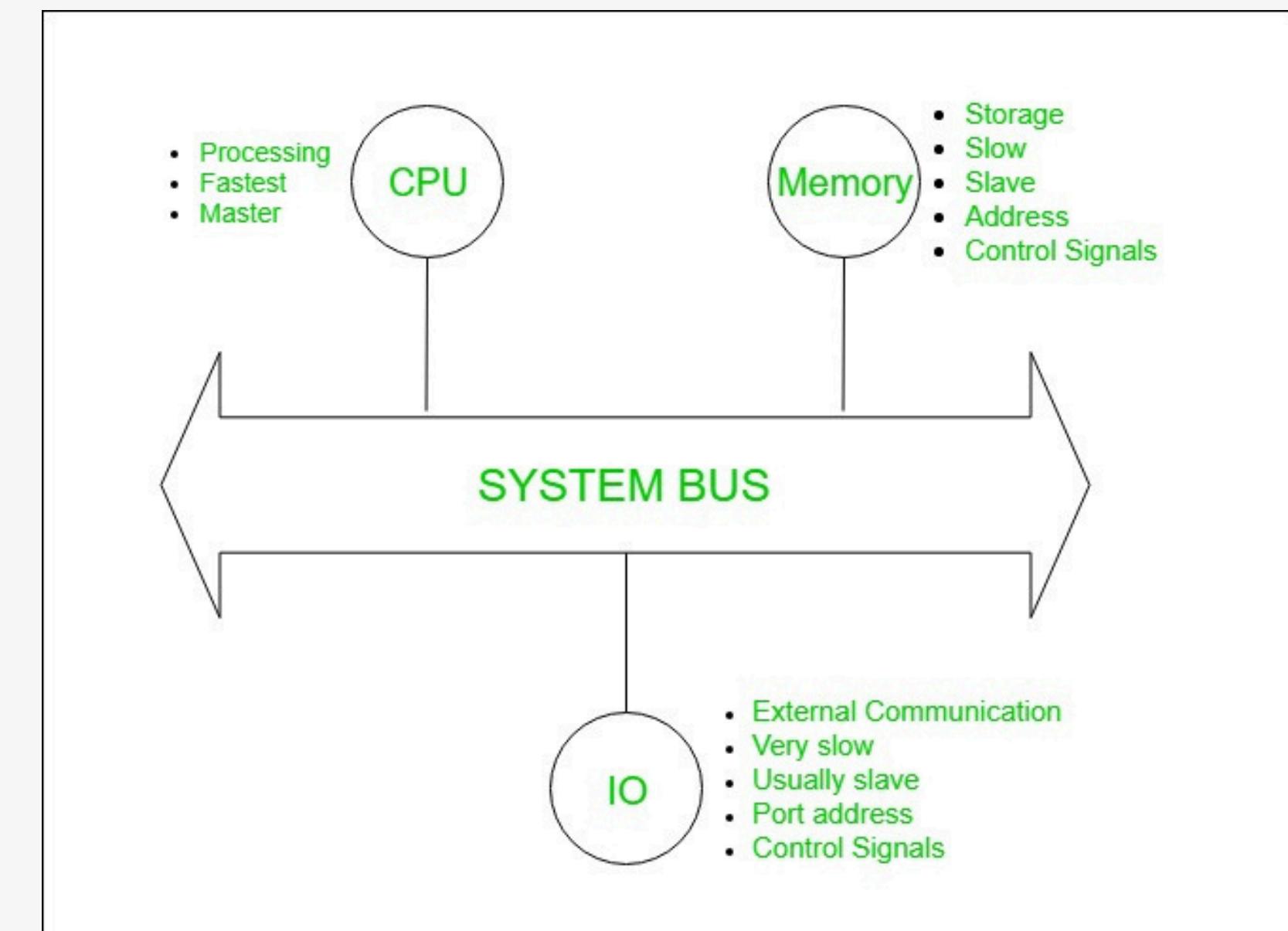
# VM & NVM Farkları

Özellik	Uçucu Bellek (Volatile Memory)	Kalıcı Bellek (Non-Volatile Memory)
Tanım	<b>Elektrik enerjisi kesildiğinde içindeki verileri kaybeden bellek türleridir.</b>	<b>Elektrik enerjisi kesildiğinde bile verileri saklamaya devam eden bellek türleridir.</b>
Veri Saklama Süresi	<b>Geçici; güç kaynağı açık olduğu sürece verileri saklar.</b>	<b>Kalıcı; verileri uzun süreli olarak saklar.</b>
Erişim Hızı	<b>Genellikle daha hızlıdır; işlemciye yakın çalışır ve hızlı veri erişimi sağlar.</b>	<b>Genellikle daha yavaştır; veriye erişim süresi daha uzundur.</b>
Kullanım Alanları	<b>RAM (Rastgele Erişimli Bellek) gibi birincil belleklerde kullanılır; işletim sistemi ve aktif uygulamaların çalışması için gereklidir.</b>	<b>Sabit diskler, SSD'ler, ROM'lar ve flash bellekler gibi ikincil depolama birimlerinde kullanılır; verilerin uzun süreli depolanması için idealdir.</b>
Avantajları	<ul style="list-style-type: none"><li>- Hızlı veri erişimi sağlar.</li><li>- İşlemcinin performansını artırır.</li></ul>	<ul style="list-style-type: none"><li>- Verileri kalıcı olarak saklar.</li><li>- Enerji kesintilerinden etkilenmez; veriler korunur.</li></ul>
Dezavantajları	<ul style="list-style-type: none"><li>- Enerji kesildiğinde veriler kaybolur.</li><li>- Sürekli enerji tüketimi gerektirir.</li></ul>	<ul style="list-style-type: none"><li>- Veri erişim hızı daha düşüktür.</li><li>- Yazma ve silme işlemleri sınırlı sayıda olabilir (ör. flash belleklerde).</li></ul>

# Bus Systems

## Örnekler

- Adres Bus – Bellek adreslerini taşıır
- Veri Bus – Verileri iletir
- Kontrol Bus – İşlemcinin kontrol sinyallerini taşıır
- Seri Bus (I2C, SPI, UART, CAN, USB) – Daha az pin kullanır, düşük maliyetlidir
- Paralel Bus (PCI, DDR, ATA) – Daha hızlı, ancak fazla pin gerektirir



# SAYI SİSTEMLERİ

# Giriş

- Bilgisayarlar veriyi farklı sayı sistemleri ile temsil eder.
- En yaygın kullanılanlar: Binary (2), Octal (8), Decimal (10), Hexadecimal (16)
- Sayı sistemleri, bilgisayar organizasyonu ve veri işleme açısından kritik öneme sahiptir.



# İkili (Binary) Sayı Sistemi

- Binary (2 tabanlı) sistem sadece 0 ve 1 rakamlarını kullanır.
- Her bit, 2'nin kuvvetleriyle temsil edilir.
- Örnek:
- $1101_2 = (1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0) = 13_{10}$
- Bilgisayarlarda veri saklama, işlemci işlemleri ve bellek adreslemede kullanılır.

# Sekizli (Octal) ve Onaltılı (Hexadecimal) Sayı Sistemleri

- Octal (8 tabanlı): 0-7 rakamlarını kullanır.
- Hexadecimal (16 tabanlı): 0-9 ve A-F harflerini içerir.
- Örnekler:
- Octal:  $55_8 = (5 \times 8^1) + (5 \times 8^0) = 45_{10}$
- Hexadecimal:  $7F_{16} = (7 \times 16^1) + (15 \times 16^0) = 127_{10}$
- Hex kodlaması, bellek adresleme ve renk kodlamada sıkça kullanılır.

# Sayı Dönüşümleri

- Decimal → Binary, Octal, Hex dönüşümleri önemlidir.
- İkili sayıların 8'lik ve 16'luk tabana dönüşümü kolaydır:
- Binary → Octal:  $101101_2 \rightarrow 55_8$
- Binary → Hex:  $101101_2 \rightarrow 2D_{16}$
- Çevrim için bölme ve kalan alma yöntemleri kullanılır.

# Bilgisayarlarda Kullanım Alanları ve Örnekler

- Adresleme, veri saklama, hata kontrol algoritmalarında kullanılır.
- Örnek: RGB renk kodlaması
- Hex: #FF5733
- Decimal: (255, 87, 51)
- Bellek adresleme:
- Örn: RAM adresleri genellikle hexadecimal formatta gösterilir.
- İletişim protokollerinde ve şifreleme sistemlerinde büyük rol oynar.

adder

flags

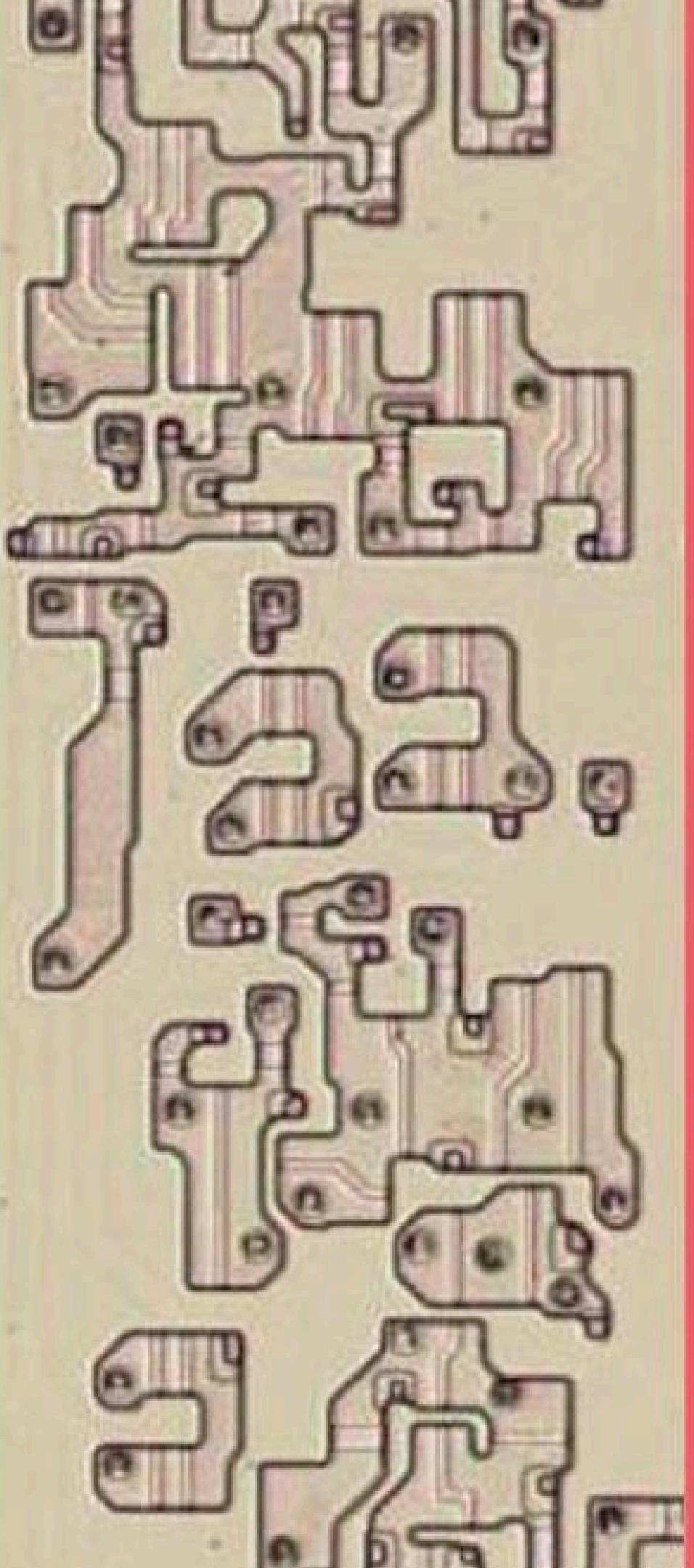
regs

ALU

ROM

15  
İşlemci Aritmatiği

8



# ALU ne yapar ?

**1**

Aritmetik işlemler (Toplama,  
çıkarma, çarpma, bölme)

**2**

**2**

Mantıksal işlemler (AND, OR, XOR,  
NOT)

**3**

Karşılaştırmalar (Büyükür,  
küçükür, eşittir)

**4**

Kaydırma ve döndürme işlemleri

# İşlemci aritmatiği

## Tam sayı aritmetiği

Toplama, çıkarma, çarpma,  
bolme

## Kayan noktalı aritmetik

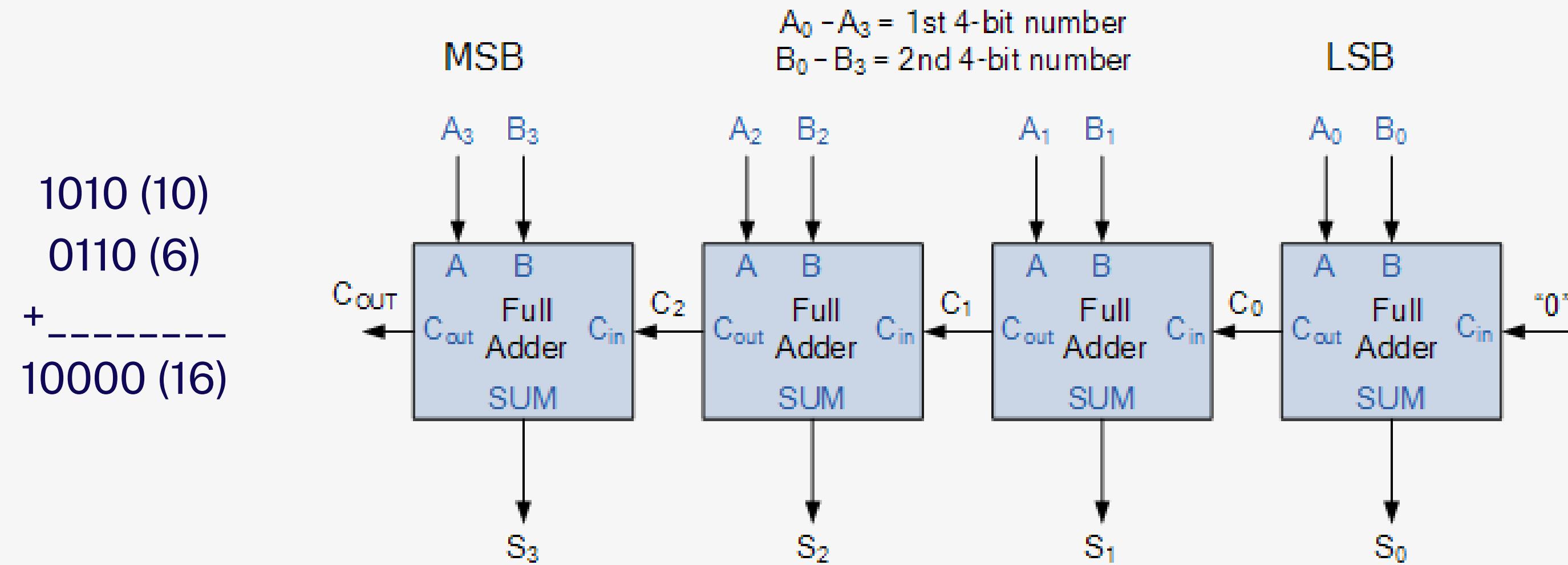
Ondalıklı işlemler (FPU -  
Floating Point Unit  
kullanır)

## İkili sayı sisteminde

**İşlem**  
Sayılar, 0 ve 1 ile temsil  
edilir

# Toplama İşlemi (Binary Addition)

- İşlemci, toplama işlemini "binary half-adder" ve "full-adder" devreleri ile yapar.



# Çıkarma İşlemi (Binary Subtraction)

- İşlemci çıkarma işlemini "2'nin tümleyeni" (Two's Complement) kullanarak toplama işlemine çevirir.

$6 = 0110$ , 1'in tümleyeni: 1001, 2'nin tümleyeni: 1010 (1 ekledik)

$$\begin{array}{r} 1010 \text{ (10)} \\ 1010 \text{ (-6)} \\ + \hline 0100 \end{array}$$

# Temel ikili (Binary) Bölme – Restoring Division

- İşlemci, bölmeyi aslında tekrar eden çıkarmalar ile yapar.

Örnek:  $10 \div 3$  işlemini 4-bit üzerinden inceleyelim.

Bölünen: 10 (1010)	Bölen: 3 (0011)
1. Adım: $1010 - 0011 = 0111$ (Pozitif, sonuçta 1 yazılır)	
2. Adım: $0111 - 0011 = 0100$ (Pozitif, sonuçta 1 yazılır)	
3. Adım: $0100 - 0011 = 0001$ (Pozitif, sonuçta 1 yazılır)	

Sonuç: Bölüm = 3 (0011), Kalan = 1 (0001)

# Booth Algoritması – İşaretli Sayılar İçin Bölme

- Negatif sayılarla bölme işlemi için Booth algoritması kullanılır.
- İşaret bitini (MSB - Most Significant Bit) dikkate alır.
- Negatif sayıları 2'nin tümleyeni yöntemiyle işler.
- Çarpma ve bölme işlemlerinde hız avantajı sağlar.

# Kayan Noktalı (Floating Point) Bölme – Newton-Raphson Yaklaşımı

- Ondalıklı sayılarla bölme için işlemci:
- Böleni ters çevirerek ( $1/b$ ) çarpma yapar.
- Newton-Raphson algoritması ile yaklaşık değeri hesaplar.
- IEEE 754 formatına uygun olarak işlemi tamamlar.

Örnek:  $10.5 \div 3.2$  işlemi için:  
3.2'nin tersi alınır (Yaklaşık 0.3125)  
 $10.5 \times 0.3125$  işlemi yapılır  
Sonuç  $\approx 3.28$  olarak hesaplanır



C Programlama Dili

# Genel bilgiler

- 1972 yılında Dennis Ritchie tarafından geliştirildi.
- Sistem programlama ve genel amaçlı yazılımlar için kullanılır.
- Hızlı, verimli ve taşınabilir bir dildir.
- UNIX işletim sisteminin büyük kısmı C diliyle yazılmıştır.

# C Standartları

- C89/C90 (ANSI C): Amerikan Ulusal Standartlar Enstitüsü (ANSI) tarafından belirlenmiştir.
- C99: Yeni veri türleri ve geliştirilmiş matematiksel fonksiyonlar eklenmiştir.
- C11: Yeni bellek yönetimi fonksiyonları ve çoklu işparalelliği destekleyen yapılar eklenmiştir.
- C18: C11'in hata düzeltmeleri ve iyileştirmelerle güncellenmiş versiyonudur.

# Kod güvenlik standartları

## MISRA-C

- C programlama dili için güvenli bir alt küme tanımlayan bir kodlama standardıdır. İlk olarak otomotiv sektörü için geliştirilmiş olsa da, günümüzde güvenlik ve güvenilirliğin kritik olduğu birçok sektörde kullanılmaktadır.
- C dilinin belirsiz ve hataya açık özelliklerini sınırlayarak, daha güvenli ve güvenilir kod yazımını teşvik eder.
- kodlama kurallarını ve direktiflerini içerir. Her bir kural, zorunlu (mandatory), gereklî (required) veya önerilen (advisory) olarak sınıflandırılır. Bu sınıflandırma, kuralların önem derecesini belirtir.

# Kod güvenlik standartları

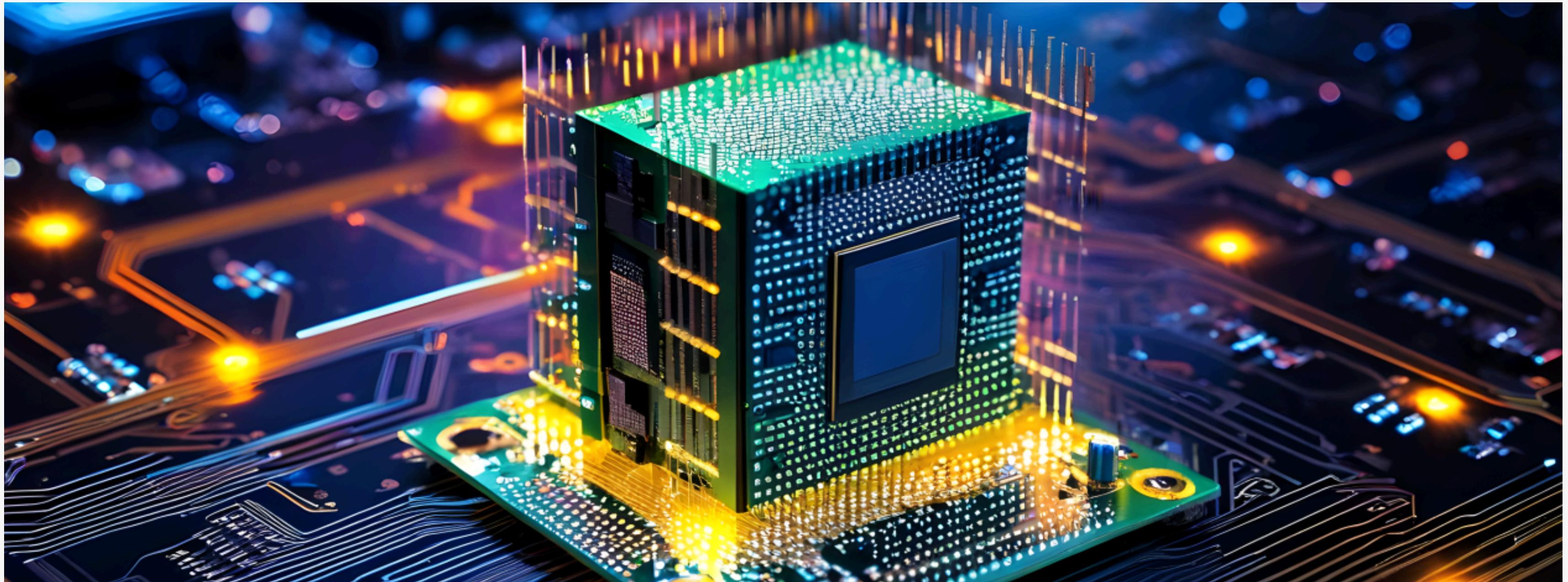
## CERT C

- Carnegie Mellon Üniversitesi'nin Yazılım Mühendisliği Enstitüsü tarafından geliştirilen bir kodlama standardıdır. Bu standart, C dilinde güvenli kod yazımı için en iyi uygulamaları ve kuralları tanımlar.
- Yazılım güvenliğini artırmak ve güvenlik açılarını en aza indirmek için C dilinin güvenli kullanımını teşvik eder.
- Farklı uygulama alanlarına uyarlanabilir ve çeşitli güvenlik tehditlerine karşı koruma sağlar.

# Kod güvenlik standartları

## ISO/IEC TS 17961:2013

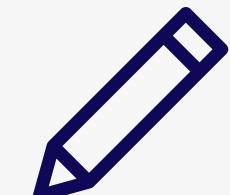
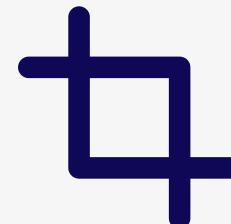
- C dilinde güvenli kodlama için teknik bir spesifikasyondur.
- C dilinin belirsiz davranışlarını ve güvenlik açıklarını tanımlayarak, güvenli kodlama uygulamalarını teşvik eder.
- Diğer güvenlik standartlarıyla uyumludur ve birlikte kullanılabilir.



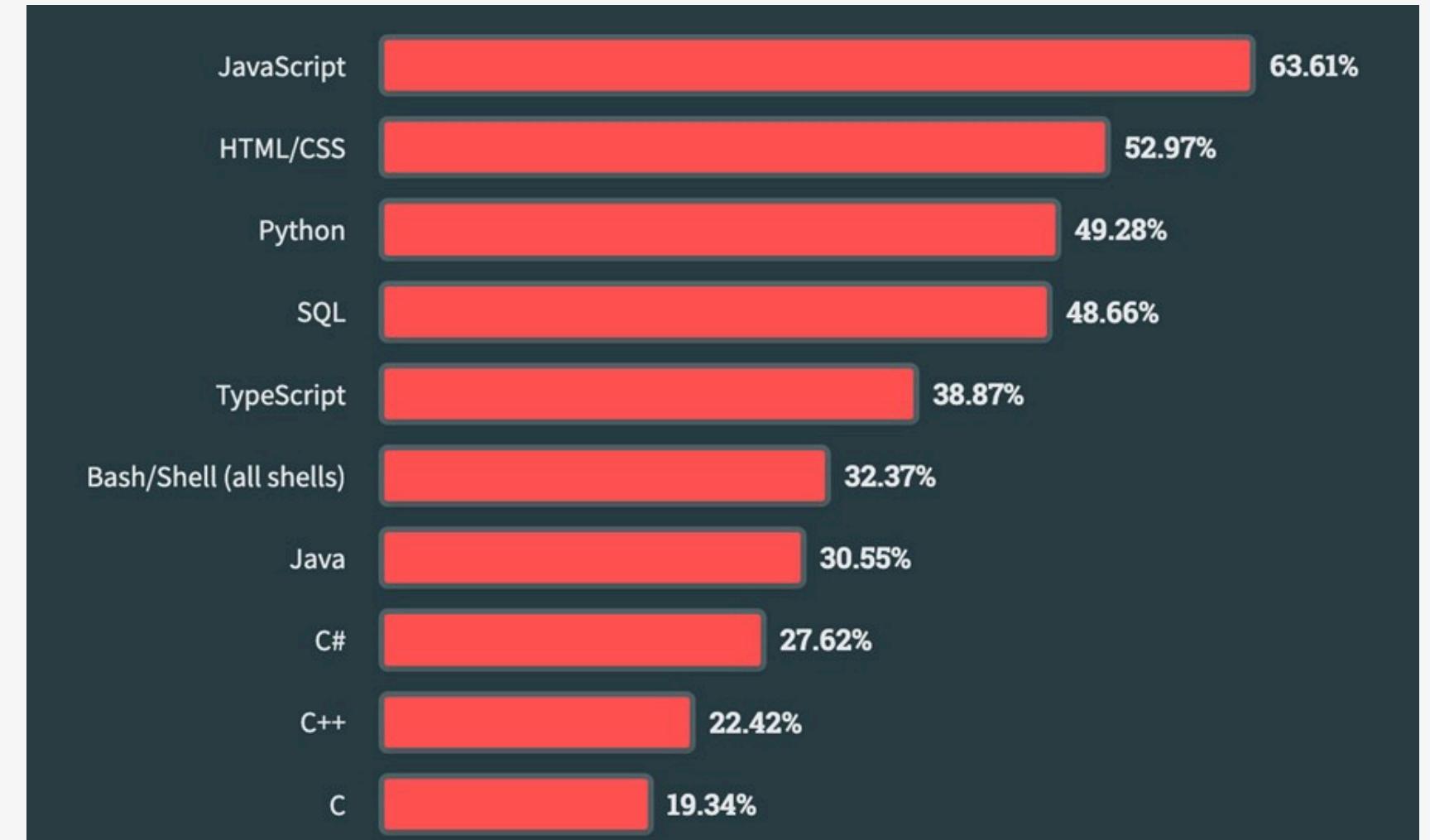
Kullanım Alanları

# Kullanım alanları

- İşletim Sistemleri
- Gömülü Sistemler
- Sistem ve Ağ Programlama
- Oyun Geliştirme
- Yeni Programlama  
Dillerinin Geliştirilmesi
- Masaüstü Uygulamaları



Jul 2024	Jul 2023	Change	Programming Language
1	1		Python
2	3	▲	C++
3	2	▼	C
4	4		Java
5	5		C#
6	6		JavaScript
7	13	▲	Go
8	7	▼	Visual Basic
9	11	▲	Fortran
10	8	▼	SQL



**TIOBE Index**  
**July 2024**

**stackoverflow**  
**2023**

**"Programlama bir bilim  
değildir. Programlama bir  
zanaattır."**

– Richard Stallman

Beni dinlediğiniz için  
teşekkür ederim.