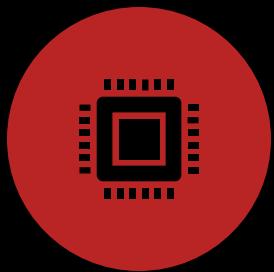


MAKİNE DİLİ VE BROOKSHEAR MİMARİSİ

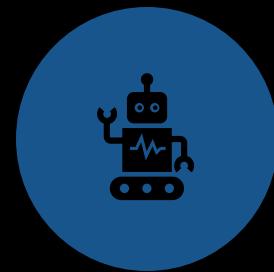
Bursa Teknik
Üniversitesi



İÇİNDEKİLER



TEMEL BİLGİSAYAR
MİMARİSİ (CPU/ ALU/
CU)



MAKİNE DİLİ KAVRAMI,
INSTRUCTION SET VE
KOMUT YAPISI



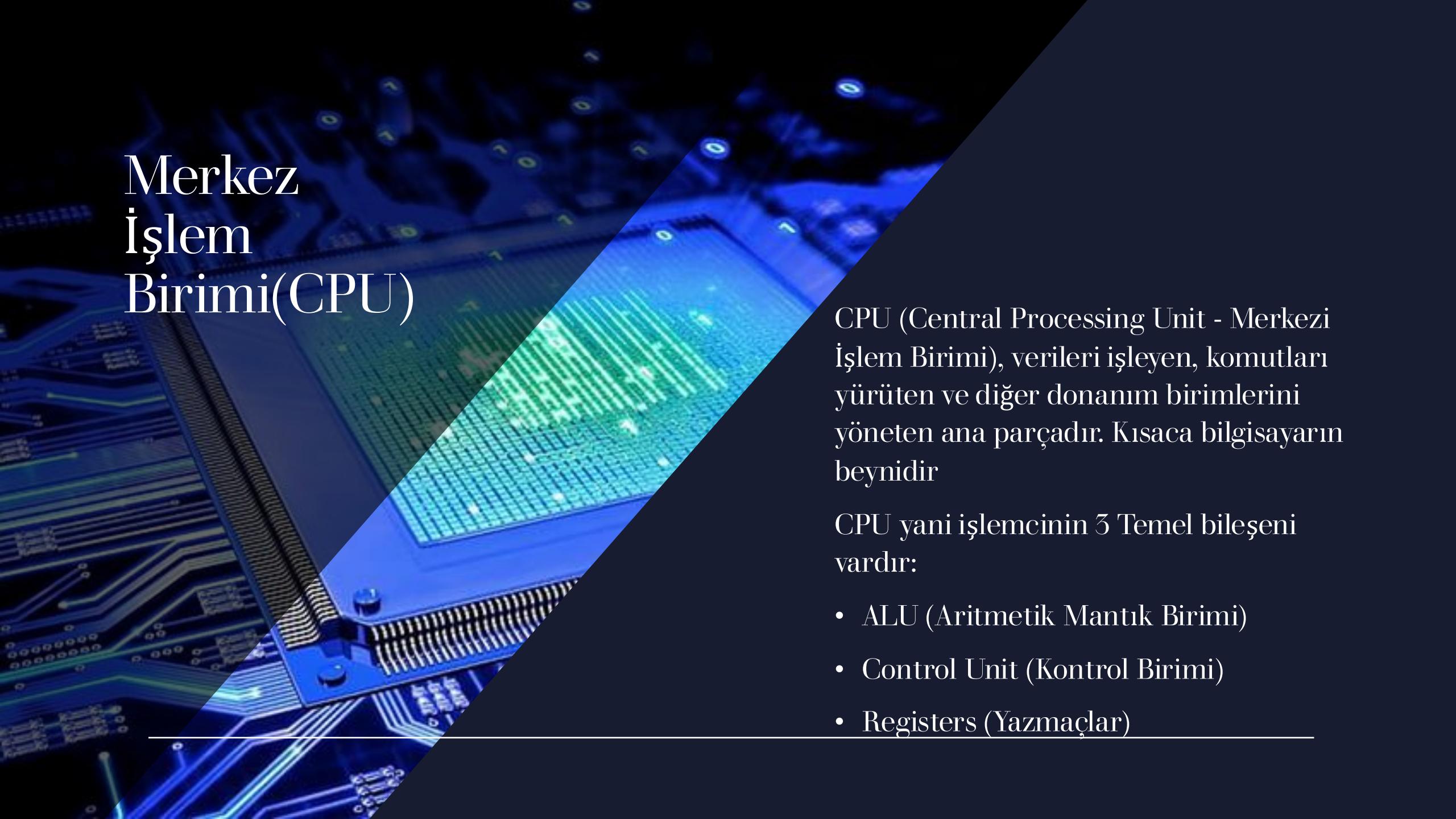
BROOKSHEAR
MAKİNESİ VE
ÖZELLİKLERİ



PROGRAMIN
YÜRÜTÜLMESİ(FETCH-
DECODE-EXECUTE)

BİLGİSAYAR MİMARISI





Merkez İşlem Birimi(CPU)

CPU (Central Processing Unit - Merkezi İşlem Birimi), verileri işleyen, komutları yürütüten ve diğer donanım birimlerini yöneten ana parçadır. Kısaca bilgisayarın beynidir

CPU yani işlemcinin 3 Temel bileşeni vardır:

- ALU (Aritmetik Mantık Birimi)
- Control Unit (Kontrol Birimi)
- Registers (Yazmaçlar)

ALU, işlemcinin hesaplama yapan beynidir ve iki ana kategoride işlem yapar:

1. Aritmetik İşlemler (Matematik Kısmı)

- ALU, sayısal veriler üzerinde dört temel matematik işlem yapar.
- **Yaptığı İşlemler:** Toplama, Çıkarma, Çarpma, Bölme.

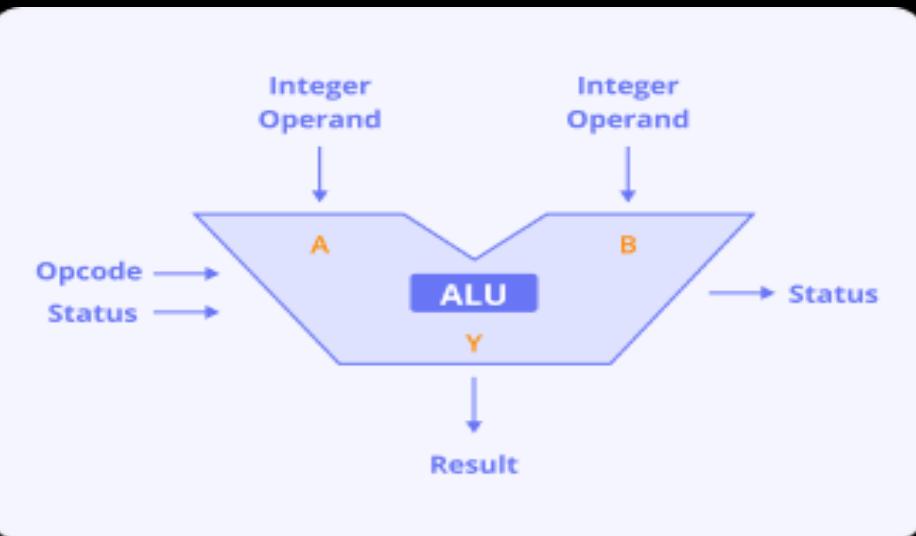
2. Mantıksal İşlemler (Karar Kısmı)

- ALU, "Evet/Hayır" veya "Doğru/Yanlış" gibi mantık yürütme işlemlerini yapar. Bilgisayarın "karar vermesi" aslında 1 ve 0'ları kıyaslamasıdır.
- **Yaptığı İşlemler:** VE (AND), VEYA (OR), ÖZEL VEYA (XOR), DEĞİL (NOT) ve Karşılaştırma (Büyükür, Küçüktür, Eşittir).



ALU (Aritmetik Mantık Birimi)

ALU NASIL ÇALIŞIR?



ALU'nun çalışma mantığı bir "V" harfine benzer (Genellikle şemalarda da V şeklinde çizilir):

Giriş (Input): İşlem yapılacak iki sayı (örneğin Register 2 ve Register 3'teki değerler) ALU'nun üstünden girer.

Komut (Opcode): "Ne yapayım?" emri (Topla, Çıkar vs.) yan taraftan girer.

Çıkış (Output): İşlem sonucu (örneğin toplam değer) alt taraftan çıkar ve bir Register'a yazılır.

Durum Bayrakları (Flags): Ayrıca ALU, işlemin sonucuna dair küçük notlar tutar. Örn: "Sonuç sıfır çıktı", "Sonuç negatif çıktı" gibi.

Control Birimi (Control Unit)

Control Birimi, işlemci içindeki tüm faaliyetleri koordine eden ve komutların doğru sırayla yürütülmesini sağlayan birimdir.

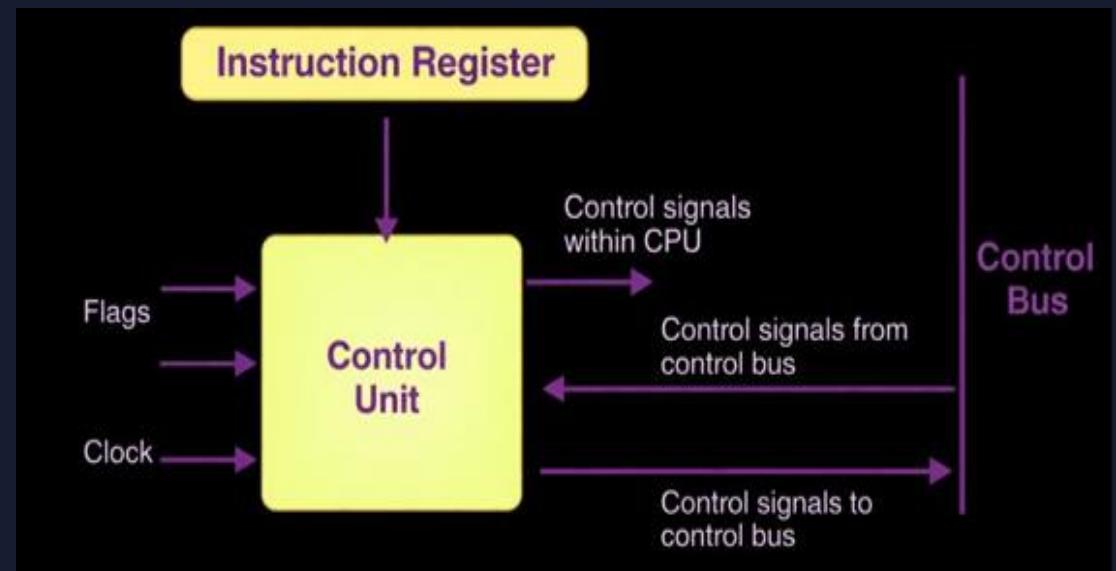
Temel Görevleri:

Getirme ve Çözme (Fetch & Decode): Bellekten gelen 0 ve 1'lerden oluşan komutu alır . Bu komutun hangi anlama geldiğini kendi içindeki devreler sayesinde çözer.

Yönlendirme (Routing): Verilerin işlemci içinde hangi yollardan geçeceğini belirler. Bu yollara Bus denir

Yönetme ve Sinyal Gönderme (Execute Control): Çözülen komuta göre işlemcinin diğer parçalarına elektrik sinyalleri gönderir.

Zamanlama (Timing): Tüm bu işlemlerin birbirine karışmadan, Saat (Clock) sinyaliyle uyumlu bir şekilde sırayla yapılmasını sağlar..

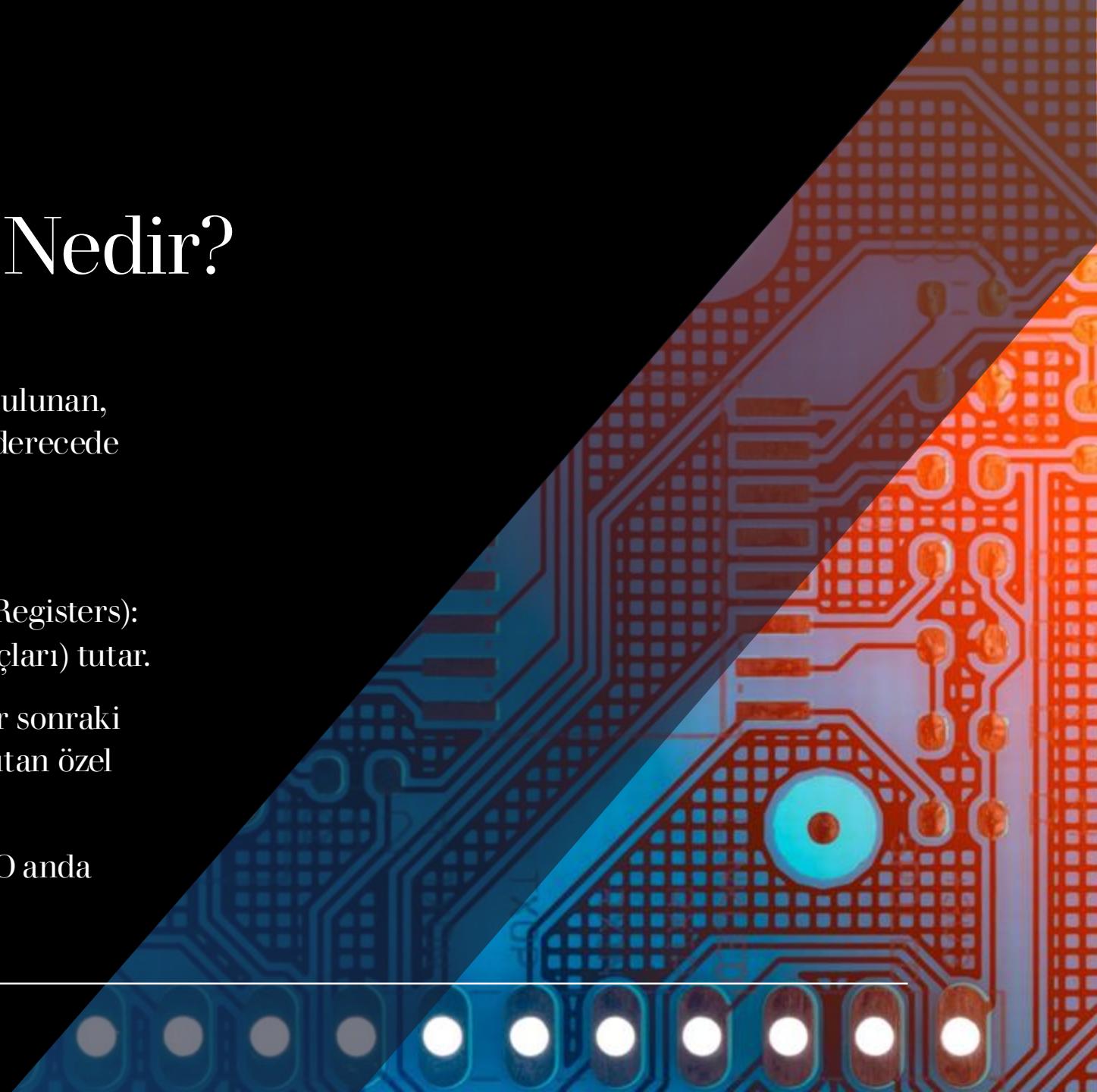


Register (Yazmaç) Nedir?

Register, doğrudan CPU'nun (İşlemci) içinde bulunan, kapasiteleri çok küçük ama hızları muazzam derecede yüksek olan geçici hafıza birimleridir.

Önemli Register Türleri:

- Genel Amaçlı Yazmaçlar (General Purpose Registers): İşlem sırasında geçici verileri (sayıları, sonuçları) tutar.
- Program Sayacı (Program Counter - PC): Bir sonraki çalıştırılacak komutun bellekteki adresini tutan özel kaydedicidir.
- Komut Yazmacı (Instruction Register - IR): O anda işlenmekte olan güncel komutu tutar.



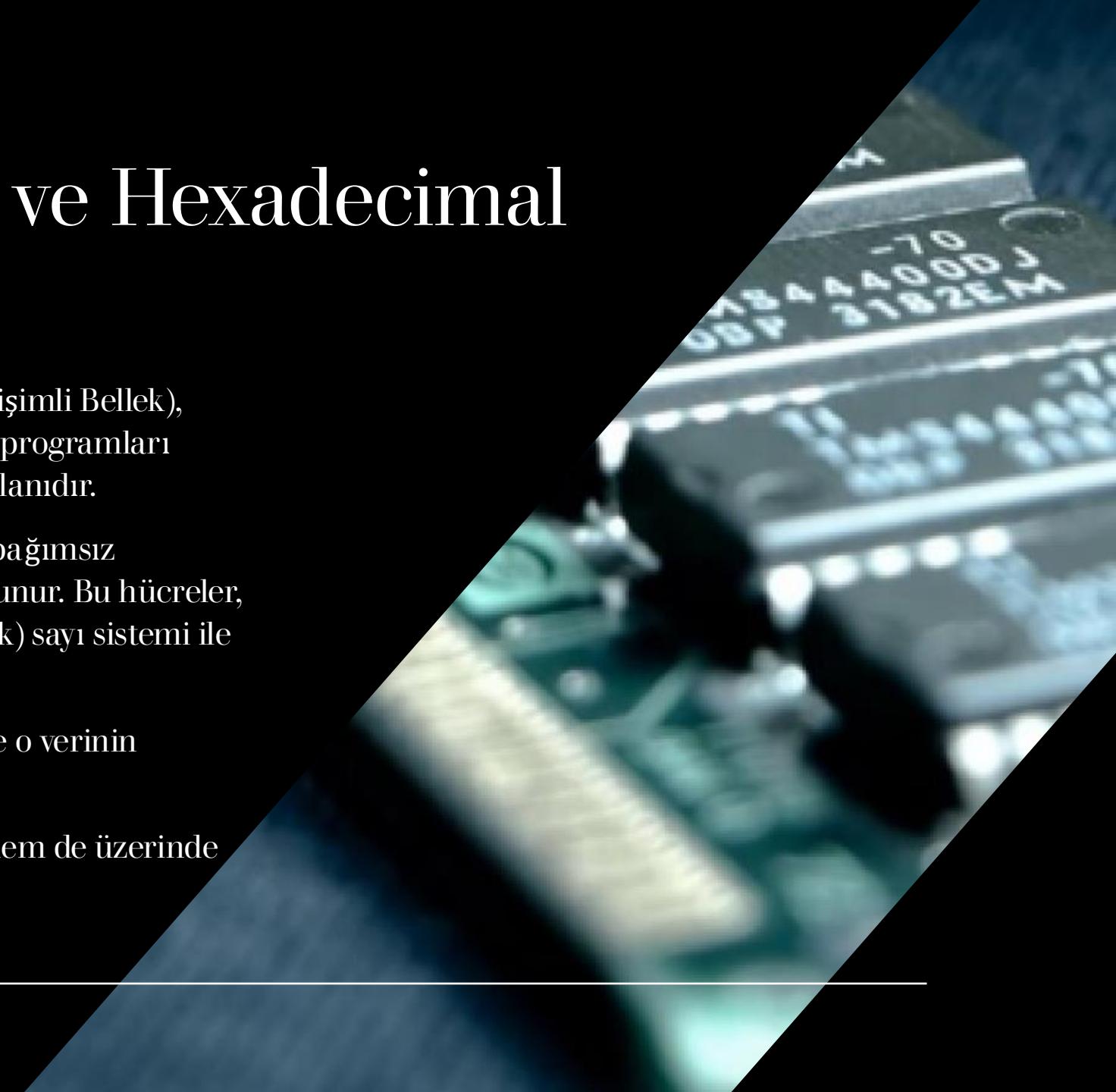
Ana Bellek (RAM) ve Hexadecimal Adresleme

RAM (Random Access Memory - Rastgele Erişimli Bellek), bilgisayarın o an üzerinde çalıştığı verileri ve programları geçici olarak sakladığı, çok hızlı bir çalışma alanıdır.

RAM, her biri 8 bit (1 Byte) veri saklayabilen bağımsız hücrelerden oluşur. Toplamda 256 hücre bulunur. Bu hücreler, karmaşıklığı önlemek için Hexadecimal (16'lık) sayı sistemi ile 00'dan FF'ye kadar adreslenir.

İşlemci, bir veriye ulaşmak istediği sadece o verinin bulunduğu hücrenin adresini kullanır.

Aynı bellek alanı hem program komutlarını hem de üzerinde işlem yapılacak sayısal değerleri saklayabilir.



MAKING DILIGENCE (MACHINE LANGUAGE)

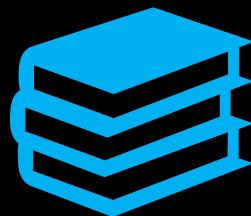
Makine Dili Nedir?

Makine dili, 0'lar ve 1'lerden (Binary sistem) oluşan, elektriksel sinyallere karşılık gelen sayısal komutlar dizisidir. İki önemli özelliği vardır:

- **Makine İçindir:** Bilgisayarın donanımının doğrudan anlayıp çalıştırabildiği tek dil makine dilidir. Günün sonunda yazdığımız her kod makine diline çevrilir.
- **Mimariye Özgündür:** Her işlemci ailesinin (Intel x86, ARM veya Brookshear Makinesi) kendine ait bir makine dili vardır. Birinin diliyle yazılan program, diğerinde çalışmaz.



Komut (Instruction) Nedir?

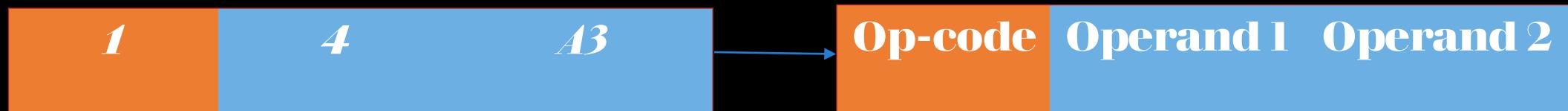


Komut (Instruction): İşlemciye "şunu yap" diyen en küçük emir cümlesiidir. Bir bilgisayar programı aslında milyarlarca "Komut"un arkaya arkaya dizilmesinden ibarettir.

Instruction Set (Komut Seti): İşlemcinin "kelime dağarcığı"dır. İşlemci sadece bu listedeki komutları (Topla, Kopyala, Dur, Zıpla vb.) bilir.

BİR KOMUTUN ANATOMİSİ (OP-CODE VE OPERAND)

- **Op-code (İşlem Kodu):** Komutun ilk karakteridir. İşlemciye “topla” , “yükle” veya “dallan” gibi ne yapması gerektiğini söyleyen ana emirdir.
- **Operand (İşlenen):** Komutun kalan 3 karakteridir. Bu kısım, işlemin kiminle veya nerede yapılacağını söyler.



BROOKSHEAR MAKİNESİ

Brookshear Makinesi, fiziksel olarak teknoloji mağazalarından satın alabileceğiniz bir bilgisayar değildir. J. Glenn Brookshear tarafından, bilgisayar bilimleri öğrencilerine von Neumann mimarisinin ve makine dilinin temellerini karmaşıklığa boğulmadan öğretmek için tasarlanmış sanal (teorik) bir bilgisayar modelidir.



BROOKSHEAR MAKİNESİ NEDİR?

Bu makinenin kapasitesi, günümüz bilgisayarlarına göre çok küçüktür ama çalışma mantığı birebir aynıdır. İçinde şu birimler bulunur:

Bellek (RAM): Toplam 256 hücreden oluşur. Her hücrenin bir adresi vardır (Hexadecimal 00'dan FF'ye kadar). Her hücre 8 bit (1 byte) veri saklar.

Kaydediciler (Registers): İşlemcinin içinde 16 adet genel amaçlı kaydedici bulunur. İsimleri, Register 0'dan Register F'ye kadardır.

Komut Uzunluğu: Her komut 16 bit (2 byte) uzunluğundadır. Bu, tek bir komutun bellekte 2 hücre yer kapladığı anlamına gelir.

BROOKSHEAR MAKİNESİ TEKNİK ÖZELLİKLERİ

Brookshear Makine Dili - 12 Temel Opcode Tablosu

Opcode (Hex)	Format	İşlem Adı	Açıklama
Veri Transfer İşlemleri			
1	1RXY	LOAD (Bellekten)	XY hafıza adresindeki içeriği, R register'ına yükler.
2	2RXY	LOAD (Değer)	XY sayısal değerini doğrudan R register'ına yükler.
3	3RXY	STORE	R register'daki içeriği, XY hafıza adresine kaydeder.
4	40RS	MOVE	S register'daki içeriği, R register'a kopyalar.
Aritmetik ve Mantıksal İşlemler (ALU)			
5	5RST	ADD (Tam Sayı)	S ve T register'larını toplar (Tam sayı), sonucu R'ye yazar.
6	6RST	ADD (Float)	S ve T register'larını toplar (Kayan noktalı), sonucu R'ye yazar.
7	7RST	OR	S VEYA T mantıksal işlemi. Sonuç R'ye.
8	8RST	AND	S VE T mantıksal işlemi. Sonuç R'ye.
9	9RST	XOR	S ÖZEL VEYA T mantıksal işlemi. Sonuç R'ye.
A	AROX	ROTATE	R içeriğini X bit sağa döndürür.
Kontrol Birimi İşlemleri			
B	BRXY	JUMP (Koşullu)	Eğer R register içeriği 0 register'ına eşitse, XY adresine atla.
C	C000	HALT	Programı durdur.

Not: R, S, T: Register Numarası (0-F), X, Y: Adres veya Veri.

Veri Transferi Komutları

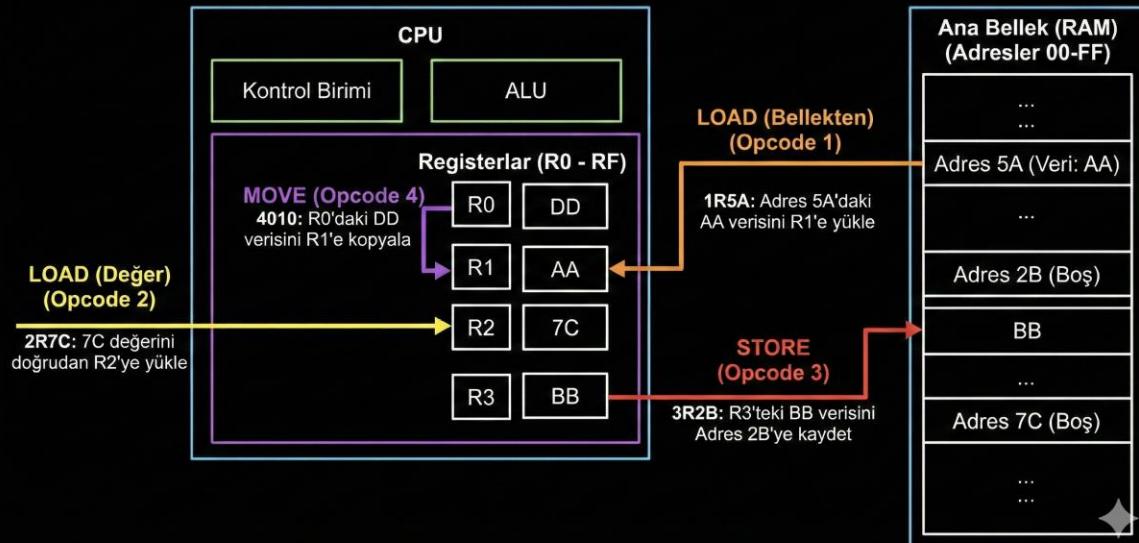
Op-code 1, LOAD (Bellekten Yükle): Bu komut, RAM'deki belirli bir adrese gider, orada ne varsa kopyalar ve CPU içindeki bir Register'a getirir.

Op-code 2, LOAD (Doğrudan Değer Yükle): Bu komut, bellekteki bir adrese bakmaz. Komutun içinde yazan sayının kendisini (Bit desenini) doğrudan Register'a yükler.

Op-code 3, STORE (Belleğe Kaydet): İşlemcide (Register) duran bir veriyi alıp RAM'e yazmak için kullanılır. Genellikle bir hesaplama bittikten sonra sonucu saklamak için yapılır.

Op-code 4, MOVE (Registerlar Arası Taşıma): Belleğe hiç uğramadan, işlemcinin kendi içindeki iki register arasında veri kopyalamayı sağlar. Çok hızlıdır.

Brookshear Makine Dili - Veri Transfer Komutları Şeması



Aritmetik İşlem Komutları

Op-code 5, Tam Sayı Toplama (ADD Integer): Bu komut, kaydedicilerdeki verileri Tam Sayı (Integer) olarak kabul eder ve toplar. Bilgisayar bilimlerinde bu genellikle "Two's Complement" (İki'ye Tümleyen) yöntemiyle saklanan sayılar için kullanılır.

Op-code 6, Kayan Noktalı Toplama (ADD Float): Bu komut, kaydedicilerdeki verileri Kayan Noktalı (Floating Point) sayı olarak kabul eder. Yani ondalıklı sayıları (Örn: $2.5 + 3.75$) toplamak için bu komut kullanılır.

İkinin Tümleyeni Yöntemi

İkinin tümleyeni yöntemi, negatif sayıların ikilik (binary) sistemde gösterilme yöntemidir.

- (1) Pozitif bir sayıyı ikilik düzende yazın.

Örnek sayı: $+5_{10} = 00000101_2$

(1)	0	0 0 0 0 0 0 1	1_2	Bitleri ters çevirin
-----	---	---------------	-------	----------------------

- (2) Tüm bitleri ters çevirin.

(2)	1	1 1 1 1 0 1 0	2	$+1$ ekleyin
-----	---	---------------	-----	--------------

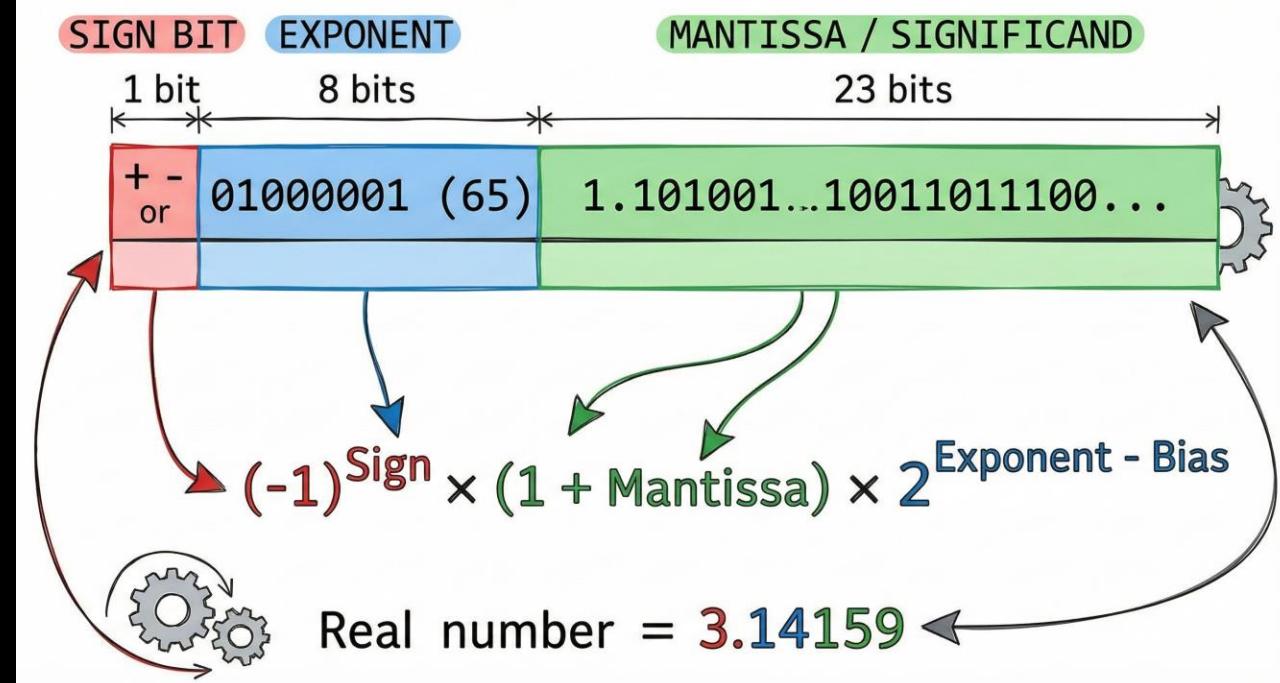
$$\begin{array}{r} \frac{11111010}{+ 1} \\ \hline 11111011_2 \end{array}$$

Negatif ikilik gösterim

- (4) Sonuç, negatif ikilik gösterimdir.

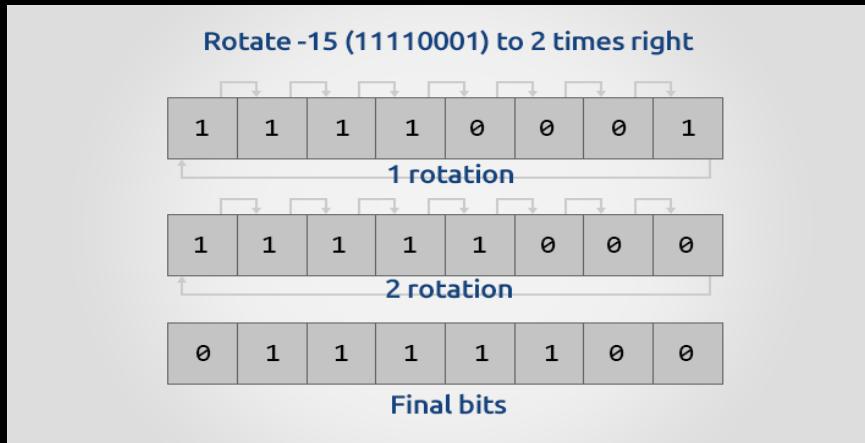
(4)	-5	1 1 1 1 0 1 1	2	Negatif ikilik gösterim
-----	----	---------------	-----	-------------------------

* En baştaki "1" sayının negatif olduğunu gösterir.



Mantıksal İşlem Komutları

AND		OR		XOR							
INPUT		OUTPUT		INPUT		OUTPUT		INPUT		OUTPUT	
A	B			A	B			A	B		
0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	1	1	1	0	1	1
0	1	0	0	0	1	1	0	0	1	0	1
1	1	1	1	1	1	1	1	1	1	1	1



Op-code 7, OR (VEYA) İşlemi: İki register'daki bitleri karşılaştırır. Bitlerden en az biri 1 ise, sonuç 1 olur.

Op-code 8, AND (VE) İşlemi: İki register'daki bitleri karşılaştırır. Aynı konumdaki bitlerden biri 0 ise sonuç 0 olur. AND işlemi ile "Maskeleme" yapılır. Bir verinin sadece belirli bir kısmını görmek, geri kalanını silmek (0 yapmak) için kullanılır.

Op-code 9, XOR (ÖZEL VEYA) İşlemi: İki register'daki bitleri karşılaştırır. Bitler birbirinden farklısa sonucu 1 yapar, aynıysa 0 yapar. Basit veri şifreleme algoritmalarında veya bir yazmacın içeriğini tamamen sıfırlamak için kullanılır.

Op-code A, ROTATE (Sağa Döndürme): Bu işlem iki register arasında değil, tek bir register üzerinde yapılır. Bitleri sağa doğru kaydırır.

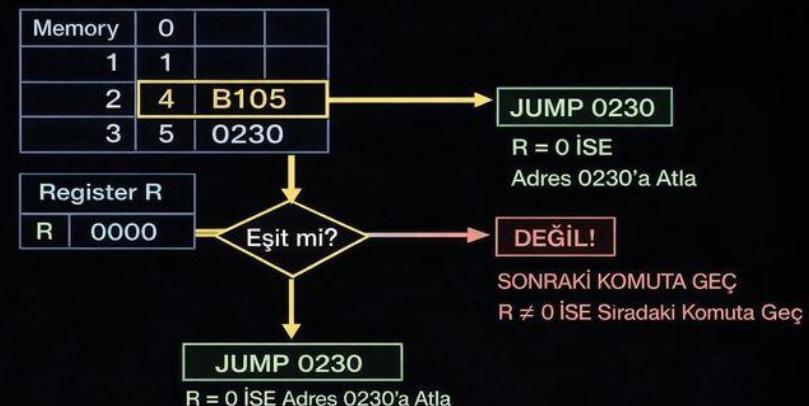
Kontrol Komutları

Op-code B, Koşullu Dallanma (Conditional JUMP):
Bu komut, programın satır satır (sırayla) çalışma akışını bozar ve belirli bir şart sağlanıyorsa programı başka bir adrese gönderir. Yazılım dillerindeki **if** (eğer) ve döngülerin (**while**, **for**) atasıdır.

Op-code C, Durdur (HALT): Makine döngüsünü sonlandırır. İşlemci çalışmayı durdurur.

Op-code B: Koşullu Dallanma (Conditional JUMP)

Format: **BRXY** R = Register içeriği 0 register'ına eşitse, XY adresine atla.



PROGRAM YÜRÜTME (PROGRAM EXECUTION)



Program Yürütme (Program Execution)

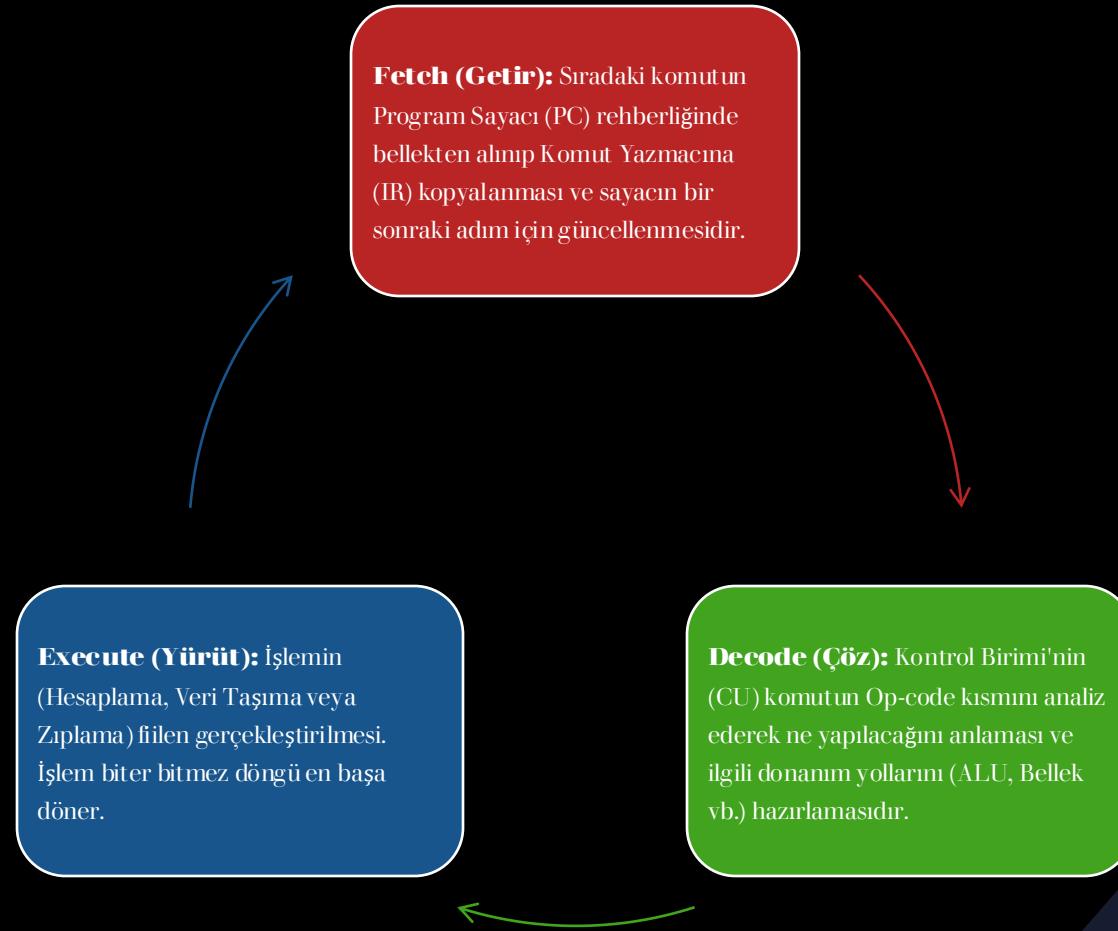
En temel tanımıyla program yürütme, bir işlemcinin kendisine verilen komutları sırasıyla okuması, anaması ve bu komutların gerektirdiği işlemleri fiziksel olarak gerçekleştirmesi sürecidir.

Program yürütülürken CPU'nun içinde şu iki özel yazmaç (register) başrol oynar:

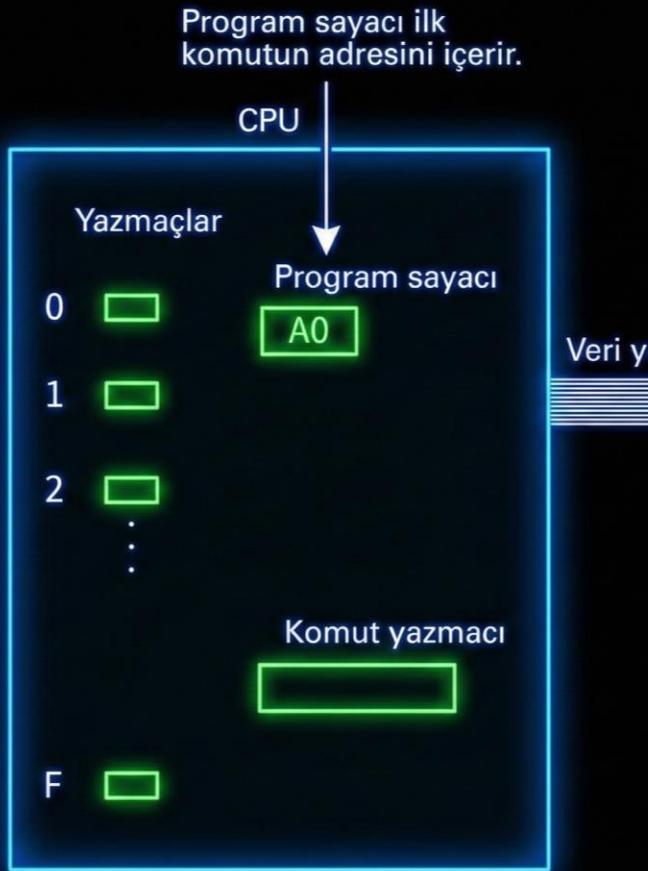
- **Program Sayacı (Program Counter - PC):** "Sırada hangi komut var?" sorusunun cevabını tutar. İşlenecek bir sonraki komutun hafıza adresini gösterir.
- **Komut Yazmacı (Instruction Register - IR):** O an işlemci tarafından yapılmakta olan komutun kopyasını tutar.



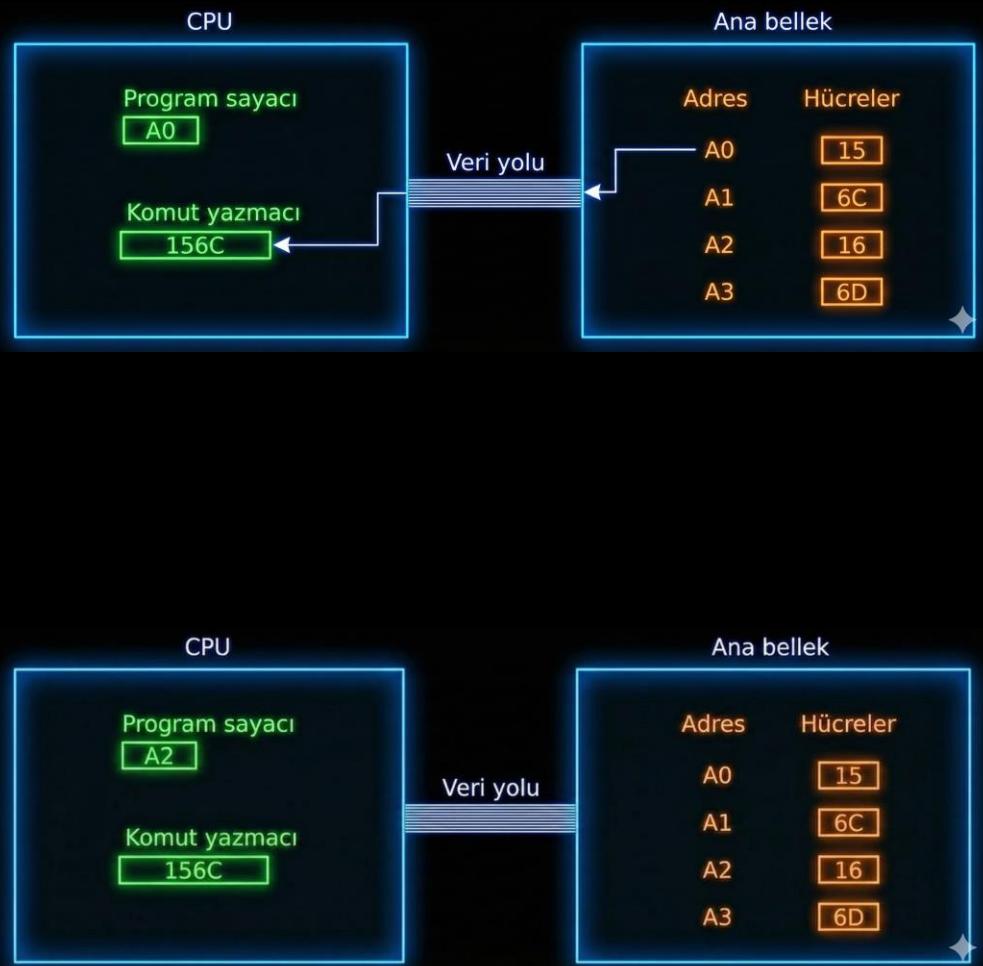
Makine Döngüsü (Machine Cycle): Bilgisayar açık olduğu sürece işlemcinin hiç durmadan tekrarladığı döngüdür. 3 adımı takip eder.



Makine Döngüsü (Machine Cycle)



Program belleğin A0 adresinden başlayarak kaydedilmiştir.



KAYNAKÇA ve TEŞEKKÜRLER

- Görseller: Google Gemini, ChatGPT ve Bing.
- Computer Science An Overview- J. Glenn Brookshear.

Hazırlayan: Hüseyin Efe Dere

Öğrenci Numarası: 25360859048
