

# Bilgisayar Mühendisliğine Giriş Dönem Projesi

AD-SOYAD: Mem Civan Elma

NUMARA: 24360859063

PROJE 4



# VERİ İŞLEME

- Bilgisayar Mimarisi (CPU,ALU,CONTROL UNIT)
  - Makine Dili (Op-Code,Operand)
  - Komut Yapısı
  - Brookshear Makinesi
  - Programın yürütülmesi
-

# BİLGİSAYAR MİMARİSİ

- Bilgisayar mimarisi, bir bilgisayar sisteminin donanımsal yapısını ve bu bileşenlerin birbirleriyle nasıl etkileştiğini inceler.
- Mimari, yazılım ile donanım arasındaki arayüzü belirler.

# Bilgisayar Sisteminin Temel Bileşenleri

- Merkezi İşlem Birimi (CPU)
- Bellek (RAM)
- Giriş / Çıkış Birimleri
- Veri Yolları (Bus)

# CPU (Central Processing Unit) Nedir

- CPU, bilgisayarın komutları yorumlayan ve çalıştırılan ana birimidir.
- Tüm hesaplama ve kontrol işlemleri burada yapılır.
- Bilgisayarın “beyni” olarak tanımlanır.

# CPU'nun Temel Alt Bileşenleri

CPU üç ana parçadan oluşur:

- ALU (Aritmetik Mantık Birimi)
- Control Unit (Kontrol Birimi)
- Registerlar (Kaydediciler)

# ALU (Aritmetik Mantık Birimi)

Aşağıdaki aritmetik işlemlerini yapar:

- Toplama
- Çıkarma

Aşağıdaki mantıksal işlemlerini yapar:

- AND
- OR
- NOT
- XOR

# ALU Örnek İşlemleri

- $7 + 5 \rightarrow 12$
- $1 \text{ AND } 0 \rightarrow 0$
- $1 \text{ OR } 0 \rightarrow 1$

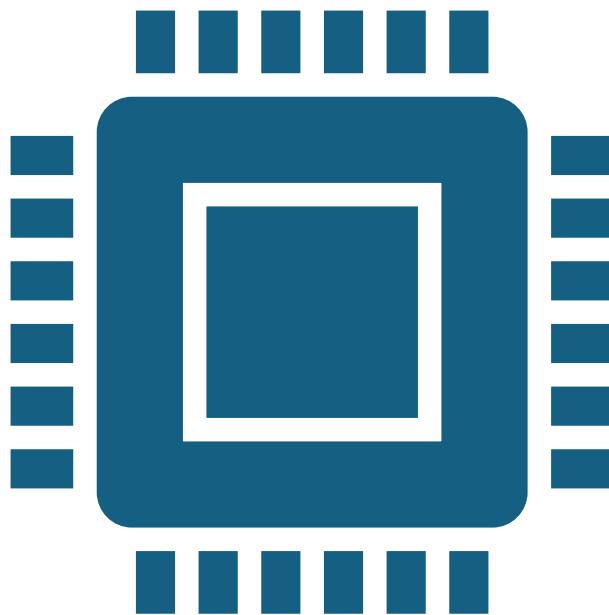
# Control Unit (Kontrol Birimi)

- CPU içindeki yönetici birimdir.
- Komutların:
  - bellekten alınmasını
  - çözülmesini
  - yürütülmesini koordine eder.



# Control Unit'in Görevleri

- Program Counter'ı yönetir.
  - Komut sırasını belirler.
  - Veri akışını kontrol eder.
  - Makine döngüsünü başlatır.
-



## Register (Kaydedici) Nedir

- CPU içinde bulunan çok hızlı bellek birimleridir.
- Boyutları küçüktür.
- Geçici veri tutarlar.

# Register–ALU–Control Unit İlişkisi

- Control Unit komutu çözer.
- Gerekli veriler registerlardan alınır.
- ALU işlemi yapar.
- Sonuç tekrar register'a yazılır.



## Bilgisayar Mimarisi Neden Önemlidir?

- Performansı belirler
- Yazılımın sınırlarını çizer..
- Makine dilinin yapısını etkiler.

# Makine Dili Nedir

- Makine dili, bilgisayarın doğrudan anlayabildiği tek dildir.
- Sadece ikili (binary) verilerden oluşur.
- 0 ve 1'ler donanım seviyesinde elektrik sinyallerini temsil eder.



# Bilgisayar Komutları Nasıl Anlar?

- Komutlar bellekte binary olarak tutulur.
  - CPU bu kodları okur.
  - Kodlar donanım devrelerine karşılık gelir.
-



# Instruction Set Nedir?

- Instruction CPU'nun çalıştırabileceği en küçük işlem birimidir.
  - Bir işlemcinin desteklediği tüm komutlara da Instruction denir.
-

# Makine Dili ve Yazılım İlişkisi

- Yüksek seviyeli diller makine diline çevrilir.
- Derleyici / yorumlayıcı çeviri yapar.
- CPU sadece makine dilini çalıştırır.

# KOMUT YAPISI (OP-CODE & OPERAND)

- Makine dilindeki her komut belirli bir formata sahiptir.
- Komutlar CPU'nun ne yapacağını açıkça belirtir.
- Temel olarak iki ana parçadan oluşur:
  - Op-code
  - Operand

# Op-code (Operation Code)

- Yapılacak işlemi belirtir
- CPU'ya hangi işlemin yapılacağını söyler
- Örnek işlemler:
  - ADD
  - SUB
  - LOAD
  - STORE



# Operand

- İşlem üzerinde çalışılacak veriyi belirtir
  - Şunlar
  - Bellek adresi
  - Register
  - Sabit değer
-



# BROOKSHEAR MAKİNESİ

- Brookshear Makinesi Nedir
  - Eğitim amaçlı tasarlanmış sanal bir makinedir
  - Gerçek CPU'ların çalışma mantığını öğretir
  - Basitleştirilmiş bir instruction set kullanır
-

# Brookshear Makinesinin Yapısı

- Bellek
- Registerlar
- Instruction Set
- Fetch–Decode–Execute mantığı  
gerçek bilgisayarın küçük modeli



# BAŞLICA KOMUTLAR

LOAD Komutu:

- Bellekten veri alır
- Veriyi Accumulator'a yükler

STORE Komutu :

- Accumulator'daki veriyi belleğe yazar

MOVE ve ADD Komutları:

- MOVE , Veriyi bir registerden diğerine taşıır
- ADD, Bellekteki veya registerdaki değeri ACC'ye ekler



## Düger Komutlar

- SUB  
Çıkarma işlemi yapar
- AND  
Mantıksal AND işlemi
- OR  
Mantıksal OR işlemi
- JUMP  
Program akışını değiştirir
- JUMPZ  
Sonuç sıfırsa atla
- JUMPN  
Sonuç negatifse atla
- LOADIM  
Sabit değeri ACC'ye yükler
- HALT  
Programı durdurur

# PROGRAMIN YÜRÜTÜLMESİ

(FETCH – DECODE – EXECUTE)

- Bir program, bellekteki komutların sırayla çalıştırılmasıdır
- CPU, komutları rastgele değil belirli bir döngüyle işler
- Bu döngüye makine döngüsü (machine cycle) denir

# Fetch (Komutun Alınması)

- Program Counter (PC), sıradaki komutun adresini tutar
- Komut bellekten alınır
- Instruction Register (IR)'a yüklenir
- PC bir sonraki komutu gösterecek şekilde artırılır

# Decode & Execute

Decode:

- Control Unit, komutu çözer
- Op-code ve operand ayrıştırılır

Execute:

- Gerekli işlem ALU'da yapılır
- Sonuç registera veya belleğe yazılır
- Gerekirse program akışı değiştirilir

*Dinlediğiniz için  
teşekkürler.*

Bu sunumda bir programın,  
bellekten alınan bir komuttan  
başlayarak CPU tarafından nasıl  
yürütüldüğü gösterilmiştir.