

VERİ MANİPÜLASYONU VE MANTIK KAPILARI

BİLGİSAYAR MÜHENDİSLİĞİNE GİRİŞ

PROF. DR.TURGAY TUGAY BİLGİN

HAZIRLAYAN: MERYEM BAYRAKTAR

ÖĞRENCİ NUMARASI: 25360859087

İÇİNDEKİLER

- Bilgisayar Mimarisi Nedir?
- Temel Donanım Bileşenleri Nelerdir?
- CPU (Merkezi İşlem Birimi)
- ALU (Aritmetik Mantık Birimi)
- Veri Manipülasyonu
- Mantık Kapıları
- Doğruluk Tabloları
- Örnek Devreler

Bilgisayar Mimarisi

- Bilgisayar mimarisi donanım bileşenlerinin bir araya gelme şekli ve bu bileşenlerin yazılımla olan iletişimidir.
- Kısaca bilgisayar mimarisini donanım ve yazılım arasındaki köprü olarak tanımlayabiliriz.
- Bilgisayar mimarisi ayrıca verinin nasıl işlediğini de açıklar.
- Performans ve verimlilikle de doğrudan ilişkilidir.

Temel Donanım Bileşenleri

■ Merkezi İşlem Birimi(CPU)

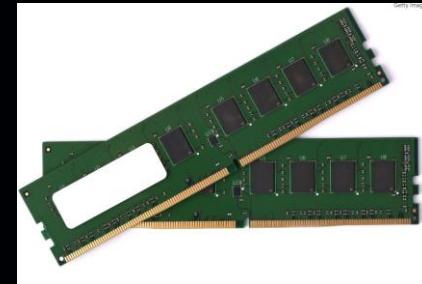
- Bilgisayarın tüm hesaplamalarını yapan ana işlemci



CPU

■ Ana Bellek(RAM)

- Çalışan uygulamaların verilerinin hızlıca erişilmesi için tutulduğu yer.



RAM

■ Kalıcı Bellek(ROM, Disk)

- Sistemin açılış bilgilerini(BIOS) saklayan , silinmeyen bellek.



ROM

■ Giriş/Çıkış Birimleri

- Kullanıcının bilgisayara komut ve veri göndermesini sağlayan araçlar.



Giriş-Çıkış

CPU(Merkezi İşlem Birimi) Nedir?

- CPU'yu aslında bilgisayarın beyni olarak adlandırabiliriz. Bilgisayarda yapılan işlemlerin büyük bir kısmı CPU tarafından yönetilir ve işlenir.
- CPU programlardan gelen komutlar alır , yorumlar ve uygular.
- Bilgisayardaki her hareket(bir tuşa basmak , video izlemek vs.) CPU'nun istediği 0 ve 1'lerden oluşur.
- Donanım ve yazılım arasındaki temel iletişimini sağlar.

CPU'nun Temel Görevleri

- CPU bir işlemi gerçekleştirmek için bu adımları sürekli tekrarlar.
 1. Komutları alma (Fetch)
 2. Komutları çözme (Decode)
 3. Komutları yürütme (Execute)

1) Komutları Alma(Fetch):

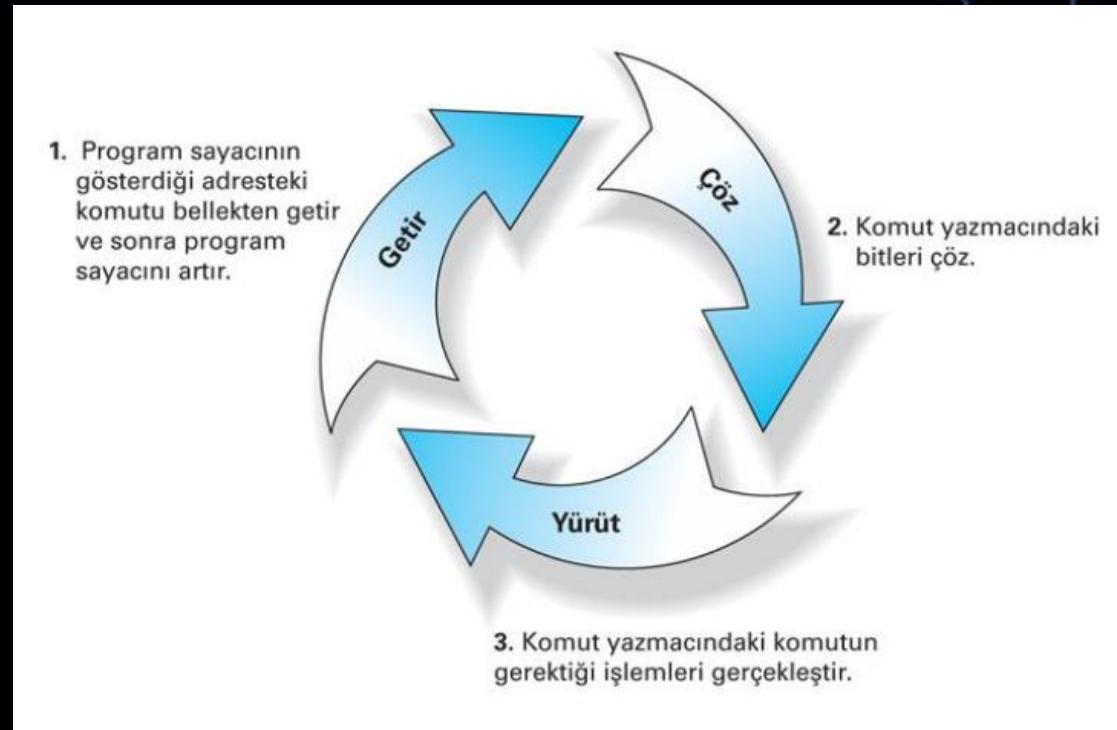
- Bellekte (RAM) bulunan komutları alır.

2) Komutları Çözme(Decode):

- Alınan komutun ne anlama geldiğini ve hangi işlemin yapılacağını

3) Komutları Yürütme(Execute):

- Matematiksel işlemler (toplama , çıkarma vb.) ve mantıksal işlemleri gerçekleştirir.



CPU ' nun Temel Bileşenleri

1. Aritmetik Mantık Birimi(ALU)
2. Kontrol Birimi
3. Registerler(Yazmaçlar)

1) ALU(Aritmetik Mantık Birimi):

- ALU'nun Görevleri
- ALU Nasıl Çalışır
- ALU 'nun Bilgisayar İçindeki Önemi

ALU'nun Görevleri

ALU CPU 'nun içindeki tüm akıllı işlerin yapıldığı yerdir . Görevleri iki ana gruba ayrılır;

1. Aritmetik Fonksiyonlar : Sayılarla yapılan dört işlemden sorumludur(+,-,x,/). Modern ALU 'lar bunları nanosaniyeler içinde bitirir.
2. Mantıksal (Logical) Fonksiyonlar : Karar verme merkezidir. İki veriyi birbiriyle kıyaslar(Eşit mi? Büyük mü? Küçük mü?). Ayrıca AND , OR , XOR , NOT gibi kapı işlemlerini yürüterek program akışını belirler.

ALU Nasıl Çalışır?

ALU tek başına karar veremez , ona ne yapacağı söylenir ve süreç şu şekilde ilerler;

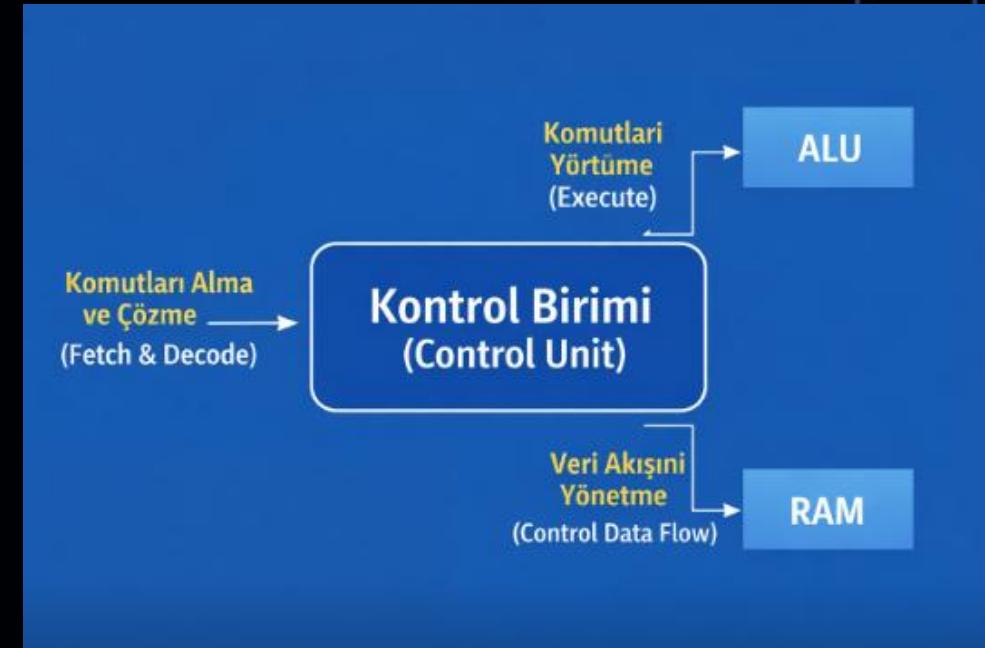
1. Operantların Gelmesi : İşlem yapılacak sayılar(örneğin 3 ve 5),register dediğimiz hızlı kutucuklardan ALU'nun girişine gelir.
2. İşlem Kodunun Belirlenmesi (Opcode) : Kontrol birimi ALU'ya bir sinyal gönderir. (Örneğin,bu iki sayıyı topla)
3. İşleme Geçiş : ALU içindeki transistörler (mantık kapıları) birleşerek elektrik sinyallerini yönlendirir ve sonucu hesaplar.
4. Bayrakların(Flags) Güncellenmesi: İşlem bittiğinde bir ‘durum’ oluşur. Örneğin sonuç 0 çıktıysa ve ya bir hata oluştuysa Durum Yazmacı (Status Register) kısmında ilgili lamba yanar.

ALU'nun Bilgisayar İçindeki Önemi

- ❖ Gerçek Hesaplayıcıdır : Kontrol birimi sadece veri taşıır ve komut verir; işi fiilen bitiren ALU 'dur.
- ❖ Hız Belirleyicidir : Bir işlemcinin hızı (GHz), ALU 'nun bu işlemi ne kadar hızlı döngüye sokabildiğiyle doğrudan ilişkilidir.
- ❖ Karar Mekanizmasıdır : Bilgisayardaki tüm "eğer (if)" komutları ALU'nun mantıksal kıyaslaması sayesinde çalışır.(Örneğin; eğer şifre doğruysa ($A=B$) giriş yap emri burada çözülür).

2) Kontrol Birimi:

- CPU İçindeki işlemleri yönetir
- ALU 'ya hangi işlemlerin yapılacağını söyler
- Veri akışını kontrol eder



3) Registerlar (Yazmaçlar):

- CPU içindeki çok hızlı belleklerdir.
- Registerlar geçici verileri tutarlar.
- RAM 'den çok daha hızlıdır.
- Program Counter (PC), Instruction Register (IR), Flag Register gibi örnekler verebiliriz.



VERİ MANİPÜLASYONU

Veri Manipülasyonu Nedir?

- Veri manipülasyonu, verinin daha kolay okunması, analiz edilmesi veya belirli bir amaç için (karar verme , raporlama) daha uygun hale getirilmesi amacıyla düzenlenmesi veya yapılandırılması sürecidir.

- Veri manipülasyonunda veriler bit seviyesinde işlenir.
- Veri manipülasyonu performans ve bellek optimizasyonu sağlar.
- Sistem programlamada yaygındır.

BITLER VE İKİLİ (BİNARY) SİSTEM:

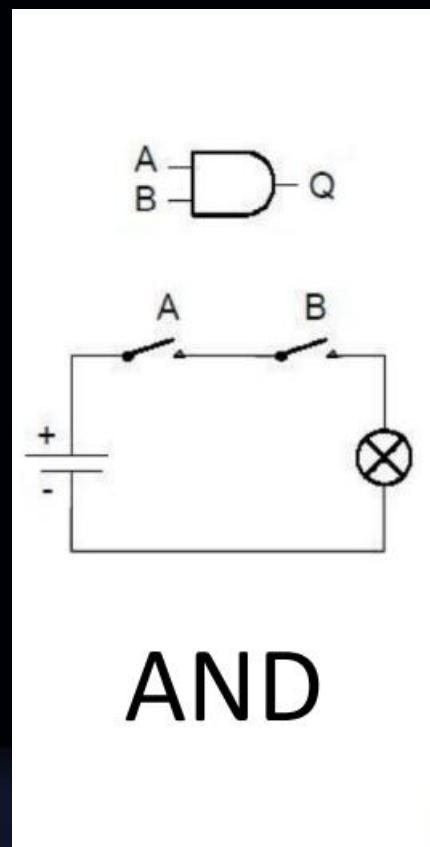
- ❖ Bilgisayarlar 0 ve 1'lerle çalışır.
- ❖ Her bit mantıksal bir durumu temsil eder
- ❖ Donanım temelliidir
- ❖ Mantık kapılarının temelini oluştururlar

MANTIK KAPILARI

- ✓ Mantık kapıları dijital devrenin temel yapı taşılarıdır.
- ✓ Temelinde ikili (binary) sistem kullanılır;
 - 1 (true/doğru) : yüksek voltaj
 - 0 (false/ yanlış) : düşük voltaj
- ✓ Giriş çıkış mantığıyla çalışırlar.
- ✓ ALU ' nun temellerini oluştururlar.
- ✓ Temel mantık kapıları şunlardır;
 - AND , OR , XOR , NOT

1-AND (VE) KAPISI

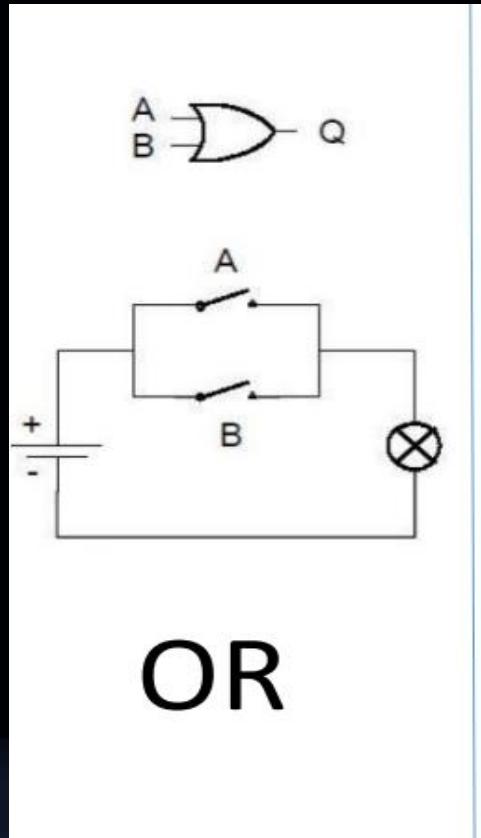
Giriş	Giriş	Çıkış
0	0	0
1	0	0
0	1	0
1	1	1



- AND kapısı kontrol devrelerinde kullanılır.
- Mantığı çarpmaya benzer
- Tüm girişler 1 olduğunda çıkış 1 olur diğer durumlarda çıkış 0'dır.

2- OR (VE YA) KAPISI

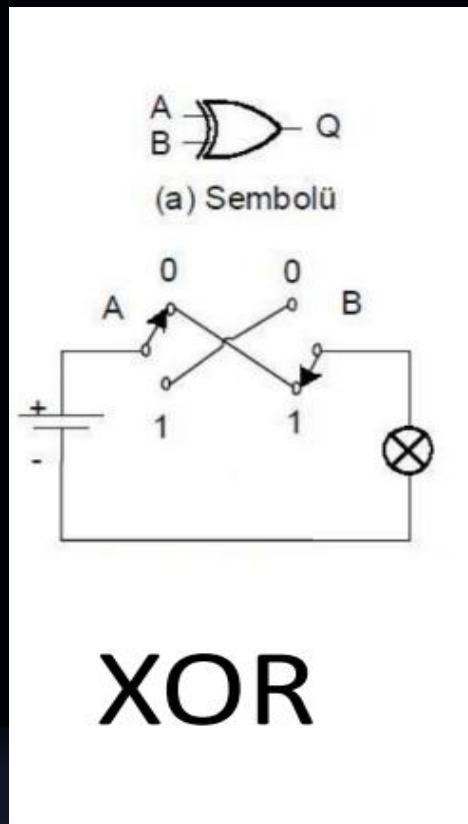
Giriş	Giriş	Çıkış
0	0	0
1	0	1
0	1	1
1	1	1



- En az bir giriş 1 olduğunda çıkış 1 olur. Ancak iki girişte 0 olursa çıkış 0 olur.

3-XOR KAPISI

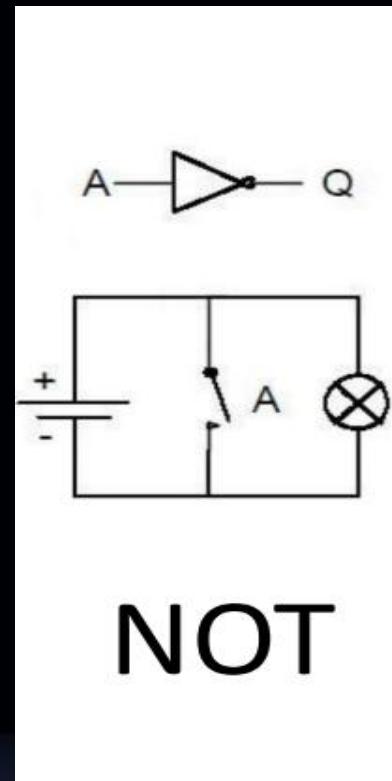
Giriş	Giriş	Çıkış
0	0	0
1	0	1
0	1	1
1	1	0



- XOR kapısında girişler farklıysa çıkış 1 , girişler aynıysa çıkış 0 olur.
- Karşılaştırma ve hata tespitinde kullanılır.

4-NOT KAPISI

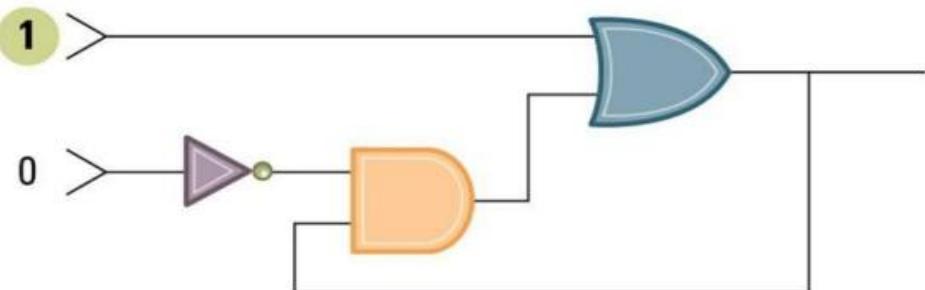
Giriş	Çıkış
1	0
0	1



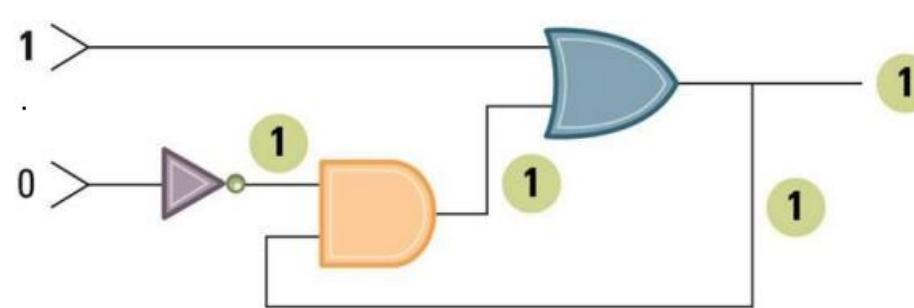
- Not kapısı girişi tersine çevirir.
- Tek girişlidir.

ÖRNEK1

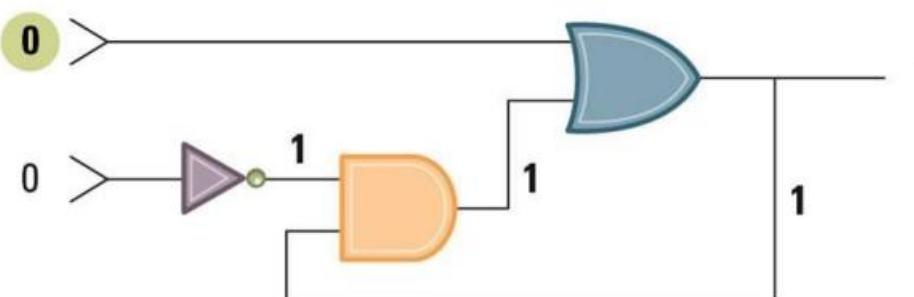
a. İlk olarak üstteki giriş değerine 1 atanır.



b. Bu, OR kapısının çıkışının 1 olmasına ve buna bağlı olarak AND kapısının da çıkışının 1 olmasına sebep olur



c. Son olarak, AND kapısından gelen 1 değeri, OR kapısındaki değer 0'a döndüğünde bile sonucun değişmesini önler.



ÖRNEK 2

A	B	C	A AND B	Çıkış
0	0	0	0	0
0	0	1	0	1
0	1	0	0	0
0	1	1	0	1
1	0	0	0	0
1	0	1	0	1
1	1	0	1	1
1	1	1	1	1

➤ $(A \text{ AND } B) \text{ OR } C$

Örnekte ilk olarak $(A \text{ AND } B)$ işlemi yapılır
daha sonra oluşan sonuçla C işleme sokulur.

Mesela A=0 , B=1 iken A AND B 0 olur
ve bu sonuçla C=0 OR kapısı ile işleme
girerse sonucumuz 0 olur.

ÖRNEK 3

A	B	C	A OR B	Çıkış
0	0	0	0	0
0	0	1	0	0
0	1	0	1	0
0	1	1	1	1
1	0	0	1	0
1	0	1	1	1
1	1	0	1	0
1	1	1	1	1

➤ (A OR B) AND C

Örnekte ilk olarak (A OR B) işlemi yapılır
daha sonra oluşan sonuçla C işleme sokulur.

Mesela A=0 , B=1 iken A OR B 1 olur ve bu
sonuçla C=1 AND kapısı ile işleme girerse
sonucumuz 1 olur.

KAYNAKÇA

https://ekampus.btu.edu.tr/pluginfile.php/504335/mod_resource/content/5/CH1-TR-2023_v2%20.pdf

https://ekampus.btu.edu.tr/pluginfile.php/504351/mod_resource/content/3/CH2-TR_2024.pdf

Kitap Adı : Computer Science An Overview (12. baskı)
Yazarlar : J. Gleen Brookshear, Dennis Brylow