

Bilgisayar Donanımı Dersi

Merkezi İşlem Birimi

CPU

Ders Notları

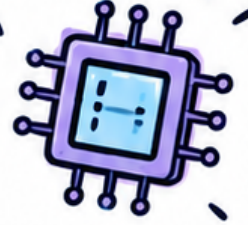
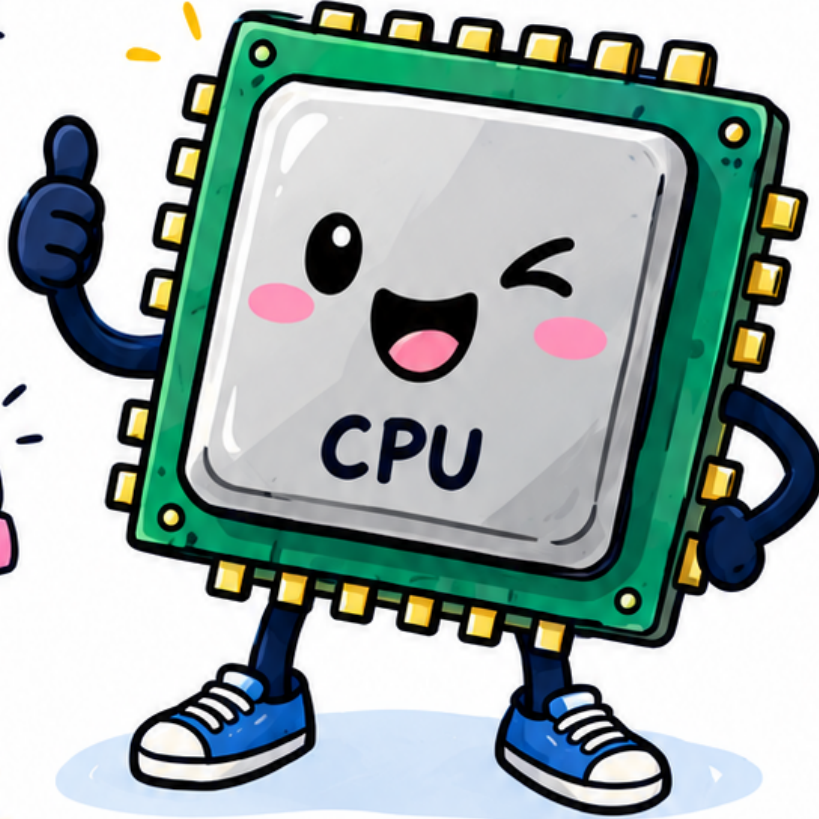


Daha hızlı bilgisayarlar için...

Küçük parçalar büyük işler yapar!



Teknoloji geleceği taşıyor!



Performans için mimari!

Daha Güçlü
Daha Akıllı
Daha Hızlı



Selçuk Üniversitesi
Huğlu Meslek Yüksekokulu

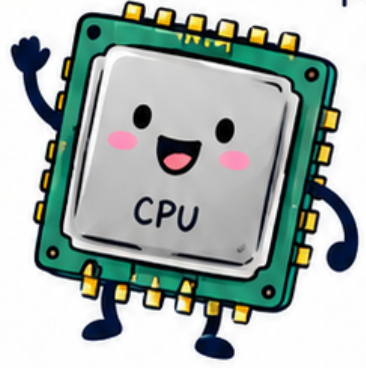
Bilgisayar Donanımı Dersi (Önlisans)

CPU

Merkezi İşlem Birimi

İçindekiler

Düşünür
Hesaplar
Yönetir



Küçük
bir çip,
büyük
bir dünya!



1 İşlemci Mimarisi 1

Temel yapı, bileşenler, veri akışı, mimari türleri

2 Çekirdek 3

Çekirdek nedir? Tek çekirdek vs. çok çekirdek

3 Çok Çekirdekli Mimariler 5

Avantajları, yapısı, paralel işlem



4 İş Parçacığı (Thread) 7

Thread nedir? SMT (Simultaneous Multi-Threading)

5 İş Hattı (Pipeline) 9

Komutların aşamalı işlenmesi, verimlilik



6 Önbellek (Cache) 11

Cache seviyeleri (L1, L2, L3), çalışma mantığı, hit/miss

7 Saat Frekansı 13

GHz nedir? Performansa etkisi, dinamik frekans



8 Komut Setleri 15

ISA (Instruction Set Architecture), x86, ARM, RISC, CISC

9 Intel ve AMD Mimarileri 17

Temel farklar, nesiller, mimari özellikler

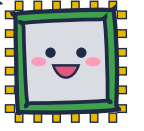


10 Gelişmiş CPU Teknolojileri 19

Hyper-Threading, Turbo Boost, Zen, 3D V-Cache, yapay zeka destekleri



Ben bilgisayarın beyniyim!

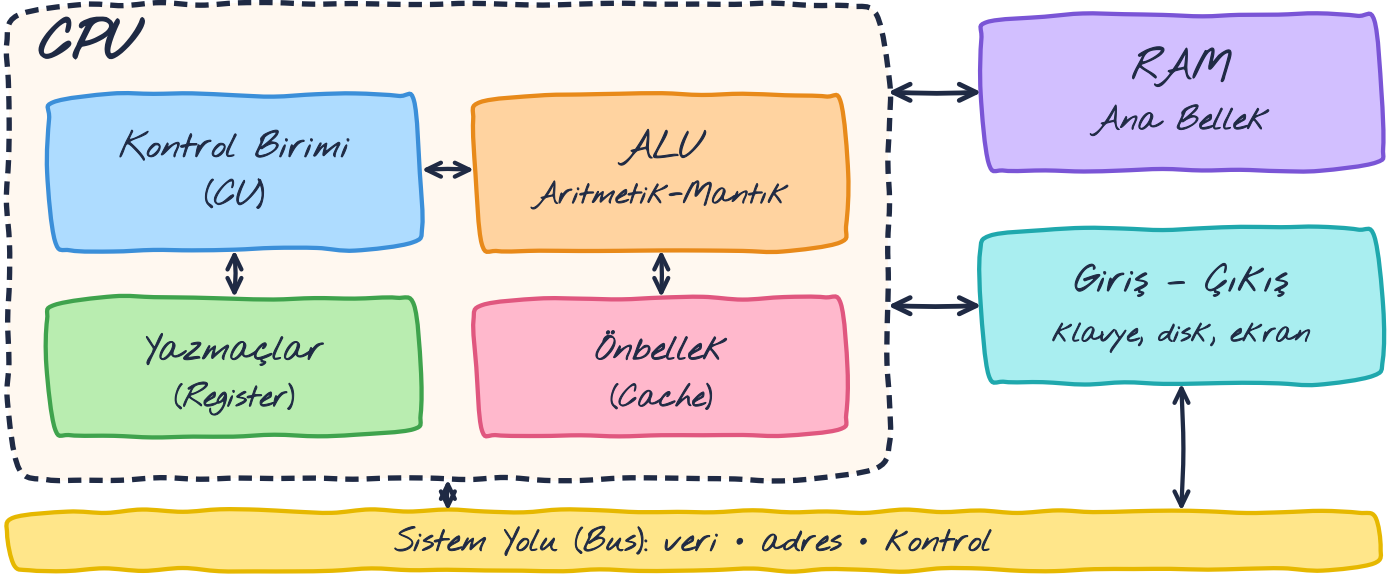


1 İşlemci Mimarisini

CPU nedir, neleri var, nasıl çalışır?

CPU Nedir?

CPU (Central Processing Unit), bilgisayarın beynidir. Bellekten Komutları alır, çözer ve çalıştırır; diğer tüm donanımı yönetir. Yani bilgisayarda olan biten her şey onun Kontrolünden geçer.



Temel Bileşenler

CU Komutu getirir ve çözer, diğer birimlere sinyal yollar. Trafik polisi gibidir.

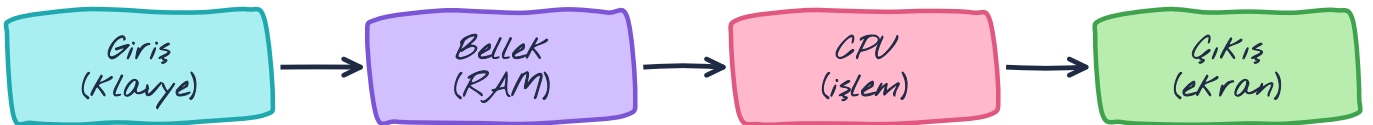
ALU Toplama, çıkarma, AND, OR, NOT gibi aritmetik ve mantık işlemlerini yapar. Hesap makinesi.

Yazmaç CPU içindeki en hızlı ve en küçük bellek. İşlenen veriyi anlık tutar. Not defteri.

Cache Sık kullanılan veriyi CPU'ya yakın tutar. Masa çekmecesini gibi hızlı erişim sağlar.

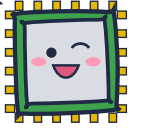
Bus Bileşenler arasında veri, adres ve Kontrol bilgisi taşır. Otoyol görevi görür.

Veri Akışı



İpucu: CPU tek başına çalışmaz; bellek ve giriş/çıkış birimleriyle sürekli veri alışverişi yapar.

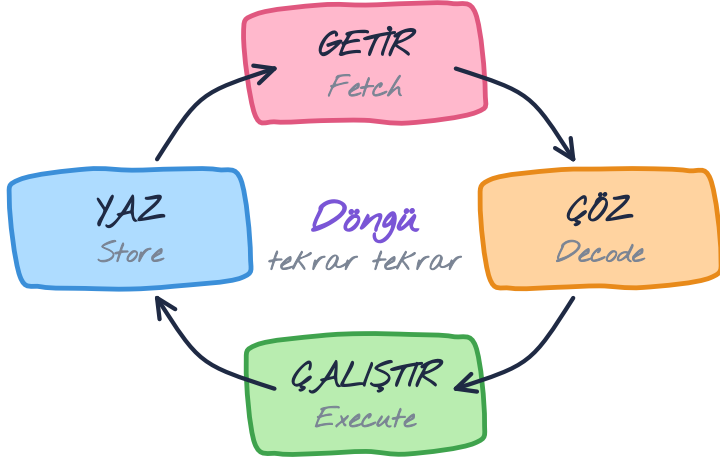
Getir, çöz,
çalıştır!



1 İşlemci Mimarisini

Komut çevrimi, yazmaçlar ve mimari türleri

Komut Çevrimi (Fetch-Decode-Execute)



- 1 Program sayacı (PC) sıradaki Komutun adresini verir; Komut bellekten **getirilir**.
- 2 Kontrol birimi Komutu **çözer**: ne yapılacak, hangi veri?
- 3 ALU işlemi **çalıştırır** (topla, karşılaştır, taşı...).
- 4 Sonuç yazmaca veya belleğe **yazılır**; PC bir sonraki Komuta ilerler.

Önemli Yazmaçlar

PC

Program Counter

Sıradaki Komutun adresini tutar

IR

Instruction Register

Şu an çözülen Komutu tutar

ACC

Accumulator

Ara sonuçları saklar

SP

Stack Pointer

Yığın (stack) tepesini gösterir

32 bit mi, 64 bit mi?

Bit sayısı, CPU'nun bir seferde işleyebildiği veri genişliğidir.

- 32 bit: en fazla 4 GB RAM adresler
- 64 bit: çok daha büyük bellek; günümüzün standardı

Mimari Türleri

Von Neumann ve Harvard

Bellek yapısı farkı: EK Konu, sayfa 21

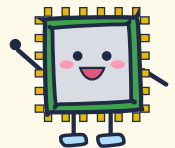
RISC ve CISC

Komut seti yaklaşımı: Bölüm 8

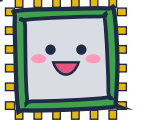


Biliyor muydun?

- 3 GHz'lık bir CPU saniyede 3 milyar çevrim üretir.
- Bir Komut, çevrimlerin bir kısmında biter; modern CPU'lar aynı anda **birden fazla Komutu** işleyebilir (Bölüm 5).
- Komut çevrimi bilgisayar açıkken **hiç durmaz**: getir, çöz, çalıştır, tekrar...



Her çekirdek
bir mini CPU!



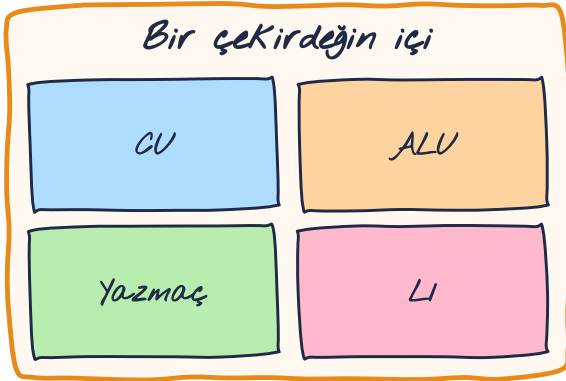
2

Çekirdek

Çekirdek nedir? Tek çekirdek - çok çekirdek

Çekirdek Nedir?

Çekirdek (core), işlemcinin içindeki **bağımsız işlem birimidir**. Kendi Kontrol birimi, ALU'su, yazmaçları ve L1 önbelleği vardır, yani tek başına Komut çalıştırabilir. Modern bir işlemci çipi, bir veya daha fazla çekirdeği aynı silikon üzerinde barındırır.



Çekirdek = mini CPU. Çekirdek sayısı arttıkça aynı anda yapılabilen iş sayısı artar.

Benzetme: Mutfak ve Aşçılar

Tek çekirdek



sipariş 1

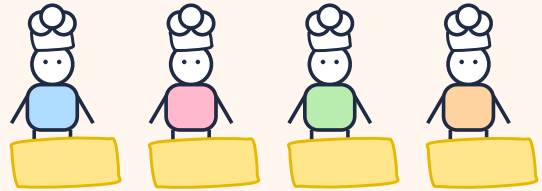
sipariş 2

sipariş 3

sipariş 4

Tek aşçı siparişleri **sırayla** yapar. Sıra uzadıkça bekleme artar.

Çok çekirdek

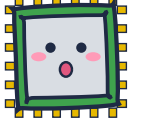


Dört aşçı siparişleri **aynı anda** hazırlar. Toplam süre kısalır.

Çekirdek Sayısına Göre İşlemciler



Hız iki katı
dur mu?



2

Çekirdek

Karşılaştırma ve efsaneler

Tek Çekirdek vs Çok Çekirdek

Özellik	Tek çekirdek	Çok çekirdek
İş yapma şekli	Sırayla	Paralel
Çoklu görev	Zayıf	Güçlü
Güç tüketimi	Düşük (tek çekirdek)	Daha yüksek, ama verimli
Isınma	Yüksek frekansla artar	Yük paylaşılır
Kullanım	Eski PC, basit cihaz	PC, telefon, sunucu

Efsane: "Çekirdek sayısı iki katına çıkarsa hız da iki katına çıkar!"

- **Yanlış.** Yazılım işi parçalara bölemiyorsa fazladan çekirdekler boşta bekler.
- Bir işin sıralı kalan kısmı toplam hızı sınırlar (Amdahl Yasası, sonraki sayfada).
- Bellek hızı ve önbellek de sonucu etkiler; **darboğaz** başka yerde olabilir.

Hangi Cihazda Kaç Çekirdek? (tipik)

Telefon

6-8 çekirdek

Dizüstü

8-16 çekirdek

Oyun PC

8-16 çekirdek

Sunucu

64-192 çekirdek

(Çubuk uzunlukları Karekök ölçeğindedir.)



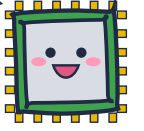
Özet: Çok çekirdek = aynı anda birden fazla iş. Ama gerçek kazanç, yazılımın **paralel çalışabilmesine** bağlıdır.

3

ÇOK ÇEKİRDEKLİ Mimariler

Yapı, avantajlar ve paralel işlem

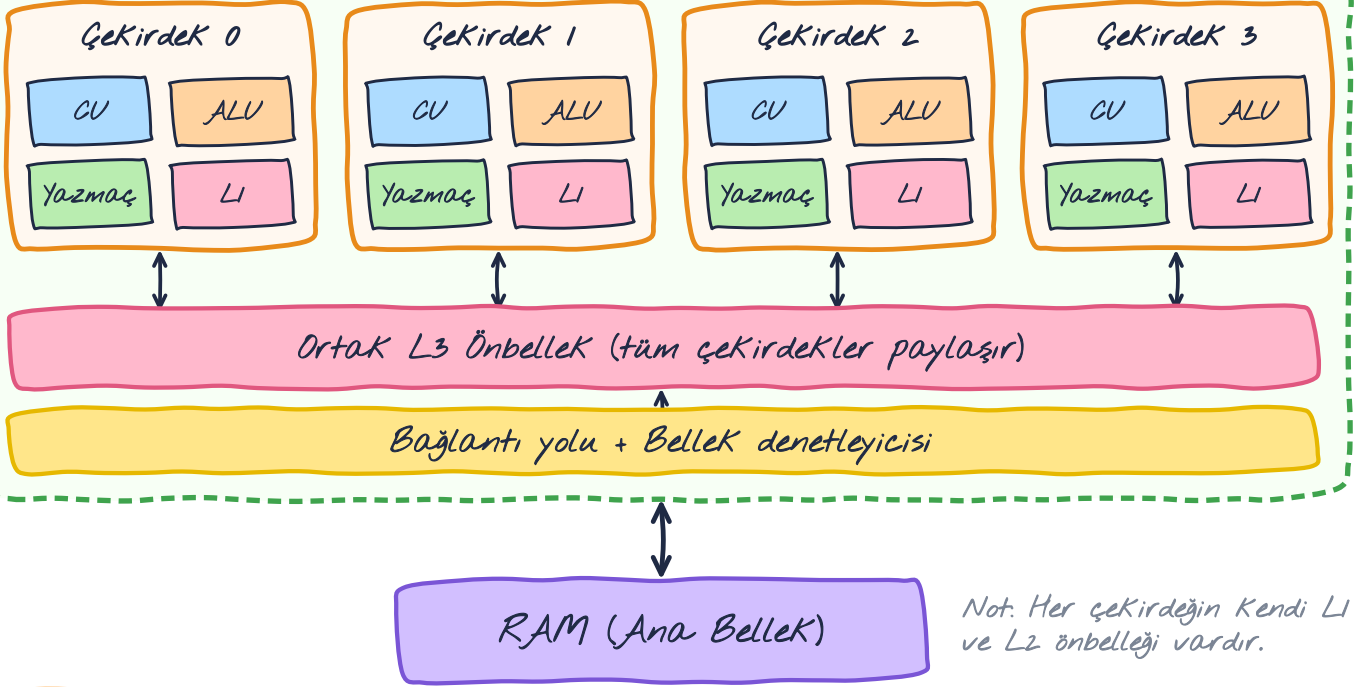
Birlikte daha hızlı!



ÇOK ÇEKİRDEKLİ İşlemci Yapısı



İşlemci çipi (die)



Not: Her çekirdeğin Kendi LI ve L2 önbelleği vardır.

Avantajları

Paralel işlem

Bir iş parçalarına bölünür, çekirdekler aynı anda çalışır.

Çoklu görev

Müzik, tarayıcı ve indirme birbirini yavaşlatmaz.

Enerji verimliliği

Düşük frekanslı çok çekirdek, tek yüksek frekansa az güç harcar.

Isı dağılımı

Yük çipe yayılır, tek noktada aşırı ısınma azalır.

Paralel İşlem Örneği: 8 saniyelik iş

1 çekirdek



4 çekirdek



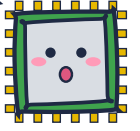
İdeal durumda
8 sn yerine 2 sn = 4 Kat hızlı!

3

Çok Çekirdekli Mimariler

Hibrit çekirdekler, Amdahl Yasası, Kullanım alanları

Bu formül önemli!



Homojen ve Heterojen (Hibrit) Yapı

Homojen



Tüm çekirdekler aynıdır. Basit ve öngörülebilir. (Örnek: AMD Ryzen masaüstü serisi)

Heterojen (hibrit)



P = performans, E = verimlilik çekirdeği. Ağır iş Pde, hafif iş Ede. (Intel hibrit, ARM bigLITTLE, Apple M)

Amdahl Yasası

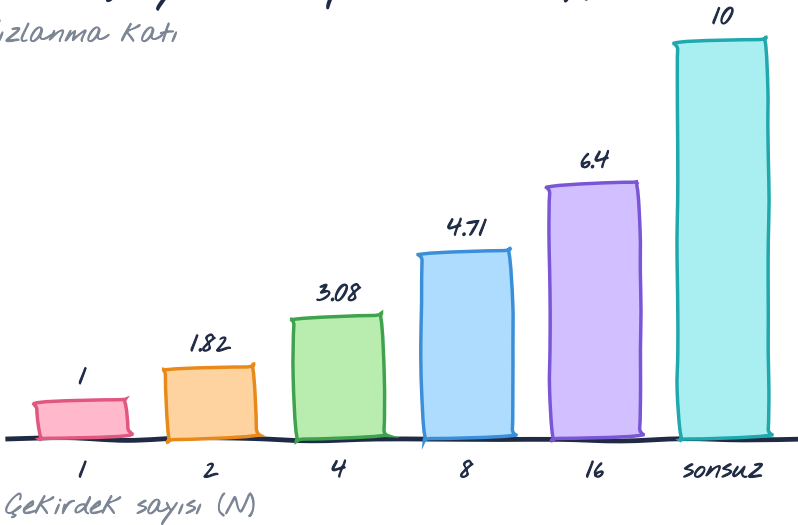
$$\text{Hızlanma} = 1 \div ((1 - P) + P \div N)$$

P: işin paralel yapılabilen oranı

N: çekirdek sayısı

Örnek: işin yüzde 90'ı paralel (P = 0,9)

Hızlanma Katı



- Sıralı kalan yüzde 10 yüzünden en fazla 10 kat hızlanılır.
- Çekirdek 8'den 16'ya çıkınca Kazanç yalnızca 4,7 ile 6,4 arası.
- Sonuç: yazılımın paralellliği çekirdek sayısı kadar önemli.

Nerede Fark Yaratır?

Oyunlar

Video render

Yapay zeka

Sanal makineler

Bilimsel hesap

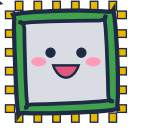
Derleme (compile)

Web sunucuları



Kural: Paralel bölünebilen iş çok çekirdeğin dostu; sıralı çalışan iş tek çekirdeğin hızına bağlı kalır.

İş parçacığı =
Küçük görev!



4

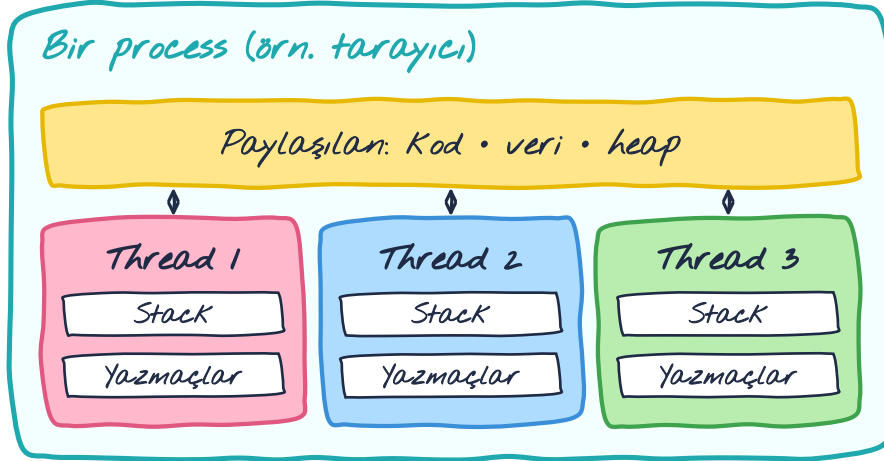
İş Parçacığı (Thread)

Process ve thread farkı, thread durumları

Process ve Thread

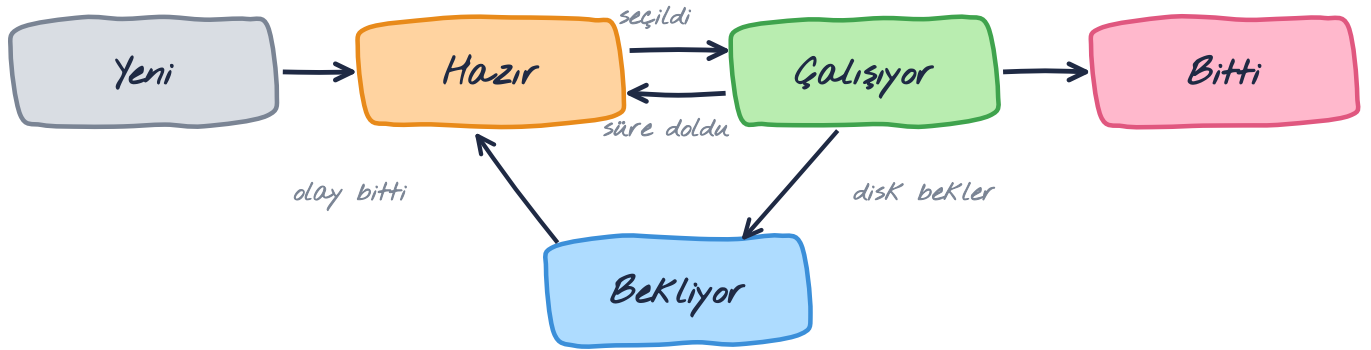
Process çalışan bir programdır ve Kendi bellek alanına sahiptir. Thread ise process içindeki en küçük çalışma birimidir; aynı process'teki thread'ler belleği paylaşır, ama her birinin Kendi yığını (stack) ve yazmaçları vardır.

Bir process (örn. tarayıcı)



Ev = process
Ev halkı = thread'ler
Mutfak = ortak bellek

Thread Durumları



Çoklu İş Parçacığı (Multithreading)

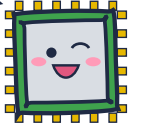
- **Tarayıcı:** her sekme ve eklenti ayrı thread ya da process; biri takılsa diğerleri çalışır.
- **Oyun:** fizik, ses, yapay zeka ve çizim ayrı thread'lerde ilerler.
- **Arayüz:** ağır iş arka plan thread'inde yapılırsa pencere donmaz.
- Çok thread, ancak çok çekirdek (veya SMT) varsa gerçekten aynı anda çalışır.



Özet: Process = ev, thread = evdeki kişi. Aynı evdeki kişiler mutfak (belleği) paylaşır, her kişinin Kendi masası (stack) vardır.



Boşlukları
doldur!



4 İş Parçacığı (Thread)

SMT ve Hyper-Threading: bir çekirdek, iki thread

SMT Nedir?

SMT (Simultaneous Multi-Threading), tek bir fiziksel çekirdeğin işletim sistemine *iki mantıksal işlemci* gibi görünmesidir. Intel'de adı *Hyper-Threading*, AMD'de *SMT*'dir. Amaç: bir thread beklerken boşta kalan yürütme birimlerini diğer thread'e vermek.

SMT Kapalı: yürütme birimleri zaman zaman boşta



SMT açık: ikinci thread (B) boşlukları doldurur



Fiziksel Çekirdek ve Mantıksal İşlemci

4 çekirdek
↓
8 thread

8 çekirdek
↓
16 thread

Apple M serisi
SMT yok
1 çekirdek = 1 thread

İpucu: Görev Yöneticisi > Performans sekmesinde "Çekirdek" ve "Mantıksal işlemci" sayılarını görebilirsin.

Avantajları

- Yürütme birimleri daha *verimli* kullanılır.
- Çok thread'li işlerde yaklaşık yüzde 15-30 kazanç (iş yüküne göre).
- Az ek silikon alanıyla elde edilir.

Dezavantajları

- Tek thread'li işte *kazanç yok*.
- İki thread aynı önbelleği ve birimleri paylaşır.
- Bazı yan kanal güvenlik açıkları görüldü.

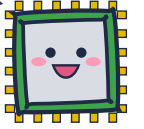
Formül



Thread sayısı = Çekirdek sayısı x 2 (SMT açıkta)



Sırayı bekleme,
akıza Katıl!



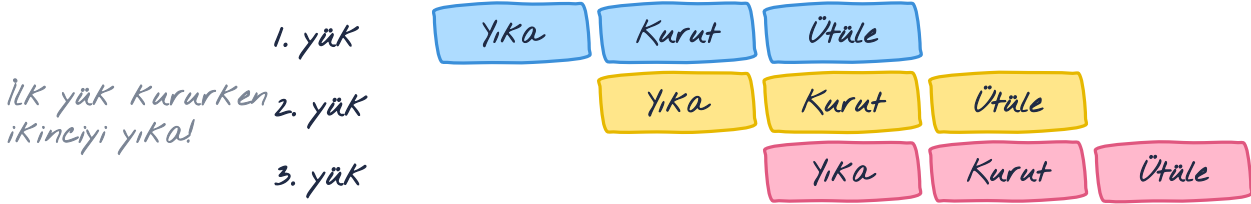
5 İş Hattı (Pipeline)

Komutların aşamalı işlenmesi

Pipeline Nedir?

Pipeline (iş hattı), bir Komutun işlenmesini **aşamalara** bölüp her aşamada farklı bir Komutu **aynı anda** çalıştırma tekniğidir. Fabrikadaki montaj bandı gibi: her istasyon sürekli iş yapar.

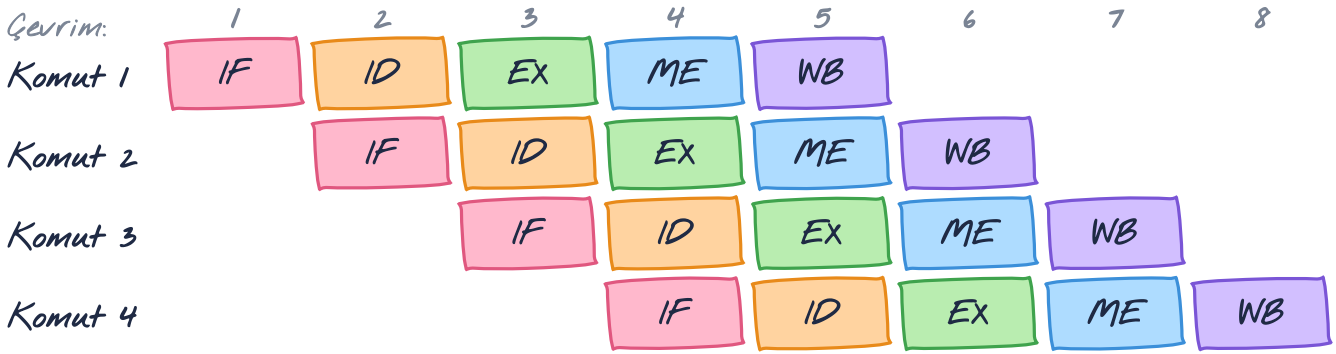
Çamaşırhane benzetmesi:



Klasik 5 Aşamalı Pipeline



Zaman Diyagramı: 4 Komut



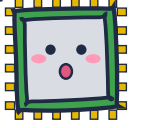
Pipeline yok

4 Komut x 5 çevrim = 20 çevrim.
Komutlar sırayla, biri bitmeden diğeri başlamaz.

Pipeline var

$5 + (4 - 1) = 8$ çevrim. Yaklaşık 2,5 kat hızlı! Komut sayısı arttıkça hızlanma 5 katına yaklaşır.

K aşamalı pipeline, n Komut. toplam çevrim = $K + n - 1$



5

İş Hattı (Pipeline)

Tehlikeler, çözümler ve süperskalar yapı

Pipeline Tehlikeleri (Hazard)

Veri tehlikesi

```
ADD R1, R2, R3
SUB R4, R1, R5
```

İkinci Komut, birincinin sonucunu bekler.

Kontrol tehlikesi

```
if (x > 0) ...
  hangi yol?
```

Dallanma sonucu bilinmeden hangi Komutların getirileceği belirsiz.

Yapısal tehlike

```
Komut A: bellek
Komut B: bellek
```

İki Komut aynı donanımı aynı anda ister.

Çözümler

Forwarding

Sonucu yazmaca yazılmadan doğrudan sonraki Komuta ileterek beklemeyi azaltır.

Bubble (Stall)

Gerekirse boş çevrim ekleyip bekler. Basit ama performans kaybettirir.

Dallanma tahmini

Hangi yolun izleneceğini önceden tahmin eder; modern CPU'larda çok yüksek doğruluk.

Sirasız çalıştırma

Hazır olan Komutu sırayı beklemeden öne alır (out-of-order).

Süperskalar: Birden Çok Hat

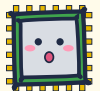


Aynı çevrimde **birden çok Komut** başlar. Böylece IPC'nin üstüne çıkar.

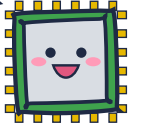


Derin pipeline: iki ucu keskin kılıç

- Kademe artarsa her Kademe Kısılır, **frekans** yükselebilir.
- Ama tahmin hatasında boşalan hat büyür, güç tüketimi artar (Pentium 4 gibi 20 üzeri Kademeli tasarımlar).



Yakın olan
hızlı olur!



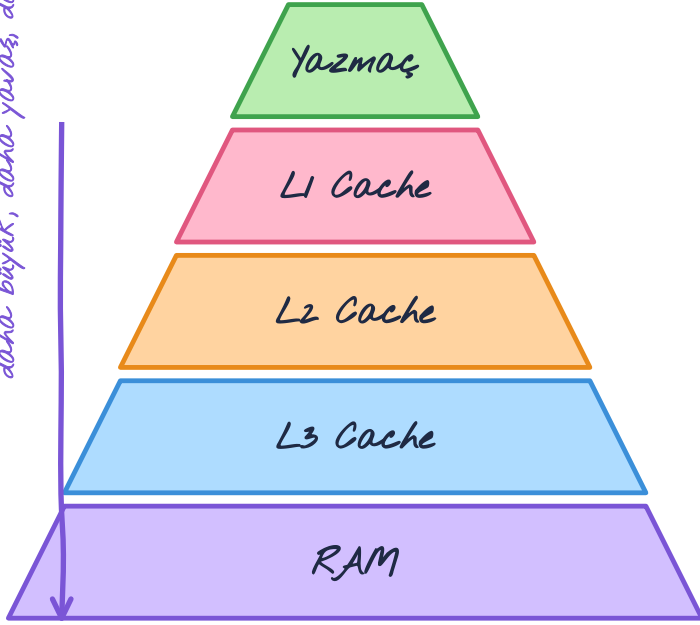
6 Önbellek (Cache)

Cache seviyeleri: L1, L2, L3

Neden Önbellek?

CPU çok hızlıdır, ana bellek (RAM) ise ona göre yavaştır. Önbellek (cache), sık kullanılan veriyi CPU'nun yanında küçük ama çok hızlı bir bellekte tutarak bekleme süresini azaltır.

daha büyük, daha yavaş, daha ucuz



birkaç yüz bayt

gecikme: bir çevrimden az

32 - 128 KB (çekirdek başına)

gecikme: yaklaşık 1 ns

256 KB - 2 MB (çekirdek başına)

gecikme: yaklaşık 3-5 ns

4 - 64 MB ve üstü (ortak)

gecikme: yaklaşık 10-20 ns

8 - 64 GB

gecikme: yaklaşık 60-100 ns

(Değerler tipik ve yaklaşıktır; işlemciye göre değişir.)

Benzetme: Çalışma Masası

L1

Elindeki not
Anında erişilir

L2

Masa çekmecesini
Bir uzarıda

L3

Odadaki dolap
Kalkıp gitmek gerekir

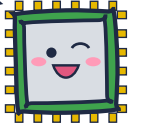
RAM

Bodrumdaki depo
Uzun yürüyüş

Bilmen Gerekenler

- Önbellek **SRAM** ile yapılır: çok hızlıdır ama pahalıdır ve çipte çok yer kaplar. RAM ise **DRAM**'dir.
- L1 genelde Komut (L1I) ve veri (L1D) diye ikiye ayrılır. L2 çoğu tasarımda çekirdeğe özeldir, L3 çekirdekler arasında paylaşılır.
- Cache boyutu büyüdükçe isabet oranı artar; ama boyut büyüdükçe **gecikme** de artar.

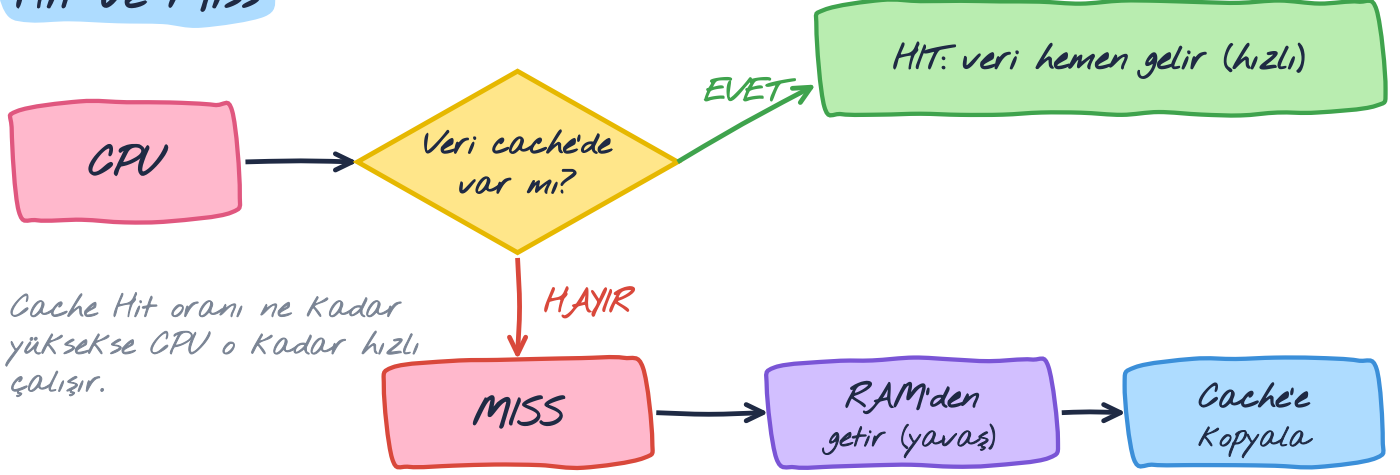
Buldum!
(hit)



6 Önbellek (Cache)

Çalışma mantığı: hit, miss ve süre hesabı

Hit ve Miss



Cache Hit oranı ne kadar yüksekse CPU o kadar hızlı çalışır.

Yerellik İlkesi (Locality)

Zamansal yerellik

Az önce kullanılan veri **yakında** yine kullanılır. (Örnek: döngü sayacı)

Mekânsal yerellik

Bir veri kullanılırken **komşuları** da kullanılır. (Örnek: dizi elemanları)
bir cache satırı (64 bayt) birlikte gelir



Ortalama Erişim Süresi

$$\text{Ortalama} = (\text{Hit oranı} \times \text{Cache süresi}) + (\text{Miss oranı} \times \text{RAM süresi})$$

Örnek: yüzde 95 hit, cache 1 ns, RAM 100 ns

$$0,95 \times 1 + 0,05 \times 100 = 5,95 \text{ ns}$$

RAM'e her seferinde gitseydik:



cache ile: 5,95 ns (yüzde 99 hit olursa 1,99 ns)

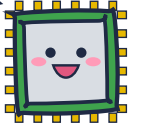
Yer değiştirme (LRU)

Cache dolunca **en uzun süredir** kullanılmayan satır atılır (Least Recently Used).

Yazma politikaları

Write-through: yazma hemen RAM'e gider. **Write-back:** RAM'e sonra yazılır, daha hızlıdır.

TİK-tak, tİK-tak!



7

Saat Frekansı

GHz nedir? Saat sinyali ve birimler

Saat Sinyali

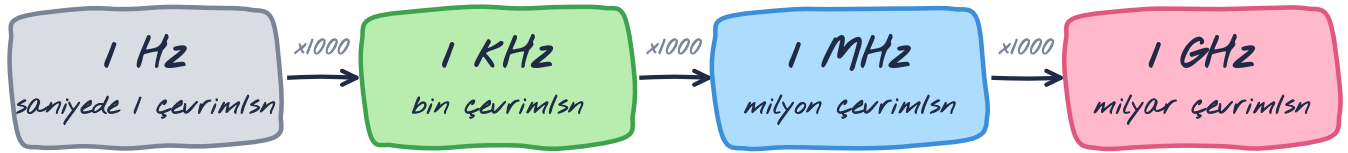
CPU içindeki işlemler bir **saat sinyalinin** ritmiyle ilerler. Saat frekansı, bu sinyalin **saniyedeki çevrim sayısıdır** ve Hertz (Hz) ile ölçülür. Çevrim ne kadar sık olursa, CPU o kadar çok adım atabilir.



bir çevrim (T)

$$T = 1 \div f \quad (\text{süre} = 1 \div \text{frekans})$$

Birimler

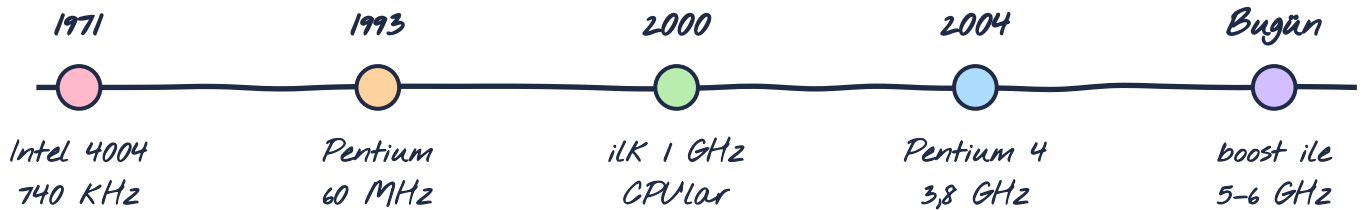


Hesap Yapalım

1 GHz $T = 1 \text{ ns}$	3 GHz $T = 0,33 \text{ ns}$	4 GHz $T = 0,25 \text{ ns}$	5 GHz $T = 0,2 \text{ ns}$
-----------------------------	--------------------------------	--------------------------------	-------------------------------

Frekans arttıkça bir çevrimin süresi kısalır: 1 ns = saniyenin milyarda biri.

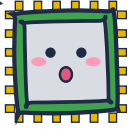
Frekansın Tarihçesi



Not: Frekans yıllarca hızla arttı; ısı ve güç sınırı yüzünden artık artış yavaşladı, çekirdek sayısı ve mimari iyileştirmeler öne çıktı.



GHz tek başına yetmez!



7

Saat Frekansı

Performansa etkisi ve dinamik frekans

Performans Neye Bağlı?

$$\text{Performans} = \text{Frekans} \times \text{IPC} \times \text{Çekirdek sayısı}$$

(yaklaşık ilişki) IPC: bir çevrimde tamamlanan komut sayısı

CPU A

4 GHz • IPC = 2

$$4 \times 2 = 8 \text{ milyar}$$

Komut 1 saniye

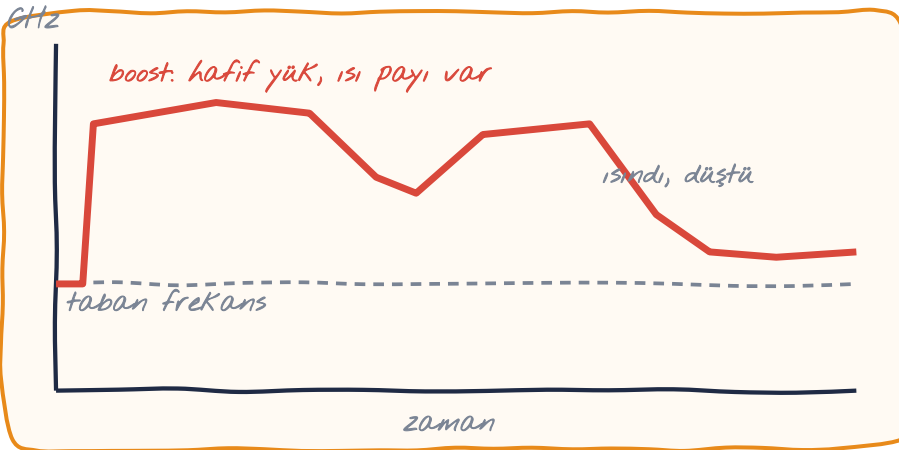
CPU B (Kazanan!)

3,5 GHz • IPC = 3

$$3,5 \times 3 = 10,5 \text{ milyar}$$

Komut 1 saniye

Taban ve Boost Frekansı (Dinamik Frekans)



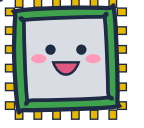
- **Taban:** her koşulda garanti edilen frekans.
- **Boost:** sıcaklık, güç ve akım payı varsa geçici yükselir.
- Intel: Turbo Boost, AMD: Precision Boost.

Isı ve Güç

$$\text{Güç} = C \times V^2 \times f \quad (\text{frekansla birlikte voltaj da artar})$$

- Frekans biraz artınca ısı çok artar.
- **Termal Kısıtlama:** aşırı ısınan CPU frekansını düşürür.
- **Overclock:** frekansını elle yükseltmek; iyi soğutma ister, garantiyi etkileyebilir.

Efsane:
"GHz yüksekse
CPU hızlıdır."
Hayır! IPC de önemli.

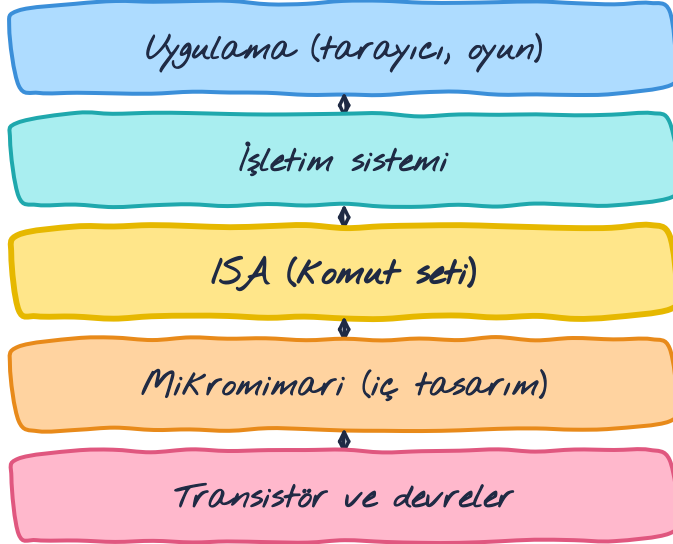


8 Komut Setleri

ISA: CPU'nun dili

ISA Nedir?

ISA (Instruction Set Architecture), yazılım ile donanım arasındaki sözleşmedir. CPU'nun anladığı Komutları, yazmaçları, veri türlerini ve bellek adresleme yöntemlerini tanımlar. Aynı ISA'yı destekleyen CPU'lar aynı programı çalıştırabilir.



Assembly örneği (x86)

`MOV AL, 5` $AL = 5$

`ADD AL, 3` $AL = AL + 3$

Makine Kodu (onaltılık):

`B0 05 04 03`

Assembly, insanın okuyabildiği; makine Kodu CPU'nun anladığı biçimdir.

Komut Türleri

Veri taşıma
`MOV, LOAD, STORE`

Aritmetik ve mantık
`ADD, SUB, AND, OR`

Kontrol akışı
`JMP, CALL, RET`

Giriş-çıkış sistem
`IN, OUT, INT`

Komut Biçimi (basit gösterim)

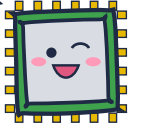


İşlem Kodu (opcode) ne yapılacağını, diğer alanlar hangi veriyle yapılacağını söyler.

Bilinen ISA Aileleri



Lego mu,
çakı mı?



8

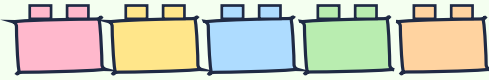
Komut Setleri

RISC ve CISC yaklaşımları

RISC ve CISC

Özellik	RISC	CISC
Komut sayısı	Az ve basit	Çok ve Karmaşık
Komut uzunluğu	Genelde sabit	Değişken
Bellek erişimi	Yalnızca yükle-sakla	Çoğu Komut bellekle çalışır
Kod boyutu	Daha büyük	Daha Küçük
Donanım	Sade, düşük güç	Karmaşık çözücü
Örnekler	ARM, RISC-V, Apple M	x86-64 (Intel, AMD)

RISC = Lego parçaları



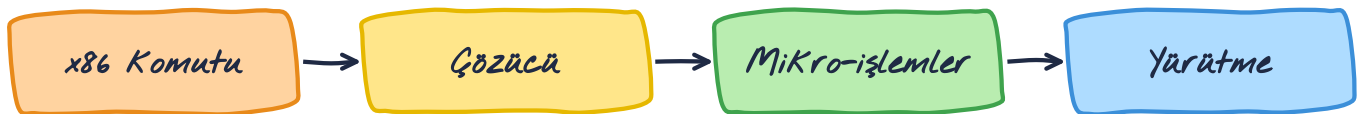
Küçük, basit parçalar; ama birleştirilerek her şey yapılır.

CISC = İsveçre çakısı

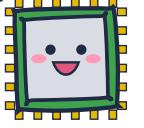


Tek araçta çok iş; her Komut çok şey yapar.

Bugün Sınır Bulanıklaştı



- Modern x86 (CISC) CPU'lar Komutları içeride Küçük mikro-işlemlere çevirir; yani içten RISC gibi çalışır.
- ISA ile mikromimari farklıdır: Intel ve AMD ikisi de x86-64 kullanır, ama iç tasarımları farklıdır.
- SIMD uzantıları (SSE, AVX): tek Komutla birden çok veriyi işler; grafik, ses ve yapay zekada işe yarar.

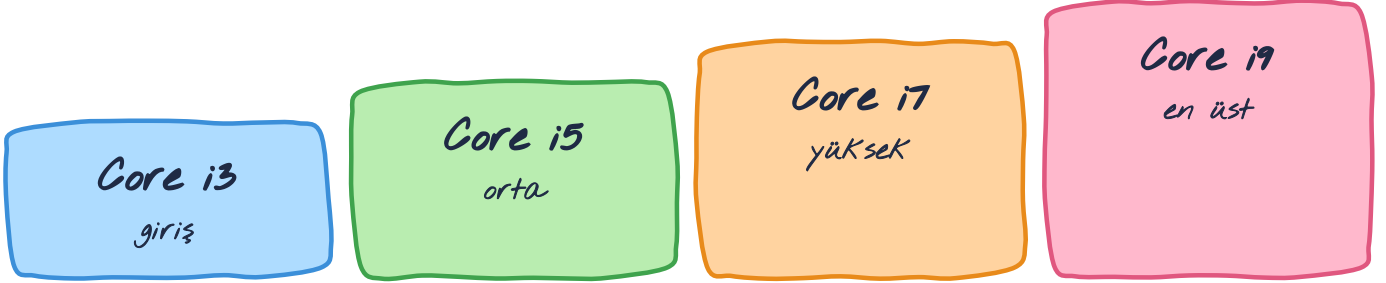


9

Intel ve AMD Mimarileri

Intel: Core ailesi, nesiller ve model kodları

Intel Core Ailesi



Yeni adlandırma: Core Ultra 5, 7, 9 (2023 sonrası nesillerde). Sayı büyüdükçe performans katmanı yükselir.

Nesiller ve Mimariler

2015	Skylake	14 nm, çoğu masaüstü modelde 4 çekirdek
2021	Alder Lake	İlk hibrit: P + E çekirdek, LGA1700 soket
2022	Raptor Lake	Daha fazla E çekirdek, daha yüksek frekans
2023	Meteor Lake	Tile (chiplet) tasarım, ilk Intel NPV
2024	Arrow Lake	Core Ultra 200S, LGA1851, Hyper-Threading yok
2025-26	Panther Lake	Intel 18A üretimi, dizüstü (Core Ultra 3)
Planlı	Nova Lake	Yeni nesil masaüstü, LGA1954 soket

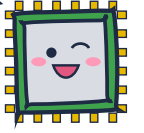
Model Sonekleri

K	çarpmanı açık (overclock)	F	iGPU yok, ekran kartı gerekir
V	düşük güçlü ince dizüstü	H, HX	yüksek performanslı dizüstü
T	düşük güçlü masaüstü	KF	K + F birlikte

Intel'e özgü teknolojiler

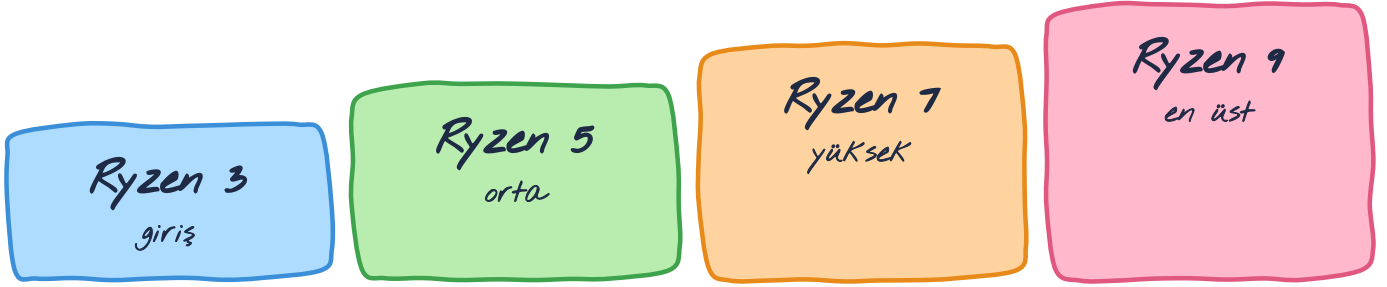
Thread Director (işletim sistemine iş yükünü P ya da E çekirdeğe yerleştirmesi için ipucu verir), Turbo Boost, Xe tamleşik grafik ve yeni nesillerde NPV.





AMD: Ryzen ve Zen mimarisi, Intel ile Karşılaştırma

AMD Ryzen Ailesi

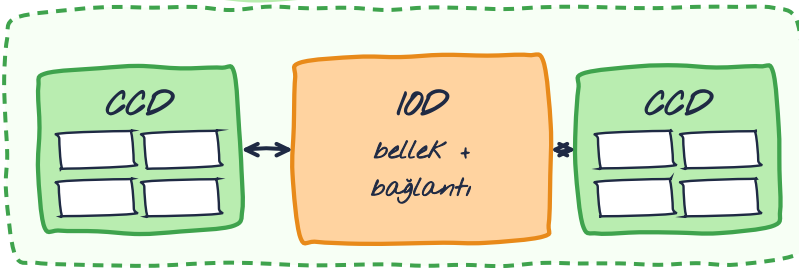


Ayrıca: Threadripper (iş istasyonu) ve EPYC (sunucu) aileleri.

Zen Mimarisi Nesilleri

2017	Zen	Ryzen 1000, AMD'nin güçlü dönüşü, SMT
2019	Zen 2	7 nm, chiplet tasarımı (CCD + IOD)
2020	Zen 3	Ortak L3 bloğu, belirgin IPC artışı
2022	Zen 4	5 nm, AM5 soket, DDR5 (Ryzen 7000)
2024	Zen 5	Ryzen 9000, daha geniş çekirdek
Beklenen	Zen 6	sonraki nesil

Chiplet Tasarımı



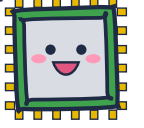
- **CCD:** çekirdeklerin bulunduğu küçük yonga.
- **IOD:** bellek ve bağlantı yongası.
- Küçük yonga = düşük maliyet, yüksek üretim verimi.

Infinity Fabric ile bağlanır

Intel ve AMD Yan Yan

(İkisi de x86-64 Komut setini kullanır)

	Intel	AMD
Adlandırma	Core i, Core Ultra	Ryzen
Çekirdek yapısı	P + E hibrit	Zen (mobilde Zen c çekirdekleri)
Yonga tasarımı	Tile (yeni nesil)	Chiplet (CCD + IOD)
Öne çıkan	Thread Director, NPV	3D V-Cache



10

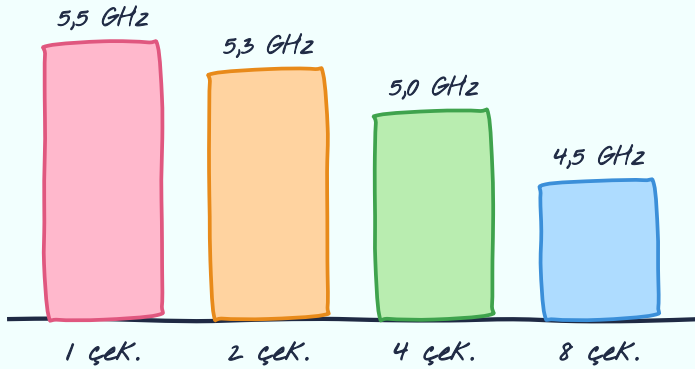
Gelişmiş CPU Teknolojileri

Boost teknolojileri, çekirdek çeşitliliği, güç yönetimi

Turbo Boost ve Precision Boost

CPU, sıcaklık, güç ve akım sınırları elverdiği sürece **frekansı otomatik yükseltir**. Az çekirdek çalışırken boost daha yüksektir; çok çekirdek yüklenince frekans düşer. Intel'de adı **Turbo Boost**, AMD'de **Precision Boost**.

Aktif çekirdek sayısına göre boost (örnek)



Sıcaklık

çok ısınırsa boost biter

Güç Limiti

watt bütçesi aşılamaz

Akım

çekilebilecek amper sınırlı

Çekirdek Çeşitliliği

P çekirdek

Performans: büyük ve hızlı, ağır işler için

E çekirdek

Verimlilik: Küçük ve az güç harcar, arka plan işleri

Zen c çekirdek

AMD'nin kompakt, çok çekirdekli tasarım seçeneği

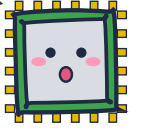
Güç Yönetimi

- **DVFS**: yüke göre frekans ve voltajı anlık ayarlar.
- **C-state (uyku durumları)**: boşta ki çekirdek uyutulur, enerji tasarrufu sağlanır.
- **Güç Kapatma (power gating)**: Kullanılmayan bölümün gücü tamamen kesilir.



Özet: Modern CPU sabit değildir; her an sıcaklık, güç ve yüke bakıp frekansını, voltajını ve hangi çekirdeklerin uyanık olacağını kendisi seçer.



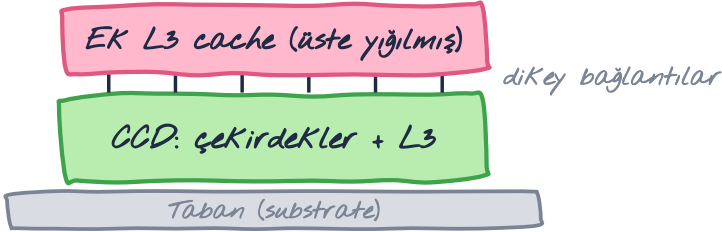


10

Gelişmiş CPU Teknolojileri

3D V-Cache, chiplet, yapay zeka ve gelecek

3D V-Cache



- L3 önbelleği **dikey yığarak** büyütür.
- Oyunlar gibi cache'e duyarlı işlerde belirgin kazanç sağlar.
- Örnek: Ryzen 7 9800X3D, toplam 96 MB L3 (X3D serisi).

Chiplet ve Tile

Büyük bir yonga yerine **küçük modüller** bir arada paketlenir. Böylece üretim verimi artar, maliyet düşer ve farklı üretim süreçleri aynı pakette karıştırılabilir.

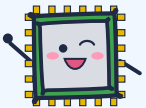
Yapay Zeka ve NPV

CPU Genel işler, esnek	GPU Büyük paralel hesap	NPV Düşük güçle yapay zeka
----------------------------------	-----------------------------------	--------------------------------------

NPV (Neural Processing Unit) yapay zeka işlerini düşük güçle hızlandırır, hızı **TOPS** ile ölçülür. Copilot+ PC'ler için 40 TOPS ve üzeri NPV aranır.

Kendini Test Et

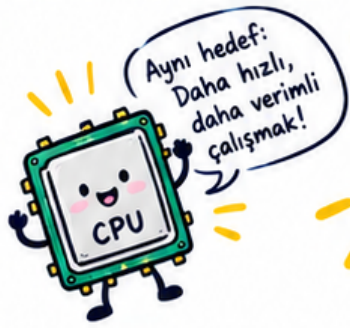
- 1 6 çekirdekli, SMT açık bir CPU kaç thread çalıştırır? (12)
- 2 5 aşamalı pipeline'da 4 Komut kaç çevrimde biter? (8)
- 3 Cache hit oranı artarsa ortalama erişim süresi ne olur? (Kısalır)
- 4 İşin yüzde 10'u sıralıysa en çok kaç kat hızlanılır? (10 Kat)



CPU her zaman sahnede!

Daha güçlü, daha akıllı, daha hızlı...



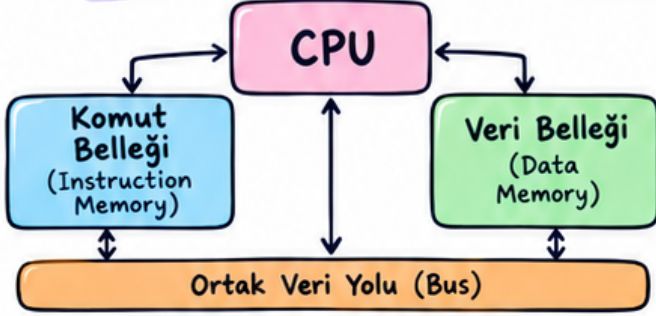


Von Neumann ve Harvard Teknolojileri Karşılaştırması

İki farklı mimari, aynı amaç: Verimi artırmak!

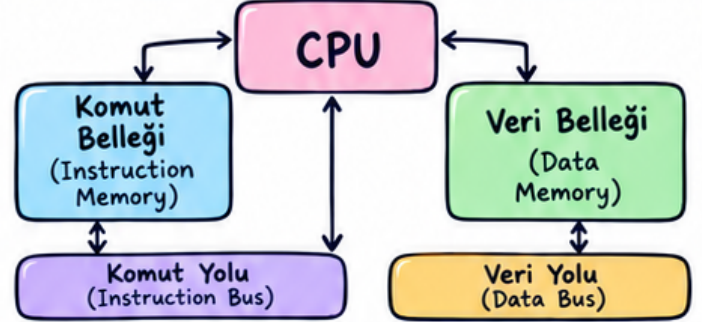


Von Neumann Mimarisi



- Komutlar ve veriler aynı bellekte tutulur.
- Tek bir veri yolu kullanılır.
- Basit ve maliyeti düşüktür.
- Daha çok genel amaçlı sistemlerde kullanılır.
- Bellek veri ve komut erişiminde sıralıdır (ardı ardına gelir).

Harvard Mimarisi



- Komut ve veri bellekleri ayrıdır.
- İki ayrı veri yolu kullanılır.
- Paralel erişim imkânı sağlar.
- Daha yüksek performans sunar.
- Özellikle gömülü sistemler, mikrodenetleyiciler ve yüksek performanslı uygulamalarda tercih edilir.

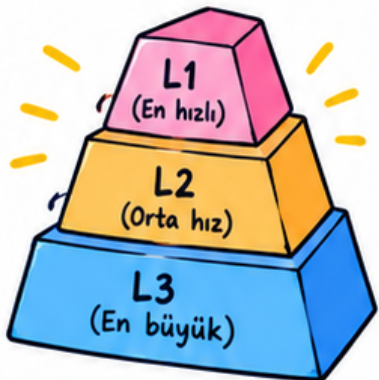
Kısaca Karşılaştırma

Özellik	Von Neumann	Harvard
Komut ve veri belleği	Aynı bellek	Ayrı bellek
Veri yolu sayısı	Tek veri yolu	İki ayrı veri yolu
Erişim	Sıralı	Paralel
Performans	Daha düşük	Daha yüksek
Karmaşıklık	Daha basit	Daha karmaşık
Kullanım alanı	Genel amaçlı sistemler	Gömülü sistemler, DSP, yüksek performans

Harvard mimarisi hız, Von Neumann mimarisi esneklik sunar!

Önbellek (Cache) Seviyeleri

CPU ile ana bellek arasındaki hız farkını azaltmak için kullanılır.



L1 Cache

- CPU içinde bulunur.
- En küçük ama en hızlıdır.
- Genellikle ayrı L1I (komut) ve L1D (veri) olarak ikiye ayrılır.
- Boyut: ~32 KB - 128 KB (çekirdek başına, tipik).

L2 Cache

- Her çekirdeğe özel olabilir (bazı mimarilerde paylaşımlı).
- L1'den daha yavaştır, fakat daha büyüktür.
- Boyut: ~256 KB - 1 MB (çekirdek başına, tipik).

L3 Cache

- Tüm çekirdekler arasında paylaşımlıdır.
- En büyük cache seviyesidir.
- Boyut: ~4 MB - 64 MB+ (işlemciye göre değişir).

Hiyerarşi (genelden özele):

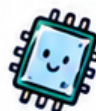


Daha fazla önbellek = Daha az bekleme = Daha yüksek performans! 😊



Donanım bilgisi, geleceğin temeli...

Selçuk Üniversitesi
Huğlu Meslek Yüksekokulu



Küçük mimariler büyük farklar yaratır!