

# BSM 420 – BİLGİSAYAR SİSTEMLERİNİN PERFORMANS DEĞERLENDİRMESİ

2.Hafta: Performans Metrikleri

# İçerik

---

- Tarihçe
- Ölçme ve Değerlendirme
- Bilgisayar Mimarisi Konuları
- İlgili Alanlar
- Bilgisayar Mühendisliği Metodolojisi
- Ölçüm Araçları
- Performans (Başarım)

# Tarihçe

## ■ Bilgisayarları sınıflandırma:

- Mainframe,
- Süper bilgisayar,
- Mini-süper bilgisayar,
- Mini bilgisayar,
- İş istasyonu,
- Kişisel bilgisayar,
- Taşınabilir bilgisayarlar,
- Çok işlemcili sistemler.



Massively Parallel  
Processors



Mainframe



Supercomputer



Mini-  
supercomputer



Mini-  
computer

Work-  
station

PC  
station

# Tarihçe (devam)

## ■ Performans

- Bilgisayar performansı yaklaşık olarak 18 ayda 2 kat artıyor.

	Kapasite	Hız (gecikme)
Lojik Devreler	3 yılda 2 kat	3 yılda 2 kat
DRAM	3 yılda 4 kat	10 yılda 2 kat
Disk	3 yılda 4 kat	10 yılda 2 kat

## ■ Fiyat

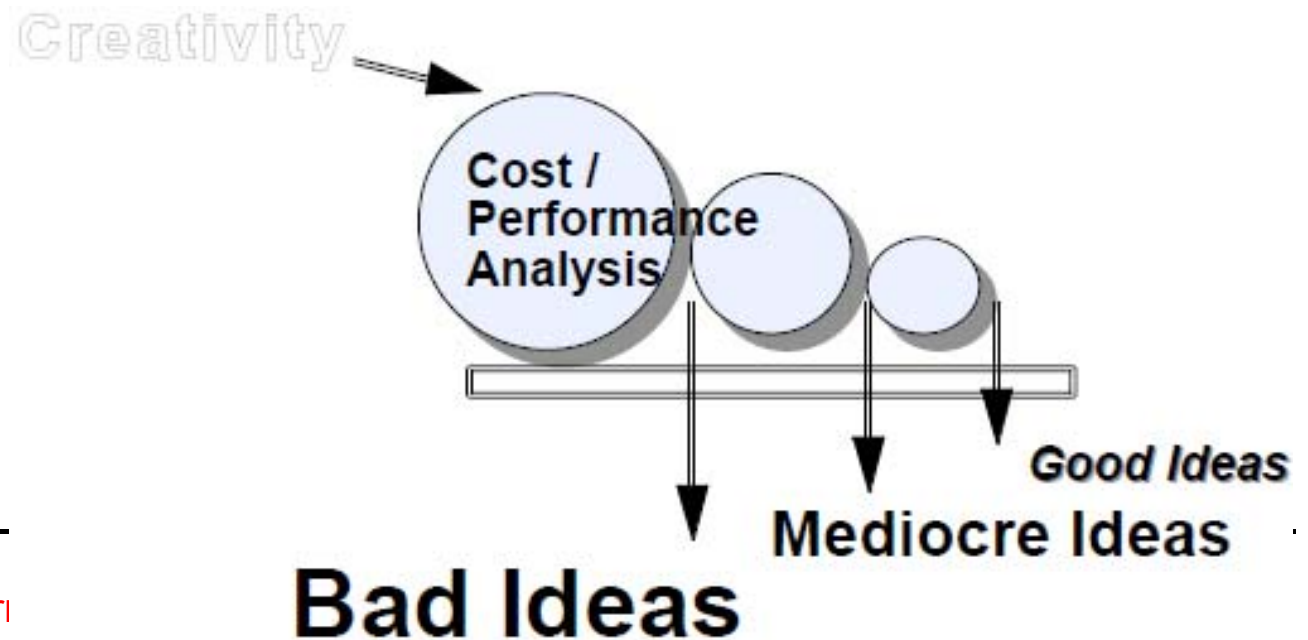
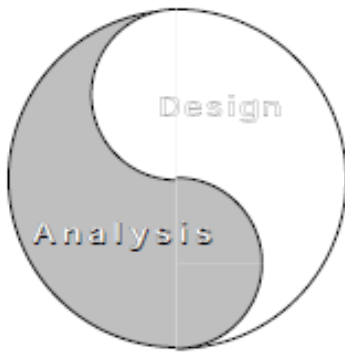
- Yıllık fiyat-performans artışı yaklaşık %70.

## ■ İşlevsellik

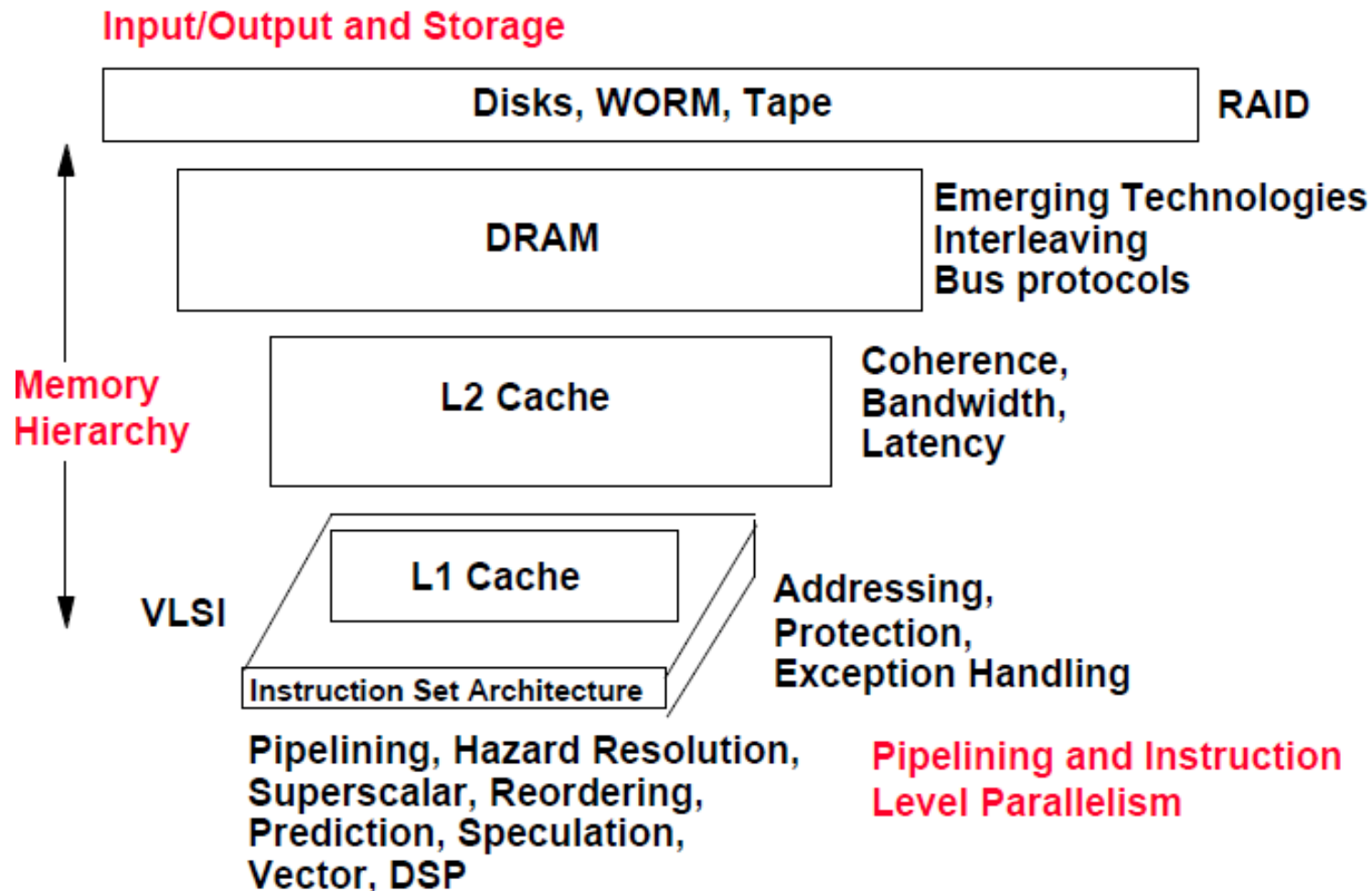
- Network ve yerel iletişim teknolojilerinin artışı

# Ölçme ve Değerlendirme

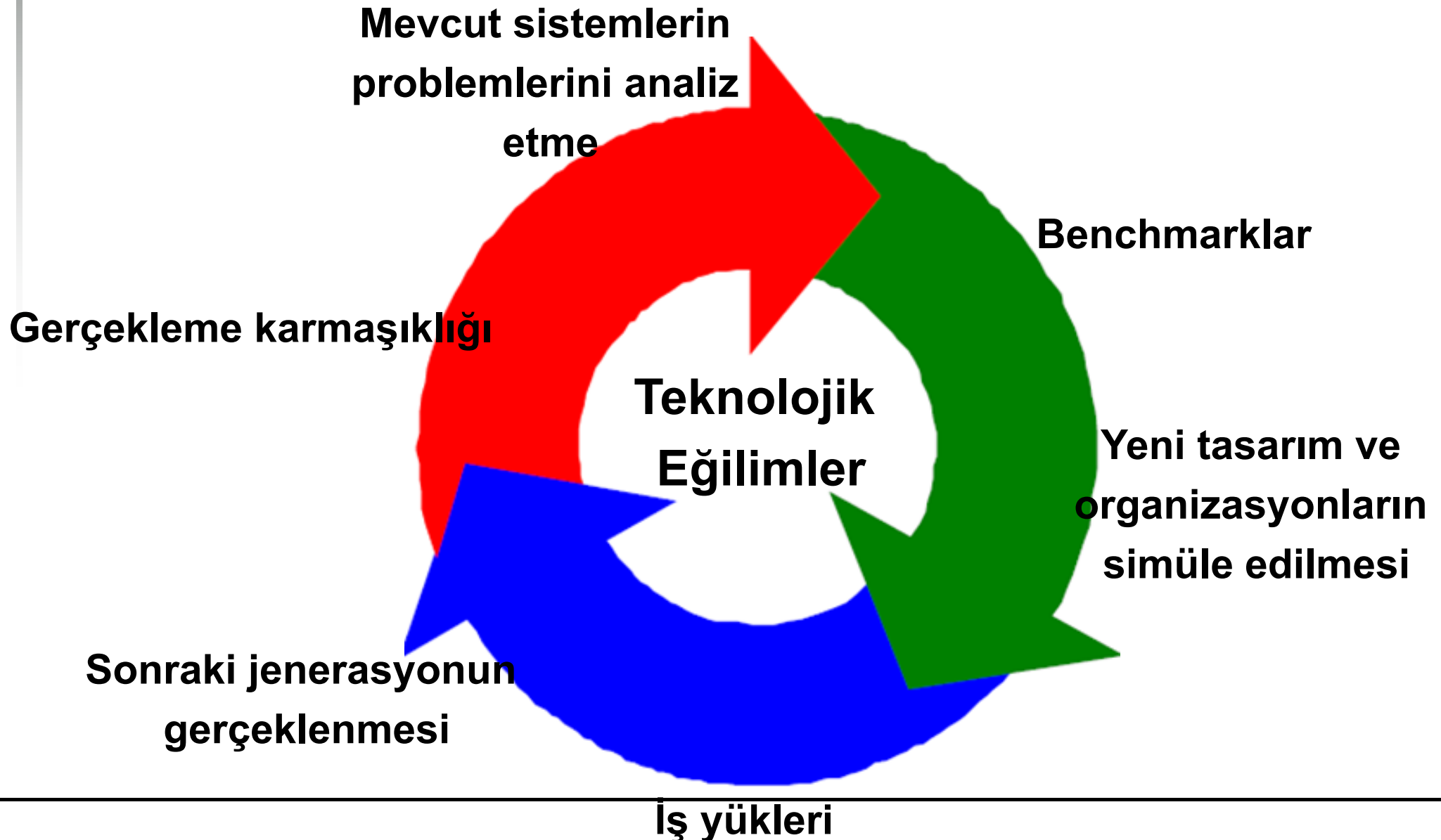
- Mimari üzerinde çalışma iteratif bir süreçtir:
  - Mümkün olan tüm tasarımlar ilgili alanlarda araştırılır
  - Bilgisayar sistemlerinin tüm katmanlarında araştırmalar devam eder.



# Bilgisayar Mimarisi Konuları



# Bilgisayar Mühendisliği Metodolojisi



# Ölçüm Araçları

- Kıyas Kümeleri (Benchmark):
  - çalışma kayıtları ve komut dağılımları (trace and instr. mixes)
- Donanım:
  - Fiyat, gecikme, kullanılan alan miktarı, güç tüketimi
- Simülasyonlar:
  - Komut kümesi seviyesi, saklayıcı seviyesi, kapı seviyesi, devre seviyesi
- Kuyruk teorileri
- Temel kurallar
- Tecrübeler



# PERFORMANS-BAŞARIM

# Performansı Ölçme

- ⑩ İki temel metrik: duvar saati zamanı (bir programın cevap zamanı ) ve çıkış (birim zamanda yerine getirilen görevler)
- ⑩ çıkışı optimize etmek için, asgari kaynak israfı olmalıdır

# Performans (Başarım)

- Bir işi yaparken geçen süre:
  - çalışma zamanı, cevap zamanı, gecikme vs.
- Bir günde, saatte, dakikada, saniyede yapılan iş
  - çıkış(throughput), band genişliği vs.

Plane	DC to Paris	Speed	Passengers	Throughput (pmp)
Boeing 747	6.5 hours	610 mph	470	286,700
BAD/Sud Concode	3 hours	1350 mph	132	178,200

# Bilgisayar Performansı: zaman

- *Cevap Zamanı (Response Time, elapsed time, latency):*
  - Bir görevin çalışması için ne kadar süre gerekli?
  - Başlangıçtan sona kadar tamamlanması için ne kadar süre gerekli?
  - Bir veritabanı sorgusu için ne kadar süre beklenmeli?
- *Performans çıkışı(Throughput):*
  - Tek seferde makine ne kadar işi çalıştırabilir?
  - Ortalama çalışma oranı ne?
  - Bir işin tamamlanabilmesi için ne kadar çalışma gerekli ?
- *Bir makineyi yeni bir işlemci ile yükselttiğimiz zaman ne kadar bir iyileşme sağlanır?*
- *Bir laba yeni bir makine konduğunda ne kadar iyileşme olur?*

# Çalışma Zamanı (Execution Time)

- *Geçen Süre (Elapsed Time)*

- Baştan sona herşeyi dikkate alır (*disk bellek erişimleri, I/O beklemeleri, diğer programların çalışması vs.*)

geçen süre = CPU zamanı+ bekleme zamanı(I/O, diğer programlar vs.)

- *CPU zamanı*

- I/O beklemesi ve diğer programların çalışması için geçen zamanı dikkate almaz
- Kullanıcı CPU zamanı ve sistem CPU zamanı (OS çağrıları) olarak ikiye ayrılır

CPU zamanı= kullanıcı CPU zamanı + sistem CPU zamanı

⇒ geçen süre = kullanıcı CPU zamanı + sistem CPU zamanı+ bekleme zamanı

- *kullanıcı CPU zamanı dikkate alınır (CPU çalışma zamanı veya sadece çalışma zamanı)*

- Program içindeki kod satırlarının çalışması için geçen süre

# Performans Tanımı

- X makinesi üzerinde çalışan bir program için:

$$\text{Performans}_X = 1 / \text{Çalışma Zamanı}_X$$

- *X makinesi Y makinesinden n kat hızlıdır:*

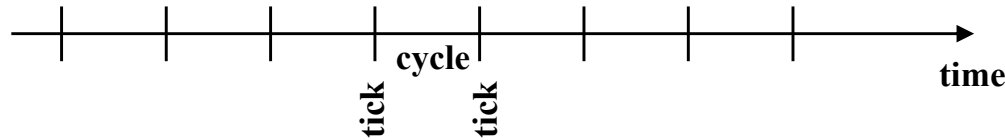
$$\text{Performance}_X / \text{Performance}_Y = n$$

# Saat Çevrimleri

- Çalışma zamanını saniyeler cinsinden sunmak yerine, çoğu kez saat çevrimlerini kullanırız.
- Modern bilgisayarlarda, olaylar çevrim çevrim ilerler : yani çarpma, toplama gibi her olay bir dizi çevrimde gerçekleşir.

$$\frac{\text{seconds}}{\text{program}} = \frac{\text{cycles}}{\text{program}} \times \frac{\text{seconds}}{\text{cycle}}$$

- *Saat tiklemeleri çevrimin başı ve sonunu gösterir:*



- *Çevrim zamanı* = tikler arası zaman = çevrim başına saniye
- *Saat oranı (frekans)* = saniye başına çevrim  
(1 Hz. = 1 çevrim/saniye, 1 MHz. =  $10^6$  çevrim / saniye)

- *Örnek* : 200 Mhz  $\frac{1}{200 \times 10^6} \times 10^9 = 5 \text{ nanoseconds}$  çevrim süresi eder

# Performans Denklemi I

$$\frac{\text{seconds}}{\text{program}} = \frac{\text{cycles}}{\text{program}} \times \frac{\text{seconds}}{\text{cycle}}$$

≡

Bir program için:

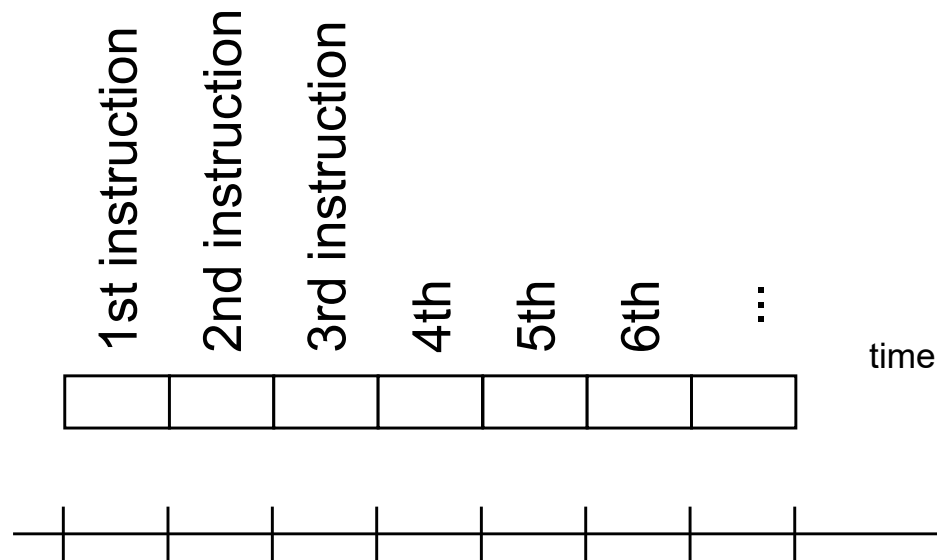
CPU çalışma zamanı = CPU saat çevrim sayısı x Saat çevrimi zamanı

- Dolayısıyla, performansı artırmak için:
  - Bir programın çevrim sayısını düşür veya
  - Saat çevrim zamanını düşür veya
  - Saat frekansını artır (overclock)



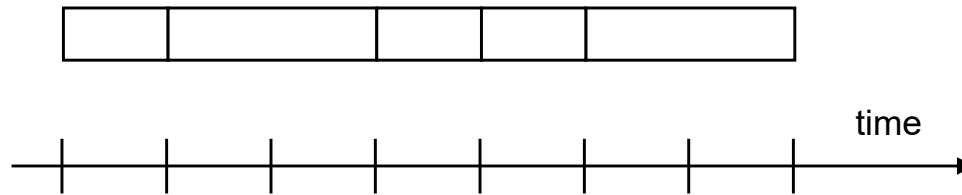
# Bir program için ne kadar çevrim gerekli?

- Çevrim sayısı = komut sayısı ??



- **Bu varsayım kesinlikle yanlıştır! Çünkü:**
  - Farklı komutların işlenmesi farklı zamanlar(çevrimler) alır
  - Neden...?

# Bir program için ne kadar çevrim gerekli?



- Çarpma toplamadan daha çok zaman alır
- Virgüllü (kayan noktalı) işlemler tamsayılardan daha çok zaman alır.
- Belleğe erişim kaydedicilere erişimden daha çok zaman alır
- *Çok önemli:* çevrim zamanını değiştirmek donanım tasarımının değişmesine neden olduğundan komutların çalışması için gerekli çevrim sayısını değiştirir.

# Örnek I

- En sevdiğimiz program 400Mhz olan A bilgisayarında 10 saniye çalışıyor.
- Bir bilgisayar tasarımcısına, bu programı 6 saniye içinde çalıştıracak yeni bir B makinesi tasarlamasına yardımcı olmaya çalışıyoruz.
- Tasarımcı, saat hızını önemli ölçüde artırmak için yeni (veya belki de daha pahalı) teknolojiyi kullanabilir, ancak bu artışın CPU tasarımının geri kalanını etkileyeceğini ve aynı program için B makinesinin A makinesinden 1.2 kat daha fazla saat çevrimi gerektirmesine neden olacağını bize bildirdi..
- Tasarımcıya hangi saat hızını hedef olarak bildirmeliyiz?

# Çözüm:

---

# Terminoloji

- Bir program çalışmak için aşağıdakileri gerektirir:
  - Komutlar
  - Çevrimler
  - Saniyeler
- Bu büyüklükleri aşağıdaki terimler ile açıklarız:
  - *Çevrim zamanı* (çevrim / saniye)
  - *Saat frekansı* (çevrim / saniye)
  - *(ortalama) CPI* (çevrim / komut - cycles per instruction)
    - Kayan nokta yoğun bir uygulamada yüksektir
  - *IPC* (komut / çevrim)
  - *MIPS* (milyon komut / saniye)
    - Basit komutlardan oluşan bir programda yüksek

# Hızlanma Vs. Oran

- ⑩ “Hızlanma” bir orandır. «**kat**», «**kere**» ile ifade edilir

= eski çalışma zamanı / yeni çalışma zamanı

- ⑩ “İyileşme- Bağlı Değişim ”, “**artış**”, “**düşüş**” genellikle belirli bir referansa göre yüzdelik değeri verir

- ⑩ «**daha**» fazla, hızlı, yavaş, çok...

= (yeni perf – eski perf) / eski perf

Bağlı değişim = hızlanma - 1

## ÖRNEK 2

Bir program eski laptopta 100 saniye, yenisinde 70 saniye çalışıyor

Hızlanma ne kadar?

$$\text{Hızlanma} = (1/70) / (1/100) = 1.42 \text{ kat}$$

- Performanstaki iyileşme /bağıl değişim oranı ne?

$$(1/70 - 1/100) / (1/100) = 42\% \text{ daha fazla}$$

- Çalışma zamanındaki düşüm ne kadar?

30% daha az

# Etki faktörleri

- Saat çevrim zamanı : teknoloji ve pipeline
- CPI: mimari ve komut seti tasarımı
- Komut sayısı: komut seti tasarımı ve derleyici
- CPI (cycles per instruction) veya IPC (instructions per cycle) matematiksel olarak doğru hesaplanamaz



# CPU Performansını Etkileyen Unsurlar

$$\text{CPU time} = \frac{\text{Seconds}}{\text{Program}} = \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Cycle}}$$

	Inst Count	CPI	Clock Rate
Program	X		
Compiler	X	(X)	
Inst. Set.	X	X	
Organization		X	X
Technology			X

# Örnek CPI hesabı

## Base Machine (Reg / Reg)

Op	Freq	Cycles	CPI(i)	(% Time)
ALU	50%	1	.5	(33%)
Load	20%	2	.4	(27%)
Store	10%	2	.2	(13%)
Branch	20%	2	.4	(27%)
			<hr/> 1.5	

Typical Mix

**Ortalama CPI = 1.5**

# Performans Ölçümü

- *Performans çalışma zamanı ile ölçülür*
- Diğer değişkenler de performans ölçümünde kullanılabilir mi?
  - Programın çalışması için gerekli çevrim sayısı?
  - Programdaki komut sayısı ?
  - Saniye başına çevrim sayısı ?
  - Komut başına ortalama çevrim sayısı ?
  - Saniye başına ortalama komut sayısı
- *ortak bir tuzak: Değişkenlerden birini kabul etmek, başka bir veri yoksa performans göstergesidir.*

# Performans Denklemi II

$$\begin{array}{l} \text{CPU execution time} \\ \text{for a program} \end{array} = \begin{array}{l} \text{Instruction count} \\ \text{for a program} \end{array} \times \text{average CPI} \times \text{Clock cycle time}$$

- *Bu denklemi 1 nolu denklemden türetiniz*

# Performans Denklemi II

CPU execution time = Instruction count  $\times$  average CPI  $\times$  Clock cycle time  
for a program for a program

- Bu denklemi 1 nolu denklemden türetiniz

$$\frac{\text{Komut Sayısı}_{\text{prog}}}{\text{Komut}} \times \frac{\text{Gevrim}}{\text{Gevrim}} = \frac{\text{Gevrim}}{\text{prog}}$$

$\downarrow$   
CPI

# Örnek III (CPI)

- Aynı komut seti mimarisinin (ISA) iki farklı uyarlaması olsun. Bir program için:
  - A makinesi 10ns saat çevrimi zamanına sahip ve  $CPI=2.0$
  - B makinesi 20ns saat çevrim zamanı ve  $CPI = 1.2$
- *Hangi makine ne kadar hızlıdır?*
- *Eğer iki makine de aynı ISA'ya sahip ise, saat hızı, CPI, çalışma zamanı, komut sayısı ve MIPS gibi büyüklüklerden hangileri benzer olur?*

# Çözüm

---

# Örnek IV(CPI)

- Bir derleyici tasarımcısı belirli bir makine için iki farklı kod arasında karar vermeye çalışmaktadır.
- Donanıma bağlı olarak, 3 farklı komut kümesi vardır: A, B ve C sınıfı, ve sırasıyla 1,2,3 çevrim değerlerine sahiptirler.
- İlk kod dizisi 5 komuta sahip:  
2 adet A, 1 adet B, ve 2 adet C
- İkinci kod dizisi 6 komuta sahip:  
4 adet A, 1 adet B, and 1 adet C.
- *Hangi kod daha hızlıdır? Ne kadar hızlıdır? Her kod dizisinin CPI değeri nedir?*



# Örnek IV (CPI)

- Bir derleyici tasarımcısı belirli bir makine için iki farklı kod arasında karar vermeye çalışmaktadır.
- Donanıma bağlı olarak, 3 farklı komut kümesi vardır: A, B ve C sınıfı, ve sırasıyla 1,2,3 çevrim değerlerine sahiptirler.
- İlk kod dizisi 5 komuta sahip:  
2 adet A, 1 adet B, ve 2 adet C
- İkinci kod dizisi 6 komuta sahip:  
4 adet A, 1 adet B, and 1 adet C.
- *Hangi kod daha hızlıdır? Ne kadar hızlıdır? Her kod dizisinin CPI değeri nedir?*

1,1 kat

# Örnek V(MIPS)

- Üç farklı komut sınıfına sahip bir 500 MHz lik makine üzerinde iki farklı derleyici test edilmektedir. Komut setleri Sınıf A, Sınıf B ve Sınıf C ve sırasıyla 1, 2 ve 3 çevrim gerektirmektedir.
- Her iki derleyici de büyük bir yazılım kodu üretmek için kullanılır.
- Derleyici 1, 5 milyar A Sınıfı komutu, 1 milyar B Sınıfı komutu ve 1 milyar Sınıf C komutu ile kod üretir.
- Derleyici 2, 10 milyar A Sınıfı komutu, 1 milyar B Sınıfı komutu ve 1 milyar Sınıf C komutu ile kod üretir.
- *Hangi derleyici MIPS e göre daha hızlıdır?*
- *Çalışma zamanına göre hangi dizi daha hızlıdır?*

# Çözüm