

Bilgisayar Mimarisi

Anahatlar ve Mimariye Giriş

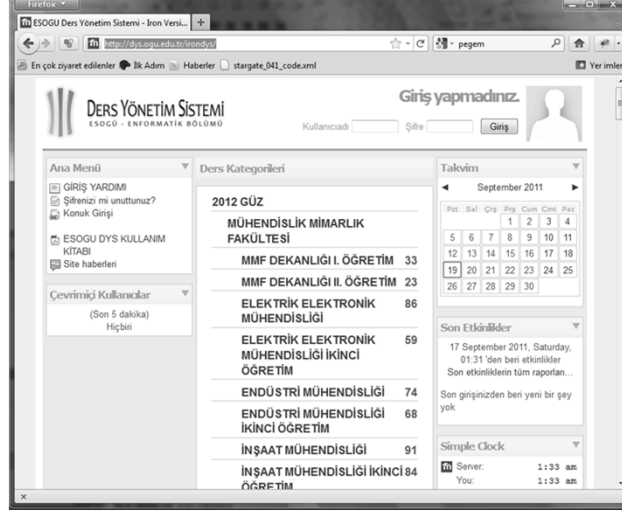
Yrd.Doç.Dr. Celal Murat KANDEMİR

ESOGÜ – Eğitim Fakültesi - BÖTE
twitter.com/cmkanemir

Yardımcı Kaynaklar

- “Computer organization and architecture : principles of structure and function”, William Stallings
- İnternet
- Ders Notları
 - Ders Notları: <http://dys.ogu.edu.tr/silverdys/>

<http://dys.ogu.edu.tr/irondys/>



3

<http://dys.ogu.edu.tr/irondys/>

◆Ekle-Sil döneminden sonra tüm öğrenciler sisteme kayıt edilmiş olacaklar. (26 Eylül 2011 sonrası)

◆**Kullanıcı Adı:** Öğrencinin numarası (Örn: 151319961001)

◆**Şifre:** Öğrencinin T.C. Kimlik numarasının ilk 6 karakteridir. Yabancı uyruklu öğrencilerde ise Y.U. numarasının ilk 6 karakteridir. Eğer bu bilgi ile sisteme giremiyorsanız öğrenci numaranızın ilk 6 karakterini şifre olarak deneyiniz.

4

Değerlendirme

- Vize : %25
- Vize : %25
- Final: %50

5

Ders ve Uygulama Kuralları

- Derse devam zorunludur.
 - Teorik ders çalışmalarının % 70 ine laboratuvar ve uygulama çalışmalarının % 80 ine devam zorunluluđu aranmasının uygun olduđuna karar verilmiştir.
- Ders başlangıç saatlerine özen gösteriniz.

6

Ders ve Uygulama Kuralları

Ders esnasında lütfen kendi aranızda (veya kendi kendinize) konuşmayın, fısıldaşmayın, mesajlaşmayın v.s

Dersi anlatan ve dinleyen kişilere lütfen saygı gösterin.

Anlatılan, anlatılmayan, merak ettiğiniz her konuda soru sormaktan çekinmeyin.

Cep telefonu v.b kişisel taşınabilir iletişim cihazlarınızı ders ve uygulama süresince mutlaka kapalı tutunuz. !!!

7

Ders Kapsamı

- Bu ders:
 - Bilgisayar Organizasyonu ve tasarımı ile ilgili bir giriş niteliğindedir.
 - Bilgisayarın donanımsal işlemlerini anlamak için gerekli olan temel bilgilerin elde edilmesini sağlar.

8

Ders Kapsamı

- Bilgisayar Mimarisi: Bilgisayar modüllerinin yapısını, davranışlarını ve kullanıcının ihtiyaçları doğrultusunda nasıl bir birleri ile etkileşim halinde bulunacaklarını kapsar. Mimari programcıya görünen kısımları ifade eder.
 - Komut kümesi (instruction set), veri gösterimindeki bit sayısı, I/O mekanizmaları, adresleme teknikleri.
 - Örnek: Çarpma komutu olup olmadığı mimariyle ilgilidir.

9

Ders Kapsamı

- Bilgisayar Organizasyonu: Donanımsal birimlerin beraber bir araya gelerek bir bilgisayar sistemi ne şekilde meydana getirdiğini kapsar. Organizasyon programcıya görünmeyen kısmı ifade eder.
 - Kontrol sinyaller, arayüzler, hafıza teknolojisi.
 - Örnek: Çarpmanın ne şekilde yapıldığı organizasyonla ilgilidir (ardarda toplama veya Booth algoritması)

10

Ders Kapsamı

- **Temel Konular:**
 - Bir bilgisayarın organizasyonu ve tasarımında kullanılan sayısal birimler
 - Temel bir bilgisayarın tasarımı ile ilgili adımlar.
 - Merkezi işlem biriminin organizasyonu ve tasarımı.
 - Giriş/Çıkış ve Bellek mimarilerinin organizasyonu.

11

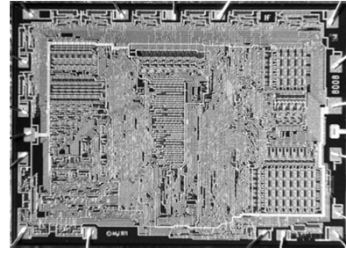
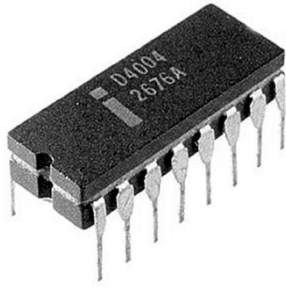
Ders İçeriği

- **Temel Bilgisayar Mimarisi ve Kavramlar**
- **Sayı sistemleri**
- **Kodlama ve Sayısal Devreler**
- **Performans Ölçütleri ve Terimleri**
- **Assembly Komut Yapıları ve Tasarımı**
- **Aritmetik İşlem Birim ve Yapıları**
- **Veriyolu Tasarımı ve Kontrol**
- **Bellek Hiyerarşisi, Ön ve Sanal Bellek**
- **G/Ç ve Arayüz Tasarımı**

12

Bilgisayar Organizasyonu ve Mimarisi

1937 yılında Howard-Aiken ilk otomatik hesap makinesi olan MARK-I 'i, 1943 yılında ise J.P.Erkert ilk işlevsel bilgisayar olan ENIAC(Elektronik Numerical Integrator and Calculator)'ı yaptı. 1971 yılında Intel firması; tüm bileşenleri kendi üzerinde olan 4 bitlik 4004 isimli mikroişlemciyi, 1972 yılında 8 bitlik 8008 işlemcisini, 1974 yılında ise 8080 işlemcisini üretti.



Bilgisayar Mimarileri

8080 işlemcisi, Intel firmasının ilk genel amaçlı mikroişlemcisiydi. Mikroişlemcilerin getirdiği en önemli avantaj programlanabilirlik özelliğiydi. Çünkü bu sayede aynı işlemci üzerinden farklı işlemler yapılabiliyordu.

(Örneğin: Bir gün matematiksel işlemlerin yapıldığı bir işlemci üzerinden, diğer gün alfabetik kelime işlemlerinin yapılabilmesi).

8080 ev bilgisayarlarında da kullanılan ilk işlemcidir. 1979 yılında IBM PC, Intel'in 8088 işlemcisini kullanmaya başladı. 8088 den sonra 80286, 80386, 80486, Pentium, Pentium II, Pentium III, Pentium IV serisi ile kullanıma devam edildi. Günümüz mikroişlemcileri 8088 den çok hızlıdır.

Von Neumann Mimarisi

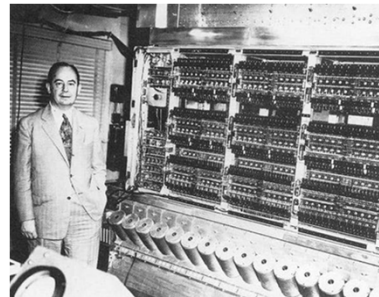
Bugünkü bilgisayarların mimarisinin modeli ENIAC üzerinde çalışmış olan **John Von Neumann** tarafından geliştirilmiştir ve Von Neumann modelinde mantıksal olarak bilgisayar sistemi tam olarak tanımlanmıştır.

Bilgisayar sisteminin; bellek, veriyolu, giriş, çıkış ve merkezi işlem biriminden ibaret olduğu düşünülmüştür.

Von Neumann mimarisine sahip bilgisayarlarda gerçekleştirilen adımlar: program sayacının gösterdiği adresten komut getirilir, program sayacı 1 artırılır, kontrol birimi getirilen kodun komutunu çözer ve tekrar ilk adıma dönlür.

John von Neumann

- ◆ John von Neumann, bilgisayar bilimlerinin öncülerindedir.
- ◆ Bilgisayar organizasyon yapısını 1950lerde öngörmüş ve bunu biçimsel hale getirmiştir.
- ◆ Bu mimari bir dönüm noktası olmuştur.



Von Neumann Mimarisi

- ◆ Bir bilgisayarı oluşturan ana bileşenleri tanımlanmaktadır.
- ◆ Üç ana bileşen öngörülür
 - ◆ "Giriş/Çıkış Birimleri (*Input/Output Devices*)": Kullanıcının komutlar girerek ve sonuçları görerek bilgisayar ile iletişim kurmasını sağlamaktadır.
 - ◆ "Bellek (*Memory*)", bilgisayar tarafından işlenecek bilgileri, programları veya bilgisayarın belirli bir işi yapmasını sağlayacak deyimleri depolamaktadır.
 - ◆ "İşlemci (*Processor*)" veya "Merkezi İşlem Birimi (*Central Processing Unit*)" ise verileri işlemek için önceden programlanmış aşamaları gerçekleştirir.
- ◆ Her üç birim de birbirlerine "veriyolu (*bus*)" adı verilen kablolar ile bağlıdır ve tüm iletişim elektronik sinyaller ile sağlanmaktadır.

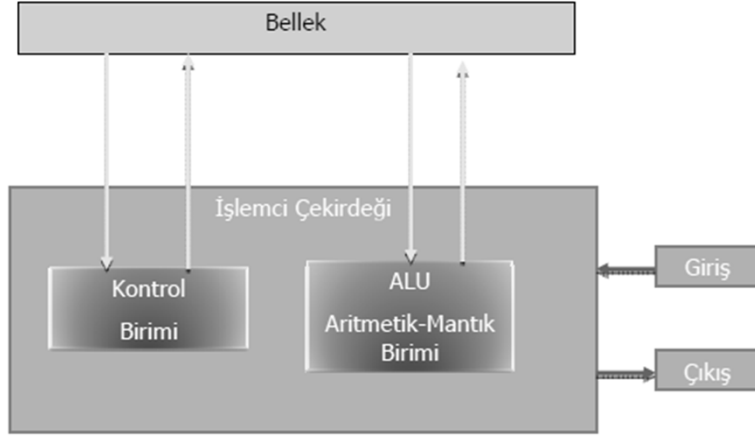
17

Von Neumann Mimarisi

- ◆ Von Neumann mimarisini kullanan tüm bilgisayarlar, "Depolanmış Program Bilgisayarları (*Stored Program Computers*)" olarak isimlendirilmektedir.
- ◆ Bu bilgisayarlar, bellekte depolanmış farklı programları alarak işleme kapasitesine sahiptirler.
- ◆ Bu mimaride, aynı anda 1'den fazla program ve veri belleğe yüklenerek işlenebilmekte, işlemci bu programlar arasında birinden diğerine gidebilmektedir.
- ◆ Günümüzde kullanılan tüm bilgisayarlar bu mimari temel alınarak üretilmiştir.

18

Von Neuman Mimarisi



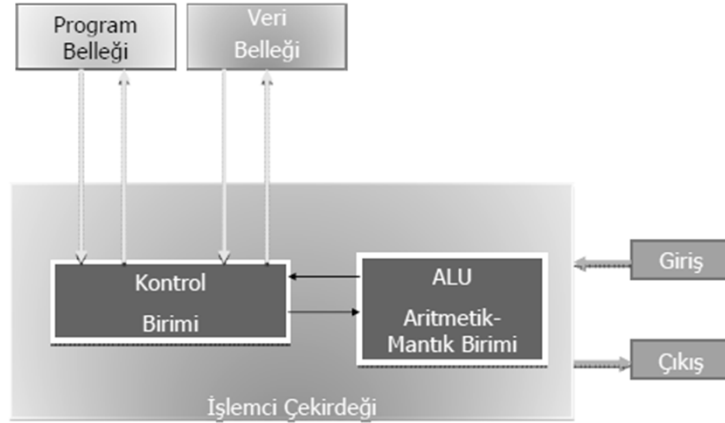
Harvard Mimarisi

Bilgisayar mimarisi tasarımı 2 yaklaşım üzerinde yoğunlaşmıştır.

Bunlardan birisi az önce bahsettiğimiz Von Neuman mimarisi, diğeri ise Harvard mimarisidir.

Harvard mimarili bilgisayar sistemlerinde veri ve buyruklar ayrı belleklerde tutulurlar. Komutla beraber veri farklı iletişim yollarını kullanarak ilgili belleklerden alınıp işlemciye getirilebilir. Getirilen komut işlenip gerekli verisi veri belleğinden alınırken sıradaki komut, komut belleğinden alınıp getirilebilir. Bu da hızı arttıran bir etkidir.

Harvard Mimarisi



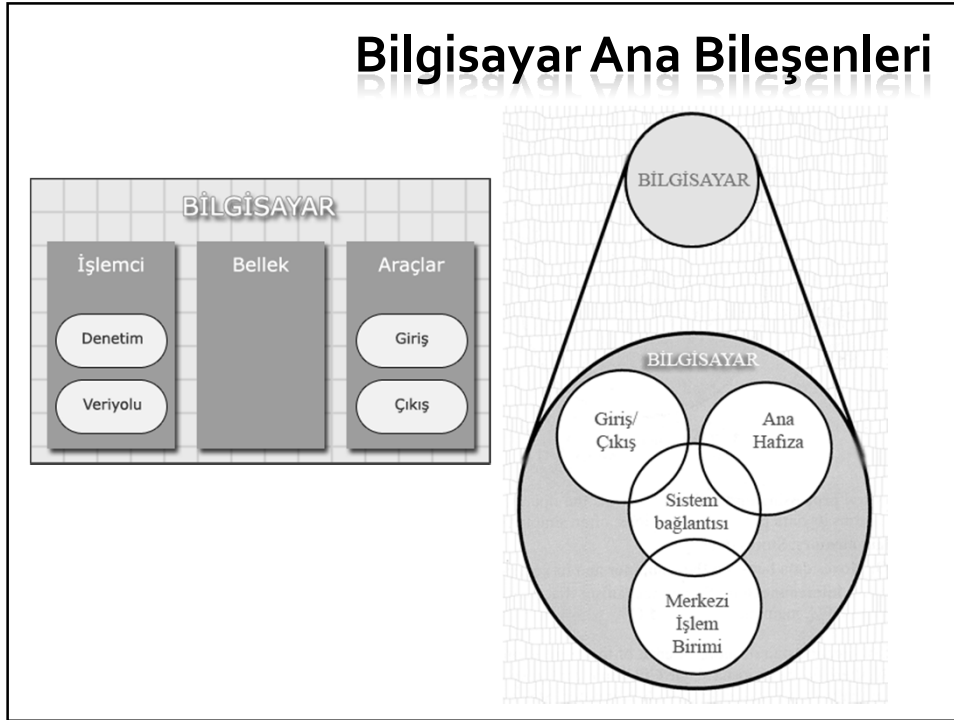
Bilgisayar Ana Bileşenler

Günümüz bilgisayarlarında, ön bellek kullanılarak bellekle tek yoldan iletişim ve buyrukla verinin aynı bellekte bulunma sorunu çözülmüştür. Önbelleğin kapasitesine göre anabellekten veriler ön belleğe alınır. Komut ve veriler önbellek denetleyicisi tarafından ayrılır ve ilgili birimlere yerleştirilir. Önbellek miktarı ne kadar fazla olursa o kadar iyi olur ancak önbelleklerin pahalı olması bir sorundur.

Bilgisayarı oluşturan 4 ana bileşen vardır. Bunlar:

- Veriyolu
- Denetim
- Bellek
- Giriş/Çıkış aygıtları

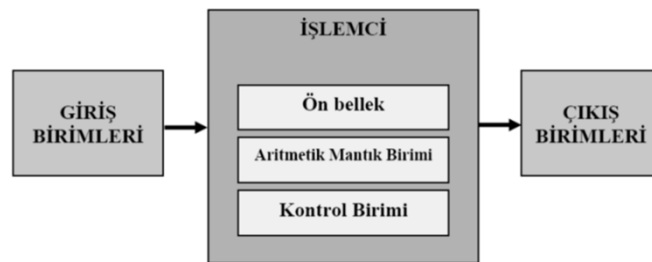
Bilgisayar Ana Bileşenleri



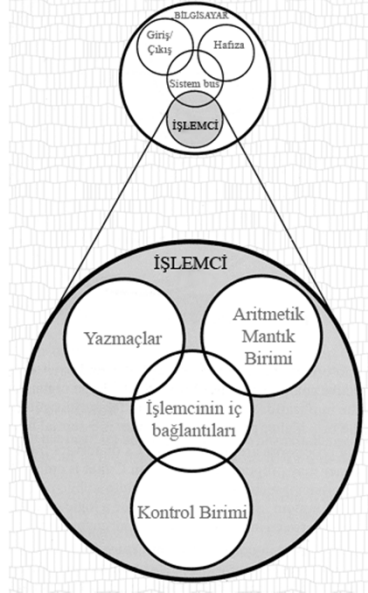
Mikroişlemciler

Mikroişlemciler(Merkezi İşlem Birimi=MİB)

Bilgisayarın beynidir. İşlemcinin görevi, buyrukların bellekten getirilmesi, çözülmesi ve çalıştırılması, sonuçların gözlenmesi, program işlenirken diğer donanım birimlerinden gelen kesme isteklerine cevap vermesi gibi işlemlerdir.

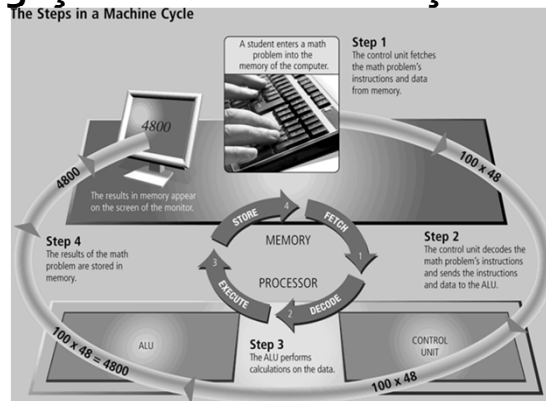


Mikroişlemci Bileşenleri



Makine Çevrimi

- ◆ Bellekteki bir komutun mikroişlemciye getirilerek kodunun çözüldükten sonra yürütülüp bir sonraki komutun getirilmesine kadar geçen zamana makine çevrimi adı verilir.



Makine Çevrimi

- ◆ İşlemci her komutta, bir makine çevrimini oluşturan, dört temel işlemi tekrar eder.
 1. Komut getirme (fetch cycle)
 2. Komut kodu çözme (decode cycle)
 3. Yürütme (execute cycle)
 4. Depolama (store cycle)
- ◆ Çoğu kaynakta Yürütme ve Depolama çevrimleri tek bir çevrim altında birleştirilerek Yürütme Çevrimi olarak isimlendirmiştir.

27

Makine Çevrimi

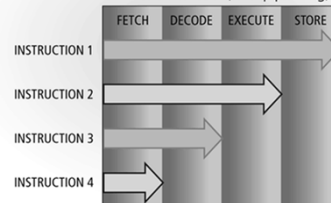
◆Günümüzdeki bilgisayarlar küme komut işlemeyi (pipelining) desteklemektedir.

- ◆İşlemci ilk talimat için makine çevrimi tamamlanmadan önce ikinci talimatı almaya başlar.

MACHINE CYCLE (without pipelining):



MACHINE CYCLE (with pipelining):



28

Komut Seti Mimarisi

- ◆ Bilgisayarın hesaplama karakteristiklerini belirler.
- ◆ Donanım sistemi mimarisi ise, MİB(Merkezi İşlem Birimi), depolama ve giriş-çıkış sistemlerinin dâhil olduğu alt sistem ve bunların bağlantı şeklidir.
- ◆ Programcının bu elemanlara yön verecek programı yazması halinde nasıl bir kabul göreceği ise yine komut kümesi mimarisi tarafından belirlenir.

Komut Seti Mimarisi

- ◆ Komut seti mimarisi, yazılım ile donanım arasındaki iletişimi sağlar.
- ◆ Yazılımdaki komut ne kadar karmaşık olursa, donanım da o kadar karmaşık olur.
- ◆ Bu yüzden komut seti ne çok karmaşık ne de çok yalın olmalıdır.

Komut Seti Mimarisi

- ◆ Programcılar özel bilgisayar sistemleri için, özel donanıma göre kod yazmaktaydılar.
- ◆ Bu yüzden bir makine için yazılan program diğer makinelerde çalışmamaktaydı ve her yeni makine için yeni kodlar yazılmak durumundaydı.
- ◆ IBM firması bu sorunu çözmek için komut kümesi mimarisi(ISA) ve mikrokod motoru denilen bir yöntem geliştirdi.

Programsal Yaklaşım (CISC Mimarisi)

- ◆ CISC (Complex Instruction Set Computer), geleneksel bilgisayar mimarisidir. İşlemci kendi üzerinde bulunan microcode adlı minyatür bir yazılımı kullanarak komut setlerini çalıştırır.
- ◆ Bu sayede komut setleri değişik uzunluklarda olabilir.
- ◆ Bütün adresleme modellerini kullanabilirler.

Programsal Yaklaşım (CISC Mimarisi)

- ◆ Dezavantajı, çalışmak için daha karmaşık bir devre tasarımına ihtiyaç duyulmasıdır.
- ◆ İşlemci üreticileri daha karmaşık (ve güçlü) işlemciler üretmek için sürekli daha büyük komut setleri kullandılar.
- ◆ Günümüz bilgisayarlarında hala mikroprogramlama esasına göre çalışan bilgisayarlar vardır. Intel ve AMD gibi işlemci üreticileri ürünlerinde ISA x86 komut kümesinin tüm özelliklerini yeniliklerle beraber kullanmaktadırlar.

Donanımsal Yaklaşım (RISC Mimarisi)

- ◆ 1974 yılında IBM'den John Cocke bir çipin daha az komutla çalışabilmesi gerektiğini düşündü ve ortaya sadece sınırlı sayıda komut setleri kullanabilen RISC (Reduced Instruction Set Computer) mimarisi çıktı.
- ◆ Komutların uzunluğu sabittir ve bu yüzden de direk olmayan adresleme modu kullanılamaz.
- ◆ Sadece tek bir saat döngüsünde veya daha az sürede çalıştırabilecek komutlar işleme konabilir.

Donanımsal Yaklaşım (RISC Mimarisi)

- ◆ RISC işlemcilerin en büyük avantajları komutları çok çabuk işleyebilmeleridir çünkü bu mimaride komutlar çok basittir.
- ◆ RISC işlemcileri tasarlayıp üretmek daha ucuzdur, çünkü bu basit komutlar için daha az transistör ve daha basit devreler gerekir
- ◆ DEC Alpha, AMD 29k, ARC, ARM, Atmel AVR, , MIPS, PA-RISC, Power PC, Super H, ve Sun Sparc

Bilgisayar Mimarisine Etki Eden Etmenler

- ◆ Teknoloji (Transistör büyüklüğü vs. etkiler)
- ◆ Programlama Dilleri(Hangi dilde yazıldığı vs. etkiler)
- ◆ Uygulama
- ◆ İşletim Sistemleri (İşletim sistemi üzerine yazılan kod ona özgü olduğu için etkiler)
- ◆ Geçmiş(Geçmişte yapılan komutların çalışması zorunlu olduğu için etkiler)

Başarım (Performans) – Saat Hızı

- ◆ Bilgisayarın başarımı sıklıkla saat hızı terimleriyle tanımlanır. (genellikle MHz ya da GHz olarak).
- ◆ Bu CPU'nun ana saatinin saniyedeki döngüleriyle ilgilidir. Ancak, Bu ölçüm bir şekilde yanlış yönlendirse de, yüksek saat hızlı bir makine olarak mutlaka daha yüksek bir performansı olmayabilir.
- ◆ Sonuç olarak AMD gibi yapımcılar saat hızını bir performans göstergesi olarak almamaktadırlar.

Başarım (Performans)

- ◆ Çağdaş CPU'lar birden fazla emri bir saat döngüsünde gerçekleştirirler ve böylece programın hızı oldukça artar.
- ◆ Veri yolu hızları, kullanılabilir bellek ve programlardaki emirlerin tipi ve sırası gibi hızı etkileyen bileşenler de çalışır.

Başarım (Performans) – Latency ve Throughput

- ◆ Latency (gecikme), bir işlemin başlangıcı ve sonuçlanması arasındaki süredir.
- ◆ Throughput ise belli bir zamanda yapılan işin miktarını belirtir.
- ◆ Kesilme gecikmesi sistemin bir elektronik olaya (disk sürücüsünün bilgi aktarımını bitirmesi gibi) vereceği garanti edilmiş en fazla yanıt zamanıdır.
- ◆

Başarım (Performans) – Latency ve Throughput

- ◆ Örneğin, önbellek eklemek gecikmeyi yavaşlatırken, throughput iyileşir.
- ◆ Donanım yöneten bilgisayarlar genellikle düşük kesilme gecikmelerine gerek duyarlar.
 - ◆ Bu bilgisayarlar gerçek zaman diliminde işlem yaparlar ve bu işlem belirlenen sürede bitmezse işlem başarısız olur.

Başarım (Performans) – Diğer Bileşenler

- ◆ Bilgisayarın performansı ,uygulama tanım alanına bağlı olarak, diğer ölçümler kullanılarak da hesaplanabilir.
 - ◆ Sistem CPU bound (sayısal hesaplamalarda)
 - ◆ G/Ç Sınırı (I/O bound) (web sunucusu uygulamalarında)
 - ◆ Bellek Sınırı (Memory bound)(video düzenlemelerde)
- ◆ Güç tüketimi; sunucu, diz üstü bilgisayarlar ve mobil cihazlar gibi taşınabilir cihazlar için önemlidir.

Başarım (Performans) – Ölçütleme

- ◆ Ölçütleme (benchmarking); bilgisayarların bir test program serisini çalıştırırken kullandığı süreyi ölçerek başarımları kıyaslamasını sağlayan işlemler bütünüdür.
- ◆ Ölçütleme gücü yüksek gösterse de bu bilgisayar seçmede yardımcı olmayabilir.
- ◆ Sıklıkla, ölçülen makineler farklı ölçümlere ayrılır.

Başarım (Performans) – Ölçütleme

- ◆ Örneğin bir sistem bilimsel uygulamaları hızlıca idare ederken, diğeri popüler video oyunlarını kolayca oynatabiliyordur.
- ◆ Yazılım veya donanımlara değişik özellikler eklemek isteyen tasarımcılar özel bir ölçüte izin verirler.
- ◆ Böylece hızlı kontrol sağlanır fakat diğerleriyle, çoğunlukla genel işlevlerle benzer avantajlara sahip olmazlar.

Başarım (Performans) – En iyileme

- ◆ En iyilemenin (optimization) genel planı bilgisayarın farklı taraflarındaki değerleri bulmak içindir.
- ◆ Denge içindeki bir bilgisayar sisteminde bilgi hızı bütün bölümler için sabit olacaktır ve değer bunu garantiye almak için eşit olarak dağıtılacaktır.

