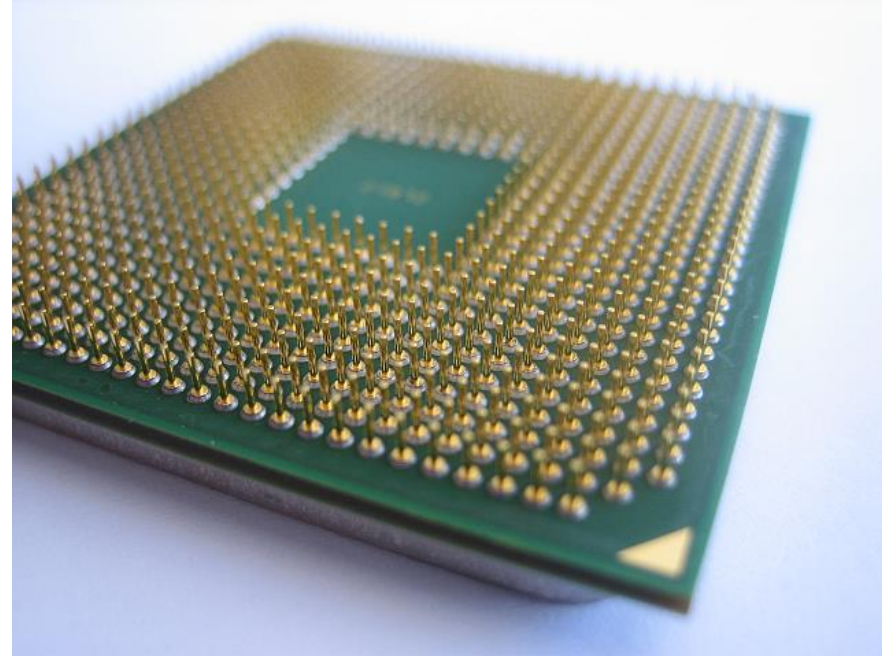


BİLGİSAYAR MİMARİSİ ve ORGANİZASYONU



5. BÖLÜM



1



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, yazılım ile donanım arasındaki iletişimi sağlar.
- ❖ Yazılımdaki komut ne kadar karmaşık olursa, donanım da o kadar karmaşık olur.

Komut Seti Mimarisi



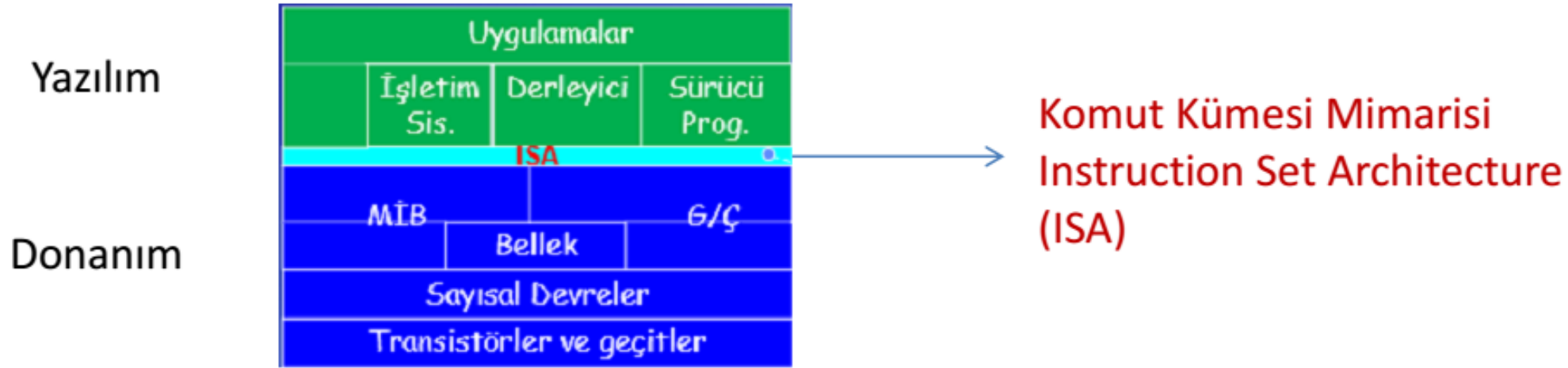
- ❖ Bu yüzden komut seti ne çok karmaşık ne de çok yalın olmalıdır.
- ❖ Programcılar özel bilgisayar sistemleri için, özel donanıma göre kod yazmaktaydılar.
- ❖ Bu yüzden bir makine için yazılan program diğer makinelerde çalışmamaktaydı ve her yeni makine için yeni kodlar yazılmak durumundaydı.
- ❖ IBM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



3



Komut Seti Mimarisi



- ❖ Komut kümesi mimarisi (ISA) bir bilgisayar donanımının, alt düzey programcıya (sistem programlarına) görünen yüzüdür.
- ❖ ISA, makine dili komutları, programcıya görünen saklayıcıları ve işlemcinin doğal veri yapılarını içerir.

Komut Seti Mimarisi



4



- ❖ CISC (Complex Instruction Set Computer) Mimarisi
 - Programsal Yaklaşım
 - Intel, AMD vb...
- ❖ RISC (Reduced Instruction Set Computer) Mimarisi
 - Donanımsal Yaklaşım
 - Power PC, Sun Sparc vb...

Programsal Yaklaşım (CISC Mimarisi)



Amaç:

- ❖ Derleyicilerin işini kolaylaştırmak: Makine dili yüksek düzeyli dillere yaklaşıp.
- ❖ Programların performansını arttırmak: Yetenekli komutlar ile daha kısa programlar yazmak mümkün olur.

5





Programsal Yaklaşım (CISC Mimarisi)

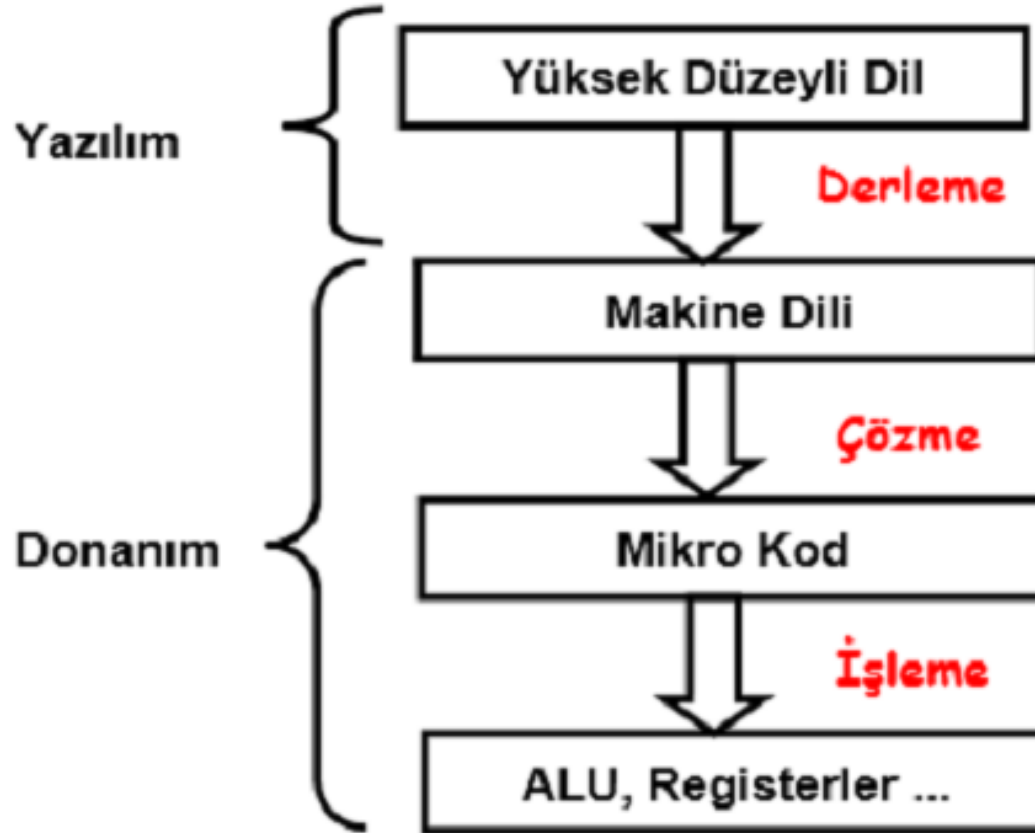
Temel Özellikleri

- ❖ CISC (Complex Instruction Set Computer), geleneksel bilgisayar mimarisidir. İşlemci kendi üzerinde bulunan mikrokod adlı minyatür bir yazılımı kullanarak komut setlerini çalıştırır.
- ❖ En önemli özelliği değişken uzunlukta ve karmaşık yapıda olan komutları barındırmasıdır. Böylelikle bellek tasarımı sağlar.
- ❖ Bütün adresleme modellerini kullanabilirler.
- ❖ Çok sayıda komut bulunur(100-250).

Programsal Yaklaşım (CISC Mimarisi)



7



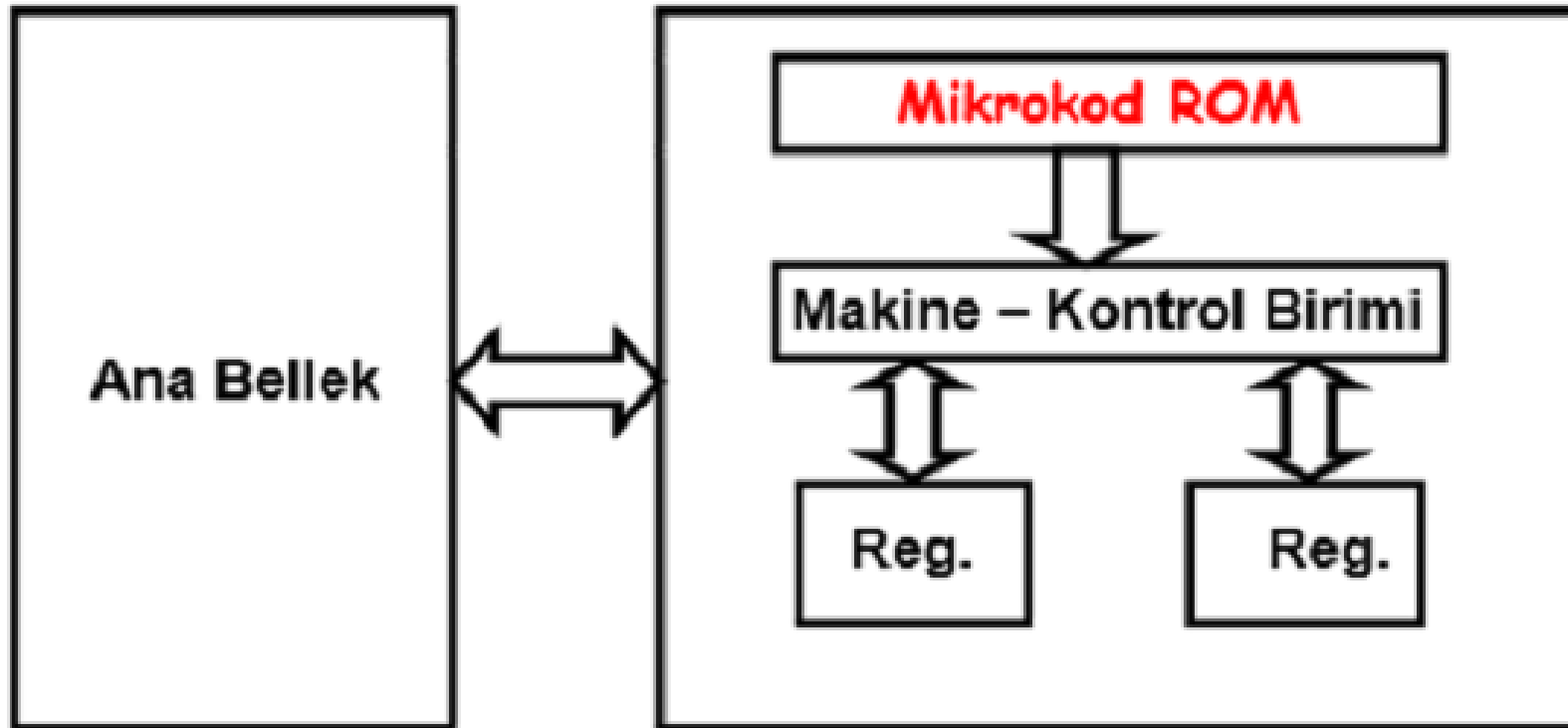
CISC tabanlı bir işlemcinin çalışma biçimi





Mikrokod

İşlemcinin, komut kodlarının her birine karşılık gelen mikrokod komut gruplarını içeren ROM belleği vardır. Bir makine kodu işlemciye eriştiğinde, işlemci kodun daha basit komutlara ayrılmış parçalarını yürütür.



Mikrokod



- ❖ Bu yapıda, komut kümesi aynı kaldığı için programlar, farklı sistemler üzerinde yeniden derlemeye gerek kalmaksızın çalıştırılabilirler.
- ❖ Geliştirilen yeni komutlar eskilerinin üzerine eklenerek geriye doğru olan uyumluluk sağlanır.



Programsal Yaklaşım (CISC Mimarisi)



10



Avantajları:

- ❖ Mikroprogramların yürütülmesi kolaydır.
- ❖ Geriye doğru uyumludur.
- ❖ Mikrokod ROM'A eklenen her bir komut ile CPU daha yetenekli olmaya başlamakta ve verilen bir görevi yürütmek için daha az zaman harcamaktadır.
- ❖ Bu tip mimaride yapılan derleyiciler karmaşık olmak zorunda değildir.



Programsal Yaklaşım (CISC Mimarisi)

Dezavantajları:

- ❖ CPU yapısı her kuşak işlemci ile beraber daha karmaşıklaşmıştır.
- ❖ Farklı uzunluklarda komutlar. Çözmek ve önceden bellekten okumak (prefetch) daha zordur.
- ❖ Özel olarak tasarlamış olan komutlar yeteri kadar sık kullanılmamaktadır. Tipik bir programda mevcut komutların yaklaşık olarak %20'si kullanılmaktadır.
- ❖ İşlemcilerin iç yapıları karmaşıktır.
- ❖ Daha karmaşık bir devre tasarımına ihtiyaç duyulmasıdır.



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.

RISC mimarisi üç temel prensibe dayanmaktadır :

- ❖ Bütün komutlar tek bir çevrimde çalıştırılmalıdır.
- ❖ Belleğe sadece “Load”ve “Store”komutlarıyla erişilmelidir.
- ❖ Bütün icra birimleri mikrokod kullanmadan donanımdan çalıştırılmalıdır.

Donanımsal Yaklaşım (RISC Mimarisi)



Özellikleri:

- ❖ Küçültülen komut kümesi ve azaltılan adresleme modlarına sahiptir.
- ❖ Bütün icra birimleri mikrokod kullanılmadan donanımsal olarak çalışmaktadır.
- ❖ Yüksek seviyeli dilleri desteklemektedir.
- ❖ Çok sayıda kaydediciye sahip olmasıdır.

13



Donanımsal Yaklaşım (RISC Mimarisi)



14



Özellikleri:

- ❖ Komutlar bellek üzerinde işlem yapmazlar, işlemler iç saklayıcılarda yapılır.
- ❖ Belleğe sadece yazma/okuma işlemleri için erişilir (load-store architecture).
- ❖ Tek çevrimde alınıp yürütülebilen komutlar (komut iş hattı (pipeline) sayesinde).
- ❖ Devrelendirilmiş (hardwired) denetim birimi.

Donanımsal Yaklaşım (RISC Mimarisi)



Diğer Özellikleri:

Aşağıdaki özelliklerin bazıları tüm RISC'lerde bulunmayabilir, bazıları ise CISC MİB'lerde de bulunabilir. Ancak bunlar RISC'ler için özellikle önemlidir.

- ❖ Çok sayıda saklayıcı (128-256) (register File)
- ❖ Kesişimli (overlapped register window) saklayıcı penceresi
- ❖ Komutlar için optimize edilebilen iş hattı
- ❖ Harvard mimarisi

Donanımsal Yaklaşım (RISC Mimarisi)



Avantajları:

- ❖ **Hız:** Azaltılmış komut kümesi sayesinde daha hızlı çalışırlar.
- ❖ **Basit donanım:** RISC işlemcilerin komut kümeleri basitleştirildiklerinden çok az yonga kullanırlar.
- ❖ **Kısa tasarım zamanı:** CISC işlemcilere göre daha çabuk tasarlanabilirler.

16



CISC ve RISC İşlemci Örnekleri



17



- ❖ **CISC:** VAX, PDP-11, Intel x86 until Pentium, Motorola 68K...
- ❖ **RISC:** MIPS, SPARC, Alpha, HP-PA, PowerPC, i860, i960, ARM, Atmel AVR...
- ❖ **Karışık (Hybrid)** (Dış kabuğu CISC özellikleri göstermektedir ancak iç çekirdeği RISC yapısındadır: Pentium, AMD Athlon...

Günümüzdeki Durum



- ❖ RISC işlemciler bazı CISC özellikleri tasarımlarına katarken bazı CISC işlemciler de RISC özellikleri içermektedir.
- ❖ Sonuç olarak güncel bazı RISC tasarımları, örneğin PowerPC, artık "saf" bir RISC değildir. Benzer şekilde bazı CISC işlemciler de, örneğin Pentium II ve sonrası, "saf" CISC tasarımlar değildir ve RISC işlemcilerin bazı özelliklerini taşımaktadırlar.



RISC VE CISC MİMARİ KARŞILAŞTIRMASI



RISC ve CISC işlemciler birbirleri ile hızlarına, komut işleme tekniklerine, kullanılan transistör sayılarına, vb. kriterlere göre karşılaştırılabilirler.

19



RISC VE CISC MİMARİ KARŞILAŞTIRMASI



Hız: İki işlemci mimarisinin karşılaştırılmasından ilk önemli farkın; hızları olduğu bulunur.

İki işlemci mimarisi arasındaki hız farkı, kullanılan komut işleme teknikleri sonucu oluşur. RISC işlemciler, genellikle aynı saat frekansında çalışan CISC işlemcilere göre daha hızlıdır.

20



RISC VE CISC MİMARİ KARŞILAŞTIRMASI



Komut işleme tekniği: Mimariler arasındaki ikinci önemli fark; komut işleme tekniğidir.

- ❖ CISC işlemcilerde ‘kademeli komut işleme’ tekniği kullanılırken, RISC işlemcilerde ‘kanal komut işleme tekniği’ (pipeline) kullanılır. CISC tekniği ile aynı anda tek bir komut işlenebildiği ve komutun, işlenmesi bitmeden yeni bir komut üzerinde çalışmaya başlanamaz.
- ❖ RISC tekniğinde ise, aynı anda çok sayıda komut işlenmektedir. Komutların birbirini takip etmesi nedeni ile her bir komut bir birim uzunluktadır ve her işlem adımında bir komuta ait işlemler bitirilir.



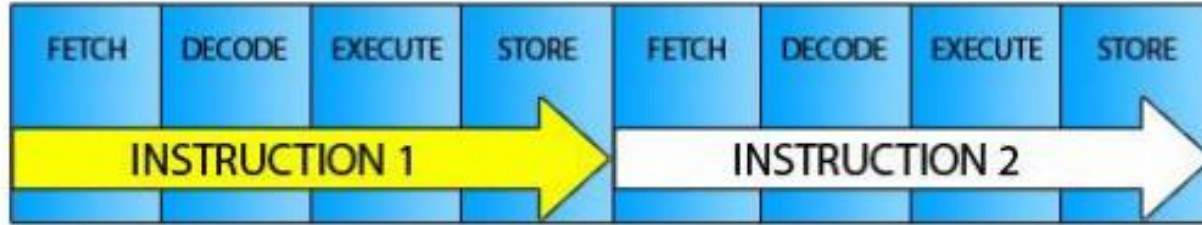
22



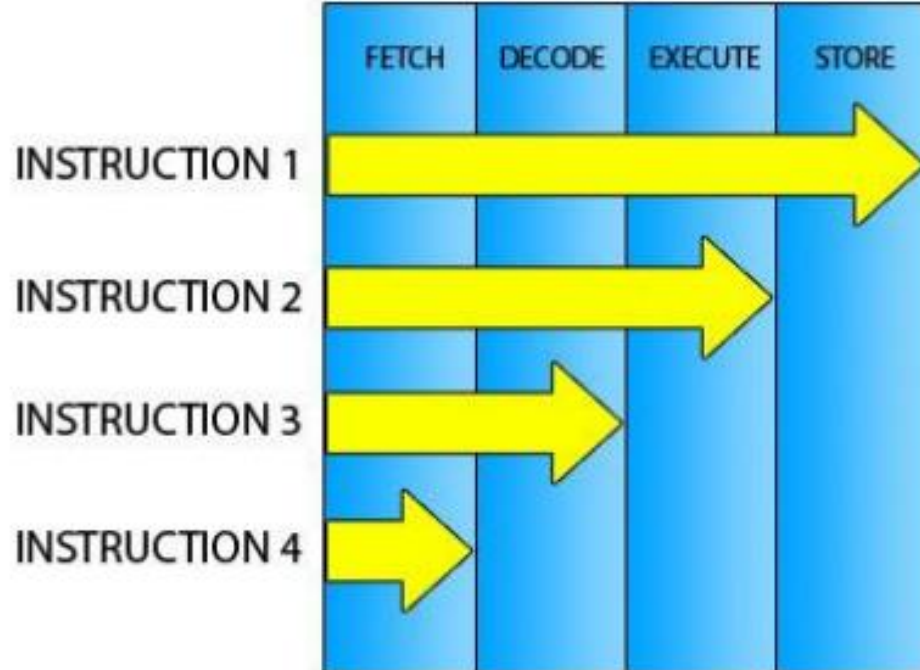
5. Bölüm

RISC VE CISC MİMARİ KARŞILAŞTIRMASI

MACHINE CYCLE (without pipeline):



MACHINE CYCLE (with pipeline):



❖ Günümüzdeki
bilgisayarlar
küme komut
işlemeyi
(pipelining)
desteklemektedir.

RISC VE CISC MİMARİ KARŞILAŞTIRMASI



23



Transistör sayısı: CISC ve RISC yapıları arasındaki üçüncü önemli fark; işlemcilerde kullanılan transistör sayısıdır. CISC işlemcilerde kullanılan transistör sayısı, RISC işlemcilere göre daha fazladır.

Daha fazla sayıda transistör kullanılması, daha geniş alan gereksinimi ve daha fazla ısı ortaya çıkarır, Oluşan daha fazla ısı nedeniyle soğutma ihtiyacı ortaya çıkar ve soğutma işlemi, ısı dağıtıcısı veya fanlar kullanılarak gerçekleştirilir.

RISC VE CISC MİMARİ KARŞILAŞTIRMASI



24



Donanımsal yapı (Tasarım şekli): İki mimari arasındaki bir diğer fark; donanımsal yapıları ve tasarım şekilleridir. RISC işlemciler , CISC işlemcilere göre daha basit yapıda olduklarından daha kolay tasarlanırlar.

Komut yapısı: RISC mimarisi, CISC'in güçlü komutlarından yoksundur ve aynı işlemi yapmak için daha fazla komuta gereksinim duyar. RISC mimaride aynı uzunlukta basit komutlar kullanılırken CISC mimaride karmaşık yapıda değişken uzunlukta komutlar kullanılır.

RISC VE CISC MİMARİ KARŞILAŞTIRMASI



25



RISC

(Hard-wired Control Unit)

Hızlı

Ucuz

Yeniden dizayn zor

Daha az komut (instruction)

Daha fazla saklayıcı bellek

CISC

(Microprogrammed Control Unit)

Nispeten yavaş

Pahalı

Esnek

Daha fazla komut (instruction)

Daha az saklayıcı bellek

Bilgisayar Mimarisine Etki Eden Etmenler



26



- ❖ 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.
- ❖ 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



28



Hızın iki ana tipi vardır:

- ❖ 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.



30



Başarım (Performans) –Bileşenleri

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

Başarım (Performans) – Ölçütleme



SON



- ❖ Ölçütleme (benchmarking); bilgisayarların bir test program serisini çalıştırırken kullandığı süreyi ölçerek başarımların kıyaslanması yapılmasını sağlayan işlemler bütünüdür.
- ❖ Sıklıkla, ölçülen makineler farklı ölçümlere ayrılır.
- ❖ Örneğin bir sistem bilimsel uygulamaları hızlıca idare ederken, diğeri popüler video oyunlarını kolayca oynatabiliyordur.
- ❖ Böylece hızlı kontrol sağlanır fakat diğerleriyle, çoğunlukla genel işlevlerle benzer avantajlara sahip olmazlar.