

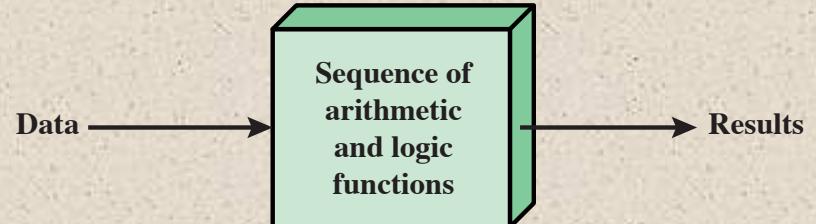
⁺ Chapter 3

Bilgisayar Bileşenleri ve Ara Bağlantılarına
Üst Seviye Bakış

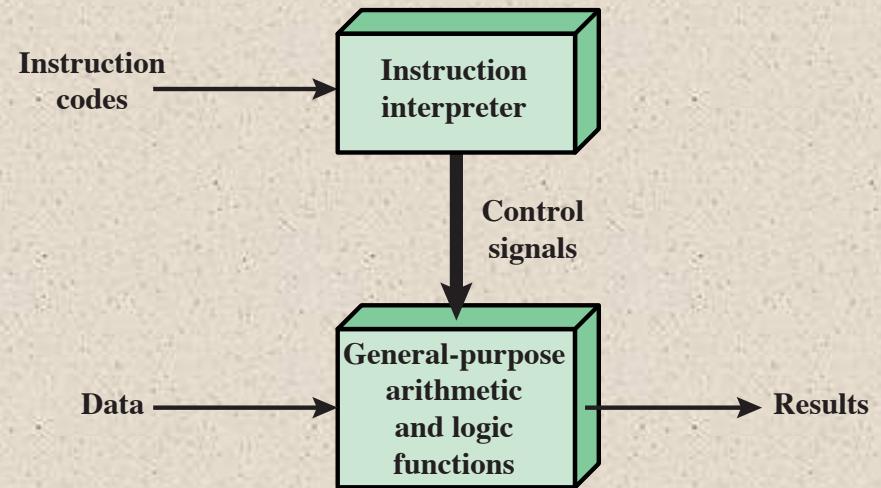
Bilgisayarın Bileşenleri

- Çağdaş bilgisayarların tasarıımı, John von Neumann'un Princeton'daki İleri Araştırmalar Enstitüsü'nde geliştirdiği kavramlara dayanmaktadır.
- *von Neumann mimarisi* olarak anılan bu tasarım aşağıda belirtilen üç temel kavrama dayanmaktadır:
 - Veri ve komutlar tek bir okuma-yazma belleğinde saklanır,
 - Bu belleğin içeriği, içерdiği verilerin türüne bakılmaksızın konuma göre adreslenir,
 - Koşturma (execution) işlemi bir komuttan diğerine sıralı bir şekilde gerçekleşir (açıkça değiştirilmemiş sürece).
- *Donanımsal Olarak Programlama (Hardwired program)*
 - Donanımsal mantıksal bileşenler yapılandırılarak belli bir veri kümesi üzerinde bir kısım aritmetik ve mantıksal işlemleri gerçekleştirmediir.
 - Çeşitli bileşenleri istenen yapılandırmaya göre birleştirme sürecinin sonucu!

Donanım ve Yazılım Yaklaşımları



(a) Programming in hardware



(b) Programming in software

Figure 3.1 Hardware and Software Approaches

Yazılım

- Bir dizi kod veya komut,
- Donanımın bir kısmı her bir komutu yorumlar ve kontrol sinyalleri üretir,
- Donanımı tekrar bağlamak yerine her yeni program için yeni kod dizisi sağlar.

Temel Bileşenler:

- CPU
 - Komutları yorumlar,
 - Verileri işler,
 - Genel amaçlı aritmetik ve mantıksal işlemler modülüdür.
- I/O Bileşenleri
 - Girdi modülü
 - Verileri ve talimatları kabul etmek ve bunları sistem tarafından kullanılabilen dahili bir sinyal biçimine dönüştürmek için temel bileşenler içerir.
 - Çıktı modülü
 - Sonuç raporlama aracı

Yazılım

I/O Bileşenleri



BELLEK

Memory address register (MAR)

- Bir sonraki okuma veya yazma işlemi için bellekteki adresi belirtir.

MAR

Memory buffer register (MBR)

- Bellekte yazılabilecek verileri içerir veya bellekten okunan verileri alır.

MBR

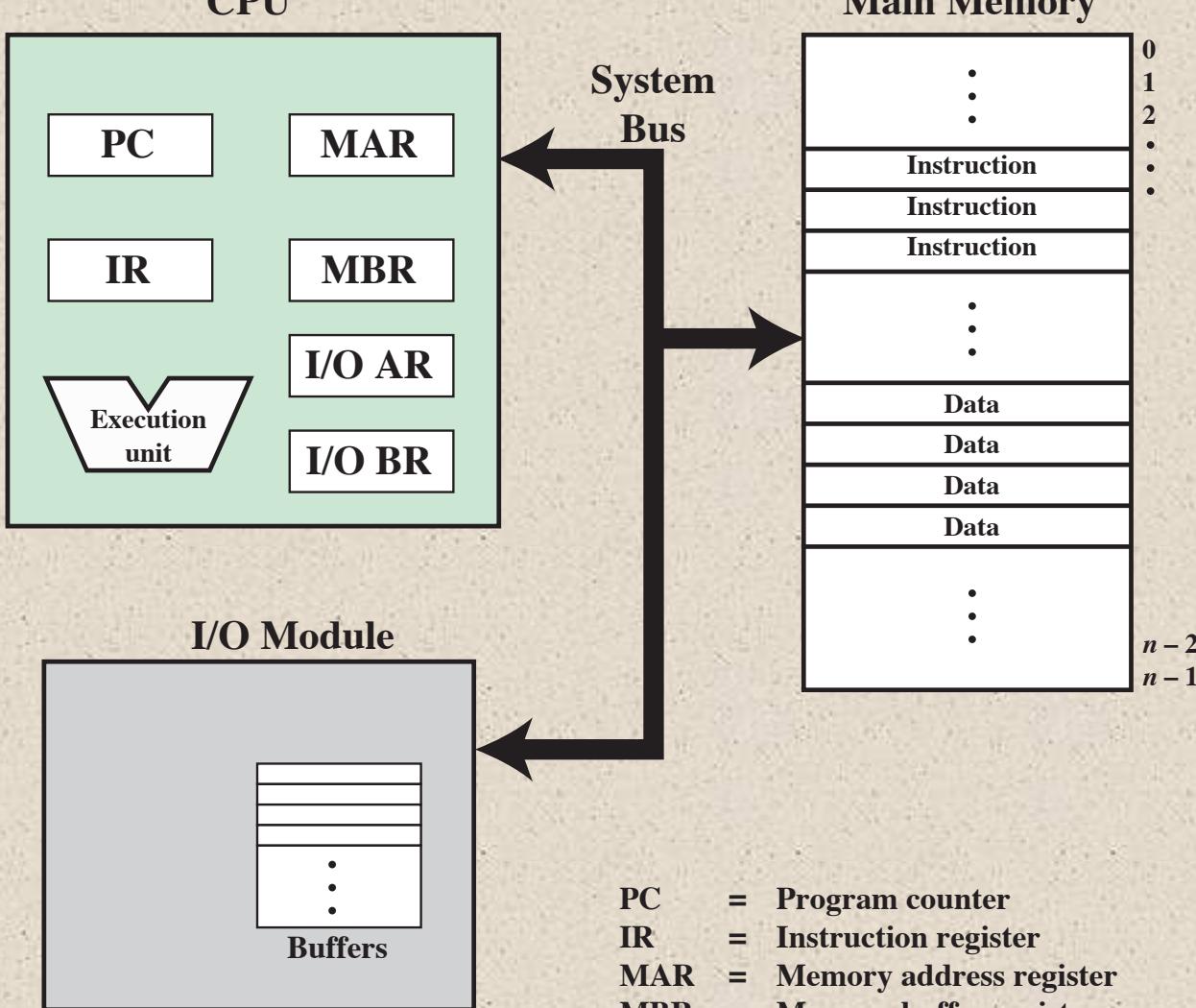
I/O address register (I/OAR)

- Belirli bir I/O aygıtını belirtir.

I/O buffer register (I/OBR)

- Bir I/O modülü ve CPU arasındaki veri alışverişi için kullanılır.

Bilgisayar Bileşenleri: Üst Seviye Bakış



- | | |
|---------------|---------------------------------|
| PC | = Program counter |
| IR | = Instruction register |
| MAR | = Memory address register |
| MBR | = Memory buffer register |
| I/O AR | = Input/output address register |
| I/O BR | = Input/output buffer register |

Figure 3.2 Computer Components: Top-Level View

+

Temel Komut Çevrimi

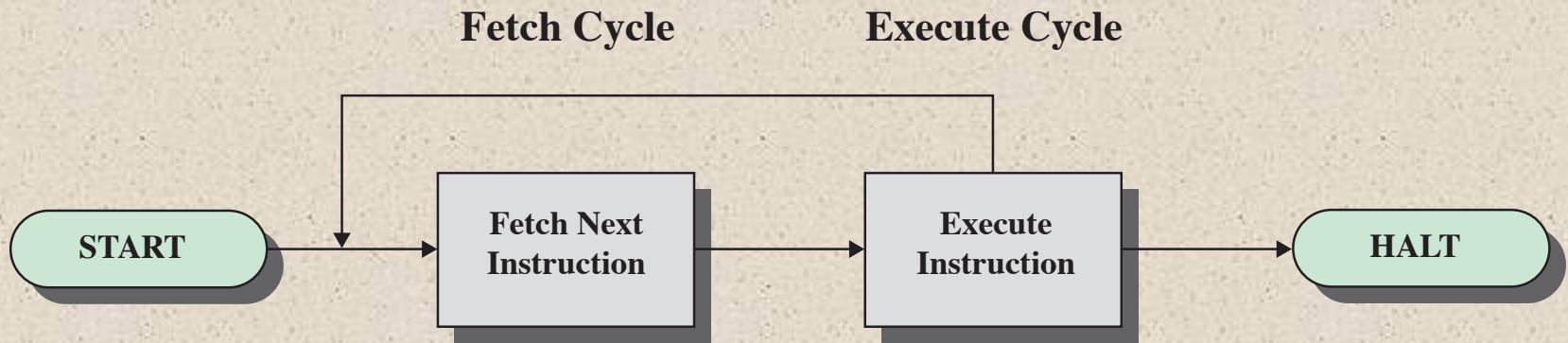
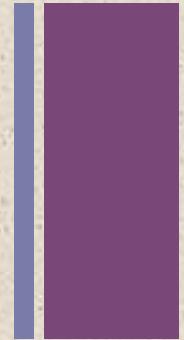


Figure 3.3 Basic Instruction Cycle

Al-Getir Çevrimi

- Her komut çevriminin başında, işlemci bellekten bir talimatı alıp getirir.
- Program sayacı (PC), daha sonra alıp getirilecek komutun adresini tutar.
- Her komut alıp getirme işleminden sonra, işlemci PC'nin değerini arttırır böylece sıradaki yani bir sonraki komutu alıp getirir.
- Alıp getirilen komut instruction register (IR)'a yüklenir.
- İşlemci komutu yorumlar ve gereken eylemleri yerine getirir.



Eylem Kategorileri

- Veri işlemciden belleğe ya da bellekten işlemciye transfer edilir.

İşlemci-Bellek

- Veri işlemci ve I/O modülü arasında transfer edilerek çevresel bir aygıta ya da çevresel bir aygıtın transfer edilir

İşlemci-I/O

- Komut yürütmenin sırasının değiştirildiğini belirtebilir.

Kontrol

Veri işleme

- İşlemci, veri üzerinde bazı aritmetik ve mantıksal işlemleri gerçekleştirir.



(a) Instruction format



(b) Integer format

Program Counter (PC) = Address of instruction

Instruction Register (IR) = Instruction being executed

Accumulator (AC) = Temporary storage

(c) Internal CPU registers

0001 = Load AC from Memory

0010 = Store AC to Memory

0101 = Add to AC from Memory

(d) Partial list of opcodes

Figure 3.4 Characteristics of a Hypothetical Machine



Program Koşma Örneği

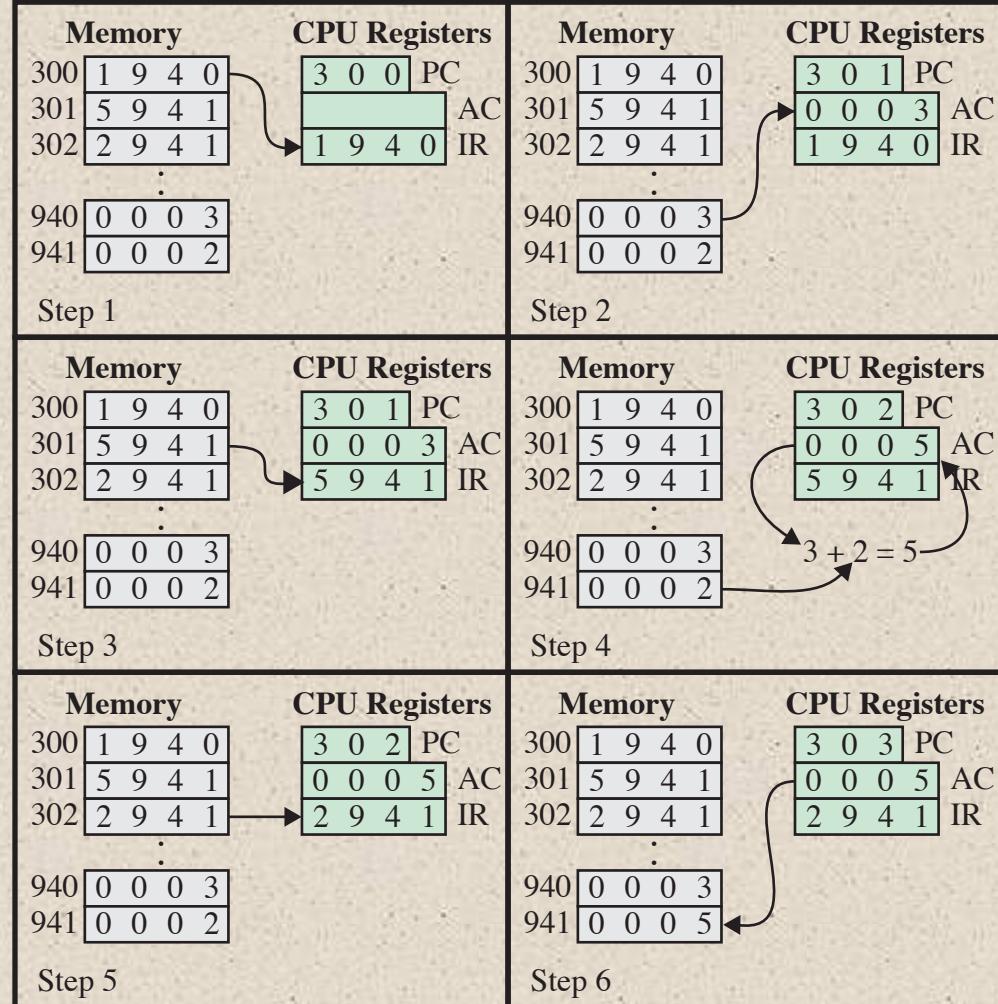


Figure 3.5 Example of Program Execution
(contents of memory and registers in hexadecimal)

Komut Çevrimi Durum Diyagramı

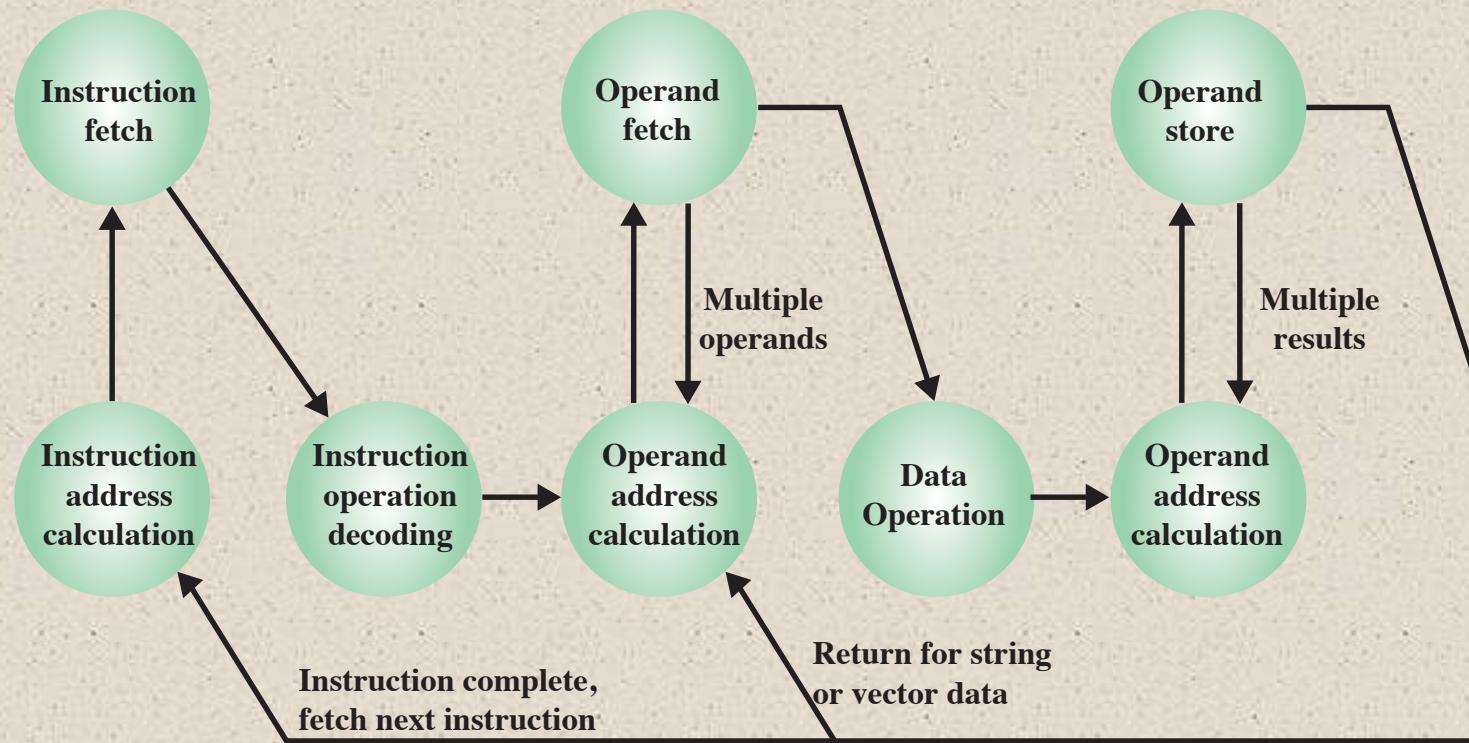
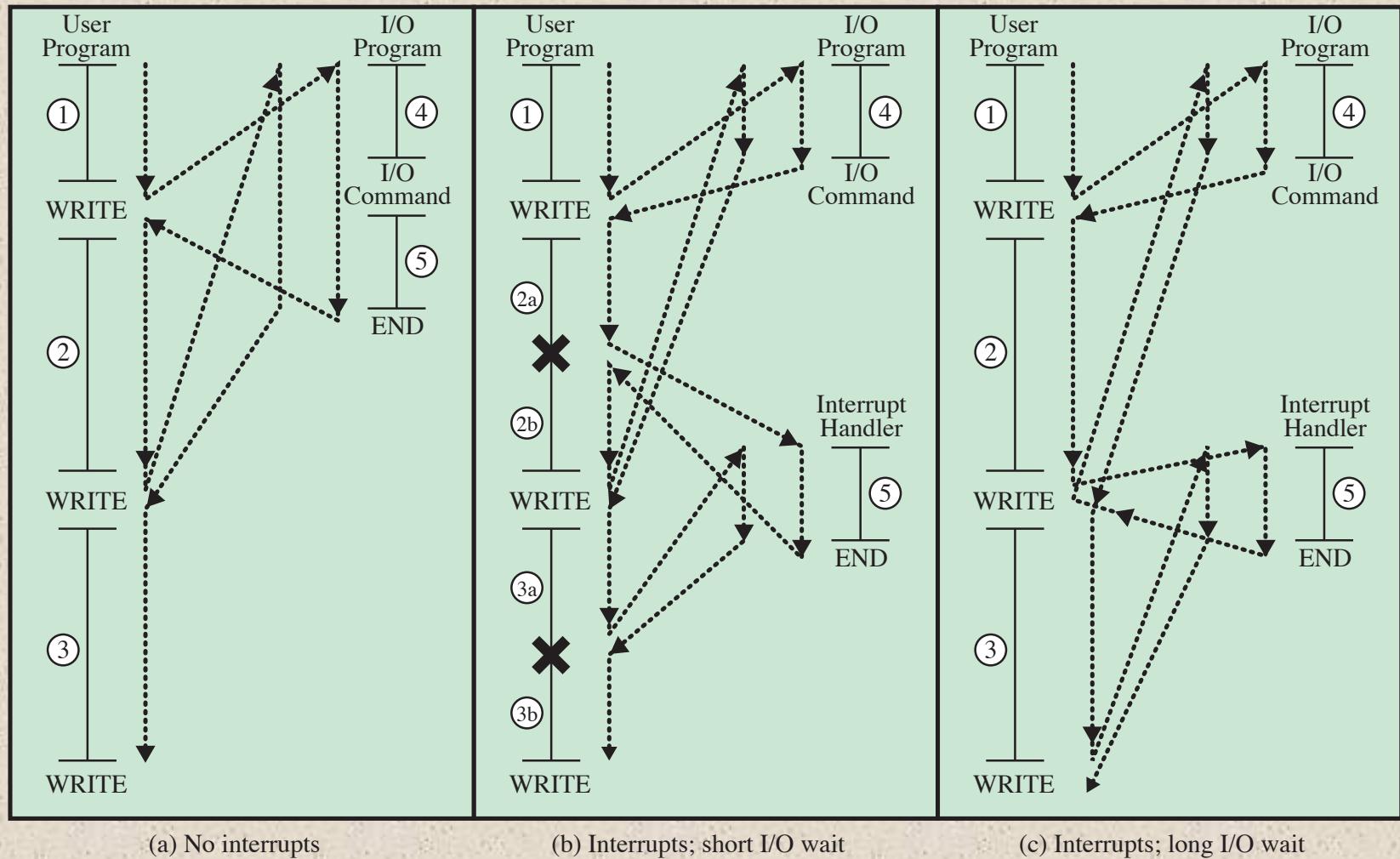


Figure 3.6 Instruction Cycle State Diagram

Kesme Çeşitleri

Program	Generated by some condition that occurs as a result of an instruction execution, such as arithmetic overflow, division by zero, attempt to execute an illegal machine instruction, or reference outside a user's allowed memory space.
Timer	Generated by a timer within the processor. This allows the operating system to perform certain functions on a regular basis.
I/O	Generated by an I/O controller, to signal normal completion of an operation, request service from the processor, or to signal a variety of error conditions.
Hardware failure	Generated by a failure such as power failure or memory parity error.

Program Akış Kontrolü



✗ = interrupt occurs during course of execution of user program

Figure 3.7 Program Flow of Control Without and With Interrupts

Kontrolün Kesmeler Aracılığı ile Transferi

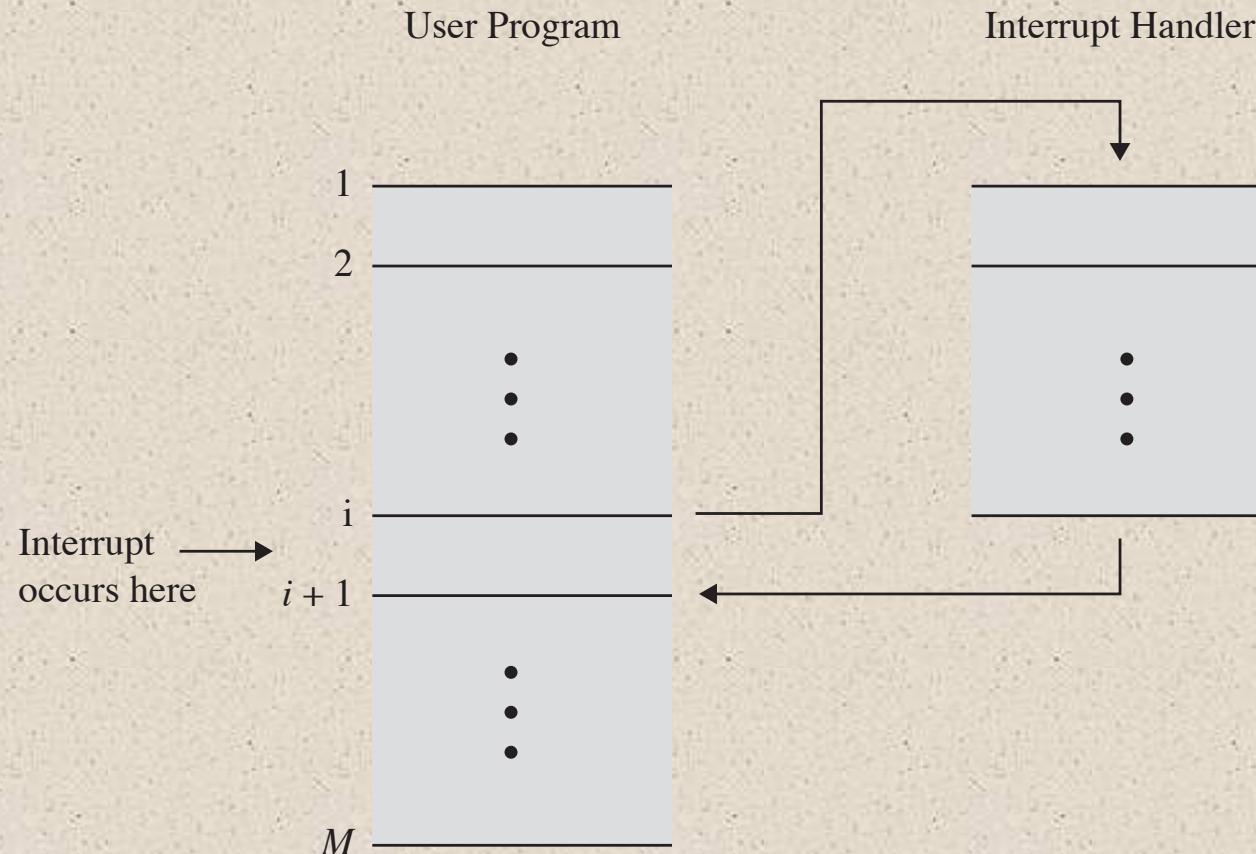


Figure 3.8 Transfer of Control via Interrupts

Kesmelerle Birlikte Komut Çevrimi

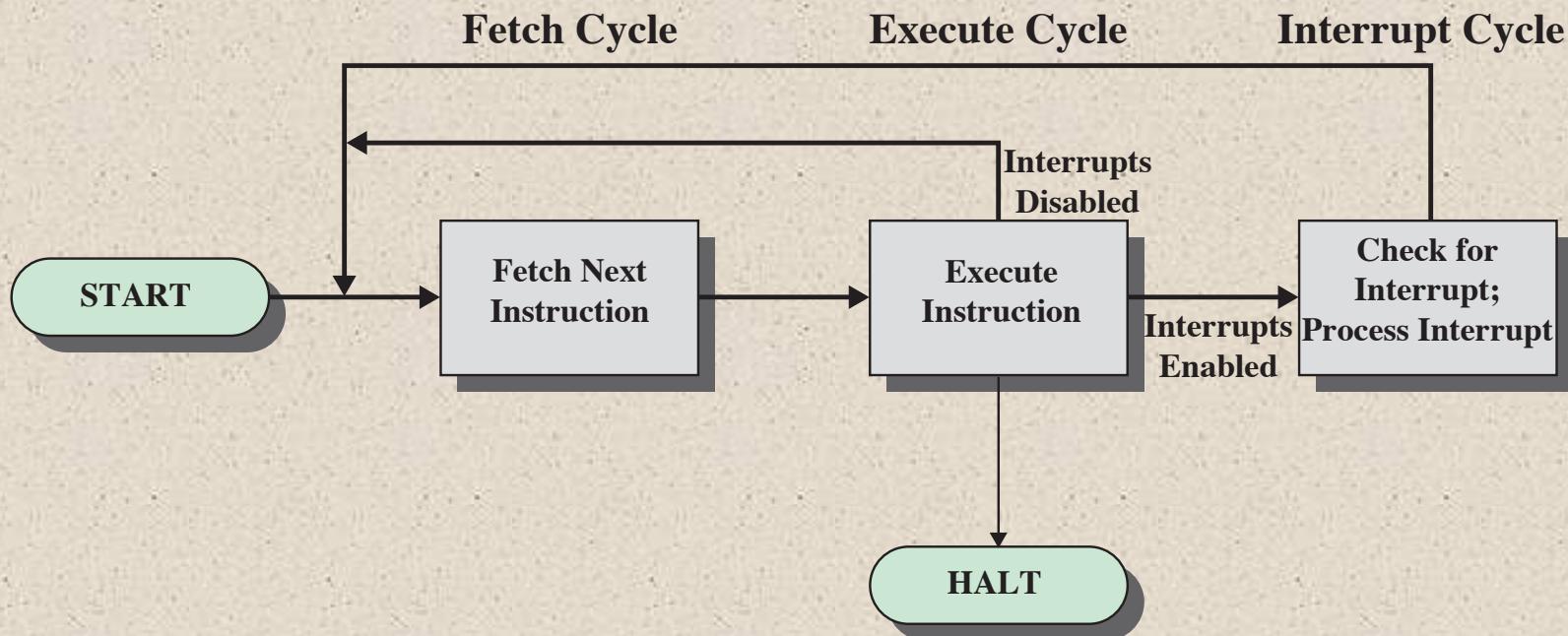


Figure 3.9 Instruction Cycle with Interrupts

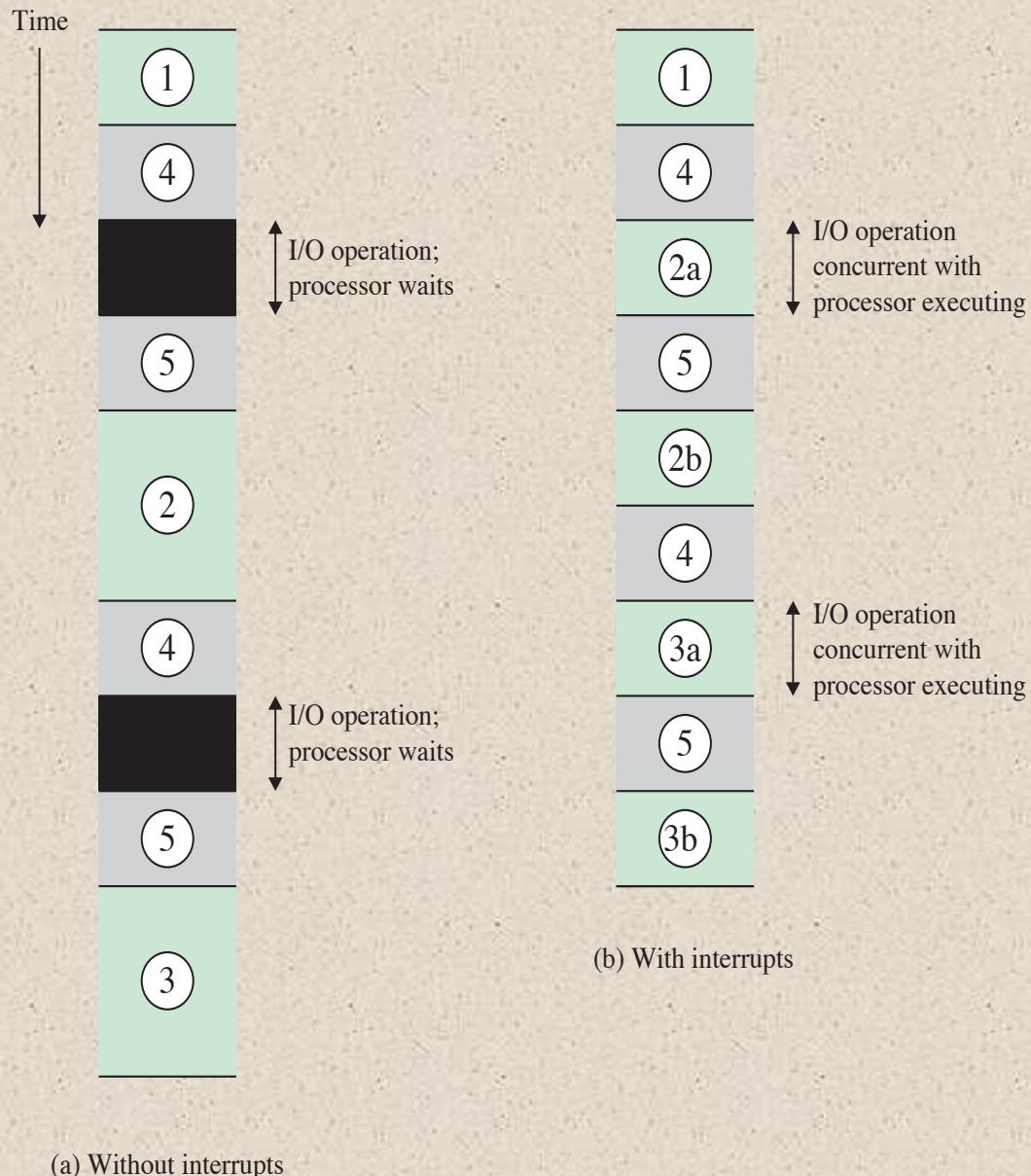
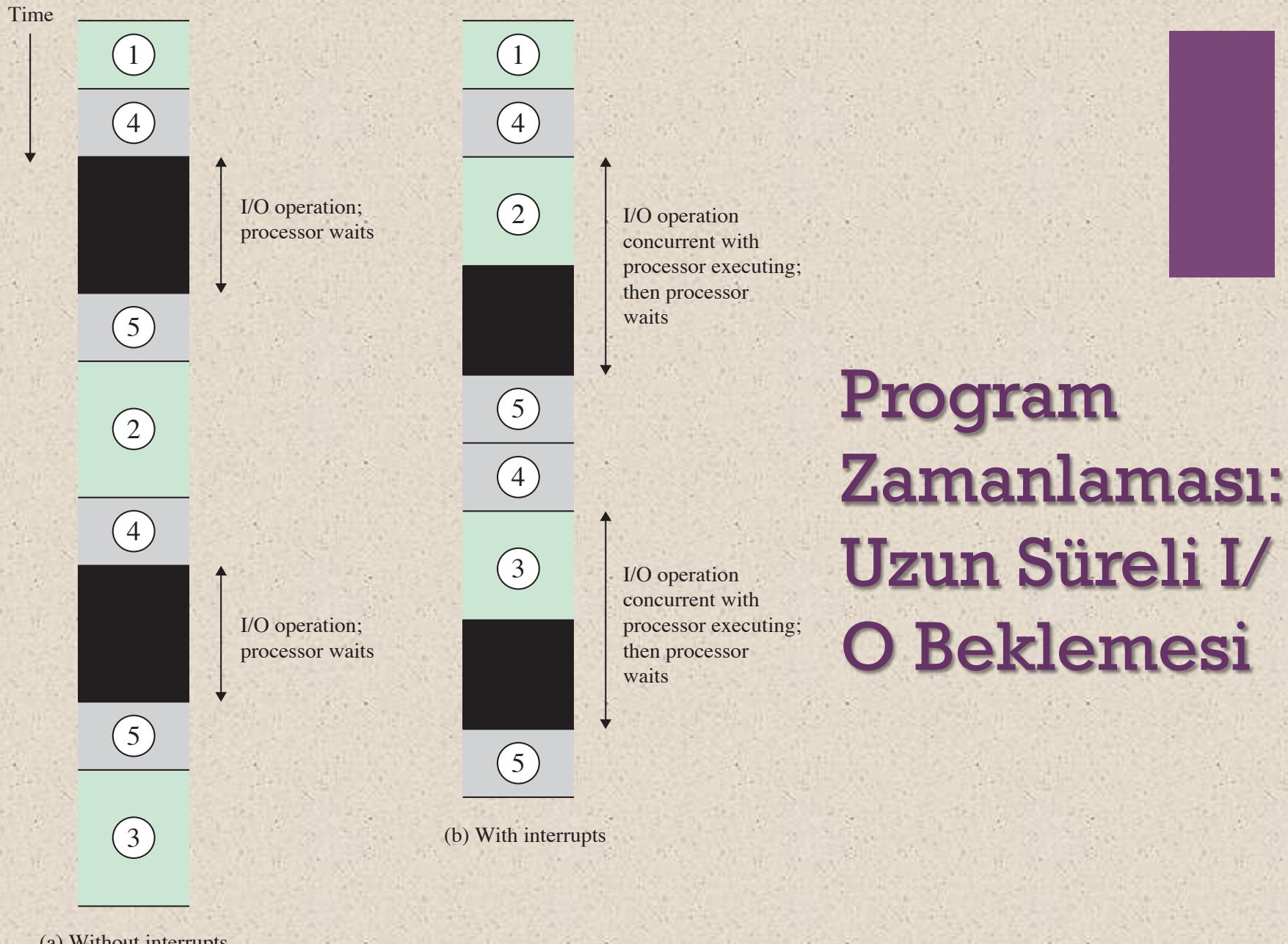


Figure 3.10 Program Timing: Short I/O Wait

Program Zamanlaması: Kısa Süreli I/O Beklemesi



Program Zamanlaması: Uzun Süreli İ/ O Beklemesi

Figure 3.11 Program Timing: Long I/O Wait

Kesmelerle Birlikte Komut Çevrimi Durum Diyagramı

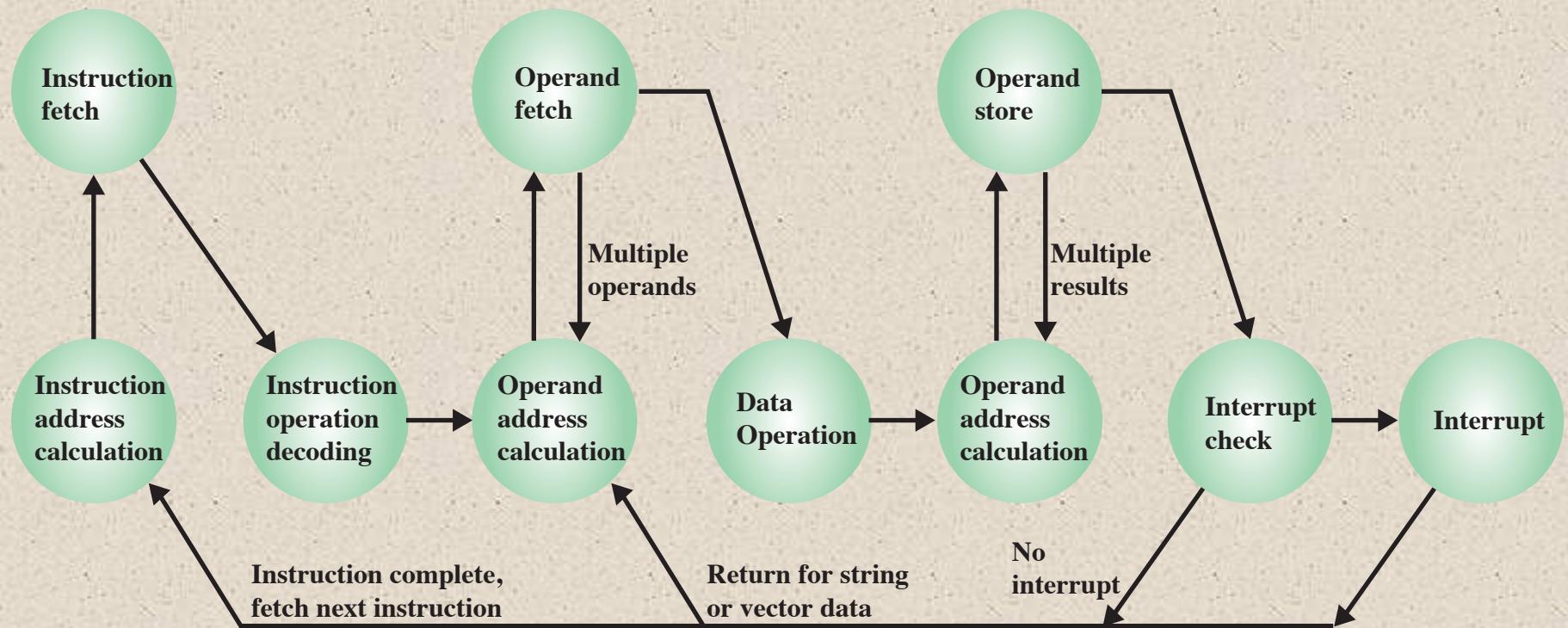
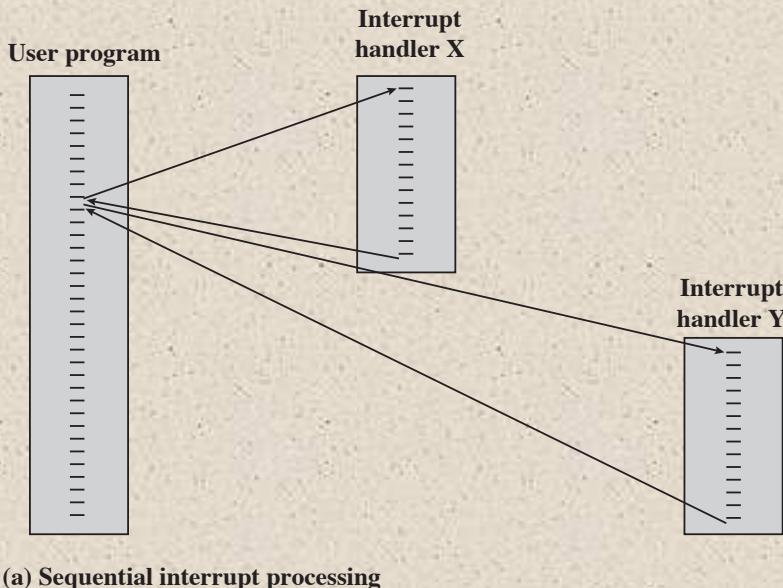


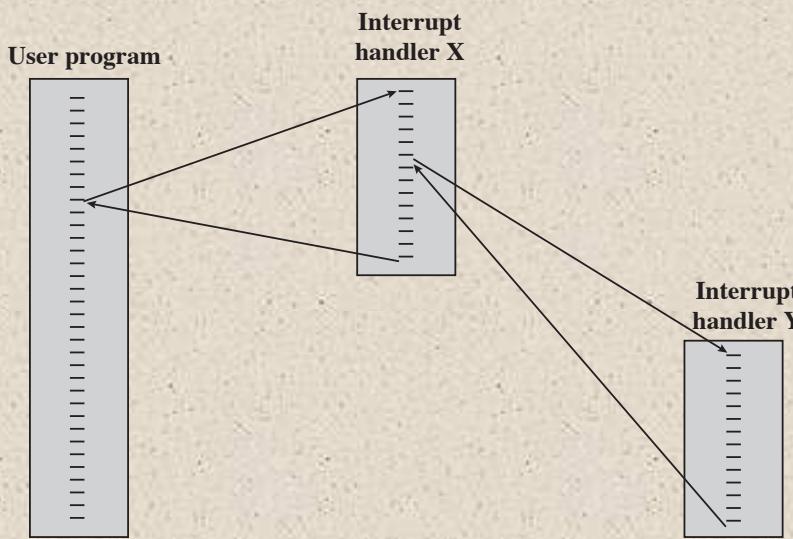
Figure 3.12 Instruction Cycle State Diagram, With Interrupts

Kontrolün Transferi

Çoklu Kesmeler



(a) Sequential interrupt processing



(b) Nested interrupt processing

Figure 3.13 Transfer of Control with Multiple Interrupts

Çoklu Kesmelerin Zaman Sıralaması

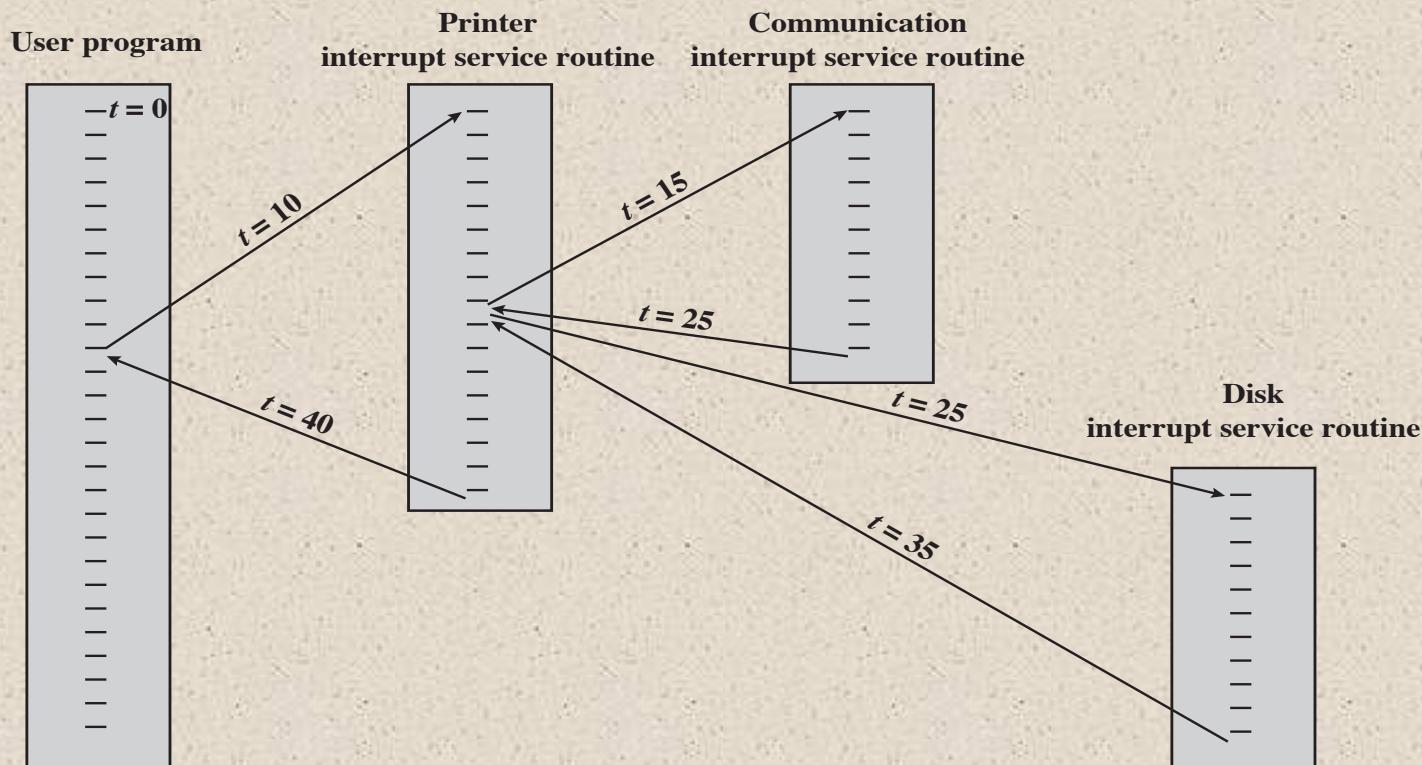


Figure 3.14 Example Time Sequence of Multiple Interrupts

I/O Fonksiyonu

- I/O modülü işlemci ile doğrudan veri alışverişi yapabilir.
- İşlemci I/O modülünden veriyi okuyabilir veya I/O modülüne veriyi yazabilir:
 - İşlemci belirli bir I/O modülü tarafından kontrol edilen belirli bir aygıtı tanımlar
 - Bellek başvuru komutları yerine I/O komutları
- Bazı durumlarda, I/O değişimlerinin doğrudan bellekle gerçekleşmesine izin vermek istenir:
 - İşlemci bir I/O modülüne, I/O'dan belleğe aktarımın işlemciye bağlı olmaksızın gerçekleşebilmesi için bellekten okuma veya belleğe yazma yetkisini verir.
 - I/O modülü okuma ve yazma komutlarını belleğe bildirerek işlemciyi değiştirme sorumluluğundan kurtarır.
 - Bu işlem Doğrudan Bellek Erişimi olarak bilinir (DMA).

Bilgisayar Modülleri

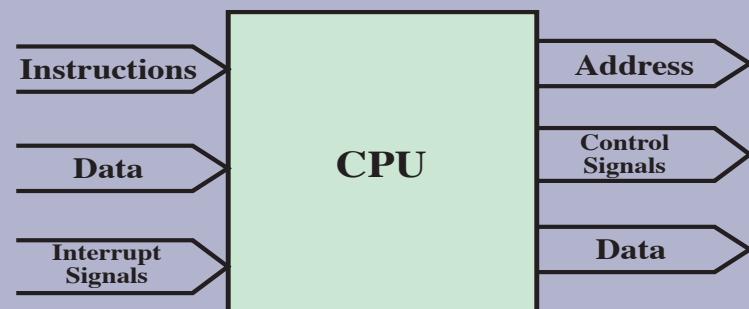
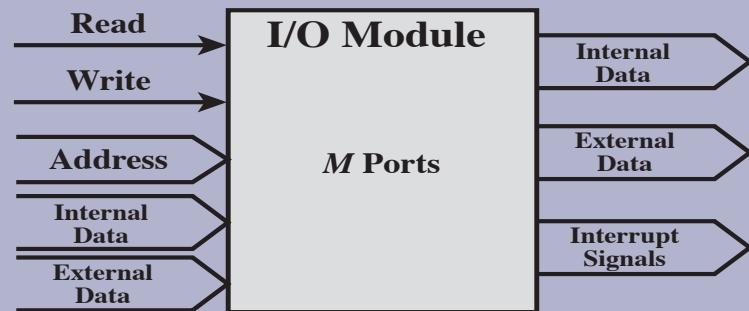
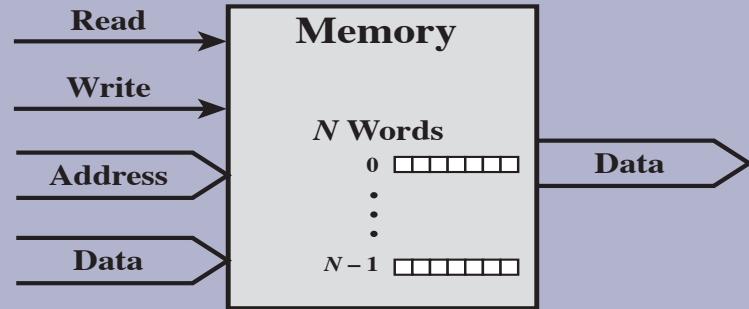


Figure 3.15 Computer Modules

Ara bağlantı yapısı aşağıdaki transferleri desteklemelidir:

Bellekten
İşlemciye

İşlemci bir komutu veya bir birimlik bilgiyi bellekten okur.

İşlemciden
Belleğe

İşlemci belleğe bir birimlik veri yazar.

I/O'dan
İşlemciye

İşlemci bir I/O modülü aracılığı ile I/O'dan veriyi okur.

İşlemciden
I/O'ya

İşlemci I/O aygıtına veri gönderir.

I/O'dan
Belleğe
veya
Bellekten I/O'ya

Bir I/O modülü veriyi doğrudan bellek ile değiştirme yetkisine sahiptir. Bunu DMA'yı kullanarak yapar dolayısıyla işlemciye gitmesine gerek yoktur.

Ortak Veri Yolu Ara Bağlantısı

İki veya daha fazla cihazı bağlayan iletişim veri yolu:

- Temel özelliği paylaşılan bir iletim ortamı olmalıdır.

Herhangi bir cihaz tarafından iletilen sinyaller, veri yoluna bağlı diğer tüm cihazlar tarafından alınabilir:

- İki cihaz aynı zaman aralığında sinyal gönderirse sinyalleri çakışacak ve bozulma olacaktır.



Tipik olarak çoklu iletişim hatlarından oluşur:

- Her hat, 1'i ve 0'ı (ikilik sayı sisteminde) temsil eden sinyalleri yaymayıabilir.

Bilgisayar sistemleri, bilgisayar sistemi hiyerarşisinin çeşitli seviyelerindeki bileşenler arasında yol sağlayan belirli sayıda farklı veri yolları içerir.



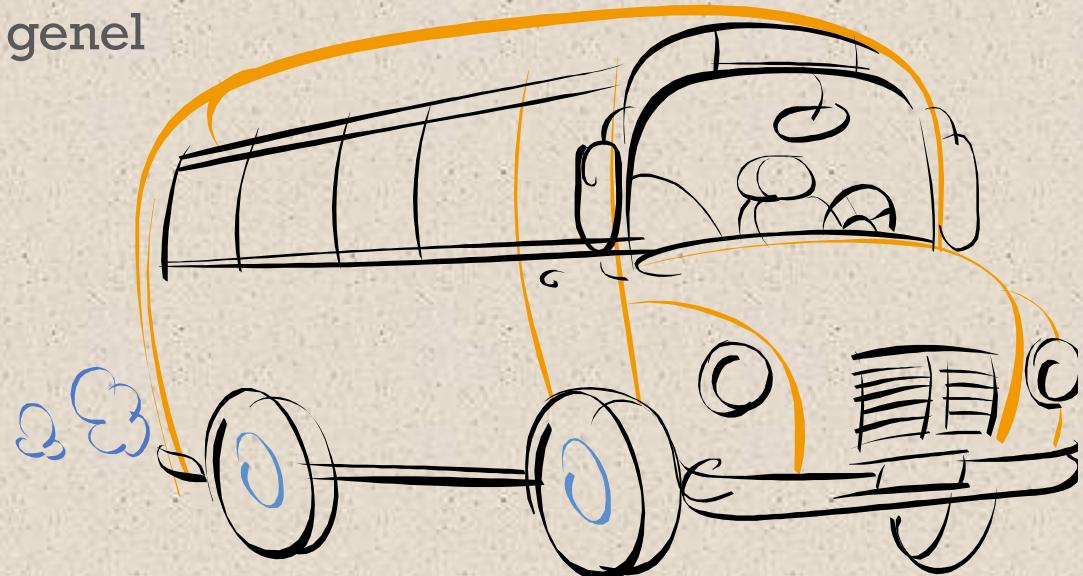
Sistem Veri Yolu:

- Temel bilgisayar bileşenlerini bağlayan veri yolu (işlemci, bellek, I/O)

En yaygın bilgisayar ara bağlantı yapıları, bir veya daha fazla sistem veri yoluna dayalıdır.

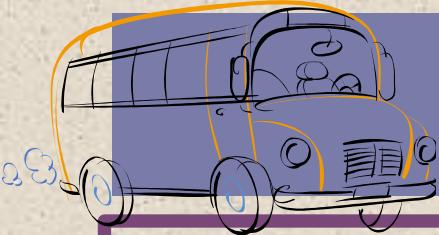
Veri Yolu

- Sistem modülleri arasında veri taşımak için bir yol sağlayan veri hatlarıdır.
- 32, 64, 128, veya daha fazla ayrı hat içerir.
- Hat sayısı, veri yolunun *genişliği* olarak ifade edilir.
- Hat sayısı, bir defada kaç bit aktarılabilceğini belirler.
- Veri yolunun genişliği, genel sistem performansını belirlemede önemli bir faktördür.



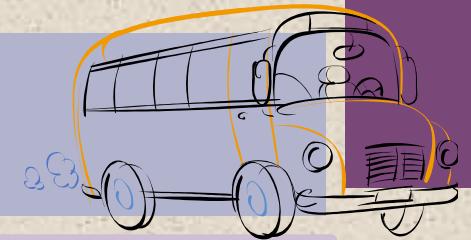
+

Adres Veri Yolu



- Veri yolundaki verilerin kaynağını veya hedefini belirlemek için kullanılır:
- İşlemci bir kelimeyi hafızadan okumak istiyorsa istenen kelimenin adresini adres satırlarına koyar.
- Ayrıca I/O portlarını adreslemek için de kullanılır.
- En önemli bitler, veri yolu üzerindeki belirli bir modülü seçmek için kullanılırken en önemsiz bitler, modülde bir bellek konumunu veya I/O bağlantı noktasını seçer.

Kontrol Veri Yolu



- Verilerin ve adres satırlarının erişimini ve kullanımını kontrol etmek için kullanılır.
- Veri ve adres satırları tüm bileşenler tarafından paylaşıldığından, kullanımlarını kontrol etmenin bir yolu olmalıdır.
- Kontrol sinyalleri hem komut hem de zamanlama bilgisini sistem modülleri arasında ileter.
- Zamanlama sinyalleri, verilerin geçerliliğini ve adres bilgisini gösterir.
- Komut sinyalleri, gerçekleştirilecek işlemleri belirtir.

Veri Yolu Ara Bağlantı Şeması

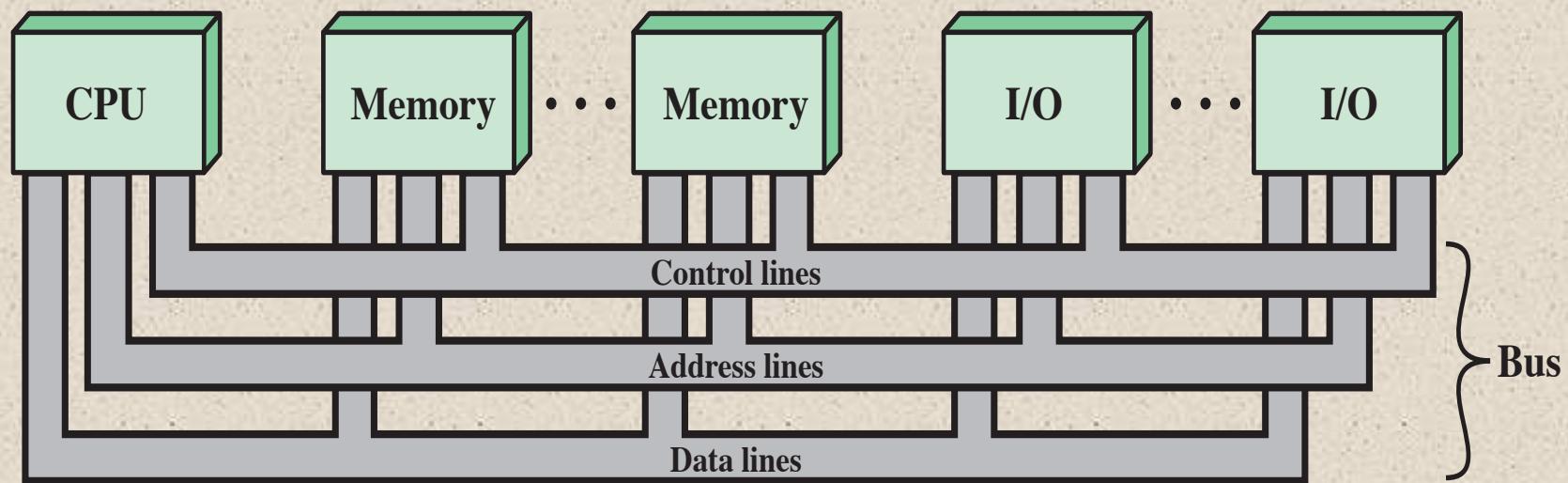
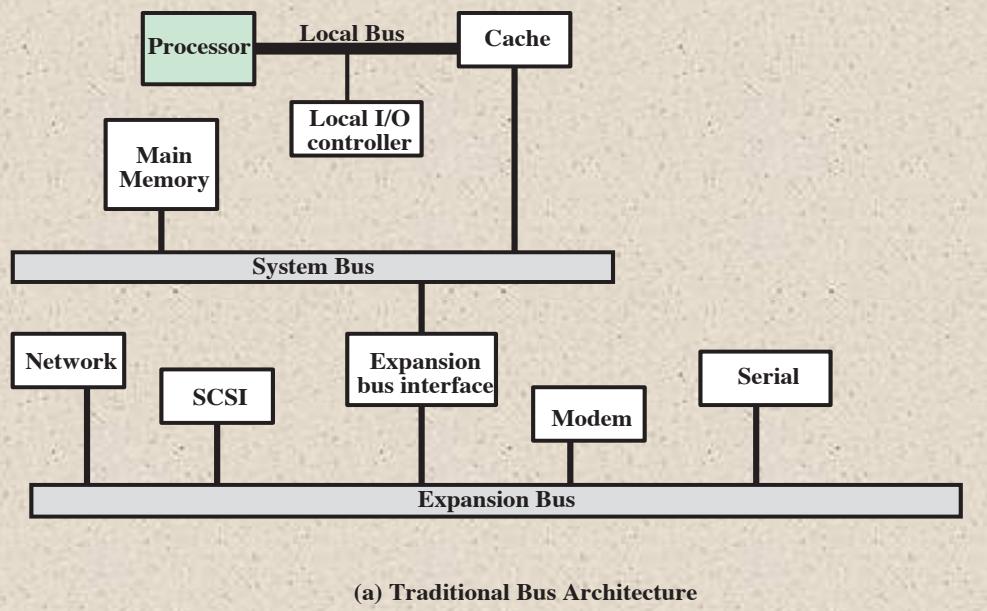
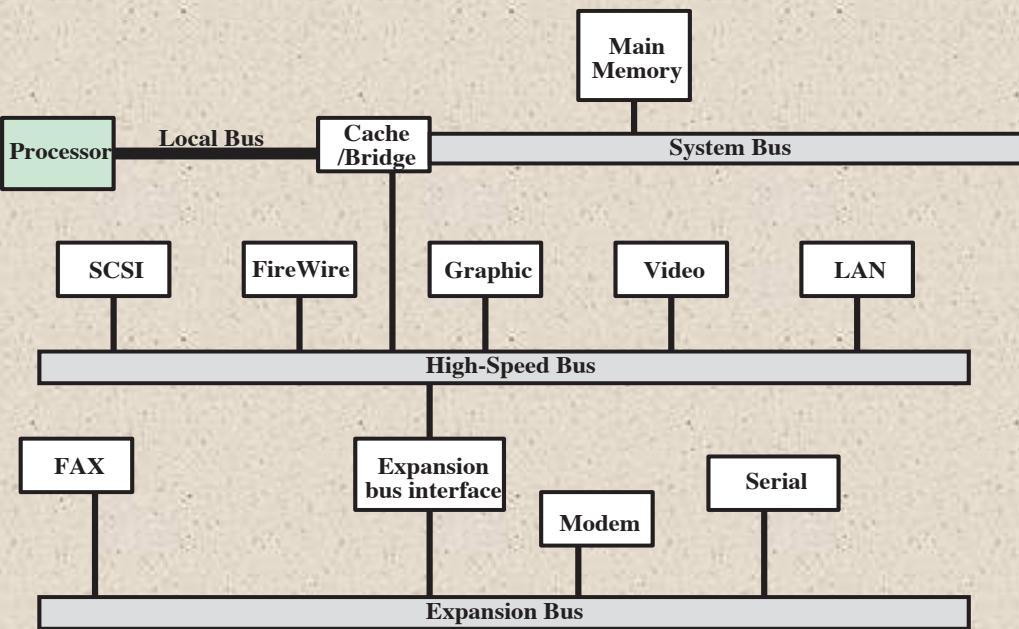


Figure 3.16 Bus Interconnection Scheme

Veri Yolu Yapılandırması



(a) Traditional Bus Architecture



(b) High-Performance Architecture

Figure 3.17 Example Bus Configurations



Veri Yolu Tasarım Elemanları

Type

Dedicated
Multiplexed

Method of Arbitration

Centralized
Distributed

Timing

Synchronous
Asynchronous

Bus Width

Address
Data

Data Transfer Type

Read
Write
Read-modify-write
Read-after-write
Block

Senkron Veri Yolu İşlemlerinin Zamanlaması

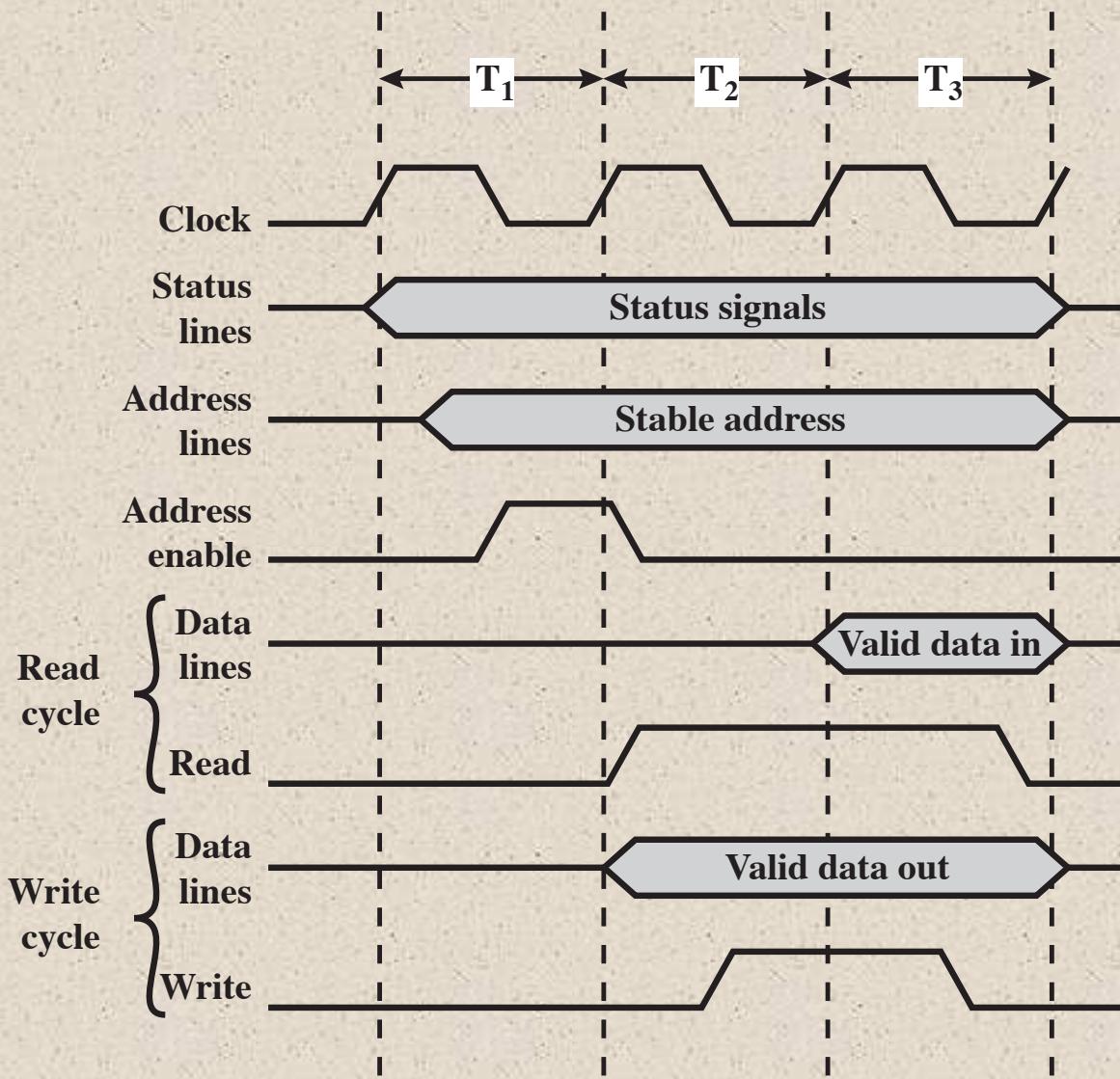
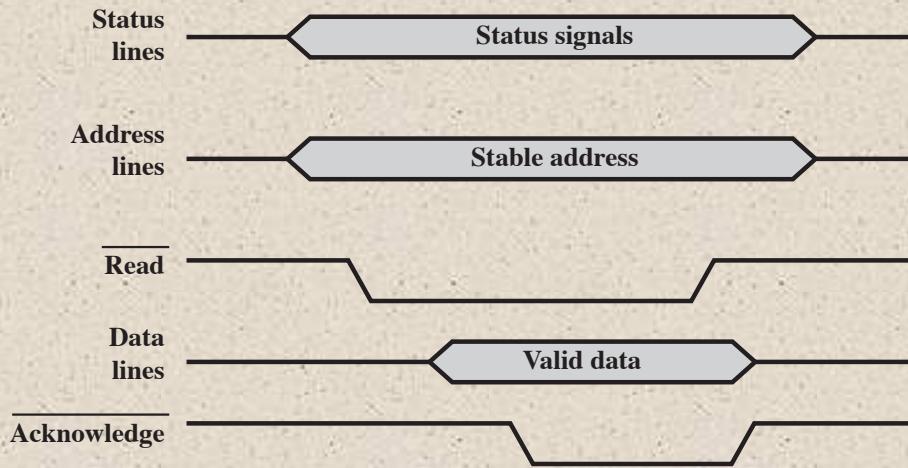


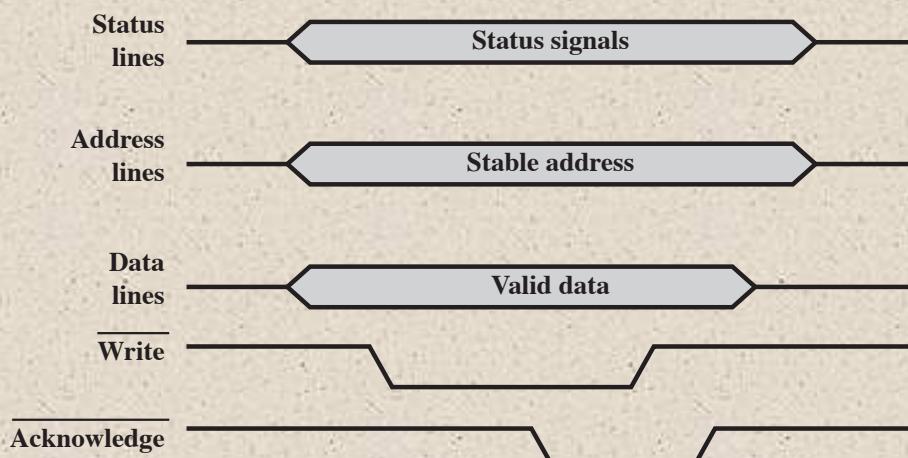
Figure 3.18 Timing of Synchronous Bus Operations



Asenkron Veri Yolu İşlemlerinin Zamanlaması



(a) System bus read cycle



(b) System bus write cycle

Figure 3.19 Timing of Asynchronous Bus Operations



Noktadan Noktaya Bağlantı

Değişikliğin asıl nedeni geniş senkron veri yollarının frekansının arttırılmasıyla karşılaşılan elektriksel kısıtlamalardır.

Daha yüksek veri hızlarında senkronizasyon ve tıpkı işlevlerini zamanında yerine getirmek giderek daha zor hale gelir.

Aynı yongadaki geleneksel bir paylaşımı veri yolu, işlemci sayısına bağlı olarak veri yolu hızını artırma ve veri yolu gecikmesini azaltma zorluklarını arttırdı.

Daha düşük gecikme süresi, daha yüksek veri hızı ve daha iyi ölçülebilirlik!

+

Quick Path Interconnect

- 2008'de tanıtıldı
- Birden fazla doğrudan bağlantı
 - Paylaşılan iletişim sistemlerinde uzlaşma ihtiyacını ortadan kaldırınan diğer bileşenlere doğrudan çift yönlü bağlantılar
- Katmanlı protokol mimarisi
 - Bu işlemci seviyesindeki ara bağlantılar, paylaşımı veri yolu düzenlemelerinde bulunan kontrol sinyallerinin basit kullanımı yerine katmanlı protokol mimarisi kullanırlar
- Paketlenmiş veri transferi
 - Veriler, her biri kontrol başlıkları ve hata kontrol kodlarını içeren bir paket dizisi olarak gönderilir

QPI



QPI Kullanan Çok Çekirdeklerin Konfigürasyonu

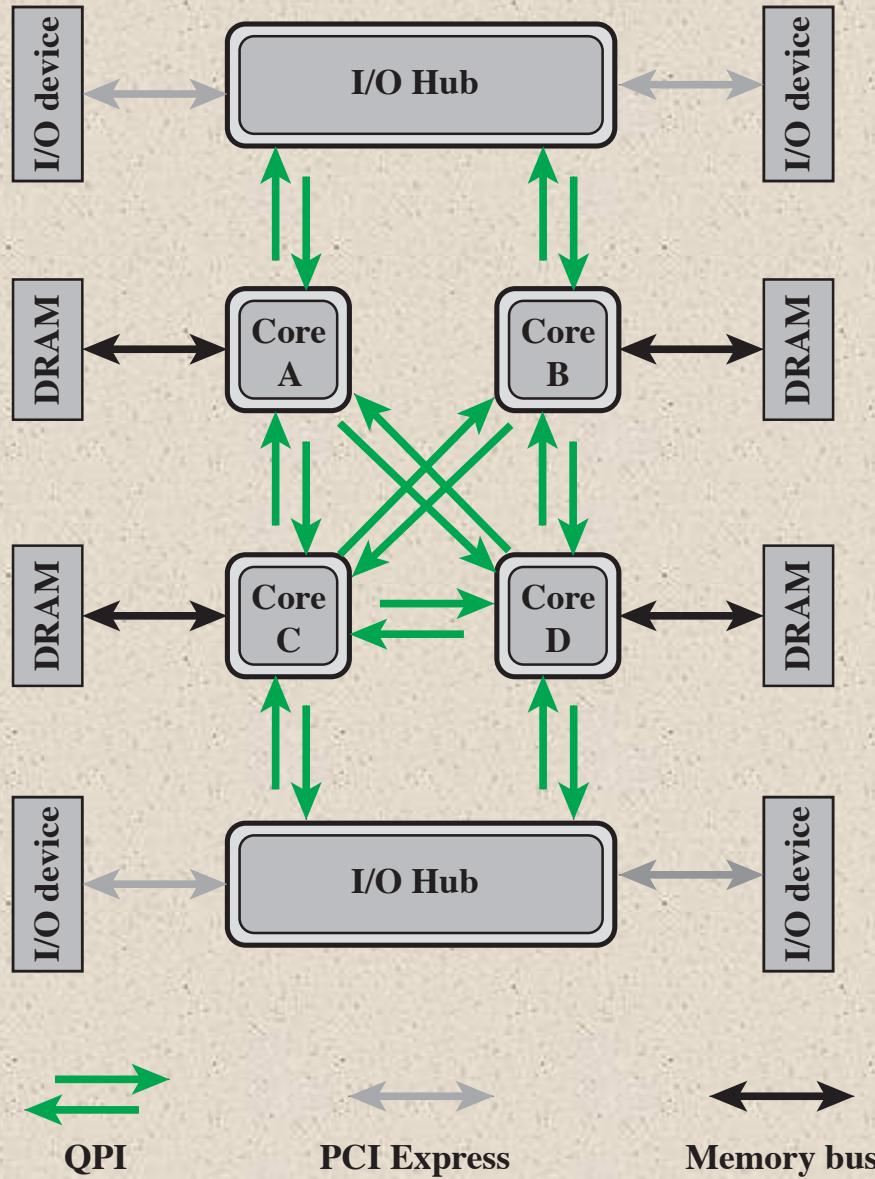


Figure 3.20 Multicore Configuration Using QPI

QPI Katmanları

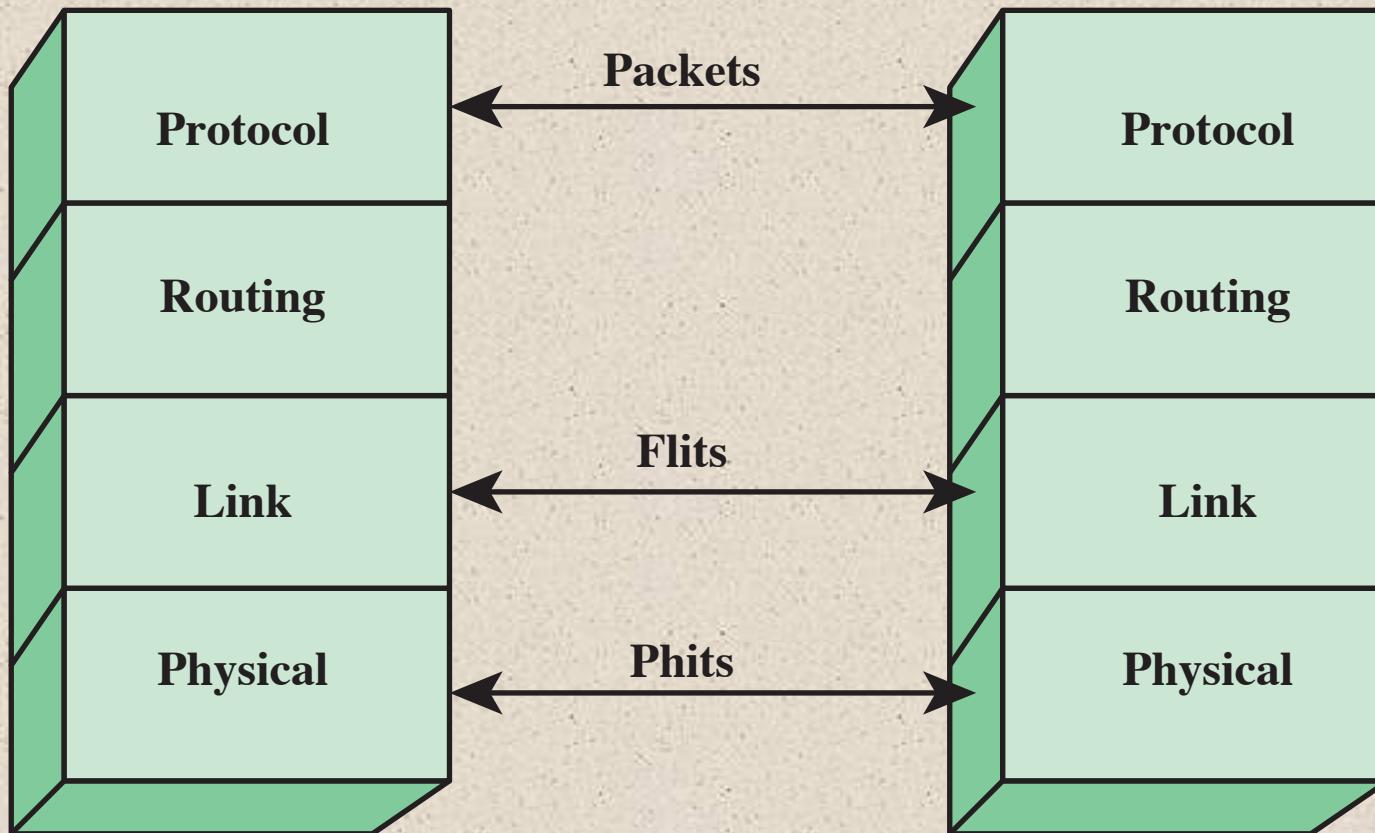


Figure 3.21 QPI Layers

Intel QPI Ara Bağlantısının Fiziksel Arayüzü

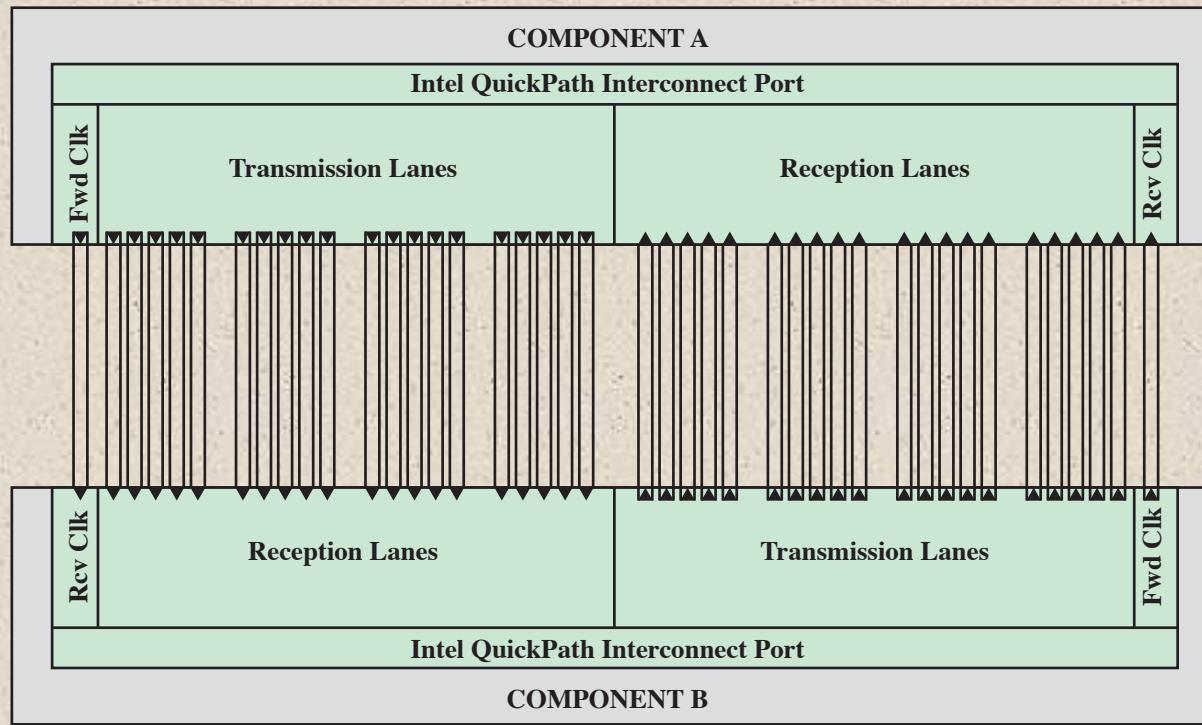


Figure 3.22 Physical Interface of the Intel QPI Interconnect

QPI Multilane Dağılımı

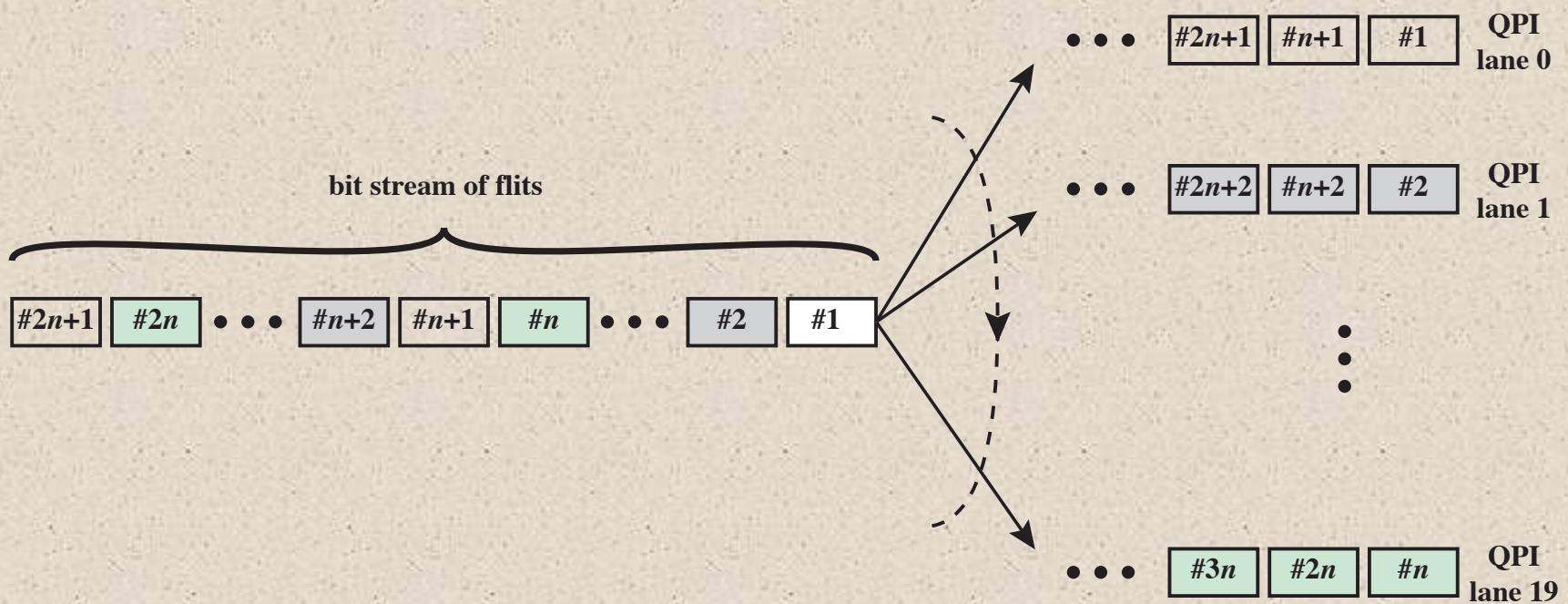


Figure 3.23 QPI Multilane Distribution

QPI Bağlantı Katmanı

- İki temel fonksiyonu gerçekleştirir: **akış kontrolü** ve **hata kontrolü**
 - Flit seviyesinde çalışır (akış kontrol ünitesi)
 - Her flit, 72 bitlik bir mesaj yükünden ve 8 bitlik bir hata kontrol kodundan oluşur ki bu kod döngüsel artıklık denetimi olarak adlandırılır (CRC)
- Akış kontrol fonksiyonu
 - Gönderen bir QPI biriminin, alıcı bir QPI birimini zora sokmamasını garantilemek gereklidir:
 - Alıcının işleyebileceğinden daha hızlı bir şekilde veriyi göndermek ve daha fazla şekilde gelecek veri için önbeliği temizlemek
- Hata kontrol fonksiyonu
 - Bit hatalarını algılar ve kurtarırlar ve böylece üst katmanlarda bit hatalarının yaşanmasının önüne geçilir

QPI Yönlendirme ve Protokol Katmanları

Yönlendirme Katmanı

- Bir paketin mevcut sistem ara bağlantıları boyunca geçeceği rotayı belirlemek için kullanılır.
- Üretici yazılımı tarafından tanımlanır ve bir paketin izleyebileceği olası yolları tanımlar.

Protokol Katmanı

- Paket, aktarım birimi olarak tanımlanır.
- Bu düzeyde gerçekleştirilen temel işlev, birden fazla önbellekte tutulan ana bellek değerlerinin tutarlı olmasını sağlayan bir önbellek tutarlılık protokolüdür.
- Tipik bir veri paketi yükü, bir önbelleğe gönderilen veya önbellekten gönderilen bir veri bloğudur.



Çevresel Bileşen Ara Bağlantısı (PCI)

- Popüler yüksek bant genişliği, bir mezzanine veya çevre birimi veri yolu olarak işlev görebilen işlemci bağımsız veri yolu
- Yüksek hızlı I/O alt sistemleri için daha iyi sistem performansı sunar
- PCI Special Interest Group (SIG)
 - PCI ayırtılarının daha da geliştirilmesi ve uyumluluğunun sağlanması için oluşturulmuştur
- PCI Express (PCIe)
 - PCI gibi veri yoluna dayalı düzenleri değiştirmeyi amaçlayan noktadan noktaya bağlantı şeması
 - Anahtar gereksinimi, Gigabit Ethernet gibi yüksek veri hızlı I/O cihazlarının ihtiyaçlarını karşılayacak yüksek kapasitedir
 - Başka bir gereklilik, zaman bağlılı veri akışlarını destekleme gereksinimi ile ilgilidir



PCIe Konfigürasyonu

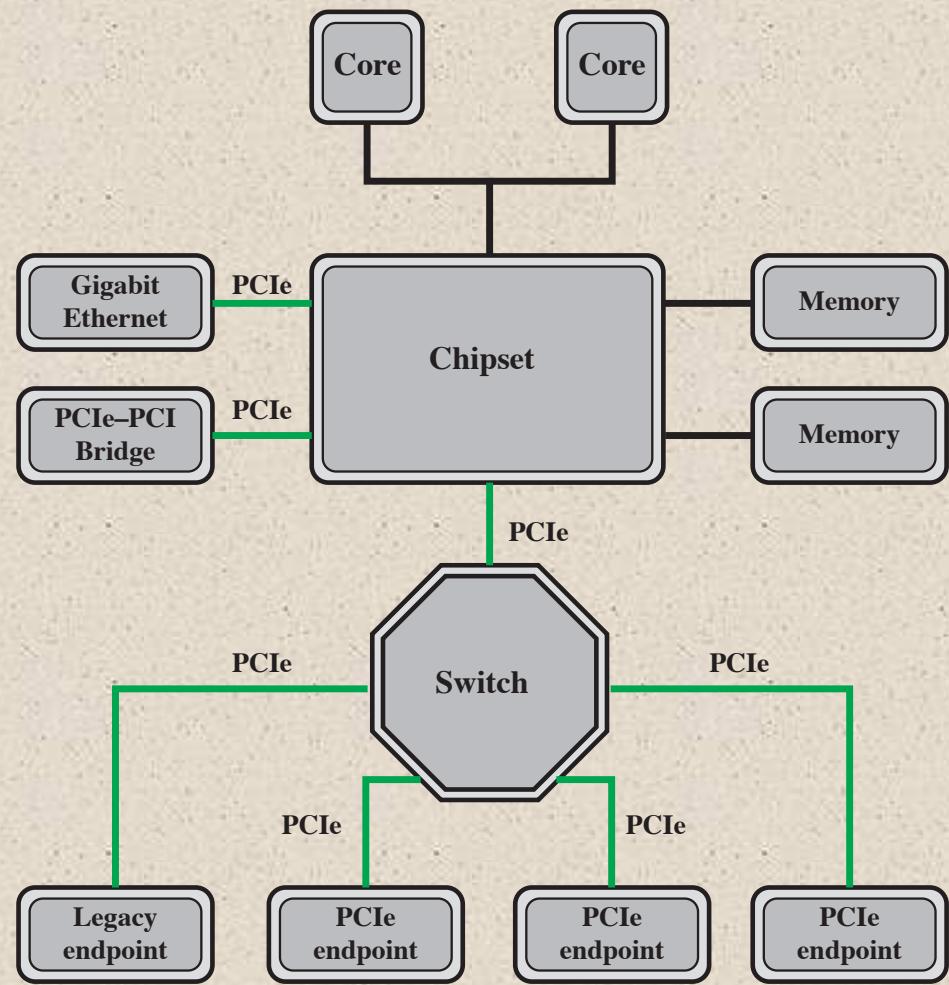


Figure 3.24 Typical Configuration Using PCIe

PCIe Protokol Katmanları

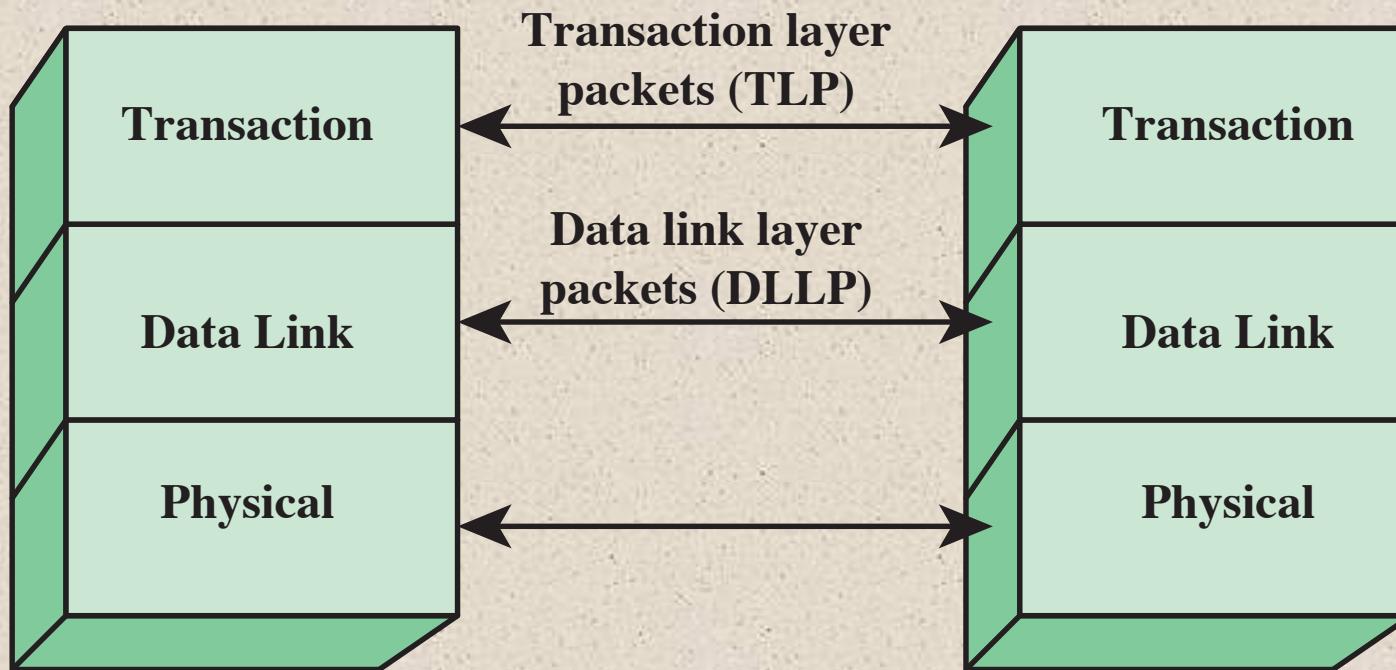


Figure 3.25 PCIe Protocol Layers

PCIe Multilane Dağılımı

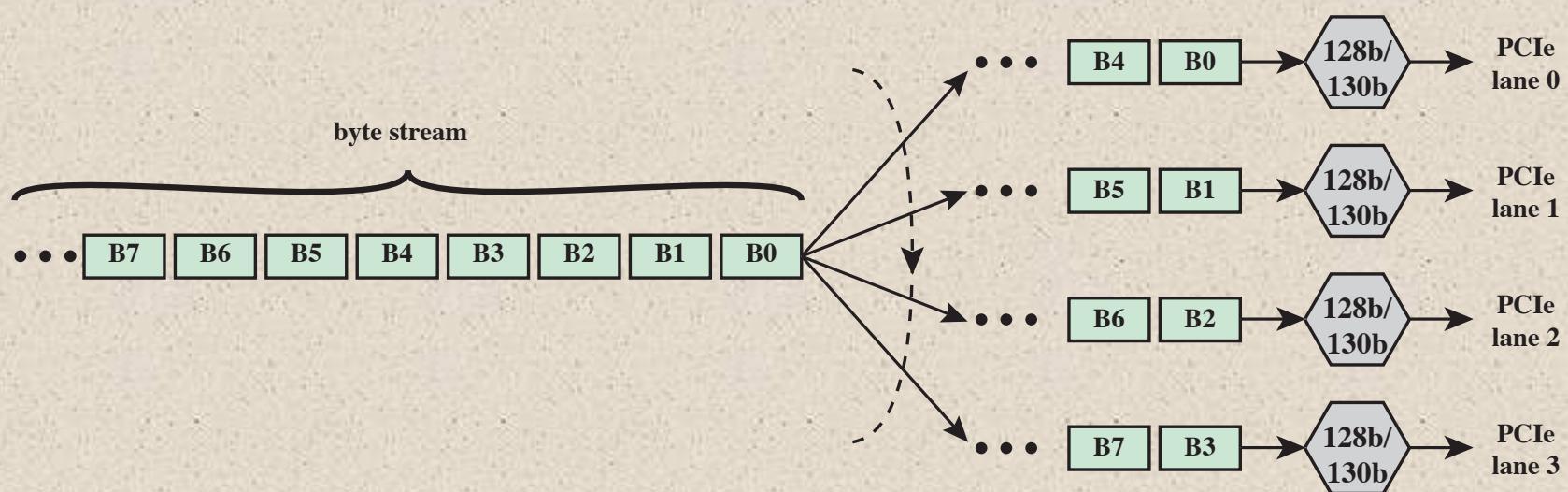


Figure 3.26 PCIe Multilane Distribution

PCIe İletim ve Alım Blok Diyagramları

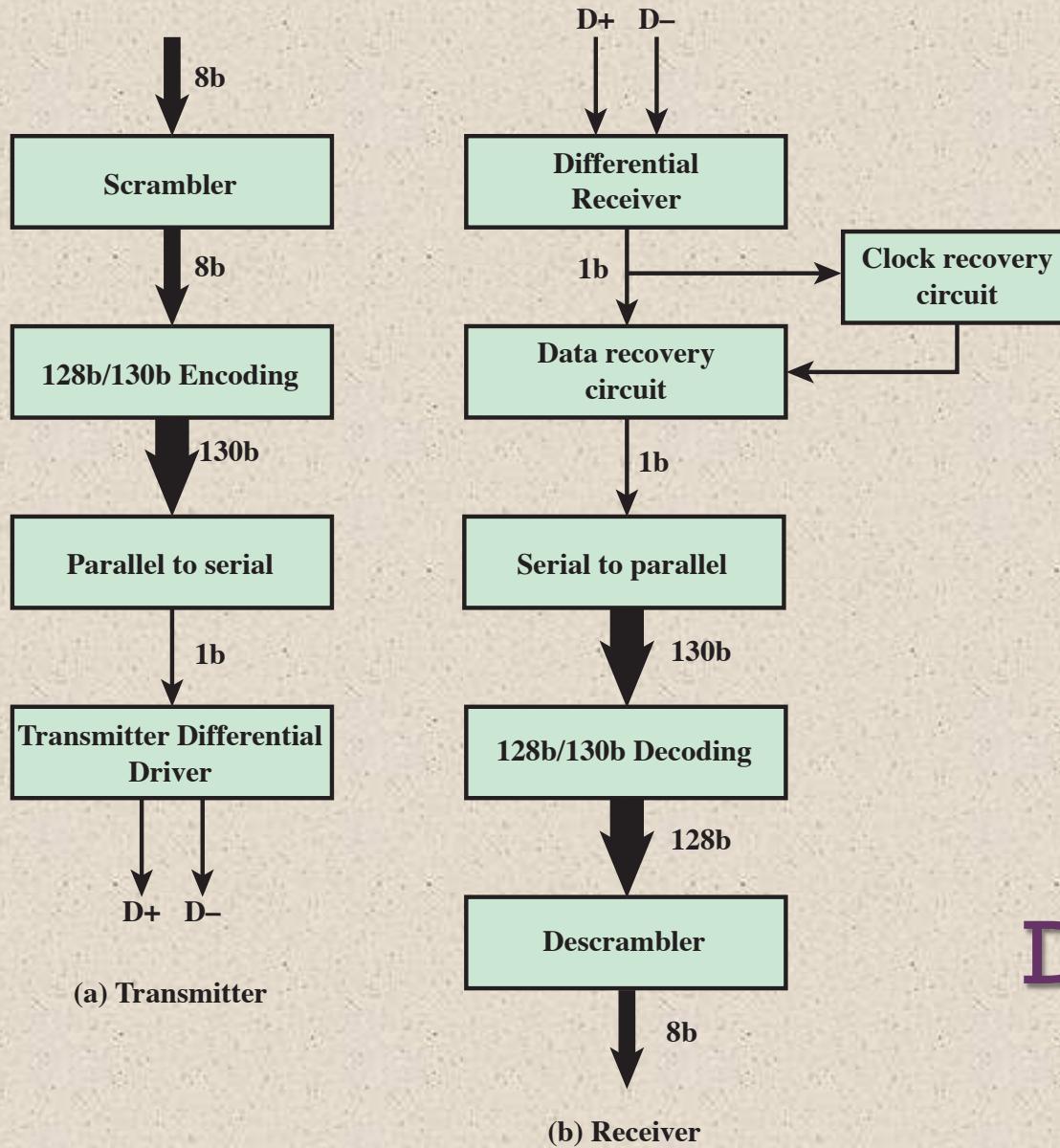


Figure 3.27 PCIe Transmit and Receive Block Diagrams



PCIe

İşlem Katmanı(TL)



- İşlem katmanı, üzerindeki yazılımdan okuma ve yazma isteklerini alır ve bağlantı katmanı yoluyla bir varış noktasına iletim için istek paketleri oluşturur.
- Pek çok işlem bir *split transaction* tekniği kullanır:
 - Bir istek paketi bir kaynak PCIe cihazı tarafından gönderilir ve daha sonra bir tamamlama paketi olarak adlandırılan bir yanıt bekler
- TL iletileri ve bazı yazma işlemleri gönderilen işlemlerdir (anlamı: herhangi bir cevap beklenmemektedir)
- TL paket formatı 32-bit bellek adreslemeyi ve genişletilmiş 64-bit bellek adreslemeyi destekler.

TL dört adres uzayını destekler:

- Bellek
 - Bellek alanı, sistem ana belleğini ve PCIe I/O aygıtlarını içerir.
 - Bellek adreslerinin belli aralıkları I/O aygıtlarına eşlenir.
- Konfigürasyon
 - Bu adres alanı, TL'ye I/O aygıtlarıyla ilişkili konfigürasyon kayıtlarını okuma/yazma imkanı sağlar.
- I/O
 - Bu adres alanı, eski I/O aygıtlarını adreslemek için kullanılan ayrılmış adres aralıklarıyla eski PCI aygıtları için kullanılır.
- Mesaj
 - Bu adres alanı, kesmeler, hata işleme ve güç yönetimi ile ilgili kontrol sinyalleri içindir.

+

Özet

Bölüm 3

- Bilgisayar bileşenleri
- Bilgisayar fonksiyonları
 - Komut alıp-getirme ve komutları koşturma
 - Kesmeler
 - I/O fonksiyonları
- Ara bağlantı yapıları
- Veri yolu ara bağlantıları
 - Veri yolu yapısı
 - Çoklu veri yolu hiyerarşileri
 - Veri yolu tasarımının unsurları

Bilgisayar Bileşenleri ve Ara Bağlantılarına Üst Seviye Bakış

- Noktadan noktaya bağlantı
 - QPI fiziksel katman
 - QPI bağlantı katmanı
 - QPI yönlendirme katmanı
 - QPI protokol katmanı
- PCI ifadesi
 - PCI fiziksel ve mantıksal mimarisi
 - PCIe fiziksel katman
 - PCIe işlem katmanı
 - PCIe veri bağlantı katmanı