

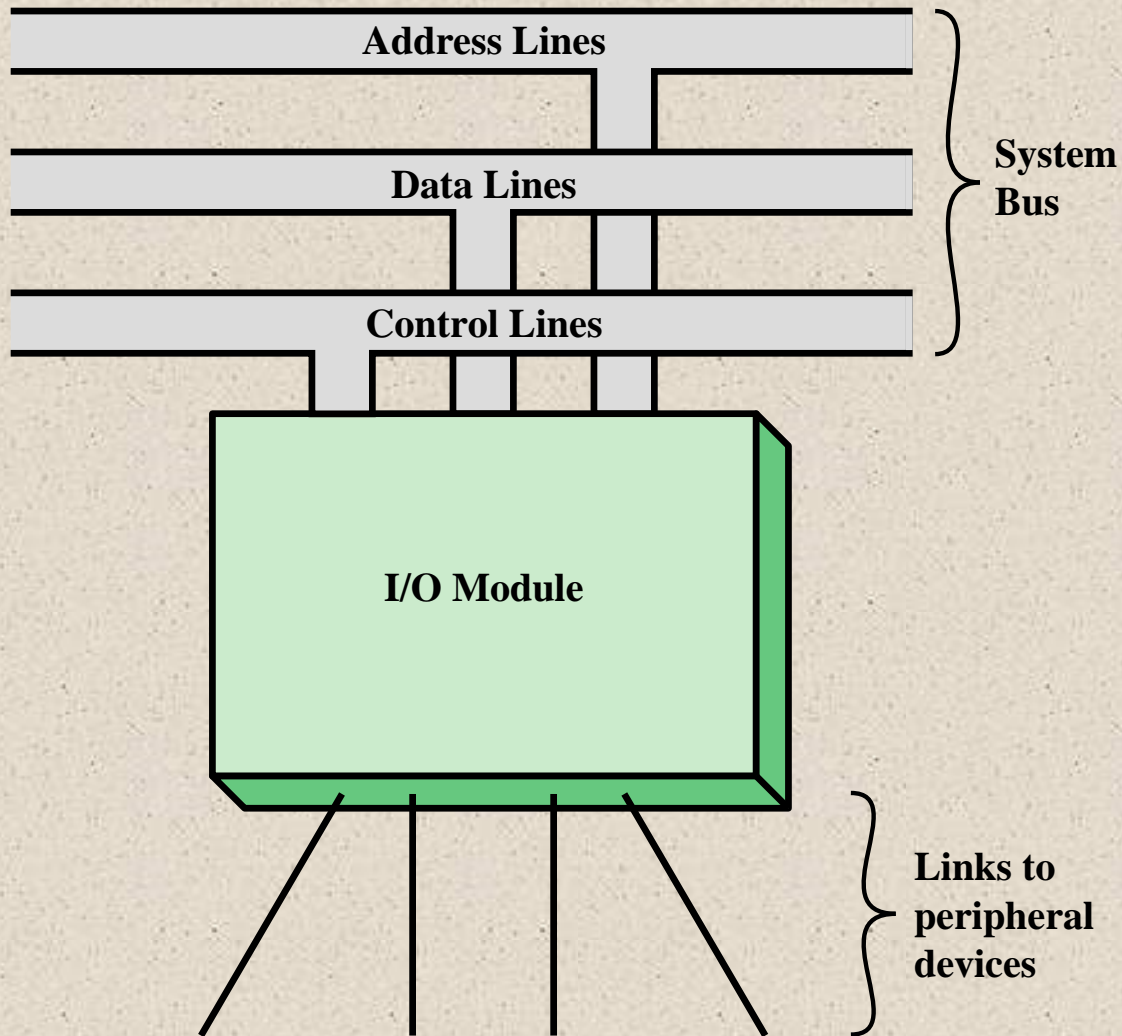
William Stallings  
Computer Organization  
and Architecture  
10<sup>th</sup> Edition

Original slaytların  
çevirisidir.



# + Chapter 7

## Input/Output



Modülün iki ana işlevi vardır:

- Sistem *bus*'ı veya merkezi *switch* aracılığıyla işlemciye ve belleğe arayüz sağlama
- Uyarlanmış veri bağlantıları ile bir veya daha fazla çevresel cihaza arayüz sağlama

**Figure 7.1 Generic Model of an I/O Module**

# + External Devices

Harici cihazları genel olarak  
üç kategoriye ayırabiliriz:

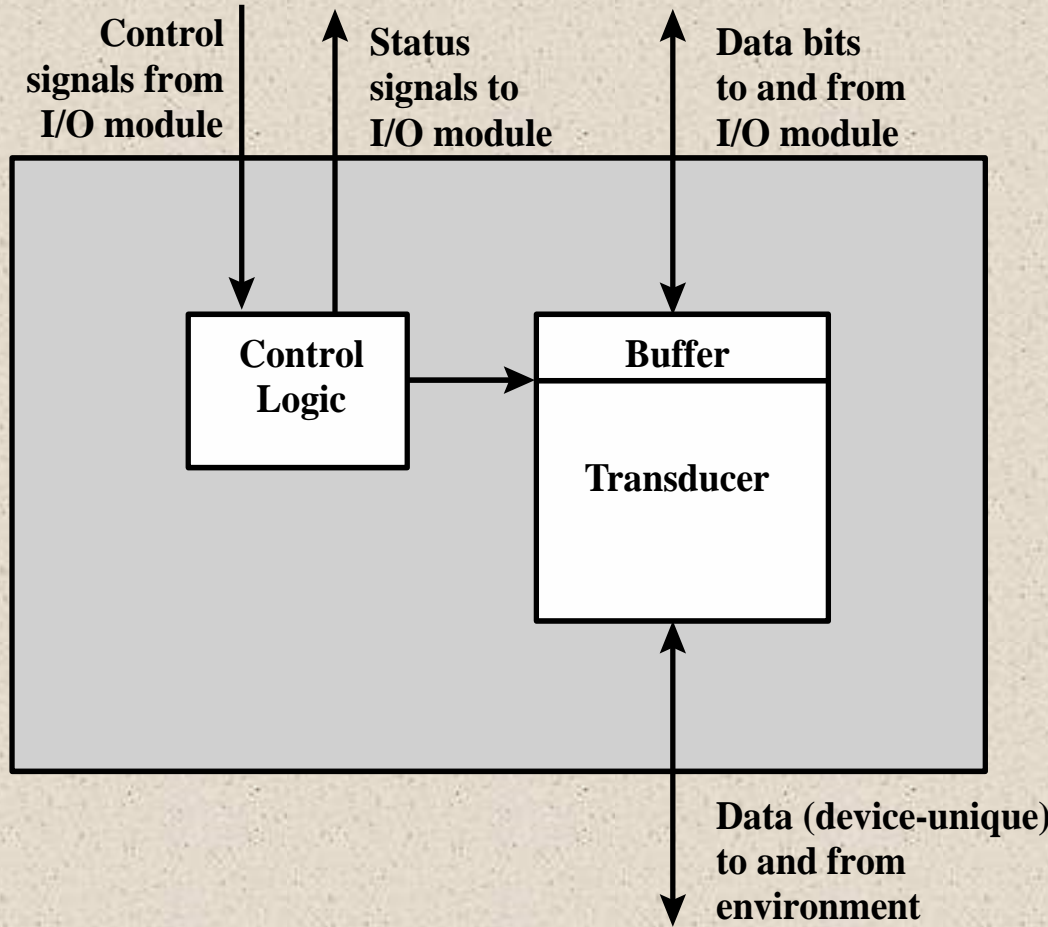


## Three categories:

- Dış ortam ve bilgisayar arasında veri alışverişi için bir yol sağlar
- Bilgisayara bir I/O modül bağlantısıyla eklenir
  - Bağlantı, I/O modülü ile harici cihaz arasında kontrol, durum ve veri alışverişi yapmak için kullanılır
- *Peripheral device*
  - Bir I/O modülüne bağlı harici bir cihaz

- Human readable
  - Bilgisayar kullanıcısı ile iletişim kurmak için uygundur
  - Video görüntüleme terminalleri (VDT'ler), yazıcılar vs.
- Machine readable
  - Ekipmanla iletişim kurmak için uygun
  - Manyetik disk ve teyp sistemleri, sensörler ve aktüatörler
- Communication
  - Bir terminal, makine tarafından okunabilen bir cihaz veya başka bir bilgisayar gibi uzak cihazlarla iletişim kurmak için uygundur





**Figure 7.2 Block Diagram of an External Device**

Çok genel bir ifadeyle, harici bir cihazın yapısı Şekil 7.2'de gösterilmektedir.

8 ila 16 bitlik bir arabellek boyutu seri aygıtlar için yaygındır, oysa disk sürücü denetleyicileri gibi blok yönelimli aygıtlar çok daha büyük arabelleklere sahip olabilir.

Kontrol sinyalleri (*Control signals*),

- I/O modülüne veri göndermek (INPUT veya READ),
- I/O modülünden (OUTPUT veya WRITE) veri kabul etmek,
- durumu raporlamak veya bazı kontrol fonksiyonlarını gerçekleştirmek (örneğin, bir disk kafasını konumlandırma) gibi cihazın gerçekleştireceği işlevi belirler.

Veriler, I/O modülüne gönderilecek veya buradan alınacak bir dizi bit biçimindedir.

Durum sinyalleri (*Status signals*), cihazın durumunu gösterir. Örneğin, aygıtın veri aktarımına hazır olup olmadığını göstermek için READY/NOT-READY sinyali.

Aygıtla ilişkili kontrol lojiği, I/O modülünden gelen komutlara yanıt olarak aygıtın çalışmasını kontrol eder.

Dönüştürücü (**Transducer**), output esnasında elektriksel enerjiden diğer enerji biçimlerine ve input esnasında diğer biçimlerden elektrige dönüştürür.

Tipik olarak, I/O modülü ile harici ortam arasında aktarılan verileri geçici olarak tutmak için dönüştürücü ile bir tampon (buffer)

# + Keyboard/Monitor

## International Reference Alphabet (IRA)

- Temel değişim (*exchange*) birimi karakterdir
  - Her karakterle ilişkili bir koddur
  - Bu koddaki her karakter, benzersiz bir 7-bitlik binary kodla temsil edilir.
    - 128 farklı karakter temsil edilebilir
- Karakterler iki türdendir:
  - Printable
    - Kağıda basılabilen veya ekranda görüntülenebilen alfabetik, sayısal ve özel karakterler
  - Control
    - Karakterlerin yazdırılmasını veya görüntülenmesini kontrol etmekle ilgili
    - Örnek satır başıdır(carriage return)
    - Diğer kontrol karakterleri iletişim prosedürleriyle ilgilidir

Bilgisayar / kullanıcı etkileşiminin en yaygın yolu

Kullanıcı klavye aracılığıyla giriş sağlar

Monitör, bilgisayar tarafından sağlanan verileri görüntüler

## Keyboard Codes

- Kullanıcı bir tuşa bastığında, klavyedeki dönüştürücü tarafından yorumlanan ve ilgili IRA kodunun bit desenine çevrilen bir elektronik sinyal üretir.
- Bu bit deseni bilgisayardaki I/O modülüne iletilir
- Çıkışta, IRA kod karakterleri I/O modülünden harici bir cihaza iletilir.
- Dönüştürücü, kodu yorumlar ve gerekli elektronik sinyalleri, belirtilen karakteri görüntülemek veya istenen kontrol işlevini gerçekleştirmek için çıkış cihazına gönderir.

# Bir I/O modülünün ana işlevleri aşağıdaki kategorilere ayrılır:

## Control and timing

- Dahili kaynaklar ve harici cihazlar arasındaki trafik akışını koordine eder

## Processor communication

- Komut kodunu çözme (**Command decoding**), veri, durum raporlama, adres tanımayı içerir

## Device communication

- Komutları, durum bilgilerini ve verileri içerir

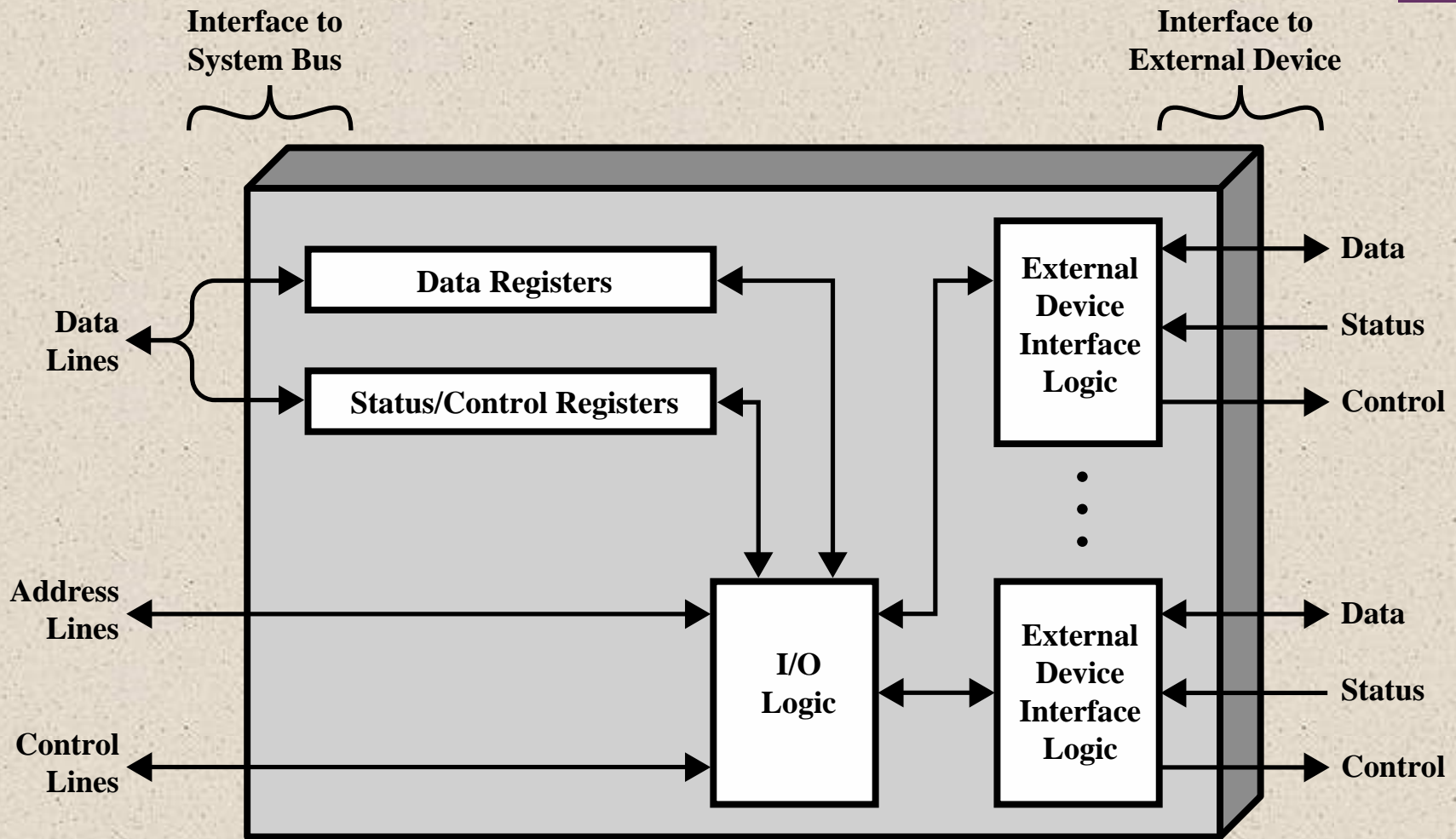
## Data buffering

- Aygıt ve bellek hızlarını dengelemek için gerekli arabelleğe alma işlemini gerçekleştirir

## Error detection

- İletim hatalarını tespit eder ve raporlar

**Command decoding** : I/O modülü işlemciden gelen komutları kabul eder ve tipik olarak kontrol veriyolunda sinyaller olarak gönderilir. Örneğin, bir disk sürücüsü için bir I/O modülü aşağıdaki komutları kabul edebilir: READ SECTOR, WRITE SECTOR, SEEK track number, ve SCAN record ID. Son iki komutun her biri veri yolunda gönderilen bir parametre içerir.



**Figure 7.3 Block Diagram of an I/O Module**



# + Programmed I/O

## I/O işlemleri için üç teknik mümkündür :

- Programmed I/O (*Programlı Giriş/Çıkış*)
  - Veriler, işlemci ve I/O modülü arasında değiş tokuş edilir
  - İşlemci, kendisine I/O işleminin doğrudan denetimini veren bir program yürütür
  - İşlemci I/O modülüne bir komut verdiğinde, I/O işlemi tamamlanana kadar beklemesi gerekir.
  - İşlemci I/O modülünden daha hızlıysa, bu işlemcinin zamanını boşa harcar
- Interrupt-driven I/O (*Kesmeye dayalı Giriş/Çıkış*)
  - İşlemci bir I/O komutu verir, diğer komutları yürütmeye devam eder ve I/O modülü işini tamamladığında I/O modülü tarafından kesmeye uğratılır
- Direct memory access (DMA)
  - I/O modülü ve ana bellek, işlemcinin müdahalesi olmadan doğrudan veri alışverişi yapar

Tablo 7.1, bu üç teknik arasındaki ilişkiyi göstermektedir



## Table 7.1

### I/O Techniques

	No Interrupts	Use of Interrupts
I/O-to-memory transfer through processor	Programmed I/O	Interrupt-driven I/O
Direct I/O-to-memory transfer		Direct memory access (DMA)



# I/O Commands



- Bir işlemci tarafından adreslendiğinde bir I/O modülünün alabileceği dört tür I/O komutu vardır:

## 1) Control

Örneğin, bir manyetik teyp birimine bir kaydı geri sarması veya bir kaydı ileri götürmesi komutu verilebilir. Bu komutlar, belirli bir çevresel aygıt türüne göre uyarlanmıştır.

- bir çevre birimini etkinleştirmek ve ona ne yapması gerektiğini söylemek için kullanılır

## 2) Test

İşlemci, ilgilenilen çevre biriminin açık ve kullanıma hazır olduğunu bilmek isteyecektir. Ayrıca en son I/O işleminin tamamlanıp tamamlanmadığını ve herhangi bir hata olup olmadığını bilmek isteyecektir.

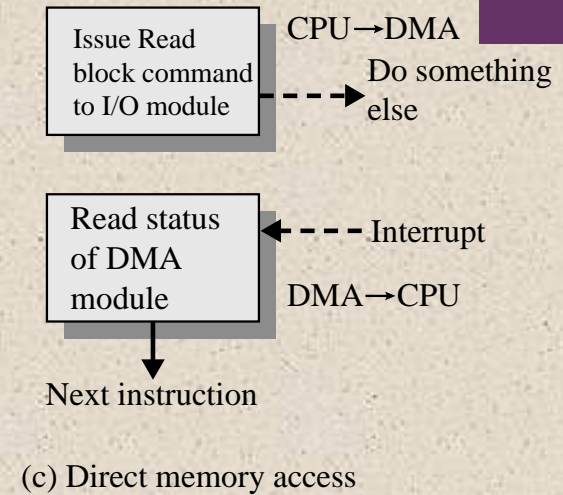
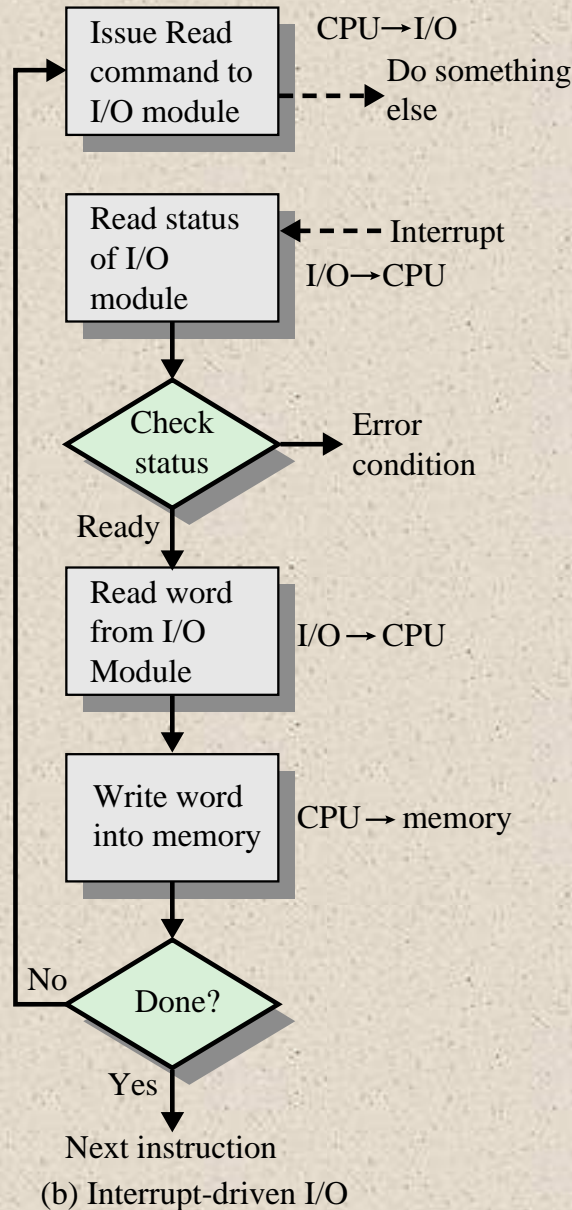
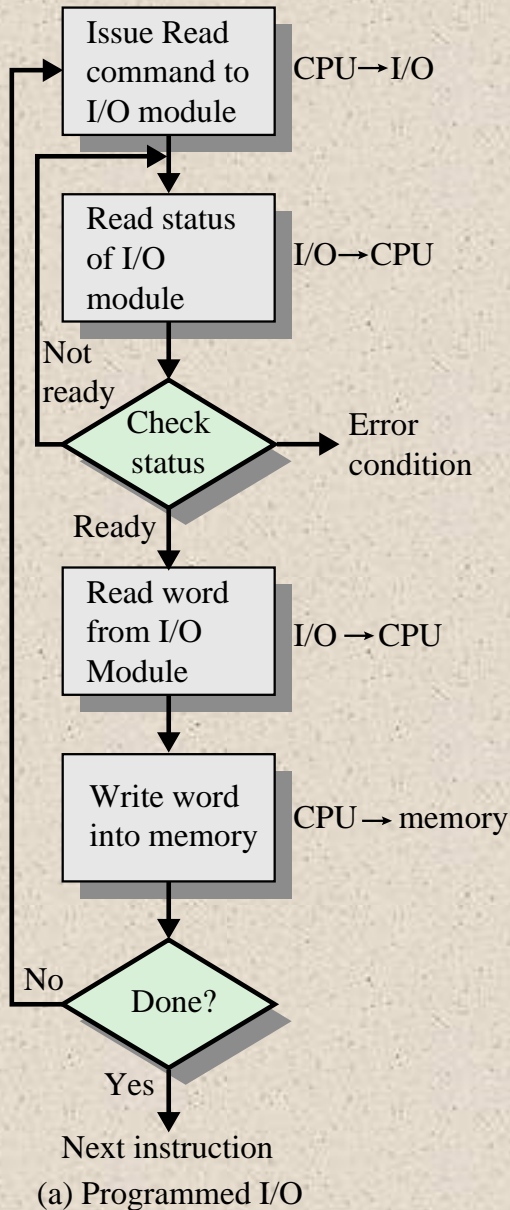
- bir I/O modülü ve çevre birimleriyle ilişkili çeşitli durum koşullarını test etmek için kullanılır

## 3) Read

- I/O modülünün çevre birimden bir veri ögesi almasını ve bunu dahili bir arabelleğe yerleştirmesini sağlar

## 4) Write

- I/O modülünün veri yolundan bir veri ögesi almasına ve ardından bu veri ögesini çevre birimine iletmesine neden olur



Şekil 7.4a, bir çevresel aygıttan (örneğin, teypten bir kayıt) belleğe bir veri bloğunu okumak için programlanmış I/O kullanımına bir örnek verir. Veriler bir seferde tek kelimeyle (örneğin 16 bit) okunur. Okunan her kelime için, işlemci, kelimenin I/O modülünün veri registerinde mevcut olduğunu belirleyene kadar bir durum kontrol döngüsünde (*status-checking cycle*) kalmalıdır. Bu akış şeması, bu tekniğin ana dezavantajını vurgulamaktadır: işlemciyi gereksiz yere meşgul eden zaman alıcı bir işlemdir.

**Figure 7.4 Three Techniques for Input of a Block of Data**



# I/O Instructions

Programlı I/O ile işlemcinin bellekten alıp getirdiği I/O ile ilgili komutlar ve işlemcinin komutları yürütmek için bir I/O modülüne verdiği I/O komutları arasında yakın bir ilişki vardır.

Komutun şekli, harici cihazların adreslenme şekline bağlıdır.

I/O modülleri aracılığıyla bağlanan her I/O cihazına benzersiz bir tanımlayıcı veya adres verilir

İşlemci bir I/O komutu verdiğinde, komut istenen aygıtın adresini içerir

Bu nedenle, her bir I/O modülü, komutun kendisi için olup olmadığını belirlemek için adres satırlarını yorumlamalıdır.

## Memory-mapped I/O

Bellek konumları ve I/O aygıtları için tek bir adres alanı vardır

Veriyolunda tek bir okuma hattı ve tek bir yazma hattına ihtiyaç vardır

İşlemci, ana bellek ve I/O ortak bir veri yolunu paylaştığında, iki adresleme modu mümkündür: «memory mapped» and «isolated»

### memory-mapped I/O

İşlemci, I/O modüllerinin durum ve veri registerlarını bellek konumları olarak ele alır ve hem belleğe hem de I/O cihazlarına erişmek için aynı makine komutlarını kullanır



# I/O Mapping Summary

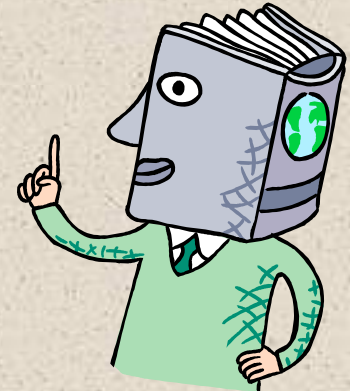


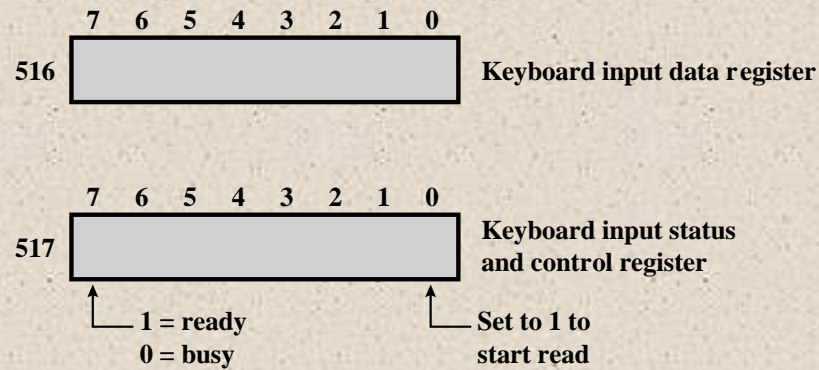
## ■ Memory mapped I/O

- Cihazlar ve bellek tek bir adres alanını paylaşır
- I/O tıpkı bellek okuma/yazma gibi görünür
- I/O için özel komut yok
  - Geniş bellek erişim komutları yelpazesi mevcuttur

## ■ Isolated I/O

- Ayrı adres alanları
- I/O veya bellek seçme hatlarına ihtiyaç var
- I/O için özel komutlar
  - Sınırlı bir set





ADDRESS	INSTRUCTION	OPERAND	COMMENT
200	Load AC	"1"	Load accumulator
	Store AC	517	Initiate keyboard read
202	Load AC	517	Get status byte
	Branch if Sign = 0	202	Loop until ready
	Load AC	516	Load data byte

(a) Memory-mapped I/O

ADDRESS	INSTRUCTION	OPERAND	COMMENT
200	Load I/O	5	Initiate keyboard read
201	Test I/O	5	Check for completion
	Branch Not Ready	201	Loop until complete
	In	5	Load data byte

(b) Isolated I/O

**Figure 7.5 Memory-Mapped and Isolated I/O**

# Interrupt-Driven I/O

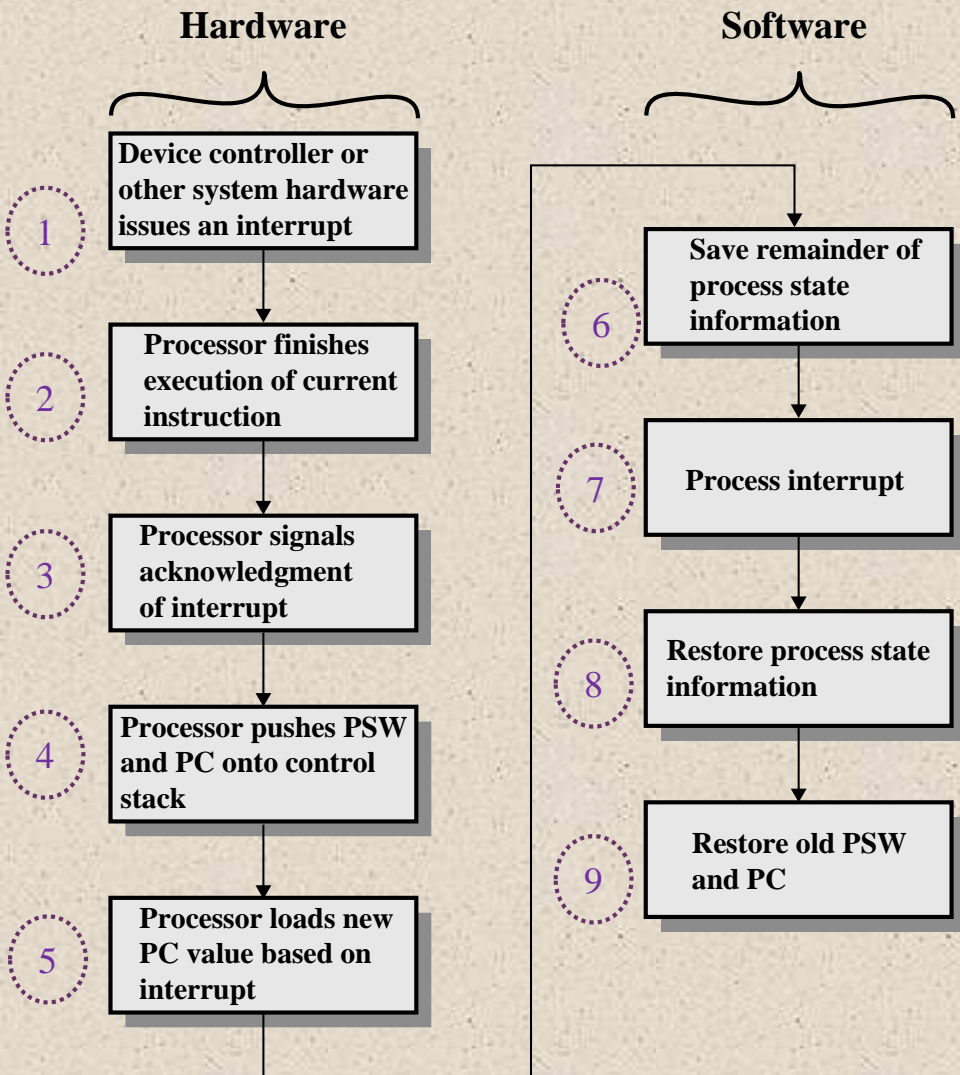
Programlı I/O ile ilgili sorun, işlemcinin I/O modülünün veri alımı veya iletimi için hazır olması için uzun süre beklemesi gerekmesidir.

Bir alternatif, işlemcinin bir modüle bir I/O komutu vermesi ve ardından başka yararlı işler yapmaya devam etmesidir.

I/O modülü, işlemciyle veri alışverişi yapmaya hazır olduğunda servis almak için kesme isteği gönderir.

İşlemci veri aktarımını yürütür ve sonrasında önceki işlemeye (kaldığı yerden) devam eder



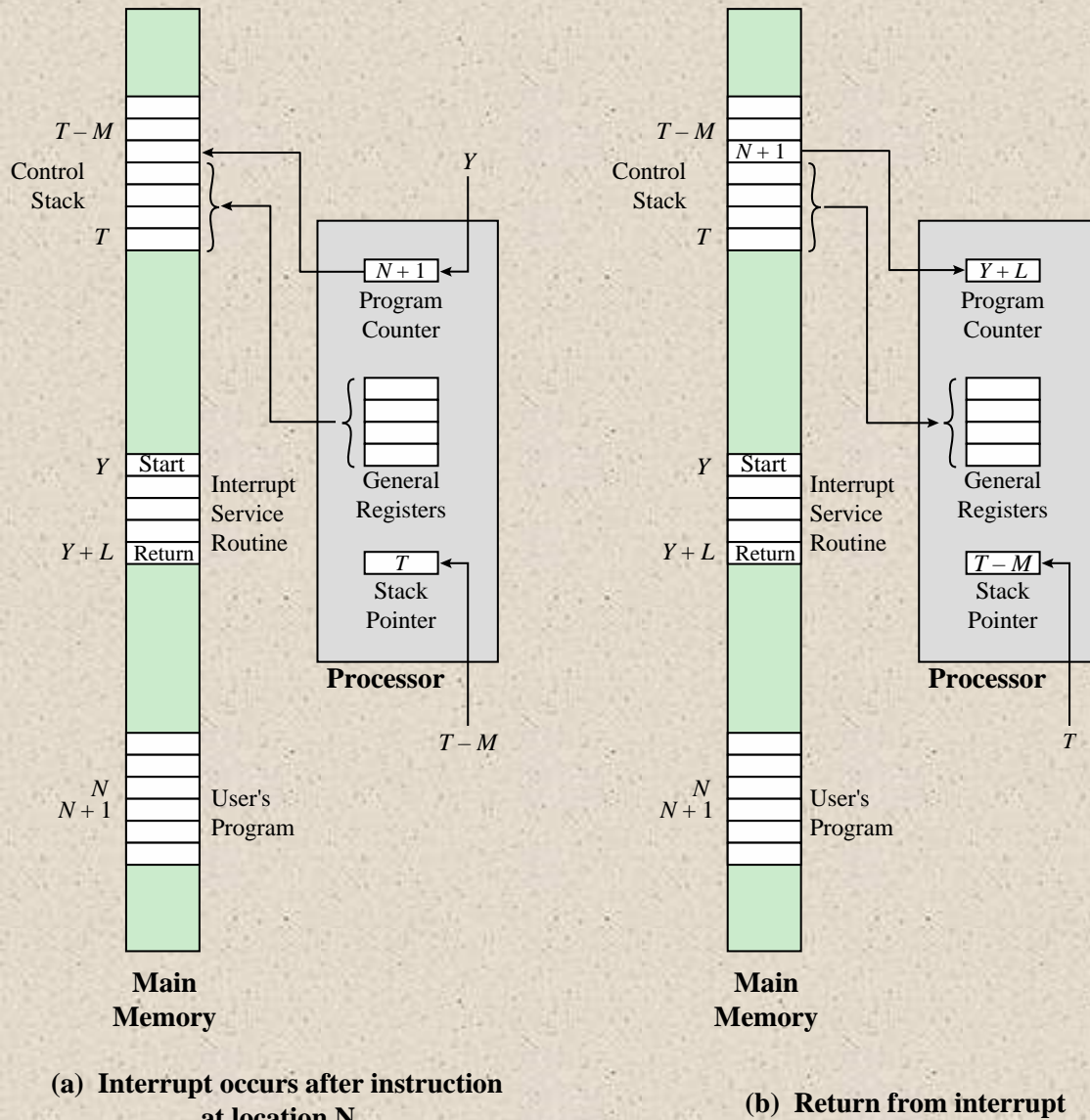


**Figure 7.6 Simple Interrupt Processing**

İşlemcinin kesmeye dayalı I/O'daki rolünü daha ayrıntılı olarak ele alalım. Bir kesmenin meydana gelmesi, hem işlemci **donanımında** hem de **yazılımda** bir dizi olayı tetikler. Şekil 7.6, tipik bir diziyi göstermektedir.

Bir I/O cihazı bir I/O işlemini tamamladığında, aşağıdaki donanım olayları dizisi gerçekleşir:

1. Cihaz, işlemciye bir kesme sinyali gönderir.
2. İşlemci, kesmeye yanıt vermeden önce mevcut komutu yürütmeyi bitirir.
3. İşlemci kesmeyi test eder, bir kesme olduğunu belirler ve kesmeyi yayınlayan cihaza bir onay sinyali gönderir. Onay, cihazın kesme sinyalini kaldırmasına izin verir.
4. İşlemcinin artık kontrolü kesme yordamına aktarmaya hazır olması gerekir. Başlamak için, kesme noktasında mevcut programı devam ettirmek için gereken bilgileri kaydetmesi gerekir. Gerekli minimum bilgi (a) program durum kelimesi (**PSW**) olarak adlandırılan bir registerda bulunan işlemcinin durumu ve (b) **program sayacında** bulunan, yürütülecek bir sonraki komutun yeridir. . Bunlar, sistem kontrol yığınına itilebilir.
5. İşlemci şimdi program sayacını, bu kesmeye yanıt verecek olan kesme işleme programının giriş konumu ile birlikte yükler.



**Figure 7.7 Changes in Memory and Registers for an Interrupt**

kontrol, interrupt-handler programına aktarılır. Bu programın yürütülmesi aşağıdaki işlemlerle sonuçlanır:

6. Bu noktada, kesmeye uğramış programla ilgili program sayacı ve PSW, sistem yığına kaydedilmiştir. Bununla birlikte, çalışan programın "durumunun" bir parçası olarak kabul edilen başka bilgiler de vardır. Özellikle, işlemci registerlarının içeriğinin kaydedilmesi gerekir, çünkü bu registerlar kesme işleyicisi (interrupt-handler) tarafından kullanılabilir. Bu nedenle, tüm bu değerlerin yanı sıra diğer durum bilgilerinin de kaydedilmesi gerekir. Tipik olarak, kesme işleyicisi tüm registerlarının içeriğini yığına kaydederek başlayacaktır.

7. Kesme işleyicisi daha sonra kesmeyi işler.

8. Kesme işlemi tamamlandığında, kaydedilen register değerleri yığından alınır ve registerlara geri yüklenir (örn., Şekil 7.7b'ye bakın).

9. Son eylem, PSW'yi ve program sayacını yığından geri yüklemektir. Sonuç olarak, yürütülecek bir sonraki komut daha önce kesmeye uğramış programdan olacaktır.

# Design Issues

**Kesmeli I / O  
uygulamasında  
iki tasarım  
sorunu ortaya  
çıkır:**

- Birden fazla I/O modülü olacağı için işlemci hangi aygıtın kesmeyi yayınladığını nasıl belirler?
- Birden fazla kesme meydana gelmişse işlemci hangisini işleyeceğine nasıl karar verir?



# + Device Identification

Dört genel teknik kategorisi yaygındır:

- **Multiple interrupt lines**

- İşlemci ve I/O modülleri arasında
- Soruna en basit yaklaşım
- Sonuç olarak, birden fazla hat kullanılsa bile, muhtemelen her bir hatta birden fazla I / O modülü bağlanacaktır.

- **Software poll**

- İşlemci bir kesme algıladığında, görevi hangi modülün kesmeye neden olduğunu belirlemek için her bir I/O modülünü yoklamak olan bir kesme servis rutinine daller.
- Zaman alıcıdır (Time consuming)

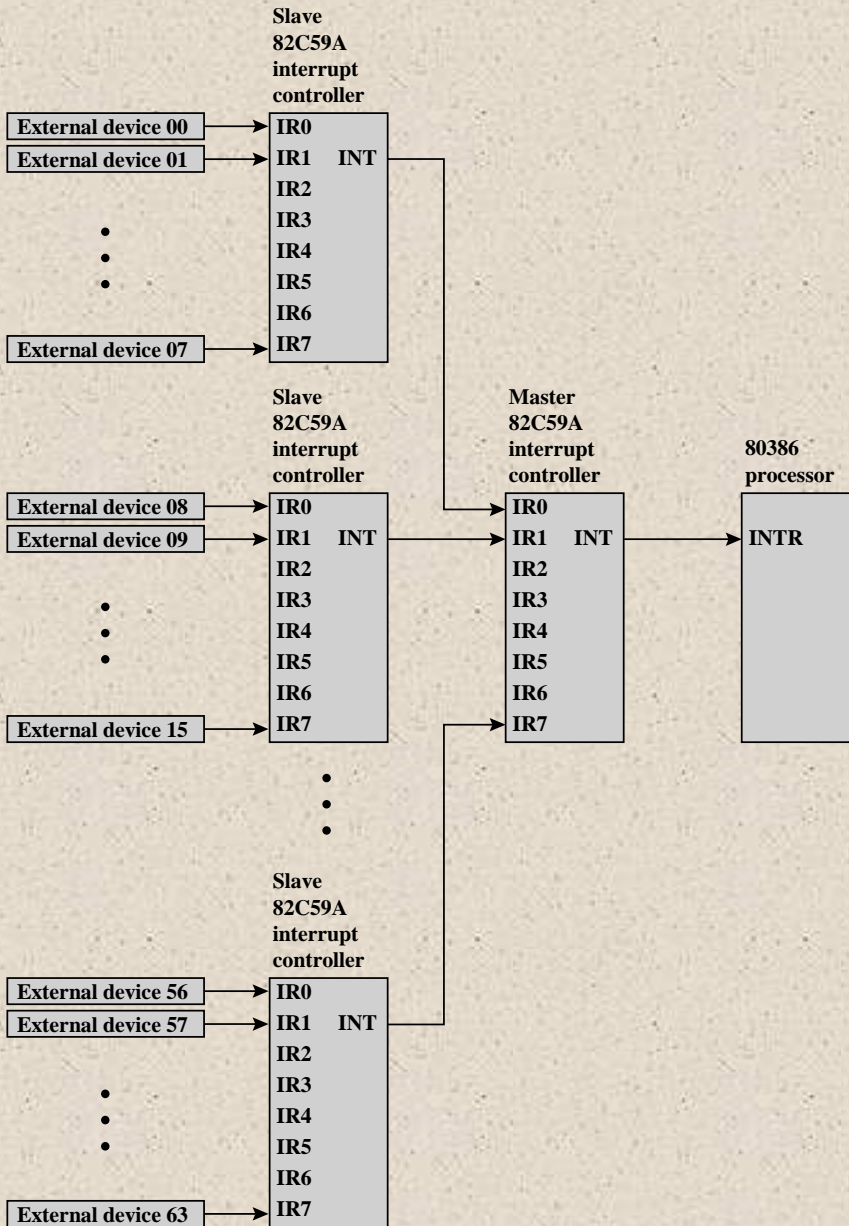
- **(Papatya zinciri) Daisy chain (hardware poll, vectored)**

- Kesmeler için, tüm I/O modülleri ortak bir kesme talebi hattını paylaşır. Kesme onay hattı, modüller aracılığıyla papatya dizimlidir
- Vector - I/O modülünün adresi veya başka bir benzersiz tanımlayıcı
- Vectored interrupt – işlemci, vektörü uygun cihaz servis rutini için bir işaretçi olarak kullanır ve önce genel bir kesme servis rutini yürütme ihtiyacını ortadan kaldırır

- **Bus arbitration (vectored)**

- Bir I/O modülü, kesme talebi hattını yükseğe çekmeden önce veri yolunun kontrolünü kazanmalıdır. Böylece, bir seferde yalnızca bir modül hattı yükseğe çekebilir.
- İşlemci, kesmeyi algıladığında, kesme onaylama hattından yanıt verir.
- Daha sonra talep eden modül vektörünü veri hatlarına yerleştirir.

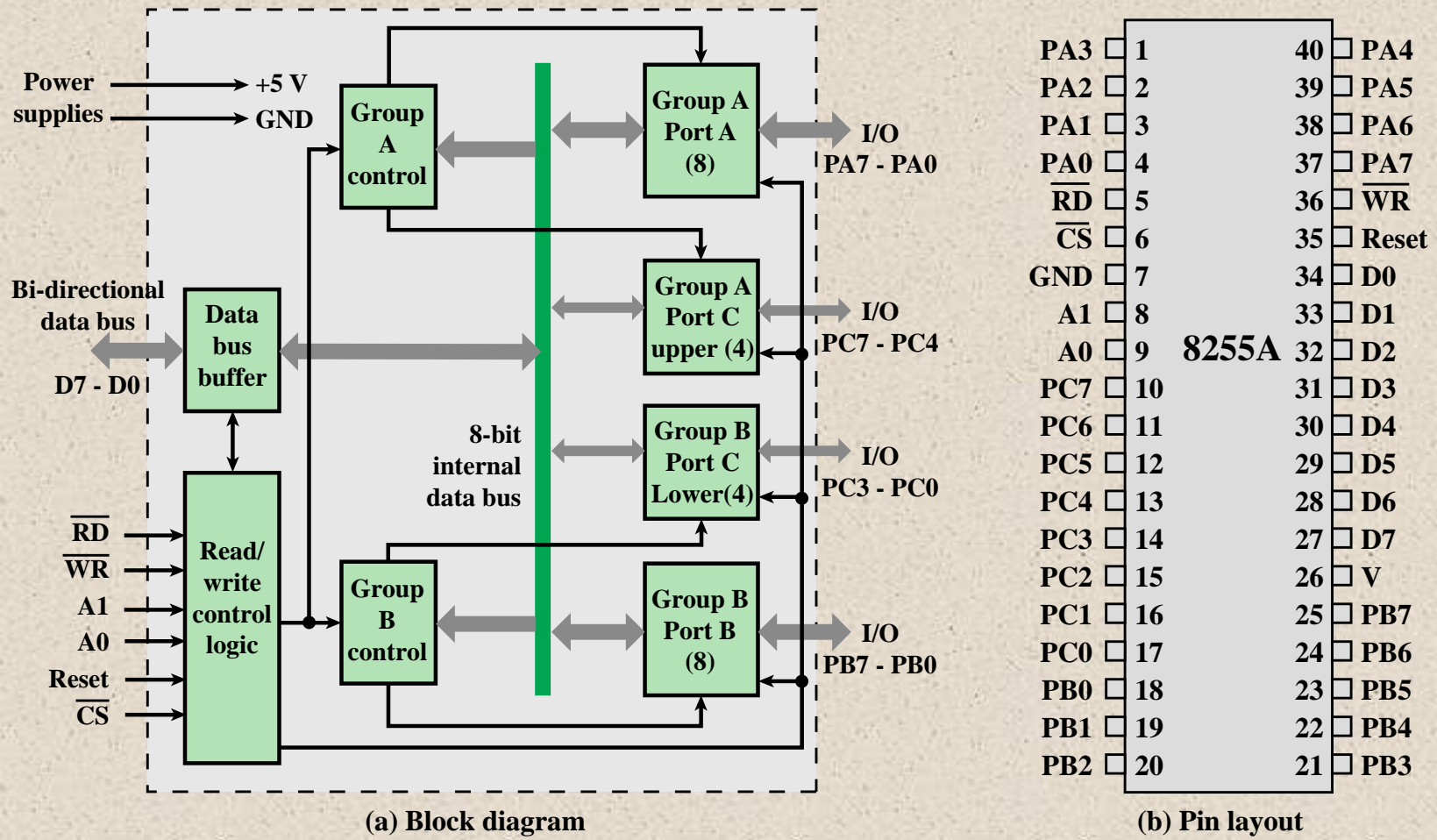




**Figure 7.8 Use of the 82C59A Interrupt Controller**

Intel 80386, tek bir Kesme İsteği (Interrupt Request -INTR) ve tek bir Kesme Onay (Interrupt Acknowledge -INTA) hattı sağlar. 80386'nın çeşitli cihazları ve öncelik yapılarını kullanmasına izin vermek için, genellikle harici bir kesme hakemi olan 82C59A ile yapılandırılır. Harici cihazlar 82C59A'ya bağlanır ve bu da 80386'ya bağlanır.

Şekil 7.8, 80386 için birden çok I/O modülünü bağlamak için 82C59A'nın kullanımını göstermektedir. Tek bir 82C59A, sekiz adede kadar modül idare edebilir. Sekizden fazla modül için kontrol gerekiyorsa, 64 modüle kadar idare etmek için kademeli bir düzenleme (*cascade arrangement*) kullanılabilir.



**Figure 7.9 The Intel 8255A Programmable Peripheral Interface**

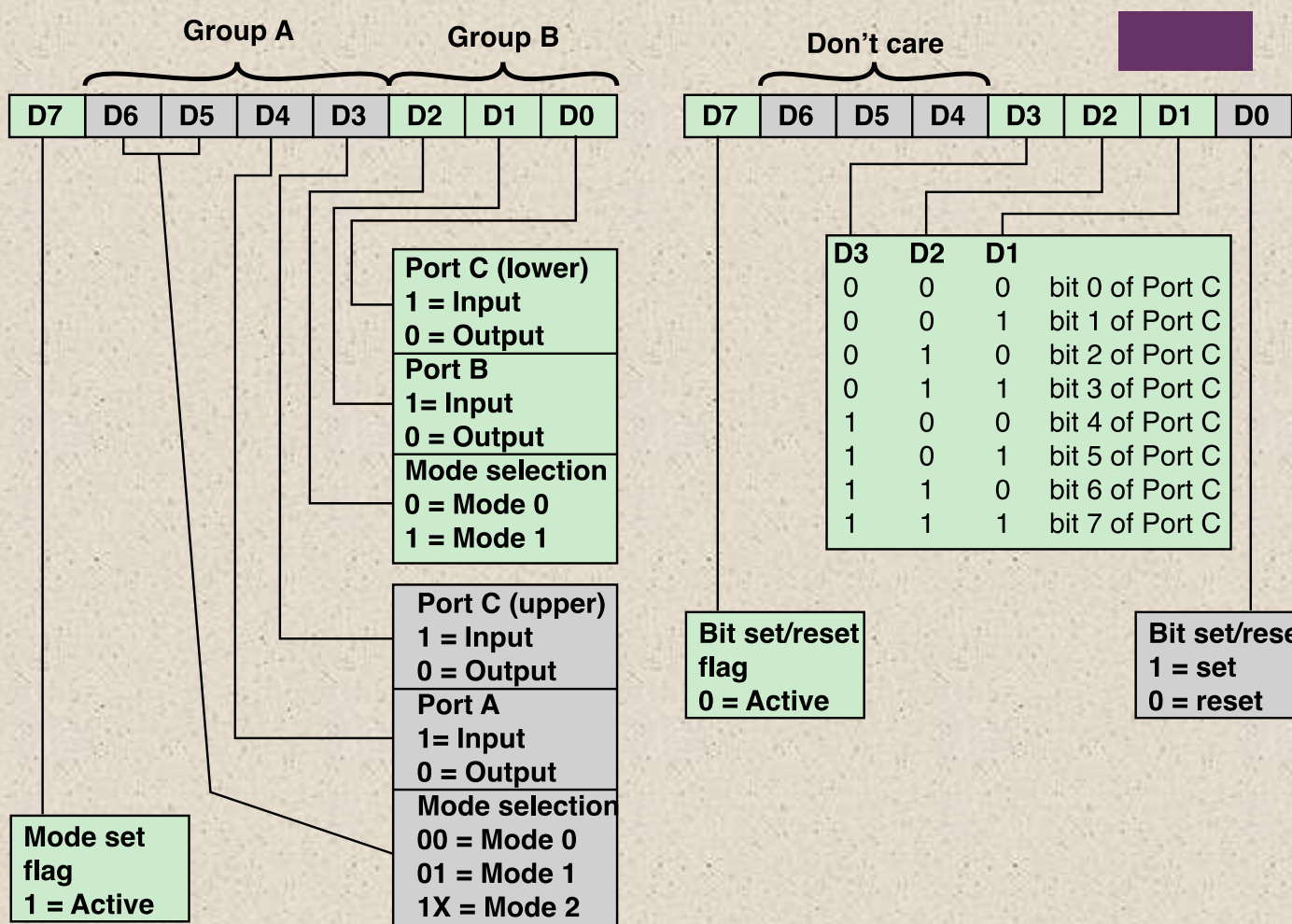
Programlı I/O ve kesmeye dayalı I/O için kullanılan bir I/O modülüne örnek olarak Intel 8255A Programlanabilir Çevre Birimi Arayüzünü ele alıyoruz. 8255A, orijinal olarak Intel 80386 işlemciyle kullanılmak üzere tasarlanmış tek yongalı, genel amaçlı bir I/O modülüdür. O zamandan beri diğer üreticiler tarafından klonlanmıştır ve yaygın olarak kullanılan bir çevresel denetleyici çipidir.

İşlemci hem A0'i hem de A1'yi 1'e ayarladığında, 8255A veri yolu üzerindeki 8 bitlik değeri bir kontrol kelimesi olarak yorumlar. İşlemci, D7 hattı 1'e ayarlanmış 8 bitlik bir kontrol kelimesini aktardığında (Şekil 7.10a), kontrol kelimesi 24 I/O hattının çalışma modunu yapılandırmak için kullanılır. Üç mod şunlardır:

■ **Mod 0:** Bu, temel I/O modudur. Sekiz harici hattan oluşan üç grup, üç adet 8 bitlik I/O bağlantı noktası olarak işlev görür

■ **Mod 1:** Bu modda, A ve B bağlantı noktaları giriş veya çıkış olarak yapılandırılabilir ve C bağlantı noktasından gelen hatlar, A ve B için kontrol hatları görevi görür. Kontrol sinyalleri iki temel amaca hizmet eder: "el sıkışma« (*handshaking*) ve kesme isteği (*interrupt request*).

**Mod 2:** Bu çift yönlü bir moddur. Bu modda, A bağlantı noktası, B bağlantı noktası üzerindeki çift yönlü trafik için giriş veya çıkış hatları olarak yapılandırılabilir ve bağlantı noktası B hatları ters yön sağlar. Yine, bağlantı noktası C hatları kontrol sinyalleri için kullanılır.

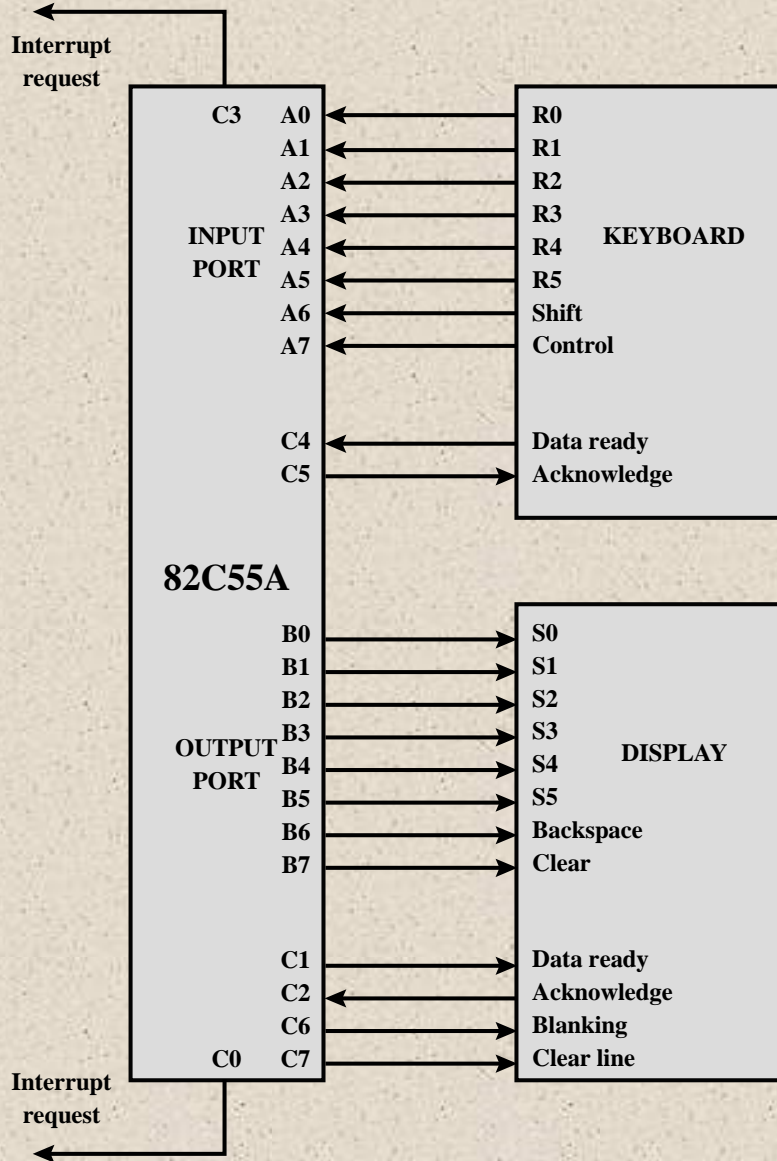


(a) Mode definition of the 8255 control register to configure the 8255

(b) Bit definitions of the 8255 control register to modify single bits of port C

Figure 7.10 The Intel 8255A Control Word





8255A, kontrol registeri aracılığıyla programlanabildiğinden, çeşitli basit çevresel cihazları kontrol etmek için kullanılabilir. Şekil 7.11, bunun bir klavye / ekran terminalini kontrol etmek için kullanımını göstermektedir.

Klavye 8 bitlik giriş sağlar. Bu bitlerden ikisi, SHIFT ve CONTROL, işlemcide yürütülen klavye kullanma programı için özel bir anlama sahiptir. Bununla birlikte, bu anlamlandırma, 8 bitlik veriyi basitçe kabul eden ve bunları sistem veri yolunda sunan 8255A'ya şeffaftır. Klavye ile kullanım için iki el sıkışma (*handshaking*) kontrol hattı sağlanmıştır.

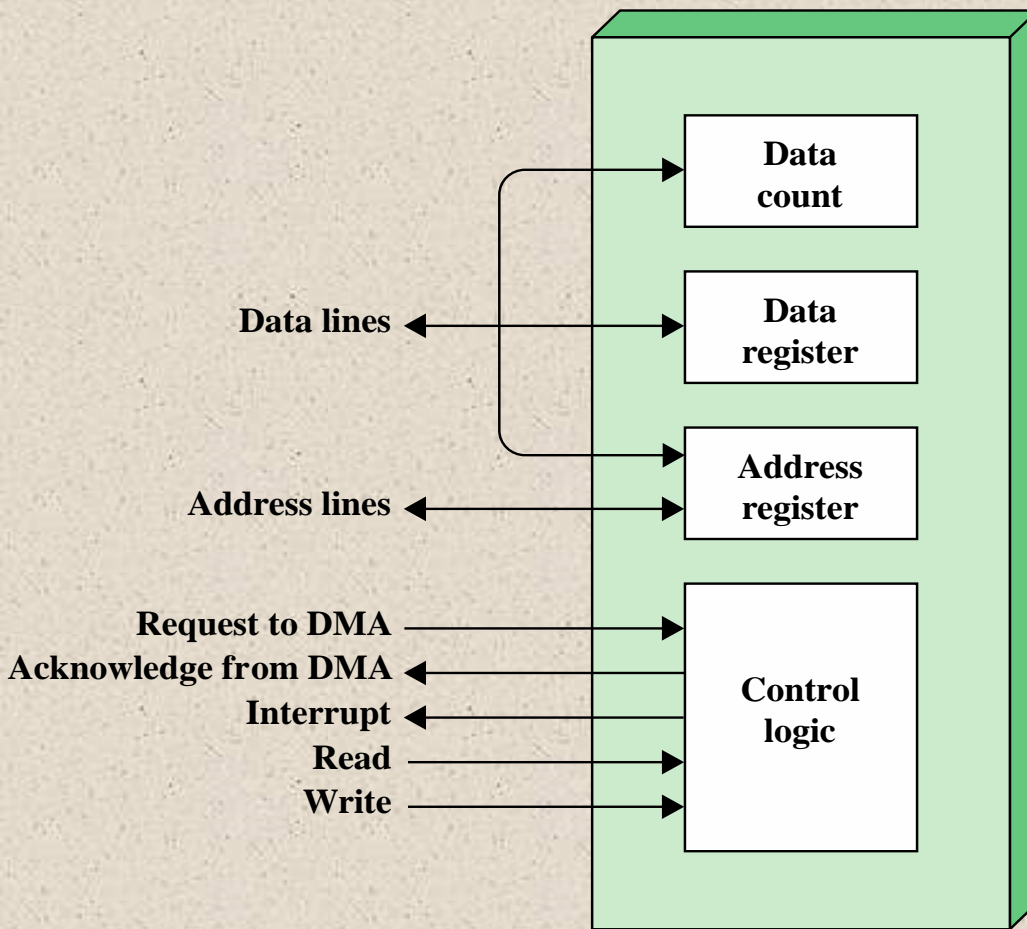
Ekran ayrıca 8 bitlik bir veri bağlantı noktasıyla bağlanır. Yine, bitlerden 8255A'ya şeffaf olan ikisinin özel anlamları vardır. İki el sıkışma hattına ek olarak, iki hat ek kontrol işlevleri sağlar.

**Figure 7.11 Keyboard/Display Interface to 82C55A**



# Drawbacks of Programmed and Interrupt-Driven I/O

- Her iki I/O biçiminin de iki doğal dezavantajı vardır:
  - 1) I/O aktarım hızı, işlemcinin bir cihazı test edip hizmet verebileceği hız ile sınırlıdır.
  - 2) İşlemci, bir I/O transferini yönetmeyle meşgul olur; her I/O aktarımı için bir dizi komut yürütülmelidir
- Büyük hacimli verilerin taşınması gerektiğinde, daha verimli bir teknik *doğrudan bellek erişimidir* (DMA)



**Figure 7.12 Typical DMA Block Diagram**

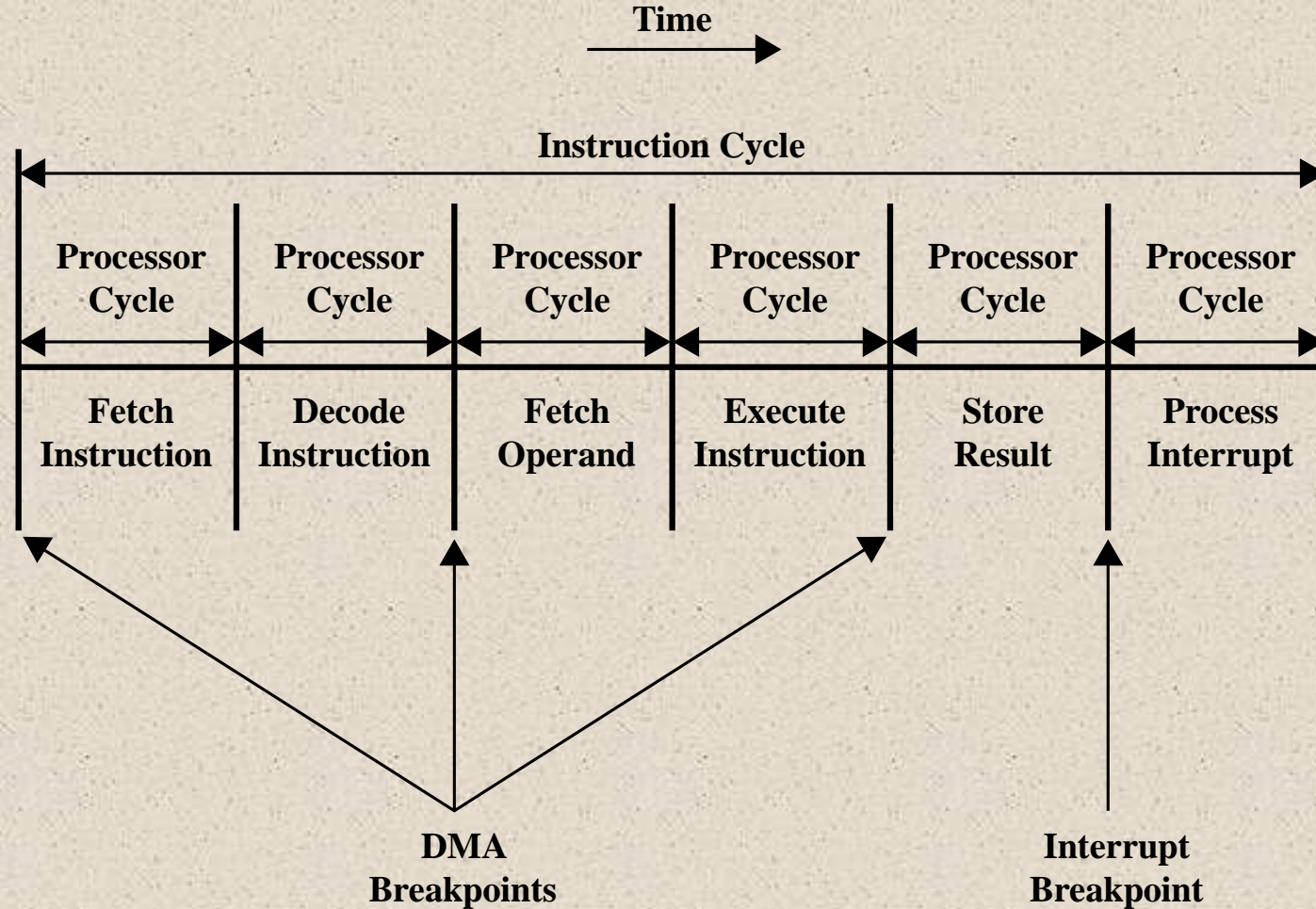
DMA, sistem veri yolunda ek bir modül içerir. DMA modülü (Şekil 7.12), işlemciyi taklit edebilir ve aslında, sistemin kontrolünü işlemciden devralabilir.

- Verileri sistem veriyolu üzerinden belleğe taşımak için bunu yapması gerekir.

Bu amaçla, DMA modülü veriyolunu yalnızca işlemcinin ihtiyacı olmadığında kullanılmalı veya işlemciyi operasyonu geçici olarak askıya almaya zorlamalıdır.

İkinci teknik daha yaygındır ve döngü çalma (***cycle stealing***) olarak adlandırılır, çünkü DMA modülü fiilen bir veri yolu döngüsünü çalar.

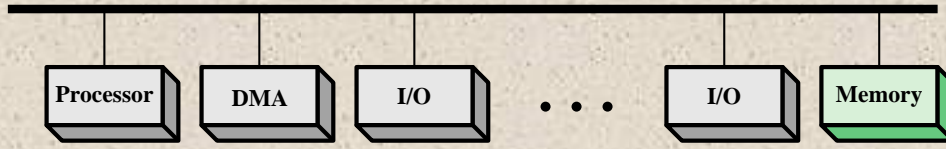
İşlemci daha sonra diğer çalışmaya devam eder. Bu I/O işlemini DMA modülüne devretmiştir. DMA modülü, işlemciden geçmeden tüm veri bloğunu her seferinde bir kelime (*word*) olmak üzere doğrudan belleğe (veya bellekten dışarıya) aktarır. Aktarım tamamlandığında, DMA modülü işlemciye bir kesme sinyali gönderir. Böylece, **işlemci yalnızca transferin başında ve sonunda yer alır.**



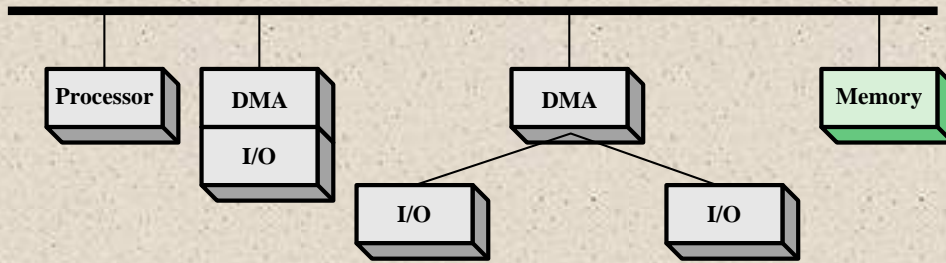
**Figure 7.13 DMA and Interrupt Breakpoints During an Instruction Cycle**

- Şekil 7.13, işlemcinin komut döngüsünde nerede askıya alınabileceğini gösterir.
- Her durumda, işlemci veri yolunu kullanması gerekmeden hemen önce askıya alınır.
- DMA modülü daha sonra bir kelimeyi aktarır ve kontrolü işlemciye döndürür.
- Bunun bir kesme olmadığını unutmayın; işlemci bir bağlamı kaydetmez (*no context saving*) ve başka bir şey yapmaz. Bunun yerine, işlemci bir veri yolu döngüsü (*one bus cycle*) için duraklar.
- Genel etki, işlemcinin daha yavaş çalışmasına neden olmaktadır.
- Bununla birlikte, çok kelimeli bir I/O aktarımı için DMA, kesmeye dayalı veya programlı I/O 'dan çok daha verimlidir.

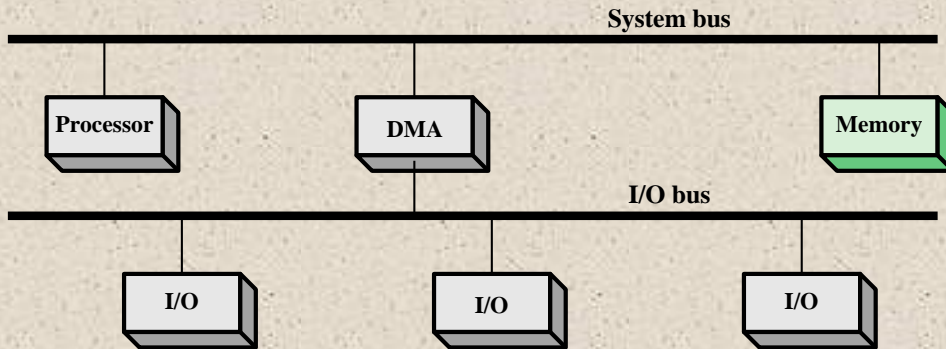




(a) Single-bus, detached DMA



(b) Single-bus, Integrated DMA-I/O



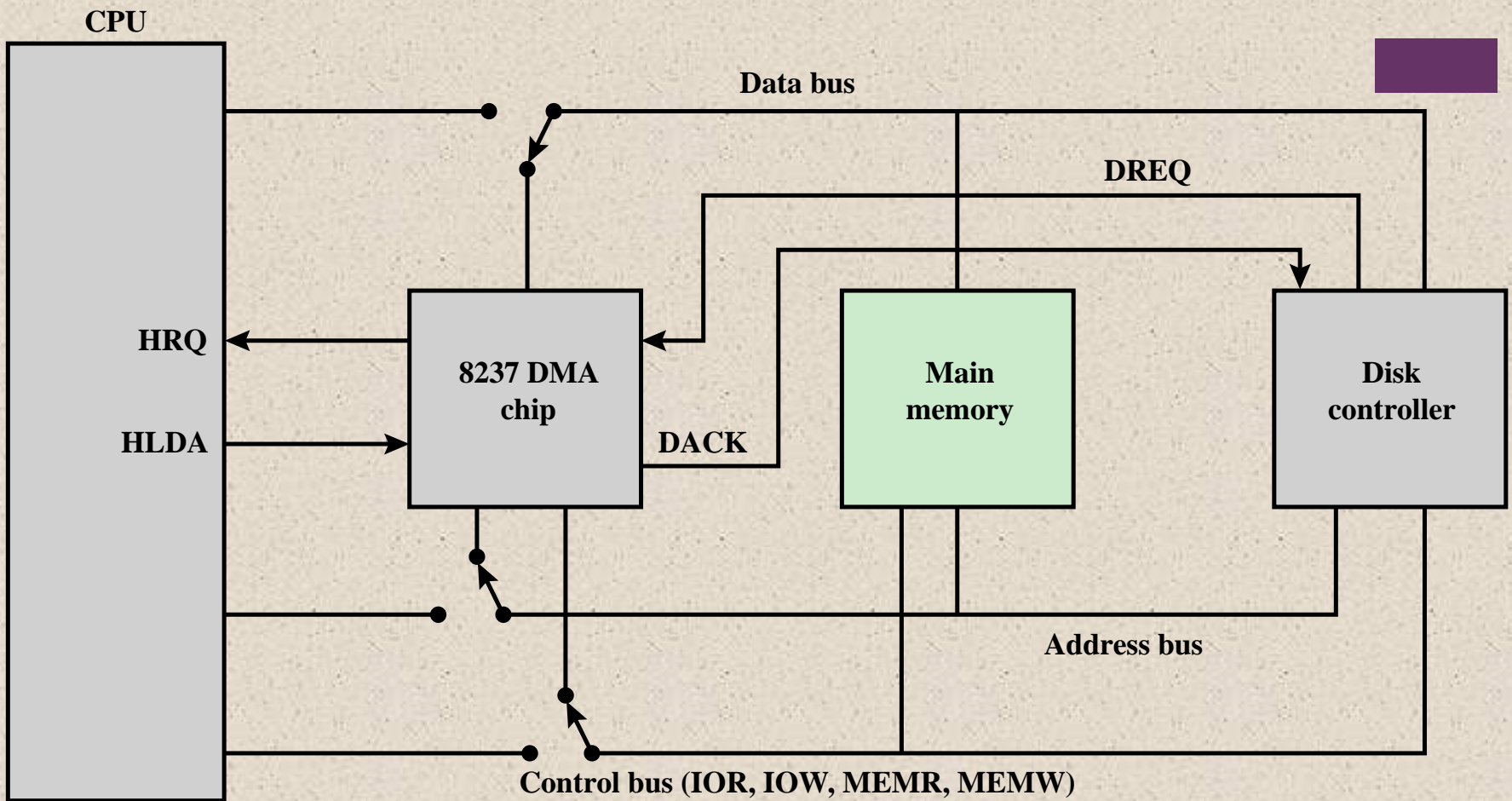
(c) I/O bus

Figure 7.14 Alternative DMA Configurations

DMA mekanizması çeşitli şekillerde yapılandırılabilir. Bazı olasılıklar Şekil 7.14'te gösterilmektedir. İlk örnekte, tüm modüller aynı sistem veriyolunu paylaşır. Yedek/vekil (*surrogate*) işlemci olarak görev yapan DMA modülü, DMA modülü aracılığıyla bellek ve bir I/O modülü arasında veri alışverişi yapmak için programlı I/O kullanır. Bu konfigürasyon, ucuz olsa da, açıkça **verimsizdir**. İşlemci kontrollü-programlı I/O 'de olduğu gibi, bir kelimenin her aktarımı iki veri yolu döngüsünü tüketir.

Gerekli veri yolu döngülerinin sayısı, DMA ve I/O fonksiyonlarının entegre edilmesiyle önemli ölçüde azaltılabilir. Şekil 7.14b'nin gösterdiği gibi, bu, DMA modülü ile sistem veriyolunu içermeyen bir veya daha fazla I/O modülü arasında bir yol olduğu anlamına gelir. DMA lojiji aslında bir I/O modülünün parçası olabilir veya bir veya daha fazla I/O modülünü kontrol eden ayrı bir modül olabilir.

Bu konsept, I/O modüllerini bir I/O veri yolu kullanarak DMA modülüne bağlayarak bir adım daha ileri götürülebilir (Şekil 7.14c). Bu, DMA modülündeki I/O arabirimlerinin sayısını bire düşürür ve kolayca genişletilebilir bir konfigürasyon sağlar. Her iki durumda da (Şekil 7.14b ve c), DMA modülünün işlemci ve bellekle paylaştığı sistem veriyolu, DMA modülü tarafından yalnızca bellekle veri alışverişi yapmak için kullanılır. **DMA ve I / O modülleri arasındaki veri alışverişi, sistem veriyolunun dışında gerçekleşir.**



DACK = DMA acknowledge  
DREQ = DMA request  
HLDA = HOLD acknowledge  
HRQ = HOLD request

**Figure 7.15 8237 DMA Usage of System Bus**

Intel 8237A DMA denetleyicisi, DMA yeteneği sağlamak için 80 x 86 işlemci ailesine ve DRAM belleğine arabirim sağlar. Şekil 7.15, DMA modülünün konumunu gösterir.

DMA modülünün veri aktarmak için sistem veri yollarını (veri, adres ve kontrol) kullanması gerektiğinde, işlemciye HOLD adı verilen bir sinyal gönderir.

İşlemci, DMA modülünün veriyolları kullanabileceğini belirten HLDA (onaylama) sinyaliyle yanıt verir. Örneğin, DMA modülü bir veri bloğunu bellekten diske aktaracaksa, aşağıdakileri yapacaktır:

1. Çevresel aygıt (disk denetleyicisi gibi), DREQ'yu (DMA talebi) yükseğe çekerek DMA hizmetini isteyecektir.
2. DMA, HRQ'sunu (tutma isteği) yükseğe çeker ve CPU'ya HOLD pini aracılığıyla veriyolları kullanması gerektiğini bildirir.
3. CPU, mevcut veri yolu döngüsünü (mevcut komut değil) bitirecek ve DMA talebine HDLA'sına yüksek yaparak (onaylama) yanıt verecek, böylece 8237 DMA'ya devam edebileceğini ve görevi gerçekleştirmek için bus'ları kullanabileceğini söyleyecektir. DMA görevini yerine getirdiği süre zarfında HOLD aktif yüksek kalmalıdır.
4. DMA, çevresel cihaza verileri aktarmaya başlayacağını söyleyen DACK'i (DMA onayı) etkinleştirecektir.
5. DMA, bloğun ilk baytının adresini adres veriyoluna koyarak ve MEMR'yi etkinleştirerek verileri bellekten çevre birimine aktarmaya başlar, böylece baytı bellekten veri yoluna okur; daha sonra çevre birimine yazmak için IOW'u etkinleştirir. Daha sonra DMA, sayacı azaltır ve adres işaretçisini artırır ve bu işlemi, sayı sıfıra ulaşana ve görev bitene kadar tekrarlar.
6. DMA işini bitirdikten sonra HRQ'yu devre dışı bırakacak ve CPU'ya veriyolları üzerinde kontrolü yeniden kazanabileceği sinyalini verecektir.



# Fly-By DMA Controller

- + DMA veri aktarmak için veri yollarını kullanırken işlemci boştaadır. Benzer şekilde, işlemci veri yolunu kullanırken DMA boştaadır. 8237 DMA, *fly-by DMA controller* olarak bilinir. Bu, bir konumdan diğerine taşınan verilerin DMA yongasından geçmediği ve DMA yongasında saklanmadığı anlamına gelir.

Veriler çipten geçmez ve DMA çipinde depolanmaz

- DMA yalnızca I/O bağlantı noktası ve bellek arasında
- İki I/O bağlantı noktası veya iki bellek konumu arasında değil

Ancak, register yoluyla bellekten belleğe (***memory-to-memory***) yapabilir

8237, dört DMA kanalı içerir

- Bağımsız olarak programlanabilir
- Herhangi biri aktifleştirilebilir
- 0, 1, 2 ve 3 numaralı





Table 7.2

Intel  
8237A  
Registers

Bit	Command	Status	Mode	Single Mask	All Mask
D0	Memory-to-memory E/D	Channel 0 has reached TC	Channel select	Select channel mask bit	Clear/set channel 0 mask bit
D1	Channel 0 address hold E/D	Channel 1 has reached TC			Clear/set channel 1 mask bit
D2	Controller E/D	Channel 2 has reached TC		Clear/set mask bit	Clear/set channel 2 mask bit
D3	Normal/compressed timing	Channel 3 has reached TC	Verify/write/read transfer	Not used	Clear/set channel 3 mask bit
D4	Fixed/rotating priority	Channel 0 request	Auto-initialization E/D		Not used
D5	Late/extended write selection	Channel 0 request	Address increment/decrement select		
D6	DREQ sense active high/low	Channel 0 request			
D7	DACK sense active high/low	Channel 0 request	Demand/single/block/cascade mode select		

E/D = enable/disable

TC = terminal count

# + Direct Cache Access (DCA)

DMA, çevresel aygıtlar ile I/O performansını artırmanın etkili bir yolunu kanıtlamıştır. Bununla birlikte, network I/O için veri hızlarındaki çarpıcı artışlarda DMA, artan talebi karşılayacak şekilde ölçeklenemez.

- DMA, network I /O veri hızlarındaki önemli artışlar nedeniyle artan talebi karşılayacak şekilde ölçeklenemiyor
- Veri tabanı sunucularından ve diğer yüksek performanslı sistemlerden büyük miktarlarda veri aktarımının üstesinden gelmek için öncelikle 10 Gb/sn ve 100 Gb/sn Ethernet switch'lerinin yaygın kullanımından talep gelmektedir.
- Başka bir trafik kaynağı da gigabit aralığındaki Wi-Fi'den geliyor
- 3,2 Gb/sn ve 6,76 Gb/sn işleyen network Wi-Fi aygıtları, yaygın olarak kullanılabilir hale geliyor ve iş (enterprise) sistemlerinde talep yaratıyor

# + Direct Cache Access (DCA)



Bu bölüm, doğrudan önbellek erişimi (DCA) olarak bilinen bir teknik olan I/O fonksiyonunun önbelleğe doğrudan erişime sahip olmasının performansı nasıl artırabileceğini gösterir.

Biz sadece ana belleğe en yakın olan ve son seviye önbellek olarak adlandırılan önbellekle ilgileniyoruz. Bazı sistemlerde bu bir L2 önbelleği, diğerlerinde bir L3 önbelleği olacaktır.

Başlangıç olarak, modern çok çekirdekli sistemlerin DMA performansını artırmak için çip üzerinde paylaşılan önbelleği kullanma şeklini açıklıyoruz.

Bu yaklaşım, DMA fonksiyonunun son seviye önbelleğe doğrudan erişmesini sağlamayı içerir. Daha sonra, yüksek hızlı network trafiği işlendiğinde ortaya çıkan önbellekle ilgili performans sorunlarını inceleyeceğiz.

Daha sonra, network protokolü işleme performansını artırmak için tasarlanmış DCA için birkaç farklı stratejiye bakacağız.

Son olarak, bu bölümde, Doğrudan Veri I/O olarak adlandırılan, Intel tarafından uygulanan bir DCA yaklaşımı açıklanmaktadır.



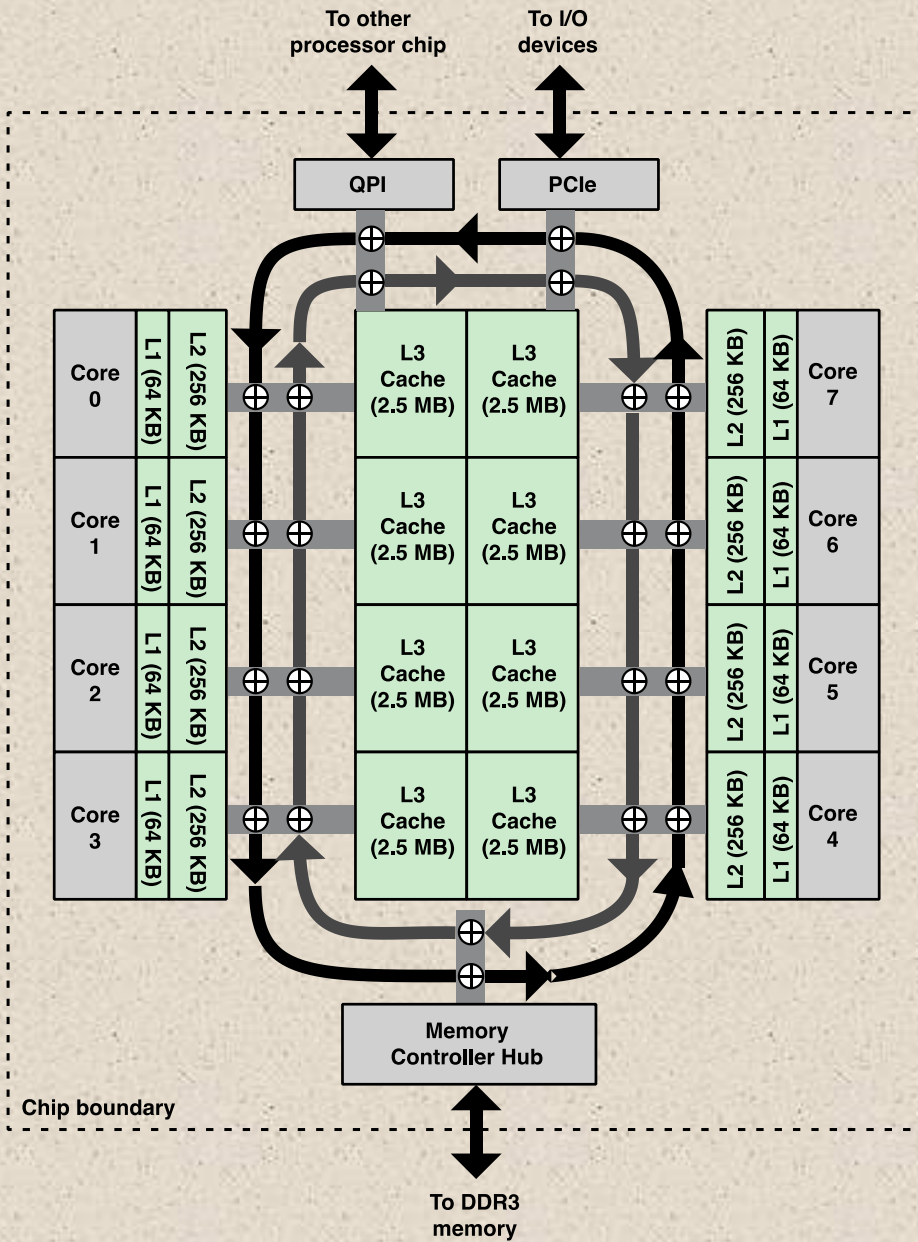


Figure 7.16 Xeon E5-2600/4600 Chip Architecture

Intel Xeon, Intel'in sunucularda, yüksek performanslı iş istasyonlarında ve süper bilgisayarlarda kullanılan yüksek kaliteli, yüksek performanslı işlemci ailesidir. Xeon ailesinin üyelerinin çoğu, Şekil 7.16'da Xeon E5-2600 / 4600 için gösterildiği gibi bir halka şeklinde ara bağlantı sistemi (ring interconnect system) kullanır.

E5-2600 / 4600, tek bir yonga üzerinde sekiz adede kadar çekirdekle yapılandırılabilir. Her çekirdeğin özel L1 ve L2 önbellekleri vardır. 20 MB'a kadar paylaşılan bir L3 önbellek var. L3 önbelleği, her bir çekirdek önbelleğinin tamamını ele alabilmesine rağmen, her bir çekirdek ile ilişkilendirilmiş dilimlere bölünmüştür. Ayrıca, her dilimin kendi önbellek pipeline'ı vardır, böylece istekler dilimlere paralel olarak gönderilebilir.

Çift yönlü (*bidirectional*) yüksek hızlı halka ara bağlantı; çekirdekleri (cores), son seviye önbelleği (last-level cache), PCIe'yi ve tümleşik bellek denetleyicisini (integrated memory controller -IMC) birbirine bağlar.



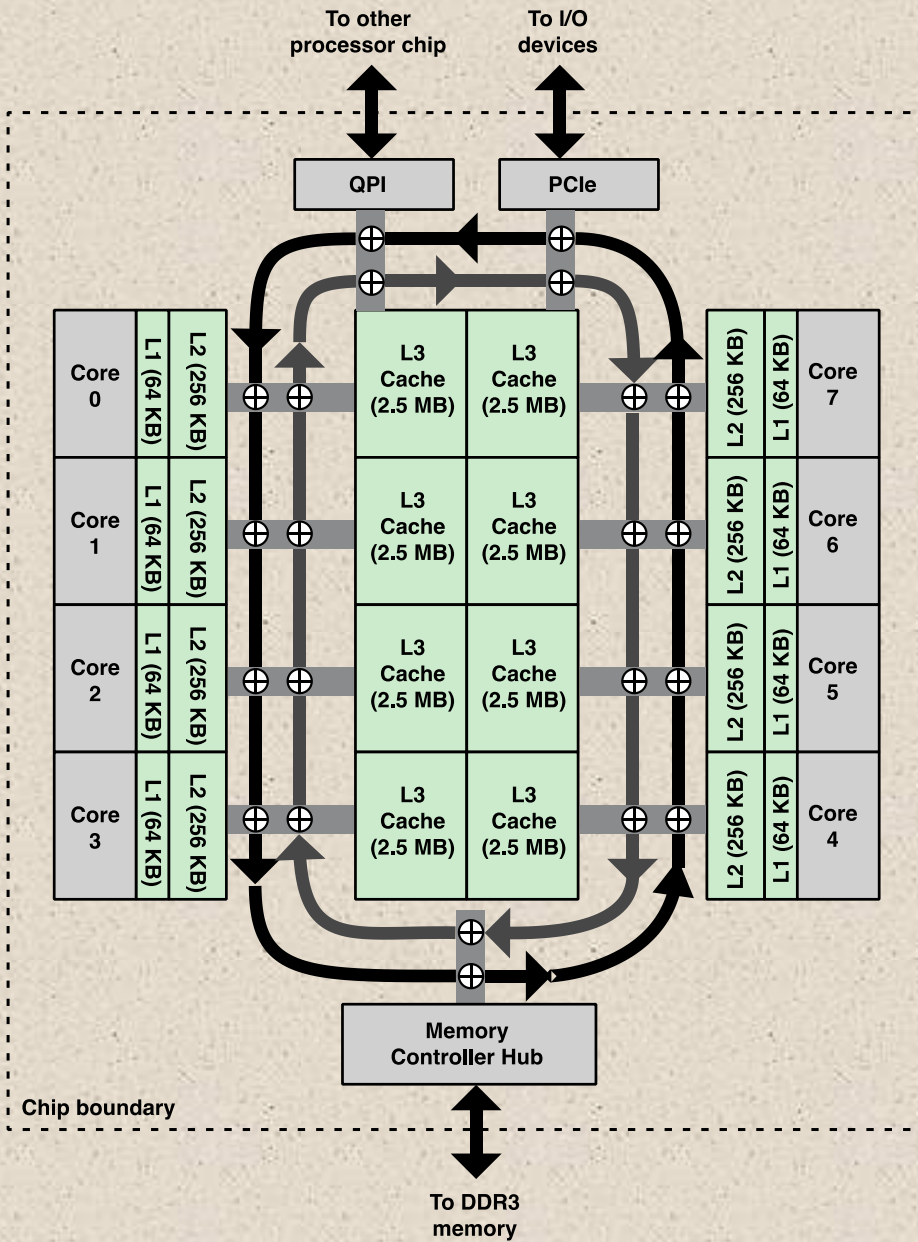


Figure 7.16 Xeon E5-2600/4600 Chip Architecture

Temelde halka yapısı şu şekilde çalışır:

1. Çift yönlü halkaya (QPI, PCIe, L3 önbellek, L2 önbellek) eklenen her bileşen bir halka temsilcisi olarak kabul edilir ve halka temsilcisi lojini uygular.
2. Halka temsilcisi, zaman dilimleri biçiminde halkaya erişim talep etmek ve tahsis etmek için dağıtılmış bir protokol aracılığıyla işbirliği yapar.
3. Bir temsilci gönderilecek veriye sahip olduğunda, hedefe giden en kısa yolla sonuçlanan halka yönünü seçer ve bir zamanlama slotu mevcut olduğunda iletir

Halka mimarisi iyi performans sağlar ve bir birden çok çekirdek için bir dereceye kadar iyi ölçeklenir. Daha fazla sayıda çekirdeğe sahip sistemler için, her bir halkanın bazı çekirdekleri desteklediği birden çok halka (multiple rings) kullanılır.

- Geleneksel DMA işleminde, veri yolu, halka veya QPI noktadan noktaya matris gibi sistem ara bağlantı yapısı aracılığıyla ana bellek ile bir I/O cihazı arasında veri alışverişi yapılır.
- Bu nedenle, örneğin, Xeon E5-2600 / 4600 geleneksel bir DMA tekniğini kullanırsa, «**output**» işlemi aşağıdaki gibi ilerleyecektir. Çekirdek üzerinde çalışan bir I/O sürücüsü, aktarılacak verileri içeren ana bellekteki buffer konumu ve boyutu ile I/O denetleyicisine (Şekil 7.16'da PCIe olarak etiketlenmiştir) bir I/O komutu gönderir.
- I/O denetleyicisi, bellek denetleyici hub'ına (MCH) yönlendirilen bir okuma isteği yayınlar. MCH, DDR3 belleğindeki verilere erişir ve I/O denetleyicisine teslim edilmek üzere sistem halkasına yerleştirir
  - L3 ön belleği bu işleme müdahil değildir ve bir veya daha fazla yonga dışı bellek okuması gereklidir.
- Benzer şekilde, «**input**» işlemi için veriler I/O denetleyicisinden gelir ve sistem halkası üzerinden MCH'ye iletilir ve ana belleğe yazılır.
  - MCH ayrıca güncellenmiş bellek konumlarına karşılık gelen herhangi bir L3 ön bellek satırını geçersiz kılmalıdır.
- Bu durumda, bir veya daha fazla yonga dışı bellek yazma işlemi gerekir. Ayrıca, bir uygulama yeni verilere erişmek isterse, bir ana bellek okuma işlemi gerekir.

Büyük miktarlarda son seviye önbelleğin mevcudiyetiyle, daha verimli bir teknik mümkündür ve Xeon E5-2600 / 4600 tarafından kullanılır.

- «**output**» işlemi için, I/O denetleyicisi bir okuma talebi yayınladığında, MCH ilk olarak verilerin L3 önbelleğinde olup olmadığını kontrol eder.
- Bir uygulama son dönemde, dışarıya aktarılacak (*to be output*) bellek bloğuna veri yazmışsa, durum muhtemelen budur. Bu durumda MCH, verileri L3 önbelleğinden I/O denetleyicisine yönlendirir; ana bellek erişimine gerek yoktur.
- Bununla birlikte, verilerin önbellekten çıkarılmasına da neden olur, yani bir I/O cihazı tarafından okuma eylemi verilerin tahliye edilmesine neden olur.
- Böylece, I/O işlemi, ana bellek erişimi gerektirmediğinden verimli bir şekilde ilerler.
- Xeon E5-2600 / 4600 üzerindeki «**input**» işlemi , önceki paragrafta açıklandığı gibi çalışır; L3 önbelleği müdahil değildir. Bu nedenle, performans iyileştirme yalnızca «**output**» işlemlerini içerir.



# Cache-Related Performance Issues

Network trafiği, paketler veya protokol veri birimleri adı verilen bir dizi protokol bloğu biçiminde iletilir.

En düşük (*lowest*) veya bağlantı (*link*) seviyesi protokolü tipik olarak Ethernet'tir, böylece her gelen ve giden veri bloğu, yük olarak daha yüksek seviyeli protokol paketini içeren bir Ethernet paketinden oluşur.

Daha yüksek seviyeli protokoller genellikle Ethernet üzerinde çalışan İnternet Protokolü (IP) ve IP'nin üstünde çalışan İletim Kontrol Protokolü (TCP) 'dir.

Ethernet yükü (*payload*), bir TCP başlığı ve bir IP başlığı olan bir veri bloğundan oluşur.

Giden veriler (*outgoing data*) için, Ethernet paketleri, I/O denetleyicisi veya ağ arabirim denetleyicisi (NIC) gibi bir çevresel bileşende oluşturulur.

Gelen trafik (*incoming traffic*) için, I/O denetleyicisi Ethernet bilgilerini çıkarır ve TCP/IP paketini ana bilgisayar (*host*) CPU'suna gönderir





# Cache-Related Performance Issues

Hem giden hem de gelen trafik için çekirdek, ana bellek ve önbellek dahil edilir

Bir DMA düzeninde, bir uygulama veri iletmek istediğinde, bu verileri ana bellekte uygulama tarafından atanan bir arabelleğe (*buffer*) yerleştirir.

- Çekirdek bunu ana bellekteki bir sistem arabelleğine aktarır ve sistem belleğinde de arabelleğe alınan gerekli TCP ve IP başlıklarını oluşturur.
- Paket daha sonra NIC aracılığıyla aktarılacak üzere DMA aracılığıyla alınır
- Bu aktivite sadece ana belleği değil, aynı zamanda önbelleği (cache) de meşgul eder
- Gelen trafik için sistem ve uygulama arabellekleri arasında benzer aktarımlar gereklidir

# + Packet Traffic Steps:

## Incoming

- Packet arrives
- DMA
- NIC interrupts host
- Retrieve descriptors and headers
- Cache miss occurs
- Header is processed
- Payload transferred

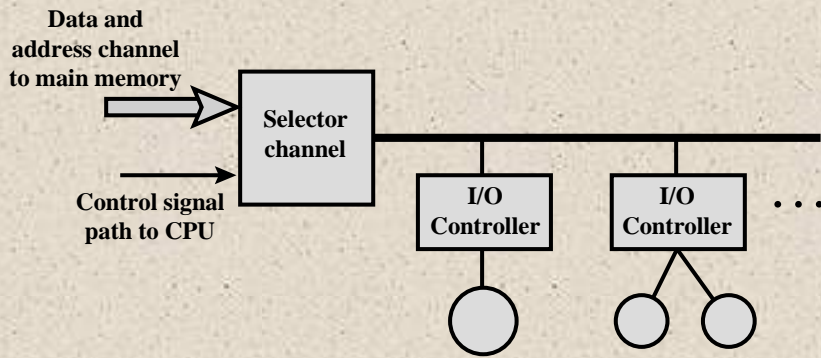
## Outgoing

- Packet transfer requested
- Packet created
- Output operation invoked
- DMA transfer
- NIC signals completion
- Driver frees buffer

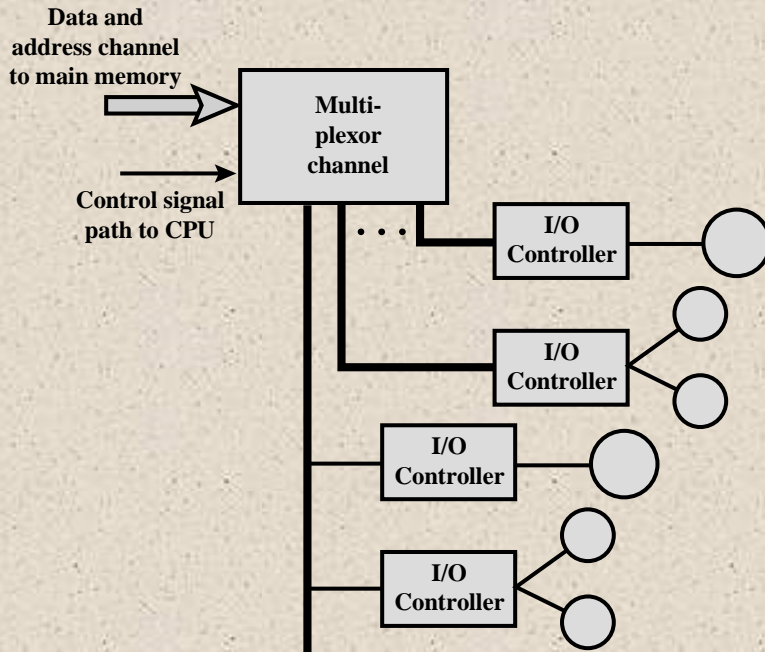
# + Evolution of the I/O Function

Bilgisayar sistemleri geliştikçe, tekil bileşenlerin kapsamlılığı ve karmaşıklığı artmaktadır. Bu, hiçbir yerde I/O fonksiyonlarından daha belirgin değildir. Bu evrimin bir kısmını zaten gördük. Bu evrimsel adımlar şu şekilde özetlenebilir:

1. CPU, bir çevre birimini doğrudan kontrol eder.
2. Bir denetleyici veya I/O modülü eklenir. CPU, programlı I/O'yu kesintisiz olarak kullanır.
3. Adım 2 ile aynı konfigürasyon kullanılır, ancak şimdi kesmeler (*interrupts*) kullanılmaktadır. CPU'nun, bir I/O işleminin gerçekleştirilmesini beklemek için zaman harcamasına gerek yoktur, bu da verimliliği artırır.
4. I/O modülüne, DMA aracılığıyla belleğe doğrudan erişim verilir. Aktarımın başlangıcı ve bitişi dışında, artık CPU'yu dahil etmeden bir veri bloğunu belleğe veya bellekten taşıyabilir.
5. I/O modülü, I/O için özel olarak hazırlanmış bir komut seti ile kendi başına bir işlemci olacak şekilde geliştirilmiştir.
6. I/O modülünün kendine ait bir yerel belleği vardır ve aslında kendi başına bir bilgisayardır. Bu mimari ile çok sayıda I/O cihazı minimum CPU müdahalesi ile kontrol edilebilir.



(a) Selector



(b) Multiplexor

Figure 7.18 I/O Channel Architecture

I/O kanalı (*channel*), DMA konseptinin bir uzantısını temsil eder. Bir I/O kanalı, I/O komutlarını yürütme yeteneğine sahiptir, bu da I/O işlemleri üzerinde tam kontrol sağlar.

Bu tür cihazlara sahip bir bilgisayar sisteminde CPU, I/O komutlarını yürütmez. Bu tür komutlar, I/O kanalının kendisinde özel amaçlı bir işlemci tarafından yürütülmek üzere ana bellekte saklanır.

Şekil 7.18'de gösterildiği gibi iki tür I/O kanalı yaygındır. Bir seçici (*selector*) kanal, birden çok yüksek hızlı cihazı kontrol eder ve herhangi bir zamanda, bu cihazlardan biriyle veri aktarımına adanmıştır. Böylece, I/O kanalı bir cihazı seçer ve veri aktarımını etkiler. Her cihaz veya küçük bir cihaz seti, bahsettiğimiz I/O modüllerine çok benzeyen bir denetleyici veya I/O modülü tarafından yönetilir. Bu nedenle, I/O kanalı, bu I/O denetleyicilerini kontrol etmede CPU yerine hizmet eder. Çoklayıcı kanal (*multiplexor channel*), aynı anda birden fazla cihazla I/O'yu işleyebilir.



# + Universal Serial Bus (USB)

- Çevresel aygıt bağlantıları için yaygın olarak kullanılır
- Klavye ve işaretleme aygıtları gibi daha düşük hızlı aygıtlar için varsayılan arabirimdir,
- ancak aynı zamanda yazıcılar, disk sürücüler ve ağ bağdaştırıcıları dahil olmak üzere yüksek hızlı I/O için de yaygın olarak kullanılır.
- Birden çok nesil geçmiştir
  - USB 1.0
    - 1.5 Mbps'lik Düşük Hızlı (*Low Speed*) veri hızı ve 12 Mbps'lik Tam Hız (*Full Speed*)
  - USB 2.0
    - 480 Mbps veri hızı sağlar
  - USB 3.0
    - USB 2.0 veri yolu ile paralel olarak *SuperSpeed* adı verilen daha yüksek hızlı veri yolu
    - SuperSpeed'in sinyal hızı 5 Gbps'dir, ancak sinyal ek yükü (***signaling overhead***) nedeniyle kullanılabilir veri hızı 4 Gbps'ye kadardır
  - USB 3.1
    - *SuperSpeed+* adlı daha hızlı bir aktarım modu içerir
    - Bu aktarım modu, 10 Gbps'lik bir sinyal hızına ve 9.7 Gbps'lik teorik olarak kullanılabilir bir veri hızına ulaşır
- Hiyerarşik ağaç topolojisine sahip yerel bir ağ oluşturmak için cihazlara bağlanan bir kök ana bilgisayar denetleyicisi (*root host controller*) tarafından kontrol edilir

# + FireWire Serial Bus

- Kişisel bilgisayarlar, iş istasyonları ve sunucular gibi daha küçük sistemlerde kullanılmak üzere küçük bilgisayar sistemi arabirimine (small computer system interface-SCSI) alternatif olarak geliştirilmiştir
- Amaç, ana bilgisayar (*mainframe*) ve süper bilgisayar sistemleri için geliştirilen hantal ve pahalı I/O kanal teknolojilerinden kaçınırken yüksek I/O hızları için artan talepleri karşılamaktır.
- IEEE standard 1394, for a High Performance Serial Bus
- 63 cihaza kadar tek bir bağlantı noktasına bağlanan bir papatya zinciri (*daisy chain*) konfigürasyonu kullanır
- 1022 tane FireWire bus'ı köprü kullanılarak birbirine bağlanabilir (Çok sayıda çevre aygıtını destekleyecek şekilde)
- Bilgisayar sistemini kapatmak veya sistemi yeniden yapılandırmak zorunda kalmadan çevre birimlerini bağlamayı ve bağlantısını kesmeyi mümkün kılan çalışırken takmayı (***hot plugging***) sağlar
- Otomatik konfigürasyon sağlar (cihaz kimliklerini manuel olarak ayarlamak veya cihazların göreceli konumlarıyla ilgilenmek gerekli değildir.)
- Sonlandırma yoktur (*No terminations*) ve sistem, adresleri atamak için otomatik olarak bir yapılandırma fonksiyonu gerçekleştirir

# +SCSI

- Small Computer System Interface
- Çevre aygıtlarını (disk, modem, printer, vb.) küçük ve orta ölçekli bilgisayarlara (*small and medium-sized computers*) bağlamak için bir zamanlar yaygın olan bir standart
- Daha küçük sistemlerde popülerliğini USB'ye ve FireWire'a kaptırdı
- Yüksek hızlı sürümleri, enterprise sistemlerde kitle bellek desteği (*mass memory support*) için popüler olmaya devam ediyor
  - Örneğin, IBM zEnterprise EC12 ve diğer IBM anabilgisayarları SCSI için destek sunar ve bir dizi Seagate sabit disk sistemi SCSI kullanır.
- Fiziksel organizasyon, standardın nesline bağlı olarak 16 veya 32 cihaza kadar destekleyebilen paylaşılan bir veri yoludur (*shared bus*).
  - Veriyolu, önceki nesillerde 16 bitlik ve sonraki nesillerde 32 bitlik veriyolu genişliğiyle seri yerine paralel iletim sağlar
  - Hızlar, orijinal SCSI-1 spesifikasyonunda 5 Mbps'den SCSI-3 U3'te 160 Mbps'ye kadar değişir







# Thunderbolt



- Genel amaçlı kullanıma uygun hale gelen en yeni ve en hızlı çevre birimi bağlantı teknolojisi
- Intel tarafından Apple'ın işbirliğiyle geliştirildi
- Bu teknoloji verileri, videoyu, sesi ve gücü; sabit sürücüler, RAID dizileri, video yakalama kutuları (*video-capture boxes*) ve ağ arabirimleri gibi çevre birimleri için tek bir yüksek hızlı bağlantıda (*high-speed connection*) birleştirir
- Her yönde 10 Gbps'ye kadar girdi-çıktı (*throughput*) ve bağlı çevre birimlerine 10 Watt'a kadar güç sağlar





# InfiniBand

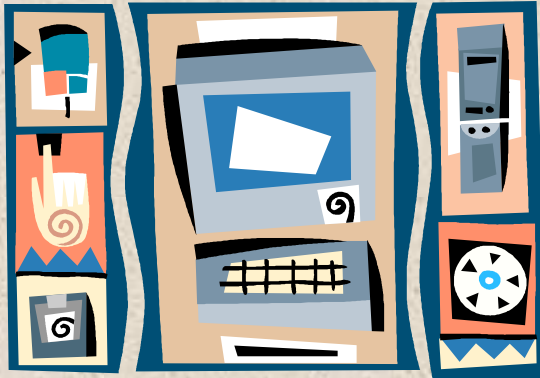


- Üst düzey sunucu pazarını (*high-end server market*) hedefleyen I/O spesifikasyonu
- İlk sürüm 2001'in başlarında yayınlandı
- IBM zEnterprise serisi mainframe bilgisayarlar tarafından büyük ölçüde güvenilen bir teknoloji
- Bu standart; işlemciler ve akıllı I/O cihazları arasındaki veri akışı için bir mimari ve spesifikasyonları tanımlar
- Depolama alanı ağı (*storage area networking*) ve diğer büyük depolama yapılandırmaları için popüler bir arayüz haline geldi
- Sunucuların, uzak depolamanın ve diğer ağ cihazlarının merkezi bir anahtar ve bağlantı yapısına bağlanmasını sağlar
- Anahtar tabanlı mimari (*switch-based architecture*), 64.000 adede kadar sunucuyu, depolama sistemini ve ağ cihazını bağlayabilir



## PCI Express

- Çok çeşitli tür ve hızlardaki çevre birimlerini bağlamak için yüksek hızlı veri yolu sistemi



## SATA

- Serial Advanced Technology Attachment
- Disk depolama sistemleri arayüzü
- Cihaz başına maksimum 300 Mbps ile birlikte 6 Gbps'ye kadar veri hızları sağlar
- Masaüstü bilgisayarlarda ve endüstriyel ve gömülü uygulamalarda yaygın olarak kullanılır

# + Ethernet



- Baskın kablolu ağ teknolojisi
- 100 Gbps'ye kadar veri hızlarını ve birkaç metreden onlarca km'ye kadar olan mesafeleri destekleyecek şekilde gelişti
- Büyük ve küçük kuruluşlarda kişisel bilgisayarları, iş istasyonlarını, sunucuları ve büyük veri depolama cihazlarını desteklemek için gerekli hale geldi
- Deneysel bir veriyolu tabanlı (*bus-based*) 3-Mbps sistem olarak başladı

- Veriyolu tabanlıdan anahtar tabanlıya geçti (*bus-based to switch-based*)
  - Veri hızı periyodik olarak büyüklük sırasına göre artmıştır
  - Tüm cihazların doğrudan anahtara bağlı olduğu merkezi bir anahtar vardır
- Ethernet sistemleri şu anda 100 Gbps'ye kadar hızlarda mevcuttur

- 1983: 10 Mbps (megabit per second, million bits per second)
- 1995: 100 Mbps
- 1998: 1 Gbps (gigabit per second, billion bits per second)
- 2003: 10 Gbps
- 2010: 40 Gbps and 100 Gbps



# Wi-Fi

- Baskın kablosuz İnternet erişim teknolojisi
- Artık bilgisayarları, tabletleri, akıllı telefonları ve video kamera, TV'ler ve termostatlar gibi diğer elektronik cihazları birbirine bağlar
- İşletmelerde, çalışan üretkenliğini ve ağ etkinliğini artırmanın temel bir yolu haline geldi
- Halka açık erişim noktaları (*Public hotspots*), çoğu halka açık yerde ücretsiz İnternet erişimi sağlamak için önemli ölçüde genişledi

- Anten teknolojisi, kablosuz iletim teknikleri ve kablosuz protokol tasarımı geliştikçe, IEEE 802.11 komitesi daha yüksek hızlarda Wi-Fi'nin yeni sürümleri için standartlar getirebildi.
- Mevcut sürüm, maksimum 3,2 Gbps veri hızına sahip 802.11ac (2014)





# + Summary

## Chapter 7

- External devices
  - Keyboard/monitor
  - Disk drive
- I/O modules
  - Module function
  - I/O module structure
- Programmed I/O
  - Overview of programmed I/O
  - I/O commands/instructions
- Direct memory access
  - Drawbacks of programmed and interrupt-driven I/O
  - DMA function
  - Intel 8237A DMA controller

## Input/Output

- Interrupt-driven I/O
  - Interrupt processing
  - Design issues
  - Intel 82C59A interrupt controller
  - Intel 82C55A programmable peripheral interface
- Direct Cache Access
  - DMA using shared last-level cache
  - Cache-related performance issues
  - Direct cache access strategies
  - Direct data I/O
- I/O channels and processors
  - The evolution of the I/O function
  - Characteristics of I/O channels