

I/O Çeşitleri

- Kesme odaklı (Interrupt-driven) I/O
- Doğrudan Bellek Erişimi
(Direct memory access - DMA)

Interrupt-Driven I/O

Kesme Odaklı G/Ç

Polling deki tüm problemler kesmeli G/Ç kullanılarak ortadan kaldırılabilir:

- Herhangi bir servis isteği olması durumunda harici cihazın bizzat kendisi CPU 'yu uyarır; CPU 'nun sorgulamasına gerek yoktur
- I/O işlemleri bir aygıt tarafından başlatılır.
- Aygıt ya da onun I/O modülü, CPU da bir kesme oluşturmak için bir sinyal oluşturur.
- Bu sinyaller kesme hattı-*interrupt lines* olarak adlandırılır.
- Genel bir CPU 8 ila 16 kesme sinyalini destekler.
- Genel isimleri: IRQ1, IRQ2, IRQ3, vs.

Interrupt Servisi

- Bir kesme meydana geldiğinde (kabul edilir), çalışan mevcut program askıya alınır.
- Özel bir rutin kesme servisi işletilir.
- Ardından, kesilen program sürdürülür.
- Bu rutin kesme, kesme hizmet programı-*interrupt service routine (ISR)* ya da kesme işleyicisi-interrupt handler olarak adlandırılır.

Registerları Kaydetme

- Kesilen programın sürdürülebilmesi için, CPU durumu ve data register kaydedilmelidir. (çünkü ISR süresince değiştirilecek)
- ISR çalıştırılmadan önce kaydedilir.
- ISR çalıştırdıktan sonra yeniden geri yüklenir.

Interruptların Kullanımı

- CPU dışındaki bir olayı bildiren
- Bir tamamlama sinyali olarak
- CPU'ya zaman tahsis aracı olarak
- Anormal bir durum göstergesi olarak

Harici olaylar için Kesme

Interrupts for External Events

- CPU yu bir olaydan ”event”haberdar etmek için, bir aygıtta bir olay meydana geldiğinde, bir kesme sinyali oluşur.
- Örneğin,
 - Klavye: Bir tuşa basıldığında, ISR tuş için kod okur)
 - Notebook kapağı: Kapak kapatıldığında ISR bilgisayarı standby moduna alır)

Interruptların Kullanımı

- CPU dışındaki bir olayı bildiren
- Bir tamamlama sinyali olarak
- CPU'ya zaman tahsisi aracı olarak
- Anormal bir durum göstergesi olarak

Tamamlama Sinyalleri için Kesme

Interrupts for Completion Signals

- Bir aygıt bir işlemi tamamladığında bir kesme sinyali oluşur. CPU bu durumu bilmelidir
- Örneğin,
 - Yazıcı: «Çıkış bufferı (tampon) boştur.» CPU daha fazla veri gönderebilir.
 - Tarayıcı: «Bir veri transferi tamamdır.» (CPU/uygulama imaj verisini işlemeye devam edebilir.)

Interruptların Kullanımı

- CPU dışındaki bir olayı bildiren
- Bir tamamlama sinyali olarak
- CPU ya zaman tahsisi aracı olarak
- Anormal bir durum göstergesi olarak

CPU'ya Zaman Tahsisi için Kesme

Interrupts for Allocating CPU Time

- multi-tasking sistemler üzerinde yararlıdır – sistemler aynı anda birden fazla programı çalıştırabilir
- Örneğin,
 - Bir zamanlayıcı (timer) CPU 'yu her 100 μ s ye de kesme için programlayabilir.
 - ISR 'nin yaptığı iş program görevlendiriciliktir.
(dispatcher program)
 - Programları çalıştırma örneğin her 100 μ s de bir diğer programın çalıştırılmasına geçebilir.

Interruptların Kullanımı

- CPU dışındaki bir olayı bildiren
- Bir tamamlama sinyali olarak
- CPU ya zaman tahsisi aracı olarak
- Anormal bir durum göstergesi olarak

Anormal durumlar için Kesme

Interrupts for Abnormal Events

- Anormal bir olay anında örneğin sistemin bakıma ihtiyacı ortaya çıktığında bir kesme sinyali oluşur
- Örneğin.,
 - CPU çipinin yanındaki ısı sensörü– eğer ısı aşırı yüksekse, bir kesme üretilir, ISR CPU yanındaki fan ı aktif eder.

I/O Çeşitleri

- Programlı I/O
- Kesme odaklı (Interrupt-driven) I/O
- Doğrudan Bellek Erişimi
(Direct memory access - DMA)

Niçin DMA?

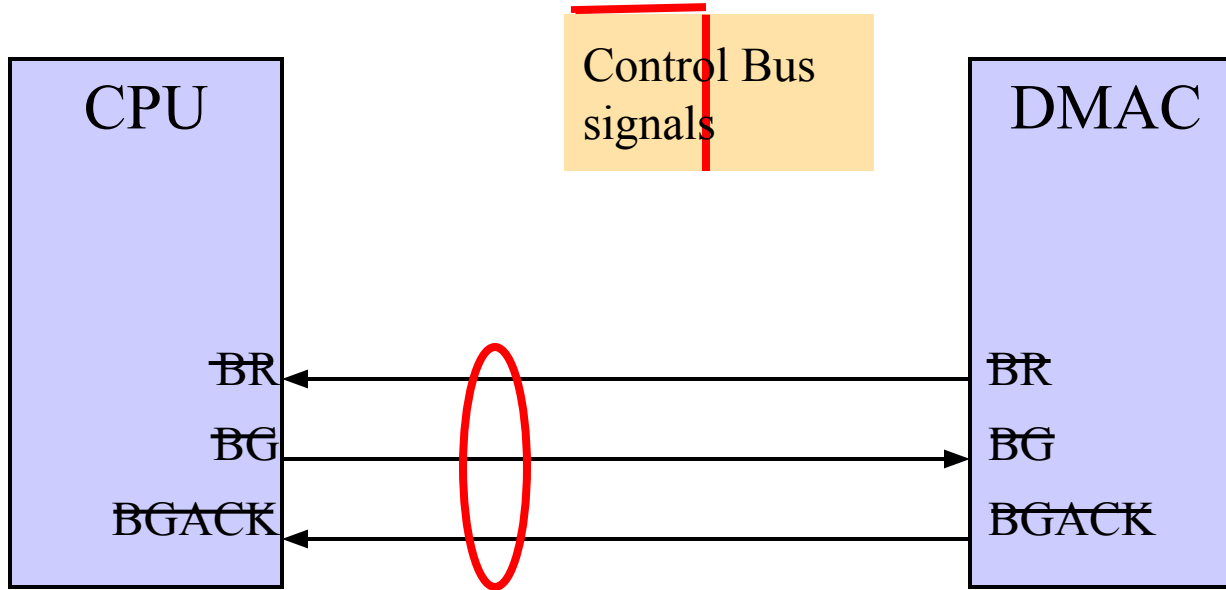
- Aygıt ve bellek arasındaki yüksek hızlı blok transferleri için kullanılır
- Transfer esnasında CPU görev almaz
- Genel DMA aygıtları:
 - Disk sürücüler,
- Hatırlatma (5. slayt)
 - Keyboard data rate \square 0.01 KB/s (her 100 ms 1 byte)
 - Disk drive data rate \square 2,000 KB/s (her 0.5 μ s 1 byte)

Çok yüksek transfer hızları CPU üzerinde çalışan bir yazılım tarafından kontrol edilmesi zor

Nasıl

- CPU DMA denetleyicisine (DMAC) bilgi aktararak DMA operasyonu hazırlar:
 - Aygıttaki verinin yeri
 - Bellekteki verinin yeri
 - Transfer için blok uzunluğu
 - Transfer yönü
 - Transfer modu
- Aygıt veri transferine hazır olduğunda, DMAC sistem veri yolu kontrolünü alır.

“Kontrolü Alma” (1/2)



\overline{BR} = Bus request-Veriyolu Talebi (DMAC: Sistem veri yolu kontrolünü alabilir miyim?)

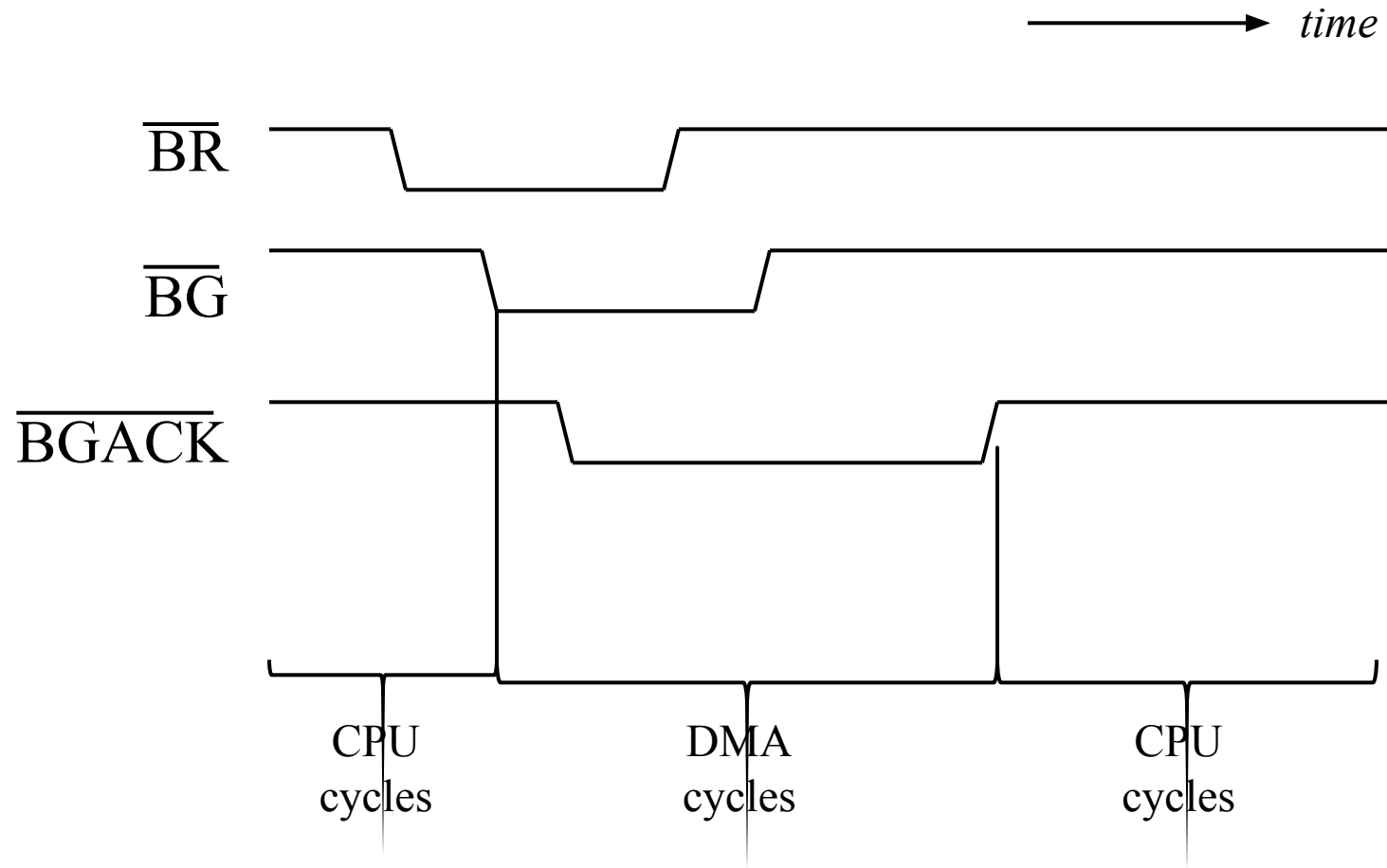
\overline{BG} = Bus grant-Veri yolu bağıışı (CPU: Evet, buyrun.)

\overline{BGACK} = BG kabulü (DMAC: Teşekkürler, Kontrol bende.)

“Kontrolü Alma” (2/2)

- DMAC bir \overline{BR} (“bus request”) sinyali verir
- CPU durur (belkide bir komut icrası ortasında!) ve bir \overline{BG} (“bus grant”) sinyali verir
- DMAC \overline{BGACK} (“bus grant acknowledge”) sinyali verir ve \overline{BR} bırakır
- DMAC sistem veri yollarının kontrolünü alır
- DMAC “CPU gibi davranır” ve gerçekleştirecek transferler için veriyolu sinyalleri üretir (adres, kontrol)

$\overline{\text{BR}}$ - $\overline{\text{BG}}$ - $\overline{\text{BGACK}}$ Timing

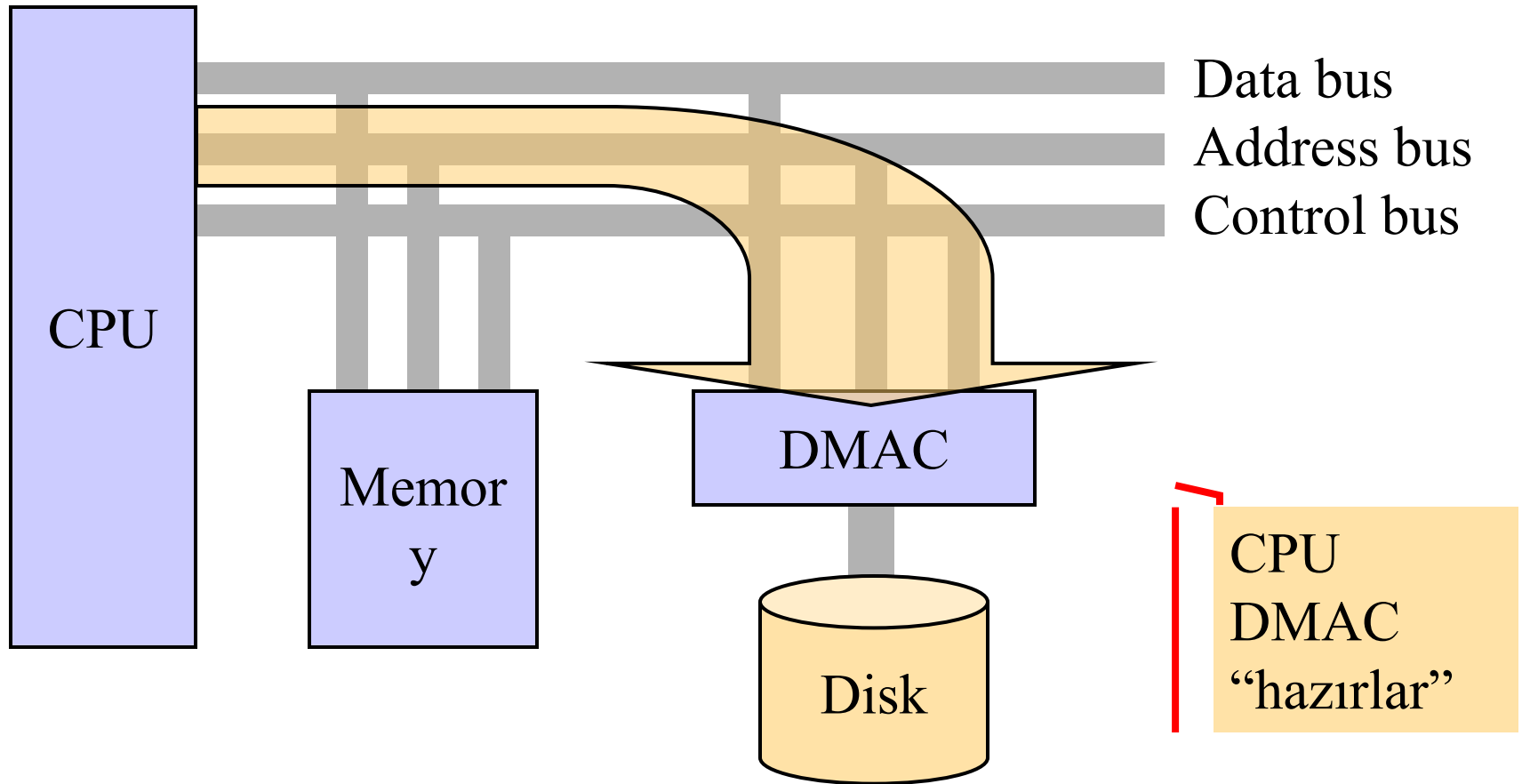


I/O Çeşitleri

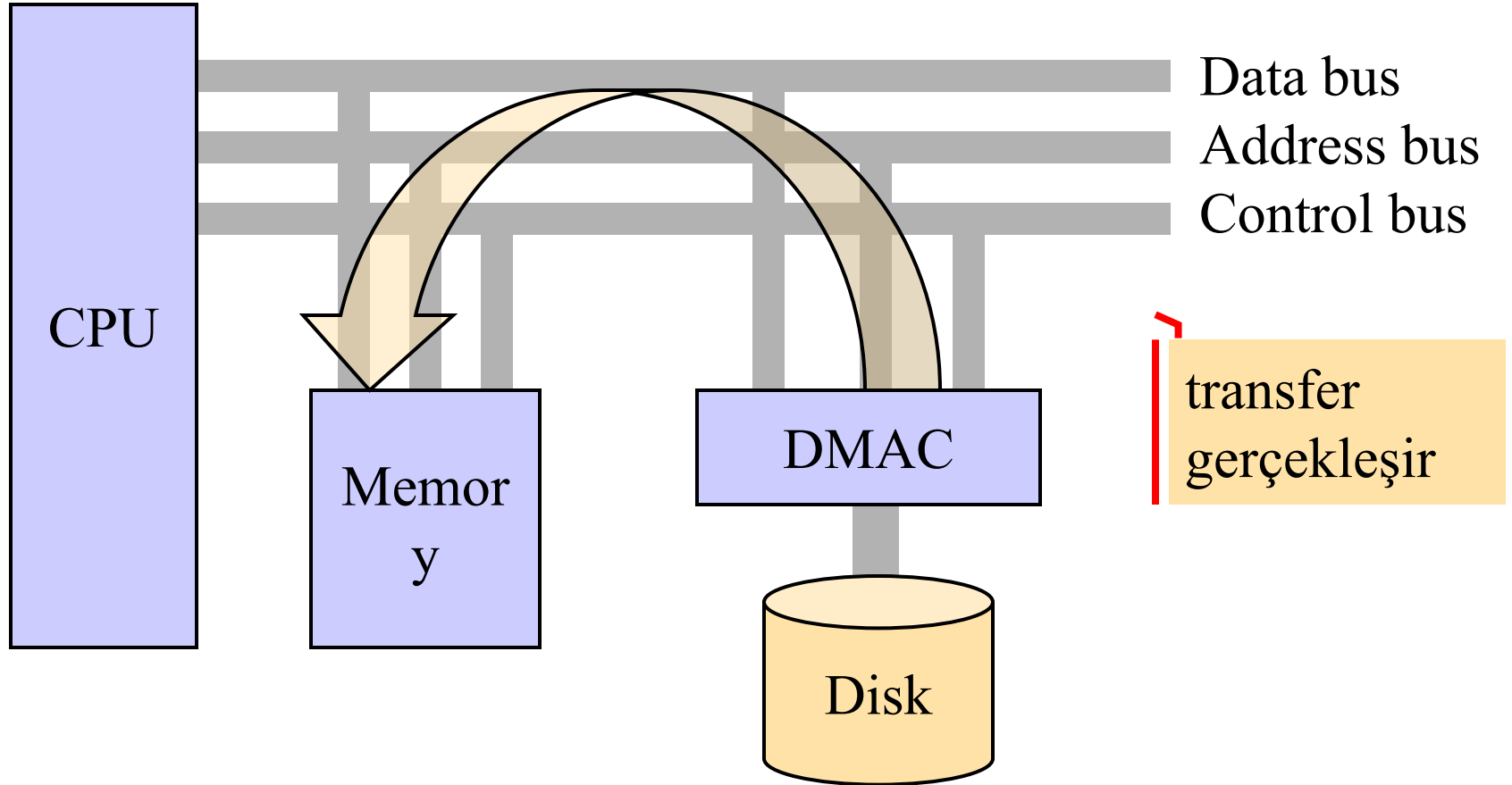
- Programmed I/O
- Interrupt-driven I/O
- Direct memory access (DMA)

DMA her 3 I/O çeşidini de içerir.

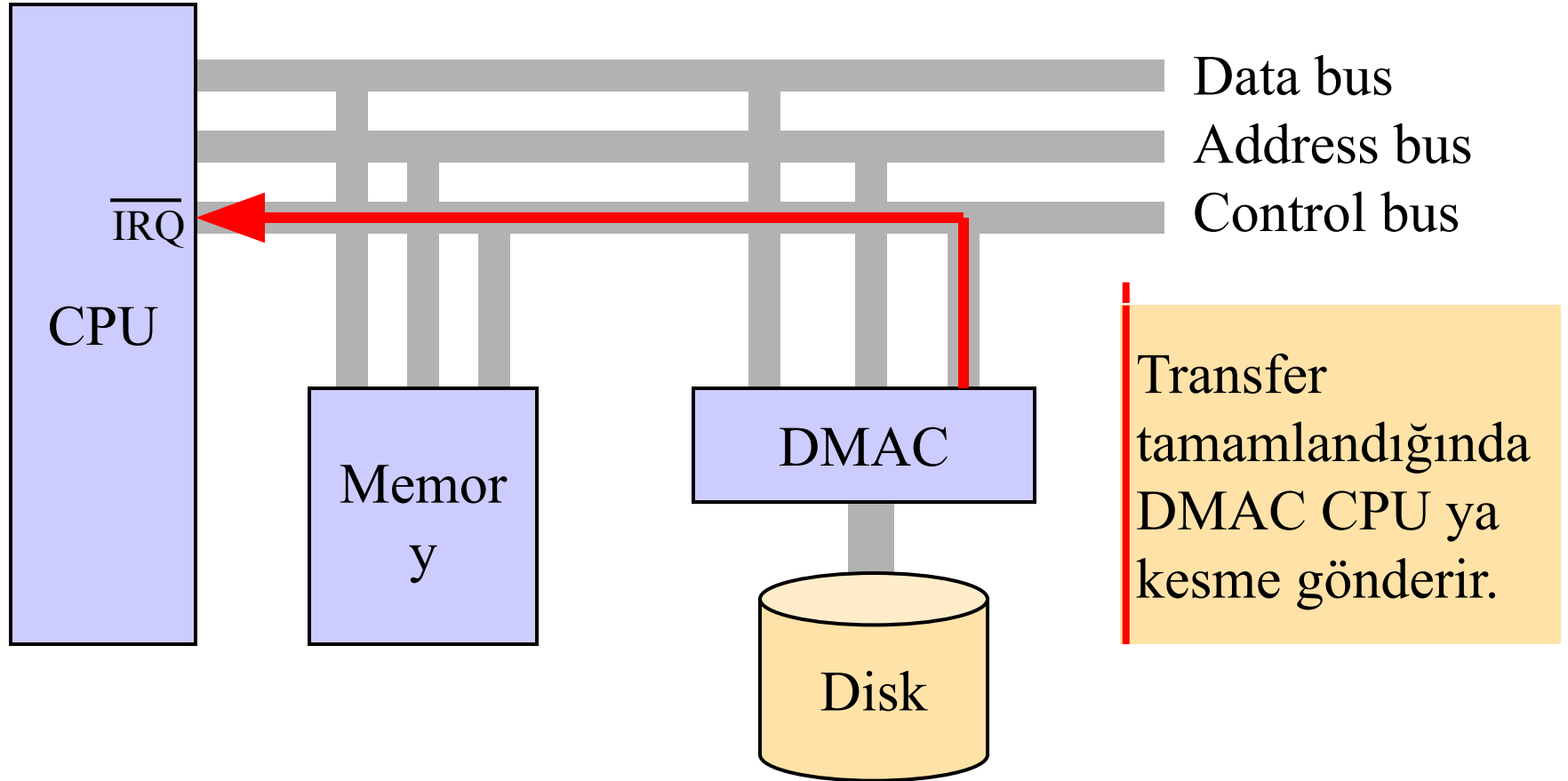
Program-Controlled I/O (DMA)

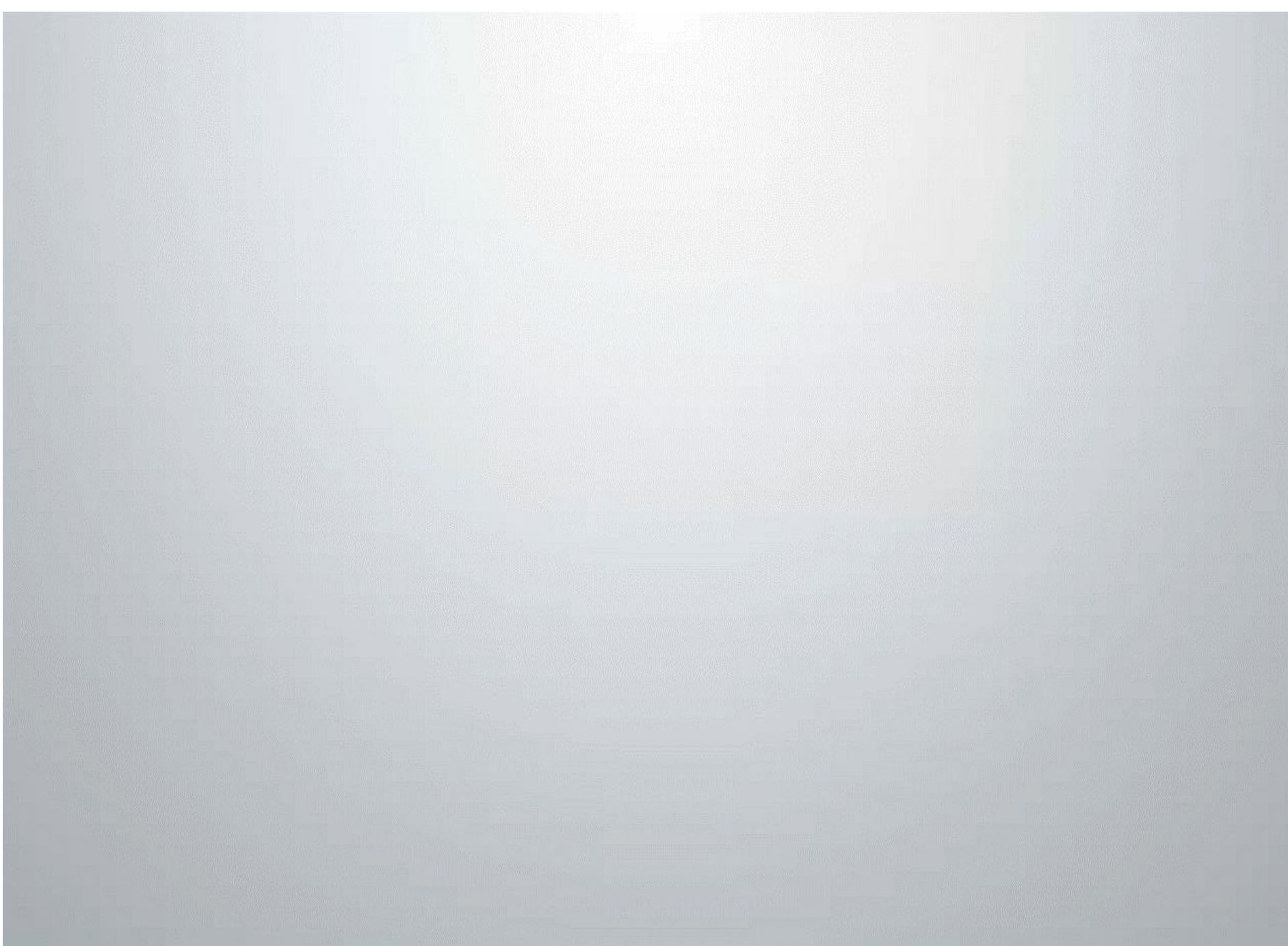


DMA



Interrupt-driven I/O (in DMA)





Teşekkür Ederim