



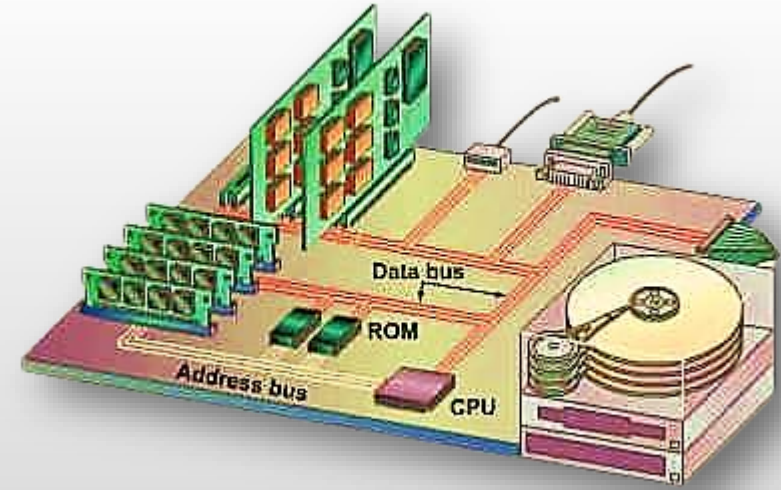
Bölüm 11: Giriş Çıkış

İşletim Sistemleri



Genel Bakış

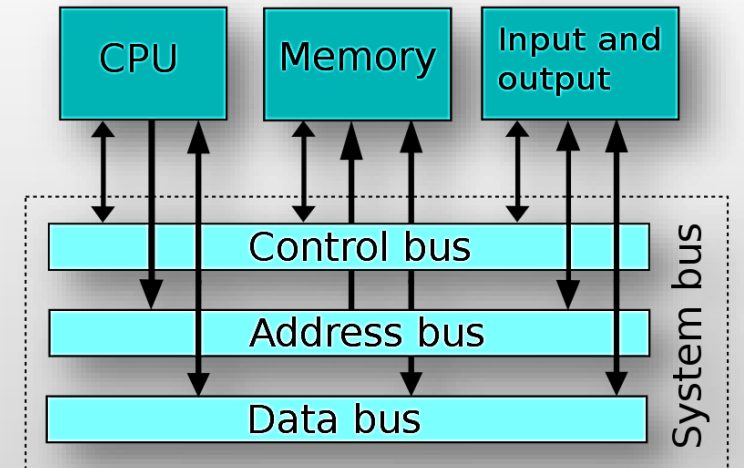
- İşletim sistemi, *G/Ç aygıtlarını* kontrol eder.
 - Komut verir.
 - Kesmeleri yönetir.
 - Durumlarını okur.
 - Hataları ele alır.
- Aygıtların kullanımını için *arayüz* sağlar.
 - Aygıttan bağımsız.
 - Katman katman yapılandırılır.





Veri Yolu

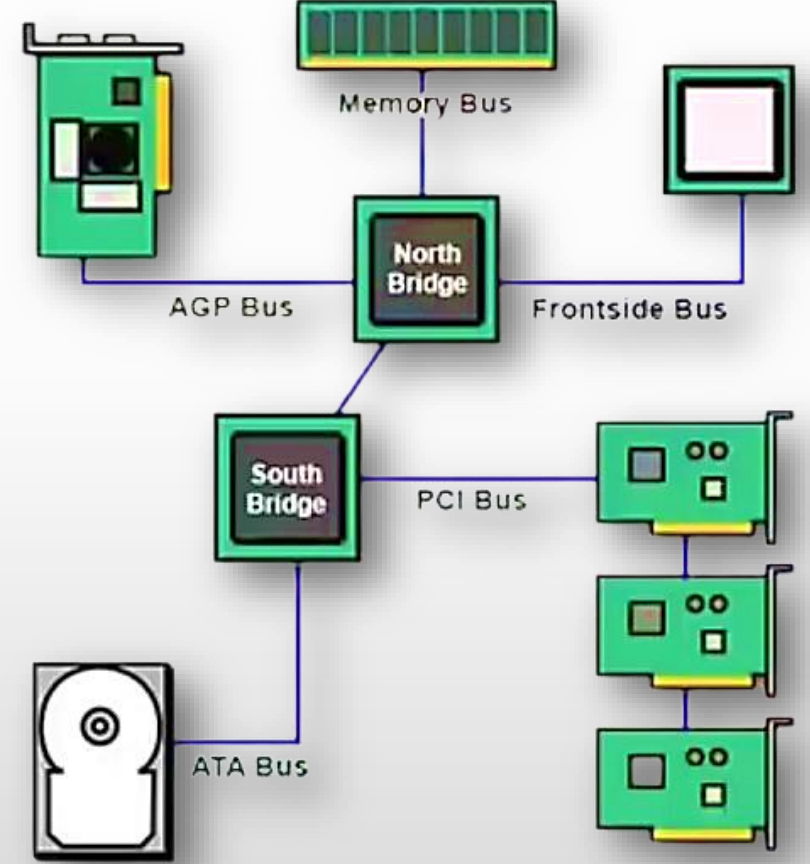
- Verilerin iletiildiği fiziksel bağlantı.
- Veri yolları
 - **Dahili:** *sistem veri yolları, bellek veri yolları, G/Ç veri yolları gibi.*
 - **Harici:** *Ethernet, USB, FireWire gibi.*
- CPU, bellek ve G/Ç aygıtları arasında veri iletmeye yarar.
- Veri aktarımı hızı.
- Aynı anda birden fazla veri akışını yönetebilme.





Veri Hızı

- Birim zamanda iletilen veri miktarı,
 - saniye başına bit (*bps*).
 - saniye başına bayt (*Bps*).
- Veri yolunun *bant genişliği*, *sinyal kalitesi*, *sinyal girişimi* veri hızını etkiler.
- Yüksek veri hızları,
 - gerçek zamanlı video ve ses iletimi gibi,
 - zamana duyarlı uygulamalar için kritik.





Örnek Veri Hızları

- **Hard disk:** 200 *MB/s* okuma, 150 *MB/s* yazma
- **Solid state:** 1.5 *GB/s* okuma, 800 *MB/s* yazma
- **USB 2.0:** 480 *Mbps*, **USB 3.0:** 5 *Gbps*
- **Ethernet:** 10/100/1000 *Mbps*
- **SATA III:** 6 *Gbps*
- **Keyboard:** 2000 *karakter*, **Mouse:** 1000 *reports*
- **Modem:** 56 *Kbps* / 1 *Mbps*
- **Camcorder:** 50 *Mbps*
- **Firewire:** 50/100 *MB/s*



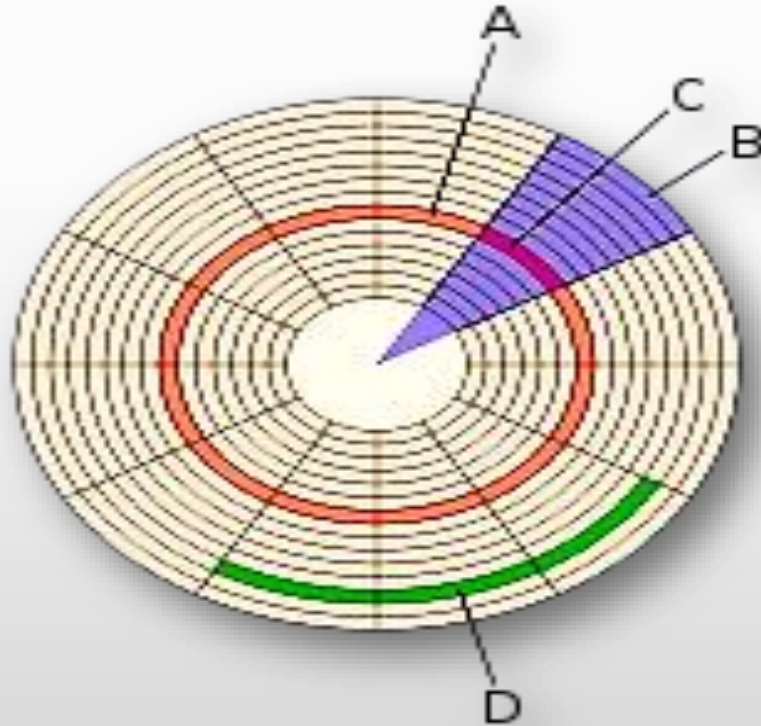
Genel Bakış

- İki tip G/Ç aygıtı,
 - blok tabanlı: blok blok okuma yapar.
 - Sabit disk, *CD-ROM*, *USB* bellek.
 - *512 bayt - 32 KB*.
 - karakter tabanlı: karakter karakter okuma yapar.
 - Yazıcı, klavye, fare, ağ arabirimleri.
- İşletim sistemi,
 - aygıtın marka, model, özelliklerinden bağımsız bir arayüz sağlar.



Disk Geometrisi

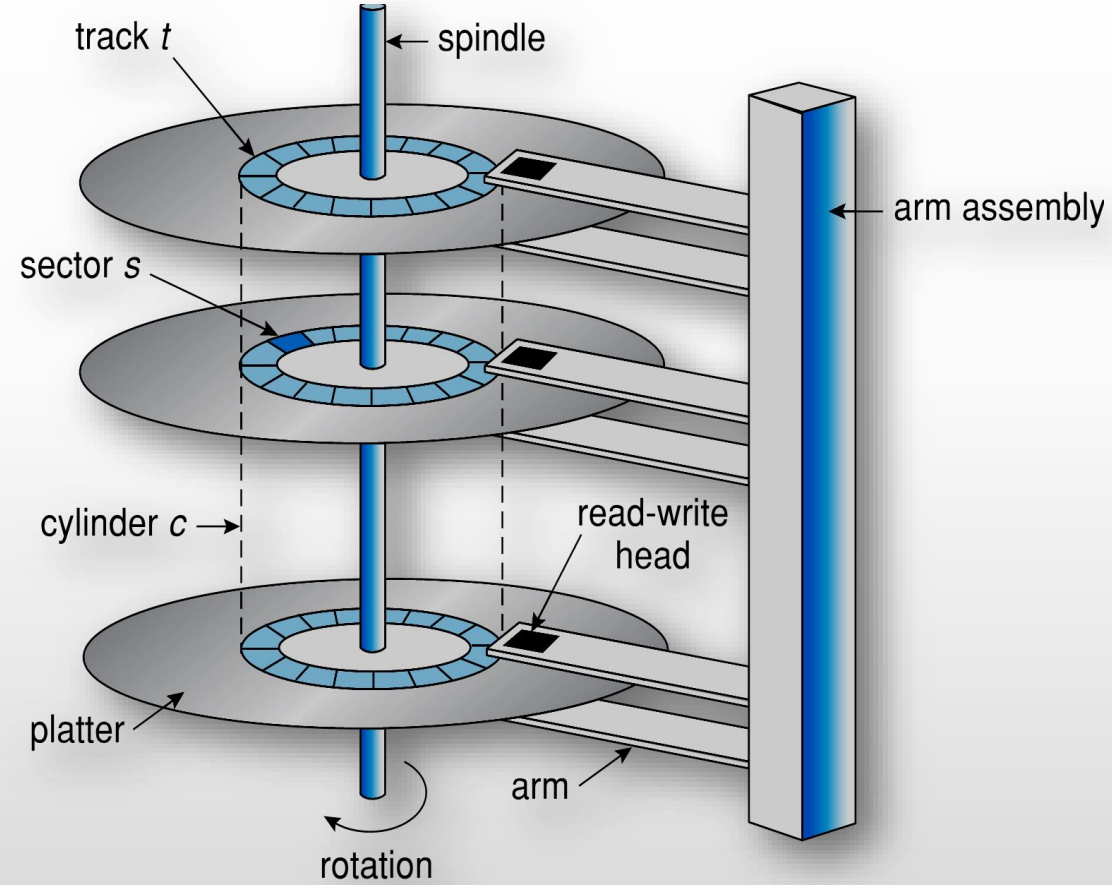
- A: iz (*track*), B: sektör, C: geometrik sektör, D: küme (*cluster*)





Disk Mekanizması

- .





Aygıt Denetleyicileri

- G/Ç birimi 2 bileşene sahiptir.
 - mekanik,
 - elektronik (*kontrol birimi*).
- Denetleyici,
 - bağlayıcı (*connector*) ve
 - yongadan (*çip*) oluşur.
- İz (*track*), 512 baytlık sektörlerden oluşur.



Aygıt Denetleyicileri

- Seri bit akışı,
 - eşzamanlama öncülü (*preamble*),
 - 4096 bit/sektör,
 - hata düzeltme kodu (*error correcting code*) içerir.
- **Öncül:** sektör numarası, silindir numarası, ve sektör boyutu.
- Denetleyici,
 - bit akışından bir blok oluşturur,
 - hata düzeltme yapar,
 - denetleyici içinde bulunan ara belleğe (*buffer*) koyar.



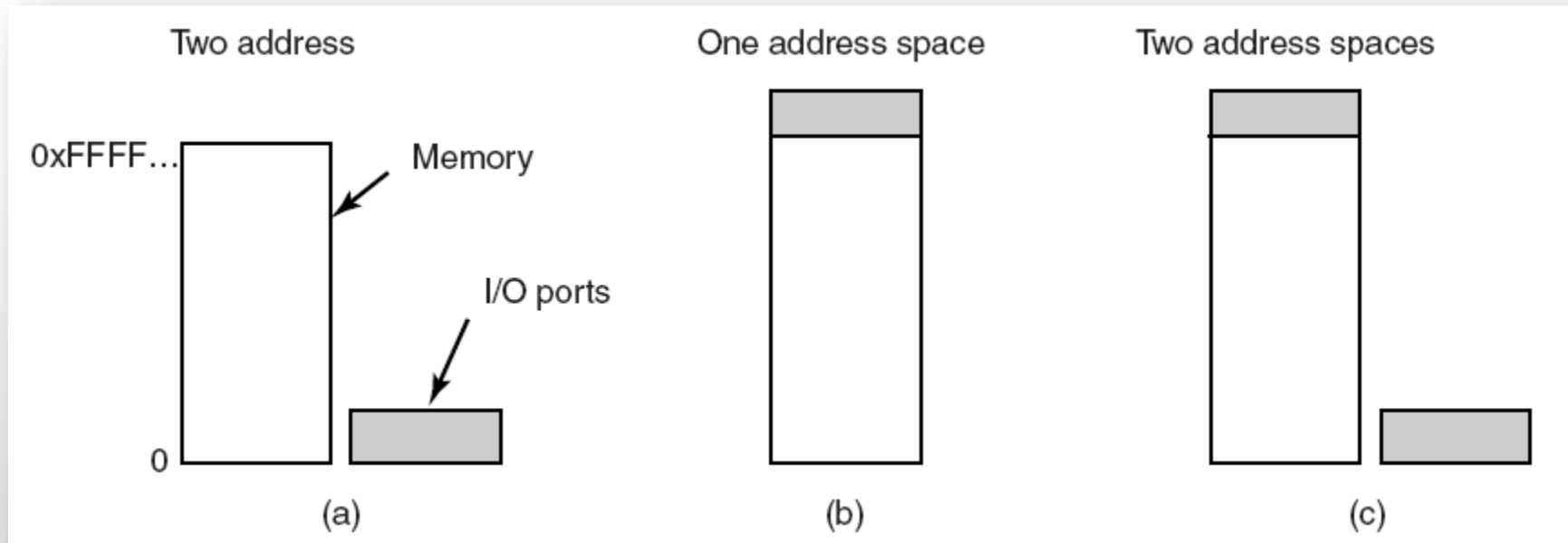
Bellek Eşlemeli G/Ç

- Denetleyici, işletim sisteminin okuyup yazabildiği yazmaçlara sahiptir.
- **write**: Aygıtta komut gönderilir.
- **read**: Aygıtın durumu okunur.
- Aygıtlar, işletim sisteminin okuyup yazabildiği arabelleği vardır.
- Örneğin, ekranda pikselleri görüntülemek için kullanılan video *RAM*.
- CPU, yazmaçlar ve arabellekle nasıl iletişim kurar?



Bellek Eşlemeli G/Ç

- (a) G/Ç kapıları ve bellek ayrı. (b) Bellek eşlemeli. (c) Hibrit yaklaşım.





Aygıt G/Ç Kapıları (Port)



| I/O address range (hexadecimal) | device |
|---------------------------------|---------------------------|
| 000–00F | DMA controller |
| 020–021 | interrupt controller |
| 040–043 | timer |
| 200–20F | game controller |
| 2F8–2FF | serial port (secondary) |
| 320–32F | hard-disk controller |
| 378–37F | parallel port |
| 3D0–3DF | graphics controller |
| 3F0–3F7 | diskette-drive controller |
| 3F8–3FF | serial port (primary) |



CPU Yazmaç ve Arabelleği Nasıl Adresler?

- İlk tasarım
 - Read komutu, kontrol satırına konulur.
 - Adres, adres satırına konulur.
 - G/Ç veya bellek alanındaki veri, sinyal hattına konulur.
 - Sinyal hattından okunur.
- Bellek eşlemeli yaklaşım,
 - Adres, adres satırına konulur.
 - Bellek ve G/Ç aygıtları,
 - adresi hizmet verdikleri aralıkta karşılaştırırlar.



Bellek Eşlemeli G/Ç Avantajları

- Kontrol yazmaçlarını okumak/yazmak için özel komutlara gerek yok.
- Doğrudan G/Ç yapılmasını engellemek için özel korumaya gerek yok.
- Bir komut, kontrol yazmaçlarına ve belleğe erişebilir.
- C dilinde bir aygıt sürücüsü yazılabilir. 😊
- G/Ç belleği, kullanıcı alanına konulmaz.



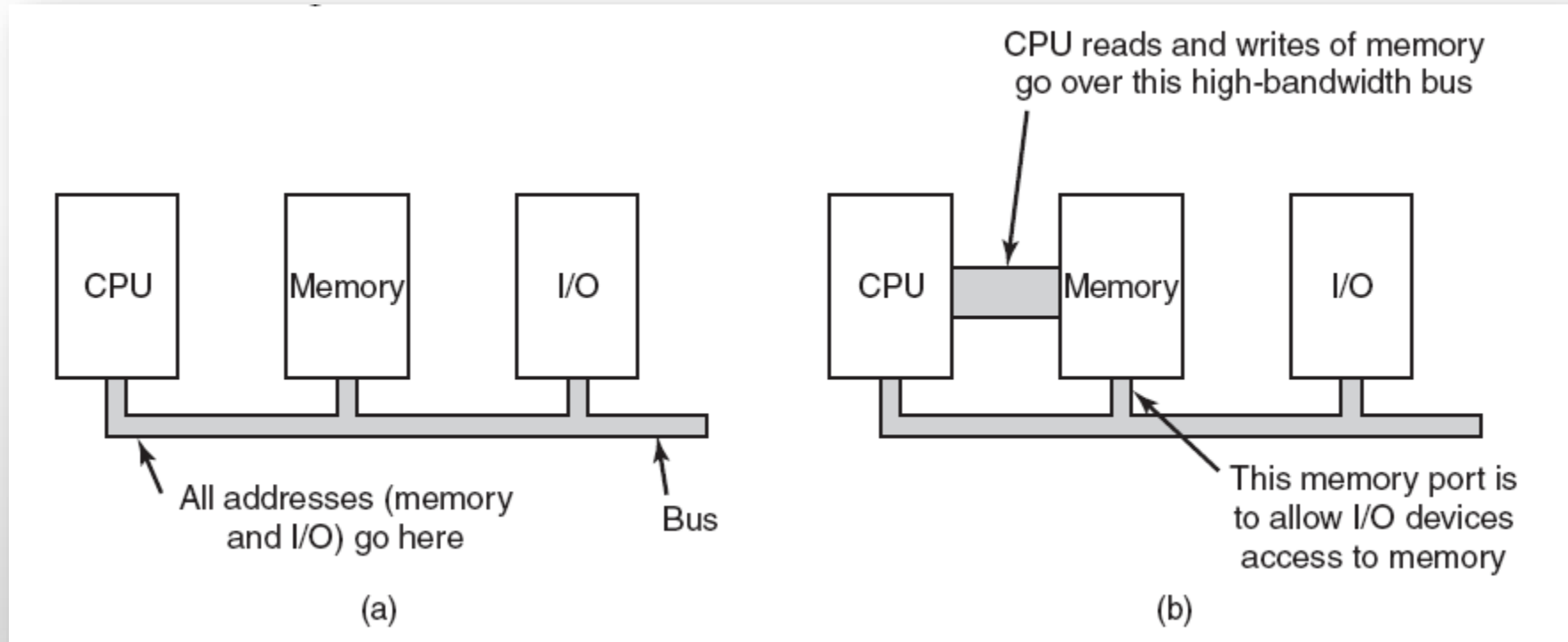
Bellek Eşlemeli G/Ç Dezavantajları

- Bellek sözcükleri önbelleğe alınır,
 - önbellekte güncel değerler olmayabilir. *veri tutarsızlığı!* ☹️
- Gerekli olduğunda önbelleğe almayı devre dışı bırakabilmelidir.
- G/Ç aygıtları ve bellek, bellek erişim isteklerine yanıt vermelidir.
- Tek veri yolu ile çalışır. ☹️
 - Çünkü, hem bellek hem de G/Ç, veri yolu üzerindeki adrese bakar.
 - Çoklu veri yolu ile çalışması zor.
 - Çünkü, G/Ç aygıtları adresi göremez.



Bellek Eşlemeli G/Ç

- (a) Tek veri yolu mimarisi. (b) Çift veri yolu mimarisi.





Doğrudan Bellek Erişimi (DMA)

- CPU, G/Ç denetleyicisinden her seferinde bir *baytlık* veri talep edebilir.
 - Büyük zaman kaybı.
- *DMA* denetleyicisi anakart üzerinde bulunur.
- CPU, denetleyicideki yazmaçlardan okuyabilir, yazmaçlara yazabilir.
 - Bellek adres yazmacı.
 - Bayt sayısı yazmacı.
 - Kontrol yazmaçları.
 - G/Ç bağlantı noktası (*port*),
 - aktarım yönü (*direction*),
 - aktarım birimi (*bayt/word (sözcük)*),
 - tek seferde aktarılacak bayt sayısı (*burst*).



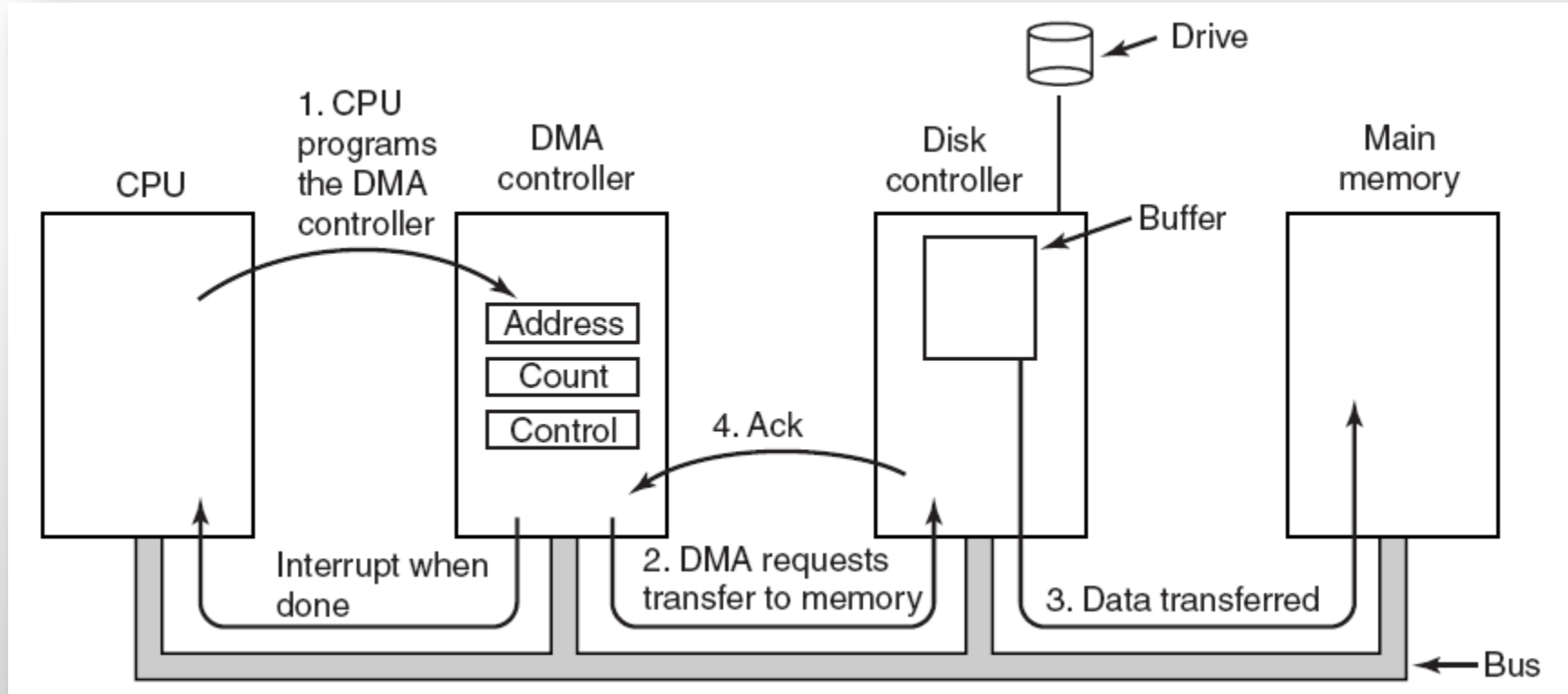
DMA Ne Zaman Kullanılmaz?

- Denetleyici,
 - belleğe bir blok okur.
 - hata kontrolü (*checksum*) yapar.
 - kesme üreterek işletim sistemini uyarır.
 - belleğe her seferinde bir *bayt* gönderir.
 - sık sık kesme üretir. ☹️



Doğrudan Bellek Erişimi (DMA)

- DMA veri transferi aşamaları.





DMA Denetleyicisi Modları

- **Cycle-Stealing:**
 - bir anda bir sözcük aktarılır.
 - veri yolunu kullanma döngüleri için *CPU* ile rekabet eder.
 - *CPU* ara sıra döngüsünü (*cycle*) *DMA* denetleyicisine bırakır.
- **Burst:**
 - *DMA* denetleyicisi veri yolunu alır ve bir blok veri gönderir.
- **Fly-By:**
 - *DMA*, aygıt denetleyicisine *bellek yerine bana gönder* der.
 - Aygıtlar arasında veri aktarımı için kullanılır.



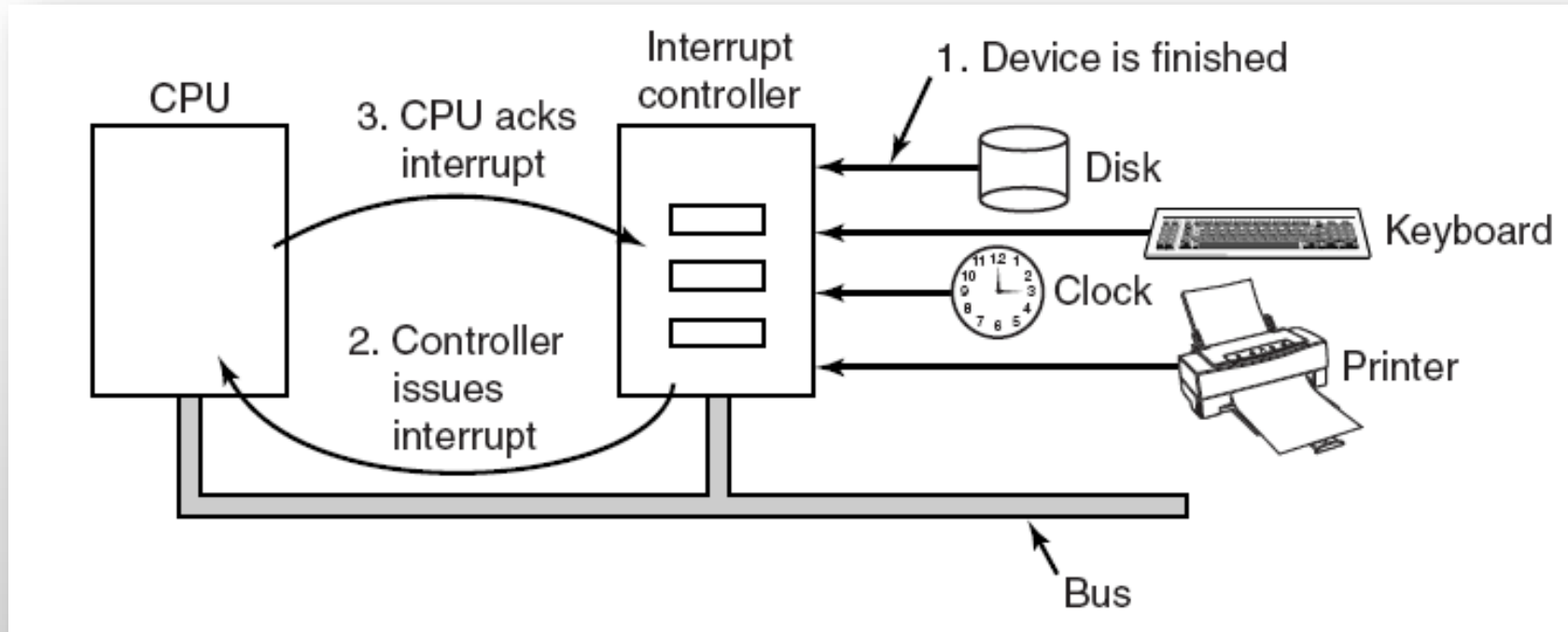
Sorular

- Neden veriler denetleyicilerin arabelleğine alınır?
 - Hata kontrolü (*Checksum*) yapılabilir.
 - Veri yolu meşgul olabilir,
 - Verilerin bir yerde saklanması gerekir.
- DMA gerçekten buna değer mi?
 - CPU, DMA denetleyicisinden çok daha hızlıdır.
 - Aktarılacak çok fazla veri yoksa değmez.



Kesmeler

- Aygıt işini bitirince kesme oluşturur,
- Kesme denetleyicisi ile veri yolundaki kesme hatlarından haberleşir.





Kesme İşleme

- Denetleyici, adres satırına bir sayı koyarak kesme üreten aygıtı söyler.
- Kesme vektörü, ilgili kesme servis prosedürünü işaret eder.
- Adres satırındaki sayı, kesme vektörüne *indis (index)* görevi görür.
- Kesme hizmeti yordamı (ISR) kesmeyi kontrol eder.
- ISR (*Interrupt service routine*).
- Yürütülmesi durdurulan süreç ile ilgili bilgileri kaydeder.



PC ve PSW Kaydedilebilir Mi?

- Ardışık düzen (*pipelined*) veya
- Süper sayıl (*superscalar*) işlemciler kullanılmıyorsa, **evet**.
- **Pipelined:**
 - bir komut grubu kısmen tamamlanır.
 - *multiple interrupts can occur in the same clock cycle.*
- **Superscalar:**
 - komutlar ayrıştırılır ve sıra dışı yürütülebilir.
 - *multiple pipelines.*



Kesin olan Kesmenin Özellikleri

- Program sayacı (*PC*) bilinen bir yere kaydedilir.
- *PC*'den önceki tüm komutlar tam olarak yerine getirilmiştir.
- *PC*'den sonraki hiçbir komut yürütülmemiştir.
- *PC*'nin gösterdiği komutun durumu bilinmektedir.



G/Ç Aygıt Kesmesi

- G/Ç aygıtı, bir durum oluştuğunda kesme üretir.
- CPU, kesmeye yanıt vermeden önce işlemekte olduğu komutu tamamlar.
- Mevcut komutun düzgün bir şekilde tamamlandığından emin olur.
- CPU, çalışan sürecin bağlamını kaydeder.
- CPU, ardından Kesme hizmet yordamını (*ISR*) çağırır.
- Kesme hizmet yordamı tamamlandığında, kaydedilen bağlam geri yüklenir.
- Yürütme, kesintiyi izleyen komuttan devam eder.



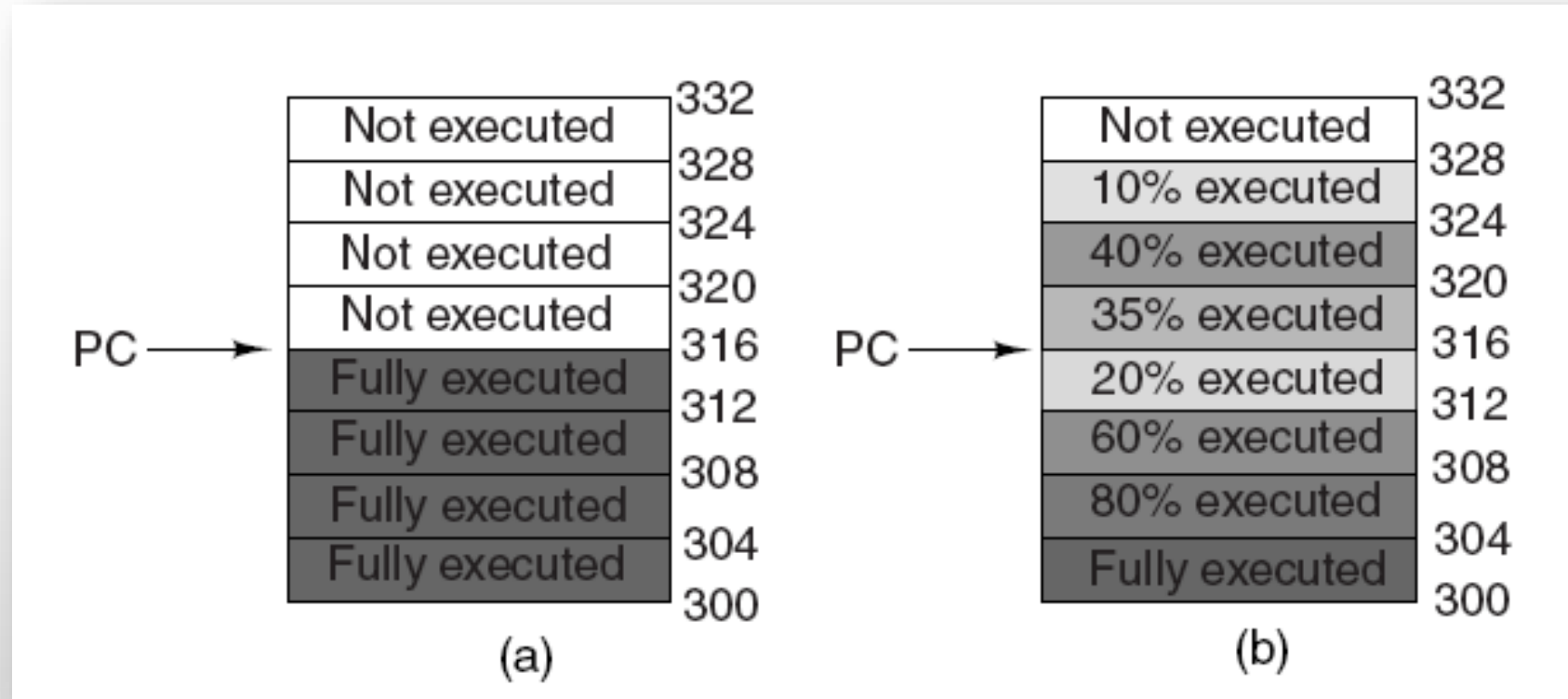
Sayfa Hatası / Aritmetik Taşma

- Page fault / arithmetic overflow.
- CPU, işlemekte olduğu komutu tamamlayamaz.
- Komut iptal (*abort*) edilir.
- CPU, çalışan sürecin bağlamını kaydeder.
- CPU, ardından Kesme hizmet yordamını (*ISR*) çağırır.
- Kesme hizmet yordamı tamamlandığında, kaydedilen bağlam geri yüklenir.
- Yürütme, tamamlanamayan komuttan devam eder.



Kesin ve Kesin Olmayan Kesmeler

- (a) Kesin (*precise*) kesme. (b) Kesin olmayan (*imprecise*) kesme.





Kesin Olmayan Kesmenin Özellikleri

- Kesmeyi yeniden başlatabilmek için,
 - Karmaşık donanım ihtiyacı,
 - backtracking, shadow, rename registers.
 - İşletim sisteminde karmaşık işlemler gerektirir.
 - Kesme oluşturan komuttan sonraki komutlar geri alınmalı (*undo*).
 - Hata veren komut tekrar başlatılmalı (*restart*).

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| ■ | ■ | ■ | ★ | ■ | ■ | ■ | ■ |



G/Ç Yazılımı Hedefler

- **Aygıt bağımsızlığı:** aygıtı erişirken aygıtı belirtmek gerekmemeli.
- **Tek tip adlandırma:** aygıt adı, tipine bağlı olmamalı.
- **Hata ele alma:** aygıtı olabildiğince yakın yerde olmalı.
- **Bloke olma:** İşletim sistemi G/Ç işlemlerini bloke edebilmeli.
 - (örneğin, okuma sırasında veri gelene kadar süreç bloke edilir.)
- **Ara bellek:** bir paket geldiğinde tampon belleğe konabilmeli.
- **Paylaşım:** Paylaşılabilen (*disk*), paylaşılabilen (*teyp*) aygıtlar ele alınmalı



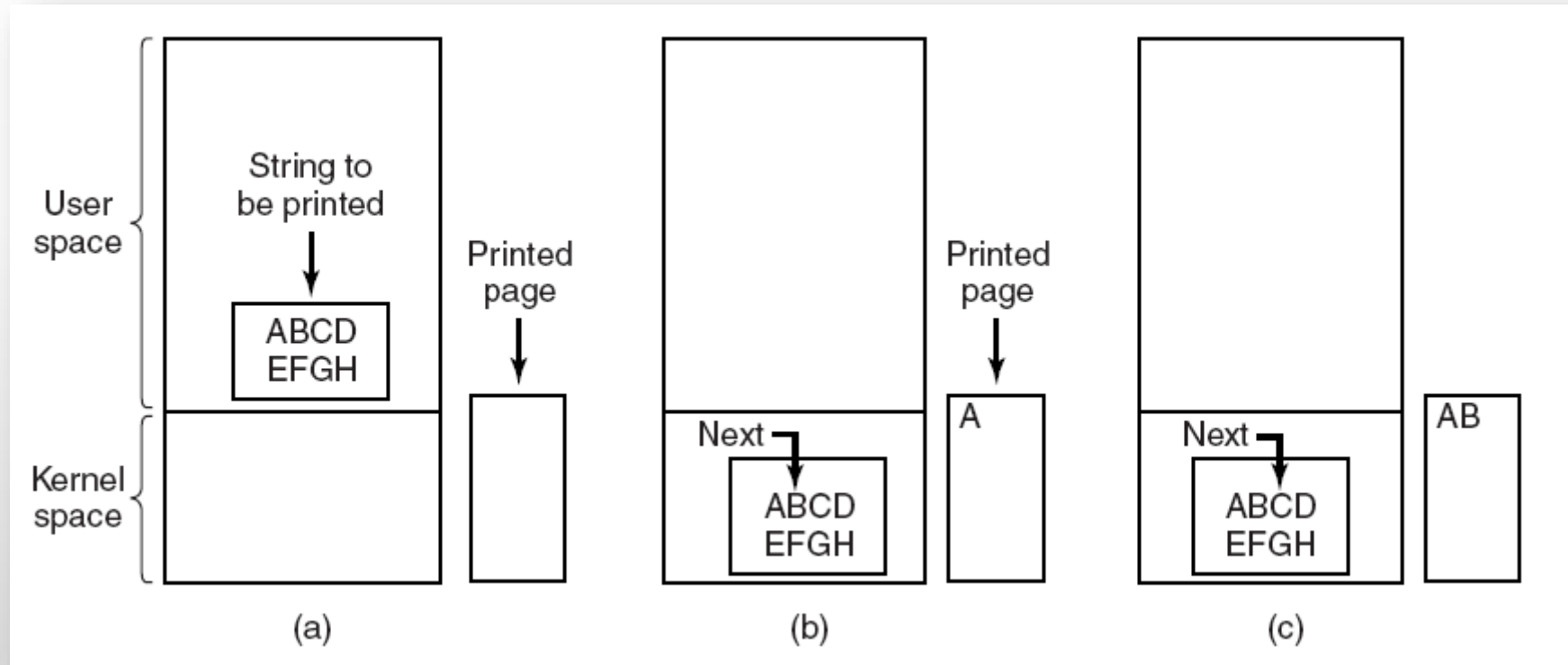
Programlanmış G/Ç

- Bir komut ile G/Ç aygıtına veri gönderilir.
- Bir komut ile G/Ç aygıtından veri okunur.
- CPU, sonraki işleme geçmeden, G/Ç komutunun tamamlanmasını bekler.
- Uygulaması basit ve kolay.
- G/Ç komutunun tamamlanmasını beklenirken CPU zamanı boşa harcanır.



Programlanmış G/Ç

- Yazıcıdan karakter dizisi yazdırma adımları.





Programlanmış G/Ç

- Programlanmış G/Ç kullanarak yazıcıya bir dizi karakter yazma.

```
copy_from_user(buffer, p, count);
for (i = 0; i < count; i++) {
    // loop on every character
    while (*printer_status_reg != READY);
    // loop until the printer is ready
    *printer_data_register = p[i];
    // output one character
}
return_to_user();
```



Programlanmış G/Ç

- Her bayt için,
 - *Durum yazmacı*, *meşgul biti* 0 olana kadar okunur.
 - Okuma veya yazma *bitine* 1 atanır.
 - Yazma komutu ise veriler *veri çıkış yazmacına* kopyalanır.
 - Komut için *hazır bitine* 1 atanır.
 - Denetleyici *meşgul bitine* 1 atar, ve veri aktarımını yürütür.
 - Aktarım tamamlandığında, *meşgul biti*, *hata biti* ve *hazır biti* temizlenir.
- Verimsiz bir yöntem.
- CPU başka sürece geçip, döngü (*cycle*) kaçırırsa,
 - üzerine yazmadan (*overwrite*) dolayı veri kaybı yaşanabilir.



Kesme Odaklı G/Ç

- G/Ç aygıtı, bir durum olduğunda *CPU*'ya kesme isteği (*IRQ*) gönderir.
- CPU, isteği alınca mevcut görevini durdurur.
- İlgili *kesme hizmeti yordamı* yürüterek G/Ç aygıtına hizmet verir.
- G/Ç işlemi eşzamansız (*asynchronous*) olarak gerçekleştirilir.
- G/Ç işlemi devam ederken CPU diğer görevlerini gerçekleştirebilir.
- CPU kullanımı ile G/Ç verimliliği arasında bir denge sağlar.
- Kesme ele alma (*interrupt handling*) ve
 - önceliklendirme (*prioritization*) mekanizması gerektirir.

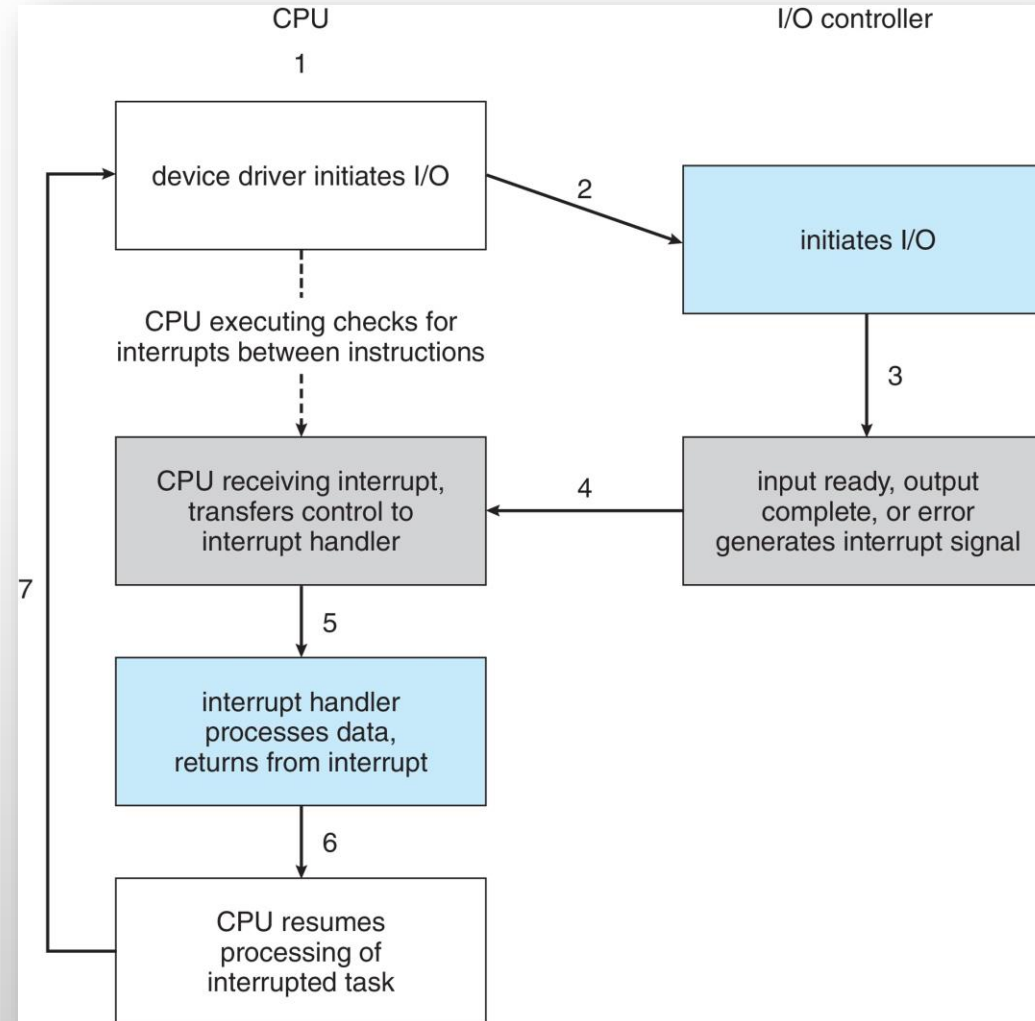


Kesme Odaklı G/Ç

- **Fikir:**
 - G/Ç isteyen süreç bloke edilir,
 - başka bir süreç çizelgelenir.
- G/Ç isteği tamamlandığında, çağırılan sürece geri dönülür.
- Yazıcı,
 - Bir karakter yazdığı anda kesme oluşturur.
 - Karakter dizisinin sonuna kadar yazdırmaya devam eder.
 - İşlem bitince, çağırılan süreç yeniden başlatılır.



Kesme Odaklı G/Ç





Kesme Odaklı G/Ç

- Yazdırma sistem çağrısı yapıldığında yürütülen kod.

```
copy_from_user(buffer, p, count); // user space to kernel  
enable_interrupts();
```

```
while (*printer_status_reg != READY); // busy-waiting  
    *printer_data_register = p[0];  
scheduler();
```



Kesme Odaklı G/Ç

- Yazıcı için kesme hizmet yordamı (*ISR*).

```
if (count == 0) { // all characters have been printed
    unblock_user(); // unblock the user-level process
} else {
    *printer_data_register = p[i];
    count = count - 1;
    i = i + 1; // point to the next character in buffer
}
acknowledge_interrupt(); // interrupt handling complete
return_from_interrupt(); // Return from interrupt context
```




DMA Kullanarak G/Ç

- G/Ç aygıtı, CPU'yu dahil etmeden sistem belleğine doğrudan erişebilir.
- CPU, G/Ç işlemleri sırasında diğer görevlerini gerçekleştirebilir.
- DMA denetleyicisi, G/Ç aygıtı ile bellek arasındaki veri transferini yönetir.
- CPU'yu serbest bırakarak sistem performansını artırır.
- Ancak; DMA denetleyici, yönetim ve bellek tahsisi açısından karmaşık.



DMA Kullanarak G/Ç

- Yazıcıya karakter dizisi gönderilmek istenirse,
 - CPU, her karakter yazdırıldığında değil,
 - yalnızca arabellek yazdırıldığında kesmeye uğrar.
- DMA ne zaman kullanılmaya değer?
 - DMA denetleyicisi, aygıtı CPU'nun çalıştırdığı hızla çalıştırabiliyor ise.
 - Yeterli miktarda aktarılacak veri var ise.



DMA Kullanarak G/Ç

- Yazdırma *sistem çağrısı* yapıldığında yürütülen kod parçası.

```
copy_from_user(buffer, p, count);  
set_up_DMA_controller();  
scheduler(); // CPU may switch to another task
```



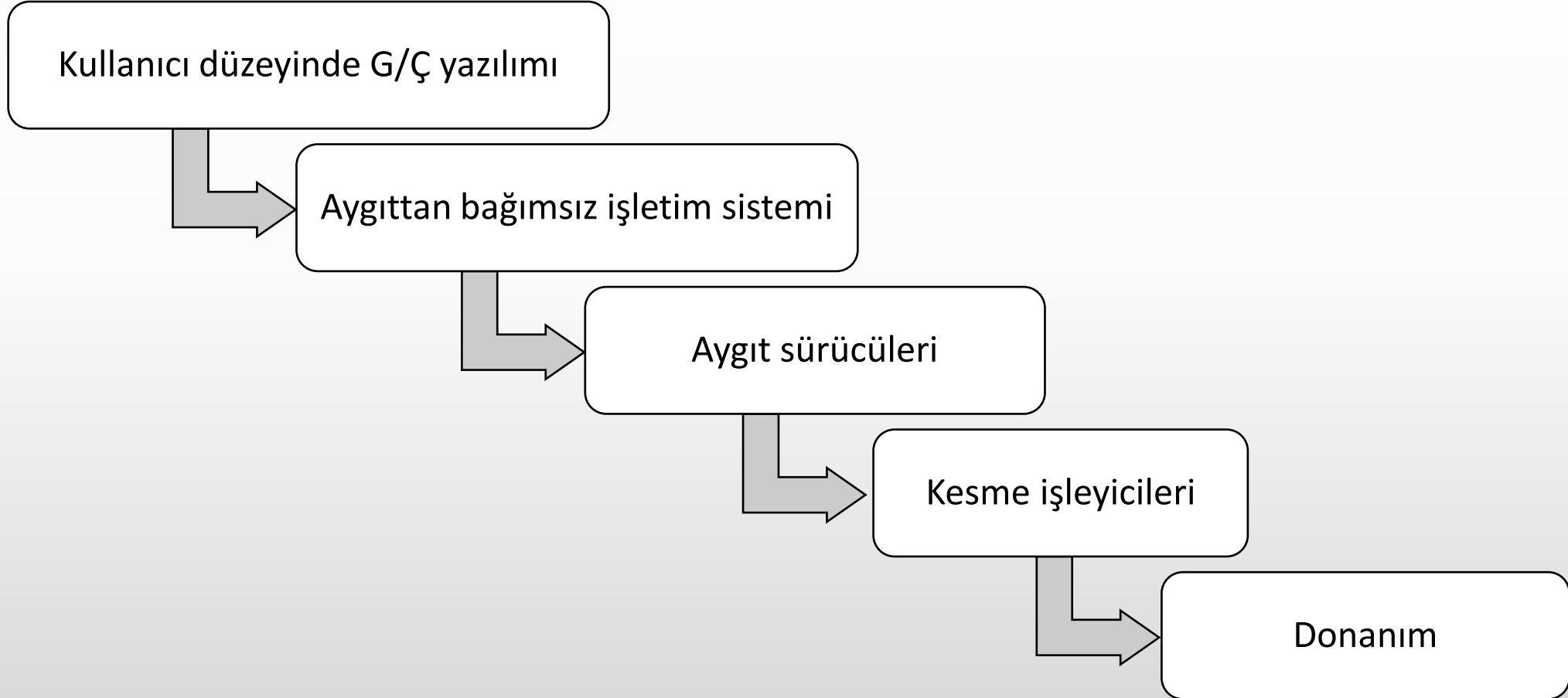
DMA Kullanarak G/Ç

- Kesme hizmet yordamı.

```
acknowledge_interrupt(); // interrupt has been received
unblock_user(); // waking up a process that was blocked
return_from_interrupt(); // signals end of ISR
```



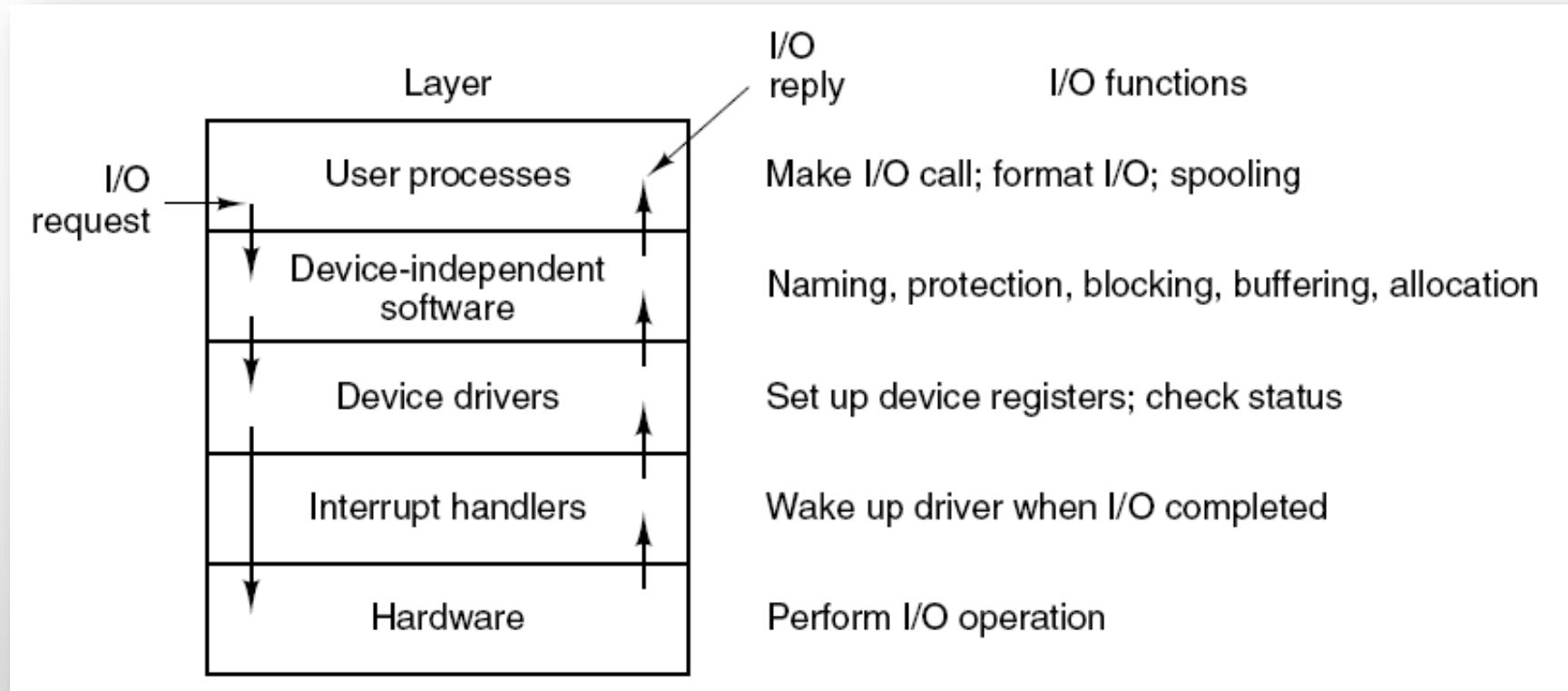
G/Ç Yazılım Katmanları





G/Ç Sisteminin Katmanları

■ .





Kesme İşleyicileri (Handler)

- Kesme donanımı tarafından yazmaçlar kaydedilir.
- Kesme hizmet yordamı için bir bağlam (*context*) ayarlanır.
- Kesme hizmet yordamı için bir yığın ayarlanır.
- Kesme denetleyicisi, kesmenin alındığına dair bilgilendirilir.
- Merkezi kesme denetleyicisi yoksa kesme yeniden etkinleştirilir (*re-enable*)
- Yazmaçlar, süreç tablosundan alınan değerlerle güncellenir.



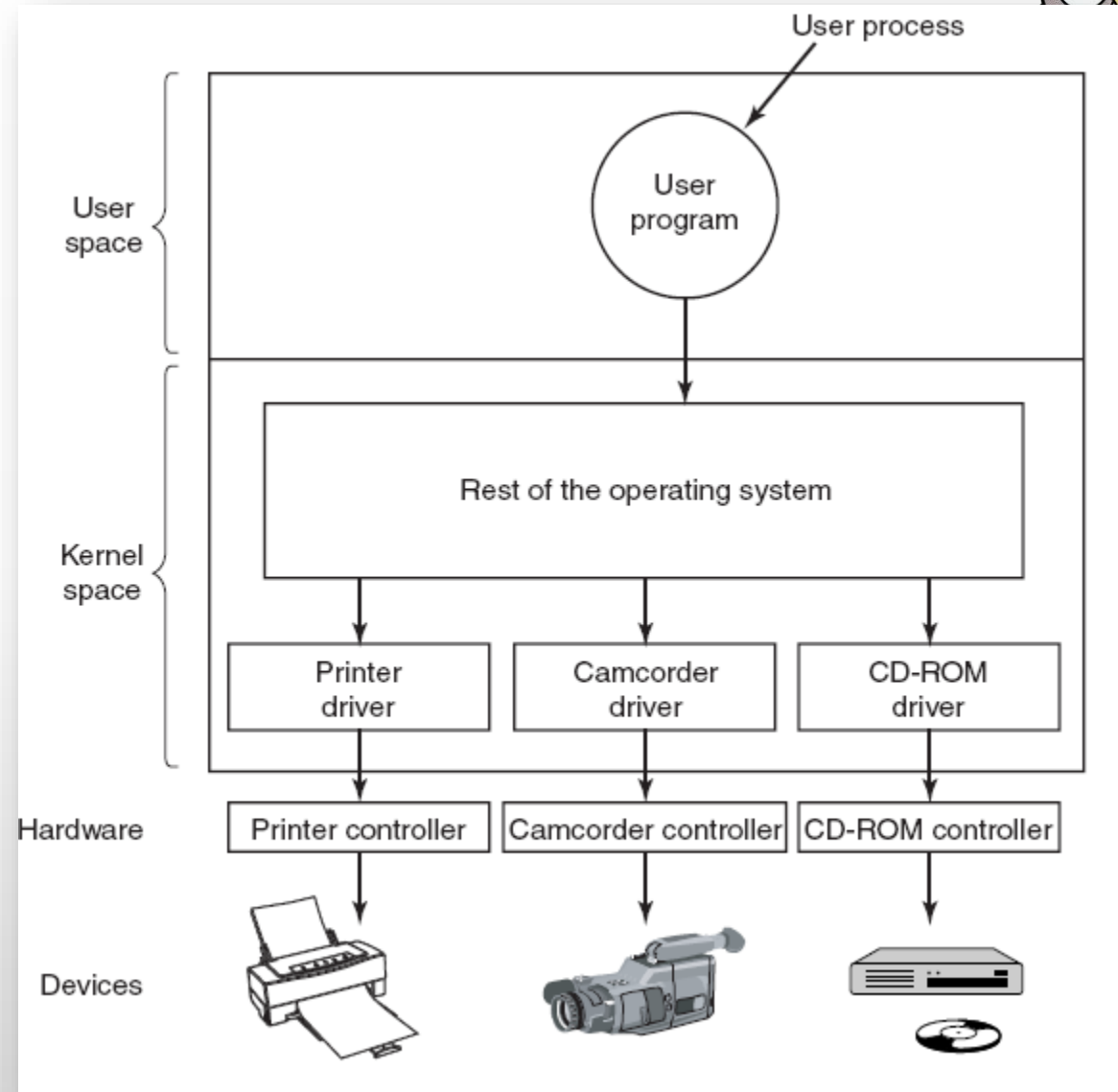
Kesme İşleyicileri (Handler)

- Kesme hizmeti yordamı yürütülür.
- Sırada hangi sürecin çalıştırılacağı seçilir.
- Bir sonraki sürecin çalışması için *MMU* içeriği ayarlanır.
- Yürütülecek süreç için yazmaçlar güncellenir.
- Yeni süreç yürütülür.



Aygıt Sürücüleri

- Sürücü ve aygıt denetleyicisi veri yolu üzerinden haberleşir.





Aygıt Sürücüleri

- Sürücü (*driver*), aygıta özel kodlanmıştır.
- Üretici (*manufacturer*) tarafından sağlanır.
- Çekirdeğe (kernel) yüklenir.
 - Sürücü için kullanıcı alanı (*user space*) daha iyi bir yer olabilir.
 - Sürücü yazılımındaki bir hata, çekirdeği bozabilir.
- İşletim sistemi arayüzüne ihtiyaç var.
 - Blok ve karakter tabanlı aygıt arayüzleri.
 - Sürücüyü kullanmak için sistem çağrıları (*bir blok oku gibi*).



Aygıt Sürücüleri

- Verilen parametrelerin tutarlılığını kontrol eder.
- Mantıksal adresi fiziksel adrese çevirir.
 - *blok numarası -> silindir, kafa, iz, sektör.*
- Aygıt durumunu kontrol eder. Başlatmak zorunda kalabilir.
- Komutları aygıt denetleyicisinin yazmaçlarına koyar.
- Kesme gelene kadar sürücü kendini bloke eder.
- Talep edilen verileri gönderir.
- Aygıtın durum bilgisini döndürür.
- Sürücüler yeniden girilebilir (*reentrant*) yapıdadır.



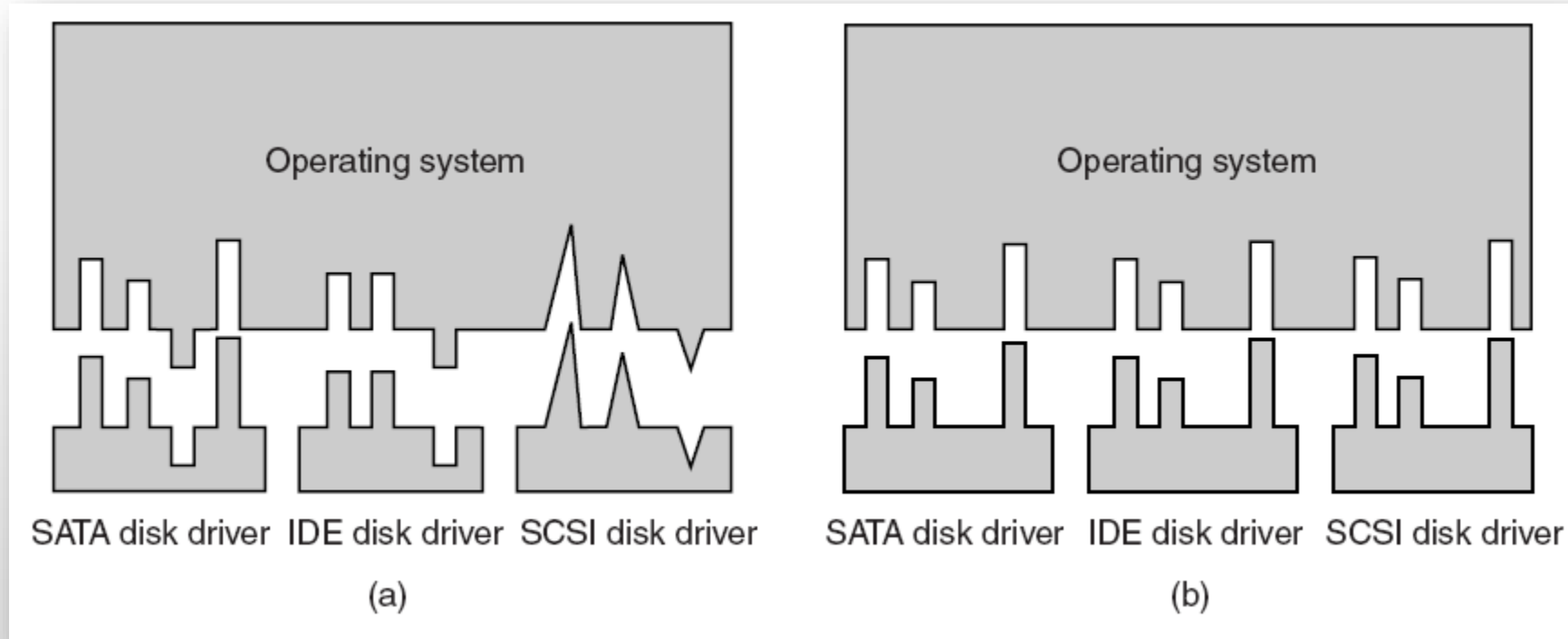
Aygıtdan Bağımsız G/Ç Yazılımı

- Aygıt sürücüleri için tek tip arayüz sağlar.
- Ara bellek kullanarak verileri saklar.
- Oluşan hataları raporlar.
- Aygıtı bir sürece tahsis eder.
- İşlem tamamlandıncaya serbest bırakır.



Aygıt Sürücüleri İçin Tek Tip Arayüz

(a) Standart bir arayüz yoksa, karmaşık. (b) varsa, düzenli.





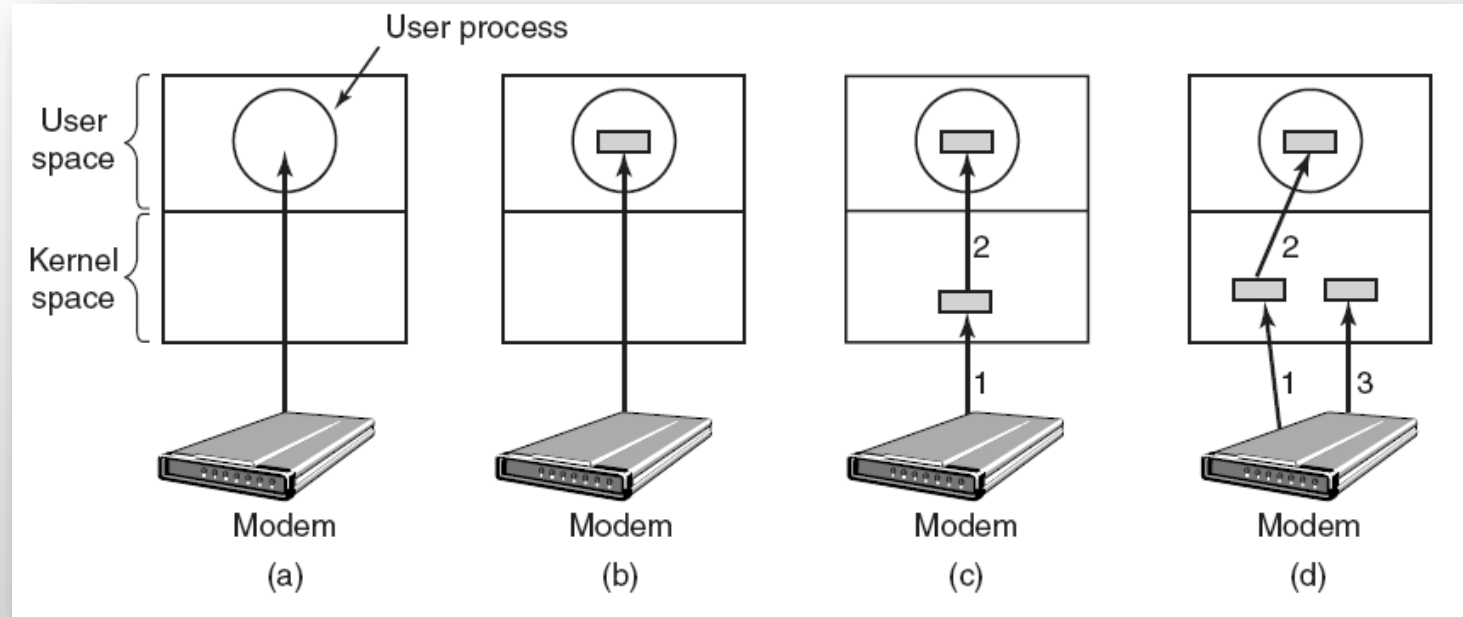
Aygıt Sürücü Fonksiyonları

- Her aygıt sınıfı için sağlanması gereken işlemler.
 - örneğin, oku, yaz, aç, kapat..
- Sürücü, doğru işlemi işaret eden, bir *işaretçi tablosuna* sahip.
- İşletim sistemi, gerekli işlemi çağırmak için tablo adresine ihtiyaç duyar.
- İşletim sistemi, sembolik aygıt adlarını doğru sürücü ile eşler. (*map*)



Ara Belleğe Alma

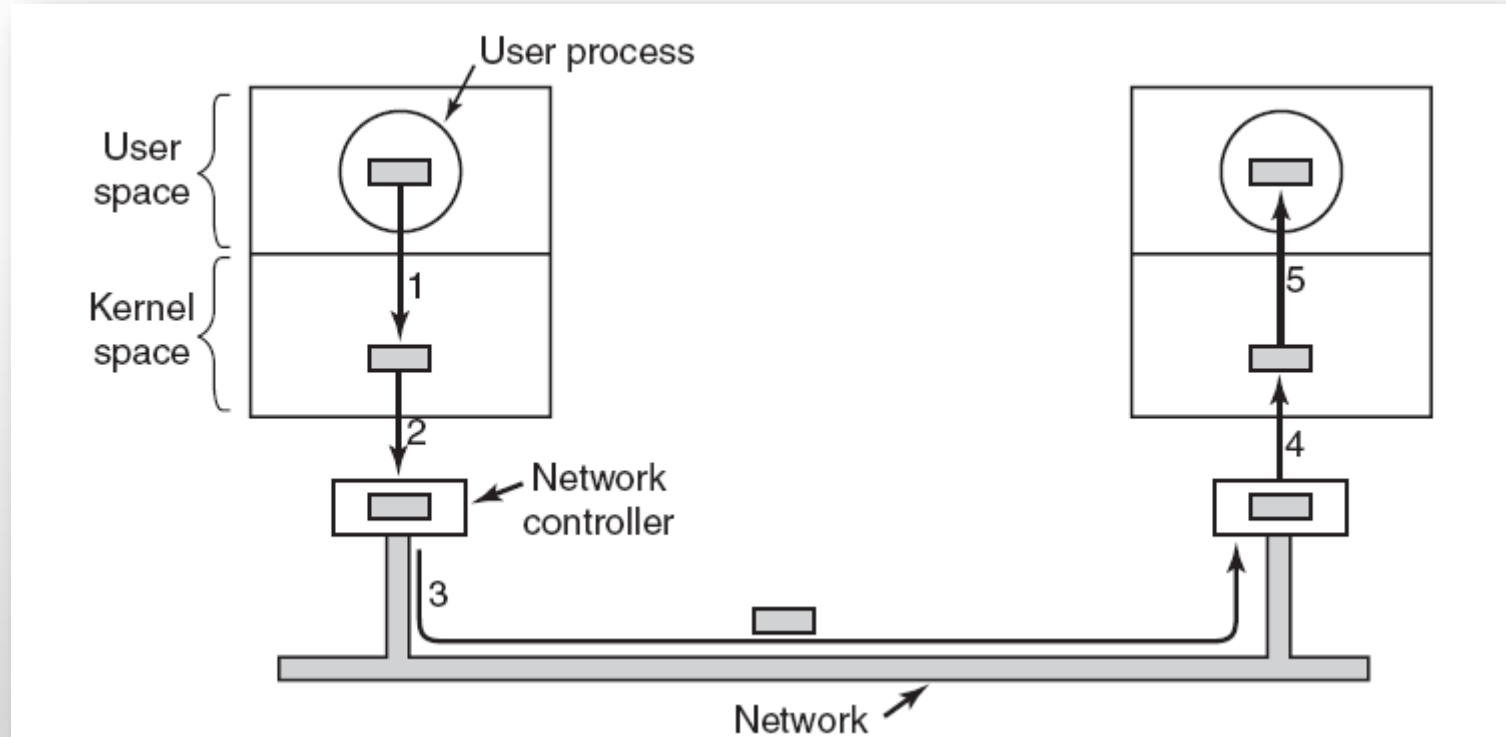
- (a) Tampon bellek (*buffer*) yok.
- (b) Kullanıcı alanında tampon bellek.
- (c) Çekirdekte tampon bellek ve kullanıcı alanına kopyalama.
- (d) Çekirdekte çift tampon bellek kullanımı.





Ara Belleğe Alma

- Ağ paketinin, birçok kopyası olabilir.





Bağımsız Yazılımın Diğer İşlevleri

- Hata raporlama:
 - Sürücü tarafından çözülemeyen yazılım donanım sorunları raporlanır.
- Aynı anda tek bir kullanıcıya hizmet veren aygıtları,
 - tahsis eder ve serbest bırakır. (*CD-ROM, scanner, printer*)
 - aygıt meşgul ise,
 - istek bir kuyruk (*queue*) yapısına eklenebilir.
 - istek iptal edilip, olumsuz cevap dönülebilir.
- Aygıttan bağımsız blok boyutu:
 - İşletim sistemi, aygıtların ayrıntılarını bilmek zorunda değil.



Kullanıcı Alanında G/Ç Yazılımı

- Kütüphane yordamları G/Ç ile ilgilidir.
 - *printf()*, *scanf()*, *read()*, *write()* gibi metotlar sistem çağrılarını yapar.
- Yapılan aygıt istekleri kuyruğa eklenerek, takip edilir.
- Yazıcıdan çıktı alma,
 - Kullanıcı dosyayı oluşturur,
 - Dosya bekleme kuyruğuna koyulur,
 - Arka plan süreci, kuyruğu izler ve dosyayı yazdırır.
- Dosya aktarımlarında da ayrı bir bekleme kuyruğu kullanılır.



Manyetik Diskler

- .

| Parameter | IBM 360-KB floppy disk | WD 18300 hard disk |
|--------------------------------|------------------------|--------------------|
| Number of cylinders | 40 | 10601 |
| Tracks per cylinder | 2 | 12 |
| Sectors per track | 9 | 281 (avg) |
| Sectors per disk | 720 | 35742000 |
| Bytes per sector | 512 | 512 |
| Disk capacity | 360 KB | 18.3 GB |
| Seek time (adjacent cylinders) | 6 msec | 0.8 msec |
| Seek time (average case) | 77 msec | 6.9 msec |
| Rotation time | 200 msec | 8.33 msec |
| Motor stop/start time | 250 msec | 20 sec |
| Time to transfer 1 sector | 22 msec | 17 μ sec |



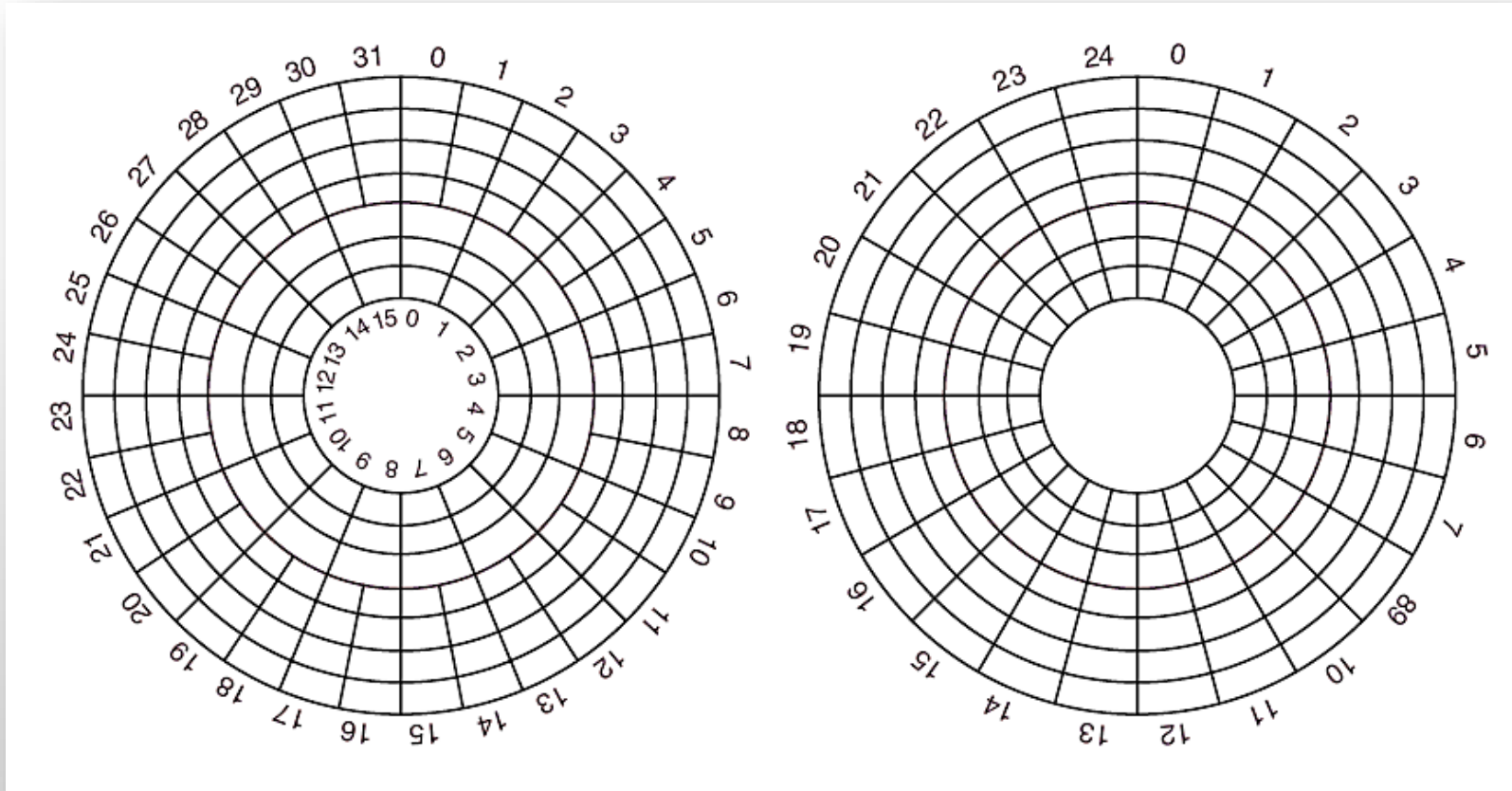
Diskler

- Disk içinde bulunan mikrodenetleyici,
 - Bozuk blokları izler, eşler, kullanılmaması için etiketler.
 - Okunan izin içeriğini önbelleğe alır.
- Bazı diskler aynı anda birden fazla arama (*seek*) yapabilir.
 - Bir plaktan okurken, diğer plağa yazabilir.
- Disk geometrisi, sürücünün kullandığı geometriden farklıdır.
- Denetleyicinin (*silindir, başlık, sektör*) talebi,
 - Gerçek disk geometrisine çevrilir.



Manyetik Diskler

- Bir diskin (a) fiziksel geometrisi. (b) sanal geometrisi.





SSD (Solid State Drive) Diskler

- Katı hal sürücüsü diskler, *NAND flash* bellek kullanır.
- Hareketli (*mekanik*) parçaları yoktur.
- Güvenilir ve daha hızlı veri erişimi sunar.
- Dayanıklılık, düşük güç tüketimi, sessiz çalışma sunar.
- Her bir hücrenin sınırlı sayıda yazma ömrü vardır.
- *Hard Disk Drive*'dan daha pahalıdır.



NVMe (Non-Volatile Memory Express)

- NVMe yüksek performanslı, ölçeklenebilir, optimize edilmiş bir protokol.
- M.2, SATA ve SAS tabanlı SSD'lere kıyasla
 - Yüksek performans, güvenilirlik sağlar.
 - Düşük gecikme süresi, yüksek bant genişliği sağlar.
 - Kablo ihtiyacını ortadan kaldırır.
 - Daha pahalıdır.
- 64 *KB*'a kadar çoklu paralel komut dizisi destekler.
- Geriye dönük uyumlu değildir.
- Özel sürücü ve donanım desteği gerektirir.



Ucuz Disklerin Artık Dizisi (RAID)

- *Redundant array of inexpensive disks.*
- SLED (*Single large expensive disk*), tek, büyük ve pahalı disk.
- Tek bir diske kıyasla paralel G/Ç işlemleri sağlar.
- İşletim sistemine tek bir disk gibi görünen bir grup disk.
- SCSI diskleri sıklıkla kullanılır, ucuzdur.
- Denetleyici başına 7 tane disk kullanılabilir.
- Seviye 0'dan 7'ye kadar farklı mimariler kullanılır.

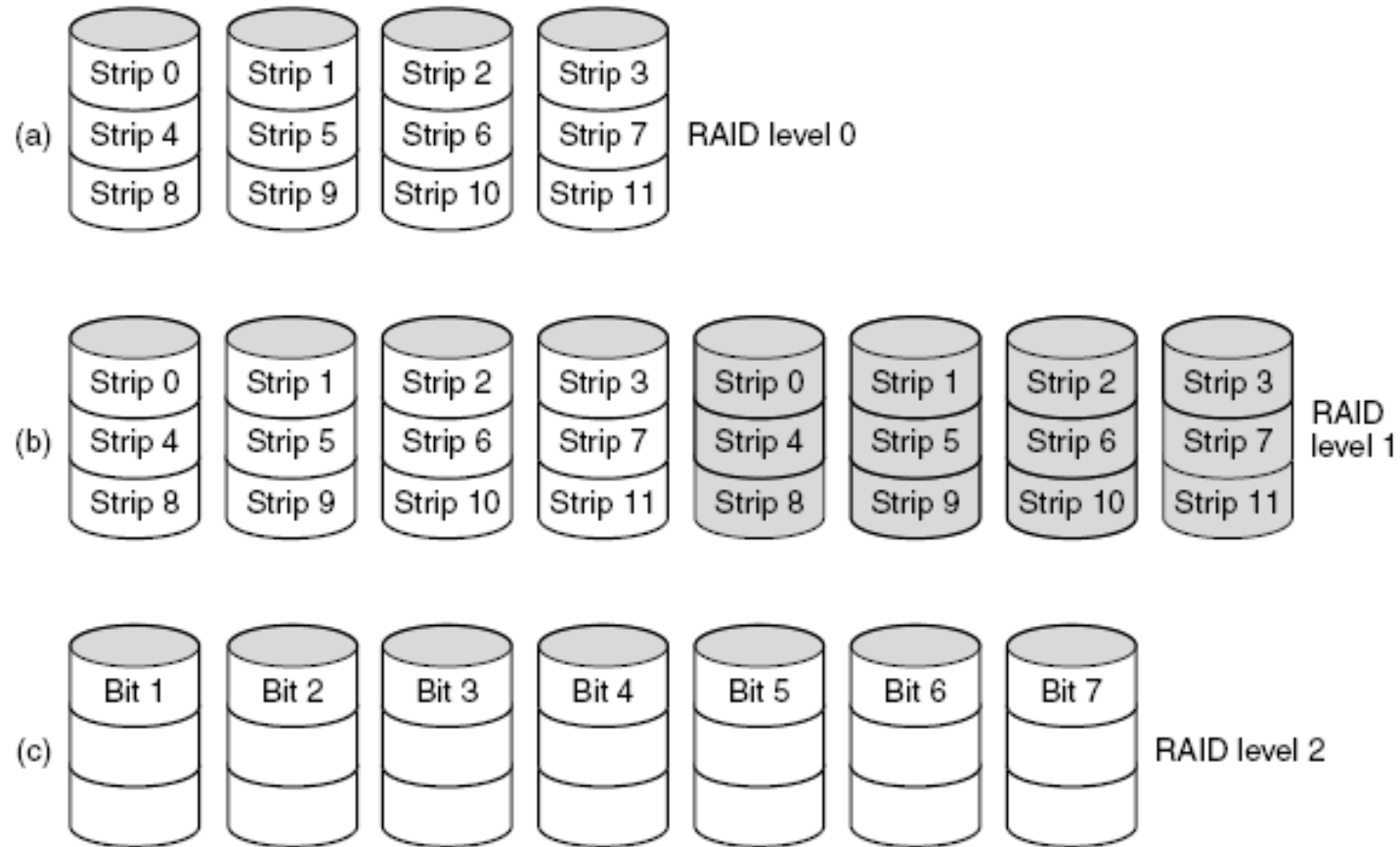


RAID Seviyeleri

- **RAID 0**, şerit (*strip*) başına k sektör şeridi kullanır.
 - Ardışık şeritler farklı disklerde bulunur.
 - Ardışık şeritlere paralel olarak yazma/okuma yapılır.
 - Büyük boyutlu istekler için iyi.
- **RAID 1**, diskleri çoğaltır.
 - Yazma işlemleri iki kez yapılır.
 - Okuma işlemleri her iki diski de kullanabilir.
 - Güvenilirliği artırır.
- **RAID 2**, tek tek sözcüklerle (*word*) çalışır, sözcük + ecc'yi disklere yazar.
 - Paralelliği sağlamak için kollar (*arm*) senkronize edilmeli.



RAID Seviyeleri (0-2)





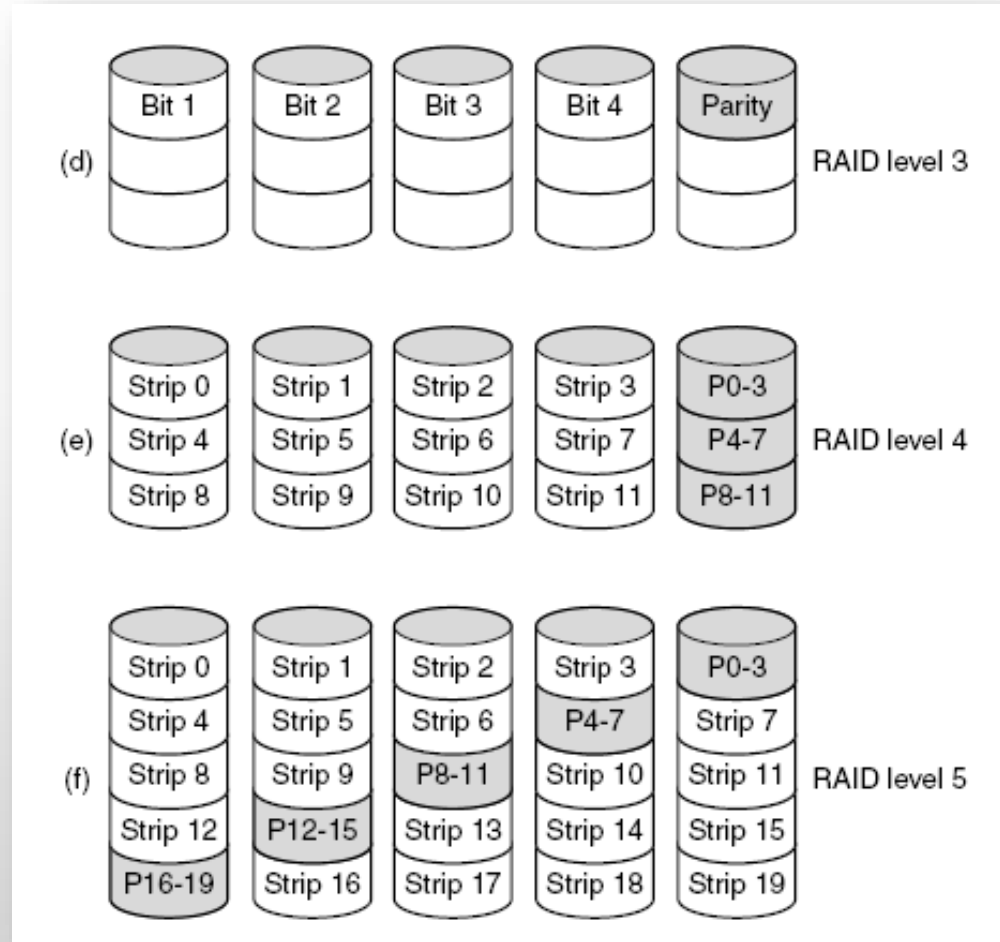
RAID Seviyeleri

- **RAID 3,**
 - Tüm eşlik (*parity*) bitleri *tek* bir sürücüye gider.
 - RAID 2 gibi çalışır.
- **RAID 4,**
 - Şeritler ile çalışır.
 - Şeritler için eşlik bitleri *aynı* sürücüye yazılır.
- **RAID 5,**
 - Şeritler için eşlik bitleri *farklı* sürücüye yazılır.



RAID Seviyeleri (3-5)

■ .



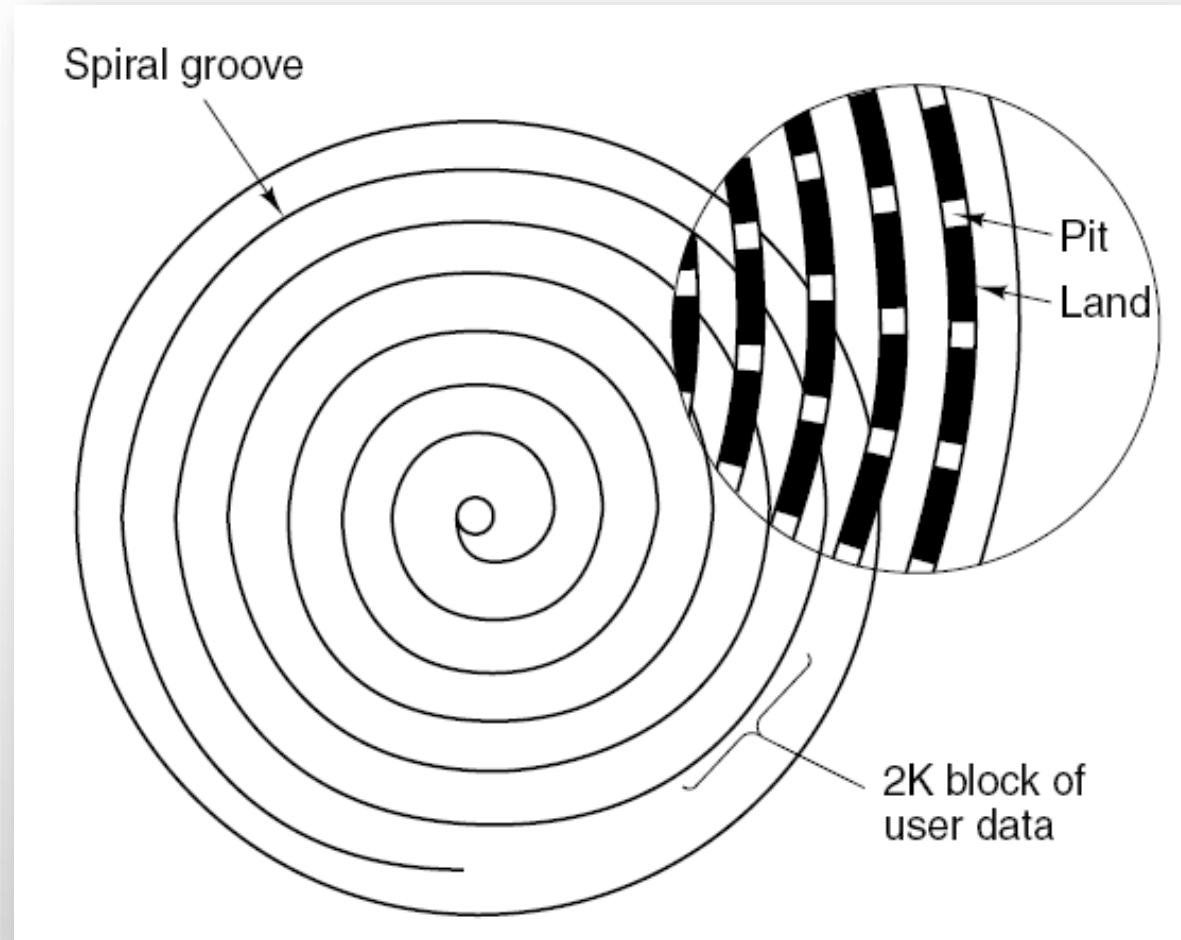


CD-ROM

- Optik diskler, manyetik disklerden daha yüksek yoğunluğa sahiptir.
- Müzik CD'lerinin yüksek üretim hacmi nedeniyle ucuz.
- İlk önce dijital olarak müzik çalmak için kullanıldı.
- Cam ile kaplanmış disk üzerindeki lazer ile delikler yakılır.
- Çukur (yakılmış), ve düz (yakılmamış) spiraller halinde düzenlenir.
- Okuma işlemleri için lazer kullanılır.



CD-ROM Kayıt Yapısı





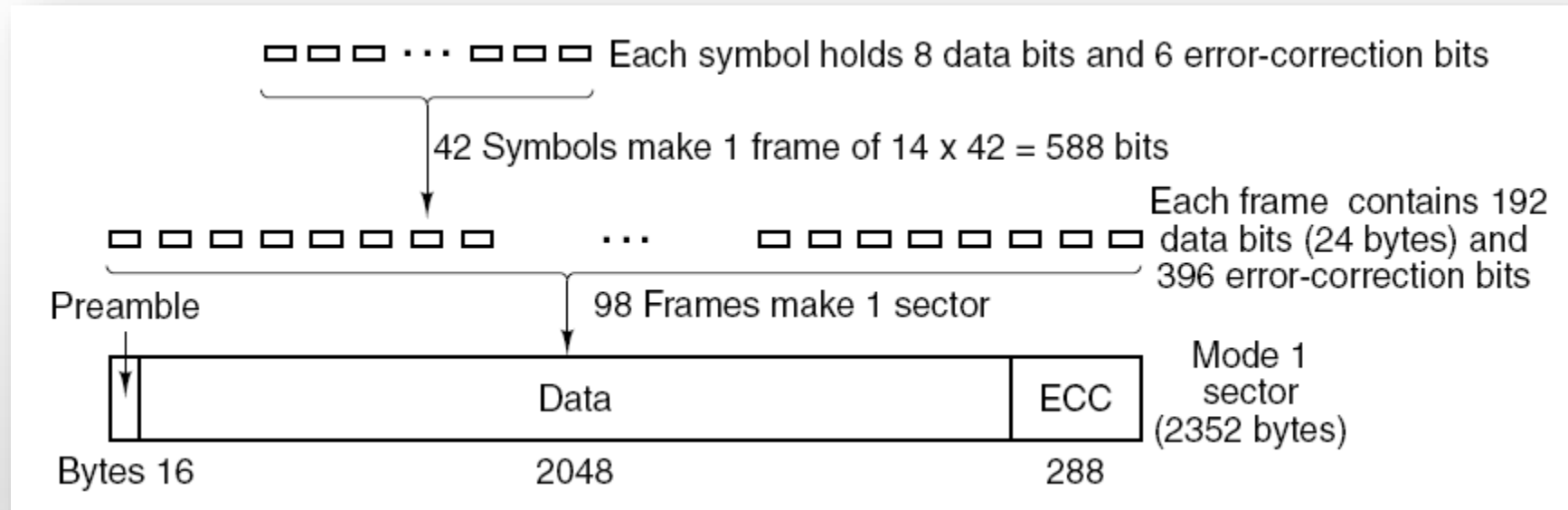
CD-ROM

- Sesin yanı sıra veri depolamak için de kullanılır.
- Hata düzeltme kodu (*error correcting code*) sektöre eklenir.
- Her bayt (*8 bit*), *6 bitlik ECC* ile beraber *14 bitlik sembol* ile kodlanır.
- 42 adet sembol bir çerçeve (*frame*) oluşturur.
- 98 adet çerçeve bir *CD-ROM sektörü* oluşturur.



CD-ROM Mantıksal Veri Düzeni

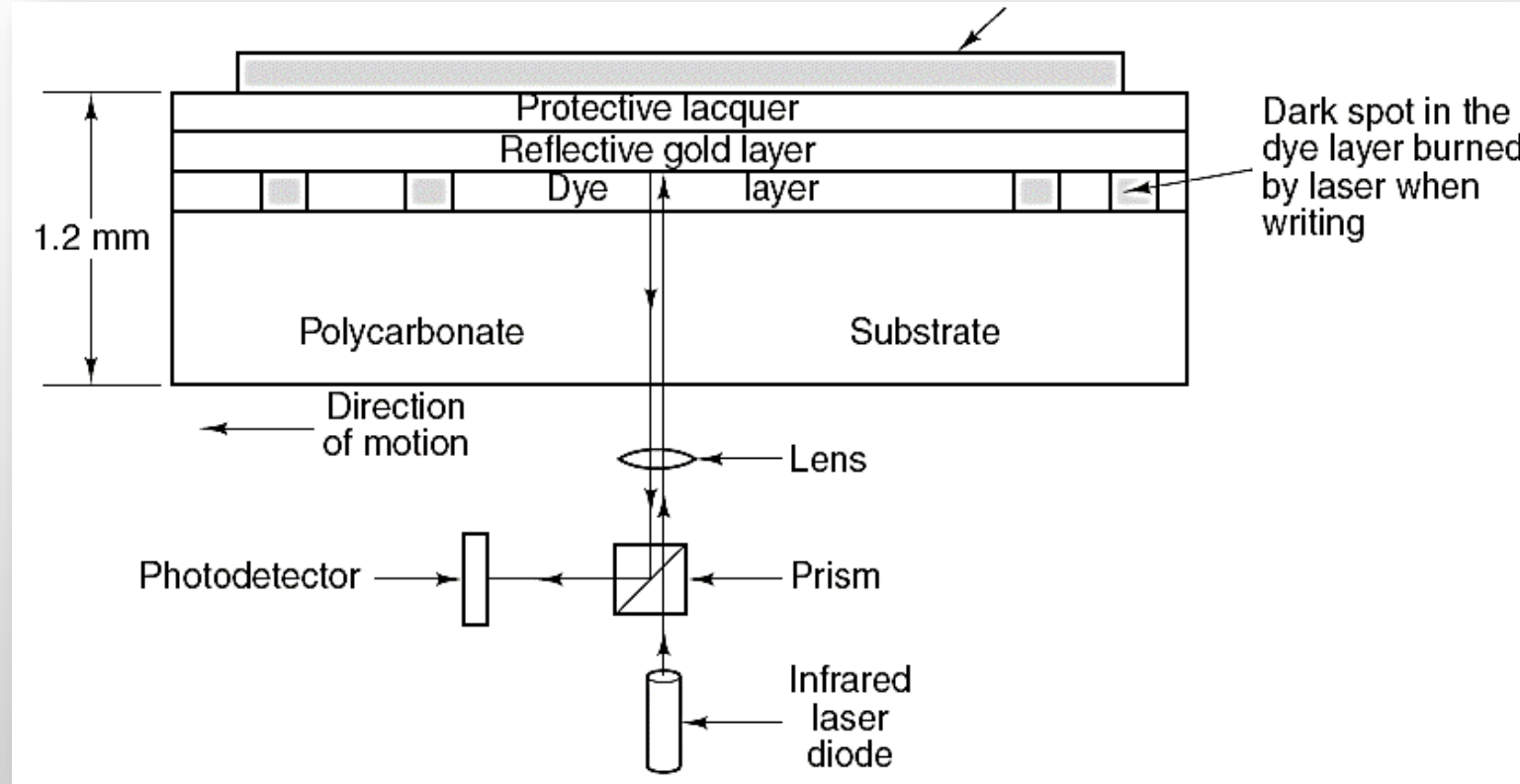
■ .





Kaydedilebilir Compact Disk (CD-R)

- CD-R'nin kesiti. CD-ROM'da koruma tabakası yok, altın yerine alüminyum.





Digital Versatile Disc (Dijital Çok Yönlü Disk)

- **µm:** micron
- Daha küçük çukurları (CD 0.8 µm, DVD 0.4 µm) destekler.
- Daha sıkı bir sarmal (CD 1.6 µm, DVD 0.74 µm).
- Kırmızı lazer (CD 0.78 µm, DVD 0.65 µm).
- DVD'ye standart bir film sığar (133 dakika).
- Hollywood bir diskte daha fazla film istiyor,
 - bu nedenle 4 farklı format var.

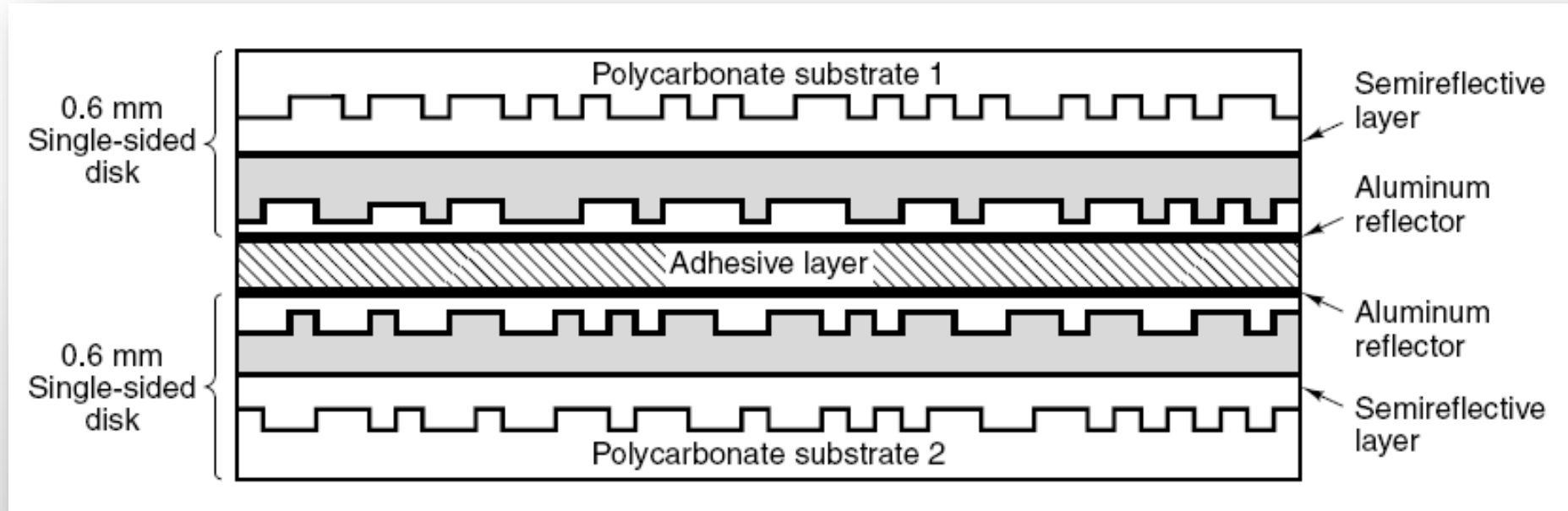


DVD Formatları

- Tek taraflı, tek katmanlı (4.7 GB).
- Tek taraflı, çift katmanlı (8.5 GB).
- Çift taraflı, tek katmanlı (9.4 GB).
- Çift taraflı, çift katmanlı (17 GB).



Çift Taraflı Çift Katmanlı DVD Disk





Sabit Disk Formatı

- Düşük seviye format,
 - Yazılım boş diskteki iz (*track*) ve sektörleri (*sector*) yerleştirir.
- Üst düzey format,
 - Bölümler (*partitions*).



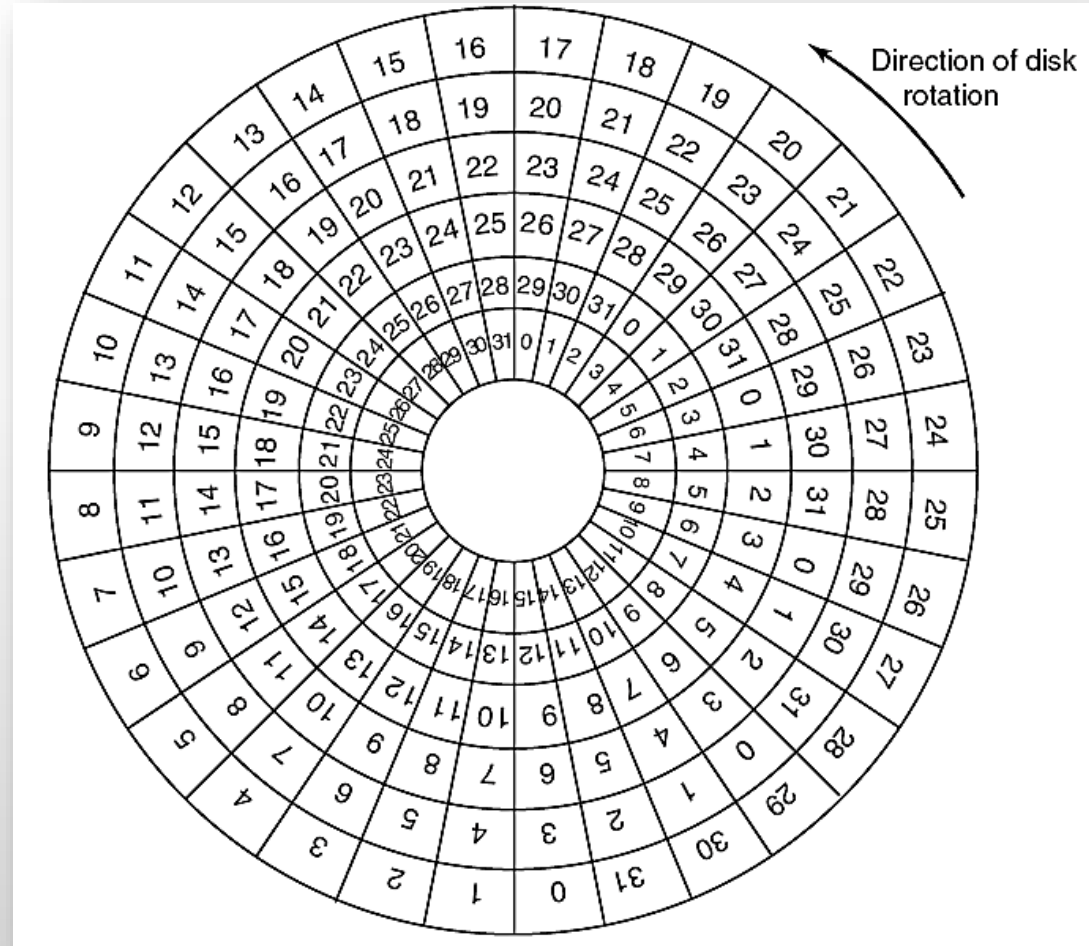
Bir Disk Sektörü

- Sektör, temel veri saklama birimidir.
- Genellikle *512 - 4096 bayt* sabit miktarda veri saklar.
- Öncül (*preamble*), sektörünün adresi, durumu gibi bilgileri sağlar.
- ECC,
 - veri yazma ve okuma sırasında oluşacak hataları tespit etmek ve
 - düzeltmek için, sektörlerle eklenen bir koddur.





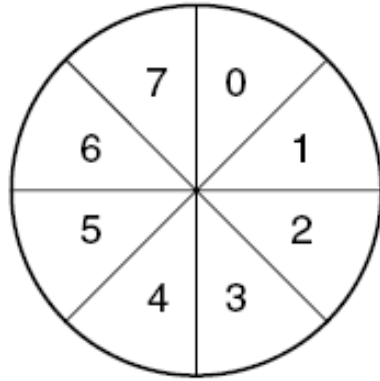
Silindir Eğriliği (Asimetri)



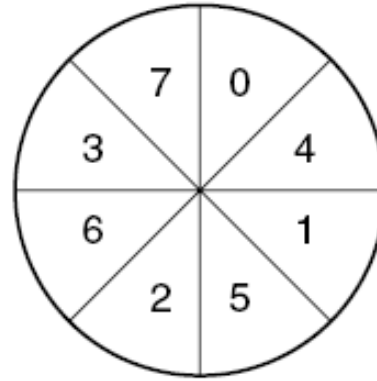


Disk Biçimlendirme

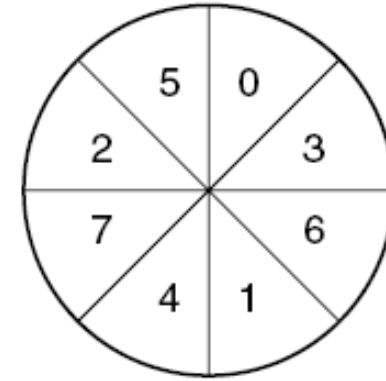
- Serpiştirme (*interleaving*) (a) yok. (b) tek aralıklı. (c) çift aralıklı.



(a)



(b)



(c)



Üst Düzey Format

- High-level formatting is the process of setting up an empty file system on a disk partition or a logical volume and for PCs, installing a boot sector.
- Ana önyükleme kaydı (*master boot record*),
 - Diskte bulunan ilk sektördür.
 - Bölüm tablosuna (*partition table*) sahiptir.
- Bölüm: aynı diske birden fazla işletim sistemi yüklenebilmesini sağlar.
 - Önyükleme bloğuna (*boot block*) sahiptir.
 - Disk 4 adet birincil (primary) bölüme sahip olabilir.
 - Önyükleme (*boot*) yapabilmek için aktif olarak seçilmelidir.



Her bir Bölüm için Üst Düzey Format

- Sektör 0'da ana önyükleme kaydı (*master boot record*).
- Bölüm tablosu (*partition table*).
 - Bölümde (*partition*) hangi dosya sisteminin olduğunu gösterir.
 - Önyükleme bloğu (*boot block*) programı.
 - Yönetmek için boş depolama alanı (*biteşlem veya boş liste*).
 - Kök dizini (*root*).
 - Boş dosya sistemi (*empty file system*).



Bilgisayara Güç Verildiğinde

- *BIOS*, ana önyükleme kaydını (*MBR*) okur.
- Önyükleme programı hangi bölümün aktif olduğunu belirler.
- Aktif bölümden önyükleme sektörünü (*boot sector*) okur.
- Önyükleme sektörü, ikinci bir önyükleme (*boot*) programı yükler.
- Önyükleme programı, işletim sistemi çekirdeğini yükler ve yürütür.



Disk Kolu Zamanlama Algoritması

- Disk kafası hareketlerini en aza indirerek,
 - disk gecikmesini azaltır ve disk verimini artırır.
- Arama süresi (*seek time*):
 - kolun doğru ize ulaşana kadar geçen süre.
- Dönme gecikmesi (*rotational latency*):
 - okuma kafasının doğru sektöre varması için geçen süre
- Veri transfer süresi (*transfer time*):
- $access\ time = seek\ time + rotational\ latency$.
- Sürücü isteklerinin listesini tutar (silindir numarası, istek zamanı).
 - arama süresini optimize etmeye çalışır.



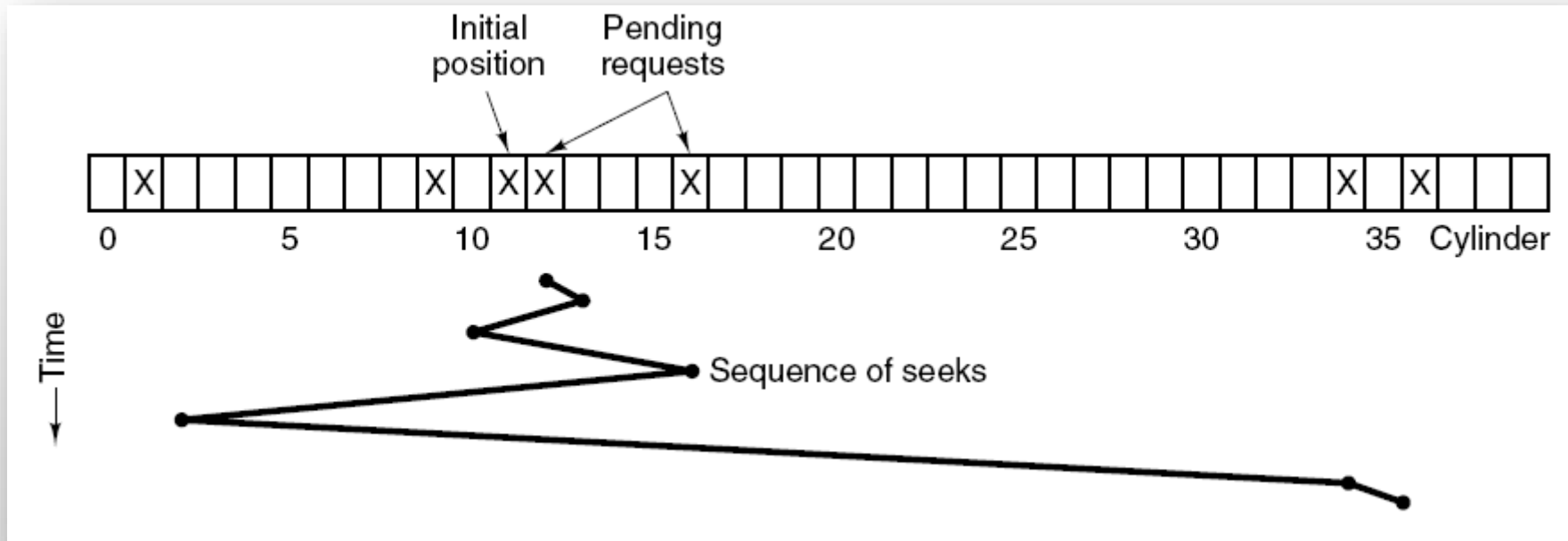
Disk Kolu Zamanlama Algoritması

- FCFS (*first come first served*)
 - İlk gelen ilk hizmet alır.
- SSTF (*shortest seek time first*)
 - En kısa arama süresi olan ilk hizmet alır.
- SCAN (*Asansör Algoritması*)
- C-SCAN (*Döngüsel (Circular) SCAN*)
- LOOK



Önce En Kısa Arama Algoritması

- Kafa 11. silindir üzerinde. Sırasıyla 1,36,16,34,9,12 istekleri gelir.
- *FCFS*: 111 (10+35+20+18+25+3),
- *SSTF*: 61 (1+3+7+15+33+2) silindir hareketi yapar.





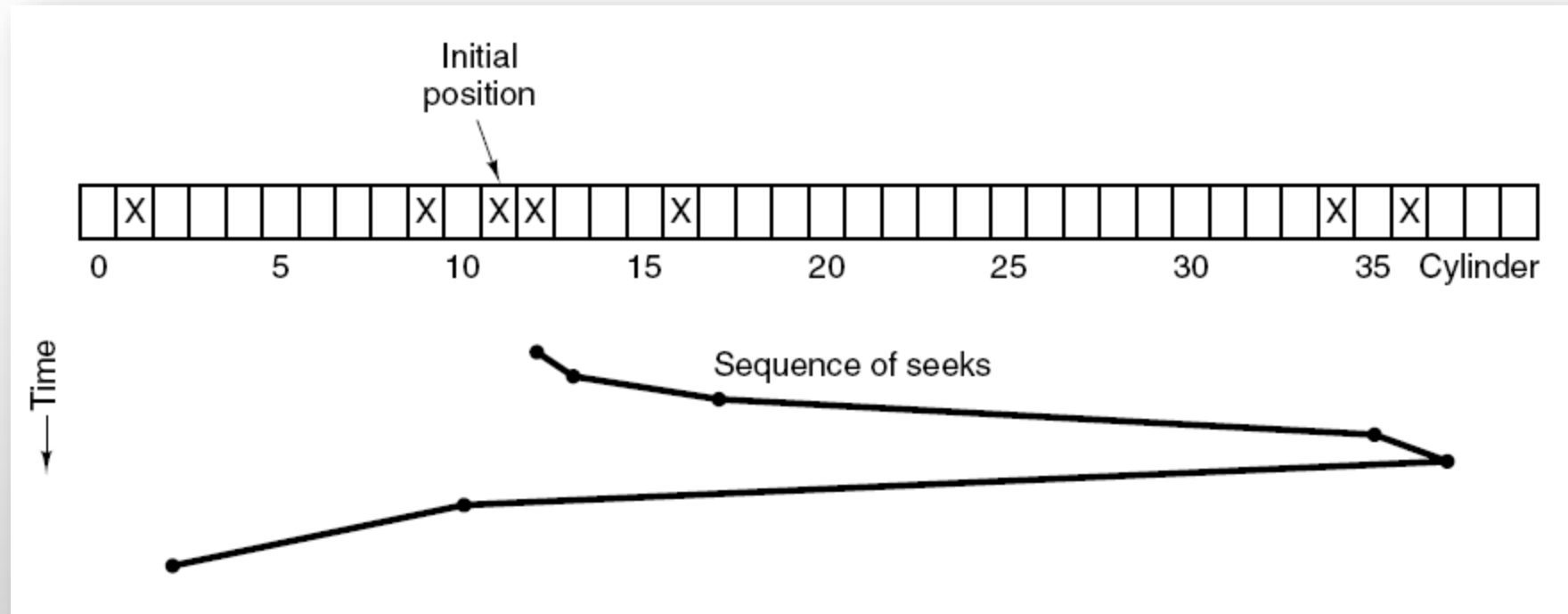
Asansör Algoritması

- Açgözlü (*greedy*) bir algoritma.
- Yoğun kullanımda kafa, diskin bir bölümünde sıkışabilir.
- Talep kalmayana kadar bir yönde devam eder,
 - ardından ters yönde devam eder.
- Gerçek asansörler bu algoritmayı kullanır.
- Önce bir yöne git, sonra ters yönde git.



Asansör Algoritması

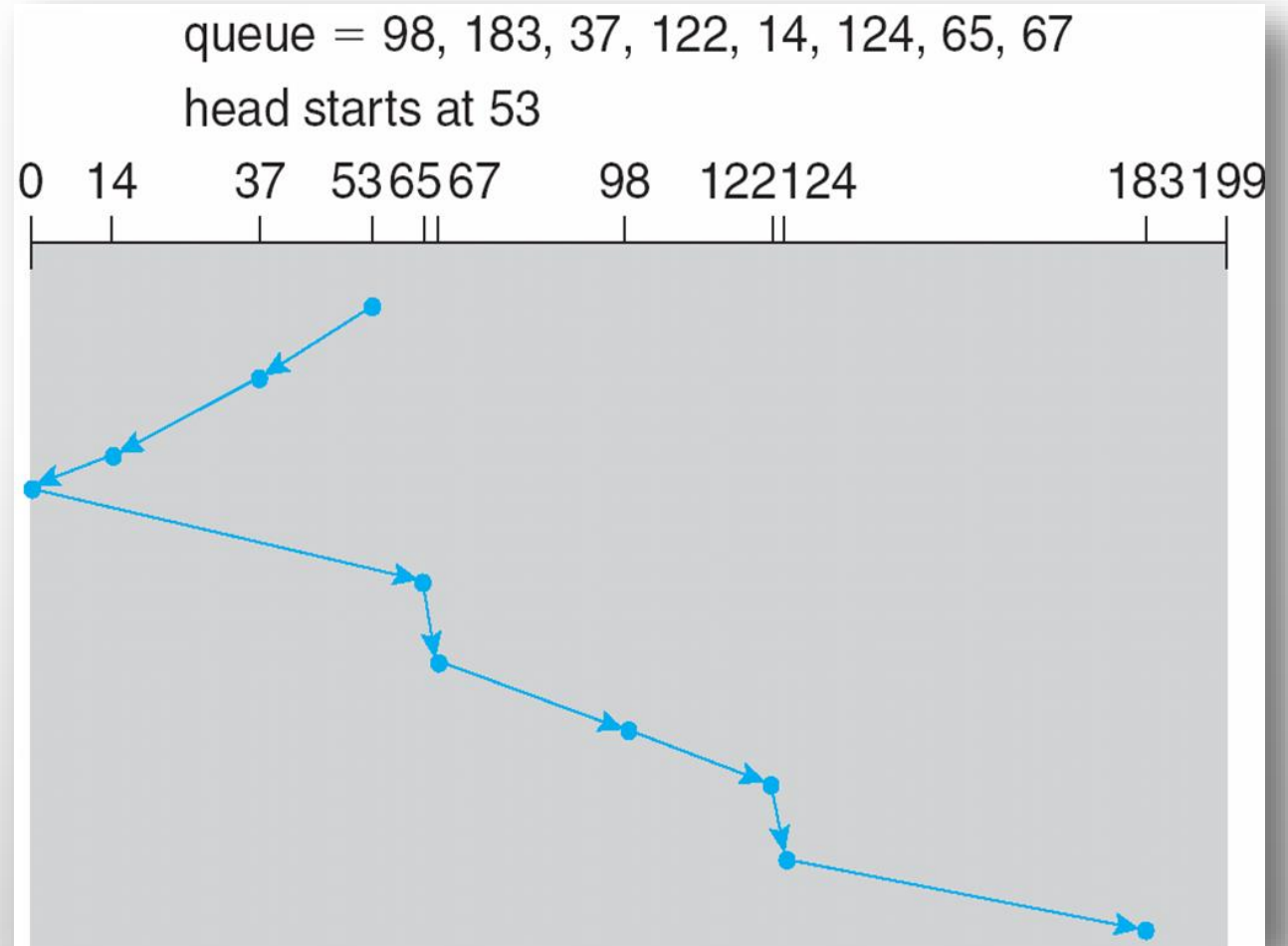
- Kafa 11. silindir üzerinde. Sırasıyla 1,36,16,34,9,12 istekleri gelir.
- *SCAN*: 60 ($1+4+18+2+27+8$) silindir hareketi yapar.





Asansör Algoritması

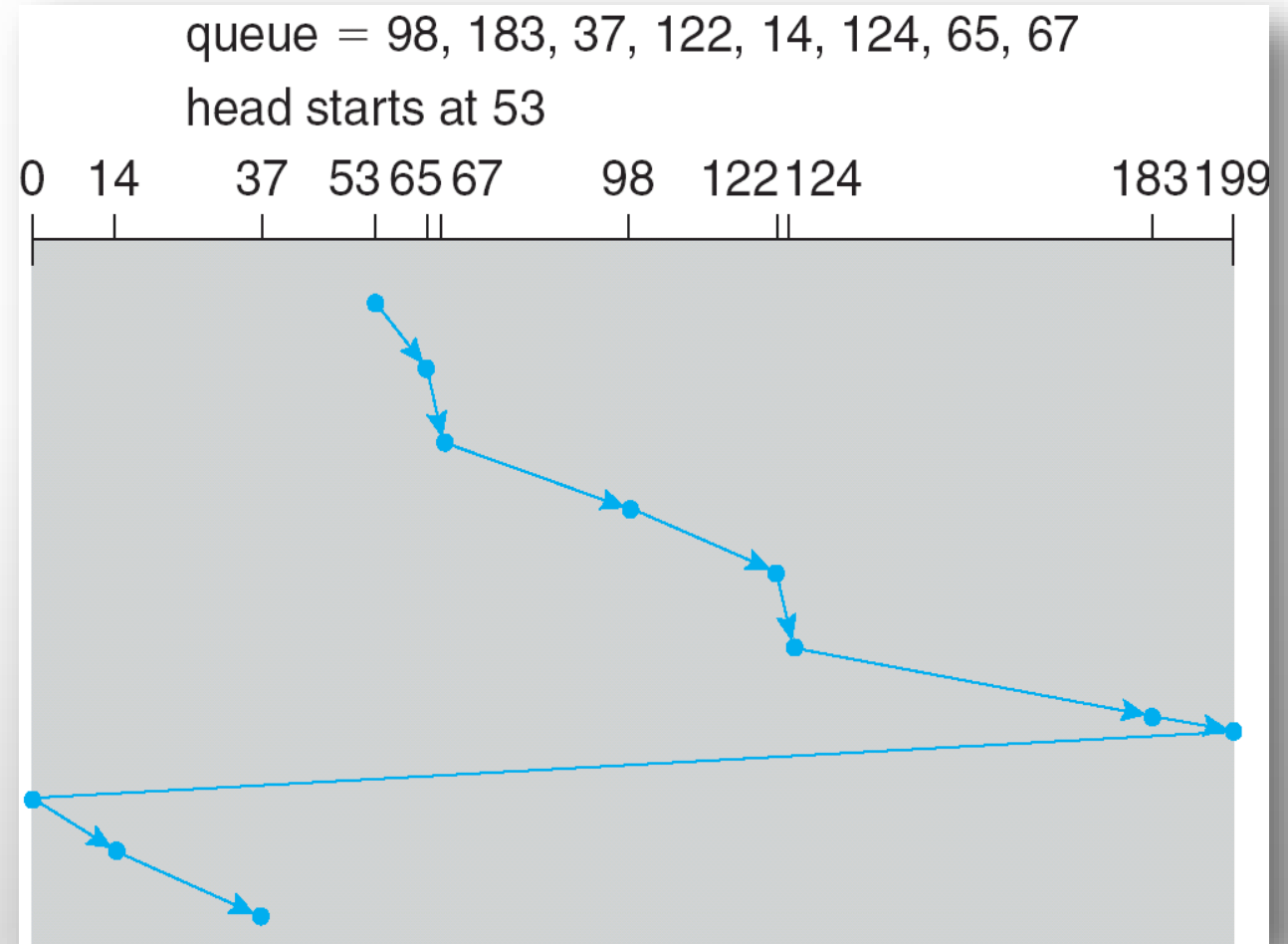
- Toplamda 208 disk silindir hareketi yapılır.





C-SCAN Algoritması

- SCAN algoritmasına benzer.
- Bir yönde istekler bitince, ters yöne dönmez, en baştan tekrar başlar.





Disk Denetleyici Önbelleği

- Disk denetleyicilerinin *kendi önbelleği vardır.*
- Önbellek, işletim sistemi önbelleğinden ayrıdır.
- İşletim sistemi, blokları diskte buldukları yerden,
 - bağımsız olarak önbelleğe alır.
- Denetleyici, okunması kolay olan, o anda disk kafasına yakın olan,
 - ancak zorunlu olarak talep edilmeyen blokları önbelleğe alır.



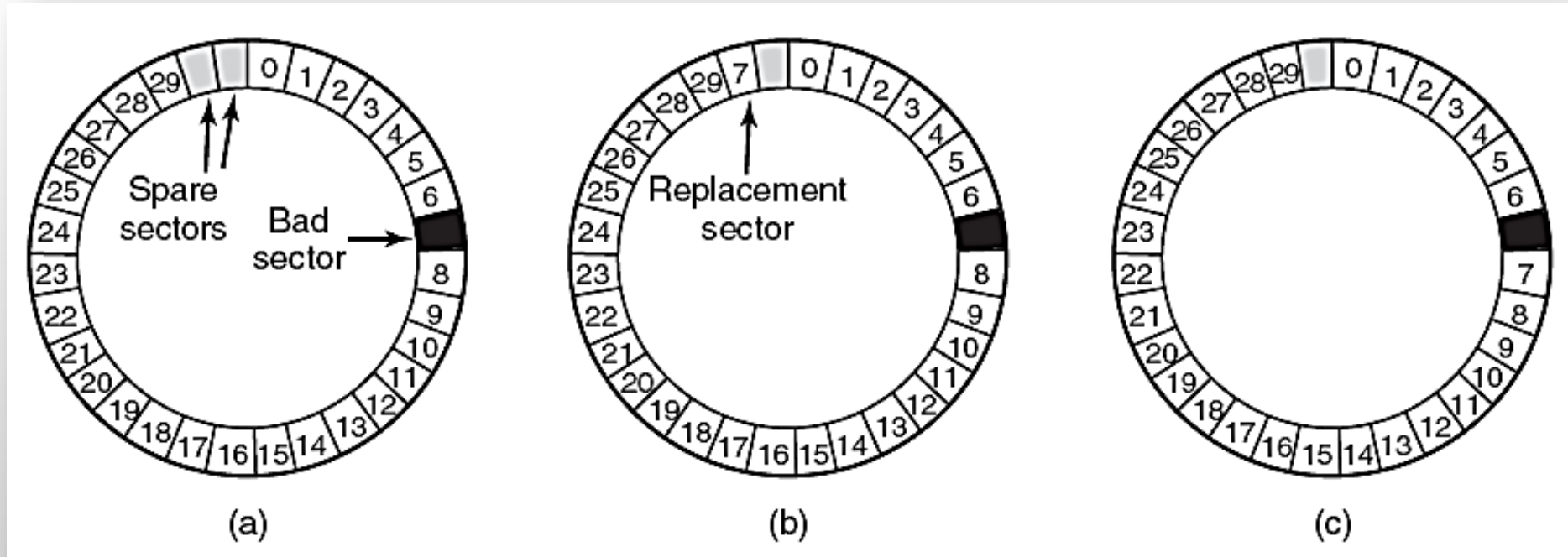
Bozuk Sektörler – Denetleyici Yaklaşımı

- Üretim hatası: yazılan veri ile geri okunan veri uyuşmaz.
- Denetleyici ve işletim sistemi bozuk sektörleri ele alır.
- Denetleyici bozuk sektörlerin bir listesini üreticiden sağlar,
 - bozuk sektörleri, sağlam yedek sektörler ile eşler.
- Denetleyici, bozuk sektörleri disk kullanımında iken fark eder ve eşler.



Bozuk Sektörleri Ele Alma

- (a) Bozuk bir sektöre, iki tane yedek sektöre sahip bir disk izi (*track*).
- (b) Bozuk sektör sağlam bir yedek ile eşlenir.
- (c) Bozuk olan sektör atlanır, diğer sektörler kaydırılır.





Kararlı (Stable) Depolama

- Ya doğru veriler yazılır, ya da eski veriler yerinde kalır.
- Kaybedilemeyecek veriler için gereklidir.
- Özdeş diskler kullanılarak kararlı depolama:
 - Kararlı yazma,
 - Kararlı okuma,
 - Çökmeden kurtarma (*crash recovery*).
 - *RAID* bozulacak sektörlere karşı koruma sağlayabilir.
 - Yazma sırasında çökmelere karşı koruma sağlayamaz.



Varsayımlar

- İki farklı diskte hatalı veri bulunma olasılığı ihmal edilebilir düzeydedir.
- CPU hatası olursa, devam eden yazma işlemi durur.
- Hatalı veriler, okuma işlemi sırasında *ECC* ile tespit edilebilir.



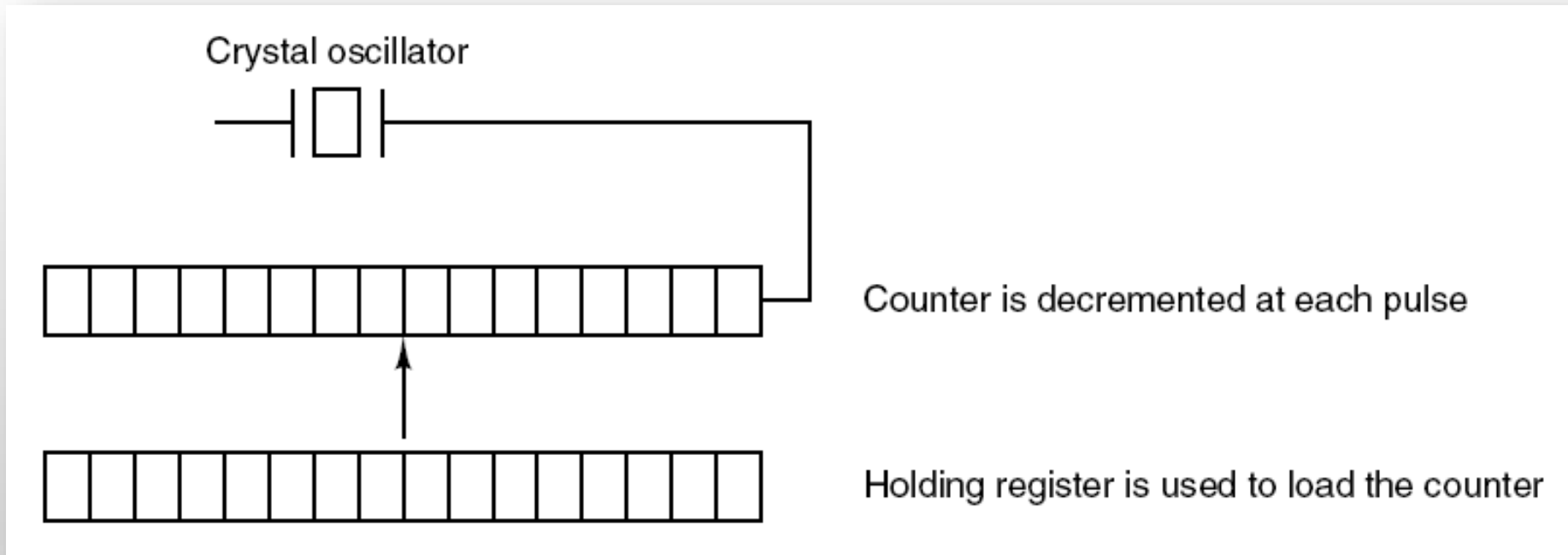
Fikir ve İşlemler

- 2 tane özdeş disk kullanılır.
- **Kararlı yazma:** yaz, oku, karşılaştır.
 - Başarılı ise ikinci diske yaz. Başarısız ise, n defa dene.
 - Hala başarısız ise, başarılı olana kadar yedek sektörleri kullan.
 - Ardından ikinci diske yaz.
- **Kararlı okuma:** doğru *ECC* elde edene kadar birinci diskten n defa oku.
 - Aksi takdirde ikinci diskten oku.
- **Hatadan kurtarma:** iki diskten de oku ve karşılaştır.
 - Bir blokta *ECC* hatası varsa, üzerine doğru bloğu yaz.
 - İkisinde de *ECC* hatası yoksa, birini seç.



Programlanabilir Saat (Clock)

- bir yazmaçta sayaç değeri tutulur.
- sayaç her bir salınım üretici (*oscillator*) darbesinde bir azaltılır.





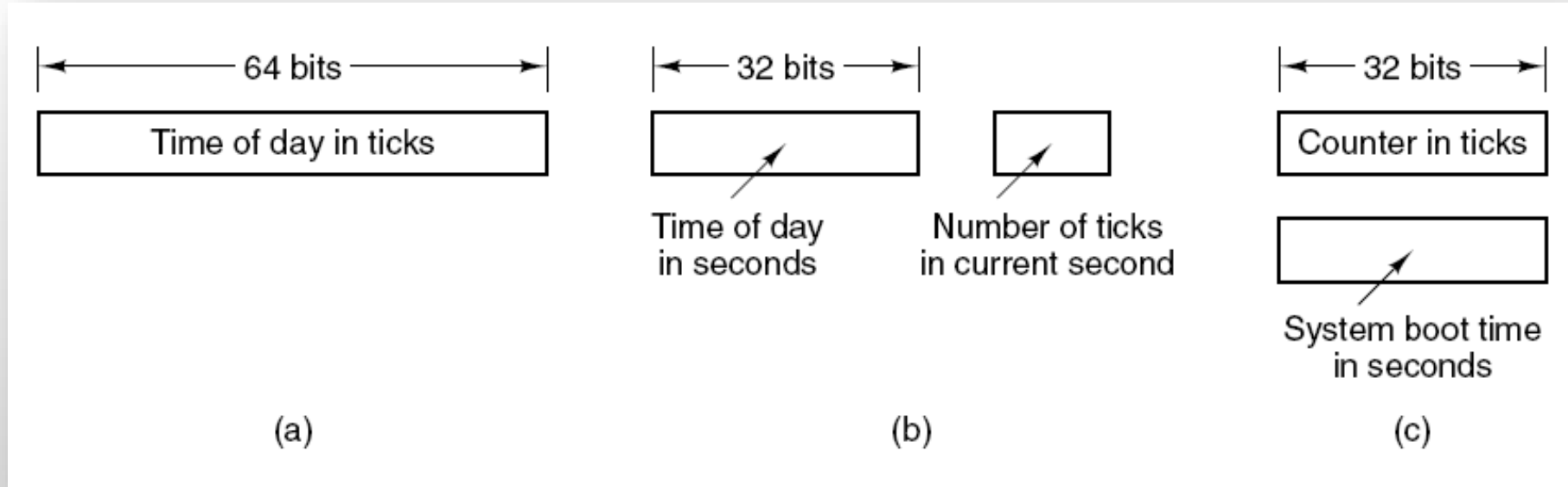
Bir Saat (Clock) Sürücüsünün Görevleri

- Günün saatini sürdürür (*maintain*).
- Süreçlerin izin verilen süreden daha uzun çalışmasını önler.
- CPU kullanımını hesaplar.
- Süreçler tarafından yapılan *alarm()* sistem çağrısını ele alır.
- Bekçi (*watchdog*) uygulaması için zamanlayıcılar (*timers*) sağlar.



Saat Yazılımı

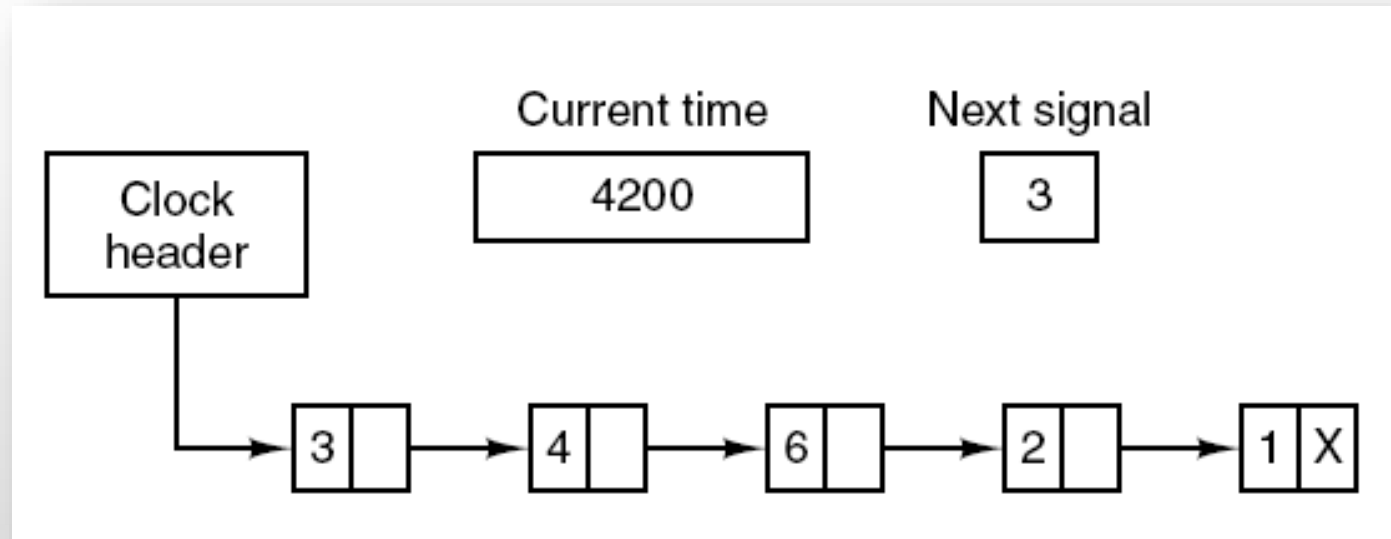
- Günün saati üç yolla saklanabilir. (a) 64 bitlik yazmaçta tık (*tick*) sayısı tutulur. (b) 32 bitlik yazmaçta saniye bilgisi, ayrı bir 32 bitlik yazmaçta tık (*tick*) sayısı tutulur. (c) 32 bitlik yazmaçta tık (*tick*) sayısı, ayrı bir 32 bitlik yazmaçta sistem ayağa kalkma zamanı tutulur.





Saat Yazılımı

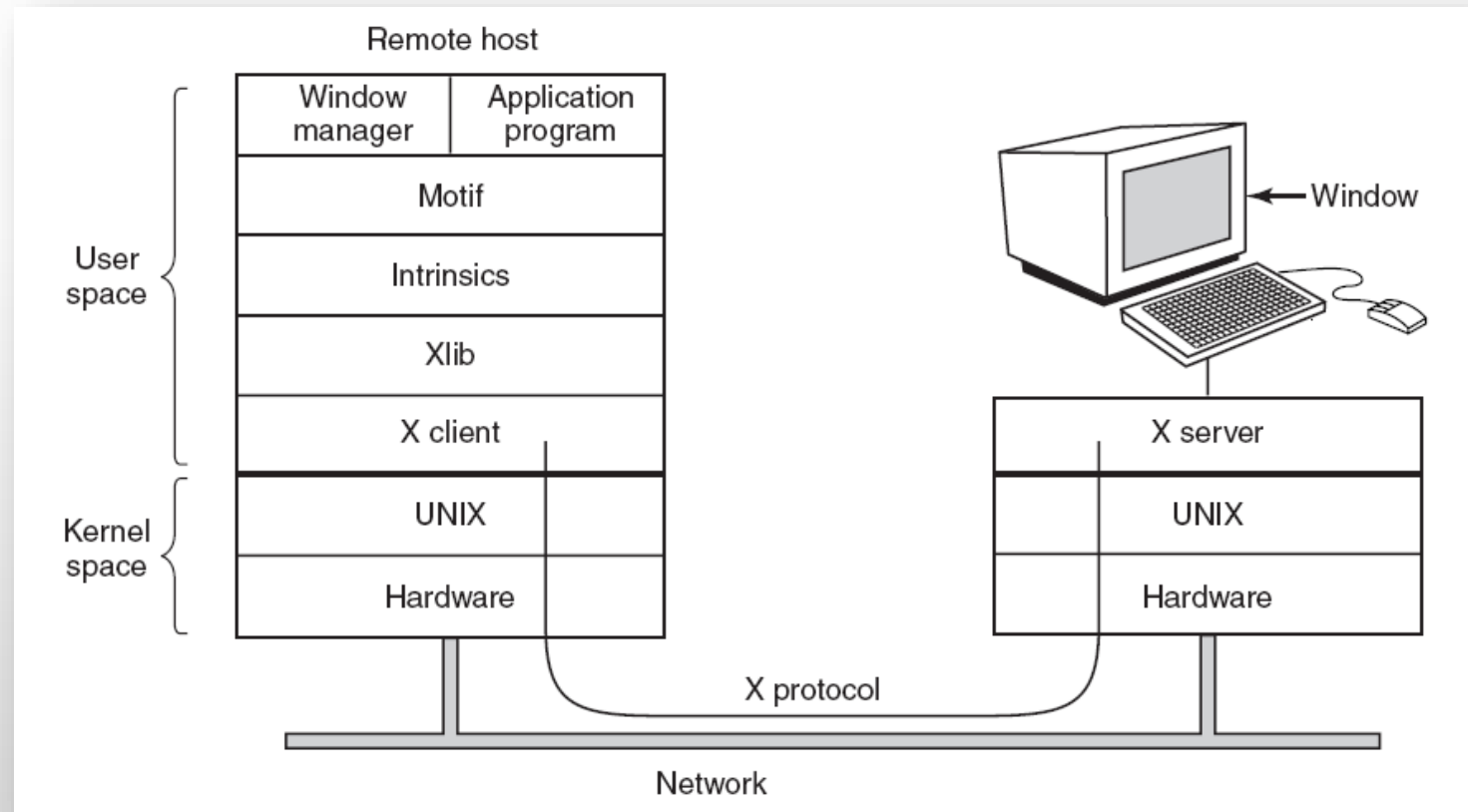
- Tek saat (*clock*) ile birden çok zamanlayıcı (*timer*) kullanılabilir.





X Pencere Sistemi

- X Pencere (*window*) sisteminde istemci ve sunucular.



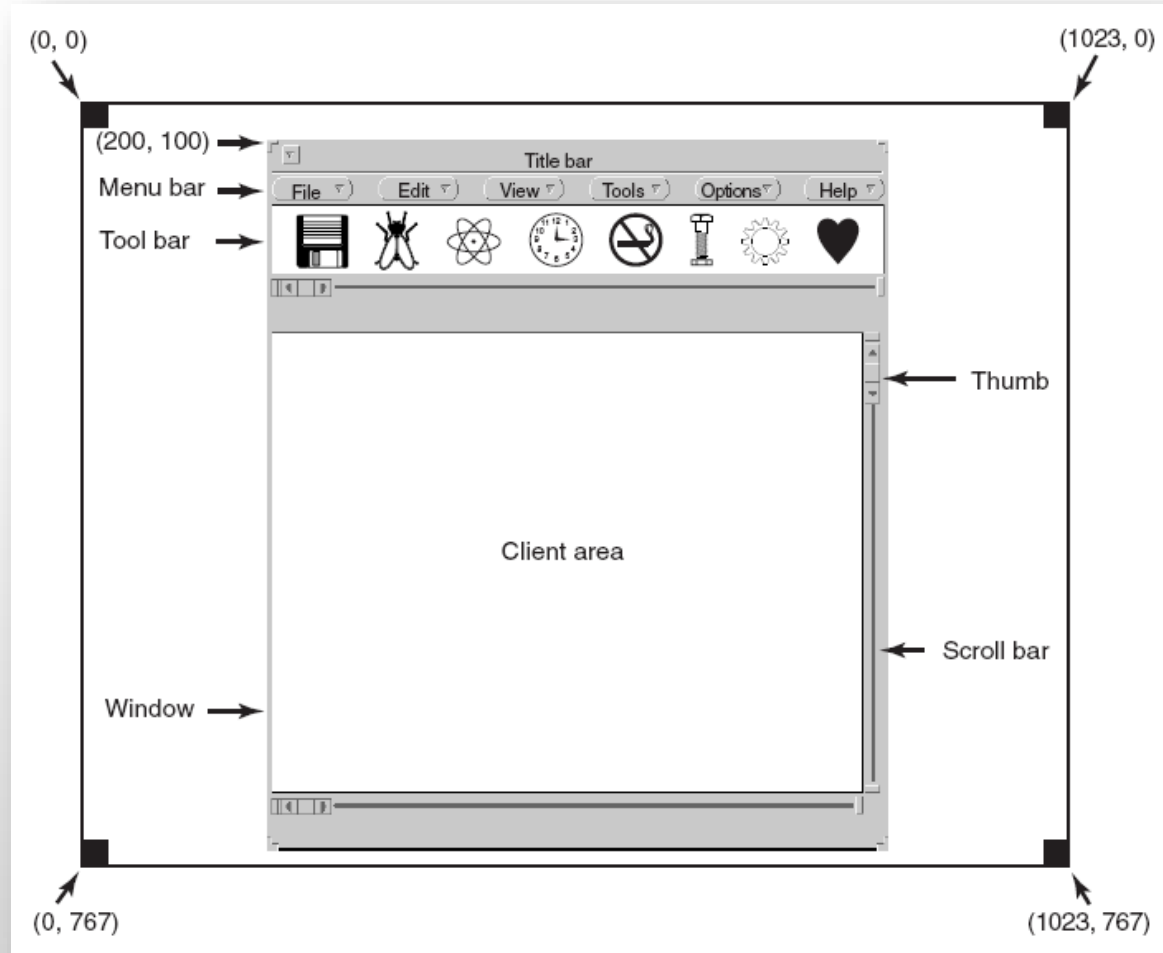


X Pencere Sistemi

- İstemci ve sunucu arasındaki mesaj türleri:
 - Programdan iş istasyonuna çizim komutları.
 - Program sorgularına iş istasyonu tarafından yanıtlar.
 - Klavye, fare ve diğer etkinlik bildirimleri.
 - Hata mesajları.



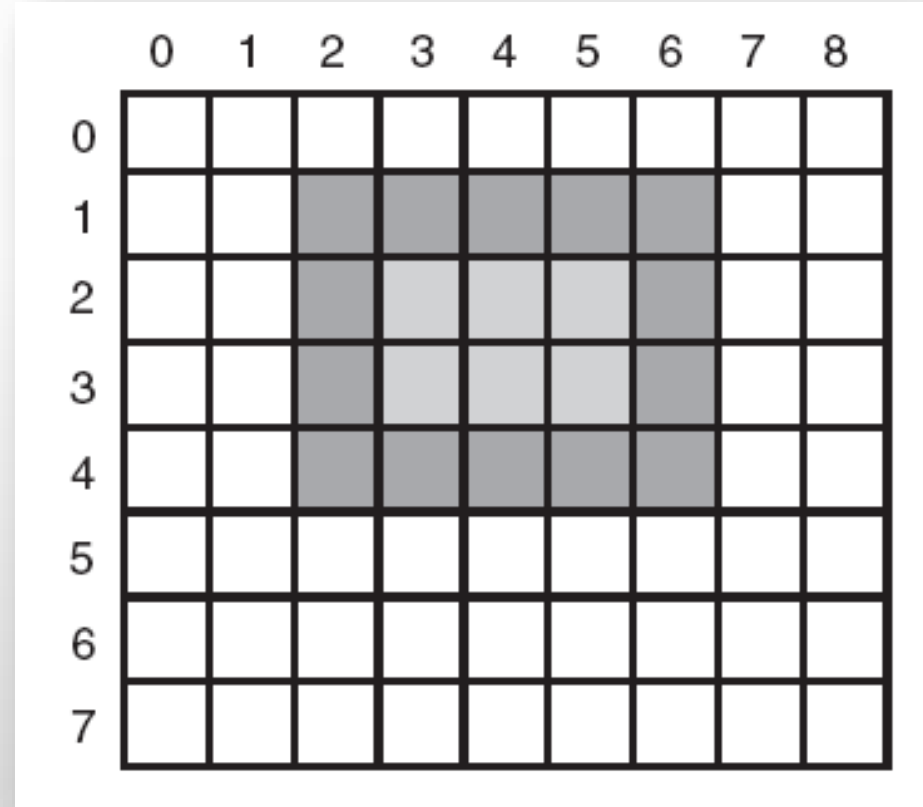
Kullanıcı Ara Yüzü (Örnek Pencere)





Biteşlem (Bitmap)

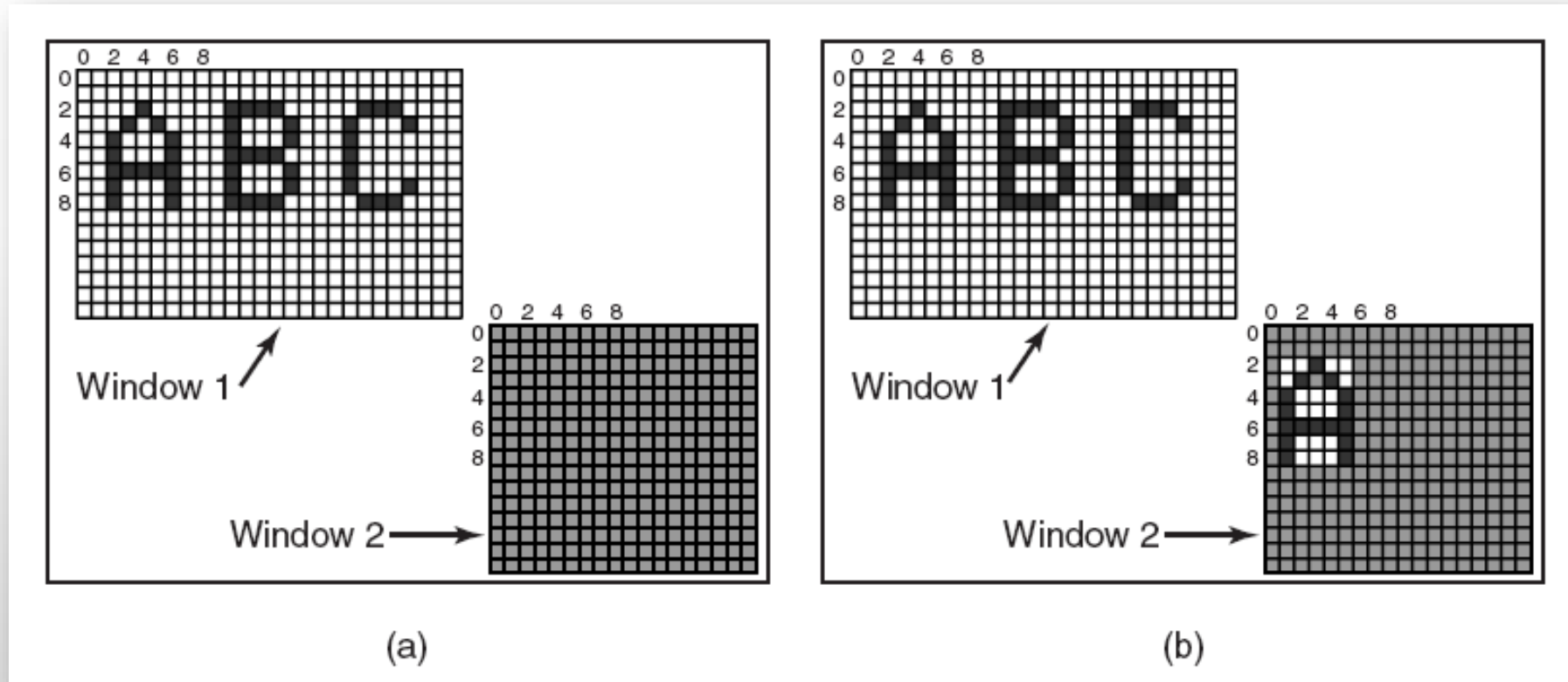
- Her kutu bir *pikseli* temsil eder.





Biteşlem

- *BitBlt* kullanarak bit eşlemleri. (a) önce. (b) sonra.





Biteşlem

- Farklı nokta boyutlarında (*point size*) karakter ana hatları.

20 pt: abcdefgh

53 pt: abcdefgh

81 pt: abcdefgh



İnce İstemciler (Thin Clients)

- İnce istemci protokolü,
 - bir istemcinin uzak bir sunucudaki kaynaklara erişmesine izin verir.
- İstemci ile sunucu arasında,
 - düşük bant genişliğine sahip,
 - düşük gecikmeli bir bağlantı sağlar.
- Kısıtlı kaynağa sahip, düşük performanslı ortamlar için ideal.
- Düşük maliyetli, az bakım gerektiren bir çözüm sağlar.

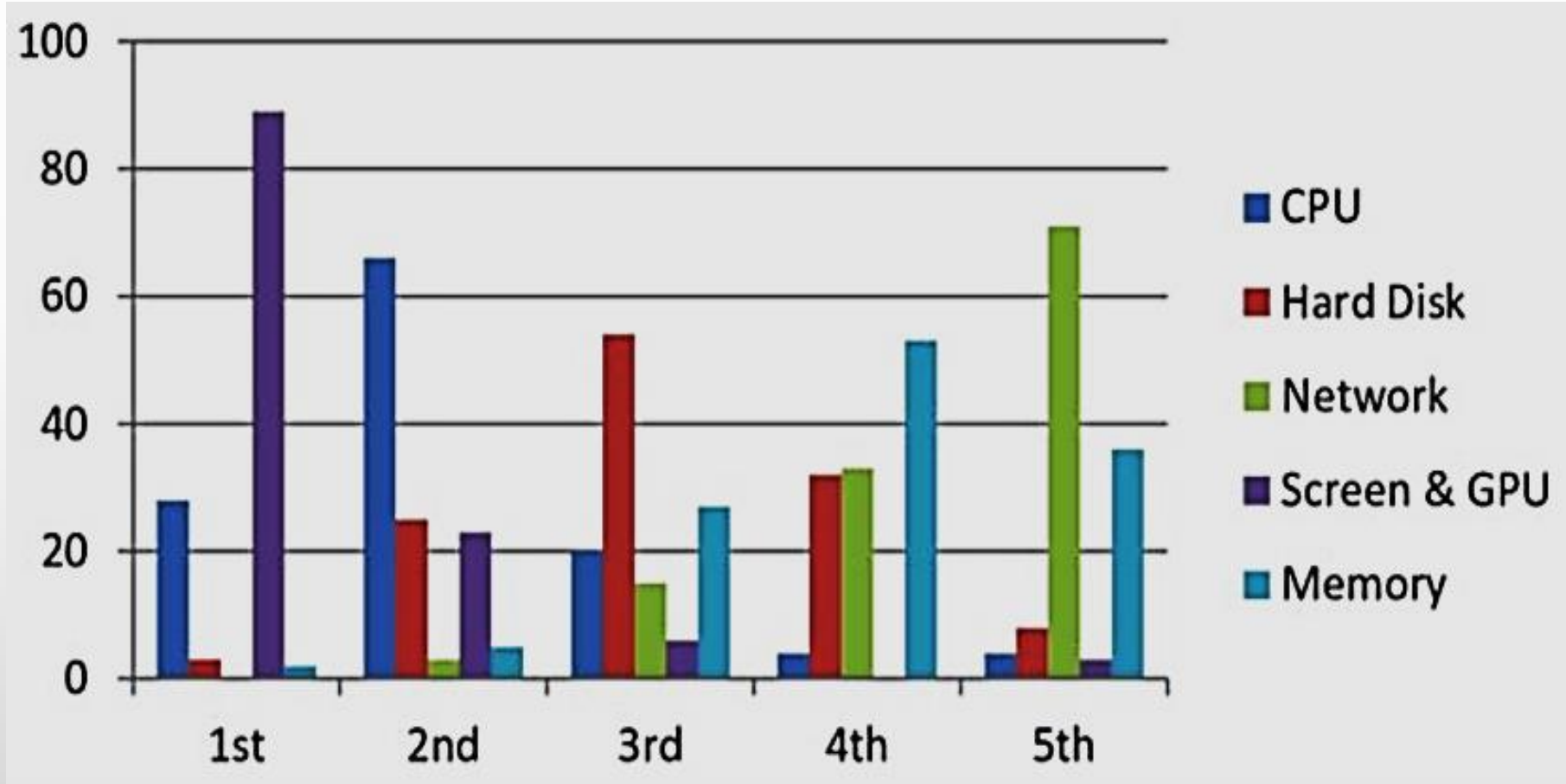


THINC (Thin Client) Protokolü

| Komut | Açıklama |
|--------|--|
| RAW | Verilen pozisyonun ham piksel verilerini görüntüler. |
| COPY | Çerçeve arabellek alanını belirtilen koordinatlara kopyalar. |
| SFILL | Bir alanı verilen piksel renk değeriyle doldurur. |
| PFILL | Bir alanı verilen piksel deseniyle doldurur. |
| BITMAP | Bit eşlem görüntüsü kullanarak bir alanı doldurur. |

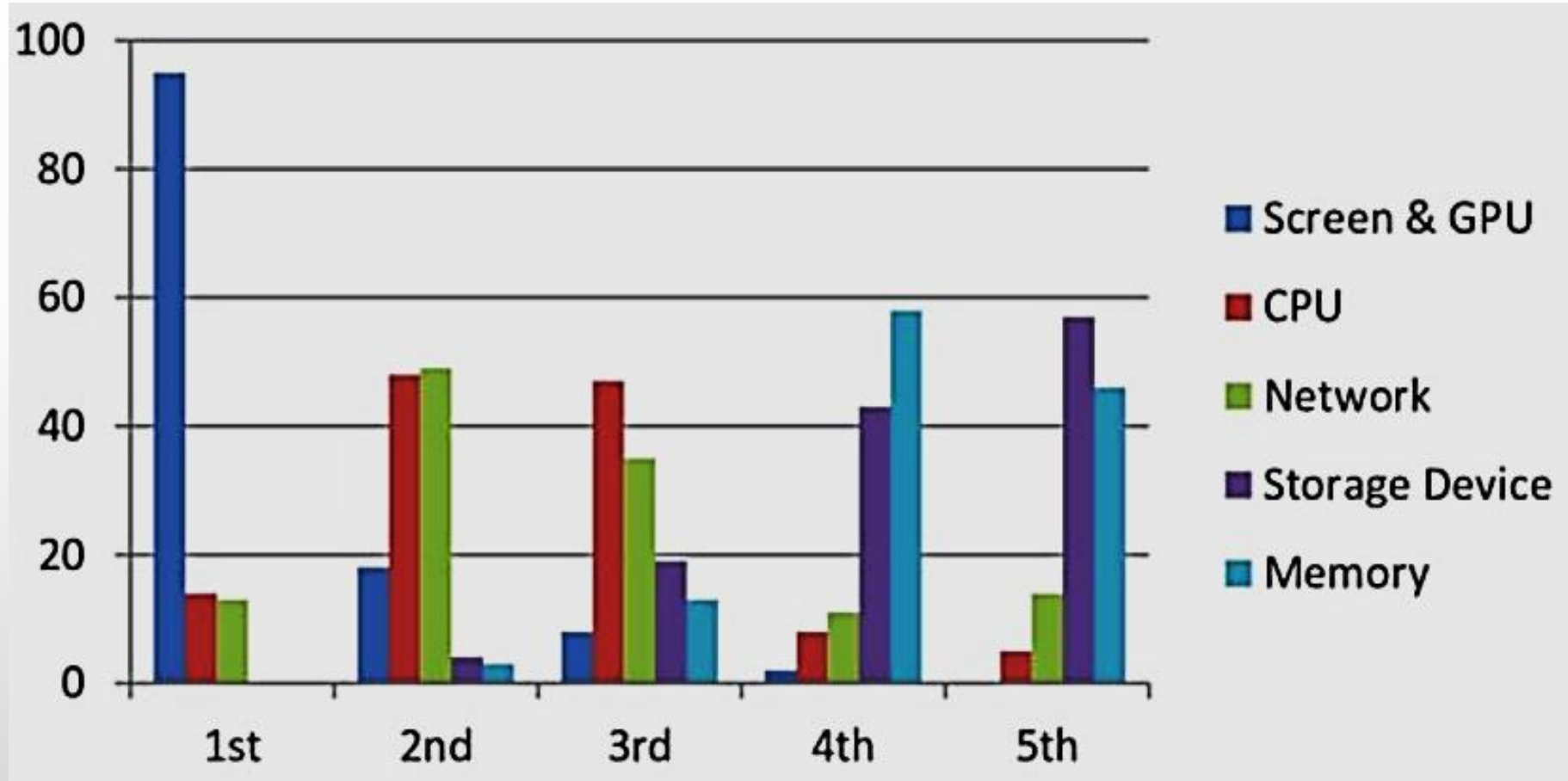


Masaüstü Aygıtların Güç Tüketimi





Dizüstü Aygıtların Güç Tüketimi





SON