

# Capitolo 5

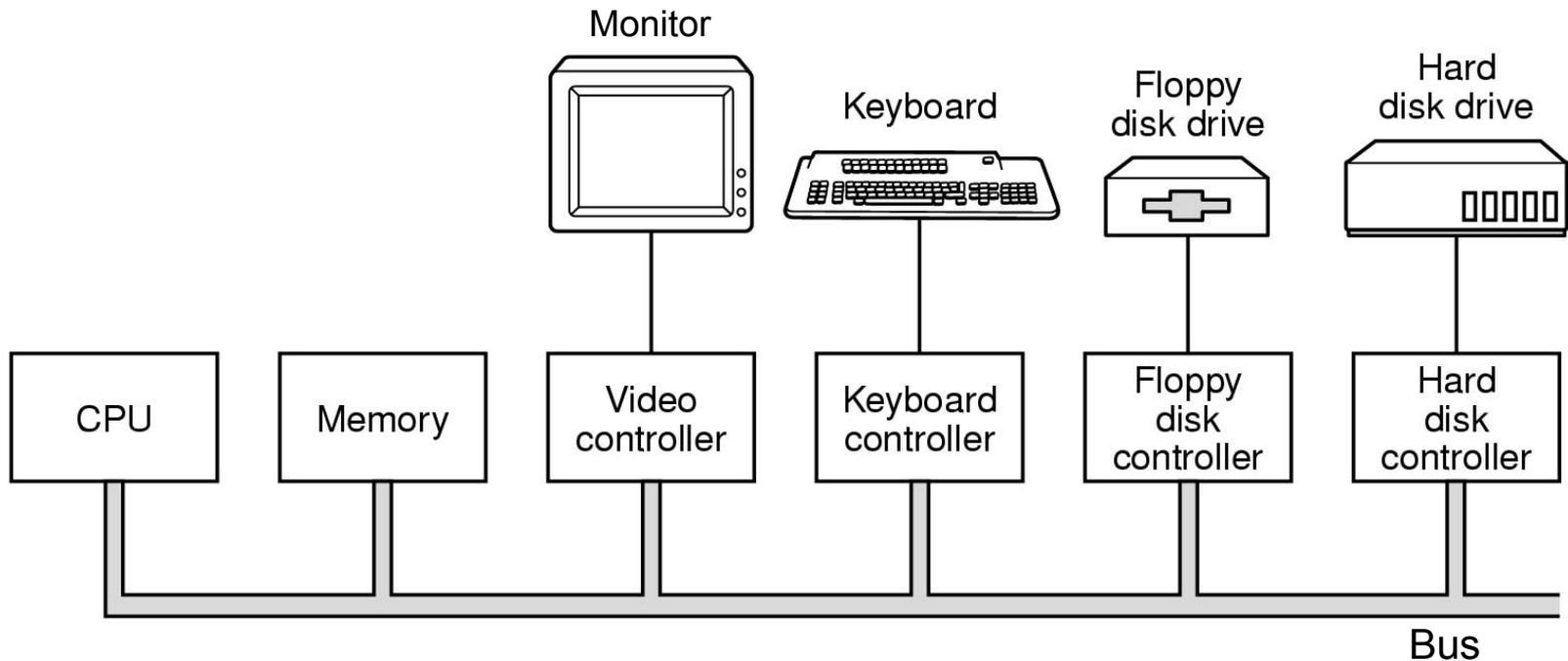
## Input/Output

- Principi di hardware di I/O
- Principi di software di I/O
- Livelli di software di I/O
- Dischi
- Video

# Controllori dei Dispositivi (1)

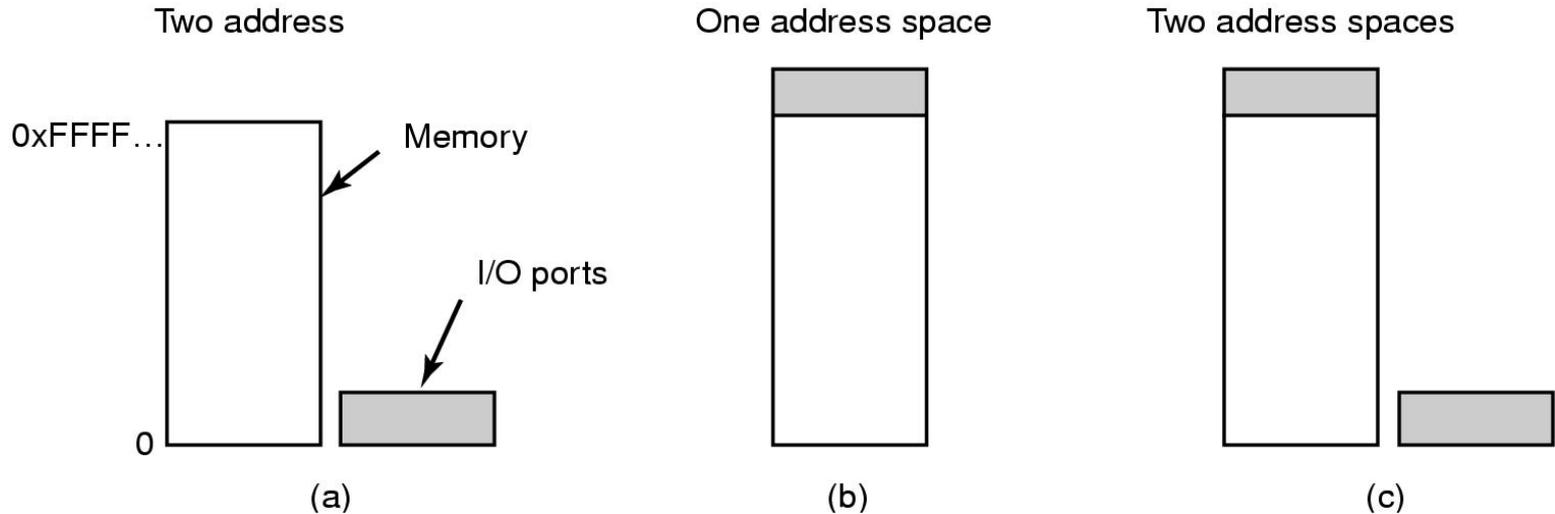
- Componenti di un dispositivo di I/O:
  - componenti meccaniche
  - componenti elettroniche
- La componente elettronica è il controllore del dispositivo (*device controller*)
  - può controllare più dispositivi alla volta
- Compiti del controllore
  - convertire un flusso (*stream*) seriale di bit in blocchi di byte
  - effettuare la correzione degli errori se necessario
  - rendere i byte disponibili per essere copiati in RAM

# Controllori dei diapositivi (2)



Componenti di un semplice PC

# Memory-Mapped I/O (1)

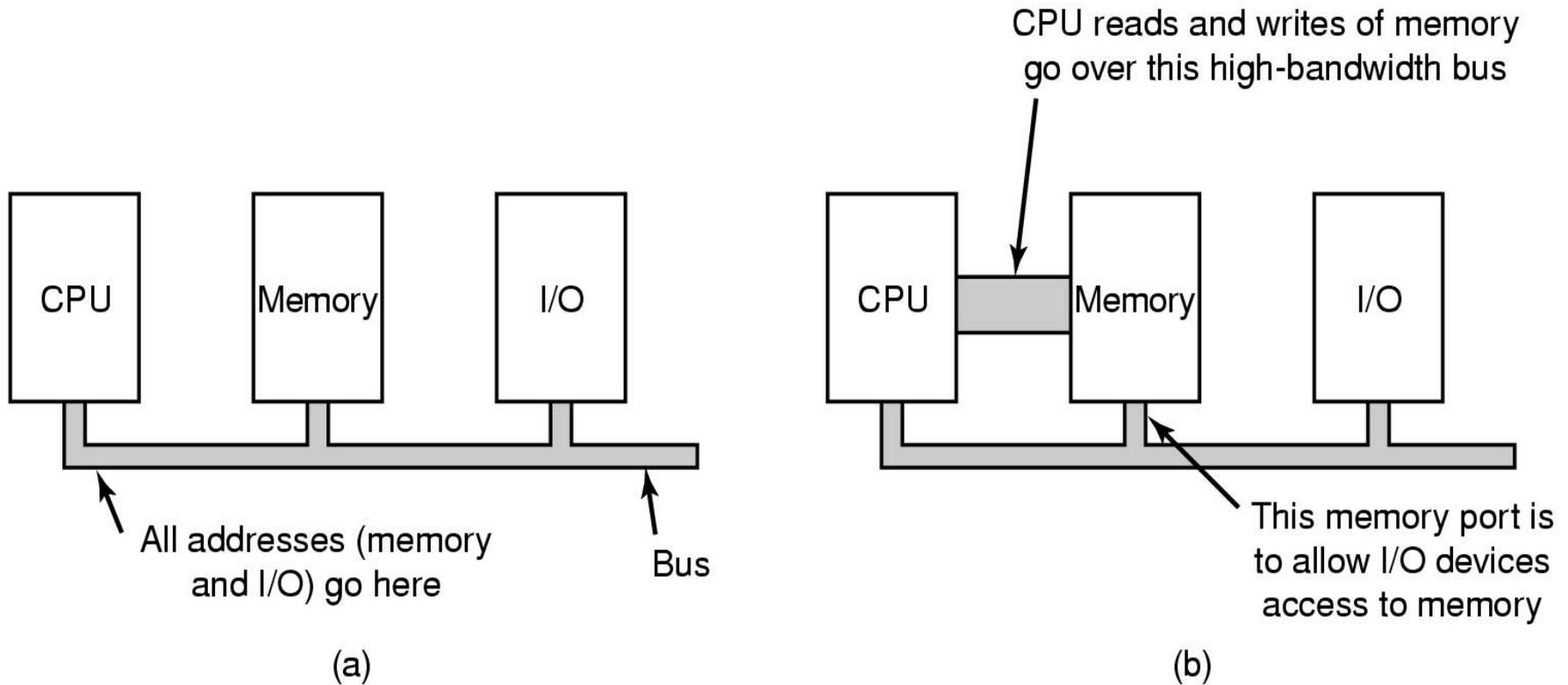


(a) Spazi di memoria ed I/O separati

(b) Memory-mapped I/O

(c) Modello ibrido

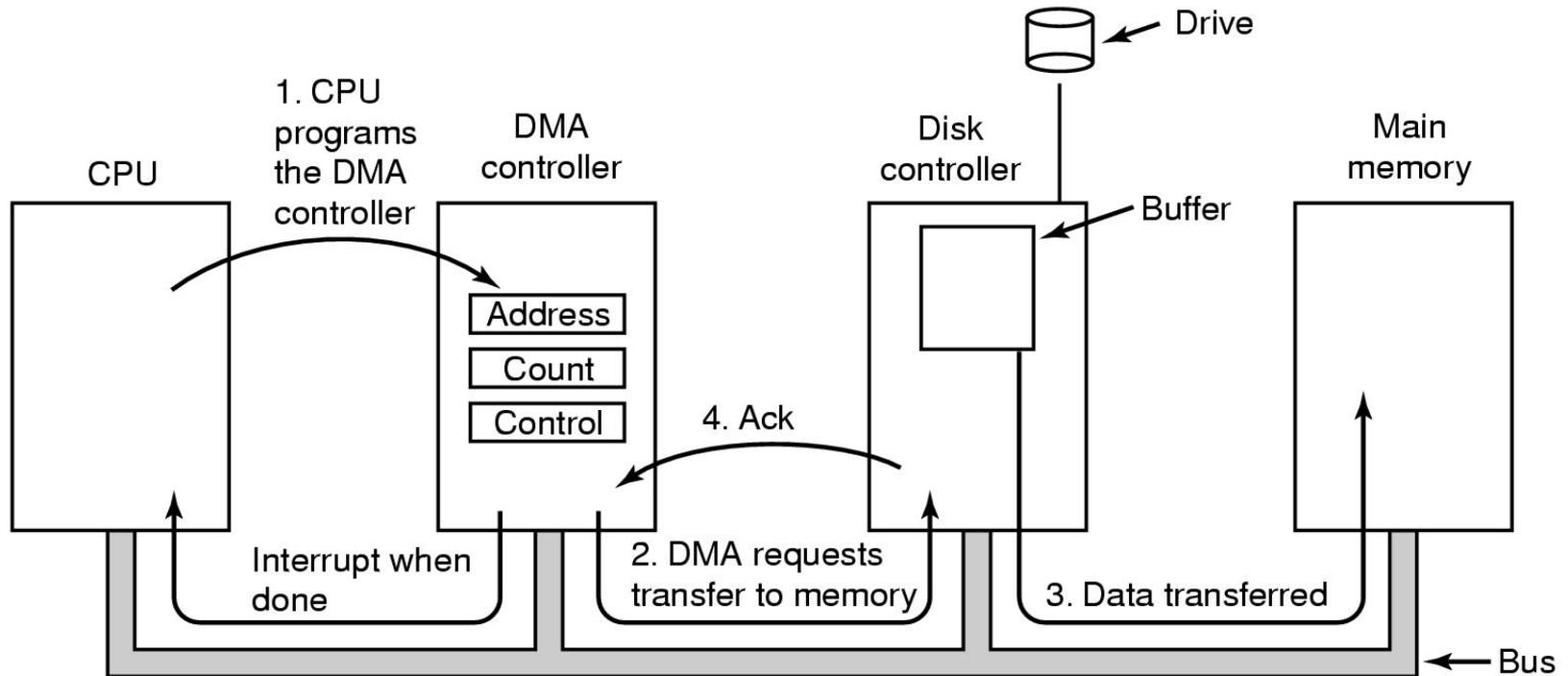
# Memory-Mapped I/O (2)



(a) Architettura con singolo bus

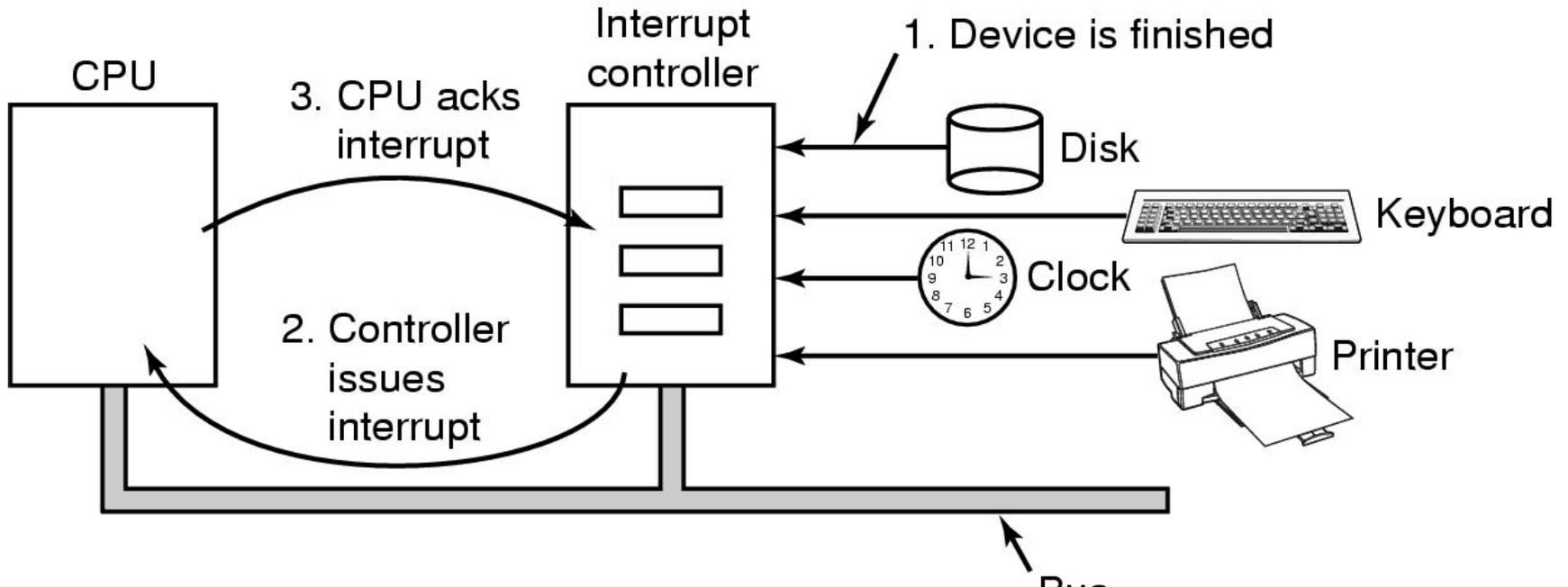
(b) Architettura con due bus (un bus è dedicato ai collegamenti veloci processore-memoria)

# Direct Memory Access (DMA)



Operazioni durante un trasferimento DMA

# Interruzioni rivisitate



Come avvengono le interruzioni. Le connessioni fra dispositivi ed *interrupt controller* in realtà utilizzano le linee di interrupt del bus e non dei collegamenti dedicati.

# Principi di Software di I/O

## Scopi del software di I/O (1)

- **Indipendenza dal dispositivo**
  - i programmi dovrebbero poter accedere alle informazioni memorizzate sui diversi dispositivi senza dover essere modificati o ricompilati se il dispositivo varia
    - (floppy, disco rigido, oppure CD-ROM)
- **Denominazione uniforme (*uniform naming*)**
  - il nome di un file o di un dispositivo dovrebbe essere una stringa o un intero e
  - non dipendere dal tipo di dispositivo

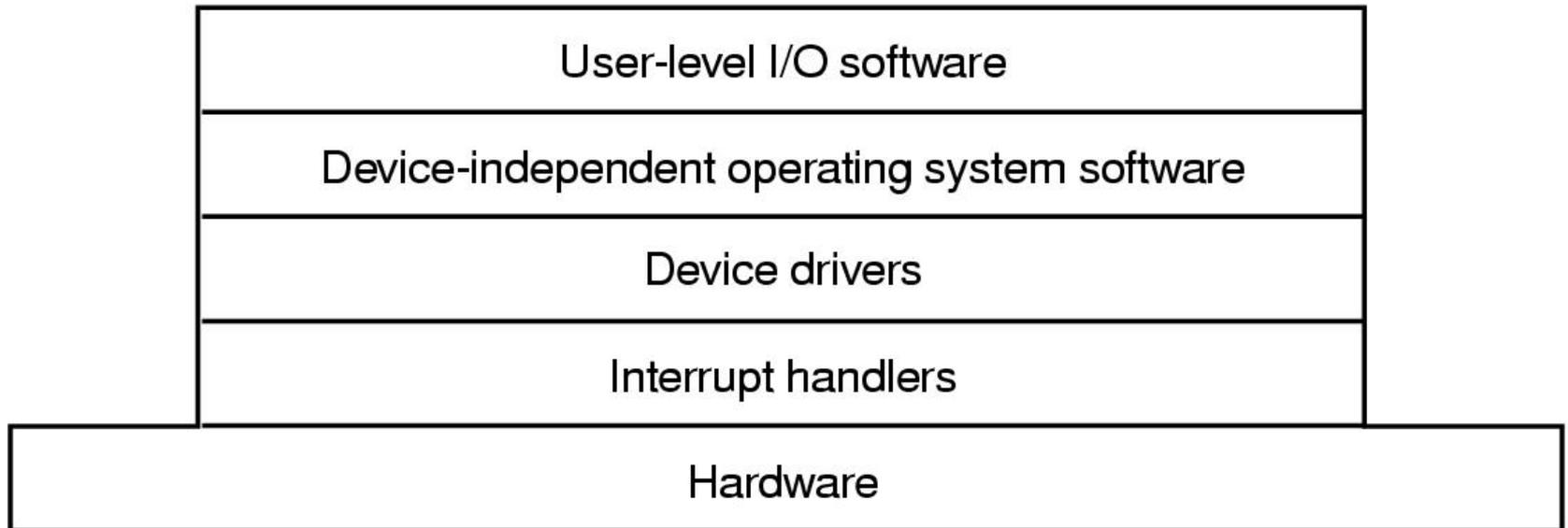
# Scopi del Software di I/O (2)

- Gestione degli errori
  - dovrebbero essere trattati il più possibile “vicino” all’hw che li ha causati
- Trasferimenti sincroni vs. asincroni
  - i programmi vedono un I/O sincrono (il processo si blocca)
  - il sistema operativo gestisce i trasferimenti asincroni (con il meccanismo delle interruzioni)

# Scopi del Software di I/O (3)

- Memorizzazione temporanea (*buffering*)
  - dati che arrivano da un dispositivo ma non possono essere ancora memorizzati nella loro destinazione finale
- Risorse condivisibili e non
  - i dischi possono essere condivisi da più utenti
  - i lettori di nastri no!
  - È possibile che si verifichi stallo.

# Livelli di Software di I/O



Livelli di software utilizzati dal sottosistema di I/O

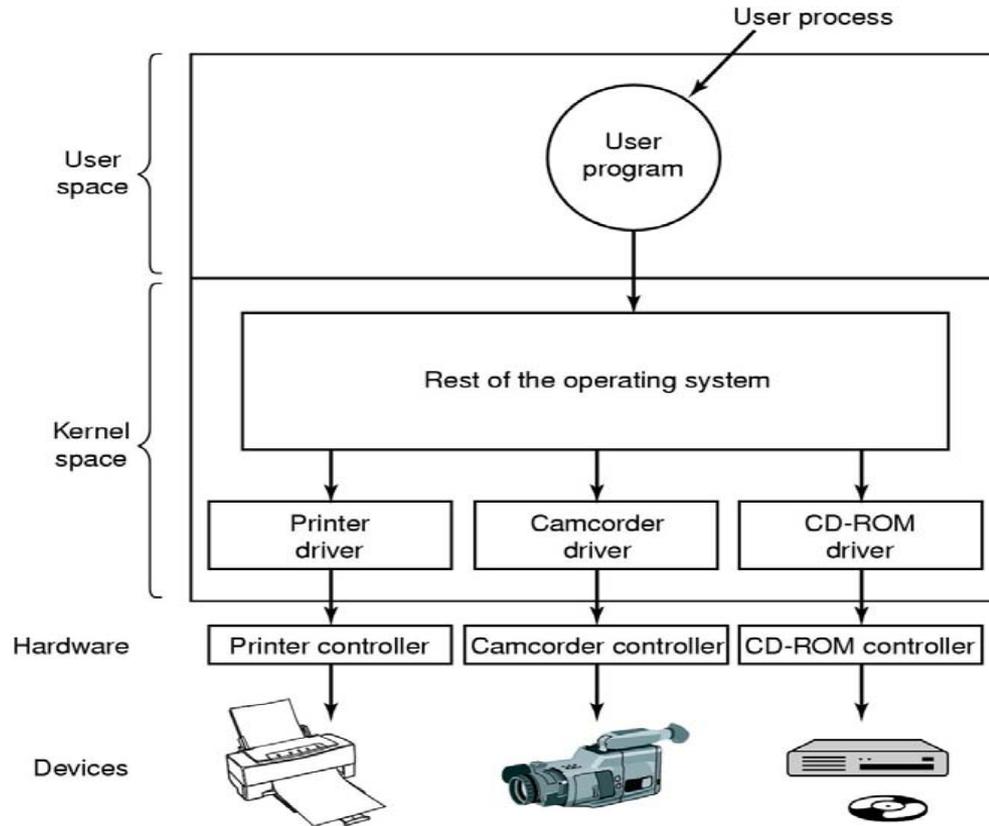
# Gestori delle Interruzioni (*Interrupt Handlers*) (1)

- Meglio nascondere i gestori delle interruzioni
  - il driver si blocca dopo aver iniziato una operazione di I/O
- La procedura di gestione dell'interrupt
  - sblocca il driver quando arriva dall'interrupt che notifica il completamento dell'operazione
- Passi che devono essere eseguiti dal software dopo aver ricevuto un interrupt
  1. Salvataggio dei registri non ancora salvati dall'hardware
  2. Preparazione del contesto di esecuzione (MMU) per la procedura di gestione dell'interrupt

# Gestori delle Interruzioni (2)

3. Preparazione dello stack per la procedura di gestione
4. Invio di un *ack* al controllore delle interruzioni, riabilitazione delle interruzioni
5. Copia dei registri da dove sono stati salvati nella tabella dei processi
6. Esecuzione della procedura di gestione
7. Scheduling
8. Preparazione del contesto di MMU per il prossimo processo da mandare in esecuzione
9. Caricamento dei registri per il nuovo processo
10. Inizio dell'esecuzione del nuovo processo

# Driver dei Dispositivi (*Device Drivers*) (1)



- Posizione logica dei driver dei dispositivi
- La comunicazione fra driver e controller avviene attraverso il bus

# Driver dei Dispositivi (*Device Drivers*) (2)

Tipico funzionamento di un driver :

1. Inizializza il dispositivo
2. Accetta richieste di operazioni e ne controlla la correttezza
3. Gestisce le code delle richieste che non possono essere subito servite
4. Sceglie la prossima richiesta da servire e la traduce in una sequenza  $S$  di comandi a basso livello da inviare al controllore
5. Trasmette i comandi in  $S$  al controllore eventualmente bloccandosi in attesa del completamento dell'esecuzione di un comando
6. Controlla l'esito di ciascun comando gestendo eventuali errori
7. Invia l'esito dell'operazione ed eventuali dati al richiedente

# Software di I/O Indipendente dal dispositivo (1)

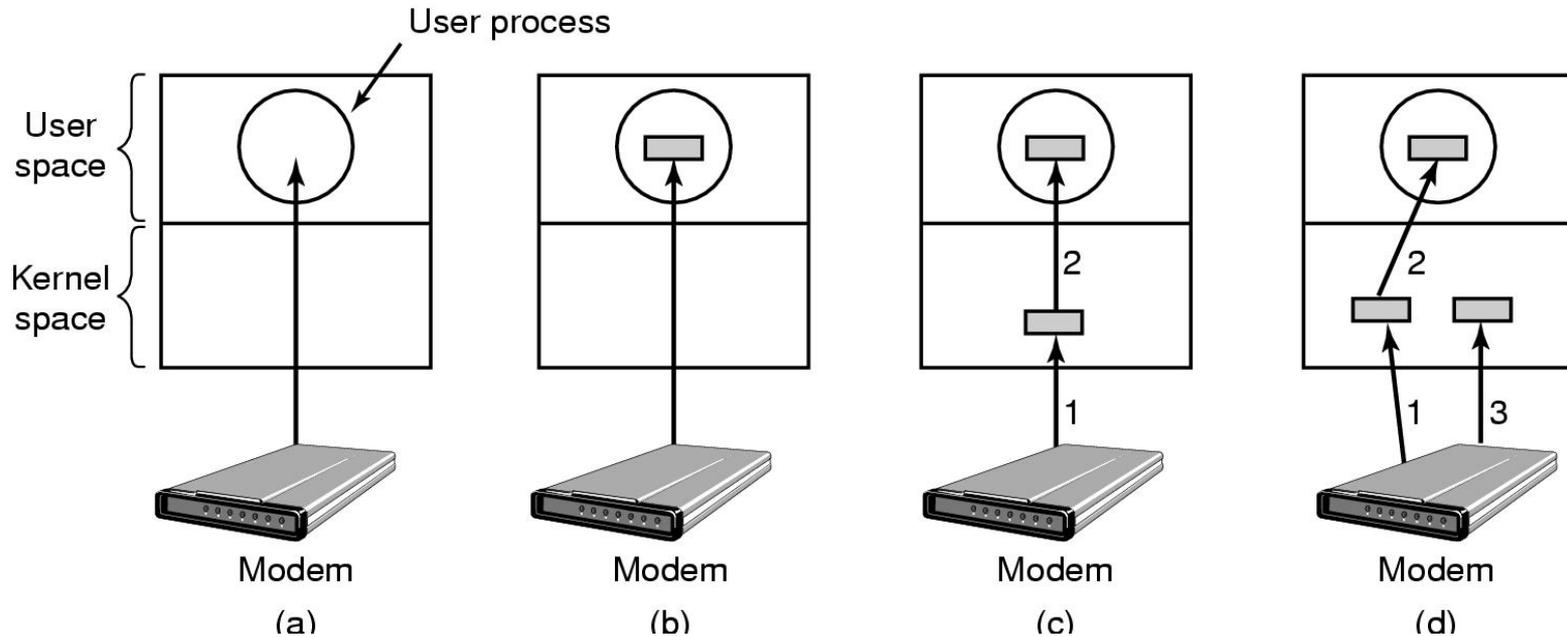
|   |
|---|
| Interfaccia uniforme per i driver       |
| Bufferizzazione                         |
| Segnalazione degli Errori               |
| Allocazione e rilascio delle risorse    |
| Block size indipendente dal dispositivo |

Funzioni del software di I/O indipendente dal dispositivo

## Software di I/O Indipendente dal dispositivo (2)

- Le richieste ai driver vengono invocate usando una interfaccia uniforme per tutti i driver della stessa classe
  - driver di dispositivi a blocchi, a caratteri
- I driver possono richiedere operazioni al kernel attraverso un insieme di funzioni uniforme fissato
  - allocazione di aree di memoria fisica contigua per i buffer
  - interazione con il controllore DMA, la MMU

# Software di I/O Indipendente dal dispositivo (3)



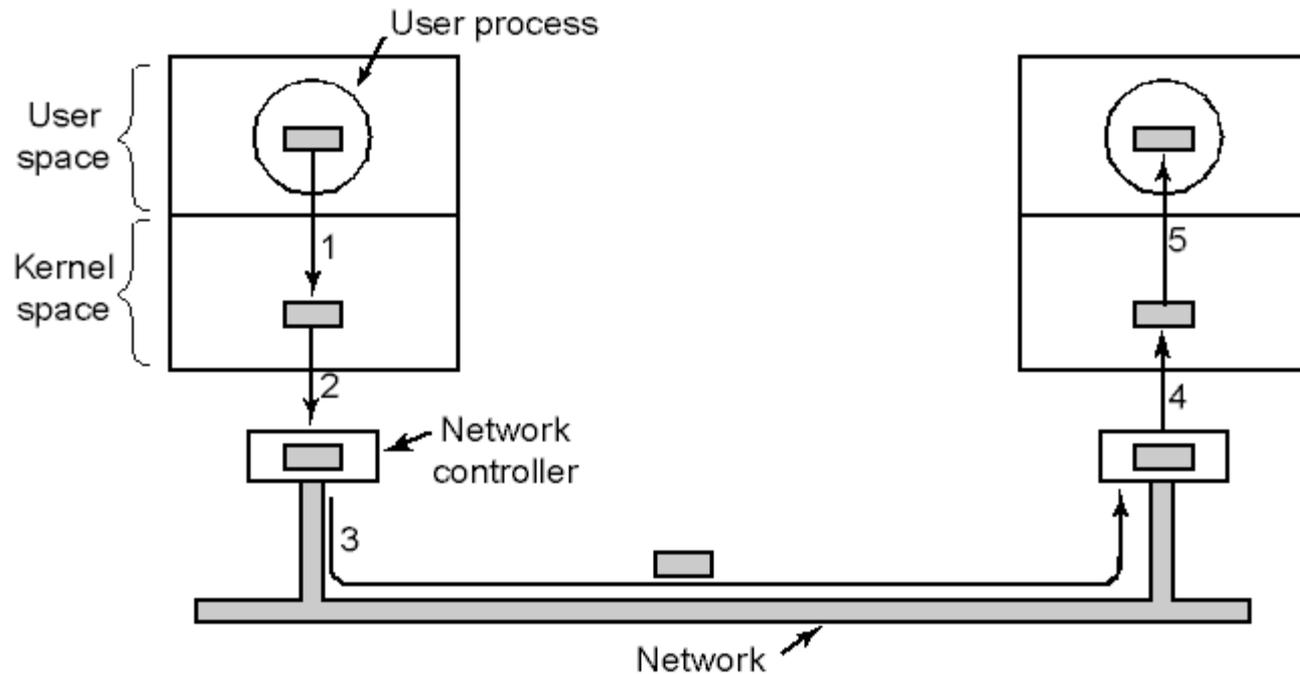
(a) Input non bufferizzato

(b) Input bufferizzato in spazio utente

(c) Input bufferizzato nel kernel, seguito da una copia in spazio utente

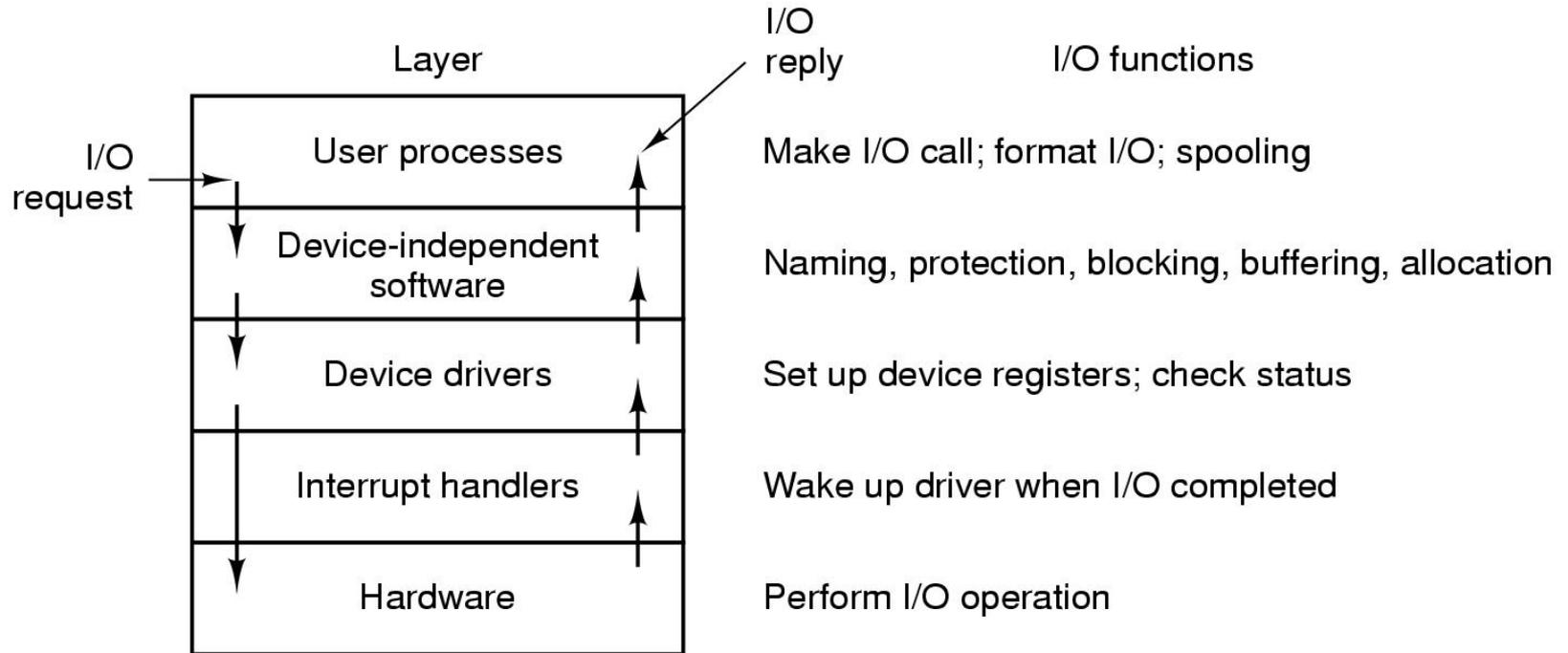
(d) Doppia bufferizzazione nel kernel

# Software di I/O Indipendente dal dispositivo (4)



La trasmissione attraverso la rete può comportare molte copie

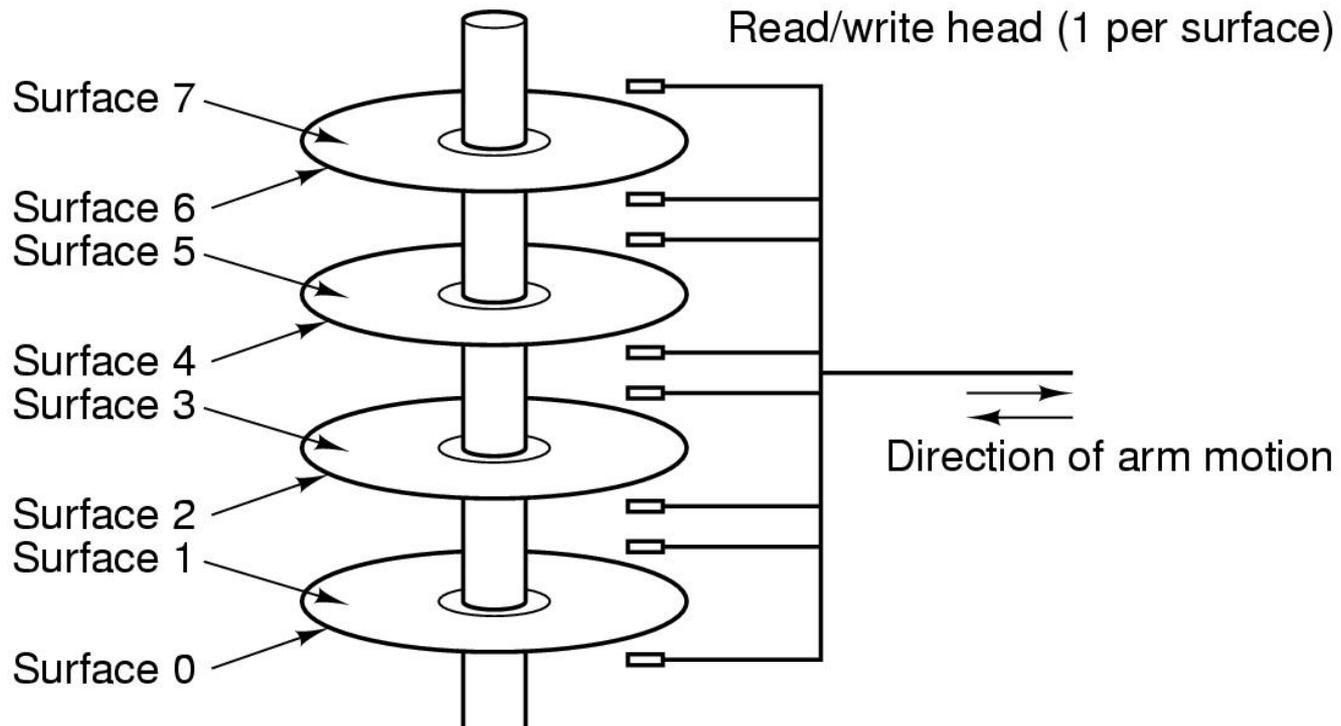
# Software di I/O in spazio utente



Livelli del sottosistema di I/O e funzioni principali di ciascun livello

# Dischi

## Hardware del disco (1)



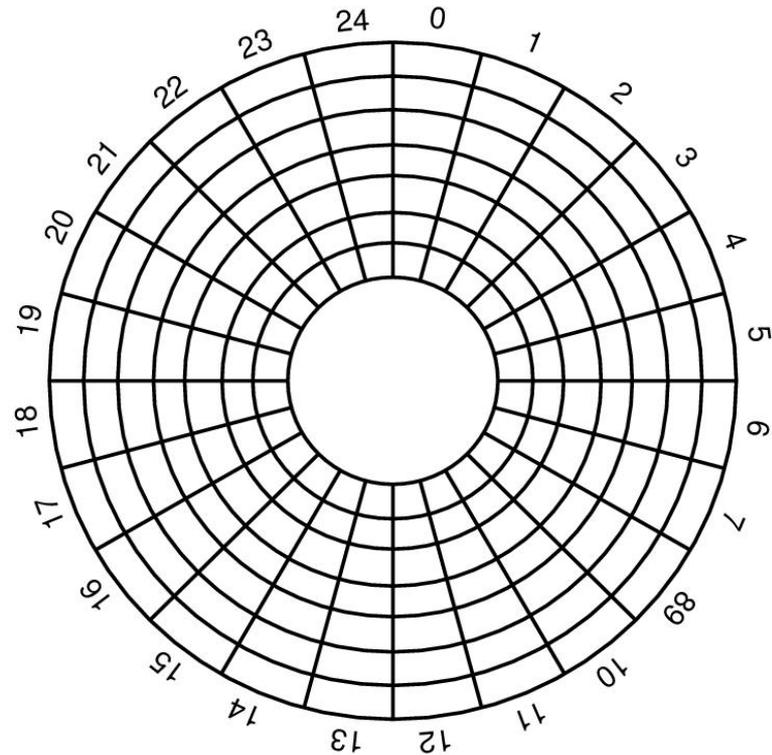
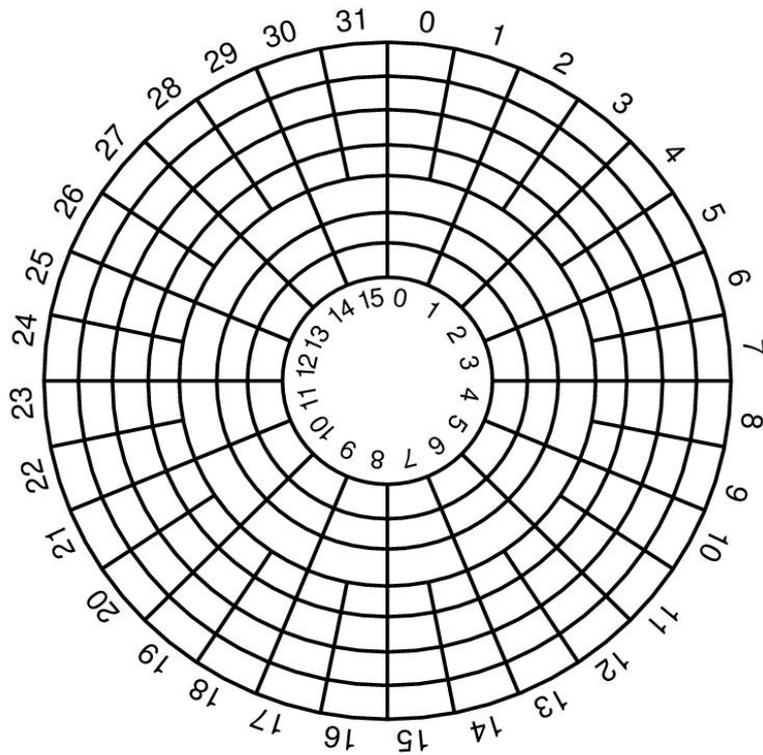
Struttura di un disco rigido

# Hardware del disco (2)

| <b>Parameter</b>               | <b>IBM 360-KB floppy disk</b> | <b>WD 18300 hard disk</b> |
|--------------------------------|-------------------------------|---------------------------|
| 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 $\mu$ sec              |

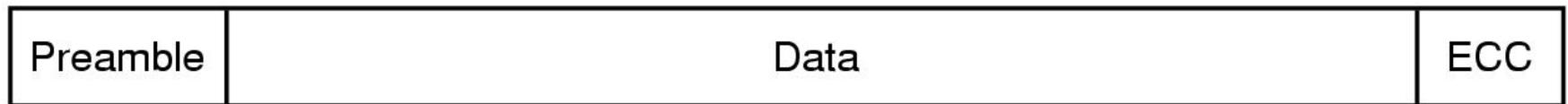
Parametri del floppy disk del PC originale di IBM e di un disco rigido Western Digital WD 18300

# Hardware del disco (3)



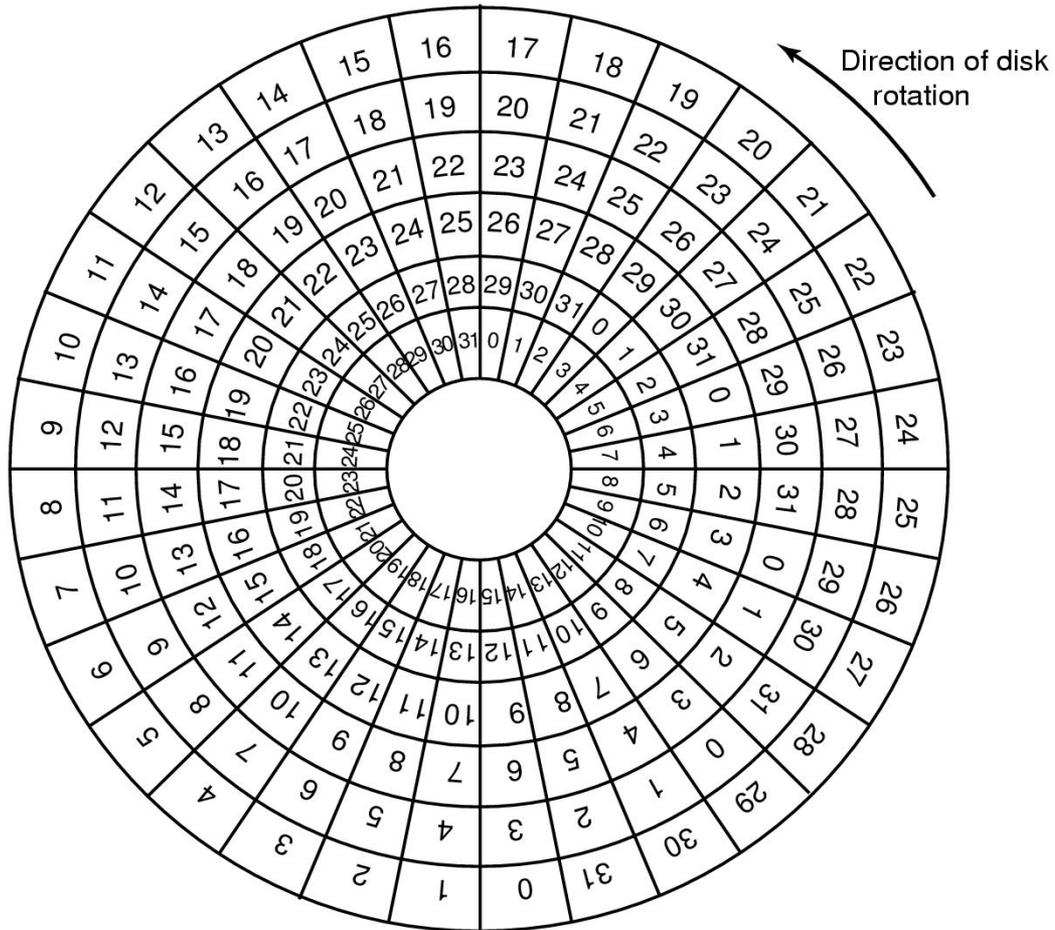
- Geometria fisica di un disco con due zone
- Una possibile geometria virtuale per lo stesso disco

# Formattazione del disco (1)



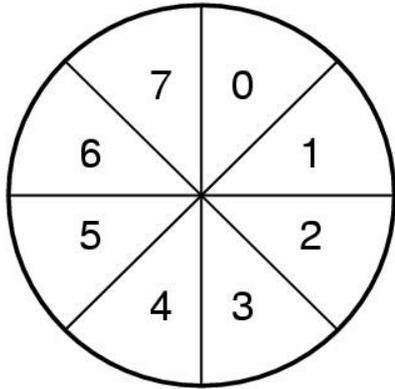
Un settore del disco

# Formattazione del disco (2)

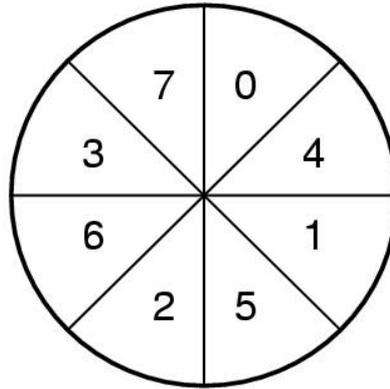


Una illustrazione del *cylinder skew*

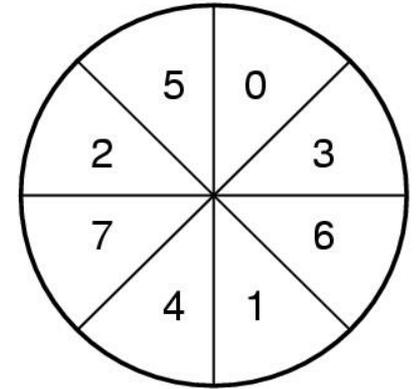
# Formattazione del disco (3)



(a)



(b)



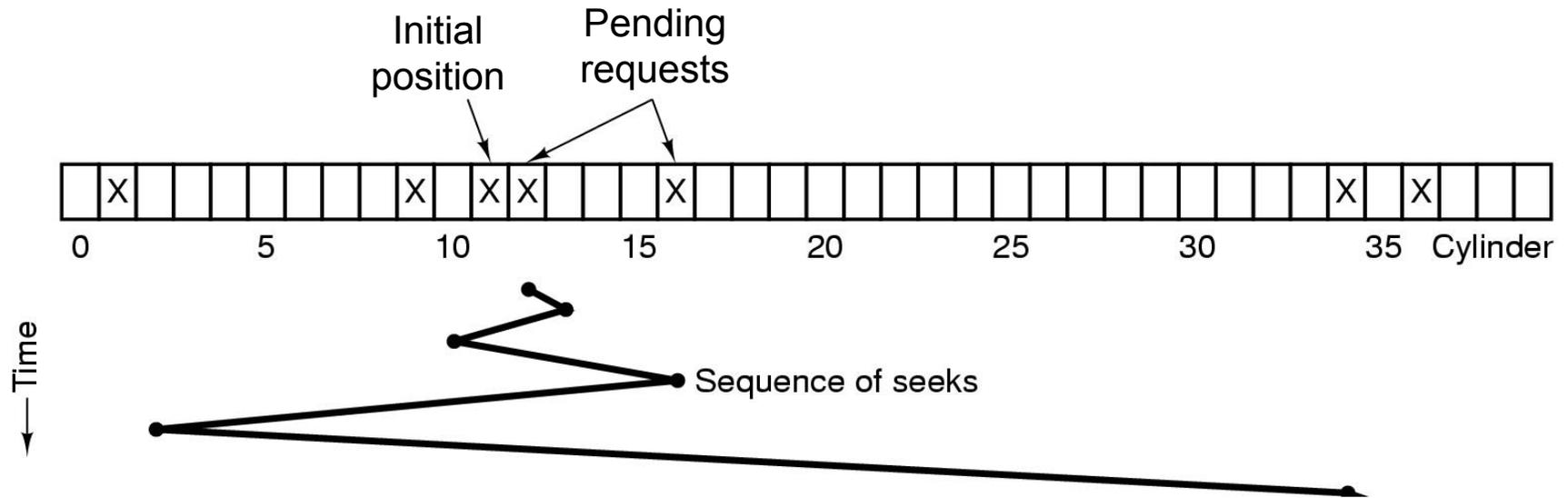
(c)

- Senza interleaving
- Con interleaving singolo
- Con interleaving doppio

# Algoritmi di scheduling per il braccio (1)

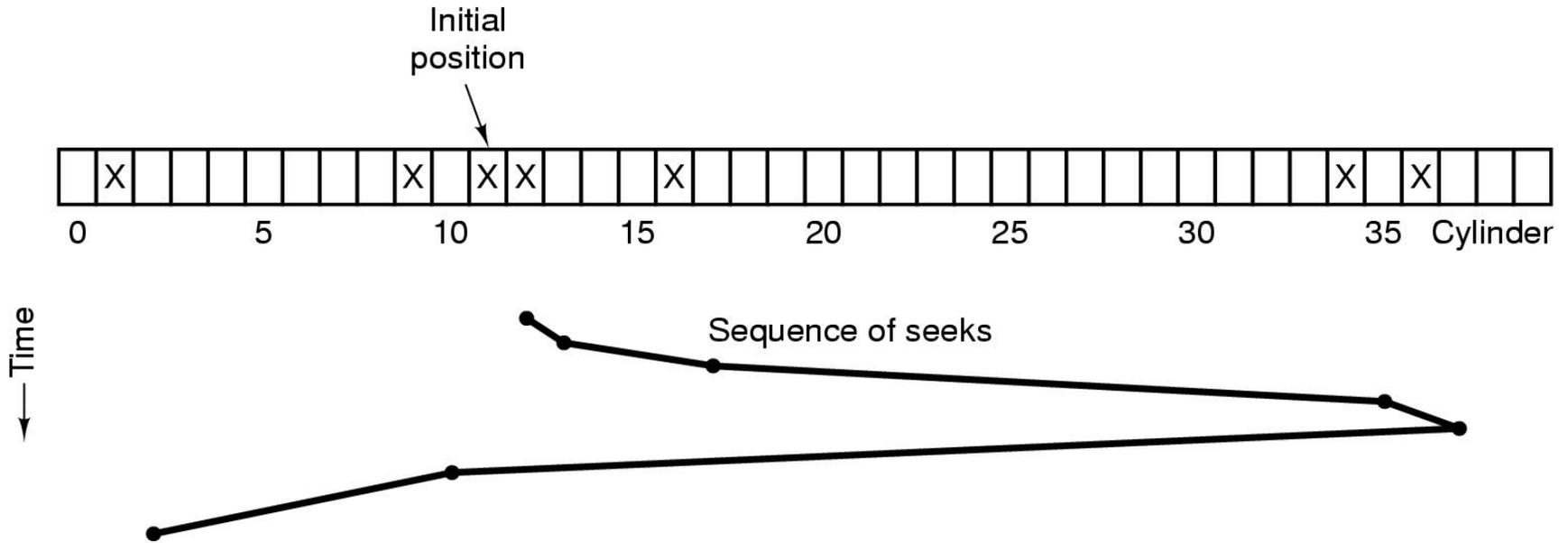
- Il tempo necessario per leggere o scrivere un blocco è determinato da tre fattori
  1. Tempo di *seek*
  2. Ritardo rotazionale (Rotational delay)
  3. Tempo di trasferimento vero e proprio
- Il tempo di seek domina
- Il controllo degli errori viene fatto dal controller

# Algoritmi di scheduling per il braccio (2)



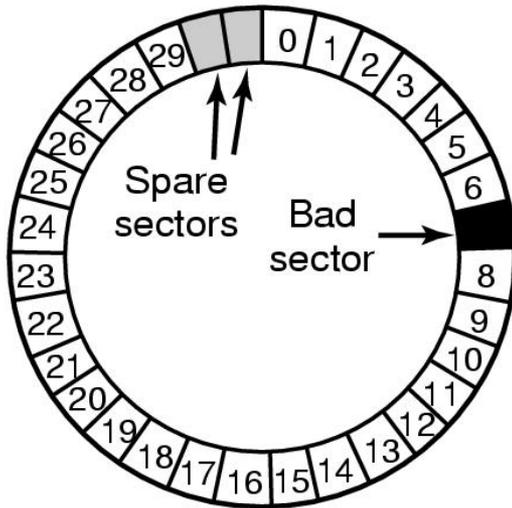
L'algoritmo di scheduling *Shortest Seek First* (SSF)

# Algoritmi di scheduling per il braccio (3)

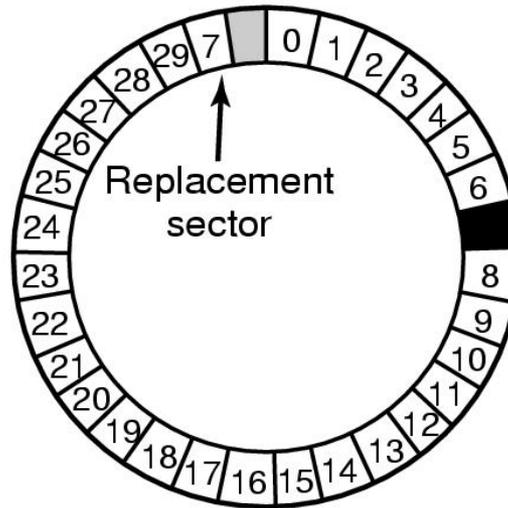


L'algoritmo di scheduling "dell'ascensore"

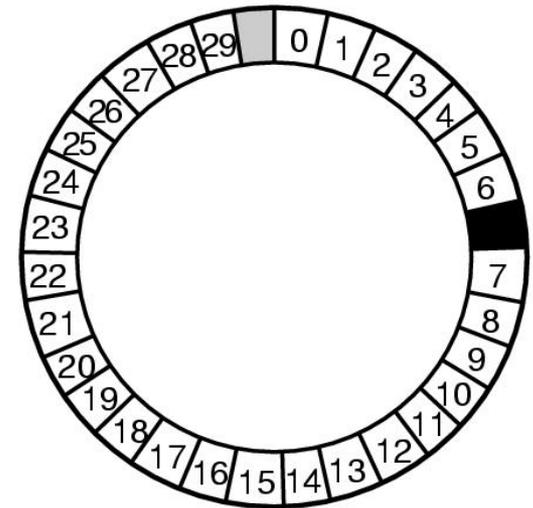
# Gestione degli errori



(a)



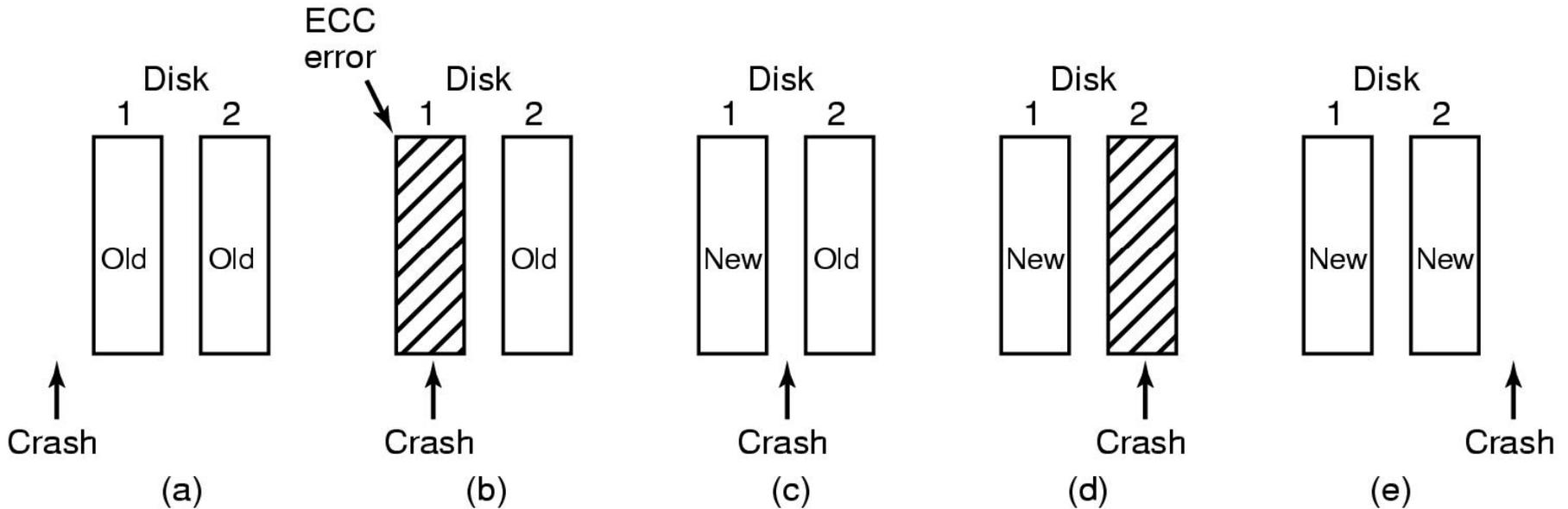
(b)



(c)

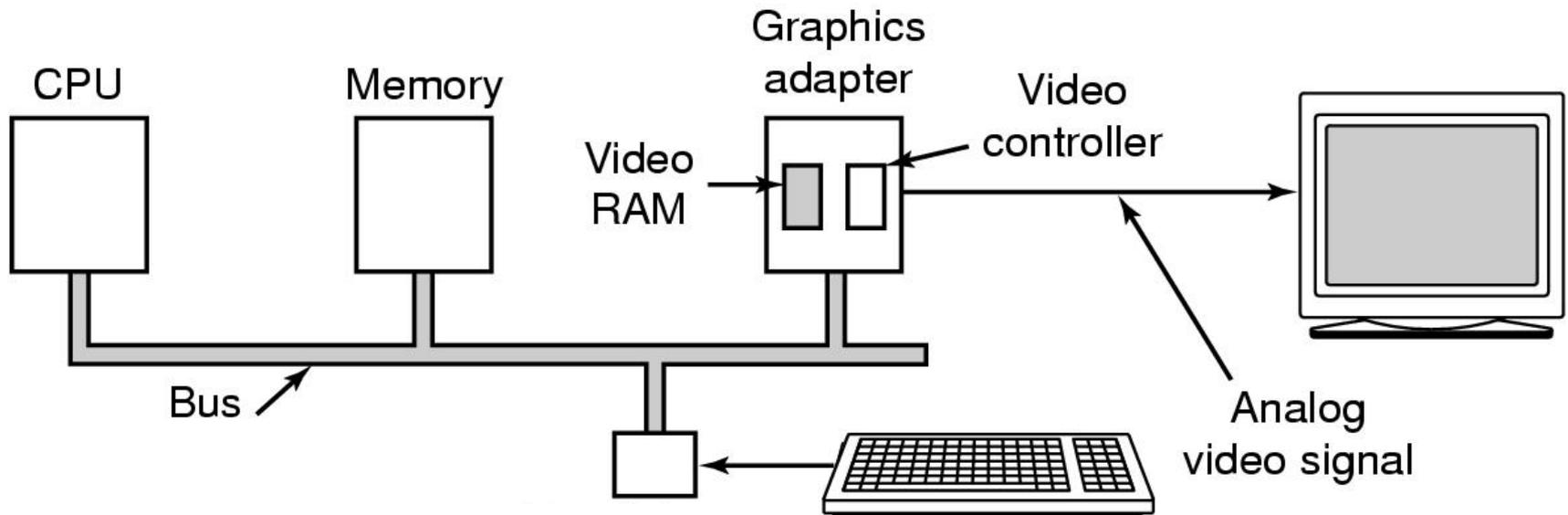
- Una traccia con un settore difettoso
- Sostituzione del settore difettoso con un settore di riserva
- Slittamento dei settori per evitare quello difettoso

# Stable Storage



Analisi dell'influenza di un crash sulle write stabili

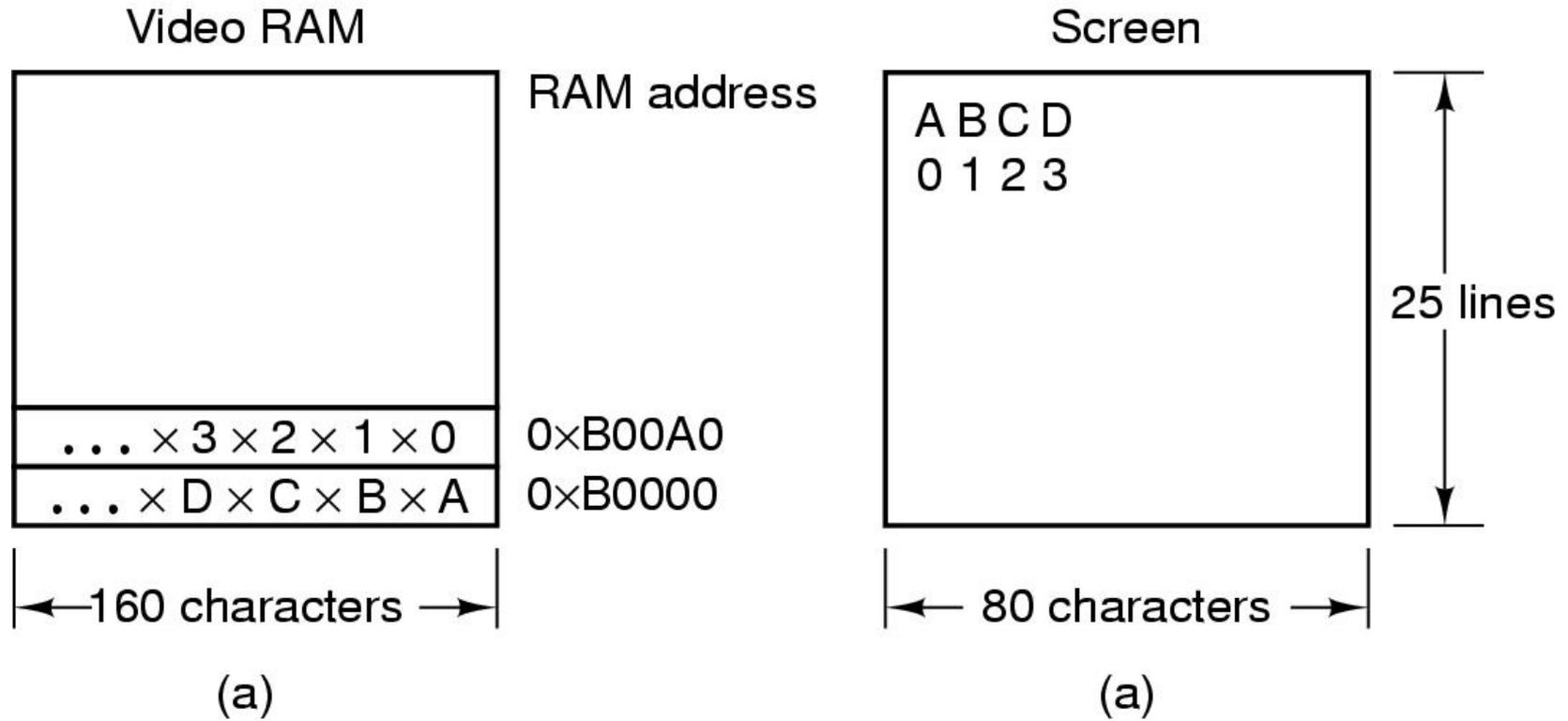
# Hardware del Video (1)



Video mappato in memoria (*memory-mapped*)

- il driver scrive direttamente nella RAM del video

# Hardware del Video (2)



- Una immagine della video RAM
  - un semplice display monocromatico, modo carattere
- La schermata corrispondente
  - gli X sono i byte degli attributi