L'input/output – Parte VI

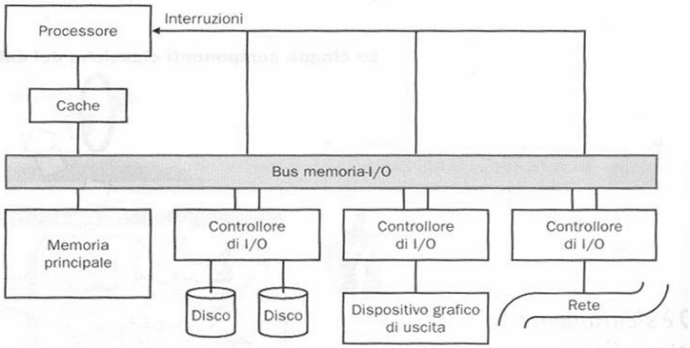
Architettura
degli Elaboratori e delle Reti

Alberto Borghese
Università degli Studi di Milano
Dipartimento di Scienze dell'Informazione
email: borghese@dsi.unimi.it

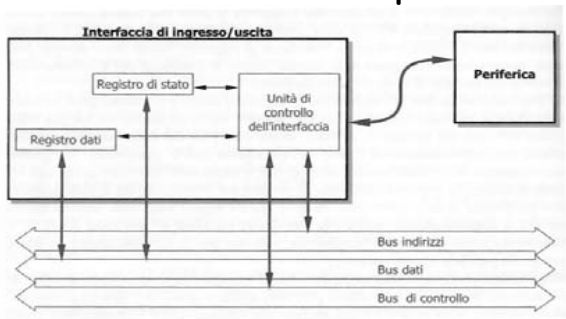
I/O

- Indirizzamento delle periferiche
- Gestione I/O a controllo di programma
- Gestione I/O ad interruzione
- Gestione I/O ad accesso diretto alla memoria (DMA)
- Tipi di bus
- Controllo accesso al bus (arbitraggio)
- I dischi

Architettura di un elaboratore

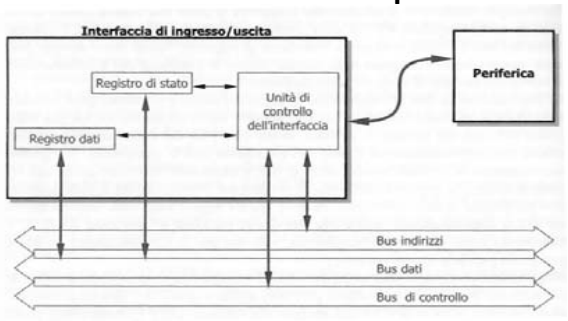


Interfacciamento di una periferica



Parallela (centronics). 1 byte alla volta.
Seriale (RS232). 1 bit alla volta. Nuovi standard sono USB e Firewire (IEEE 1394) e Bluetooth (wireless).

Interfacciamento di una periferica



Registri:
•Dati
•Stato: situazione della periferica (idle, busy, down….) e comando in esecuzione.

Indirizzamento dell'I/O

• Memory-mapped
• Istruzioni speciali di I/O

Istruzioni speciali di I/O

Istruzioni appartenente alla ISA che indirizzano direttamente il dispositivo:
- Numero del dispositivo
- Parola di comando (o indirizzo della parola di comando)

Indirizzamento memory-mapped

•I registri del device controller sono considerati come celle di memoria RAM.
•I loro indirizzi saranno diversi da quelli delle celle di memoria.
•Il processore esegue operazioni di I/O come se fossero operazioni di lettura/scrittura in memoria.

Esempio:

```
sw $s0, indirizzo
lw  $s0, indirizzo
```

dove l'indirizzo è al di fuori dallo spazio fisico della memoria.

Indirizzamento memory-mapped

• I controller ascoltano tutti i segnali in transito sul bus (*bus snooping*) e si attivano solamente quando riconoscono sul bus indirizzi, l'indirizzo corrispondente alla propria locazione di memoria.

• Gli indirizzi riservati ai registri del controller fanno di solito riferimento alla porzione di memoria riservata al SO e non accessibile quindi al programma utente.

• I programmi utente devono quindi passare dal SO per accedere a questi indirizzi riservati (**modalità kernel**) e quindi effettuare operazioni di I/O. Questo è quanto viene fatto ricorrendo alle System Call.

Gestione dell'I/O

- A controllo di programma diretto
- A controllo di programma con polling
- Ad interruzione
- Ad accesso diretto alla memoria (DMA)

I/O a controllo di programma

e.g. chiamata syscall per la stampa di una stringa.

La periferica ha un ruolo passivo. Il processore esegue tutto il lavoro.

Svantaggio: La CPU dopo avere predisposto il controller all'esecuzione dell'I/O si ferma e si mette ad interrogare il registro di stato della periferica in attesa che il ready bit assuma un determinato valore. Stato *busy waiting* o *spin lock*.

```
begin
        1. Predisponi i registri del controller ad effettuare una
                operazione di lettura.
        2. While (ready-bit = = 0) do;          // spin lock
        3. Carica il dato acquisito;
end;
```


Esempio: Receiver

```
.text
.globl main
main:
        li$t0, 0xffff0000       # indirizzo del receiver control register
        li $t2, 0xffff0004       # indirizzo del receiver data register

# Ciclo di lettura di un carattere
ciclo:  lw $t1, 0($t0)          # Contenuto del registro di controllo
        rem $t1, $t1, 2         # If $t1 = = 1 (resto = 1) esci
        beqz $t1, ciclo
        lw $a0, 0($t2)

        li $v0, 10
        syscall
```


I/O a gestione di programma - costo

Ipotesi:
1) Tastiera gestita a controllo di programma che opera a 0,01Kbyte/s.
2) Frequenza di clock: 50Mhz.

Determinare il tempo in cui verrebbe effettivamente utilizzata la CPU per trasferire 1 parola, tenendo conto che ci vogliono 20 cicli di clock per ogni byte.

T = 4 / 0,01 Kbyte /s = 0,4s #cicli_clock = $50*10^6 * 0,4 = 20,0 * 10^6$

Invece ne utilizza solo 20 * 4 = 80 per trasferire i dati.

%sfruttamento della CPU è: $(80/20,0 * 10^6) *100 = 0,0004\%$

I/O a gestione di programma - costo

Ipotesi:
1) Floppy che opera a 50Kbyte/s.
2) Hard-disk che lavora a 2MByte /s.
3) Frequenza di clock: 50Mhz.

Determinare la percentuale di tempo in cui verrebbe effettivamente utilizzata la CPU per trasferire 1 parola, tenendo conto che ci vogliono 20 cicli di clock per ogni byte.

Floppy-disk:
4 byte / 50kbyte/s = 8ms =>$50x10^6$ x $8x10^{-3}$ = 400 x 10^3 cicli_clock => 0,02%

Hard-disk:
4 byte / 2Mbyte /s = 2μs =>$50x10^6$ x $2x10^{-6}$ = 100 cicli_clock => 80%

Gestione dell'I/O

- A controllo di programma diretto
- A controllo di programma con polling
- Ad interruzione
- Ad accesso diretto alla memoria (DMA)

Polling

•Interrogazione del registro di stato della periferica.
•Ciclo di polling: durante un ciclo di busy-waiting su un dispositivo si esegue il polling sugli altri dispositivi di I/O.
•Quando una periferica necessita di un qualche intervento, si soddisfa la richiesta e si prosegue il ciclo di polling sugli altri I/O.

```
// Leggi dato da perif_x
begin
a. Predisponi i registri dei controller ad eseguire una read;
b. if(ready_bit (perif_1) = = 1) servi perif_1;
   if (ready_bit (perif_2) = = 1) servi perif_2;
        ….
   if (ready_bit (perif_n) = = 1) servi perif_n;
   goto b;
end;
```


I/O a gestione di programma – costo polling

Ipotesi:
1) Costo del polling (# cicli di clock per un'operazione di polling, costituita da trasferimento del controllo alla procedura di polling, accesso al dispositivo di I/O, e ritorno al programma utente): 400.
2) Frequenza di clock: 500Mhz.

Determinare l'impatto del polling per 3 dispositivi diversi:
A) Mouse. Deve essere interrogato almeno 30 volte al secondo per non perdere alcun movimento dell'utente.
B) Floppy disk. Trasferisce dati al processore in parole da 16 bit ad una velocità di 50 Kbyte/s.
C) Hard disk. Trasferisce dati al processore in blocchi di 4 parole e può trasferire 4 Mbyte/s.

I/O a gestione di programma - costo

Mouse:

30 x 400 = 12,000 cicli_clock/s
12,000 / 500,000,000 = 0,000024s => 0,0024%
Piccolo impatto sulle prestazioni.

Floppy disk:

Per ogni accesso possiamo trasferire 2byte.
Occorrono quindi 25k accessi/s.
In termini di cicli di clock: 25k x 400 = 10Mcicli_clock/s
(ignorando discrepanza tra 25k in base 2 ed in base 10) => 2%
Medio impatto sulle prestazioni.

Hard disk:

Per ogni accesso possiamo trasferire 16byte.
Occorrono quindi 250k accessi/s
In termini di cicli di clock: 250k x 400 = 100Mcicli_clock/s
(ignorando discrepanza tra 250k in base 2 ed in base 10) => 20%
Alto impatto sulle prestazioni.

I/O a controllo di programma

Periferiche pilotate dall'esterno (dall'utente, e.g. mouse)
Periferiche pilotate dal SO (dischi).

Nei dischi, si potrebbe attivare il polling quando ne è richiesto l'utilizzo.
Posizionamento delle testine (=> spin lock)

•I miglioramenti del polling rispetto al controllo di programma sono molto limitati.
•I problemi principali del polling (e dell'I/O a controllo di programma) sono:

- con periferiche lente, un eccessivo spreco del tempo di CPU, che per la maggior parte del tempo rimane occupata nel ciclo di busy waiting.
- Con periferiche veloci, il lavoro svolto dalla CPU è quasi interamente dovuto all'effettivo trasferimento dei dati.

Gestione dell'I/O

- A controllo di programma
- Ad interruzione (interrupt)
- Ad accesso diretto alla memoria (DMA)

Interrupt

E' la periferica a segnalare al processore (su una linea del bus dedicata) di avere bisogno di attenzione.

La segnalazione viene chiamata ***interrupt*** perché interrompe il normale funzionamento del processore (***interrupt request***).

Quando il processore "se ne accorge" (fase di fetch), riceve un segnale di ***interrupt acknowledge***.

Viene eseguita una procedura speciale, chiamata ***procedura di risposta all'interrupt***.

Problema: Il programma utente deve potere procedere dal punto in cui è stato interrotto ➔ *Salvataggio del contesto.*

Interrupt – esempio – comando print

1. Invio del comando print.
2. Se la periferica è in stato busy, CPU torna alla sua attività, scaricando sul registro di controllo la richiesta di output.
3. Quando la periferica diventa ready, viene inviato un interrupt.
4. Il programma di risposta all'interruzione, provvederà a trasferire alla periferica il dato che su vuole stampare.

Interruzioni, eccezioni e trap

Eccezione. Evento inatteso, **interno** al processore che cambia il flusso di esecuzione del programma.
Trap. Meccanismo con cui l'utente chiede esplicitamente l'intervento del SO (e.g. breakpoints).
Interrupt. Evento inatteso, asincrono, esterno che cambia il flusso di esecuzione di un programma.

Tipo di evento	Provenienza	Terminologia MIPS
Richiesta di un dispositivo di I/O	Esterna	Interruzione
Chiamata al SO da parte di un applicativo	Interna	Eccezione
Overflow aritmetico	Interna	Eccezione
Malfunzionamento hardware	Interna / Esterna	Eccezione / Interruzione
Uso di un'istruzione indefinita	Interna	Eccezione

Individuazione dell'eccezione

• Registro causa (**cause**, MIPS)
La procedura del SO deve decodificare il registro cause.

• Registro di stato (**state** register)
contiene le informazioni necessarie per gestire l'interrupt

• Interruzione vettorializzata (Intel).
Salto ad un indirizzo diverso per ogni evento.
Numero limitato di istruzioni per servire l'interruzione.

La risposta alle eccezioni

1. La CPU salva l'***indirizzo*** dell'istruzione nel registro **EPC**.
2. Salvataggio dello status register.
3. Caricamento nel cause register del codice di eccezione (vale 0 nel caso di interrupt esterno per il MIPS) .
4. Trasferire il controllo ad una procedura dedicata del SO (exception handler).
5. Esecuzione della procedura.
6. Terminazione del programma, fornire dei servizi o ritorno alla procedura interrotta.

Salvataggio del contesto

E' il programma di risposta all'interruzione che salva l'informazione (i registri) del programma che verranno utilizzati all'interno del programma di risposta.

I registri possono essere memorizzati nell'area di kernel, non nell'area stack.

Soluzione più semplice: le procedure di risposta alle eccezioni sono *non rientranti.*

I/O ad interrupt - costo

Hard disk:

Frequenza di clock è 500Mhz
Trasferimento di blocchi di 4 parole
Trasferimento a 4Mbyte/s
Il disco sta trasferendo dati solamente per il 5% del suo tempo.
Il costo di ogni interruzione è 500 cicli di clock.

Trasferimento di 1Mword/s => Occorrono 250k interruzioni/s
Costo dell'interruzione 250k int/s * 500 cicli_clock = 125M cicli_clock/s
Frazione di utilizzo del processore per il trasferimento (interrupt):
125M / 500M = 25%
Frazione del processore utilizzata in realtà: 125M / 500M *0.05 = 1,25%

Interrupt gerarchici

Accodamento degli input: FIFO.
Annidamento degli input: LIFO (cf. stack)

Occorre definire una priorità.

Maschere di interrupt.

Ad ogni interrupt viene associato un livello.
Il livello corrente è associato allo stato del sistema.
La maschera dell'interrupt viene memorizzata nello status register (8 livelli per il MIPS)
La maschera degli interrupt pending viene caricata nel cause register. Per ogni bit a 1, esiste pending un interrupt di quel livello.

Codifica di priorità.

Gestione dell'I/O

- A controllo di programma
- Ad interruzione (interrupt)
- Ad accesso diretto alla memoria (DMA)

DMA

Tra il momento in cui termina l'invio del comando al controller ed il momento in cui il dato è disponibile sul controller, la CPU può fare altro (tipicamente l'esecuzione di un altro programma).

Il meccanismo interrupt driven non svincola la CPU dal dovere eseguire le operazioni di trasferimento dati.

Per periferiche veloci, le operazioni di trasferimento dati occupano un tempo preponderante rispetto al tempo speso in spin lock.

Per evitare l'intervento della CPU nella fase di trasferimento dati, è stato introdotto il protocollo di trasferimento in Direct Memory Access (DMA).

Caratteristiche della DMA

Il DMA controller è un processore specializzato nel trasferimento dati tra dispositivo di I/O e memoria centrale.

Per attivare il trasferimento viene richiesto alla CPU:
1. Spedire al DMA controller il tipo di operazione richiesta
2. Spedire al DMA controller l'indirizzo da cui iniziare a leggere/scrivere i dati.
3. Spedire al DMA controller il numero di byte riservati in memoria.

Per attivare il trasferimento al controller viene richiesta la corretta lettura dello stato della memoria e l'aggiornamento dell'indirizzo a cui trasferire il dato. *E' il controller che gestisce il trasferimento del singolo dato.*

Caratteristiche della DMA

La CPU si svincola completamente dall'esecuzione dell'operazione di I/O.

Il controller avvia l'operazione richiesta e trasferisce i dati da/verso memoria mentre la CPU sta facendo altro.

Dopo avere trasferito tutti i dati, il DMA invia un interrupt alla CPU per segnalare il completamento del trasferimento.

La CPU perciò controlla il controller.

I/O DMA - costo

Hard disk:

Frequenza di clock è 500Mhz
Trasferimento di blocchi di 8kbyte per ogni DMA.
Trasferimento a 4Mbyte/s
Il costo dell'inizializzazione del DMA è di 1000 cicli di clock.
Il costo dell'interruzione al termine del DMA è di 500 cicli di clock.

Per ciascun trasferimento DMA occorre:

1000 + 500 cicli di clock = tempo di inizio + tempo di fine
Numero di DMA: 4(Mbyte/s) / 8kbyte = 500 / s
Numero di cicli di clock richiesti: 1500 * 500 = 750,000
Frazione del processore utilizzata: 750k / 500M = 0,2%

Problema?

Il bus

- Tipi di bus
- Controllo accesso al bus (arbitraggio)

Arbitraggio del bus

Il DMA controller sottrae alla CPU il controllo del bus per il tempo necessario al trasferimento.

La comunicazione su bus deve essere regolata attraverso un **protocollo di comunicazione**.

Arbitraggio del bus

Viene introdotto il concetto di bus master (padrone del bus), il cui scopo è quello di controllare l'accesso al bus.

L'architettura più semplice è quella che prevede un unico bus master (il processore) in cui tutte le comunicazioni vengono mediate dal processore stesso.

Questo può creare un collo di bottiglia. Ad esempio nel caso di trasferimento di dati da I/O a memoria.

Si utilizza allora un'architettura con più dispositivi master.

In questo caso occorre definire e rispettare un protocollo che coordini i vari dispositivi bus master.

Protocollo di arbitraggio

Occorre stabilire quale master autorizzare all'utilizzo del bus.

Ad ogni dispositivo viene assegnata una *priorità*.

Il dispositivo a priorità maggiore può accedere prima al bus.

Meccanismo di accesso al bus diventa:
1. Richiesta del bus (bus request)
2. Assegnamento del bus (bus grant)

Problema è assicurare una *fairness*.

Compromesso tra fairness e priorità.

Schemi di arbitraggio

- Daisy chain. Semplice. Può essere poco fair.
- Arbitraggio centralizzato parallelo. Complesso. Centralizzato. Può diventare il collo di bottiglia nell'utilizzo del bus.
- Arbitraggio distribuito con auto-selezione. Ogni unità determina se la sua richiesta può essere accolta o meno analizzando il proprio stato e quello delle altre richieste.
- Arbitraggio distribuito con rilevamento delle collisioni. Richiesta indipendente del bus. Se contemporanea, protocollo di selezione. Schema utilizzato dalla rete Ethernet.

Il bus

- Tipi di bus
- Controllo accesso al bus (arbitraggio)

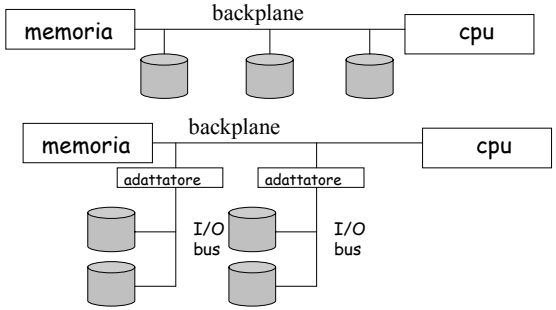
Il bus di sistema

- Il bus di sistema costituisce il canale di comunicazione tra le varie componenti di un elaboratore.
- Il bus di sistema è composto da:
  - **Bus indirizzi**
  - **Bus dati**
  - **Bus di controllo**
- Può diventare un collo di bottiglia in quanto le sue prestazioni sono limitate da:
  - la lunghezza;
  - il numero di device connessi
    (Formato PAL video: 720 x 560 x 3 byte x 25 frame/s => 30,240,000 words/s)

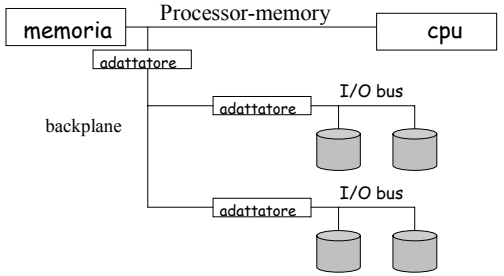
Tipologie di bus

- All'interno dell'ampiezza di banda massima si può:
  - **Aumentare la velocità di trasferimento.** Buffer grossi.
  - **Ridurre i tempi di risposta.** Buffer piccoli.
- Tre tipi principali:
  - **processor-memory**: lunghezza ridotta, molto veloci
  - **I/O**: notevole lunghezza, molti device connessi
  - **backplane:** servono per far coesistere la memoria, il processore e i dispositivi di I/O su di un unico bus

Tipologie di bus



Tipologie di bus



Tipologie di bus

- Esistono due schemi principali di comunicazione su di un bus:
  - sincrono
  - asincrono

Bus sincroni

- La linea di controllo è dotata di un segnale di sincronizzazione **(clock)** ed esiste un protocollo di comunicazione scandito dai cicli di clock.
- Questo tipo di protocollo permette di ottenere bus molto veloci
- Svantaggi:
  - ogni device deve essere sincronizzato
  - i bus non possono avere lunghezze elevate
- I bus processor-memory sono spesso sincroni in quanto:
  - hanno dimensioni ridotte
  - hanno pochi elementi connessi

Bus asincroni

- Un bus asincrono **non** è dotato di clock.
- La comunicazione tra due parti avviene mediante un protocollo di **handshaking**.
- I bus asincroni possono avere lunghezza elevata e connettere molti dispositivi.
- Spesso i bus di I/O sono asincroni.

I bus

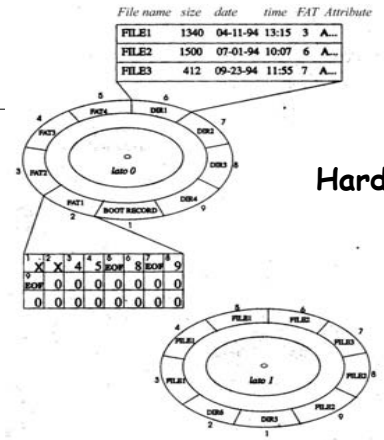
Caratteristica	PCI	SCSI
Tipo di bus	Back-plane	I/O
Ampiezza di base del bus (numero di segnali)	32-64	8-32
Numero di dispositivi master	molti	molti
Temporizzazione	Sincrono 33-66Mhz	Asincrono o sincrono (5-20Mhz)
Ampiezza di banda di picco teorica	133-512MB/2	5-30MB/s
Ampiezza di banda stimata raggiungibile per bus di base	80MB/s	2,5-40MB/s
Massimo numero di dispositivi	1024 (32 dispositivi per segmento)	7-31
Massima lunghezza del bus	0,5 metri	25 metri
Nome dello standard	PCI	ANSI X3.131

Dischi magnetici

- Consentono di memorizzare dati in modo non volatile.
- I dati sono letti/scritti mediante una testina.
- I dischi magnetici sono di due tipi principali:
  - hard disk
  - floppy disk

Hard disk

- Costituiti da un insieme di piatti rotanti (da 1 fino a 25) ognuno con due facce.
- La pila dei piatti viene fatta ruotare alla stessa velocità (5,400 – 10,000 rpm = revolutions per minute).
- Ogni faccia è divisa in circonferenze concentriche chiamate **tracce** (1000-5000).
- Ogni traccia è suddivisa in **settori** (64 - 200).
- Il settore è la più piccola unità che può essere letta/scritta da/su disco (blocco da 512 Byte o 1024 Byte).
- Esiste una testina per ogni faccia.
- Le testine di facce diverse sono collegate tra loro e si muovono contemporaneamente.

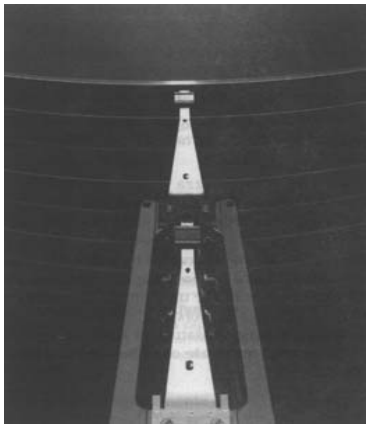


Hard-disk: memorizzazione delle informazioni

La traccia 0 contiene il contenuto del disco.

Hard disk

- Le testine si muovono in modo solidale.
- L'insieme delle tracce di ugual posto su piatti diversi è chiamato **cilindro**.
- La quantità di dati che possono essere memorizzati per traccia dipende dalla qualità del disco.
- Solitamente, ogni traccia di un disco contiene la stessa quantità di bit ⇒ le tracce più esterne memorizzano informazione con densità minore.



Hard disk – lettura / scrittura

- Per leggere/scrivere informazioni sono necessari tre passi:
  - la testina deve essere posizionata sulla traccia corretta;
  - il settore corretto deve passare sotto la testina;
  - i dati devono essere letti o scritti.

- **Tempo di seek (ricerca)**: tempo per muovere la testina sulla traccia corretta.
- **Tempo di rotazione**: tempo medio per raggiungere il settore da trasferire (tempo per 1/2 rotazione).
- **Tempo di trasferimento**: tempo per trasferire l'informazione.
- A questi tempi va aggiunto il tempo per le operazioni del controller.

Hard disk - prestazioni

- Tempo medio di seek: da 8 a 20 ms (può diminuire di più del 75% se si usano delle ottimizzazioni).
- Tempo medio di rotazione: da 2.8 ms a 5.6 ms.
- Tempo medio di trasferimento: 2/15 MB per secondo e oltre con cache.
- Tempo di controllo (utilizzato dalla logica del controller).

Qual è il tempo di lettura/scrittura di un settore di 512byte in un disco che ha velocità di rotazione di 7,200 rpm? Il tempo medio di seek è di 12ms, la velocità di trasferimento di 10Mbyte/s ed il tempo aggiuntivo richiesto dal controllore è di 2ms.

12ms + 4,2ms + 0,5kbyte / 10Mbyte/s + 2ms = 12 + 4,2 + 0,05 + 2 = 18,25ms

Per un tempo di seek medio pari al 25% del tempo nominale (t_seek = 3ms)
3ms + 5,6ms + 0,5kbyte / 10Mbyte/s + 2ms = 9,25ms

CD-ROM

- I CD-ROM sono basati sulla tecnologia laser per la memorizzazione delle informazioni.
- Memorizzano l'informazione codificata con incisioni di forme caratteristiche sulla superficie del disco.
- Un raggio laser colpisce la superficie del disco e viene da questa riflesso in modo diverso a seconda della forma della superficie colpita.
- Su CD-ROM è possibile immagazzinare informazione con una densità maggiore rispetto ai dischi magnetici.
- Un CD-ROM può memorizzare più di 600 MB di dati.