

4 La gestione dell'I/O

Indice

- 4.1 Prestazioni e generalità
- 4.2 Modelli di funzionamento
- 4.3 Supporti su disco
- 4.4 Gestione software

4.1 Prestazioni e generalità

I/O come collo di bottiglia

- **CPU:** la velocità raddoppia ogni 18 mesi
- **DRAM:** la capacità raddoppia ogni 2-3 anni
- **Disco rigido:** la capacità raddoppia ogni 3 anni
- **I/O:** le prestazioni migliorano del 10% all'anno
 - Limiti derivanti da seri problemi *meccanici*
- Le prestazioni generali degli elaboratori aumentano poco per via del *collo di bottiglia* causato dall'I/O

4.1 Prestazioni e generalità

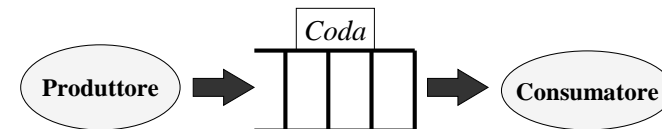
Misura delle prestazioni

- Le prestazioni di I/O si misurano in termini di
 - **Tempo di risposta** (o di *latenza*)
 - L'intervallo di tempo tra l'emissione di una richiesta di I/O ed il suo soddisfacimento
 - **Banda passante** (o *throughput*)
 - Il numero medio di operazioni di I/O completabili per unità di tempo (equivalente alla quantità totale di dati trasmessi)

4.2 Modelli di funzionamento

Modello di coda - 1

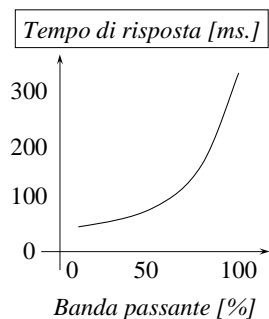
Un sistema di I/O può essere rappresentato con un modello produttore-consumatore dotato di una coda dei prodotti per compensare le diverse velocità relative



4.2 Modelli di funzionamento

Modello di coda - 2

- Si ha basso tempo di risposta con coda vuota e consumatore libero
- Si ha elevata banda passante con coda mai vuota e consumatore mai inoperoso



Lezione

Architettura degli Elaboratori - 1 - A. Sperduti

Pagina 5

4.2 Modelli di funzionamento

Classificazione dei dispositivi di I/O

- **A blocchi**
 - Il dispositivo emette e riceve dati strutturati in *blocchi* di dimensione fissa (512B-32kB), ciascuno con un indirizzo proprio
 - Disco, nastro
 - I blocchi possono essere indirizzati (per lettura o scrittura) *indipendentemente*
- **A caratteri**
 - Il dispositivo emette e riceve sequenze *non strutturate* di caratteri
 - Stampante, interfaccia di rete, *mouse*
 - Non consente indirizzamento dei dati

Lezione

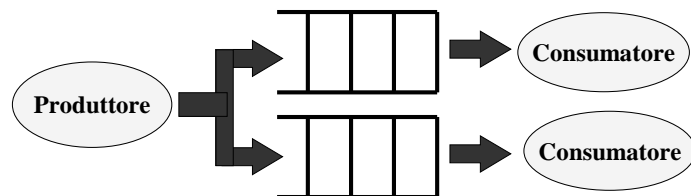
Architettura degli Elaboratori - 1 - A. Sperduti

Pagina 7

4.2 Modelli di funzionamento

Modello di coda - 3

- Il tempo di risposta non è facilmente migliorabile
- La banda passante può essere aumentata moltiplicando i produttori e/o i consumatori



Lezione

Architettura degli Elaboratori - 1 - A. Sperduti

Pagina 6

4.2 Modelli di funzionamento

Struttura dei dispositivi di I/O - 1

- **Parte meccanica**
 - Il dispositivo vero e proprio
- **Parte elettronica**
 - Chiamata *device controller* (o adattatore)
 - Un singolo *controller* può gestire più dispositivi
- **L'interfaccia** tra dispositivo e *controller* è spesso di basso livello (primitivo)
 - La funzione del *controller* è precisamente quella di offrire al S/O un' interfaccia meno rozza verso il dispositivo

Lezione

Architettura degli Elaboratori - 1 - A. Sperduti

Pagina 8

4.2 Modelli di funzionamento

Struttura dei dispositivi di I/O - 2

- I *controller* utilizzano registri e banchi di memoria per comunicare con la CPU
 - I registri contengono comandi
 - Di configurazione, di controllo, di esecuzione
 - I banchi di memoria (buffer) contengono i dati in emissione o in ricezione
 - Il S/O ha accesso ad entrambi

Lezione

Architettura degli Elaboratori - 1 - A. Sperduti

Pagina 9

4.2 Modelli di funzionamento

Struttura dei dispositivi di I/O - 4

- La tecnica *memory-mapped I/O* ha diversi vantaggi
 - Non necessita di istruzioni speciali
 - Le istruzioni che accedono memoria 'normale' accedono anche le aree di I/O
 - Il software di controllo di dispositivo può essere scritto interamente in linguaggi ad alto livello
 - Consente una più agevole protezione
 - è sufficiente nascondere le aree di I/O allo spazio di indirizzamento di utente (*privilegi*)
 - Con la segmentazione, *più* aree di I/O possono mappare sul *medesimo* spazio di indirizzamento fisico

Lezione

Architettura degli Elaboratori - 1 - A. Sperduti

Pagina 11

4.2 Modelli di funzionamento

Struttura dei dispositivi di I/O - 3

- Varie tecniche permettono di includere i registri del *controller* nello spazio di indirizzamento della CPU
 - Ciascun registro viene visto come una **porta speciale di I/O** indirizzata da istruzioni speciali
 - Tutti i registri vengono visti e trattati come parte della memoria (*memory-mapped I/O*)
 - La corrispondente area non viene allocata liberamente
 - Soluzioni ibride (Pentium)
 - Banchi di memoria del controller come *memory-mapped I/O*
 - Registri di controllo come porte speciali di I/O

Lezione

Architettura degli Elaboratori - 1 - A. Sperduti

Pagina 10

4.2 Modelli di funzionamento

Struttura dei dispositivi di I/O - 5

- La tecnica *memory-mapped I/O* presenta anche alcuni svantaggi
 - Non si presta all'uso di cache
 - Il dato rilevante è *sempre e solo* nella memoria del dispositivo
 - Occorre disabilitare selettivamente la cache
 - Non è compatibile con architetture a 'bus multipli'
 - I dispositivi di I/O non possono rispondere ad indirizzi emessi su bus non connessi
 - Occorre filtrare gli indirizzi emessi dalla CPU ed instradarli sul bus appropriato
 - Filtraggio a sorgente piuttosto che a destinazione

Lezione

Architettura degli Elaboratori - 1 - A. Sperduti

Pagina 12

4.2 Modelli di funzionamento

Accesso ai dispositivi di I/O - 1

- Tutti i dispositivi si affacciano sul canale indirizzi controllato dalla CPU
- Per ogni indirizzo emesso sul canale, la CPU specifica se per lettura o scrittura
- Il *tipo* di indirizzo emesso individua univocamente l'unità chiamata a rispondere
 - Blocchi diversi di indirizzi sono assegnati ad unità diverse
 - Nessuna sovrapposizione, nessun conflitto

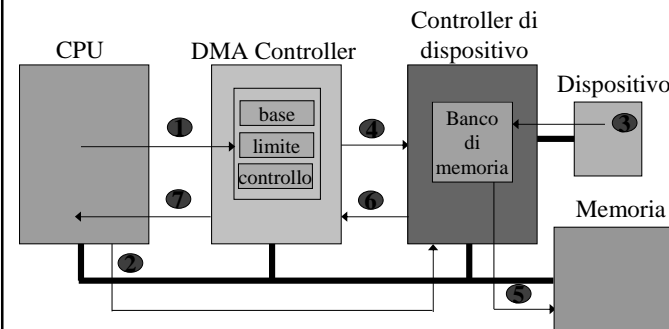
Lezione

Architettura degli Elaboratori - 1 - A. Sperduti

Pagina 13

4.2 Modelli di funzionamento

Accesso ai dispositivi di I/O - 3



Lezione

Architettura degli Elaboratori - 1 - A. Sperduti

Pagina 15

4.2 Modelli di funzionamento

Accesso ai dispositivi di I/O - 2

- Quando CPU e dispositivo hanno velocità relative molto diverse non conviene che la più veloce debba attendere quella più lenta
- Il meccanismo detto **Direct Memory Access (DMA)** consente tale disaccoppiamento
- Questo richiede l'uso di un *DMA controller* capace di prendere occasionalmente possesso dei canali indirizzi e dati

Lezione

Architettura degli Elaboratori - 1 - A. Sperduti

Pagina 14

4.2 Modelli di funzionamento

Accesso ai dispositivi di I/O - 4

- La CPU programma il *DMA controller* informandolo della destinazione e dell'ampiezza del trasferimento da effettuare
- La CPU comanda al *controller* di dispositivo di acquisire dati dal dispositivo
- Il *controller* di dispositivo salva i dati acquisiti nella sua memoria interna
- Il *DMA controller* comanda al *controller* di dispositivo di trasferire i dati acquisiti in memoria principale
- Il *controller* di dispositivo effettua il trasferimento
- Il *controller* di dispositivo informa il *DMA controller* l'esito del trasferimento
- Il *DMA controller* informa la CPU dell'esito del trasferimento

Lezione

Architettura degli Elaboratori - 1 - A. Sperduti

Pagina 16

4.2 Modelli di funzionamento

Accesso ai dispositivi di I/O - 5

- Il *DMA controller* può accedere al canale dati in uno di due modi
 - **Una parola alla volta**, sottraendo di tanto in tanto alla CPU il controllo sul canale (*cycle stealing*)
 - **Per blocchi**, prendendo possesso del canale per una serie di trasferimenti (*burst mode*)
 - La CPU è bloccata in entrambi i casi, ma il *burst mode* è **più efficace** perché l'acquisizione del canale è onerosa
- La CPU può fare a meno del DMA se essa è molto più veloce del *DMA controller*

Lezione

Architettura degli Elaboratori - 1 - A. Sperduti

Pagina 17

4.3 Supporti su disco

Dischi magnetici - 2

Diametro dei piatti: da 1"1/4 a 14"

Capacità: da 360kB a 5 GB

Piatti e testine: da 1 a 16

WD 18300 hard disk

Cilindri: 10601

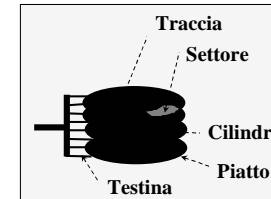
Tracce per cilindro: 12

Settori per traccia: 281 (in media)

Capacità settore: 512B

Rivoluzioni al minuto (RPM): 7202

Velocità angolare costante



Lezione

Architettura degli Elaboratori - 1 - A. Sperduti

Pagina 19

4.3 Supporti su disco

Dischi magnetici - 1

- Memoria secondaria o di livello superiore
- Adibiti alla memorizzazione *permanente*
- Grande capacità e relativa lentezza
 - *Floppy disk*
 - Di facile trasporto e di costo contenuto
 - *Hard disk*
 - Di maggiore capacità e velocità di accesso (più piatti in parallelo)

Lezione

Architettura degli Elaboratori - 1 - A. Sperduti

Pagina 18

4.3 Supporti su disco

Dischi magnetici - 3

$T_A =$ Tempo medio di risposta (*access time*)

$$T_A = T_S + T_L$$

$T_S =$ Tempo di posizionamento delle testine sopra la traccia desiderata (*seek time*)

5-20 ms difficilmente riducibile

$T_L =$ Arrivo del settore sotto la testina (*latency time*)

Dipende dalla velocità di rotazione

Esempio

$RPM=3600 \Rightarrow RPS=60 \Rightarrow 1 \text{ rotazione} \approx 16.7ms \Rightarrow T_L=8.35ms$

Lezione

Architettura degli Elaboratori - 1 - A. Sperduti

Pagina 20

4.3 Supporti su disco

Dischi magnetici - 4

- La banda passante dipende da
 - Dimensione dell'unità minima di trasferimento
 - singolo settore o gruppi di settori (*cluster*)
 - Velocità di rivoluzione dei piatti
 - Densità di memorizzazione
 - Disponibilità di dispositivi cache integrati
- Banda minima: 2 MB/s
- Banda tipica: 15-40 MB/s

Lezione Architettura degli Elaboratori - 1 - A. Sperduti Pagina 21

4.3 Supporti su disco

Dischi ottici - 1

- Rimovibili, economici e veloci se a sola lettura
- Preferiti dai produttori perché la riproduzione (scrittura) è ancora costosa

	CDROM	nastro
dimensione	5"1/4	8mm
capacità	0,6 GB	5 GB
£ supp.	3 K	50 K
£ drive	200 K	400 K
accesso	R	R/W

Lezione Architettura degli Elaboratori - 1 - A. Sperduti Pagina 23

4.3 Supporti su disco

Nastri magnetici

- Utilizzano la stessa tecnologia di memorizzazione dei dischi
- Permettono solo accesso *sequenziale*
- Hanno lunghezza virtualmente illimitati ma minor banda passante
- Sono facilmente rimovibili
- Si prestano bene al salvataggio di grandi archivi di dati e/o di interi sistemi
- Tecnologia emergente: Rotating head Digital Audio Tape (R-DAT)

Lezione Architettura degli Elaboratori - 1 - A. Sperduti Pagina 22

4.3 Supporti su disco

Dischi ottici - 2

- Si basano su un'unica spirale che muove dal centro
- La rotazione avviene a velocità angolare costante
 - Nei più recenti ad *accelerazione* angolare costante
- Ogni blocco dati può avere il proprio codice di correzione (ECC)
- Possono essere facilmente riprodotti
 - La produzione di massa è molto economica

Lezione Architettura degli Elaboratori - 1 - A. Sperduti Pagina 24

4.3 Supporti su disco

Dischi ottici - 3

- L'impressione delle tracce è fatta mediante depressioni (*pit*) che emettono un particolare riflesso quando illuminate
- Una transizione da una depressione ad una zona piatta (*land*) o viceversa denotano un 1; la loro assenza denota uno 0
- La spirale, sulla quale *pit* e *land* sono impresse, effettua 22188 rivoluzioni con un raggio massimo di 32 mm (complessivamente 5,6 km di traccia!)

Lezione

Architettura degli Elaboratori - 1 - A. Sperduti

Pagina 25

4.4 Gestione software

Classificazione

- Il metodo più appropriato di gestione software dell'I/O dipende dal *tempo di risposta* del dispositivo
 - Dispositivi più lenti della CPU
 - Dispositivi veloci come la CPU
 - Dispositivi più veloci della CPU
 - Dispositivi a velocità variabile
- Occorre anche tenere conto della *banda passante*

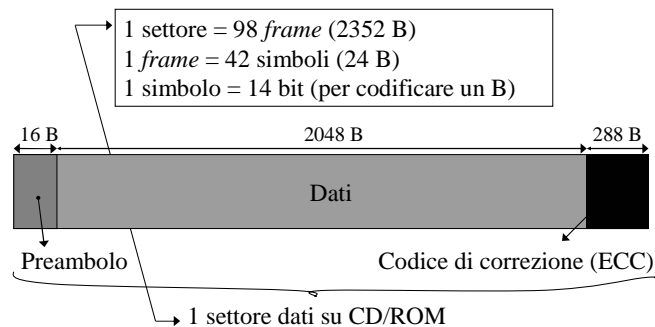
Lezione

Architettura degli Elaboratori - 1 - A. Sperduti

Pagina 27

4.3 Supporti su disco

Dischi ottici - 4



Lezione

Architettura degli Elaboratori - 1 - A. Sperduti

Pagina 26

4.4 Gestione software

Periferiche lente

- Occorre fornire una coda *ampia* all'ingresso del dispositivo e *disaccoppiare* l'attività di I/O dalla sua gestione
 - Dimensionamento della coda
 - Scelta progettuale tra
 - **Gestione programmata dell'I/O**
 - Con controllo esplicito a software (*polling / busy waiting*)
 - **Gestione dell'I/O a domanda**
 - Con controllo software solo quando richiesto (*interrupt-driven*)

Lezione

Architettura degli Elaboratori - 1 - A. Sperduti

Pagina 28

4.4 Gestione software

Periferiche veloci

- Non richiedono particolari accorgimenti in quanto i tempi di risposta sono confrontabili a quelli della CPU
 - Richiedono una coda *piccola* per disaccoppiare meglio l'operazione di I/O dalla sua gestione
 - Consentono l'uso sia della tecnica del *polling* che di quella dell'*interrupt*, con prestazioni analoghe

Lezione

Architettura degli Elaboratori - 1 - A. Sperduti

Pagina 29

4.4 Gestione software

Modalità polling (busy waiting)

- La CPU ciclicamente interroga il dispositivo per verificarne lo stato di libero e l'eventuale esito dell'operazione
 - Questa modalità forza *sincronizzazione* tra I/O e gestore software
 - La CPU ha controllo *totale* sull'I/O
 - La CPU dedica tempo di esecuzione al controllo dello stato e del progresso delle operazioni di I/O
 - Spesso a vuoto se il dispositivo è molto lento

Lezione

Architettura degli Elaboratori - 1 - A. Sperduti

Pagina 31

4.4 Gestione software

Periferiche molto veloci

- Richiedono una coda *ampia* in uscita dal dispositivo ed il trasferimento *diretto* tra dispositivo e memoria
 - Gestione dell'I/O mediante **DMA**
 - Minimo coinvolgimento software ad onere contenuto
 - La CPU perde il controllo assoluto dei canali di comunicazione (*bus*)
 - Occorrono canali molto ampi e veloci per il trasferimento diretto

Lezione

Architettura degli Elaboratori - 1 - A. Sperduti

Pagina 30

4.4 Gestione software

Onere di polling (overhead)

F = frequenza di clock della CPU = 200 MHz

NP = numero cicli di clock per lettura (*poll*) = 200

NL = numero letture necessarie al secondo =

– Mouse = 30

– Floppy = 50 kB/s emessi su canale a 16 bit
= 25.000

– Hard disk = 4 MB/s emessi su canale a 32 bit
= 1.000.000

PO = onere di polling (*overhead*) = $(NL \times NP) / F$ [%]

Mouse = 0,003% Floppy = 2,5% Hard disk = 100%

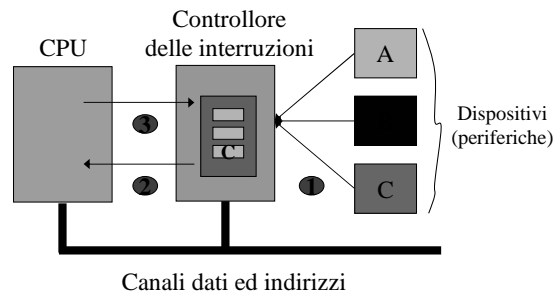
Lezione

Architettura degli Elaboratori - 1 - A. Sperduti

Pagina 32

4.4 Gestione software

Il meccanismo delle interruzioni - 1



Lezione

Architettura degli Elaboratori - 1 - A. Sperduti

Pagina 33

4.4 Gestione software

Il meccanismo delle interruzioni - 3

- Il valore emesso sul canale dal controllore delle interruzioni designa il servizio richiesto alla CPU
- Tale valore è un indice nel **vettore delle interruzioni**, contenente l'indirizzo della prima istruzione della procedura di servizio
- La base del vettore delle interruzioni è fissa e *nota* al S/O

Lezione

Architettura degli Elaboratori - 1 - A. Sperduti

Pagina 35

4.4 Gestione software

Il meccanismo delle interruzioni - 2

- Il dispositivo **C** segnala che ha bisogno di attenzione (p.es. ha completato un trasferimento)
- Il controllore delle interruzioni segnala alla CPU che il dispositivo **C** chiede attenzione
- La CPU attiva il servizio di gestione dell'interruzione richiesta e ne notifica l'inizio al controllore delle interruzioni
 - Solo *dopo* tale notifica il controllore delle interruzioni si dispone ad accogliere altre richieste dai dispositivi

Lezione

Architettura degli Elaboratori - 1 - A. Sperduti

Pagina 34

4.4 Gestione software

Il meccanismo delle interruzioni - 4

- L'esecuzione della procedura di servizio *interrompe* l'attività corrente della CPU
- Per poter successivamente riprendere tale attività, occorre salvare l'informazione necessaria per ripristinare il corretto *contesto di esecuzione*
 - Cosa determina il contesto (minimo)
 - Dove salvarlo

Lezione

Architettura degli Elaboratori - 1 - A. Sperduti

Pagina 36

4.4 Gestione software

Il meccanismo delle interruzioni - 5

- Minimizzazione del contesto da salvare
 - Per minimizzare il costo temporale di attivazione del servizio di interruzione e del successivo ripristino
 - Più lungo è il tempo di attivazione, più lungo è il periodo nel quale il controller di interruzioni *non* presta ascolto ai dispositivi controllati
 - Il contesto minimo comprende il valore corrente di IP e quello dei registri di configurazione del processo (p.es. base della tabella dei descrittori di segmento; registro di segmento)

Lezione

Architettura degli Elaboratori - 1 - A. Sperduti

Pagina 37

4.4 Gestione software

Il meccanismo delle interruzioni - 7

- Quale è la relazione tra lo stato interno della CPU e l'arrivo (asincrono) di una interruzione?
 - L'interruzione dovrebbe idealmente avere effetto al confine tra due istruzioni successive
 - Le CPU con *pipeline* sovrappongono l'esecuzione di molte istruzioni
 - Confine meno chiaro
 - Le CPU super-scalari (*out-of-order execution*) modificano l'ordine di esecuzione specificato dal programma
 - Confine totalmente confuso

Lezione

Architettura degli Elaboratori - 1 - A. Sperduti

Pagina 39

4.4 Gestione software

Il meccanismo delle interruzioni - 6

- Localizzazione dell'area di salvataggio
 - Deve essere *sicura*
 - Una eccezione di accesso durante un servizio di interruzione sarebbe disastrosa
 - Meglio fidarsi del S/O piuttosto che del processo interrotto (di utente)
 - *Non* deve essere *costosa* da accedere
 - L'accesso a zone di memoria di S/O può richiedere l'aggiornamento dei registri di configurazione ed il relativo caricamento da memoria secondaria

Lezione

Architettura degli Elaboratori - 1 - A. Sperduti

Pagina 38

4.4 Gestione software

Il meccanismo delle interruzioni - 8

- Le **interruzioni precise** hanno 4 proprietà
 - L'IP viene salvato in un luogo noto e sicuro
 - Tutte le istruzioni precedenti ad IP sono state completate
 - Nessuna istruzione successiva ad IP è stata completata
 - Possono essere state emesse ma il loro effetto deve essere annullato
 - Lo stato di esecuzione dell'istruzione designata da IP è noto

Lezione

Architettura degli Elaboratori - 1 - A. Sperduti

Pagina 40

4.4 Gestione software

Il meccanismo delle interruzioni - 9

- Modelli di interruzioni che non abbiano tali proprietà sono detti *imprecisi* e sono estremamente complessi da gestire per il S/O
- Architetture super-scalari come il Pentium Pro sono capaci di offrire interruzioni precise al prezzo di una logica assai complessa
 - Questo è uno dei costi della *backward compatibility*
- Alcune architetture consentono di disabilitare la modalità imprecisa ma le prestazioni crollano

Lezione

Architettura degli Elaboratori - 1 - A. Sperduti

Pagina 41

4.4 Gestione software

Onere di DMA (overhead)

- F** = frequenza di clock della CPU = 200 MHz
NP = cicli di clock per inizializzazione di DMA = 800
 = per trasferimento di 512 B (1 settore) = 1024
 in *burst mode* (sottraendo cicli alla CPU)
- NL** = numero trasferimenti DMA al secondo =
 – Floppy = 50 kB/s ÷ 1 settore = 100
 – Hard disk = 4 MB/s ÷ 1 settore ≈ 8.000
- PO** = onere di DMA (*overhead*) = $(NL \times NP) / F$
Floppy = 0,05% **Hard Disk = 4,2%**

Lezione

Architettura degli Elaboratori - 1 - A. Sperduti

Pagina 43

4.4 Gestione software

Onere di interruzione (overhead)

- F** = frequenza di clock della CPU = 200 MHz
NP = numero cicli di clock per un servizio basico di interruzione = 300
 = con trasferimento di 512 B (1 settore) = 2.860
- NL** = numero interruzioni al secondo =
 – Mouse = 30
 – Floppy = 50 kB/s ÷ 1 settore = 100
 – Hard disk = 4 MB/s ÷ 1 settore ≈ 8.000
- PO** = onere di interruzione (*overhead*) = $(NL \times NP) / F$
Mouse = 0,0045% **Floppy = 0,1%** **Hard disk = 11%**

Lezione

Architettura degli Elaboratori - 1 - A. Sperduti

Pagina 42