

I/O

Ovviamente in un PC esistono le periferiche. Esse collaborano con la CPU per inviarsi e ricevere dati usando dei canali denominati BUS.

Il bus comune/di sistema collega la CPU con la memoria e le periferiche, ed è composto da:
bus di controllo (operazioni + sincronizzazione tra dispositivi), indirizzo e dati.

Bus di Sistema

- Vantaggi: elevata flessibilità, semplicità, basso costo
- Svantaggi: gestione complessa del canale condiviso.
- Svantaggi: gestione complessa del canale condiviso

Interfaccia: hardware(dispositivo) + software(driver che sta nella RAM) → collegati alla cpu con il bus.

In primis ogni periferica ha un area di memoria riservata nella RAM in cui sono presenti i registri che essa utilizza. Quest'area è accessibile solo dall'OS, l'utente/programma deve chiedere all'OS se vuole accedere a quell'area di memoria.

Non è l'unica maniera per gestire il problema, infatti è possibile avere dei BUS dedicati (uno per la CPU e uno per la periferica)

Nella parte software hanno dei registri di dati (buffer). Due tipi di registri: registro dei dati
registro di stato/controllo (status register): letto dalla cpu, stato delle periferica.

Non è la CPU a controllare le periferiche ma decide lei quando "occuparsi della questione".

Ci sono 3 diversi metodi con cui un PC può gestire le periferiche:

GESTIONE DEL PROGRAMMA

La periferica ha un ruolo passivo, attende l'invio dei dati da parte della cpu, poi fa quello che deve fare e infine ripete tutto da capo.

Periferica attenda → CPU controlla registro di stato e trasferisce quando registro è ready → RAM → CPU → Periferica

La CPU è bloccata nello stato di busy waiting.

Il vantaggio è che la risposta è immediata.

INTERRUZIONI (implica hardware aggiuntivo)

L'avverarsi dello stato di ready viene controllato dalla periferica stessa. Il processore, mentre aspetta che la periferica passi nello stato di ready, fa altro.

L'interrupt è sempre tra 2 istruzioni, viene prima completata una e poi si passa a gestire l'interrupt.

La periferica avvisa la CPU attraverso un segnale di interrupt proveniente da un bus di controllo. La CPU quando se ne accorge gestisce l'interrupt (se questo è in stato di ready, altrimenti lo ignora) e poi riprende la normale esecuzione.

In questo modo la CPU non fa più busy waiting ma rimane il fatto che un interrupt chiama il gestore delle eccezioni e fa

perdere tempo alla CPU.

DMA (Direct Memory Access)

Conserva gli interrupt ma evita, parola per parola, l'iterazione del trasferimento dei dati.

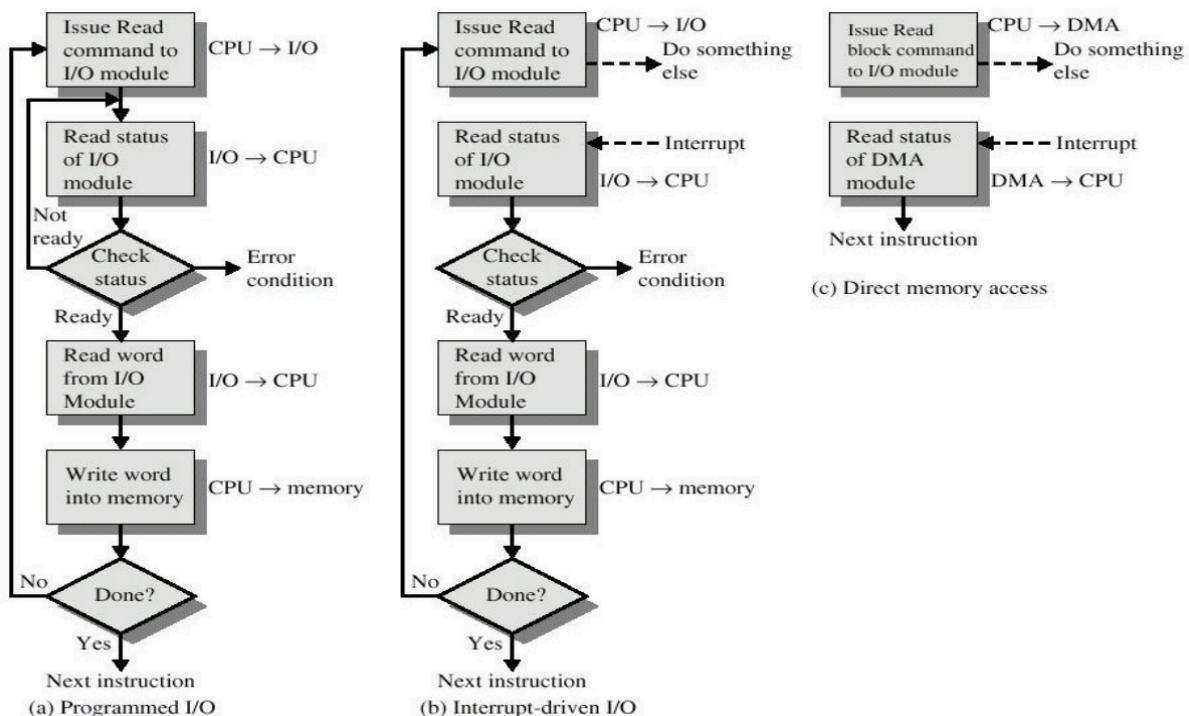
La CPU delega al DMA che carica direttamente i dati, quindi il processore è libero. Il DMA chiama il gestore delle eccezioni e si occupa della questione.

Quando tutto sarà finito gli HW si risincronizzeranno con un interrupt.

Il vantaggio è quando si trasferiscono grandi quantità di dati e si evita di sovraccaricare la CPU.

Lo svantaggio è avere HW in più. Necessità di due registri in più per ogni periferica. Si aggiungono in più un registro che indichi l'indirizzo di memoria da/dove trasferire i dati. Un registro che indichi la quantità dei dati da trasferire.

Inoltre il DMA ha un suo circuito all'interno della periferica.



Polling = interrogare costantemente il processore/periferica per verificare qualcosa.

PRESTAZIONI

Servono delle misure che permettano di valutare l'efficienza della gestione delle operazioni di I/O.

- Banda Passante → quantità di dati che si può trasferire per unità di tempo. (Byte al secondo)
- Latenza → tempo che intercorre tra quando la periferica è pronta e quando effettivamente il dato viene trasferito (secondi)

Tanta banda = buono

Poca Latenza = buono

Esercizi

Con riferimento ai principi di funzionamento dell'I/O visti a lezione, si indichi un motivo plausibile per cui la modalità di I/O in DMA è più efficiente rispetto all'I/O a controllo di programma e ad Interruzione:

- perché il DMA è migliaia di volte più veloce della CPU nel trasferire i dati tra Periferiche e Memoria
- perché, a differenza della CPU, il DMA non deve effettuare busy wait
- perché il DMA è migliaia di volte più veloce delle altre tecniche nel completare l'I/O
- perché, a differenza della CPU, il DMA effettua il polling della periferica molto più velocemente
- non rispondo

Perché con il DMA la CPU è impegnata solo per predisporre l'inizio e gestire il termine dell'I/O.

✓ 16) Con riferimento ai principi di funzionamento dell'I/O, il DMA richiede una circuiteria apposita, con capacità di accesso al bus e alle periferiche e per questo è tipicamente realizzata da:
Un circuito dedicato, concettualmente pensabile interno alla periferica.

✓ 17) Con riferimento ai principi di funzionamento dell'I/O, il processore dopo aver avviato il DMA:
È libero di proseguire l'esecuzione di programmi.

✓ 18) Con riferimento ai principi di funzionamento dell'I/O, il DMA richiede l'utilizzo di:
Più registri periferica rispetto al controllo di programma.

✓ 19) Con riferimento ai principi di funzionamento dell'I/O, nel caso di I/O Memory Mapped il bit di ready di una periferica, fisicamente è realizzato:
Nei circuiti di una periferica di I/O, ma appare come uno dei bit di una specifica locazione dello spazio di indirizzamento di memoria.

SPIM: il terminale ha due dispositivi distinti e indipendenti:
Receiver: legge i caratteri ASCII dalla tastiera
Transmitter:
visualizza caratteri ASCII sulla Console

Esercizio 1

Determinare la percentuale di tempo in cui viene effettivamente utilizzata la CPU per trasferire **1 parola da una tastiera** utilizzando la tecnica di **gestione a controllo di programma** se:

- la tastiera trasferisce **10 byte/s**
- frequenza di clock: **50 MHz**
- sono richiesti **20 cicli di clock per ogni byte**

Quindi sappiamo che **banda**: 10 byte/s, frqz clock: $50 * 10^6$ Hz
→ quanti clock fa la cpu al secondo
e per gestire 1 byte ci voglio 20 cicli di clock. Noi voglia trasferire 1 parola (4 byte).

La richiesta è di calcolare la % di CPU effettivamente utilizzata quindi:

cicli per gestire la parola / numero cicli su un certo tempo

Con numero cicli su un certo tempo intendo quanti cicli avvengono in un dato tempo:

numero cicli su un certo tempo = la frequenza di clock * tempo di trasferimento

Il tempo di trasferimento è quanto tempo ci mette la periferica a trasferire tot byte:

numero di byte / banda periferica = $\frac{4\text{byte}}{10\text{byte/sec}}$ = 0,4 sec

quindi numero cicli su un certo tempo: $50 * 10^6 \text{ Hz} * 0,4 \text{ sec} = 2 * 10^7 \text{ cicli}$

Calcoliamo quanti cicli ci voglio per gestire una parola:

cicli per gestire la parola = cicli di clock per gestire un byte * numero di byte = $20 * 4 \text{ byte} = 80 \text{ cicli}$

Infine prendiamo la formula iniziale e quella darà la percentuale di cpu utilizzata effettivamente:

p.s il tempo è compreso già nei calcoli quindi è come se calcolassi la percentuale di cpu utilizzata nel tempo che serve per trasferire la parola.

$$80 / 2 * 10^7 = 4 * 10^{-6} = 0,0004\%$$

Esercizio 2

Qual è la frazione del tempo del processore usata per la **gestione degli interrupt** necessari al trasferimento da/verso una periferica di I/O, se abbiamo:

- Frequenza di clock di **250 Mhz**
- Trasferimento di **4 Mbyte/s**, a blocchi di **4 word**
- Il costo di ogni interruzione è **500 cicli di clock**

Cosa dobbiamo trovare: % di tempo del processore usata per la gestione degli interrupt

Costo interrupt totale / frequenza di clock

Calcoliamo prima quanti interrupt possono avvenire in un secondo, sappiamo che il trasferimento avviene a blocchi di 4 word quindi 16 byte e sappiamo che si trasferiscono 4Mbyte/sec quindi:

$$\text{Interrupt al secondo} = \frac{\text{trasferimento}}{\text{dimensione in byte}}$$

$$4 * 10^6$$

$$/ 16 \text{ byte} = 250\text{K interrupt/sec}$$

Poi si calcola il costo di interruzioni totale, cioè quanti cicli ci metto per gestire tutti gli interrupt in un secondo.

$$\text{Costo interruzioni totale} = \text{interrupt al secondo} * \text{cicli di gestione per 1 interruzione}$$

$$250\text{K interrupt/sec} * 500 \text{ cicli/interrupt} = 125\text{K cicli/sec}$$

Ora possiamo calcolare la % di cpu: $125\text{M}/250\text{M} = 50\%$

Esercizio 3

Qual è la frazione del tempo del processore usata per la **gestione con DMA** di un trasferimento dati da periferica, se abbiamo:

- Frequenza di clock è **250 Mhz**
- Trasferimento di blocchi di **8 kbyte per ogni DMA**
- Trasferimento a **4 Mbyte/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**

L'esercizio 3 funziona allo stesso modo.

- Quanti DMA al sec: $\frac{\text{trasferimento}}{\text{dimensione}} = \frac{4 * 10^6}{8 * 10^3} \text{ byte/sec} = 500 \text{ dma/sec}$

- Costo (numeri cicli di clock cpu richiesti per gestire la dma) = DMA al secondo * costo gestione di un DMA = $500 \text{ dma/sec} * 1500 \text{ cicli} = 750.000 \text{ cicli/sec}$
- % utilizzo = costo dma totale / frequenza clock = $750 * 10^3 / 250 * 10^6 = 3 * 10^{-3} = 0,003 = 0,3\%$

Esercizio 4

Si supponga di voler trasferire un testo di 2 Mbyte e che

1. la CPU esegua 1 istruzione ogni ciclo di clock
2. il clock della CPU sia di 500 MHz ($500 * 10^6 \text{ Hz}$)
3. il device emetta al massimo 1 KB/s (1000 byte/s)
4. impostare i registri per DMA richiede 10 cicli di clock
5. eseguire alcune istruzioni di prologo e epilogo per quanto riguarda il controllo di programma sia trascurabile

Qual è il rapporto tra il tempo di impiego della CPU nel caso di DMA e di I/O a controllo di programma?

L'unica formula "nuova" è quella del rapporto tra tempo CPU in caso di DMA e caso di I/O a controllo di programma; che alla fine è una semplice frazione di rapporto.

$$\boxed{\text{Rapporto} = \frac{\text{tempo DMA}}{\text{tempo I/O}}}$$

Quindi calcoliamo il tempo in **caso di DMA**; sappiamo che tempo in cpu = **Costo DMA totali / frequenza di clock** in questo caso c'è un solo DMA e richiede 10 cicli di clock mentre la frequenza della CPU è di 500 MHz. $\rightarrow \frac{10}{500 * 10^6} = 0,02 * 10^{-6} = 2 * 10^{-8}$ secondi

Mentre per il tempo in

caso di I/O a controllo di programma (es1 tempo di trasferimento): **numero di byte / banda periferica** = $2 \text{ MByte}/1000 \text{ byte/sec} = 2.000.000/1000 = 2000 \text{ sec} = 2 * 10^3 \text{ sec}$
Non ci resta che fare

il rapporto del tempo: $\frac{2*10^{-8} \text{ sec}}{2*10^3 \text{ sec}} = 10^{-11}$ //se si fa il rapporto tra cicli cpu per dma e cicli cpu per I/O esce lo stesso risultato.

Esercizio 5

Determinare la percentuale di tempo in cui verrebbe effettivamente utilizzata la CPU per trasferire **1 parola** (con gestione di programma) da un floppy disk e da un hard disk, tenendo conto che sono richiesti **20 cicli di clock per ogni byte**, che la **frequenza di clock è 50 Mhz** e

1. Floppy disk opera a **50 Kbyte/s**
2. Hard-disk lavora a **2 Mbyte/s**

Semplice esercizio utilizzando le formule dell'esercizio uno.
Trasferiamo 1 parola = 4 byte e sono richiesti 20 cicli per ogni byte → tempo di cpu richiesto per trasferire una parola = 80 cicli

1. Per Floppy disk la banda periferica è 50 Kbyte/sec
formula % CPU

cicli per gestire la parola / numero cicli su un certo tempo

Con numero cicli su un certo tempo intendo quanti cicli avvengono in un dato tempo:

numero cicli su un certo tempo = la frequenza di clock * tempo di trasferimento

tempo di trasferimento = numero di byte /

banda periferica

$$\rightarrow 4 \text{ byte} / 50 \cdot 10^3 = 0,08 \cdot 10^{-3} \text{ sec}$$

$$\rightarrow 50 \cdot$$

$$10^6 \text{ byte/sec} \cdot 0,08 \cdot 10^{-3} \text{ sec} = 4 \cdot 10^3 \text{ cicli clock}$$

$$\rightarrow \% \text{cpu} = 80 / 4000 = 2\%$$

Idem per hard disk $\rightarrow \% \text{cpu} = 80\%$

Tabella Riassuntiva

Tutorial Per Domande del Tipo: Si consideri un processore MIPS32 con clock.....

Risultato		Riposta
Cpu > Periferica		Si riuscendo anche...
Cpu < Periferica		Non è possibile....
Cpu = Periferica		In linea teorica....

Esercizi della Tabella

1) Si consideri un processore come il MIPS32 con clock a 10MHz, in grado di eseguire tutte le istruzioni del suo ISA in 5 cicli di clock.

Questo processore ha una periferica che genera dati da 4byte alla frequenza di 1kHz. Il registro dati della periferica è pure a 32bit.

Supponendo che per arrivare all'istruzione che sposta il dato dalla periferica ci vogliono 120 istruzione dell'handler e che dopo il trasferimento ci vogliono altre 80 istruzioni dell'handler, si dica se è possibile gestire il trasferimento o se il processore perde dati.

$$f_{Clock} = 10\text{Mhz} = 10000000 \text{ hz} = 10\ 000\ 000 \text{ cicli/s}$$

1 istruzione = 5 cicli di clock

$$10\ 000\ 000 : X = 5 : 1$$

$$X = 10\ 000\ 000 / 5 = 2000000 \text{ istruzioni totali}$$

$$f_{Periferica} = 1\text{kHz} = 1000 \text{ cicli/s}$$

$$\text{istruzioniTotali per spostare dato} = (120+80) * 1000 = 200 * 1000 = 200000 \text{ istruzioni}$$

X > istruzioniTotali quindi il processore riesce a gestire il trasferimento

Esercizio 2:

2) Si consideri il processore come il MIPS32 con clock a 10 MHz, in grado di eseguire tutte le istruzioni del suo ISA in 5 cicli di clock.

Questo processore ha una periferica che genera dati da 4 byte alla frequenza di 10 kHz. Il registro dati della periferica è pure a 32 bit.

Supponendo che per arrivare all'istruzione che sposta il dato dalla periferica ci vogliono 120 istruzioni dell'handler e che dopo il trasferimento ci vogliono oltre 80 istruzioni dell'handler, si dica se è possibile gestire il trasferimento o se il processore perde dati.

$$f_{Clock} = 10\ 000\ 000 \text{ cicli/sec.}$$

1 istruzione = 5 cicli

$$\text{istruzioni totali per CPU} = 10\ 000\ 000 / 5 = 2\ 000\ 000$$

$$f_{Periferica} = 10\ 000 \text{ hz}$$

$$\text{istruzioni totali per trasferire il dato} = (120+80) * 10000 = 2\ 000\ 000 \text{ istruzioni}$$

2 000 000 = 2 000 000 → Con questi valori numerici, la risposta in linea teorica è sì, ma il programma che viene interrotto ad ogni richiesta di trasferimento non avanzerebbe

Esercizio 117:

117) Si consideri un processore come il MIPS32 con un clock a 10MHz, in grado di eseguire tutte le istruzioni del suo ISA in 5 cicli di clock. Questo processore ha una periferica che genera dati da 4byte alla frequenza di 100kHz. Il registro dati della periferica è pure a 32bit. Supponendo che per arrivare all'istruzione che sposta il dato dalla periferica ci vogliono 120 istruzioni dell'handler e che dopo il trasferimento ci vogliono altre 80 istruzioni dell'handler, si dica se è possibile gestire il trasferimento o se il processore perde dati:

$$f_{Clock} = 10\text{ MHz} = 10\,000\,000 \text{ Hz} = 10\,000\,000 \text{ cicli/sec}$$

$$1 \text{ istruzione} = 5 \text{ cicli di clock}$$

$$10\,000\,000 : X = 5 : 1$$

$$X = 10\,000\,000 / 5 = 2\,000\,000 \text{ di istruzioni totali possibili dal processore}$$

$$f_{Periferica} = 100\text{ kHz} = 100\,000 \text{ Hz} = 100\,000 \text{ cicli/sec}$$

$$\text{totIstruzioni per trasferire il dato} = (120 + 80) * 100\,000 = 20\,000\,000 \text{ istruzioni da fare}$$

X < totIstruzioni allora non è possibile gestire il trasferimento perché il processore perderebbe dati

Una CPU MIPS esegue 1 istruzione ogni ciclo di clock

- Il clock della CPU è di 500 MHz // $500 * 10^6$ hertz oppure $5 * 10^8$ hertz

- Il device emette al massimo 1 KB/s (1000 byte/s)

- Impostare i registri del DMA occupa 10 cicli di clock

- Prologo/epilogo del controllo di programma trascurabili

Si vuole trasmettere un testo di 2Mb // $2 * 10^6$ byte

Qual è il rapporto dei tempi di impiego della CPU nel caso di DMA e di I/O a controllo di programma?

Controllo di programma:

$$\begin{aligned} \text{Tempo trasferimento} &= \text{byte da trasferire} / \text{Banda} \\ &= (2 * 10^6 \text{ byte/s}) / 1000 \text{ byte} = 2 * 10^3 \text{ secondi} \end{aligned}$$

DMA: la CPU è usata solo per i 10 cicli di clock dell'impostazione dei registri

$$\begin{aligned} \text{Tempo per ciclo} &= 1 / \text{Frequenza} \\ &= 1 / (5 * 10^8) \end{aligned}$$

$$\begin{aligned} \text{Tempo totale del trasferimento} &= (\text{cicli di clock per impostare i registri del DMA}) * \\ &\quad \text{tempo per ciclo} \end{aligned}$$

$$10 * [1 / (5 * 10^8)] = 1 / (5 * 10^7) = 1/5 * 1/10^7 = 0,2 * 10^{-7} = (2 * 10^{-1}) * 10^{-7} = 2 * 10^{-8}$$

Poi faccio un rapporto dei tempi del controllo di programma con quelli del DMA, ossia:

$$(2 * 10^3) / (2 * 10^{-8}) = 10^3 * 10^8 = 10^{11}$$