

La Gestione dei Dispositivi di Input/Output

Matteo Sonza Reorda

Politecnico di Torino
Dip. di Automatica e Informatica

1

M. Sonza Reorda - a.a. 2001/2002

Sommario

- **Introduzione**
- **I/O programmato**
- **Interrupt**
- **DMA**

2

M. Sonza Reorda - a.a. 2001/2002

Introduzione

Una delle principali funzioni svolte dai sistemi di elaborazione è quella di interagire e di scambiare informazioni con il mondo esterno.

Tale funzione viene svolta attraverso il sotto-sistema di I/O.

Sotto-sistema di I/O

Il sotto-sistema di I/O di un calcolatore è composto di:

- dispositivi di Input/Output (periferiche)
- unità per il controllo di questi dispositivi
- software per la loro gestione.

Dispositivi di I/O

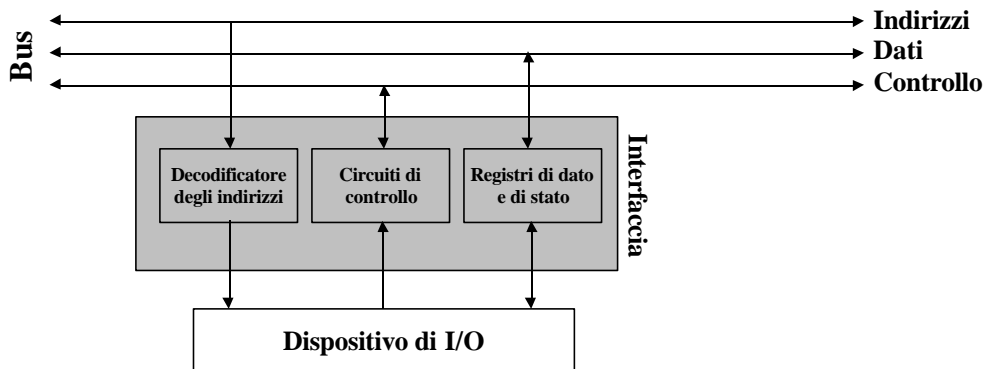
Oltre a quelli ben noti (tastiera, mouse, video, stampante, etc.) vanno ricordati quelli utilizzati in ambiente industriale, quali *sensori* ed *attuatori* di vario tipo.

5

M. Sonza Reorda - a.a. 2001/2002

Indirizzamento dei Dispositivi di I/O

A ciascun dispositivo di I/O è di solito associata un'interfaccia, che lo collega al bus del sistema.



6

M. Sonza Reorda - a.a. 2001/2002

Porte

La comunicazione tra il processore (o la memoria) ed il dispositivo avviene attraverso i registri (o *porte*).

Questi sono accessibili tramite il bus di sistema, al pari delle celle di memoria, in quanto a ciascuno di essi è associato un indirizzo.

La loro connessione al bus può avvenire secondo 2 modalità:

- *memory-mapped I/O*
- *isolated I/O*, o *I/O-mapped I/O*

Memory-Mapped I/O

I registri dei dispositivi di I/O sono connessi come le normali celle di memoria.

Lo spazio di indirizzamento per la memoria è quindi ridotto.

Si può fare accesso ai registri delle periferiche utilizzando tutte le istruzioni ed i modi di indirizzamento utilizzabili per accedere alla memoria.

È la soluzione adottata dal 68000.

Isolated I/O

Gli spazi di indirizzamento per la memoria e per le porte di I/O sono separati, e sono attivati alternativamente da appositi segnali (ad esempio IO/M nell'8086).

Per accedere alle porte di I/O si devono utilizzare apposite istruzioni (IN e OUT).

Istruzioni di I/O

- **IN:** lettura da una porta
- **OUT:** scrittura su una porta.

Sincronizzazione

Le velocità di operazione della CPU e dei vari dispositivi periferici possono essere molto diverse.

Il processore deve quindi sincronizzarsi con i dispositivi di I/O per eseguire le operazioni che li interessano.

La sincronizzazione può venire svolta in vari modi.

Meccanismi di Gestione dell'I/O

Si differenziano per il diverso livello di coinvolgimento della CPU nella gestione dei dispositivi di I/O.

Possibili soluzioni sono:

- **I/O programmato**
- **Interrupt**
- **Direct Memory Access (DMA)**
- **Processori di I/O.**

I/O Programmato

In questo caso la gestione dei dispositivi di I/O è totalmente demandata alla CPU.

Ogni dato viene prima trasferito dal buffer associato alla periferica ad un registro interno della CPU, e poi immagazzinato in memoria (o viceversa).

Lo spostamento di ciascun dato implica l'esecuzione di almeno un'istruzione da parte della CPU.

Quando l'I/O programmato è basato sulla ripetizione di un test sul registro di stato per verificare quando il programma può procedere oltre, si parla di *polling*.

Caratteristiche

L'I/O programmato viene normalmente utilizzato nei sistemi più piccoli e meno complessi, in quanto ha le seguenti caratteristiche:

- + poco costoso in termini di hardware
- poco efficiente.

Limiti dell'I/O programmato

Ogni dispositivo deve dipendere dalla CPU per essere servito. Ne consegue che:

- l'efficienza (in termini di uso del dispositivo) dipende dalla frequenza con cui il test viene ripetuto
- tutti i dati devono passare attraverso la CPU, e non esiste connessione diretta tra dispositivo e memoria
- la CPU dedica una parte del suo tempo ad eseguire banali operazioni di test e trasferimento dati.

Interrupt

È basato su un segnale asincrono che il dispositivo invia alla CPU quando ha bisogno di un servizio.

In questo modo:

- il tempo per ottenere l'attenzione della CPU si riduce
- la CPU non perde tempo a scandire in polling il dispositivo per testarne lo stato.

Interrupt

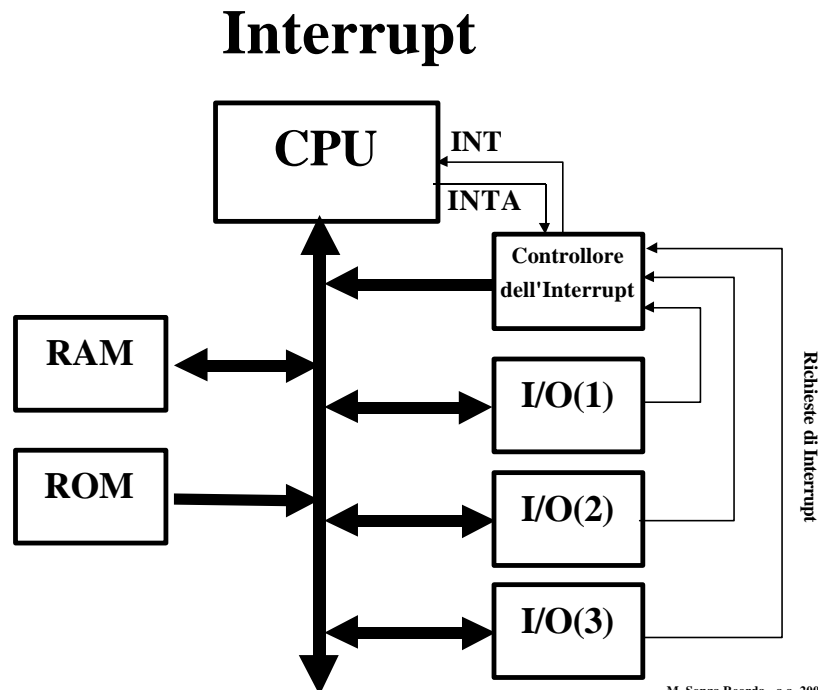
La gestione del servizio dei dispositivi di I/O tramite *interrupt* permette un migliore efficienza dell'esecuzione della elaborazione della CPU.

Un input esterno *asincrono* informa il microprocessore che un dispositivo esterno richiede di essere servito. La CPU interrompe l'esecuzione del programma corrente e salta all'esecuzione di una *procedura di servizio* dell'interruzione (*Interrupt Service Routine - ISR*).

Il dispositivo di controllo dell'interrupt funziona come un gestore delle richieste di interruzione tra i dispositivi periferici e la CPU.

17

M. Sonza Reorda - a.a. 2001/2002



18

M. Sonza Reorda - a.a. 2001/2002

Procedure di Servizio dell'Interrupt

Differiscono dalle normali procedure in quanto, quando vengono attivate, interrompono un programma a priori non noto.

Di conseguenza, al loro termine tale programma deve poter continuare la propria attività come se non fosse stato interrotto.

Per questo all'attivazione di una procedura di servizio dell'interrupt vengono salvati nello stack:

- il PC e la parola di stato (via hardware)
- eventuali registri o parole di memoria utilizzate (via software).

Latenza di Interrupt

È il tempo tra la richiesta di interrupt e la partenza della relativa procedura di servizio.

In talune applicazioni (ad esempio quelle per le elaborazioni in tempo reale) deve essere estremamente ridotta.

Disabilitazione degli Interrupt

È necessario evitare che la stessa richiesta di interrupt scateni due volte l'attivazione della procedura di servizio.

La soluzione di solito prevede che

- vi sia un apposito bit di abilitazione degli interrupt nel registro di stato del processore
- il bit venga automaticamente settato quando il processore entra nella procedura di servizio
- il bit venga nuovamente resettato al termine della procedura di servizio.

Identificazione del dispositivo

Quando più di un dispositivo può richiedere un interrupt, è necessario un meccanismo per l'individuazione del dispositivo la cui procedura di servizio deve essere attivata per prima.

Esistono varie soluzioni:

- linee di Interrupt multiple
- Polling
- Daisy Chain
- Arbitraggio del Bus.

Linee di Interrupt Multiple

La CPU possiede diversi piedini per le richieste di Interrupt. Ogni periferica è collegata ad un diverso piedino.

È una soluzione impraticabile, poichè il numero di periferiche è quasi sempre superiore a quello dei piedini disponibili.

Polling

Esiste un solo piedino per le richieste di Interrupt.

Quando la CPU percepisce la richiesta, inizia a scandire le parole di stato di tutte le periferiche per individuare quella che ha fatto la richiesta.

Interrupt Vettorizzati

- 1 Quando il processore è pronto a servire la richiesta di interrupt, invia un segnale di *Interrupt Acknowledge*.
- 2 Il dispositivo che ha fatto la richiesta pone sul bus un codice di identificazione.
- 3 Il codice viene utilizzato dal processore per determinare l'indirizzo della procedura di servizio, usandolo come indice per accedere ad un vettore degli indirizzi delle procedure di servizio.

Priorità dei dispositivi

Nei sistemi complessi è spesso necessario assegnare a ciascun dispositivo una priorità, e fare in modo che la procedura di servizio di un interrupt sia interrompibile solo da un dispositivo con priorità maggiore.

In alcuni casi le priorità assegnate ai dispositivi cambiano nel tempo.

La priorità della procedura in corso di esecuzione può essere scritta nella parola di stato.

In altri casi è il controllore degli interrupt che gestisce le priorità.

Richieste simultanee

Nel caso di richieste di interrupt simultanee, è necessario che venga servito prima il dispositivo con priorità maggiore.

Se si usa il meccanismo delle linee di interrupt multiple, è il processore che risolve il problema.

Nel caso del polling, il processore serve prima il dispositivo che compare prima nella sequenza con cui i dispositivi vengono interrogati.

Nel caso degli interrupt vettorizzati, si usa di solito il meccanismo del *Daisy Chaining*.

27

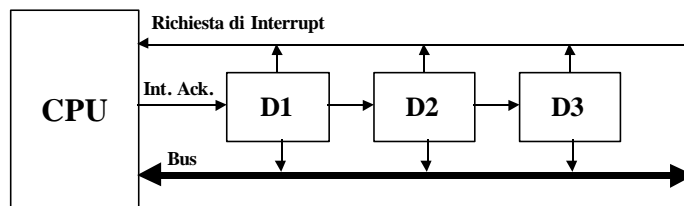
M. Sonza Reorda - a.a. 2001/2002

Daisy Chaining

Ciascun dispositivo possiede una circuiteria che riceve il segnale di Interrupt Acknowledge.

Quando questo viene attivato, se il dispositivo ha fatto richiesta di interrupt mette sul bus il suo codice.

Se non ha fatto richiesta, attiva il segnale di Interrupt Acknowledge in uscita.



28

M. Sonza Reorda - a.a. 2001/2002

Eccezioni

Sono gli eventi in grado di interrompere l'esecuzione del programma corrente.

Esistono vari tipi di eccezioni:

- interrupt di I/O
- segnali di errore
- segnali per il debug
- eccezioni di privilegio.

Ciascuno di questi segnali attiva opportune procedure di servizio.

Segnali di errore

Possono essere generati

- dai circuiti di controllo delle memorie, ad esempio quando si verifica un errore di parità
- dal circuito di decodifica delle istruzioni, quando il codice operativo dell'istruzione di cui si è fatto il fetch non corrisponde a nessuna delle istruzioni lecite
- dai circuiti aritmetici, quando si verifica una condizione di errore (ad esempio divisione per 0).

Segnali per il debug

Sono utilizzati dai *debugger*.

Molti processori prevedono una modalità *trace*; quando viene attivata il processore scatena un'eccezione dopo l'esecuzione di ciascuna istruzione.

Tale eccezione viene utilizzata per gestire il modo di esecuzione *single step*.

Inoltre, i processori dispongono spesso di apposite istruzioni che scatenano un'eccezione voluta.

Queste possono venire utilizzate per implementare i breakpoint.

Eccezioni di privilegio

Se il processore prevede più modi di funzionamento in base alla priorità del programma in esecuzione (ad esempio *utente* o *supervisore*), viene attivata un'eccezione quando si tenta di eseguire un'istruzione non permessa dal livello di priorità corrente.

Analogamente, si genera una simile eccezione quando si tenta di fare accesso ad un'area di memoria senza averne i diritti.

L'interrupt nel 68000

- 8 livelli di priorità degli interrupt, codificati su 3 bit
- il livello corrente è memorizzato nel registro di stato
- una richiesta di interrupt è accettata solo se possiede priorità superiore al livello corrente
- le richieste di livello 7 vengono sempre accettate (*interrupt non mascherabile*)
- quando viene accettata una richiesta di interrupt, nello stack vengono salvato l'indirizzo di ritorno ed il registro di stato
- l'istruzione RTE (Return from Exception) termina una procedura di servizio, ripristinando il registro di stato.

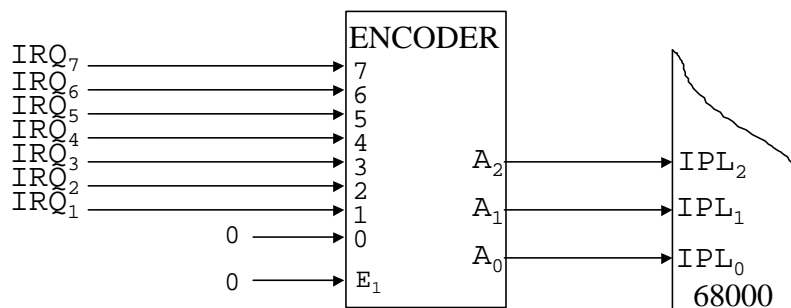
33

M. Sonza Reorda - a.a. 2001/2002

68000: segnali di interrupt

Il 68000 possiede 3 segnali per le richieste di interrupt. Il valore su questi tre segnali determina la priorità della richiesta.

Ciascun livello può corrispondere a più dispositivi, gestiti in daisy chain.



34

M. Sonza Reorda - a.a. 2001/2002

68000: riconoscimento del dispositivo

Può avvenire in due modi:

- in modo *vettorizzato*: il dispositivo mette sul bus un codice da 8 bit, con il quale il processore accede ad una tabella degli interrupt, dove sono memorizzati gli indirizzi di partenza delle procedure di servizio
- in modo *autovettorizzato*: attraverso un apposito segnale, il dispositivo segnala che non metterà sul bus il suo codice; in tal caso il codice dell'interrupt è automaticamente associato al livello della richiesta (IRQ1=25, IRQ2=26, etc.).

L'interrupt nel PowerPC

- 1 sola linea per le richieste di interrupt
- bit per l'abilitazione/disabilitazione degli interrupt nel registro di stato (Machine State Register, MSR)
- indirizzo di ritorno e registro di stato salvati automaticamente in due registri (SRR0 e SRR1)
- tabella contenente gli indirizzi di partenza delle procedure di servizio.

Uso dell'Interrupt

I Sistemi Operativi sfruttano pesantemente il meccanismo dell'Interrupt:

- **le procedure che compongono il Sistema Operativo (ad esempio per la gestione delle periferiche) sono attivate**
 - tramite interrupt software dai programmi
 - tramite interrupt hardware dalle periferiche stesse
- **il passaggio all'esecuzione delle procedure di interrupt corrisponde spesso al passaggio ad una modalità speciale (supervisor).**

DMA

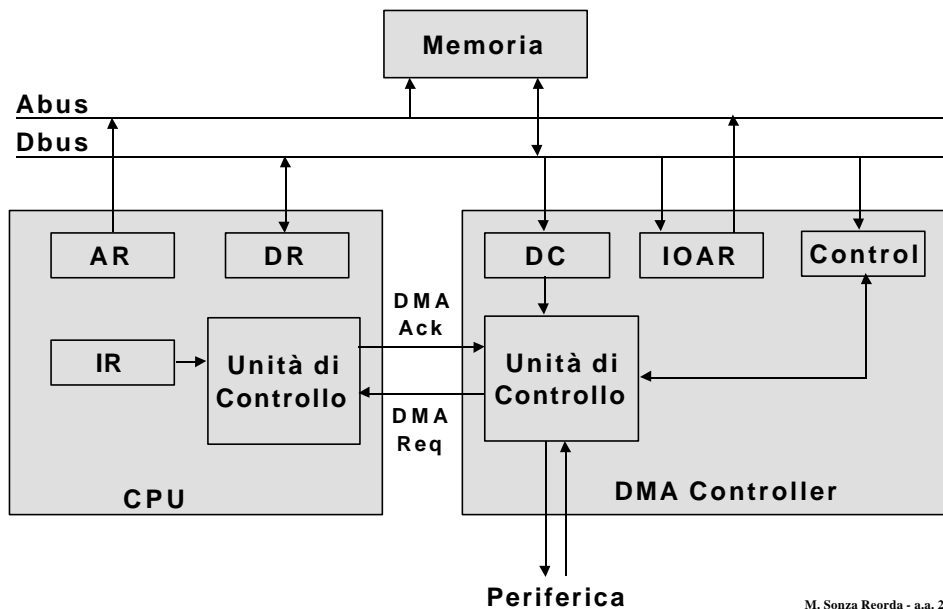
È il metodo preferito quando si devono trasferire grosse moli di dati.

Una circuiteria apposita (*DMA Controller*) provvede ad eseguire il trasferimento di dati da una periferica alla memoria (o viceversa).

La circuiteria deve essere in grado di fungere da *bus master*, ossia deve generare gli indirizzi ed i segnali di controllo secondo la tempistica opportuna.

Il DMA Controller deve inoltre essere in grado di negoziare con la CPU l'acquisizione del controllo del bus ed il suo rilascio.

Circuiteria per il DMA



39

M. Sonza Reorda - a.a. 2001/2002

Canali di DMA

Ciascun DMA controller può normalmente gestire più periferiche (*canali*). Per ciascuna periferica il DMA controller dispone di

- Linee di richiesta e di acknowledge
- Registri DC e IOAR
- Circuiteria di controllo.

In tal caso il DMA controller è anche in grado di arbitrare tra richieste di DMA contemporanee.

40

M. Sonza Reorda - a.a. 2001/2002

Trasferimento in DMA: svolgimento

Il trasferimento di un blocco in DMA si articola in varie fasi:

- la CPU carica nei registri IOAR e DC l'indirizzo dell'area di memoria ed il numero di parole da trasferire; la CPU informa inoltre il DMA controller della direzione del trasferimento (memoria ® periferica o viceversa);
- il DMA Controller riceve una richiesta di trasferimento da parte di una periferica;
- Il DMA Controller invia un segnale di DMA Request alla CPU;
- quando la CPU giunge ad un punto di rilevamento del segnale di DMA Request, rilascia il bus e attiva il segnale di DMA Acknowledge;
- il DMA Controller inizia il trasferimento; dopo il trasferimento di ciascuna parola, IOAR e DC vengono aggiornati;

41

M. Sonza Reorda - a.a. 2001/2002

Trasferimento in DMA: svolgimento (II)

- il DMA Controller può sospendere temporaneamente il trasferimento (ad esempio perchè la periferica non ha più dati da trasferire) disattivando DMA Request; la CPU disattiva DMA Acknowledge, e riprende il controllo del bus;
- quando DC giunge a zero, il trasferimento termina;
- il DMA Controller invia un Interrupt alla CPU.

42

M. Sonza Reorda - a.a. 2001/2002

DMA: modi di funzionamento

Il trasferimento dei dati in DMA può avvenire in vari modi:

- trasferimento a blocchi (*burst transfer*)
- trasferimento con *cycle stealing*
- trasferimento in *transparent DMA*.

Trasferimento a blocchi

Prevede che il DMA Controller, una volta acquisito il controllo del bus, lo mantenga per tutto il tempo richiesto per trasferire un blocco di dati.

In tal modo

- il trasferimento avviene alla massima velocità

ma

- la CPU è bloccata per tutta la durata del trasferimento.

Il trasferimento a blocchi è importante per periferiche quali i dischi magnetici, dove il trasferimento del blocco non può essere interrotto).

Trasferimento con *Cycle Stealing*

Il DMA Controller trasferisce i dati in piccoli blocchi, occupando il bus per periodi limitati di tempo.

In tal modo:

- la velocità di trasferimento è minore

ma

- la CPU non è bloccata per periodi troppo lunghi.

Trasferimento in *Transparent DMA*

Il DMA Controller è in grado di rilevare quando la CPU non utilizza il bus, e solo in quei periodi esegue il trasferimento dei dati.

In tal modo la CPU non è praticamente rallentata dal DMA Controller.

Arbitraggio del Bus

Il DMA Controller è uno dei dispositivi che possono divenire Master del bus.

Un dispositivo può essere master del bus se è in grado di decidere l'operazione che deve essere svolta sul bus (lettura, scrittura, etc.).

Se esistono diversi DMA Controller, è necessario aggiungere la circuiteria per la gestione dei conflitti.

DMA Chaining

Il DMA Controller della famiglia 68000 supporta un meccanismo denominato *DMA Chaining*.

Quando il DMA Controller termina il trasferimento di un blocco, accede ad una particolare zona di memoria, ove è memorizzata una coppia di valori, che vengono caricati nei registri DC e IOAR.

Il DMA Controller procede quindi al trasferimento di un nuovo blocco di dati.

In tal modo la preparazione delle nuove parole di controllo per il DMA Controller da parte della CPU può avvenire prima che questo termini il proprio lavoro.