

Corso di Calcolatori Elettronici I A.A. 2010-2011

Interruzioni parte prima

Lezione 32

Prof. Roberto Canonico



Università degli Studi di Napoli Federico II
Facoltà di Ingegneria
Corso di Laurea in Ingegneria Informatica (allievi A-DE+Q-Z)
Corso di Laurea in Ingegneria dell'Automazione

Inquadramento del problema

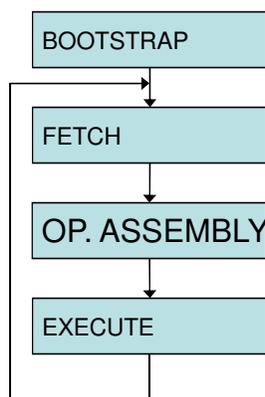
- Sistema delle interruzioni:
 - Infrastruttura per la gestione di specifici eventi
 - Primo grado di libertà:
 - Esistono molti processori, con diverse organizzazioni
 - Secondo grado di libertà:
 - Esistono molti circuiti hardware dedicati, con diverse caratteristiche, che possono collaborare con il processore nella gestione delle interruzioni
 - Problema:
 - Fare riferimento ad un modello GENERALE
 - Evitare che tale modello sia GENERICO
 - ⇒ Ove necessario, sarà preso in esame un processore reale (68000)
-

Le Interruzioni

Nello I/O Controllato da programma il processore passa tutto il tempo ad aspettare una periferica (tipicamente più lenta).

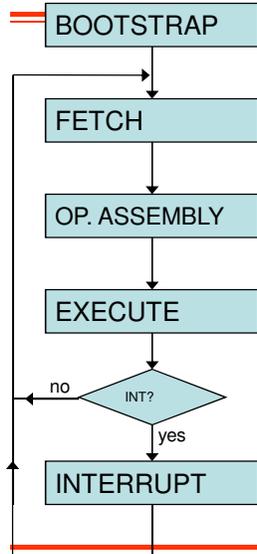
Una interruzione è un segnale mandato direttamente dalla periferica al processore per interromperne l'esecuzione e lanciare l'esecuzione di un programma differente.

Il ciclo del processore semplificato



- Se il ciclo del processore fosse effettivamente quello mostrato in figura, sorgerebbero alcuni problemi, come per esempio:
 - un'applicazione "prepotente" potrebbe impadronirsi della risorsa processore senza mai lasciarla
 - non ci sarebbe modo di rimuovere forzatamente un'applicazione che entri per errore in un ciclo infinito
 - il sistema operativo, in generale, avrebbe un controllo limitato sul sistema

Il ciclo del processore esteso al meccanismo delle interruzioni



- La soluzione comunemente adottata consiste nel permettere al “supervisore” di prendere il controllo del processore al termine di ciascun ciclo
- Questo avviene esclusivamente nel caso in cui si verificano eventi “eccezionali”, di solito *asincroni* con l’esecuzione del programma correntemente in corso
- In assenza di tali eventi l’elaborazione procede nella maniera consueta

Innesco

```

while (true) {
    Fetch();
    Decode();
    Execute();
    CheckForInterrupt();
}
  
```

Se non succede niente di “eccezionale”, non fa praticamente niente

Eventi Eccezionali

- Reset
 - Riporta la macchina in uno stato iniziale noto
 - È generato da condizioni d'errore non recuperabili
 - Traps
 - Forniscono un meccanismo controllato ed efficiente di passaggio allo stato supervisore
 - Sono eventi sincroni (rispetto all'elaborazione)
 - Interrupts
 - Permettono di gestire richieste di "attenzione" da parte di dispositivi (tipicamente di I/O)
 - Sono eventi asincroni (rispetto all'elaborazione)
-

Elementi Fondamentali

- Segnale di Interrupt (INT)
 - Segnale di interruzione per il processore
 - Segnale di Ack
 - Segnale di riscontro dell'interruzione
 - ISR (Procedura di Servizio degli Interrupt)
 - Procedura lanciata quando giunge l'interrupt
-

La fase INTERRUPT Hardware (1/2)

INTERRUPT

- La fase di INTERRUPT viene attivata nel caso in cui il segnale INT è asserito
- Questo evento è sintomatico del fatto che alcuni eventi sono “pendenti” e devono essere “serviti”
- Gli eventi possono essere di natura diversa e possono essere generati da diverse cause

La fase INTERRUPT hardware (2/2)

INTERRUPT

- Durante questa fase del ciclo del processore, comunque, non viene eseguito un programma
- Per eseguire un programma (*software*), infatti, sarebbe necessario trovarsi all'interno del ciclo principale e muoversi tra le fasi di *fetch* ed *execute*
- Ciò che avviene nella fase di *interrupt* consiste invece in una serie di meccanismi *hardware* che “preparano” il processore a gestire l'interruzione

Sequenza di eventi durante un Interrupt (dall'esterno)

- Il Dispositivo genera il segnale;
 - il Processore interrompe il programma in esecuzione;
 - il Dispositivo viene informato che l'interrupt è stato ricevuto (ack);
 - viene eseguita la procedura (ISR);
 - si ripristina il Programma originale.
-

Sequenza di eventi durante un Interrupt (dall'interno)

- Esecuzione normale
 - Servizio dell'interruzione
 - Salvataggio del contesto (hardware)
 - Identificazione del device
 - Salto all'entry point della Interrupt Service Routine (ISR)
 - Salvataggio del contesto (software)
 - Servizio dell'interruzione
 - Ripristino del contesto
 - Esecuzione normale
-

Il ripristino del programma

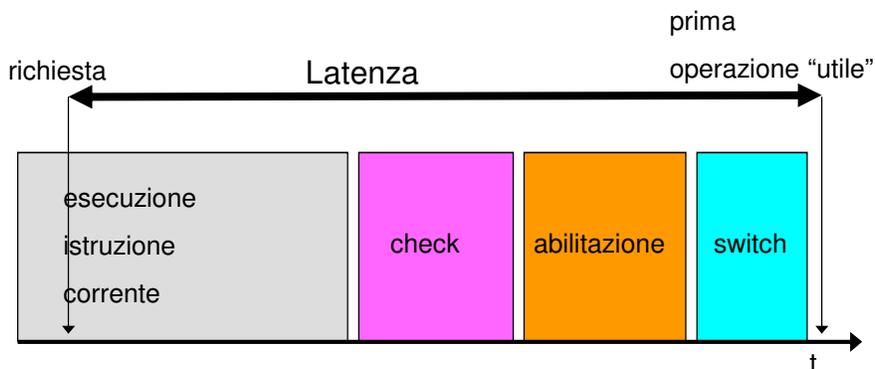
- Un' interruzione potrebbe eseguire un'elaborazione *B* completamente indipendente da quella *A* correntemente in corso sul processore, interrompendola
 - La gestione delle interruzioni deve quindi anche provvedere a mettere *A* in condizioni di continuare successivamente senza "accorgersi" di nulla
 - Sorge la necessità di salvare (prima) e ripristinare (dopo) lo stato del programma che viene di volta in volta interrotto
 - In questo modo *A* può continuare la sua elaborazione senza risentire in alcun modo del servizio dell'interruzione (a parte il ritardo dovuto al servizio dell'interruzione)
 - Le informazioni che devono essere salvate e ripristinate comprendono di solito il PC, i flag dei codici di condizione e il contenuto di qualsiasi registro che sia usato sia dal programma che dalla routine di gestione dell'interruzione
-

Il salvataggio dello stato

- L'operazione di salvataggio può essere svolta in parte o completamente in *hardware* o in *software*
 - Un'esigenza comune resta comunque quella di salvare lo stretto indispensabile poiché il salvataggio richiede trasferimenti di dati, eventualmente da e verso la memoria
 - Data la frequenza con cui le interruzioni vengono prodotte, questo rappresenta quindi un carico aggiuntivo che deve essere ridotto al minimo
 - Il salvataggio dello stato incrementa il ritardo tra l'istante di ricezione della richiesta di interruzione e l'istante in cui inizia l'esecuzione della routine di interrupt
 - Questo tempo viene detto *latenza di interrupt*
-

Latenza di un'interruzione

- È il tempo massimo che intercorre tra la richiesta di attenzione e l'effettivo servizio dell'interruzione



Problemi da Affrontare

- Le Priorità
- Interrupt Inneitati
- Presenza di più dispositivi

Le priorità

La maschera delle interruzioni è un vettore di bit che indica il livello di priorità dell'interruzione attualmente in esecuzione (0 nessuna interruzione)

- Ciascun Interrupt viene associato ad un livello di priorità.
 - Tutti gli interrupt che giungono con priorità più bassa non vengono eseguiti.
 - Nella maschera degli interrupt si memorizza il livello di priorità attuale.
 - Spesso esiste un livello di priorità non mascherabile
-

Interrupt Innestati

- Il Programma sta eseguendo la propria operazione
 - Giunge un interrupt e viene fatta partire la ISR
 - Cosa avviene se giunge un nuovo interrupt?
-

Interrupt Inneitati

- Soluzione 1:
 - Non controllo la linea Interrupt fino a quando non ho terminato la procedura attuale
 - Soluzione 2:
 - Memorizzo nello SR che sto eseguendo un Interrupt, la maschera degli interrupt mi dice quali possono avvenire
 - Soluzione 3:
 - Controllo la linea degli Interrupt solo sul fronte di salita.
-

Interrupt Inneitati (gestione con priorità)

- Il Dispositivo genera il segnale;
 - il Processore interrompe il programma in esecuzione;
 - **gli Interrupt di priorità più bassa vengono disabilitati;**
 - registri, PC e SR vengono salvati nello Stack;
 - il Dispositivo viene informato che l'interrupt è stato ricevuto (ack);
 - viene eseguita la procedura (ISR);
 - si ripristina il Programma originale.
-

Identificazione dei dispositivi

(1/3)

- Se ci sono più dispositivi, il processore deve essere in grado di identificare il dispositivo che ha generato l'interruzione, poiché probabilmente diverse azioni dovranno essere intraprese a seconda del particolare dispositivo
- I dispositivi hanno una linea comune attraverso la quale segnalano richieste di interruzioni (INT)
- Quando INT è alto si pone il problema di identificare da quale dispositivo è partita la richiesta

Identificazione dei dispositivi

(2/3)

- INT potrebbe alzarsi anche in seguito a richieste "contemporanee" di due o più dispositivi
- Esistono diverse soluzioni a questo problema
- Tutte le soluzioni impiegano un misto di hardware e di software
- Tutte le soluzioni dipendono fortemente sia dall'architettura del sistema che da quella del processore

Identificazione dei dispositivi (3/3)

- Polling
 - Si interroga il registro di stato di tutti i dispositivi.
 - Interrupt vettorizzato
 - Il Dispositivo manda un identificativo in risposta al segnale di ack.
 - Interrupt autovettorizzato
 - L'indice viene associato direttamente al segnale di interruzione
-

La soluzione a registri di stato: polling

- Una possibile soluzione consiste nel dotare ogni dispositivo di un registro di stato
 - Quando un dispositivo richiede un'interruzione, inizializza un bit nel registro di stato, il bit di richiesta di interrupt (Interrupt Request, IRQ)
 - La procedura di servizio inizia interrogando tutti i dispositivi in un certo ordine e, non appena trova un bit alto, fa partire la corrispondente routine di interrupt
 - Questa interrogazione ciclica (polling) è semplice da realizzare, ma ha lo svantaggio di richiedere un certo tempo per interrogare anche i dispositivi che non hanno invocato alcun servizio
-

Il Vettore delle Interruzioni

- Identificare un dispositivo corrisponde anche a scegliere la corretta ISR da eseguire
 - Per distinguere le differenti ISR si utilizza il “Vettore delle Interruzioni”
 - Il Vettore delle Interruzioni è un array contenente indirizzi di memoria
 - Gli indirizzi corrispondono alla locazione di memoria dove si trova la ISR da eseguire
-

Interrupt vettorizzato

- Si manda il segnale di Interrupt
 - Il processore risponde con un ack
 - Il dispositivo manda un codice di interruzione
 - Il codice è un indice nel “Vettore delle Interruzioni”
 - Il contenuto del “vettore delle interruzioni” è l’indirizzo di memoria dove si trova la procedura che gestisce l’interruzione al codice fornito.
-

Interrupt autovettorizzato

- Si manda il segnale di Interrupt
 - In base alla priorità del segnale di interruzione si accede ad una locazione di memoria prefissata contenente l'indirizzo della ISR.
 - Dal vettore delle interruzioni si ricava quindi l'indirizzo di memoria della corretta ISR da eseguire
-