Corso di Calcolatori Elettronici I

Interruzioni

Prof. Roberto Canonico

Università degli Studi di Napoli Federico II
Dipartimento di Ingegneria Elettrica e
delle Tecnologie dell'Informazione
Corso di Laurea in Ingegneria Informatica
Corso di Laurea in Ingegneria dell'Automazione

Inquadramento del problema

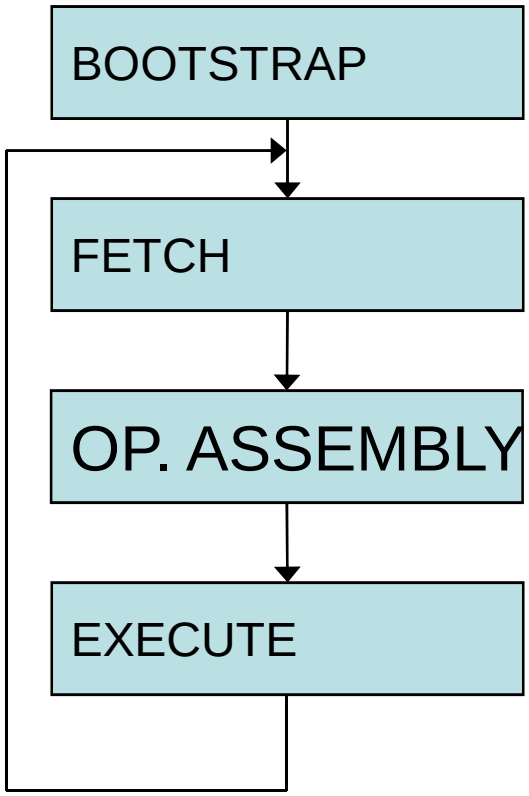
- Sistema delle interruzioni:
  - Infrastruttura per la gestione di specifici eventi
- Primo grado di libertà:
  - Esistono molti processori, con diverse organizzazioni
- Secondo grado di libertà:
  - Esistono molti circuiti hardware dedicati, con diverse caratteristiche, che possono collaborare con 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).

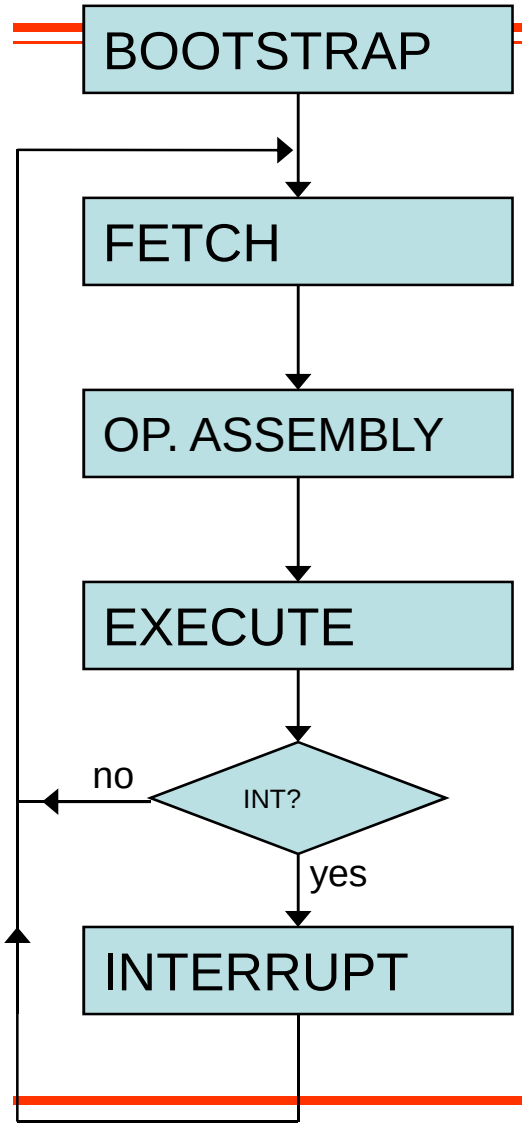
Una interruzione è un segnale mandato direttamente dalla periferica al processore per interromperne l'esecuzione e lanciare l'esecuzione di un programma differente.

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

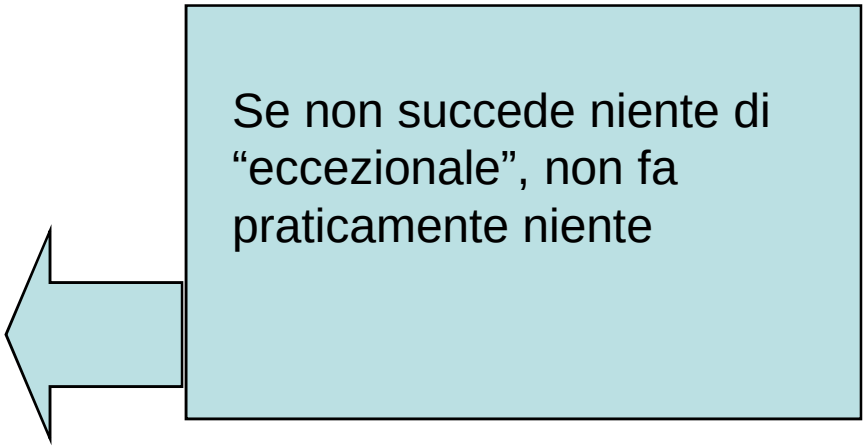
Il ciclo del processore esteso al meccanismo delle interruzioni



- 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 verifichino 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()
}
```



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 (software)
  - Ripristino del contesto (hardware)
- Esecuzione normale

Il ripristino del programma

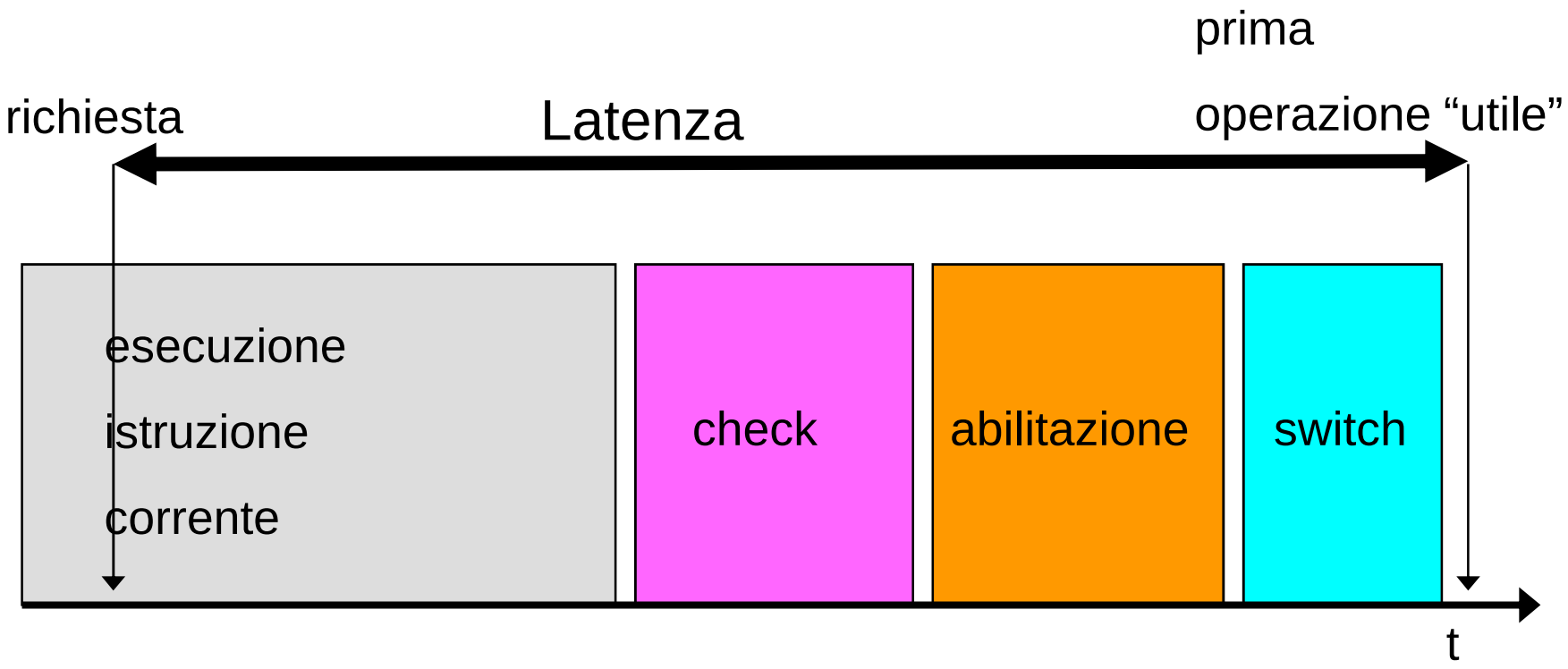
- Un' interruzione potrebbe eseguire un'elaborazione *B* completamente indipendente da quella *A* correntemente in corso sul processore, interrompendola
  - La gestione delle interruzioni deve quindi anche provvedere a mettere *A* in condizioni di continuare successivamente senza "accorgersi" di nulla
- 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 (nella ISR)*
- 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 Innestati
- 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 Innestati

- 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

Interrupt Innestati (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 di servizio (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.

Esempio di "Exception Vector Table"

0	RESET (SSP)
1	RESET (PC)
16-23	UNASSIGNED, RESERVED
25	LEVEL 1 AUTOVECTOR
31	LEVEL 7 AUTOVECTOR
32-47	TRAP #0-15 INSTRUCTIONS
64-255	USER DEVICE INTERRUPTS

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

Interruzioni e 68000

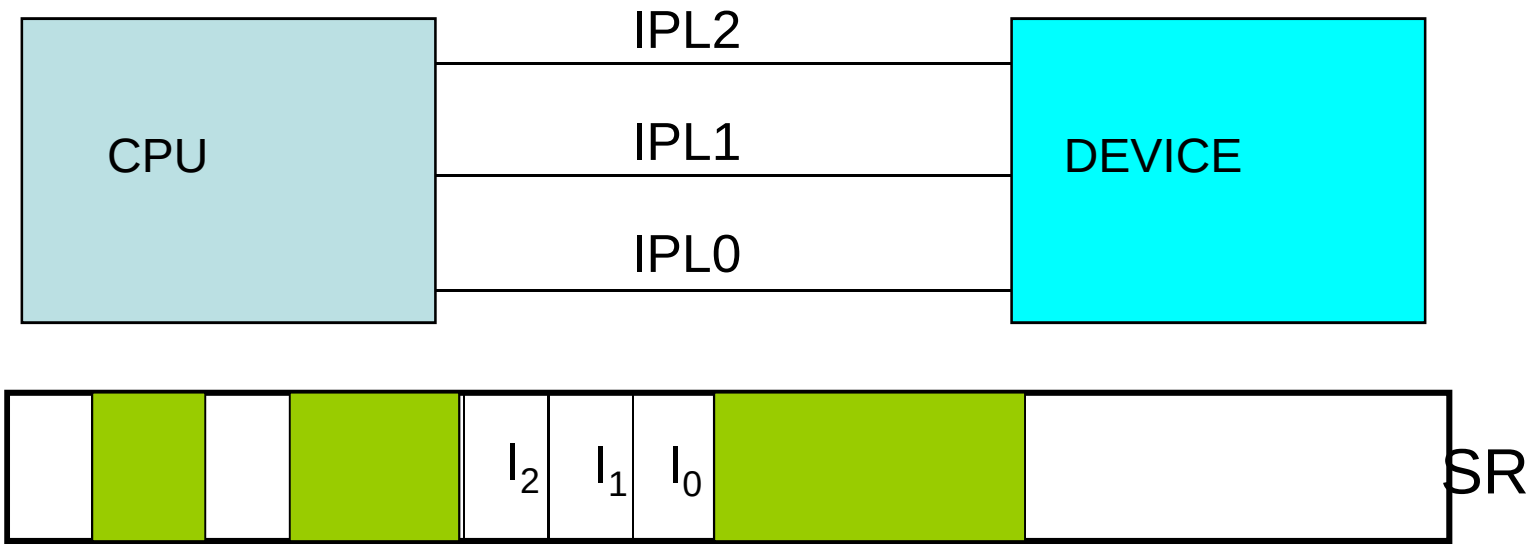
- Il 68000 Supporta le interruzioni vettorizzate
- E’ possibile operare in entrambi i modi descritti
- La periferica emette un segnale sul bus oltre a quella di interrupt per segnalare l’utilizzo dell’interrupt autovettorizzato

La Soluzione del 68000

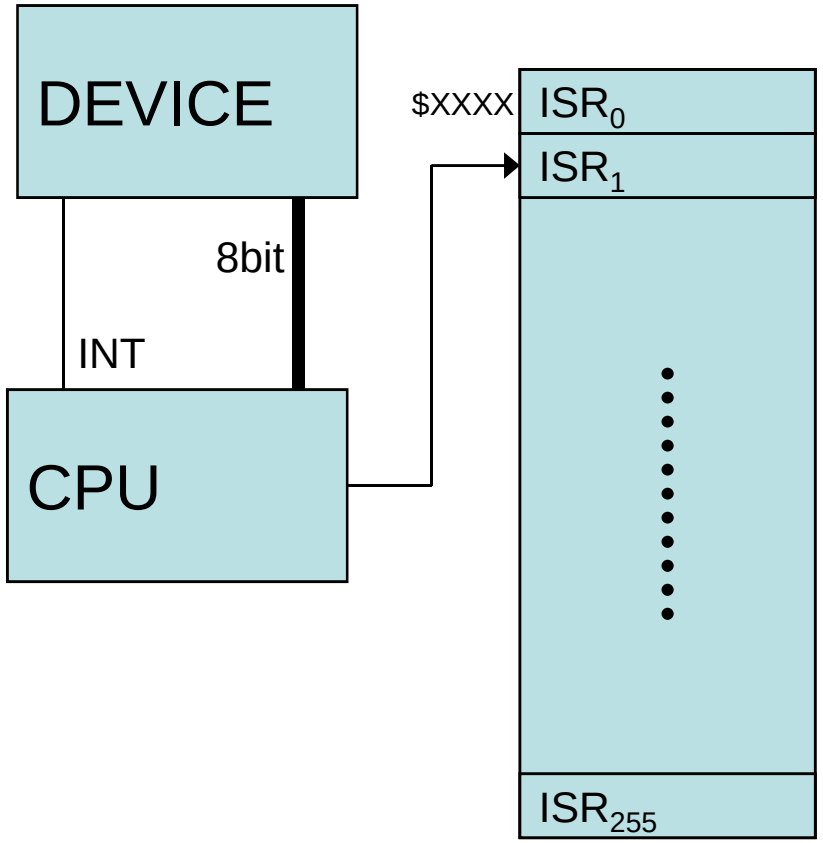
- Segnali:
  - Tre segnali INTL0, INTL1, INTL2
  - 8 Livelli di Priorità
  - 0: nessun Interrupt
  - 1-6: Interrupt Mascherabili
  - 7: Interrupt non mascherabile.
- Dispositivi Esterni gestiscono soluzioni come la Daisy Chain. (es. il PIC)

La Soluzione del 68000

- Tre linee di interruzione sul processore:
  - Interrupt Priority Level
- Tre bit di priorità nello Status Register:
  - Processor Priority Level



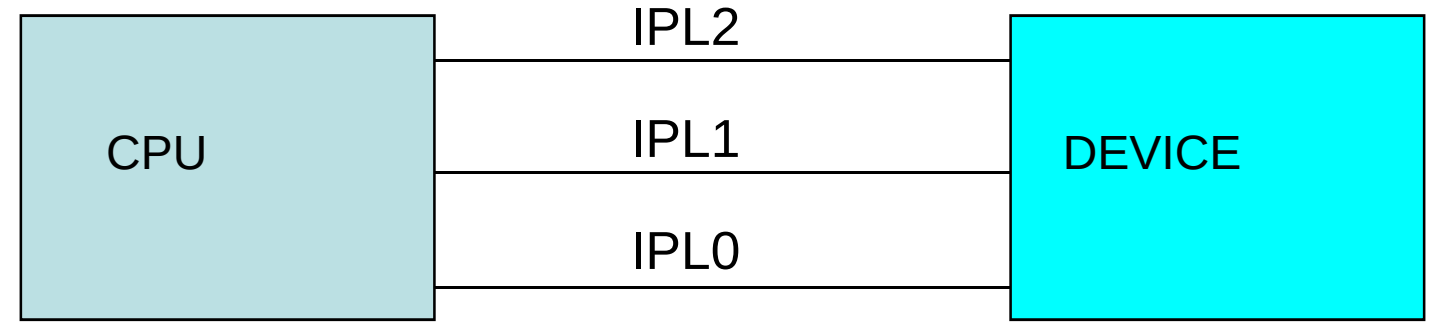
La Soluzione del 68000



- Il processore M68000 utilizza il meccanismo degli interrupt vettorizzati
- In memoria sono presenti 256 locazioni consecutive dette *vettori di interrupt*
- Ciascuna di queste locazioni contiene l'indirizzo di una ISR
- Quando un dispositivo richiede un interrupt, invia al processore un numero di 8 bit che rappresenta il *vettore di interrupt* da utilizzare

Gestione delle priorità

- Problemi:
  - Mascheramento
  - Abilitazione
- Soluzione del 68K:
  - Interrupt Priority Level
  - Processor Priority Level
  - Le interruzioni a priorità 7 non sono mascherabili



Exception Vector Table nel 68000

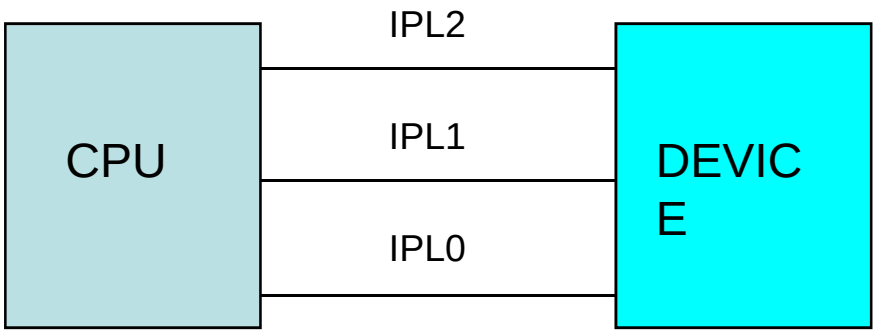
0	RESET (SSP)
1	RESET (PC)
16-23	UNASSIGNED, RESERVED
25	LEVEL 1 AUTOVECTOR
31	LEVEL 7 AUTOVECTOR
32-47	TRAP #0-15 INSTRUCTIONS
64-255	USER DEVICE INTERRUPTS

Servizio mediante autovettore

$64_{HEX}=100_{DEC}$ → 25 LEVEL 1 AUTOVECTOR

$7C_{HEX}=124_{DEC}$ → 31 LEVEL 7 AUTOVECTOR

$(60 + 4 * n)_{HEX}$



Driver con Interruzioni

- Si sviluppa la procedura ISR
- La si carica in memoria all'indirizzo predefinito
- Si associa la procedura al suo codice e lo si fornisce al dispositivo.

Stati di esecuzione del processore 68000

Il MC68000 dispone di tre stati di funzionamento:

- Normale: il processore esegue le istruzioni proprie di un programma applicativo

- Halted: il processore non esegue alcuna istruzione; si può trovare in questo stato a causa di un errore di sistema catastrofico (errori hardware esterni o double bus fault) o se viene asserita la linea di HALT.

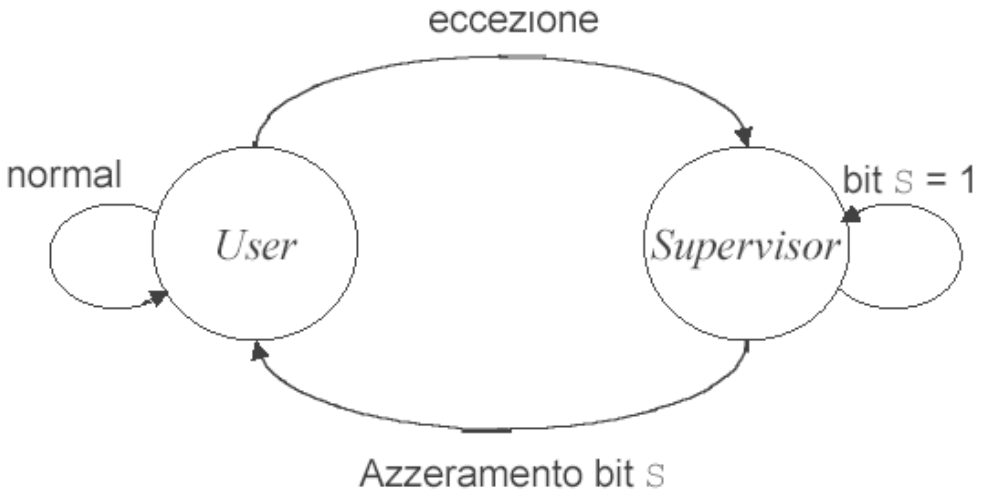
- Eccezione: il processore esegue una routine di gestione di un'eccezione. Questo stato è associato alle istruzioni di trap, interruzioni, alle condizioni di tracing e ad altre condizioni di eccezioni.

Stati di privilegio

Il processore può lavorare in due stati di privilegio:

- *utente* (*user*): livello di privilegio più basso (non è possibile accedere alle istruzioni privilegiate);
- *supervisore* (*supervisor*): livello di privilegio più alto (si può accedere a tutte le istruzioni e a tutte le risorse del sistema; tipico della gestione delle eccezioni).

La presenza di due stati di privilegio aumenta la sicurezza all'interno del sistema.



Sicurezza

Le istruzioni privilegiate che possono essere eseguite solo nel modo *supervisor* sono:

- STOP
- RESET
- RTE
- MOVE to SR
- ANDI to SR
- EORI to SR
- ORI to SR
- MOVE to USP.

Porta il processore nello stato *halted*; non può essere concessa ad un utente perchè ciò potrebbe bloccare i processi di altri utenti.

Invia un segnale di reset a tutti i periferici connessi all'esterno. In un ambiente multi-utente tali dispositivi possono essere usati anche da altri utenti e dunque occorre impedire che un utente possa eseguire questa operazione.

Esegue il ritorno da una procedura di gestione delle eccezioni. Tutte le procedure di gestione delle eccezioni hanno il privilegio di *supervisor*.

Cambiamento dello stato di privilegio

Le istruzioni per il cambiamento dello stato di privilegio (bit S) sono istruzioni privilegiate.
Di conseguenza, il passaggio dal livello utente a quello supervisore avviene solo nel momento in cui bisogna gestire un'eccezione.

Per eseguire una transizione dallo stato utente allo stato supervisore occorre:

- settare il bit S (transizione da stato utente a stato supervisore)
- salvare il contenuto del registro di stato

Cambiamento dello stato di privilegio

La transizione da stato supervisore a stato utente può avvenire in diversi modi:

- attraverso l'esecuzione di un'istruzione RTE l'eccezione termina ed il processore torna allo stato utente;

- azzerando il bit S attraverso un'esplicita istruzione di MOVE to SR o attraverso una delle istruzioni logiche ANDI to SR e EORI to SR.

Vettore delle eccezioni

I vettori delle eccezioni sono locazioni di memoria da cui il processore preleva l'indirizzo di una routine che si occupa della gestione di un'eccezione

Tutti i vettori delle eccezioni sono situati nello spazio dati del Supervisore (tabella dei vettori delle eccezioni, dall'indirizzo 0 all'indirizzo 1023)

Tutti i vettori delle eccezioni sono lunghi 2 word, eccetto il vettore di reset che è lungo 4 word: la prima long word contiene il nuovo valore dello Stack Pointer Supervisore; la seconda contiene l'indirizzo della routine di inizializzazione del sistema

Interruzioni

Il 68000 presenta un bus per la richiesta degli interrupt a 3 bit, composto dalle linee IPL2, IPL1 e IPL0, ai quali un dispositivo esterno può applicare un numero di 3 bit che codifichi il proprio *interrupt-request priority level* (IPL).
Sono disponibili sette livelli di priorità (da 1 a 7).

IPL=0 → non è presente alcuna richiesta di interruzione

IPL=7 → interruzione non mascherabile

Le richieste di interruzione in arrivo al processore non forzano immediatamente il passaggio alla gestione dell'eccezione, ma restano pendenti e vengono rilevate alla fine del ciclo di esecuzione dell'istruzione corrente.
Il processore accetta di servire le richieste con priorità maggiore della maschera delle interruzioni (PPL, *processor priority level*) o con IPL=7.

Nel momento in cui viene accettata un'interruzione, viene posto PPL=IPL

Interruzioni

Il 68000 supporta due modalità di riconoscimento dell'interruzione

Interruzione vettorizzata

Il dispositivo che interrompe si identifica fornendo un codice (*vector number*) a partire dal quale è possibile risalire all'indirizzo di partenza della ISR in grado di servirlo

Interruzione autovettorizzata

Il dispositivo che interrompe richiede una gestione automatica dell'interruzione e non fornisce alcun codice. Il vector number è pari a 24+IPL

In ogni caso, l'indirizzo del vettore è pari a 4*n (dove n è il vector number)

Interruzioni non Mascherabili

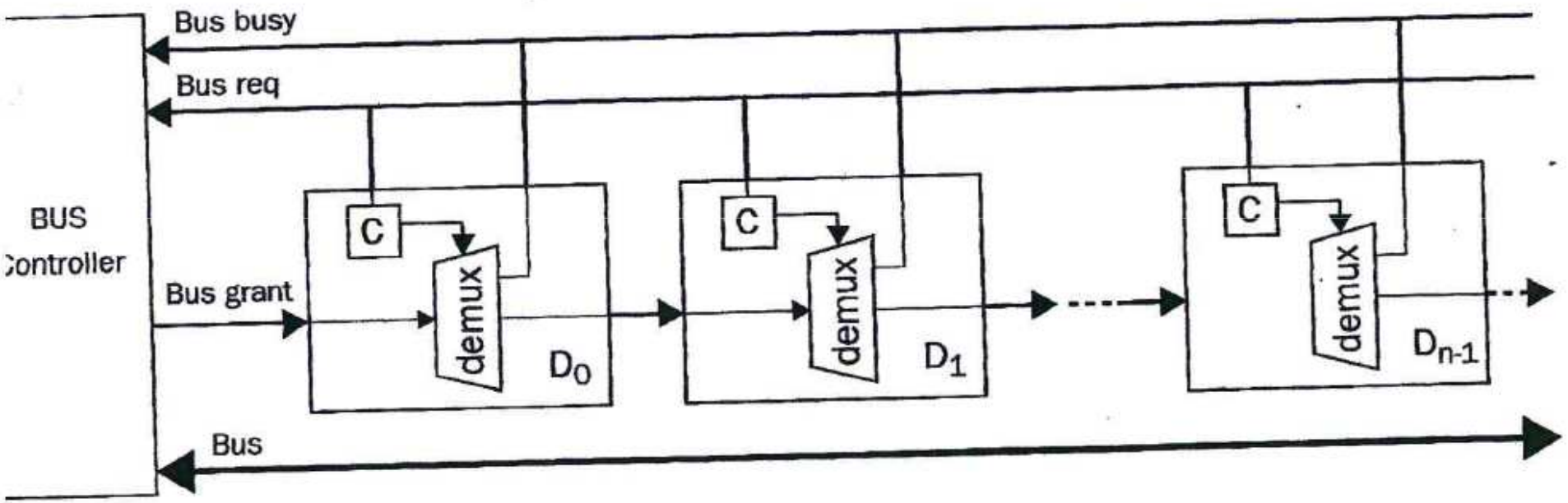
Il 68000 non presenta ingressi espliciti per le interruzioni non mascherabili. Se IPL=7, l'interruzione è sempre servita (anche se PPL=7).

Le interruzioni non mascherabili sono edge-triggered: gli ingressi relativi sono sensibili sul fronte.

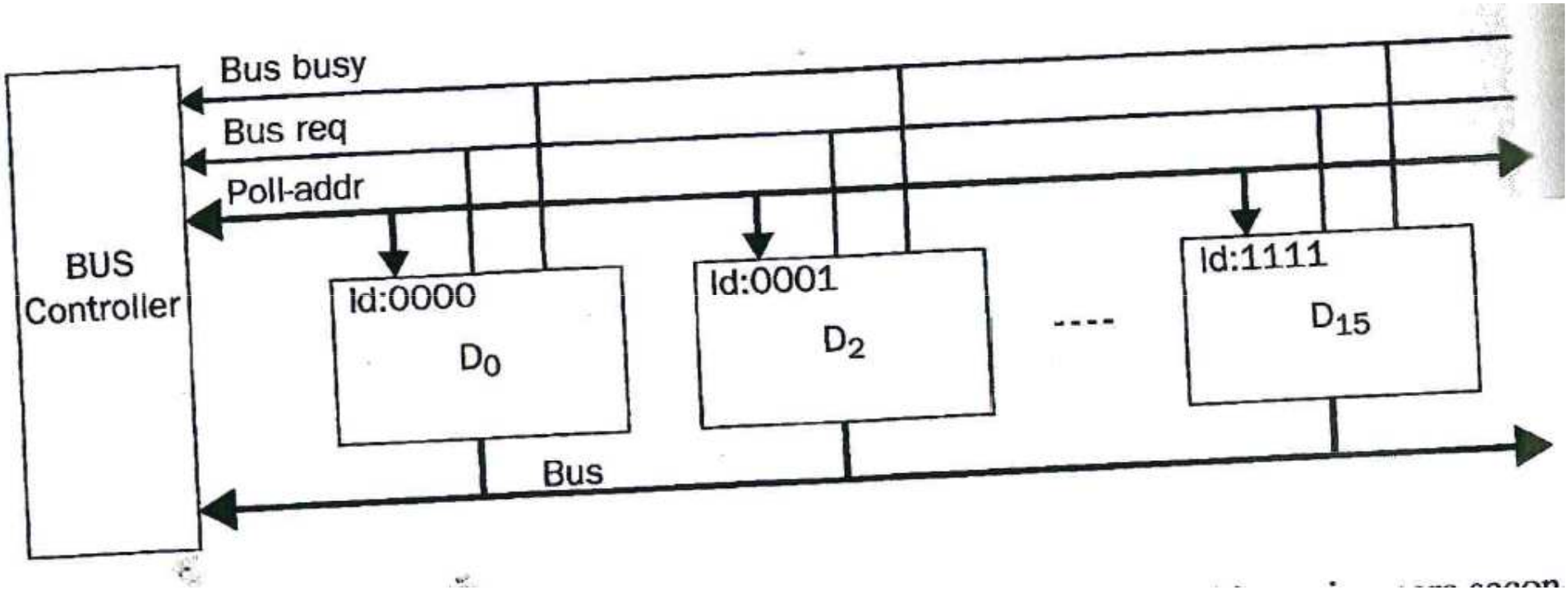
Per evitare che la stessa richiesta reinterrompa, nelle normali interruzioni è sufficiente porre PPL=IPL.

Ciò non vale, invece, nel caso in cui IPL=7. Allora, è richiesto un comportamento sensibile sul fronte di modo che venga riconosciuta una sola transizione dallo stato IPL<7 allo stato IPL=7.

Daisy-chaining



Polling


Bus busy
Bus req
Poll-addr
BUS
Controller
Id:0000
D_0
Id:0001
D_2

Id:1111
D_{15}
Bus

Hand-shaking