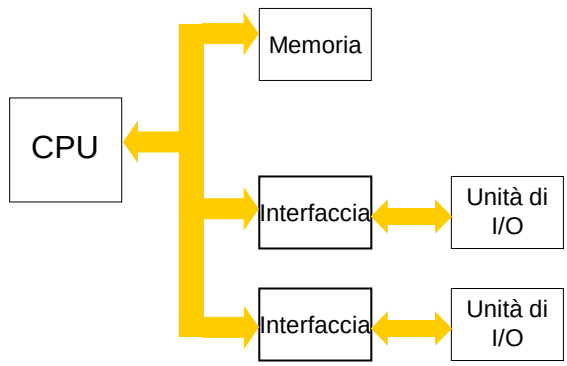
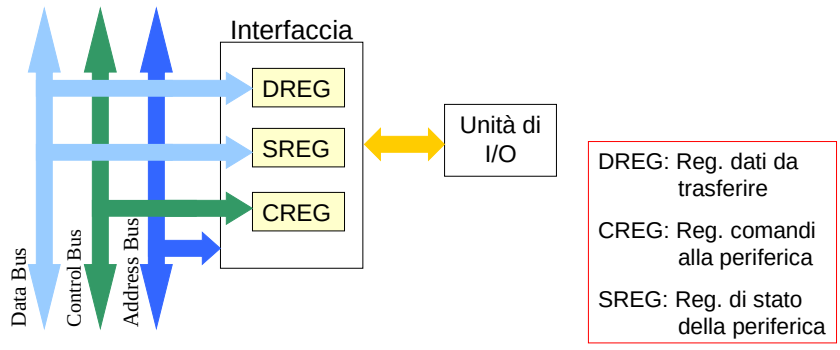
Il sistema di I/O

Architettura a bus singolo



Interfaccia

- Svolge la funzione di adattamento sia elettrico sia logico tra le unità periferiche e il calcolatore



Indirizzamento dell' I/O

- I registri di interfaccia sono caratterizzati da un proprio indirizzo, in modo analogo a ciò che avviene per le celle di memoria.
- Lo spazio degli indirizzi di I/O è disgiunto da quello della memoria
- Esistono due organizzazioni:
  - I/O mappato in memoria (memory mapped)
    - Si usano le stesse istruzioni per la lettura/scrittura in memoria
    - Lo spazio di I/O è mappato in una porzione dello spazio di memoria
  - I/O isolato (I/O mapped)
    - Si usano istruzioni dedicate alle operazioni di I/O

Tecniche di gestione delle unità periferiche

- Controllo di programma: polling
- Controllo di interruzione
- Accesso diretto alla memoria

Controllo di programma: Polling

- In questo caso la gestione dei dispositivi di I/O è totalmente demandata alla CPU.
- E' basato sulla scansione periodica di tutti i dispositivi per verificare se qualcuno di essi richiede un servizio
- La gestione del polling viene fatta attraverso un ciclo software di lettura dello stato dei dispositivi di I/O.
- La maggior parte del tempo è impiegata dal programma principale nell'esecuzione del ciclo di polling.
- 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 del polling

- Ogni dispositivo deve dipendere dalla CPU per essere servito. Ne consegue che:
  - il tempo che nel caso peggiore il dispositivo deve attendere prima che la CPU esegua il trasferimento richiesto può essere alto, e dipende da quanti dispositivi di I/O sono connessi
  - 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.

Costo del Polling?

Assumiamo che per un processore a 500 MHz siano richiesti 400 cicli di clock per l'operazione di polling.

- Mouse: testato 30 volte/sec per non perdere i movimenti dell'utente
  - Polling Clock/sec= 30*400=12000 clock/sec
  - %Processore per polling = $12*10^3/(500*10^6)$= 0,002% *Impatto limitato*

- Floppy: trasferimento dati di 2 byte alla velocità di 50 KB/sec per non perdere dati
  - Polling Clock/sec=50KB/2*400=10,000,000 clock/sec
  - %Processore per polling= $10*10^6/(500*10^6)$= 2% *Basso*

- Hard disk: trasferimento di blocchi di 16 byte alla velocità di 8MB/sec per non perdere dati
  - Hard Disk Polling Clock/sec=8MB/16*400=$200*10^6$ clock/sec
  - %Processore per polling= $200*10^6/(500*10^6)$= 40% *Inaccettabile*

Un alternativa al polling: il sistema di interruzione

È il dispositivo di I/O che comunica al processore il suo stato di pronto.

Il processore, presumibilmente impegnato nella esecuzione di una sequenza di istruzioni "utili", la interrompe temporaneamente per eseguire la sequenza prevista dal protocollo del I/O

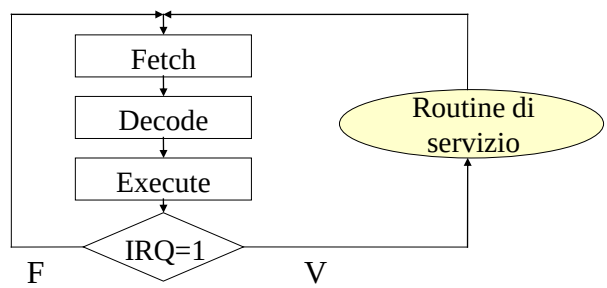
Tale meccanismo viene realizzato dal ***sistema di interruzione***

Il segnale inviato dal dispositivo di I/O (IRQ) prende il nome di ***richiesta di interruzione***.

La sequenza di istruzioni che il processore esegue in seguito ad una interruzione viene detta ***routine di servizio***.

Sistema di interruzione

- La richiesta di interruzione di un dispositivo di I/O è asincrona rispetto all'esecuzione delle istruzioni
  - Non è associata ad alcuna istruzione e può essere attivata durante l'esecuzione di ogni istruzione
  - Viene valutata solo alla fine dell'esecuzione di ogni istruzione



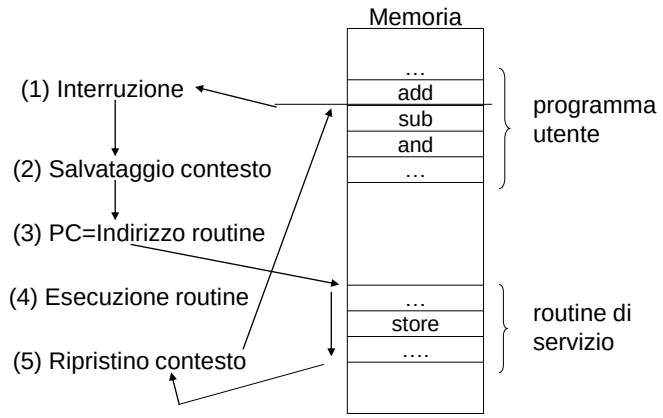
Funzioni del sistema di interruzione

F1. Deve garantire che una interruzione non provochi interferenze sul programma interrotto.

F2. È necessario che il sistema di interruzione riconosca il dispositivo interrompente

F3. Deve provvedere alla gestione delle priorità delle richieste di interruzioni

Cambio di contesto



Salvataggio del contesto

- **Contesto:** Program Counter (PC), Registro di Stato (SR), Registri di uso generale

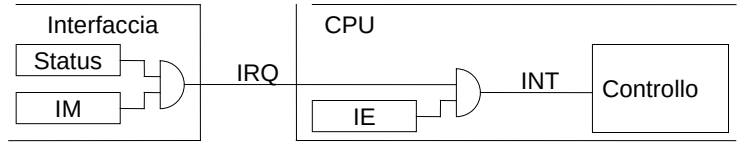
- PC e SR devono essere salvati via hardware nello stack.
  - MEM[SP]=SR; SP--;
  - MEM[SP]=PC; SP--;

- Gli altri registri possono essere salvati via software nello **stack** (PUSH $i)
  - Vengono salvati solo i registri che verranno utilizzati.
  - Questo compito è demandato alla routine di servizio che lo svolge nel suo preambolo.

Salvataggio del contesto

- Il salvataggio del contesto deve essere non interrompibile per evitare situazioni anomale
  - Il processore viene dotato di un flag IE indicante la interrompibilità del processore.
  - Nel momento in cui viene accettata la richiesta di interruzione IE viene resettato e il processore diventa non interrompibile
  - Per rendere nuovamente interrompibile il processore bisogna usare un'apposita istruzione

Un sistema di interruzione



IM=Interrupt Mask

- Quando il dispositivo è pronto, pone STATUS=1
- Se IM=1 e le interruzioni sono abilitate (IE=1) viene servita la richiesta di interruzione (al termine dell'istruzione corrente).
  - Viene posto IE=0
  - Viene salvato il contesto
  - PC=Indirizzo della Routine di servizio

Ripristino del contesto

- Via software, nell'epilogo della routine di servizio, vengono ripristinati i valori dei registri salvati nello stack (POP)
- L'uscita dalla routine di servizio avviene mediante un'apposita istruzione di ritorno da interruzione (RTI) che ripristina la parte di contesto salvata via hardware
  - SR=MEM[SP]; SP++;
  - PC=MEM[SP]; SP++;
- In alcuni processori la RTI riabilita anche il flip flop IE (IE=1), in altri è necessario utilizzare un'apposita istruzione;

Benefici dell'I/O mediante sistema di interruzione

- Supponiamo che siano richiesti 500 cicli di clock (a 500 MHz) per ogni trasferimento, compreso l'interrupt.
- Trovare la % del tempo consumato dal processore se lo hard disk è attivo il 5% del tempo.
- Interrupt rate = polling rate
  - Disk Interrupts/sec = 8 MB/s /16B
    = 500K interrupts/sec
  - Disk Polling Clocks/sec = 500K * 500
    = 250,000,000 clocks/sec
  - % Processor for during transfer: $250*10^6/(500*10^6)= 50\%$
- Disk active 5% $\Rightarrow$ 5% * 50% $\Rightarrow$ 2.5% busy

Interruzioni non mascherabili

- Il flip-flop IE serve a mascherare le richieste di di interruzioni.
- Quasi tutti i calcolatori sono dotati di una linea di interruzione non mascherabile (NMI) che viene sempre rilevata se asserita.
- È impiegata per gestire situazioni di emergenza (tipicamente la caduta di tensione).
- Questo tipo di interruzione deve essere prioritaria rispetto ad altre richieste

Riconoscimento del dispositivo di I/O

- In un calcolatore possono essere collegati diversi dispositivi ciascuno caratterizzato da un propria routine di servizio
- È necessario che il sistema di interruzione riconosca il dispositivo interrompente per attivare la relativa routine
- Approcci possibili:
  - via software (*polling*): tramite una sequenza di istruzioni viene individuato il dispositivo interrompente e viene mandata in esecuzione la relativa routine
  - via hardware (*interruzioni vettorizzate*): il dispositivo interrompente invia il codice identificativo

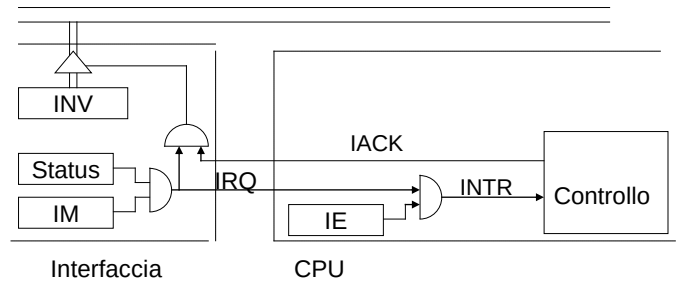
Riconoscimento mediante polling

- In seguito alla richiesta di interruzione:
  1. viene salvato il contesto del programma interrotto
  2. Viene mandata in esecuzione una sequenza di identificazione dell'interruzione che interroga ad uno ad uno i dispositivi fino a trovare quello che ha fatto la richiesta
  3. Viene mandata in esecuzione la routine individuata al passo 2.

- Nel caso in cui più richieste di interruzione sono presenti, viene servita quella che per prima viene incontrata nella sequenza.

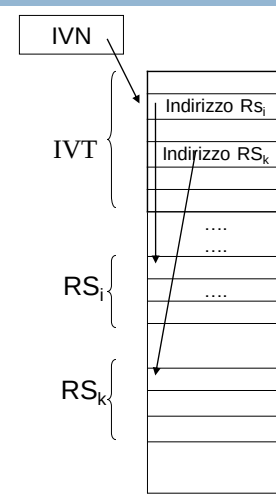
Riconoscimento mediante vettore delle interruzioni

- Nell'interfaccia è presente un registro in cui è memorizzato il codice identificativo del dispositivo (INV).
- In seguito alla richiesta di interruzione IRQ=1, se il processore è interrompibile (IE=1), attiva in risposta il segnale IACK.
- Il codice identificativo viene inviato sul bus dati in risposta al segnale IACK generato dal processore.

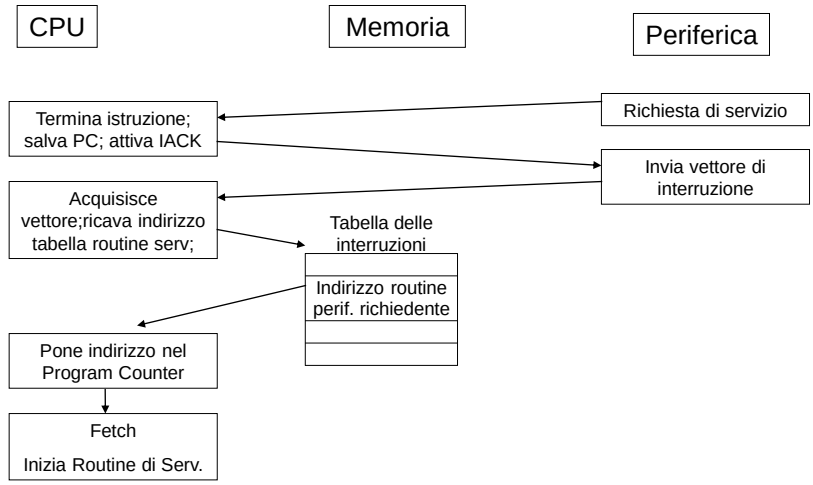


Riconoscimento mediante vettore delle interruzioni

- In memoria è presente una tabella, ***Interrupt Vector Table (IVT)***, che contiene gli indirizzi delle routine di servizio dei dispositivi.
- Il codice inviato dal dispositivo di I/O, ***Interrupt Vector Number (IVN)***, rappresenta l'indice della tabella corrispondente alla routine di servizio.
- La IVT di norma è memorizzata a partire dalle prime posizioni della memoria in modo da codificare l'IVN con pochi bit.
- Il riempimento della IVT (o di parte di essa) viene eseguita ad opera del programmatore



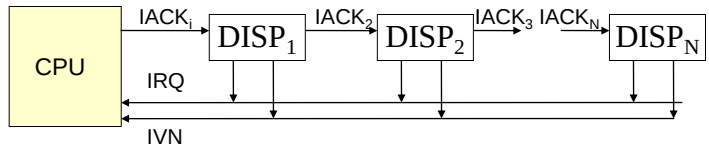
Riconoscimento mediante vettore delle interruzioni



Gestione dei conflitti tra richieste di interruzione

- Più richieste di interruzioni possono entrare in conflitto
- Cause conflitto:
  - Richieste contemporanee
  - Richieste quando è in esecuzione la routine di servizio di una interruzione

Interruzione vettorizzata con Daisy chain



- In seguito ad richiesta di interruzione IRQ, la CPU attiva il segnale IACK che viene inviato ai dispositivi mediane un'unica linea.
- Se un dispositivo $DISP_i$ ha richiesto una interruzione non propaga il segnale IACKi, ponendo $IACK_{i+1}=0$
- Se il dispositivo DISPi non ha fatto alcuna richiesta propaga il segnale in ingresso ponendo $IACK_{i+1}=1$
- **Problema:** la priorità dei dispositivi dipende dalla posizione nella daisy chain: i più vicini alla CPU hanno una più alta priorità statica rispetto a quelli che seguono.

Gerarchia di priorità: interruzione di una routine di servizio

- Approccio software
  - Nel preambolo della routine di servizio attivata, mentre la CPU è non interrompibile, oltre a salvare il contesto vengono definiti i dispositivi di I/O che hanno priorità superiore e che possono interrompere la routine corrente.
  - I dispositivi di I/O vengono abilitati settando i rispettivi flip-flop IM
- Il principale problema è il tempo perso per settare i bit dei dispositivi di I/O

Gerarchia di priorità: interruzione di una routine di servizio

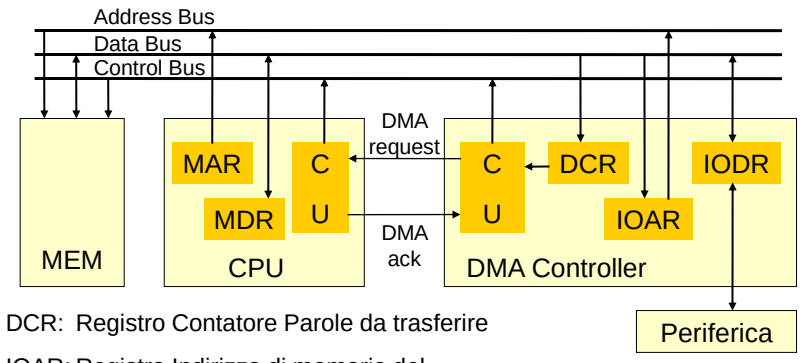
Approccio hardware

- Ad ogni dispositivo di I/O è associata un livello di priorità
- Il processore in ogni istante è caratterizzato da un livello di priorità corrente NPR
- Quando si verifica una richiesta di interruzione prodotta da un dispositivo con priorità K, NPR assumerà il valore K
- La routine di servizio può essere interrotta solo dai dispositivi con priorità >NPR
  - Se NPR=0 non è servita alcuna interruzione
  - Se NPR=N la routine è non interrompibile
- Poiché NPR fa parte del contesto, prima di eseguire la routine di servizio è necessario salvare il suo valore

Accesso diretto alla memoria (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



DCR: Registro Contatore Parole da trasferire

IOAR: Registro Indirizzo di memoria del prossimo dato da trasferire

IODR: Registro dati da trasferire in lettura o scrittura

Trasferimento in DMA

- Il trasferimento di un blocco in DMA si articola in varie fasi:
  - la CPU carica nei registri IOAR e DCR l'indirizzo dell'area di memoria ed il numero di parole da trasferire;
  - quando il DMA Controller è pronto per eseguire il trasferimento invia un segnale di DMA Request alla CPU; quando questa giunge ad un punto di rilevamento di tale segnale, rilascia il bus e attiva il segnale di DMA Acknowledge;
  - il DMA Controller esegue il trasferimento; dopo il trasferimento di ciascuna parola, IOAR e DCR vengono decrementati;

Trasferimento in DMA

- quando il DMA Controller termina il trasferimento (ad esempio perchè la periferica non invia più dati) disattiva DMA Request; la CPU disattiva DMA Acknowledge, e riprende il controllo del bus;

Modalità di trasferimento

- **Il trasferimento dei dati in DMA può avvenire in vari modi:**
  - **trasferimento a blocchi *(burst transfer)***
    - 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.
    - 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.