

Sistemi di INTERRUZIONI



Sono sistemi che modificano il normale flusso di esecuzione delle istruzioni per gestire situazioni anomale o particolari :

- ▶ trasferimenti di dati tra dispositivi,
- ▶ errori durante la esecuzione delle istruzioni,
- ▶ condizioni anomale durante l'esecuzione delle istruzioni, ecc...

Trasferimento dati tra dispositivi

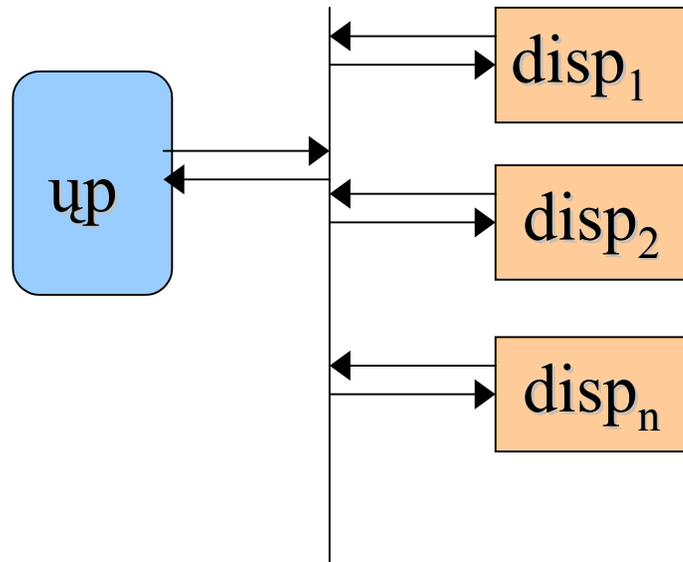
- Il trasferimento dati può essere controllato da :
 - **Programma**: il processore controlla periodicamente lo stato del dispositivo
 - **Interrupts**: il dispositivo invia sul bus un segnale di interruzione per segnalare la sua disponibilità al trasferimento dati

interrupt

Il processore : interrompe il programma in esecuzione e disabilita altri interrupt

. esegue la procedura di servizio associata all'interrupt

. ripristina la esecuzione del programma interrotto, riabilitando altri interrupt



polling

Il processore attende che il dispositivo sia pronto (**busy waiting**) iterando una istruzione che esamina lo stato del dispositivo (*flag di stato*)

Classificazione delle interruzioni

✓ *eccezioni*

✓ *internal interrupt*

✓ *interruzioni sincrone*

✓ *interrupt*

✓ *external interrupt*

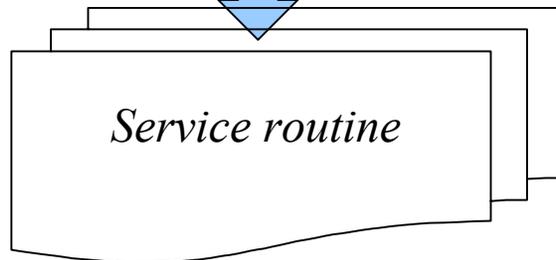
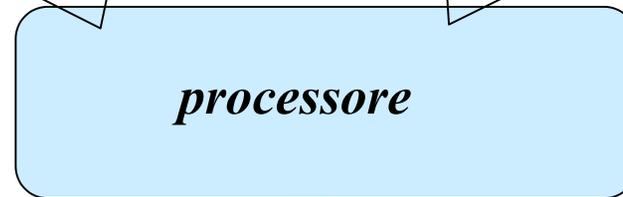
✓ *interruzioni asincrone*

Sono interruzioni generate dalla stessa Unità Centrale durante l'esecuzione di una istruzione in maniera sincrona rispetto ai segnali di clock

Sono interruzioni generate da eventi esterni su dispositivi che operano in parallelo ed indipendentemente dalla Unità Centrale.

- *Errori o condizioni anomale durante l'esecuzione di istruzioni,*
- *richieste di cambiamento di stato del sistema*

- *Condizioni anomale che alterano il normale funzionamento della CPU,*
- *evento legato a scadenze temporali,*
- *richieste di comunicazioni da altri processi*



FUNZIONI di un SISTEMA di INTERRUZIONI

1) Garantire che il verificarsi di una interruzione non provochi interferenze indesiderate sul programma che viene interrotto

(commutazione del contesto - transparency)

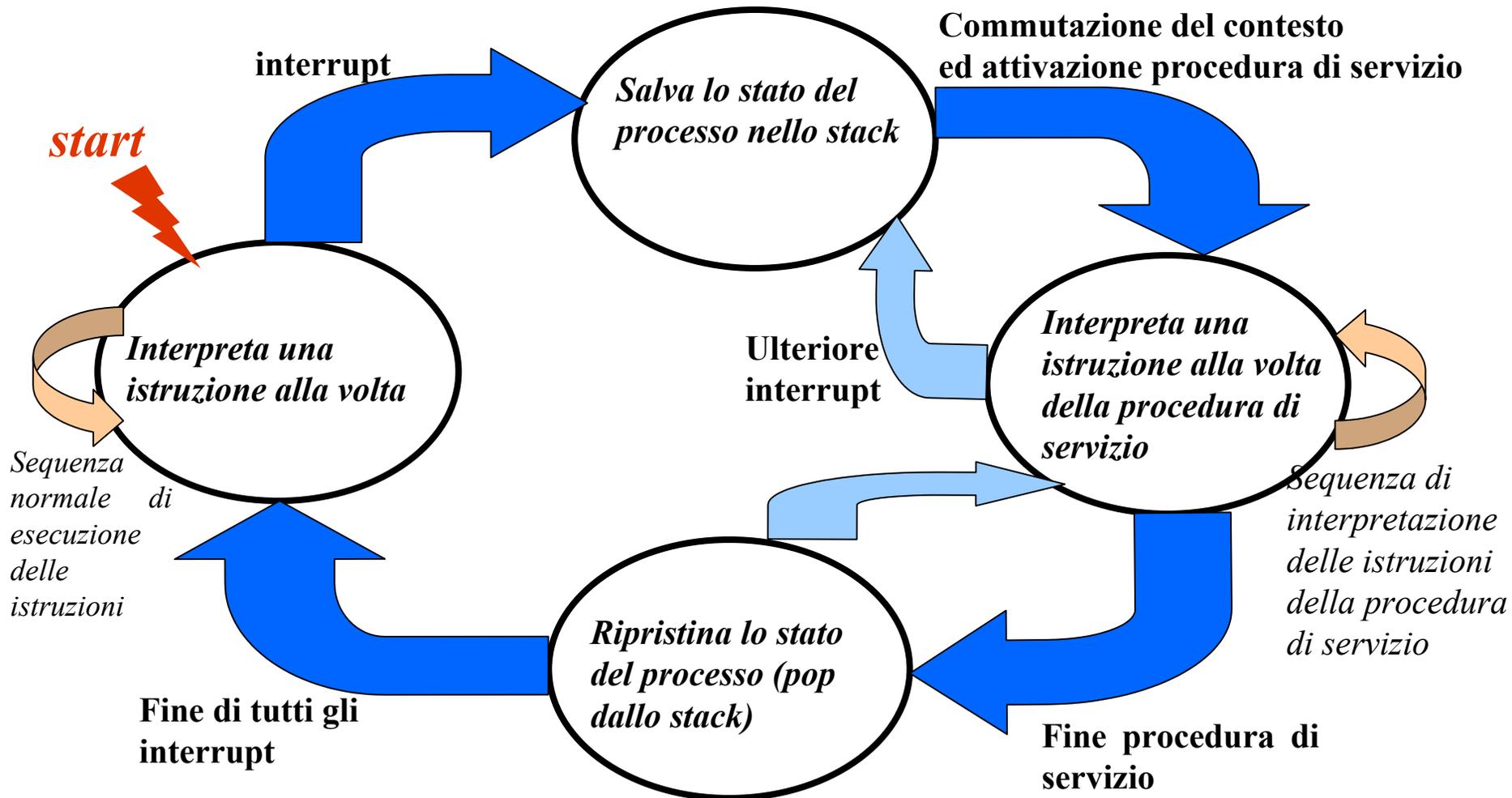
2) il sistema deve saper identificare l'interruzione al fine di selezionare ed attivare la routine di servizio ad essa associata

(interrupt vector)

3) il sistema deve poter gestire gerarchie di priorità delle richieste di interruzione, per evitare i problemi che sorgono quando le richieste di interruzione generate da dispositivi diversi entrano in conflitto

(multiple interrupt, daisy chain)

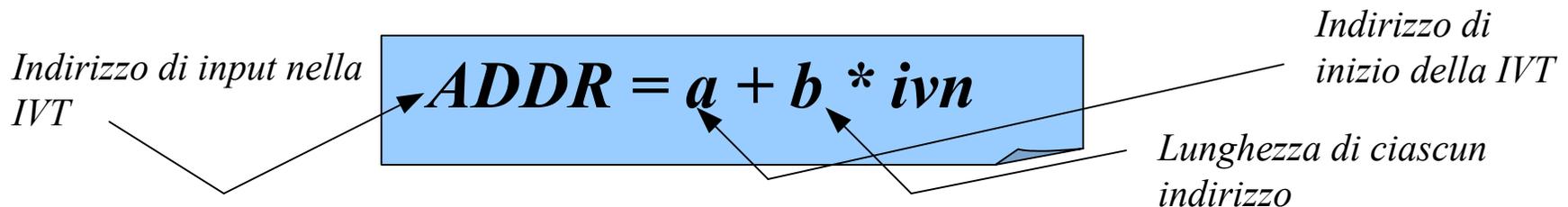
Diagramma di stato per la gestione delle interruzioni



Identificazione degli interrupt

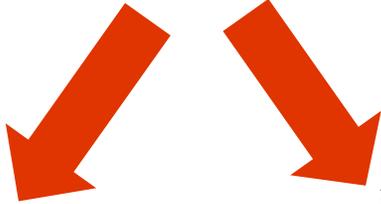
- Gli indirizzi di memoria delle procedure di servizio sono memorizzati in memoria centrale in una tabella (***INTERRUPT VECTOR TABLE - IVT***).
- Una interruzione provocata da un dispositivo o da un evento genera automaticamente un ***INTERRUPT VECTOR NUMBER (IVN)*** che corrisponde all'indice nell'Interrupt vector table della procedura di servizio corrispondente all'interrupt generato.

In questo modo il processore ottiene non solo il riconoscimento del tipo di interruzione, ma anche l'attivazione automatica della corrispondente routine di servizio, tramite un salto indiretto all'indirizzo fornito dall'interrupt vector number



Gerarchie di priorità

Eventi che generano conflitti :



Presenza contemporanea di due o più richieste di interrupt (interruzioni multiple)

Richiesta di interruzione mentre è in corso l'esecuzione di una routine di servizio di un'altra interruzione

La priorità della gestione degli interrupt è determinata da:

→ tipo di dispositivo che ha generato l'interruzione, in genere si assegna priorità più alta a dispositivi con esigenze di tempo reale più critiche

→ tipo di interruzione (stato di interrompibilità)

Tipi di interruzioni

INTERRUPT INTERNI o SINCRONI

TRAP : interruzione generata alla conclusione della esecuzione di una istruzione , cioè dopo che si è verificata la trasformazione del registro di stato (es. overflow, parity error)

FAULT : interrupt generato durante l'esecuzione dell'istruzione con una trasformazione di stato recuperabile , quando cioè il PC punta ancora alla istruzione corrente (es. division by 0)

ABORT: come il fault ma non recuperabile (es. illegal instruction o software interrupt)

TRACING : quando ogni istruzione genera una eccezione subito dopo la sua esecuzione.

INTERRUPT ESTERNI o ASINCRONI

BUS ERROR : dovuto ad un segnale di sincronizzazione che tarda ad arrivare o ad una violazione di protezione di memoria (es. paging)

RESET : in seguito ad un evento catastrofico che provoca il blocco del processore ed il restart del sistema (es. power failure)