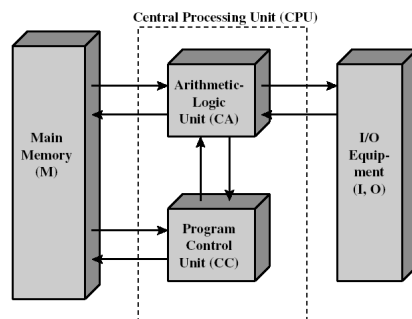


## Architettura e funzionalità di un Sistema di Elaborazione (SE)

## Componenti di un calcolatore

- L'architettura dei calcolatori è generalmente detta architettura di von Neumann

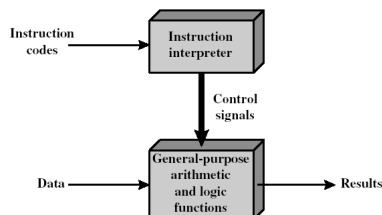


## Componenti di un calcolatore

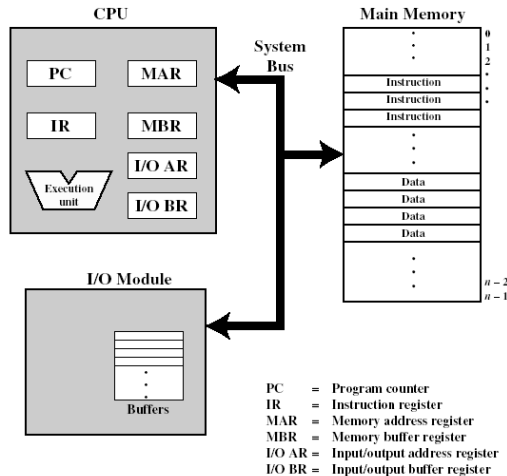
- L'architettura di von Neumann si basa su tre concetti chiave:
  - Dati e istruzioni sono memorizzati in una memoria di lettura e scrittura
  - I contenuti della memoria sono indirizzabili in base alla loro posizione, indipendentemente dal tipo
  - L'esecuzione del codice avviene in modo sequenziale da un'istruzione alla successiva (a meno che non sia esplicitamente indicato il contrario)

## Componenti di un calcolatore

- I calcolatori sono composti da un piccolo numero di componenti logiche
  - È possibile combinare opportunamente queste componenti per risolvere qualunque problema
- Per "costruire" opportune configurazioni aritmetico-logiche si fa in modo che l'hardware esegua operazioni differenti in base ad segnali di controllo
  - Microprogrammazione

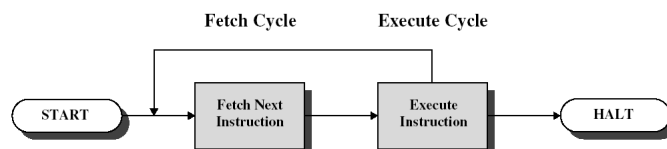


# Componenti di un calcolatore



# Funzionalità di un calcolatore

- La funzione principale di un calcolatore è quella di eseguire un programma, formato da un insieme di istruzioni immagazzinate in memoria.
- L'elaborazione di una singola istruzione è detta ciclo di istruzione

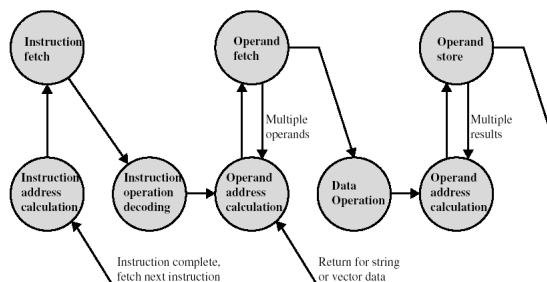


## Caricamento ed esecuzione di una istruzione

- Ad ogni ciclo il processore carica un'istruzione dalla memoria:
  - Il PC contiene l'indirizzo della prossima istruzione che deve essere eseguita
- L'istruzione caricata viene memorizzata nel registro IR
  - Il codice operativo dice quale azione deve essere intrapresa
- Ci possono essere 4 tipi di "azioni":
  - Trasferimento dati processore-memoria
  - Trasferimento dati processore-I/O
  - Elaborazione dati
  - Controllo

## Caricamento ed esecuzione di una istruzione

- A seconda del numero di operandi coinvolti nell'operazione da eseguire, il ciclo di esecuzione di un'istruzione può coinvolgere più di un riferimento alla memoria e/o a dispositivi di I/O



## Classificazione dei processori

- I processori possono essere classificati in base al numero di operandi esprimibili nella singola istruzione.
- Se il numero di operandi è minore, il tempo di decodifica ed esecuzione dell'istruzione è minore, come pure la memoria richiesta dal codice.
- D'altro canto una stessa operazione richiede un numero di istruzioni maggiore per essere eseguita, se queste hanno un numero ridotto di operandi.

## Processori a 3 operandi

- Ogni istruzione contiene l'indicazione dei 2 operandi e del risultato
- Esempio: si supponga di voler eseguire l'operazione  
 $X = A \times B + C \times C$
- Codice:  
MULT T, A, B  
MULT X, C, C  
ADD X, X, T

## Processori a 2 operandi

- Ogni istruzione contiene l'indicazione dei 2 operandi; il risultato coincide con uno dei 2.
- Esempio:

```
MOV T, A
MULT T, B
MOV X, C
MULT X, C
ADD X, T
```

## Processori a 1 operando

- Il secondo operando ed il risultato corrispondono al registro accumulatore.
- Esempio:

```
LOAD A
MULT B
STORE T
LOAD C
MULT C
ADD T
STORE X
```

## Processori a 0 operandi

- Ogni istruzione preleva i 2 operandi dallo stack tramite 2 operazioni di push, e memorizza il risultato nello stack tramite una pop.

- Esempio:

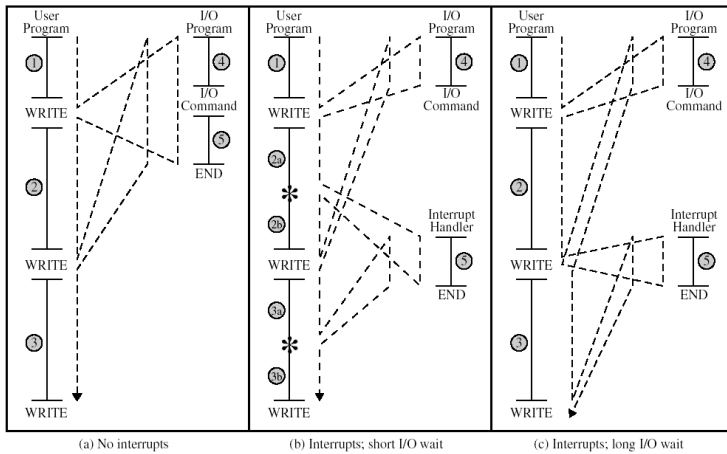
```
PUSH A
PUSH B
MULT
PUSH C
PUSH C
MULT
ADD
POP X
```

## Interruzioni

- In tutti i calcolatori esiste un meccanismo che permette alla memoria e ai dispositivi di I/O di poter interrompere l'elaborazione eseguita dal processore attraverso delle "interruzioni".

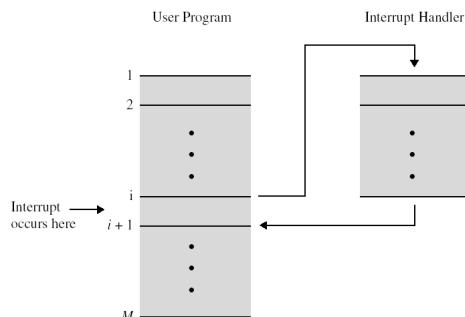
|           |  |
|-----------|--|
| Programma | Generato da una condizione che si verifica in seguito all'esecuzione di un'istruzione (overflow, divisione per zero, ecc.) |
| Timer     | Generato da un timer interno al processore; permette al sistema operativo di eseguire alcune operazioni in modo regolare   |
| I/O       | Generato da un controllore di I/O per segnalare il completamento di un'operazione o un errore                              |
| Guasto    | Generato da un guasto (errore di parità, ecc.)   |

# Flusso di programma con e senza interruzioni

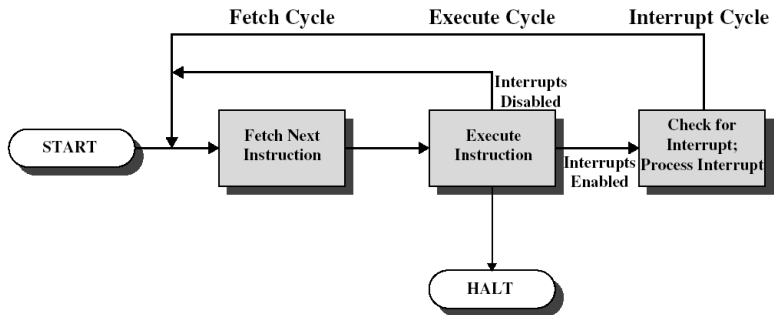


# Interruzioni e ciclo di istruzione

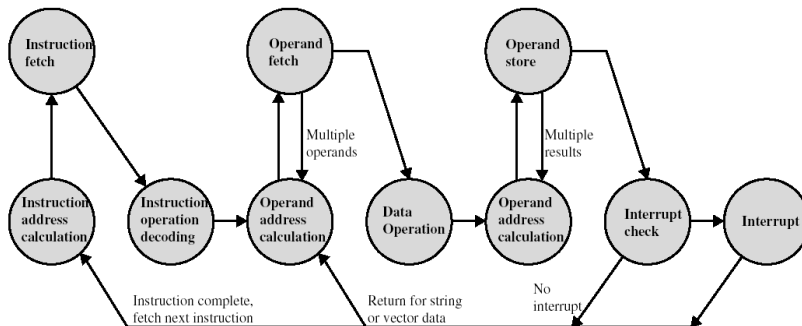
- Con l'uso delle interruzioni il processore può occuparsi dell'esecuzione di altre istruzioni mentre un'operazione di I/O è in corso.
- Dal punto di vista del programma utente un'interruzione è vista come una qualunque routine:
  - Processore e sistema operativo sono gli unici responsabili della sospensione e del ripristino del programma utente



# Interruzioni e ciclo di istruzione



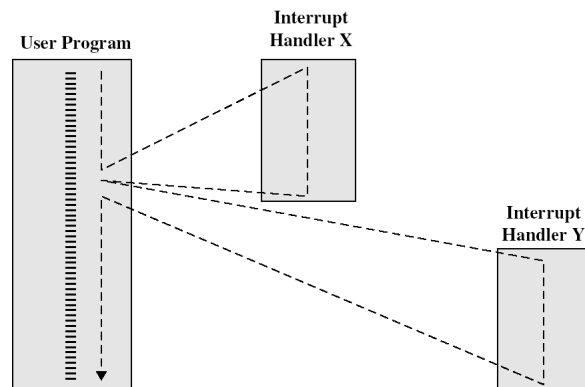
# Interruzioni e ciclo di istruzione



## Interruzioni multiple

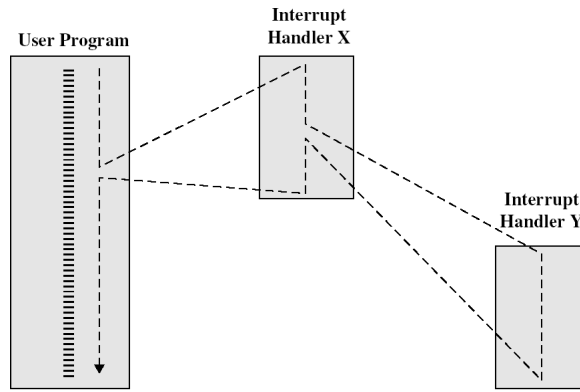
- È possibile che un programma riceva un'interruzione mentre è in esecuzione la procedura di gestione di un'altra interruzione.
- Sono possibili 2 approcci:
  - Le interruzioni vengono disabilitate durante l'esecuzione di una routine di interruzione
  - SI assegnano delle priorità alle interruzioni per permettere che un'interruzione a priorità alta ne possa "bloccare" una a priorità più bassa

## Interruzioni multiple



(a) Sequential interrupt processing

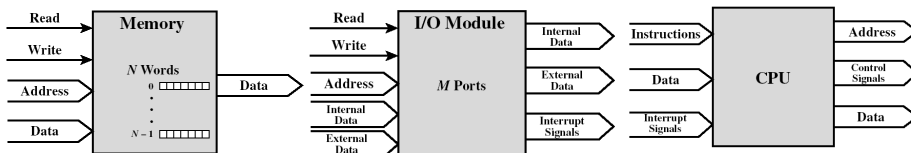
# Interruzioni multiple



(b) Nested interrupt processing

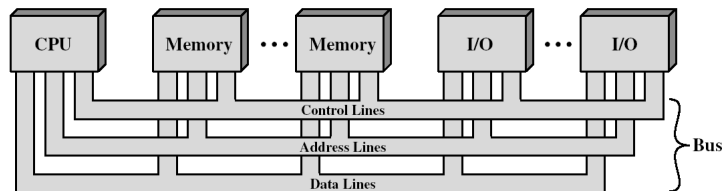
# Strutture di interconnessione

- Un calcolatore è formato da componenti (moduli) di tre tipi base:
  - CPU
  - Memoria
  - I/O
- La connessione di questi elementi è detta struttura di interconnessione



## Struttura del bus

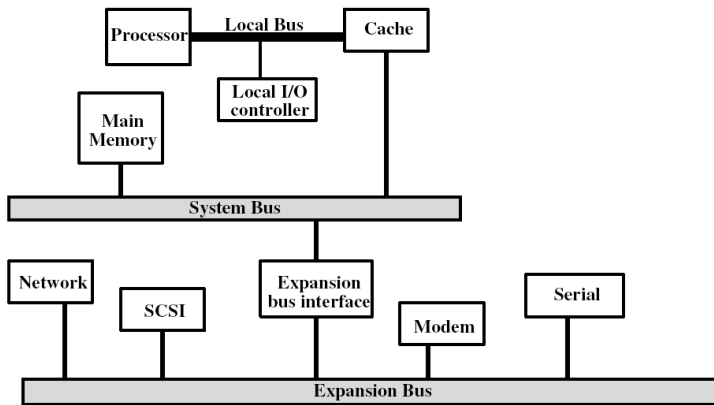
- Un bus di sistema è formato da un insieme (50-100) di linee diverse ad ognuna delle quali è assegnato un particolare significato.
- È possibile classificare le linee in tre gruppi:
  - Linee di dati
  - Linee di indirizzo
  - Linee di controllo



## Gerarchie di bus multipli

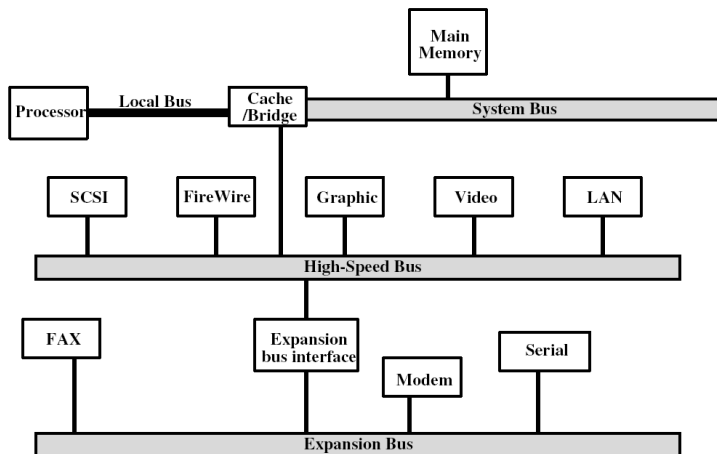
- Aumentando il numero di dispositivi collegati ad un bus si avrà un calo delle prestazioni:
  - Più dispositivi sono collegati al bus più, in generale, sarà lungo il bus e maggiori saranno i ritardi di propagazione dei segnali
  - Il bus può diventare un collo di bottiglia se la mole di dati da trasferire si avvicina alla sua capacità
- Soluzione: gerarchia di bus multipli

# Gerarchie di bus multipli



(a) Traditional Bus Architecture

# Gerarchie di bus multipli



(b) High-Performance Architecture

## Elementi di progettazione di un bus

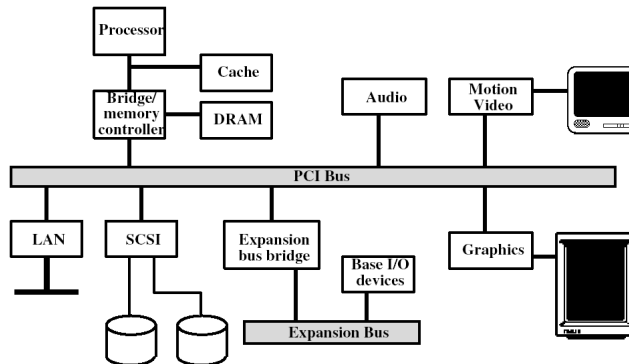
- È possibile identificare alcuni parametri fondamentali per classificare e differenziare i bus.
- Tipo:
  - Dedicato
  - Generico (multiplexed)
- Metodo di arbitraggio:
  - Centralizzato
  - Distribuito
- Temporizzazione:
  - Sincrona
  - Asincrona

## Elementi di progettazione di un bus

- Larghezza del bus:
  - Indirizzo
  - Dati
- Tipo di trasferimento dei dati:
  - Lettura
  - Scrittura
  - Lettura-modifica-scrittura
  - Lettura-dopo-scrittura
  - Blocco

# Bus PCI

- PCI: Peripheral Component Interconnect
- Lo standard attuale permette, al massimo, l'uso di:
  - 64 linee di dati a 66 MHz per una larghezza di banda di 528 Mbyte/s



(a) Typical Desktop System

# Bus PCI: arbitraggio

- Il PCI usa uno schema di arbitraggio centralizzato sincrono:
  - Ogni modulo dispone di una linea per la richiesta del bus (REQ) e per ricevere la concessione (GNT)
  - Queste linee sono collegate ad un arbitro centrale
  - Lo standard PCI non indica uno specifico algoritmo di arbitraggio

