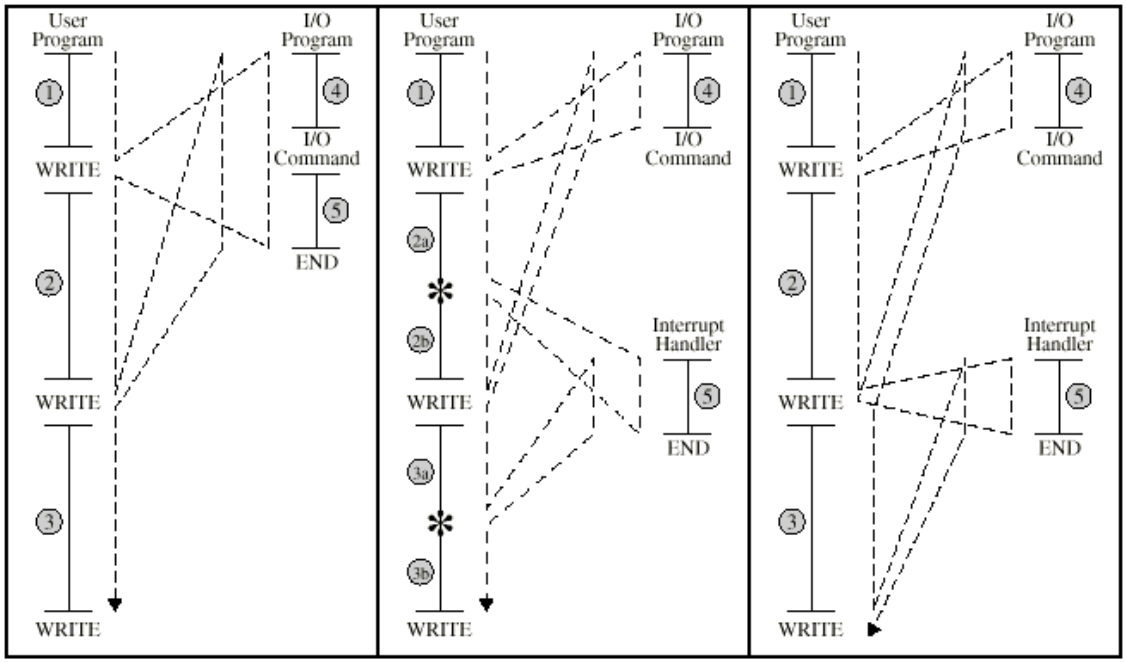
Interruzioni

- Meccanismo tramite il quale altri moduli (esempio I/O) possono interrompere la normale sequenza di esecuzione
- Tipiche interruzioni
  - Program
    - Esempio: overflow, division by zero
  - Timer
    - Generata da un timer interno alla CPU
  - I/O
    - Per segnalare la fine di un'operazione di I/O
  - Guasto hardware
    - Esempio: mancanza di alimentazione

Perché interrompere?

- Per migliorare l'efficienza della elaborazione
- Esempio:
  - Molti dispositivi esterni sono più lenti del processore
  - Per evitare che la CPU attenda la fine di un'operazione di I/O

Esempio



Trasferimento del controllo per una interruzione

USER PROGRAM

⟨statement⟩
⟨statement⟩
⋮
⟨statement⟩
} Code segment 1

WRITE

⟨statement⟩
⟨statement⟩
⋮
⟨statement⟩
} Code segment 2

WRITE

⟨statement⟩
⟨statement⟩
⋮
⟨statement⟩
} Code segment 3

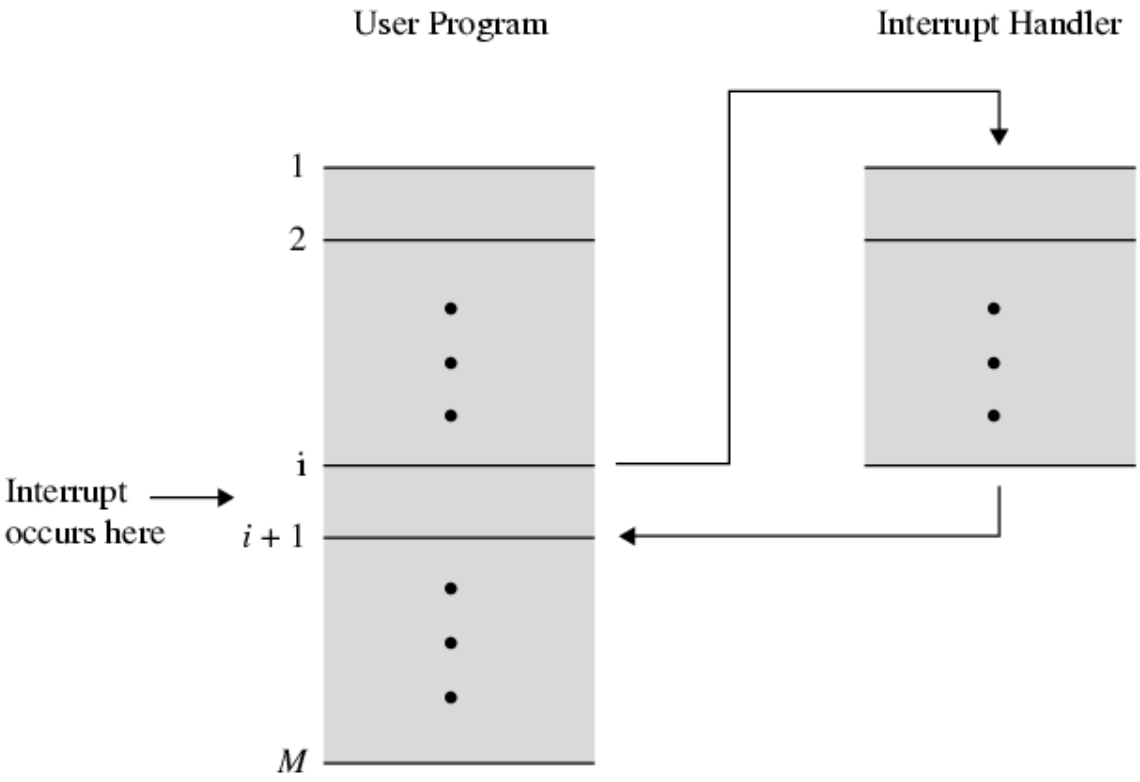
I/O PROGRAM

⟨statement⟩
⟨statement⟩
⋮
⟨statement⟩
} Code segment 4

I/O command

⟨statement⟩
⟨statement⟩
⋮
⟨statement⟩
} Code segment 5

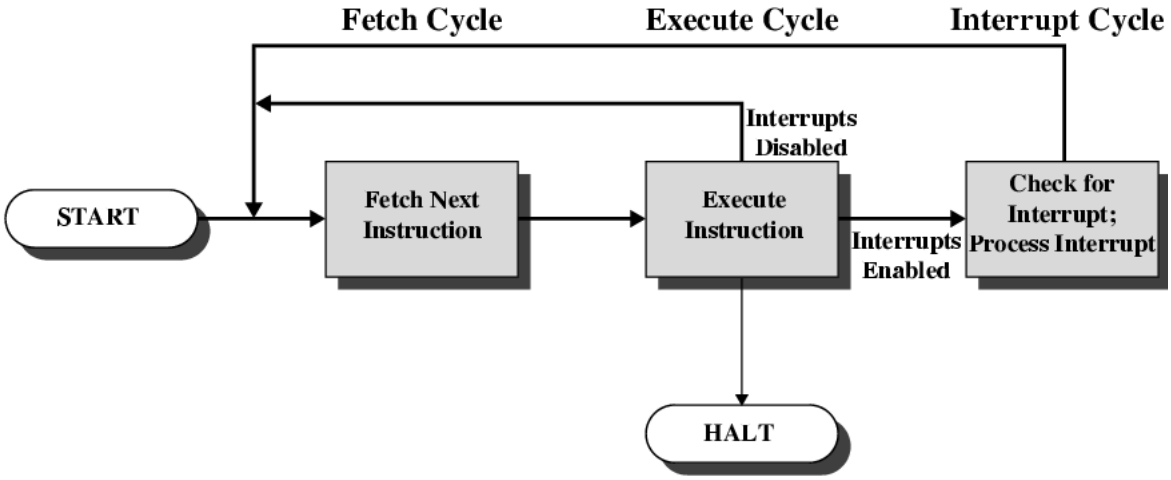
Trasferimento del controllo per una interruzione



Ciclo di interruzione

- Aggiunto al ciclo di esecuzione
- La CPU controlla se ci sono interruzioni pendenti
- Se no, prende la prossima istruzione
- Se si:
  - Sospende l'esecuzione del programma corrente
  - Salva il contesto (es.: indirizzo prossima istruzione)
  - Imposta il PC all'indirizzo di inizio del programma di gestione dell'interruzione
  - Esegue il programma di gestione dell'interruzione
  - Rimette il contesto al suo posto e continua il programma interrotto

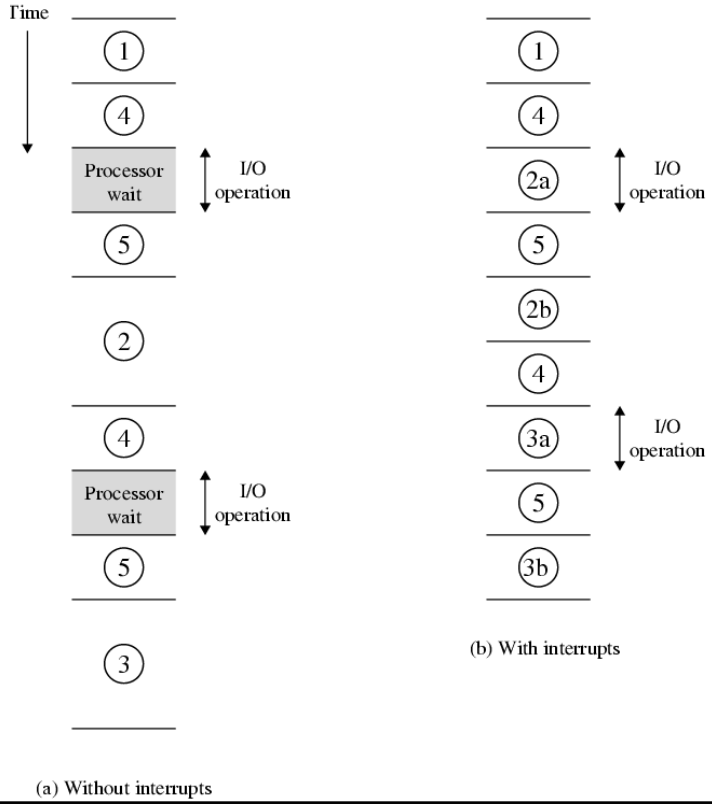
Ciclo di esecuzione con interruzioni



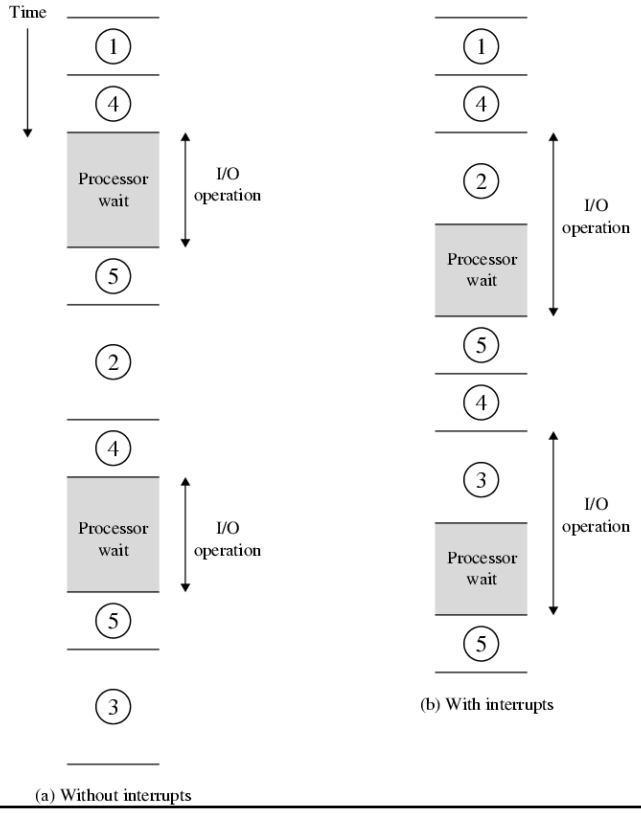
Breve attesa di I/O

Tempo per operazione di I/O minore del tempo tra due istruzioni WRITE

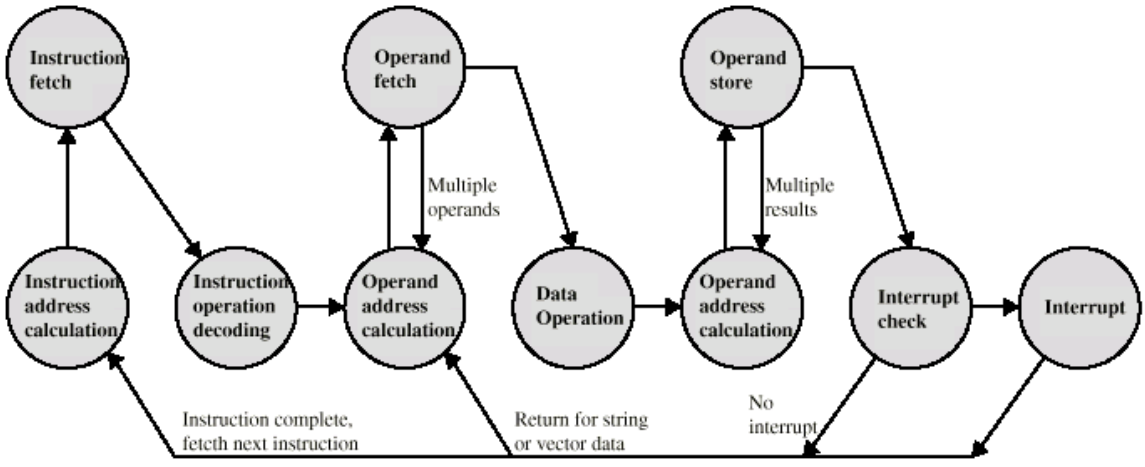
Le interruzioni eliminano le attese



Lunga attesa di I/O



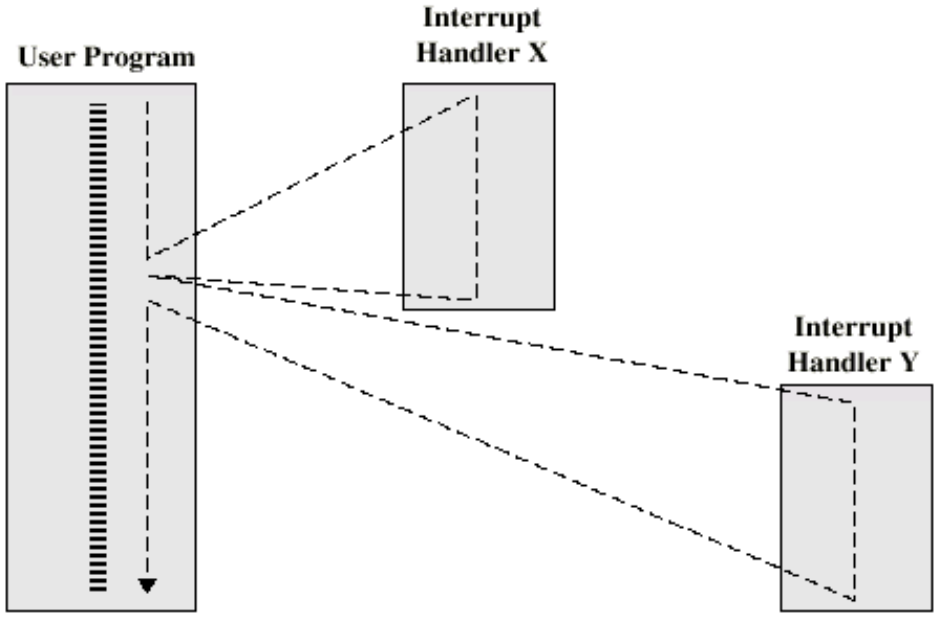
Ciclo con interruzioni



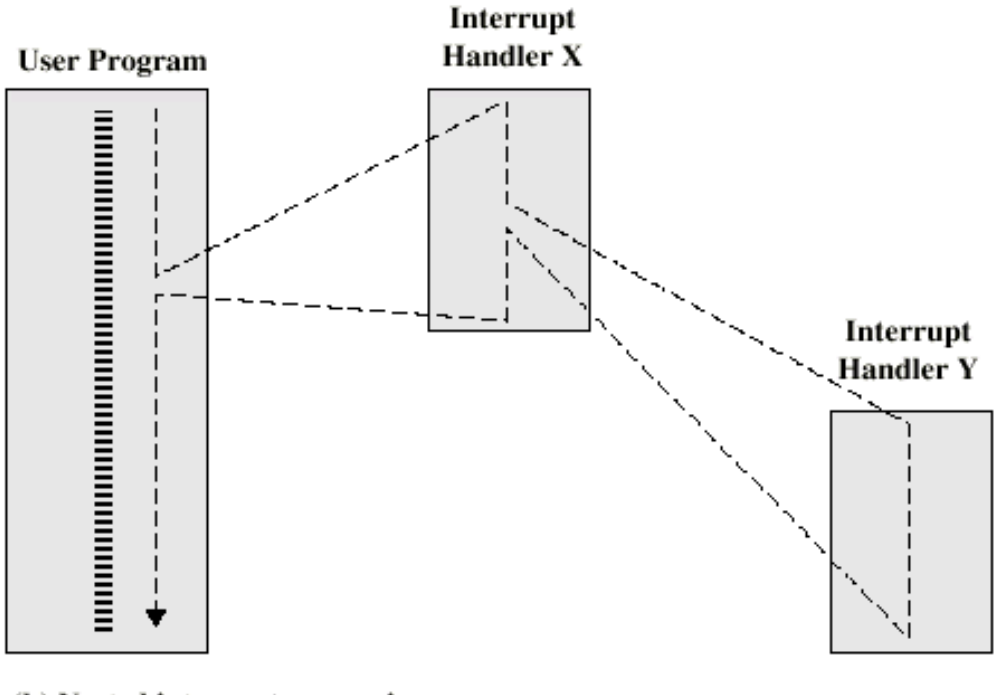
Interruzioni multiple

- Disabilitare le interruzioni
  - La CPU ignorerà altre interruzioni mentre gestisce la prima
  - Le interruzioni rimangono pendenti e sono controllate solo dopo che la prima è stata gestita completamente
  - Interruzioni gestite nella sequenza in cui sono richieste
- Definire delle priorità
  - Interruzioni con bassa priorità possono essere interrotte da interruzioni con priorità più alta
  - Quando l'interruzione con priorità più alta è stata gestita, la CPU ritorna all'interruzione precedente

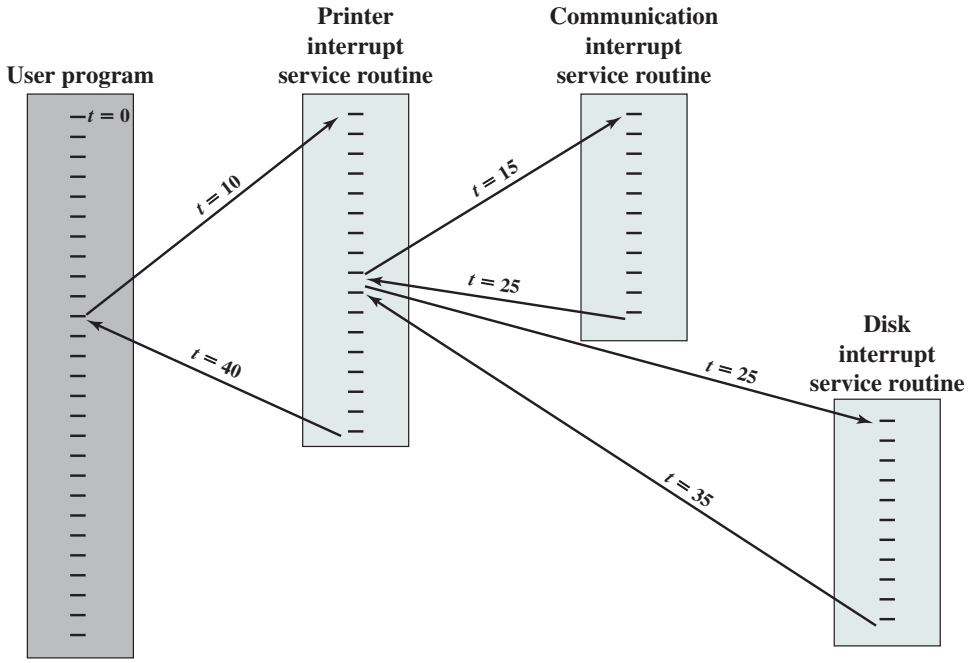
Interruzioni multiple – sequenziali



Interruzioni multiple – annidate

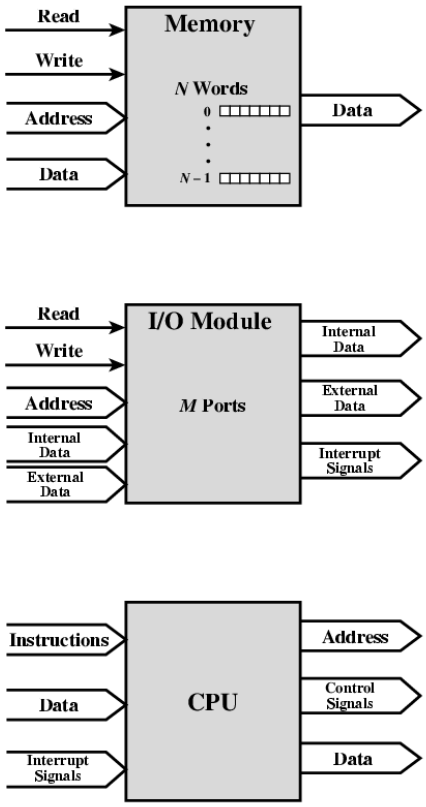


Esempio di Interruzioni



Connessioni

- Tutte le componenti di un calcolatore devono essere connesse
- Tipi diversi di connessione per diversi tipi di componente
  - Memoria
  - Input/Output
  - CPU



Connessioni per la memoria

- Riceve e spedisce dati (scrittura e lettura)
- Riceve indirizzi (di locazioni di M)
- Riceve segnali di controllo
  - Lettura
  - Scrittura

Connessioni dell' Input/Output (1)

- Modulo di I/O: simile ad una memoria dal punto di vista della CPU
- Operazioni di Output
  - Riceve dati dalla CPU
  - Manda dati alle periferiche
- Operazioni di Input
  - Riceve dati dalle periferiche
  - Manda dati alla CPU

Connessioni dell'Input/Output (2)

- Riceve segnali di controllo dalla CPU
- Manda segnali di controllo alle periferiche
- Riceve indirizzi dalla CPU (n.ro di porta per identificare una periferica)
- Manda segnali di interruzione

Connessioni per la CPU

- Legge istruzioni e dati
- Scrive dati (dopo l'elaborazione)
- Manda segnali di controllo alle altre unità
- Riceve segnali di interruzione

Connessioni

- Da M a CPU: la CPU legge un'istruzione o un dato dalla M
- Da CPU a M: la CPU scrive un dato in M
- Dall'I/O alla CPU: la CPU legge i dati di una periferica
- Dalla CPU all'I/O: la CPU invia dati ad una periferica
- Dall'I/O alla M o viceversa: accesso diretto alla M da parte di un dispositivo di I/O

Bus

- Collega due o più dispositivi
- Mezzo di trasmissione condiviso
- Un segnale trasmesso da uno dei dispositivi collegati ad un bus è disponibile a tutti gli altri
- Solo un dispositivo alla volta può trasmettere, altrimenti i segnali si sovrappongono
- Più linee di comunicazione, ogni linea trasmette uno 0 o un 1
- Insieme, più linee trasmettono in parallelo numeri binari
  - Esempio: dato da 8 bit tramesso in parallelo da un bus a 8 bit

Bus di sistema

- Connette CPU, I/O, M
- Da 50 a qualche centinaio di linee (ampiezza del bus)
- Tre gruppi di linee
  - Dati: su cui viaggiano i dati (bus dati)
  - Indirizzi
  - Controllo

Bus dati

- Trasporta i dati (o le istruzioni)
- L'ampiezza è importante per l'efficienza del sistema
  - Se poche linee, più accessi in M per prendere un dato

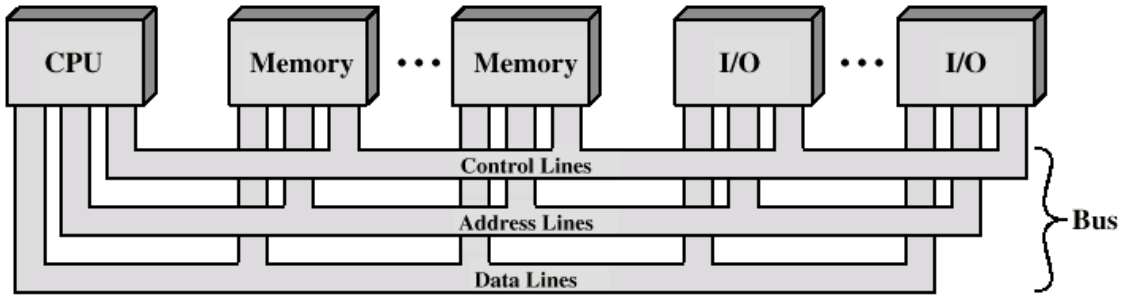
Bus indirizzi

- Indica la sorgente o la destinazione dei dati
  - Es.: la CPU vuole leggere un dato dalla M
- L'ampiezza determina la massima quantità di M indirizzabile

Bus di controllo

- Per controllare accesso e uso delle linee dati e indirizzi
  - M write: scrittura dei dati sul bus alla locazione di M
  - M read: mette sul bus i dati della locazione di M
  - Richiesta bus: un modulo vuole il controllo del bus
  - Bus grant: è stato concesso il controllo ad un modulo
  - Interrupt request: c'è una interruzione pendente
  - Clock: per sincronizzare le operazioni

Schema di interconnessione a bus



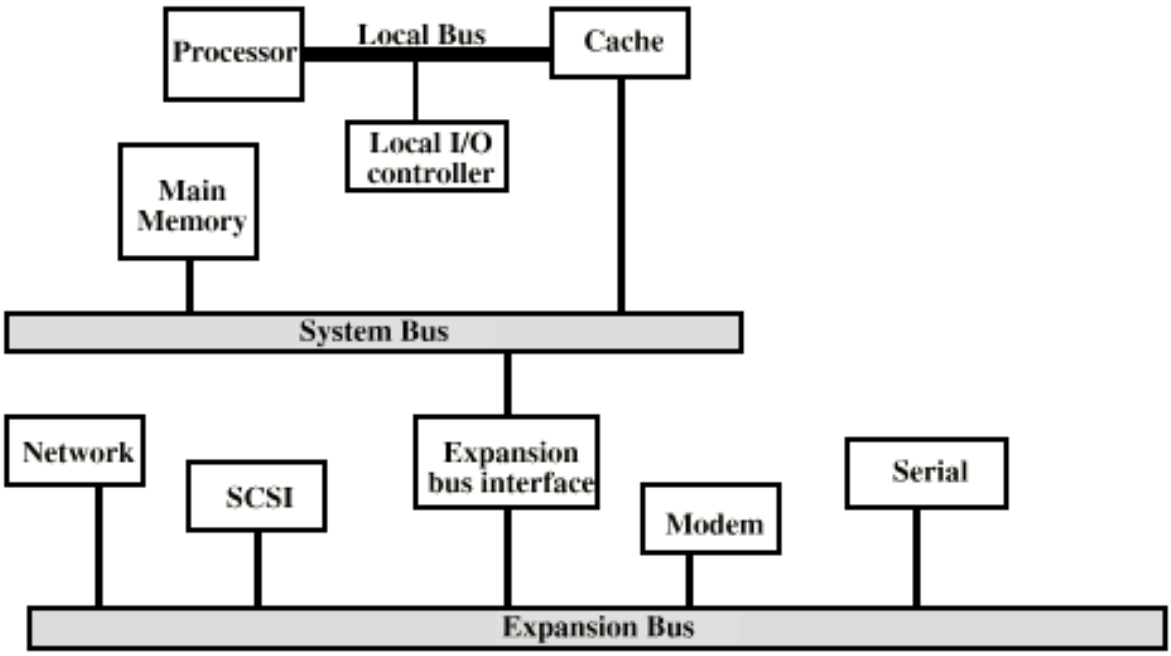
Uso del bus

- Se un modulo vuole inviare dati ad un altro, deve:
  - Ottenere l'uso del bus
  - Trasferire i dati sul bus
- Se un modulo vuole ricevere dati da un altro modulo, deve:
  - Ottenere l'uso del bus
  - Trasferire una richiesta all'altro modulo sulle linee di controllo
  - Attendere l'invio dei dati

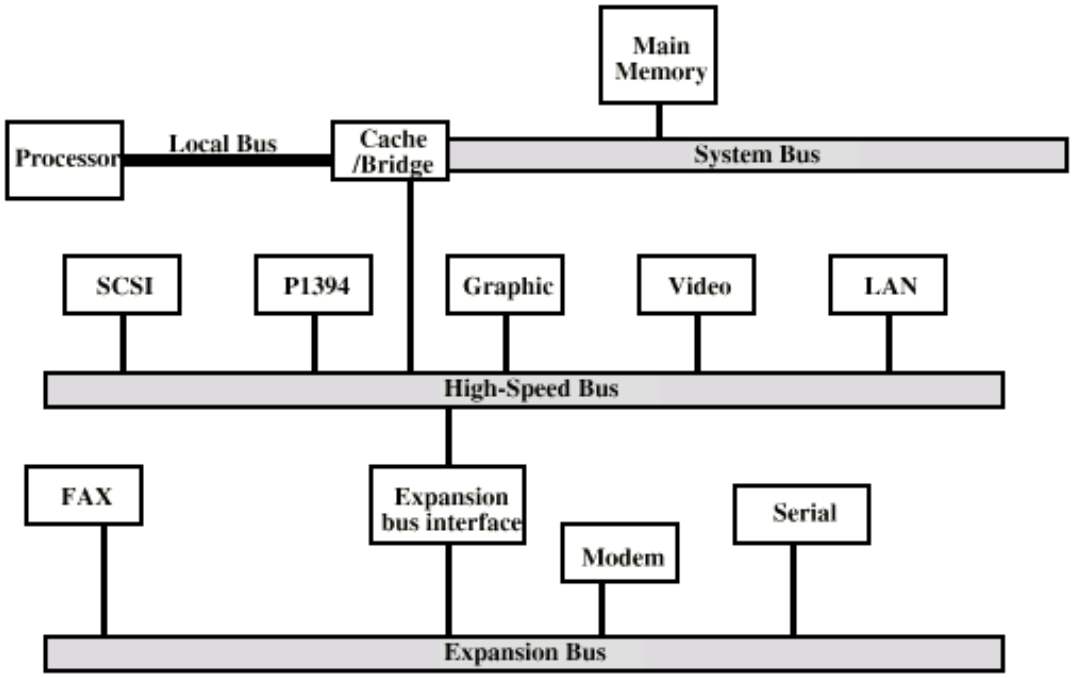
Bus singoli e multipli

- Se un solo bus, possibilità di ritardo e congestione
- Molti sistemi usano più bus per risolvere questi problemi

Bus multipli



Bus multipli



Temporizzazione

- Coordinazione degli eventi su un bus
- Sincrona
  - Eventi determinati da un clock
  - Una linea di clock su cui viene spedita una sequenza alternata di 0 e 1 di uguale durata
  - Una singola sequenza 1-0 è un ciclo di clock
  - Tutti i dispositivi connessi al bus possono leggere la linea di clock
  - Tutti gli eventi partono dall'inizio di un ciclo di clock

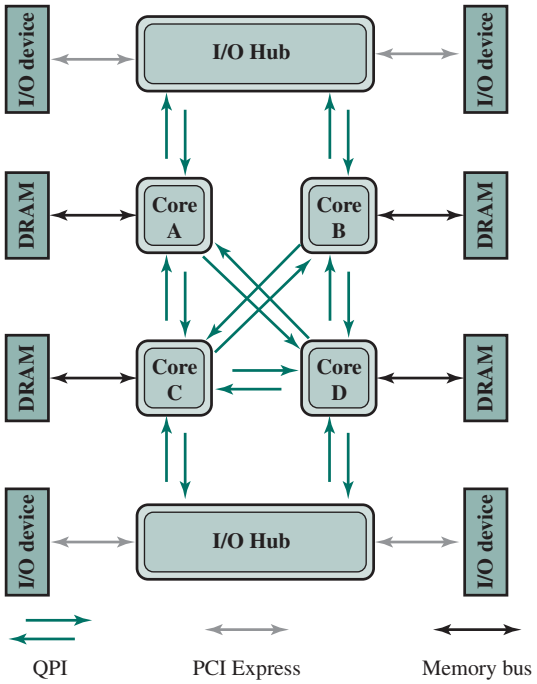
Interconnessione punto a punto

Connessioni dirette multiple: più componenti all'interno del sistema godono di connessioni dirette a coppie con altri componenti

Architettura di protocollo a strati/livelli (layer): come si trova negli ambienti di rete

Trasferimento dati a pacchetto: i dati non vengono inviati come flusso di bit non elaborato ma inviati come una sequenza di pacchetti, ognuno dei quali include intestazioni di controllo e codici di controllo degli errori.

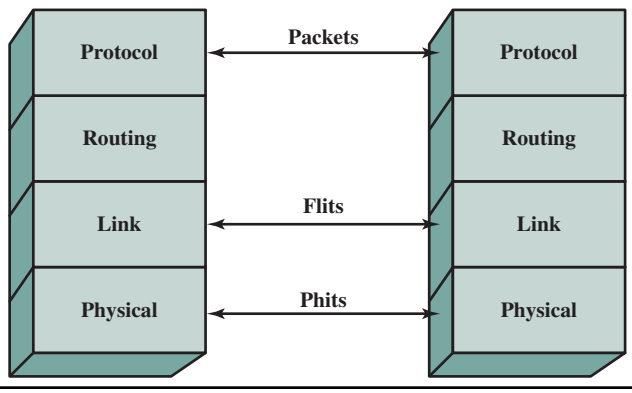
QuickPath Interconnect (QPI)



Livelli QPI

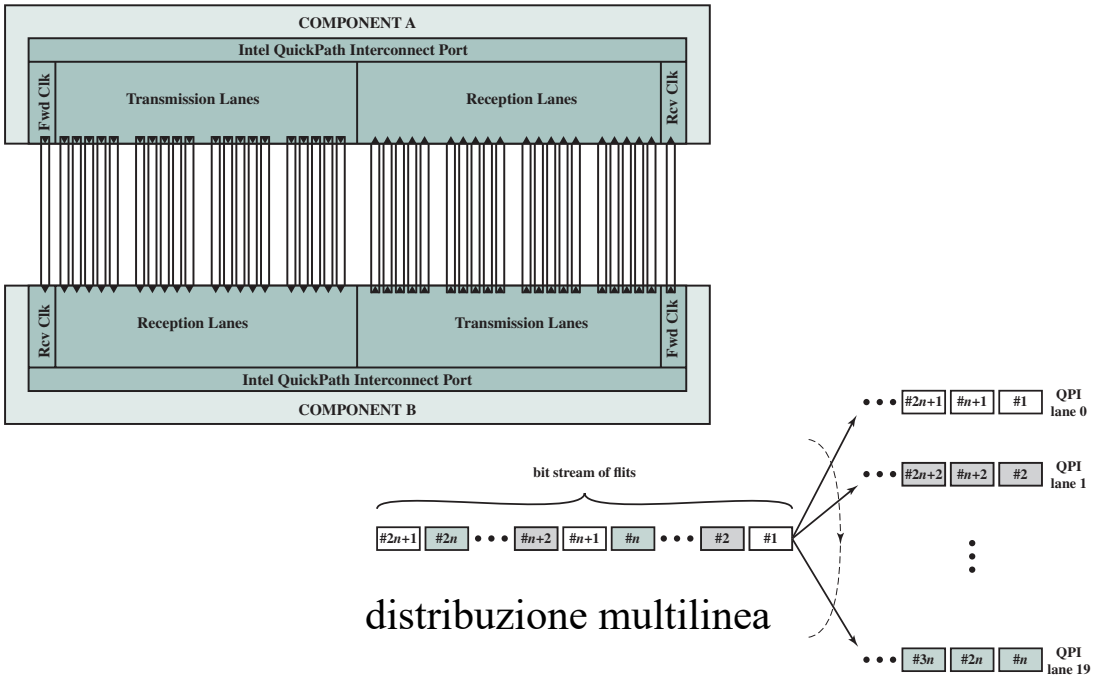
Fisico: è costituito dai cavi, circuiti e logica per supportare le funzioni ausiliarie per trasmissione e ricezione bit. L'unità di trasferimento a livello fisico è di 20 bit, chiamata Phit (unità fisica).

- **Link:** responsabile della trasmissione affidabile e del controllo del flusso. L'unità di trasferimento è un Flit (unità di controllo del flusso) a 80 bit.
- **Routing:** fornisce la struttura per dirigere i pacchetti attraverso la struttura.
- **Protocollo:** l'insieme di regole di alto livello per lo scambio di pacchetti di dati tra dispositivi. Un pacchetto è composto da un numero intero di Flits.



Livello Fisico

connessioni



distribuzione multilinea

Livello Link

Usa un protocollo con pacchetti da 72 (dati/messaggi) + 8 (codice correzione di errore) bit

Due funzioni:

- **controllo del flusso**: evita che il mittente invvi più dati di quanti il destinatario possa ricevere (sistema a crediti)

- **controllo dell'errore**: 8 bit sono utilizzati per rilevare errori di trasmissione sui 72 bit di dati/messaggi (vedremo in seguito come funziona); in caso di errore il mittente deve re-inviare il pacchetto con l'errore (e altri successivamente inviati)

Livello Routing (instradamento)

Determina il percorso che un pacchetto deve seguire all'interno del sistema

Supportato da:

- **Tabelle di instradamento**:
  - definite dal firmware;
  - descrivono i possibili percorsi che un pacchetto può seguire;
  - utile soprattutto in sistemi di dimensione maggiore;

Livello Protocollo

Pacchetto definito come unità di trasferimento

Caratteristiche:

- definizione contenuto pacchetto flessibile, in modo da coprire esigenze diverse;
- supporta protocollo di coerenza della cache, in modo da garantire coerenza fra i contenuti delle cache dei core e la memoria principale;