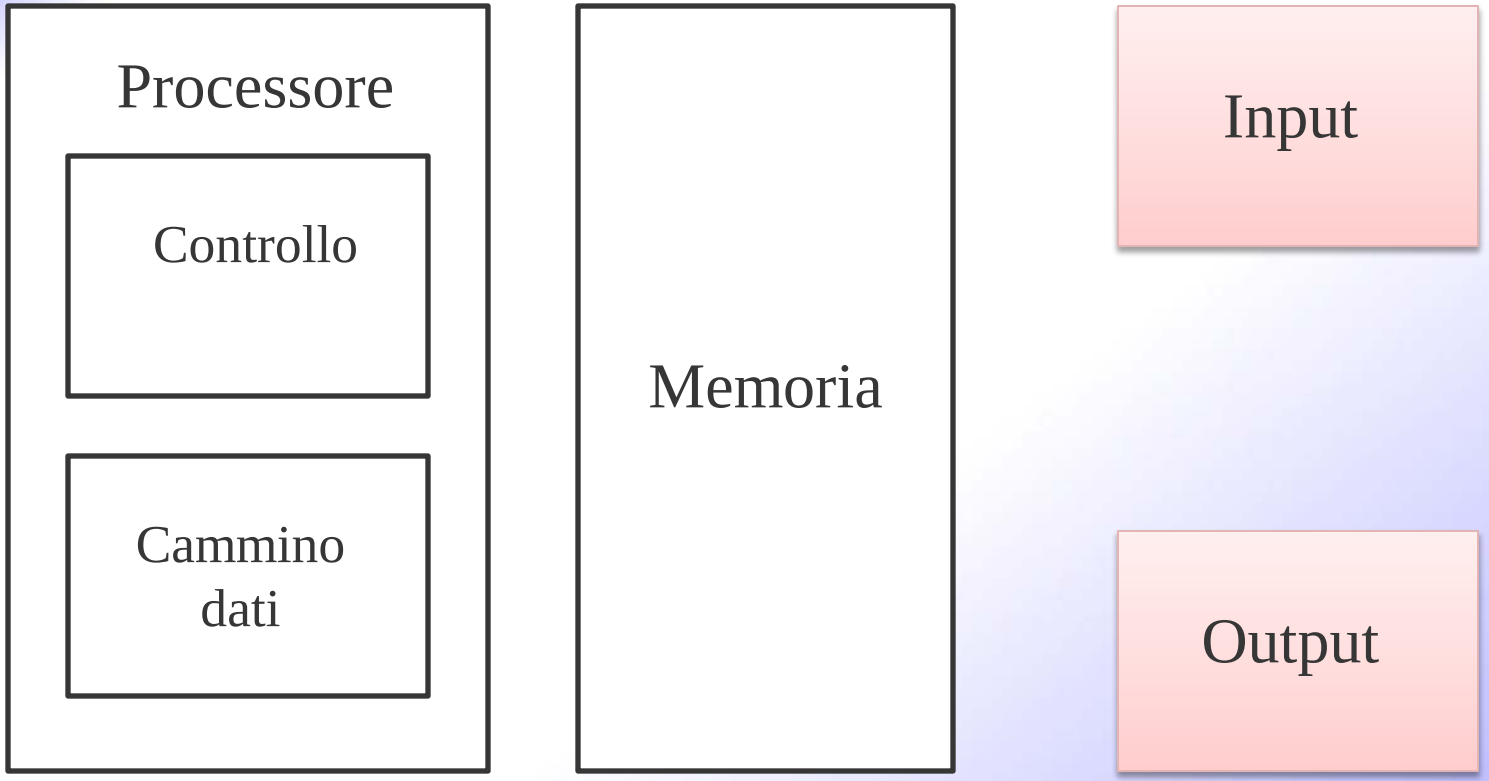
L'interfaccia tra processori e periferiche

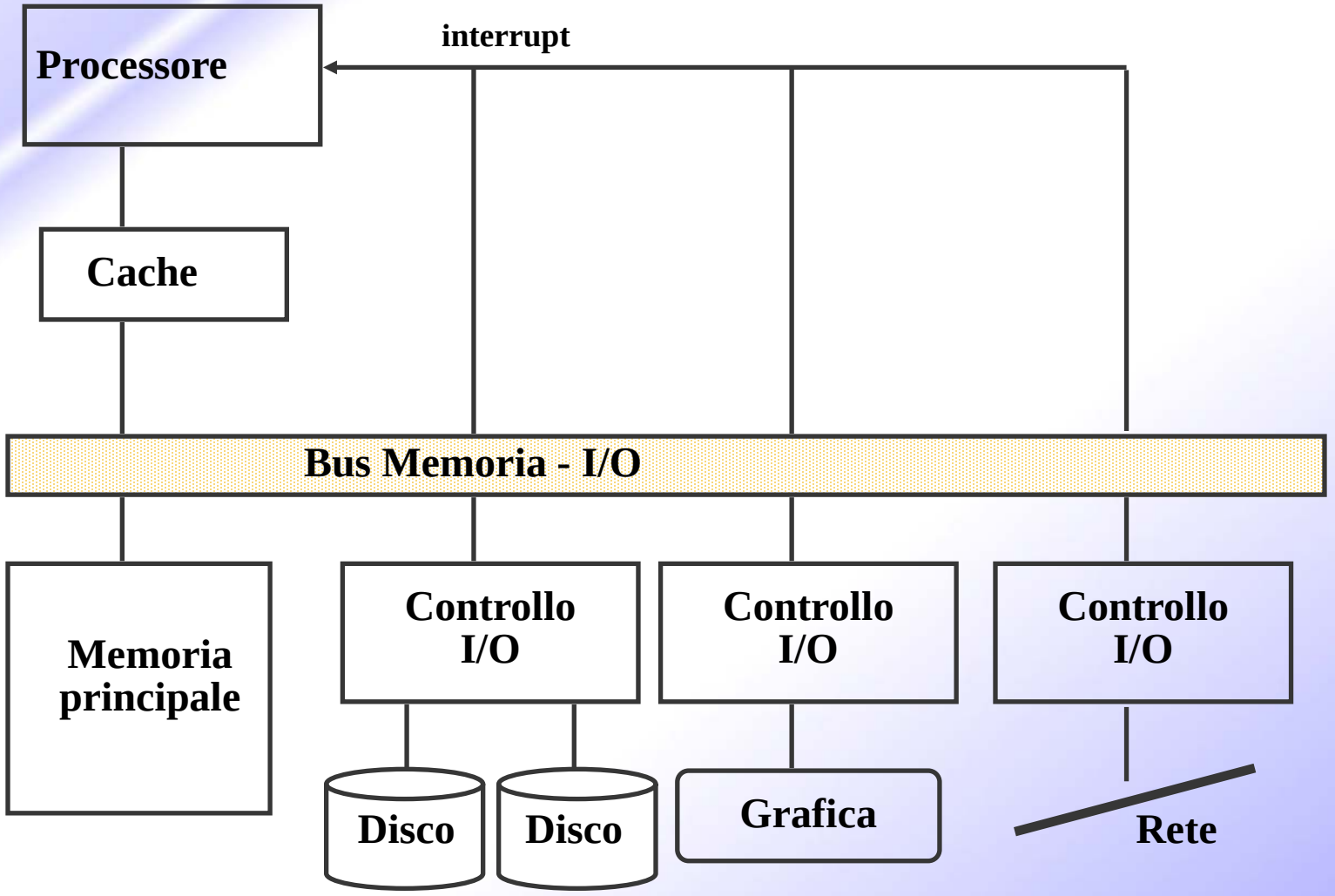
Il punto



Problematiche

- Progetto di sistemi I/O condizionato da diversi fattori
  - espandibilità
  - flessibilità in caso di guasti
  - prestazioni
    - più complesso (ad esempio, scelta tra latenza e banda)
    - dipende da molti aspetti del sistema
      - caratteristiche dei dispositivi, tipo di connessione, gerarchia di memoria e così via
  - varietà di utenti

Sistema I/O tipico



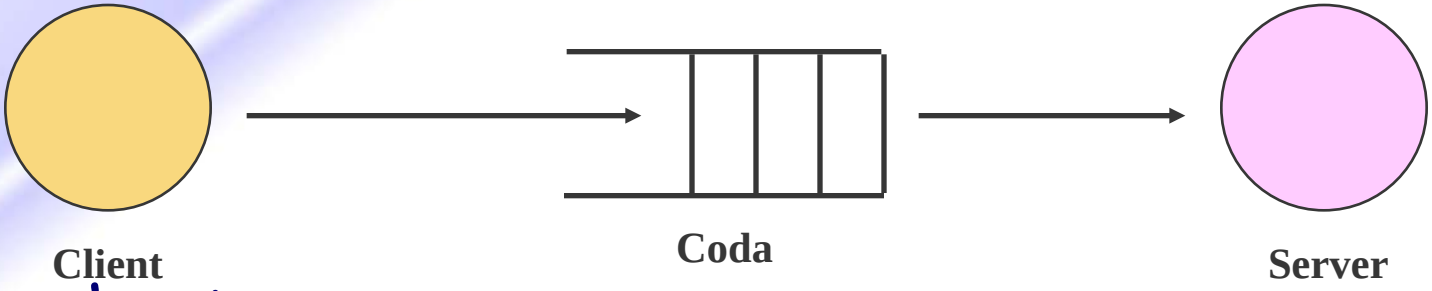
Impatto di I/O su prestazioni

- Legge di Amdhal
  - Benchmark con tempo esecuzione = 90 sec (CPU) + 10 sec (I/O)
    - tempo I/O = 10% tempo totale
  - CPU migliora del 50% ogni anno
  - I/O rimane invariato
  - Dopo 5 anni, tempo di esecuzione = 12 sec (CPU) + 10 sec (I/O)
    - tempo I/O = 45% tempo totale

Prestazioni di un sistema I/O

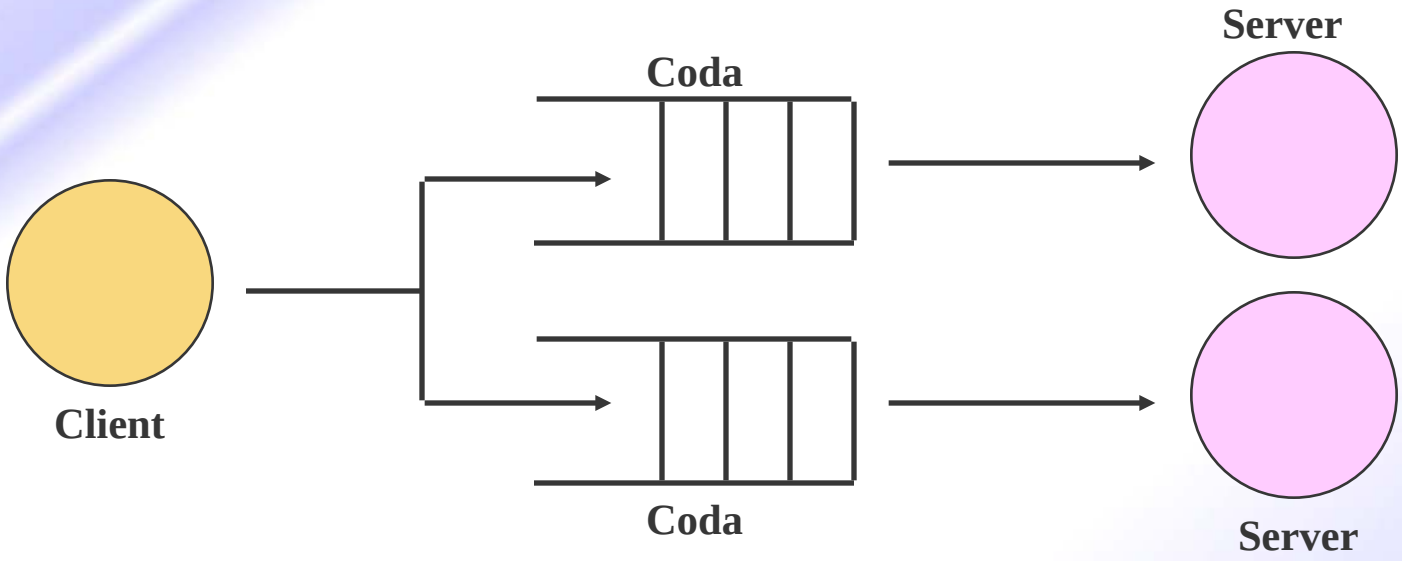
- Dipende da molti aspetti del sistema:
  - CPU
  - Sistema di memoria:
    - Cache interna ed esterna
    - Memoria principale
  - Connessioni soggiacenti (bus)
  - Controllo dell'I/O
  - Dispositivo I/O
  - Velocità del software I/O (sistema operativo)
  - Efficienza di utilizzo del software dei dispositivi I/O
- Due metriche comuni:
  - Throughput: banda I/O
  - Tempo di risposta: latenza

Semplice modello client-server



- Throughput:
  - Il numero di compiti completati dal server nell'unità di tempo
  - Per avere il massimo throughput possibile:
    - Il server non deve mai essere indisponibile
    - La coda non deve mai essere vuota
- Tempo di risposta:
  - Inizia quando un compito è messo in coda
  - Finisce quando è completato dal server
  - Per minimizzare il tempo di risposta:
    - La coda dovrebbe essere vuota
    - Il server si renderà indisponibile ad altre richieste durante l'esecuzione del compito

Miglioramento del throughput

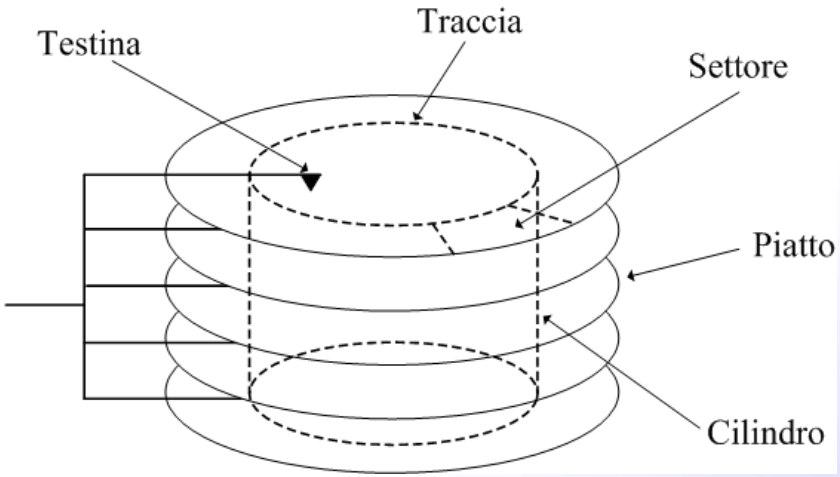


- In generale, il throughput può essere migliorato dedicando più hardware al problema
- Tempo di risposta più difficile da ridurre

Esempi di dispositivi I/O (Velocità CPU ≈ 2.6 GHz – 3.6 GHz)

Dispositivo	Comportamento	Partner	Frequenza dati (Mbit/sec)
Tastiera	Input	Uomo	0,0001
Mouse	Input	Uomo	0,0038
Input vocale	Input	Uomo	0,2640
Input sonoro	Input	Macchina	3,0000
Scanner	Input	Uomo	3,2000
Output vocale	Output	Uomo	0,2640
Output sonoro	Output	Uomo	8,0000
Stampante Laser	Output	Uomo	3,2000
Display grafico	Output	Uomo	800,0000-8000,0000
Modem	Input o output	Macchina	0,0160-0,0640
Rete/LAN	Input o output	Macchina	100,0000-1000,0000
Rete/LAN wireless	Input o output	Macchina	11,0000-54,0000
Disco ottico	Storage	Macchina	80,0000
Nastro magnetico	Storage	Macchina	32,0000
Disco magnetico	Storage	Macchina	240,0000-2560,0000

Dischi magnetici



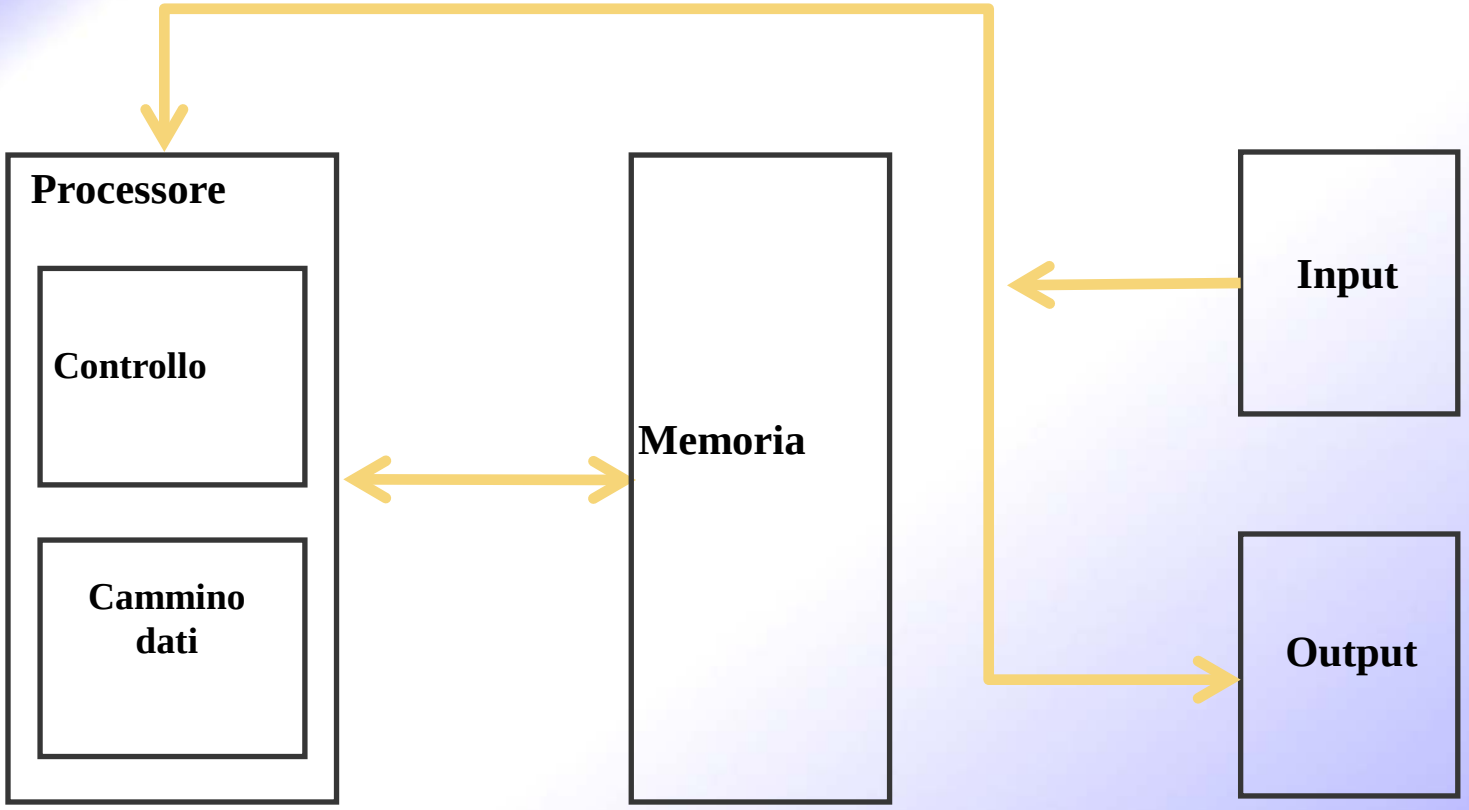
- Lettura/scrittura in tre fasi:
  - Tempo di ricerca: posiziona testina su traccia giusta
  - Latenza rotazionale: aspetta che il settore desiderato ruoti sotto la testina
  - Tempo di trasferimento: trasferisce il blocco di bit (settore) sotto la testina
- Tempo di ricerca medio riportato dall'industria:
  - Tipicamente tra 8 ms e 12 ms
  - Somma tempo tutte possibile ricerche/N. possibili ricerche
  - Per principi di località, tempo medio reale di ricerca solo 25%-33% di quello riportato

Prestazioni disco I/O

- Tempo di accesso al disco =
  - Tempo di ricerca + Latenza rotazionale + Tempo di trasferimento + Tempo controllo + Ritardo di coda
- Esempio:
  - Settore di 512 byte, 5400 RPM, tempo medio di ricerca 12 ms, tasso di trasferimento 5 MB/sec, tempo controllo 2 ms, nessun tempo di servizio
  - Tempo di accesso al disco =
    12ms + 0.5 / 5400RPM + 0.5KB / 5MB/s + 2ms + 0ms =
    = 12ms + 5.6ms + 0.1ms + 2ms + 0ms = 19.7ms
- Se il tempo reale medio di ricerca è 1/4 di quello riportato, allora 10.7ms
  - ritardo rotazionale pari al 50% di tutto il tempo

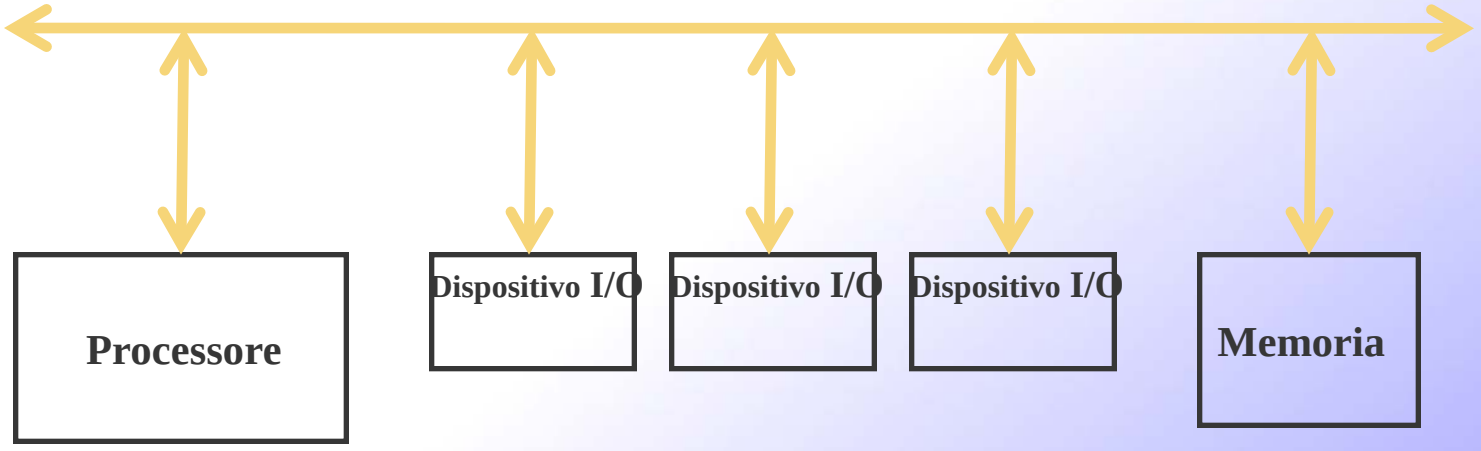
Bus

- Canale di comunicazione condiviso
- Singolo insieme di cavi usato per connettere più sottosistemi



Vantaggi dei bus

- Versatilità:
  - Nuovi dispositivi possono essere aggiunti facilmente
  - Periferiche possono essere spostate tra sistemi di computer che usano lo stesso standard di bus
- Basso costo:
  - un singolo insieme di cavi è condiviso in più modi

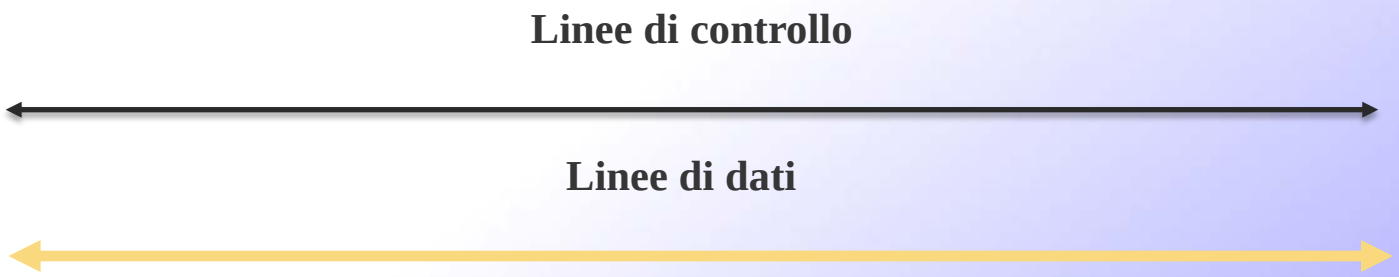


Svantaggi dei bus

- Crea un collo di bottiglia nella comunicazione
  - La banda del bus può limitare il massimo throughput di I/O
- La velocità massima di un bus è soprattutto limitata da:
  - Lunghezza del bus
  - Numero di dispositivi sul bus
- La necessità di supportare un insieme di dispositivi con:
  - Latenze molto differenti
  - Tassi di trasferimento dati molto diversi

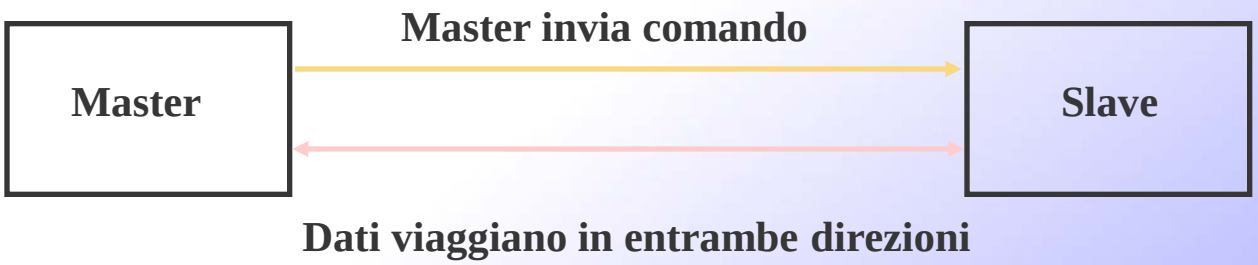
Organizzazione generale

- Linee di controllo:
  - Segnali di richiesta e di ricevuta
  - Indicano quale tipo di informazione è sulle linee di dati
- Linee di dati:
  - Dati ed indirizzi
  - Comandi complessi



Schema master-slave

- Una transazione include due parti:
  - Inviare il comando (ed indirizzo) – **richiesta**
  - Trasferire i dati – **azione**
- Il master inizia la transazione:
  - Inviando il comando (ed indirizzo)
- Lo slave risponde alla richiesta:
  - Inviando i dati al master se il master richiede dati
  - Ricevendo dati dal master se il master vuole inviare dati

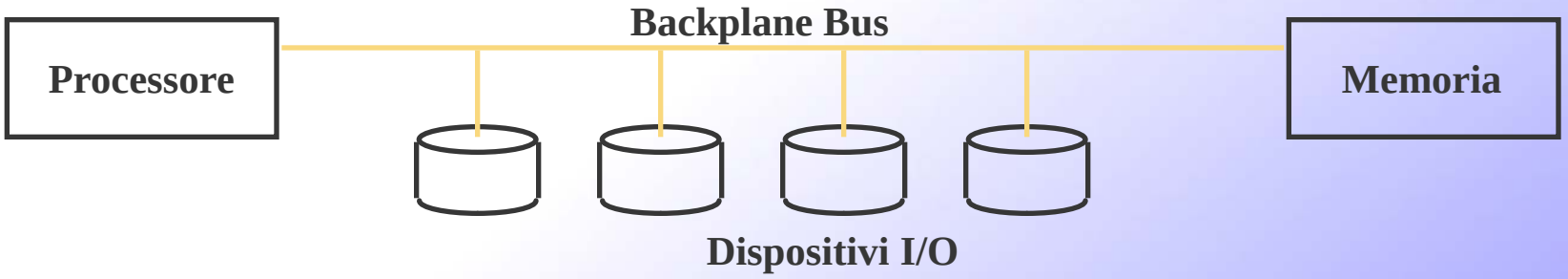


Tipi di bus

- Bus processore-memoria (per uno specifico calcolatore)
  - corti ed ad alta velocità
  - cercano di adattarsi al sistema di memoria
    - Massimizzare la banda dalla memoria al processore
  - si connettono direttamente al processore
  - **ottimizzati per trasferimenti di blocchi di cache**
- Bus generico di sistema (standard oppure proprietario)
  - backplane: struttura di interconnessione nel piano posteriore dello chassis
  - permettono a processori, memoria e dispostivi di I/O di coesistere
  - vantaggio economico: un bus per tutte le componenti
- Bus di I/O (standard)
  - generalmente lunghi e più lenti
  - devono adattarsi a un insieme ampio di dispositivi di I/O
  - si connettono al bus processore-memoria o al bus generico di sistema

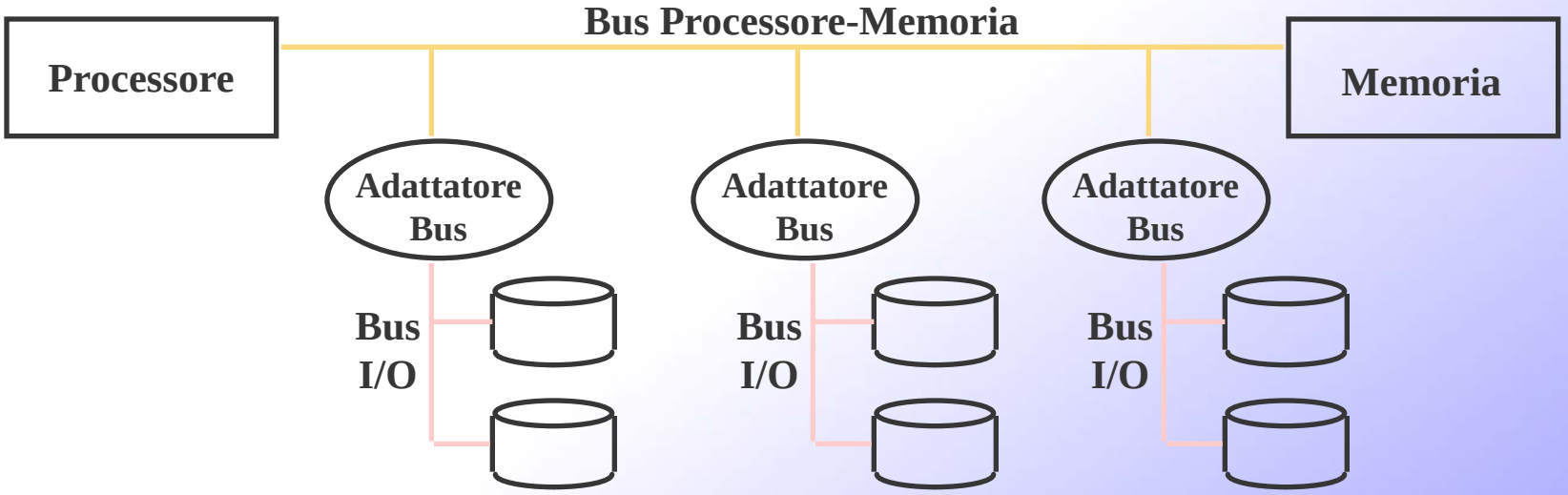
Sistema con un backplane bus

- Un solo bus (backplane bus) è usato per:
  - comunicazione tra processore e memoria
  - comunicazione tra memoria e dispositivi di I/O
- Vantaggi:
  - semplice ed a basso costo
- Svantaggi:
  - lento ed il bus può diventare il collo di bottiglia principale
    - I dispositivi di I/O sono molto più lenti della memoria



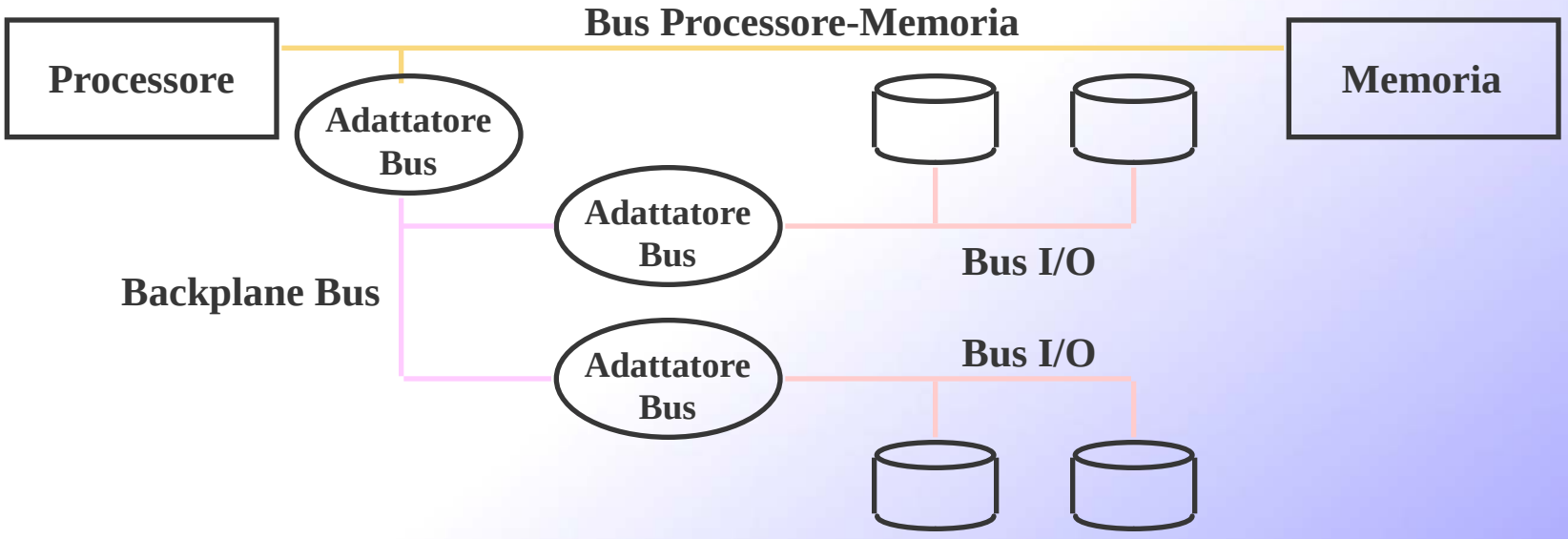
Sistema a due bus

- Bus I/O si interfacciano al bus processore-memoria tramite adattatori di bus:
  - bus processore-memoria: bus dedicato per lo scambio di dati fra processore e memoria ad alta velocità
  - bus I/O: più lenti, forniscono slot di espansione per dispositivi I/O.



Sistema a tre bus

- Backplane bus connesso a bus processore-memoria
  - bus processore-memoria usato per traffico veloce processore-memoria
  - bus I/O connessi a backplane bus
- Vantaggio:
  - Si limitano le connessioni sul bus veloce processore-memoria



Caratteristiche di un bus

Protocollo di transazione

Specifiche temporizzazione e segnali

Gruppo di cavi

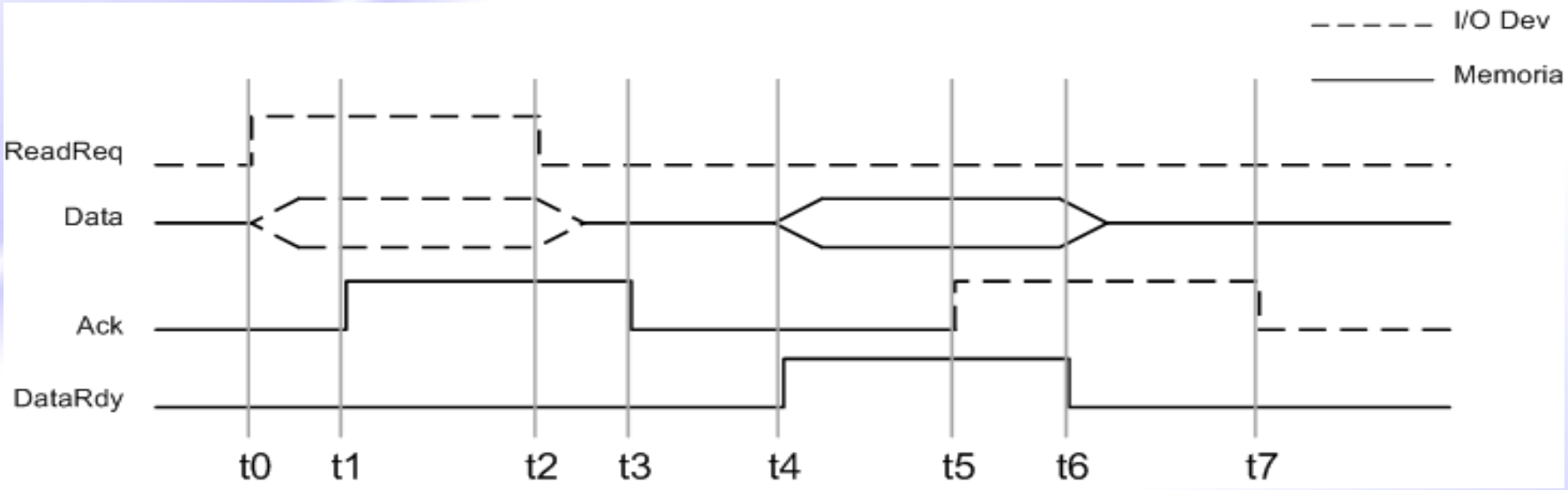
Specifiche elettriche

Caratteristiche fisico/meccaniche connettori

Bus sincroni e asincroni

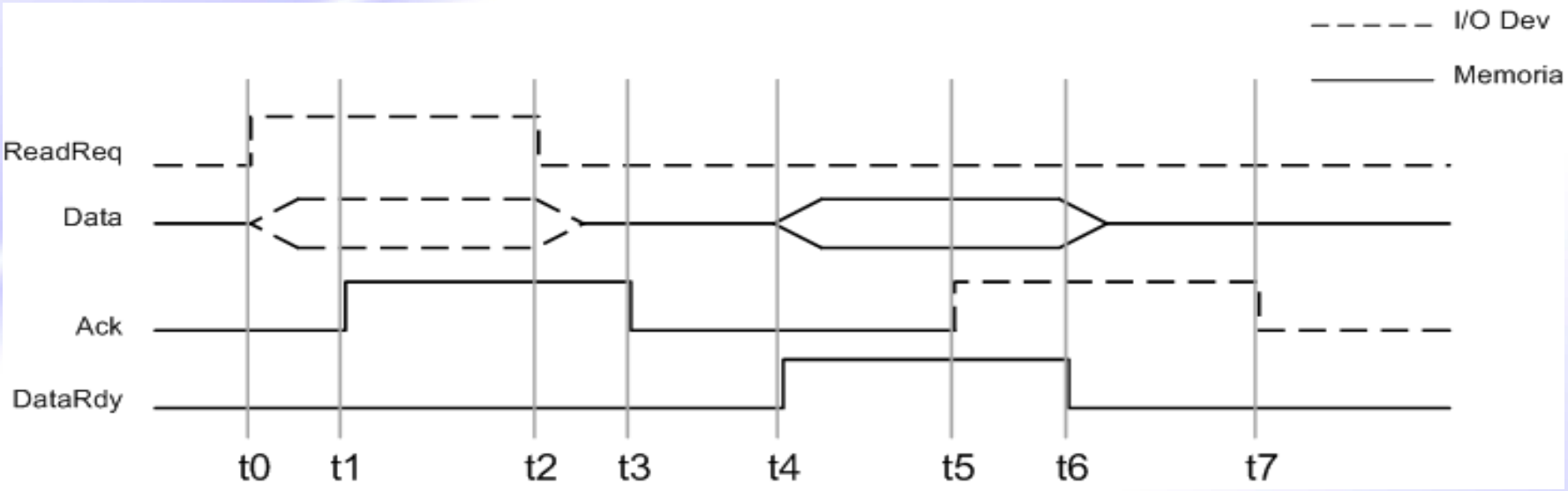
- **Bus sincrono:**
  - include un clock nelle linee di controllo
  - un protocollo per la comunicazione fissato e basato sul clock
  - vantaggio: poca logica e può essere molto veloce
  - svantaggi:
    - ogni dispositivo sul bus deve operare alla stessa frequenza di clock
    - devono essere corti, per essere veloci ed evitare sfasamento del clock
- **Bus asincrono:**
  - non è temporizzato
  - vantaggi:
    - può interfacciare un ampia gamma di dispositivi
    - non ha vincoli sulla lunghezza
  - svantaggi:
    - richiede un protocollo di accordo (*handshaking*) e quindi più lento

Protocollo asincrono di lettura (*Handshaking*)



- t0: Il protocollo inizia quando il dispositivo di I/O avanza una richiesta attivando il segnale ReadReq e mettendo l'indirizzo sulle linee Data;
- t1: la Memoria rileva ReadReq alto,
  → quindi legge l'indirizzo su Data e DOPO attiva Ack;
- t2: Il dispositivo I/O rileva Ack alto,
  → quindi disattiva ReadReq e rilascia Data;
- t3: la Memoria rileva ReadReq basso,
  → quindi disattiva Ack;

Protocollo asincrono di lettura (*Handshaking*)



- t4: la Memoria ha i dati pronti,
  → quindi immette i dati su Data e attiva DataReady;
- t5: Il dispositivo I/O rileva DataReady alto,
  → quindi legge i dati da Data e DOPO attiva Ack;
- t6: la Memoria rileva Ack alto,
  → quindi disattiva DataReady e rilascia Data;
- t7: il dispositivo I/O rileva DataReady basso,
  → quindi disattiva Ack.

Prestazioni

- Bus sincrono:
  - ciclo di clock 50ns ed un ciclo per transazione
  - tempo di lettura di una parola: 300ns
    - invio indirizzo: 50ns
    - lettura memoria: supponiamo 200ns
    - invio parola: 50ns
  - banda: 4byte/300ns = 13.3MB/sec
- Bus asincrono:
  - 40ns per stretta di mano
  - tempo di lettura di una parola: 360ns
    - Passo 1: 40ns
    - Passi 2, 3, 4 (parallelo a lettura): max(120ns, 200ns) = 200ns
    - Passi 5, 6, 7: 120ns
  - banda: 4byte/360ns = 11.1MB/sec

Accesso al bus

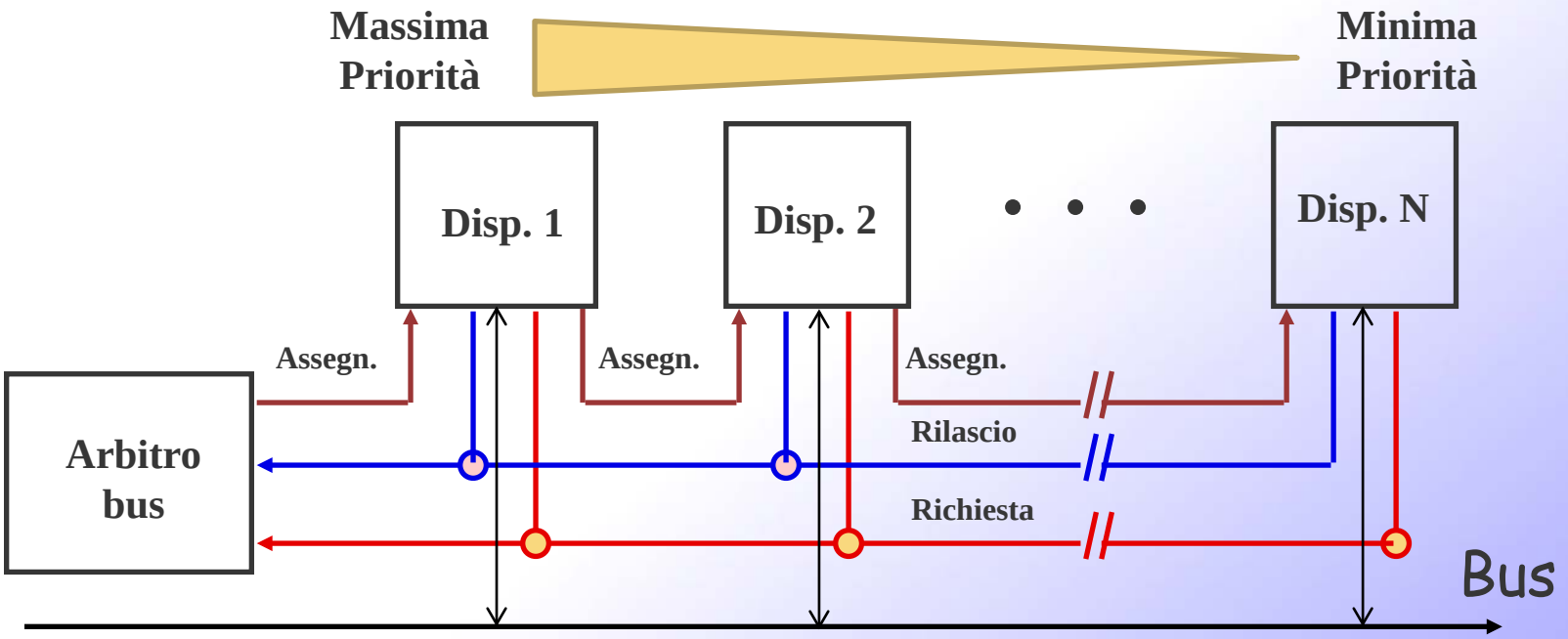
- Uno dei problemi principali:
  - come può un dispositivo che desidera usare il bus ottenere l'accesso?
- Caos è evitato da un accordo:
  - Solo il bus master può controllare l'accesso al bus:
    - inizia e controlla tutte le richieste di bus
  - Uno slave risponde a richieste di lettura e scrittura
- Sistema più semplice:
  - processore è il solo bus master
  - tutte le richieste di bus devono essere controllate dal processore
  - svantaggio: il processore è coinvolto in ogni transazione

Arbitraggio

- Più bus master implica utilizzo di schema di arbitraggio:
  - Un bus master che vuole usare il bus invia la richiesta di bus (*bus request*)
  - Un bus master non può usare il bus fino a che non riceve il segnale di assegnazione del bus (*bus grant*)
  - Un bus master deve informare l'arbitro quando ha finito di usare il bus
- Bilanciamento di due fattori:
  - *Priorità*: il dispositivo con priorità più alta va servito prima
  - *Fairness*: anche i dispositivi a bassa priorità devono essere serviti
- Quattro classi di schemi:
  - Daisy chain
  - Centralizzato e parallelo
  - Distribuito con auto-selezione
  - Distribuito con rilevamento di collisione: Ethernet

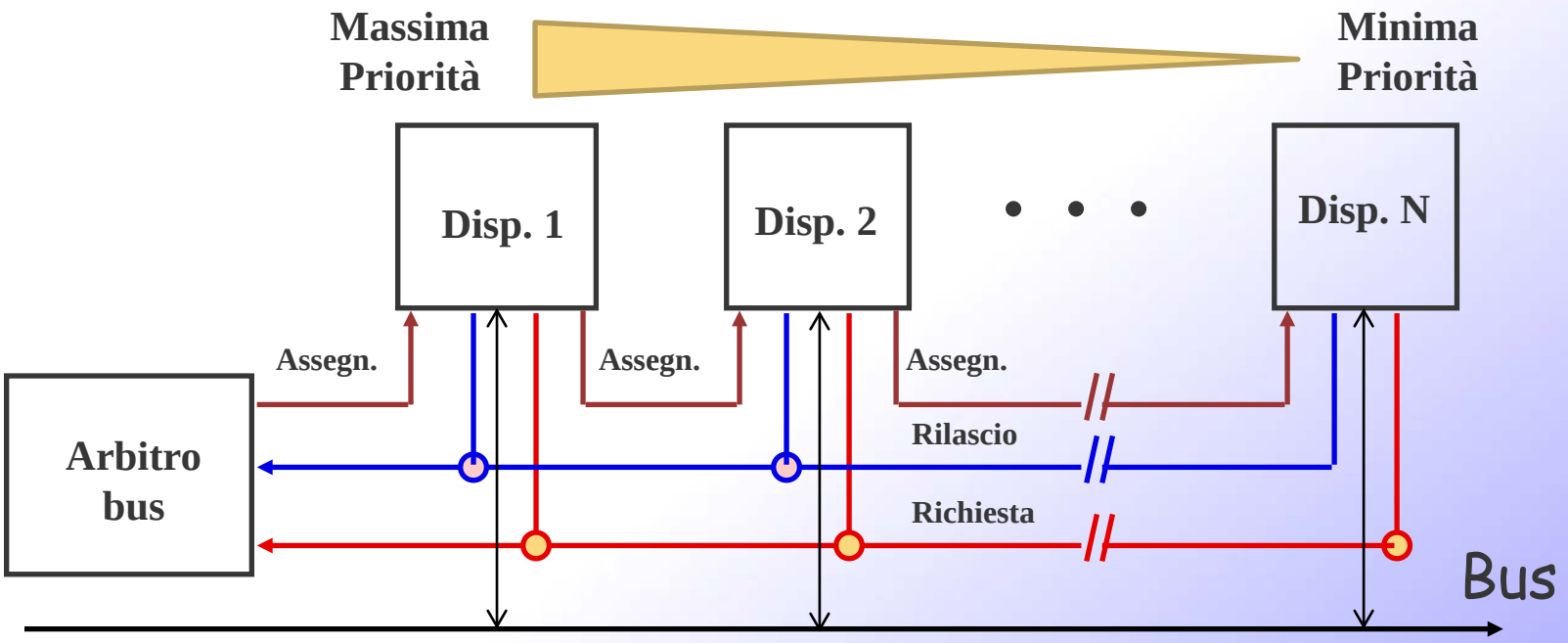
Daisy chain

- La linea di assegnazione passa da un dispositivo all'altro a partire da quello con priorità più alta
- Un dispositivo che desidera accedere al bus non fa altro che intercettare il segnale di assegnazione



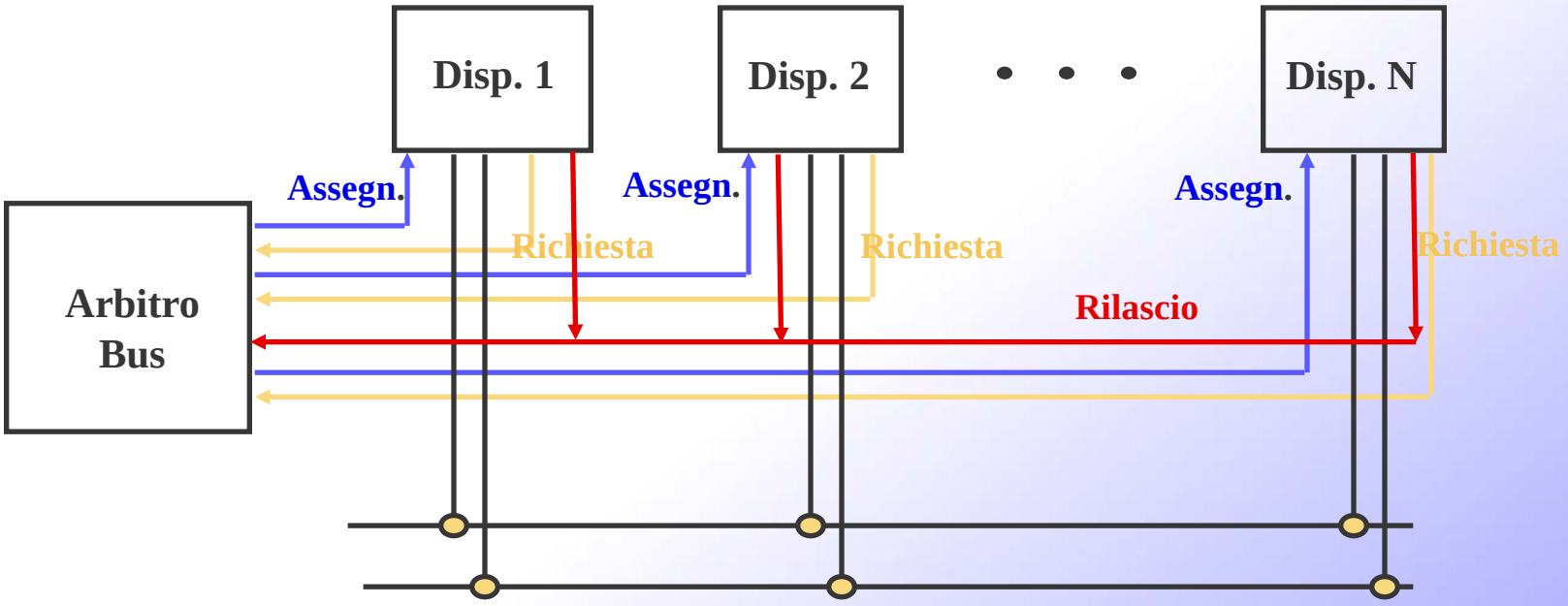
Daisy chain

- Vantaggio: semplice
- Svantaggi:
  - non assicura la fairness
  - limita la velocità del bus



Centralizzato e parallelo

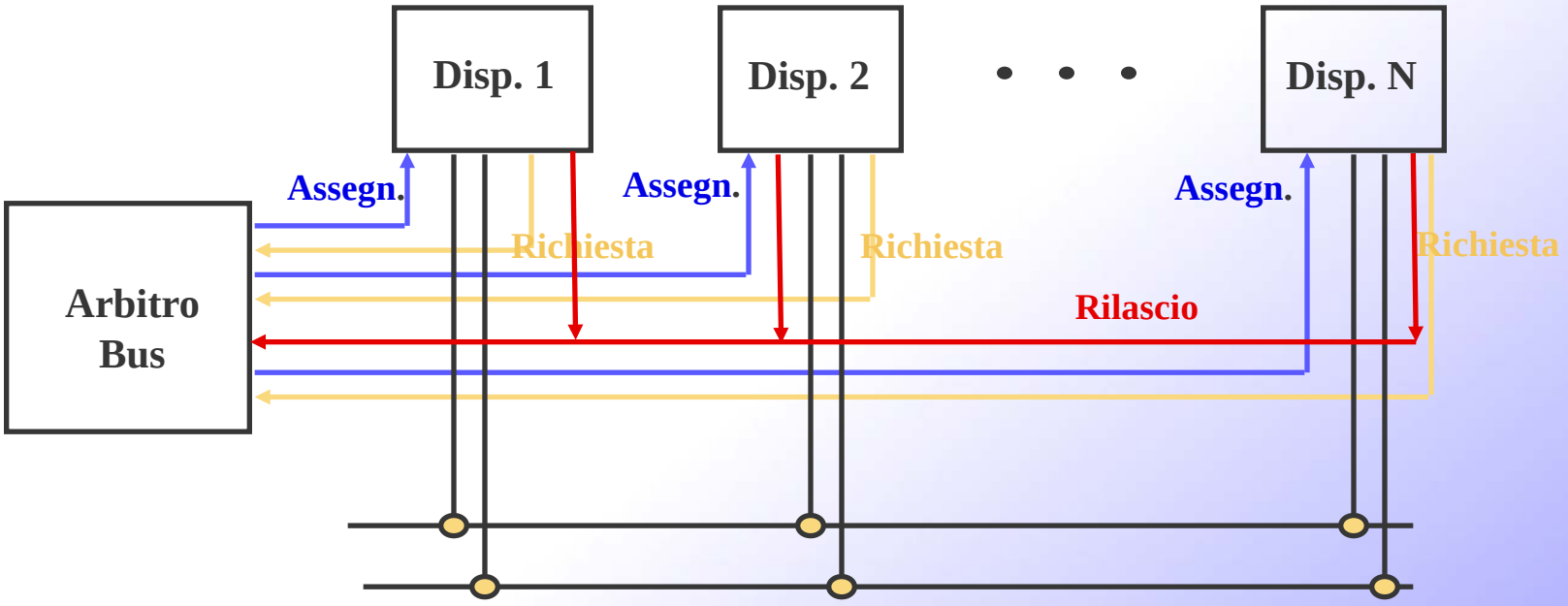
- Richieste di bus avanzate indipendentemente dai vari dispositivi sulle rispettive linee di richiesta
- L'arbitro centralizzato decide di volta in volta a quale dispositivo assegnare il bus



Centralizzato e parallelo

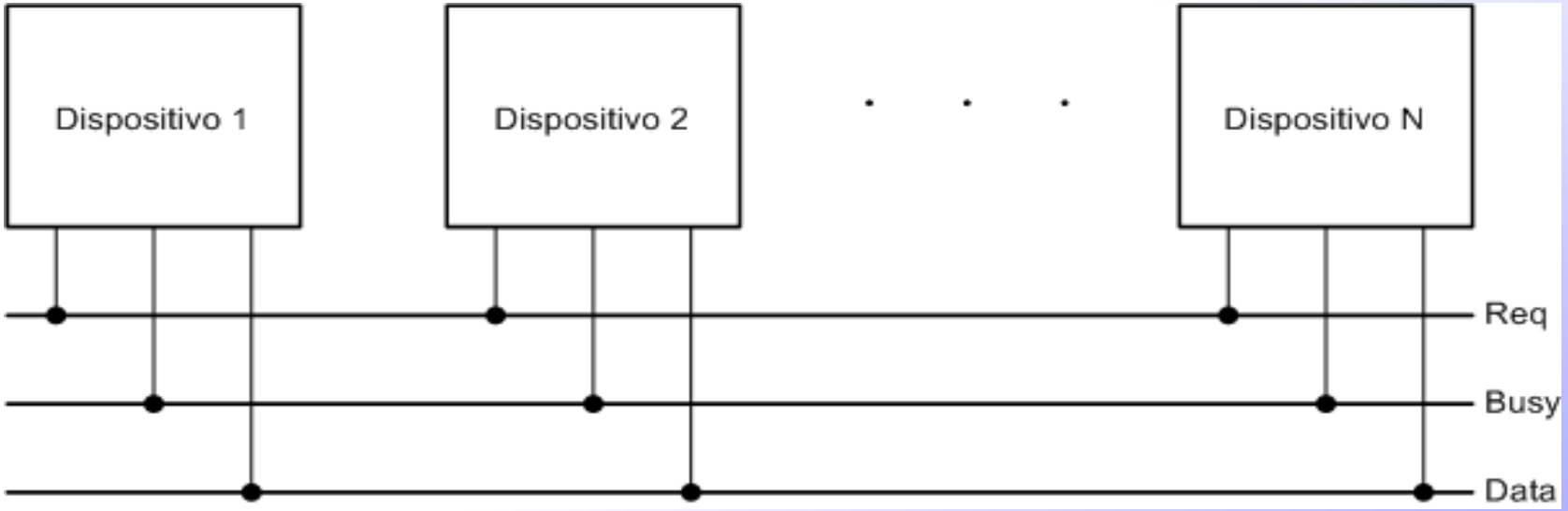
- Svantaggio:

  L'arbitro centralizzato può divenire il collo di bottiglia



Distribuito con auto-selezione -1

- Ciascun dispositivo che vuol accedere al bus scrive sulla linea Req del BUS un codice che lo identifica e controlla se il bus è libero (leggendo la linea Busy):
  - I codici dei vari dispositivi sono tali che se due dispositivi scrivono entrambi il proprio codice una lettura restituisce il codice del dispositivo a priorità piu' alta!!!

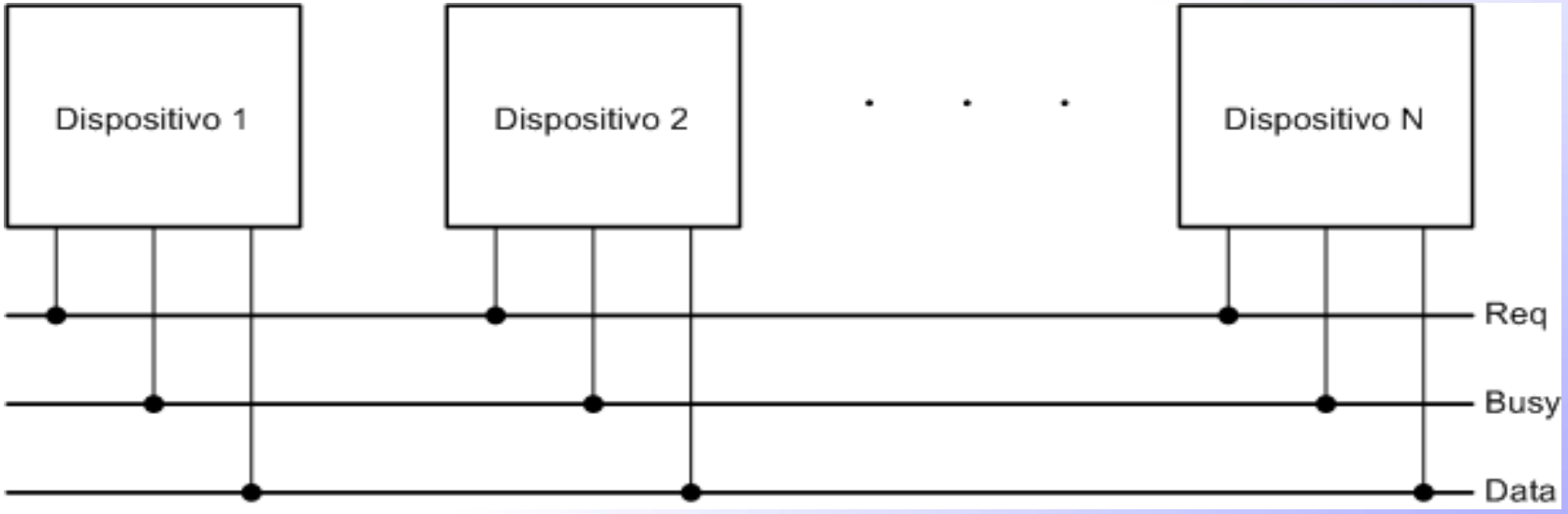


Distribuito con auto-selezione

- Se dalla lettura il bus risulta libero:

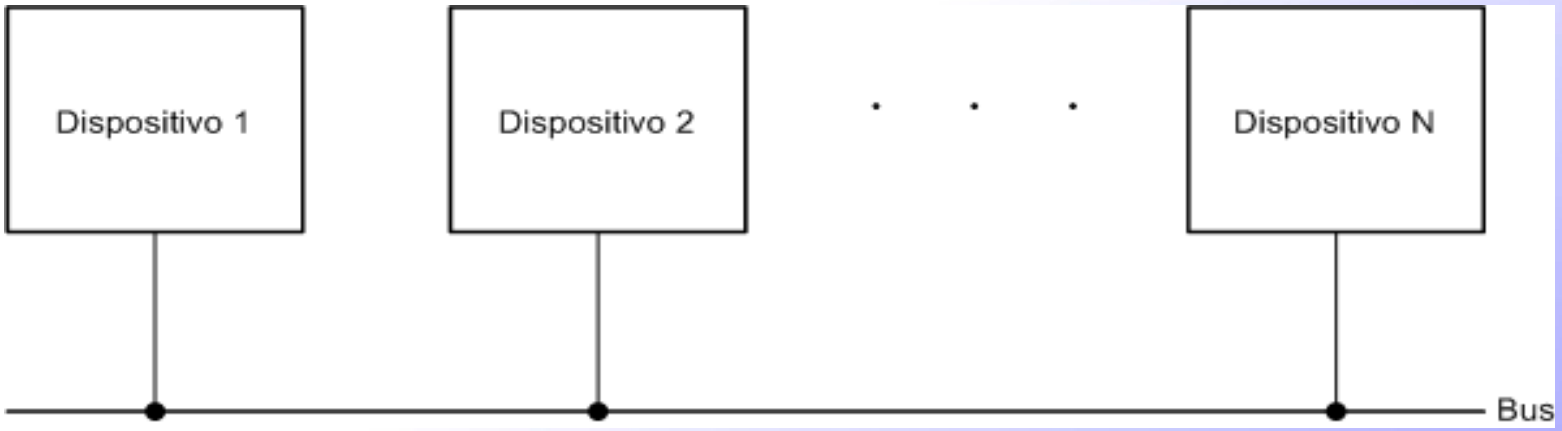
  il dispositivo legge sulla linea Req

  - se legge il proprio codice capisce di essere quello a più alta fra quelli in attesa e allora lo occupa (attivando Busy).;
  - Se legge un altro codice → ci sono richieste di priorità maggiore, e continua ad attendere



Distribuito con rilevamento delle collisioni: Ethernet

- Ciascun dispositivo che vuol accedere al bus avanza indipendentemente la propria richiesta
  - Comntrolla (leggendo dal bus) se il bus è libero,
    → allora inizia a trasmettere,
    → altrimenti attende un tempo esponenziale;
  - Se durante la trasmissione si accorge che anche un'altro dispositivo sta trasmettendo (si dice che si è verificata una *collisione*), (leggendo dal bus e rilevando che le informazioni lette sono diverse da quelle scritte da lui)
    → allora abortisce la trasmissione ed attende un tempo esponenziale.



Aumentare la banda del bus

- Linee dati ed indirizzo separate:
  - indirizzi e dati possono viaggiare nello stesso ciclo di bus
  - costo:
    a) più linee di bus;
    b) maggiore complessità;
- Larghezza del bus dati:
  - aumentando la larghezza del bus dati, trasferimento di più parole richiede meno cicli di bus
    - esempio: SPARCstation con 20 bus di memoria larghi 128 bit
  - costo: più linee di bus
- Trasferimento di blocchi:
  - permettere trasferimento di più parole in cicli di bus adiacenti
    - solo un indirizzo deve essere spedito all'inizio
    - il bus non viene rilasciato fino a che l'ultima parola è stata trasferita
  - costo:
    a) maggiore complessità;
    b) aumento del tempo di risposta per richiesta.

Analisi delle prestazioni

- Sistema bus/memoria:
  - Accesso a 4 oppure 16 parole di 32 bit
  - Bus sincrono a 64 bit con clock a 200MHz:
    - un ciclo per ogni trasferimento di 64 bit
    - un ciclo per l'invio dell'indirizzo di memoria
    - due cicli di clock tra due operazioni di bus
  - Accesso a memoria:
    - 200ns per le prime 4 parole
    - 20ns per i successivi gruppi di 4 parole
    - invio di dati in parallelo a lettura di dati successivi
  - Lettura di 256 parole:
    - banda, latenza, numero di transazioni sul bus al secondo?

Trasferimento di 4 parole

- Ogni blocco (di 4 parole):
  - 1 ciclo di clock per inviare l'indirizzo
  - 200ns x 200MHz=200/5=40 cicli di clock per leggere il blocco
  - 2 cicli di clock per inviare dati
  - 2 cicli di clock di attesa
- Totale: 45 cicli di clock per transazione
- 256/4=64 transazioni:
  - latenza: 45x64x5ns=14400ns
  - banda: (256x4/14400)byte/sec=71.11MB/sec
  - transazioni al secondo: 64/14400=4.44M

Trasferimento di 16 parole

- Primo blocco di 4 parole:
  - 1 ciclo di clock per inviare l'indirizzo
  - 200ns x 200MHz=200/5=40 cicli di clock per leggere il blocco
  - 2 cicli di clock per inviare dati
    - in parallelo inizia lettura del blocco successivo di 4 parole
  - 2 cicli di clock di attesa
- Successivi tre blocchi di 4 parole:
  - ultimi due passi
- Totale: 1+40+16=57 cicli di clock per transazione
- 256/16=16 transazioni:
  - latenza: 57x16x5ns=4560ns
  - banda: (256x4/4560)byte/sec=224.56MB/sec
  - transazioni al secondo: 16/4560=3.51M

Quattro standard a confronto

caratteristica	PCI	SCSI	Firewire (1394)	USB
Tipo di bus	backplane	I/O	I/O	I/O
Ampiezza base del bus dati (num. di segnali)	32-64	8-32	32	16
Indirizzi/dati	Stesse linee	Stesse linee	-	-
Master del bus	multipli	multipli	-	-
Arbitraggio	Centralizzato, parallelo	Distribuito, auto-selezione	-	-
Temporizzazione	Sincrono	Entrambi	Asincrono	Asincrono
Banda teorica	133-512 MB/s	5-40MB/s	50-100 MB/s	1,5-12 MB/s
Banda stimata	80 MB/s	2,5-40 MB/s (sincrono) 1,5 MB/s (asincrono)	-	-
Numero dispositivi max	1024	7-31	63	127
Lunghezza max	0,5 m	25 m	4,5 m	5 m
Nome standard	PCI	ANSI X3.131	IEEE 1394, 1394b	USE Implementors Forum

Responsabilità del SO

- I protocolli stabiliscono secondo quale schema deve svolgersi la comunicazione tra i dispositivi connessi attraverso i bus
- Il sistema operativo attiva tale comunicazione agendo da interfaccia tra l'hardware I/O ed i programmi che ne fanno richiesta, così da garantire le seguenti caratteristiche che deve avere un sistema di I/O:
  - La ***condivisione*** del sistema I/O da parte di più programmi che usano lo stesso processore;
  - la ***gestione degli interrupt*** che i dispositivi I/O utilizzano per comunicare informazione relative alle operazioni di I/O;
  - la trasparenza del controllo a basso livello dei dispositivi I/O che rende possibile un più alto livello d'***astrazione*** nel loro utilizzo.

Compiti del sistema operativo

Condivisione

- Fornire protezione a risorse I/O condivise
  - garantire che un programma utente acceda solo alle parti di un dispositivo I/O al quale l'utente ha diritto
- Consentire accesso equo a risorse I/O condivise:
  - tutti i programmi utente devono avere (uguale) diritto di accedere alle risorse I/O
- Schedulare gli accessi in modo da migliorare il throughput del sistema

Gestione Interrupt

- Gestire interrupt generati dai dispositivi I/O

Astrazione

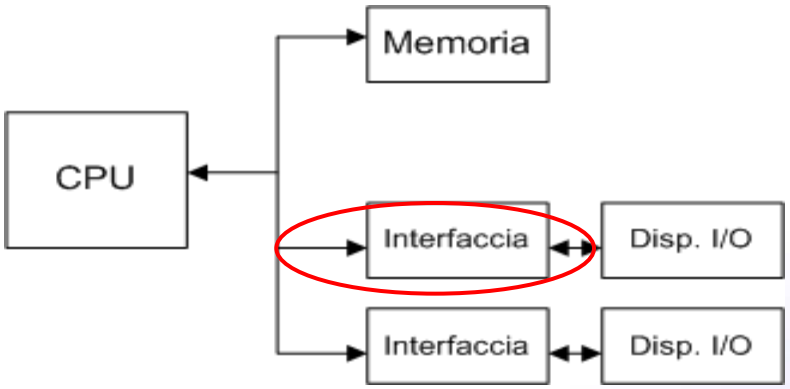
- Consentire astrazione per l'accesso ai dispositivi
  - fornire routine che gestiscono le operazioni a basso livello del dispositivo

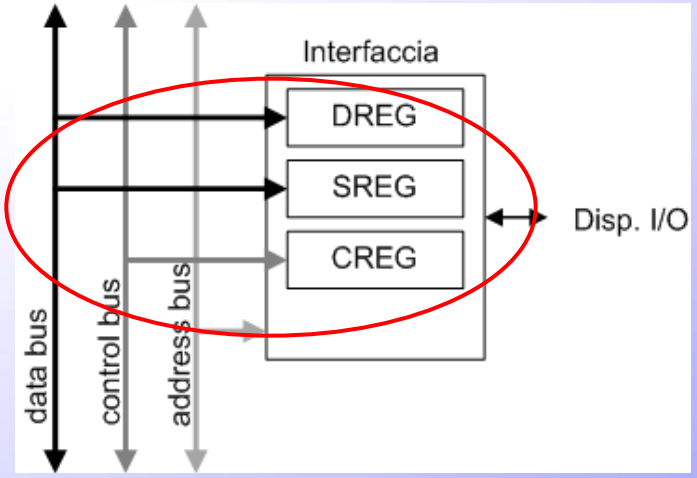
Tipi di comunicazione fra il S.O. ed i dispositivi I/O

- Deve essere possibile per il S.O. inviare ***comandi*** ai dispositivi di I/O
- Deve essere possibile per i dispositivi di I/O informare il S.O. del proprio ***stato***:
  - operazione completata
  - errore
- Deve essere possibile il trasferimento dei ***dati*** fra memoria e dispositivo e viceversa

DREG, SREG e CREG

- Ciascun dispositivo procede alla propria velocità ed in modo asincrono rispetto alla CPU
- Per rendere possibile la comunicazione fra la CPU e un dispositivo di I/O è necessario che quest'ultimo sia connesso al bus attraverso un'interfaccia, la quale deve:
  - Fornire eventuali registri dove possano essere appoggiati i dati da trasferire;
  - Fornire eventuali registri di appoggio per i comandi alla periferica
  - Tenere traccia dello stato della periferica





Dare comandi ai dispositivi I/O

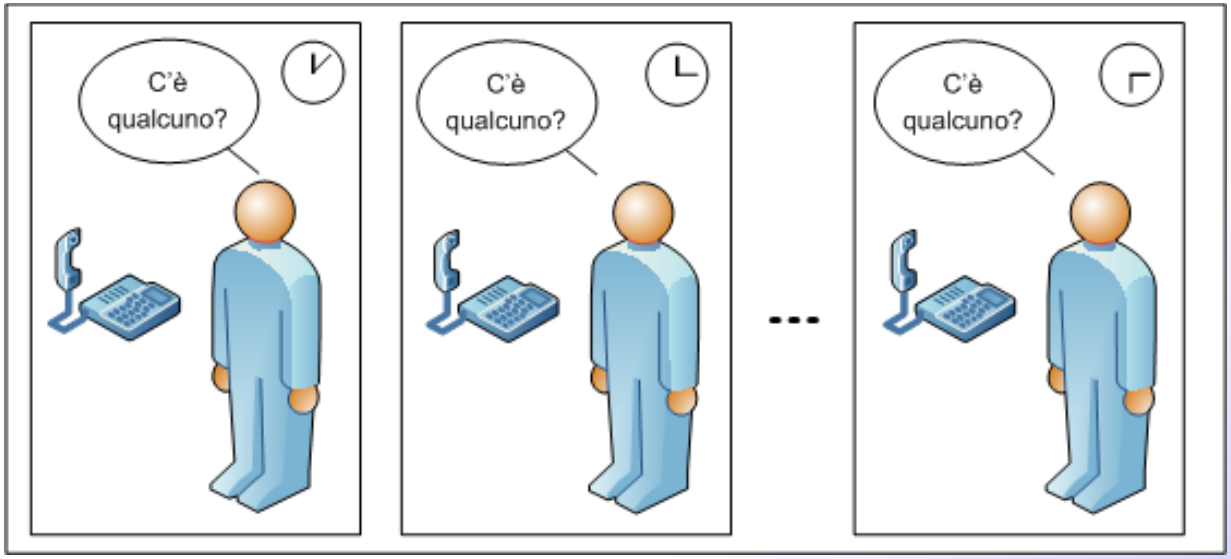
- Due metodi:
  - Istruzioni speciali di I/O (I/O Mapped I/O)
  - Indirizzi speciali di memoria (Memory Mapped I/O)
- Istruzioni speciali specificano:
  - Numero dispositivo (mediante cavi dedicati)
  - Parola di comando (mediante parte dati del bus)
- Indirizzi speciali di memoria:
  - Parti dello spazio indirizzi è riservato a dispositivi I/O
  - Letture e scritture su quegli indirizzi sono interpretati come comandi ai dispositivi I/O
  - Programmi utente non possono inviare operazioni I/O direttamente:
    - Lo spazio di indirizzi I/O è protetto

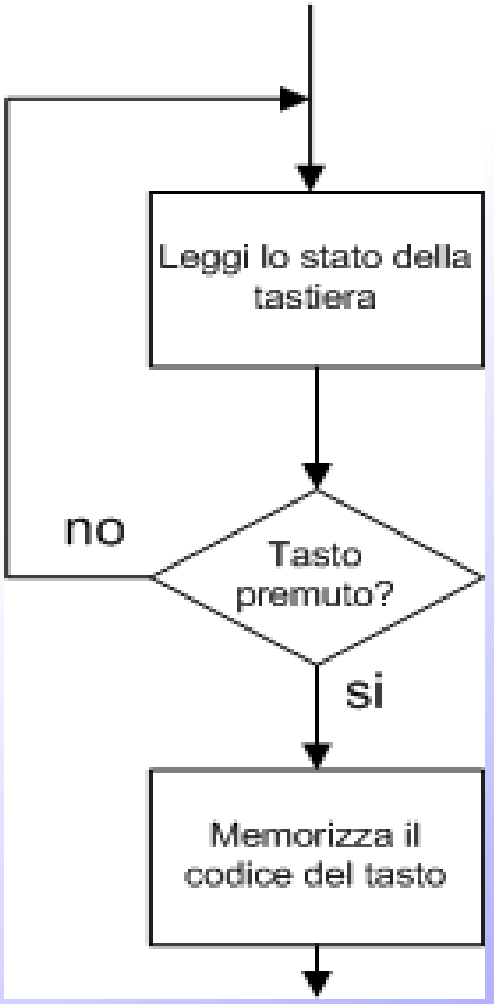
Notificare il processore

- Il processore deve conoscere lo stato di un dispositivo di I/O, cioè deve sapere quando:
  - Il dispositivo I/O ha completato un'operazione
  - Un'operazione I/O ha generato un errore
  - Il dispositivo I/O richiede attenzione
- Due metodi:
  - Interrogazione periodica (*polling*):
    - Il dispositivo mette l'informazione in un registro di stato
    - Il processore periodicamente verifica il registro di stato
  - Interrupt I/O:
    - Quando un dispositivo I/O ha bisogno di attenzione dal processore, lo interrompe da ciò che sta facendo in quel momento

Interrogazione periodica (polling)

- È il processo di verifica periodica del bit di stato di un dispositivo di I/O
- Esempio: telefono senza suoneria
  - Per sapere se c'è una chiamata in arrivo è necessario ascoltare periodicamente la cornetta





Interrogazione periodica (polling)

- Vantaggi:
  - Semplice da implementare: il processore ha il controllo e fa tutto il lavoro
  - Non richiede hardware aggiuntivo
- Svantaggi:
  - Tiene occupata la CPU anche quando non è necessario
  - Il polling deve essere eseguito ad una frequenza sufficiente a far si che nessun dato venga mai perso, quindi nel caso di dispositivi ad elevata frequenza di dati, si impiega una elevata frazione del tempo della CPU
    - Mouse → 30 volte al secondo
    - xxx → xx.000 volte al secondo
    - Disco Fisso → 250.000 volte al secondo

Prestazioni

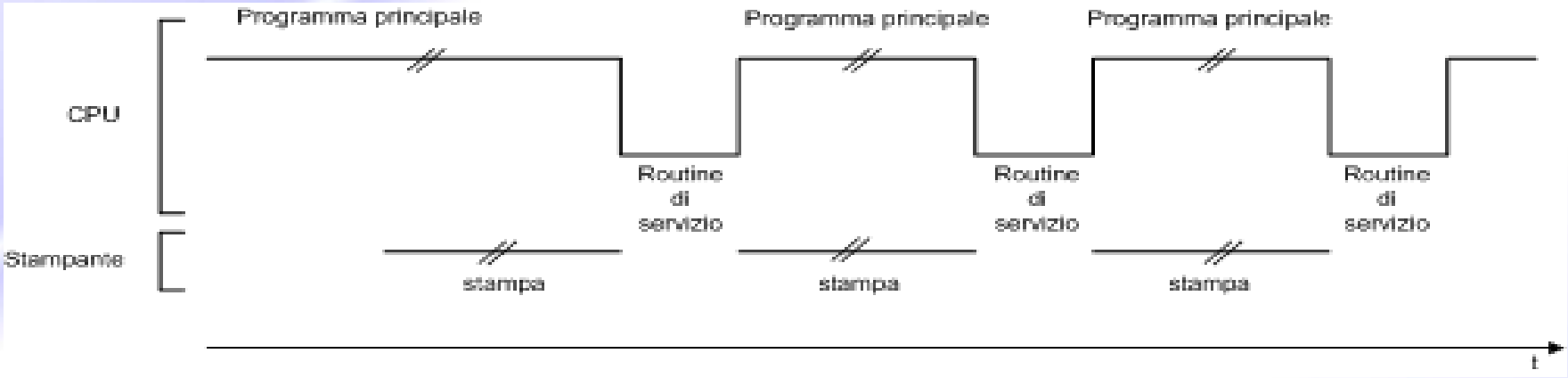
- Processore a 500MHz
  - 400 cicli di CPU per polling
- Mouse:
  - 30 interrogazioni al sec.
  - Cicli di polling al secondo: 30x400=12K
  - Percentuale tempo di CPU: 12K/500M=0,002%
- Disco fisso:
  - trasferimenti: 4 parole (16 byte)
  - banda: 4MB/sec
  - Trasferimenti al secondo: 4M/16=250K
  - Cicli di polling al secondo: 250Kx400=100M
  - Percentuale tempo di CPU: 100M/500M=20%

- Polling inaccettabile → INTERRUPT

Interruzioni I/O (Interrupt I/O)

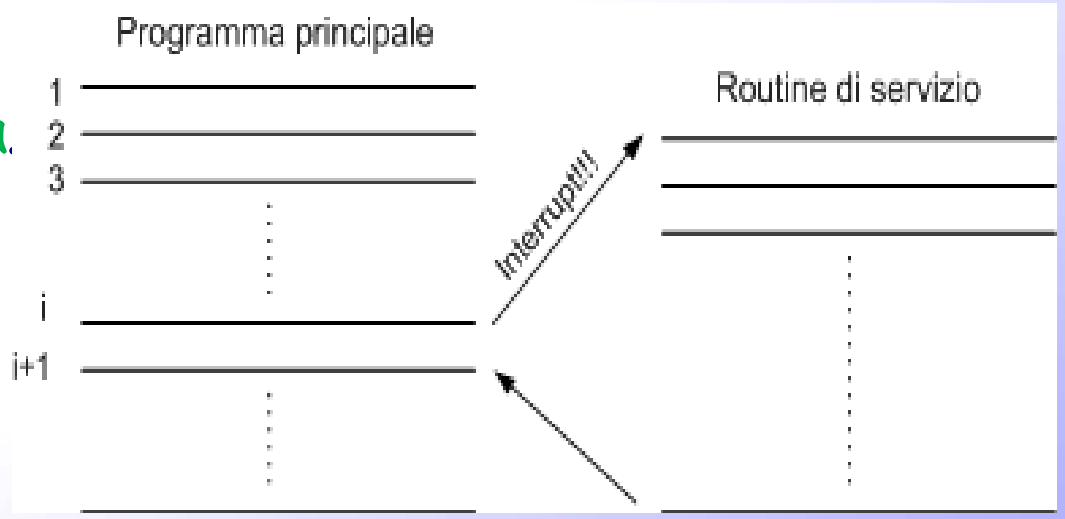
- Le interruzioni sono il meccanismo attraverso il quale un dispositivo di I/O segnala al processore quando necessita la sua attenzione.
- Esempio: telefono con suoneria
  - Il telefono richiede attenzione con la suoneria (interrupt), quindi rispondiamo al telefono solamente quando è necessario

Gestione di un Interrupt



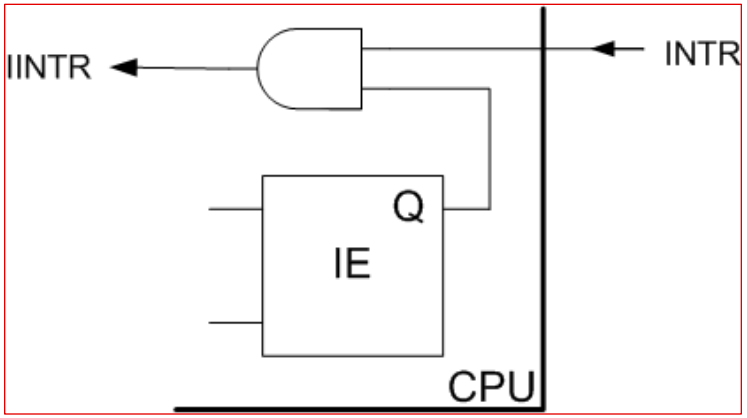
Le interruzioni si manifestano in modo asincrono rispetto all'esecuzione del programma.

La routine di servizio viene eseguita in maniera trasparente tra due istruzioni i e i+1 del programma principale



Controllo del sistema di interruzione

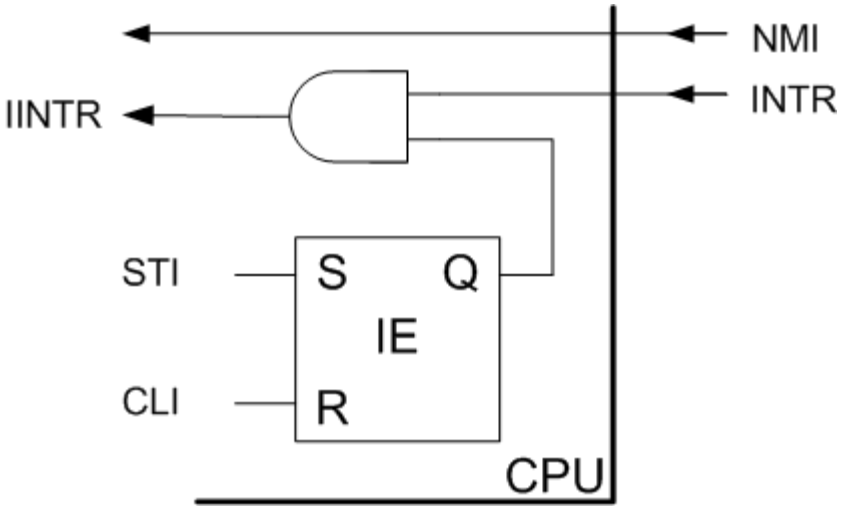
- È possibile "abilitare il sistema di interruzione" introducendo un flip-flop di nome IE (Interrupt Enabled) all'interno della CPU
  - La logica della CPU esamina IINTR per determinare se c'è una richiesta di interruzione
  - La richiesta di interruzione non viene rilevata se IE è in stato basso



  - In questo modo durante la gestione di un interrupt si puo non accettarne nessun altro

Interruzioni non mascherabili

- Il flip-flop IE serve ad abilitare/disabilitare il sistema di interruzione, ovvero a mascherare la richiesta INTR
- Quasi tutti i calcolatori prevedono tuttavia una linea di interruzione non mascherabile NMI che viene sempre rilevata nel caso sia asserita
  - Tipicamente impiegata nelle situazioni di emergenza (caduta di tensione)

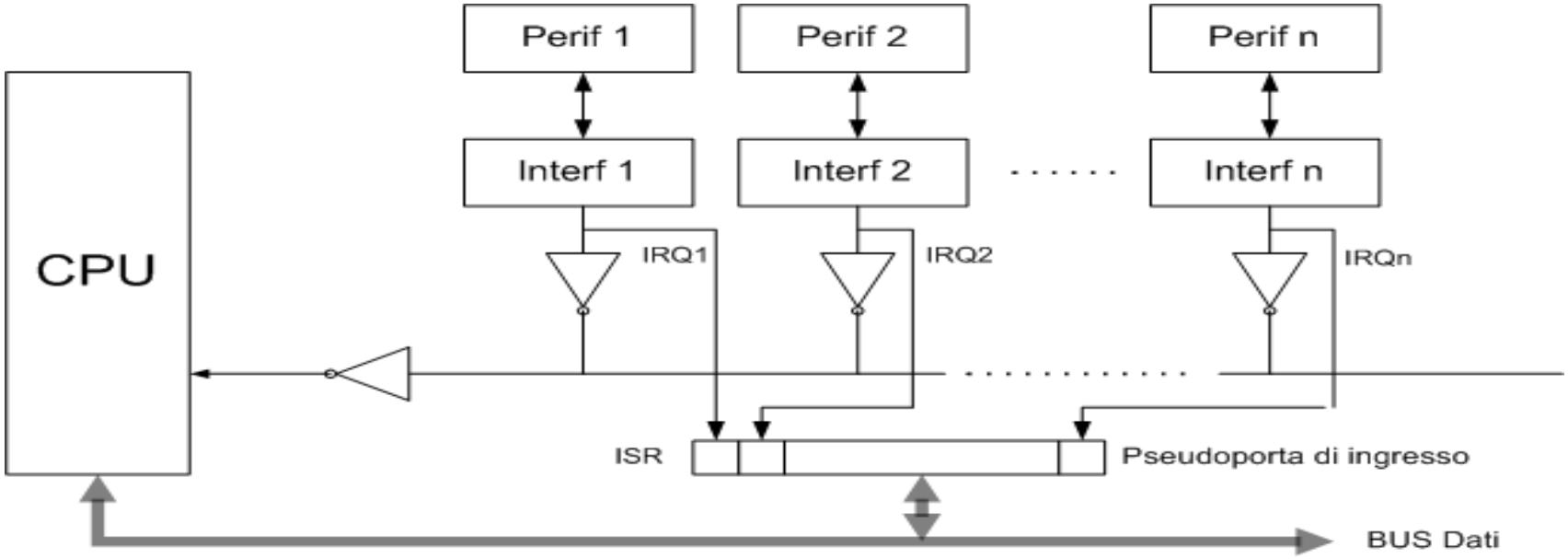


Interruzioni da parte di più periferiche

- In realtà la CPU deve gestire più di una periferica in grado di generare interrupt, è necessario quindi:
  - Riconoscere la periferica che ha generato l'interruzione
  - Scegliere la routine di servizio da eseguire
  - Trattare la priorità delle interruzioni in caso di richieste contemporanee
  - Gestire l'interoperabilità fra le routine di servizio
    - Una routine a bassa priorità può essere interrotta da un'altra a priorità più alta.

Priorità software delle Interruzioni

- Le interruzioni vengono discriminate leggendo una pseudoporta di I/O, ISR (Interrupt Status Register) i cui bit rappresentano lo stato delle singole IRQi
- La routine di servizio, dopo aver salvato i registri di interesse, legge ISR e stabilisce qual è la richiesta da servire

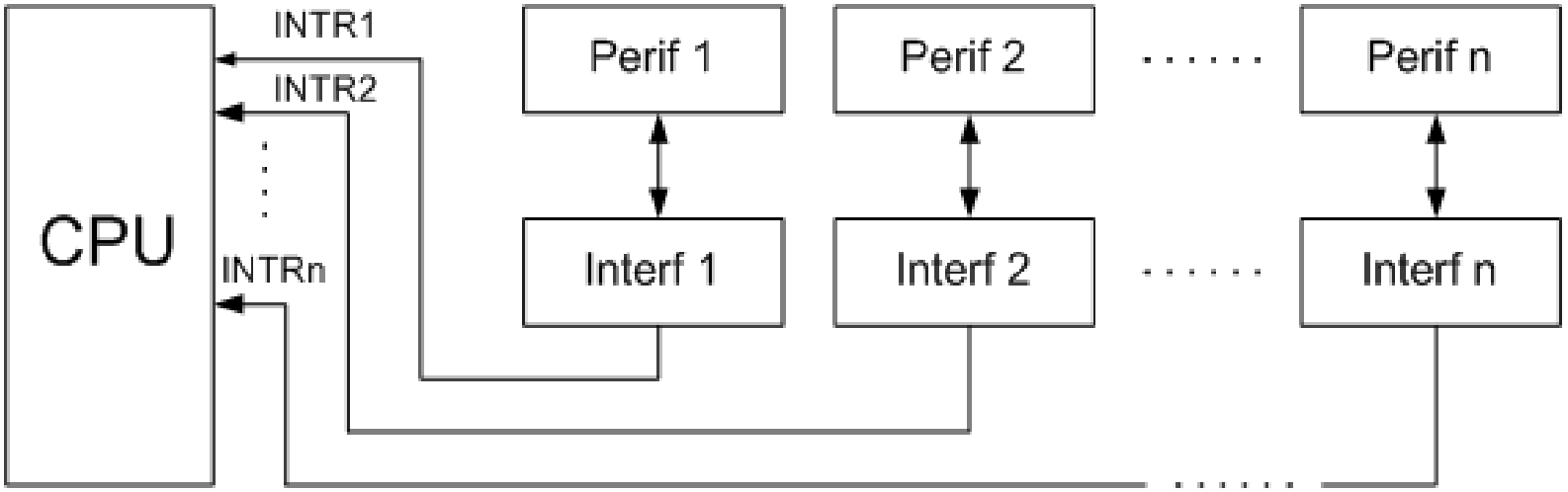


Interruzioni vettorizzate

- L'inefficienza del metodo precedente viene superata con un meccanismo capace di:
  - Identificare la richiesta da servire
  - Passare in modo automatico alla routine associata
- Si distinguono 2 casi fondamentali
  1. Il sistema presenta più linee di richiesta di interruzione, indipendenti e vettorizzate internamente alla CPU.
  2. C'è una sola linea di interruzione comune a tutte le richieste e la vettorizzazione è esterna attraverso una daisy chain o un controllore di interruzioni.

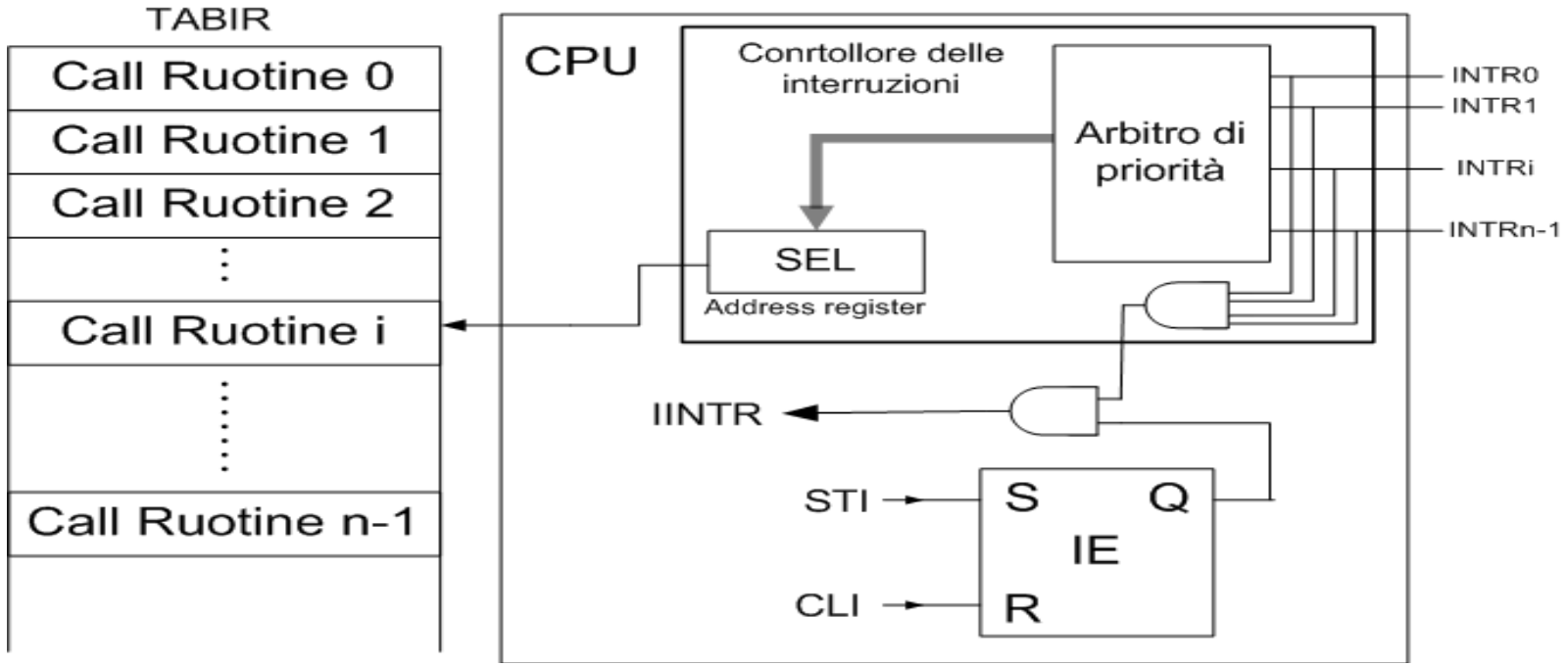
Interruzioni vettorizzate

- 1-a: linee di richiesta di interruzione separate
  - L'interruzione con più linee di richiesta indipendenti costituisce il metodo di gestione delle interruzioni più veloce e concettualmente più semplice, ma risulta anche il più dispendioso in termini di collegamenti



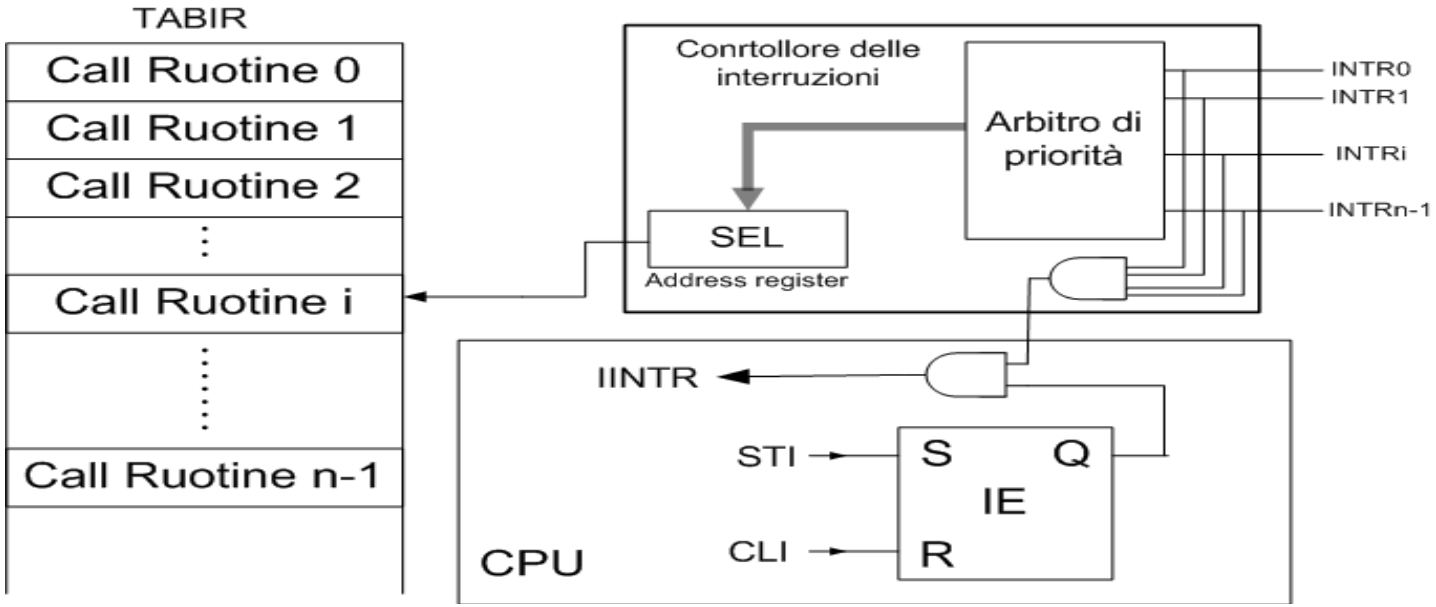
Interruzioni vettorizzate

- 1-b: linee di richiesta di interruzione separate e Controllore delle interruzioni
  - L'arbitro di priorità fornisce il numero dell'interruzione con priorità più alta, che viene interpretato come selettore di interruzione ed impiegato per trovare il vettore di interruzione nella tabella TABIR



Interruzioni vettorizzate

- 2: linea di richiesta di interruzione singola e Controllore delle interruzioni esterno
  - Nella costruzione dei microprocessori si preferisce, per motivi di economia, riservare un solo piedino per le interruzioni e prevedere un circuito esterno di appoggio, il controllore delle interruzioni, che svolga le funzioni di codificatore delle priorità.
  - Si porta il controllore delle interruzioni all'esterno della CPU, gestendo l'interazione fra di essi con opportuni protocolli.



Prestazioni (con interrupt)

- Processore a 500MHz
  - 500 cicli di CPU per ogni interrupt
- Disco fisso:
  - trasferimenti: 4 parole (16 byte)
  - banda: 4MB/sec
  - Trasferimenti al secondo: 4M/16=250K
  - Cicli di CPU al secondo: 250Kx500=125M
  - Percentuale tempo di CPU: 125M/500M=25% (polling)
  - Assumendo trasferimenti per il 5% del tempo:
    - Percentuale tempo di CPU = 25% x 5% = 1,25%

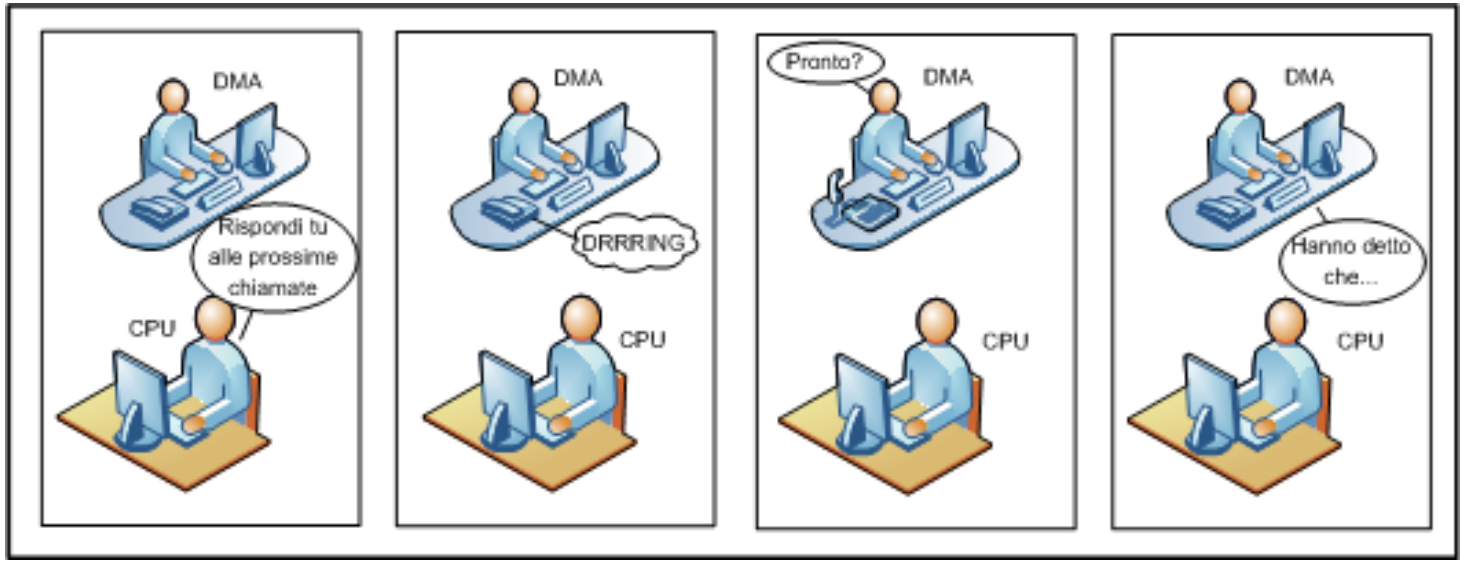
Gestione I/O tramite Interrupt

- Vantaggi:
  - Un programma utente è interrotto solo in caso di effettivo trasferimento dei dati
- Svantaggi:
  - È necessario hardware speciale per
    - generare gli interrupt da parte dei dispositivi I/O
    - rilevare interrupt (processore)
    - salvare gli stati da riesumare dopo la gestione dell'interrupt (processore)
  - La CPU resta comunque impegnata durante l'intero processo di trasferimento dei dati
  - Con dispositivi a banda elevata anche la gestione degli I/O tramite interrupt richiede troppo tempo di CPU

    → DMA

DMA (Direct Memory Access)

- Il processore incarica il DMA di gestire il trasferimento dei dati fra memoria e dispositivi I/O, al termine dell'operazione di I/O il DMA informa il processore
- Esempio: La segretaria che risponde al telefono
  - Assumiamo una segretaria che risponda al telefono per conto nostro



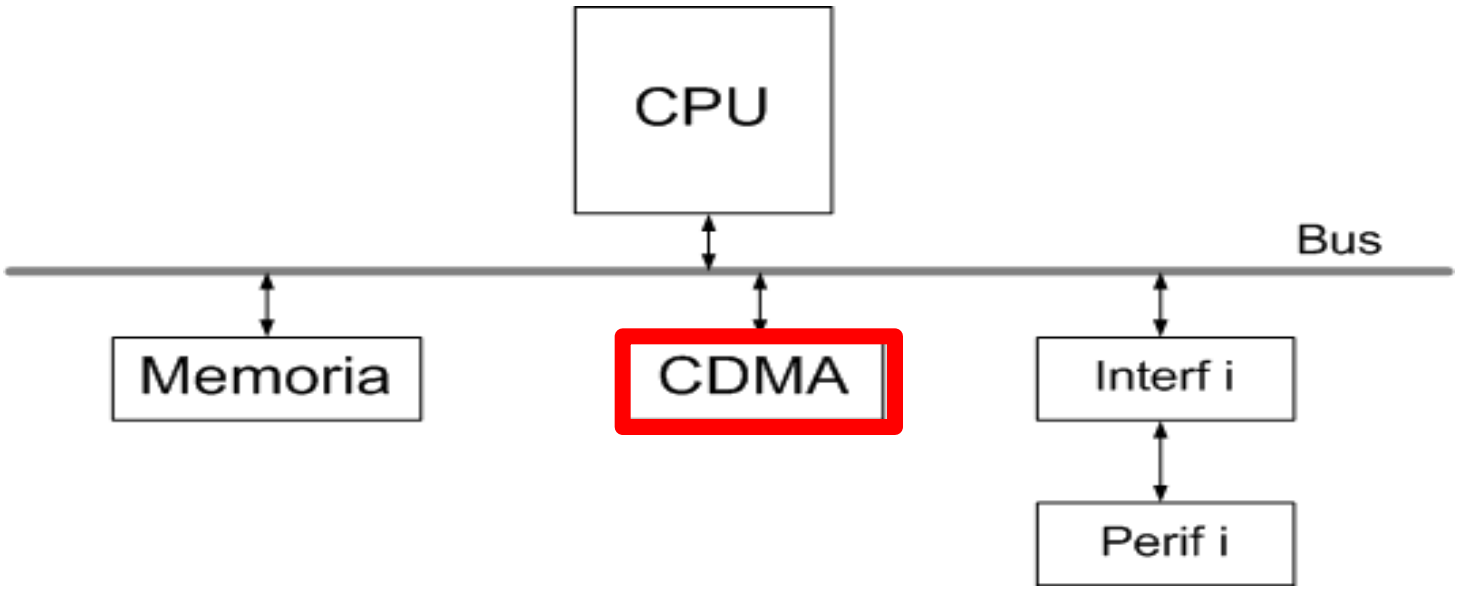
DMA (Direct Memory Access)

- Risparmia lavoro al processore facendo in modo che il controllore del dispositivo provveda direttamente a trasferire i dati da o verso la memoria, senza coinvolgere il processore.

- Il sistema di interruzioni viene comunque usato dal dispositivo per comunicare con il processore, ma soltanto al completamento del trasferimento di I/O o quando si verifica un errore.

DMA (Direct Memory Access) - 1

- Il DMA viene implementato attraverso un controllore che, diventando master del bus, trasferisce i dati.



DMA (Direct Memory Access) -2

- Un trasferimento consiste in 3 passi:
  - Il processore programma l'operazione di DMA fornendo
    - l'identità del dispositivo
    - l'operazione da eseguire
    - L'indirizzo di memoria
    - Il numero di byte da trasferire

- Il DMA provvede all'arbitraggio del bus, e quando il dato è disponibile esegue il trasferimento.
- Quando il trasferimento è completato, il controllore interrompe il processore, che potrà determinare se l'intera operazione ha avuto successo.

Prestazioni (con DMA)

- Processore a 500MHz
  - 1000 cicli per configurazione DMA
  - 500 cicli per gestire interrupt al termine del DMA
- Disco rigido:
  - Si trasferiscono in media 8KB di dati
  - banda: 4MB/sec
  - Ogni trasferimento richiede 8K/4M=0,002sec
  - Trasferire dati di continuo richiede
    1500/0,002= 750K cicli di clock / secondo
  - Percentuale tempo di CPU: 750K/500M= 0,15%

Concludendo

- Interazione con dispositivi di I/O fondamentale nel progetto di un sistema
- Dispositivi molto disomogenei
- BUS
  - Diversi tipi
  - Sincroni/asincroni
- Arbitraggio
- Ruolo eresponsabilità del S.O.
- Ntificare il processore
  - Polling
  - Interrupt
  - DMA