



UNIVERSITÀ DEGLI STUDI DI PARMA

Corso di Laurea in

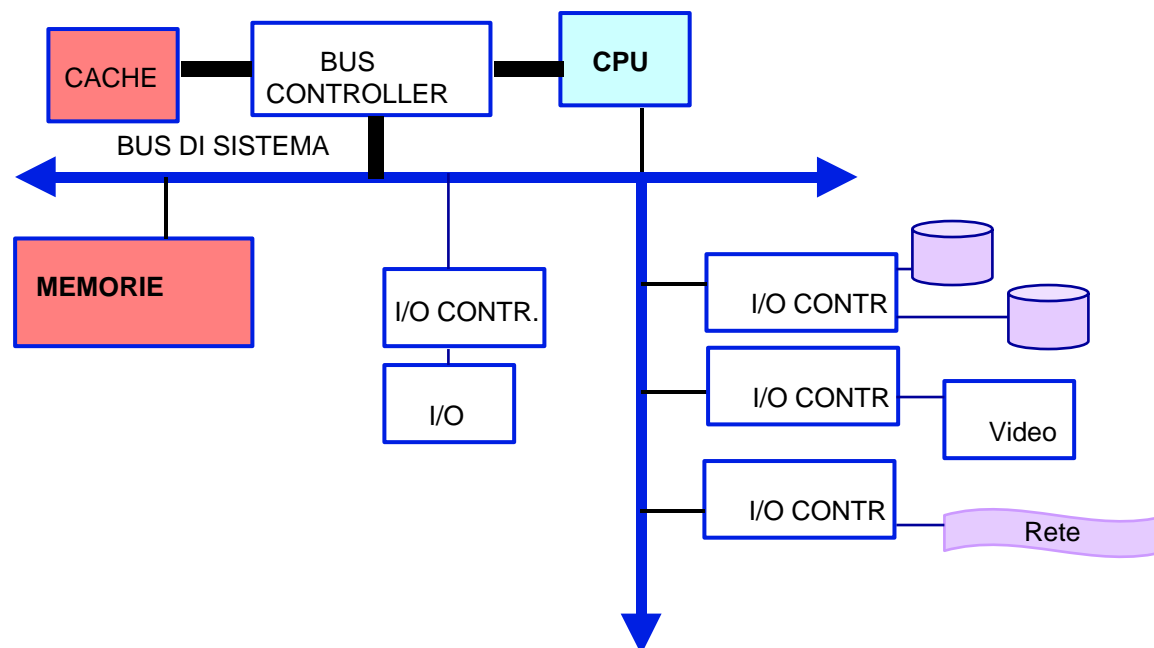
"Ingegneria Informatica, Elettronica e delle Telecomunicazioni"

Architettura dei Calcolatori Elettronici

La gestione dei dispositivi di I/O

Andrea Prati

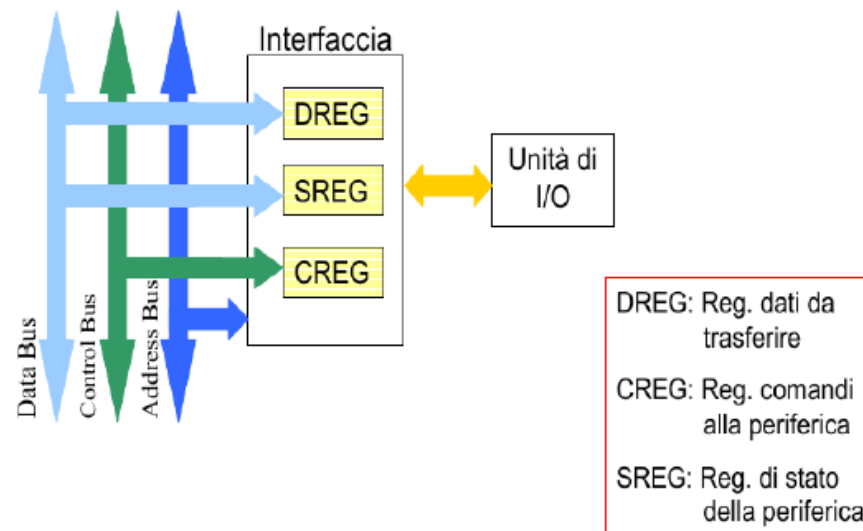
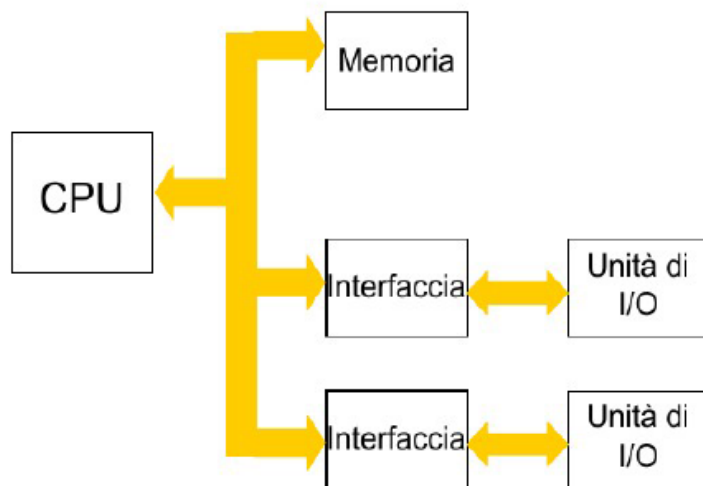
Interfaccia con le periferiche



- La CPU comunica con periferiche (esterne ed interne) e moduli di I/O attraverso controllori di I/O che colloquiano con la CPU attraverso il bus di sistema (e se esiste con il bus di I/O)
- Operazioni di input: dati vengono letti dalla CPU o trasferiti da I/O a memoria
- Operazioni di output: dati vengono scritti dalla CPU o trasferiti da memoria a I/O

Interfaccia con le periferiche

- Interfaccia deve utilizzare registri per:
 - Supporto trasferimento dati
 - Supporto comandi a periferica
 - Supporto lettura stato periferica



Interfaccia con le periferiche

- Lettura o scrittura
- Spesso pochi bit sufficienti a descrivere lo stato e il comando:
 - pseudoregistri
- Vi accedo tramite indirizzo
 - Lo spazio degli indirizzi di I/O è disgiunto da quello della memoria
- Due strategie per organizzazione indirizzi
 - I/O mappato in memoria (memory mapped I/O)
 - I/O isolato (I/O mapped I/O)

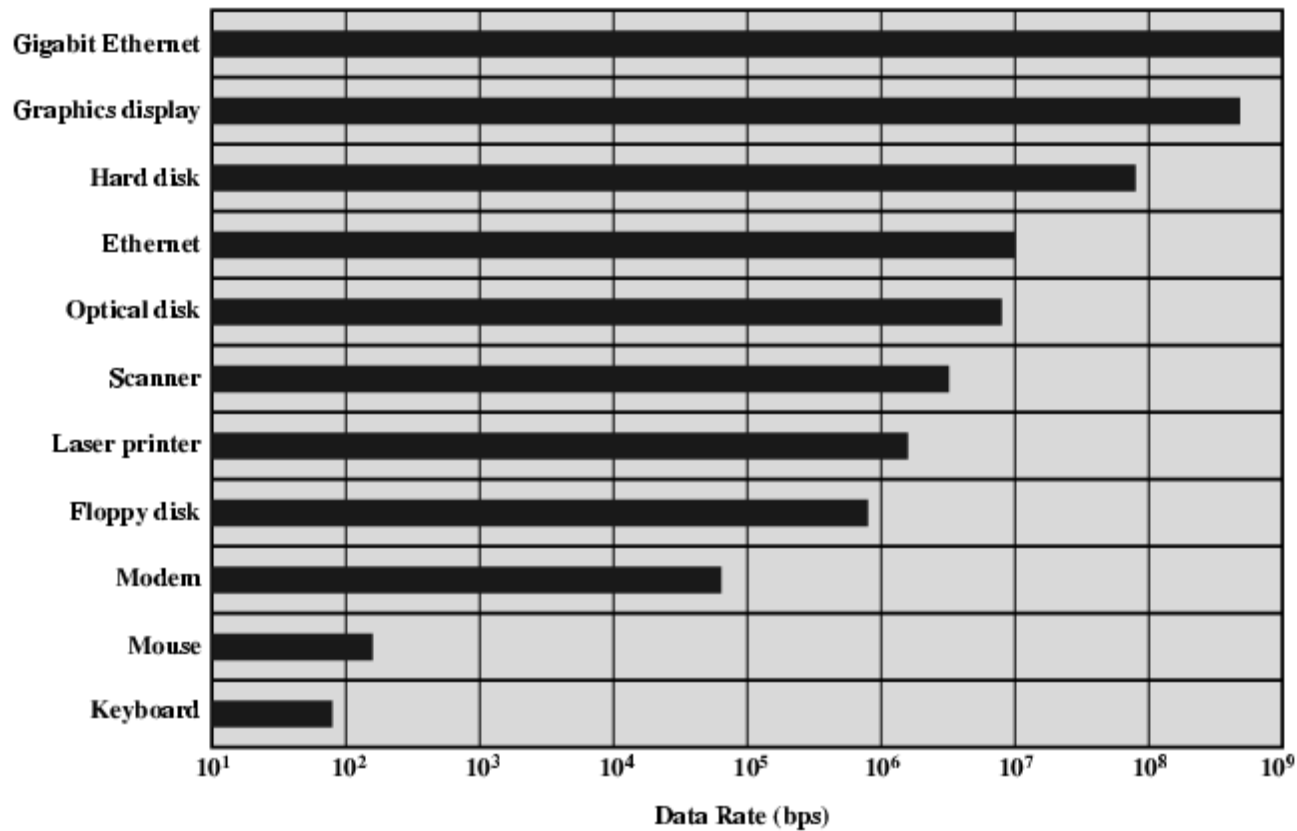
Accesso all'I/O

L'accesso alle periferiche avviene attraverso il bus e dipende:

- 1) dal tipo di protocollo di bus, dalla configurazione multimaster (che richiede o no arbitraggio)
 - 2) dal parallelismo di dati trasferiti
 - 3) dalla possibilità di trasferimento a blocchi (burst)
 - 4) dalla possibilità di accedere contemporaneamente a più dispositivi di I/O (transazioni suddivise)
 - 5) dalla possibilità che l'I/O lavori concorrentemente ad altre risorse e alla CPU.
- L'accesso all'I/O avviene nei calcolatori normalmente attraverso il sistema operativo che:
 - gestisce le protezioni e i privilegi dell'utente
 - fornisce le astrazioni necessarie per mascherare l'interfaccia di basso livello
 - gestisce l'accesso al basso livello eseguendo opportuni *driver*

Velocità dell' I/O

- Esempi di bps trasferiti



Gestione degli eventi

Gestione degli eventi

(segnali esterni, trasferimenti di dato verso l'I/O, controllo delle periferiche...)



a controllo di programma

POLLING

Gestito dal master
(la CPU)



ad interruzioni

gestione di eventi asincroni
disaccoppiamento delle funzioni

Il metodo più semplice è a controllo di programma con interrogazione periodica (polling).

Transazione in polling

La CPU verifica lo stato della periferica (leggendo in locazioni di I/O corrispondenti a registri di stato)

Continua la richiesta fino a che non verifica che è pronta per ricevere dati in uscita (o fornire dati in ingresso)

Inizia il trasferimento dati (in scrittura verso la periferica o in lettura)

Ricomincia se necessario

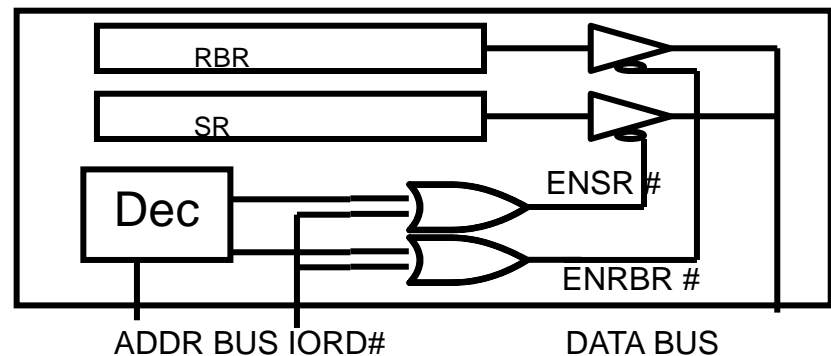
Controllo di programma

- Esempio: accesso ad una tastiera (lettura da I/O).
- **RBR** (Read buffer register) all'indirizzo 0060h; **SR** (status register) all'indirizzo 0064h contenente un bit di flag (1 dato presente, 0 dato assente); a controllo di programma la CPU continua a leggere fino a che un tasto non viene premuto
- **Driver:** Il driver della periferica è il programma che la CPU esegue
 - 1) per programmare la periferica
 - 2) per accedere ai suoi dati

```

adr_SR equ 0064h
adr_RBR equ 0060h
datook equ b0000001
read_tast proc
tast:   in al,adr_SR
        test al,datook
        je tast
        in al,adr_rbr
        <elabora al o ritorno>
        ret

```



Limiti del controllo di programma

- Ogni dispositivo deve dipendere dalla CPU per essere servito → Non solo la CPU spreca tempo per controllo
- Le periferiche sono lente → La CPU deve anche aspettare
- Non c'è connessione memoria/periferica → La CPU deve gestire trasferimento dati ed eseguire operazioni di test
- 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 = $30 \cdot 400 = 12000$ clock/sec
 - % Processore per polling = $12 \cdot 10^3 / (500 \cdot 10^6) = 0,002\%$
 - Floppy: trasferimento dati di 2 byte alla velocità di 50 kB/sec per non perdere dati
 - Polling Clock/sec = $50 \text{ kB} / 2 \cdot 400 = 10,000,000$ clock/sec
 - % Processore per polling = $10 \cdot 10^6 / (500 \cdot 10^6) = 2\%$
 - Hard disk: trasferimento di blocchi di 16 byte alla velocità di 8 MB/sec per non perdere dati
 - Hard Disk Polling Clock/sec = $8 \text{ MB} / 16 \cdot 400 = 200 \cdot 10^6$ clock/sec
 - % Processore per polling = $200 \cdot 10^6 / (500 \cdot 10^6) = 40\%$

Interruzioni

INTERRUZIONE:

Sospensione forzata del processo di esecuzione e trasferimento di controllo ad una “routine di servizio” che soddisfa le richieste dell'evento che ha provocato l'interruzione, al termine della quale il controllo viene restituito al processo sospeso

Tipologie di interruzioni:

- **HARDWARE:**
 - mascherabili
 - non mascherabili
- **SOFTWARE:**
 - nel programma
 - indirette (stato della cpu)

Interruzioni

Tutti i calcolatori impiegano il meccanismo delle interruzioni. Tale meccanismo si distingue nelle seguenti fasi:

- 1) notifica delle interruzioni
- 2) accettazione della interruzione
- 3) identificazione della sorgente e reperimento dell'indirizzo della RRI
- 4) salvataggio dello stato della CPU e modifica del PC
- 5) esecuzione delle RRI
- 6) ritorno al programma chiamante e ripristino dello stato

1) notifica dell'interruzione

- **in software** (alcuni bit di flag come overflow, divisione per 0, page fault....)
- **a controllo di programma**
 - nelle macchine Intel → INT n
- **in hardware**
 - segnale esterno che viene sentito al termine dell'istruzione seguente (o al primo clock nello stadio di esecuzione delle pipeline)

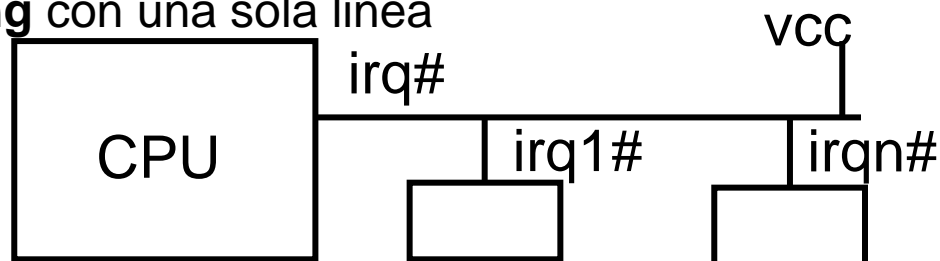
Interruzioni nell'8086

2) Accettazione dell'interruzione

- interruzioni software sono sempre accettate; quelle hardware dipende. Esistono due segnali che arrivano alla CPU nelle macchine Intel:
 - NMI: non mascherabile
 - INTR: mascherabile

3) Identificazione

1) in **polling** con una sola linea



- più sorgenti di int possono essere collegati ad una sola linea. La CPU in polling deve richiedere quale è la sorgente di int

2) in modo **vettorizzato**

Interruzioni vettorizzate

1) Vettorizzate internamente:

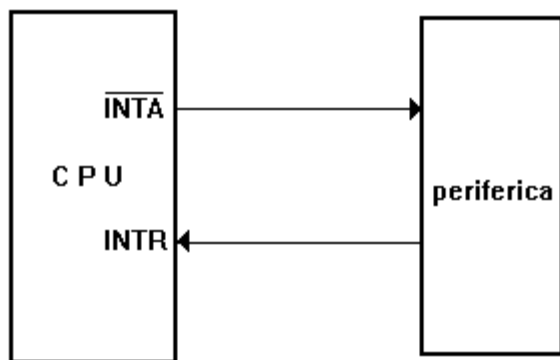
- alla CPU arrivano più richieste; all'interno della CPU un arbitro di priorità decide quale interruzione servire e lo indica al PC che seleziona la tabella (vettore) delle interruzioni

2) Vettorizzate con controllore esterno:

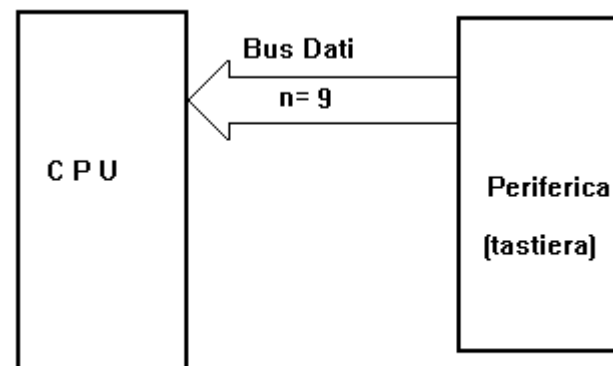
- L'identificazione viene fatta in hardware dal richiedente che manda l'identificatore della RRI (routine di risposta all'interrupt) da cui risalire all'indirizzo di partenza
- di solito attraverso un *interrupt controller* esterno che gestisce le priorità e sa quale è l'indirizzo di partenza nel vettore delle interruzioni

Nelle macchine Intel: accettazione e interruzioni vettorizzate (1/2)

- 1) La CPU invia il segnale $\overline{\text{INTA}}$ (Interrupt Acknowledgement) al dispositivo esterno per comunicargli che l'interruzione è stata accettata (attraverso uno o due impulsi del segnale)
- 2) Il dispositivo esterno mette sul Bus Dati della CPU un codice di 8 bit, chiamato vettore d'interruzione



1) accettazione di un interrupt

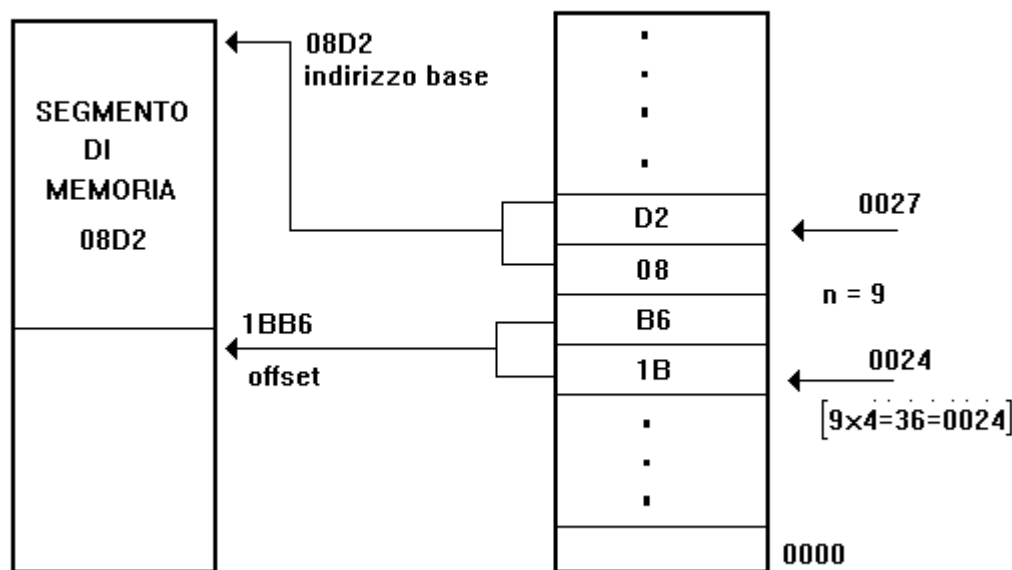


2) la periferica invia l'interrupt vector $n = 9$

Nelle macchine Intel: accettazione e interruzioni vettorizzate (2/2)

- 3) La CPU legge il vettore d'interruzione, lo moltiplica per 4 e forma un offset di 16 bit con cui accede alla tabella delle interruzioni

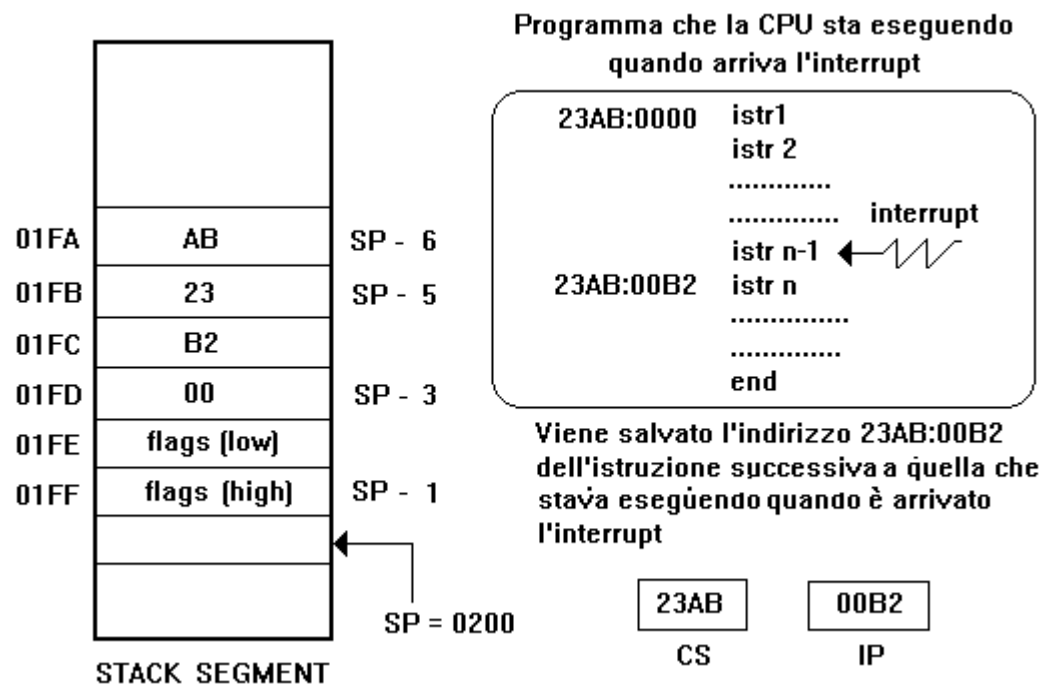
L'ingresso $n=9$ della tabella interrupt contiene l'indirizzo 08D2:1BB6 a partire dal quale è memorizzata la routine di servizio della tastiera



La tabella delle interruzioni è allocata a partire da 0000:0000

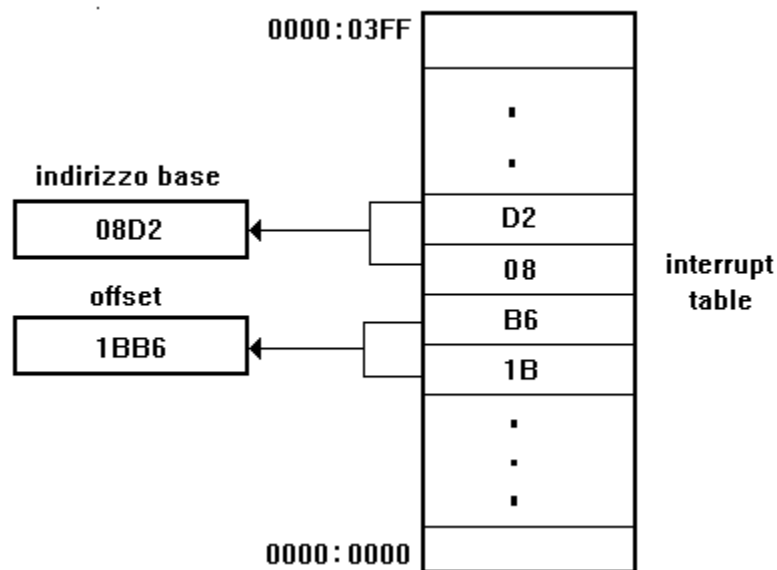
Nelle macchine Intel: RRI (1/2)

- 4) La CPU salva nel segmento di stack l'indirizzo completo (base+offset) dell'istruzione successiva a quella che ha eseguito prima dell'arrivo dell'interrupt, e il registro dei flags



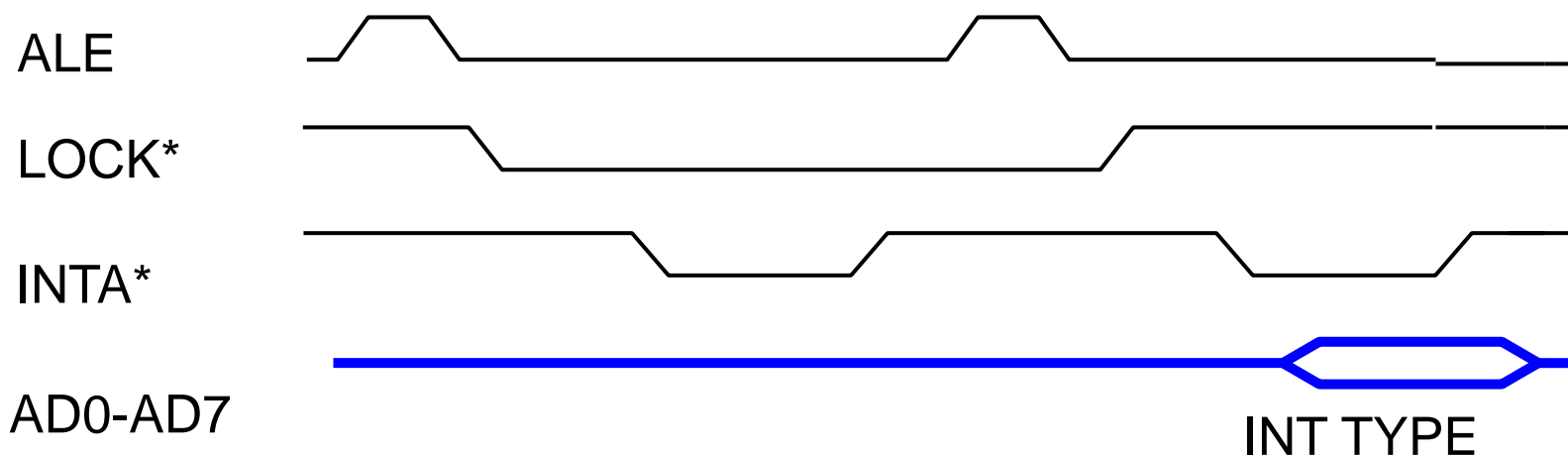
Nelle macchine Intel: RRI (2/2)

- 5) La CPU carica l'indirizzo (base+offset) contenuto nella tabella delle interruzioni nei registri CS e IP
- 6) La CPU esegue la routine d'interrupt
- 7) Al termine della routine d'interrupt, la CPU rimette in CS, IP e nel registro dei flags i valori che aveva salvato precedentemente, riprendendo in tal modo l'esecuzione del programma interrotto



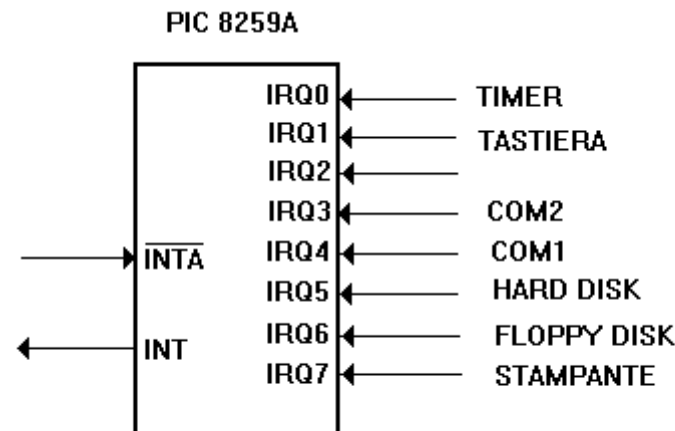
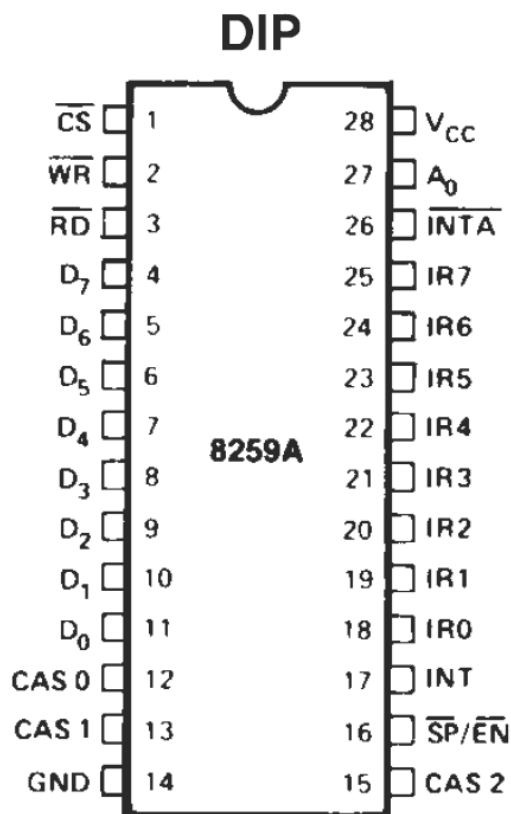
Interfaccia di interruzioni

- Nell'8086
 - Normale ciclo di bus in cui il segnale INTA* sostituisce il segnale di READ* (il segnale di WAIT* sono riconosciuti anche nel ciclo di interruzione)
 - Gli indirizzi emessi sono privi di significato: i segnali ALE sono generati solo per indicare l'inizio del relativo ciclo di bus
 - Il primo ciclo di INTA* serve a risolvere i conflitti e a congelare la situazione
 - La periferica risponde al secondo INTA inviando sul bus dati l'interrupt type



Gestioni delle priorità

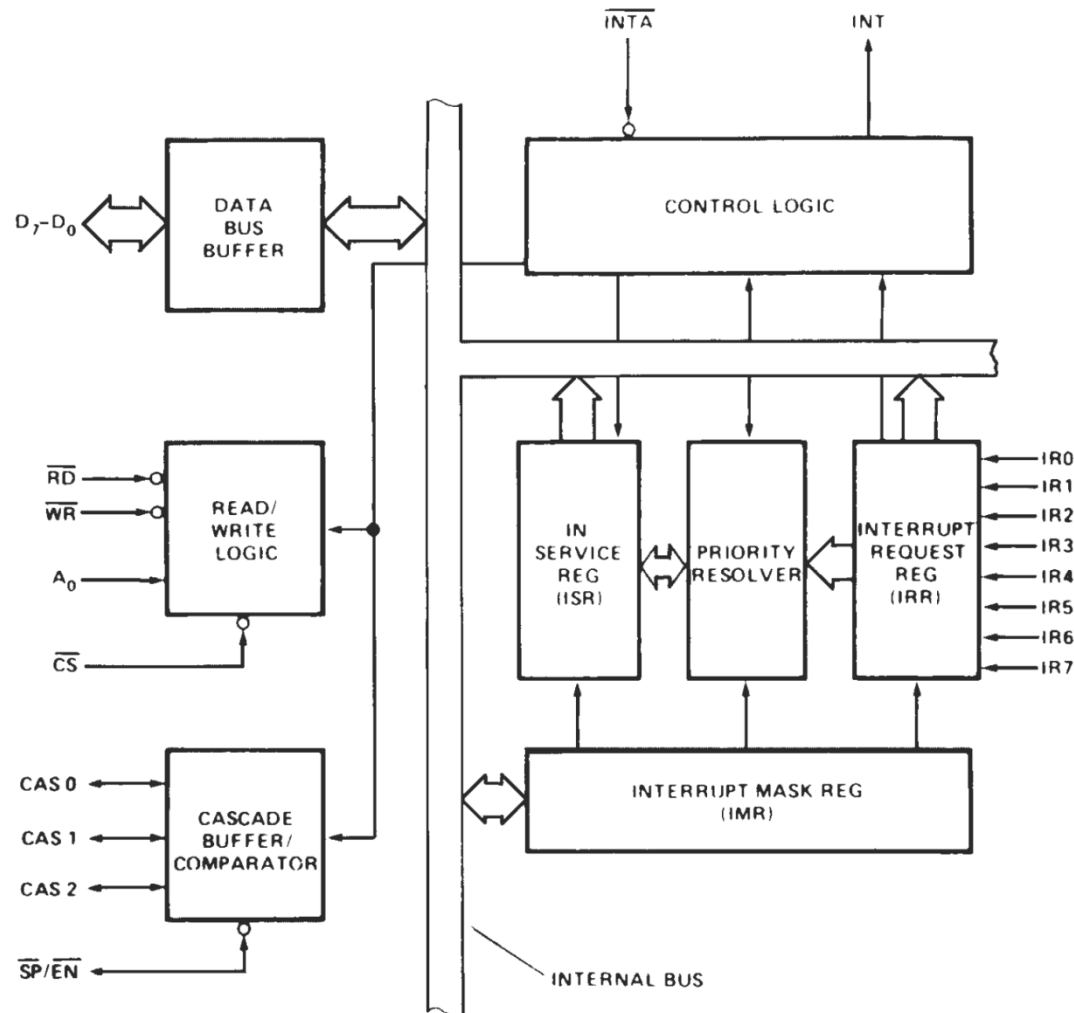
– Uso del controllore 8259



- Gestisce 8 interruzioni da solo e fino a 64 se connesso in cascata
- Occupa 2 locazioni di indirizzo
- Riconosce le interruzioni esterne a **livello o sul fronte positivo**
- Durante il ciclo di riconoscimento della interruzione il segnale INTA* ingloba CS* e RD*
- Modifica delle priorità, mascheratura, polling, scelta dell'EOI, ecc.

8259

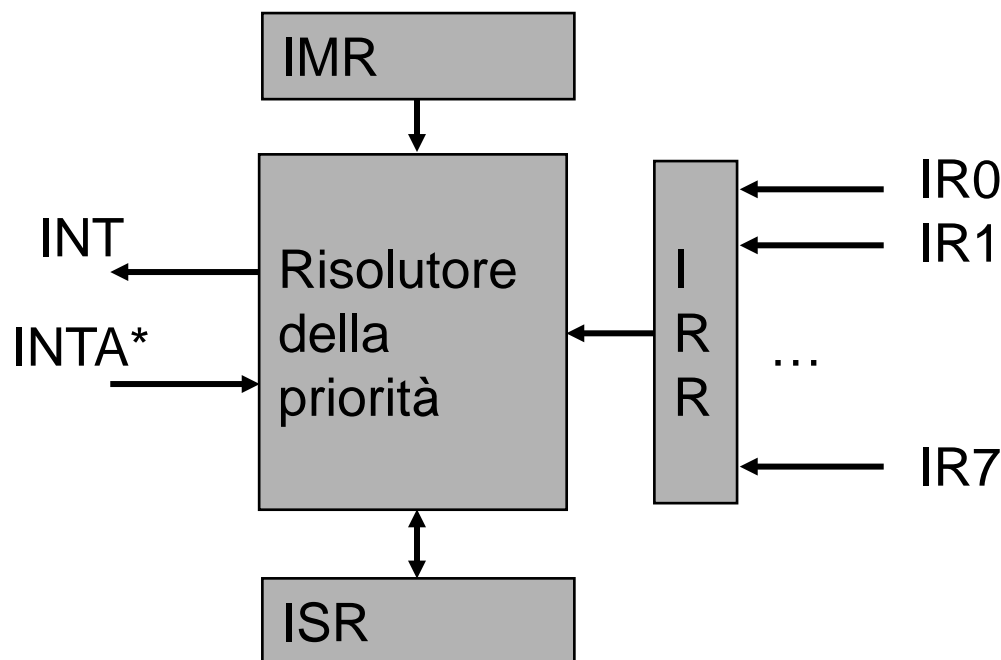
- Struttura interna: 8 interruzioni all'IRR (Interrupt Request Register); quella che vince la priorità passa nell'ISR (In Service Register)



I registri interni

- **IRR** (Interrupt Request Register) Memorizza le richieste d'interruzione; il bit n viene settato quando su IRQ_n viene presentata una richiesta di interruzione
- **PRIORITY RESOLVER** determina se la richiesta d'interruzione deve essere inoltrata alla CPU o no in base ad una gerarchia di priorità che (per default) va dalla più alta, IRQ_0 , alla più bassa, IRQ_7 .
- **ISR** (In Service Register) Memorizza le interruzioni in esecuzione. Il bit n viene settato se la richiesta IRQ_n è stata accettata dalla CPU

IMR (Interrupt Mask Register) Permette di mascherare una richiesta d'interruzione. Se il bit n viene posto a 1, il PIC ignora la richiesta su IRQ_n

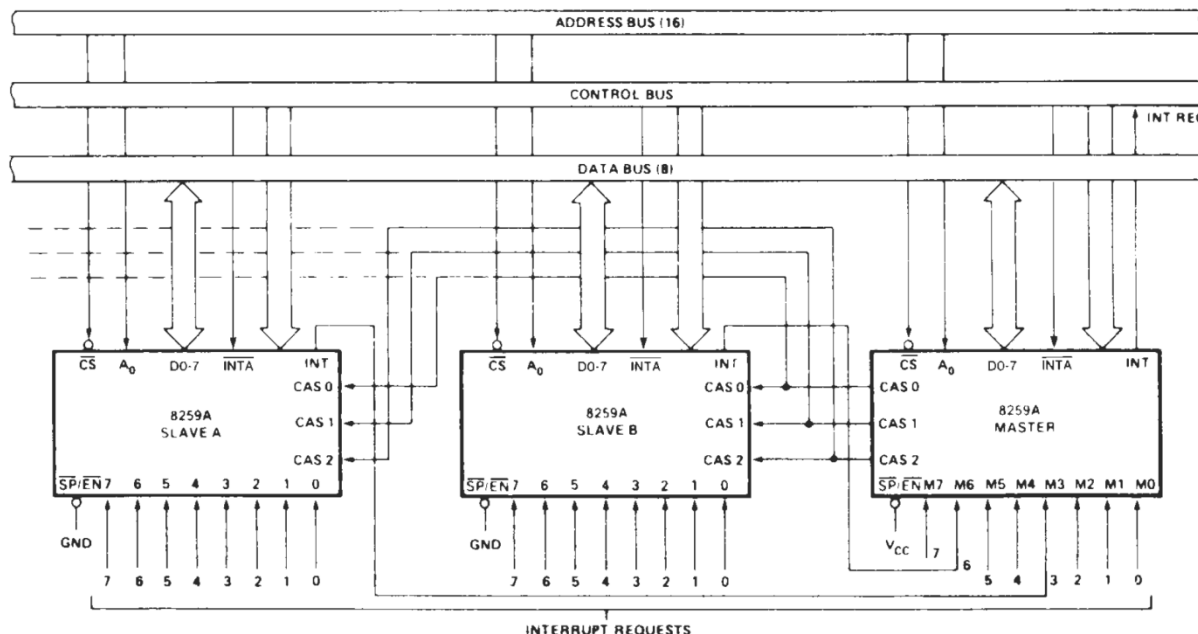


Funzionamento 8259 PIC

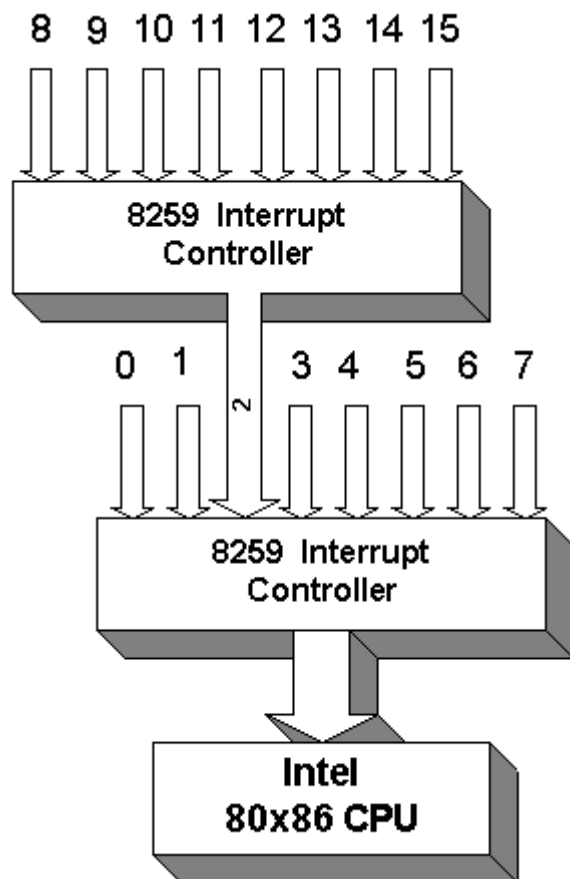
- 1) il PIC rileva una transizione 0-1 su uno dei suoi ingressi IRQ_n, che indica una richiesta d'interruzione da parte di una periferica
 - 2) il PIC pone a 1 il bit del registro IRR (Interrupt Request Register); la richiesta viene passata al blocco PRIORITY RESOLVER che esamina se ci sono interruzioni in servizio con priorità più elevata. In caso affermativo il PIC sospende temporaneamente la richiesta in attesa che la routine d'interrupt a priorità più alta si concluda. Se invece non ci sono interrupt a priorità più elevata in servizio, la richiesta viene inoltrata alla CPU
 - 3) all'arrivo di INTA il PIC setta il corrispondente bit di ISR, resetta il corrispondente bit IRR e invia alla CPU il vettore d'interruzione; il PIC attende ora che la CPU gli invii un comando End of Interrupt (EOI) che gli notifichi la fine della routine di interrupt.
 - 4) all'arrivo di EOI, il PIC resetta il bit di ISR esamina se esiste qualche richiesta d'interrupt pendente (cioè rimasta in sospeso perché a più bassa priorità) e, in caso affermativo la inoltra. Se non ci sono richieste pendenti, si mette in attesa di una successiva richiesta
- Questo modo di funzionamento prende il nome di **Fully Nested Interrupt** perché permette che una routine d'interrupt corrente possa essere interrotta da un'altra, purché a più alta priorità

In cascata

- **Fino a 8 in cascata**
- INTA* va a tutti i dispositivi
- Nella struttura in cascata emette l'interrupt TYPE lo slave che il master seleziona tramite i CASi. L'interrupt type è quindi emesso dallo slave selezionato.
- Il MASTER emette l'interrupt TYPE solo per livelli NON collegati a SLAVES
- La funzione di MASTER o SLAVE può essere definita via hardware (SP/EN*) o via software
- Il meccanismo Special Fully Nested Mode mantiene la struttura di priorità completa anche nella cascata
- Ogni dispositivo (MASTER o SLAVE) deve essere programmato individualmente



USO NEI PC Intel

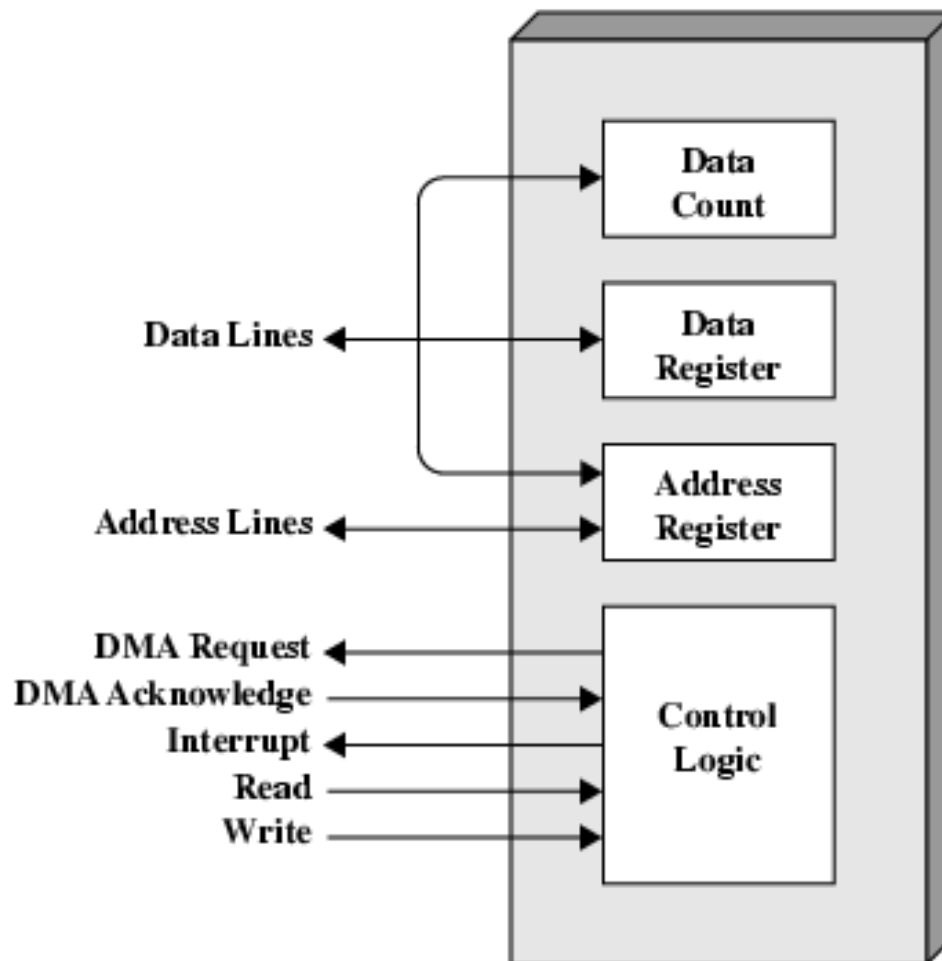


- Nei PC ISA bus vengono usati due 8259 in cascata per avere a disposizione 15 linee di interruzione.
- A cosa sono collegate?
- Sono definite nel BIOS. Sulle macchine Windows è indicato nelle configurazioni di sistema

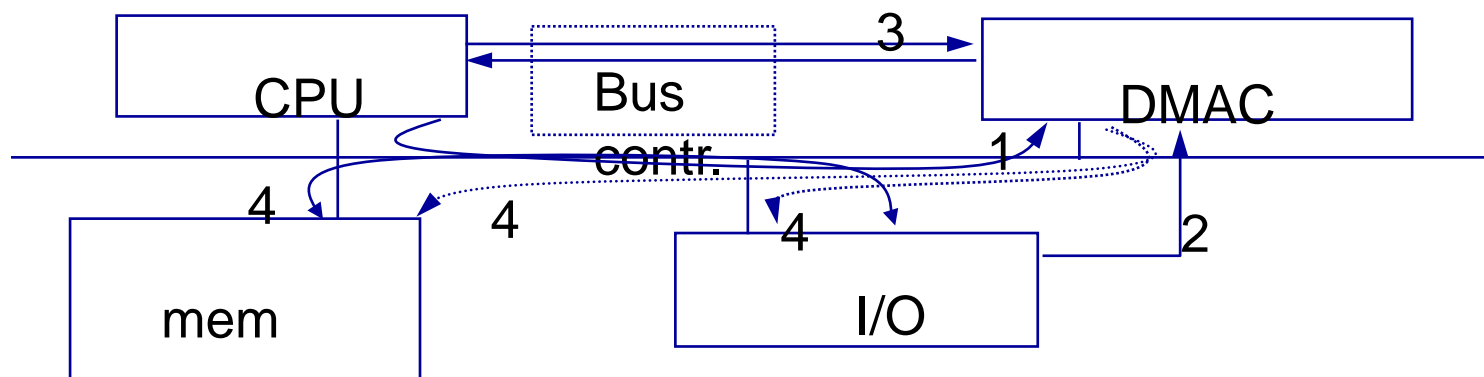
Direct Memory Access

- **DMA** è il meccanismo per cui, sotto controllo hardware del **DMAC** (DMA Controller) una periferica si interfaccia direttamente con la memoria senza passare per la CPU. Il DMAC, similamente alla CPU, prende il controllo del bus e gestisce il trasferimento
- 1. la CPU vede il DMAC come una periferica e la programma con
 - Richiesta di RD o WR
 - indirizzo di I/O e indirizzo di memoria
 - n. di parole da trasferire
- 2. il DMA quando richiesto (via hw o sw) è in grado di
 - indirizzare la memoria e gestire i segnali di controllo
 - indirizzare l'I/O e gestire i segnali di controllo
 - incrementare l'indirizzo in memoria
 - controllare l'arbitraggio per diventare master del bus.

Struttura di un DMAC



DMAC (1/2)

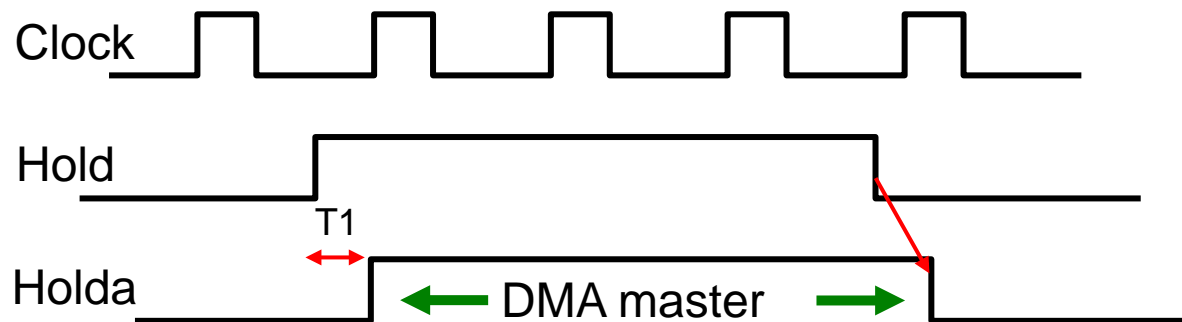


1. La CPU programma il DMAC
 2. arriva la richiesta di trasferimento
 3. CPU e DMAC eseguono l'arbitraggio: il DMA ha il controllo del bus
 4. mentre la CPU non lavora sul bus, il DMAC gestisce indirizzi di memoria ed I/O e avviene il trasferimento; al termine il bus viene rilasciato (di solito una interruzione segnala avvenuto trasferimento)
- **Trasferimento fly-by:** dati non attraversano il DMAC
 - **Trasferimento flow-through:** dati attraversano il DMAC e vengono memorizzati in registri interni del DMAC (permette così il trasferimento anche memory-to-memory)

DMAC (2/2)

Arbitraggio del bus Intel

- Il DMAC deve poter avere accesso al bus ed avere maggiore priorità della CPU
- Architetture Intel arbitraggio con 2 segnali HOLD ed HOLDA
- HOLD non può essere inibito



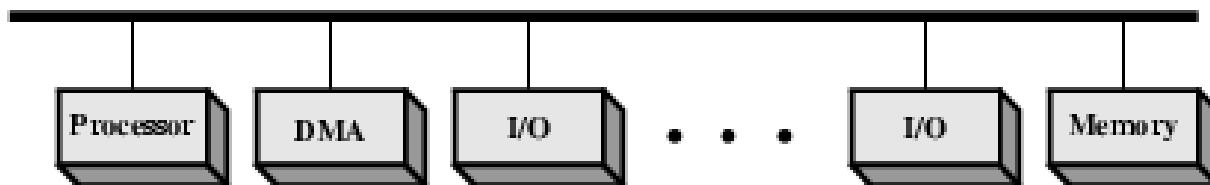
- Quando la CPU asserisce Holda rilascia il bus (indirizzi dati e segnali di controllo in terzo stato)
- Quando il DMA termina il trasferimento rilascia HOLD

Trasferimenti

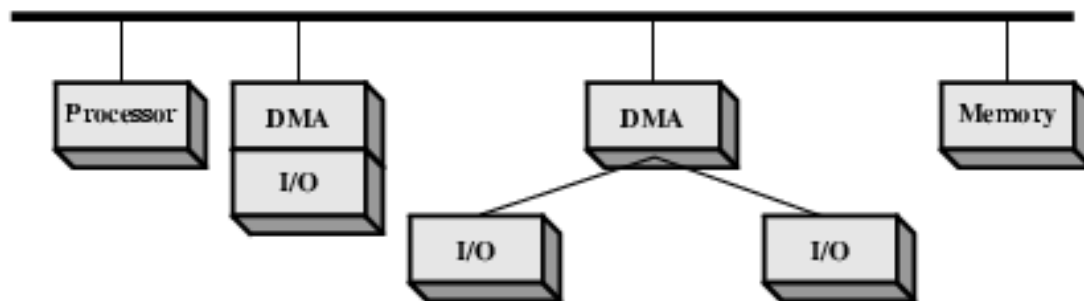
- **Trasferimenti**
 - **singoli** (cicli rubati) ad ogni richiesta si ha il trasferimento di una parola
 - a **blocchi** viene programmato per trasferire blocchi di lunghezza data
 - **on-demand** (Intel) fino a che c'è richiesta si trasferisce (entro un certo limite)
- Se il processore vuole leggere o scrivere un blocco di dati deve in generale:
 - 1) indicare se lettura o scrittura
 - 2) fornire indirizzo del dispositivo di I/O
 - 3) fornire indirizzo di memoria che viene memorizzato dal DMAC
 - 4) indicare il numero di parole
- Poi lavora il DMAC... e manda una interruzione al termine

DMA e I/O channel (1/2)

Con un solo bus di sistema la CPU è interrotta ad ogni ciclo di DMA se lavora in flow through per 2 cicli: uno per I/O e uno per la memoria



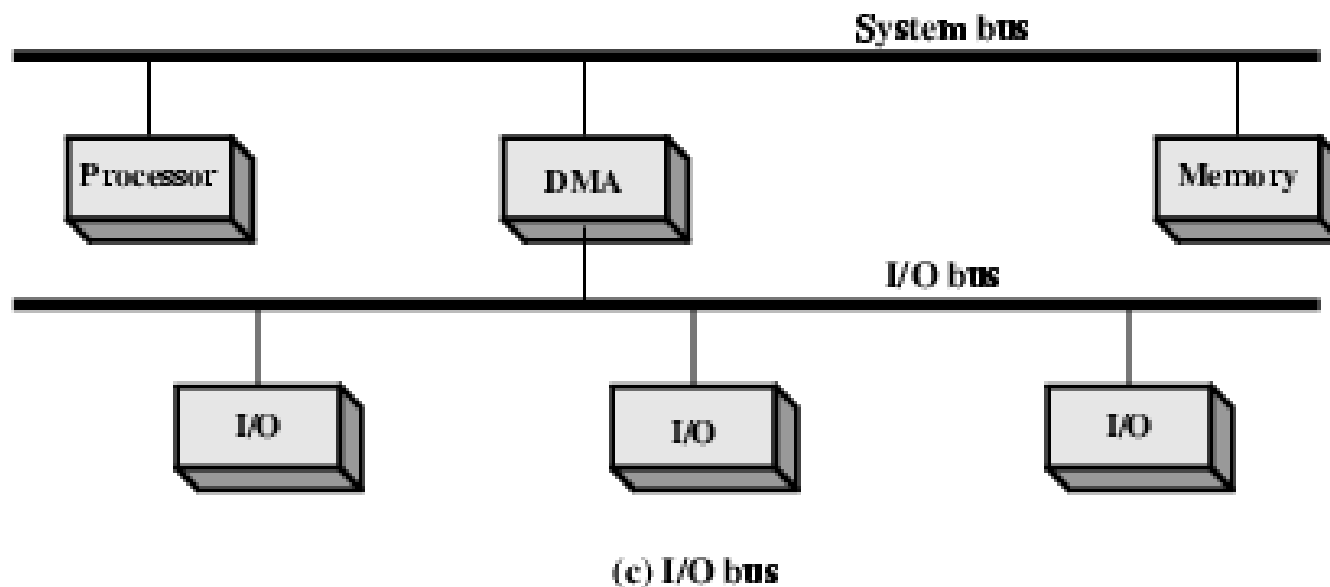
In questo modo, invece, la CPU è interrotta solo se il DMAC va in memoria



(b) **Single-bus, Integrated DMA-I/O**

DMA e I/O channel (2/2)

- Il DMAC fa da bridge per bus di I/O separati
- Vengono poi chiamati I/OP channel o processori di I/O che vengono programmati dalla CPU e hanno una propria memoria locale



MEMORIE ESTERNE

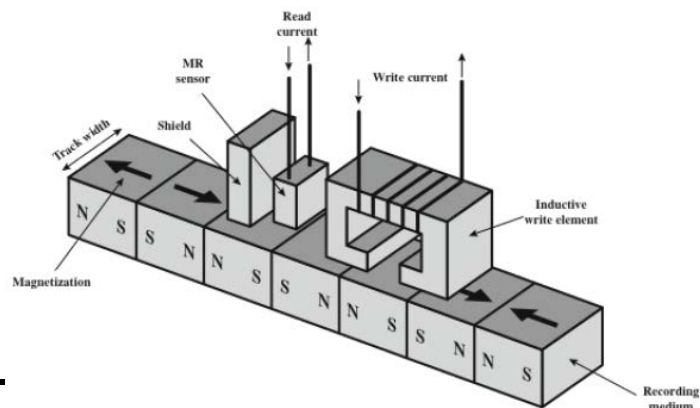
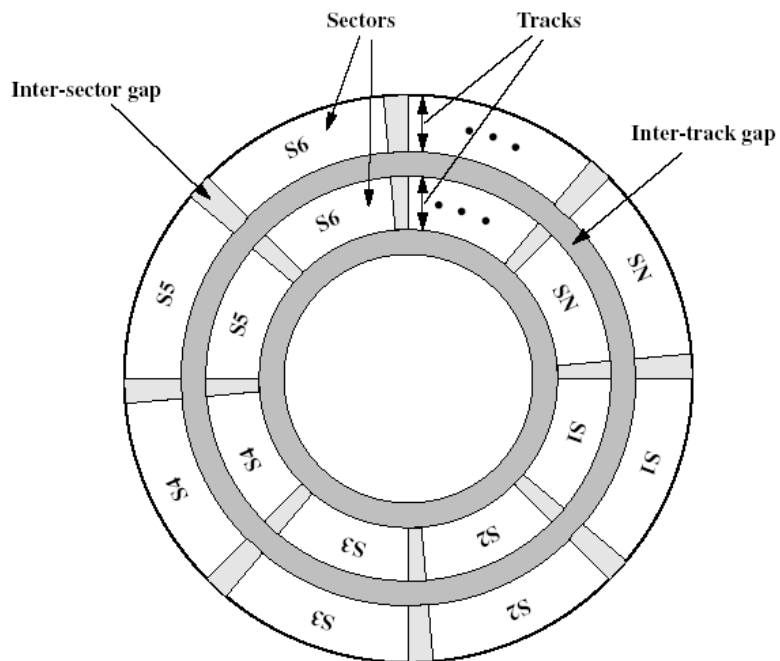
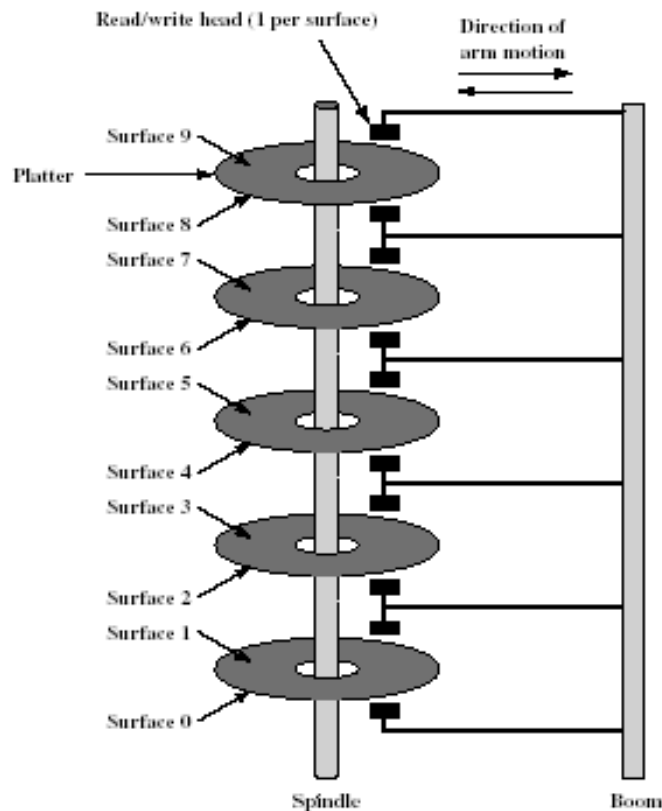
Memorie a disco magnetico

- **Memorie secondarie:**
- In un calcolatore esistono diverse memorie secondarie (o memorie di massa) ad alta capacità, bassi costi e non volatili
- **Ad es: memoria a disco magnetico**
 - L'elemento di memoria è un disco ricoperto di materiale magnetico, su cui esistono una serie di *tracce* concentriche.
 - L'unità di memoria può essere costituita da più dischi: in tal caso essi sono connessi ad un unico asse e ruotano a velocità costante. Ogni superficie (*faccia*) è dotata di una testina in grado di muoversi radialmente fin sulla traccia desiderata. Le varie testine si muovono di solito in maniera solidale.
 - L'insieme delle tracce ad uguale distanza dal centro poste su facce diverse è denominato *cilindro*.
 - Ogni traccia contiene la stessa quantità di informazione, ed è quindi caratterizzata da una densità crescente andando verso il centro. Ogni traccia è organizzata in *settori*, corrispondenti all'unità di trasferimento.
 - Sono essenzialmente di 2 tipi:
 - a disco rigido (hard disk)
 - a disco flessibile (floppy disk).

Funzionamento

- Piatto circolare costituito da uno strato di materiale magnetizzabile depositato su substrato inerte
 - Substrato in alluminio
 - Più recentemente composto vetroso
 - Più uniforme
 - Meno difetti
 - Più rigido
 - Minor distanza testina/disco
 - Maggior resistenza urti (!)
- La testina è caratterizzata da un materiale ferromagnetico
 - Forma a ferro di cavallo con bobina elettrica
 - Testina "ferma", disco ruota
 - Scrittura
 - Tramite bobina induco campo magnetico che agisce su strato magnetizzabile
 - Lettura
 - Strato magnetizzabile in movimento induce corrente su bobina
 - Spesso testine di lettura e scrittura separate

Hard disk



- Testina con solenoide e separata da cuscinetto d'aria ("a tocco" nei floppy)

Hard disk

- Per lavorare la testina deve generare o sentire il campo magnetico
- Più la testina è vicina al disco e più può essere piccola
 - Maggior densità dati
 - Maggior rischio contatto
 - Negli HD due tecnologie
 - Distanza fissa
 - Tecnologia "Winchester", la testina appoggia a disco fermo ma la sua forma aerodinamica la fa planare a disco in movimento
 - Si parla di 10-100 nm
 - Diametro capello umano: 20.000-80.000 nm
 - Inquinamento, PM10: 10.000 nm

Hard disk

- Hard-Disk NEC D2257: dischi e testine sono sigillati in una scatola a tenuta. Si compone di 5 dischi da 8", su cui vengono utilizzate 8 facce, ognuna con la sua testina.
- **Caratteristiche**
 - #tracce per superficie 1024
 - max densità di memorizzazione 9420 bit/inch
 - capacità per traccia 20,480 byte
 - capacità totale 167.7 Mbyte
 - velocità di rotazione 3510 giri/min
 - Average Seek Time 20 ms
- Il tempo di accesso t_A è determinato da:
- T_S : tempo per posizionare la testina sulla traccia opportuna (seek time, o tempo di ricerca);
- T_L : tempo per posizionare la testina sul dato, all'interno della traccia (latency time)
- T_D : tempo per leggere serialmente i dati (data-transfer time, o tempo di trasferimento).
- Si ha quindi che

$$T_{\text{ACCESSO}} = T_S + T_L + T_D$$

- Supponendo di voler trasferire un blocco di 1 kbyte di memoria, il tempo di accesso è pari a $(20 + 17,09 + 0,85) \text{ msec} = 37,94 \text{ msec}$

Futuro degli Hard Disk

- Tecnologie a Stato Solido → SSD
 - Veloci
 - Solo elettronica
 - Azzerano i tempi di latenza
 - Più robuste a danni meccanici
 - Costose → memorie flash
- Tecnologie ibride → SSHD
 - Componente "meccanica" e componente a stato solido
 - Il firmware gestisce il trasferimento dati
 - In un certo senso la componente a Stato Solido si comporta come una "cache"

	NAND Flash Drives	Disk Drives
I/O per second (sustained)	Read: 45,000 Write: 15,000	300
Throughput (MB/s)	Read: 200+ Write: 100+	up to 80
Random access time (ms)	0.1	4–10
Storage capacity	up to 256 GB	up to 4 TB

SSD: problemi

- Le prestazioni decadono col tempo
 - Tutto un blocco deve essere letto alla volta
 - La scrittura richiede la cancellazione completa di un blocco e il suo completo aggiornamento
- Le memorie FLASH diventano inutilizzabili dopo un certo numero di scritture
 - Tecniche per prolungarne la vita:
 - Cache che raggruppi le funzioni di scrittura
 - Algoritmi che "spalmino" le scritture su differenti blocchi
 - Gestione dei "bad blocks"
 - Inoltre alcuni dispositivi sono in grado di segnalare la "vita rimanente"

Dischi ottici

CD

Compact Disk. A nonerasable disk that stores digitized audio information. The standard system uses 12-cm disks and can record more than 60 minutes of uninterrupted playing time.

CD-ROM

Compact Disk Read-Only Memory. A nonerasable disk used for storing computer data. The standard system uses 12-cm disks and can hold more than 650 Mbytes.

CD-R

CD Recordable. Similar to a CD-ROM. The user can write to the disk only once.

CD-RW

CD Rewritable. Similar to a CD-ROM. The user can erase and rewrite to the disk multiple times.

DVD

Digital Versatile Disk. A technology for producing digitized, compressed representation of video information, as well as large volumes of other digital data. Both 8 and 12 cm diameters are used, with a double-sided capacity of up to 17 Gbytes. The basic DVD is read-only (DVD-ROM).

DVD-R

DVD Recordable. Similar to a DVD-ROM. The user can write to the disk only once. Only one-sided disks can be used.

DVD-RW

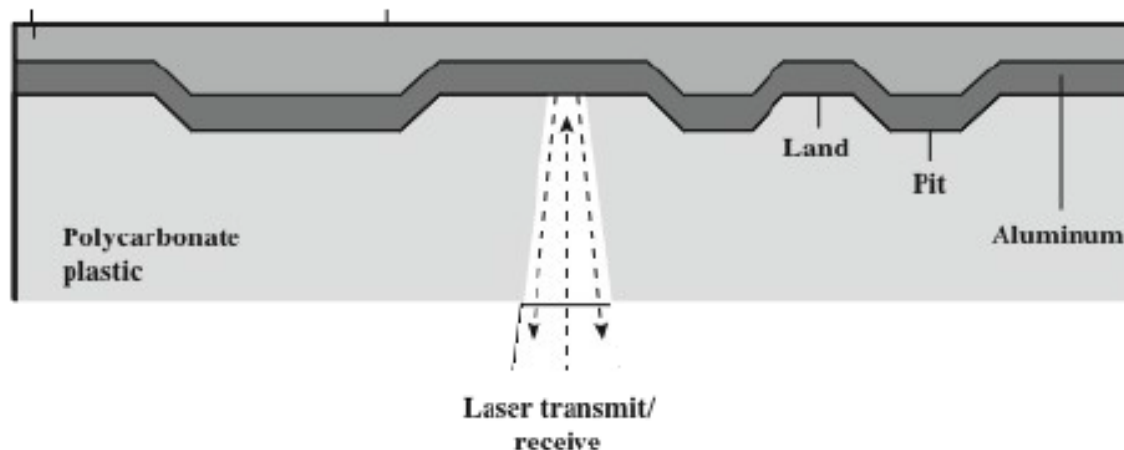
DVD Rewritable. Similar to a DVD-ROM. The user can erase and rewrite to the disk multiple times. Only one-sided disks can be used.

Blu-Ray DVD

High definition video disk. Provides considerably greater data storage density than DVD, using a 405-nm (blue-violet) laser. A single layer on a single side can store 25 Gbytes.

Dischi ottici: tecnologia

- La tecnologia base è quella dei CD
 - Ma: correzioni d'errore più avanzate
- Tecnologia
 - Base di polycarbonato
 - Informazione codificata con *Pit* su substrato metallico riflettente
 - I *Pit* riflettono male
 - Le *Land* riflettono bene



STRUTTURE DI INTERCONNESSIONE

Esigenze

- Differenti moduli che necessitano di interoperare (comunicare)
 - Modello di Von Neumann
- CPU
- Memoria
- Moduli di I/O
- Struttura di interconnessione assicura la comunicazione tra le componenti dell'elaboratore

Interconnessione a bus

- Definizione: il Bus è un mezzo di comunicazione che collega due o più dispositivi
 - Sistema condiviso
 - Segnale trasmesso da un dispositivo ricevibile da tutti
 - Due dispositivi non possono trasmettere in contemporanea
 - Linee di comunicazione a singolo bit, su ogni linea è trasmesso un bit
- Strutture seriali e parallele
- Tipologie di segnali differenti

Elementi di progettazione

Tipo	Ampiezza del bus
Dedicato	Indirizzo
Generico	Dati
Metodo di arbitraggio	Tipo di trasferimento dati
Centralizzato	Lettura
Distribuito	Scrittura
Temporizzazione	Lettura-modifica-scrittura
Sincrona	Lettura-dopo-scrittura
Asincrona	Blocco

- Compromesso costi prestazioni

Alte prestazioni		Bassi costi
dati e indirizzi separati	← largh. di bus	→ bus multiplexati
alto parallel. (64 bit)	← Parallelismo di data bus	→ basso parall. (es 8 bit)
burst	← largh.	→ Trasferimento singolo
multiplo con arbitr.	← master	→ singolo
Split transaction	← impiego del bus	→ no split

Tipi di bus

- Linee dedicate o generiche (multiplexed)
 - Dedicato
 - Assegnata in modo permanente ad una funzione o insieme fisico di componenti
 - Ad esempio separo indirizzi e dati
 - Superabile con utilizzo della linea di controllo
 - Multiplexato (stesse linee per scopi diversi)
 - Mediante controllo segnale cosa trasmetto
 - L'indirizzo di destinazione è posto sul bus
 - Intervallo di tempo limitato per la lettura dell'indirizzo
 - Successivamente l'indirizzo è rimosso dal bus e la connessione è usata per trasferimento dati
- Costi vs Prestazioni
 - Meno linee e quindi costi più bassi, ma più circuiteria, quindi riduco le prestazioni
 - No contemporaneità di utilizzo

Metodo di arbitraggio

- Più di un modulo può aver bisogno del bus, solo uno può usarlo, occorre arbitrare
- Centralizzato
 - Un solo dispositivo gestisce il controllo del bus
 - Tipicamente CPU o dispositivo dedicato (detto controllore o arbitro)
- Distribuito
 - I dispositivi cooperano per accesso
 - Logica per l'accesso al controllo: algoritmo di accesso
 - Devo evitare collisioni

Temporizzazione

- Modo in cui gli eventi sono coordinati sul bus
- **Bus sincrono**
 - Un segnale di clock scandisce tutte le operazioni (una linea del bus dedicata al clock)
 - Fronte di salita
 - Attenzione ai ritardi
 - Segnali di stabilizzazione
 - Bus cycle o clock cycle: singola trasmissione della sequenza 1-0, unità di tempo
 - PRO: semplicità di progetto e di controllo
 - CONTRO: spreco di tempo → ogni operazione si deve svolgere in un numero intero di cicli di clock
- **Bus asincrono**
 - Ogni operazione viene innescata dalla precedente
 - PRO: maggior efficienza nell'uso dei cicli → l'operazione si completa esattamente nel tempo necessario
 - CONTRO: complessità di progetto e di controllo.
- I BUS dei calcolatore sono nella maggior parte dei casi di tipo sincrono.

Ampiezza del bus

- Bus dati: numero di bit trasferibili in contemporanea
 - velocità
- Bus indirizzi: spazio indirizzamento
 - Capacità del sistema = massimo intervallo di locazioni indirizzabili
- A logica maggior parallelismo → maggior velocità
 - Ma allora perché quasi tutti i bus "moderni sono seriali"?
 - Troppe linee, routing complicato e occupa troppo spazio
 - Con bus seriali posso fare linee più lunghe, ho meno interferenza tra linee

Tipi di trasferimento dati

- Tutti i bus permettono lettura e scrittura
- Tipologie particolari di funzionamento
 - Read modify write
 - Lettura seguita da scrittura sullo stesso indirizzo, trasmesso una sola volta
 - Operazione indivisibile
 - Read after write
 - Operazione indivisibile
 - Realizzata come forma di controllo
 - Servono per casi specifici
 - Sistemi multiprogrammati
 - Sistemi multiprocessori
 - Trasferimento a blocchi
 - Un ciclo di indirizzo è seguito da n cicli di dati (ciascun dato dopo il primo farà riferimento ad indirizzi successivi)

Organizzazione fisica



Bus di sistema

- In un calcolatore, bus con caratteristiche diverse che forniscono cammini diversi, a vari livelli della gerarchia di sistema
- Causa prestazioni → più bus
 - Gerarchie di bus
- Bus di sistema connette i principali elementi di un calcolatore
 - CPU
 - Memoria
 - I/O

Gerarchia di bus

VELOCITA'

- **Processor Bus:** bus interno alla CPU di alto livello utilizzato per inviare le informazione da e per i registri della CPU
- **Cache Bus:** bus dedicato per l'accesso alla cache di Sistema
- **System Bus:** bus (dati, indirizzi e controllo) che connette, principalmente, il processore con le memorie
- **Local I/O Bus:** bus I/O ad alta velocità usato per collegare periferiche critiche in termini di prestazioni alla memoria e al processore (es: schede video, hard disk – es: bus PCI)
- **Standard I/O Bus:** bus usato per periferiche più lente (es: bus ISA)

Gerarchia di bus

Bus	Bus Width (bits)	Bus Speed (MHz)	Bus Bandwidth (MBytes/sec)
8-bit ISA	8	8.3	7.9
16-bit ISA	16	8.3	15.9
EISA	32	8.3	31.8
VLB	32	33	127.2
PCI	32	33	127.2
64-bit PCI 2.1	64	66	508.6
AGP	32	66	254.3
AGP (x2 mode)	32	66x2	508.6
AGP (x4 mode)	32	66x4	1,017.3

Architetture precedenti - bus unico PDP

- Approccio usato fin da PDP-11 → PC
- Vantaggi
 - Modulare
 - Facilmente standardizzato
 - ISA/EISA
 - Industry Standard Architecture (ISA) è un bus parallelo destinato al collegamento di generiche schede d'espansione (schede video, schede audio, schede di rete, ecc.) di norma installate internamente al computer.
 - Extended Industry Standard Architecture (EISA) è la prima evoluzione dello standard ISA, larghezza di bus maggiore
- Svantaggi:
 - Condivido dispositivi "lenti" e "veloci"
 - Memoria e mouse hanno richieste differenti
 - All'aumentare dei dispositivi → bus lungo
 - Lentezza di propagazione segnali
 - Limite ampiezza
 - In realtà non esclusivo dei bus unici
 - Significativi i costi per aumentare prestazioni

Architetture precedenti - ISA

- **ISA** (Industry Standard Architecture)
- Utilizzato nel PC IBM originale nella versione a 8 bit e 4.77 MHz, la stessa banda dati e velocità del processore Intel 8088.
- Nel 1984 sul PC IBM AT viene introdotta la versione a 16 bit e 8.3 MHz, la stessa banda dati e velocità del processore Intel 80286.
- Utilizzato sulle motherboard fino al Pentium per l'utilizzo di schede che non richiedono prestazioni particolarmente elevate (es. modems)

Architetture precedenti - MCA

- **MCA** (Micro Channel Architecture)
- Introdotto nel 1987 da IBM per sostituire il Bus ISA ed adattarsi alle caratteristiche del processore Intel 80386, che aveva un bus dati a 32 bit
- Aveva caratteristiche simili a quelle che avrebbe avuto il Bus PCI ben 7 anni dopo: 32-bit Bus width, Bus Mastering e Plug'n'Play. Non ebbe successo principalmente per due motivi:
 - incompatibilità con lo standard ISA
 - era un bus “proprietario” di IBM che lo utilizzo sul PS/2
- E' stato comunque utilizzato su alcune macchine RISC IBM.

Architetture precedenti - EISA

- **EISA** (Extended ISA)
- Fu la risposta di COMPAQ al Bus MCA di IBM, e diversamente dal MCA era compatibile con il Bus ISA e non era “proprietario”.
- Ha caratteristiche molto simili al MCA: 32-bit Bus width, Bus Mastering, Plug’n’Play
- Utilizzato su macchine dedicate (es. alcuni server di rete), ma praticamente inesistente nel mondo dei PC:
 - molto più costoso di altri tipi di sistema
 - poche schede di tipo EISA disponibili sul mercato
 - prestazioni inferiori rispetto ad altri sistemi come VESA Local Bus e PCI

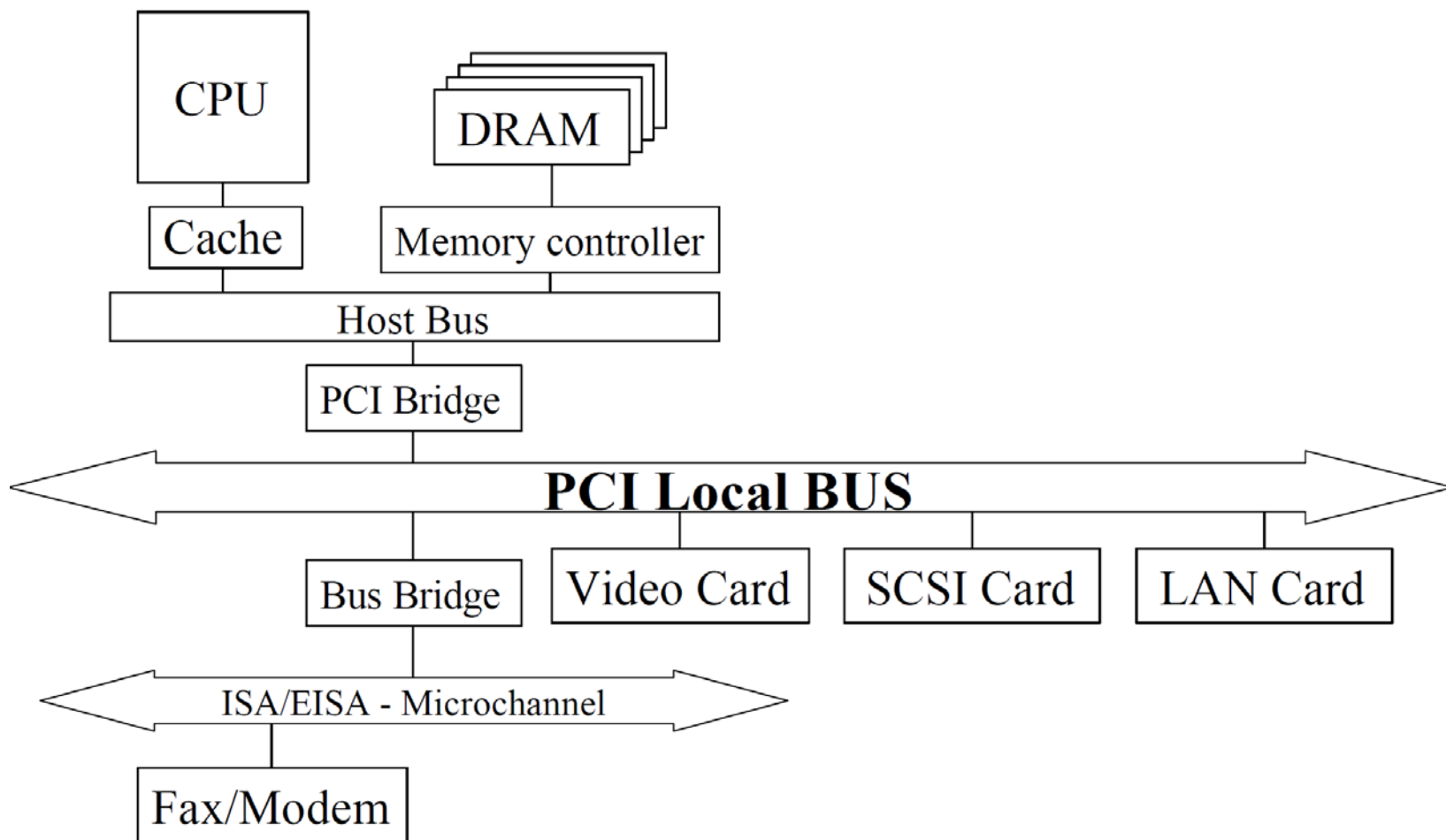
Architetture precedenti - VLB

- **VLB - VESA** (Video Electronics Standards Association) **local bus**
- Fu introdotto nel 1992 come diretta estensione del bus ISA e fu il primo Bus Locale ad avere una certa diffusione.
- Introdotto per migliorare le prestazioni video dei PC, aveva una larghezza di banda a 32 bit ed una velocità di 33 MHz. Su alcuni sistemi sono possibili velocità maggiori.
- L'uso di schede video e di I/O controller di tipo VLB aumentava notevolmente le prestazioni dei sistemi basati sul Bus ISA
- Il VLB fu abbandonato con l'introduzione del processore Pentium e del bus PCI, in quanto essendo fortemente basato sull'architettura del 80486 creava problemi di compatibilità sui Pentium. Inoltre difficilmente gestiva più di una o due schede, non supportava il Bus Mastering e il Plug 'n' Play.

Il bus PCI

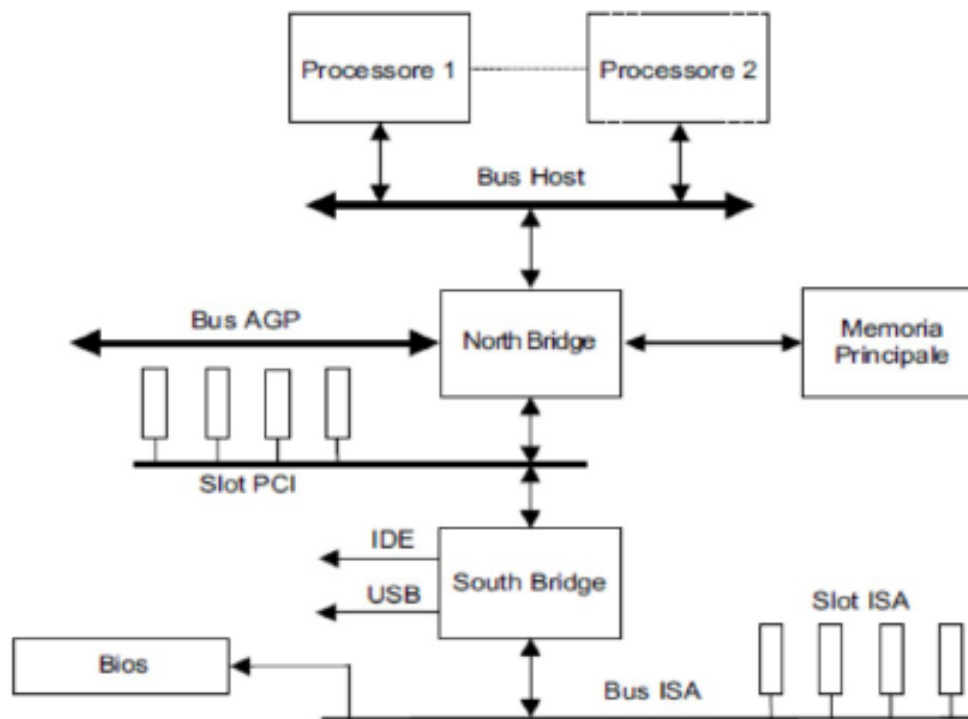
- **PCI** (Peripheral Component Interconnect)
- Sviluppato da INTEL e introdotto nel 1993 su PC di classe Pentium e nell'ultima generazione di PC basati sul processore 80486.
- Non è legato al tipo di piattaforma sul quale viene impiegato.
- Fornisce funzionalità Plug'n'Play e Bus Mastering grazie all'utilizzo di un chipset dedicato.
- Standard amministrato dal PCI Special Interest Group

Il bus PCI



Esempio PCI/AGP

- Bus ISA diventa bus per periferici lenti
- Due componenti rilevanti, North Bridge e South Bridge:
 - adattano livelli di bus diversi
 - rappresentano potenzialità del sistema



PCI

- Intel lo ha rilasciato come pubblico
 - Associazione industriale
- 32 o 64 bit
- 49 linee obbligatorie, 51 linee opzionali
- Fino a 66MHz
- 4,2 Gbps



PCI: prestazioni

- **Burst Mode:** Dopo aver individuato l'indirizzo iniziale, un molteplice flusso di dati può essere trasferito in un colpo a burst
- **Bus Mastering:** Il collegamento peer-to-peer sul Bus, accesso diretto tra memoria centrale, dispositivi PCI e PCI Bridge, è assicurato dal Bus Arbiter centralizzato.
- **Funzionamento sincrono e asincrono:** Nel funzionamento sincrono il Bus va a metà della velocità del bus di sistema, ad es. System bus a 66 MHz => PCI bus 33 MHz
- **PCI Expansion slots:** Sul bus PCI c'è la possibilità di inserire più schede di espansione attraverso connettori di tipo "Edge" (molto economici)
- **Linee multiplexate:** Stesse linee fisiche usate da più linee logiche
- **PCI IDE Bus Mastering:** Il controller dei dischi collegato al Bus PCI, fornisce funzionalità DMA (Direct Memory Access)

PCI: burst mode

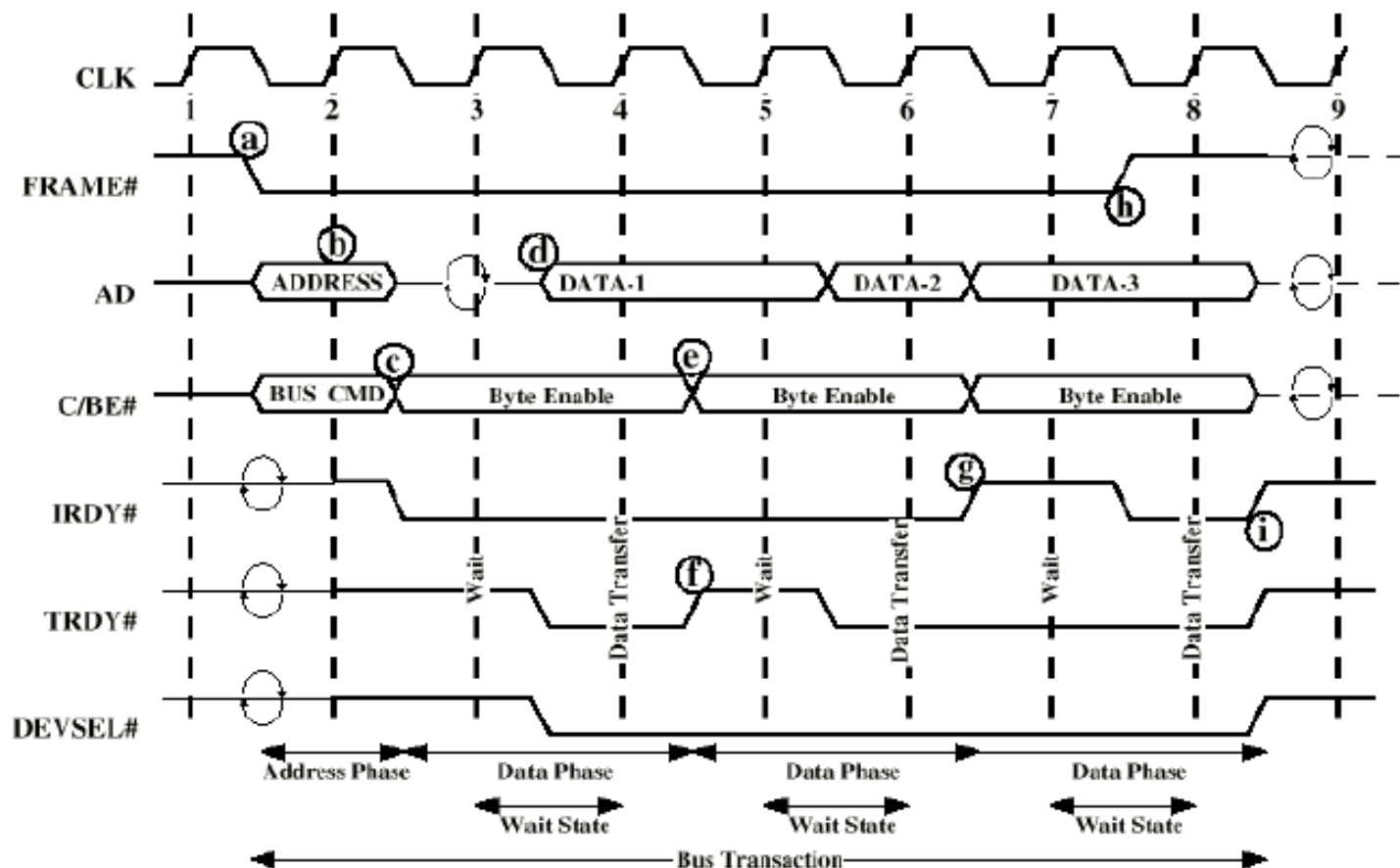
- Il trasferimento dati con le periferiche PCI avviene secondo la modalità “Burst”
- Significa che una volta ottenuto lo stato di “master” sul bus da parte di una periferica (**initiator**), questa trasferisce un flusso di dati la cui lunghezza viene stabilita all’inizio del trasferimento stesso verso il dispositivo di destinazione (**target**).
- Un solo dispositivo alla volta può assumere lo stato di bus master, e non può essere interrotto dagli altri.
- Si ottiene così un velocità di picco nel trasferimento dati di 132 MB/s (32 bit @ 33 MHz), 528 MB/s (64 bit @ 66 MHz - specifiche PCI 2.1)

PCI: bus multiplexato

- L'utilizzo di linee Dati e Indirizzi multiplexate permette di utilizzare dei connettori con un numero inferiore di contatti, con vantaggi sia in termini di costi, sia in termini di dimensione dei connettori.
- Le tipiche schede PCI a 32 bit utilizzano circa 50 linee, di cui 32 sono le linee dati e indirizzi multiplexate AD[31:0].
- **Address phase:** I cicli di Bus PCI iniziano con un ciclo di clock nel quale viene messo sulle AD[31:0] un indirizzo. Questa fase è segnalata dall'attivazione del segnale FRAME#.
- **Data phases:** Al ciclo di clock successivo inizia il trasferimento dei dati su AD[31:0]. Può durare più cicli di clock.

PCI: esempio di transazione

- Operazione di lettura
- CLK=clock; FRAME# = inizio e durata transazione; IRDY# = initiator ready
- TRDY# = target ready; C/BE# = comando di bus (destinazione)
- DEVSEL# = selezione dispositivo



PCI: esempio di transazione

L'*initiator* richiede lo stato di master all'arbitro centrale asserendo il segnale REQ#, e riceve l'assegnazione sulla linea GNT#.

(a) Preso il controllo, l'*initiator* asserisce FRAME#, attiva i segnali C/BE[3:0]# durante la address phase per segnalare al sistema il tipo di transazione (memory read, memory write, I/O read, I/O write).

(b) Il *target* riconosce proprio indirizzo su linea AD

(c) *Initiator* smette uso AD e prepara uso bus per *target*, asserisce IRDY# per indicare che dato è pronto

(d) *Target* asserisce DEVSEL#, posiziona dati su AD e asserisce TRDY#. Si ha un trasferimento valido solo quando sia IRDY# che TRDY# sono asseriti

(e) Dati letti viene cambiato abilitazione byte

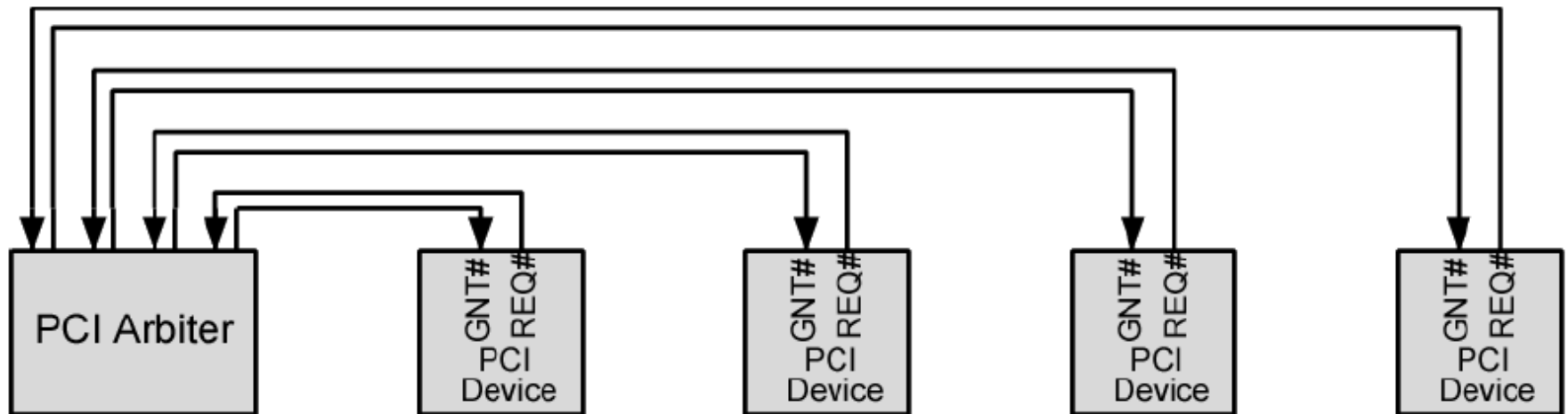
(f) *Target* non ha dati ancora disponibili, disabilita TRDY#

(g) *Target* pone terzo blocco ma *initiator* non ancora pronto per leggerli

(h) *Initiator* segnala *target* che è ultimo blocco e annulla FRAME#

(i) *Initiator* ha letto tutto e disasserisce anche IRDY#

PCI: arbitraggio centralizzato

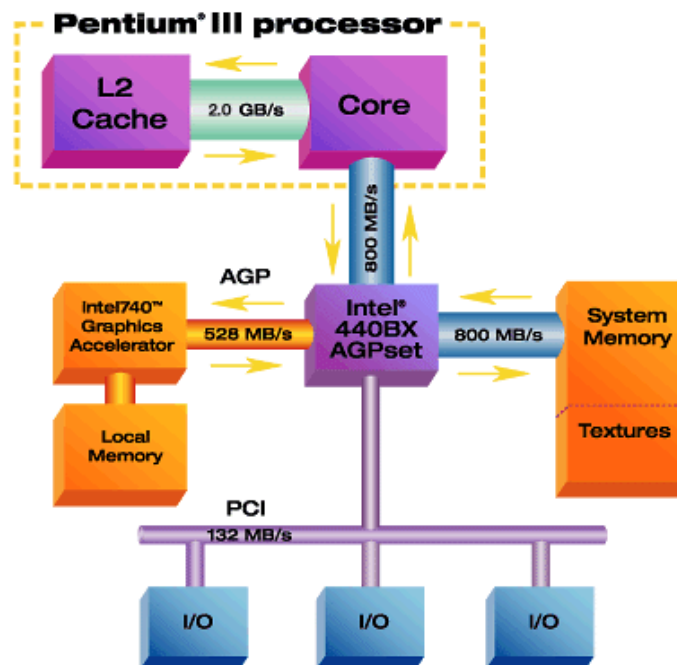


Evoluzione del PCI

Year	Bandwidth	Frequency/Speed
1992	133MB/s (32 bit simplex)	33 Mhz (PCI)
1993	533MB/s (64 bit simplex)	66 Mhz (PCI 2.0)
1999	1.06GB/s (64 bit simplex)	133 Mhz (PCI-X)
2002	2.13GB/s (64 bit simplex)	266 Mhz (PCI-X 2.0)
2002	8GB/s (x16 duplex)	2.5 GHz (PCIe 1.x)
2006	16GB/s (x16 duplex)	5.0 GHz (PCIe 2.x)
2010	32GB/s (x16 duplex)	8.0 GHz (PCIe 3.x)
2017	64GB/s (x16 duplex)	16.0 GHz (PCIe 4.0)
2019	128GB/s (x16 duplex)	32.0 GHz (PCIe 5.0)

AGP: Accelerated Graphics Port

- L'interfaccia AGP viene introdotta da Intel su architetture basate sul processore Pentium II, per l'utilizzo di applicazioni di grafica 3D e digital video.
- Infatti, il Bus PCI non ha l'ampiezza di banda sufficiente (132 MB/s) per le "nuove" applicazioni 3D ed la scheda grafica deve dividerlo con altri dispositivi.



AGP: Accelerated Graphics Port

- Il Bus AGP è fisicamente separato dal Bus PCI ed offre nuove funzionalità come l'accesso alla memoria di tipo pipeline dedicato.
- Il Bus PCI, liberato dal traffico generato dalla scheda video, può ora essere utilizzato dalle schede di rete veloci (100Mb/s), dai controller Ultra DMA, e da altri dispositivi ad alta velocità.
- Le schede AGP hanno accesso diretto alla memoria centrale, nella quale possono venir memorizzati dati (texture) di grandi dimensioni, senza quindi occupare memoria grafica.
- Le specifiche AGP oggi permettono una velocità di trasferimento dati fino a 1 GB/s nella modalità AGP 4x.

Seriale vs Parallelo

- Il parallelismo fa aumentare prestazioni
- Ideale se più dispositivi
 - Chiaramente devo risolvere contese
 - Costoso se bus lungo
- Ma quindi perché si sta abbandonando il parallelismo?
 - Connessioni seriali punto a punto
 - Sicuramente elimino esigenze arbitraggio
 - Timing skew: rappresenta un limite massimo all'utilizzo di bus paralleli

