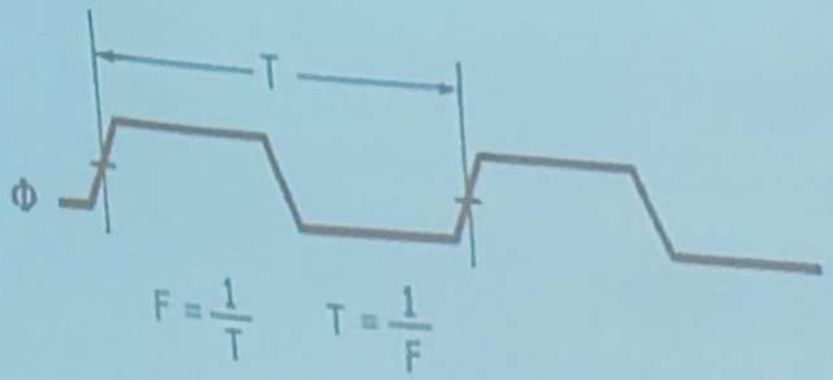
Temporizzazione del bus

- Dal punto di vista della temporizzazione esistono due differenti approcci: **bus sincroni** e **asincroni**.

- I bus sincroni utilizzano un clock che determina la temporizzazione delle attività sul bus (**ciclo di bus**): ogni operazione richiede un numero di periodi di clock per essere eseguita.

Temporizzazione del bus

- Per questioni di affidabilità e retrocompatibilità i bus moderni operano a velocità di clock non elevata (per esempio il bus PCI ha una frequenza di clock nell'intervallo 33 ÷ 66 MHz).



Se la frequenza di clock è 100 MHz il ciclo di bus ha un periodo di 10 ns

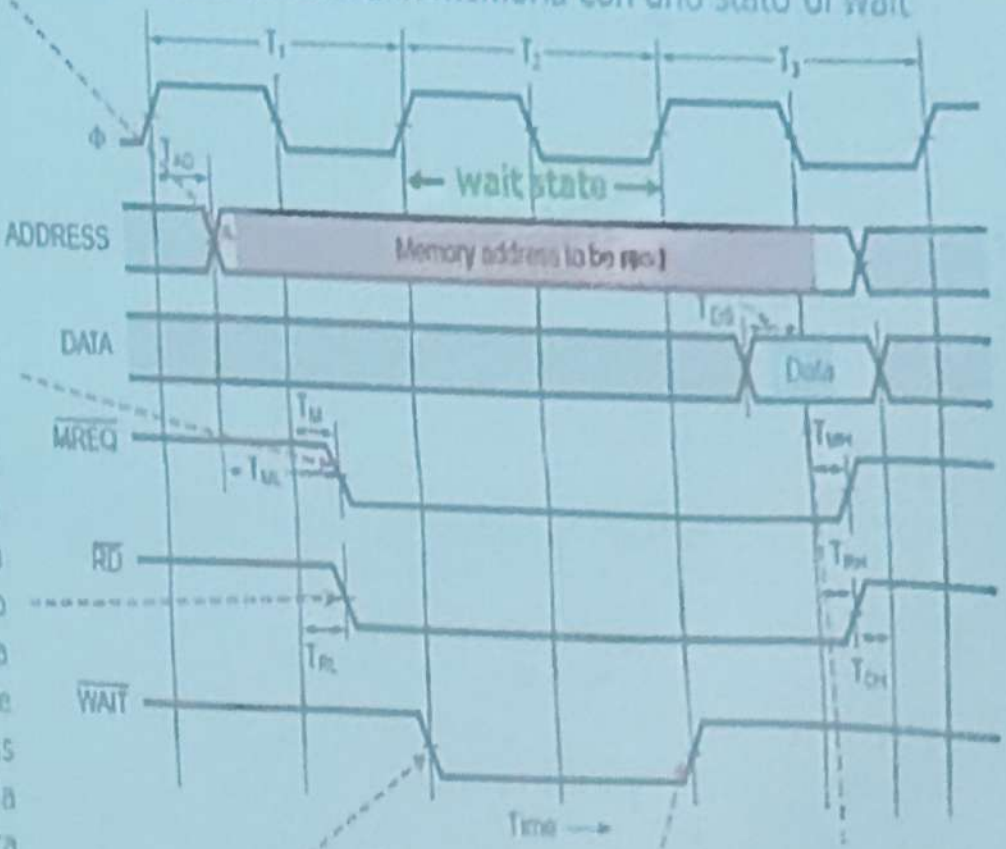
- Il bus asincrono invece il ciclo di bus può avere qualsiasi durata e quindi è più efficiente. Tuttavia è più difficile da costruire rispetto ad uno sincrono.

Bus sincrono

1) La CPU (**master**) pone l'indirizzo di memoria sullo address bus in modo che le linee si stabilizzino.

2) La CPU comunica al sistema che l'operazione che intende svolgere è con la memoria.

3) La CPU comunica che si tratta di una operazione in lettura, a questo punto la memoria (**slave**) deve fornire sul data bus il contenuto della cella indirizzata dall'address bus.

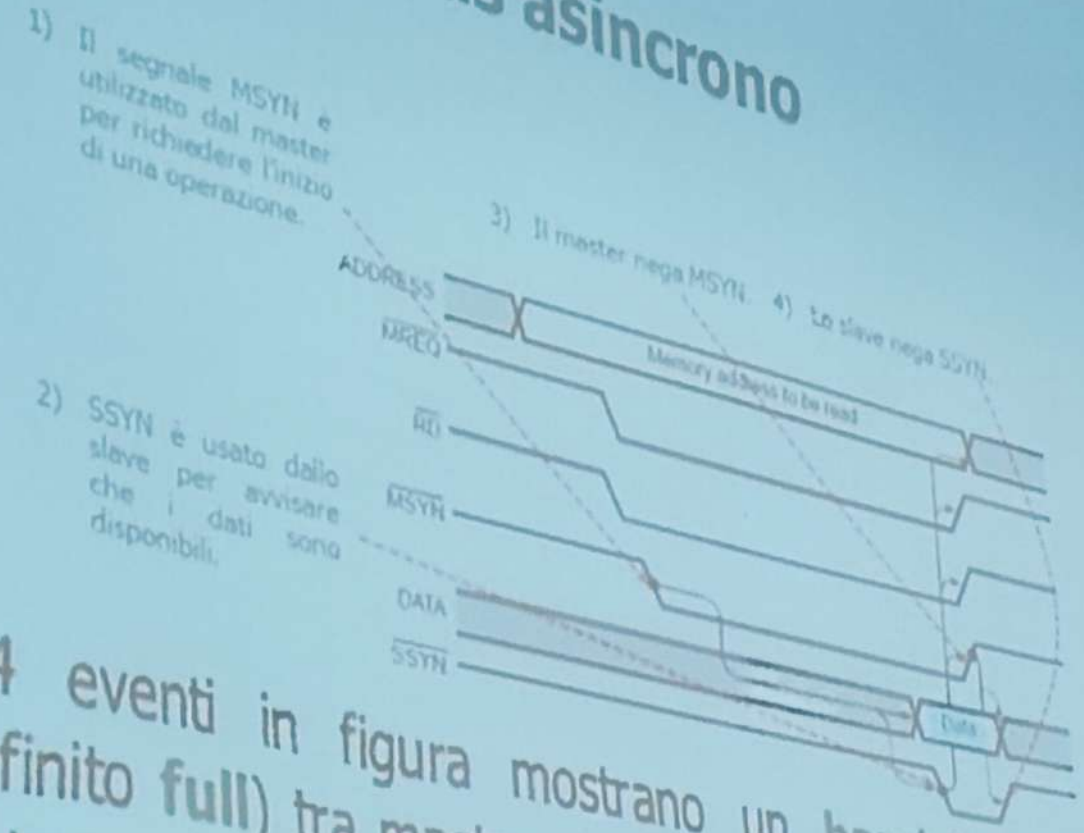


4) Poiché la memoria è meno veloce della CPU, le chiede uno stato di attesa aggiuntivo (**wait state**).

5) La memoria è pronta a fornire i dati allora nega il segnale di WAIT e la CPU potrà leggere i dati.

6) La CPU legge i dati sul fronte in discesa di T_3.

Bus asincrono



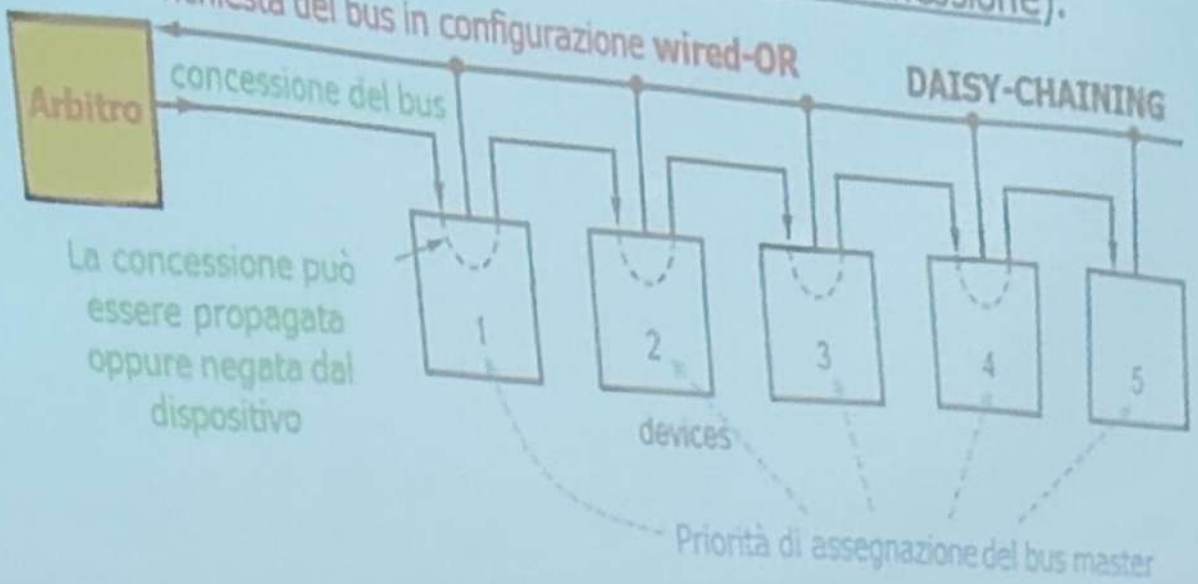
- I 4 eventi in figura mostrano un **handshake** completo (definito **full**) tra master e slave, il tutto avviene senza base dei tempi.
- In questo modo il bus si adatta alla velocità del dispositivo collegato ed un dispositivo lento non rallenta l'intero sistema!
- La maggior parte dei bus sono sincroni perché più semplice da realizzare e a causa degli enormi investimenti fatti fin ora.

Arbitraggio del bus

- Ciascun dispositivo "intelligente" del computer (CPU, coprocessor, I/O devices,...) può diventare, a turno, master del bus.
- L'arbitraggio del bus è utilizzato per prevenire situazioni di conflitto in cui due o più dispositivi tentano di diventare, nello stesso momento, master del bus.
- L'arbitraggio può essere **centralizzato** o **decentralizzato**.
- Il primo necessita della presenza di un **arbitro**, normalmente integrato nel chip della CPU, che può diventare anche un fattore di criticità in caso di rottura.

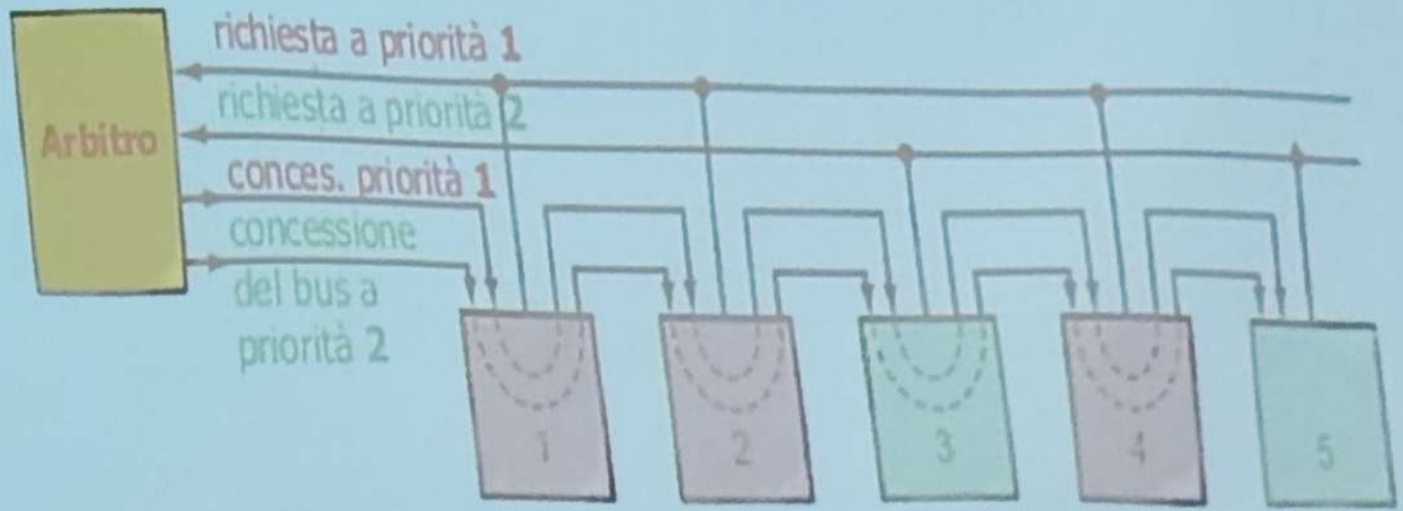
Arbitraggio centralizzato del bus

- Quando l'arbitro riceve una **richiesta**, concede la richiesta asserendo una linea di **concessione del bus**.
- Quando il dispositivo più vicino all'arbitro vede la concessione:
  - Se è lui che lo ha chiesto, blocca la line negandola a tutti i sui successori;
  - Altrimenti mantiene asserita la linea.
- Quando due, o più, dispositivi fanno richiesta vince il più vicino all'arbitro (la priorità è nel cablaggio della connessione).



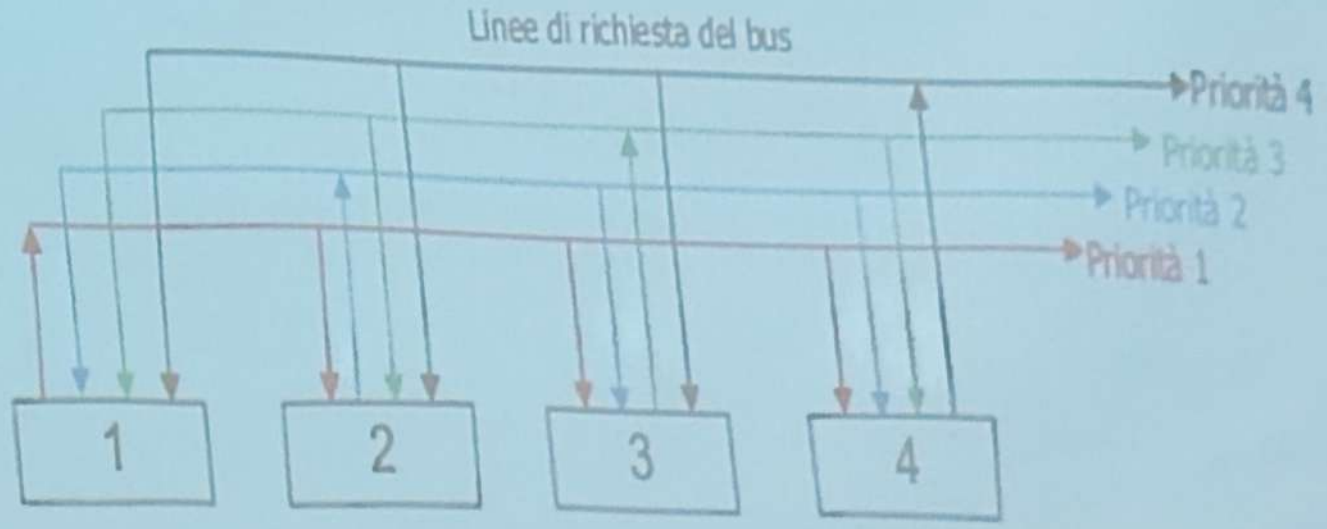
Arbitraggio con più linee di priorità

- Per superare il problema che la priorità è cablata nella connessione molti bus definiscono più livelli di priorità utilizzando differenti linee richiesta-concessione (4, 8 o 16).
- In questo caso l'arbitro concede il bus al dispositivo con la priorità più alta, a parità di priorità vince chi è più vicino all'arbitro nella catena (daisy-chain).



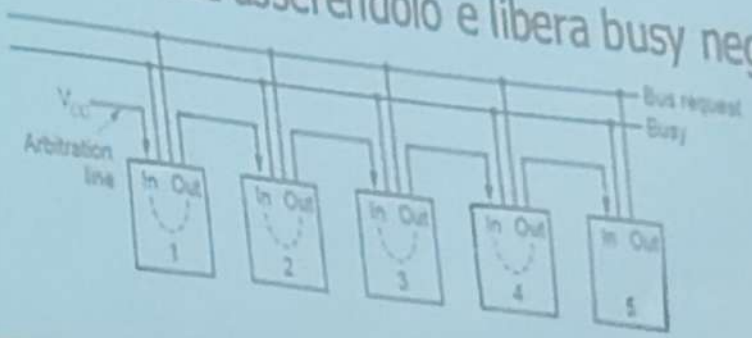
Arbitraggio decentralizzato del bus

- Ogni dispositivo ha una propria linea di richiesta ed una priorità.
- Prima di richiedere il bus ciascuno deve verificare che non ci sia già una richiesta con priorità più alta.
- Al termine dell'utilizzo del bus, la linea di richiesta deve essere negata.
- Svantaggi: troppi collegamenti, il numero di dispositivi non può superare il numero di linee.



Arbitraggio decentralizzato del bus

- Un differente schema decentralizzato di arbitraggio utilizza tre linee:
  - La linea di **richiesta del bus** (in configurazione wired-OR).
  - La linea the **busy** asserita dal dispositivo bus master corrente.
  - La linea di **arbitraggio** propagata tra i dispositivi in cascata.
- Per ottenere il bus un dispositivo deve:
  - Controllare che busy sia negata e che l'ingresso **In** sia asserito.
  - In questo caso nega **Out**, asserisce la linea busy e diventa il master del bus.
- Al termine, sblocca **Out** asserendolo e libera busy negandolo.

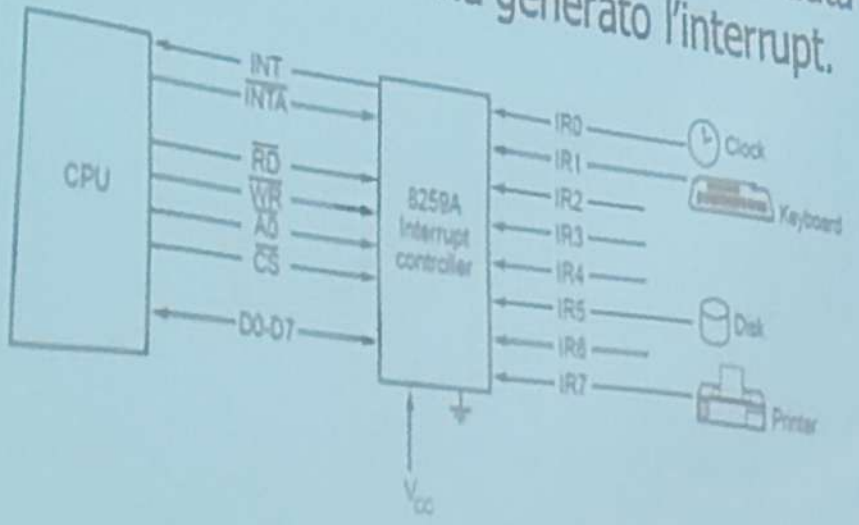


Interrupt handling

- Esistono altre operazioni che possono essere eseguite sul bus come ad esempio la gestione delle interruzioni.
- Se la CPU comanda ad un dispositivo di I/O di svolgere un lavoro, non rimane in attesa dello svolgimento ma si aspetta un interrupt nel momento in cui il dispositivo ha terminato il suo compito.
- I dipositivi possono generare più interruzioni nello stesso momento, per questa ragione sono utilizzati dei chip dedicati per svolgere la funzione di arbitro centrale ed assegnare la corretta priorità ai vari dispositivi.
- Il PC basati su processore Intel il chipset incorpora il controllore 8259A.

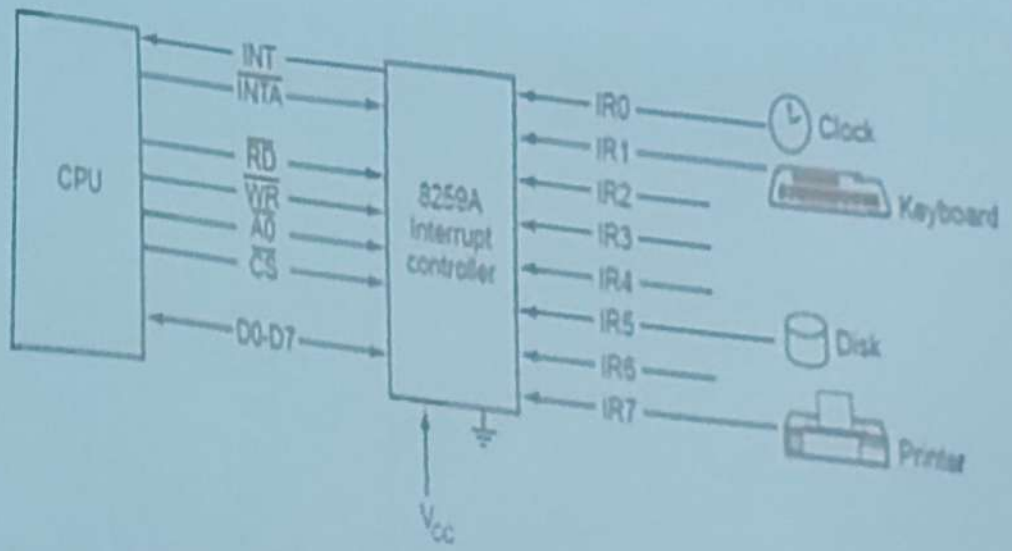
Interrupt handling

- Il chip 8259A può ricevere fino ad 8 richieste di interrupt ($IR_0 \div IR_7$).
- Quando uno o più dispositivi richiede una interruzione l'8259A asserisce il segnale **INT** (INTerrupt) alla CPU.
- Se la CPU è in grado di gestire la richiesta di interruzione, risponde al controllore con il segnale **INTA** (Acknowledge signal). A questo punto l'8259A deve porre sul data bus il numero del dispositivo che ha generato l'interrupt.



Interrupt handling

- La CPU utilizza quel numero come indice per accedere al **vettore di interruzione** e trovare l'indirizzo della **ISR** (Interrupt Service Routine).
- Per gestire più di 8 dispositivi i controllori 8259A possono essere collegati in cascata.

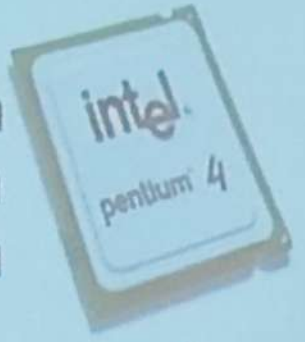


Argomenti

- ESEMPI DI CPU
  - Intel Pentium 4
  - Intel Core i7
  - UltraSPARC III
  - Intel 8051
- ESEMPI DI BUS
  - Bus ISA
  - Bus PCI
  - PCI Express
  - USB
- INTERFACCE
  - Parallel I/O (PIO)
  - Decodifica dell'indirizzo

Pentium 4

- Il Pentium 4 è un diretto discendente della CPU 8088 utilizzata nei primi PC IBM.
- Contiene **55** millioni di transistor (la **larghezza di linea** cioè la distanza tra transistor è di **90** nm), ed opera ad una frequenza di clock fino a 3,2 GHz.
- È in grado di scambiare dati con la memoria a 64-bit ma dal punto di vista software il programmatore vede una macchina a 32-bit (compatibile con i vecchi software scritti per 80386, 80486, Pentium,...).



Un capello ha un diametro circa di:
- 20 µm se chiaro
- 100 µm se scuro.

Quindi **1÷4,5 millesimi** di un capello!!

Pentium 4

- La microarchitettura interna (**NetBurst**) è molto diversa dai sui predecessori:
  - Ha una pipeline più profonda delle precedenti architetture.
  - Due unità ALU che operano al doppio della frequenza di clock.
  - Supporta l'hyperthreading.
  - Ha due insiemi di registry che permettono ai programmi di funzionare come se ci fossero due CPU fisiche.

- A seconda del modello può avere 2 o 3 livelli di cache.

Pentium 4

- Tutti i modelli hanno 8 KB di SRAM sul chip per la cache di primo livello (**L1**).
- Il secondo livello di cache (**L2**) è in grado di memorizzare fino a 1 MB.
- L'Extreme Edition ha anche 2 MB per la cache di terzo livello (L3).
- Ogni CPU mantiene la consistenza tra le cache attraverso il processo di **snooping** dei riferimenti di memoria sull'address bus.

Pentium 4

- Il Pentium 4 ha due bus sincroni primari.
- Il **memory bus** è utilizzato per accedere alla memoria principale (S)DRAM.
- Il **bus PCI** è utilizzato per il colloquio con i dispositivi di I/O.
- Un problema comune a tutte le attuali CPU è il consumo energetico ed il calore prodotto: il Pentium 4 consuma tra **63÷82** W in funzione della frequenza di funzionamento.
- Intel è constantemente alla ricerca di modi per gestire il calore prodotto dalle proprie CPU.

Gli stati di funzionamento del Pentium 4

- In accordo con le leggi della fisica qualsiasi dispositivo elettronico che produce molto calore deve assorbire molta energia.

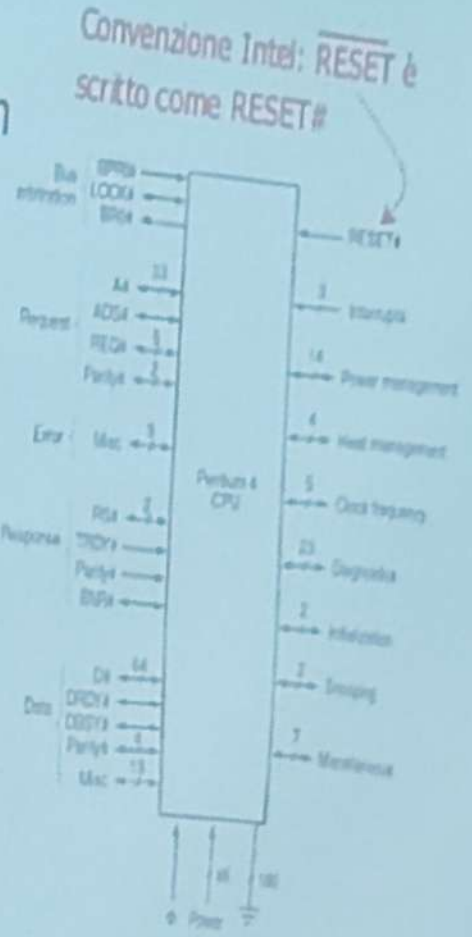
- In un computer portatile utilizzare troppa energia non è desiderabile poiché si consuma presto la carica della batteria.
- Intel allora ha progettato un modo per addormentare le CPU quando sono in uno stato di inattività (o idle) e in un sonno profondo, quando è probabile che si rimanga in questo stato per più tempo.

Gli stati di funzionamento del Pentium 4

- Sono stati definiti 5 livelli di funzionamento dallo stato attivo al sonno profondo.
- Negli stati intermedi alcune funzionalità importanti (come lo snooping della cache snooping e la gestione degli interrupt handling) sono abilitate, mentre altre sono disattivate.
- Quando la CPU è in sonno profondo:
  - I valori delle cache e dei registri sono preservati, ma il clock e le unità interne sono spente.
  - Solo un segnale hardware può risvegliarla.

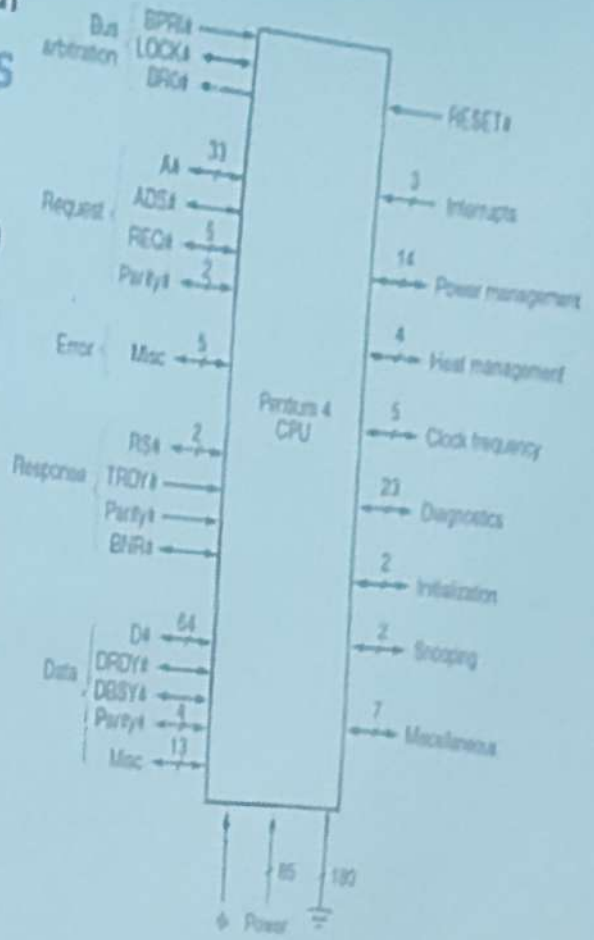
Pin e segnali del Pentium 4

- Pentium 4 ha 478 pin:
  - 198 sono utilizzati per i segnali.
  - 85 sono di alimentazione (con differenti voltaggi).
  - 180 sono per la massa.
  - 15 sono liberi per usi futuri.
- Segnali di **arbitraggio del bus**:
  - ***BR0#*** è per la richiesta del bus
  - ***BPRI#*** è per le richieste del bus ad alta priorità
  - ***LOCK#*** è utilizzato per indicare che il bus è occupato.



Pin e segnali del Pentium 4

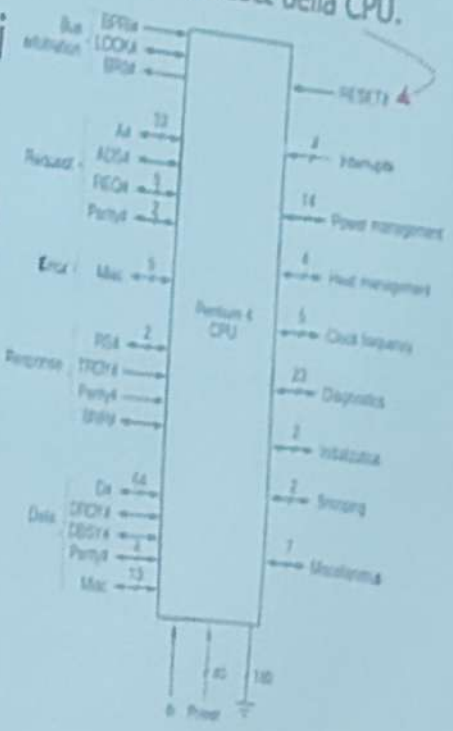
- Segnali dell'**Address bus** e di **controllo** (utilizzati dal bus master):
  - **A#** 36 bit di indirizzamento di cui 3 bit meno significativi impostati a 0.
  - **ADS#** è utilizzato per indicare che il bus indirizzi contiene un riferimento valido.
  - **REQ#** specifica il tipo di operazione (lettura/scrittura).



Pin e segnali del Pentium 4

- Ci sono vari segnali di errore per rappresentare errori derivanti dal calcolo (floating-point), dei dispositivi interni, dei controlli macchina e altri di tipo generico.
- Segnali di **risposta** relativi al **bus** (utilizzati dal **bus slave** per comunicare con il **bus master**):
  - **RS#** contiene il codice di stato.
  - **TRDY#** indica che lo slave è pronto per accettare dati.
  - **BNR#** permette allo slave di inserire stati di attesa (**wait state**).

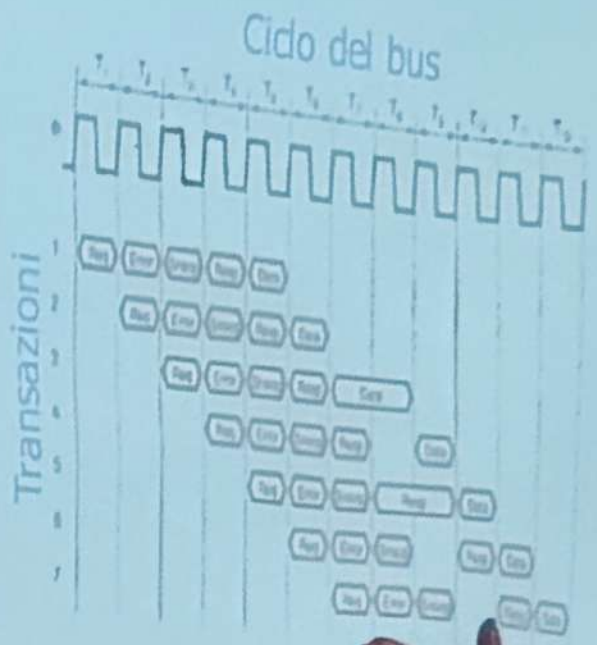
RESET# è utilizzato per il reset della CPU.



Pipeline nel Pentium 4

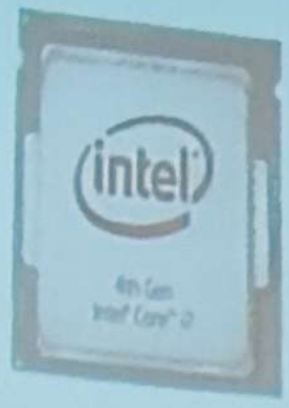
- Le CPU attuali sono più veloci delle memorie centrali (basate su DRAM) quindi è essenziale ottenere il massimo throughput dalla memoria per non lasciare in attesa la CPU.

- Il bus che collega la CPU alla memoria è altamente parallelizzato, con 8 transazioni concorrenti.



Intel Core i7

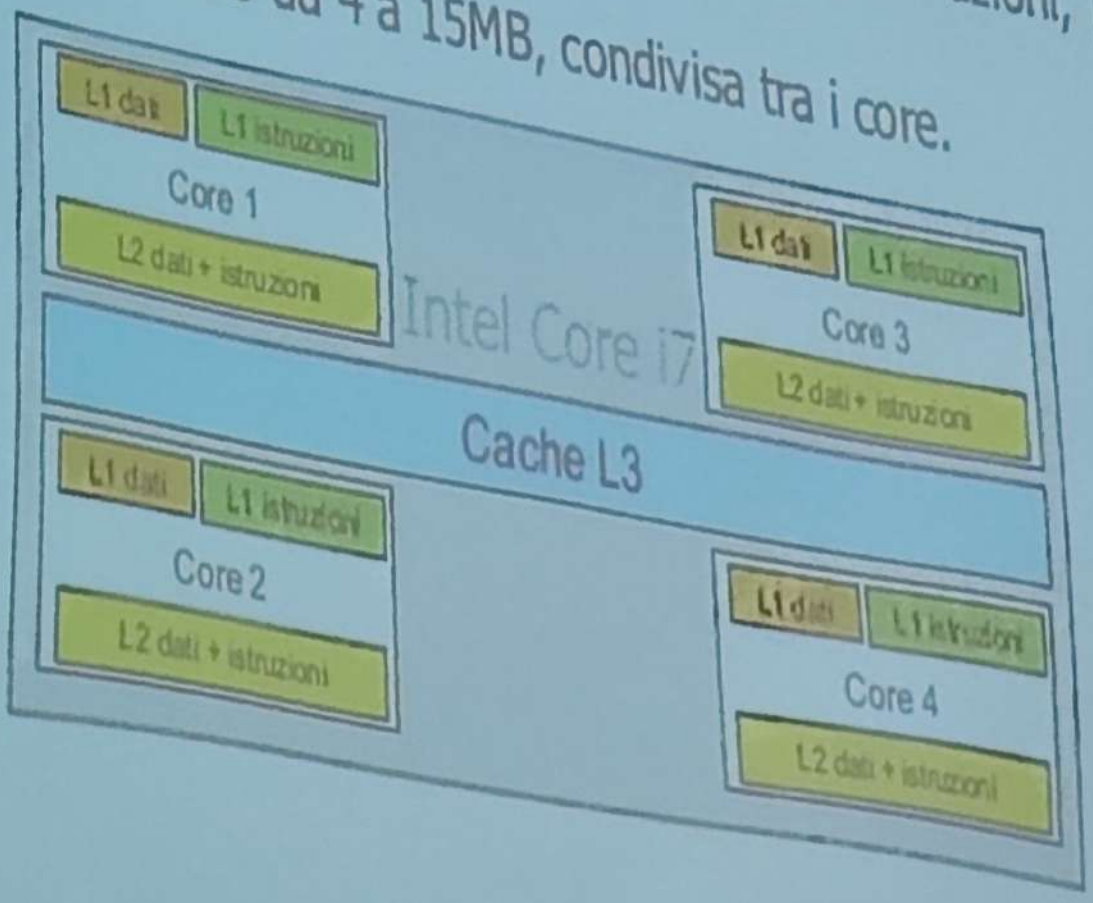
- È una macchina a 64bit, anch'esso è un diretto discendente della CPU 8088.
- La prima versione Core i7 (2008) conteneva 4 core con **731** milioni di transistor, larghezza di linea **45**nm, frequenza **3,2** GHz.



- Ogni core è "**hyperthreaded**" (multithread simultaneo), più thread hardware attivi in parallelo sullo stesso core.
- Dispone di 3 livelli di cache, ciascun core effettua uno **snooping** sul bus di collegamento con la memoria per garantire la consistenza delle informazioni.

Intel Core i7

- Dispone di 3 livelli di cache:
  - Due distinte cache L1 da 32KB per dati e istruzioni, per core.
  - Una cache L2 da 256KB integrata di dati e istruzioni, per core.
  - Una cache L3 da 4 a 15MB, condivisa tra i core.



UltraSPARC III

- La famiglia UltraSPARC era la linea di CPU RISC a 64-bit prodotta dalla Sun Microsystem (dal 2010 acquistata dalla Oracle).
- La UltraSPARC III era pienamente compatibile con la precedente architettura a 32-bit (SPARC V8) e veniva utilizzata per le workstation e i server prodotti dalla Sun Microsystem.
- Il chip era prodotto dalla Texas Instruments.
- Nel 2002, la larghezza di linea era di 130 nm e il clock di 1,2 GHz.
- Questi chips consumavano 50W di potenza e avevano gli stessi problemi di dissipazione termica del P4.

UltraSPARC III

- Non essendo possibile confrontare processori con architettura differente (CISC vs RISC), si preferisce delineare l'insieme delle caratteristiche della UltraSPARC III.
- La CPU poteva eseguire 4 istruzioni per cido di clock ed aveva:
  - 6 pipeline interne:
    - 2 a 14-stadi per operazioni su interi,
    - 2 per operazioni in virgola mobile.
    - 1 per le operazioni in memoria (load/store)
    - 1 per i salti e i branch del programma.

(segue

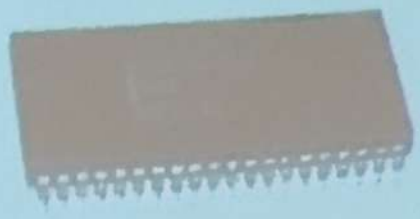
CPU più economiche

- Sia il Pentium 4 sia UltraSPARC III sono stati esempi di CPU ad alte prestazioni per la costruzione di PC e server estremamente veloci.

- Esiste un'alta classe di CPU destinate ai sistemi embedded che possiamo trovare dentro gli elettrodomestici, i cellulari, i giochi elettronici, le protesi,...

Il microcontrollore 8051

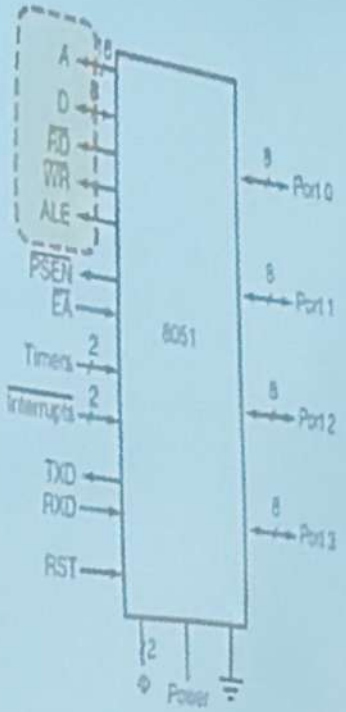
- Il chip Intel 8051 è stato uno tra i più diffusi microcontrollori nelle applicazioni di controllo industriali in virtù del suo basso costo.
- È un circuito integrato da 40 pin con 16-bit di address (può indirizzare fino a 64KB di memoria) e 8-bit per il bus dati.
- A differenza di una CPU pura (come il P4 e l'UltraSPARC III) ha 32 line di I/O, organizzate in 4 gruppi di 8 bit ciascuno.



Pinout fisico dell'8051

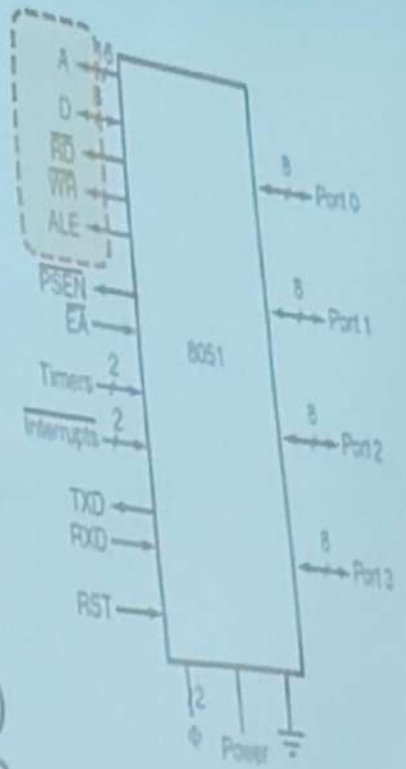
Il microcontrollore 8051

- Ha 4 set di registri ciascuno da 8 registri (R0÷R7); i registri e l'ALU sono a 8-bit.
- **A** è il bus indirizzi a 16-bit per la memoria esterna, **D** contiene un bus dati a 8-bit.
- I segnali **$\overline{RD}$** e **$\overline{WR}$** servono a leggere o scrivere dati da/verso la memoria esterna.
- **ALE** (Address Latch Enable) indica la presenza di un indirizzo valido sul bus.



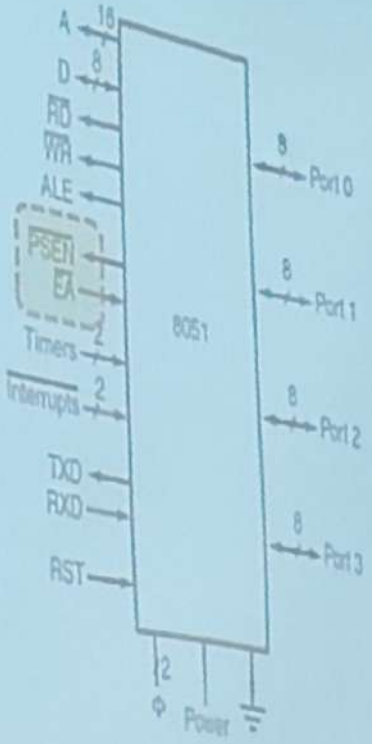
Il microcontrollore 8051

- Ha 4 set di registri ciascuno da 8 registri (R0÷R7); i registri e l'ALU sono a 8-bit.
- **A** è il bus indirizzi a 16-bit per la memoria esterna, **D** contiene un bus dati a 8-bit.
- I segnali **$\overline{RD}$** e **$\overline{WR}$** servono a leggere o scrivere dati da/verso la memoria esterna.
- **ALE** (Address Latch Enable) indica la presenza di un indirizzo valido sul bus.



Il microcontrollore 8051

- **$\overline{\text{PSEN}}$** (Program Store Enable) indica che la CPU vuole leggere il programma dalla memoria.
- **$\overline{\text{EA}}$** (External Access) può essere collegato:
  - **alto**, per usare sia la memoria interna (4 KB) sia quella esterna (sopra i 4 KB);
  - **low**, per utilizzare soltanto la memoria esterna ed escludere quella interna.

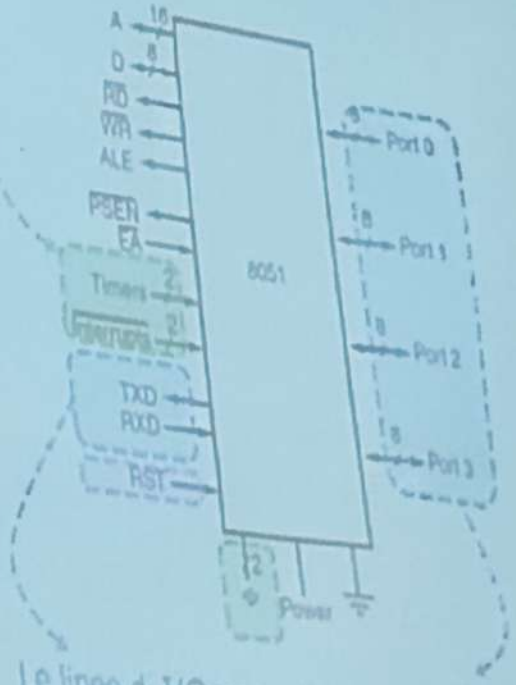


Il microcontrollore 8051

- Ha due clock esterni, due contatori a 16-bit, due livelli di priorità di interruzione, asseriti al livello basso.

- Le linee di I/O sono:
  - **TXD**, per l'uscita seriale.
  - **RXD**, per l'ingresso seriale.
  - 4 porte bi-direzionali ciascuna con 8-bit paralleli (complessivamente 32 linee di I/O).

- Il reset del chip avviene con il segnale **RST**.



Le linee di I/O permettono un utilizzo specifico del chip rispetto ad una tradizionale CPU che invece ha dei controllori sofisticati fuori dal chip.

Esempi di bus

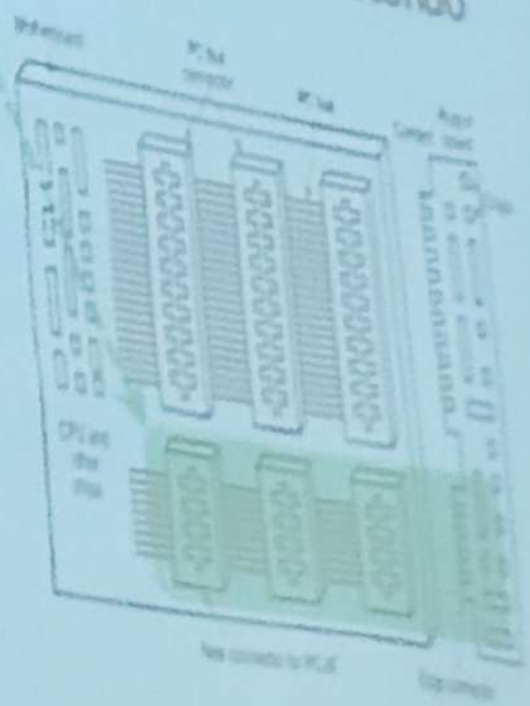
- I bus sono il collante che tiene insieme le componenti del computer.
- Oggi i più diffusi sono il bus **PCI**, **PCI Express** (o **PCIe**) e l'**USB**.
- Mentre l'USB è un bus per periferiche a bassa velocità (tra cui mouse e tastiera), il PCI e PCIe sono utilizzati per connettere le periferiche veloci.

I primi bus

- Il bus del primo PC IBM è stato lo standard de facto dei sistemi basati su architettura 8088 era parallelo e aveva 62 segnali, tra cui:
  - Segnali di controllo (memory read/write, I/O read/write,...).
  - 20 bit per l'address bus;
  - 8 bit per il bus dati.
  - Altri segnali (INT, DMA,...).

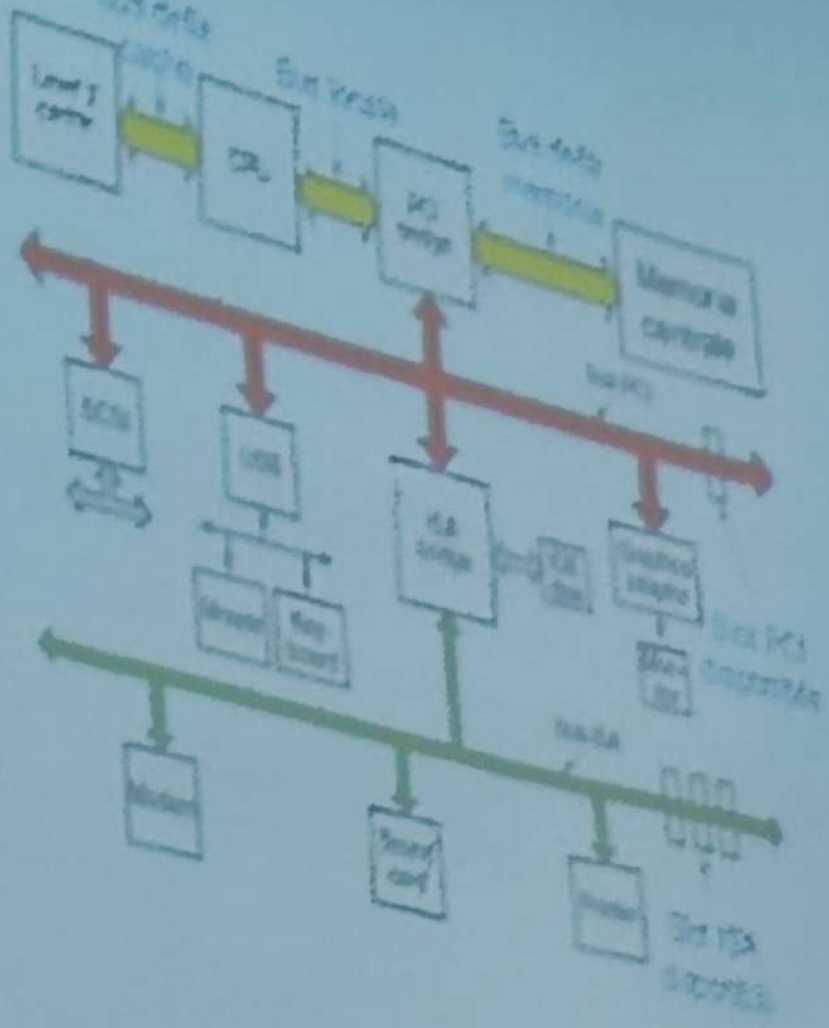
I primi bus

- Per mantenere la compatibilità con le schede esistenti quado naque il PC/AT 80286 fu aggiunto un secondo connettore.
- Il bus ISA (Industry Standard Architecture) era una copia di quello del 80286 e funzionava con un clock di 8,33 MHz.
- Il successore del bus ISA fu esteso a 32-bit e, per questa ragione, fu chiamato l'EISA (Extended ISA).



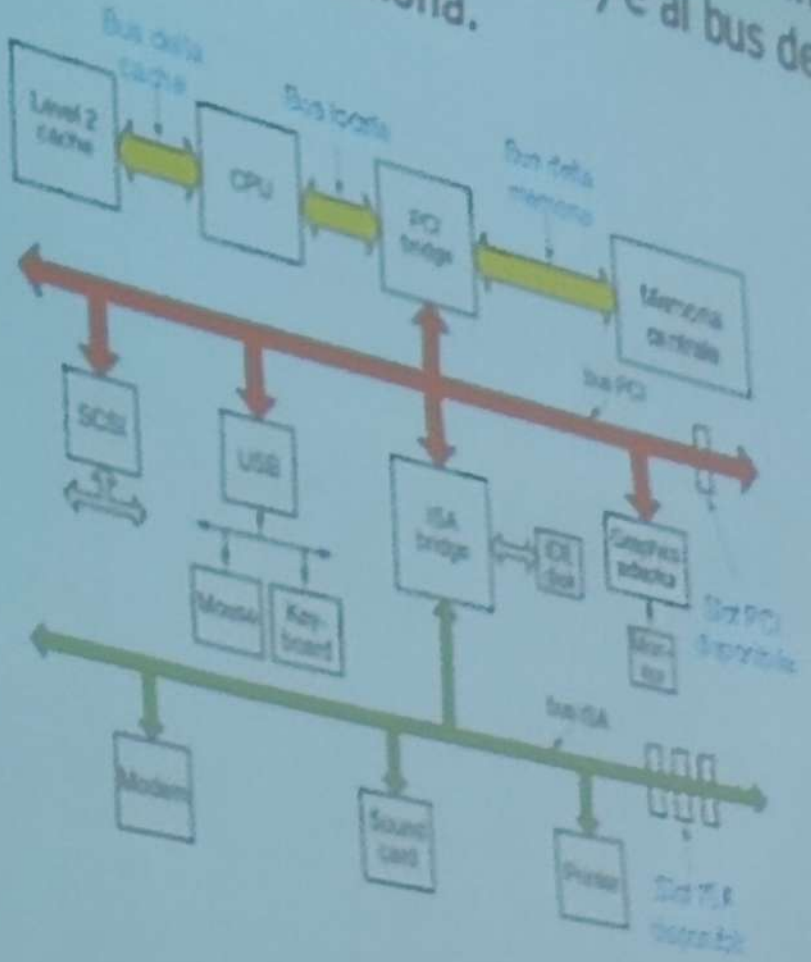
The Bus PCI

- Con l'introduzione dei giochi multimediali e video a pieno schermo, la velocità offerta dal bus ISA (16,7 MB/s) divenne presto insufficiente.
- Il bus PCI (Peripheral Component Interconnect) può funzionare conuna frequenza di clock fino a 66 MHz, gestire trasferimenti a 64-bits, con una banda totale di 528 MB/s.



The Bus PCI

- A partire dal Pentium, il bus PCI è utilizzato insieme al bus ISA (per ragioni di compatibilità) e al bus dedicato al collegamento con la memoria.

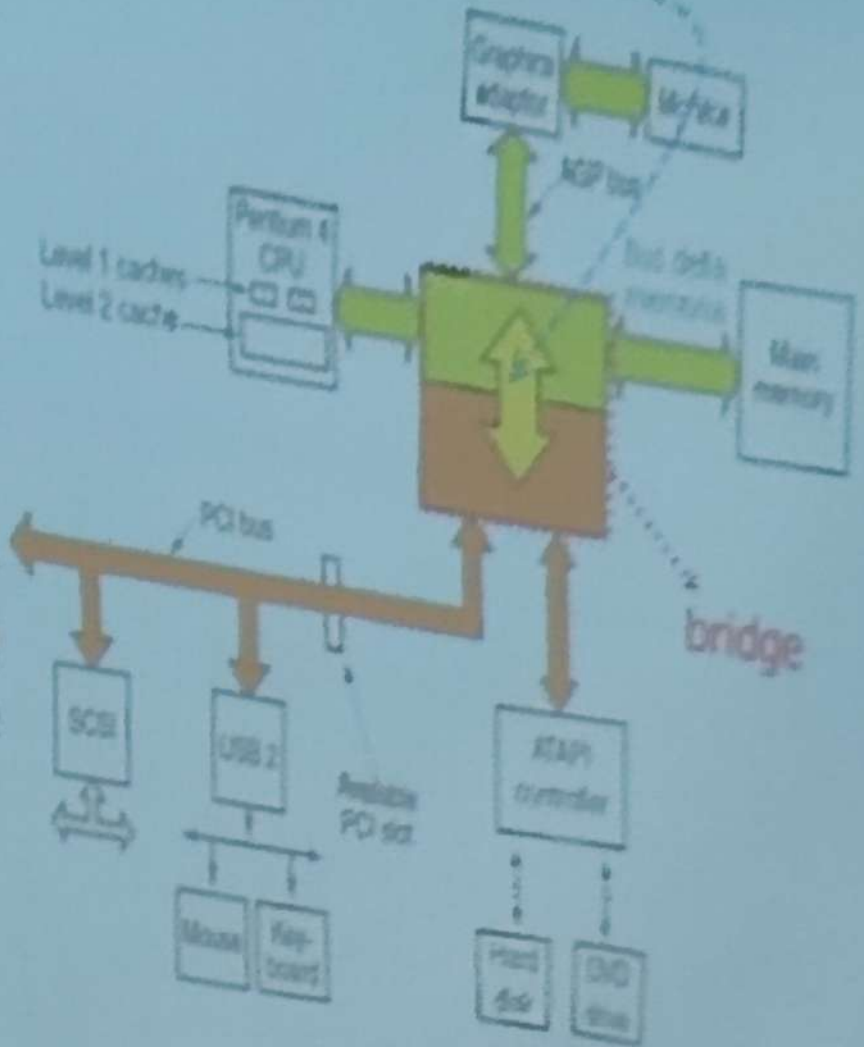


Il bus AGP

- Alla fine degli anni 90 fu introdotto un bus dedicato per le schede grafiche l'AGP (Accelerated Graphics Port) che funzionava a 2,1 GB/sec.

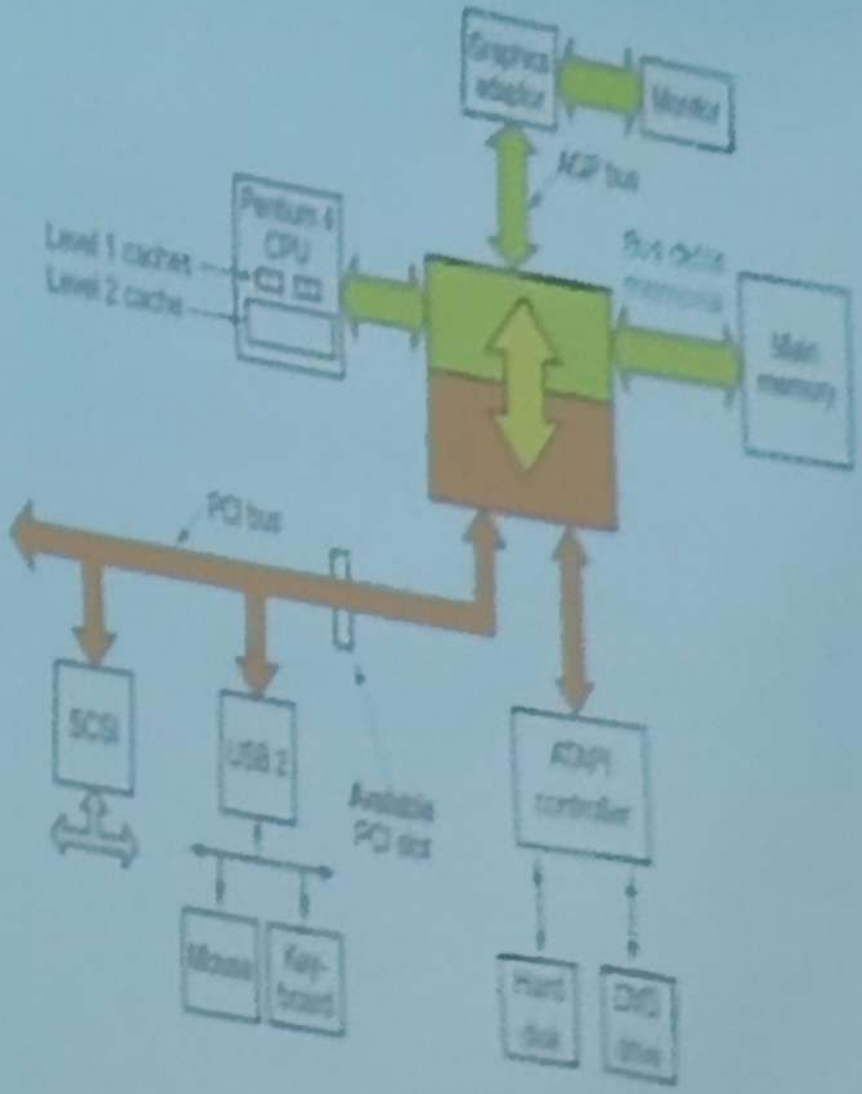
Il bus AGP

- Nel p4 esiste bridge che collega tutti i dispositivi ed è suddiviso in due sub-bridge collegati con una interconnessione veloce:
  - Una connette CPU, memoria e controller video.
  - L'altra il controller ATAPI (HD e DVD) e il bus PCI (SCSI e USB2).



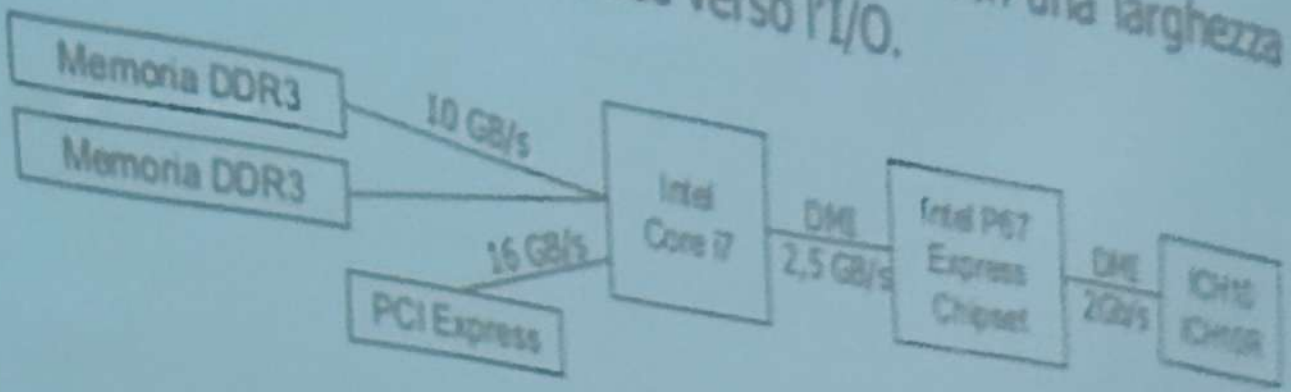
Il bus AGP

- Il bus PCI bus è sincrono: le transazioni iniziano con un master che invia dati ad uno slave.
- Il bus di 64-bit è multiplexato tra dati e indirizzi (ciascuno è composto di 64-bit).



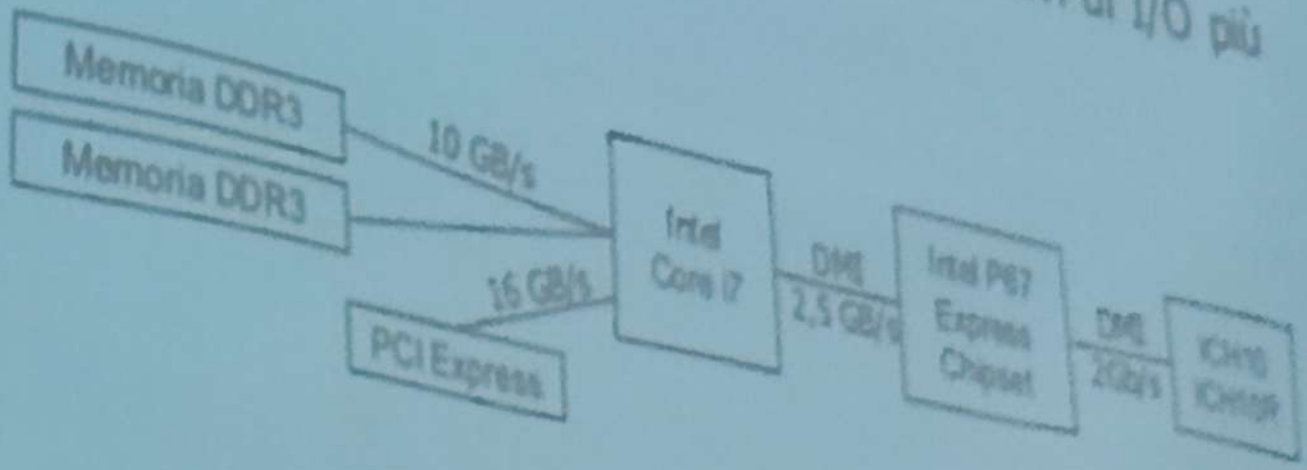
Il bus PCI express

- Il bus PCI express può arrivare fino a 16 GB/sec su collegamenti seriali ad alta velocità.
- In un sistema basato su Core i7 molte interface sono integrate sul chip della CPU:
  - Due canali di memoria a 1,333 GHz hanno una larghezza di banda di 10 GB/sec.
  - Un canale PCI Express a 16 corsie con una larghezza di banda di 16 GB/sec verso l'I/O.



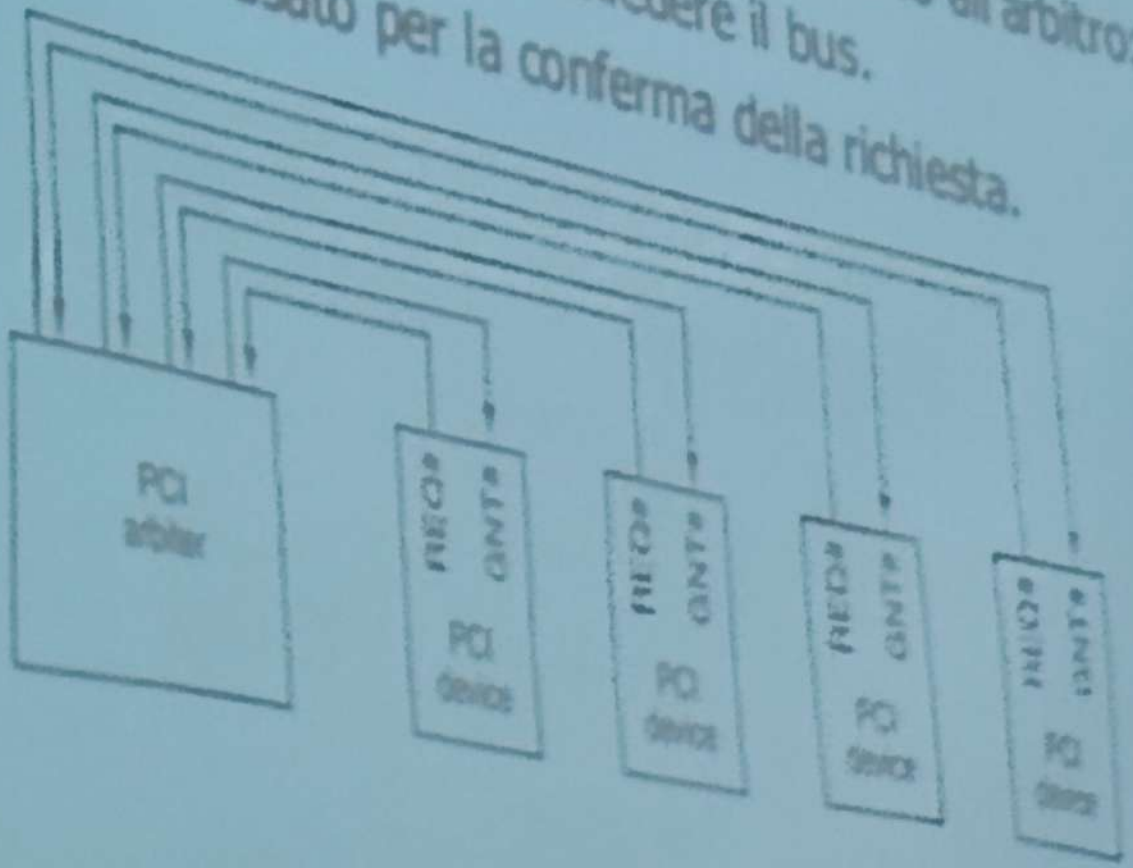
Il bus PCI express

- Il bridge in questo caso è l'Intel P67 collegato alla CPU tramite una interfaccia seriale DMI ed in grado di collegare dispositivi di I/O veloci.
- Il chip ICH10 permette di collegare dispositivi di I/O più lenti o più vecchi.



Arbitraggio del bus PCI

- Il bus PCI utilizza l'arbitraggio centralizzato e l'arbitro è di solito inserito in uno dei chip di bridge.
- Ogni dispositivo ha due line che lo connettono all'arbitro:
  - REQ# è utilizzato per richiedere il bus.
  - GNT# è usato per la conferma della richiesta.

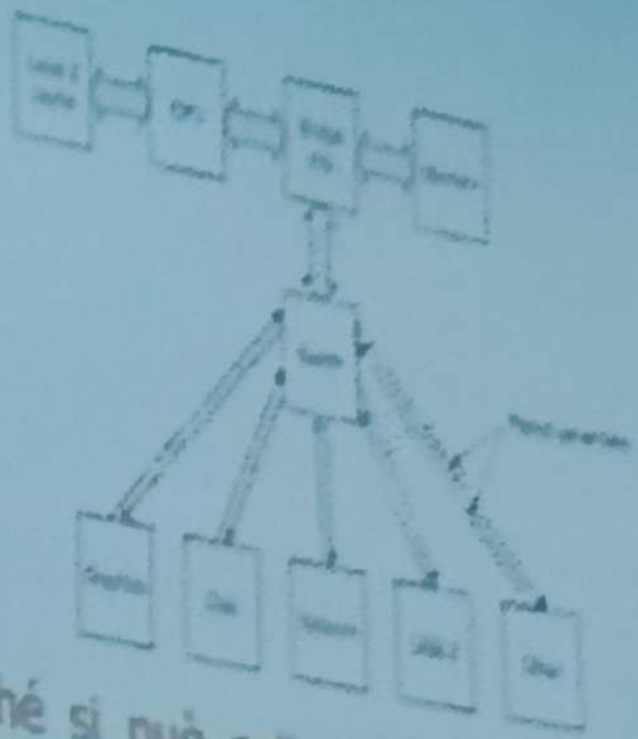


Il bus PCI Express

- Il bus PCI Express (PCIe) cambia il concetto di bus parallelo (utilizzato da ISA/EISA/PCI bus) proponendo un'architettura basata su connessioni seriali punto-punto.
- La CPU, memoria e la cache sono connesse al chip di bridge nel modo tradizionale.
- Il cuore dell'architettura è uno switch: una connessione dedicata punto-punto è realizzata per ogni chip di I/O.

L'architettura del bus PCI Express

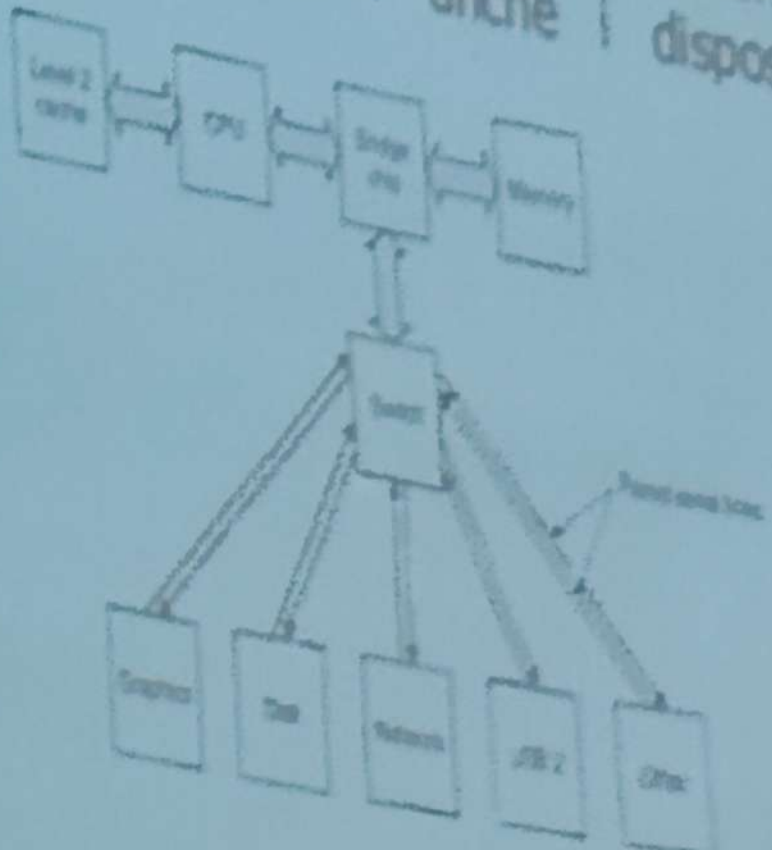
- In effetti in un PC con bus PCI Express abbiamo una mini rete a commutazione di pacchetto.
- La connessione tra switch e dispositivi non può eccedere i 50cm.



- Il sistema è espandibile poiché si può collegare un altro switch al posto di un dispositivo creando così un albero di switch (fino ad un massimo di tre).

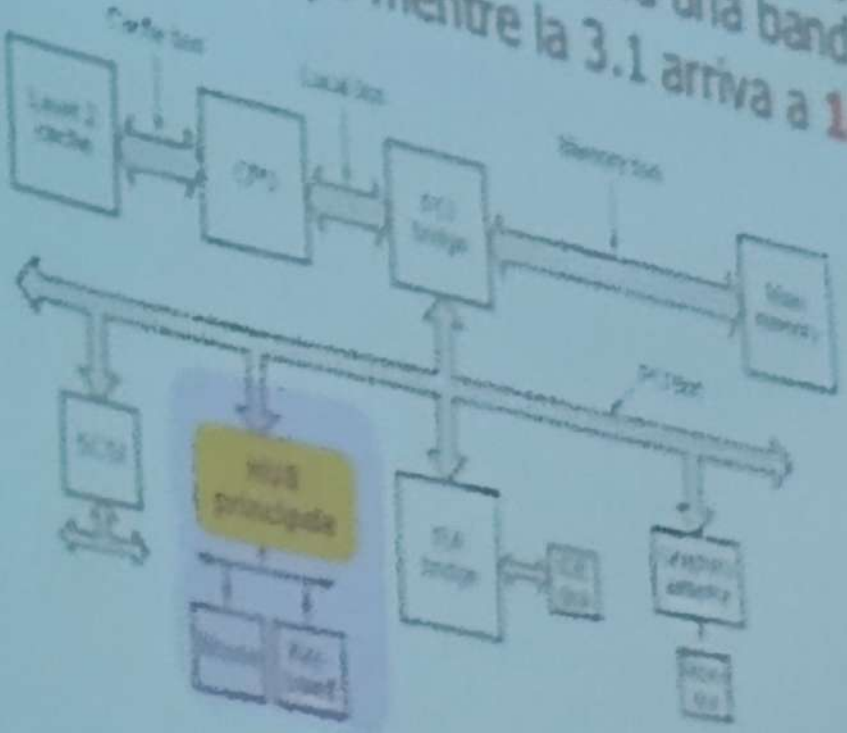
L'architettura del bus PCI Express

- I dispositivi sono essere inseriti o rimossi a «caldo» cioè quando il sistema è in funzione.
- I connettori seriali sono più piccoli dei connettori paralleli, quindi risultano più piccolo anche i dispositivi e i computer.



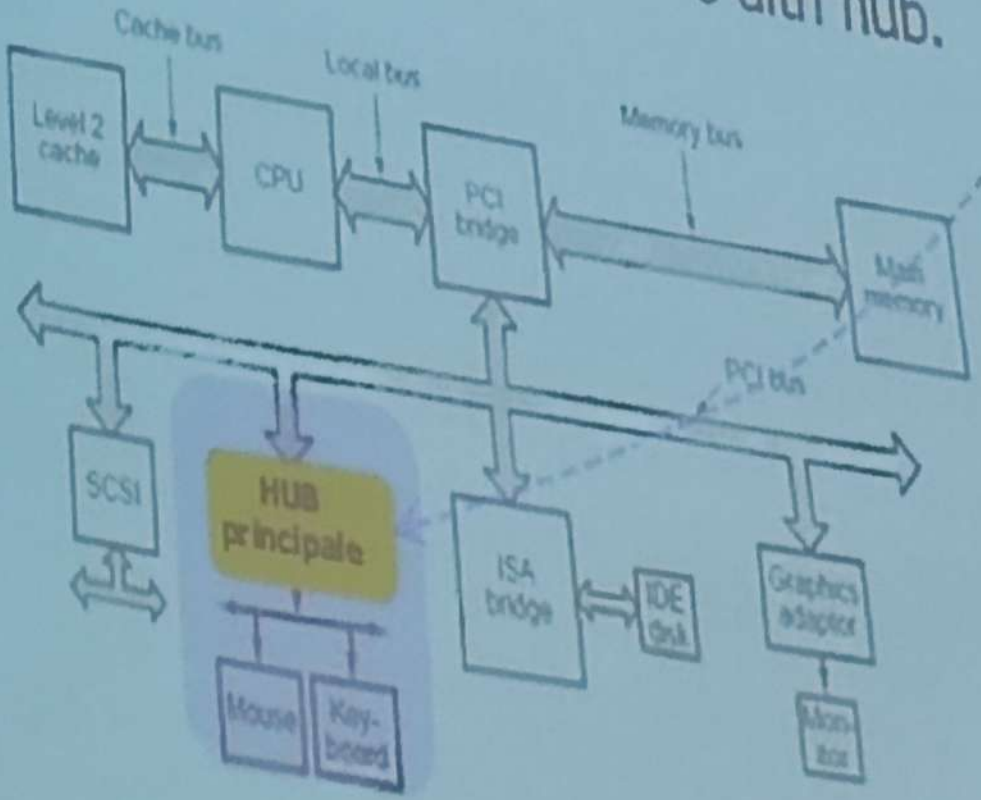
Universal Serial Bus (USB)

- I bus PCI e PCI Express sono ottimi per connettere periferiche ad alta velocità, ma sono troppo costosi per quelle a bassa velocità.
- USB è stato standardizzato nel 1998 per il collegamento con dispositivi lenti, la versione 1.0 ha una banda di 1,5 Mbps, la 2.0 480 Mbps mentre la 3.1 arriva a 10 Gbps.



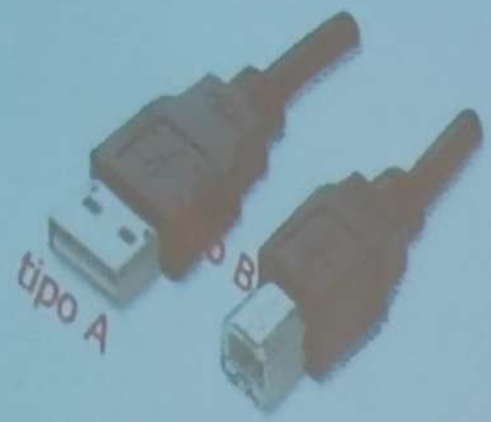
Universal Serial Bus (USB)

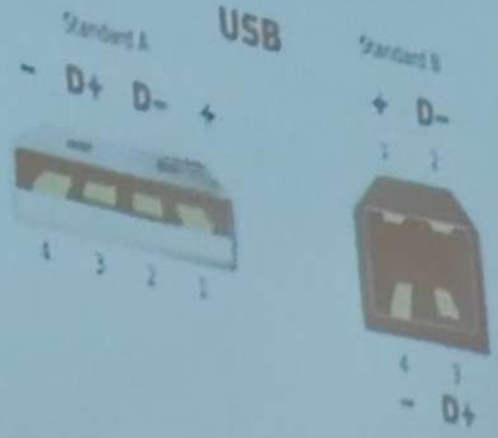
- Un sistema USB si compone di un **hub principale** (o **root hub**) connesso al bus di sistema dove, a sua volta, si possono collegare le periferiche o altri hub.



Universal Serial Bus

- Il cavo di collegamento di USB 1.1÷2.0 si compone di quattro fili:
  - due per i dati (D+ e D-).
  - uno di alimentazione (+5 V).
  - uno per la massa.





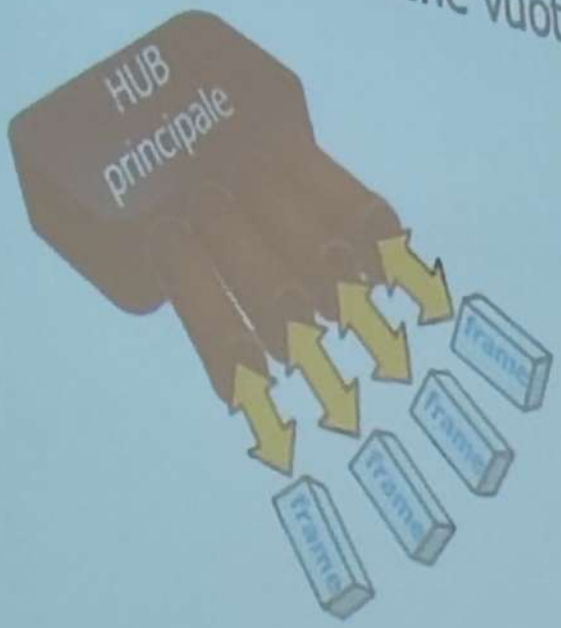
- USB 3.0 ha invece 10 fili.

Universal Serial Bus

- Quando un nuovo dispositivo è collegato, l'**hub root** rileva questo evento e genera un interruzione per il sistema operativo che:
  - interroga il dispositivo per sapere di che tipo di periferica si tratta e di che banda ha bisogno.
  - se la larghezza di banda è sufficiente, il sistema operativo gli assegna un numero unico (**1÷127**) e carica le informazioni del dispositivo.
  - ora è pronto per funzionare e non sono necessarie altre operazioni (è stato aggiunto "al volo").

Universal Serial Bus

- L'**hub** effettua un collegamento punto-punto con i dispositivi di I/O come se ci fossero dei tubi (l'hub non consente collegamenti dispositivo-dispositivo).
- Per mantenere sincronismo, l'**hub** ogni ms spedisce broadcast un nuovo **frame** (anche vuoto).



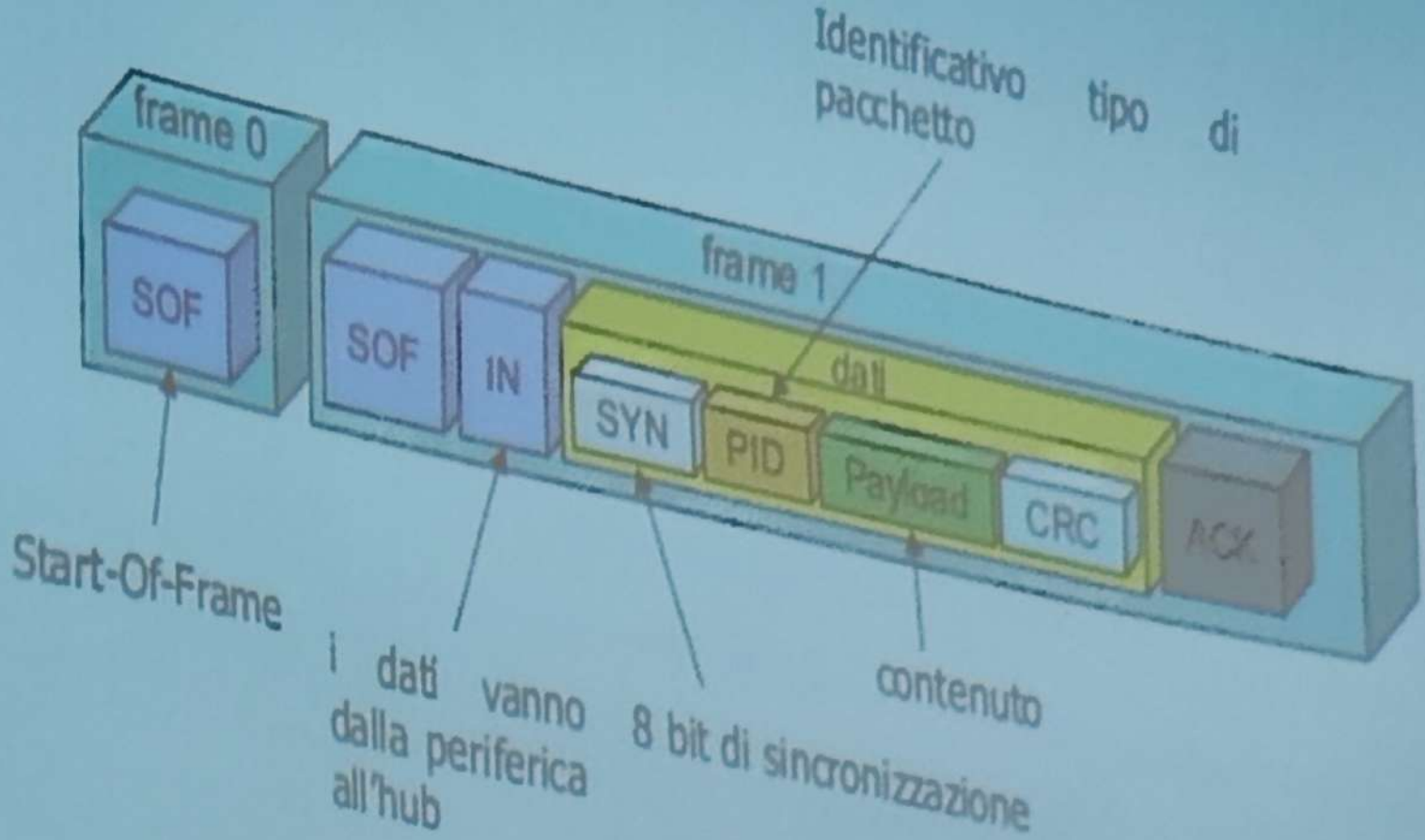
Tipi di frame

- L'USB supporta quattro tipi di **frame**:
  - **controllo** utilizzati per configurare il dispositivo, inviargli comandi, interrogarlo sullo stato.
  - **isocroni** utilizzati per dispositivi real-time come microfoni o telefoni che necessitano di spedire o accettare dati ad intervalli di tempo precisi (in caso di errore non forniscono ritrasmissione).
  - **bulk** utilizzati per grandi trasferimenti di dati come nel caso delle stampanti che non richiedono un funzionamento in tempo-reale.
  - **interrupt** sono fondamentali in quanto USB non supporta il concetto di interruzione quindi il sistema operativo senza di essi sarebbe costretto ad interrogare in polling il dispositivo.

Tipo di pacchetti

- Un frame contiene uno o più **pacchetti**.
- Esistono Quattro tipi di pacchetto:
  - token utilizzati il controllo del sistema dall'hub al device:
    - SOF, Start-Of-Frame.
    - IN, i dati vanno dalla periferica all'hub.
    - OUT, i dati vanno dall'hun alla periferica
    - SETUP, i dati di configurazione saranno inviati alla periferica.
    - ....
  - dati 8 bit di sincronizzazione, identificatore del tipo pacchetto (PID), **payload** e CRC (16 bit).
  - handshake ACK (l'hub ha ricevuto bene), NAK (cè un errore di CRC) e STALL (attendere).
  - speciali utilizzati per usi specifici.

Un Esempio


frame 0
SOF
frame 1
SOF
IN
dati
SYN
PID
Payload
CRC
ACK
Identificativo tipo di pacchetto
Start-Of-Frame
i dati vanno dalla periferica all'hub
8 bit di sincronizzazione
contenuto

Interfacce di I/O

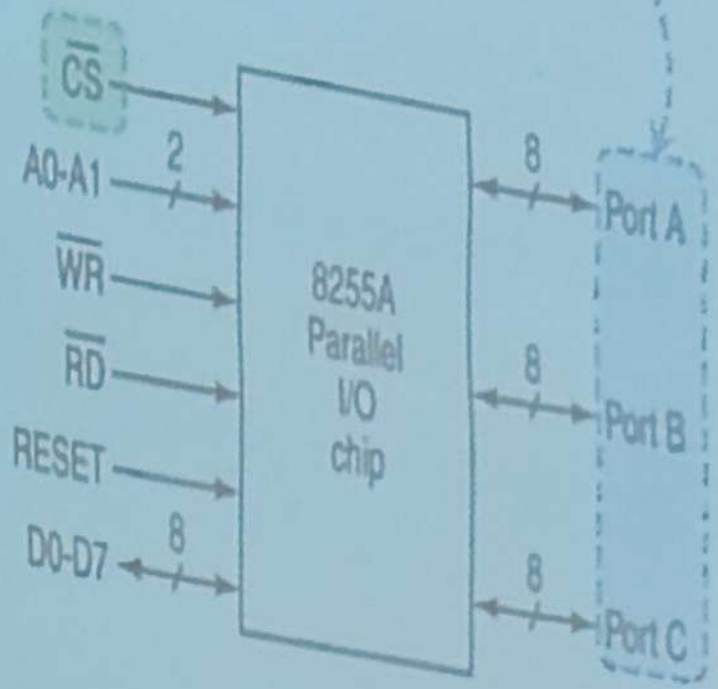
- Le interfacce di I/O sono le schede che permettono ai dispositivi di I/O di collegarsi sul bus e di scambiare dati all'interno del computer.
- Esistono dei chip standard per la realizzazione di questi controllori:
  - **UART** (Universal Asynchronous Receiver Trasmitter), legge un byte dal bus e trasmette un bit alla volta su una linea seriale (utilizzata in passato per il collegamento con i terminali) oppure compie il lavoro opposto.
  - **USART** (Universal Synchronous Asynchronous Receiver Trasmitter), aggiungono alle UART la possibilità di effettuare trsmissioni sincrone.
  - **PIO** (Parallel I/O), chip per il collegamento di un dispositivo di I/O con comunicazione parallela.

Interfacce di I/O

- Le interfacce di I/O sono le schede che permettono ai dispositivi di I/O di collegarsi sul bus e di scambiare dati all'interno del computer.
- Esistono dei chip standard per la realizzazione di questi controllori:
  - **UART** (Universal Asynchronous Receiver Trasmitter), legge un byte dal bus e trasmette un bit alla volta su una linea seriale (utilizzata in passato per il collegamento con i terminali) oppure compie il lavoro opposto.
  - **USART** (Universal Synchronous Asynchronous Receiver Trasmitter), aggiungono alle UART la possibilità di effettuare trsmissioni sincrone.
  - **PIO** (Parallel I/O), chip per il collegamento di un dispositivo di I/O con comunicazione parallela.

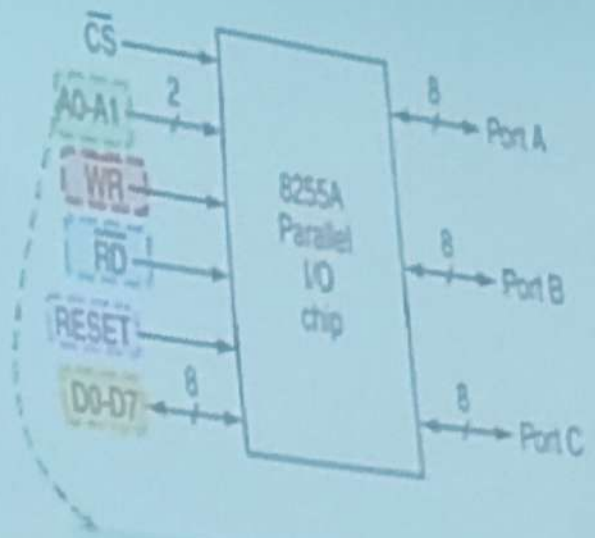
Una PIO di esempio: Intel 8255A

- Ha 3 porte di I/O (**A**, **B** e **C**) ciascuna da 8-bit con associato un un registro latch più un registro interno.
- L'ingresso per l'abilitazione del chip ($\overline{CS}$) è usato per collegare più PIO in parallelo.



Una PIO di esempio: Intel 8255A

- Due linee di indirizzamento della porta o del registro interno (A_0, A_1).
- Il segnale $\overline{RD}$ indica che la CPU sta effettuando una lettura dal bus dati.
- Il segnale $\overline{WR}$ indica che la CPU ha emesso i dati sul bus e sono validi per una operazione di scrittura.
- Il segnale di RESET.
- 8 linee 3-state per il collegamento al bus dati ($D_0 \div D_7$).



A_1	A_0	Porta selezionata
0	0	Porta A
0	1	Porta B
1	0	Porta C
1	1	registro di controllo

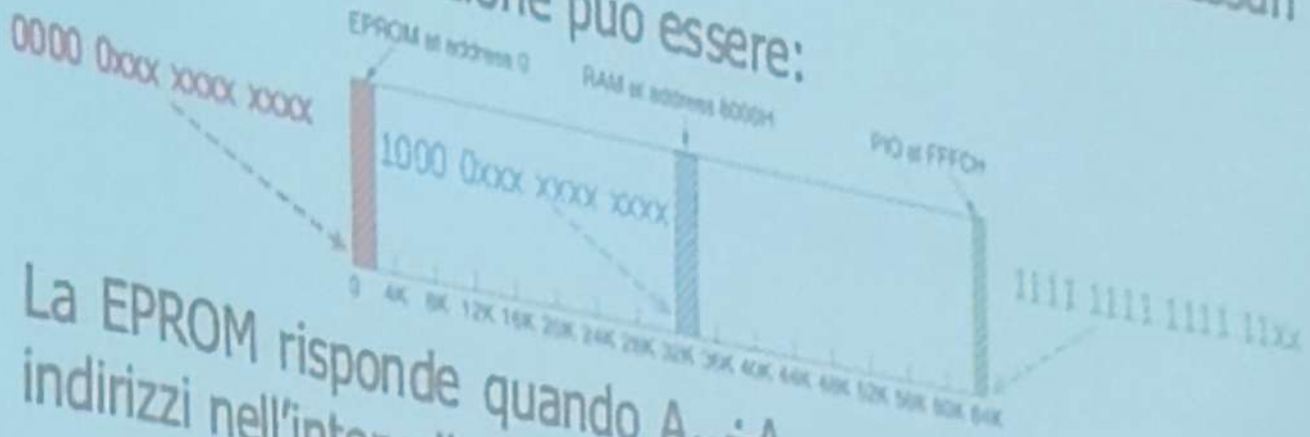
Indirizzamento dell'I/O

- I dispositivi di I/O possono essere indirizzati in due modi:

1. **port-mapped I/O** o **I/O isolato** ovvero come un dispositivo di I/O reale.
   - è necessaria una linea del control bus che distingue se l'operazione deve essere eseguita in memoria oppure su I/O.
   - sono utilizzate delle istruzioni specifiche (es. IN e OUT).
2. **memory-mapped I/O** ovvero come parte della memoria.
   - occorre riservare uno spazio in memoria che sarà destinato all'I/O.
   - le operazioni di lettura e scrittura in memoria eseguite in quello spazio di indirizzamento saranno dirottate sull'I/O.

Decodifica dell'indirizzo

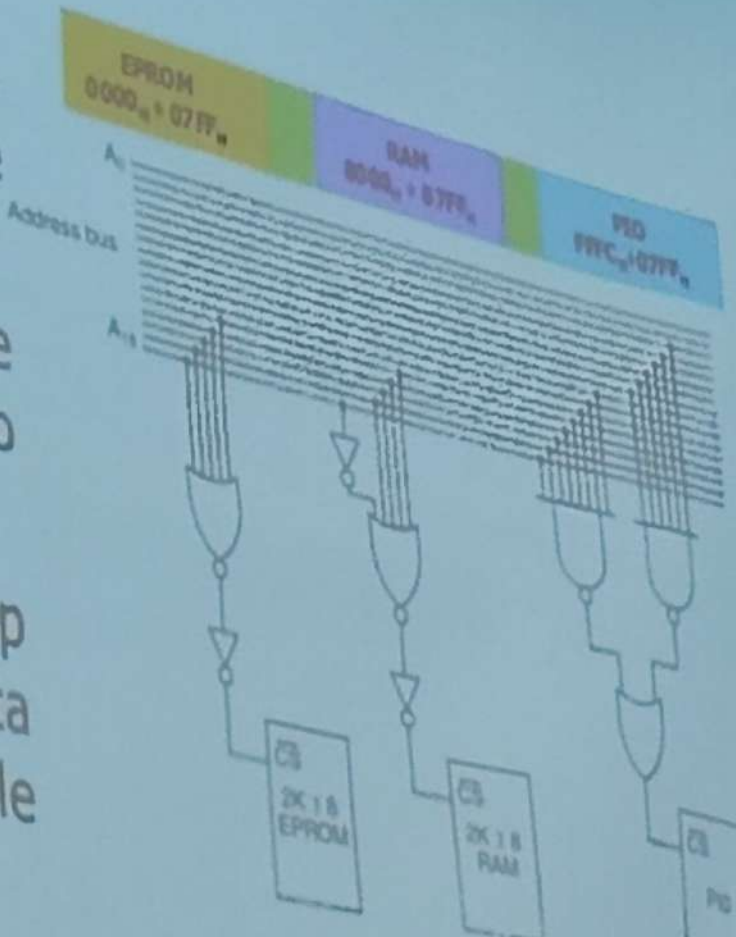
- Lo spazio di indirizzamento è 2^{16} = 64 KB.
- Si può scegliere di allocare il programma e i dati in un qualsiasi segmento da 2KB, mentre alla PIO sono necessari solo 4 byte.
- Una possibile allocazione può essere:



- La EPROM risponde quando $A_{15} \div A_{11}$ sono tutti bassi ovvero indirizzi nell'intervallo $0000_H \div 07FF_H$
- La RAM risponde quando A_{15} è alto e $A_{14} \div A_{11}$ sono bassi cioè indirizzi nell'intervallo $8000_H \div 87FF_H$
- La PIO risponde quando $A_{15} \div A_2$ sono tutti alti quindi indirizzi nell'intervallo $FFFC_H \div FFFF_H$

Memory-Mapped I/O

- La EPROM risponde quando $A_{15} \div A_{11}$ sono tutti bassi.
- La RAM risponde quando A_{15} è alto e $A_{14} \div A_{11}$ sono bassi.
- La PIO risponde quando $A_{15} \div A_2$ sono tutti alti.
- L'abilitazione del chip può essere realizzata con l'utilizzo delle porte logiche.



Conclusioni

- Sono state analizzate alcune CPU fondamentali CISC e RISC.
- Studiate le caratteristiche dei computer embedded in apparecchiature destinate a scopi specifici.
- Analizzato i principali bus del calcolatore ed il problema dell'arbitraggio.
- Studiato le problematiche legate all'indirizzamento dell'I/O.