

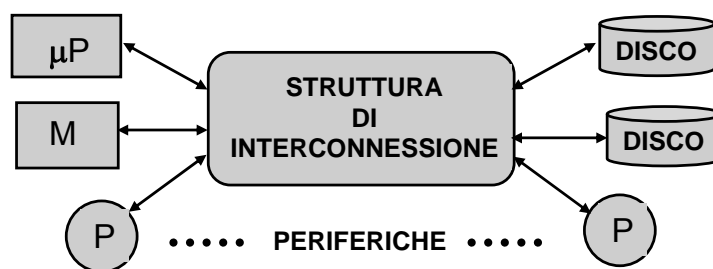
---

## Parte IV

### Bus e microprocessori

---

## Comunicazione nell'elaboratore



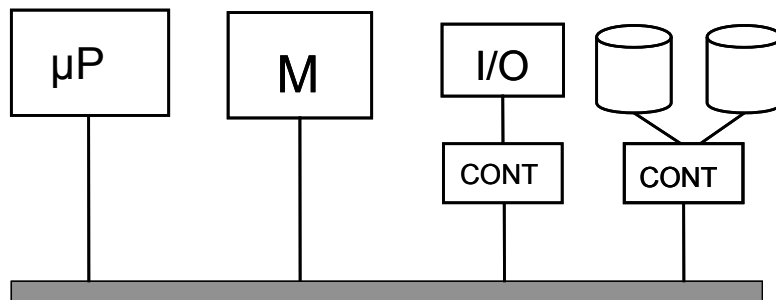
La struttura di interconnessione assicura la comunicazione fra le diverse componenti dell'elaboratore

- CPU-Memoria: *fetch* di istruzioni e *load* e *store* dei dati
- Dischi-Memoria: letture e scritture su memoria di massa
- Periferiche-Memoria: operazioni di I/O

**N.B.** La memoria è coinvolta in tutti i trasferimenti

# Architettura a bus

---



- L'idea di bus nasce con il minicomputer, e viene adottata dal PC
- Linee di controllo e dati condivise da più dispositivi
- Una sola comunicazione per volta
- Condivisione gestita a livello hardware e firmware
- Struttura semplice ed economica
- La banda passante limita la scalabilità
- Le architetture più recenti prevedono più di un bus

# Comunicazione sul bus

---

- Tutti i dispositivi sono sempre *in ascolto*
- Il bus è condiviso a livello di intervalli temporali detti ciclo
- In ciascun ciclo il bus è utilizzato da una *coppia di dispositivi* per trasferire una determinata quantità di informazione
  - Master: dispositivo che svolge un ruolo *attivo*
  - Slave: dispositivo che svolge un ruolo *passivo*
- L'accesso al bus è regolato da un rigido *protocollo*, che stabilisce tempi e modi della comunicazione
- I dispositivi si prenotano e si accodano per l'uso del bus

L'assegnazione del bus è regolata da un *arbitro*:

- Discipline: come scegliere la prossima coppia
- Priorità: Modificano l'ordine di assegnazione

## Tipologie di bus

---

- In base alle modalità di trasferimento:
  - Seriale: viene trasmesso un bit per volta su un'unica coppia di linee
  - Parallelo: vengono trasmessi molti bit (byte) utilizzando una molteplicità di linee dati
- In base alla temporizzazione del ciclo:
  - Sincrono: la temporizzazione è regolata da un clock, la durata di un ciclo è costante
  - Asincrono: la temporizzazione è regolata da segnali che i dispositivi si scambiano

**N.B.** Nei bus asincroni la velocità del bus dipende dalla velocità dei dispositivi

## Prestazioni dei bus

---

- Frequenza: è la frequenza del clock che determina nei bus sincroni la durata del ciclo
- Banda: è la quantità di informazione trasferita nell'unità di tempo, dipende da due fattori:
  - Frequenza: numero di cicli al secondo
  - Parallelismo: numero di bit (byte) trasferiti per ciclo

### ES

*Bus con 64 linee dati che lavora a 100 MHz (ciclo 10 ns):*

- *Ogni ciclo vengono trasferiti 8 byte*
- *100 milioni di cicli al secondo*
- *Banda 800 MB/sec*

## Bus paralleli o seriali?

---

- Il parametro di prestazione che interessa è la banda
- Apparentemente più interessanti i bus paralleli
- Vero, *ma a parità di frequenza*
- Nei bus paralleli la frequenza è limitata da due fattori
  - Lunghezza (e quindi numero di dispositivi collegati)
  - Bus skew: differenza nei tempi di propagazione delle linee parallele

*Il trend evolutivo è oggi un riflusso verso bus seriali, o meglio verso collegamenti seriali. Si sta riaffermando un'architettura basata su connessioni punto-punto.*

*Esempio: Serial ATA e SAS*

*Un ritorno al passato?*

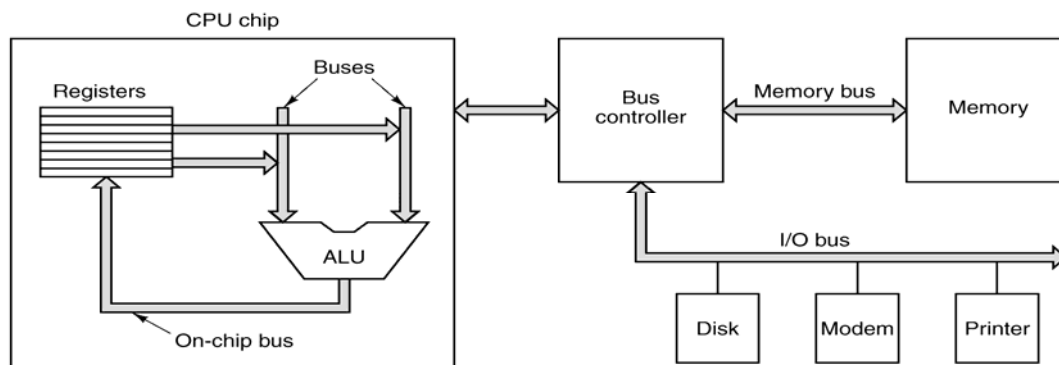
## Standardizzazione dei bus

---

- Elemento fondamentale per l'evoluzione dei sistemi e la crescita del mercato
- Standard emessi da organizzazioni (IEEE, ACM etc.) o da appositi consorzi (USB, SATA, SCSI)
- Lo standard definisce *tutto* quanto concerne bus e dispositivi:
  - Cavi e connettori (con relative tolleranze)
  - Segnali elettrici
  - Protocolli e temporizzazioni (con tolleranze)
- La standardizzazione consente la progettazione e la realizzazione di dispositivi compatibili da parte di produttori senza che essi debbano comunicare tra loro

*Tentativi 'protezionistici' di definire standard proprietari hanno portato a fallimenti clamorosi e miserabili*  
*Es. Microchannel di IBM*

# Architettura a più bus



- Diversi bus, *interni ed esterni* al chip
- Soddisfano diverse esigenze:
  - Velocità di trasferimento
  - Numero di linee
  - Più trasferimenti paralleli
  - Compatibilità all'indietro

## Protocolli di bus

- La comunicazione sui bus è regolata da rigidi protocolli

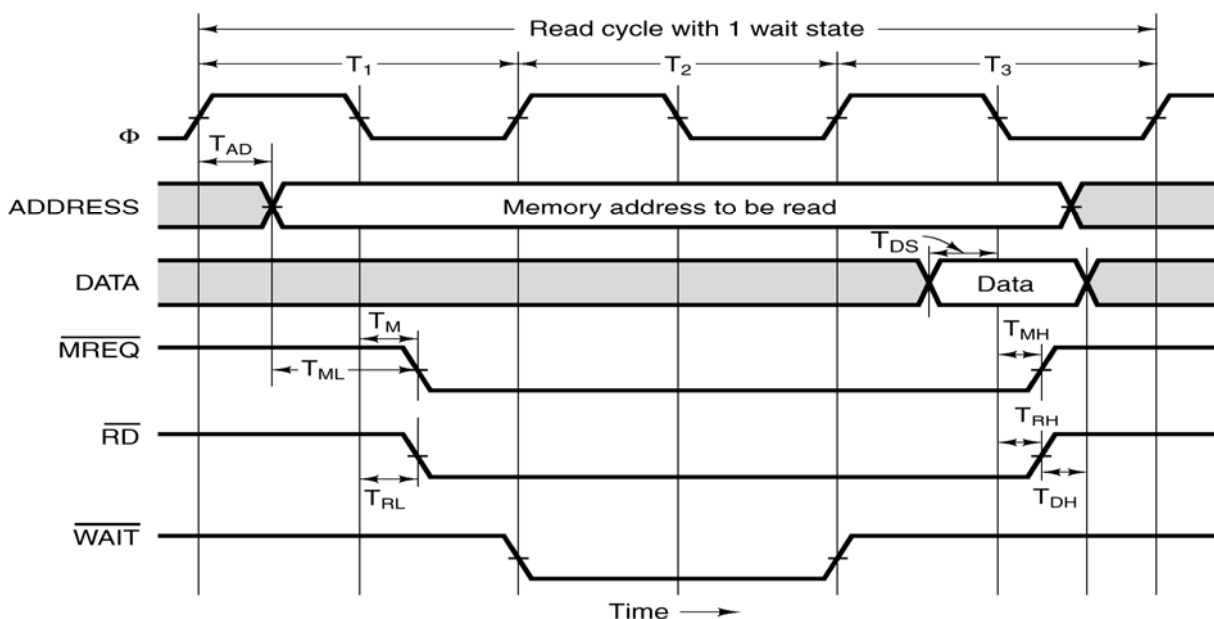
*Un protocollo di bus è una serie di regole alle quali devono attenersi i dispositivi che condividono il bus e lo usano per comunicare tra di loro*

- I protocolli sono definiti dagli standard e specificano i segnali applicati alle linee del bus e la loro temporizzazione
- Esaminiamo la struttura fondamentale del protocollo per le due grandi tipologie di bus:
  - Bus sincroni
  - Bus asincroni

## Segnali asseriti e negati

- In alcuni casi un segnale provoca l'azione corrispondente *quando la sua tensione è alta*, in altri quando è bassa
- Per evitare confusione si parla di:
  - **Segnale asserito**: quando assume il valore che provoca l'azione
  - **Segnale negato**: altrimenti
- Si adotta la seguente notazione:
  - **S**: segnale che è asserito alto
  - **$\bar{S}$** : segnale che è asserito basso
- Ulteriore notazione (usata da Intel):
  - **S**: segnale che è asserito alto
  - **S#**: segnale che è asserito basso

## Bus sincroni: ciclo di lettura



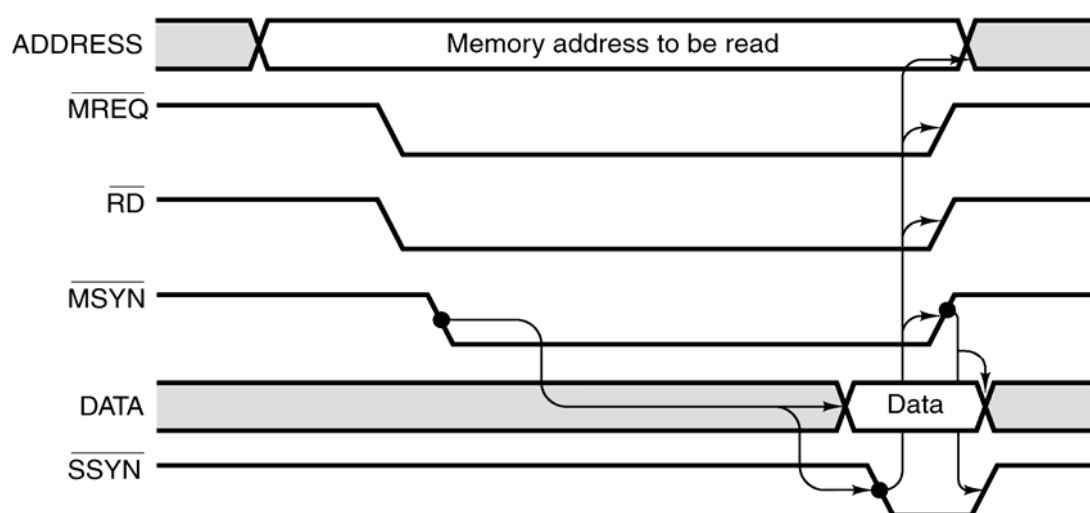
- Tutte le azioni avvengono sui fronti del clock
- Se la memoria mantiene asserito **WAIT** il ciclo si prolunga

## Bus sincroni: temporizzazione

Symbol	Parameter	Min	Max	Unit
$T_{AD}$	Address output delay		11	nsec
$T_{ML}$	Address stable prior to $\overline{MREQ}$	6		nsec
$T_M$	$\overline{MREQ}$ delay from falling edge of $\Phi$ in $T_1$		8	nsec
$T_{RL}$	RD delay from falling edge of $\Phi$ in $T_1$		8	nsec
$T_{DS}$	Data setup time prior to falling edge of $\Phi$	5		nsec
$T_{MH}$	$\overline{MREQ}$ delay from falling edge of $\Phi$ in $T_3$		8	nsec
$T_{RH}$	$\overline{RD}$ delay from falling edge of $\Phi$ in $T_3$		8	nsec
$T_{DH}$	Data hold time from negation of $\overline{RD}$	0		nsec

- Lo standard di un bus sincrono definisce anche le tolleranze sui tempi relativi alle varie fasi del ciclo ( $T = 25 \text{ ns}$ )
- Diversi fabbricanti fanno riferimento indipendentemente allo standard, e producono dispositivi sicuramente compatibili

## Bus asincroni: ciclo di lettura



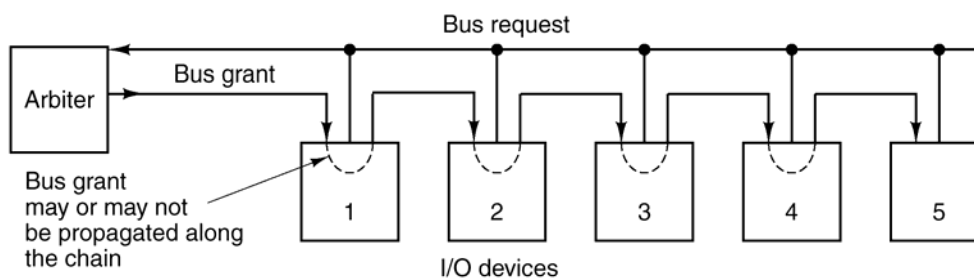
- Accoppiamento di dispositivi con velocità diverse
- Sincronismo tramite segnali di controllo
- Gli eventi avvengono in risposta ad altri eventi (*rapporto di causa ed effetto*)

# Protocollo di Full Handshake

- Protocollo di sincronizzazione su bus asincroni
  - *Full handshake*: stretta di mano completa
  - I dispositivi *master* e *slave* si scambiano i segnali
    - **MSYN** (*Master Synchronization*)
    - **SSYN** (*Slave Synchronization*)
1. **MSYN** asserito: indirizzi e segnali di controllo sono sul bus
  2. **SSYN** asserito in risposta a **MSYN**: i dati sono sul bus
  3. **MSYN** negato in risposta a **SSYN**: i dati sono stati letti
  4. **SSYN** negato in risposta a **MSYN** negato: il bus è libero

*Il protocollo di Full Handshake è più complesso di quello dei bus sincroni, poiché manca il riferimento costituito dai fronti del clock*

# Arbitraggio del bus

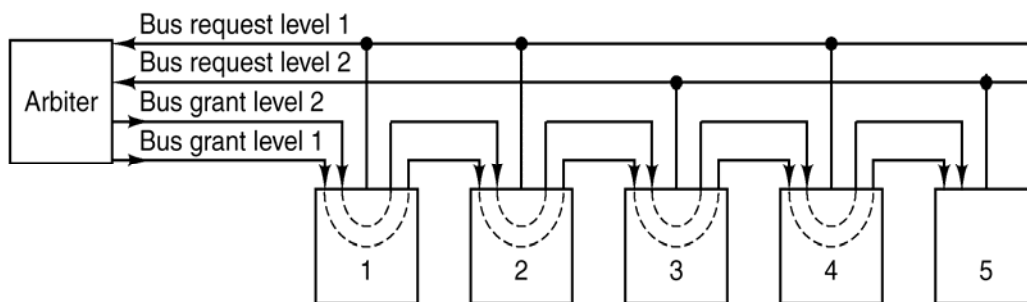


- Determina quale dispositivo sarà il prossimo *Bus Master* risolvendo eventuali conflitti
- Spesso l'arbitro è nel chip del  $\mu P$
- Linea di richiesta condivisa
- *Bus grant* propagato dall'arbitro poco prima dell'inizio del ciclo
- *Vince* il primo che intercetta il *bus grant*

**NB** *Sono favoriti i dispositivi situati vicino all'arbitro*

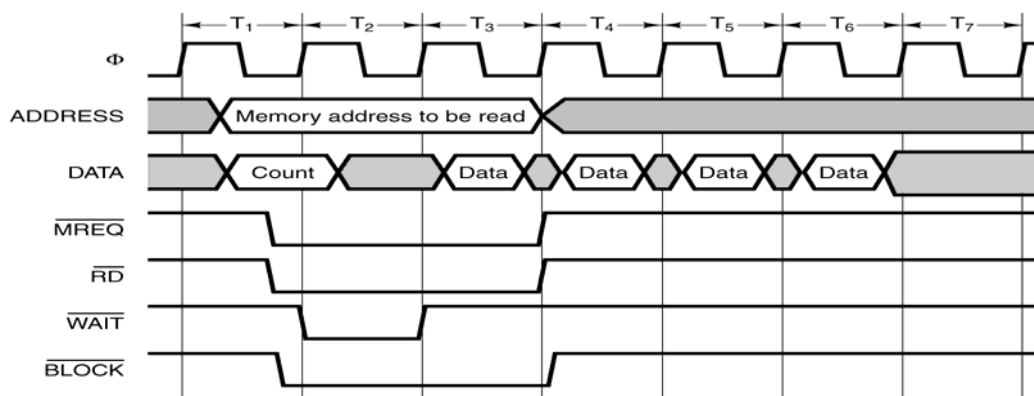


## Livelli multipli di priorità



- Diverse linee di richiesta associate a diversi *livelli di priorità*
- In caso di conflitto favorite le catene a priorità più alta
- All'interno di ciascuna catena vale la posizione
- In genere se c'è un solo bus con anche la memoria la CPU ha priorità più bassa dei dispositivi di I/O (e.g. dischi)

## Block transfers



- Permette di leggere più parole *consecutive*
- Usato per *trasferire blocchi di cache*
- Numero di parole specificato durante  $T_1$
- Dopo la prima viene trasferita una word ogni ciclo (*invece di una ogni tre cicli*)
- Per leggere quattro word occorrono 6 cicli invece di 12
- Il segnale BLOCK viene asserito per chiedere un block transfer

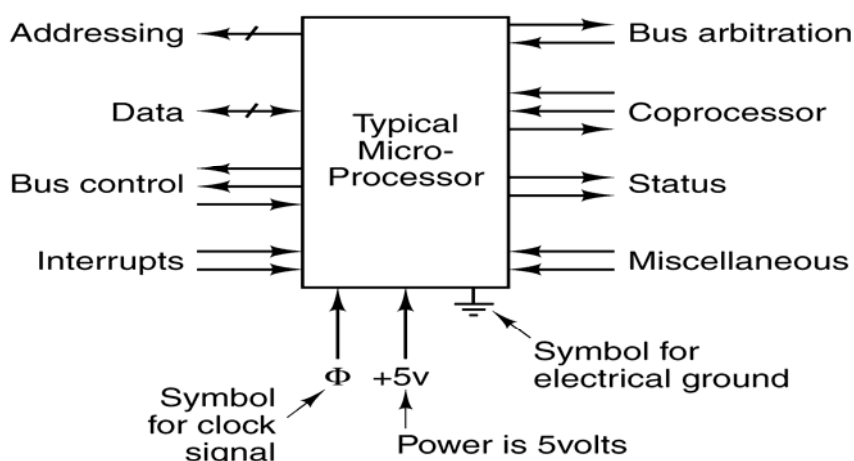
# Il microprocessore

---

- Il microprocessore è costituito da un unico chip in cui sono integrate tutte le funzioni fondamentali di elaborazione
- Contiene al suo interno diversi componenti fondamentali
  - ALU (*Aritmetic Logical Unit*)
  - Unità Floating-point (per i calcoli in virgola mobile)
  - Unità di controllo
  - Registri
  - Cache di primo livello
- Comunica con il resto della struttura tramite una serie di piedini
- La struttura interna del microprocessore sarà discussa nella parte V

## Pinout logico del $\mu P$

---



- Il microprocessore comunica col mondo esterno tramite un numero alto, ma comunque limitato di piedini
- Scambia dati, indirizzi e segnali di controllo di vario tipo
- Lo stesso piedino fisico è usato spesso per più segnali logici

# L'architettura Intel

---

- La stragrande maggioranza dei PC sono oggi basati sulla cosiddetta piattaforma Intel
- Questa comprende:
  - Il microprocessore (8086, 386, .. , Pentium, .. , Pentium 4)
  - Una serie di bus di interconnessione tra  $\mu$ P, memoria e periferiche di I/O (inclusi dischi)
  - Il *chipset*, costituito ad un insieme di chip che gestiscono la piattaforma
- Altri costruttori (AMD) commercializzano  $\mu$ P funzionalmente equivalenti, cioè che eseguono le stesse istruzioni
- I chipset più recenti (Centrino) includono anche funzionalità aggiuntive (es. connettività wireless)

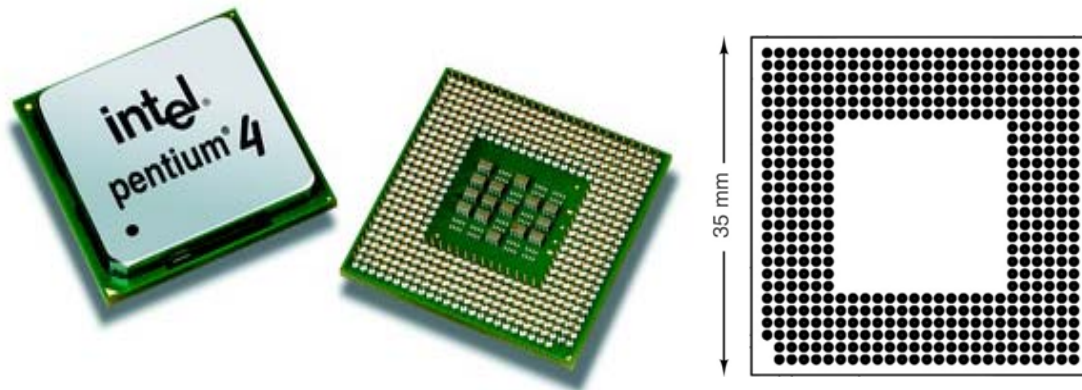
## Pentium 4

---

- Architettura a 32 bit compatibile con i predecessori
- Trasferimenti interni a 64 bit
- Architettura superscalare: unità funzionali multiple
- Oltre 50 milioni di transistor, clock a oltre 3 GHz
- Aritmetica intera e Floating-point IEEE 754
- Bus di memoria a 64 bit
- Cache 1° livello 8K+8K
- Cache 2° livello da 256 KB a 2 MB a seconda dei tipi
- Cache 3° livello nel tipo Extreme (L2=512 KB, L3=2 MB)
- Front side bus a 800 MHz
- Dissipa oltre tra i 50 e i 100W!
- Speed-step: varia frequenza e tensione per risparmiare energia

## Pentium 4: chip e zoccolo

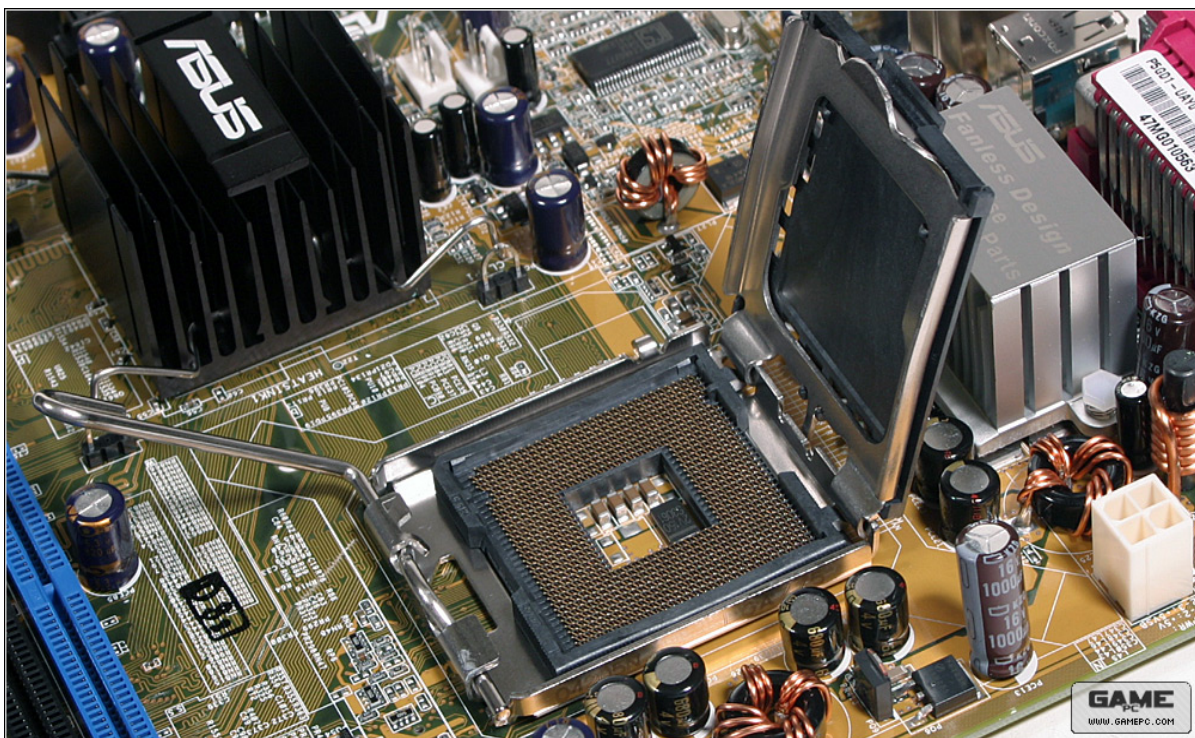
---



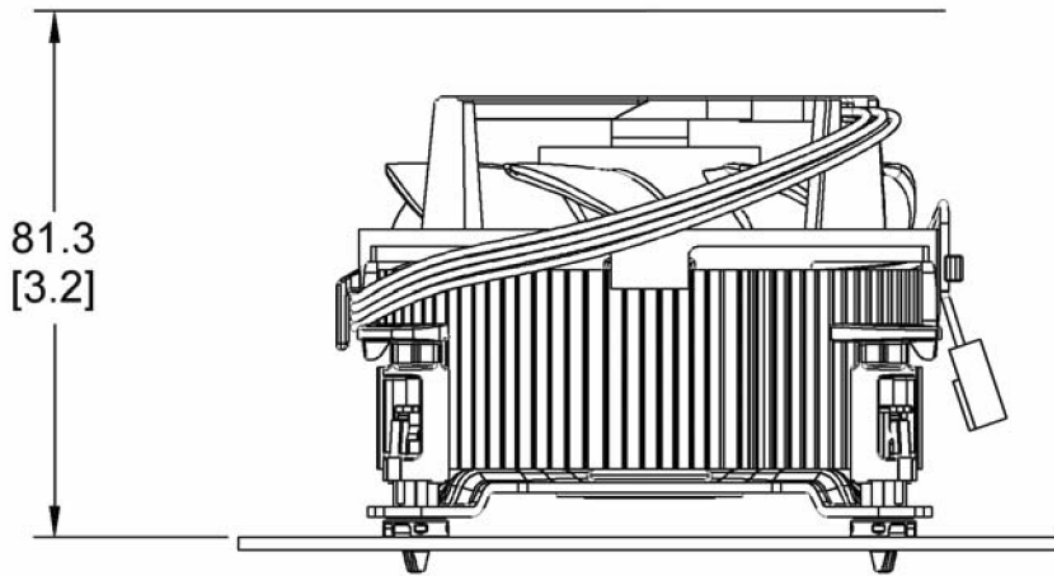
- 478 piedini fisici (vari piedini logici per molti di essi)
- 85 piedini di alimentazione
- 180 piedini di massa
- A causa dell'energia dissipata occorre montare un grosso dissipatore con ventola

## Pentium 4: lo zoccolo

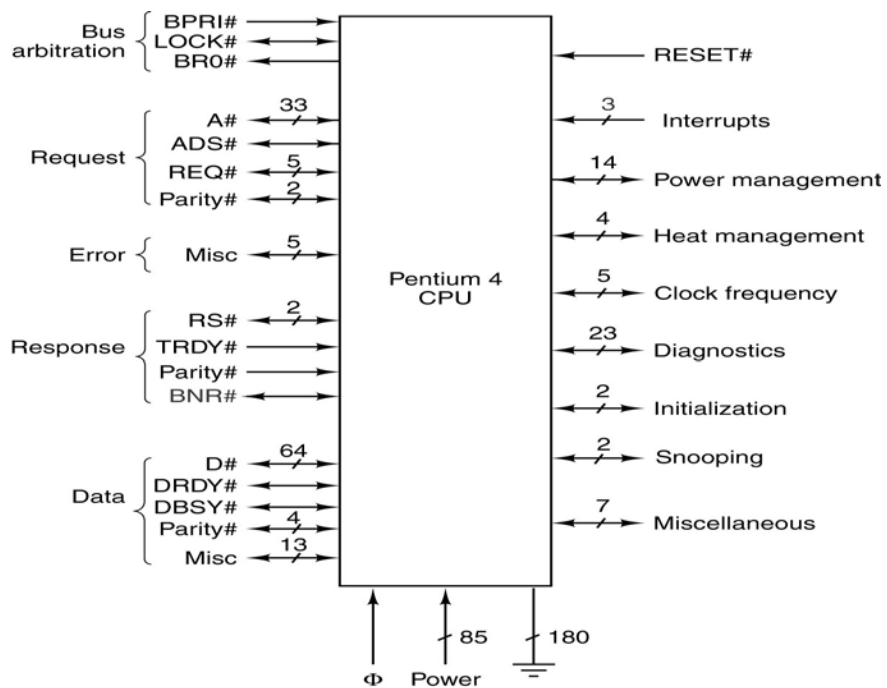
---



# Pentium 4: dissipatore e ventola



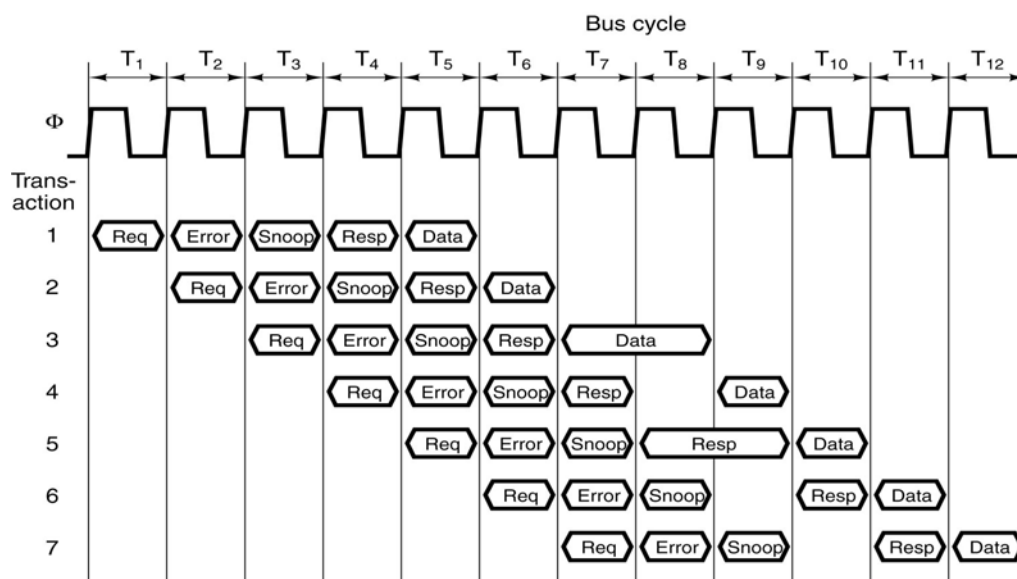
# Pentium 4: pinout logico



## Pentium 4: pinout logico (2)

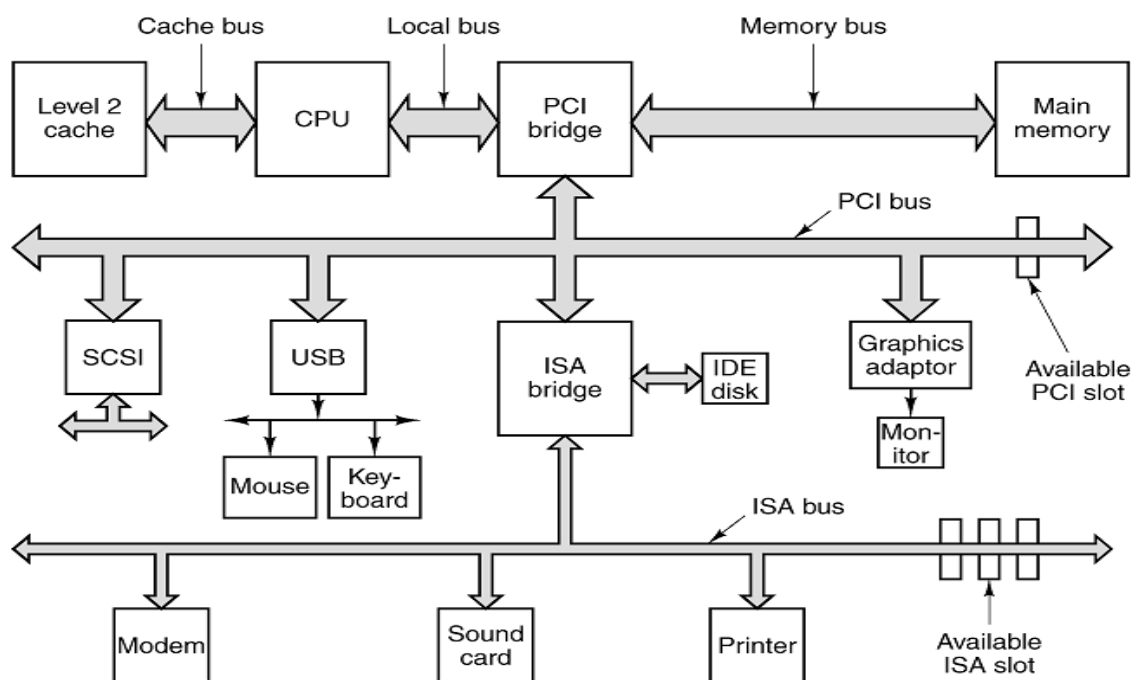
- Indirizzi a 36 bit
  - I tre bit meno significativi sempre a 0
  - Letture sempre in blocchi di 8 byte (*usati solo 33 bit*)
  - Potenzialmente indirizza fino a 64 GB
- Bus con 64 linee dati
- Segnali di *Snoop* per la *coerenza di cache*
- Gestione delle interruzioni sia come l'8088 che tramite dispositivo APIC (Advanced Programmable Interr. Controller)
- 4 piedini per il controllo della temperatura
- *Power Management*: permette di mettere la CPU in stato di *sleep* e *deep sleep*
- 23 linee di diagnosi secondo lo standard IEEE 1149.1 JTAG

## Pentium 4: accesso alla memoria



- Gestione in pipeline dei cicli di memoria
- Possibile sovrapporre fino a 8 transazioni

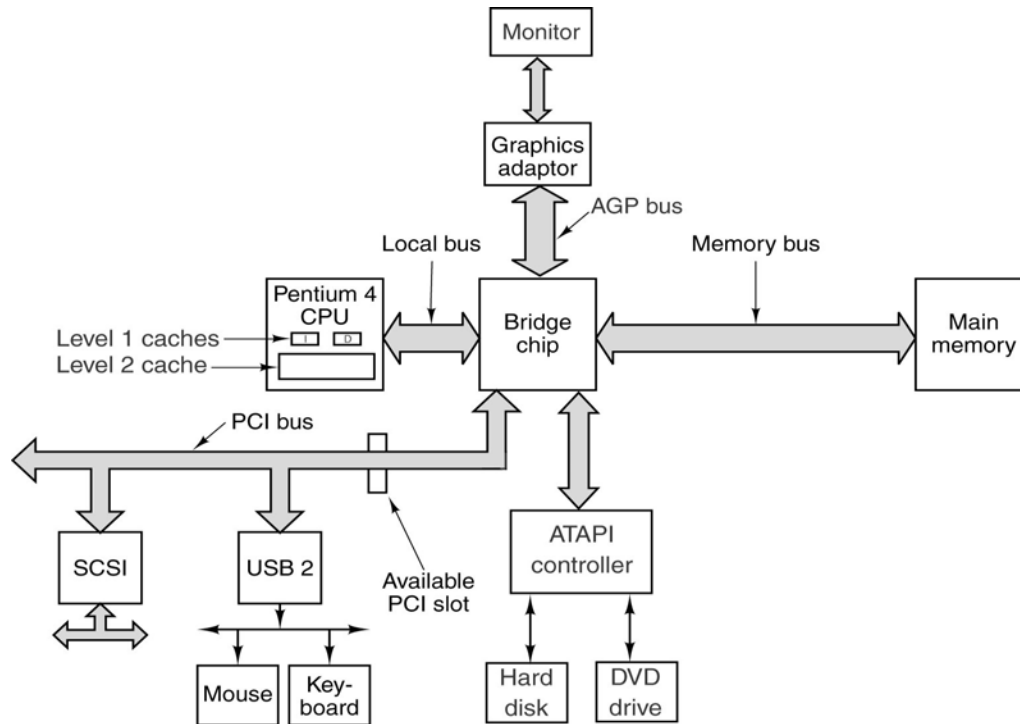
## Architettura 'classica' della piattaforma Intel



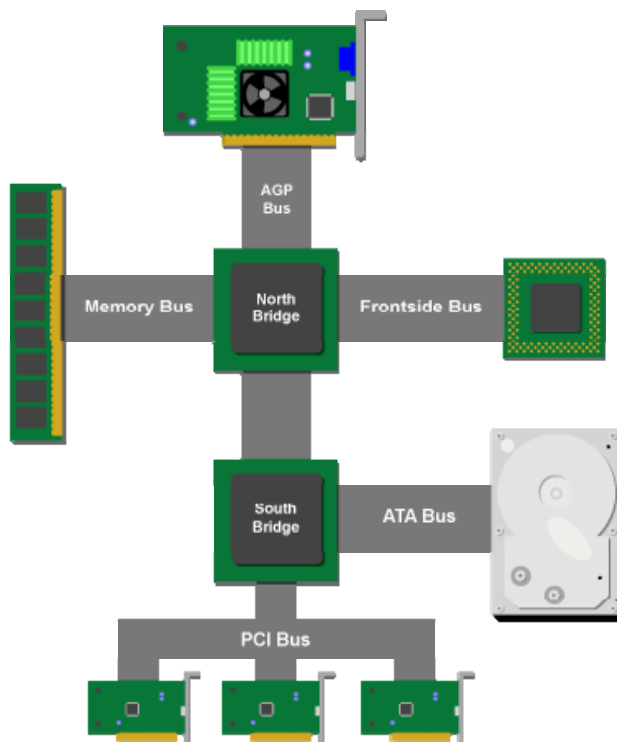
## Bus della piattaforma Intel

- La piattaforma prevede una molteplicità di bus
- La CPU accede alla memoria centrale tramite il *Local Bus* (detto anche *Front Side Bus*) ed il *Memory Bus*
- Il *PCI Bridge* costituisce il principale componente del cosiddetto *chipset*, consente l'accesso condiviso, da parte della CPU e dei controllori di I/O, alla memoria centrale
- Il bus PCI è il bus di I/O principale
- I dischi IDE-ATA vengono gestiti da un controllore autonomo
- Un eventuale bus SCSI è interfacciato tramite un *adapter*
- Per limitare la congestione sul bus PCI è stato aggiunto il bus AGP, cui viene interfacciata la scheda grafica
- Lo *ISA Bridge* interfaccia verso il bus legacy ISA (*Industry Standard Architecture*), che era il vecchio bus del PC IBM

# Evoluzione della piattaforma Intel



# North bridge e South bridge



- North bridge
  - CPU
  - Memoria
  - Scheda video
- South bridge
  - Tutto il resto dell'I/O
  - Dischi
  - Bus PCI



## Il bus PCI

---

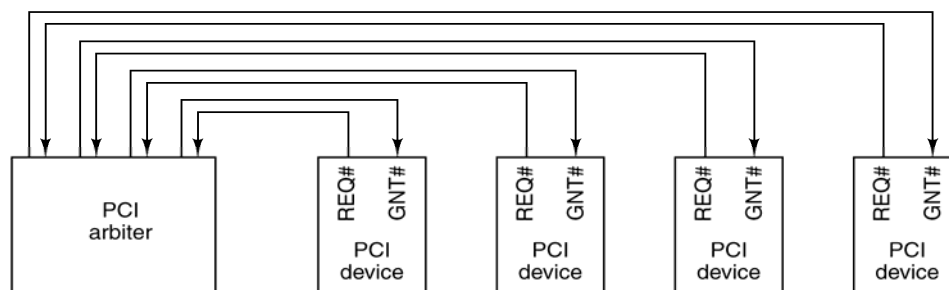
- PCI (*Peripheral Component Interconnect*)
- Introdotto da Intel per supportare le esigenze video
  - Es: risoluzione di 1024×768×3 Byte a 30 frame/sec
  - Richiede una banda di 67.5 MB/sec
- Standard proposto da Intel ma *non proprietario*
- Versione base (1992) a 32 bit, 33 MHz: 133 MB/sec
- Estensione a 64 bit e 133 MHz: fino a 1 GB/sec
- Gestione flessibile degli indirizzi e dei numeri di interruzione
- Supporta il *Plug and Play*
- Limitata lunghezza e numero di dispositivi: al massimo 3-4
- Possibile espansione con bridge *PCI-to-PCI*
- Le piattaforme parallele hanno in genere più bus PCI

## Il bus PCMCIA

---

- PCMCIA (*Personal Computer Memory Card Intl Association*)
- Nato nel 1990 per connettere estensioni di memoria ai portatili
- Successivamente estesa per periferiche ISA e PCI
- PCMCIA 2.1 (1993):
  - Bus a 16 bit
  - Clock a 10 MHz
  - Banda 20 MBps
- *Plug and Play* e *hot swappable*
- Estensione successiva a 32 bit (CardBus)
  - Clock a 33 MHz
  - Banda 120 MBps
  - Supporta DMA e bus mastering

## Bus PCI: arbitraggio



- Arbitraggio centralizzato (nel Bridge)
- Ogni *PCI device* ha due linee dedicate
- Il device fa la *richiesta* tramite REQ#
- Il *grant* viene concesso tramite GNT#
- Diversi algoritmi di arbitraggio:
  - *Round Robin*
  - *Priorità*
  - *Altro*

## Bus PCI: segnali obbligatori

Signal	Lines	Master	Slave	Description
CLK	1			Clock (33 MHz or 66 MHz)
AD	32	×	×	Multiplexed address and data lines
PAR	1	×		Address or data parity bit
C/BE	4	×		Bus command/bit map for bytes enabled
FRAME#	1	×		Indicates that AD and C/BE are asserted
IRDY#	1	×		Read: master will accept; write: data present
IDSEL	1	×		Select configuration space instead of memory
DEVSEL#	1		×	Slave has decoded its address and is listening
TRDY#	1		×	Read: data present; write: slave will accept
STOP#	1		×	Slave wants to stop transaction immediately
PERR#	1			Data parity error detected by receiver
SERR#	1			Address parity error or system error detected
REQ#	1			Bus arbitration: request for bus ownership
GNT#	1			Bus arbitration: grant of bus ownership
RST#	1			Reset the system and all devices

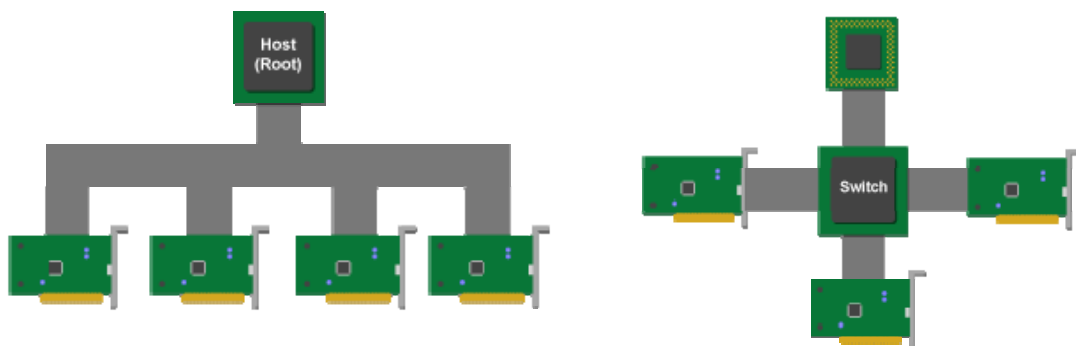
## Bus PCI: segnali

---

- Bus a 120 o 180 linee
- Oltre ai segnali sono distribuiti anche alimentazioni e masse
- 32(+32) linee AD multiplexate tra dati e indirizzo con 1(+1) bit di parità PAR
- C/BE# (in cicli diversi) invia comandi e specifica quanti e quali byte leggere
- FRAME# e IRDY (lettura) usati dal master
- DEVSEL# e TRDY# usati dallo slave
- IDSEL indirizza nel *configuration space* di un device invece che in memoria (usato per gestire il *Plug and Play*)
- STOP# e PERR# segnalano errori
- RST# induce un reset nella CPU e in tutti i device sul bus
- SBO e SDONE segnali di *cache snooping*

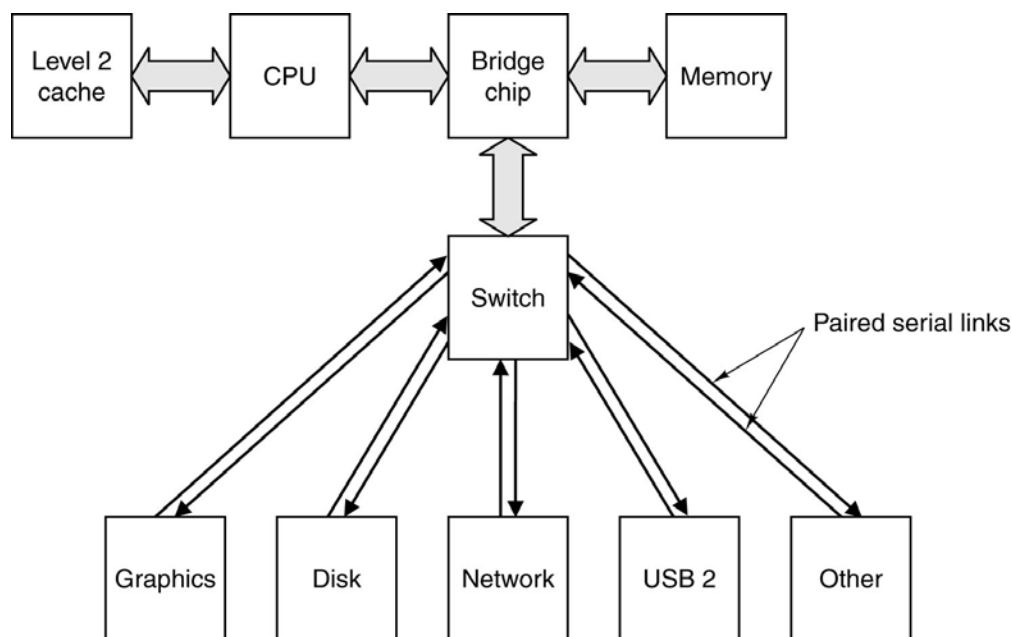
## Bus PCI Express (PCIe)

---



- Il PCI ha il suo limite nell'essere un bus condiviso
- Per ottenere una banda maggiore occorre passare ad una architettura a *switch*
- Topologia punto-punto: ciascun dispositivo ha un canale di comunicazione dedicato
- La banda cresce al crescere del numero di dispositivi

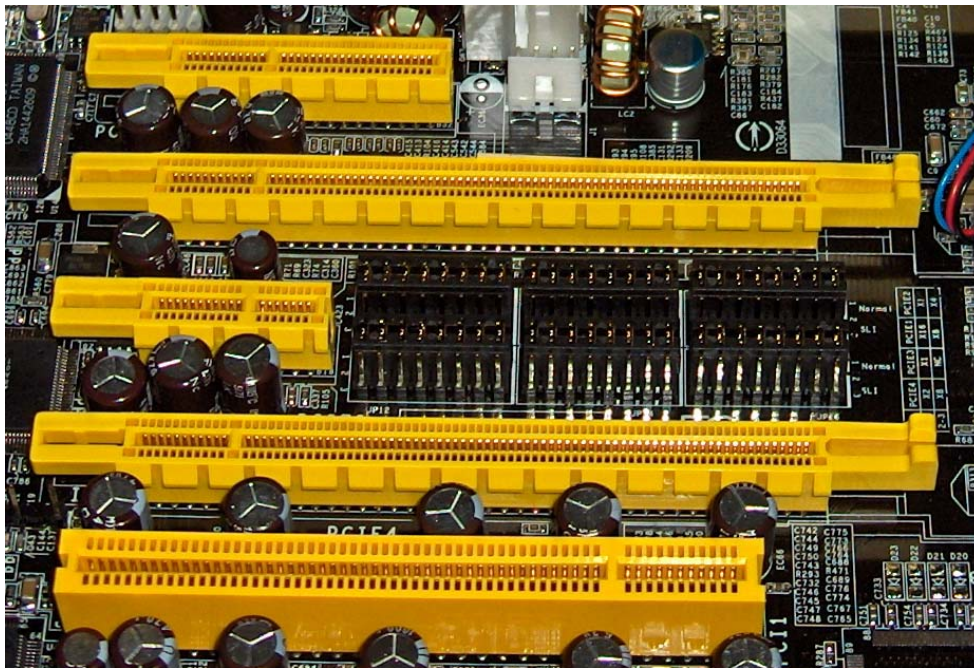
# PCI Express: topologia di interconnessione



## Organizzazione del PCI Express

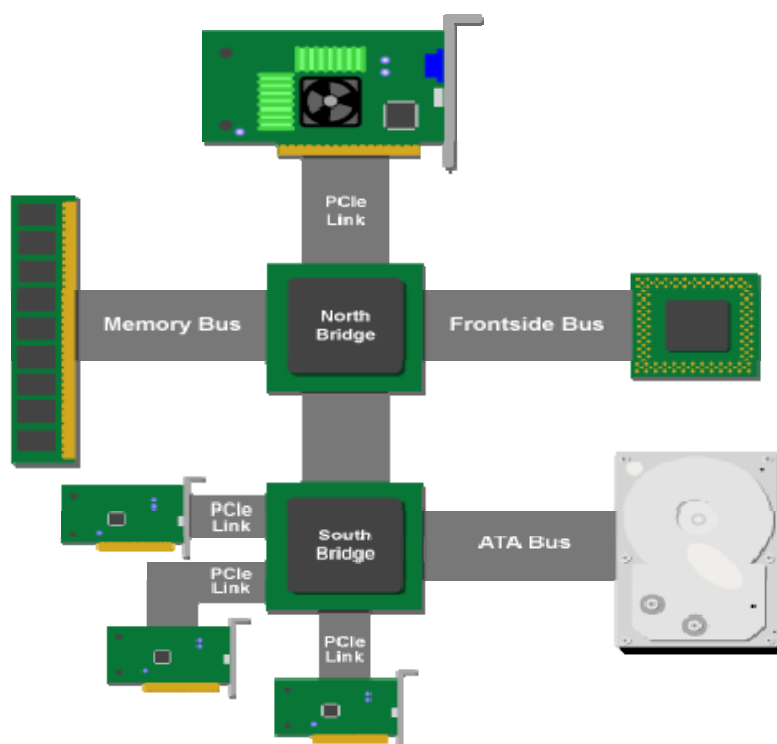
- Ogni dispositivo ha una sua connessione punto-punto
- Ciascun dispositivo è connesso tramite una coppia di *canali unidirezionali*
- Ogni canale è seriale: costituito da una coppia di conduttori
- Per gestire dispositivi con bande diverse si possono avere connessioni con più coppie di canali (*lane*)
- Possibile avere 2, 4, 16 e 32 lane
- Ciascuna connessione ha una banda tra 2.5 Gbps e 10 Gbps
- Scambi di dati a pacchetti, come in una rete
- La struttura è espandibile: allo switch possono essere connessi altri switch

# Connettori PCI Express



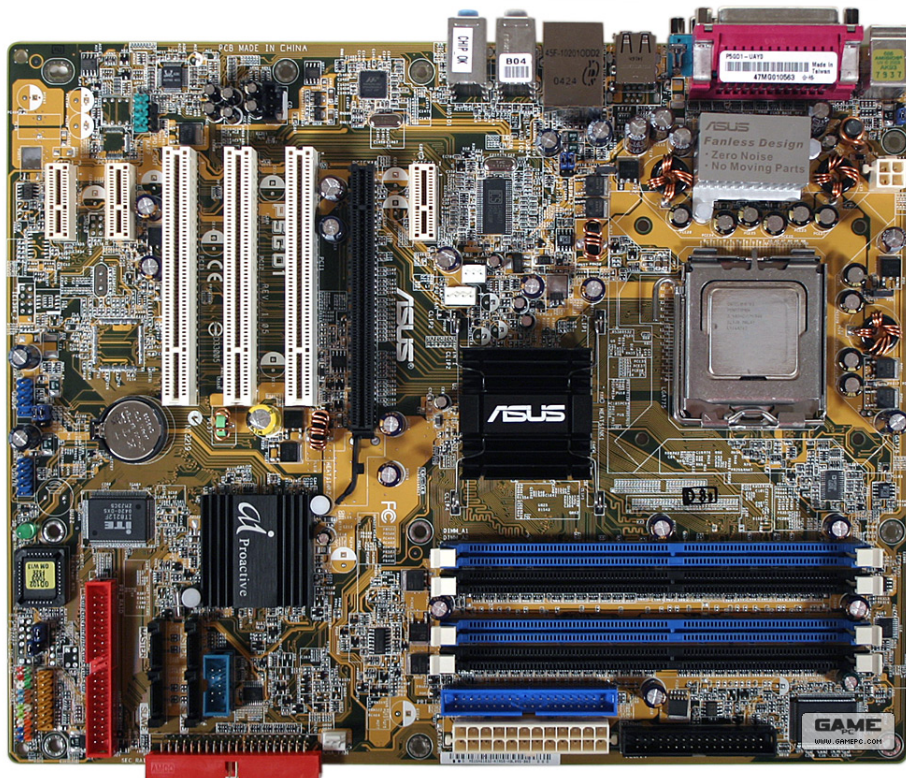
Dall'alto: PCIe x 4, PCIe x 16, PCIe x 1, PCIe x 16, vecchio PCI a 32 bit

# PCI express nei nuovi chipset



## Scheda madre socket 775

---



Calcolatori Elettronici - prof. Giuseppe Santucci

IV.43

## USB Universal Serial Bus

---

- Bus *economico* concordato da varie aziende per la gestione di device di I/O a bassa velocità (~ 1995)
- Obiettivi:
  - Evitare switch, jumpers
  - Installazione di tipo *esterno*
  - Cavo di connessione unificato
  - Alimentazione fornita dal cavo
  - Fino a 127 dispositivi collegabili
  - Supporto di dispositivi *real-time*
  - Installazione a PC acceso
  - *Reboot* non necessario
  - Bus e dispositivi economici

Calcolatori Elettronici - prof. Giuseppe Santucci

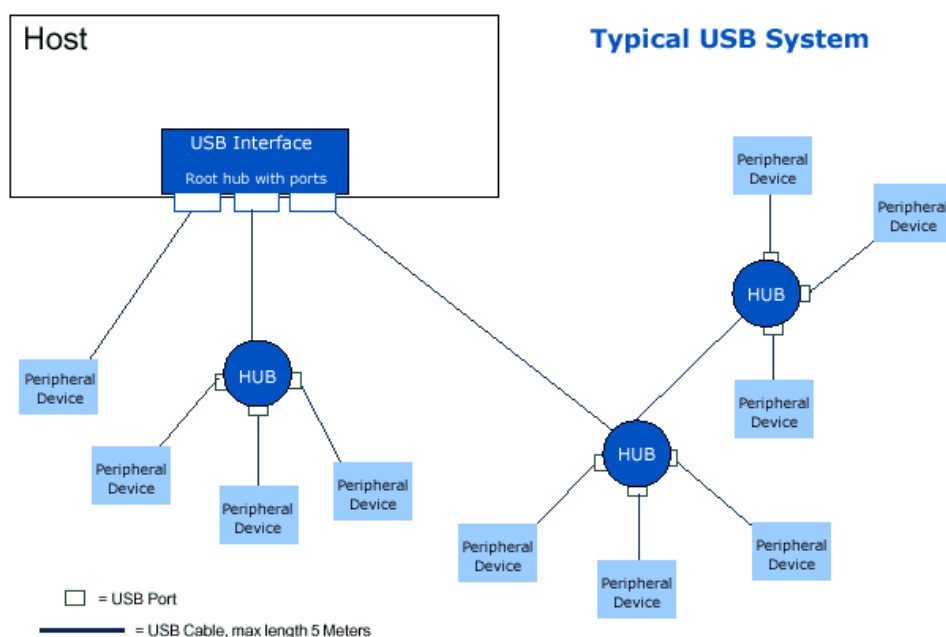
IV.44



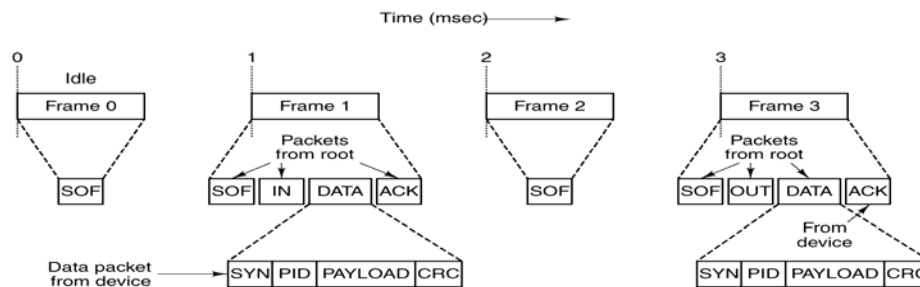
# USB 1: specifiche fondamentali

- Banda complessiva 1.5 Mbps (USB 1.0), limitata per ragioni di costo, sale a 12 Mbps in USB 1.1
- *Root hub* di connessione al bus PCI
- Connessione di dispositivi e di altri *hub*
- Struttura complessiva ad albero con massima ramificazione di 16
- Cavo a 4 fili: +5V, GND, 2 di segnale
- Connessione di un dispositivo:
  - Il root hub genera un'interruzione: intervento del SO
  - Il dispositivo specifica la *banda richiesta*
  - Il SO assegna al dispositivo un indirizzo da 1 a 128
- L'indirizzo 0 è riservato all'inizializzazione
- Si crea una connessione logica tra *root hub* e ciascun *device* cui corrisponde una *bit pipe dedicata*

# USB: connessione dei dispositivi



# USB: comunicazione con i dispositivi



- Il tempo è diviso in *frame* di durata fissa, scanditi dall'host:
  - SOF (*Start of Frame*): specifica il dispositivo interessato
  - IN / OUT: specifica la direzione della comunicazione
  - DATA: *payload* più controllo e codice di errore
  - ACK/NACK: acknowledge o segnalazione di errore
- *Idle frame* se non c'è comunicazione

**N.B.** Non ci sono interruzioni: comunicazione basata sul *polling*

## USB 2.0: evoluzione dello standard

- Lo standard USB ha avuto grande successo
- Estensione per consentire la connessione di dispositivi che richiedono una banda maggiore
- USB 2.0 è completamente compatibile all'indietro
- Lo host controller determina il tipo di dispositivo, e lo gestisce di conseguenza
- Tre categorie di dispositivi:
  - *high speed*: fino a 480 Mbps (60 MBps)
  - *full speed*: 12 Mbps (1.5 MBps)
  - *low speed*: 1.5 Mbps

*Un'evoluzione probabile è verso una semplificazione e la presenza nei PC di sole interfacce USB e Firewire*



# Raffronto fra i tipi di bus

Type of Bus	Bus Clock Signal	Bus Width	Theoretical Maximum Transfer Rate	Advantages	Disadvantages
ISA	8 MHz	16-bit	8 Mbytes/sec (64 Mbits/sec)	low cost, compatibility, widely used	low speed, Jumpers & DIP switches. becoming obsolete
Microchannel	10 MHz	32-bit	40 Mbytes/sec (240 Mbits/sec)	higher speed than ISA	obsolete
PCI	133 MHz	64-bit	1 Gbytes/sec (8 Gbits/sec)	very high speed, Plug & Play, dominant board-level bus	incompatible with older systems, can cost more
CompactPCI	33 MHz	64-bit	132 Mbytes/sec (1 Gbit/sec)	designed for industrial use, hot swapping/Plug & Play, ideal for embedded systems	lower speed than PCI, need adapter for PC use, incompatible with older systems
PCMCIA	10 MHz	16-bit	20 Mbytes/sec (160 Mbits/sec)	Ideal for portable systems, hot swappable, Plug & Play	lower speed, needs special drive for use in desktop PCs
USB 1.1	n/a	n/a	1.5 Mbytes/sec (12 Mbits/sec)	low cost, ideal for portable systems, hot swapping/plug & play, up to 127 devices via 1 port	slower than PCI and other plug-in busses (such as Firewire), not compatible with older peripherals
USB 2.0	n/a	n/a	60 Mbytes/sec (480 Mbits/sec)	All the advantages of USB plus significantly higher speeds making it compatible with high-speed peripherals such as data drives and video cameras.	Not compatible with older peripherals, still slower than PCI
Ethernet 10 Base T	n/a	n/a	1.25 Mbyte/sec (10 Mbits/sec)	Enables multiple PCs to remotely share information and peripheral devices, provides error checking lacking in standard serial communication	Most peripheral devices cannot be connected directly to ethernet, adapter is required. Slow data communication by current standards Possible security issues.
Ethernet 100 Base T (Fast Ethernet)	n/a	n/a	12.5 MBytes/sec (100 Mbits/sec)	All advantages of 10BaseT, with significant speed improvement. Backward compatible with 10Base T installations.	Most peripheral devices cannot be connected directly to ethernet, adapter is required. Possible security issues.

## Gestione dell'I/O

- I dispositivi di I/O comprendono sia le periferiche di ingresso e uscita vere e proprie che i dispositivi di memoria di massa
- Ciascun dispositivo è gestito direttamente da un hardware specifico denominato *controllore* (in genere include un  $\mu P$ )
- La CPU può leggere e scrivere nei registri dei controllori, indirizzandoli come se fossero celle di memoria
- La comunicazione fra CPU e controllori avviene con due modalità distinte:

- POLLING: il controllore è *passivo* e la CPU lo interroga periodicamente per vedere se ha bisogno di attenzione
- INTERRUZIONI: il controllore invia alla CPU appositi segnali (interruzioni) quando ha bisogno di attenzione

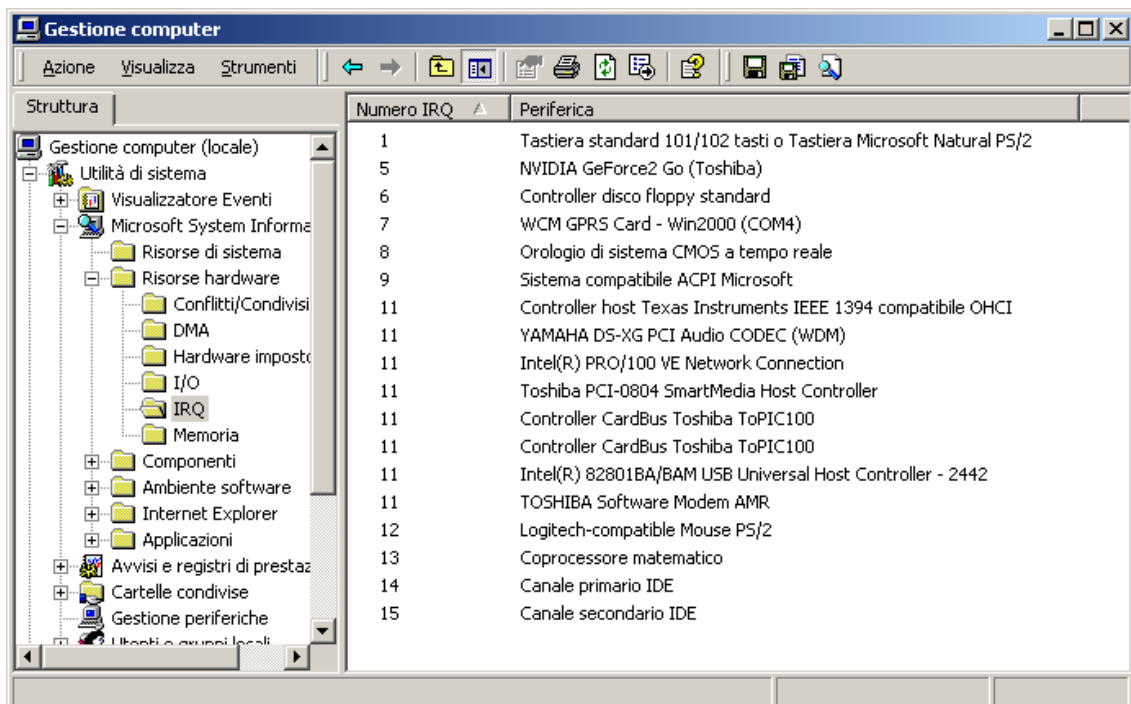
# I/O con interrupt

- I dispositivi inviano alla CPU *segnali di interruzione* utilizzando apposite linee del bus
- La CPU interrompe l'elaborazione corrente e interagisce con il dispositivo che la ha interrotta per gestire l'operazione di I/O

## ES Lettura da disco

- La CPU avvia l'operazione di I/O scrivendo le informazioni opportune nelle porte del controller
- La CPU passa all'elaborazione di un altro task
- Il controller avvia e sovrintende allo svolgimento dell'operazione di I/O (posizionamento delle testine ecc.)
- Quando i dati sono disponibili il controller interrompe la CPU
- La CPU è direttamente coinvolta nel trasferimento dei dati tra controller, essa legge i dati e li copia in memoria

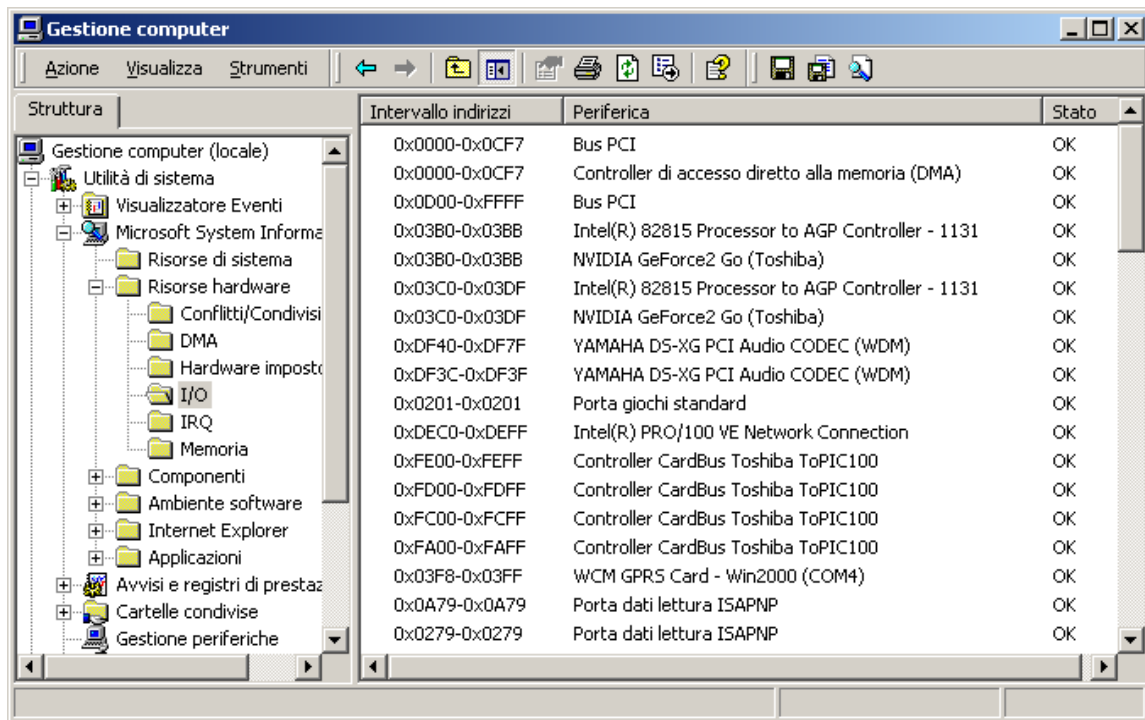
# Numeri di interruzione in Windows (IRQ)



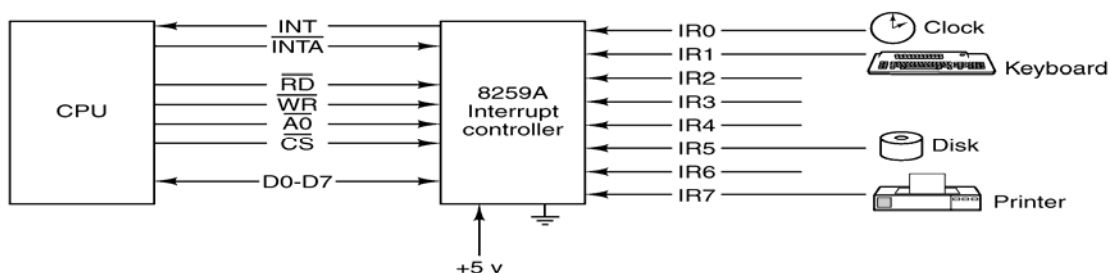
The screenshot shows the 'Gestione computer' (Computer Management) window in Windows XP. The 'Struttura' (Structure) pane on the left is expanded to 'Risorse hardware' > 'I/O' > 'IRQ'. The main pane displays a table of IRQs and their corresponding devices.

Numero IRQ	Periferica
1	Tastiera standard 101/102 tasti o Tastiera Microsoft Natural PS/2
5	NVIDIA GeForce2 Go (Toshiba)
6	Controller disco floppy standard
7	WCM GPRS Card - Win2000 (COM4)
8	Orologio di sistema CMOS a tempo reale
9	Sistema compatibile ACPI Microsoft
11	Controller host Texas Instruments IEEE 1394 compatibile OHCI
11	YAMAHA DS-XG PCI Audio CODEC (WDM)
11	Intel(R) PRO/100 VE Network Connection
11	Toshiba PCI-0804 SmartMedia Host Controller
11	Controller CardBus Toshiba ToPIC100
11	Controller CardBus Toshiba ToPIC100
11	Intel(R) 82801BA/BAM USB Universal Host Controller - 2442
11	TOSHIBA Software Modem AMR
12	Logitech-compatible Mouse PS/2
13	Coprocessore matematico
14	Canale primario IDE
15	Canale secondario IDE

# Indirizzi di I/O in Windows



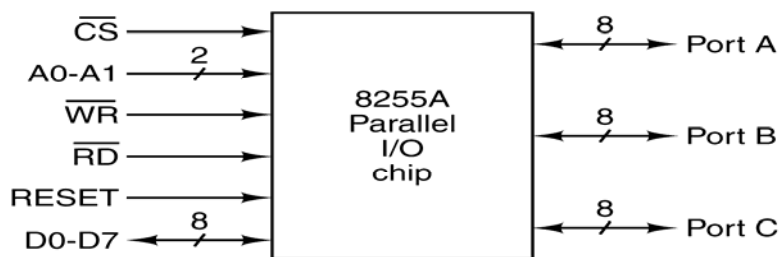
## Controllore delle interruzioni



- Chip Intel 8259A usato dal PC IBM e successori
- Gestisce 8 linee di interrupt
  - **INT**: interruzione inviata alla CPU
  - **INTA**: *acknowledge* della CPU
  - *Vettore di Interrupt* passato sul bus
  - **IR0-IR7**: linee di interrupt sul bus
- Il vettore di interrupt consente di saltare alla relativa routine
- Registri all'interno del chip scrivibili dalla CPU per programmare lo 8259A

# Chip di I/O: UART, USART e PIO

---



- **UART** (Univ. Async. Rec. Transm.)
- **USART** (... Sync. Async. ....)
- Usati per interconnettere dispositivi al microprocessore
- **PIO** (Parallel Input/Output)
  - Configurabile dalla CPU
  - 3 Porti indipendenti da 8 bit con latch
  - La CPU legge e scrive nei porti