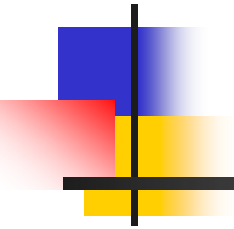
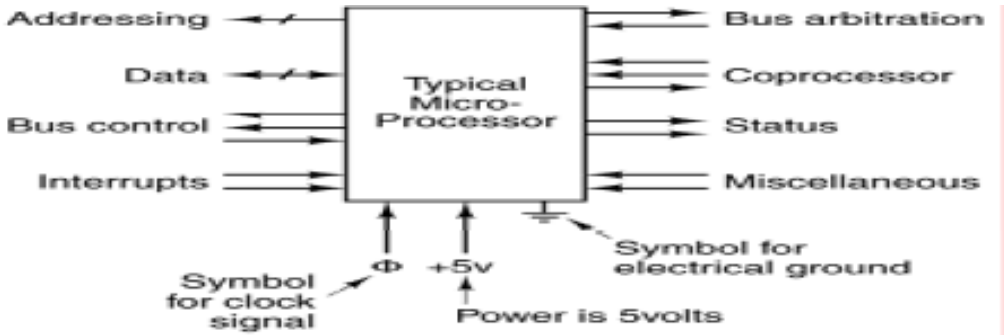


Cal Corso di Calcolatori Elettronici (per Elettronici)

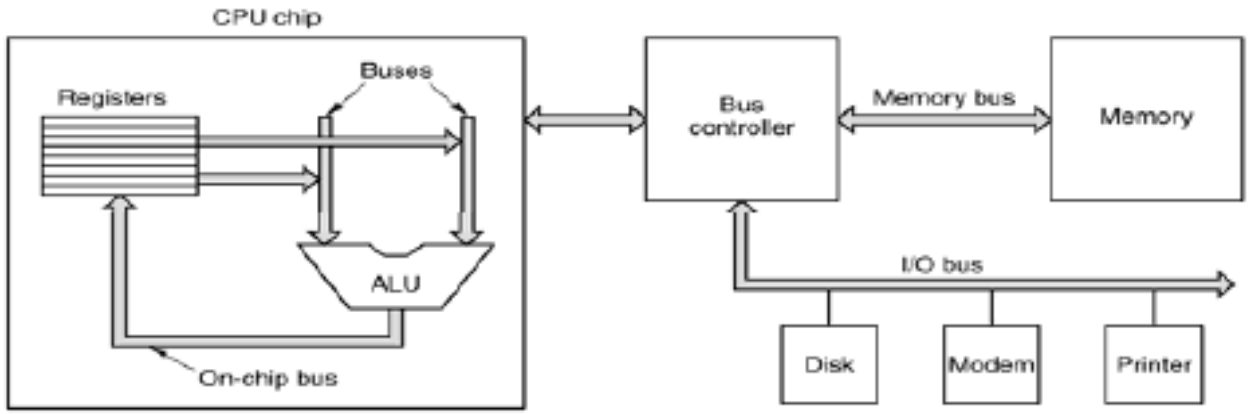
Parte VI: BUS e I/O

Pin-out Logico di un Micro-processore



- Indirizzamento
- Dati
- Controllo:
  - Controllo del bus
  - Gestione delle interruzioni
  - Arbitraggio del bus
  - Gestione del coprocessore
  - Segnalazione di stato
  - Vari (alimentazione etc.)

Architettura a più Bus



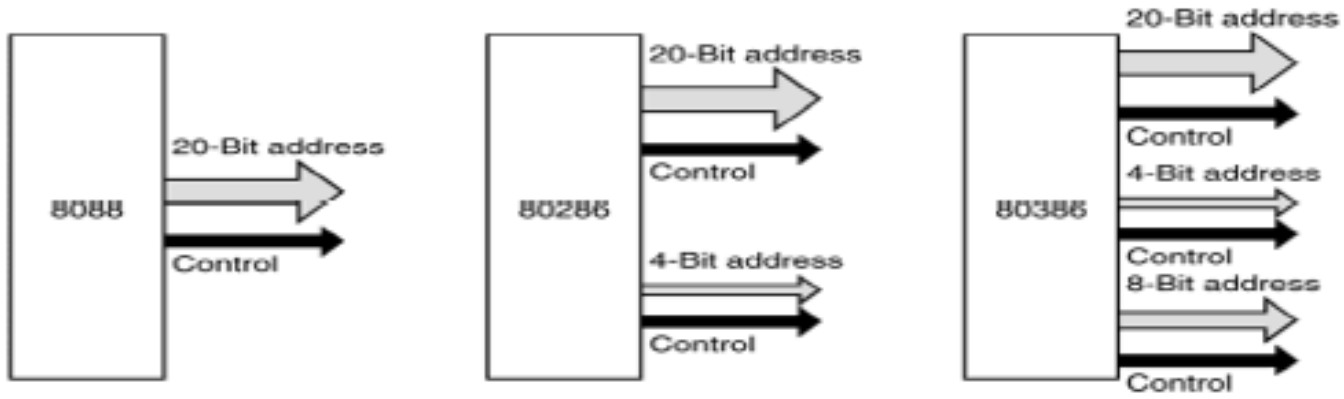
- Diversi bus, interni ed esterni al chip
- Soddisfano diverse esigenze:
  - Velocità di trasferimento
  - Numero di linee
  - Più trasferimenti paralleli
  - Compatibilità all'indietro
- Nei primi PC c'era un unico bus
- Negli attuali PC almeno tre bus esterni

Comunicazione sul Bus

Master	Slave	Example
CPU	Memory	Fetching instructions and data
CPU	I/O device	Initiating data transfer
CPU	Coprocessor	CPU handing instruction off to coprocessor
I/O	Memory	DMA (Direct Memory Access)
Coprocessor	CPU	Coprocessor fetching operands from CPU

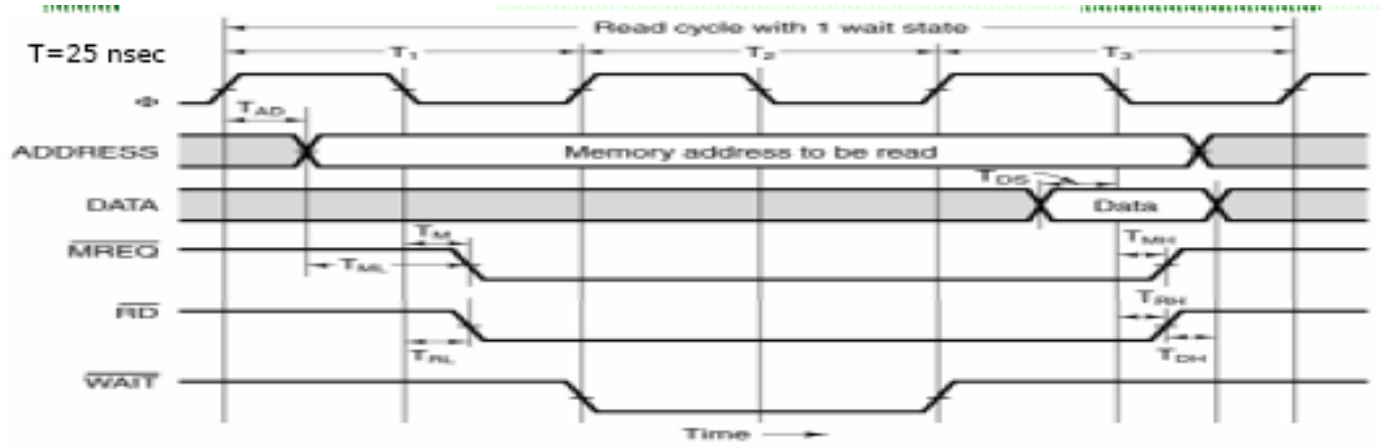
- La comunicazione sul bus è regolata da un *protocollo di bus*
- In ciascun ciclo comunicano due soli dispositivi il **master** e lo **slave**
- Lo stesso dispositivo può avere ruoli diversi a seconda dei casi
- I dispositivi sono connessi al bus tramite un *bus transceiver*
- La connessione al bus o avviene tramite dispositivi a tre stati oppure è di tipo *open collector*

Larghezza del Bus



- Larghezza = numero di linee
- Linee dati: banda di trasferimento
- Linee indirizzo: dimensione dello spazio (di memoria) indirizzabile, 2^n locazioni con *n* bit di indirizzo
- **Problema:** al crescere della velocità del bus aumenta il *bus skew* (differenza nella velocità di propagazione dei segnali su linee diverse)
- Multiplexamento di più segnali sulla stessa linea per diminuire i costi

Bus sincroni: ciclo di lettura



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 Φ in T_1		8	nsec
T_{RL}	$\overline{RD}$ delay from falling edge of Φ in T_1		8	nsec
T_{DS}	Data setup time prior to falling edge of Φ	5		nsec
T_{MH}	$\overline{MREQ}$ delay from falling edge of Φ in T_3		8	nsec
T_{RH}	$\overline{RD}$ delay from falling edge of Φ in T_3		8	nsec
T_{DH}	Data hold time from negation of $\overline{RD}$	0		nsec

- Tutte le azioni avvengono sui fronti
- Se la memoria mantiene asserito $\overline{WAIT}$ il ciclo si prolunga

Bus sincrono: Temporizzazione

ES

Frequenza 40 MHz, periodo 25 nsec.

- primo vincolo: tempo a disposizione della memoria fra:
  a) la comparsa dell'indirizzo sul Bus
  b) la disponibilità dei dati sul Bus

$$\tau_1 = 2.5 \times T - T_{AD} - T_{DS} = 62.5 - 16 = 46.5 \text{ nsec}$$

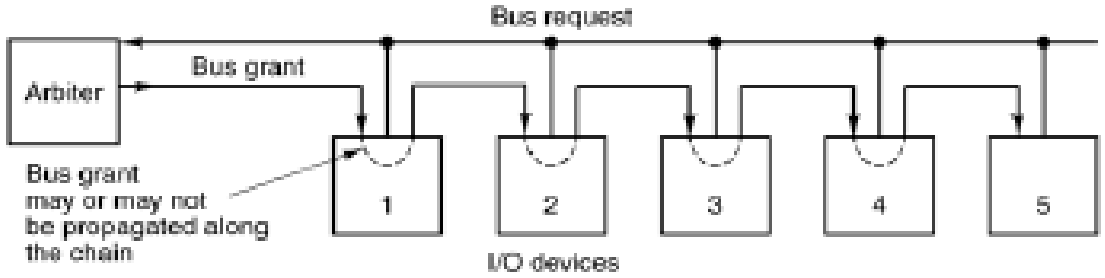
(una memoria da 40 nsec ce la fa di sicuro)

- secondo vincolo: tempo a disposizione della memoria fra:
  a) l'asserzione di $\overline{\text{MREQ}}$ e $\overline{\text{RD}}$
  b) la disponibilità dei dati sul Bus

$$\tau_2 = 2 \times T - T_M - T_{DS} = 50 - 13 = 37 \text{ nsec}$$

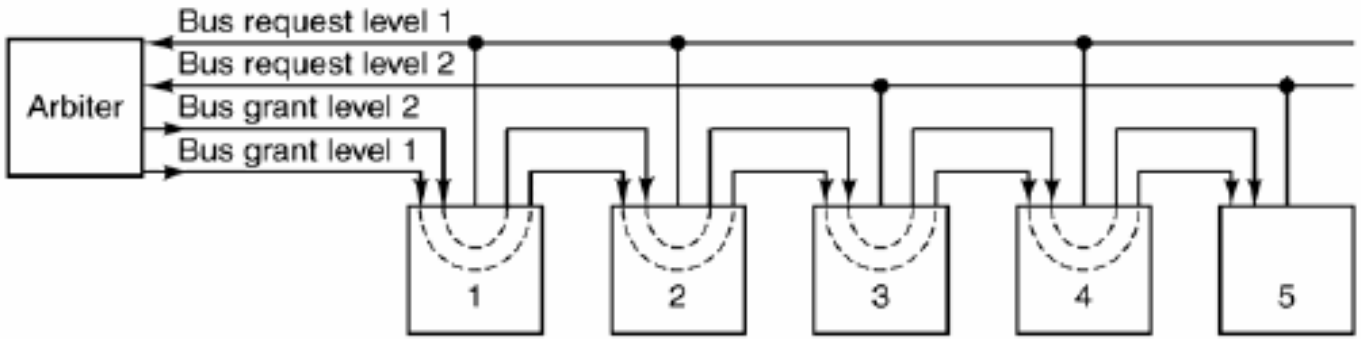
Se il chip di memoria non soddisfa questi requisiti mantiene asserito il segnale di $\overline{\text{WAIT}}$ per introdurre *stati di wait*, cioè cicli di bus addizionali.

Arbitraggio del Bus



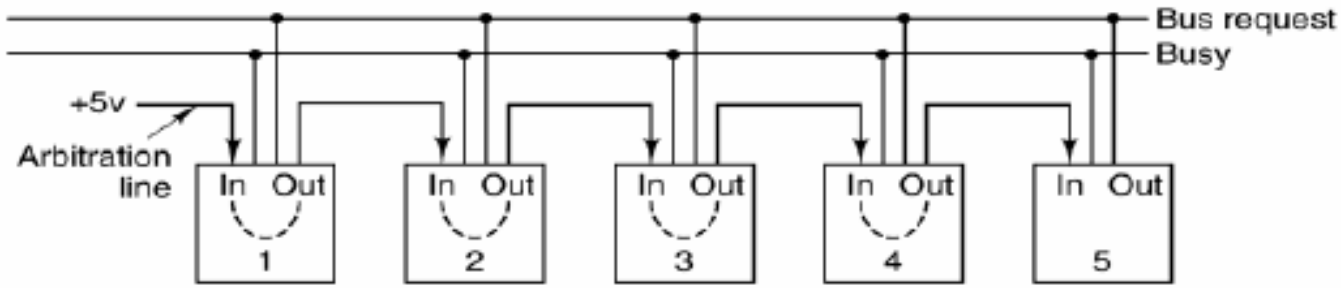
- Permette di decidere quale dispositivo sarà il prossimo Bus Master risolvendo eventuali conflitti
- Spesso l'arbitro è nel chip del µP
- Linea di richiesta condivisa
- Il *Bus grant* è propagato dall'arbitro poco prima dell'inizio del ciclo
- Viene intercettato dal futuro master
- NB: Favoriti i dispositivi situati vicino all'arbitro

Livelli multipli di priorità



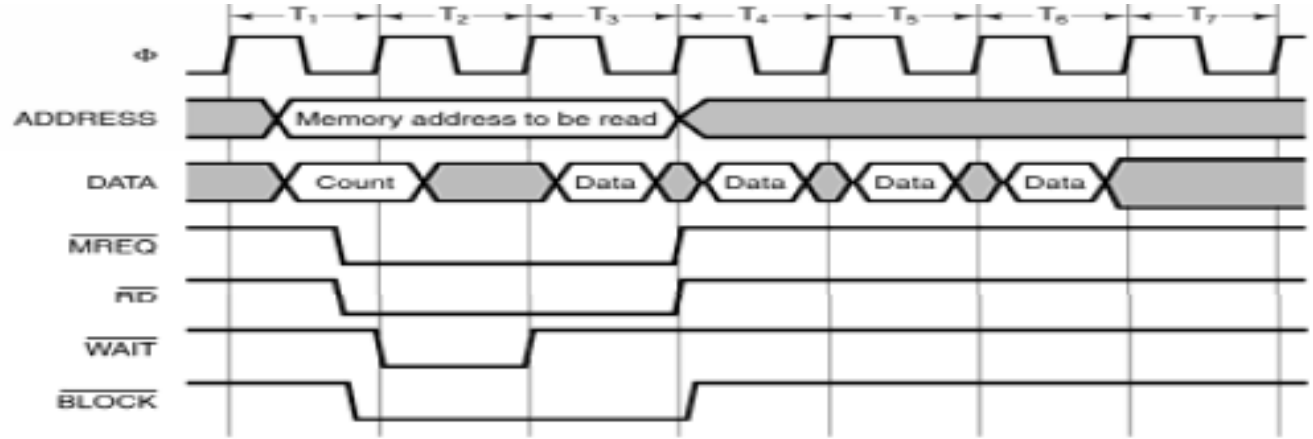
- Diverse linee di richiesta associate a diversi *livelli di priorità*
- In caso di conflitto favorite le catene a priorità più alta
- All'interno di ciascuna catena vale la posizione
- In genere se c'è un solo bus con anche la memoria, la CPU ha priorità più bassa dei dispositivi di I/O (e.g. dischi)

Arbitraggio decentralizzato



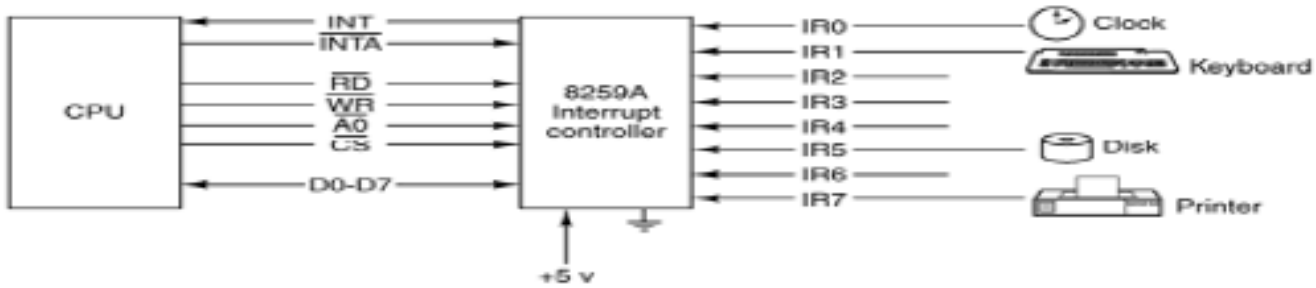
- Quando nessun dispositivo vuole il Bus, la linea di arbitraggio è asserita con propagazione a tutti i dispositivi
- Quando un dispositivo vuole il Bus:
  - invia una richiesta di bus
  - verifica che il bus e' libero
  - se **In** non e' asserito non diventa master e nega **Out**
  - se **In** e' asserito diventa master, nega **Out** e asserisce **Busy**
- Non necessita di arbitro, è più semplice e più veloce

Block transfers



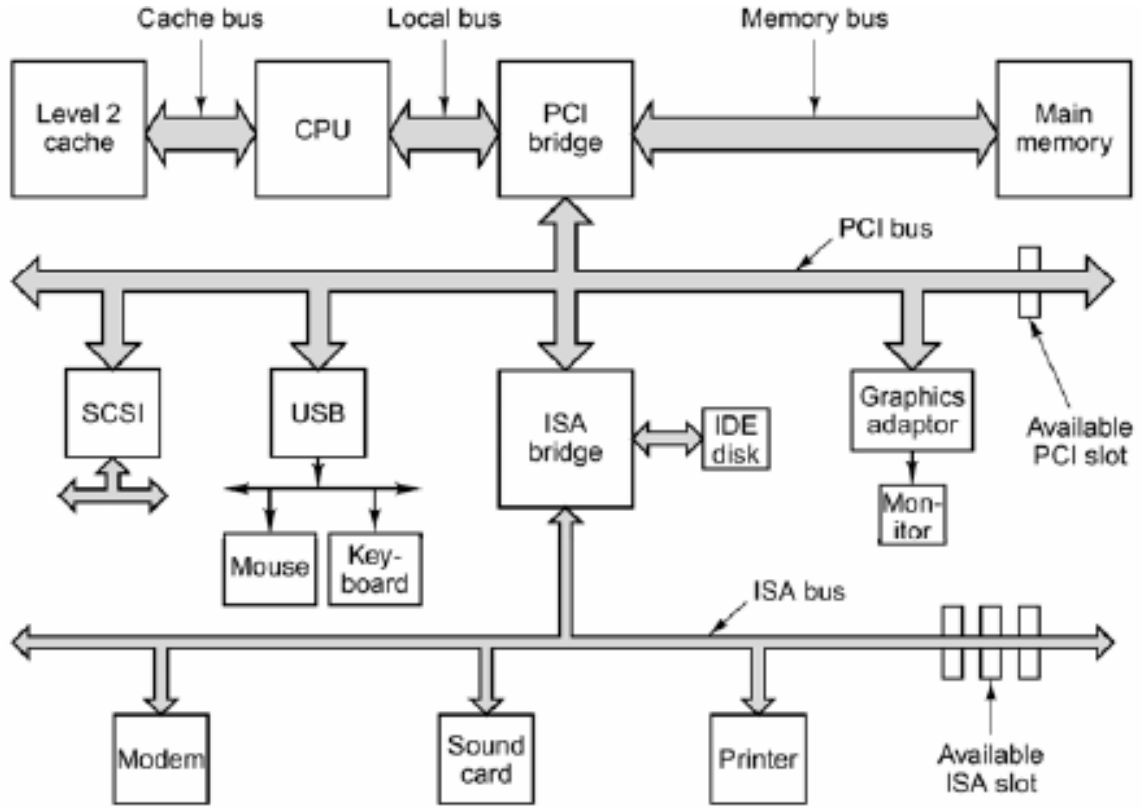
- Permette di leggere più parole consecutive
- Usato per trasferire *blocchi* di cache
- Numero di parole specificato durante T_1
- Dopo la prima viene trasferita una parola ogni ciclo (invece di una ogni tre cicli)
- Per leggere quattro parole occorrono 6 cicli invece di 12
- Il segnale $\overline{\text{BLOCK}}$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

Gestione delle interruzioni



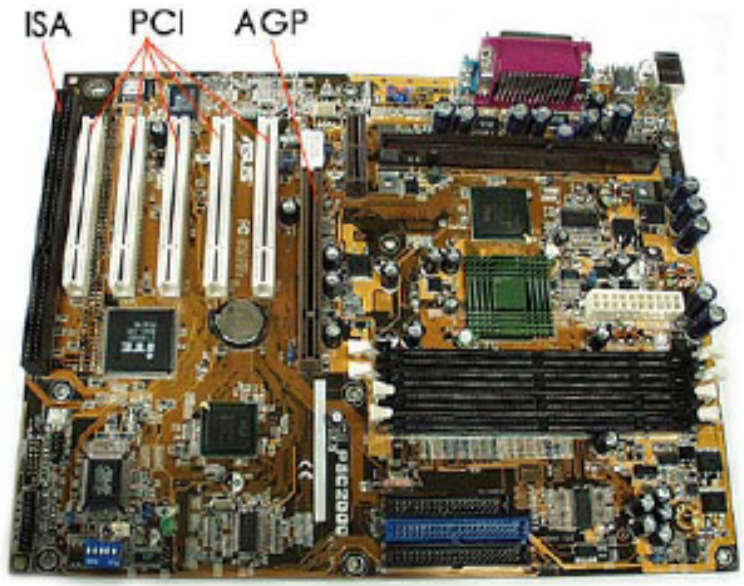
- Chip controllore di interruzioni Intel 8259A usato dal PC IBM e successori
- Gestisce 8 linee di interrupt
  - INT: interruzione inviata alla CPU
  - $\overline{\text{INTA}}$: *acknowledge* della CPU
  - numero del dispositivo che indirizza un *vettore di interrupt*
  - IR0-IR7: linee di interrupt sul Bus
- Il vettore di interrupt è usato dalla CPU per invocare la relativa routine
- Registri all'interno del chip scrivibili dalla CPU per programmare il controller

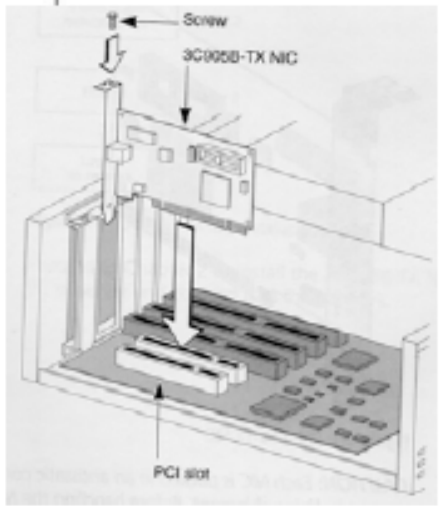
Il Bus PCI: architettura complessiva



Schede e slot PCI

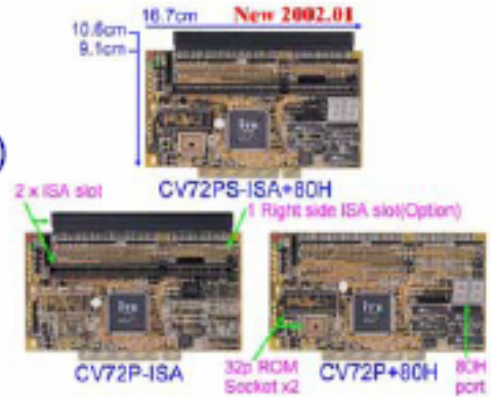
Schede e slot PCI



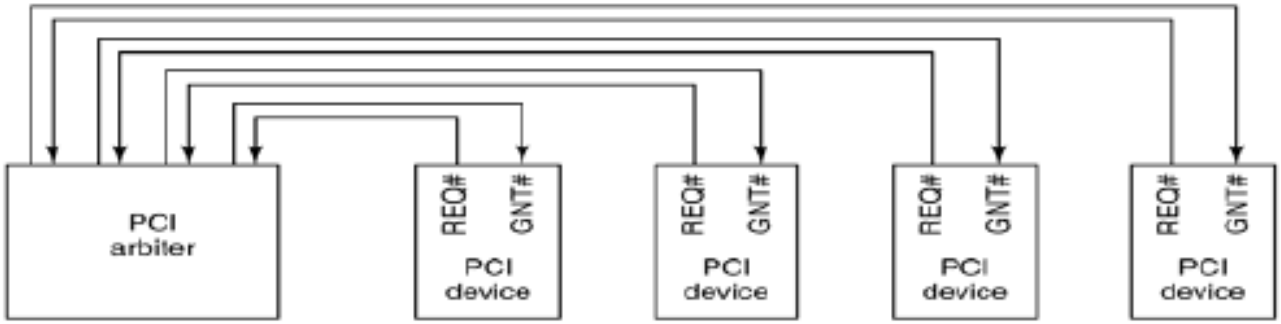


Il Bus PCI: specifiche

- **PCI** (**P**eripheral **C**omponent **I**nterconnect)
- Introdotto da Intel per applicazioni video
- Video 1024×768×3 Byte a 30 frame/sec richiede una banda di 67.5 MB/sec
- Standard *non proprietario*, adottato da molti, (ma Intel vende i chip di gestione…)
- Versione base a 32 bit, 33 MHz: 133 MB/sec
- Estensione a 64 bit e 66 MHz: 528 MB/sec
- *Local Bus* e *Memory Bus* separati (più veloci)
- Connessione tramite chip **PCI bridge**
- Varie opzioni di tensione (5 V e 3.3 V)
- Schede con 120 e 120+64 contatti
- **Bridge ISA** (include doppio controller IDE)
- Controllori addizionali SCSI e USB
- Bus sincrono, transazioni tra **initiator** e **target**
- Linee indirizzo e dati multiplexate

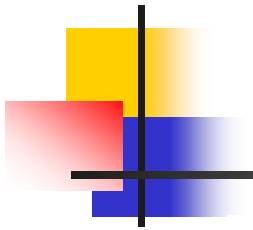


Il Bus PCI: arbitraggio



- Arbitraggio centralizzato (nel Bridge)
- Ogni PCI device ha due linee dedicate
- Il dispositivo fa la richiesta tramite REQ#
- Il grant viene concesso tramite GNT#
- Diversi algoritmi di arbitraggio:
  - Round Robin
  - Priorità
  - Altro
- Transazioni su più cicli separate da cicli di idle

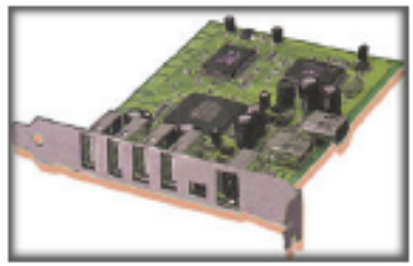
Bus USB (Universal Serial BUS)



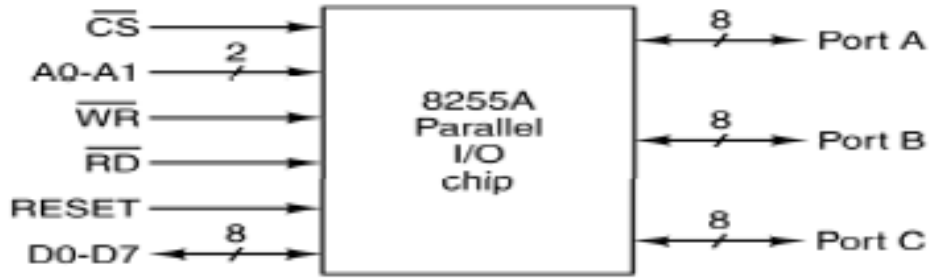
- Bus economico concordato da varie aziende per la gestione di dispositivi di I/O a bassa velocità (~ 1995)
- Obiettivi:
  1) Evitare *switch*, *jumpers*
  2) Installazione di tipo esterno
  3) Cavo di connessione unificato
  4) Alimentazione fornita dal cavo
  5) Fino a 127 dispositivi collegabili
  6) Supporto di dispositivi real-time
  7) Installazione a PC acceso
  8) Reboot non necessario
  9) Bus e dispositivi economici
- Tutti gli obiettivi sono di fatto rispettati

USB: specifiche fondamentali

- Banda complessiva 1.5 MB/sec
- Limitata per ragioni di costo
- *Root hub* di connessione al bus PCI
- Connessione di dispositivi e di altri hub
- Struttura complessiva ad albero
- Connettori ai capi del cavo diversi
- Cavo a 4 fili: +5V, GND, 2 di segnale
- Alla connessione di un dispositivo:
  - Interrupt: intervento del SO
  - Richiesta di banda
  - Assegnazione di indirizzo
- Indirizzo 0 usato per inizializzazione
- Logicamente connessione tra root hub e ciascun device con bit pipe dedicata

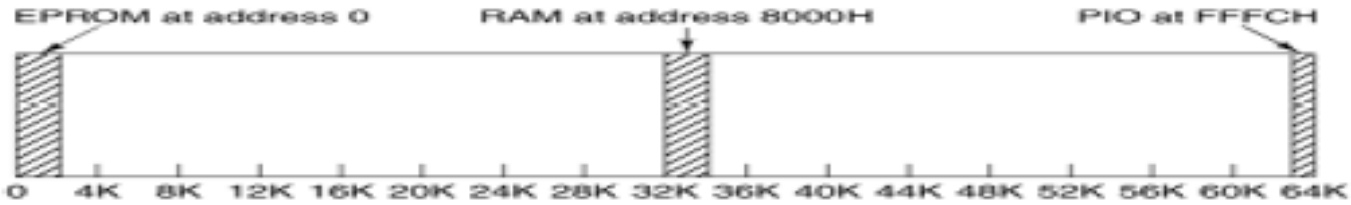


Chip I/O: UART, USART e PIO



- UART (Univ. Async. Rec. Transm.)
- USART (... Sync. Async. .....)
- Usati in interfacce parallelo/seriale
- PIO (Parallel Input/Output)
  - Configurabile dalla CPU
  - 3 Porte indipendenti da 8 bit con latch
  - La CPU legge e scrive direttamente nelle porte
  - Possibile gestire anche semplici protocolli di handshaking CPU-device
  - Per gestire dispositivi *TTL*-compatibili

Decodifica degli indirizzi

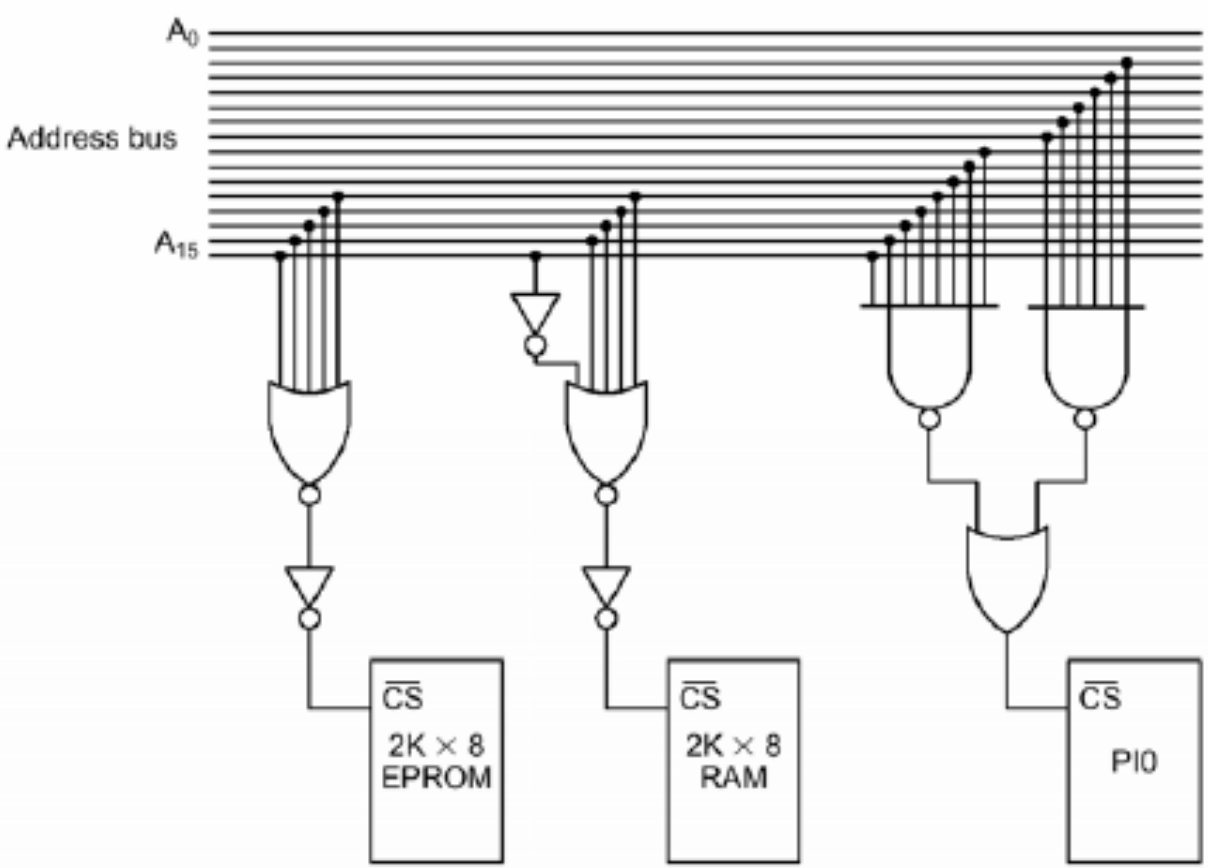


- Spazio separato di I/O, oppure unico spazio (Memory-Mapped I/O)
- Alcuni indirizzi riservati alle porte di I/O
- Problema: ricavare i segnali di chip select

ES

EPROM: 0-2K	0000 0XXX XXXX XXXX
	0000 0XXX XXXX XXXX
RAM: 32K-34K	1000 0XXX XXXX XXXX
	1000 0XXX XXXX XXXX
PIO: FFFC-FFFF	1111 1111 1111 1100
	1111 1111 1111 1111

Codifica completa



Decodifica parziale