



Bus di Sistema



Indice

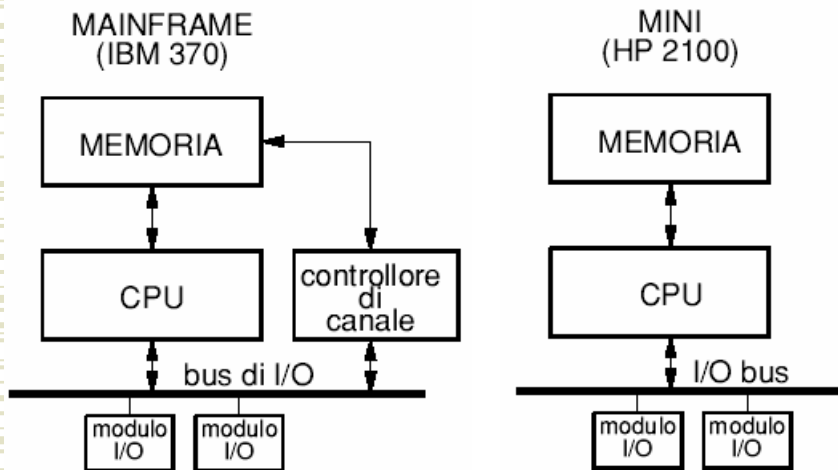


- ◆ Introduzione
- ◆ I Bus di Sistema
 - Architettura
 - Principi di Funzionamento
- ◆ IL BUS VME

Cosa Sono i Bus di sistema?

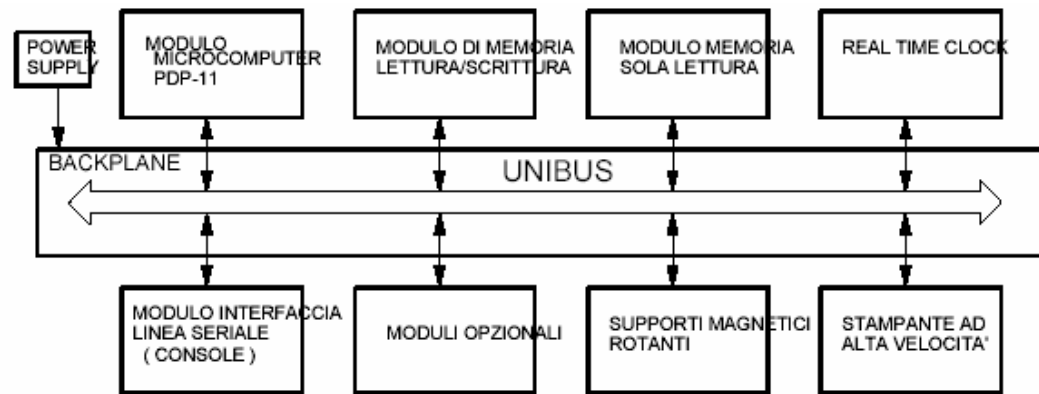
- Un bus è un comune collegamento elettrico tra molti dispositivi.
- Il bus di sistema, presente in tutti i microcalcolatori, è composto da 50 a 100 fili in rame incisi sulla scheda madre ed è dotato di connettori separati ad intervalli regolari per l'innesto dei moduli di memoria e di I/O
- Un bus che collega 2 componenti appartenenti alla stessa scheda integrata è detto INTERNAL BUS (di solito proprietario), se collega due componenti generici è detto EXTERNAL BUS. Se c'è 1 solo bus esterno è detto System bus.

La nascita del Bus di Sistema: UNIBUS



Le Architetture dei
primi sistemi IBM
370, HP 2100

L'UNIBUS



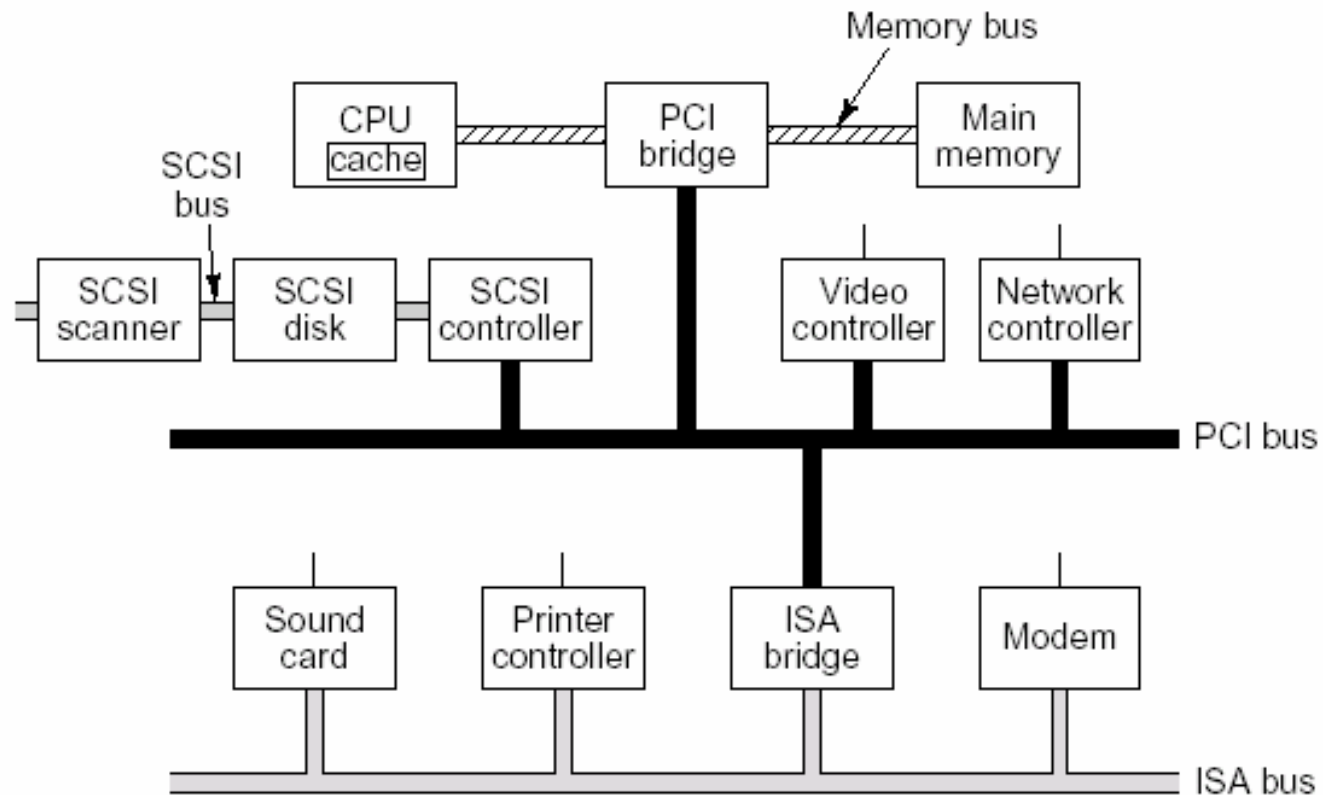
Evoluzione dei BUS (I)

- ◆ IBM74 (Anni '60)
 - Comunicazioni indipendenti
- ◆ UNIBUS (Anni '60-'70)
 - Introduzione del Backplane, un unico canale
- ◆ BUS industriali e VME(dagli anni '80)
 - Massimo utilizzo del concetto di bus di sistema, modularità portata all'estremo, lo standard VME

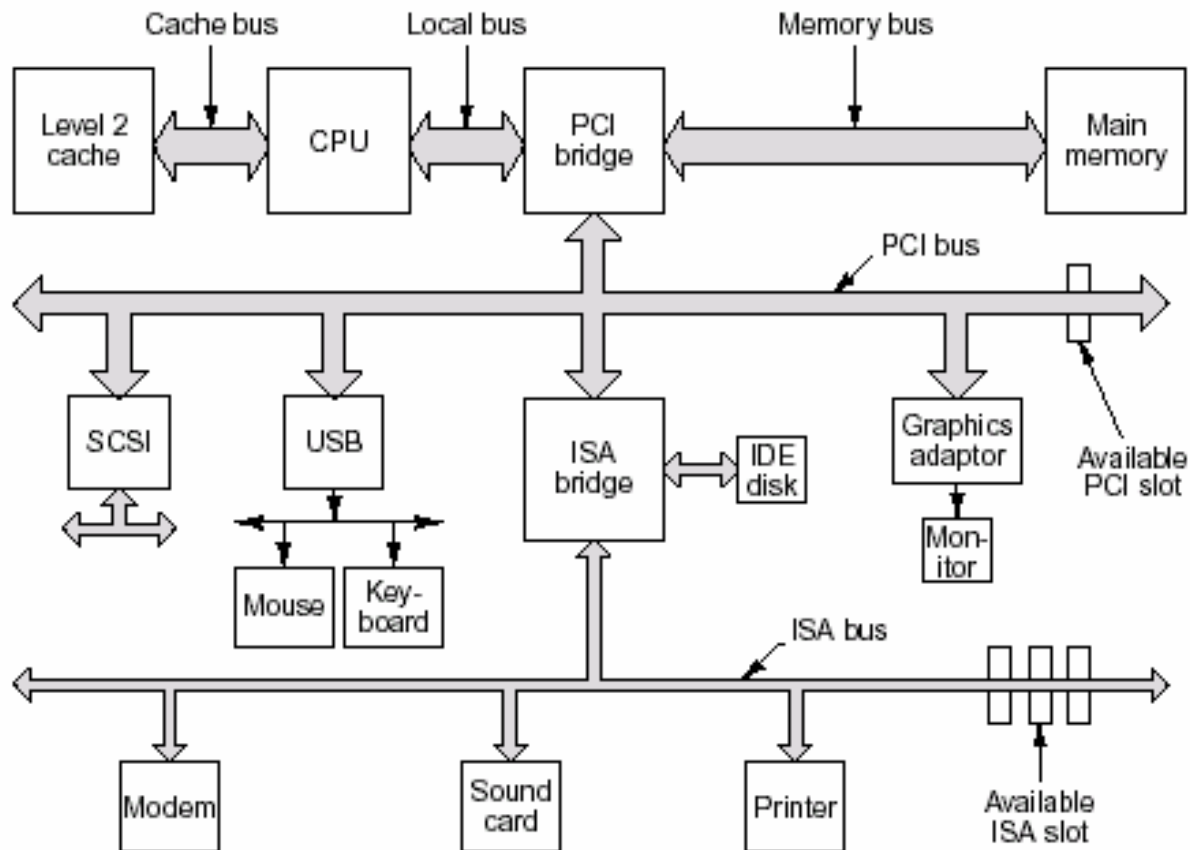
Evoluzione dei BUS (II)

- ◆ I PC: dal PC Bus a PCI (dagli anni '80)
 - Molteplici Bus, sia per la compatibilità con il passato (ISA), che per la gestione dedicata di parti del sistema (AGP), il PCI come standard de facto
- ◆ Reti Industriali
 - I moduli diventano nodi di una rete, i concetti di bus di sistema e quello di rete si cominciano a confondere, lo standard Profibus

Architettura di un bus di un PC moderno



Architettura di un bus di un Pentium II



Gli Standard

Esempi di standard:

- ◆ IEEE 796 (Multibus I)
- ◆ IEEE 896 (Futurebus)
- ◆ IEEE 1014 (VME)
- ◆ IEC 61158 (Profibus, Foundation Fieldbus)

Devono definire

- ◆ Caratteristiche Meccaniche,
- ◆ Caratteristiche Elettriche,
- ◆ Caratteristiche Logico/Funzionali

Sintesi di alcune Caratteristiche Meccaniche ed Elettriche

Meccaniche

- ◆ Inserzione
 - Diretta
 - Indiretta
- ◆ Dimensione
 - Standardizzazione delle Unità

Elettriche

- ◆ Logica Positiva o Negativa
- ◆ Tempi di ritardo
- ◆ Tempi di salita/discesa

Caratteristiche Logico Funzionali

Struttura

- ◆ Dimensione del Bus
- ◆ Numero e tipo dei segnali
- ◆ Modalità
 - Sincrona
 - Asincrona
- ◆ Meccanismo di allocazione del Bus
- ◆ Gestione delle Interruzioni

Protocollo

- ◆ Insieme di regole per lo scambio dei segnali
- ◆ Descrizione della Tempificazione di ciascuno dei segnali coinvolti

Caratteristiche Logico Funzionali di alcuni BUS

ISA bus (IBM):

- ◆ 62 linee di segnali (controllo, dati 8 ed indirizzi 20),
- ◆ 8.33 Mhz,
- ◆ E' compatibile con il precedente PC/AT (formato particolare dei connettori per gestire più linee).

EISA 32 bits (4 byte per ciclo = 33.3 MB/sec)

PCI bus (Intel 1990):

32 bit per ciclo a 33Mhz
(max bandwidth
133MB/sec)



I BUS di Sistema

Principi di Funzionamento

I ruoli dei Dispositivi

- ◆ **Dispositivi Attivi (Master)**
 - Sono in grado di avviare dei trasferimenti sul bus
- ◆ **Dispositivi Passivi (Slave)**
 - Sono in attesa, aspettando richieste di trasferimento
- ◆ Il ruolo di un dispositivo può cambiare nel tempo;
- ◆ un dispositivo può comportarsi da master o da slave in contesti differenti.
- ◆ Lo standard che definisce il bus deve fornire le regole per gestire tali condizioni o vietarle

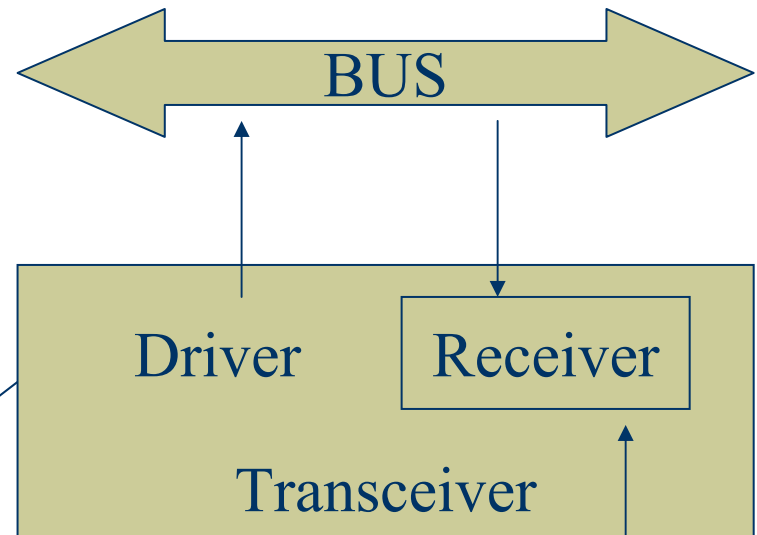
Cenni a Problemi Elettrici

- *Driver del Bus* -

I segnali binari emessi dai dispositivi non sono in grado di alimentare il bus, il Driver del bus è essenzialmente un amplificatore digitale

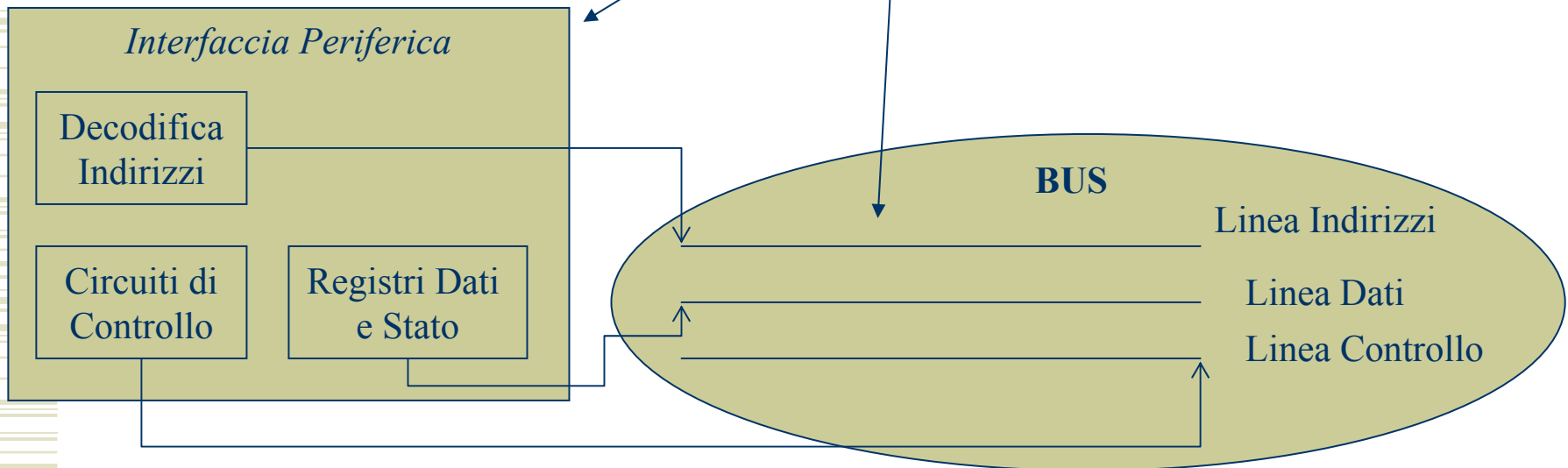
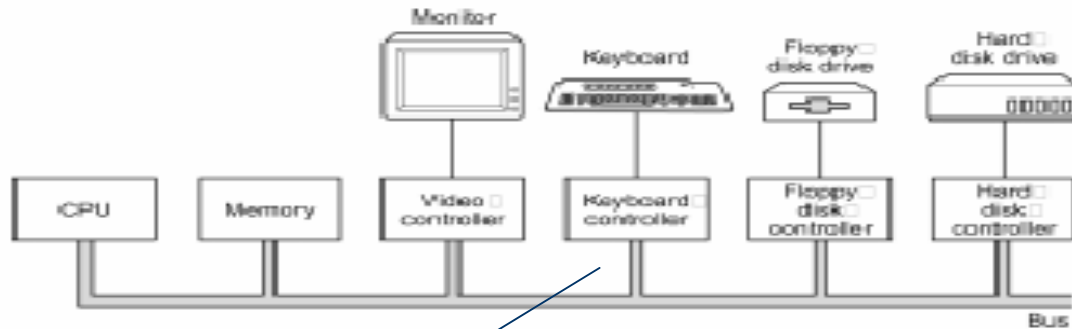
Ruolo Master

Il Dispositivo è sia Master che Slave



Ruolo Slave

Le linee del BUS



Protocolli di Comunicazione

Un Protocollo è quell'insieme di regole che gestiscono la comunicazione tra due entità. Esistono due diversi meccanismi di temporizzazione dei segnali

Protocollo Sincrono

E' previsto un segnale di sincronizzazione (clock) che permette di gestire la temporizzazione delle comunicazioni

Protocollo Asincrono

Tutta la temporizzazione della comunicazione è gestita dal protocollo stesso attraverso lo scambio dei messaggi.

Protocollo Sincrono

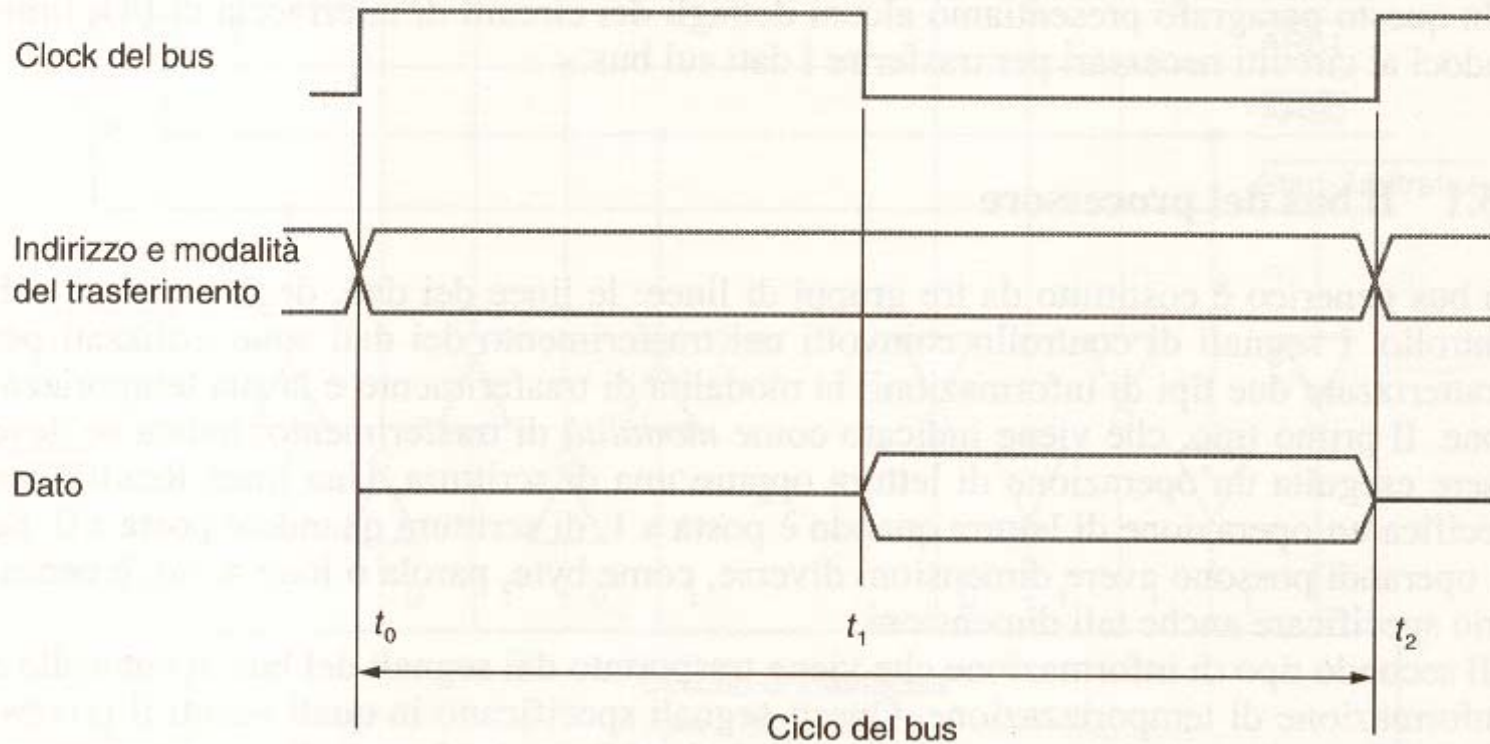


Figura 4.19 Temporizzazione di un trasferimento dei dati in ingresso al processore su un bus sincrono.

Protocollo Sincrono (2)

- ◆ All'istante t_0 la CPU pone gli indirizzi sul bus ed attiva la funzione di read;
- ◆ Nell'intervallo $t_1 - t_0$ tutti i dispositivi hanno letto e decodificato i segnali di indirizzamento e controllo;
- ◆ All'istante t_1 il dispositivo indirizzato è pronto a rispondere e mette i dati sul bus.

Protocollo Sincrono (3)

- ◆ All'istante t_2 la CPU guarda la linea dati e ne carica il contenuto nel MDR;
- ◆ $t_2 - t_1$ serve per propagare il dato e “tenerlo” in ingresso al registro MDR per un tempo sufficiente al suo caricamento. La freq. Del clock deve essere scelta in modo da considerare i ritardi più lunghi sul bus.
- ◆ **OSSERVAZIONE:** il processore non sa quando il dispositivo risponde realmente, esso assume semplicemente che all'istante t_2 il dato è disponibile; se si verifica un errore **NON** verrà rilevato.

Protocollo Asincrono (Ingresso)

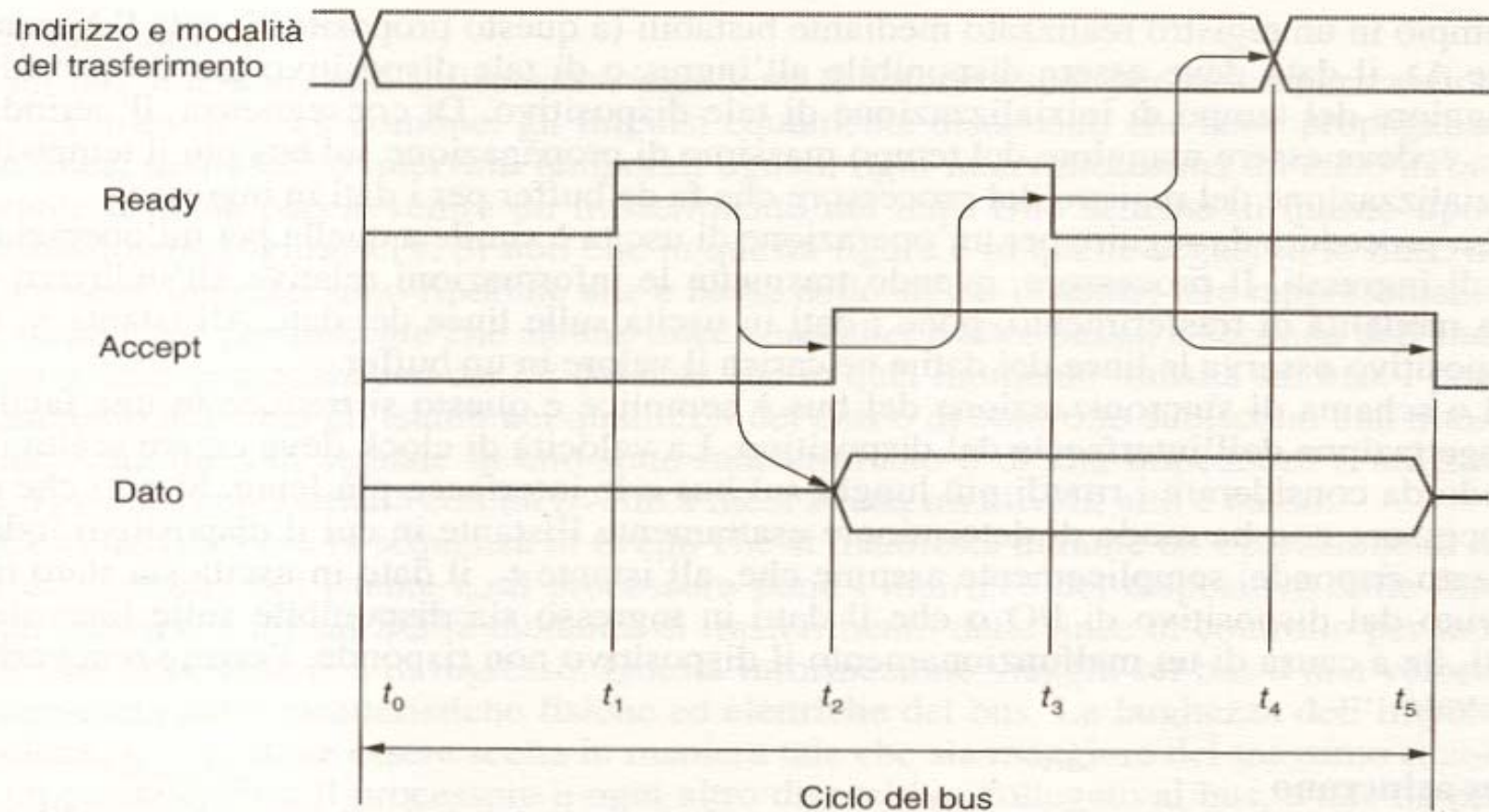


Figura 4.20 Controllo handshake del trasferimento di un dato durante un'operazione di ingresso al processore.

Protocollo Asincrono (Uscita)

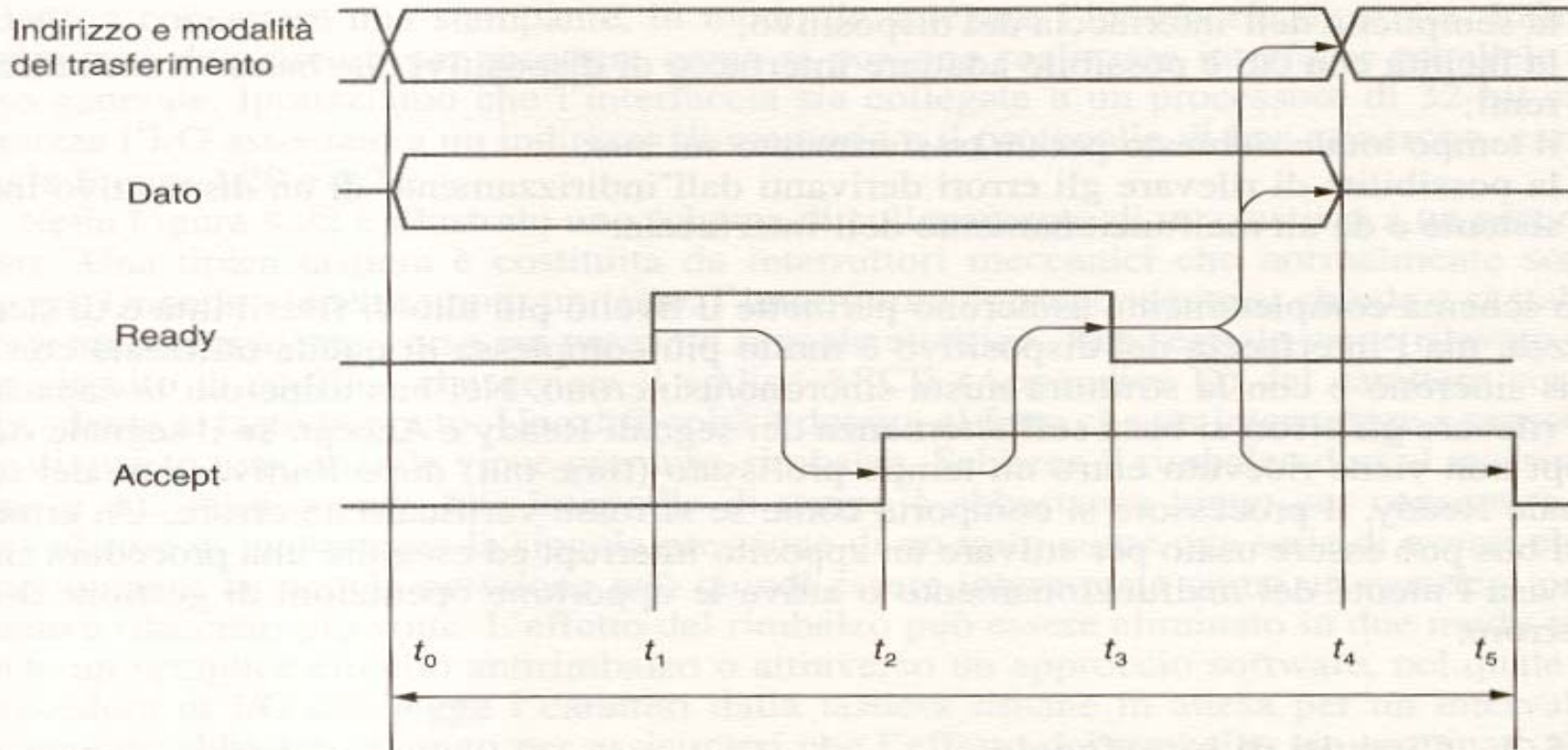


Figura 4.21 Controllo handshake del trasferimento di un dato durante un'operazione di uscita dal processore.

Protocollo Asincrono

- ◆ $t_1 - t_0$ annulla i possibili bus skew (ritardi differenti sulle varie linee),
- ◆ $t_2 - t_1$ dipende dalla distanza tra CPU e periferica ed è funzione dei ritardi (è questa variabilità che rende preferibile il protocollo asincrono)

Sincrono vs. Asincrono

- ◆ La scelta dipende da:
 - Interfaccia del dispositivo,
 - Adattare interfacce con differenti velocità,
 - Tempo totale richiesto per il trasferimento,
 - Possibilità di rilevare errori.
 - Uno schema completamente asincrono è affidabile e flessibile ma le interfacce e la logica di controllo sono molto più complicate da realizzare.

Esempio di bus operation da approfondire

- ◆ Paragrafo 4.xx di:
- ◆ MC68000 16-bit microprocessor User's manual
- ◆ Vedere esempio di word e byte read flow chart e timing diagram;
- ◆ Vedere timing di Read-modify-Write cycle (usato dall'istruzione TEST and SET del MC68000)

Arbitraggio del Bus

Il Problema

- ◆ Bus MultiMaster:
 - E' ammesso più di un dispositivo Master sul BUS
- ◆ Arbitraggio:
 - Gestione delle contese tra i vari Master

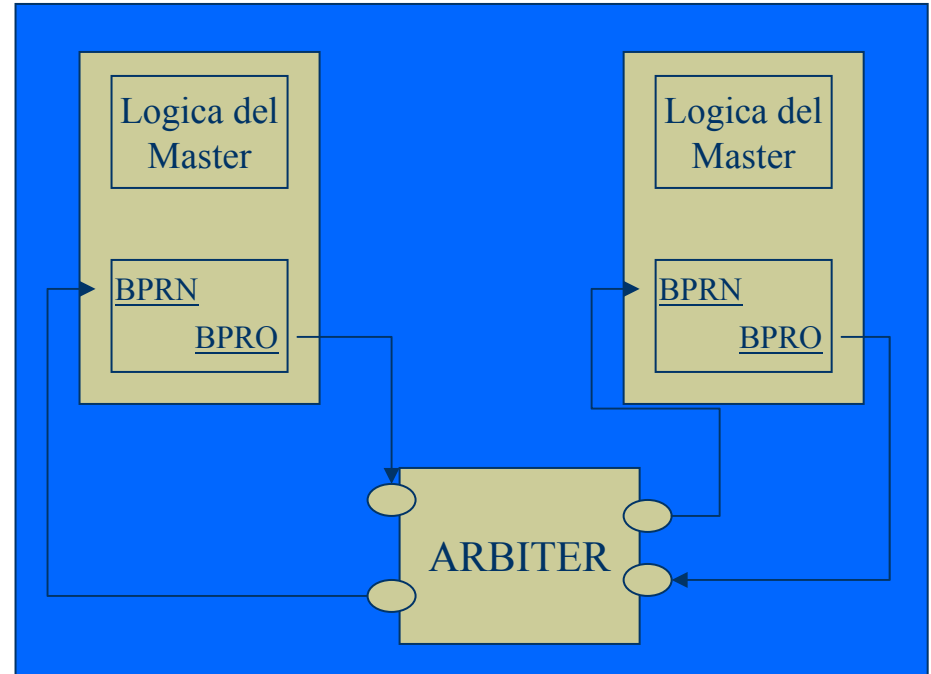
Le Soluzioni

- ◆ Centralizzato
- ◆ Distribuito

Centralizzato (Multibus I)

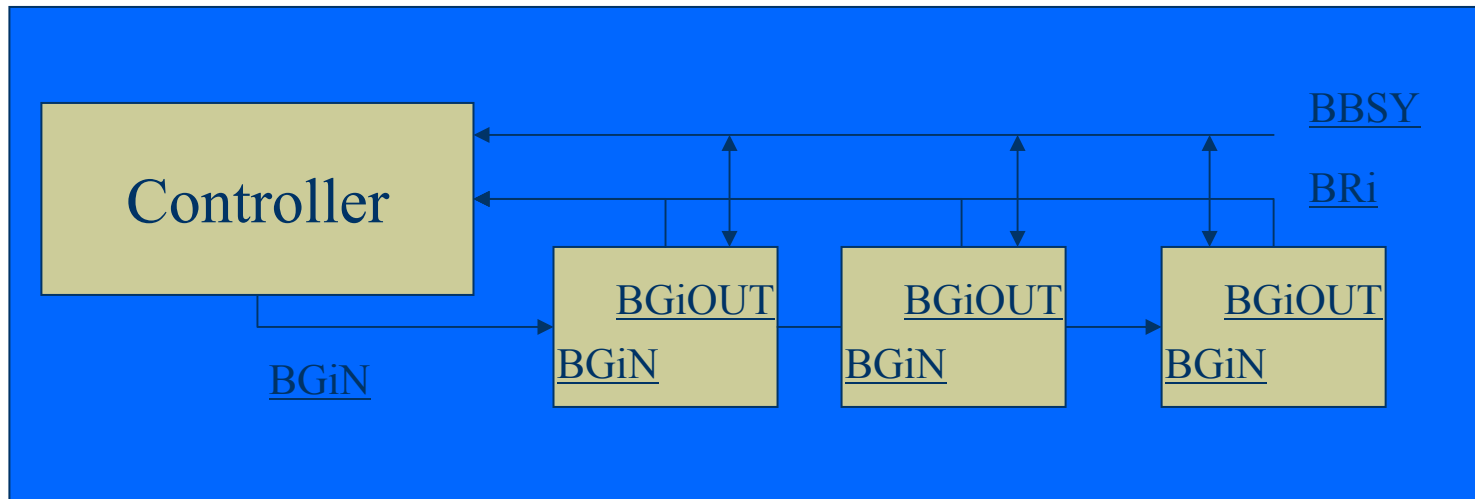
- ◆ Tutte le linee di richiesta vanno all'arbitro
- ◆ Tutte le linee di ack vanno all'arbitro
- ◆ L'arbitro definisce le priorità

Nota: adottato anche in EISA e PCI



Centralizzato Daisy Chain (VME)

- ◆ Il Controllore gestisce diverse liste
- ◆ La Linea BBSY è comune a tutte e permette di sincronizzare le richieste
- ◆ Nessun modulo può fare richiesta con BBSY asserita



Esempio di Daisy Chain con i DMA

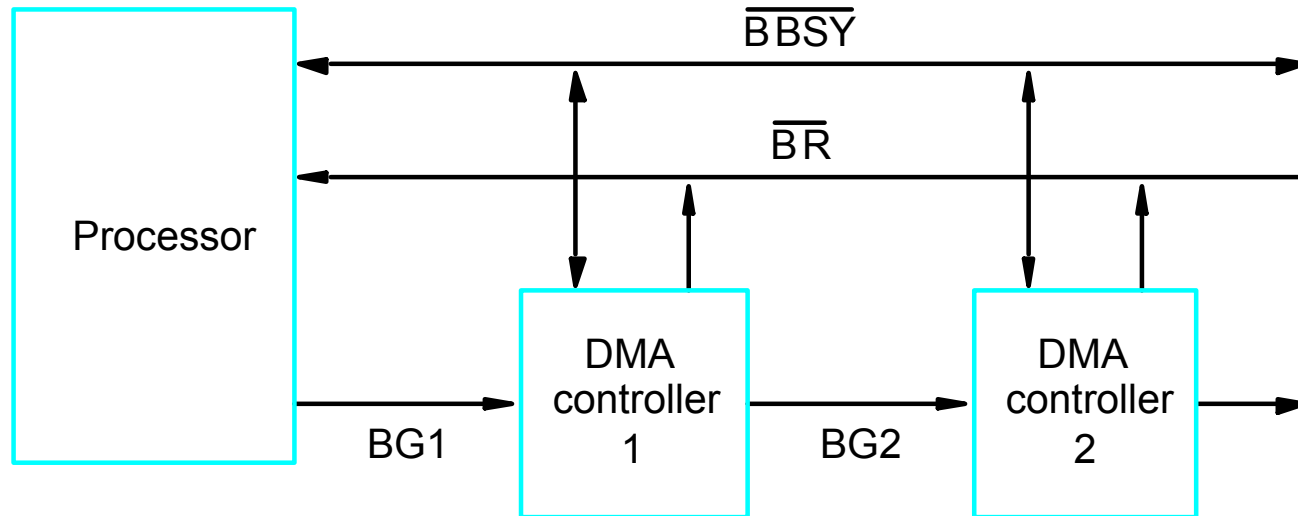


Figure 4.20. A simple arrangement for arbitration using a daisy chain

Esempio di Daisy Chain: temporizzazione

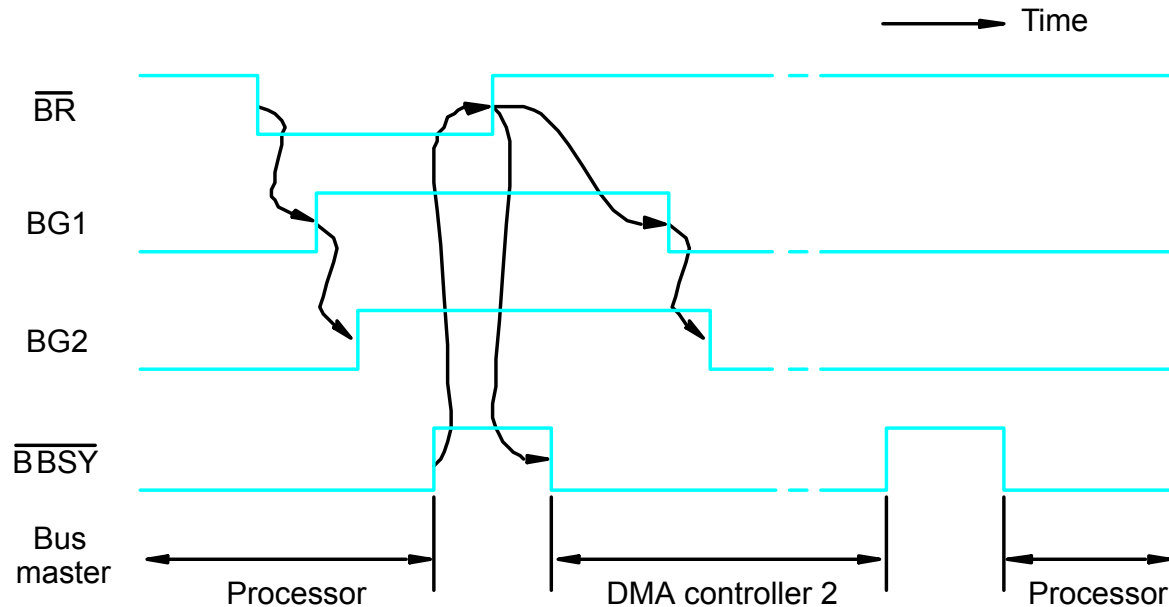


Figure 4.21. Sequence of signals during transfer of bus mastership for the devices in Figure 4.20.

Arbitraggio Distribuito: esempio

- ◆ Ogni dispositivo sul bus è identificato da un ind. A 4 bit;
- ◆ Quando un dispositivo vuole usare il bus attiva il segnale Start Arbitration ponendo il proprio id sulle linee ARBi di tipo Open Collector (wired OR).
- ◆ Se l'id 5 e 6 richiedono il bus (0101 or 0110 = 0111) sulle linee sarà presente l'id 0111.
- ◆ Ogni dispositivo confronta il proprio id con queste linee, il primo che si accorge di non essere stato indirizzato disabilita le linee ponendo uno 0.

Arbitraggio Distribuito: schema

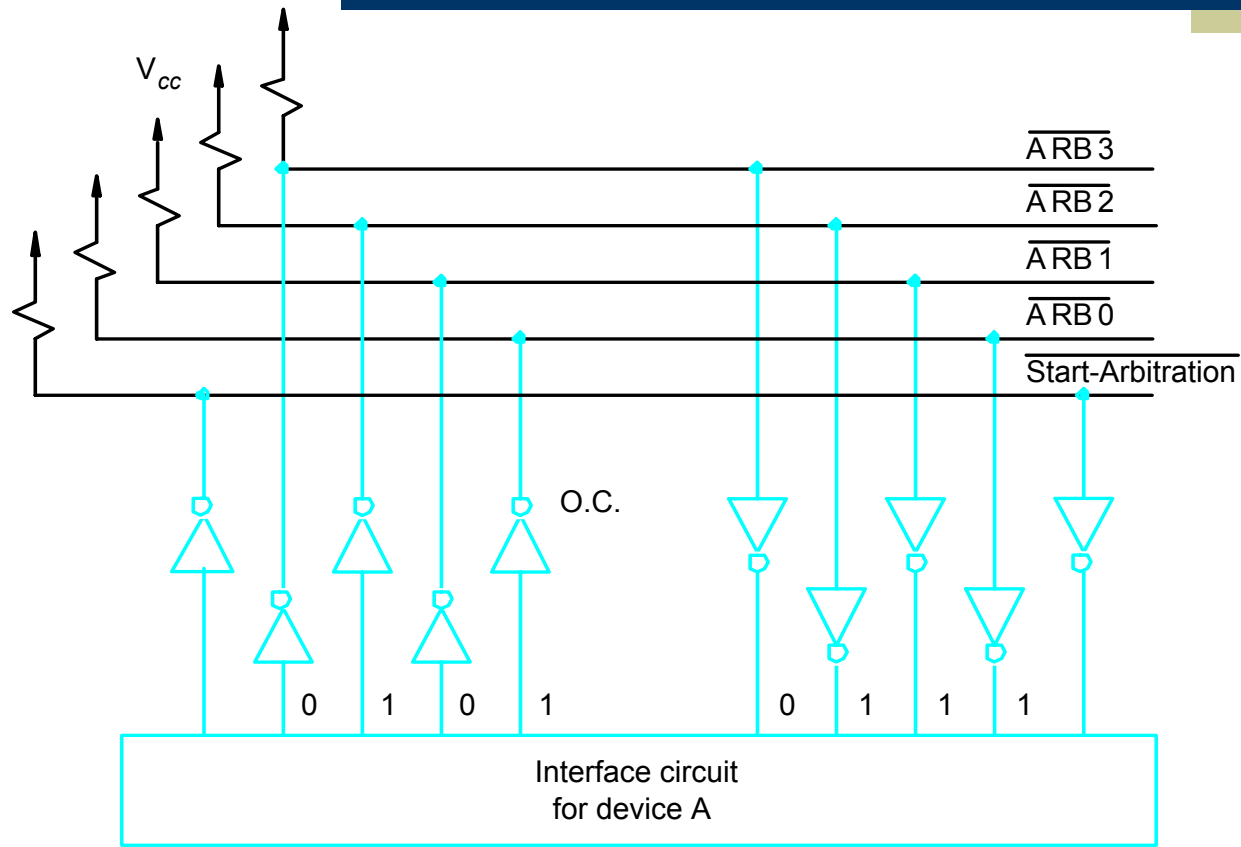


Figure 4.22. A distributed arbitration scheme.

Riferimenti

- ◆ Tanenbaum:
 - CAP 2 e 3;
- ◆ Hamacher:
 - CAP 4;
- ◆ User's guide MC68000:
 - CAP 4.

I BUS di Sistema

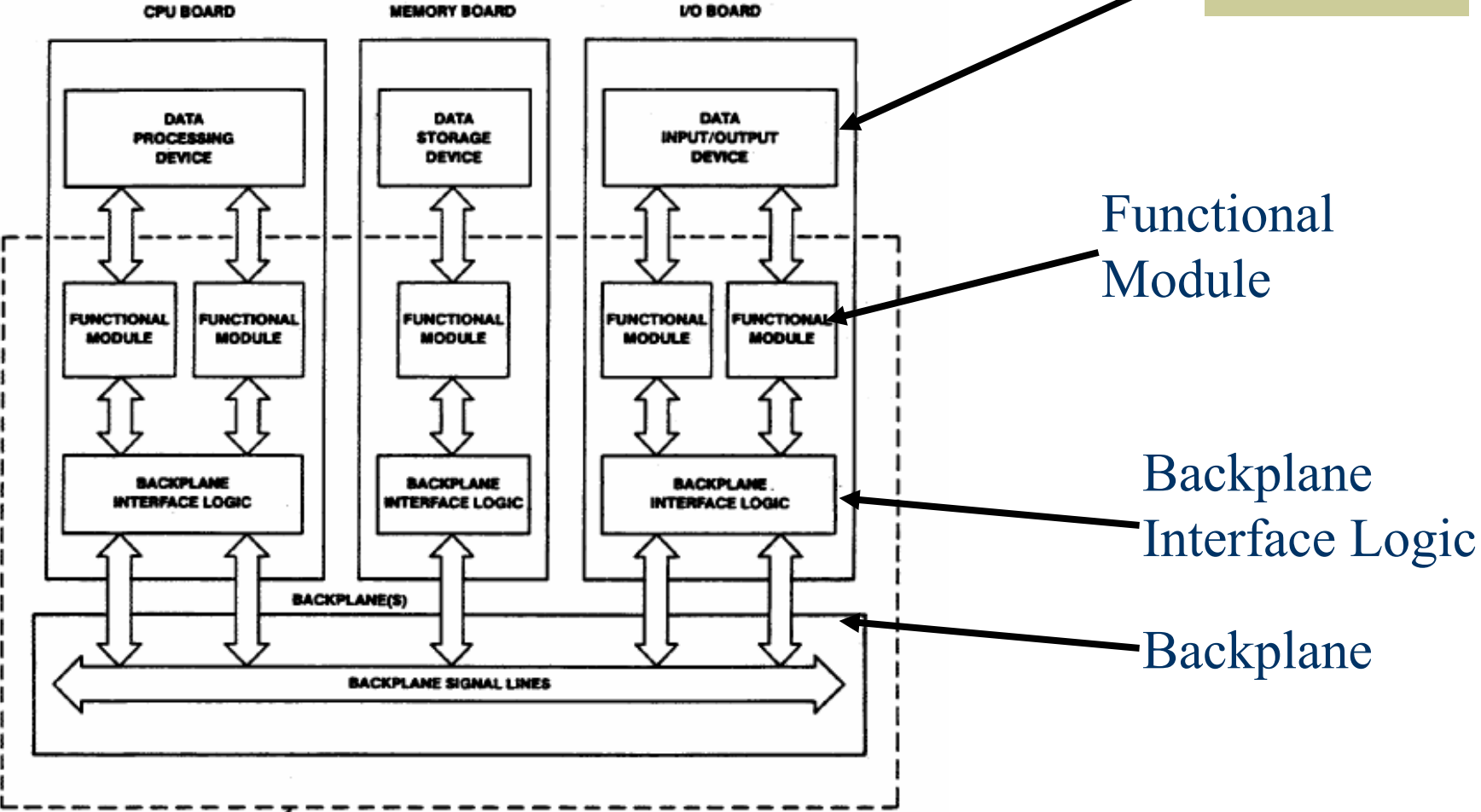
L'esempio del VME

Lo Standard VME

- ◆ Standard IEEE 1014
- ◆ Nato nel 1987, è il più diffuso dei BUS industriali
- ◆ Lo standard è considerato un riferimento generale per la sua chiarezza

Gli elementi definiti dallo standard

Devices



Functional Module

Backplane Interface Logic

Backplane

Definizioni (1)

- ◆ **backplane:** A printed-circuit board (pcb) with 96-pin connectors and signal paths that connect corresponding connector pins. Some systems have a single pcb, J1 backplane. It provides the signal paths needed for basic operation.
- ◆ **slot:** A position where a printed-circuit board (pcb) can be inserted into the backplane.
- ◆ **backplane interface logic:** Special logic that takes into account the characteristics of the backplane; its signal line impedance, propagation time, termination values, etc.

Definizioni (2)

- ◆ **interrupter:** A functional module that generates an interrupt request on the priority interrupt bus, and then provides status/ID information when the interrupt handler requests it.
- ◆ **interrupt handler:** A functional module that detects interrupt requests generated by interrupters and responds to those requests by asking for status/ID information.
- ◆ **master:** A functional module that initiates data transfer bus (DTB) cycles to transfer data between itself and a slave module.
- ◆ **slave:** A functional module that detects data transfer bus (DTB) cycles initiated by a master and, when those

Definizioni (3)

- ◆ **priority interrupt bus:** One of the four buses provided by the backplane. The priority interrupt bus allows interrupter modules to send interrupt requests to interrupt-handler modules.
- ◆ **utility bus:** This bus includes signals that provide periodic timing and coordinate the power-up and power-down of the systems. It is one of the four buses provided by the backplane.
- ◆ **data transfer bus (DTB):** One of the four buses provided by the backplane. The data transfer bus allows masters to direct the transfer of binary data among themselves and slaves.
- ◆ **arbitration bus:** One of the four buses provided by the backplane. This bus allows an arbiter module and several requester modules to coordinate use of the DTB.

Tipi di ciclo

- ◆ **read (write) cycle:** A data transfer bus (DTB) cycle that is used to transfer 1, 2, 3, or 4 bytes from a slave to a master (from a master to a slave).
- ◆ **block read (write) cycle:** A data transfer bus (DTB) cycle that is used to transfer a block of bytes ranging in number from 1 to 256 bytes from a slave to a master (from a master to a slave).
- ◆ **read-modify-write cycle:** A data transfer bus (DTB) cycle that is used to both read from, and write to, a slave location without permitting any other master to access that location.
- ◆ **address-only cycle:** A data transfer bus (DTB) cycle that consists of an address broadcast, but does not have a data transfer.
- ◆ **interrupt acknowledge cycle:** A data transfer bus (DTB)



Le Linee del Bus VME

- ◆ Trasferimento dati (DTB)
- ◆ Arbitraggio del bus
- ◆ Bus per la Priorità degli Interrupt
- ◆ Bus di Utilità

Data Transfer Bus Lines

- ◆ Gestisce esplicitamente anche sistemi piccoli
 - Trasferimenti da 8,16,32 bit
 - Indirizzi a 16,24,32 bit
- ◆ 33 diversi tipi di cicli di BUS
 - Trasferimenti a blocchi (limitati a 256 byte)
 - ciclo indivisibile lettura/modifica/scrittura
 - ciclo solo indirizzo

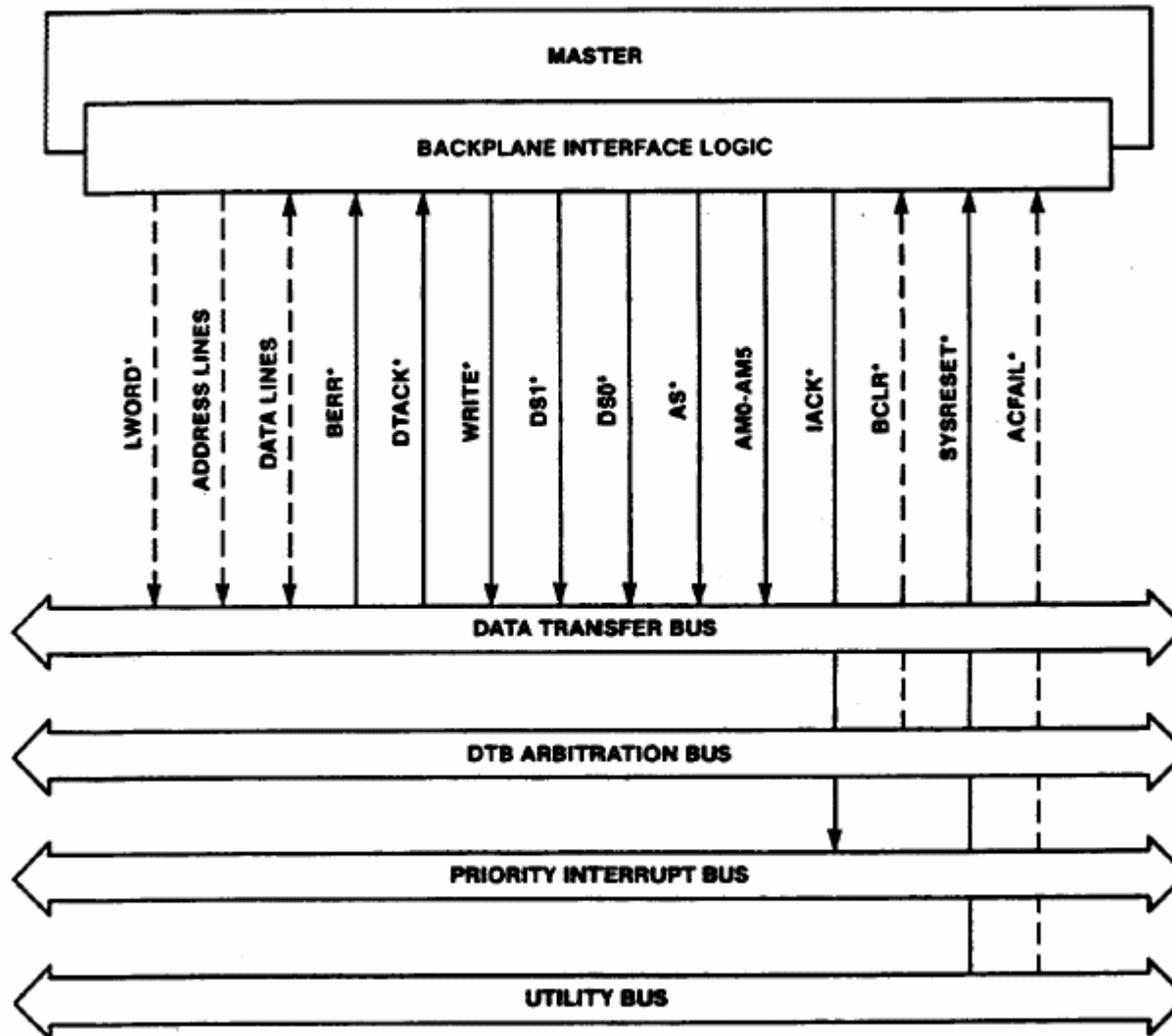
Data Transfer Bus Lines

- ◆ 4 tipi di dispositivi:
 - Master e Slave
 - Controllore di locazione
 - (genera interrupt su certi indirizzi)
 - Timer del Bus

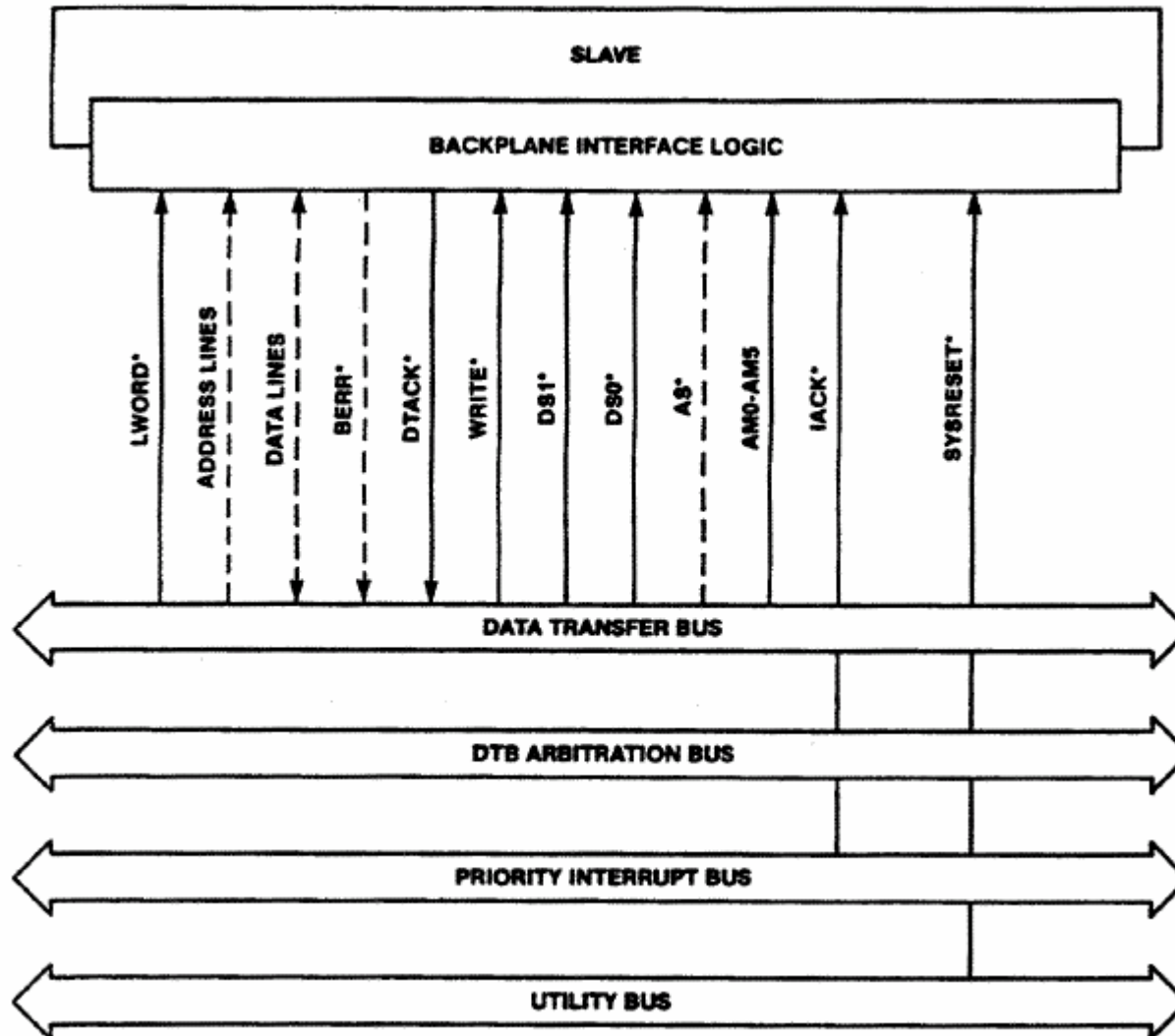
Data Transfer Bus Lines (2)

Addressing Lines	Data Lines	Control Lines
A01–A31	D00–D31	AS*
AM0–AM5		DS0*
DS0*		DS1*
DS1*		BERR*
LWORD*		DTACK*
		WRITE*

II Master

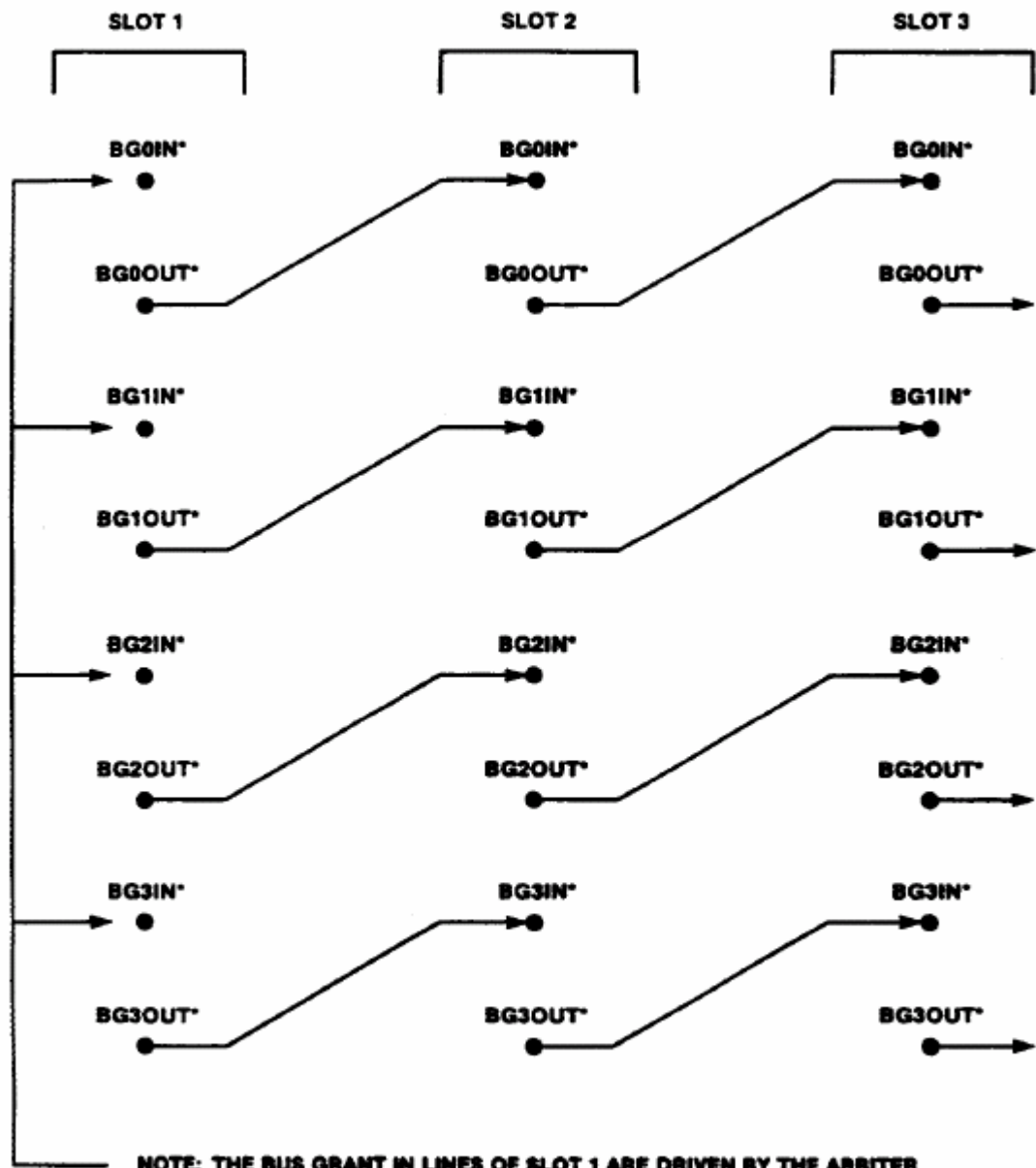


Lo Slave



Arbitraggio del Bus

- ◆ Linee
 - 6 linee bus e 4 linee per il canale Daisy-Chain
- ◆ Tre tipi di arbitraggio
 - A priorità
 - RoundRobin
 - Singolo livello

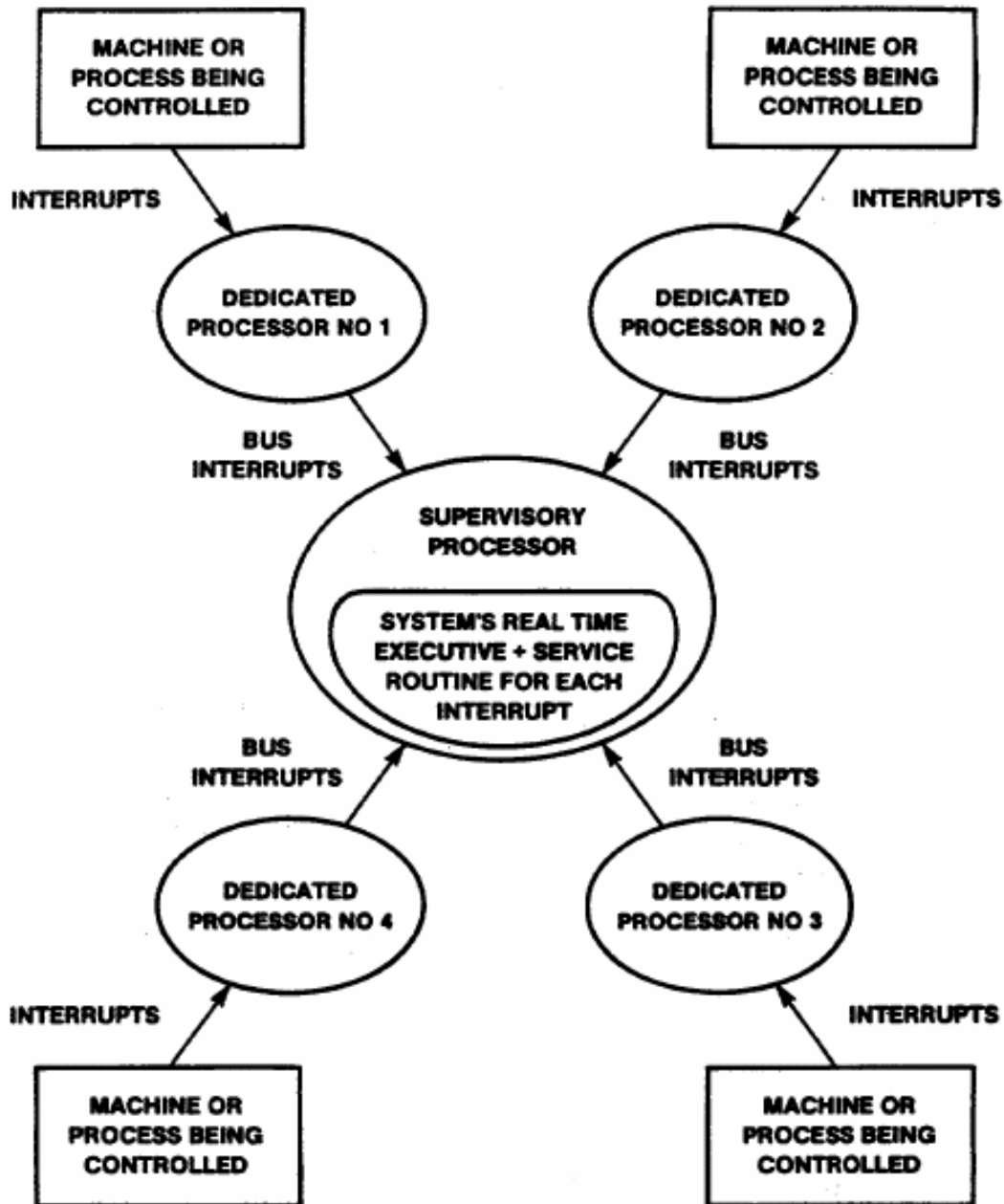


NOTE: THE BUS GRANT IN LINES OF SLOT 1 ARE DRIVEN BY THE ARBITER WHICH IS ALWAYS LOCATED ON THE BOARD PLUGGED INTO SLOT 1.

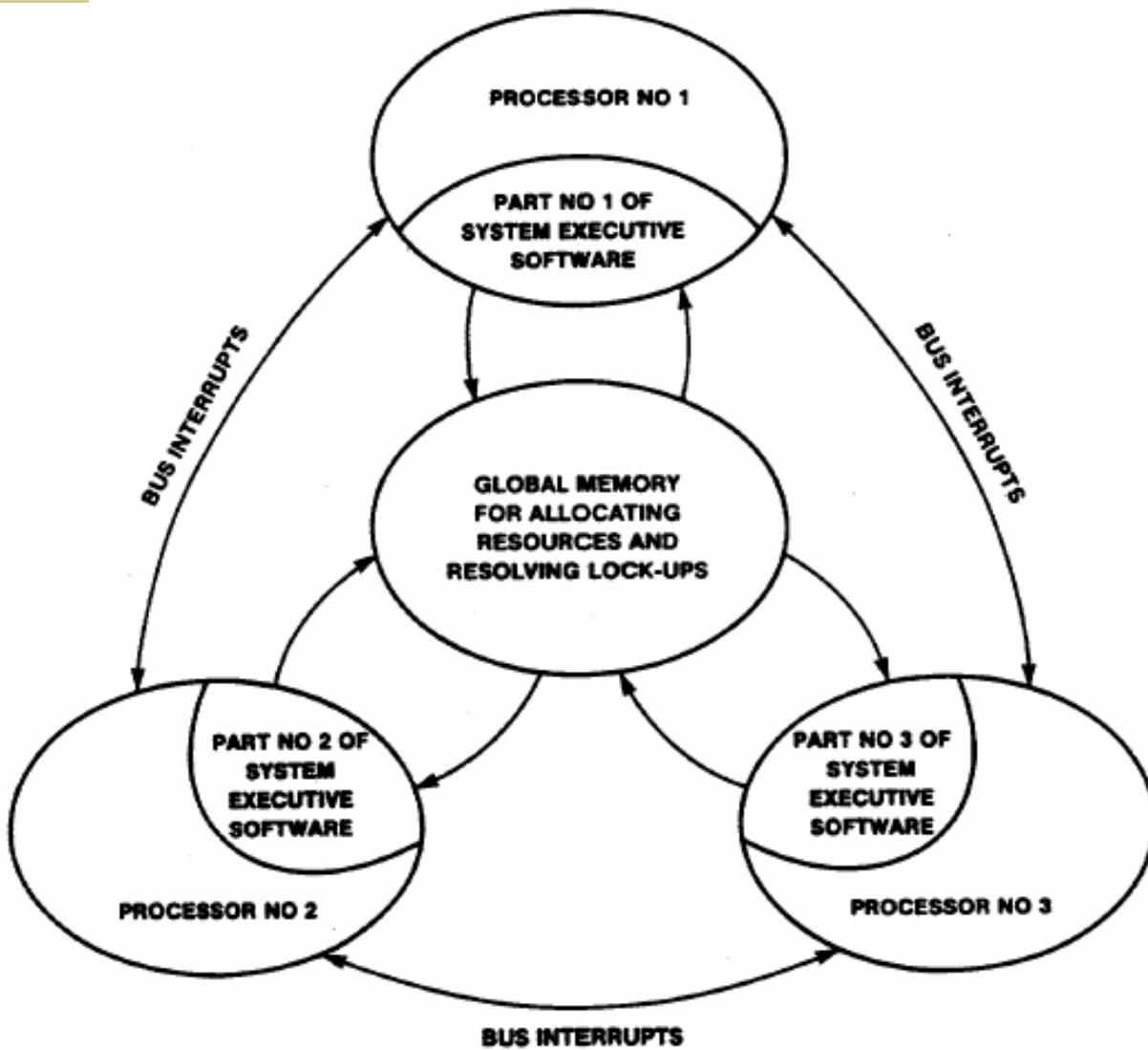


Priority Interrupt Bus lines

- ◆ Due classi di sottosistemi per le priorità
 - Sistema a singola gestione (singolo interrupt handler)
 - Sistema a gestione distribuita (2 o più interrupt handler)



Single Interrupt Handler



Distributed Interrupt Handler

Linee di utilità

- ◆ Segnale di clock a 16 Mhz
 - NON viene usata per i cicli di bus, ma per le misurazioni del tempo
- ◆ 2 linee per il Bus seriale VMS
- ◆ Linee di inizializzazione, ripristino e controllo



Riferimenti



- ◆ Standard IEEE 1014