



Le interconnessioni tra i dispositivi

Prof. Alberto Borghese
Dipartimento di Informatica
alberto.borghese@unimi.it

Università degli Studi di Milano

Riferimento Patterson: 3.7 per sub-word parallelism, Patterson vers. 4 inglese per le interconnessioni



Sommario

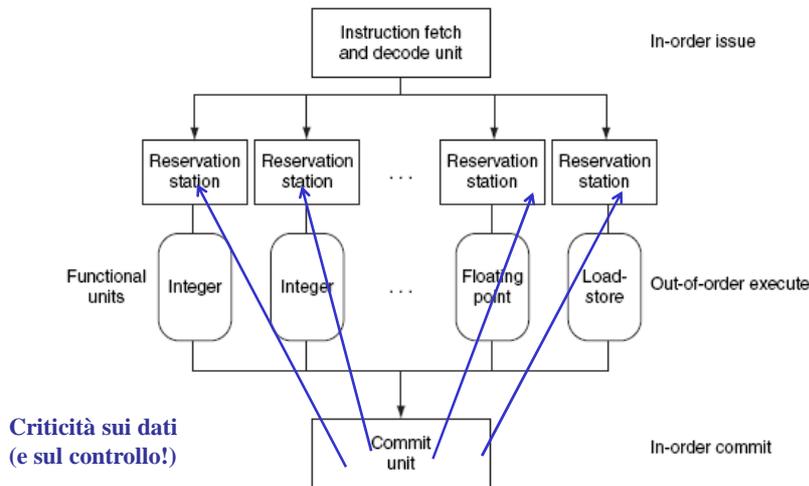
Parallelismo a livello di parola

I bus

Le transazioni su bus



Pipeline con schedulazione dinamica



Esistono diversi cammini paralleli (per la fase di esecuzione e memoria) dell'istruzione, vengono scelti dinamicamente, run-time.

A.A. 2013-2014

3/38

<http://borghese.di.unimi.it>



Come sfruttare il parallelismo a livello di parola



Estensioni MMX e SSE:

Trasferimento dati	Aritmetica	Comparazione
MOV {A/U} {SS/PS/SD/PD} xmm, mem/xmm	ADD {SS/PS/SD/PD} xmm, mem/xmm	CMP {SS/PS/SD/PD}
	SUB {SS/PS/SD/PD} xmm, mem/xmm	
MOV {H/L} {PS/PD} xmm, mem/xmm	MUL {SS/PS/SD/PD} xmm, mem/xmm	
MOV {H/L} {PS/PD} xmm, mem/xmm	DIV {SS/PS/SD/PD} xmm, mem/xmm	
	SQRT {SS/PS/SD/PD} xmm, mem/xmm	
	MAX {SS/PS/SD/PD} xmm, mem/xmm	
	MIN {SS/PS/SD/PD} xmm, mem/xmm	

Different data quantities can be inserted into an xmm register (128 bit):

SS – Scalar, Single precision FP: 1 operand on 32 bit
 PS – Packed Single precision FP: 4 operands on 32 bit
 SD – Scalar, Double precision FP: 1 operand on 64 bit
 PD – Packed Double precision FP: 2 operands on 64 bit
 A – 128 bit aligned in memory

A.A. 2013-2014

4/38

<http://borghese.di.unimi.it>



Sub-words vector operation



Single operation specifies more data.

```
#include <x86intrin.h>
addpd %xmm0, %xmm4    # Multiply two 64 bits FP variables in registers xmm
```

In 2011 *Advanced Vector Extension* (AVX) has been provided by Intel, with registers of 256 bit (internal registers ymm).

```
#include <x86intrin.h>
vaddpd %ymm0, %ymm4   # Multiply four 64 bits FP variables
```

It supports also operations on three registers.

Efficient use when FP adjacent in memory are loaded into registers.



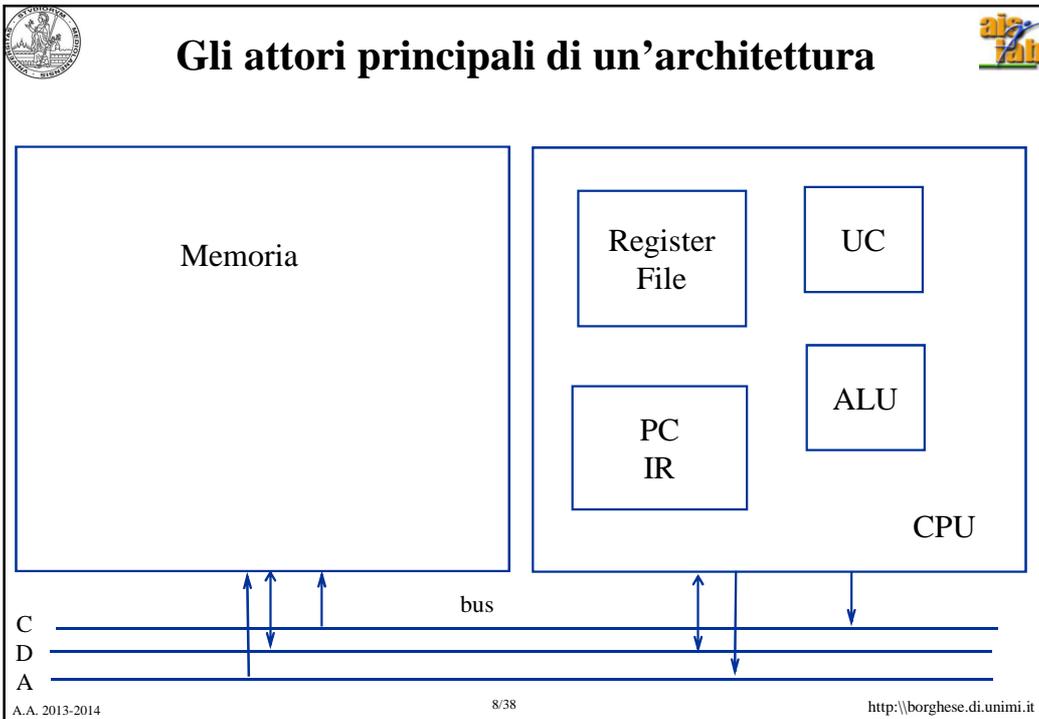
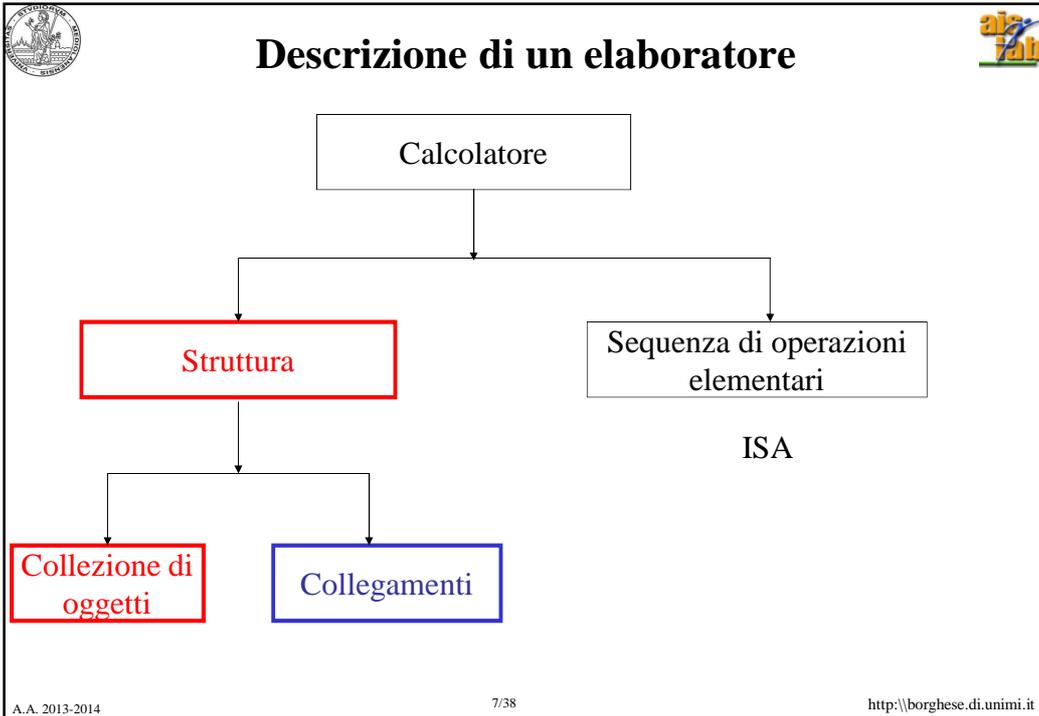
Sommario



Parallelismo a livello di parola

I bus

Le transazioni su bus





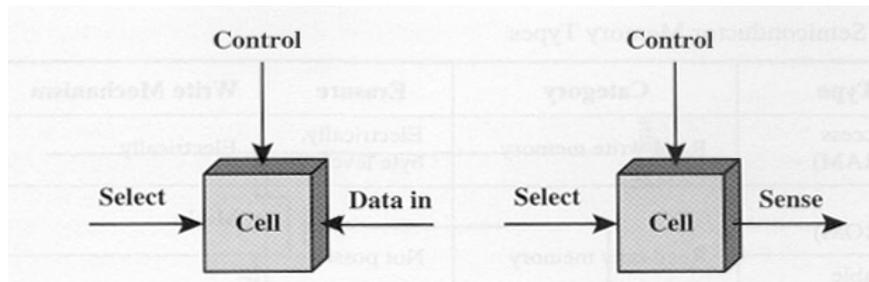
Accesso ad una cella di memoria



La memoria è suddivisa in celle, ciascuna delle quali assume un valore binario stabile.

Si può scrivere il valore 0/1 in una cella.

Si può leggere il valore di ciascuna cella.



Quale struttura di memoria abbiamo già incontrato?

Control (lettura – abilitazione; scrittura)

Select (cf. dataport)

Data in & Sense (Data in & Data out).

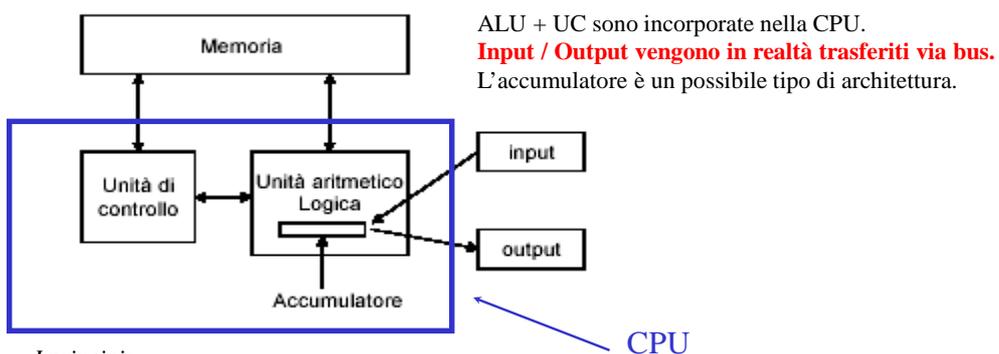
A.A. 2013-2014

9/38

<http://borghese.di.unimi.it>



Architettura di Von Neumann



ALU + UC sono incorporate nella CPU.

Input / Output vengono in realtà trasferiti via bus.

L'accumulatore è un possibile tipo di architettura.

I principi:

- I dati e le istruzioni sono memorizzate in una memoria read/write.
- Il contenuto della memoria può essere recuperato in base alla sua posizione, e non è funzione del tipo di dato.
- L'esecuzione procede sequenzialmente da un'istruzione alla seguente.
- Già' vista e modificata (evoluzione nel tempo).

A.A. 2013-2014

10/38

<http://borghese.di.unimi.it>



I/O



E' la parte più complessa di un'architettura per la sua variabilità

Dispositivi eterogenei per:

velocità di trasferimento.

latenze.

sincronismi.

modalità di interazione (con l'uomo o con una macchina)



Dispositivi di I/O - classificazione



Device	Behavior	Partner	Data rate (Mbit/sec)
Keyboard	Input	Human	0.0001
Mouse	Input	Human	0.0038
Voice Input	Input	Human	0.2640
Sound Input	Input	Machine	3.0000
Scanner	Input	Human	3.2000
Voice output	Output	Human	0.2640
Sound output	Output	Human	8.0000
Laser printer	Output	Human	3.2000
Graphics display	Output	Human	800.0000-8000.0000
Cable modem	Input or output	Machine	0.1280-6.0000
Network/LAN	Input or output	Machine	100.0000-10000.0000
Network/wireless LAN	Input or output	Machine	11.0000-54.0000
Optical disk	Storage	Machine	80.0000-220.0000
Magnetic tape	Storage	Machine	5.0000-120.0000
Flash memory	Storage	Machine	32.0000-200.0000
Magnetic disk	Storage	Machine	800.0000-3000.0000



La comunicazione tra i componenti

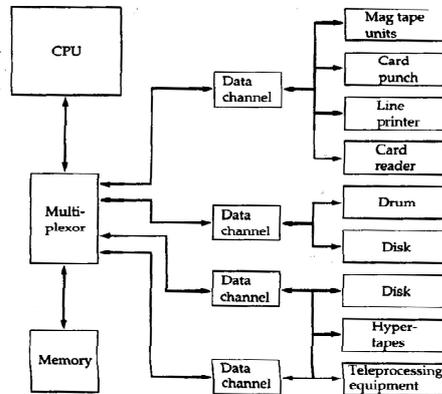


Figure 2.5 An IBM 7094 Configuration

Switch centralizzato (multiplexor)

Architettura a nodo comune (a bus) (cf. bus PCI)

Programma di "canale"

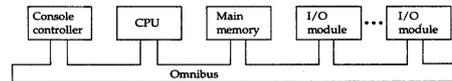


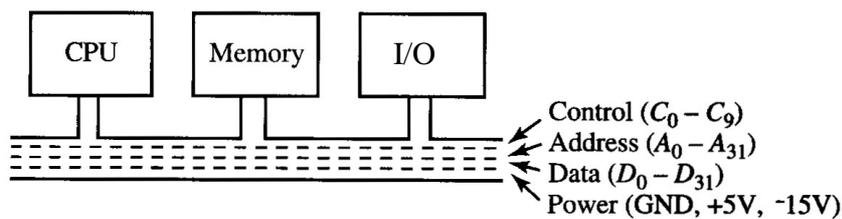
Figure 2.9 PDP-8 Bus Structure

A.A. 2013-2014

1:



Il bus (connessione a cammino comune)



Pathway che connette tutti i dispositivi in modo bidirezionale (a partire dal PDP-8, omnibus, 1965). Trasferimento parallelo dei dati. Connessioni bidirezionali.

Principali vantaggi della struttura a bus singolo:

- elevata flessibilità
- bassi costi.
- Problema: i dispositivi non possono trasmettere contemporaneamente.

Le architetture contengono uno o più bus che collegano questi tre componenti.

A.A. 2013-2014

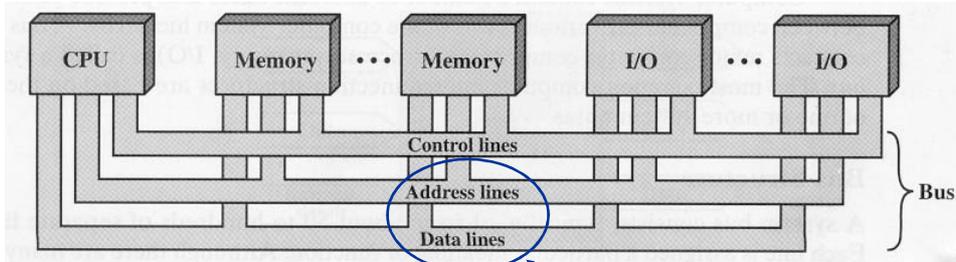
14/38

<http://borghese.di.unimi.it>



La struttura del bus

3 gruppi funzionali: dati, indirizzi e segnali di controllo.



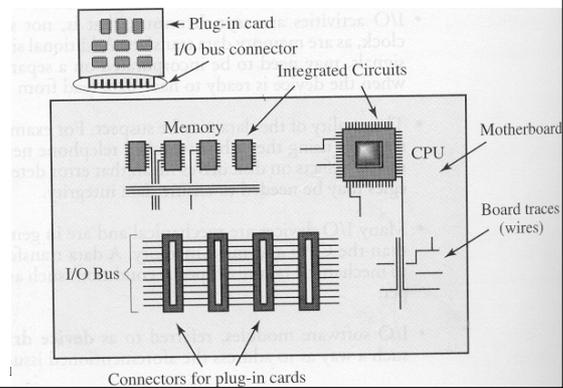
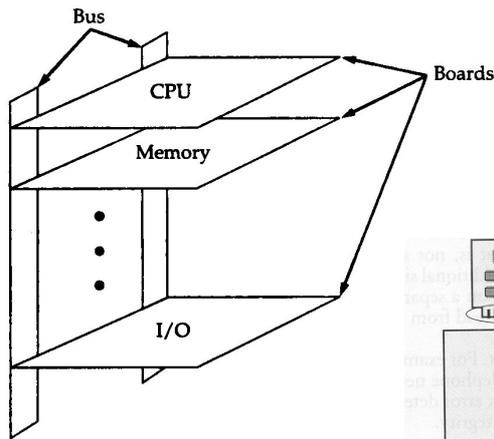
Fisicamente coincidenti nei bus recenti.

Data lines: **ampiezza del bus** = ampiezza della parola dati.
 Address lines: **capacità di indirizzamento** (memoria principale + I/O).
 Control lines: **comandi e stato dei dispositivi**.

Problema: quale dispositivo può trasferire sul bus in caso di richieste multiple? Chi decide?

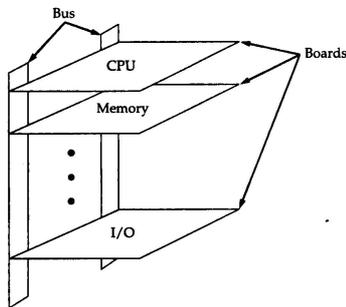


Struttura fisica del bus

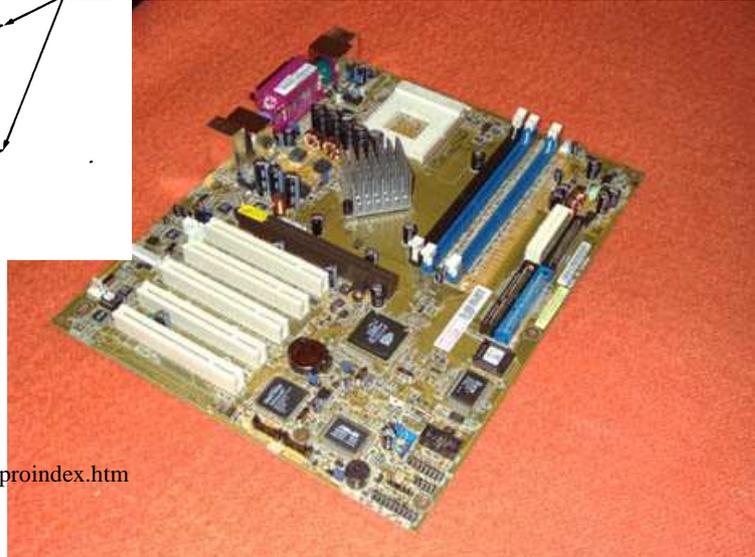




Esempio di mother board



Asus A7N8X Deluxe Specifications

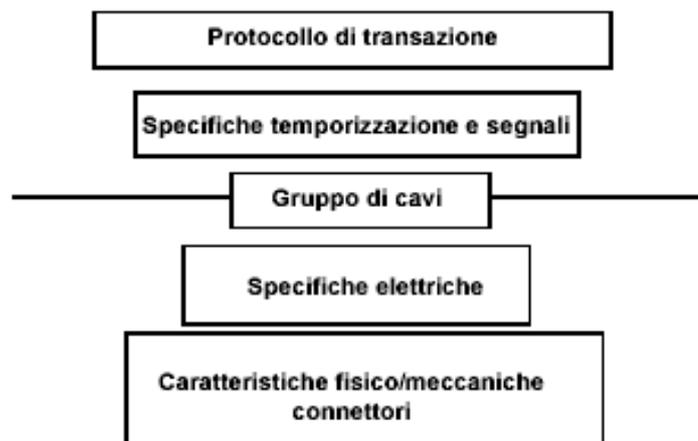


<http://www.asus.it/products/proindex.htm>

A.A. 2013-2014



Livelli di definizione



Transazione su bus: sequenza di operazioni che partono da una richiesta, e si concludono con il trasferimento di dati.

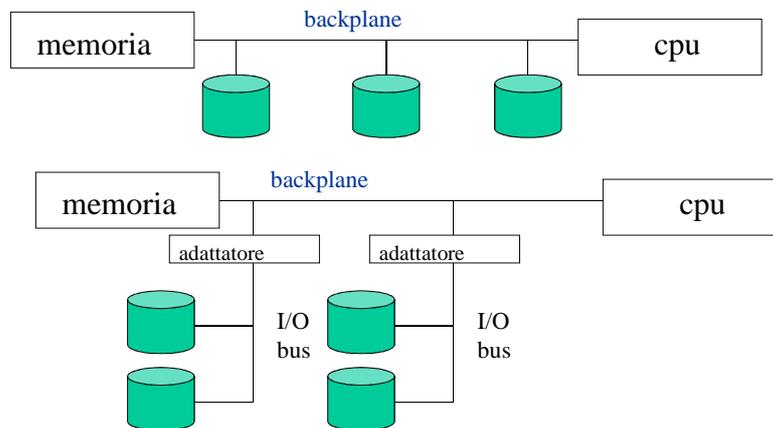
A.A. 2013-2014

18/38

<http://borghese.di.unimi.it>



Tipologie di bus – vecchio stile



A.A. 2013-2014

19/38

<http://borghese.di.unimi.it>

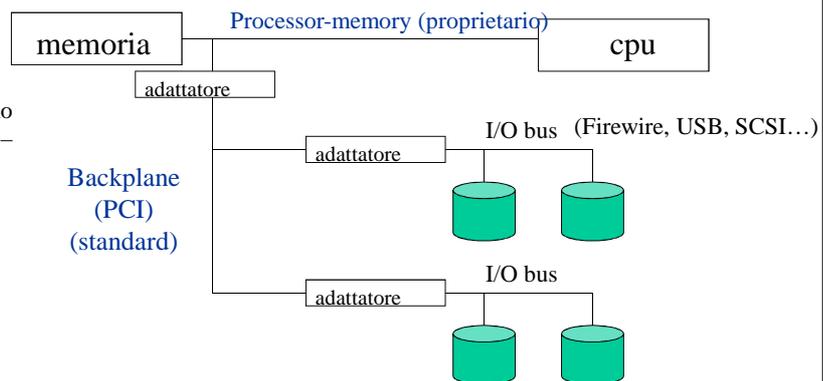


La gerarchia dei bus



Gli adattatori sono dispositivi attivi – bridges.

Backplane (PCI) (standard)



Processore-Memoria (cache): lunghezza ridotta, molto veloci, tipicamente sincroni.

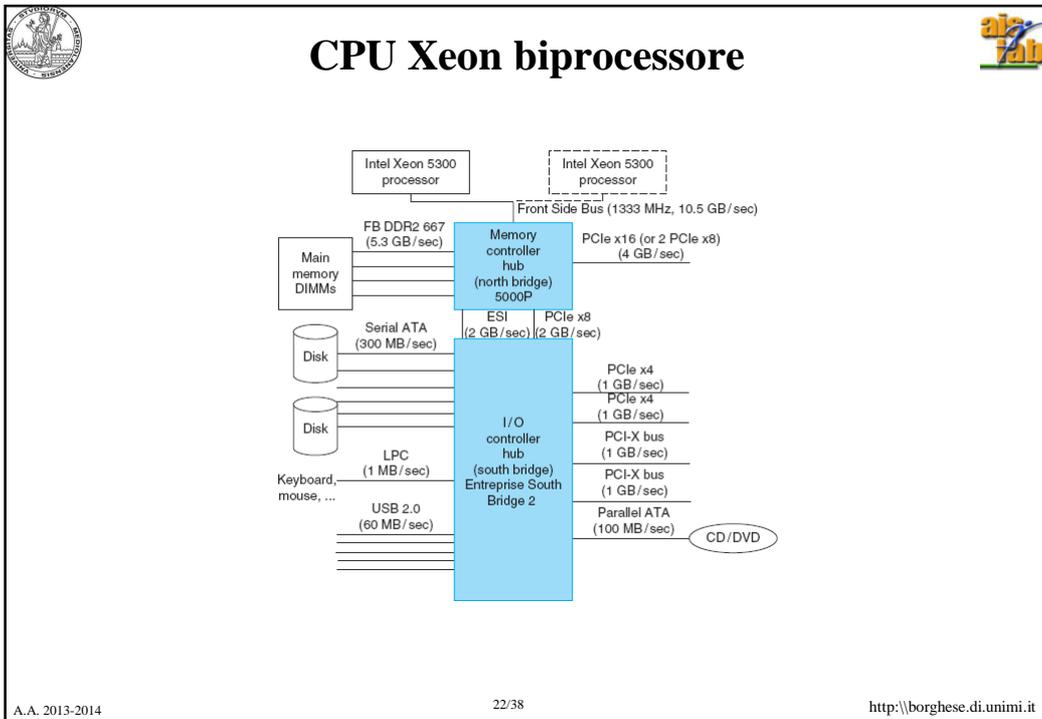
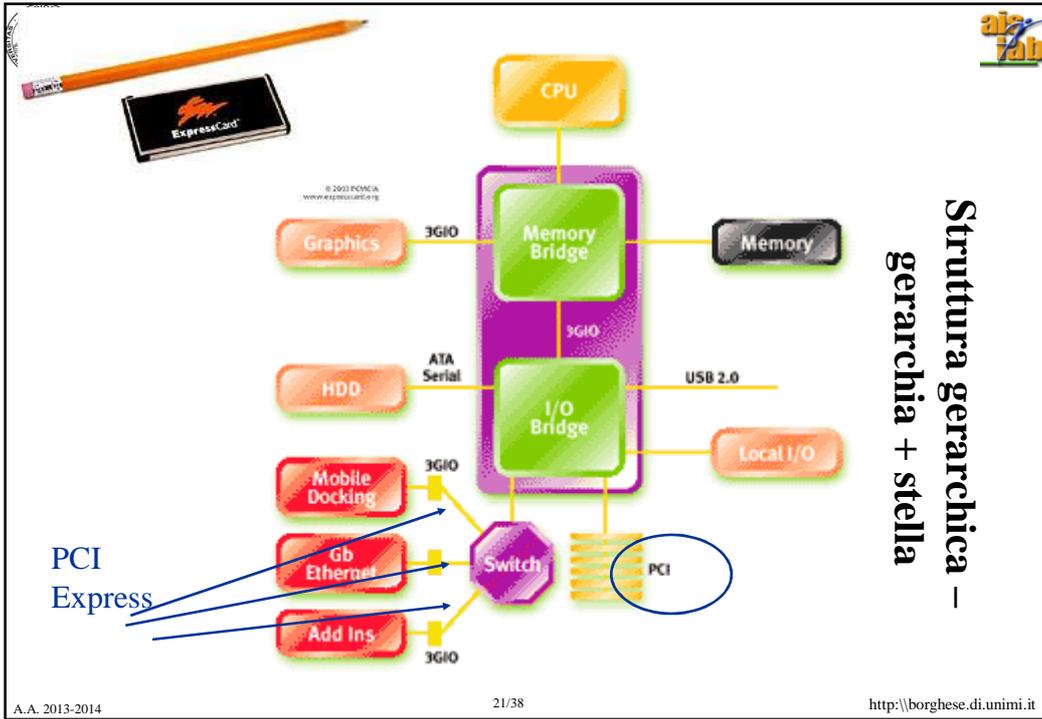
Back-plane: notevole lunghezza, molti device connessi.

I/O: servono per far coesistere la memoria, il processore e i dispositivi di I/O su di un unico bus. Tipicamente asincroni.

A.A. 2013-2014

20/38

<http://borghese.di.unimi.it>





	Intel 5000P chip set	Intel 975X chip set	AMD 580X CrossFire
Target segment	Server	Performance PC	Server/Performance PC
Front Side Bus (64 bit)	1066/1333 MHz	800/1066 MHz	—
Memory controller hub ("north bridge")			
Product name	Blackbird 5000P MCH	975X MCH	
Pins	1432	1202	
Memory type, speed	DDR2 FBDIMM 667/533	DDR2 800/667/533	
Memory buses, widths	4 x 72	1 x 72	
Number of DIMMs, DRAM/DIMM	16, 1 GB/2 GB/4 GB	4, 1 GB/2 GB	
Maximum memory capacity	64 GB	8 GB	
Memory error correction available?	Yes	No	
PCIe/External Graphics Interface	1 PCIe x16 or 2 PCIe x	1 PCIe x16 or 2 PCIe x8	
South bridge interface	PCIe x8, ESI	PCIe x8	
I/O controller hub ("south bridge")			
Product name	6321 ESB	ICH7	580X CrossFire
Package size, pins	1284	652	549
PCHbus: width, speed	Two 64-bit, 133 MHz	32-bit, 33 MHz, 6 masters	—
PCI Express ports	Three PCIe x4		Two PCIe x16, Four PCI x1
Ethernet MAC controller, interface	—	1000/100/10 Mbit	—
USB 2.0 ports, controllers	6	8	10
ATA ports, speed	One 100	Two 100	One 133
Serial ATA ports	6	2	4
AC-97 audio controller, interface	—	Yes	Yes
I/O management	SMBus 2.0, GPIO	SMBus 2.0, GPIO	ASF 2.0, GPIO

I bridge



Caratteristiche di bus asincroni

Caratteristiche	Firewire (1394)	USB 2.0
Tipo di BUS	I/O	I/O
Ampiezza bus (numero segnali)	4	2
Clock	Asincrono / Sincrono	Asincrono
Picco di velocità	50Mbyte/s (Firewire 400) 100Mbyte/s (Firewire 800)	0.2Mbyte/s (low speed) 1.5Mbyte/s (full speed) 60Mbyte/s (high speed)
Numero massimo di device	63	127
Nome standard	IEEE 1394a, 1394b	USE Implementors Forum
Lunghezza massima	4.5m	5m

Bus seriali ad alta velocità, punto a punto, con commutazione.



Il PCI Express



Caratteristica	PCI	PCI Express
Tipo di bus	Back-plane	Versatile (back-plane, I/O)
Ampiezza di base del bus (numero di segnali per i dati)	32-64 (collegamento bidirezionale)	4 + 4 (In/Out) (collegamento monodirezionale)
Numero di dispositivi master	molti	1
Temporizzazione	Sincrono 33-66Mhz	Sincronizzato: 2.5GHz
Modalità di funzionamento	Sincrona parallela (1,024Mbit/s – 4,096Mbit/s)	Sincrona seriale (2,5Gbit/s)
Ampiezza di banda di picco teorica	133-512MB/s (PCI64)	300MB/s per direzione
Ampiezza di banda stimata raggiungibile per bus di base	80MB/s	1,200MB/s (x 4 linee)
Massimo numero di dispositivi	1024 (32 dispositivi per segmento)	1
Massima lunghezza del bus	0,5 metri	0,5 metri
#Mbyte / s / linea	4-16	75 + 75
Nome dello standard	PCI	PCI - Express

bottleneck

A.A. 2013-2014

ghese.di.unimi.it



Sommario



Parallelismo a livello di parola

I bus

Le transazioni su bus

A.A. 2013-2014

26/38

http://borghese.di.unimi.it



La transazione su bus



Invio dell'indirizzo
Lettura / scrittura nel dispositivo di I/O

Esistono due schemi principali di comunicazione su di un bus (di operare una transazione):

Sincrono
Asincrono



Bus sincroni

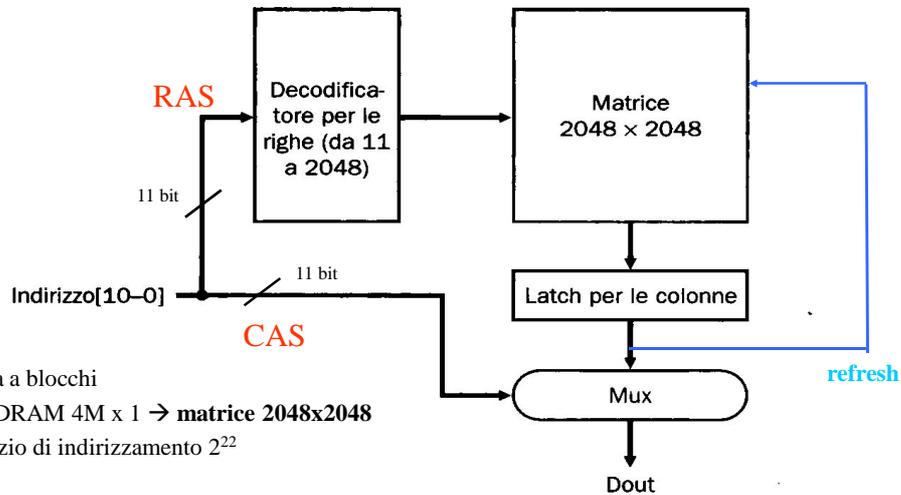


- The le linee di controllo è presente la linea che porta il segnale di clock (**bus clock**). Esiste un protocollo di comunicazione scandito dai cicli di clock, in generale diverso (ma sincronizzato) da quello della CPU.
- Questo tipo di protocollo permette di ottenere bus molto veloci.
- *Svantaggi:*
 - ◆ Ogni device deve essere sincronizzato.
 - ◆ Lunghezza limitata (per evitare che i ritardi nei fronti dovuti alla propagazione producano disallineamenti, clock skew).
 - ◆ Tutti i dispositivi devono potere lavorare alla frequenza imposta dal bus clock.
- I bus processor-memory sono spesso sincroni in quanto:
 - ◆ hanno dimensioni ridotte.
 - ◆ hanno pochi elementi connessi.

Ciclo di bus (**bus cycle**): numero di cicli per effettuare una transazione: tipicamente da 2 a 5 cicli di bus clock.



Struttura a 2 livelli (matrice) di una DRAM



- Struttura a blocchi
 - es. DRAM 4M x 1 → matrice 2048x2048
 - Spazio di indirizzamento 2^{22}

•Accesso:

selezione riga (RAS) + selezione colonna (CAS)

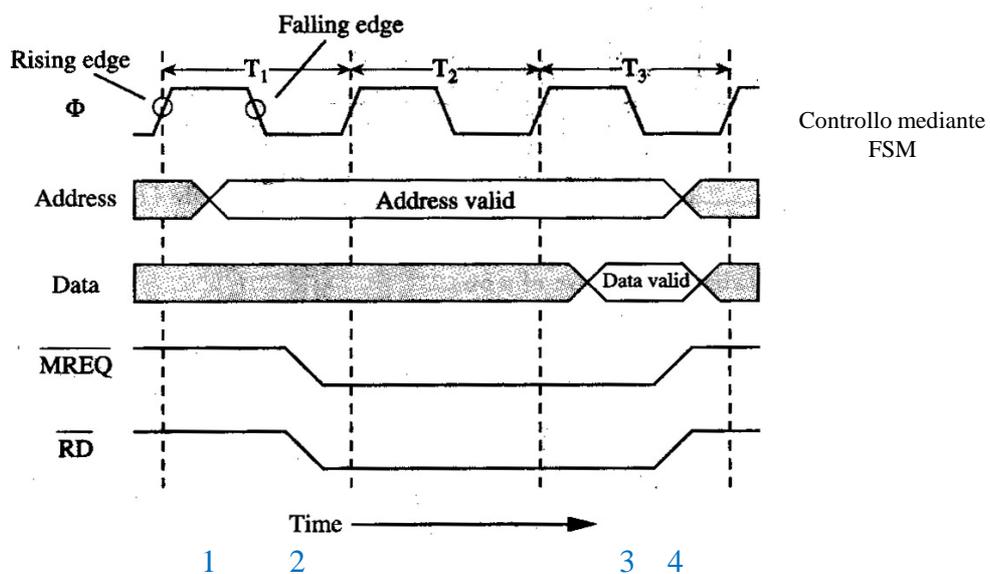
Efficiente per il refresh (refresh di riga) – (35-70ms, tempo di carica dei condensatori).

Utilizzo per la Memoria Principale.

tes.dsi.unimi.it/~borghese



Bus sincroni: esempio



A.A. 2013-2014

30/38

http://borghese.di.unimi.it



Bus asincroni

- Un bus asincrono **non** è dotato di clock.
- La comunicazione tra due parti avviene mediante un protocollo di **handshaking**.
(!MSYN) -> Job -> (!SSYN) -> (MSYN) -> (SSYN)
- I bus asincroni possono avere lunghezza elevata per connettere molti dispositivi.
- Sono efficienti quando i tempi di esecuzione delle varie periferiche variano molto tra loro.
- Spesso i bus di I/O sono asincroni.

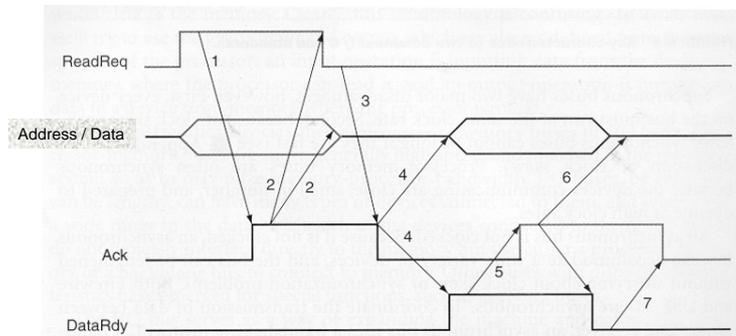


Esempio di handshaking

ReadReq: segnale di controllo di MemoryRead (MREQ+RD). In corrispondenza di questo segnale l'**indirizzo** della parola di memoria viene inviato sul bus dati.

Data Rdy: viene utilizzato per indicare che la parola è pronta sulle linee di dato. Questo segnale viene inviato dalla memoria quando il dato è disponibile in uscita dalla memoria.

Ack: viene utilizzato come risposta ai 2 segnali precedenti. Feed-back (letto indirizzo, letto / scritto il dato).



I segnali di ReadReq e DataReady rimangono alti fino a quando il ricevente (la memoria) non ha visto il comando e letto / scritto i dati corrispondenti.



Arbitraggio del bus



Protocollo La comunicazione su bus deve essere regolata attraverso un **protocollo di comunicazione**.

Viene introdotto il concetto di **bus master (padrone del bus)**, il cui scopo è quello di controllare l'accesso al bus.

L'architettura più semplice è quella che prevede un unico bus master (il processore) in cui tutte le comunicazioni vengono mediate dal processore stesso.

Questo può creare un collo di bottiglia. Ad esempio nel caso di trasferimento di dati da I/O a memoria.

Si utilizza allora un'architettura con più dispositivi master.

In questo caso occorre definire e rispettare una policy che coordini i vari dispositivi bus master. Questa policy si chiama di **arbitraggio** del bus. Un solo dispositivo alla volta può essere master, tutti gli altri ascoltano.

Questo è il principale inconveniente dei bus a nodo comune.



Protocollo di arbitraggio



Occorre stabilire quale master autorizzare all'utilizzo del bus.

Ad ogni dispositivo viene assegnata una *priorità*.

Il dispositivo a priorità maggiore può accedere prima al bus.

Meccanismo di accesso al bus diventa:

1. Richiesta del bus (*bus request*)
2. Assegnamento del bus (*bus grant*)

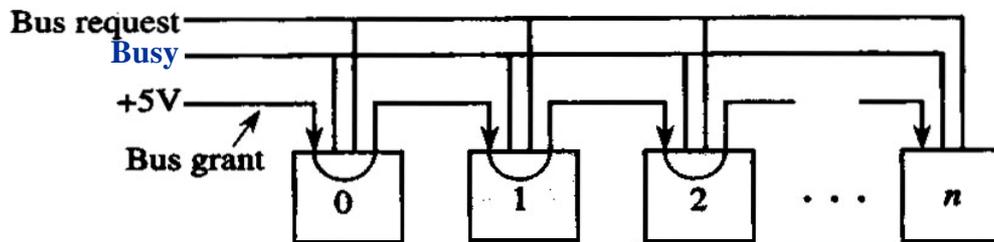
Problema è assicurare una *fairness*.

Compromesso tra *fairness* e *priorità*.

Un arbitro si preoccupa quindi di gestire *bus request* e *bus grant*.



Arbitraggio distribuito in Daisy Chain



Viene introdotto il segnale di Busy.



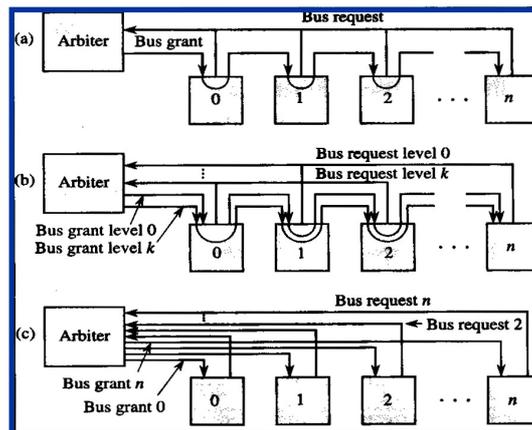
Schemi di arbitraggio centralizzati



Arbitraggio
In Daisy Chain

Arbitraggio
Centralizzato
Con priorità

Arbitraggio
Centralizzato
Parallelo



- Arbitraggio centralizzato parallelo non scala con il numero di dispositivi.
- Nell'arbitraggio centralizzato con priorità, un dispositivo elabora un solo segnale di grant, gli altri segnali di grant vengono fatti passare inalterati.



Arbitraggio distribuito con autoselezione



In questo schema un dispositivo che vuole prendere il controllo del bus, deve:

- 1) Inviare il segnale di richiesta del bus.
- 2) Scrivere sul bus il codice che lo identifica.
- 3) Controllare se il bus è libero.
- 4) Se il bus non è occupato ma ci sono richieste contemporanee di bus, controllare il codice dei dispositivi che hanno fatto richiesta.
- 5) Occupare il bus se è libero o i dispositivi hanno priorità minore: inviare 0 sulla linea di bus grant ai dispositivi successivi, asserisce la linea di busy (il bus è occupato), altrimenti non fare nulla. *Ciascun dispositivo è arbitro.*
- 6) Deassertire la richiesta del bus.

Problema: **rilevamento delle collisioni**. Occorre prevedere un segnale di ricevuto.



Sommario



I bus

Le transazioni su bus