

# Sistemi e Reti

Istituto Tecnico - Indirizzo Info & Telecom  
Articolazione Informatica - Classi Terze

## Panoramica

- [Tipi ed architetture dei Computer](#)
- [Il ruolo della CPU](#)
- [Le memorie](#)
- [Il bus secondo il modello di Von Neumann](#)
- [I bus presenti sul PC](#)
- [La gestione degli I/O](#)
- [Le architetture non Von Neumann](#)

# Tipi di Computer (1)

Classificazione in parte obsoleta, ma utile per ragioni storiche:

- Mainframe
- Supercomputer (Titan, Jaguar)
- Minicomputer
- Home computer (Anni -70/80: Atari, Commodore ecc)
- Personal Computer (Anni 80 e segg.)
- Workstation (PC molto potenti per applicazioni particolari come ricerca, montaggio video, effetti speciali cinematografici, ecc)
- Microcontrollori (SOC – System-On-Chip)

# Tipi di Computer (2)

- Notebook o Laptop
- Netbook (ormai poco diffusi)
- Palmari o PDA (ormai soppiantati da Smartphone e tablet)
- Tablet
- Smartphone
- Console per videogames (computer *special purpose*)

# Architettura dei computer

## **Modello di Von Neumann (1945-51)**

- CPU
- Memoria contenente sia le Istruzioni (assembly) sia i Dati su cui esse operano
- Input/Output
- Velocità caratterizzata dalla frequenza di clock (misurata in Hz)
- Capacità di calcolo misurata in MIPS

# Architettura dei computer

## **Modello Harvard**

- CPU
- Memoria dedicata alle Istruzioni
- Memoria dedicata ai Dati
- Input/Output

Utilizzato per processori molto specializzati, come tipicamente i DSP

# Architettura dei computer

## **Memoria**

- RAM, volatile, riscrivibile, veloce (tempo di accesso di qualche ns)
- ROM, permanente, sola lettura

## **Input/Output**

- I dispositivi di I/O sono collegati al Bus tramite circuiti di interfaccia (*controller*)

## **CPU (processore)**

- lettura (fetch) ed esecuzione (execute) delle istruzioni
- trasferimento dati da/a CPU - memoria - I/O attraverso il Bus

# Architettura dei computer

## **Clock**

- Fornisce i segnali di sincronizzazione a tutti i componenti del computer (come un metronomo in musica)

## Le componenti della CPU

- CU – Control Unit: sovrintende alle operazioni del sistema
- ALU – Arithmetic and Logic Unit: esegue i calcoli aritmetici e logici
- Registri: locazioni di memoria interne alla CPU
  - PC (Program Counter) punta all'istruzione corrente
  - PSW (Process Status Word) con tutti i suoi flag
  - Registri Generali (R1 .. Rn) dimensioni e numero dipendono dal particolare processore

## Il funzionamento della CPU

- *Fetch* dell'istruzione (dalla RAM)
- *Decode* dell'istruzione
- *Fetch* degli operandi (p.es. dalla RAM, o da un registro interno)
- *Execute* (che a sua volta può prevedere la scrittura di un risultato in RAM o in un registro, oppure il salto ad una locazione diversa del programma, ecc)

# Architetture CISC e RISC

- **CISC**

più istruzioni a disposizione del programmatore, quindi programmi più compatti. Chip più complesso (e costoso)

Es.: famiglia Intel x86

- **RISC**

poche istruzioni ma molto veloci, programmi più grandi. Chip più semplice (ed economico)

Es.: famiglia ARM (smartphone Android, iPhone, iPod, iPad, Apple TV, Blackberry ...)

# I tipi di memoria

- **RAM**

- DRAM: necessita di continuo refresh; relativamente lenta (20-70ns); economica; usata come RAM "generica"
- SRAM: non richiede refresh; molto veloce (5-10ns); costosa; usata nelle *cache*

- **ROM**

- non programmabili (costruite già con il programma/dati al loro interno)
- PROM Programmabili una sola volta
- EPROM Cancellabili con raggi UV
- EEPROM Cancellabili elettricamente

# Indirizzamento delle memorie

- Spazio di indirizzamento
  - con N bit =  $2^N$  (N = numero di fili del bus indirizzi)
  - Max numero di Byte indirizzabili (quasi mai sono tutti presenti!)
- Ampiezza di parola (word)
  - il numero di bit che si possono trasferire simultaneamente (= numero di fili del bus dati)

# Gestione memoria in un PC

- **Problema:** la comunicazione fra CPU e memoria è il *collo di bottiglia* del sistema (CPU >> veloce della DRAM)
- **Obiettivo:** ottenere prestazioni (tempi di accesso ai dati) migliori mantenendo costi contenuti senza sacrificare le capacità di memoria
- **Soluzione:** organizzare le memorie in modo gerarchico:

LIVELLO	CAPACITÀ	VELOCITÀ
Cache	Bassa	Alta
DRAM	Media	Media
Memorie di massa	Alta	Bassa

## Gestione memoria in un PC

- Le cache si trovano all'interno dei processori (architettura Harvard: dati separati dalle istruzioni)
- Tipicamente sono a 3 livelli (L1, L2, L3) di dimensioni crescenti (dell'ordine dei MB)
- Nelle CPU multi-core si possono avere p.es.
  - L1 riservati a ciascun core
  - L2 comuni a più core
  - L3 condivisa fra tutti i core
- Funzionamento: cache *hit* e cache *miss*  
(inst\_read miss ≠ data\_read miss ≠ data\_write miss)

N.B. meccanismi di *caching* vengono utilizzati anche nell'accesso a *dischi*, al *web*, nei *motori di ricerca*, nei *database*, nei *DNS* ...

## Caratteristiche della memoria

I moduli di memoria RAM sono caratterizzate da alcuni parametri fondamentali:

- **Capacità:** dimensione in Byte (p.es. 2048 MB)
- **Tipo:** (p.es. DDR1, DDR2, DDR3)
- **Frequenza:** la frequenza del FSB (p.es. 1333 MHz)
- **Latenza:** numero di cicli del FSB per leggere un dato dalla memoria (p.es. 2 o 3)



# Caratteristiche della DRAM

Parametri caratteristici delle **DRAM**, tutti espressi in cicli del FSB clock (p.es.: 2-4-4-8):

- CAS Latency Time
- RAS Precharge Time
- RAS to CAS Delay
- Active to Precharge

Questi (e molti altri) parametri fondamentali per il suo corretto funzionamento sono memorizzati sulla **SPD (Serial Presence Detect)** una EPROM montata sulla scheda di memoria

*N.B.: Il modulo di memoria più lento definisce la velocità di tutta la RAM installata*

# Il Bus (arch. Von Neumann)

Il Bus è un insieme di fili che attraversano la motherboard ed a cui si collegano i vari moduli.

- Permette a due qualsiasi moduli che lo condividono di scambiarsi dati
- È necessario un arbitraggio per evitare conflitti con altri moduli
- Trasporta **una informazione alla volta**
- Un dispositivo **trasmette**, un altro **riceve**, mentre gli altri restano scollegati (tri-state)

Il bus in realtà è costituito da 3 "sotto-bus": bus **dati**, **indirizzi** e **controlli**, descritti nelle slide che seguono.

## Il Bus Dati

- **Data Bus:** su cui transitano i dati nei trasferimenti(\*)
  - fra CPU e memoria (read/write o load/store)
  - fra CPU e porte di I/O (in/out)

È un Bus bi-direzionale

La sua ampiezza rappresenta il numero di bit che si possono trasferire simultaneamente.

(Con le CPU più recenti: 64)

(\*) *N.B.: Non è possibile trasferire direttamente dati da RAM a RAM*

## Il Bus Indirizzi

- **Address Bus:** determina l'indirizzo di memoria (o di I/O) da/verso cui la CPU sta leggendo/scrivendo
  - mono-direzionale
  - pilotato dalla CPU

La sua ampiezza  $n$  determina la dimensione (in Byte) dello *spazio indirizzabile*  $M$  della CPU attraverso la relazione

$$M = 2^n$$

## Il Bus di Controllo

- **Control Bus:** Non è un vero e proprio bus, ma un insieme di fili che trasportano segnali indispensabili per il coordinamento delle operazioni di trasferimento dati.
  - **RD** da MEM (oppure I/O) verso CPU
  - **WR** da CPU (oppure I/O) verso MEM
  - **MEM** accesso che riguarda la memoria
  - **I/O** accesso che riguarda l'I/O
  - **INT** richiesta di interrupt da parte dell'I/O (quando è pronto per ricevere/inviare dati dalla/alla CPU)
  - **Clock** il segnale che sincronizza tutte le operazioni

*N.B.: RD e WR sono mutuamente esclusivi, così come MEM e I/O.*

## I bus presenti sul PC

I PC degli anni '70 e '80 avevano un solo bus detto **bus di sistema** (system bus) cui si affacciavano tutti i dispositivi che vi accedevano tutti alla stessa velocità.

Esempi di tale bus sono il PC Bus, XT Bus (8-bit), AT Bus [aka ISA Bus] (16-bit), EISA Bus (32-bit).

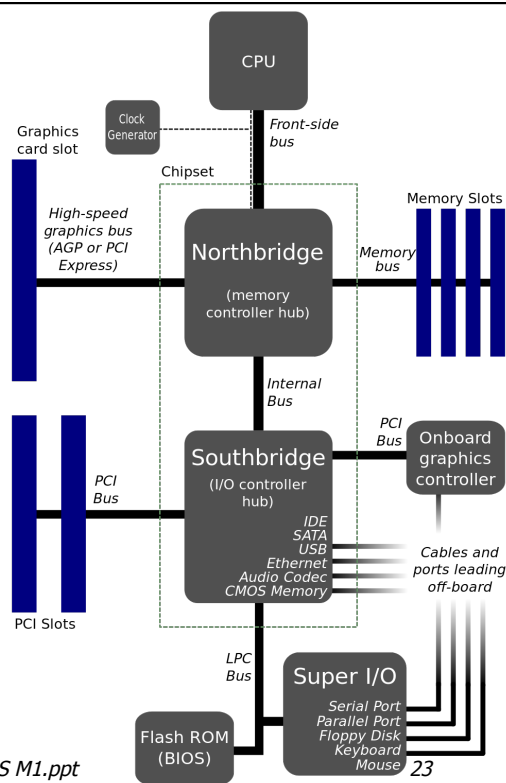
Per evitare di rallentare troppo la CPU, negli anni '90, con i processori Pentium Pro e Pentium II, è stato introdotto il concetto di **Dual Independent Bus** (DIB):

- Front-Side Bus (FSB) verso la memoria centrale e le periferiche
- Back-Side Bus verso altre (eventuali) CPU e memorie cache

## Schema di una Motherboard

**Northbridge:**  
chipset dedicato ad interfacciare i dispositivi veloci

**Southbridge:**  
chipset dedicato ad interfacciare l'I/O



PS 09/2013

Traccia SIS M1.ppt

## I bus presenti sul PC

Attualmente all'interno di un PC vi sono più bus, ciascuno specializzato in un tipo di traffico:

- Back-Side Bus collega la CPU alla cache
- Front-Side Bus collega la CPU alla RAM ed ai bus di espansione (v. sotto)
- Expansion Bus alcuni esempi
  - INTERNI:
    - AGP: Schede grafiche ad alte prestazioni
    - PCI Express:
    - PCI: Qualsiasi tipo di periferica
    - IDE (ATA, P-ATA) per collegare fino a due HD
    - SCSI: Nastri e HD
  - ESTERNI (Plug-and-Play, PnP):
    - USB: Mouse, Tastiera, fino a Memorie flash, HDD, Fotocamere, Videocamere, Stampanti, Scanner, Microfoni, Casse, ecc
    - FireWire: Fotocamere, Videocamere, altri computer
    - SATA HD

PS 09/2013

Traccia SIS M1.ppt

24

## Plug-and-Play (PnP)

Indica la capacità di un sistema di auto-configurare la gestione di una periferica senza bisogno di intervento manuale.

Richiede di essere supportata a livello di:

- BIOS
- Sistema Operativo
- Dispositivo

Introdotta con Windows 95, prima era necessario configurare manualmente (quindi con rischio di errori):

- IRQ: Interrupt utilizzato dal dispositivo
- DMA: canale di accesso alla memoria senza passare per la CPU
- Indirizzi di memoria dedicata
- Porte di I/O

## La gestione degli I/O

I dispositivi di I/O convertono fenomeni fisici del mondo reale (es.: pressione di un tasto, movimento del mouse su una superficie, accelerazione, luce, suono) in segnali elettrici discreti (e viceversa).

- 0V            valore logico 0
- +5V          valore logico 1

Un generico dispositivo di I/O si può vedere come costituito di due parti:

- La **periferica**, esterna al computer
- Il **controllore**, facente parte del computer, che converte i segnali provenienti dalla periferica in segnali digitali intelligibili dalla CPU (e viceversa)

## La gestione degli I/O

La distinzione fra *periferica* e *controller* non è sempre così esplicita, dato che spesso i dispositivi di I/O contengono al loro interno anche il controller, che a sua volta è costituito da un vero e proprio **microcomputer** (es.: tastiera, webcam, microfono USB, scanner, stampante)

Addirittura, nel caso delle schede grafiche, siamo confrontati a potenti **computer embedded** dotati di GPU, RAM locale, software dedicato

## Le porte di I/O

Ogni porta di I/O è dotata di un *indirizzo* ed al minimo può avere la dimensione di 1 Byte.

In genere lo spazio di indirizzamento dell'I/O è diverso dallo spazio di indirizzamento della memoria.

In taluni casi però si sceglie di dedicare parte dello spazio di memoria all'I/O che a questo punto può essere trattato come se fosse parte della memoria (*memory-mapped I/O*).

## Connettori di I/O di un PC

- **PS/2**(\*) connettori per mouse e tastiera
- **seriale**(\*) per il collegamento di periferiche lente (p.es. modem telefonico)
- **parallela**(\*) tipicamente usata per collegare stampanti
- **rete** (connettore RJ-45) per collegarsi ad una rete locale
- **USB** per collegare praticamente qualsiasi tipo di dispositivo (attenzione alla versione)
- **FireWire** come e meglio dell'USB, ma meno diffusa
- entrate ed uscite **analogiche** audio (microfono, cuffie, altoparlanti, ecc)

## Architetture non Von Neumann

Tecniche per migliorare le prestazioni all'interno della stessa architettura:

- aumento *frequenza di clock*
- aumento *ampiezza di parola* (bus dati)
- aumento *spazio di indirizzamento* (bus indirizzi)

Vi sono però dei fattori che impediscono di aumentare le prestazioni oltre un certo limite:

- *memory wall*: la lentezza della memoria rispetto al processore
- *ILP (instruction level parallelism) wall*: difficoltà di individuare istruzioni da poter eseguire in parallelo
- *power wall*: aumento esponenziale del consumo di energia per aumento fattoriale della frequenza di clock

# Architetture non Von Neumann

Se il problema è strutturale, occorre introdurre modifiche strutturali, essenzialmente introducendo funzionalità di **elaborazione parallela**.

A questo riguardo si individuano le seguenti architetture(\*):

- **SISD** Single Instruction Single Data - è l'architettura classica secondo Von Neumann
- **SIMD** Single Instruction Multiple Data - una istruzione può essere eseguita su più dati; è il tipico caso delle CPU grafiche (GPU)
- **MISD** Multiple Instruction Single Data - più unità eseguono istruzioni sullo stesso dato; utilizzata in sistemi fault-tolerant
- **MIMD** Multiple Instruction Multiple Data - più processori lavorano in parallelo su più dati; è il caso dei processori multi-core divenuti ormai popolari

(\* ) *Classificazione introdotta da Michael G. Flynn nel 1966*

# Evoluzione dei sistemi

L'evoluzione delle architetture non Von Neumann riguarda essenzialmente tre aree:

- Elaborazione
- Memoria centrale
- Input/Output



## Evoluzioni dell'elaborazione

- Esecuzione fuori ordine (OOE)  
CPU analizza al volo il codice ed individua gruppi di istruzioni indipendenti che non hanno vincolo sequenziale => possono essere eseguite in parallelo
- Prefetch  
riguarda sia i dati che le istruzioni – molto complicato da realizzare al volo – possono esistere apposite istruzioni per gestirlo
- Speculative execution  
Esecuzione di entrambi i rami di un salto

[cfr testo a pag. 80 e segg]

PS 09/2013

Traccia SIS M1.ppt

33

## Evoluzioni dell'elaborazione

- Pipeline  
La CPU lavora come una catena di montaggio (pipeline); si presentano però situazioni in cui occorre rallentare l'esecuzione o ricaricare la pipeline
- Tecnologie superscalari  
Più pipeline che lavorano in parallelo  
Più processori su uno stesso chip (multi-core)
- Branch prediction  
Visto che i salti (mediamente uno ogni 6-7 istruzioni) influenzano negativamente la pipeline, si può cercare di prevederli per caricare le istruzioni "giuste"

PS 09/2013

Traccia SIS M1.ppt

34

## Evoluzioni della memoria centrale

- Cache memory

Sfrutta i principi di località spaziale (dei dati) e località temporale (delle istruzioni) - costituita da SRAM - sempre suddivisa a livelli L1/L2/L3 ecc - eventualmente condivisa fra più core sullo stesso chip.

*Esempio* di organizzazione:

- L1: direttamente all'interno del processore
- L2: esterna, collegata tramite il BSB
- L3: esterna, sulla motherboard

Per conservare la coerenza vi sono diverse politiche:

- Cache *write-through*
- Cache *write-back*

[cfr [wikipedia: CPU Cache#Multi-level caches](#)]

## Evoluzioni della memoria centrale

- Virtual memory

Meccanismo perché la CPU possa "vedere" una quantità di memoria centrale molto superiore alla RAM realmente esistente.

La RAM disponibile è suddivisa in pagine (tipicamente da qualche kB) i cui indirizzi reali vengono mappati in indirizzi virtuali.

La mappatura è responsabilità della MMU (Memory Management Unit).

Quando tutta la RAM è occupata, alcune pagine vengono spostate su disco (con effetti disastrosi sulle prestazioni ...)

# Evoluzioni dell'I/O

- **DMA (Direct Memory Access)**

Meccanismo che scarica la CPU dai trasferimenti di grandi quantità di dati. Richiede la presenza di un DMA controller, che usa lo stesso system bus della CPU quando la CPU non lo utilizza.

La CPU interviene solo all'inizio ed alla fine del trasferimento, il lavoro pesante lo fa il DMA controller.

- **Coprocessori**

CPU secondarie specializzate in determinati compiti:

- FPU (Floating Point Unit) coprocessore matematico, può essere sullo stesso chip o su un chip separato
- GPU (Graphic Processing Unit) coprocessore grafico (anche multicore) spesso dotato di una sua scheda (scheda video) con RAM dedicata