

Sistemi di Elaborazione

CdL Economia Informatica
AA 2006/07 - Facoltà di Economia
Università di Chieti - Pescara

Dott. Fabio Fioravanti



Note introduttive

- Lezioni
- Libri di testo
- Programma
- Materiale didattico
- Pagina web del corso



Programma del corso



- Architetture degli elaboratori
 - Logica digitale e rappresentazione dei dati
 - Architettura di von Neumann: CPU, memoria, I/O, Fasi di esecuzione delle Istruzioni.
 - Classificazione delle memorie e loro caratteristiche.
- Sistemi operativi
 - Struttura dei sistemi di calcolo e dei sistemi operativi
 - Gestione dei processi: processi, threads, scheduling, sincronizzazione e deadlock.
 - Gestione della memoria
 - File system
 - Sistemi di input/output
 - Sistemi distribuiti
 - Sicurezza
- Cenni su architetture, sistemi e tecnologie avanzate.

Testi consigliati



- Libro di testo
 - A. Silberschatz, P. Galvin, G. Gagne
Sistemi operativi: Concetti ed esempi
7a Edizione - 2006 Pearson
- Testi di consultazione
 - A. Tanenbaum, Architettura dei calcolatori: Un approccio strutturale - 5a Edizione - 2006 Pearson
 - I. Englander, The Architecture of Computer Hardware and Systems Software: An Information Technology Approach, 3rd Edition, Wiley

Architetture degli elaboratori



- Principi di base
 - Hardware / Software
 - Programmi
 - Codifica dell'informazione
- Architettura di Von Neumann
 - CPU
 - Registri, ALU, FPU, CU, CISC/RISC
 - RAM
 - BUS
 - Periferiche input/output
 - Clock
- Ciclo della CPU
- Assembly, linguaggio macchina e assembler

Il processo di elaborazione

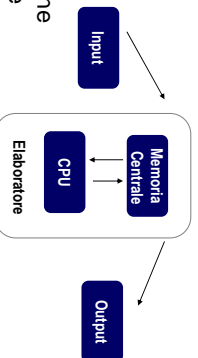


- Elaboratore svolge una funzione ben precisa in maniera automatica.
- L'automatismo avviene attraverso il passaggio di energia elettrica attraverso i componenti del computer, circuiti, transistor, etc..

La macchina computer



- In generale, un computer:
 - esegue *operazioni* logiche e aritmetiche,
 - ha una *memoria* per conservare i dati.
- Un *programma* contiene le informazioni relative alle operazioni da eseguire.



Schema di un sistema di Elaborazione



- Operare con **Information Technology** significa avere a che fare con un sistema di elaborazione delle informazioni



La rappresentazione delle informazioni (cont.)



- Le ragioni di questa scelta sono prevalentemente di tipo tecnologico:
 - Due possibili stati di polarizzazione di una sostanza magnetizzabile;
 - Passaggio/non passaggio di corrente attraverso un conduttore;
 - Passaggio/non passaggio della luce attraverso una fibra ottica.

Hardware vs. Software



- L'*hardware* denota la struttura fisica del computer, costituita di norma da componenti elettronici che svolgono specifiche funzioni nel trattamento dell'informazione.
- Il *software* denota l'insieme delle istruzioni che consentono all'hardware di svolgere i propri compiti.

La rappresentazione delle informazioni



- Tutte le informazioni sono rappresentate in forma **binaria** o **digitale** utilizzando due soli simboli: 0 ed 1.
- BIT (Binary digIT): Unità fisica di informazione che vale 0 oppure 1.
- Si utilizzano i multipli del bit:

• Kilo	Kb	2 ¹⁰	~ un migliaio	(1024)
• Mega	Mb	2 ²⁰	~ un milione	(1024x1024)
• Giga	Gb	2 ³⁰	~ un miliardo	(1Mb x 1024)
• Tera	Tb	2 ⁴⁰	~ mille miliardi	(1Gb x 1024)

Codifica binaria



- Per poter rappresentare un numero maggiore di informazioni è necessario utilizzare *sequenze* di bit.
 - Utilizzando due bit si possono rappresentare quattro informazioni diverse:

00	01	10	11
----	----	----	----
- Il processo che fa corrispondere ad una informazione una configurazione di bit prende il nome di *codifica dell'informazione*.
- Con k bit rappresento 2^k informazioni

Sequenze di bit



Numero di bit nella sequenza	Informazioni rappresentabili
2	4
3	8
4	16
5	32
6	64
7	128
8	256

Il byte



- Un gruppo di 8 bit viene denominato Byte.
 - Corrisponde ad un carattere.
 - Unità di misura della capacità di memoria.
- Si utilizzano i multipli del Byte:
 - Kilo KB 2^{10} ~ un migliaio (1024)
 - Mega MB 2^{20} ~ un milione (1024x1024)
 - Giga GB 2^{30} ~ un miliardo (1MBx1024)
 - Tera TB 2^{40} ~ mille miliardi (1GBx1024)

Rappresentazione di dati alfabetici



- Un codice numerico per ogni carattere
- Codifiche standard:
 - **ASCII**, 8 bit per carattere, rappresenta 256 caratteri.
 - **UNICODE**, 16 bit per carattere
 - ASCII e caratteri etnici.
- Codifiche proprietarie:
 - **MSWindows**, 16 bit per carattere
 - simile ad UNICODE.

Sistema di numerazione decimale



- La numerazione decimale utilizza una **notazione posizionale** basata su 10 cifre (da 0 a 9) e sulle potenze di 10
 - Il numero 234 può essere rappresentato esplicitamente come:
 $2 \times 10^2 + 3 \times 10^1 + 4 \times 10^0$

Sistema di numerazione binario



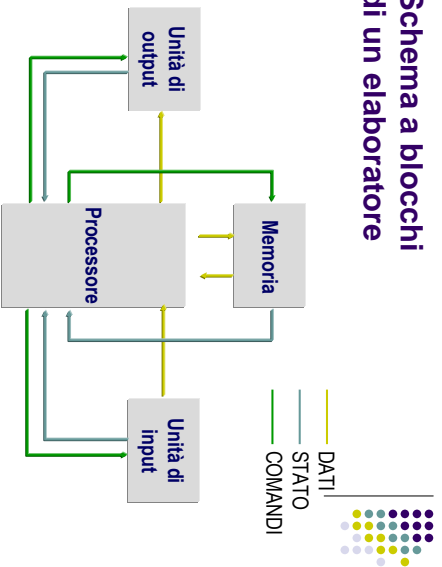
- Il sistema di numerazione binario utilizza una **notazione posizionale** basata su 2 cifre (0 e 1) e sulle potenze di 2
 - Il numero 1001 può essere rappresentato esplicitamente come:
 $1001_2 = 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 9_{10}$

Architettura dei computer



- In un computer possiamo distinguere tre unità funzionali:
 - **Processore**
 - fornisce la capacità di elaborazione delle informazioni;
 - **Memoria (centrale e di massa);**
 - **Dispositivi di input/output;**
 - che comunicano attraverso un canale detto BUS costituito da un insieme di linee elettriche digitali.
 - **Macchina di von Neumann**
 - Programma memorizzato
 - Non c'è differenza in memoria tra dati e istruzioni

Schema a blocchi di un elaboratore



Programma ed istruzioni

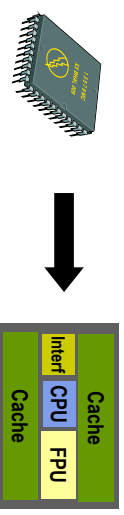
- **Programma**: specifica un'ivoca di una serie di operazioni che l'elaboratore deve svolgere.
- E' costituito da una sequenza ordinata di **istruzioni macchina** scritte in **codice macchina**.

Istruzioni e codice macchina

- Istruzione macchina:
 - Specifica un'istruzione elementare che il processore è in grado di svolgere.
 - Generalmente composta da:
 - **Codice operazione** (indica cosa fare),
 - **Uno o due operandi** (su cosa operare),
 - **Destinazione del risultato** (dove memorizzarlo).
- **Codice macchina**:
 - Codice binario usato per specificare le istruzioni macchina al processore.
 - Diverso per ogni processore (non esiste codice universale standard).

Processore

- Composto da blocchi con funzionalità diverse:
 - CPU (Central Processing Unit), unità centrale di elaborazione
 - FPU (Floating Point Unit),
 - Cache,
 - Interfacce varie.
- Se integrato su un unico chip prende il nome di **microprocessore**.



Tipi di processore

- Ogni processore possiede un "set" di istruzioni macchina che costituiscono i programmi.
- Una distinzione fondamentale fra i processori è quella che li differenzia in:
 - **CISC**: **Complex Instructions Set Computer**.
 - **RISC**: **Reduced Instructions Set Computer**.

Tipi di processore (cont.)

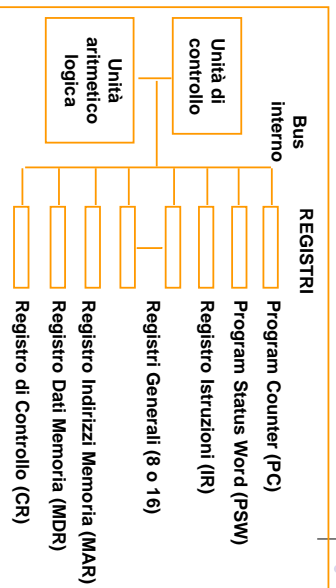
- La differenza è nel set di istruzioni.
 - I CISC hanno un **linguaggio macchina** formato da un numero elevato di istruzioni, anche complesse.
 - **Es.:** Intel x86, Motorola 68000.
 - I RISC sono dotati di istruzioni più semplici e in numero minore. Quelle complesse vengono emulate via software (interprete)
 - **Es.:** PowerPC, Sparc.

Central Processing Unit (CPU)



- Svolge tutte le operazioni di:
 - elaborazione numerica,
 - controllo e coordinamento di tutte le attività.
- Si suddivide in:
 - Unità logico-aritmetica (ALU): esegue istruzioni aritmetiche e logiche.
 - Unità di controllo (CU): sovrintende all'elaborazione dei dati e alle operazioni di input e output.
 - Registri, memoria locale per memorizzare dati e lo stato di avanzamento delle istruzioni
 - Generali o Speciali

Componenti della CPU

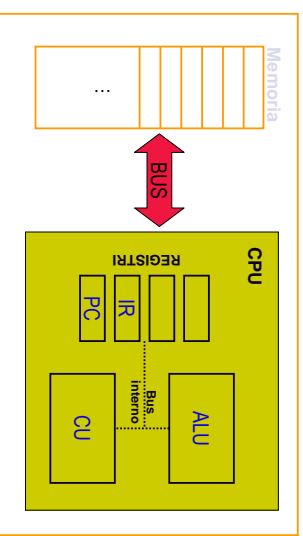


Registri



- Memoria locale usata per memorizzare:
 - dati acquisiti dalla memoria centrale o dalla unità di input,
 - risultati delle operazioni eseguite dall' ALU.
- Numero limitato: tipicamente da 8 a 256.
- Unità di memoria estremamente veloci.
- Le dimensioni di un registro sono una caratteristica fondamentale del processore: 16, 32, 64 bit.

Schema della CPU



Arithmetic Logic Unit (ALU)



- Svolge tutti i calcoli *logici ed aritmetici* (complementazione, somma intera, confronto, etc).
 - Opera direttamente sui registri generali.
- E' costituita da circuiti elettronici in grado di eseguire la *somma* di due numeri binari contenuti in due registri oppure di eseguire il *confronto* tra due numeri.

Registri (cont.)



- I registri contengono dati ed informazioni che vengono immediatamente elaborate.
- Esistono due tipi di registri:
 - i *registri speciali* utilizzati dalla CU per scopi particolari,
 - i *registri di uso generale* (registri aritmetici).

Registri speciali



- Il *Program Counter (PC)* contiene l'indirizzo di memoria che contiene la prossima istruzione da eseguire.
- L' *Instruction Register (IR)* contiene l'istruzione attualmente in esecuzione.
- Il *registro di stato (PSW)* contiene delle informazioni sullo stato di esecuzione del processore.
 - In particolare esso può segnalare eventuali errori che possono avvenire durante l'esecuzione di un programma.

Control Unit (CU)



- E' la parte più importante del processore:
 - In base:
 - al *programma* in esecuzione ed
 - allo stato di tutte le unità,decide l'operazione da eseguire ed emette gli ordini relativi.
- In pratica, esegue le istruzioni macchina.

Floating Point Unit (FPU)

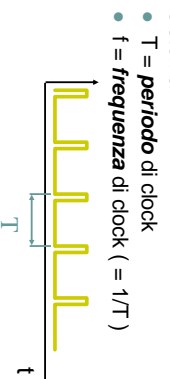


- In molti elaboratori si può avere una FPU specializzata in operazioni matematiche complesse (numeri reali).
- Svolge tutte le operazioni che la ALU della CPU non è in grado di svolgere:
 - Somma/sottrazione reale,
 - Moltiplicazione intera/reale,
 - Divisione,
 - Funzioni matematiche complesse.
- Nei calcolatori di generazioni precedenti era esterna alla CPU (**coprocessore matematico**).

Il clock



- Ogni elaboratore contiene un circuito di temporizzazione (**clock**) che genera un riferimento temporale comune per tutti gli elementi del sistema.



- Frequenze tipiche delle ultime generazioni:
 - $f > 1000$ MHz, $T < 10^{-9}$ secondi.
 - 4 Megahertz= 4 milioni di cicli al secondo

Tempistica delle istruzioni



- Un *ciclo-macchina* è il tempo richiesto per svolgere un'operazione elementare.
 - È un multiplo del periodo del clock.
- Un'istruzione macchina è ottenuta da una sequenza di operazioni elementari.
 - Dunque, l'esecuzione di un'istruzione macchina richiede un numero intero di cicli macchina, variabile a seconda del tipo di istruzione.

Velocità del microprocessore



- La velocità di elaborazione di un processore dipende dalla frequenza del clock.
 - I processori attuali hanno valori di frequenza di clock che varia tra gli 8 MHz ed i 3000 MHz.

Memoria



- Viene utilizzata per conservare dati e programmi.
- Si suddivide in:
 - **Memoria di lavoro** (memoria principale).
 - Memoria in grado di conservare dinamicamente dati e programmi che il processore sta utilizzando.
 - RAM (Random Access Memory)
 - ROM (Read Only Memory)
 - **Memoria magazzino** (memoria di massa).

Organizzazione della memoria principale (cont.)



- Ogni locazione ha la stessa dimensione
 - 8, 16, 32, o 64 bit.
- Ogni locazione viene anche detta **parola (word)** e costituisce un blocco unico.
- Le uniche operazioni che si effettuano sulla memoria sono **lettura** e **scrittura**.
- Una locazione di memoria può contenere dati o istruzioni.

Organizzazione della memoria principale



- La memoria è organizzata funzionalmente in **celle** indipendenti.
- Ad ogni **cella** è associato un **indirizzo**
 - cioè, un numero progressivo a partire da 0.

0	00101101
1	10011001
2	10010110
3	
4	
5	
6	
7	
8	

Indirizzi di memoria



- L'indirizzo di una locazione di memoria è un numero intero codificato in binario.
 - Ogni computer utilizza un numero di bit costante per rappresentare gli indirizzi.
- Maggiore è il numero di bit utilizzati, maggiore sarà il numero di locazioni indirizzabili: **Spazio di indirizzamento**.

Indirizzi di memoria (cont.)



- Se l'elaboratore utilizza:
 - 16 bit per l'indirizzo, la memoria conterrà fino a 65.536 locazioni (cioè 64 KB di memoria).
 - 32 bit per l'indirizzo, la memoria potrà contenere fino a 4.294.967.296 locazioni (cioè 4 GB di memoria).

Ciclo della CPU



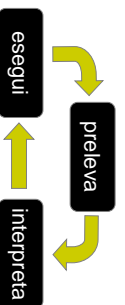
- Programma e dati sono caricati in memoria.
- Ciclo della CPU
 - 1) La CU *preleva* un'istruzione dalla memoria.
 - 2) L'istruzione viene *decodificata*.
 - 3) L'istruzione viene *eseguita*.
 - 4) La CU passa all'istruzione successiva e cioè al punto 1).

Ciclo della CPU



- Il processore esegue in continuazione il seguente ciclo

- *fetch* (preleva)
- *decode* (interpreta)
- *execute* (esegui).



Ciclo della CPU



- Casi particolari

- Avvio
- Quale istruzione eseguire per prima?
- Assumiamo PC=1
- Arresto
- Istruzione HALT (o HLT) per arrestare il ciclo

Ciclo della CPU



1. La CU *preleva* l'istruzione che si trova all'indirizzo di memoria contenuto nel PC e la trasferisce nell'IR (fase di *fetch*).
2. L'istruzione in IR viene *decodificata* ed *eseguita*.
 - L'esecuzione consiste nell'invio da parte della CU di opportuni comandi all'unità interessata:
 - Calcoli → ALU,
 - Acquisizione/Stampa → Unità di I/O,
 - Trasferimento dati → Memoria.
3. La CU *incrementa* opportunamente il PC ed il processo si ripete passando all'istruzione successiva in memoria (si torna al passo 1).

Bus di sistema



- Connette le varie unità e permette loro di comunicare
- Distinzione tra
 - **master** (controllore)
 - **slave** (controllato)
- **Molteplici** (sotto)bus:
 - **bus di dati** (trasferisce dati)
 - **bus di indirizzi** (trasferisce indirizzi)
 - **bus di controlli** (trasferisce comandi della CPU e lo stato delle unità)

Dispositivi di Input/Output



- I dispositivi di I/O, detti anche *periferiche*, permettono di realizzare l'interazione uomo/macchina.
- La loro funzione principale è quella di consentire l'ingresso e l'uscita delle informazioni elaborate.
- Il controllo è eseguito da un software di gestione, detto "*driver*", che ne gestisce le funzionalità.

Dispositivi di I/O



- Terminali.
- Tastiera.
- Video.
- Dispositivi di puntamento.
 - Mouse.
 - Trackball.
 - Schermo sensibile (Touch Screen).
- Stampanti.
 - Ad aghi.
 - Ink jet.
 - Laser.
- Scanner, modem, plotter, scheda audio...

Interfacce dispositivi I/O



- Per comunicare con le periferiche si usano processori dedicati: **interfacce**
- Le interfacce sono dotate di unità di controllo proprie e di registri
 - **Registro dati della periferica (PDR)**
 - **Registro comando della periferica (PCR)**
 - **Registro di stato della periferica (PSR)**

Accesso diretto alla memoria



- Direct memory access (DMA) permette ad alcuni dispositivi di accedere in lettura/scrittura alla memoria di sistema senza usare la CPU.
 - Molto più efficiente

Random Access Memory (RAM)



- È una **memoria volatile**
 - senza alimentazione i dati vengono persi
- Il **tempo di accesso** è il tempo necessario per leggere o scrivere l'informazione in una **parola**.
- **Accesso casuale**: ogni cella ha lo stesso tempo di accesso delle altre (indipendentemente dall'indirizzo).

Caratteristiche della RAM



- La **dimensione** di una RAM varia a seconda del tipo di computer e viene espressa in MB.
 - Le dimensioni tipiche della RAM di un *Personal computer* vanno da 256 MB ad alcuni GB.
 - La RAM di un *Server* è generalmente maggiore
- **Veloce**.
 - Il tempo di accesso è dell'ordine di poche decine di nano-secondi (10^{-9} sec).
- **Costosa** (~10 EUROcent/MB).

Read Only Memory (ROM)



- È una memoria di *sola lettura* che viene scritta direttamente dal produttore del computer su circuiti appositi.
- Viene utilizzata per contenere le informazioni di *inizializzazione* usate ogni volta che si accende il elaboratore.
- Le istruzioni contenute nella ROM dei PC formano il BIOS (Basic Input/Output System)
 - Diagnostica iniziale
 - Caricamento sistema operativo
 - Interfaccia software verso l'hardware e le periferiche I/O
- EEPROM (Electrically Erasable and Programmable ROM)
- EPROM (Erasable Programmable Read Only Memory) ultravoletti

Inizializzazione



- Cosa succede all'accensione di un calcolatore?
- Nel PC viene forzato l'indirizzo della cella di memoria ove inizia il primo programma da eseguire (programma di *bootstrap*).
 - Questo programma iniziale risiede in ROM.
 - Non è modificabile!!!!

Memoria Cache



- Per migliorare le prestazioni di un computer si inserisce una *memoria intermedia* tra CPU e memoria centrale (RAM) detta **Cache**:
 - In genere è interna al processore.
 - Più veloce della RAM: $T_{Cache} \sim 1/5 T_{RAM}$
 - Molto più costosa della RAM

Memoria Cache (cont.)

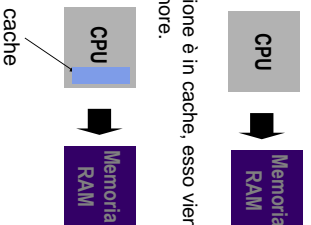


- I dati e le istruzioni più frequentemente richiesti vengono memorizzati nella *cache*, in modo da diminuire il tempo di accesso ed aumentare quindi le prestazioni medie.
- Diventa cruciale il metodo per selezionare i dati e le istruzioni da inserire nella cache.
- Dimensioni tipiche:
 - da **256 KB** a qualche **MB** di cache.

Memoria Cache (cont.)



- Senza cache:
 - Tempo di accesso = tempo di accesso alla memoria.
- Con cache:
 - Se il dato/istruzione è in cache, esso viene prelevato in un tempo minore.



Memoria secondaria



- La *memoria secondaria* (o di massa) è utilizzata per memorizzare in modo *non volatile* le informazioni.
 - Generalmente di *grande capacità* ed è costituita da dischi magnetici, CD-ROM, DVD, nastri, ...
 - Quando si vuole eseguire un programma, questo viene caricato dalla memoria di massa alla memoria principale.

Memoria secondaria (cont.)



- Due tecnologie possibili:
 - **Magnetica**
 - **Dischi magnetici** (Floppy-disk, Hard-disk, Disk-pack)
 - Accesso casuale
 - **Nastri magnetici**
 - Accesso sequenziale (legato alla posizione del dato)
 - **Ottica**
 - **CD-ROM, DVD**
 - Accesso casuale
- Futuro: memorie olografiche?
 - Dati memorizzati su un volume, non una superficie
 - Maggiore capacità e velocità

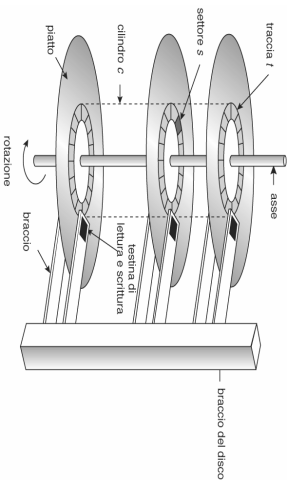
Organizzazione fisica dei dati nei dischi



- I dischi sono suddivisi in anelli concentrici (*tracce*)
- Le tracce sono suddivise in **settori**.
- I settori sono raggruppati in blocchi
- La suddivisione di un disco in tracce e settori viene indicata con il termine "*formattazione*".

Schema hard disk

- (dal Silberchatz)



Organizzazione fisica dei dati nei dischi (cont.)

- Il *tempo di accesso* alle informazioni (access time) dipende da tre fattori fondamentali:
 - **Seek time.**
 - La testina si sposta in senso radiale fino a raggiungere la traccia desiderata.
 - **Latency time.**
 - Il settore desiderato passa sotto la testina.
 - La velocità di rotazione è espressa in *rpm* (round per minute).
 - **Transfer time.**
 - Tempo di lettura vero e proprio.

Dischi ottici

- Un disco CD-ROM (memoria ottica) ha la capacità di circa 640 MB.
- Con i moderni DVD la capacità è aumentata parecchio (da 4,7 a 17 GB).
 - Nei DVD la tecnica usata è quella del multistrato.
 - Inoltre, per unità di superficie, è possibile immagazzinare molti più dati grazie alla maggior precisione del fascio laser.
- Standard successore DVD: Blue-ray vs HD-DVD
- Futuro: HVD (tecn. Olografica, circa 4 TB)

Organizzazione fisica dei dati nei dischi (cont.)

- Il disco è in rotazione
- Per effettuare un'operazione di lettura (scrittura) su un blocco è necessario che la testina raggiunga il blocco desiderato.
 - Single-sided.
 - Double-sided (doppia faccia, ho bisogno di due testine)

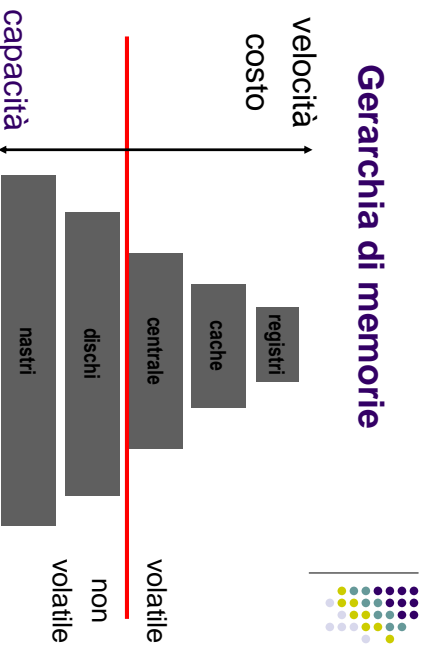
Dischi magnetici

- Hanno una grande capacità di memorizzazione.
 - I floppy-disks hanno una capacità di 1,44 MB.
 - Un disco fisso in genere ha la capacità di decine/centinaia di GB.
- Sono molto più lenti delle memorie primarie.
 - Tempo di accesso nell'ordine dei milli-secondi, 10^{-3} sec.
- Sono molto più economici

Nastri magnetici.

- Sono stati molto utilizzati nei primi calcolatori.
- Oggi vengono soprattutto utilizzati come unità di *backup*, data la loro grande capacità di memorizzazione ed economicità.

Gerarchia di memorie



Prestazioni di un elaboratore

- Valutazione molto difficile perchè dipende dal:
 - programma eseguito,
 - tipo di dati,
 - architettura dell'elaboratore (istruzioni macchina, cache, ...).

Classi di elaboratori

- Personal Computers.
- Portatili (Laptop o Notebook, PalmTop, PenPC).
- Micro e Mini computers: un sistema di elaborazione di medie dimensioni in grado di eseguire sia il multitasking che la multiprogrammazione. Utilizzato da piccole aziende.
- Workstations, computer stupidi, non dotati di capacità di calcolo autonoma
- Mainframes: un sistema di elaborazione complesso, sopporta migliaia di posti di lavoro (banche).
- Supercalcolatori (NASA,...): raggiungono altissime velocità
- Elaboratori paralleli (dotati di più processori).

Prestazioni di un elaboratore (cont.)

- MIPS - Millions of Instructions Per Second
 - Istruzione: un'operazione in codice macchina.
 - Poco indicativo.
- MFLOPS -Millions of Floating Operations Per Second
 - Operazione: operazione numerica su reali.
- Benchmark
 - Tempo richiesto per eseguire una suite di programmi applicativi con calcoli real/interi.
 - Viene restituito un indice (valore numerico) delle prestazioni con riferimento ad un particolare elaboratore.

Linguaggio macchina e assembly

- Linguaggio macchina
 - Linguaggio comprensibile direttamente dal processore della macchina (binario, sequenze di 0 e 1)
- Assembly
 - versione simbolica del linguaggio macchina
 - nomi delle operazioni e degli operandi sono indicati con codici simbolici
- Per 'parlare direttamente con la macchina' si usa l'assembly
 - Assembler
 - programmi che traducono il codice assembly in linguaggio macchina

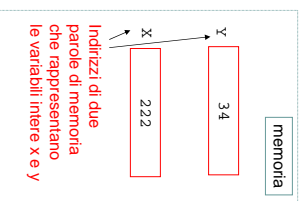
Linguaggio macchina e assembly

- Alcune istruzioni in linguaggio assembly
 - Lettura/scrittura dalla memoria
 - LOAD, STORE
 - Input / output
 - IN, OUT
 - Aritmetiche
 - ADD, SUB
 - Logiche
 - AND, OR, NOT
 - Salto
 - JUMP, JZ, JNZ
 - Altre
 - NOP, HALT
- Operandi
 - Variano in numero e tipo (valore, indirizzo di memoria, registro)

Linguaggio macchina e assembly



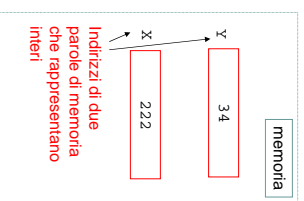
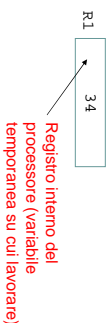
- Esempio $x:=y+2$
LOAD Y, R1 *Legge il valore in Y e lo scrive in R1*
ADD 2, R1
STORE R1, X



Linguaggio macchina e assembly



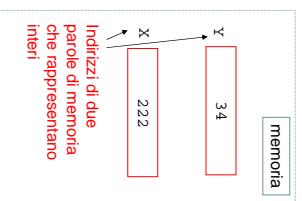
- Esempio $x:=y+2$
LOAD Y, R1
ADD 2, R1 *Aggiunge 2 a R1*
STORE R1, X



Linguaggio macchina e assembly



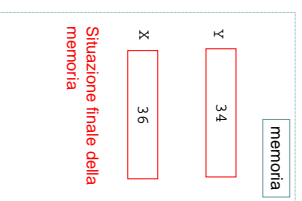
- Esempio $x:=y+2$
LOAD Y, R1
ADD 2, R1
STORE R1, X



Linguaggio macchina e assembly



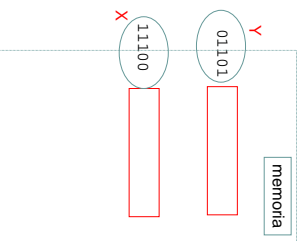
- Esempio $x:=y+2$
LOAD Y, R1
ADD 2, R1
STORE R1, X



Linguaggio macchina e assembly



- Traduzione binaria (in linguaggio macchina) di
LOAD Y, R1
ADD 2, R1
STORE R1, X
- Prima di tutto dobbiamo decidere quale è il vero indirizzo di X e Y



Linguaggio macchina e assembly



- Codifica binaria di
LOAD 01101, R1
ADD 2, R1
STORE R1, 11100
- Ogni operazione si codifica secondo un formato stabilito da chi costruisce l'hw
- un formato semplificato è

Codice operativo	Modo 1	Op1	Modo 2	Op2
------------------	--------	-----	--------	-----

Linguaggio macchina e assembly



- Vediamo i vari campi del formato :

Codice operativo	Modo1	Op1	Modo2	Op2
------------------	-------	-----	-------	-----

È la codifica dell'operazione da eseguire
es:

```
ADD    0001
LOAD  0110
STORE 0111
```

Linguaggio macchina e assembly



- Tipicamente la codifica di una istruzione è lunga come una parola o multipli della parola per poterla leggere dalla memoria in modo più efficiente :
 - es : con parole di 4 byte (32 bit)

Codice operativo	Modo1	Op1	Modo2	Op2
4bit	2bit	12bit	2bit	12bit

Linguaggio macchina e assembly



- Codifica binaria di
LOAD 01101, R1
ADD 2, R1
STORE R1, 11100

```
MODI
00 registro
01 memoria
10 immediato
CODICI OPERATIVI
ADD 0001
LOAD 0110
STORE 0111
```

Codice operativo	Modo1	Op1	Modo2	Op2
4bit	2bit	12bit	2bit	12bit
0110	01	01101	00	00001
0001	10	00010	00	00001
0111	00	00001	01	11100

load
add
store

Linguaggio macchina e assembly



- Vediamo i vari campi del formato :

Codice operativo	Modo1	Op1	Modo2	Op2
------------------	-------	-----	-------	-----

È la codifica primo operando, **MODO1**
dice a cosa si riferisce **OP1**
es:

```
se MODO1 = 00 1'operando è nel registro
                interno del processore
                (e Op1 è il numero del registro)
se MODO1 = 01 1'operando è in memoria
                (e Op1 è 1'indirizzo)
se MODO1 = 10 1'operando è immediato, cioè
                Op1 è direttamente il valore
                dell'operando
```

Lo stesso vale per il secondo operando!

Linguaggio macchina e assembly



- Problema
 - es : con 12 bit posso indirizzare 'solo' 2¹² parole di memoria centrale (RAM) !

Codice operativo	Modo1	Op1	Modo2	Op2
4bit	2bit	12bit	2bit	12bit

Cioè posso avere al massimo una RAM di 4K parole ...
se ne ho di più (oggi si arriva tranquillamente a 512K
e siamo sempre in aumento ...) devo inventarmi
codifiche diverse

Simulatori



- Lista simulatori di vario tipo e complessità
<http://www.sosresearch.org/caale/caalesimulators.html>
- xComputer
<http://math.hws.edu/TMCM/java/abs/xComputerLab1.html>

Little Man Computer
http://www.herts.ac.uk/lms/lidu_temp/lidu/projects/lmms/

Little Man Computer: codifica istruzioni



- | | |
|--------------|------------------------|
| • 1 Load | • 7 Halt |
| • 2 Store | • 800 Skip If Negative |
| • 3 Add | • 801 Skip If Zero |
| • 4 Subtract | • 802 Skip If Positive |
| • 5 Input | • 9 Jump |
| • 6 Output | |

Little Man Computer



Specifica	Assembly	Linguaggio macchina PMC
1. Leggi Y	1. IN Y	1. 500
2. Leggi X	2. IN X	2. 299
3. $Y := X + Y$	3. 1. LOAD Y 2. ADD X 3. STORE Y	2. 1. 500 2. 298
4. Stampa Y	4. OUT Y	3. 1. 199 2. 398 3. 299
	5. HALT	4. 1. 199 (inutile) 2. 600

- Indirizzo di X è 98
- Indirizzo di Y è 99

5. 700

Credits



- Parte del materiale contenuto in queste slides è frutto della revisione e della rielaborazione di materiale trovato in rete.
 - R. Giugno – Unict
 - S. Antonelli – Unipi
 - Pollicriti - Uniuud