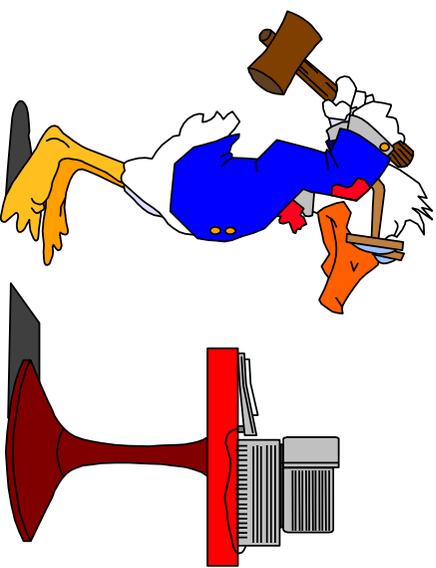


## La struttura del calcolatore

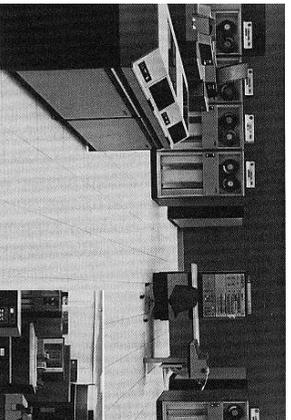
- La struttura del calcolatore
  - L'architettura a BUS
  - La memoria principale
  - La CPU
  - Il BUS
  - La memoria secondaria
  - I dispositivi di I/O
- Linguaggio macchina e assembler
- Il sistema operativo



1

## Il mercato informatico nel 1964

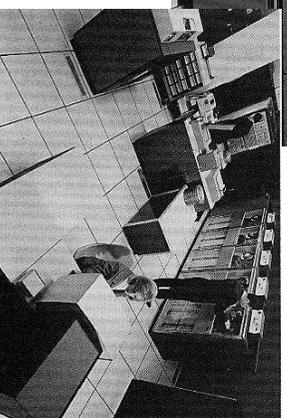
IBM S/360 Mod 40  
1.6 MHz 32KB-256KB  
225.000\$



IBM S/360 Mod 50  
2.0 MHz 128KB-256KB  
550.000\$



IBM S/360 Mod 65  
5.0 MHz 256KB-1MB  
1.200.000\$



2

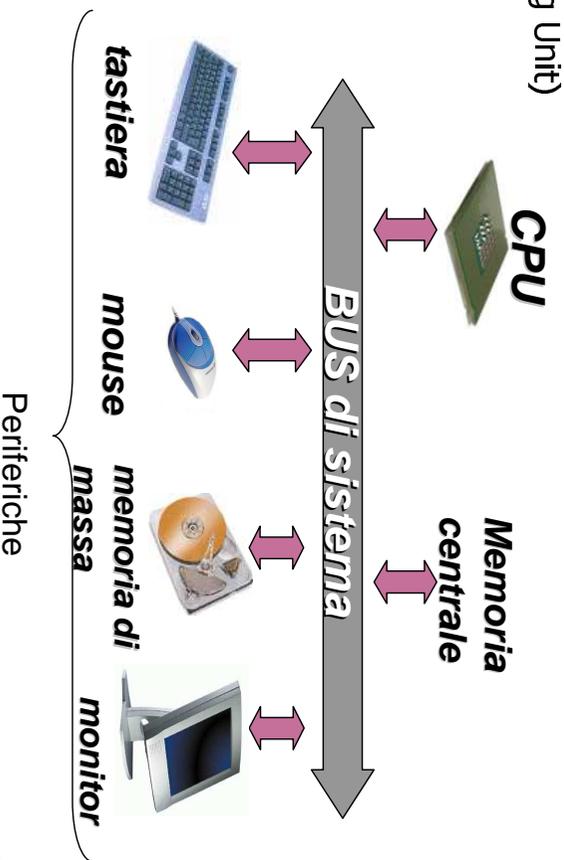
# La struttura del calcolatore

Slides realizzare a partire da materiale  
di Monica Bianchini e Leonardo Rigutini

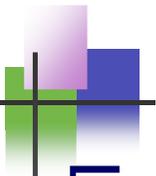
3

## L'architettura di Von Neuman

- Tutti i calcolatori attuali utilizzano l'architettura di Von Neuman costituita da 4 componenti
  - CPU (Central Processing Unit)
  - Memoria centrale
  - Il Bus di sistema
  - Le periferiche



4



# La memoria centrale

## Caratteristiche

- Vi si memorizza il “materiale di lavoro” su cui l’elaboratore opera:  
**Dati ed istruzioni**
- E’ una memoria volatile, ovvero i dati memorizzati vanno persi in caso di assenza di corrente: è necessario
  - caricare l’informazione dalla memoria di massa prima che possa essere utilizzata
  - salvarla nella memoria di massa se essa dovrà essere disponibile in un secondo momento
- è in genere di dimensioni ridotte a causa del maggiore costo
  - Più veloce → maggiore costo

5



# La memoria centrale

- Concettualmente è una sequenza di celle di memoria:
  - Le celle hanno comunque lunghezza multipla della potenze del 2 di un byte: 8,16,32, 64 bit (1,2,4,8 bytes)
  - Le celle di memoria di un calcolatore hanno tutte la stessa capacità, ma in calcolatori diversi possono avere dimensioni differenti
  - Il contenuto delle celle sono sequenze di bit che hanno significati diversi a seconda dei casi: dati, istruzioni, indirizzi ecc...

6

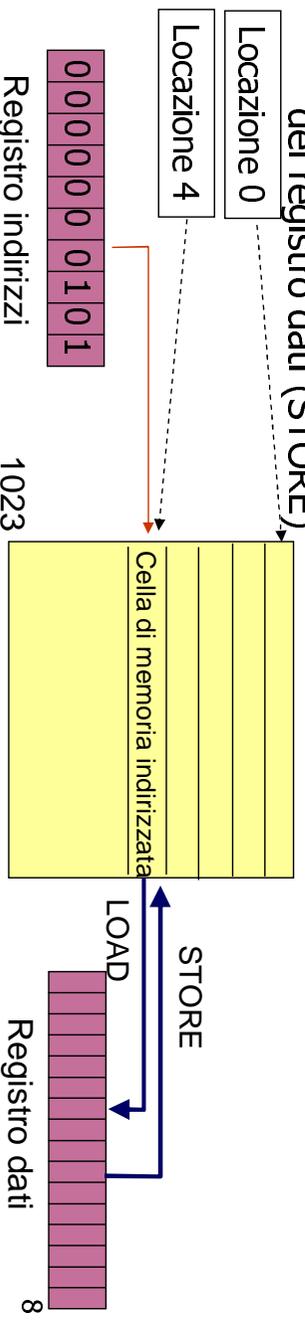
# Indirizzamento della memoria

- Ogni cella ha un suo **indirizzo**
- L'**indirizzo** è la sua **posizione relativa** rispetto alla prima cella della memoria centrale
- Ogni indirizzo è una parola binaria di k bit:
  - Quindi solamente  $2^k$  celle di memoria possono essere indirizzate
- Lo **spazio degli indirizzi** è il range di indirizzi che possono essere utilizzati dal calcolatore:
  - Con 16 bit →  $2^{16}$  (65,536) celle di memoria (64KB, indirizzando i byte)
  - Con 32 bit →  $2^{32}$  (4,294,967,296) celle di memoria (4GB)
  - Con 64 bit →  $2^{64}$  ( $18 \times 10^{18}$ ) celle di memoria, (18 milioni di tera Byte)

7

## Letture e scrittura

- La CPU usa coppie di **registri** per accedere la memoria:
  - uno per i dati e uno per l'indirizzo
- **Letture dalla memoria:**
  - La CPU memorizza nel registro dati il contenuto della cella indirizzata dal registro indirizzi (LOAD)
- **Scrittura in memoria:**
  - La CPU memorizza nella cella indirizzata dal registro indirizzi il contenuto del registro dati (STORE)



# ROM e BIOS

## RAM e ROM

- **RAM** (Random Acces Memory) è una memoria volatile con cui si realizza la normale memoria centrale
- **ROM** (Read Only Memory) è una memoria non volatile e non riscrivibile in cui si memorizza il **BIOS** .....
- nota che i BIOS delle macchine attuali sono aggiornabili !!!

## Il BIOS

- Il BIOS è il primo software caricato all'avvio del PC
- Permette al calcolatore di **riconoscere e configurare le componenti hardware** e di **avviare il sistema operativo**
- Risiede in memoria principale, ma è memorizzato su memoria non volatile
- Perché un calcolatore ha bisogno del BIOS e non può lanciare subito il sistema operativo ?

9

# La CPU

- La **CPU**, **Central Processing Unit**, è l'unità di elaborazione centrale
- Le funzioni della CPU vengono integrate in un componente chiamato **microprocessore**



- La CPU lavora a **N**GHz: segue un ritmo, definito dall'orologio del sistema, di **N** miliardi di impulsi al secondo; questi impulsi determinano la velocità del computer
  - es., il microprocessore **Intel Pentium IV** con un clock a 3 GHz è temporizzato da tre miliardi di impulsi al secondo

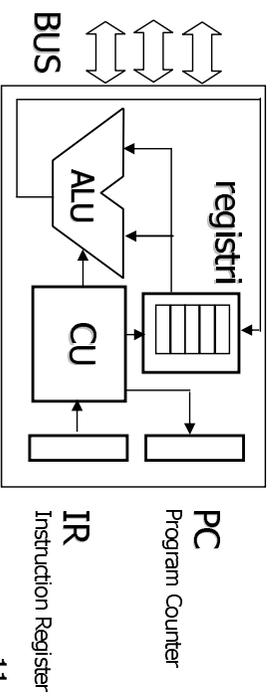
10

## La CPU II

- La **funzione della CPU** è eseguire i programmi contenuti nella memoria centrale:
  - Prelievo dell'istruzione
  - Decodifica dell'istruzione
  - Esecuzione dell'istruzione

- La **CPU è formata da varie unità funzionali**:

- Unità di controllo
- Registri
- Unità Aritmetico-Logica
- Orologio di sistema



11

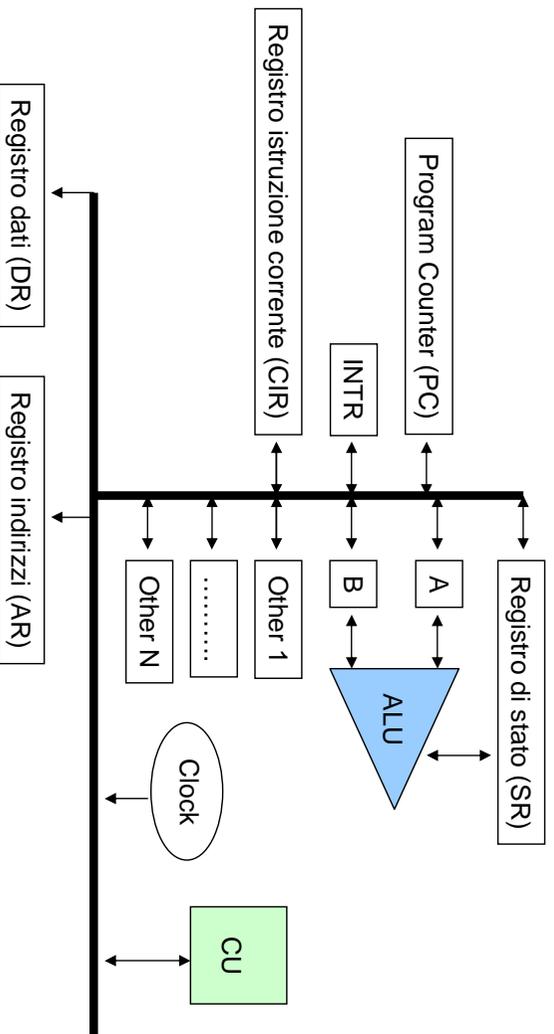
## Unità funzionali della CPU

- **Unità di controllo (CU):**
  - È responsabile del prelievo e della decodifica delle istruzioni
  - Si occupa inoltre di inviare i segnali di controllo sul bus per i trasferimenti o le elaborazioni necessarie per l'esecuzione dell'istruzione decodificata
- **Aritmetic and Logic Unit (ALU):**
  - Realizza le operazioni aritmetico/logiche
  - Lavora sui dati presenti nei registri interni
- **Orologio di sistema (clock):**
  - Sincronizza le operazioni rispetto ad una data frequenza.

12

# Registri

- Sono locazioni di memoria ad accesso molto veloce contenute nella CPU
- Memorizzano dati/istruzioni su cui si sta lavorando



13

# Registri II

- Registro istruzione corrente (CIR)  
Contiene il codice che identifica l'istruzione in esecuzione in quel momento (ADD, JUMP, ...)
- Contatore di programma (PC)  
Contiene l'indirizzo in memoria della prossima istruzione del programma in esecuzione
- Registro dati (DR) e registro indirizzi (AR)  
Contengono il dato e l'indirizzo di memoria su cui effettuare l'operazione corrente
- Registro interruzioni (INTR)  
Contiene informazioni relative allo stato di funzionamento delle periferiche. Se una periferica deve contattare la CPU, segnala questo fatto nel registro INTR. Es. fine di una stampa

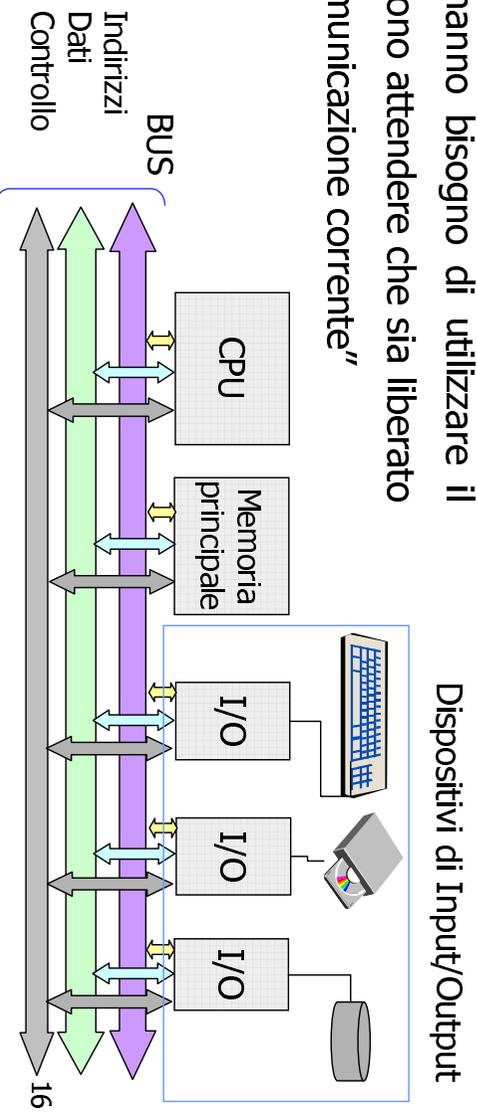
14

## Registri III

- **Registri di lavoro**  
Molto simili ad una memoria in miniatura in cui la CPU memorizza le istruzioni ed i dati utilizzati più di recente per un accesso più rapido in futuro
- **Registri per la ALU**  
Contengono gli operandi ed il risultato delle elaborazioni svolte dalla ALU
- **Registro di stato (SR)**  
E' un particolare registro in cui ogni bit assume un significato diverso, relativo al risultato delle operazioni svolte dalla ALU:
  - **Bit di carry**, indica il riporto in una operazione aritmetica
  - **Bit zero**, indica la presenza di un valore nullo nel registro A
  - **Bit di segno**, riporta il segno del risultato di un'operazione aritmetica
  - **Bit di overflow**, rileva la condizione di overflow che si verifica quando il risultato dell'ultima operazione della ALU supera il valore rappresentabile nel registro A<sup>15</sup>

## Il bus

- E' costituito da un **insieme di connessioni elementari, o linee**, lungo le quali viene trasferita informazione
- **Collega fra loro la CPU, la memoria e le varie periferiche di ingresso/uscita**
- Può collegare **solamente due unità**. Se le altre hanno bisogno di utilizzare il BUS, devono attendere che sia liberato dalla "comunicazione corrente"



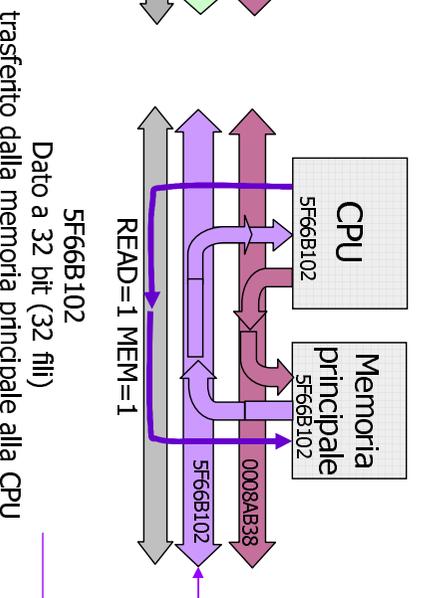
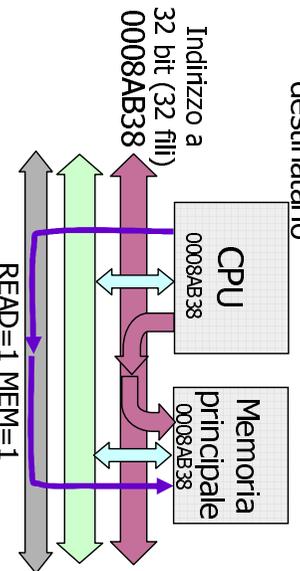
## Il bus II

- Le linee del BUS vengono divise in tre categorie a seconda della loro funzione: **bus dati**, **bus indirizzi**, **bus controllo**
- Il bus dati**
  - Trasferisce i dati
  - Ed es. nell'operazione STORE di memorizzazione dati nella memoria centrale, i dati transitano nel bus dati in direzione CPU → Memoria
- Il bus indirizzi**
  - Indica l'indirizzo interessato dall'operazione
  - Ad es. trasferisce l'indirizzo della memoria centrale in una operazione di STORE
- Il bus controllo**
  - Trasferisce il codice dell'operazione da eseguire (lettura, scrittura, etc)
  - Ad es., in una operazione di STORE, la memoria deve essere avvertita di memorizzare il dato nella locazione corretta

17

## Il bus III

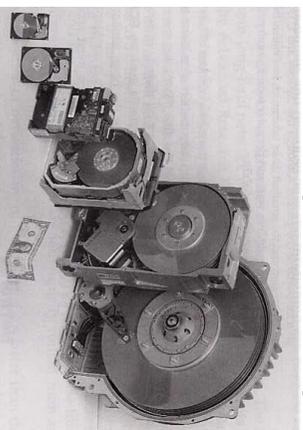
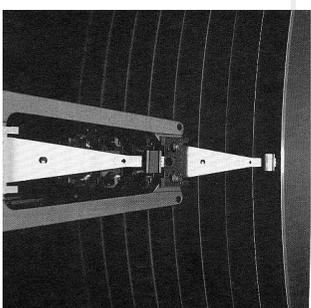
- Esempio:** il bus è utilizzato per trasferire dati fra due unità funzionali
  - L'unità che inizia il trasferimento (in genere la CPU) fornisce l'indirizzo, che individua univocamente il dato, sulle linee del **bus indirizzi**, e configura le linee del **bus controllo**, inviando un comando al dispositivo che contiene il dato (es. READ)
  - Il dato da trasferire è reso disponibile sul **bus dati** e viene ricopiato nel dispositivo destinatario



18

## La memoria secondaria

- Esistono diversi dispositivi di memoria secondaria:
  - **dischi magnetici** (hard disk e floppy), **dischi ottici** (CD-R, CD-RW, DVD), **nastri magnetici**
  - **Memoria non volatile** ad alta capacità (ad es. penne USB)



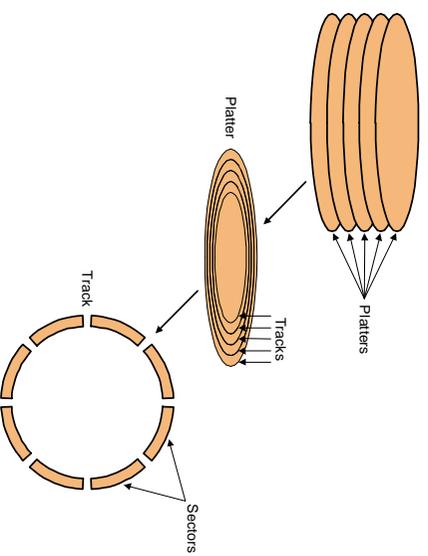
Dischi magnetici

- Il disco fisso è costituito da uno o più **piatti** ricoperti di materiale magnetico su entrambe le facce
- Ciascuna superficie è associata ad una o più **testine** di lettura/scrittura che si muovono radialmente per leggere/scrivere l'informazione

19

## I dischi magnetici

- L'informazione è disposta sul disco in **tracce**, ovvero in **cerchi concentrici**
- Le tracce sono divise in **settori**
- Il numero di bit contenuti in una traccia è dell'ordine delle centinaia di migliaia (512/4096 byte a settore)



- sul disco, la testina si deve **posizionare sulla traccia** che contiene il dato, ed attendere che il dato passi sotto di essa
- L'unirà di informazione **minima trasferibile** dal disco è un **setto**

20

## Accesso al disco

- Il tempo medio di accesso all'informazione memorizzata su disco è dato da

$$T = t_{\text{seek}} + t_{\text{lat}} + t_{\text{tr}}$$

- $t_{\text{seek}}$  è il **tempo di ricerca**, necessario per posizionare la testina sulla traccia che contiene l'informazione; dipende dall'ampiezza dello spostamento (dischi attuali, in media circa 4-8ms)
- $t_{\text{lat}}$  è il **tempo di latenza**, necessario perché l'informazione ricercata passi sotto la testina; dipende dalla velocità di rotazione dei dischi (dischi attuali, in media circa 2-7ms)
- $t_{\text{tr}}$  è il **tempo di trasferimento**; dipende dalla velocità di rotazione, dalla densità di registrazione e dalla quantità di informazione da trasferire. La velocità di trasferimento è spesso limitata dalla velocità delle interfacce e del BUS. (Usando ATA/100, 100MBs)

21

## Osservazione importante

- La velocità di accesso alla memoria principale e alla memoria di massa è molto diversa
- A seconda di dove si trovano i dati da elaborare, la velocità di un'operazione cambia totalmente
  - Nell'elaborazione di dati memorizzati su un disco (ad esempio la somma di numeri memorizzati in un file) **la maggior parte del tempo può essere speso nel leggere i dati da disco**
  - Un calcolatore aggiornatissimo con poca memoria principale può essere molto più lento di un vecchio calcolatore con molta memoria

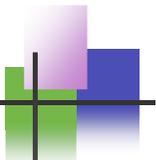
Accesso ai registri: stessa velocità del processore (0.5-2ns, clock 1-4GHz, )

Accesso alla cache: 1-4 volte più lenta dei registri

Accesso alla RAM: 5-10 volte più lente dei registri (5-7ns, clock 333-900MHz)

Accesso ai dischi: 1.000 volte più lente della RAM, circa 10 microsecondi

22



## I dispositivi di Input–Output

- I dispositivi di I/O sono molti vari
- Si possono considerare tre diverse caratteristiche:
- **Comportamento:** Input (solo lettura), output (solo scrittura), Input/Output (Lettura/Scrittura)
- **Partner:** uomo o macchina
- **Velocità del flusso dei dati:** quantità di dati nell'unità di tempo trasferiti da o verso la CPU o la memoria centrale

Tastiera	input	uomo	0.01	KB/s
Mouse	input	uomo	0.02	KB/s
Rete	input/output	macchina	10	MB/s
Hard Disk	storage	macchina	100	MB/s

23



## Le periferiche

- Le periferiche interagiscono con la CPU attraverso interfacce di I/O (input/output)
- Un periferica può essere di due tipi:
  - un “device stupido” che si limita ad eseguire un operazione una volta comandato (es. il monitor)
  - un “device intelligente”, in grado di elaborare a sua volta i dati inviati (es. le schede grafiche)
- Delegando molte operazioni a periferiche intelligenti, la CPU può guadagnare “tempo” per effettuare altre operazioni:
  - Es. il processore chiede alla scheda grafica di disegnare un rettangolo e di riempirlo con un dato colore inviandogli solo le coordinate e il colore desiderato

24

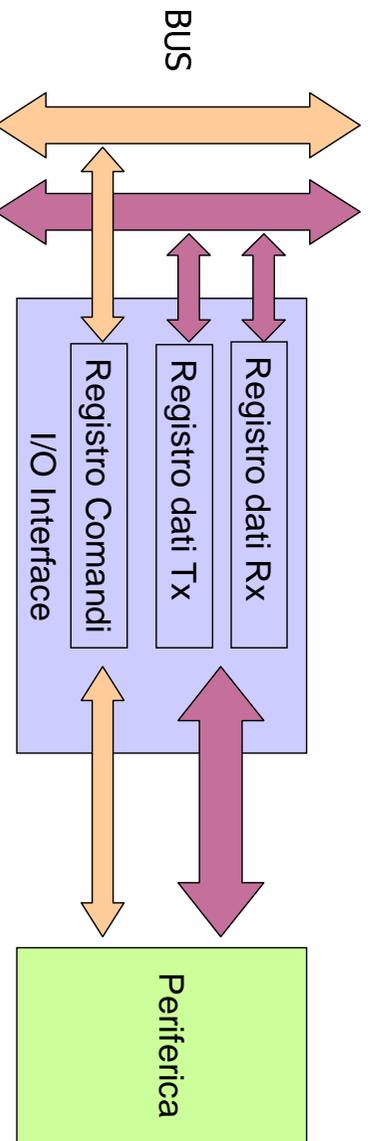
## Le interfacce I/O

- Le interfacce di I/O sono componenti che permettono di comunicare con le periferiche
- Le interfacce possono essere molto diverse tra loro:
  - Terminali, stampanti, hard disk, lettori ottici ecc....
- Comunque, **le interfacce mostrano alla CPU una periferica standard** (o quasi), in modo che la CPU non si debba preoccupare del modello di periferica ma solo del tipo di interfaccia a cui essa è "attaccata"

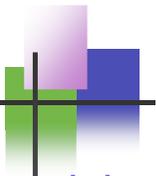
25

## Interfaccia standard

- Una interfaccia I/O standard è composta in genere dai seguenti elementi:
- Un **registro dati** della periferica (PDR), utilizzato per scambiare dati con la periferica.
  - Un **registro di comando** per la periferica (PCR), in cui è memorizzato il comando che la periferica stessa dovrà eseguire
  - Un **registro di stato** della periferica (PSR), che contiene le informazioni sullo stato della periferica (es. stampante pronta a ricevere, occupata o errore)



26



## I calcolatori moderni sono ancora più complessi

- La nostra descrizione è modo molto semplificata, ad esempio, i calcolatori moderni:
  - hanno bus diversi per accedere alla RAM e per accedere alle altre periferiche
  - Esistono più memorie centrali, dalle più veloci (cache di primo e secondo livello, ...) a quelle più lente (RAM)
  - La CPU incorpora moduli di cache e un'unità (coprocessore matematica) in grado di realizzare anche operazioni in virgola mobile complesse come i logaritmi e le potenze
  - La CPU incorpora più ALU
  - La CPU è in grado di eseguire più operazioni contemporaneamente attraverso meccanismi diversi (parallelismo, pipeline, ...)

27



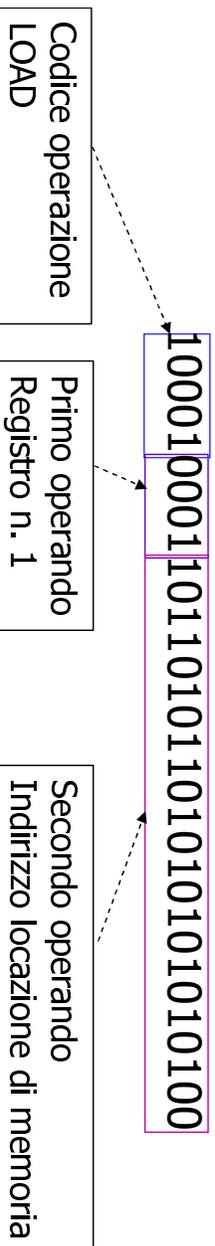
## Il linguaggio macchina

28

## Il set di istruzioni macchina (Instruction Set)

- Il calcolatore utilizza la rappresentazione binaria per le istruzioni ed i dati
- La CPU ha un **set di istruzioni** predefinito in fase di progetto:
  - Ogni istruzione è individuata da un codice operazione OP\_CODE
  - Alcune operazioni richiedono dati su cui operare e quindi hanno bisogno di operandi
- Le istruzioni macchina ed i dati sono quindi rappresentati da bytes:
  - Architetture a 8,16,32 o 64 bit

Esempio: Istruzione che carica nel registro 1 il contenuto di una cella di memoria



29

## Il linguaggio macchina

- Un programma in esecuzione è memorizzato nella memoria principale
- Un programma è una sequenza di numeri binari che codificano
  - le istruzioni eseguibili dall'unità centrale
  - Le variabili (i dati) su cui si opera
- Il registro PC individua l'istruzione (ne contiene l'indirizzo) che si sta eseguendo attualmente

```
000000001010000010000000000011000      ← PC
00000000100011100001100000100001
10001100011000100000000000000000
10001100111001000000000000000100
10101100111100100000000000000000
```

30

# Esempio di programma in linguaggio macchina

0100000000010000	leggi un valore in ingresso e ponilo nella cella numero 16 (variabile x)
0100000000010001	leggi un valore e ponilo nella cella numero 17 (variabile y)
0100000000010010	leggi un valore e ponilo nella cella numero 18 (variabile z)
0100000000010011	leggi un valore e ponilo nella cella numero 19 (variabile r)
0000000000010000	carica il registro A con il contenuto della cella 16
0001000000010001	carica il registro B con il contenuto della cella 17
0110000000000000	somma i contenuti dei registri A e B
0010000000010100	copia il contenuto del registro A nella cella 20 (risultato, variabile s)
0000000000010010	carica il registro A con il contenuto della cella 18
0001000000010011	carica il registro B con il contenuto della cella 19
0110000000000000	somma i contenuti dei registri A e B
0001000000010100	carica il registro B con il contenuto della cella 20
1000000000000000	moltiplica i contenuti dei registri A e B
0010000000010100	copia il contenuto del registro A nella cella numero 20
0101000000010100	scrivi in output il contenuto della cella numero 20
1101000000000000	arresta l'esecuzione (HALT)
.....	spazio per la variabile x (cella 16)
.....	spazio per la variabile y (cella 17)
.....	spazio per la variabile z (cella 18)
.....	spazio per la variabile r (cella 19)
.....	spazio per la variabile s (cella 20)

31

## Il set di istruzioni macchina

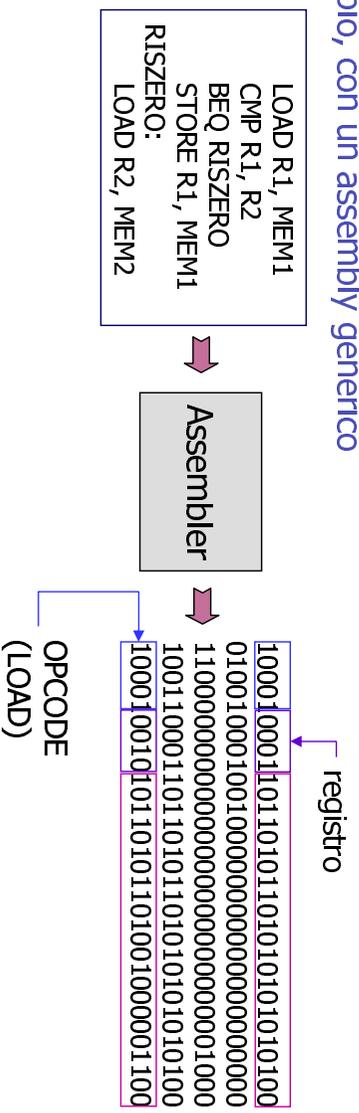
- L'insieme delle istruzioni eseguibili, e la relativa codifica, sono generalmente diverse per modelli diversi di processore (es. Intel Pentium, Power PC, Alpha)
- Le istruzioni possono essere codificate con un numero variabile di bit (es. Pentium) o con un numero fisso (es. MIPS – 32 bit)
- Le categorie di istruzioni normalmente disponibili sono:
  - ◆ **Trasferimento dati:** spostano dati (byte, word) tra registri, memoria principale e dispositivi di ingresso/uscita (I/O)
  - ◆ **Aritmetico-logiche:** eseguono i calcoli nell'ALU
  - ◆ **Salti (condizionati e incondizionati):** prendono decisioni e alterano la normale esecuzione sequenziale delle istruzioni

32

## Assembler

- Per facilitare la programmazione è stato definito il linguaggio **assembly**
- L'assembly impiega una notazione simbolica che è in stretta relazione con i codici in linguaggio macchina; il programma scritto in assembly è convertito automaticamente in linguaggio macchina per mezzo del programma traduttore, l'**assembler**

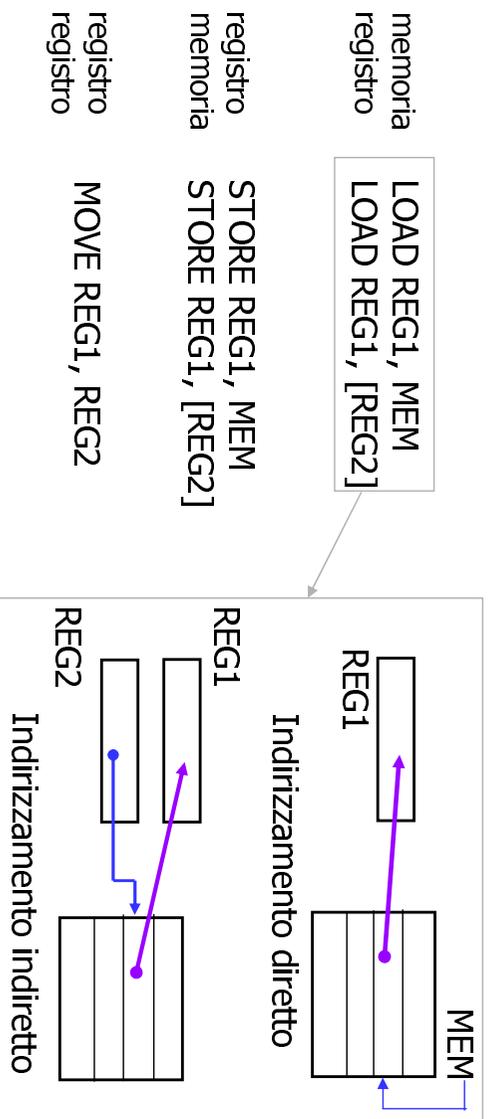
Esempio, con un assembly generico



33

## Trasferimento dei dati

- Le istruzioni di trasferimento dati permettono di copiare il valore di un dato fra registri o fra un registro e la memoria

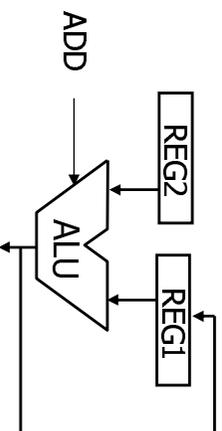


34

## Operazioni aritmetico-logiche

- Permettono di eseguire operazioni aritmetiche o logiche su due operandi

ADD REG1, REG2



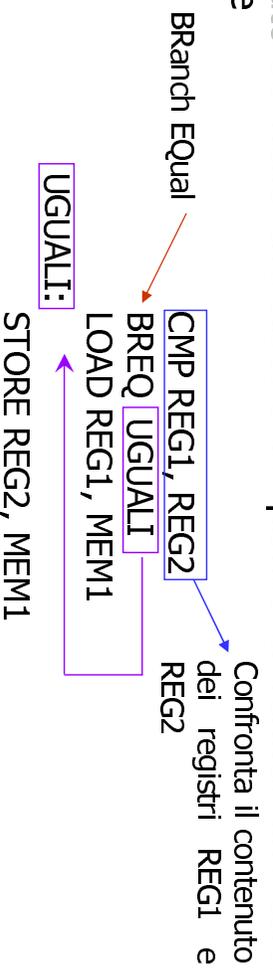
OR REG1, REG2

```
REG1 0010 1100 1100 0101 1100 1111 0101 0000
REG2 1010 1110 0000 0000 0010 0001 1000 0001
-----
REG1 1010 1110 1100 0101 1110 1111 1101 0001
```

35

## Istruzioni di salto

- Modificano il flusso di esecuzione del programma; il salto è **condizionato** se viene effettuato solo quando si verifica una certa condizione



- L'istruzione `CMP` confronta i due argomenti, assegnando valori particolari a bit di condizione che si trovano in un registro speciale della CPU
- L'istruzione di salto condizionato può essere basata su diverse condizioni: `BRNE` (branch not equal), `BRLE` (branch less equal), etc.
- L'istruzione di salto incondizionato, `BRANCH`, effettua sempre il salto

36

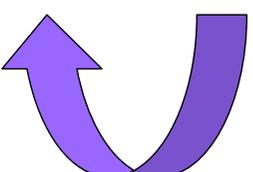
## Assembler e linguaggi ad alto livello

Il programma in assembler...

```
LOAD REG1, a
LOAD REG2, b
ADD REG1, REG2
LOAD REG3, c
LOAD REG4, d
ADD REG3, REG4
MULT REG1, REG3
STORE REG1, e
```

Corrisponde all'unica  
istruzione Java...

```
e=(a+b)*(c+d);
```



37

## Il sistema operativo

```
INTERLINK  EXE  177197  11-17-94  1:00P
XOPCOPY  EXE  31737  11-17-94  1:00P
TOIN      EXE  10279  11-17-94  4:09P
PKUNZIP   EXE  29378  4-03-95  1:00P
DRVLOCK   EXE  6501  11-17-94  1:00P
FIND      EXE  5814  11-17-94  1:00P
RAMSETUP  EXE  89649  11-17-94  1:00P
POWER     EXE  8806  11-17-94  1:00P
RCALC     EXE  22851  11-17-94  1:00P
NL$FLUNC  EXE  5609  11-17-94  1:00P
MEM        EXE  16231  11-17-94  1:00P
APPEND    EXE  7735  11-17-94  1:00P
SMARTDRV  EXE  44121  9-13-93  3:36P
ZIP        EXE  125964  9-07-93  8:42A
ZIPNOTE   EXE  22942  10-09-95  7:59P
UNZIP$FX  EXE  26331  10-09-95  7:59P
UNZIP     EXE  166332  11-17-94  12:00P
REXXDUMP  EXE  968  11-17-94  12:00P
CPSCHED   EXE  4946  11-17-94  1:00P
IBMPASP   EXE  158977  11-17-94  12:00P
IBMPDOST  EXE  164272  11-17-94  1:00P
PRMBOOST  EXE  2580199  bytes used
59 file(s)  113414144 bytes free
```

MS DOS 3.0



Mac OS 10.2

## I sistemi operativi sono moltissimi

- Ogni computer ha bisogno di un **sistema operativo**!!
- Un sistema operativo e' un software molto complesso: **contiene milioni di linee di codice**
- I PC
  - Microsoft Windows
  - Linus
  - Mac OS
  - ....
- I palmari
  - Palm OS
  - Windows CE
  - ....
- Mainframe
  - IBM AIX
  - z/OS
  - ...

39

## Il sistema operativo

- Il software può essere diviso in due grandi classi:
  - il **sistema operativo**: insieme di programmi di sistema, che gestiscono le funzionalità del calcolatore: memoria, capacità' di calcolo, periferiche, ...
  - le **applicazioni**: programmi che risolvono i problemi degli utenti
- **Lo scopo del sistema operativo**
  - gestire in maniera efficiente le risorse del calcolatore
  - implementare un'interfaccia utente user-friendly

- **Definizione:**

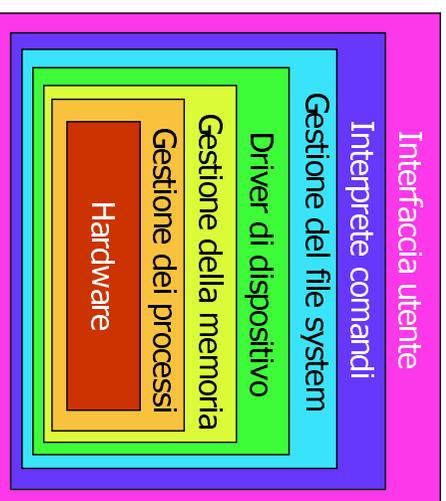
"Un sistema operativo è un programma che controlla l'esecuzione dei programmi applicativi ed agisce come interfaccia fra le applicazioni e l'hardware del calcolatore"

40

## Il sistema operativo come macchina virtuale II

Il SO e' realizzato a strati

- ogni strato realizza un insieme di funzionalita' e li mette a disposizione dello strato superiore
- tale organizzazione semplifica la realizzazione del SO e lo rende piu' modulare
- i tre strati piu' profondi sono chiamati **nucleo (kernel)** del SO: il nucleo influenza in modo fondamentale la sicurezza e la stabilita' del SO



## Compiti/strati del sistema operativo

Gestione dei processi

- Un **processo**
  - e' un programma in esecuzione
  - un processo evolve nel tempo, un programma e' statico
  - ci possono essere piu' processi corrispondenti allo stesso programma: ad esempio, si e' lanciato ie explorer due volte
  - in un sistema operativo ci possono essere piu' processi attivi contemporaneamente
- il SO deve garantire l'esecuzione concorrente di processi multipli, decidendo a quale di essi assegnare l'uso della/delle CPU
- Il SO **virtualizza** le CPU: l'utente ha l'impressione di avere una CPU tutta per se
- **Gestisce gli interrupt** provenienti dalle periferiche:
  - ad. es. il processo che esegue Word visualizza una pagina sullo schermo e poi si mette in attesa .... verra' risvegliato dal SO quando l'utente preme un tasto



## Compiti/strati del sistema operativo II

### Gestione della memoria

- Alloca la memoria primaria e la ripartisce fra i vari programmi che la richiedono
- Lo strato del gestore di memoria offre agli strati superiori una macchina virtuale in cui ciascun programma opera come se avesse a disposizione una memoria dedicata
- Simula una memoria principale più grande di quella disponibile scaricando e caricando da memoria secondaria (SWAP)

### Driver di dispositivo

- Sono responsabili delle operazioni di ingresso/uscita che coinvolgono le periferiche
- Ciascun driver è un modulo software dedicato a periferica specifica: ne conosce (e ne occulta) le caratteristiche hardware

43



## Compiti/strati del sistema operativo III

### Gestione del file system

#### ■ Il file system

- è il meccanismo con il quale i file sono immagazzinati e organizzati sui dispositivi di massa
  - I file sono tipicamente organizzati in un albero costituito da un insieme annidato di directory (cartelle in windows)
- Il SO fornisce all'utente un insieme di funzioni di alto livello per operare su file e directory, mascherando le operazioni realmente effettuate per allocare la memoria e per accedervi

- Tramite il file system, ciascun utente può organizzarsi la propria area di memoria e garantirne la protezione da accessi

44

## Compiti/strati del sistema operativo IV

### Gestione dell'interprete comandi

- L'interprete comandi consente all'utente di attivare le funzionalita' del SO
  - Eseguire un programma
  - Fermare un Programma
  - Creare, modificare, copiare ... file
  - Stampare un documento
  - Gestire gli utenti: inserire un nuovo utente, definirne I diritti di accesso, etc

### Gestione dell'interfaccia utente

- E' un'interfaccia con cui si possono inviare i comandi all'interprete
- I sistemi operativi moderni per i PC mettono a disposizione sia un'interfaccia grafica (GUI) che una testuale (shell comandi)

45

## Il sistema operativo

### Cosa sono ?

- Gli editori di testo (Word, notepad,...) → Applicazioni
- Programmi gestione degli utenti → Sistema operativo
- I programmi di visualizzazione stampanti → Sistema operativo
- Codice che permette di muovere le finestre → Sistema operativo
- Il contenuto di una finestra → Dipende.....
- La calcolatrice di windows → Applicazione
- MS Explorer → Applicazione
- Programma di gestione delle connessioni → Sistema operativo
- Server Web → Applicazione
- Gestione della memoria → Sistema operativo

### Applicazione o

### parte sistema operativo

Interfaccia + funzionalità del

46

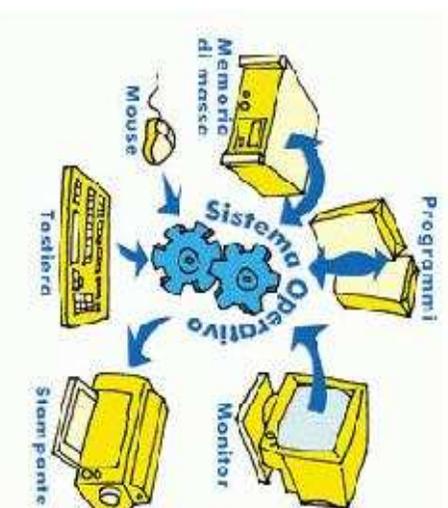
## Il sistema operativo come gestore delle risorse: un esempio

- Si consideri un ristorante con un capo-cuoco, i suoi aiutanti e I clienti:
  - I **clienti** scelgono un piatto dal menù
  - Un **cameriere** prende l'ordine e lo consegna al **capo-cuoco**
  - Il capo-cuoco riceve l'ordine e assegna uno o più aiutanti alla **preparazione del piatto**
  - Ogni **aiutante** si dedicherà alla preparazione di un piatto svolgendo diverse operazioni seguendo una **ricetta** e usando **piatti, pentole, etc** disponibili **in cucina**
  - Il capo-cuoco supervisiona la preparazione dei piatti e gestisce le risorse
- In questo esempio....
  - Il capo-cuoco è il **sistema operativo!**
  - I clienti sono gli **utenti**
  - Le ricette associate ai piatti sono i **programmi**
  - Il menù ed il cameriere costituiscono l'**interfaccia del sistema operativo**
  - Gli aiutanti sono le **CPU**
  - La preparazione di un piatto e' un **processo**
  - La cucina è il **computer**; pentole, fornelli, etc. sono le componenti **hardware**

47

## Il sistema operativo come gestore delle risorse: un esempio II

- **Problemi del capo-cuoco:**
  - Esecuzione fedele delle ricette
  - Allocazione efficiente delle risorse esistenti (aiutanti, fornelli, ingredienti, etc.)
  - Coordinamento dell'esecuzione delle ricette
- **Problemi del sistema operativo:**
  - Efficienza nell'uso delle risorse (processori, memoria, dischi, etc.)
  - Protezione nell'uso delle risorse
  - Coordinamento dei processi (I processi sono i programmi in esecuzione)



48

## Il sistema operativo come macchina virtuale

- Il Sistema Operativo (SO) rende virtuali le
  - caratteristiche dell'hardware, offrendo all'utente la visione di una macchina astratta più potente e più semplice da utilizzare di quella fisica
- In questa visione, un sistema operativo
  - ...nasconde a programmatori/utenti i dettagli dell'hardware e fornisce un'interfaccia facile da usare
  - ...agisce come intermediario tra programmatore/utente e hardware
- Il SO rende totalmente disponibile all'utente l'hardware del computer

