Calcolatori Elettronici

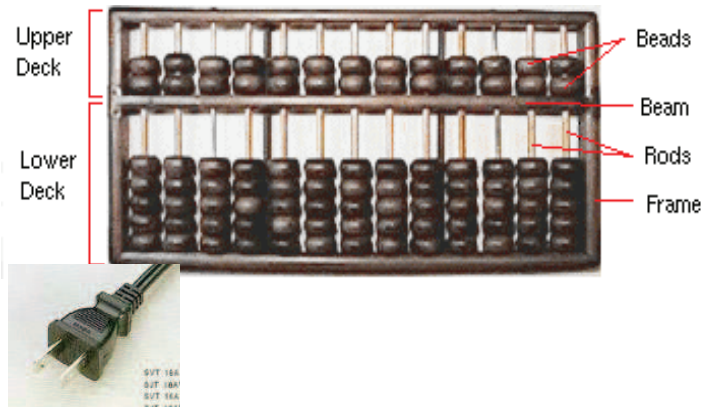
Introduzione

Prof. Francesco Lo Presti

Cosa e' un calcolatore?

- "A device that computes, especially a programmable electronic machine that performs high-speed mathematical or logical operations or that assembles, stores, correlates, or otherwise processes information"
  -- *The American Heritage Dictionary of the English Language*, 4th Edition, 2000

Cosa e' un calcolatore?



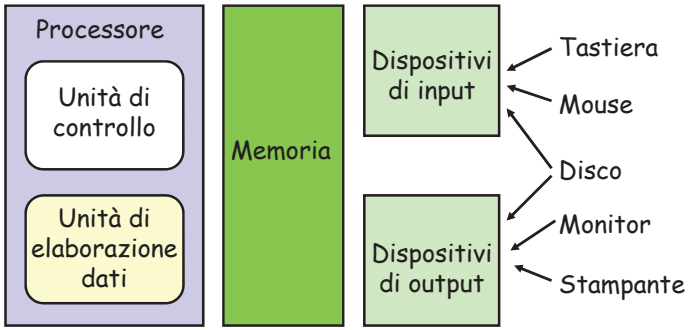
Calcolatore Elettronico

- Macchina per l'esecuzione automatica di algoritmi
  - General Purpose: puo' svolgere qualsiasi elaborazione di cui sia noto un algoritmo risolutivo
- Programma e' un algoritmo espresso in un linguaggio di programmazione
- Linguaggio macchina binario: il linguaggio direttamente eseguibile dal calcolatore
- Istruzione Macchina
  - operazione logico/aritmetica su dati o di trasferimento dati
  - codificata in binario
    - ✓ Esempio 10000110010100000 istruisce il calcolatore di effettuare una somma
- Programma eseguibile: un algoritmo espresso in linguaggio macchina

Calcolatore Elettronico

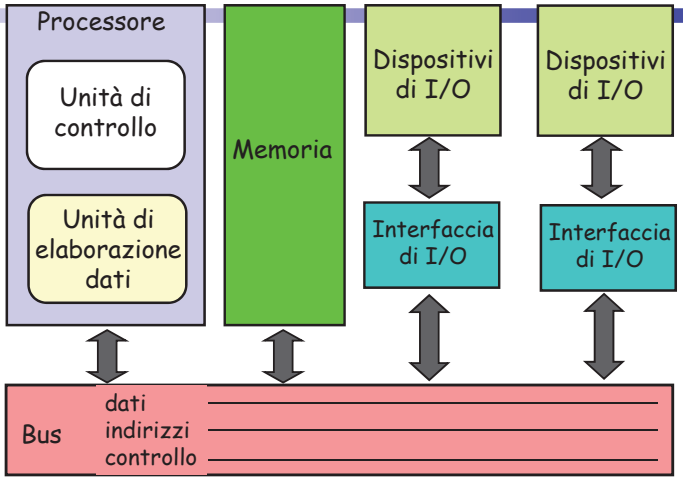
1. Accetta in ingresso informazioni codificate in forma digitale
2. Le elabora attraverso un programma memorizzato
3. Produce informazioni in uscita

Componenti principali di un calcolatore



- Processore o unità centrale di elaborazione (Central Processing Unit – CPU)
- Memoria principale (Main Memory – MM)
- Dispositivi di Input/Output (I/O) e loro interfacce

… e loro interconnessione



- Bus di sistema: collega i componenti principali di un calcolatore (linee dati, indirizzo, controllo)

Processore - Central Processing Unit (CPU)

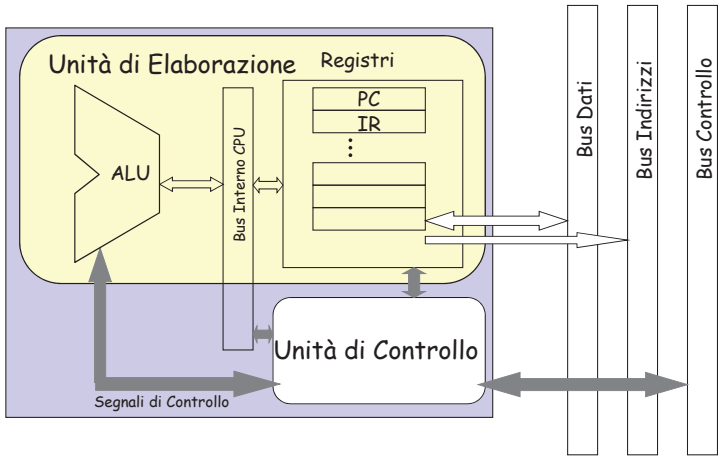
- Provvede all'esecuzione delle istruzioni macchina
- Ciclo di Esecuzione
  1. Prelievo Istruzione dalla Memoria
  2. Decodifica Istruzione
  3. Esecuzione Istruzione
- Ogni Processore e' caratterizzato da un proprio linguaggio macchina

Processore - Central Processing Unit (CPU)

- Processore e' composto da due sottosistemi:

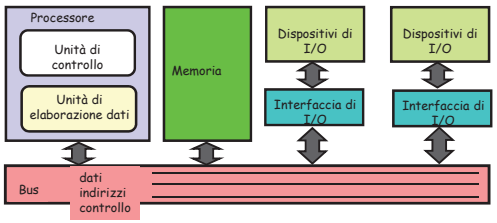
1. Unità di Controllo (Control) – Parte di Controllo
   - Controlla il sequenziamento e l'esecuzione delle istruzioni generando i segnali di controllo
2. Unita' di Elaborazione (Datapath) – Parte Operativa
   - Esegue le istruzioni
   - ALU
     - ✓ Esegue operazioni logico aritmetiche sui dati
   - Banco di Registri (Register File)
     - ✓ Memoria interna CPU
     - ✓ Program Counter (PC)
       - Indirizzo Prossima Istruzione
     - ✓ Instruction Register (IR)
       - Codice Istruzione da eseguire

Processore - Central Processor Unit (CPU)



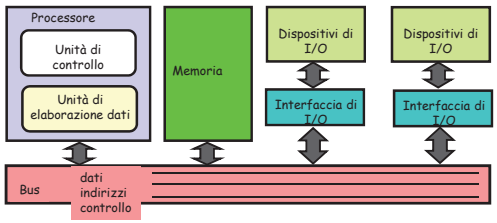
Il Bus

- La struttura di interconnessione più comune
  - percorsi di comunicazione tra due o più dispositivi
  - mezzo di trasmissione condiviso
  - Usualmente di tipo broadcast
    - ✓ I dati sono visibili da tutte le unità connesse



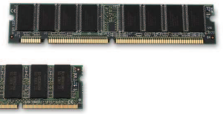
Il Bus

- Composto da più linee.
  - Ogni linea:
    - ✓ è conduttore elettrico (traccia in rame)
    - ✓ Può trasmettere segnali che rappresentano 0 o 1, un bit
- Tre tipi di linee
  - Linee dati: Bus dati
  - Linee indirizzo: Bus indirizzi
  - Linee di controllo: Bus controllo



La Memoria

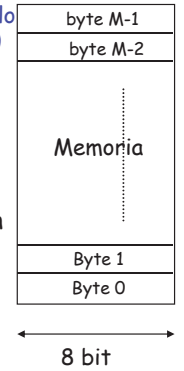
- Memoria Primaria: Memoria Centrale
  - Contiene istruzioni/dati dei programmi in esecuzione
    - ✓ …in formato binario
  - Volatile
  - RAM – Random Access Memory
    - ✓ Memoria ad accesso casuale
      - Tempo di accesso costante
    - ✓ SRAM, DRAM, etc.
  - Veloce (~10-100ns) Costosa
  - Dimensioni contenute (fino a qualche Gigabyte)
  - E' organizzata come una gerarchia di memorie
    - ✓ Cache di primo e secondo livello…



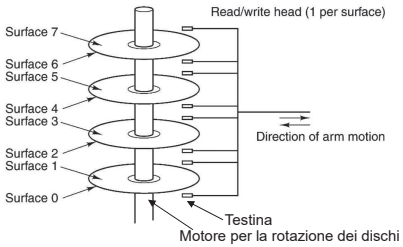
- Memoria Secondaria: Dischi, CD, etc..
  - Memoria di lungo periodo - non volatile
  - Tempo di accesso maggiori (~ms e piu'), economica

La Memoria Centrale

- Composta di celle, o locazioni, a loro volta composte da un numero fisso di bit
  - Cella elementare di memoria può memorizzare solo due valori: 0 o 1, cifra binaria (binary digit -> bit)
  - Tipicamente cella=byte (8 bit)
- Ogni locazione e' associata ad un indirizzo nell' intervallo [0,1,…,M-1]
  - M dimensione della memoria
  - La memoria e' vista come un vettore di byte
- La CPU (ma non solo) accede alle informazioni in scrittura/lettura tramite indirizzo della cella
  - Indirizzi a m bit: spazio di indirizzamento 2^m
    - ✓ Non necessariamente M= 2^m
- Operazioni: Lettura/Scrittura
  - Per ragioni di efficienza le operazioni di lettura/scrittura vengono effettuate per gruppi di byte detti parola (word)
  - 1 parola= 1,2,4,8 byte

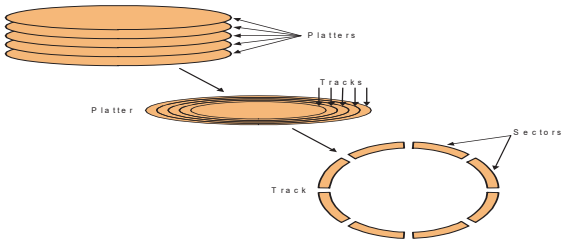


Disco rigido



- Costituito da un insieme di piatti rotanti (da 1 a 15)
- Piatti rivestiti di una superficie magnetica
- Esiste una testina (bobina) per ogni faccia
  - Generalmente piatti a doppia faccia
- Le testine di facce diverse sono collegate tra di loro e si muovono contemporaneamente
- Velocità di rotazione costante (ad es. 7200 RPM)

L'organizzazione dei dati sul disco



- Suddivisione della superficie del disco in anelli concentrici, detti *tracce*
- Registrazione seriale su tracce concentriche
  - 1000-5000 tracce
- Tracce adiacenti separate da spazi
- Ciascuna traccia è divisa in *settori*
  - Il settore è la più piccola unità che può essere trasferita (scritta o letta)
  - Centinaia di settori per traccia, generalmente di lunghezza fissa (ad es., 512 byte)
- La stessa traccia su piatti diversi forma un *cilindro*

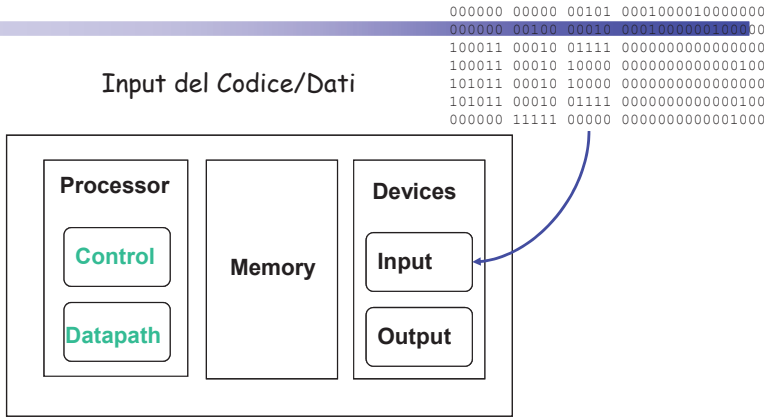
Memoria SSD – Solid State Drive

- Memoria di massa a semiconduttore che usa memorie allo stato solido per archiviazione di dati
  - Tipicamente memoria flash basata su MOSFET (transistor ad effetto di campo)
  - Non utilizza componenti meccanici (piatti, testine, etc.)

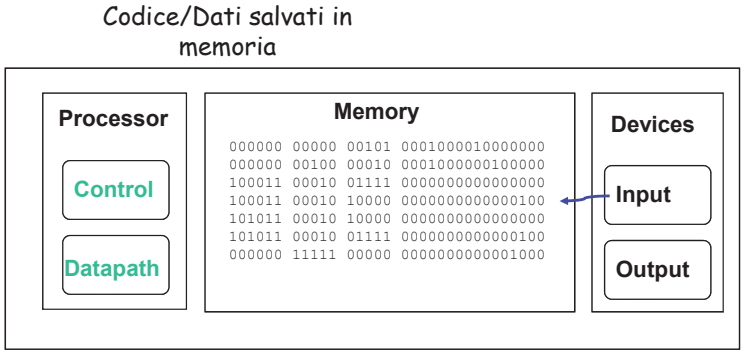
Calcolatore Elettronico

1. Accetta in ingresso informazioni codificate in forma digitale
2. Le elabora attraverso un programma memorizzato
3. Produce informazioni in uscita

Esecuzione di un Programma

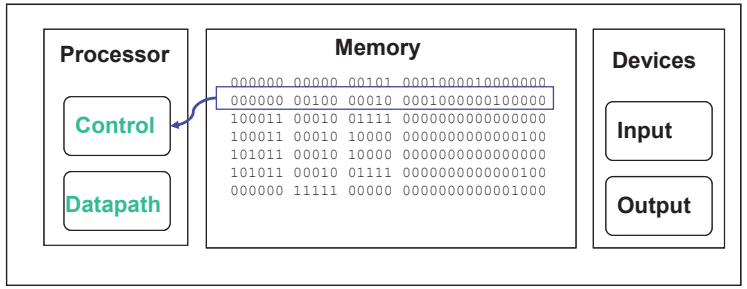


Esecuzione di un Programma



Esecuzione di un Programma

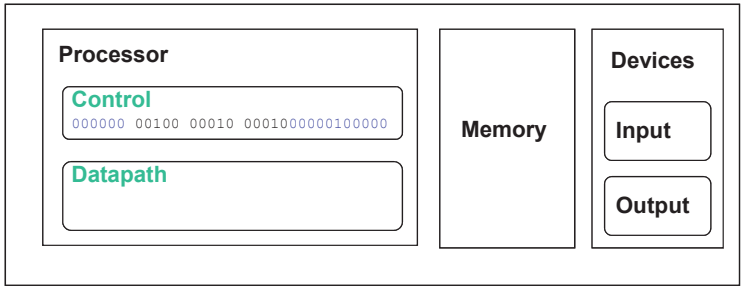
Il Processore preleva un'istruzione



Esecuzione di un Programma

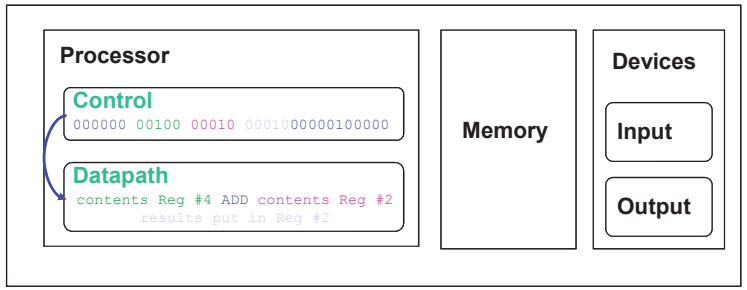
L'unita' di controllo del processore decodifica l'istruzione

Che devo fare?

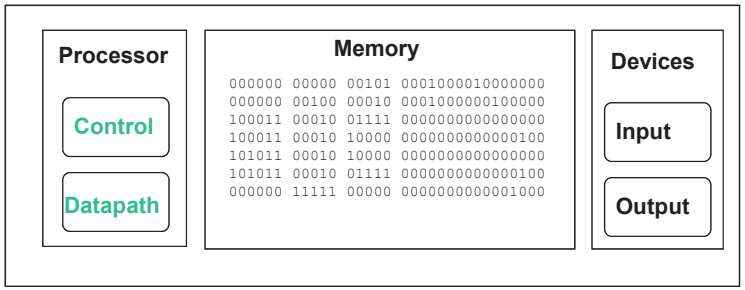


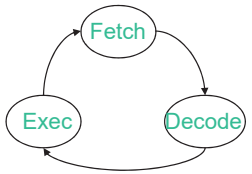
Esecuzione di un Programma

L'unita' di esecuzione del processore esegue l'istruzione (eventualmente prelevando operandi/salvando il risultato)



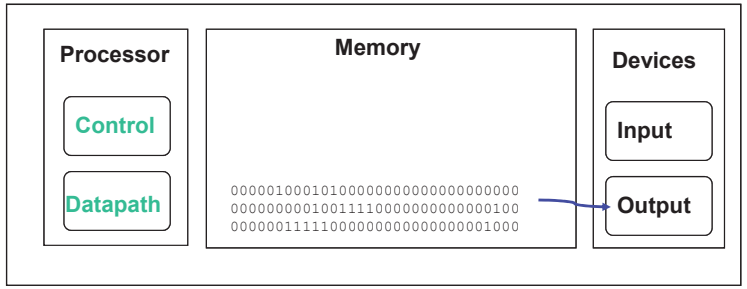
Esecuzione di un Programma





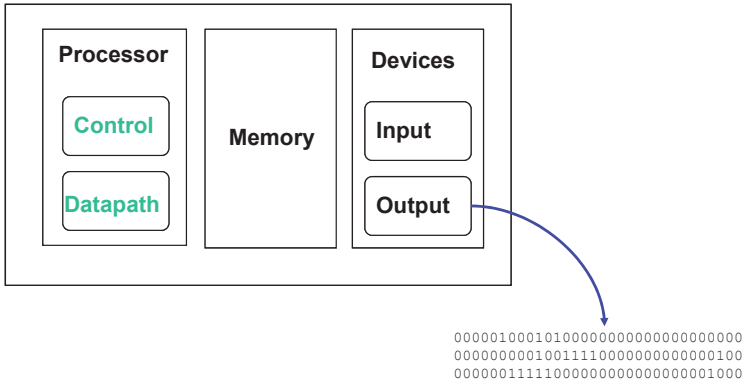
Esecuzione di un Programma

A completamento del programma, i risultati vengono salvati/inviati su un dispositivo di output



Esecuzione di un Programma

A completamento del programma, i risultati vengono salvati/inviati su un dispositivo di output



Algoritmi e Programmi

- Un programma e' un algoritmo espresso in un linguaggio di programmazione
- Il linguaggio direttamente eseguibile dal calcolatore e' il suo linguaggio macchina binario
- Un algoritmo espresso nel linguaggio macchina binario prende il nome di programma eseguibile

```
00000000101000010000000000011000
00000000100011100001100000100001
10001100011000100000000000000000
10001100111100100000000000000100
10101100111100100000000000000000
10101100011000100000000000000100
00000011111000000000000000001000
```

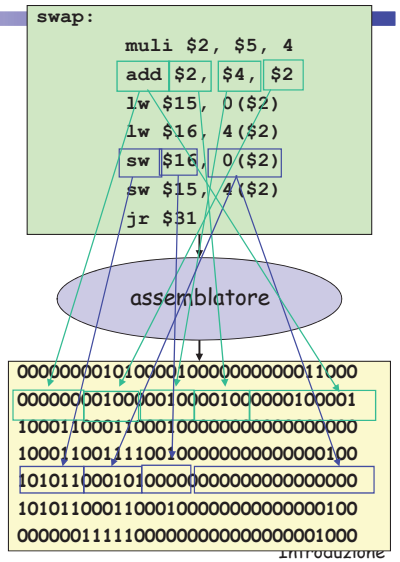

Dalle istruzioni in binario all'assembler

- Programmazione in linguaggio macchina binario
  - Per semplificare e velocizzare e' preferibile impiegare una rappesentazione simbolica delle istruzioni
- Linguaggio assembler: Linguaggio composto da istruzioni simboliche che corrispondono a istruzioni binarie
  - simbolico `add A,B` al posto di 1000110010100000
- Programma assemblatore (assembler):
  - Prende in ingresso un programma scritto in assembler
  - Genera il corrispondente programma in linguaggio macchina

```
00000000101000010000000000011000
00000000100011100001100000100001
10001100011000100000000000000000
10001100111100100000000000000100
10101100111100100000000000000000
10101100011000100000000000000100
00000011111000000000000000001000
```

Un programma in assembler

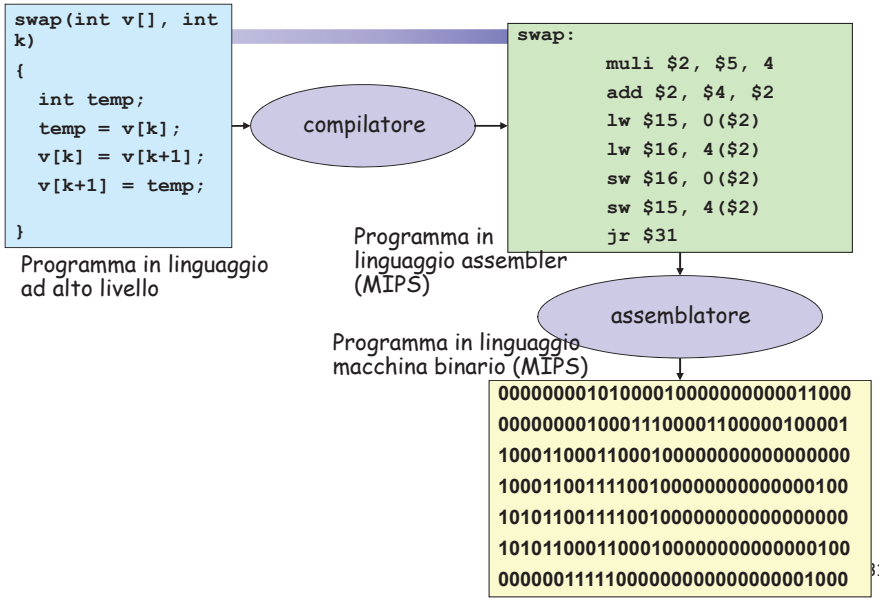
- Un esempio di traduzione
- Da un programma in assembler MIPS ad un programma in linguaggio macchina binario



Da un linguaggio di programmazione ad alto livello all'assembler

- Un linguaggio di programmazione ad alto livello
  - Offre astrazioni notevoli
    - ✓ Variabili, tipizzazione dei dati
    - ✓ Procedure, funzioni
    - ✓ Programmazione ad oggetti
    - ✓ Gestione di eccezioni
  - Aumenta la produttività del programmatore
  - Permette al programma di essere indipendente dal computer sul quale viene sviluppato
- Programma *compilatore*
  - traduce da un linguaggio di programmazione ad alto livello al linguaggio assembler
  - Talvolta traduce direttamente da linguaggio di programmazione ad alto livello a linguaggio macchina

Il processo di traduzione completo

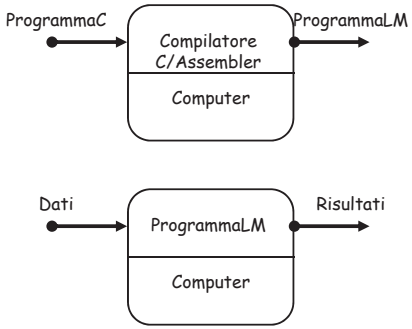


Compilazione ed Interpretazione

- La compilazione e' un processo di traduzione che, a partire da un programma scritto in linguaggio simbolico, ne genera una versione equivalente in assembler/linguaggio macchina
  - Il programma che effettua la traduzione prende il nome di compilatore
  - Il programma risultante puo' essere eseguito direttamente sul calcolatore
- L' interpretazione e' un processo di esecuzione indiretta di un programma ad opera di un programma detto interprete
  - L' interprete legge il codice del programma da eseguire e ne simula l' esecuzione, generando i relativi risultati
  - L' interprete e' , in sostanza, un simulatore di un calcolatore virtuale il cui linguaggio macchina e' il linguaggio interpretato

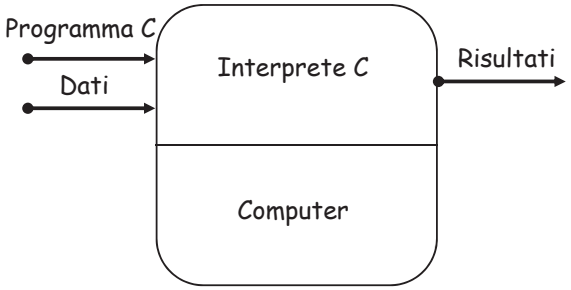
Il Compilatore

- Il Compilatore sostituisce a ogni istruzione del programma PL1, scritto nel linguaggio L1, una sequenza di istruzioni scritte nel linguaggio L0 di piu' basso livello
  - Il programma risultante PL0 equivalente a PL1 ma scritto nel linguaggio L0
    - ✓ Se L0 e' il linguaggio macchina binario LM, PL0 puo' essere direttamente eseguito
    - ✓ Altrimenti puo' essere ulteriormente compilato, oppure interpretato



L' interprete

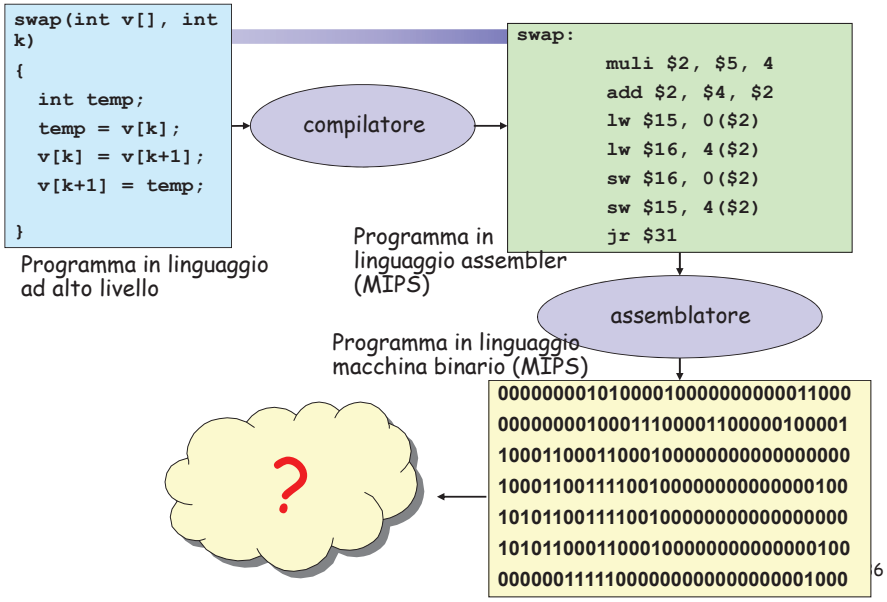
- Per eseguire PL1, scritto in L1, l' interprete, scritto in L0, analizza – passo passo - ogni istruzione da eseguire di PL1 e ne realizza l' effetto, eseguendo una sequenza di istruzioni del linguaggio di L0 ad essa equivalenti



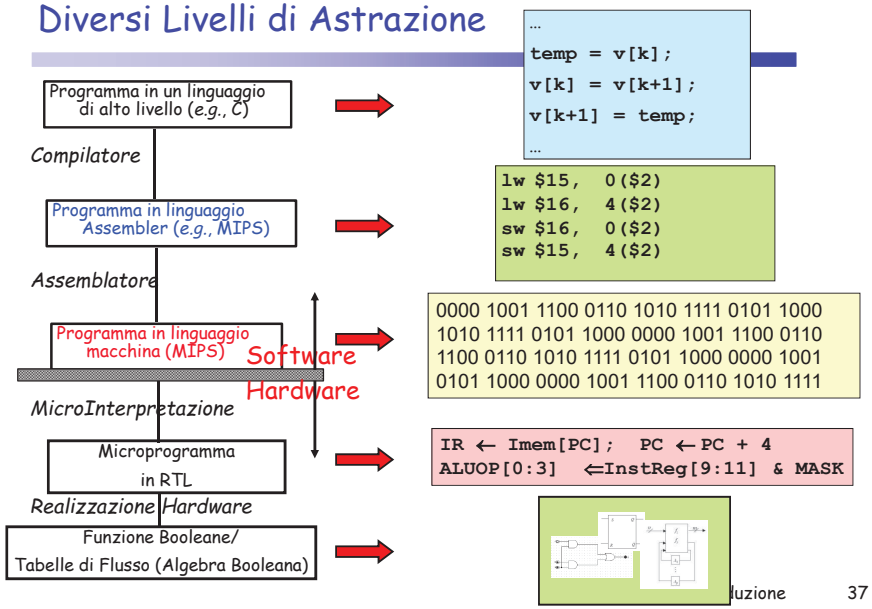
Interpretazione o compilazione?

- Ogni linguaggio puo' essere sia interpretato che compilato
  - Il C, C++ sono normalmente compilati
- L' interpretazione e' piu' lenta dell' esecuzione diretta
  - Migliore diagnostica
  - Maggiore portabilita'
    - ✓ L' interprete, se scritto in un linguaggio diffuso, puo' essere reso piu' facilmente disponibile su un nuovo calcolatore
- Talvolta, compilazione ed interpretazione sono combinate
  - Per combinare i vantaggi
  - Si compila in un linguaggio intermedio che poi e' interpretato
    - ✓ Approccio utilizzato per l' esecuzione dei programmi in Java

Diversi Livelli di Astrazione

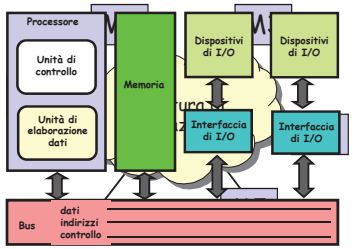


Diversi Livelli di Astrazione



Architettura a Livelli

- Calcolatore come insieme di livelli o Macchine Virtuali {MV0, MV1,…,MVn}
  - MVi rappresenta un diverso livello di astrazione del calcolatore
- Macchina Virtuale a livello i
  - Moduli che operano al livello i
    - ✓ e loro struttura d' interazione
  - Linguaggio Li per la programmazione a livello i
    - ✓ Elemento Caratterizzante
  - Risorse Ri visibili al livello i
  - (Li,Ri) l' architettura del livello i
    - ✓ Aspetti visibili a chi programma al livello i

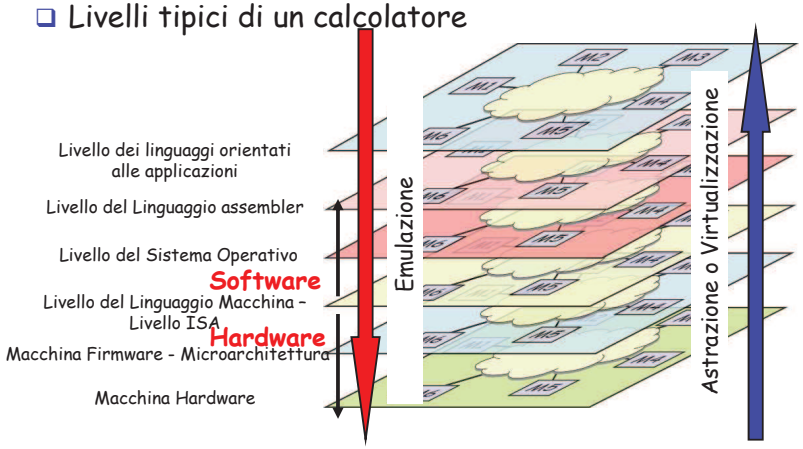


Architettura a Livelli

- Calcolatore come insieme di livelli o Macchine Virtuali {MV0, MV1,…,MVn}
  - MVi rappresenta un diverso livello di astrazione del calcolatore
- I livelli MV0,…,MVn sono ordinati seconda una relazione gerarchica
- Livello MV0 (Macchina Hardware) e' il livello dei componenti fisici a partire dal quale viene costruita una astrazione o virtualizzazione sempre maggiore

Architettura a Livelli

- Livelli tipici di un calcolatore



Architettura a Livelli

- Relazione Gerarchica fra livelli
- Ogni istruzione I_k di Li e' implementata da un "programma" P_{ikj} scritto in linguaggio Lj, j<i
- Corrispondenza tra istruzione I_k e programma P_{ikj} puo' essere:

1. Statica Compilazione
2. Dinamica Interpretazione

Architettura a Livelli

- Linguaggi ai vari livelli e loro relazioni

5. Livello dei Linguaggi orientati alle appl. (C,C++, Java, Python)
   ↓ Compilazione o Interpretazione
4. Livello del Linguaggio Assembler (Assembler)
   ↓ Assemblaggio (Assembler)
3. Livello del Sistema Operativo (Linguaggio Macchina+Chiamate di Sistema)
   ↓ Interpretazione parziale
2. Livello del linguaggio macchina – ISA (Linguaggio Macchina)
   ↓ Esecuzione diretta o microinterpretazione
1. Macchina Firmware (Linguaggio di microprogrammazione)
   ↓ Realizzazione hardware
0. Macchina Hardware (Algebra booleana)

Software

Hardware

Architettura a Livelli

- Livello 0: Macchina Hardware
  - Livello della logica digitale
  - Moduli: Reti combinatorie e sequenziali
  - L0: Algebra booleana binaria
  - R0: Porte logiche
- Livello 1: Macchina Firmware – Microarchitettura
  - Interpreta ed esegue le istruzioni del linguaggio macchina
  - E' direttamente realizzato con i componenti della macchina hardware
  - Moduli - Unita' di Elaborazione: CPU, Memoria, Unita' di I/O
  - L1: Linguaggio di Microprogrammazione
  - R1: Reti combinatorie e sequenziali

Architettura a Livelli

- Livello 2: Livello del Linguaggio Macchina (ISA)
  - Macchina nuda come appare al programmatore di sistema. Le istruzioni del suo linguaggio sono interpretate ed eseguite dai microprogrammi del processore
  - Moduli: Programmi
  - L2: Linguaggio macchina
  - R2: Registri, spazio di memoria
- Livello 3: Livello del Sistema Operativo
  - Gestisce le risorse del sistema nei confronti dei livelli superiori
    - ✓ Processore (Multiprogrammazione), Memoria Principale (Memoria Virtuale), Memoria Secondaria (File System)
  - L3: Linguaggio macchina+Chiamate di sistema
  - R3: R2+spazi di memoria, dispositivi di I/O, processori,...
  - Interpretazione parziale (per le chiamate di sistema)

Architettura a Livelli

- Livello 4: Livello del Linguaggio Assembler
  - Livello piu' basso utilizzabile dal programmatore di applicazioni
  - Moduli: Processi
  - L4: linguaggio assembler
    - ✓ Forma simbolica dei linguaggi L2, L3
  - E' implementato tramite "assemblaggio"
- Livello 5: Livello dei Linguaggi orientati alle appl.
  - Livello usato per sviluppare applicazioni
  - Moduli: Processi
  - L5: C, C++, Java, etc..
  - E' implementato tramite compilazione e/o interpretazione

Architettura a Livelli

- Linguaggi ai vari livelli e loro relazioni

5. Livello dei Linguaggi orientati alle appl. (C,C++, Java)
   - ↓ Compilazione o Interpretazione (Java)
4. Livello del Linguaggio Assembler (Assembler)
   - ↓ Assemblaggio (Assembler)
3. Livello del Sistema Operativo (Linguaggio Macchina+Chiamate di Sistema)
   - ↓ Interpretazione parziale

Software

2. Livello del linguaggio macchina – ISA (Linguaggio Macchina)
   - ↓ Esecuzione diretta o microinterpretazione

Hardware

1. Macchina Firmware (Linguaggio di microprogrammazione)
   - ↓ Realizzazione hardware
0. Macchina Hardware (Algebra booleana?)

Architettura dei Calcolatori

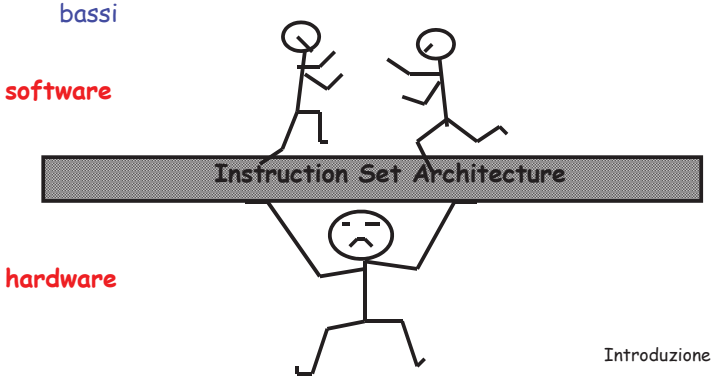
Instruction Set Architecture (ISA) + Machine Organization (MO)

- ISA - Cosa il calcolatore e' capace di fare
  - *Quali istruzioni?*
- MO - Come la macchina implementa ISA
  - *Loro Implementazione Fisica*

Instruction Set Architecture (ISA)

- ISA-Interfaccia tra Hardware e Software

  Deve:
  - Fornire funzionalita' ai livelli piu' alti
  - Permettere un efficiente implementazione ai livelli piu' bassi



Instruction Set Architecture (ISA)

"... the attributes of a [computing] system as seen by the programmer, *i.e.* the conceptual structure and functional behavior, as distinct from the organization of the data flows and controls, the logic design, and the physical implementation."

— Amdahl, Blaaw, and Brooks, 1964

- Insieme delle istruzioni
- Numero di bit che rappresentano tipi di dati
- Tecniche di indirizzamento della memoria, ...

Organizzazione del Calcolatore

- Caratteristiche delle principali unita' funzionali
  - Registri, ALU, etc.
- Strutture di Interconnessione
- Unita' di Elaborazione e Unita' di Controllo
- Organizzazione del sottosistema di Memoria
- Descrizione tramite *Register Transfer Level* (RTL)

Evoluzione delle ISA

- Dai primi calcolatori (anni '50)
  - Semplici
  - Pochi tipi di istruzioni e modi di indirizzamento
- ...ai CISC (Complex Instruction Set Computer) (anni'70)
  - ISA e processori via via piu' complessi
  - Riduzione gap semantico tra linguaggi di alto livello e linguaggi macchina
  - Difficili da ottimizzare
- ...ai RISC (Reduced Instruction Set Computer)
  - ISA e processori semplici
  - Facilita l'ottimizzazione
    - ✓ Software: Compilatori Ottimizzanti
    - ✓ Hardware: Processori veloci (impiego efficiente delle tecniche di pipelining)

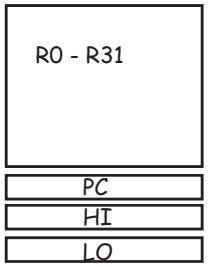
Le Caratteristiche dei RISC

- Le architetture RISC sono caratterizzate da poche proprieta' chiave
  - Numero elevato di registri generali
  - Pochi formati di istruzione
    - ✓ Generalmente della stessa lunghezza
  - Numero limitato di codici operativi e di modi di indirizzamento
  - Tutte le operazioni logico/aritmetiche hanno come operandi i registri
  - Le sole operazioni che coinvolgono la memoria sono quelle di load e store

MIPS R3000 ISA

- Categorie di Istruzioni:
  - Load/Store
  - Computational
    - ✓ Registri come operandi
  - Jump and Branch
  - Floating Point
    - ✓ coprocessor
  - Gestione della Memoria
  - Speciali

Registri



3 formati istruzione: tutti a 32 bits