

# Calcolatori Elettronici B

a.a. 2007/2008

## **INTRODUZIONE AL CORSO**

*Massimiliano Giacomini*

# Contattarmi

**Ufficio** n. 27 del DEA

**E-mail:** giacomini@ing.unibs.it

Orario di ricevimento:

**GIOVEDI' DALLE 10.00 alle 13.00**

**[in caso di problemi: altre giornate su appuntamento]**

- Garantito in periodo di lezione [eventi speciali - p.es. sedute di laurea - a parte] e di norma anche dopo, ma si consiglia appuntamento via mail per sicurezza
- Per particolari esigenze (es. NO per chiedere quando escono i risultati!):  
si consiglia di usare l'e-mail
- Eventuali **variazioni di orario/giorno** indicate nel **sito internet** del corso.

# PROPEDEUTICITA' E PRECEDENZE OBBLIGATORIE

Corso di laurea primo livello Ing. Informazione

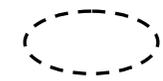
PROPEDEUTICITA' (ovvero: esami "ufficialmente consigliati")

⊖ **Fondamenti di Informatica A** ⊕

- **Fondamenti di Informatica B**

⊖ **Calcolatori Elettronici A** ⊕

- Fondamenti di Informatica C



Fortemente  
consigliati  
da me

INFO A e INFO B sono precedenze obbligatorie (devono essere stati registrati per poter registrare calcolatori B: controllate voi!!!)

Corso di laurea specialistica Ing. Informatica e Ing. Elettronica per l'Automaz.

PRECEDENZA OBBLIGATORIA:

- Calcolatori Elettronici A

# Modalità di Esame

## PROCEDURA “TRADIZIONALE”:



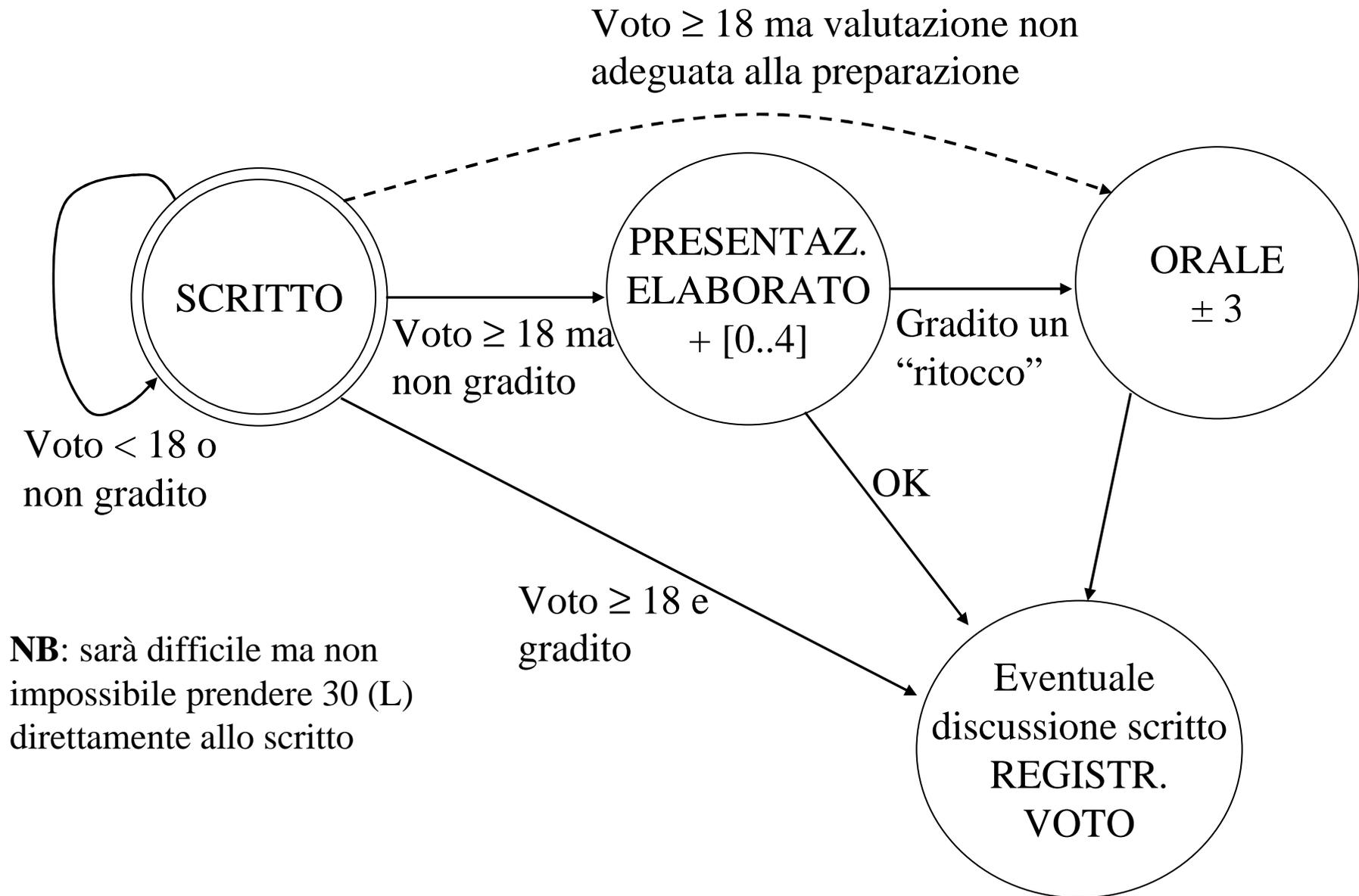
### Principi di progettazione suggeriti dal Patterson:

- rendere veloce l'evento più frequente
- un buon progetto è frutto di buoni compromessi



USEREMO LA MODALITA' SEGUENTE...

[la comprensione delle istruzioni  
è propedeutica allo svolgimento dell'esame]



**Evento più frequente:** solo scritto, pretese di voto non eccessive

**Compromesso:** elaborato per appassionati (NO COPIA E INCOLLA!!!)

## TIPOLOGIA DI ESAME SCRITTO

- Lungo e quanto più possibile ampio nei contenuti
- Teoria + esercizi
- Diversi quesiti (domande a risposta aperta, a crocette, esercizi, ecc.); ciascuno ha una valutazione compresa tra 0 e max punteggio indicato, somma compresa tra 30 e 32 (a seconda dei casi)
- Tempo di svolgimento sovrabbondante (di solito 3 ore + ev. recupero)

## TIPOLOGIA DI ORALE

- Breve e “di aggiustamento”
- Discussione dello scritto

## TIPOLOGIE DI ELABORATI

- Argomenti proposti dal docente (non affrontati o solo accennati nel corso)
- Sviluppo di software/semplificati dimostratori per la didattica
- Casi concreti di applicazione di argomenti teorici visti a lezione su processori e architetture attuali
- Sviluppo in linguaggio VHDL o Verilog?
- Argomenti particolarmente interessanti proposti dallo studente

NB: max 2-3 persone

richiesta interazione/discussione determinante ai fini della valutazione

# Materiale di studio

**Sito Internet del Corso:** <http://zeus.ing.unibs.it/calcb/>

- Lucidi del corso
- Link, informazioni varie, eventuale software da scaricare
- Regole, news, risultati degli esami, ecc.
- Ogni altra cosa dovesse risultare utile!

## **Libro**

- Patterson & Hennessy:

*Computer Organization and Design [Third Edition]*

[Morgan Kaufmann - Elsevier]

anche in italiano: “*Struttura e progetto dei calcolatori*”

– seconda edizione Zanichelli condotta sulla terza edizione americana

*Lo stesso di Calcolatori A*

## **Libri per la consultazione:**

- Hamacher, Vranesic, Zaky:

*Introduzione all'architettura dei calcolatori – seconda edizione*

[Mc Graw-Hill, 2007]

- Giacomo Bucci:

*Architettura e organizzazione dei calcolatori elettronici*

*[Vol.1: Fondamenti – Vol. 2: Strutture Avanzate]*

[Mc Graw-Hill, 2005 e 2006]

- A. Tanenbaum:

*Structured Computer Organization – quinta edizione*

[Prentice Hall, 2005]

NB: in ogni caso, il programma è costituito da tutti e soli gli argomenti trattati a lezione!!!

# Programma (di massima) del corso

NB: il corso approfondisce ed estende i concetti trattati nel corso di Calcolatori A (conviene averlo già superato!)

1. Brevi richiami su reti logiche (richiami dal corso di Calcolatori A)
2. Tecniche di specifica e realizzazione del controllo (singolo ciclo, multi-ciclo a stato esplicito e microprogrammato)
3. La struttura a pipeline e gestione delle criticità, prestazioni vs. realizzazione multi-ciclo.
4. Pipeline superscalari e dinamiche.
5. Memoria virtuale e gerarchia di memorie.
6. Gestione dell'Input/Output e bus.
7. Cenni al linguaggio Assembler Intel

## Nuovi Argomenti trattati nel corso di Calcolatori B

### Calcolatori A

Assembler MIPS

CPU: Multi-Ciclo

Memoria Cache

I/O

### Calcolatori B

Assembler Intel (cenni)

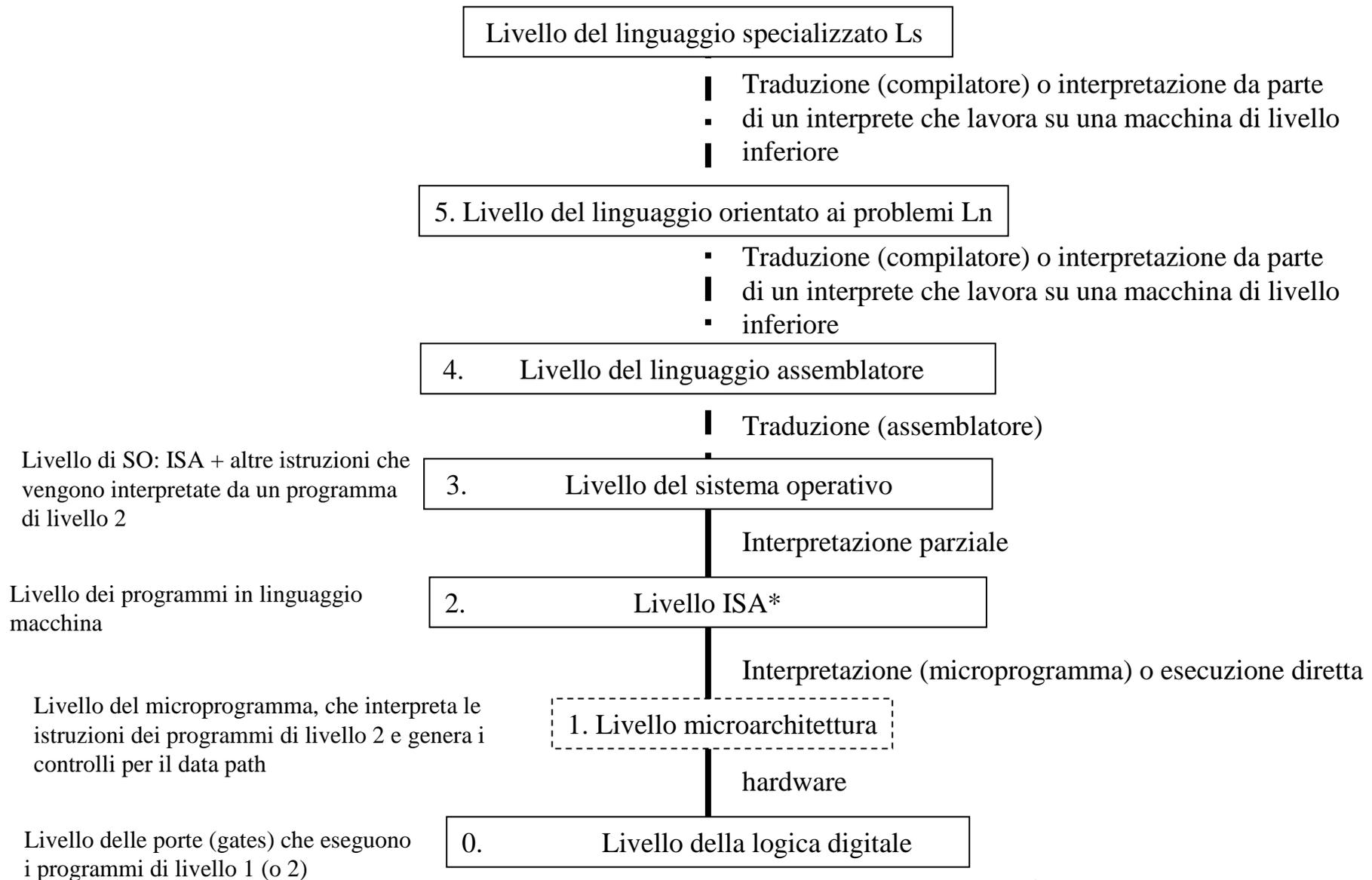
Pipeline

Memoria virtuale

Bus sincroni e asincroni

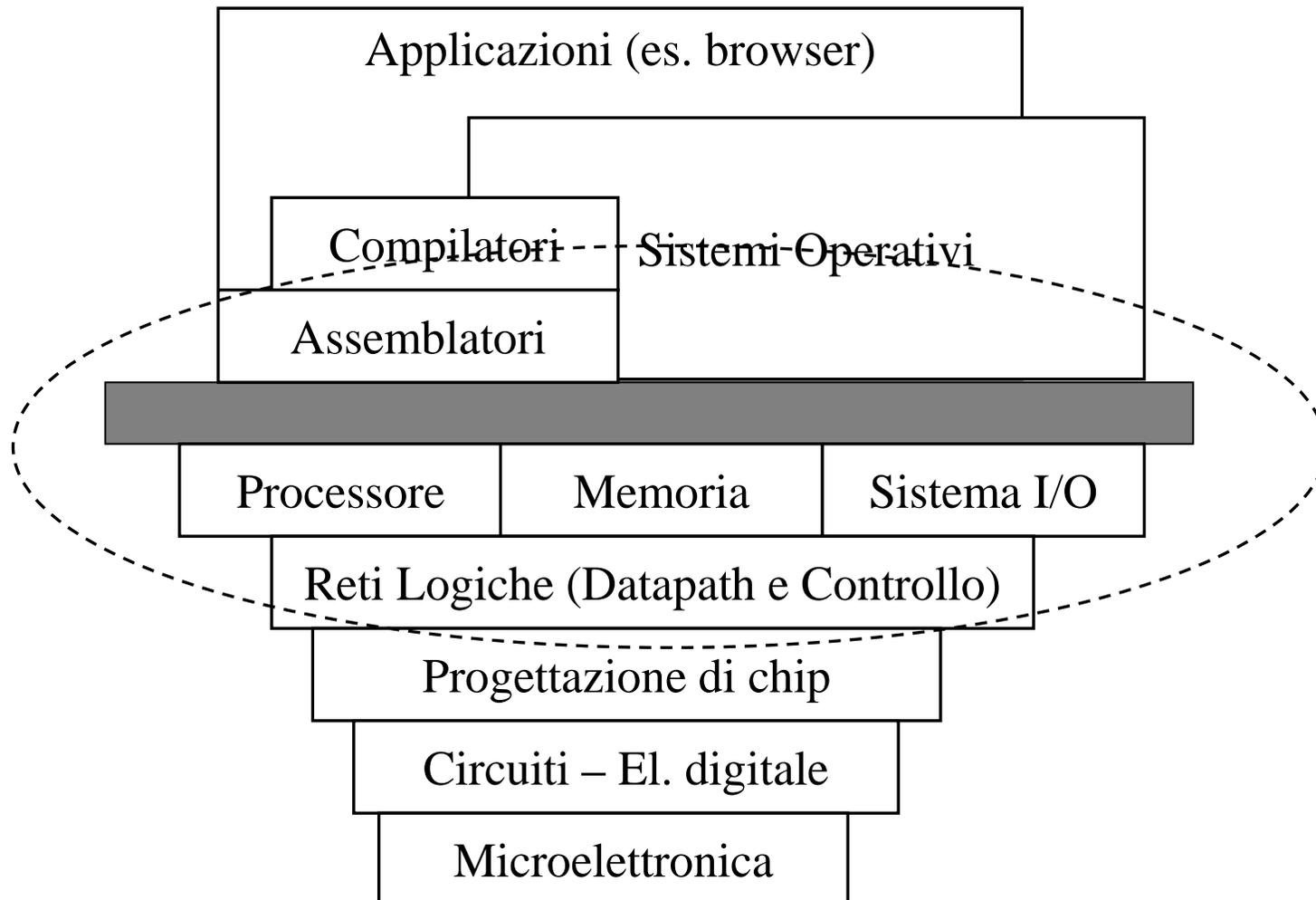
# Architetture

descrivono il calcolatore a diversi livelli di astrazione



- Il calcolatore è progettato come una sequenza di macchine a diversi livelli, ognuna costruita sulle macchine definite ai livelli sottostanti.
- Ogni livello rappresenta una distinta astrazione, caratterizzata da differenti oggetti e operazioni.
- I tipi di dati, le operazioni e le caratteristiche di ogni livello sono chiamate **architettura**.
  - Dal livello 5 in su, i linguaggi in cui esprimere i programmi sono chiamati linguaggi di alto livello: forniscono dati e operazioni per descrivere soluzioni di problemi in termini comprensibili per persone esperte in un certo campo.
  - I programmi del livello 4 sono in forma simbolica; vengono generalmente tradotti (o a volte interpretati) da altri programmi.
  - I programmi ai livelli 1, 2 e 3 sono sempre interpretati; la forma interpretata dalla macchina è sempre numerica.

# FOCUS DEI CORSI DI CALCOLATORI (A+B)



I confini tra hardware e software sono sfumati:

Hardware and software are logically equivalent (Andrew Tanenbaum)

Hardware is just petrified software (Karen Panetta Lenz)

ma “dovendo scegliere”:

E' vero che il software non potrebbe esercitare i poteri della sua leggerezza se non mediante la pesantezza dell'hardware; ma è il software che comanda, che agisce sul mondo esterno e sulle macchine, le quali esistono solo in funzione del software, si evolvono in modo d'elaborare programmi sempre più complessi. (Italo Calvino)

NB: Anche se si lavora ad un livello, è importante conoscere gli altri! Si richiede il coordinamento di molti livelli di astrazione, mai completamente indipendenti!

- Necessità di conoscere i livelli più bassi (esempi):
  - Sistema Operativo richiede di conoscere organizzazione del calcolatore
  - Prestazioni influenzate da ISA, organizzazione calcolatore e livelli dell'elettronica digitale/microelettronica
  - Compilatore può produrre software “ottimizzato” per l'hw sottostante
- Necessità di conoscere i livelli più alti (esempi):
  - le prestazioni sono definite con riferimento al sw (carico di lavoro-benchmark)
  - le ottimizzazioni dell'hw si fanno sulla base dei carichi di lavoro previsti