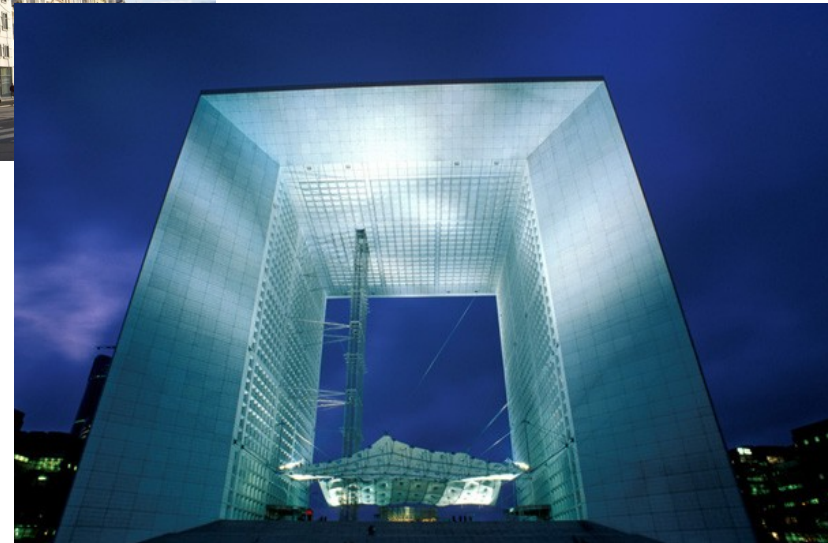
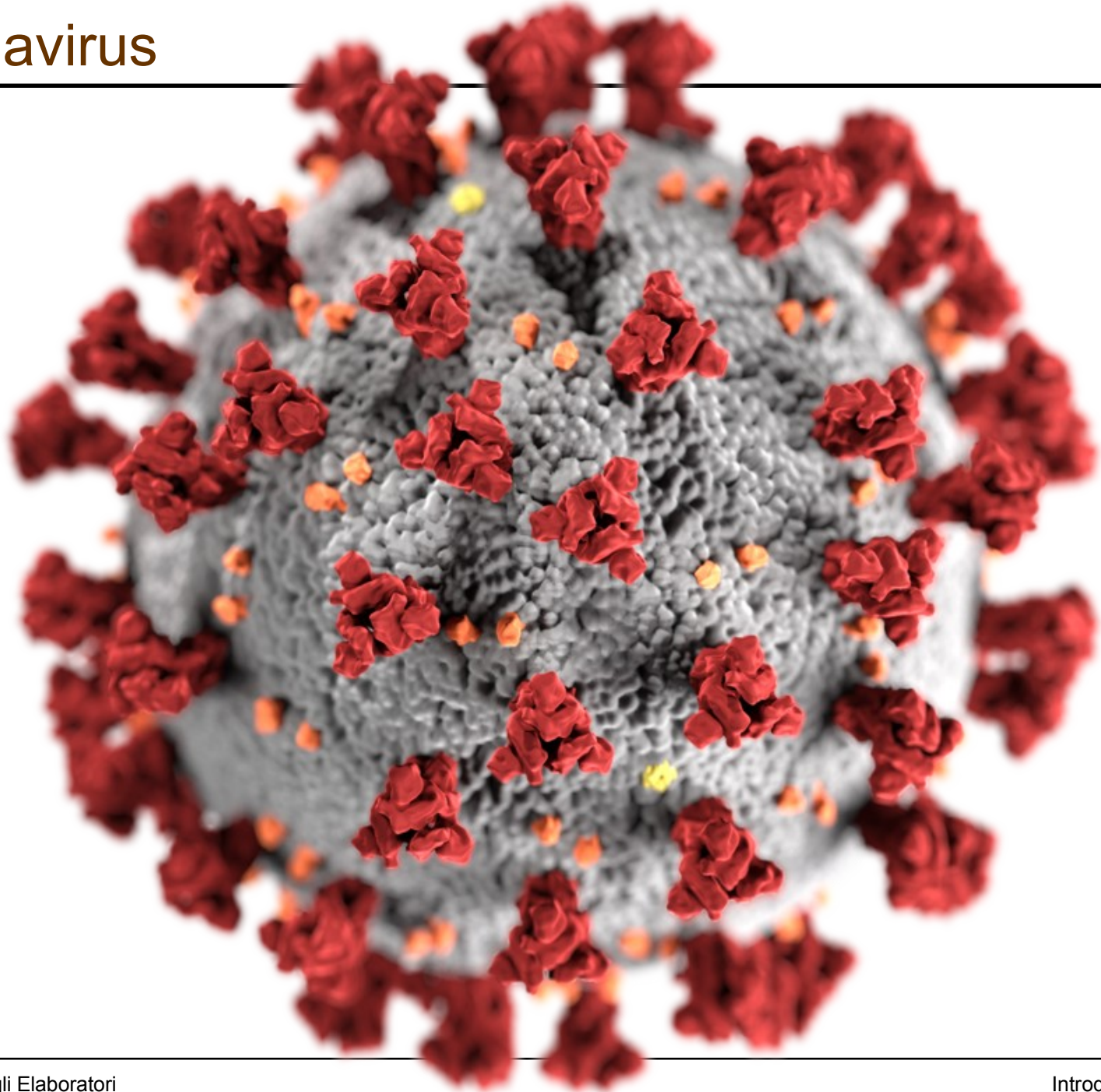


Architettura degli Elaboratori



Prof. Ivan Lanese

Coronavirus



Coronavirus

Come sapete siamo in una situazione molto particolare a causa del coronavirus: prima di qualunque considerazione sulla didattica dobbiamo garantire la sicurezza e la salute di tutti

Per seguire in presenza dovete prenotare tramite presente e avere il green pass: verranno fatti controlli a campione

Seguite sempre le linee guida che vi sono state fornite:

- Usate le mascherine, che coprano naso e bocca
- Mantenete le distanze interpersonali
- Igienizzate spesso le mani
- Rimanete a casa se non in buona salute (seguite le lezioni da remoto)

<https://www.unibo.it/it/ateneo/covid-19-misure-adottate-da-alma-mater>

Posso garantire per esperienza diretta che questo virus è pericoloso e può uccidere

Coronavirus

Il coronavirus avrà un impatto non trascurabile sulla didattica

Per chi segue le lezioni da casa i punti critici sono la concentrazione e l'interazione

- L'interazione via chat sta andando molto bene
- Incoraggio anche chi e' in aula a tener d'occhio la chat da computer o smartphone

Concentrazione

A casa avete più fonti di distrazione e io non ho modo di capire se state seguendo (e capendo) o meno

Una lezione universitaria va seguita con attenzione, se uno si perde ha poche occasioni di recuperare

Cercate di eliminare le fonti di distrazione e di rimanere concentrati

Se vi perdete cercate di recuperare entro la lezione successiva, usando le slide, il libro o facendovi domande (vedi interazione)

L'interazione è fondamentale e si svolge in due modi principali:

- Domande durante la lezione
 - Domande attinenti all'argomento di cui si sta parlando
 - Se non capite chiedete chiarimenti, specifici per quanto possibile
 - Per domande su argomenti vecchi sfruttate i momenti di pausa (prima della lezione, dopo la fine, o durante la pausa)
- Correzione degli esercizi e domande durante la correzione
 - Gli esercizi verranno corretti da volontari
 - Se non capite la soluzione o avete soluzioni diverse chiedete chiarimenti (alcuni esercizi possono avere più soluzioni corrette anche molto diverse fra loro)

Architettura degli Elaboratori

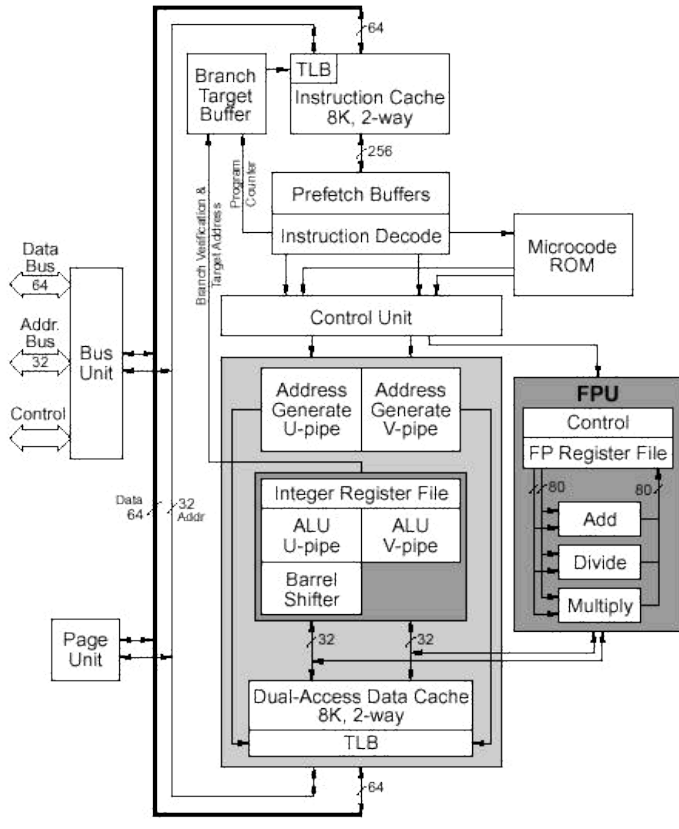
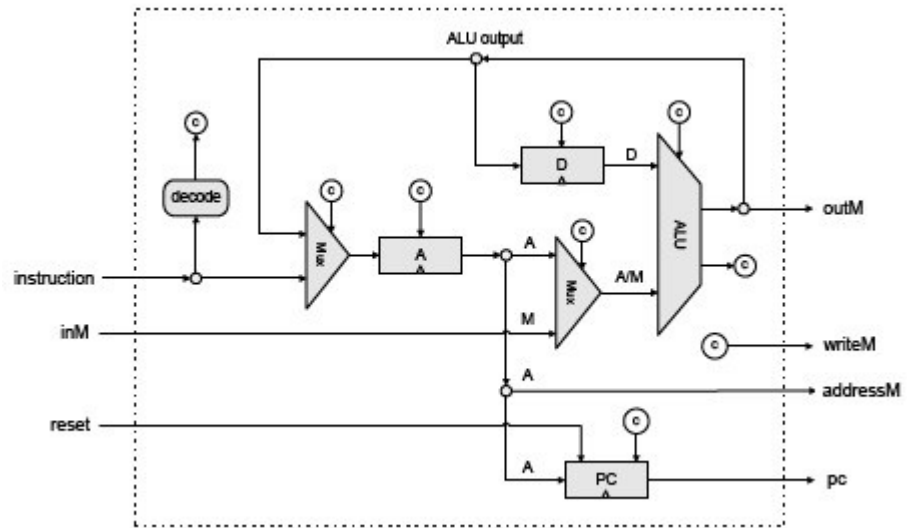


Figure 1. Pentium block diagram.



Prof. Ivan Lanese

Perchè studiare l'Architettura degli Elaboratori

Applicazioni software: Conoscete già una grande quantità di applicazioni software

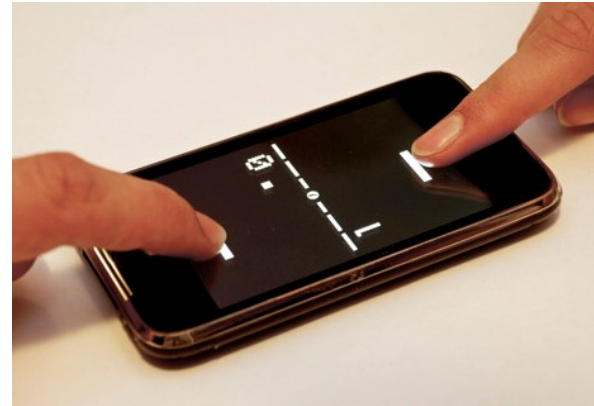
Svilupperete nuove applicazioni software: Corsi come "Programmazione", "Linguaggi", "Ingegneria del Software", ... vi forniranno conoscenze e strumenti per sviluppare nuove applicazioni software

Ma cosa permette a tali applicazioni di funzionare?
Il corso di Architettura degli Elaboratori cerca di dare risposta a tale domanda

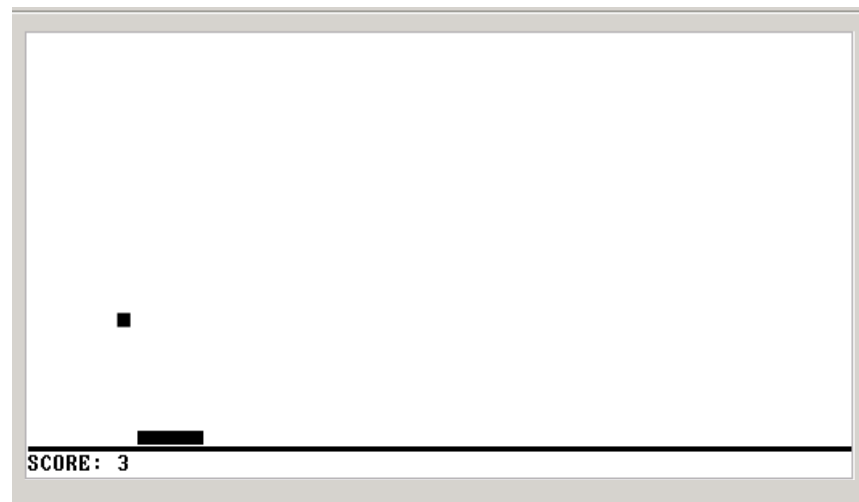
Un caso di studio



Pong, 1985



Pong, 2011



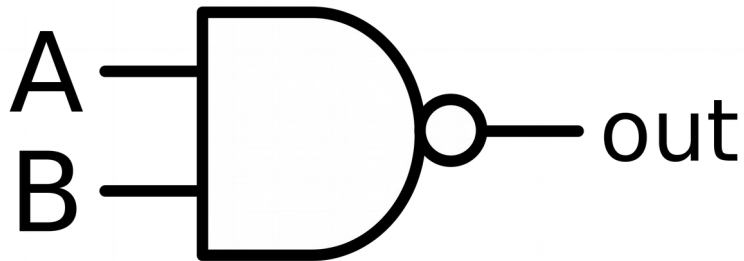
Il nostro "Pong"

Un caso di studio

Non solo studieremo ..

.. ma **REALIZZEREMO** in tutti i suoi minimi dettagli un elaboratore che permette di eseguire il nostro "Pong"

Cosa useremo? Come unico "mattoncino" useremo la porta logica NAND

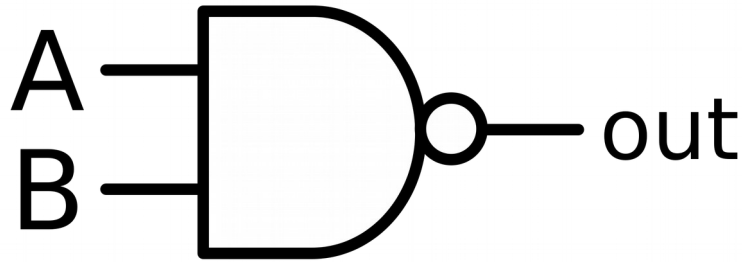


A	B	out
0	0	1
0	1	1
1	0	1
1	1	0

(useremo anche fili per collegare gli ingressi e le uscite dei "mattoncini", un display 256x512 pixel b/w e una tastiera)



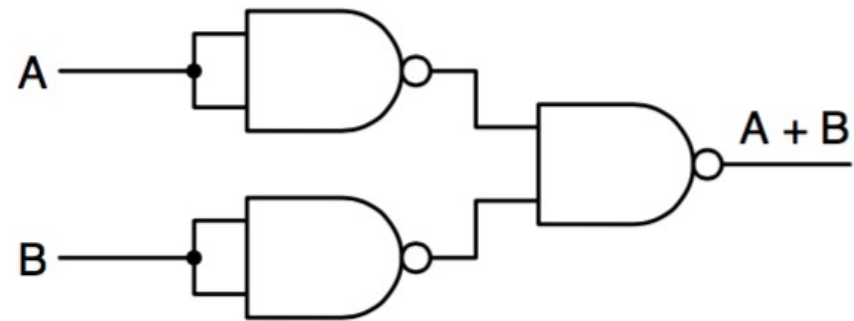
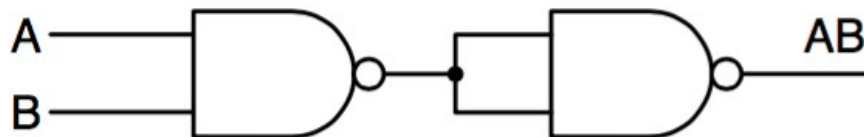
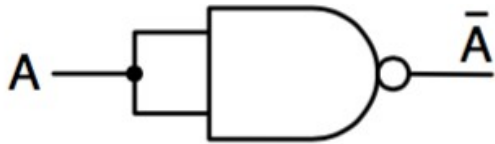
Perchè usare la porta logica NAND?



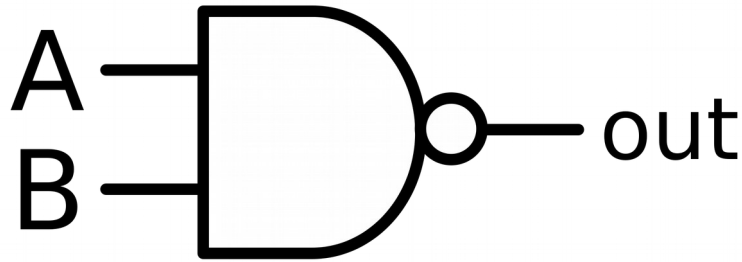
A	B	out
0	0	1
0	1	1
1	0	1
1	1	0

E' una porta logica **UNIVERSALE**:

combinando porte NAND si possono realizzare tutte le funzioni logiche "booleane" (= funzioni a due valori 0,1)



Perchè usare la porta logica NAND?



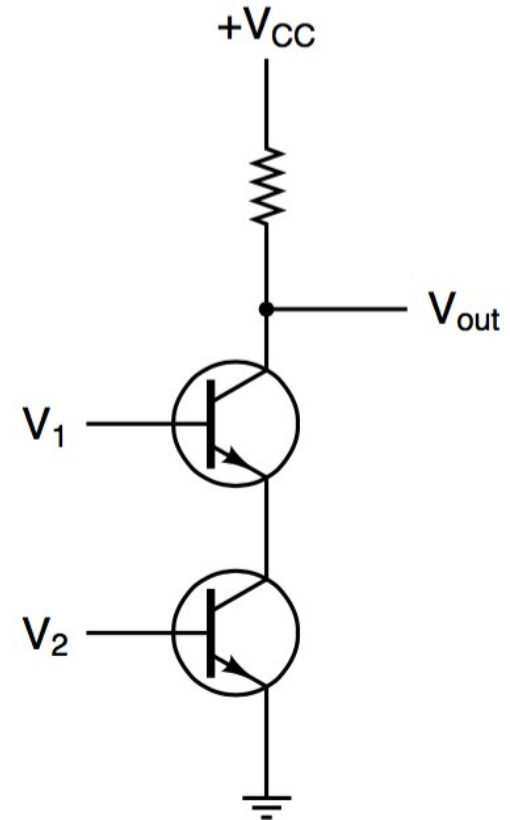
A	B	out
0	0	1
0	1	1
1	0	1
1	1	0

Inoltre è facile da realizzare usando "transistor"

(0 = no tensione ,
1 = tensione)

(transistor:

- senza tensione in ingresso è una resistenza infinita
- con tensione in ingresso è un conduttore ideale)



Ma quanto è difficile?

Complessità: Realizzare il nostro "Pong" usando solo porte NAND è estremamente complesso!

Come affrontare tale complessità?

- useremo il principio di "astrazione / implementazione"
- tale principio è uno dei concetti base dell'Informatica ..
- .. ma di che si tratta?

Astrazione / implementazione: si crea un "mattoncino" (implementazione) e poi lo si usa dimenticandosi (astrazione) di come è stato costruito

Uso del principio di astrazione/implementazione

Astrazione/implementazione fondamentale in ogni campo

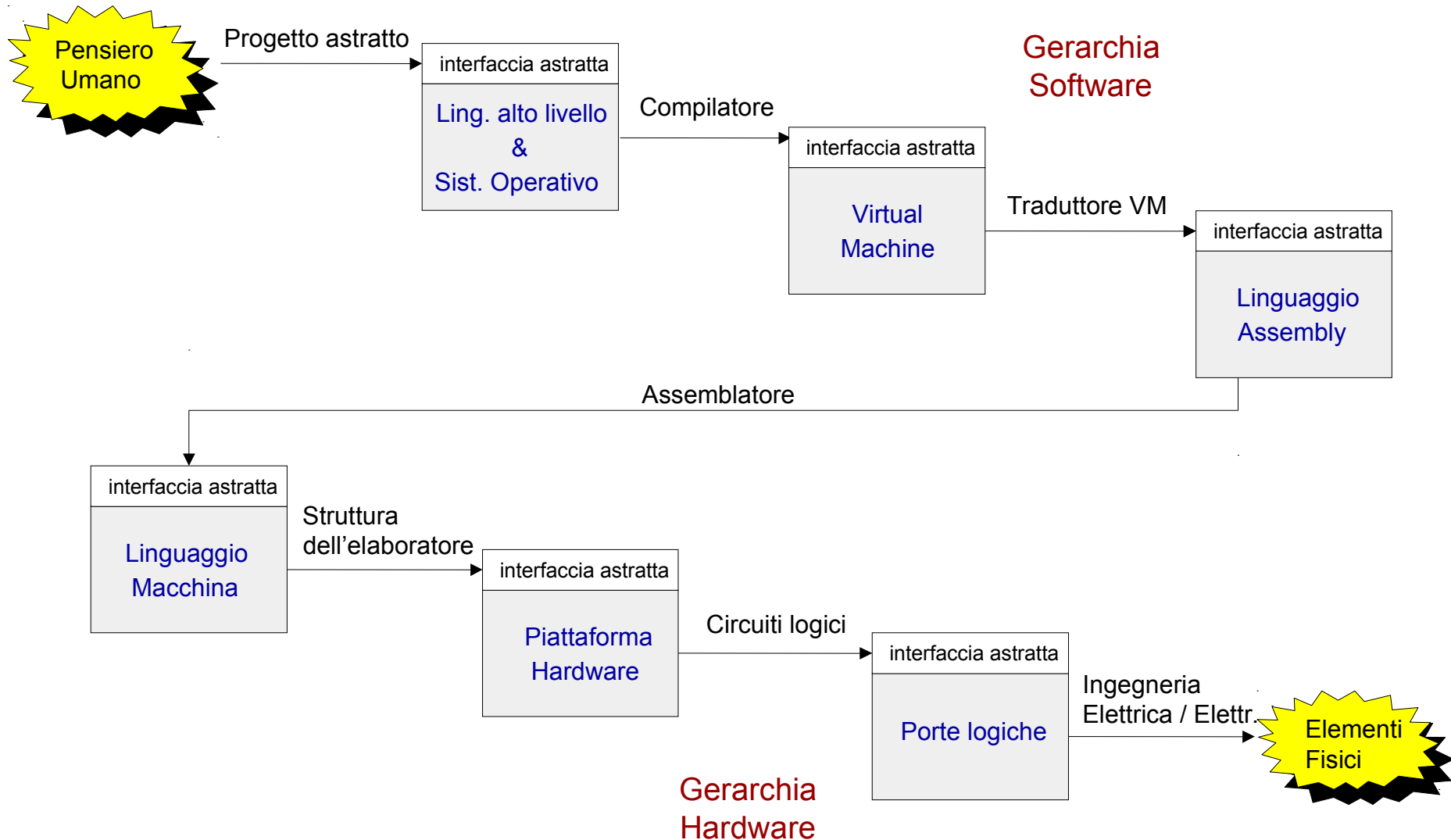
- posso usare la tv senza sapere come è fatta internamente
- posso scrivere un programma senza sapere come sono eseguite le istruzioni
- posso chiamare una funzione senza sapere il suo codice

Mi basta conoscere le loro **interfacce**

Come realizzeremo il nostro "Pong"?

- implementando "Pong" in un linguaggio per una "virtual machine"
- tradurremo il linguaggio per "virtual machine" in linguaggio assembly
- tradurremo il linguaggio assembly in linguaggio macchina
- eseguiremo il linguaggio macchina usando un elaboratore elettronico
- realizzando l'elaboratore usando porte logiche NAND

Astrazione / Implementazione che useremo per il nostro "Pong"



Principio di astrazione/implementazione

Astrazione: Si presenta la soluzione ad un problema concentrandosi solo su alcuni aspetti "rilevanti"
- ad esempio, come ottenere la soluzione componendo soluzioni di problemi più semplici

Implementazione: Si realizza la soluzione aggiungendo gli aspetti astratti nella prima fase
- ad esempio, si mostra come si possono risolvere i problemi più semplici

Gli elaboratori digitali sono macchine multilivello

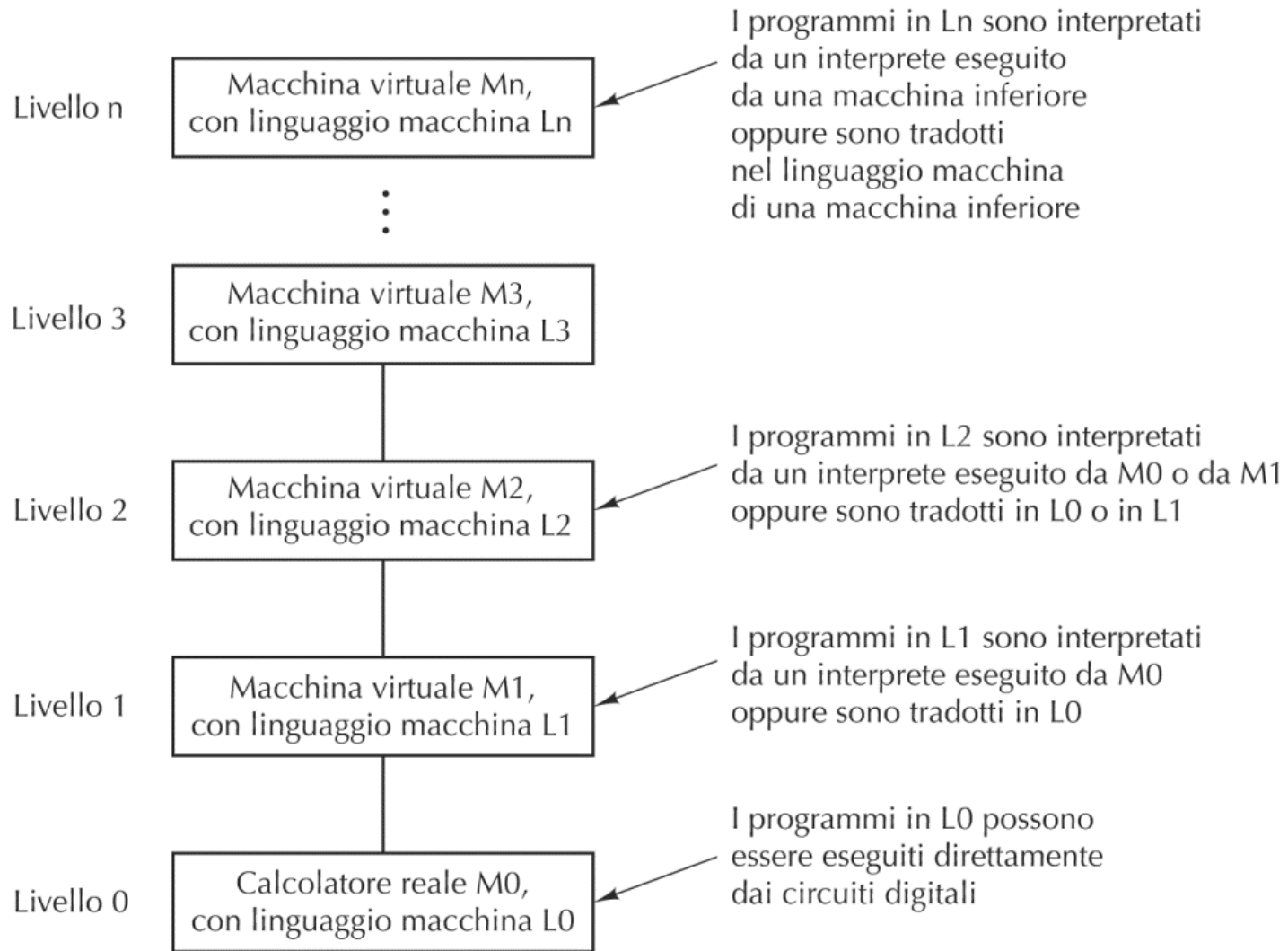


Figura 1.1 Macchina multilivello.

Tipico elaboratore a 6 livelli

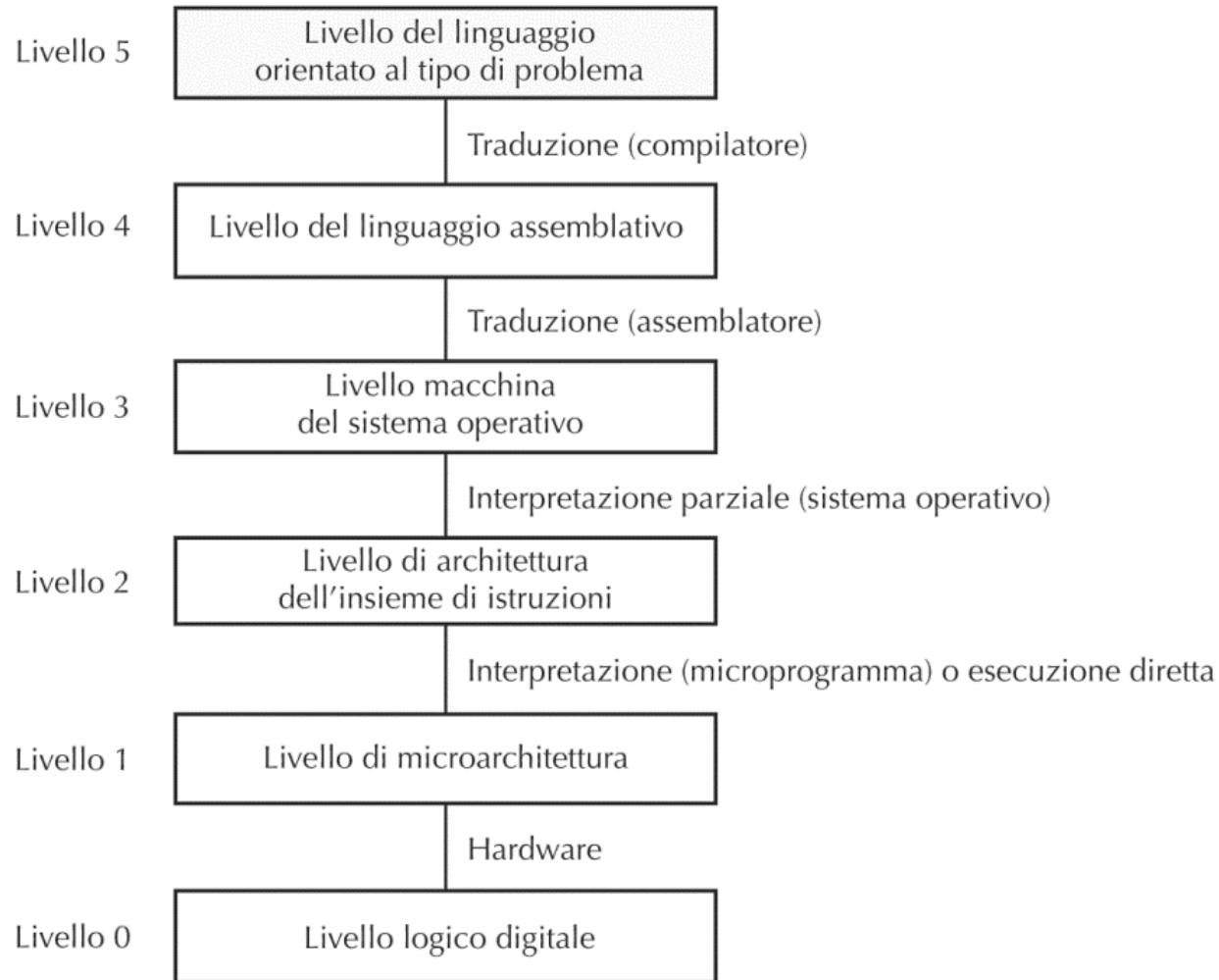
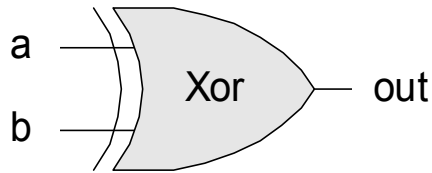


Figura 1.2 Computer a sei livelli. Sotto ciascun livello è indicato il metodo di supporto (oltre al nome del programma corrispondente).

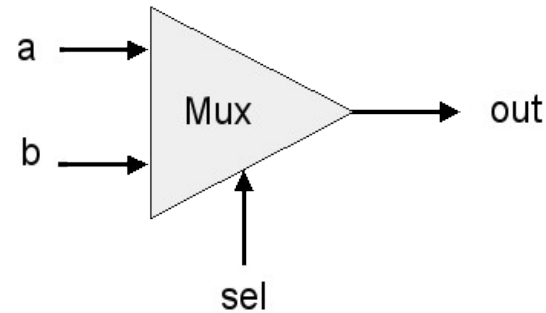
Livello 0: livello logico digitale

Porte logiche:



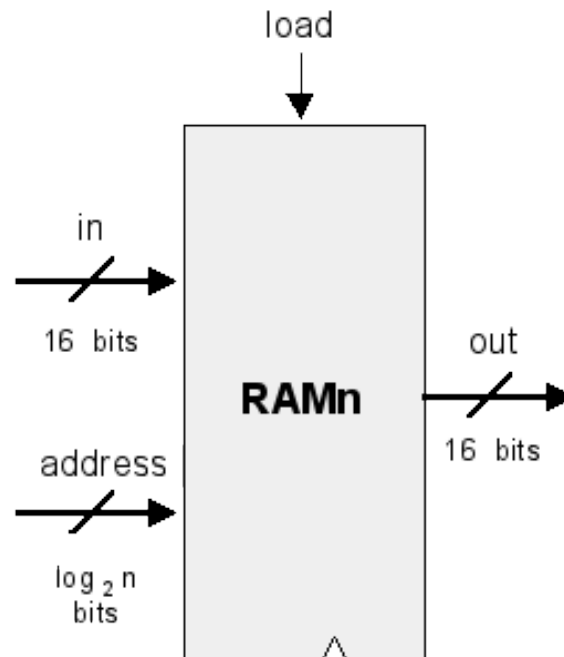
a	b	out
0	0	0
0	1	1
1	0	1
1	1	0

Circuiti combinatori:



a	b	sel	out
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

Circuiti sequenziali:



Livello 2: livello ISA (Instruction Set Architecture)

Istruzioni macchina:

insieme di istruzioni eseguibili dalla microarchitettura

Livello solitamente considerato come

“interfaccia” fra hardware e software

31	2827	1615	87	0	Tipo di istruzione					
Cond	0 0 I	Opcode	S	Rn	Rd	Operand2	Elaborazione dati / Traferimento di PSR			
Cond	0 0 0 0 0 0	A S	Rd	Rn	RS	1 0 0 1	Rm	Moltiplicazione		
Cond	0 0 0 0 1	U A S	RdHi	RdLo	RS	1 0 0 1	Rm	Moltiplicazione Long		
Cond	0 0 0 1 0	B 0 0	Rn	Rd	0 0 0 0	1 0 0 1	Rm	Swap		
Cond	0 1 I	P U	B W L	Rn	Rd	Offset		Load/Store Byte/Word		
Cond	1 0 0	P U	S W L	Rn	Register List			Load/Store Multipla		
Cond	0 0 0	P U	1 W L	Rn	Rd	Offset1	1 S H 1	Offset2	Trasferimento di mezza parola: spiazzamento immediato	
Cond	0 0 0	P U	0 W L	Rn	Rd	0 0 0 0	1 S H 1	Rm	Trasferimento di mezza parola: spiazzamento relativo a registro	
Cond	1 0 1	L	Offset						Salto	
Cond	0 0 0 1	0 0 1 0	1 1 1 1	1 1 1 1	1 1 1 1	0 0 0 1	Rn	Scambio di salto (branch exchange)		
Cond	1 1 0	P U	N W L	Rn	CRd	CPNum	Offset		Trasferimento dati del coprocessore	
Cond	1 1 1 0	Op1	CRn	CRd	CPNum	Op2	0	CRm	Operazione su dati del coprocessore	
Cond	1 1 1 0	Op1	L	CRn	Rd	CPNum	Op2	1	CRm	Trasferimento registri del coprocessore
Cond	1 1 1 1	SWI Number							Interrupt software	

Figura 5.14 Formati delle istruzioni a 32 bit di ARM.

Livello 3-4: Sistema Operativo - Ling. Assemblativo

Livelli "ibridi": non sono così rigidamente separati come i livelli precedenti

Sistema Operativo:
fornisce servizi per gestione risorse (memoria, I/O, ...) ed esecuzione processi

Linguaggio Assemblativo:
permette di programmare i livelli sottostanti (nessuno programma in linguaggio macchina!)

Tabella delle pagine

Pagina virtuale Blocco di memoria

15	0	0
14	1	4
13	0	0
12	0	0
11	1	5
10	0	0
9	0	0
8	1	3
7	0	0
6	1	7
5	1	6
4	0	0
3	1	2
2	0	0
1	1	0
0	1	1

Memoria principale	Blocco di memoria
Pagina virtuale 6	7
Pagina virtuale 5	6
Pagina virtuale 11	5
Pagina virtuale 14	4
Pagina virtuale 8	3
Pagina virtuale 3	2
Pagina virtuale 0	1
Pagina virtuale 1	0

1 = Presente in memoria principale
0 = Assente dalla memoria principale

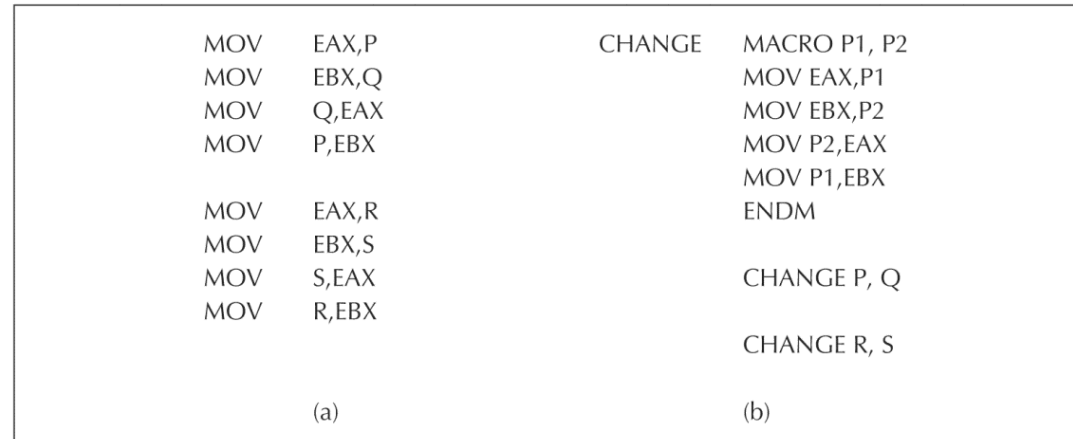


Figura 7.5 Sequenze d'istruzioni quasi identiche. Senza (a) e con (b) macro.

Figura 6.5 Assegnamento di 16 pagine virtuali in una memoria principale composta di otto blocchi.

Conclusioni assumendo di fondere insieme i livelli 3 - 4



Struttura del corso

- Approfondiremo i vari livelli, focalizzandoci sui concetti più interessanti, seguendo il libro:

*"Architettura dei calcolatori. Un approccio strutturale",
Sesta Edizione. Andrew S. Tanenbaum, Todd Austin.
Pearson.*

- In parallelo, svilupperemo l'elaboratore su cui eseguire il nostro "Pong", seguendo il libro:

*"The Elements of Computing Systems".
Noam Nisan and Shimon Schocken. MIT Press.
<http://www.nand2tetris.org>*

Importante: iniziate a installare il software disponibile (in particolare l'Hardware Simulator che useremo per realizzare il livello logico digitale e il CPU Emulator, quello che esegue il nostro "Pong" e gli altri programmi in linguaggio assembly)



Cosa mi aspetto da voi

- Che abbiate una buona idea degli argomenti spiegati nelle lezioni precedenti
 - L'architettura è a livelli, è molto difficile capire un livello senza aver capito il precedente
 - Se avete dubbi o domande chiedete chiarimenti all'inizio della lezione
- Che partecipiate: apprezzo
 - Chi si offre per correggere gli esercizi o partecipa al loro svolgimento con suggerimenti
 - Chi fa domande ragionevoli
- So che seguire le lezioni da remoto non incentiva la partecipazione: io cercherò di stimolarla ma mi serve la vostra collaborazione

Perchè affrontare questo viaggio insieme?

■ Motivi culturali

- L'architettura è alla base dell'esecuzione del software e quindi dell'informatica
- Non avrete molte occasioni di tornare su questi argomenti durante il corso di laurea

■ Motivi economici

- Come informatici dovrete riuscire a trovare lavoro ...
- ... ma le aziende non cercano studenti, cercano studenti bravi ...
- ... e non tutte le aziende sono uguali

- Esame finale: scritto 2 appelli a gennaio-febbraio, 3 a maggio-giugno-luglio, 1 a settembre
 - Possibile parte orale in caso di esame fatto da remoto
- Consegne dei progetti relativi alla parte di laboratorio (lavoro individuale, facoltativo ma altamente consigliato)
 - E' possibile (ma più difficile) superare l'esame senza effettuare le consegne, ma non ottenere il massimo dei voti
 - Stabiliremo di volta in volta la deadline per il progetto
 - Alcune deadline cadranno durante il periodo delle lezioni, altre dopo il termine delle lezioni

Valutazione

- Esame scritto: 27 punti
- Progetto: 6 punti, divisi tra le varie consegne
 - I progetti vanno consegnati entro le scadenze
- 1 punto bonus per chi mostra la sua soluzione di un esercizio (e questa è ragionevolmente corretta)

■ Sul sito:

<https://virtuale.unibo.it/course/view.php?id=30691>

troverete tutto il materiale e gli strumenti per seguire al meglio il corso:

- slide usate durante le lezioni (aggiornate poco prima o poco dopo le lezioni)
- i file usati durante le esercitazioni in aula
- specifiche degli elaborati da sviluppare in autonomia
- lo strumento per effettuare la consegna dei vostri elaborati
- ...

Le lezioni

- Lunedì: 13:00 - 15:00
- Martedì: 13:00 - 16:00
- Venerdì: 12:00 - 14:00

Tonelli (matematica)
Aula magna Filippo Re 10
M1 (mineralogia)