



# Architetture degli elaboratori - I Introduzione



Prof. Alberto Borghese  
Dipartimento di Informatica  
Laboratorio di Sistemi Intelligenti Applicati (AIS-Lab)  
[alberto.borghese@unimi.it](mailto:alberto.borghese@unimi.it)

Università degli Studi di Milano

Patterson & Hennessy: Section 1.12 on the WEB



## Sommario della lezione



- **Informazioni su corso ed esame**
- Architettura dell'elaboratore
- Ciclo di esecuzione di un'istruzione
- Storia dell'elaboratore.



## Obiettivo del corso



- Fornire i fondamenti per capire cosa succede dentro ad un elaboratore.
- Quali sono le problematiche e come viene elaborata l'informazione?
- Qual'è il linguaggio di un elaboratore (ISA)? Come funziona? (programmazione in piccolo).
- **Analisi e progettazione (sintesi).**



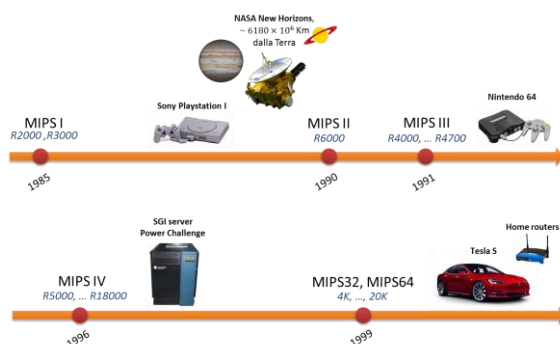
## Architettura base del corso: MIPS R3000



### Features-front



AIBO (Sony, 2003) - MIPS 7000, ARMv8, sistemi embedded che montano Windows CE, PlayStation 2, router, gateway...



Samsung S21  
Exynos ARM Processor





# MIPS



E' un'architettura semplice ma potente. La semplicità dell'architettura emerge anche a livello di Assembler (Architettura II).

"Hello world" in x86 (Intel su 64 bit su Linux)

```
.file "hello_world.c"
.section .rodata
.LC0:
.string "Hello world!"
.text
.globl main
.type main, @function
main:
.LFB0:
.cfi_startproc
pushq %rbp
.cfi_def_cfa_offset 16
.cfi_def_cfa 16, -16
movq %rsp, %rbp
.cfi_def_cfa_register 6
movl $.LC0, %edi
call puts
movl $0, %eax
popq %rbp
.cfi_def_cfa 7, 8
ret
.cfi_endproc
.LFE0:
.size main, -main
.ident "GCC: (Ubuntu 5.4.0-6ubuntu1~16.04.9) 5.4.0 20160609"
.section .note.GNU-stack,"",@progbits
```

"Hello world" in MIPS (32 bit)

```
.data
hello: .asciiz "\nHello, World!\n"

.text
.globl main
main:
li $v0, 4
la $a0, hello
syscall

li $v0, 10
syscall
```



## Capire l'HW per scrivere SW efficace



### Architettura I (dalle porte logiche alla CPU)

- Implementazione HW delle operazioni
- Implementazione HW delle sequenze di operazioni

- Porte logiche ed algebra di Boole
- Circuiti combinatori
- Circuiti sequenziali
- Macchine a stati finiti
- Firmware e micro-programmi
- CPU

### Architettura II

- CPU avanzate
- Gestione delle gerarchie di memoria e memoria virtuale
- Parallelizzazione dell'esecuzione
- Gestione dell'I/O
- Architetture avanzate (GPU, DSA)



# Architettura I - 6 CFU (Cognomi A-G) (Cognomi H-Z Prof. Nicola Basilico)



## Sito principale:

[http://borgnese.di.unimi.it/Teaching/Architettura\\_I/\\_Arch\\_I.html](http://borgnese.di.unimi.it/Teaching/Architettura_I/_Arch_I.html)

## Programma:

[http://borgnese.di.unimi.it/Teaching/Architettura\\_I/Programma\\_2022-2023.html](http://borgnese.di.unimi.it/Teaching/Architettura_I/Programma_2022-2023.html)

## Materiale sensibile su ARIEL.

Let's try to keep the course interactive

## Orario turno I Prof. Borgnese:

Lunedì, lezione, ore 8.30-10.30, aula V3 (via Venezian)

Giovedì, lezione, ore 8.30-10.30, aula V3 (via Venezian)

## Strumento principale di contatto: email ([alberto.borghese@unimi.it](mailto:alberto.borghese@unimi.it))

Ricevimento su appuntamento



## Il sito del corso



Architettura I 2022-2023 AIS-lab

Architetture degli elaboratori - I  
Turno 1: cognomi A-G – Prof. [Alberto Borgnese](#)  
(Il turno 2, cognomi H-Z, verrà erogato dal Prof. [Nicola Basilico](#))

Laboratorio: [Dott.ssa Gabriella Trucco](#) (Turno A: A-FIN), [Dott. Massimo Rivolta](#) (Turno B: FIO-PAL), [Dott. Matteo Re](#) (Turno C: PAM-Z)

Corso di laurea triennale in Informatica, Università di Milano. A.A. 2022-2023. Primo Semestre.

Avvisi: Lo scritto del 27.09.2022 si terrà in aula 405 con inizio alle ore 8.30. Il corso avrà inizio lunedì 3 Ottobre 2022.

**Orario del corso:**  
Lunedì, lezione, ore 8.30-10.30, aula V3 (via Venezian)  
Giovedì, lezione, ore 8.30-10.30, aula V3 (via Venezian)

**Orario del laboratorio:**  
Turno A - Martedì  
Turno B - Mercoledì  
Turno C - Venerdì

**Programma dettagliato** A.A. 2022-2023.  
Per il laboratorio fate riferimento al sito ARIEL del corso.

**Materiale bibliografico del corso, Esercizi.**  
[Program and bibliographic material: in english](#)


Per il programma, temi d'esame e materiale dei corsi degli anni precedenti click [haga](#).

**N.B.:** Il diritto a scaricare il materiale accessibile da questa pagina è riservato solamente agli studenti regolarmente iscritti al corso.  
**Notice:** The right to download the material accessible from this page is granted only to the students regularly enrolled in the hereabove University course.


**Modalità d'esame: scritto + orale + progetto (le votazioni parziali hanno validità di 2 appelli, approssimativamente 6 mesi).** Per dettagli sulla modalità di presentazione dei progetti d'esame, consultare: [Modalità per la presentazione dei progetti](#) nella home page della Dott.ssa Trucco.

**Temi d'esame:**

25 Gennaio 2022	24 Febbraio 2022	23 Giugno 2022	19 Luglio 2022	27 Settembre 2022
21 Gennaio 2021	16 Febbraio 2021	22 Giugno 2021	20 Luglio 2021	27 Settembre 2021



# Programma



**+Programma del corso di Architettura degli Elaboratori - parte I**  
**Programma A.A. 2022-2023**

N.B: Il diritto a scaricare il materiale accessibile da questa pagina è riservato solamente agli studenti regolarmente iscritti al corso.  
Notice: The right to download the material accessible from this page is granted only to the students regularly enrolled in the hereabove University course.

Le lezioni di esercitazione sono riportate in colore rosso, le lezioni di laboratorio in blu e le lezioni frontali in nero.

Le slide sono da considerare bozze avanzate fino al giorno della lezione. Le slide in versione definitiva, saranno disponibili sul sito il giorno dopo la lezione.

Introduzione	
L_01	03/10/2022 Introduzione. L'architettura di riferimento. Il ciclo di esecuzione di un'istruzione. Storia dell'Elaboratore (Prof. Borghese, ultima modifica 08.10.21).
L_02	06/10/2022 Codifica dell'informazione. Operazioni su numeri binari. Le operazioni fondamentali: somma e sottrazione. Rappresentazione binaria dei numeri decimali. <a href="#">Esercizi</a> (Prof. Borghese, ultima modifica 12.10.21).
Logica combinatoria	
L_03	10/10/2022 L'algebra combinatoria: variabili ed operatori. Implementazione circuitale (porte logiche). Dal circuito alla funzione. Algebra Booleana. Le porte universali (Prof. Borghese, ultima modifica 18.10.21)
L_04	11/10/2022 Laboratorio. Codifica dell'informazione numerica: notazione posizionale, cambio di base, somma e sottrazione, complementi a 2, overflow (2 ore)
L_04	13/10/2022 Dalla tabella della verità al circuito: la prima forma canonica. Criteri di ottimalità. Semplificazione algebrica. (Prof. Borghese, ultima modifica 18.10.21).
L_05	18/10/2022 Implementazione circuitale di funzioni logiche mediante PLA e ROM. Circuiti combinatori notevoli. (Prof. Borghese, ultima modifica 21.10.21).
L_02	19/10/2022 Laboratorio -Codifica dell'informazione numerica: rappresentazione dei numeri reali, numeri subnormalizzati (2 ore)
Le unità aritmetico-logiche	
L_06	21/10/2022 Addizione. Anticipazione del riporto (Prof. Borghese, ultima modifica 01.11.21).
L_07	25/10/2022 Moltiplicatori hardware. Progettazione di una ALU. 1 day study. (Prof. Borghese, ultima modifica 01.11.21).
L_03	26/10/2022 Laboratorio - Introduzione a Legosim: presentazione della piattaforma e realizzazione di semplici circuiti combinatori (manipolazioni algebriche) (2 ore)
L_08	28/10/2022 Comparazione e Overflow: Temporizzazione dei circuiti booleani. Circuiti sequenziali. 1 latch SR. (Prof. Borghese, ultima modifica 04.11.21).
L_04	02/11/2022 Laboratorio - SOP <b>FDS (seconda forma canonica)</b> , cammino critico, <b>mappe di Karnaugh</b> (3 ore)
<b>La logica sequenziale</b>	
	31/10/2022 Chimera università per poche Ossessanti
	01/11/2022 Festa di Omisssanti

- Le slide sono solo una traccia, occorre capire in profondità
- Gli argomenti sono collegati gli uni agli altri per tutte e due i corsi.

A.A. 2022-2023
9/70
<http://borghese.di.unimi.it/>



# Materiale didattico



[http://borghese.di.unimi.it/Teaching/Architettura\\_I/References.rtf](http://borghese.di.unimi.it/Teaching/Architettura_I/References.rtf)

**Testo di base (è disponibile sia in inglese che in italiano):**

Struttura e progetto dei calcolatori: l'interfaccia hardware-software, D.A. Patterson and J.L. Hennessy, Quinta edizione, Zanichelli, gennaio 2022. **Edizione MIPS** (Nota: la quinta edizione Zanichelli è la traduzione della sesta edizione inglese).

"Computer Organization & Design: The Hardware/Software Interface", D.A. Patterson and J.L. Hennessy, Morgan Kaufmann Publishers, Sixth Edition, 2020 (**MIPS edition**).

**Il testo copre il contenuto dei corsi di Architettura I e II. Parte del material (appendici) si trova su WEB sul sito dell'editore (sia per la versione italiana che per la versione inglese).**



<http://borghese.di.unimi.it/>



## Materiale didattico



[http://borghese.di.unimi.it/Teaching/Architettura\\_I/References.rtf](http://borghese.di.unimi.it/Teaching/Architettura_I/References.rtf)

**Per un approfondimento sui circuiti combinatori e sequenziali:**

"Progettazione digitale" F. Fummi, M.G. Sami, C. Silvano, McGrawHill. Seconda edizione, 2007.



A.A. 2022-2023

11/70

<http://borghese.di.unimi.it/>



## Non solo teoria

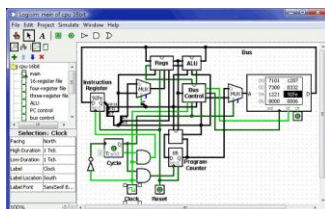


«learn by doing» is equally important -> laboratorio

Dott.ssa Gabriella Trucco (A-FAS)  
[Gabriella.trucco@unimi.it](mailto:Gabriella.trucco@unimi.it)

Dott. Massimo Rivolta (FAT-MOR)  
[Massimo.rivolta@unimi.it](mailto:Massimo.rivolta@unimi.it)

Dott. Matteo Re (MOS-Z)  
[Matteo.Re@unimi.it](mailto:Matteo.Re@unimi.it)



Simulatore di circuiti digitali Logisim:  
<http://ozark.hendrix.edu/~burch/logisim/>

12/70

<http://borghese.di.unimi.it/>



## Modalità di esame



### **Parte teorica (2/3 del voto). Riferimento: Prof. Borghese.**

Prova scritta + orale

- 2 Appelli a Gennaio / Febbraio
- 2 Appelli a Giugno / Luglio
- 1 Appello a Settembre

### **In alternativa:**

2 prove in itinere (compitini) durante l'anno. I compitini sostituiscono interamente scritto e orale. L'orale con i compitini è facoltativo.

**Laboratorio (1/3 del voto).** Progetto di laboratorio in Logisim o prova scritta (verrà deciso nella fase iniziale del corso di laboratorio)



## Studiare è



- Acquisire un quadro generale di un argomento
- Riflettere sui concetti che costituiscono il quadro
- Riflettere sulla relazione tra i concetti
- Sperimentare il funzionamento
- Collegare l'argomento con gli altri argomenti del corso e di altri corsi nonché alle conoscenze pregresse.
  
- lo studio è una **funzione attiva**, occorre pensare.



## Studiare non è



- Imparare a memoria il contenuto delle slide
- Imparare a memoria il contenuto del libro
- Leggere e ripetere
  
- lo studio è una **funzione attiva**, occorre pensare.
  
- La fatica e il tempo in più richiesto si traducono alla fine in tempo risparmiato e in una maggiore soddisfazione.



## Sommario della lezione

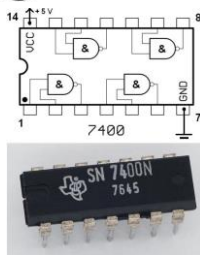


- Informazioni su corso ed esame
- **Architettura dell'elaboratore**
- Ciclo di esecuzione di un'istruzione
- Storia dell'elaboratore.





# Contenuto del corso

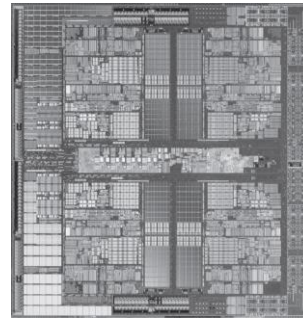
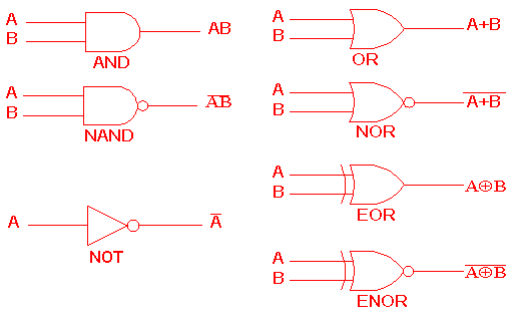


From logic gates to

.... multi-core and GPUs



Salvo stamp  
Newsletter  
Recensioni  
Case Stud  
Flussi RSS



A.A. 2022-2023

17/70

<http://borghese.di.unimi.it/>



# Le architetture



La casa



A.A. 2022-2023

18/70

<http://borghese.di.unimi.it/>



# Le architetture

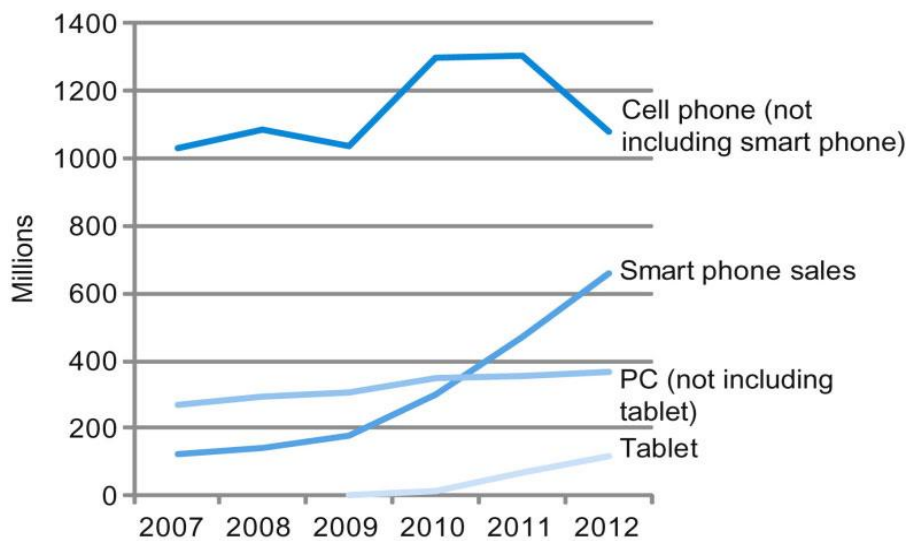


A.A. 2022-2023

19/70



# I calcolatori nel mondo: verso l'era PostPC



A.A. 2022-2023

20/70

<http://borgnese.di.unimi.it/>



## La legge di Moore



Year	Name	Size (cu. ft.)	Power (watts)	Performance (adds/sec)	Memory (KB)	Price	Price-performance vs. UNIVAC	Adjusted price (2003 \$)	Adjusted price-performance vs. UNIVAC
1951	UNIVAC I	1,000	125,000	2,000	48	\$1,000,000	1	\$6,107,600	1
1964	IBM S/360 model 50	60	10,000	500,000	64	\$1,000,000	263	\$4,792,300	318
1965	PDP-8	8	500	330,000	4	\$16,000	10,855	\$75,390	13,135
1976	Cray-1	58	60,000	166,000,000	32,000	\$4,000,000	21,842	\$10,756,800	51,604
1981	IBM PC	1	150	240,000	256	\$3,000	42,105	\$5,461	154,673
1991	HP 9000/ model 750	2	500	50,000,000	16,384	\$7,400	3,556,188	\$9,401	16,122,356
1996	Intel PPro PC (200 MHz)	2	500	400,000,000	16,384	\$4,400	47,846,890	\$4,945	239,078,908
2003	Intel Pentium 4 PC (3.0 GHz)	2	500	6,000,000,000	262,144	\$1,600	1,875,000,000	\$1,600	11,452,000,000

In circa 18 mesi raddoppiano le prestazioni ed il numero di transistor e raddoppiano le capacità delle memorie (DRAM). [Legge di Moore](#).

La **velocità** di accesso alla memoria cresce molto più lentamente.

Bad news: la legge di Moore non è più valida! => Domain Specific Architectures

A.A. 2022-2023

21/70

<http://borghese.di.unimi.it/>



## Determinanti della legge di Moore



Il primo circuito integrato nel 1961 conteneva 4 (**quattro!**) transistor. Nel 1965 erano già 64 transistor e nel 1975 erano 32,000. In un Core i7 del 2012 si trovano **1,4 miliardi** di transistor.

Nel 2014 sono stati prodotti  $250 \times 10^{18}$  transistor (250 miliardi di miliardi, 25 volte il numero di stelle della via lattea e 75 volte il numero di galassie dell'Universo conosciuto). Ogni secondo vengono prodotti 8 miliardi di transistor. Più transistor nel 2014 che fino al 2011.

Abbiamo incontrato la barriera dell'energia e siamo nell'era postPC. I programmi devono essere efficienti anche in senso energetico. Occorre che consumino poca energia => **Come possiamo aumentare il numero di transistor, consumare poca energia e aumentare le prestazioni?** Conoscere l'organizzazione dei calcolatori.

La legge di Moore riguardava il numero di transistor che possono essere impacchettati tale per cui il costo per transistor è minimo (c'è un guadagno di scala all'aumentare del numero di transistor fino ad un certo valore, ma oltre questo valore i difetti rendono la produzione meno vantaggiosa)

I fattori previsti da Moore erano:

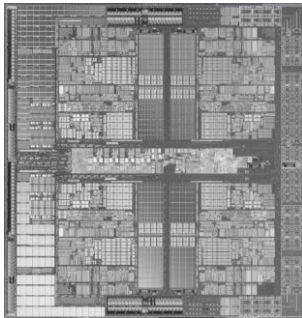
- Aumento della dimensione dei chip (più transistor per chip)
- Diminuzione delle dimensioni (chip più piccoli, aumento del numero di chip, integrazione di chip)
- «Device cleverness» (multi-core)



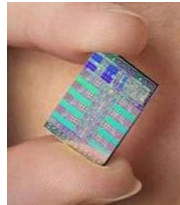
Da IEEE Spectrum, April 2015



## Architetture recenti



AMD Barcelona  
(quad-core)



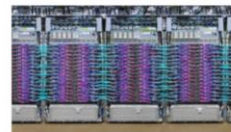
Cell processor  
(IBM, Sony, Toshiba)  
Playstation 3



Cloud TPU v3  
420 teraflops  
128 GB HBM



nVidia 9800 GTX,  
Streaming processors  
128 core



Cloud TPU v3 Pod  
100+ petaflops  
32 TB HBM  
2-D toroidal mesh network



## Obiettivo di un'architettura



Elabora in modo adeguato un input per produrre l'output.

- Le unità di *ingresso* (tastiera, mouse, rete, interfacce con dispositivi di acquisizione, ecc.) permettono al calcolatore di acquisire informazioni dall'ambiente esterno.
- L'architettura di elaborazione.
- Le unità di *uscita* (terminale grafico, stampanti, rete, ecc.) consentono al calcolatore di comunicare i risultati ottenuti dall'elaborazione all'ambiente esterno.





## Cosa fa un elaboratore?



- Algoritmi (sequenza di istruzioni).  
Calcoli (calcolatore).  
**Operazioni logiche** (elaboratore).

- Programma (Ada Byron Lovelace, 1830) = *Algoritmi in Software*.



**Come lo fa?** *Hardware*.

Input ==> Elaborazione ==> Output

- Terza rivoluzione della nostra civiltà: la rivoluzione agricola, la rivoluzione industriale e la rivoluzione dell'informatica.



## Operazioni elementari e codifica dell'informazione



Operazioni elementari necessarie ad eseguire algoritmi:

Calcolo (somma, sottrazione, prodotto....)

Controllo del flusso (if, for....)

L'informazione viene rappresentata utilizzando solamente due simboli (base 2: 0,1 -> acceso, spento). Ogni elemento (cifra) può assumere solo due valori: bit (binary digit)

I calcoli ed i controlli sono eseguiti utilizzando **esclusivamente!** le 3 operazioni fondamentali della logica classica: AND, OR, NOT.



# I principi delle Architetture



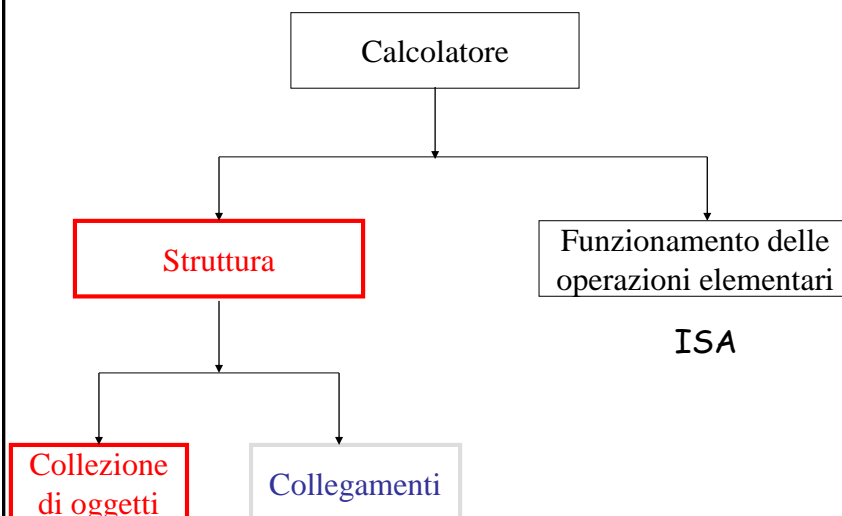
Turing: “Universal Turing machine” (1936). Macchina di esecuzione di algoritmi universale.

I principi come sono stati codificati da Von Neumann negli anni 40.

- Dato che il dispositivo è essenzialmente una macchina di calcolo, ci sarà un'unità che è devota essenzialmente ai calcoli (ALU).
- I dati e le istruzioni sono memorizzate **separatamente** in una memoria read/write.
- Ci sarà una parte che gestisce tutto il sistema di elaborazione: trasferimento dei dati, comanda le operazioni, comanda I/O. Livello gerarchico superiore: UC.
- Un computer deve essere collegato all'esterno. Occorre quindi un equipaggiamento per l'I/O.
- Il contenuto della memoria può essere recuperato in base alla sua posizione (**indirizzo**), e non è funzione del tipo di dato.
- L'esecuzione procede **sequenzialmente** da un'istruzione alla seguente (algoritmo, sequenza di passi...). Nelle architetture più avanzate l'esecuzione procede sequenzialmente per gruppi di istruzioni.

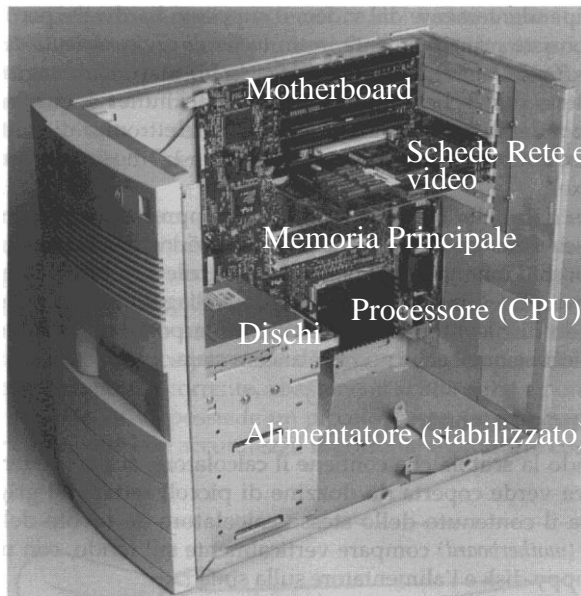


# Descrizione di un elaboratore





# Struttura dell'elaboratore



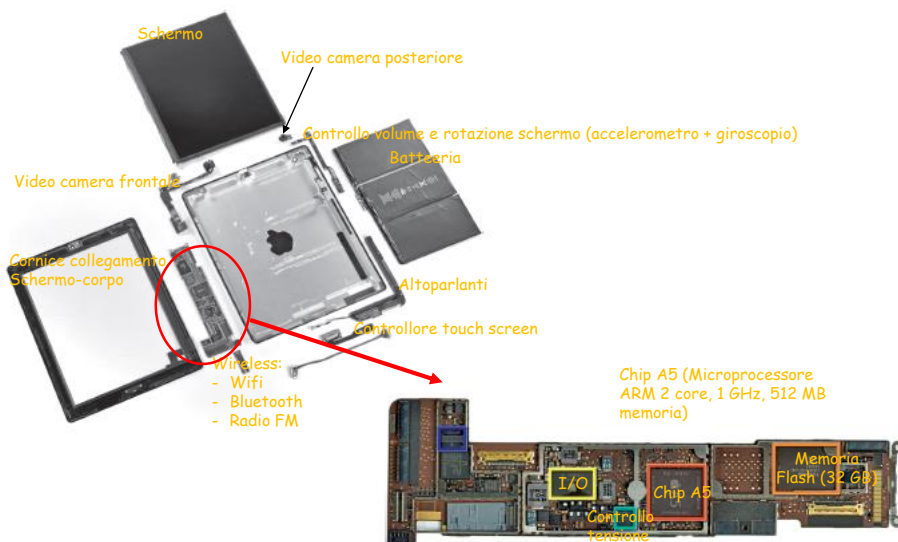
A.A. 2022-2023

29/70

<http://borghese.di.unimi.it/>



# Struttura di un PMD - I-Pad2



A.A. 2022-2023

30/70

<http://borghese.di.unimi.it/>



## Struttura dell'elaboratore: descrizione



- Elementi principali di un elaboratore:
  - ◆ Unità centrale di elaborazione (*Central Processing Unit - CPU*).
  - ◆ Memoria di lavoro o memoria principale (*Main Memory - MM*) e dischi.
- Sulla motherboard: collegamenti principali di un calcolatore:
  - ◆ Bus di sistema (dati, indirizzi, controllo)
  - ◆ Bus di I/O (USB, Firewire): interfacce per i dispositivi di *Input/Output - I/O*: memoria di massa (dischi magnetici o a stato solido, pen drive), rete, altri dispositivi.
- In alternative i collegamenti attraverso i bridge



## Unità centrale di elaborazione (*Central Processing Unit - CPU*)



- La *CPU* provvede ad eseguire le istruzioni che costituiscono i diversi programmi elaborati dal calcolatore.
- Eseguire un'istruzione vuol dire operare delle scelte, eseguire dei calcoli a seconda dell'istruzione e dei dati a disposizione.





## Elementi principali della CPU



- Banco di registri (*Register File*) ad accesso rapido, in cui memorizzare i dati di utilizzo più frequente. Il tempo di accesso ai registri è circa 10 volte più veloce del tempo di accesso alla memoria principale. Il register file è evoluto in cache + registri.
- Registro *Program counter (PC)*. Contiene l'indirizzo dell'istruzione corrente da aggiornare durante l'evoluzione del programma, in modo da prelevare dalla memoria la corretta sequenza di istruzione;
- Registro *Instruction Register (IR)*. Contiene l'istruzione in corso di esecuzione.
- Unità per l'esecuzione delle operazioni aritmetico-logiche (*Arithmetic Logic Unit - ALU*). I dati forniti all'*ALU* provengono direttamente da registri interni alla CPU. Possono provenire anche dalla memoria, ma in questo caso devono essere prima trasferiti in registri interni alla CPU. Dipende dalle modalità di indirizzamento previste;
- Unità aggiuntive per elaborazioni particolari come unità aritmetiche per dati in virgola mobile (*Floating Point Unit - FPU*), sommatore ausiliari, ecc.;
- **Unità di controllo**. Controlla il flusso e determina le operazioni di ciascun blocco.



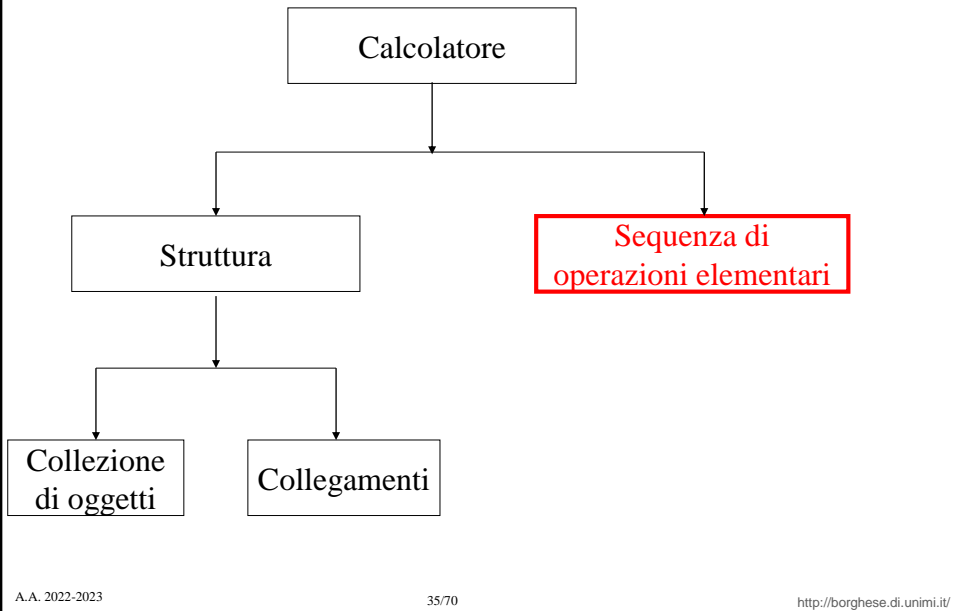
## Sommario della lezione



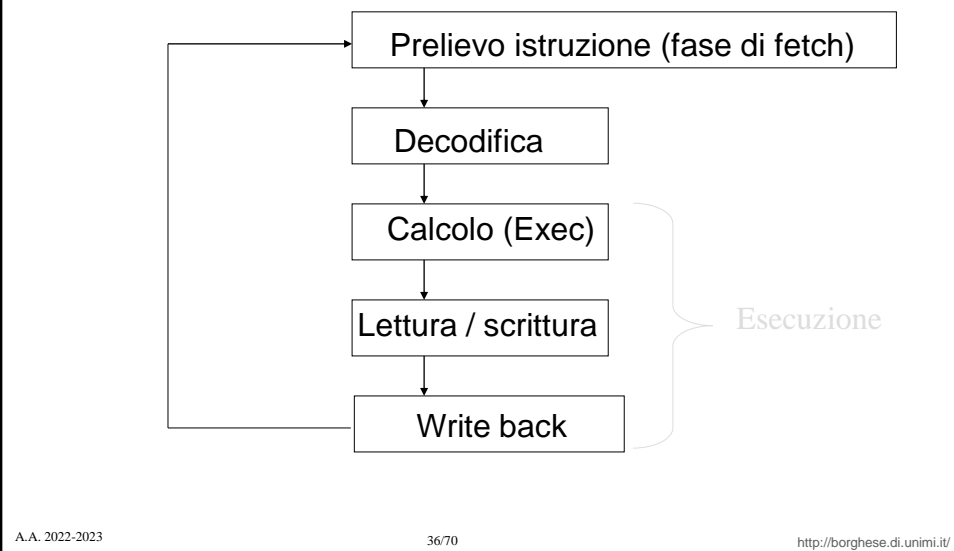
- Informazioni su corso ed esame
- Architettura dell'elaboratore
- **Ciclo di esecuzione di un'istruzione**
- Storia dell'elaboratore.



## Descrizione di un elaboratore



## Ciclo di esecuzione di un'istruzione MIPS







## Decodifica dell'istruzione



- L'istruzione contenuta nel registro IR viene decodificata per essere eseguita. Alla fase di decodifica corrisponde la predisposizione della CPU (apertura delle vie di comunicazione appropriate) all'esecuzione dell'istruzione.
- In questa fase vengono anche recuperati gli operandi. Nelle architetture MIPS gli operandi possono essere solamente nel Register File oppure letti dalla memoria.



## Esecuzione



Viene selezionato il circuito / i circuiti appropriati per l'esecuzione delle operazioni previste dall'istruzione e determinate in fase di decodifica.

L'esecuzione può prevedere: calcolo, interazione con la memoria, controllo di flusso.



## Scrittura in register file (write-back)



- Il risultato dell'operazione può essere memorizzato nei registri ad uso generale oppure in memoria.
- Non appena è terminato il ciclo di esecuzione dell'istruzione corrente (termina la fase di Write Back), si preleva l'istruzione successiva dalla memoria.



## Esempio di istruzione



Somma:           0x80000:  addi \$s3, \$s2, 4  
  00100010010100011000000000000100

Somma il contenuto del registro \$s2 con la costante 4 e scrivi il risultato nel registro \$s3

Fase di fetch:   Caricamento dell'istruzione dall'indirizzo 0x80000.

Decodifica:     Preparazione della CPU a svolgere una somma.  
                          Determinazione dei segnali di controllo.  
                          Lettura degli operandi (operando 1 nel registro \$s2,  
  operando 2 è la costante 4).

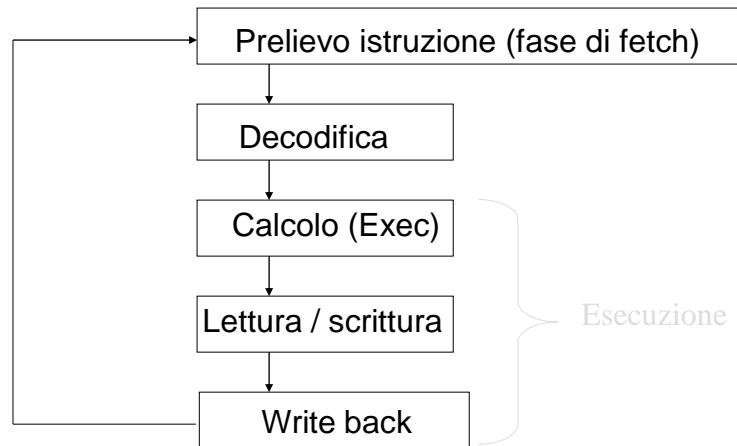
Esecuzione:     Esecuzione della somma.

Memoria:        *Nulla*

Write-back:     Scrittura del registro \$s3.



## Ciclo di esecuzione di un'istruzione MIPS



## Sommario della lezione



- Informazioni su corso ed esame
- Architettura dell'elaboratore
- Ciclo di esecuzione di un'istruzione
- **Storia dell'elaboratore.**



## Storia dell'elaboratore



### Filo conduttore:

Aumento della velocità di elaborazione

Diminuzione della dimensione dei componenti.

Aumento della capacità e velocità dell'I/O.

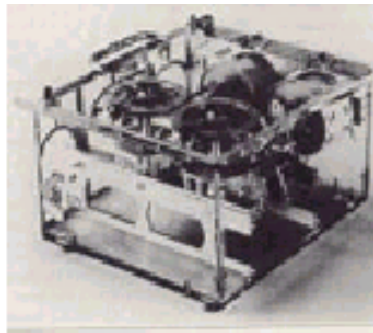
Adozione di tecnologie diverse (meccanica, elettrica, elettronica).



## Storia del calcolatore (i primi passi)



- Abaco, Babilonesi, X secolo a.C.
- B. Pascal (Pascalina, somma e sottrazione).



- G. von Leibnitz (moltiplicazioni e divisioni come addizioni ripetute).



## Le calcolatrici



- Sviluppo di calcolatrici da tavolo meccaniche (diffusione nel commercio).



Millionaire, Steiger, 1892

Moltiplicazioni in un  
“colpo di manovella”.



- Texas Instruments (1972) – prima calcolatrice tascabile.

A.A. 2022-2023

47/70

<http://borghese.di.unimi.it/>



## Un'architettura efficace



*Una macchina per risolvere un problema industriale.*

***Telaio Jacquard (1801)***

- Programma di lavoro su schede
- Macchina dedicata (antesignana delle macchine CAM).



A.A. 2022-2023

48/70

<http://borghese.di.unimi.it/>





## Charles Babbage



Le prime architetture furono pneumatiche

### *Charles Babbage*

- Papà del calcolatore moderno.
- “Analytical Engine” i comandi erano a vapore!
- Utilizza il concetto di programma su (su schede) proposto da Ada Lovelace (1830).



## Nasce l'IBM (1900-1930)



- Non solo architettura.....
- H. Hollerith: Schede perforate a lettura elettromeccanica (relais) combinato con il «Millionaire»

Meccanismo più semplice di gestione del controllo.

*Nel 1890, 46,804 macchine censirono 62,979,766 persone in pochi giorni.  
Il censimento precedente, del 1870, durò 7 anni!!*

- T.J. Watson rilevò il brevetto e fondò l' IBM fondendo la società di Hollerith con altre piccole società (1932).

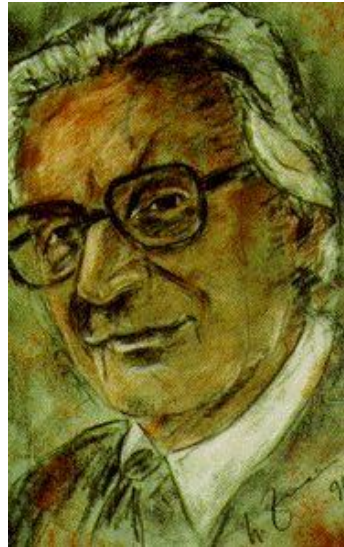


## Il papà non riconosciuto



Konrad Zuse, 1936  
Ingegnere civile.

Z1 -> 1938  
**Z3 -> 1941**



Auto-ritratto del 1994



## Storia dell'elaboratore - Mark I - 1944

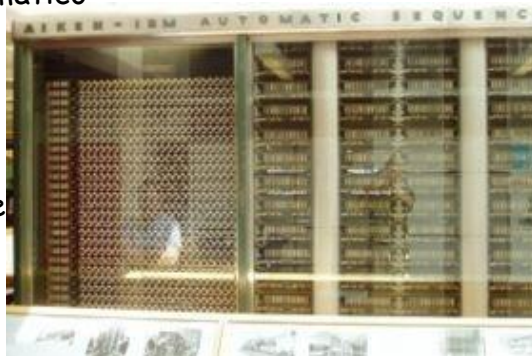


Primo computer automatico

Elettromeccanico

15.3s per divisione  
6s per moltiplicazione

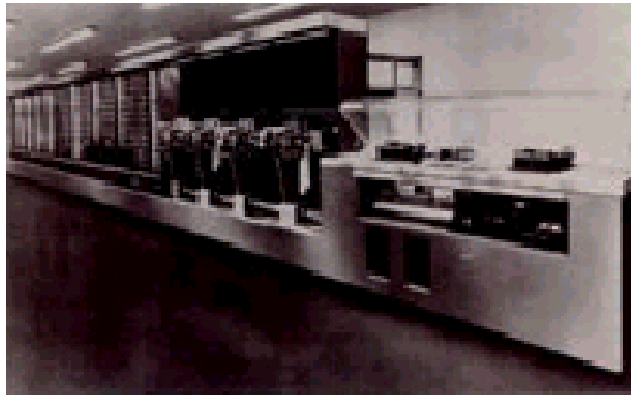
OpCode + operandi



**Automatic Sequence Controlled Calculator - H. Aiken, IBM**



## Storia dell'elaboratore (IIa Guerra mondiale)



- ABC - Atanasoff Berry Computer (University of Iowa). Ampio utilizzo di elettrovalvole. Memoria rigenerativa (cancellabile e riscrivibile). Non funzionò mai completamente

A.A. 2022-2023

53/70

<http://borghese.di.unimi.it/>



## La prima generazione (ENIAC: 1946-1955)

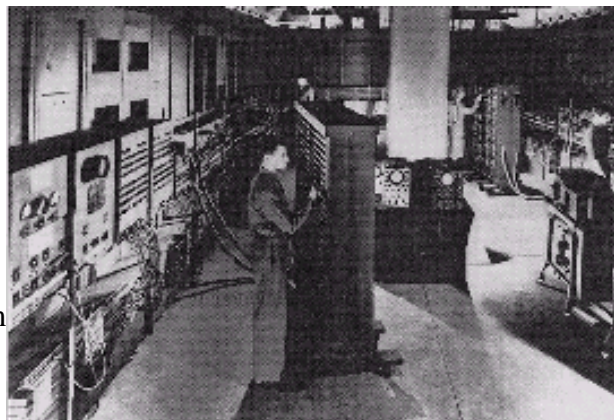


Elettronica (valvole: diodo, triodo). Aumento di prestazioni di 1,000 volte.

- ENIAC (Electronic Numerical Integrator And Calculator), University of Pennsylvania.

### *Caratteristiche:*

- 20 registri da 10 cifre.
- 18,000 valvole.
- 70,000 resistenze.
- 10,000 condensatori.
- 6,000 interruttori.
- Dimensioni: 30mx2.5m
- Consumo: 140kW.
- 100 operazioni/s.
- 30 tonnellate.



- Il programma veniva realizzato cambiando manualmente il cablaggio.



## Defining characteristics of five early digital computers



Computer	First operation	Place	Decimal /Binary	Elect ronic	Programmabi le	Turing complete
<u>Zuse Z3</u>	May 1941	<u>Germany</u>	binary	No	By punched film stock	Yes (1998)
<u>Atanasoff–Berry Computer</u>	Summer 1941	<u>USA</u>	binary	Yes	No	No
<u>Colossus</u>	December 1943 / January 1944	<u>UK</u>	binary	Yes	Partially, by rewiring	No
<u>Harvard Mark I – IBM ASCC</u>	1944	<u>USA</u>	decimal	No	By punched paper tape	Yes (1998)
<u>ENIAC</u>	1944	<u>USA</u>	decimal	Yes	Partially, by rewiring	Yes
	1948	<u>USA</u>	decimal	Yes	By Function Table <u>ROM</u>	Yes

A.A. 2022-2023

55/70

<http://borghese.di.unimi.it/>



## Eckbert & Mauchly

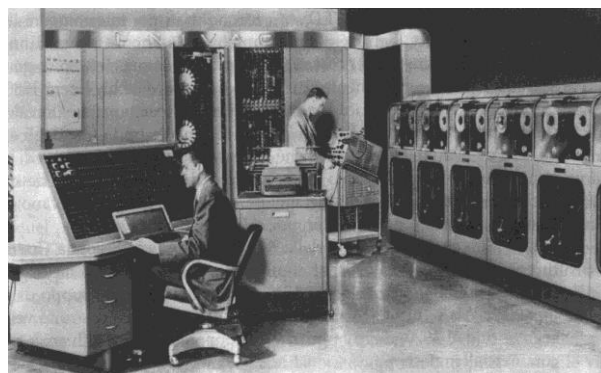


- **EDVAC**, Eckbert, Mauchly, **Von Neuman**. Moore school, Pennsylvania University. **Programma memorizzato**.

- **EDSAC**, Eckert, Cambridge, 1949, (=> Mark I, 1948).

- **UNIVAC I** (Universal Automatic Computer) I (1951), Eckbert e Mauchly. E' il primo calcolatore commercializzato.

48 esemplari a 1M\$



A.A. 2022-2023

56/70

<http://homes.dsi.unimi.it/~borghese/>



## La seconda generazione (1952- 1963)



- Introduzione dell'elettronica allo stato solido.
- Introduzione delle memorie ferromagnetiche.

IBM:

- Modello 701 – 1953 per calcolo scientifico.
- Modello 702 – 1955 per applicazioni gestionali

- IBM704 - Memoria con nuclei di ferrite: 32,000 parole e velocità di commutazione di pochi microsecondi = qualche kHz).

- IBM709 nel 1958 - Introduzione del “canale” di I/O.
- IBM 7094 (1962) Introduzione della formalizzazione del controllo di flusso.

- Introduzione del Fortran (Formula Translator).

CDC:

- CDC 6600 - Primo supercalcolatore. 1962.
- CDC 3600 - Multi-programmazione. 1963.

Digital equipment

- PDP - 1

A.A. 2022-2023

57/70

<http://homes.dsi.unimi.it/~borghese/>



## La comunicazione tra i componenti



Switch centralizzato

(multiplexor -> bridge)

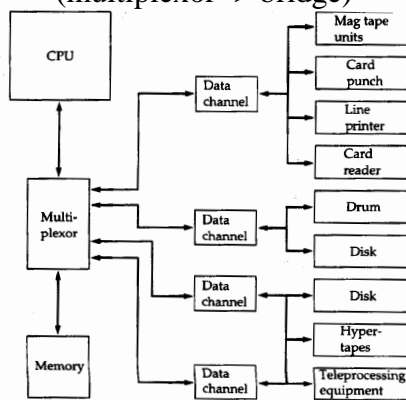
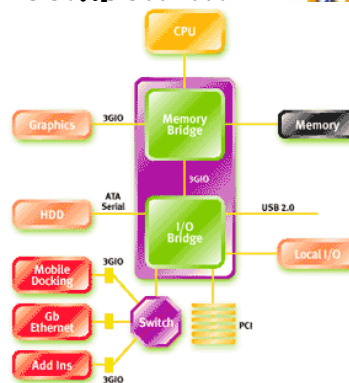


Figure 2.5 An IBM 7094 Configuration

Programma di “canale”



Architettura a nodo comune  
(a bus, cf. bus PCI)

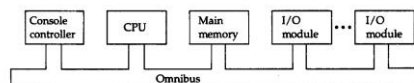


Figure 2.9 PDP-8 Bus Structure

A.A. 2022-2023

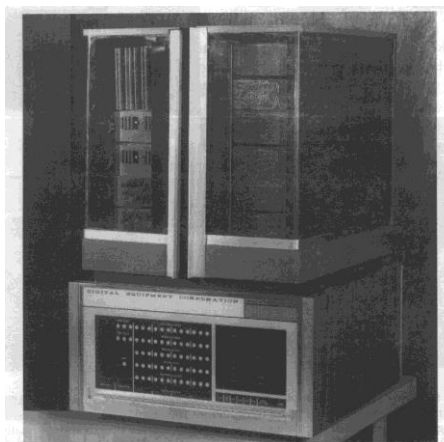
58/70



## La terza generazione (1964-1971)



- Introduzione dei circuiti integrati (LSI).
- IBM360 (1964) - Prima famiglia di calcolatori (architettura di calcolatori). Costo 360,000\$  
Registri a 32 bit.  
Clock 1-4Mhz.
- Digital PDP-8 (1965) - Il primo minicalcolatore.  
Costo < 20,000\$.
- PDP-11 (1970).



A.A. 2022-2023

59/70

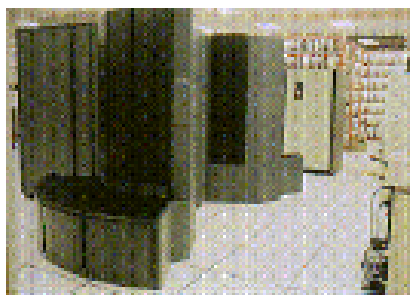
<http://borghese.di.unimi.it/>



## La quarta generazione (1971-1977)



- Cray I (1976) - Primo supercalcolatore. Vettoriale (cf. SIMD)



A.A. 2022-2023

60/70

<http://homes.dsi.unimi.it/~borghese/>



## La quarta generazione (1971-1977)



- Introduzione del microprocessore (VLSI).
- Memorie a semiconduttori.
- Intel 4004 (1971, F. Faggin) - 2,300 transistor. Sommatore a 4 bit. 16 registri a 4 bit + RAM + ROM -> Sistema MCS-4.
- Intel 8080 (1974) - 8bit su chip.

### Xerox research laboratories & Steve Jobs

Primo Personal Computer:  
MacIntosh II di Apple Computer  
(1977).

Sistema operativo a finestre:  
Lisa (1984), MacIntosh II, 1985.  
Processore Motorola.  
Costo medio 2,000\$.



A.A. 2022-2023

61/70

<http://borghese.di.unimi.it/>



## La quinta generazione: i PC (1978-2003)



- Il primo PC (1981) IBM
  - Sistema operativo DOS (Microsoft di Bill Gates).
  - Processore Intel 8086.
  - Windows 1.0 nel 1987.
  - Coprocessore Matematico Intel 8087.
- PC come Workstation
  - Potenziamento della grafica. Coprocessore grafico (acceleratori).
  - Introduzione di elaborazione parallela (multi-threading) con esecuzione parzialmente sovrapposta (pipeline).
  - Processori RISC (Reduced Instruction Set Code).
  - MMU (Unità intelligenti per la gestione della memoria).
  - Definizione di GL -> OpenGL (Workstation Silicon Graphics)



SGI - Indigo2

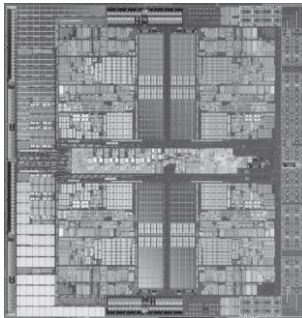
A.A. 2022-2023

62/70

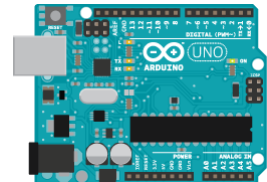
<http://borghese.di.unimi.it/>



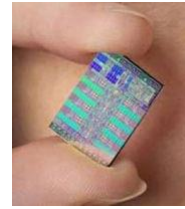
## La sesta generazione (postPC)



AMD Barcelona  
(quad-core)



Arduino microcontroller  
(M. Banzi)



Cell processor  
(IBM, Sony, Toshiba)  
Playstation 3

nVidia 9800 GTX, Streaming  
processors 128 core



A.A. 2022-2023

63/70

<http://borghese.di.unimi.it/>



## Caratteristiche della sesta generazione



- Attualmente la frequenza di clock limite è 4Ghz: barriera dell'energia.
- Rivoluzione del parallelismo: la soluzione è quella di utilizzare diversi microprocessori (core) più piccoli e veloci.
  - ◆ Cell (IBM, Sony, Toshiba): 9-core microprocessors, 2006 (playstation 3, Sony).
  - ◆ Multi-core (Core2 Intel, AMD Barcelona...)
  - ◆ Schede grafiche di ATI e Nvidia (dal 2000) → CUDA programming language
  - ◆ Settembre 2006. Prototipo Intel con 80 processori on single chip. Obiettivo è raggiungere 1,000,000 Mflops.
- **Come?**
  - ◆ Parallelizzazione del codice. (e.g. RapidMind Development Platform).
  - ◆ Nuovo modo di ragionare durante la programmazione software.
  - ◆ Tool di aiuto.
  - ◆ Parallelizzazione automatica del codice è ancora molto lontana.
  - ◆ Problema principale è la coerenza dei dati.

A.A. 2022-2023

64/70

<http://borghese.di.unimi.it/>





# Il futuro



- **Integrazione dei media.**
- **Wearable devices**
- PC + telefono                      Calcolatori ottici.
- Wearable PC                        Calcolatori chimici.
- Co-processor on-board, specializzati per:
  - Ricerca in data-base.
  - Genomica.
  - Machine learning (it is a reality!)
- Macchine intelligenti e sensibili.
- Sistemi multimediali.



# Il futuro

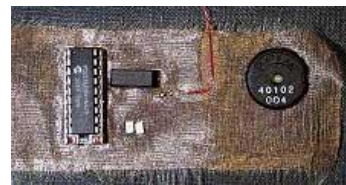


Cloud TPU v3  
420 teraflops  
128 GB HBM

- **Pervasive computing**
- **Dedicated architectures**



E-textile



Circuito con CPU stampato su stoffa



Cloud TPU v3 Pod  
100+ petaflops  
32 TB HBM  
2-D toroidal mesh network



Smart watch 5



## Classificazione dei calcolatori



- Centri di calcolo (Google, ...)
- Cluster (gruppi di calcolatori che lavorano per risolvere un problema complesso).
- Server (calcolatore in grado di eseguire un gran numero di processi in un'unità di tempo).
- Workstation
- Fissi (desktop)
- Portatili (laptop)
- Palmari.
- Smart phone: I-Phone, Blackberry...
- Microcontrollori (micro-architetture: Arduino, Raspberry PI,...)
- FPGA (architetture digitali programmabili)



## Alcuni problemi



La velocità delle memorie non cresce con la velocità del processore.

Memorie gerarchiche – cache.

Aumento della parola di memoria.

high-speed bus (gerarchie di bus).

Tecniche di velocizzazione dell'elaborazione.

Predizione dei salti.

Scheduling ottimale delle istruzioni (analisi dei segmenti di codice).

Esecuzione speculativa.

Tecniche di I/O.

UDP.

Trasferimento in streaming (DMA).

Architetture dedicate alla grafica (GPU)



## Caratteristiche comuni



Architettura di riferimento (Von Neuman)

Ciclo di *esecuzione* delle istruzioni



## Sommario della lezione



- Informazioni su corso ed esame
- Architettura dell'elaboratore
- Ciclo di esecuzione di un'istruzione
- Storia dell'elaboratore.