



# 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

A.A. 2023-2024

1/70

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



## Sommario della lezione



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

A.A. 2023-2024

2/70

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



## Obbiettivo 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).
- **Sviluppo della capacità di analisi e progettazione (sintesi).**

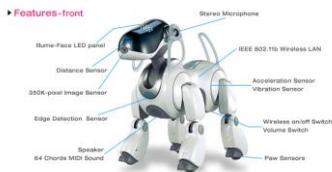
A.A. 2023-2024

3/70

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



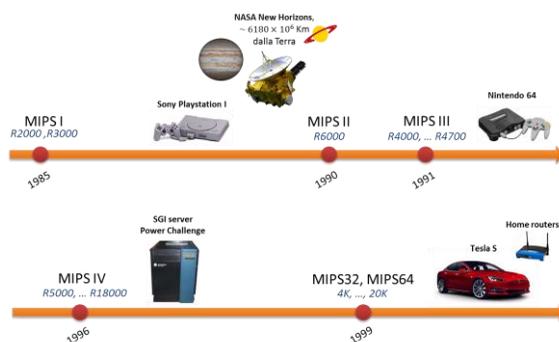
## Architettura base del corso: MIPS R3000



MIPS 7000, ARMv8, sistemi embedded che montano Windows CE, PlayStation 2, router, gateway...



Samsung S21



Exynos ARM Processor



A.A. 2023-2024

4/70

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



# MIPS



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

"Hello world" in Assembler x86 (SO 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_offset 6, -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 di base

### 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\\_2023-2024.html](http://borgnese.di.unimi.it/Teaching/Architettura_I/Programma_2023-2024.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 G11

Giovedì, lezione, ore 8.30-10.30, aula G11

**Strumento principale di contatto: email (alberto.borgnese@unimi.it)**

**Ricevimento su appuntamento**

A.A. 2023-2024

7/70

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



## Il sito del corso



**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), Dott. Massimo Rivolta (Turno B), Dott. Matteo Re (Turno C)**

**Corso di laurea triennale in Informatica, Università di Milano, A.A. 2023-2024, Primo Semestre.**

**Avviso: Il corso inizierà Lunedì 2 Ottobre 2023. La prova scritta dell'Appello del 20 Settembre 2023 si terrà in Aula Chioma, Dipartimento di Matematica, via Saldaia 50, con inizio alle ore 8.30.**

**Orario del corso:**  
Lunedì, lezione, ore 8.30-10.30, aula G11  
Giovedì, lezione, ore 8.30-10.30, aula G11

**Orario del laboratorio:**  
Turno A - Guardare la pagina del laboratorio  
Turno B - Guardare la pagina del laboratorio  
Turno C - Guardare la pagina del laboratorio  
Per il laboratorio fare riferimento al sito ARIEL del corso.

**Programma dettagliato A.A. 2023-2024.**

**Materiale bibliografico del corso. Esamisti**

**Program and bibliography material in english**

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

**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 bachelor's University course.

**Modalità d'esame: scritto + orale (per la parte di teoria). Le votazioni parziali hanno validità di 2 appelli, approssimativamente 6 mesi!** Per dettagli sulla modalità della parte di laboratorio, consultare il sito della Dott.ssa Trucco

Temi d'esame:					
21 Gennaio 2023	22 Febbraio 2023	21 Giugno 2023	26 Luglio 2023	20 Settembre 2023	
25 Gennaio 2022	24 Febbraio 2022	23 Giugno 2022	19 Luglio 2022	27 Settembre 2022	
21 Gennaio 2021	18 Febbraio 2021 ore 9.00	22 Giugno 2021 ore 14.30	20 luglio 2021 ore 14.30	21 Settembre 2021 ore 14.00	
23 Gennaio 2020	20 Febbraio 2020	20 Giugno 2020	23 Luglio 2020	22 Settembre 2020	
18 Gennaio 2019	17 Febbraio 2019	20 Giugno 2019	17 Luglio 2019	18 Settembre 2019	
15 Gennaio 2018	28 Febbraio 2018	21 Giugno 2018	28 Luglio 2018	27 Settembre 2018	
27 Gennaio 2017	24 Febbraio 2017	15 Giugno 2017	25 luglio 2017	28 Settembre 2017	
28 Gennaio 2016	24 Febbraio 2016	14 Giugno 2016	24 Luglio 2016	28 Settembre 2016	
29 Gennaio 2015	27 Febbraio 2015	24 Giugno 2015	21 Luglio 2015	21 Settembre 2015	
28 Gennaio 2014	27 Febbraio 2014	26 Giugno 2014	24 luglio 2014	30 Settembre 2014	
18 Febbraio 2013	28 Febbraio 2013	25 Giugno 2013	23 Luglio 2013	26 Settembre 2013	
21 Gennaio 2012	28 Feb 2012	25 Giu 2012	23.07.2012	24.09.2012	

A.A. 2023-2024

8/70

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



# Programma



Programma del corso di Architettura degli Elaboratori - parte I  
Programma A.A. 2023-2024

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 laboratory 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.

Data	Contenuto della lezione
02/10/2023	Introduzione. L'architettura di riferimento. Il ciclo di esecuzione di un'istruzione. Storia dell'Elaboratore (Prof. Borghese, ultima modifica 04/10/22).
09/10/2023	Codifica dell'informazione. Operazioni su numeri binari. Le operazioni fondamentali: somma e sottrazione. Rappresentazione binaria dei numeri decimali. <b>Autocod</b> (Prof. Borghese, ultima modifica 08/10/22).
16/10/2023	Interviste sulla codifica binaria e sulle operazioni fondamentali. Codifica IEEE754 e codifica Brain Float di Google dei numeri in virgola mobile (Prof. Borghese, ultima modifica 10/10/22).
<b>Logica combinatoria</b>	
23/10/2023	Laboratorio. Codifica dell'informazione numerica: variazione posizionale, cambio di base, somma e sottrazione, complemento a 2, overflow (2 ore).
30/10/2023	Algebra combinatoria: variabili ed operatori. Implementazione circuitale (porte logiche). Dal circuito alla funzione. Algebra Booleana. Le porte universali (Prof. Borghese, ultima modifica 17/10/22).
06/11/2023	Dalla tabella della verità al circuito: la prima forma canonica. Criteri di ottimizzazione. Semplificazione Algebrica. (Prof. Borghese, ultima modifica 28/10/22).
13/11/2023	Laboratorio. Codifica dell'informazione numerica: rappresentazione dei numeri reali, numeri subnormalizzati (2 ore).
20/11/2023	Implementazione circuitale di funzioni logiche mediante PLA e ROM. Circuiti combinatori esternali. (Prof. Borghese, ultima modifica 24/10/22).
<b>Le unità aritmetico-logiche</b>	
27/11/2023	Alfabetazione. Anticipazione del report (Prof. Borghese, ultima modifica 08/11/22).
04/12/2023	Laboratorio. Introduzione a Logisim: presentazioni della piattaforma e realizzazione di semplici circuiti combinatori (manipolazioni algebriche) (2 ore).
11/12/2023	Moltiplicatori hardware. Progettazione di una ALU a due stadi. (Prof. Borghese, ultima modifica 01/11/22).
18/12/2023	Comparatore a Overflow. Implementazione dei vari tipi binari. Circuiti sequenziali: Latch SR, SR Flip-Flop. (Prof. Borghese, ultima modifica 08/11/22).
25/12/2023	Laboratorio - SOP POS (seconda forma canonica), minimo critico, mappe di Karnaugh (1 ore).
<b>La logica sequenziale</b>	
01/01/2024	Latch SR e D e Flip-flop. (Prof. Borghese, ultima modifica 10/11/22).
08/01/2024	Macchine a stati finiti. Dalle specifiche al progetto. State Transition Graph, State Transition Table, Codifica della STT. Sistemi del circuito. (Prof. Borghese, ultima modifica 14/11/22).
15/01/2024	Laboratorio - Circuiti combinatori: decodici, multiplexers, sommatore Half Adder e Full Adder (2 ore).
22/01/2024	Presentazione sulle macchine a stati finiti e di esempio sulla prima parte del corso (Prof. Borghese, ultima modifica 11/11/22).
29/01/2024	Lezione sospesa per prova in itinere di matematica.
05/02/2024	Laboratorio - Circuiti combinatori: addizionatore a 4 bit, circuito somma e differenza, overflow e overflow (2 ore).
12/02/2024	Laboratorio - Circuiti combinatori: Moltiplicazione, ALU (3 ore).
19/02/2024	Prima prova in itinere: fino alla lezione 10 (Macchine a Stati Finiti). Ore XXXX in aula XXXX. Per partecipare al compito occorre iscriversi sul SISA. Testo versione 1. Testo versione 2. Risultati.
<b>Il firmware</b>	
26/02/2024	Introduzione al firmware. Circuiti firmware della moltiplicazione istata (Prof. Borghese, ultima modifica 28/11/22).

- 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. 2023-2024

9/70

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



# Materiale didattico



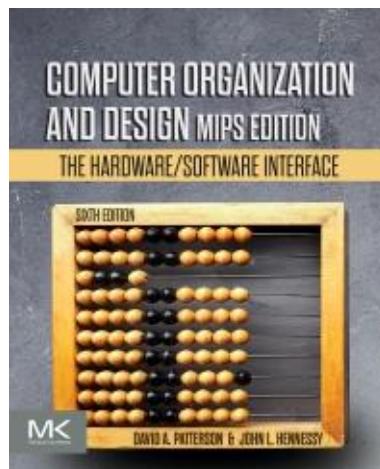
[http://borghese.di.unimi.it/Teaching/Architettura\\_I/References.rtf](http://borghese.di.unimi.it/Teaching/Architettura_I/References.rtf)

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 materiale (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. Terza edizione, 2023.



A.A. 2023-2024

11/70

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



# Non solo teoria

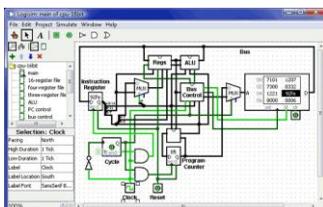


«learn by doing» is equally important -> laboratorio

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
- Riflettere sulla relazione tra i concetti
  
- Sperimentare l'applicazione dei concetti
  
- Collegare un argomento con gli altri argomenti del corso e di altri corsi nonché con le conoscenze pregresse.
  
- lo studio è una **funzione attiva**, che richiede energia e impegno: occorre **pensare**.



## Studiare non è



- Imparare a memoria il contenuto delle slide
- Imparare a memoria il contenuto del libro
- Leggere e ripetere
  
- L'energia e la fatica messa nello studio si traducono alla fine in tempo risparmiato e in una maggiore soddisfazione personale: **si impara.**



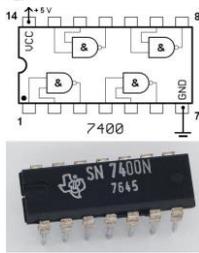
## Sommario della lezione



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

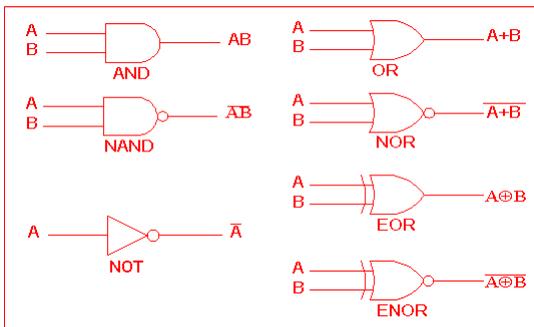


# Contenuto del corso



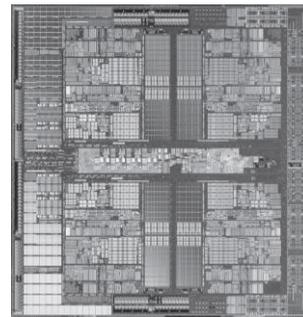
From logic gates to

.... multi-core and GPUs



A.A. 2023-2024

17/70



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



# Le architetture



## La casa



A.A. 2023-2024

18/70

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



# Le architetture

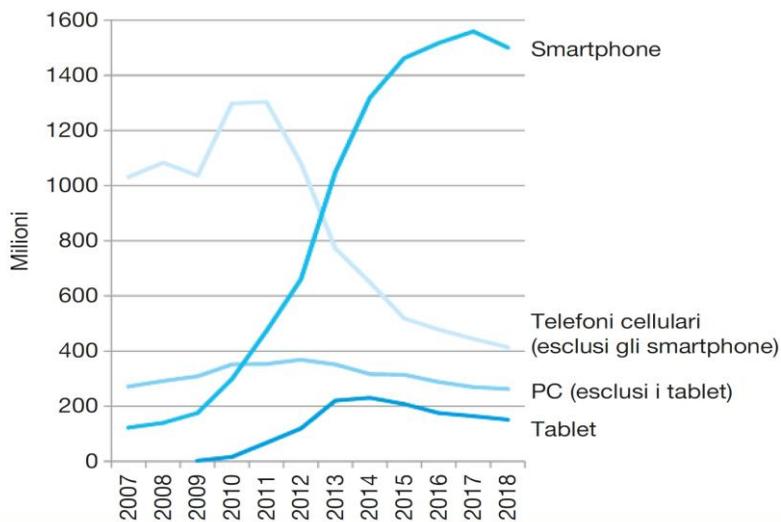


A.A. 2023-2024

19/70



# I calcolatori nel mondo: verso l'era PostPC



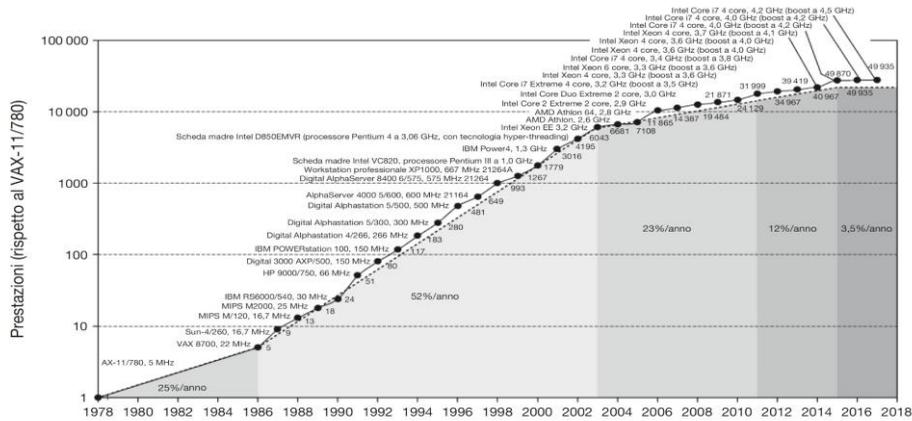
A.A. 2023-2024

20/70

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



# La legge di Moore



**Figura 1.17** Crescita nelle prestazioni dei processori a partire dalla metà degli anni Ottanta. Questo grafico riporta le prestazioni relative al VAX11/780, misurate attraverso i benchmark SPECint (vedi il paragrafo 1.10). Prima della metà degli anni Ottanta, l'aumento delle prestazioni dei processori era dovuto principalmente alla tecnologia, ed era dell'ordine del 25% all'anno. L'aumento delle prestazioni nel periodo seguente è stato di circa il 52% all'anno, grazie a nuove idee nella progettazione delle architetture e nell'organizzazione dei calcolatori. Questo ha portato a un aumento delle prestazioni che nel 2002 è stato di sette volte l'aumento che si sarebbe verificato con un aumento di prestazioni del 25% all'anno. A partire dall'anno 2002 il limite sulla potenza assorbita, il parallelismo implicito delle istruzioni e la latenza della memoria hanno rallentato l'aumento delle prestazioni delle architetture monoprocesore a un 3,5% all'anno. (Da J.L. Hennessy, D.A. Patterson. *Computer Architecture: A Quantitative Approach*. Ed. 6. Waltham, MA: Elsevier, 2017).

In circa 18 mesi raddoppiano le prestazioni ed il numero di transistor e raddoppiano le capacità delle memorie (DRAM). **Legge di Moore**. Non vale più!! => Domain Specific Architectures  
La **velocità** di accesso alla memoria cresce molto più lentamente.



# 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:

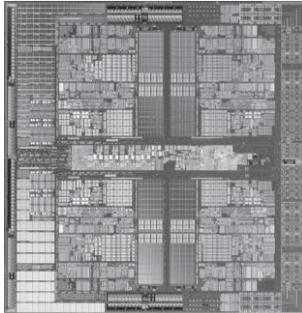
- Aumento della dimensione dei chip (più transistor per chip)
- Diminuzione delle dimensioni (chip più piccoli, aumento del numero di chip, integrazione di chip)
- «Device cleverness» (multi-core)



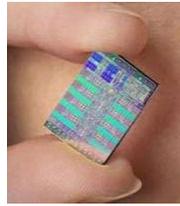
Da IEEE Spectrum, April 2015



## Architetture recenti



AMD Barcelona  
(quad-core)



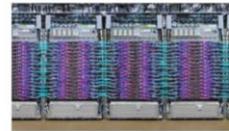
Cell processor  
(IBM, Sony, Toshiba)  
Playstation 3



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

A.A. 2023-2024

23/70



## 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.

A.A. 2023-2024

24/70

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



## 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



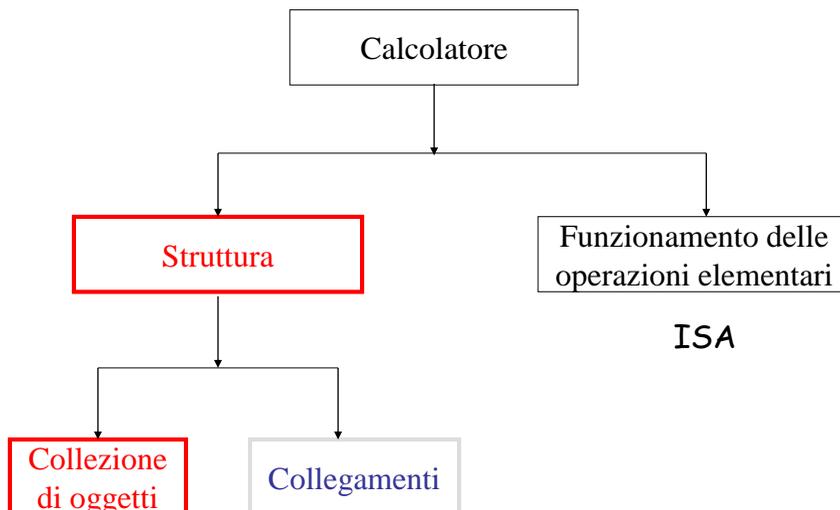
Turing: “Universal Turing machine” (1936). Macchina di esecuzione di algoritmi universale.

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.

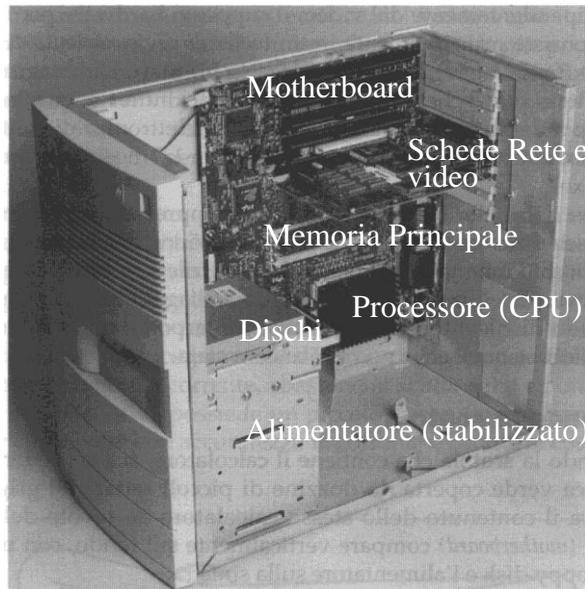


# Descrizione di un elaboratore





# Struttura dell'elaboratore



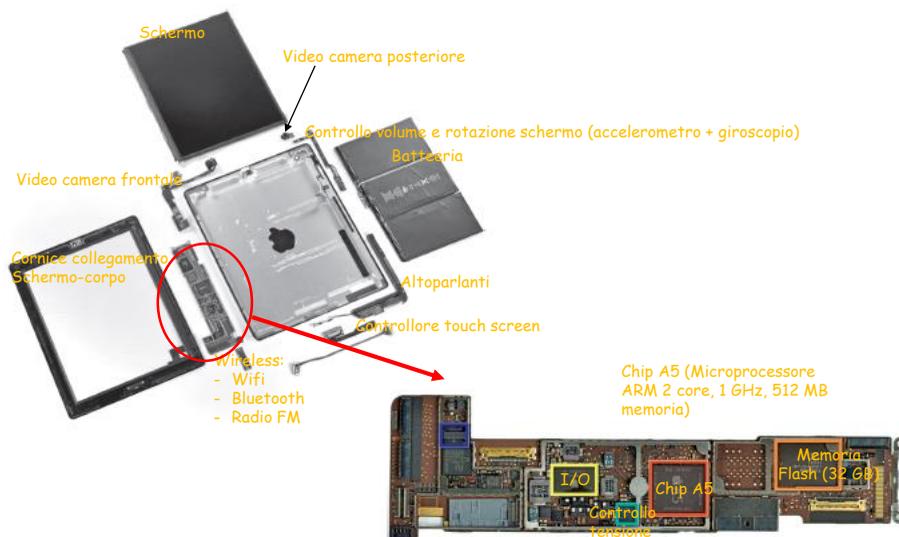
A.A. 2023-2024

29/70

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



# Struttura di un PMD - I-Pad2



A.A. 2023-2024

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, 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.



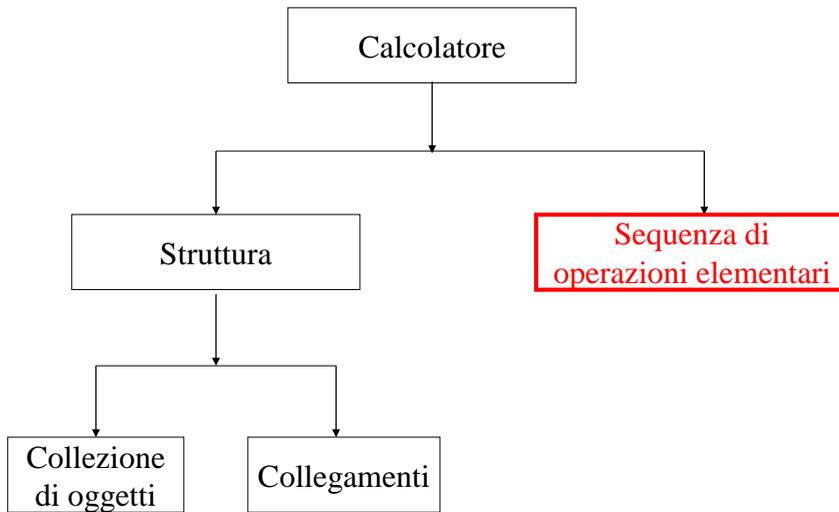
## Sommario della lezione



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



## Descrizione di un elaboratore



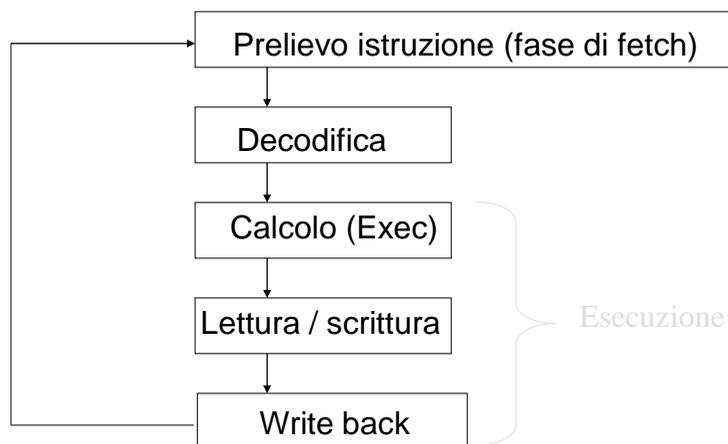
A.A. 2023-2024

35/70

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



## Ciclo di esecuzione di un'istruzione MIPS



A.A. 2023-2024

36/70

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





## 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



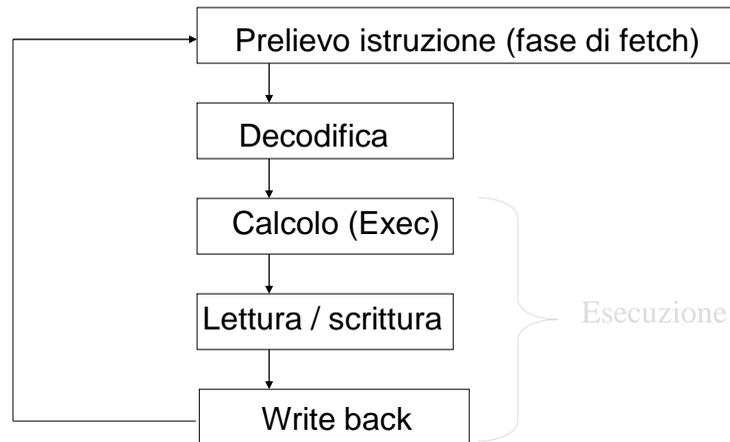
Viene selezionato il circuito / i circuiti appropriati per l'esecuzione delle operazioni previste dall'istruzione e determinate in fase di decodifica.

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





## Ciclo di esecuzione di un'istruzione MIPS



A.A. 2023-2024

43/70

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



## Sommario della lezione



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

A.A. 2023-2024

44/70

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



## Storia dell'elaboratore



### Filo conduttore:

Aumento della velocità di elaborazione

Diminuzione della dimensione dei componenti.

Aumento della capacità e velocità dell'I/O.

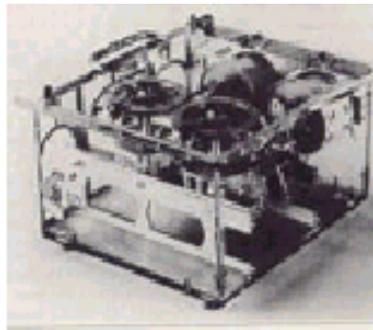
Adozione di tecnologie diverse (meccanica, elettrica, elettronica).



## 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. 2023-2024

47/70

<http://borgnese.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. 2023-2024

48/70

<http://borgnese.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).



A.A. 2023-2024

49/70

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



## 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).

A.A. 2023-2024

50/70

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

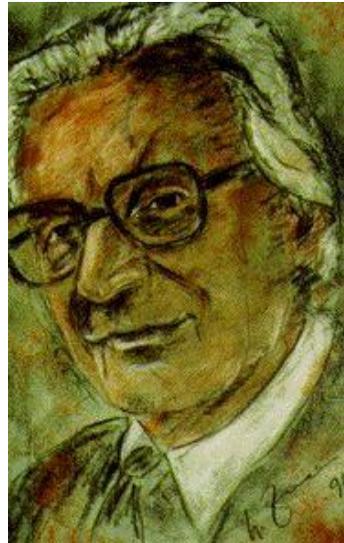


## Il papà non riconosciuto



Konrad Zuse, 1936  
Ingegnere civile.

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



Auto-ritratto del 1944

A.A. 2023-2024

51/70

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



## Storia dell'elaboratore - Mark I - 1944

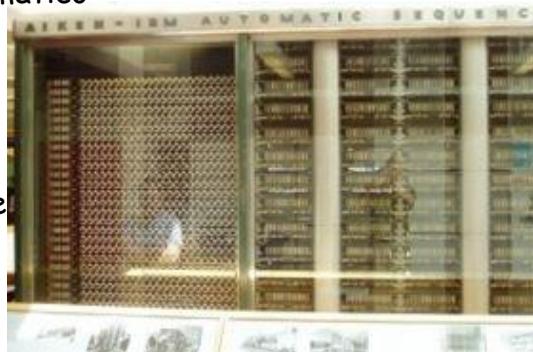


Primo computer automatico

Elettromeccanico

15.3s per divisione  
6s per moltiplicazione

OpCode + operandi



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

A.A. 2023-2024

52/70

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



## 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. 2023-2024

53/70

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



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

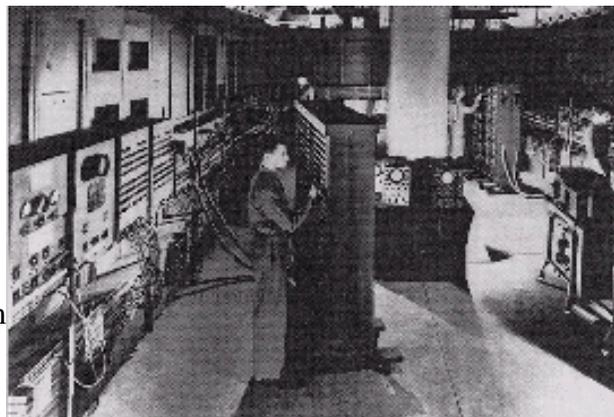


Elettronica (valvole: diodo, triodo). Aumento di prestazioni di 1,000 volte.

- 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. 2023-2024

55/70

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



## Eckbert & Mauchly

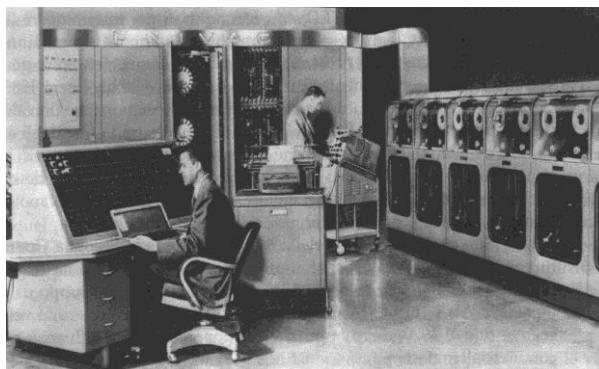


- **EDVAC**, Eckbert, Mauchly, **Von Neuman**. Moore school, Pennsylvania University. **Programma memorizzato**.

- **EDSAC**, Eckert, Cambridge, 1949, ( $\Rightarrow$  Mark I, 1948).

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

48 esemplari a 1M\$



A.A. 2023-2024

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. 2023-2024

57/70

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



# La comunicazione tra i componenti



Switch centralizzato

(multiplexor -> bridge)

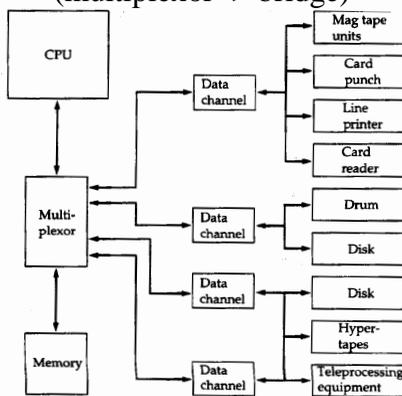
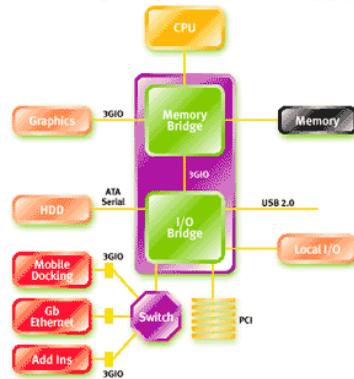


Figure 2.5 An IBM 7094 Configuration

Programma di "canale"



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

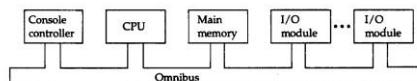


Figure 2.9 PDP-8 Bus Structure

A.A. 2023-2024

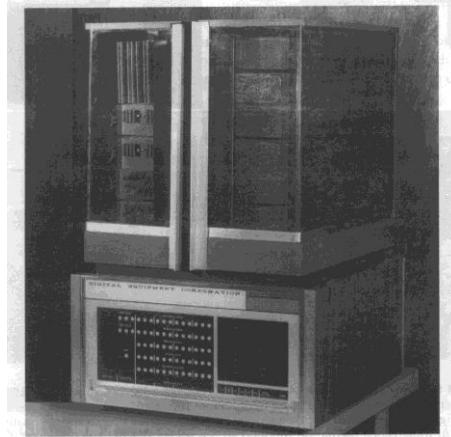
58/70



## 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. 2023-2024

59/70

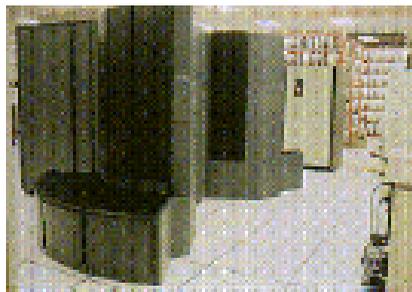
<http://borghese.di.unimi.it/>



## La quarta generazione (1971-1977)



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



A.A. 2023-2024

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. 2023-2024

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

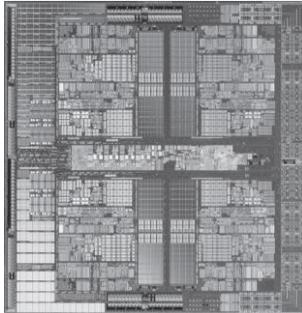
A.A. 2023-2024

62/70

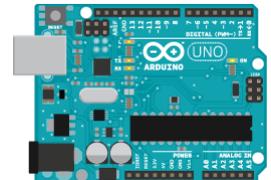
<http://borghese.di.unimi.it/>



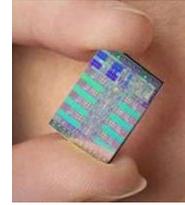
## La sesta generazione (postPC)



AMD Barcelona  
(quad-core)



Arduino microcontroller  
(M. Banzi)



Cell processor  
(IBM, Sony, Toshiba)  
Playstation 3

nVidia 9800 GTX, Streaming  
processors 128 core



A.A. 2023-2024

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. 2023-2024

64/70

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



# Il futuro



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

Calcolatori ottici.  
Calcolatori chimici.



# Il futuro

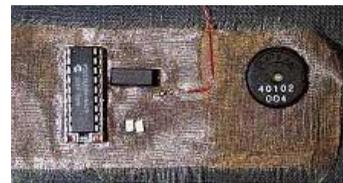


**Cloud TPU v3**  
420 teraflops  
128 GB HBM

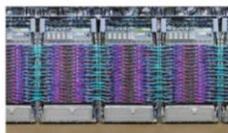
- **Pervasive computing**
- **Dedicated architectures**



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'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

A.A. 2023-2024

69/70

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



## Sommario della lezione



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

A.A. 2023-2024

70/70

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