



# 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

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).**



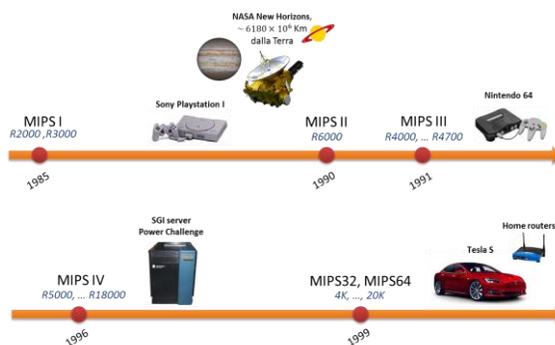
## Architettura base del corso: MIPS R3000



### ► Features-front



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



Samsung S5  
Snapdragon 801 Processor  
ARM Kreit 400 CPU



# MIPS



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

"Hello world" in x86 (64 bit)

```
.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

### Architettura II

- CPU avanzate
- Gestione delle gerarchie di memoria e memoria virtuale
- Parallelizzazione dell'esecuzione
- Gestione dell'I/O



# Architettura I - 6 CFU (Cognomi A-G) (Cognomi H-Z Prof. Marco Tarini)



## 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\\_2019-2020.html](http://borgnese.di.unimi.it/Teaching/Architettura_I/Programma_2019-2020.html)

Let's try to keep the course interactive

## Orario turno I Prof. Borghese:

Martedì Ore 10.30-12.30 Aula G24

Giovedì Ore 10.30-12.30 Aula G12

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

Ricevimento su appuntamento

A.A. 2019-2020

7/68

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



# Il sito del corso



Architettura degli elaboratori - I - Prof. [Alberto Borghese](#) (cognomi A-G) - Prof. [Marco Tarini](#) (cognomi H-Z)  
 Laboratorio: [Dr. Nicola Basilico](#) (cognomi A-C), [Dott.ssa Gabriella Tracco](#) (cognomi D-N), [Dott. Matteo Re](#) (cognomi O-Z)  
 Corso di laurea triennale in Informatica, Università di Milano, A.A. 2019-2020, Primo Semestre.

Avvisi:

**Orario del corso:**  
 Martedì, lezione, ore 10.30-12.30, aula G24.  
 Giovedì, lezione, ore 10.30-12.30, aula G14.

Nota: lo scritto verrà sugli argomenti presentati nell'anno in cui il corso è stato seguito.

[Programma dettagliato A.A. 2019-2020.](#)

[Materiale bibliografico del corso. Esercizi](#)

[Programmi e bibliografie materiali in corso](#)

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

**NB: È il diritto a scaricare il materiale accessibile da questa pagina e 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 Informatics University course.

**Modalità d'esame: scritto + orale + progetto (le vacanze pasquali hanno validità di 18 mesi).** Per dettagli sulla modalità di presentazione dei progetti d'esame, consultare [Modalità per la presentazione dei progetti](#) nella home page del Dott. Basilico.

**Simulatore SW per la realizzazione di circuiti digitali: Gatesim.** <http://www.kubit.net/gatesim/> (ultima revisione 25 Novembre 2012).

**Temi d'esame:**

23 Gennaio 2020	23 Gennaio 2020			
26 Gennaio 2019	21 Febbraio 2018	20 Giugno 2019	23 Luglio 2019	19 Settembre 2019
23 Gennaio 2018	21 Febbraio 2018	21 Giugno 2018	26 Luglio 2018	27 Settembre 2018
21 Gennaio 2017	21 Febbraio 2017	15 Giugno 2017	23 Luglio 2017	28 Settembre 2017
29 Gennaio 2016	21 Febbraio 2016	11 Giugno 2016	21 Luglio 2016	28 Settembre 2016
29 Gennaio 2015	21 Febbraio 2015	24 Giugno 2015	21 Luglio 2015	25 Settembre 2015
21 Gennaio 2014	21 Febbraio 2014	26 Giugno 2014	24 Luglio 2014	20 Settembre 2014
01 Febbraio 2013	21 Febbraio 2013	25 Giugno 2013	22 Luglio 2013	26 Settembre 2013
21 Gennaio 2012	25.02.2012	22.06.2012	23.07.2012	21.09.2012
21 Gennaio 2011	24 Febbraio 2011	24 Giugno 2011	26 Luglio 2011	29 Settembre 2011

Per i corsi degli Anni Accademici Precedenti, [click here](#)

A.A. 2019-2020

8/68

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



# Alcune avvertenze



**Programma del corso di Architettura degli Elaboratori - parte I  
A.A. 2018-2019**

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

Le lezioni di esercitazione sono riportate in colore rosso, le lezioni di laboratorio in Via e le lezioni frontali in nero.  
Le slide sono da considerare bozze e vanno usate fino al giorno della lezione. Le slide in versione definitiva, saranno disponibili sul sito il giorno dopo la lezione.

Introduzione	
L_01	01/10/2019 Introduzione. L'architettura di riferimento. Il ciclo di esecuzione di un'istruzione. Storia dell'Elaboratore (Prof. Borghese, ultima modifica 10/10/18).
L_02	03/10/2018 Codifica dell'informazione. Operazioni su numeri binari. Le operazioni fondamentali: somma e sottrazione. Rappresentazione binaria dei numeri decimali. Lo standard IEEE754 (Prof. Borghese, ultima modifica 10/10/18).
Implementazione del Dipartimento	
Logica combinatoria	
L_03	04/10/2019 L'algebra combinatoria: variabili ed operatori. Implementazione circuitale (porte logiche). Dal circuito alla funzione: Algebra Booleana. Le porte universali (Prof. Borghese, ultima modifica 10/10/18).
L_04	10/10/2019 Dalla tabella della verità al circuito: la prima forma canonica. Critici di ominalità. Semplificazione algebrica. (Prof. Borghese, ultima modifica 16/10/18).
L_05	11/10/2019 Laboratorio
L_06	13/10/2019 Massone all'interno del docente
L_07	17/10/2019 Massone all'interno del docente
L_08	18/10/2019 Laboratorio
L_09	22/10/2019 Implementazione circuitale di funzioni logiche mediante PLA e ROM. Circuiti combinatori multipli. (Prof. Borghese, ultima modifica 16/10/18).
L_10	24/10/2019 Lezione sospesa per partecipazione a conferenza del docente
L_11	25/10/2019 Laboratorio
Le unit aritmetico-logiche	
L_12	29/10/2019 Massone all'interno del docente
L_13	31/10/2019 Aggiornamento. Anticipazione del progetto (Prof. Borghese, ultima modifica 23/10/18).
L_14	01/11/2019 Fratello di Tutti i Santi
L_15	05/11/2019 Moltiplicatore hardware. Progettazione di una ALU: 1 due stati. (Prof. Borghese, ultima modifica 20/10/18).
La logica sequenziale	
L_16	07/11/2019 Comparazione e OverFlow. Temporizzazione dei circuiti booleani. Circuiti sequenziali: 1 latch SC. (Prof. Borghese, ultima modifica 20/10/18).
L_17	08/11/2019 Laboratorio
L_18	12/11/2019 Stati nascosti e flip-flop. Prof. Borghese, ultima modifica 20/10/17).
L_19	14/11/2019 Macchine a stati finiti. Dalle specifiche al progetto: State Transition Graph. State Transition Table. Codifica della STT. Stato del circuito. (Prof. Borghese, ultima modifica 13/11/18).
L_20	15/11/2019 Laboratorio
L_21	09/11/2019 Massone del docente

- 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. 2019-2020 9/68 http://borghese.di.unimi.it/



# Materiale didattico



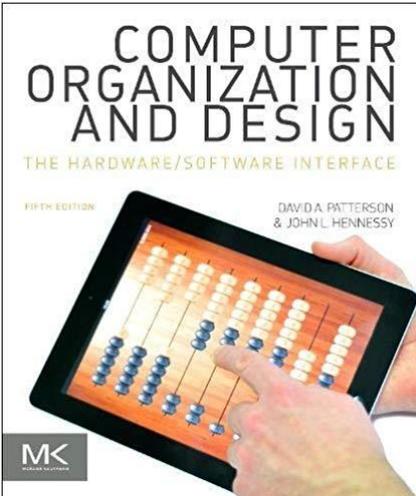
[http://borghese.di.unimi.it/Teaching/Architettura\\_I/References.rtf](http://borghese.di.unimi.it/Teaching/Architettura_I/References.rtf)

**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, Quarta edizione, Zanichelli, estate 2015. **Edizione MIPS** (Nota: la quarta edizione Zanichelli è la traduzione della quinta edizione inglese).

"Computer Organization & Design: The Hardware/Software Interface", D.A. Patterson and J.L. Hennessy, Morgan Kaufmann Publishers, Fifth Edition, 2013 (**MIPS edition**).

**Il testo copre il contenuto dei corsi di Architettura I e II. Parte del materiale si trova su WEB sul sito dell'editore.**



A.A. 2019-2020 10/68 http://borghese.di.unimi.it/



## Materiale didattico



[http://borgnese.di.unimi.it/Teaching/Architettura\\_I/References.rtf](http://borgnese.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. 2019-2020

11/68

imi.it/



## Non solo teoria



«learn by doing» is equally important -> laboratorio

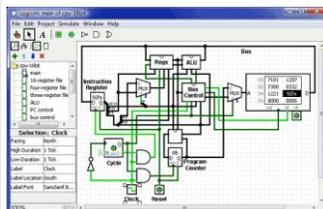
Cognomi A-C, Dott. Nicola Basilico  
[Nicola.basilico@unimi.it](mailto:Nicola.basilico@unimi.it)



Cognomi D-N, Dott.ssa Gabriella Trucco  
[Gabriella.trucco@unimi.it](mailto:Gabriella.trucco@unimi.it)



Cognomi O-Z, Dott. Matteo Re  
[Matteo.re@unimi.it](mailto:Matteo.re@unimi.it)



<http://ozark.hendrix.edu/~burch/logisim/>

12/68

<http://borgnese.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)

Alcuni progetti notevoli: Pong, Snake, Master mind, Space Invaders, Sudoku...



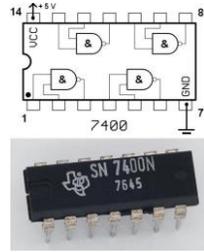
## Sommario della lezione



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



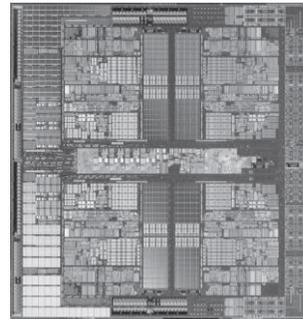
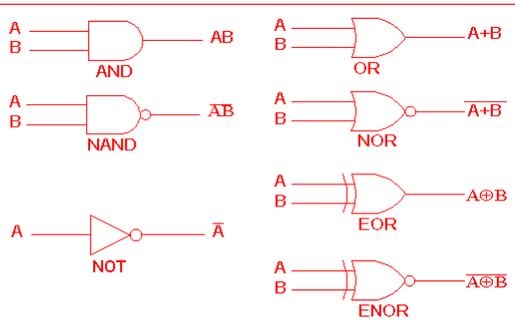
# Contenuto del corso



From logic gates to multi-core and GPUs



- Sala stampa
- Newsletter
- Recensioni
- Case Stud
- Flussi RSS



A.A. 2019-2020

15/68

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



# Le architetture



La casa



A.A. 2019-2020

16/68

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



## Le architetture

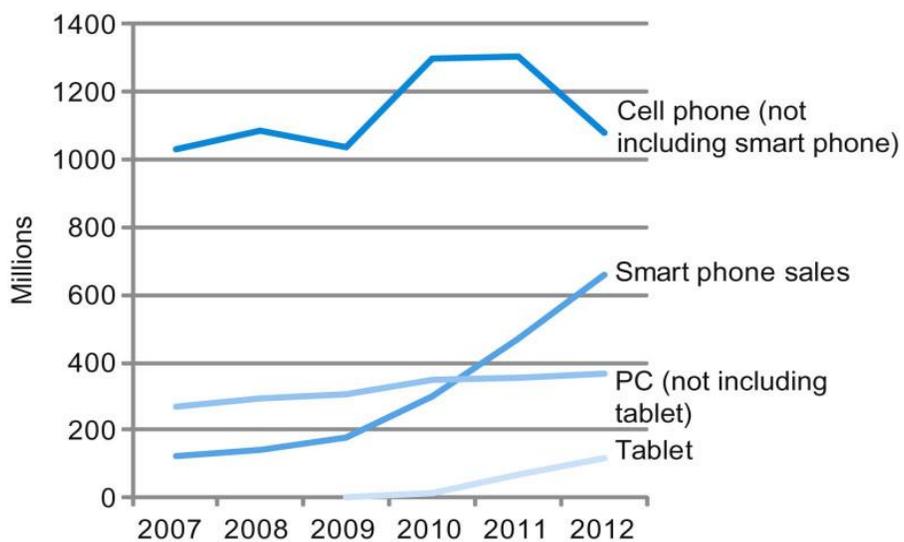


A.A. 2019-2020

17/68



## I calcolatori nel mondo: verso l'era PostPC



A.A. 2019-2020

18/68

<http://borghese.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.

A.A. 2019-2020

19/68

<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.

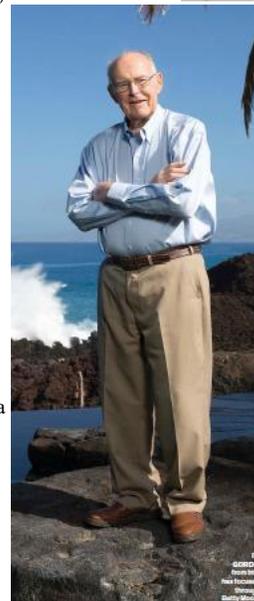
Nel 2014 sono stati prodotti  $250 \times 10^{18}$  transistor (250 miliardi di miliardi, 25 volte il numero di stelle della via lattea e 75 volte il numero di galassie dell'Universo conosciuto). Ogni secondo vengono prodotti 8 miliardi di transistor. Più transistor nel 2014 che fino al 2011.

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:

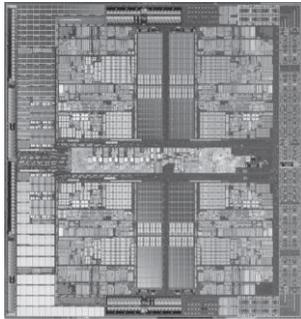
- Aumento della dimensione dei chip (più transistor per chip)
- Diminuzione delle dimensioni (chip più piccoli, aumento del numero di chip, integrazione di chip)
- «Device cleverness» (multi-core)



Da IEEE Spectrum, April 2015

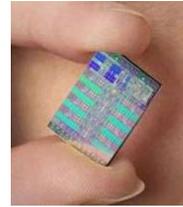


## Le architetture più recenti



AMD Barcelona  
(quad-core)

nVidia 9800 GTX,  
Streaming processors  
128 core



Cell processor  
(IBM, Sony, Toshiba)  
Playstation 3



## 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.



## 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....)

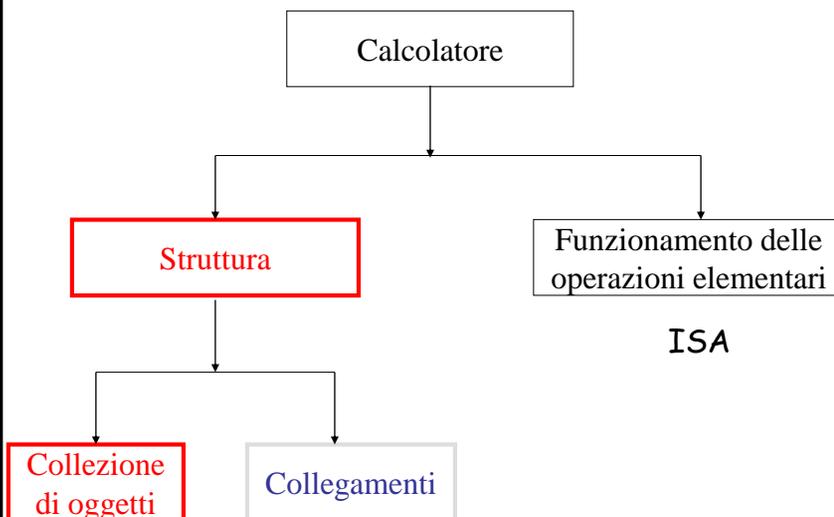
Controllo del flusso (if, for....)

L'informazione viene rappresentata utilizzando solamente due simboli (base 2: 0,1 -> acceso, spento).

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

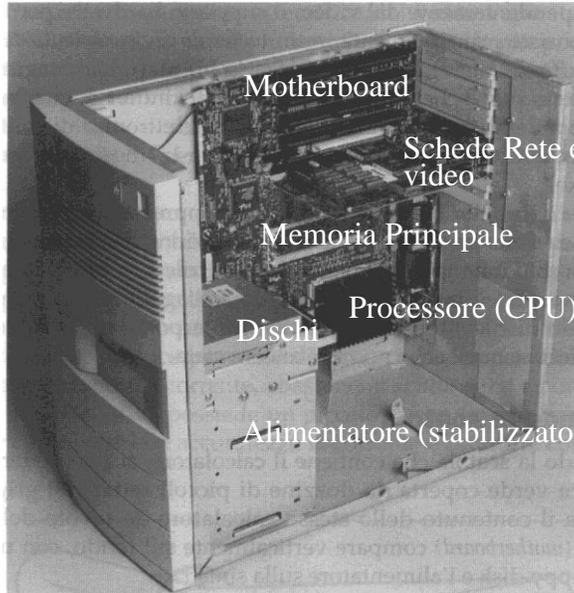


## Descrizione di un elaboratore





# Struttura dell'elaboratore



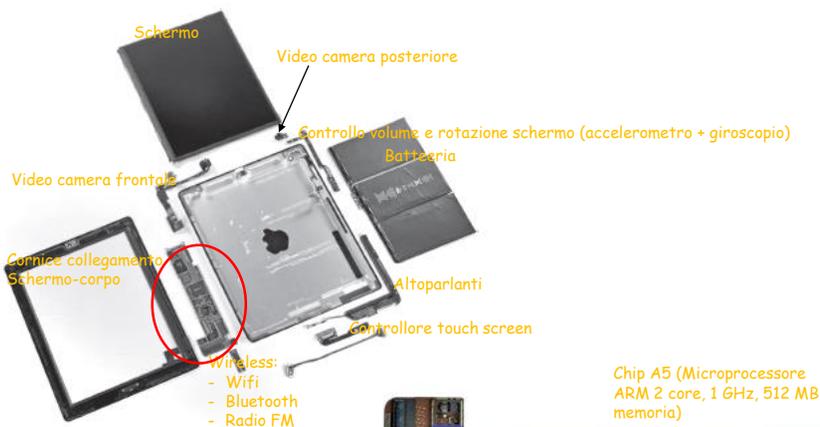
A.A. 2019-2020

27/68

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

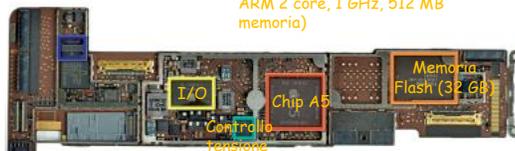


# Struttura di un PMD - I-Pad2



- Wireless:
- Wifi
  - Bluetooth
  - Radio FM

Chip A5 (Microprocessore ARM 2 core, 1 GHz, 512 MB memoria)



A.A. 2019-2020

28/68

<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.



## 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.



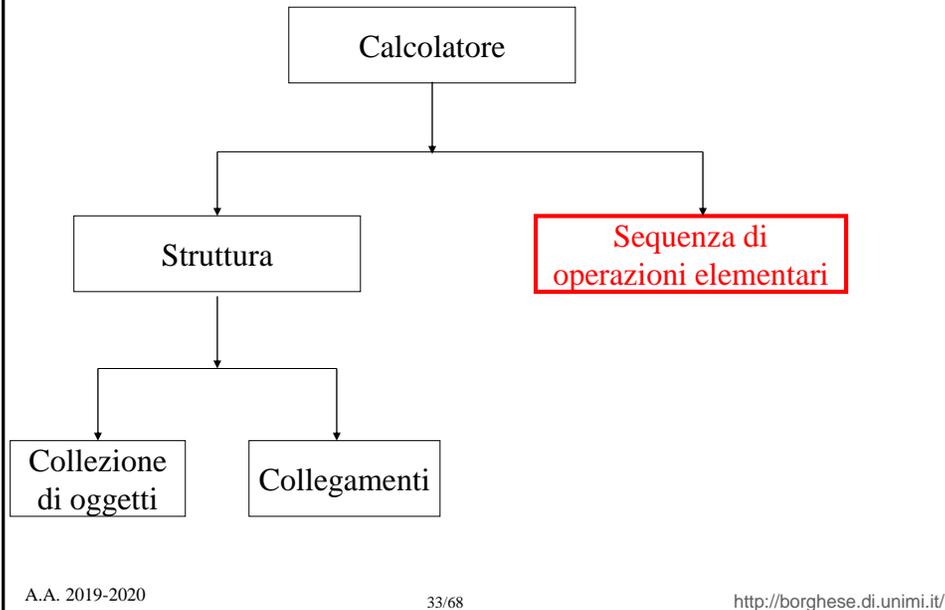
## Sommario della lezione



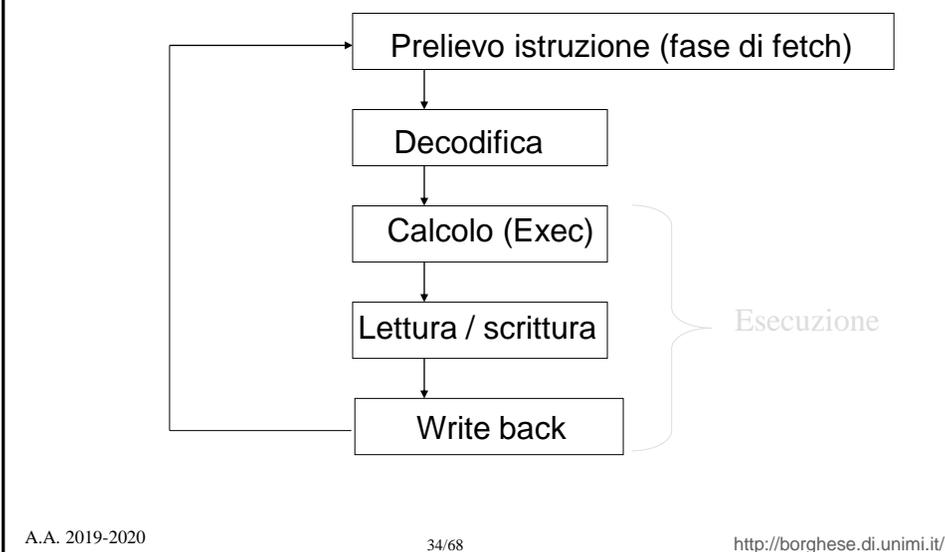
- Informazioni su corso ed esame
- Architettura dell'elaboratore
- **Ciclo di esecuzione di un'istruzione**
- Storia dell'elaboratore.



## Descrizione di un elaboratore



## Ciclo di esecuzione di un'istruzione MIPS





## Letture dell'istruzione (fetch)



- Istruzioni e dati risiedono nella memoria principale, dove sono stati caricati attraverso un'unità di ingresso.
- L'esecuzione di un programma inizia quando il registro PC punta alla (contiene l'indirizzo della) prima istruzione del programma in memoria.
- Il segnale di controllo per la lettura (READ) viene inviato alla memoria.
- Trascorso il tempo necessario all'accesso in memoria, la parola indirizzata (in questo caso la prima istruzione del programma) viene letta dalla memoria e trasferita nel registro IR.
- Il contenuto del PC viene incrementato in modo da puntare all'istruzione successiva.

A.A. 2019-2020

35/68

<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.
  - ◆ Architetture a registri:
    - Se un operando risiede in memoria, deve essere prelevato caricando l'indirizzo dell'operando nel registro MAR della memoria e attivando un ciclo di READ della memoria.
    - L'operando letto dalla memoria viene posto nel registro della memoria MDR per essere trasferito alla ALU, che esegue l'operazione. Nelle architetture MIPS, l'operando viene trasferito nel Register file nella fase di Scrittura.
  - ◆ Architetture LOAD/STORE:
    - Le istruzioni di caricamento dalla memoria sono separate da quelle aritmetico/logiche.

A.A. 2019-2020

36/68

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



## Calcolo dell'istruzione (execution)



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

Tra le operazioni previste, c'è anche la formazione dell'indirizzo di memoria da cui leggere o su cui scrivere un dato.



## Letture / Scrittura in memoria



In questa fase il dato presente in un registro, viene scritto in memoria oppure viene letto dalla memoria un dato e trasferito ad un registro.

Questa fase non è richiesta da tutte le istruzioni!

Nel caso particolare di Architetture LOAD/STORE, quali MIPS, le istruzioni di caricamento dalla memoria sono separate da quelle aritmetico/logiche. Se effettua una Lettura / Scrittura, **non** esegue operazioni aritmetico logiche sui dati.

Sistema di memoria "sganciato" dalla coppia register-file + CPU.



## 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 ciclo di esecuzione



Somma:           0x80000:  add \$s3, \$s2, \$s1  
  00000010010100011001100000100000

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 (che sono contenuti nei registri \$s2, \$s1).

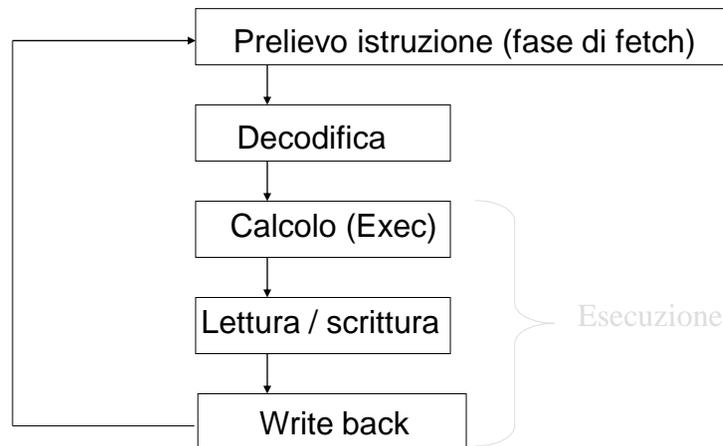
Esecuzione:      Esecuzione della somma.

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.

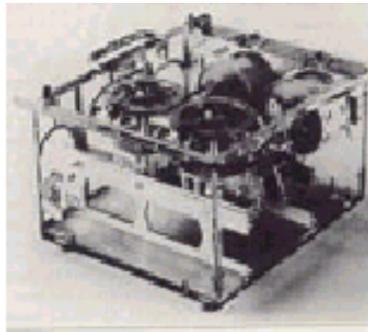
Adozione di tecnologie diverse (meccanica, elettrica, elettronica).



## 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. 2019-2020

45/68

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



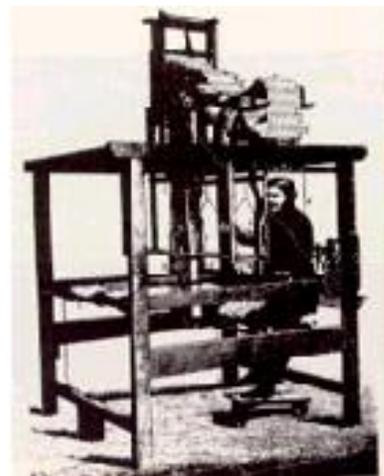
## Un'architettura efficace



*Una macchina per risolvere un problema industriale.*

### ***Telaio Jacquard (1801)***

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



A.A. 2019-2020

46/68

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



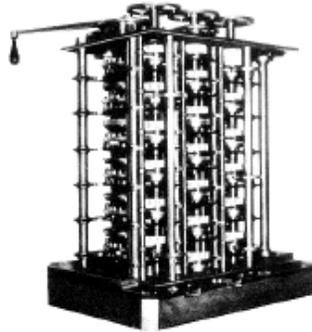
## Charles Babbage



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).

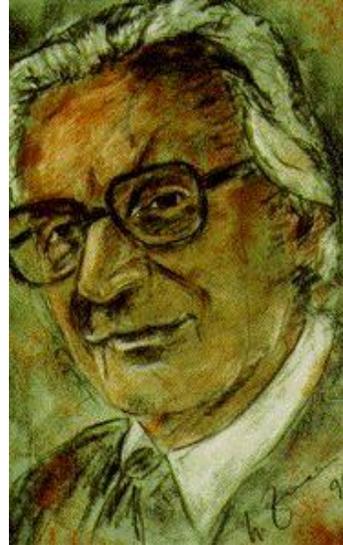


## Il papà non riconosciuto



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

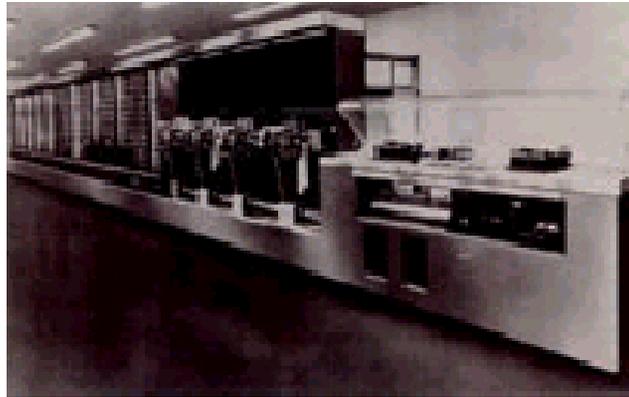
OpCode + operandi



**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. 2019-2020

51/68

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



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

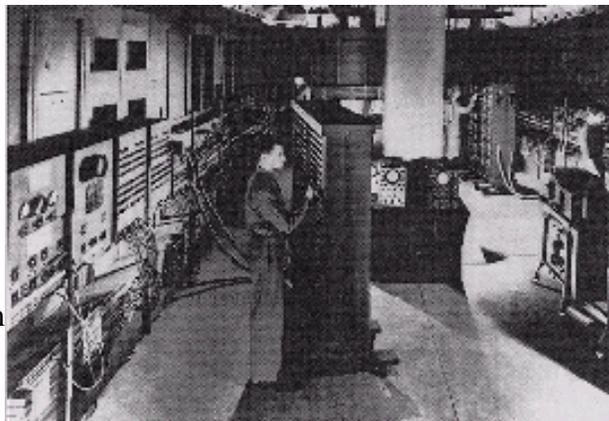


Elettronica (valvole: diodo, triodo). Aumento di prestazioni di 1,000 volte.

- 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	Electronic	Programmable	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. 2019-2020

53/68

<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. 2019-2020

54/68

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



## 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. 2019-2020

55/68

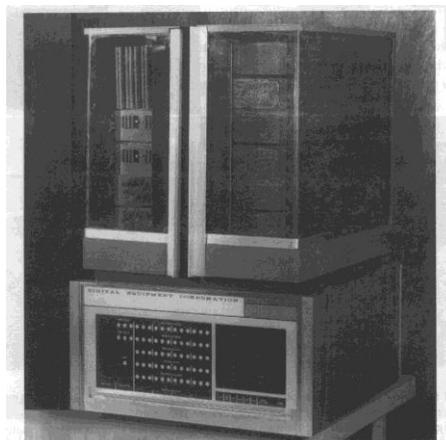
<http://homepages.dsi.unige.it/~borghese>



## 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. 2019-2020

56/68

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



## La comunicazione tra i componenti

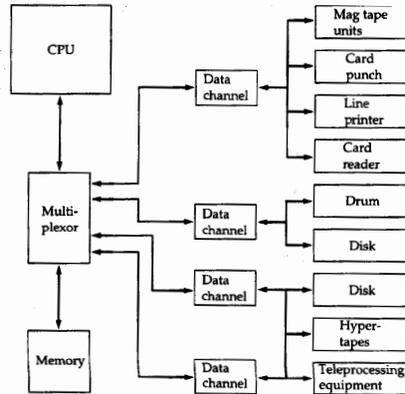


Figure 2.5 An IBM 7094 Configuration

Switch centralizzato (multiplexer) (cf. bridge)

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

Programma di "canale"

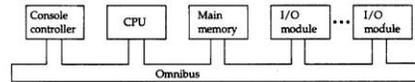


Figure 2.9 PDP-8 Bus Structure

A.A. 2019-2020

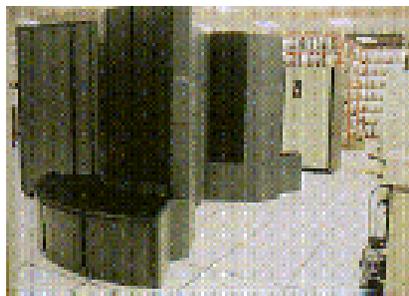
57/61



## La quarta generazione (1971-1977)



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



A.A. 2019-2020

58/68

<http://home.uniroma3.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. 2019-2020

59/68

<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.  
Coproprocessore 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

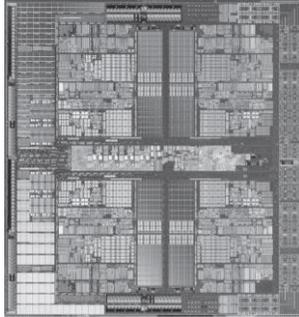
A.A. 2019-2020

60/68

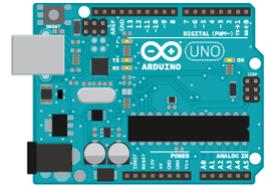
<http://borghese.di.unimi.it/>



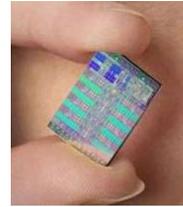
## La sesta generazione (postPC)



AMD Barcelona  
(quad-core)



Arduino microcontroller  
(M. Banzi)



Cell processor  
(IBM, Sony, Toshiba)  
Playstation 3

nVidia 9800 GTX, Streaming  
processors 128 core



A.A. 2019-2020

61/68

<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?**
  - ◆ **Parallellizzazione del codice. (e.g. RapidMind Development Platform).**
  - ◆ **Nuovo modo di ragionare durante la programmazione software.**
  - ◆ **Tool di aiuto.**
  - ◆ **Parallellizzazione automatica del codice è ancora molto lontana.**
  - ◆ **Problema principale è la coerenza dei dati.**

A.A. 2019-2020

62/68

<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.
- Macchine intelligenti e sensibili.
- Sistemi multimediali.

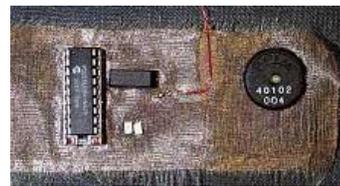


## Il futuro



E-textile

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



Circuito con CPU stampato su stoffa



Computer palmare



Wrist-net  
N3



## 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.