



Architetture degli elaboratori - I Introduzione



Prof. Alberto Borghese
Dipartimento di Informatica
Laboratorio di Sistemi Intelligenti Applicati (AIS-Lab)
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



Motivazione del corso



- Cosa vuol dire scrivere del software?
- Cosa vuol dire scrivere un programma?

Obbiettivi:

- 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)? Che significato ha? Che impatto ha sul software?
- **Sviluppo della capacità di analisi e progettazione (sintesi).**



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).**



Architettura base del corso: MIPS R3000



Features-front



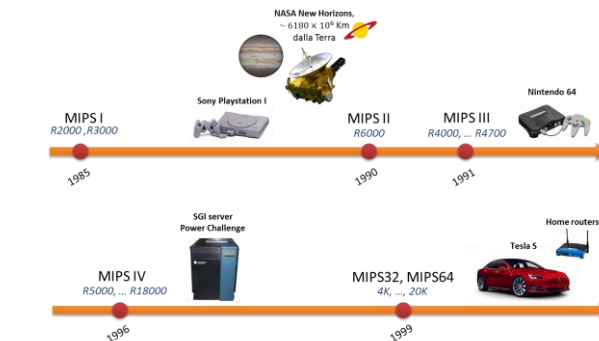
ARMv8, RISC V

Sistemi embedded che montano Windows CE, PlayStation 2, router, gateway...



Samsung S21

Exynos ARM Processor



A.A. 2024-2025

5/69

<http://borgnese.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_wold.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
```

A.A. 2024-2025

6/69

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



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

Programma:

http://borgnese.di.unimi.it/Teaching/Architettura_I/Programma_2024-2025.html

Materiale sensibile su MyARIEL.

Let's try to keep the course interactive



Orario turno I Prof. Borgnese:

Lunedì, lezione, ore 13.30-15.30, aula V2

Giovedì, lezione, ore 8.30-10.30, aula V2

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

Ricevimento su appuntamento

Il sito del corso

Architetture degli elaboratori - I
Turno 1: cognomi A-G -- Prof. [Alberto Borghese](#)
(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. 2024-2025. Primo Semestre.
Avviso: La prova scritta dell'appello del 24 Settembre si terrà in aula G21 con inizio alle ore 14.30. Il corso inizierà Lunedì 30 Settembre 2024..

Orario del corso:
Lunedì, lezione, ore 13.30-15.30, aula V2
Giovedì, lezione, ore 8.30-10.30, aula V2

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. 2024-2025.
[Materiale bibliografico del corso, Esercizi](#)
[Programmi aut bibliografici materiali in pdf](#)

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 hereabove 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:

23 Gennaio 2024	20 Febbraio 2024	18 Giugno 2024	23 Luglio 2024	24 Settembre 2024
23 Gennaio 2023	22 Febbraio 2023	21 Giugno 2023	26 Luglio 2023	20 Settembre 2023

A.A. 2024-2025
9/69
<http://borghese.di.unimi.it/>

Programma

Programma del corso di Architettura degli Elaboratori - parte I
Programma A.A. 2024-2025

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.

Data	Contenuto della lezione
L_01	30/09/2024
L_02	03/10/2024
E_01	07/10/2024
	Logica combinatoria
B_01	08/10/2024
L_03	10/10/2024
L_04	14/10/2024
B_02	15/10/2024
L_05	17/10/2024
	Le unità aritmetico-logiche
L_06	21/10/2024
B_03	22/10/2024
	24/10/2024
L_07	28/10/2024
B_04	29/10/2024

- 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. 2024-2025
10/69
<http://borghese.di.unimi.it/>



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**.



Materiali didattici

http://borghese.di.unimi.it/Teaching/Architettura_I/References.rtf

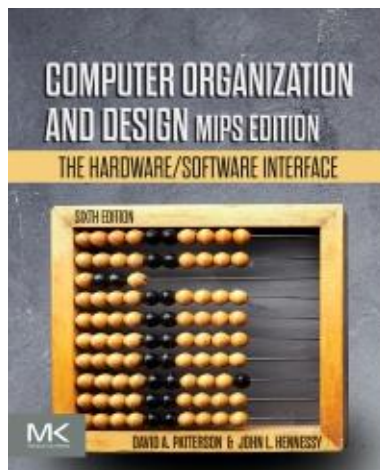


Testo di base (è disponibile sia in inglese che in italiano):

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/>



Materiali didattici

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.





Non solo teoria

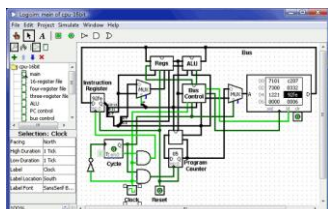


«learn by doing» is equally important -> laboratorio

Dott.ssa Gabriella Trucco (A-FAS)
Gabriella.trucco@unimi.it

Dott. Massimo Rivolta (FAT-MOR)
Massimo.rivolta@unimi.it

Dott. Matteo Re (MOS-Z)
Matteo.Re@unimi.it



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

15/69

<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)

La votazione maturata nella prova di laboratorio e in quella di teoria hanno ciascuna validità 2 sessioni d'esame. Trascorso questo lasso di tempo, la votazione non è più valida.



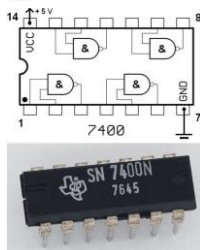
Sommario della lezione



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

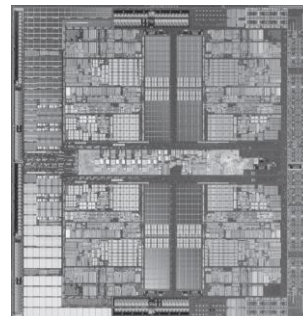
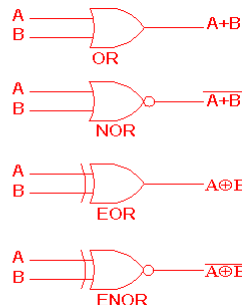
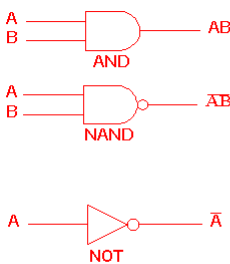


Contenuto del corso



From logic gates to

.....
multi-core and GPUs





Le architetture



La casa



A.A. 2024-2025

19/69

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



Le architetture



PC

A.A. 2024-2025

20/69



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 250×10^{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 impaccati 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



Obbiettivo 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)

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 il controllo sono eseguiti utilizzando **esclusivamente!** le 3 operazioni fondamentali della logica classica: AND, OR, NOT.



I principi delle Architetture



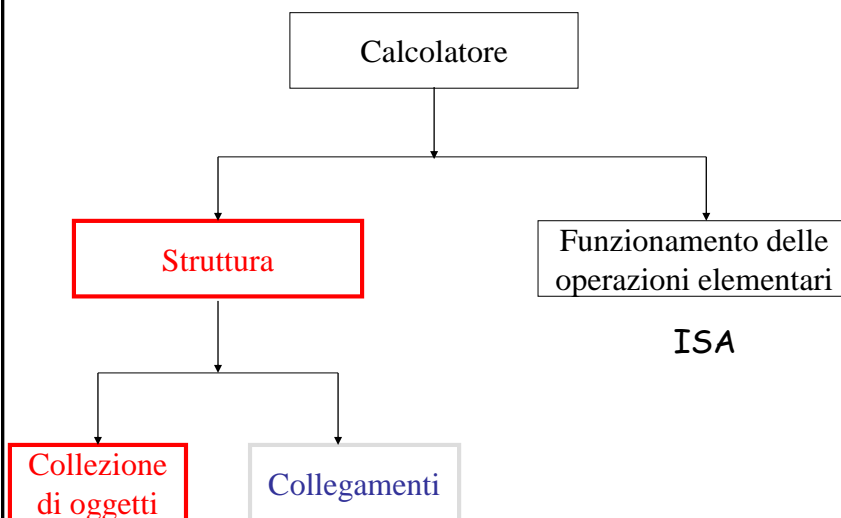
Turing: “Universal Turing machine” (1936). Macchina di esecuzione di algoritmi universale.

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.
- 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.
- Un computer deve essere collegato all'esterno. Occorre quindi un equipaggiamento per l'I/O.

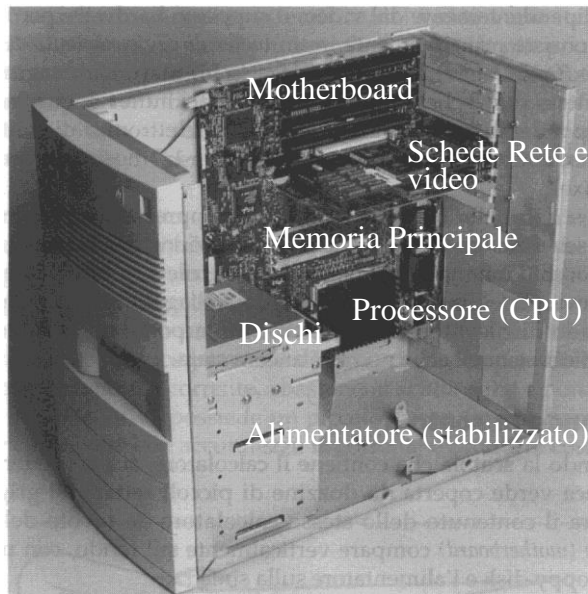


Descrizione di un elaboratore





Struttura dell'elaboratore



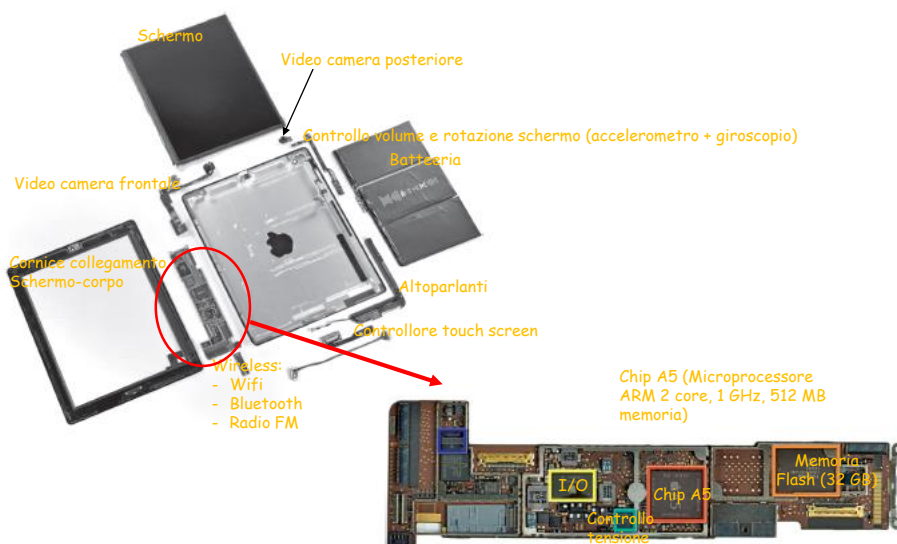
A.A. 2024-2025

29/69

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



Struttura di un PMD - I-Pad2



A.A. 2024-2025

30/69

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



Struttura dell'elaboratore: descrizione



- Elementi principali di un elaboratore:
 - Unità centrale di elaborazione (*Central Processing Unit - CPU*), processore.
 - Memoria di lavoro o memoria principale (*Main Memory - MM*) e dischi.

- Collegamenti con gli altri dispositivi di un elaboratore:
 - Bus di sistema (dati, indirizzi, controllo)
 - Bridge
 - Bus di I/O (USB, Firewire).

Dispositivi di *Input/Output - I/O*:

- memoria di massa (dischi magnetici o a stato solido, pen drive),
- rete,
- altri dispositivi.



Unità centrale di elaborazione o processore (*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*), sommatori ausiliari, ecc.;
- **Unità di controllo**. Controlla il flusso e determina le operazioni di ciascun blocco.



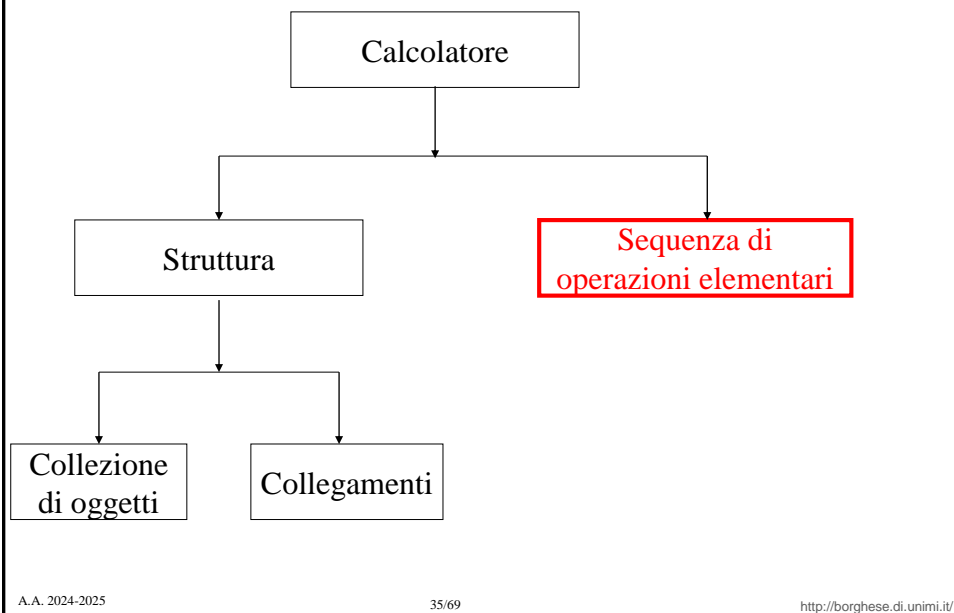
Sommario della lezione



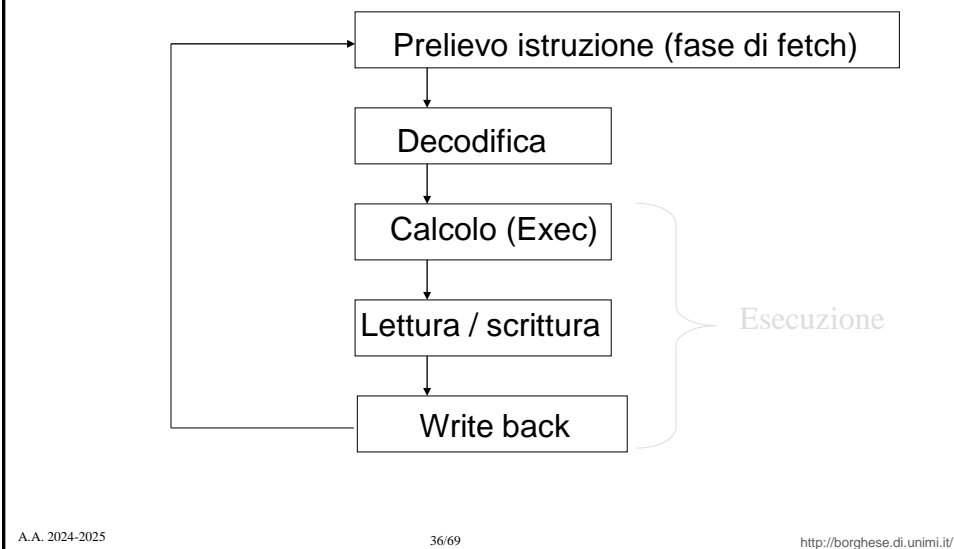
- Informazioni su corso ed esame
- Architettura dell'elaboratore
- **Ciclo di esecuzione di un'istruzione**
- Storia dell'elaboratore.



Descrizione di un elaboratore



Ciclo di esecuzione di un'istruzione MIPS





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



Lettura 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.



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



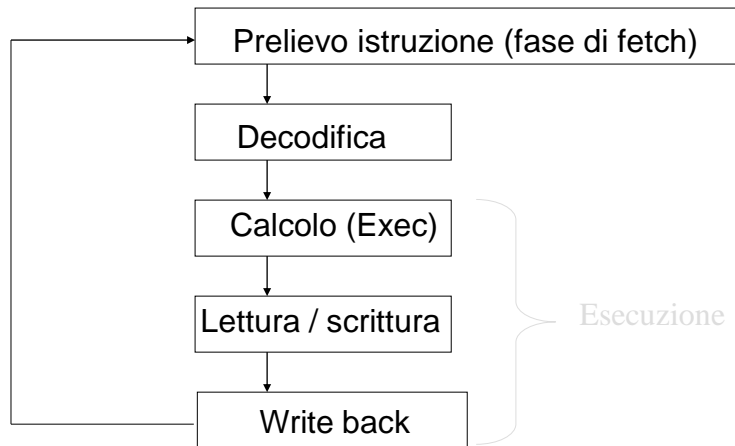
Viene selezionato il circuito / i circuiti appropriati per l'esecuzione delle operazioni previste dall'istruzione e determinate in fase di decodifica.

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





Ciclo di esecuzione di un'istruzione MIPS



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:

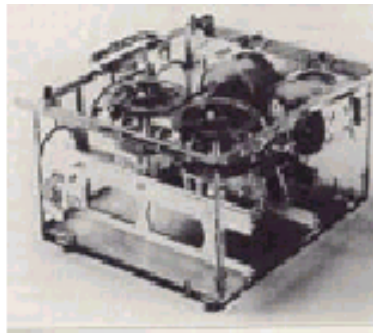
- Aumento della velocità di elaborazione
- Diminuzione della dimensione dei componenti.
- Aumento della capacità e velocità dell'I/O.
- Adozione di tecnologie diverse (meccanica, elettrica, elettronica).



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.



Un'architettura efficace



Una macchina per risolvere un problema industriale.

Telaio Jacquard (1801)

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





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

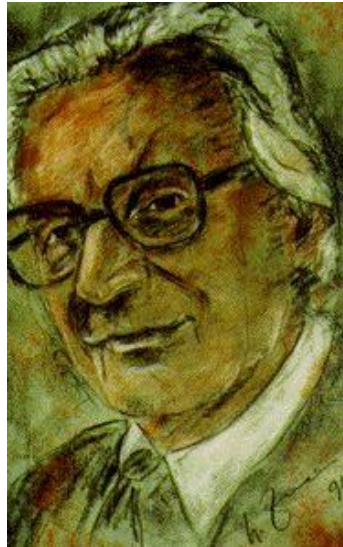


Konrad Zuse, 1936
Ingegnere civile.

Z1 -> 1938

Z3 -> 1941

Auto-ritratto del 1994



Storia dell'elaboratore - Mark I - 1944



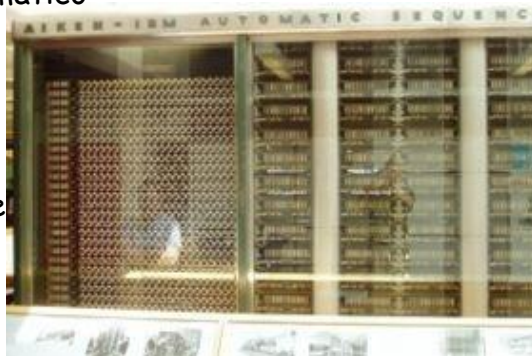
Primo computer automatico

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. 2024-2025

53/69

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



La prima generazione (ENIAC: 1946-1955)

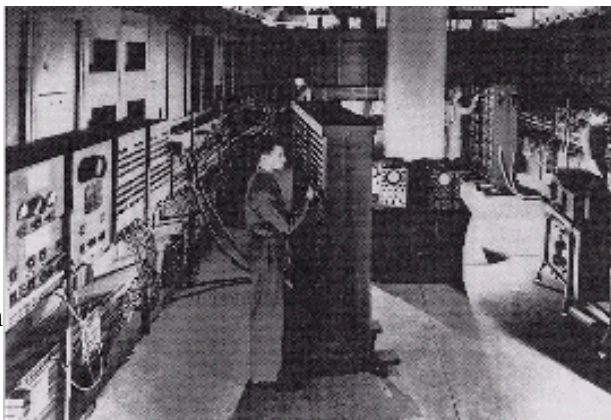


Elettronica (valvole: diodo, triodo). Aumento di prestazioni di 1,000 volte.

- 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 <u>film stock</u>	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. 2024-2025

55/69

<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),
Echbert e Mauchly.
E' il primo calcolatore commercializzato.

48 esemplari a 1M\$



A.A. 2024-2025

56/69

<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. 2024-2025

57/69

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



La comunicazione tra i componenti



Switch centralizzato
(multiplexer)

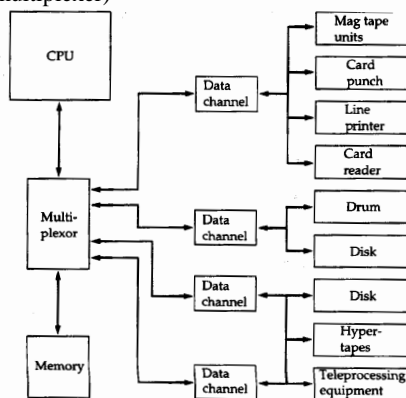
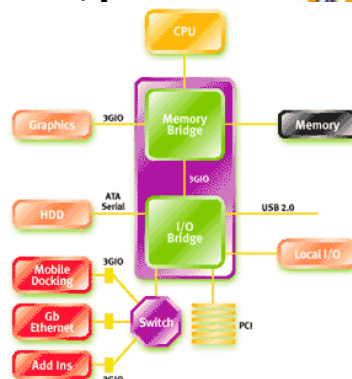


Figure 2.5 An IBM 7094 Configuration

Architettura a stella (bridge)



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

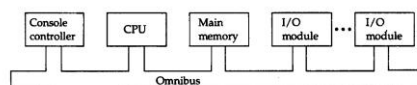


Figure 2.9 PDP-8 Bus Structure

Programma di “canale”

A.A. 2024-2025

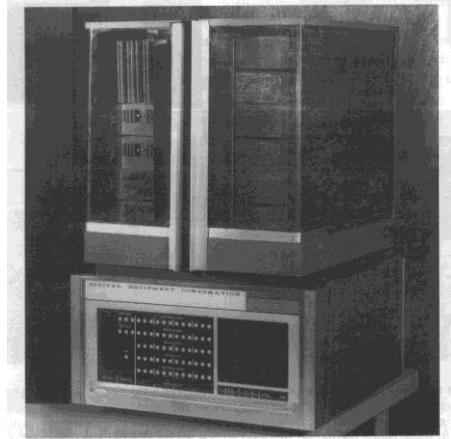
58/69



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. 2024-2025

59/69

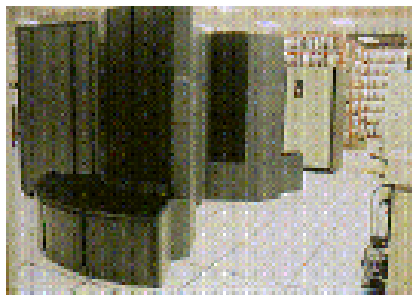
<http://borgnese.di.unimi.it/>



La quarta generazione (1971-1977)



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



A.A. 2024-2025

60/69

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



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. 2024-2025

61/69

<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

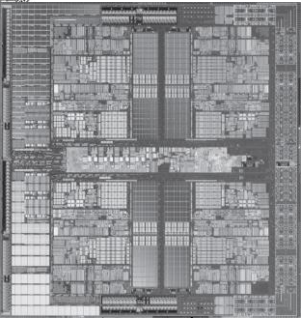
A.A. 2024-2025

62/69

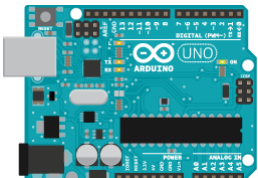
<http://borghese.di.unimi.it/>




La sesta generazione (postPC)




AMD Barcelona
(quad-core)



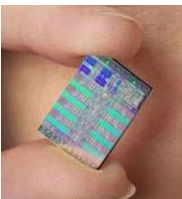
Arduino microcontroller
(M. Banzi)




Cloud TPU v3
420 teraflops
128 GB HBM




nVidia 9800 GTX, Streaming processors 128 core




Cell processor
(IBM, Sony, Toshiba)
Playstation 3




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



Smart watch 5
<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. 2024-2025

64/69

<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!)
- Domain Specific Architecture
- Macchine intelligenti e sensibili.
- Sistemi multimediali.



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).
- Fissi (desktop, workstation)
- 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.