
UNIMORE
UNIVERSITÀ DEGLI STUDI DI
MODENA E REGGIO EMILIA

Dipartimento di Scienze Fisiche,
Informatiche e Matematiche

# 2. I Computer: Astrazioni e Tecnologia

## Architettura dei calcolatori [MN1-1143]

*Corso di Laurea in INFORMATICA*
(D.M.270/04) [16-215]
Anno accademico 2020/2021

**Prof. Andrea Marongiu**
andrea.marongiu@unimore.it

# Perché la *computer science*?

- I computer sono **pervasivi**
- La loro continua evoluzione rende possibili applicazioni sempre nuove
  - *Dispositivi personali*
  - *Auto a guida autonoma, droni*
  - *Human brain project*
  - *World Wide Web/Internet, Cloud, IoT*
- Ciò è reso possible dal costante avanzamento della tecnologia
  - *Legge di Moore*

*«the number of transistors in an IC doubles every two years»*

Moore, G.E., *Cramming more components onto integrated circuits*. Electronics, 38(8), April 1965

Gordon E Moore (1929– )

# Classi di Computer

- Personal computers
  - *General purpose, varietà di software*
  - *Trade-off costi/prestazioni*

- Server computers
  - *Network*
  - *Alta capacità (storage), performance, affidabilità*
  - *Da piccolo server privati ad interi edifici*

# Classi di Computer

- Supercomputers
  - *Calcoli scientifici e ingegneristici ad alta richiesta di prestazioni*
  - *Prestazioni più alte in assoluto*
    - *Una frazione relativamente piccolo del totale mercato, ma sta crescendo…*

- Embedded computers
  - *Nascosti come component di un sistema*
  - *Vincoli di potenza/performance/costo stringenti*
  - *Requisiti di tipo Real-time e di affidabilità*

# L’era Post PC

- Personal Mobile Device (PMD)
  - *Dispositivi a batteria (basso consumo)*
  - *Alta connettività (Internet and more)*
  - *Centinaia di euro (basso costo)*
  - *Smart phones, tablets, electronic glasses*


- Cloud computing
  - *Warehouse Scale Computers (WSC)*
  - *Software as a Service (SaaS)*
  - *Portion of software run on a PMD and a portion run in the Cloud*
  - *Amazon, Microsoft, Google*

# Data Center

Microsoft Data Center eastern US - 2017

# Getting bigger


Planned expansion: 2km long…

# Tipi di Computer

SERVER

LAPTOP
NOTEBOOK
ULTRABOOK

DESKTOP

SMARTPHONE

TABLET

SMARTWATCH
SMARTBAND

Dimensioni maggiori
Alti consumi
Maggiore performance

Dimensioni ridotte
Basso consumo
Minore performance

# Tipi di Computer

SERVER

DESKTOP

**CLOUD COMPUTING**

LAPTOP
NOTEBOOK
ULTRABOOK

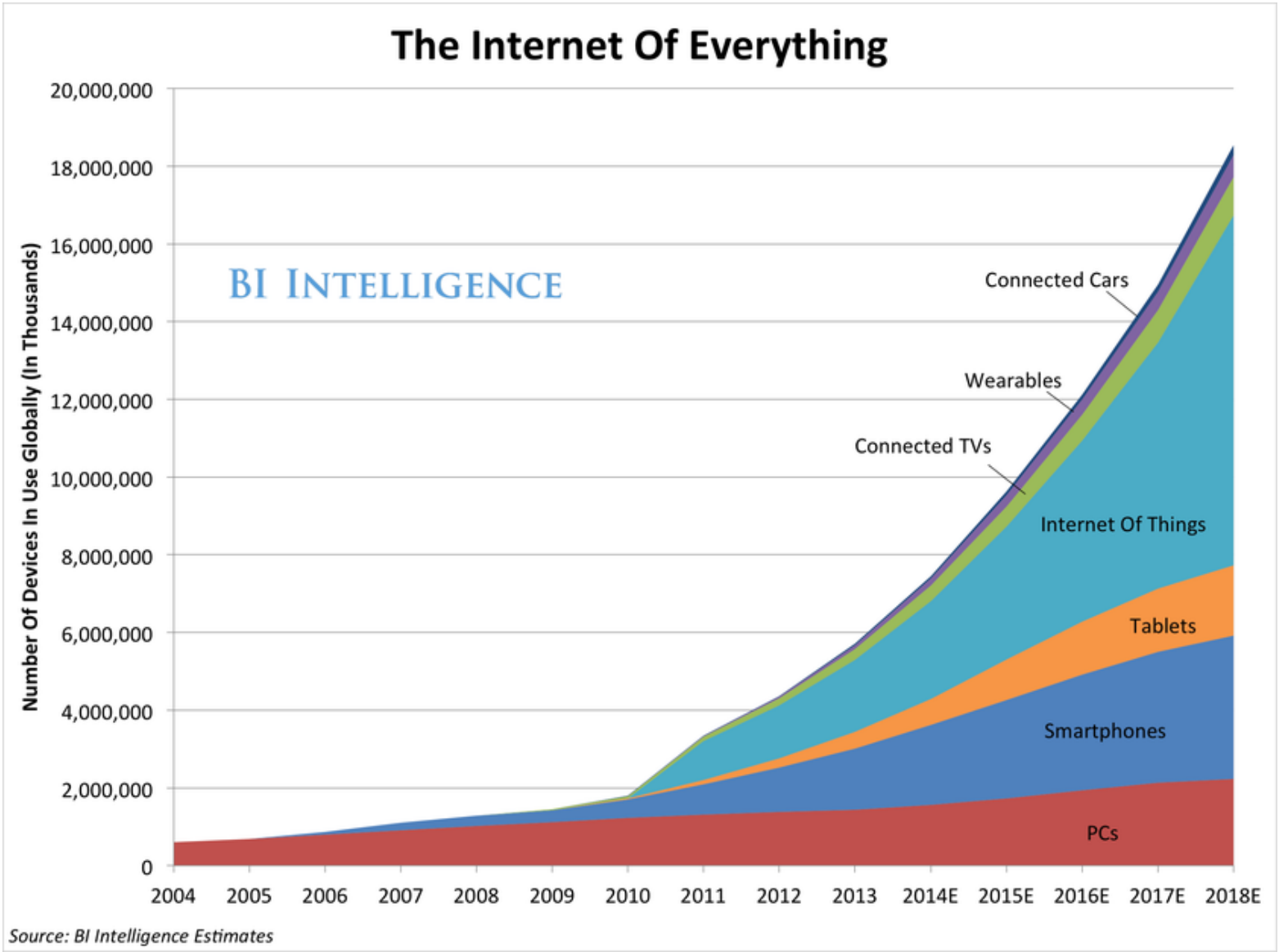
**INTERNET OF THINGS (IoT)**

SMARTPHONE

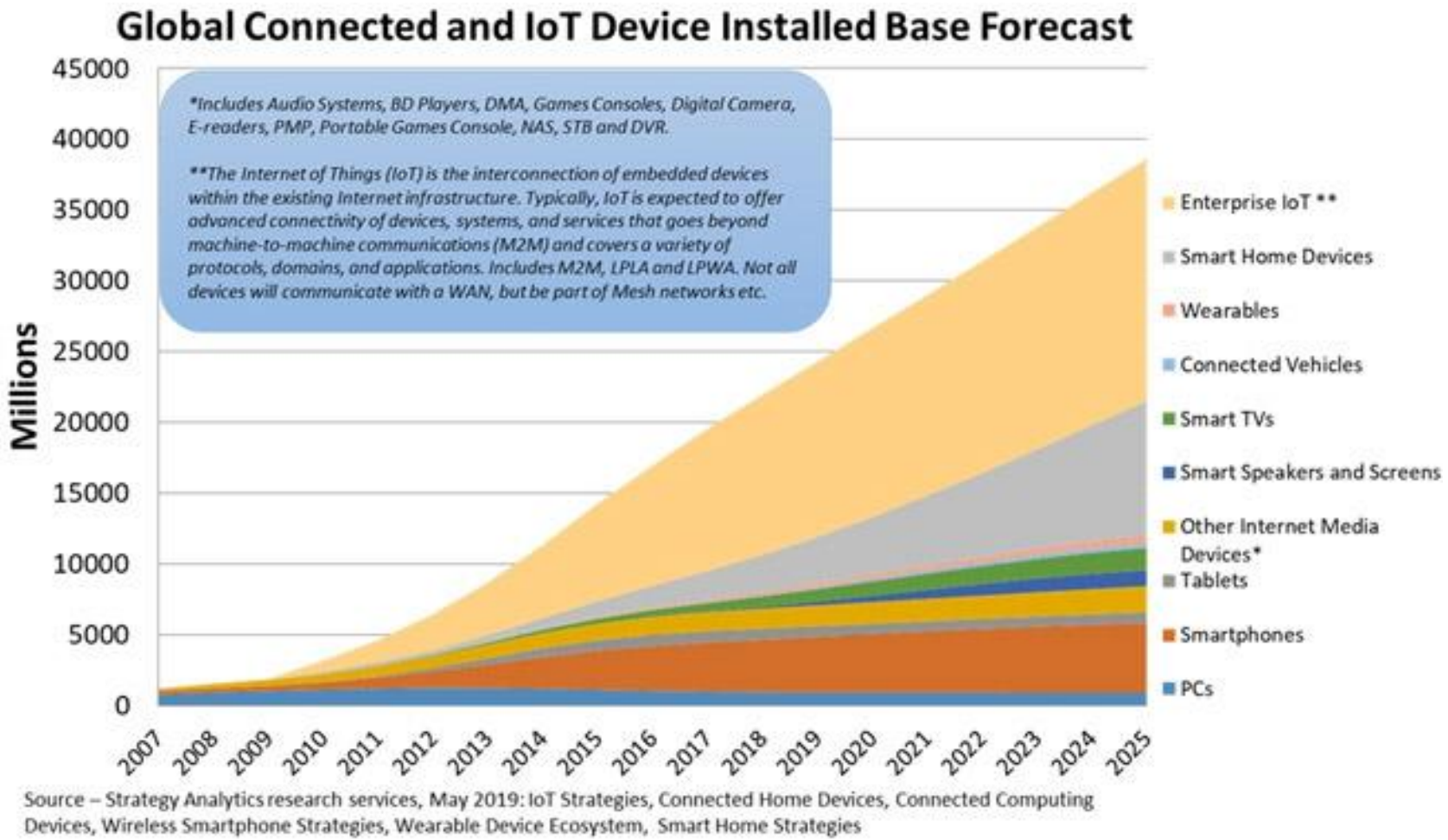
TABLET

SMARTWATCH
SMARTBAND

# L’era Post PC – la visione del 2013



# L’era Post PC – la visione odierna




Source – Strategy Analytics research services, May 2019: IoT Strategies, Connected Home Devices, Connected Computing Devices, Wireless Smartphone Strategies, Wearable Device Ecosystem, Smart Home Strategies

# Perché la *computer science*?

- In breve…
- I computer sono **pervasivi** e **onnipresenti** in tutti gli aspetti della nostra vita quotidiana
- **Lo studio** di come sono progettati e programmati i computer è **fondamentale** in un mondo (e un mercato) che è dominato da questa tecnologia

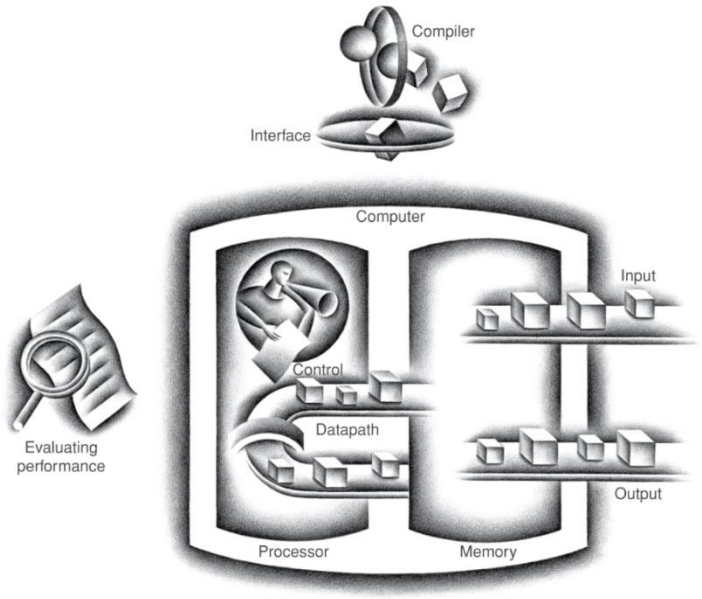
# Perché la *computer science*?

- Ma cosa c'è dentro a un computer?
- E i vari tipi di computer sono tanto diversi tra loro?

# Perché la *computer science*?

- **Ma cosa c'è dentro a un computer?**
- E i vari tipi di computer sono tanto diversi tra loro?

# Cosa c’è dentro a un computer?

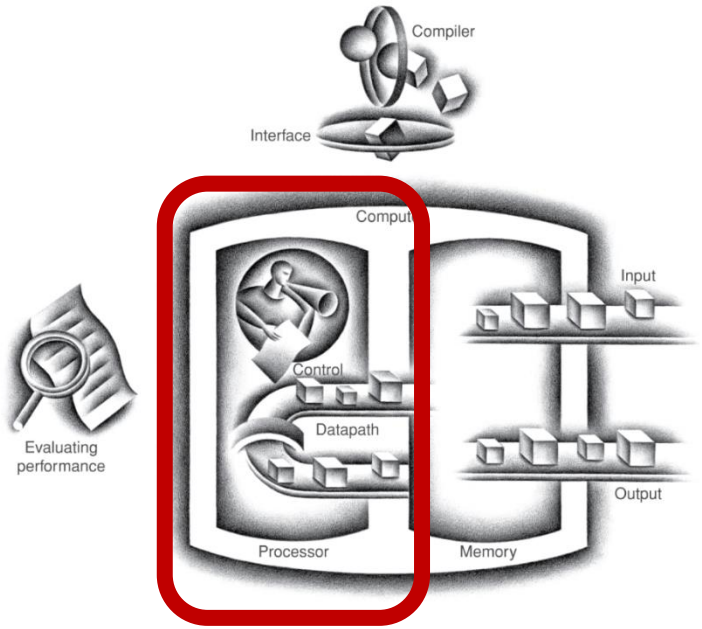



## Componenti di un computer

- **[SPOILER]** Gli stessi per tutti i tipi di computer
  - *Desktop, server, embedded*

- **Processore**
- **Memoria**
- **Input/Output (I/O)**

# Cosa c'è dentro a un computer?

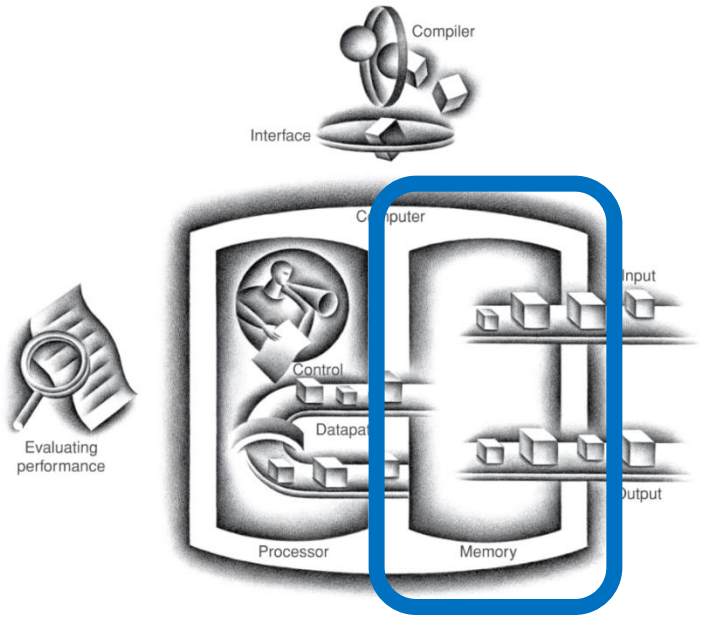



## Componenti di un computer

- **[SPOILER]** Gli stessi per tutti i tipi di computer
  - *Desktop, server, embedded*

- **Processore**
  - Central Processing Unit (**CPU**)
  - Il cuore di un computer: processa i dati in ***input*** per produrre l'***output***
  - **Datapath**: esegue le operazioni sui ***dati***
  - **Control**: controlla il funzionamento del datapath, memoria, ...

# Cosa c'è dentro a un computer?




## Componenti di un computer

- **[SPOILER]** Gli stessi per tutti i tipi di computer
  - *Desktop, server, embedded*

- **Memoria**
  - Organizzata come una ***gerarchia***
    - Performance/capacity/cost tradeoff
    - La ***DRAM*** è capiente, ma leeenta…
    - I ***registri*** (SRAM[1]) sono veloci quanto la CPU, ma piiiccoli…
  - La **cache** memory sta in mezzo
    - SRAM

*1. SRAM is faster but less dense, and hence more expensive, than DRAM*

# Cosa c’è dentro a un computer?

## E se spengo il computer?

- La memoria principale (DRAM) è ***volatile***
  - Perde tutta l’informazione quando spengo il PC

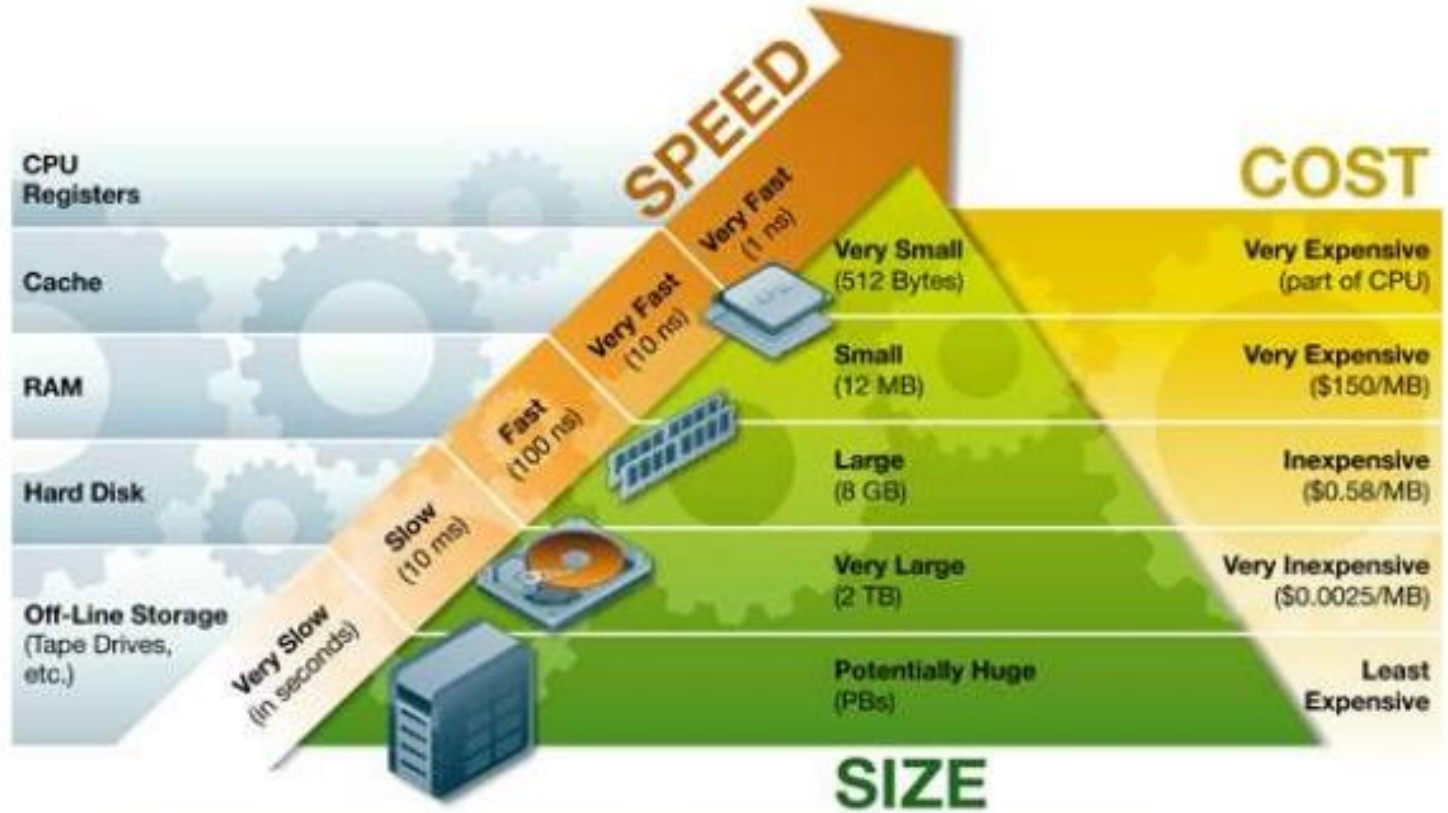

- HARD DISK – Memoria secondaria non-volatile
  - Magnetic disk (HDD)
  - Flash memory (SD, SSD)
  - Optical disk (CDROM, DVD)

HDD

SSD

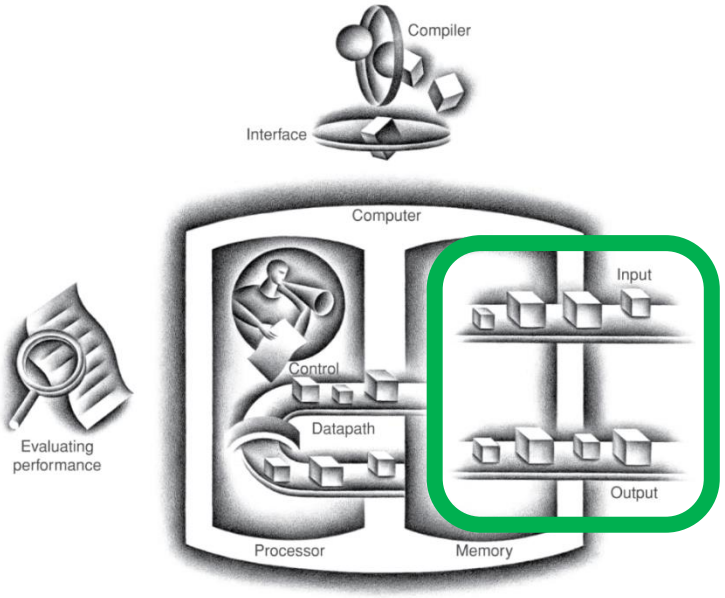
# Cosa c'è dentro a un computer?

## The memory hierarchy




Source: http://www.ts.avnet.com/uk/products_and_solutions/storage/hierarchy.html

# Cosa c'è dentro a un computer?



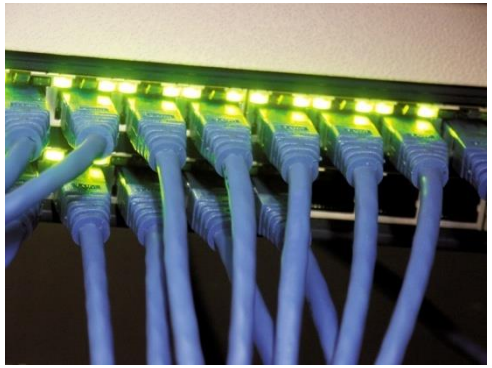

## Componenti di un computer

- **[SPOILER]** Gli stessi per tutti i tipi di computer
  - *Desktop, server, embedded*
- **Input/output (I/O)**
  - *User-interface devices*
    - *Display, keyboard, mouse*
  - ***Network adapters***
    - *For communicating with other computers*

# Cosa c'è dentro a un computer?

## Network adapters

- Communication, resource sharing, nonlocal access
- Local area network (LAN)
  - Ethernet (10/100 Gbit/s)
- Wide area network (WAN): the Internet
- Wireless network (IEEE 802.11)
  - WiFi, Bluetooth → 1-100 Mbit/s

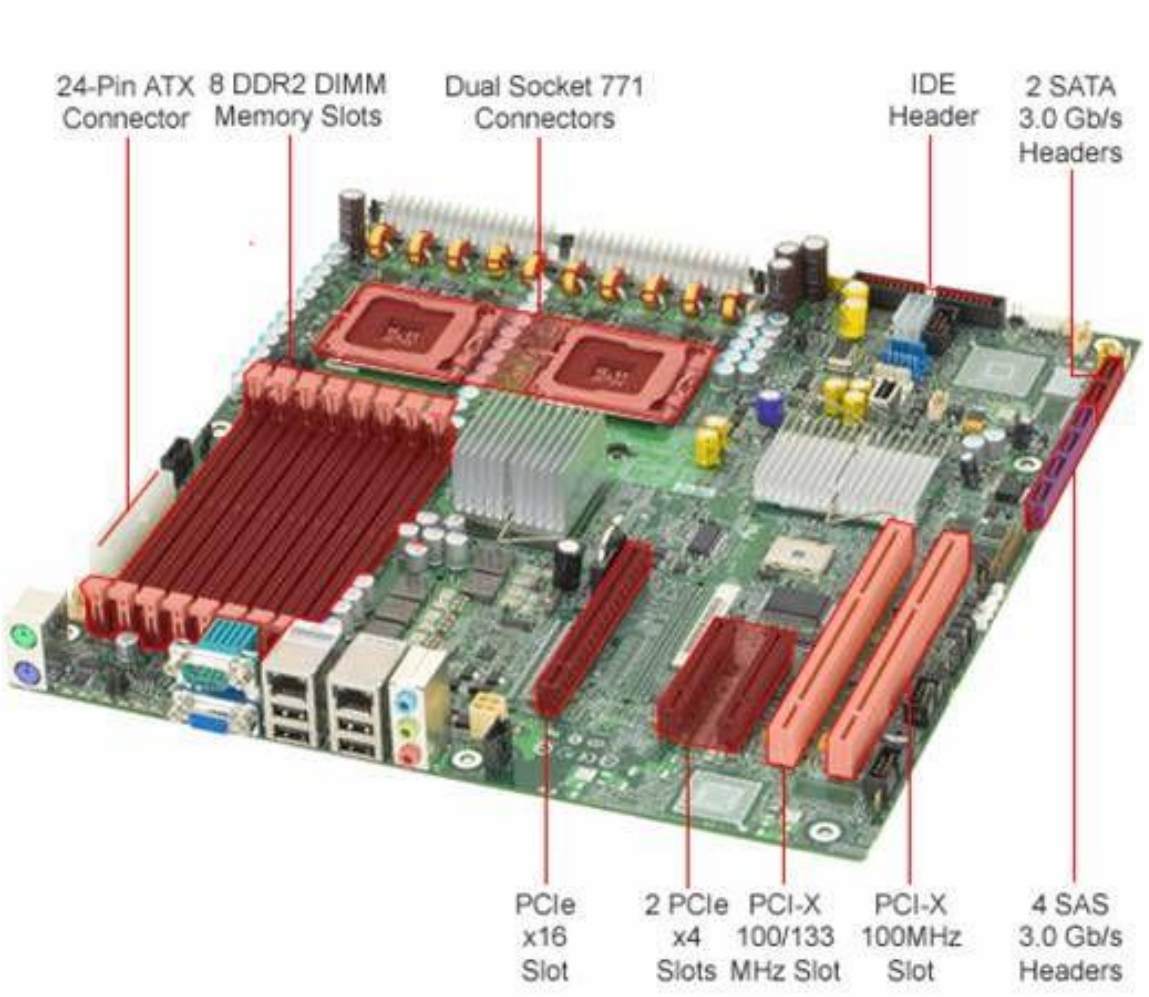


# Perché la *computer science*?

- Ma cosa c'è dentro a un computer?
- **E i vari tipi di computer sono tanto diversi tra loro?**

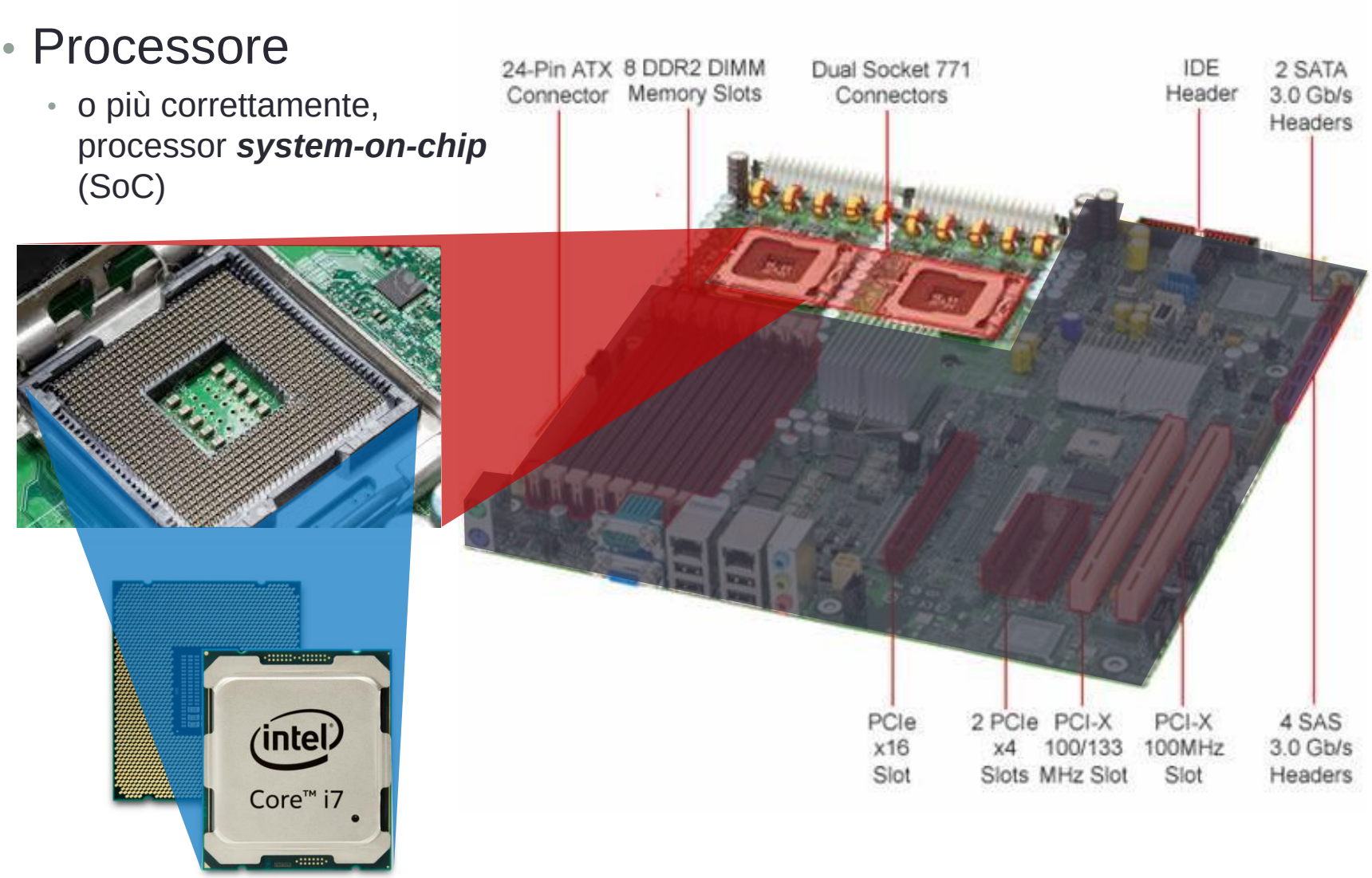
# Anatomia di un PC

- Desktop PC
- SCHEDA MADRE

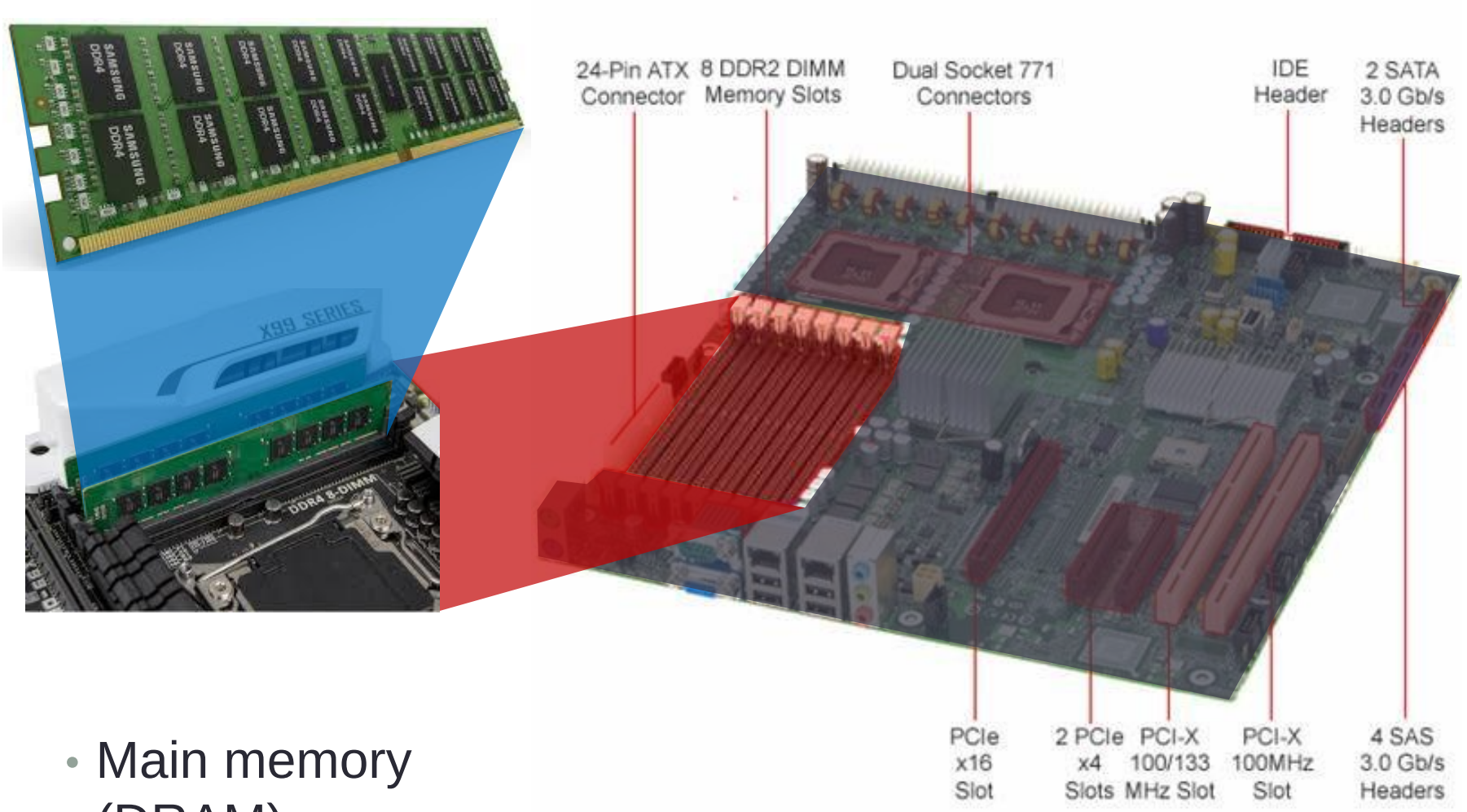


# Anatomia di un PC

- Processore
  - o più correttamente, processor ***system-on-chip*** (SoC)



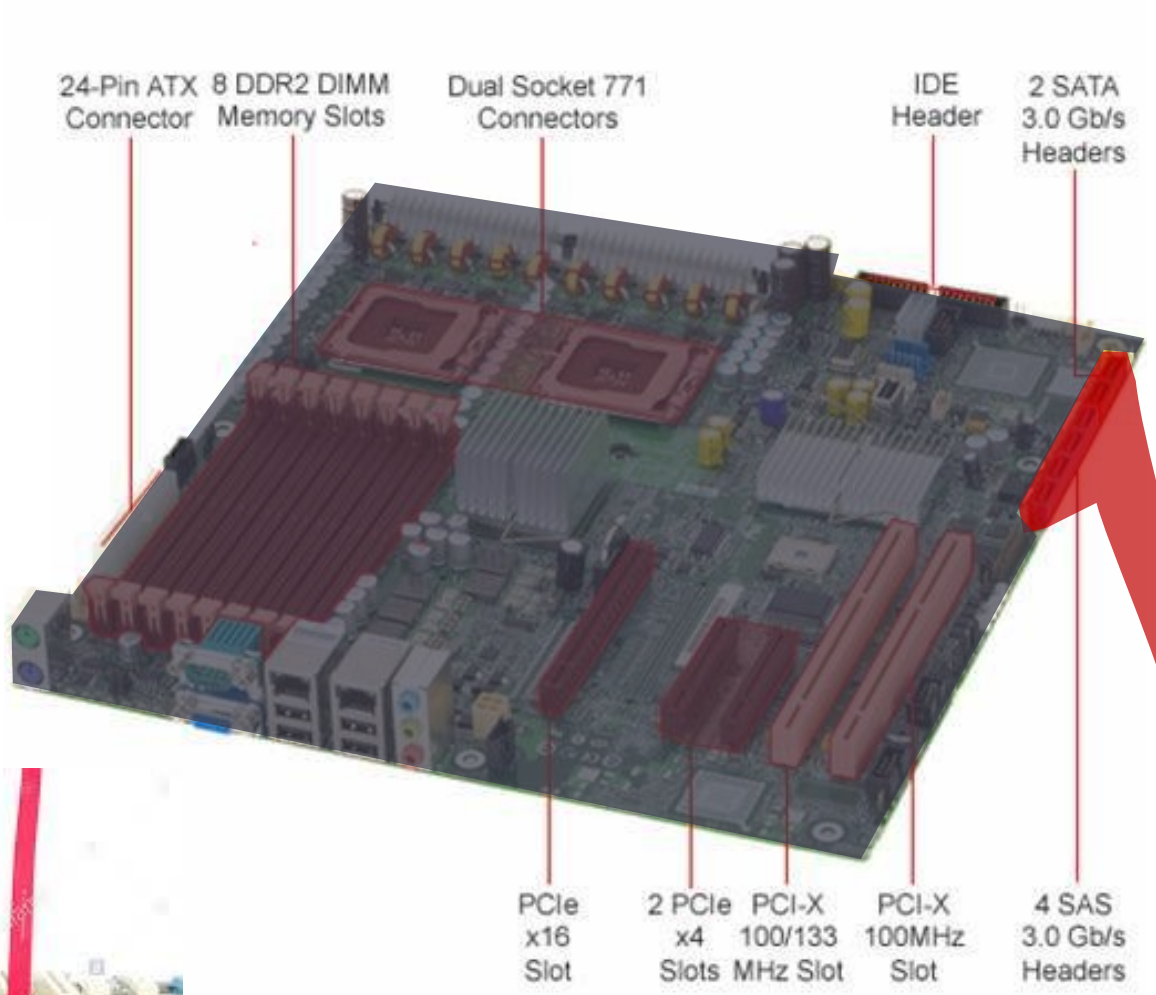
# Anatomia di un PC




- Main memory (DRAM)

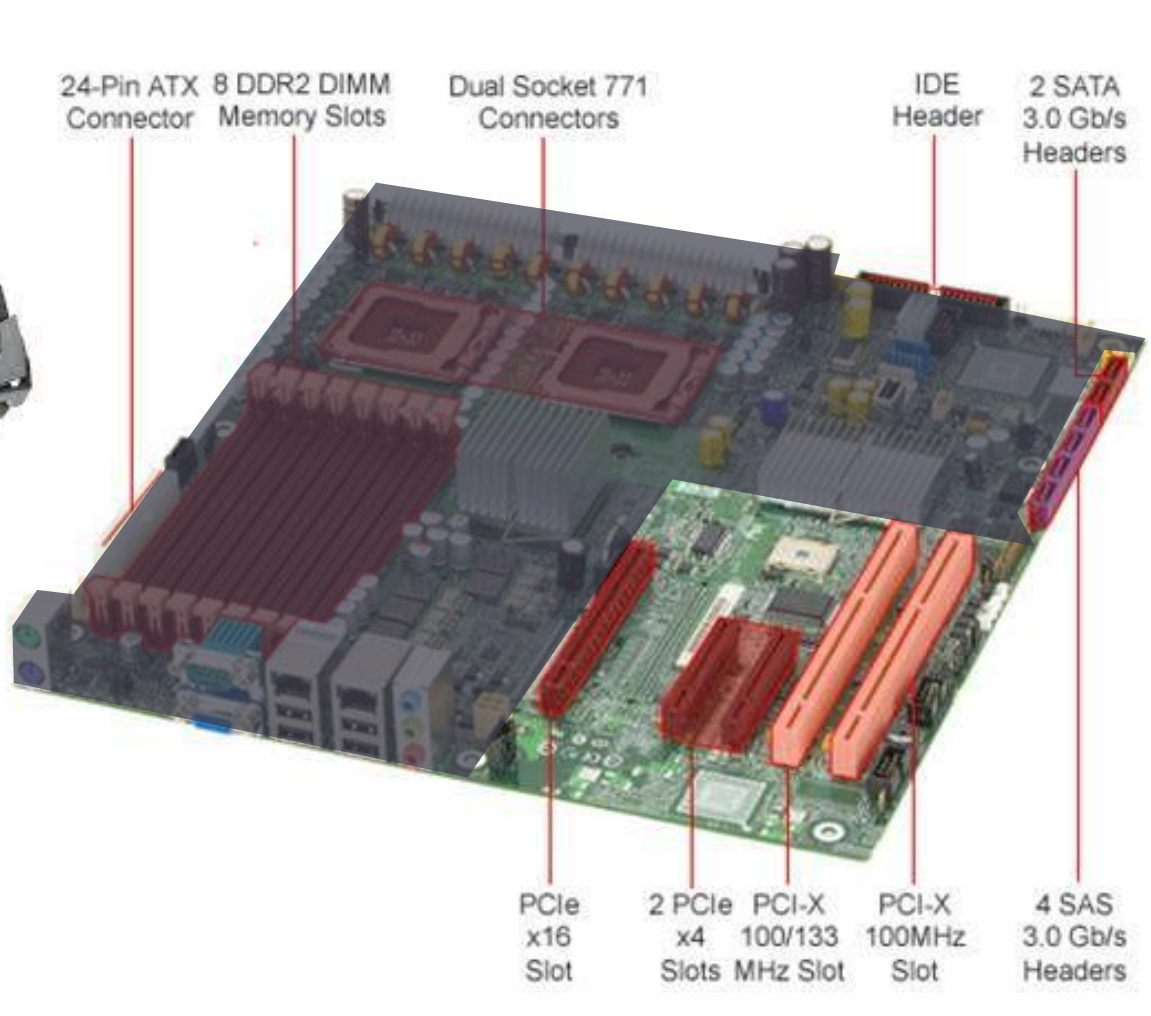
# Anatomia di un PC

- HARD DISK (solid state disk)

# Anatomia di un PC

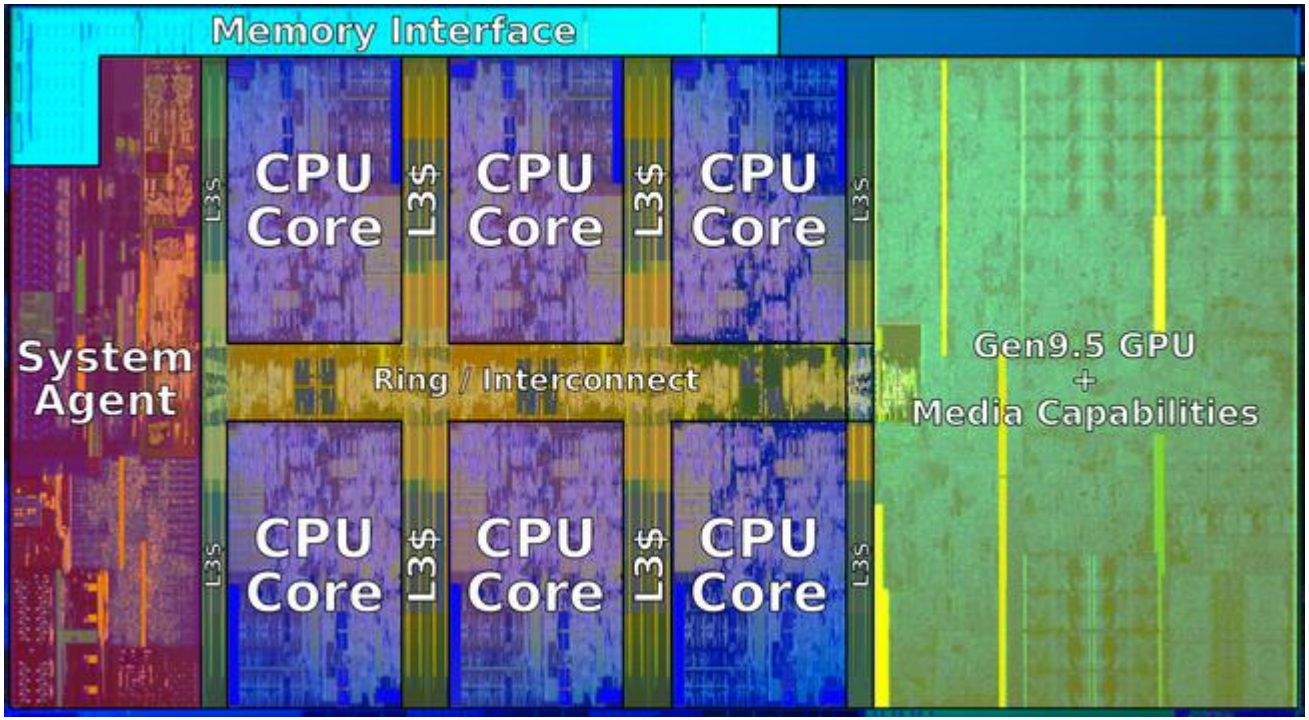
- GPU
  - graphics acceleration

# Anatomia di un PC

E cosa c'è dentro al **Processor SoC?**

- Coffee Lake *system-on-chip (2018)*
- **Processo a 14nm**
- **Multi-Core (2 to 8) CPU**
- **GPU**
- **Media**



# Anatomia di un Tablet

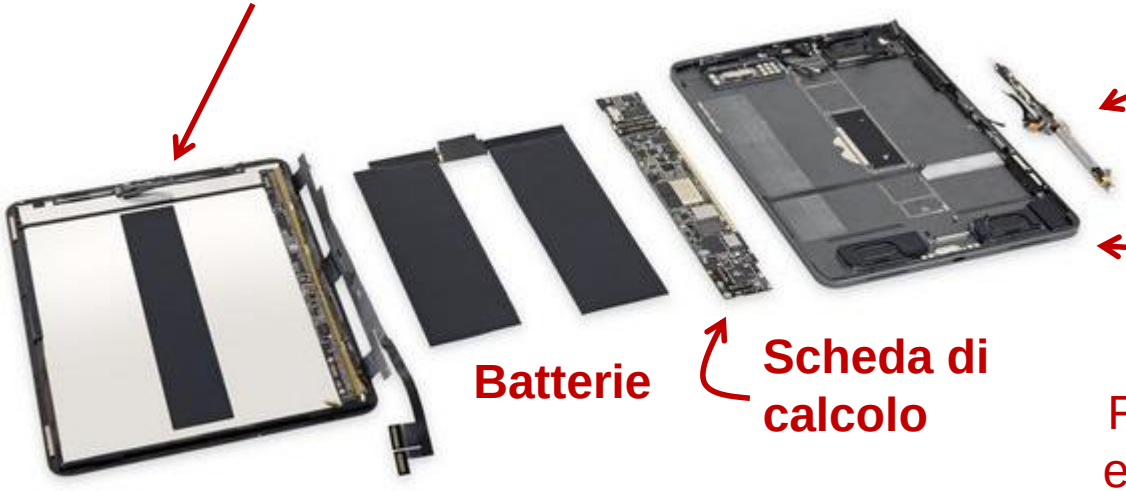
- Un dispositivo dell'era **PostPC**
- Touchscreen
  - rimpiazza keyboard e mouse
  - di tipo Capacitivo
    - consente multipli punti di "tocco" simultaneamente

iPad Pro 11", 2018

# Anatomia di un Tablet

**Display** TFT da 11" Liquid Retina
2388×1668 (264 ppi)

**Apple pencil**

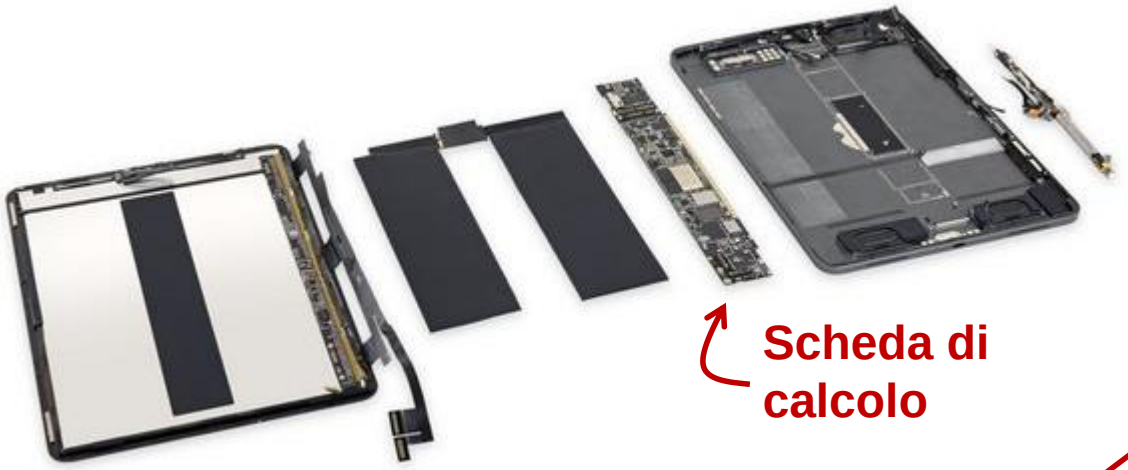


**Batterie**

**Scheda di calcolo**

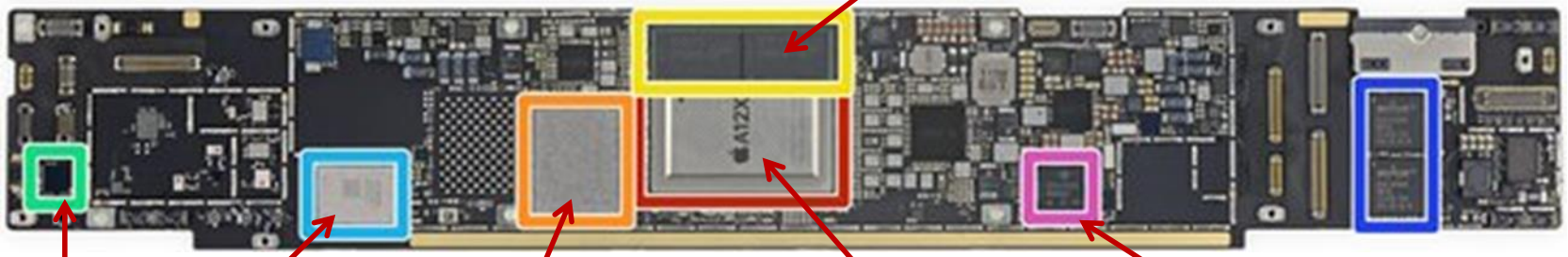
Fotocamera posteriore da 12 MP e fotocamera TrueDepth da 7 MP
quattro altoparlanti, Face ID,
cinque microfoni, sensore luce ambiente, accelerometro,
barometro e giroscopio a tre assi
Wi-Fi + Bluetooth 5.0

# Anatomia di un Tablet



Scheda di calcolo

2x RAM (4 GB)




Power, I/O controllers

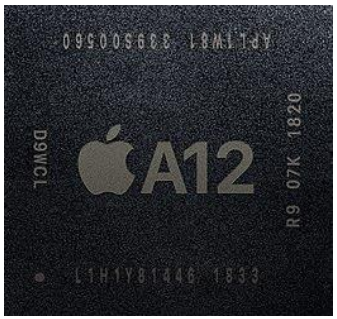
HARD DISK Flash 64 GB

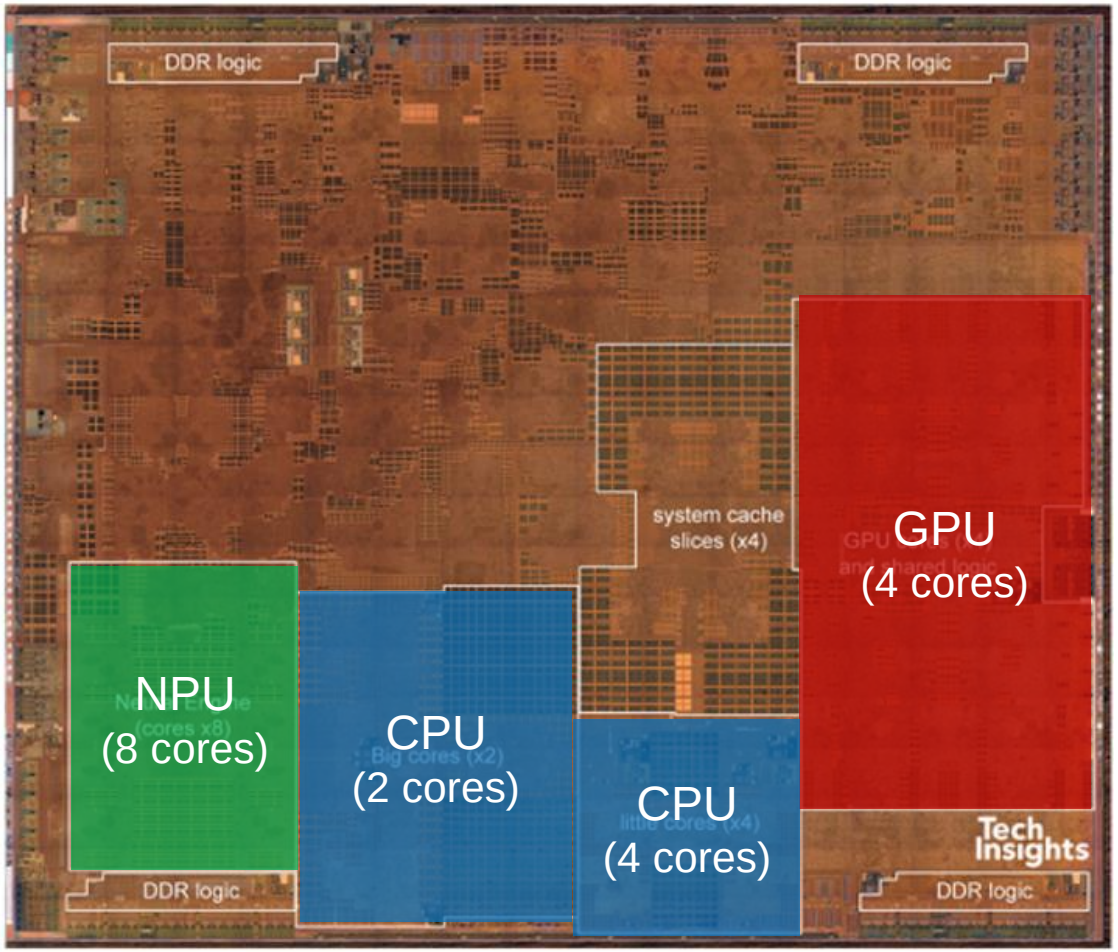
Processor SoC

# Anatomia di un Tablet

E cosa c'è dentro al **Processor SoC?**

- Apple A12 Bionic *system-on-chip*
  - **83,27 mm²**
    - (8,42 mm x 9,89 mm)
  - **Processo a 7nm**
  - **6,9 miliardi di transistor**
  - **Hexa-Core (6) CPU**
    - 2 (Vortex) @2,49GHZ
    - 4 (Tempest) @1,49GHz
  - **4-Core GPU**
  - **8-Core NPU**



# Architettura dei Calcolatori

- Quindi in questo corso si studia l'**anatomia dei calcolatori**?
  - Sì, ma non solo…

# L’astrazione nei calcolatori


**I PROGRAMMI**

(C/C++, JAVA, OpenCV, CUDA…)

compiler

linguaggio macchina

**IL COMPUTER**

**LE DISCIPLINE**

- Programmazione
- Compilatori
- *sistemi operativi*
- **Architettura dei calcolatori**

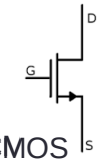
# L’astrazione nei calcolatori


**I PROGRAMMI**

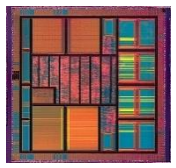
(C/C++, JAVA, OpenCV, CUDA…)

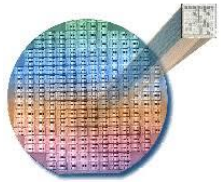
**IL COMPUTER**

VLSI design




transistor CMOS



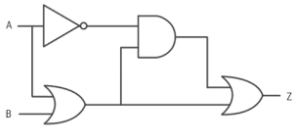


- Livello della progettazione su silicio (CMOS)

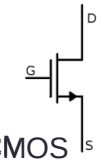
# L’astrazione nei calcolatori


**I PROGRAMMI**
(C/C++, JAVA, OpenCV, CUDA…)

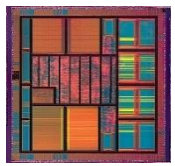
**Logic circuits**

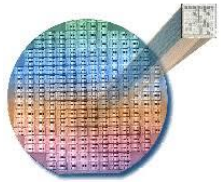



VLSI design




transistor CMOS



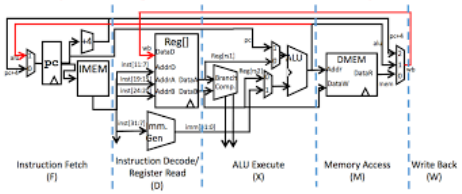


- Livello della progettazione con logica binaria
- Livello della progettazione su silicio (CMOS)

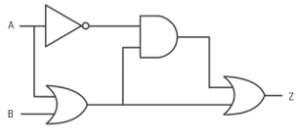
# L’astrazione nei calcolatori


**I PROGRAMMI**
(C/C++, JAVA, OpenCV, CUDA…)

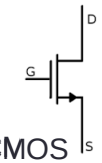
**CPU (RISC-V)**



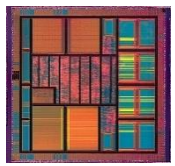

**Logic circuits**

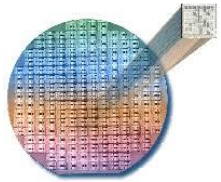



VLSI design




transistor CMOS



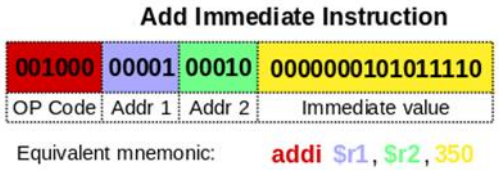


- Livello della progettazione architetturale
- Livello della progettazione con logica binaria
- Livello della progettazione su silicio (CMOS)

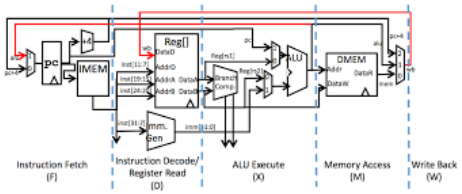
# L’astrazione nei calcolatori


**I PROGRAMMI**
(C/C++, JAVA, OpenCV, CUDA…)

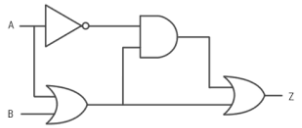
**ISA**



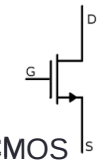

**CPU (RISC-V)**



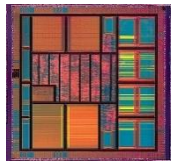

**Logic circuits**

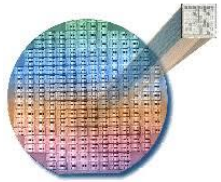



VLSI design




transistor CMOS





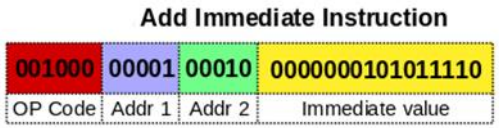
- Livello della progettazione di interfaccia col software
- Livello della progettazione architetturale
- Livello della progettazione con logica binaria
- Livello della progettazione su silicio (CMOS)

# L'astrazione nei calcolatori

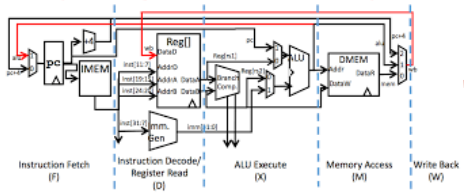
- Dai **transistor** ai **programmi**

**I PROGRAMMI**
(C/C++, JAVA, OpenCV, CUDA…)

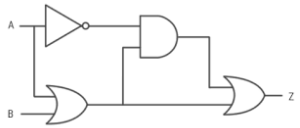
**ISA**



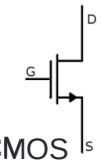

**CPU (RISC-V)**



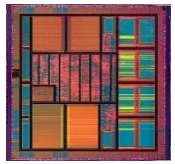

**Logic circuits**

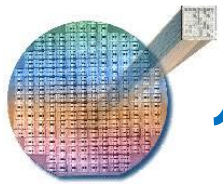



VLSI design




transistor CMOS





**Compilatori**: Come i programmi sono tradotti in linguaggio macchina

L'interfaccia hardware/software
- L'**Instruction Set Architecture** (ISA)

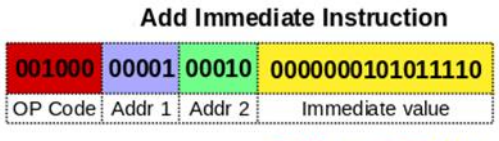
- Come l'hardware esegue il programma
- Ciò che determina la performance del programma e del sistema
- Come I progettisti hardware migliorano la performance

# L'astrazione nei calcolatori

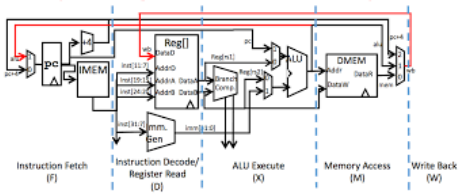
- Dai **transistor** ai **programmi**
  - O, limitandoci al programma del corso, dalle **reti logiche** all'**ISA**

I PROGRAMMI
(C/C++, JAVA, OpenCV, CUDA…)

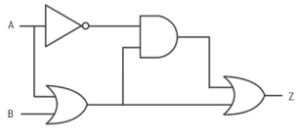
**ISA**




**CPU (RISC-V)**




**Logic circuits**




VLSI design

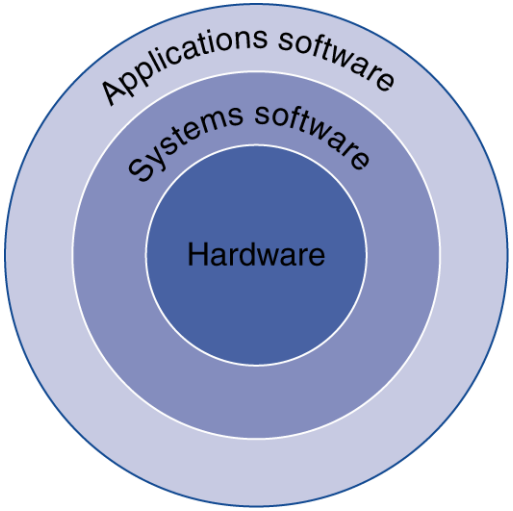

transistor CMOS

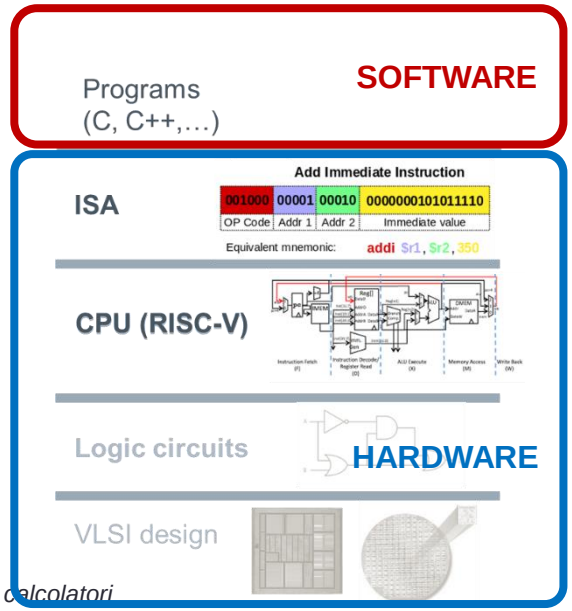
**Il contenuto principale del corso di Architettura dei Calcolatori**

**Ma prima di immergerci nel contenuto del corso…**

**Un po' di contesto**

# La dicotomia **Hardware/Software**




**Il vostro programma**

- Application **software**
  - *Written in high-level language*
- System **software**
  - ***Compiler**: translates HLL code to binary (machine) code*
  - ***Operating System**: service code*
    - *Handling input/output*
    - *Managing memory and storage*
    - *Scheduling tasks & sharing resources*

- **Hardware**
  - *Processor, memory, I/O controllers*

# Il codice del programma

- **Linguaggio di alto livello (C, C++, Java)**
  - Livello di astrazione prossimo al dominio del problema
  - Facilita la produttività e la portabilità
- **Linguaggio assembly**
  - Una rappresentazione delle istruzioni macchina comprensibile agli umani
- **Il binario (o eseguibile)**
  - Istruzione e dati codificati come stringhe di bit

High-level language program (in C)

```
swap(int v[], int k)
{int temp;
   temp = v[k];
   v[k] = v[k+1];
   v[k+1] = temp;
}
```

Compiler

Assembly language program (for RISC-V)

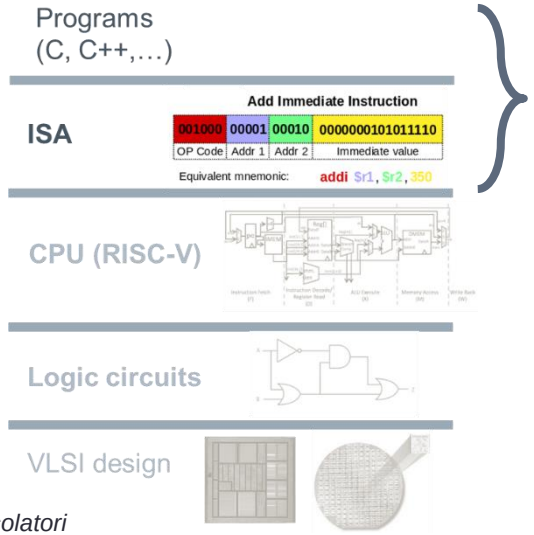
```
swap:
        slli x6, x11, 3
        add  x6, x10, x6
        ld   x5, 0(x6)
        ld   x7, 8(x6)
        sd   x7, 0(x6)
        sd   x5, 8(x6)
        jalr x0, 0(x1)
```

Assembler

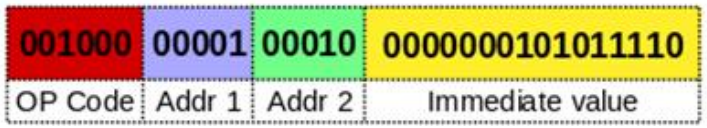
Binary machine language program (for RISC-V)

```
00000000001101011001001100010011
00000000011001010000001100110011
00000000000000011001001010000011
00000001000001100110011110000011
00000000011100110011000000100011
00000000010100110011010000100011
00000000000000001000000001100111
```

language of the CPU




ISA

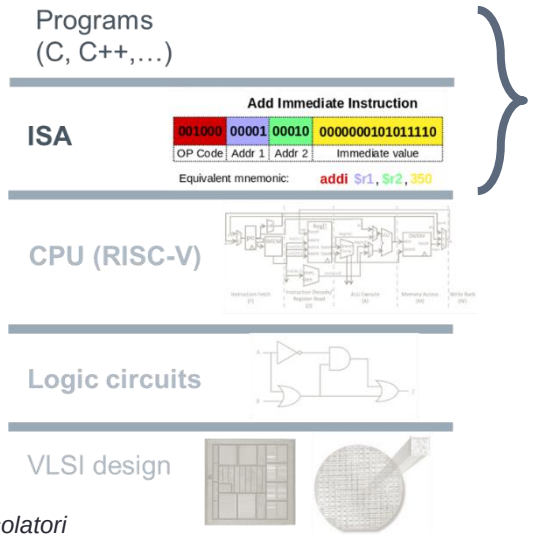


# L'Instruction Set Architecture (ISA)

- Una **instruction set architecture** (**ISA**) è un modello astratto di un computer

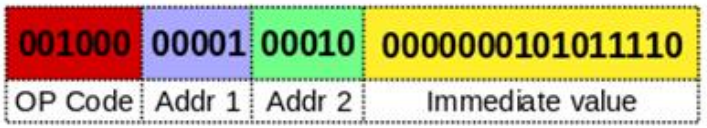
- Una realizzazione di una ISA, ovvero una CPU, è detta una ***implementazione*** di quell'ISA.

- Una ISA definisce i **tipi di dato** supportati, i **registri**, il supporto hardware per interagire con la **memoria principale** e l'**I/O**.




- L'**interfaccia HW/SW**

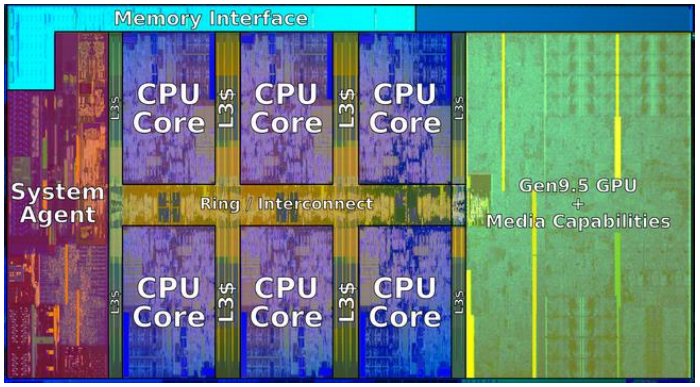
**ISA**



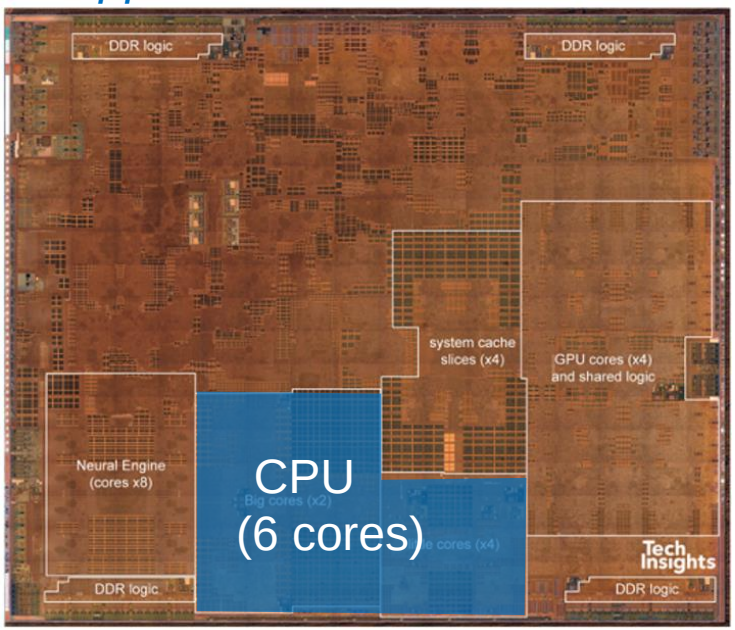
# La CPU: l'implementazione dell'ISA

- Vi ricordate i **Processor SoC** del PC e del Tablet?
  - Ognuno con tante CPU

*Intel i7 Coffee Lake SoC*



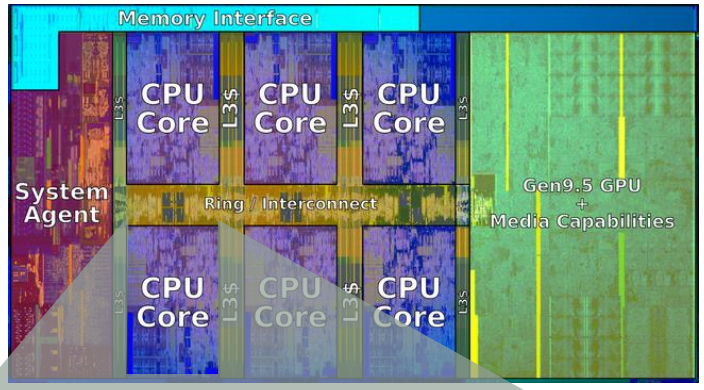

*Apple A12 Bionic SoC*



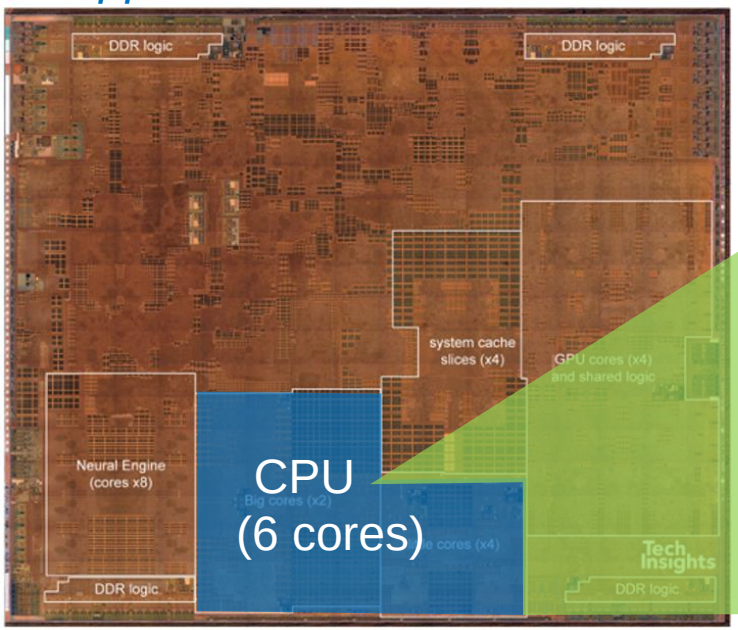
# La CPU: l'implementazione dell'ISA

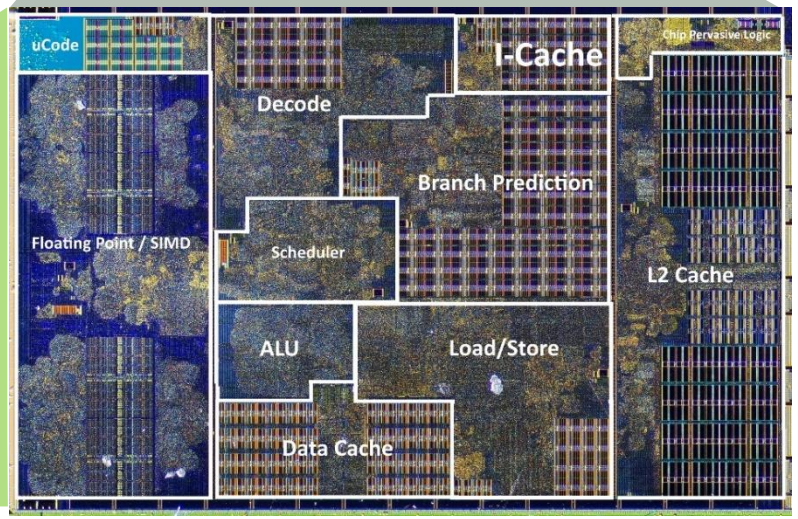
- Vi ricordate i **Processor SoC** del PC e del Tablet?
  - Ognuno con tante CPU

*Intel i7 Coffee Lake SoC*




*Apple A12 Bionic SoC*



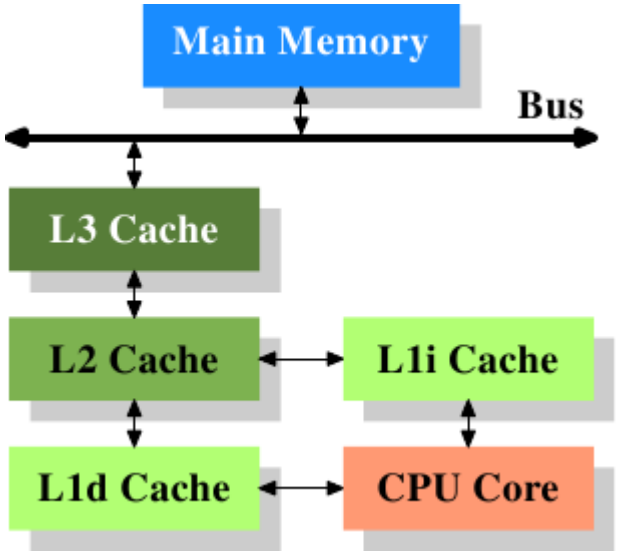



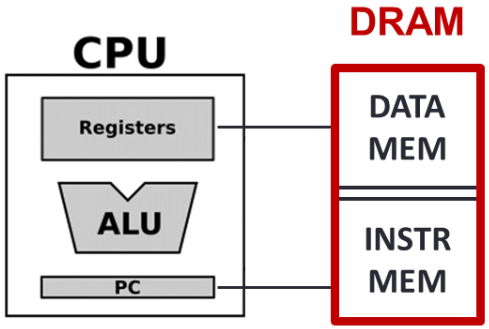

AMD Zen2 core

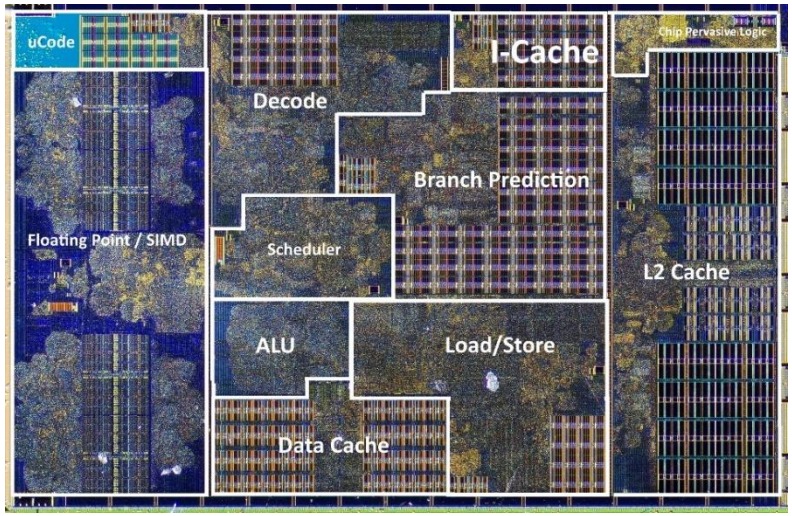
ATTENZIONE: non è la CPU di nessuno dei SoC

# La CPU: l'implementazione dell'ISA

- E vi ricordate la **gerarchia di memoria**?



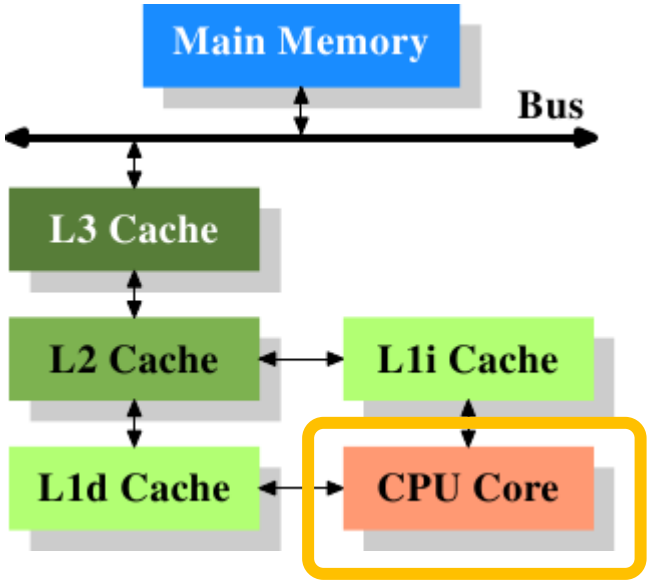



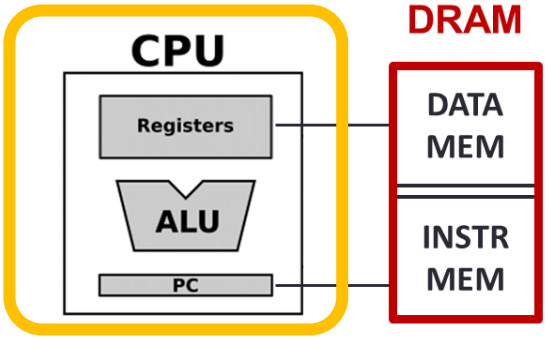



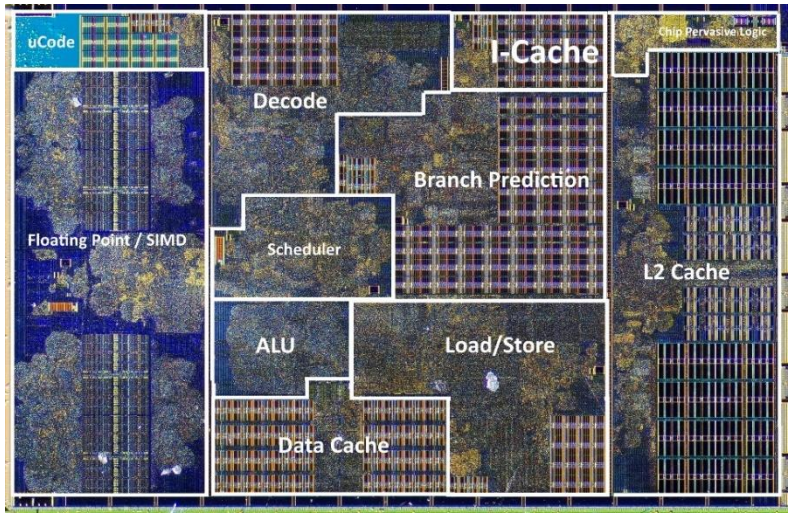

AMD Zen2 core

# La CPU: l'implementazione dell'ISA

- E vi ricordate la **gerarchia di memoria**?

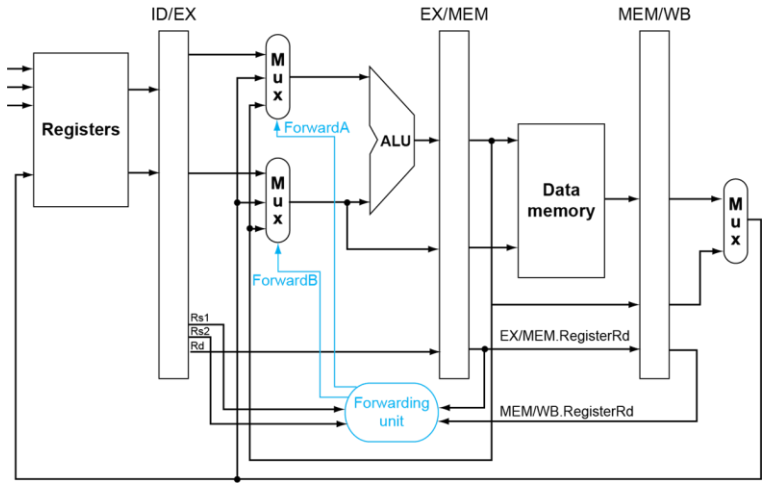


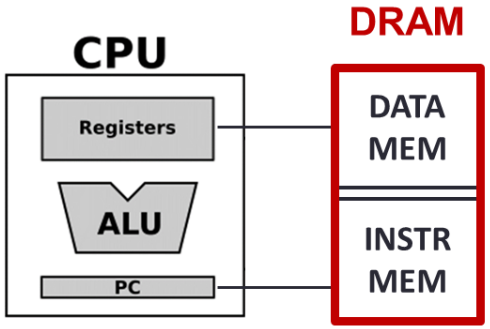



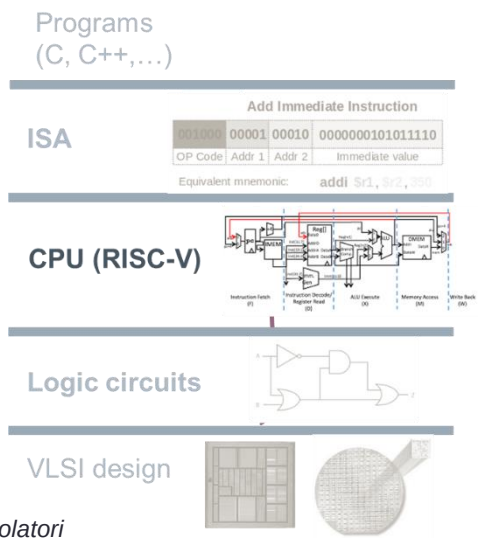

AMD Zen2 core

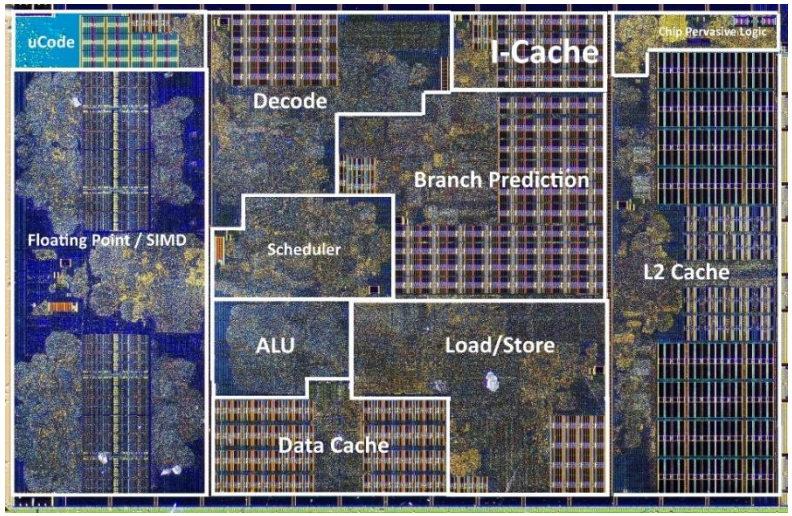
# La CPU: l'implementazione dell'ISA




- **CPU PIPELINE**
  - ALU
  - Registers
  - Data mem

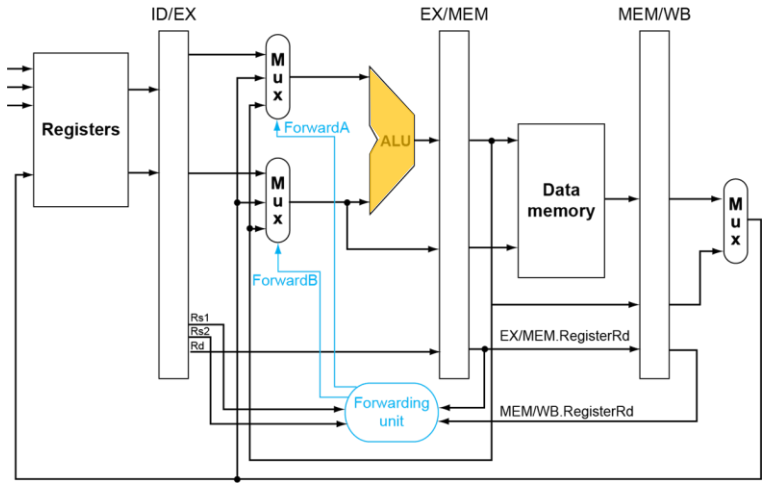


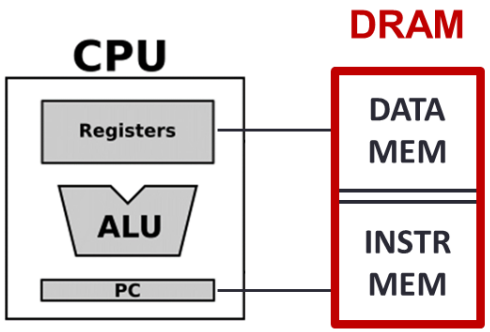



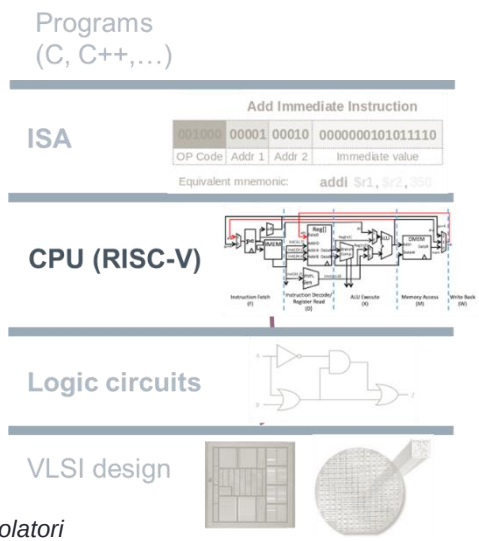

AMD Zen2 core

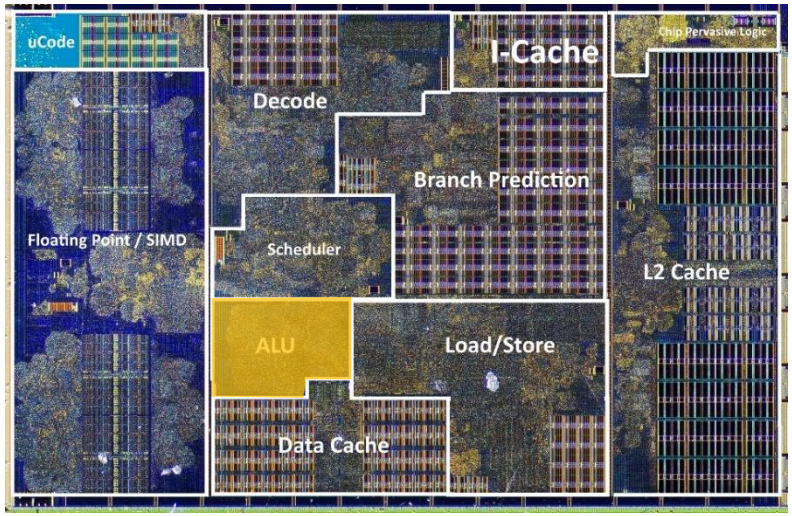
# La CPU: l'implementazione dell'ISA




- **CPU PIPELINE**
  - ALU
  - Registers
  - Data mem

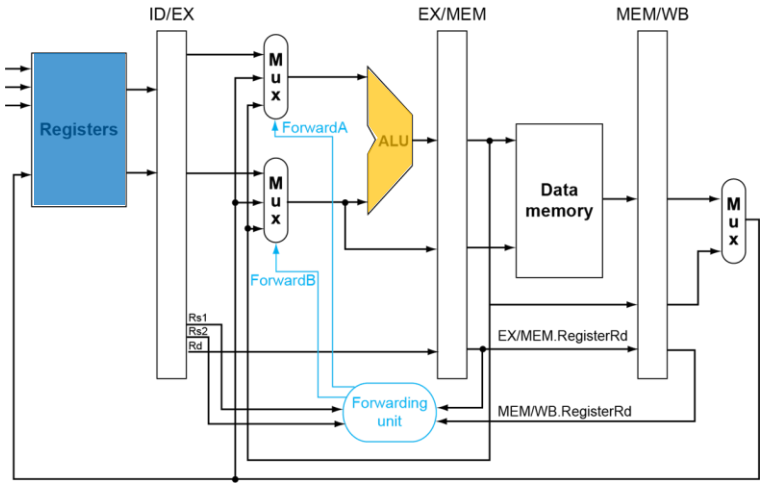


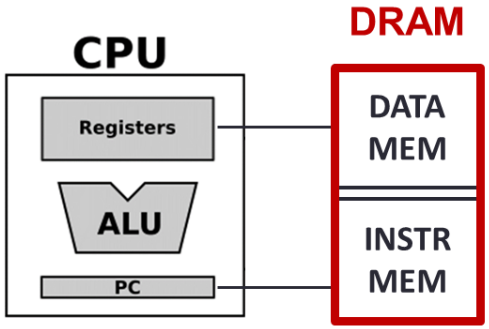



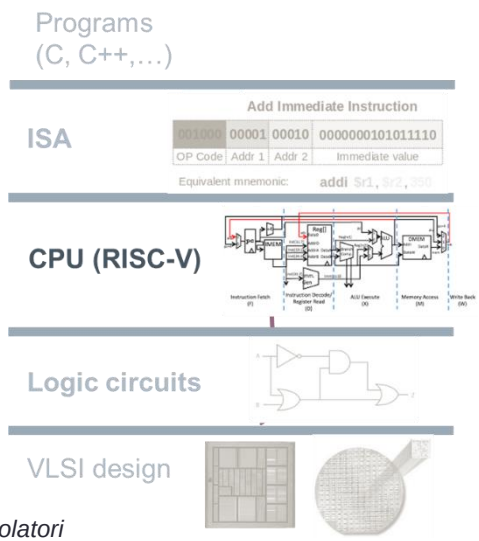

AMD Zen2 core

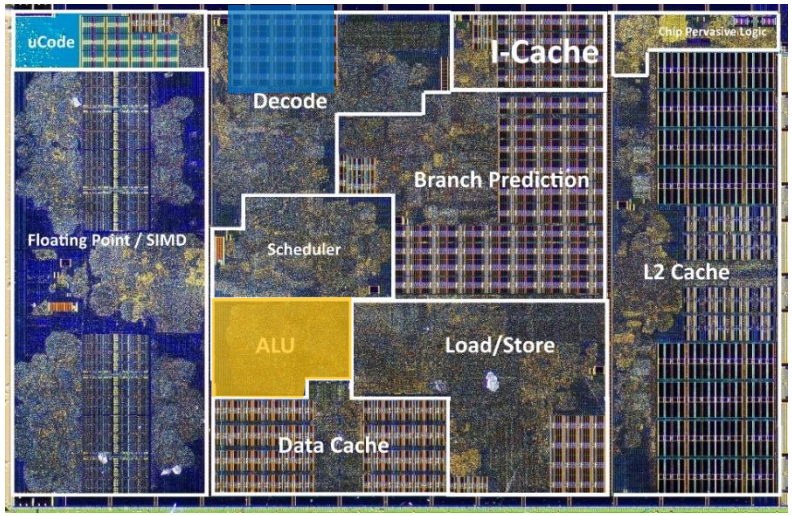
# La CPU: l'implementazione dell'ISA




- **CPU PIPELINE**
  - ALU
  - Registers
  - Data mem

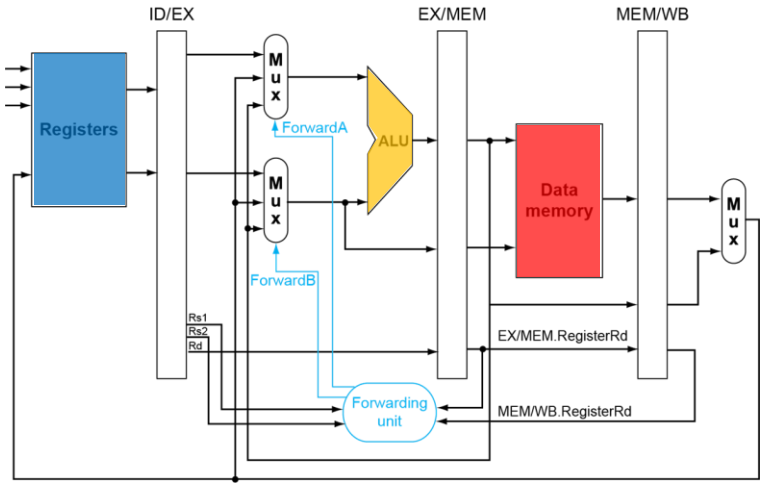


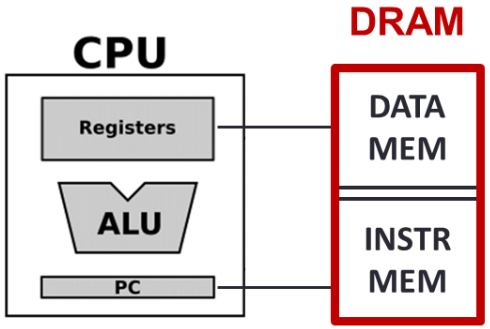



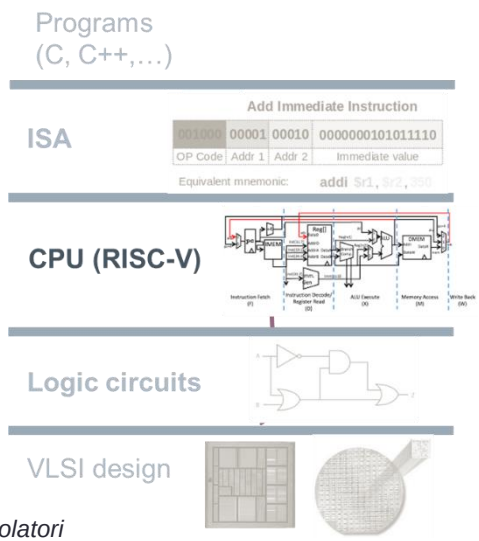

AMD Zen2 core

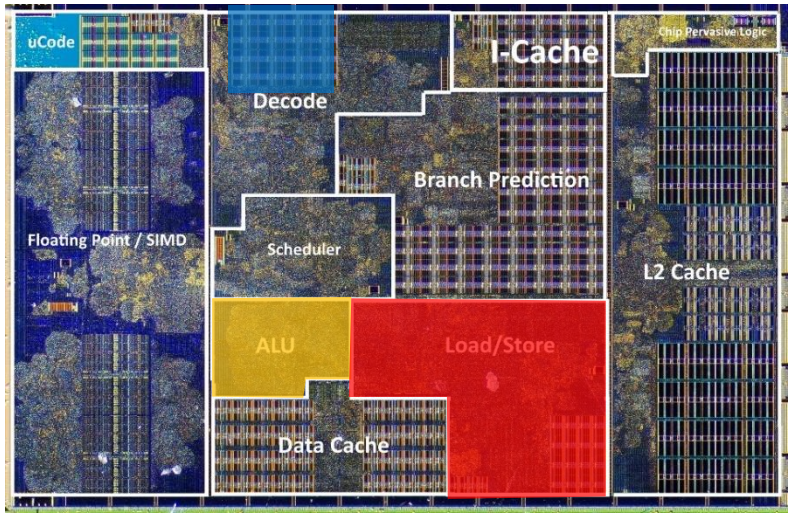
# La CPU: l'implementazione dell'ISA




- **CPU PIPELINE**
  - ALU
  - Registers
  - Data mem

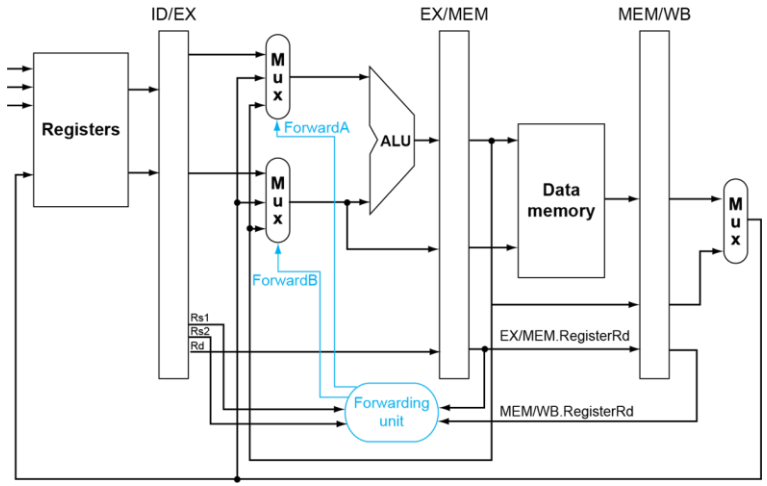


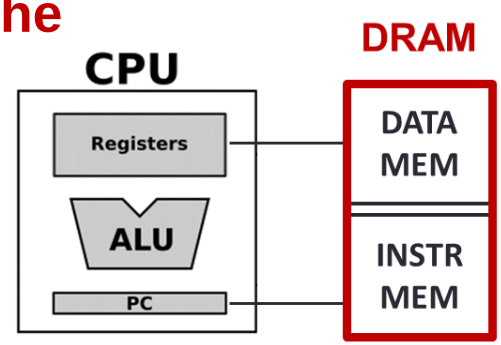



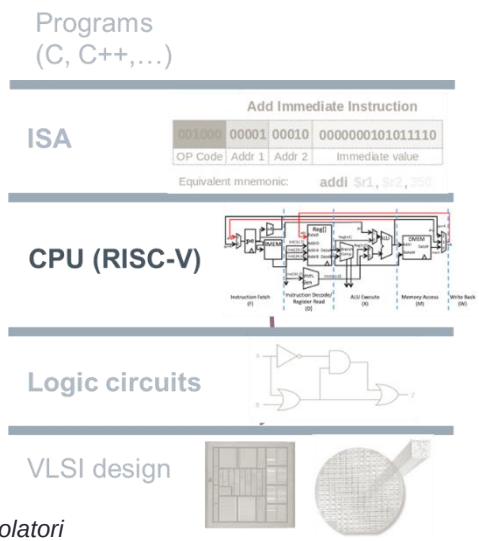

AMD Zen2 core

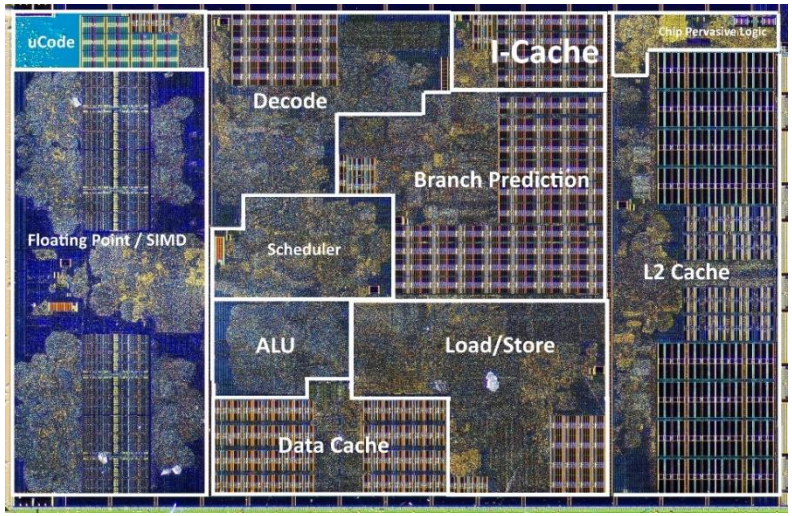
# La CPU: l'implementazione dell'ISA




- **CPU PIPELINE**
  - Costituita da blocchi progettati come **reti logiche**



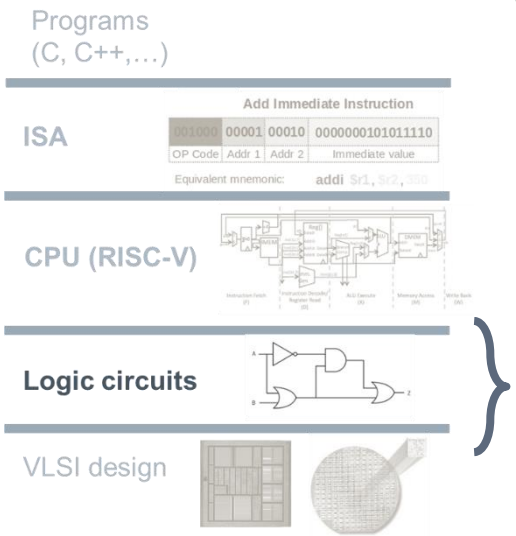




AMD Zen2 core

# Reti logiche

- Le **Reti Logiche** forniscono un'astrazione funzionale di come funziona la tecnologia sottostante (vedi sotto), e sono composte dall'interconnessione di numerose **porte logiche**

- **Porta logica** = dispositivo ideale che realizza una funzione booleana
  - Un'operazione logica eseguita su uno o più input binari che produce un singolo output binario.

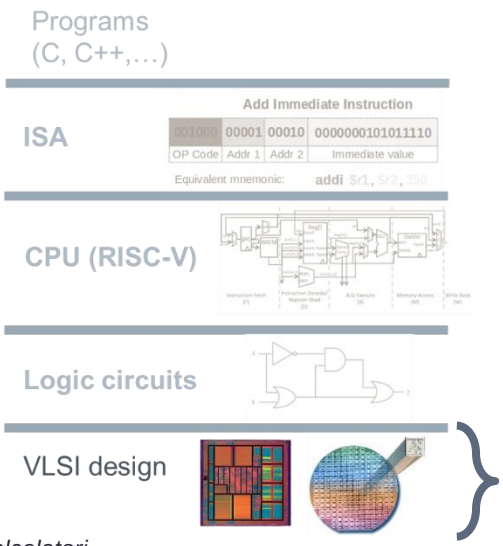



- Le porte logiche sono implementate con ***transistor*** che fungono da *interruttori elettronici*
  - ON/OFF → TRUE/FALSE

# Tecnologia CMOS per le reti logiche

- Le **Reti Logiche** forniscono un'astrazione funzionale di come funziona la tecnologia sottostante (vedi sotto), e sono composte dall'interconnessione di numerose **porte logiche**

- **Porta logica** = dispositivo ideale che realizza una funzione booleana
  - Un'operazione logica eseguita su uno o più input binari che produce un singolo output binario.

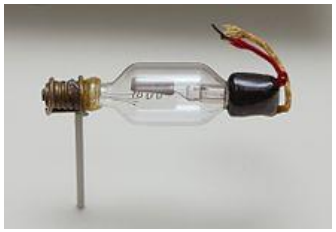



- Le porte logiche sono implementate con ***transistor*** che fungono da *interruttori elettronici*
  - ON/OFF → TRUE/FALSE
  - La maggior parte dei chip esistenti utilizza logica **CMOS**[1] per realizzare i transistor.
    - *Ma si possono realizzare con valvole termoioniche, nanotubi di carbonio, logica fluidica, pneumatica, ottica, molecolare…*

1. *Complementary metal-oxide-semiconductor*

# Technology Trends

- Electronics technology continues to evolve
  - *Increased capacity and performance*
  - *Reduced cost*



The first triode, the De Forest Audion, invented in 1906

The 1946 ENIAC computer used 17,468 vacuum tubes and consumed 150 kW of power

| Year | Technology | Relative performance/cost |
|---|---|---|
| 1951 | Vacuum tube | 1 |
| | | |
| | | |
| | | |
| | | |

# Technology Trends

- Electronics technology continues to evolve
  - *Increased capacity and performance*
  - *Reduced cost*

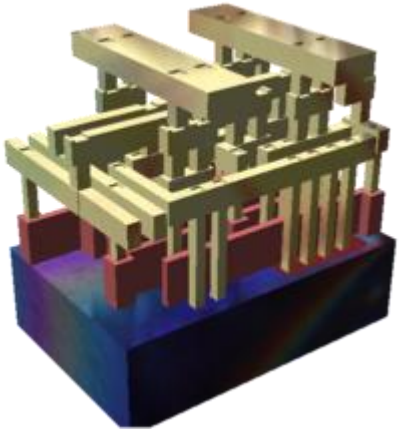

Assorted discrete transistors


The 1955 CADET computer used 324 point-contact transistors

| Year | Technology | Relative performance/cost |
|---|---|---|
| 1951 | Vacuum tube | 1 |
| 1965 | Transistor | 35 |
| | | |
| | | |
| | | |

# Technology Trends

- Electronics technology continues to evolve
  - *Increased capacity and performance*
  - *Reduced cost*



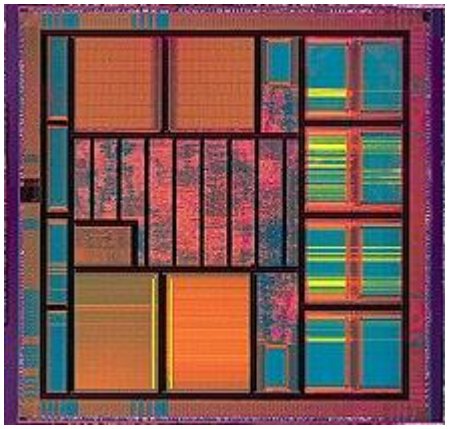
Electronic circuits are gradually created on a wafer made of pure semiconducting material. Silicon is almost always used

Robert Noyce invented the first monolithic IC chip in 1959. It was made from silicon and was fabricated using Hoerni's planar process and Atalla's surface passivation process.

| Year | Technology | Relative performance/cost |
|---|---|---|
| 1951 | Vacuum tube | 1 |
| 1965 | Transistor | 35 |
| 1975 | Integrated circuit (IC) | 900 |
| | | |
| | | |

# Technology Trends

- Electronics technology continues to evolve
  - *Increased capacity and performance*
  - *Reduced cost*



A VLSI integrated-circuit die

**VLSI** is the process of creating an integrated circuit (IC) by combining millions of MOS transistors onto a single chip

| Year | Technology | Relative performance/cost |
|---|---|---|
| 1951 | Vacuum tube | 1 |
| 1965 | Transistor | 35 |
| 1975 | Integrated circuit (IC) | 900 |
| 1995 | Very large scale IC (VLSI) | 2,400,000 |
| | | |

# Technology Trends

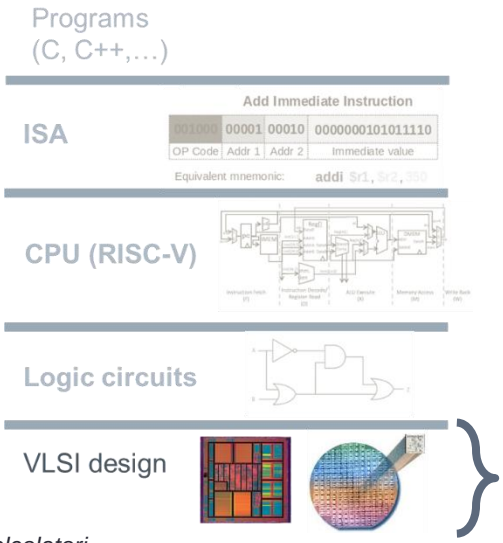
- Electronics technology continues to evolve
  - *Increased capacity and performance*
  - *Reduced cost*

| Name | Signification | Year | Transistors | Logic gates |
|---|---|---|---|---|
| SSI | *small-scale integration* | 1964 | 1 to 10 | 1 to 12 |
| MSI | *medium-scale integration* | 1968 | 10 to 500 | 13 to 99 |
| LSI | *large-scale integration* | 1971 | 500 to 20.000 | 100 to 9999 |
| VLSI | *very large-scale integration* | 1980 | 20.000 to 1.000.000 | 10.000 to 99.999 |
| ULSI | *ultra-large-scale integration* | 1984 | > 1.000.000 | >= 100.000 |

| Year | Technology | Relative performance/cost |
|---|---|---|
| 1951 | Vacuum tube | 1 |
| 1965 | Transistor | 35 |
| 1975 | Integrated circuit (IC) | 900 |
| 1995 | Very large scale IC (VLSI) | 2,400,000 |
| 2013 | Ultra large scale IC | 250,000,000,000 |

# Semiconductor Technology

- **Silicon**: semiconductor
  - A **semiconductor** material has an *electrical conductivity* value falling between that of a **conductor**, such as metallic copper, and an **insulator**, such as glass.
  - Its conducting properties may be altered in useful ways by introducing impurities ("**doping**") into the crystal structure



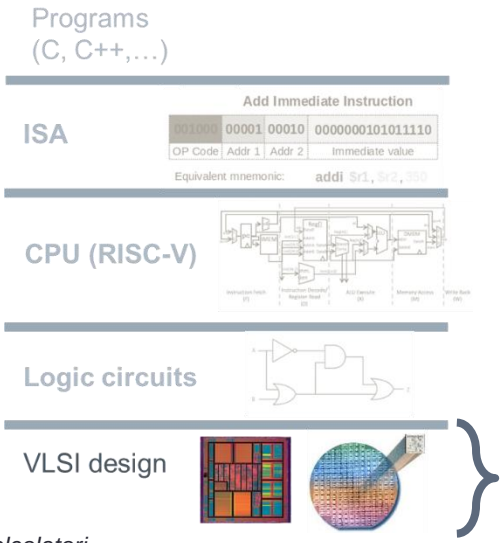

Atom size is 20-200 pm

Semiconductor Manufacturing Process

| | |
|---|---|
| 10 µm | – 1971 |
| 6 µm | – 1974 |
| 3 µm | – 1977 |
| 1.5 µm | – 1982 |
| 1 µm | – 1985 |
| 800 nm | – 1989 |
| 600 nm | – 1994 |
| 350 nm | – 1995 |
| 250 nm | – 1997 |
| 180 nm | – 1999 |
| 130 nm | – 2001 |
| 90 nm | – 2004 |
| 65 nm | – 2006 |
| 45 nm | – 2008 |
| 32 nm | – 2010 |
| 22 nm | – 2012 |
| 14 nm | – 2014 |
| 10 nm | – 2017 |
| 7 nm | – *~2019* |
| 5 nm | – *~2021* |

# Semiconductor Technology

- **Silicon**: semiconductor
- Add materials to transform properties:
  - ***Conductors***
    - *microscopic copper or aluminum wire*
  - ***Insulators***
    - *plastic sheathing or glass*




Atom size is 20-200 pm

Semiconductor Manufacturing Process

| | |
|---|---|
| 10 µm | – 1971 |
| 6 µm | – 1974 |
| 3 µm | – 1977 |
| 1.5 µm | – 1982 |
| 1 µm | – 1985 |
| 800 nm | – 1989 |
| 600 nm | – 1994 |
| 350 nm | – 1995 |
| 250 nm | – 1997 |
| 180 nm | – 1999 |
| 130 nm | – 2001 |
| 90 nm | – 2004 |
| 65 nm | – 2006 |
| 45 nm | – 2008 |
| 32 nm | – 2010 |
| 22 nm | – 2012 |
| 14 nm | – 2014 |
| 10 nm | – 2017 |
| 7 nm | – *~2019* |
| 5 nm | – *~2021* |

# Manifacturing ICs

**Remember this guy?**

**Remember this guy?**

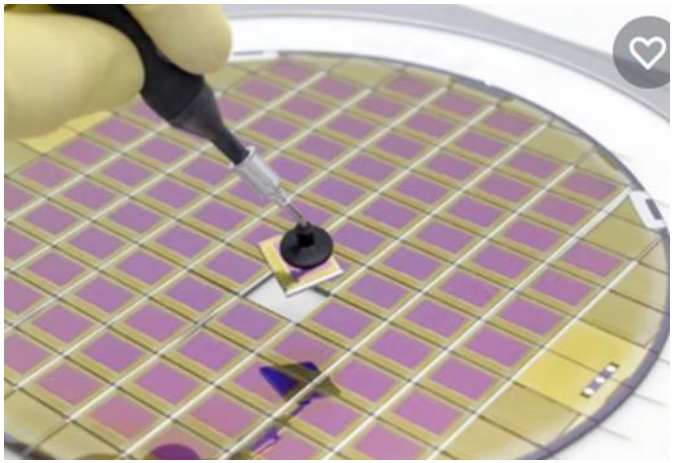
# Manifacturing ICs


From a silicon ingot (99.999999999% pure)…

**Remember this guy?**

# Manifacturing ICs


From a silicon ingot
(99.999999999% pure)…

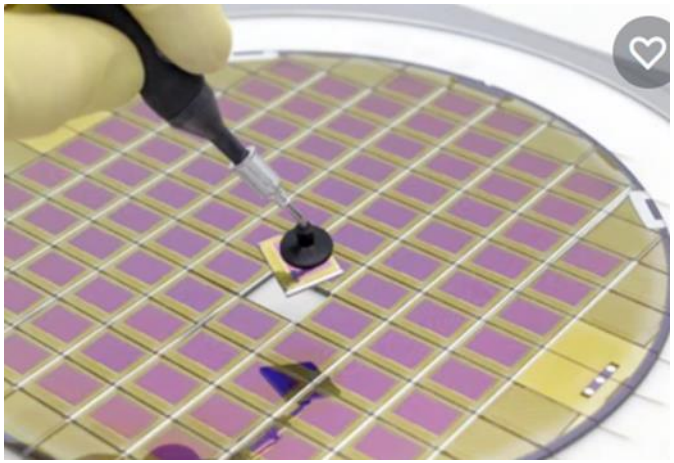


…to a silicon wafer with Ics…

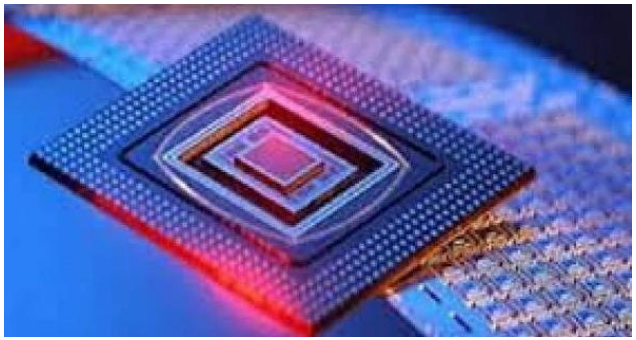
**Remember this guy?**

# Manifacturing ICs

From a silicon ingot
(99.999999999% pure)…

…to a silicon wafer with Ics…

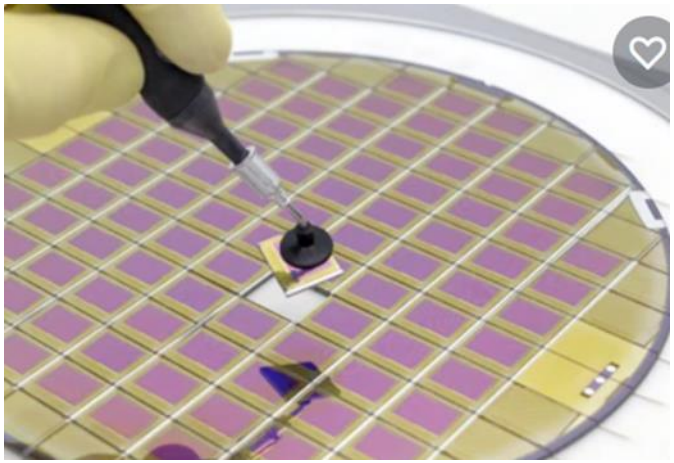


…ready for packaging

**Remember this guy?**

# Manifacturing ICs

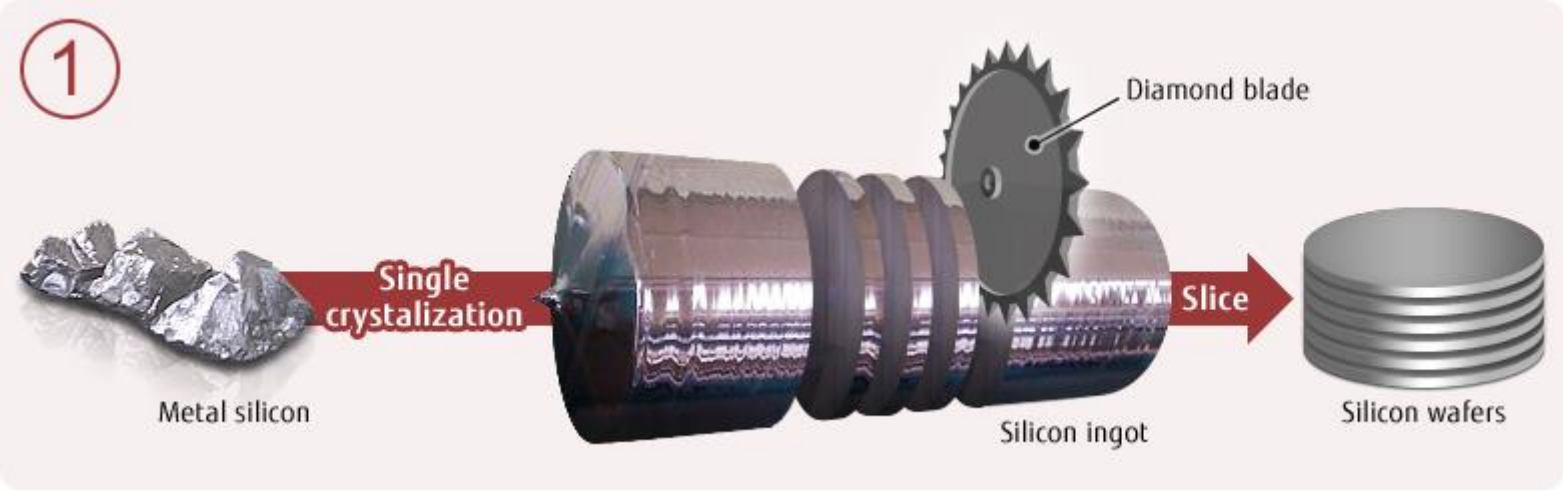

From a silicon ingot
(99.999999999% pure)…



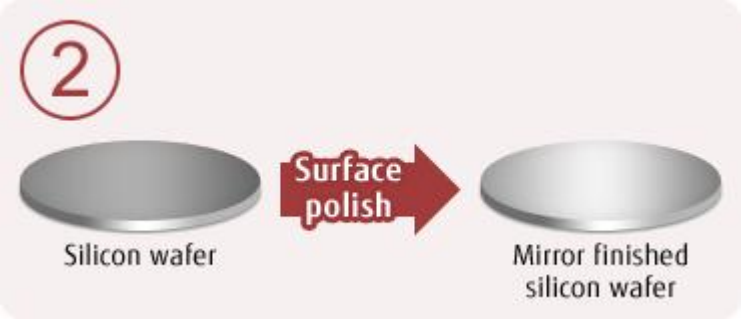
**HOW DO WE GET THERE?**

…to a silicon wafer with Ics…

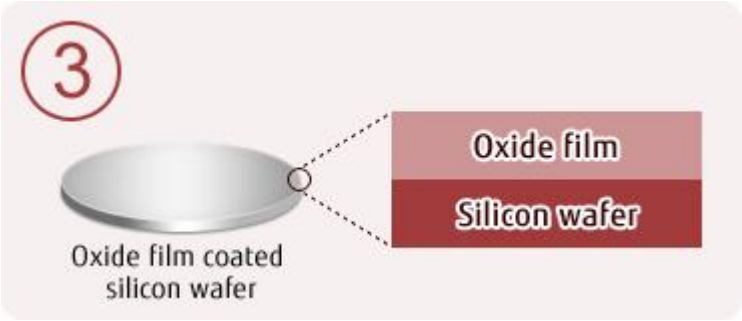
# Manifacturing ICs




**The silicon ingot is sliced into thin silicon wafers.**



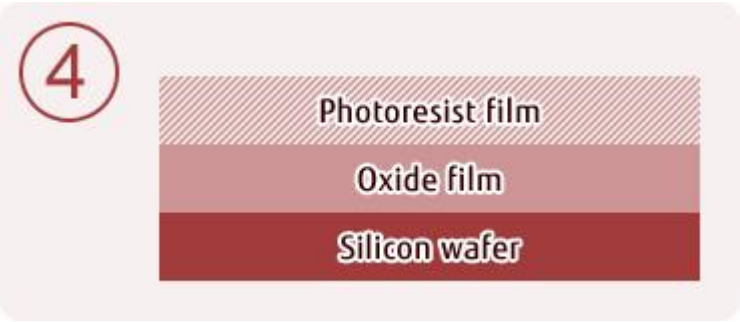

**The surface of the silicon wafer is polished to create a mirror finish.**



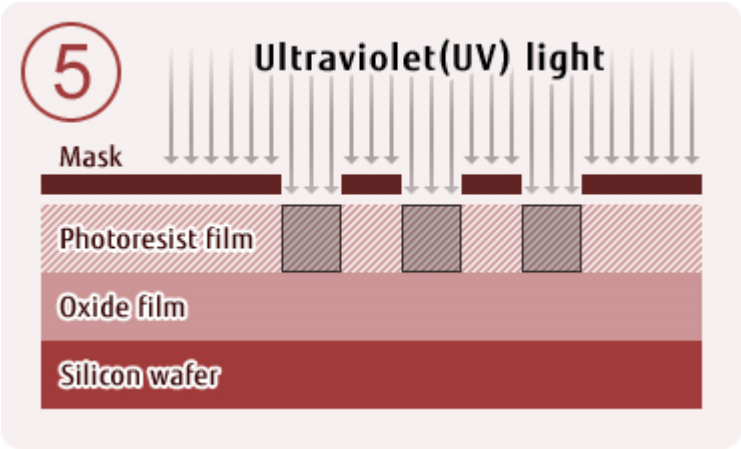

**An oxide film is grown onto the wafer.**

*https://www.fujitsu.com/ph/about/businesspolicy/tech/k/whatis/processor/cpu.html*

# Manifacturing ICs



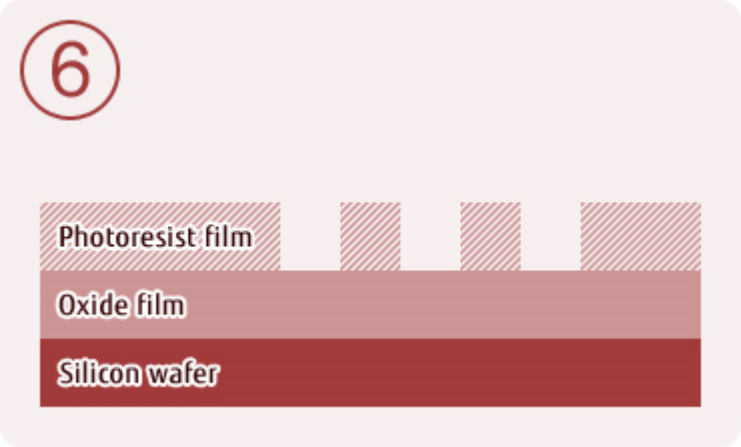

**A photoresist film is coated on the wafer surface.**




**The photoresist film is exposed to ultraviolet (UV) light through the pattern on the mask.**

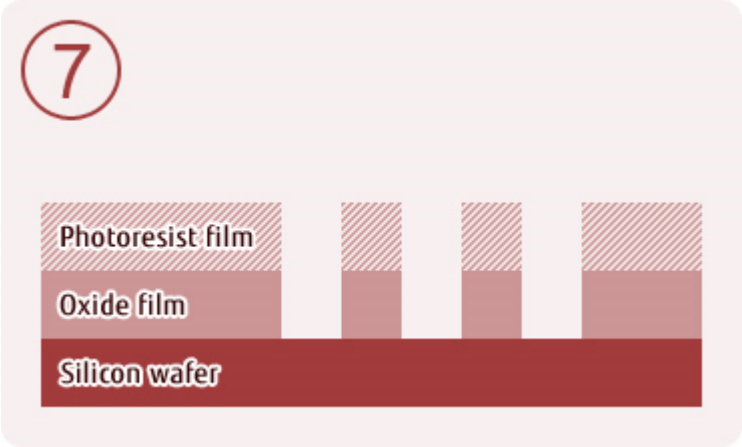
*https://www.fujitsu.com/ph/about/businesspolicy/tech/k/whatis/processor/cpu.html*

# Manifacturing ICs



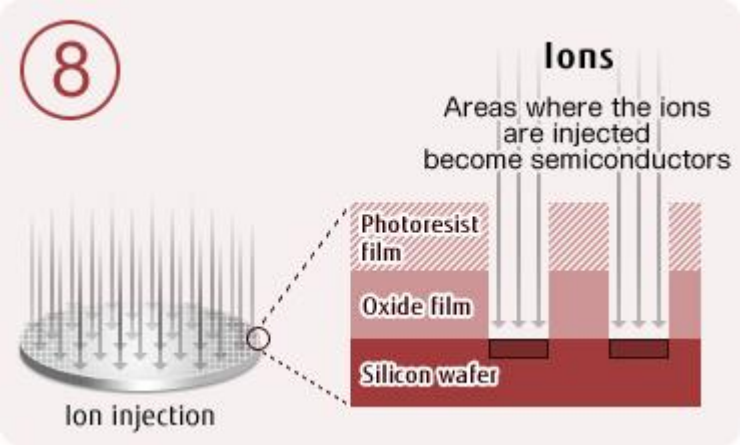

**The photoresist region that was exposed to the UV light is removed using a developer. (The exposed area changes to a substance that is dissolved by the developer.)**

**The oxide film is then removed using a caustic agent. This exposes the silicon surface.**



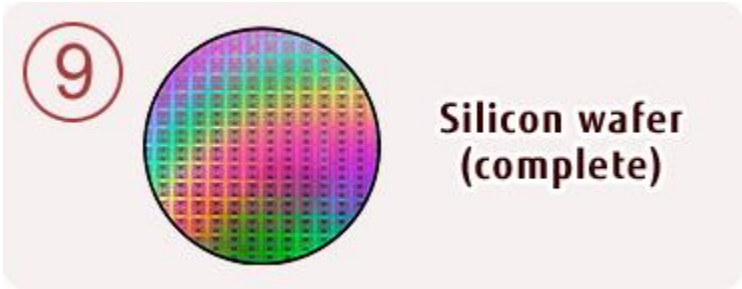

*https://www.fujitsu.com/ph/about/businesspolicy/tech/k/whatis/processor/cpu.html*

# Manifacturing ICs



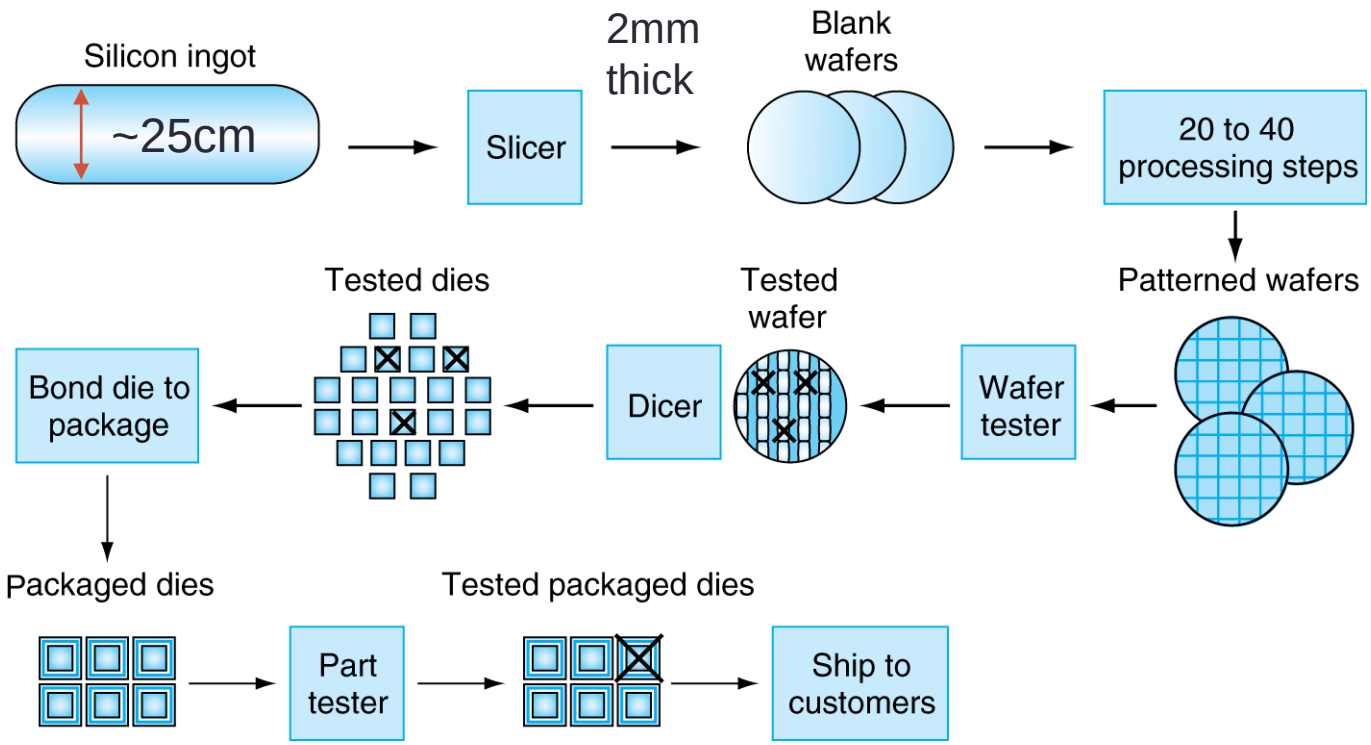

**The required ions are then injected into the silicon surface. The character of the silicon then changes into a semiconductor, which is a state where elements with electrical characteristics can be created.**

**Repeat steps 4 to 8 many times…**



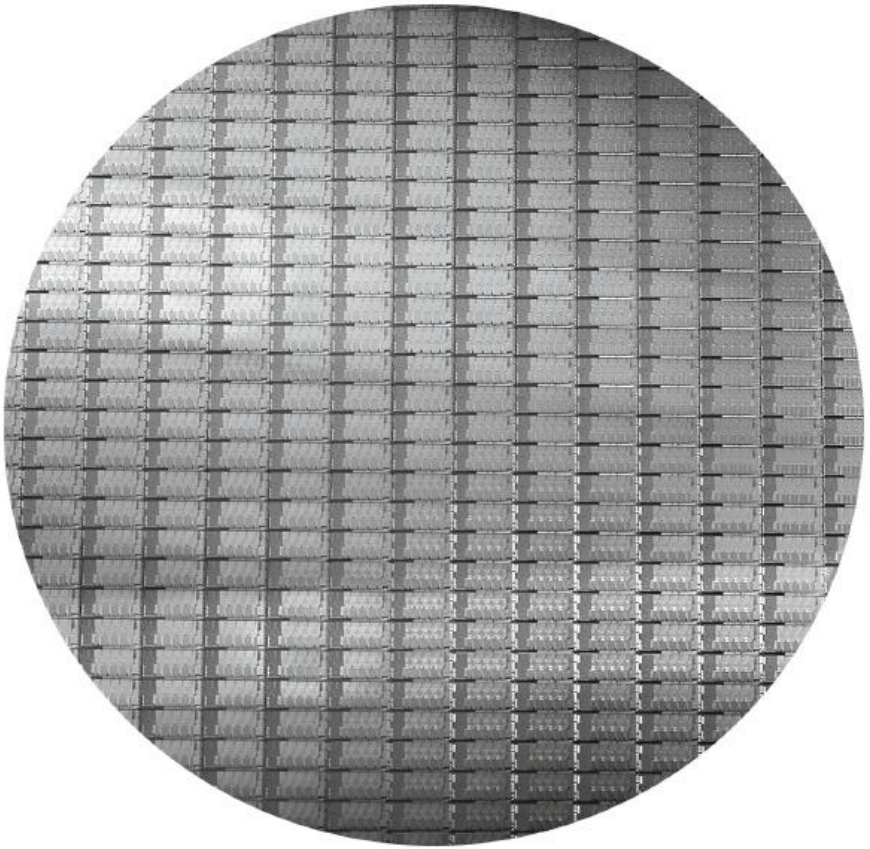

*https://www.fujitsu.com/ph/about/businesspolicy/tech/k/whatis/processor/cpu.html*

# Manufacturing ICs




- One layer of transistors and 2-8 levels of metal conductor, separated by layers of insulators
- **Yield**: proportion of working dies per wafer

# Intel Core i7 Wafer



- 300mm wafer, 280 chips, 32nm technology
- Each chip is 20.7 x 10.5 mm

# Integrated Circuit Cost

$$\text{Cost per die} = \frac{\text{Cost per wafer}}{\text{Dies per wafer} \times \text{Yield}}$$

$$\text{Dies per wafer} \approx \text{Wafer area} / \text{Die area}$$

$$\text{Yield} = \frac{1}{(1 + (\text{Defects per area} \times \text{Die area}/2))^2}$$

- The cost of an integrated circuit rises quickly as the die size increases
  - lower yield and fewer dies that fit on a wafer.
  - nonlinear relation to area and defect rate
    - Wafer cost and area are fixed
    - Defect rate determined by manufacturing process
    - Die area determined by architecture and circuit design