CSL7070: Computer Architecture

Lecture 01, 5th January 2022

Dip Sankar Banerjee

Indian Institute of Technology, Jodhpur
January-April 2022

Welcome

- To the path of understanding the engineering in Computer Science Engineering.
- To the art of hardware design.
- To the art of abstractions.
- To the world of systems.

What you will learn in CSL7070

- What's under the hood of a computer?
- How does it work?
- What are the logical components that implement the computer?
- How does software execute on a computer?
- What is the interface between the hardware of the computer and the software that runs on it?
- Something everyone who uses as computer should know!

Pervasive Idea

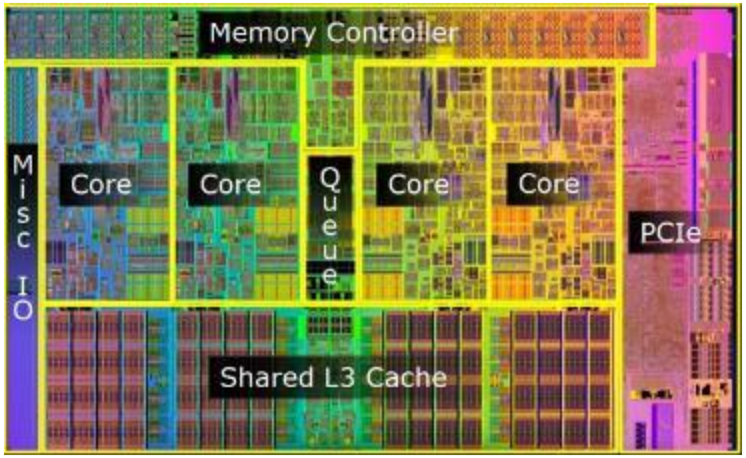
- **Abstraction:** only way of dealing with complex systems
  - Divide world into objects, each with an…
    - Interface: knobs, behaviors, knobs → behaviors
    - Implementation: “black box” (ignorance+apathy)
  - Only specialists deal with implementation, rest of us with interface
  - Example: car, only mechanics know how implementation works

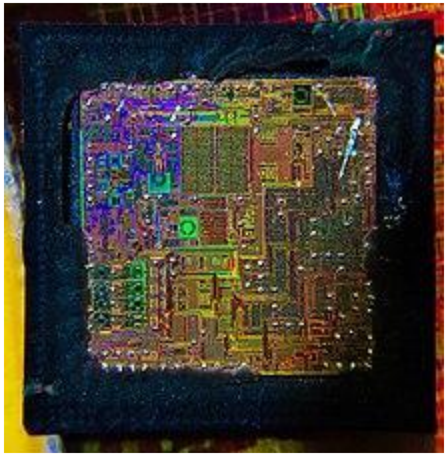
- **Layering:** abstraction discipline makes life even simpler
  - Divide objects in system into layers, layer n objects…
    - Implemented using interfaces of layer n – 1
    - Don’t need to know interfaces of layer n – 2 (sometimes helps)

Pervasive Idea

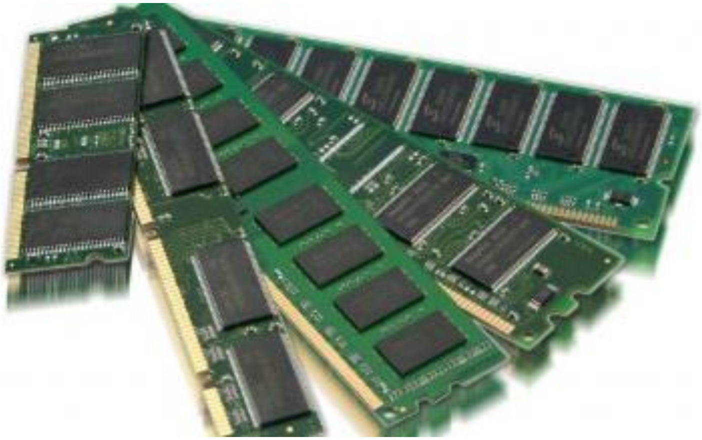
- **Inertia:** a dark side of layering
  - Layer interfaces become entrenched over time ("standards")
  - Very difficult to change even if benefit is clear (example: Digital TV)
- **Opacity:** hard to reason about performance across layers

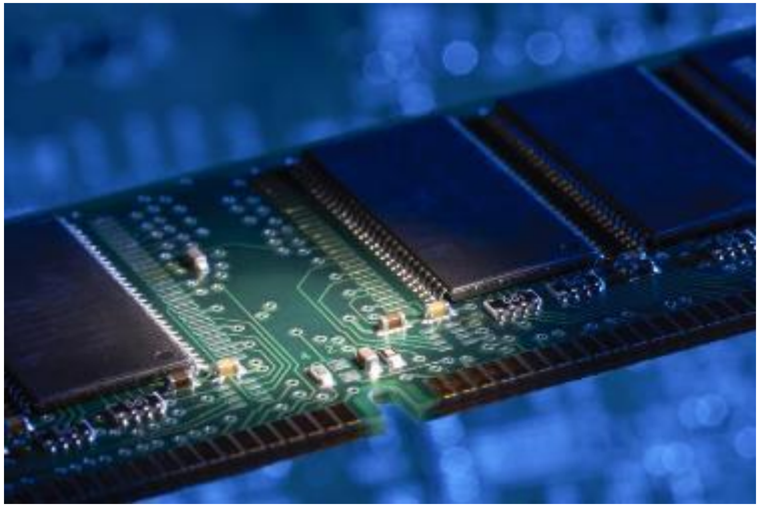
What is a Computer ? CPU

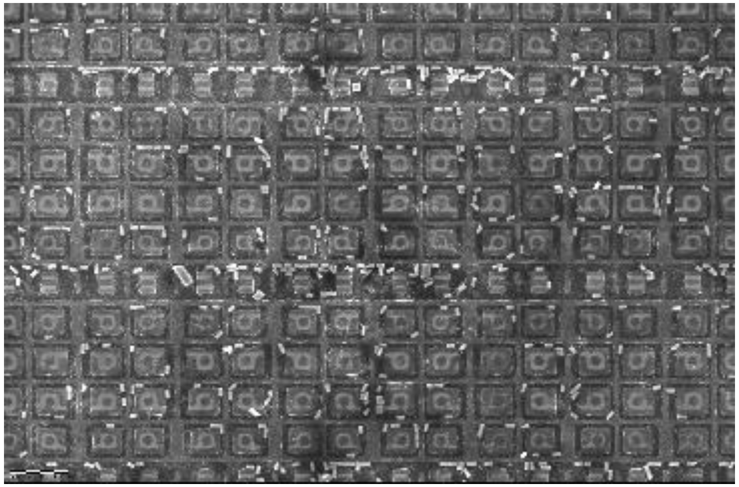


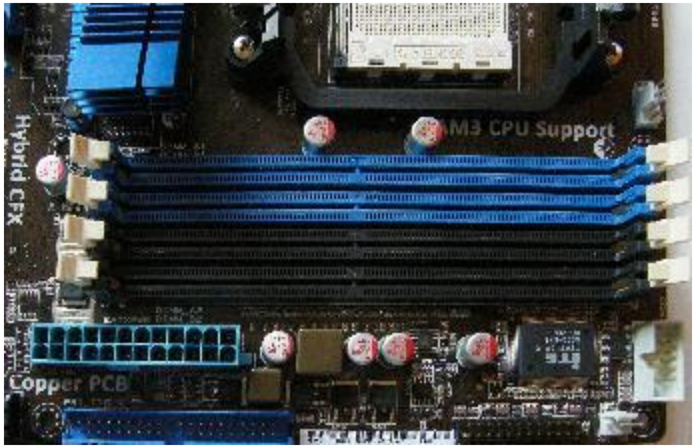


What is a Computer ? Main Memory



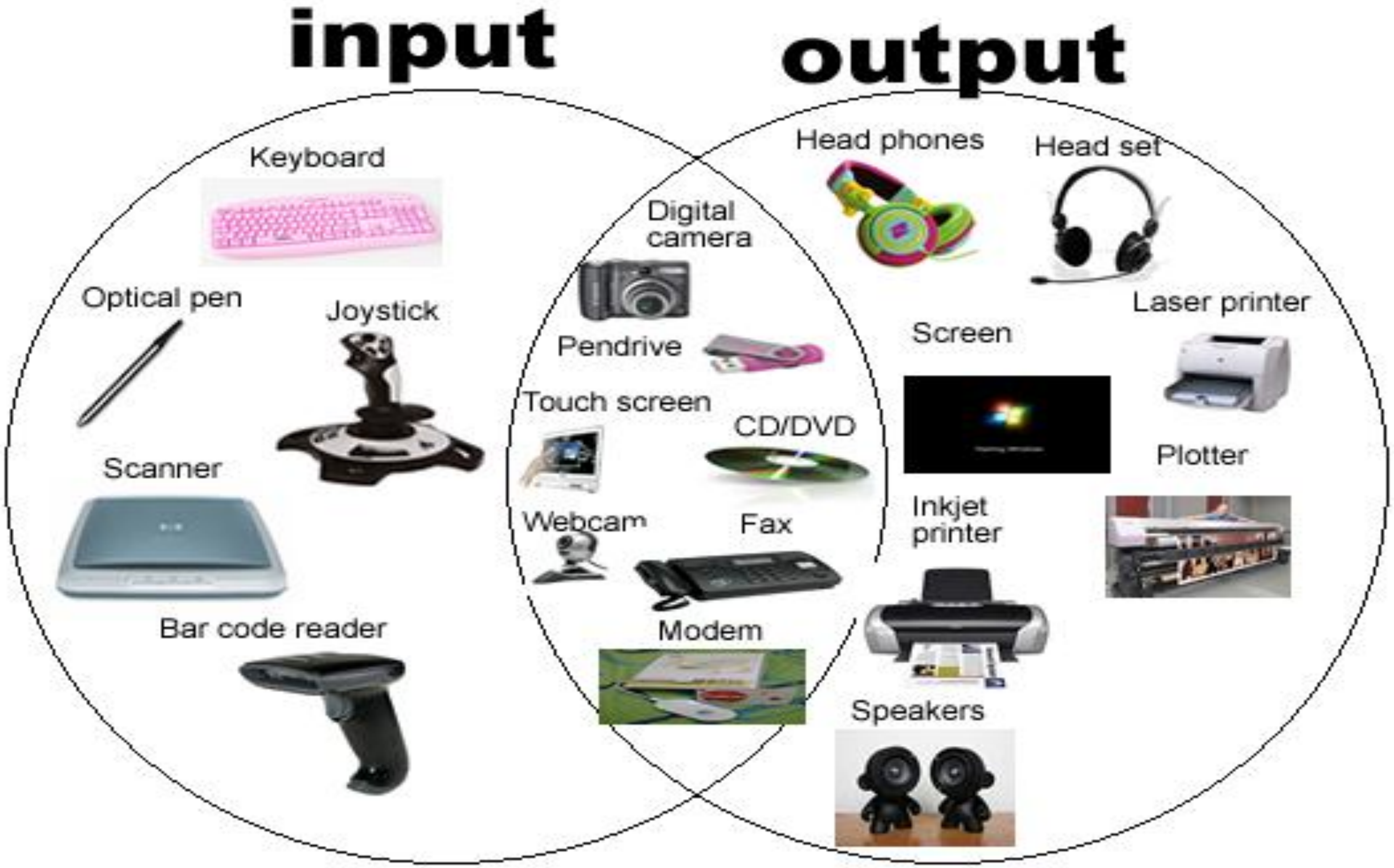




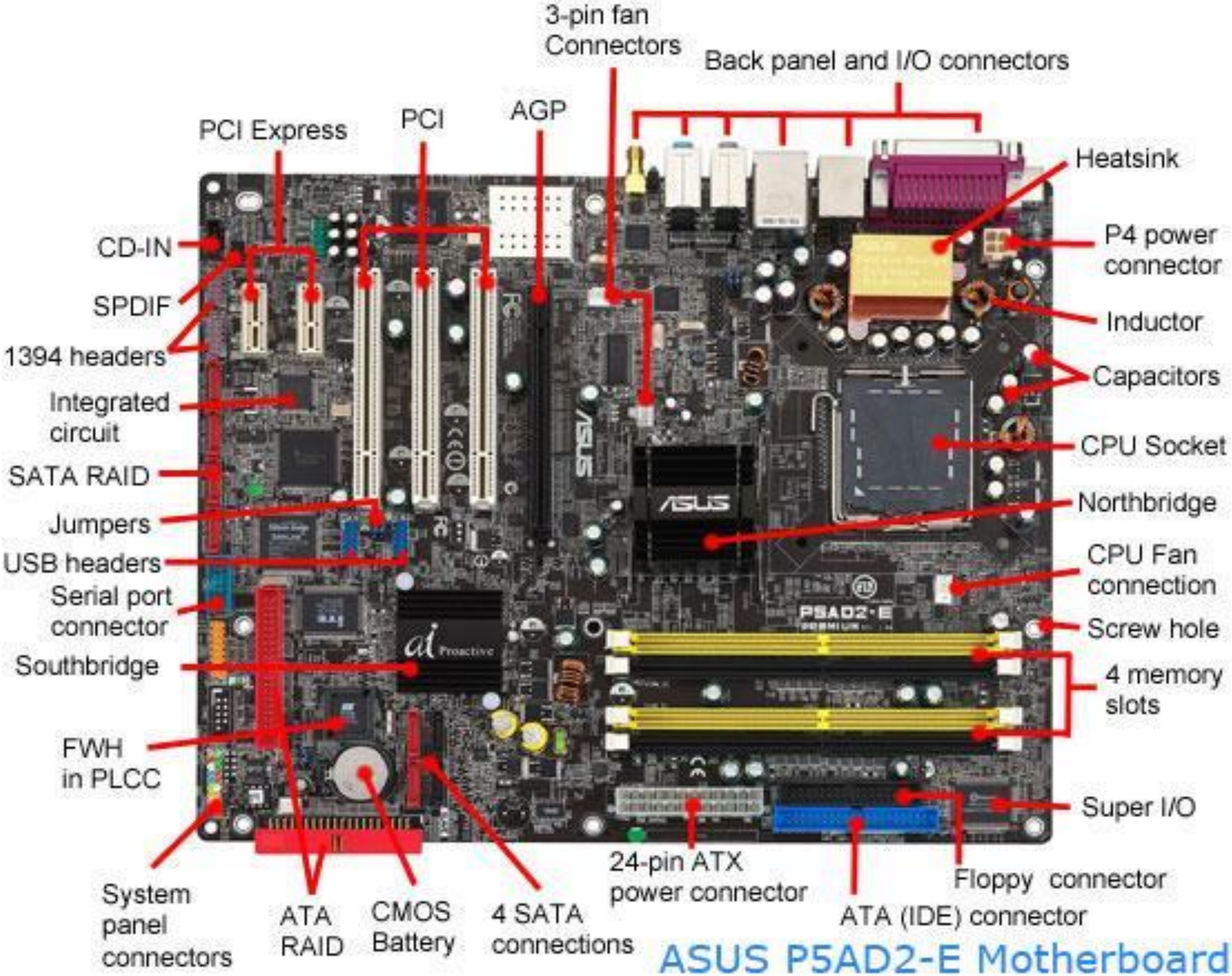


What is a Computer ? Secondary Mem

What is a Computer ? I/O



What is a Computer ? Motherboard



ASUS P5AD2-E Motherboard

ComputerHope.com

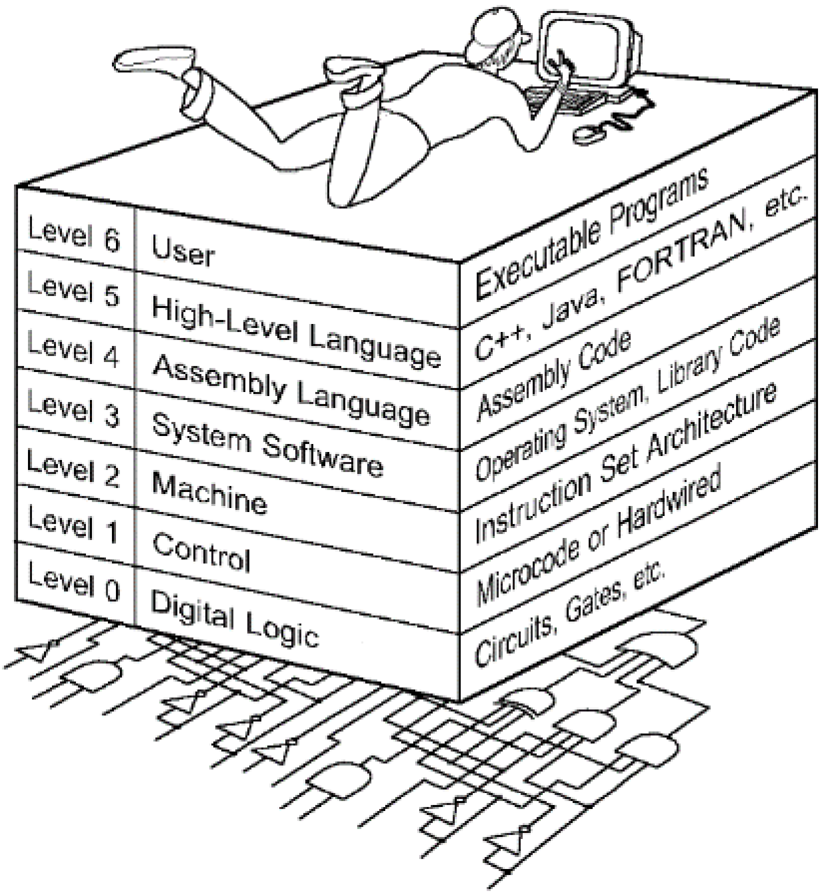
What is a Computer ?

- Components:
  - input (mouse, keyboard, camera, microphone...)
  - output (display, printer, speakers....)
  - memory (caches, DRAM, SRAM, hard disk drives, Flash....)
  - network (both input and output)
- Our primary focus: the processor (datapath and control)
  - implemented using billions of transistors
  - Impossible to understand by looking at each transistor
  - We need...abstraction!

An abstraction omits unnecessary detail, helps us cope with complexity.

How do Computers Work ?

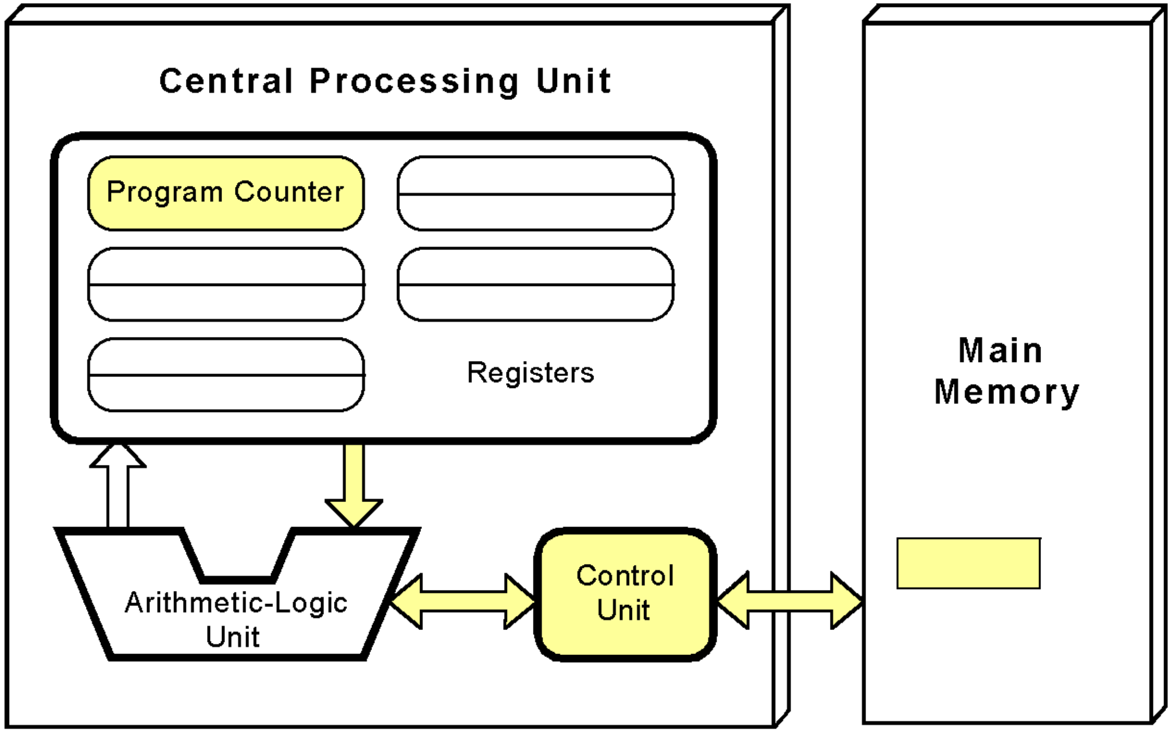
- **Each of the following abstracts everything below it:**
  - Applications software
  - Systems software
  - Assembly Language
  - Machine Language
  - Architectural Approaches: Caches, Virtual Memory, Pipelining
  - Sequential logic, finite state machines
  - Combinational logic, arithmetic circuits
  - Boolean logic, 1s and 0s
  - Transistors used to build logic gates (e.g. CMOS)
  - Semiconductors/Silicon used to build transistors
  - Properties of atoms, electrons, and quantum dynamics



- *Notice how abstraction hides the detail of lower levels, yet gives a useful view for a given purpose*

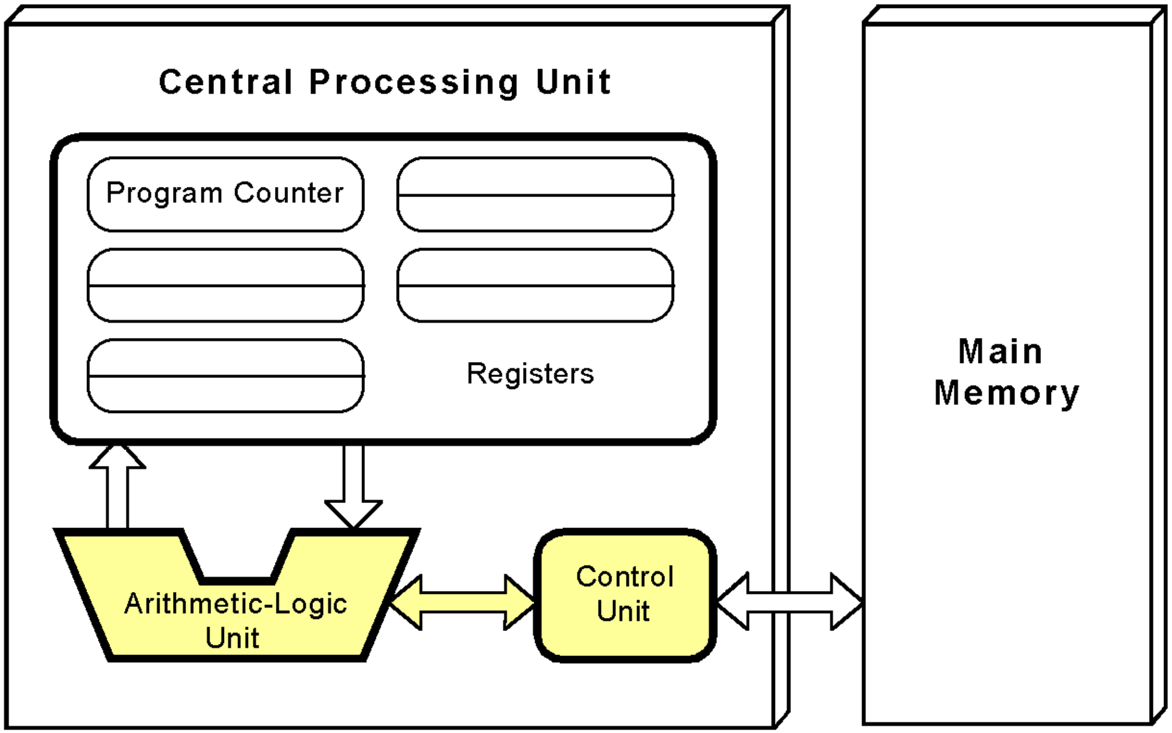
The von Neumann Model

- The control unit fetches the next instruction from memory using the program counter to determine where the instruction is located



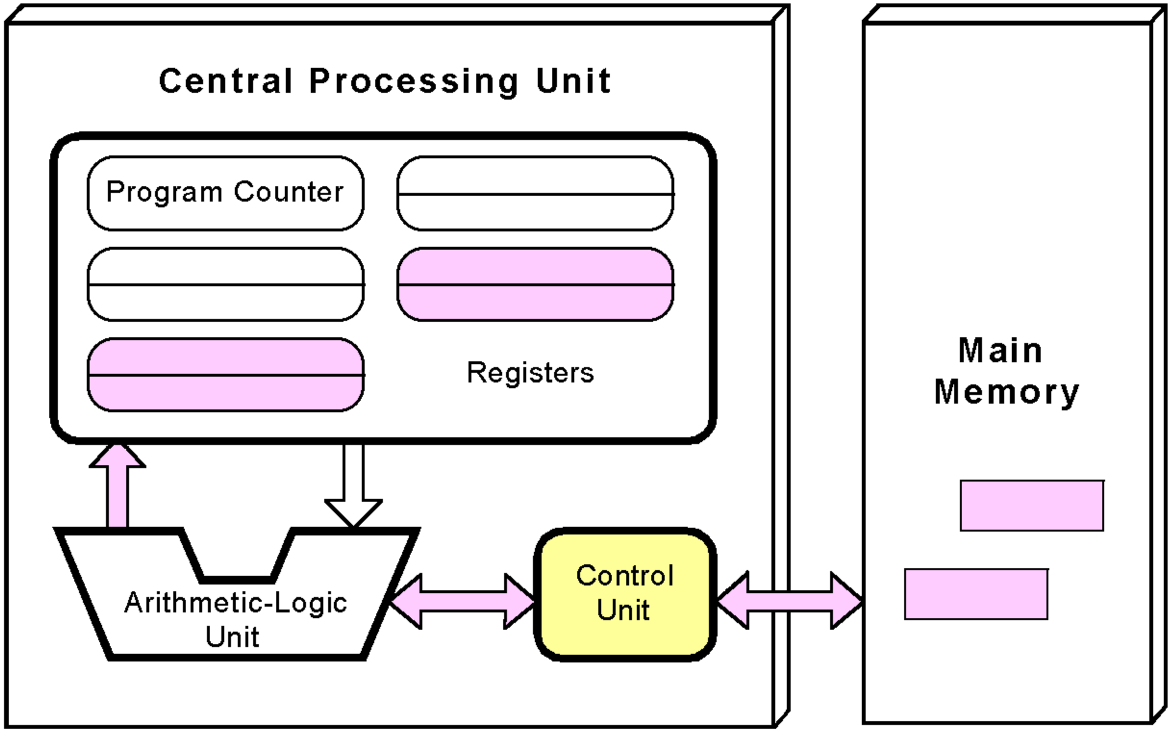
The von Neumann Model

- The instruction is decoded into a language that the ALU can understand.



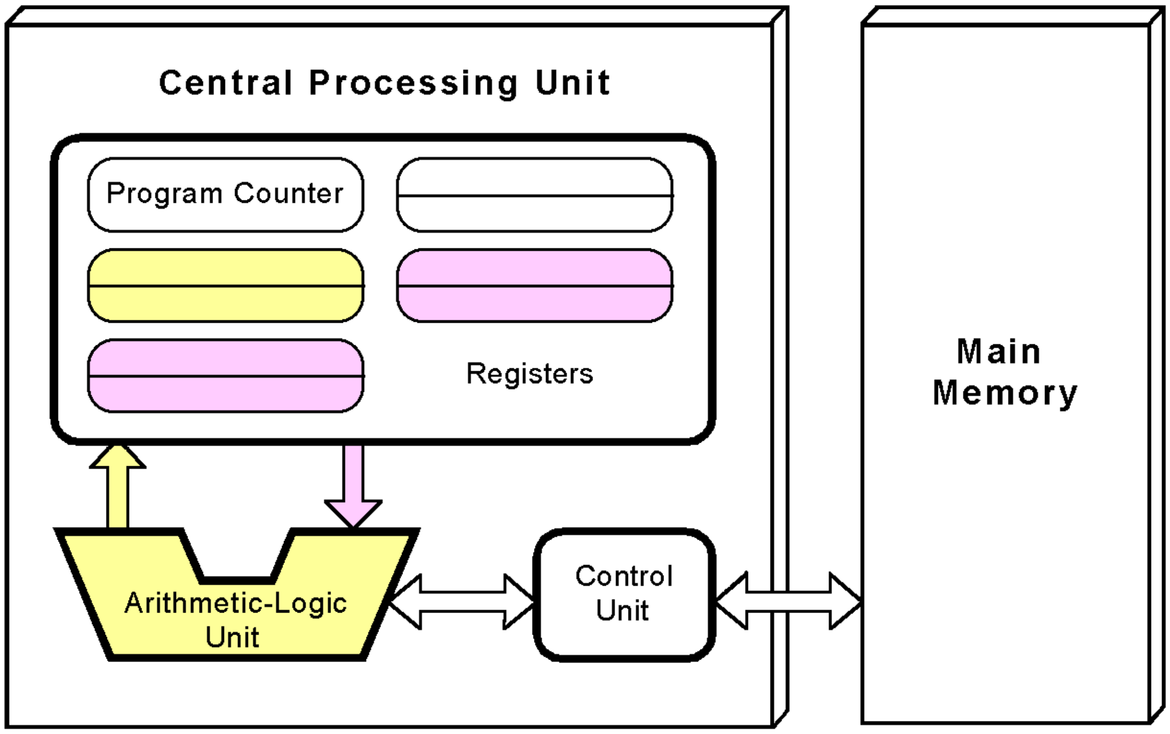
The von Neumann Model

- Any data operands required to execute the instruction are fetched from memory and placed into registers within the CPU.

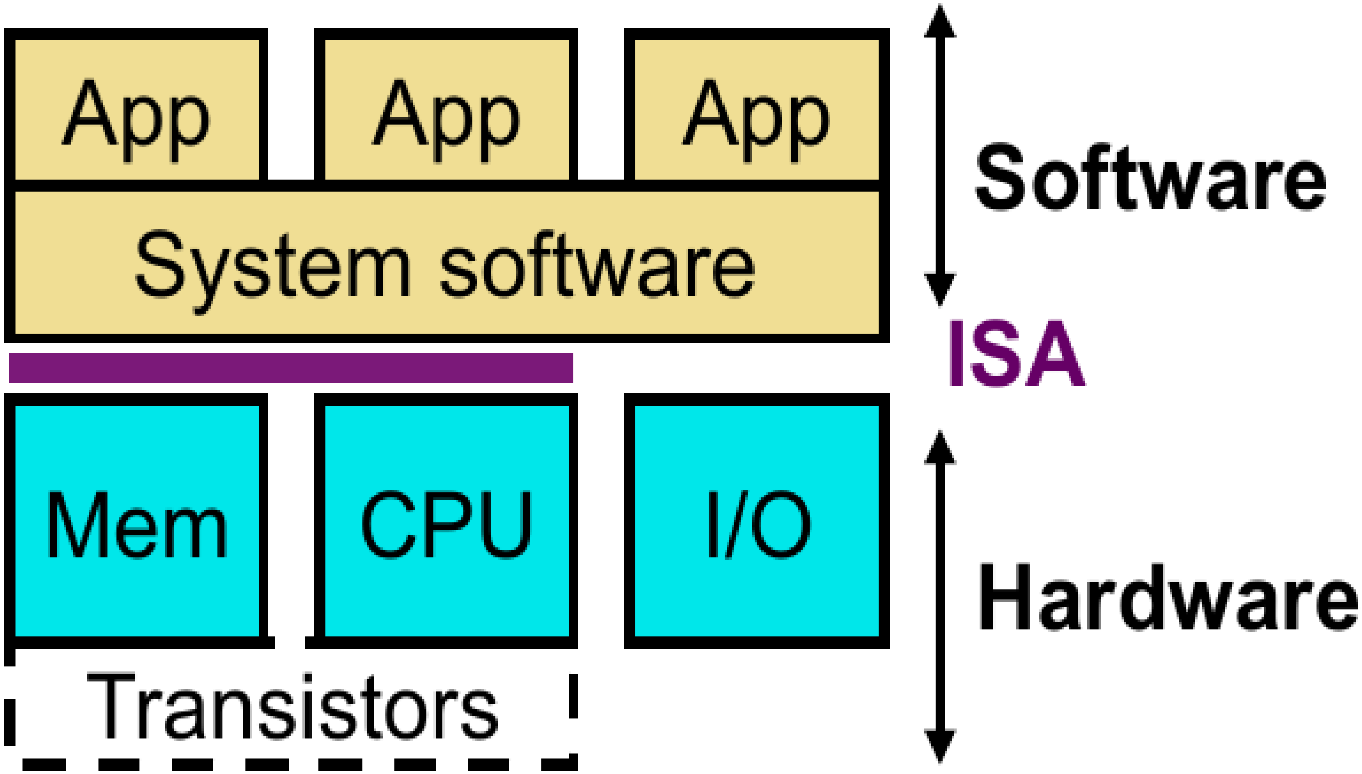


The von Neumann Model

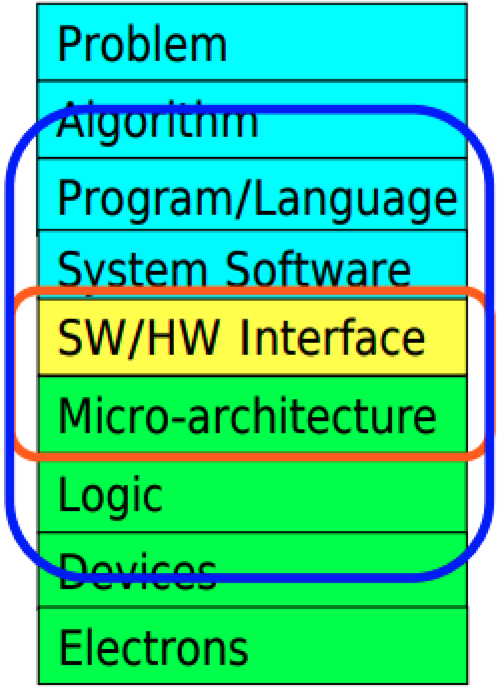
- The ALU executes the instruction and places results in registers or memory.



Abstraction, Layering and Computers



Computer Architecture



Computer Architecture (expanded view)

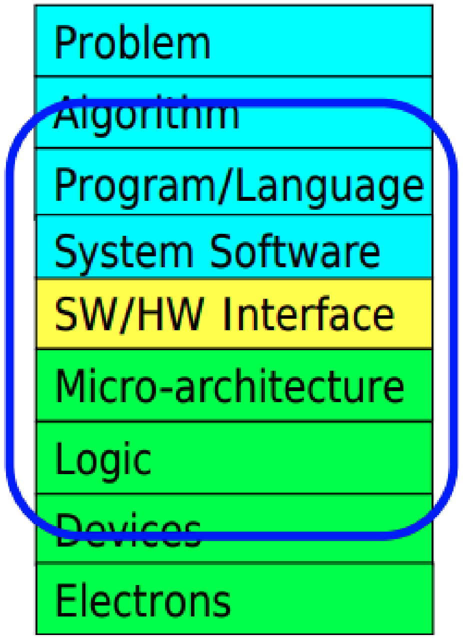
Computer Architecture (narrow view)

Axiom

To achieve the highest energy efficiency and performance:

we must take the expanded view

of computer architecture



Co-design across the hierarchy: Algorithms to devices

Specialize as much as possible within the design goals

Different Platforms, Different Goals

Different Platforms, Different Goals

Different Platforms, Different Goals

Different Platforms, Different Goals

Different Platforms, Different Goals

Different Platforms, Different Goals

Different Platforms, Different Goals

Figure 3. TPU Printed Circuit Board. It can be inserted in the slot for an SATA disk in a server, but the card uses PCIe Gen3 x16.

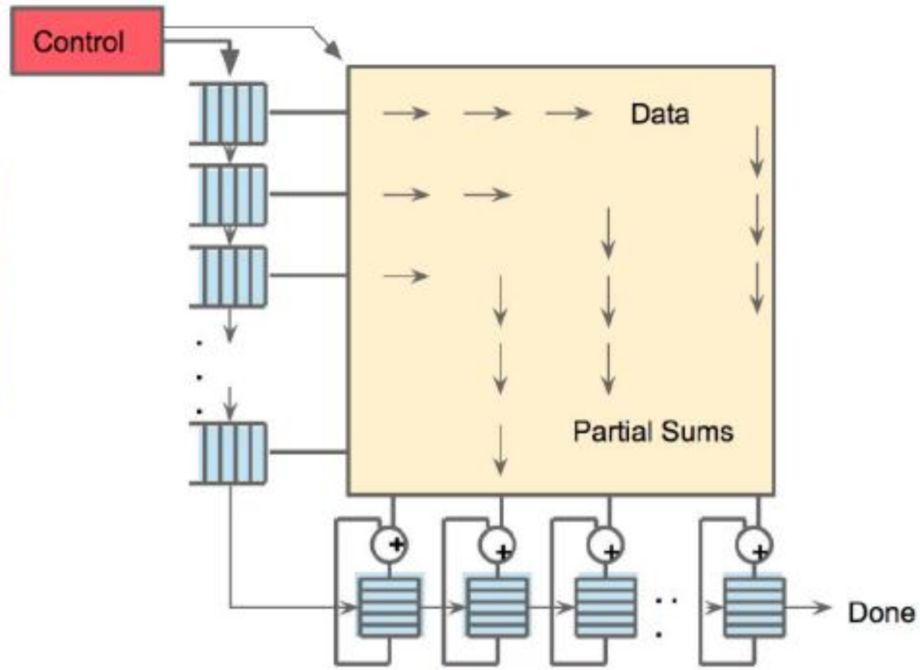
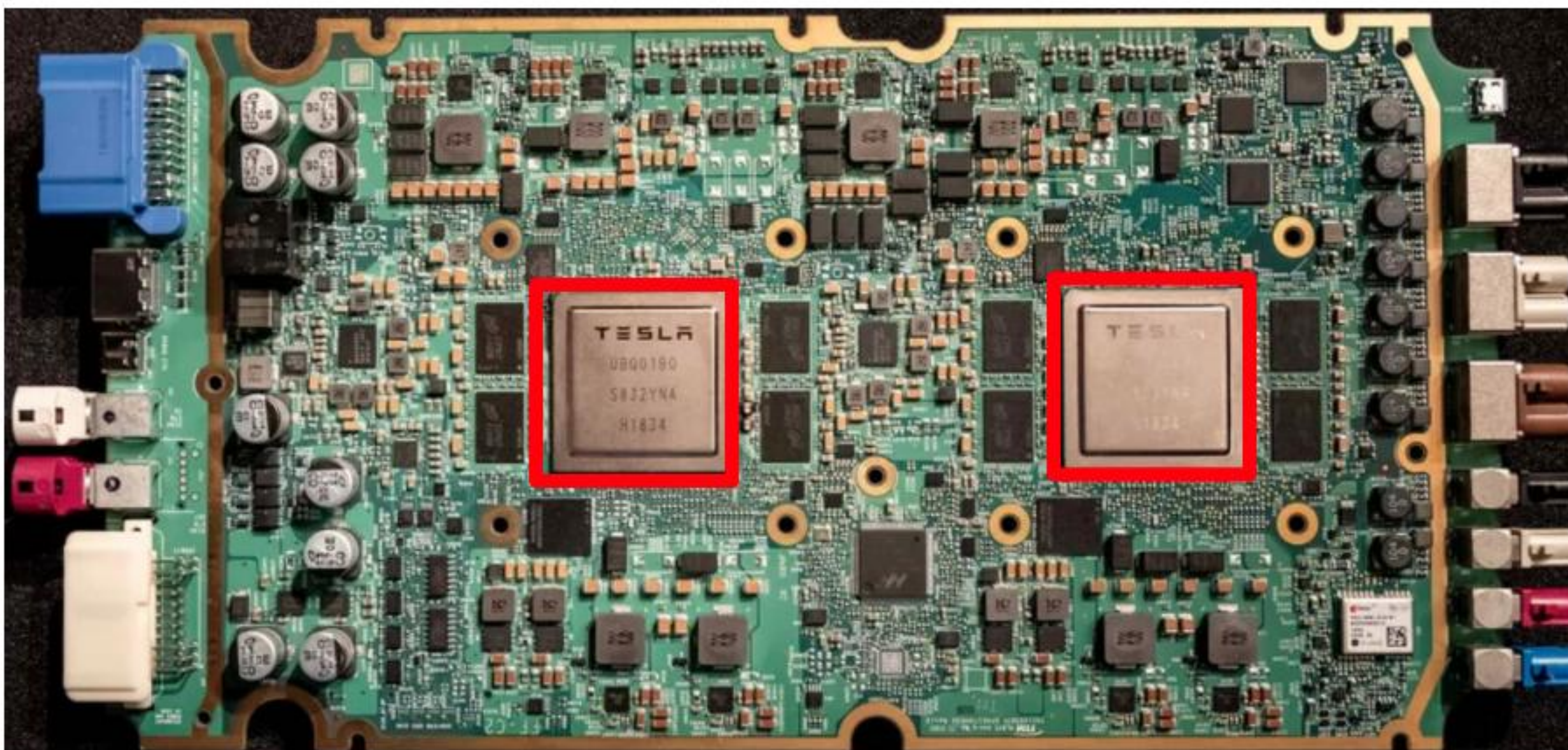


Figure 4. Systolic data flow of the Matrix Multiply Unit. Software has the illusion that each 256B input is read at once, and they instantly update one location of each of 256 accumulator RAMs.

Jouppi et al., "In-Datacenter Performance Analysis of a Tensor Processing Unit", ISCA 2017.

Different Platforms, Different Goals

- ML accelerator: 260 mm^2, 6 billion transistors, 600 GFLOPS GPU, 12 ARM 2.2 GHz CPUs.
- Two redundant chips for better safety.



What is Computer Architecture?

- The science and art of designing, selecting, and interconnecting hardware components and designing the hardware/software interface to create a computing system that meets functional, performance, energy consumption, cost, and other specific goals.

What is Computer Architecture?

- The science and art of designing, selecting, and interconnecting hardware components and designing the hardware/software interface to create a computing system that meets functional, performance, energy consumption, cost, and other specific goals.

Why study Computer Architecture?

- Enable better systems: make computers faster, cheaper, smaller, more reliable, …
  - By exploiting advances and changes in underlying technology/circuits

- Enable new applications
  - Life-like 3D visualization 20 years ago? Virtual reality?
  - Self-driving cars?
  - Personalized genomics? Personalized medicine?

- Enable better solutions to problems
  - Software innovation is built on trends and changes in computer architecture
    - > 50% performance improvement per year has enabled this innovation

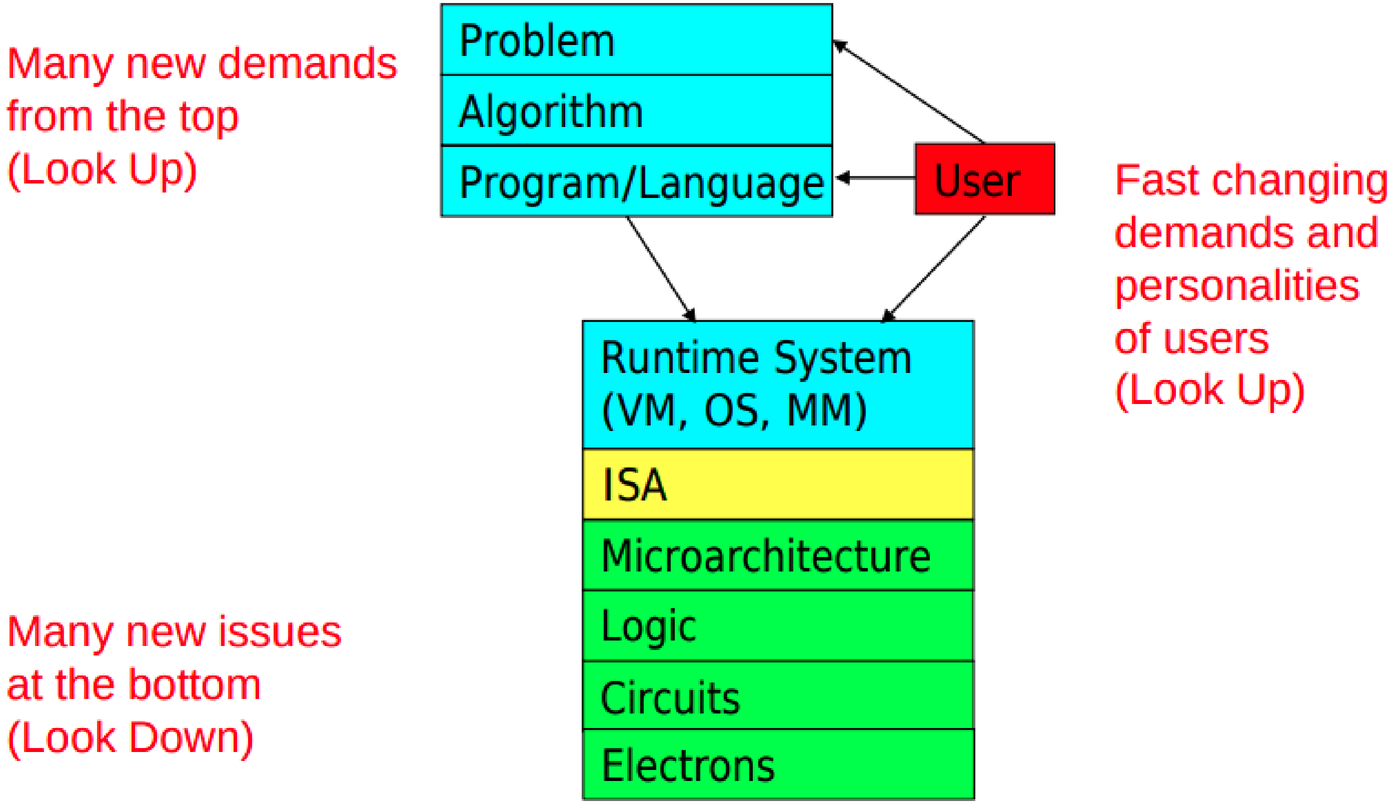
- Understand why computers work the way they do

Computer Architecture Today

- Today is a very exciting time to study computer architecture
- Industry is in a large paradigm shift (to novel architectures) – many different potential system designs possible
- Many difficult problems *motivating* and *caused by* the shift
  - Huge hunger for data and new data-intensive applications
  - Power/energy/thermal constraints
  - Complexity of design
  - Difficulties in technology scaling
  - Memory bottleneck
  - Reliability problems
  - Programmability problems
  - Security and privacy issues
- No clear, definitive answers to these problems

Computer Architecture Today

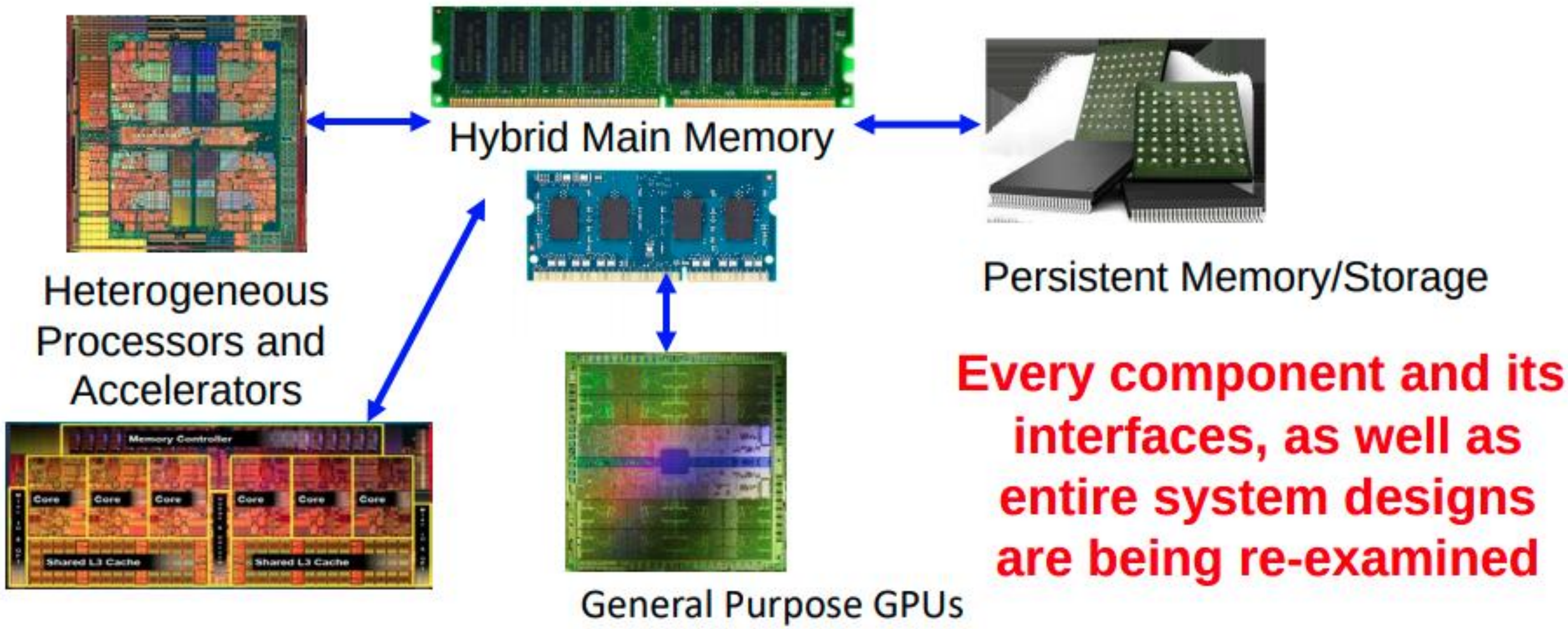
- These problems affect all parts of the computing stack – if we do not change the way we design systems



- No clear, definitive answers to these problems

Computer Architecture Today

- Computing landscape is very different from 10-20 years ago
- Both UP (software and humanity trends) and DOWN (technologies and their issues), FORWARD and BACKWARD, and the resulting requirements and constraints



Course Logistics

Intended Learning Outcomes

- What are you expected to do after the course?
  - Remembering: Describe, define, draw, recite, quote, name, tell, write
  - Understanding: Classify, compare, exemplify, conclude, demonstrate, illustrate, interpret.
  - Applying: Apply, change, choose, compute, dramatize, implement, prepare, produce.
  - Analysing: Characterize, compare, contrast, debate, deduce, differentiate, distinguish, examine.
  - Evaluating: Appraise, argue, assess, choose, justify, predict, prioritize, prove, rank, monitor.
  - Creating: Construct, design, develop, generate, invent, compose, plan, make, hypothesize.
- Source : Bloom's Taxonomy. Anderson and Krathwohl (2001)

Course

- Course website to be hosted on Piazza.
  - Detailed syllabus and lecture plan already online.
  - For all lecture slides
  - Reading Materials
  - Discussions/Doubts/Clarifications

- Office Hours:
  - Anytime

Assessment

- TBD

Academic Integrity

- Be careful about:
  - Verbatim reproduction from external resources
  - Paraphrasing from earlier work
  - Disrupting classroom activity
  - Cheat, copy, and collaborate on assignments
- Participate in discussions
  - There are no “stupid” questions ☺
  - Be considerate of your fellow classmates