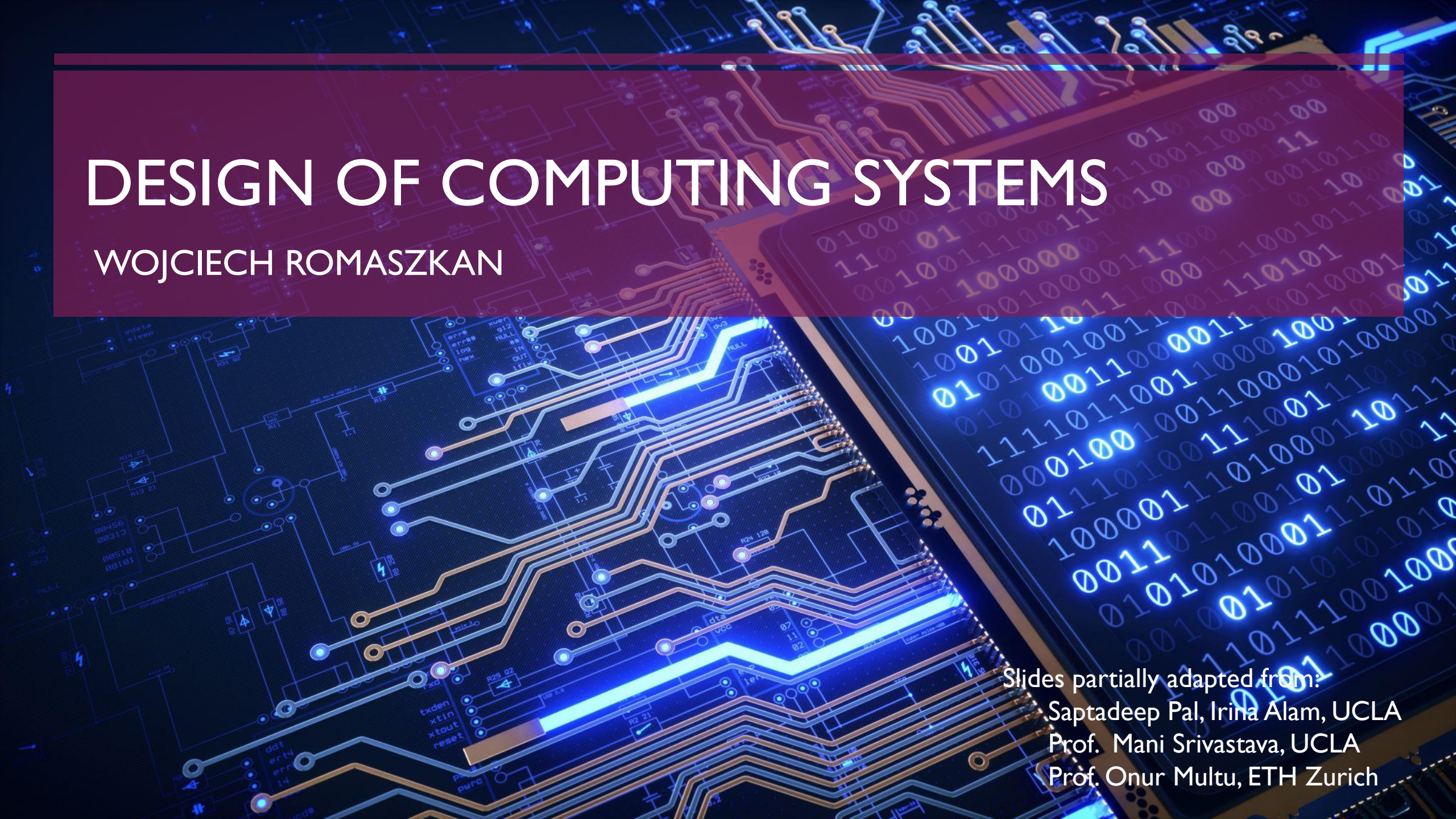
DESIGN OF COMPUTING SYSTEMS

WOJCIECH ROMASZKAN



Slides partially adapted from:
Saptadeep Pal, Irina Alam, UCLA
Prof. Mani Srivastava, UCLA
Prof. Onur Multu, ETH Zurich

WHAT ARE COMPUTERS?

Computers are number crunching machines

WHY DO WE NEED COMPUTERS?

TO SOLVE PROBLEMS

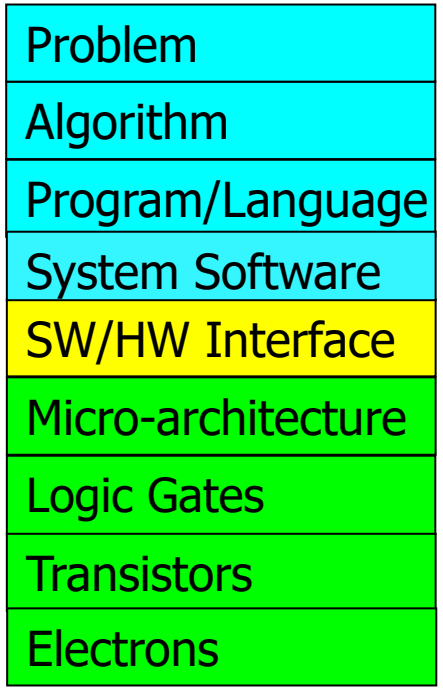
TO GAIN INSIGHT

TO ENABLE A BETTER LIFE AND FUTURE

HOW DO COMPUTERS SOLVE PROBLEMS?

Algorithm

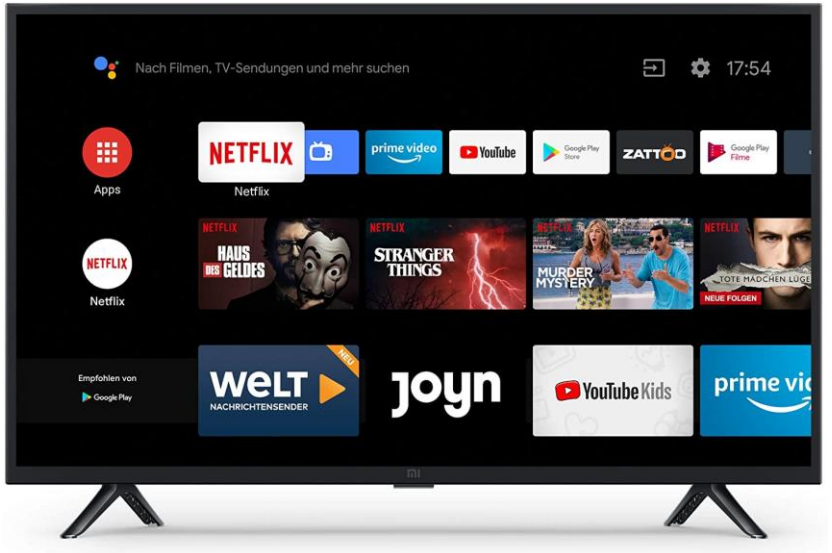
- Step by step procedure that is guaranteed to terminate
- Each step is precisely stated
- Can be carried out by a computer



COMPUTING SYSTEMS AROUND US (INSIDE OUR HOMES)

CAMERAS AND DRONES

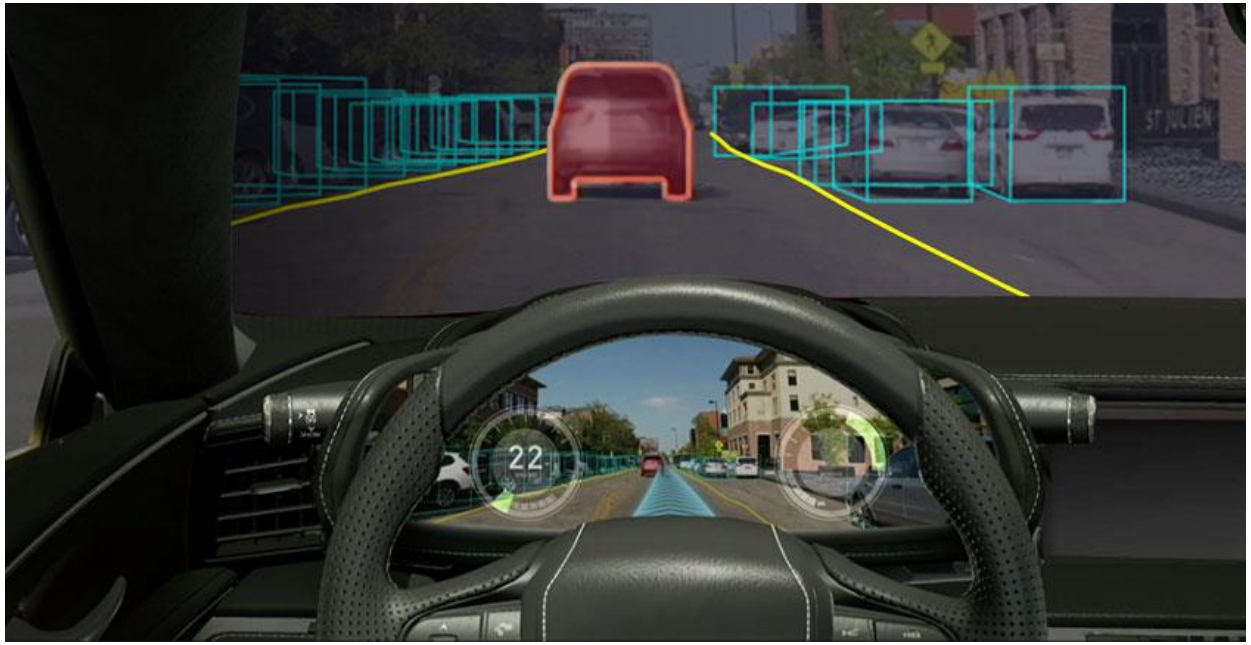
COMPUTING SYSTEMS AROUND US



PERSONAL COMPUTERS

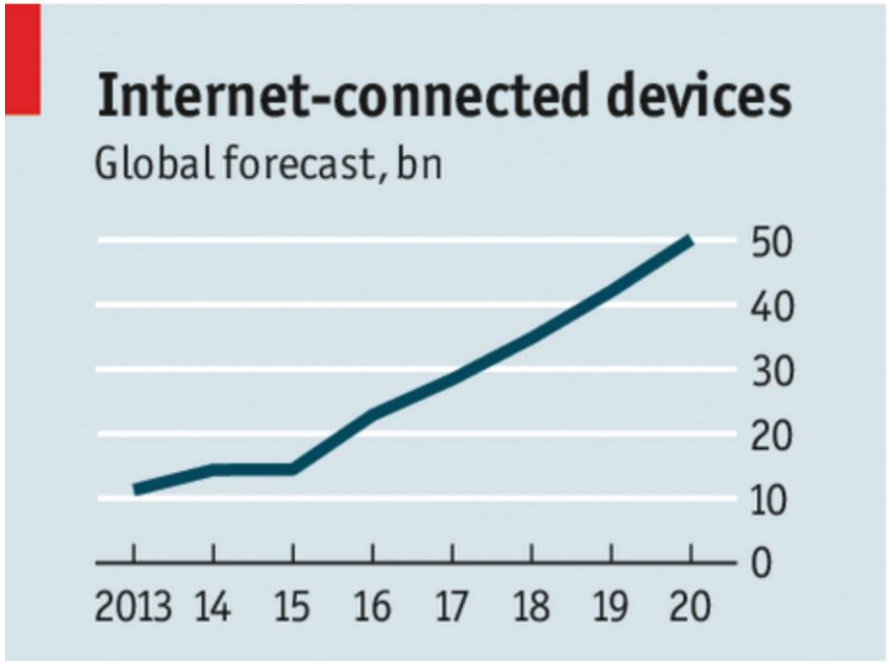
SUPERCOMPUTERS

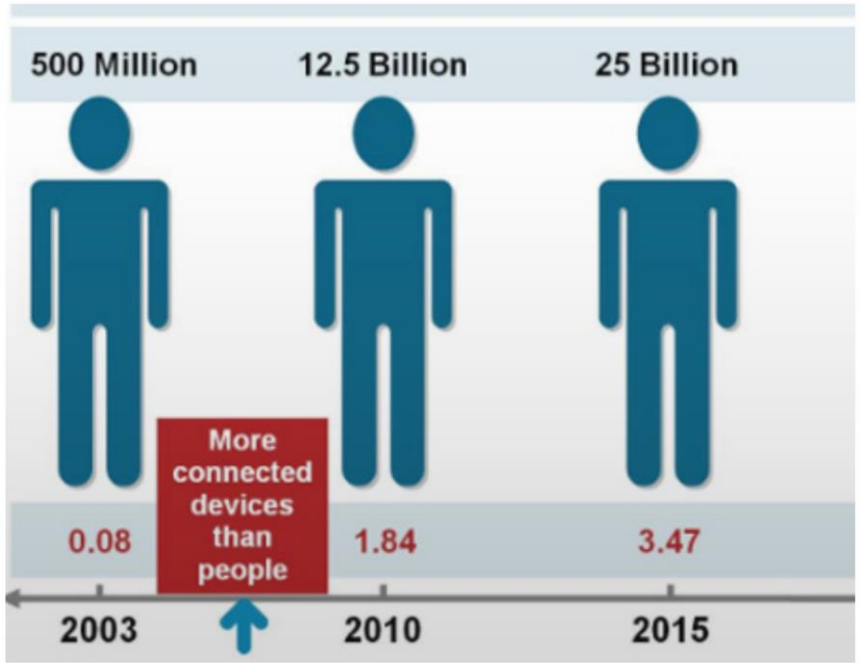
FROM CARS TO SPACESHIPS

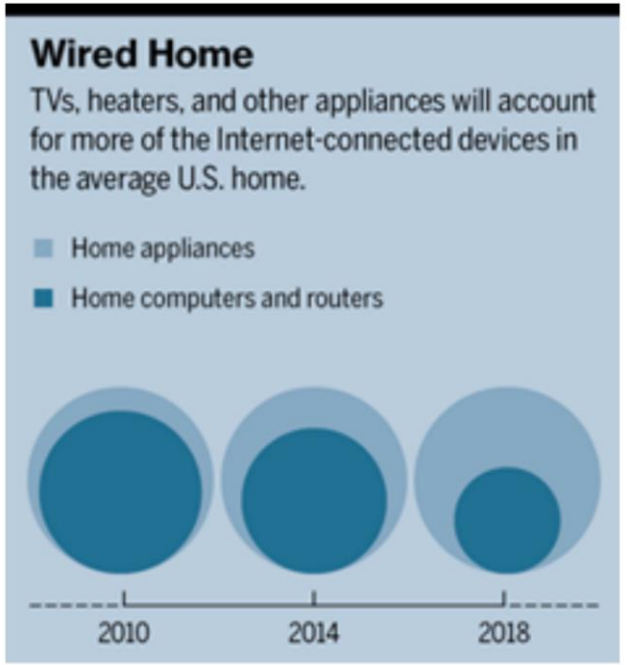


EMBEDDED AND CONNECTED EVERYWHERE

- Everything has an Embedded Computer
- The Internet is Everywhere
- All Computers have a Network Interface

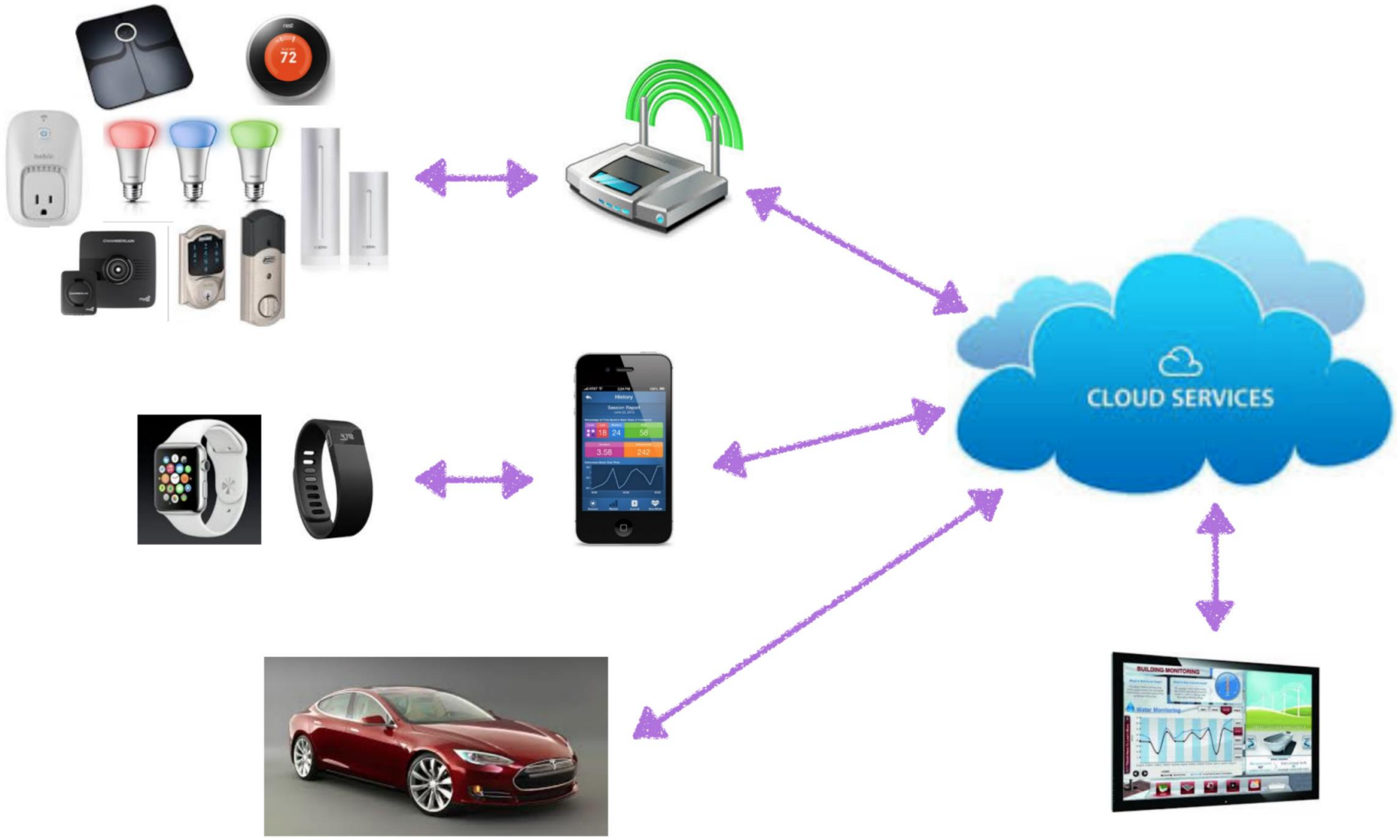




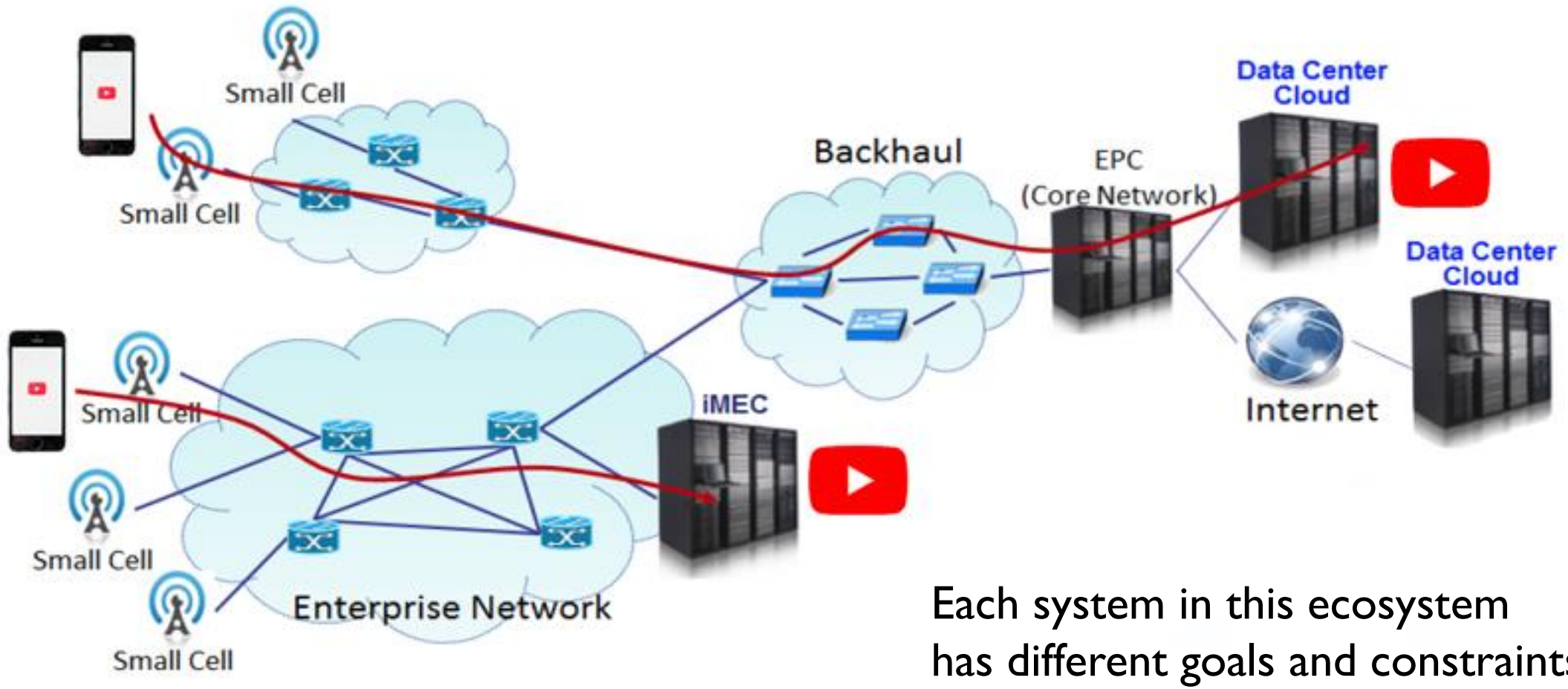


MIT Technology Review

A CONNECTED WORLD

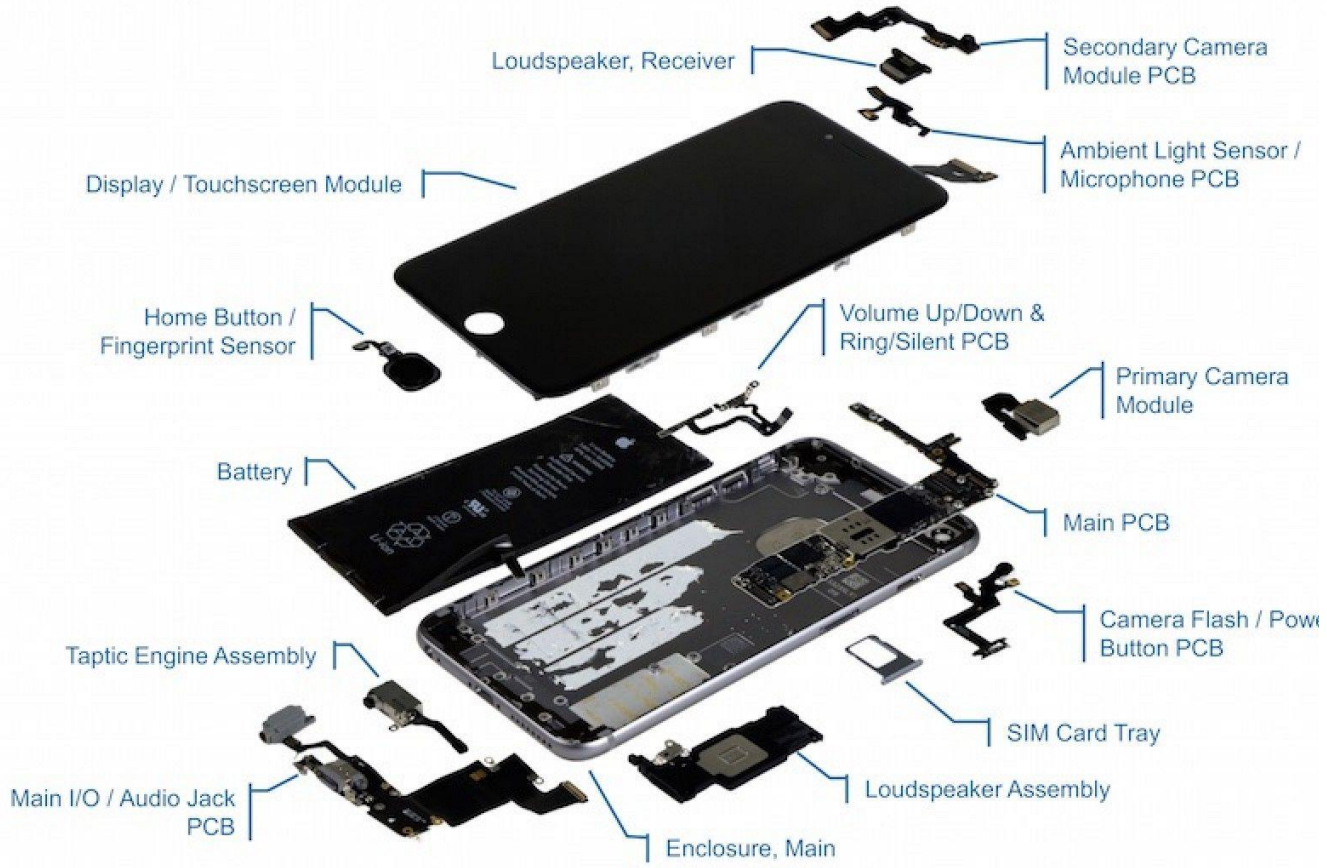


NETWORKED SYSTEMS



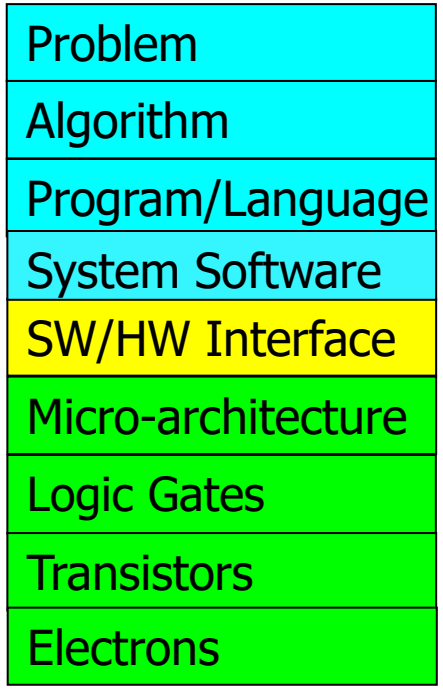
Each system in this ecosystem has different goals and constraints

LET'S LOOK AT AN OVERALL SYSTEM (GADGETS)



EVERY COMPUTING SYSTEM IS TRYING TO SOLVE PROBLEM(S)

- Different systems, different ***problems to solve***
- Different systems, different ***goals to achieve***



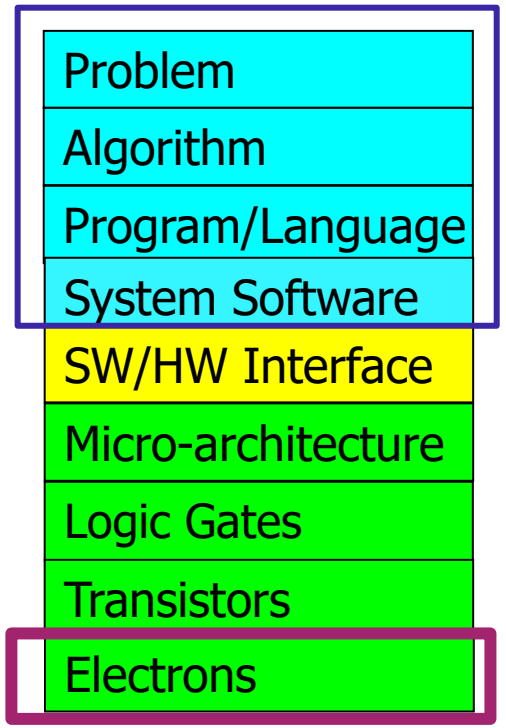
ARCHITECTING COMPUTING SYSTEMS

- Science and Art of designing computing platforms (hardware, interface, system software and programming model)

- Achieve a set of design goals
  - Best possible performance (supercomputers)
  - Longest battery life/ less power consumption (sensors, wearables)
  - Small form factor and light weight (Phones, watches, drones)
  - Good average performance at affordable cost (Laptops, desktops)
  - Highly reliable systems (autonomous vehicles, medical robots, spaceships)
  - …….

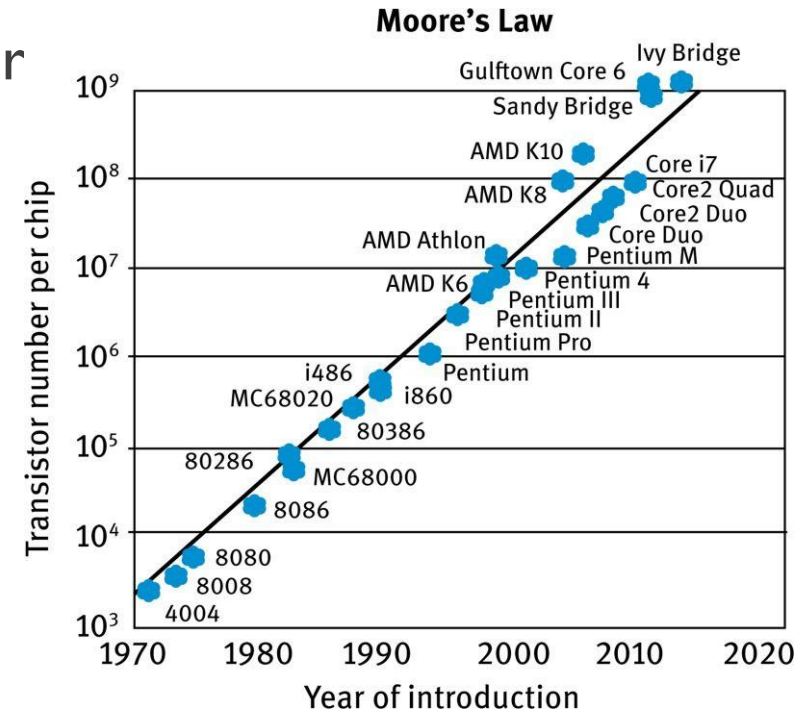
COMPONENTS OF A COMPUTING SYSTEM - ELECTRONS

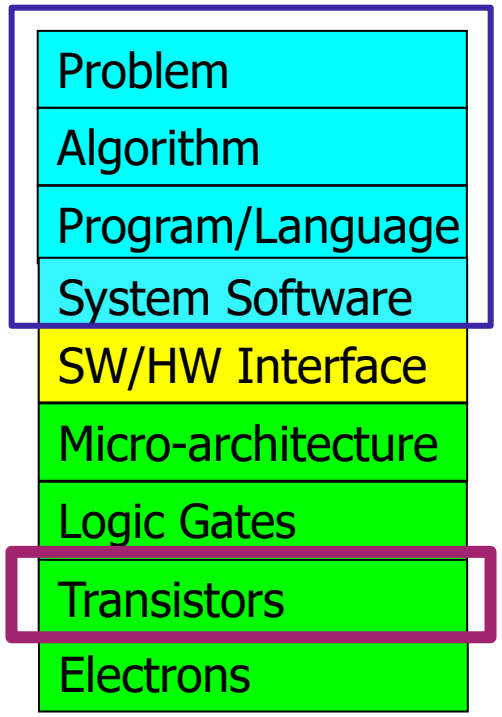
- **Electron**ics is driven by laws of physics
- Computing systems orchestrate electrons to do tasks



COMPONENTS OF A COMPUTING SYSTEM - TRANSISTORS

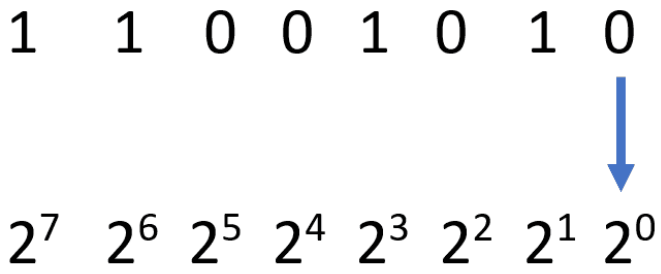
- **Transistors** control flow of electrons
- Transistors act as switches in digital systems – 1/0
- Semiconductor physics deals with design of transistors
- More the number of transistor computing system
  - Also, more area and power
- Over years, transistors have become smaller
  - Same area – more transistors
  - Reduced power



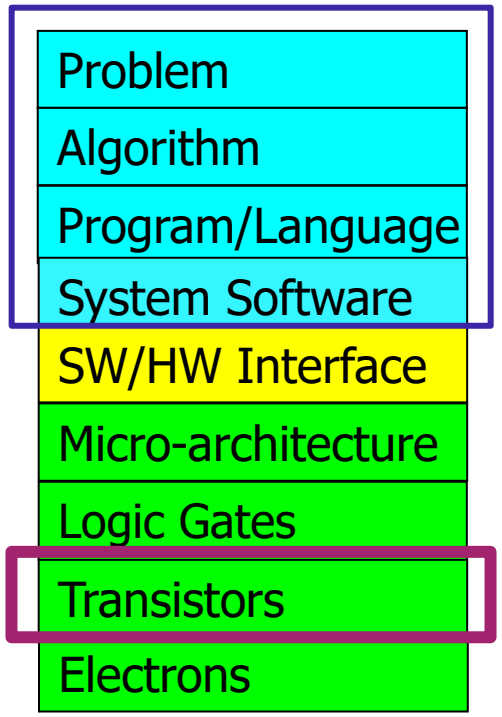


ELECTRONIC NUMBER REPRESENTATION

- If a signal can represent only 0 or 1, how can we represent bigger numbers?
  - Binary representation!

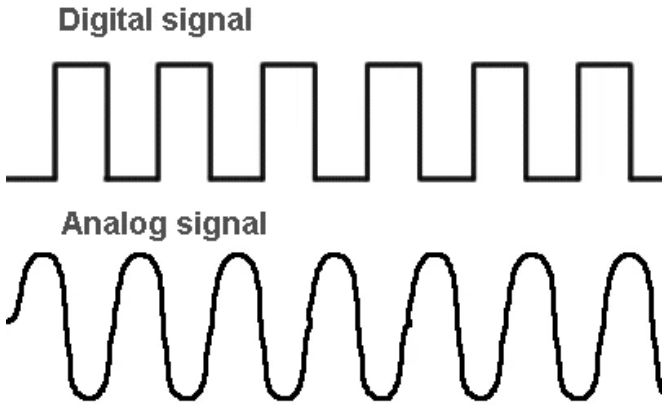


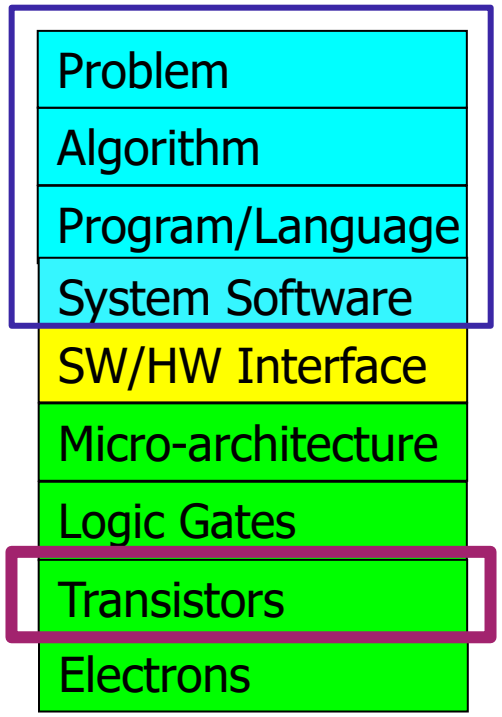
- What about negative values? Non-integers?

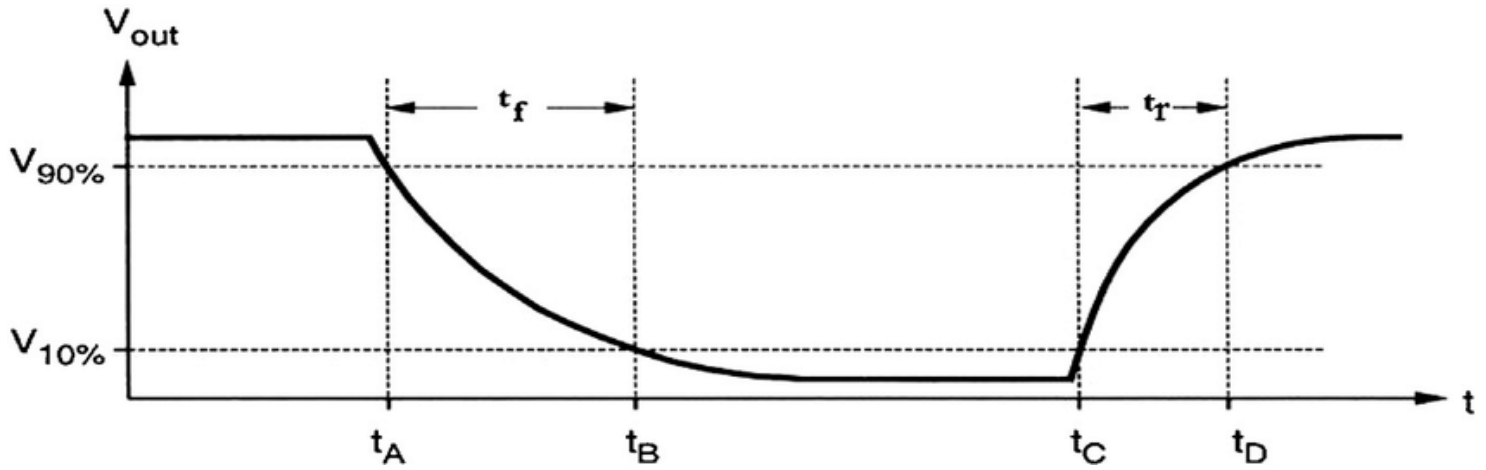


WHAT IS A DIGITAL SYSTEM?

- Digital signals are discrete.
- Real world is not discrete.
- Are digital electronics a lie?



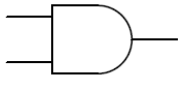




Source: researchgate.net

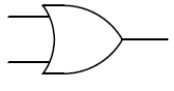
COMPONENTS OF A COMPUTING SYSTEM – LOGIC GATES

- **Logic gates** are the building blocks of digital systems
- Multiple transistors are combined to build logic gates
- Examples: AND, OR, NOT, XOR



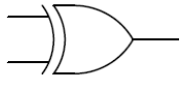
AND

A	B	Output
0	0	0
0	1	0
1	0	0
1	1	1



OR

A	B	Output
0	0	0
0	1	1
1	0	1
1	1	1

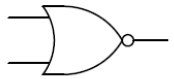


XOR

A	B	Output
0	0	0
0	1	1
1	0	1
1	1	0

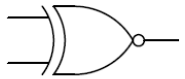
NAND

A	B	Output
0	0	1
0	1	1
1	0	1
1	1	0



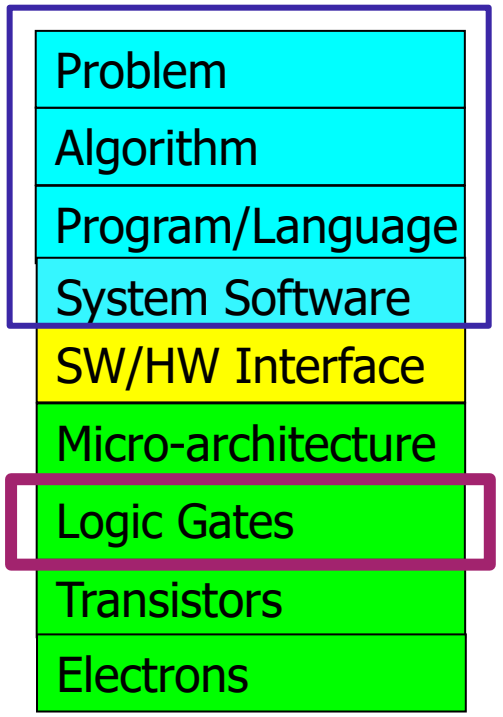
NOR

A	B	Output
0	0	1
0	1	0
1	0	0
1	1	0



XNOR

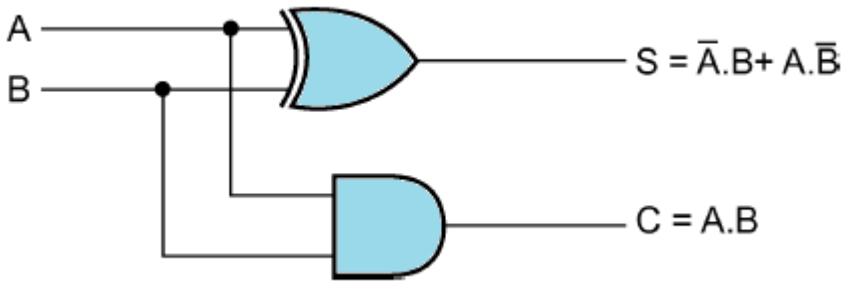
A	B	Output
0	0	1
0	1	0
1	0	0
1	1	1



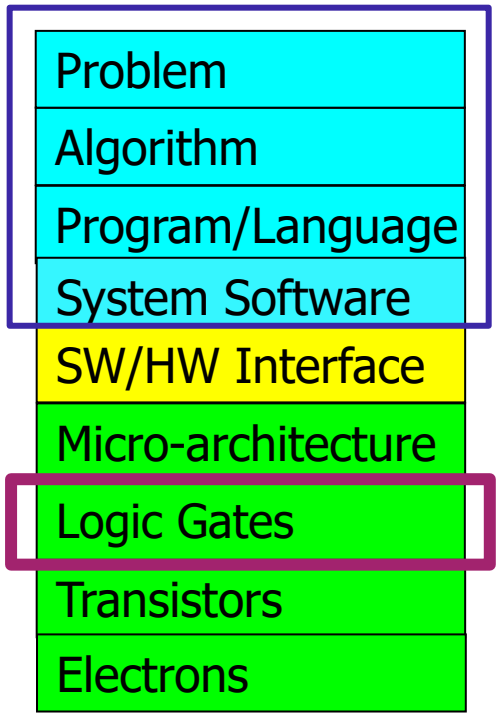
COMPONENTS OF A COMPUTING SYSTEM – LOGIC GATES

- Logic gates are pretty basic – what can you do with them?
- Let's figure out how to add two one-bit numbers.
  - Each one is either 0 or 1.
  - How many bits is the output?

A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

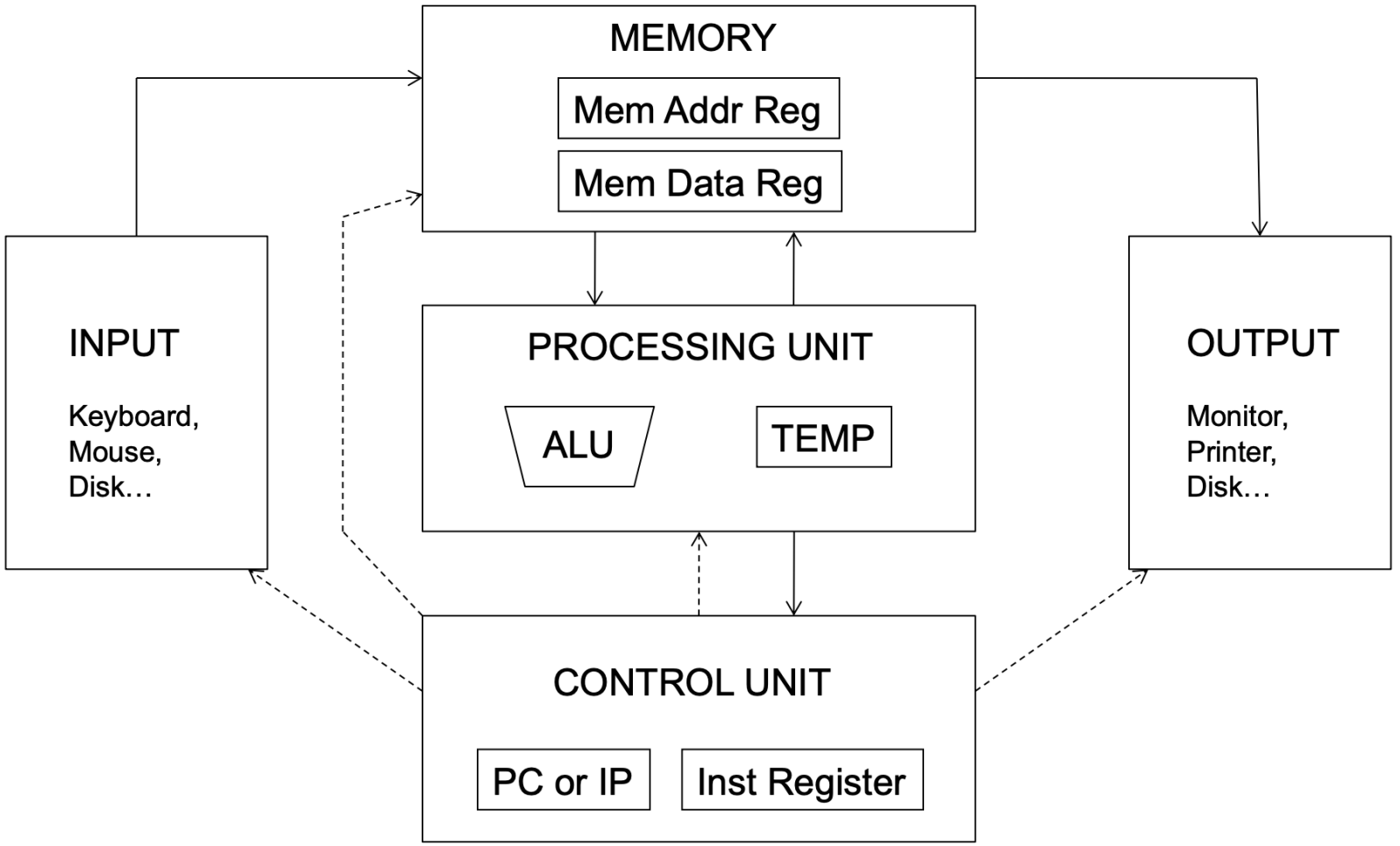


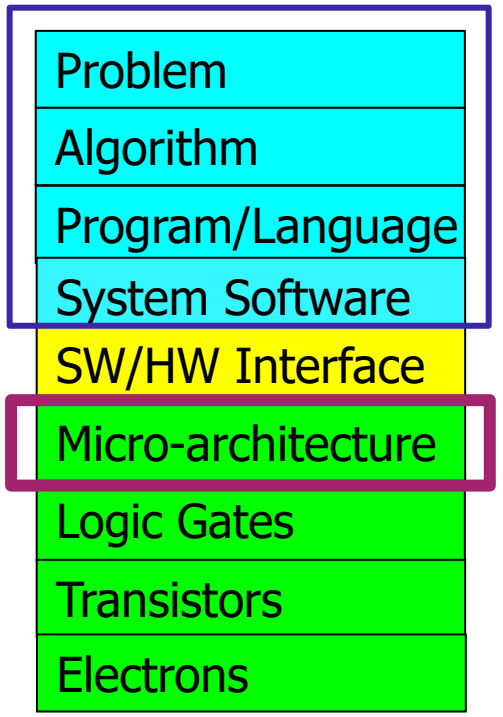
- How to build an n-bit adder? Multiplier?



COMPONENTS OF A COMPUTING SYSTEM – MICRO-ARCHITECTURE

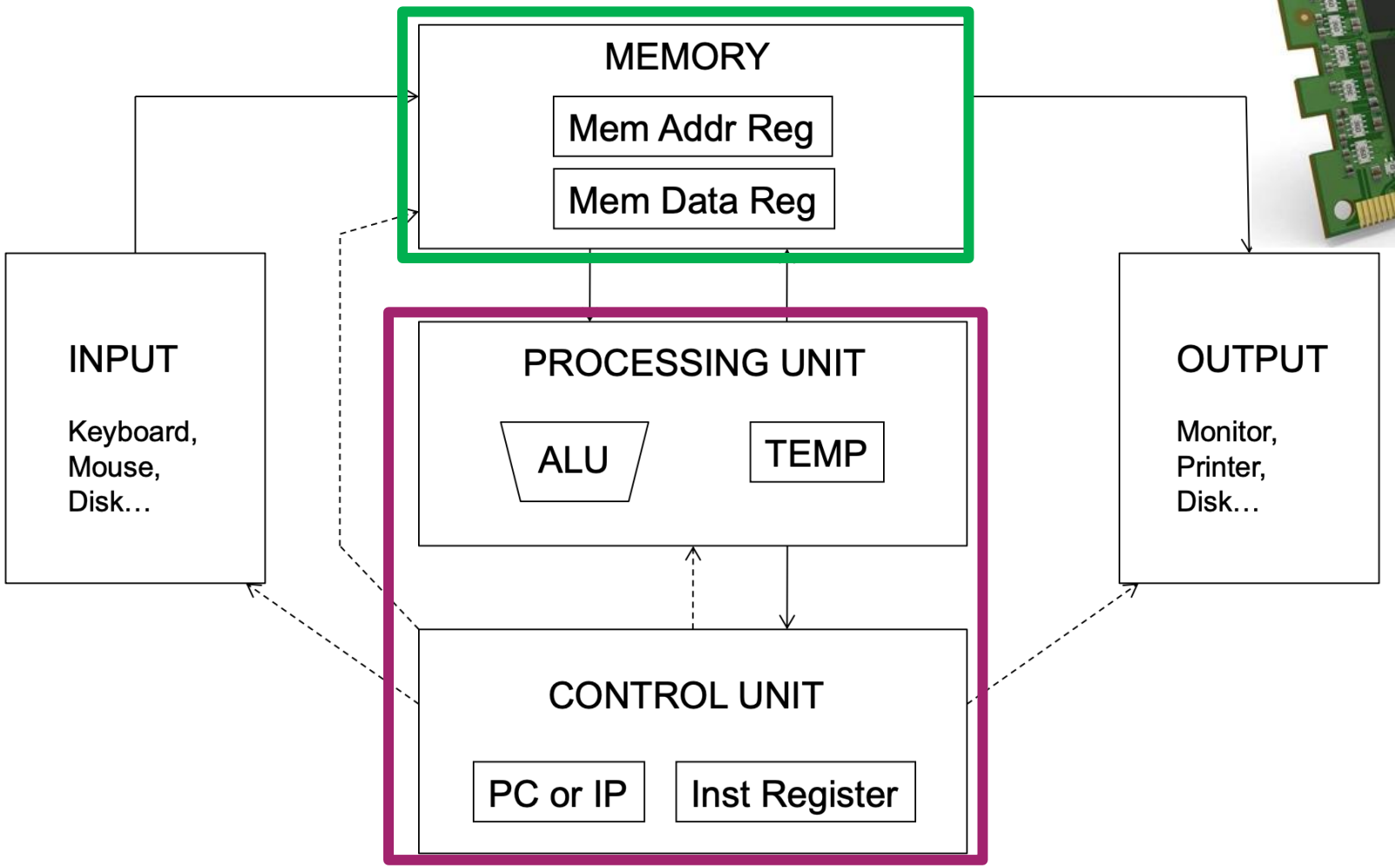
- **The Von Neumann Model**

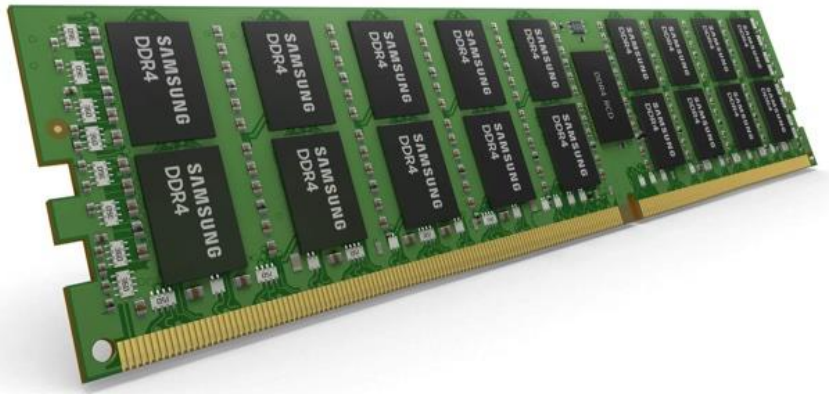




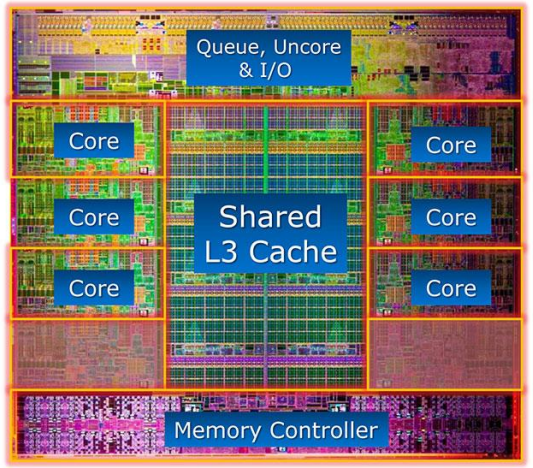
COMPONENTS OF A COMPUTING SYSTEM – MICRO-ARCHITECTURE

- **The Von Neumann Model**



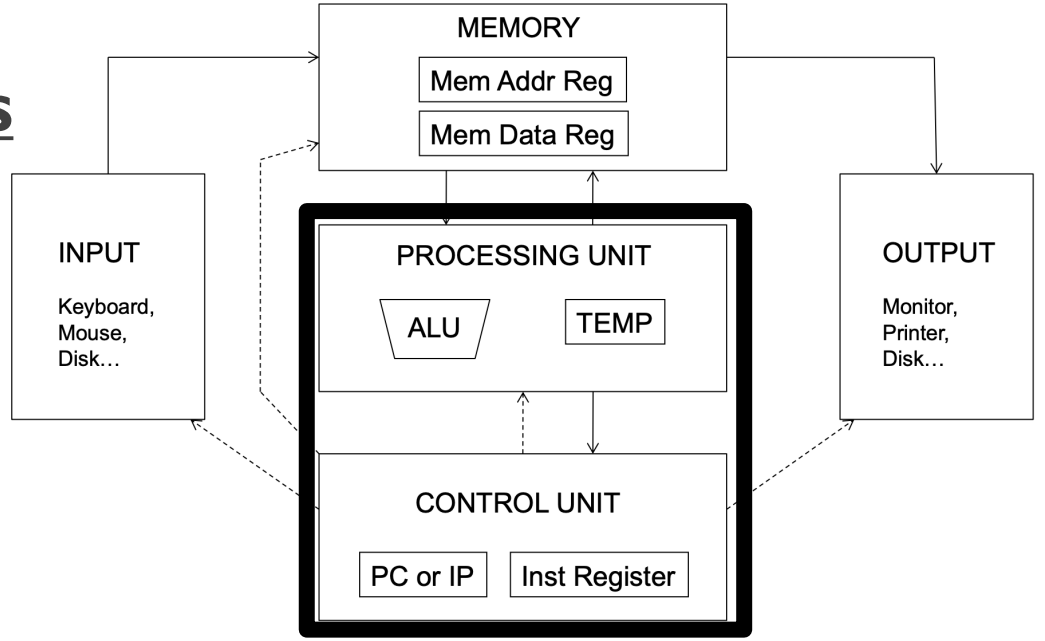


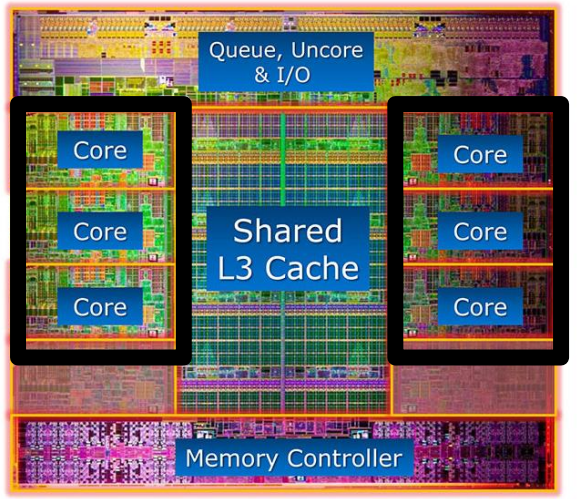
Intel® Core™ i7-3960X Processor Die Detail



DIFFERENT FLAVORS OF PROCESSING UNIT

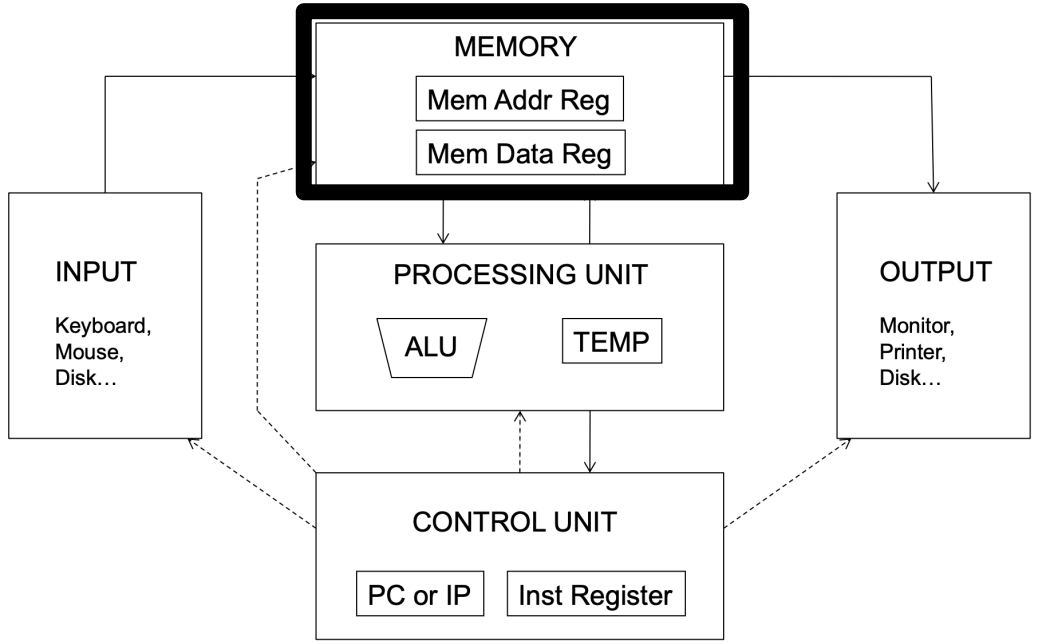
- **Arithmatic and Logic Unit + Registers**
  - Single or multiple cores in a system
  - Each core – several processing elements
  - Different types of cores –
    - Fast, High energy consumption
    - Slow, energy efficient

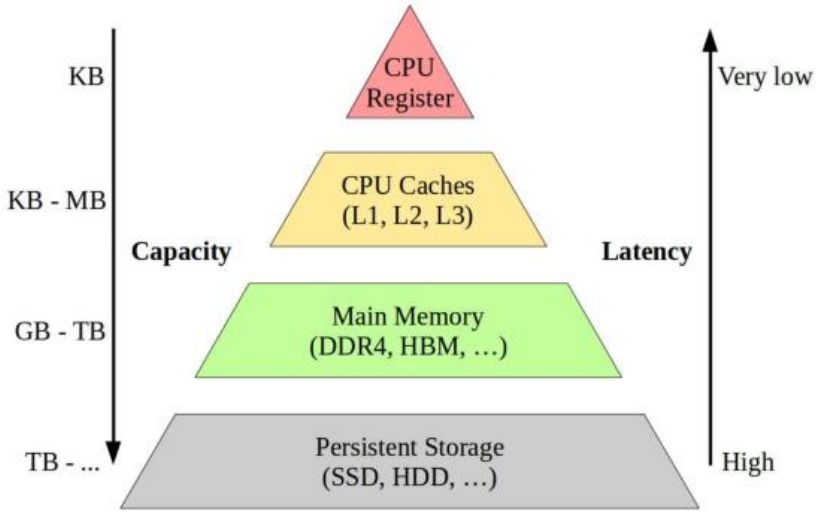


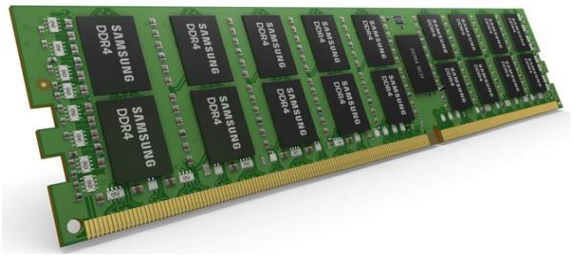


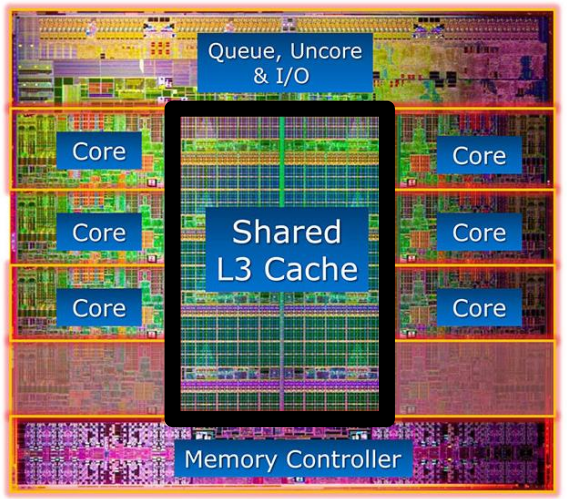
DIFFERENT FLAVORS OF MEMORY

- **Caches + Main Memory + Storage**
  - Caches – Smaller, faster on-chip memory, temporary
  - Main Memory – Bigger, moderately fast, off-chip, temporary/persistent
  - Storage – Biggest, slowest, off-chip, persistent

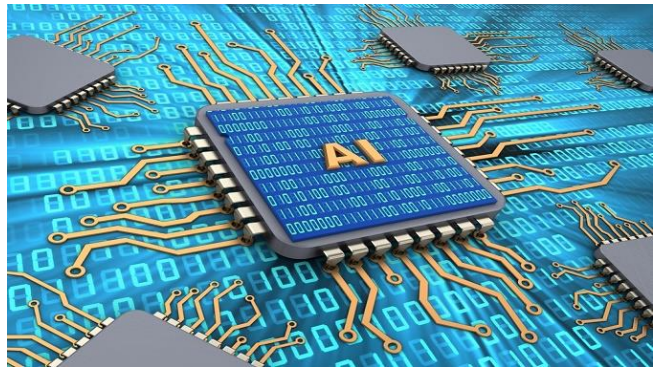


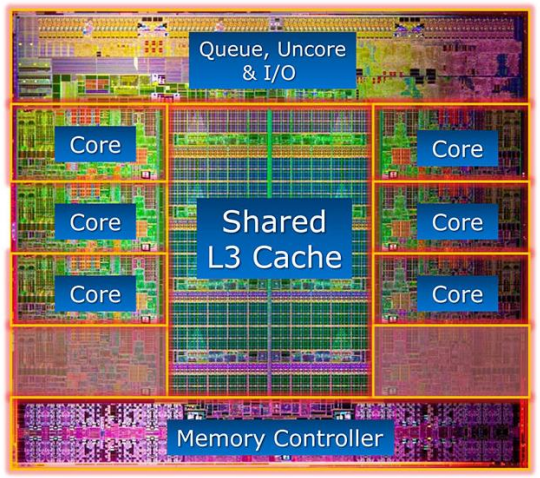






GENERAL PURPOSE vs. APPLICATION SPECIFIC

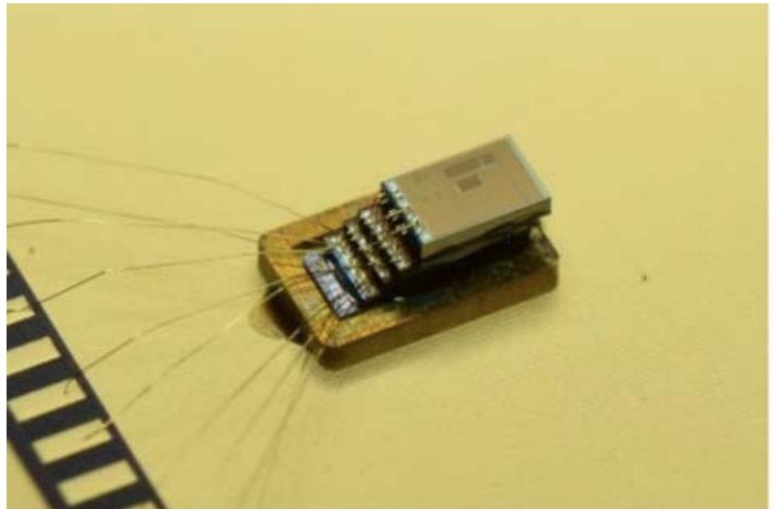


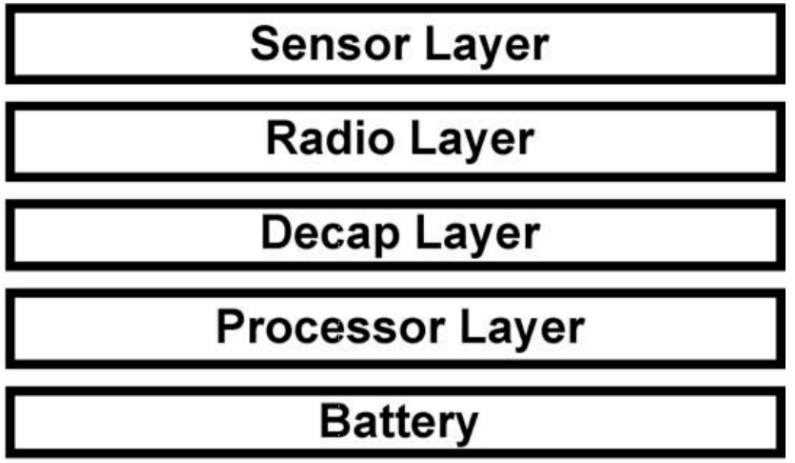


DIFFERENT PLATFORMS, DIFFERENT GOALS

Temperatur Sensor

- Very Lightweight processor
- Few thousand transistors
- Area – 0.09mm^2 (180nm technology)
- Power – 71nW
- Memory – Few 10s of kilobytes (KBs)





DIFFERENT PLATFORMS, DIFFERENT GOALS

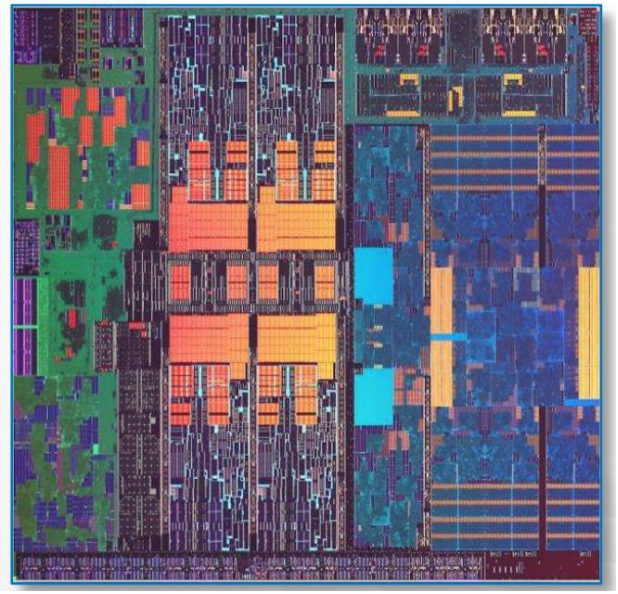
Raspberry Pi

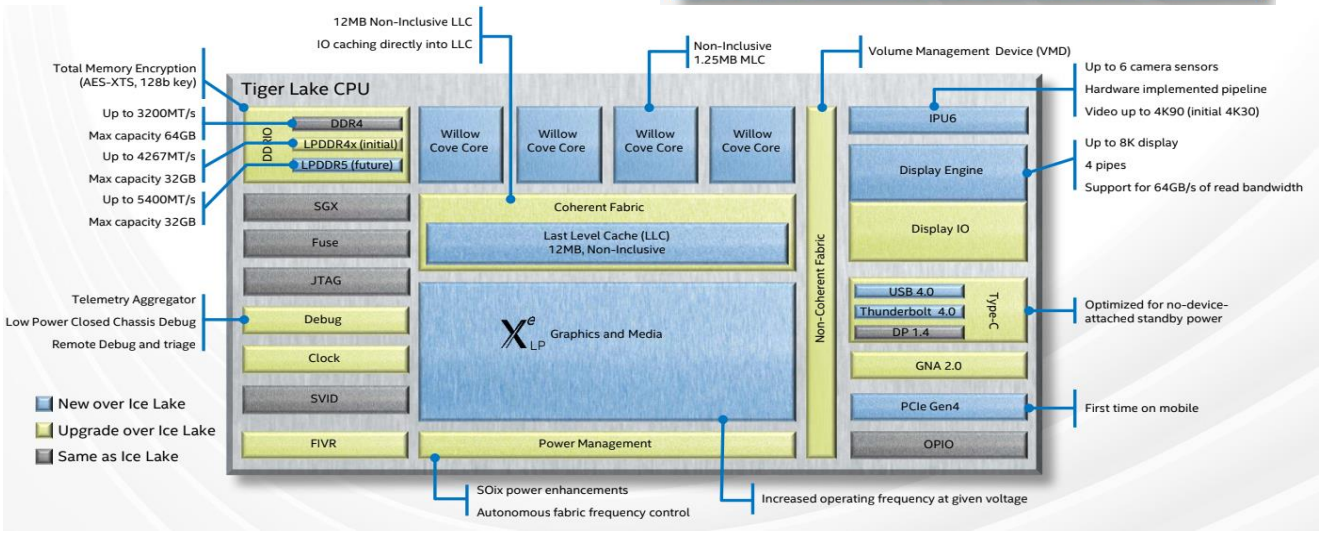
- Lightweight processor
- ~250 million transistors
- Area – ~50mm^2 (40nm technology)
- Power – 1-2W
- Memory – Few gigabytes (GBs) of memory

DIFFERENT PLATFORMS, DIFFERENT GOALS

Intel Tiger Lake

- Mobile processor
- >10 billion transistors
- Area – ~150mm^2 (10nm technology)
- Power – 15W
- Memory – 10s of MBs of ultra-fast memory
  10s of GBs of slower memory

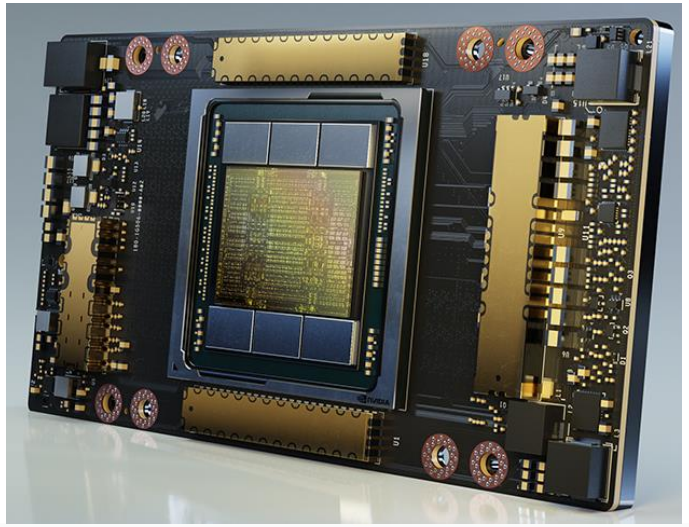




DIFFERENT PLATFORMS, DIFFERENT GOALS

NVIDIA A100 GPU

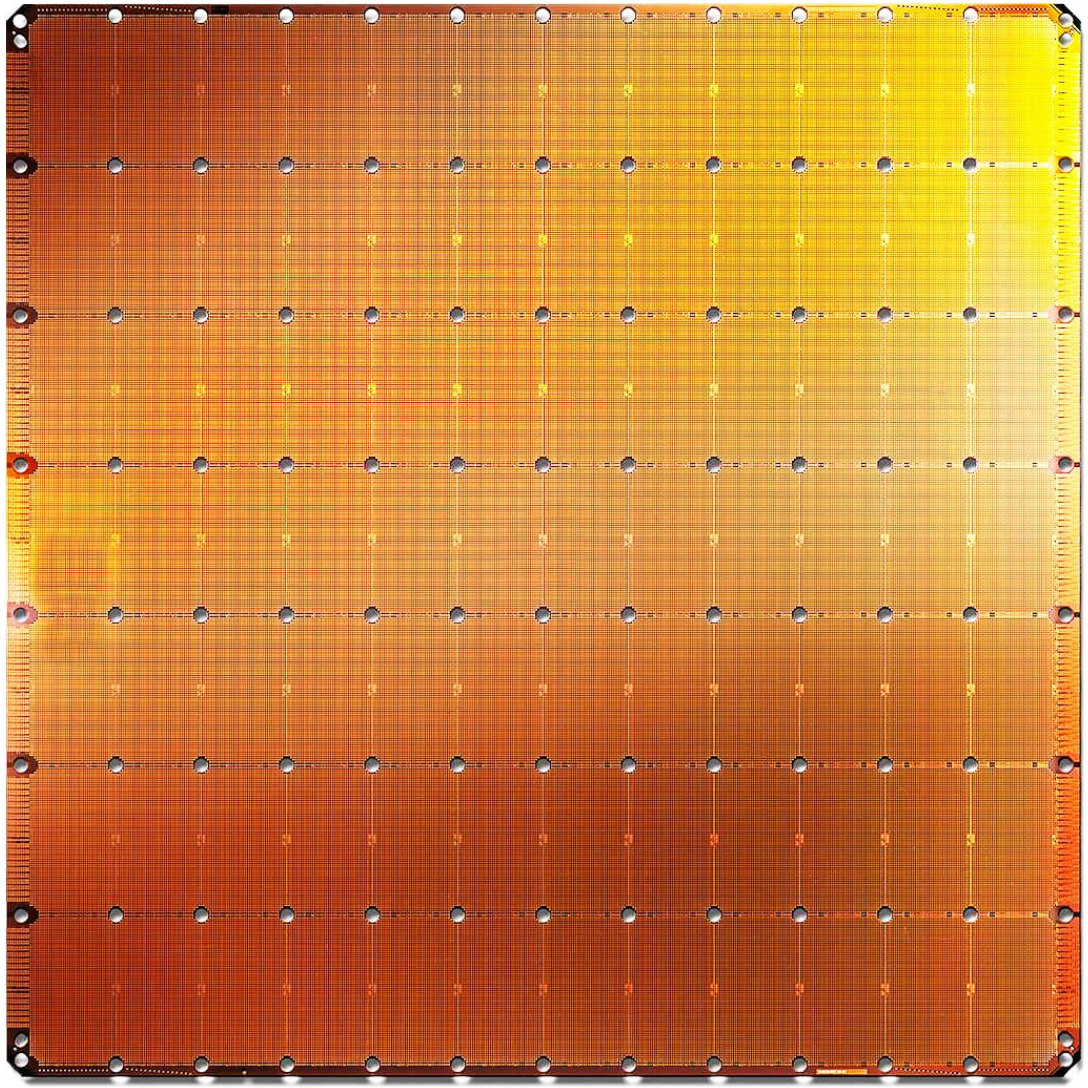
- Used for graphics and AI training
- >54 billion transistors
- Area – ~800mm^2 (7nm technology)
- Power – 400W
- Memory – 100s of MBs of ultra-fast memory
  100s of GBs of fast memory
  TBs of slower memory



DIFFERENT PLATFORMS, DIFFERENT GOALS

Cerebras Waferscale System

- Used for AI training and inference
- >2 trillion transistors
- Area – ~46000mm^2 (7nm technology)
- Power – 20KW
- Memory – 10s of GBs of ultra-fast memory



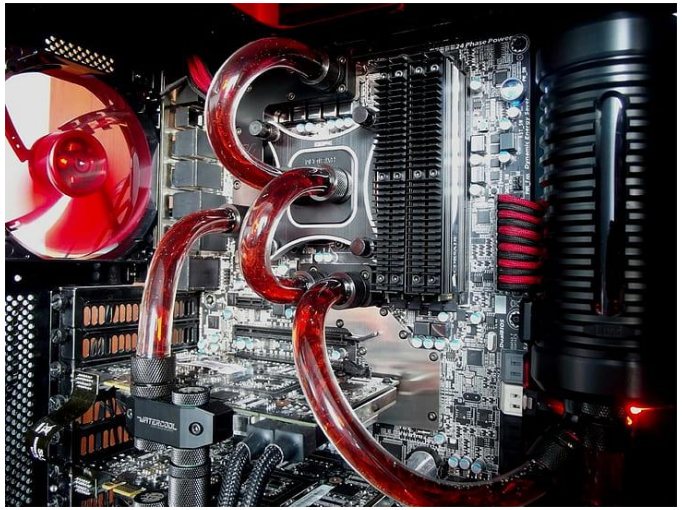
DIFFERENT PLATFORMS, DIFFERENT CONSTRAINTS (POWER)

What is the difference between the design goals of these two systems?

DIFFERENT PLATFORMS, DIFFERENT CONSTRAINTS (RESILIENCY)

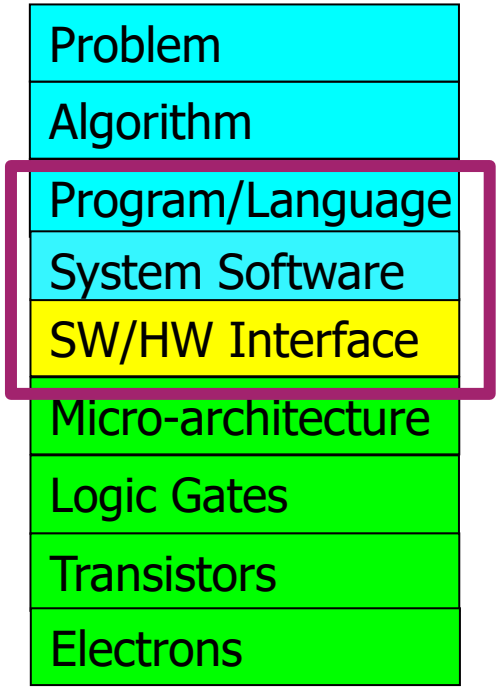
What is the difference between the design goals of these systems?

DIFFERENT PLATFORMS, DIFFERENT CONSTRAINTS (THERMAL)



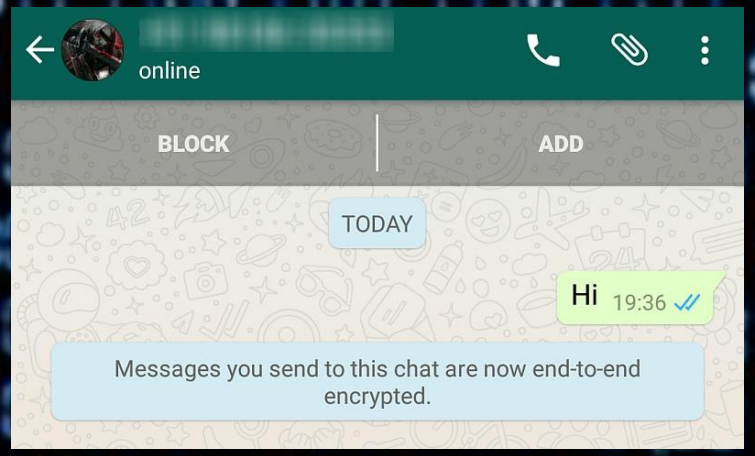
COMPUTING SYSTEMS RUN SOFTWARE

- Wide choice of software abstractions exist
- General purpose languages like Python/C++ etc. are easy to learn
- Not all systems can be programmed using these languages
- Domain specific hardware often performs efficiently when programmed using not commonly used languages
- Providing **nice** software interface to programmers is important
- Designing software stack - important part of designing hardware systems
- Hot and active field in Computer Engineering

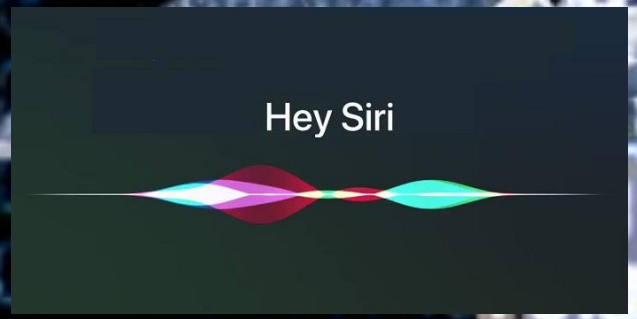


CRUX OF DESIGNING COMPUTING SYSTEMS

- Different systems, different ***problems to solve***
- Different systems, different ***goals to achieve***
- Different systems, different ***constraints to obey***
- Multiple choices available to achieve same goal
  - Computing system design is the art and science of choosing the optimal set
- Designing computing systems – interdisciplinary task
  - From controlling electrons efficiently to writing optimized programs
  - Holistic across-the-stack thinking is required



SECURITY – ANOTHER IMPORTANT ASPECT OF SYSTEM DESIGN



THANK YOU!

EXCITING EXERCISES TO FOLLOW NEXT