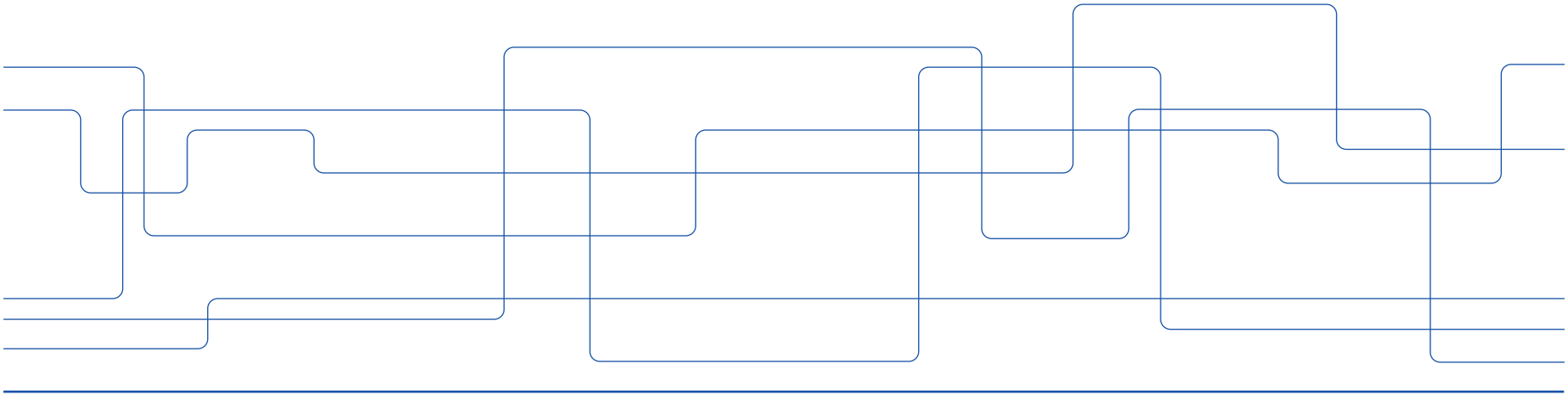
DD2358 – Fundamentals of Computer Systems – Computing Units

Stefano Markidis

KTH Royal Institute of Technology



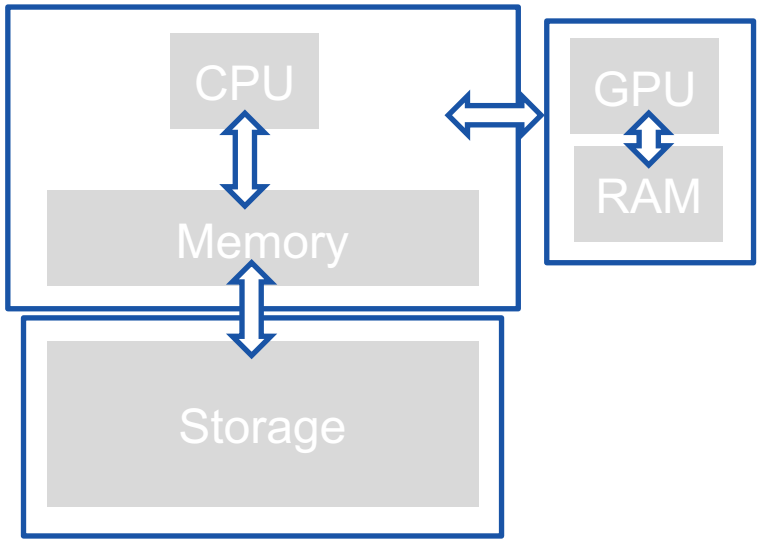
Intended learning outcomes

- List the main computer system components
- State the difference between CPU and accelerators (GPUs, FPGAs, …)
- Describe the main parameters characterizing the computing units performance
- Describe the historical development of the CPUs
- Motivate the end of the Dennard's scaling
- Motivate why the Amdahl's law limits the parallel performance.

Computer Components

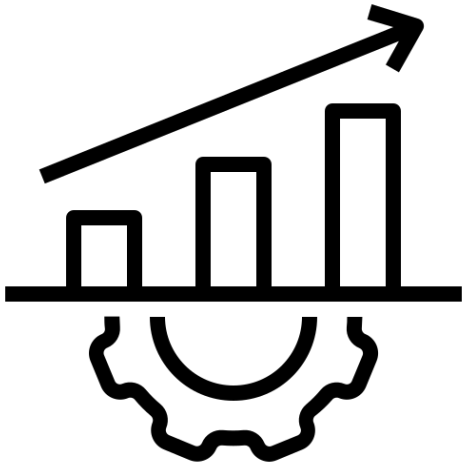
The underlying components that make up a computer can be simplified into three basic parts:

1. Computing units
2. Memory and storage units
3. Connections between them (Buses)



Computer Units Performance Characteristics

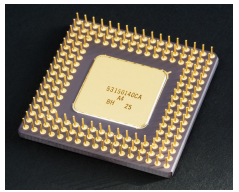
- The computational unit has the property of **how many computations** it can do per second
- The memory unit has the properties of **how much data** it can hold and how fast we can read from and write to it
- Connections have the property of how **fast they can move data** from one place to another.



Computing Units / CPU

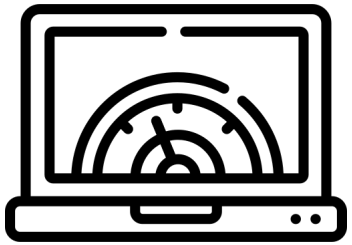
Computing Units / CPU

- The *computing unit* of a computer provides the **ability to transform any bits it receives** into other bits or to change the state of the current process.
- **CPUs** are the most commonly used computing unit
  - CPUs are general-purpose
  - **Vendors:** Intel, AMD, Apple, …
- Graphics processing units (GPUs), Fields Programmable Gate Arrays and AI **Accelerators** are gaining popularity as auxiliary computing units.
  - **Specialized computing units**: very good only for certain tasks
  - **Vendors:** Nvidia, AMD, Intel, Xilinx, Google, Amazon, Tesla…



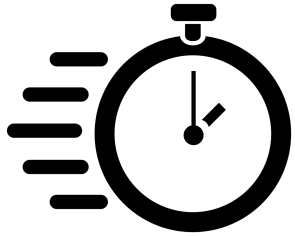
HPC - Characterizing Computing Performance

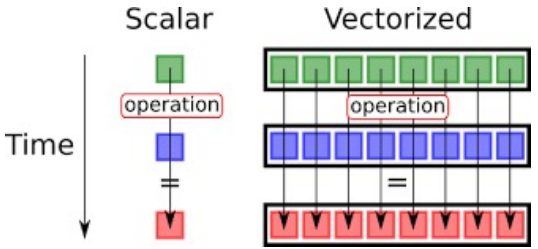
- The main properties of interest in a computing unit are
  1. The number of **operations it can do in one cycle (IPC)**
     - > *Depends on CPU and vector units/extension*
     - > *Depends on the workload / code*
     - > *Info not shared by vendors specs*
     - > *Measurable with benchmarks*
  2. The **number of cycles it can do in one second (Clock Frequency)**.
     - > *Provided by vendors*
     - > *Frequency: Typically on the order 1-4 GHz today*
  3. **FLOPS/s** = number of floating point operations per second
     1. *Benchmark and models*
     2. *Modern computing units can reach hundred MFLOPS/s /andTFLOPS/s*



HPC - Performance with Clock Frequency and IPC

1. Increasing **clock speed** almost immediately speeds up all programs running on that computational unit
   - they are simply able to do more **calculations per second**
   - **Power Wall**
2. Having a higher IPC can also drastically affect computing by changing the level of ***vectorization*** **that is possible**.
   - Vectorization occurs when a CPU is provided with multiple pieces of data at a time and is able to operate on all of them at once.
   - This sort of CPU instruction is known as **single instruction, multiple data (SIMD).**

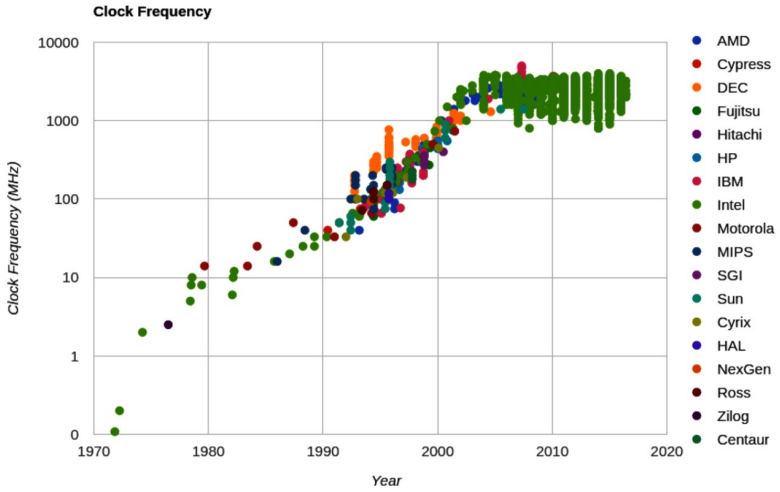




Source:
https://lappweb.in2p3.fr/~paubert/ASTERICS_HPC/6-6-1-985.html

Historical Improvement of Computing Units

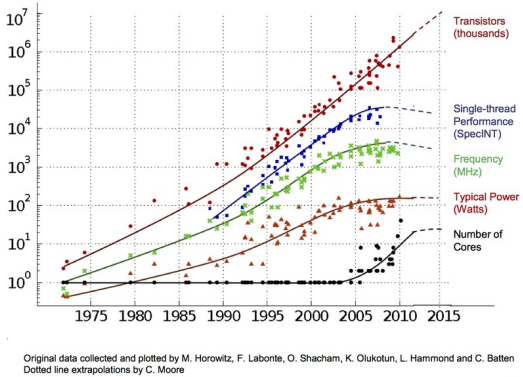
- In general, computing units have advanced quite slowly over the past decade.
  - Moore's law is still true?
- Clock speeds and IPC have both been stagnant because of the physical limitations of making transistors smaller and smaller.
- As a result, chip manufacturers have been relying on other methods to gain more speed, including
  - **Simultaneous multithreading** (where multiple threads can run at once)
  - More clever out-of-order execution
  - **Multicore** architectures.



Clock speed of CPUs over time (from CPU DB)

The End of Dennard's Scaling – The Power Wall

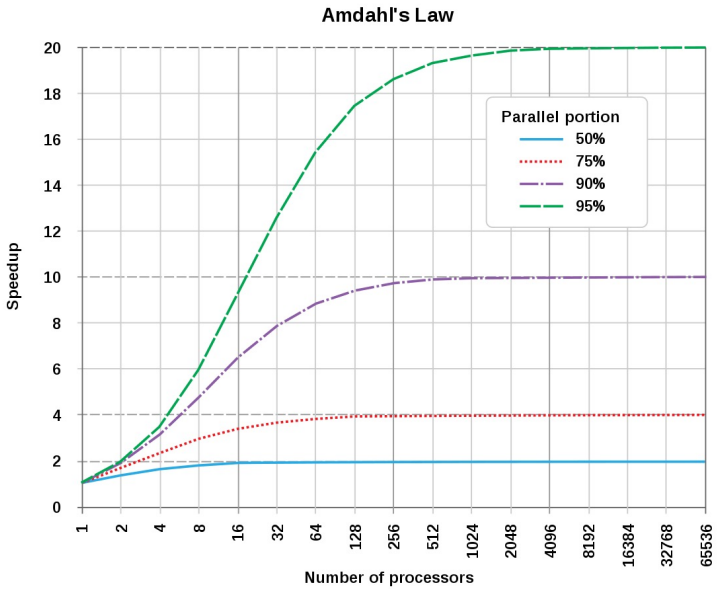
- **Dennard scaling** is states that, as transistors get smaller, their **power density stays constant**, so that the power use stays in proportion with area
- Dennard's scaling stops **mainly due to supply voltage limits, power densities rapidly increase on the chip**.
  - A significant amount of on-chip resources needs to stay dark, i.e., power-gated, in order to avoid thermal emergencies (**Dark silicon**)



Original data collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond and C. Batten
Dotted line extrapolations by C. Moore

Parallel Performance Limitation - Amdahl's Law

- Simply **adding more cores** to a CPU **does not always speed** up a program's execution time.
  - It depends on the algorithm that can offer different degrees of parallelization!
  - Many algorithms have a serial part limiting the parallel speed-up
- This is because of *Amdahl's law:*
  - if a program designed to run on multiple cores has some subroutines that must run on one core, this will be the limitation for the maximum speedup that can be achieved by allocating more cores.



To Summarize

- CPU are general-purpose while accelerators (GPUs, FPGAs, …) are specialized computing units
- Computing unit performance is characterized by three numbers: IPC, clock frequency, FLOPS/s
- The clock frequency of processors stagnated in the last few decades (discussion on the Moore's law death). However, the parallelism increased by adding multiple cores.
- End of Dennard's scaling: going to smaller transistor is not leading to use less power
- Amdahl law's: algorithms that have a serial part can't fully exploit available parallelism (multicore processors)