CM12002
Computer Systems Architectures

Parallel Architectures

Fabio Nemetz

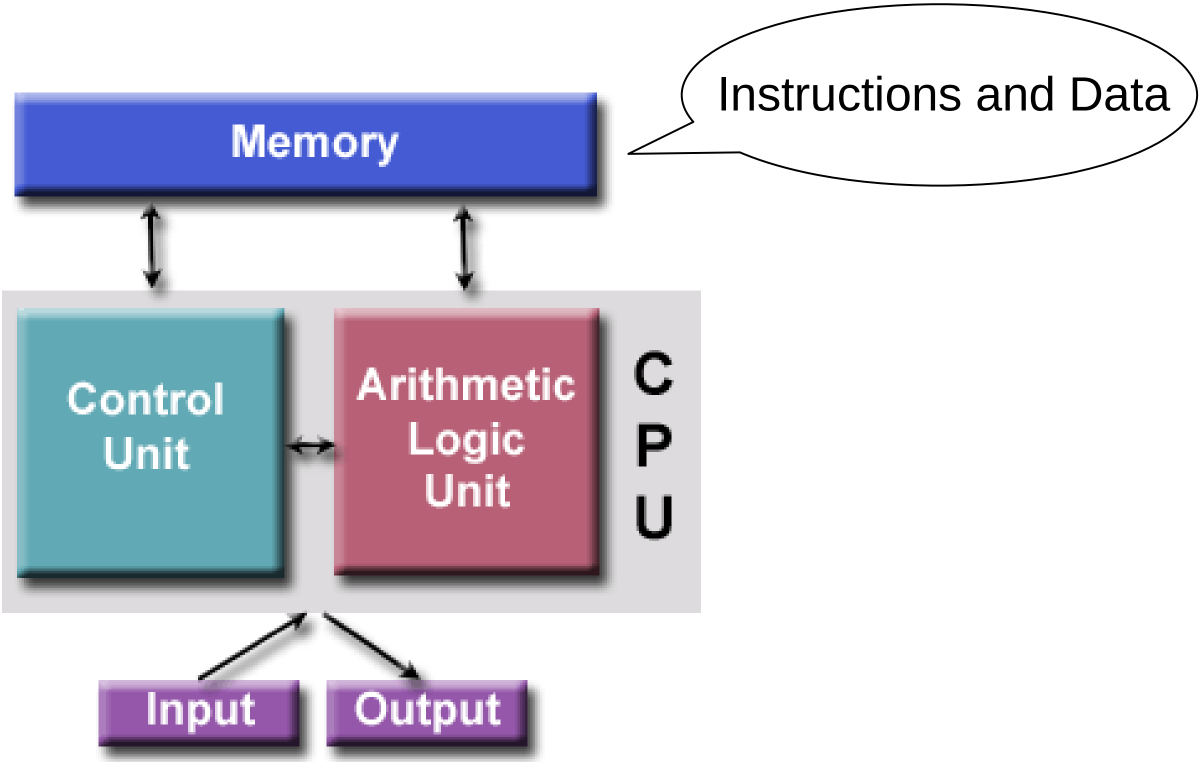
Parallel Architectures

In this lecture:

- The von Neumann Bottleneck
- Different kinds of parallel architectures, examples of their applications
- Memory in parallel architectures.

The von Neumann Architecture

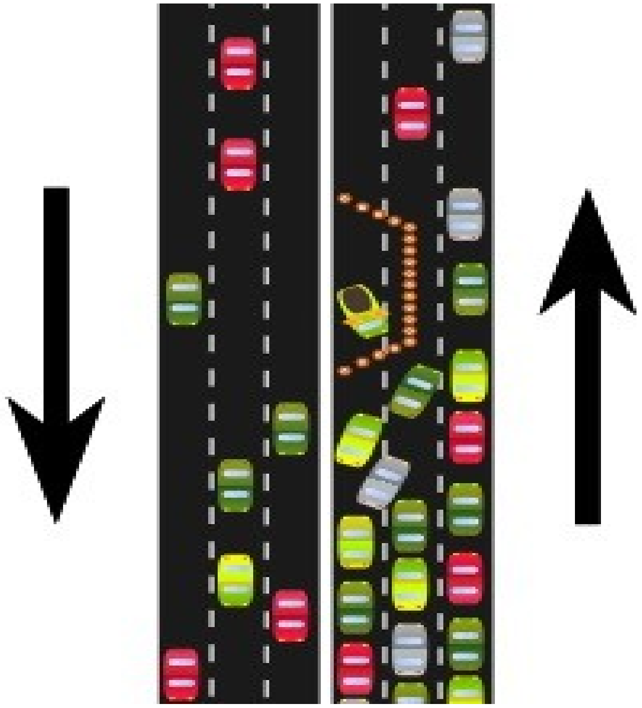
A uniprocessor, sequential architecture.



The von Neumann bottleneck: (or *memory wall*)

The speed at which data and instructions can be retrieved from memory (store) becomes a limit on the speed at which the CPU can operate.

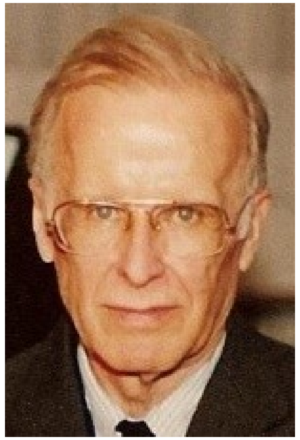
main memory access is very slow compared to the rate at which the CPU operates.



The von Neumann Bottleneck

“The shared bus between the program memory and data memory leads to the ***Von Neumann bottleneck***, the limited throughput* (data transfer rate) between the CPU and memory compared to the amount of memory. Because program memory and data memory cannot be accessed at the same time, throughput is much smaller than the rate at which the CPU can work. This seriously limits the effective processing speed when the CPU is required to perform minimal processing on large amounts of data.

The CPU is continually forced to wait for needed data to be transferred to or from memory. Since CPU speed and memory size have increased much faster than the throughput between them, the bottleneck has become more of a problem, a problem whose severity increases with every newer generation of CPU.” (from https://en.wikipedia.org/wiki/Von_Neumann_architecture)



John Backus

- The term "von Neumann bottleneck" was coined by John Backus in his 1977 ACM Turing Award lecture.

John Backus (1924-2007): invented the first widely used high-level programming language (FORTRAN) and was the inventor of the Backus-Naur form (BNF), a widely used notation to define formal language syntax (basis for compilers)

*We’ll discuss the concept of “throughput” during a future lecture

The von Neumann Bottleneck

Solutions:

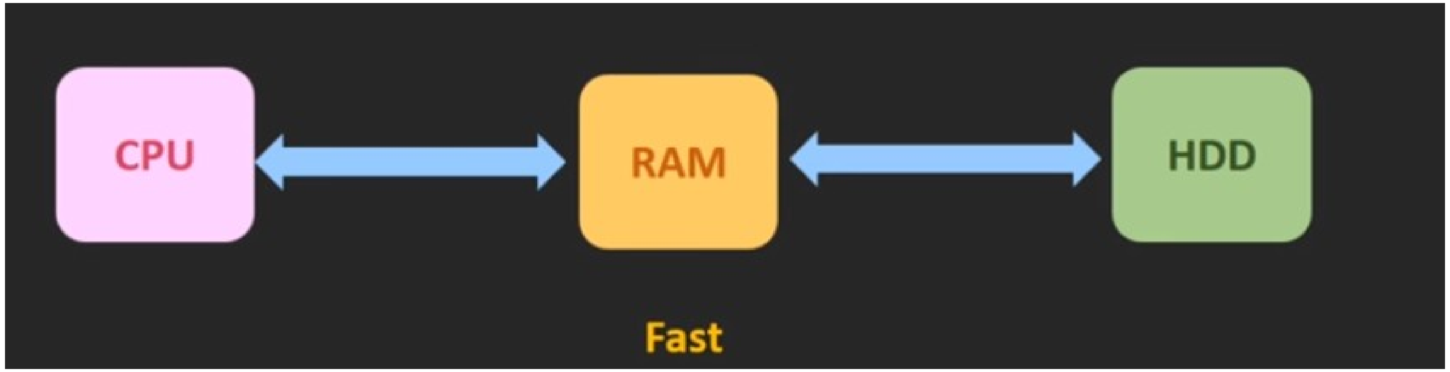
- Alternative architectures (Harvard architecture, distributed storage). ✓
- Minimize use of main memory
  - caching,
  - internal registers

The von Neumann Bottleneck

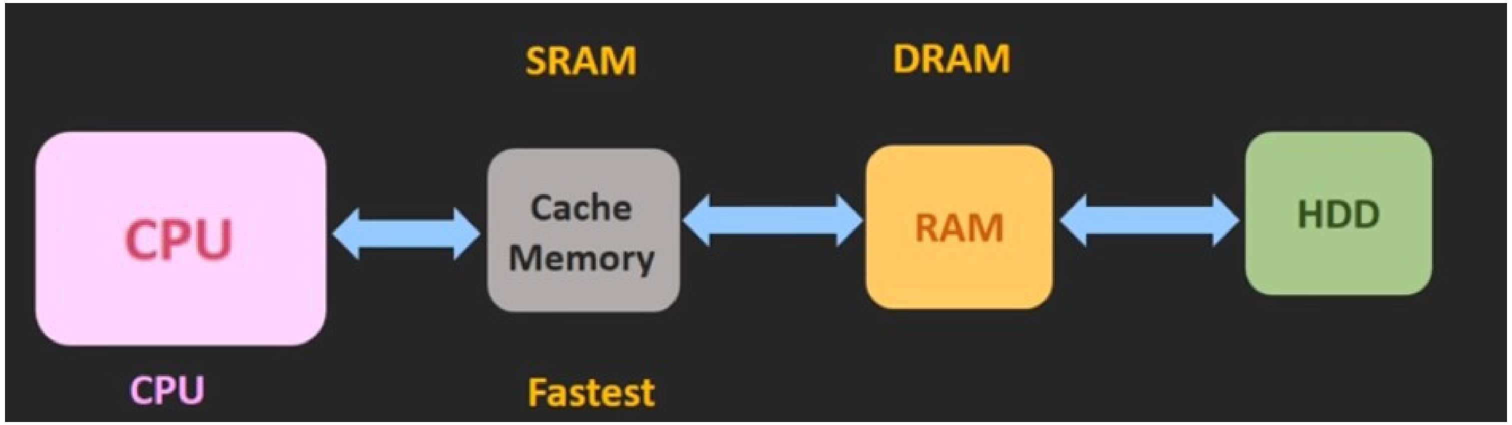
Solutions:

- Alternative architectures (Harvard architecture, distributed storage). ✓
- Minimize use of main memory
  - caching,
  - internal registers

CPU storage and memory hierarchy



Cache: a data store, duplicating some of the main memory, but capable of more rapid access (making the data used most often by the CPU instantly available.)

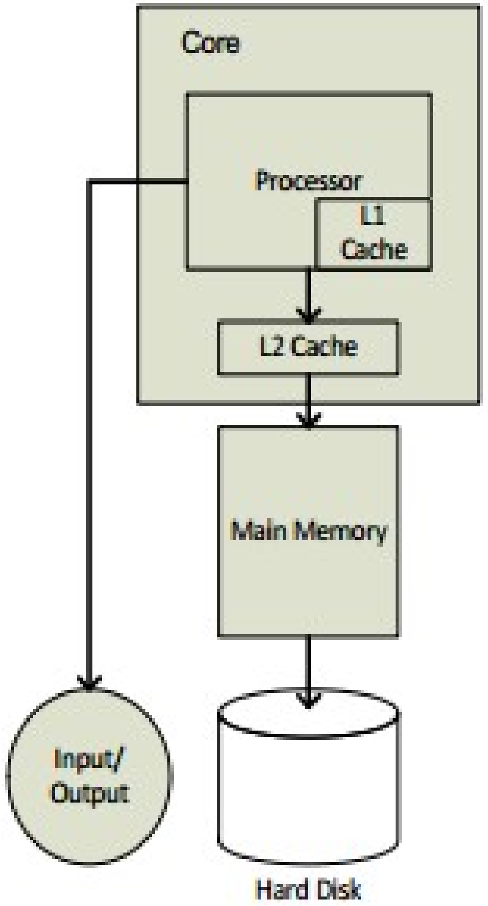


SRAM: static RAM; DRAM: dynamic RAM HDD: hard disk drive

CPU storage and memory hierarchy

Cache: a data store, duplicating some of the main memory, but capable of more rapid access (making the data used most often by the CPU instantly available.)

- Cache Level 1 (L1):
  - closest to the processor
  - this is very fast memory used to store data frequently used by the processor.
- Cache Level 2 (L2):
  - just off-chip, slower than L1 cache, but still much faster than main memory
  - L2 cache is larger than L1 cache.
- Main memory
  - larger and slower than cache



and SSD (solid state drive using flash memory)

For more details: http://en.wikipedia.org/wiki/Memory_hierarchy

.

The von Neumann Bottleneck

Solutions:

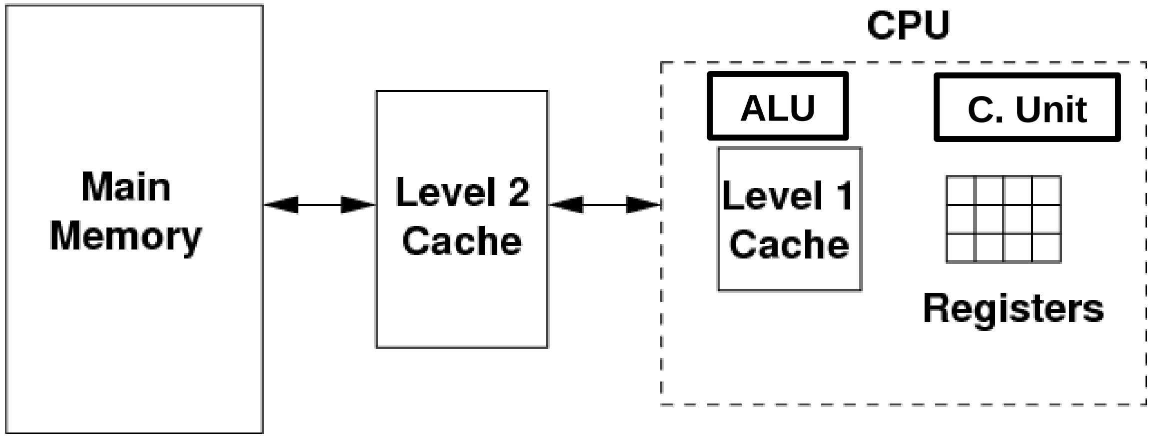
- Alternative architectures (Harvard architecture, distributed storage). ✓

- Minimize use of main memory
  - caching, ✓
  - internal registers

CPU storage

Registers

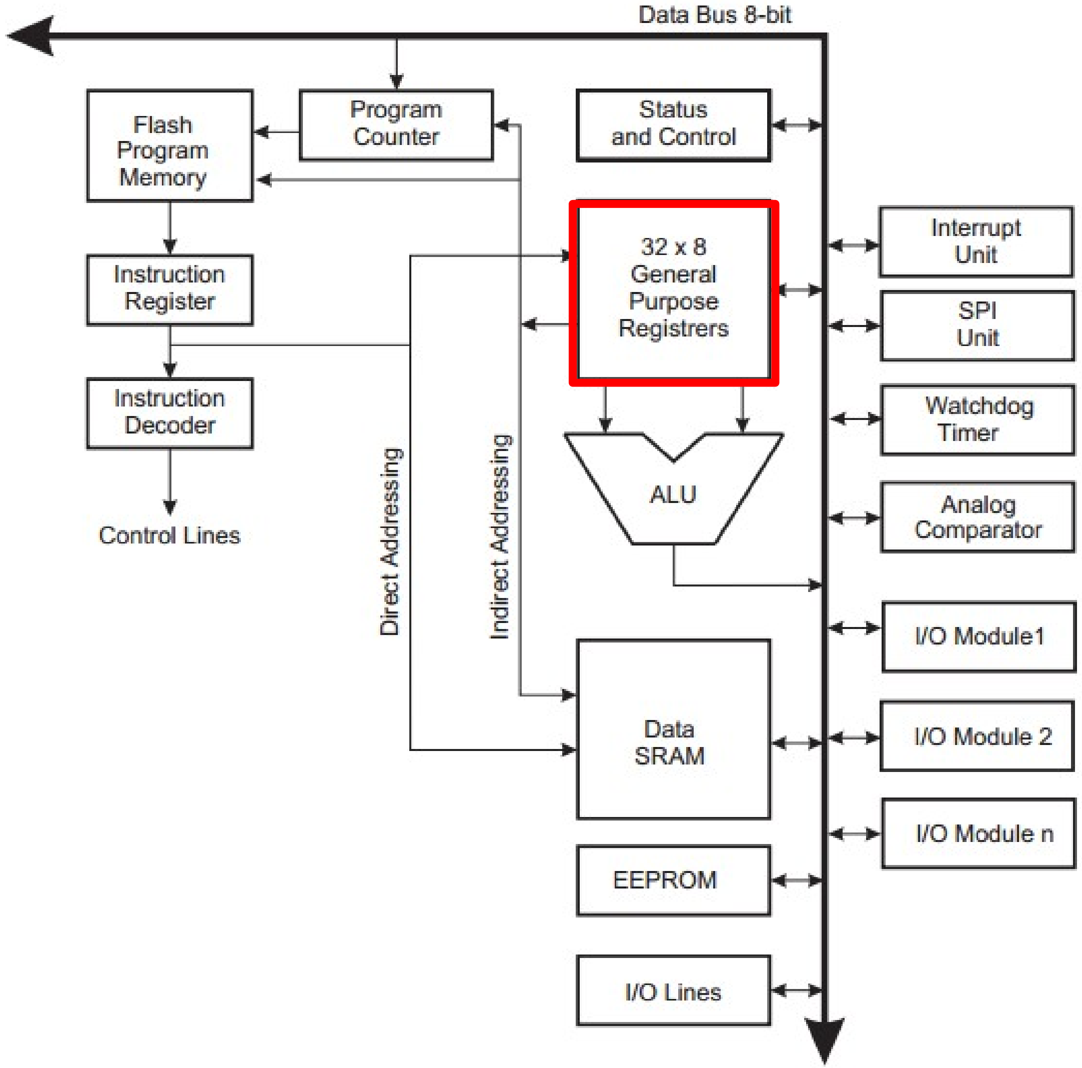
- A (rapid access) data store on the CPU store values used in ongoing computations - the fastest possible access



- Modern Programming languages do not have direct access to registers
  - exceptions: assembly and C (inline assembly)

Arduino UNO

- Microcontroller: ATmega 328
- CPU: AVR
- 32 General Purpose Registers
  - 8 bits (we'll cover this later in the course)



The von Neumann Bottleneck

Solutions:

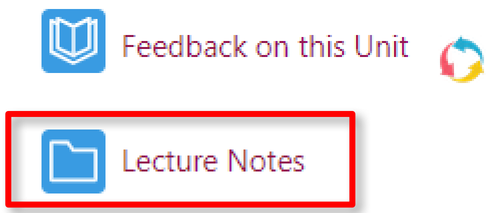
- Alternative architectures (Harvard architecture).
    - Losing flexibility

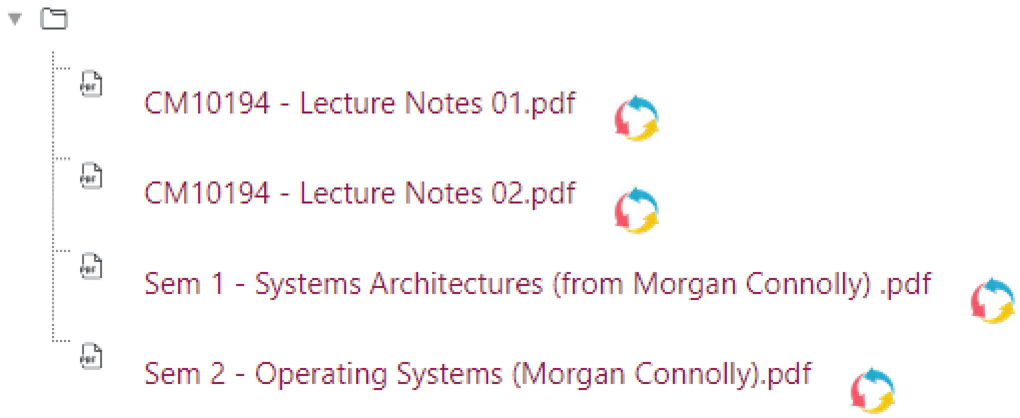
- Minimize use of main memory
    - caching,
    - internal registers

But what if we could execute multiple operations simultaneously?

Before we answer the question, some announcements

- On Moodle, Lecture Notes part 1





- Revision Exercises on Moodle
  - Revision 01 uploaded (please note, it doesn’t include week 1 content)
  - They’ll help you to see if you are following the unit. Do them individually and check answers with your classmates. Any remaining difficulties, do post on Moodle (or let me know)