General Structure of an OS

Separation of a computer in four sections

- **Hardware :**

  collection of devices that allow the execution of programs

- **OS :**

  management and coordination of system hardware

- **Software :**

  any program that can be executed inside the OS

- **User :**

  any device or being that can interact with the system

Hardware (simplified)

- Systembus connects all devices
  - one or more CPUs for program execution
  - shared memory for tasks of the CPU and other devices
  - Controller for IO devices
    - Hard Drives
    - HID
    - Network Interface
    - ...

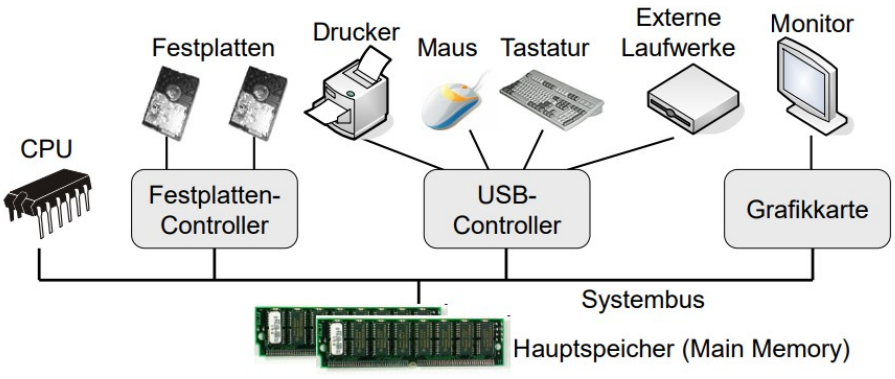


Figure 1: "Systembus Floiwchart"

Computer Architecture : von-Neumann

- reference model for computers
- separation between code execution and data

  - Separation between CPU and memory
  - Separation between Execution Unit and ALU
- this adds component communication overhead in program execution
  - Data has to be moved from memory to CPU and back, to be used
  - the OS provides functionality to use the given resources efficiently

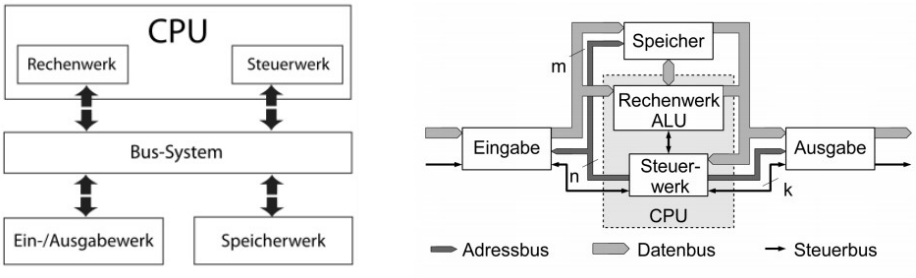
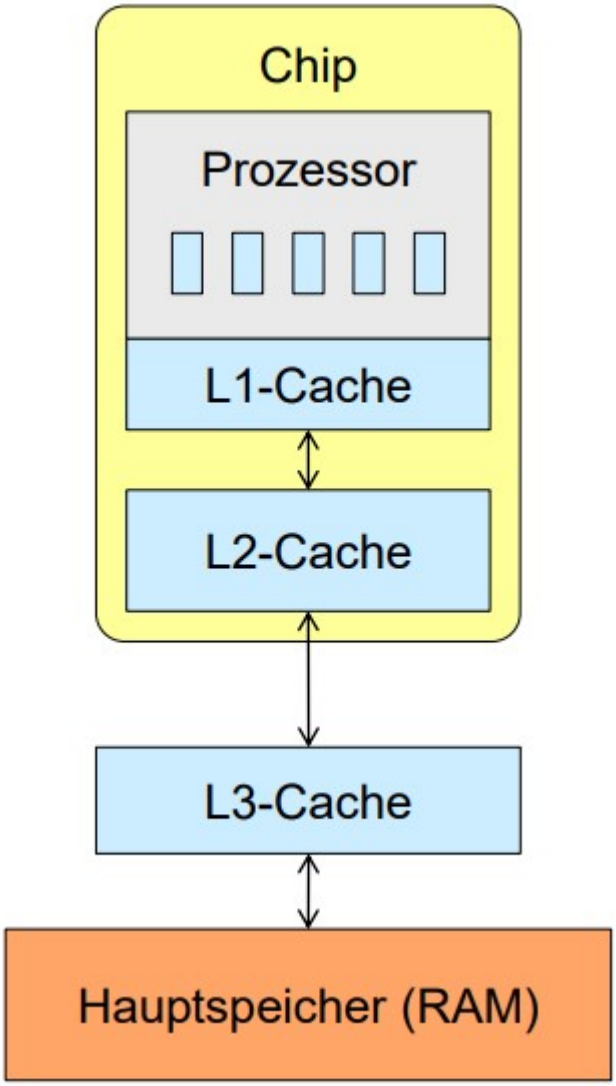


Figure 2: "CPU Model Flowchart"

- CPU has multiple registers
  - data registers, address registers, special registers, ...
- additional cache
  - fast buffer memory
  - access to cache is much faster compared to memory access
  - smaller cache = lower access times
  - caches are transparent to the OS
  - Types of cache
    - L1-cache
      - close to the main execution unit, very small, very little latency
      - saves future instructions for faste execution
    - L2-cache
      - larger and slightly slower
    - L3-cache
      - faster than main memory
      - smaller than main memory
      - extra chip outside the main processor



- registers tend to be very small, no more than the size of a DWORD but extremely fast
  - used for calculation or comparison
- cache is still very fast, but is usually slower than registers, while having a larger size
- main memory is very large, but needs many cycles to move data to CPU
  - OS needs to handle access times and data transport
  - every time we access data from a hard drive, we have to stop program execution so the processor can continue

Processor Cores and Caches

- each CPU tends to have its own L1 and L2 cache, sharing the L3 cache between all cores
- all processors can access system BUS and main memory individually
- communication and access latency is still a bottleneck in modern hardware
- Hyperthreading
  - process interweaving, so that program execution can be sped up

Hardware component interplay

- CPU executes operations
- CPU and IO-devices are used asynchronously
  - every IO-controller controls one type of device
  - the CPU is needed to execute an operation
    - every controller has its own registers
    - CPU moves data from main memory and cache
    - operation is started after moving the data
  - Today: *DMA (Direct Memory Access)*
    - seperate controller for the movement of data
    - takes load away from CPU

Simplified computer architecture

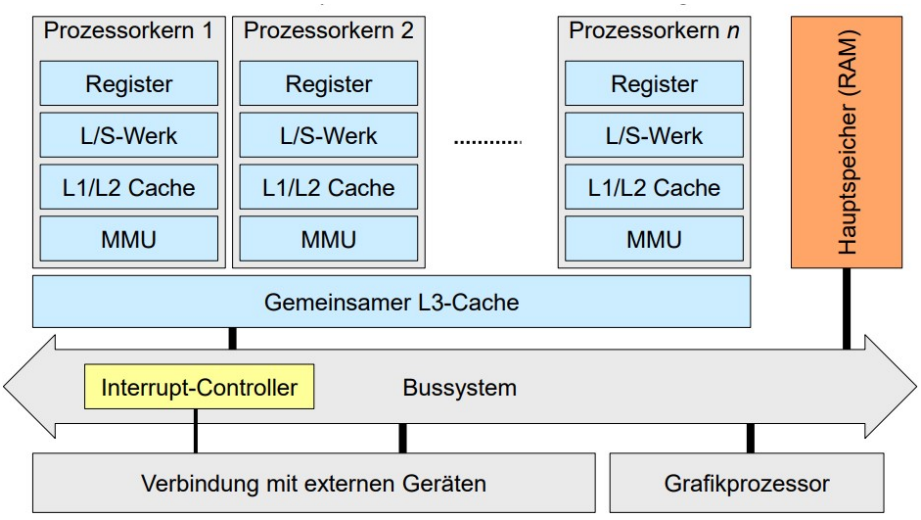


Figure 3: "Main Components of a computer"