Hardware & Software

Gökçe Aydos

Lecture

Goals

- know different classes of computers and be able to compare their performance
- know primary hardware components
- understand software as a sequence of instructions
- know different software layers and be able to recognize them

Prep Questions

- which components does a computer need to work? E.g., how does your smartphone communicate with you?
- look at your smartphone.
  - what kind of interfaces does your smartphone have?
  - what is the purpose of these interfaces?

Computers

- crucial part of our lives
- we love & hate them
- understanding how they work will hopefully reduce your anxiety and anger over these machines without emotions (yet)

Why Computers?

- humans built machines for moving or manipulating things
- how could a thinking machine look like?
    - e.g., for solving math problems

History

- mechanical calculating machines called *abacus*, e.g.,
  - yupana based on a table
  - quipu based on knotted strings
- computer referred to a human who carried out calculations
  - by 1943 most computers were human
- electronic calculator

Today

- most computers are mostly used for more than calculations
- we will call all the *programmable* devices as computers

Discussion

Where do you see computers in your current environment?

First Computers

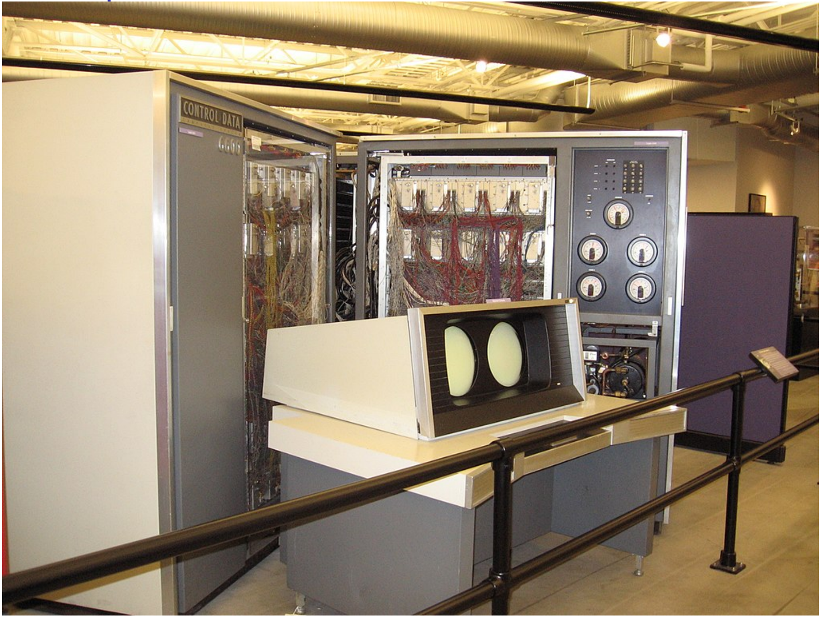

Figure 1: Samm [CC BY-SA 2.0]

Example - Tesla Model S

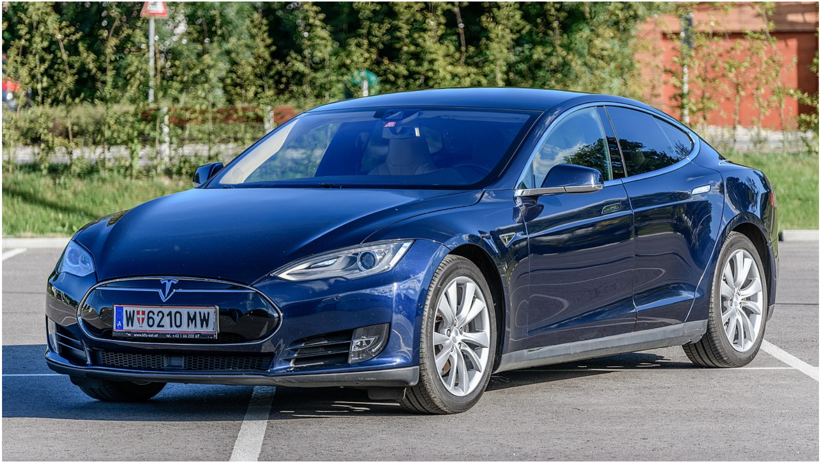

Figure 2: Granada [CC BY-SA 4.0]

Example - Tesla Model S II

- *computer on wheels*
- sensors like camera, radar, ultrasonic proximity sensors
- operational info like energy use, position of wheels, brakes, door handles, speed
- persistent internet connection, 4G, Wi-Fi
- over-the-air update support
  - the manufacturer can avoid costly recalls
- the car has an API

Example - Cattle Tracking System

- tail and neck attached device Moocall from Ireland
- calving sensor
  - measures tail movement patterns and labor contractions (Geburtswehen)
  - 4G connection
- heat sensor
  - measures if the cows are in heat

Discussion - computers examples

- why were first computers huge?
- what does the computer in a Tesla do?
- what does a computer in a Moocall calving sensor do?

Classes of computers

- supercomputer
- mainframe
- workstation
- personal computer (PC)
- smartphone
- computer in a washing machine

Ingredients for an information machine

- input
- storage
- data processing
- output

A Desktop PC Exploded

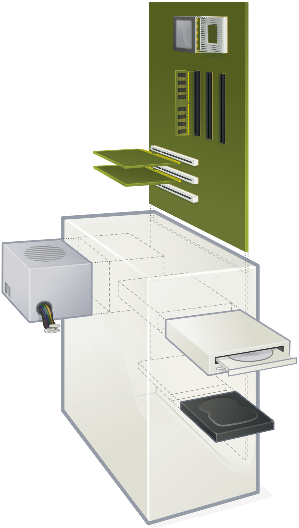

Figure 3: Gustavb [CC BY-SA 3.0]

Components of computers

```
.
              Bus
               |-> graphics card w/ GPU <-> monitor
RAM <-> CPU <->|-> USB controller       <-> mouse, keyb., e
               |-> sound controller     <-> speakers
               |-> harddisk controller  <-> harddisk
               ...
```

Computer Bus

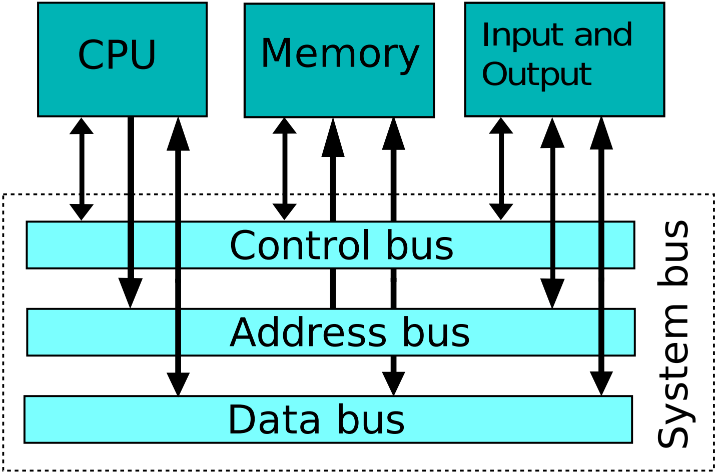

Figure 4: W Nowicki [CC BY-SA 3.0]

Bus Example - Peripheral Component Interconnect (PCI)

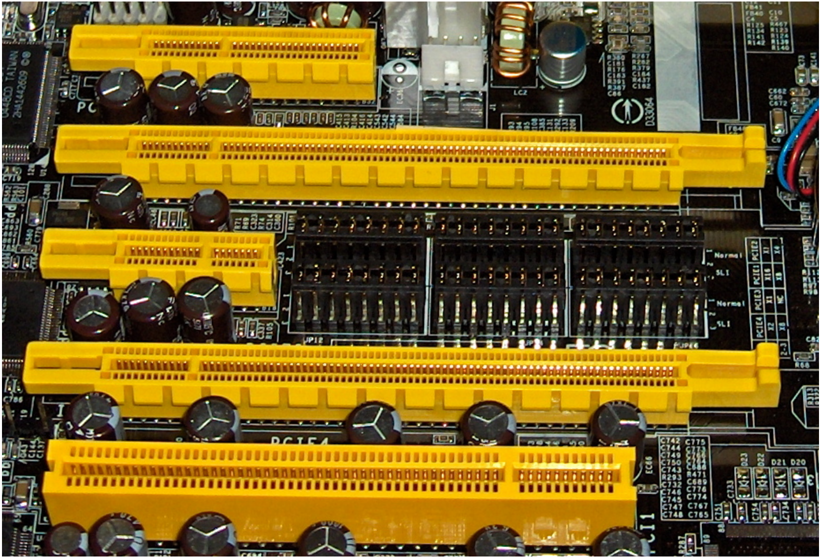

Figure 5: snickerdo [CC BY-SA 3.0]

Example - Thunderbolt 3.0

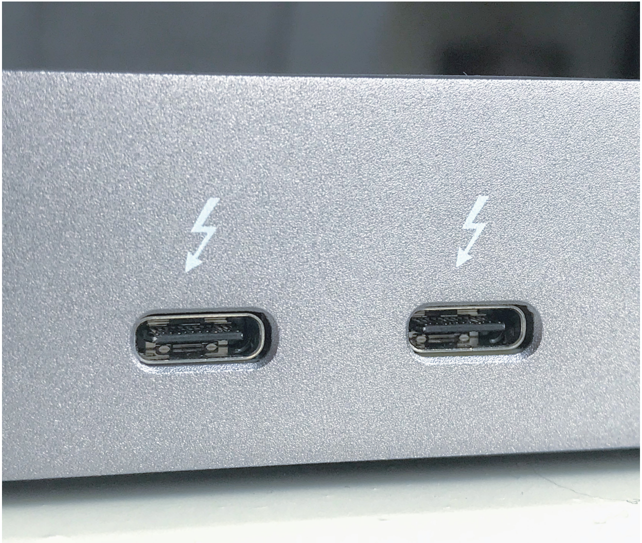

Figure 6: Amin [CC BY-SA 4.0]

Thunderbolt 3.0

includes:

- PCI
- Displayport (2x monitors)
- USB 3.1

Discussion - components

Where are input, output, processing, storage?

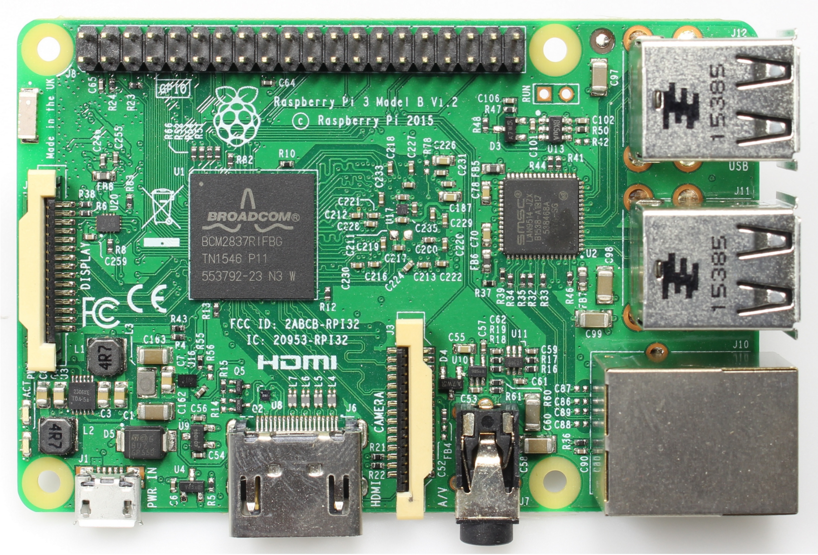

Figure 7: Make Magazin [CC BY-SA 4.0]

Discussion - components II

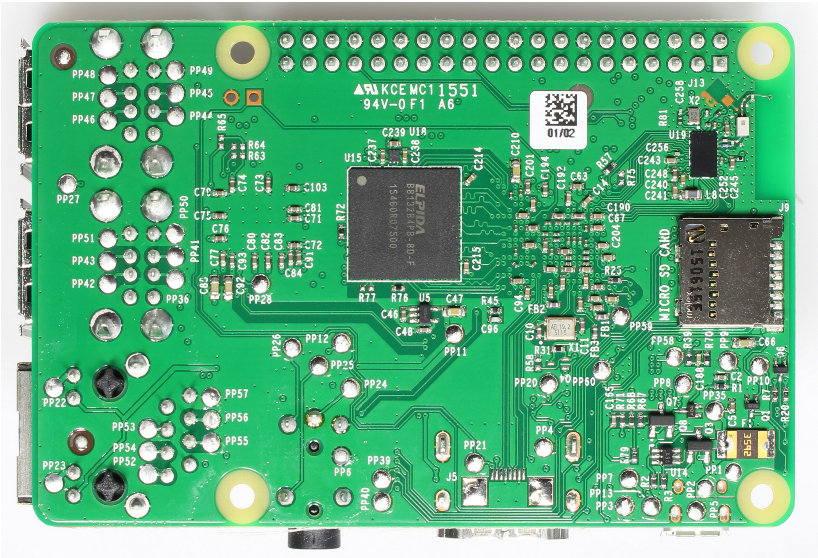

Figure 8: Make Magazin [CC BY-SA 4.0]

Input

- mouse
- microphone
- sensors, e.g., temperature, step counter

Sensors

- sensors
    - sense something..., e.g., magnetic field, your mood
    - analyzing vs sensing
    - *physical quantities* $=>$ *digital representation*

A Sensor: Accelerometer

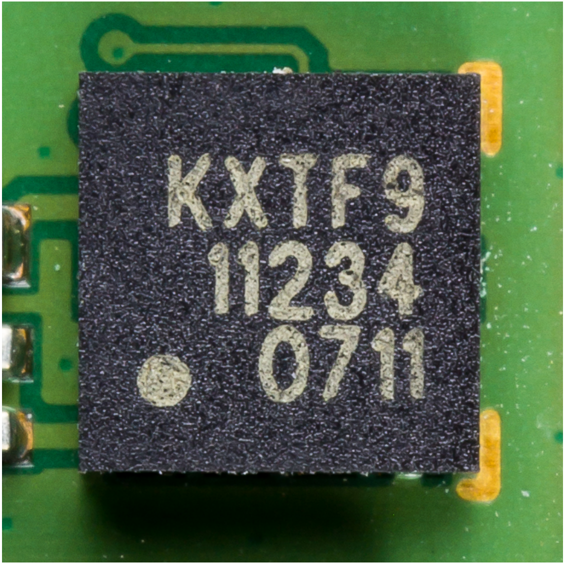

Figure 9: © Raimond Spekking [CC BY-SA 4.0]

Inside an Accelerometer

How does an accelerometer work?

Data Processing

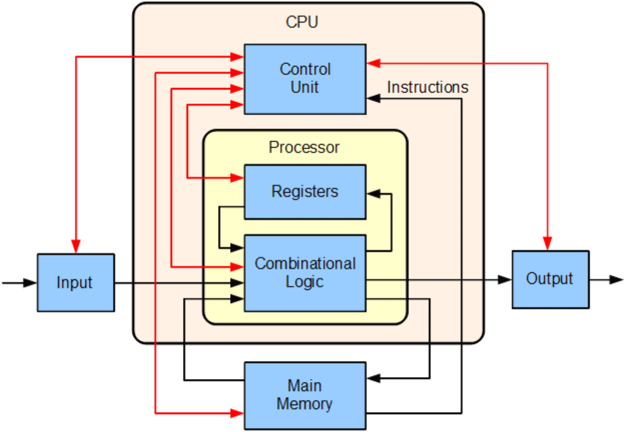

Figure 10: Lambtron [CC BY-SA 4.0]

Data Processing II

done by *central processing unit* (CPU), consists of

- *logic unit* does the processing
- *registers* nearest memory (faster than cache memory)
- *cache memory* nearer than RAM

Instruction processing

- CPU understands commands called *instructions*, e.g.:
  - add the numbers on memory cells 4 and 29, write the result on memory cell 10
  - copy the result on memory cell 10 to the memory of graphics card
- a sequence of these instructions is called *software*

Exercise - software

what does the following instruction sequence (program) do?

1. read a number from the keyboard and write it to the memory cell 1
2. read a number from the keyboard and write it to the memory cell 2
3. if the content of memory cell 1 is less than the content of memory cell 2, continue at instruction 5
4. swap the contents of memory cell 1 and 2
5. send the content of the memory cell 1 to the monitor
6. send the content of the memory cell 2 to the monitor

CPU — Server vs PC

- examples
  - a laptop processor (on Thinkpad X13 Yoga)
  - a server processor (on Dell Poweredge R730)
- discussion: how can we compare these processors?

How fast are the CPUs?

Year	Processor	frequency MHz	processor speed MIPS
1975	6502 (C64)	1	0.365
1985	80386	16	4
2005	Pentium 4	2800	~5,000
2016	Core i7	3500	~320,000

- processor frequency does not imply actual performance.
- MIPS: million instructions per second

RAM

- random access memory
  - contains instructions, data
  - analogy: *your desk* compared to your *shelf* (could be harddisk)
  - data vanishes if powered down (volatile)
- *random-access*

  data can be accessed in same amount of latency regardless of its physical location, cf. magnetic-band, -harddisk

How much RAM?

Year	typical RAM size
1975	16 kB
1985	128 kB
1995	32 MB
2005	1 GB
2015	8 GB

Data storage device

- generally referred as: *harddisk drive (HDD)*
- data persists after poweroff
- higher density compared to RAM

How much storage?

typical data storage device sizes nowadays:

data storage device	size
SSD	~1 TB
HDD	~4 TB magnetic HDD

Graphics processor (GPU)

- understands special instructions like
  - draw a line between (x1, y1) and (x2, y2)
  - rotate this triangle 60 degrees
- generally GPUs excel at 3D operations and floating point arithmetics. Typical CPUs also contain GPUs

GPU performance - FLOPS

- *floating point operations per second* (FLOPS) instead of mips
- example floating point operation:

  ```
  5 * 2^-3 + 2 * 2^-14
  ```

GPU performance comparison

Year	GPU	GFLOPS (FP32)
2019	UHD Graphics on a i5-10210U (iGPU)	∼ 500
2020	GeForce GTX 1650 (dGPU)	∼3,000
2020	GeForce RTX 3080 (dGPU)	∼25,000

some GPUs are also used in high-performance computing (e.g., big data simulations), because their higher FLOPS compared to CPUs.

Output

- computer peripherals
  - monitor
  - printer
- actuators
  - relay
  - motor

Relay

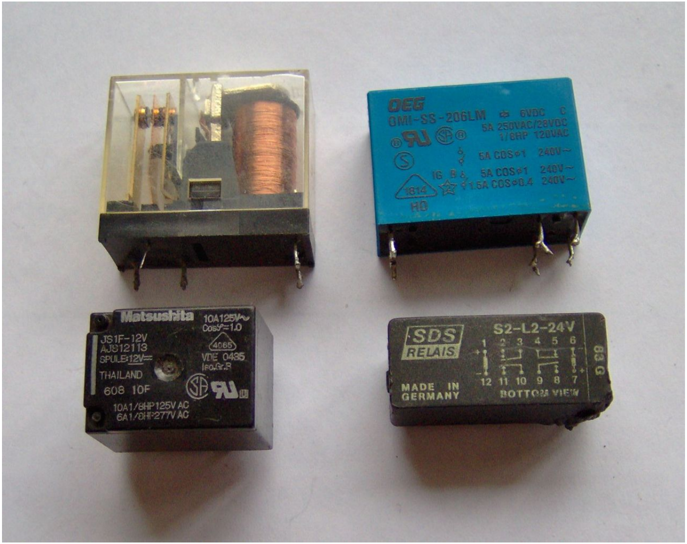

Figure 11: FDominec [CC BY-SA 3.0]

Relay II

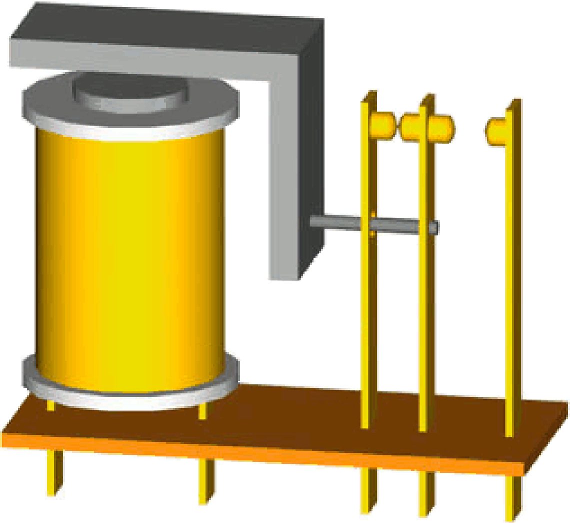

Figure 12: Digigalos [CC BY-SA 3.0]

Expansion Devices

- via USB, PCIe
- graphics card (also called GPU)
  - generates images for a monitor
  - nowadays also used for data processing acceleration
- network interface card
  - adds additional networking ports
- data processing acceleration card
  - similar to the GPU, but based on special chips for accelerating particular algorithms
  - e.g., FPGA programmable acceleration card

Popular Computer Interfaces

- USB
- PCIe
- HDMI
- Thunderbolt (USB-C)

Popular Networking Interfaces

- WLAN, WiFi
- Ethernet
- Bluetooth
- WWAN
- RFID

Need for software

for *programmability* of computers we need software

Need for software layers

```
store the result of 1+3 on memory address 4
```

What happens if another program wants to use memory address 4?

```
print 'hello' on row 17 column 0 of my screen
```

We normally do not care where the text should be printed.

=> For convenience and independence from hardware we need different software layers responsible for different tasks.

Software layers

1. BIOS or UEFI
2. operating system
3. graphical user interface
4. application

BIOS

- *basic input output system*
- tasks[1]
  - reading a keystroke from keyboard
  - writing data to disk
  - hardware settings, e.g., overclocking
  - power management through ACPI
- not used after booting

[1] https://en.wikipedia.org/wiki/BIOS#Operating_system_services>

UEFI

- *unified extensible firmware interface*
- a *mini* operating system
- replaces BIOS on modern computers
- tasks augmenting BIOS
  - secure boot
  - can run UEFI applications, e.g., UEFI shell

Operating system

- abbreviated as *OS*
- tasks
  - memory management
    memory address 0 of Firefox is not the same as *memory address 0* of Word
  - data management
    programs can conveniently access `/users/goekce/diary.txt` instead of *HDD 1, head 3, sector 429, cylinder 23, bits 947 to 27734* (HDD structure)
  - process management
    priorization of programs running in parallel

Graphical user interface

- abbreviated as *GUI*
- tasks
  - drawing windows
  - user interaction
- nowadays part of OS

Applications

- users interact with applications
- tasks
  - everything you can imagine/implement
- *should be user-friendly*
  - intuitive
  - fault-tolerant

Moore's law

- the density of transistors on chips doubles every two years
  - linked to the performance of CPUs
- the number of transistors on chips until 2018

Summary

- computer is *a programmable calculator*
- components: peripherals (input, output), processor, memory
- software is *a sequence of instructions*

Appendix

Storage types

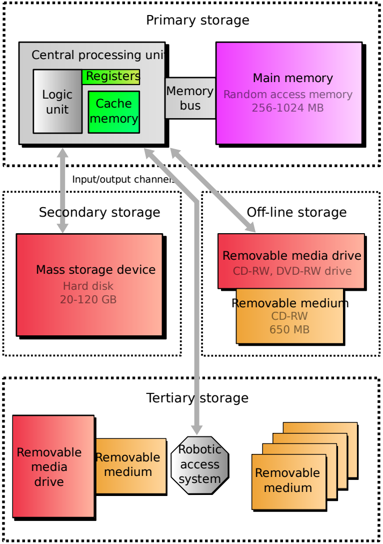

Figure 13: I, Surachit [CC BY-SA 3.0]

Image is outdated, e.g., we have nowadays 8G-16G RAM on PCs

Main Memory

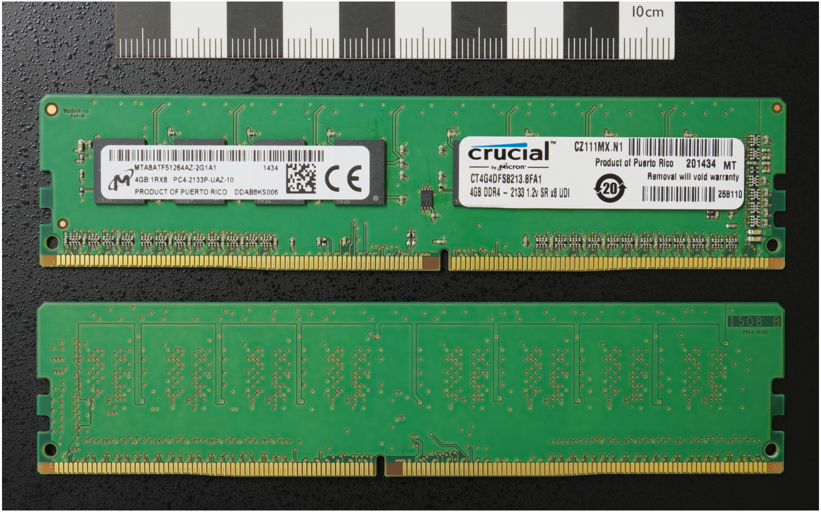
10cm
MTA8ATF51264AZ-2G1A1
1434
4GB 1RX8 PC4-2133P-UAZ-10
PRODUCT OF PUERTO RICO DDAB8KS006
crucial
by Micron
CZ111MX.N1
Product of Puerto Rico 201434 MT
Removal will void warranty
CT4G4DFS8213.8FA1
4GB DDR4 - 2133 1.2v SR x8 UDI
20
268110
1508 B

Figure 14: smial [FAL]

Main Memory II

- also called RAM (random access memory)
  - where does the word 'random' come from?
  - PC, Laptop *>=8GB*
  - smartphone *>=3GB*
  - high-performance computing server *>=64GB*
    - e.g., Dell Poweredge R730

High-performance Computing

- abbreviated as *HPC*
  - also called *supercomputers*
- for memory- and processing-extensive applications
  - gene-sequence matching
  - ray-tracing, ...
- example: HPC cluster on THD
  - HPC cluster on THD (university login req.)

Non-volatile Storage

- for immediate access
  - hard disk
- for physical data exchange
  - CD, Blu-ray, USB-Stick
- for archiving (long-term data storage)
  - Magnetic tape
  - optical disk archive

Tape library for archiving

- example:
  - https://www.dkrz.de/systems/datenarchiv