Chapter 2

Choosing a specialized CPU
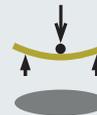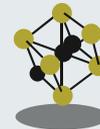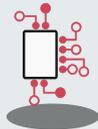
ENSI
CAEN

ÉCOLE PUBLIQUE D'INGÉNIEURS
CENTRE DE RECHERCHE

2021-2022

The objective of an application is to fulfill specifications (or requirements).

COMPUTER
Software layers model

(web,
players a/v, etc) qqGb
APPLICATIONS

Java, C, Python, C++, C#
.NET, JavaScript, SQL, PHP
Objective-C, Go, etc

OPERATING SYSTEM qq100Mb
(shell, std services, compilers, etc)

C, assembly

KERNEL qqMb
(scheduler, memory mngt, drivers, network, etc)

HARDWARE
(CPU, memory, peripherals, etc)

Electronic system (silicon industry)
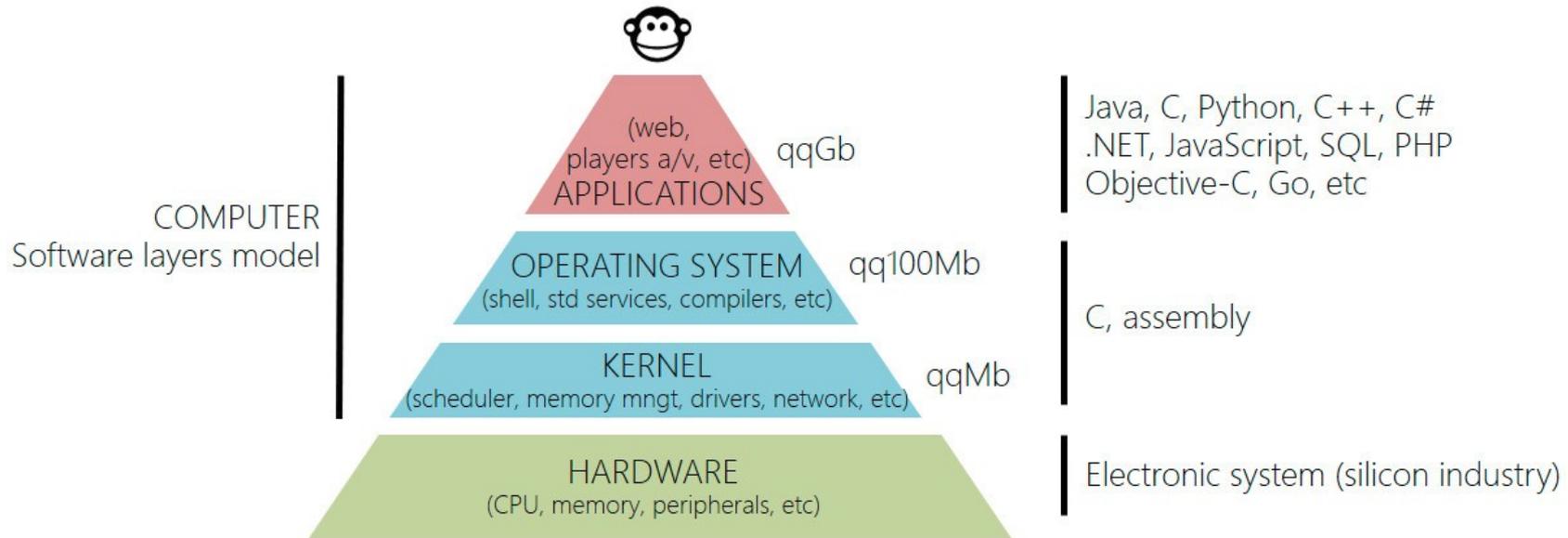
About 90 % of the time, the processing consists of a **simple supervision**.
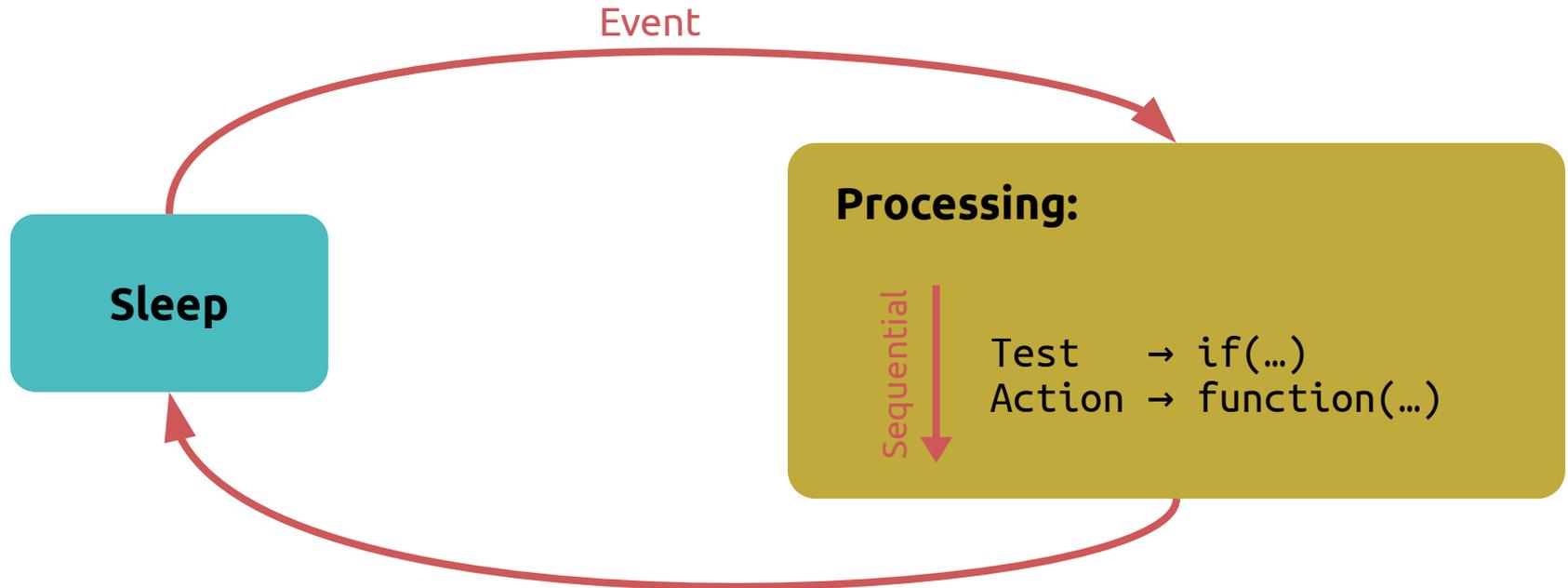
→ Opt for MCU, AP or GPP architectures

From time to time the function to process might be an algorithm,
i.e. apply a processing to a certain amount of data (information).

Algorithm examples: search, sort, digital signal processing (audio, radar, comms, ...), ...

The first choice of processor should always be a general-purpose processor.

However if it does not match the specifications, it is wise to switch to a processing-specialized architecture so that we can:

- Reduce the processing time

- Reduce the code size and/or its memory footprint

Note that switching to a specialized processor should be justified with measurements.
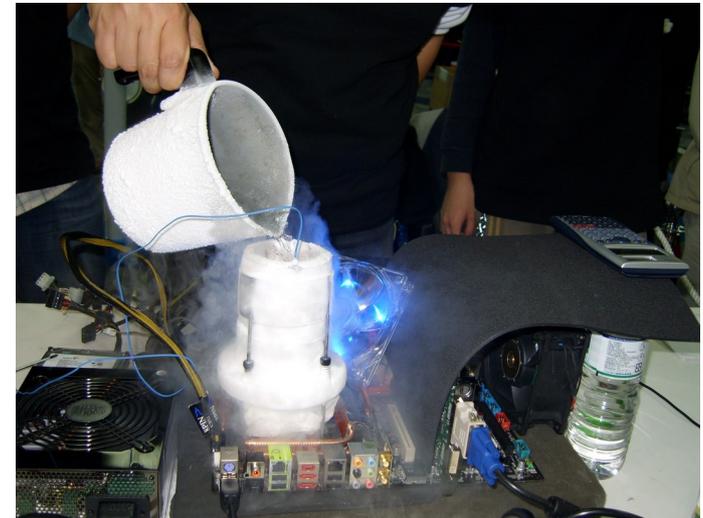
DFT algorithm example

Take for example the DFT algorithm:

For ONE frequency sample

$$S(k) = \sum_{n=0}^{N-1} s(n) \times e^{-j 2 \pi k \frac{n}{N}}$$

Sum   Product $\left.\begin{array}{l}\\ \\\end{array}\right\}$ Sum Of Product (SOP)
Multiply-Accumulate (MAC)

- Each product is independent from another
- → Parallelism available!
- Same for the processing every single frequency sample
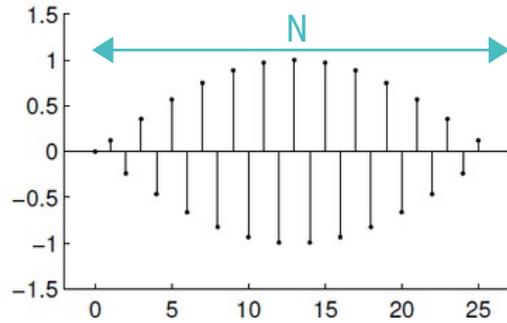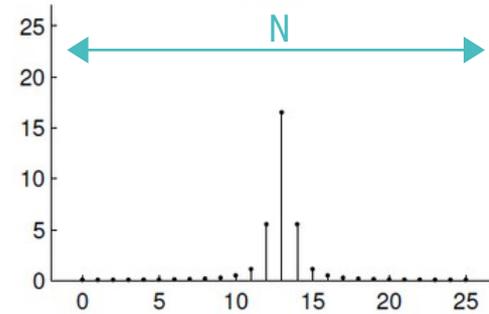
DFT algorithm example
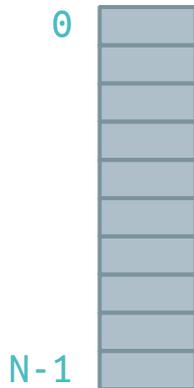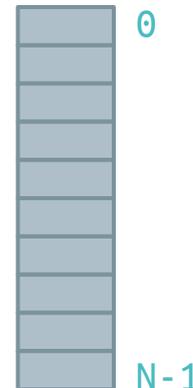
Finally, choose the CPU according to your needs.

DSP: low-power, low-cost, very low-level development (C, asm)

GPU: high-power, high-cost, high-level development (C++, OpenMP, Cuda, ...), high-parallelism potential

MPPA: Massively Parallel Processor Array, not widespread yet, but huge potential (dispatch cores to specific algorithms).