

# High Performance Programming

## SIMD – MMX/SSE/AVX

FISE2-INFO2

Guillaume MULLER

April 11, 2021

# Why having HPP lectures in FISE2? Why HPP?

# Why having HPP lectures in FISE2? Why HPP?

Classical approach to HPP = distribute on cluster

- What if the initial code is inefficient?
- $\Rightarrow$  optimize locally first!

# Why having HPP lectures in FISE2? Why HPP?

Classical approach to HPP = distribute on cluster

- What if the initial code is inefficient?
- $\Rightarrow$  optimize locally first!

Current machines already are massively parallel

- Multi-Core, Multi-Thread...

# Why having HPP lectures in FISE2? Why HPP?

Classical approach to HPP = distribute on cluster

- What if the initial code is inefficient?
- $\Rightarrow$  optimize locally first!

Current machines already are massively parallel

- Multi-Core, Multi-Thread...

A large part of optimizations can not be treated automatically

- $\Rightarrow$  impossible to rely on tools written by others
- $\Rightarrow$  as (future) engineers in CS: mandatory knowledge

# Why SIMD/MMX/SSE/AVX?

## Task Parallelism

- Execute coarse-grained pieces of code on  $\neq$  pieces of hardware
- Multi-Cores, **Multi-Threading**...

# Why SIMD/MMX/SSE/AVX?

## Task Parallelism

- Execute coarse-grained pieces of code on  $\neq$  pieces of hardware
- Multi-Cores, **Multi-Threading**...

## Instruction/Data Parallelism

- Tiny pieces of code/data simultaneously on = hardware
- **SIMD**

# Why SIMD/MMX/SSE/AVX?

## Task Parallelism

- Execute coarse-grained pieces of code on  $\neq$  pieces of hardware
- Multi-Cores, **Multi-Threading**...

## Instruction/Data Parallelism

- Tiny pieces of code/data simultaneously on = hardware
- **SIMD**

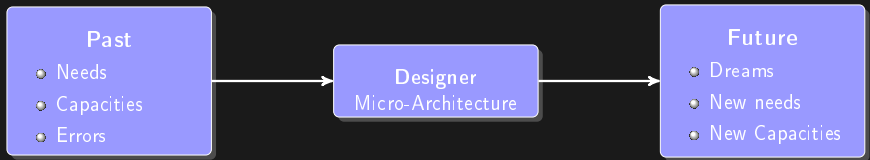
- *Who has already used/programmed a processor  $\neq$  Intel?*



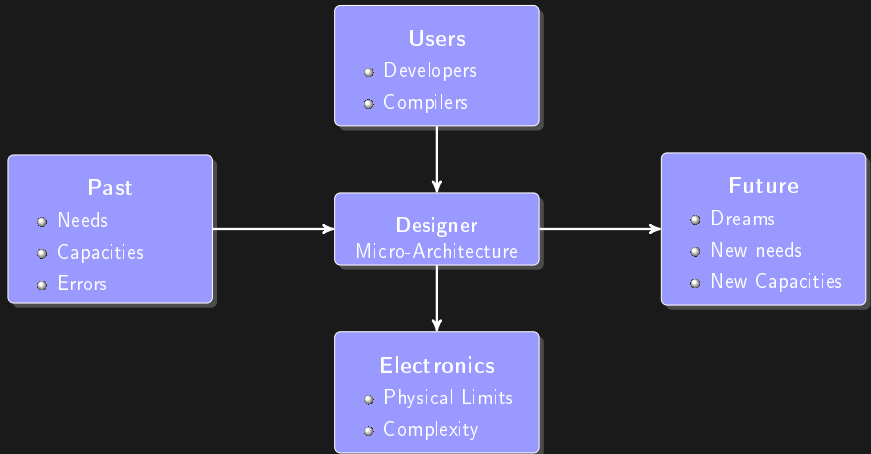
"Ça dépend" ("ça dépasse"?)

Designer  
Micro-Architecture

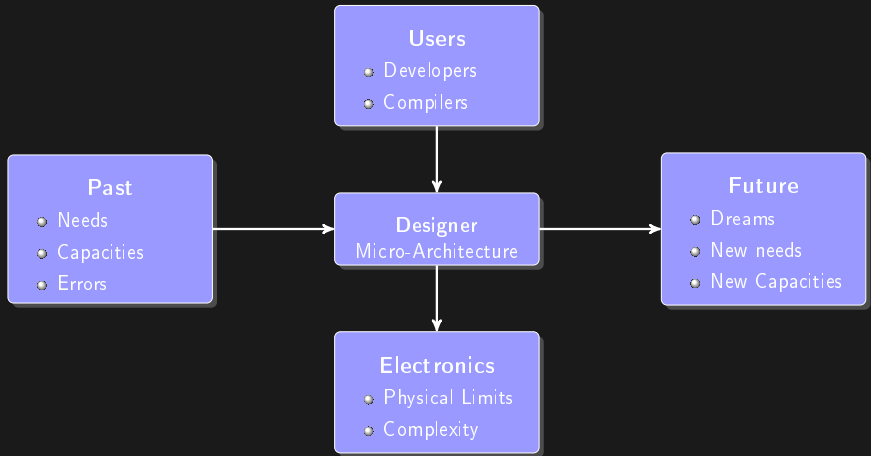
# "Ça dépend" ("ça dépasse"?)



# "Ça dépend" ("ça dépasse"?)



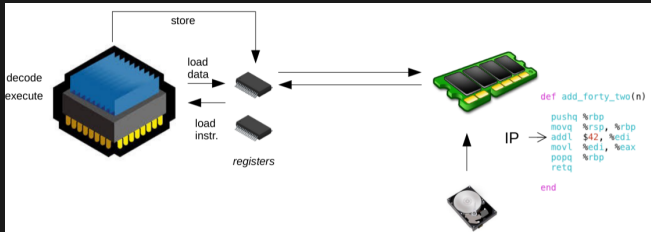
# "Ça dépend" ("ça dépasse"?)



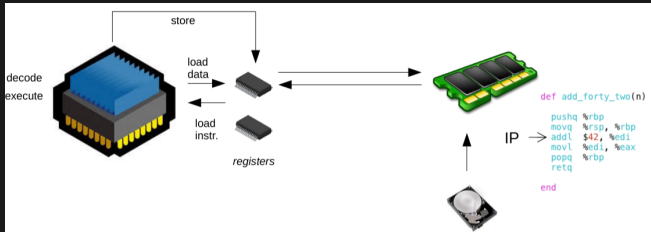
Typical answer in this course

"ça dépend"

# "Reminder"



# "Reminder"



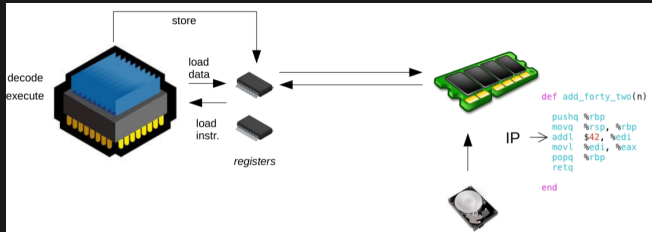
## Instruction Types

**Flow** IP / JMP ...

**Mem** LD / ST ...

**Calc** ADD / SUB / MULT / DIV ...

# "Reminder"



## Instruction Types

**Flow** IP / JMP ...

**Mem** LD / ST ...

**Calc** ADD / SUB / MULT / DIV ...

## Instruction Cycle

- Fetch
- Decode
- Execute
- ...