Proteus

An Extensible RISC-V Core for Hardware Extensions

Marton Bognar, Job Noorman, Frank Piessens (imec-DistriNet, KU Leuven)

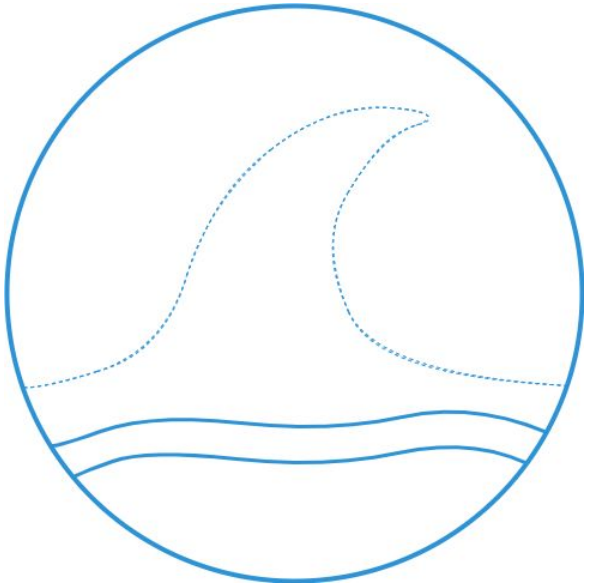
WiP

RISC-V

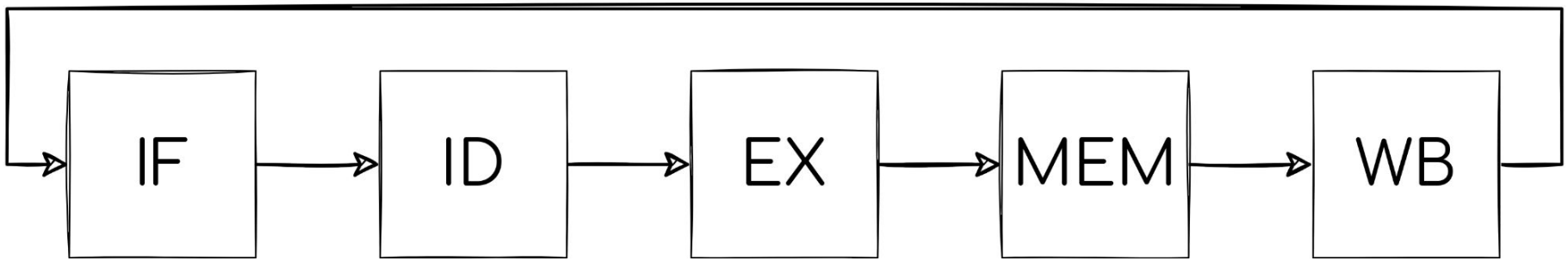
- Easy-to-extend ISA
  - New instructions
  - New control and status registers
  - Also official extensions: SIMD, vector operations, hypervisor, …
- Let’s build easy-to-extend hardware as well!

Proteus

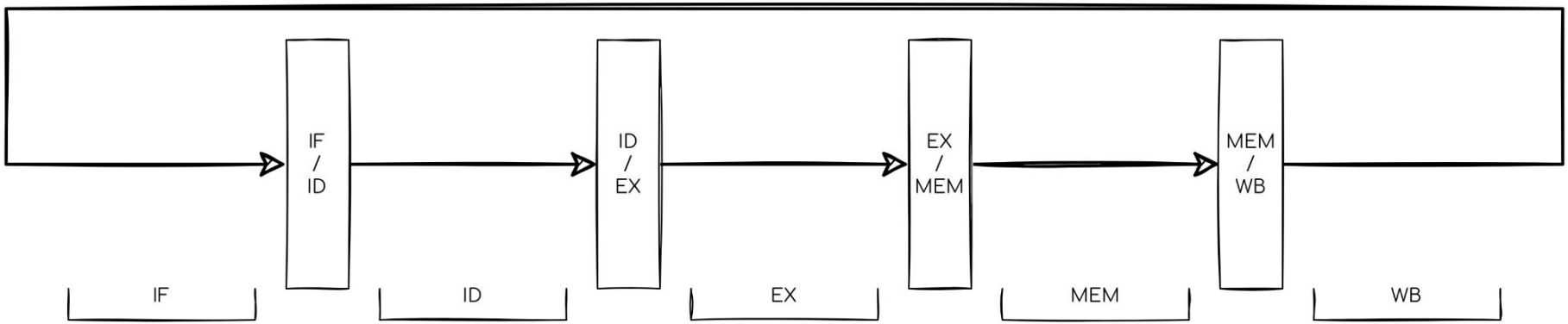
- Free and open-source RISC-V softcore
- Focus on extensibility, ease of development
- Written in SpinalHDL
- Based on a plugin system (💕 VexRiscv)
- Implementing RV32IM + Zicsr
- Comes with two implementations
  - In-order pipeline
  - Out-of-order pipeline
- Both passing the official RISC-V unit tests
- Newlib board support package



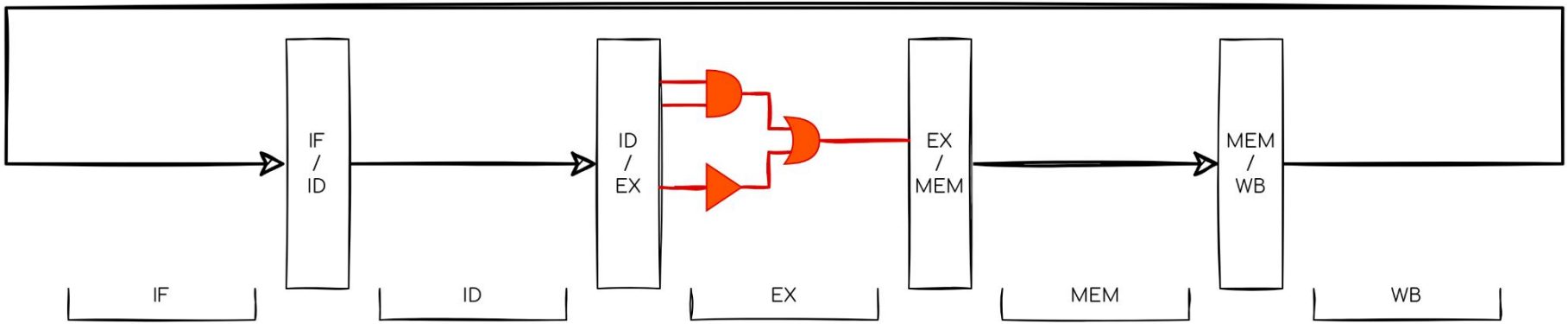
In-order pipeline



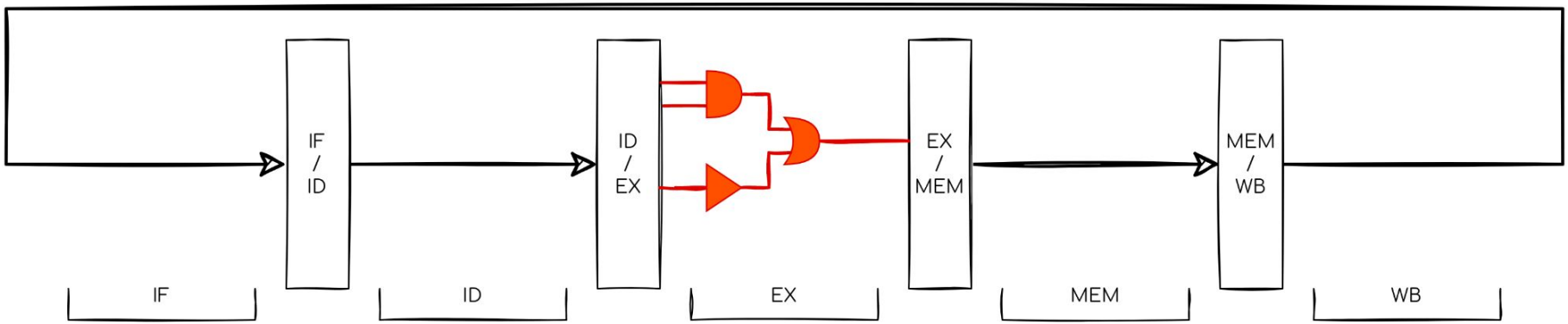
In-order pipeline



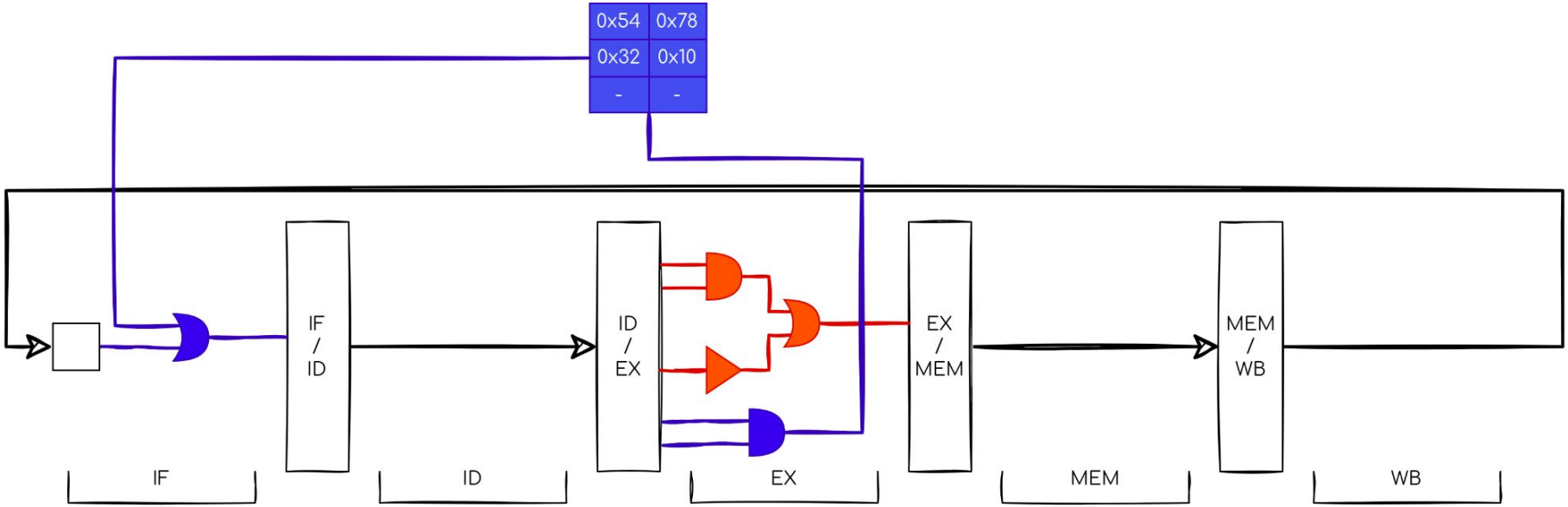
In-order pipeline + plugins



In-order pipeline + plugins

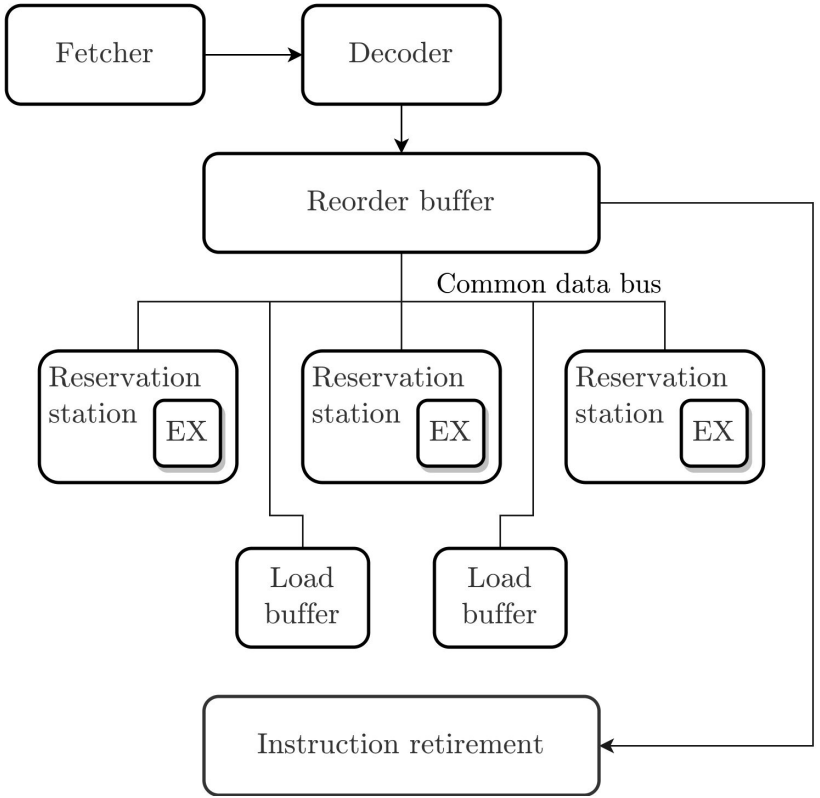



```
val op = input(Data.ALU_OP)
switch(op) {
  is (AluOp.SUB) {
    result := src1 - src2
  }
}
when (input(Data.ALU_COMMIT_RESULT)) {
  output(pipeline.data.RD_DATA) := result
  output(pipeline.data.RD_DATA_VALID) := True
}
```


0x54
0x78
0x32
0x10
-
-
IF / ID
ID / EX
EX / MEM
MEM / WB
IF
ID
EX
MEM
WB

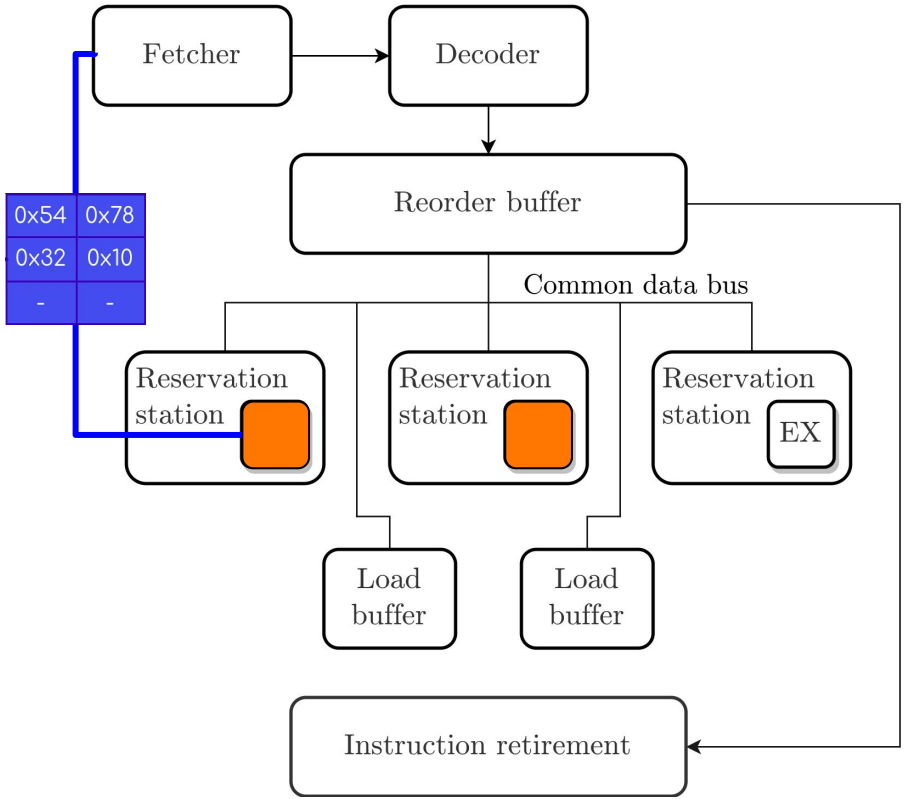
Out-of-order pipeline

- Tomasulo's algorithm
- Textbook implementation, but close to real hardware
- Reusing many of the in-order plugins
- Configurable
  - Number of reorder buffer entries
  - Number of reservation stations
  - Types of execution units
  - Branch predictor

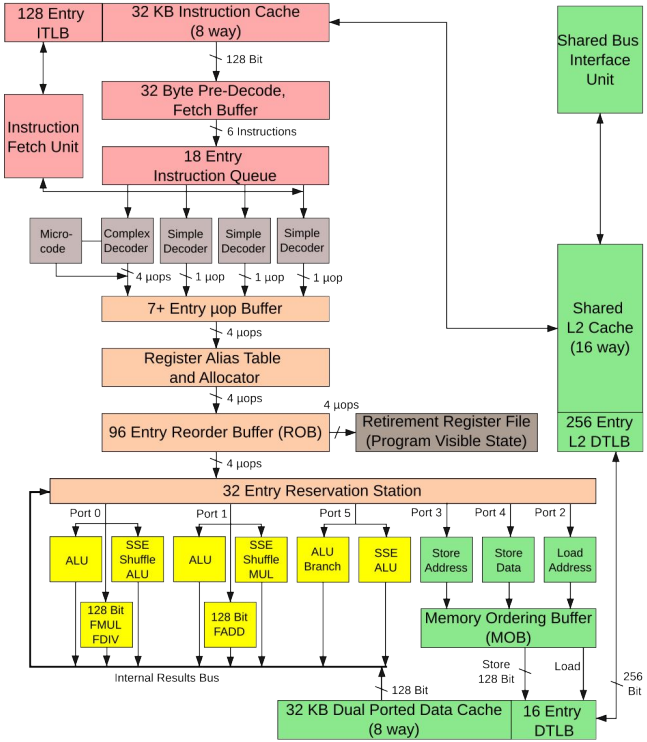


Out-of-order pipeline

- Tomasulo's algorithm
- Textbook implementation, but close to real hardware
- Reusing many of the in-order plugins
- Configurable
  - Number of reorder buffer entries
  - Number of reservation stations
  - Types of execution units
  - Branch predictor

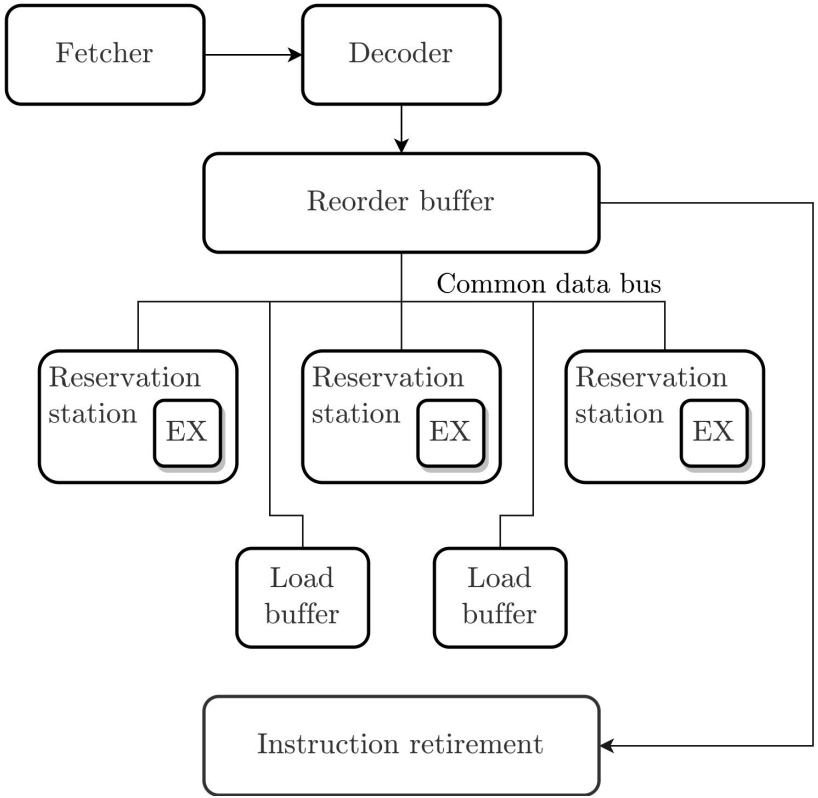


Real-world relevance



Intel Core 2 Architecture

https://commons.wikimedia.org/wiki/File:Intel_Core2_arch.svg



Extensions

- Swapping out certain plugins:
  - New branch prediction algorithm
- Writing new plugins, adding extra functionality:
  - In-order pipeline: CHERI, CHERI-TrEE (hardware capabilities)
  - Out-of-order pipeline: ProSpeCT (secure speculation) → See ORSHIN talk on Friday!
  - Non-security proposals
- Future work:
  - Improving the modularity of the out-of-order pipeline → Gaining more experience!
  - Experimenting with performance, optimizing for FPGA/ASIC
  - Booting Linux

Get involved!

https://github.com/proteus-core

- Docker playground
- RISC-V unit tests
- Synthesis instructions
- Newlib BSP
- Example code
- Extensions

✉️ marton.bognar@kuleuven.be