KU LEUVEN

DEPARTMENT OF COMPUTER SCIENCE

Proteus: An Extensible RISC-V Core for Hardware Extensions

Marton Bognar, Job Noorman, Frank Piessens
imec-DistriNet, KU Leuven, Belgium
marton.bognar@kuleuven.be

Proteus in a nutshell

Motivation:
- RISC-V provides an easy-to-extend ISA
- Proteus provides an easy-to-extend hardware implementation!

Main properties:
- Free and open-source RISC-V softcore
- Focus on extensibility, ease of development
- Written in SpinalHDL
- Plugin system inspired by VexRiscv
- RV32IM + Zicsr, passing the official RISC-V unit tests
- Includes a Newlib board support package

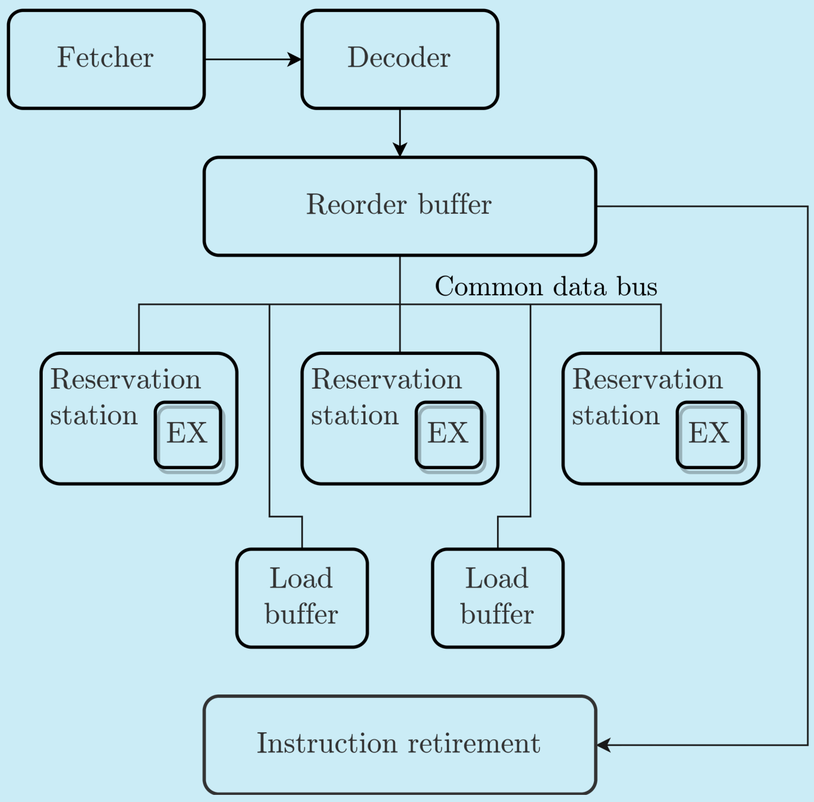
SpinalHDL

- Domain-specific language embedded in Scala
  - Provides a higher abstraction level compared to other HDLs
  - Speeds up development
- Compiles to Verilog

```
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
}
```

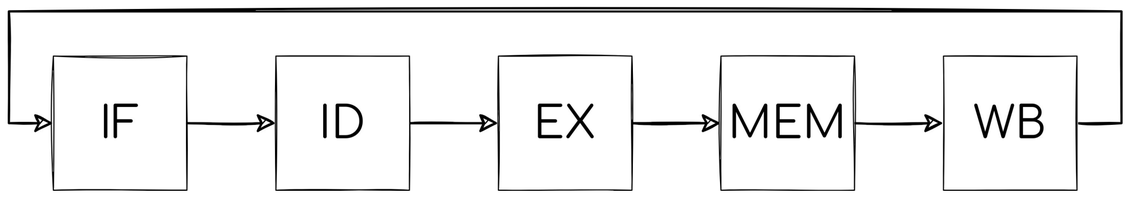
Out-of-order pipeline

- Implements Tomasulo's algorithm
  - Textbook implementation
  - Still close to real hardware
- Reuses many of the plugins used in the in-order pipeline
- Configurable in many settings
  - Number of reorder buffer entries
  - Number of reservation stations
  - Types of execution units
  - Branch prediction algorithm

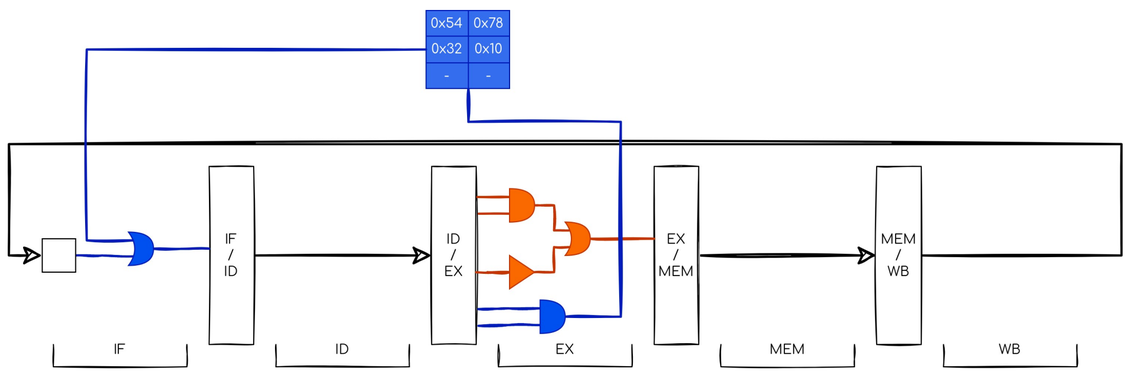


In-order pipeline + plugins

The in-order pipeline consists of a sequence of stages
- Number of stages and their functionality can be configured
- Functionality of stages is provided by plugins



Plugins provide logic between pipeline registers (and across the pipeline):



Orange plugin: arithmetic and logic unit, plugging into the execute stage
Blue plugin: branch predictor, plugging into the execute and fetch stages

Resources

GitHub organization: https://github.com/proteus-core

- Docker environment
- RISC-V unit tests
- Instructions for synthesis
- Newlib board support package
- Example code
- Extensions

Extended abstract: https://mici.hu/papers/bognar23proteus.pdf

Extensions on Proteus

Multiple published security architectures extending Proteus
- Capability Hardware Enhanced RISC Instructions (CHERI)
- CHERI-TrEE: Flexible enclaves on capability machines (EuroS&P '23)
- ProSpeCT: Provably Secure Speculation for the Constant-Time Policy (USENIX Security '23)
- Architectural Mimicry (S&P '24)

We are also interested in working on non-security extensions!

References

SpinalHDL: https://github.com/SpinalHDL/SpinalHDL
VexRiscv: https://github.com/SpinalHDL/VexRiscv

This research was partially funded by the ORSHIN project (Horizon Europe grant agreement No. 101070008).

KU Leuven
Department of Computer Science
Celestijnenlaan 200A box 2402
3001 Leuven, Belgium

DistriNet