

[master](#) 1 Branch

0 Tags

 Go to file[Code](#)

nbdd0121 Use vendor script to vendor lowrisc_ip 9bd00ad · 9 months ago 413 Commits

.github Consolidate report action into CI workflow 9 months ago

doc Clarify that FD are supported 2 years ago

flows Use vendor script from CI 2 years ago

ip Clean up redundant file watchers 9 months ago

test Fix the constant used to set mcountinhhibit 4 years ago

util Include vendor script and vendor hjson 9 months ago

vendor Use vendor script to vendor lowrisc_ip 9 months ago

.gitignore Reorganise READMEs and build instructions 5 years ago

CLA Initial commit 6 years ago

CONTRIBUTING.md Initial commit 6 years ago

LICENSE Initial commit 6 years ago

Makefile Remove hardcoded fusesoc path 9 months ago

README.md Add a reference to the demo SoC in the ready 4 years ago

check tool requirements.core Vendor in necessary modules for fusesoc 6 years ago

Muntjac

About

64-bit multicore Linux-capable RISC-V processor

Readme

Apache-2.0 license

Contributing

Activity

Custom properties

105 stars

9 watching

13 forks

Report repository

Releases

No releases published

Packages

No packages published

Contributors 5

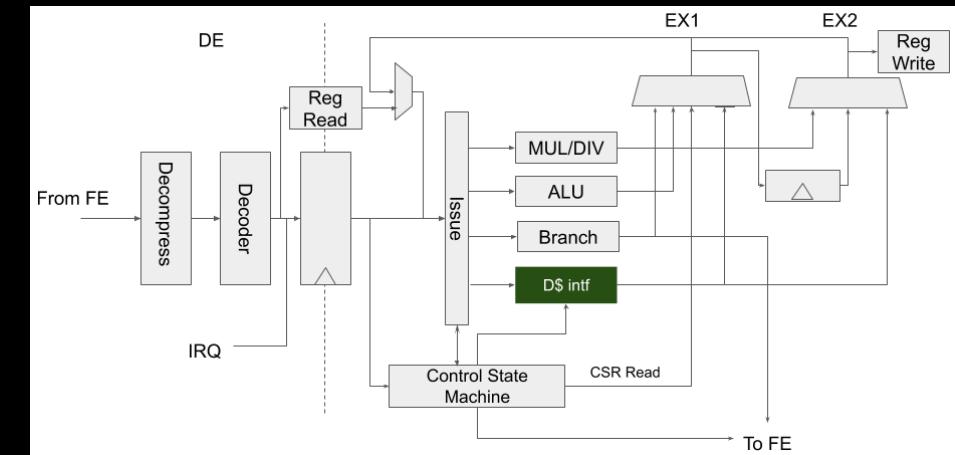
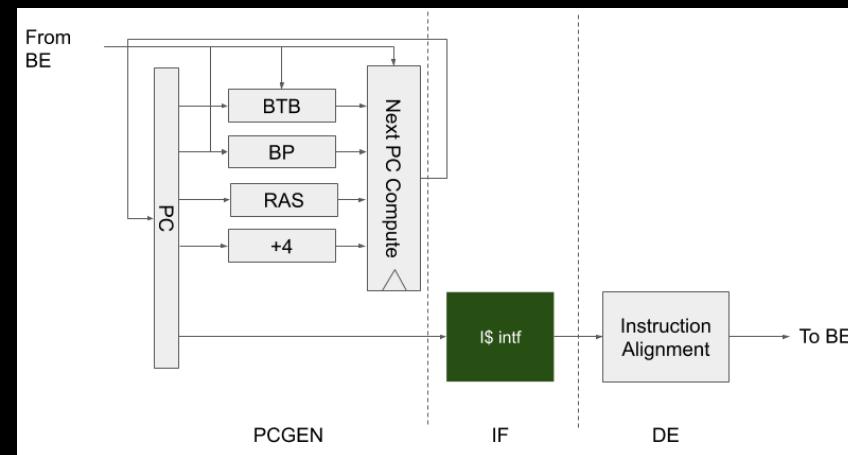
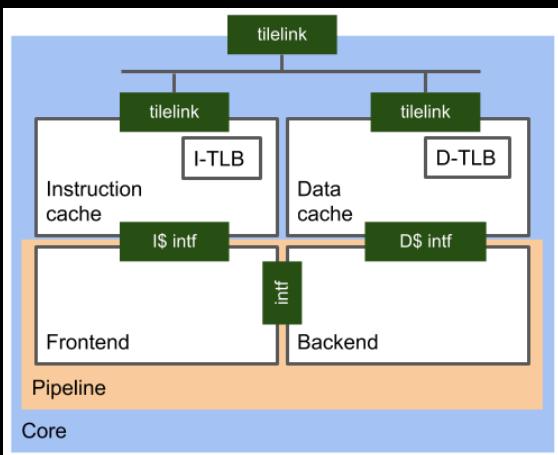


What? What is this?

- Minimal 64-bit RISC-V multicore processor that's easy to understand, verify, and extend.
- Focuses on clean, well-tested design which others can build upon and further customise
 - SoC example also provided in another repo
- Correctness > performance
 - Baseline design for educational, academic, or real-world use.
- By lowRISC, non-profit company in Cambridge, UK
 - Works with University of Cambridge, other academic and corporate partners

Architecture

- RV64IMAC (and optional FD for floats and such)
- Supports U/S/M privilege modes and Sv39 virtual addressing
- 5-stage pipeline
- Modular architecture for easy extension and modification



Core

Front-end

Back-end

Repo

- Written in SystemVerilog
- Licensed with Apache 2.0 (quite permissive)
- Supports Verilator for simulation and Vivado for FPGA synthesis
- Also uses FuseSOC and Edalize for package management, some tests written in python, and C++ code for Verilator
 - 17.7k lines of SV, 1.7k lines of C/C++, 500 lines of python
- Has build instructions for Verilator sim
 - We tried, and failed