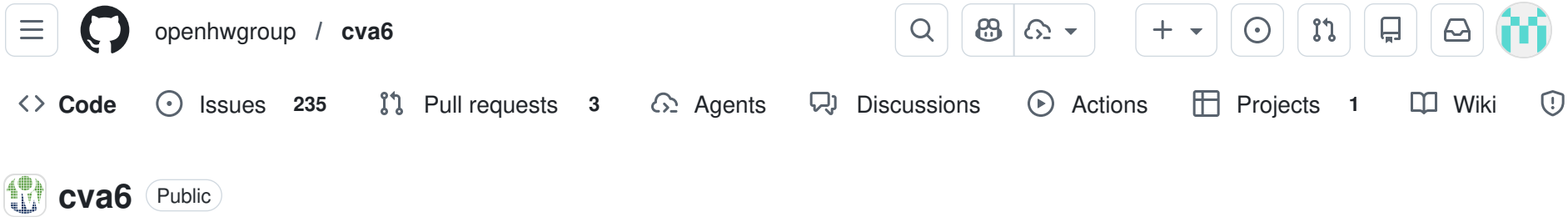

openhwgroup / cva6
Code
Issues 235
Pull requests 3
Agents
Discussions
Actions
Projects 1
Wiki
cva6
Public

JeanRochCoulon Ask for permission rights on PR bra…

338cb95 · 2 days ago

.github	pulp-actions: bump ver…	4 days ago
.gitlab-ci	Fix packet_type_o port …	3 days ago
ci	Remove edatools repo…	10 months ago
common/local/util	Fix FPU writeback log …	3 months ago
config	Fix typos (#3029)	8 months ago
core	Fix a bug that causes t…	2 days ago
corev_apu	Fix packet_type_o port …	3 days ago
docs	Updating programmers'…	2 days ago
pd/synth	Fix gate simulation: Up…	2 years ago
perf-model	RVFI-based flamegrap…	2 weeks ago
spyglass	Increase W123 warnin…	3 weeks ago
tutorials	Agilex hps load image f…	2 months ago
util	create patch to add cva…	3 months ago
vendor	Add big endian support…	3 weeks ago

About

The CORE-V CVA6 is a highly configurable, 6-stage RISC-V core for both application and embedded applications. Application class configurations are capable of booting Linux.

docs.openhwgroup.org/projects/…

#asic #cpu #fpga #risc-v #systemverilog-hdl #rv64gc #ariane

Readme

Unknown and 2 other licenses found

Contributing

Cite this repository

Activity

Custom properties

2.8k stars

92 watching

883 forks

Report repository

Releases 9

CVA6 v5.3.0 Latest

verif	Instruction Tracing (#3…	3 months ago
.editorconfig	Small SoC modifications	8 years ago
.gitignore	CV32A60X ISA (#2922)	10 months ago
.gitlab-ci.yml	Disable cvxif job in .gitl…	3 weeks ago
.gitmodules	Vendorize AXI atomics.…	last month
.readthedocs.yaml	fix `023e67e` (define PAT…	last year
ACKNOWLEDG…	Replacing KDT with Ch…	3 years ago
Bender.yml	Adding core config pkg …	3 weeks ago
CHANGELOG.md	Fix typos (#3029)	8 months ago
CITATION.cff	Clean-up README.md…	3 years ago
CODEOWNERS	CODEOWNERS: remo…	11 months ago
CONTRIBUTIN…	Ask for permission right…	2 days ago
Flist.ariane	Add ALU to ALU bypas…	6 months ago
LICENSE	Add SolderPad Hardwa…	9 years ago
LICENSE.Berkeley	chipyard: Add condition…	6 years ago
LICENSE.SiFive	✨ Add SiFive debug rom	8 years ago
Makefile	Vendorize AXI atomics.…	last month

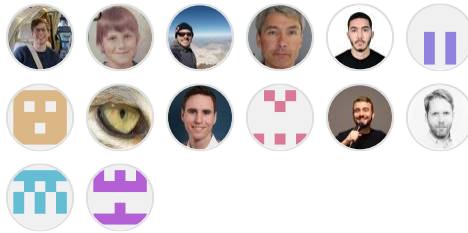
on Feb 3, 2025

+ 8 releases

Packages

No packages published

Contributors 184



+ 170 contributors

Deployments 500+

github-pages 2 days ago

+ more deployments

Languages

- **Assembly** 55.4%
- **SystemVerilog** 25.5%
- **Tcl** 7.7%
- **C** 3.8%
- **Python** 2.9%
- **Shell** 1.2%
- **Other** 3.5%

README.md	Instruction Tracing (#3…	3 months ago
RESOURCES.md	doc(RESOURCES): ad…	3 days ago
ariane.core	integrate unified mmu …	2 years ago
src_files.yml	bp: add BHT with privat…	last year
verilator_config.vlt	removing lint warnings.…	3 years ago

README Contributing License BSD-3-Clause license Apache-2.0 license

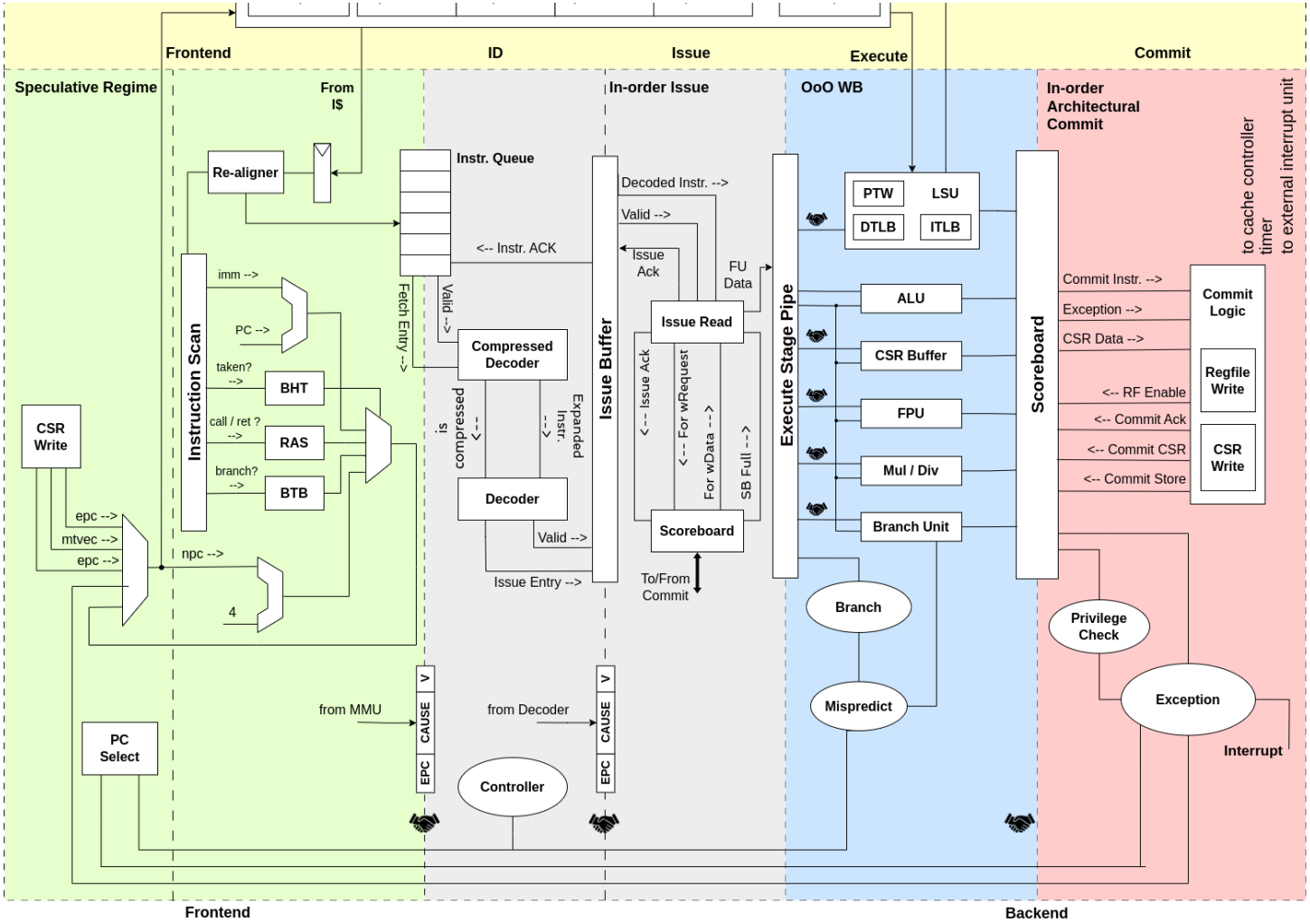
CVA6 RISC-V CPU

CVA6 is a 6-stage, single-issue, in-order CPU which implements the 64-bit RISC-V instruction set. It fully implements I, M, A and C extensions as specified in Volume I: User-Level ISA V 2.3 as well as the draft privilege extension 1.10. It implements three privilege levels M, S, U to fully support a Unix-like operating system. Furthermore, it is compliant to the draft external debug spec 0.13.

It has a configurable size, separate TLBs, a hardware PTW and branch-prediction (branch target buffer and branch history table). The primary design goal was on reducing critical path length.

The CVA6 core is part of a vivid ecosystem. In this document, we gather pointers to this ecosystem (building blocks, designs, partners...).

A performance model of CVA6 is available in the `perf-model/` folder of this repository. It can be used to investigate performance-related micro-architecture changes.



Quick setup

The following instructions will allow you to compile and run a Verilator model of the CVA6 APU (which instantiates the CVA6 core) within the CVA6 APU testbench (corev_apu/tb).

Throughout all build and simulations scripts executions, you can use the environment variable `NUM_JOBS` to set the number of concurrent jobs launched by `make` :

- if left undefined, `NUM_JOBS` will default to 1, resulting in a sequential execution of `make` jobs;
- when setting `NUM_JOBS` to an explicit value, it is recommended not to exceed 2/3 of the total number of virtual cores available on your system.

1. Checkout the repository and initialize all submodules.

```
git clone https://github.com/openhwgroup/cva6.git
cd cva6
git submodule update --init --recursive
```

2. Install the GCC Toolchain build prerequisites then the toolchain itself.

⚠️ It is **strongly recommended** to use the toolchain built with the provided scripts.

3. Install `cmake` , version 3.14 or higher.

4. Set the RISCV environment variable.

```
export RISCV=/path/to/toolchain/installation/directory
```

5. Install `help2man` and `device-tree-compiler` packages.

For Debian-based Linux distributions, run :

```
sudo apt-get install help2man device-tree-compiler
```

6. Install the riscv-dv requirements:

```
pip3 install -r verif/sim/dv/requirements.txt
```

7. Run these commands to install a custom Spike and Verilator (i.e. these versions must be used to simulate the CVA6) and these tests suites.

```
# DV_SIMULATORS is detailed in the next section
export DV_SIMULATORS=veri-testharness,spike
bash verif/regress/smoke-tests.sh
```

Tutorials

- **Running Simulations**
- **ASIC Implementation**
- **FPGA Implementation and running an OS**
- **Instruction Tracing**

Directory Structure

The directory structure separates the CVA6 RISC-V CPU core from the CORE-V-APU FPGA Emulation Platform. Files, directories and submodules under `cva6` are for the core *only* and should not have any dependencies on the APU. Files, directories and submodules under `corev_apu` are for the FPGA Emulation platform. The CVA6 core can be compiled stand-alone, and obviously the APU is dependent on the core.

The top-level directories of this repo:

- **ci**: Scriptware for CI.
- **common**: Source code used by both the CVA6 Core and the COREV APU. Subdirectories from here are `local` for common files that are hosted in this repo and `submodules` that are hosted in

other repos.

- **core**: Source code for the CVA6 Core only. There should be no sources in this directory used to build anything other than the CVA6 core.
- **corev_apu**: Source code for the CVA6 APU, exclusive of the CVA6 core. There should be no sources in this directory used to build the CVA6 core.
- **docs**: Documentation.
- **pd**: Example and CI scripts to synthesis CVA6.
- **util**: General utility scriptware.
- **vendor**: Third-party IP maintained outside the repository.
- **verif**: Verification environment for the CVA6. The verification files shared with other cores are in the core-v-verif repository on GitHub. core-v-verif is defined as a cva6 submodule.

verif Directories

- **bsp**: board support package for test-programs compiled/assembled/linked for the CVA6. This BSP is used by both `core` testbench and `uvmt_cva6` UVM verification environment.
- **regress**: scripts to install tools, test suites, CVA6 code and to execute tests
- **sim**: simulation environment (e.g. riscv-dv)
- **tb**: testbench module instancing the core
- **tests**: source of test cases and test lists

Contributing

We highly appreciate community contributions. To ease the work of reviewing contributions, please review CONTRIBUTING.

Contributions to the documentation (`docs/` and `tutorials/` directories) are very welcome as well.

If you find any problems or issues with CVA6 or the documentation, please check out the issue tracker and create a new issue if your problem is not yet tracked.
The CVA6 Kanban Board loosely tracks planned improvements.

Publication

If you use CVA6 in your academic work you can cite us:

▸ CVA6 Publication

Acknowledgements

Check out the acknowledgements.