riscv-dbg

Public

13 Branches **18** Tags Go to file t Go to file Add file Code

bluewww Merge pull request #185 from pulp-platform/phsaut…

b118d75 · 2 weeks ago

ci	tb: Rewrite buggy open…	4 years ago
debug_rom	debug_rom: Fix inferre…	3 years ago
doc	Update documentation	4 years ago
src	dm_csrs: Replace fifo_…	2 months ago
sva	Fixed enumeration name	6 years ago
tb	dm_csrs: Replace fifo_…	2 months ago
.gitignore	Address code review nits	4 years ago
.travis.yml	Add tb running dm+ri5c…	6 years ago
Bender.yml	Bump to latest commo…	4 years ago
CHANGELOG.md	Update CHANGELOG.…	3 months ago
LICENSE	Fork from Ariane	8 years ago
LICENSE.SiFive	Add SiFive license for …	7 years ago
README.md	README.md: Add jtag …	2 years ago
src_files.yml	Merge pull request #66…	6 years ago

About

RISC-V Debug Support for our PULP RISC-V Cores

#riscv #debug

Readme

Unknown, Apache-2.0 licenses found

Activity

Custom properties

294 stars

17 watching

91 forks

Report repository

Releases 11

RISC-V Debug Support … Latest
on Nov 23, 2025

+ 10 releases

Packages

No packages published

Contributors 24

README License Apache-2.0 license

RISC-V Debug Support for various Cores

This module is an implementation of a debug unit compliant with the RISC-V debug specification v0.13.1. It is used in the cva6, cv32e40p and ibex cores.

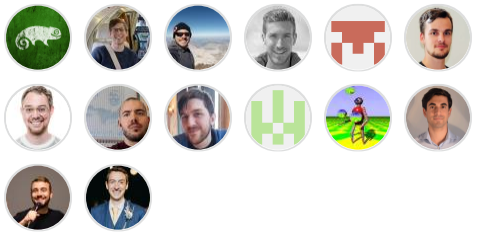
Implementation

We use an execution-based technique, also described in the specification, where the core is running in a "park loop". Depending on the request made to the debug unit via JTAG over the Debug Transport Module (DTM), the code that is being executed is changed dynamically. This approach simplifies the implementation side of the core, but means that the core is in fact always busy looping while debugging.

Features

The following features are currently supported

- Parametrizable buswidth for `XLEN=32` `XLEN=64` cores
- Accessing registers over abstract command
- Program buffer
- System bus access (only `XLEN`)



+ 10 contributors

Languages

- **SystemVerilog** 64.2%
- **C** 19.5%
- **Makefile** 4.3%
- **Linker Script** 3.4%
- **Assembly** 2.7%
- **Python** 2.0%
- **Other** 3.9%

- DTM with JTAG interface

These are not implemented (yet)

- Trigger module
- Quick access using abstract commands
- Accessing memory using abstract commands
- Authentication

Limitations

- The JTAG clock frequency needs to be lower than the system's clock frequency (see also #163).

Tests

We use OpenOCD's RISC-V compliance tests, our custom testbench in `tb/` and riscv-tests/debug.