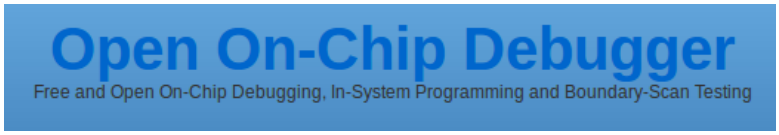
OpenOCD for RISC-V

2019-10-14

1 About

OpenOCD is a software that provides on-chip debugging, in-system programming and boundary-scan testing tools.

The official website is `openocd.org`



A fork with RISC-V support is available on `https://github.com/riscv/riscv-openocd`

OpenOCD is free software and is licensed under the **GNU General Public License v2.0**

This document presents a way of using OpenOCD to debug a RISC-V platform using a JTAG connection.

2 Position in the debug system

OpenOCD can be used as a translator between GDB and a RISC-V platform, as can be seen in the RISC-V external debug specification.

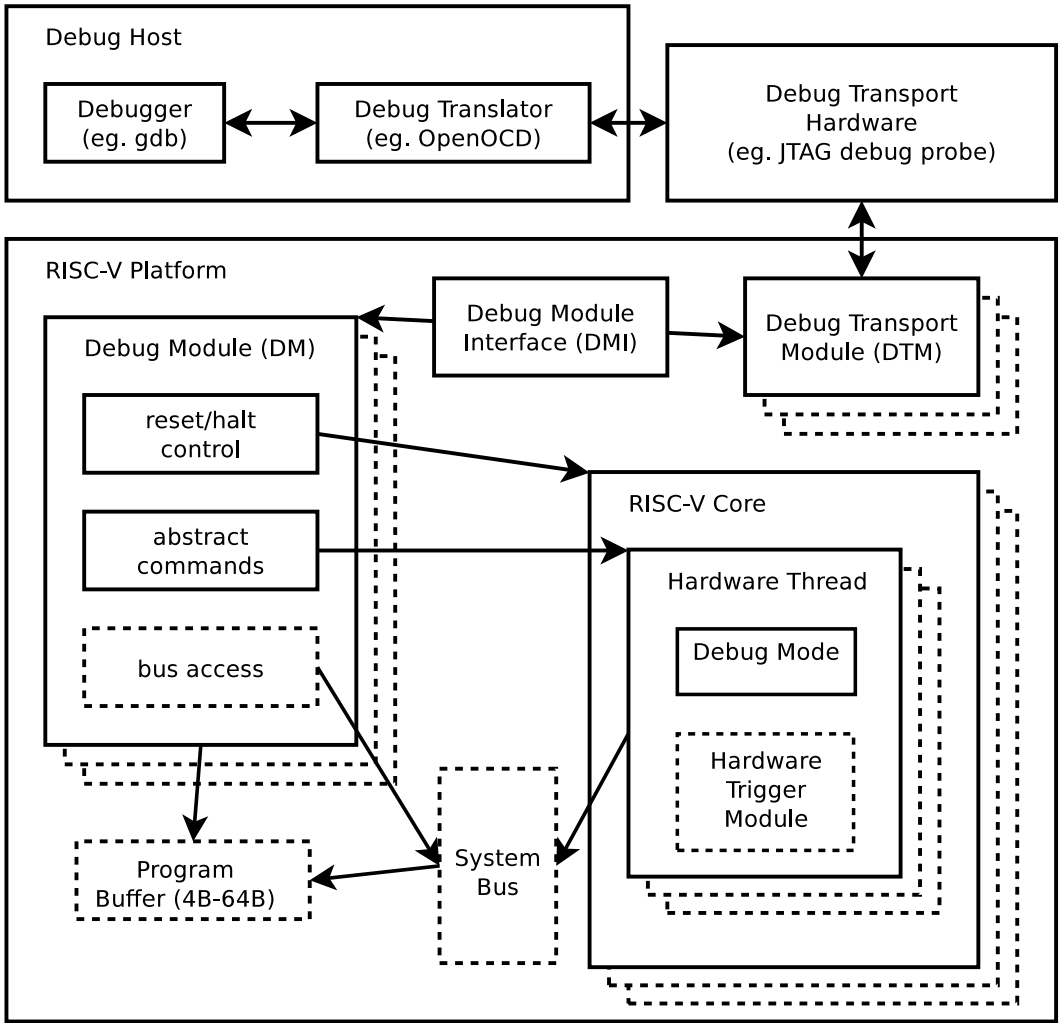


Figure 1: RISC-V Debug System Overview

OpenOCD can connect to GDB in two ways: with a TCP/IP socket or with pipes (stdin/stdout).

More information here: `http://openocd.org/doc-release/html/GDB-and-OpenOCD.html`

A debug adapter is needed to connect OpenOCD to the RISC-V platform. This document focuses on using JTAG transport hardware.

3 Config file

`http://openocd.org/doc-release/html/OpenOCD-Project-Setup.html`

A typical usage of OpenOCD is:

```
$ openocd −f config_file.cfg
```

A config file contains commands that OpenOCD will execute using its Jim-Tcl interpreter.

A single file can be broken into several ones. The command line call would then list every file in the correct order:

```
$ openocd −f config_file_1.cfg config_file_2.cfg config_file_3.cfg
```

It is also possible to launch a single command:

```
$ openocd −c "a command"
```

3.1 Interface

`http://openocd.org/doc-release/html/Debug-Adapter-Configuration.html`

The interface configuration tells OpenOCD how to use the transport hardware.

Config files for a lot of debug adapters can be found where OpenOCD was installed. The path should be **build/share/openocd/scripts/interface**.

Otherwise, refer to examples and the official documentation.

3.2 TAP declaration

`http://openocd.org/doc-release/html/TAP-Declaration.html`

A device with a JTAG interface means the device has a Test Access Port (TAP). You need to set up the active TAPs of the device by declaring them inside a configuration file.

Typically, this is achieved in a few lines:

```
set _CHIPNAME riscv
jtag newtap $_CHIPNAME cpu −irlen 5 −expected−id 0x12345678
```