Installation instructions

for tutorial: "A Tour of the RISC-V ISA Formal Specification"

RISC-V Foundation ISA Formal Spec Technical Committee

At RISC-V Summit, San Jose
December 12, 2019

Outline

About this slide deck

This is a standalone slide deck that accompanies the main slide deck for the tutorial: "A Tour of the RISC-V ISA Formal Specification", first presented at the RISC-V Summit, December 12, 2019, San Jose.

We recommend taking either Step A, or Steps A and B, depending on your objectives, in advance of the tutorial.

Step A: If you just want to learn how to *read and consult the spec*

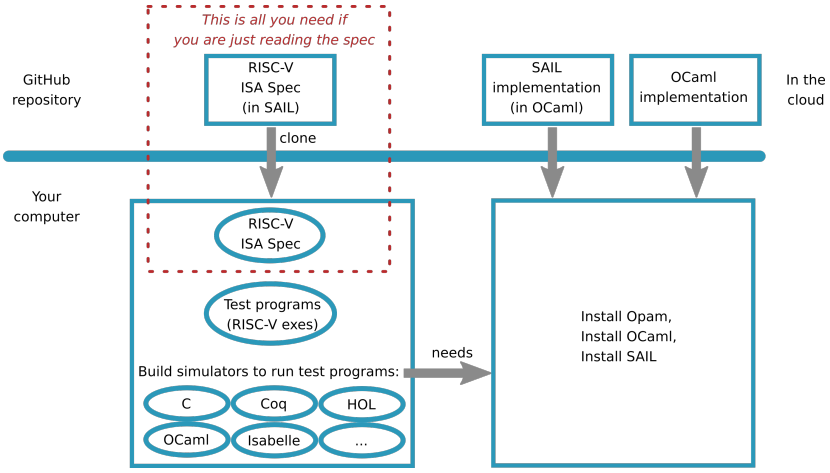
This merely git-clones a certain repo which contains the SAIL source code for the RISC-V ISA Formal Spec.

Step B: If you also want to learn how to *execute the spec*

This will compile a RISC-V ISA simulator from the SAIL formal spec, which you can use to execute:

- The standard suite of RISC-V ISA tests
- The standard RISC-V Compliance Test suite
- RISC-V ELF binaries that you create from other source codes

Installation Overview



Step A Installation: Cloning the SAIL RISC-V ISA Formal Spec

Just one step:

```
$ git clone https://github.com/rems-project/sail-riscv
```

What you get:

```
$ sail-riscv
$ tree -d
|-- ...
|-- model/
|-- ...
```

This directory contains all the spec files

That's all you need, for just reading and consulting (not executing) the spec!

Step B Installation: to create an executable version of the spec

Safety net, in case things go wrong:

The instructions in these slides are collected here from various sources for your convenience. In case of trouble, the original full instructions can be found at:

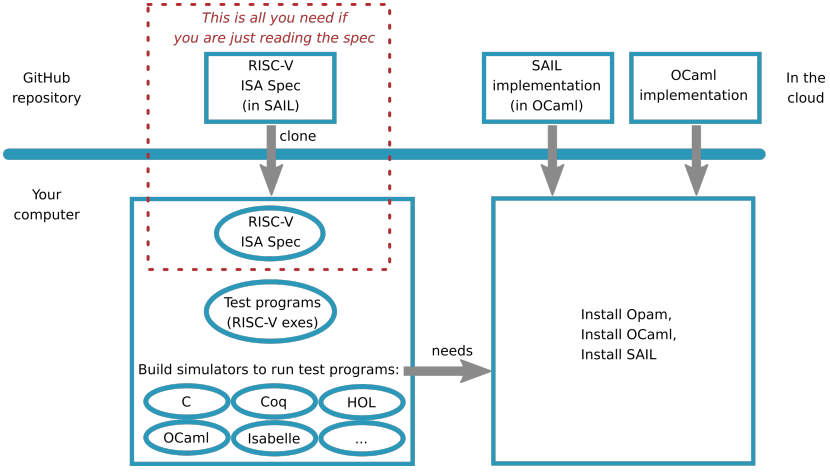
- Installing Opam:
  `https://opam.ocaml.org/doc/Install.html`
- Installing Ocaml for SAIL, and installing SAIL:
  `https://github.com/rems-project/sail/wiki/OPAMInstall`

OS requirements

These instructions are given for Debian/Ubuntu Linux. Alternatives:

- You could install a virtual machine running Debian/Linux and follow these instructions
- OCaml and SAIL will also install on other OSes. Wherever you see "`apt get`" here, which is the standard package manager for Debian/Ubuntu, please substitute the package manager for your OS. The websites mentioned under "Safety net" above may also have more information for other OSes.

Installation Overview



Step B.1: Installing Opam, the package manager for OCaml

Reminder: Step B is not necessary if you're only reading and consulting the spec. It's necessary to build simulator(s) from the SAIL RISC-V ISA Formal Spec that can execute RISC-V binaries.

Use either of the following methods to install Opam:

Step B.1 Method 1: Use curl to get an install script and run it directly:

Install curl if you don't already have it:

```
$ sudo apt-get install curl
```

Install Opam:

```
$ sudo sh <(curl -sL https://raw.githubusercontent.com/ocaml/opam/master/shell/install.sh)
```

Installing Opam (contd.)

Step B.1 Method 2: Download the install script and run it explicitly

Download `install.sh` from the above web link, and run it:

```
$ ... download   https://raw.githubusercontent.com/ocaml/opam/master/shell/install.sh
$ sudo sh install.sh
## Downloading opam 2.0.5 for linux on x86_64...
## Downloaded.
## Where should it be installed ?  [/usr/local/bin]
## opam 2.0.5 installed to /usr/local/bin

## Run this script again with '--restore ' to revert.
```

Step B.1: Verify successful opam installation (after using either method above)

```
$ which opam
/usr/local/bin/opam
$ opam --version

2.0.5
```

Step B.2: Installing OCaml using Opam

Once Opam is installed, you can use it to install OCaml and SAIL. First, OCaml:

Step B.2: Installing OCaml, and verifying we've got it

```
# Environment setup
$ opam init
$ eval ‘opam env‘
# Install specific version of OCaml
$ opam switch create ocaml-base-compiler.4.06.1
$ eval ‘opam config env‘

# Verify we've got it
$ which ocaml
/home/nikhil/.opam/ocaml-base-compiler.4.06.1/bin/ocaml
$ ocaml -version
The OCaml toplevel, version 4.06.1
```

Note: 4.06.1 is not the latest version of OCaml, but it is known to be suitable for SAIL (it is the version used during CI of SAIL).

Step B.3: Installing SAIL using Opam

Some prerequisites: install certain libraries needed by SAIL (if not already installed on your system)

On Linux (Debian, Ubuntu, ...)

```
$ sudo apt-get install build-essential libgmp-dev z3 m4 pkg-config zlib1g-dev
$ sudo apt-get install device-tree-compiler       Needed by OCaml-based simulator
```

Homebrew (Mac OS)

```
$ brew install gmp z3
```

Step B.3 (contd.): Installing SAIL using Opam

Set up opam so it knows where to get SAIL

```
$ opam repository add rems https://github.com/rems-project/opam-repository.git
```

Install SAIL, and verify we've got it

```
$ opam install sail

# Verify we've got it
$ which sail
/home/nikhil/.opam/ocaml-base-compiler.4.06.1/bin/sail
$ sail --help
Sail 0.11 (sail2 @ opam)
usage:  sail <options> <file1.sail> ...  <fileN.sail>

 -o <prefix> select output filename prefix
 -i start interactive interpreter

 ...
```

Step B.4: Building RISC-V simulator(s) from the SAIL spec

A simulator can be used to execute RISC-V binaries.

Step B.4: Building RISC-V simulator(s):

```
$ cd sail-riscv          i.e., be in the git-cloned directory
$ make
```

Creates the following (RV64 is the default):

```
c_emulator/riscv_sim_RV64              Generates a C simulator and compiles it
ocaml_emulator/riscv_ocaml_sim_RV64    Generates an OCaml simulator and compiles it
```

... and Coq, Isabelle and HOL4 models in other directories ...

You can also create corresponding RV32 simulators:

```
$ ARCH=RV32 make
```