RISC-V ISA Fo

QUESTIONS RESPONSES 12

12 responses

SUMMARY INDIVIDUAL

Accepting responses

Your name

12 responses

Luke Kenneth Casson Leighton

Josh Scheid

Chuanhua Chang

Frédéric Pétrot

Tariq Kurd

andrew dellow

Håkan Thörngren

Nathan Studer

Jesse Millwood

Andrew Tolmach

Edwin Török

Jan Paesmans

Your email address (optional)

10 responses

lkcl@lkcl.net

jscheid@ventanamicro.com

chchang@andestech.com

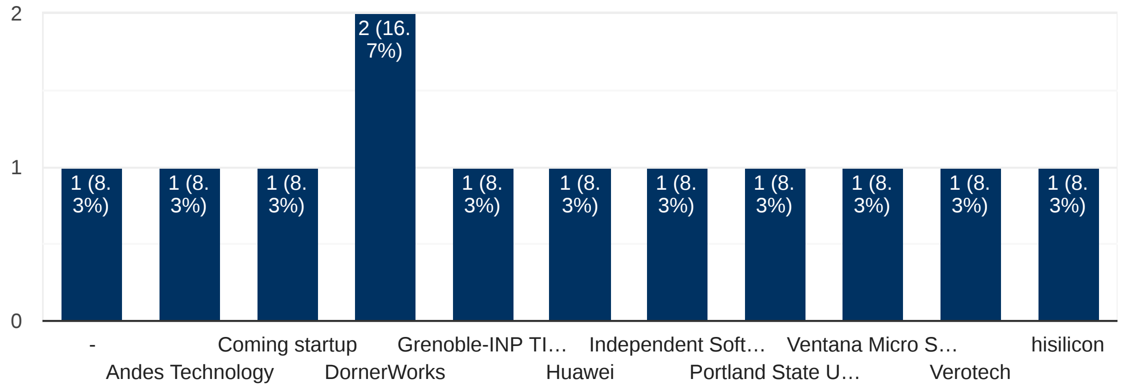
QUESTIONS RESPONSES 12

tolmach@pdx.edu

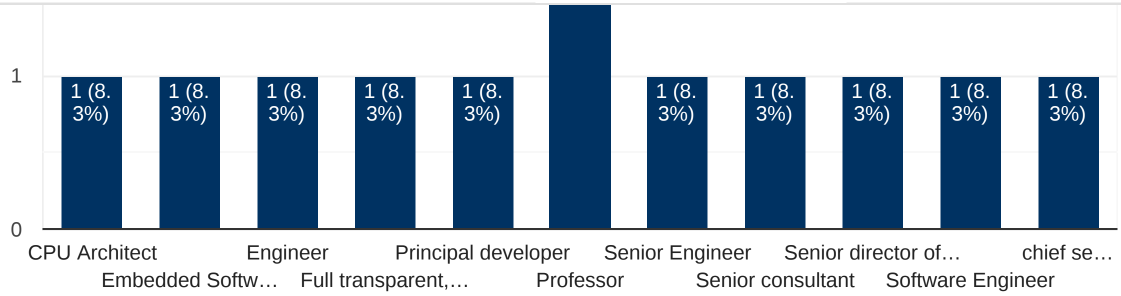
edwin@etorok.net

Your organisation

12 responses



QUESTIONS RESPONSES 12



The group you are speaking for, if any

3 responses

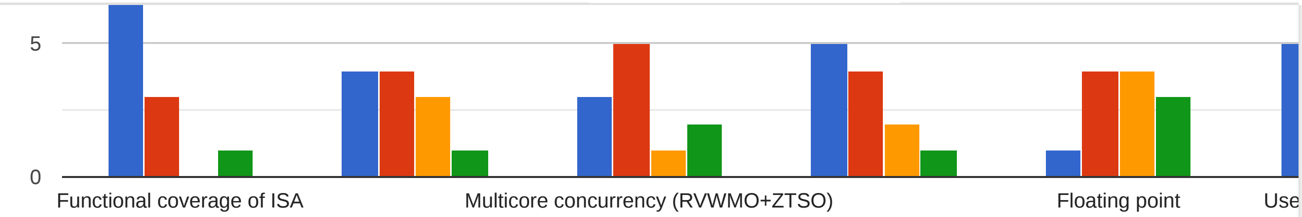
n/a

None

Huawei RISC-V development

How important is each aspect of a formal ISA specification for RISC-V?

QUESTIONS RESPONSES 12



Comments on any of the above

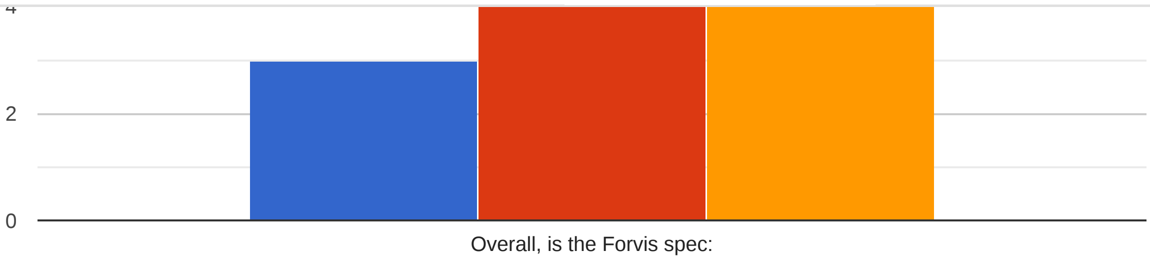
2 responses

The intent behind the content of the prose specification versus the formal specification should be explicit. This will help guide the content so that the two appear as complementary instead of redundant.

Obviously, each of these may be crucial to somebody: I've answered what is important to me.

The Candidate Formal Models

QUESTIONS RESPONSES 12



Forvis - comments

7 responses

the answer is the same for all of the formal models: it is too early to make a decision. each of the models is extremely good: it's just that they're (all of them) incomplete (still under development in some way). in addition, i think you'll find that even *making* a choice will result in that team becoming a critical dependency *for the entire RISC-V ecosystem*. if they're an academic team, that's unfortunate: once the project no longer receives funding or the research project ends, then so does RISC-V "conformance". and if they're a Corporation, the Corporation may manipulate the RISC-V ecosystem for profit-maximising purposes, and if it goes bust, the project ends, and so does RISC-V "conformance". so not only is it a bad idea to pick one *right now*, it's a bad idea to pick only *ONE* of these formal verification suites *at all*. instead it would be far, far better for the RISC-V Formal Verification Group to develop a *STANDARD* for Formal Verification, to which **ALL** of these may comply. that's what a Standards Organisation does: develop *STANDARDS*, *NOT* select some random codebase off the internet and say "here! this is now a standard!". so you need to define the *expected results*, in sufficient detail and with sufficient clarity such that *ALL* of the FIVE formal models may conform and comply with it, in a machine-executable fashion. if that's too challenging, then at least some human-verifiable expectations may be defined.

Concurrency.

Supports neither instruction encodings and asm syntax nor conccurency

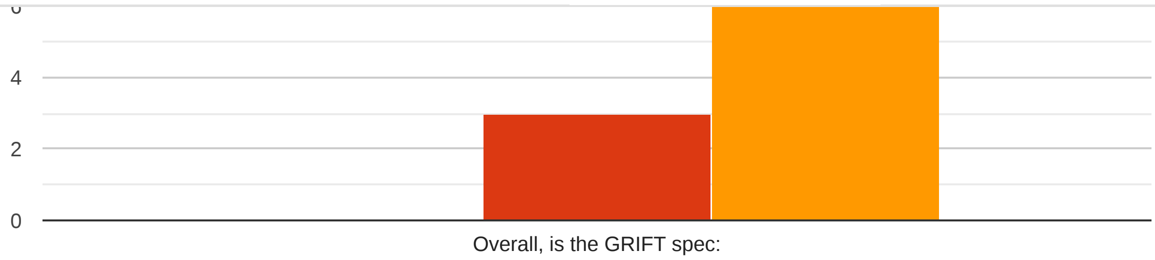
too slow

Haskell is a strongly desired by me and Forvis has is a permissive license. It seems quite well executed. I can definitely see this one would be useful to me and it should serve well as a formal model.

Uses Haskell to good effect without (hopefully) scaring off readers more used to conventional ISA descriptions. But I'm not sure how easy it will be to extract definitions suitable for use in a theorem prover.

Concurrency and synthesis support would be useful

QUESTIONS RESPONSES 12



GRIFT - comments

7 responses

the answer is the same for all of the formal models: it is too early to make a decision. each of the models is extremely good: it's just that they're (all of them) incomplete (still under development in some way). in addition, i think you'll find that even *making* a choice will result in that team becoming a critical dependency *for the entire RISC-V ecosystem*. if they're an academic team, that's unfortunate: once the project no longer receives funding or the research project ends, then so does RISC-V "conformance". and if they're a Corporation, the Corporation may manipulate the RISC-V ecosystem for profit-maximising purposes, and if it goes bust, the project ends, and so does RISC-V "conformance". so not only is it a bad idea to pick one *right now*, it's a bad idea to pick only *ONE* of these formal verification suites *at all*. instead it would be far, far better for the RISC-V Formal Verification Group to develop a *STANDARD* for Formal Verification, to which **ALL** of these may comply. that's what a Standards Organisation does: develop *STANDARDS*, *NOT* select some random codebase off the internet and say "here! this is now a standard!". so you need to define the *expected results*, in sufficient detail and with sufficient clarity such that *ALL* of the FIVE formal models may conform and comply with it, in a machine-executable fashion. if that's too challenging, then at least some human-verifiable expectations may be defined.

Privilege levels and concurrency.

Seems to support the least features among the tools you propose

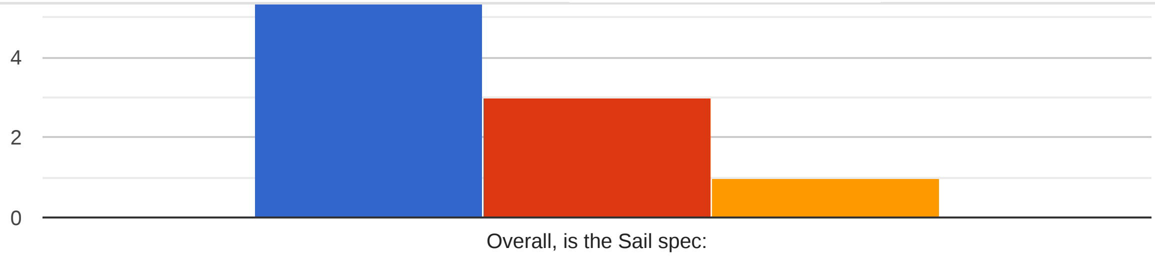
too slow

GPL, totally unusable for partial inclusion in commercial products. While it is a formal specification, I can definitely see that there are chance for an executable specification to be at least partially included in actual products, test or development and there is a wide grey zone. People may say that a license will say what is allowed or not, but many commercial users will be very careful with this, and there are alternatives without this problem. I will for sure stay at a safe distance from this one.

Only accessible to Haskell experts.

A more permissive license might encourage more collaboration

QUESTIONS RESPONSES 12



Sail - comments

7 responses

the answer is the same for all of the formal models: it is too early to make a decision. each of the models is extremely good: it's just that they're (all of them) incomplete (still under development in some way). in addition, i think you'll find that even *making* a choice will result in that team becoming a critical dependency *for the entire RISC-V ecosystem*. if they're an academic team, that's unfortunate: once the project no longer receives funding or the research project ends, then so does RISC-V "conformance". and if they're a Corporation, the Corporation may manipulate the RISC-V ecosystem for profit-maximising purposes, and if it goes bust, the project ends, and so does RISC-V "conformance". so not only is it a bad idea to pick one *right now*, it's a bad idea to pick only *ONE* of these formal verification suites *at all*. instead it would be far, far better for the RISC-V Formal Verification Group to develop a *STANDARD* for Formal Verification, to which **ALL** of these may comply. that's what a Standards Organisation does: develop *STANDARDS*, *NOT* select some random codebase off the internet and say "here! this is now a standard!". so you need to define the *expected results*, in sufficient detail and with sufficient clarity such that *ALL* of the FIVE formal models may conform and comply with it, in a machine-executable fashion. if that's too challenging, then at least some human-verifiable expectations may be defined.

More features, good readability, can generate C

The tool covers encodings and concurrency, generation for theorem provers, and is the fastest one in simulation. Futhermore it has a BSD licence

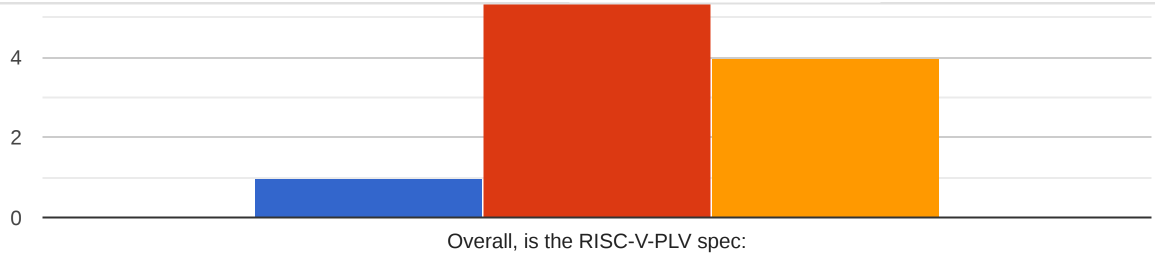
Faster, specification language is better

I put it on Adequate, I have somewhat limited use of it as I am focused on Haskell and will have most use for such specification, but I am not to dismiss it.

Probably the best choice at full scale; language and definition style seem a little heavy-weight for the minimal subsets of the ISA. Current extraction to Coq does not produce very idiomatic definitions, but this can no doubt be improved over time.

Ticks all the boxes. Floating point might be nice, and also some integration with HardCaml.

QUESTIONS RESPONSES 12



RISC-V-PLV - comments

7 responses

the answer is the same for all of the formal models: it is too early to make a decision. each of the models is extremely good: it's just that they're (all of them) incomplete (still under development in some way). in addition, i think you'll find that even *making* a choice will result in that team becoming a critical dependency *for the entire RISC-V ecosystem*. if they're an academic team, that's unfortunate: once the project no longer receives funding or the research project ends, then so does RISC-V "conformance". and if they're a Corporation, the Corporation may manipulate the RISC-V ecosystem for profit-maximising purposes, and if it goes bust, the project ends, and so does RISC-V "conformance". so not only is it a bad idea to pick one *right now*, it's a bad idea to pick only *ONE* of these formal verification suites *at all*. instead it would be far, far better for the RISC-V Formal Verification Group to develop a *STANDARD* for Formal Verification, to which **ALL** of these may comply. that's what a Standards Organisation does: develop *STANDARDS*, *NOT* select some random codebase off the internet and say "here! this is now a standard!". so you need to define the *expected results*, in sufficient detail and with sufficient clarity such that *ALL* of the FIVE formal models may conform and comply with it, in a machine-executable fashion. if that's too challenging, then at least some human-verifiable expectations may be defined.

Concurrency.

Does not cover most of the things I feel interesting (encoding) or very hard to have right (concurrency).

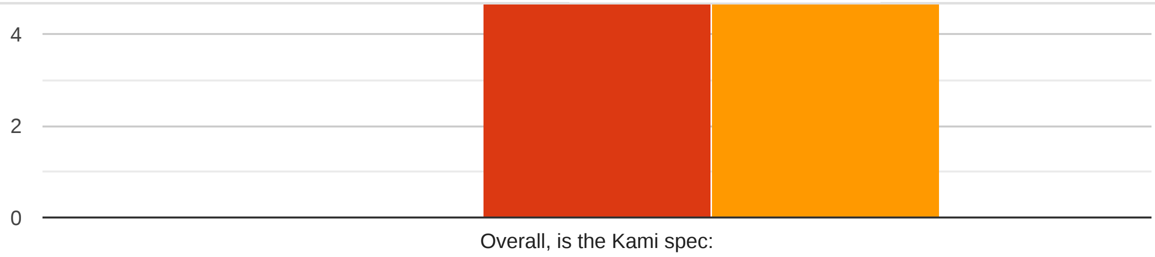
too slow

Haskell, permissive license, but it is an academic project and to me such projects are often orphaned as soon as the academic interest shifts, as it often does. Lots of good stuff come out of academia, but it is often not entirely complete or long-term.

Similar to Forvis, but requires somewhat more Haskell expertise to read (OK for me, but probably not ideal for broader community).

Concurrency support would be nice

QUESTIONS RESPONSES 12



Kami - comments

7 responses

the answer is the same for all of the formal models: it is too early to make a decision. each of the models is extremely good: it's just that they're (all of them) incomplete (still under development in some way). in addition, i think you'll find that even *making* a choice will result in that team becoming a critical dependency *for the entire RISC-V ecosystem*. if they're an academic team, that's unfortunate: once the project no longer receives funding or the research project ends, then so does RISC-V "conformance". and if they're a Corporation, the Corporation may manipulate the RISC-V ecosystem for profit-maximising purposes, and if it goes bust, the project ends, and so does RISC-V "conformance". so not only is it a bad idea to pick one *right now*, it's a bad idea to pick only *ONE* of these formal verification suites *at all*. instead it would be far, far better for the RISC-V Formal Verification Group to develop a *STANDARD* for Formal Verification, to which **ALL** of these may comply. that's what a Standards Organisation does: develop *STANDARDS*, *NOT* select some random codebase off the internet and say "here! this is now a standard!". so you need to define the *expected results*, in sufficient detail and with sufficient clarity such that *ALL* of the FIVE formal models may conform and comply with it, in a machine-executable fashion. if that's too challenging, then at least some human-verifiable expectations may be defined.

Privilege levels and concurrency.

Covers half my needs !

unknown speed, no privilege level support?

Another non-Haskell specification, see comments about Sail.

Organization focused on synthesis is not so natural for other purposes. Coq is not very accessible to broader community.

Coverage of privilege levels would be nice p

QUESTIONS

RESPONSES 12

Still need SAIL to support F-extension, otherwise it's the best