Technical Committee Update

Yunsup Lee and Silviu Chiricescu
yunsup@sifive.com
silviu.chiricescu@baesystems.com

Immediate Goals

Immediate Goals

- To maintain a roadmap of the RISC-V ISA

Immediate Goals

- To maintain a roadmap of the RISC-V ISA
- To provide and maintain a golden simulator for the RISC-V ISA

Immediate Goals

- To maintain a roadmap of the RISC-V ISA
- To provide and maintain a golden simulator for the RISC-V ISA
- To provide and maintain a set of verification/validation tests to ensure conformance with the evolving RISC-V ISA and its extensions

Immediate Goals

- To maintain a roadmap of the RISC-V ISA
- To provide and maintain a golden simulator for the RISC-V ISA
- To provide and maintain a set of verification/validation tests to ensure conformance with the evolving RISC-V ISA and its extensions
- To upstream the software development tools (compiler, debugger, etc.)

Immediate Goals

- To maintain a roadmap of the RISC-V ISA
- To provide and maintain a golden simulator for the RISC-V ISA
- To provide and maintain a set of verification/validation tests to ensure conformance with the evolving RISC-V ISA and its extensions
- To upstream the software development tools (compiler, debugger, etc.)
- To maintain and update a list of hardware implementations of the architecture

Longer Term Goals

Longer Term Goals

- Establish processes to define and standardize future ISA extensions

Longer Term Goals

- Establish processes to define and standardize future ISA extensions
- Provide guidelines for platform integration to avoid fragmentation in the RISC-V software ecosystem

Longer Term Goals

- Establish processes to define and standardize future ISA extensions
- Provide guidelines for platform integration to avoid fragmentation in the RISC-V software ecosystem
- Setting up program committees for future RISC-V workshops to make it a prominent venue to present in-progress work related to RISC-V

Technical Committee Task Groups

Technical Committee Task Groups

- Currently consists of 7 task groups:

Technical Committee Task Groups

- Currently consists of 7 task groups:
  - Opcode space management, Krste Asanovic, UC Berkeley

Technical Committee Task Groups

- Currently consists of 7 task groups:
  - Opcode space management, Krste Asanovic, UC Berkeley
  - Privileged ISA specification, Andrew Waterman, SiFive

Technical Committee Task Groups

- Currently consists of 7 task groups:
  - Opcode space management, Krste Asanovic, UC Berkeley
  - Privileged ISA specification, Andrew Waterman, SiFive
  - Formal specification, Rishiyur Nikhil, Bluespec

Technical Committee Task Groups

- Currently consists of 7 task groups:
  - Opcode space management, Krste Asanovic, UC Berkeley
  - Privileged ISA specification, Andrew Waterman, SiFive
  - Formal specification, Rishiyur Nikhil, Bluespec
  - Debug specification, Tim Newsome, SiFive

Technical Committee Task Groups

- Currently consists of 7 task groups:
  - Opcode space management, Krste Asanovic, UC Berkeley
  - Privileged ISA specification, Andrew Waterman, SiFive
  - Formal specification, Rishiyur Nikhil, Bluespec
  - Debug specification, Tim Newsome, SiFive
  - Security, Joe Xie, NVIDIA

Technical Committee Task Groups

- Currently consists of 7 task groups:
  - Opcode space management, Krste Asanovic, UC Berkeley
  - Privileged ISA specification, Andrew Waterman, SiFive
  - Formal specification, Rishiyur Nikhil, Bluespec
  - Debug specification, Tim Newsome, SiFive
  - Security, Joe Xie, NVIDIA
  - Vector extensions, Krste Asanovic, UC Berkeley

Technical Committee Task Groups

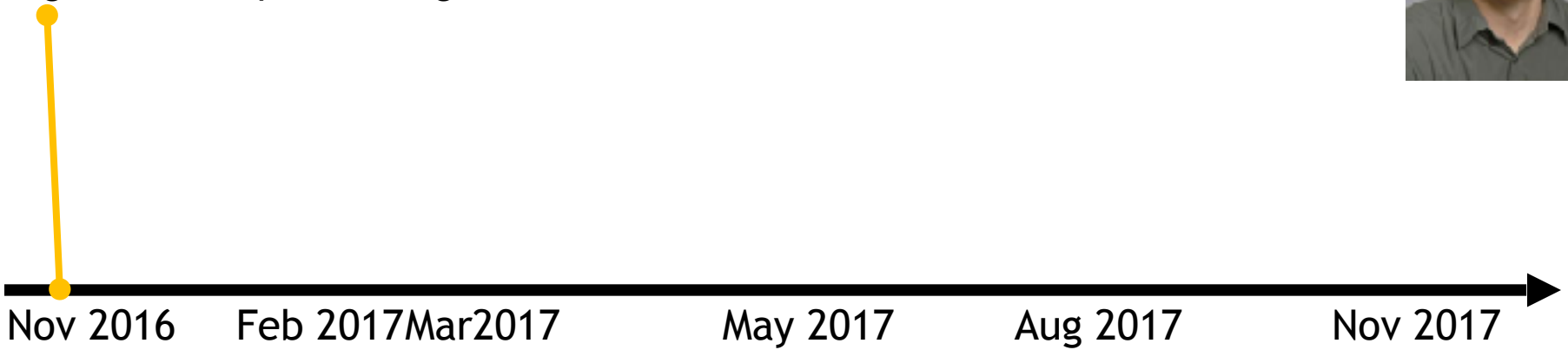
- Currently consists of 7 task groups:
  - Opcode space management, Krste Asanovic, UC Berkeley
  - Privileged ISA specification, Andrew Waterman, SiFive
  - Formal specification, Rishiyur Nikhil, Bluespec
  - Debug specification, Tim Newsome, SiFive
  - Security, Joe Xie, NVIDIA
  - Vector extensions, Krste Asanovic, UC Berkeley
  - SW tool chain, Arun Thomas, BAE Systems

Technical Committee Task Groups

- Currently consists of 7 task groups:
  - Opcode space management, Krste Asanovic, UC Berkeley
  - Privileged ISA specification, Andrew Waterman, SiFive
  - Formal specification, Rishiyur Nikhil, Bluespec
  - Debug specification, Tim Newsome, SiFive
  - Security, Joe Xie, NVIDIA
  - Vector extensions, Krste Asanovic, UC Berkeley
  - SW tool chain, Arun Thomas, BAE Systems
- Task groups could be added/phased out as needed

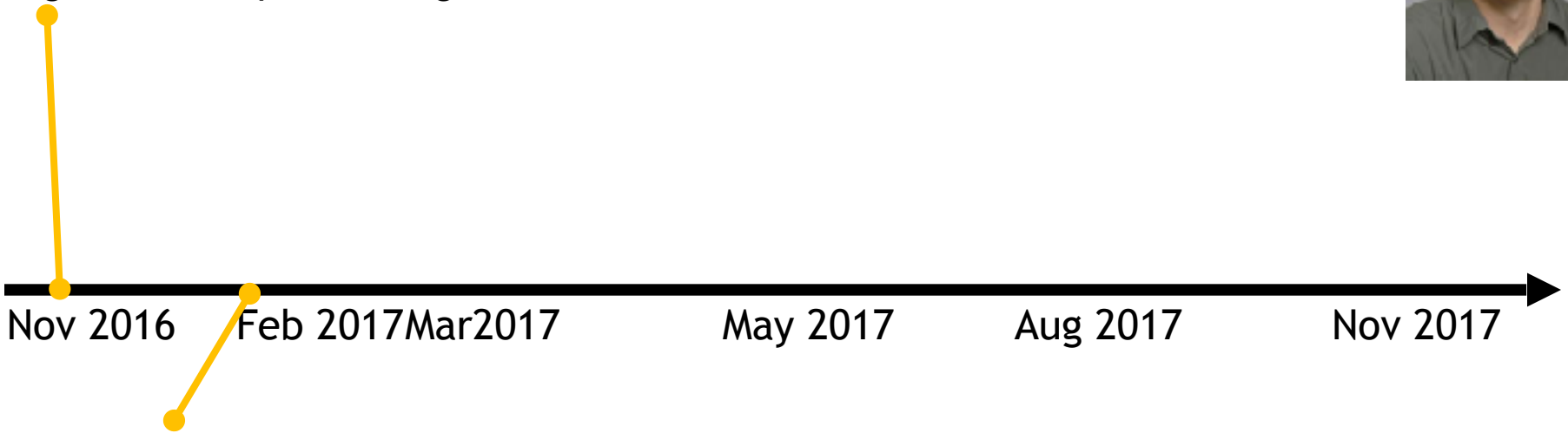
Opcode Management TG

Agree/tweak plan, assign more leaders and doers



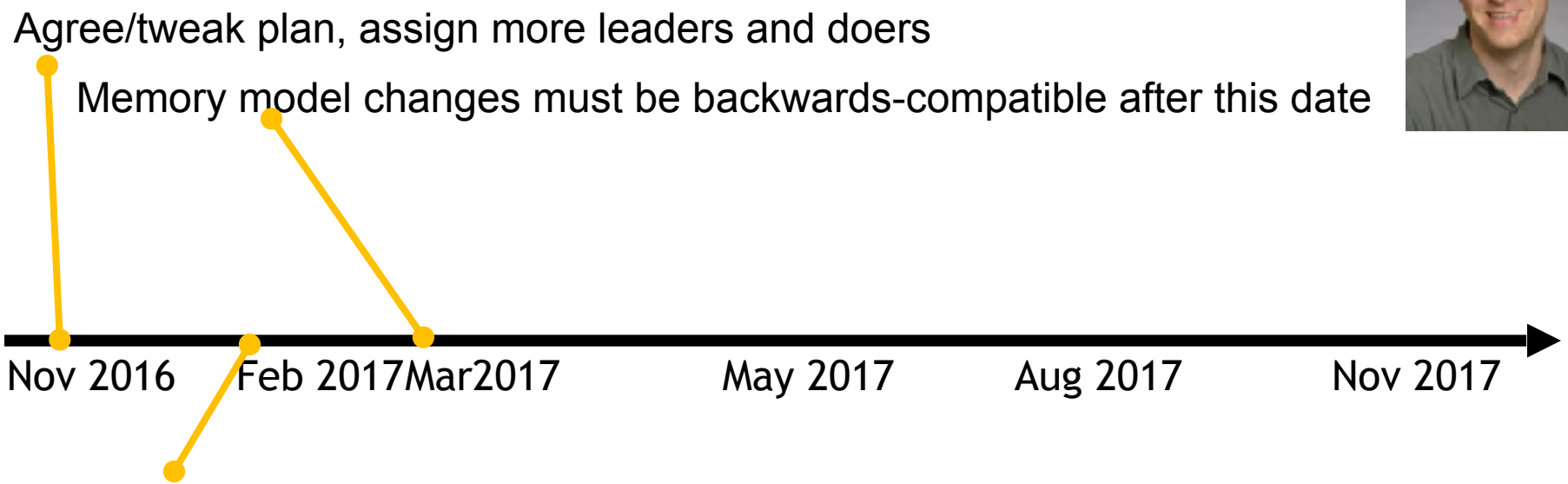
Opcode Management TG

Agree/tweak plan, assign more leaders and doers



* Debug spec ratified by Foundation
* Calling convention fixed and documented
* ELF format fixed and documented
* Priv-1.10.0
* M-mode/S-mode changes must be backwards-compatible after this date

Opcode Management TG



* Debug spec ratified by Foundation
* Calling convention fixed and documented
* ELF format fixed and documented
* Priv-1.10.0
* M-mode/S-mode changes must be backwards-compatible after this date

Opcode Management TG

Agree/tweak plan, assign more leaders and doers

Memory model changes must be backwards-compatible after this date

* Priv-1.11.0
* RV32EMAC RV32IMAFDQC RV64IMAFDQC ratified

Nov 2016 Feb 2017 Mar2017 May 2017 Aug 2017 Nov 2017

* Debug spec ratified by Foundation
* Calling convention fixed and documented
* ELF format fixed and documented
* Priv-1.10.0
* M-mode/S-mode changes must be backwards-compatible after this date

Opcode Management TG

Agree/tweak plan, assign more leaders and doers

Memory model changes must be backwards-compatible after this date

* Priv-1.11.0
* RV32EMAC RV32IMAFDQC RV64IMAFDQC ratified

Priv-1.12.0

Nov 2016 Feb 2017 Mar2017 May 2017 Aug 2017 Nov 2017

* Debug spec ratified by Foundation
* Calling convention fixed and documented
* ELF format fixed and documented
* Priv-1.10.0
* M-mode/S-mode changes must be backwards-compatible after this date

Opcode Management TG

Agree/tweak plan, assign more leaders and doers

Memory model changes must be backwards-compatible after this date

* Priv-1.11.0
* RV32EMAC RV32IMAFDQC RV64IMAFDQC ratified

Priv-1.12.0

Nov 2016 Feb 2017 Mar2017 May 2017 Aug 2017 Nov 2017

* Debug spec ratified by Foundation
* Calling convention fixed and documented
* ELF format fixed and documented
* Priv-1.10.0
* M-mode/S-mode changes must be backwards-compatible after this date

* V ratified by Foundation
* Priv-1.13.0 -> Priv-2.0 ratified?
* Complete Linux/KVM platform spec agreed, supports other OS (FreeBSD etc.)

Privileged ISA Spec TG

Privileged ISA Spec TG

- Charter
  - To define and specify a unified RISC-V privileg
    architecture and hardware platform

Privileged ISA Spec TG

- Charter
  - To define and specify a unified RISC-V privileg architecture and hardware platform
- Goals
  - Version 1.10 of supervisor-mode and machine-mode architecture proposal by Feb ’17
    - no backwards-incompatible changes after this point
  - Complete Unix platform spec proposal, incl. hypervisor support, by Nov ’17
  - Ratification end of ’17/beginning of ’18

Formal Spec TG

Formal Spec TG

- Charter
  - To produce a *formal* specification of the RISC-V ISA that is
    - Implementation independent (no micro-arch details), high level, executable, suitable for mechanized formal tools
    - Complementary/in addition to the textual ISA spec docs

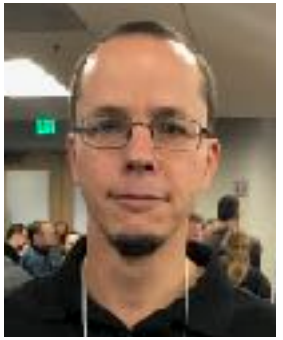
Formal Spec TG

- Charter
  - To produce a *formal* specification of the RISC-V ISA that is
    - Implementation independent (no micro-arch details), high level, executable, suitable for mechanized formal tools
    - Complementary/in addition to the textual ISA spec docs
- Initial Targets
  - Just the basic ISA (RV32 and RV64) and the common standard extensions (I, M)
  - For F/D, since it is IEEE standard, not sure we have much to contribute

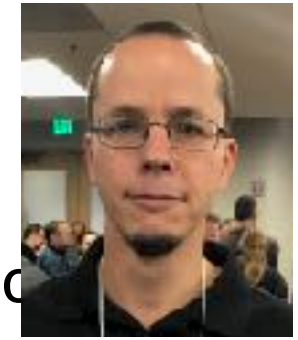
Formal Spec TG

- Charter
  - To produce a *formal* specification of the RISC-V ISA that is
    - Implementation independent (no micro-arch details), high level, executable, suitable for mechanized formal tools
    - Complementary/in addition to the textual ISA spec docs
- Initial Targets
  - Just the basic ISA (RV32 and RV64) and the common standard extensions (I, M)
  - For F/D, since it is IEEE standard, not sure we have much to contribute
- Status
  - Basic spec: Models written in L3 (SRI), in BSV (Bluespec), in Coq (MIT)
  - Memory model: Two talks this morning

Debug Spec TG

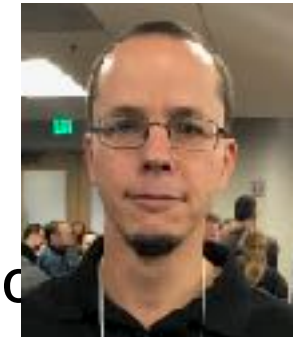


Debug Spec TG



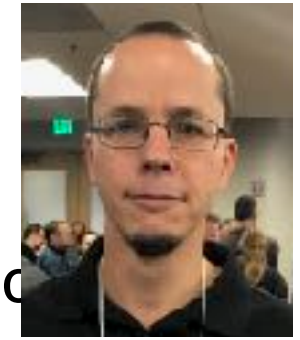
- Charter
  - To specify a standardized way to debug RISC-V co

Debug Spec TG



- Charter
  - To specify a standardized way to debug RISC-V co
- Status
  - There are 2 spec proposals
    - Instruction stuffing/fetching, SiFive
    - Memory-mapped IO interface, ROA Logic
  - Could use insight of the people who are going to do the hardware implementation. Meeting Thursday at 11am.

Debug Spec TG



- Charter
  - To specify a standardized way to debug RISC-V co
- Status
  - There are 2 spec proposals
    - Instruction stuffing/fetching, SiFive
    - Memory-mapped IO interface, ROA Logic
  - Could use insight of the people who are going to do the hardware implementation. Meeting Thursday at 11am.
- Goals
  - Ratification Feb ’17

Security TG

Security TG

- Charter
  - Define RISC-V security extension specification including:
    - Secure processing model and necessary ISA support
    - Crypto algorithm instructions (AES, SHA, RSA, ECC, Random)
  - Provide necessary HW/SW support according to the spec

Security TG

- Charter
  - Define RISC-V security extension specification including:
    - Secure processing model and necessary ISA support
    - Crypto algorithm instructions (AES, SHA, RSA, ECC, Random)
  - Provide necessary HW/SW support according to the spec
- Goals
  - Security extension spec 0.9 open to public review by the 6th workshop

Security TG

- Charter
  - Define RISC-V security extension specification including:
    - Secure processing model and necessary ISA support
    - Crypto algorithm instructions (AES, SHA, RSA, ECC, Random)
  - Provide necessary HW/SW support according to the spec
- Goals
  - Security extension spec 0.9 open to public review by the 6th workshop
- Meetings
  - every other Wed from 6-7 PST. Meeting Thursday 1pm.

Software Tool Chain TG

Software Tool Chain TG

- Charter:
  - To define a standard, easy to build and use software toolchain for the RISC-V ecosystem

Software Tool Chain TG

- Charter:
  - To define a standard, easy to build and use software toolchain for the RISC-V ecosystem
- Goals:
  - Documenting current SW stack https://github.com/arunthomas/riscv-sw/wiki
  - Upstreaming (Binutils, GCC, LLVM, etc.)
  - QEMU: Update to priv 1.9.1, device support
  - Creating an ABI specification
  - Continue to push on Linux distribution support (e.g., Debian, Fedora)
  - Documentation
  - Docker images, cross-toolchain packages in Linux distros

Call for Participation

Call for Participation

- All member organizations should nominate a person for the technical committee
  - If not, please contact us

Call for Participation

- All member organizations should nominate a person for the technical committee
  - If not, please contact us
- Please let us know if you'd like to participate in a TG or start a TG
  - Your help is needed
  - Come to tech committee meeting on Thursday 8:30am

Call for Participation

- All member organizations should nominate a person for the technical committee
  - If not, please contact us
- Please let us know if you'd like to participate in a TG or start a TG
  - Your help is needed
  - Come to tech committee meeting on Thursday 8:30am
- Finally, thank you for your contribution!

Call for Participation

- All member organizations should nominate a person for the technical committee
  - If not, please contact us
- Please let us know if you'd like to participate in a TG or start a TG
  - Your help is needed
  - Come to tech committee meeting on Thursday 8:30am
- Finally, thank you for your contribution!
- Any questions?