RISC-V international academia and training activities thematic session

Michael Engel, Professor, Univ. of Bamberg and NTNU

RISC-V Academic & Training Special Interest Group activities

Keith Graham, Head of University Program at Codasip

The Codasip RISC-V University Program

Vladimir Herdt, Post-doctoral researcher DFKI/Univ. of Bremen

RISC-V Virtual Prototyping and Verification

Discussion
RISC-V Academic and Training Special Interest Group activities

Michael Engel (michael.engel@uni-bamberg.de)
Professor, Lehrstuhl für Praktische Informatik, insb. Systemnahe Programmierung, Universität Bamberg, Germany
https://www.uni-bamberg.de/sysnap
Chair of the RISC-V academia & training SIG

Licensed under CC BY-SA 4.0 unless noted otherwise
Education: expectations vs. reality?

- What does a degree prepare for?
- How can we keep knowledge up to date?

Academic career <-> Career in industry
Challenges for academic education

Challenges

• Are the **methods and material** we use in teaching a **good fit** for
  • a career in industry?
  • a career in academia?

• How can we support **lifelong learning**?
  • how to teach engineers in industry about current developments?

• Which **hardware and software platform** should be used?
  • Practical experience is highly relevant (+theoretical background)
What can RISC-V contribute?

• **Idea:** enable students to "dig deeper"
  • Use a platform that allows to **look at the internals**
  • Enable **vertical integration** using a single platform
    • hardware design → system software → applications
  • Use the platform in **different scenarios**
    • embedded/IoT → desktop/server → HPC
• Why is **RISC-V** specifically well suited for this?
  • **Openness** – not only open specs, but open implementations
    • Hardware: CPU cores, simulation models, emulators
    • Software: Compilers, operating systems, runtimes
  • **Simplicity** – basic RISC-V is simpler than x86 or ARMv8
• **Vision for education:** a completely **open source computer**
  • What is needed to achieve this and how can we support it?
RISC-V-based open source

• **Processor cores** (examples)
  - academia: BOOM, PULP, PicoRV, FemtoRV, SERV, …
  - industry: WD SweRV, XuanTie C/E90x, …
  - List at [https://riscv.org/exchanges/cores-socs/](https://riscv.org/exchanges/cores-socs/)

• **Firmware**
  - OpenSBI, coreboot, oreboot, u-boot

• **Operating systems and hypervisors**
  - Linux, seL4, FreeRTOS, Zephyr, Haiku, Plan 9/Inferno, Oberon
  - Xen, KVM (work in progress)

• **Compilers and development tools**
  - C/C++: gcc and clang/LLVM
  - Go, Rust, Java (WiP)
  - Debugging: gdb, lldb, OpenOCD

• **Simulation and emulation**
  - qemu, tinyemu, SystemC models, …
The RISC-V academia & training SIG

Educator’s challenge

• How can I integrate RISC-V in my courses?

The Mission of the **RISC-V Academia and Training Special Interest Group** is to

• promote and increase adoption of RISC-V in universities to prepare computer and electronic engineers for the challenges and opportunities of the future
• promote RISC-V as a common platform based on an open ISA
• support educators and students with resources to further their education on all levels of the RISC-V hardware and software stack, using the RISC-V ecosystem of solutions
Academia and training SIG activities

What we have learned so far?

• Educators want to integrate relevant and current concepts and practical work into the curriculum
• Support using accessible tools that allow students to learn skills that will be useful in their careers
• Require modular, reusable teaching materials
  • with graphics, real world examples, reference materials
  • reduce student workload – 2500 pp. data books too complex?

• **Wanted:** an augmentation tool and modules to use in teaching, not a fully-created course
  • Reusable materials that are time-consuming and error-prone to create, e.g. graphics, tables, etc.
Interested in industry opinion on

• What RISC-V related skills and theory do students need to have when they graduate?
  • how would these skills be measured?
• Is there a top 10 list of skills or tasks that students should be able to perform when they graduate?
• What are typical interview questions to this community?
• What’s a typical interview format?
  • importance of whiteboard-based interviews to experience interviewee thought processes or correct answers?
• "Shopping list" from industry
• Approach for possible assignments with real-world background
What can the A&T SIG provide?

- Access to **tools**, **dev boards**, **software**, **resources** and **guidance**
  - Central web site directory linking to offers from different academics and companies
- Tutorials from tool creators on how to best teach and use tools
- Provide points of contact for questions on topics
  - Up-to-date list of contact details?
- Quality assurance:
  - Peer-reviewed materials with a "stamp of approval"

**Idea:** Industry provides RISC-V with a contact for universities for collaboration opportunities and a way to get in touch when they have technical and hardware questions and requests
Existing teaching material (examples)

• **Textbooks**
  - Patterson/Hennessy: Computer Organization and Design, RISC-V Edition
  - Harris/Harris: Digital Design and Computer Architecture, RISC-V Edition
  - **Missing topics**: e.g. system software, and compilation/code generation

• **Courses**
  - The SIG is collecting a list of courses with slide sets, assignments, videos, labs, etc.

• **Software**
  - MIT’s xv6 OS
Experience with RISC-V in teaching

Operating system projects (at NTNU and now Uni Bamberg)
- Enable students to **learn about the HW/SW interface of RISC-V** by porting small real-world operating systems
  - Plan 9 and Inferno (WiP) on RV64GC
  - Oberon ([https://github.com/solbjorg/oberon-riscv](https://github.com/solbjorg/oberon-riscv)) on RV32IM
  - f9 microkernel (WiP) on ESP32-C3
- Enable students to **explore RISC-V ISA extensions**
  - Rust-based hypervisor using the H extension (WiP)
- **Operating systems engineering** course (at Uni Bamberg)
  - Design and implement ideas from OS research papers in xv6
    - e.g. virtual memory and virtualization, efficient syscalls…
  - RISC-V platform: used
  - Emulation (qemu) and real hardware (D1 Nezha boards)
RISC-V hardware used

- Nezha and LicheeRV boards based on Allwinner D1 SoC
  https://linux-sunxi.org/D1 – $99 vs. $25
- Single-core 1 GHz 64-bit RV64GC, 512 MB–2 GB RAM
- Peripherals: USB, Wifi/BT, Ethernet, audio, multiple GPIO
- Our xv6 OS port running "bare metal" on D1 platforms:
  https://github.com/michaelengel/xv6-d1
What’s next?

• Consultations with industry and academia about topics of interest

• Setup of a central web presence at riscv.org with links to submitted resources

• Creation of a RISC-V professional certification program

• Interested in contributing? Please get in touch!
  • Courses, materials, hardware, software IP, feedback, …

Michael Engel – michael.engel@uni-bamberg.de
Megan Lehn – megan@riscv.org
Additional material: RISC-V boards

- Development boards with different ASICs
  - from embedded microcontroller to multicore
  - from $5 – $1000

- In addition, FPGA boards with hard or soft RISC-V cores
  - Microchip Polarfire FPGAs (4+1 hard RV64GV cores + FPGA)
  - Open source projects for all FPGA vendors
  - Simple system building tools
    - LiteX, FuseSoC
    - Open source hardware synthesis: F4PGA
Available RISC-V hardware

- SiFive FE310
  - RV32IMAC, 150 MHz, 16 KB Instruction Cache, 16 KB Data Scratchpad, external flash
  - GPIO, UART, SPI, PWM
- Available on the Dr Who inventor kit
  - WiFi & BT via external espressif ESP32 module
  - Light, acceleration sensor, compass, pushbuttons
  - Color LED matrix (6x8)
## Available RISC-V hardware

<table>
<thead>
<tr>
<th></th>
<th>GigaDevice GD32VF103</th>
<th>WCH CH32V307</th>
<th>Bouffalo Labs BL602/604</th>
<th>Espressif ESP32-C3</th>
<th>Kendryte K210</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Core speed</strong></td>
<td>RV32IMAC 108 MHz</td>
<td>RV32IMAC 144 MHz</td>
<td>RV32IMAC 192 MHz</td>
<td>RV32IMAC 160 MHz</td>
<td>dual RV64GC 400 MHz</td>
</tr>
<tr>
<td><strong>Flash RAM</strong></td>
<td>128 kB 32 kB</td>
<td>256 kB 64 kB</td>
<td>0-4 MB 277 kB</td>
<td>– 400 kB</td>
<td>– (external) 8 MB</td>
</tr>
<tr>
<td><strong>Peripherals</strong></td>
<td>USB, GPIO, UART, IIC, SPI, I2S, PWM</td>
<td>USB, GPIO, UART, IIC, SPI, I2S, PWM</td>
<td>USB, GPIO, UART, IIC, SPI, I2S, PWM</td>
<td>GPIO, UART, IIC, SPI, I2S, PWM</td>
<td>USB, GPIO, UART, IIC, SPI, I2S, PWM</td>
</tr>
<tr>
<td><strong>Radio / network</strong></td>
<td>–</td>
<td>1 Gbps MAC 10 Mbps PHY 2 x CAN 2.0B</td>
<td>WiFi 802.11b/g/n BT5 (LE)</td>
<td>WiFi 802.11b/g/n BT5 (LE)</td>
<td>–</td>
</tr>
<tr>
<td><strong>Special I/O</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Camera, mic. array</td>
</tr>
<tr>
<td><strong>Other</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MMU Neural accel.</td>
</tr>
</tbody>
</table>
Available RISC-V hardware

- SiFive HiFive Unmatched
- SiFive Freedom U740 SoC (quad RV64GC, 1.5 GHz)
- 16 GB DDR4 RAM
- Gigabit Ethernet
- 4x USB 3.2 Gen 1 Type A
- x16 PCIe® Gen 3 Expansion Slot (8-lanes Useable)
- M.2 M-Key Slot (PCIe Gen 3 x4) for NVME 2280 SSD Module
- M.2 E-Key Slot (PCIe Gen 3 x1) for Wi-Fi / Bluetooth Module