

lowRISC: Plans for RISC-V in 2016



lowRISC

Alex Bradbury

asb@lowrisc.org @asbradbury @lowRISC

06/01/15

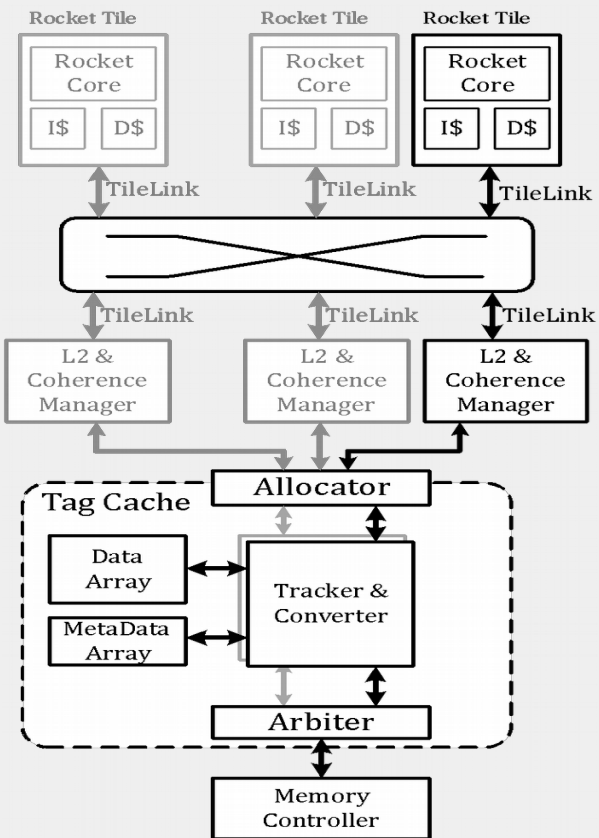
What is lowRISC

- A not-for-profit, open project. 'Linux of the hardware world'
- An open source SoC that 'runs Linux well'
- A platform, on which others can base derivative designs
- Implements the RISC-V ISA (application cores are Rocket derivatives)
- Follows on from Raspberry Pi experience
- Technical focuses: flexibility and security
- Core team based at the University of Cambridge Computer Lab

The lowRISC approach

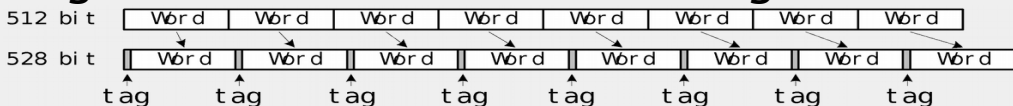
- Produce low-cost development boards
 - 'Raspberry Pi for grownups'
- Regular tape-outs. **Not just a one-off effort**
- Form collaborations. We can't do this alone
- Initial funding from private donor, recently from Google. Eventually self-sustaining
- Simple, permissive licensing

2015 in review: Tagged memory



Credit: Wei Song

Augment each 64-bit word with tag bits



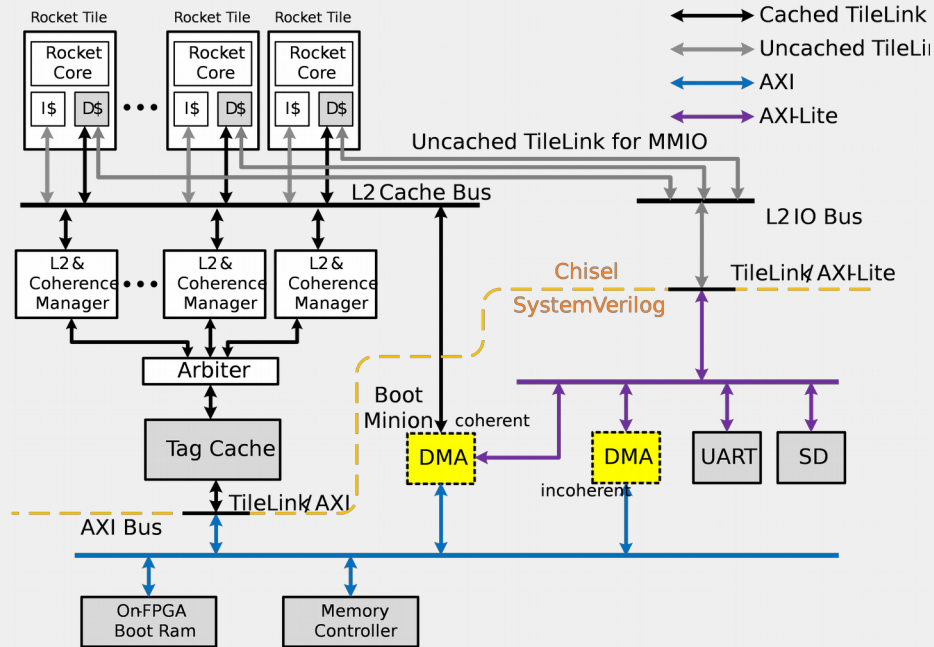
Motivation: security and other applications

- An end to control-flow hijacking attacks
- Flexible security policies. Also uses for debug, performance monitoring
- Initial implementation and extensive documentation released

See lowrisc.org and previous RISC-V workshop presentations

2015 in review: Untethered SoC

- See Wei's talk at 11.15



2015 in review: Summer of Code

- Google + lowRISC Summer of Code supported 6 projects + 2 local interns
- To pick a few:
 - A port of the seL4 verified microkernel to RISC-V (Hesham Almatary)
 - Porting the jor1k emulator to RISC-V (Prannoy Pilligundla)
 - TCP/IP Offload to Minion Cores using Rump Kernels (Sebastian Wicki)

lowRISC in 2016

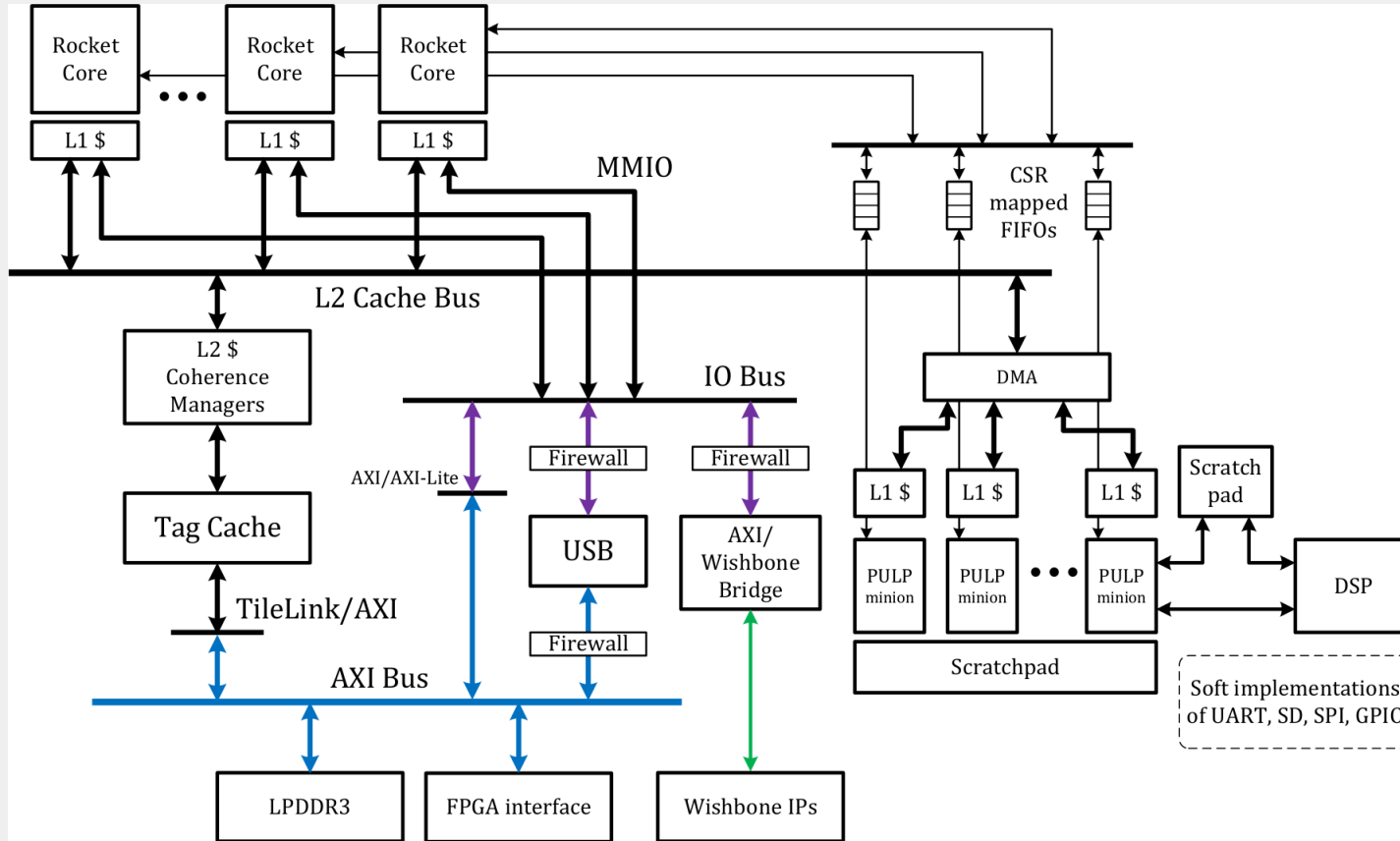
Continuing untethering work

- Kernel changes
- Replace FPGA vendor-provided IP with vendor-neutral, open peripherals (help wanted!)
- Interrupt controller
 - BERI PIC
 - See ML post “Choosing a de facto standard programmable interrupt controller”
 - Samuel Falvo (Kestrel) has interesting ideas on scaling it down to smaller systems

2016 test chip

- Note: all subject to change. Comments and advice welcome
- Tape out by the end of 2016
- 3mm x 3mm 28nm die, wire-bond BGA package
- 4 cores (evaluating BOOM), each with 32KiB I+D\$
 - BERI PIC, tagged memory, >1GHz, run-control+trace debug, RV64G+C
- 512KiB shared L2
- 128KiB tag cache
- LPDDR3 memory controller+PHY, 32-bit wide
- 8 Minion cores (PULP-based) with shim. 500MHz+. Provide SDHC, SPI, I2C, I2S, UART
- USB 2.0 host PHY and controller
- High-speed I/O to FPGA (tbd, input very welcome)

2016 test chip



Third-party IP

- Ultimate goal: all digital logic is completely open-source
- Much like the GNU project's work on a free UNIX, this will be an incremental process
- Provide hardware firewalls
- The potential for open-source PHYs seems much weaker than for digital logic (economics, heavy IP protection of process technology)
 - Dissenting opinions welcome :)

Coming up in 2016

- Re-integrate tagged memory. Optimisations. Further software work
- Integrate minion cores
- Shim implementation
- Integration of third-party IP
- Determine IC packaging solution
- Benchmarking and performance analysis
- Verification, bug hunting (particularly for multi-core)
- Trace debug (Stefan Wallentowitz, opensocdebug.org)

Conclusion

- We will have succeeded when the use of open source hardware designs is as common and accepted as for open source software
- Thank you
 - Donors, contributors, collaborators, technical advisory board, supporters
- See also: lowrisc.org, our mailing list, phab.lowrisc.org, [@lowRISC](https://twitter.com/lowRISC)
- Email: asb@lowrisc.org
- Join our team - job advert soon, informal enquiries to Robert.Mullins@cl.cam.ac.uk
- Stickers!