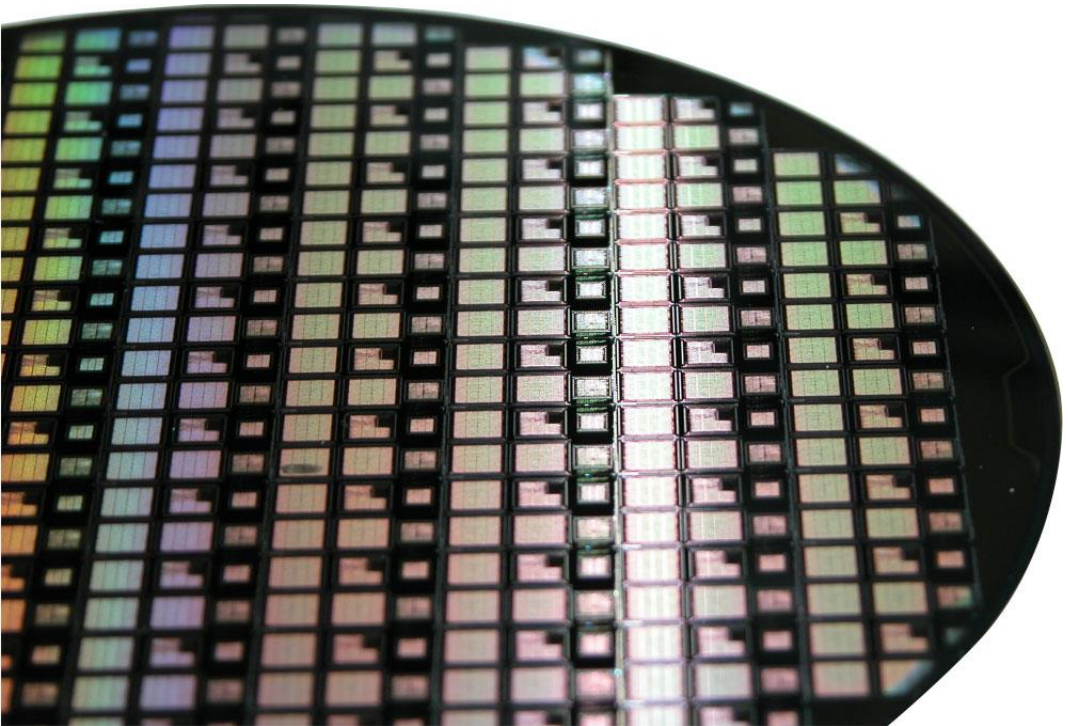
ETH zürich

Ariane: An open-source 64-bit RISC-V Application-Class Processor and latest Improvements



9 May 2018

Florian Zaruba

Professor Luca Benini
Integrated Systems Laboratory
ETH Zürich

RISC-V cores under development at IIS

32 bit

Low Cost Core

- **Zero-riscy**
  - RV32-ICM
- **Micro-riscy**
  - RV32-CE

Core with DSP enhancements

- **RI5CY**
  - RV32-ICMX
    - SIMD
    - HW loops
    - Bit manipulation
    - Fixed point

Floating-point capable Core

- **RI5CY + FPU**
  - RV32-ICMFX

64 bit

Linux capable Core

- **Ariane**
  - RV64-IC(MA)
  - Full privileged specification

A new perspective: Application class processor

- **Currently: Bare metal**
- **Virtual Memory**
  - Multi-program environment
  - Efficient sharing and protection
- **Operating System**
  - Highly sequential code
  - Increase frequency to gain performance
- **Large software infrastructure**
  - Drivers for hardware (PCIe, ethernet)
  - Application SW (e.g.: Tensorflow, …)

- **Larger address space** (64-bit)
- **Requires more hardware support**
  - MMU (TLBs, PTW)
  - Privilege Levels
  - More Exceptions (page fault, illegal access)

→ **Ariane an application class processor**

ARIANE: Linux Capable 64-bit core

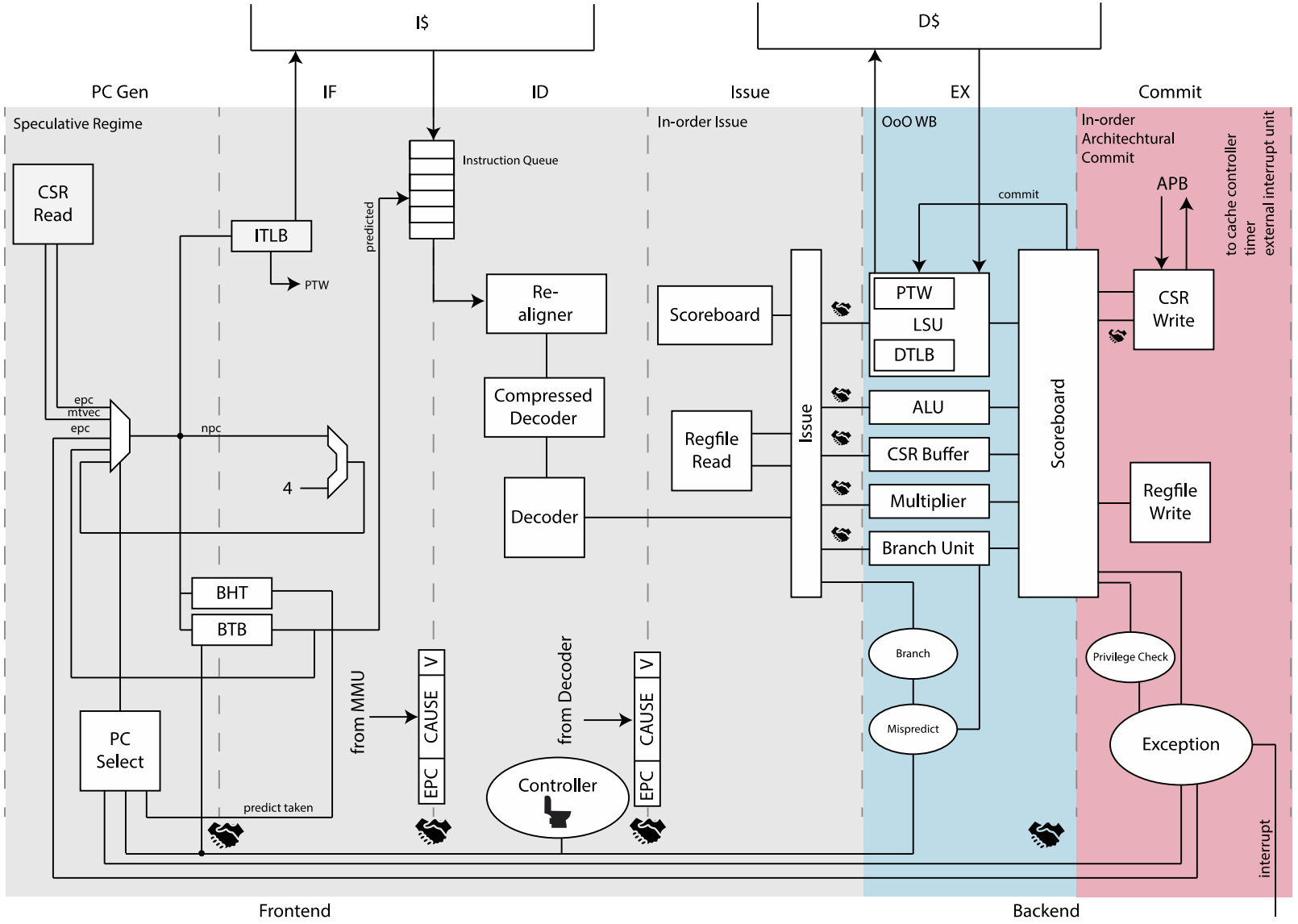
- **Application class processor**
- **Linux Capable**
  - M, S and U privilege modes
  - TLB
  - Tightly integrated D$ and I$
  - Hardware PTW
- **Optimized for performance**
  - Frequency: > 1.5 GHz (22 FDX)
  - Area: 185 kGE
  - Critical path: ~ 25 logic levels
- **6-stage pipeline**
  - In-order issue
  - Out-of-order write-back
  - In-order commit
- **Branch-prediction**
- **Scoreboarding**
- **Designed for extendability**

Other open-source Linux capable RISC-V Cores

- **Rocket:** (5-stages, in-order)
  - RV64-GC
  - Part of generator written in Chisel
- **Boom:** (super-scalar, out-of-order)
  - RV64-GC
  - Configurable issue-width
- **SHAKTI C-Class**:
  - RV64G
  - BlueSpec Verilog
- **Ariane:** (6-stages, in-order)
  - RV64-IMC
  - System Verilog
- **Why develop yet another core?**
  - We do not want a SoC generator
  - We are a research group – try out different things!
  - Don’t be governed by a 3rd party
  - Because it is fun 😀

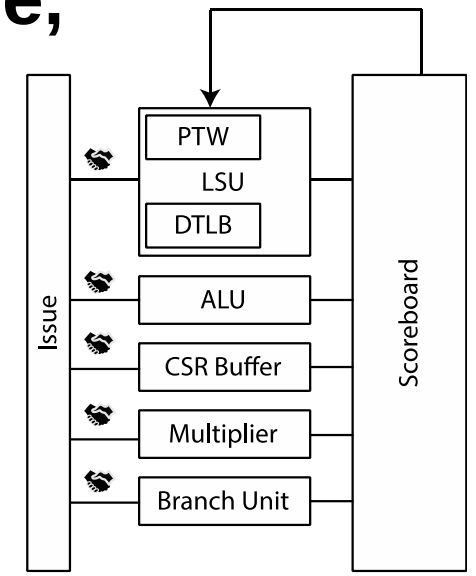
First Implementation…

- …after 4 months!
- External data cache and instruction cache
- Scoreboard
- Basic Branch prediction
- Modular design



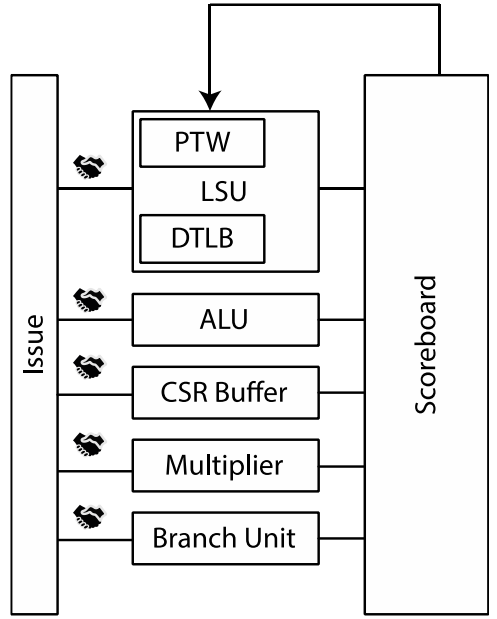
Functional Overview

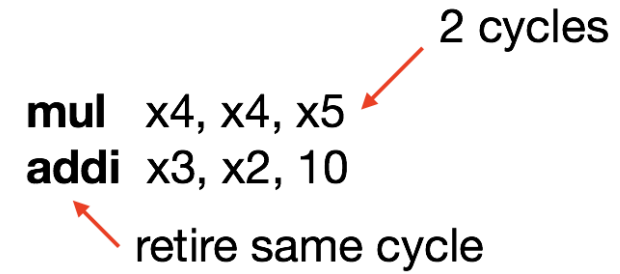
- **6-stage pipeline**
- **Branch prediction: 2-bit saturation counter, BHT**
- **Scoreboard manages dependencies and re-orders instructions**
- **All instructions until commit stage are speculative, single point for committing instructions**
  - Easier to implement interrupts and debug functionality
  - Easier to do tandem verification
- **Designed for higher performance in the future:**
  - Dual Issue and/or OoO issue
  - Modular functional unit design



Scoreboarding and Merged Re-order Buffer

- Hide latency of multi-cycle instructions
- Clean and modular interface to functional units ➔ scalability (FPU)
- Add issue port: Dual-Issue implementation
- Split execution into four steps:
  - **Issue**: Relatively complex issue logic (extra pipeline-stage)
  - **Read Operands**: From register file or forwarded
  - **Execute**
  - **Write Back**: Mitigate structural hazards on write-back path
- Implemented as a **circular buffer**





Verification Strategy

- **RISC-V tests**
  - They help to get reasonably fast up and running
- **Torture test framework**
  - More thorough checking
  - (Known) weaknesses on compressed instructions
- **Running applications on the FPGA (e.g.: booting Linux)**

→The existing RISC-V (test) infrastructure massively helps in kick starting the development of a new design

- **Still: Verification is not exhaustive looking into more alternatives**

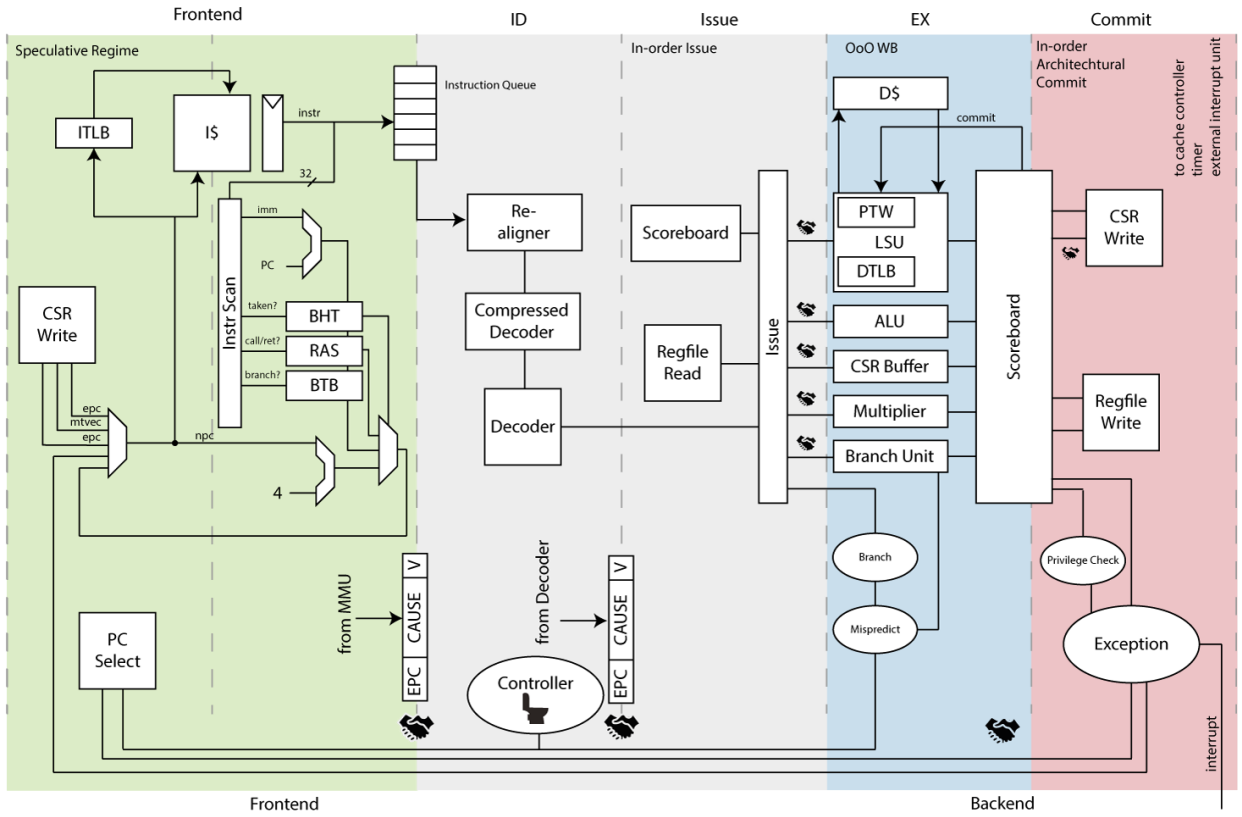
Open-sourcing

- **Ariane has been open-sourced in February 2018**
  - Continue development on our public GitHub servers
  - We've been the most trending SV a couple of weeks ☺
- **We provide a Verilator port for an easy first evaluation**
  - Works with a forked version of riscv-fesvr
- **Unstable development can be found in the ariane_next branch**
  - Fixes are being maintained on both branches
- **Travis CI**
  - Relies on Verilator and riscv-test to pass
  - We will tighten CI in the future to ensure only high quality commits being merged

Contributions

- **We've got some non-trivial external contributions already**
  - Moving branch-comparisons to ALU
  - Issues and bug fixes (for example related to flushing)
  - Testing and fixing the simulation environment
- **They help me/us getting more work done, making the design continuously better**
- **Sometimes it may take some time until I can merge a PR – bare with me!**
- **Every contribution makes this open-source project more successful**

Latest Improvements



- **Merged L1 D$ and I$ into the core**
- **Completely revised instruction front-end**
  - First implementation was very naïve
  - Virtually Indexed – Physically tagged I$
  - Split BTB and BHT – merged version was suffering a lot of capacity issues
  - Added RAS
- **Re-naming in issue stage**

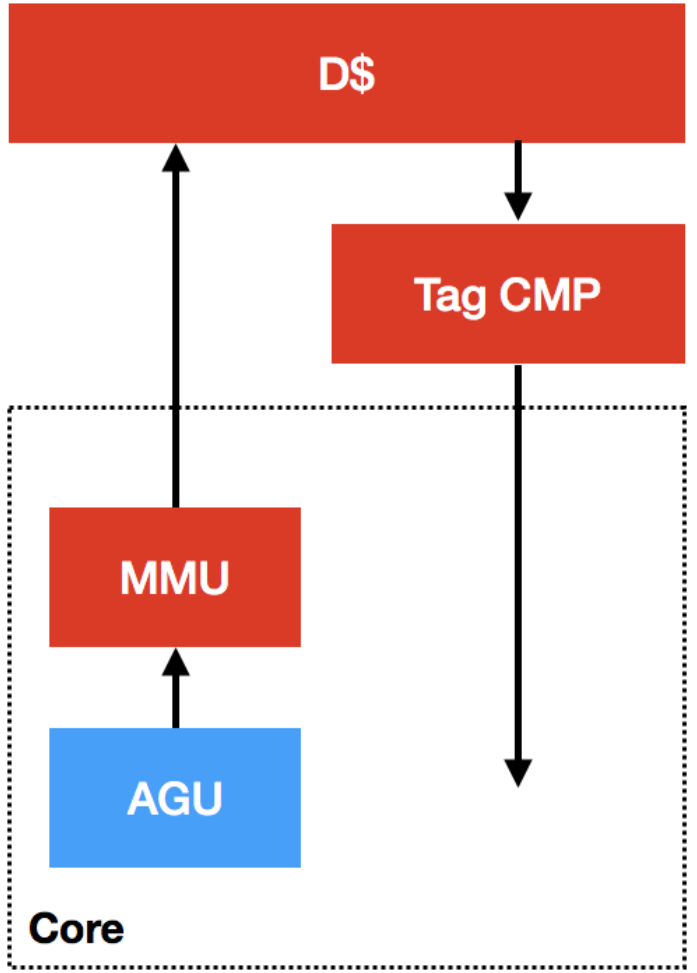
Improvement Details

- **Scan instruction words for ctrl flow changes (180 GE)**
- **New return address stack (RAS, 2-entries – 1 kGE)**
  - Cheap to detect calls and returns
  - Already depth 2 increases IPC* by 20% (323k cycles)
- **Unconditional jumps are resolved immediately**
  - They have been handled in the ex-stage before
  - Needs another adder
  - Reduced pressure on BHT
  - IPC* increased further by 11% (291k cycles)
- **Simple re-naming (1.5 kGE)**
  - Resolves WAW dependencies (288k cycles)
- **No negative impact on timing**

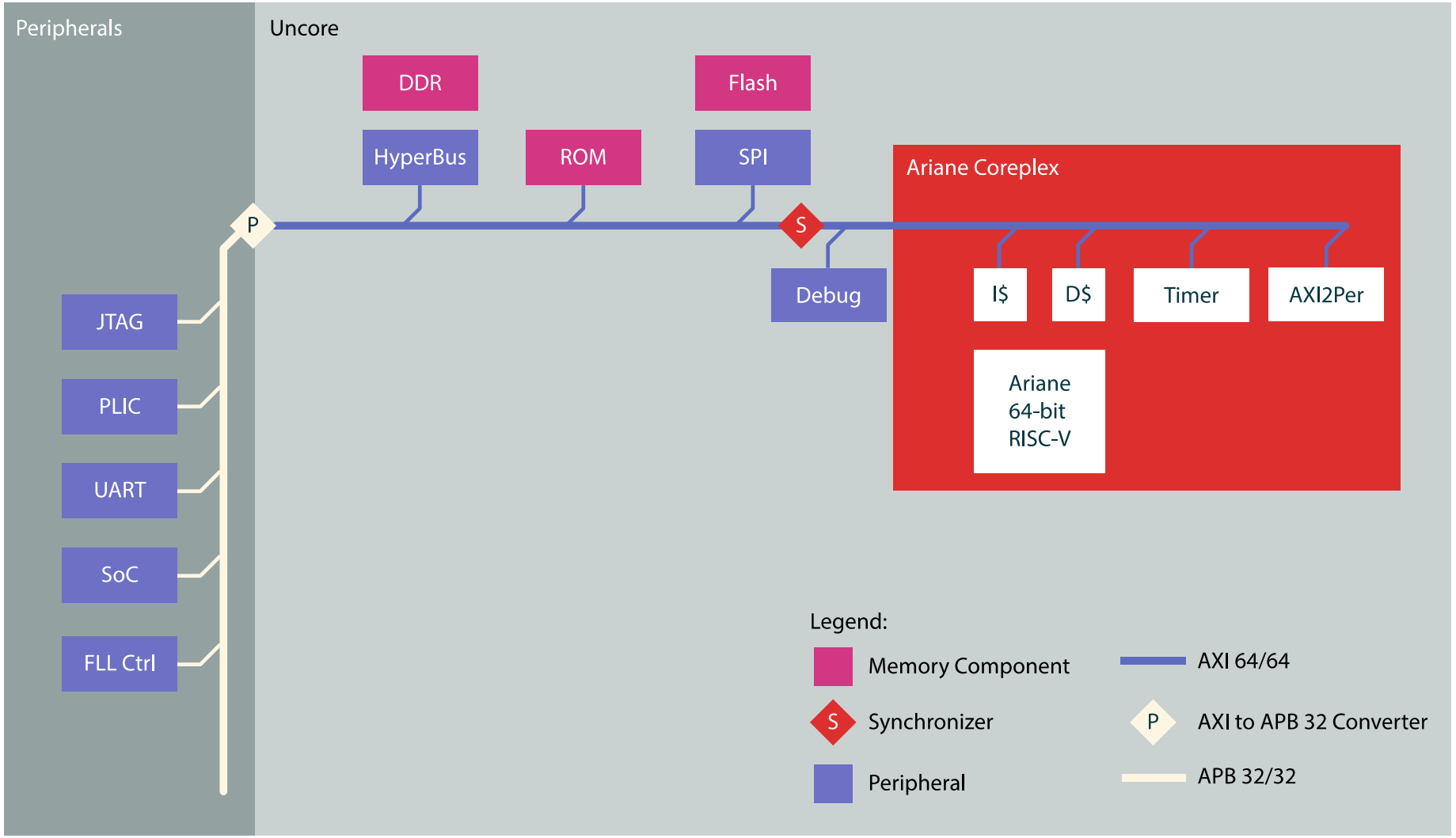
* measured on Dhrystone benchmark (387k cycles)

Critical Paths (Memory Interfaces)

- **Load** and **stores** are very **common** in RISC architectures
- Latency hurts in in-order designs
- **Caches add** (costly) tag-comparison
- **Address translation adds** to this already critical path
- A fast CPU design needs to account for these effects as much as possible
  - Virtually indexed, physically tagged caches
  - De-skewing



Kerbin: Proof of concept SoC for Ariane



SoC

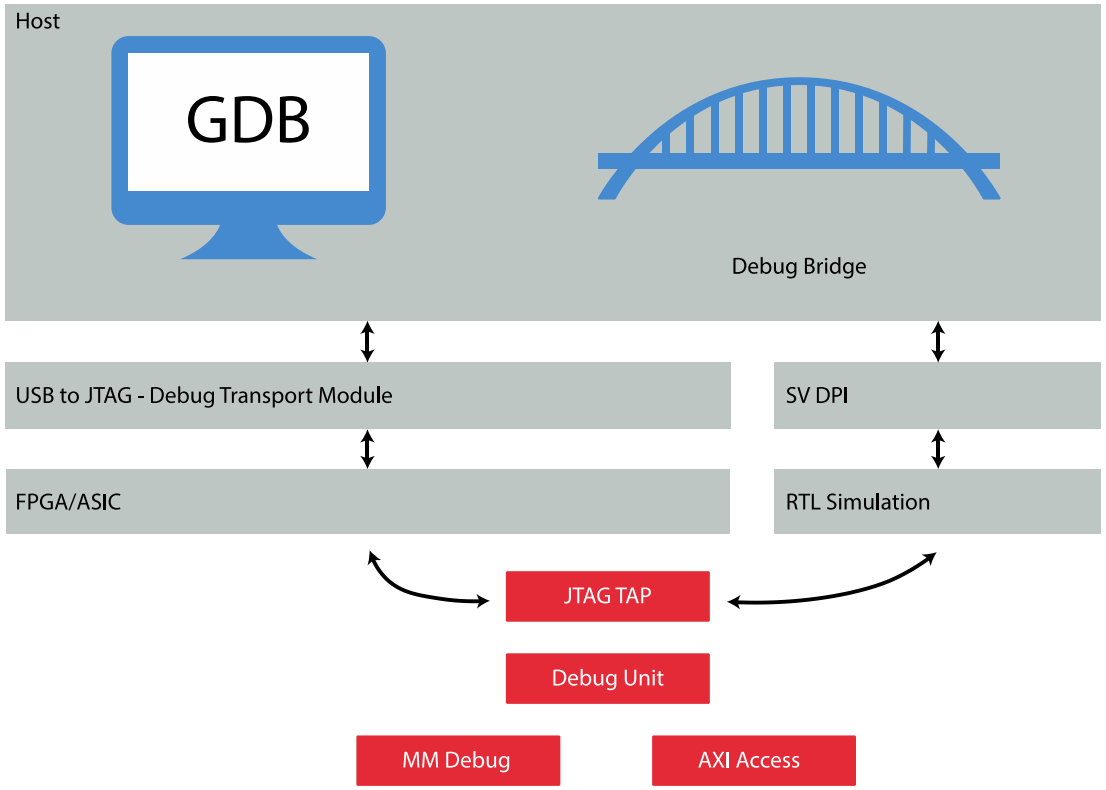
- PULP Peripherals

CLUSTER

- 64-bit interconnect
- Debug support

Full Debug support

- Leveraging existing infrastructure: **RISC-V GDB**
- **Debug Bridge** to communicate with hardware
- Allows for:
  - run-control
  - single-step
  - inspection
  - (hardware) breakpoints
- Essential for SW debug and hardware bring-up
- Trace debugging capabilities are planned

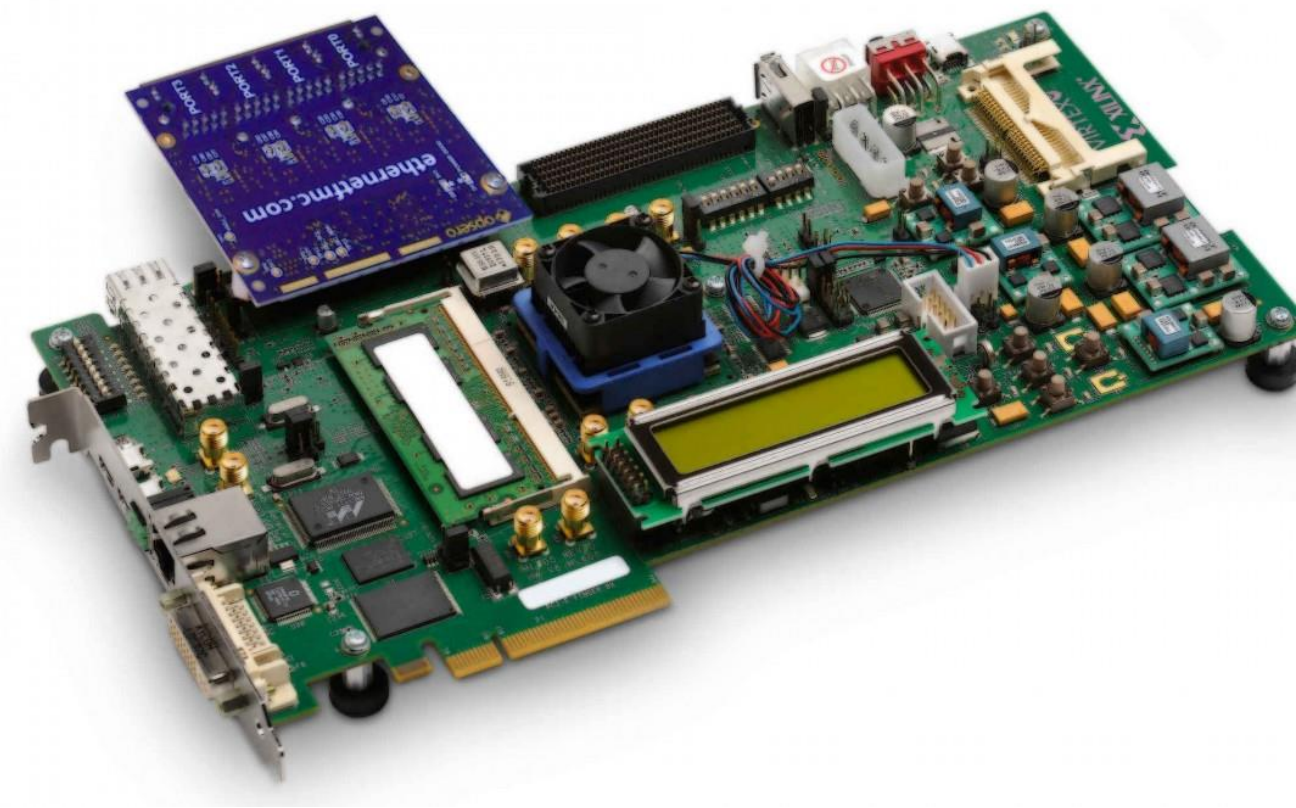


First Milestone: Booting Linux

- **After 5 months managed to boot to user space on the FPGA**
- **Preloading the RAM**
  - only initramfs
- **At the time of development atomics where not mandatory**
  - Busybox needed them anyway
  - Emulation in ERL (fork of BBL)
- **A lot has changed since then**
  - want to open-source the SoC in the near future

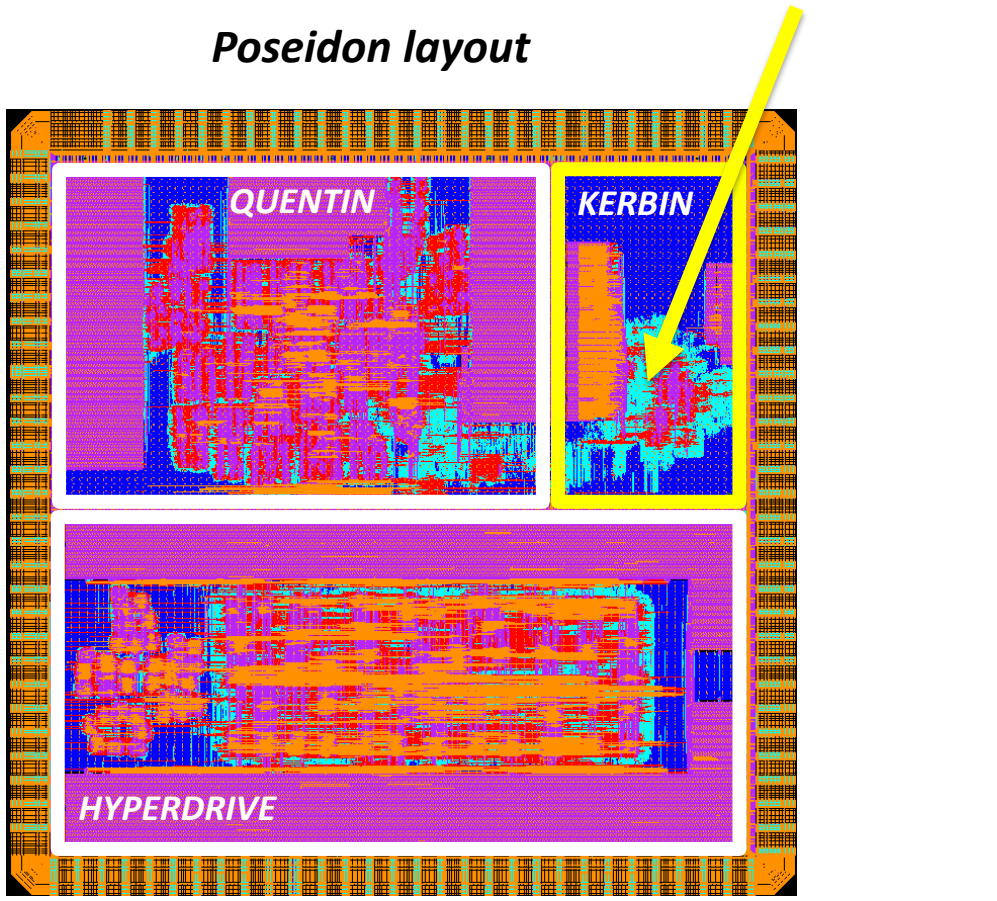
FPGA Mapping

- **Full FPGA implementation**
  - Xilinx Vertex 7 – VC707
  - **Core:** 50 – 100 MHz
  - **Core:** 15 kLUTs
  - 1 GB DDR3
- FPGA implementation allows for fast prototyping – HW/SW codesign
- Area allows for exploration of multi-processor systems



ASIC Mapping (Poseidon)

- **Ariane has been taped-out December 2017 in Globalfoundries 22nm FDX**
- **Silicon implementation in GF22FDX, mixed LVT and SLVT libraries.**
- **The system features 16 kByte of instruction and 32 kByte of data cache.**
  - D$: 8-way, 4 kByte
  - I$: 4-way, 4 kByte
- **Timing closure: 910 MHz @ SSG, 125/-40 °C, 0.72V – NO BB**
- **Area: 0.23 mm2 – 175 kGE**
- **I/O and L2 are shared with Quentin**

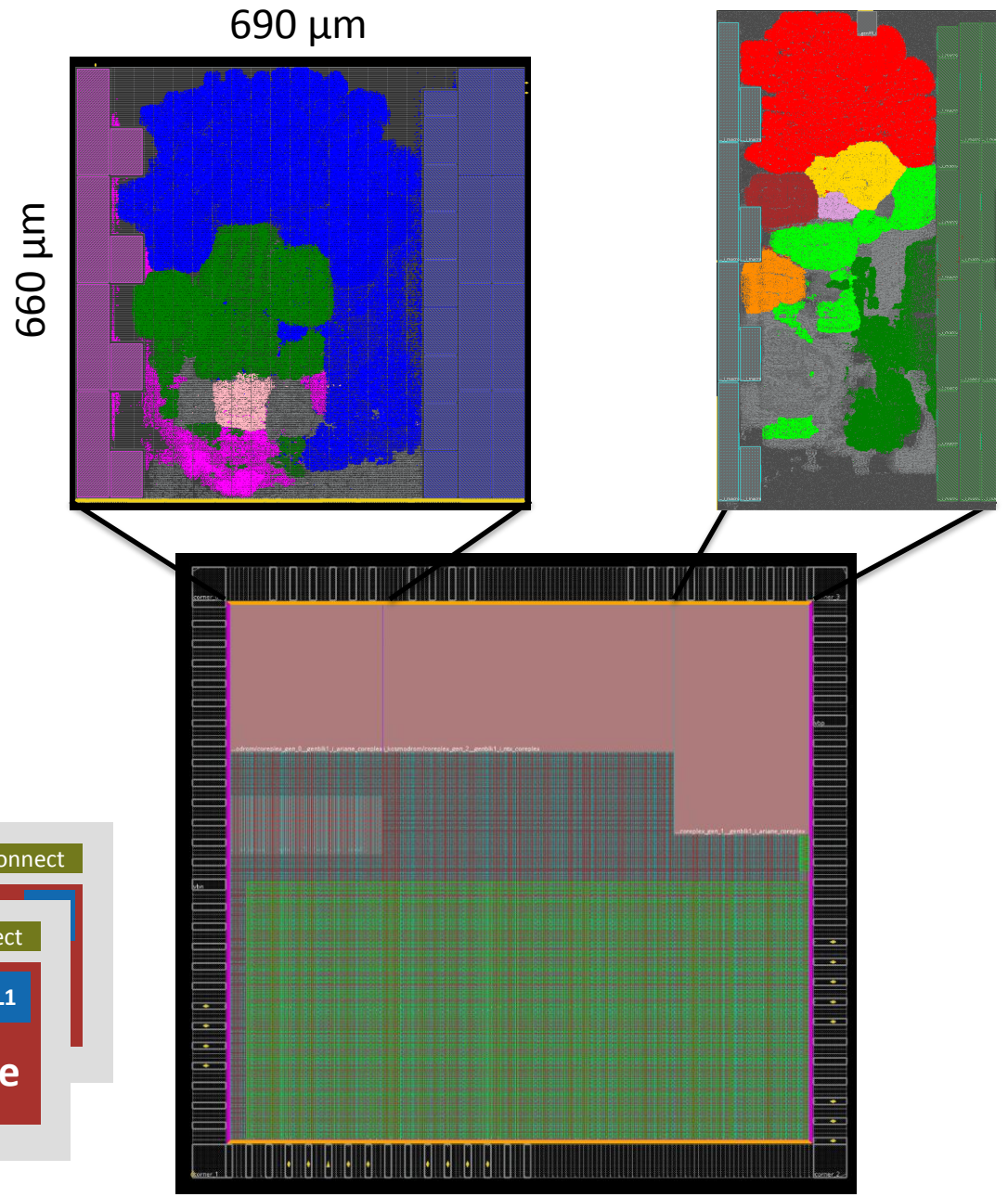


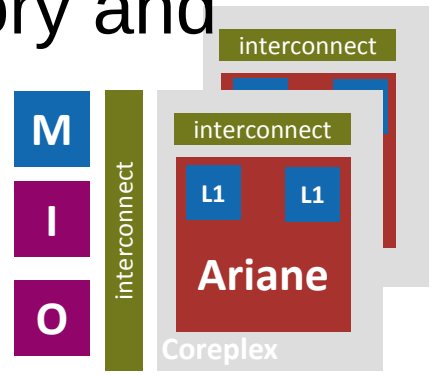
Challenges in Higher Speed Designs (Physical Design)

- **Memories are slow compared to logic:**
  - Use fast cache cuts (not always available)
  - We need reasonably large L1 memories (especially data cache)
  - High set-associativity leads to congestion issues
- **Clock distribution:**
  - Shielding: takes away valuable routing resources
- **Useful Skew:**
  - Needed to balance request and response path
  - **Request path: fast** (only address calculation)
  - **Response path: slow** (address translation, tag comparison, way select)
  - Too much de-skewing leads to a lot of hold-time violation on the memories

Kosmodrom

- **We are currently working on another tape-out in GF22** - heterogeneous dual-core system
  - A high performance variant (> 1 GHz @ 0.8V)
  - A ultra low power version @0.5V
- Coreplices share memory and I/O
- FP Accelerator

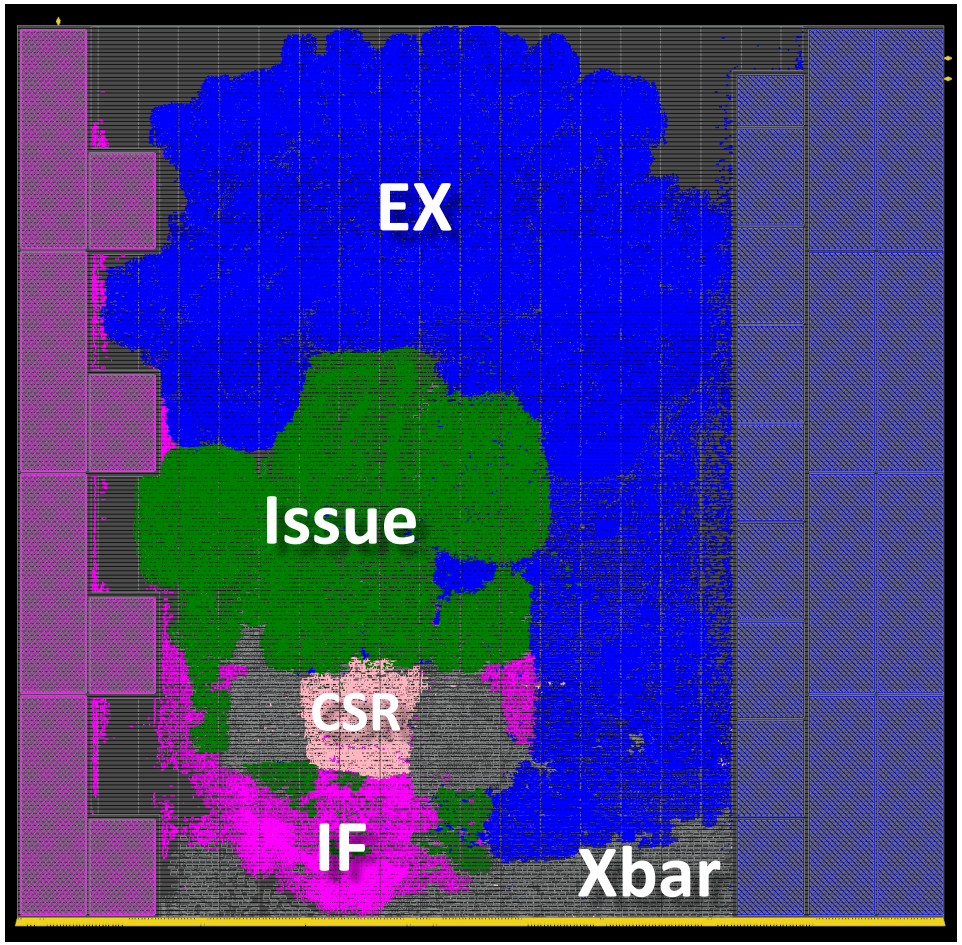




Kosmodrom – A closer Look

690 µm

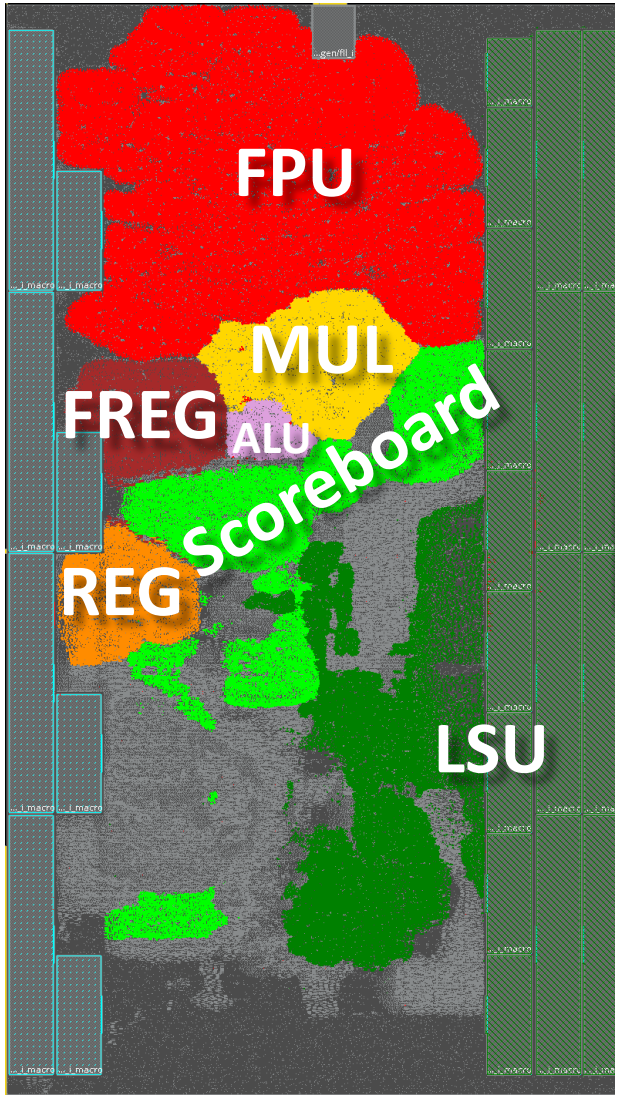
660 µm



High Performance Implementation
(8T; 0.8V; 20, 24, 28 S/LVT)

610 µm

1000 µm



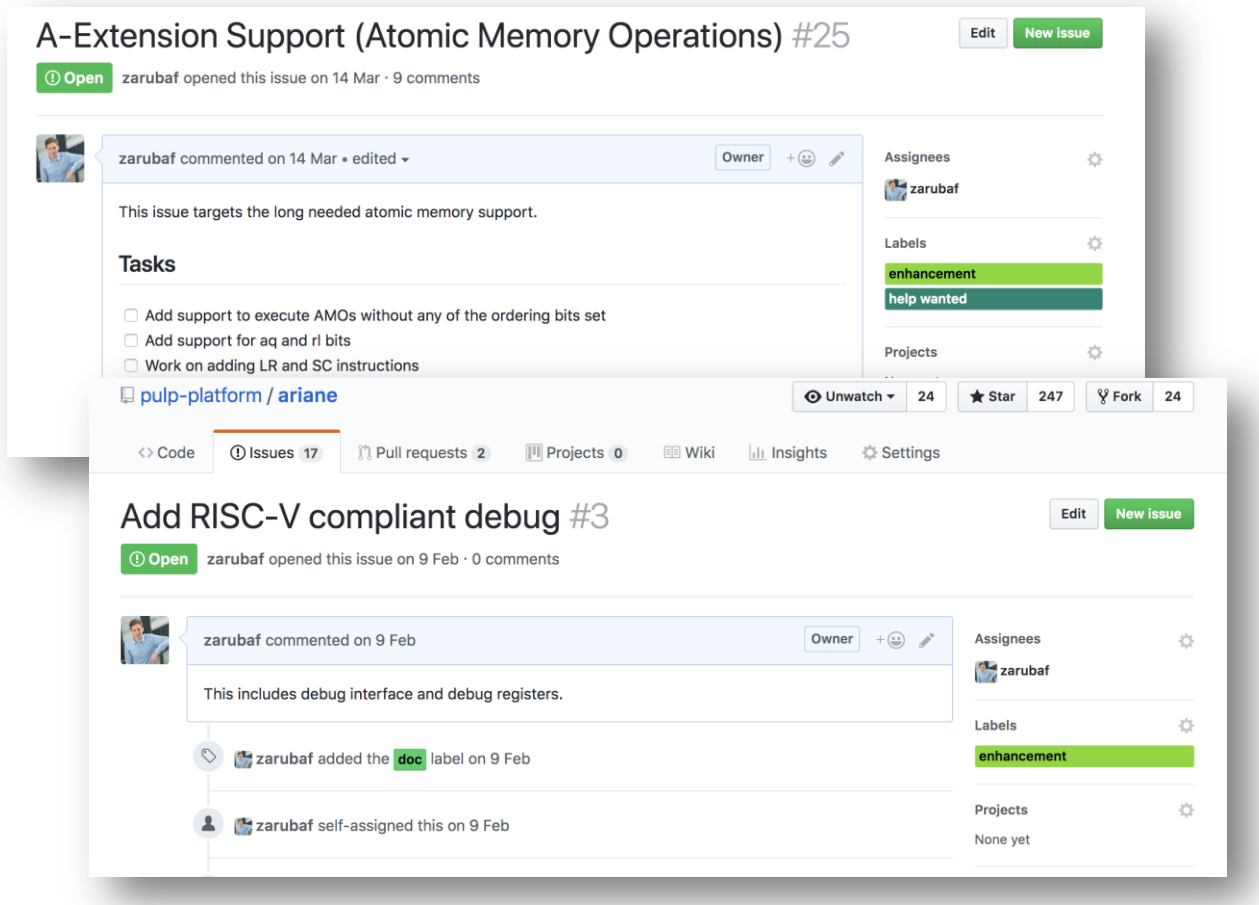
ULP Implementation
(7.5T; 0.5V; 28, 32, 36 S/LVT)

What we are currently working on…

- **Currently working on supporting F and D-extension**
- **Reduced precision vector operations** (IEEE 754 – FP8, FP16, FP16alt and FP32)
- **Stand-alone floating point unit** (~ 200 kGE) will be released in the next months
- **Improved integer divider (SRT-4)**
- **Vector Unit (Matheus, talk to him)**
- **Hardware support for atomic memory operations** (currently emulated in BBL)

Help Wanted…

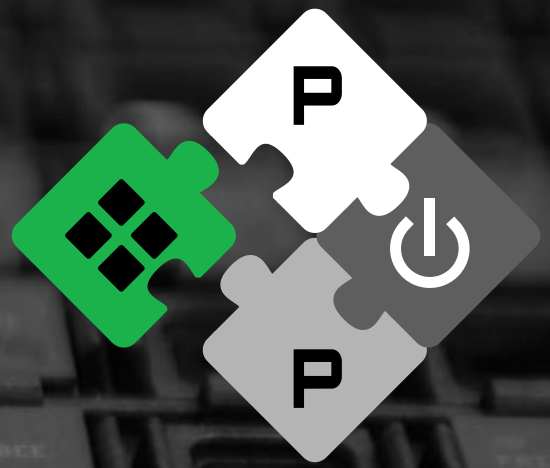
- **Managed via Github's issue tracker**
- **Improved branch-predictors, dual-issue, multithreading,…**
- **Cache-coherent interconnect and caches**
- **Support for official RISC-V debug**
- **Use it and give me honest feedback!**
- **Approach me at the workshop!**



Questions?

www.pulp-platform.org

@pulp_platform



PULP

Parallel Ultra Low Power

Florian Zaruba[2], Davide Rossi[1], Antonio Pullini[2], Francesco Conti[1], Michael Gautschi[2],
Frank K. Gürkaynak[2], Florian Glaser[2], Stefan Mach[2], Giovanni Rovere[2], Igor Loi[1]
Davide Schiavone[2], Germain Haugou[2], Manuele Rusci[1], Alessandro Capotondi[1],
Giuseppe Tagliavini[1], Daniele Palossi[2], Andrea Marongiu[1,2], Fabio Montagna[1],
Simone Benatti[1], Eric Flamand[2], Fabian Schuiki[2], Andreas Kurth[2], Luca Benini[1,2]

[1]*Department of Electrical, Electronic and Information Engineering*

ETH zürich

[2]*Integrated Systems Laboratory*