PULPino: A small single-core RISC-V SoC

Andreas Traber, Florian Zaruba, Sven Stucki, Antonio Pullini, Germain Haugou, Eric Flamand, Frank K. Gürkaynak, Luca Benini

Our group: Prof. Luca Benini

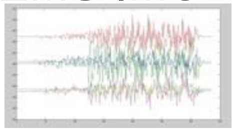
- ETH Zurich, Integrated Systems Lab (IIS)
- University of Bologna, Micrel Lab
- Around 40 people
  - Most involved in the PULP project
- Close Collaborations
  - Politecnico di Milano
  - CEA-LETI
  - EPFL
- Industrial Support from STMicroelectronics
  - Silicon access to 28nm FDSOI

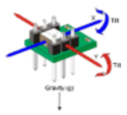
ETH zürich | ALMA MATER STUDIORUM UNIVERSITA DI BOLOGNA

Computing for the Internet of Things

Sense

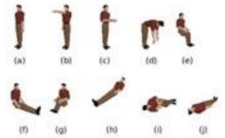
MEMS IMU



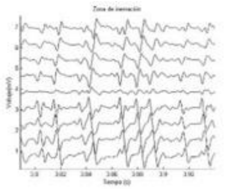


MEMS Microphone

ULP Imager

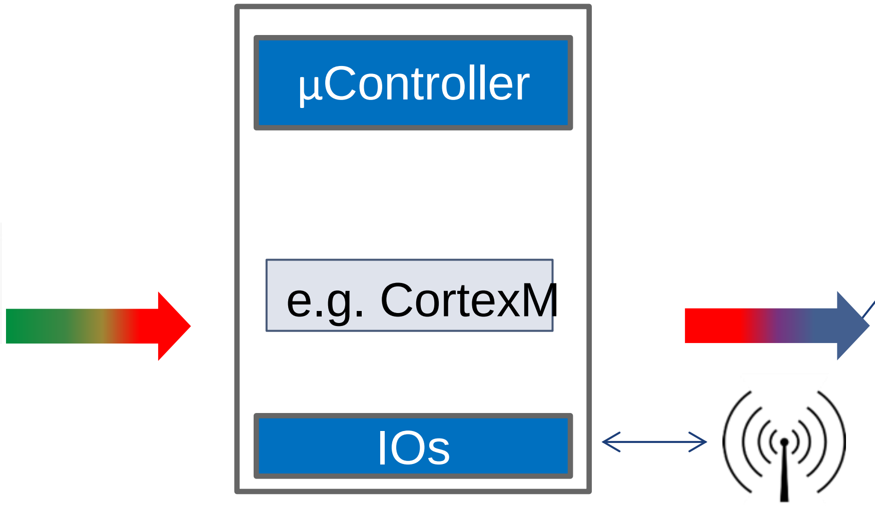


EMG/ECG/EIT



100 µW ÷ 2 mW

Analyze and Classify



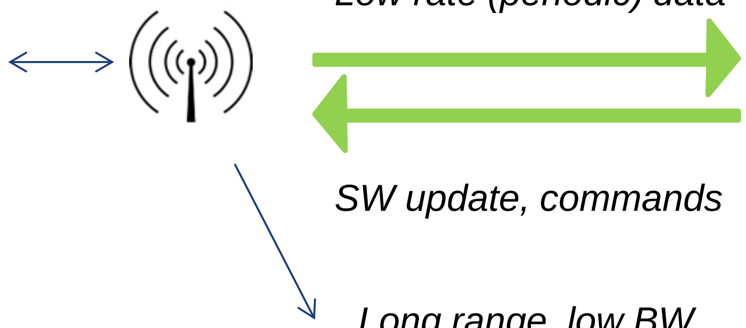
1 ÷ 25 MOPS
1 ÷ 10 mW

Battery + Harvesting powered
→ a few mW power envelope

Transmit

Short range, medium BW

Low rate (periodic) data



SW update, commands

Long range, low BW

LoRa SEMTECH

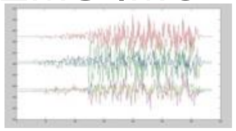
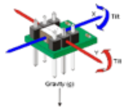
Idle: ~1µW
Active: ~ 50mW

ETH zürich — ALMA MATER STUDIORUM UNIVERSITA DI BOLOGNA

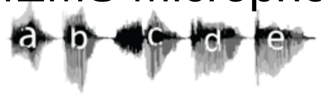
Computing for the Internet of Things

Sense

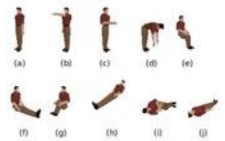
MEMS IMU

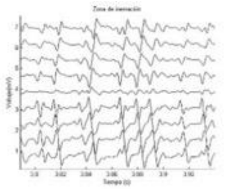
MEMS Microphone



ULP Imager

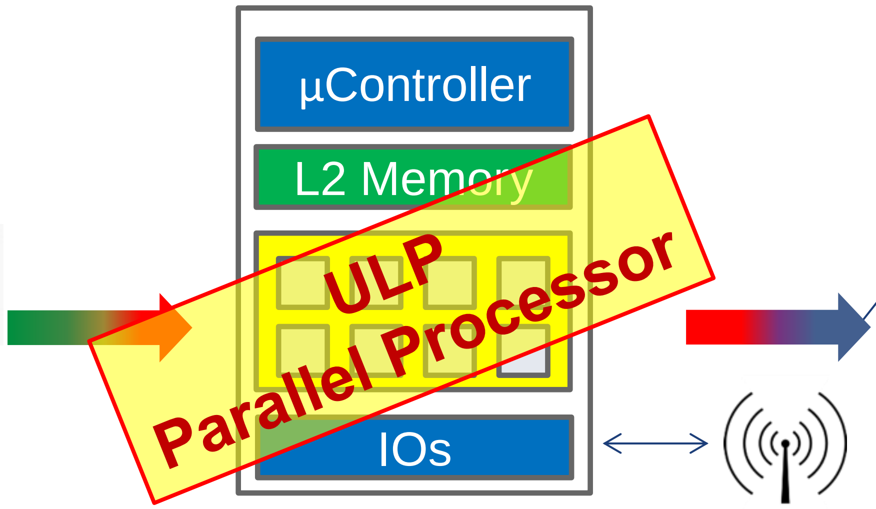


EMG/ECG/EIT



100 µW ÷ 2 mW

Analyze and Classify



1 ÷ 2000 MOPS
1 ÷ 10 mW

Battery + Harvesting powered
→ a few mW power envelope

Transmit

Short range, medium BW

Low rate (periodic) data

SW update, commands

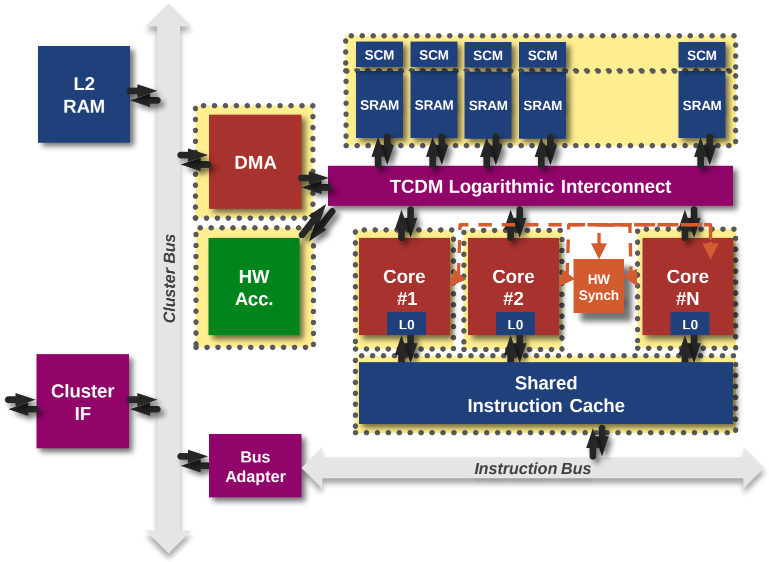
Long range, low BW

LoRa SEMTECH

Idle: ~1µW
Active: ~ 50mW

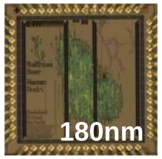
PULP: Parallel Ultra-Low-Power Processor

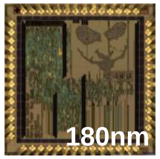
- Exploit parallelism
  - Multiple small cores organized in clusters
  - Share memory within the cluster
  - Multiple clusters per chip
- Simple but efficient processor cores
  - Based on OpenRISC/RISC-V
  - Custom ISA extensions
- Dedicated accelerators
- Multiple Technologies
  - Near-threshold operation

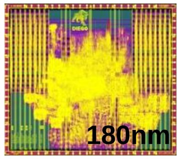


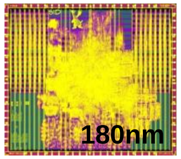
PULP related chips

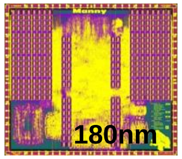
- **Main PULP chips** (ST 28nm FDSOI)
  - PULPv1
  - PULPv2
  - PULPv3 (in production)
  - PULPv4 (in progress)
- **PULP development** (UMC 65nm)
  - Artemis - IEEE 754 FPU
  - Hecate - Shared FPU
  - Selene - Logarithic Number System FPU
  - Diana - Approximate FPU
  - Mia Wallace – full system
  - Imperio - PULPino chip (Jan 2016)
  - Fulmine – Secure cluster (Jan 2016)
- **RISC-V based systems** (GF 28nm)
  - Honey Bunny
- **Early building blocks** (UMC180)
  - Sir10us
  - Or10n
- **Mixed-signal systems** (SMIC 130nm)
  - VivoSoC
  - EdgeSoC (in planning)
- **IcySoC chips approx. computing platforms** (ALP 180nm)
  - Diego
  - Manny
  - Sid

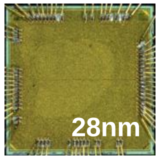


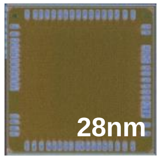


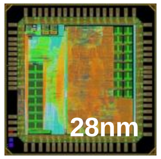


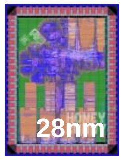


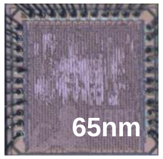


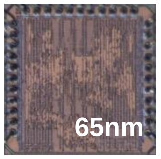


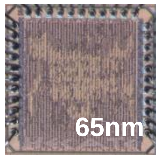


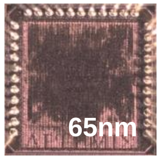










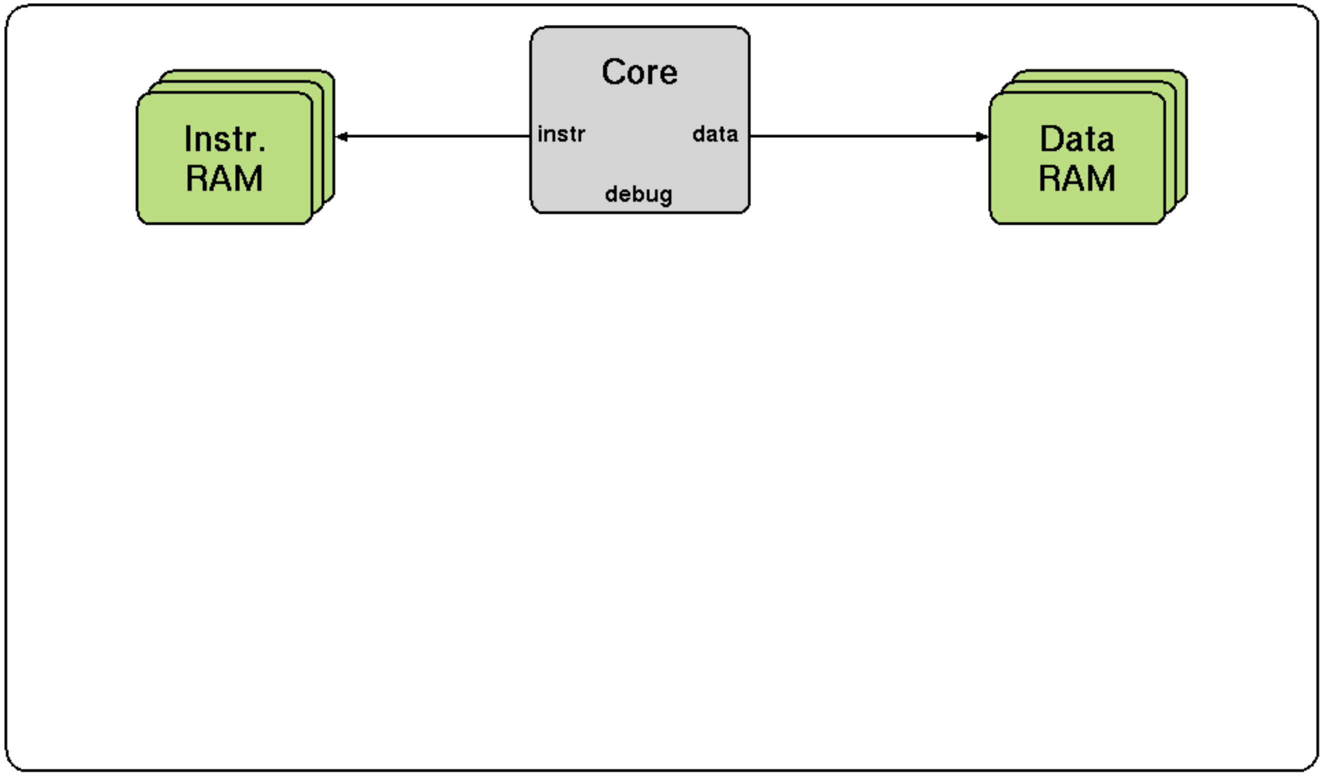


PULPino: A single-core RISC-V SoC
Motivation

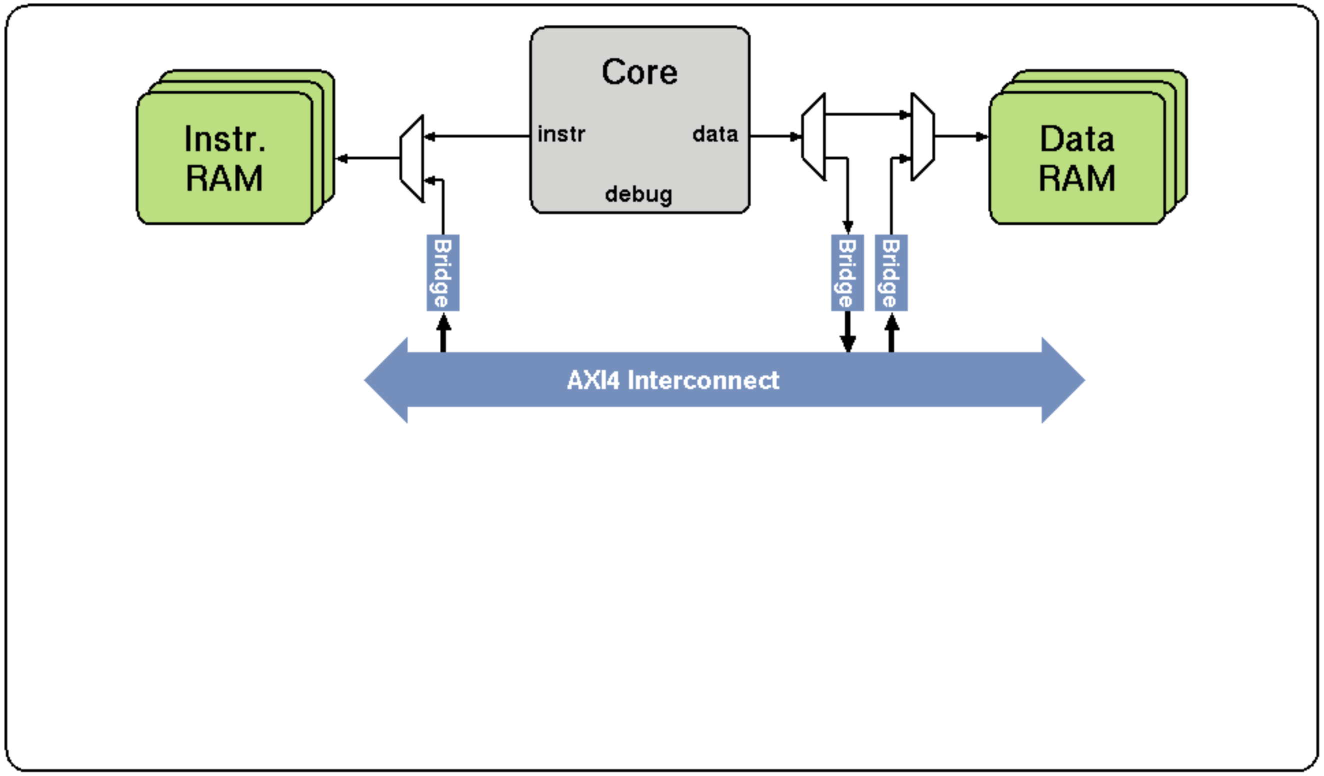
- We want to publish our work as open-source
- But PULP is huge
  - Energy efficient many-core SoC
  - Large number of IPs
  - Software frameworks (OpenMP, OpenVX, …)
  - Custom toolset (Virtual Platform, profiling tools, toolchain, …)
- PULPino as a first step
  - Focused on simplicity
  - Minimal system
  - Single-core

PULPino: A single-core RISC-V SoC Overview

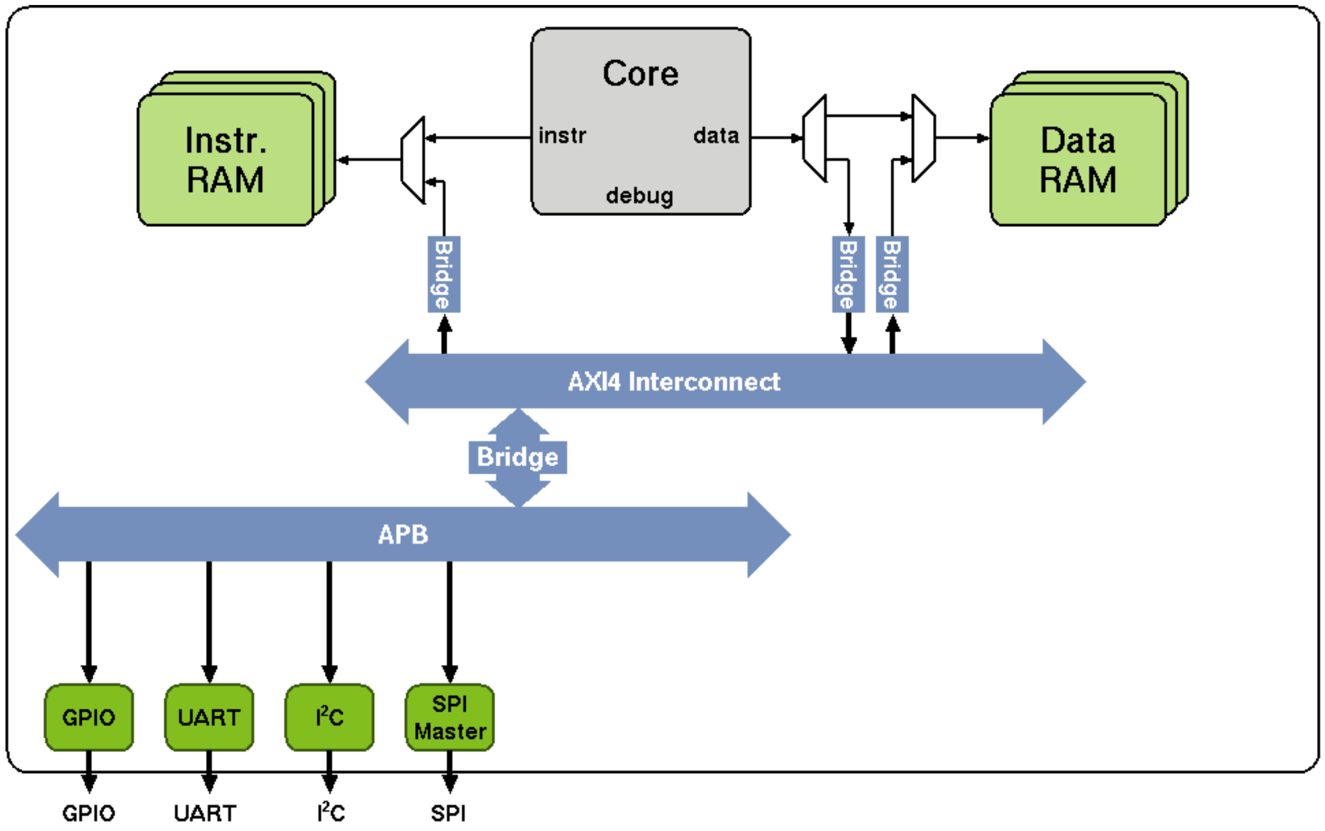
- Microcontroller-style platform
- Focused on core
- No caches, no memory hierarchy, no DMA
- Re-uses IPs from the PULP project
  - Same peripherals, same cores
- Available for FPGA
- First ASIC tapeout end of January
  - Student project using UMC 65nm



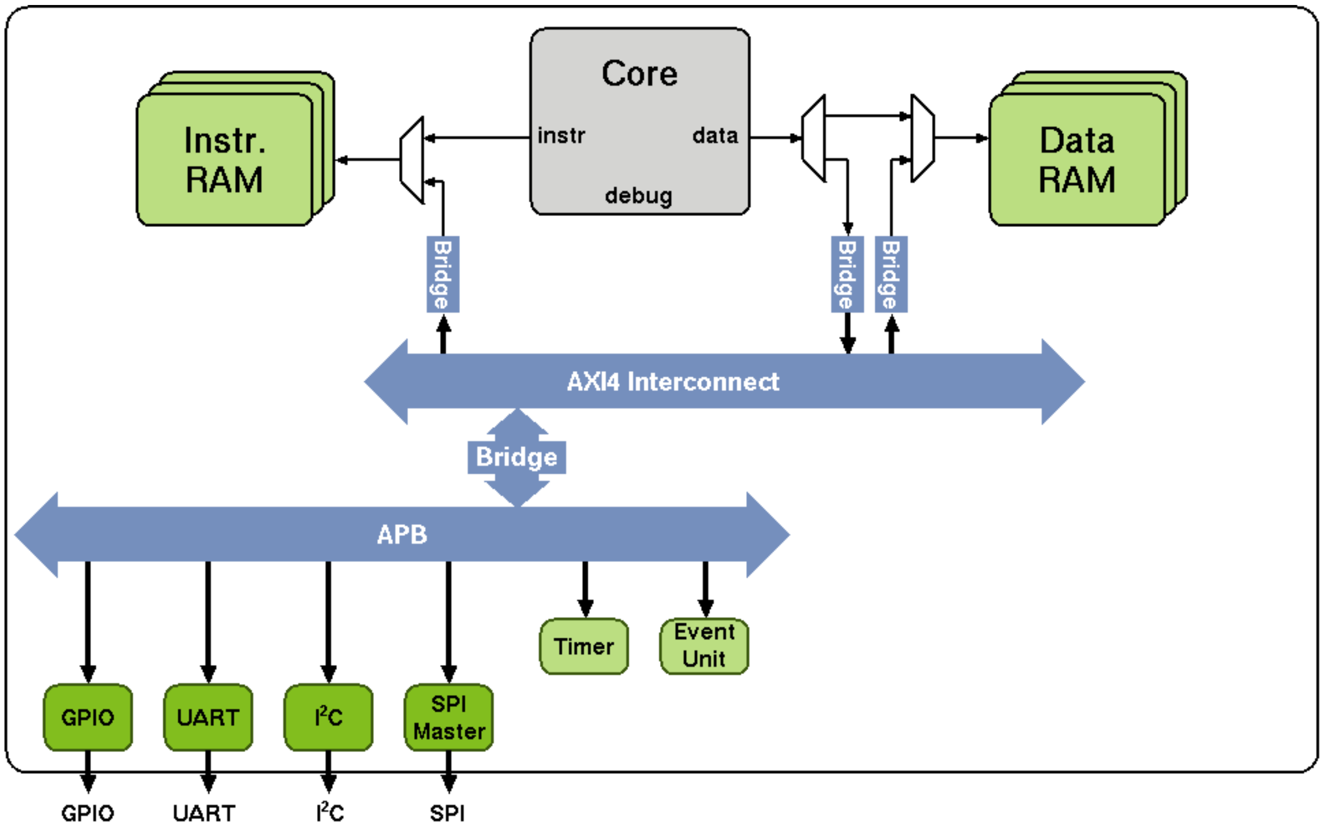
- Core directly connected to instruction and data RAM
  - Single-cycle access, no waiting for memories
- Single-port RAMs
  - Multi-banked RAMs for ASIC
  - Block RAMs for FPGA



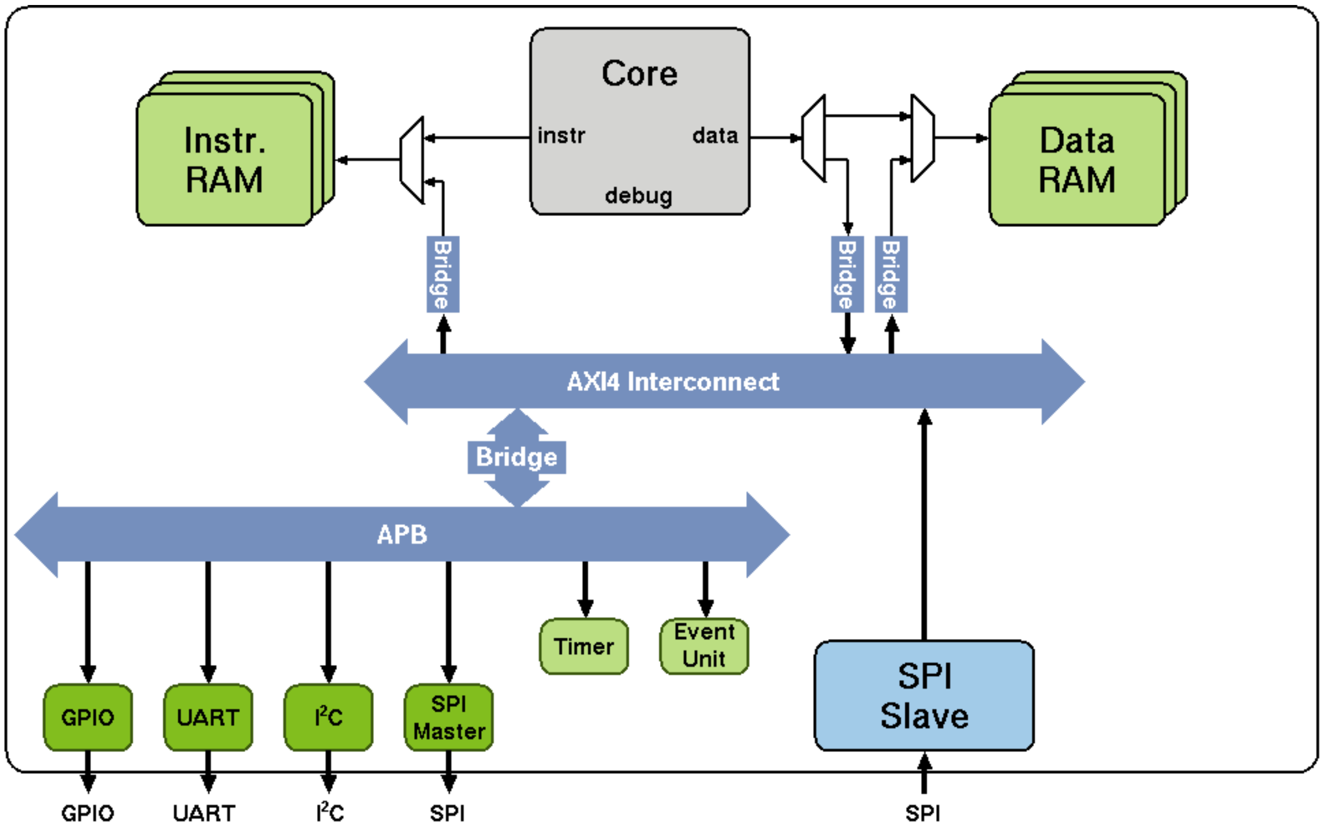
- Central AXI interconnect
  - Core connects via bridge
  - RAM multiplexed between AXI port and core port



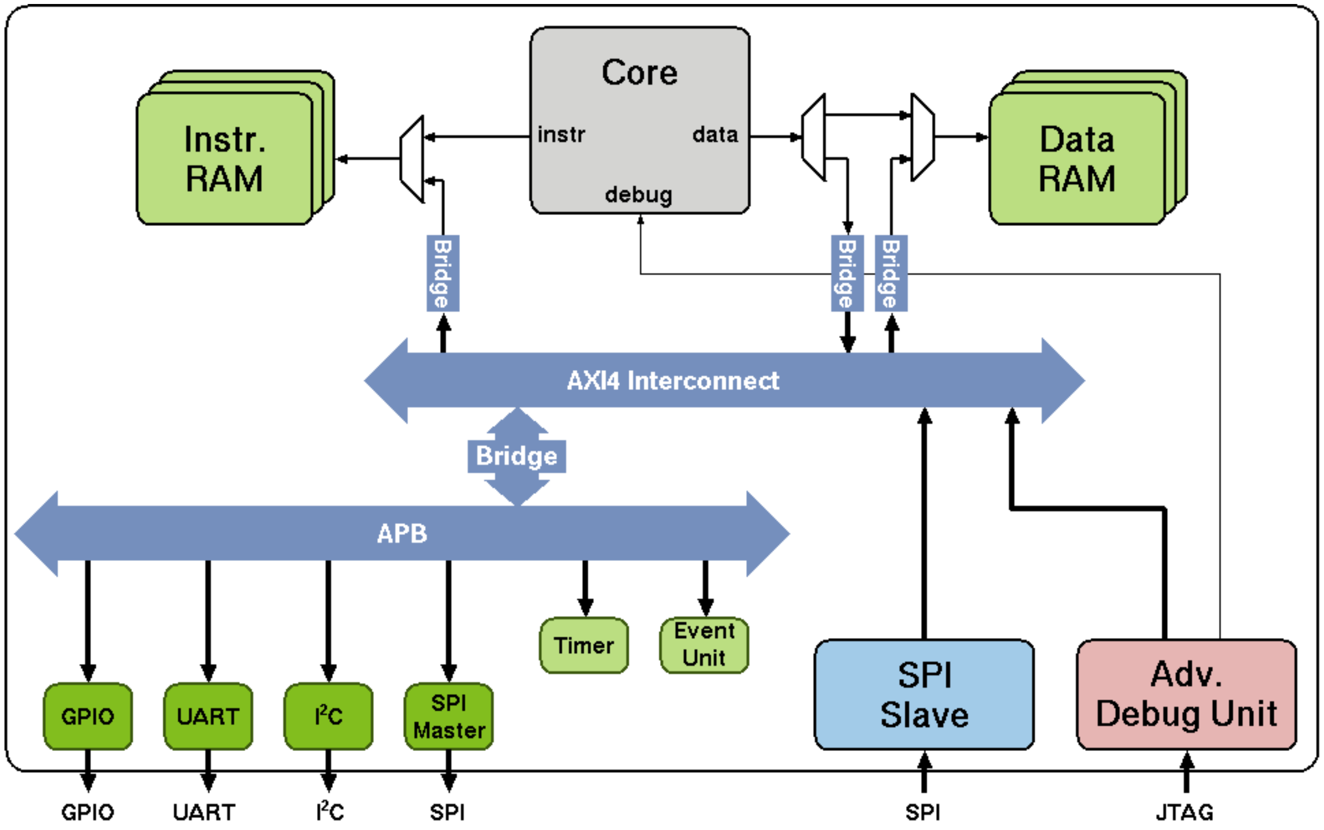
- APB for peripherals
  - Fine grained clock gating for peripherals
  - Easy to add new peripherals



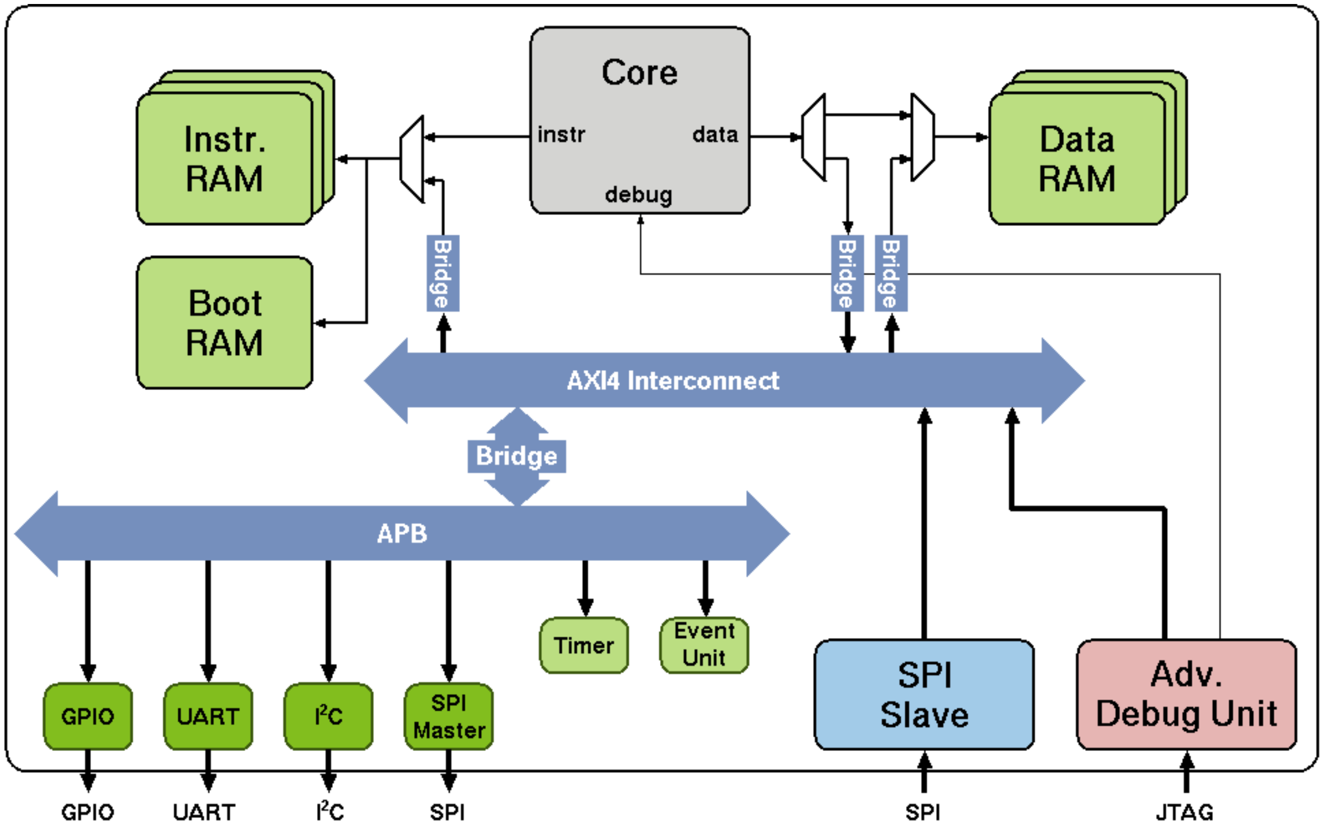
- Internal event unit
  - Clock gates core when inactive (sleep)
  - Waits for events and wakes up core



- SPI slave acts as master on AXI
  - Provides access to whole memory map of the SoC from external
- SPI slave supports standard SPI and QPI

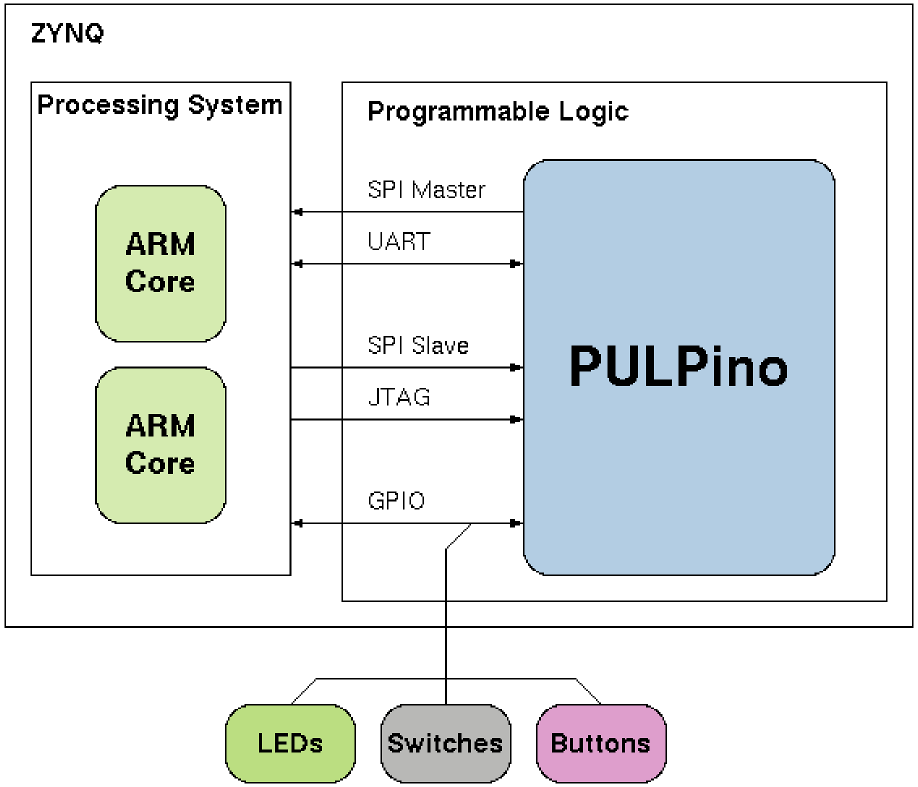


- Debug support via adv. debug unit
  - Provides JTAG port
  - Access to whole memory map via JTAG



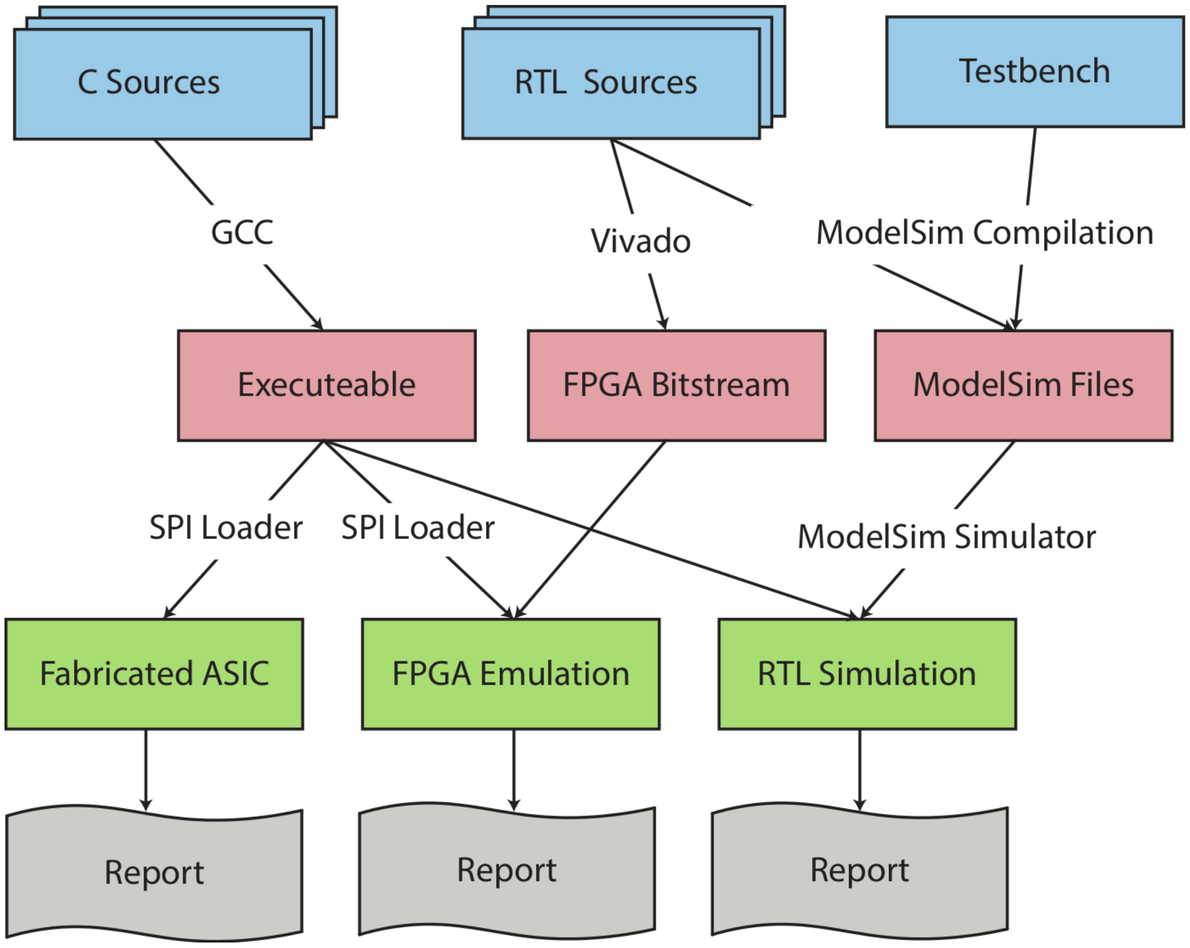
- Boot ROM
  - Bootloader loads program from SPI flash

PULPino on FPGA ZedBoard



- Program PULPino via SPI or JTAG
- Interface fully compatible with final ASIC

PULPino Build & Run Flow
Managed via CMake



RISC-V & OpenRISC

- Evaluating RISC-V in comparisons to OpenRISC
- Our interest in RISC-V
  - More modern than OpenRISC
  - No set flag instructions
  - No delay slot
  - Compressed instructions
  - Easily extendable

Extending RISC-V

- Our goals
  - Energy-efficient signal processing
  - Tune the core micro-architecture and ISA towards this
  - Extensions should have low overhead in area & power
- Adding non-standard extensions for
  - Hardware loops
  - Post-incrementing load and store instructions
  - Multiply-Accumulate
  - ALU extensions (min, max, absolute value, …)

RI5CY Extension: Hardware Loops

Simple vector addition

```
for (i = 0; i < 100; i++) {
  d[i] = a[i] + 1;
}
```

Baseline

```
        mv   x4, 0
        mv   x5, 100
Lstart: lw   x2, 0(x10)
        addi x10, x10, 4
        addi x2, x2, 1
        sw   x2, 0(x11)
        addi x11, x11, 4
        addi x4, x4, 1
        bne  x4, x5, Lstart
```

7 instr. + branch cost

with hardware loops

```
      lp.setupi 100, Lend
      lw   x2, 0(x10)
      addi x10, x10, 4
      addi x2, x2, 1
      sw   x2, 0(x11)
Lend: addi x11, x11, 4
```

5 instr.

RI5CY Extension: Post-incrementing LD/ST

Simple vector addition

```
for (i = 0; i < 100; i++) {
  d[i] = a[i] + 1;
}
```

with hardware loops

```
      lp.setupi 100, Lend
      lw   x2, 0(x10)      ┐
      addi x10, x10, 4     │
      addi x2, x2, 1       ├ 5 instr.
      sw   x2, 0(x11)      │
Lend: addi x11, x11, 4     ┘
```

with hardware loops & post-incr.

```
      lp.setupi 100, Lend
      lw   x2, 4(x10!)     ┐
      addi x2, x3, 1       ├ 3 instr.
Lend: sw   x2, 4(x11!)     ┘
```

- 4 instructions + branch cost saved

ETH zürich | ALMA MATER STUDIORUM UNIVERSITA DI BOLOGNA

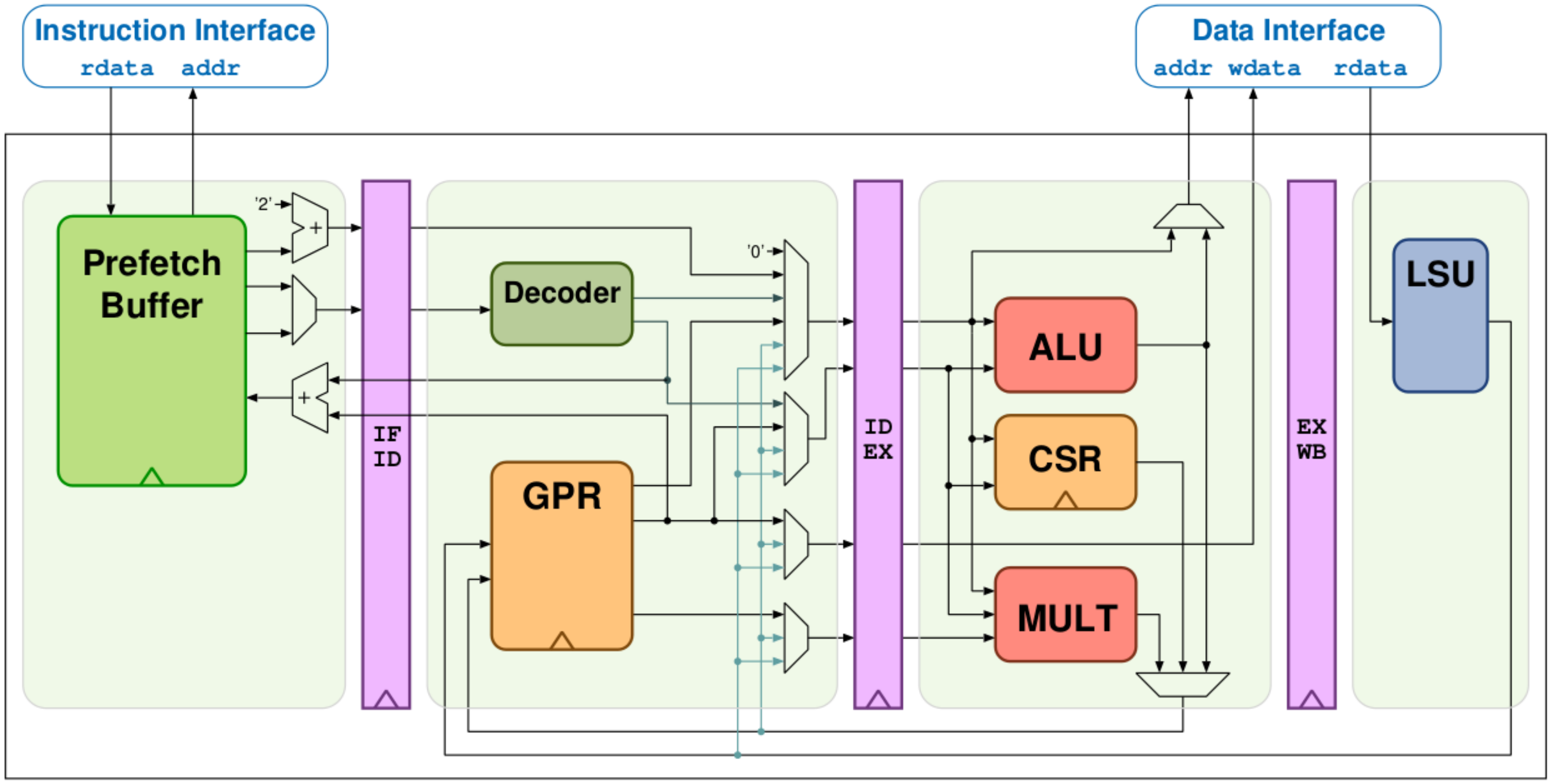
RI5CY Core
ISA Support

- Full support for RV32I
- Partial support for RV32M
  - Basically just the mul instruction
  - Single-cycle multiplication
- Support for compressed instructions
- Support for our own custom instruction set extensions

RI5CY Core Features

- Interrupts
  - Vectorized on 32 lines
- Events
  - Allows the core to sleep and wait for an event
- Exceptions
- Debug
  - Software breakpoints
  - Access to all registers
- Performance Counters

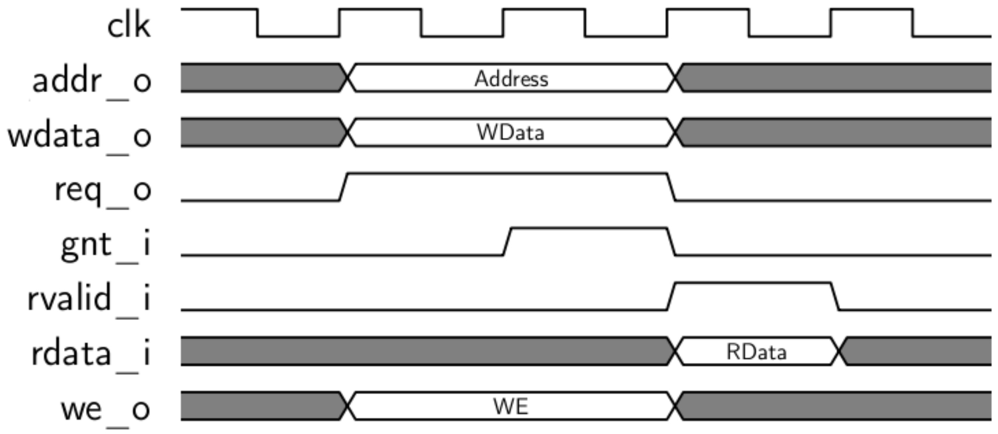
RI5CY Core



- Simple 32-bit 4-stage in-order RISC-V core

RI5CY Interfaces

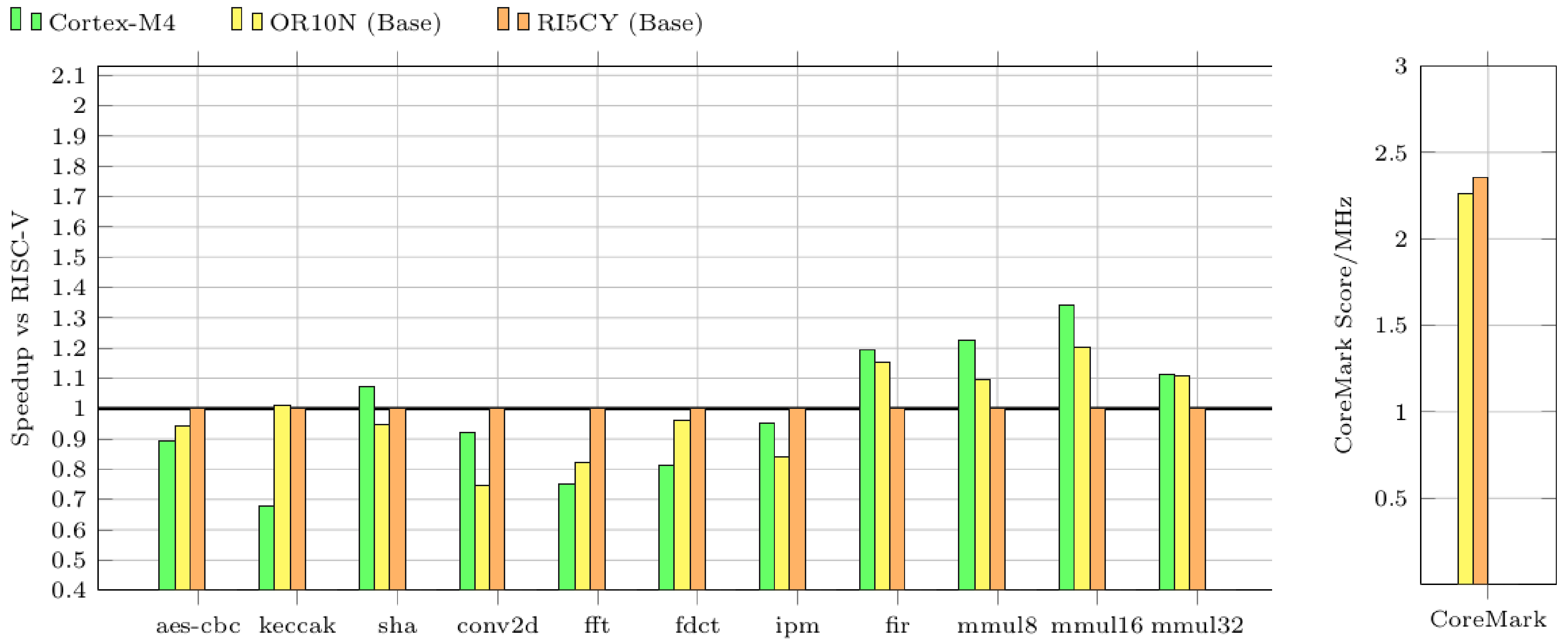
- Data Interface
  - Supports variable-latency systems
  - Request-Grant protocol



- Instruction Interface
  - Identical to Data Interface, but read-only
- Debug Interface
  - Aligned with Adv. Debug Bridge

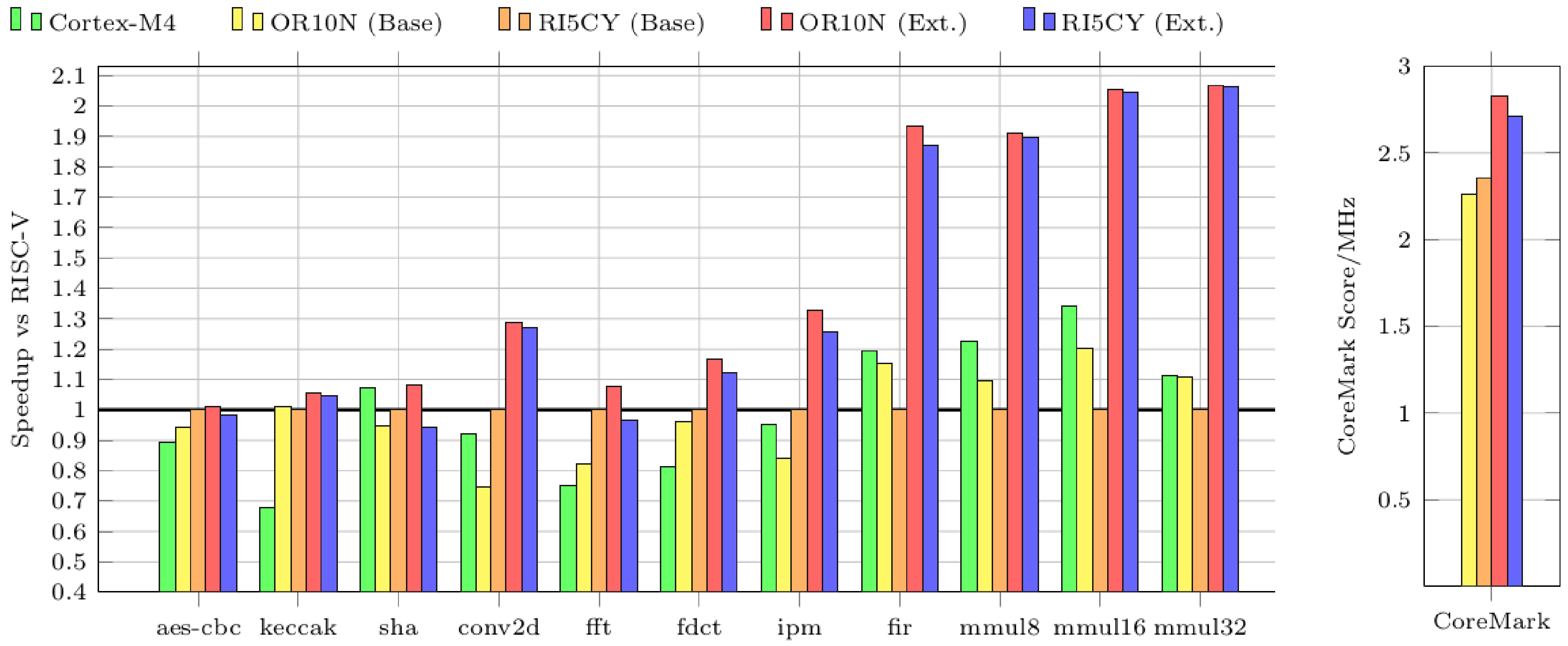
ETH zürich | ALMA MATER STUDIORUM UNIVERSITA DI BOLOGNA

RI5CY Performance



- Compiler used
  - ARM: Official GCC ARM Embedded Toolchain (GCC 4.9)
  - OR1k: Custom GCC Toolchain (GCC 5.2)
  - RISC-V: Custom GCC Toolchain (GCC 5.2)

RI5CY Performance



- Compiler used
  - ARM: Official GCC ARM Embedded Toolchain (GCC 4.9)
  - OR1k: Custom GCC Toolchain (GCC 5.2)
  - RISC-V: Custom GCC Toolchain (GCC 5.2)

RI5CY in Comparison

	RI5CY	ARM Cortex M4[2]	
Technology	65 nm	90nm	65 nm
Conditions	25°C, 1.2V	25°C, 1.2V	25°C, 1.2V
Dynamic Power $[\mu W/MHz]$	17.5	32.82	23.2[1]
Area $[mm^2]$	0.050	0.119	0.062[1]

Critical paths in RI5CY

	Delay	Delay [#ND2 Equiv.]
Internal	1.15 ns	29
Memory Request	0.6 ns + mem. delay	15 + mem. delay
Memory Response	mem. delay + 0.4 ns	mem. delay + 10

[1]scaled from 90nm to 65nm

[2]ARM Cortex M4, http://www.arm.com/products/processors/cortex-m/cortex-m4-processor.php

Area Breakdown

PULPino

Component	Area [GE]	
RI5CY Core	34’500	6.9%
Peripherals	50’000	10%
Instruction RAM (32kB)	190’000	38%
Data RAM (32kB)	190’000	38%
AXI Interconnect	10’000	2%
Adv. Dbg Unit	6’000	1.2%
Total	**500’000**	**100%**

RI5CY

Component	Area [GE]	
Prefetch Buffer	3’500	10.1%
Decoder	420	1.2%
Compressed Decoder	400	1.1%
Register File	10’600	30.7%
Multiplier	4’300	12.5%
ALU	2’700	7.8%
LSU	1’500	4.3%
CSR	2’160	6.3%
Total	**34’500**	**100%**

Open Source Release

- What we release
  - Complete PULPino RTL source
    - Including RI5CY core
    - Including all IPs
  - FPGA build flow
  - Simulation/build infrastructure for software
  - FreeRTOS port
- Liberal license: SolderPad
- When?
  - Awaiting final approval

Summary

- PULP: Energy-efficient many-core SoC
- PULPino: Single-core SoC
  - A complete system that can be easily employed in various projects
  - Easily extendable
  - Ready to use system for educational purposes
- RI5CY: Small and optimized RISC-V core

- Check our website: pulp.ethz.ch

Outlook

- First tape out of PULPino end of this month
- IP-XACT port of PULPino
- RI5CY features
  - Floating-point support
    - Already available for our OpenRISC core
  - Branch prediction
  - Evaluate further non-standard ISA extensions

Questions?

http://pulp.ethz.ch