Working with RISC-V

Part 5 of 5: PULP based chips

Davide Rossi <davide.rossi@unibo.it>
Luca Benini <luca.benini@unibo.it>
Summary

- Part 1 – Introduction to RISC-V ISA
- Part 2 – Advanced RISC-V Architectures
- Part 3 – PULP concepts
- Part 4 – PULP Extensions and Accelerators
- Part 5 – PULP based chips
  - From concept to reality
  - Single core microcontrollers PULPino to PULPissimo
  - Many core systems OpenPULP
  - Advanced systems with accelerators
  - Lessons learned, the good, the bad and the ugly.
We will discuss chips we have made with PULP

- Why make chips at all?
  - MPW: Only limited samples
  - Use cases

- Single core PULP chips
  - PULPino (Imperio)
  - PULPissimo (Arnold)

- Many core PULP chips
  - Cluster only (Honey Bunny, Dustin)
  - PULPopen (Mr. Wolf, Vega)

- Advanced PULP chips
  - Kosmodrom: 2x 64b Ariane cores + ML accelerators
  - Making use of technology: Body biasing

- Lessons learned
  - There are many pitfalls
  - We had great success, but..
  - Sometimes you have embarrassing failures. Part of the process
Multi Project Wafer, chips for prototyping

- **Cost sharing method for ICs**
  - Multiple ICs are manufactured together. They share the mask costs
    - 1.5M cost / 10 projects = 150k per project
    - But you only get 1 / 10 of the area
  - Dedicated MPW services available
    - Europractice-IC for SMEs and academia

- **You only get few chips**
  - Usually 50 to 200
  - Per chip costs very high (few kUSD)

- **All our chips through MPWs**
Our ASICs have different use cases

- Chips characterized on an IC tester (*Poseidon 22nm*)
- Research demonstrators (*Nano drone with Mr. Wolf/GAP8*)
- Industrial uses of our cores/peripherals (*open-isa.org Vega*)
Most of what we show is openly available

- All our development is on GitHub
  - HDL source code, testbenches, software development kit, virtual platform
    https://github.com/pulp-platform

- PULP is released under the permissive Solderpad license
  - Allows anyone to use, change, and make products without restrictions.
PULP has released a large number of IPs

RISC-V Cores
- RI5CY 32b
- Ibex 32b
- Snitch 32b
- Ariane + Ara 64b

Platforms
- Single Core: PULPino, PULPissimo
- Multi-core: Fulmine, Mr. Wolf
- Multi-cluster: Hero, Open Piton

Peripherals
- JTAG
- UART
- SPI
- I2S
- DMA
- GPIO
- I2S
- UART
- SPI
- JTAG

Interconnect
- Logarithmic interconnect
- APB – Peripheral Bus
- AXI4 – Interconnect

Accelerators
- Neurostream (ML)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
- HWCE (convolution)
PULPino: Our first open source release

- Simple design
  - Meant as a quick release

- Separate data and inst. mem
  - Makes it easy in HW
  - Not meant as a Harvard arch.

- Can use all our 32bit cores
  - RI5CY, Zero/Micro-Riscy (Ibex)

- Peripherals from other projects
  - Any AXI and APB peripherals could be used
Imperio – 65nm RISC-V core

- **Chip implemented in 65nm**
  - Using RI5CY (RV32IMC) core
  - 64 kBytes of memory
  - Basic peripherals (GPIO, SPI, I2C)
  - Working debug interface

- **Functional up to 500 MHz**
  - Main challenge was to find fast memory cuts to work at that speed.
  - Memory made of multiple smaller cuts to maximize the operating speed.
Working chip on an Arduino compatible board
Arnold (2018) – Fastest collaboration

- **GF22nm**
  - RISC-V microcontroller with eFPGA
  - Based around PULPissimo

- **Collaboration with Quicklogic**
  - Met at GTC 2017 by coincidence
  - In one year chip was taped out
  - Only possible because of open source nature

- **Quicklogic is going open source**
  - They announced June 2020 the Quicklogic Open Reconfigurable Computing
  
  https://www.quicklogic.com/QORC/

---

PULPissimo: very good platform for extensions

- eFPGA added as accel.
  - Easy plug and play
  - Configuration over APB
  - Additional ALU and memory
  - Uses the same memory

- Multiple operation modes
  - Configurable peripheral
  - Accelerator for core
  - Accelerator for independent I/O
Experimental platform with many configurations

- I/O subsystem accel
  - 6.0mW, 2.5x
- Custom I/O interface
  - BNN interface 12.5mW 2.2x
- CPU accelerator
  - CRC 7.5mW 42x
- Many more ideas
  - Dynamic reconfiguration
Arnold test board with D. Schiavone
Full Multi-Cluster SoCs

SoC

L2 Mem

interconnect

DMA

Mem

Mem

Mem

Mem

Tightly Coupled Data Memory

interconnect

Event Unit

RISC-V core

RISC-V core

RISC-V core

RISC-V core

I$ I$ I$ I$

I/O

CLUSTER

ACACES 2021 - Sept 2021
Mr. Wolf (TSMC 40): 8+1 core IoT Processor

- One cluster with
  - 8 RISC-V cores
  - 2x shared FPU units
  - 64 kByte of TCDM

- One controller with
  - 512 kByte L2 RAM
  - Peripherals

- On chip voltage regulators
  - By Dolphin Integration

On-chip regulators allow different power modes

<table>
<thead>
<tr>
<th>Power Mode</th>
<th>VDD</th>
<th>Frequency Range</th>
<th>Power</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deep Sleep</td>
<td>0.8 V</td>
<td>n.A.</td>
<td>72 µW</td>
</tr>
</tbody>
</table>
It is possible to keep memory state intact

<table>
<thead>
<tr>
<th>Power Mode</th>
<th>VDD</th>
<th>Frequency Range</th>
<th>Power</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deep Sleep</td>
<td>0.8 V</td>
<td>n.A.</td>
<td>72 μW</td>
</tr>
<tr>
<td>State Retentive Deep Sleep</td>
<td>0.8 V</td>
<td>n.A.</td>
<td>77 – 108 μW</td>
</tr>
</tbody>
</table>
SoC is awake but is clock gated

<table>
<thead>
<tr>
<th>Power Mode</th>
<th>VDD</th>
<th>Frequency Range</th>
<th>Power</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deep Sleep</td>
<td>0.8 V</td>
<td>n.A.</td>
<td>72 μW</td>
</tr>
<tr>
<td>State Retentive Deep Sleep</td>
<td>0.8 V</td>
<td>n.A.</td>
<td>77 – 108 μW</td>
</tr>
<tr>
<td>SoC Idle</td>
<td>0.8 – 1.1 V</td>
<td>SoC clock gated</td>
<td>0.55 – 1.96 mW</td>
</tr>
</tbody>
</table>

Controller

<table>
<thead>
<tr>
<th>Power Control</th>
<th>R5</th>
<th>Interconnect</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td>M</td>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td>M</td>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td>M</td>
<td>M</td>
<td>M</td>
</tr>
</tbody>
</table>

Cluster

<table>
<thead>
<tr>
<th>M</th>
<th>M</th>
<th>M</th>
<th>M</th>
<th>M</th>
<th>M</th>
<th>M</th>
<th>M</th>
</tr>
</thead>
<tbody>
<tr>
<td>R5</td>
<td>R5</td>
<td>R5</td>
<td>R5</td>
<td>R5</td>
<td>R5</td>
<td>R5</td>
<td>R5</td>
</tr>
</tbody>
</table>
Only SoC with a single RISC-V core running

<table>
<thead>
<tr>
<th>Power Mode</th>
<th>VDD</th>
<th>Frequency Range</th>
<th>Power</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deep Sleep</td>
<td>0.8 V</td>
<td>n.A.</td>
<td>72 μW</td>
</tr>
<tr>
<td>State Retentive Deep Sleep</td>
<td>0.8 V</td>
<td>n.A.</td>
<td>77 – 108 μW</td>
</tr>
<tr>
<td>SoC Idle</td>
<td>0.8 – 1.1V</td>
<td>SoC clock gated</td>
<td>0.55 – 1.96 mW</td>
</tr>
<tr>
<td>SoC active</td>
<td>0.8 – 1.1V</td>
<td>32 kHz – 450 MHz</td>
<td>0.97 – 38 mW</td>
</tr>
</tbody>
</table>
## Cluster is active, but clock gated

<table>
<thead>
<tr>
<th>Power Mode</th>
<th>VDD</th>
<th>Frequency Range</th>
<th>Power</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deep Sleep</td>
<td>0.8 V</td>
<td>n.A.</td>
<td>72 µW</td>
</tr>
<tr>
<td>State Retentive Deep Sleep</td>
<td>0.8 V</td>
<td>n.A.</td>
<td>77 – 108 µW</td>
</tr>
<tr>
<td>SoC Idle</td>
<td>0.8 – 1.1 V</td>
<td>SoC clock gated</td>
<td>0.55 – 1.96 mW</td>
</tr>
<tr>
<td>SoC active</td>
<td>0.8 – 1.1 V</td>
<td>32 kHz – 450 MHz</td>
<td>0.97 – 38 mW</td>
</tr>
<tr>
<td>Cluster Idle</td>
<td>0.8 – 1.1 V</td>
<td>Cluster clock gated</td>
<td>1.2 – 4.6 mW</td>
</tr>
</tbody>
</table>

**Controller**

- Power Control
  - R5
- Interconnect

**Cluster**

- M M M M M
- R5 R5 R5 R5 R5 R5 R5 R5
Cluster with 8 RISC-V cores is active

<table>
<thead>
<tr>
<th>Power Mode</th>
<th>VDD</th>
<th>Frequency Range</th>
<th>Power</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deep Sleep</td>
<td>0.8 V</td>
<td>n.A.</td>
<td>72 μW</td>
</tr>
<tr>
<td>State Retentive Deep Sleep</td>
<td>0.8 V</td>
<td>n.A.</td>
<td>77 – 108 μW</td>
</tr>
<tr>
<td>SoC Idle</td>
<td>0.8 – 1.1 V</td>
<td>SoC clock gated</td>
<td>0.55 – 1.96 mW</td>
</tr>
<tr>
<td>SoC active</td>
<td>0.8 – 1.1 V</td>
<td>32 kHz – 450 MHz</td>
<td>0.97 – 38 mW</td>
</tr>
<tr>
<td>Cluster Idle</td>
<td>0.8 – 1.1 V</td>
<td>Cluster clock gated</td>
<td>1.2 – 4.6 mW</td>
</tr>
<tr>
<td>Cluster Active</td>
<td>0.8 – 1.1 V</td>
<td>32 kHz – 350 MHz</td>
<td>1.6 – 153 mW</td>
</tr>
</tbody>
</table>

Controller

<table>
<thead>
<tr>
<th>Power Control</th>
<th>M</th>
<th>M</th>
<th>M</th>
<th>M</th>
<th>M</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interconnect</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R5</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
</tr>
</tbody>
</table>

Cluster

<table>
<thead>
<tr>
<th>M</th>
<th>M</th>
<th>M</th>
<th>M</th>
<th>M</th>
<th>M</th>
<th>M</th>
<th>M</th>
<th>M</th>
</tr>
</thead>
<tbody>
<tr>
<td>R5</td>
<td>R5</td>
<td>R5</td>
<td>R5</td>
<td>R5</td>
<td>R5</td>
<td>R5</td>
<td>R5</td>
<td>R5</td>
</tr>
</tbody>
</table>
Our OpenPULP release is Mr. Wolf

- With Mr. Wolf, most of what we have is open sourced
  - This is a complex IoT processor, not like the much simpler PULPino
  - 8 + 1 cores, FPUs, shared accelerators, multiple power down modes.

- Still many parts can still not be open source
  - Technology specific information, P&R scripts
  - Memory macros, selected cuts, their performance
  - I/O cells
  - FLL, analog macros, I/O cells, memory cuts (affects performance), P&R scripts

- Interesting industry collaboration
  - Greenwaves, BitCraze, Dolphin
Mr. Wolf has been used in multiple systems

- Designed as an application processor
  - We still build boards with it
  - Despite only 200 manufactured

- Widespread industrial use:
  - Dolphin IP was validated on this chip
  - Greenwaves GAP8 is based on the open source release OpenPULP
  - BitCraze AI Deck is related
Complete Application: DroNET on NanoDrone

Pluggable PCB: PULP-Shield
- ~5g, 30×28mm
- GAP8 SoC
- 8 MB HDRAM
- 16 MB HFlash
- QVGA ULP HiMax camera
- Crazyflie 2.0 nano-drone (27g)

Only onboard computation for autonomous flight + obstacle avoidance
no human operator, no ad-hoc external signals, and no remote base-station!
VEGA: Extreme Edge IoT Processor

- Fully programmable RISC-V based cluster targeting highly dynamic Near-Sensor Analytic Applications (NSAA)

- 1.7 $\mu$W cognitive unit for autonomous wake-up from retentive sleep mode
VEGA: Extreme Edge IoT Processor

- Fully programmable RISC-V based cluster targeting highly dynamic Near-Sensor Analytic Applications (NSAA)
- 1.7 µW cognitive unit for autonomous wake-up from retentive sleep mode
- Fully integrated execution of real-life DNN from 4 MB of non-volatile MRAM (first time for an IoT end-node)
SoC Overview

- 32-bit RISC-V core (Fabric Controller)
- 1.6 MB L2 SRAM
- 4 MB non-volatile MRAM
- Standard set of peripherals (SPI, I2C, UART, CSI2...)
- Off-chip memory (*HyperRAM™ DRAM / Flash)
- Autonomous I/O DMA
- Cognitive smart wake-up
- 3 Frequency-Locked Loops (FLL)
- 2 Voltage regulators (HP/LP) + 1 LDO (COTS) + PMU

*https://www.cypress.com/products/hyperram-memory
Software-Programmable Accelerator

- 9 RISC-V DSP cores
- 128KB 16-Banks TCDM (scratchpad, no cache)
- Single-cycle latency, word-level interleaved Interconnect
- DMA for explicit memory mgmt.
- I$: 9x 0.5kB L1 I$ + 4KB L1.5 I$
- Hardware Synchronizer (HW SYNC)
- Shared SIMD Floating-Point Unit (FPU)
- DNN Accelerator
Digital computing platforms for near-sensor processing at the extreme edge of the IoT

Full DNN Performance (MobileNetV2)

@ $V_{dd\_SOC}=0.8\,\text{V}, f_{\text{SOC}}=250\,\text{MHz}, f_{\text{CL}}=250\,\text{MHz}$

- MRAM→L2 w/ I/O DMA
- L2→L1 w/ Cluster DMA
- Computation

Conv 1x1
DwConv 3x3
Conv 1x1
Add
Conv2d 3x3
Bottleneck1
Bottleneck2
Bottleneck3
Bottleneck4
Bottleneck5
Bottleneck6
Bottleneck7
Conv2d 1x1
Conv2d 1x1

‘royale with cheese’

weights on MRAM
weights on HyperRAM

Time per Inference [ms]

Time [us]

x2 times
x3 times
x4 times
x3 times
x3 times
final Conv2d layer is MRAM-bound

@ Vdd_SOC=0.8V, f_SOC=250 MHz, f_CL=250 MHz

weights on MRAM
weights on HyperRAM

0.0 10.0 20.0 30.0 40.0 50.0 60.0 70.0 80.0

84.7 ms
87.7 ms

ACACES 2021 - Sept 2021

ETH Zürich
Full DNN Energy (MobileNetV2)

Bandwidth [MB/s]

<table>
<thead>
<tr>
<th>Bandwidth [MB/s]</th>
<th>Energy per byte [pJ/B]</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>8000</td>
<td>8000</td>
</tr>
</tbody>
</table>

- **HyperRAM (ext)→L2 w/ I/O DMA**
- **MRAM→L2 w/ I/O DMA**
- **L2→L1 w/ Cluster DMA**
- **L1 access**

Weights on MRAM: 1.19 mJ
Weights on HyperRAM: 4.16 mJ

3.5x less energy

End-to-end on-chip computation

Digital computing platforms for near-sensor processing at the extreme edge of the IoT
## World-Record Efficiency Among IoT Processors

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Embedded NVM</strong></td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td><strong>Wake-up Sources</strong></td>
<td>WiC</td>
<td>GPIO, RTC</td>
<td>WuR, RTC, Int, GPIO, RTC, Cognitive</td>
</tr>
<tr>
<td><strong>Best Int Perf. @ Perf.</strong></td>
<td>31 MOPS (32b)</td>
<td>12.1 GOPS 32 MOPS</td>
<td>1.5 GOPS @ 230 MOPS</td>
</tr>
<tr>
<td></td>
<td>97 MOPS/mW (32b)</td>
<td>190 GOPS/W @ 3.8 GOPS</td>
<td>1.4 GOPS/W @ 7.6 GOPS</td>
</tr>
<tr>
<td><strong>Best FP Perf. @Perf</strong></td>
<td>-</td>
<td>1 GFLOPS @ 350 MFLOPS</td>
<td>2 GFLOPS @ 1 GFLOPS</td>
</tr>
<tr>
<td><strong>Best ML Perf. @Perf</strong></td>
<td>-</td>
<td>36 GOPS @ 1.3 TOPS/W</td>
<td>32.2 GOPS @ 1.56 GOPS</td>
</tr>
</tbody>
</table>

3.2x Efficiency @ 2x Performance
4.3x Efficiency @ 2.8x Performance
Similar Efficiency @ 5.5x Performance

---

D. Rossi *et al.*, "4.4 A 1.3TOPS/W @ 32GOPS Fully Integrated 10-Core SoC for IoT End-Nodes with 1.7μW Cognitive Wake-Up From MRAM-Based State-Retentive Sleep Mode,” *2021 IEEE International Solid-State Circuits Conference (ISSCC)*, 2021, pp. 60-62
DUSTIN: Mixed-Precision Cluster

- 16 RI5CY (*) cores augmented with 2b-to-32b SIMD instructions;
- Software Configurable Vector Lockstep Execution Mode (VLEM);
- Single-cycle latency TCDM interco.
- 128 kB of Shared Tightly-Coupled L1 Data Memory;
- Hierarchical Instruction Cache;
- High performance DMA (L2 <-> L1);
- Event Unit supporting efficient synchronization among the cores;
Vector Lockstep Exec. Mode (VLEM)

Cluster in MIMD mode

Cluster in VLEM

TCDM INTERCONNECT

TCDM INTERCO. – LKS UNIT – BROADCAST

CORE 0
WB
EX
ID
IF
I$-0
I$-1
I$-14
I$-15

CORE 1
WB
EX
ID
IF
I$-0
I$-1
I$-14
I$-15

CORE 14
WB
EX
ID
IF
I$-0
I$-1
I$-14
I$-15

CORE 15
WB
EX
ID
IF
I$-0
I$-1
I$-14
I$-15

SHARED I$

LKS EN
CLK
VLEM: Broadcast Unit

- **Overhead**: at least 16 clk cycles to unlock the execution in case of concurrent accesses;

- **Solution**:
  - eliminate the overhead in case of access to the same mem address → BROADCAST UNIT.
  - Misalign static data and stacks to avoid accesses to the same mem bank;

Cluster in VLEM
Energy Efficiency on MatMul kernels

8x4 MatMul Power in MIMD: **29.2 mW**

- CORE_0: 4%
- CORES_1_15_ID_EX: 43.3%
- CORES_1_15_IF: 10%
- I$: 20%
- TCDM: 22%

Power in VLEM: **16.1 mW** (~45% reduction wrt MIMD)

- CORE_0: 6%
- CORES_1_15_ID_EX: 74%
- CORES_1_15_IF: 0%
- I$: 7.3%
- TCDM: 6.3%

#Cycles (normalized)

- MIMD: 1.0
- VLEM: 1.7
- VLEM + BRD + MIS. DATA: 1.02

Energy Efficiency (0.8V@60 MHz)

- MIMD: Up to 1.5x improvement
## Comparison with the SoA

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Die Area</td>
<td>0.68 mm&lt;sup&gt;2&lt;/sup&gt;</td>
<td>4.5 mm&lt;sup&gt;2&lt;/sup&gt;</td>
<td>10 mm&lt;sup&gt;2&lt;/sup&gt;</td>
<td>12 mm&lt;sup&gt;2&lt;/sup&gt;</td>
<td>16 mm&lt;sup&gt;2&lt;/sup&gt;</td>
</tr>
<tr>
<td>Applications</td>
<td>IoT GP</td>
<td>IoT GP + DNN</td>
<td>IoT GP + DNN</td>
<td>IoT GP + DNN + A+DNN</td>
<td>IoT GP + DNN + QNNs</td>
</tr>
<tr>
<td>CPU/ISA</td>
<td>CM0DS Thumb-2 subset</td>
<td>1x RISCY RVC32IMFXpulp</td>
<td>9 x RISCY RVC32IMFXpulp</td>
<td>12 x RISCY RVC32IMFXpulp</td>
<td>16 x RISC PIC CORES (RISC-V)</td>
</tr>
<tr>
<td>Int Precision (bits)</td>
<td>32</td>
<td>8, 16, 32</td>
<td>8, 16, 32</td>
<td>8, 16, 32</td>
<td>2, 4, 8, 16, 32 (plus Mixed-Precision)</td>
</tr>
<tr>
<td>Supply Voltage</td>
<td>0.4 - 0.8 V</td>
<td>0.45 - 0.9 V</td>
<td>0.8 - 1.2 V</td>
<td>0.8 - 1.2 V</td>
<td>0.8 - 1.2 V</td>
</tr>
<tr>
<td>Max Frequency</td>
<td>80 MHz</td>
<td>350 MHz</td>
<td>400 MHz</td>
<td>450 MHz</td>
<td>205 MHz</td>
</tr>
<tr>
<td>Power Envelope</td>
<td>320 µW</td>
<td>66 mW</td>
<td>153 mW</td>
<td>2.4 mW</td>
<td>156 mW</td>
</tr>
<tr>
<td><strong>1</strong>Best Integer Performance</td>
<td>31 MOPS (32b)</td>
<td>1.5 GOPS (8b)&lt;sup&gt;2&lt;/sup&gt;</td>
<td>12.1 GOPS (8b)</td>
<td>15.6 GOPS (8b)</td>
<td>15 GOPS (8b)</td>
</tr>
<tr>
<td><strong>1</strong>Best Integer Efficiency</td>
<td>97 MOPS/mW @ 18.6 MOPS (32b)</td>
<td>230 GOPS/W @ 110 MOPS (8b)</td>
<td>190 GOPS/W @ 3.8 GOPS (8b)</td>
<td>614 GOPS/W @ 7.6 GOPS</td>
<td>303 GOPS/W @ 4.4 GOPS (8b)</td>
</tr>
</tbody>
</table>

---

Dustin supports Mixed-Precision computation in HW
Better efficiency wrt solutions in 28nm and 40nm tech node
Comparable efficiency wrt Vega (22 nm)


---

Presented at ESSCIRC 2021
Moving to HPC: Kosmodrom

- **Globalfoundries 22FDX**
  - In 2018, most advanced node for us
  - Minimum size 3mm x 3mm
    - That fits about 100 million transistors
  - Allows body biasing

- **With great power comes...**
  - Designs in 22FDX are more involved
  - More blocks, more functionality
    - More things that can go wrong
  - Challenging design
  - Collaboration with Globalfoundries
Kosmodrom: Main components

- **2x Ariane 64b RISC-V cores**
  - AHP optimized for high speed
  - ALP optimized for low power

- **Automatic Body Bias Gen.**
  - IP by INVECAS
  - Allows body bias to be tuned

- **NTX: Neural Training Accelerator**
  - 260 Gflops/Watt efficiency

- **Common infrastructure**
  - SRAM, Debug, I/Os
Fine-Grained Shared-Memory Accelerators

Similar concept as OpenPULP, but fewer RISC-V cores and more accelerators
NTX uses 1 RISC-V core to control 8 units

- NTX runs at up to 1.25 GHz
- Compute of 20 Gflop/s
- Bandwidth of 5 GB/s
- At 9.3 pJ/flop and using only 0.51 mm²
- Scale up by replicating cluster

Kosmodrom Demonstration Board

- STM microcontroller for control
- USB connection to computer
- Analog to Digital Converter module
- Test socket for Kosmodrom chip
- Body bias voltage generation
- Supply voltage generation
- Measurement points for all supplies
Boosting performance with Body Biasing

- We set the performance target (730MHz, @0.65V, ~40mW)
- Actual chip performance is measured
- Forward VBB is applied (positive VBP and negative VBN)
- Until we reach the performance goals
- By individually applying VBB to chips we can improve yield

50% Performance gain with Body Biasing
Gaining Energy Efficiency with Body Biasing

- We set the desired operating frequency (800MHz).
- We decrease the voltage to the minimum level the chip will work (0.8V).
- At this point, we start reducing voltage further (~0.65V).
- Maximum operating frequency will also drop (~500MHz).
- We compensate for the lost performance with forward VBB (positive VBP and negative VBN).
- Until we reach the desired operating frequency.

At least 20% more efficiency with VBB.
The good the bad and the ugly

- We designed and tested 43 chips as part of PULP project
- Most worked great
- But there were also mistakes
- Here is a look at some highs and some lows
Good: Fulmine the award winning one

- **UMC65**

- **Earlier chip (2015)**
  - 4x OpenRISC cores (not yet RISC-V)
  - 192 kBytes L2 + 64 kBytes TCDM
  - 2x HW accelerators
    - HW – Crypt (together with TU-Graz)
    - HW – Convolution Engine

- **Publication from this chip**
  
Bad: Bonding issues on Poseidon

- First GF22nm chip
  - Used Europractice IC service
  - Cost 150k CHF for 50 samples

- Has three parts (trident..)
  - PULPissimo system
  - Ariane core
  - Independent ML accelerator

- 30 of 50 chips were packaged
  - We provide a bonding diagram
  - Mostly simple manual work
Bad: Bonding issues on Poseidon

- Look closer on the right side
  - There is a pad that is not bonded
- We skipped one pad
  - All connections are shifted by one
- VDD and GND are one after other
  - Bonding causes shorts between VDD and GND
  - Pretty much catastrophic.
- **Fortunately**: unpackaged dies
  - There were 20 unpackaged dies
  - We could bond those correctly
Downright Ugly, reset problem of Urania

- 2 PULP clusters, each with
  - 4x RV32 RI5CY cores
  - 4x transprecision FPUs
  - 1x PULPO accelerator
  - 64 kB TCDM in 8 banks

- Ariane RV64 host processor
  - 128 KiB Shared LLC
  - software-managed IOMMU

- DDR3 DRAM Controller + PHY by TUKL
The reset can not be released for clusters

- Chip has many modules
  - 1x Ariane core
  - 1x DDR interface
  - 2x Clusters

- Reset to clusters is stuck 0
  - Design flow mistake
  - Some other control signals are stuck as well affecting Ariane performance

- DDR interface is functional
  - Not everything is lost
IC Design is tricky and demands attention

- Even the simplest things can derail a complex chip
  - A copy paste error in a bonding diagram, a mistake in reset

- Academic research chips are not industrial products
  - Designed to test and verify ideas, not mass production
  - Much more effort needed in DfT and verification to make a successful product

- Experience is key in IC Design
  - All the mistakes we make, add to our future success
  - Some lessons you learn the hard way
  - But these stay with you and help you for your future designs
We hope this was helpful/fun for you

- Covered the basics of RISC-V
  - Explained the ISA
  - Examples of Implementations
  - Advanced cores and Concepts

- Talked about building open source systems around RISC-V
  - Showed the main concepts and talked about our ICs

- You can find PULP related information
  - GitHub: http://github.com/pulp_platform
  - PULP Webpage: http://pulp-platform.org
  - Follow us on Twitter: @pulp_platform
Luca Benini, Davide Rossi, Andrea Borghesi, Michele Magno, Simone Benatti, Francesco Conti, Francesco Beneventi, Daniele Palossi, Giuseppe Tagliavini, Antonio Pullini, Germain Haugou, Manuele Rusci, Florian Glaser, Fabio Montagna, Bjoern Forsberg, Pasquale Davide Schiavone, Alfio Di Mauro, Victor Javier Kartsch Morinigo, Tommaso Polonelli, Fabian Schuiki, Stefan Mach, Andreas Kurth, Florian Zaruba, Manuel Eggimann, Philipp Mayer, Marco Guermandi, Xiaying Wang, Michael Hersche, Robert Balas, Antonio Mastrandrea, Matheus Cavalcante, Angelo Garofalo, Alessio Burrello, Gianna Paulin, Georg Rutishauser, Andrea Cossettini, Luca Bertaccini, Maxim Mattheeuws, Samuel Riedel, Sergei Vostrikov, Vlad Niculescu, Hanna Mueller, Matteo Perotti, Nils Wistoff, Luca Bertaccini, Thorir Ingulfsson, Thomas Benz, Paul Scheffler, Alessio Burello, Moritz Scherer, Matteo Spallanzani, Andrea Bartolini, Frank K. Gurkaynak, and many more that we forgot to mention

http://pulp-platform.org @pulp_platform