The PULP team and open source hardware at ETH Zürich

Part of the Digital Design and Computer Architectures 2024 course

Frank K. Gürkaynak  kgf@iis.ee.ethz.ch

PULP Platform
Open Source Hardware, the way it should be!
Introducing the PULP team at ETH Zurich

• Started by Prof. Luca Benini at D-ITET

• We wanted to design energy efficient computing systems
  • Equally efficient for IoT and HPC over a wide range

• Key points
  • Parallel processing
  • Near threshold computing
  • Efficient switching between operating modes
  • Making best use of technology
  • Heterogeneous acceleration

• Parallel Ultra Low-Power (PULP) platform was born
Team of 100 people in ETH Zürich – University of Bologna

- Research on open-source energy-efficient computing architectures

Started in 2013, celebrated 10 years of PULP last year

Led by Luca Benini

Involves ETH Zürich (Switzerland) and University of Bologna (Italy)

Large group of almost 100 people
Too much to do and not enough resources

Energy efficient systems that can process more on edge nodes

Less data to transmit, large savings on energy needed for data transmissions

Accelerators for more efficient computing
We have designed and tested more than 55 PULP ICs

<table>
<thead>
<tr>
<th>Year</th>
<th>Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2013</td>
<td>(3)</td>
<td>PULPv1 STM 28FDSoI Multi-core processor</td>
</tr>
<tr>
<td>2014</td>
<td>(5)</td>
<td>Diana UMC 65 4-core system with approximate FPUs</td>
</tr>
<tr>
<td>2015</td>
<td>(10)</td>
<td>Fulmine UMC 65 4-core system with ML and Crypto accelerators</td>
</tr>
<tr>
<td>2016</td>
<td>(3)</td>
<td>VivoSoC 2.001 SMIC 130 Mixed signal system for biosignal acquisition</td>
</tr>
<tr>
<td>2017</td>
<td>(2)</td>
<td>Mr. Wolf TSMC 40 8+1 core IoT processor</td>
</tr>
<tr>
<td>2018</td>
<td>(6)</td>
<td>Poseidon GF 22FDX 64bit RISC-V core, 32bit Microcontroller system, ML accelerator</td>
</tr>
<tr>
<td>2019</td>
<td>(7)</td>
<td>Baikonur GF 22FDX Dual 64bit RISC-V core, 3x 8core snitch clusters, Body biasing test vehicle</td>
</tr>
<tr>
<td>2020</td>
<td>(3)</td>
<td>Dustin TSMC 65 IoT processor with 16 cores and QNN enhancements</td>
</tr>
<tr>
<td>2021</td>
<td>(7)</td>
<td>Kraken GF 22FDX IoT processor with Spiking Neural and Ternary Inference Engines</td>
</tr>
<tr>
<td>2022</td>
<td>(9)</td>
<td>Occamy GF 12LPP ML accelerator with 216 + 1 cores and HBM interface</td>
</tr>
</tbody>
</table>

Check http://asic.ethz.ch for all our chips
Including this one

OCCAMY

432 RISC-V cores

GF12LPP

HBM2E
RISC-V Instruction Set Architecture

- Started by UC-Berkeley in 2010
- Open Standard governed by RISC-V foundation
  - ETHZ is a founding member of the foundation
  - Necessary for the continuity
  - Extensions are still being developed
- Defines 32, 64 and 128 bit ISA
  - No implementation, just the ISA
  - Different RISC-V implementations (both open and close source) are available

Spec separated into “extensions”

<table>
<thead>
<tr>
<th>Spec</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>Integer instructions</td>
</tr>
<tr>
<td>E</td>
<td>Reduced number of registers</td>
</tr>
<tr>
<td>M</td>
<td>Multiplication and Division</td>
</tr>
<tr>
<td>A</td>
<td>Atomic instructions</td>
</tr>
<tr>
<td>F</td>
<td>Single-Precision Floating-Point</td>
</tr>
<tr>
<td>D</td>
<td>Double-Precision Floating-Point</td>
</tr>
<tr>
<td>C</td>
<td>Compressed Instructions</td>
</tr>
<tr>
<td>X</td>
<td>Non Standard Extensions</td>
</tr>
</tbody>
</table>

We will cover ISA in our lecture soon.
Why is RISC-V so special: Freedom to Explore and Fail!

• The ISA provides a contract between HW and SW
  • As long as you stick to the ISA, you can develop HW and SW independently
  • All RISC-V research in HW can continue to rely on growing SW ecosystem for RISC-V

• RISC-V comes with plenty of options for extensions
  • There are reserved encoding spaces for instruction set extensions

• Being able to change everything gives great flexibility
  • Do you want 33 registers, or a 48 bit accumulator.. No problem
  • You need to bring the SW support for your additions.

MIPS in our lecture is a pre-cursor of RISC-V
<table>
<thead>
<tr>
<th>32 bit</th>
<th>64 bit</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Low Cost Core</strong></td>
<td><strong>Linux capable Core</strong></td>
</tr>
<tr>
<td>IBEX</td>
<td>Ariane/CVA6</td>
</tr>
<tr>
<td>Zero-riscy</td>
<td>RV64-ICMAFD</td>
</tr>
<tr>
<td>Micro-riscy</td>
<td>Maintained by OpenHW</td>
</tr>
<tr>
<td>2 options</td>
<td></td>
</tr>
<tr>
<td>RV32-ICM</td>
<td></td>
</tr>
<tr>
<td>RV32-CE</td>
<td></td>
</tr>
<tr>
<td><strong>Core with DSP support</strong></td>
<td><strong>Core for Streaming</strong></td>
</tr>
<tr>
<td>CV32E40P</td>
<td>Snitch</td>
</tr>
<tr>
<td>RV32-ICMXF</td>
<td>RV32-IMAFD</td>
</tr>
<tr>
<td>SIMD</td>
<td>Small Int core</td>
</tr>
<tr>
<td>HW loops</td>
<td>Big 64-bit FPU</td>
</tr>
<tr>
<td>Bit manipulation</td>
<td>Extensions for streaming</td>
</tr>
<tr>
<td>Fixed point</td>
<td></td>
</tr>
<tr>
<td><strong>Brand New Core</strong></td>
<td><strong>Brand New Core</strong></td>
</tr>
<tr>
<td>Maintained by LowRISC</td>
<td>Maintained by LowRISC</td>
</tr>
<tr>
<td>Maintained by OpenHW</td>
<td>Brand New Core</td>
</tr>
<tr>
<td>Brand New Core</td>
<td></td>
</tr>
</tbody>
</table>

**Notes:**
- Low Cost Core: Low Cost Core
- Core with DSP support: Core with DSP support
- Core for Streaming: Core for Streaming
- Linux capable Core: Linux capable Core
- 32 bit and 64 bit: 32 bit and 64 bit
- Options: Options
- Maintained by LowRISC: Maintained by LowRISC
- Maintained by OpenHW: Maintained by OpenHW
- Brand New Core: Brand New Core
- Fixed point: Fixed point
- Extensions for streaming: Extensions for streaming
- Bit manipulation: Bit manipulation
- SIMD: SIMD
Our research is not implementing RISC-V cores

- **We develop efficient programmable architectures**
  - Processor cores of various capabilities are required for that

- **We need efficient implementations of cores for our research**
  - To produce relevant results, our cores have to perform **as good as other solutions**
  - We ended up spending quite an effort to make sure they perform really well

- **Processor core alone is not enough**
  - You need peripherals, interconnect solutions, programming support...

- **PULP Platform provides us a playground for our research**
  - And we share it as open source
What PULP provides is a box of building blocks

**RISC-V Cores**
- RI5CY: 32b
- Ibex: 32b
- Snitch: 32b
- Ariane: 64b

**Platforms**
- Single Core
  - PULPino
  - PULPissimo
- Multi-core
  - OpenPULP
  - Mr. Wolf
- Multi-cluster
  - Occamy
  - Hero

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

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

**Accelerators**
- HWCE (convolution)
- Neurostream (ML)
- HWCrypt (crypto)
- PULPO (1st order opt)
All of our designs are open-source hardware

Every single entry in the GitHub Trending SystemVerilog repositories for the past month:
- is either directly from our group
- or has originated in our group
For us open source is a necessity, not an idealogic crusade

- Managing Complex Designs
- Faster Collaborations
- Facilitates Industry/Academia Relationships
- Auditable Designs, Reproducible Results
In the last 20 years IC Design has changed a lot.

What used to be a complete chip is now a small part of a SoC!
There is so much that makes up a modern SoC.

The PULP team and open source hardware at ETH Zürich.
In a typical design, innovation is only in a limited scope

Open-source silicon-proven SoC template helps concentrate work where it counts

The PULP team and open source hardware at ETH Zürich
Faster collaborations through open source

• No background IP
  • Everything is already open

• No long discussions on IP ownership
  • Open source licensing for OUR foreground

• Friendly licensing for commercial purposes
  • Permissive licensing allows commercial exploitation, foreground of partners can be closed

• We can start right away
  • Time is spent on design not necessary paperwork
  • The licensing settles most of the needed discussions (who owns it, who can do what)

• You can see what we have and evaluate us in advance
PULP has been used in all our publicly funded research

Swiss Funding (nano-era)
ICYSoC
Near threshold computing

EU funded research projects
OPRECOMP
Approximate computing
Multi-precision arithmetic

European FPA
EPI / EUPilot
Snitch based accelerators

International projects
MITACS / Polara
Vector extensions for RISC-V processors
How does PULP collaborate with 3rd parties?

The PULP team and open source hardware at ETH Zürich
The open model led to successful industry collaborations

**Arnold (GF22)**
eFPGA with RISC-V core

**Vega (GF22)**
IoT Processor with ML acceleration

**Marsellus (GF22)**
IoT Processor with low power modes and event based computing

**Siracusa (TSMC16)**
IoT Processor with NVM technology
And many have used our work for their research.

RISC-V week Barcelona 2018

The Deep Learning Revolution and Its Implications for Computer Architecture and Chip Design

ISSCC Keynote 2020 – Nature 2020

An 8-core RISC-V Processor with Compute near Last Level Cache in Intel 4 CMOS

Gregory K. Chen, Phil C. Knag, Carlos Tokunaga, Ram K. Krishnamurthy
Circuit Research Lab, Intel Corporation, Hillsboro, OR, USA, gregory.k.chen@intel.com

The PULP team and open source hardware at ETH Zürich

VLSI Symposium 2022

AutoDMP: Automated DREAMPlace-based Macro Placement

Anthony Agarwala
anagarwa@nvidia.com
NVIDIA Corporation
Santa Clara, CA, USA

Parantip Rajanath
prajanath@nvidia.com
NVIDIA Corporation
Santa Clara, CA, USA

Tian Yang
tyang@nvidia.com
NVIDIA Corporation
Santa Clara, CA, USA

Haoning Ren
haoning@nvidia.com
NVIDIA Corporation
Austin, TX, USA

Geraldo Pradipita
gpradipita@nvidia.com
NVIDIA Corporation
Santa Clara, CA, USA

Ben Keller
benk@nvidia.com
NVIDIA Corporation
Santa Clara, CA, USA

Brucik Khalilani
brucik@nvidia.com
NVIDIA Corporation
Santa Clara, CA, USA

Figure 7: Pre-CTS placements of the logical groups and cell densities of the MemPool Group designs using NanGate 45nm process (Freq. – 335 MHz, Density – 4k), Congestion (IEEE): InnerBus (2.66%/1.54%), AutoDMP (3.48%/1.86%).
Lessons learned from 10+ years of open source

Licensing has challenges

Still not everything can be open (PDK, IP, EDA)

Governance, contributions, maintenance needs help

Ultimately open source HW works even better than expected
The license is a crucial part of open source

- There is a lot of difference between licenses in open source world
  - Permissive (i.e. Apache)
  - Reciprocal (i.e. GPL)
  - Many more or in between

- Each license has a very legitimate reason to exist
  - Choosing one is not very trivial
  - There are a lot of dogmatic decisions
  - People need to understand and be aware of the pros/cons

- Yet to see an open source hardware to be challenged in court
  - Lawyers have opinions, only some of them are right and could prove this in court
Maybe we can use open source SW licenses?

Virtually Indistinguishable from Standard Software
Some (surprising) side effects of open source hardware

- There is a surprising amount of bureaucracy involved
  - Code Ph.D. students/staff develop belongs to the university (they pay us)
  - Master/semester thesis students own the work they produce
  - Need to get proper approval for everyone involved.

- Most agreements with companies are not meant for open source
  - Instead of paying for exclusive IP, we need sponsoring agreements
  - Important to make sure we do not sign anything that binds our open source effort

- There are worries that if you see inside you will be more vulnerable to patent trolls / litigation
  - Some companies do not want it known that they use our designs
Simplified IC design flow
Simplified IC design flow: at the moment only RTL

3rd Party IP

Library

PDK

Fab

GDS2

Chip

RTL

Netlist

Verification

Synthesizer

P&R

The PULP team and open source hardware at ETH Zürich
Simplified IC design flow: proprietary tools

EDA vendors limit the output of their tools
Simplified IC design flow: technology provider

Fabs do not make PDK information accessible
Simplified IC design flow: 3rd party IP

3rd party IP when included can limit what can be open sourced
Simplified IC design flow: road to opennes

We are getting there, first fully open chips are underway
Hardware design flows for PCB/FPGA/ASIC are different

• The differences complicate things
  • Not a uniform way to discuss the issue, depends on the design flow

• Different actors, different revenue streams
  • Not every actor in the current design flow, earns money the same way
  • EDA companies are long complaining that they are out of the loop
    Their income is not based on the amount of ICs produced.
  • Not everybody will be happy if open hardware will be more common
  • Important to understand the relationships

• Interestingly most intermediate file formats are open, readable
  • Verilog, Liberty, SPICE, EDIF, CIF, LEF, DEF, OA (open access)
Open source PDKs are the key

• Physical designs will contain technology relevant information
  • Nothing can be released (even if there were perfect open EDA tools) without technology info
  • So no open PDK no open source releases beyond RTL/HLS code

• IP Providers and EDA companies use PDK as an excuse
  • “Our design contains proprietary information on technology, we can not release it further”
  • Open PDKs will allow us to change this
  • Some will have to make a decision,
    support open source and adapt, resist and lose market share

• Some good progress, more is needed
  • Skywater (130nm), IHP (130nm), GF(500nm)
  • A 40nm/65nm open PDK would be a game changer, many viable products could be designed
What is the "secret" information that is in the PDK?

Design Rules
Layer Stack
Transistor Models

Area
Parasitics
Speed

All information on this slide is publicly available. From left to right: taken layout from http://vlsi.wpi.edu, layer stack from http://xfab.com, transistor models from ptm.asu.edu.
Recently Open PDKs: IHP, Skywater, Globalfoundries

- **IHP (130nm)**
  - Fab in Germany (Frankfurt an der Oder)

- **Skywater (130nm)**
  - Popular through OpenLane
  - Supported through efforts of Google
  - [https://www.skywatertechnology.com/sky130-open-source-pdk/](https://www.skywatertechnology.com/sky130-open-source-pdk/)

- **Globalfoundries (500nm)**
  - Name suggests 180nm, but it is a high-voltage process
  - [https://github.com/google/gf180mcu-pdk](https://github.com/google/gf180mcu-pdk)
What about open source EDA?

- It works surprisingly well
  - ETHZ is actively collaborating with UCSD

- Open source EDA will allow you to make your own chips
  - Good for many applications/needs
  - Will be great for teaching
  - A gap to SoA will remain

- Will not replace commercial EDA
  - PPA, reliability, service, support

- Summer school at ETH Zürich
  - June 3rd – 7th

https://efcl.ethz.ch/efcl-summer-school.html
Working together to improve open source design flows

The PULP team and open source hardware at ETH Zürich
Iguana, Basilisk Linux capable fully-open RISC-V processor

Supported by student projects at ETHZ

The PULP team and open source hardware at ETH Zürich
Professional help is needed for maintenance/governance

• We (ETHZ and University of Bologna) are research groups
  • Motivated to develop new architectures and systems
  • We needed efficient RISC-V cores (and peripherals) for our work
  • Not so good (or interested) in providing industrial level support for these cores

• We need help to
  • Provide support
  • Develop industrial verification
  • Governance of open source repositories

• Happy to receive this help from
  • Open HW group (Ariane -> CVA6, RI5CY -> CV32E40P)
  • LowRISC (ZeroRiscy -> Ibex)
  • EU projects (Tristan/Isolde), Europractice
Open source sees no borders

- **Trend to ‘nationalize’ processor activities**
  - There is **no** ‘European/Chinese/American Open Source’,
  - There **can be** ‘European/Chinese/American support for Open Source’
  - Open source is global, but it can have **more or less** support in regions/countries

- **Currently open source helps with some challenges related to export controls**
  - If something is public knowledge, you can not demand some people not know it
  - This is more an unintended side effect, rather than explicit design
  - Unlikely that it will be possible to challenge it
In a nutshell: Why Open Source Hardware?

• **It is a necessity**
  - We can not afford to make everything ourselves, we need to collaborate
  - Makes it possible to work together quickly
  - Your results are more trustworthy, anybody can verify it!

• **It works**
  - We have more projects, and more funding due to our open source activities
  - We were able to start many interesting and fruitful collaborations

• **It helps others as well**
  - Many companies, universities, individuals are using pieces of PULP
  - There is already significant commercial use, some we don’t even know about
The future of open source HW is bright