PULP PLATFORM
Open Source Hardware, the way it should be!

Hype vs Reality

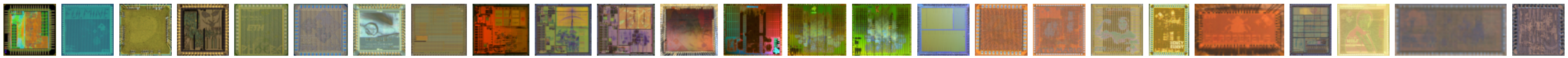
What can RISC-V do for research in safety, reliability and security

Frank K. Gürkaynak, ETH Zürich

http://pulp-platform.org @pulp_platform

PULP started in 2013

- **Luca wanted to work on NEW energy efficient architectures**
  - Keywords were: parallel processing, near threshold operation, energy efficiency
  - **P**arallel **U**ltra **L**ow-**P**ower platform was born
- **Large group of 60 people in ETH Zurich and University of Bologna**
  - Working on technology, IC design, architecture, programming, and applications.
- **At the moment month, we have 34 ASICs taped out**
  - **4x** 22nm, **4x** 28nm, **1x** 40nm, **15x** 65nm, **5x** 130nm, **5x**180nm



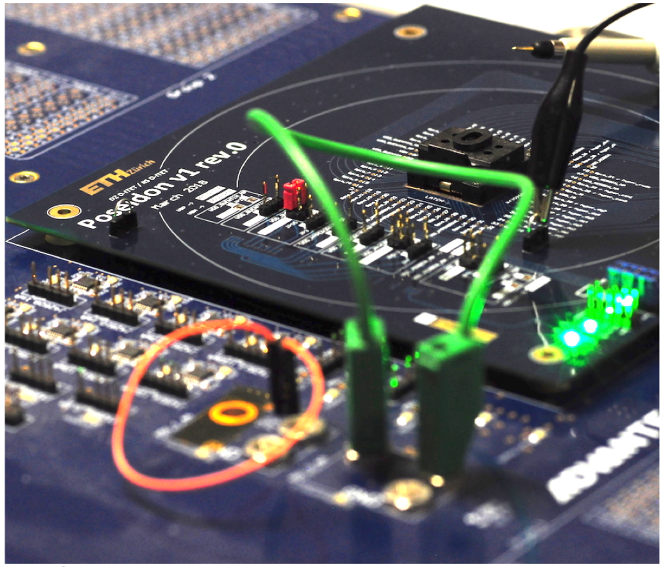
+ 2 more that still need a picture

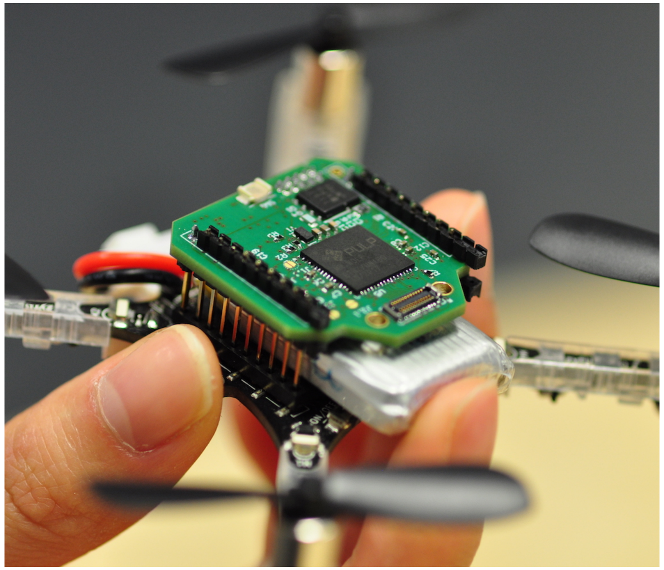
Committed to open source from day one

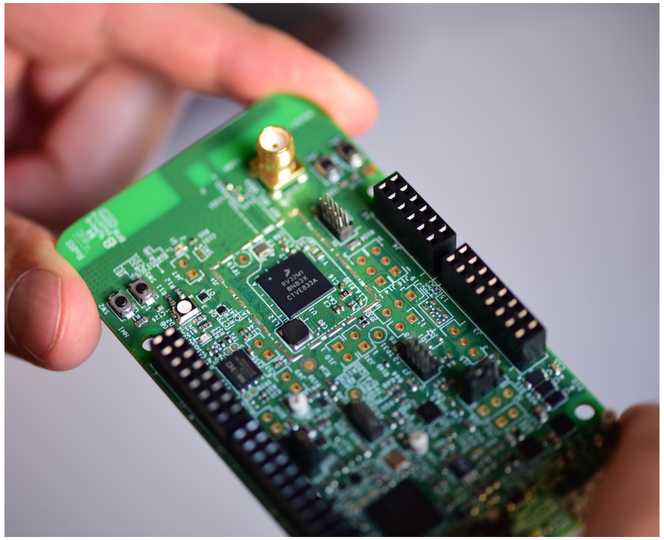
- **Our goal was to release everything (we could) as open source**
  - There are still discussions on what can be released (HDL source, scripts, netlist, GDS)
  - PULP has been using a permissive Solderpad license since the beginning
- **Our first open source release was in February 2016 (PULPino)**
  - Very simple microcontroller using a single 32-bit RISC-V core (RI5CY)
- **As of now (2019) we have released:**
  - Single core platforms: PULPino, PULPissimo
  - Cluster-based multi-core platforms: OpenPULP, HERO, Open Piton + Ariane
  - And a range of RISC-V cores, peripherals, accelerators and interconnect solutions

Our ASICs have different use cases

- **Chips characterized on an IC tester** *(Poseidon 22nm)*
- **Research demonstrators** *(Nano drone with Mr. Wolf)*
- **Industrial uses of our cores/peripherals** *(open-isa.org Vega board)*







RISC-V for security and safety is popular

- **The PULP project is a very good platform for collaboration**
  - It is open source, has been silicon proven and can be used for quite powerful systems
  - We have many discussions with project partners about possible projects
- **More than half of the project ideas we discuss are on**
  - Securing processors against side-channel attacks
  - Implementing systems with improved safety and reliability
- **In this talk, I will explain where RISC-V helps and where it doesn't**

RISC-V is maintained by RISC V foundation

- **Founded in 2015**
  - ETH Zürich is a **founding member**
  - More than 275 members
- **ISA is essentially a document**
  - Defines 32/64/128 bit architectures
  - What are the instructions, what effect do they have
- **ISA divided into several extensions**
  - Working groups decide and work on the definitions
  - Several are **ratified**, work continues on others

Name	Description
I	Integer
E	Integer with 16 registers
C	Compressed Instructions
M	Multiplication
F	IEEE 32b floating point
D	IEEE 64b floating point
Q	IEEE 128b floating point
A	Atomic instructions
V	Vector extensions
P	Packed SIMD extensions
B	Bit manipulation
…	and more

RISC-V foundation only **defines** the ISA

- **The ISA is free, implementations can be done by anyone**
  - ETH Zürich specializes in efficient SystemVerilog based open source implementations
    - **RI5CY**: 32bit Micro-processor with DSP extensions (will be part of OpenHW Core-V)
    - **Ibex**: 32bit minimal processor (maintained by LowRISC)
    - **Ariane**: 64 bit Linux capable core (will be part of OpenHW Core-V)
  - There are many others (SiFive, Codasip, Andes, WesternDigital, IIT-Madras,.. and more)
  - Implementations **can also be commercial**, it is only the ISA that is open
- **The foundation is working on a set of compliance tools**
  - Only foundation members are allowed to officially call their implementations RISC-V

What is so special about RISC-V

- **It is FREE**
  - Everybody can build, sell, and make RISC-V cores available
- **It is a modern design, no historical baggage**
  - Some of the more common ISAs (ARM, Intel..) have been around for 20+ years Newer implementations, still need to be compatible to older designs.
  - RISC-V benefited form the mistakes made by others, cleaner design
  - Major design decisions have been properly motivated and explained
- **Reserved space for extensions, modular**
- **Open standard, you can help decide how it is developed**

The FREEDOM in RISC-V is implementation

- **You can access all ISAs without (many) restrictions**
  - SW tools need to be developed so that they can generate code for that ISA for example
- **But most ISAs are closed. Only specific vendors can implement it**
  - If you want to use a core that implements an ISA, you have to license/buy it from vendor
  - So open source SW (for the ISA) is possible (i.e. compilers) but **building HW is not allowed**

RISC-V

Integer Register-Register Operations

RV32I defines several arithmetic R-type operations. All operations read the *rs1* and *rs2* registers as source operands and write the result into register *rd*. The *funct7* and *funct3* fields select the type of operation.

31–25	24–20	19–15	14–12	11–7	6–0
funct7	rs2	rs1	funct3	rd	opcode
7	5	5	3	5	7
0000000	src2	src1	ADD/SLT/SLTU	dest	OP

ARM

C2.9 ADD

Add without Carry.

Syntax

```
ADD{S}{cond} {Rd}, Rn, Operand2
ADD{cond} {Rd}, Rn, #imm12 ; T32, 32-bit encoding only
```

Are RISC-V processors better than XYZ?

- **Actual performance depends on the implementation**
  - RISC-V does not specify implementation details (on purpose)
- **It is a modern design, should deliver comparable performance**
  - If implemented well, it should perform as good as other modern ISA implementations
  - In our (ETH Zürich) experiments, we see no weaknesses when compared to other ISAs
  - It also is not magically 2x better
- **High-end processor performance is not much about ISA**
  - Implementation details like technology capabilities, memory hierarchy, pipelining, and power management are more important.

RISC-V has space for custom instructions (X)

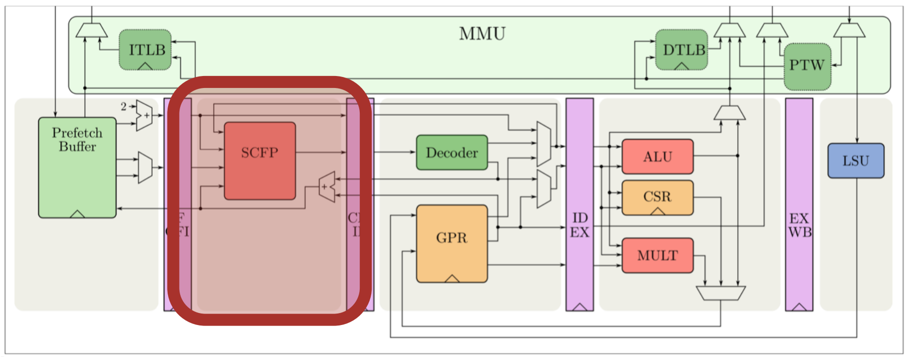
- **There is a reserved decoding space for custom instructions**
  - Allows everyone to add new instructions to the core
  - The address decoding space is reserved, it will not be used by future extensions
  - Any implementation that supports custom instructions will be compatible with standard ISA
    - Code compiled for standard RISC-V will run without issues
  - The user has to provide support to take advantage of the additional instructions
    - Compiler that generates code for the custom instructions
- **ETH regularly uses these instructions**
  - Great tool for exploring
  - The goal is to help ratify some of these extensions as standards through working groups

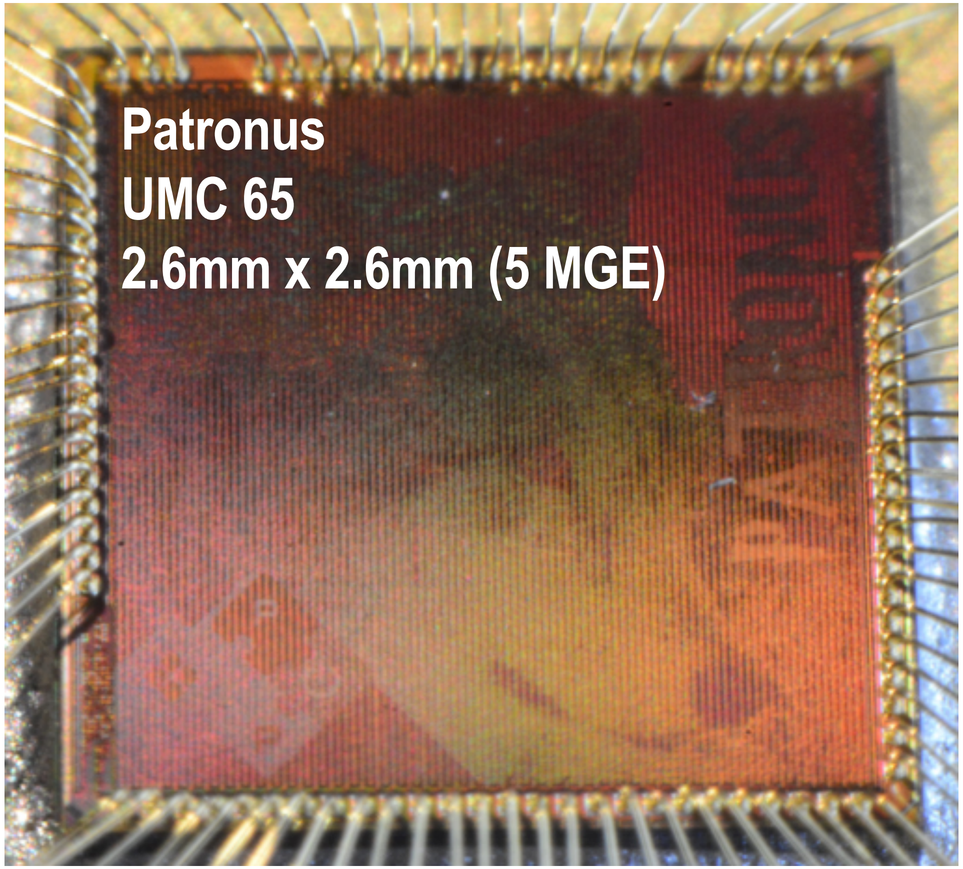
For safety and security RISC-V offers

- **A modern processor ISA that is open**
  - Many implementations, and a good number of them open source that **can be analyzed**
- **It is possible to extend and experiment with it**
  - For open source implementations, easy to get access to them (GitHub)
  - Possible to change and modify the implementation
  - Add extensions to try new ideas and approaches
  - Run comparisons, experiments, benchmarks
- **The changes can also be used further (even commercially)**
  - Also possible to contribute back to the RISC-V community (working groups)

Example project: Control Flow Integrity

- Collaboration with TU-Graz
- Instructions decrypted by state
  - **Additional pipeline stage** in Zero-riscy
  - Sponge based state, PRESENT as cipher
  - Only instructions in right order decrypt properly
  - Need to modify state for branches and calls





Knowing how things exactly work is vital

- **From the "ZombieLoad" paper**

  From section 3.2, emphasis added for this presentation

  "While we identified some necessary building blocks to observe the leakage (cf. Section 5), we ***can only provide a hypothesis on why*** *the interaction of the building blocks leads to the observed leakage. As we could only observe data leakage on Intel CPUs, we assume that this is indeed an implementation issue (such as Meltdown) and not an issue with the underlying design (as with Spectre)."*

- **Closed implementations hide/abstract many secrets from users**
  - Being able to see inside and run experiments are vital for safety and security experts

M. Schwarz, M. Lipp, D. Moghimi, J. Van Bulck, J. Stecklina, T. Prescher, D. Gruss, "ZombieLoad: Cross-Privilege-Boundary Data Sampling", arXiv:1905.05726

Just because it is open, bugs won't go away

- **Debian openssl -- predictable random number generator**
  - On **2 May 2006**, a patch was applied to Debian sources to fix some unitialized variables
  - These unitialized variables were intentionally used as part of the random number generator
  - … which generated seeds for (among others) RSA keys in the openSSL library
  - After the patch, there was very little entropy left. It was possible to guess (private) RSA keys
    - It took only 6 hours to generate all possible 4096 bit RSA keys using 32 Xeon cores.
  - The issue was discovered on **18 May 2008**. Almost exactly two years later
- **A serious security bug remained in plain sight for two years**
  - Although everything was open source and all changes were logged, **nobody noticed!**

David Ahmed, "Two Years of Broken Crypto: Debian's Dress Rehearsal for a Global PKI Compromise", IEEE Security and Privacy, vol 6, pp 70-73, Sep/Oct 2008

What is not so good about RISC-V?

- **Still in development**
  - Some standards (privilege, vector, debug etc.) still being refined, adjusted.
  - Tools and development environment needs to catch up.
- **No canonical implementation (the RISC-V core)**
  - It is free to implement, so many people did so, resulting in many cores
- **Higher end (64 bit, out of order, superscalar) cores not yet mature**
  - In theory there is nothing to prevent a RISC-V based Linux laptop.
  - It will take some more time until RISC-V implementations can compete with other commercial processors (which needed hundreds of man months of work).

There is a huge momentum in RISC-V

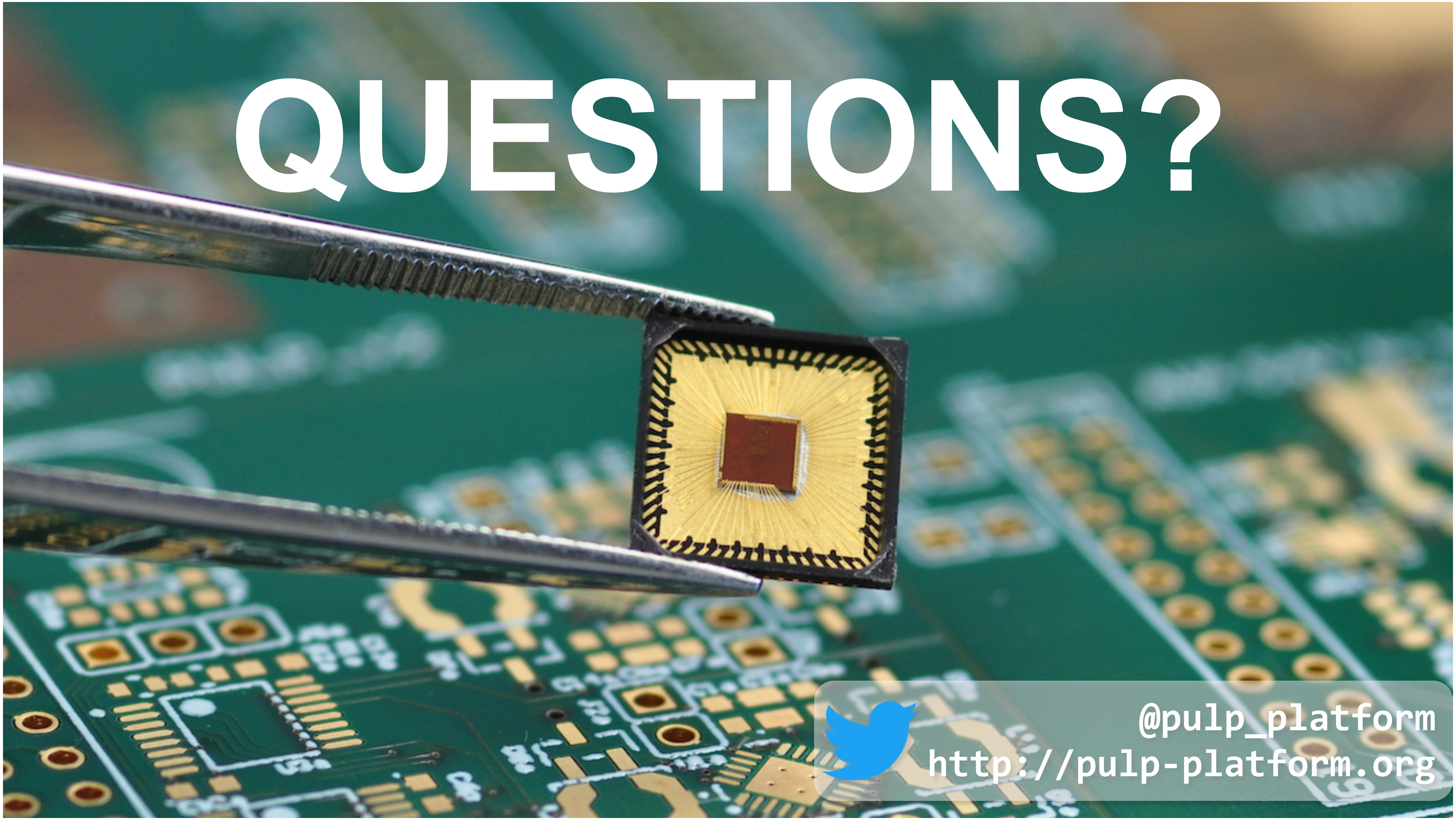
- **In a very short time, RISC-V has firmly established itself**
  - From a processor used in classroom exercises in UC Berkeley
  - Currently RISC-V processors compete with best-in-class commercial microcontrollers
- **There is serious commercial interest and investment**
  - Google, Samsung, NXP, Thales, Nvidia, IBM, Huawei, Infineon, ST Microelectronics
  - New non-profit groups: Chips Alliance, OpenHWgroup to promote open source HW
  - This will only accelerate the development and industrial acceptance
- **Cat is out of the bag, open source HW is here to stay**
  - RISC-V implementations will become more established

What is PULP doing for maintaining cores?

- **We (ETH Zürich 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
- **Goal is to collaborate with groups to maintain our cores/systems**

RISC-V provides solid base for development

- **Allows open source HW implementations**
  - There are already many (high-quality) implementations, more will come
  - Very good base to examine, experiment and develop new ideas
  - These ideas can also be used further
    - Commercial use is possible (assuming license of the implementation allows it, PULP does)
    - Ideas and extensions can be contributed back to the RISC-V ISA (by members of the foundation)
- **However, RISC-V is not magically able to solve all problems**
  - Especially high-end cores are still a few years away from being competitive
  - Many security and safety issues are not only rooted in the processor core
    - Being able to freely develop these systems helps


QUESTIONS?
@pulp_platform
http://pulp-platform.org