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

Will open source hardware solve your security issues?

Frank K. Gürkaynak, ETH Zürich

CS² - 7th Workshop on Cryptography and Security in Computing Systems, 20.01.2020 Bologna, ITALY

 http://pulp-platform.org  @pulp_platform

Will open source HW solve security issues?

NO

… but it can help

Seems like a short talk, what is on the menu?

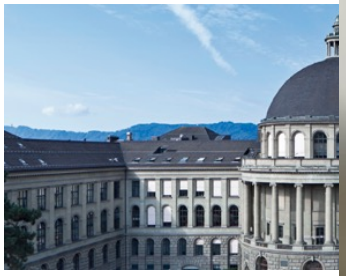
- Who am I, what does the Parallel Ultra Low Power project do?
- How do we see security issues?
- How can open source HW and RISC-V help?
- What have we been doing in this field?
- A brief summary
- Shameless plug for upcoming events
  - Eurolab4HPC Industrial Session, tomorrow at 14:00
  - FOSSistanbul, 13-15 March, Istanbul

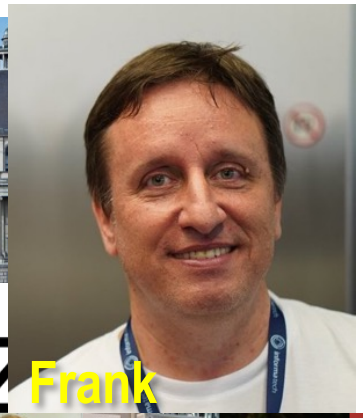
PULP

I am part of the PULP project since 2013

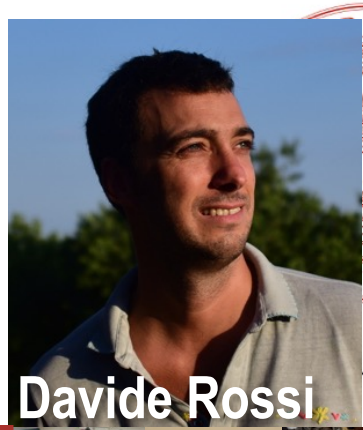
- **Luca wanted to work on NEW energy efficient architectures**
  - Keywords were: parallel processing, near threshold operation, energy efficiency
  - **P**arallel **U**ltra **L**ow-**P**ower platform was born
- **Large group of 60 people in ETH Zurich and University of Bologna**
  - Working on technology, IC design, architecture, programming, and applications.
- **Experienced in ASIC design. We have 37 PULP ASICs taped out**
  - Recent chips in 22nm, 40nm and 65nm
  - See the complete list at `http://asic.ethz.ch`

Who is behind PULP?





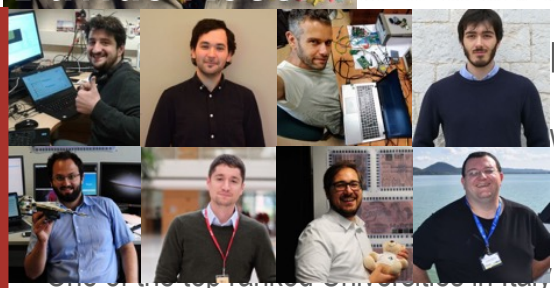
Prof. Luca Benini



Davide Rossi

In total about 60 people work on projects related to PULP in Zurich and Bologna

https://pulp-platform.org/team.html



- Chief Architect in STMicroelectronics (2009-2012)

STUDIORUM DI BOLOGNA

na

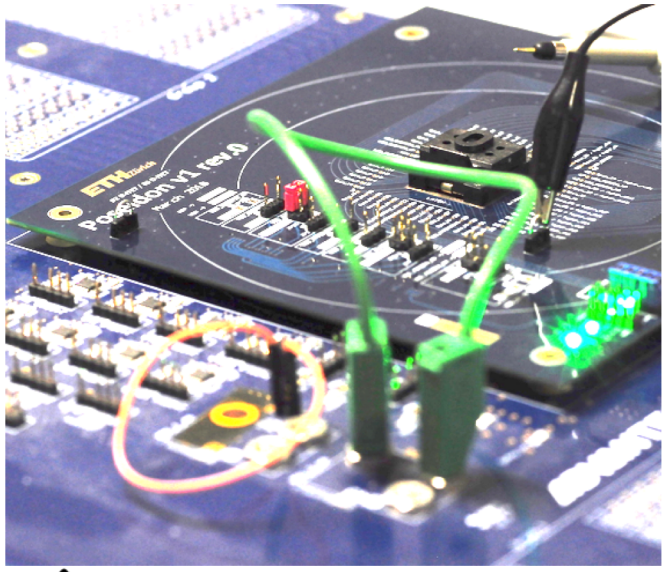
world

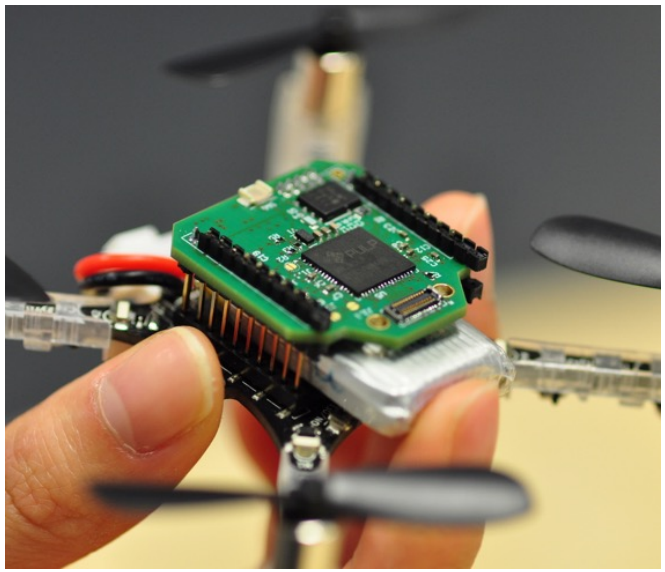
and Europe

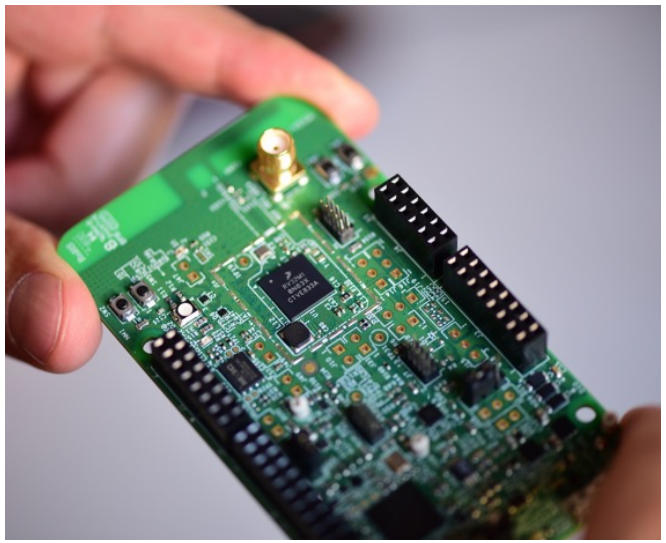
PULP

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)*

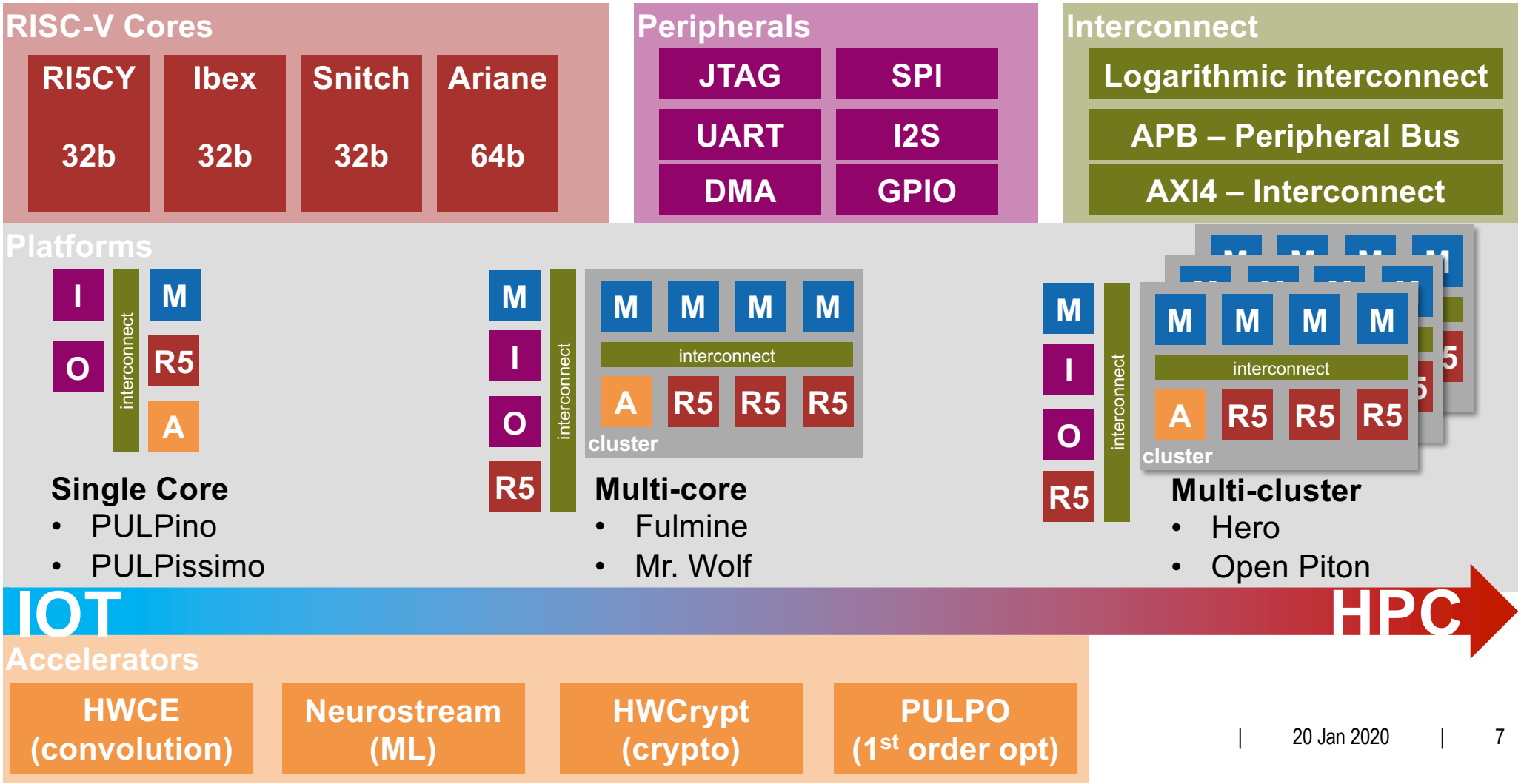






PULP

PULP has released a large number of IPs



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 (start of 2020) 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

Open HW 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
- **Based on this experience:**
  - Allow me to make some comments on the pros and cons of OpenHW in security

Securing systems is a VERTICAL problem

Abstraction Layer	Example	Attacks
Service	E-Voting serv	Legal challenges
Users	Voters	Social engineering
Application	Swiss Po	Bugs / backdoors in SW
Algorithms / libraries	RSA / op	Weaknesses in Algorithms
Operating Systems	seL4	Privilege elevation
Architecture	NXP - i.M	Memory/cache organization
Microarchitecture	RISC-F	Attacks on control flow
Digital Electronics	Adders, g OS	Side channel leakage
Physics	Electrons, Qu um states	Environment

Security issues at ALL Levels

Open ce HW

Solutions are needed at MULTIPLE levels

- **Hardware is only one part of the system**
  - In some cases security problems are caused by multiple levels interacting with each other
- **Open source hardware can help provide solutions**
  - Many application / libraries / operating system solutions rely on open source software
  - Open source hardware broadens the scope of what can be done
  - But it can not solve all the issues alone
- **It is important to understand what it can and can not deliver**
  - We have seen that people have unrealistic expectations from open source hardware
    When these are inevitably not met, open source HW gets unfairly criticized in the end

How can open source HW help?

- Know what is **really** inside
- More and independent **verification** of blocks
- Be able to **experiment** without constraints
- **Share** the information freely
- Fairer **benchmarking**

- After all: **Open source SW has proven useful** why should open source HW be different?

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

It is not that cores from XYZ are insecure

- **Most commercial processors have well thought out solutions**
  - In most likelihood better than anything we have in open source hardware
- **But a security researcher does not always have access**
  - Work and insights can not be shared freely between researchers
  - Experimenting (an important part of research) is limited, you work with what is given
  - Results and changes can not be verified independently
- **This is where open source hardware can help the most**

Enter RISC-V to the rescue

- **RISC-V Foundation established 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

PULP

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**

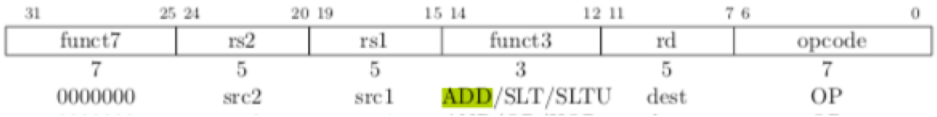
The FREEDOM in RISC-V is implementation

- **You can access all ISAs without (many) restrictions**
  - SW tools need to be developed so that they can generate code for that ISA

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.



ARM

C2.9 ADD

Add without Carry.

Syntax

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

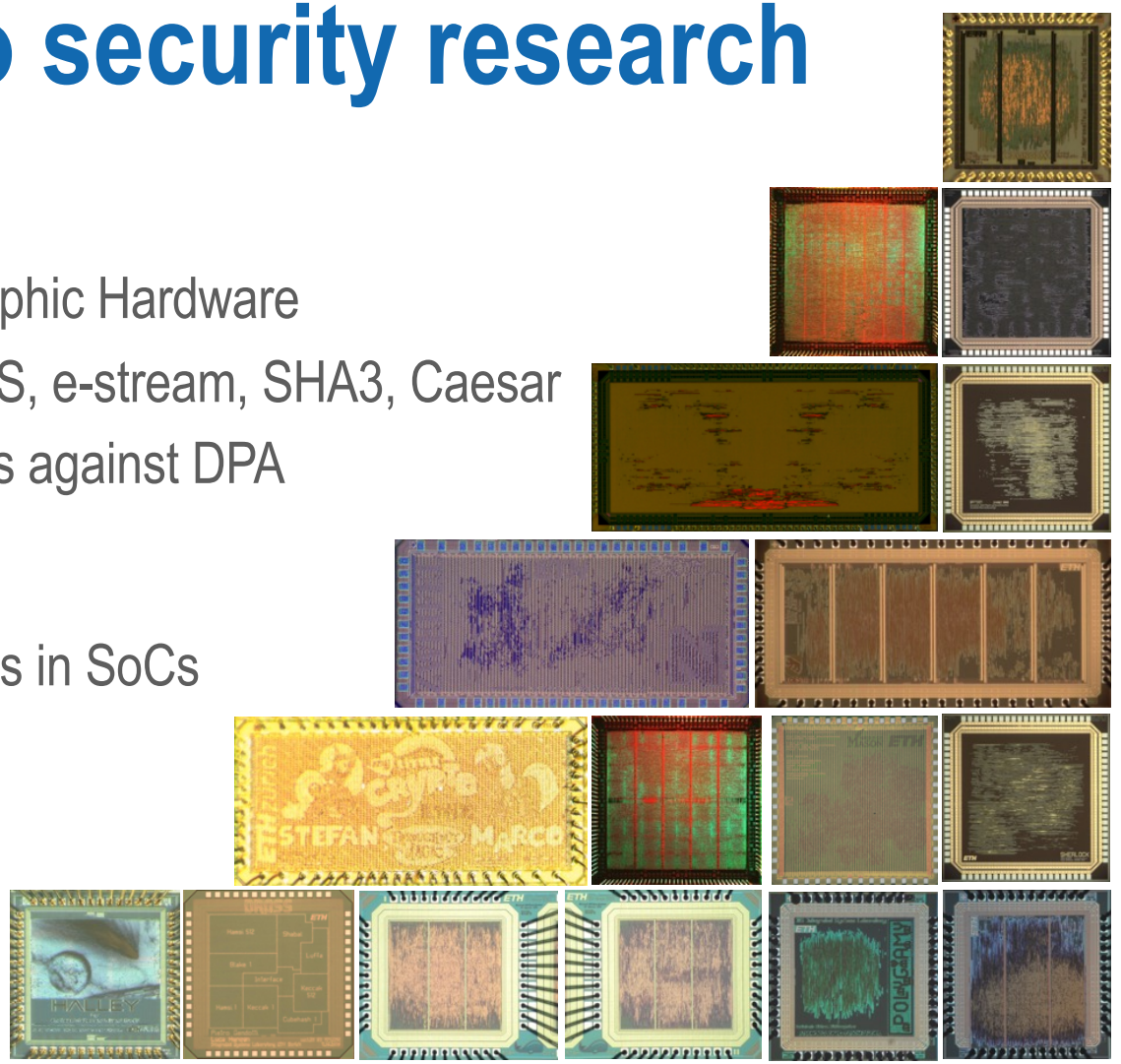
- **But most ISAs are closed. Only specific vendors can implement it**
  - If you want to use a core that implements an ISA, you have to license/buy it from vendor
  - So open source SW (for the ISA) is possible (i.e. compilers) but **building HW is not allowed**

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.

Our contributions to security research

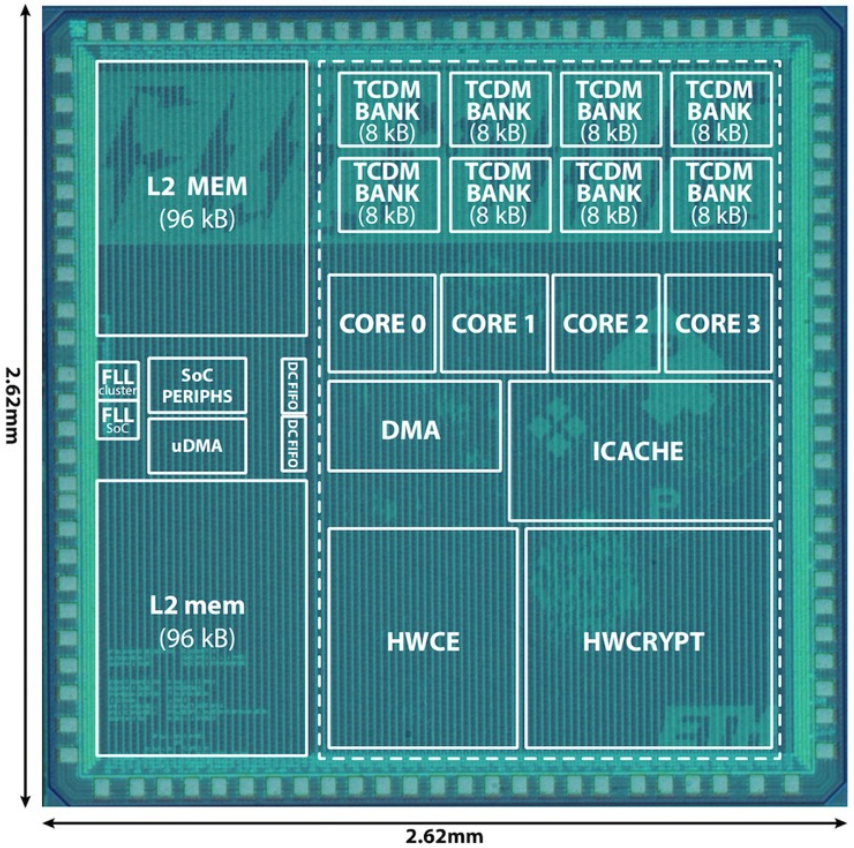
- **Before PULP**
  - Efficient implementations of Cryptographic Hardware
  - Contributions to public evaluations AES, e-stream, SHA3, Caesar
  - Practical attacks and countermeasures against DPA
- **After PULP**
  - Acceleration of Cryptographic functions in SoCs
  - Control flow integrity
  - Side-channel resilience
  - Securing covert channels to prevent information leakage



PULP

Accelerating cryptographic functions

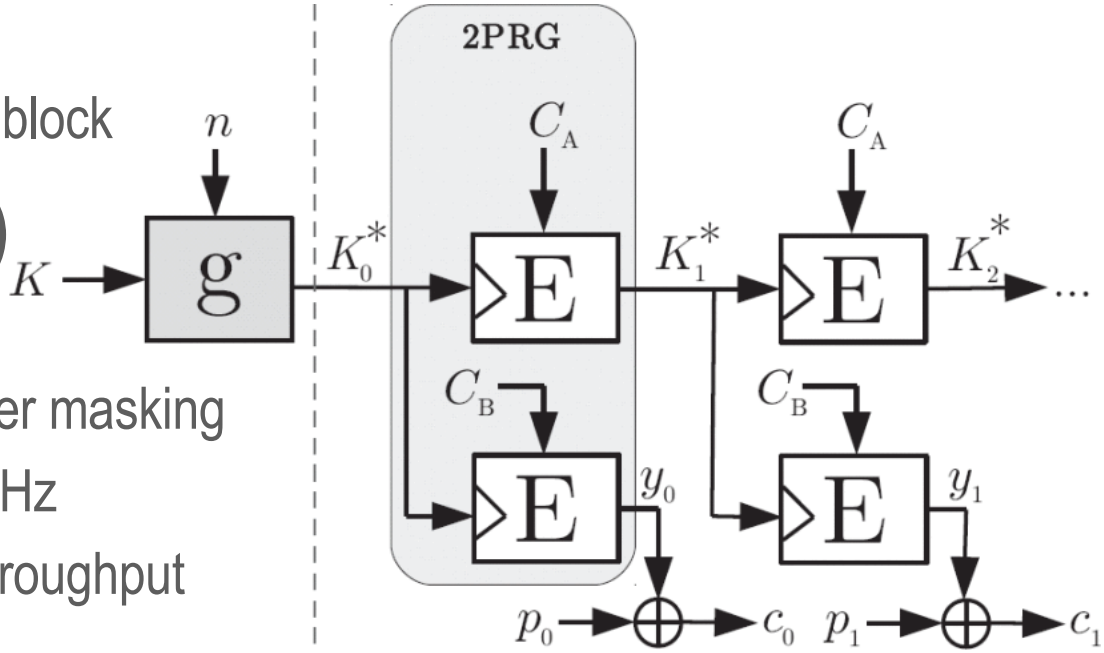
- **Key challenge: I/O bandwidth**
  - Not so difficult to design fast crypto HW
  - Need to match the rest of the system
  - Bandwidth to memory/bus the issue
- **Fulmine (UMC65)**
  - 2 TCDM ports 64bits/cycle
  - AES unit (2 rounds/cycle)
    - 0.38 cpb (8 kByte block); Intel Xeon AES-NI 1.18 cpb
    - @0.8V and 84 MHz, 1.76 Gbit/s, 120 pJ per byte (chip)
  - Also SHA3 unit and other accelerators



F. Conti et al., "An IoT Endpoint System-on-Chip for Secure and Energy-Efficient Near-Sensor Analytics," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 64, no. 9, pp. 2481-2494, Sept. 2017.

Leakage resilient cryptography

- **Reduce Attack surface**
  - A new key (**K***) is generated per data block
- **Encryption example (2PRG)**
  - **E** function is AES
  - **g** finite field multiplication with 1st order masking
  - Max throughput 5.29 Gbit/s @ 256 MHz
  - Needs **2x Block ciphers** for same throughput

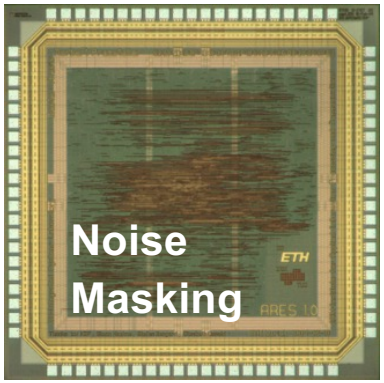


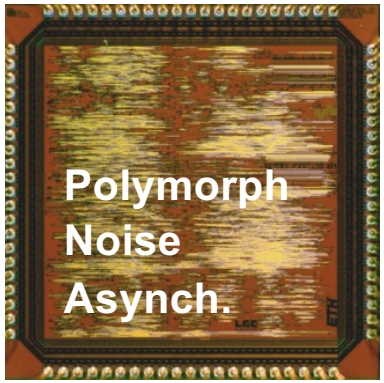
- **Strong side channel resilience within IoT Power budget**
  - Implemented and tested in **Fulmine** (last slide)

Robert Schilling, Thomas Unterluggauer, Stefan Mangard, Frank Gürkaynak, Michael Muehlberghuber, Luca Benini, "High-Speed ASIC Implementations of Leakage-Resilient Cryptography", DATE 2018

Other tricks against side channel security

- **Power by far the most common side-channel attack for CMOS**
- **Basic approaches to protection:**
  - Add **noise** to make measurements difficult
  - Implement **masking**/**sharing** techniques to de-correlate secrets from input data
  - Change the way the operation is organized randomly (**polymorphism**)
  - Use **digital logic** with circuit **styles** that have (less) data dependent consumption

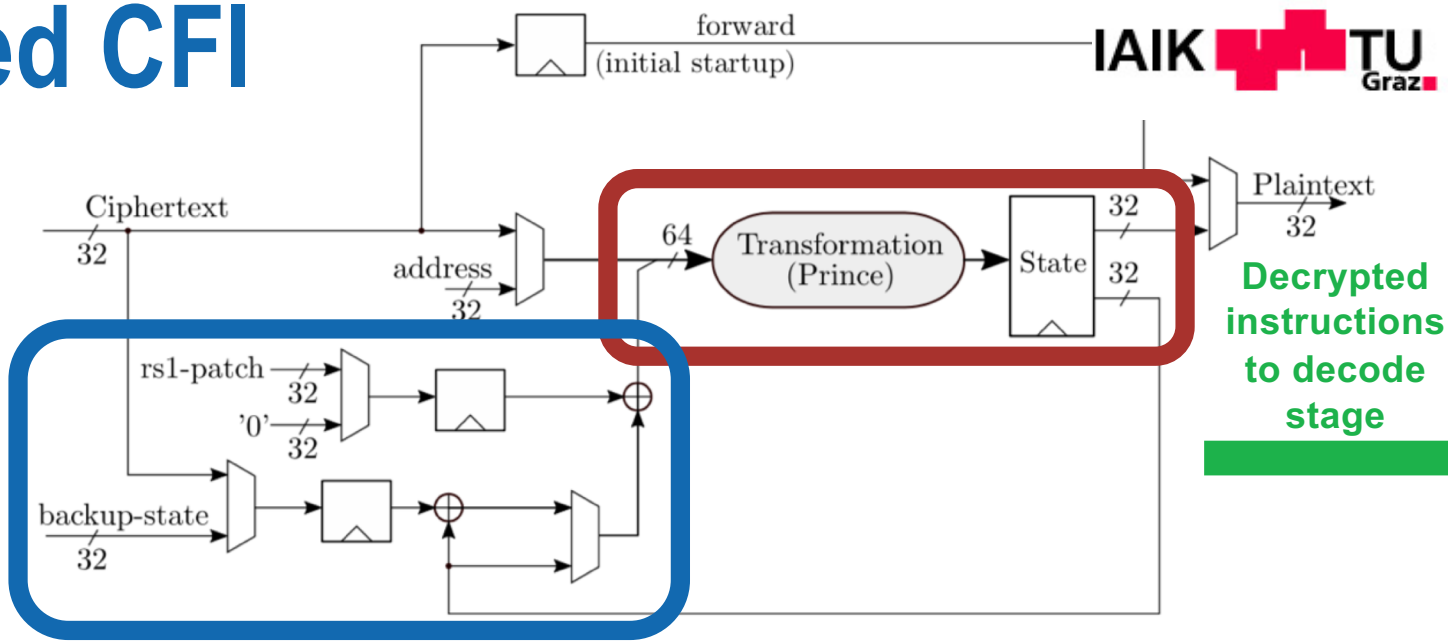




Attacks against the control flow

- **Can be realized in both HW and SW**
  - A successful attack on a processor changes the order of executed instructions
  - Can be used to execute malicious code or jump over security checks
- **HW attacks can be realized by controlling environment**
  - Clock or voltage glitches
  - Injecting electromagnetic pulses
- **Small IoT devices more vulnerable**
  - They operate in potentially hostile environment
  - Have less resources to withstand attacks from a capable adversary

Sponge based CFI



Encrypted instructions from memory

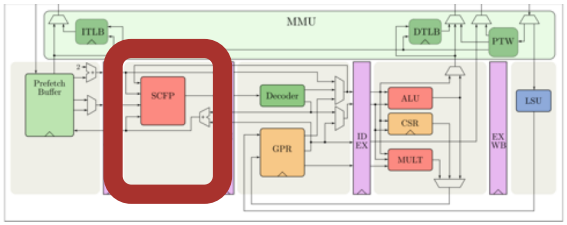
Ability to patch the state for branches/jumps

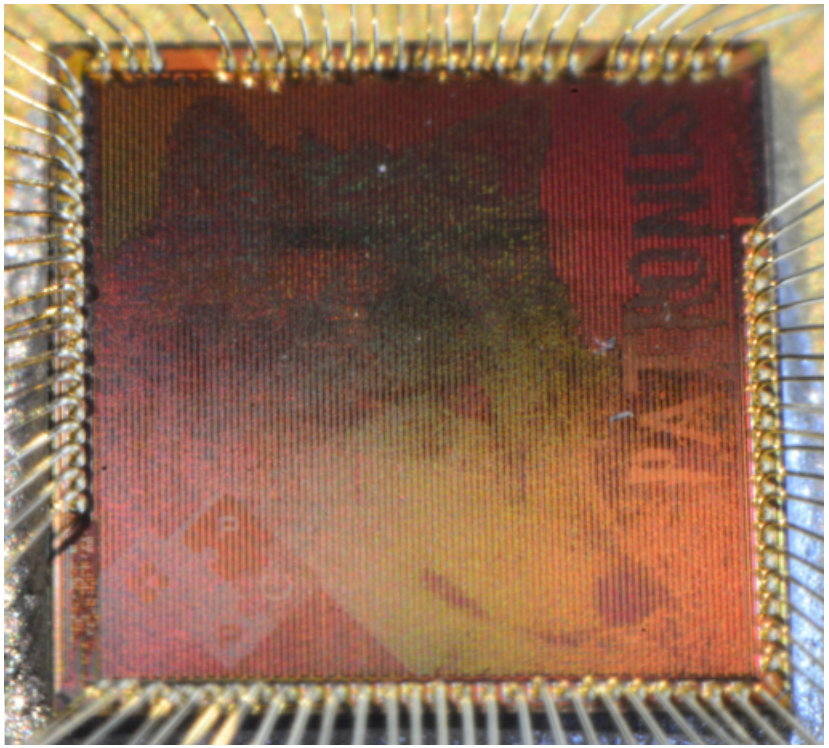
Decrypted instructions to decode stage

- Sponge based construction to decrypt instructions
  - AEE Light with 32 bit **state** and 32 bit capacity in APE mode
  - Used **Prince** for permutation allowing single cycle execution
- Attacker has to **change** instructions and **state** at the same time

Patronus: RISC-V system with CFI

- Additional pipeline stage in Ibex for decryption
  - LLVM based compilation flow
- Only 25-35% power/area overhead
- **Additional instructions** for branches added as instruction set extensions
- About 10% runtime overhead due to patches and additional commands
- Probability of **illegal instruction trap** when instruction altered
  - 91.51% within 1 cycle
  - 99.19% within 2 cycles
  - **99.95%** within 3 cycles





Mario Werner, Thomas Unterluggauer, David Schaffenrath, Stefan Mangard, "Sponge-Based Control-Flow Protection for IoT devices", 2018 IEEE European Symposium on Security and Privacy

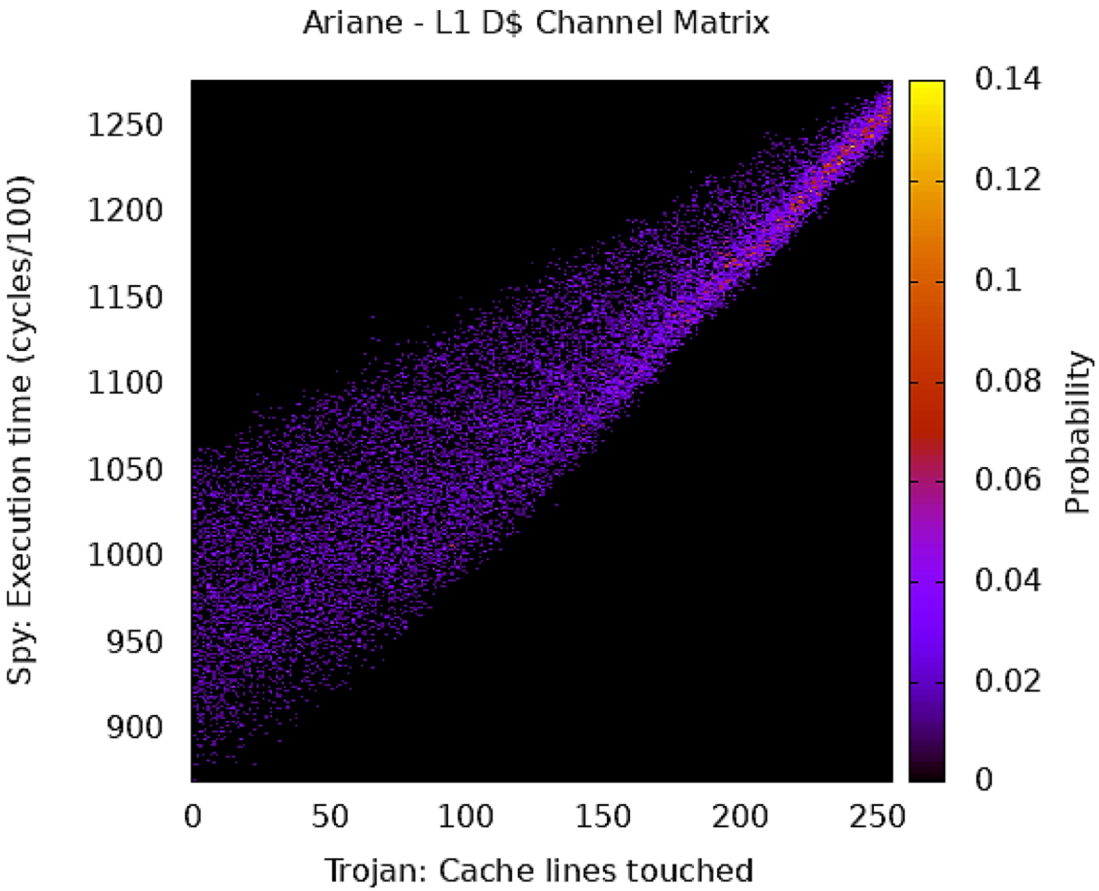
Securing covert channels

- Several attacks are based on passing information between tasks

- Covert channels are used to pass information between tasks
  - Most channels are based on state of hardware that is retained between task switches
  - Branch prediction history, caches, reorder buffers
- Attacks can be mitigated by 'securing' covert channels

Evaluation

- **Master thesis by Nils Wistoff**
  - Perform attacks running on RISC-V cores
  - Investigate and evaluate efficient mitigation mechanisms
- **Simple processor not enough**
  - Most covert channels are due to performance optimizations in high performance processors
  - Ariane (6 stage, in-order 64bit RISC-V)
- **Working seL4 port on Ariane**



How open source helped us in our projects

- **Start from a working system, no need to reinvent everything**
  - Our 32bit microcontrollers are quite mature, Ariane runs Linux and Sel4
- **Easy to extend with accelerators and custom blocks/memories**
  - Platforms designed for heterogeneous acceleration (PULPissimo, OpenPULP, HERO)
- **Not limited by previous design choices**
  - Do you want to have 2 additional bits for your registers, drop instructions, add new ones?
- **Full source code allows you to observe/record everything**
  - Not limited to which performance counters, timers are available, build add/your own
- **Possible to exploit the results commercially**

Not everything is perfect, still work to do

- **Modern attacks exploit features of high performance processors**
  - Current open source offerings not at the same level of high-end commercial processors
  - At the moment, no out of order, multi-issue, SMP systems within PULP ecosystem
  - Designing such cores is not really our research goal
- **Standards of RISC-V will continue to evolve (slowly..)**
  - Standards are discussed openly, this takes time, specifications evolve before being ratified
- **Providing support to all users is not our strength**
  - We try our best, but our main workforce are Ph.D. students that need to do research
  - Get community involved in supporting what we have released so far... and...

Open Titan project

- **Open source Root of Trust project led by LowRISC**

  "OpenTitan is the first open source project building a transparent, high-quality reference design and integration guidelines for silicon root of trust (RoT) chips"

- **Many partners involved**

- **ETH Zurich has contributed their 32bit RISC-V core (Ibex)**

OpenHW Group

GROUP OPENHW PROVEN PROCESSOR IP

OpenHW Group is a not-for-profit, global organization driven by its members and individual contributors where HW and SW designers collaborate in the development of open-source cores, related IP, tools and SW such as the CORE-V Family of cores.

- **OpenHW will take over and support our RI5CY and Ariane cores**
  - Better verification, documentation, user support
- **Visit their booth outside to learn more**

Open HW group members

Open source HW is a great tool for security

- **Investigate, observe, change and share**
  - Great toolbox to test and validate ideas, develop new concepts
- **Platforms with good maturity (and great price/performance)**
  - Huge effort by industry supported non-profits to getting additional manpower to make it even better underway
- **Permissive licensing does not put burdens on commercialization**
- **Access our releases through GitHub**

https://github.com/pulp_platform

Eurolab4HPC Industrial Session: Tue 14:00

In this session our goal is to bring together major supporters of open source hardware projects within the industry and discuss what role Open Source Hardware will play in industrial applications in the coming years.

- **Calista Redmond** (CEO, RISC-V)
- **Rick O'Connor** (President, OpenHW group)
- **Ted Marena** (Director, Chips Alliance)
- **Dominic Rizzo** (Google, Open Titan, LowRISC)
- **John Davis** (BSC, EPI)
- **Christian Fabre** (CEA)

Eurolab4HPC

PULP

FOSSistanbul, March 13-15, Istanbul

FOSSistanbul will bring together, enthusiasts, members of industry and academia that are working on open source hardware design, in a lively and attractive city.

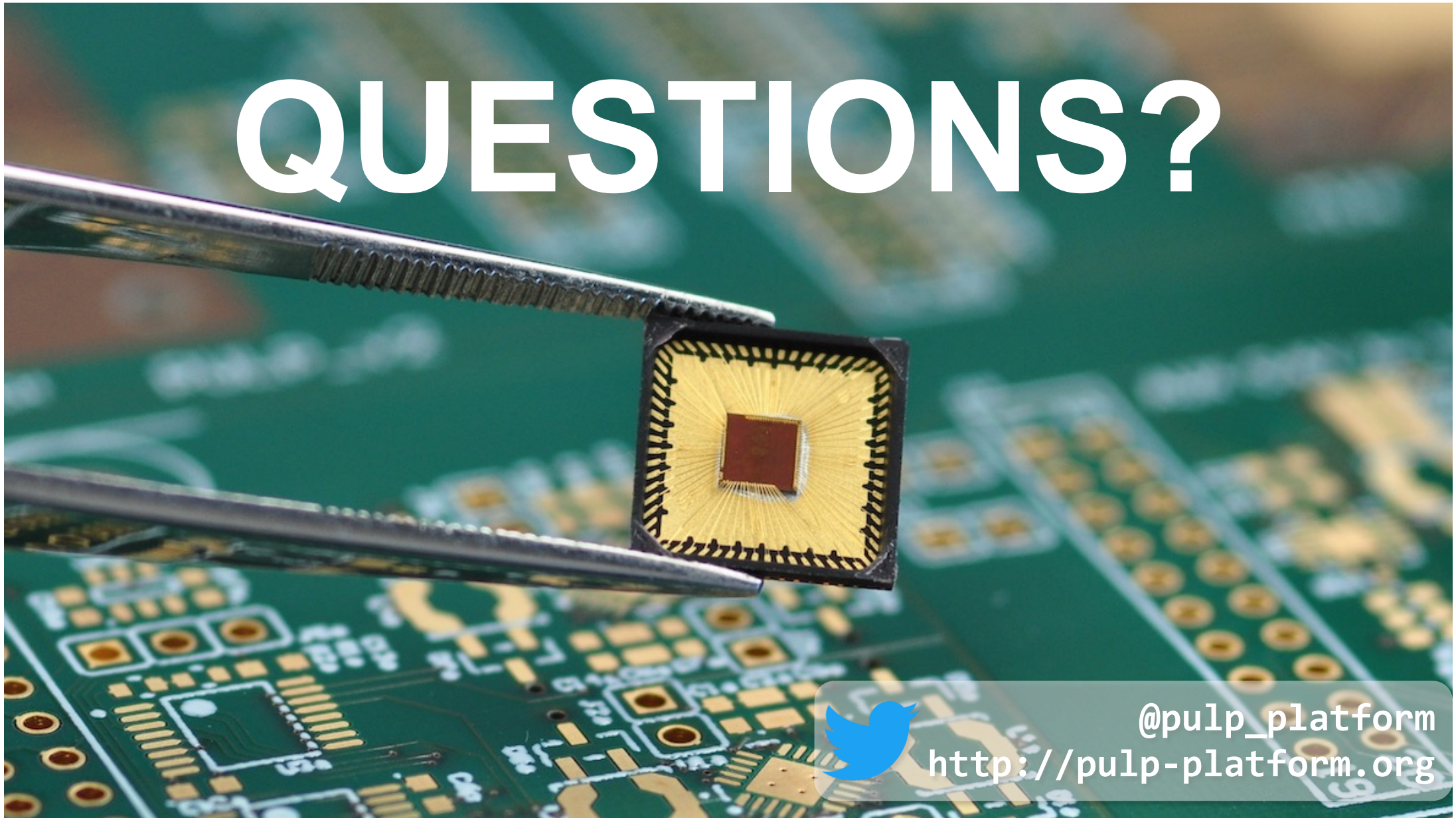
With keynotes by: Luca Benini, Nele Mentens, Onur Mutlu

Register for FREE

https://fossi-foundation.org/fossistanbul/

FOSSIstanbul
13-15 March 2020

PULP


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