ETH zürich

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

vCLIC: Towards Fast Interrupt Handling in Virtualized RISC-V Mixed-criticality Systems

The 42nd IEEE International Conference on Computer Design (ICCD 2024)

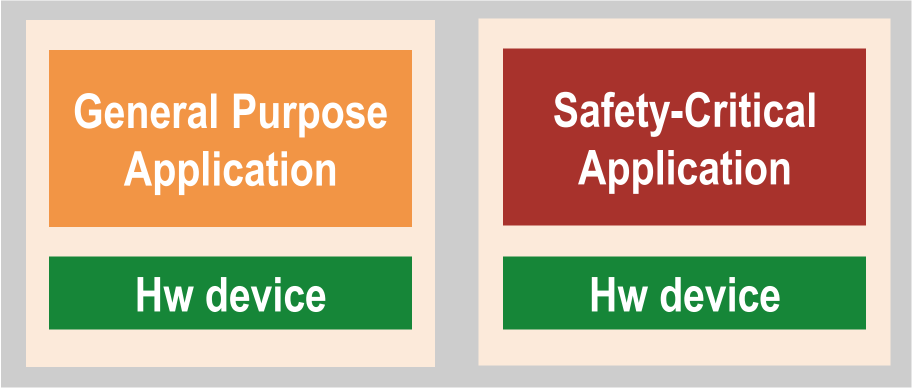
Enrico Zelioli ezelioli@iis.ee.ethz.ch
Alessando Ottaviano aottaviano@iis.ee.ethz.ch
Robert Balas balasr@iis.ee.ethz.ch
Nils Wistoff nwistoff@iis.ee.ethz.ch
Angelo Garofalo agarofalo@iis.ee.ethz.ch
Luca Benini lbenini@iis.ee.ethz.ch

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

@pulp_platform
pulp-platform.org
youtube.com/pulp_platform

Motivation

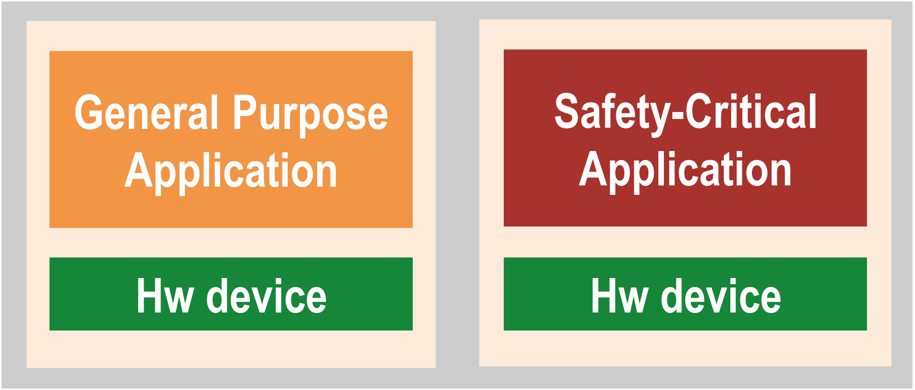
- **MCS integrate applications with different criticality levels**
  - Widely spread (e.g., automotive, space)
  - Modern MCS have high computational demands (e.g. autonomous driving cars)
  - **Challenges**: scalability, cost, connectivity of 100s of components



Partitioned MCS

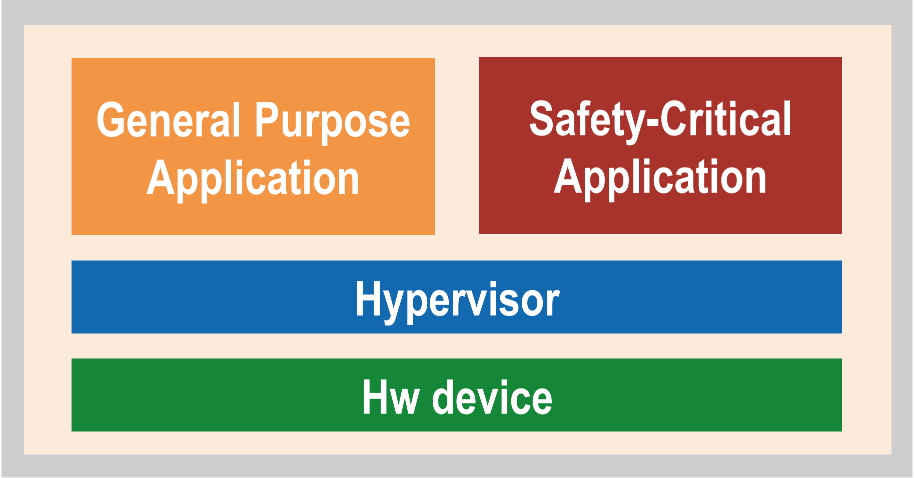
Motivation

- **MCS integrate applications with different criticality levels**
  - Widely spread (e.g., automotive, space)
  - Modern MCS have high computational demands (e.g. autonomous driving cars)
  - **Challenges**: scalability, cost, connectivity of 100s of components



Partitioned MCS

- ***Virtualization* is key**
  - Improves efficiency by sharing hardware resources
  - Hardware support required to reduce performance overhead
  - **Challenges**: isolation, **real-rime responsiveness**



Virtualized MCS

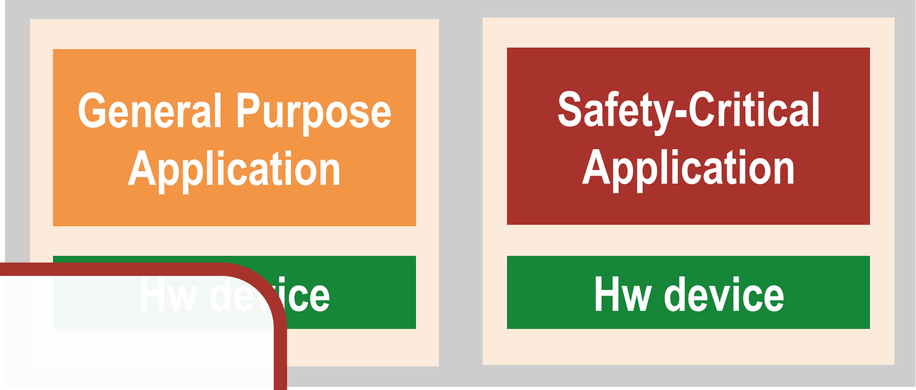
Motivation

- **MCS integrate applications with different criticality levels**
  - Widely spread (e.g., automotive, space)
  - Modern MCS have high computational demands (e.g. autonomous driving cars)
  - **Challenges**: scalability, cost, connectivity of 100s of components

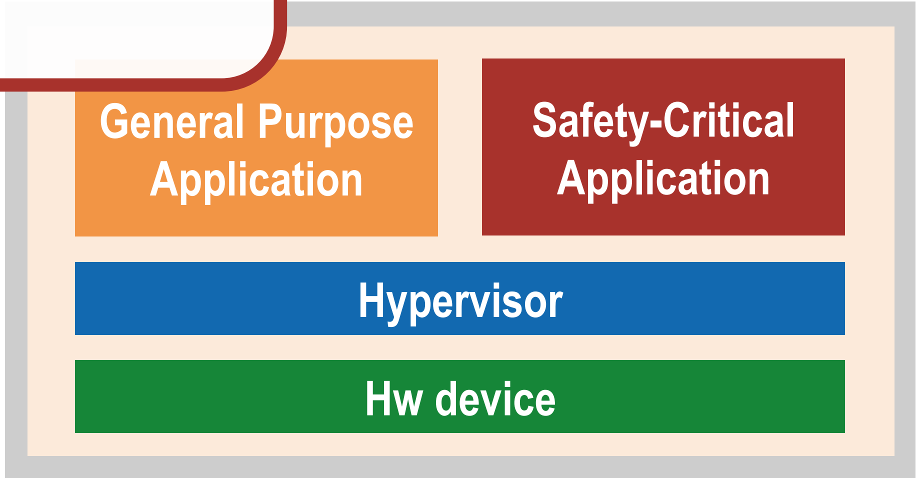
- ***Virtualization* is key**
  - Improves efficiency by sharing hardware resources
  - Hardware support required to reduce performance overhead
  - **Challenges**: isolation, **real-rime responsiveness**

Research Question

How to guarantee real-time **responsiveness** of **safety-critical** components in **virtualized MCS**?



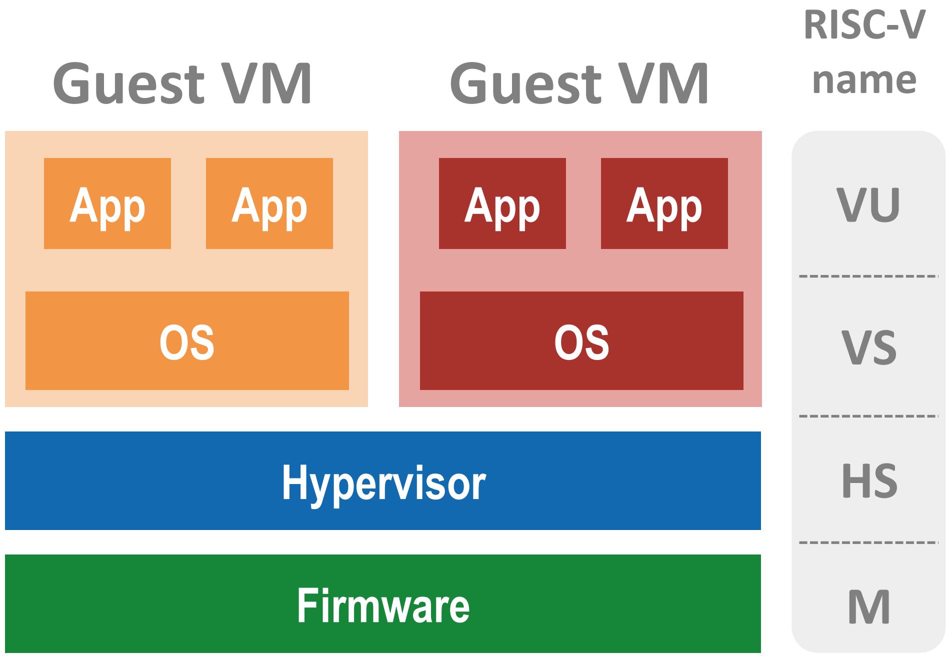
Partitioned MCS



Virtualized MCS

Background

- **RISC-V Mixed-Criticality Systems**
  - Open, modular ISA with wide range of applications
  - Supports virtualization through dedicated **Hypervisor extension**
  - Drives research in safety, security, real-time processing

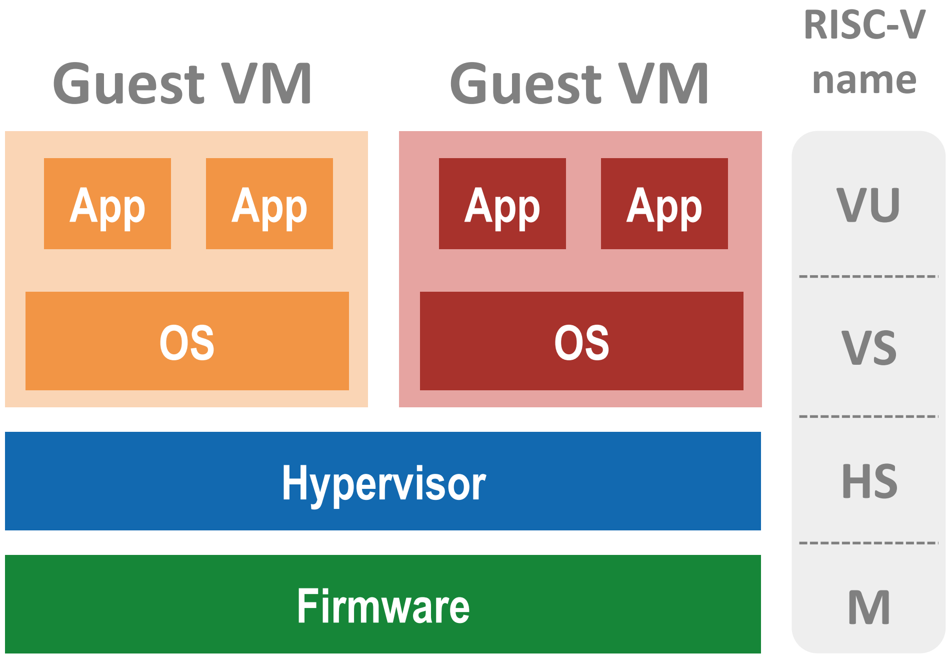


RISC-V Privilege Modes

Background

- **RISC-V Mixed-Criticality Systems**
  - Open, modular ISA with wide range of applications
  - Supports virtualization through dedicated **Hypervisor extension**
  - Drives research in safety, security, real-time processing

- **Interrupt virtualization**
  - Extends *hardware support* for virtualization to interrupt controller
  - Enables direct injection of interrupts to VMs
  - Reduces **interrupt latency** by minimizing hypervisor intervention



RISC-V Privilege Modes

RISC-V Interrupt Controllers

	Designed for Real-time	Supports Virtualization
RISC-V base interrupt architecture [1] • Original spec of interrupt • Core-local + platform-level interrupt controllers	✗	✗

[1] RISC-V Privileged Specification
[2] RISC-V AIA Specification
[3] “Interrupting” the Status Quo (F. Marques et al.)

RISC-V Interrupt Controllers

	Designed for Real-time	Supports Virtualization
RISC-V base interrupt architecture [1] - Original spec of interrupt - Core-local + platform-level interrupt controllers	✗	✗
Advanced Interrupt Architecture (AIA) [2] - Designed for high-performance RISC-V systems - Support for Message-Signaled Interrupts (MSI) - Active research on embedded versions [3]	✗	✓

[1] RISC-V Privileged Specification
[2] RISC-V AIA Specification
[3] "Interrupting" the Status Quo (F. Marques et al.)

RISC-V Interrupt Controllers

	Designed for Real-time	Supports Virtualization
RISC-V base interrupt architecture [1] • Original spec of interrupt • Core-local + platform-level interrupt controllers	✗	✗
Advanced Interrupt Architecture (AIA) [2] • Designed for high-performance RISC-V systems • Support for Message-Signaled Interrupts (MSI) • Active research on embedded versions [3]	✗	✓
RISC-V fast-interrupts (CLIC) • Designed for fast interrupt response • Interrupt vectoring, nesting, tail-chaining	✓	✗

[1] RISC-V Privileged Specification

[2] RISC-V AIA Specification

[3] “Interrupting” the Status Quo (F. Marques et al.)

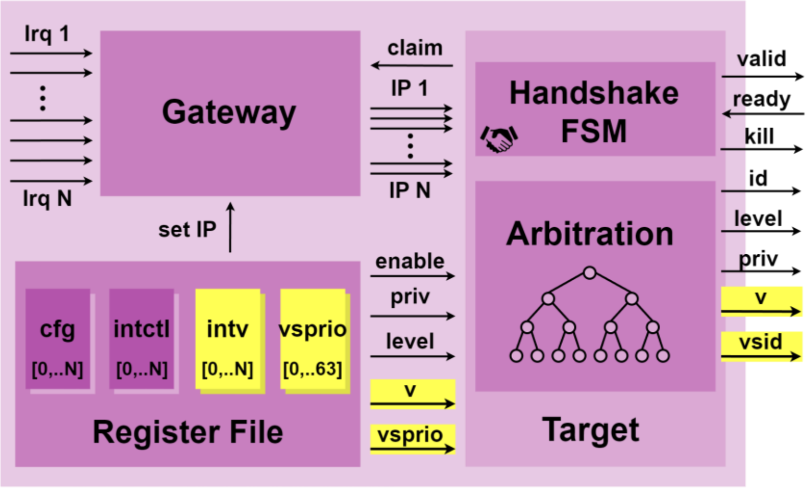
RISC-V Interrupt Controllers

	Designed for Real-time	Supports Virtualization
RISC-V base interrupt architecture [1] - Original spec of interrupt - Core-local + platform-level interrupt controllers	✗	✗
Advanced Interrupt Architecture (AIA) [2] - Designed for high-performance RISC-V systems - Support for Message-Signaled Interrupts (MSI) - Active research on embedded versions [3]	✗	✓
Our Work: vCLIC “Enable fast interrupt response in RISC-V virtualized MCS”	✓	✓

[1] RISC-V Privileged Specification
[2] RISC-V AIA Specification
[3] “Interrupting” the Status Quo (F. Marques et al.)

vCLIC: design and architecture

- **Design principles**
  - **Modularity**
    - *VSCLIC*: direct injection of interrupts
    - *VSPRIO*: prioritization based on VM criticality
  - **Configurability**
    - Fine-grained tuning of resource utilization
  - **Compatibility**
    - Minimal deviation from CLIC spec*



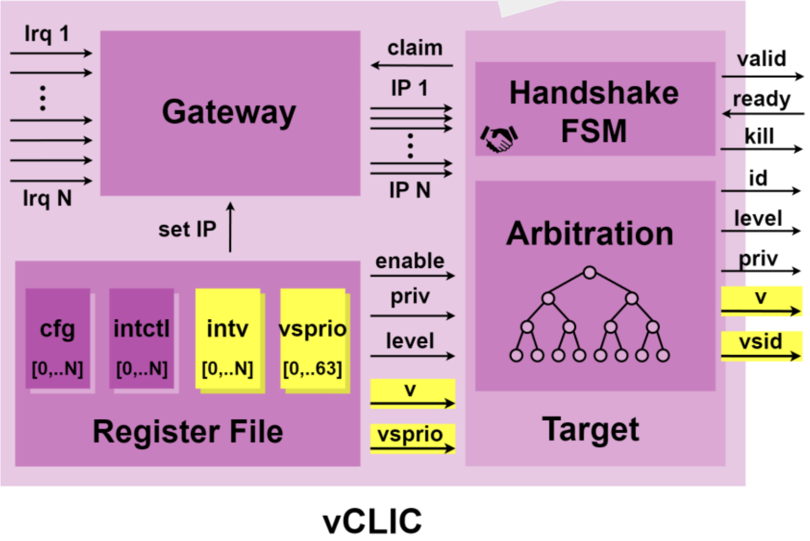
vCLIC

* implemented according to v0.7 of CLIC specification

vCLIC: design and architecture

- **Design principles**
  - **Modularity**
    - *VSCLIC*: direct injection of interrupts
    - *VSPRIO*: prioritization based on VM criticality
  - **Configurability**
    - Fine-grained tuning of resource utilization
  - **Compatibility**
    - Minimal deviation from CLIC spec*

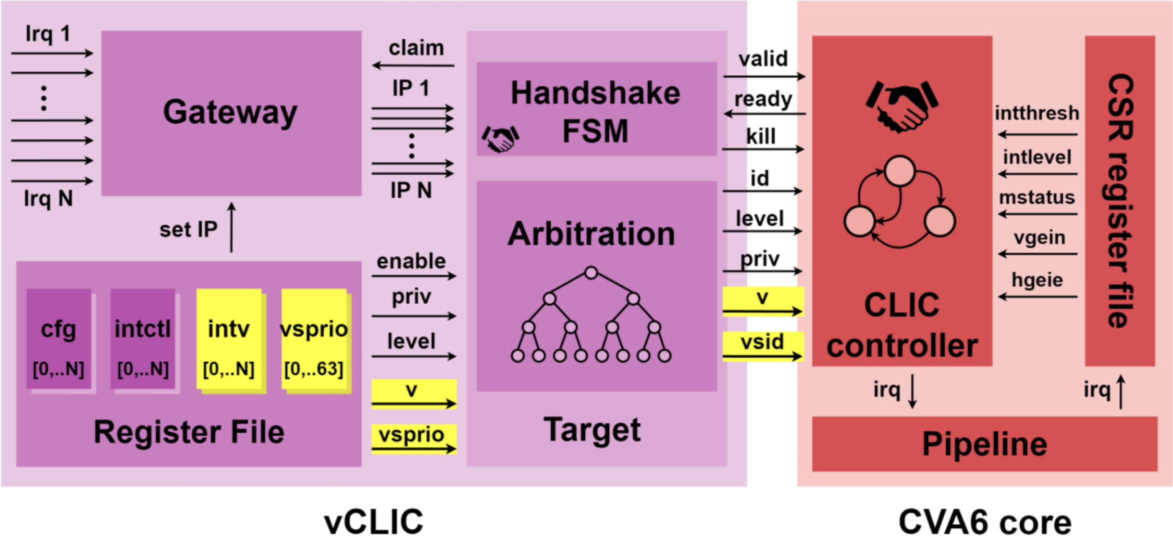
RTL available open-source



* implemented according to v0.7 of CLIC specification

vCLIC: design and architecture

- **Design principles**
  - **Modularity**
    - *VSCLIC*: direct injection of interrupts
    - *VSPRIO*: prioritization based on VM criticality
  - **Configurability**
    - Fine-grained tuning of resource utilization
  - **Compatibility**
    - Minimal deviation from CLIC spec*



- **System integration**
  - Coupled to CVA6 core
  - Integrated in Cheshire SoC
  - FPGA emulation for prototyping

* implemented according to v0.7 of CLIC specification

vCLIC: design and architecture

- **Design principles**
  - **Modularity**
    - *VSCLIC*: direct injection of interrupts
    - *VSPRIO*: prioritization based on VM criticality
  - **Configurability**
    - Fine-grained tuning of resource utilization
  - **Compatibility**
    - Minimal deviation from CLIC spec*

- **System integration**
  - Coupled to CVA6 core
  - Integrated in Cheshire SoC
  - FPGA emulation for prototyping

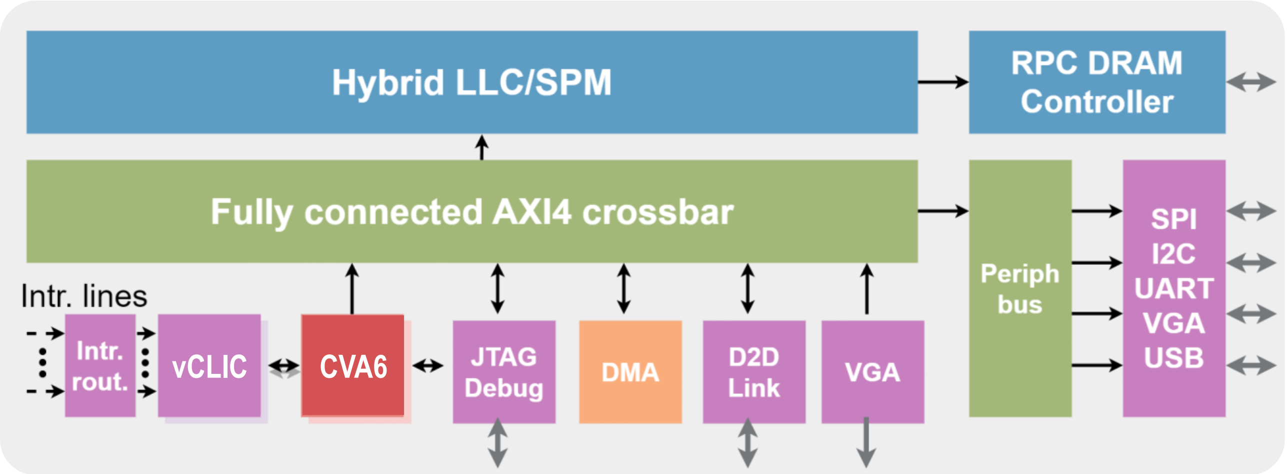
* implemented according to v0.7 of CLIC specification

vCLIC: design and architecture

- **Design principles**
  - **Modularity**
    - *VSCLIC*: direct injection of interrupts
    - *VSPRIO*: prioritization based on VM criticality
  - **Configurability**
    - Fine-grained tuning of resource utilization
  - **Compatibility**
    - Minimal deviation from CLIC spec*

- **System integration**
  - Coupled to CVA6 core
  - Integrated in Cheshire SoC
  - FPGA emulation for prototyping

Cheshire SoC



* implemented according to v0.7 of CLIC specification

Evaluation

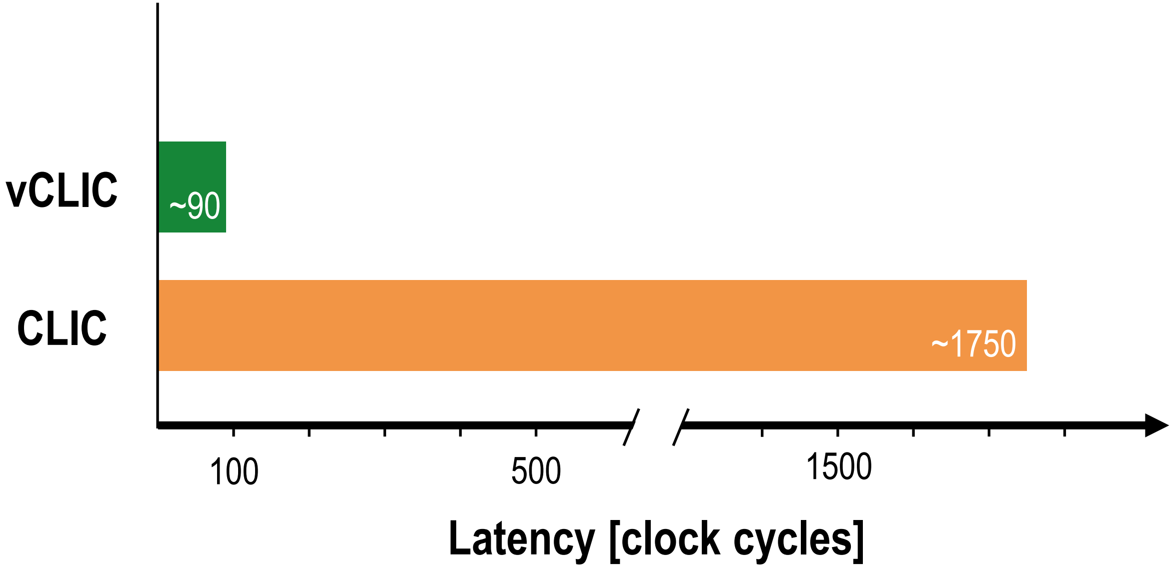
- **Software stack**
  - OpenSBI firmware
  - *Bao hypervisor* [1], *Xvisor* [2]
  - FreeRTOS, Linux

[1] Bao Hypervisor [2] Xvisor

[3] “Interrupting” the Status Quo (F. Marques et al.)

Evaluation

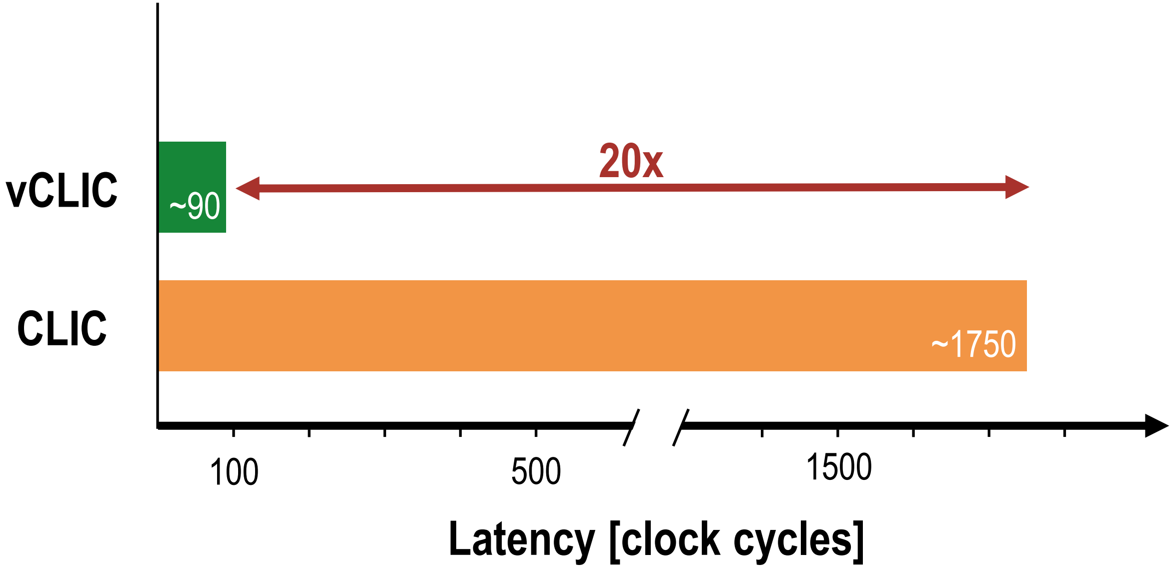
- **Software stack**
  - OpenSBI firmware
  - *Bao hypervisor* [1], *Xvisor* [2]
  - FreeRTOS, Linux
- **Interrupt latency**



[1] Bao Hypervisor [2] Xvisor
[3] "Interrupting" the Status Quo (F. Marques et al.)

Evaluation

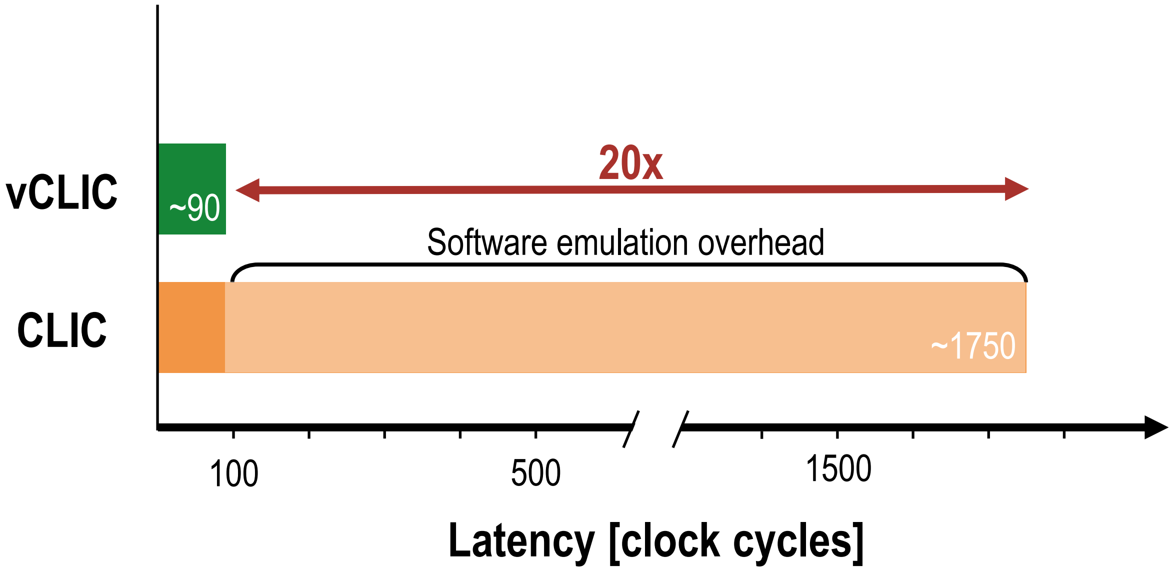
- **Software stack**
  - OpenSBI firmware
  - *Bao hypervisor* [1], *Xvisor* [2]
  - FreeRTOS, Linux
- **Interrupt latency**
  - **20x** reduced latency wrt software emulation



[1] Bao Hypervisor [2] Xvisor
[3] "Interrupting" the Status Quo (F. Marques et al.)

Evaluation

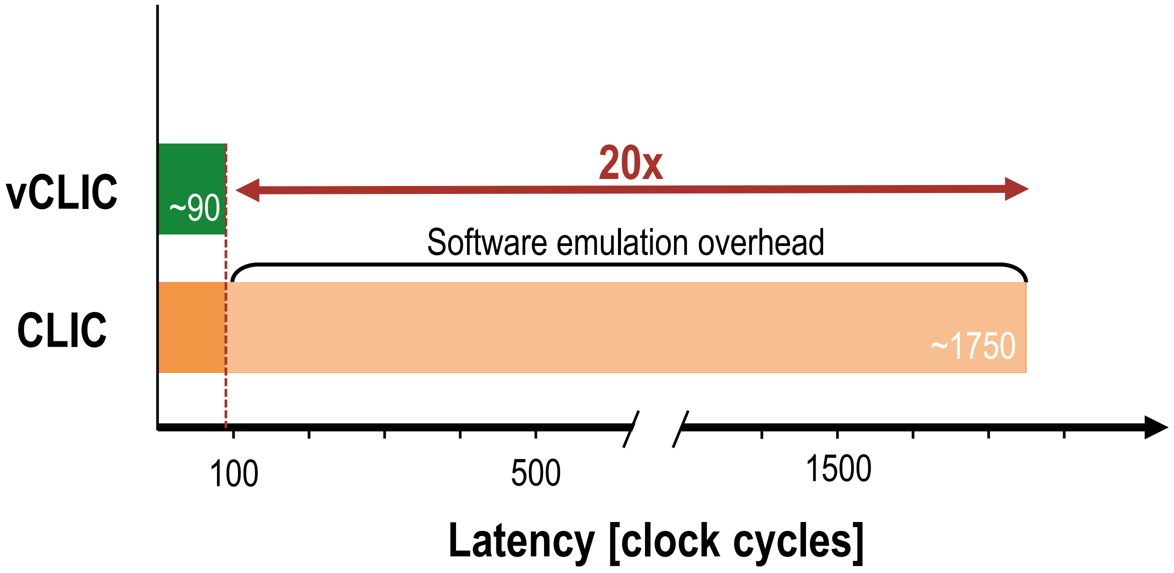
- **Software stack**
  - OpenSBI firmware
  - *Bao hypervisor* [1], *Xvisor* [2]
  - FreeRTOS, Linux
- **Interrupt latency**
  - **20x** reduced latency wrt software emulation
  - Same interrupt latency as non-virtualized system



[1] Bao Hypervisor [2] Xvisor
[3] "Interrupting" the Status Quo (F. Marques et al.)

Evaluation

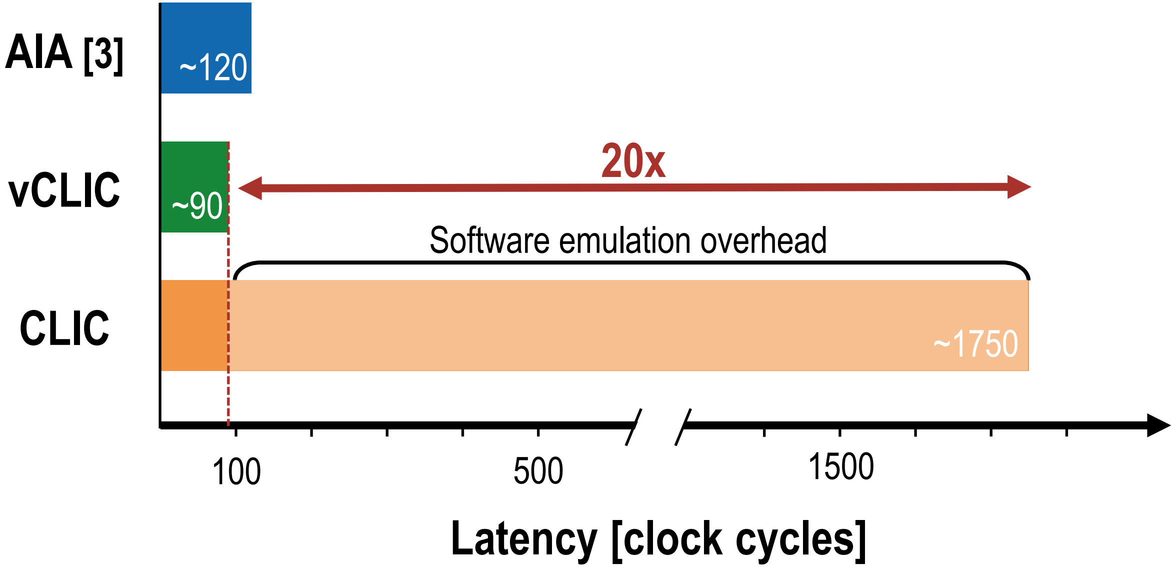
- **Software stack**
  - OpenSBI firmware
  - *Bao hypervisor* [1], *Xvisor* [2]
  - FreeRTOS, Linux
- **Interrupt latency**
  - **20x** reduced latency wrt software emulation
  - Same interrupt latency as non-virtualized system



[1] Bao Hypervisor [2] Xvisor
[3] "Interrupting" the Status Quo (F. Marques et al.)

Evaluation

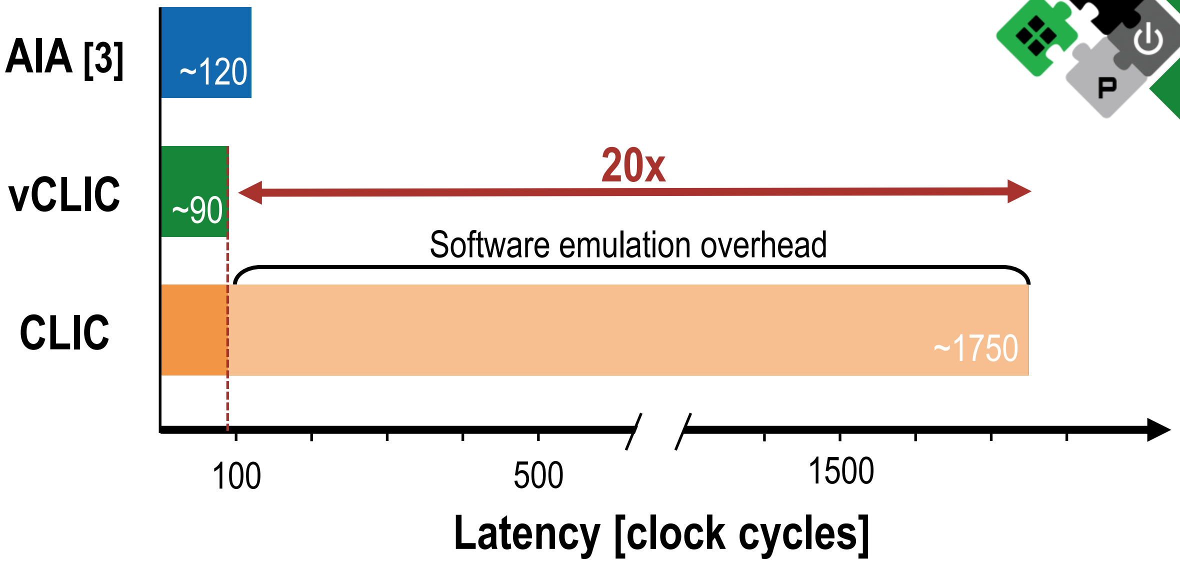
- **Software stack**
  - OpenSBI firmware
  - *Bao hypervisor* [1], *Xvisor* [2]
  - FreeRTOS, Linux
- **Interrupt latency**
  - **20x** reduced latency wrt software emulation
  - Same interrupt latency as non-virtualized system



[1] Bao Hypervisor [2] Xvisor
[3] “Interrupting” the Status Quo (F. Marques et al.)

Evaluation

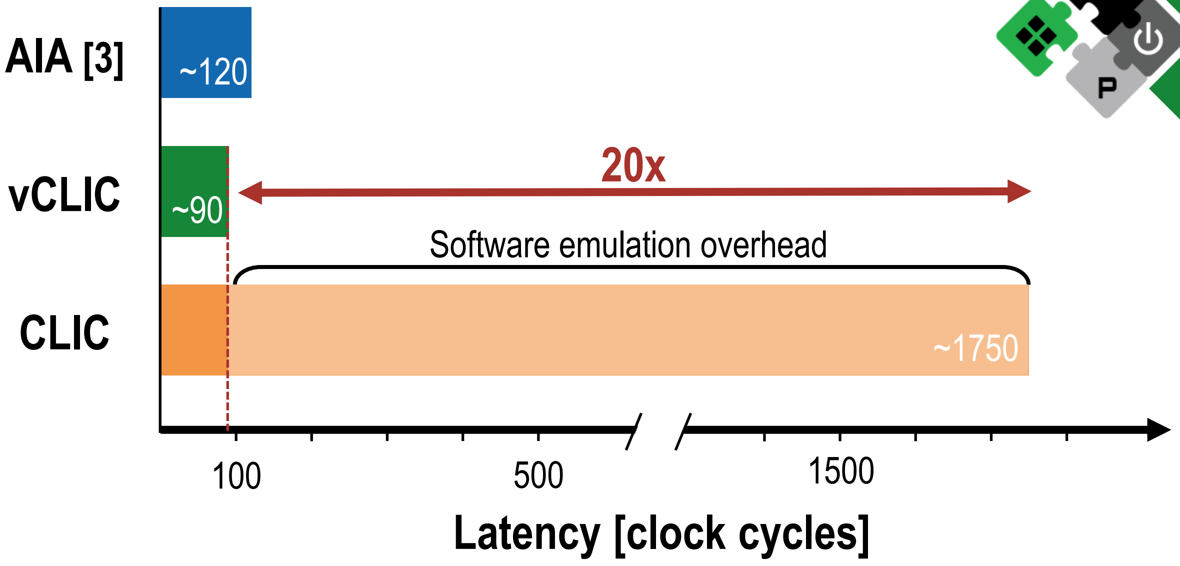
- **Software stack**
  - OpenSBI firmware
  - *Bao hypervisor* [1], *Xvisor* [2]
  - FreeRTOS, Linux
- **Interrupt latency**
  - **20x** reduced latency wrt software emulation
  - Same interrupt latency as non-virtualized system

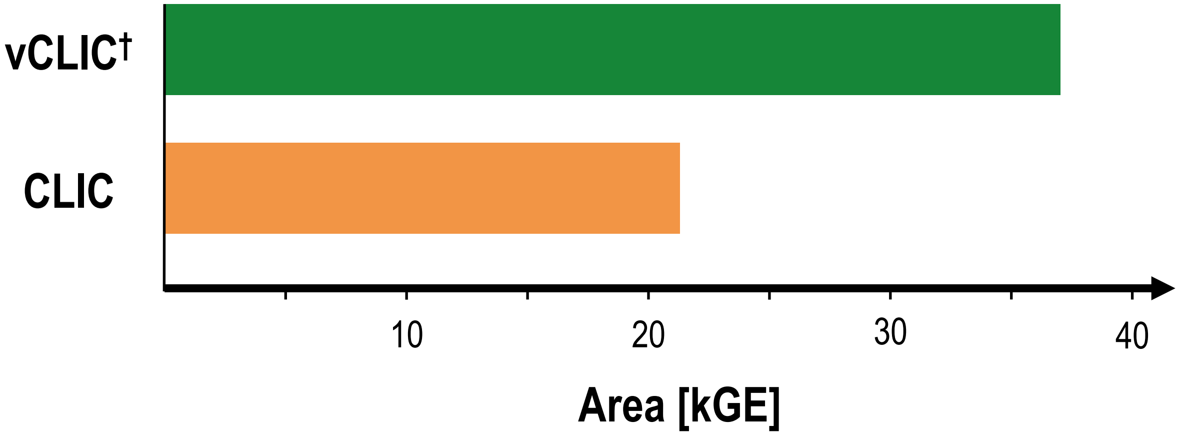


[1] Bao Hypervisor [2] Xvisor
[3] "Interrupting" the Status Quo (F. Marques et al.)

Evaluation

- **Software stack**
  - OpenSBI firmware
  - *Bao hypervisor* [1], *Xvisor* [2]
  - FreeRTOS, Linux
- **Interrupt latency**
  - **20x** reduced latency wrt software emulation
  - Same interrupt latency as non-virtualized system
- **Physical design***
  - Minimal area overhead (**<20kGE**)
  - Iso-frequency with existing CLIC (>1GHz)





* Implemented in 16nm Intel technology

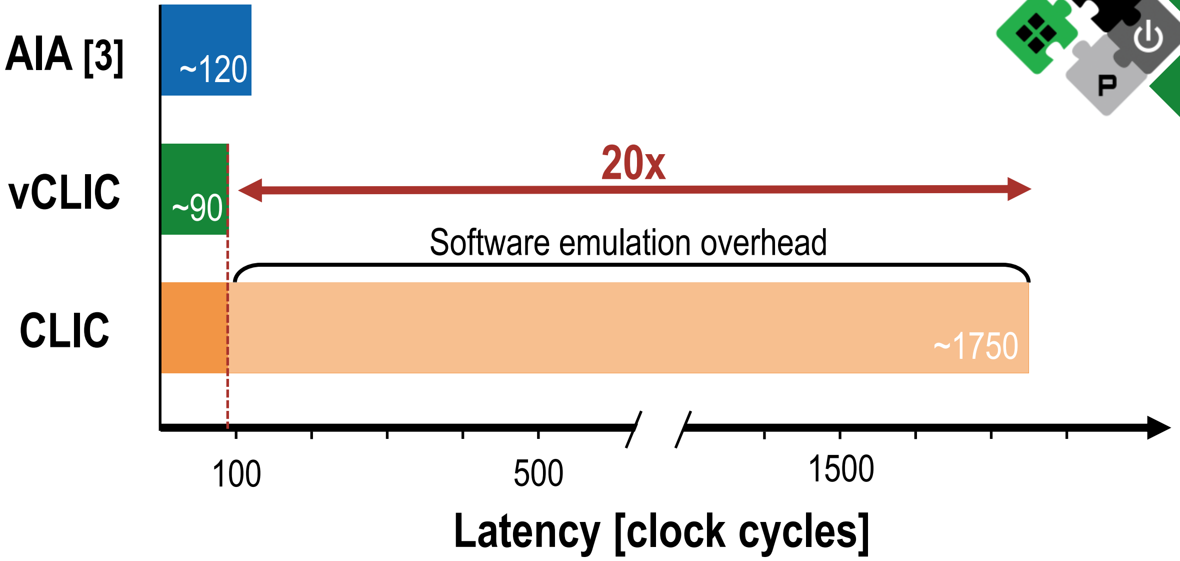
† Configuration: VSCLIC + VSPRIO bits = 1

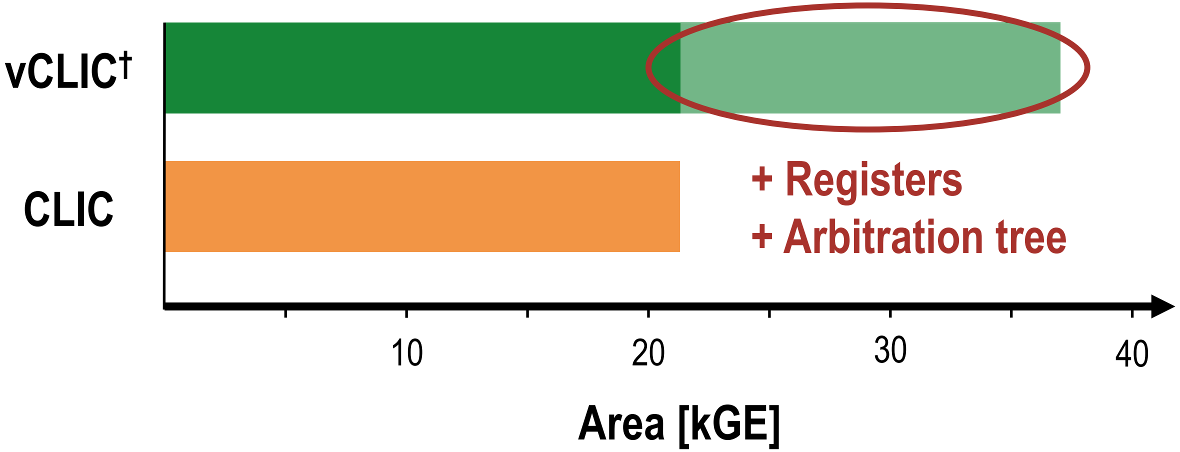
[1] Bao Hypervisor [2] Xvisor

[3] “Interrupting” the Status Quo (F. Marques et al.)

Evaluation

- **Software stack**
  - OpenSBI firmware
  - *Bao hypervisor* [1], *Xvisor* [2]
  - FreeRTOS, Linux
- **Interrupt latency**
  - **20x** reduced latency wrt software emulation
  - Same interrupt latency as non-virtualized system
- **Physical design***
  - Minimal area overhead (**<20kGE**)
  - Iso-frequency with existing CLIC (>1GHz)





* Implemented in 16nm Intel technology

† Configuration: VSCLIC + VSPRIO bits = 1

[1] Bao Hypervisor [2] Xvisor

[3] “Interrupting” the Status Quo (F. Marques et al.)

Conclusion

- **We propose vCLIC:**
  - Virtualization extension of CLIC
  - Up to **20x** reduced interrupt latency wrt software emulation
  - Limited area cost (<20kGE)
  - First RISC-V interrupt controller designed for fast-interrupt response in virtualized systems

Our work is open-source at:

https://github.com/pulp-platform/clic

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

Q&A

Enrico Zelioli	ezelioli@iis.ee.ethz.ch
Alessandro Ottaviano	aottaviano@iis.ee.ethz.ch
Robert Balas	balasr@iis.ee.ethz.ch
Nils Wistoff	nwistoff@iis.ee.ethz.ch
Angelo Garofalo	agarofalo@iis.ee.ethz.ch
Luca Benini	lbenini@iis.ee.ethz.ch

Institut für Integrierte Systeme – ETH Zürich
Gloriastrasse 35
Zürich, Switzerland

DEI – Universitá di Bologna
Viale del Risorgimento 2
Bologna, Italy

@pulp_platform
pulp-platform.org
youtube.com/pulp_platform