Quantifying the Latency and Possible Throughput of External Interrupts on Cyber-Physical Systems

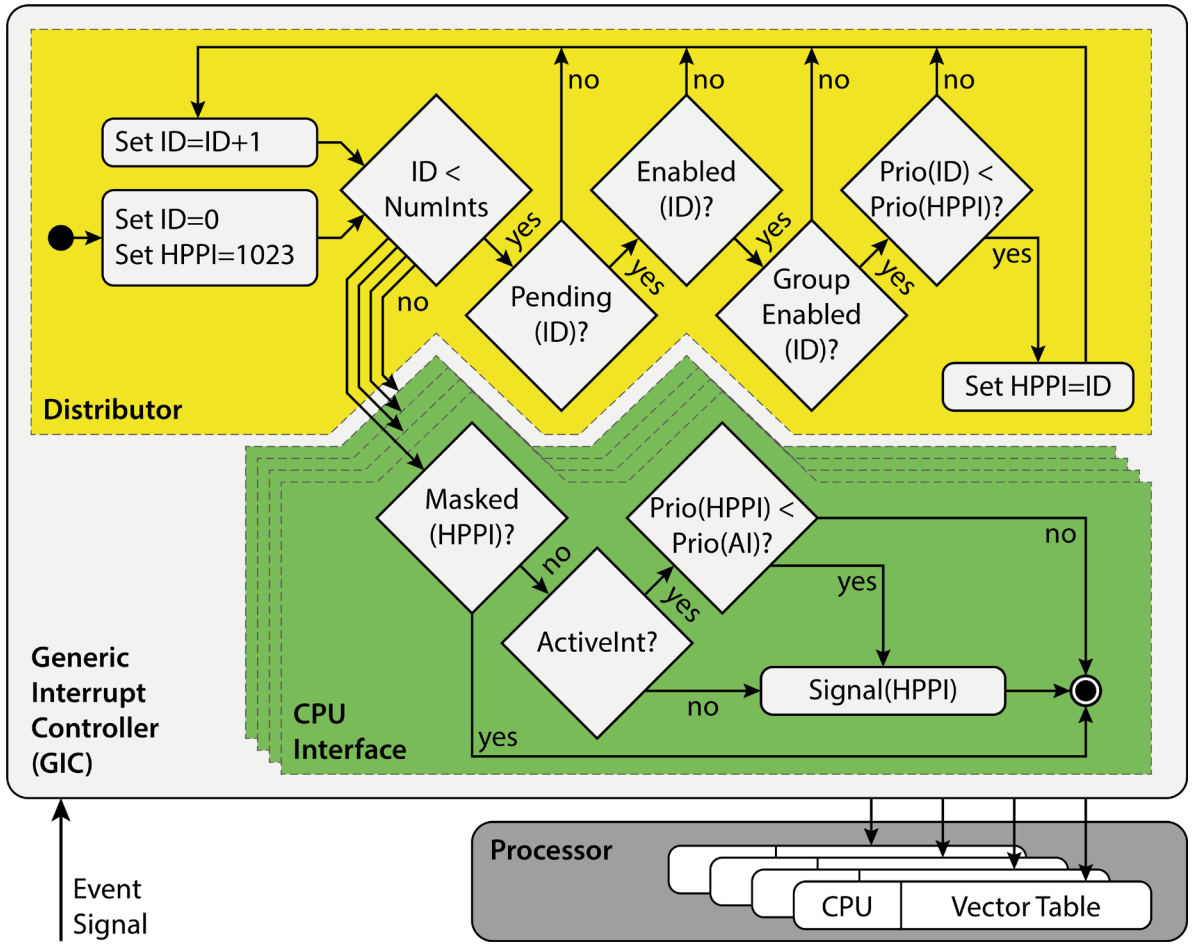
Oliver Horst, Johannes Wiesböck,
Raphael Wild, Uwe Baumgarten

Introduction

- Cyber-physical systems demand both, timely and predictable responses to physical events
- The interrupt handling process makes up a large part of this
- Previous studies focused on external measurements that are not applicable to modern, complex SoC, due to occurring inaccuracies
- Hence, we propose a flexible measurement procedure for ARMv8-A IP-core based platforms that provide a trace port
- We see a wide applicability of our measurement procedure, due to the increased market share of ARM and the wide adoption of ARM-based systems in the automotive industry (e.g., Mercedes MBUX, Continental Body Computers, and Tesla Autonomous Driving Platform)
- Precise measurements on the interrupt performance are beneficiary for the design of virtualization solutions, operating system, and runtime environments for time-critical systems

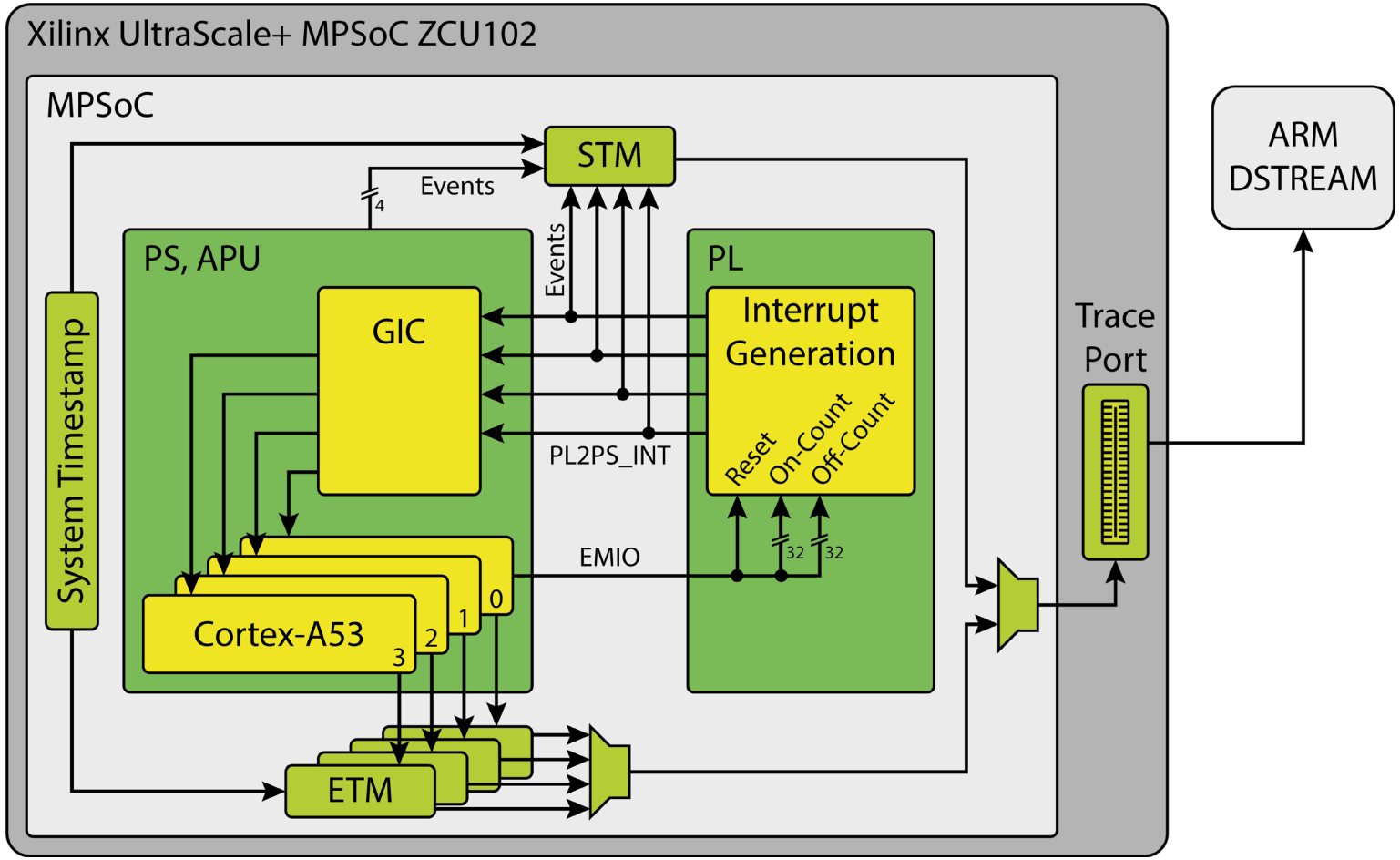
ARMv8-A Interrupt Handling Procedure

- 4 types of interrupts:
  - Peripheral
  - Software generated
  - Virtual
  - Maintenance
- 2 processing schemes:
  - Regular (IRQ)
  - Fast (FIQ)
- We focus on peripheral interrupts, triggered by an external stimuli, processed in the regular way (IRQ)



Measurement Setup

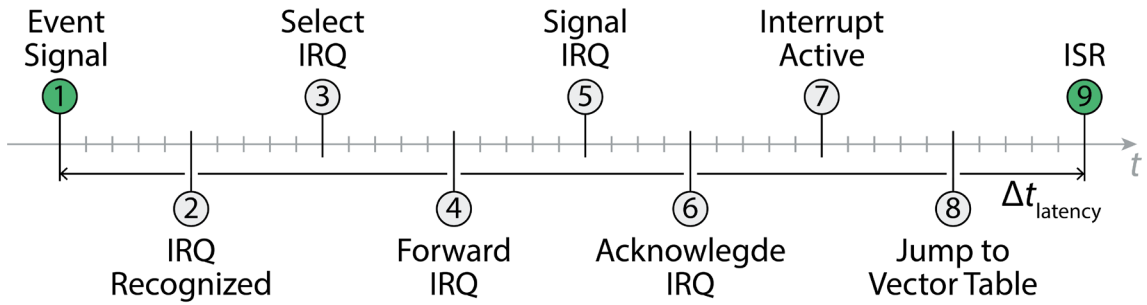
- Xilinx ZCU102 evaluation board with
  4x A53-, 2x R5-cores, and an on-chip FPGA
- ARM DSTREAM hardware trace
- Verilog based interrupt generation block controlled by the APU
- ARM CoreSight ETM and STM based measurements provide a common time-line for internal and external events



Measurement Procedure

- Measurements on 2 platforms
  - Baremetal (unoptimized)
  - FreeRTOS
- Based on the open-source build- and test platform *toki* with the same low-level interrupt dispatching routine provided by Xilinx
- Easily extensible to Linux based systems

- Measurements on APU core 0, stimulation by all cores
- Designed for and tested on GICv2 but applicable to v3 and v4 as well



- Latency Measurements
  - 1ms interrupt trigger, 4ms pause
  - 30s trace capture (i.e., 2.400 stimulation phases)
- Throughput Measurements
  - 9.75s interrupt trigger, 250ms pause
  - 120s trace capture (i.e., 20 stimulation phases)

Precision, Limitations & Benefits

Precision

- Measurement precision is bounded by the timestamp clock, configured to 250 MHz
  - ⇨ 4ns time resolution
  - ⇨ An order of magnitude below the measured latency

- Modern oscilloscopes can reach up to 20 GSa/s, i.e., a 0.05ns time resolution, but their measurements are imprecise

Limitations

- Only applicable to ARM based systems with JTAG- and trace-port
  - ARM IP designs have a 40% market share
  - Wide availability of evaluation boards

Benefits

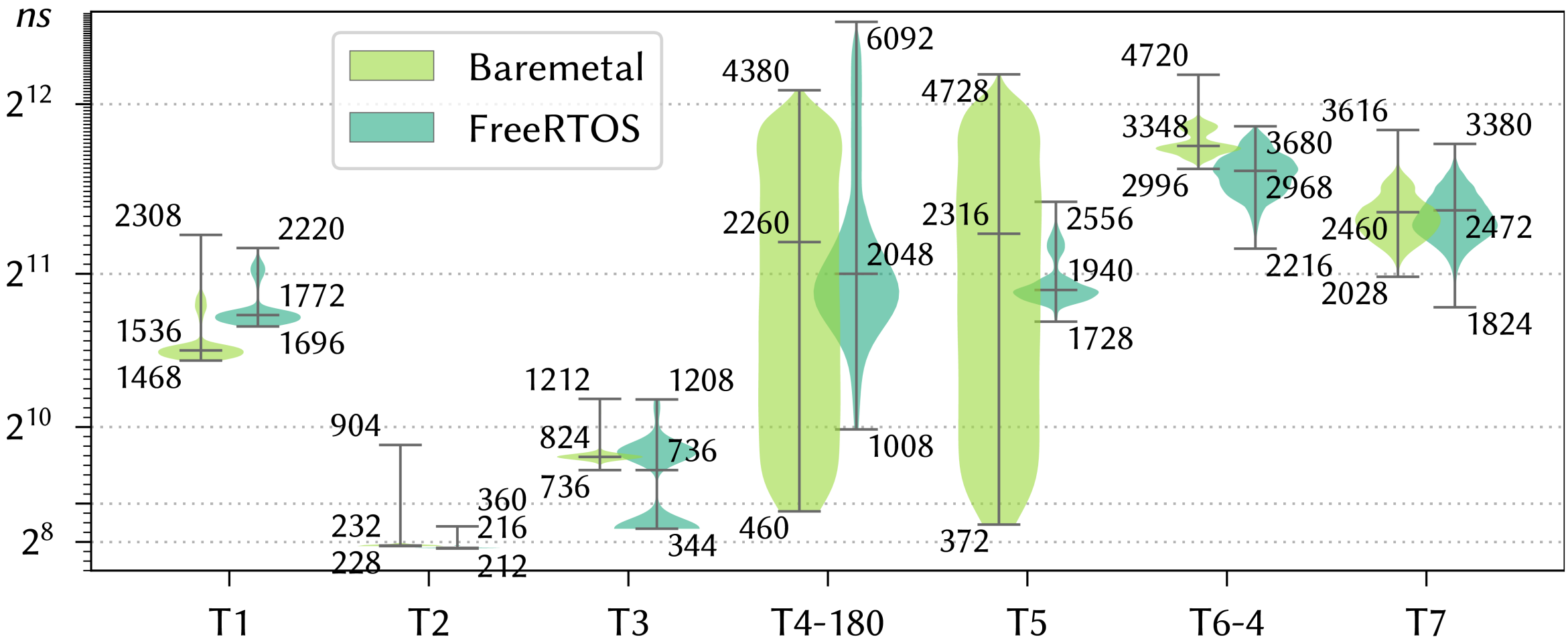
- No in-depth knowledge about the hardware platform needed
- Measurement procedure applicable to other software stacks as well

Analyzed Test-cases

Description	Targeted Comp.	Measurements		Enabled Interrupts	Cache Config	Enabled Cores	Benchmarks		
		L	T				L_{min}	L_{max}	T_{max}
T1: Baseline	-	X	X	1	Disabled	1			
T2: Caches Enabled	Cache	X	X	1	Enabled	1	X		X
T3: Caches Invalid.	Cache	X		1	Invalidated	1			
T4: Enabled Int.	GIC	X		2-181	Disabled	2		X	
T5: Order of Prio.	GIC	X		15	Disabled	2			
T6: Parallel Int. Hdl.	GIC	X	X	1	Disabled	2, 3, 4		X	X
T7: Rnd Mem Access	Memory	X		1	Disabled	4			

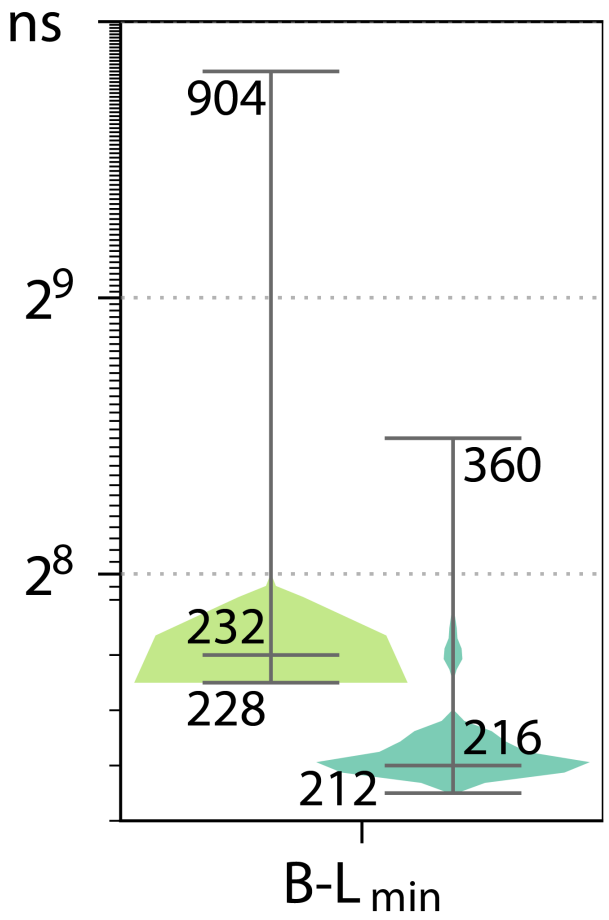
Measurement Results

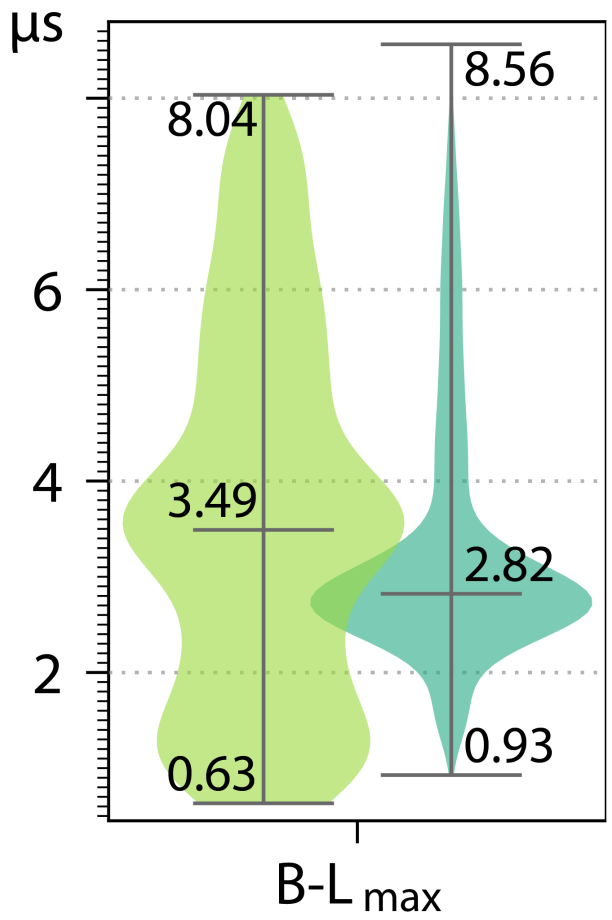
Latency Measurements (1/2)



Measurement Results

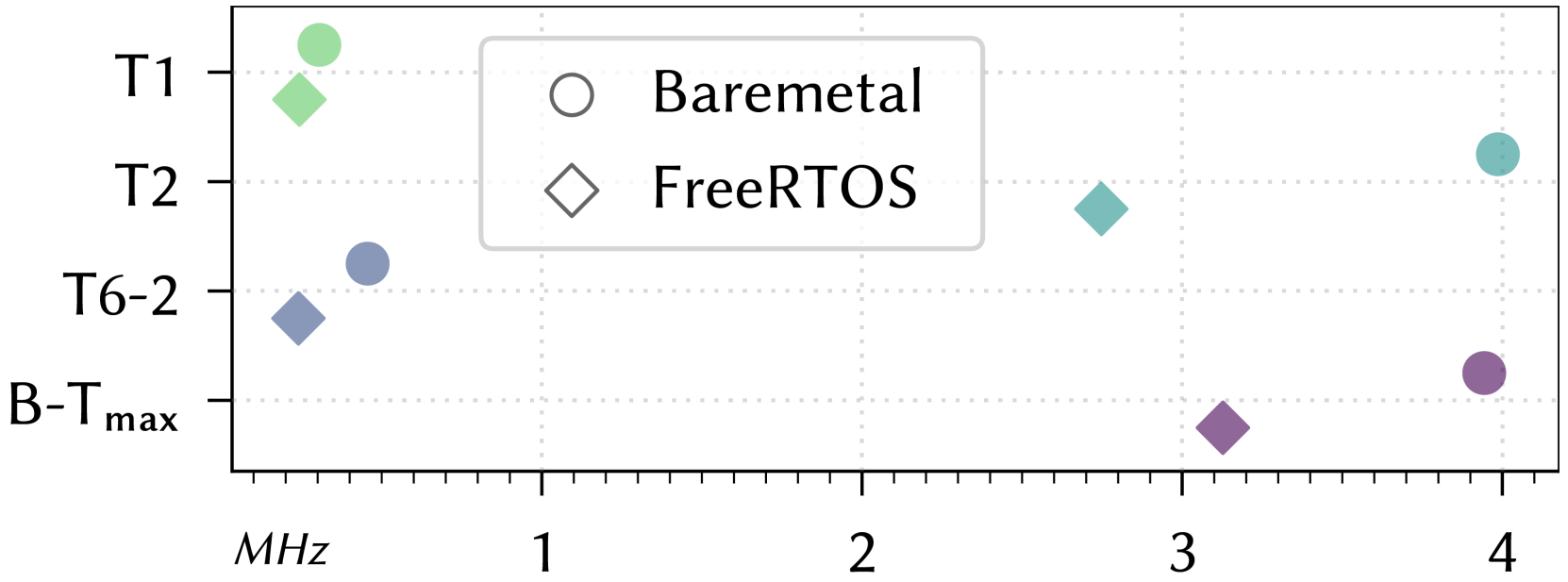
Latency Measurements (2/2)





Measurement Results

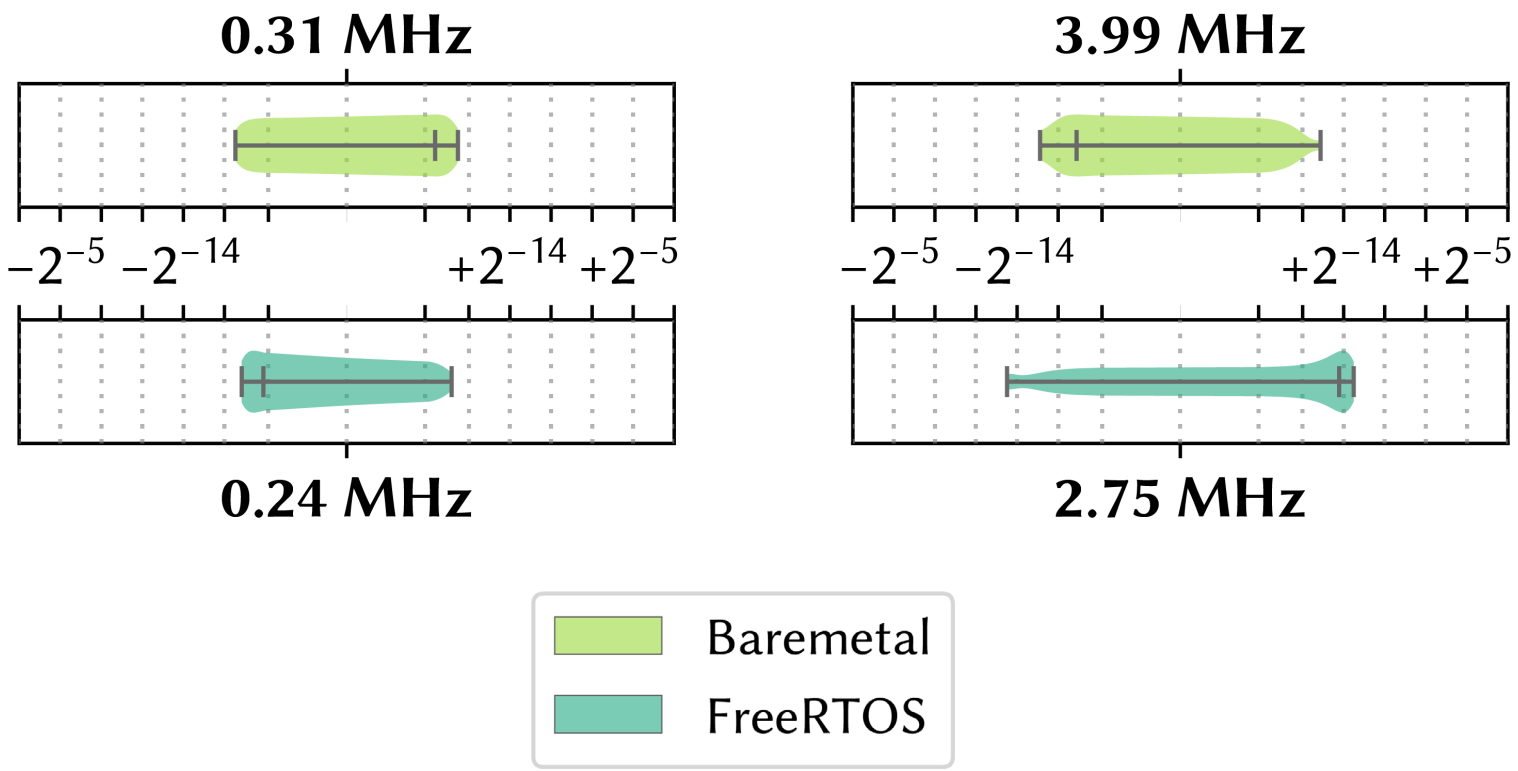
Throughput Measurements (1/3)



Measurement Results

Throughput Measurements (2/3)

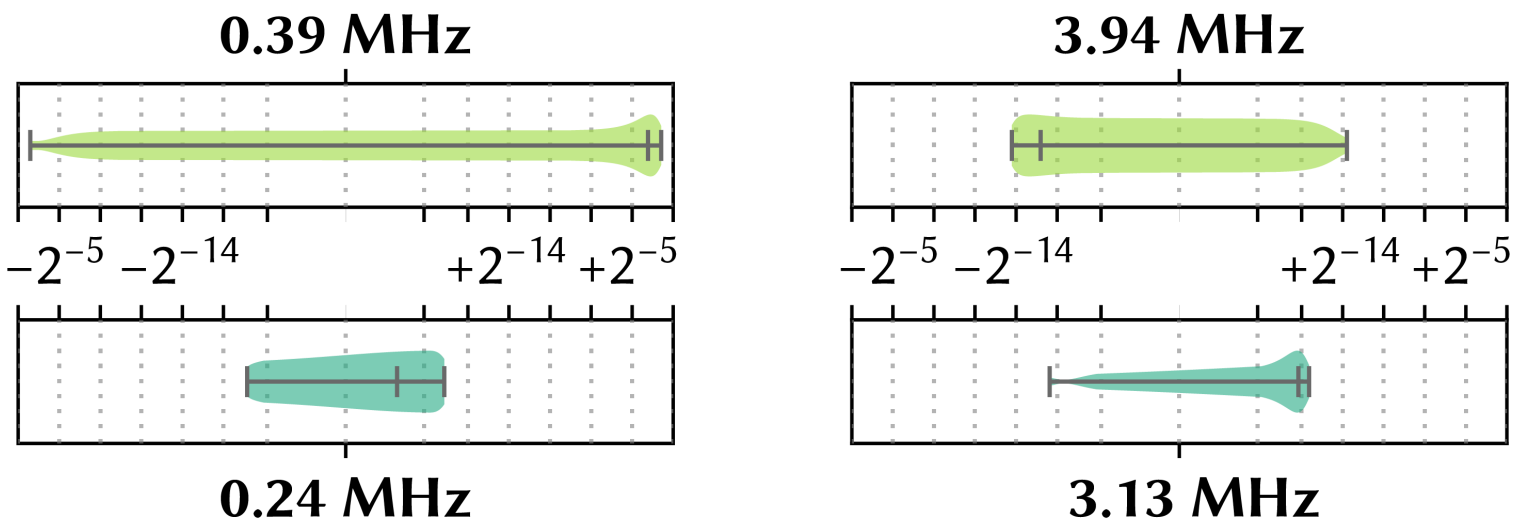
- T1
- T2



Measurement Results

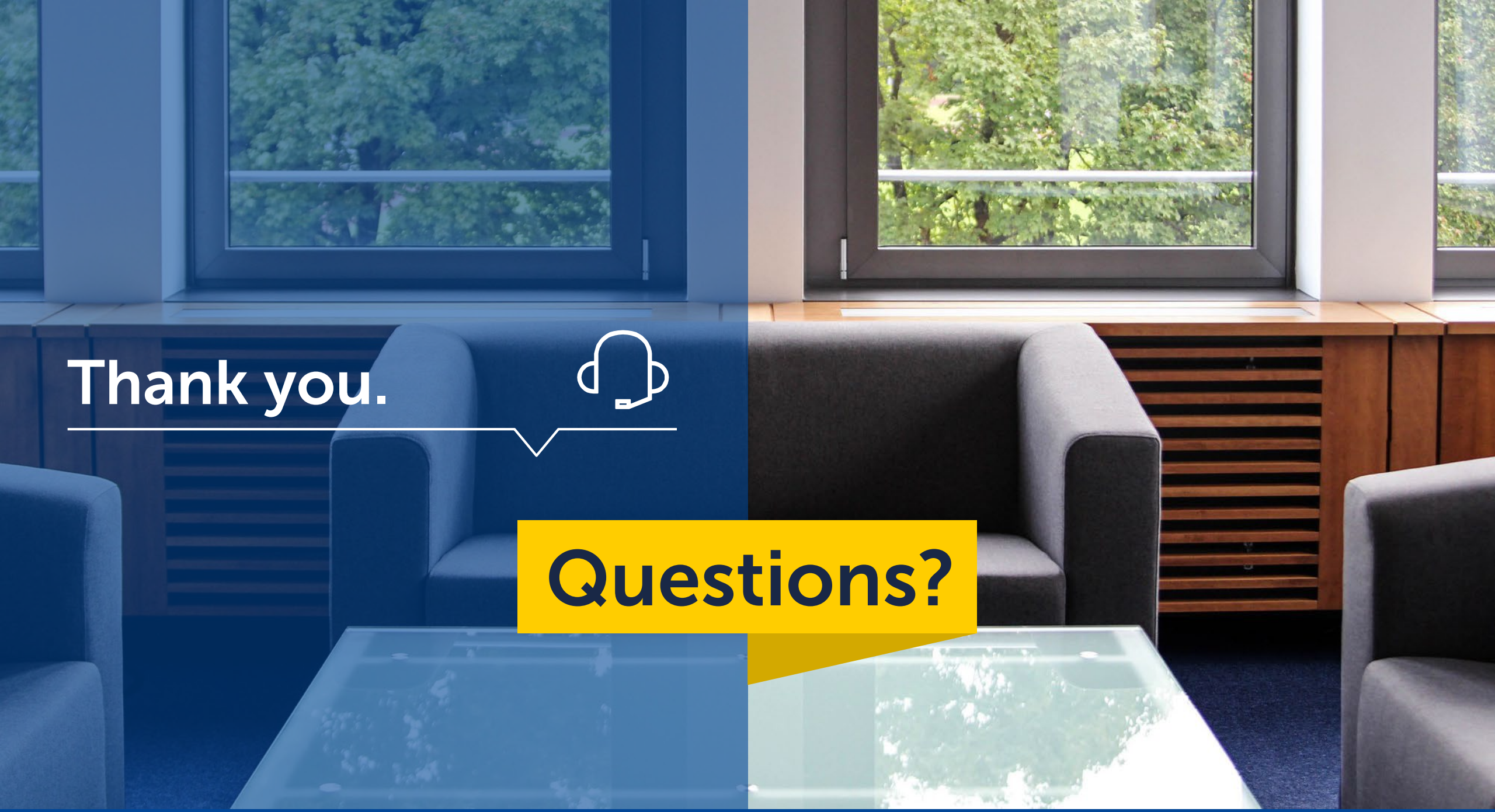
Throughput Measurement (3/3)

- T6-2
- B-T_{max}



Conclusion & Outlook

- Precise method to measure the interrupt performance of complex ARM based SoCs without expert knowledge
- Set of benchmark functions that provoke the best and worst interrupt latency and maximal throughput
- Evaluation platform and test-cases are available open-source, check it out under https://git.fortiss.org/toki
- Software design can have large effects on the predictability of the interrupt latency and throughput
- Partially unexpected results require further investigation, we see two options:
  - Implementing a fully optimized, assembly-only bare-metal stack
  - Analyzing the hardware effects with a cycle accurate simulator


Thank you.
Questions?

Contact

Oliver Horst

fortiss GmbH
Guerickestr. 25 • 80805 Munich • GERMANY

horst@fortiss.org
www.fortiss.org

tel +49 89 3603522 15
fax +49 89 3603522 50

fortiss