

A Power Model of System Clocks to Minimize Energy Consumption on Constrained MCUs

Michel Rottleuthner* michel.rottleuthner@haw-hamburg.de

Thomas C. Schmidt* t.schmidt@haw-hamburg.de

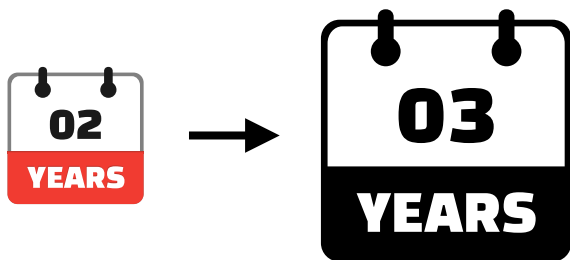
*Hamburg University of Applied Sciences

Matthias Wählich† m.waehlich@tu-dresden.de

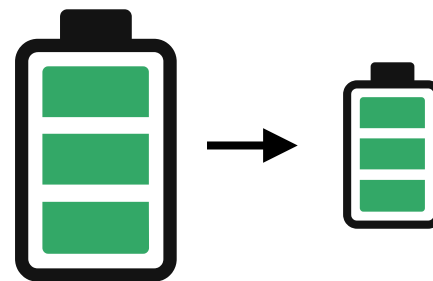
†TU Dresden and Barkhausen Institut



Motivation: Minimize Energy

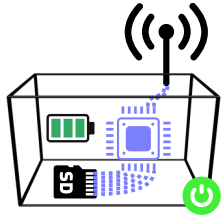


Increase lifetime.



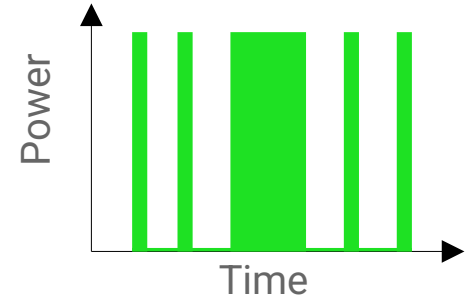
Reduce battery size.

Minimizing Energy on Constrained Devices



Duty Cycling

- Put system to sleep while idle
- Wake up via timers or interrupts
- + Very effective, very easy
- Minimizes energy only during idle phase

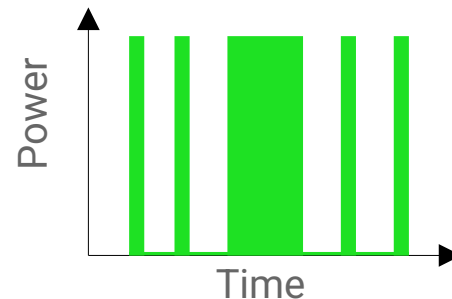


Minimizing Energy on Constrained Devices



Duty Cycling

- Put system to sleep while idle
- Wake up via timers or interrupts
- + Very effective, very easy
- Minimizes energy only during idle phase

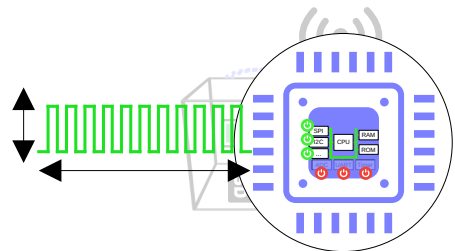
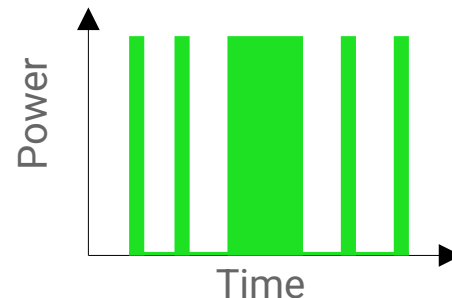


Minimizing Energy on Constrained Devices



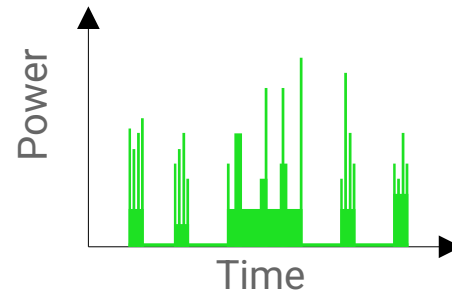
Duty Cycling

- Put system to sleep while idle
- Wake up via timers or interrupts
- + Very effective, very easy
- Minimizes energy only during idle phase

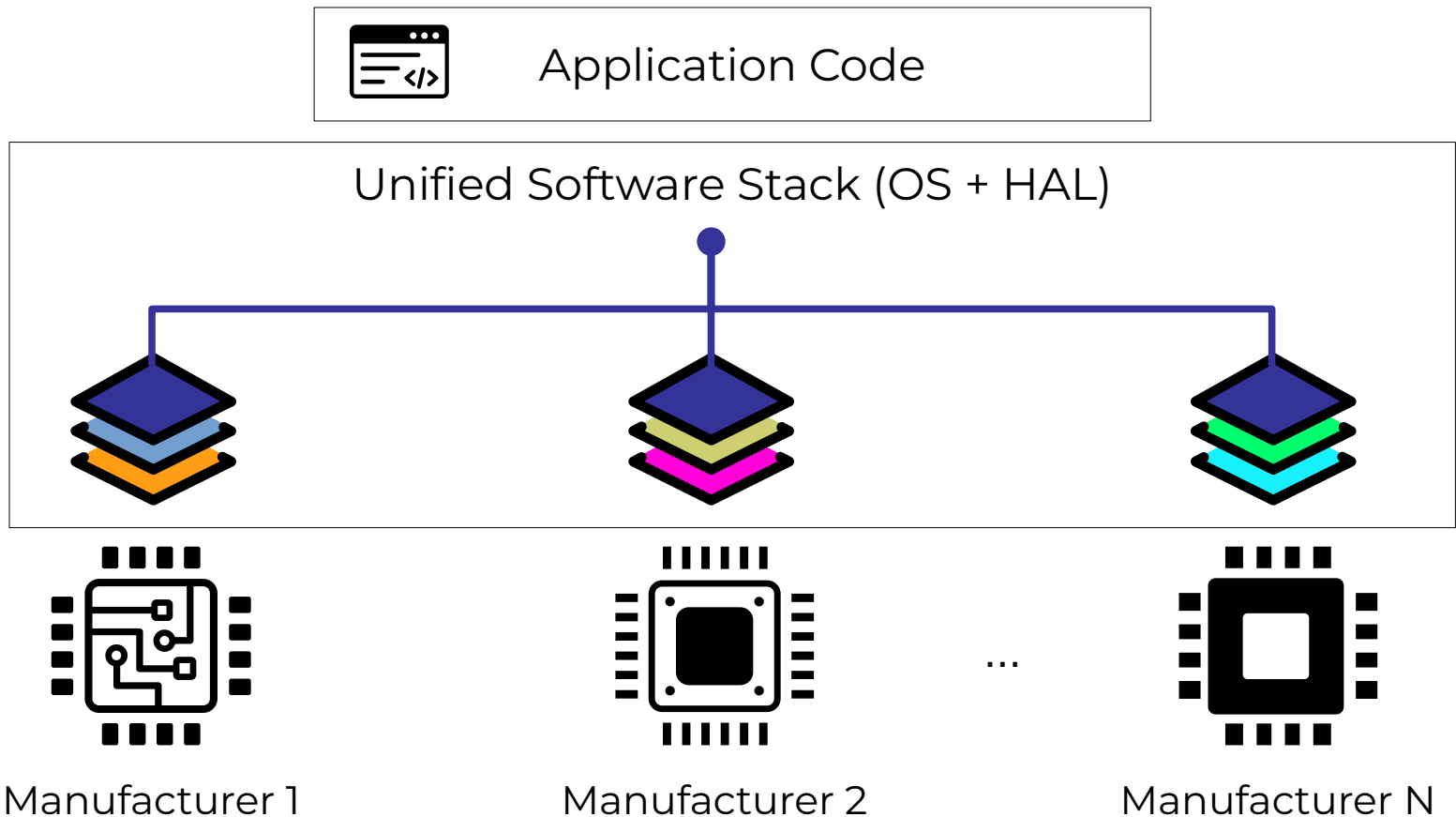


Dynamic Voltage and Frequency Scaling

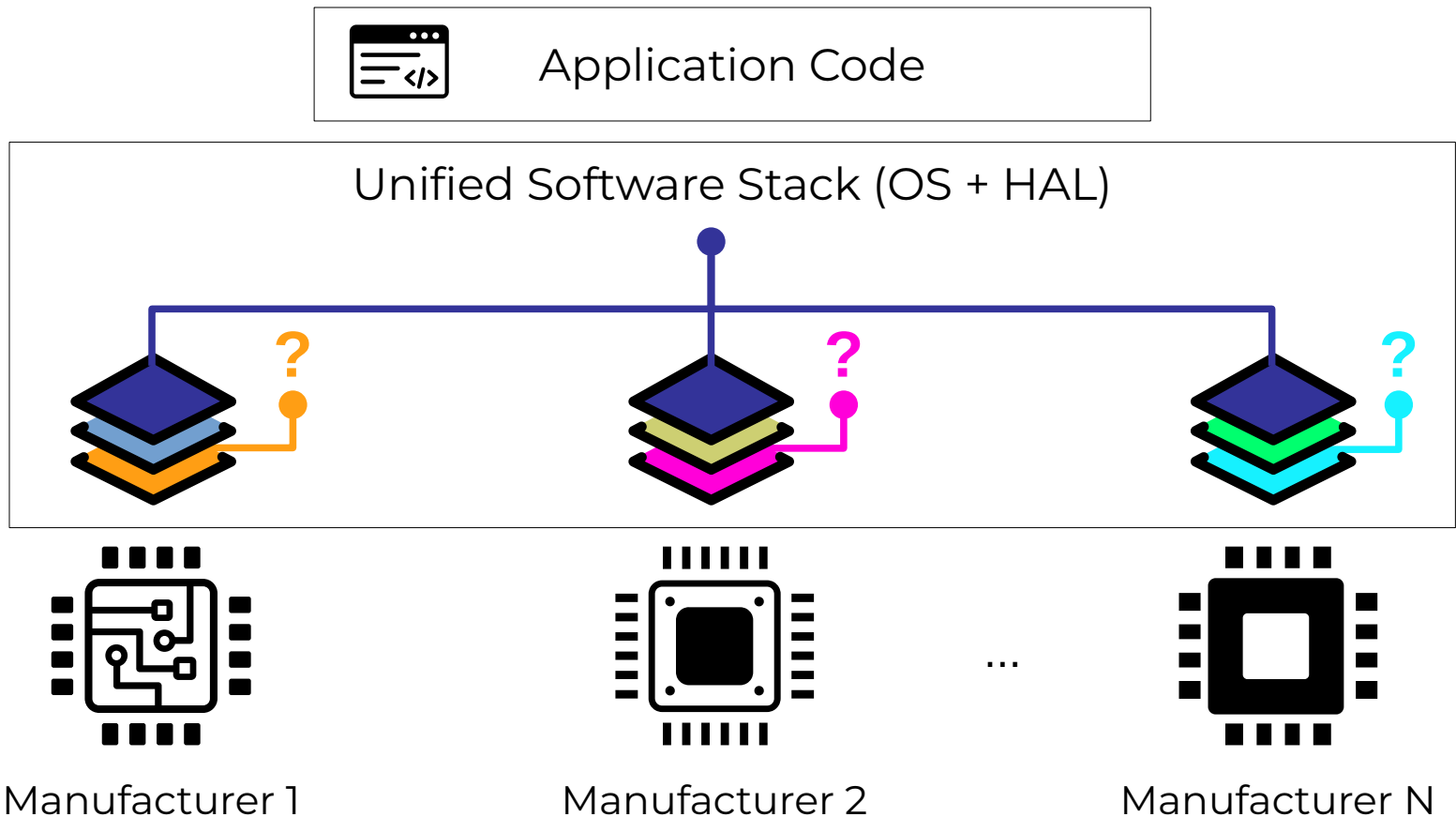
- Fine-grained performance control
- + Minimizes energy during active phase
- Very complex
- Requires detailed hardware knowledge
- Hard to generalize



Platform Agnostic Programming

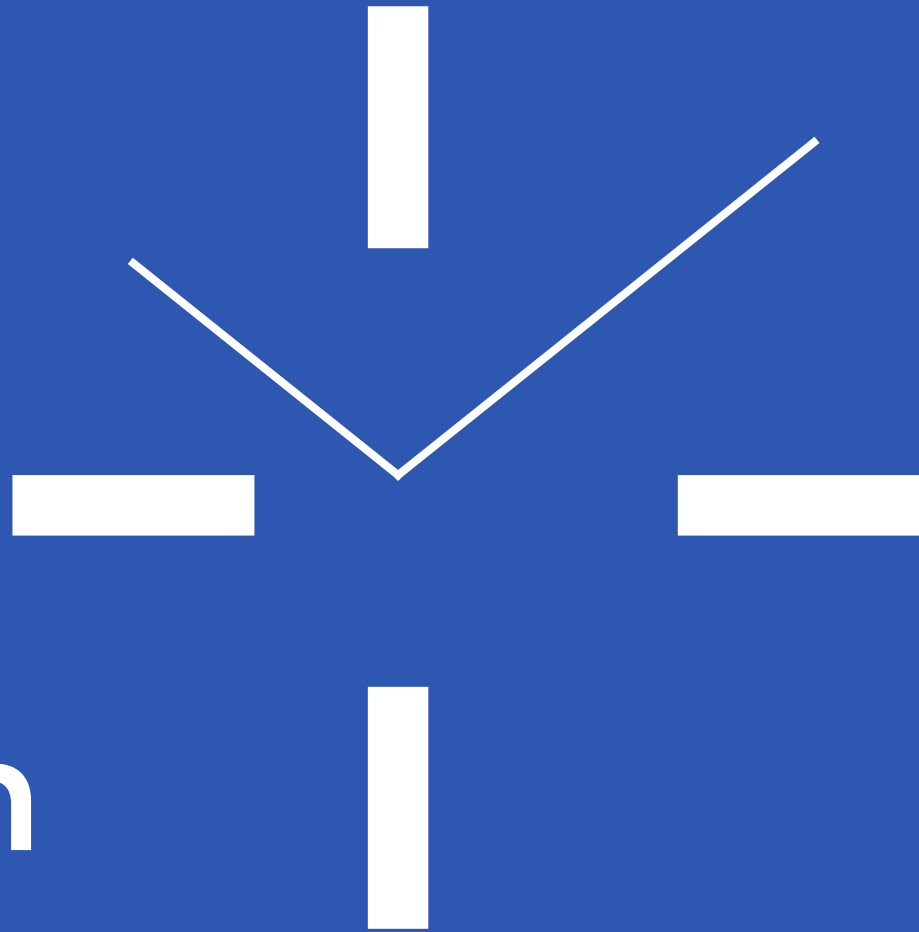


Platform Agnostic Programming

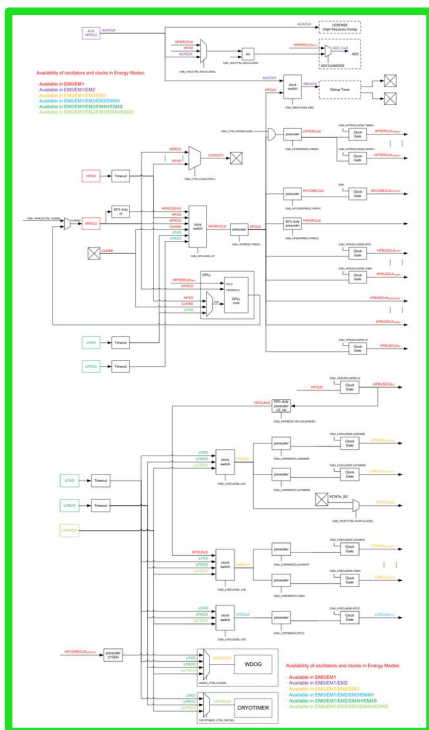


- Motivation and Background
- Clock Configuration Abstraction
- Clock Tree Power Modeling
- Evaluation
- Conclusion and Outlook

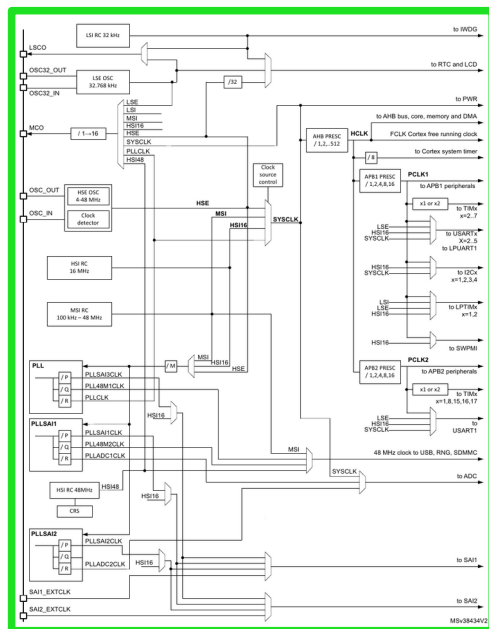
The Clock Configuration Abstraction



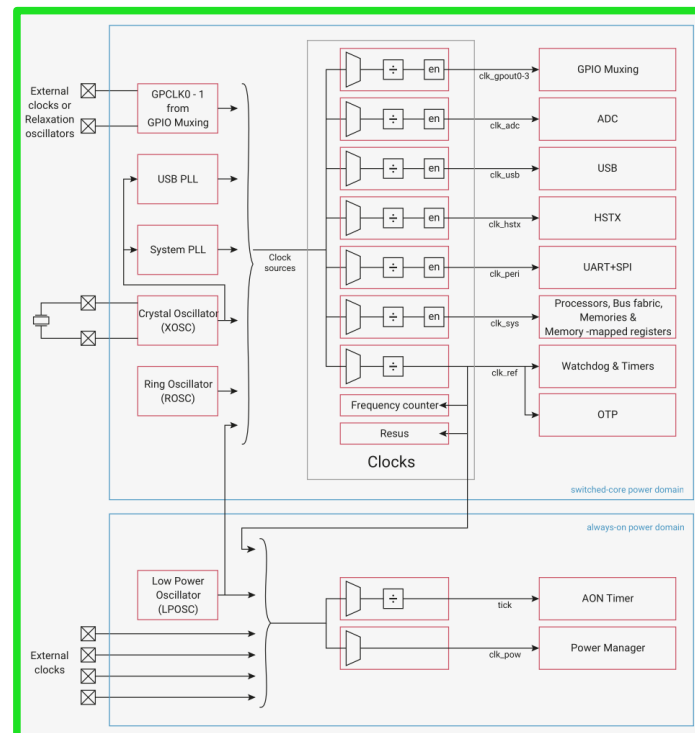
Clock Tree Challenges



EFM32PG12B

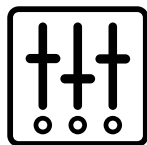


STM32L476RG



RP2350

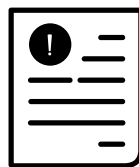
Clock Tree Challenges



Abstractions for
low-level clock
settings
on constrained
devices.

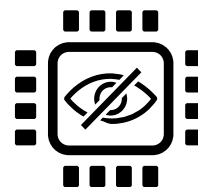
Previous work

EFM32PG12B



Datasheets do not
cover the entire
clock configuration
space.

STM32L476RG

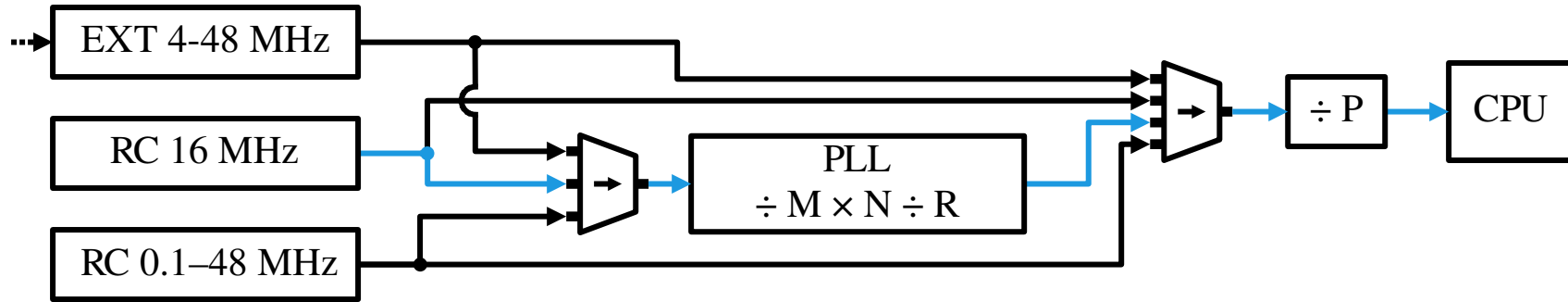


Physical properties
of proprietary
hardware designs
are not disclosed.

RP2350

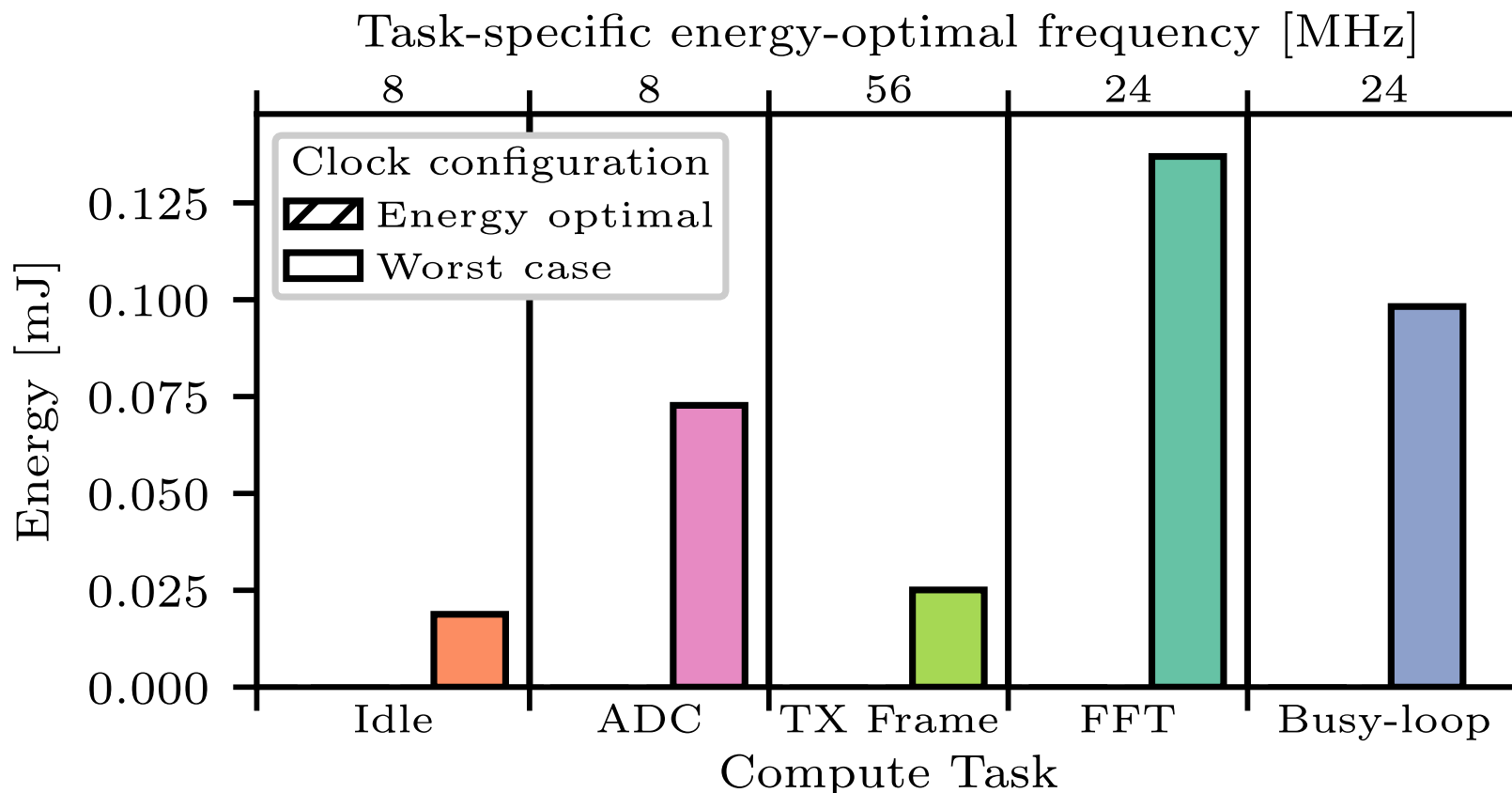
Our contribution fixes this

The Clock Configuration Space is Large

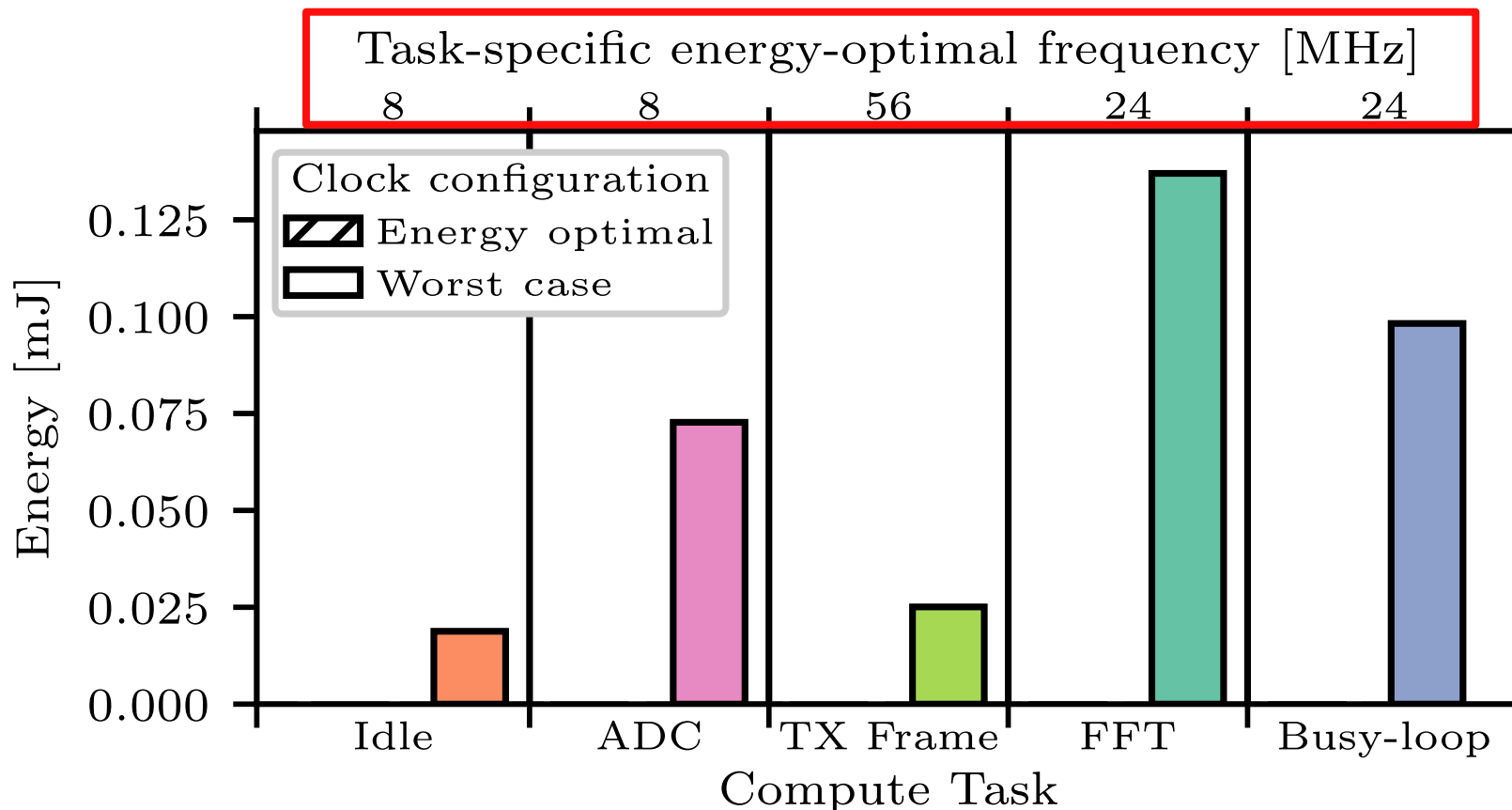


- 3 different oscillators
- 6 different topologies for clocking the CPU
- 5 scale factor settings
- Tens to thousands of configurations for generating the same frequency

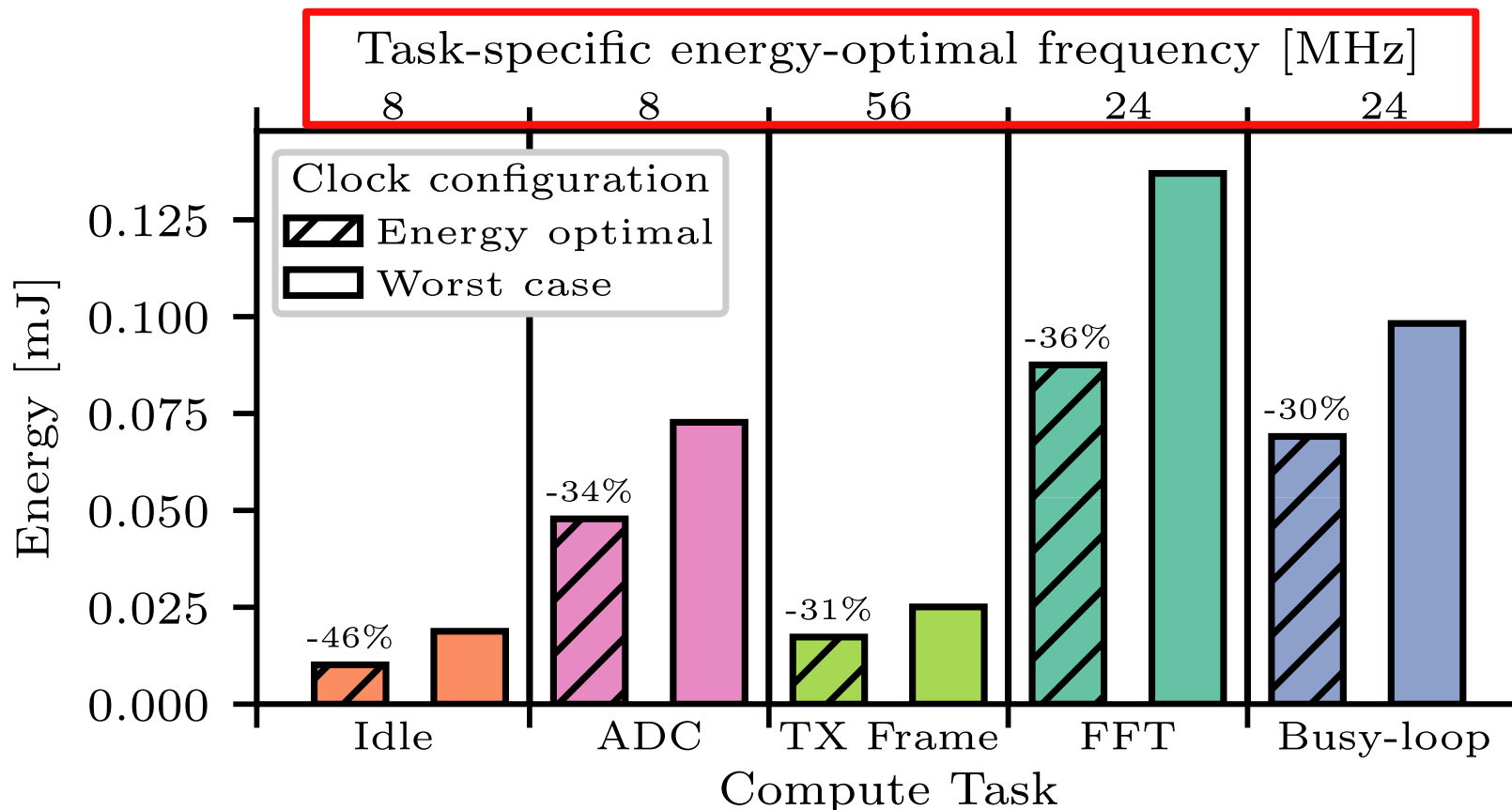
Same Frequency != Same Power



Same Frequency != Same Power



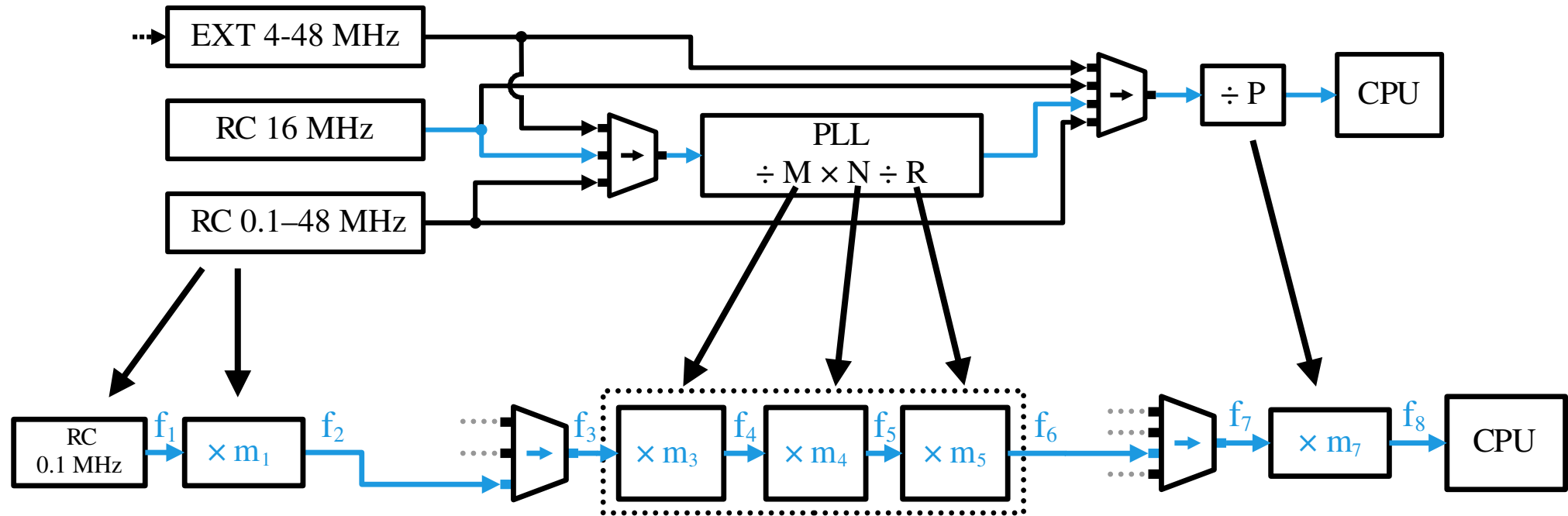
Same Frequency != Same Power



Clock-Tree Power Modeling

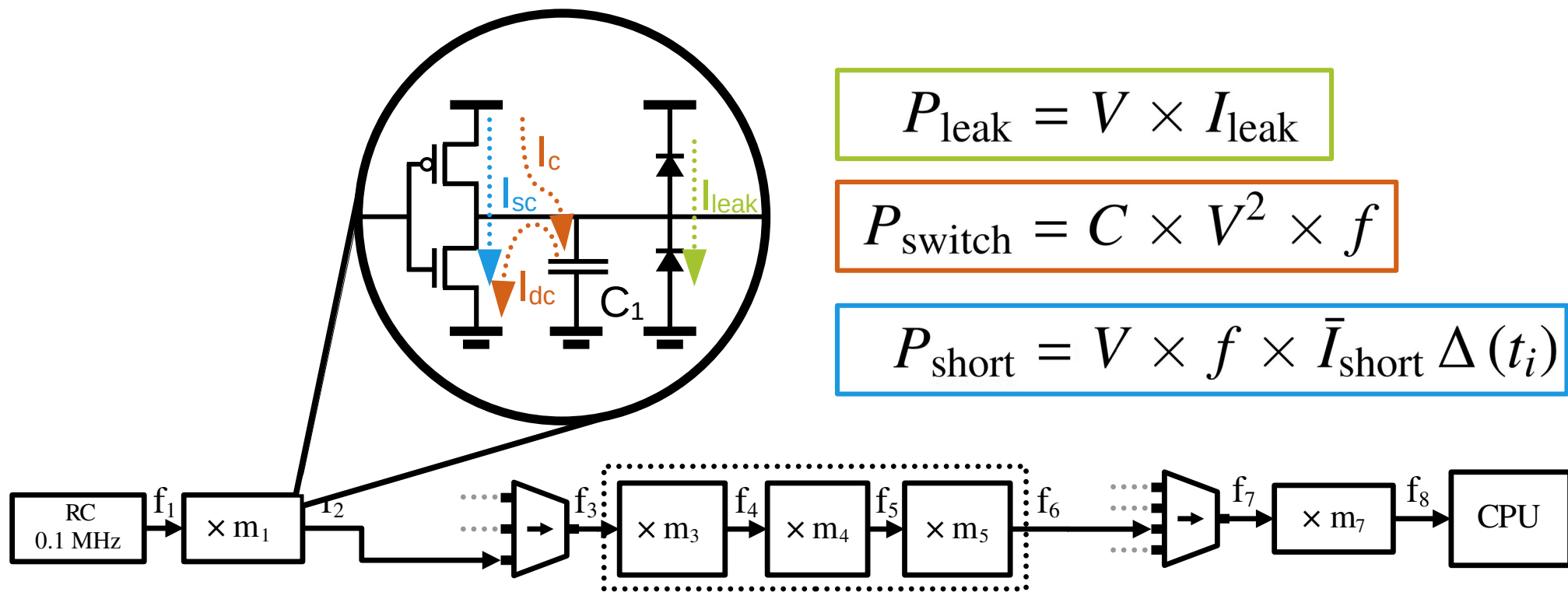


Clock-Tree Power Model: The Idea



Map all scale factors to separate node

Clock-Tree Power Model: The Idea



$$P_{\text{leak}} = V \times I_{\text{leak}}$$

$$P_{\text{switch}} = C \times V^2 \times f$$

$$P_{\text{short}} = V \times f \times \bar{I}_{\text{short}} \Delta(t_i)$$

Assign consumption terms to surrogate CMOS circuit

Modeling the Full Clock-Tree

$$\begin{aligned} P_{\text{sys-clock}}(\mathcal{N}) &= P_{\text{leak}}(\mathcal{N}) + P_{\text{switch}}(\mathcal{N}) + P_{\text{short}}(\mathcal{N}) \\ &= \sum_{i=1}^k \{C_{\Gamma(n_i)} \times V^2 \times f_{\Gamma(n_i)} + V \times f_{\Gamma(n_i)} \times \bar{I}_{\text{short}}(\Gamma(n_i)) \Delta(t_i) + V \times I_{\text{leak}}(\Gamma(n_i))\} \\ &= \sum_{i=1}^k V^2 \{C_{\Gamma(n_i)} \times f_{\Gamma(n_i)}\} + V \{f_{\Gamma(n_i)} \times \bar{I}_{\text{short}}(\Gamma(n_i)) \Delta(t_i) + I_{\text{leak}}(\Gamma(n_i))\} \end{aligned}$$

$$P_{\text{sys}}(\mathcal{N}) = P_{\text{sys-stat}}(V) + P_{\text{sys-clock}}(\mathcal{N}) + P_{\text{sys-task}}(V, f)$$

$$\mathbf{min} \text{ SSE} = \sum_{\Gamma} (P_{\text{mod}}(\Gamma) - P_{\text{emp}}(\Gamma))^2$$

$$\mathbf{s.t.} \ C_{\Gamma(n_i)} \geq 0$$

$$I_{\text{leak}}(\Gamma(n_i)) \geq 0 \quad \text{and} \quad \bar{I}_{\text{short}}(\Gamma(n_i)) \geq 0$$

Modeling the Full Clock-Tree

$$P_{\text{sys-clock}}(\mathcal{N}) = P_{\text{leak}}(\mathcal{N}) + P_{\text{switch}}(\mathcal{N}) + P_{\text{short}}(\mathcal{N})$$

„Summarize consumption terms of all clock nodes
and make them depend on configuration values and
hardware properties“

$$= \sum_{i=1}^k V^2 \{C_{\Gamma(n_i)} \times f_{\Gamma(n_i)}\} + V \{f_{\Gamma(n_i)} \times I_{\text{short}}(\Gamma(n_i)) \Delta(t_i) + I_{\text{leak}}(\Gamma(n_i))\}$$

$$P_{\text{sys}}(\mathcal{N}) = P_{\text{sys-stat}}(V) + P_{\text{sys-clock}}(\mathcal{N}) + P_{\text{sys-task}}(V, f)$$

$$\mathbf{min} \text{ SSE} = \sum_{\Gamma} (P_{\text{mod}}(\Gamma) - P_{\text{emp}}(\Gamma))^2$$

$$\mathbf{s.t.} \ C_{\Gamma(n_i)} \geq 0$$

$$I_{\text{leak}}(\Gamma(n_i)) \geq 0 \quad \text{and} \quad \bar{I}_{\text{short}}(\Gamma(n_i)) \geq 0$$

Modeling the Full Clock-Tree

$$\begin{aligned}P_{\text{sys-clock}}(\mathcal{N}) &= P_{\text{leak}}(\mathcal{N}) + P_{\text{switch}}(\mathcal{N}) + P_{\text{short}}(\mathcal{N}) \\&= \sum_{i=1}^k \{C_{\Gamma(n_i)} \times V^2 \times f_{\Gamma(n_i)} + V \times f_{\Gamma(n_i)} \times \bar{I}_{\text{short}}(\Gamma(n_i)) \Delta(t_i) + V \times I_{\text{leak}}(\Gamma(n_i))\} \\&= \sum_{i=1}^k V^2 \{C_{\Gamma(n_i)} \times f_{\Gamma(n_i)}\} + V \{f_{\Gamma(n_i)} \times \bar{I}_{\text{short}}(\Gamma(n_i)) \Delta(t_i) + I_{\text{leak}}(\Gamma(n_i))\}\end{aligned}$$

„Relate clock-tree consumption to the overall system power“

$$\begin{aligned}\mathbf{min} \quad & \text{SSE} = \sum_{\Gamma} (P_{\text{mod}}(\Gamma) - P_{\text{emp}}(\Gamma))^2 \\ \mathbf{s.t.} \quad & C_{\Gamma(n_i)} \geq 0 \\ & I_{\text{leak}}(\Gamma(n_i)) \geq 0 \quad \text{and} \quad \bar{I}_{\text{short}}(\Gamma(n_i)) \geq 0\end{aligned}$$

Modeling the Full Clock-Tree

$$\begin{aligned}
 P_{\text{sys-clock}}(\mathcal{N}) &= P_{\text{leak}}(\mathcal{N}) + P_{\text{switch}}(\mathcal{N}) + P_{\text{short}}(\mathcal{N}) \\
 &= \sum_{i=1}^k \{C_{\Gamma(n_i)} \times V^2 \times f_{\Gamma(n_i)} + V \times f_{\Gamma(n_i)} \times \bar{I}_{\text{short}}(\Gamma(n_i)) \Delta(t_i) + V \times I_{\text{leak}}(\Gamma(n_i))\} \\
 &= \sum_{i=1}^k V^2 \{C_{\Gamma(n_i)} \times f_{\Gamma(n_i)}\} + V \{f_{\Gamma(n_i)} \times \bar{I}_{\text{short}}(\Gamma(n_i)) \Delta(t_i) + I_{\text{leak}}(\Gamma(n_i))\}
 \end{aligned}$$

$$P_{\text{sys}}(\mathcal{N}) = P_{\text{sys-stat}}(V) + P_{\text{sys-clock}}(\mathcal{N}) + P_{\text{sys-task}}(V, f)$$

„Objective function to approximate the hidden hardware properties from empirical measurements“

$$\text{s.t. } C_{\Gamma(n_i)} \geq 0$$

$$I_{\text{leak}}(\Gamma(n_i)) \geq 0 \quad \text{and} \quad \bar{I}_{\text{short}}(\Gamma(n_i)) \geq 0$$

Modeling the Full Clock-Tree

$$\begin{aligned}
 P_{\text{sys-clock}}(\mathcal{N}) &= P_{\text{leak}}(\mathcal{N}) + P_{\text{switch}}(\mathcal{N}) + P_{\text{short}}(\mathcal{N}) \\
 &= \sum_{i=1}^k \{C_{\Gamma(n_i)} \times V^2 \times f_{\Gamma(n_i)} + V \times f_{\Gamma(n_i)} \times \bar{I}_{\text{short}}(\Gamma(n_i)) \Delta(t_i) + V \times I_{\text{leak}}(\Gamma(n_i))\} \\
 &= \sum_{i=1}^k V^2 \{C_{\Gamma(n_i)} \times f_{\Gamma(n_i)}\} + V \{f_{\Gamma(n_i)} \times \bar{I}_{\text{short}}(\Gamma(n_i)) \Delta(t_i) + I_{\text{leak}}(\Gamma(n_i))\}
 \end{aligned}$$

$$P_{\text{sys}}(\mathcal{N}) = P_{\text{sys-stat}}(V) + P_{\text{sys-clock}}(\mathcal{N}) + P_{\text{sys-task}}(V, f)$$

$$\mathbf{min} \text{ SSE} = \sum_{\Gamma} (P_{\text{mod}}(\Gamma) - P_{\text{emp}}(\Gamma))^2$$

$$\mathbf{s.t.} \ C_{\Gamma(n_i)} \geq 0$$

$$I_{\text{leak}}(\Gamma(n_i)) \geq 0 \text{ and } \bar{I}_{\text{short}}(\Gamma(n_i)) \geq 0$$

„Account for physical constraints“

Modeling the Full Clock-Tree

$$\begin{aligned} P_{\text{sys-clock}}(\mathcal{N}) &= P_{\text{leak}}(\mathcal{N}) + P_{\text{switch}}(\mathcal{N}) + P_{\text{short}}(\mathcal{N}) \\ &= \sum_{i=1}^k \{C_{\Gamma(n_i)} \times V^2 \times f_{\Gamma(n_i)} + V \times f_{\Gamma(n_i)} \times \bar{I}_{\text{short}}(\Gamma(n_i)) \Delta(t_i) + V \times I_{\text{leak}}(\Gamma(n_i))\} \\ &= \sum_{i=1}^k V^2 \{C_{\Gamma(n_i)} \times f_{\Gamma(n_i)}\} + V \{f_{\Gamma(n_i)} \times \bar{I}_{\text{short}}(\Gamma(n_i)) \Delta(t_i) + I_{\text{leak}}(\Gamma(n_i))\} \end{aligned}$$

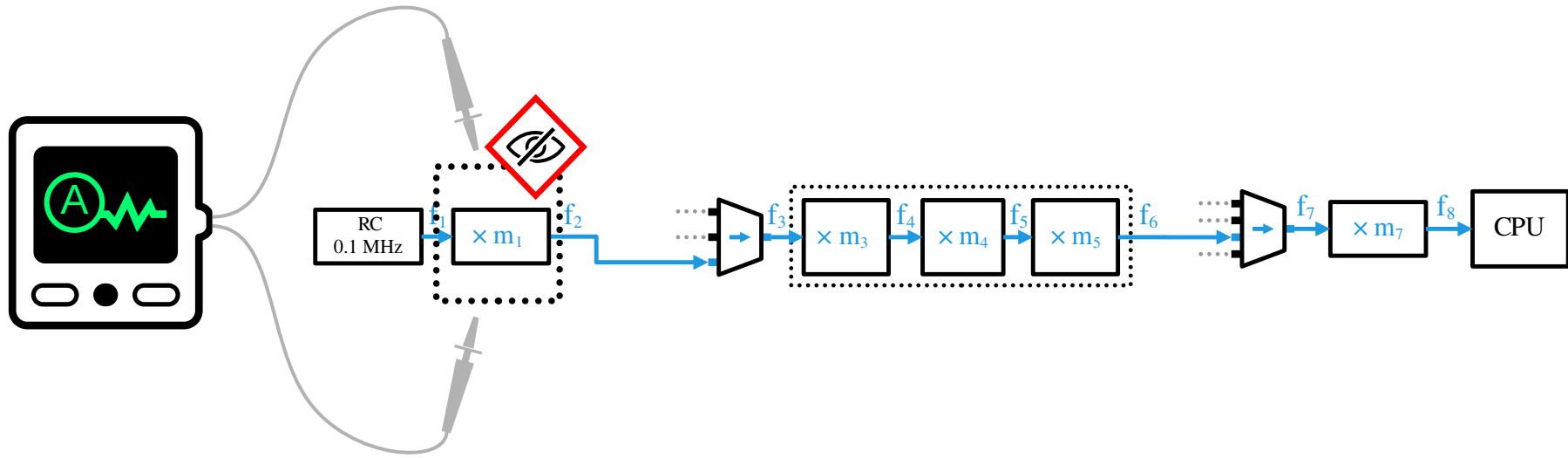
$$P_{\text{sys}}(\mathcal{N}) = P_{\text{sys-stat}}(V) + P_{\text{sys-clock}}(\mathcal{N}) + P_{\text{sys-task}}(V, f)$$

$$\mathbf{min} \text{ SSE} = \sum_{\Gamma} (P_{\text{mod}}(\Gamma) - P_{\text{emp}}(\Gamma))^2$$

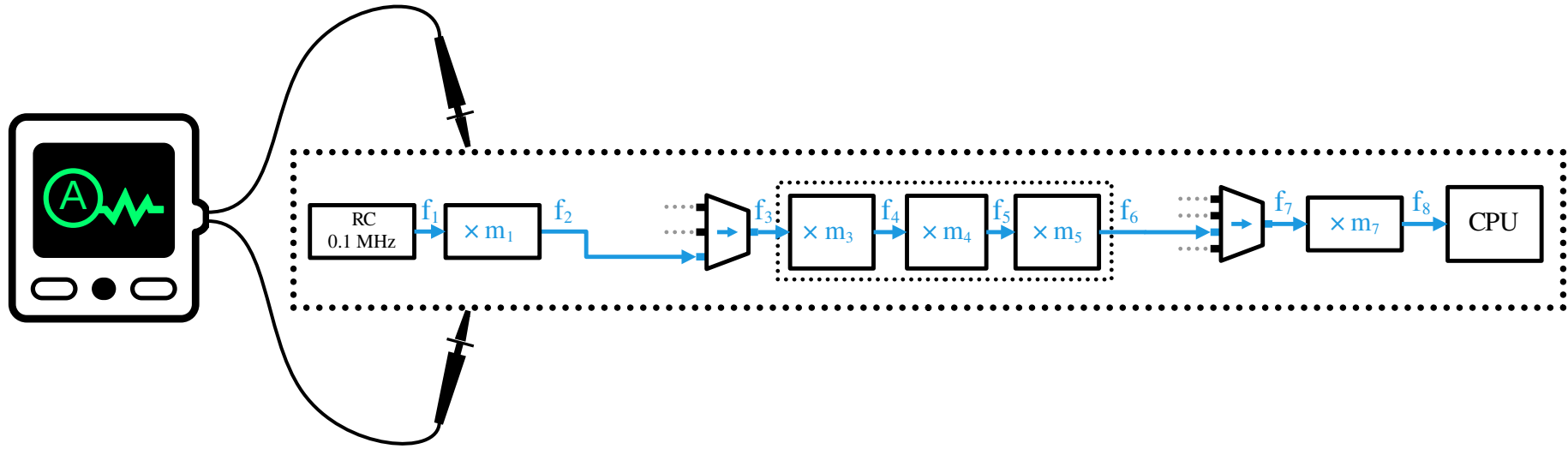
$$\mathbf{s.t.} \ C_{\Gamma(n_i)} \geq 0$$

$$I_{\text{leak}}(\Gamma(n_i)) \geq 0 \quad \text{and} \quad \bar{I}_{\text{short}}(\Gamma(n_i)) \geq 0$$

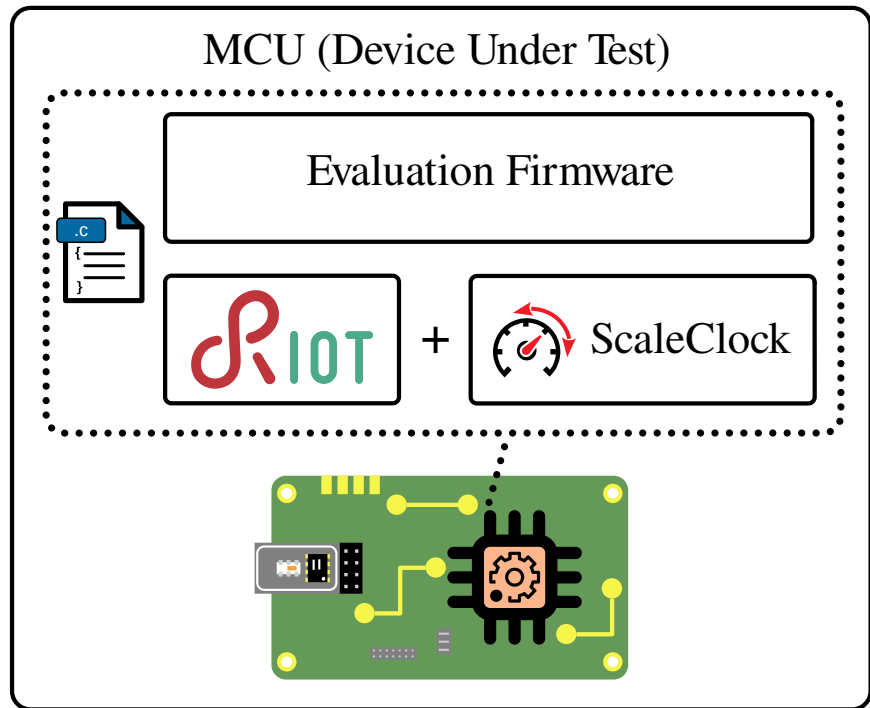
Model Input: Measurement



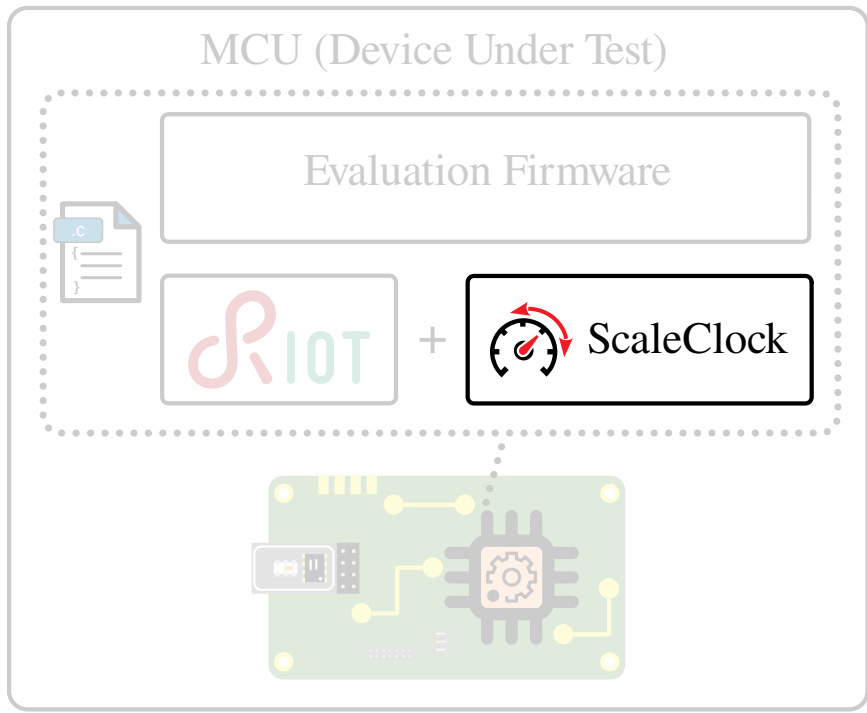
Model Input: Measurement



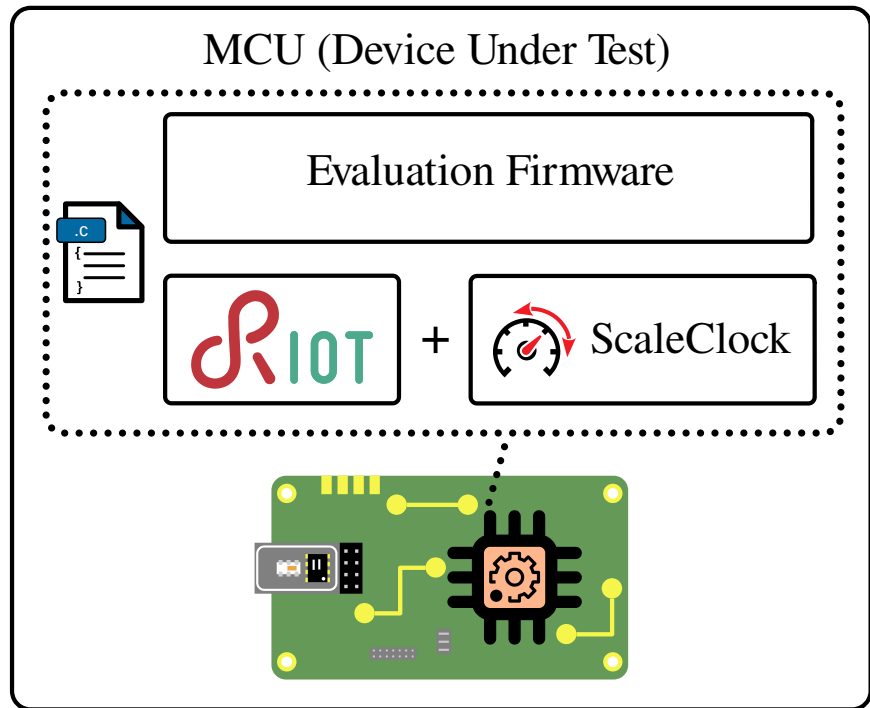
Automatic Parametrization of the Model



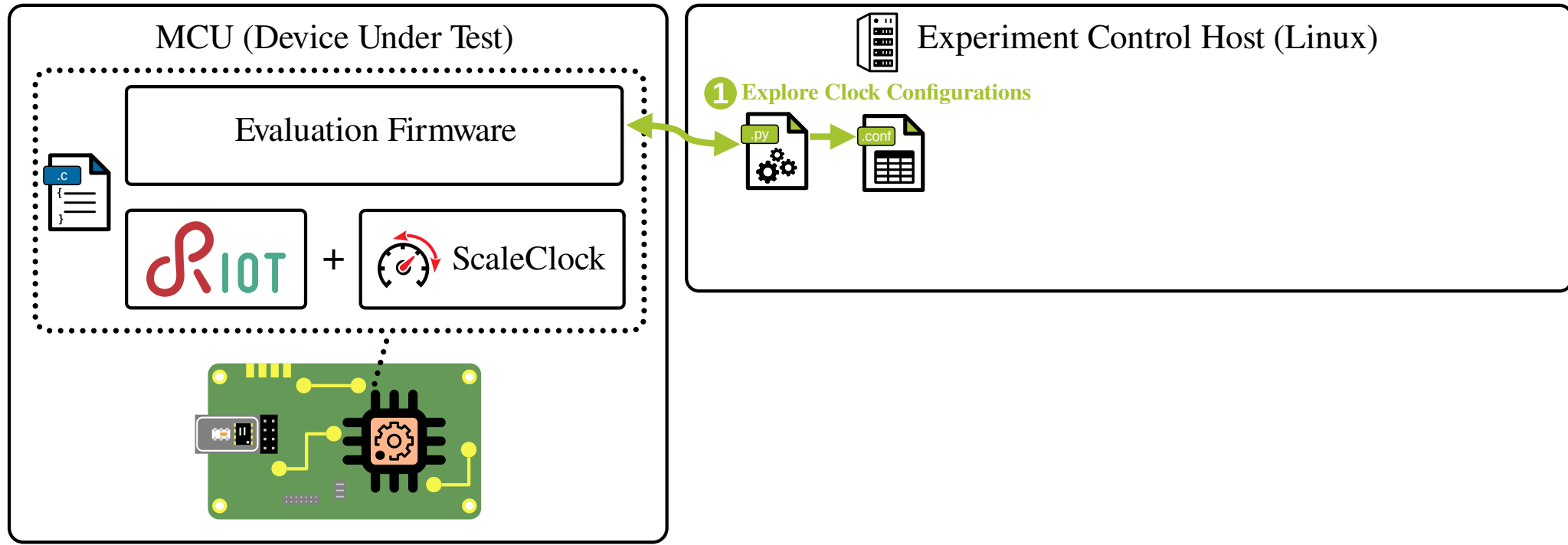
Automatic Parametrization of the Model



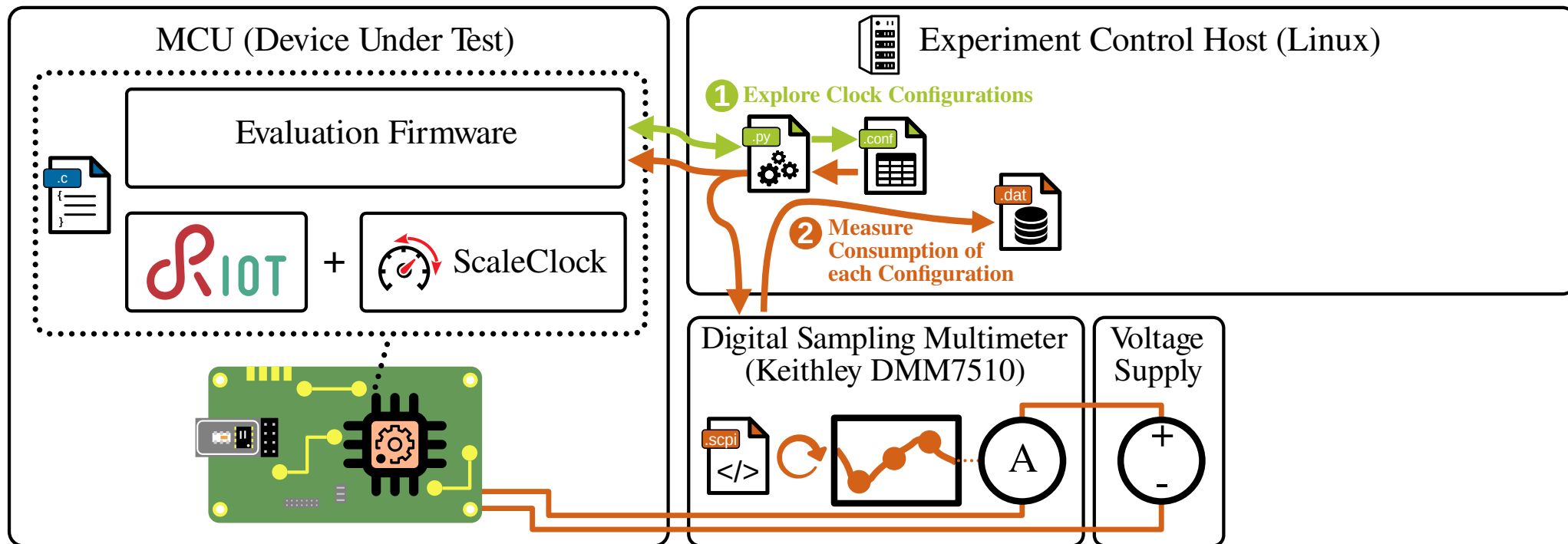
Automatic Parametrization of the Model



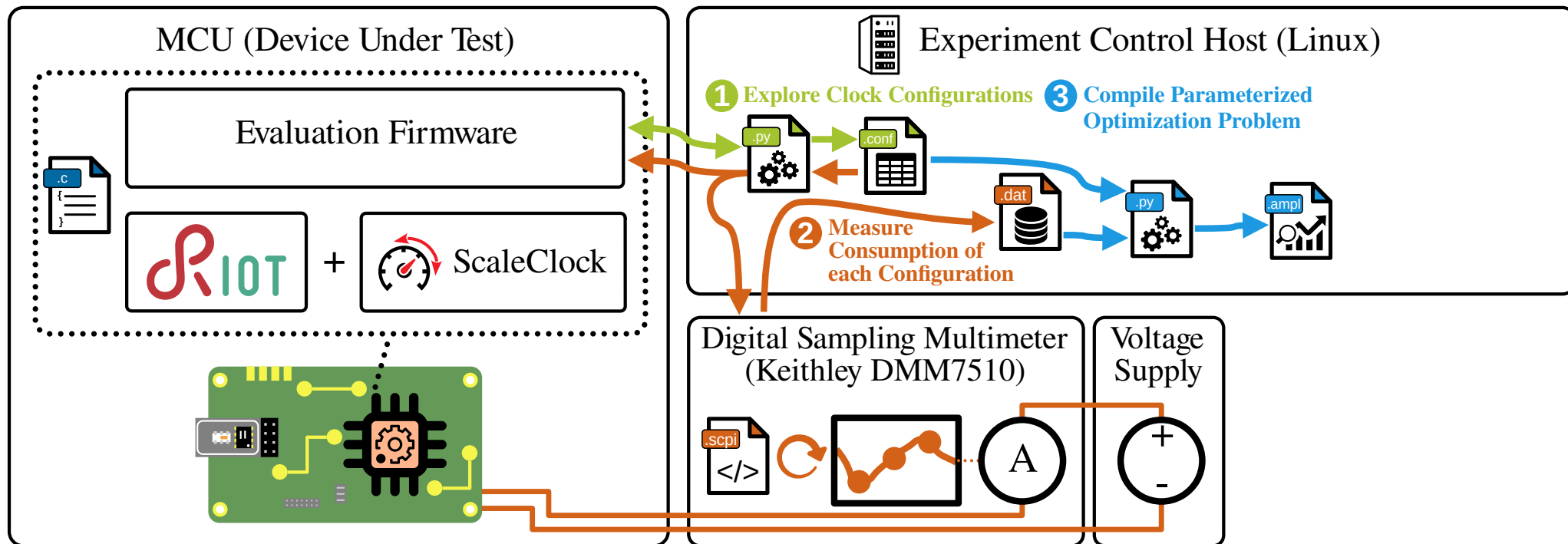
Automatic Parametrization of the Model



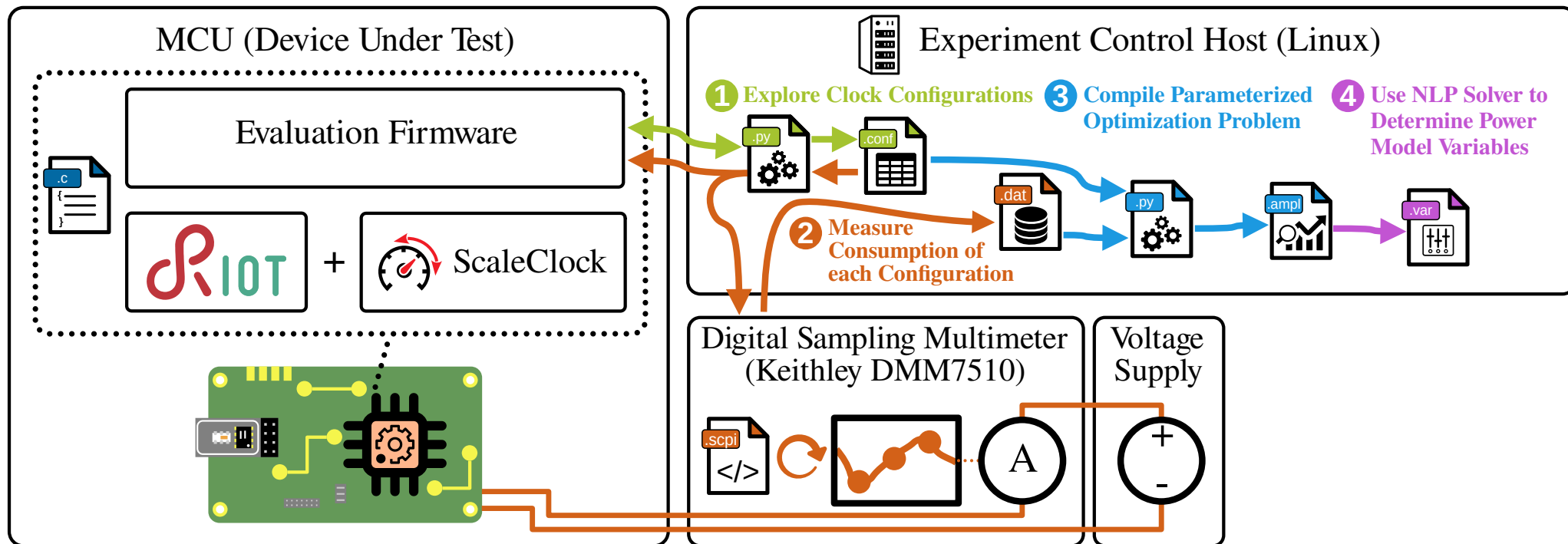
Automatic Parametrization of the Model



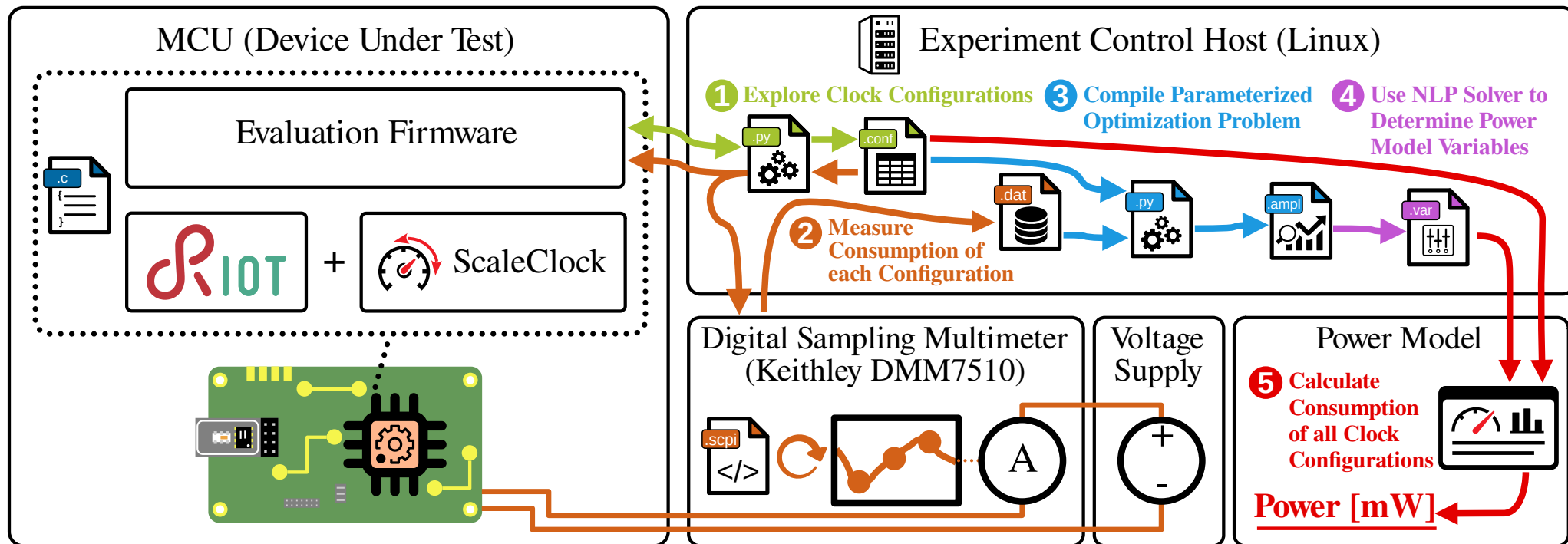
Automatic Parametrization of the Model



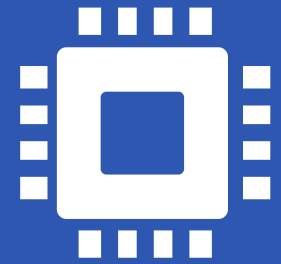
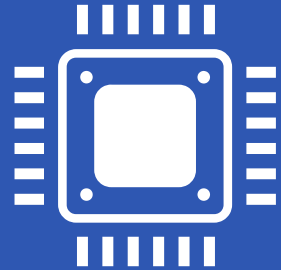
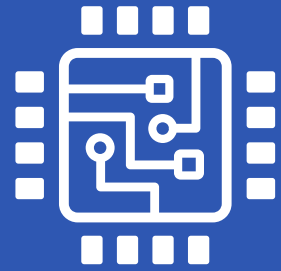
Automatic Parametrization of the Model



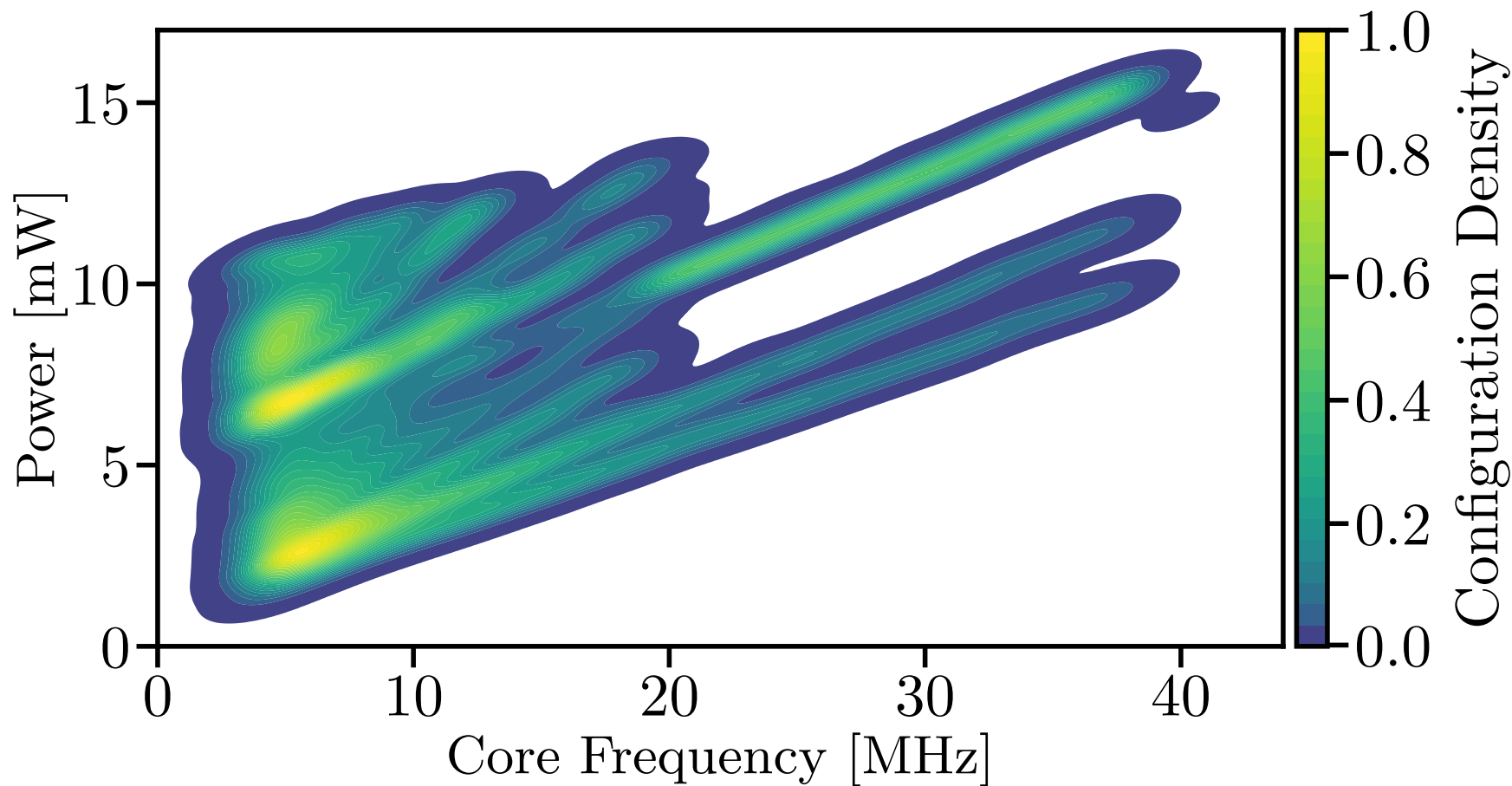
Automatic Parametrization of the Model



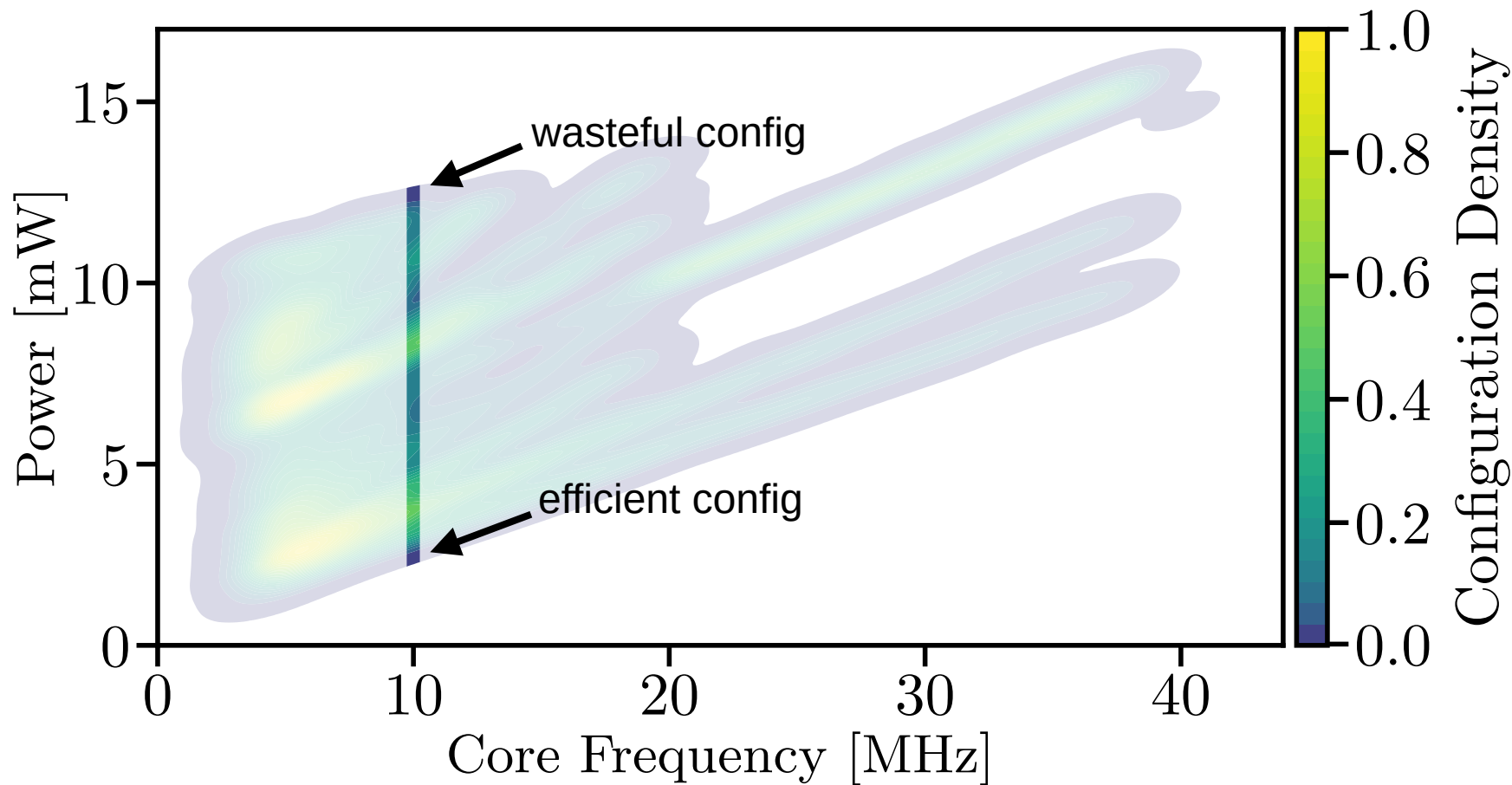
Evaluation: Three Very Different Boards



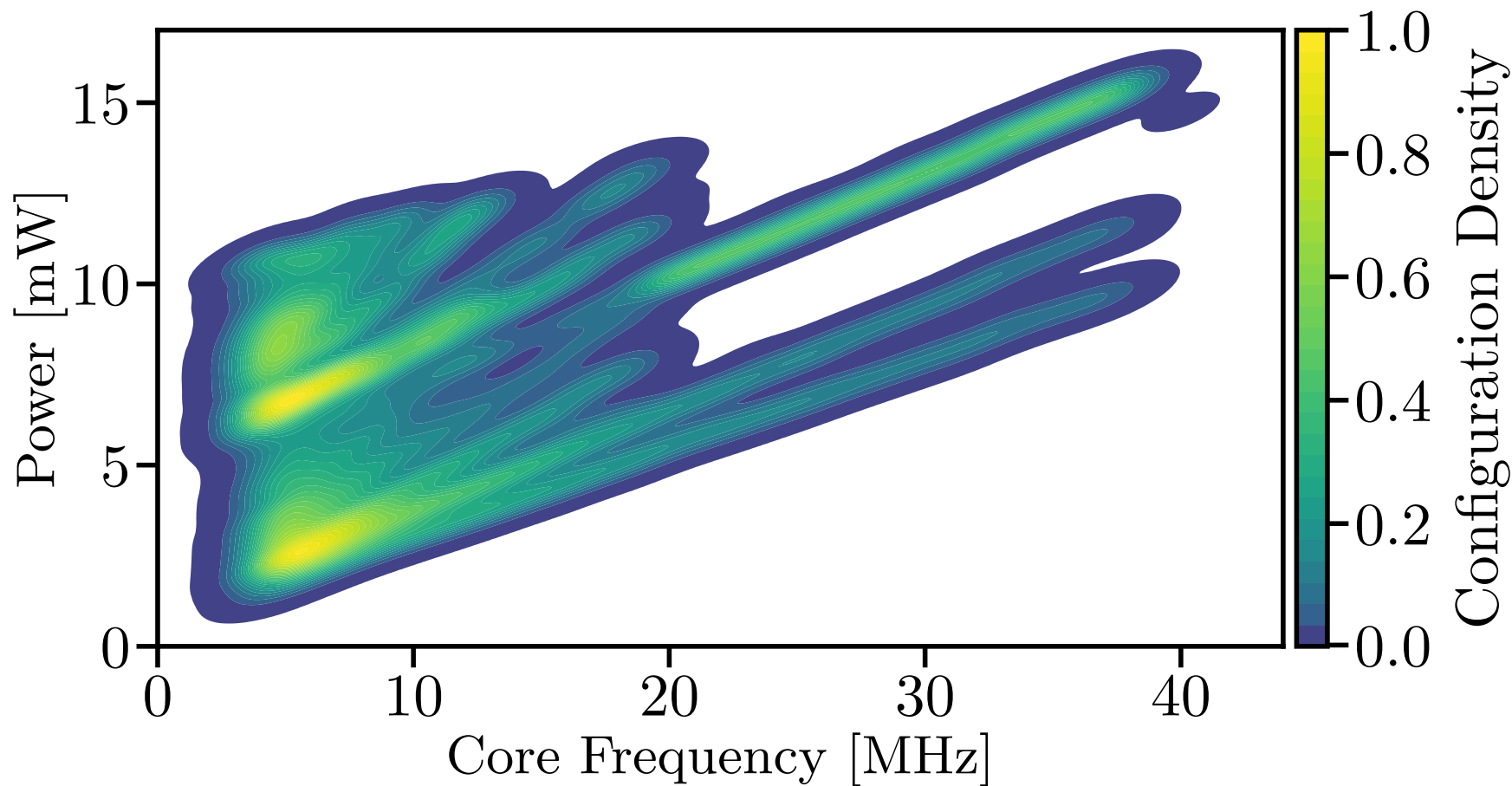
Clock Configuration Space (slstk3402a)



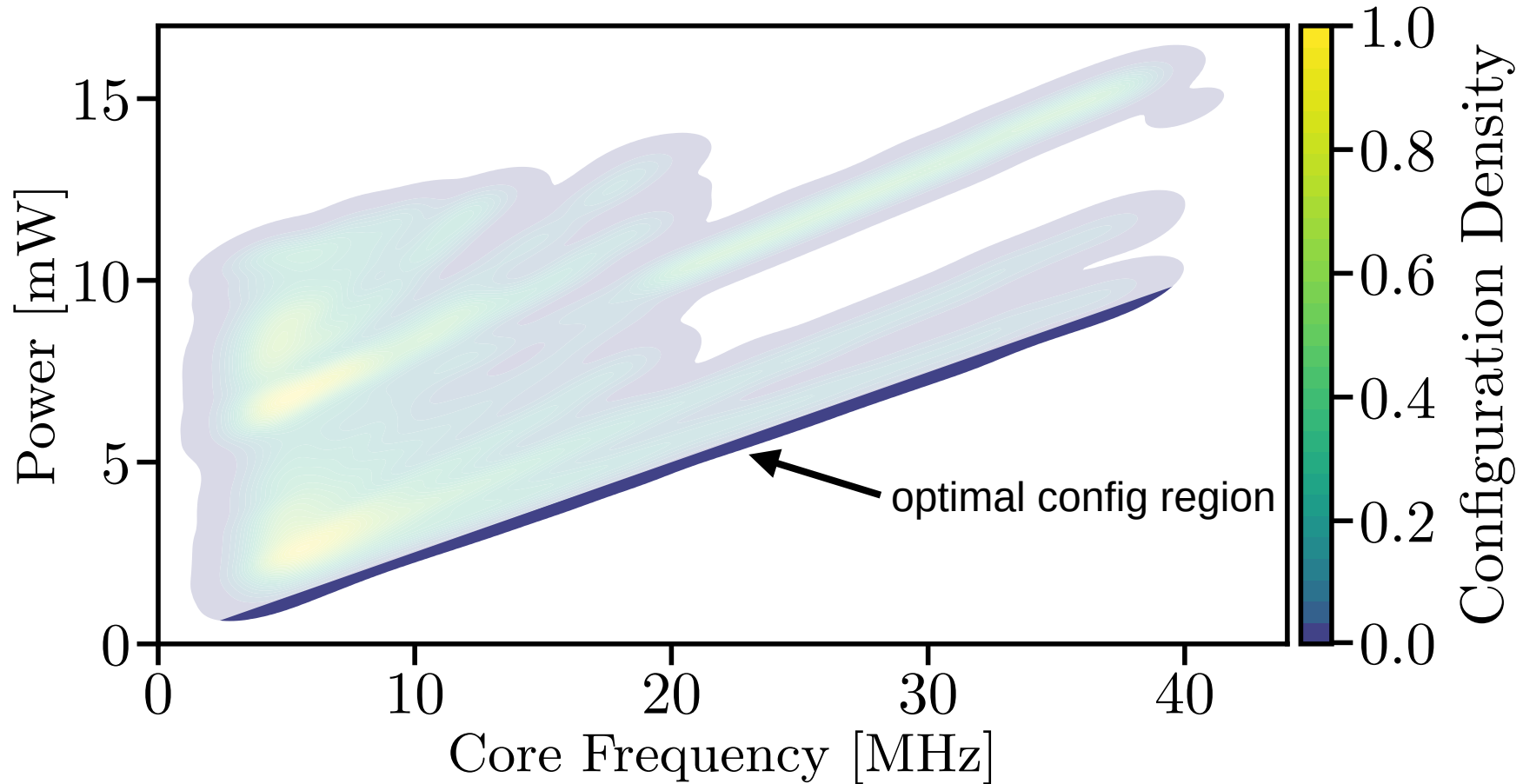
Clock Configuration Space (slstk3402a)



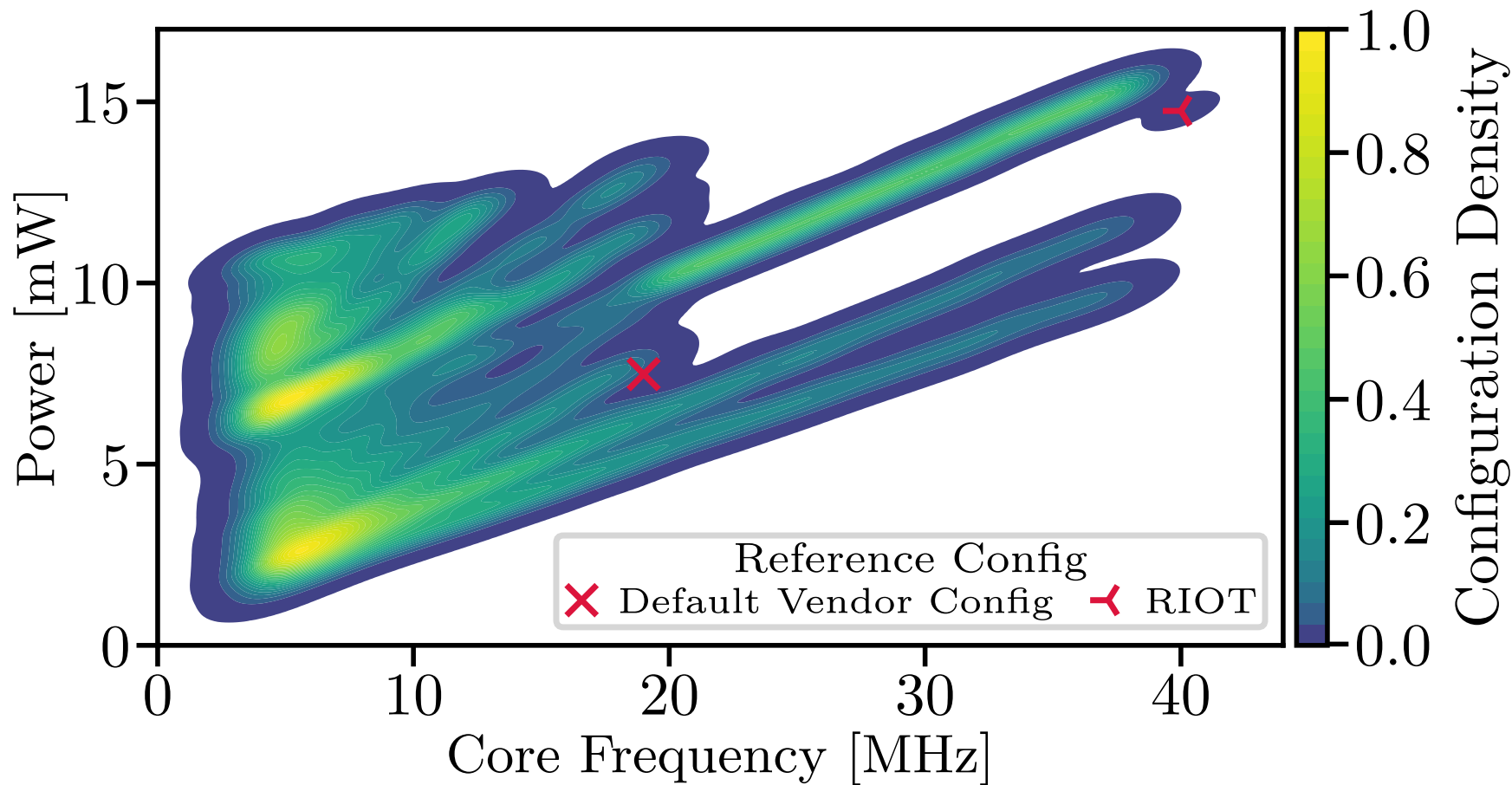
Clock Configuration Space (slstk3402a)



Clock Configuration Space (slstk3402a)

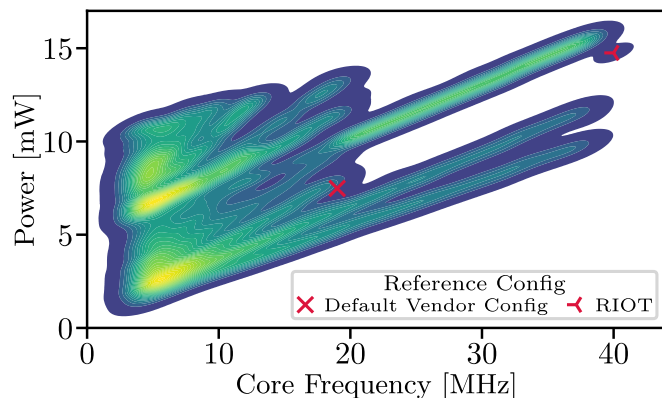


Clock Configuration Space (slstk3402a)

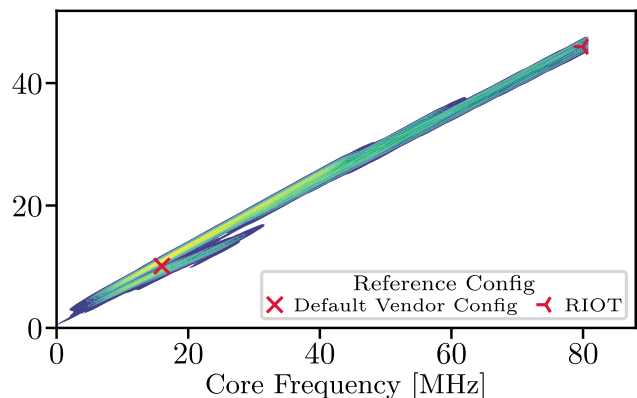


Clock Configuration Space

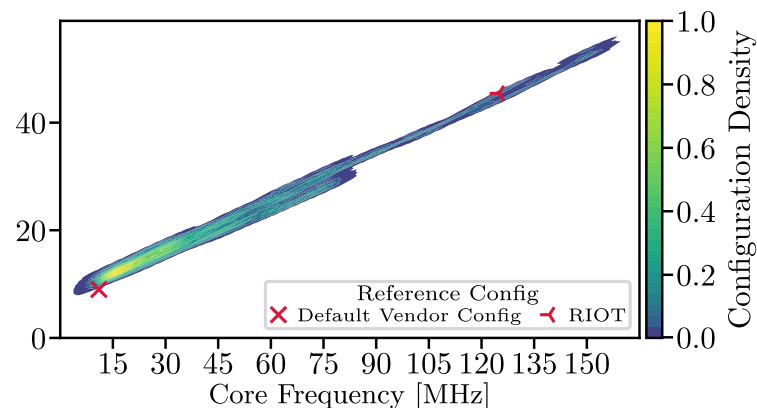
slstk3402a



nucleo-l476rg



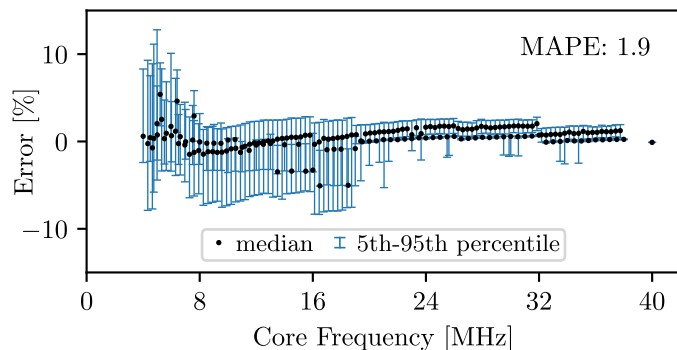
rpi-pico-2-riscv



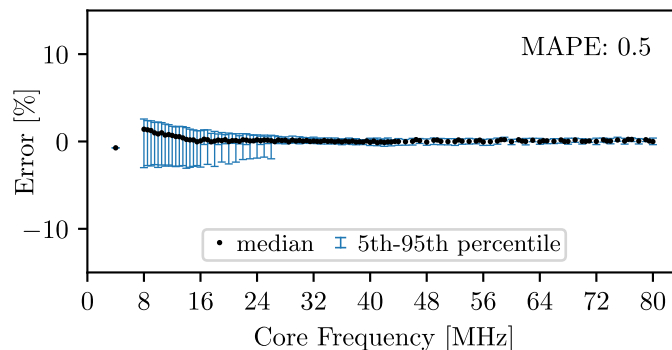
Lower frequencies and more flexible tree → wider power range
Best config saves 22% - 47% vs. vendor- and 2% - 4% vs. OS-default

Power Model Accuracy

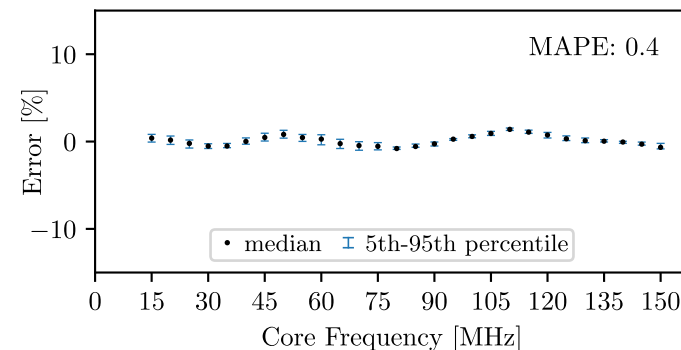
slstk3402a



nucleo-l476rg



rpi-pico-2-riscv



Model generalizes across manufacturers and core architectures
Configuration-space is covered with small errors (MAPE < 2%)

Conclusion and Outlook

Conclusion and Outlook

- Power estimation requires more than core frequency and voltage
 - Clock topology and intermediate scale factors matter
- Our power model of system clocks
 - Covers the full clock configuration space
 - Accurately predicts the power consumption (MAPE 0.4% - 1.9%)
 - Generalizes to heterogeneous target devices
(Evaluated for Cortex-M and RISC-V on Silicon Labs, STMicroelectronics, Raspberry Pi)
- Automatic framework to measure devices and parametrize our model
- Application directions
 - Dynamic and static scenarios
 - CI and dev tool integration for compile-time hints
 - On device deployment for adaptive power-awareness

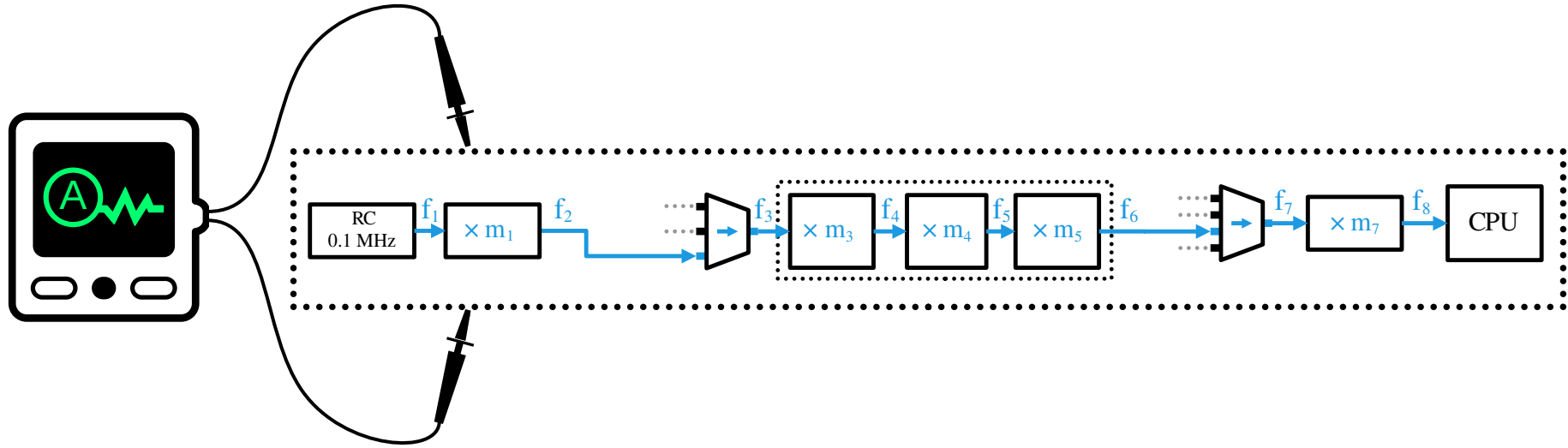
Source Code



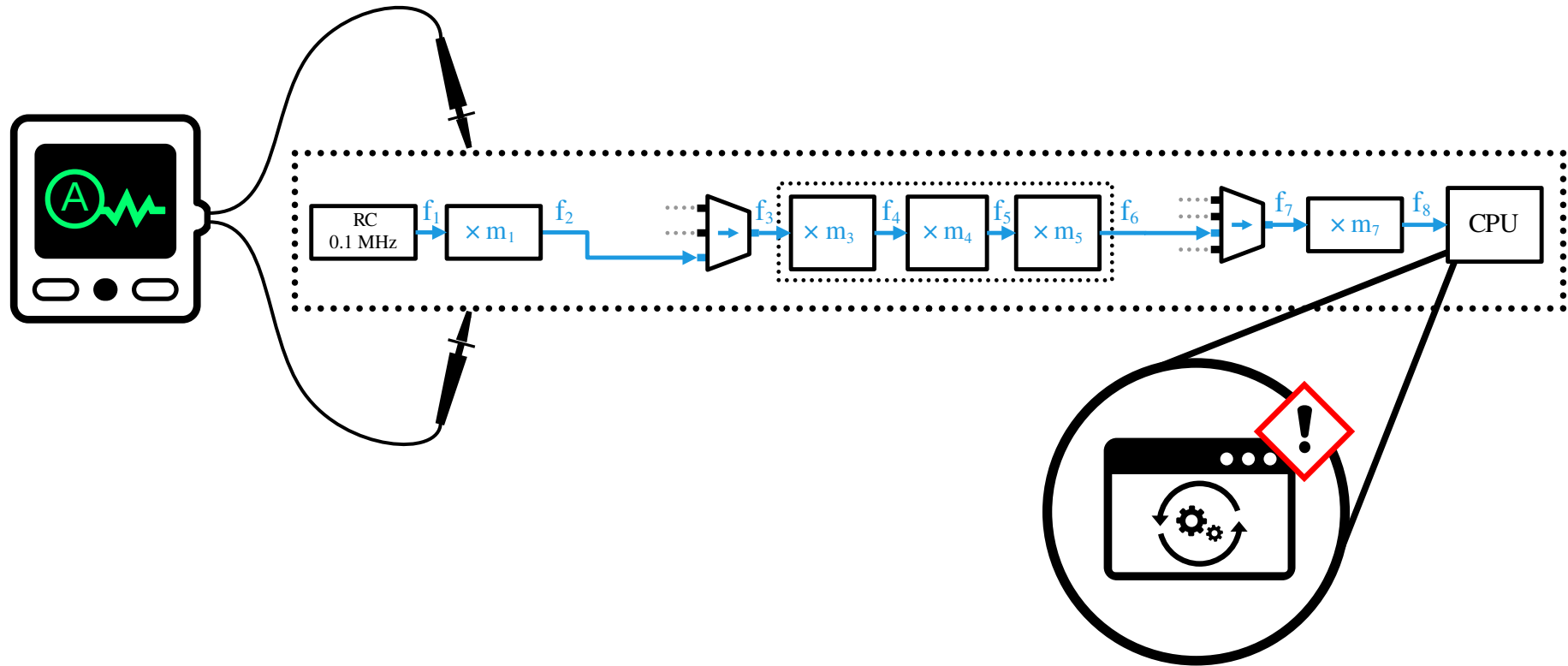
<https://git.inet.haw-hamburg.de/inetrg/clock-power-model> →

Questions?

Clock-Tree Power Model: Task Independence



Clock-Tree Power Model: Task Independence



Clock-Tree Power Model: Task Independence

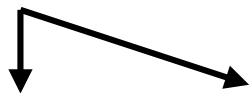
Hypothesis: at the same CPU frequency and voltage the difference in system consumption for two configurations should be equal to the difference of the clock tree consumption.

$$P_{\text{sys}}(\mathcal{N}) = P_{\text{sys-stat}}(V) + P_{\text{sys-clock}}(\mathcal{N}) + P_{\text{sys-task}}(V, f)$$

Clock-Tree Power Model: Task Independence

Hypothesis: at the same CPU frequency and voltage the difference in system consumption for two configurations should be equal to the difference of the clock tree consumption.

$$P_{\text{sys}}(\mathcal{N}) = P_{\text{sys-stat}}(V) + P_{\text{sys-clock}}(\mathcal{N}) + P_{\text{sys-task}}(V, f)$$



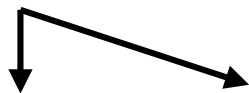
$$\Delta = P_{\text{sys}}(\mathcal{N}) - P_{\text{sys}}(\mathcal{N}') = P_{\text{sys-clock}}(\mathcal{N}) - P_{\text{sys-clock}}(\mathcal{N}')$$

$$\underline{\underline{\Delta(w_i) - \Delta(w_0) \approx 0}}$$

Clock-Tree Power Model: Task Independence

Hypothesis: at the same CPU frequency and voltage the difference in system consumption for two configurations should be equal to the difference of the clock tree consumption.

$$P_{\text{sys}}(\mathcal{N}) = P_{\text{sys-stat}}(V) + P_{\text{sys-clock}}(\mathcal{N}) + P_{\text{sys-task}}(V, f)$$



$$\Delta = P_{\text{sys}}(\mathcal{N}) - P_{\text{sys}}(\mathcal{N}') = P_{\text{sys-clock}}(\mathcal{N}) - P_{\text{sys-clock}}(\mathcal{N}')$$

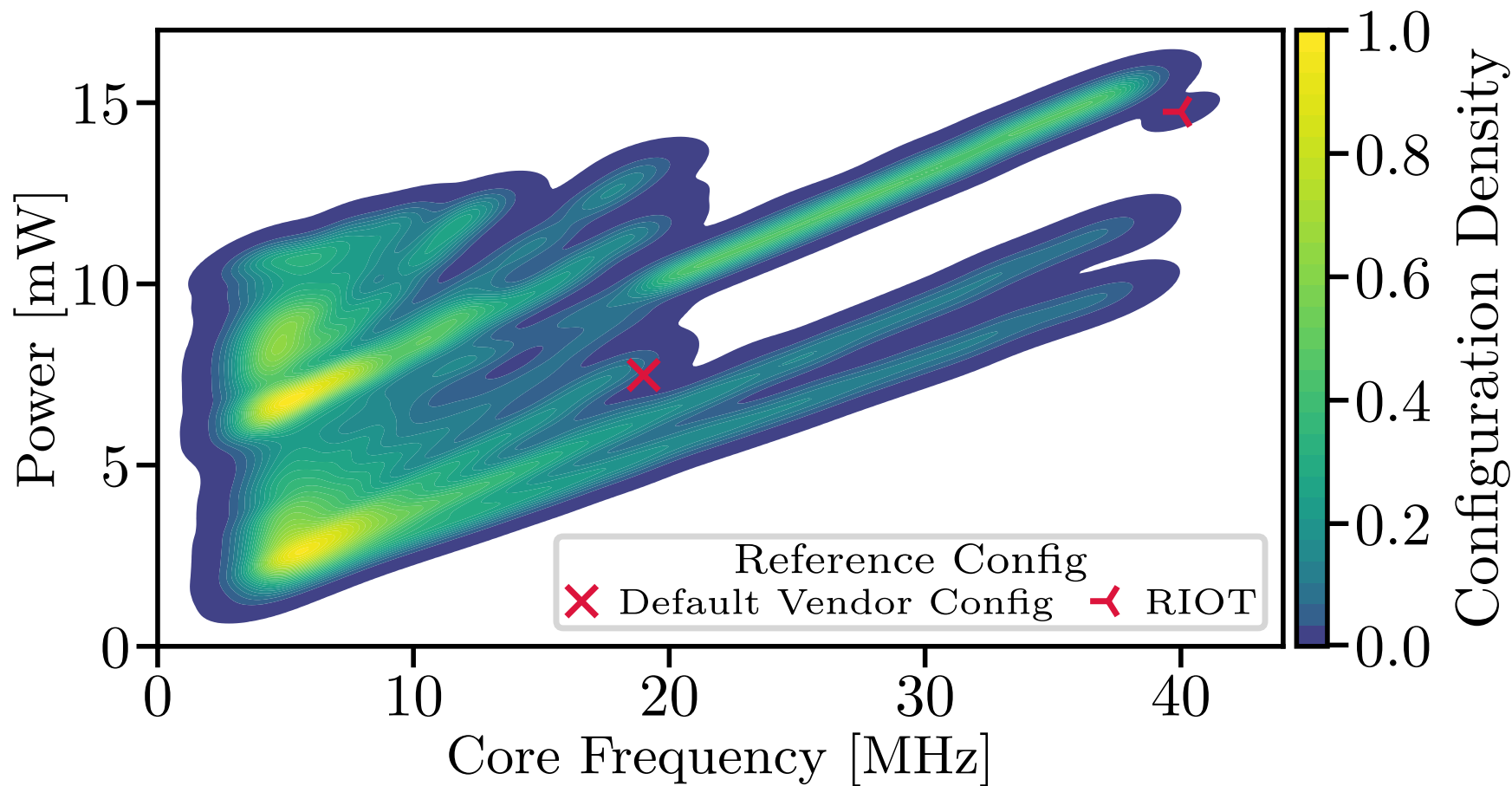
$$\Delta(w_i) - \Delta(w_0) \approx 0$$

Confirmation:

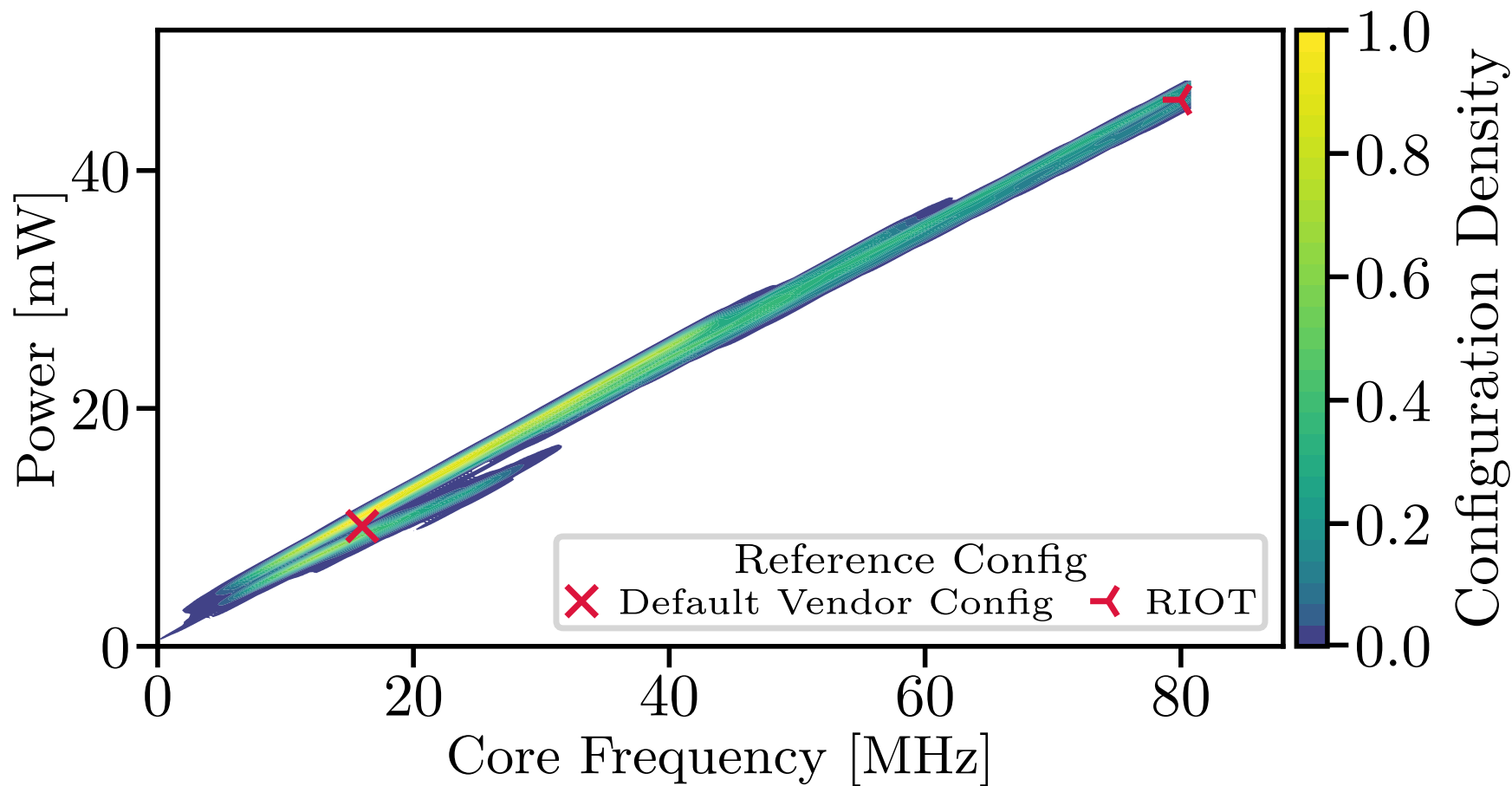
		Residual Error		
		Total Consumption	Mean	Standard Deviation
w_i {	flash	23.6 - 25.6 mW	11.8 μ W (0.05 %)	15.0 μ W (0.06 %)
	count	13.2 - 15.3 mW	7.3 μ W (0.05 %)	5.4 μ W (0.04 %)
	timer	12.2 - 14.3 mW	7.0 μ W (0.05 %)	5.2 μ W (0.04 %)



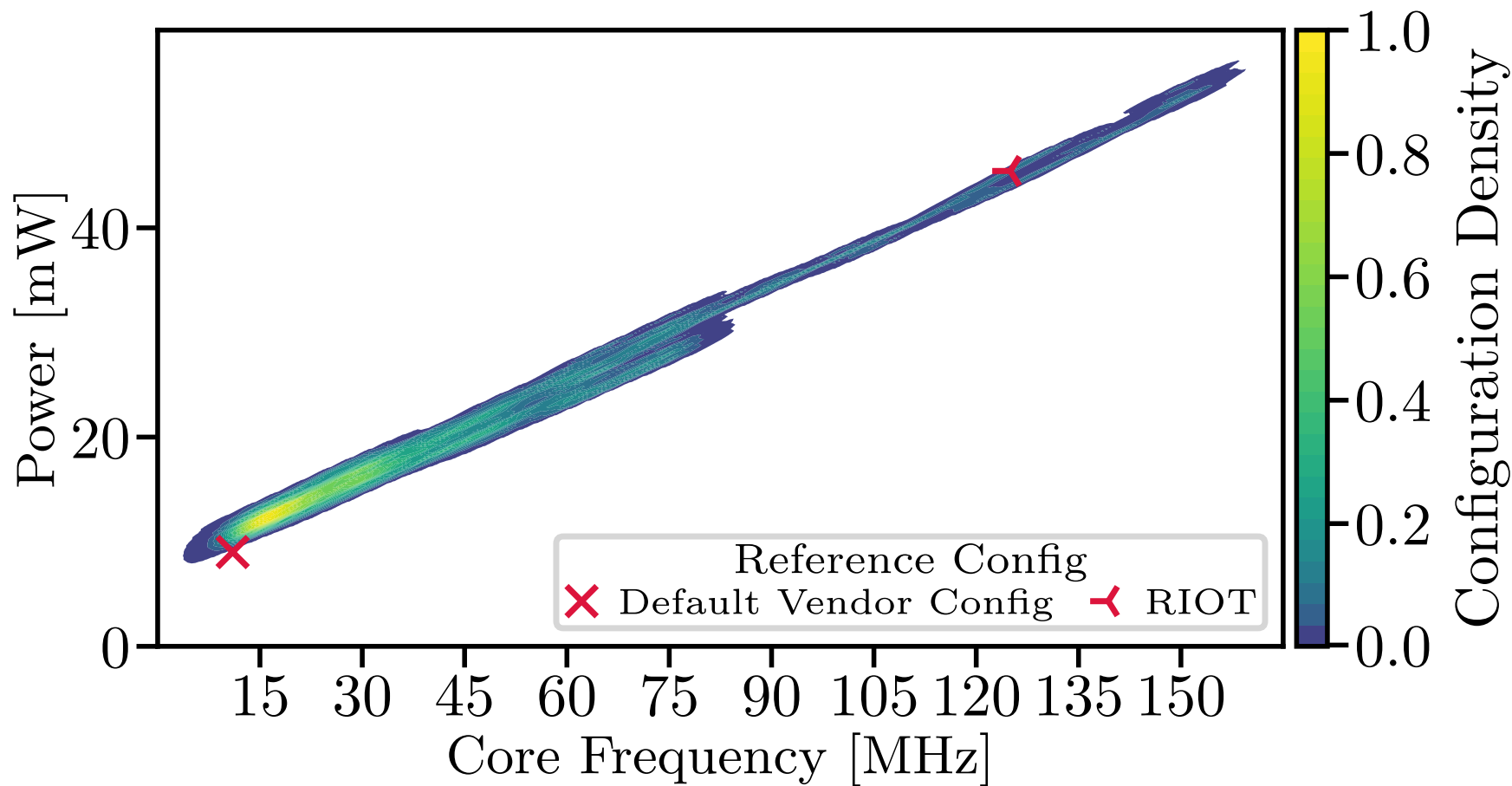
Clock Configuration Space (slstk3402a)



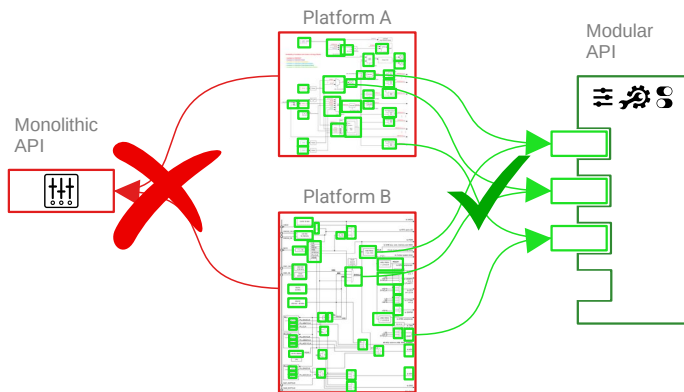
Clock Configuration Space (nucleo-l476rg)



Clock Configuration Space (rpi-pico-2-riscv)



Clock Tree Abstraction



Architecture & OS-Integration

