Reducing Energy Consumption in MCU-based Platforms with Low Design Margin Co-Processors

A. Gomez[1], C. Pinto[2], A. Bartolini[1,2], D. Rossi[2], L. Benini[1,2], H. Fatemi[3], and J. Pineda de Gyvez[3]

[1]Integrated Systems Laboratory
ETH Zurich, Switzerland
firstname.lastname@iis.ee.ethz.ch

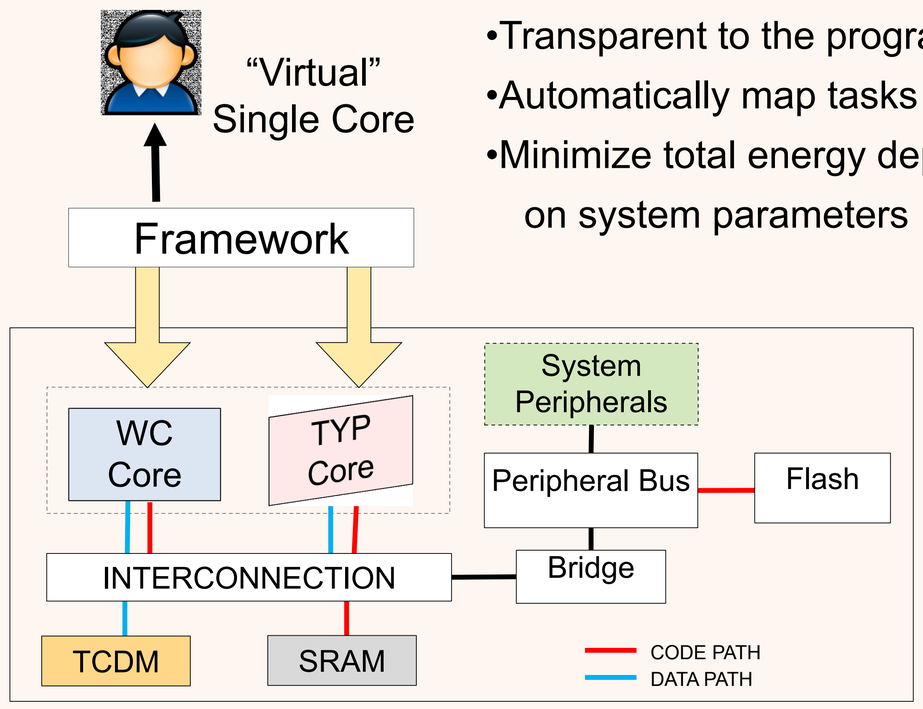
[2]DEI
University of Bologna, Italy,
firstname.lastname@unibo.it

[3]NXP Semiconductors, Central R&D
The Netherlands
firstname.lastname@nxp.com

- Worst-Case (WC) Core manufactured with conservative margins has **low** energy efficiency
- Typical (TYP) Core manufactured with aggressive margins has a **high** energy efficiency
- Post-synthesis simulations show up to **30% efficiency boost**
  - TYP might run at a **reduced** frequency
  - WC core **needed** to guarantee minimum performance

Proposed Architecture:

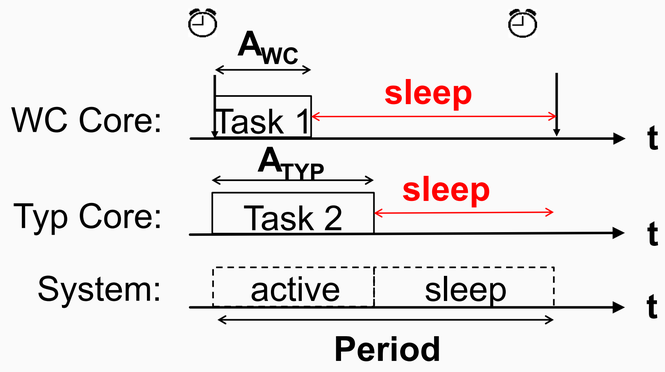
- Dual-core containing one **WC**, and one **TYP** core
- **Guaranteed** performance with possible **energy savings**



Framework requirements:

- Transparent to the programmer
- Automatically map tasks to cores
- Minimize total energy depending on system parameters

Sample Dual-Core (Parallel) Execution:

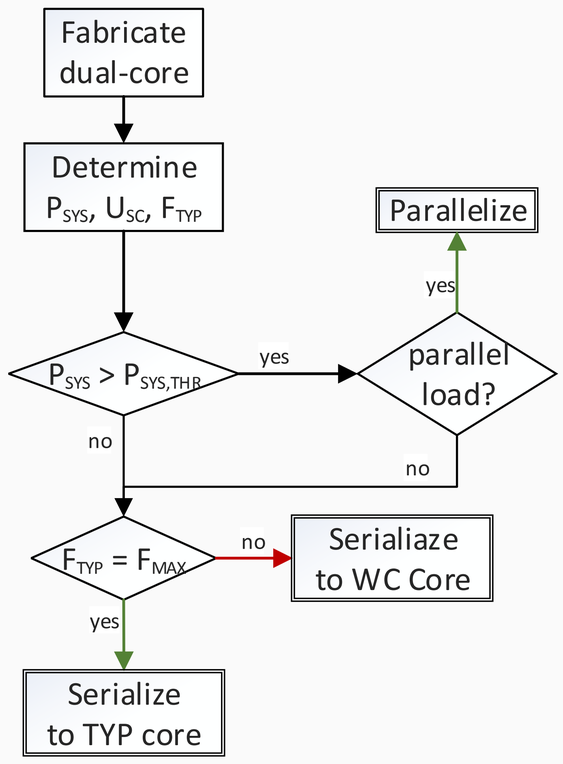


Power Model:

	Active	Sleep
WC	163.2 mW	9.6 mW
TYP	114.24 mW	6.74 mW

- P_{SYS} is an unknown constant.
- $P_{SYS} = P_{sleep,SYS}$ only when **both** cores in sleep

Dual-Core MCU Flow Chart:



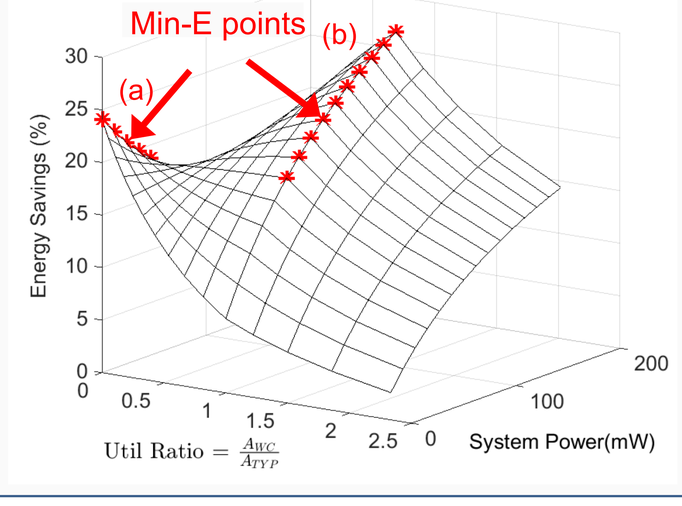
Optimization problem:

$$\min E_{TOT,DC} \Longrightarrow \min A_D * (P_{\Delta,\mathrm{TYP}} - P_{\Delta,\mathrm{WC}} + P_{\Delta,\mathrm{SYS}})$$

$$A_D^* = \begin{cases} A_{TYP} + A_{WC} \text{ if } (P_{\Delta,\mathrm{TYP}} - P_{\Delta,\mathrm{WC}} + P_{\Delta,\mathrm{SYS}}) < 0 \\ 0 \text{ if } (P_{\Delta,\mathrm{TYP}} - P_{\Delta,\mathrm{WC}} + P_{\Delta,\mathrm{SYS}}) \geq 0 \end{cases}$$

↳ Optimal mapping depends on **power ratios**

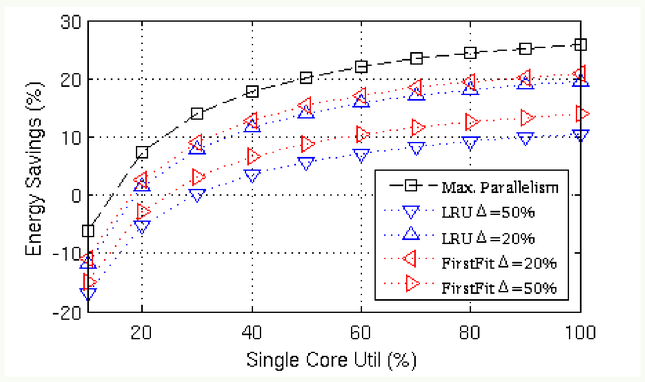
Design Space:

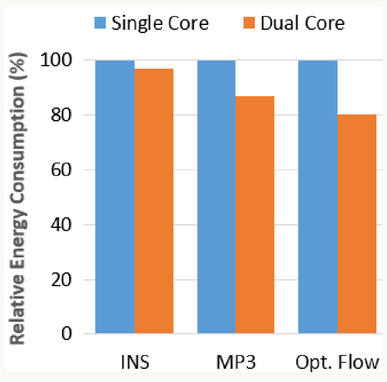


Experimental Results:

- Instruction Accurate (SystemC) simulation environment
- Synthetic + real world benchmarks show significant savings

Results calculated with: $P_{SYS} = P_{WC}$ and $F_{TYP} = F_{MAX}$

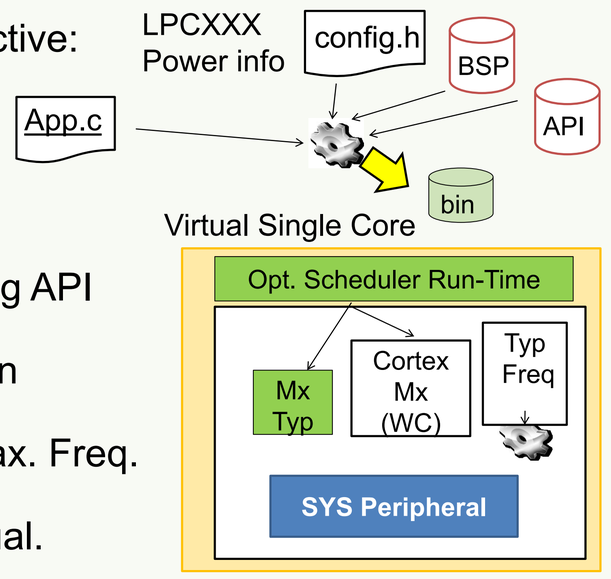




End Product:

From the developer's perspective:

- Select the platform.
  - Using config.h
- Port and compile APP, linking API
- At boot time – self-calibration
  - Determine TYP Core's max. Freq.
- Execute the program as usual.



References:

[1] D. Bortolotti, C. Pinto, A. Marongiu, M. Ruggiero, and L. Benini, "VirtualSoC: A Full-System Simulation Environment for Massively Parallel Heterogeneous System-on-Chip," in IEEE IPDPSW, 2013.
[2] K. Jeong, A. B. Kahng, and K. Samadi, "Quantified Impacts of Guardband Reduction on Design Process Outcomes," in IEEE ISQED, 2008.
[3] A. Lukefahr, S. Padmanabha, R. Das, F. M. Sleiman, R. Dreslinski, T. F. Wenisch, and S. Mahlke, "Composite Cores: Pushing Heterogeneity Into a Core," in MICRO. IEEE, 2012.
[4] R. Nishtala, D. Mossé, and V. Petrucci, "Energy-aware thread colocation in heterogeneous multicore processors," in IEEE EMSOFT, 2013, pp. 1 – 9.