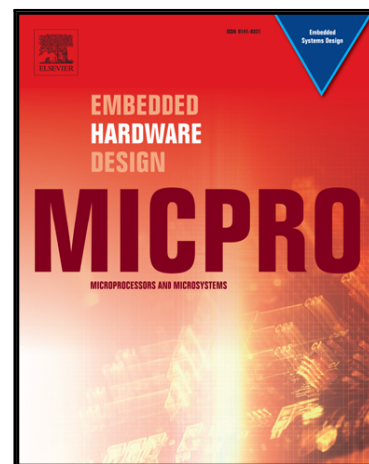


Increasing the Energy Efficiency of Microcontroller Platforms with Low-Design Margin Co-Processors

Andres Gomez, Andrea Bartolini, Davide Rossi, Barş Can Kara, Hamed Fatemi, José Pineda de Gyvez, Luca Benini

PII: S0141-9331(16)30245-9
DOI: [10.1016/j.micpro.2017.05.012](https://doi.org/10.1016/j.micpro.2017.05.012)
Reference: MICPRO 2561



To appear in: *Microprocessors and Microsystems*

Received date: 8 October 2016
Revised date: 6 March 2017
Accepted date: 23 May 2017

Please cite this article as: Andres Gomez, Andrea Bartolini, Davide Rossi, Barş Can Kara, Hamed Fatemi, José Pineda de Gyvez, Luca Benini, Increasing the Energy Efficiency of Microcontroller Platforms with Low-Design Margin Co-Processors, *Microprocessors and Microsystems* (2017), doi: [10.1016/j.micpro.2017.05.012](https://doi.org/10.1016/j.micpro.2017.05.012)

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

Increasing the Energy Efficiency of Microcontroller Platforms with Low-Design Margin Co-Processors

A. Gomez*, A. Bartolini*, D. Rossi[†], B. Can Kara[‡], H. Fatemi[‡], J. Pineda de Gyvez[‡], L. Benini*

*D-ITET, ETH Zurich, Switzerland
Email: {firstname.lastname}@ethz.ch

[†]DEIS, University of Bologna, Italy
Email: {firstname.lastname}@unibo.it

[‡]NXP Semiconductors, Netherlands
Email: {firstname.lastname}@nxp.com

Abstract—Reducing the energy consumption in low cost, performance-constrained microcontroller units (MCU's) cannot be achieved with complex energy minimization techniques (i.e. fine-grained DVFS, Thermal Management, etc), due to their high overheads. To this end, we propose an energy-efficient, multi-core architecture combining two homogeneous cores with different design margins. One is a performance-guaranteed core, also called Heavy Core (HC), fabricated with a worst-case design margin. The other is a low-power core, called Light Core (LC), which has only a typical-corner design margin. Post-silicon measurements show that the *Light* core has a 30% lower power density compared to the *Heavy* core, with only a small loss in reliability. Furthermore, we derive the energy-optimal workload distribution and propose a runtime environment for Heavy/Light MCU platforms. The runtime decreases the overall energy by exploiting available parallelism to minimize the platform's active time. Results show that, depending on the core to peripherals power-ratio and the *Light* core's operating frequency, the expected energy savings range from 10 to 20%.

I. INTRODUCTION

In the mid-to-high performance range, symmetric multi-core architectures have typically been used to achieve a higher throughput with a lower power consumption than single-core systems. Such systems exploit several architectural techniques to improve their energy efficiency proportionally to performance requirements.

More recently, heterogeneous computing has been proposed for the purpose of increasing the energy efficiency for a wide range of performance targets, such as the big.LITTLE architectures [10]. A big.LITTLE multi-core is composed of one (or a set of) high performance cores (e.g. ARM Cortex A15), and a set of smaller but more power-efficient cores (e.g. ARM Cortex A7). In this way, if the low-power cores can satisfy an application's performance requirements, it is possible to achieve significant energy savings. In these systems, the use of advanced hardware infrastructure such as fine-grained Dynamic Voltage and Frequency Scaling (DVFS) can further improve the energy efficiency. These systems, however, require memory virtualization, multi-threading and full-featured operating systems, such as Linux or Android. If these requirements are met, task allocation and voltage/frequency configuration processes can be easily integrated with the operating system.

On the other side of the spectrum, the low-end micro controller unit (MCU) market is dominated by products based

on the ARM Cortex M processor family. These platforms have very simple cores, mainly optimized for low power and cost. They are cache-less and do not support memory virtualization, limiting them to either bare-metal applications or very basic operating systems. Full-blown DVFS is generally not supported for cost reasons: a DVFS-ready switching-mode power converter (SMPS) is a complex and expensive component usually optimized for efficiency at high currents, and featuring a complex SW interface for voltage control. In the low-complexity, low-power domains the SMPS's have limited tuning range for cost and efficiency reasons. More flexible and inexpensive LDO's have significant losses when downconverting, which negates most of the savings expected by voltage scaling. Hence, these low-end systems typically rely on frequency scaling and shutdown as the main power management knobs. Furthermore, logic synthesis is problematic for low-power IC's with ultra-wide voltage and frequency scaling, since the behavior of synthesis tools for timing optimization at a high voltage and an ultra-low-voltage are so conflicting that convergence in one scenario can create violations in other scenarios [29]. This translates to further inefficiencies in the final device.

At the same time, multi-core architectures are beginning to penetrate the microcontroller business segment: recently, a new class of heterogeneous dual-core MCU products appeared in the market. These devices have cores with different instruction set architectures [26] and rely on the architectural heterogeneity to achieve energy efficiency with a principle similar to the mid-to-high-end Big-Little multi-cores described previously, but with some limitations. The restrictions imposed by the simple nature of the processor architectures, namely the lack of first level caches and support for virtualization, impose two big restrictions in the task allocation policies. First, tasks must be statically partitioned at design time, since processors execute different instruction sets. Second, the limited-performance nature of MCUs does not support the computational load required to implement complex feedback loop allocation policies, as well as voltage and frequency tuning. In recent years, the semiconductor industry has started to reduce model guardbands as a way to decrease core area and increase energy efficiency, at the cost of reliability. Though the economic viability of relaxed process variations is still an open

Download English Version:

<https://daneshyari.com/en/article/4956693>

Download Persian Version:

<https://daneshyari.com/article/4956693>

[Daneshyari.com](https://daneshyari.com)