INTEGRATION the VLSI journal xxx (xxxx) xxx–xxx

Contents lists available at ScienceDirect

INTEGRATION, the VLSI journal

journal homepage: www.elsevier.com/locate/vlsi

Mixed-criticality scheduling on heterogeneous multicore systems powered by energy harvesting

Yi Xiang[a], Sudeep Pasricha[b,*]

[a] Synopsys Inc., United States
[b] Colorado State University, Fort Collins, United States

ARTICLE INFO

Keywords:
Energy harvesting
Task scheduling
Real-time systems
Mixed-criticality
Heterogeneous platforms

ABSTRACT

In this paper, we address the scheduling problem for single-ISA heterogeneous multicore processors running hybrid mixed-criticality workloads with a limited and fluctuating energy budget provided by solar energy harvesting. The hybrid workloads consist of a set of firm-deadline timing-centric applications and a set of soft-deadline throughput-centric multithreaded applications. Our framework exploits traits of the different types of cores in heterogeneous multicore systems to service timing-centric workloads with a few big out-of-order cores, while servicing throughput-centric workloads with many smaller in-order cores clocked in the energy-efficient near-threshold computing (NTC) region. Guided by a novel timing intensity metric, our mixed-criticality scheduling framework creates an optimized schedule that minimizes overall miss penalty for a time-varying energy budget. Experimental results indicate that our framework achieves a 9.5% miss penalty reduction with the proposed timing intensity metric compared to metrics from prior work, a 13.6% performance improvement over a state-of-the-art scheduling approach for single-ISA heterogeneous platforms, and a 23.2% performance benefit from exploiting platform heterogeneity.

1. Introduction

Recent years have seen billions of embedded systems deployed around the world to support a variety of different applications domains. For an increasing number of embedded applications, there is a critical need for energy autonomous devices that can utilize ambient energy from the environment to perform computations without relying on an external power supply or frequent battery charges. Solar energy has been considered as one of most important source of ambient energy in research on management of harvesting-aware embedded systems [1–6], as the advancement of photovoltaic energy harvesting technologies at various scales have paved the way to deploy such embedded system types more widely [53–55].

Embedded computing systems that include timing behavior as part of their performance or correctness criteria are called real-time embedded systems. In such real-time systems, a deadline is called *firm* if missing it results in an immediate performance penalty, otherwise the deadline is considered to be *soft*. If critical system failure can happen after a deadline miss, the deadline is considered to be a *hard* deadline [7]. Due to the variable nature of solar radiation intensity, the most suitable role of embedded systems with solar energy harvesting as the only energy source is to host applications without strict real-time requirements. Thus it may not be desirable to consider such systems for real-time applications with *hard* deadlines, such as for life-support systems, automotive system control, aircraft navigation, etc., for which any deadline miss results in a critical system failure that may have catastrophic consequences. Instead, it is more practical to deploy such systems without energy guarantees, for best-effort execution of applications, where a *firm* or *soft* deadline miss is not considered a failure of the entire system.

Consider such a best-effort embedded system powered by energy harvesting, deployed for continuous data collection, data post-processing, and data transmission at a remote location. For each operation interval (e.g., every few minutes), a data point can be recorded from sensor modules by executing certain control tasks, for which a missed deadline results in inaccuracy in the averaged values of data features. Such tasks can be considered to be *timing-centric with firm deadlines*. On the other hand, post-processing of raw data and data transmission tasks can be delayed somewhat as the system can buffer a certain amount of data or clients can accept a lower rate of transmitted data. Such tasks are generally *throughput-centric with soft deadlines*. In this paper, we represent such applications (with different levels of real-time constraints) as mixed-criticality workloads that consist of a mix of timing-centric tasks with firm deadlines and throughput-centric tasks with soft deadlines [8,9].

---

* Corresponding author.
E-mail addresses: yi.xiang@synopsys.com (Y. Xiang), sudeep@colostate.edu (S. Pasricha).

https://doi.org/10.1016/j.vlsi.2017.11.007
Received 28 April 2017; Received in revised form 26 September 2017; Accepted 27 November 2017
0167-9260/ © 2017 Published by Elsevier B.V.

Please cite this article as: Xiang, Y., INTEGRATION the VLSI journal (2017), https://doi.org/10.1016/j.vlsi.2017.11.007

Recent years have also seen the rise of multicore processing in low-power embedded devices [10]. Due to the increasing demand for performance in emerging embedded applications, we are starting to see manycore platforms with greater core counts while maintaining energy-efficiency. As an example, MediaTek's Helix ×30 for mobile embedded applications released in 2017 [12] is a heterogeneous architecture with 10 heterogeneous ARM cores. For mixed-criticality applications, ARM is rolling out Cortex-R52 [52], a new generation of embedded processor with up to 4 cores and improved performance to address mixed-criticality workloads. Multiple Cortex-R52 processors can be integrated into a SoC, together with other ARM processor variants, for greater performance in embedded applications.

Multicore processors with heterogeneous cores have also been shown to provide substantial improvements in energy-efficiency and performance for energy-constrained systems [11]. Thus the pairing of manycore and heterogeneous computing is not a privilege for large-scale high-performance systems anymore, whose adoption to embedded processors can be seen in the big. LITTLE architecture from ARM that is continuously evolving with increased core counts [10]. With the rise in computing capabilities of emerging heterogeneous multicore processors, run-time workload distribution and energy-management in these architectures are becoming crucial steps towards minimizing the overall system energy consumption while maximizing achievable application performance. Heterogeneous platforms are particularly well-suited to execute mixed-criticality workloads as different types of cores can be utilized to better match specific criticality requirements of different types of tasks.

In addition to multiprocessing and heterogeneous computing, a new design paradigm has emerged to further help minimize energy in contemporary chip designs, called near-threshold computing (NTC) [13–18]. In NTC, the supply voltage is set just slightly higher than threshold voltage, and execution at this NTC mode achieves several times better energy-efficiency than conventional super-threshold computing (STC) [16] operation modes. NTC is thus a very effective strategy to minimize energy for energy-constrained embedded systems. However, as NTC mode operation typically sacrifices performance in favor of energy-efficiency, it is not straightforward to use it for mixed-criticality real-time systems with timing constraints.

In this paper, we focus on the important problem of *design and management of STC/NTC capable heterogeneous multicore platforms powered by solar energy harvesting and running mixed-criticality workloads, to optimize cost, performance and energy efficiency of such systems*. We propose a novel mixed-criticality scheduling framework (*McSF*), that for the first time, addresses the problem of allocating and scheduling workloads with different degrees of criticality on a heterogeneous multicore embedded system powered by energy harvesting and supporting NTC operation. Our framework employs NTC for *throughput-centric tasks* with loose timing constraints and a high degree of parallelism (DoP), maintaining their computation throughput by executing their threads concurrently on many cores in an energy-efficient manner. By improving the energy-efficiency for throughput-centric tasks, more energy budget becomes available for *timing-centric tasks*, which are allocated with awareness of harvested energy fluctuations. The novel contributions of our work can be summarized as follows:

1. Unlike any prior work, we formulate and solve the challenging problem of scheduling mixed-criticality, real-time applications on heterogeneous energy-harvesting embedded system platforms;
2. A hybrid mapping and scheduling framework is proposed to offload scheduling complexity of timing-centric task graphs to a comprehensive design-time methodology so that only lightweight adjustments are required at run-time (e.g., selecting among a small set of schedule templates, core operation modes, and task DoPs) to cope with changing energy harvesting scenarios over time;
3. For efficient execution of throughput-centric tasks, we utilize near-threshold computing (*NTC*) on several small cores to maintain high throughput levels without sacrificing energy efficiency of the computation;
4. A new energy-aware priority metric, *timing intensity-aware penalty density*, is proposed to dynamically measure the importance of instances of different task criticality types within a mixed-criticality workload.
5. Our experiments analyze the proposed mixed-criticality scheduling framework and show notable improvements in miss penalty reduction and overall performance improvement compared to a state-of-the-art scheduling approach for single-ISA heterogeneous platforms.

2. Related work

In this section, we review prior work that is relevant to our contribution in this paper. First we discuss scheduling algorithms for single-core energy harvesting embedded systems, followed by scheduling algorithms for multi-core energy harvesting embedded systems. Next we discuss work on near-threshold computing (NTC) to save energy in emerging computing platforms. Lastly, we discuss efforts to schedule mixed-criticality workloads on various platforms.

Several prior efforts have explored workload scheduling for embedded systems with solar energy harvesting, e.g., [2–6,19–22]. An early work by Moser et al. proposed the lazy scheduling algorithm that executed tasks as late as possible, reducing task deadline miss rates compared to the classical earliest deadline first algorithm [2]. However, for low-power systems with frequency scaling capabilities, delaying task execution can lead to higher frequency requirements and thus lower energy-efficiency. Liu et al. [3] used frequency scaling to slow down the execution speed of arriving tasks as evenly as possible, to save energy. In [4], a utilization-based technique was proposed for periodic task scheduling in energy-harvesting embedded systems. Zhang et al. [5] combined task scheduling with a power model of a DC-DC converter to adjust the workload for higher charging efficiency. Chetto [19] proposed a semi-online EDF-based scheduling algorithm that is theoretically optimal. *However, these works only consider single-core systems and simple independent task models with no inter-task dependencies.*

Xiang et al. [6,21] proposed a framework to manage task execution on energy harvesting powered homogeneous multicore systems with a hybrid battery/supercapacitor for energy storage. The work addressed the scheduling problem for independent periodic tasks, with awareness of the on-chip temperature profile and process variations. Xiang et al. [22] also proposed a framework to schedule task graphs on homogeneous multicore embedded systems and react to soft errors at run-time to reduce their impact on performance. Zhang et al. [20] introduced a deadline-aware scheduling algorithm with energy migration strategies specifically designed to manage distributed supercapacitors in sensor networks. *None of these prior works on scheduling for embedded systems with solar energy harvesting consider the scheduling problem for heterogeneous multicore systems, utilize the NTC execution paradigm, or support mixed-criticality workloads.*

The high energy-efficiency achievable with near-threshold computing (NTC) and its design challenges are discussed in [13]. In [14] NTC is explored as a way to address the power density problem for 3D-stacked chips. As NTC systems tend to be more sensitive to process variations with their lower supply voltage, a few recent works propose novel management techniques for NTC to alleviate the performance impact of process variations [15–17]. More recently, Karpuzcu et al. [18] proposed Accordion, a framework that executes workloads with adjustable problem sizes and fault resilience on NTC-enabled cores. Chen et al. [23] studied the impact of NTC on architectural design of processors by analyzing resulting shifts in performance bottlenecks. But to the best of our knowledge, *no prior work has addressed the scheduling problem for NTC-enabled cores powered by energy harvesting. Moreover prior work has also not considered allocation of mixed criticality workloads on heterogeneous NTC-capable platforms.*

Download English Version:

https://daneshyari.com/en/article/6942162

Download Persian Version:

https://daneshyari.com/article/6942162

Daneshyari.com