Microprocessors and Microsystems 40 (2016) 74–87

Contents lists available at ScienceDirect

Microprocessors and Microsystems

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

A generic energy optimization framework for heterogeneous platforms using scaling models

Ujjwal Gupta *, Spurthi Korrapati, Navyasree Matturu, Umit Y. Ogras

School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ 85281, United States

ARTICLE INFO

Keywords:
Energy optimization
Heterogeneous architectures
Mobile platforms
MpSoC
Performance

ABSTRACT

Mobile platforms are becoming highly heterogeneous by combining a powerful multiprocessor system-on-a-chip (MpSoC) with numerous other resources, including display, memory, power management IC, battery and wireless modems into a compact package. Furthermore, the MpSoC itself is a heterogeneous resource that integrates many processing elements such as CPU cores, GPU, video, image, and audio processors. Platform energy consumption and responsiveness are two major considerations for mobile systems, since they determine the battery life and user satisfaction, respectively. As a result, energy minimization approaches targeting mobile computing need to consider the platform at various levels of granularity. In this paper, we first present power consumption, response time, and energy consumption models for mobile platforms. Using these models, we optimize the energy consumption of baseline platforms under power, response time, and thermal constraints with and without introducing new resources. Finally, we validate the proposed framework through experiments on Qualcomm's Snapdragon 800 Mobile Development Platforms.

1. Introduction

Increasing computational capabilities at mobile form factor, and integration of new heterogeneous resources, such as gyroscopes, are leading to emerging scenarios like navigation and 4 K displays [42]. In turn, rapidly evolving embedded systems are being driven by emerging applications that demand higher energy efficiency and performance. Consequently, embedded systems have become highly heterogenous computing platforms that host a variety of resources including a processor, display, flash memory, DRAM, baseband and radio frequency chips, power management IC, voltage regulators, camera, touch panel, and battery, as shown in Fig. 1a. Unlike traditional PC and server systems, the role of a CPU in mobile systems is replaced with a MpSoC, also called an application processor. The application processor itself is a heterogenous resource that integrates many processing elements (PEs), e.g., graphics cores, memory controllers, video and audio accelerators, security engines, sensors, as well as multiple CPU cores on a single die [25,26], as shown in Fig. 1b. While the CPU cores still orchestrate the operation of other PEs, they neither dominate the cost [35], nor determine the power consumption and performance under many application scenarios [7]. In particular, the display contributes to a significant portion of power consumption (∼30%) across a wide range of application use cases, such as, video playback, and simultaneous browsing/audio playback, as illustrated in Fig. 2a. This paper broadens the research focus to pay attention to the *platform as a whole* to minimize energy consumption rather than focusing on a subset.

Despite the richness of the underlying hardware platform only a subset of the resources are invoked during the lifetime of an application. For example, a navigation application goes through a number of phases, as illustrated in Fig. 2b. After a user launches the application, the core running the OS fetches the host code from the memory and starts execution. Next, it triggers the accelerator (the GPS module), which loads the data from the memory, performs the assigned task, and writes the results back. Hence, the run-time and power consumption of different application phases depend on the particular resource employed, while the synchronization overhead is determined by the communication network and I/O speed. Furthermore, the latency to reach the memory through the interconnection network and memory controller, as well as the memory access time are additive to the total time. Hence, the response-time for the GPS example can be obtained as, $t_{rsp} = t_{CPU} + t_{comm} + t_{mem} + t_{GPS}$ where the terms represent the times contributed by the CPU, communication, memory, and GPS module. Likewise, power consumption is determined by the active resources during the lifetime of the application and their particular power states. Hence, the ability of the dynamic power management algorithms to put different PEs to sleep or into

* Corresponding author.
E-mail addresses: ujjwal@asu.edu (U. Gupta), umit@asu.edu (U.Y. Ogras).

http://dx.doi.org/10.1016/j.micpro.2015.06.009
0141-9331/

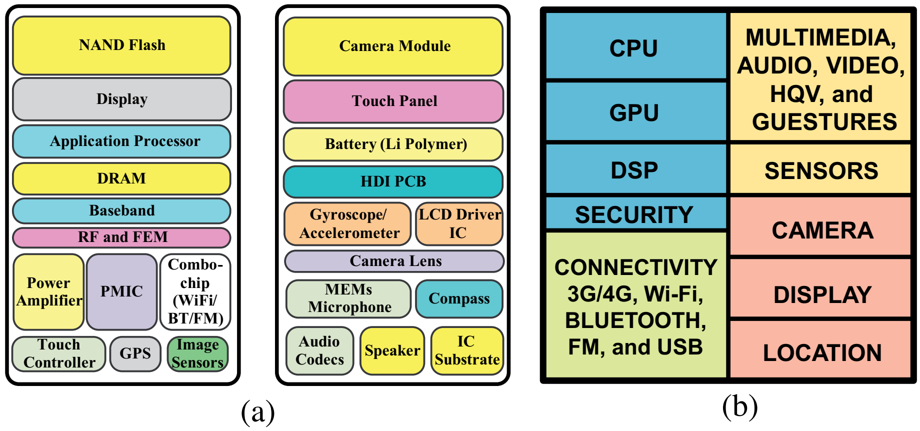

Fig. 1. (a) A heterogenous mobile platform consisting of several components/resources [35]. Application processor is only one of the resources, not dominating in cost or power. (b) Qualcomm® Snapdragon processor's processing elements [42].

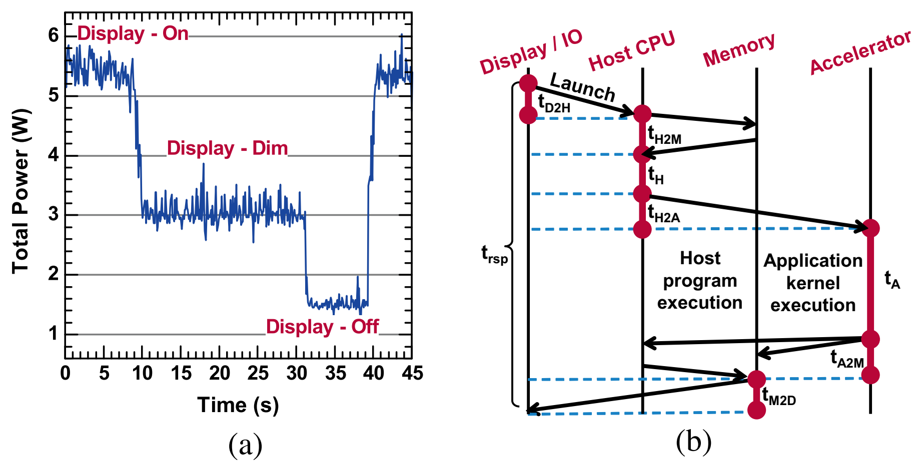

Fig. 2. (a) Example of total power consumed in Qualcomm Snapdragon 800 MDP for full, dimmed, and off display brightness. For a detailed explanation, refer to Section 3.2.1. (b) Illustration of the resource centric view.

other low power states has a significant impact on the total power consumption.

This paper presents an energy optimization framework for heterogeneous mobile platforms. The proposed framework enables reasoning about optimization, both at the MpSoC and platform level, by exposing the impact of each resource on power, performance, and energy. It can be used for platform exploration to determine the type and characteristics of resources that should be employed, or for finding the optimal operating conditions of a given hardware configuration. It is also possible to utilize the proposed framework to *quantify* the energy savings that can be obtained by adding new resources or by replacing existing ones with more energy efficient resources. This is achieved by employing a resource centric view in two dual energy optimization formulations. The first formulation minimizes the total energy drawn from the battery with a constraint on the response-time, while the second formulation puts a constraint on the power consumption. Hence, the proposed framework aims to ensure responsiveness and best use of a fixed power budget while minimizing total energy consumption. Our major contributions are as follows:

- We present a framework that provides quantitative results of energy savings obtained by adding new resources or replacing existing ones with more energy efficient resources.
- The proposed framework enables co-optimization of the platform resources at once and demonstrates that co-optimization is superior to optimizing the resources one-by-one.
- We develop a detailed experiment methodology for collecting reliable experimental data, and perform an extensive set of experiments using Qualcomm's mobile development platforms (MDP) [41].

Other contributions include a comparison of the energy savings obtained using the proposed framework with on-demand governor [37] during run-time in the MDP smart-phone. The results presented in this paper not only confirm that voltage and frequency scaling offers limited improvements in energy efficiency due to the inverse relationship between power consumption and performance, but that they also provide *precise numeric evaluations*. We also show how the presented performance model can be used to generalize Amdahl's law similar to [8,33].

The rest of this paper is organized as follows. Related research is presented in Section 2. Energy optimization framework and generalization of Amdahl's law are presented in Section 3. Experimental methodology and results are presented in Sections 4 and 5, respectively. Section 6 concludes the paper and summarizes the future directions.

2. Related work

Power consumption has been a major architectural design consideration for many years [34]. This has led to the study of energy efficient techniques to harness the processing power within the power and thermal budgets [10,13,50]. For example, due to the